i.MX 8M Dual/8M QuadLite/8M Quad Applications Processors Reference Manual

Document Number: IMX8MDQLQRM
Rev. 2, 08/2019

Contents

Section number **Title** **Page**

Chapter 1
Introduction

Chapter 2
Memory Map

Chapter 3
Security

Chapter 4
ARM Platform and Debug

Chapter 14
Video Processing Unit (VPU)

Chapter 15
Display Controller Subsystem (DCSS)

Chapter 16
Low Speed Communication and Interconnects

Chapter 1
Introduction

1.1 Product Overview

This chapter introduces the architecture of the i.MX 8M Dual/8M QuadLite/8M Quad Applications Processors.

The i.MX 8M Dual/8M QuadLite/8M Quad is a family of products focused on delivering an excellent 4K video and audio experience, combining media-specific features with high-performance processing optimized for low-power consumption.

1.2 Target Applications

The i.MX 8M Dual/8M QuadLite/8M Quad Applications Processors achieve both high performance and low power consumption and rely on a powerful, fully-coherent core complex based on a quad Cortex-A53 cluster, with graphics processing GPU supporting the latest graphic APIs.

The i.MX 8M Family provides additional computing resources and peripherals:

- Advanced security modules for secure boot, cipher acceleration and DRM support
- General purpose Cortex-M4 processor for low power processing
- A wide range of audio interfaces including I2S, AC97, TDM and S/PDIF
- Large set of peripherals that are commonly used in consumer/industrial markets including USB 3.0, PCIe and Ethernet

1.3 Acronyms and Abbreviations

The table below contains acronyms and abbreviations used in this document.

Acronyms and Abbreviated Terms

Term	Meaning
ADC	Analog-to-Digital Converter
AHB	Advanced High-performance Bus
AIPS	Arm IP Bus
ALU	Arithmetic Logic Unit
AMBA	Advanced Microcontroller Bus Architecture
APB	Advanced Peripheral Bus
ASRC	Asynchronous Sample Rate Converter
AXI	Advanced eXtensible Interface
BIST	Built-In Self Test
CA/CM	Arm Cortex-A/Cortex-M
CAAM	Cryptographic Acceleration and Assurance Module
CAN	Controller Area Network
CCM	Clock Controller Module
CPU	Central Processing Unit
CSI	CMOS Sensor Interface
CSU	Central Security Unit
CTI	Cross Trigger Interface
DAP	Debug Access Port
DDR	Double data rate
DMA	Direct memory access
DPLL	Digital phase-locked loop
DRAM	Dynamic random access memory
ECC	Error correcting codes
ECSPI	Enhanced Configurable SPI
LPSPI	Low-power SPI
EDMA	Enhanced Direct Memory Access
EIM	External Interface Module
ENET	Ethernet
EPIT	Enhanced Periodic Interrupt Timer
EPROM	Erasable Programmable Read-Only Memory
ETF	Embedded Trace FIFO
ETM	Embedded Trace Macrocell
FIFO	First-In-First-Out
GIC	General Interrupt Controller
GPC	General Power Controller
GPIO	General-Purpose I/O
GPR	General-Purpose Register
GPS	Global Positioning System
GPT	General-Purpose Timer
GPU	Graphics Processing Unit
GPV	Global Programmers View

Table continues on the next page...

Term	Meaning
HAB	High-Assurance Boot
I2C or I^2C	Inter-Integrated Circuit
IC	Integrated Circuit
IEEE	Institute of Electrical and Electronics Engineers
IOMUX	Input-Output Multiplexer
IP	Intellectual Property
IrDA	Infrared Data Association
JTAG	Joint Test Action Group (a serial bus protocol usually used for test purposes)
LCDIF	Liquid Crystal Display Interface
LDO	Low-Dropout
LIFO	Last-In-First-Out
LRU	Least-Recently Used
LSB	Least-Significant Byte
LUT	Look-Up Table
LVDS	Low Voltage Differential Signaling
MAC	Medium Access Control
MCM	Miscellaneous control Module
MMC	Multimedia Card
MSB	Most-Significant Byte
MT/s	Mega Transfers per second
OCRAM	On-Chip Random-Access Memory
OCOTP	On-Chip One-Time Programmable Controller
PCI	Peripheral Component Interconnect
PCIe	PCI express
PCMCIA	Personal Computer Memory Card International Association
PGC	Power Gating Controller
PIC	Programmable Interrupt Controller
PMU	Power Management Unit
POR	Power-On Reset
PSRAM	Pseudo-Static Random Access Memory
PWM	Pulse Width Modulation
PXP	Pixel Pipeline
QoS	Quality of Service
R2D	Radians to Degrees
RISC	Reduced Instruction Set Computing
ROM	Read-Only Memory
ROMCP	ROM Controller with Patch
RTOS	Real-Time Operating System
Rx	Receive
SAI	Synchronous Audio Interface
SCU	Snoop Control Unit

Table continues on the next page...

Term	Meaning
SD	Secure Digital
SDIO	Secure Digital Input/Output
SDLC	Synchronous Data Link Control
SDMA	Smart DMA
SIM	Subscriber Identification Module
SNVS	Secure Non-Volatile Storage
SoC	System-on-Chip
SPBA	Shared Peripheral Bus Arbiter
SPDIF	Sony Phillips Digital Interface
SPI	Serial Peripheral Interface
SRAM	Static Random-Access Memory
SRC	System Reset Controller
TFT	Thin-Film Transistor
TPIU	Trace Port Interface
TSGEN	Time Stamp Generator
Tx	Transmit
TZASC	TrustZone Address Space Controller
UART	Universal Asynchronous Receiver/Transmitter
USB	Universal Serial Bus
USDHC	Ultra Secured Digital Host Controller
WDOG	Watchdog
WLAN	Wireless Local Area Network
WXGA	Wide Extended Graphics Array

1.4 Features

1.4.1 Arm Cortex-A53 MPCore™ Platform

The i.MX 8M Family Applications Processors are based on the Arm Cortex-A53 MPCore™ Platform, which has the following features:

- Quad symmetric Cortex-A53 processors, including:
  - 32 KB L1 Instruction Cache
  - 32 KB L1 Data Cache
  - Media Processing Engine (MPE) with NEON technology supporting the Advanced Single Instruction Multiple Data architecture
  - Floating Point Unit (FPU) with support of the VFPv4-D16 architecture

- Support of 64-bit Armv8-A architecture
- 1 MB unified L2 cache
- Target frequency of 1.4GHz

1.4.2 Arm Cortex-M4 Platform

Cortex-M4 Core Platform include the following:

- Low power microcontroller available for customer application:
  - Low power standby mode
  - IoT features including Weave
  - Manage IR or wireless remote
- Arm Cortex M4 CPU Processor, including:
  - 16 KB L1 Instruction Cache
  - 16 KB L1 Data Cache
  - 256 KB TCM
  - Target frequency of 266MHz

1.4.3 System Bus and Interconnect

System bus and interconnect include the following:

- Network interconnect (NoC) AXI arbiter
- Quality of service controller (QoSC) to configure priorities and limits of AXI transcations
- Performance monitor (PERFMON) to monitor AXI bus activity
- Debug monitor (DBGMON) to record AXI transactions preceding a system reset

1.4.4 Clocking and Resets

Clocking and resets include:

- Clock control module (CCM) provides centralized clock generation and control
  - Simplified clock tree structure
  - Unified clock programming model for each clock root
  - Multicore awareness for resource domains
- System reset controller (SRC) provides reset generation and distribution

1.4.5 Interrupts and DMA

Interrupts and DMA include:

- 128 shared peripheral interrupts routed to Cortex-A53 Global Interrupt Controller (GIC) and Cortex-M4 nested vector interrupt controller (NVIC) for flexible interrupt handling
- Two Smart direct memory access (SDMA) engines. Although these two engines are identical to each other, they are integrated into the processor to serve different peripherals. Each SDMA controller supports 48 DMA requests

1.4.6 On-Chip Memory

The on-chip memory system consists of the following:

- Boot ROM (96KB)
- On-chip RAM (128KB + 32KB)

1.4.7 External Memory Interface

The external memory interfaces supported on this chip include:

- 16/32-bit DRAM Interface:
  - PoP LPDDR4-3200
  - Non-PoP LPDDR4-3200, DDR4-2400, DDR3L-1600
- 8-bit NAND FLASH, including support for Raw MLC/SLC devices, BCH ECC up to 62-bit, and ONFi3.2 compliance (clock rates up to 100 MHz and data rates up to 200 MB/sec)
- eMMC 5.1 FLASH (2 interfaces)
- SPI NOR FLASH (3 interfaces)
- FlexSPI FLASH with support for XIP (for M4 in low-power mode) and parallel read mode of two identical FLASH devices

1.4.8 Timers

The timers on this chip include:

- One local generic timer integrated into each Cortex-A53 CPU
- Global system counter with timer bus interface to Cortex-A53 MPCore generic timers
- One local system timer (SysTick) integrated into the Cortex-M4 CPU
- Six general purpose timer (GPT) modules
- Three watchdog timer (WDOG) modules
- Four pulse width modulation (PWM) modules

1.4.9 Graphics Processing Unit (GPU)

The chip incorporates the following Graphics Processing Unit (GPU) features:

- 4 shader
- Supports OpenGL ES 3.1, 3.0, 2.0, 1.0, OpenCL
- Target frequency of 800 MHz
- Frame Buffer Compression – Lossless compression of buffers
- TrustZone support using a local MMU to manage secure regions

1.4.10 Video Processing Unit (VPU)

The chip incorporates the following Video Processing Unit (VPU) features:

- VP9 Profile 0, 2 (10 bit) decoder (VPU G2)
- HEVC/H.265 Main, Main10 decoder (VPU G2)
- AVC/H.264 Baseline, Main, High decoder (VPU G1)
- VP8 decoder (VPU G1)
- Frame Buffer Compression – Lossless compression of buffers
- TrustZone support

1.4.11 Display Interfaces

The chip has the following display support:

- LCD Interface with MIPI-DSI Output:
  - MIPI-DSI 4 channels supporting one display. Supports resolution up to 1920x1080p60, 1800x1200p60
  - LCDIF display controller
  - Output can be LCDIF output or Display Controller output (turns off HDMI)
- Two MIPI-CSI2 Display Interfaces:

- Each MIPI-CSI2 is 4 channels supporting one camera input. Supports 5M pixel at 15 fps, 1080p30, 720p60, VGA at 60 fps
- Maximum bit rate of 1.5 Gbps

1.4.12 Audio

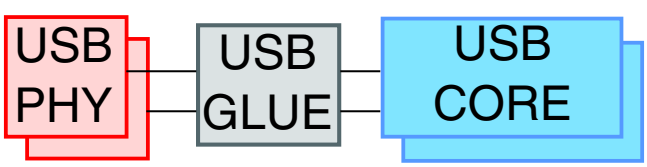
Audio include the following:

- S/PDIF Input and Output, including a Raw Capture input mode
- Five SAI (synchronous audio interface) modules supporting I2S, AC97, TDM, and codec/DSP interfaces, including one SAI with 8 TX and 8 RX lanes, one SAI with 4 TX and 4 RX lanes, and three SAI with 1 TX and 1 RX lanes. Supports over 20 channels of audio
- One internal SAI port with 4 I2S lanes to drive HDMI audio output
- One internal S/PDIF interface to capture HDMI ARC at up to 192kHz Fs

1.4.13 General Connectivity Interfaces

The chip contains a rich set of general connectivity interfaces, including:

- Two PCI Express (PCIe):
  - Single lane supporting PCIe Gen 2
  - Dual mode operation to function as root complex or endpoint
  - Integrated PHY interface
  - Supports L1 substate
- Two USB 3.0/2.0 OTG controllers with integrated PHY interface



  - USB host mode, support SS/HS/FS/LS
  - USB device mode, support SS/HS/FS
  - USB2.0 OTG mode, ADP/SRP/HNP
  - Battery charger
  - Spread spectrum clock support
- Two Ultra Secure Digital Host Controller (uSDHC) interfaces
  - MMC 5.1 compliance with HS400 DDR signaling to support up to 400 MB/sec

    - SD/SDIO 3.01 compliance with 200 MHZ SDR signaling to support up to 100 MB/sec
    - Support for SDXC (extended capacity)
- One Gigabit Ethernet controller with support for EEE, Ethernet AVB and IEEE1588
- Four universal asynchronous receiver/transmitter (UART) modules
- Four I2C modules
- Three SPI modules

1.4.14 Security

Security functions are enabled and accelerated by the following hardware:

- RDC – Resource Domain Controller:
    - Supports 4 domains and up to 8 regions
- Arm TrustZone including the TZ architecture:
    - ARM Cortex-A53 MPCore TrustZone support
- On-chip RAM (OCRAM) secure region protection using OCRAM controller
- High Assurance Boot (HAB)
- Cryptographic Acceleration and Assurance Module (CAAM)
    - Support Widevine and PlayReady content protection
    - Public Key Cryptography (PKHA) with RSA and Elliptic Curve (ECC) algorithms
    - Real-time integrity checker (RTIC)
    - DRM support for RSA, AES, 3DES, DES
    - Side channel attack resistance
    - True random number generation (RNG)
    - Manufacturing protection support
- Secure Non-Volatile Storage (SNVS), including
    - Secure Real Time Clock (SRTC)
- Secure JTAG Controller (SJC)

1.4.15 Multicore Support

Multicore support contains:

- Resource domain controller (RDC) to support isolation and safe sharing of system resources
- Messaging unit (MU)

- Hardware Semaphore (SEMA42)
- Shared bus topology

1.4.16 GPIO and Pin Multiplexing

- General-purpose input/output (GPIO) modules with interrupt capability
- Input/output multiplexing controller (IOMUXC) to provide centralized pad control

1.4.17 Power Management

The power management unit consists of:

- Temperature sensor with programmable trip points
- Flexible power domain partitioning with internal power switches to support efficient power management

1.4.18 System Debug

The system debug features are:

- ARM CoreSight debug and trace architecture
- Trace Port Interface Unit (TPIU) to support off-chip real-time trace
- Embedded Trace FIFO (ETF) with 4 KB internal storage to provide trace buffering
- Unified trace capability for Quad Cortex-A53 and Cortex-M4 CPUs
- Cross Triggering Interface (CTI)
- Support for 5-pin (JTAG) debug interfaces

1.5 Architectural Overview

This section contains the i.MX 8M Dual/8M QuadLite/8M Quad architectural details.

1.5.1 Block Diagram

The high-level block diagram is shown in the figure below.

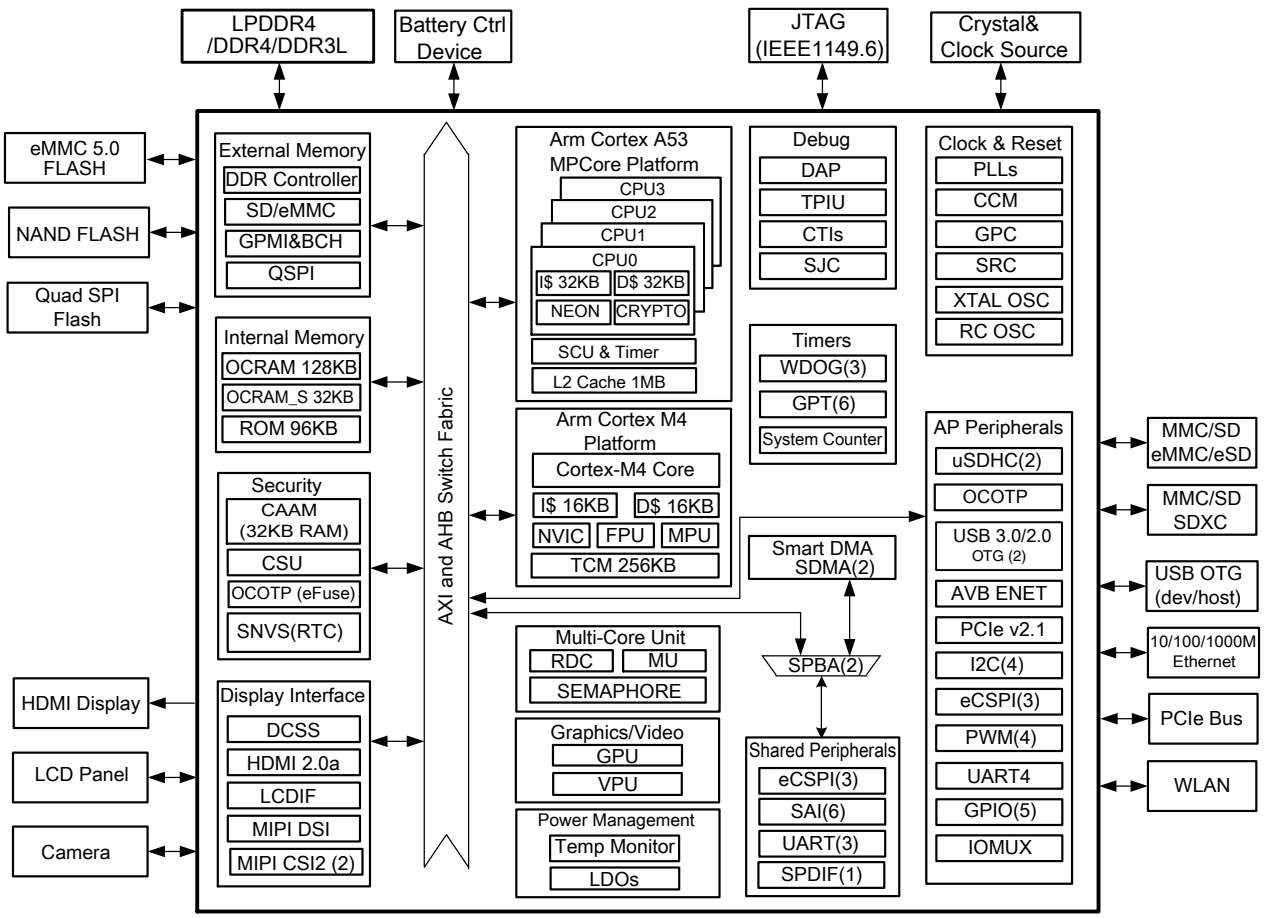


Figure 1-1. Block Diagram

1.6 Primary Boot Options

The i.MX 8M Dual/8M QuadLite/8M Quad supports the following boot devices:

- NAND FLASH (including SLC and MLC)
- SDIO / MMC / SDXC
- eSD 3.0/eMMC 5.1 (fast boot)
- SPI (serial FLASH)
- USB
- QSPI
- Ethernet (via plug-in mode)

The Quad-A53 core on i.MX 8M Dual/8M QuadLite/8M Quad is enabled during boot as the primary core to handle the entire secure boot flow. The chip will always boot from the A53 core first, the M4 core will be held in reset during the A53 boot and won't run until it is enabled by the A53 core. The image for the M4 core will be loaded into memory and authenticated by the A53 core.

1.7 Endianness Support

This chip supports Little Endian mode only.

Chapter 2
Memory Map

2.1 Memory

This chapter introduces the memory architecture of the chip. The system memory high-level partition is defined below:

2.1.1 Memory system overview

2.1.1.1 On-chip L1, L2 caches, TCM

Cortex-A53 MPcore Platform

- Level 1 Cache (4x per Cortex-A53 Core)
  - Instruction (32 KB)
  - Data (32 KB)
- Level 2 Cache, shared by the four Cortex-A53 cores:
  - Unified instruction and data (1 MB)

Cortex M4 Platform

- Cache
  - Instruction (16 KB)
  - Data (16 KB)
- Tightly-Coupled-Memory
  - TCML on Code Bus (128KB)
  - TCMH on System Bus (128KB)

2.1.1.2 On-chip memories

- Boot ROM (128 KB)

- On-Chip RAM - OCRAM (128 KB)
- On-Chip RAM for State Retention - OCRAM_S (32 KB)

2.1.1.3 External L3 memories

The chip supports external memories, via the following memory interfaces / controllers:

- 16/32-bit DRAM Interface:
  - PoP LPDDR4-3200, Non-PoP LPDDR4-3200, DDR4-2400, DDR3L-1600
- 8-bit NAND FLASH, including support for Raw MLC/SLC devices, BCH ECC up to 62-bit, and ONFI 2.x compliance (clock rates up to 100 MHz and data rates up to 200 MB/sec)
- eMMC 5.0 FLASH
- SPI NOR FLASH
- Quad SPI FLASH with support for XIP (for M4 in low-power mode) and parallel read mode of two identical FLASH devices

2.1.2 Cortex-A53 Memory Map

Start Address	End Address	Region	Size	Description
1_0000_0000	1_3FFF_FFFF	DDR Address	1024MB	DDR Memory (Quad-A53 only)
4000_0000	FFFF_FFFF	DDR Address	3072MB	DDR Memory (All modules)
3E00_0000	3FFF_FFFF	Reserved	32MB	Reserved
3DC0_0000	3DFF_FFFF	DDRC	4MB	Reserved
3D80_0000	3DBF_FFFF		4MB	DDR PERF_MON
3D40_0000	3D7F_FFFF		4MB	DDR CTL
3C00_0000	3D3F_FFFF		20MB	DDR PHY
3890_0000	3BFF_FFFF	Reserved	55MB	Reserved
3880_0000	388F_FFFF	GIC	1MB	GIC REG
3870_0000	387F_FFFF	Reserved	1MB	Reserved
3830_0000	384F_FFFF	VPU	2MB	VPU_Decoder
3820_0000	382F_FFFF	USB	1MB	USB2 REG
3810_0000	381F_FFFF		1MB	USB1 REG
3801_0000	380F_FFFF	Reserved	960KB	Reserved
3800_0000	3800_FFFF	GPU REG	64KB	GPU REG
3600_0000	37FF_FFFF	QSPI RX Buffers	32MB	Reserved
3400_0000	35FF_FFFF		32MB	QSPI1 RX Buffer
3390_0000	33FF_FFFF	Reserved	7.9MB	Reserved
3381_4000	3381_FFFF	PCIe REG	48KB	Reserved

Table continues on the next page...

Start Address	End Address	Region	Size	Description
3381_0000	3381_3FFF		4MB	PCIe2 REG
3380_4000	3380_FFFF		48KB	Reserved
3380_0000	3380_3FFF		4MB	PCIe1 REG
3310_0000	337F_FFFF	MMAP Peipherals	7960KB	Reserved
3300_8000	3300_FFFF		32KB	Reserved (APBH DMA)
3300_0000	3300_7FFF		32KB	APBH DMA
32C0_0000	32FF_FFFF	Periph (AIPS)	4MB	AIPS4
3280_0000	32BF_FFFF	Reserved	4MB	Reserved
3270_0000	327F_FFFF	GPV_7	1MB	NoC configuration
3260_0000	326F_FFFF	GPV_6	1MB	"m4" configuration port
3250_0000	325F_FFFF	GPV_5	1MB	"display" configuration port
3240_0000	324F_FFFF	GPV_4	1MB	"enet" configuration port
3230_0000	323F_FFFF	GPV_3	1MB	"per_m" configuration port
3220_0000	322F_FFFF	GPV_2	1MB	"per_s" configuration port
3210_0000	321F_FFFF	GPV_1	1MB	"wakeup" configuration port
3200_0000	320F_FFFF	GPV_0	1MB	"main" configuration port
3140_0000	31FF_FFFF	Reserved	12MB	Reserved
3100_0000	313F_FFFF	ARM Peripherals	4MB	Reserved
30C0_0000	30FF_FFFF	Periph (AIPS)	4MB	Reserved
3080_0000	30BF_FFFF		4MB	AIPS-3, See IP listing on separate map.
3040_0000	307F_FFFF		4MB	AIPS-2, See IP listing on separate map.
3000_0000	303F_FFFF		4MB	AIPS-1, See IP listing on separate map.
2900_0000	2FFF_FFFF	Reserved	112MB	Reserved
2800_0000	28FF_FFFF	A53 / DAP	16MB	A53 / DAP
2000_0000	27FF_FFFF	PCIe-2	128MB	PCIe-2
1800_0000	1FFF_FFFF	PCIe-1	128MB	PCIe-1
0800_0000	17FF_FFFF	QSPI	256MB	QSPI
00B0_0000	07FF_FFFF	Reserved	117MB	Reserved
0092_0000	00AF_FFFF	OCRAM	1.9MB	Reserved
0090_0000	0091_FFFF		128KB	OCRAM 128KB
0084_0000	008F_FFFF	Reserved	768KB	Reserved
0082_0000	0083_FFFF	TCM	128KB	Reserved
0080_0000	0081_FFFF		128KB	TCMU
007E_0000	007F_FFFF		128KB	TCML
007C_0000	007D_FFFF		128KB	Reserved
0019_0000	007B_FFFF	Reserved	1221MB	Reserved
0018_8000	0018_FFFF	OCRAM_S	32KB	Reserved
0018_0000	0018_7FFF		32KB	OCRAM_S
0011_0000	0017_FFFF	Reserved	448KB	Reserved
0010_8000	0010_FFFF	CAAM	32KB	Reserved
0010_0000	0010_7FFF		32KB	CAAM (32K secure RAM)

Table continues on the next page...

Start Address	End Address	Region	Size	Description
0002_0000	000F_FFFF	Reserved	896KB	Reserved
0001_8000	0001_FFFF	Boot ROM	32KB	Reserved
0001_E800	0001_FFFF		4KB	Boot ROM - Protected 4KB area
0000_0000	0001_E7FF		128KB	Boot ROM

2.1.3 Cortex-M4 Memory Map

Start Address	End Address	Region	Size	Description
E010_0000	FFFF_FFFF	511MB	Reserved	
E000_0000	E00F_FFFF	CM4 PPB	1MB	CM4 PPB
D800_0000	DFFF_FFFF	PCIe-2	128MB	PCIe-2
D000_0000	D7FF_FFFF	PCIe-1	128MB	PCIe-1
C000_0000	CFFF_FFFF	FLASH	256MB	QSPI
4000_0000	BFFF_FFFF	DDR Address	2048MB	DDR Memory
3E00_0000	3FFF_FFFF	Reserved	32MB	Reserved
3DC0_0000	3DFF_FFFF	DDRC	4MB	Reserved
3D80_0000	3DBF_FFFF		4MB	DDR PERF_MON
3D40_0000	3D7F_FFFF		4MB	DDR CTL
3C00_0000	3D3F_FFFF		20MB	DDR PHY
3890_0000	3BFF_FFFF	Reserved	55MB	Reserved
3880_0000	388F_FFFF	GIC	1MB	GIC400
3870_0000	387F_FFFF	Reserved	1MB	Reserved
3830_0000	384F_FFFF	VPU	2MB	VPU_Decoder
3820_0000	382F_FFFF	USB	1MB	USB2 REG
3810_0000	381F_FFFF		1MB	USB1 REG
3801_0000	380F_FFFF	Reserved	960KB	Reserved
3800_0000	3800_FFFF	GPU REG	64KB	GPU REG
3600_0000	37FF_FFFF	QSPI RX Buffers	32MB	Reserved
3400_0000	35FF_FFFF		32MB	QSPI1 RX Buffer
3382_0000	33FF_FFFF	Reserved	7.9MB	Reserved
3381_4000	3381_FFFF	PCIe REG	48KB	Reserved
3381_0000	3381_3FFF		4MB	PCIe2 REG
3380_4000	3380_FFFF		48KB	Reserved
3380_0000	3380_3FFF		4MB	PCIe1 REG
3310_0000	337F_FFFF	MMAP Peripherals	7MB	Reserved
3301_0000	330F_FFFF		960KB	Reserved
3300_8000	3300_FFFF		32KB	Reserved (APBH DMA)
3300_0000	3300_7FFF		32KB	APBH DMA

Table continues on the next page...

Start Address	End Address	Region	Size	Description
32C0_0000	32FF_FFFF	Periph (AIPS)	4MB	AIPS4
3280_0000	32BF_FFFF	Reserved	4MB	Reserved
3270_0000	327F_FFFF	GPV_7	1MB	NoC configuration
3260_0000	326F_FFFF	GPV_6	1MB	"m4" configuration port
3250_0000	325F_FFFF	GPV_5	1MB	"display" configuration port
3240_0000	324F_FFFF	GPV_4	1MB	"enet" configuration port
3230_0000	323F_FFFF	GPV_3	1MB	"per_m" configuration port
3220_0000	322F_FFFF	GPV_2	1MB	"per_s" configuration port
3210_0000	321F_FFFF	GPV_1	1MB	"wakeup" configuration port
3200_0000	320F_FFFF	GPV_0	1MB	"main" configuration port
3100_0000	31FF_FFFF	Reserved	16MB	Reserved
30C0_0000	30FF_FFFF	Pheriph (AIPS)	4MB	Reserved
3080_0000	30BF_FFFF		4MB	AIPS-3, See IP listing on separate map.
3040_0000	307F_FFFF		4MB	AIPS-2, See IP listing on separate map.
3000_0000	303F_FFFF		4MB	AIPS-1, See IP listing on separate map.
2900_0000	2FFF_FFFF	Reserved	112MB	Reserved
2800_0000	28FF_FFFF	A53/ DAP	16MB	A53/ DAP
2040_0000	27FF_FFFF	Reserved	124MB	Reserved
2022_0000	203F_FFFF	OCRAM	1.9MB	Reserved
2020_0000	2021_FFFF		128KB	OCRAM_128KB
2019_0000	201F_FFFF	Reserved	448KB	Reserved
2018_8000	2018_FFFF	OCRAM	32KB	Reserved
2018_0000	2018_7FFF		32KB	OCRAM_S
2011_0000	2017_FFFF	Reserved	448KB	Reserved
2010_8000	2010_FFFF	CAAM	32KB	Reserved
2010_0000	2010_7FFF		32KB	CAAM (32K secure RAM)
2004_0000	200F_FFFF	Reserved	768KB	Reserved
2003_8000	2003_FFFF	ROMCP	32KB	Reserved
2002_0000	2003_7FFF		128KB	Boot ROM (ROMCP)
2000_0000	2001_FFFF	TCM	128KB	TCMU
1FFE_0000	1FFF_FFFF		128KB	TCML
1000_0000	1FFD_FFFF	CM4 ALIAS CODE	262016KB	DDR Code alias
0800_0000	0FFF_FFFF		128MB	QSPI Code alias
0400_0000	07FF_FFFF		64MB	Reserved
00B0_0000	03FF_FFFF	Reserved	53MB	Reserved
0092_0000	00AF_FFFF	OCRAM	1896KB	Reserved
0090_0000	0091_FFFF		128KB	OCRAM_128KB
0019_0000	008F_FFFF	Reserved	7.5MB	Reserved
0018_8000	0018_FFFF	OCRAM_S	32KB	Reserved
0018_0000	0018_7FFF		32KB	OCRAM_S
0011_0000	0017_FFFF	Reserved	448KB	Reserved

Table continues on the next page...

Start Address	End Address	Region	Size	Description
0010_8000	0010_FFFF	CAAM	32KB	Reserved
0010_0000	0010_7FFF		32KB	CAAM (32K secure RAM)
0002_0000	000F_FFFF	Reserved	896KB	Reserved
0000_0000	0001_FFFF	TCML	128KB	TCML alias

2.1.4 DMA memory maps

The Smart DMA memory maps are defined in the following tables.

Table 2-1. SDMA1 Peripheral Memory Map

Address	Size	Peripheral
0xF000	4KB	SPBA
0xE000	4KB	Reserved for SDMA internal registers
0xD000	4KB	Reserved
0xC000	4KB	SAI3
0xB000	4KB	SAI2
0xA000	4KB	SPDIF2
0x9000	4KB	UART2
0x8000	4KB	UART3
0x7000	4KB	Reserved for SDMA internal registers
0x6000	4KB	UART1
0x5000	4KB	Reserved
0x4000	4KB	eCSPI3
0x3000	4KB	eCSPI2
0x2000	4KB	eCSPI1
0x1000	4KB	SPDIF1
0x0000	4KB	Reserved for SDMA internal memory

Table 2-2. SDMA2 Peripheral Memory Map

Address	Size	Peripheral
0xF000	4KB	SPBA
0xE000	4KB	Reserved for SDMA internal registers
0xD000	4KB	Reserved
0xC000	4KB	Reserved
0xB000	4KB	Reserved
0xA000	4KB	Reserved
0x9000	4KB	Reserved

Table continues on the next page...

Table 2-2. SDMA2 Peripheral Memory Map (continued)

Address	Size	Peripheral
0x8000	4KB	Reserved
0x7000	4KB	Reserved for SDMA internal registers
0x6000	4KB	Reserved
0x5000	4KB	SAI4
0x4000	4KB	SAI5
0x3000	4KB	SAI6
0x2000	4KB	Reserved
0x1000	4KB	SAI1
0x0000	4KB	Reserved for SDMA internal memory

2.1.5 AIPS Memory Maps

Table 2-3. AIPS1 Memory Map

Start Address	End Address	Region	NIC Port	Size
303F_0000	303F_FFFF	AIPS-1 (s_b_0)	Reserved	64KB
303E_0000	303E_FFFF		CSU	64KB
303D_0000	303D_FFFF		RDC	64KB
303C_0000	303C_FFFF		SEMAPHORE2	64KB
303B_0000	303B_FFFF		SEMAPHORE1	64KB
303A_0000	303A_FFFF		GPC	64KB
3039_0000	3039_FFFF		SRC	64KB
3038_0000	3038_FFFF		CCM	64KB
3037_0000	3037_FFFF		SNVS_HP	64KB
3036_0000	3036_FFFF		ANA_PLL	64KB
3035_0000	3035_FFFF		OCOTP_CTRL	64KB
3034_0000	3034_FFFF		IOMUXC_GPR	64KB
3033_0000	3033_FFFF		IOMUXC	64KB
3032_0000	3032_FFFF		LCDIF	64KB
3031_0000	3031_FFFF		ROMCP	64KB
3030_0000	3030_FFFF		Reserved	64KB
302F_0000	302F_FFFF		GPT3	64KB
302E_0000	302E_FFFF		GPT2	64KB
302D_0000	302D_FFFF		GPT1	64KB
302C_0000	302C_FFFF		SDMA2	64KB
302B_0000	302B_FFFF		Reserved	64KB
302A_0000	302A_FFFF		WDOG3	64KB

Table continues on the next page...

Table 2-3. AIPS1 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
3029_0000	3029_FFFF		WDOG2	64KB
3028_0000	3028_FFFF		WDOG1	64KB
3027_0000	3027_FFFF		ANA_OSC	64KB
3026_0000	3026_FFFF		ANA_TSENSOR	64KB
3025_0000	3025_FFFF		Reserved	64KB
3024_0000	3024_FFFF		GPIO5	64KB
3023_0000	3023_FFFF		GPIO4	64KB
3022_0000	3022_FFFF		GPIO3	64KB
3021_0000	3021_FFFF		GPIO2	64KB
3020_0000	3020_FFFF		GPIO1	64KB
301F_0000	301F_FFFF		AIPS1_Configuration	64KB
3014_0000	301E_FFFF	AIPS-1 Glob. Module Enable	Reserved	704KB
3010_0000	3013_FFFF		Reserved	256KB
300F_0000	300F_FFFF	AIPS-1 (s_b_1, via SPBA) Glob. Module Enable	SPBA2	64KB
300E_0000	300E_FFFF		Reserved for SDMA2 internal memory	64KB
300D_0000	300D_FFFF		Reserved	64KB
300C_0000	300C_FFFF		Reserved	64KB
300B_0000	300B_FFFF		Reserved	64KB
300A_0000	300A_FFFF		Reserved	64KB
3009_0000	3009_FFFF		Reserved	64KB
3008_0000	3008_FFFF		Reserved	64KB
3007_0000	3007_FFFF		Reserved for SDMA2 internal memory	64KB
3006_0000	3006_FFFF		Reserved	64KB
3005_0000	3005_FFFF		SAI4	64KB
3004_0000	3004_FFFF		SAI5	64KB
3003_0000	3003_FFFF		SAI6	64KB
3002_0000	3002_FFFF		Reserved	64KB
3001_0000	3001_FFFF		SAI1	64KB
3000_0000	3000_FFFF		Reserved for SDMA2 internal memory	64KB

Table 2-4. AIPS2 Memory Map

Start Address	End Address	Region	NIC Port	Size
307F_0000	307F_FFFF	AIPS-2 (s_b_1)	QoSC	64KB
307E_0000	307E_FFFF		Reserved	64KB
307D_0000	307D_FFFF		PERFMON2	64KB
307C_0000	307C_FFFF		PERFMON1	64KB

Table continues on the next page...

Table 2-4. AIPS2 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
307B_0000	307B_FFFF		Reserved	64KB
307A_0000	307A_FFFF		Reserved	64KB
3079_0000	3079_FFFF		Reserved	64KB
3078_0000	3078_FFFF		Reserved	64KB
3077_0000	3077_FFFF		Reserved	64KB
3076_0000	3076_FFFF		Reserved	64KB
3075_0000	3075_FFFF		Reserved	64KB
3074_0000	3074_FFFF		Reserved	64KB
3073_0000	3073_FFFF		Reserved	64KB
3072_0000	3072_FFFF		Reserved	64KB
3071_0000	3071_FFFF		Reserved	64KB
3070_0000	3070_FFFF		GPT4	64KB
306F_0000	306F_FFFF		GPT5	64KB
306E_0000	306E_FFFF		GPT6	64KB
306D_0000	306D_FFFF		Reserved	64KB
306C_0000	306C_FFFF		System_Counter_CTRL	64KB
306B_0000	306B_FFFF		System_Counter_CMP	64KB
306A_0000	306A_FFFF		System_Counter_RD	64KB
3069_0000	3069_FFFF		PWM4	64KB
3068_0000	3068_FFFF		PWM3	64KB
3067_0000	3067_FFFF		PWM2	64KB
3066_0000	3066_FFFF		PWM1	64KB
3065_0000	3065_FFFF		Reserved	64KB
3064_0000	3064_FFFF		Reserved	64KB
3063_0000	3063_FFFF		Reserved	64KB
3062_0000	3062_FFFF		Reserved	64KB
3061_0000	3061_FFFF		Reserved	64KB
3060_0000	3060_FFFF		Reserved	64KB
305F_0000	305F_FFFF		AIPS2_configuration	64KB
3050_0000	305E_FFFF	AIPS-2 Glob. Module Enable	Reserved	960KB
3040_0000	304F_FFFF		Reserved	1024KB

Table 2-5. AIPS3 Memory Map

Start Address	End Address	Region	NIC Port	Size
30BF_0000	30BF_FFFF	AIPS-3 (s_b_2)	Reserved	64KB
30BE_0000	30BE_FFFF		ENET1	64KB
30BD_0000	30BD_FFFF		SDMA1	64KB
30BC_0000	30BC_FFFF		Reserved	64KB

Table continues on the next page...

Table 2-5. AIPS3 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
30BB_0000	30BB_FFFF		QSPI	64KB
30BA_0000	30BA_FFFF		Reserved	64KB
30B9_0000	30B9_FFFF		Reserved	64KB
30B8_0000	30B8_FFFF		CSI2	64KB
30B7_0000	30B7_FFFF		MIPI_CSI_PHY2	64KB
30B6_0000	30B6_FFFF		MIPI_CSI2	64KB
30B5_0000	30B5_FFFF		uSDHC2	64KB
30B4_0000	30B4_FFFF		uSDHC1	64KB
30B3_0000	30B3_FFFF		Reserved	64KB
30B2_0000	30B2_FFFF		Reserved	64KB
30B1_0000	30B1_FFFF		Reserved	64KB
30B0_0000	30B0_FFFF		Reserved	64KB
30AF_0000	30AF_FFFF		Reserved	64KB
30AE_0000	30AE_FFFF		Reserved	64KB
30AD_0000	30AD_FFFF		Reserved	64KB
30AC_0000	30AC_FFFF		SEMAPHORE_HS	64KB
30AB_0000	30AB_FFFF		MU_B	64KB
30AA_0000	30AA_FFFF		MU_A	64KB
30A9_0000	30A9_FFFF		CSI1	64KB
30A8_0000	30A8_FFFF		MIPI_CSI_PHY1	64KB
30A7_0000	30A7_FFFF		MIPI_CSI1	64KB
30A6_0000	30A6_FFFF		UART4	64KB
30A5_0000	30A5_FFFF		I2C4	64KB
30A4_0000	30A4_FFFF		I2C3	64KB
30A3_0000	30A3_FFFF		I2C2	64KB
30A2_0000	30A2_FFFF		I2C1	64KB
30A1_0000	30A1_FFFF		MIPI_DSI	64KB
30A0_0000	30A0_FFFF		MIPI_PHY	64KB
309F_0000	309F_FFFF		AIPS3_Configuration	64KB
3094_0000	309E_FFFF	AIPS-3 Glob. Module Enable	Reserved	704KB
3090_0000	3093_FFFF		CAAM	256KB
308F_0000	308F_FFFF	AIPS-3 (s_b_2, via SPBA) Glob. Module Enable	SPBA1	64KB
308E_0000	308E_FFFF		Reserved for SDMA internal registers	64KB
308D_0000	308D_FFFF		Reserved	64KB
308C_0000	308C_FFFF		SAI3	64KB
308B_0000	308B_FFFF		SAI2	64KB
308A_0000	308A_FFFF		SPDIF2	64KB
3089_0000	3089_FFFF		UART2	64KB
3088_0000	3088_FFFF		UART3	64KB

Table continues on the next page...

Table 2-5. AIPS3 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
3087_0000	3087_FFFF		Reserved for SDMA internal registers	64KB
3086_0000	3086_FFFF		UART1	64KB
3085_0000	3085_FFFF		Reserved	64KB
3084_0000	3084_FFFF		eCSPI3	64KB
3083_0000	3083_FFFF		eCSPI2	64KB
3082_0000	3082_FFFF		eCSPI1	64KB
3081_0000	3081_FFFF		SPDIF1	64KB
3080_0000	3080_FFFF		Reserved for SDMA internal memory	64KB

Table 2-6. AIPS4 Memory Map

Start Address	End Address	Region	NIC Port	Size
32FF_0000	32FF_FFFF	AIPS-4 (s_h_10)	Reserved	64KB
32FE_0000	32FE_FFFF		PLATFORM_CTRL	64KB
32FD_0000	32FD_FFFF		Reserved	64KB
32FC_0000	32FC_FFFF		Reserved	64KB
32FB_0000	32FB_FFFF		MTR	64KB
32FA_0000	32FA_FFFF		Reserved	64KB
32F9_0000	32F9_FFFF		Reserved	64KB
32F8_0000	32F8_FFFF		TZASC	64KB
32F7_0000	32F7_FFFF		Reserved	64KB
32F6_0000	32F6_FFFF		Reserved	64KB
32F5_0000	32F5_FFFF		Reserved	64KB
32F4_0000	32F4_FFFF		Reserved	64KB
32F3_0000	32F3_FFFF		Reserved	64KB
32F2_0000	32F2_FFFF		Reserved	64KB
32F1_0000	32F1_FFFF		Reserved	64KB
32F0_0000	32F0_FFFF		Reserved	64KB
32EF_0000	32EF_FFFF		Reserved	64KB
32EE_0000	32EE_FFFF		Reserved	64KB
32ED_0000	32ED_FFFF		Reserved	64KB
32EC_0000	32EC_FFFF		Reserved	64KB
32EB_0000	32EB_FFFF		Reserved	64KB
32EA_0000	32EA_FFFF		Reserved	64KB
32E9_0000	32E9_FFFF		Reserved	64KB
32E8_0000	32E8_FFFF		Reserved	64KB
32E7_0000	32E7_FFFF		Reserved	64KB
32E6_0000	32E6_FFFF		Reserved	64KB

Table continues on the next page...

Table 2-6. AIPS4 Memory Map (continued)

Start Address	End Address	Region	NIC Port	Size
32E5_0000	32E5_FFFF		Reserved	64KB
32E4_0000	32E4_FFFF		HDMI_SEC	64KB
32E3_0000	32E3_FFFF		DC_MST3	64KB
32E2_0000	32E2_FFFF		DC_MST2	64KB
32E1_0000	32E1_FFFF		DC_MST1	64KB
32E0_0000	32E0_FFFF		DC_MST0	64KB
32DF_0000	32DF_FFFF		AIPS4_configuration	64KB
32D0_0000	32DE_FFFF	AIPS-4 Glob. Module Enable	Reserved	960KB
32C0_0000	32CF_FFFF		HDMI_CTRL	1024KB

2.1.6 DAP Memory Map

Table 2-7. DAP Memory Map Table

Start Address	End Address	Size	Allocation
28C0_A000	28FF_FFFF	4056KB	Reserved
28C0_9000	28C0_9FFF	4KB	HUGO_CXCTI1
28C0_8000	28C0_8FFF	4KB	HUGO_CXCTI0
28C0_7000	28C0_7FFF	4KB	CXTPIU
28C0_6000	28C0_6FFF	4KB	CXTMC_ETR
28C0_5000	28C0_5FFF	4KB	ATB_REPLICATOR
28C0_4000	28C0_4FFF	4KB	CXTMC_ETB
28C0_3000	28C0_3FFF	4KB	HUGO_ATB_FUNNEL
28C0_2000	28C0_2FFF	4KB	CXTSGEN_READ
28C0_1000	28C0_1FFF	4KB	CXTSGEN_CTRL
28C0_0000	28C0_0FFF	4KB	HUGO ROM Table
2875_0000	28BF_FFFF	4800KB	Reserved
2874_0000	2874_FFFF	64KB	MP4-CPU3 Trace
2873_0000	2873_FFFF	64KB	MP4-CPU3 PMU
2872_0000	2872_FFFF	64KB	MP4-CPU3 CTI
2871_0000	2871_FFFF	64KB	MP4-CPU3 Debug
2865_0000	2870_FFFF	768KB	Reserved
2864_0000	2864_FFFF	64KB	MP4-CPU2 Trace
2863_0000	2863_FFFF	64KB	MP4-CPU2 PMU
2862_0000	2862_FFFF	64KB	MP4-CPU2 CTI
2861_0000	2861_FFFF	64KB	MP4-CPU2 Debug
2855_0000	2860_FFFF	768KB	Reserved

Table continues on the next page...

Table 2-7. DAP Memory Map Table (continued)

Start Address	End Address	Size	Allocation
2854_0000	2854_FFFF	64KB	MP4-CPU1 Trace
2853_0000	2853_FFFF	64KB	MP4-CPU1 PMU
2852_0000	2852_FFFF	64KB	MP4-CPU1 CTI
2851_0000	2851_FFFF	64KB	MP4-CPU1 Debug
2845_0000	2850_FFFF	768KB	Reserved
2844_0000	2844_FFFF	64KB	MP4-CPU0 Trace
2843_0000	2843_FFFF	64KB	MP4-CPU0 PMU
2842_0000	2842_FFFF	64KB	MP4-CPU0 CTI
2841_0000	2841_FFFF	64KB	MP4-CPU0 Debug
2840_0000	2840_FFFF	64KB	MP4 ROM Table
2801_0000	283F_FFFF	4032KB	Reserved
2800_0000	2800_FFFF	64KB	DAP ROM Table

Chapter 3
Security

3.1 System Security

3.1.1 Overview

The security system modules are described in the sections below.

3.1.2 Central Security Unit (CSU)

The chip uses the CSU to manage security for all masters/slaves that don’t directly support security like the CPU core.

3.1.3 Cryptographic Acceleration and Assurance Module (CAAM)

CAAM is the chip's cryptographic acceleration and assurance module, which serves as NXP's latest cryptographic acceleration and offloading hardware. It combines functions previously implemented in separate modules to create a modular and scalable acceleration and assurance engine. It also implements block encryption algorithms, stream cipher algorithms, hashing algorithms, public key algorithms, run-time integrity checking, a secure memory controller, and a hardware random number generator.

CAAM supports the following key features:

- PKHA block to support Public Key Cryptography with RSA 4096 and ECC algorithms
- 3 Job Rings
- RTIC (real time integrity checking)
- AES, DES, 3DES with Side channel attack resistance
- Widevine cipher text stealing (AES-CBC-CTS mode)
- PlayReady content protection

In order to provide better video content protection, CAAM also supports the domain based resource protection.

3.1.4 Secure Non-Volatile Storage (SNVS)

Secure Non-Volatile Storage (SNVS) is a companion logic block to the Cryptographic Acceleration and Assurance Module (CAAM). This block is the chip's central reporting point for security-relevant events, such as the success or failure of boot software validation and the detection of potential security compromises. This security event information allows the SNVS to determine whether the chip is in a trustworthy state.

3.1.5 On-Chip OTP Controller (OCOTP_CTRL)

The OCOPT_CTRL is used to control the 8K bit OTP fuse in the chip. It supports:

- Load fuse into shadow registers after power-on reset;
- Register interface to allow SW to read, override or program the fuse
- Flexible permission control to the fuse, including read-protect, override-protect, program-protect, lock and etc
- Programming sequence to allow single bit to be programmed individually, and also allow the non-programmed bit to be programmed later

For the 8K bits fuse, following bits are reserved for various IP requirements:

- Fuse control to disable each of the four Cortex A53 cores.
- Fuse control to disable individual IP modules such as MIPI, HDMI, ENET, USB, PCIe and etc.
- Fuse control to disable the Dolby Vision logic inside the display controller block.
- Fuse for ROM patch, with a dedicated fuse lock bit for security purpose.
- Fuse for HDMI firmware SRK and HDCP key

3.1.6 TrustZone

TrustZone security architecture is supported in the chip. For on-chip RAM, both OCRAM/OCRAM_S have TrustZone access control support through its TZ wrapper logic. One region with configurable address range of the OCRAM/OCRAM_S can be set to TZ access only. DRAM has a dedicated TZASC block that can support up to 16 configurable memory regions.

3.1.7 Resource Domain Controller (RDC)

The chip uses domain based resource control architecture for the memory/peripheral resource sharing and isolation between the Quad-core Cortex A53 platform, Cortex M4 core, and other bus masters. RDC supports flexible configuration on IP's access permission, each individual IP can, for example, be configured as A-core only or M-core only. RDC also supports multiple regions with flexible access permission control for each memory space. The memory region access control is defined in the table below. The security and exclusive support for DRAM will be done by the TZASC and the DRAM controller itself.

NOTE

The RDC and TrustZone features are completely independent of each other but will be used together.

Table 3-1. Memory Region Access Control

Memory Space	Regions Supported	Granularity	Security Support	Exclusive Support
DRAM	8 regions	4K Bytes	Disable	Disable
QSPI	8 regions	4K Bytes	Enable	Enable
PCIe	8 regions	4K Bytes	Enable	Enable
OCRAM	5 regions	128 Bytes	Disable	Enable
OCRAM_S	5 regions	128 Bytes	Disable	Enable
TCM	5 regions	128 Bytes	Enable	Enable
GIC	4 regions	4K Bytes	Enable	Disable
USBMIX	4 regions	4K Bytes	Enable	Disable
GPUMIX	4 regions	4K Bytes	Enable	Disable
VPUMIX(Decoder)	4 regions	4K Bytes	Enable	Disable
VPUMIX(Peripheral)	4 regions	4K Bytes	Enable	Disable
DRAM Register	4 regions	4K Bytes	Enable	Disable
DEBUG	4 regions	4K Bytes	Enable	Disable

3.2 Resource Domain Controller (RDC)

3.2.1 Overview

The Resource Domain Controller (RDC) provides robust support for the isolation of destination memory mapped locations such as peripherals and memory to a single core, a bus master, or set of cores and bus masters.

Many of today's processors have multiple cores for increased performance and flexibility. In some cases, the cores serve different functions (e.g. user level applications versus real time machine control) and in such cases the software for each core may be developed by different providers.

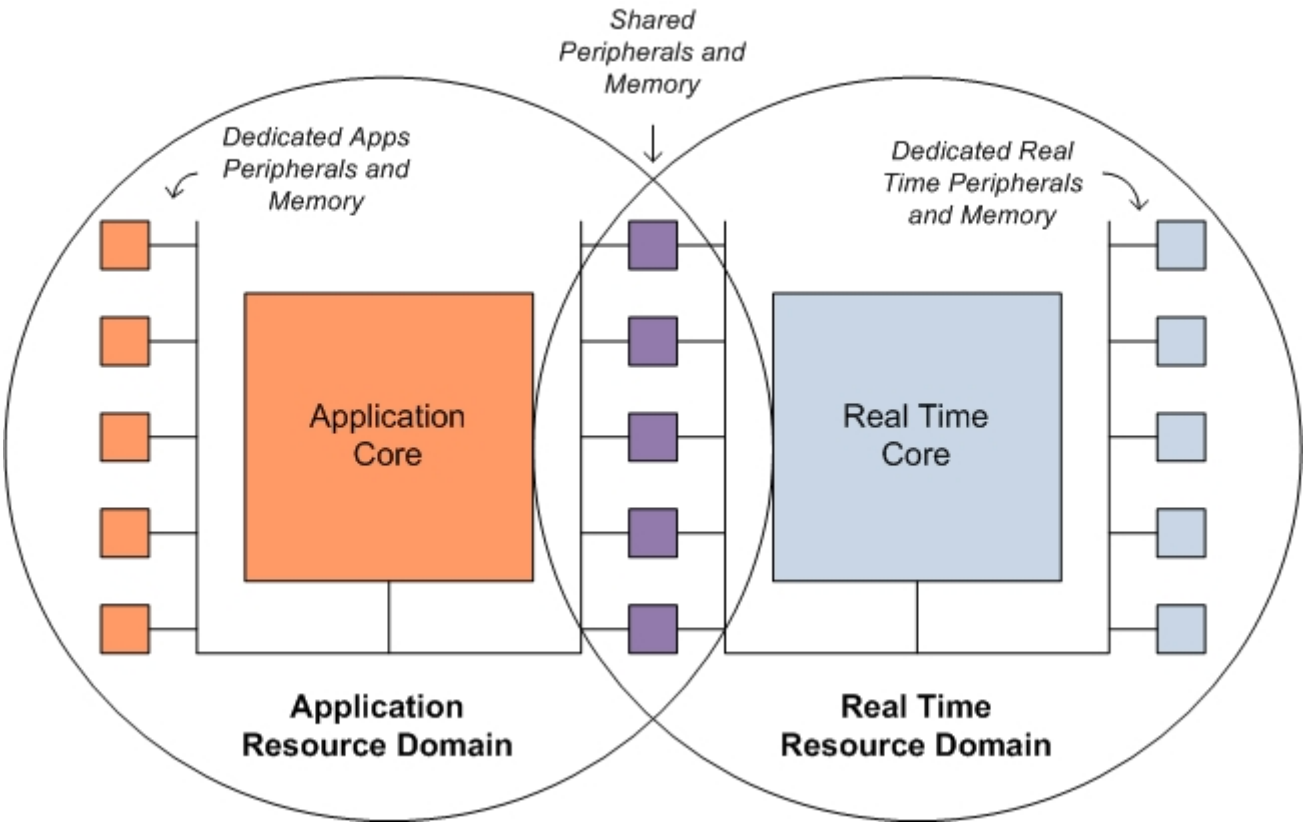


Figure 3-1. Dedicated and Shared Peripherals

For efficiency reasons the code on the cores may share chip resources such as peripherals and memory. The sharing of chip resources between the somewhat independent processing domains allows for the opportunity of data collisions where information stored in peripherals or memory by a process on one core is overwritten by software running on another core. Without careful collaboration between the two operating systems inadvertent malfunction or degradation in performance may result.

The RDC provides a mechanism to allow boot time configuration code to establish resource domains by assigning cores, bus masters, peripherals and memory regions to domain identifiers. Once configured, bus transactions are monitored to restrict accesses initiated by cores and bus masters to their respective peripherals and memory.

For shared peripherals, the RDC provides a semaphore-based locking mechanism to provide for temporary exclusivity while the domain software uses the peripheral. Once the software of one domain has finished the task and finished with the peripheral then it may release the semaphore making the peripheral available to the other domain.

3.2.1.1 Features

Resource domain subsystem has the following features:

- Assignment of cores, bus masters, peripherals, and memory regions to a resource domain
- Fixed memory resolution of 128 Bytes for small address spaces and 4 KB for large address spaces
- Four resource domain identifiers
- Memory read/write access controls for each resource domain and region
- Optional semaphore-based, hardware-enforced exclusive access of shared peripherals to a resource domain
- Prioritized access permissions for overlapping memory regions
- Automatic restoration of resource domain access permissions to memory regions in the power-down domain

3.2.2 Functional Description

The RDC is the central location for creation of isolated resource domains and for the enablement of semaphore-based access also known as “safe sharing”. Configuration software assigns one of four resource domain identifiers to each core and bus master, and allocates each memory region and peripheral to one or more resource domains.

Memory Read or Write access privileges for each resource domain are declared for each memory region. In addition, the software configuration determines which shared peripherals (those peripherals allocated to more than one domain) require safe sharing by setting the semaphore-required configuration for each peripheral.

The RDC configuration information is sent to the fabric ports, memories gaskets, semaphore controller, and peripherals to control access based on domain assignments. The fabric uses the domain identifier associated with each port to include this information along with the bus transaction. When the slave gasket encounters a bus transaction it makes a comparison of the transaction domain ID to the RDC-provided list of allowed domains. If the transaction domain ID is on the list then access may be permitted.

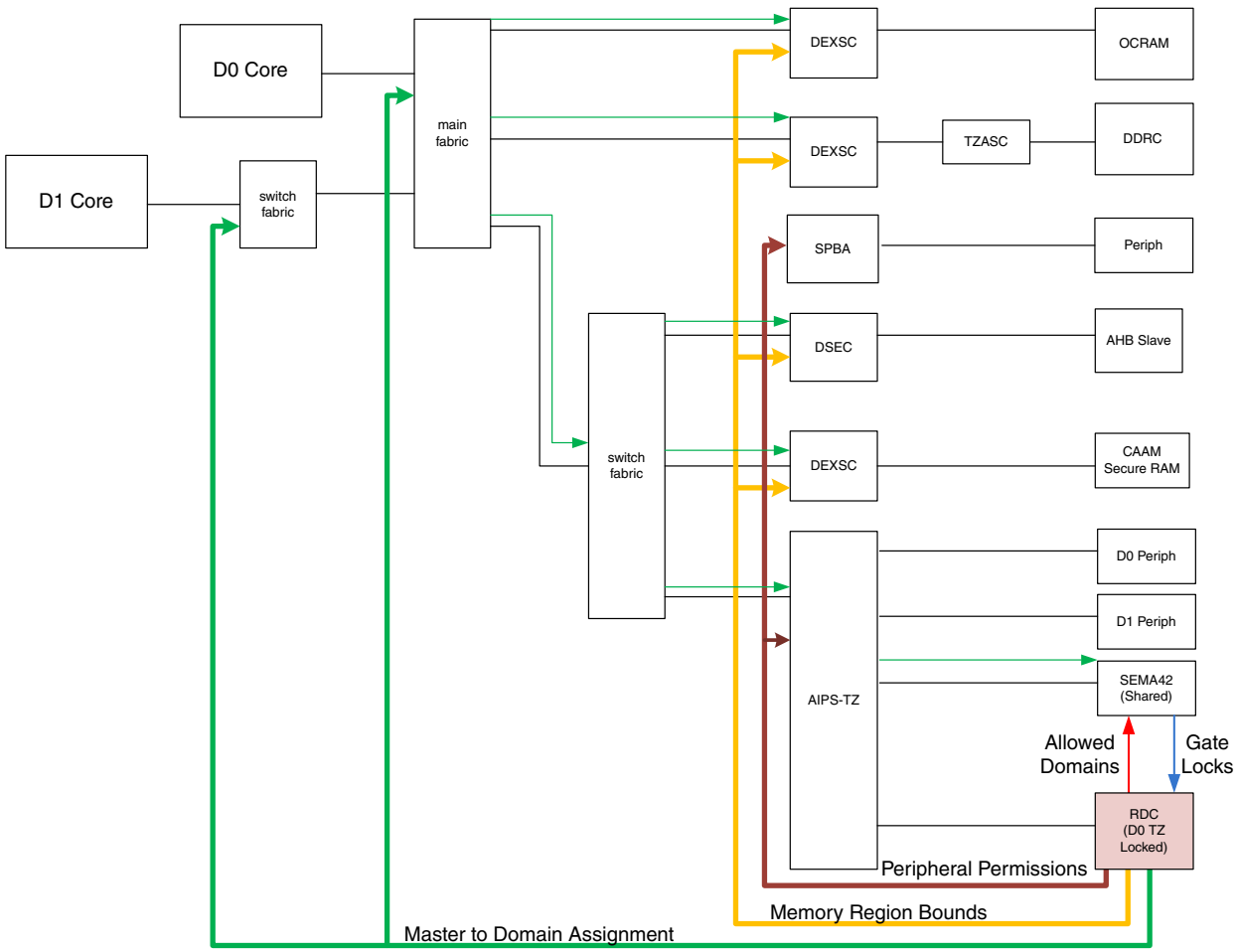


Figure 3-2. Example RDC Connections

For shared peripherals, RDC permits more than one domain access to a single peripheral. RDC also provides three ways to control synchronized use of shared peripherals. These methods include hardware-enforced synchronization, software-based semaphores, or no synchronization. The latter may be suitable for well-tuned multi-core operating systems that handle synchronization in the core platform, for instance.

For hardware-enforced synchronization, also known as "safe sharing", ownership of the peripheral must be claimed in the semaphore controller before access is allowed to the shared peripheral. The "semaphore required" bit (SREQ) is set in the PDAP register corresponding to the shared peripheral which causes the RDC to require that a semaphore is obtained by a domain before access by that domain to the shared peripheral is allowed. During the time that the domain has the semaphore in possession its bus masters have exclusive access to the peripheral.

When the semaphore is released then no domain masters have access until the semaphore is obtained again. When the SREQ is set, RDC does not allow masters to obtain semaphores of peripherals to which it is not allocated; the master must have designated access in the D-registers of the corresponding PDAP register (e.g. D3R bit set for Domain 3 access of the shared peripheral). There is a one-to-one mapping between the semaphore controller gate and the resource domain controller peripheral. The mapping of PDAP registers and peripherals can be found in the Peripheral Map section of the RDC chapter.

3.2.2.1 Domain ID

The RDC provides for an isolation of domain resources by use of a identifier called the Domain ID (DID). A core and its resources including memory, bus masters, and peripherals are all associated with a single DID. When software or a DMA attempts to access a peripheral or memory, the corresponding bus transaction includes the DID along with the other bus control information such as Read, Write, and privilege mode.

3.2.2.2 Resource Assignment

The RDC allows assignment of peripherals and memories to one or more domains while each bus master or core is placed in one of four domains. The masters are assigned a domain in the MDA register. A peripheral is given R/W access permissions to each domain in the PDAP register. Memory regions are bound by address space in start and end registers, the MRSA and MREA. Each memory region is assigned one or more allowed domains and R/W permissions in the MRC control register. Memory regions must be enabled before the permissions are active. Otherwise the permissions are not restricted.

The RDC itself should be isolated to ensure that only a trustworthy resource manager can configure the RDC registers. This process may either be present initially, during secure boot, or during the runtime in the secure world, for example. If the operating system does not support a runtime trusted execution then during the secure boot process the RDC configuration can be locked to prevent further modification after the operating systems are running.

NOTE

The CCM supports multicore awareness based on resource domain assignments programmed into the RDC. Refer to the CCM chapter regarding the relationship between core resource domains and their respective CCM resources. Failing to follow the proper sequence when updating the resource domain

assignments of the core can result in clocks being inadvertently gated.

3.2.2.3 Safe Sharing

For shared peripherals, the RDC can be configured to require a domain to obtain a semaphore lock before access to the peripheral is allowed. This feature helps prevent collisions from processes on separate cores that may want to use the same peripheral at the same time. The RDC sends a list of eligible domains to the semaphore module for each gate/peripheral. The eligible domains are those that are set in the peripheral domain access permissions (PDAP) registers. There is a one-to-one correspondence between semaphore gates and peripherals so each gate in the semaphore block represents a peripheral. The RDC receives semaphore locks from the hardware semaphore module (SEMA42). A semaphore lock is acquired when a core or bus master from a given domain requests a lock for a particular gate. The semaphore module compares the request's domain ID against the list of eligible domain IDs. If the domain ID is on the list and the lock is available then the lock is set and a signal is sent back to the RDC module indicating a lock has been acquired for a particular gate and to which domain ID the lock belongs. The RDC then restricts access to the corresponding peripheral to only transactions originating from the domain that has the lock. Another domain, though on the shared list to access the peripheral, must then wait until the lock is released before acquiring the lock and gaining access to the peripheral. To enable this feature of hardware enforcement for the semaphore locks, the SREQ bit is set in the RDC resource register.

If the SREQ is set, then when a process determines it needs a shared peripheral, it must first lock the resource in the semaphore module. Once the resource is locked, the semaphore module sends a signal to the RDC indicating the domain has access to the resource. The RDC will then set the access permissions to allow that domain access to the peripheral.

For a domain to acquire a lock on a peripheral, the domain must have been assigned to the peripheral in the RDC Peripheral Domain Access Permissions register (PDAP). The semaphore module only allows safe-sharing locks for those domains that are assigned to the peripheral. The semaphore module does not consider the access type (Read or Write) when allowing domains to acquire locks.

The SEMA42 module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Module definition supporting 64 hardware-enforced gates in a multi-processor configuration, where up to 15 processors can be supported; cp*X* is meant to represent core processor *X*
  - Gates appear as an *n*-entry byte-size array with read and write accesses (*n* = 16, 32, 64).
    - Processors lock gates by writing "Master_index" to the appropriate gate and must read back the gate value to verify the lock operation was successful. The Master_index value for the processors can be found in the Master Index Allocation table, which can be found in the AIPSTZ block.. Also note that after locking, the gate register contains the master_id value of the locking processor (in bits [3:0]), and also the value of the locking domain (in bits [5:4]).
    - Once locked, the gate is unlocked by a write of zeroes from the locking processor.
    - The number of implemented gates is specified by a hardware configuration define.
  - Each hardware gate appears as a 16-state, 4-bit state machine.
    - 16-state implementation

      if gate = 0x0, then state = unlocked
      if gate = 0x1, then state = locked by processor (master_index) 0
      if gate = 0x2, then state = locked by processor (master_index) 1
      …
      if gate = 0xF, then state = locked by processor (master_index) 14
    - Uses the logical bus master number (master_index) as a reference attribute plus the specified data patterns to validate all write operations.
    - Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor.
  - Secure reset mechanisms are supported to clear the contents of individual gates, as well as a clear_all capability.
- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses

3.2.2.4 Resource Domain Control and Security Considerations

Conceptually, the RDC configuration is independent of the processor privilege mode and security domain. It is intended to allow for isolation between core processing environments to prevent collisions and increase reliability. Access between resource domains is mutually exclusive and each domain should be in control of its own privilege modes and access rights.

However, it is important to realize multi-core processors may have a multiple resource domains but only one overarching security domain. Chip security controls reside in one resource domain. In this configuration, a domain can affect at least one level of access privileges in the other domain. This may be acceptable but clarity and care is needed to ensure expected functionality.

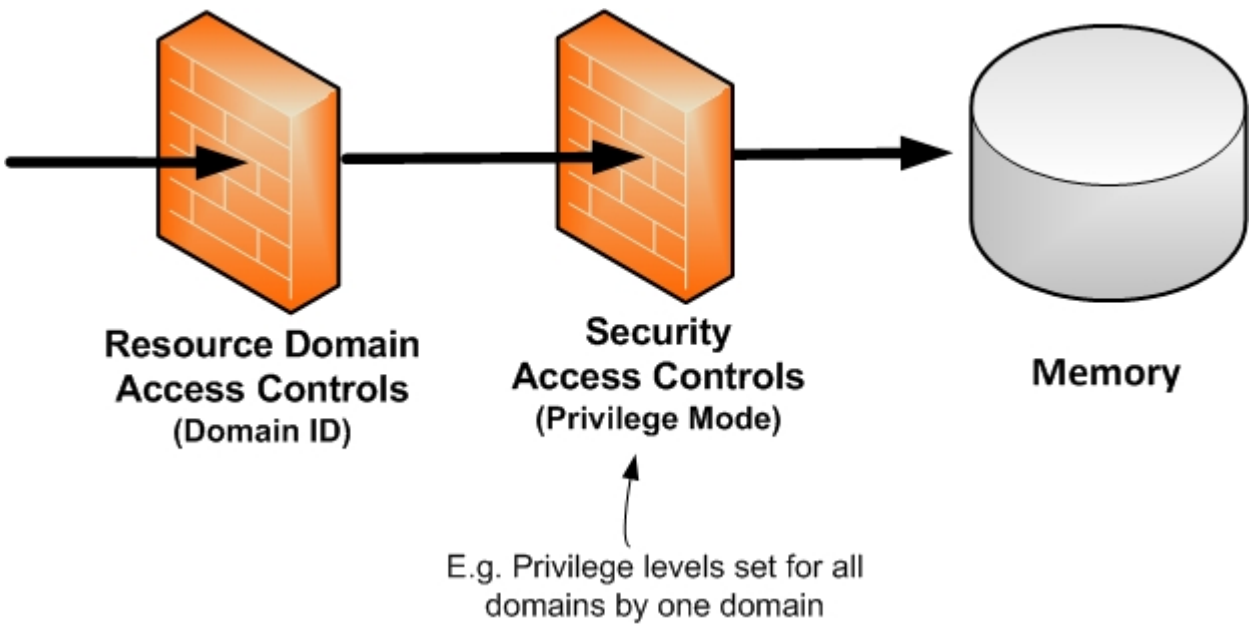


Figure 3-3. Access Control to Memory

Therefore, access to the security controls should be restricted to the most trustworthy operating mode of the core and privilege levels should be coordinated to ensure that shared peripherals and memory regions are accessible by both cores. For instance, if a memory region is designated for secure accesses then all domain masters that share that region must have secure privileges.

3.2.3 Modes of Operation

The RDC provides access controls to the resource domain subsystem. When the device is in a low power mode then some memory regions in the subsystem may be powered off. RDC responds to the impacted memory regions by automatically reconfiguring the memory regions once power returns and blocking access to those memory regions until the reconfiguration process is complete.

3.2.3.1 Low Power Modes

The RDC loads configuration information for memory regions (MRSA, MRSE, MRC) into access control mechanisms (gaskets) at the memory interface. The location of this configuration information may reside inside power domains that lose power during sleep modes for energy savings. To restore configuration information upon return from sleep mode, the RDC receives a global power control signal indicating power is restored. The RDC then automatically reconfigures the memory regions with the configuration information.

During reconfiguration, access is blocked to the previously powered down memories. When the RDC completes reconfiguration it issues an interrupt and allows access to the memory regions. Only the powered down regions are blocked during the reconfiguration. Memory regions in the "always-on" power domain (still powered during sleep mode) remain available according to the programmed access rights. If no memory regions were enabled then the powered down regions are available immediately when power is restored.

The figure below shows the Global Power Control signal which RDC uses to invalidate the configuration upon deassertion and to restore the configuration when re-asserted. The configuration is valid and bus transactions allowed once the memory regions have been restored.

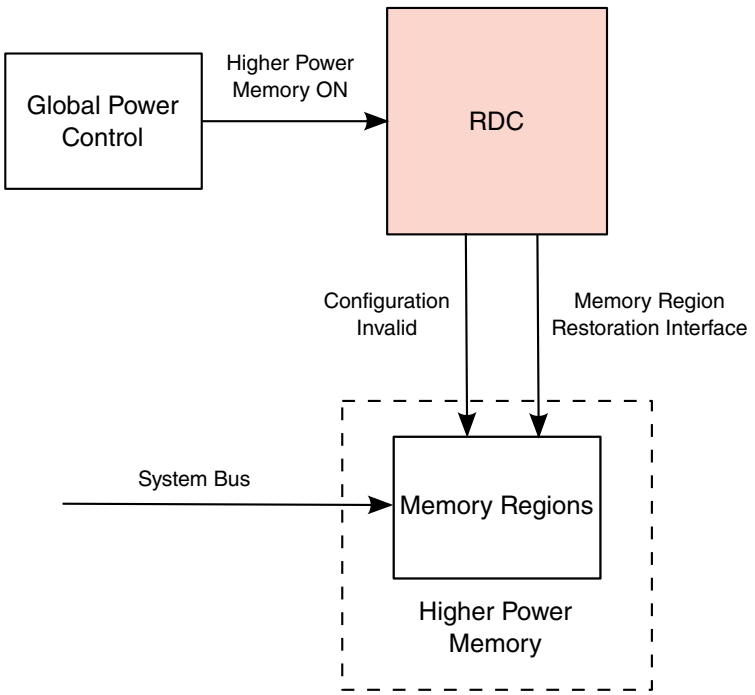


Figure 3-4. Memory Restoration Signaling

3.2.4 Programming Interface

This section provides product specific details describing the mapping of resources - peripherals, bus masters, and memory regions - to corresponding resource domain controls RDC registers.

The RDC and RDC_SEMA42 register maps are combined in this chapter. The base address for the one RDC map and two SEMA42 maps are each separated by 4KB. While there are two SEMA42 submodules and therefore two sets of SEMA42 registers, this chapter describes one. Please refer to the peripheral memory map for the base addresses of the RDC and SEMA42 modules.

3.2.4.1 Master Assignment Registers

Table 3-2. Master Assignment Mapping

Master	RDC MDA Register
Quad A53	RDC_MDA0
M4	RDC_MDA1

Table continues on the next page...

Table 3-2. Master Assignment Mapping (continued)

Master	RDC MDA Register
PCIE_CTRL1	RDC_MDA2
PCIE_CTRL2	RDC_MDA3
VPU Decoders	RDC_MDA4
LCDIF	RDC_MDA5
CSI-1	RDC_MDA6
CSI-2	RDC_MDA7
Coresight	RDC_MDA8
DAP	RDC_MDA9
CAAM	RDC_MDA10
SDMA (p)	RDC_MDA11
SDMA (b)	RDC_MDA12
APBHDMA	RDC_MDA13
RAWNAND	RDC_MDA14
uSDHC1	RDC_MDA15
uSDHC2	RDC_MDA16
Display Controller	RDC_MDA17
GPU	RDC_MDA18
USB1	RDC_MDA19
USB2	RDC_MDA20
TESTPORT	RDC_MDA21
ENET1_TX	RDC_MDA22
ENET1_RX	RDC_MDA23
SDMA2 (p)	RDC_MDA24
SDMA2 (b)	RDC_MDA24
SDMA2 to SPBA2	RDC_MDA24
Reserved	RDC_MDA25
SDMA1 to SPBA1	RDC_MDA26

3.2.4.2 Peripheral Mapping

Each peripheral has a corresponding resource domain assignment register in the RDC and semaphore lock register in the RDC_SEMA42 module. The following table shows allocation of the RDC PDAP and RDC_SEMA4 GATE registers for peripheral resource domain assignment.

NOTE

Access control of the RDC registers can be programmed using the respective PDAP register. The default setting of the PDAP

register for the RDC allows access from all domains. Use caution when restricting access of the RDC registers to avoid conditions where access to the RDC registers is needed but no master is assigned to a domain with access rights to the RDC.

Table 3-3. RDC Peripheral Mapping

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
GPIO1	RDC_PDAP0	SEMA42 B1 / G0
GPIO2	RDC_PDAP1	SEMA42 B1 / G1
GPIO3	RDC_PDAP2	SEMA42 B1 / G2
GPIO4	RDC_PDAP3	SEMA42 B1 / G3
GPIO5	RDC_PDAP4	SEMA42 B1 / G4
Reserved	RDC_PDAP5	SEMA42 B1 / G5
ANA_TSENSOR	RDC_PDAP6	SEMA42 B1 / G6
ANA_OSC	RDC_PDAP7	SEMA42 B1 / G7
WDOG1	RDC_PDAP8	SEMA42 B1 / G8
WDOG2	RDC_PDAP9	SEMA42 B1 / G9
WDOG3	RDC_PDAP10	SEMA42 B1 / G10
Reserved	RDC_PDAP11	SEMA42 B1 / G11
SDMA2	RDC_PDAP12	SEMA42 B1 / G12
GPT1	RDC_PDAP13	SEMA42 B1 / G13
GPT2	RDC_PDAP14	SEMA42 B1 / G14
GPT3	RDC_PDAP15	SEMA42 B1 / G15
Reserved	RDC_PDAP16	SEMA42 B1 / G16
ROMCP	RDC_PDAP17	SEMA42 B1 / G17
LCDIF	RDC_PDAP18	SEMA42 B1 / G18
IOMUXC	RDC_PDAP19	SEMA42 B1 / G19
IOMUXC_GPR	RDC_PDAP20	SEMA42 B1 / G20
OCOTP_CTRL	RDC_PDAP21	SEMA42 B1 / G21
ANA_PLL	RDC_PDAP22	SEMA42 B1 / G22
SNVS_HP	RDC_PDAP23	SEMA42 B1 / G23
CCM	RDC_PDAP24	SEMA42 B1 / G24
SRC	RDC_PDAP25	SEMA42 B1 / G25
GPC	RDC_PDAP26	SEMA42 B1 / G26
SEMAPHORE1	RDC_PDAP27	SEMA42 B1 / G27
SEMAPHORE2	RDC_PDAP28	SEMA42 B1 / G28
RDC	RDC_PDAP29	SEMA42 B1 / G29
CSU	RDC_PDAP30	SEMA42 B1 / G30
Reserved	RDC_PDAP31	SEMA42 B1 / G31
DC_MST0	RDC_PDAP32	SEMA42 B1 / G32
DC_MST1	RDC_PDAP33	SEMA42 B1 / G33

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
DC_MST2	RDC_PDAP34	SEMA42 B1 / G34
DC_MST3	RDC_PDAP35	SEMA42 B1 / G35
HDMI_SEC	RDC_PDAP36	SEMA42 B1 / G36
Reserved	RDC_PDAP37	SEMA42 B1 / G37
PWM1	RDC_PDAP38	SEMA42 B1 / G38
PWM2	RDC_PDAP39	SEMA42 B1 / G39
PWM3	RDC_PDAP40	SEMA42 B1 / G40
PWM4	RDC_PDAP41	SEMA42 B1 / G41
System_Counter_RD	RDC_PDAP42	SEMA42 B1 / G42
System_Counter_CMP	RDC_PDAP43	SEMA42 B1 / G43
System_Counter_CTRL	RDC_PDAP44	SEMA42 B1 / G44
HDMI_CTRL	RDC_PDAP45	SEMA42 B1 / G45
GPT6	RDC_PDAP46	SEMA42 B1 / G46
GPT5	RDC_PDAP47	SEMA42 B1 / G47
GPT4	RDC_PDAP48	SEMA42 B1 / G48
Reserved	RDC_PDAP49	SEMA42 B1 / G49
Reserved	RDC_PDAP50	SEMA42 B1 / G50
Reserved	RDC_PDAP51	SEMA42 B1 / G51
Reserved	RDC_PDAP52	SEMA42 B1 / G52
Reserved	RDC_PDAP53	SEMA42 B1 / G53
Reserved	RDC_PDAP54	SEMA42 B1 / G54
Reserved	RDC_PDAP55	SEMA42 B1 / G55
TZASC	RDC_PDAP56	SEMA42 B1 / G56
Reserved	RDC_PDAP57	SEMA42 B1 / G57
Reserved	RDC_PDAP58	SEMA42 B1 / G58
MTR	RDC_PDAP59	SEMA42 B1 / G59
PERFMON1	RDC_PDAP60	SEMA42 B1 / G60
PERFMON2	RDC_PDAP61	SEMA42 B1 / G61
PLATFORM_CTRL	RDC_PDAP62	SEMA42 B1 / G62
QoSC	RDC_PDAP63	SEMA42 B1 / G63
MIPI_PHY	RDC_PDAP64	SEMA42 B2 / G0
MIPI_DSI	RDC_PDAP65	SEMA42 B2 / G1
I2C1	RDC_PDAP66	SEMA42 B2 / G2
I2C2	RDC_PDAP67	SEMA42 B2 / G3
I2C3	RDC_PDAP68	SEMA42 B2 / G4
I2C4	RDC_PDAP69	SEMA42 B2 / G5
UART4	RDC_PDAP70	SEMA42 B2 / G6
MIPI_CSI1	RDC_PDAP71	SEMA42 B2 / G7
MIPI_CSI_PHY1	RDC_PDAP72	SEMA42 B2 / G8

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
CSI1	RDC_PDAP73	SEMA42 B2 / G9
MU_A	RDC_PDAP74	SEMA42 B2 / G10
MU_B	RDC_PDAP75	SEMA42 B2 / G11
SEMAPHORE_HS	RDC_PDAP76	SEMA42 B2 / G12
Reserved for SDMA2 internal memory	RDC_PDAP77	SEMA42 B2 / G13
SAI1	RDC_PDAP78	SEMA42 B2 / G14
Reserved	RDC_PDAP79	SEMA42 B2 / G15
SAI6	RDC_PDAP80	SEMA42 B2 / G16
SAI5	RDC_PDAP81	SEMA42 B2 / G17
SAI4	RDC_PDAP82	SEMA42 B2 / G18
Reserved for SDMA2 internal memory	RDC_PDAP83	SEMA42 B2 / G19
uSDHC1	RDC_PDAP84	SEMA42 B2 / G20
uSDHC2	RDC_PDAP85	SEMA42 B2 / G21
MIPI_CSI2	RDC_PDAP86	SEMA42 B2 / G22
MIPI_CSI_PHY2	RDC_PDAP87	SEMA42 B2 / G23
CSI2	RDC_PDAP88	SEMA42 B2 / G24
Reserved for SDMA2 internal memory	RDC_PDAP89	SEMA42 B2 / G25
SPBA2	RDC_PDAP90	SEMA42 B2 / G26
QSPI	RDC_PDAP91	SEMA42 B2 / G27
Reserved	RDC_PDAP92	SEMA42 B2 / G28
SDMA1	RDC_PDAP93	SEMA42 B2 / G29
ENET1	RDC_PDAP94	SEMA42 B2 / G30
Reserved	RDC_PDAP95	SEMA42 B2 / G31
Reserved for SDMA internal memory	RDC_PDAP96	SEMA42 B2 / G32
SPDIF1	RDC_PDAP97	SEMA42 B2 / G33
eCSPI1	RDC_PDAP98	SEMA42 B2 / G34
eCSPI2	RDC_PDAP99	SEMA42 B2 / G35
eCSPI3	RDC_PDAP100	SEMA42 B2 / G36
Reserved	RDC_PDAP101	SEMA42 B2 / G37
UART1	RDC_PDAP102	SEMA42 B2 / G38
Reserved for SDMA internal registers	RDC_PDAP103	SEMA42 B2 / G39
UART3	RDC_PDAP104	SEMA42 B2 / G40
UART2	RDC_PDAP105	SEMA42 B2 / G41
SPDIF2	RDC_PDAP106	SEMA42 B2 / G42
SAI2	RDC_PDAP107	SEMA42 B2 / G43
SAI3	RDC_PDAP108	SEMA42 B2 / G44
Reserved	RDC_PDAP109	SEMA42 B2 / G45
Reserved for SDMA internal registers	RDC_PDAP110	SEMA42 B2 / G46
SPBA1	RDC_PDAP111	SEMA42 B2 / G47

Table continues on the next page...

Table 3-3. RDC Peripheral Mapping (continued)

Peripheral	RDC PDAP register	RDC_SEMA42 block/gate register
module_en_glbl[0]	RDC_PDAP112	SEMA42 B2 / G48
module_en_glbl[0]	RDC_PDAP113	SEMA42 B2 / G49
CAAM	RDC_PDAP114	SEMA42 B2 / G50

3.2.4.3 Memory Region Map

The number of memories with domain isolation support varies per device. The number of memory regions for a particular memory and the size of those regions varies per memory gasket. Each region of memory has a set of registers to define the boundaries of the region based on start and end addresses, a control register to set the domain access permissions and enable the region, and a status register to determin if access was denied to a region.

For this device, refer to the table below to determine the memories with domain support, the number of regions for each memory, the region resolution, the identifying numbers for the sets of memory region registers, and the addresses of the RDC registers to access the sets of Memory Region registers.

Table 3-4. Memory Region Mapping

Memory/Port	Number of Regions	Region Resolution	Memory Region Register Set Number (e.g. MRSA, MREA, MRC, MRVS)	Register Address Range
DRAM	8	4 KB	0-7	0x800-0x87C
PCIe2	4	4 KB	8-11	0x880-0x8BC
QSPI1	8	4 KB	12-19	0x8C0-0x93C
PCIe1	4	4 KB	20-23	0x940-0x97C
OCRAM	5	128 B	24-28	0x980-0x9CC
OCRAM_S	5	128 B	29-33	0x9D0-0xA1C
TCM	5	128 B	34-38	0xA20-0xA6C
GIC	4	4 KB	39-42	0xA70-0xAAC
USBMIX	4	4 KB	43-46	0xAB0-0xAEC
GPU	4	4 KB	47-50	0xAF0-0xB2C
VPU (Decoder)	4	4 KB	51-54	0xB30-0xB6C
DEBUG (DAP)	4	4 KB	55-58	0xB70-0xBAC
DDRC (REG)	5	4 KB	59-63	0xBB0-0xBFC

3.2.4.4 Memory Region Access Control

The RDC also supports multiple regions with flexible access permission control for each memory space. The memory region access control is defined in the table below. The security and exclusive support for DRAM is done by the TZASC and the DRAM controller itself. Note that the RDC and TrustZone features are completely independent of each other but will be used together.

Table 3-5. Memory Region Access Control

Memory Space	Regions Supported	Granularity	Security Support	Exclusive Support
DRAM	8	4 KB	Disable	Disable
QSPI	8	4 KB	Enable	Enable
PCIe	8	4 KB	Enable	Enable
OCRAM	5	128 KB	Disable	Enable
OCRAM_S	5	128 KB	Disable	Enable
TCM	5	128 KB	Enable	Enable
GIC	4	4 KB	Enable	Disable
USBMIX	4	4 KB	Enable	Disable
GPU	4	4 KB	Enable	Disable
VPU (Decoder)	4	4 KB	Enable	Disable
VPU (Peripheral)	4	4 KB	Enable	Disable
DRAM Register	4	4 KB	Enable	Disable
DEBUG	4	4 KB	Enable	Disable

3.2.5 RDC Memory Map/Register Definition

RDC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0000	Version Information (RDC_VIR)	32	R	0376_E204h	3.2.5.1/61
303D_0024	Status (RDC_STAT)	32	R/W	0000_0100h	3.2.5.2/62
303D_0028	Interrupt and Control (RDC_INTCTRL)	32	R/W	0000_0000h	3.2.5.3/63
303D_002C	Interrupt Status (RDC_INTSTAT)	32	R/W	See section	3.2.5.4/63
303D_0200	Master Domain Assignment (RDC_MDA0)	32	R/W	0000_0000h	3.2.5.5/64
303D_0204	Master Domain Assignment (RDC_MDA1)	32	R/W	0000_0000h	3.2.5.5/64
303D_0208	Master Domain Assignment (RDC_MDA2)	32	R/W	0000_0000h	3.2.5.5/64

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_020C	Master Domain Assignment (RDC_MDA3)	32	R/W	0000_0000h	3.2.5.5/64
303D_0210	Master Domain Assignment (RDC_MDA4)	32	R/W	0000_0000h	3.2.5.5/64
303D_0214	Master Domain Assignment (RDC_MDA5)	32	R/W	0000_0000h	3.2.5.5/64
303D_0218	Master Domain Assignment (RDC_MDA6)	32	R/W	0000_0000h	3.2.5.5/64
303D_021C	Master Domain Assignment (RDC_MDA7)	32	R/W	0000_0000h	3.2.5.5/64
303D_0220	Master Domain Assignment (RDC_MDA8)	32	R/W	0000_0000h	3.2.5.5/64
303D_0224	Master Domain Assignment (RDC_MDA9)	32	R/W	0000_0000h	3.2.5.5/64
303D_0228	Master Domain Assignment (RDC_MDA10)	32	R/W	0000_0000h	3.2.5.5/64
303D_022C	Master Domain Assignment (RDC_MDA11)	32	R/W	0000_0000h	3.2.5.5/64
303D_0230	Master Domain Assignment (RDC_MDA12)	32	R/W	0000_0000h	3.2.5.5/64
303D_0234	Master Domain Assignment (RDC_MDA13)	32	R/W	0000_0000h	3.2.5.5/64
303D_0238	Master Domain Assignment (RDC_MDA14)	32	R/W	0000_0000h	3.2.5.5/64
303D_023C	Master Domain Assignment (RDC_MDA15)	32	R/W	0000_0000h	3.2.5.5/64
303D_0240	Master Domain Assignment (RDC_MDA16)	32	R/W	0000_0000h	3.2.5.5/64
303D_0244	Master Domain Assignment (RDC_MDA17)	32	R/W	0000_0000h	3.2.5.5/64
303D_0248	Master Domain Assignment (RDC_MDA18)	32	R/W	0000_0000h	3.2.5.5/64
303D_024C	Master Domain Assignment (RDC_MDA19)	32	R/W	0000_0000h	3.2.5.5/64
303D_0250	Master Domain Assignment (RDC_MDA20)	32	R/W	0000_0000h	3.2.5.5/64
303D_0254	Master Domain Assignment (RDC_MDA21)	32	R/W	0000_0000h	3.2.5.5/64
303D_0258	Master Domain Assignment (RDC_MDA22)	32	R/W	0000_0000h	3.2.5.5/64
303D_025C	Master Domain Assignment (RDC_MDA23)	32	R/W	0000_0000h	3.2.5.5/64
303D_0260	Master Domain Assignment (RDC_MDA24)	32	R/W	0000_0000h	3.2.5.5/64
303D_0264	Master Domain Assignment (RDC_MDA25)	32	R/W	0000_0000h	3.2.5.5/64
303D_0268	Master Domain Assignment (RDC_MDA26)	32	R/W	0000_0000h	3.2.5.5/64
303D_0400	Peripheral Domain Access Permissions (RDC_PDAP0)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0404	Peripheral Domain Access Permissions (RDC_PDAP1)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0408	Peripheral Domain Access Permissions (RDC_PDAP2)	32	R/W	0000_00FFh	3.2.5.6/65
303D_040C	Peripheral Domain Access Permissions (RDC_PDAP3)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0410	Peripheral Domain Access Permissions (RDC_PDAP4)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0414	Peripheral Domain Access Permissions (RDC_PDAP5)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0418	Peripheral Domain Access Permissions (RDC_PDAP6)	32	R/W	0000_00FFh	3.2.5.6/65
303D_041C	Peripheral Domain Access Permissions (RDC_PDAP7)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0420	Peripheral Domain Access Permissions (RDC_PDAP8)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0424	Peripheral Domain Access Permissions (RDC_PDAP9)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0428	Peripheral Domain Access Permissions (RDC_PDAP10)	32	R/W	0000_00FFh	3.2.5.6/65
303D_042C	Peripheral Domain Access Permissions (RDC_PDAP11)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0430	Peripheral Domain Access Permissions (RDC_PDAP12)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0434	Peripheral Domain Access Permissions (RDC_PDAP13)	32	R/W	0000_00FFh	3.2.5.6/65

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0438	Peripheral Domain Access Permissions (RDC_PDAP14)	32	R/W	0000_00FFh	3.2.5.6/65
303D_043C	Peripheral Domain Access Permissions (RDC_PDAP15)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0440	Peripheral Domain Access Permissions (RDC_PDAP16)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0444	Peripheral Domain Access Permissions (RDC_PDAP17)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0448	Peripheral Domain Access Permissions (RDC_PDAP18)	32	R/W	0000_00FFh	3.2.5.6/65
303D_044C	Peripheral Domain Access Permissions (RDC_PDAP19)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0450	Peripheral Domain Access Permissions (RDC_PDAP20)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0454	Peripheral Domain Access Permissions (RDC_PDAP21)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0458	Peripheral Domain Access Permissions (RDC_PDAP22)	32	R/W	0000_00FFh	3.2.5.6/65
303D_045C	Peripheral Domain Access Permissions (RDC_PDAP23)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0460	Peripheral Domain Access Permissions (RDC_PDAP24)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0464	Peripheral Domain Access Permissions (RDC_PDAP25)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0468	Peripheral Domain Access Permissions (RDC_PDAP26)	32	R/W	0000_00FFh	3.2.5.6/65
303D_046C	Peripheral Domain Access Permissions (RDC_PDAP27)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0470	Peripheral Domain Access Permissions (RDC_PDAP28)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0474	Peripheral Domain Access Permissions (RDC_PDAP29)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0478	Peripheral Domain Access Permissions (RDC_PDAP30)	32	R/W	0000_00FFh	3.2.5.6/65
303D_047C	Peripheral Domain Access Permissions (RDC_PDAP31)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0480	Peripheral Domain Access Permissions (RDC_PDAP32)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0484	Peripheral Domain Access Permissions (RDC_PDAP33)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0488	Peripheral Domain Access Permissions (RDC_PDAP34)	32	R/W	0000_00FFh	3.2.5.6/65
303D_048C	Peripheral Domain Access Permissions (RDC_PDAP35)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0490	Peripheral Domain Access Permissions (RDC_PDAP36)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0494	Peripheral Domain Access Permissions (RDC_PDAP37)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0498	Peripheral Domain Access Permissions (RDC_PDAP38)	32	R/W	0000_00FFh	3.2.5.6/65
303D_049C	Peripheral Domain Access Permissions (RDC_PDAP39)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04A0	Peripheral Domain Access Permissions (RDC_PDAP40)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04A4	Peripheral Domain Access Permissions (RDC_PDAP41)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04A8	Peripheral Domain Access Permissions (RDC_PDAP42)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04AC	Peripheral Domain Access Permissions (RDC_PDAP43)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04B0	Peripheral Domain Access Permissions (RDC_PDAP44)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04B4	Peripheral Domain Access Permissions (RDC_PDAP45)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04B8	Peripheral Domain Access Permissions (RDC_PDAP46)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04BC	Peripheral Domain Access Permissions (RDC_PDAP47)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04C0	Peripheral Domain Access Permissions (RDC_PDAP48)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04C4	Peripheral Domain Access Permissions (RDC_PDAP49)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04C8	Peripheral Domain Access Permissions (RDC_PDAP50)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04CC	Peripheral Domain Access Permissions (RDC_PDAP51)	32	R/W	0000_00FFh	3.2.5.6/65

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_04D0	Peripheral Domain Access Permissions (RDC_PDAP52)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04D4	Peripheral Domain Access Permissions (RDC_PDAP53)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04D8	Peripheral Domain Access Permissions (RDC_PDAP54)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04DC	Peripheral Domain Access Permissions (RDC_PDAP55)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04E0	Peripheral Domain Access Permissions (RDC_PDAP56)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04E4	Peripheral Domain Access Permissions (RDC_PDAP57)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04E8	Peripheral Domain Access Permissions (RDC_PDAP58)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04EC	Peripheral Domain Access Permissions (RDC_PDAP59)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04F0	Peripheral Domain Access Permissions (RDC_PDAP60)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04F4	Peripheral Domain Access Permissions (RDC_PDAP61)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04F8	Peripheral Domain Access Permissions (RDC_PDAP62)	32	R/W	0000_00FFh	3.2.5.6/65
303D_04FC	Peripheral Domain Access Permissions (RDC_PDAP63)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0500	Peripheral Domain Access Permissions (RDC_PDAP64)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0504	Peripheral Domain Access Permissions (RDC_PDAP65)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0508	Peripheral Domain Access Permissions (RDC_PDAP66)	32	R/W	0000_00FFh	3.2.5.6/65
303D_050C	Peripheral Domain Access Permissions (RDC_PDAP67)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0510	Peripheral Domain Access Permissions (RDC_PDAP68)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0514	Peripheral Domain Access Permissions (RDC_PDAP69)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0518	Peripheral Domain Access Permissions (RDC_PDAP70)	32	R/W	0000_00FFh	3.2.5.6/65
303D_051C	Peripheral Domain Access Permissions (RDC_PDAP71)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0520	Peripheral Domain Access Permissions (RDC_PDAP72)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0524	Peripheral Domain Access Permissions (RDC_PDAP73)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0528	Peripheral Domain Access Permissions (RDC_PDAP74)	32	R/W	0000_00FFh	3.2.5.6/65
303D_052C	Peripheral Domain Access Permissions (RDC_PDAP75)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0530	Peripheral Domain Access Permissions (RDC_PDAP76)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0534	Peripheral Domain Access Permissions (RDC_PDAP77)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0538	Peripheral Domain Access Permissions (RDC_PDAP78)	32	R/W	0000_00FFh	3.2.5.6/65
303D_053C	Peripheral Domain Access Permissions (RDC_PDAP79)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0540	Peripheral Domain Access Permissions (RDC_PDAP80)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0544	Peripheral Domain Access Permissions (RDC_PDAP81)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0548	Peripheral Domain Access Permissions (RDC_PDAP82)	32	R/W	0000_00FFh	3.2.5.6/65
303D_054C	Peripheral Domain Access Permissions (RDC_PDAP83)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0550	Peripheral Domain Access Permissions (RDC_PDAP84)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0554	Peripheral Domain Access Permissions (RDC_PDAP85)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0558	Peripheral Domain Access Permissions (RDC_PDAP86)	32	R/W	0000_00FFh	3.2.5.6/65
303D_055C	Peripheral Domain Access Permissions (RDC_PDAP87)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0560	Peripheral Domain Access Permissions (RDC_PDAP88)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0564	Peripheral Domain Access Permissions (RDC_PDAP89)	32	R/W	0000_00FFh	3.2.5.6/65

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0568	Peripheral Domain Access Permissions (RDC_PDAP90)	32	R/W	0000_00FFh	3.2.5.6/65
303D_056C	Peripheral Domain Access Permissions (RDC_PDAP91)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0570	Peripheral Domain Access Permissions (RDC_PDAP92)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0574	Peripheral Domain Access Permissions (RDC_PDAP93)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0578	Peripheral Domain Access Permissions (RDC_PDAP94)	32	R/W	0000_00FFh	3.2.5.6/65
303D_057C	Peripheral Domain Access Permissions (RDC_PDAP95)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0580	Peripheral Domain Access Permissions (RDC_PDAP96)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0584	Peripheral Domain Access Permissions (RDC_PDAP97)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0588	Peripheral Domain Access Permissions (RDC_PDAP98)	32	R/W	0000_00FFh	3.2.5.6/65
303D_058C	Peripheral Domain Access Permissions (RDC_PDAP99)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0590	Peripheral Domain Access Permissions (RDC_PDAP100)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0594	Peripheral Domain Access Permissions (RDC_PDAP101)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0598	Peripheral Domain Access Permissions (RDC_PDAP102)	32	R/W	0000_00FFh	3.2.5.6/65
303D_059C	Peripheral Domain Access Permissions (RDC_PDAP103)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05A0	Peripheral Domain Access Permissions (RDC_PDAP104)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05A4	Peripheral Domain Access Permissions (RDC_PDAP105)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05A8	Peripheral Domain Access Permissions (RDC_PDAP106)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05AC	Peripheral Domain Access Permissions (RDC_PDAP107)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05B0	Peripheral Domain Access Permissions (RDC_PDAP108)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05B4	Peripheral Domain Access Permissions (RDC_PDAP109)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05B8	Peripheral Domain Access Permissions (RDC_PDAP110)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05BC	Peripheral Domain Access Permissions (RDC_PDAP111)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05C0	Peripheral Domain Access Permissions (RDC_PDAP112)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05C4	Peripheral Domain Access Permissions (RDC_PDAP113)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05C8	Peripheral Domain Access Permissions (RDC_PDAP114)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05CC	Peripheral Domain Access Permissions (RDC_PDAP115)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05D0	Peripheral Domain Access Permissions (RDC_PDAP116)	32	R/W	0000_00FFh	3.2.5.6/65
303D_05D4	Peripheral Domain Access Permissions (RDC_PDAP117)	32	R/W	0000_00FFh	3.2.5.6/65
303D_0800	Memory Region Start Address (RDC_MRSA0)	32	R/W	Undefined	3.2.5.7/66
303D_0804	Memory Region End Address (RDC_MREA0)	32	R/W	Undefined	3.2.5.8/68
303D_0808	Memory Region Control (RDC_MRC0)	32	R/W	0000_00FFh	3.2.5.9/69
303D_080C	Memory Region Violation Status (RDC_MRVS0)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0810	Memory Region Start Address (RDC_MRSA1)	32	R/W	Undefined	3.2.5.7/66
303D_0814	Memory Region End Address (RDC_MREA1)	32	R/W	Undefined	3.2.5.8/68
303D_0818	Memory Region Control (RDC_MRC1)	32	R/W	0000_00FFh	3.2.5.9/69
303D_081C	Memory Region Violation Status (RDC_MRVS1)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0820	Memory Region Start Address (RDC_MRSA2)	32	R/W	Undefined	3.2.5.7/66
303D_0824	Memory Region End Address (RDC_MREA2)	32	R/W	Undefined	3.2.5.8/68
303D_0828	Memory Region Control (RDC_MRC2)	32	R/W	0000_00FFh	3.2.5.9/69
303D_082C	Memory Region Violation Status (RDC_MRVS2)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0830	Memory Region Start Address (RDC_MRSA3)	32	R/W	Undefined	3.2.5.7/66
303D_0834	Memory Region End Address (RDC_MREA3)	32	R/W	Undefined	3.2.5.8/68
303D_0838	Memory Region Control (RDC_MRC3)	32	R/W	0000_00FFh	3.2.5.9/69
303D_083C	Memory Region Violation Status (RDC_MRVS3)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0840	Memory Region Start Address (RDC_MRSA4)	32	R/W	Undefined	3.2.5.7/66
303D_0844	Memory Region End Address (RDC_MREA4)	32	R/W	Undefined	3.2.5.8/68
303D_0848	Memory Region Control (RDC_MRC4)	32	R/W	0000_00FFh	3.2.5.9/69
303D_084C	Memory Region Violation Status (RDC_MRVS4)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0850	Memory Region Start Address (RDC_MRSA5)	32	R/W	Undefined	3.2.5.7/66
303D_0854	Memory Region End Address (RDC_MREA5)	32	R/W	Undefined	3.2.5.8/68
303D_0858	Memory Region Control (RDC_MRC5)	32	R/W	0000_00FFh	3.2.5.9/69
303D_085C	Memory Region Violation Status (RDC_MRVS5)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0860	Memory Region Start Address (RDC_MRSA6)	32	R/W	Undefined	3.2.5.7/66
303D_0864	Memory Region End Address (RDC_MREA6)	32	R/W	Undefined	3.2.5.8/68
303D_0868	Memory Region Control (RDC_MRC6)	32	R/W	0000_00FFh	3.2.5.9/69
303D_086C	Memory Region Violation Status (RDC_MRVS6)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0870	Memory Region Start Address (RDC_MRSA7)	32	R/W	Undefined	3.2.5.7/66
303D_0874	Memory Region End Address (RDC_MREA7)	32	R/W	Undefined	3.2.5.8/68
303D_0878	Memory Region Control (RDC_MRC7)	32	R/W	0000_00FFh	3.2.5.9/69
303D_087C	Memory Region Violation Status (RDC_MRVS7)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0880	Memory Region Start Address (RDC_MRSA8)	32	R/W	Undefined	3.2.5.7/66
303D_0884	Memory Region End Address (RDC_MREA8)	32	R/W	Undefined	3.2.5.8/68
303D_0888	Memory Region Control (RDC_MRC8)	32	R/W	0000_00FFh	3.2.5.9/69
303D_088C	Memory Region Violation Status (RDC_MRVS8)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0890	Memory Region Start Address (RDC_MRSA9)	32	R/W	Undefined	3.2.5.7/66
303D_0894	Memory Region End Address (RDC_MREA9)	32	R/W	Undefined	3.2.5.8/68
303D_0898	Memory Region Control (RDC_MRC9)	32	R/W	0000_00FFh	3.2.5.9/69
303D_089C	Memory Region Violation Status (RDC_MRVS9)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_08A0	Memory Region Start Address (RDC_MRSA10)	32	R/W	Undefined	3.2.5.7/66
303D_08A4	Memory Region End Address (RDC_MREA10)	32	R/W	Undefined	3.2.5.8/68
303D_08A8	Memory Region Control (RDC_MRC10)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08AC	Memory Region Violation Status (RDC_MRVS10)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_08B0	Memory Region Start Address (RDC_MRSA11)	32	R/W	Undefined	3.2.5.7/66
303D_08B4	Memory Region End Address (RDC_MREA11)	32	R/W	Undefined	3.2.5.8/68
303D_08B8	Memory Region Control (RDC_MRC11)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08BC	Memory Region Violation Status (RDC_MRVS11)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_08C0	Memory Region Start Address (RDC_MRSA12)	32	R/W	Undefined	3.2.5.7/66
303D_08C4	Memory Region End Address (RDC_MREA12)	32	R/W	Undefined	3.2.5.8/68
303D_08C8	Memory Region Control (RDC_MRC12)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08CC	Memory Region Violation Status (RDC_MRVS12)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_08D0	Memory Region Start Address (RDC_MRSA13)	32	R/W	Undefined	3.2.5.7/66
303D_08D4	Memory Region End Address (RDC_MREA13)	32	R/W	Undefined	3.2.5.8/68
303D_08D8	Memory Region Control (RDC_MRC13)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08DC	Memory Region Violation Status (RDC_MRVS13)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_08E0	Memory Region Start Address (RDC_MRSA14)	32	R/W	Undefined	3.2.5.7/66
303D_08E4	Memory Region End Address (RDC_MREA14)	32	R/W	Undefined	3.2.5.8/68
303D_08E8	Memory Region Control (RDC_MRC14)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08EC	Memory Region Violation Status (RDC_MRVS14)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_08F0	Memory Region Start Address (RDC_MRSA15)	32	R/W	Undefined	3.2.5.7/66
303D_08F4	Memory Region End Address (RDC_MREA15)	32	R/W	Undefined	3.2.5.8/68
303D_08F8	Memory Region Control (RDC_MRC15)	32	R/W	0000_00FFh	3.2.5.9/69
303D_08FC	Memory Region Violation Status (RDC_MRVS15)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0900	Memory Region Start Address (RDC_MRSA16)	32	R/W	Undefined	3.2.5.7/66
303D_0904	Memory Region End Address (RDC_MREA16)	32	R/W	Undefined	3.2.5.8/68
303D_0908	Memory Region Control (RDC_MRC16)	32	R/W	0000_00FFh	3.2.5.9/69
303D_090C	Memory Region Violation Status (RDC_MRVS16)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0910	Memory Region Start Address (RDC_MRSA17)	32	R/W	Undefined	3.2.5.7/66
303D_0914	Memory Region End Address (RDC_MREA17)	32	R/W	Undefined	3.2.5.8/68
303D_0918	Memory Region Control (RDC_MRC17)	32	R/W	0000_00FFh	3.2.5.9/69
303D_091C	Memory Region Violation Status (RDC_MRVS17)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0920	Memory Region Start Address (RDC_MRSA18)	32	R/W	Undefined	3.2.5.7/66
303D_0924	Memory Region End Address (RDC_MREA18)	32	R/W	Undefined	3.2.5.8/68
303D_0928	Memory Region Control (RDC_MRC18)	32	R/W	0000_00FFh	3.2.5.9/69
303D_092C	Memory Region Violation Status (RDC_MRVS18)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0930	Memory Region Start Address (RDC_MRSA19)	32	R/W	Undefined	3.2.5.7/66
303D_0934	Memory Region End Address (RDC_MREA19)	32	R/W	Undefined	3.2.5.8/68
303D_0938	Memory Region Control (RDC_MRC19)	32	R/W	0000_00FFh	3.2.5.9/69
303D_093C	Memory Region Violation Status (RDC_MRVS19)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0940	Memory Region Start Address (RDC_MRSA20)	32	R/W	Undefined	3.2.5.7/66
303D_0944	Memory Region End Address (RDC_MREA20)	32	R/W	Undefined	3.2.5.8/68
303D_0948	Memory Region Control (RDC_MRC20)	32	R/W	0000_00FFh	3.2.5.9/69
303D_094C	Memory Region Violation Status (RDC_MRVS20)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0950	Memory Region Start Address (RDC_MRSA21)	32	R/W	Undefined	3.2.5.7/66
303D_0954	Memory Region End Address (RDC_MREA21)	32	R/W	Undefined	3.2.5.8/68
303D_0958	Memory Region Control (RDC_MRC21)	32	R/W	0000_00FFh	3.2.5.9/69
303D_095C	Memory Region Violation Status (RDC_MRVS21)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0960	Memory Region Start Address (RDC_MRSA22)	32	R/W	Undefined	3.2.5.7/66
303D_0964	Memory Region End Address (RDC_MREA22)	32	R/W	Undefined	3.2.5.8/68
303D_0968	Memory Region Control (RDC_MRC22)	32	R/W	0000_00FFh	3.2.5.9/69
303D_096C	Memory Region Violation Status (RDC_MRVS22)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0970	Memory Region Start Address (RDC_MRSA23)	32	R/W	Undefined	3.2.5.7/66
303D_0974	Memory Region End Address (RDC_MREA23)	32	R/W	Undefined	3.2.5.8/68
303D_0978	Memory Region Control (RDC_MRC23)	32	R/W	0000_00FFh	3.2.5.9/69
303D_097C	Memory Region Violation Status (RDC_MRVS23)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0980	Memory Region Start Address (RDC_MRSA24)	32	R/W	Undefined	3.2.5.7/66
303D_0984	Memory Region End Address (RDC_MREA24)	32	R/W	Undefined	3.2.5.8/68
303D_0988	Memory Region Control (RDC_MRC24)	32	R/W	0000_00FFh	3.2.5.9/69
303D_098C	Memory Region Violation Status (RDC_MRVS24)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0990	Memory Region Start Address (RDC_MRSA25)	32	R/W	Undefined	3.2.5.7/66
303D_0994	Memory Region End Address (RDC_MREA25)	32	R/W	Undefined	3.2.5.8/68
303D_0998	Memory Region Control (RDC_MRC25)	32	R/W	0000_00FFh	3.2.5.9/69
303D_099C	Memory Region Violation Status (RDC_MRVS25)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_09A0	Memory Region Start Address (RDC_MRSA26)	32	R/W	Undefined	3.2.5.7/66
303D_09A4	Memory Region End Address (RDC_MREA26)	32	R/W	Undefined	3.2.5.8/68
303D_09A8	Memory Region Control (RDC_MRC26)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09AC	Memory Region Violation Status (RDC_MRVS26)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_09B0	Memory Region Start Address (RDC_MRSA27)	32	R/W	Undefined	3.2.5.7/66
303D_09B4	Memory Region End Address (RDC_MREA27)	32	R/W	Undefined	3.2.5.8/68
303D_09B8	Memory Region Control (RDC_MRC27)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09BC	Memory Region Violation Status (RDC_MRVS27)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_09C0	Memory Region Start Address (RDC_MRSA28)	32	R/W	Undefined	3.2.5.7/66
303D_09C4	Memory Region End Address (RDC_MREA28)	32	R/W	Undefined	3.2.5.8/68
303D_09C8	Memory Region Control (RDC_MRC28)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09CC	Memory Region Violation Status (RDC_MRVS28)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_09D0	Memory Region Start Address (RDC_MRSA29)	32	R/W	Undefined	3.2.5.7/66
303D_09D4	Memory Region End Address (RDC_MREA29)	32	R/W	Undefined	3.2.5.8/68
303D_09D8	Memory Region Control (RDC_MRC29)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09DC	Memory Region Violation Status (RDC_MRVS29)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_09E0	Memory Region Start Address (RDC_MRSA30)	32	R/W	Undefined	3.2.5.7/66
303D_09E4	Memory Region End Address (RDC_MREA30)	32	R/W	Undefined	3.2.5.8/68
303D_09E8	Memory Region Control (RDC_MRC30)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09EC	Memory Region Violation Status (RDC_MRVS30)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_09F0	Memory Region Start Address (RDC_MRSA31)	32	R/W	Undefined	3.2.5.7/66
303D_09F4	Memory Region End Address (RDC_MREA31)	32	R/W	Undefined	3.2.5.8/68
303D_09F8	Memory Region Control (RDC_MRC31)	32	R/W	0000_00FFh	3.2.5.9/69
303D_09FC	Memory Region Violation Status (RDC_MRVS31)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A00	Memory Region Start Address (RDC_MRSA32)	32	R/W	Undefined	3.2.5.7/66
303D_0A04	Memory Region End Address (RDC_MREA32)	32	R/W	Undefined	3.2.5.8/68
303D_0A08	Memory Region Control (RDC_MRC32)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A0C	Memory Region Violation Status (RDC_MRVS32)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A10	Memory Region Start Address (RDC_MRSA33)	32	R/W	Undefined	3.2.5.7/66
303D_0A14	Memory Region End Address (RDC_MREA33)	32	R/W	Undefined	3.2.5.8/68
303D_0A18	Memory Region Control (RDC_MRC33)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A1C	Memory Region Violation Status (RDC_MRVS33)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0A20	Memory Region Start Address (RDC_MRSA34)	32	R/W	Undefined	3.2.5.7/66
303D_0A24	Memory Region End Address (RDC_MREA34)	32	R/W	Undefined	3.2.5.8/68
303D_0A28	Memory Region Control (RDC_MRC34)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A2C	Memory Region Violation Status (RDC_MRVS34)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A30	Memory Region Start Address (RDC_MRSA35)	32	R/W	Undefined	3.2.5.7/66
303D_0A34	Memory Region End Address (RDC_MREA35)	32	R/W	Undefined	3.2.5.8/68
303D_0A38	Memory Region Control (RDC_MRC35)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A3C	Memory Region Violation Status (RDC_MRVS35)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A40	Memory Region Start Address (RDC_MRSA36)	32	R/W	Undefined	3.2.5.7/66
303D_0A44	Memory Region End Address (RDC_MREA36)	32	R/W	Undefined	3.2.5.8/68
303D_0A48	Memory Region Control (RDC_MRC36)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A4C	Memory Region Violation Status (RDC_MRVS36)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A50	Memory Region Start Address (RDC_MRSA37)	32	R/W	Undefined	3.2.5.7/66
303D_0A54	Memory Region End Address (RDC_MREA37)	32	R/W	Undefined	3.2.5.8/68
303D_0A58	Memory Region Control (RDC_MRC37)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A5C	Memory Region Violation Status (RDC_MRVS37)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A60	Memory Region Start Address (RDC_MRSA38)	32	R/W	Undefined	3.2.5.7/66
303D_0A64	Memory Region End Address (RDC_MREA38)	32	R/W	Undefined	3.2.5.8/68
303D_0A68	Memory Region Control (RDC_MRC38)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A6C	Memory Region Violation Status (RDC_MRVS38)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A70	Memory Region Start Address (RDC_MRSA39)	32	R/W	Undefined	3.2.5.7/66
303D_0A74	Memory Region End Address (RDC_MREA39)	32	R/W	Undefined	3.2.5.8/68
303D_0A78	Memory Region Control (RDC_MRC39)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A7C	Memory Region Violation Status (RDC_MRVS39)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A80	Memory Region Start Address (RDC_MRSA40)	32	R/W	Undefined	3.2.5.7/66
303D_0A84	Memory Region End Address (RDC_MREA40)	32	R/W	Undefined	3.2.5.8/68
303D_0A88	Memory Region Control (RDC_MRC40)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A8C	Memory Region Violation Status (RDC_MRVS40)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0A90	Memory Region Start Address (RDC_MRSA41)	32	R/W	Undefined	3.2.5.7/66
303D_0A94	Memory Region End Address (RDC_MREA41)	32	R/W	Undefined	3.2.5.8/68
303D_0A98	Memory Region Control (RDC_MRC41)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0A9C	Memory Region Violation Status (RDC_MRVS41)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0AA0	Memory Region Start Address (RDC_MRSA42)	32	R/W	Undefined	3.2.5.7/66
303D_0AA4	Memory Region End Address (RDC_MREA42)	32	R/W	Undefined	3.2.5.8/68
303D_0AA8	Memory Region Control (RDC_MRC42)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0AAC	Memory Region Violation Status (RDC_MRVS42)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0AB0	Memory Region Start Address (RDC_MRSA43)	32	R/W	Undefined	3.2.5.7/66
303D_0AB4	Memory Region End Address (RDC_MREA43)	32	R/W	Undefined	3.2.5.8/68
303D_0AB8	Memory Region Control (RDC_MRC43)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0ABC	Memory Region Violation Status (RDC_MRVS43)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0AC0	Memory Region Start Address (RDC_MRSA44)	32	R/W	Undefined	3.2.5.7/66
303D_0AC4	Memory Region End Address (RDC_MREA44)	32	R/W	Undefined	3.2.5.8/68
303D_0AC8	Memory Region Control (RDC_MRC44)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0ACC	Memory Region Violation Status (RDC_MRVS44)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0AD0	Memory Region Start Address (RDC_MRSA45)	32	R/W	Undefined	3.2.5.7/66
303D_0AD4	Memory Region End Address (RDC_MREA45)	32	R/W	Undefined	3.2.5.8/68
303D_0AD8	Memory Region Control (RDC_MRC45)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0ADC	Memory Region Violation Status (RDC_MRVS45)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0AE0	Memory Region Start Address (RDC_MRSA46)	32	R/W	Undefined	3.2.5.7/66
303D_0AE4	Memory Region End Address (RDC_MREA46)	32	R/W	Undefined	3.2.5.8/68
303D_0AE8	Memory Region Control (RDC_MRC46)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0AEC	Memory Region Violation Status (RDC_MRVS46)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0AF0	Memory Region Start Address (RDC_MRSA47)	32	R/W	Undefined	3.2.5.7/66
303D_0AF4	Memory Region End Address (RDC_MREA47)	32	R/W	Undefined	3.2.5.8/68
303D_0AF8	Memory Region Control (RDC_MRC47)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0AFC	Memory Region Violation Status (RDC_MRVS47)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0B00	Memory Region Start Address (RDC_MRSA48)	32	R/W	Undefined	3.2.5.7/66
303D_0B04	Memory Region End Address (RDC_MREA48)	32	R/W	Undefined	3.2.5.8/68
303D_0B08	Memory Region Control (RDC_MRC48)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0B0C	Memory Region Violation Status (RDC_MRVS48)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0B10	Memory Region Start Address (RDC_MRSA49)	32	R/W	Undefined	3.2.5.7/66
303D_0B14	Memory Region End Address (RDC_MREA49)	32	R/W	Undefined	3.2.5.8/68
303D_0B18	Memory Region Control (RDC_MRC49)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0B1C	Memory Region Violation Status (RDC_MRVS49)	32	R/W	0000_0000h	3.2.5.10/ 70

Table continues on the next page...

RDC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303D_0B20	Memory Region Start Address (RDC_MRSA50)	32	R/W	Undefined	3.2.5.7/66
303D_0B24	Memory Region End Address (RDC_MREA50)	32	R/W	Undefined	3.2.5.8/68
303D_0B28	Memory Region Control (RDC_MRC50)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0B2C	Memory Region Violation Status (RDC_MRVS50)	32	R/W	0000_0000h	3.2.5.10/ 70
303D_0B30	Memory Region Start Address (RDC_MRSA51)	32	R/W	Undefined	3.2.5.7/66
303D_0B34	Memory Region End Address (RDC_MREA51)	32	R/W	Undefined	3.2.5.8/68
303D_0B38	Memory Region Control (RDC_MRC51)	32	R/W	0000_00FFh	3.2.5.9/69
303D_0B3C	Memory Region Violation Status (RDC_MRVS51)	32	R/W	0000_0000h	3.2.5.10/ 70

3.2.5.1 Version Information (RDC_VIR)

The VIR provides version information including the number of domains, number of master slots, number of peripheral slots, and number of memory regions.

Address: 303D_0000h base + 0h offset = 303D_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved				NRGN								NPER								NMSTR								NDID			
W																																
Reset	0	0	0	0	0	0	1	1	0	1	1	1	0	1	1	0	1	1	1	0	0	0	1	0	0	0	0	0	0	1	0	0

RDC_VIR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–20 NRGN	Number of Memory Regions Indicates the number of memory regions in this instance of the RDC.
19–12 NPER	Number of Peripherals Indicates the number of peripherals that can be isolated or safe-shared
11–4 NMSTR	Number of Masters Indicates the number of masters supported by this instance of RDC.
NDID	Number of Domains Indicates the number of domain ids supported by this instance of the RDC. Add one to the register value to get the actual number of domains.

3.2.5.2 Status (RDC_STAT)

Address: 303D_0000h base + 24h offset = 303D_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

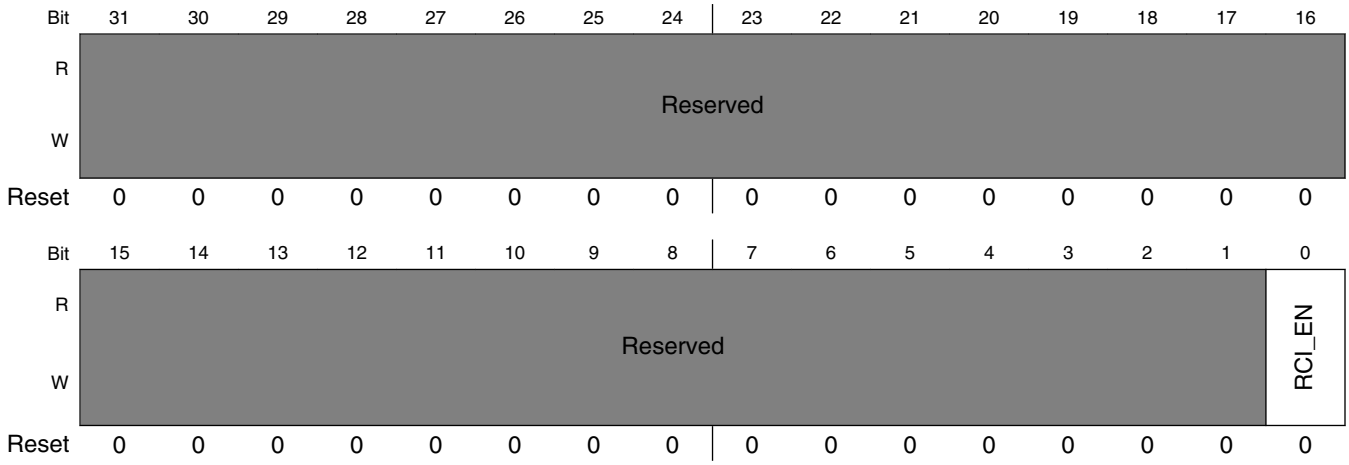
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							PDS	Reserved				DID			
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

RDC_STAT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 PDS	Power Domain Status Indicates if the "Power Down" memory regions are powered and available. Power Down memory regions are only those memory regions susceptible to power outage for power savings are unavailable if this is zero. "Always-On" memory regions remain available. Always On memory regions are those regions that are not powered down unless the entire SoC is powered down. This signal remains low until all access controls have been restored to the domain. 0 Power Down Domain is OFF 1 Power Down Domain is ON
7–4 Reserved	This field is reserved.
DID	Domain ID The Domain ID of the core or bus master that is reading this. The value is different for requests from different domains.

3.2.5.3 Interrupt and Control (RDC_INTCTRL)

Address: 303D_0000h base + 28h offset = 303D_0028h



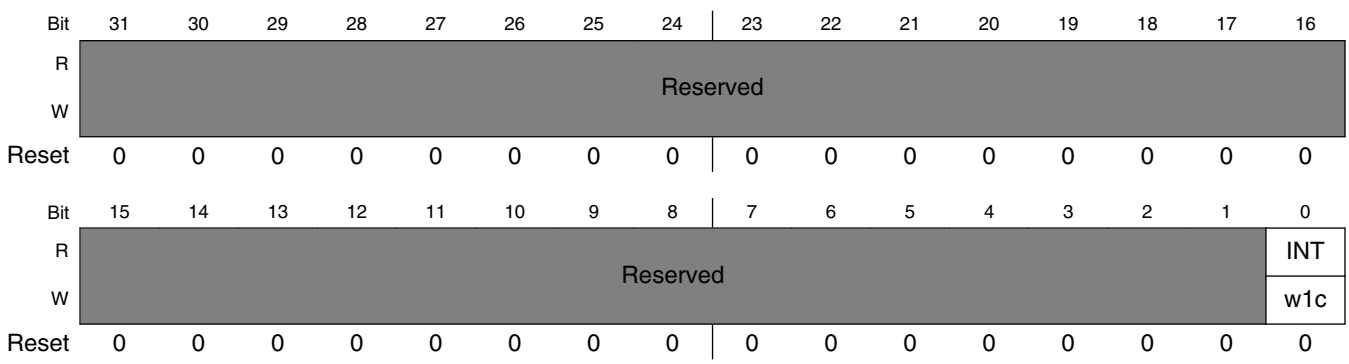
RDC_INTCTRL field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 RCI_EN	Restoration Complete Interrupt Interrupt generated when the RDC has completed restoring state to a recently re-powered memory regions. 0 Interrupt Disabled 1 Interrupt Enabled

3.2.5.4 Interrupt Status (RDC_INTSTAT)

Indication of Interrupt Pending for State Restoration

Address: 303D_0000h base + 2Ch offset = 303D_002Ch



RDC_INTSTAT field descriptions

Field	Description
31–1 Reserved	This field is reserved.
0 INT	Interrupt Status Indicates state of interrupt signal for state restoration. This is that status of the interrupt enabled in RDC_INTCTRL. Write one to interrupt status to clear it. 0 No Interrupt Pending 1 Interrupt Pending

3.2.5.5 Master Domain Assignment (RDC_MDA*n*)

Address: 303D_0000h base + 200h offset + (4d × i), where i=0d to 26d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DID	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_MDA*n* field descriptions

Field	Description
31 LCK	0 Not Locked 1 Locked
30–2 Reserved	This field is reserved.
DID	Domain ID Indicates the domain to which the Master is assigned 00 Master assigned to Processing Domain 0 01 Master assigned to Processing Domain 1 10 Master assigned to Processing Domain 2 11 Master assigned to Processing Domain 3

3.2.5.6 Peripheral Domain Access Permissions (RDC_PDAP*n*)

Address: 303D_0000h base + 400h offset + (4d × i), where i=0d to 117d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	SREQ	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								D3R	D3W	D2R	D2W	D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RDC_PDAP*n* field descriptions

Field	Description
31 LCK	Peripheral Permissions Lock When set prevents further modification of the Peripheral Domain Access Permissions (sticky bit until reset) 0 Not Locked 1 Locked
30 SREQ	Semaphore Required When set the hardware semaphore state enforces the semaphore lock. If a domain has access permissions and a semaphore has locked a shared peripheral then only the domain holding the semaphore signal can access this peripheral. 0 Semaphores have no effect 1 Semaphores are enforced
29–8 Reserved	This field is reserved.
7 D3R	Domain 3 Read Access 0 No Read Access 1 Read Access Allowed
6 D3W	Domain 3 Write Access 0 No Write Access 1 Write Access Allowed
5 D2R	Domain 2 Read Access 0 No Read Access 1 Read Access Allowed
4 D2W	Domain 2 Write Access 0 No Write Access 1 Write Access Allowed
3 D1R	Domain 1 Read Access

Table continues on the next page...

RDC_PDAP*n* field descriptions (continued)

Field	Description
	0 No Read Access 1 Read Access Allowed
2 D1W	Domain 1 Write Access 0 No Write Access 1 Write Access Allowed
1 D0R	Domain 0 Read Access 0 No Read Access 1 Read Access Allowed
0 D0W	Domain 0 Write Access 0 No Write Access 1 Write Access Allowed

3.2.5.7 Memory Region Start Address (RDC_MRSA*n*)

NOTE

The DDR space is 33-bit width. The RDC memory region registers are 32-bit width. The RDC configuration is the most significant bits in the DDR address space (32:1). To set the start address for this configuration, the MRSA value should be shifted 1-bit and added to the DDR base address. The example below illustrates how to calculate the proper start and end address value. Please refer to the Memory Map for the actual DDR base address.

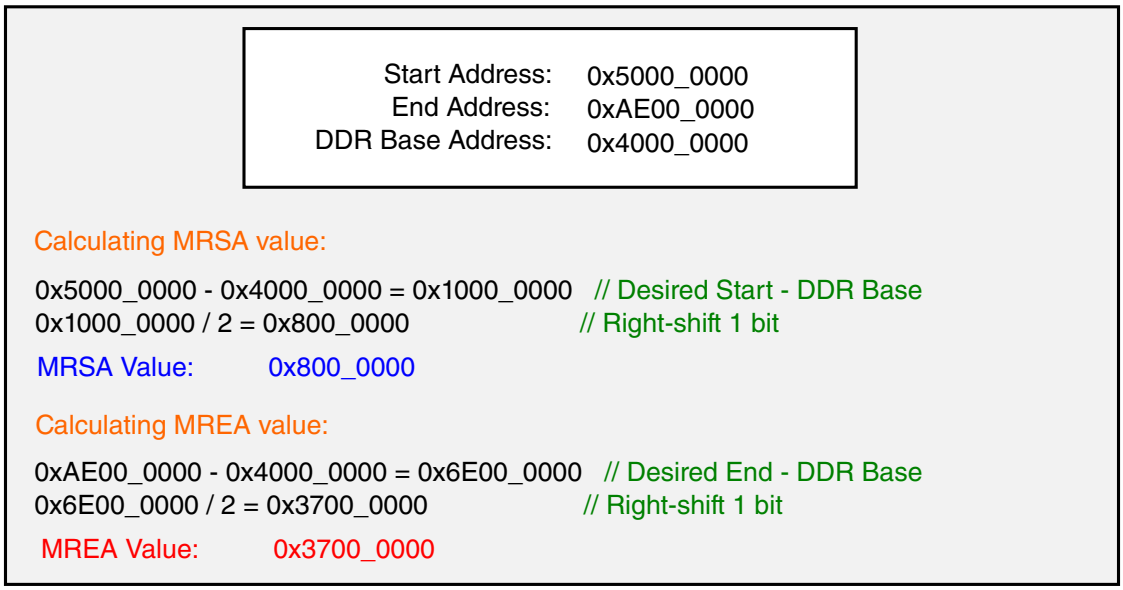


Figure 3-5. Calculating Address Value Examble

Address: 303D_0000h base + 800h offset + (16d × i), where i=0d to 51d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

RDC_MRSA*n* field descriptions

Field	Description
31–7 SADR	Start address for memory region Lower bound (inclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

3.2.5.8 Memory Region End Address (RDC_MREA*n*)

NOTE

The DDR space is 33-bit width. The RDC memory region registers are 32-bit width. The RDC configuration is the most significant bits in the DDR address space (32:1). To set the start address for this configuration, the MRSA value should be shifted 1-bit and added to the DDR base address. The example below illustrates how to calculate the proper start and end address value. Please refer to the Memory Map for the actual DDR base address.

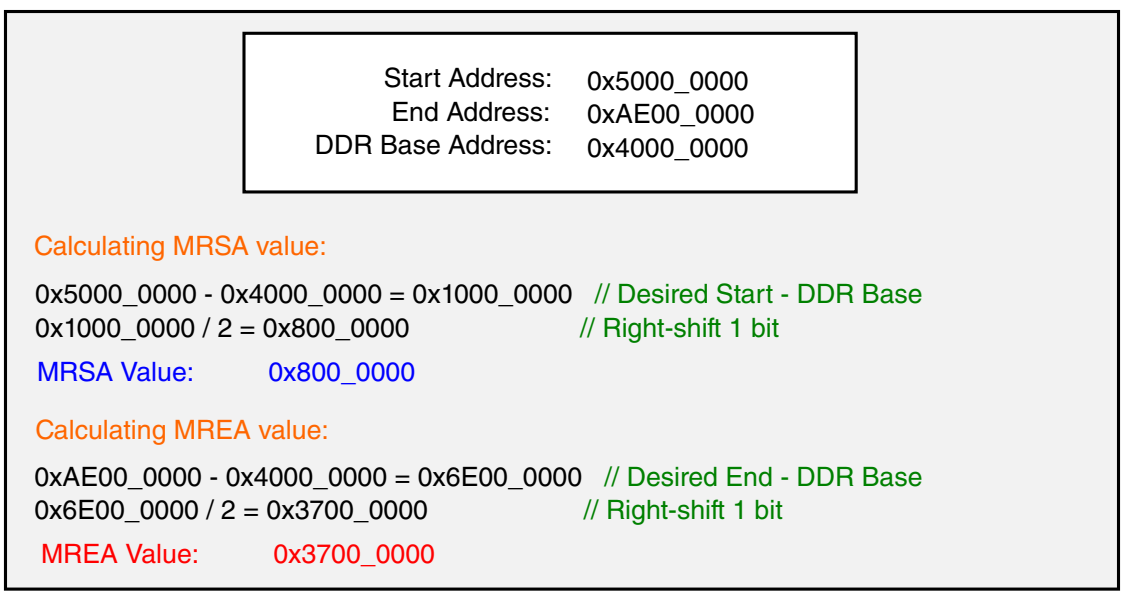


Figure 3-6. Calculating Address Value Examble

Address: 303D_0000h base + 804h offset + (16d × i), where i=0d to 51d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EADR																Reserved															
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

RDC_MREA*n* field descriptions

Field	Description
31–7 EADR	Upper bound for memory region

Table continues on the next page...

RDC_MREA*n* field descriptions (continued)

Field	Description
	Upper bound (exclusive) modulo the defined granularity byte size of a region. The region size (granularity) is defined for each Memory/Port in the Memory Region Map section. Region boundaries are aligned to the minimum possible region size for the Memory/Port.
Reserved	This field is reserved.

3.2.5.9 Memory Region Control (RDC_MRC*n*)

Address: 303D_0000h base + 808h offset + (16d × i), where i=0d to 51d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LCK	ENA	Reserved													
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								D3R	D3W	D2R	D2W	D1R	D1W	D0R	D0W
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

RDC_MRC*n* field descriptions

Field	Description
31 LCK	Region Lock Locks all region fields from further modification except ENA, which can be set but not reset after LCK is set. LCK is a sticky bit. 0 No Lock. All fields in this register may be modified. 1 Locked. No fields in this register may be modified except ENA, which may be set but not cleared.
30 ENA	Region Enable Activates the memory region. If the region is not activated then the permissions and address boundaries have not affect and the region will be fully accessible. 0 Memory region is not defined or restricted. 1 Memory boundaries, domain permissions and controls are in effect.
29–8 Reserved	This field is reserved.
7 D3R	Domain 3 Read Access to Region 0 Processing Domain 3 does not have Read access to the memory region 1 Processing Domain 3 has Read access to the memory region
6 D3W	Domain 3 Write Access to Region 0 Processing Domain 3 does not have Write access to the memory region 1 Processing Domain 3 has Read access to the memory region
5 D2R	Domain 2 Read Access to Region

Table continues on the next page...

RDC_MRC*n* field descriptions (continued)

Field	Description
	0 Processing Domain 2 does not have Read access to the memory region 1 Processing Domain 2 has Read access to the memory region
4 D2W	Domain 2 Write Access to Region 0 Processing Domain 2 does not have Write access to the memory region 1 Processing Domain 2 has Write access to the memory region
3 D1R	Domain 1 Read Access to Region 0 Processing Domain 1 does not have Read access to the memory region 1 Processing Domain 1 has Read access to the memory region
2 D1W	Domain 1 Write Access to Region 0 Processing Domain 1 does not have Write access to the memory region 1 Processing Domain 1 has Write access to the memory region
1 D0R	Domain 0 Read Access to Region 0 Processing Domain 0 does not have Read access to the memory region 1 Processing Domain 0 has Read access to the memory region
0 D0W	Domain 0 Write Access to Region 0 Processing Domain 0 does not have Write access to the memory region 1 Processing Domain 0 has Write access to the memory region

3.2.5.10 Memory Region Violation Status (RDC_MRVS*n*)

Address: 303D_0000h base + 80Ch offset + (16d × i), where i=0d to 51d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VADR															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VADR											AD	Reserved		VDID	
W												w1c				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_MRVS*n* field descriptions

Field	Description
31–5 VADR	Violating Address The address of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. Clearing of contents occurs only when the status is read by the memory region's associated domain ID (s).

Table continues on the next page...

RDC_MRVS*n* field descriptions (continued)

Field	Description
4 AD	Access Denied Access to a memory region denied. This bit is cleared when this bit is written by one of the allowed domains.
3–2 Reserved	This field is reserved.
VDID	Violating Domain ID The domain ID of the denied access. The first access violation is captured. Subsequent violations are ignored until the status register is cleared. Contents are cleared upon reading the register. 00 Processing Domain 0 01 Processing Domain 1 10 Processing Domain 2 11 Processing Domain 3

3.2.6 RDC SEMA42 Memory Map/Register Definition

Only Supervisor Mode accesses are allowed on these registers. User accesses generate an error termination.

RDC_SEMAPHORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_0000	Gate Register (RDC_SEMAPHORE1_GATE0)	8	R/W	00h	3.2.6.1/75
303B_0001	Gate Register (RDC_SEMAPHORE1_GATE1)	8	R/W	00h	3.2.6.1/75
303B_0002	Gate Register (RDC_SEMAPHORE1_GATE2)	8	R/W	00h	3.2.6.1/75
303B_0003	Gate Register (RDC_SEMAPHORE1_GATE3)	8	R/W	00h	3.2.6.1/75
303B_0004	Gate Register (RDC_SEMAPHORE1_GATE4)	8	R/W	00h	3.2.6.1/75
303B_0005	Gate Register (RDC_SEMAPHORE1_GATE5)	8	R/W	00h	3.2.6.1/75
303B_0006	Gate Register (RDC_SEMAPHORE1_GATE6)	8	R/W	00h	3.2.6.1/75
303B_0007	Gate Register (RDC_SEMAPHORE1_GATE7)	8	R/W	00h	3.2.6.1/75
303B_0008	Gate Register (RDC_SEMAPHORE1_GATE8)	8	R/W	00h	3.2.6.1/75
303B_0009	Gate Register (RDC_SEMAPHORE1_GATE9)	8	R/W	00h	3.2.6.1/75
303B_000A	Gate Register (RDC_SEMAPHORE1_GATE10)	8	R/W	00h	3.2.6.1/75
303B_000B	Gate Register (RDC_SEMAPHORE1_GATE11)	8	R/W	00h	3.2.6.1/75
303B_000C	Gate Register (RDC_SEMAPHORE1_GATE12)	8	R/W	00h	3.2.6.1/75
303B_000D	Gate Register (RDC_SEMAPHORE1_GATE13)	8	R/W	00h	3.2.6.1/75
303B_000E	Gate Register (RDC_SEMAPHORE1_GATE14)	8	R/W	00h	3.2.6.1/75

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_000F	Gate Register (RDC_SEMAPHORE1_GATE15)	8	R/W	00h	3.2.6.1/75
303B_0010	Gate Register (RDC_SEMAPHORE1_GATE16)	8	R/W	00h	3.2.6.1/75
303B_0011	Gate Register (RDC_SEMAPHORE1_GATE17)	8	R/W	00h	3.2.6.1/75
303B_0012	Gate Register (RDC_SEMAPHORE1_GATE18)	8	R/W	00h	3.2.6.1/75
303B_0013	Gate Register (RDC_SEMAPHORE1_GATE19)	8	R/W	00h	3.2.6.1/75
303B_0014	Gate Register (RDC_SEMAPHORE1_GATE20)	8	R/W	00h	3.2.6.1/75
303B_0015	Gate Register (RDC_SEMAPHORE1_GATE21)	8	R/W	00h	3.2.6.1/75
303B_0016	Gate Register (RDC_SEMAPHORE1_GATE22)	8	R/W	00h	3.2.6.1/75
303B_0017	Gate Register (RDC_SEMAPHORE1_GATE23)	8	R/W	00h	3.2.6.1/75
303B_0018	Gate Register (RDC_SEMAPHORE1_GATE24)	8	R/W	00h	3.2.6.1/75
303B_0019	Gate Register (RDC_SEMAPHORE1_GATE25)	8	R/W	00h	3.2.6.1/75
303B_001A	Gate Register (RDC_SEMAPHORE1_GATE26)	8	R/W	00h	3.2.6.1/75
303B_001B	Gate Register (RDC_SEMAPHORE1_GATE27)	8	R/W	00h	3.2.6.1/75
303B_001C	Gate Register (RDC_SEMAPHORE1_GATE28)	8	R/W	00h	3.2.6.1/75
303B_001D	Gate Register (RDC_SEMAPHORE1_GATE29)	8	R/W	00h	3.2.6.1/75
303B_001E	Gate Register (RDC_SEMAPHORE1_GATE30)	8	R/W	00h	3.2.6.1/75
303B_001F	Gate Register (RDC_SEMAPHORE1_GATE31)	8	R/W	00h	3.2.6.1/75
303B_0020	Gate Register (RDC_SEMAPHORE1_GATE32)	8	R/W	00h	3.2.6.1/75
303B_0021	Gate Register (RDC_SEMAPHORE1_GATE33)	8	R/W	00h	3.2.6.1/75
303B_0022	Gate Register (RDC_SEMAPHORE1_GATE34)	8	R/W	00h	3.2.6.1/75
303B_0023	Gate Register (RDC_SEMAPHORE1_GATE35)	8	R/W	00h	3.2.6.1/75
303B_0024	Gate Register (RDC_SEMAPHORE1_GATE36)	8	R/W	00h	3.2.6.1/75
303B_0025	Gate Register (RDC_SEMAPHORE1_GATE37)	8	R/W	00h	3.2.6.1/75
303B_0026	Gate Register (RDC_SEMAPHORE1_GATE38)	8	R/W	00h	3.2.6.1/75
303B_0027	Gate Register (RDC_SEMAPHORE1_GATE39)	8	R/W	00h	3.2.6.1/75
303B_0028	Gate Register (RDC_SEMAPHORE1_GATE40)	8	R/W	00h	3.2.6.1/75
303B_0029	Gate Register (RDC_SEMAPHORE1_GATE41)	8	R/W	00h	3.2.6.1/75
303B_002A	Gate Register (RDC_SEMAPHORE1_GATE42)	8	R/W	00h	3.2.6.1/75
303B_002B	Gate Register (RDC_SEMAPHORE1_GATE43)	8	R/W	00h	3.2.6.1/75
303B_002C	Gate Register (RDC_SEMAPHORE1_GATE44)	8	R/W	00h	3.2.6.1/75
303B_002D	Gate Register (RDC_SEMAPHORE1_GATE45)	8	R/W	00h	3.2.6.1/75
303B_002E	Gate Register (RDC_SEMAPHORE1_GATE46)	8	R/W	00h	3.2.6.1/75
303B_002F	Gate Register (RDC_SEMAPHORE1_GATE47)	8	R/W	00h	3.2.6.1/75
303B_0030	Gate Register (RDC_SEMAPHORE1_GATE48)	8	R/W	00h	3.2.6.1/75
303B_0031	Gate Register (RDC_SEMAPHORE1_GATE49)	8	R/W	00h	3.2.6.1/75
303B_0032	Gate Register (RDC_SEMAPHORE1_GATE50)	8	R/W	00h	3.2.6.1/75
303B_0033	Gate Register (RDC_SEMAPHORE1_GATE51)	8	R/W	00h	3.2.6.1/75
303B_0034	Gate Register (RDC_SEMAPHORE1_GATE52)	8	R/W	00h	3.2.6.1/75

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303B_0035	Gate Register (RDC_SEMAPHORE1_GATE53)	8	R/W	00h	3.2.6.1/75
303B_0036	Gate Register (RDC_SEMAPHORE1_GATE54)	8	R/W	00h	3.2.6.1/75
303B_0037	Gate Register (RDC_SEMAPHORE1_GATE55)	8	R/W	00h	3.2.6.1/75
303B_0038	Gate Register (RDC_SEMAPHORE1_GATE56)	8	R/W	00h	3.2.6.1/75
303B_0039	Gate Register (RDC_SEMAPHORE1_GATE57)	8	R/W	00h	3.2.6.1/75
303B_003A	Gate Register (RDC_SEMAPHORE1_GATE58)	8	R/W	00h	3.2.6.1/75
303B_003B	Gate Register (RDC_SEMAPHORE1_GATE59)	8	R/W	00h	3.2.6.1/75
303B_003C	Gate Register (RDC_SEMAPHORE1_GATE60)	8	R/W	00h	3.2.6.1/75
303B_003D	Gate Register (RDC_SEMAPHORE1_GATE61)	8	R/W	00h	3.2.6.1/75
303B_003E	Gate Register (RDC_SEMAPHORE1_GATE62)	8	R/W	00h	3.2.6.1/75
303B_003F	Gate Register (RDC_SEMAPHORE1_GATE63)	8	R/W	00h	3.2.6.1/75
303B_0040	Reset Gate Write (RDC_SEMAPHORE1_RSTGT_W)	16	R/W	0000h	3.2.6.2/76
303B_0040	Reset Gate Read (RDC_SEMAPHORE1_RSTGT_R)	16	R/W	0000h	3.2.6.3/77
303C_0000	Gate Register (RDC_SEMAPHORE2_GATE0)	8	R/W	00h	3.2.6.1/75
303C_0001	Gate Register (RDC_SEMAPHORE2_GATE1)	8	R/W	00h	3.2.6.1/75
303C_0002	Gate Register (RDC_SEMAPHORE2_GATE2)	8	R/W	00h	3.2.6.1/75
303C_0003	Gate Register (RDC_SEMAPHORE2_GATE3)	8	R/W	00h	3.2.6.1/75
303C_0004	Gate Register (RDC_SEMAPHORE2_GATE4)	8	R/W	00h	3.2.6.1/75
303C_0005	Gate Register (RDC_SEMAPHORE2_GATE5)	8	R/W	00h	3.2.6.1/75
303C_0006	Gate Register (RDC_SEMAPHORE2_GATE6)	8	R/W	00h	3.2.6.1/75
303C_0007	Gate Register (RDC_SEMAPHORE2_GATE7)	8	R/W	00h	3.2.6.1/75
303C_0008	Gate Register (RDC_SEMAPHORE2_GATE8)	8	R/W	00h	3.2.6.1/75
303C_0009	Gate Register (RDC_SEMAPHORE2_GATE9)	8	R/W	00h	3.2.6.1/75
303C_000A	Gate Register (RDC_SEMAPHORE2_GATE10)	8	R/W	00h	3.2.6.1/75
303C_000B	Gate Register (RDC_SEMAPHORE2_GATE11)	8	R/W	00h	3.2.6.1/75
303C_000C	Gate Register (RDC_SEMAPHORE2_GATE12)	8	R/W	00h	3.2.6.1/75
303C_000D	Gate Register (RDC_SEMAPHORE2_GATE13)	8	R/W	00h	3.2.6.1/75
303C_000E	Gate Register (RDC_SEMAPHORE2_GATE14)	8	R/W	00h	3.2.6.1/75
303C_000F	Gate Register (RDC_SEMAPHORE2_GATE15)	8	R/W	00h	3.2.6.1/75
303C_0010	Gate Register (RDC_SEMAPHORE2_GATE16)	8	R/W	00h	3.2.6.1/75
303C_0011	Gate Register (RDC_SEMAPHORE2_GATE17)	8	R/W	00h	3.2.6.1/75
303C_0012	Gate Register (RDC_SEMAPHORE2_GATE18)	8	R/W	00h	3.2.6.1/75
303C_0013	Gate Register (RDC_SEMAPHORE2_GATE19)	8	R/W	00h	3.2.6.1/75
303C_0014	Gate Register (RDC_SEMAPHORE2_GATE20)	8	R/W	00h	3.2.6.1/75
303C_0015	Gate Register (RDC_SEMAPHORE2_GATE21)	8	R/W	00h	3.2.6.1/75
303C_0016	Gate Register (RDC_SEMAPHORE2_GATE22)	8	R/W	00h	3.2.6.1/75
303C_0017	Gate Register (RDC_SEMAPHORE2_GATE23)	8	R/W	00h	3.2.6.1/75
303C_0018	Gate Register (RDC_SEMAPHORE2_GATE24)	8	R/W	00h	3.2.6.1/75

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303C_0019	Gate Register (RDC_SEMAPHORE2_GATE25)	8	R/W	00h	3.2.6.1/75
303C_001A	Gate Register (RDC_SEMAPHORE2_GATE26)	8	R/W	00h	3.2.6.1/75
303C_001B	Gate Register (RDC_SEMAPHORE2_GATE27)	8	R/W	00h	3.2.6.1/75
303C_001C	Gate Register (RDC_SEMAPHORE2_GATE28)	8	R/W	00h	3.2.6.1/75
303C_001D	Gate Register (RDC_SEMAPHORE2_GATE29)	8	R/W	00h	3.2.6.1/75
303C_001E	Gate Register (RDC_SEMAPHORE2_GATE30)	8	R/W	00h	3.2.6.1/75
303C_001F	Gate Register (RDC_SEMAPHORE2_GATE31)	8	R/W	00h	3.2.6.1/75
303C_0020	Gate Register (RDC_SEMAPHORE2_GATE32)	8	R/W	00h	3.2.6.1/75
303C_0021	Gate Register (RDC_SEMAPHORE2_GATE33)	8	R/W	00h	3.2.6.1/75
303C_0022	Gate Register (RDC_SEMAPHORE2_GATE34)	8	R/W	00h	3.2.6.1/75
303C_0023	Gate Register (RDC_SEMAPHORE2_GATE35)	8	R/W	00h	3.2.6.1/75
303C_0024	Gate Register (RDC_SEMAPHORE2_GATE36)	8	R/W	00h	3.2.6.1/75
303C_0025	Gate Register (RDC_SEMAPHORE2_GATE37)	8	R/W	00h	3.2.6.1/75
303C_0026	Gate Register (RDC_SEMAPHORE2_GATE38)	8	R/W	00h	3.2.6.1/75
303C_0027	Gate Register (RDC_SEMAPHORE2_GATE39)	8	R/W	00h	3.2.6.1/75
303C_0028	Gate Register (RDC_SEMAPHORE2_GATE40)	8	R/W	00h	3.2.6.1/75
303C_0029	Gate Register (RDC_SEMAPHORE2_GATE41)	8	R/W	00h	3.2.6.1/75
303C_002A	Gate Register (RDC_SEMAPHORE2_GATE42)	8	R/W	00h	3.2.6.1/75
303C_002B	Gate Register (RDC_SEMAPHORE2_GATE43)	8	R/W	00h	3.2.6.1/75
303C_002C	Gate Register (RDC_SEMAPHORE2_GATE44)	8	R/W	00h	3.2.6.1/75
303C_002D	Gate Register (RDC_SEMAPHORE2_GATE45)	8	R/W	00h	3.2.6.1/75
303C_002E	Gate Register (RDC_SEMAPHORE2_GATE46)	8	R/W	00h	3.2.6.1/75
303C_002F	Gate Register (RDC_SEMAPHORE2_GATE47)	8	R/W	00h	3.2.6.1/75
303C_0030	Gate Register (RDC_SEMAPHORE2_GATE48)	8	R/W	00h	3.2.6.1/75
303C_0031	Gate Register (RDC_SEMAPHORE2_GATE49)	8	R/W	00h	3.2.6.1/75
303C_0032	Gate Register (RDC_SEMAPHORE2_GATE50)	8	R/W	00h	3.2.6.1/75
303C_0033	Gate Register (RDC_SEMAPHORE2_GATE51)	8	R/W	00h	3.2.6.1/75
303C_0034	Gate Register (RDC_SEMAPHORE2_GATE52)	8	R/W	00h	3.2.6.1/75
303C_0035	Gate Register (RDC_SEMAPHORE2_GATE53)	8	R/W	00h	3.2.6.1/75
303C_0036	Gate Register (RDC_SEMAPHORE2_GATE54)	8	R/W	00h	3.2.6.1/75
303C_0037	Gate Register (RDC_SEMAPHORE2_GATE55)	8	R/W	00h	3.2.6.1/75
303C_0038	Gate Register (RDC_SEMAPHORE2_GATE56)	8	R/W	00h	3.2.6.1/75
303C_0039	Gate Register (RDC_SEMAPHORE2_GATE57)	8	R/W	00h	3.2.6.1/75
303C_003A	Gate Register (RDC_SEMAPHORE2_GATE58)	8	R/W	00h	3.2.6.1/75
303C_003B	Gate Register (RDC_SEMAPHORE2_GATE59)	8	R/W	00h	3.2.6.1/75
303C_003C	Gate Register (RDC_SEMAPHORE2_GATE60)	8	R/W	00h	3.2.6.1/75
303C_003D	Gate Register (RDC_SEMAPHORE2_GATE61)	8	R/W	00h	3.2.6.1/75
303C_003E	Gate Register (RDC_SEMAPHORE2_GATE62)	8	R/W	00h	3.2.6.1/75

Table continues on the next page...

RDC_SEMAPHORE memory map (continued)

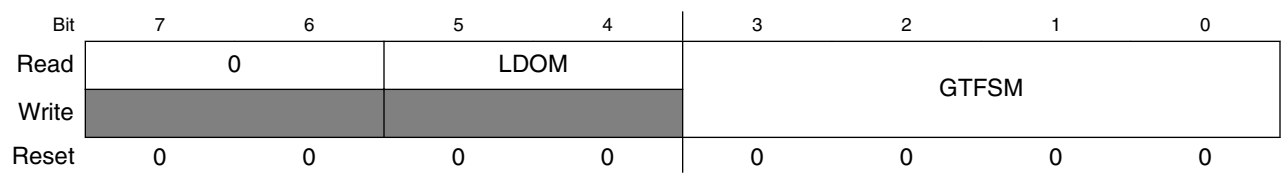
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303C_003F	Gate Register (RDC_SEMAPHORE2_GATE63)	8	R/W	00h	3.2.6.1/75
303C_0040	Reset Gate Write (RDC_SEMAPHORE2_RSTGT_W)	16	R/W	0000h	3.2.6.2/76
303C_0040	Reset Gate Read (RDC_SEMAPHORE2_RSTGT_R)	16	R/W	0000h	3.2.6.3/77

3.2.6.1 Gate Register (RDC_SEMAPHORE*x*_GATE*n*)

Each semaphore gate is implemented in a 4-bit finite state machine, right-justified in a byte data structure. The hardware uses the logical bus master number (master_index) in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. Attempted writes with a data value that is neither the unlock value nor the appropriate lock value (master_index + 1) are simply treated as "no operation" and do not affect any gate state. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes. Processor dex values can be found in AIPSTZ Memory Map/Register Definition.

Address: Base address + 0h offset + (1d × i), where i=0d to 63d



RDC_SEMAPHORE*x*_GATE*n* field descriptions

Field	Description
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 LDOM	Read-only bits. They indicate which domain had currently locked the gate. 00 The gate is locked by domain 0. (True if bits [3:0] do not equal 0000.) 01 The gate has been locked by domain 1. 10 The gate has been locked by domain 2. 11 The gate has been locked by domain 3.
GTFSM	Gate Finite State Machine. The state of the gate reflects the last processor that locked it, which can be useful during system debug.

Table continues on the next page...

RDC_SEMAPHORE*x*_GATE*n* field descriptions (continued)

Field	Description
	The hardware gate is maintained in a 16-state implementation, defined as: 0000 The gate is unlocked (free). 0001 The gate has been locked by processor with master_index = 0. 0010 The gate has been locked by processor with master_index = 1. 0011 The gate has been locked by processor with master_index = 2. 0100 The gate has been locked by processor with master_index = 3. 0101 The gate has been locked by processor with master_index = 4. 0110 The gate has been locked by processor with master_index = 5. 0111 The gate has been locked by processor with master_index = 6. 1000 The gate has been locked by processor with master_index = 7. 1001 The gate has been locked by processor with master_index = 8. 1010 The gate has been locked by processor with master_index = 9. 1011 The gate has been locked by processor with master_index = 10. 1100 The gate has been locked by processor with master_index = 11. 1101 The gate has been locked by processor with master_index = 12. 1110 The gate has been locked by processor with master_index = 13. 1111 The gate has been locked by processor with master_index = 14.

3.2.6.2 Reset Gate Write (RDC_SEMAPHORE*x*_RSTGT_W)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte (RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second

access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.

3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 40h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0							
Write									RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_W field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
RSTGDP	Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xE2 while the second write requires RSTGDP = 0x1D.

3.2.6.3 Reset Gate Read (RDC_SEMAPHOREx_RSTGT_R)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the RDC Semaphores module implements a "secure" reset mechanism that allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the RDC_SEMA42RSTGT memory location. The least significant byte (RDC_SEMA42RSTGT[RSTGDP]) must be 0xE2; the most significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the RDC_SEMA42RSTGT location. For this write, the lower byte

(RDC_SEMA42RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1D) and the upper byte (RDC_SEMA42RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or else that all gates are to be cleared. If the same processor writes incorrect data on the second access or another processor performs the second write access, the special gate reset sequence is aborted and no error signal will be asserted.

3. Reads of the RDC_SEMA42RSTGT location return information on the 2-bit state machine (RDC_SEMA42RSTGT[RSTGSM]) that implements this function, the bus master performing the reset (RDC_SEMA42RSTGT[RSTGMS]), and the gate number(s) last cleared (RDC_SEMA42RSTGT[RSTGTN]). Reads of the RDC_SEMA42RSTGT register do not affect the secure reset finite state machine in any manner.

Address: Base address + 40h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	RSTGTN								0		RSTGSM		RSTGMS			
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

RDC_SEMAPHOREx_RSTGT_R field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 RSTGSM	Reset Gate Finite State Machine. Reads of the RDC_SEMA42RSTGT register return the encoded state machine value. Note the RSTGSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write. 10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. The "01" state persists for only one clock cycle. Software will never be able to observe this state. 11 This state encoding is never used and therefore reserved.
RSTGMS	Reset Gate Bus Master. This 4-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register must be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device, and the logical processor number is SoC-specific. Consult the device reference manual for this information.

Chapter 4
ARM Platform and Debug

4.1 ARM Cortex A53 Platform (A53)

4.1.1 Overview

The Cortex-A53 cluster is a mid-range, low-power processor that implements the ARMv8-A architecture. The Cortex-A53 cluster has four cores, each with an L1 memory system and a single shared L2 cache that has a set of additional functions, which are included in a single APR region.

The core supports debug through real-time trace via ETM, and static debug via JTAG. The core platform supports static debug through the debug logic to SoC. This includes the capability of real-time trace via ARM's CoreSight ETM modules. The CTI and CTM modules allow cross-triggering of internal and external trigger sources.

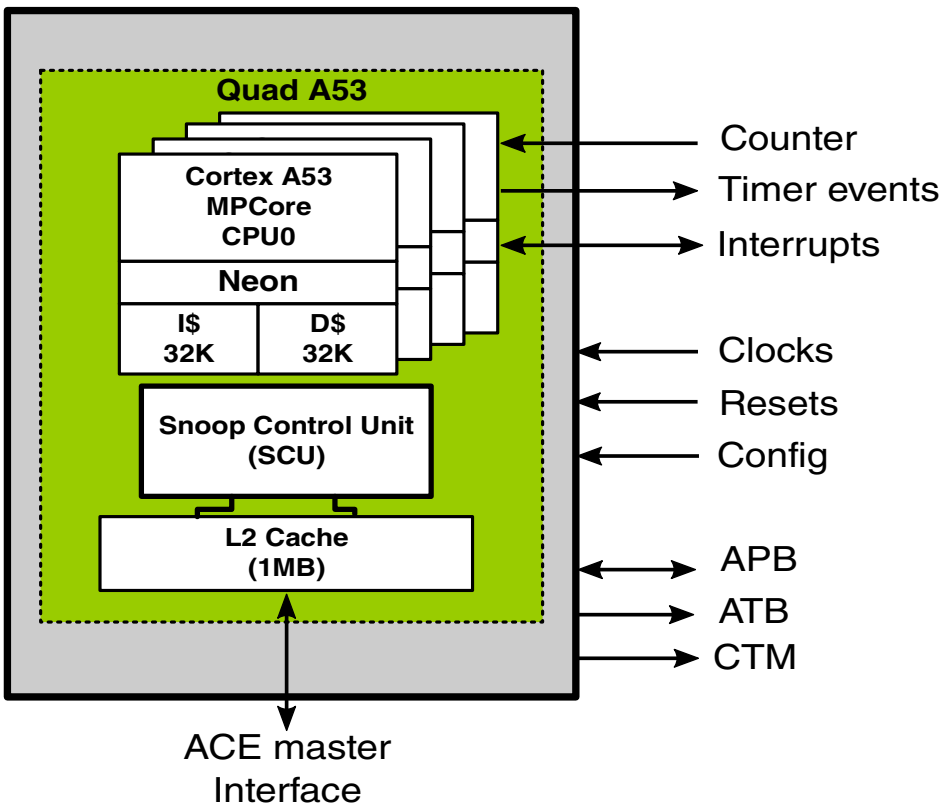


Figure 4-1. Cortex-A53 Block Diagram

4.1.1.1 Features

- 4x cores
- L1 instruction cache 32K with parity
- L1 data cache 32K with SECDED
- Advanced SIMD (NEON) per core
- Crypto extension per core
- CPU cache protection per core
- AMBA 4 ACE interface
- No ACP is included
- utilizes the ARMv8 debug map
- 1MB of L2 Cache
  - 1M with SECDED
  - Input latency 2 cycles
  - Output latency 2 cycles
  - SCU-L2 cache protection

4.1.2 Configuration

The configuration of the bus, cores and memory are detailed in the features and sections below. The core revision is r0p4-51rel0.

4.1.3 Performance

This section will discuss power and clocking performance of the Cortex-A53 Platform.

4.1.3.1 Power

There are several power states supported by the A53 Core Complex. Supported primary states are listed below:

- **Run** – At least one of 4 cores is running. The other cores may either be running, clock gated, or powered down.
- **L2 only coherent** – The L2 is powered up and servicing snoops. The cores are stopped (either powered down or clock gated). In this state the cache is retained coherent to the system.
- **L2 only non-coherent** – Both cores are stopped and powered down, the logic in the L2 controller are retaining the arrays only. In this state the L2 is not coherent, and as a result, the other AP cores must also be stopped.
- **Cluster Off** – The L2 has been flushed from the L2 only coherent state, using the HW flush mechanism (no core required). Then the cluster is fully powered off including the L2 arrays.

The power supply for the cluster is separated into two regions. The first is the control domain and second is for the remainder of the cluster (core and cluster domains). These regions are listed below:

- **Control domain** – The AON control domain contains the controls for powering up and down the rest of the core. The control domain is always powered on first and powered off last.
- **Core domains** – The core domains contains the whole core. The supply for the core domains is the same as the cluster domain, but require separate power down switches.
- **Cluster domain** – The cluster domain contains the rest of the cluster outside of the cores, which includes the shared logic of the L2 memory. The supply for the cluster power domain is the same as the core domains, but does not require power switches as it's shut down externally.

Isolation cells are required between each of these domains and is controlled by the signals on the control domain. These cells are necessary to force the output signal to be isolated to null values when the local power shuts down.

4.1.3.2 Clocking

The A53 platform is provided a main processor clock that supplies the component clocks to the cluster components, including CoreSight debug components. The maximum frequency targets are specified in the chip datasheet.

The cores are intended to support up to 1.5 GHz dependent on forward bias within the operating temperature range. Please see the datasheet for more information. The clocks are described in the table below:

Table 4-1. A53 Clocks

Clock Signal	Clock Name	Frequency	Description
Main Clock	CLKIN	Target	Main input clock.
APB Clock	PCLKENDBG	CLKIN/4	APB clock controls the timing on the debug port. Fixed frequency ratio to main frequency.
ACE Bus Clock	ACLKENM	CLKIN/2	ACE Interface bus clock. Controls the timing on the interface to CCI, but is not synchronous to CCI. Frequency is fixed ratio.
ATB Clock	ATCLKEN	CLKIN/4	ATB clock provides the clocks or outgoing trace. This clock needs to be reasonably high to enable sending samples out, but also needs to be the same as CNTCLKEN. Link to same signal as CNTCLKEN (1:4 core frequency fixed ratio).
Input Counter Clock	CNTCLKEN	CLKIN/4	Input counter clock. Timing is identical to debug port. The frequency is a fixed ratio (1:4) with the core clock. The same signal may be used for both inputs.

4.1.4 Platform sub-blocks

The sections below discuss the high-level overview of the ARM® Cortex®-A53 Platform components.

4.1.4.1 ARM Cortex-A53 MPCore Processor

The information presented in this section focuses on the design aspects of the ARM Cortex-A53 MPCore in the AP subsystem. The Cortex-A53 processor is a mid-range, low-power processor that implements the ARMv8-A architecture. The A53 complex has four cores, each with an L1 memory system and a single shared L2 cache.

The ARM Cortex-A53 MPCore processor consists of:

- Tightly-coupled L2 cache and an integrated Snoop Control Unit (SCU), connecting the four cores within the cluster providing cluster memory coherency, and a configurable coherent external interface supporting AMBA4 bus architecture.
- The Cortex-A53 implements the ARM Generic Interrupt Controller v3 (GICv3) architecture.

4.1.4.2 Advanced SIMD (Neon)

The Cortex-A53 processor supports the Advanced SIMD and Scalar Floating-point instructions in the A64 instruction set, and the Advanced SIMD and VFP instructions in the A32 and T32 instruction sets.

The ARMv8 architecture eliminates the concept of version numbers for Advanced SIMD and Floating-point in the AArch64 execution state.

4.1.4.3 Generic Interrupt Controller (GIC)

The Generic Interrupt Controller (GIC) supports and manages interrupts in the cluster. The GIC provides registers for managing:

- Interrupt sources
- Interrupt behavior
- Interrupt routing to one or more cores

The Cortex-A53 processor implements the GIC CPU interface as described in the Generic Interrupt Controller (GICv4) architecture. This interfaces with an external GICv3 or GICv4 interrupt distributor component within the system. The GICv4 architecture supports:

- Two security states
- Interrupt virtualization
- Software-generated Interrupts (SGIs)
- Message Based Interrupts
- System register access
- Memory-mapped register access
- Interrupt masking and prioritization

- Cluster environments
- Wake-up events in power management environments

The GIC includes interrupt grouping functionality that supports:

- Signaling interrupt groups to the target core using either the IRQ or the FIQ exception request, based on software configuration
- A unified scheme for handling the priority of Group 0 and Group 1 interrupts.

4.1.4.4 L1 Cache

The L1 instruction memory system has the following features:

- 32KB Instruction Cache
  - Instruction side cache line length of 64 bytes
  - 2-way set associative L1 Instruction cache
  - 128-bit read interface to the L2 memory system
- 32KB Data Cache
  - Data side cache line length of 64 bytes
  - 4-way set associative L1 Data cache
  - 256-bit write interface to the L2 memory system
  - 128-bit read interface to the L2 memory system
  - Read buffer that services the Data Cache Unit (DCU), the Instruction Fetch Unit (IFU) and the TLB
  - 64-bit read path from the data L1 memory system to the datapath
  - 128-bit write path from the datapath to the L1 memory system
  - Support for three outstanding data cache misses
  - Merging store buffer capability. This handles writes to:
    - Device memory
    - Normal Cacheable memory
    - Normal Non-cacheable memory
    - Data side prefetch engine

4.1.4.5 L2 Cache

The L2 cache consists of an integrated Snoop Control Unit (SCU), connecting four cores within the A53 cluster. The SCU also has duplicate copies of the L1 Data cache tags for coherency support. The L2 memory system interfaces to the external memory system through an AMBA 128-bit bus. The tightly-coupled L2 cache includes the following features:

- 1MB shared cache size
- AMBA 4 ACE Interface
- Fixed line length of 64 bytes

- Physically indexed and tagged cache
- 16-way set-associative cache structure
- No ACP support
- ECC/parity support

4.2 ARM Cortex M4 Platform (M4)

4.2.1 Overview

This block details the ARM Cortex-M4 core and platform. The Cortex-M4 implements the ARMv7- ME instruction set architecture (ISA) and adds significant capabilities with DSP and SIMD extensions. The ARM Cortex-M4 core provides additional general processing capability to the SoC with lower power and fast interrupt response time.

The Cortex-M4 also includes a single-precision floating-point unit (FPU) and two 32-bit system bus interfaces. The Cortex-M4 implementation includes two tightly coupled local memories, two cache memories connected to bus interfaces, 64-bit system bus interconnect, and supports a 32-byte cache line size.

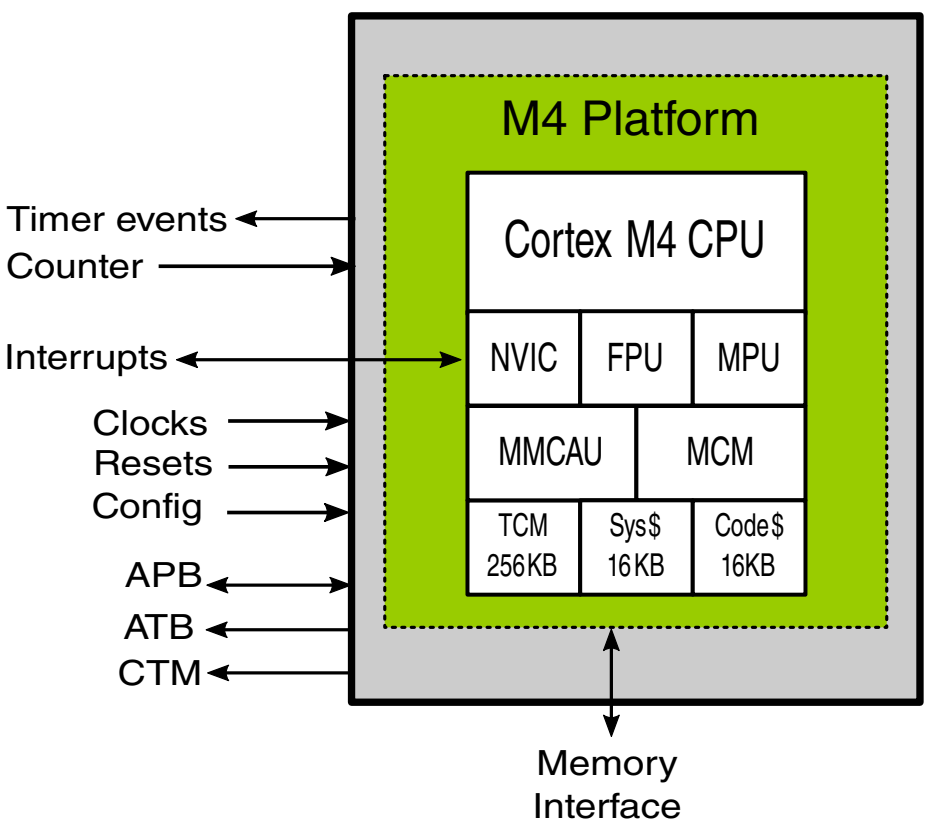


Figure 4-2. Cortex-M4 Block Diagram

4.2.1.1 Features

The features of the Cortex-M4 platform are detailed below:

- 1x Cortex-M4 processor
- AHB LMEM (Local Memory Controller) including controllers for TCM and cache memories
- 256 KB TCM (128 KB TCMU, 128 KB TCML)
- 16 KB Code Bus Cache
- 16 KB System Bus Cache
- ECC for TCM memories and parity for code and system caches
- Integrated Nested Vector Interrupt Controller (NVIC)
- Wakeup Interrupt Controller (WIC)
- FPU (Floating Point Unit)
- Core MPU (Memory Protection Unit)
- Support for exclusive access on the system bus
- MMCAU (Crypto Acceleration Unit)
- MCM (Miscellaneous Control Module)

4.2.2 Configuration

The configuration of the bus, core, and memory are detailed in the features and sections below.

Table 4-2. Cortex-M4 Configuration

Parameter	Description	Setting
NUM_IRQ	Number of Interrupts	128
LVL_WIDTH (translates to 2^n levels)	Number of interrupt priority bits	4
MPU_PRESENT	Exclude (0) or include (1) optional ARM MPU	1
DEBUG_LVL	Level of debug support	3
TRACE_LVL	Level of trace support 0 = No Trace 1 = ITM and DWT 2 = ITM, DWT, and ETM 3 = ITM, DWT, ETM, and HTM	2
RESET_ALL_REGS	Select whether all registers are reset (1)	1
JTAG_PRESENT	SWJ-DP (1) SW-DP (0)	0
CLKGATE_PRESENT	Disable (0) or enable (1) instantiation of architectural clock gates at all levels	1

Table continues on the next page...

Table 4-2. Cortex-M4 Configuration (continued)

WIC_PRESENT	Exclude (0) or include (1) optional ARM provided WIC	1
WIC_LINES	Select number of interrupts and/or events that WIC is sensitive to	NUM_IRQ+3
FPU_PRESENT	Exclude (0) or include (1) Floating Point Unit (FPU)	1
BB_PRESENT	Exclude (0) or include (1) Bit Banding logic	0
CONST_AHB_CTRL	Specifies whether the external AHB-Lite buses maintain control information during wait stated transfers	0

4.2.3 Performance

This section will discuss power and clocking performance of the Cortex-M4 Platform.

4.2.3.1 Power

There are several power states supported by the Cortex-M4. Supported primary core states are listed below:

- **Run** - Normal run state. CM4 Platform, TCM and cache memories on nominal voltage.
- **Sleep** – Wait power state. Depending on power state, CM4 Platform, TCM and cache memories can either be in nominal or low-voltage. State recovery through NVIC. NVIC clock is still running in this state.
- **Deep Sleep** – Stop power state. Depending on power state, CM4 Platform, TCM and cache memories can either be in nominal or low-voltage. State recovery through AWIC. Clocks can be completely stopped in this state.

4.2.3.2 Clocking

The M4 platform is provided a main processor clock that supplies the component clocks to the cluster components. The maximum frequency targets are specified in the chip datasheet. Please see the datasheet for more information. The clocks are described in the table below:

Table 4-3. CM4 Clocks

Clock Signal	Clock Name	Target Frequency	Description
Core gated clock	cpu_hclk	266 MHz	Cortex-M4 core clock.
Core Free-Running Clock	cpu_fclk	266 MHz	CM4 NVIC and timer clock

Table continues on the next page...

Table 4-3. CM4 Clocks (continued)

TCM Controller Clock	tcmc_hclk	266 MHz	CM4 platform TCM controller clock
Platform Clock	hclk	266 MHz	CM4 platform AXBS fabric and bus masters. Synchronous to the core clock.
Bus Clock	ipg_clk	133 MHz	Clock for bus slaves and peripherals. Synchronous to the system clock.

4.2.4 Platform sub-blocks

The sections below discuss the high-level overview of the ARM® Cortex®-M4 Platform components.

4.2.4.1 ARM Cortex-M4 Processor

The Cortex-M4 processor is a low-power processor that features low gate count, low interrupt latency, and low-cost debug. The Cortex-M4 includes floating point arithmetic functionality. The processor is intended for deeply embedded applications that require fast interrupt response features.

The ARM Cortex-M4 processor consists of:

- A processor core
- A Nested Vectored Interrupt Controller (NVIC) closely integrated with the processor core to achieve low-latency interrupt processing
- Multiple high-performance bus interfaces
- A low-cost debug solution with the optional ability to:
  - Implement breakpoints and code patches
  - Implement watchpoints, tracing, and system profiling
  - Support printf() style debugging
  - Bridge to a Trace Port Analyzer (TPA)
- Memory Protection Unit (MPU)
- Floating Point Unit (FPU)
- Miscellaneous Control Module (MCM)
- The Cortex-M4 implements the ARM the ARMv7-M architecture.

4.2.4.2 Nested Vectored Interrupt Controller (NVIC)

The M4 platform includes a Nested Vectored Interrupt Controller (NVIC) that is closely integrated with the processor core to achieve low-latency interrupt processing. The platform supports 16 priority levels for interrupts. The NVIC IPR registers will define 4 bits per IRQ.

4.2.4.3 Floating Point Unit (FPU)

The Cortex-M4 platform supports Floating Point Unit (FPU). The FPU implements the single precision variant of the ARMv7-M Floating-Point Extension (FPv4-SP), which supports single-precision add, subtract, multiply, divide, multiply and accumulate, and square root operations. FPU also provides conversions between fixed-point and floating-point data formats, and floating-point constant instructions.

The FPU provides floating-point computation functionality that is compliant with the ANSI/IEEE Std 754-2008, IEEE Standard for Binary Floating-Point Arithmetic, referred to as the IEEE 754 standard. The FPU contains 32 single-precision extension registers, which you can also access as 16 doubleword registers for load, store, and move operations.

4.2.4.4 Memory Protection Unit (MPU)

The Cortex-M4 platform supports Memory Protection Unit (MPU). The MPU enforces privilege rules, separates processes, and enforces access rules to memory, and supports the standard ARMv7 Protected Memory System Architecture model. The MPU provides full support for:

- Protection regions
- Overlapping protection regions, with ascending region priority:
  - 7 = highest priority
  - 0 = lowest priority
- Access permissions
- Exporting memory attributes to the system

MPU mismatches and permission violations invoke the programmable-priority MemManage fault handler. You can use the MPU to:

- Enforce privilege rules
- Separate processes
- Enforce access rules

4.2.4.5 Tightly-Coupled Memory (TCM)

The Cortex-M4 uses Tightly-Coupled Memory (TCM) to perform low-latency memory operations including speculative read accesses. The TCM is split between upper and lower regions with each TCM interface operating independently of each other.

The Cortex-M4 supports 32-bit ECC error detection and correction for the TCM memories. TCM ECC can indicate to the processor that an access must be retried to return the corrected data.

Because AHB-Lite does not support write data strobes when accessing AHB-Lite slaves from an AXI master, care must be taken not to generate transactions that have partial strobes. Make sure to not have unaligned accessing to TCM from an AXI master. For example, when writing data to TCM from A53, ensure every write strobe address is 64bit aligned. When the MMU is enabled, the TCM memory range must have the MT_DEVICE_NGNRNE type attribute set. This will avoid A53 sparse writes to the TCM memory region.

4.2.5 Local Memory Controller (LMEM)

The Local Memory Controller provides the Arm®Cortex-M4™ processor with tightly-coupled processor-local memories and bus paths to all slave memory spaces.

4.2.5.1 LMEM Block Diagram

The Cortex-M4 processor has a modified 32-bit Harvard bus architecture. Using a 32-bit address space, low-order addresses (0x0000_0000 through 0x1FFF_FFFF) use the Processor Code (PC) bus, and high-order addresses (0x2000_0000 through 0xFFFF_FFFF) use the Processor System (PS) bus. As the bus names imply, normal operation has code accesses on the PC bus and data accesses on the PS bus.

This device has been augmented with tightly-coupled memories for the PC and PS buses. The memories include RAMs and caches. These local memories provide zero wait state access to RAM and cacheable address spaces.

The local memory controller includes four memory controllers and their attached memories:

- SRAM lower (SRAM_L) controller via the PC bus
- SRAM upper (SRAM_U) controller via the PS bus
- Cache memory controller via the PC bus
- Cache memory controller via the PS bus

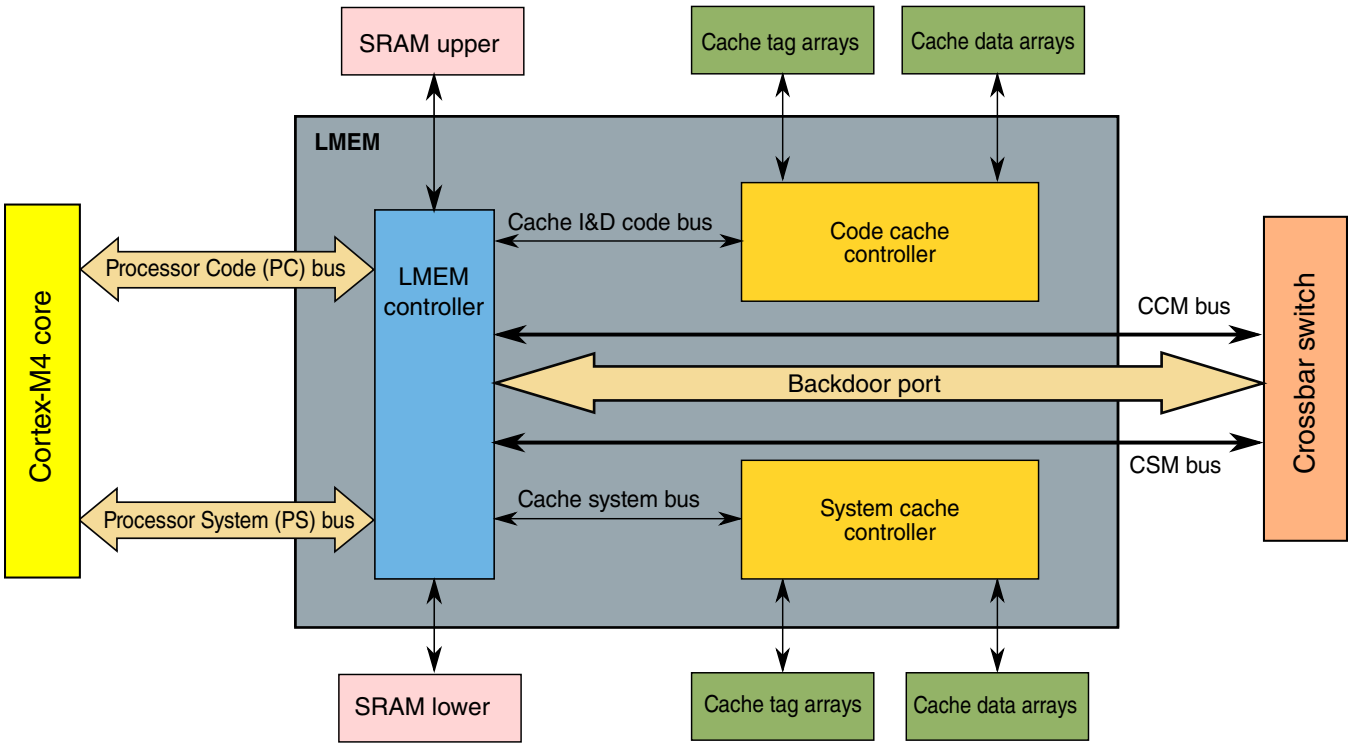


Figure 4-3. Local memory controller block diagram

NOTE

The SRAM and cache controllers reside within the LMEM, but the single-port synchronous RAM arrays used by these controllers are external.

The LMEM contains address decode logic for the PC and PS buses. This logic routes the core's accesses to the various system resources. The address spaces are device-specific and are specified in the device's Chip Configuration chapter.

4.2.5.2 Cache features

A cache is a block of high-speed memory locations containing address information (commonly known as a tag) and the associated data. The purpose is to decrease the average time of a memory access. Caches operate on two principles of locality:

- Spatial locality — An access to one location is likely to be followed by accesses from adjacent locations (for example, sequential instruction execution or usage of a data structure).
- Temporal locality — An access to an area of memory is likely to be repeated within a short time period (for example, execution of a code loop).

To minimize the quantity of control information stored, the spatial locality property is used to group several locations together under the same tag. This logical block is commonly known as a cache line.

When data is loaded into a cache, access times for subsequent loads and stores are reduced, resulting in overall performance benefits. An access to information already in a cache is known as a cache hit, and other accesses are called cache misses.

Normally, caches are self-managing, with the updates occurring automatically. Whenever the processor wants to access a cacheable location, the cache is checked. If the access is a cache hit, the access occurs immediately. Otherwise, a location is allocated and the cache line is loaded from memory. Different cache topologies and access policies are possible. However, they must comply with the memory coherency model of the underlying architecture.

Caches introduce a number of potential problems, mainly because of:

- memory accesses occurring at times other than when the programmer would normally expect them,
- the existence of multiple physical locations where a data item can be held.

The local memory controller supports three modes of operation:

1. Write-through — access to address spaces with this cache mode are cacheable.
    - A read miss on the input bus causes a line read on the output bus of a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and is marked as valid and not modified.
    - A write-through read hit to a valid cache location returns data from the cache with no output bus access.
    - A write-through write miss bypasses the cache and writes to the output bus (no allocate on write miss policy for write-through mode spaces).
    - A write-through write hit updates the cache hit data and writes to the output bus.
2. Write-back — access to address spaces with this cache mode are cacheable.
    - A write-back read miss on the input bus will cause a line read on the output bus of a 32-byte-aligned memory address containing the desired address. This miss data is loaded into the cache and marked as valid and not modified.
    - A write-back read hit to a valid cache location will return data from the cache with no output bus access.
    - A write-back write miss will do a "read-to-write" (allocate on write miss policy for write-back mode spaces). A line read on the output bus of a 16 byte aligned memory address containing the desired write address is performed. This miss data is loaded into the cache and marked as valid and modified; and the write data will then update the appropriate cache data locations.

3. Non-cacheable — access to address spaces with this cache mode are not cacheable. These accesses bypass the cache and access the output bus.

4.2.5.3 LMEM Function

The LMEM receives the following requests:

- Core master bus requests on the Processor Code (PC) bus,
- Core master bus requests on the Processor Space (PS) bus, and
- SRAM controller requests from all other bus masters on the backdoor port.

The LMEM address decode logic routes these accesses and also provides any crossbar switch slave target logic. Finally, the Local Memory controller provides the needed MPU connections for checking all SRAM controller and cacheable accesses.

The programming model for the Code and System Caches is accessed via the core's Private Peripheral Bus (PPB).

4.2.5.3.1 Processor Code accesses

Processor Code accesses are routed to the SRAM_L if they are mapped to that space. All other PC accesses are routed to the Code Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master0 port.

4.2.5.3.2 Processor Space accesses

Processor Space accesses are routed to the SRAM_U if they are mapped to that space. All other PS accesses are routed to the PS Cache Memory Controller. This controller then processes the cacheable accesses as needed, while bypassing the non-cacheable, cache write-through, cache miss, and cache maintenance accesses to the CCM bus and the crossbar switch using the Master1 port.

4.2.5.3.3 Backdoor port accesses

All LMEM backdoor port accesses are for the SRAM controller. These accesses go to the SRAM_L or the SRAM_U depending on their specific address.

4.2.5.3.4 SRAM Function

4.2.5.3.4.1 SRAM Configuration

The figure below shows how the SRAM controller is configured.

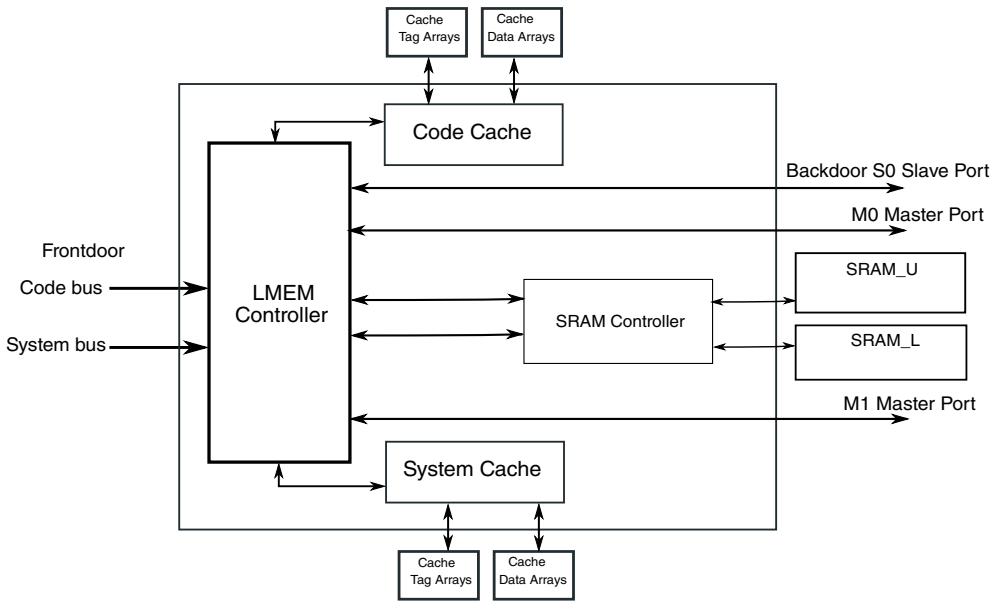


Figure 4-4. SRAM Configuration

4.2.5.3.4.2 SRAM Arrays

The on-chip SRAM is split into two logical arrays, SRAM_L and SRAM_U.

From equal-sized memories, valid address ranges for SRAM_L and SRAM_U are then defined as:

- SRAM_L = 0x1FFE_0000 - (0x1FFE_0000 + SRAM_size/2)
- SRAM_U = 0x2000_0000 - (0x2000_0000 + SRAM_size/2)

4.2.5.3.4.3 SRAM accesses

The SRAM is split into two logical arrays that are 64-bits wide:

- SRAM_L — Accessible by the code bus of the Cortex-M4 core and by the backdoor port.
- SRAM_U — Accessible by the system bus of the Cortex-M4 core and by the backdoor port.

The backdoor port makes the SRAM accessible to the non-core bus masters (such as DMA).

The figure below illustrates the SRAM accesses within the device.

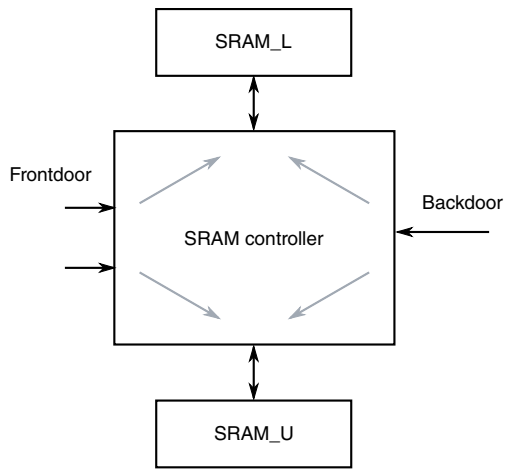


Figure 4-5. SRAM access diagram

The following simultaneous accesses can be made to different logical halves of the SRAM:

- Core code and core system
- Core code and non-core master
- Core system and non-core master

NOTE

Two non-core masters cannot access SRAM simultaneously. The required arbitration and serialization is provided by the crossbar switch. The SRAM_{L,U} arbitration is controlled by the SRAM controller based on the configuration bits in the MCM module.

4.2.5.3.5 Cache Function

The caches on this device are structured as follows. Both caches have a 2-way set-associative cache structure with a total size of 32 KBytes. The caches have a 32-bit address, 64-bit data paths and a 32-byte line size. The cache tags and data storage use single-port, synchronous RAMs.

For these 16-KByte caches, each cache TAG function uses two 256 x 22-bit RAM arrays and the cache DATA function uses two 1024 x 32-bit RAM arrays. The cache TAG entries store 20 bits of upper address as well as a modified and valid bit per cache line. The cache DATA entries store eight byes of code or data.

All normal cache accesses use physical addresses. This leads to the following cache address use:

CACHE - 16 KByte size = (256 sets) x (32-byte lines) x (2-way set associative)

TAG:

- Only address[31:29] and [21:13] are implemented. All other bits are tied to zero. Only the memories below are supported:
  - address[31] - DDR space, first 2M (0x8000_0000 - 0x801F_FFFF)
  - address[30:29] - FlexSPI channel A, first 2M (0x6000_0000 - 0x601F_FFFF)
  - address[30:29]+[21] - FlexSPI channel A, second 2M (0x6020_0000 - 603F_FFFF)
  - address[29]+[21] - OCRAM (0x2020_0000 - 0x203F_FFFF)

NOTE

To use cache, user needs to configure MPU to set those memories as cacheable and all the other memories set as non-cacheable.

DATA

- address[31:13] not used
- address[12:5] used to select one of 256 sets
- address[4:2] used to select one of eight 32-bit words within a set
- address[1:0] used to select the byte within the 32-bit word

4.2.5.3.6 Cache Control

The Code and System Caches are disabled at reset. Cache tag and data arrays are not cleared at reset. Therefore, to enable the caches, cache commands must be done to clear and initialize the required tag array bits and to configure and enable the caches.

4.2.5.3.6.1 Cache set commands

The cache set commands may operate on:

- all of way 0,
- all of way 1, or
- all of both ways (complete cache).

Cache set commands are initiated using the upper bits in the CCR register. Cache set commands perform their operation on the cache independent of the cache enable bit, CCR[ENCACHE].

A cache set command is initiated by setting the CCR[GO] bit. This bit also acts as a busy bit for set commands. It stays set while the command is active and is cleared by the hardware when the set command completes.

Supported cache set commands are given in the table below. Set commands work as follows:

- Invalidate – Unconditionally clear valid and modify bits of a cache entry.
- Push – Push a cache entry if it is valid and modified, then clear the modify bit. If entry not valid or not modified, leave as is.
- Clear – Push a cache entry if it is valid and modified, then clear the valid and modify bits. If entry not valid or not modified, clear the valid bit.

Table 4-4. Cache Set Commands

CCR[27:24]				Command
PUSHW1	INVW1	PUSHW0	INVW0	
0	0	0	0	NOP
0	0	0	1	Invalidate all way 0
0	0	1	0	Push all way 0
0	0	1	1	Clear all way 0
0	1	0	0	Invalidate all way 1
0	1	0	1	Invalidate all way 1; invalidate all way 0 (invalidate cache)
0	1	1	0	Invalidate all way 1; push all way 0
0	1	1	1	Invalidate all way 1; clear all way 0
1	0	0	0	Push all way 1
1	0	0	1	Push all way 1; invalidate all way 0
1	0	1	0	Push all way 1; push all way 0 (push cache)
1	0	1	1	Push all way 1; clear all way 0
1	1	0	0	Clear all way 1
1	1	0	1	Clear all way 1; invalidate all way 0
1	1	1	0	Clear all way 1; push all way 0
1	1	1	1	Clear all way 1; clear all way 0 (clear cache)

After a reset, complete an invalidate cache command before using the cache. It is possible to combine the cache invalidate command with the cache enable. That is, setting CCR to 0x8500_0003 will invalidate the cache and enable the cache and write buffer.

4.2.5.3.6.2 Cache line commands

Cache line commands operate on a single line in the cache at a time. Cache line commands can be performed using a physical or cache address.

- A cache address consists of a set address and a way select. The line command acts on the specified cache line.
- Cache line commands with physical addresses first search both ways of the cache set specified by bits [11:4] of the physical address. If they hit, the commands perform their action on the hit way.

Cache line commands are specified using the upper bits in the CLCR register. Cache line commands perform their operation on the cache independent of the cache enable bit (CCR[ENCACHE]). Using a cache address, the command can be completely specified using the CLCR register. Using a physical address, the command must also use the CSAR register to specify the physical address.

A line cache command is initiated by setting the line command go bit (CLCR[LGO] or CSAR[LGO]). This bit also acts a a busy bit for line commands. It stays set while the command is active and is cleared by the hardware when the command completes.

The CLCR[27:24] bits select the line command as follows:

Table 4-5. Cache Line Commands

CLCR[27:24]			Command
LACC	LADSEL	LCMD	
0	0	00	Search by cache address and way
0	0	01	Invalidate by cache address and way
0	0	10	Push by cache address and way
0	0	11	Clear by cache address and way
0	1	00	Search by physical address
0	1	01	Invalidate by physical address
0	1	10	Push by physical address
0	1	11	Clear by physical address
1	0	00	Write by cache address and way
1	0	01	Reserved, NOP
1	0	10	Reserved, NOP
1	0	11	Reserved, NOP
1	1	*xx*	Reserved, NOP

4.2.5.3.6.2.1 Executing a series of line commands using cache addresses

A series of line commands with incremental cache addresses can be performed by just writing to the CLCR.

- Place the command in CLCR[27:24],
- Set the way (CLCR[WSEL]) and tag/data (CLCR[TDSEL]) controls as needed,

- Place the cache address in CLCR[CACHEADDR], and
- Set the line command go bit (CLCR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the cache address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CLCR[LGO]).

4.2.5.3.6.2.2 Executing a series of line commands using physical addresses

Perform a series of line commands with incremental physical addresses using the following steps:

- Write to the CLCR.
  - Place the command in CLCR[27:24]
  - Set the tag/data (CLCR[TDSEL]) control
- Place the physical address in CSAR[PHYADDR] and set the line command go bit (CSAR[LGO]).

When one line command completes, initiate the next command by following these steps:

- Increment the physical address (at bit 2 to step through data or at bit 4 to step through lines), and
- Set the line command go bit (CSAR[LGO]).

The line command go bit is shared between the CLCR and CSAR registers, so that the above steps can be completed in a single write to the CSAR register.

4.2.5.3.6.2.3 Line command results

At completion of a line command, the CLCR register contains information on the initial state of the line targeted by the command. For line commands with cache addresses, this information is read before the line command action is performed from the targeted cache line. For line commands with physical addresses, this information is read on a hit before the line command action is performed from the hit cache line or has initial valid bit cleared if the command misses. In general, if the valid indicator (CLCR[LCIVB] is cleared, the targeted line was invalid at the start of the line command and no line operation was performed.

Table 4-6. Line command results

CLCR[22:20]			For cache address commands	For physical address commands
LCWAY	LCIMB	LCIVB		
0	0	0	Way 0 line was invalid	No hit
0	0	1	Way 0 valid, not modified	Way 0 valid, not modified
0	1	0	Way 0 line was invalid	No hit
0	1	1	Way 0 valid and modified	Way 0 valid and modified
1	0	0	Way 1 line was invalid	No hit
1	0	1	Way 1 valid, not modified	Way 1 valid, not modified
1	1	0	Way 1 line was invalid	No hit
1	1	1	Way 1 valid and modified	Way 1 valid and modified

At completion of a line command other than a write, the CCVR (Cache R/W Value Register) contains information on the initial state of the line tag or data targeted by the command. For line commands, CLCR[TDSEL] selects between tag and data. If the line command used a physical address and missed, the data is don't care. For write commands, the CCVR holds the write data.

4.2.6 Miscellaneous Control Module (MCM)

The Miscellaneous Control Module (MCM) provides a myriad of miscellaneous control functions.

4.2.6.1 MCM features

The MCM includes the following features:

- Program-visible information on the platform configuration and revision

4.2.6.2 MCM Interrupts

The MCM generates the following interrupt requests:

- Normal interrupt

4.2.6.2.1 Normal interrupt

The MCM's normal interrupt is generated if any of the following is true:

- ISCR[ETBI] is set, when
  - The ETB counter is enabled, ETBCC[CNTEN] = 1
  - The ETB count expires
  - The response to counter expiration is a normal interrupt, ETBCC[RSPT] = 01

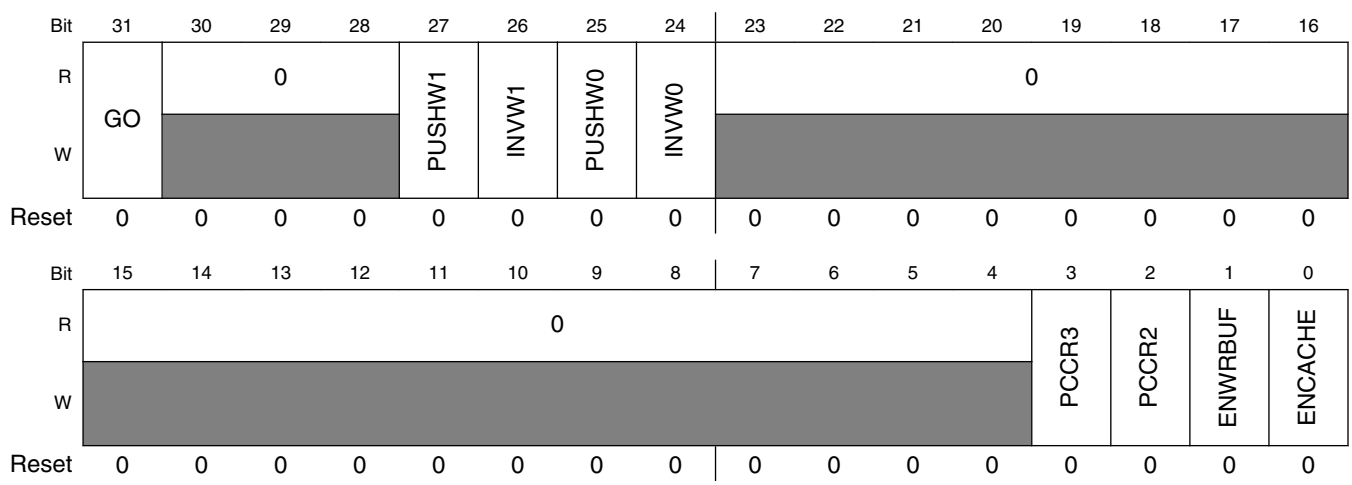
4.2.7 LMEM Memory Map/Register Definition

LMEM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_2000	Cache control register (LMEM_PCCCR)	32	R/W	0000_0000h	4.2.7.1/101
E008_2004	Cache line control register (LMEM_PCCLCR)	32	R/W	0000_0000h	4.2.7.2/103
E008_2008	Cache search address register (LMEM_PCCSAR)	32	R/W	0000_0000h	4.2.7.3/105
E008_200C	Cache read/write value register (LMEM_PCCCVR)	32	R/W	0000_0000h	4.2.7.4/106
E008_2800	Cache control register (LMEM_PSCCR)	32	R/W	0000_0000h	4.2.7.5/107
E008_2804	Cache line control register (LMEM_PSCLCR)	32	R/W	0000_0000h	4.2.7.6/108
E008_2808	Cache search address register (LMEM_PSCSAR)	32	R/W	0000_0000h	4.2.7.7/111
E008_280C	Cache read/write value register (LMEM_PSCCVR)	32	R/W	0000_0000h	4.2.7.8/112

4.2.7.1 Cache control register (LMEM_PCCCR)

Address: E008_2000h base + 0h offset = E008_2000h



LMEM_PCCCR field descriptions

Field	Description
31 GO	Initiate Cache Command

Table continues on the next page...

LMEM_PCCCR field descriptions (continued)

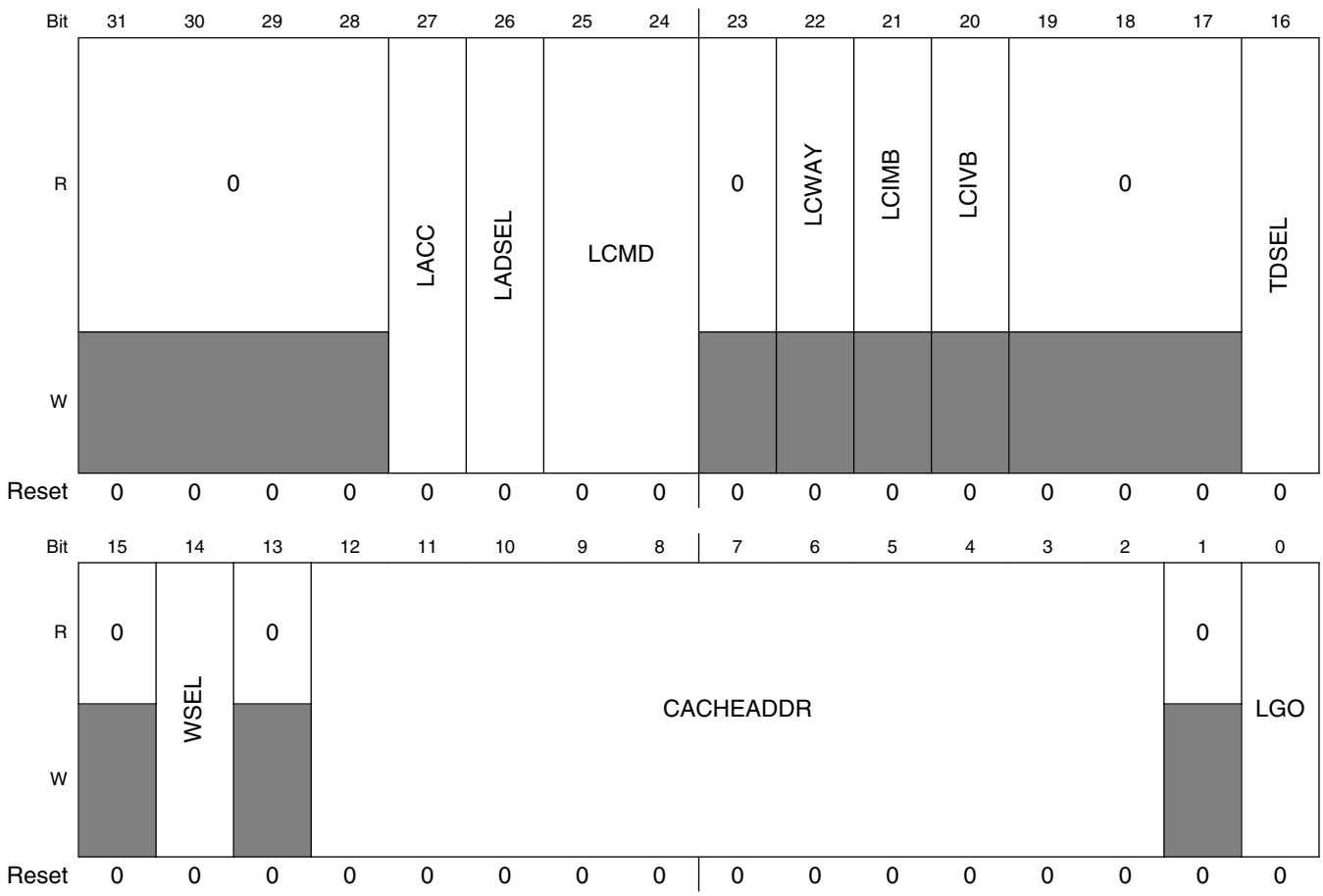
Field	Description
	Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 Write: no effect. Read: no cache command active. 1 Write: initiate command indicated by bits 27-24. Read: cache command active.
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 PUSHW1	Push Way 1 0 No operation 1 When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 No operation 1 When setting the GO bit, invalidate all lines in way 1
25 PUSHW0	Push Way 0 0 No operation 1 When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0 **NOTE:** If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 No operation 1 When setting the GO bit, invalidate all lines in way 0.
23–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 PCCR3	Forces no allocation on cache misses (must also have ACCR2 asserted)
2 PCCR2	Forces all cacheable spaces to write through
1 ENWRBUF	Enable Write Buffer 0 Write buffer disabled 1 Write buffer enabled
0 ENCACHE	Cache enable 0 Cache disabled 1 Cache enabled

4.2.7.2 Cache line control register (LMEM_PCCLCR)

This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

Address: E008_2000h base + 4h offset = E008_2004h



LMEM_PCCLCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LACC	Line access type 0 Read 1 Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL].

Table continues on the next page...

LMEM_PCCLCR field descriptions (continued)

Field	Description
	When using the physical address, both ways are searched and the command is performed only if a hit. 0 Cache address 1 Physical address
25–24 LCMD	Line Command 00 Search and read or write 01 Invalidate 10 Push 11 Clear
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LCWAY	Line Command Way Indicates the way used by the line command.
21 LCIMB	Line Command Initial Modified Bit If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 Data 1 Tag
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 WSEL	Way select Selects the way for line commands. 0 Way 0 1 Way 1
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–2 CACHEADDR	Cache address CLCR[11:4] bits are used to access the tag arrays CLCR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

LMEM_PCCLCR field descriptions (continued)

Field	Description
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits 27-24. Read: line command active.

4.2.7.3 Cache search address register (LMEM_PCCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

Address: E008_2000h base + 8h offset = E008_2008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHYADDR														0	LGO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCSAR field descriptions

Field	Description
31–2 PHYADDR	Physical Address PHYADDR represents bits [31:2] of the system address. CSAR[31:12] bits are used for tag compare CSAR[11:4] bits are used to access the tag arrays CSAR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO]

Table continues on the next page...

LMEM_PCCSAR field descriptions (continued)

Field	Description
	0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

4.2.7.4 Cache read/write value register (LMEM_PCCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

Address: E008_2000h base + Ch offset = E008_200Ch

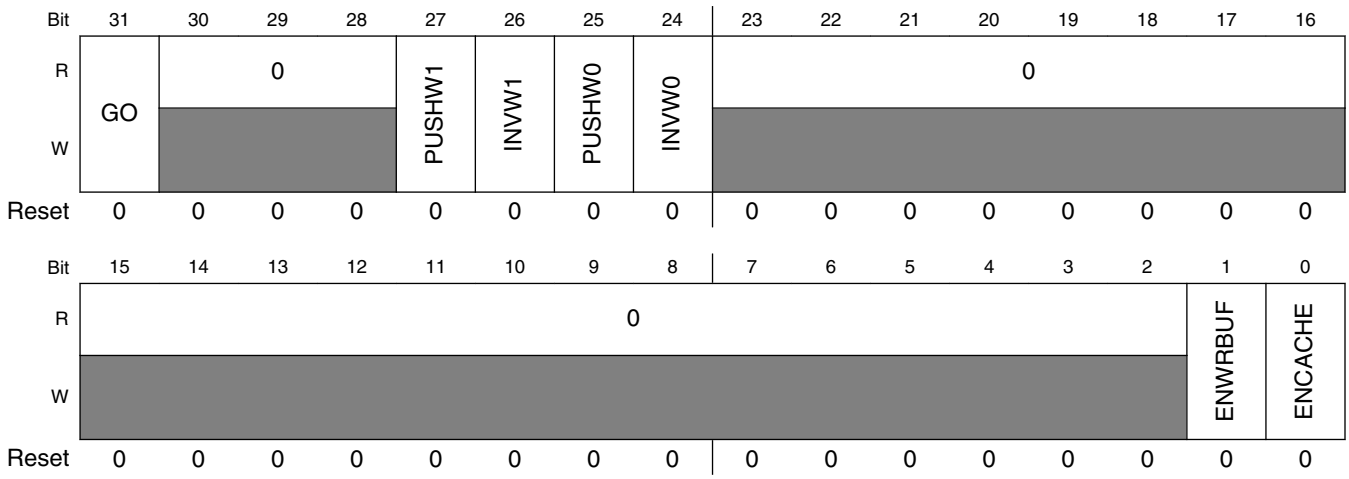
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LMEM_PCCCVR field descriptions

Field	Description
DATA	Cache read/write Data For tag search, read or write: • CCVR[31:12] bits are used for tag array R/W value • CCVR[11:4] bits are used for tag set address on reads; unused on writes • CCVR[3:2] bits are reserved For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

4.2.7.5 Cache control register (LMEM_PSCCR)

Address: E008_2000h base + 800h offset = E008_2800h



LMEM_PSCCR field descriptions

Field	Description
31 GO	Initiate Cache Command Setting this bit initiates the cache command indicated by bits 27-24. Reading this bit indicates if a command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. 0 Write: no effect. Read: no cache command active. 1 Write: initiate command indicated by bits 27-24. Read: cache command active.
30–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 PUSHW1	Push Way 1 0 No operation 1 When setting the GO bit, push all modified lines in way 1
26 INVW1	Invalidate Way 1 **NOTE:** If the PUSHW1 and INVW1 bits are set, then after setting the GO bit, push all modified lines in way 1 and invalidate all lines in way 1 (clear way 1). 0 No operation 1 When setting the GO bit, invalidate all lines in way 1
25 PUSHW0	Push Way 0 0 No operation 1 When setting the GO bit, push all modified lines in way 0
24 INVW0	Invalidate Way 0

Table continues on the next page...

LMEM_PSCCR field descriptions (continued)

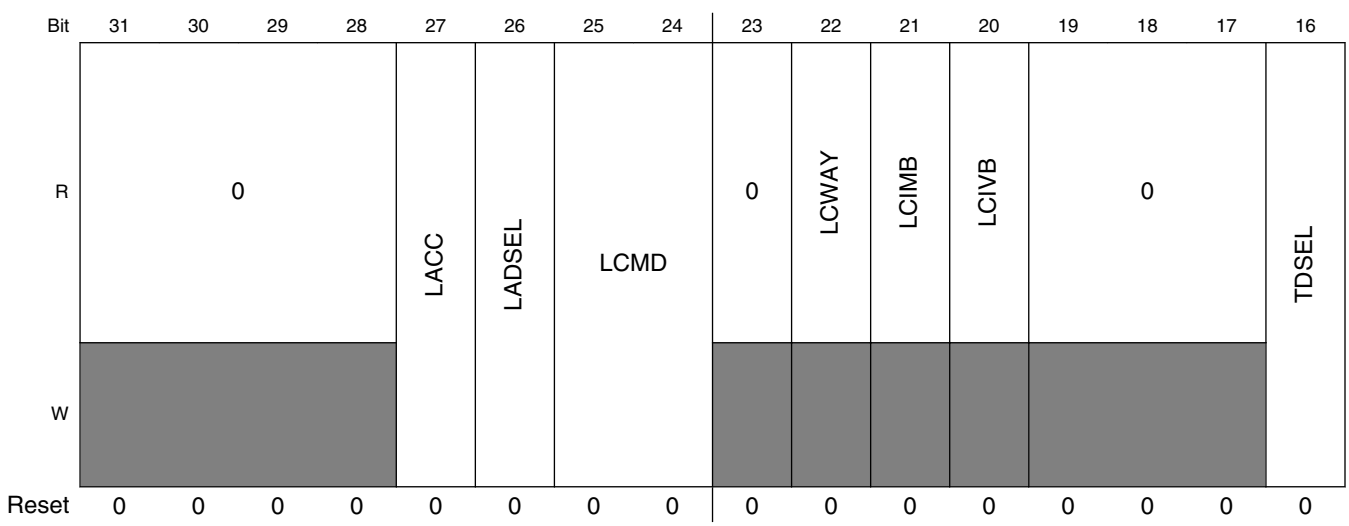
Field	Description
	NOTE: If the PUSHW0 and INVW0 bits are set, then after setting the GO bit, push all modified lines in way 0 and invalidate all lines in way 0 (clear way 0). 0 No operation 1 When setting the GO bit, invalidate all lines in way 0.
23–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 ENWRBUF	Enable Write Buffer 0 Write buffer disabled 1 Write buffer enabled
0 ENCACHE	Cache enable 0 Cache disabled 1 Cache enabled

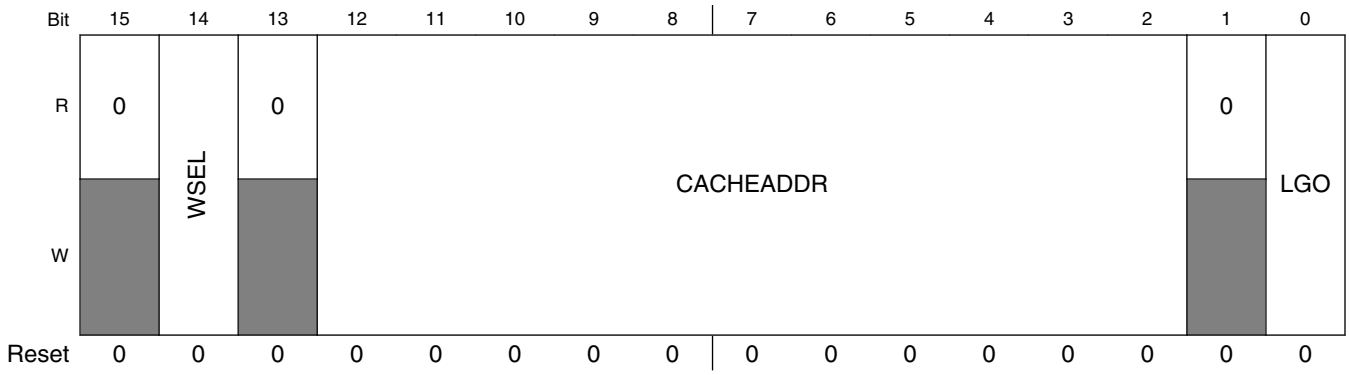
4.2.7.6 Cache line control register (LMEM_PSCLCR)

This register defines specific line-sized cache operations to be performed using a specific cache line address or a physical address.

If a physical address is specified, both ways of the cache are searched, and the command is only performed on the way which hits.

Address: E008_2000h base + 804h offset = E008_2804h





LMEM_PSCLCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27 LACC	Line access type 0 Read 1 Write
26 LADSEL	Line Address Select When using the cache address, the way must also be specified in CLCR[WSEL]. When using the physical address, both ways are searched and the command is performed only if a hit. 0 Cache address 1 Physical address
25–24 LCMD	Line Command 00 Search and read or write 01 Invalidate 10 Push 11 Clear
23 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
22 LCWAY	Line Command Way Indicates the way used by the line command.
21 LCIMB	Line Command Initial Modified Bit If command used cache address and way, then this bit shows the initial state of the modified bit If command used physical address and a hit, then this bit shows the initial state of the modified bit. If a miss, this bit reads zero.
20 LCIVB	Line Command Initial Valid Bit If command used cache address and way, then this bit shows the initial state of the valid bit If command used physical address and a hit, then this bit shows the initial state of the valid bit. If a miss, this bit reads zero.
19–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

LMEM_PSCLCR field descriptions (continued)

Field	Description
16 TDSEL	Tag/Data Select Selects tag or data for search and read or write commands. 0 Data 1 Tag
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14 WSEL	Way select Selects the way for line commands. 0 Way 0 1 Way 1
13 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
12–2 CACHEADDR	Cache address CLCR[11:4] bits are used to access the tag arrays CLCR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CSAR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits 27-24. Read: line command active.

4.2.7.7 Cache search address register (LMEM_PSCSAR)

The CSAR register is used to define the explicit cache address or the physical address for line-sized commands specified in the CLCR[LADSEL] bit.

Address: E008_2000h base + 808h offset = E008_2808h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHYADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHYADDR														0	LGO
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

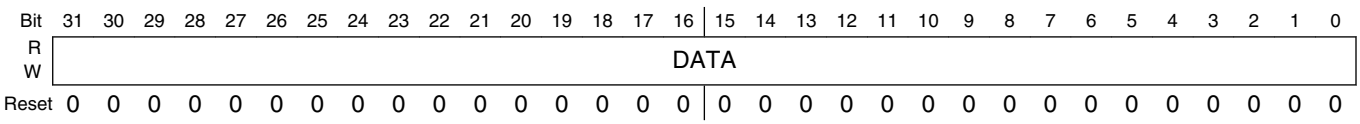
LMEM_PSCSAR field descriptions

Field	Description
31–2 PHYADDR	Physical Address PHYADDR represents bits [31:2] of the system address. CSAR[31:12] bits are used for tag compare CSAR[11:4] bits are used to access the tag arrays CSAR[11:2] bits are used to access the data arrays
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 LGO	Initiate Cache Line Command Setting this bit initiates the cache line command indicated by bits 27-24. Reading this bit indicates if a line command is active **NOTE:** This bit stays set until the command completes. Writing zero has no effect. **NOTE:** This bit is shared with CLCR[LGO] 0 Write: no effect. Read: no line command active. 1 Write: initiate line command indicated by bits CLCR[27:24]. Read: line command active.

4.2.7.8 Cache read/write value register (LMEM_PSCCVR)

The CCVR register is used to source write data or return read data for the commands specified in the CLCR register.

Address: E008_2000h base + 80Ch offset = E008_280Ch



LMEM_PSCCVR field descriptions

Field	Description
DATA	Cache read/write Data For tag search, read or write: • CCVR[31:12] bits are used for tag array R/W value • CCVR[11:4] bits are used for tag set address on reads; unused on writes • CCVR[3:2] bits are reserved For data search, read or write: • CCVR[31:0] bits are used for data array R/W value

4.2.8 MCM Memory Map/Register Definition

MCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
E008_0008	Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)	16	R	0002h	4.2.8.1/113
E008_000A	Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)	16	R	0003h	4.2.8.2/113
E008_000C	Crossbar Switch (AXBS) Control Register (MCM_PLACR)	32	R/W	0000_0000h	4.2.8.3/114
E008_0020	Fault address register (MCM_FADR)	32	R	Undefined	4.2.8.4/114
E008_0024	Fault attributes register (MCM_FATR)	32	R	Undefined	4.2.8.5/115
E008_0028	Fault data register (MCM_FDR)	32	R	Undefined	4.2.8.6/117

4.2.8.1 Crossbar Switch (AXBS) Slave Configuration (MCM_PLASC)

PLASC is a 16-bit read-only register identifying the presence/absence of bus slave connections to the device's crossbar switch.

Address: E008_0000h base + 8h offset = E008_0008h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ASC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

MCM_PLASC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
ASC	Each bit in the ASC field indicates whether there is a corresponding connection to the crossbar switch's slave input port. 0 A bus slave connection to AXBS input port *n* is absent 1 A bus slave connection to AXBS input port *n* is present

4.2.8.2 Crossbar Switch (AXBS) Master Configuration (MCM_PLAMC)

PLAMC is a 16-bit read-only register identifying the presence/absence of bus master connections to the device's crossbar switch.

Address: E008_0000h base + Ah offset = E008_000Ah

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								AMC							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

MCM_PLAMC field descriptions

Field	Description
15–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
AMC	Each bit in the AMC field indicates whether there is a corresponding connection to the AXBS master input port.

Table continues on the next page...

MCM_PLAMC field descriptions (continued)

Field	Description
	0 A bus master connection to AXBS input port *n* is absent 1 A bus master connection to AXBS input port *n* is present

4.2.8.3 Crossbar Switch (AXBS) Control Register (MCM_PLACR)

The PLACR register selects the arbitration policy for the crossbar masters.

Address: E008_0000h base + Ch offset = E008_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																Reserved															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MCM_PLACR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved.

4.2.8.4 Fault address register (MCM_FADR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting address is captured in the MCM_FADR register. The MCM logic supports capturing a single cache write buffer bus error event; if a subsequent error is detected before the captured error information has been read from the corresponding registers and the MCM_ISCR[CWBER] indicator cleared, the MCM_FATR[BEOVR] flag is set. However, no additional information is captured.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 20h offset = E008_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS																															
W																																
Reset	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*	x*

* Notes:
- x = Undefined at reset.

MCM_FADR field descriptions

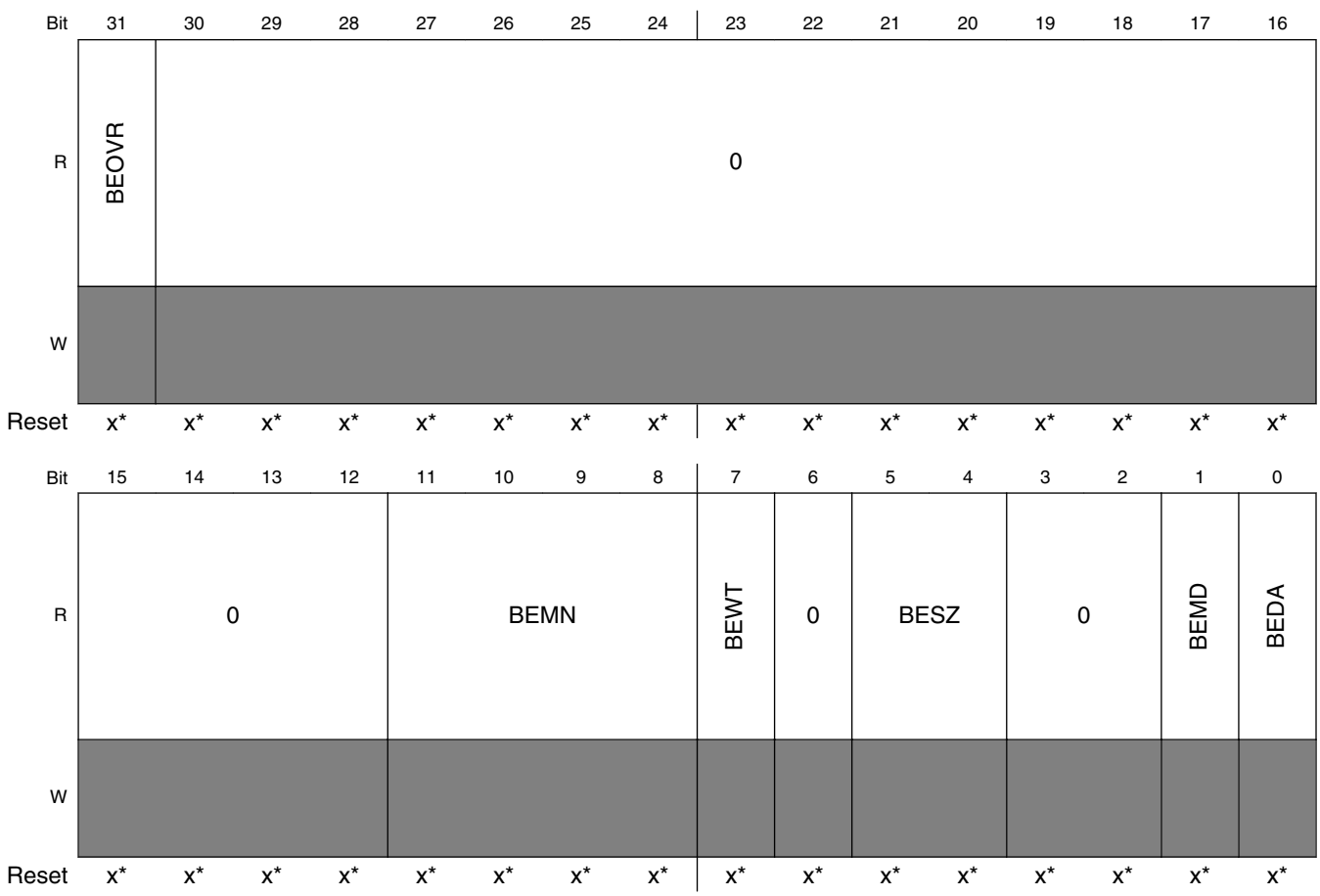
Field	Description
ADDRESS	Fault address

4.2.8.5 Fault attributes register (MCM_FATR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting attributes are captured in the MCM_FATR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 24h offset = E008_0024h



* Notes:
- x = Undefined at reset.

MCM_FATR field descriptions

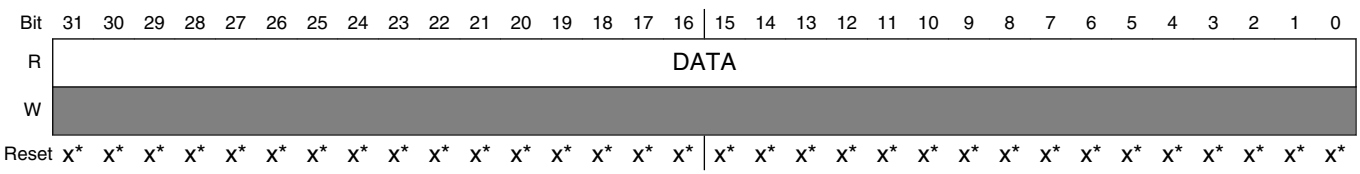
Field	Description
31 BEOVR	Bus error overrun Indicates if anothercache write buffer bus error is detected before system software has retrieved all the error information from the original event, this overrun flag is set. The window of time is defined from the detection of the original cache write buffer error termination until the MCM_ISCR[CWBER] is written with a 1 to clear it and rearm the capture logic. This bit is set by the hardware and cleared whenever software writes a 1 to the CWBER bit. 0 No bus error overrun 1 Bus error overrun occurred. The FADR and FDR registers and the other FATR bits are not updated to reflect this new bus error.
30–12 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
11–8 BEMN	Bus error master number Crossbar switch bus master number of the captured cache write buffer bus error. For this device, this value is always 0x1.
7 BEWT	Bus error write Indicates the type of system bus access when the error was detected. Since this logic is monitoring data transfers from the cache write buffer, this bit is always a logical one, signaling a write operation. 0 Read access 1 Write access
6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5–4 BESZ	Bus error size Indicates the size of the cache write buffer access when the error was detected. 00 8-bit access 01 16-bit access 10 32-bit access 11 Reserved
3–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
1 BEMD	Bus error privilege level Indicates the privilege level of the cache write buffer access when the error was detected. 0 User mode 1 Supervisor/privileged mode
0 BEDA	Bus error access type Indicates the type of cache write buffer access when the error was detected. This attribute is always a logical one signaling a data reference. 0 Instruction 1 Data

4.2.8.6 Fault data register (MCM_FDR)

When a properly-enabled cache write buffer error interrupt event is detected, the faulting data is captured in the MCM_FDR register.

The bits in this register are set by hardware and signaled by the assertion of MCM_ISCR[CWBER]. For byte and halfword writes, only the accessed byte lanes contain valid data; the contents of the other bytes are undefined. Attempted writes to this location are terminated with an error.

Address: E008_0000h base + 28h offset = E008_0028h



* Notes:
- x = Undefined at reset.

MCM_FDR field descriptions

Field	Description
DATA	Fault data

4.3 Messaging Unit (MU)

4.3.1 Overview

The Messaging Unit module enables two processors within the SoC to communicate and coordinate by passing messages (e.g. data, status and control) through the MU interface. The MU also provides the ability for one processor to signal the other processor using interrupts.

Because the MU manages the messaging between processors, the MU uses different clocks (from each side of the different peripheral buses). Therefore, the MU must synchronize the accesses from one side to the other. The MU accomplishes synchronization using two sets of matching registers (Processor A-facing, Processor B-facing).

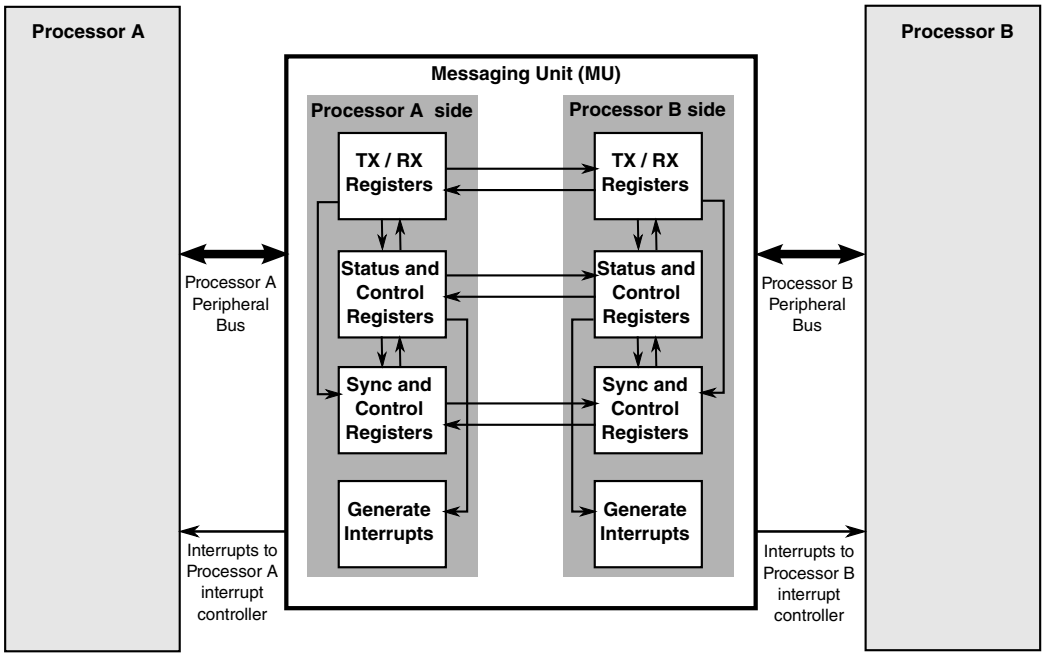


Figure 4-6. MU Block Diagram

4.3.1.1 Features

The MU includes the following features:

- Messaging control by interrupts or by polling
- The Processor B can take the Processor A out of low-power modes by asserting one of the above twelve interrupts to the Processor A and vice versa
- Symmetrical processor interfaces with each side supporting the following:
  - Four general-purpose interrupt requests reflected to the other side
  - Three general-purpose flags reflected to the other side
  - Four receive registers with maskable interrupt
  - Four transmit registers with maskable interrupt

4.3.1.2 Modes of Operation

The MU supports the modes described in the indicated sections:

- Operating Modes
- Low Power Modes

4.3.2 Functional Description

Table 4-7. Major Features of the MU

Major Feature	Description
Interprocessor Interrupts	• The MU has 12 interrupt sources on each side (Processor A-side, Processor B-side) that are used for signaling the other processor. The interrupts can be used for notification of RX/TX events and general-purpose signaling between the processors.
MU Reset	• The Processor A can issue a reset to the entire MU, using a control bit (MUR) in the Processor A Control Register (ACR). • The MUR bit is a self-clearing bit.

Processor B Boot Configuration	• The Boot Source for Processor B can be configured with the BBOOT bits in the ACR register.
	• Boot Source Options are:
	  • DMEM Base Address
	  • IMEM Base Address
	  • Address 0x00
	• The value at reset is loaded from Flash IFR

Processor B Reset Hold	• Processor B can be held in reset following any reset event. This is done by setting the BRSTH bit in the ACR register. • Processor B will be released from reset when this bit is cleared. • The value at reset is loaded from Flash IFR.
Processor A/B Clock Enable	• The Processor A/B platform clock can be enabled to continue running when Processor A/B enters Stop Mode, until Processor B/A also enters Stop Mode. This allows Processor B/A to continue accessing peripherals on Processor A/B's AIPS bus even when it has entered a Stop Mode.
Status and Control Communications between Cores	• The MU provides a way for the two cores to communicate using the status and control registers present on both the Processor B and Processor A sides of the MU. • The status register of one MU side reflects the status of the other MU side. • The control register is used for control operations, such as enabling an interrupt and sending an interrupt to the other processor.
Synchronized Message Transfers between Cores	• The transfer of data messages between cores uses transmit empty and receive full flags provided on both sides of the MU. • The update of these transmit and receive flags is accomplished using a synchronization mechanism. There is inherent latency between updating the flag on one side and reflecting its status on other side. For more about latency, see Event Update Timing
Accessing Shared Memory Directly and Avoiding Collisions	• For sending data or messages from one MU-side to the other MU-side, the MU provides 4 transmit registers and 4 receive registers on each side of the MU. • The Processor A or Processor B can access shared memory resources of the SoC directly. However, to avoid simultaneous access to shared memory by both cores, the MU provides a method (to prevent simultaneous access) using interrupts and transmit-receive registers for both processors.
Support for Different Clocks in the Two Cores	• The heart of the MU module is the event control mechanism, which synchronizes the access of one MU-side to the other MU-side, because these two MU-sides can operate using different clocks. • Formulated event update latency.
Memory-Mapped Registers	• The MU is connected as a peripheral under the Peripheral bus on both sides—on the Processor A-side, the Processor A Peripheral Bus, and on the Processor B-side, the Processor B Peripheral Bus.

4.3.2.1 Processor A Side Memory-Mapping

The messaging, control, and status registers of the Processor A-side for the MU are mapped to the Processor A memory as a regular peripheral. The Peripheral bus data bus is 32 bits wide inside the MU module.

4.3.2.2 Processor B Side Memory-Mapping

The messaging, control, and status registers of the Processor B-side for the MU are mapped to the Processor B memory as a regular peripheral. The Peripheral bus data bus is 32 bits wide inside the MU module.

4.3.2.3 MU Messaging

The MU provides 32-bit status and control registers to the Processor B and Processor A sides for control operations (such as interrupts and reset), and for status checking of the other MU-side.

For messaging, the MU has four, 32-bit write-only transmit registers and four, 32-bit read-only receive registers on the Processor B and Processor A-sides. These registers are used for sending messages to each other. These messages can be also be controlled using the 3 general purpose flags provided in the control and status registers of either MU-side.

4.3.2.3.1 Programmer Model

The messaging logic is used in conjunction with external memory. You have various messaging methods, which you can use to implement a messaging protocol. Some of these messages could mean “I have just written a message of N words, starting at offset X in the memory,” or “I have just finished reading the previous data block that was sent.” Having the messaging logic independent from the memory array does not restrict you to a predefined hardware protocol. On the other hand, the software needed to manage the messaging is short and straightforward.

Most of the messaging mechanisms are symmetric; they are duplicated and are available on both the Processor B-side and the Processor A-side. The messaging mechanisms are:

- Four, 32-bit write-only transmit registers, which are each reflected in four, read-only receive registers in the other processor’s side. You can use these registers to transfer 32-bit word messages or frame information of messages written to the shared memory (number of words, initial address, and message type code).

- A write to a transmit register on the transmitter side clears a “transmitter empty” bit in the Status Register on the transmitter side, and sets a “receiver full” bit in the Status Register on the receiver side. The setting of the bit at the receiver side can optionally trigger an interrupt at the receiver side (maskable receive interrupt).
- A read of one of the receive registers at the receiver side clears the “receiver full” bit in the Status Register at the receiver side, and sets the “transmitter empty” bit in the Status Register on the transmitter side. The setting of the “transmitter empty” bit can optionally trigger an interrupt at the transmitter side (maskable transmit interrupt).
- Four general purpose flags are reflected in the Status Register on the receiver side
- A read/write access to any reserved location and a write to a read-only register on the Processor A-side of the MU will generate a module transfer error acknowledge to the Processor A.
- A read/write access to any reserved location and write to a read-only register on the Processor B-side of the MU will generate a module transfer error acknowledge to the Processor B.

4.3.2.3.2 Messaging Examples

The following are messaging examples:

- **Passing short messages:** Transmit register(s) can be used to pass short messages from one to four words in length. For example, when a four-word message is desired, only one of the registers needs to have its corresponding interrupt enable bit set at the receiver side; the message’s first three words are written to the registers whose interrupt is masked, and the fourth word is written to the other register (which triggers an interrupt at the receiver side).
- **Passing frame information:** Transmit registers can be used to pass frame information for long messages written to the shared system (SDRAM and SyncFLASH). Such frame information normally includes a start address, number of words, and perhaps a message type code.
- **Passing event notices and requests:** Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the general interrupts, such as acknowledging that a long message was read from the shared system memory.
- **Passing fixed length data:** Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general interrupt (Processor A or Processor B) to signal the other processor that the data is ready.
- **Passing announcements:** The three flags can be used by a processor to announce its current program state or other billboard messages to the other processor.

Figure 4-7 shows the MU registers schematic.

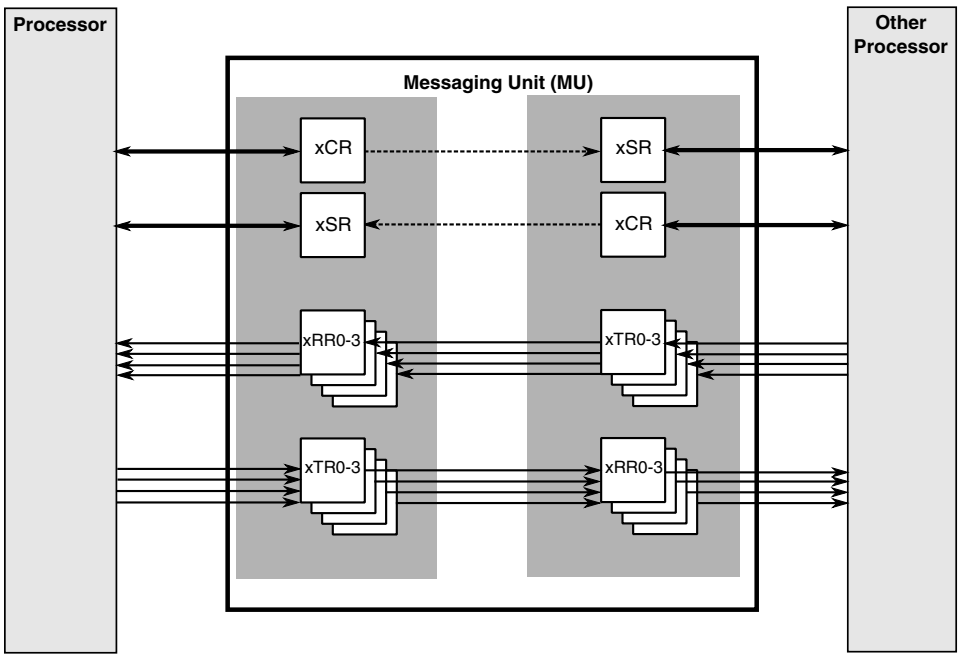


Figure 4-7. MU Registers

4.3.2.4 Operating Modes

This section describes all functional operation modes of the module.

4.3.2.5 Low Power Modes

This section describes the low power operating modes of the MU module.

4.3.2.5.1 Low Power Clocks and Synchronization

The Processor B and the Processor A clocks operate at different frequencies and from different sources. The MU design does not assume any frequency relationship between the Processor A and the Processor B clocks. Be aware, however, that the frequency relationship affects the MU's throughput performance.

- The data buffers and control logic of each MU-side operate with its corresponding clock.

4.3.2.5.2 Processor Low Power Modes

The Processors have four power modes:

- Run
- WAIT

- STOP
- DSM

The Processor can be awakened from a low-power mode by any enabled Processor side MU interrupt, as reflected in the xSR “status” register (RF0–3, TE0–3, GIP0–3 bits are set) and enabled in the xCR control register. Using these bits, the Processor can actively control when to wake the other Processor.

While the Processor is in STOP mode (such that the xSR register bits cannot be updated with events), special logic drives the enabled Processor interrupts directly from the other Processor-side (instead of from the xSR register).

While the Processor is in STOP mode, the asynchronous Processor interrupt will be asserted to wake the Processor:

- If any transmit data register of the other Processor-side is full, because of a write to it (transmit data register); that is, its “empty” bit in the xSR register is cleared while its corresponding receive interrupt is enabled on the Processor-side.
- If any receive data register of the other Processor-side is empty, because of a read on the other Processor -side; that is, its “full” bit in the xSR register is cleared while its corresponding transmit interrupt is enabled on the Processor-side.
- If any general purpose interrupt is set in the xCR register while the corresponding interrupt is enabled on the Processor-side.
- If the other Processor issues a non-maskable interrupt to the Processor.

The logic enables the other Processor to operate independently while the Processor is in any power mode (including STOP). However, the Processor power mode change protocol should be handled with care regarding:

- The interrupts that are enabled on the Processor-side
- The events that could be triggered by the other Processor-side
- The compatibility with the other Processor protocol of entering STOP mode

If the Processor is in STOP mode and an event on the other Processor is triggered, the EP bit (in the xSR register) will remain high until the Processor wakes up.

Before entering STOP mode, the Processor programmer should verify that the EP bit (in the xSR register) is cleared. This check is needed to ensure that all pending updates from the Processor, including the power mode change when STOP or WAIT is executed, will be updated in the xSR register.

- If the other Processor is in STOP mode or DSM mode, the EP bit (in the xSR register) may be stuck high; in this case, the Processor need not check the EP bit before entering STOP mode.

4.3.2.6 Event Update Timing

Each processor's MU messaging side (Processor B or Processor A) has a hardware mechanism to send "event update requests" to the other processor's side. An "event" is considered when any information change should be reflected at the Status Register of the receiving processor. The event update latency is the delay between the event being ready at one processor and the resulting update at the Status Register of the other processor.

- The minimum event latency is "1 clock of the sending side" + "2 1/2 clocks of the receiving side". The minimum case is if there is no event pending when the new event occurs.
- The maximum event latency is "6 clocks of the sending side" + "6 1/2 clocks of the receiving side." The maximum case is if the event occurred just after a previous event was sent to the other side. The event update latency will vary between the above-mentioned minimum and maximum latencies, depending on the time at which the subsequent event is triggered.

4.3.2.7 Interrupts

The MU controls the Processor B interrupt requests to the Processor A, and the Processor A interrupt requests to the Processor B. This section describes all the interrupts that the module generates.

4.3.2.7.1 Interrupts to the Processors

There are 12 interrupt sources from the MU to the Processors:

- Four receive interrupts (asserted when the Processors receive full bits are set and enabled in the xCR register) for each of the receive registers
- Four transmit interrupts (asserted when the Processor transmit empty bits are set and enabled in the xCR register) for each of the transmit registers
- Four general purpose interrupts (asserted when the GIP bits are set and enabled in the xCR register)

All the interrupts are maskable in the Processor Control Register (xCR). The MU does not assume any internal priority of these interrupts. Multiple interrupts (for example, Receive 0 and Receive 1 interrupts or any of the transmit and general purpose interrupts) can be asserted at one time. The priority of these interrupts should be resolved by the interrupt controller at the chip level.

The General Purpose Interrupt Pending bits (GIP0, GIP1, GIP2, and GIP3) should be cleared by the software (as part of the interrupt service routine) to de-assert the request to the interrupt controller.

4.3.2.7.2 General Purpose Interrupt Clearing Sequence

When a Processor writes to the general interrupt bit (GIR), the write event is synchronized to the other Processor clock to set the general interrupt request pending bit (GIP). When the GIP bit is set, and if the general purpose interrupt is enabled on the transmitting Processor side (GIE bit is set), then the receiving Processor general purpose interrupt is issued to the transmitting Processor. The transmitting Processor clears this interrupt by writing a “1” on the GIP bit. The interrupt is de-asserted as soon as the GIP bit is written. The write event of the GIP bit is synchronized to the other Processor clock. The synchronized signal clears the GIR bit. The software should not write the GIR bit again until the GIR bit is cleared.

4.3.2.8 Interrupt Messaging Protocols

4.3.2.8.1 Messaging Protocols using Interrupts

The example below describes a four-word messaging sequence sent by the Processor to the other Processor.

In this example, the first, second, and third receive interrupts are disabled, and the fourth receive interrupt is enabled. We write registers sequentially for n = 0, 1, 2, 3. For n = 0, 1, 2, the interrupts are disabled, therefore no interrupt will go to the other core (although interrupt conditions occur). For n = 3, the interrupt is enabled, and the last Receive Interrupt request is generated.

1. Write Sequence
   - The Processor writes the message information sequentially to its Transmit Registers 0, 1, 2.
   - When the write to the Transmit Register 3 occurs, the RF3 bit of the xSR is set after synchronization, and it immediately trigger the Receive 3 interrupt to the other Processor.
2. Read Sequence
   - The other Processor receives the Receive 3 interrupt and starts reading the message transferred from the receive registers.
   - After Receive Register 3 is read, the interrupt bit is cleared.

Figure 4-8 shows the programmer’s model of a messaging protocol using transmit and receive registers. Use Table 4-8 and Figure 4-8 to understand the generalized protocol sequence.

Table 4-8. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A Data write	A data write to the ATRn register by Processor A is immediately reflected in the Processor B BRRn register.
2	Clear Tx Empty bit and Set Rx Full bit	The data write to the ATRn register • Clears the transmitter empty bit (TEn) in the Processor A Transmit Status Register • Sets the receiver full bit (RFn) in the Processor B Receive Status Register
3	Generate Receive Interrupt request	The setting of the receiver full bit (RFn) in the Receive Status Register generates a Receive Interrupt request to Processor B.
4	Processor B Data read	After receiving the Receive Interrupt request, Processor B performs a data read of the BRRn register.
5	Clear Rx Full bit and Set Tx Empty bit	Reading the data out of the BRRn register • Clears the receiver full bit (RFn) in the Processor B Receive Status Register • Sets the transmitter empty bit (TEn) in the Processor A Transmit Status Register
6	Generate Transmit Interrupt request	The setting of the transmitter empty bit (TEn) in the Transmit Status Register generates a Transmit Interrupt request to Processor A.

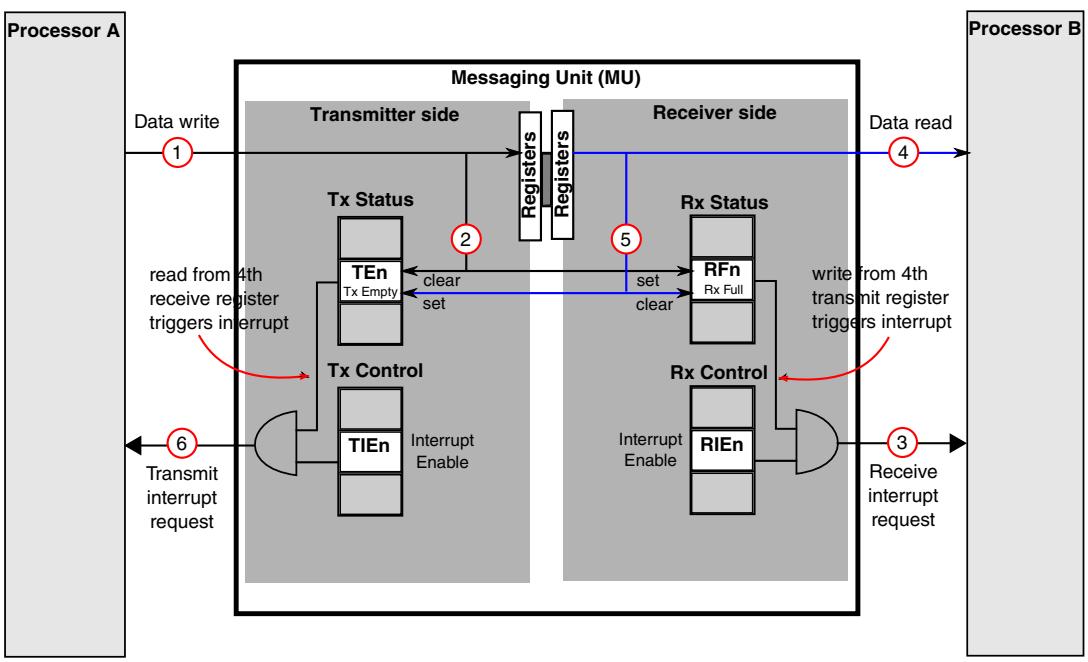


Figure 4-8. Messaging Model Using Transmit and Receive Registers

NOTE

The Transmit registers can be used to pass frame information on long messages written to the shared memory. Such frame

information would typically include an initial address, number of words, and perhaps a message type code.

The messaging hardware can be used by software to implement messaging protocols for a wide array of message types. Full support is given for both interrupt and polling management schemes.

4.3.2.8.2 Messaging Protocols using Event Interrupts

Events and requests that do not include data words can be signaled from the Processor B to the Processor A using the two general interrupts.

Formatted data with a fixed length can be written in predetermined locations in the shared memory. A processor can use a general purpose interrupt to signal the other processor that the data is ready.

The three flags can be used by a processor to announce to the other processor the program state it is currently in, or to announce similar messages.

Table 4-9 and Figure 4-9 describe the event sequence when the Processor triggers an interrupt.

Table 4-9. Interrupt Messaging Protocol (Generalized)

Sequence	Action	Description
1	Processor A sets General Interrupt request bit	Processor A sets its associated General Interrupt request bit (GIRn = 1) in the control register (ACR).
2	General Interrupt Request Pending status bit is set	The General Interrupt Request Pending status bit (GIPn) in the status register (BSR) is set to "1"
3	General Interrupt request to Processor B is generated	Setting the GIPn bit generates the General Interrupt request to Processor B (Interrupt Request Enable bit, GIEn, must be set for Processor B)
4	Processor B reads status register	The Processor B reads the GIPn bit in the BSR register.
5	Processor B services the interrupt	-
6	Processor B sets GIPn bit to clear interrupt	The Processor B writes "1" to the corresponding GIPn bit to clear the interrupt
7	GIRn bit is cleared	Setting the GIPn bit to "1" clears the General Interrupt request bit (GIRn) in the Processor A control register (ACR).

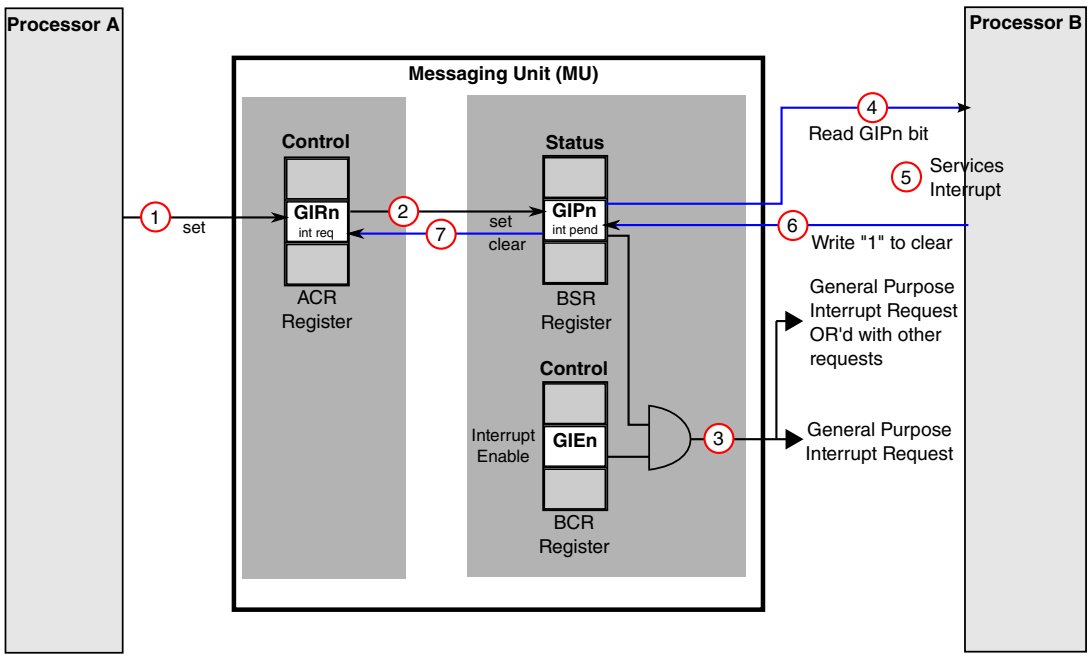


Figure 4-9. Messaging Model Using a General Purpose Interrupt

4.3.2.9 Exclusive Access to Shared Memory

You can use the MU to signal one processor about its current access to the shared memory, so that the data is not overwritten by the other processor during the exclusive memory access period.

The following tables describe the signaling protocol that the Processor A uses to inform the Processor B about its current access (write) to the shared memory, assuming that the set of bits and registers (GIR0 bit, BRR0 register, BTR0 register, GIR0 bit, ARR0 register, ATR0 register) are reserved to support exclusive access to the shared memory protocol.

Table 4-10. How the Processor A Performs an Exclusive Access to Shared Memory

Sequence	Action	Description
1	Processor A sends GIRn request to Processor B using Processor A control register	When the Processor A wants to perform an exclusive access to the shared memory, the Processor A sends an GIR0 request to the Processor B.
2	Processor A sends an exclusive-access request using a transmit data register (ATRn)	The Processor A will send an exclusive-access request (command, location, and length of target access) to Processor B using a selected transmit data register (ATR0).
3	Processor A waits for a dedicated interrupt from Processor B	The Processor A waits for a dedicated interrupt (as an acknowledgement) triggered by the Processor B before proceeding.

Table continues on the next page...

Table 4-10. How the Processor A Performs an Exclusive Access to Shared Memory (continued)

Sequence	Action	Description
4	Processor A accesses shared memory	After receiving a dedicated interrupt from the Processor B, Processor A proceeds.

Table 4-11. How the Processor B Scans for Transaction Information

Sequence	Action	Description
1	Processor B receives an interrupt from a receive data register (BRRn)	-
2	Processor B reads the receive data register (BRRn)	-
3	Processor B scans the receive data register contents	For transaction information (whether Processor A has requested an exclusive-access)

Table 4-12. How the Processor B Accepts Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B triggers a dedicated interrupt	Processor B acknowledges the Processor A request by triggering a dedicated interrupt (ack) to the Processor A.
2	Processor B sends a code message to Processor A	Along with the acknowledge interrupt, the Processor B sends a code message to the Processor A through the selected transmit register (BTRn). The message informs the Processor A that it can exclusively access the shared memory.

Table 4-13. How the Processor B Rejects Exclusive Access by Processor A

Sequence	Action	Description
1	Processor B ignores Processor A request for exclusive access	If the Processor B does not want to give go-ahead permission to the Processor A, Processor B ignores the exclusive access request.

4.3.2.10 Packet Data Transfers

The following example describes the packet transfer sequence between the Processor B and Processor A subsystems:

Table 4-14. Packet Data Transfer Sequence

Action	Sequence	Description
Processor B requests DMA	1	The Processor B sends a DMA request to initiate the packet data transfer

Table continues on the next page...

Table 4-14. Packet Data Transfer Sequence (continued)

Action	Sequence	Description
DMA data transfer	2	DMA acknowledges.
	3	DMA starts transferring data from the specified Processor B location to the specified shared memory
	4	DMA interrupts the Processor B to signal that the packet transfer has finished.
Processor B informs Processor A that data is in shared memory	5	Using an MU Processor B-side transmit register, the Processor B sends a packet information message to the Processor A to inform the Processor A of the arrival of new packet data that is stored in shared memory . The message contains the command, location, and length of packet data information.
Processor A receives interrupt	6	The Processor A receives an interrupt (assuming its corresponding Processor A MU-side receive interrupt is enabled), and the pending processing task becomes active and processes packet data from memory.
Processor A reads data, writes data	7	The Processor A reads or processes packet data from shared memory.
	8	The Processor A writes the result from packet processing to a separate buffer.
Processor A informs Processor B that transfer is finished	9	After the processing of the packet data finishes, the Processor A informs the Processor B (using the MU Processor A-side transmit register, ATRn).
Processor A sends interrupt to Processor B (request for more data)	10	The Processor B receives the next interrupt from the Processor A, in which the Processor A requests more packet data.

4.3.2.11 MU Resets

The MU has two sources of reset, and each reset has a different function from the MU or system perspective.

- One asynchronous system that is connected to both sides of the MU interface.
- One programmable hardware reset (MUR bit) in the ACR register (on the Processor A-side).

Table 4-15. MU programmable resets

Reset	Description
Processor A MU reset	• Processor A MU Reset bit (MUR) of the ACR register • The MUR reset affects the messaging section on both the Processor A and the Processor B sides. The MUR reset causes all control and status registers to return to their default values and all internal states to be cleared. • It is up to the Processor A software to decide whether to use the MUR reset or not. • The instruction immediately following assertion of the MUR bit should not write to MU registers. Such a write may be overwritten by the reset sequence and the register will remain with the reset value. You should wait at least one instruction (after assertion of the MUR bit) before attempting a write to MU registers.

After issuing MUR bit reset events, the Processor A programmer can verify that the reset sequence on the Processor B-side has ended, by checking the RS bit in the ASR register.

NOTE

MUR bit assertion is a delicate operation because it affects the other side's registers asynchronously. MUR bit assertion may cause unpredictable behavior if, for example, the Processor B is concurrently testing an MU register bit (TE bit in Processor B SR register). Before asserting the MUR bit, you should verify that the Processor B is not presently engaged in an MU signalling activity.

4.3.3 Software Restrictions

This section describes certain software restrictions when accessing the MU.

4.3.3.1 General Restrictions

This section lists the restrictions that apply to both the sides (Processor A, Processor B) of the MU.

4.3.3.1.1 Write-After-Write to a Transmit Register

A write to a transmit register signals the receiver side that data is ready for retrieval.

- Writing to the transmit register again without verifying that the data was retrieved is prohibited, because the transmitter side has no way of knowing the exact time that the receiver will attempt to retrieve the data.
- Before attempting to write the transmit register again, the transmitter side should wait for a "Transmitter Empty" interrupt, or should poll the "Transmitter Empty" bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being read on the receiver side of the MU.

4.3.3.1.2 Read-After-Read from a Receive Register

A read of a receive register signals the transmitter side that data can be written to that register. In the same way, the receiver processor should not read a receive register before receiving a "Receiver Full" interrupt or polling the "Receiver Full" bit in the Status Register.

- Reading the receive register again without verifying that the data was written is prohibited, because the receiver side has no way of knowing the exact time that the transmitter will attempt to write the data.
- Before attempting to read the receive register again, the receiver side should wait for a “Receiver Full” interrupt, or should poll the “Receiver Full” bit in the Status Register.
- Failure to follow this restriction may result in the wrong data being written on the transmitter side of the MU.

4.3.3.2 Processor Restrictions

This section lists the restrictions that apply each side of the processor in the MU.

4.3.3.2.1 Before Entering Low Power Mode

Before entering Low Power mode, the Processor should verify that the Processor Event Pending (EP) bit in the Status Register is cleared.

- If the Event Pending bit (EP) is still set to “1”, then the Processor should wait and poll the EP bit until it is cleared, before executing the LPM instruction.
- Note that if the other Processor is in Low Power mode (programmed for clock gating in CCM), the EP bit may be stuck high. In this case, the other Processor clock must be turned ON to get the EP bit cleared before the Processor can enter Low Power mode.
- To discover which power mode the other Processor is in, the Processor can check the PM bits in the xSR register.

4.3.3.2.2 Before Setting a General Interrupt Request Bit (GIR0–3)

Before setting a General Interrupt Request bit (GIR0–3), you must verify that the GIRn bit is cleared, which means that a general interrupt is not pending. Generally, setting the GIRn bit while the bit is set to “1” will be ignored, but in some cases it may issue a second interrupt. This restriction is meant to prevent this indeterministic behavior.

4.3.3.2.3 Reset Bit Restrictions

The reset bit (MUR, HR) restrictions are:

- Before asserting the MUR bit in the ACR register, verify that the Processor B-side is not engaged in some MU activity.
- Do not write to an MU register in the instruction immediately after the assertion of the MUR bit in the ACR register, because the written data can be overridden by the reset value.

4.3.4 MU Processor A-side Memory Map/Register Definition

This section contains the detailed register descriptions for the Processor A-side MU registers.

MUA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AA_0000	Processor A Transmit Register 0 (MUA_ATR0)	32	R/W	0000_0000h	4.3.4.1/133
30AA_0004	Processor A Transmit Register 1 (MUA_ATR1)	32	R/W	0000_0000h	4.3.4.2/134
30AA_0008	Processor A Transmit Register 2 (MUA_ATR2)	32	R/W	0000_0000h	4.3.4.3/135
30AA_000C	Processor A Transmit Register 3 (MUA_ATR3)	32	R/W	0000_0000h	4.3.4.4/135
30AA_0010	Processor A Receive Register 0 (MUA_ARR0)	32	R	0000_0000h	4.3.4.5/136
30AA_0014	Processor A Receive Register 1 (MUA_ARR1)	32	R	0000_0000h	4.3.4.6/137
30AA_0018	Processor A Receive Register 2 (MUA_ARR2)	32	R	0000_0000h	4.3.4.7/137
30AA_001C	Processor A Receive Register 3 (MUA_ARR3)	32	R	0000_0000h	4.3.4.8/138
30AA_0020	Processor A Status Register (MUA_ASR)	32	R/W	00F0_0080h	4.3.4.9/139
30AA_0024	Processor A Control Register (MUA_ACR)	32	R/W	0000_0000h	4.3.4.10/ 142

4.3.4.1 Processor A Transmit Register 0 (MUA_ ATR0)

Use Processor A Transmit Register 0 (ATR0, 32-bit, write-only) to transmit a message or data to the Processor B.

- You can only write to the ATR0 register when the TE0 bit in ASR register is set to “1”.
- Reading the ATR0 register returns all zeros.

Address: 30AA_0000h base + 0h offset = 30AA_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ATR0 field descriptions

Field	Description
ATR0	Processor A Transmit Register 0. (Write-only)

MUA_ATR0 field descriptions (continued)

Field	Description
	• Data written to the ATR0 register is reflected on the Processor B-side in the Processor B Receive Register 0 (BRR0). The ATR0 and BRR0 registers are not double-buffered—a write to the ATR0 register overrides the data readable at the BRR0 register. • A write to the transmit register clears a “transmitter empty” bit (TE0) in the Processor A Status Register (ASR) on the transmitter side, and sets a “receiver full” bit (RF0) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 0 on the Processor B-side). • Any write to the ATR0 register will update all status information.

4.3.4.2 Processor A Transmit Register 1 (MUA_ATR1)

Use Processor A Transmit Register 1 (ATR1, 32-bit, write-only) to transmit a message or data to the Processor B.

- You can only write to the ATR1 register when the TE1 bit in ASR register is set to “1”.
- Reading the ATR1 register returns all zeros.

Address: 30AA_0000h base + 4h offset = 30AA_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ATR1 field descriptions

Field	Description
ATR1	Processor A Transmit Register 1. (Write-only) • Data written to the ATR1 register is reflected on the Processor B-side in the Processor B Receive Register 1 (BRR1). The ATR1 and BRR1 registers are not double-buffered—a write to the ATR1 register overrides the data readable at the BRR1 register. • A write to the transmit register clears a “transmitter empty” bit (TE1) in the Processor A Status Register (ASR) on the transmitter side, and sets a “receiver full” bit (RF1) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 1 on the Processor B-side). • Any write to the ATR1 register will update all status information.

4.3.4.3 Processor A Transmit Register 2 (MUA_ATR2)

Use Processor A Transmit Register 2 (ATR2, 32-bit, write-only) to transmit a message or data to the Processor B.

- You can only write to the ATR2 register when the TE2 bit in ASR register is set to "1".
- Reading the ATR2 register returns all zeros.

Address: 30AA_0000h base + 8h offset = 30AA_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ATR2 field descriptions

Field	Description
ATR2	Processor A Transmit Register 2. (Write-only) • Data written to the ATR2 register is reflected on the Processor B-side in the Processor B Receive Register 2 (BRR2). The ATR2 and BRR2 registers are not double-buffered—a write to the ATR2 register overrides the data readable at the BRR2 register. • A write to the transmit register clears a "transmitter empty" bit (TE2) in the Processor A Status Register (ASR) on the transmitter side, and sets a "receiver full" bit (RF2) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 2 on the Processor B-side). • Any write to the ATR2 register will update all status information.

4.3.4.4 Processor A Transmit Register 3 (MUA_ATR3)

Use Processor A Transmit Register 3 (ATR3, 32-bit, write-only) to transmit a message or data to the Processor B.

- You can only write to the ATR3 register when the TE3 bit in ASR register is set to "1".
- Reading the ATR3 register returns all zeros.

Address: 30AA_0000h base + Ch offset = 30AA_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATR3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ATR3 field descriptions

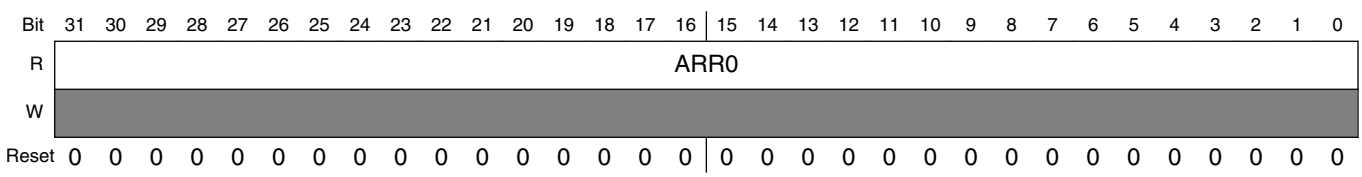
Field	Description
ATR3	Processor A Transmit Register 3. (Write-only) • Data written to the ATR3 register is reflected on the Processor B-side in the Processor B Receive Register 3 (BRR3). The ATR3 and BRR3 registers are not double-buffered—a write to the ATR3 register overrides the data readable at the BRR3 register. • A write to the transmit register clears a “transmitter empty” bit (TE3) in the Processor A Status Register (ASR) on the transmitter side, and sets a “receiver full” bit (RF3) in the Processor B Status Register (BSR) on the receiver side (optionally triggering an interrupt 3 on the Processor B-side). • Any write to the ATR3 register will update all status information.

4.3.4.5 Processor A Receive Register 0 (MUA_ARR0)

Use Processor A Receive Register 0 (ARR0, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR0 register is immediately reflected in the ARR0 register.
- You can only read the ARR0 register when the RF0 bit in the ASR register is set to “1”.
- Writing to the ARR0 register generates an error response to the Processor A.

Address: 30AA_0000h base + 10h offset = 30AA_0010h



MUA_ARR0 field descriptions

Field	Description
ARR0	Processor A Receive Register 0. (Read-only) • Reflects the data written to Processor B Transmit Register 0 (BTR0). • Reading the ARR0 register clears the “receiver full” bit (RF0) in the Processor A Status Register (ASR) on the receiver side, and sets the “transmitter empty” bit (TE0) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor B-side). • Any read of the ARR0 register will update all status information.

4.3.4.6 Processor A Receive Register 1 (MUA_ARR1)

Use Processor A Receive Register 1 (ARR1, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR1 register is immediately reflected in the ARR1 register.
- You can only read the ARR1 register when the RF1 bit in the ASR register is set to "1".
- Writing to the ARR1 register generates an error response to the Processor A.

Address: 30AA_0000h base + 14h offset = 30AA_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARR1																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ARR1 field descriptions

Field	Description
ARR1	Processor A Receive Register 1. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (BTR1). • Reading the ARR1 register clears the "receiver full" bit (RF1) in the Processor A Status Register (ASR) on the receiver side, and sets the "transmitter empty" bit (TE1) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor B-side). • Any read of the ARR1 register will update all status information.

4.3.4.7 Processor A Receive Register 2 (MUA_ARR2)

Use Processor A Receive Register 2 (ARR2, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR2 register is immediately reflected in the ARR2 register.
- You can only read the ARR2 register when the RF2 bit in the ASR register is set to "1".
- Writing to the ARR2 register generates an error response to the Processor A.

Address: 30AA_0000h base + 18h offset = 30AA_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ARR2																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUA_ARR2 field descriptions

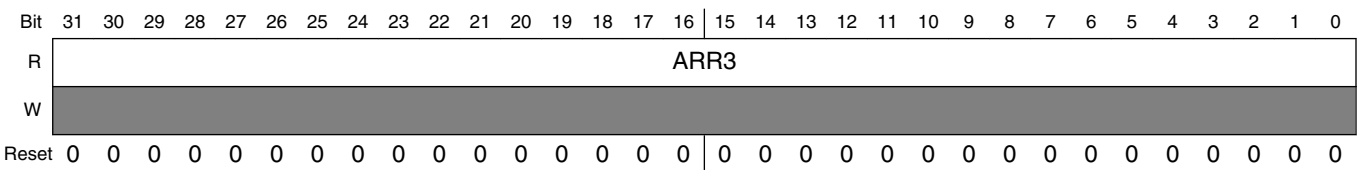
Field	Description
ARR2	Processor A Receive Register 2. (Read-only) • Reflects the data written to Processor B Transmit Register 1 (BTR2). • Reading the ARR2 register clears the “receiver full” bit (RF2) in the Processor A Status Register (ASR) on the receiver side, and sets the “transmitter empty” bit (TE2) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor B-side). • Any read of the ARR2 register will update all status information.

4.3.4.8 Processor A Receive Register 3 (MUA_ARR3)

Use Processor A Receive Register 3 (ARR3, 32-bit, read-only) to receive a message or data from the Processor B.

- Data written to the BTR3 register is immediately reflected in the ARR3 register.
- You can only read the ARR3 register when the RF3 bit in the ASR register is set to “1”.
- Writing to the ARR3 register generates an error response to the Processor A.

Address: 30AA_0000h base + 1Ch offset = 30AA_001Ch



MUA_ARR3 field descriptions

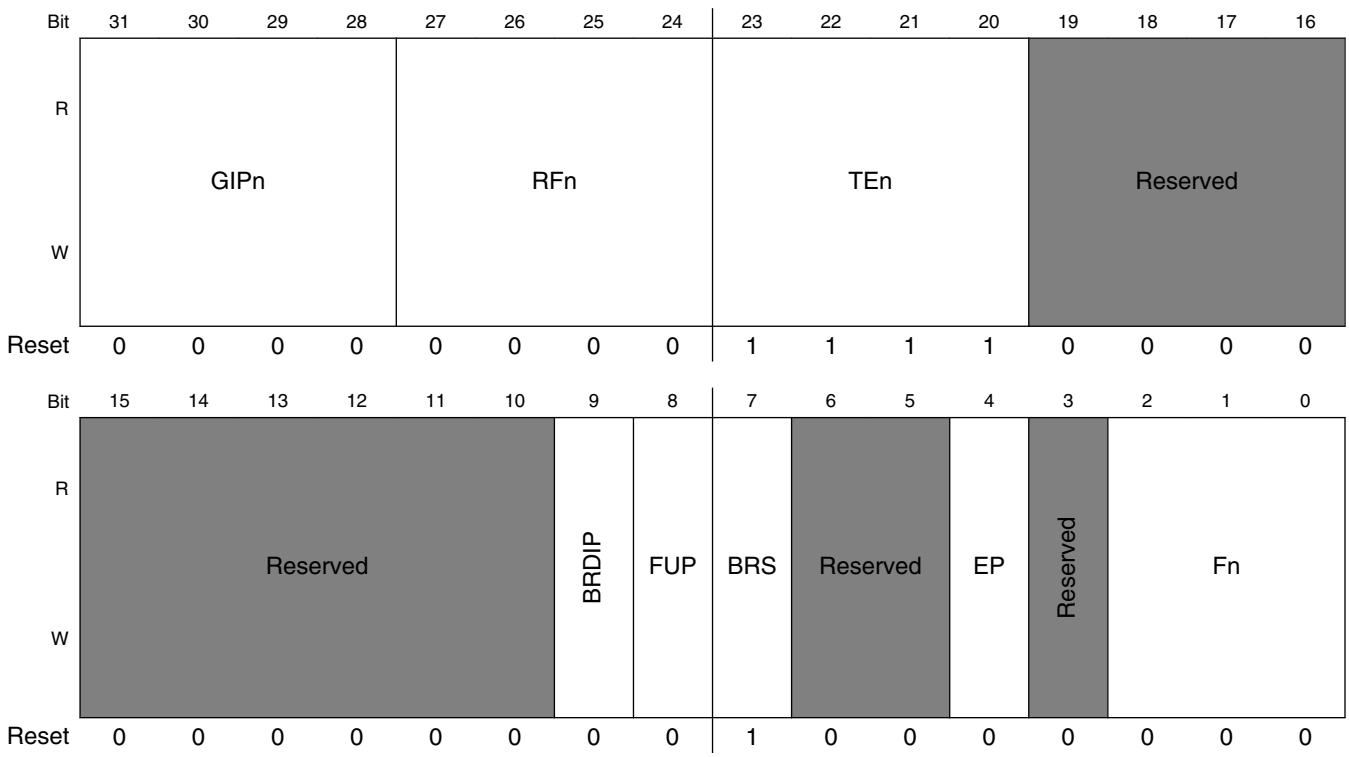
Field	Description
ARR3	Processor A Receive Register 3. (Read-only) • Reflects the data written to Processor B Transmit Register 3 (BTR3). • Reading the ARR3 register clears the “receiver full” bit (RF3) in the Processor A Status Register (ASR) on the receiver side, and sets the “transmitter empty” bit (TE3) in the Processor B Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor B-side). • Any read of the ARR3 register will update all status information.

4.3.4.9 Processor A Status Register (MUA_ASR)

Use the Processor A Status Register (ASR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, and to set dual function control-status bits.

- Some dual-purpose bits are set by the MU logic, and cleared by the Processor A-side programmer
- Other dual-purpose bits are set by the Processor A-side programmer, and cleared by the MU logic.

Address: 30AA_0000h base + 20h offset = 30AA_0020h



MUA_ASR field descriptions

Field	Description
31–28 GIPn	For n = {0, 1, 2, 3} Processor A General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor A that the GIRn bit in the BCR register on the Processor B-side was set from “0” to “1”. If the GIEn bit in the ACR register is set to “1”, a General Interrupt n request is issued. • The GIPn bit is cleared by writing it back as “1”. Writing “0”, or writing “1” when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. The proper bit clearing sequence is: clear an Processor A register, set the desired bit in it (Processor A register), and write it to the ASR register, thus clearing the GIPn bit. • GIPn bit is cleared when the MU is reset.

Table continues on the next page...

MUA_ASR field descriptions (continued)

Field	Description
	0 Processor A general purpose interrupt n is not pending. (default) 1 Processor A general purpose interrupt n is pending.
27–24 RFn	For n = {0, 1, 2, 3} Processor A Receive Register n Full. (Read-only) • The RFn bit is set to “1” when the BTRn register is written on the Processor B-side. • After the RFn bit is set to “1”, the RFn bit signals the Processor A-side that new data is ready to be read by the Processor A in the ARRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the ACR register has been set to “1”). • RFn bit is cleared when the ARRn register is read, and when the MU is reset. 0 ARRn register is not full (default). 1 ARRn register has received data from BTRn register and is ready to be read by the Processor A.
23–20 TEn	For n = {0, 1, 2, 3} Processor A Transmit Register n Empty. (Read-only) • The TEn bit is set to “1” after the BRRn register is read on the Processor B-side. • After the TEn bit is set to “1”, the TEn bit signals the Processor A-side that the ATRn register is ready to be written on the Processor A-side, and a Transmit n interrupt is issued on the Processor A-side (if the TEn bit in the ACR register is set to “1”). • TEn bit is cleared after the ATRn register is written on the Processor A-side. • TEn bit is set to “1” when the MU is reset. 0 ATRn register is not empty. 1 ATRn register is empty (default).
19–10 -	This field is reserved. Reserved.
9 BRDIP	Processor B Reset De-asserted Interrupt Pending. (Read-Write) • BRDIP bit signals the Processor A-side that the Processor B-side has come out of reset. • BRDIP bit is set to “1” after the MU Processor B-side comes out of reset, after synchronization. The interrupt generated by a Processor B-side reset de-assertion is ORed with the Processor A general purpose interrupt 3. The Processor A general purpose interrupt 3 is issued when the Processor B-side comes out of reset (if the interrupt is enabled). • To clear the BRDIP bit, write “1”, which also clears general purpose interrupt 3. • When Processor A-side of MU comes out of reset BRDIP bit has value “0”(default).Then Processor A sees the status of Processor B-side and if Processor B-side has come out of reset then BRDIP bit goes high.This takes 5-6 clock cycles.And if you read BRDIP bit now you will see it as high although its reset value was "0". 0 The Processor A general purpose interrupt 3, because of a Processor B-side reset de-assertion, is cleared (default). 1 The Processor B-side is out of reset.
8 FUP	Processor A Flags Update Pending. (Read-only) • FUP bit is set to “1” when the Processor A-side sends a Flags Update request to the Processor B-side. • A Flags Update request is generated when the ABF[2:0] bits of the ACR register change. No flag update changes are allowed while the FUP bit is set to “1”. Any write to the ABF[2:0] bits, while the FUP bit is set to “1”, will not generate a Flags Update event, and the ABF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor B-side, and during MU reset.

Table continues on the next page...

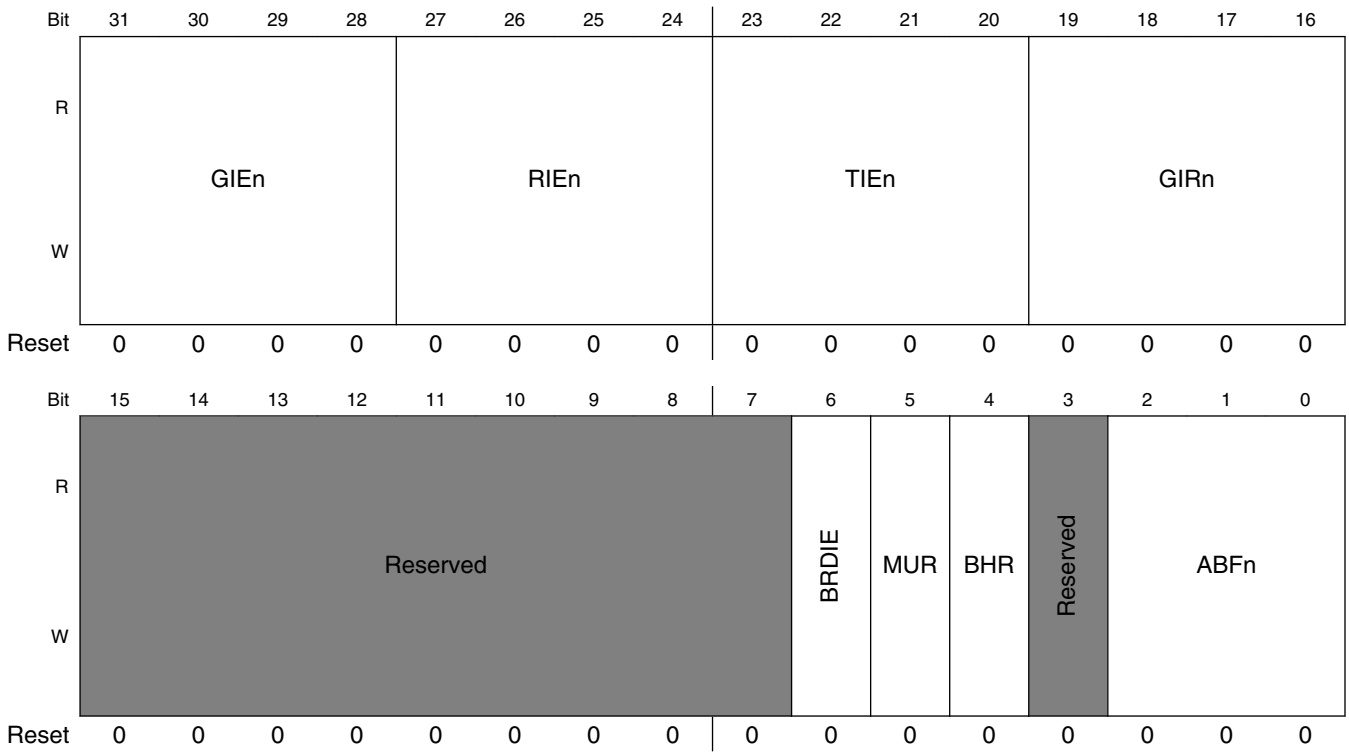
MUA_ASR field descriptions (continued)

Field	Description
	0 No flags updated, initiated by the Processor A, in progress (default) 1 Processor A initiated flags update, processing
7 BRS	Processor B-side Reset State. (Read-only) • BRS bit indicates if the Processor B-side of the MU is in a reset state or not. • If the BRS bit is set to “1”, then the Processor B-side of the MU is still in the reset state. • If the BRS bit is cleared, then the Processor B-side of the MU are out of reset. • The BRS bit is set to “1” during: a Processor B system reset, or an MU reset (caused by setting the MUR bit at the ACR register). • The BRS bit is cleared when the reset sequence on the Processor B-side of the MU ends. After issuing any of the reset events mentioned previously, you should verify that the BRS bit is cleared before starting any accesses. • When Processor A side of MU comes out of reset BRS bit has value “1”(default).Then Processor A sees the status of Processor B-side and if Processor B-side has come out of reset then BRS bit goes low.This takes 5-6 clock cycles.And if you read BRS bit now you will see it as low although its reset value was "1" . 0 The Processor B-side of the MU is not in reset. 1 The Processor B-side of the MU is in reset.
6–5 -	This field is reserved. Reserved
4 EP	Processor A-Side Event Pending. (Read-only) • EP bit is set to “1” when the Processor A-side mechanism sends an event update request to the Processor B-side. • EP bit is cleared when the event update acknowledge is received. An “event” is any hardware message that is reflected in the BSR register on the Processor B-side (for example, “transmit register 0 written”). During normal operations, you do not have to deal with the state of the EP bit because the event update mechanism works automatically. • To ensure events have been posted to Processor B before entering STOP mode, you should verify that the EP bit is cleared. If EP bit is set to “1”, you should wait and continue to poll it (EP bit) before entering STOP mode. • Reading the ASR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOPor WAIT modes; otherwise, the EP bit may be set by subsequent additional actions. • The EP bit is cleared when the MU resets. 0 The Processor A-side event is not pending (default). 1 The Processor A-side event is pending.
3 -	This field is reserved. Reserved.
Fn	For n = {0, 1, 2} Processor A-Side Flag n. (Read-only) • Fn bit is the Processor A-side flag that reflects the values written to the BAFn bit in the Processor B control register. • Every time that the BAFn bit is written, the BAFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor B and the Processor A. 0 BAFn bit in BCR register is written 0 (default). 1 BAFn bit in BCR register is written 1.

4.3.4.10 Processor A Control Register (MUA_ACR)

Use the Processor A Control Register (ACR, 32-bit, read-write) to enable the MU interrupts on the Processor A-side, and trigger events and interrupts on the Processor B-side (general purpose interrupt, flag update).

Address: 30AA_0000h base + 24h offset = 30AA_0024h



MUA_ACR field descriptions

Field	Description
31–28 GIEn	For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Enable n. (Read-Write) When GIEn=0 corresponds to the high order bit and GIE3 corresponds to the low order bit. • GIEn bit enables Processor A General Interrupt n. • If GIEn bit is set to “1” (enabled), then a General Interrupt n request is issued when the GIPn bit in the ASR register is set to “1”. • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0 Disables Processor A General Interrupt n. (default) 1 Enables Processor A General Interrupt n.
27–24 RIEn	For n = {0, 1, 2, 3} Processor A Receive Interrupt Enable n. (Read-Write)

Table continues on the next page...

MUA_ACR field descriptions (continued)

Field	Description
	• RIEn bit enables Processor A Receive Interrupt n. • If RIEn bit is set to “1” (enabled), then an Processor A Receive Interrupt n request is issued when the RFn bit in the ASR register is set to “1”. • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor A Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0 Disables Processor A Receive Interrupt n. (default) 1 Enables Processor A Receive Interrupt n.
23–20 TIEn	For n = {0, 1, 2, 3} Processor A Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor A Transmit Interrupt n. • If TIEn bit is set to “1” (enabled), then an Processor A Transmit Interrupt n request is issued when the TEn bit in the ASR register is set to “1”. • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor A Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0 Disables Processor A Transmit Interrupt n. (default) 1 Enables Processor A Transmit Interrupt n.
19–16 GIRn	For n = {0, 1, 2, 3} Processor A General Purpose Interrupt Request n. (Read-Write) • Writing “1” to the GIRn bit sets the GIPn bit in the BSR register on the Processor B-side. If the GIEn bit in the BCR register is set to “1” on the Processor B-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the BSR register on the Processor B-side) is cleared by writing it (GIPn bit) as “1”, thereby signalling the Processor A that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as “0” on the Processor A-side. • To ensure proper operations, you must verify that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0 Processor A General Interrupt n is not requested to the Processor B (default). 1 Processor A General Interrupt n is requested to the Processor B.
15–7 -	This field is reserved. Reserved.
6 BRDIE	Processor B Reset De-assertion Interrupt Enable. (Read-Write) • BRDIE bit enables Processor A General Interrupt 3. • If BRDIE bit is set to “1”, then General Interrupt 3 request is issued to the Processor A when the BRDIP bit in the ASR register is set to “1”. • If BRDIE is cleared, then the value of the BRDIP bit is ignored and no General Interrupt 3 request will be issued. • The BRDIE bit is cleared when the MU resets. 0 Disables the Processor A General Purpose Interrupt 3 request due to the Processor B reset de-assertion to the Processor A. Processor B reset deassertion causes Processor B and MU-Processor B side to come out of reset thus setting BRDIP bit to “1”. 1 Enables Processor A General Purpose Interrupt 3 request due to the Processor B reset de-assertion to the Processor A.
5 MUR	Processor A MU Reset.

Table continues on the next page...

MUA_ACR field descriptions (continued)

Field	Description
	• Setting MUR bit to “1” resets both the Processor B and the Processor A sides of the MU module, forcing all control and status registers to return to their default values (except the BHR bit in the ACR register and BHRM bit in BCR register), and all internal states to be cleared. • Before setting the MUR bit to “1”, it is advisable to interrupt the Processor B, because setting the MUR bit may affect the ongoing Processor B program. • After setting the MUR bit, you should monitor the value of the BRS bit in the ASR register to know when the reset sequence on the Processor B-side has ended. • MUR bit can only be written as “1”. • MUR bit is always read as “0”. • MUR bit is cleared during the MU reset sequence. 0 N/A. Self clearing bit (default). 1 Asserts the Processor A MU reset.
4 BHR	Processor B Hardware Reset. (Read-Write) • BHR bit asserts and de-asserts the hardware reset of the Processor B. • Set BHR bit to “1” to start a hardware reset of the Processor B. • Clear the BHR bit to de-assert the Processor B hardware reset input. • Assert the BHR bit for a minimum of 3 clock cycles of network clock (sampling clock in SRC) clock. The BRS bit in MU_ASR register (b[7]) indicates the state of the Processor B. As soon as the Processor B goes into Reset (BRS bit is set to “1”), the BHR bit can be de-asserted. • Strobe-setting the BHR bit will not cause an internal MU reset but will be routed outside MU to Processor B domain reset logic • After clearing the BHR bit, monitor the value of the BRS bit at the ASR to know when the Processor B reset sequence has ended. • The BHR reset issued by the Processor A to the Processor B is maskable by the Processor B (according to the settings of the BHRM bit in the BCR register). If the BHRM bit (in the BCR register) is set to “1”, then the BHR reset is masked; if the BHRM bit (in the BCR register) is cleared (default), then the BHR reset is enabled. • The BHR bit does not return to the reset value during the software (MUR) reset. 0 De-assert Hardware reset to the Processor B. (default) 1 Assert Hardware reset to the Processor B.
3 -	This field is reserved. Reserved
ABFn	For n = {0, 1, 2} Processor A to Processor B Flag n. (Read-Write) • ABFn bit is a read-write flag that is reflected in Fn bit in the BSR register on the Processor B-side. • ABFn bit is cleared when the MU resets. 0 N/A. Self clearing bit (default). 1 Asserts the Processor A MU reset.

4.3.5 MU Processor B-side Memory Map/Register Definition

This section contains the detailed register descriptions for the Processor B-side MU registers.

MUB memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AB_0000	Processor B Transmit Register 0 (MUB_BTR0)	32	R/W	0000_0000h	4.3.5.1/145
30AB_0004	Processor B Transmit Register 1 (MUB_BTR1)	32	R/W	0000_0000h	4.3.5.2/146
30AB_0008	Processor B Transmit Register 2 (MUB_BTR2)	32	R/W	0000_0000h	4.3.5.3/146
30AB_000C	Processor B Transmit Register 3 (MUB_BTR3)	32	R/W	0000_0000h	4.3.5.4/147
30AB_0010	Processor B Receive Register 0 (MUB_BRR0)	32	R	0000_0000h	4.3.5.5/148
30AB_0014	Processor B Receive Register 1 (MUB_BRR1)	32	R	0000_0000h	4.3.5.6/148
30AB_0018	Processor B Receive Register 2 (MUB_BRR2)	32	R	0000_0000h	4.3.5.7/149
30AB_001C	Processor B Receive Register 3 (MUB_BRR3)	32	R	0000_0000h	4.3.5.8/150
30AB_0020	Processor B Status Register (MUB_BSR)	32	R/W	00F0_0080h	4.3.5.9/151
30AB_0024	Processor B Control Register (MUB_BCR)	32	R/W	0000_0000h	4.3.5.10/ 154

4.3.5.1 Processor B Transmit Register 0 (MUB_BTR0)

Use Processor B Transmit Register 0 (BTR0, 32-bit, write-only) to transmit a message or data to the Processor A.

- You can only write to the BTR0 register when the TE0 bit in BSR register is set to “1”.
- Reading the BTR0 register returns all zeros.

Address: 30AB_0000h base + 0h offset = 30AB_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUB_BTR0 field descriptions

Field	Description
BTR0	Processor B Transmit Register 0. (Write-only) • Data written to the BTR0 register is reflected on the Processor A-side in the Processor A Receive Register 0 (ARR0). The BTR0 and ARR0 registers are not double-buffered—a write to the BTR0 register overrides the data readable at the ARR0 register. • A write to the transmit register clears a “transmitter empty” bit (TE0) in the Processor B Status Register (BSR) on the transmitter side, and sets a “receiver full” bit (RF0) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 0 on the Processor A-side). • Any write to the BTR0 register will update all status information.

4.3.5.2 Processor B Transmit Register 1 (MUB_BTR1)

Use Processor B Transmit Register 1 (BTR1, 32-bit, write-only) to transmit a message or data to the Processor A.

- You can only write to the BTR1 register when the TE1 bit in BSR register is set to "1".
- Reading the BTR1 register returns all zeros.

Address: 30AB_0000h base + 4h offset = 30AB_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUB_BTR1 field descriptions

Field	Description
BTR1	Processor B Transmit Register 1. (Write-only) • Data written to the BTR1 register is reflected on the Processor A-side in the Processor A Receive Register 1 (ARR1). The BTR1 and ARR1 registers are not double-buffered—a write to the BTR1 register overrides the data readable at the ARR1 register. • A write to the transmit register clears a "transmitter empty" bit (TE1) in the Processor B Status Register (BSR) on the transmitter side, and sets a "receiver full" bit (RF1) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 1 on the Processor A-side). • Any write to the BTR1 register will update all status information.

4.3.5.3 Processor B Transmit Register 2 (MUB_BTR2)

Use Processor B Transmit Register 2 (BTR2, 32-bit, write-only) to transmit a message or data to the Processor A.

- You can only write to the BTR2 register when the TE2 bit in BSR register is set to "1".
- Reading the BTR2 register returns all zeros.

Address: 30AB_0000h base + 8h offset = 30AB_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BTR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUB_BTR2 field descriptions

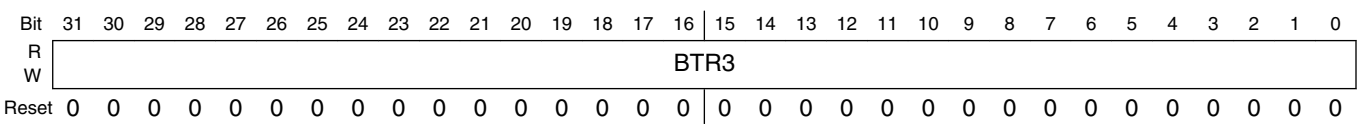
Field	Description
BTR2	Processor B Transmit Register 2. (Write-only) • Data written to the BTR2 register is reflected on the Processor A-side in the Processor A Receive Register 2 (ARR2). The BTR2 and ARR2 registers are not double-buffered—a write to the BTR2 register overrides the data readable at the ARR2 register. • A write to the transmit register clears a “transmitter empty” bit (TE2) in the Processor B Status Register (BSR) on the transmitter side, and sets a “receiver full” bit (RF2) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 2 on the Processor A-side). • Any write to the BTR2 register will update all status information.

4.3.5.4 Processor B Transmit Register 3 (MUB_BTR3)

Use Processor B Transmit Register 3 (BTR3, 32-bit, write-only) to transmit a message or data to the Processor A.

- You can only write to the BTR3 register when the TE3 bit in BSR register is set to “1”.
- Reading the BTR3 register returns all zeros.

Address: 30AB_0000h base + Ch offset = 30AB_000Ch



MUB_BTR3 field descriptions

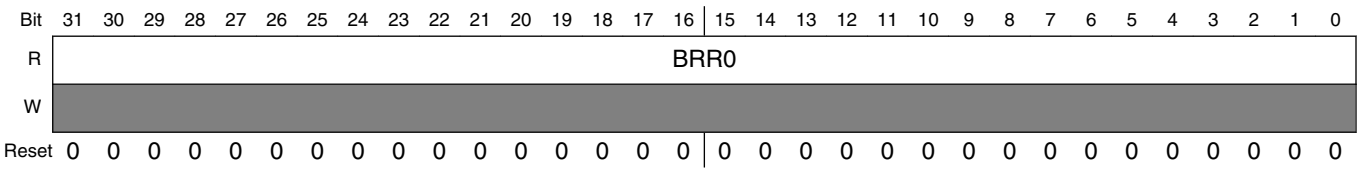
Field	Description
BTR3	Processor B Transmit Register 3. (Write-only) • Data written to the BTR3 register is reflected on the Processor A-side in the Processor A Receive Register 3 (ARR3). The BTR3 and ARR3 registers are not double-buffered—a write to the BTR3 register overrides the data readable at the ARR3 register. • A write to the transmit register clears a “transmitter empty” bit (TE3) in the Processor B Status Register (BSR) on the transmitter side, and sets a “receiver full” bit (RF3) in the Processor A Status Register (ASR) on the receiver side (optionally triggering an interrupt 3 on the Processor A-side). • Any write to the BTR3 register will update all status information.

4.3.5.5 Processor B Receive Register 0 (MUB_BRR0)

Use Processor B Receive Register 0 (BRR0, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR0 register is immediately reflected in the BRR0 register.
- You can only read the BRR0 register when the RF0 bit in the BSR register is set to "1".
- Writing to the BRR0 register generates an error response to the Processor B.

Address: 30AB_0000h base + 10h offset = 30AB_0010h



MUB_BRR0 field descriptions

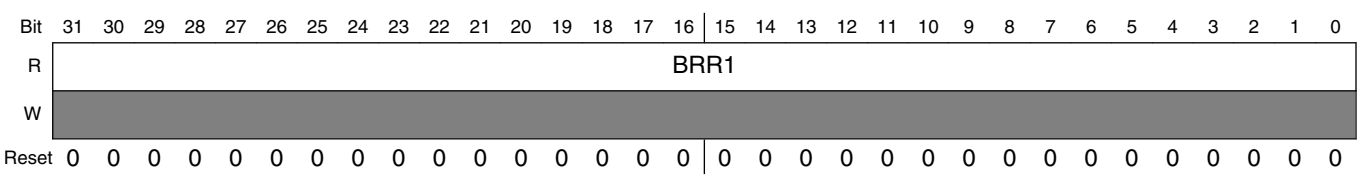
Field	Description
BRR0	Processor B Receive Register 0. (Read-only) • Reflects the data written to Processor A Transmit Register 0 (ATR0). • Reading the BRR0 register clears the "receiver full" bit (RF0) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE0) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 0 on the Processor A-side). • Any read of the BRR0 register will update all status information.

4.3.5.6 Processor B Receive Register 1 (MUB_BRR1)

Use Processor B Receive Register 1 (BRR1, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR1 register is immediately reflected in the BRR1 register.
- You can only read the BRR1 register when the RF1 bit in the BSR register is set to "1".
- Writing to the BRR1 register generates an error response to the Processor B.

Address: 30AB_0000h base + 14h offset = 30AB_0014h



MUB_BRR1 field descriptions

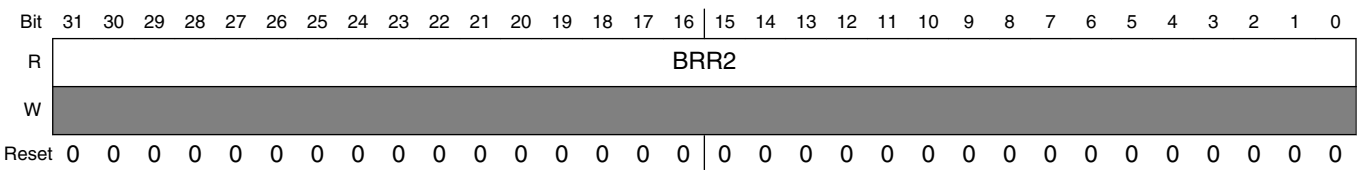
Field	Description
BRR1	Processor B Receive Register 1. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (ATR1). • Reading the BRR1 register clears the “receiver full” bit (RF1) in the Processor B Status Register (BSR) on the receiver side, and sets the “transmitter empty” bit (TE1) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 1 on the Processor A-side). • Any read of the BRR1 register will update all status information.

4.3.5.7 Processor B Receive Register 2 (MUB_BRR2)

Use Processor B Receive Register 2 (BRR2, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR2 register is immediately reflected in the BRR2 register.
- You can only read the BRR2 register when the RF2 bit in the BSR register is set to “1”.
- Writing to the BRR2 register generates an error response to the Processor B.

Address: 30AB_0000h base + 18h offset = 30AB_0018h



MUB_BRR2 field descriptions

Field	Description
BRR2	Processor B Receive Register 2. (Read-only) • Reflects the data written to Processor A Transmit Register 1 (ATR2). • Reading the BRR2 register clears the “receiver full” bit (RF2) in the Processor B Status Register (BSR) on the receiver side, and sets the “transmitter empty” bit (TE2) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 2 on the Processor A-side). • Any read of the BRR2 register will update all status information.

4.3.5.8 Processor B Receive Register 3 (MUB_BRR3)

Use Processor B Receive Register 3 (BRR3, 32-bit, read-only) to receive a message or data from the Processor A.

- Data written to the ATR3 register is immediately reflected in the BRR3 register.
- You can only read the BRR3 register when the RF3 bit in the BSR register is set to "1".
- Writing to the BRR3 register generates an error response to the Processor B.

Address: 30AB_0000h base + 1Ch offset = 30AB_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	BRR3																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MUB_BRR3 field descriptions

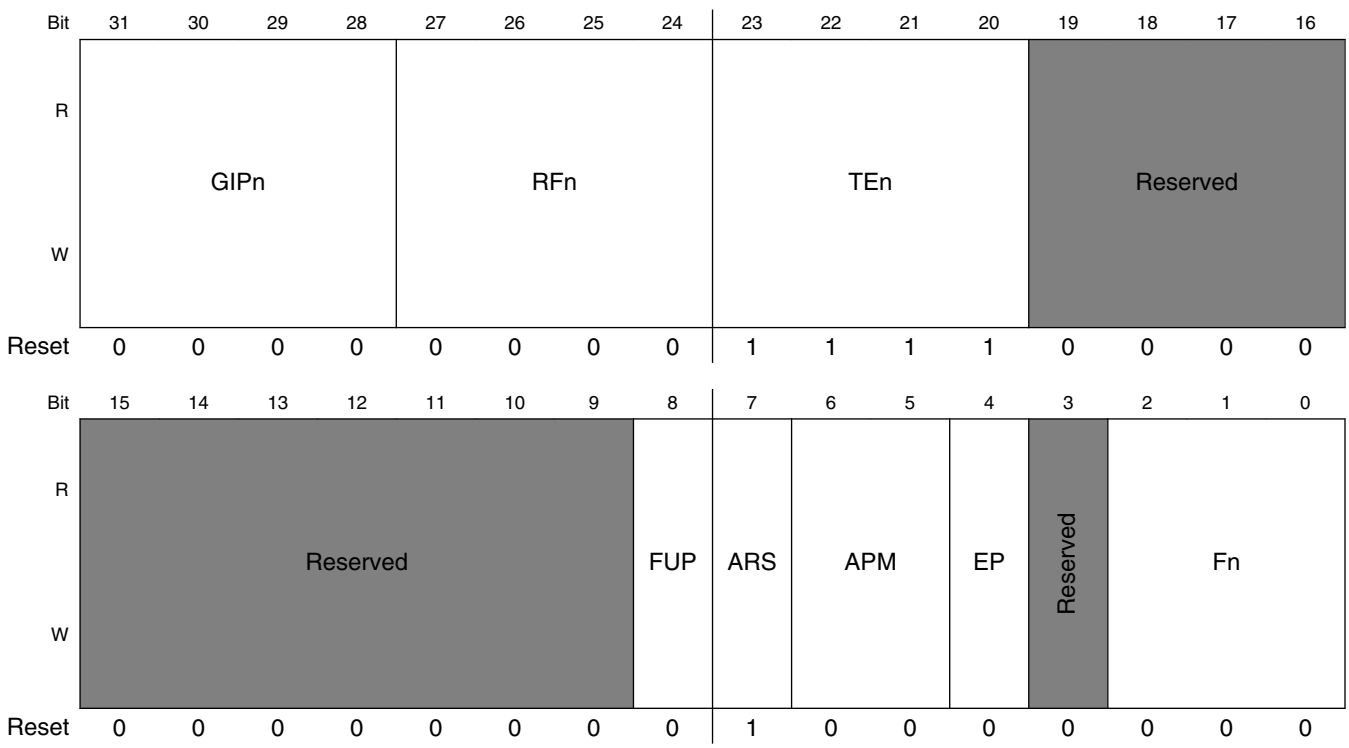
Field	Description
BRR3	Processor B Receive Register 3. (Read-only) • Reflects the data written to Processor A Transmit Register 3 (ATR3). • Reading the BRR3 register clears the "receiver full" bit (RF3) in the Processor B Status Register (BSR) on the receiver side, and sets the "transmitter empty" bit (TE3) in the Processor A Status Register on the transmitter side (optionally triggering a transmit interrupt 3 on the Processor A-side). • Any read of the BRR3 register will update all status information.

4.3.5.9 Processor B Status Register (MUB_BSR)

Use the Processor B Status Register (BSR, 32-bit, read-write) to show interrupt status from the Processor B, general purpose flags, the Processor A power mode, and to set dual function control-status bits.

- Dual-purpose bits are set by the Processor B-side programmer, and cleared by the MU logic.

Address: 30AB_0000h base + 20h offset = 30AB_0020h



MUB_BSR field descriptions

Field	Description
31–28 GIPn	For n = {0, 1, 2, 3} Processor B General Interrupt Request n Pending. (Read-Write) • GIPn bit signals the Processor B that the GIRn bit in the ACR register on the Processor A-side was set from “0” to “1”. If the GIEn bit in the BCR register is set to “1”, a General Interrupt n request is issued. • The GIPn bit is cleared by writing it back as “1”. Writing “0”, or writing “1” when the GIPn bit is cleared is ignored. Use this feature in the interrupt routine, where the GIPn bit is cleared in order to de-assert the interrupt request source at the interrupt controller. • GIPn bit is cleared when the MU is reset. 0 Processor B general purpose interrupt n is not pending. (default) 1 Processor B general purpose interrupt n is pending.

Table continues on the next page...

MUB_BSR field descriptions (continued)

Field	Description
27–24 RFn	For n = {0, 1, 2, 3} Processor B Receive Register n Full. (Read-only) • RFn bit signals to the Processor B-side that new data was written by the Processor A to the ATRn register, and is ready to be read by the Processor B in the BRRn register. • The RFn bit is set to “1” when the ATRn register is written on the Processor A-side. • After the RFn bit is set to “1”, the RFn bit signals the Processor B-side that new data is ready to be read by the Processor B in the BRRn register, and a Receive n interrupt is issued on the Processor A-side (if the RIEn bit in the BCR register has been set to “1”). • RFn bit is cleared when the BRRn register is read, and when the MU is reset. 0 BRRn register is not full (default). 1 BRRn register has received data from ATRn register and is ready to be read by the Processor B.
23–20 TEn	For n = {0, 1, 2, 3} Processor B Transmit Register n Empty. (Read-only) • When TEn = “1”, it signals to the Processor B-side that the BTRn register is ready to be written on the Processor B-side. • The TEn bit is set to “1” after the ARRn register is read on the Processor A-side. • Setting TEn bit will issue a transmit n interrupt on the Processor B-side (if the TIEn bit in the BCR register is set to “1”. • TEn bit is cleared after the BTRn register is written on the Processor B-side. • TEn bit is set to “1” when the MU is reset. 0 BTRn register is not empty. 1 BTRn register is empty (default).
19–9 Reserved	This field is reserved.
8 FUP	Processor B Flags Update Pending. (Read-only) • FUP bit is set to “1” when the Processor B-side sends a Flags Update request to the Processor A-side. • A Flags Update request is generated when the BAF[2:0] bits of the BCR register change. No flag update changes are allowed while the FUP bit is set to “1”. Any write to the BAF[2:0] bits, while the FUP bit is set to “1”, will not generate a Flags Update event, and the BAF[2:0] bits will stay unchanged. • FUP bit is cleared when this Flags Update request is internally acknowledged (that the flag is updated) from the MU Processor A-side, and during MU reset. 0 No flags updated, initiated by the Processor B, in progress (default) 1 Processor B initiated flags update, processing
7 ARS	Processor A Reset State. (Read-only) • ARS bit indicates if the Processor A-side of the MU is in a reset state or not. • If the ARS bit is set to “1”, then the Processor A-side of the MU is still in the reset state. • If the ARS bit is cleared, then both the Processor A and the Processor A-side of the MU are out of reset. • The ARS bit is set to “1” during: a Processor A system reset, or an MU reset (caused by setting the MUR bit at the BCR register).

Table continues on the next page...

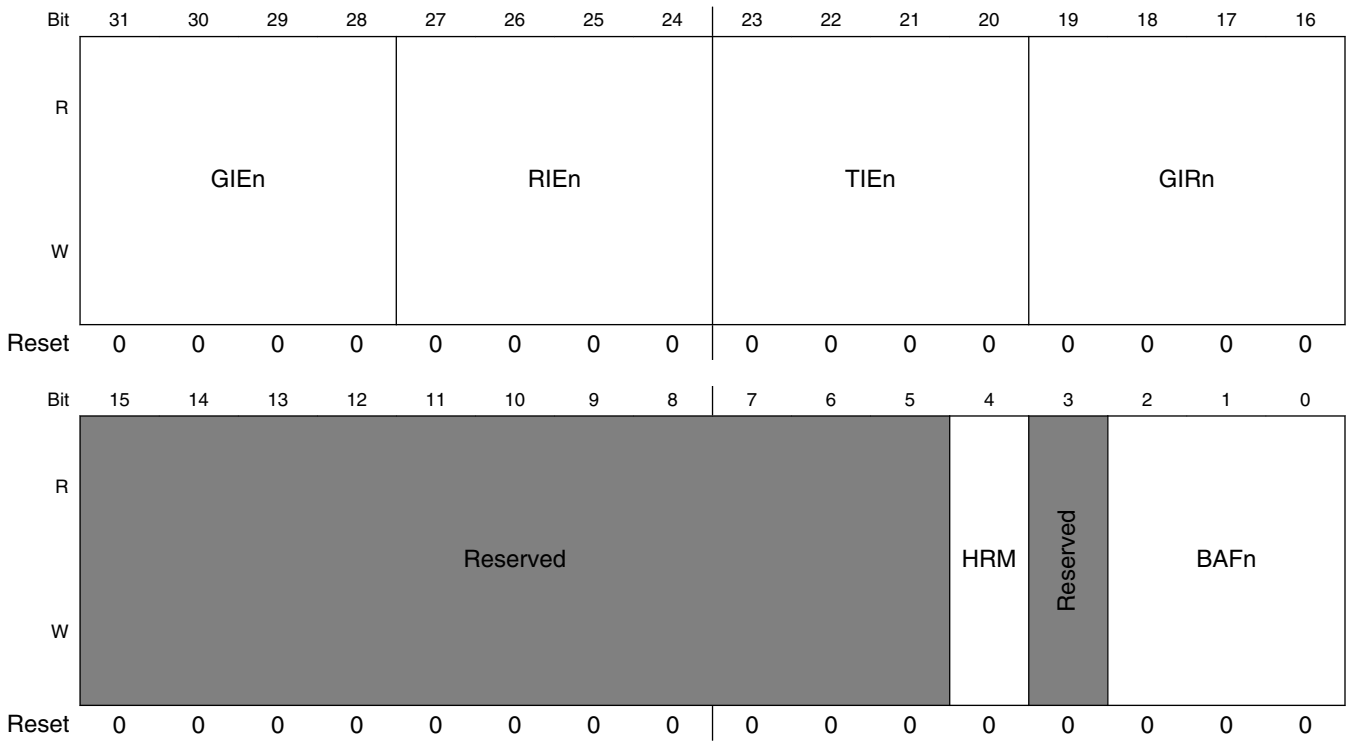
MUB_BSR field descriptions (continued)

Field	Description
	• The ARS bit is cleared when the reset sequence on the Processor A-side of the MU ends. After issuing any of the three reset events mentioned previously, you should verify that the ARS bit is cleared before starting any accesses. • When Processor B side of MU comes out of reset ARS bit has value “1”(default).Then Processor B sees the status of Processor A side and if Processor A has come out of reset then ARS bit goes low.This takes 5-6 clock cycles.And if you read ARS bit now you will see it as low although its reset value was "1" . 0 The Processor A or the Processor A-side of the MU is not in reset. 1 The Processor A or the Processor A-side of the MU is in reset.
6–5 APM	Processor A Power Mode. (Read-only) • APM[1:0] bits indicate the Processor A power mode. 00 The System is in Run Mode. 01 The System is in WAIT Mode. 10 Reserved. 11 The System is in STOP Mode.
4 EP	Processor B-Side Event Pending. (Read-only) • EP bit is set to “1” when the Processor B-side mechanism sends an event update request to the Processor A-side. • EP bit is cleared when the event update acknowledge is received. An “event” is any hardware message that is reflected in the ASR register on the Processor A-side (for example, “transmit register 0 written”). During normal operations, you do not have to deal with the state of the EP bit because the event update mechanism works automatically. • To ensure events have been posted to Processor A before entering STOP mode, you should verify that the EP bit is cleared. If EP bit is set to “1”, you should wait and continue to poll it (EP bit) before entering STOP mode. • Reading the BSR register (to check the EP bit) should be the last access to the MU that should be performed before entering STOP mode; otherwise, the EP bit may be set by subsequent additional actions. • Due to Processor B pipeline effects, three NOP operations (or their timing equivalent) should be given after an instruction that sets an event before the EP bit can reflect this event. • The EP bit is cleared when the MU resets. 0 The Processor B-side event is not pending (default). 1 The Processor B-side event is pending.
3 Reserved	This field is reserved.
Fn	For n = {0, 1, 2} Processor B-Side Flag n. (Read-only) • Fn bit is the Processor B-side flag that reflects the values written to the ABFn bit in the Processor A control register. • Every time that the ABFn bit is written, the ABFn bit write event updates the Fn bit after the event update latency, which is measured in terms of the number of clocks of the Processor A and the Processor B. 0 ABFn bit in ACR register is written 0 (default). 1 ABFn bit in ACR register is written 1.

4.3.5.10 Processor B Control Register (MUB_BCR)

Use the Processor B Control Register (BCR, 32-bit, read-write) to enable the MU interrupts on the Processor B-side, and trigger events and interrupts on the Processor A-side (wake from STOP, hardware reset, flag update).

Address: 30AB_0000h base + 24h offset = 30AB_0024h



MUB_BCR field descriptions

Field	Description
31–28 GIEn	For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Enable n. (Read-Write) • GIEn bit enables Processor B General Interrupt n. • If GIEn bit is set to “1” (enabled), then a General Interrupt n request is issued when the GIPn bit in the BSR register is set to “1”. • If GIEn is cleared (disabled), then the value of the GIPn bit is ignored and no General Interrupt n request will be issued. • GIEn bit is cleared when the MU resets. 0 Disables Processor B General Interrupt n. (default) 1 Enables Processor B General Interrupt n.
27–24 RIEn	For n = {0, 1, 2, 3} Processor B Receive Interrupt Enable n. (Read-Write) • RIEn bit enables Processor B Receive Interrupt n.

Table continues on the next page...

MUB_BCR field descriptions (continued)

Field	Description
	• If RIEn bit is set to “1” (enabled), then an Processor B Receive Interrupt n request is issued when the RFn bit in the BSR register is set to “1”. • If RIEn bit is cleared (disabled), then the value of the RFn bit is ignored and no Processor B Receive Interrupt n request will be issued. • RIEn bit is cleared when the MU resets. 0 Disables Processor B Receive Interrupt n. (default) 1 Enables Processor B Receive Interrupt n.
23–20 TIEn	For n = {0, 1, 2, 3} Processor B Transmit Interrupt Enable n. (Read-Write) • TIEn bit enables Processor B Transmit Interrupt n. • If TIEn bit is set to “1” (enabled), then an Processor B Transmit Interrupt n request is issued when the TEn bit in the BSR register is set to “1”. • If TIEn bit is cleared (disabled), then the value of the TEn bit is ignored and no Processor B Transmit Interrupt n request will be issued. • TIEn bit is cleared when the MU resets. 0 Disables Processor B Transmit Interrupt n. (default) 1 Enables Processor B Transmit Interrupt n.
19–16 GIRn	For n = {0, 1, 2, 3} Processor B General Purpose Interrupt Request n. (Read-Write) • Writing “1” to the GIRn bit sets the GIPn bit in the ASR register on the Processor A-side. If the GIEn bit in the ACR register is set to “1” on the Processor A-side, a General Purpose Interrupt n request is triggered. • The GIRn bit is cleared if the GIPn bit (in the ASR register on the Processor A-side) is cleared by writing it (GIPn bit) as “1”, thereby signalling the Processor B that the interrupt was accepted (cleared by the software). The GIPn bit cannot be written as “0” on the Processor B-side. • To ensure proper operations, you must verify that the GIRn bit is cleared (meaning that there is no pending interrupt) before setting it (GIRn bit). • GIRn bit is cleared when the MU resets. 0 Processor B General Interrupt n is not requested to the Processor A (default). 1 Processor B General Interrupt n is requested to the Processor A.
15–5 Reserved	This field is reserved.
4 HRM	Processor B Hardware Reset Mask. (Read-Write) • The Processor A can give a hardware reset to the Processor B by setting the BHR bit in the ACR Register to “1”. • When the HRM bit is set to “1” by the Processor B, the BHR reset issued by the Processor A is masked (disabled by the Processor B). • When the HRM bit is cleared, the BHR reset issued by the Processor A to the Processor B is not masked (enabled by the Processor B). 0 BHR bit in ACR is not masked, enables the hardware reset to the Processor B (default after hardware reset). 1 BHR bit in ACR is masked, disables the hardware reset request to the Processor B.
3 Reserved	This field is reserved.
BAFn	For n = {0, 1, 2} Processor B to Processor A Flag n. (Read-Write)

Table continues on the next page...

MUB_BCR field descriptions (continued)

Field	Description
	• BAFn bit is a read-write flag that is reflected in Fn bit in the ASR register on the Processor A-side. • BAFn bit is cleared when the MU resets. 0 Clears the Fn bit in the ASR register. 1 Sets the Fn bit in the ASR register.

4.4 Semaphore (SEMA4)

4.4.1 Overview

The IPS_Semaphores module provides a platform IPS slave device which implements 16 hardware-enforced gates with the following features:

- Module definition supports 16 hardware-enforced gates in a dual-processor configuration, where cp0 is core processor 0 and cp1 is core processor 1
  - Hardware gates appear as a 16-entry byte-size array with read and write accesses
    - Processors lock gates by writing "processor_number+1" to the appropriate gate and must read back the gate value to verify the lock operation was successful
    - Once locked, the gate is unlocked by a write of zeroes from the locking processor
  - Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
  - Secure reset mechanisms are supported to clear the contents of individual semaphore gates or notification logic, as well as a clear_all capability
  - Programming model allocates memory space to support up to 8 processors and up to 64 gates

A simplified block diagram of the Semaphores module is shown in Figure 4-10. In the diagram, the register blocks named gate0, gate1,…, gate 15 include the finite state machines implementing the semaphore gates plus the interrupt notification logic.

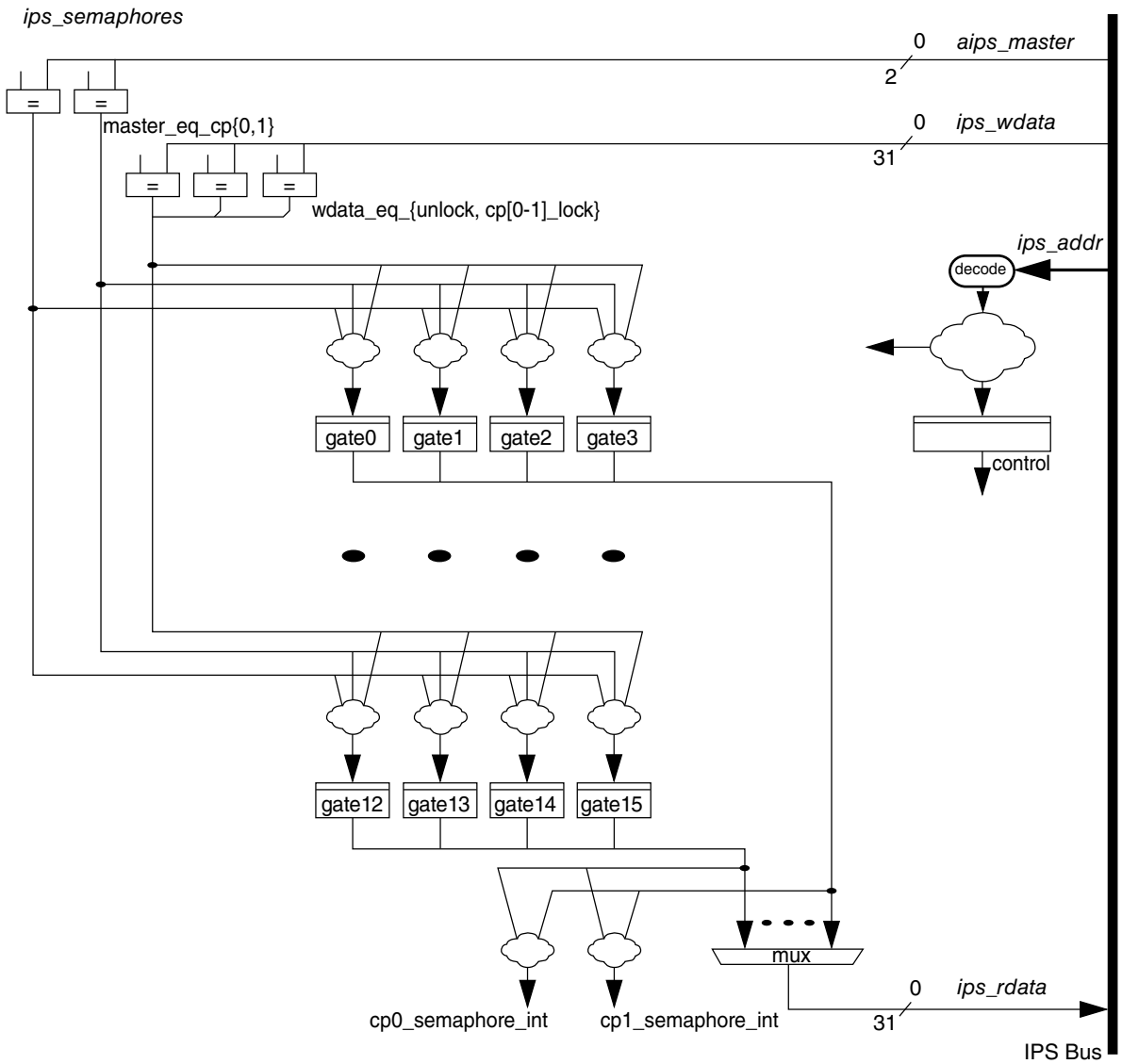


Figure 4-10. IPS_Semaphores Block Diagram

4.4.1.1 Features

The Semaphores module implements hardware-enforced semaphores as an IPS-mapped slave peripheral device. The feature set includes:

- Support for 16 hardware-enforced gates in a dual-processor configuration
  - Each hardware gate appears as a 3-state, 2-bit state machine, with all 16 gates mapped as a byte-size array
    - 3-state implementation

      if gate = 0b00, then state = unlocked
      if gate = 0b01, then state = locked by processor 0

if gate = 0b10, then state = locked by processor 1

- Uses the bus master number as a reference attribute plus the specified data patterns to validate all write operations
- Once locked, the gate can (and must) be unlocked by a write of zeroes from the locking processor

- Optional interrupt notification after a failed lock write provides a mechanism to indicate when the gate is unlocked
- Secure reset mechanisms are supported to clear the contents of individual gates or notification logic, as well as a clear_all capability

- Memory-mapped IPS slave peripheral platform module
  - Interface to the IPS bus for programming-model accesses
  - Two outputs (one per processor) for interrupt notification of failed lock writes

4.4.1.2 Modes of Operation

The Semaphores module does not support any special modes of operation. As a slave peripheral memory-mapped device located on the platform's IPS slave bus, it responds based strictly on the memory addresses of the connected bus. The IPS bus is used to access the Semaphores ' programming model.

4.4.2 External Signal Description

The Semaphores module does not include any external interfaces.

4.4.3 Functional Description

In this section, the functional operation of the Semaphores module, specifically the state machines of the SEMA4_GATEn and SEMA4_CPnNTF registers are detailed.

4.4.3.1 SEMA4_GATEn Operation

Recall each of the SEMA4_GATEn registers implements a 2-bit, 3-state machine. The state transitions for each gate are shown in the following figure.

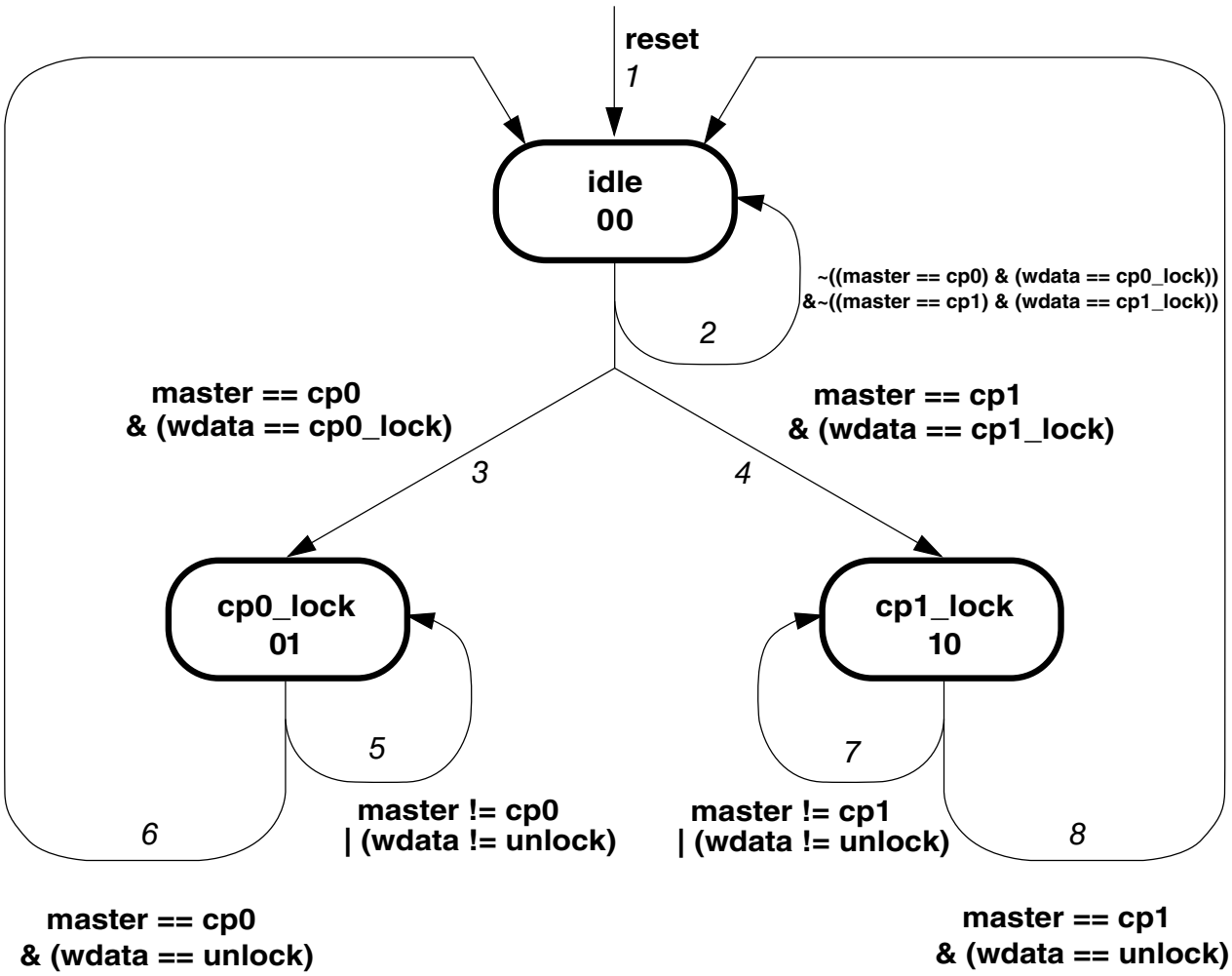


Figure 4-11. SEMA4_GATEn State Machine

The bus master number is used to identify core processor 0 (cp0) or core processor 1 (cp1). The Standard (or Reduced) Product Platform passes the AHB bus master number (`hmaster[2:0]`) through the AIPS (or AIPS-Lite) controller and drives an `aips_master[2:0]` output to the Semaphores module as an IPS sideband signal.

The state transitions for SEMA4_GATEn are defined in the following table.

Table 4-16. SEMA4_GATEn State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, whether a system reset or an individual gate reset, unconditionally forces the gate into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor occurs, the gate remains in the idle state.
idle	cp0_lock	3	When a write of the "cp0_lock" data value is initiated by processor 0, the gate transitions into the cp0_lock state.

Table continues on the next page...

Table 4-16. SEMA4_GATEn State Transitions (continued)

Current State	Next State	Transition	Description
idle	cp1_lock	4	When a write of the "cp1_lock" value is initiated by processor 1, the gate transitions into the cp1_lock state.
cp0_lock	cp0_lock	5	Once in this state, the gate remains here if any attempted write is not from cp0 with the unlock data value.
cp0_lock	idle	6	The gate returns to the idle (unlocked) state once a write from cp0 with the unlock data value occurs.
cp1_lock	cp1_lock	7	Once in this state, the gate remains here if any attempted write is not from cp1 with the unlock data value.
cp1_lock	idle	8	The gate returns to the idle (unlocked) state once a write from cp1 with the unlock data value occurs.

4.4.3.2 SEMA4_CPnNTF Operation

The failed lock write notification interrupt request is implemented in a 3-bit, 5-state machine which records failed lock attempts and transitions based on gate locking and unlocking. Two specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn]. See the following figure.

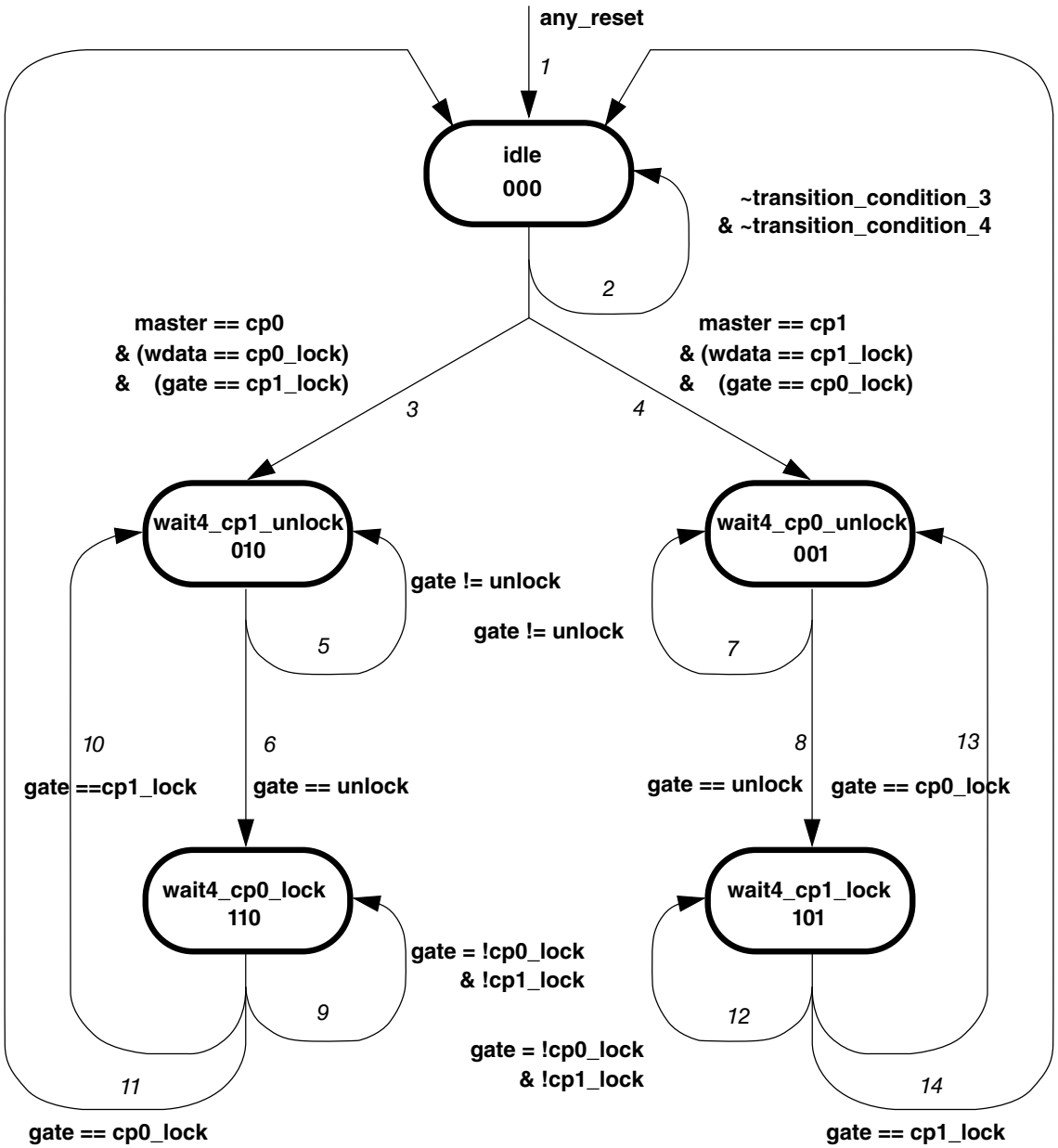


Figure 4-12. IRQ Notification State Machine

The state transitions of the IRQ notification function are defined in the following. Specific states of this machine are program-visible as the SEMA4_CPnNTF registers. In particular, two states are program-visible:

```
if state = wait4_cp0_lock (0b110) // generate cp0_semaphore_int if properly enabled
               then SEMA4_CP0NTF[GNn] = 1; else SEMA4_CP0NTF[GNn] = 0
if state = wait4_cp1_lock (0b101) // generate cp1_semaphore_int if properly enabled
               then SEMA4_CP1NTF[GNn] = 1; else SEMA4_CP1NTF[GNn] = 0
```

Table 4-17. IRQ Notification State Transitions

Current State	Next State	Transition	Description
–	idle	1	Any reset, including a system reset or an individual notification or secure gate reset, unconditionally forces the machine into the idle state.
idle	idle	2	Unless a write of the appropriate lock value from the corresponding processor to an already-locked gate occurs, the machine remains in the idle state.
idle	wait4_cp1_unlock	3	When a write of the "cp0_lock" data value is initiated by processor 0 but the gate is already locked by cp1, the machine transitions into this state, where it waits for cp1 to unlock the gate.
idle	wait4_cp0_unlock	4	When a write of the "cp1_lock" data value is initiated by processor 1but the gate is already locked by cp0, the machine transitions into this state, where it waits for cp0 to unlock the gate
wait4_cp1_unlock	wait4_cp1_unlock	5	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp1_unlock	wait4_cp0_lock	6	From this state, the machine transitions into the next state, waiting for cp0 to lock the gate, once it has been unlocked.
wait4_cp0_unlock	wait4_cp0_unlock	7	Once in this state, the machine remains here until the gate is unlocked.
wait4_cp0_unlock	wait4_cp1_lock	8	From this state, the machine transitions into the next state, waiting for cp1 to lock the gate, once it has been unlocked.
wait4_cp0_lock	wait4_cp0_lock	9	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 0 or the gate is again locked by processor 1.
wait4_cp0_lock	wait4_cp1_unlock	10	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 1. With this transition, the notification interrupt request is negated.
wait4_cp0_lock	idle	11	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	wait4_cp1_lock	12	In this state, the machine generates the notification interrupt (if properly-enabled) and remains here until the gate is locked by processor 1or the gate is again locked by processor 0.
wait4_cp1_lock	wait4_cp0_unlock	13	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is again locked by processor 0. With this transition, the notification interrupt request is negated.
wait4_cp1_lock	idle	14	In this state, the machine generates the notification interrupt (if properly-enabled) and transitions if the gate is finally locked by processor 1. With this transition, the notification interrupt request is negated.

The Semaphores module generates two interrupt request output signals, one per processor, combining the SEMA4_CPnINE and SEMA4_CPnNTF registers, where the boolean equations are:

```
cp0_semaphore_int
    =    sema4_cp0ine[ine0]    &    sema4_cp0ntf[gn0]
    |    sema4_cp0ine[ine1]    &    sema4_cp0ntf[gn1]
    |    sema4_cp0ine[ine2]    &    sema4_cp0ntf[gn2]
    ...
    |    sema4_cp0ine[ine15]    &    sema4_cp0ntf[gn15]
cp1_semaphore_int
    =    sema4_cp1ine[ine0]    &    sema4_cp1ntf[gn0]
    |    sema4_cp1ine[ine1]    &    sema4_cp1ntf[gn1]
    |    sema4_cp1ine[ine2]    &    sema4_cp1ntf[gn2]
    ...
    |    sema4_cp1ine[ine15]    &    sema4_cp1ntf[gn15]
```

4.4.4 Initialization Information

The reset state of the IPS_Semaphores module allows it to begin operation without the need for any further initialization. All the internal state machines are cleared by any reset event, allowing the module to immediately begin operation.

4.4.5 Application Information

In an operational multi-core system, most interactions involving the Semaphores module involves reads and writes to the SEMA4_GATEn registers for implementation of the hardware-enforced software gate functions. Typical code segments for gate functions perform the following operations:

- To lock (close) a gate
    - The processor performs a byte write of "logical_processor_number + 1" to gate[i]
    - The processor reads back gate[i] and checks for a value of "logical_processor_number + 1"

If the compare indicates the expected value, then the gate is locked; proceed with the protected code segment. If the compare does not indicate the expected value, the lock operation failed; repeat the process beginning with byte write to gate[i] in spin-wait loop, or proceed with another execution path and wait for failed lock interrupt notification.

A simple C-language example of a `gateLock` function is shown in the following figure. This function follows the Hennessy/Patterson example described in Multi-Core Programming 101: Software Gates.

```
#define UNLOCK     0
#define CP0_LOCK   1
#define CP1_LOCK   2

void gateLock (n)
int   n;                                          /* gate number to lock */
{
   int i;
   int current_value;
   int locked_value;

   i = processor_number();  /* obtain logical CPU number */

   if (i == 0)
      locked_value = CP0_LOCK;
   else
      locked_value = CP1_LOCK;

   /* read the current value of the gate and wait until the state == UNLOCK */
   do {
        current_value = gate[n];
   } while (current_value != UNLOCK);

   /* the current value of the gate == UNLOCK. attempt to lock the gate for this
        processor. spin-wait in this loop until gate ownership is obtained */
   do {
      gate[n] = locked_value;  /* write gate with processor_number + 1 */
      current_value = gate[n]; /* read gate to verify ownership was obtained */
   } while (current_value != locked_value);
```

Figure 4-13. Sample gateLock Function

- To unlock (open) a gate
    - After completing the protected code segment, the locking processor performs a byte write of zeroes to gate[i], opening (unlocking) the gate

A few comments on the logical CPU number are appropriate. In this example, a reference to `processor_number()` is used to retrieve this hardware configuration value. Typically, the logical processor numbers are defined by a hardwired input vector to the individual cores. The exact method for accessing the logical processor number varies by architecture. For PowerPC cores, there is a processor ID register (PIR) which is SPR 286 and contains this value. A single instruction can be used to move the contents of the PIR into a general-purpose register: mfspr rx,286 where rx is the destination GPRn. Other architectures may support a specific instruction to move the contents of the logical processor number into a general-purpose register, e.g., rdcpn rx for a "read CPU number" instruction.

If the optional failed lock IRQ notification mechanisms are used, then accesses to the related registers (SEMA4_CPnINE, SEMA4_ CPnNTF) are required. Note that there is no required negation of the failed lock write notification interrupt as the request is automatically negated by the Semaphores module once the gate has been successfully locked by the "failing" processor.

Finally, in the event a system state requires a software-controlled reset of a gate or IRQ notification register(s), accesses to the secure reset control registers (SEMA4_RSTGT, SEMA4_RSTNTF) are required. For these situations, it is recommended that the appropriate IRQ notification enable(s) (SEMA4_CPnINE) bits be disabled *before* initiating the secure reset 2-write sequence to avoid any race conditions involving spurious notification interrupt requests.

4.4.6 Memory map and register definition

The Semaphores module provides an IPS programming model mapped to an SPP-standard on-platform 16 KB space. The description here specifies a dual-core configuration with 16 semaphore gates. All the register names are prefixed with "Sema4" as an abbreviation for the full module name.

The programming model is referenced using 8-, 16- and 32-bit accesses. Reads can use any reference size, while writes are generally restricted to the size of the register. Exceptions to the write size restrictions are detailed in the individual register descriptions. Attempted references using inappropriate access sizes, to undefined (reserved) addresses, or with a non-supported access type (for example, a write to a read-only register) generate an IPS error termination.

Finally, the programming model allocates space for a definition with up to 64 gates and up to 8 processor cores, even though this definition is considerably larger than any currently-planned module implementations. The number of gates and supported processor cores are independent; there is no relationship between these two system variables.

The 16 KB Semaphores programming model map is shown in the following table.

SEMA4 memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AC_0000	Semaphores Gate 0 Register (SEMA4_Gate00)	8	R/W	00h	4.4.6.1/166
30AC_0001	Semaphores Gate 1 Register (SEMA4_Gate01)	8	R/W	00h	4.4.6.2/167
30AC_0002	Semaphores Gate 2 Register (SEMA4_Gate02)	8	R/W	00h	4.4.6.3/168
30AC_0003	Semaphores Gate 3 Register (SEMA4_Gate03)	8	R/W	00h	4.4.6.4/169
30AC_0004	Semaphores Gate 4 Register (SEMA4_Gate04)	8	R/W	00h	4.4.6.5/170
30AC_0005	Semaphores Gate 5 Register (SEMA4_Gate05)	8	R/W	00h	4.4.6.6/171
30AC_0006	Semaphores Gate 6 Register (SEMA4_Gate06)	8	R/W	00h	4.4.6.7/172
30AC_0007	Semaphores Gate 7 Register (SEMA4_Gate07)	8	R/W	00h	4.4.6.8/173

Table continues on the next page...

SEMA4 memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30AC_0008	Semaphores Gate 8 Register (SEMA4_Gate08)	8	R/W	00h	4.4.6.9/174
30AC_0009	Semaphores Gate 9 Register (SEMA4_Gate09)	8	R/W	00h	4.4.6.10/ 175
30AC_000A	Semaphores Gate 10 Register (SEMA4_Gate10)	8	R/W	00h	4.4.6.11/ 176
30AC_000B	Semaphores Gate 11 Register (SEMA4_Gate11)	8	R/W	00h	4.4.6.12/ 177
30AC_000C	Semaphores Gate 12 Register (SEMA4_Gate12)	8	R/W	00h	4.4.6.13/ 178
30AC_000D	Semaphores Gate 13 Register (SEMA4_Gate13)	8	R/W	00h	4.4.6.14/ 179
30AC_000E	Semaphores Gate 14 Register (SEMA4_Gate14)	8	R/W	00h	4.4.6.15/ 180
30AC_000F	Semaphores Gate 15 Register (SEMA4_Gate15)	8	R/W	00h	4.4.6.16/ 181
30AC_0040	Semaphores Processor n IRQ Notification Enable (SEMA4_CP0INE)	16	R/W	0000h	4.4.6.17/ 182
30AC_0048	Semaphores Processor n IRQ Notification Enable (SEMA4_CP1INE)	16	R/W	0000h	4.4.6.17/ 182
30AC_0080	Semaphores Processor n IRQ Notification (SEMA4_CP0NTF)	16	R	0000h	4.4.6.18/ 184
30AC_0088	Semaphores Processor n IRQ Notification (SEMA4_CP1NTF)	16	R	0000h	4.4.6.18/ 184
30AC_0100	Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)	16	R/W	0000h	4.4.6.19/ 186
30AC_0104	Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)	16	R/W	0000h	4.4.6.20/ 187

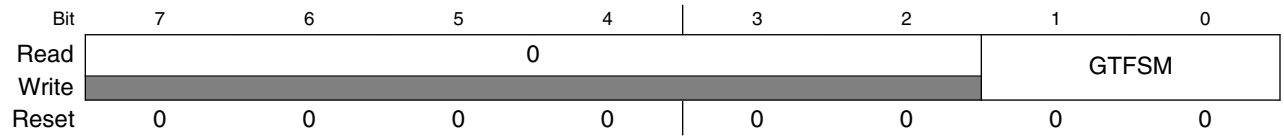
4.4.6.1 Semaphores Gate 0 Register (SEMA4_Gate00)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate

register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 0h offset = 30AC_0000h



SEMA4_Gate00 field descriptions

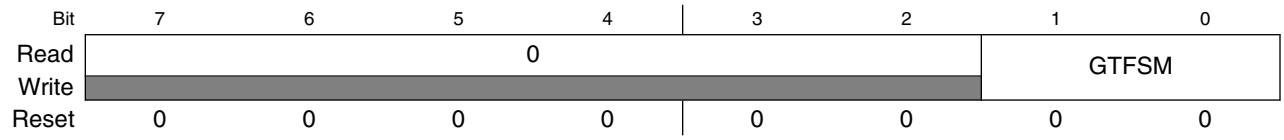
Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.2 Semaphores Gate 1 Register (SEMA4_Gate01)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 1h offset = 30AC_0001h



SEMA4_Gate01 field descriptions

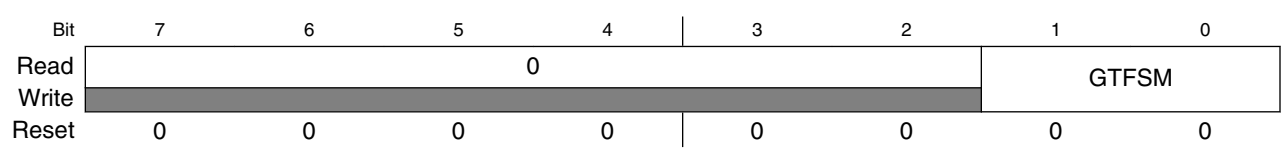
Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.3 Semaphores Gate 2 Register (SEMA4_Gate02)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 2h offset = 30AC_0002h



SEMA4_Gate02 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.4 Semaphores Gate 3 Register (SEMA4_Gate03)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 3h offset = 30AC_0003h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate03 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine.

Table continues on the next page...

SEMA4_Gate03 field descriptions (continued)

Field	Description
	Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.5 Semaphores Gate 4 Register (SEMA4_Gate04)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 4h offset = 30AC_0004h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate04 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation .

Table continues on the next page...

SEMA4_Gate04 field descriptions (continued)

Field	Description
	NOTE: The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.6 Semaphores Gate 5 Register (SEMA4_Gate05)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 5h offset = 30AC_0005h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate05 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free).

Table continues on the next page...

SEMA4_Gate05 field descriptions (continued)

Field	Description
	01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.7 Semaphores Gate 6 Register (SEMA4_Gate06)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 6h offset = 30AC_0006h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate06 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.8 Semaphores Gate 7 Register (SEMA4_Gate07)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 7h offset = 30AC_0007h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate07 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.9 Semaphores Gate 8 Register (SEMA4_Gate08)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 8h offset = 30AC_0008h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate08 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.10 Semaphores Gate 9 Register (SEMA4_Gate09)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + 9h offset = 30AC_0009h

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate09 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.11 Semaphores Gate 10 Register (SEMA4_Gate10)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Ah offset = 30AC_000Ah

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate10 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.12 Semaphores Gate 11 Register (SEMA4_Gate11)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Bh offset = 30AC_000Bh

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate11 field descriptions

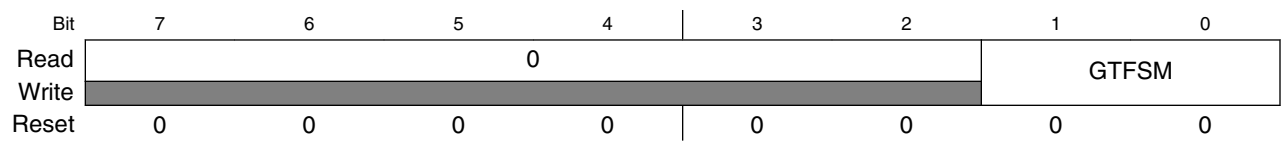
Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.13 Semaphores Gate 12 Register (SEMA4_Gate12)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Ch offset = 30AC_000Ch



SEMA4_Gate12 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.14 Semaphores Gate 13 Register (SEMA4_Gate13)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Dh offset = 30AC_000Dh

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate13 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.15 Semaphores Gate 14 Register (SEMA4_Gate14)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Eh offset = 30AC_000Eh

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate14 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.16 Semaphores Gate 15 Register (SEMA4_Gate15)

Each semaphore gate is implemented in a 2-bit finite state machine, right-justified in a byte data structure. The hardware uses the bus master number in conjunction with the data patterns to validate all attempted write operations. Only processor bus masters can modify the gate registers. Once locked, a gate can (and must) be opened (unlocked) by the locking processor core.

Multiple gate values can be read in a single access, but only a single gate can be updated via a write operation at a time. 16- and 32-bit writes to multiple gates are allowed, but the write data operand must only update the state of a single gate. A byte write data value of 0x03 is defined as "no operation" and does not affect the state of the corresponding gate register. Attempts to write multiple gates in a single aligned access with a size larger than an 8-bit (byte) reference generate an error termination and do not allow any gate state changes

Address: 30AC_0000h base + Fh offset = 30AC_000Fh

Bit	7	6	5	4	3	2	1	0
Read	0						GTFSM	
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_Gate15 field descriptions

Field	Description
7–2 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
GTFSM	Gate Finite State Machine. Gate Finite State Machine. The hardware gate is maintained in a 3-state implementation-unlocked, locked by processor 0 or locked by processor 1. For more details, see SEMA4_GATEn Operation . **NOTE:** The state of the gate reflects the last processor that locked it, which can be useful during system debug. 00 The gate is unlocked (free). 01 The gate has been locked by processor 0. 10 The gate has been locked by processor 1. 11 This state encoding is never used and therefore reserved. Attempted writes of 0x03 are treated as "no operation" and do not affect the gate state machine.

4.4.6.17 Semaphores Processor n IRQ Notification Enable (SEMA4_CP*n*INE)

The application of a hardware semaphore module provides an opportunity for implementation of helpful system-level features. An example is an optional mechanism to generate a processor interrupt after a failed lock attempt. Recall traditional software gate functions execute a spin-wait loop in an effort to obtain and lock the referenced gate. With this module, the processor that fails in the lock attempt could continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function.

The optional notification interrupt function consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnINE registers, unimplemented bits read as zeroes, and writes are ignored.

Address: 30AC_0000h base + 40h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read Write	INE8	INE9	INE10	INE11	INE12	INE13	INE14	INE15
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	INE0	INE1	INE2	INE3	INE4	INE5	INE6	INE7
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*INE field descriptions

Field	Description
15 INE8	Interrupt Request Notification Enable 8. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 8. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
14 INE9	Interrupt Request Notification Enable 9. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 9. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
13 INE10	Interrupt Request Notification Enable 10. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 10. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
12 INE11	Interrupt Request Notification Enable 11. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 11.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
11 INE12	Interrupt Request Notification Enable 12. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 12. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
10 INE13	Interrupt Request Notification Enable 13. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 13. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
9 INE14	Interrupt Request Notification Enable 14. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 14. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
8 INE15	Interrupt Request Notification Enable 15. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 15. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
7 INE0	Interrupt Request Notification Enable 0. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 0. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
6 INE1	Interrupt Request Notification Enable 1. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 1. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
5 INE2	Interrupt Request Notification Enable 2. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 2. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
4 INE3	Interrupt Request Notification Enable 3. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 3. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
3 INE4	Interrupt Request Notification Enable 4. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 4. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
2 INE5	Interrupt Request Notification Enable 5. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 5.

Table continues on the next page...

SEMA4_CP*n*INE field descriptions (continued)

Field	Description
	0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
1 INE6	Interrupt Request Notification Enable 6. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 6. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.
0 INE7	Interrupt Request Notification Enable 7. This field is a bitmap to enable the generation of an interrupt notification from a failed attempt to lock gate 7. 0 The generation of the notification interrupt is disabled. 1 The generation of the notification interrupt is enabled.

4.4.6.18 Semaphores Processor n IRQ Notification (SEMA4_CP*n*NTF)

The Semaphores module optionally allows the processor that fails in the lock attempt to continue with other tasks and allow a properly-enabled notification interrupt to return its execution to the original lock function rather than simply execute in a spin-wait loop.

The optional notification interrupt mechanism consists of two registers for each processor: an interrupt notification enable register (SEMA4_CPnINE) and the read-only notification interrupt request register (SEMA4_CPnNTF). To support implementations with more than 16 gates, these registers can be referenced with aligned 16- or 32-bit accesses. For the SEMA4_CPnNTF registers, unimplemented bits read as zeroes.

The notification interrupt is generated via a unique finite state machine, one per hardware gate. This machine operates in the following manner:

1. When an attempted lock fails, the FSM enters a first state where it waits until the gate is unlocked.
2. Once unlocked, the FSM enters a second state where it generates an interrupt request to the “failed lock” processor.
3. When the “failed lock” processor succeeds in locking the gate, the IRQ is automatically negated and the FSM returns to the idle state. However, if the other processor again locks the gate, the FSM returns to the first state, negates the interrupt request, and then waits for the gate to be unlocked (again).

The notification interrupt request is implemented in a 3-bit, 5-state machine, where tw specific states are encoded and program-visible as SEMA4_CP0NTF[GNn] and SEMA4_CP1NTF[GNn].

Address: 30AC_0000h base + 80h offset + (8d × i), where i=0d to 1d

Bit	15	14	13	12	11	10	9	8
Read	GN8	GN9	GN10	GN11	GN12	GN13	GN14	GN15
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	GN0	GN1	GN2	GN3	GN4	GN5	GN6	GN7
Write								
Reset	0	0	0	0	0	0	0	0

SEMA4_CP*n*NTF field descriptions

Field	Description
15 GN8	Gate 8 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 8. For more details, see SEMA4_CPnNTF Operation .
14 GN9	Gate 9 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 9. For more details, see SEMA4_CPnNTF Operation .
13 GN10	Gate 10 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 10. For more details, see SEMA4_CPnNTF Operation .
12 GN11	Gate 11 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 11. For more details, see SEMA4_CPnNTF Operation .
11 GN12	Gate 12 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 12. For more details, see SEMA4_CPnNTF Operation .
10 GN13	Gate 13 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 13. For more details, see SEMA4_CPnNTF Operation .
9 GN14	Gate 14 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 14. For more details, see SEMA4_CPnNTF Operation .
8 GN15	Gate 15 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 15. For more details, see SEMA4_CPnNTF Operation .
7 GN0	Gate 0 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 0. For more details, see SEMA4_CPnNTF Operation .
6 GN1	Gate 1 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 1. For more details, see SEMA4_CPnNTF Operation .
5 GN2	Gate 2 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 2. For more details, see SEMA4_CPnNTF Operation .
4 GN3	Gate 3 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 3. For more details, see SEMA4_CPnNTF Operation .
3 GN4	Gate 4 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 4. For more details, see SEMA4_CPnNTF Operation .
2 GN5	Gate 5 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 5. For more details, see SEMA4_CPnNTF Operation .
1 GN6	Gate 6 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 6. For more details, see SEMA4_CPnNTF Operation .
0 GN7	Gate 7 Notification. This read-only field is a bitmap of the interrupt request notification from a failed attempt to lock gate 7. For more details, see SEMA4_CPnNTF Operation .

4.4.6.19 Semaphores (Secure) Reset Gate n (SEMA4_RSTGT)

Although the intent of the hardware gate implementation specifies a protocol where the locking processor must unlock the gate, it is recognized that system operation may require a reset function to re-initialize the state of any gate(s) without requiring a system-level reset.

To support this special gate reset requirement, the Semaphores module implements a "secure" reset mechanism which allows a hardware gate (or all the gates) to be initialized by following a specific dual-write access pattern. Using a technique similar to that required for the servicing of a software watchdog timer, the secure gate reset requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the specified gate(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTGT memory location. The most significant byte (SEMA4_RSTGT[RSTGDP]) must be 0xe2; the least significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTGT location. For this write, the upper byte (SEMA4_RSTGT[RSTGDP]) is the logical complement of the first data pattern (0x1d) and the lower byte (SEMA4_RSTGT[RSTGTN]) specifies the gate(s) to be reset. This gate field can specify a single gate be cleared, or that all gates are cleared.
3. Reads of the SEMA4_RSTGT location return information on the 2-bit state machine (SEMA4_RSTGT[RSTGSM]) which implements this function, the bus master performing the reset (SEMA4_RSTGT[RSTGMS]) and the gate number(s) last cleared (SEMA4_RSTGT[RSTGTN]). Reads of the SEMA4_RSTGT register do not affect the secure reset finite state machine in any manner.

Address: 30AC_0000h base + 100h offset = 30AC_0100h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTGTN								RSTGSM_RSTGMS_RSTGDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTGT field descriptions

Field	Description
15–8 RSTGTN	Reset Gate Number. This 8-bit field specifies the specific hardware gate to be reset. This field is updated by the second write. If RSTGTN < 64, then reset the single gate defined by RSTGTN, else reset all the gates. The corresponding secure IRQ notification state machine(s) are also reset.
RSTGSM_ RSTGMS_ RSTGDP	**NOTE:** This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having

Table continues on the next page...

SEMA4_RSTGT field descriptions (continued)

<table>
<tr><th>Field</th><th>Description</th></tr>
<tr><td></td><td>write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.
<table>
<tr><th>Access</th><th>Sub-Field</th><th>Description</th></tr>
<tr><td rowspan="4">Read-Only</td><td>7-6
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>5-4
RSTGSM</td><td>Reset Gate Finite State Machine. Reads of the SEMA4_RSTGT register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as:
00 Idle, waiting for the first data pattern write.
01 Waiting for the second data pattern write.
10 The 2-write sequence has completed. Generate the specified gate reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note that the RSTGSM = 0b10 state is valid for only a single machine cycle, so it is impossible for a read to return this value
11 This state encoding is never used and therefore reserved.</td></tr>
<tr><td>3
Reserved</td><td>Reserved. Always reads 0.</td></tr>
<tr><td>2-0
RSTGMS</td><td>Reset Gate Bus Master. This 3-bit read-only field records the logical number of the bus master performing the gate reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.</td></tr>
<tr><td>Write-Only</td><td>7-0
RSTGDP</td><td>Reset Gate Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the gate reset mechanism. For the first write, RSTGDP = 0xe2 while the second write requires RSTGDP = 0x1d.</td></tr>
</table></td></tr>
</table>

4.4.6.20 Semaphores (Secure) Reset IRQ Notification (SEMA4_RSTNTF)

As with the case of the secure reset function and the hardware gates, it is recognized that system operation may require a reset function to re-initialize the state of the IRQ notification logic without requiring a system-level reset.

To support this special notification reset requirement, the Semaphores module implements a "secure" reset mechanism which allows an IRQ notification (or all the notifications) to be initialized by following a specific dual-write access pattern. When successful, the specified IRQ notification state machine(s) are reset. Using a technique

similar to that required for the servicing of a software watchdog timer, the secure reset mechanism requires two consecutive writes with predefined data patterns from the same processor to force the clearing of the IRQ notification(s). The required access pattern is:

1. A processor performs a 16-bit write to the SEMA4_RSTNTF memory location. The most significant byte (SEMA4_RSTNTF[RSTNDP]) must be 0x47; the least significant byte is a "don't_care" for this reference.
2. The same processor then performs a second 16-bit write to the SEMA4_RSTNTF location. For this write, the upper byte (SEMA4_RSTNTF[RSTNDP]) is the logical complement of the first data pattern (0xb8) and the lower byte (SEMA4_RSTNTF[RSTNTN]) specifies the notification(s) to be reset. This field can specify a single notification be cleared, or that all notifications are cleared.
3. Reads of the SEMA4_RSTNTF location return information on the 2-bit state machine (SEMA4_RSTNTF[RSTNSM]) which implements this function, the bus master performing the reset (SEMA4_RSTNTF[RSTNMS]) and the notification number(s) last cleared (SEMA4_RSTNTF[RSTNTN]). Reads of the SEMA4_RSTNTF register do not affect the secure reset finite state machine in any manner.

Address: 30AC_0000h base + 104h offset = 30AC_0104h

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	RSTNTN								RSTNSM_RSTNMS_RSTNDP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SEMA4_RSTNTF field descriptions

Field	Description
15–8 RSTNTN	Reset Notification Number. This 8-bit field specifies the specific IRQ notification state machine to be reset. This field is updated by the second write. If RSTNTN < 64, then reset the single IRQ notification machine defined by RSTNTN, else reset all the notifications.
RSTNSM_ RSTNMS_ RSTNDP	**NOTE:** This field contains subfields that vary depending on whether it is being read or written. Sub-fields indicated as having read access are valid only for read operations. Sub-fields indicated as having write access are valid only for write operations. Bit numbering in the descriptions begins with the most significant bit numbered 0. See the following table for details.

Access	Sub-Field	Description
Read-Only	7-6 Reserved	Reserved. Always reads 0.
	5-4 RSTNSM	Reset Notification Finite State Machine. Reads of the SEMA4_RSTNTF register return the encoded state machine value. The reset state machine is maintained in a 2-bit, 3-state implementation, defined as: 00 Idle, waiting for the first data pattern write. 01 Waiting for the second data pattern write.

Table continues on the next page...

SEMA4_RSTNTF field descriptions (continued)

Field	Description		
	Access	**Sub-Field**	**Description**
			10 The 2-write sequence has completed. Generate the specified notification reset(s). After the reset is performed, this machine returns to the idle (waiting for first data pattern write) state. Note the RSTNSM = 10 state is valid for only a single machine cycle, so it is impossible for a read to return this value. 11 This state encoding is never used and therefore reserved..
		3 Reserved	Reserved. Always reads 0.
		2-0 RSTNMS	Reset Notification Bus Master. This 3-bit read-only field records the logical number of the bus master performing the notification reset function. The reset function requires that the two consecutive writes to this register be initiated by the same bus master to succeed. This field is updated each time a write to this register occurs. The association between system bus master port numbers, the associated bus master device and the logical processor number is SoC-specific. See the chip configuration chapter for this information.
	Write-Only	7-0 RSTNDP	Reset Notification Data Pattern. This write-only field is accessed with the specified data patterns on the two consecutive writes to enable the notification reset mechanism. For the first write, RSTNDP = 0x47 while the second write requires RSTNDP = 0xb8.

4.5 On-Chip RAM Memory Controller (OCRAM)

4.5.1 Overview

Various options are provided for adding a pipeline or wait-states in a read/write access, in order to ensure flexible timing control at both high and low frequencies.

The internal block diagram is shown in the figure below.

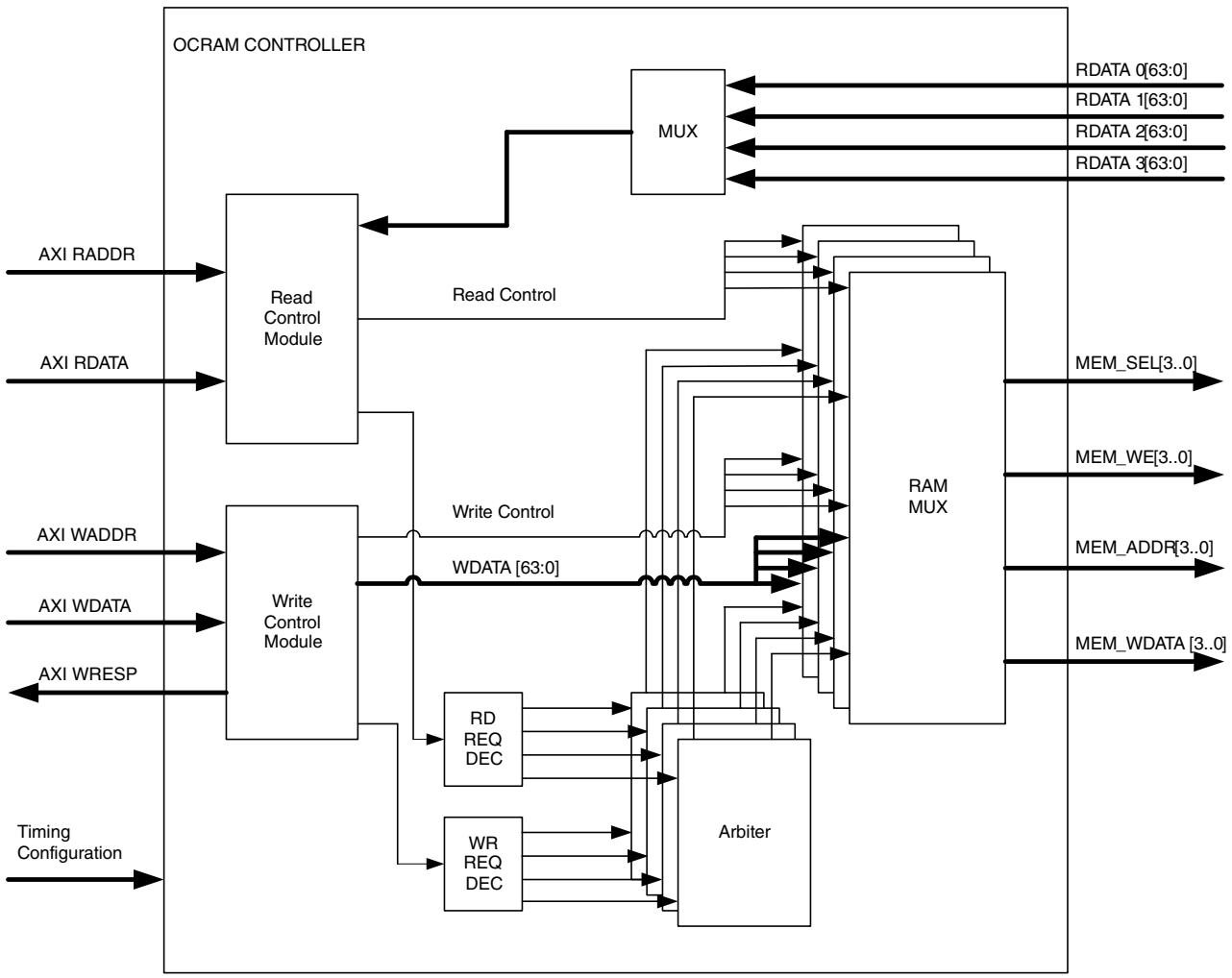


Figure 4-14. On-chip RAM Block Diagram

4.5.2 Basic Functions

4.5.2.1 Read/Write Arbitration

The detailed rules used in arbitration are as follows:

- If there is no granted read or write in the last cycle, and there is only a read request or a write request, the request will be granted.
- If there is no granted read or write in the last cycle, and there are both read or write requests coming in at the same time, the read request will be granted first.

- If a granted read/write transaction has just finished, the write/read request will have the higher priority in the next cycle.
- If the first read/write access request in a transaction is granted, all the data transfer in this burst will be finished before the next arbitration begins, that is, the round-robin arbitration mechanism is based on AXI transaction, not data access.

4.5.3 Advanced Features

This section describes some advanced features designed to avoid timing issues when the on-chip RAM is working at high frequency.

4.5.3.1 Read Data Wait State

When the wait state is enabled, it will take 2 cycles for each read access (each beat of a read burst).

This can avoid the potential timing problem caused by the longer memory access time at higher frequency.

When this feature is disabled, it only takes 1 clock cycle to finish a read transaction. That is, read data is available in the next cycle of read request becomes valid on the bus.

4.5.3.2 Read Address Pipeline

When this feature is enabled, the read address from the AXI master is delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issues for the read access on the memory cell at high frequency. Enabling this feature can cost, at most, 1 more clock cycle for each AXI read transaction, that is, at most 1 more clock cycle for each read burst with multiple beats of data.

When this feature is disabled, the read address from the AXI master can be accepted by the on-chip RAM without delay, and data can become ready for master at next clock cycle (if no other access and no read data wait).

4.5.3.3 Write Data Pipeline

When this feature is enabled, the write data from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enabling this feature would cost at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write data from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle).

4.5.3.4 Write Address Pipeline

When this feature is enabled, the write address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM.

This can avoid setup time issue for the write access on the memory cell at high frequency. Enabling this feature would take at most 1 more clock cycle for each AXI write transaction, that is, at most 1 more clock cycle for each write burst with multiple beats of data.

When this feature is disabled, the write address from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle).

4.5.4 Programmable Registers

There are no programmable registers in this block;

4.6 Network Interconnect Bus System (NOC)

4.6.1 Overview

This section provides an overview of the on chip Network Interconnection Bus System. The Bus System is composed of two kinds of IP

- NIC-301 (Network Inter-Connect) AXI arbiter IP

  The NIC-301 (by Arm Ltd.) is a configurable AXI arbiter between several masters and slaves. The NIC-301 IP is designed so that many configuration options are

selected at the hardware design stage, determined by SoC characteristics and needs, while several other configuration options are software-controlled.

- NoC (Network on Chip) fabric IP

  The NoC (by Arteris Ltd.) is a configurable high efficiency and performance fabric IP. Similar to the NIC-301 IP the majority design options are configured during hardware design stage. Mean while those application specific configurations are required to be configured by software.

NOTE

The NIC-301 and NoC default settings are configured by NXP's board support package (BSP), and in most cases should not be modified by the customer. The default settings have gone through exhaustive testing during the validation of the part, and have proven to work well for the part's intended target applications. Changes to the default settings may result in a degradation in system performance.

4.6.2 External Signals

There are no external I/O interfaces for NIC-301 and NoC.

4.7 AHB to IP Bridge (AIPSTZ)

4.7.1 Overview

This section provides an overview of the AHB to IP Bridge (AIPSTZ). This particular peripheral is designed as the bridge between AHB bus and peripherals with the lower bandwidth IP Slave (IPS) buses.

4.7.1.1 Features

The following list summarizes the key features of the bridge:

- The bridge supports the IPS slave bus signals.
- The bridge supports 8-, 16-, and 32-bit IPS peripherals. (Accesses larger than the size of a peripheral are not supported, except to 32-bit memory.)
- The bridge supports a pair of IPS accesses for 64-bit and certain misaligned AHB transfers to 32-bit memory in 64-bit platforms.

- The bridge directly supports up to 32 64-Kbyte external IPS peripherals, and 2 global external IPS peripheral spaces. The bridge occupies 1 MBytes of total address space.
- The bridge provides configurable per-block and per-master access protections. Access permissions are based on bus master (e.g. DMA or core) privilege levels and resource domain. More details on the protection features and configuration can be found in the Security Reference Manual
- Peripheral read transactions require a minimum of 2 hclk clocks, and unbuffered write transactions require a minimum of 3 hclk clocks.
- The bridge uses one single asynchronous reset and one global clock.

4.7.2 Clocks

The following table describes the clock sources for AIPSTZ. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 4-18. AIPSTZ Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	Module clock

4.7.3 Functional Description

The AIPS bridge serves as a protocol translator between the AHB system bus and the IP bus.

Support is provided for generating a pair of 32-bit IP bus accesses when targeted by a 64-bit system bus access, or a misaligned access which crosses a 32-bit boundary. No other bus-sizing access support is provided.

The AHB to IP bridge is the interface between the AHB and on-chip IPS peripherals, which are sub-blocks containing readable/writable control and status registers.

The AHB master reads and writes these registers through the AIPSTZ. The bridge generates block enables, the block address, transfer attributes, byte enables and write data as inputs to the IPS peripherals. The bridge captures read data from the IPS interface and drives it on the AHB.

Each bridge that connects to the IPS (or peripherals) are referred as AIPS. The chip has several separate AIPS modules, and peripherals are grouped and assigned under each AIPS block. The list of peripherals are indicated as n-1, ... and n-*x* for AIPS-1, ... and AIPS-*x* respectively.

AIPS occupies a 1-Mbyte portion of the address space. The register maps of the IPS peripherals are located on 64-Kbyte boundaries. Each IPS peripheral is allocated one 64-Kbyte block of the memory map, and is activated by one of the block enables from the bridge. Up to thirty-two 64-Kbyte external IPS peripherals may be implemented, occupying contiguous blocks of 64-Kbytes. Two global external IPS block enables are available for the remaining address space to allow for customization and expansion of addressed peripheral devices. In addition, a single "non-global" block enable is also asserted whenever any of the thirty-two non-global block enables is asserted.

The bridge is responsible for indicating to IPS peripherals if an access is in supervisor or user mode. It may block user mode accesses to certain IPS peripherals or it may allow the individual IPS peripherals to determine if user mode accesses are allowed. In addition, peripherals may be designated as write-protected.

The bridge supports the notion of "trusted" masters for security purposes. Masters may be individually designated as trusted for reads, trusted for writes, or trusted for both reads and writes, as well as being forced to look as though all accesses from a master are in user-mode privilege level. Refer to AIPSTZ Memory Map/Register Definition for more information.

The AIPSTZ prevents access to a peripheral if the transaction originated from a source from a resource domain that has been explicitly omitted. Resource domains are assigned in the RDC submodule. Please refer to the RDC chapter for programming details.

All peripheral devices are expected to only require aligned accesses equal to or smaller in size than the peripheral size. An exception to this rule is supported for 32-bit peripherals to allow memory to be placed on the IPS.

4.7.4 Access Protections

The AIPSTZ bridge provides programmable access protections for both masters and peripherals. It allows the privilege level of a master to be overridden, forcing it to user-mode privilege, and allows masters to be designated as trusted or untrusted.

Peripherals may require supervisor privilege level for access, may restrict access to a trusted master only, and may be write-protected. IP bus peripherals are subject to access control policies set in both CSU registers and AIPSTZ registers. An access is blocked if it is denied by either policy.

Masters and peripherals are assigned to one or more resource domains in the RDC submodule (see the RDC chapter for details). Depending on RDC programming, masters transactions through the AIPSTZ may or may not be allowed access to peripherals in different resource domains.

4.7.5 Access Support

Aligned 64-bit accesses, aligned and misaligned word and half word accesses, as well as byte accesses are supported for 32-bit peripherals. Misaligned accesses are supported to allow memory to be placed on the IPS.

Peripheral registers must not be misaligned, although no explicit checking is performed by the AIPS bridge. The bridge will perform two IPS transfers for 64-bit accesses, word accesses with byte offsets of 1, 2, or 3, and for half word accesses with a byte offset of 3. All other accesses will be performed with a single IPS transfer.

Only aligned half word and byte accesses are supported for 16-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

Only byte accesses are supported for 8-bit peripherals. All other accesses types are unsupported, and results of such accesses are undefined. They are not terminated with an error response.

4.7.6 Initialization Information

The AIPS bridge should be programmed before use.

The following registers should be initialized: The Master Privilege Registers (AIPSTZ_MPRs), the Peripheral Access Control registers (AIPSTZ_PACRs), and the Off-platform Peripheral Access Control registers (AIPSTZ_OPACRs) described in AIPSTZ Memory Map/Register Definition.

4.7.6.1 Security Block

The AIPSTZ contains a security block that is connected to each off-platform peripheral. This block filters accesses based on write/read, non-secure, and supervisor signals.

Each peripheral can be individually configured to allow or deny each of the following transactions as described in the table below:

Table 4-19. Peripheral Access Configuration options

Config Bit	Write	Non-Secure	Supervisor	Meaning
0	0	0	0	Secure User Read
1	0	0	1	Secure Supervisor Read

Table continues on the next page...

Table 4-19. Peripheral Access Configuration options (continued)

Config Bit	Write	Non-Secure	Supervisor	Meaning
2	0	1	0	Non-Secure User Read
3	0	1	1	Non-Secure Supervisor Read
4	1	0	0	Secure User Write
5	1	0	1	Secure Supervisor Write
6	1	1	0	Non-Secure User Write
7	1	1	1	Non-Secure Supervisor Write

Each peripheral has a security configuration (sec_config_X) input for determining whether to allow or deny a given access type. These are 8-bit vectors, with each bit corresponding to one of the transactions above as listed in the Config Bit column of Table 4-19. If the bit is asserted (1'b1), the transaction is allowed. If the bit is negated (1'b0), the transaction is not allowed.

For example, if peripheral 0 is configured as follows:

sec_config_0 [7:0] = 8'b0011_0011

This peripheral can only be accessed by secure transactions. Bits 0, 1, 4, and 5 are asserted and these bits refer to the four types of secure transactions. If an insecure transaction is attempted to this peripheral, it will result in an error.

Eight bits per peripheral across an entire system can result in a large number of configuration bits that must be assigned and controlled, most likely in a series of registers in another block. To reduce the number of register bits required predefined sets of security profiles can be defined and encapsulated in an external security translation block. The table below describes one set of security profiles that has been proposed for use with the AIPSTZ.

Table 4-20. Security Levels

CSU_SEC_LEVEL	Non-Secure User	Non-Secure Supervisor	Secure User	Secure Supervisor
0	RD+WR	RD+WR	RD+WR	RD+WR
1	NOT ALLOWED	RD+WR	RD+WR	RD+WR
2	Read Only	Read Only	RD+WR	RD+WR
3	NOT ALLOWED	Read Only	RD+WR	RD+WR
4	NOT ALLOWED	NOT ALLOWED	RD+WR	RD+WR
5	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	RD+WR
6	NOT ALLOWED	NOT ALLOWED	Read Only	Read Only
7	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED	NOT ALLOWED

Information regarding CSU is provided in the Security Reference Manual. Contact your NXP representative for information about obtaining this document.

A 3-bit input, 8-bit output translation block can be used such that only three register bits are required to set the security profile and the translation block will drive the correct 8-bit configuration vector. Each peripheral connected to the AIPSTZ would require this translation block. The top level AIPSTZ has this three bit input line `csu_sec_level[2:0]' corresponding to each peripheral X.

4.7.7 AIPSTZ Memory Map/Register Definition

The memory map for the AIPS SW-visible registers is shown in the table below.

The MPROT and OPACR fields are 4 bits in width. Some bits may be reserved depending on device.

AIPSTZ memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3000_0000	Master Priviledge Registers (AIPSTZ_MPR)	32	R/W	7700_0000h	4.7.7.1/198
3000_0040	Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR)	32	R/W	4444_4444h	4.7.7.2/201
3000_0044	Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR1)	32	R/W	4444_4444h	4.7.7.3/204
3000_0048	Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR2)	32	R/W	4444_4444h	4.7.7.4/207
3000_004C	Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR3)	32	R/W	4444_4444h	4.7.7.5/210
3000_0050	Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR4)	32	R/W	4444_4444h	4.7.7.6/213

4.7.7.1 Master Priviledge Registers (AIPSTZ_MPR)

Each AIPSTZ_MPR specifies 16 4-bit fields defining the access privilege level associated with a bus master in the platform, as well as specifying whether write accesses from this master are bufferable shown in Table 4-21

The registers provide one field per bus master, where field 15 corresponds to master 15, field 14 to master 14,... field 0 to master 0 (typically the processor core). The master index allocation is shown in the table below.

Table 4-21. MPROT Field

Bit	Field	Description
3	MBW	**Master Buffer Writes** - This bit determines whether the AIPSTZ is enabled to buffer writes from this master.
2	MTR	**Master Trusted for Reads** - This bit determines whether the master is trusted for read accesses.
1	MTW	**Master Trusted for Writes** - This bit determines whether the master is trusted for write accesses.
0	MPL	**Master Privilege Level** - This bit determines how the privilege level of the master is determined.

NOTE

The reset value is set to 0000_0000_7700_0000, which makes master 0 and master 1 (Arm CORE) the trusted masters. Trusted software can change the settings after reset.

Table 4-22. Master Index Allocation

Master Index	Master Name	Comments
Master 0	All masters excluding Arm core	Share the same number allocation.
Master 1	Arm A53	
Master 4	SDMA	
Master 6	M4	

Address: 3000_0000h base + 0h offset = 3000_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MPROT0				MPROT1				MPROT2				MPROT3				Reserved				MPROT5				Reserved							
Reset	0	1	1	1	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

AIPSTZ_MPR field descriptions

Field	Description
31–28 MPROT0	Master 0 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses.

Table continues on the next page...

AIPSTZ_MPR field descriptions (continued)

Field	Description
	0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
27–24 MPROT1	Master 1 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
23–20 MPROT2	Master 2 Priviledge, Buffer, Read, Write Control xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
19–16 MPROT3	Master 3 Priviledge, Buffer, Read, Write Control. xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses. x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
15–12 -	This field is reserved. Reserved
11–8 MPROT5	Master 5 Priviledge, Buffer, Read, Write Control. xxx0 **MPL0** — Accesses from this master are forced to user-mode (ips_supervisor_access is forced to zero) regardless of the hprot[1] access attribute. xxx1 **MPL1** — Accesses from this master are not forced to user-mode. The hprot[1] access attribute is used directly to determine ips_supervisor_access. xx0x **MTW0** — This master is not trusted for write accesses. xx1x **MTW1** — This master is trusted for write accesses. x0xx **MTR0** — This master is not trusted for read accesses.

Table continues on the next page...

AIPSTZ_MPR field descriptions (continued)

Field	Description
	x1xx **MTR1** — This master is trusted for read accesses. 0xxx **MBW0** — Write accesses from this master are not bufferable 1xxx **MBW1** — Write accesses from this master are allowed to be buffered
-	This field is reserved. Reserved

4.7.7.2 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-23

Table 4-23. OPAC Field

Bit	Field	Description
3	BW	**Buffer Writes** - This bit determines whether write accesses to this peripheral are allowed to be buffered.[1]
2	SP	**Supervisor Protect** - This bit determines whether the peripheral requires supervisor privilege level for access.
1	WP	**Write Protect** - This bit determines whether the peripheral allows write accesses.
0	TP	**Trusted Protect** - This bit determines whether the peripheral allows accesses from an untrusted master.

1. Buffered writes are not available for AIPSTZ. This bit should be set to '0'.

Address: 3000_0000h base + 40h offset = 3000_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC0				OPAC1				OPAC2				OPAC3				OPAC4				OPAC5				OPAC6				OPAC7			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZ_OPACR field descriptions

Field	Description
31–28 OPAC0	Off-platform Peripheral Access Control 0 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZ_OPACR field descriptions (continued)

Field	Description
	x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC1	Off-platform Peripheral Access Control 1 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC2	Off-platform Peripheral Access Control 2 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC3	Off-platform Peripheral Access Control 3 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses.

Table continues on the next page...

AIPSTZ_OPACR field descriptions (continued)

Field	Description
	x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC4	Off-platform Peripheral Access Control 4 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC5	Off-platform Peripheral Access Control 5 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC6	Off-platform Peripheral Access Control 6 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the

Table continues on the next page...

AIPSTZ_OPACR field descriptions (continued)

Field	Description
	master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC7	Off-platform Peripheral Access Control 7 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.7.3 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR1)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-23

Address: 3000_0000h base + 44h offset = 3000_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC8				OPAC9				OPAC10				OPAC11				OPAC12				OPAC13				OPAC14				OPAC15			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZ_OPACR1 field descriptions

Field	Description
31–28 OPAC8	Off-platform Peripheral Access Control 8 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZ_OPACR1 field descriptions (continued)

Field	Description
	x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC9	Off-platform Peripheral Access Control 9 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC10	Off-platform Peripheral Access Control 10 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC11	Off-platform Peripheral Access Control 11 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses.

Table continues on the next page...

AIPSTZ_OPACR1 field descriptions (continued)

Field	Description
	x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC12	Off-platform Peripheral Access Control 12 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC13	Off-platform Peripheral Access Control 13 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC14	Off-platform Peripheral Access Control 14 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the

Table continues on the next page...

AIPSTZ_OPACR1 field descriptions (continued)

Field	Description
	master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC15	Off-platform Peripheral Access Control 15 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.7.4 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR2)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-23

Address: 3000_0000h base + 48h offset = 3000_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC16				OPAC17				OPAC18				OPAC19				OPAC20				OPAC21				OPAC22				OPAC23			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZ_OPACR2 field descriptions

Field	Description
31–28 OPAC16	Off-platform Peripheral Access Control 16 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses.

Table continues on the next page...

AIPSTZ_OPACR2 field descriptions (continued)

Field	Description
	xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC17	Off-platform Peripheral Access Control 17 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC18	Off-platform Peripheral Access Control 18 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC19	Off-platform Peripheral Access Control 19 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZ_OPACR2 field descriptions (continued)

Field	Description
	x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC20	Off-platform Peripheral Access Control 20 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC21	Off-platform Peripheral Access Control 21 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC22	Off-platform Peripheral Access Control 22 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses.

Table continues on the next page...

AIPSTZ_OPACR2 field descriptions (continued)

Field	Description
	x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC23	Off-platform Peripheral Access Control 23 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.7.5 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR3)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-23

Address: 3000_0000h base + 4Ch offset = 3000_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC24				OPAC25				OPAC26				OPAC27				OPAC28				OPAC29				OPAC30				OPAC31			
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZ_OPACR3 field descriptions

Field	Description
31–28 OPAC24	Off-platform Peripheral Access Control 24 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZ_OPACR3 field descriptions (continued)

Field	Description
	xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC25	Off-platform Peripheral Access Control 25 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
23–20 OPAC26	Off-platform Peripheral Access Control 26 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
19–16 OPAC27	Off-platform Peripheral Access Control 27 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses.

Table continues on the next page...

AIPSTZ_OPACR3 field descriptions (continued)

Field	Description
	xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
15–12 OPAC28	Off-platform Peripheral Access Control 28 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
11–8 OPAC29	Off-platform Peripheral Access Control 29 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
7–4 OPAC30	Off-platform Peripheral Access Control 30 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus.

Table continues on the next page...

AIPSTZ_OPACR3 field descriptions (continued)

Field	Description
	x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
OPAC31	Off-platform Peripheral Access Control 31 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.

4.7.7.6 Off-Platform Peripheral Access Control Registers (AIPSTZ_OPACR4)

Each of the off-platform peripherals have an Off-platform Peripheral Access Control Register (AIPSTZ_OPACR) which defines the access levels supported by the given block.

Each AIPSTZ_OPACR has the following format shown in Table 4-23

Address: 3000_0000h base + 50h offset = 3000_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OPAC32				OPAC33				Reserved																							
Reset	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0

AIPSTZ_OPACR4 field descriptions

Field	Description
31–28 OPAC32	Off-platform Peripheral Access Control 32 xxx0 **TP0** — Accesses from an untrusted master are allowed.

Table continues on the next page...

AIPSTZ_OPACR4 field descriptions (continued)

Field	Description
	xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
27–24 OPAC33	Off-platform Peripheral Access Control 33 xxx0 **TP0** — Accesses from an untrusted master are allowed. xxx1 **TP1** — Accesses from an untrusted master are not allowed. If an access is attempted by an untrusted master, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. xx0x **WP0** — This peripheral allows write accesses. xx1x **WP1** — This peripheral is write protected. If a write access is attempted, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. x0xx **SP0** — This peripheral does not require supervisor privilege level for accesses. x1xx **SP1** — This peripheral requires supervisor privilege level for accesses. The master privilege level must indicate supervisor via the hprot[1] access attribute, and the MPROTx[MPL] control bit for the master must be set. If not, the access is terminated with an error response and no peripheral access is initiated on the IPS bus. 0xxx **BW0** — Write accesses to this peripheral are not bufferable by the AIPSTZ. 1xxx **BW1** — Write accesses to this peripheral are allowed to be buffered by the AIPSTZ.
-	This field is reserved. Reserved

4.8 Shared Peripheral Bus Arbiter (SPBA)

4.8.1 Overview

The Shared Peripheral Bus Arbiter (SPBA) is a three-to-one IP Bus interface arbiter. Three masters arbitrate for shared peripheral access through the SPBA.

The SPBA has three primary functions:

- The IP Bus Line switches a master to one peripheral

- The Masters arbiter arbitrates between the three masters to solve concurrent access or restricted access to peripherals
- The Control Registers and Ownership Control includes a set of registers which are reachable through software and permit the access scheme to be defined for each peripheral (Resource Ownership and Access Control). It generates signals for the external steering logic of interrupts and DMA signals.

The figure below shows the SPBA block diagram

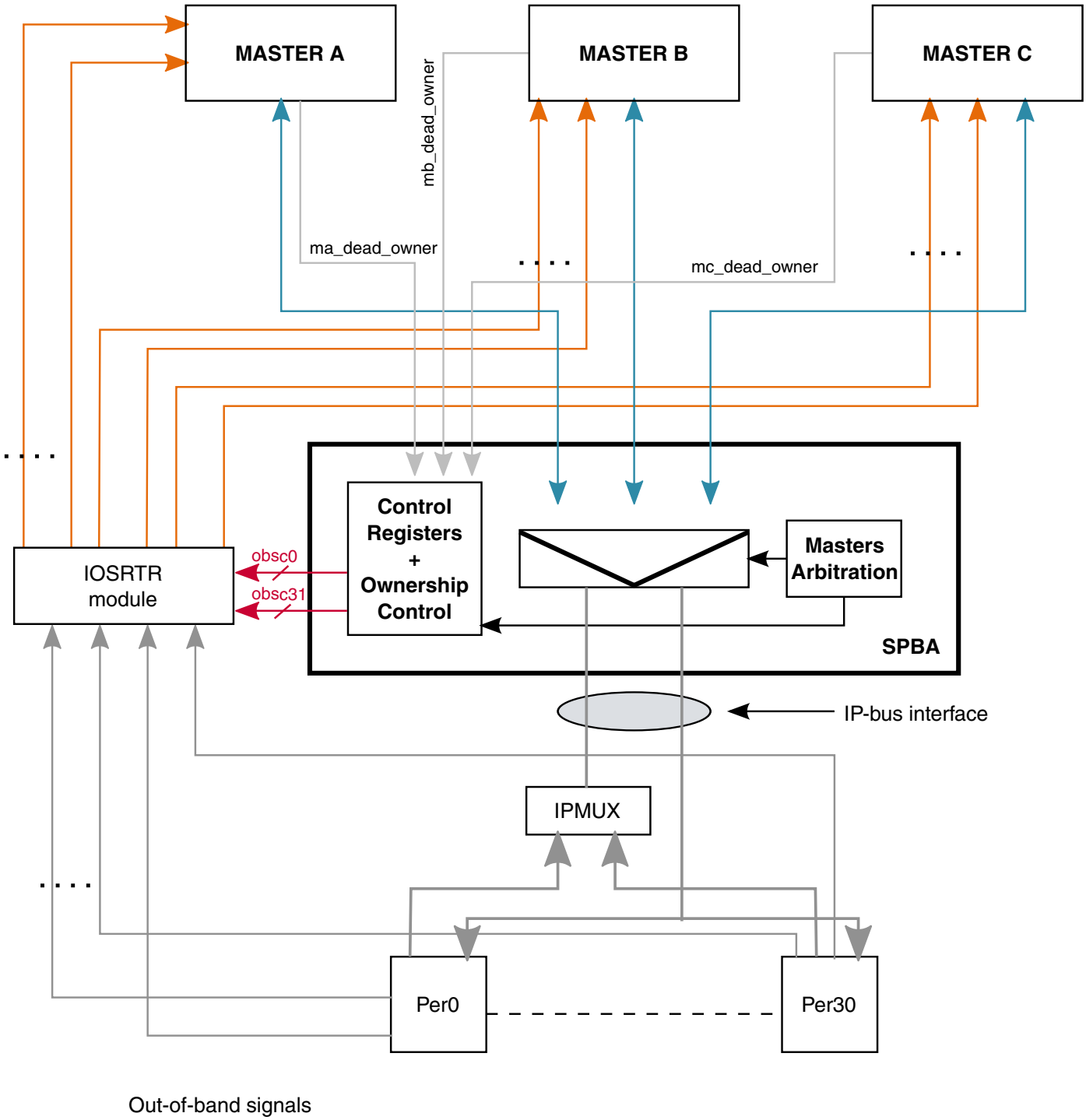


Figure 4-15. SPBA Block Diagram

4.8.1.1 Features

The SPBA includes the following features:

- Three IP Bus masters arbitration: Master A, B and C
- Support for DMA masters
- 32-bit data
- Supports up to 31 shared peripherals, each consuming 64 kilobytes of address space
- SPBA can be considered the 32nd peripheral, used for resource ownership and access control of the 31 peripherals
- Provides 31 sets of out of band steering control (OBSC) signals to the off-block steering logic
- Operating frequency up to 67 MHz
- Clocks: ipg_clk, ipg_clk_s

4.8.1.2 Modes of operation

SPBA behavior is transparent when accessing a peripheral, though it has these distinct modes of operation.

Reset/Abort

The SPBA has a hardware reset which initializes all registers, arbitration and peripherals rights registers (PRRs).

An abort signal input is provided allowing each master to abort its current access and release ownership (in case of master reset sequence).

Functional

Once a master request is granted, its IP Bus signals are steered to the requested peripheral.

Standby

No clock needed. The SPBA needs clocks only during access to the PRRs, arbitration, and abort phases. It generates two clock enable signals indicating when the clocks must be provided.

Configuration

During this phase, a master accesses the SPBA PRRs. The SPBA memory-mapped registers are seen as a shared peripheral.

4.8.2 Clocks

The table found here describes the clock sources for SPBA.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 4-24. SPBA Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

4.8.3 Functional description

4.8.3.1 Masters arbitration

The arbitration mechanism determines which port will control the master port, based on a simple round-robin arbitration scheme.

There are several use cases to consider.

- Only one master request per access. The master is switched to the shared peripheral bus, without arbitration. Figure 4-16 shows the MB request on the global module enable signal, served without wait state.
- If two masters simultaneously access SPBA, the last granted master is held off using the <master>_ips_xfr_wait output signal (default value is high). When the master is granted sips_xfr_wait, shared IP Bus peripheral is connected to <master>_ips_xfr_wait outputs.
- If three masters simultaneously access SPBA, then the last two granted masters are held off using <master>_ips_xfr_wait. Figure 4-17 shows a case in which the last two accesses granted are MA and MB. The requests are used even if they are in the same cycle.
- If after reset, at the first multiple access, no master has been granted, the priority is static: Master A (MA), Master B (MB) and last Master C (MC) port.
- No master request. No master switch to shared peripherals.

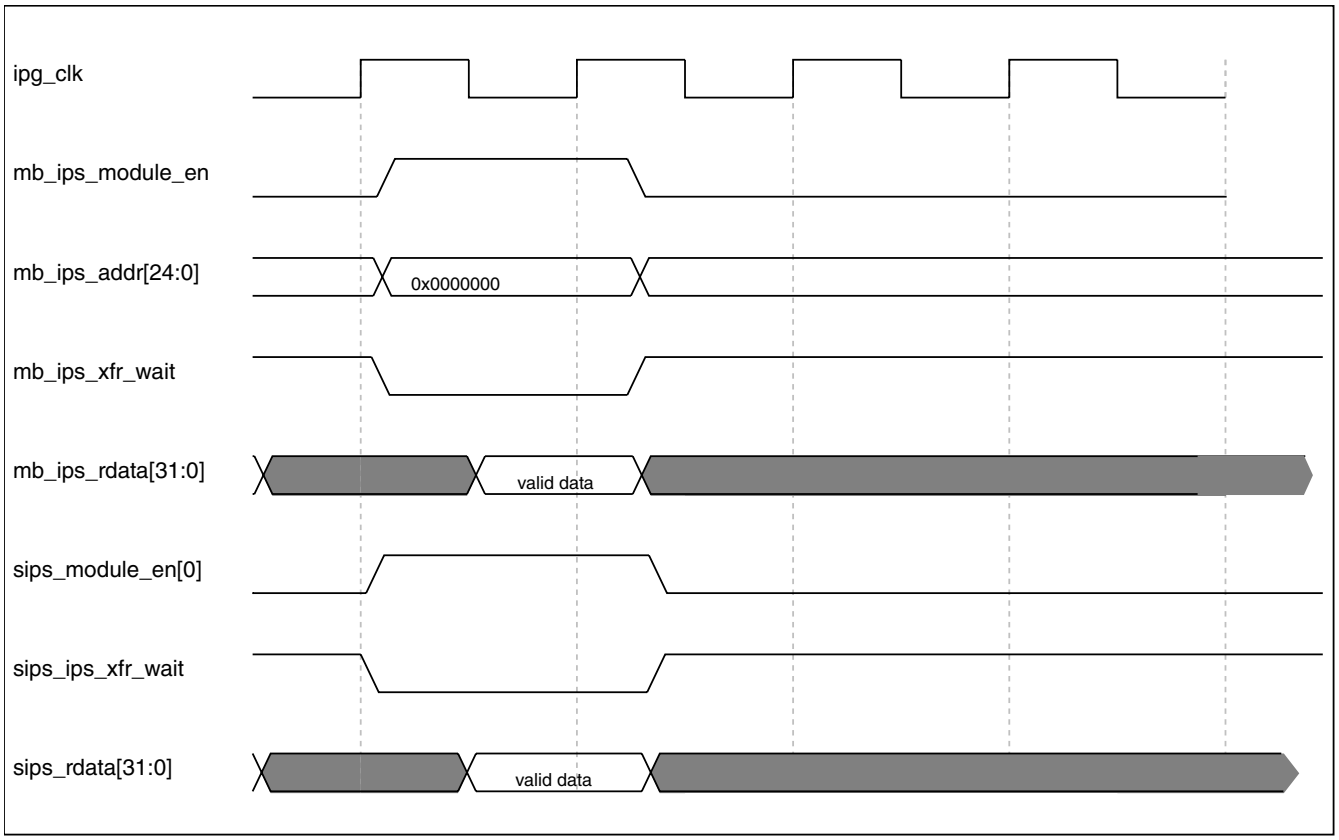


Figure 4-16. Example of one master request, no SPBA arbitration

The following figure assumes MA and MB have been the last two masters granted in the previous transfers (MA then MB).

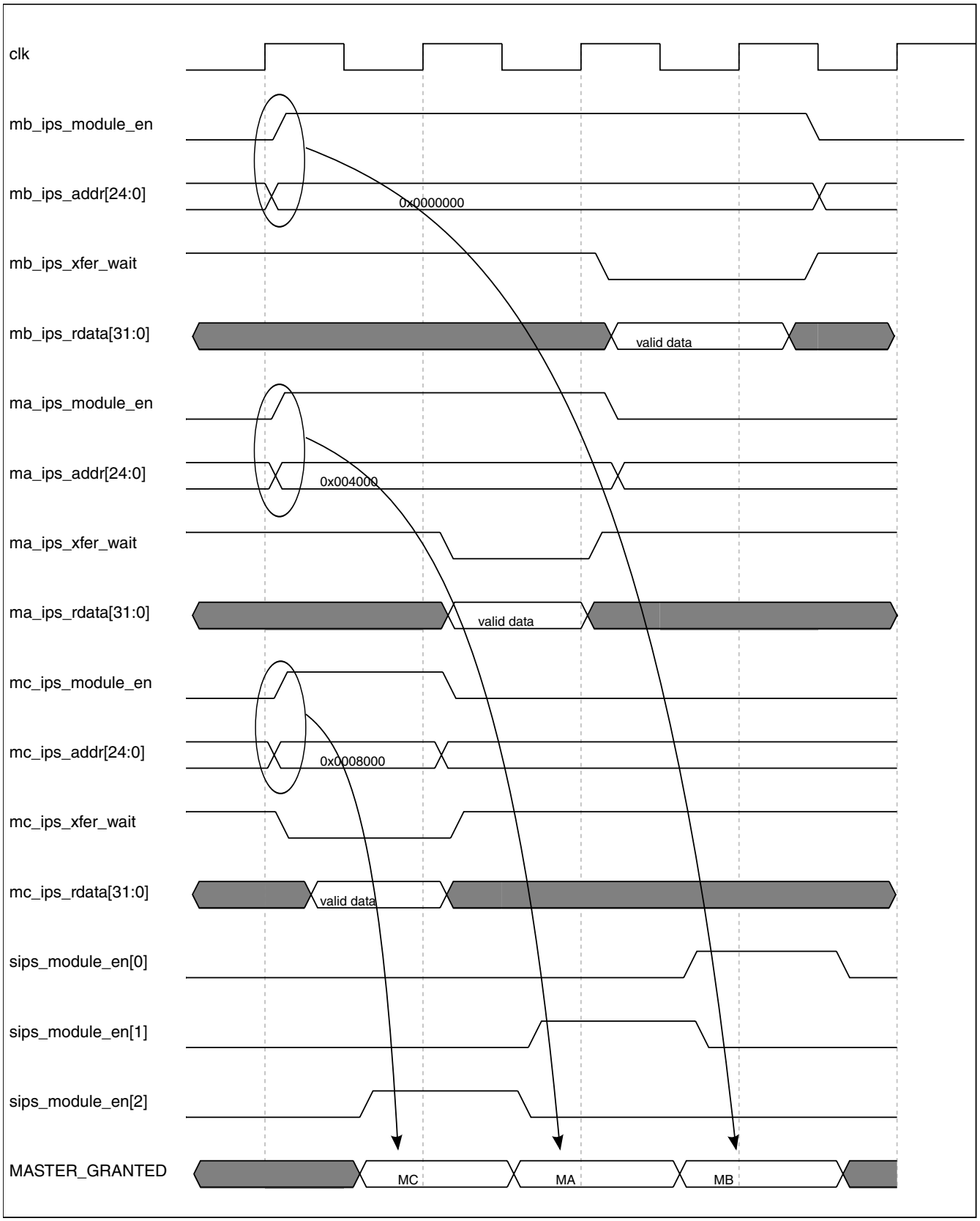


Figure 4-17. Example of three master requests: Masters already granted are "waited";

4.8.4 Resource ownership control

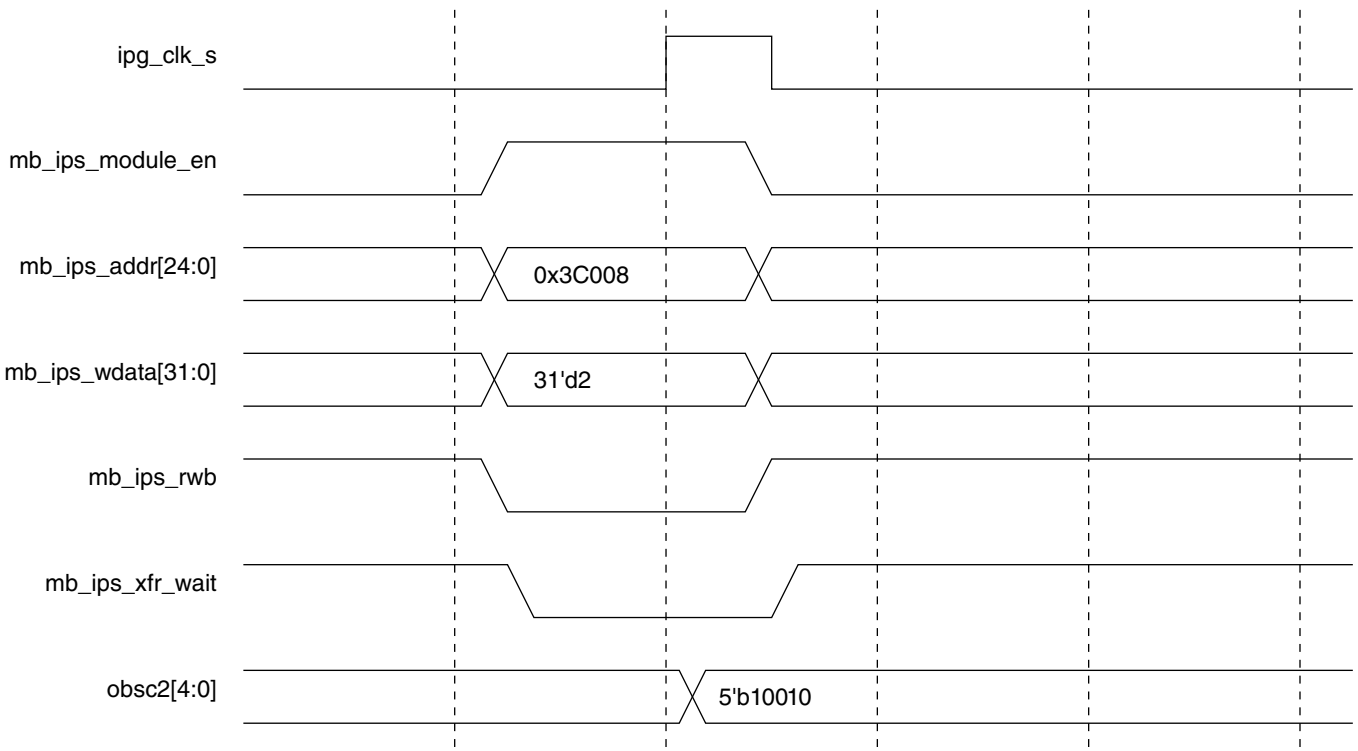
The resource ownership control regulates access to the shared peripherals and determines the steering of out-of-band signals.

4.8.4.1 Access control

4.8.4.1.1 Peripheral access

The peripheral access (resource access) of the requesting master is given by the corresponding RAR bit of the Peripheral Right Register. It determines if the master has access privilege to the resource.

Any attempt at access made by a requesting master whose access privilege bit is not set (in the PRR) is terminated with a bus error (**<master>_ips_xfr_err** is asserted by SPBA logic). The master that owns the resource can lock the peripheral for itself and/or grant other masters access to the peripheral by setting the appropriate bit(s) in the RAR field.



Master B is taking ownership of peripheral 2 by writing 3'b010 in the SPBA peripheral 2 right register (rarfield)
This ownership can be checked on obsc2 output as roi2[1:0] = 2'b10 and rar2[2:0] = 3'b010
(obsc[4:0] = {roi2[1], roi2[0], rar2[2], rar2[1], rar2[0]})

Figure 4-18. Example of one master B gaining ownership of peripheral 2

4.8.4.1.2 Peripheral Right Register access

The ROI bits of the Peripheral Right Register (PRR) determine which master is allowed to make write access to PRR. The identification of the requesting master is compared to the ROI bits of the PRR to determine if the master has ownership of the corresponding register.

Any attempted write access to a PRR already owned by another master will be ignored.

4.8.4.2 Owner election

When the peripheral is not owned by any master (ROI="00", after coming out of reset for instance), the first master to perform successfully a write to the RAR bits of the PRR is granted ownership of the peripheral and its associated PRR.

After writing to the PRR (RAR bit(s)), the master must read it back to make sure that it was granted ownership. If the RMO field is 2'b11, then the ownership claim is successful. If RMO is 2'b10, another master claimed ownership before this master was able to complete its write. This resolves the case in which two or more masters attempt to write the PRR at the same time; only the first master will be granted ownership. However all masters must read the PRR to determine if this case occurred, and if so, whether they were the first master which was granted ownership.

NOTE

A master that has been granted ownership of the PRR does not automatically have the right access to the peripheral; it must still set its own RAR bits in the PRR to access the peripheral.

4.8.4.3 Ending ownership

Ownership may be voluntarily ended by the owning master, or automatically upon assertion of a master-specific dead_owner signal.

The former is appropriate for software-controlled yielding of ownership. The latter is appropriate for automatic yielding of ownership when the owner has gone into reset.

When a master is reset, it clears the ROI bits of the PRRs owned by the corresponding master. When the owner is dead (in reset), all peripherals previously owned by that master must be changed to the un-owned state.

NOTE

It is the programmer's responsibility to make sure the peripherals are placed in an appropriate state before ending ownership.

4.8.4.3.1 Software Controlled Ownership Ending

The ROI bits will be automatically cleared when the master that owns the PRR access right clears (write) the RAR bits (Table 2).

It will then end the ownership of the PRR.

4.8.4.4 The Un-owned State

During the time when the peripheral is un-owned (i.e the ROI field contains all 0's), all masters have full access to it (RAR bits can then be modified by a master if ROI[1:0] = 2'b0).

In such cases it is necessary for software to ensure any necessary coherency in the resource, there is no hardware protection.

4.8.5 SPBA Memory Map/Register Definition

The SPBA control registers (Peripheral Right Registers) are mapped as a virtual shared peripheral.

SPBA can support up to 31 shared peripherals. Each of them has its own Peripheral Right Register (PRR) accessible within the SPBA memory-mapped registers, and consists of the Requesting Master Owner, the Resource Owner ID and the Resource Access Right fields.

SPBA memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
300F_0000	Peripheral Rights Register (SPBA2_PRR0)	32	R/W	0000_0007h	4.8.5.1/226
300F_0004	Peripheral Rights Register (SPBA2_PRR1)	32	R/W	0000_0007h	4.8.5.1/226
300F_0008	Peripheral Rights Register (SPBA2_PRR2)	32	R/W	0000_0007h	4.8.5.1/226
300F_000C	Peripheral Rights Register (SPBA2_PRR3)	32	R/W	0000_0007h	4.8.5.1/226
300F_0010	Peripheral Rights Register (SPBA2_PRR4)	32	R/W	0000_0007h	4.8.5.1/226
300F_0014	Peripheral Rights Register (SPBA2_PRR5)	32	R/W	0000_0007h	4.8.5.1/226
300F_0018	Peripheral Rights Register (SPBA2_PRR6)	32	R/W	0000_0007h	4.8.5.1/226
300F_001C	Peripheral Rights Register (SPBA2_PRR7)	32	R/W	0000_0007h	4.8.5.1/226
300F_0020	Peripheral Rights Register (SPBA2_PRR8)	32	R/W	0000_0007h	4.8.5.1/226
300F_0024	Peripheral Rights Register (SPBA2_PRR9)	32	R/W	0000_0007h	4.8.5.1/226
300F_0028	Peripheral Rights Register (SPBA2_PRR10)	32	R/W	0000_0007h	4.8.5.1/226
300F_002C	Peripheral Rights Register (SPBA2_PRR11)	32	R/W	0000_0007h	4.8.5.1/226
300F_0030	Peripheral Rights Register (SPBA2_PRR12)	32	R/W	0000_0007h	4.8.5.1/226
300F_0034	Peripheral Rights Register (SPBA2_PRR13)	32	R/W	0000_0007h	4.8.5.1/226
300F_0038	Peripheral Rights Register (SPBA2_PRR14)	32	R/W	0000_0007h	4.8.5.1/226
300F_003C	Peripheral Rights Register (SPBA2_PRR15)	32	R/W	0000_0007h	4.8.5.1/226

Table continues on the next page...

SPBA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
300F_0040	Peripheral Rights Register (SPBA2_PRR16)	32	R/W	0000_0007h	4.8.5.1/226
300F_0044	Peripheral Rights Register (SPBA2_PRR17)	32	R/W	0000_0007h	4.8.5.1/226
300F_0048	Peripheral Rights Register (SPBA2_PRR18)	32	R/W	0000_0007h	4.8.5.1/226
300F_004C	Peripheral Rights Register (SPBA2_PRR19)	32	R/W	0000_0007h	4.8.5.1/226
300F_0050	Peripheral Rights Register (SPBA2_PRR20)	32	R/W	0000_0007h	4.8.5.1/226
300F_0054	Peripheral Rights Register (SPBA2_PRR21)	32	R/W	0000_0007h	4.8.5.1/226
300F_0058	Peripheral Rights Register (SPBA2_PRR22)	32	R/W	0000_0007h	4.8.5.1/226
300F_005C	Peripheral Rights Register (SPBA2_PRR23)	32	R/W	0000_0007h	4.8.5.1/226
300F_0060	Peripheral Rights Register (SPBA2_PRR24)	32	R/W	0000_0007h	4.8.5.1/226
300F_0064	Peripheral Rights Register (SPBA2_PRR25)	32	R/W	0000_0007h	4.8.5.1/226
300F_0068	Peripheral Rights Register (SPBA2_PRR26)	32	R/W	0000_0007h	4.8.5.1/226
300F_006C	Peripheral Rights Register (SPBA2_PRR27)	32	R/W	0000_0007h	4.8.5.1/226
300F_0070	Peripheral Rights Register (SPBA2_PRR28)	32	R/W	0000_0007h	4.8.5.1/226
300F_0074	Peripheral Rights Register (SPBA2_PRR29)	32	R/W	0000_0007h	4.8.5.1/226
300F_0078	Peripheral Rights Register (SPBA2_PRR30)	32	R/W	0000_0007h	4.8.5.1/226
300F_007C	Peripheral Rights Register (SPBA2_PRR31)	32	R/W	0000_0007h	4.8.5.1/226
308F_0000	Peripheral Rights Register (SPBA1_PRR0)	32	R/W	0000_0007h	4.8.5.1/226
308F_0004	Peripheral Rights Register (SPBA1_PRR1)	32	R/W	0000_0007h	4.8.5.1/226
308F_0008	Peripheral Rights Register (SPBA1_PRR2)	32	R/W	0000_0007h	4.8.5.1/226
308F_000C	Peripheral Rights Register (SPBA1_PRR3)	32	R/W	0000_0007h	4.8.5.1/226
308F_0010	Peripheral Rights Register (SPBA1_PRR4)	32	R/W	0000_0007h	4.8.5.1/226
308F_0014	Peripheral Rights Register (SPBA1_PRR5)	32	R/W	0000_0007h	4.8.5.1/226
308F_0018	Peripheral Rights Register (SPBA1_PRR6)	32	R/W	0000_0007h	4.8.5.1/226
308F_001C	Peripheral Rights Register (SPBA1_PRR7)	32	R/W	0000_0007h	4.8.5.1/226
308F_0020	Peripheral Rights Register (SPBA1_PRR8)	32	R/W	0000_0007h	4.8.5.1/226
308F_0024	Peripheral Rights Register (SPBA1_PRR9)	32	R/W	0000_0007h	4.8.5.1/226
308F_0028	Peripheral Rights Register (SPBA1_PRR10)	32	R/W	0000_0007h	4.8.5.1/226
308F_002C	Peripheral Rights Register (SPBA1_PRR11)	32	R/W	0000_0007h	4.8.5.1/226
308F_0030	Peripheral Rights Register (SPBA1_PRR12)	32	R/W	0000_0007h	4.8.5.1/226
308F_0034	Peripheral Rights Register (SPBA1_PRR13)	32	R/W	0000_0007h	4.8.5.1/226
308F_0038	Peripheral Rights Register (SPBA1_PRR14)	32	R/W	0000_0007h	4.8.5.1/226
308F_003C	Peripheral Rights Register (SPBA1_PRR15)	32	R/W	0000_0007h	4.8.5.1/226
308F_0040	Peripheral Rights Register (SPBA1_PRR16)	32	R/W	0000_0007h	4.8.5.1/226
308F_0044	Peripheral Rights Register (SPBA1_PRR17)	32	R/W	0000_0007h	4.8.5.1/226
308F_0048	Peripheral Rights Register (SPBA1_PRR18)	32	R/W	0000_0007h	4.8.5.1/226
308F_004C	Peripheral Rights Register (SPBA1_PRR19)	32	R/W	0000_0007h	4.8.5.1/226
308F_0050	Peripheral Rights Register (SPBA1_PRR20)	32	R/W	0000_0007h	4.8.5.1/226
308F_0054	Peripheral Rights Register (SPBA1_PRR21)	32	R/W	0000_0007h	4.8.5.1/226

Table continues on the next page...

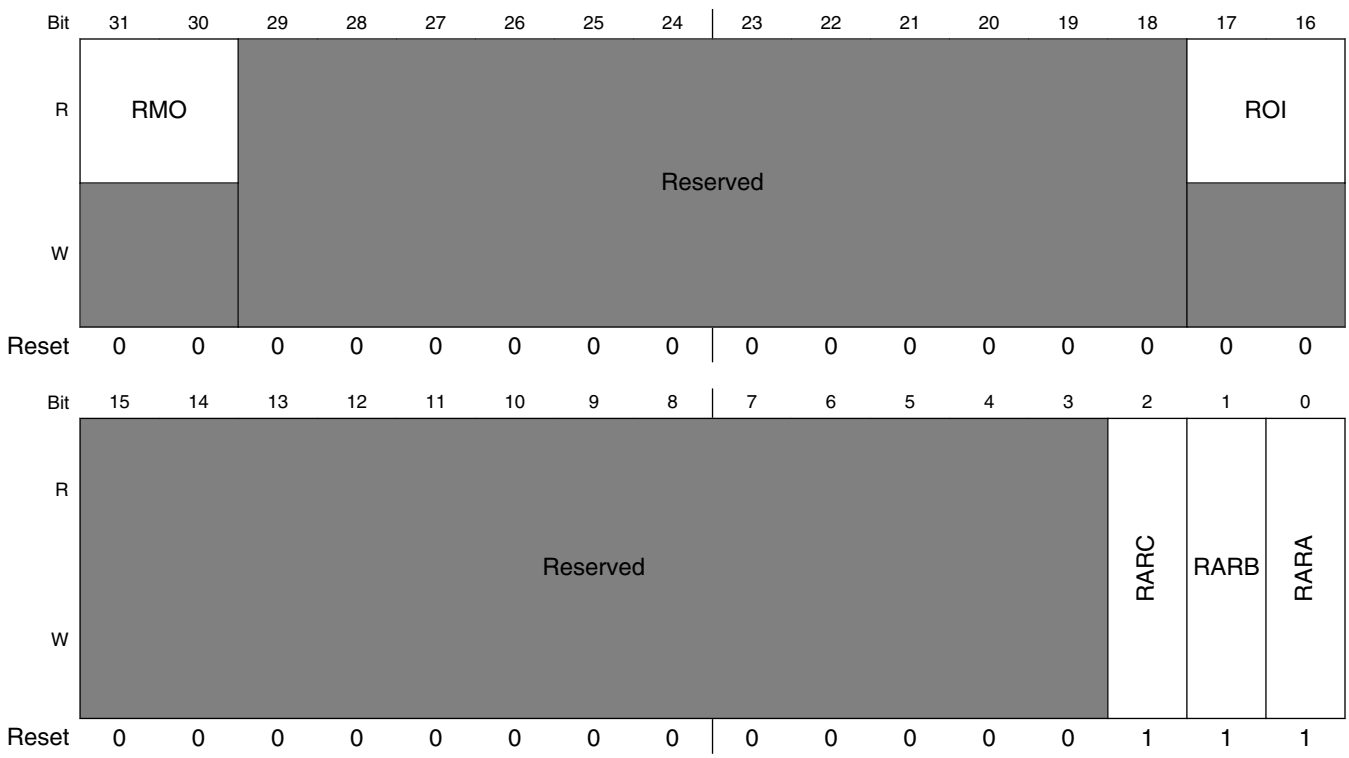
SPBA memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
308F_0058	Peripheral Rights Register (SPBA1_PRR22)	32	R/W	0000_0007h	4.8.5.1/226
308F_005C	Peripheral Rights Register (SPBA1_PRR23)	32	R/W	0000_0007h	4.8.5.1/226
308F_0060	Peripheral Rights Register (SPBA1_PRR24)	32	R/W	0000_0007h	4.8.5.1/226
308F_0064	Peripheral Rights Register (SPBA1_PRR25)	32	R/W	0000_0007h	4.8.5.1/226
308F_0068	Peripheral Rights Register (SPBA1_PRR26)	32	R/W	0000_0007h	4.8.5.1/226
308F_006C	Peripheral Rights Register (SPBA1_PRR27)	32	R/W	0000_0007h	4.8.5.1/226
308F_0070	Peripheral Rights Register (SPBA1_PRR28)	32	R/W	0000_0007h	4.8.5.1/226
308F_0074	Peripheral Rights Register (SPBA1_PRR29)	32	R/W	0000_0007h	4.8.5.1/226
308F_0078	Peripheral Rights Register (SPBA1_PRR30)	32	R/W	0000_0007h	4.8.5.1/226
308F_007C	Peripheral Rights Register (SPBA1_PRR31)	32	R/W	0000_0007h	4.8.5.1/226

4.8.5.1 Peripheral Rights Register (SPBAx_PRR*n*)

This register controls master ownership and access for a peripheral.

Address: Base address + 0h offset + (4d × i), where i=0d to 31d



SPBAx_PRR*n* field descriptions

Field	Description
31–30 RMO	Requesting Master Owner. This 2-bit register field indicates if the corresponding resource is owned by the requesting master or not. This register is reset to 2'b0 if ROI = 2'b0. 00 **UNOWNED** — The resource is unowned. 01 Reserved. 10 **ANOTHER_MASTER** — The resource is owned by another master. 11 **REQUESTING_MASTER** — The resource is owned by the requesting master.
29–18 -	This field is reserved. Reserved
17–16 ROI	Resource Owner ID. This field indicates which master (one at a time) can access to the PRR for rights modification. This is a read-only register. After reset, ROI bits are cleared ("00" -> un-owned resource). A master performing a write access to the an un-owned PRR will get its ID automatically written into ROI, while modifying RARx bits. It can then read back the RMO, RAR, ROI bits to make sure RMO returns the right value, ROI bits contain its ID and RARx bits are correctly asserted. Then no other master (whom ID is different from the one stored in ROI) will be able to modify RAR fields. Owner master of a peripheral can assert its dead_owner signal, or write 1'b0 in the RARx to release the ownership (ROI[1:0] reset to 2'b0). 00 **UNOWNED** — Unowned resource. 01 **MASTER_A** — The resource is owned by master A port. 10 **MASTER_B** — The resource is owned by master B port. 11 **MASTER_C** — The resource is owned by master C port.
15–3 -	This field is reserved. Reserved
2 RARC	Resource Access Right. Control and Status bit for master C. This field indicates whether master C can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
1 RARB	Resource Access Right. Control and Status bit for master B. This field indicates whether master B can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.
0 RARA	Resource Access Right. Control and Status bit for master A. This field indicates whether master A can access the peripheral. From 0 up to 3 masters can have permission to access a resource (all the master can be granted on a peripheral, but only one access at a time will be granted by SPBA). 0 **PROHIBITED** — Access to peripheral is not allowed. 1 **ALLOWED** — Access to peripheral is granted.

4.9 ROM Controller with Patch (ROMCP)

4.9.1 Overview

The Read Only Memory Controller with ROM Patch (ROMC) acts as an interface between the Arm advanced high-performance bus (AHB - Lite) and the Read Only Memory. The ROMC consists of a ROM Controller and a ROM Patch. The ROM Patch is used to either patch code routines or fix data tables in the ROM area. There is an IP Bus interface to access the ROM Patch Registers. The following figure depicts the main functional sub-blocks of the ROMC.

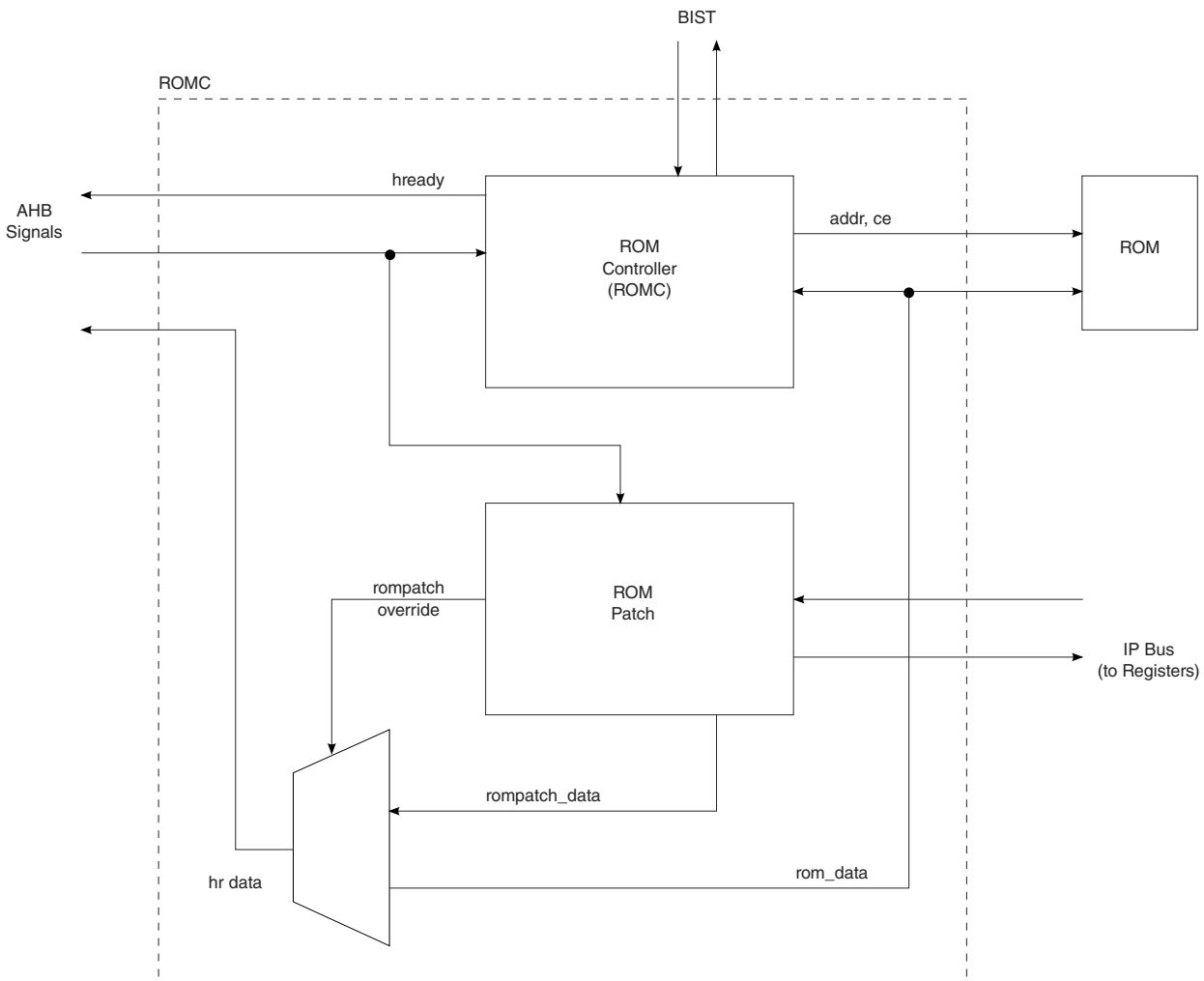


Figure 4-19. ROMC Block Diagram

4.9.1.1 Features

- Supports ROM size ranges from 16 Kbyte up to 4 Mbyte with increments of 1 Kbyte
- Supports opcode patching for a maximum of 16 different addresses in 4 Mbytes of ROM space
- Supports one-word data fixes for a max of 8 memory locations in 4 Mbytes of ROM space
- Supports patching of the Reset Vector (at 0x0000_0000) to allow external booting

4.9.1.2 Modes of Operation

There are two modes of operation: normal mode and BIST mode. In normal mode, the ROMC ensures correct reads from the ROM, assuming the memory complies with the characteristics and requirements for which the ROMC was designed.

4.9.1.2.1 Low Power Mode

There are two clock enables that are used to switch off parts of the ROMC logic when inactive. The first clock enable is used to disable the ROM Controller when the master connected to the AHB interface is not initiating a read to the ROM. The second clock enable is used to disable the registers used to program the ROM patch feature when the registers are not being accessed.

4.9.2 Clocks

The table found here describes the clock sources for ROMCP.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 4-25. ROMCP Clocks

Clock name	Clock Root	Description
hclk	ahb_clk_root	System / bus clock
hclk_reg	ipg_clk_root	System access clock
96krom_CLK	ahb_clk_root	ROM clock

4.9.3 Memory Map

4.9.3.1 ROM Memory Map in detail

The ROMC supports ROM sizes with a range of 16 Kbyte to 4 Mbyte with an increment of 1 Kbyte. The 16 Kbyte lower limit was chosen because the minimum size of security code on an Arm platform is approximately 16 Kbyte of code, which is only accessible in supervisor mode. Note that it is the MMU that controls whether any region of memory is secure.

The exception vectors must be secured as well, and must be put in the same area as the security code. Since they must reside at address 0x0000_0000, the entire 16 Kbyte of ROM which can only be accessible in supervisor mode is located at the very beginning of the platform memory map.

If the user chooses not to use the security code, a memory size smaller than 16 Kbyte can be connected to the platform (minimum of 1 Kbyte). The MMU can be programmed to allow any kind of access into this memory. However, if the ROM size is less than 16 Kbyte, memory aliasing will occur for all invalid addresses greater than the memory size but within the 16 Kbyte of space.

For ROM sizes bigger than 16 Kbyte, the rest of its physical size resides at the address starting at 0x0040_4000 (4 M+16 Kbyte) going up to [0x0040_4000 + (mem. size - 16Kbyte)].

4.9.4 Functional Description

This section is divided up into the ROM Controller Functional Description and the ROMC functional description.

4.9.4.1 ROM Controller (ROMC) Functional Description

4.9.4.1.1 Functionality overview

The ROMC serves two main functions. First, as an interface between the AHB-Lite bus on an Arm platform and the ROM. Second, it drives and receives several signals for the BIST engine. In normal mode of operation, the ROMC monitors the AHB-Lite for memory access requests and performs the memory operation to the ROM.

The ROMC includes the option to wait state all accesses from either the Arm or non-Arm masters to ROM in the event that timing requirements will not allow single HCLK clock cycle reads. If a wait state is required, the static inputs rom_wait_arm or

rom_wait_alt_mstr can be set to 1 and accesses will take two HCLK clock cycles. If wait states are not required, rom_wait_arm or rom_wait_alt_mstr can be set to 0 and accesses will take one HCLK clock cycle to complete.

4.9.4.2 ROMC Functional Description

4.9.4.2.1 ROMC Disabling

All the bits in the ROMC_ROMPATCHENL register are cleared on Reset, disabling all the address comparators. Once the comparators have been enabled, the ROMC functions of data fixing and opcode patching can be quickly disabled by setting the DIS bit in the ROMC_ROMPATCHCNTL register. This bit is used to enable secure operations in which patching functions need to be disabled. This bit is cleared on Reset.

4.9.4.2.2 ROMC Event Priority

The ROMC has a total of 16 address comparators. The first 8 (0 through 7) comparators can be programmed for the data fixing function (through the 8 data fix enable bits in the ROMC_ROMPATCHCNTL register) while the rest are for opcode patching by default. This allows for potential multiple matching events involving both data fixing and opcode patch types. In these cases the ROMC assigns the highest priority to a data fixing event.

For example, if the ROMC is set up to data fix a certain address with comparator 4 and also opcode patch the same address with comparator 7, it will let comparator 4 have higher priority in indicating a match, and data from ROMC_ROMPATCHD4 will be put on the rompatch_romc_hrdata bus as the override value.

If multiple address matches of the same type level occur concurrently, then the ROMC will choose the source number based on the one with the highest source number. For example, the ROMC is setup to data fix the same location with address comparators 4 and 7, then address comparator 7 will have higher priority in indicating a match, and the value from ROMC_ROMPATCHD7 will be put on the rompatch_romc_hrdata bus as the orverride value. The same priority applies for an opcode patch event, except the override data is in the form of an SWI instruction with the comment field set to the source number with the highest priority.

4.9.4.2.3 Data Fixing

The data fixing feature allows ROM data to be updated by direct replacement when it is being read. This data usually originates from data tables, but can include Arm instructions. To enable data fixing on a certain address, this address value is written in to one of the first eight (0 through 7) of ROMC_ROMPATCHAxx registers and the same

numbered bit set in the ROMC_ROMPATCHENL and ROMC_ROMPATCHCNTL registers. The data to be used for replacement is placed in the corresponding ROMC_ROMPATCHDxx.

The ROMC looks for a read access to ROM (either code fetch or data load) by snooping the AHB interface for read transactions. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC places the value in the corresponding ROMC_ROMPATCHDxx register on the read data bus by overriding the read data coming from the actual ROM (see the mux in Figure 4-19). The value on the read data bus is maintained until hready is asserted to terminate the access. In data fixing, the entire word is replaced so if a byte or half-word access occurs on a "data fix" location, the entire data word is replaced. The word being replaced is word aligned. (The two LSBs of the matching ROMC_ROMPATCHAxx are ignored in the data fix operation.)

4.9.4.2.4 Opcode Patching

The opcode patch feature provides the Arm core a mechanism to fetch updated versions of code routines that were originally programmed in ROM. This patching mechanism makes use of the SWI (software interrupt instruction) and a table of function pointers residing in writable memory. The opcode being patched is replaced with a SWI instruction by the ROMC. Subsequent processing of the SWI reads from a function pointers table to obtain the address of the replacement code. Execution resumes with this code patch.

To enable opcode patching of a certain address, this address value is written into one of the ROMPATCHAxx registers and the corresponding bit set in the ROMPATCHENL to enable the associated comparator. The register's LSB (ROMC_ROMPATCHxx[0]) should be set if THUMB mode patching is in effect for this address. The ROMC identifies a ROM read access by snooping the AHB interface. The address is compared with the values stored in the ROMC_ROMPATCHAxx[22:2] registers. If a match occurs from one of the comparators, the ROMC generates the opcode of a software interrupt (SWI) instruction with the comment field containing the number of the matching address comparator. This opcode and comment is placed on the read data bus until hready is asserted by the ROM controller to terminate the read access.

The type of SWI generated, (that is, either Arm or THUMB), is determined by the LSB of the ROMC_ROMPATCHAxx register associated with the opcode patch. This bit is cleared for Arm mode (32 bits). The ROMC generates a 32-bit SWI (opcode field is 0xEF, occupying bits [31:24] of the word), with the least significant 5 bits of the 24-bit comment field (bits [23:0]) containing the number of the matching address comparator.

The rest of the comment field is filled with zeros. This means that the ROMC will use 16 of the 16777216 possible software interrupts. The ROMC overrides the read data from the ROM.

If the LSB of the matching ROMC_ROMPATCHAxx register is set, the opcode patch is in THUMB mode (16 bits or half word). The ROMC generates a 16-bit SWI instruction (opcode field is 0xDF, occupying bits [15:8] of the half word) with the least significant 5 bits of the 8-bit comment field containing with the source number of the address comparator. The rest of the comments field is filled with zeros. This means that the ROMC will use 16 of the 256 possible software interrupts. The ROMC puts this 16 bit SWI instruction value on the proper half of the rompatch_romc_hrdata bus. The other half is zeroed out. Which half of the bus contains the SWI opcode and comment depends on the mode (Big Endian or Little Endian) and the bit 1 of the matching ROMC_ROMPATCHAxx register. In Little Endian mode, the lower half is bits {15:0] and the upper half is bits {31:16]. The order is reversed in Big Endian mode.

In Little Endian mode (bigend signal negated), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the lower 16 bits of the read data bus and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten by the ROMC data. If ROMC_ROMPATCHAxx[1] is set (upper half word selected), the SWI instruction is put on the upper 16 bits of the read data bus and the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten.

In Big Endian mode (bigend asserted), if bit 1 of the matching ROMC_ROMPATCHAxx is cleared (lower half word selected) then the SWI instruction is put on the upper 16 bits of the read data bus while the lower 16 bits are zeroed out. Only the upper 16 bits of the read data bus is overwritten. If ROMC_ROMPATCHAxx[1] is set (upper word selected), the SWI instruction is put on the lower 16 bits and the upper 16 bits are zeroed out. Only the lower 16 bits of the read data bus is overwritten.

The eventual execution of the SWI causes the Arm to save the CPSR in SPSR_SVC, the address of the next instruction after the SWI in R14_SVC, enter Supervisor mode, and fetch the SWI vector at 0x8, which then takes it to a handler for further processing as described in the next section.

4.9.4.2.4.1 Typical Software Response to Opcode Patch

When the SWI handler executes it needs to determine whether the SWI was generated by the ROMC. This is done by loading the SWI instruction and extracting its comment field. The state of the Arm core (Arm or THUMB) when the SWI was executed dictates

whether to load the instruction word (Arm) or half word (THUMB). This state information can be determined by testing the T bit (bit 5) of the SPSR. If it's set, the execution was in THUMB mode.

By convention, if the comment field of the SWI is greater than 16, the software interrupt was initiated by software (i.e. an operating system call), and a branch is taken to the appropriate handler routine for further processing. If the comment field is less than 16, the SWI was generated by the ROMC performing a code patch operation. In this case, the software then reads from a table of function pointers, using the value in the SWI comment field as the index into the table. The value that is read is the address of the code patch. This value is loaded into the PC to begin the execution of the code patch. The following code segment illustrates a typical handling of the SWI.

```
stmfd        sp!, {r0-r1,lr}       @ push register onto SWI stack
mrs         r0, spsr        @ get saved status register
tst          r0, #0x20     @ check if call was in THUMB mode
ldrneh      r0, [lr,#-2]       @ yes: load opcode half-word and
bicne       r0, r0, #0xff00   @ yes: extract THUMB comment
ldreq       r0, [lr,#-4]       @ no: load opcode word and
biceq       r0, r0, #0xff000000 @ no: extract ARM comment
                  @ now r0 has comment field
cmp        r0, #16      @ compare to 16 (maximum for ROMC)
ldrlt         lr, =rompatch_tbl_ptr @ < 16: get top of current ROMC
                  @ table; global variable which is
                  @ changeable per context
ldrlt         r1, [lr, r0, lsl #2]   @ < 16: read function pointer from
                  @ table assumed an array of pointers
                  @ patch functions
strlt          r1, [sp, #8]     @ < 16: store function pointer onto
                  @ stack in position of link register
ldmltfd       sp!, {r0-r1,pc}^      @ < 16: "fake" return from SWI, will
                  @ vector core to appropriate patch
                  @ function and set core back to previous
                  @ mode of operating
ldr         r1, =swi_hdler   @ >= 16: pointer to standard SWI
                  @ handler
mov        lr, pc         @ >= 16: set link register
bx         r1          @ >= 16: jump to standard SWI
                  @ handler
ldmfd       sp!, {r0-r1,pc}^      @ >= 16: pop registers from stack
```

4.9.4.2.5 External Boot Feature

Following a Reset event, the Arm issues an instruction fetch of the Reset Vector from address 0x0. This instruction, normally residing in ROM is usually a branch to a Reset handler or boot code which also normally resides in ROM. The ROMC external boot feature allows the bypassing of this code, using a different boot code residing perhaps in external memory.

This feature uses the data fix mechanism and works as follows: if the boot_int signal is negated when a Reset event occurred, the ROMC will perform a data fix of the Reset Vector at 0x0 with the following instruction (opcode 0xE59FF00C):

```
ldr         pc,[pc,#12]      @ read 0x0000_0014 for reset_vector
```

The value of PC when this instruction is executed is 8 so that a PC relative offset of 12 makes the source address 20 or 0x14. When this instruction executes, the Arm core reads from address 0x0000_0014, triggering a ROMC data fix operation which places the value taken from the external boot address on the read data bus, with the two LSBs zeroed out. This value is returned to the Arm to be placed in the PC causing code fetch and execution to start from that address.

4.9.4.2.6 Alternate Masters and ROMC

The ROMC sits on the AHB bus of the internal ROM (ROMC). This means that the ROMC can modify values on the read data bus going to the master. Therefore, any master which reads an opcode patched or data patched location will read patched data.

4.9.5 ROMCP Memory Map/Register Definition

All registers are accessible through an IP Bus and can only be accessed in privileged mode. These registers can only be written with 32-bits stores and are clocked by hclk_reg.

ROMC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3031_00D4	ROMC Data Registers (ROMC_ROMPATCH7D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00D8	ROMC Data Registers (ROMC_ROMPATCH6D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00DC	ROMC Data Registers (ROMC_ROMPATCH5D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00E0	ROMC Data Registers (ROMC_ROMPATCH4D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00E4	ROMC Data Registers (ROMC_ROMPATCH3D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00E8	ROMC Data Registers (ROMC_ROMPATCH2D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00EC	ROMC Data Registers (ROMC_ROMPATCH1D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00F0	ROMC Data Registers (ROMC_ROMPATCH0D)	32	R/W	0000_0000h	4.9.5.1/236
3031_00F4	ROMC Control Register (ROMC_ROMPATCHCNTL)	32	R/W	0840_0000h	4.9.5.2/237
3031_00F8	ROMC Enable Register High (ROMC_ROMPATCHENH)	32	R	0000_0000h	4.9.5.3/238
3031_00FC	ROMC Enable Register Low (ROMC_ROMPATCHENL)	32	R/W	0000_0000h	4.9.5.4/238
3031_0100	ROMC Address Registers (ROMC_ROMPATCH0A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0104	ROMC Address Registers (ROMC_ROMPATCH1A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0108	ROMC Address Registers (ROMC_ROMPATCH2A)	32	R/W	0000_0000h	4.9.5.5/239
3031_010C	ROMC Address Registers (ROMC_ROMPATCH3A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0110	ROMC Address Registers (ROMC_ROMPATCH4A)	32	R/W	0000_0000h	4.9.5.5/239

Table continues on the next page...

ROMC memory map (continued)

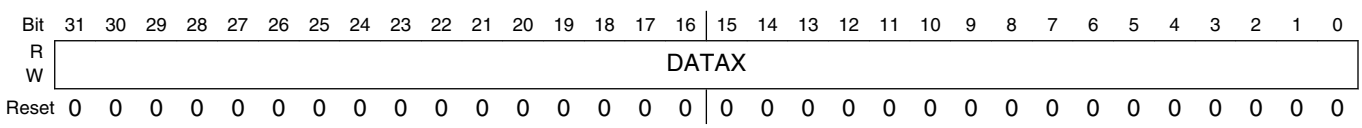
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3031_0114	ROMC Address Registers (ROMC_ROMPATCH5A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0118	ROMC Address Registers (ROMC_ROMPATCH6A)	32	R/W	0000_0000h	4.9.5.5/239
3031_011C	ROMC Address Registers (ROMC_ROMPATCH7A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0120	ROMC Address Registers (ROMC_ROMPATCH8A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0124	ROMC Address Registers (ROMC_ROMPATCH9A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0128	ROMC Address Registers (ROMC_ROMPATCH10A)	32	R/W	0000_0000h	4.9.5.5/239
3031_012C	ROMC Address Registers (ROMC_ROMPATCH11A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0130	ROMC Address Registers (ROMC_ROMPATCH12A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0134	ROMC Address Registers (ROMC_ROMPATCH13A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0138	ROMC Address Registers (ROMC_ROMPATCH14A)	32	R/W	0000_0000h	4.9.5.5/239
3031_013C	ROMC Address Registers (ROMC_ROMPATCH15A)	32	R/W	0000_0000h	4.9.5.5/239
3031_0208	ROMC Status Register (ROMC_ROMPATCHSR)	32	w1c	0000_0000h	4.9.5.6/240

4.9.5.1 ROMC Data Registers (ROMC_ROMPATCH*n*D)

The ROMC data registers (ROMC_ROMPATCH 7D through ROMC_ROMPATCH 0D) store the data to use for the 8 1-word data fix events. Each register is associated with an address comparator (7 through 0). When a data fixing event occurs, the value in the data register corresponding to the comparator that has the address match is put on the romc_hrdata[31:0] bus until romc_hready is asserted by the ROM controller to terminate the access. A MUX external to the ROMC will select this data over that of romc_hrdata[31:0] in returning read data to the Arm core. The selection is done with the control bus rompatch_romc_hrdata_ovr[1:0] with both bits asserted by the ROMC.

If more than one address comparators match, the highest-numbered one takes precedence, and the value in corresponding data register is used for the patching event.

Address: 3031_0000h base + D4h offset + (4d × i), where i=0d to 7d



ROMC_ROMPATCH*n*D field descriptions

Field	Description
DATAX	Data Fix Registers - Stores the data used for 1-word data fix operations. The values stored within these registers do not affect the writes to the memory system. They are selected over the read data from ROM when a data fix event occurs. If any part of the 1-word data fix is read, then the entire word is replaced. Therefore, a byte or half-word read will cause the ROMC to replace the entire word. The word is word address aligned.

4.9.5.2 ROMC Control Register (ROMC_ROMPATCHCNTL)

The ROMC control register (ROMC_ROMPATCHCNTL) contains the block disable bit and the data fix enable bits. The block disable bit provides a means to disable the ROMC data fix and opcode patching functions, even when the address comparators are enabled. The External Boot feature is not affected by this bit. The eight data fix enable bits (0 through 7), when set, assign the associated address comparators to data fix operations

NOTE

Bits 27 and 22 always read as 1s.

Address: 3031_0000h base + F4h offset = 3031_00F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		DIS	Reserved												
Reset	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DATAFIX							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHCNTL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29 DIS	ROMC Disable -- This bit, when set, disables all ROMC operations. This bit is used to enable secure operations. 0 Does not affect any ROMC functions (default) 1 Disable all ROMC functions: data fixing, and opcode patching
28–8 -	This field is reserved. Reserved
DATAFIX	**Data Fix Enable - Controls the use of the first 8 address comparators for 1-word data fix or for code patch routine.** 0 Address comparator triggers a opcode patch 1 Address comparator triggers a data fix

4.9.5.3 ROMC Enable Register High (ROMC_ROMPATCHENH)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_OMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: 3031_0000h base + F8h offset = 3031_00F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHENH field descriptions

Field	Description
-	This field is reserved. Reserved

4.9.5.4 ROMC Enable Register Low (ROMC_ROMPATCHENL)

The ROMC enable register high (ROMC_ROMPATCHENH) and ROMC enable register low (ROMC_ROMPATCHENL) control whether or not the associated address comparator can trigger a opcode patch or data fix event. This implementation of the ROMC only has 16 comparators, therefore ROMC_ROMPATCHENH and the upper half of ROMC_ROMPATCHENL are read-only. ROMC_ROMPATCHENL[15:0] are associated with comparators 15 through 0. ROMC_ROMPATCHENLH[31:0] would have been associated with comparators 63 through 32.

Address: 3031_0000h base + FCh offset = 3031_00FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ENABLE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

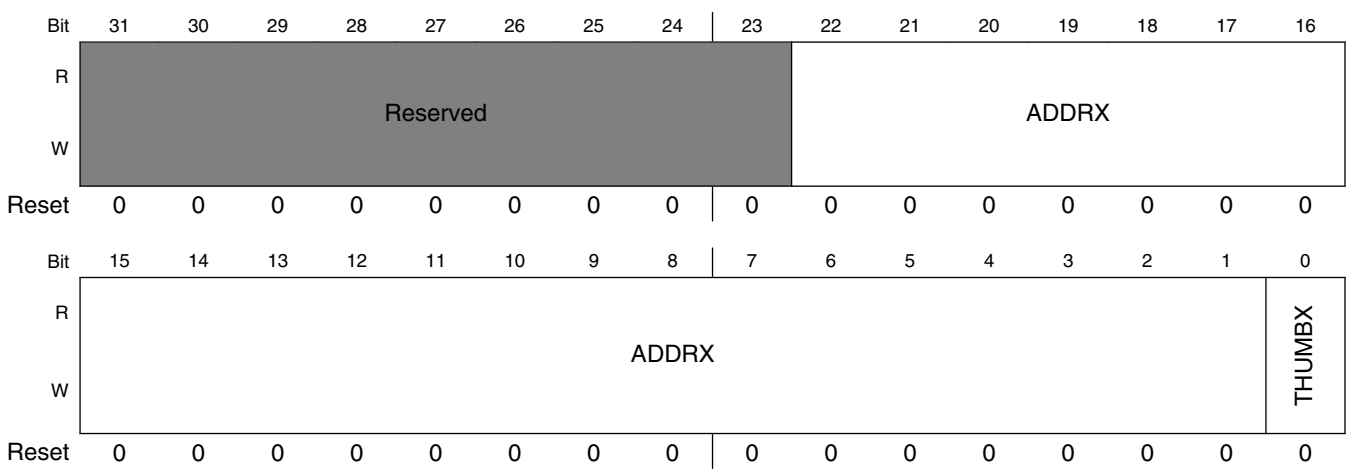
ROMC_ROMPATCHENL field descriptions

Field	Description
31–16 Reserved	This field is reserved.
ENABLE	**Enable** Address **Comparator -** This bit enables the corresponding address comparator to trigger an event. 0 Address comparator disabled 1 Address comparator enabled, ROMC will trigger a opcode patch or data fix event upon matching of the associated address

4.9.5.5 ROMC Address Registers (ROMC_ROMPATCH*n*A)

The ROMC address registers (ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15) store the memory addresses where opcode patching begins and data fixing occurs. The address registers ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are each 21 bits wide and dedicated to one 4 Mbyte memory space. Bits 21 through 2 are address bits, to be compared with romc_haddr[21:2] for a match; bit 1 is also an address bit used for half word selection. Bit 0 is the mode bit (set to 1 for THUMB mode). 1-word data fixing can only be used on the first 8 of the address comparators. ROMC_ROMPATCHA0 through ROMC_ROMPATCHA15 are associated each with address comparators 0 through 15.

Address: 3031_0000h base + 100h offset + (4d × i), where i=0d to 15d



ROMC_ROMPATCH*n*A field descriptions

Field	Description
31–23 -	This field is reserved. Reserved

Table continues on the next page...

ROMC_ROMPATCH*n*A field descriptions (continued)

Field	Description
22–1 ADDRX	Address Comparator Registers - Indicates the memory address to be watched. All 16 registers can be used for code patch address comparison. Only the first 8 registers can be used for a 1-word data fix address comparison. Bit 1 is ignored if data fix. Only used in code patch
0 THUMBX	THUMB Comparator Select - Indicates that this address will trigger a THUMB opcode patch or an Arm opcode patch. If this watchpoint is selected to be a data fix, then this bit is ignored as all data fixes are 1-word data fixes. 0 Arm patch 1 THUMB patch (ignore if data fix)

4.9.5.6 ROMC Status Register (ROMC_ROMPATCHSR)

The ROMC status register (ROMC_ROMPATCHSR) indicates the current state of the ROMC and the source number of the most recent address comparator event.

Address: 3031_0000h base + 208h offset = 3031_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														SW	Reserved
W															w1c	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										SOURCE					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ROMC_ROMPATCHSR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17 SW	ROMC AHB Multiple Address Comparator matches Indicator - Indicates that multiple address comparator matches occurred. Writing a 1 to this bit will clear this it. 0 no event or comparator collisions 1 a collision has occurred
16–6 -	This field is reserved. Reserved
SOURCE	ROMC Source Number - Binary encoding of the number of the address comparator which has an address match in the most recent patch event on ROMC AHB. If multiple matches occurred, the highest priority source number is used.

Table continues on the next page...

ROMC_ROMPATCHSR field descriptions (continued)

Field	Description
	0 Address Comparator 0 matched 1 Address Comparator 1 matched 15 Address Comparator 15 matched

4.10 Watchdog Timer (WDOG)

4.10.1 Overview

The Watchdog Timer (WDOG) protects against system failures by providing a method by which to escape from unexpected events or programming errors.

Once the WDOG is activated, it must be serviced by the software on a periodic basis. If servicing does not take place, the timer times out. Upon timeout, the WDOG asserts the internal system reset signal, WDOG_RESET_B_DEB to the System Reset Controller (SRC).

There is also a provision for WDOG signal assertion by timeout counter expiration. There is an option of programmable interrupt generation before the counter actually times out. The time at which the interrupt needs to be generated prior to counter timeout is programmable. There is a power down counter which is enabled out of any reset (POR, Warm/Cold). This counter has a fixed timeout period of 16 seconds, upon which it asserts the WDOG signal.

Flow diagrams for the timeout counter, power down counter and interrupt operations are are shown in Flow Diagrams.

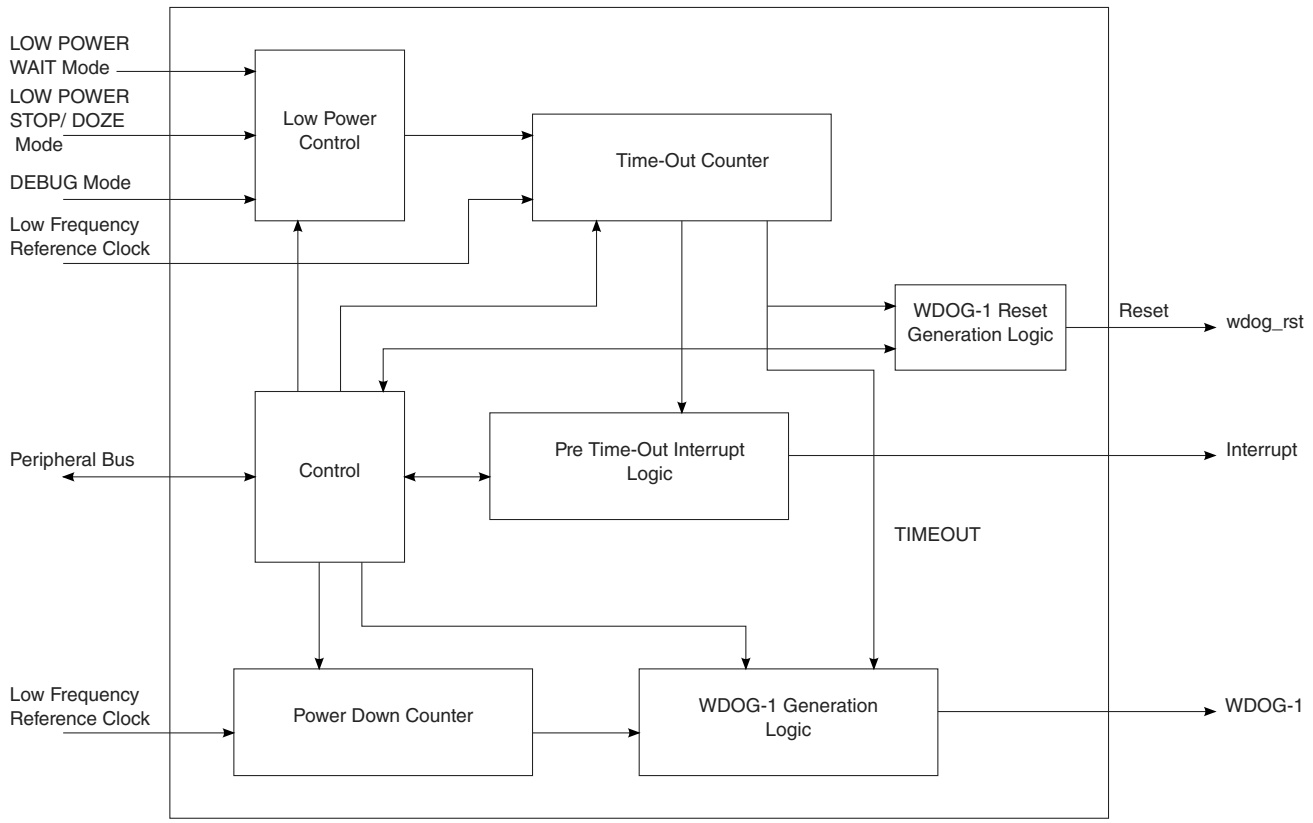


Figure 4-20. WDOG Diagram

4.10.1.1 Features

The WDOG features are listed below:

- Configurable timeout counter with timeout periods from 0.5 to 128 seconds which, after timeout expiration, result in the assertion of WDOG_RESET_B_DEB reset signal .
- Time resolution of 0.5 seconds
- Configurable timeout counter that can be programmed to run or stop during low-power modes
- Configurable timeout counter that can be programmed to run or stop during DEBUG mode
- Programmable interrupt generation prior to timeout
- The duration between interrupt and timeout events can be programmed from 0 to 127.5 seconds in steps of 0.5 seconds.
- Power down counter with fixed timeout period of 16 seconds, which if not disabled after reset will assert WDOG_B signal low

- Power down counter will be enabled out of any reset (POR, Warm / Cold reset) by default.

4.10.2 Clocks

This section describes clocks and special clocking requirements of the block.

The WDOG uses the low frequency reference clock for its counter and control operations. The peripheral bus clock is used for register read/write operations.

The following table describes the clock sources for WDOG. Please see Introduction for clock setting, configuration and gating information.

Table 4-26. WDOG Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	IP Global functional clock. All functionality inside the WDOG module is synchronized to this clock.
ipg_clk_s	ipg_clk_root	IP slave bus clock. This clock is synchronized to ipg_clk and is only used for register read/write operations.
ipg_clk_32k	ckil_sync_clk_root	Low frequency (32.768 kHz) clock that continues to run in low-power mode. It is assumed that the Clock Controller will provide this clock signal synchronized to ipg_clk in the normal mode, and switch to a non-synchronized signal in low-power mode when the ipg_clk is off.

4.10.3 Functional description

This section provides a complete functional description of the block.

4.10.3.1 Timeout event

The WDOG provides timeout periods from 0.5 to 128 seconds with a time resolution of 0.5 seconds.

The user can determine the timeout period by writing to the WDOG timeout field (WT[7:0]) in the Watchdog Control Register (WDOG_WCR). The WDOG must be enabled by setting the WDE bit of Watchdog Control Register (WDOG_WCR) for the timeout counter to start running. After the WDOG is enabled, the counter is activated, loads the timeout value and begins to count down from this programmed value. The timer will time out when the counter reaches zero and the WDOG outputs a system reset signal, WDOG_RESET_B_DEB and asserts WDOG_B (WDT bit should be set in Watchdog Control Register (WDOG_WCR)).

However, the timeout condition can be prevented by reloading the counter with the new timeout value (WT[7:0] of WDOG_WCR) if a service routine (see Servicing WDOG to reload the counter) is performed before the counter reaches zero. If any system errors occur which prevent the software from servicing the Watchdog Service Register (WDOG_WSR), the timeout condition occurs. By performing the service routine, the WDOG reloads its counter to the timeout value indicated by bits WT[7:0] of the Watchdog Control Register (WDOG_WCR) and it restarts the countdown.

A system reset will reset the counter and place it in the idle state at any time during the countdown. The counter flow diagram is shown in Flow Diagrams.

NOTE

The timeout value is reloaded to the counter either at the time WDOG is enabled or after the service routine has been performed.

4.10.3.1.1 Servicing WDOG to reload the counter

To reload a timeout value to the counter the proper service sequence begins by writing 0x_5555 followed by 0x_AAAA to the Watchdog Service Register (WDOG_WSR). Any number of instructions can be executed between the two writes. If the WDOG_WSR is not loaded with 0x_5555 prior to writing 0x_AAAA to the WDOG_WSR, the counter is not reloaded. If any value other than 0x_AAAA is written to the WDOG_WSR after 0x_5555, the counter is not reloaded. This service sequence will reload the counter with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR). The timeout value can be changed at any point; it is reloaded when WDOG is serviced by the core.

4.10.3.2 Interrupt event

Prior to timeout, the WDOG can generate an interrupt which can be considered a warning that timeout will occur shortly.

The duration between interrupt event and timeout event can be controlled by writing to the WICT field of Watchdog Interrupt Control Register (WDOG_WICR). It can vary between 0 and 127.5 seconds. If the WDOG is serviced (Servicing WDOG to reload the counter)before the interrupt generation, the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR) and the interrupt will not be triggered.

4.10.3.3 Power-down counter event

The power-down counter inside WDOG will be enabled out of reset. This counter has a fixed timeout value of 16 seconds, after which it will drive the WDOG_B signal low.

To prevent this, the software must disable this counter by clearing the PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) within 16 seconds of reset deassertion. Once disabled, this counter can't be enabled again until the next system reset occurs. This feature is intended to prevent the hanging up of cores after reset, as WDOG is not enabled out of reset.

4.10.3.4 Low power modes

4.10.3.4.1 STOP and DOZE mode

If the WDOG timer disable bit for low power STOP and DOZE mode (WDZST) bit in the Watchdog Control Register (WDOG_WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock. If the low power enable (WDZST) bit is set, the WDOG timer operation will be suspended in low power STOP or DOZE mode. Upon exiting low power STOP or DOZE mode, the WDOG operation returns to what it was prior to entering the STOP or DOZE mode.

4.10.3.4.2 WAIT mode

If the WDOG timer disable bit for low power WAIT mode (WDW) bit in the Watchdog Control Register (WDOG_WCR), is cleared, the WDOG timer continues to operate using the low frequency reference clock . If the low power WAIT enable (WDW) bit is set, the WDOG timer operation will be suspended. Upon exiting low power WAIT mode, the WDOG operation returns to what it was prior to entering the WAIT mode.

NOTE

The WDOG timer won't be able to detect events that happen for periods shorter than one low frequency reference clock cycle. For example, in repeated WAIT mode entry or exit, if the RUN mode time is less than one low frequency reference clock cycle and if the WDW bit is set, the WDOG timer may never time out, even though the system is in RUN mode for a finite duration; WDOG may not see a low frequency reference clock edge during its wake time.

4.10.3.5 Debug mode

The WDOG timer can be configured for continual operation, or for suspension during debug mode. If the WDOG debug enable (WDBG) bit is set in the Watchdog Control Register (WDOG_WCR), the WDOG timer operation is suspended in debug mode. If the WDBG bit is set and the debug mode is entered, WDOG timer operation is suspended after two low frequency reference clocks. Similarly, WDOG timer operation continues after two low frequency reference clocks of debug mode exit. Register read and write accesses in debug mode continue to function normally. Also, while in debug mode, the WDE bit of Watchdog Control Register (WDOG_WCR) can be enabled/disabled directly. If the WDOG debug enable (WDBG) bit is cleared then WDOG timer operation is not suspended. The power-down counter is not affected by debug mode entry/exit.

NOTE

If the WDE bit of Watchdog Control Register (WDOG_WCR) is set/cleared while in debug mode, it remains set/cleared even after exiting debug mode.

4.10.3.6 Operations

4.10.3.6.1 Watchdog reset generation

The WDOG generated reset signal WDOG_RESET_B_DEB is asserted by the following operations:

- A software write to the Software Reset Signal (SRS) bit of the Watchdog Control Register (WDOG_WCR).
- WDOG timeout. See Timeout event.

The $\overline{\text{wdog\_rst}}$ will be asserted for one clock cycle of low frequency reference clock for both a timeout condition and a software write occurrence. It remains asserted for 1 clock cycle of low frequency reference clock even if a system reset is asserted in between. Figure 4-22 shows the timing diagram of this signal due to a timeout condition.

4.10.3.6.2 WDOG_B generation

The WDOG asserts WDOG_B in the following scenarios:

- Software write to WDA bit of Watchdog Control Register (WDOG_WCR). WDOG_B signal remains asserted as long as the WDA bit is "0".
- WDOG timeout condition, WDT bit of Watchdog Control Register (WDOG_WCR) must be set for this scenario. A description of the timeout condition can be found in the Timeout event. WDOG_B signal remains asserted until a power-on reset (POR)

occurs. It gets cleared after the POR occurs (not due to any other system reset). Figure 4-23 shows the timing diagram of WDOG_B due to timeout condition.

- WDOG power-down counter timeout, PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should not be cleared for this scenario. A description of this counter can be found in the Power-down counter event. WDOG_B signal remains asserted for one clock cycle of low frequency reference clock.

Figure 4-21 shows the scenarios under which WDOG_B gets asserted.

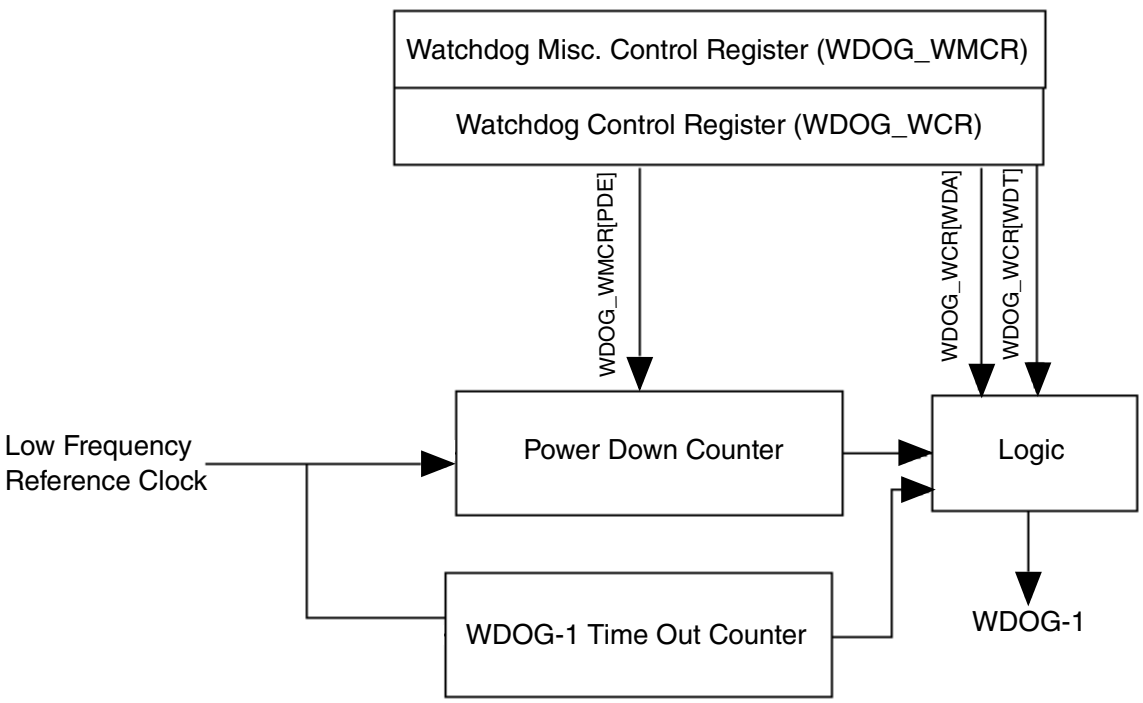


Figure 4-21. WDOG_B generation

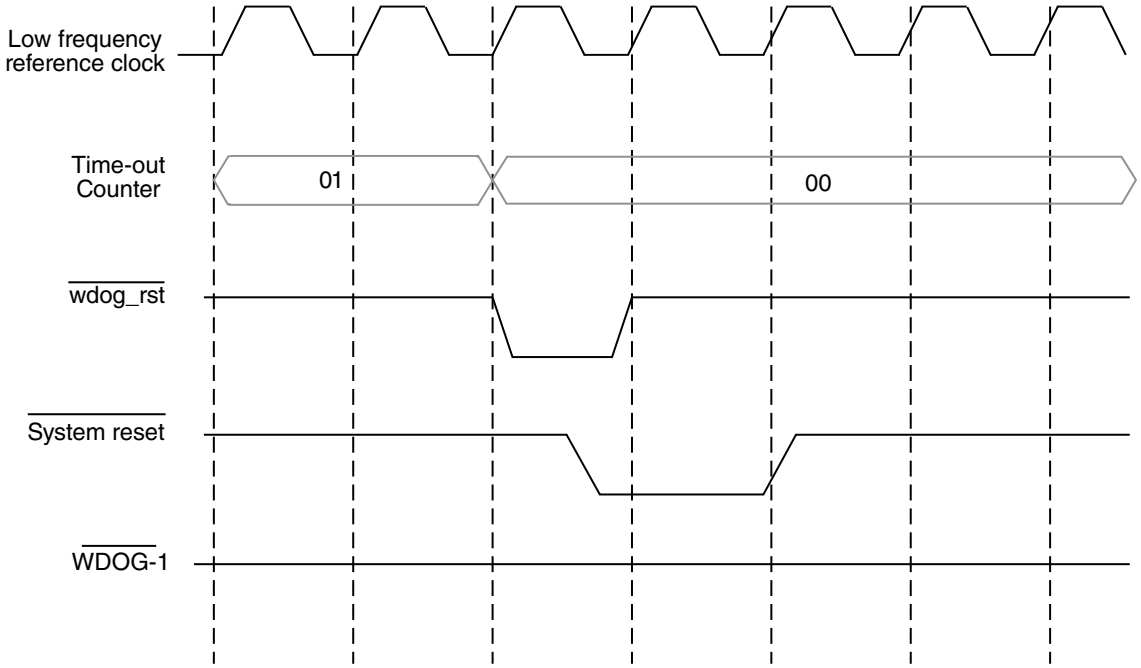


Figure 4-22. WDOG timeout condition/WDT bit is not set

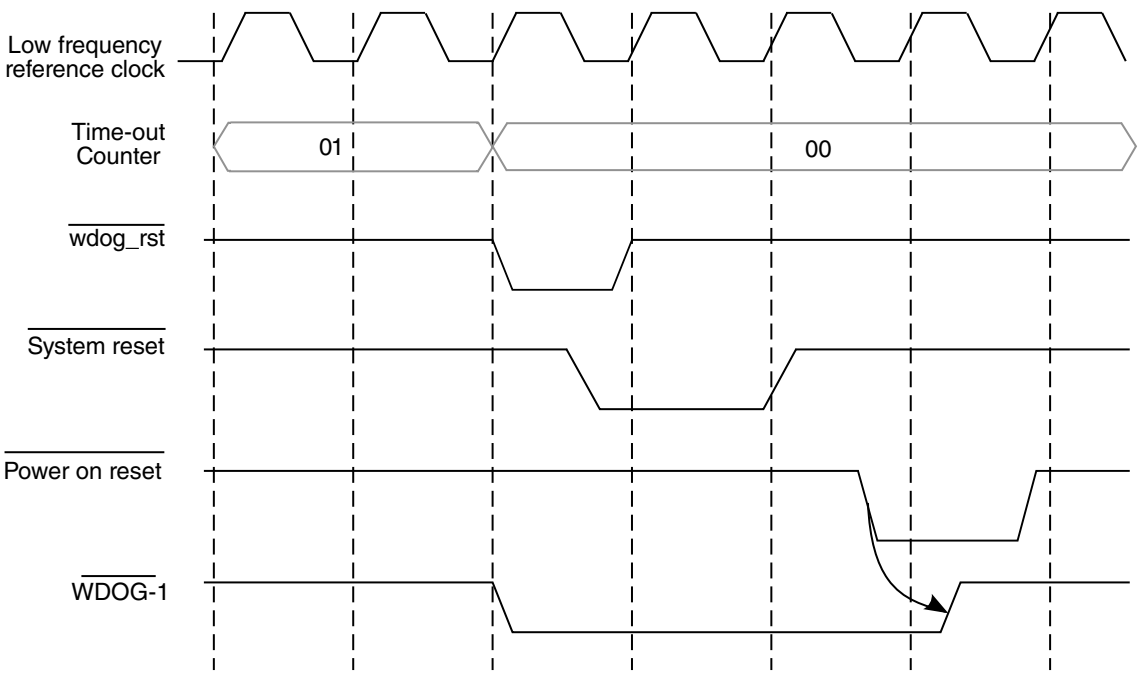


Figure 4-23. WDOG timeout condition/WDT bit is set

4.10.3.7 Reset

The block is reset by a system reset and the WDOG counter will be disabled. The power-down counter is enabled and starts counting.

4.10.3.8 Interrupt

The WDOG has the feature of Interrupt generation before timeout.

The interrupt will be generated only if the WIE bit in Watchdog Interrupt Control Register (WDOG_WICR) is set. The exact time at which the interrupt should occur (prior to timeout) depends on the value of WICT field of Watchdog Interrupt Control Register (WDOG_WICR). For example, if the WICT field has a value 0x04, then the interrupt will be generated two seconds prior to timeout. Once the interrupt is triggered the WTIS bit in Watchdog Interrupt Control Register (WDOG_WICR) will be set. The software needs to clear this bit to deassert the interrupt. If the WDOG is serviced before the interrupt generation then the counter will be reloaded with the timeout value WT[7:0] of Watchdog Control Register (WDOG_WCR) and interrupt would not be triggered.

4.10.3.9 Flow Diagrams

A flow diagram of WDOG operation is shown below.

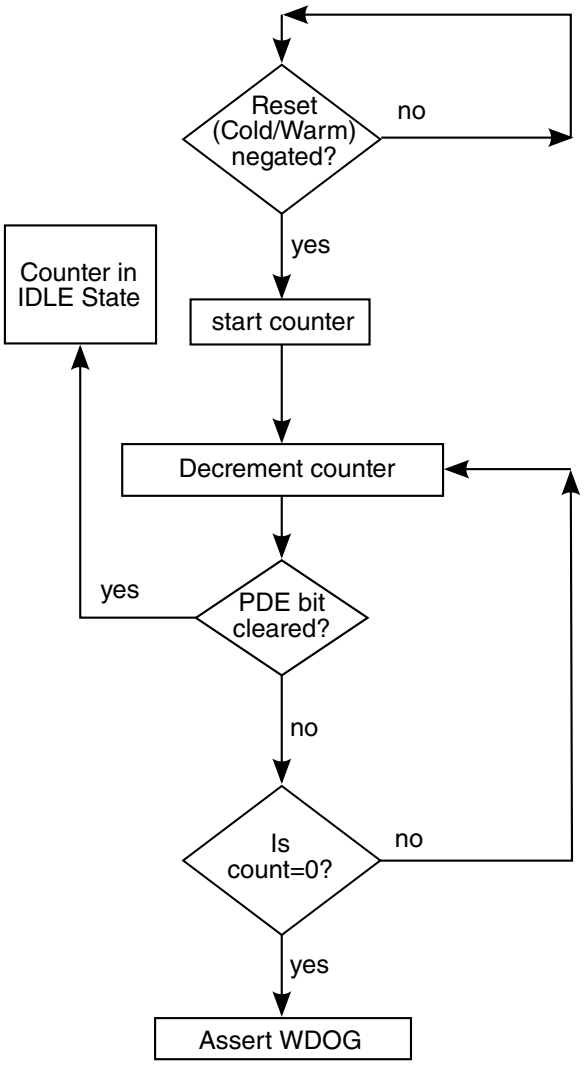


Figure 4-24. Power-Down Counter Flow Diagram

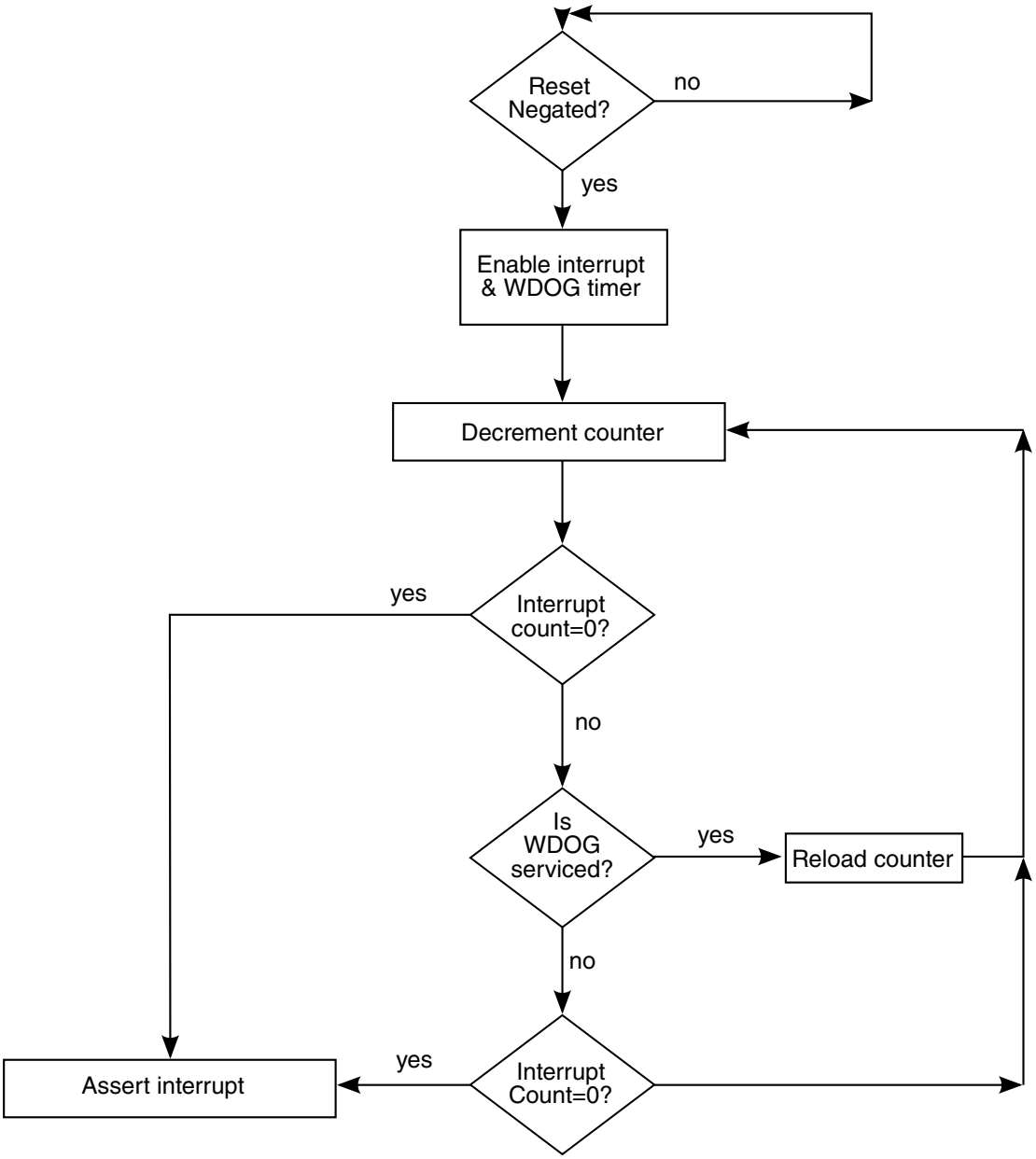


Figure 4-25. Interrupt Generation Flow Diagram

4.10.4 Initialization

The following sequence should be performed for WDOG initialization.

- PDE bit of Watchdog Miscellaneous Control Register (WDOG_WMCR) should be cleared to disable the power down counter.

- WT field of Watchdog Control Register (WDOG_WCR) should be programmed for sufficient timeout value.
- WDOG should be enabled by setting WDE bit of Watchdog Control Register (WDOG_WCR) so that the timeout counter loads the WT field value of Watchdog Control Register (WDOG_WCR) and starts counting.

4.10.5 WDOG Memory Map/Register Definition

The WDOG Memory Map/Register Definition can be found here.

The WDOG has user-accessible, 16-bit registers used to configure, operate, and monitor the state of the Watchdog Timer. Byte operations can be performed on these registers. If a 32-bit access is performed,the WDOG will not generate a peripheral bus error but will behave normally, like a 16-Bit access, making read/write possible. A 32-Bit access should be avoided, as the system may go to an unknown state.

WDOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3028_0000	Watchdog Control Register (WDOG1_WCR)	16	R/W	0030h	4.10.5.1/ 253
3028_0002	Watchdog Service Register (WDOG1_WSR)	16	R/W	0000h	4.10.5.2/ 255
3028_0004	Watchdog Reset Status Register (WDOG1_WRSR)	16	R	0000h	4.10.5.3/ 255
3028_0006	Watchdog Interrupt Control Register (WDOG1_WICR)	16	R/W	0004h	4.10.5.4/ 256
3028_0008	Watchdog Miscellaneous Control Register (WDOG1_WMCR)	16	R/W	0001h	4.10.5.5/ 257
3029_0000	Watchdog Control Register (WDOG2_WCR)	16	R/W	0030h	4.10.5.1/ 253
3029_0002	Watchdog Service Register (WDOG2_WSR)	16	R/W	0000h	4.10.5.2/ 255
3029_0004	Watchdog Reset Status Register (WDOG2_WRSR)	16	R	0000h	4.10.5.3/ 255
3029_0006	Watchdog Interrupt Control Register (WDOG2_WICR)	16	R/W	0004h	4.10.5.4/ 256
3029_0008	Watchdog Miscellaneous Control Register (WDOG2_WMCR)	16	R/W	0001h	4.10.5.5/ 257
302A_0000	Watchdog Control Register (WDOG3_WCR)	16	R/W	0030h	4.10.5.1/ 253

Table continues on the next page...

WDOG memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302A_0002	Watchdog Service Register (WDOG3_WSR)	16	R/W	0000h	4.10.5.2/ 255
302A_0004	Watchdog Reset Status Register (WDOG3_WRSR)	16	R	0000h	4.10.5.3/ 255
302A_0006	Watchdog Interrupt Control Register (WDOG3_WICR)	16	R/W	0004h	4.10.5.4/ 256
302A_0008	Watchdog Miscellaneous Control Register (WDOG3_WMCR)	16	R/W	0001h	4.10.5.5/ 257

4.10.5.1 Watchdog Control Register (WDOGx_WCR)

The Watchdog Control Register (WDOG_WCR) controls the WDOG operation.

- WDZST, WDBG and WDW are write-once only bits. Once the software does a write access to these bits, they will be locked and cannot be reprogrammed until the next system reset assertion.
- WDE is a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next system reset.
- WDT is also a write one once only bit. Once software performs a write "1" operation to this bit it cannot be reset/cleared until the next POR. This bit does not get reset/ cleared due to any system reset.

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8
Read Write	WT							
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read Write	WDW	SRE	WDA	SRS	WDT	WDE	WDBG	WDZST
Reset	0	0	1	1	0	0	0	0

WDOGx_WCR field descriptions

Field	Description
15–8 WT	Watchdog Time-out Field. This 8-bit field contains the time-out value that is loaded into the Watchdog counter after the service routine has been performed or after the Watchdog is enabled. After reset, WT[7:0] must have a value written to it before enabling the Watchdog otherwise count value of zero which is 0.5 seconds is loaded into the counter. **NOTE:** The time-out value can be written at any point of time but it is loaded to the counter at the time when WDOG is enabled or after the service routine has been performed. For more information see Timeout event .

Table continues on the next page...

WDOGx_WCR field descriptions (continued)

Field	Description
	0x00 - 0.5 Seconds (Default). 0x01 - 1.0 Seconds. 0x02 - 1.5 Seconds. 0x03 - 2.0 Seconds. 0xff - 128 Seconds.
7 WDW	Watchdog Disable for Wait. This bit determines the operation of WDOG during Low Power WAIT mode. This is a write once only bit. 0 Continue WDOG timer operation (Default). 1 Suspend WDOG timer operation.
6 SRE	Software Reset Extension. Required to be set to 1 when used in conjunction with the Software Reset Signal (SRS). 0 Reserved 1 This bit must be set to 1.
5 WDA	WDOG_B assertion. Controls the software assertion of the WDOG_B signal. 0 Assert WDOG_B output. 1 No effect on system (Default).
4 SRS	Software Reset Signal. Controls the software assertion of the WDOG-generated reset signal WDOG_RESET_B_DEB. This bit automatically resets to 1 after it has been asserted to 0. For proper operation of this function, the SRE bit in this register must be set to 1. **NOTE:** This bit does not generate the software reset to the block. 0 Assert system reset signal. 1 No effect on the system (Default).
3 WDT	WDOG_B Time-out assertion. Determines if the WDOG_B gets asserted upon a Watchdog Time-out Event. This is a write-one once only bit. **NOTE:** There is no effect on WDOG_RESET_B_DEB (WDOG Reset) upon writing on this bit. WDOG_B gets asserted along with WDOG_RESET_B_DEB if this bit is set. 0 No effect on WDOG_B (Default). 1 Assert WDOG_B upon a Watchdog Time-out event.
2 WDE	Watchdog Enable. Enables or disables the WDOG block. This is a write one once only bit. It is not possible to clear this bit by a software write, once the bit is set. **NOTE:** This bit can be set/reset in debug mode (exception). 0 Disable the Watchdog (Default). 1 Enable the Watchdog.
1 WDBG	Watchdog DEBUG Enable. Determines the operation of the WDOG during DEBUG mode. This bit is write once only. 0 Continue WDOG timer operation (Default). 1 Suspend the watchdog timer.
0 WDZST	Watchdog Low Power. Determines the operation of the WDOG during low-power modes. This bit is write once-only. **NOTE:** The WDOG can continue/suspend the timer operation in the low-power modes (STOP and DOZE mode).

Table continues on the next page...

WDOGx_WCR field descriptions (continued)

Field	Description
	0 Continue timer operation (Default). 1 Suspend the watchdog timer.

4.10.5.2 Watchdog Service Register (WDOGx_WSR)

When enabled, the WDOG requires that a service sequence be written to the Watchdog Service Register (WSR) to prevent the timeout condition.

NOTE

Executing the service sequence will reload the WDOG timeout counter.

Address: Base address + 2h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read Write	WSR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

WDOGx_WSR field descriptions

Field	Description
WSR	Watchdog Service Register. This 16-bit field contains the Watchdog service sequence. Both writes must occur in the order listed prior to the time-out, but any number of instructions can be executed between the two writes. The service sequence must be performed as follows: 0x5555 Write to the Watchdog Service Register (WDOG_WSR). 0xAAAA Write to the Watchdog Service Register (WDOG_WSR).

4.10.5.3 Watchdog Reset Status Register (WDOGx_WRSR)

The WRSR is a read-only register that records the source of the output reset assertion. It is not cleared by a hard reset. Therefore, only one bit in the WRSR will always be asserted high. The register will always indicate the source of the last reset generated due to WDOG. Read access to this register is with one wait state. Any write performed on this register will generate a Peripheral Bus Error .

A reset can be generated by the following sources, as listed in priority from highest to lowest:

- Watchdog Time-out
- Software Reset

Address: Base address + 4h offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	0			POR	0		TOUT	SFTW
Write								
Reset	0	0	0	0	0	0	0	0

WDOGx_WRSR field descriptions

Field	Description
15–5 Reserved	This read-only field is reserved and always has the value 0.
4 POR	Power On Reset. Indicates whether the reset is the result of a power on reset. 0 Reset is not the result of a power on reset. 1 Reset is the result of a power on reset.
3–2 Reserved	This read-only field is reserved and always has the value 0.
1 TOUT	Timeout. Indicates whether the reset is the result of a WDOG timeout. 0 Reset is not the result of a WDOG timeout. 1 Reset is the result of a WDOG timeout.
0 SFTW	Software Reset. Indicates whether the reset is the result of a WDOG software reset by asserting SRS bit 0 Reset is not the result of a software reset. 1 Reset is the result of a software reset.

4.10.5.4 Watchdog Interrupt Control Register (WDOGx_WICR)

The WDOG_WICR controls the WDOG interrupt generation.

Address: Base address + 6h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	WIE	WTIS	0						WICT							
Write		w1c														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

WDOGx_WICR field descriptions

Field	Description
15 WIE	Watchdog Timer Interrupt enable bit. Reset value is 0.

Table continues on the next page...

WDOGx_WICR field descriptions (continued)

Field	Description
	NOTE: This bit is a write once only bit. Once the software does a write access to this bit, it will get locked and cannot be reprogrammed until the next system reset assertion 0 Disable Interrupt (Default). 1 Enable Interrupt.
14 WTIS	Watchdog Timer Interrupt Status bit will reflect the timer interrupt status, whether interrupt has occurred or not. Once the interrupt has been triggered software must clear this bit by writing 1 to it. 0 No interrupt has occurred (Default). 1 Interrupt has occurred
13–8 Reserved	This read-only field is reserved and always has the value 0.
WICT	Watchdog Interrupt Count Time-out (WICT) field determines, how long before the counter time-out must the interrupt occur. The reset value is 0x04 implies interrupt will occur 2 seconds before time-out. The maximum value that can be programmed to WICT field is 127.5 seconds with a resolution of 0.5 seconds. **NOTE:** This field is write once only. Once the software does a write access to this field, it will get locked and cannot be reprogrammed until the next system reset assertion. 0x00 WICT[7:0] = Time duration between interrupt and time-out is 0 seconds. 0x01 WICT[7:0] = Time duration between interrupt and time-out is 0.5 seconds. 0x04 WICT[7:0] = Time duration between interrupt and time-out is 2 seconds (Default). 0xff WICT[7:0] = Time duration between interrupt and time-out is 127.5 seconds.

4.10.5.5 Watchdog Miscellaneous Control Register (WDOGx_WMCR)

WDOG_WMCR Controls the Power Down counter operation.

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0															PDE
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

WDOGx_WMCR field descriptions

Field	Description
15–1 Reserved	This read-only field is reserved and always has the value 0.
0 PDE	Power Down Enable bit. Reset value of this bit is 1, which means the power down counter inside the WDOG is enabled after reset. The software must write 0 to this bit to disable the counter within 16 seconds of reset de-assertion. Once disabled this counter cannot be enabled again. See Power-down counter event for operation of this counter. **NOTE:** This bit is write-one once only bit. Once software sets this bit it cannot be reset until the next system reset.

Table continues on the next page...

WDOGx_WMCR field descriptions (continued)

Field	Description
	0 Power Down Counter of WDOG is disabled. 1 Power Down Counter of WDOG is enabled (Default).

4.11 TrustZone Address Space Controller (TZASC)

4.11.1 Overview

The TrustZone Address Space Controller (TZASC) protects security-sensitive SW and data in a trusted execution environment against potentially compromised SW running on the platform.

The TZASC block diagram is shown in figure below.

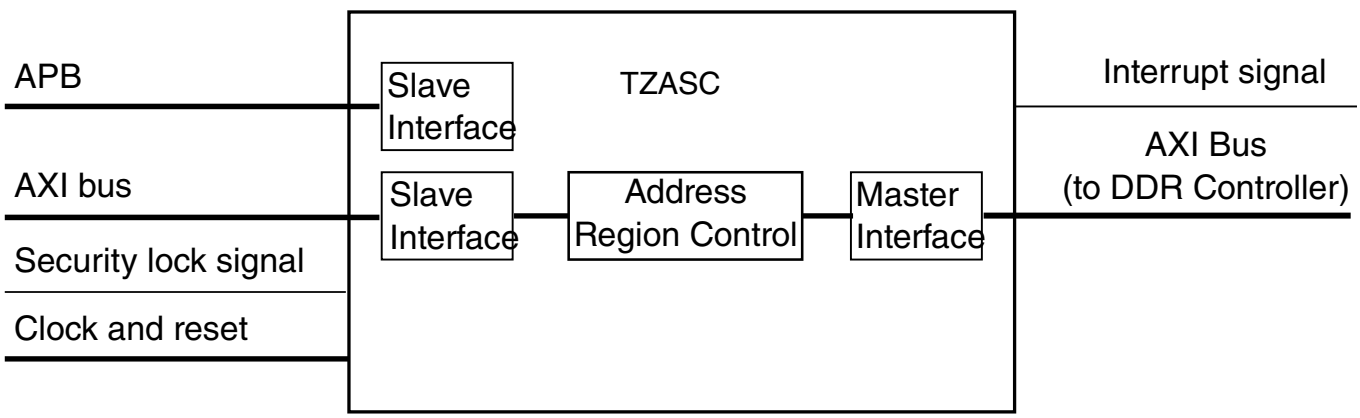


Figure 4-26. TZASC Block Diagram

The TZASC is an IP by Arm ("CoreLink™ TrustZone Address Space Controller TZC-380"), designed to provide configurable protection over program (SW) memory space.

The main features of TZASC are:

- Supports 16 independent address regions
- Access controls are independently programmable for each address region
- Sensitive registers may be locked
- Host interrupt may be programmed to signal attempted access control violations
- AXI master/slave interfaces for transactions
- APB slave interface for configuration and status reporting

NOTE

When illegal access is blocked by TZASC, the AXI ID read from the TZASC is not the AXI ID from the bus master. It cannot be used to identify the master which sends out the illegal access.

4.11.2 Clocks

The table found here describes the clock sources for TZASC.

See Clock Controller Module (CCM) for clock setting, configuration, and gating information.

Table 4-27. TZASC Clocks

Clock name	Clock Root	Description
aclk	ccm_clk_root	Module clock

4.11.3 Address Mapping in various memory mapping modes

The TZASC region base address starts at the beginning of DDR memory space (0x40000000) instead of the beginning of memory map (0x00000000). In this case the addresses configured in TZASC controller will be 1GB (0x40000000) offset and does not match the local addresses.

For example, setting region_setup_low_x=0xBE000000 maps DDR_ADDR=0xFE000000, the same behavior is observed with fail_address_x registers.

Memory "aliasing" implications on TZASC settings - in systems which does not utilize the maximal supported DDR space the controller is designed for, the whole DDR memory map becomes "aliased" (replicated) by the size of the physical memory used. In such cases, the TZASC must be configured to protect all aliased regions as well (i.e. effectively reducing the number of available TZASC regions, since all aliased regions must be handled, for each "real" space needing protection).

For complete details on TZASC functionality and the programming model, see the Arm document, “CoreLink™ TrustZone Address Space Controller TZC-380 Technical Reference Manual, (Rev r0p1 or newer)”, available at **http://infocenter.arm.com**.

4.12 System Debug

4.12.1 Debug

4.12.1.1 Debug Architecture

The chapter describes the debug architecture of the chip.

4.12.1.2 Debug System Features

The chip debug is based on Arm's CoreSight platform, with support for Quad-core A53 platform and Cortex-M4 core. The key features of the debug system include:

- Support 5-pins(JTAG) interface.
- Support both non-intrusive and halt-mode trace/debug options.
- MDM-AP registers for debugger to control mutli-core halt/resume cores.
- Trace Memory Controller (TMC) is used to enable capturing trace.
- 4KB in SOC trace block.
- ETR is used to allow routing trace data to system memory.
- Support ARM real time trace interface (TPIU) 16-bit @133MHz.
- Support cross trigger between Quad Cortex-A53 and Cortex-M4.
- 4 JTAG security levels, via SJC security functions together with e-Fuse (challenge response, field return, intrusive detection)

4.12.1.3 System level debug architecture

The debug architecture is shown in the following figure:

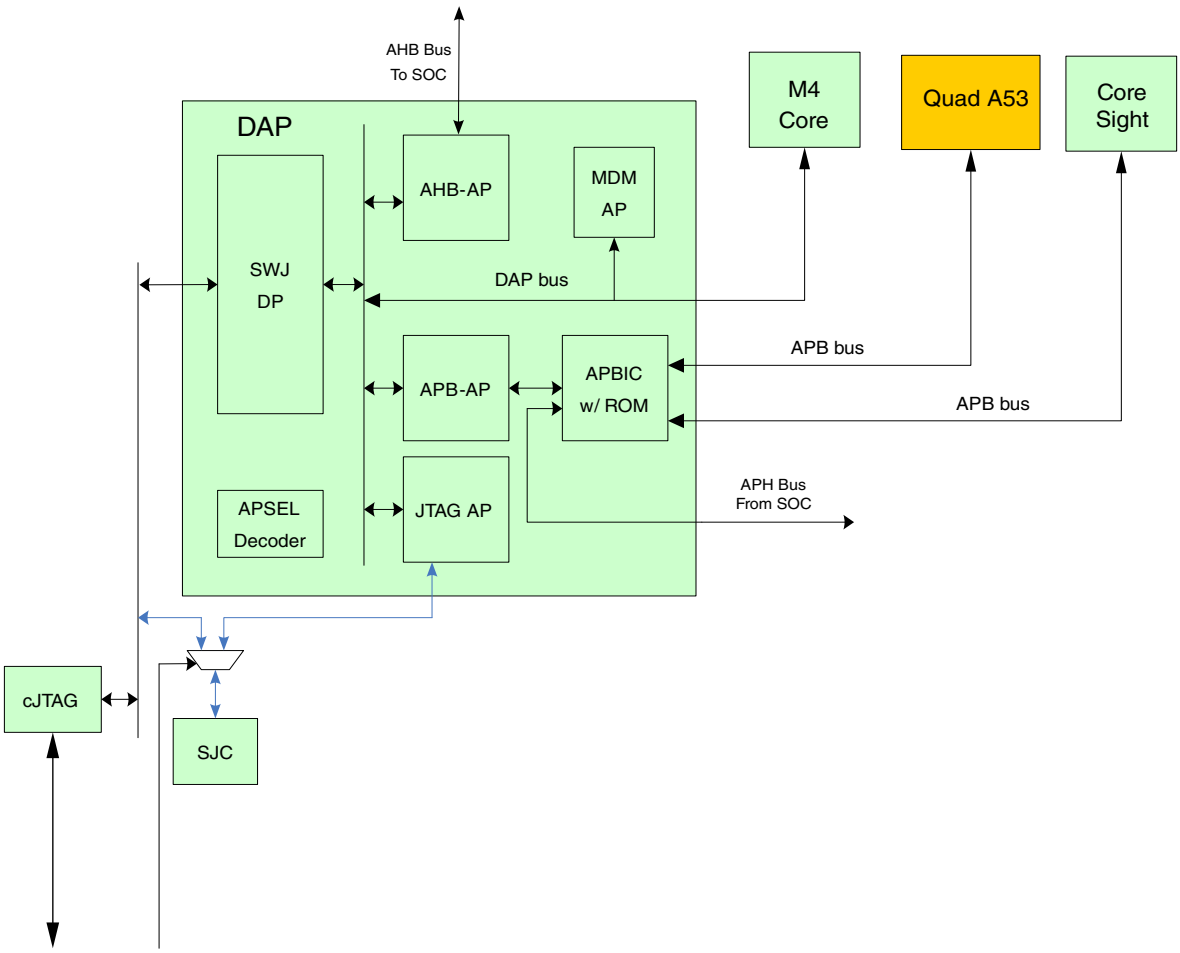


Figure 4-27. Debug Architecture Diagram

4.12.1.4 Functional description

This section gives a brief overview of the modules that are implemented within the Cortex-M/Cortex-A Core Platform. The debug blocks are part of the overall CoreSight platform debug system, which include the ETR, CTM, CTI, ATB replicator, APB address decode, TPIU and DAP. The CoreSight™ compatible Embedded Trace Macrocell (ETM) enables traces of program flow to be collected, compressed, and fed into the trace infrastructure. The Cross Trigger Interface (CTI) is included to provide a common programming model for use by the debug tools, control the trigger sources, and interface to the Cross Trigger Matrix (CTM). The debug is controlled via an ARM Debug Access Port (DAP).

4.12.1.4.1 Embedded Trace Router (ETR)

The Embedded Trace Router (ETR) is a CoreSight trace component which buffers trace data before dumping that data into DRAM through the AXI bus (via AXI master port). The output of the funnel is the ETR, and has the ability to buffer a certain amount before writing the data to DRAM.

4.12.1.4.2 Embedded Trace Macrocell (ETM)

The Embedded Trace Macrocell (ETM) is a CoreSight component. The ETM trace unit is a module that performs real-time instruction flow tracing based on the Embedded Trace Macrocell (ETM) architecture ETMv4.

4.12.1.4.3 Cross Trigger Matrix (CTM)

The CoreSight CTI channel signals from all the cores are combined using a Cross Trigger Matrix (CTM) block so that a single cross trigger channel interface is presented in the Cortex processor. This module can combine four internal channel interfaces corresponding to each core along with one external channel interface.

In the Cortex processor CTM, the external channel output is driven by the OR output of all internal channel outputs. Each internal channel input is driven by the OR output of internal channel outputs of all other CTIs in addition to the external channel input.

4.12.1.4.4 Cross Trigger Interface (CTI)

The Cortex processor has a single external cross trigger channel interface. This external interface is connected to the CoreSight Cross Trigger Interface (CTI) interface corresponding to each core through a Cross Trigger Matrix (CTM). A number of Embedded Cross Trigger (ECT) trigger inputs and trigger outputs are connected between debug components in the Cortex-A processor and CoreSight CTI blocks.

The CTI enables the debug logic, ETM trace unit, and performance monitoring, to interact with each other and with other CoreSight components. This is called cross triggering. For example, you configure the CTI to generate an interrupt when the ETM trace unit trigger event occurs.

4.12.1.4.5 AMBA Trace Bus Interface (ATB)

The AMBA Trace Bus (ATB) CoreSight debug components are part of the Trace stream block. The ATB components consist of the ATB Funnel and ATB Replicator. The ATB Funnel is used to merge several streams together to produce a single output to the ATB Replicator. The ATB Replicator enables two trace sinks to be wired together and receive ATB trace data from the same trace source.

4.12.1.4.6 Advanced Peripheral Bus Interface (APB)

The APB asynchronous interface / bridge connects several debug components. Each component has a single separate APB interface from the APB-IC (w/ ROM), which hooks up to a separate APB asynchronous component. The APB interface connects the core debug functions to the DAP.

4.12.1.4.7 Instrumentation Trace Macrocell (ITM)

The Instrumentation Trace Macrocell (ITM) is a application-driven trace source that supports printf style debugging to trace operating system and application events. The ITM generates diagnostic system information as packets. Multiple sources can generate packets. If multiple sources generate packets at the same time, the ITM arbitrates the order in which packets are output. These sources in decreasing order of priority are:

- **Software trace.** Software can write directly to ITM stimulus registers to generate packets.
- **Hardware trace.** The DWT generates these packets, and the ITM outputs them.
- **Time stamping.** Timestamps are generated relative to packets. The ITM contains a 21-bit counter to generate the timestamp. The Cortex-M processor clock or the bitclock rate of the Serial Wire Viewer (SWV) output clocks the counter.

4.12.1.4.8 Data Watchpoint and Trace (DWT)

The Data Watchpoint and Trace (DWT) debug unit provides watchpoints, data tracing, and system profiling for the processor.

4.12.1.4.9 Debug Access Port (DAP)

The Debug Access Port (DAP) is a physical port that connects to external debug tools. The DAP is part of the standardized ARM Debug Interface, and provides a bridge between a JTAG pin interface and on-chip memory mapped peripherals via the DAP bus. Transactions generated by the DAP are referred to as External Debugger Accesses.

4.12.1.5 JTAG topology

There is only one JTAG on the chip, and two JTAG modes are supported. Select via the JTAG_MOD pin.

- Debug mode: JTAG_MOD == 0, DAP is the only TAP controller in the daisy chain. SJC will be attached to JTAG-AP of DAP.
- Test mode: JTAG_MOD == 1, SJC is the only TAP controller in the daisy chain. 1149.1-compliant, and support 1149.6 AC coupled test.

When the JTAG interface is in Debug Mode, it can be operating in standard 5-pin JTAG interface. cJTAG/SWD interface is not supported by this chip.

4.13 System JTAG Controller (SJC)

4.13.1 Overview

The System JTAG Controller (SJC) provides debug and test control with the maximum security.

The test access port (TAP) is designed to support features compatible with the IEEE Standard 1149.1 v2001 (JTAG).

The figure below shows an overview of the JTAG architecture.

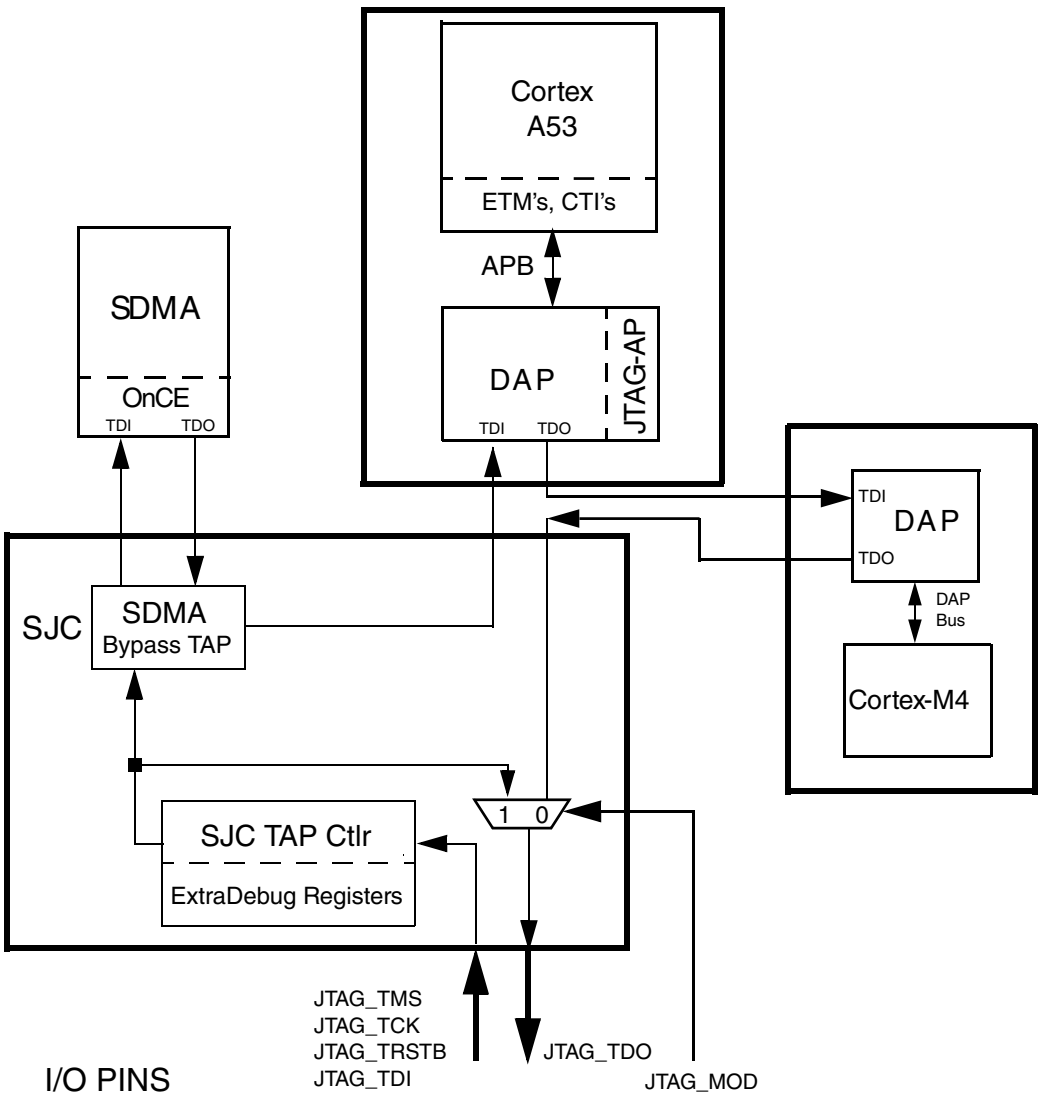


Figure 4-28. System JTAG Controller (SJC) Block Diagram

4.13.1.1 Features

The JTAG interface of chip, shared by SJC and DFT_TAP (for IEEE1149.1), provides the following capabilities:

- JTAG IEEE1149.1 mandatory instructions, see EXTEST Instruction, SAMPLE/ PRELOAD Instruction , and BYPASS Instruction .
- JTAG IEEE1149.1 optional instructions, see ID_CODE Instruction (SJC IDCODE / DFT_TAP IDCODE), HIGHZ Instruction, and CLAMP Instruction.

- JTAG IEEE P1149.1 (standard JTAG) interface to off-chip test and development equipment for true IEEE 1149.1 compliance, used primarily for board-level implementation of boundary scan.
- Debug-related control and status, such as putting selected cores into reset and/or debug mode and the ability to monitor individual core status signals via JTAG.
- Provides means for accessing each OnCE/ICE TAP controller independently to control a target system (see Modes of Operation).
- ExtraDebug logic (see ENABLE_ExtraDebug Instruction).
- The maximum clock speed of the SJC is one-eight of the lowest frequency of the accessed OnCE/ICE. For example in normal operation (no core in low-power mode), this frequency is one-eight of the SDMA frequency if this core is present in the TDI-TDO chain (serially connected with other cores or standalone). The user must also consider the 25 MHz frequency limitation on the CE bus.
- Core compliant modes to support standalone core debuggers (see Modes of Operation).
- Multi-cores daisy chained mode (default one) to support multi-core debuggers (see Modes of Operation).

Detailed information about the SJC is provided in the Security Reference Manual. Contact your NXP representative for information about obtaining this document.

4.13.1.2 Modes of Operation

The SJC modes are controlled through both the TAP select register (SJC_TSR) and the MOD input port.

The MOD port (typically connected to pad of the same name) selects between two possible topologies of TAP connections while using SJC functions, as seen at SoC level:

- Negating it (this should be the default state) selects all the TAPs (SJC, SDMA, DAP and Arm/ETM) to be connected in the TDI-TDO chain, which is referred to as "daisy chain" mode, throughout this chapter.
- Asserting it only selects the SJC TAP to be connected in the TDI-TDO chain.

SJC features are enabled by configuring the SJC input pin: MOD. Refer to the following table for MOD settings details:

Table 4-28. SJC Modes

MOD	Name	Description
0	Daisy chain ALL	For common SW debug (High speed and production)
1	SJC only	SJC compliant mode

NOTE

IEEE1149.1 standard features are enabled by the following IO configurations: (BOOT_MODE0, BOOT_MODE1, GPIO1_IO00, GPIO1_IO01, GPIO1_IO02, GPIO1_IO03, TEST_MODE) = (0000011)

The following figure shows the SJC mode selection flow. The numbers shown in parenthesis below each block name indicates the TAP's IR length.

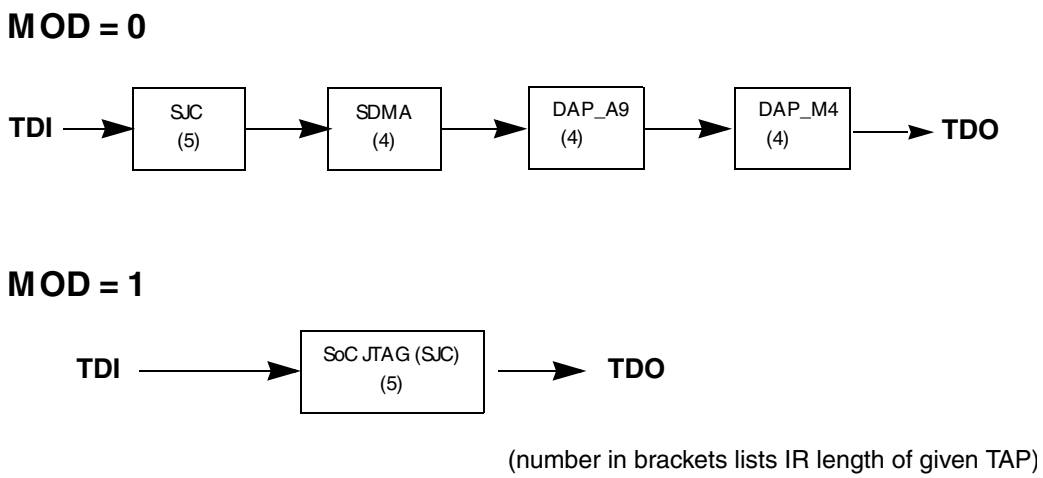


Figure 4-29. SJC Mode Selection Using MOD Pin Sampling

The Connect SDMA bit inside TAP select register controls the SDMA TAP bypass.

- When negated (should be the default state), the SDMA TAP is bypassed with a single D-FF (Flip-flop) during Shift-Dr path
- When asserted SDMA TAP is connected inside the chain
- When taking the SDMA into bypass or out of bypass (by writing to tapsel reg), additional cycle with TMS '0' should be given

The TAP selection block (TSB) provides a simple method of integrating various pieces of IP that have embedded TAPs.

- Provides a way to connect up multiple TAPs within a single SoC
- Follow the state of SJC TAP, and when the Test-Logic-Reset (TLR) state is reached, reset all TAPs

The figure below shows the TAP Selection Block and SOC TAP Chain Scheme.

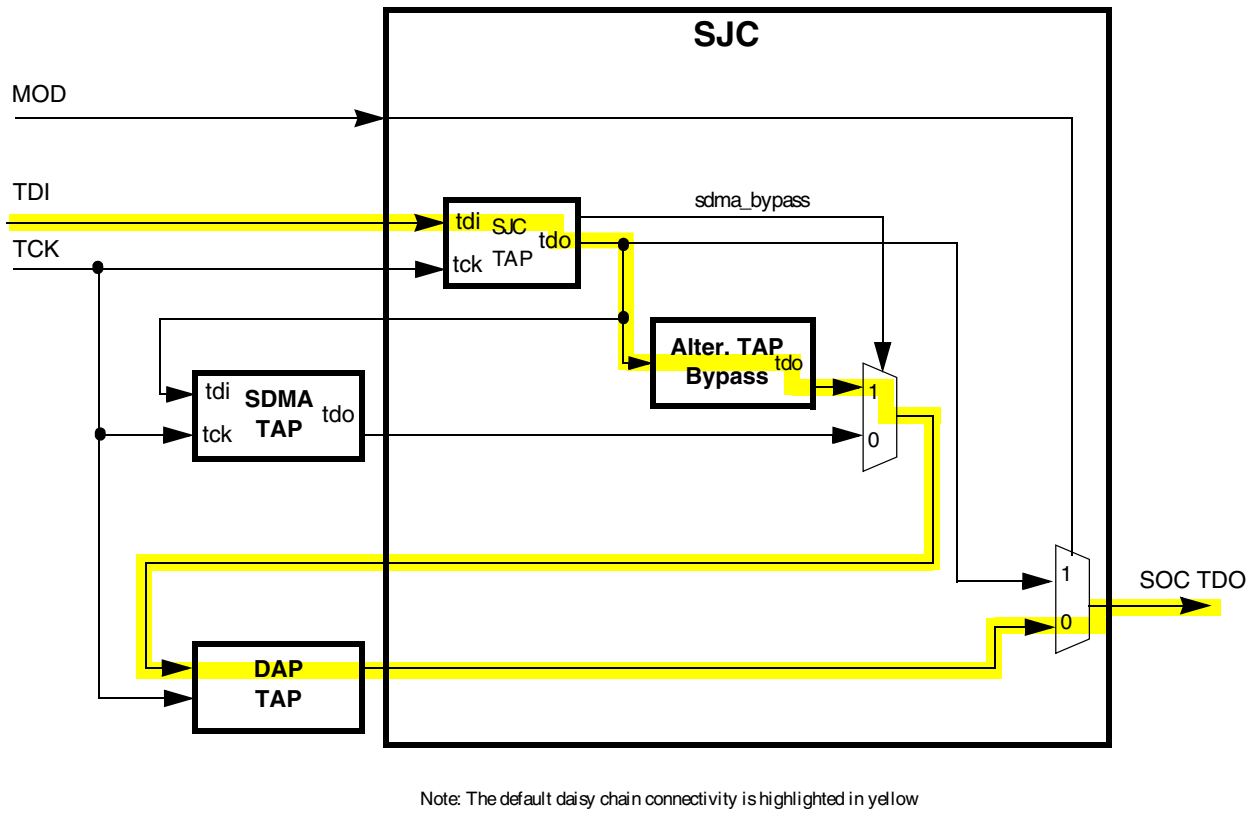


Figure 4-30. TAP Selection Block and SoC TAP Chain Scheme

NOTE

It is the responsibility of the user to ensure that in any configuration of the TAP controllers chosen, all of the TAPs in the chain comply with the demands of TCK clock frequency as well as the required ratio between TCK clock frequency and that of the core's to which the TAP refers.

4.13.2 TAP Selection Block (TSB)

As described in Modes of Operation, the SJC can access cores in different modes selected through a TSB.

4.13.2.1 Select Mode Using Software

Conceptually, the SJC_TSR is a data register which is accessed through Access TSR IR instruction of SJC TAP.

The following figure shows the process of using reserved IR to access the SJC_TSR.

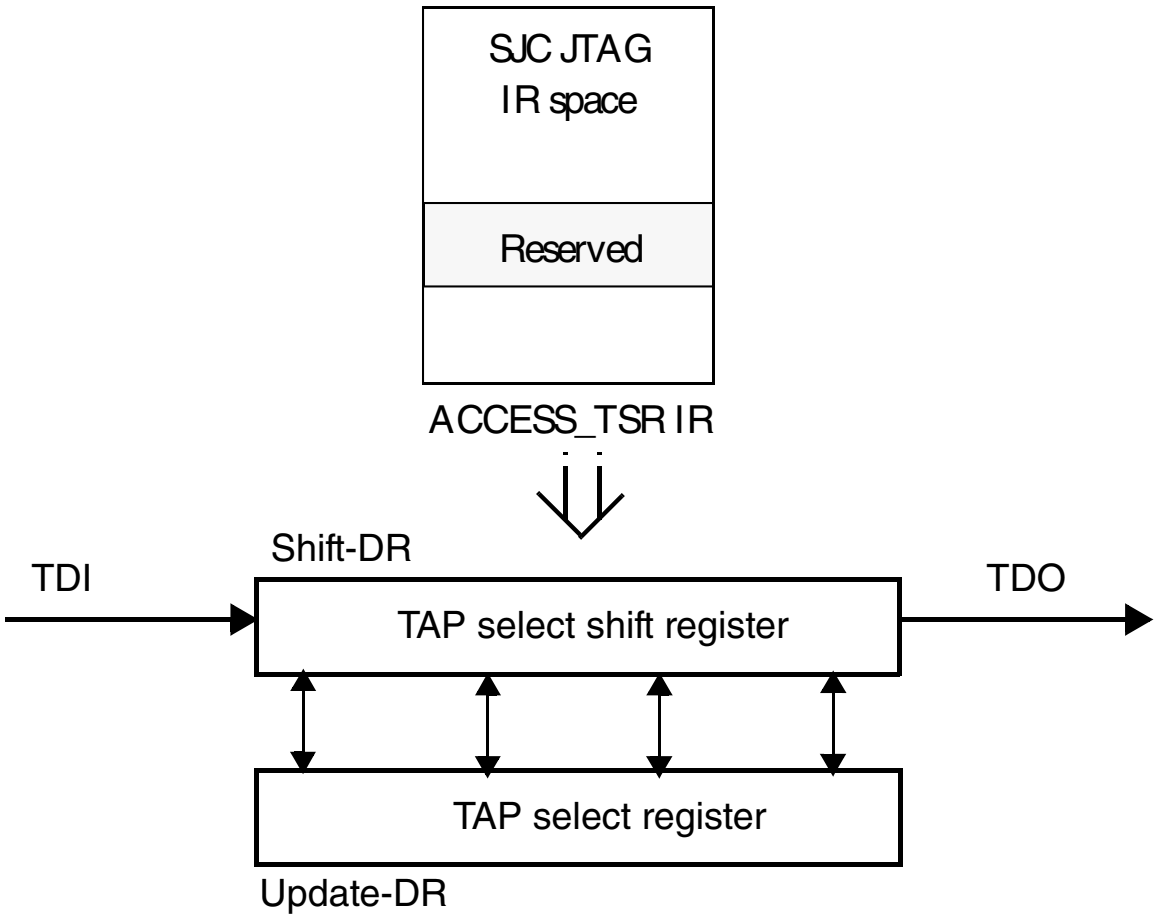


Figure 4-31. Using Reserved IR to Access the TAP Select Register (SJC_TSR)

The SJC_TSR can only be changed during the update-DR state of the TSB JTAG state machine. This is necessary to prevent a TAP that is being selected from losing synchronization with the TSB state machine when the TSB state machine returns to run-test-idle. Therefore, an associated shift register for the SJC_TSR is loaded into the SJC_TSR during the update-DR state (see the figure above). The shift register must also capture the state of the SJC_TSR when in the Capture-DR state for visibility of the contents of the SJC_TSR. See TAP Select Instruction , for more information.

4.13.3 Boundary Scan Register (BSR)

The Boundary Scan Register (BSR) in the JTAG implementation contains bits for all device signal and clock pins and associated control signals.

All SoC bidirectional pins have a single register bit in the boundary scan register for pin data, and are controlled by an associated control bit in the boundary scan register.

4.13.4 SJC Instruction Register (SJIR)

The SJC Instruction register is provided in the following table. The SJC Instruction register is 5 bits wide.

Table 4-29. SJC Instruction Register (SJIR)

Code					SJC IR
B4	B3	B2	B1	B0	
0	0	0	0	0	SJC IDCODE
0	0	0	0	1	Reserved
0	0	0	1	0	
0	0	0	1	1	Reserved
0	0	1	0	0	ENABLE_ExtraDebug
0	0	1	0	1	ENTER_DEBUG (secured)
0	0	1	1	0	Reserved
0	0	1	1	1	TAP select
0	1	0	0	0	
0	1	0	0	1	
0	1	0	1	0	Reserved
0	1	0	1	1	Reserved
0	1	1	0	0	Security Output challenge
0	1	1	0	1	Security Enter response
-					Reserved
1	1	1	1	1	Reserved

The instruction register is reset to 0b00000 in the test-logic-reset controller state which is equivalent to the SJC IDCODE instruction.

During the capture-IR controller state, the parallel inputs to the instruction register are loaded with the code 01 in the least significant bits as required by the standard; the most significant bits are loaded with the values 00, leading to a capture value of 0b00001.

4.13.4.1 DFT_TAP JTAG Instruction Register (DJIR)

The DFT_TAP JTAG Instruction register is provided in the following table. The DFT_TAP JTAG Instruction register is 4 bits wide.

Table 4-30. DFT_TAP JTAG Instruction Register (DJIR)

Code				DFT_TAP IR
B3	B2	B1	B0	
1	1	1	1	BYPASS
0	0	1	0	EXTEST
0	0	1	1	SAMPLE/PRELOAD
0	1	1	0	CLAMP
0	1	0	0	DFT_TAP IDCODE
0	1	1	1	HIGHZ

The DFT_TAP JTAG instruction register is reset to 0b0000, which is reserved.

4.13.4.2 ID_CODE Instruction (SJC IDCODE / DFT_TAP IDCODE)

Selects the ID register, and the system logic controls the I/O pins. This instruction is provided as a public instruction to allow the manufacturer, part number and version of a component to be determined through the TAP.

The table below shows the ID register configuration.

Table 4-31. ID Configuration Register (SJC IDCODE / DFT_TAP IDCODE)

IDCODE				ID Configuration Register												
	BIT 31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	BIT 16
	Version Information[3:0]			Part Number (Bits 27-16)												
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	0	0	1	0	0	0	1	0	0	0	1	1	1	0
Note:																
	BIT 15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	BIT 0
	Part Number (Bits 15-12)			Manufacturer Identity												1
TYPE	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r	r
RESET	0	0	1	0	0	0	0	0	0	0	0	1	1	1	0	1
Note:																

Table 4-32. SJC ID Configuration Register Description (SJC IDCODE)

Field	Description
31-28 Version Information	IC/SoC Version information number. Initial value: '0000' This number is subject to changes, for new IC/SoC (System On A Chip) revision releases.
27-12 Part Number	Customer Part Number The 16-bit Part Number value is unique for every NXP SoC / IC. See System Debug chapter for exact register value for a specific SoC.
11-1 Manufacturer Identity	Manufacturer Identity NXP Manufacturer Identity code. Bits [11:1] - 00000001110
0	Tied to logic 1.

Table 4-33. DFT_TAP ID Configuration Register Description (DFT_TAP IDCODE)

Field	Description
31-28 Version Information	IC/SoC Version information number. Initial value: '0001' This number is subject to changes, for new IC/SoC (System On A Chip) revision releases.
27-12 Part Number	Customer Part Number The 16-bit Part Number value is unique for every NXP SoC / IC. Value: 1100111110000000 See System Debug chapter for exact register value for a specific SoC.
11-1 Manufacturer Identity	Manufacturer Identity NXP Manufacturer Identity code. Value: 01010101001
0	Tied to logic 1.

One application of the ID register is to distinguish the manufacturer(s) of components on a board when multiple sourcing is used. As more components emerge which conform to the IEEE 1149.1 standard, it is desirable to allow for a system diagnostic controller unit to blindly interrogate a board design to determine the type of each component in each location. This information is also available for factory process monitoring and for failure mode analysis of assembled boards.

Once the IDCODE instruction is decoded, it selects the ID register which is a 32 Bit data register. Because the bypass register loads a logic 0 at the start of a scan cycle, whereas the ID register loads a logic 1 into its least significant bit, examination of the first bit of data shifted out of a component during a test data scan sequence immediate following exit from Test-Logic-Reset controller state shows whether such a register is included in

the design. When the IDCODE instruction is selected, the operation of the test logic has no effect on the operation of the on-chip system logic as required by the IEEE 1149.1 standard.

4.13.4.3 SAMPLE/PRELOAD Instruction

Selects the boundary scan register and the system logic controls the I/O pins.

The SAMPLE/PRELOAD instruction provides two separate functions:

- First, it provides a means to obtain a snapshot of system data and control signals. The snapshot occurs on the rising edge of TCK in the capture-DR controller state. The data can be observed by shifting it transparently through the boundary scan register.
- The second function of SAMPLE/PRELOAD is to initialize the boundary scan register output cells prior to selection of EXTEST. This initialization ensures that known data appears on the outputs when entering the EXTEST instruction.

NOTE

Because there is no internal synchronization between the JTAG clock (TCK) and the system clock (CLK), the user must provide some form of external synchronization to achieve meaningful results.

For more details on the function and use of SAMPLE/PRELOAD, refer to the appropriate IEEE 1149.1 document.

4.13.4.4 EXTEST Instruction

Selects the boundary scan register, and the 1149.1 test logic has control of the I/O pins.

By using the TAP controller, the register is capable of:

- Scanning user-defined values into the output buffers,
- Capturing values presented to input pins
- Controlling the direction of bidirectional pins,
- Controlling the output drive of tri-statable output pins.

For more details on the function and use of EXTEST, refer to the appropriate IEEE 1149.1 document.

The EXTEST instruction also asserts internal reset for the cores (through CCM, refer to Figure 4-34) to force a predictable internal state while performing external boundary scan operations.

4.13.4.5 HIGHZ Instruction

All output drivers, including the two-state drivers, are turned off (that is, high impedance). The instruction selects the bypass register.

In this mode, all internal pullup resistors on all the pins (except for the TMS, TDI, TCK, TRSTB pins) are disabled. This disabling functionality is not built into SJC, but should be implemented by some logic in the SOC/IO Pads.

For more details on the function and use of HIGHZ, refer to the IEEE 1149.1 document.

The HIGHZ instruction also asserts internal reset for the cores (through CCM, refer to Figure 4-34) to force a predictable internal state while performing external boundary scan operations.

4.13.4.6 BYPASS Instruction

Selects the single Bit bypass register and the system logic controls the I/O pins.

This creates a shift-register path from TDI to the bypass register and, finally, to TDO, circumventing the boundary scan register. This instruction is used to enhance test efficiency when a component other than the SoC Core based device becomes the device under test.

When the bypass register is selected by the current instruction, the shift-register stage is set to a logic zero on the rising edge of TCK in the capture-DR controller state. Therefore, the first bit to be shifted out after selecting the bypass register is always a logic zero.

For more details on the function and use of BYPASS, refer to the appropriate IEEE 1149.1 document.

4.13.4.7 CLAMP Instruction

The CLAMP instruction is established initially with the SAMPLE and PRELOAD instructions.

It drives preset values onto the outputs of devices and then, unlike the SAMPLE and PRELOAD instructions, it selects the bypass register between TDI and TDO. CLAMP can be used to set values on the outputs of certain devices to avoid bus contention problems.

4.13.4.8 ENABLE_ExtraDebug Instruction

The TDI and TDO pins are connected directly to the ExtraDebug registers, the SJC TAP controller remaining connected to TDI and TMS.

The ExtraDebug shift register consists of 38 bits (maximum) comprising a 32-bits data field (maximum length, see Accessing ExtraDebug Registers,), a 5 bits address field and read/write bit. On a register read, the data field does not need to be filled in. The particular ExtraDebug register connected between TDI and TDO at a given time is selected by the ExtraDebug controller depending on the ExtraDebug Address being currently decoded. All communication with the ExtraDebug controller is done through the Select-DR-Scan path of the JTAG TAP Controller.

4.13.4.9 ENTER_DEBUG instruction

The ENTER_DEBUG instruction is used to generate a debug request event to SDMA and the Arm Core Platform simultaneously (practically, inherited minimal skew is expected, due to difference in event signal propagation in the different modules).

The TDI and TDO are connected to the Instruction Register (IR). After the acknowledgment of the Debug Mode is received (can be checked by reading the Core Status Register part of the ExtraDebug logic), the user can perform system debug functions on the cores.

NOTE

The ENTER_DEBUG event issue to the cores, can be masked, by bits in DCR register.

NOTE

It is user's responsibility to shift-in another IR value (like IDCODE) before trying to bring the cores out of debug mode, as the debug request signals to the cores remains asserted as long as ENTER_DEBUG IR is in place.

NOTE

The user need to check that cores are in debug mode (watching debug acknowledge signal) before leaving ENTER_DEBUG instruction, otherwise debug request might not take affect.

4.13.4.10 TAP Select Instruction

By means of TAP select instruction a user can access TAP select register and by controlling its only bit SDMA Bypass, control whether SDMA TAP is bypassed or not.

Table 4-34. TAP Select Register (TSR)

	TAP Select Register
	BIT 0
	Connect SDMA
TYPE	rw
RESET	0
Note:	

Table 4-35. TAP Select Register Description

Field	Description
0 SDMA Bypass	Connect SDMA Control whether SDMA TAP is bypassed or not: • 0 - SDMA TAP is bypassed by the alternate TAP inside SJC (emulating 4-bit IR and 1-bit bypass path). • 1 - SDMA TAP is connected to the TDI-TDO chain. **NOTE:** Additional cycle with TMS '0' should be inserted, after writing to this register, to allow the SDMA tap be sync before SDMA get into / out of bypass.

4.13.5 Security

JTAG manipulation is one of the known hackers' ways of executing unauthorized program code, getting control over the OS and run code in privileged modes.

The SJC provides a debug access to several H/W blocks including the Arm processor and the system bus. This allows for program control and manipulation as well as visibility into system peripherals and memory. The ETM and NEXUS interfaces allow bus transactions to be traced. Together these tools provide the hacker all the access needed to completely comprise the system. Means must be provided to block any malicious JTAG access.

The SJC provides a way of regulating the JTAG access.

The following are the different JTAG security modes:

- **Mode #1: No Debug**-Maximum Security. All security sensitive JTAG features are permanently blocked.

- **Mode #2: Secure JTAG**-High security. JTAG use is regulated by secret key based authentication mechanism.
- **Mode #3: JTAG Enabled**-Low security. JTAG always enabled.

The JTAG security modes are configured using eFUSEs which can burned after packaging by applying electrical signals. The fuse burning is irreversible process, once a fuse is burned (e-fuse or laser fuse) it is impossible to change the fuse back to the un-burned state.

4.13.5.1 JTAG Security Modes

JTAG can be in one of JTAG security modes which is selected by setting the SJC eFUSE configuration. The physical location of the fuses is not in the SJC.

4.13.5.1.1 Mode 1: No Debug - Maximum Security

No Debug JTAG security mode provides the highest security level.

In this mode, all JTAG features are disabled except for:

- ScanBoundary Scan
- PLL bypass- Bypass Arm or/and USB PLL.
- Visibility of the following status bits: power mode - normal, standby, stop, shutdown, and so on

These features do not reduce the security level of the product, and they allows to perform important tests and board connectivity checks.

4.13.5.1.2 Mode 2: Secure JTAG - High Security

The Secure JTAG mode limits the JTAG access by using challenge/response based authentication mechanism. Any access to JTAG port is being checked. Only authorized debug devices (that is, devices having the right response) can access the JTAG, unauthorized JTAG access attempts are denied.

The intent of this mode is to allow return field testing. When a secured JTAG device is being returned for debugging, this mode allows authorized re-activation of the JTAG.

4.13.5.1.2.1 Challenge/Response Mechanism in System JTAG Mode

When SJC is in Sysytem JTAG mode the authentication process is as follows:

1. Shift Output Challenge instruction to IR.

2. Passing through Capture-DR state of the SJC and by performing Shift-DR operations Challenge code can be accessed from TDO.
3. Shift Enter Response instruction to IR. By performing Shift-DR, operations enter Response code value through TDI. As Update-DR state is entered, Response code is compared with the correct one.

In Fixed challenge-response pair mode, each part has its individual challenge - response pair which is determined at manufacturing time, and does not change later on. The SJC compares the user's response to the expected response.

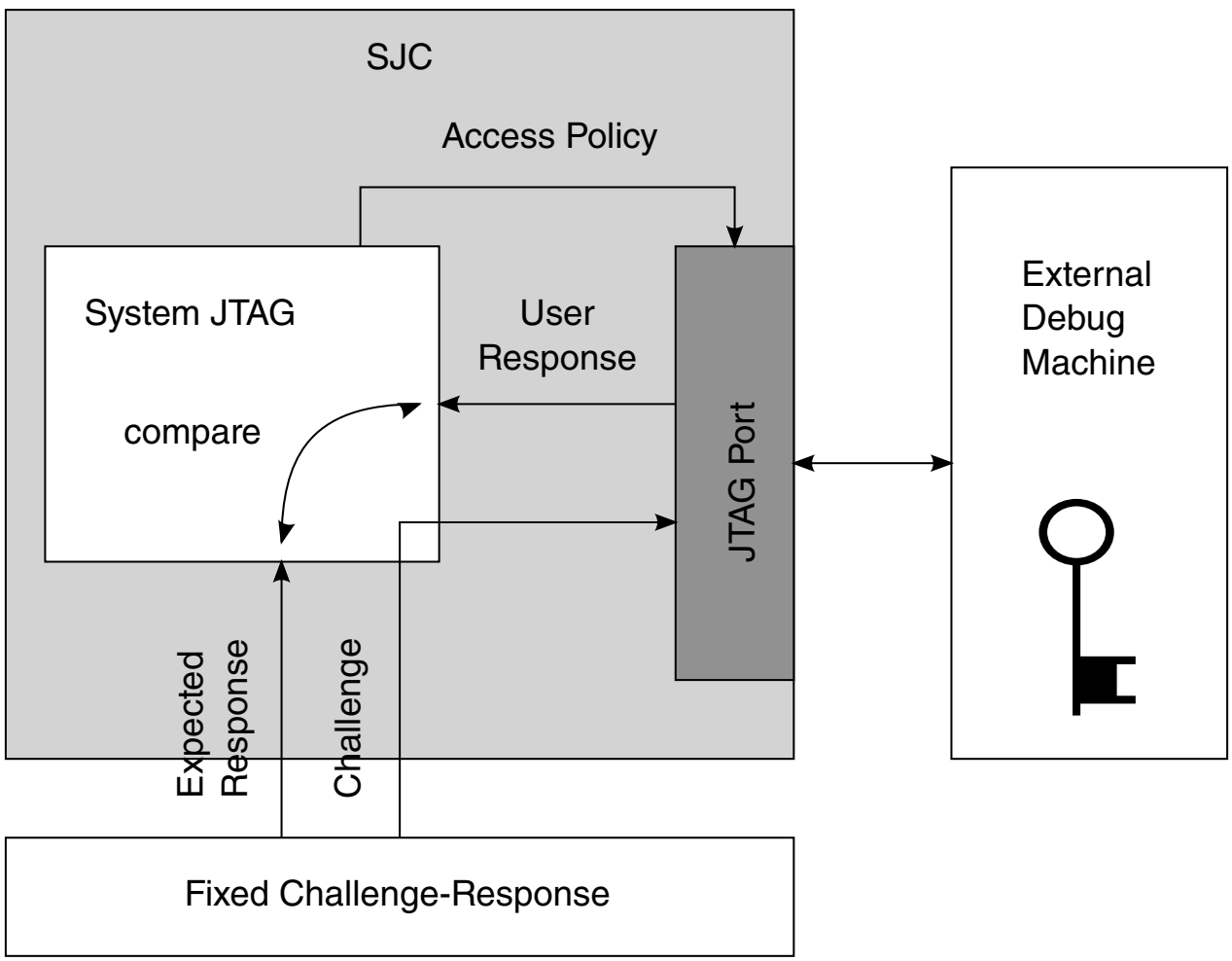


Figure 4-32. Mode #2 - Secure JTAG with Fixed Challenge-response Pair

4.13.5.1.3 Mode 3: JTAG Enabled - Low Security

In the JTAG Enabled JTAG security mode, all JTAG features are enabled.

4.13.5.2 Software Enabled JTAG

To increase the flexibility of the SJC, an option to enable the JTAG via software is added and is available only in Secure JTAG mode. By writing '1' to HAB_JDE (HAB JTAG DEBUG ENABLE) bit in the e-fuse controller module, the JTAG is opened, regardless of its security mode. It is the responsibility of software to assert or negate this bit.

Additionally, a corresponding lock bit is available (in the e-fuse control module) to ensure that only trusted software is able to set the JDE bit. When the LOCK bit is set, no future change of JDE is possible, until the next POR (power-on-reset) cycle.

The platform initialization software should set the LOCK bit for JDE bit before transferring control to the application code.

The S/W JTAG enable allows JTAG enabling without activating the challenge-Response mechanism (which requires JTAG access tool enhancement or special H/W). The JTAG S/W enable does not allow debug in case of boot or memory fault as it requires reset before entering debug.

This feature can be permanently blocked by burning the dedicated e-fuse.

NOTE

The S/W enabled JTAG feature reduces the overall security level of the system as it relies on S/W protections. If this feature is not required, it is strongly recommended to burn the JTAG_HEO e-fuse which disables this feature.

4.13.5.3 Kill Trace

The kill trace signal disables any output of the ETM block. The ETM can be accessed either via JTAG port and/or by direct software code. Blocking the JTAG port also yields assertion of the kill trace signal. This resulted in blocking of trace port. The intention of this action is to block any attempt to break into the system via software manipulation of the debug modules. The kill trace, when active, prevents trace output even in case where it can be activated via chip pin.

The kill trace feature needs to be activated by burning a dedicated e-fuse. If the fuse is left intact, kill trace is never activated as seen in Figure 4-33.

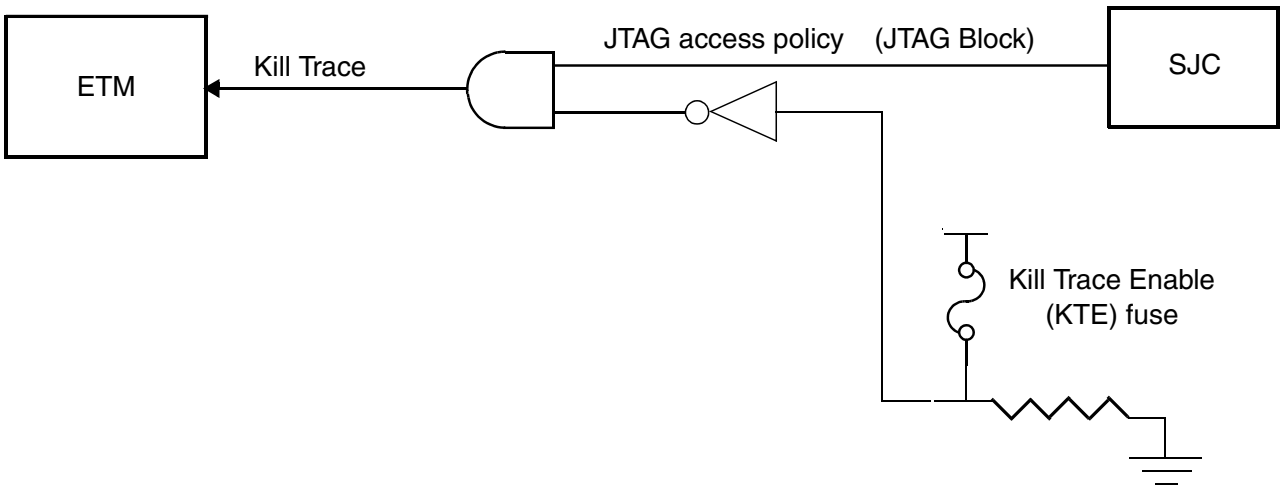


Figure 4-33. Kill Trace eFUSE

The kill trace is asserted when "kill trace enable" fuse is burned and "ipt_secur_block" signal in SJC is asserted, which happens when at least one of the following is true:

- Mode #2 (Secure JTAG) and no code has been entered
- Mode #2 (Secure JTAG) with burned Bypass and Re-enable fuses
- Mode #2 (Secure JTAG) with incorrect response entered
- Mode #1 (No debug)
- TRST_B signal is active
- POR has not ever been asserted

4.13.5.4 SJC Disable Fuse

In addition to the different JTAG security modes that are implemented internally in the System JTAG Controller (SJC), there is an option to disable the SJC functionality by eFUSE configuration. This creates additional JTAG mode that is, JTAG Disabled with highest level of JTAG protection. In this mode all JTAG features are disabled.

Specifically, the following debug features are disabled in addition to the features that were already disabled in No Debug JTAG mode:

- Boundary scan register (SJC_BSR)
- Non-Secure JTAG control registers (PLL configuration, Deterministic Reset, PLL bypass)
- Non-Secure JTAG status registers (Core status)
- Chip Identification Code (IDCODE)

4.13.6 Functional Description

This section provides a complete functional description of the block.

4.13.6.1 Static Core Debug

The SJC JTAG TAP controller is fully compatible with the IEEE 1149.1a-2001 Standard Test Access Port and Boundary Scan Architecture specifications.

The Arm platform has an integrated JTAG interface and a TAP controller to manage its own ICE. Also it can access an embedded trace ETM interface, see Arm core and ETM Technical reference guide for more information.

The SDMA has a TAP controller to manage its own OnCE, see SDMA OnCE specifications for more details.

The OnCE and ICE provide a mean of interacting with the cores and their peripherals non-intrusively so that a user may examine registers, memories to facilitate hardware and software development. Refer to TAP Selection Block (TSB), for more information.

4.13.6.2 Reset Mechanism

The following figure shows the SJC reset logic

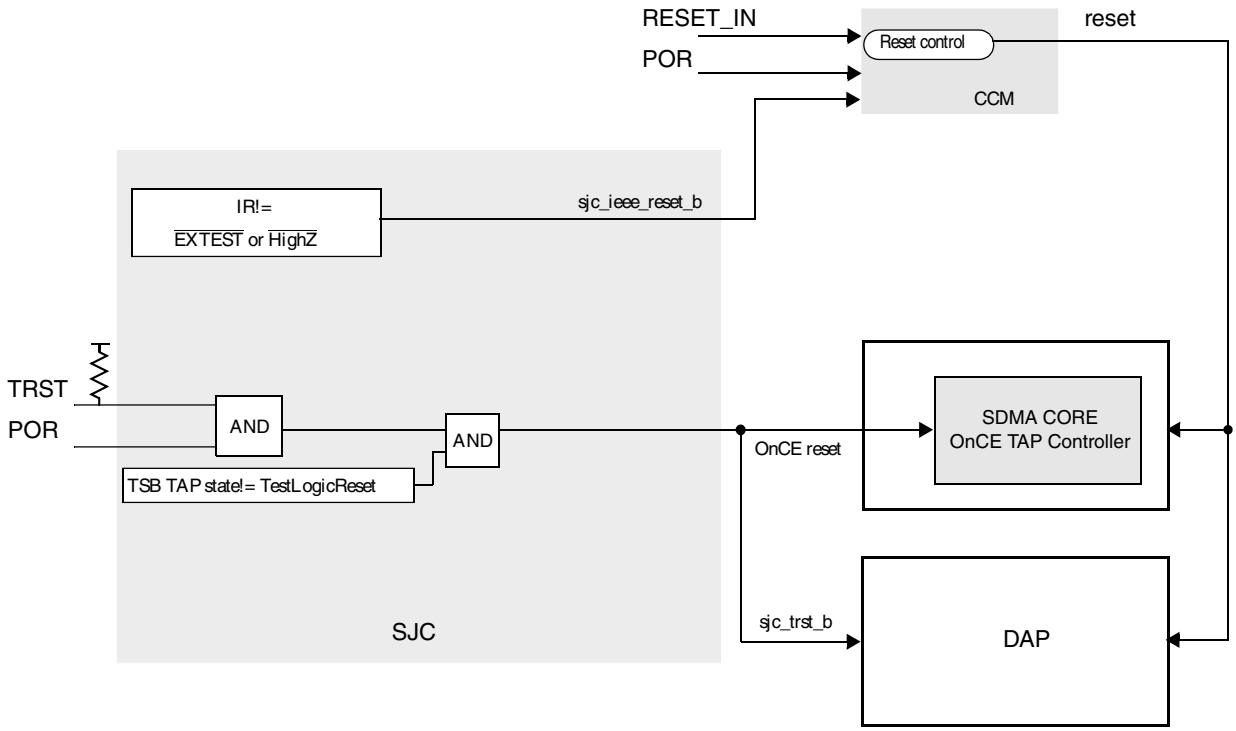


Figure 4-34. SJC Reset Logic

NOTE

- Asserting TRSTB in any scan mode resets the TCR loosing the testmode configuration and selects default TAP.
- SJC generates an IEEE reset signal to the CCM when in one of the IEEE modes HIGHZ or EXTEST. This signal generates a system reset to the cores until exit from one of these modes.
- The TSB generates Once/ICE reset (either TRSTB if implemented or other) when its TAP state reaches Test-Logic-Reset (meaning that TAP accessed is also reaching Test-Logic-Reset).

4.13.7 Initialization/Application Information

The control afforded by the output enable signals using the boundary scan register and the EXTEST instruction requires a compatible circuit-board test environment to avoid device-destructive configurations. The user must avoid situations in which the SJC output drivers are enabled into actively driven networks.

There are two constraints related to the JTAG interface:

- Ensure that the JTAG test logic is kept transparent to the system logic by forcing TAP into the Test-Logic-Reset controller state. During power-up, SJC's internal TRSTB is asserted as IC's POR_B is asserted which forces the TAP controller into this state. After that, if TMS either remains unconnected or is connected to VCC, then the TAP controller cannot leave the Test-Logic-Reset state, regardless of the state of TCK.
- DE_B is an IO pin with pullup and care must be taken of the direction when driving this signal.

4.13.8 SJC Memory Map/Register Definition

In addition to the standard accessible JTAG registers (per IEEE1149.1 standard) listed in SJC Instruction Register (SJIR) , the chip contains the following registers accessed using the ExtraDebug mechanism, controlled via "ENABLE_ExtraDebug" IR instruction.

NOTE

SJC registers are only accessible by JTAG interface. They are not memory mapped to processor address space, so the absolute addresses provided by default in the SJC memory map are not valid.

This section assumes the JTAG controller is accessed in standalone mode or daisy chained (defined by TAP Selection Block) using the appropriate TSB configuration.

See "System Debug" chapter for more details about the general purpose register descriptions that are unique to this chip.

SJC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	General Purpose Unsecured Status Register 1 (SJC_GPUSR1)	32	R	0000_0000h	4.13.8.1/ 285
1	General Purpose Unsecured Status Register 2 (SJC_GPUSR2)	32	R	0000_0000h	4.13.8.2/ 287
2	General Purpose Unsecured Status Register 3 (SJC_GPUSR3)	32	R	0000_0000h	4.13.8.3/ 287
3	General Purpose Secured Status Register (SJC_GPSSR)	32	R	0000_0000h	4.13.8.4/ 288
4	Debug Control Register (SJC_DCR)	32	R/W	0000_0000h	4.13.8.5/ 289

Table continues on the next page...

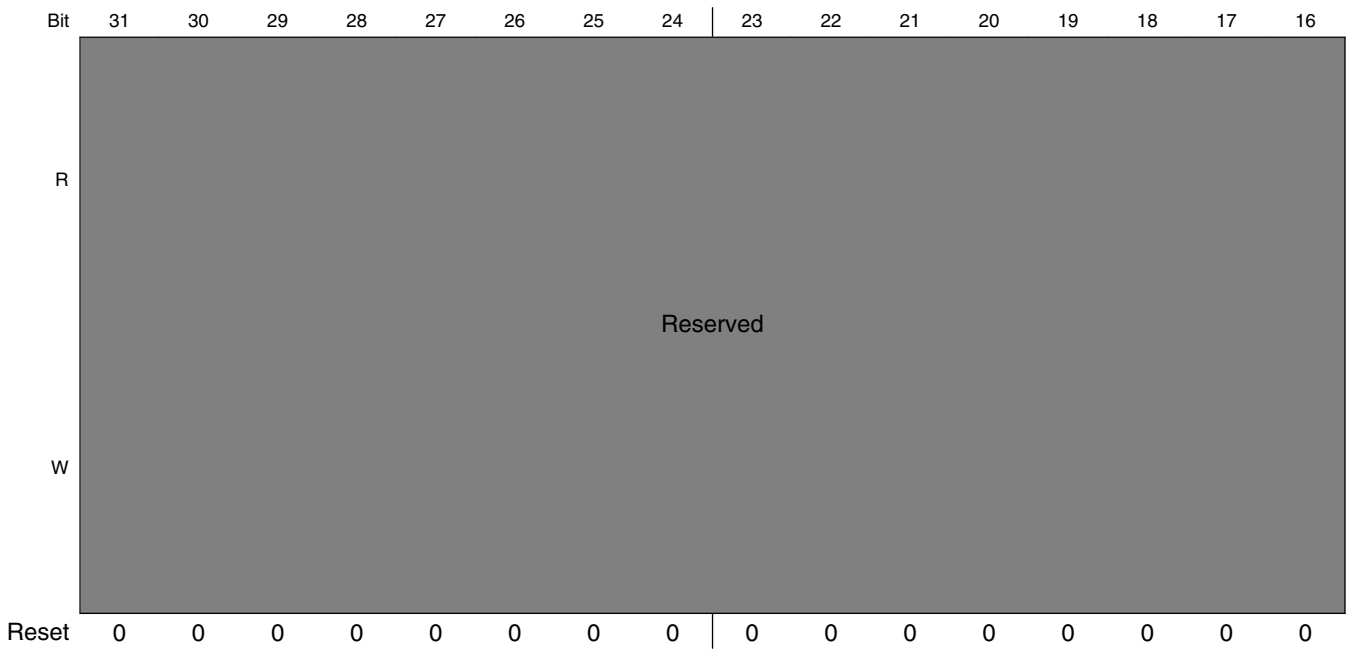
SJC memory map (continued)

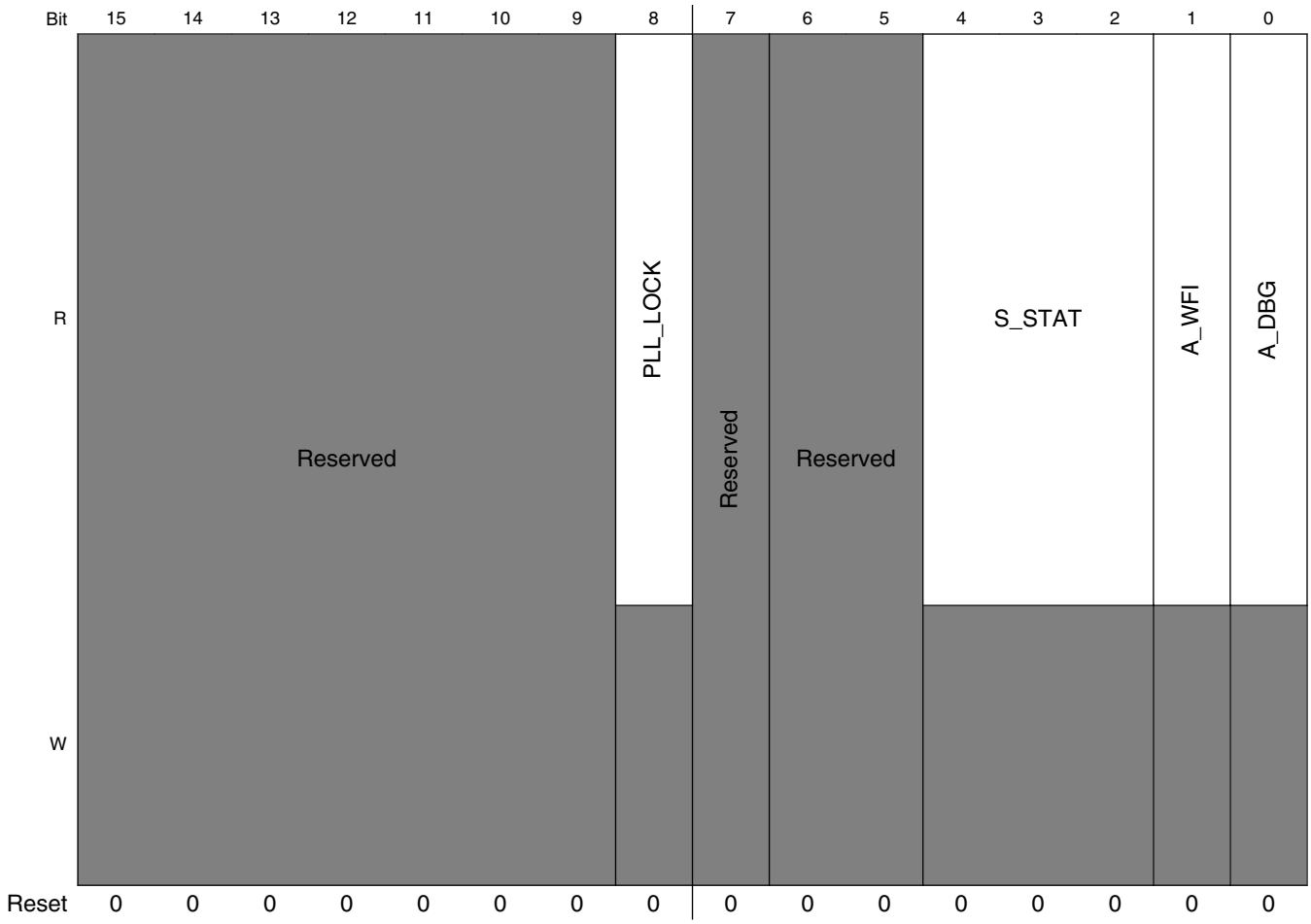
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
5	Security Status Register (SJC_SSR)	32	R	See section	4.13.8.6/ 291
7	General Purpose Clocks Control Register (SJC_GPCCR)	32	R/W	0000_0000h	4.13.8.7/ 294

4.13.8.1 General Purpose Unsecured Status Register 1 (SJC_GPUSR1)

The General Purpose Unsecured Status Register 1 is a read only register used to check the status of the different Cores and of the PLL. The rest of its bits are for general purpose use.

Address: 0h base + 0h offset = 0h



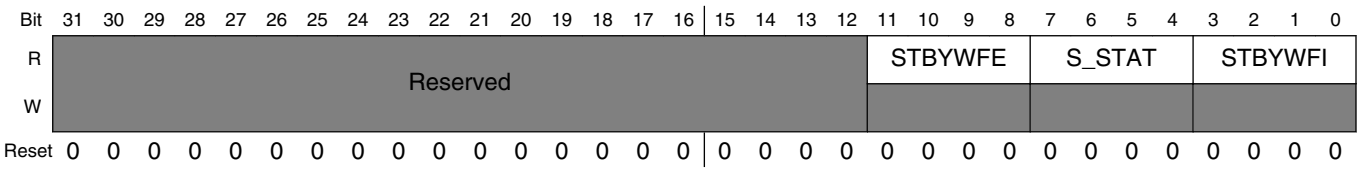


SJC_GPUSR1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved.
8 PLL_LOCK	PLL_LOCK A Combined PLL-Lock flag indicator, for all the PLL's.
7 -	This field is reserved. Reserved
6–5 -	This field is reserved. Reserved.
4–2 S_STAT	3 LSBits of SDMA core statusH.
1 A_WFI	Arm core wait-for interrupt bit Bit 1 is the Arm core standbywfi (stand by wait-for interrupt). When this bit is HIGH, Arm core is in wait for interrupt mode.
0 A_DBG	Arm core debug status bit Bit 0 is the Arm core DBGACK (debug acknowledge) DBGACK can be overwritten in the Arm core DCR to force a particular DBGACK value. Consequently interpretation of the DBGACK value is highly dependent on the debug sequence. When this bit is HIGH, Arm core is in debug.

4.13.8.2 General Purpose Unsecured Status Register 2 (SJC_GPUSR2)

Address: 0h base + 1h offset = 1h

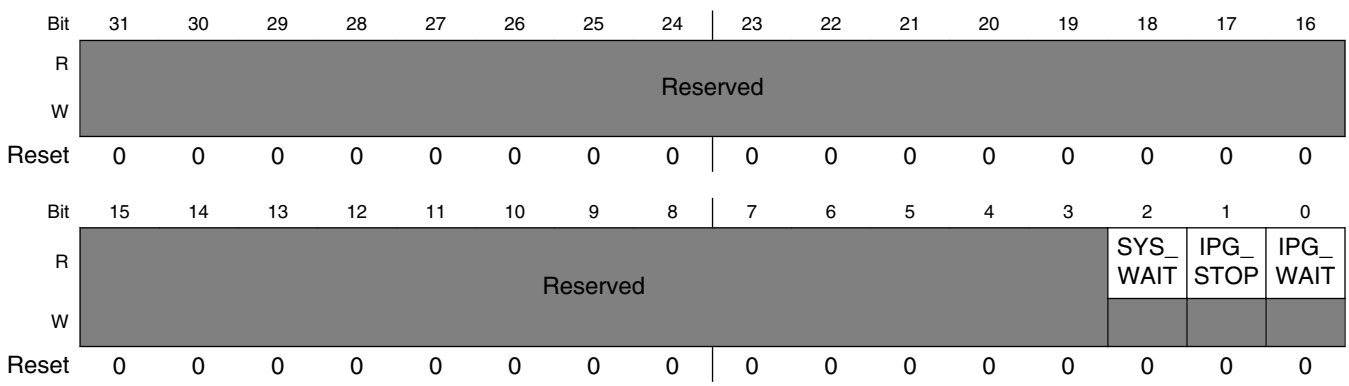


SJC_GPUSR2 field descriptions

Field	Description
31–12 -	This field is reserved. Reserved
11–8 STBYWFE	STBYWFE[3:0] Reflecting the "Standby Wait For Event" signals of all cores.
7–4 S_STAT	S_STAT[3:0] SDMA debug status bits: debug_core_state[3:0]
STBYWFI	STBYWFI[3:0] These bits provide status of "Standby Wait-For-Interrupt" state of all Arm cores.

4.13.8.3 General Purpose Unsecured Status Register 3 (SJC_GPUSR3)

Address: 0h base + 2h offset = 2h



SJC_GPUSR3 field descriptions

Field	Description
31–3 -	This field is reserved. Reserved

Table continues on the next page...

SJC_GPUSR3 field descriptions (continued)

Field	Description
2 SYS_WAIT	System In wait Indication on System in wait mode (from CCM).
1 IPG_STOP	IPG_STOP CCM's "ipg_stop" signal indication
0 IPG_WAIT	IPG_WAIT CCM's "ipg_wait" signal indication

4.13.8.4 General Purpose Secured Status Register (SJC_GPSSR)

The General Purpose Secured Status Register is a read-only register used to check the status of the different critical information in the SoC. This register cannot be accessed in secure modes.

Address: 0h base + 3h offset = 3h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GPSSR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

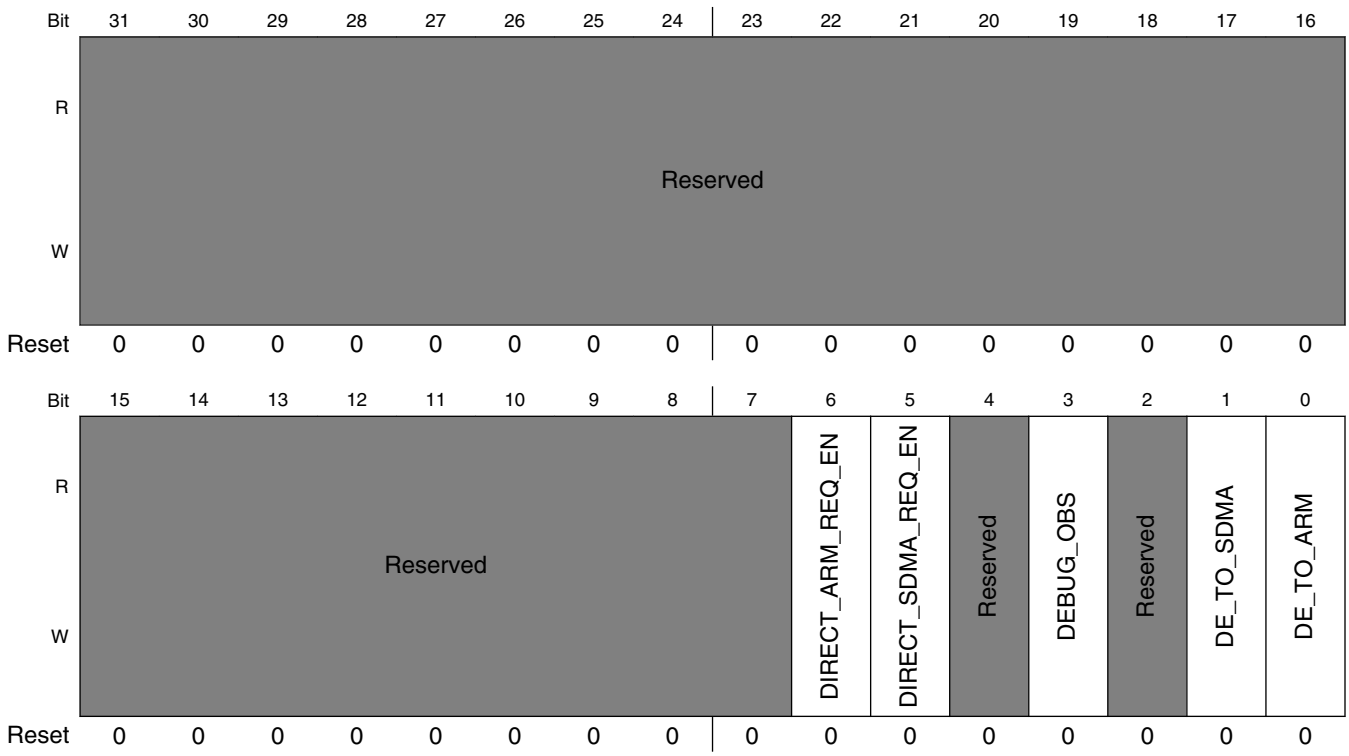
SJC_GPSSR field descriptions

Field	Description
GPSSR	General Purpose Secured Status Register Register is used for testing and debug.

4.13.8.5 Debug Control Register (SJC_DCR)

This register is used to control propagation of debug request from DE_B pad to the cores and debug signals from internal logic to the DE_B pad.

Address: 0h base + 4h offset = 4h



SJC_DCR field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 DIRECT_ARM_REQ_EN	Pass Debug Enable event from DE_B pin to Arm platform debug request signal(s). This bit controls the propagation of debug request DE_B to the Arm platform. 0 Disable propagation of system debug to (DE_B pin) to Arm platform. 1 Enable propagation of system debug to (DE_B pin) to Arm platform.
5 DIRECT_SDMA_REQ_EN	Debug enable of the sdma debug request This bit controls the propagation of debug request DE_B to the sdma. 0 Disable propagation of system debug to (DE_B pin) to sdma. 1 Enable propagation of system debug to (DE_B pin) to sdma.
4 -	This field is reserved. Reserved

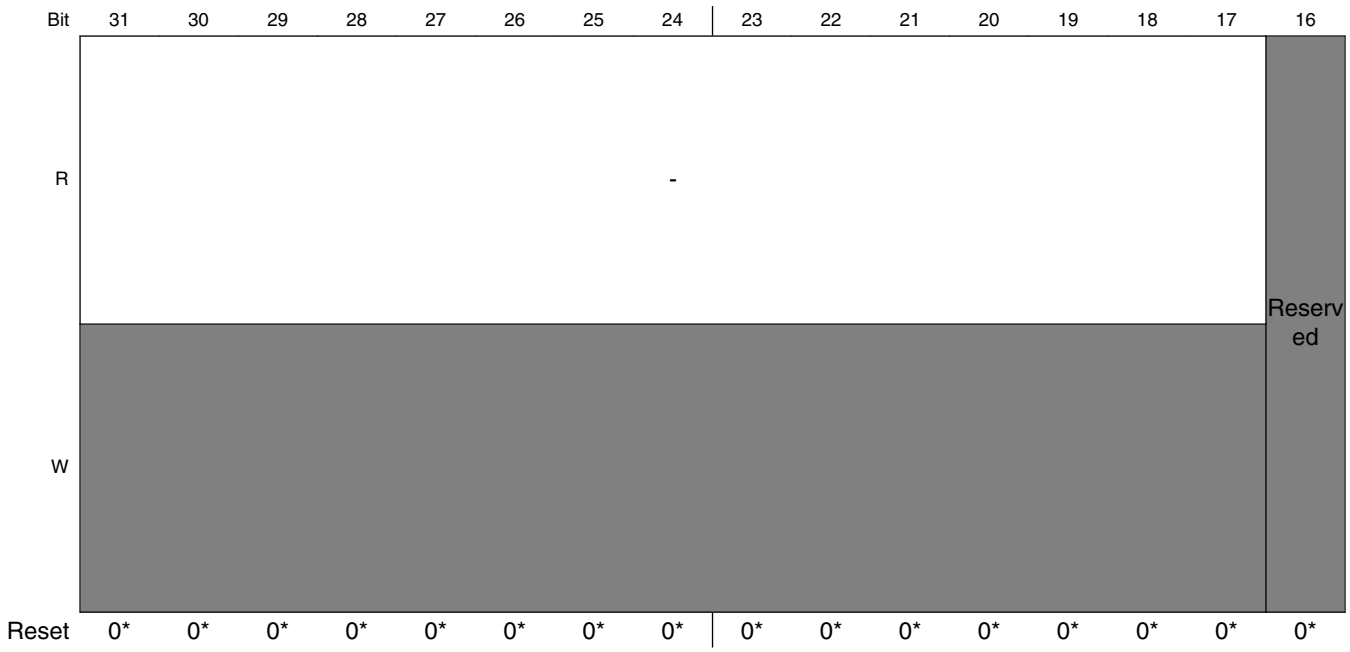
Table continues on the next page...

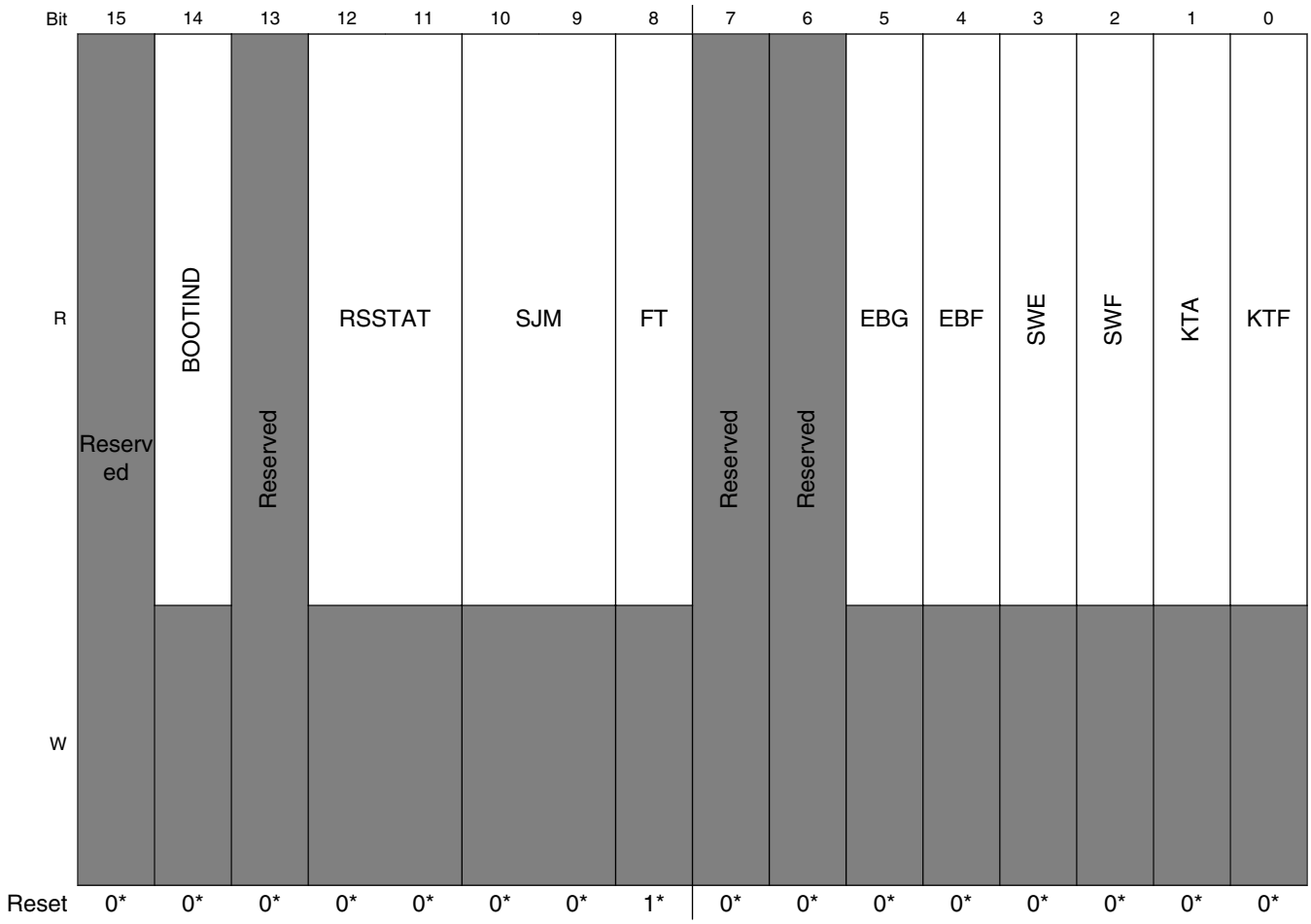
SJC_DCR field descriptions (continued)

Field	Description
3 DEBUG_OBS	Debug observability This bit controls the propagation of the "system debug" input to SJC (driven by the ECT logic), to the DE_B pad. (This logic can be used to pass debug acknowledge event from ECT out to the PAD, for example). The SJC's "system_debug" input is tied to logic HIGH value, therefore, set of "debug_obs" bit, will result in unconditional assertion of DE_B pad. 0 Disable propagation of system debug to DE_B pin 1 Unconditional assertion of pad DE_B
2 -	This field is reserved. Reserved
1 DE_TO_SDMA	SDMA debug request input propagation This bit controls the propagation of debug request to SDMA, when the JTAG state machine is put in "ENTER_DEBUG" IR instruction.. 0 Disable propagation of debug request to SDMA 1 Enable propagation of debug request to SDMA
0 DE_TO_ARM	Arm platform debug request input propagation This bit controls the propagation of debug request to Arm platform ("dbgreq"), when the JTAG state machine is put in "ENTER_DEBUG" IR instruction. 0 Disable propagation of debug request to Arm platform 1 Enable propagation of debug request to Arm platform

4.13.8.6 Security Status Register (SJC_SSR)

Address: 0h base + 5h offset = 5h





* Notes:
- The SJM reset value, reflects the JTAG security state, as defined by status of JTAG_SMODE[1:0] fuses. See the SJM bitfield description for details on valid values.

SJC_SSR field descriptions

Field	Description
31–17 -	Reserved.
16–15 -	This field is reserved. Reserved
14 BOOTIND	Boot Indication Inverted Internal Boot indication, i.e inverse of SRC: "src_int_boot" signal
13 -	This field is reserved. Reserved
12–11 RSSTAT	Response status Response status bits 00 Response wasn't entered 01 Response was entered but not verified

Table continues on the next page...

SJC_SSR field descriptions (continued)

Field	Description
	10 Response was entered and is incorrect 11 Response is correct
10–9 SJM	SJC Secure mode Secure JTAG mode, as set by external fuses. 00 No debug (#1) 01 Secure JTAG (#2) 10 Reserved 11 JTAG enabled (#3)
8 FT	Fuse type Fuse type bit - e-fuse or laser fuse 0 E-fuse technology 1 Laser fuse technology
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5 EBG	External boot granted External boot enabled, requested and granted 1 granted 0 not granted
4 EBF	External Boot fuse Status of the external boot disable fuse 0 (intact) - external boot is allowed 1 (burned) - external boot is disabled
3 SWE	SW enable SW JTAG enable status 1 enabled 0 disabled
2 SWF	Software JTAG enable fuse Status of the no SW disable JTAG fuse 0 (intact) - SW enable possible 1 (intact) - no SW enable possible
1 KTA	Kill Trace is active 1 active 0 not active
0 KTF	Kill Trace Enable fuse value 0 (intact) - kill trace is never active 1 (burned) - kill trace functionality enabled

4.13.8.7 General Purpose Clocks Control Register (SJC_GPCCR)

This register is used to configure clock related modes in SOC, see System Configuration chapter for more information. Those bits are directly connected to JTAG outputs. Bit 0 of GPCCR controls SDMA clocks invocation. When out of reset, the SDMA is in sleep mode with no SDMA clock running. Unlike events, debug requests does not wake SDMA if it is in sleep mode. The debug request is recognized by the SDMA only when it exits sleep mode upon reception of an event. To be able to enter debug mode even if no event is triggered, the SDMA clock on bit needs to be set prior to sending the debug request (clear at reset).

Address: 0h base + 7h offset = 7h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	-															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	-														ACLKOFFDIS	SCLKR
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SJC_GPCCR field descriptions

Field	Description
31–2 -	Reserved
1 ACLKOFFDIS	Disable/prevent Arm platform clock/power shutdown
0 SCLKR	SDMA Clock ON Register - This bit forces the clock on of the SDMA

Chapter 5
Clocks and Power Management

5.1 Clock Control Module (CCM)

5.1.1 Overview

Clock Control Module (CCM) manages the on-chip module clocks. CCM receives clocks from PLLs and oscillators and creates clocks for on-chip peripherals through a set of multiplexers, dividers and gates. When entering or exiting a low power mode, CCM automatically turns on and off PLLs and peripheral clocks.

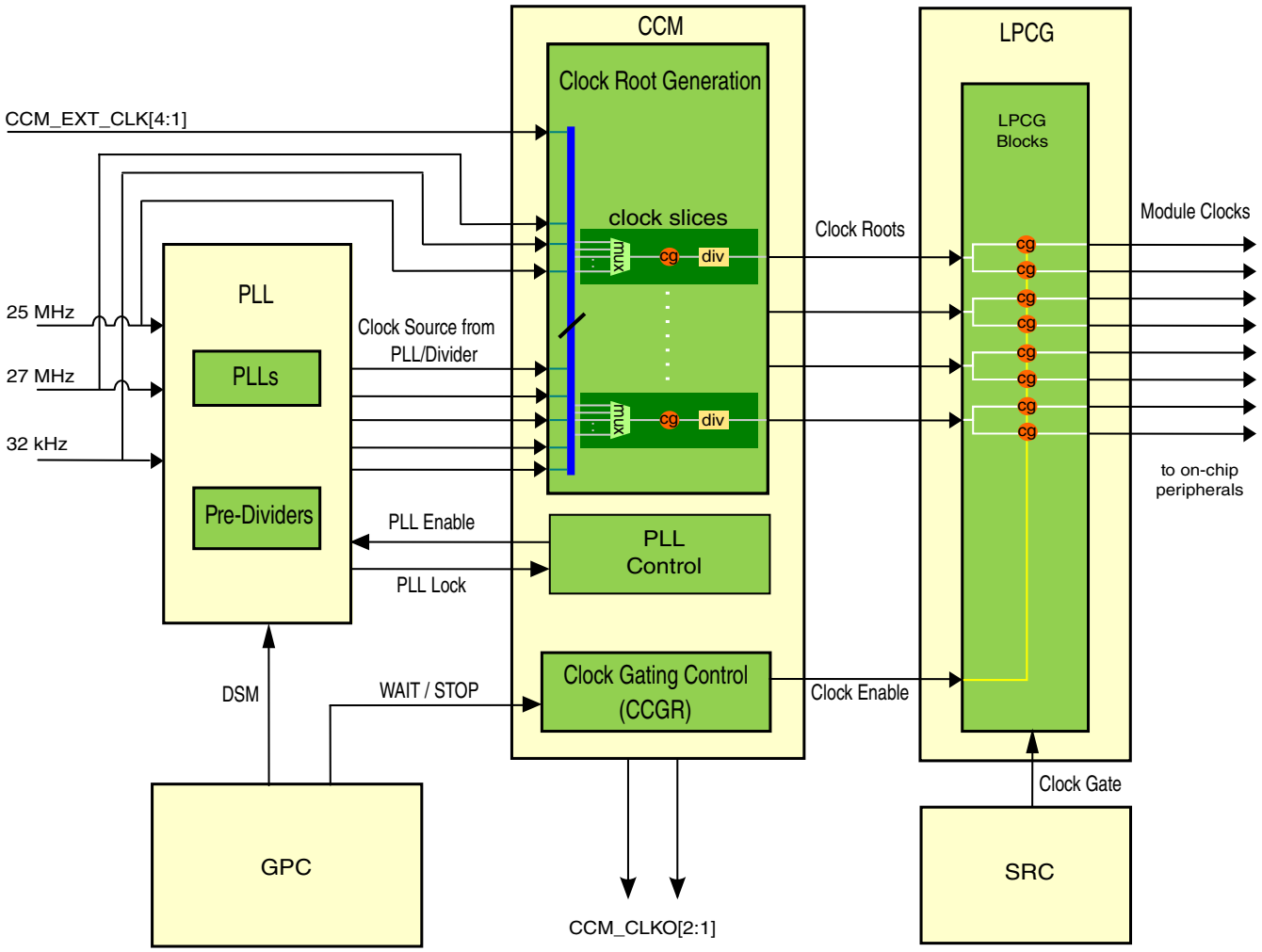


Figure 5-1. CCM Block Diagram

5.1.2 Clock Root Selects

The table below details the clock root slices and clock source inputs.

NOTE

The value of all clock root slice registers are zero after POR with the exception of DRAM clock. Please see System Boot for ROM reset values and default frequency settings for the clock root slices.

Table 5-1. Clock Root Table

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
0	ARM_A53_CLK_ROOT	1000	000 - 25M_REF_CLK 001 - ARM_PLL_CLK 010 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL1_DIV2 110 - AUDIO_PLL1_CLK 111 - SYSTEM_PLL3_CLK
1	ARM_M4_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL2_DIV4 011 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL1_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - SYSTEM_PLL3_CLK
2	VPU_A53_CLK_ROOT	800	000 - 25M_REF_CLK 001 - ARM_PLL_CLK 010 - SYSTEM_PLL2_DIV2 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL1_DIV2 110 - AUDIO_PLL1_CLK 111 - VPU_PLL_CLK
3	GPU_CORE_CLK_ROOT	800	000 - 25M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
4	GPU_SHADER_CLK_ROOT	1000	000 - 25M_REF_CLK 001 - GPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
16	MAIN_AXI_CLK_ROOT	333	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV3 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV4 100 - SYSTEM_PLL2_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - SYSTEM_PLL1_DIV8
17	ENET_AXI_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV4 100 - SYSTEM_PLL2_DIV5 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - SYSTEM_PLL3_CLK
18	NAND_USDHC_BUS_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV5 100 - SYSTEM_PLL1_DIV6 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - AUDIO_PLL1_CLK
19	VPU_BUS_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - VPU_PLL_CLK 011 - AUDIO_PLL2_CLK 100 - SYSTEM_PLL3_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL2_DIV5 111 - SYSTEM_PLL1_DIV8
20	DISPLAY_AXI_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV8 010 - SYSTEM_PLL1_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL1_DIV20 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1 111 - EXT_CLK_4
21	DISPLAY_APB_CLK_ROOT	200	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV8 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL1_DIV20 101 - AUDIO_PLL2_CLK 110 - EXT_CLK_1 111 - EXT_CLK_3
22	DISPLAY_RTRM_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL1_DIV2 100 - AUDIO_PLL1_CLK 101 - VIDEO_PLL1_CLK 110 - EXT_CLK_2 111 - EXT_CLK_3
23	USB_BUS_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV2 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL2_DIV5 101 - EXT_CLK_2 110 - EXT_CLK_4 111 - AUDIO_PLL2_CLK
24	GPU_AXI_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - GPU_PLL_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
25	GPU_AHB_CLK_ROOT	400	000 - 25M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			001 - SYSTEM_PLL1_CLK 010 - GPU_PLL_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
26	NOC_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - SYSTEM_PLL3_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL2_DIV2 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
27	NOC_APB_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV2 010 - SYSTEM_PLL3_CLK 011 - SYSTEM_PLL2_DIV3 100 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL1_CLK 110 - AUDIO_PLL1_CLK 111 - VIDEO_PLL1_CLK
32	AHB_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL1_DIV2 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK 111 - VIDEO_PLL1_CLK
34	AUDIO_AHB_CLK_ROOT	500	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV2 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL2_DIV6 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL1_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			111 - VIDEO_PLL1_CLK
36	MIPI_DSI_ESC_RX_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
64	DRAM_ALT_CLK_ROOT	800	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - SYSTEM_PLL1_DIV8 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL2_DIV4 101 - SYSTEM_PLL1_DIV2 110 - AUDIO_PLL1_CLK 111 - SYSTEM_PLL1_DIV3
65	DRAM_APB_CLK_ROOT	200	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL1_DIV5 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - AUDIO_PLL2_CLK
66	VPU_G1_CLK_ROOT	600	000 - 25M_REF_CLK 001 - VPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL1_DIV8 101 - SYSTEM_PLL2_DIV8 110 - SYSTEM_PLL3_CLK 111 - AUDIO_PLL1_CLK
67	VPU_G2_CLK_ROOT	300	000 - 25M_REF_CLK 001 - VPU_PLL_CLK 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_CLK 100 - SYSTEM_PLL1_DIV8

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			101 - SYSTEM_PLL2_DIV8
			110 - SYSTEM_PLL3_CLK
			111 - AUDIO_PLL1_CLK
68	DISPLAY_DTRC_CLK_ROOT	600	000 - 25M_REF_CLK
			001 - VIDEO_PLL2_CLK
			010 - SYSTEM_PLL1_CLK
			011 - SYSTEM_PLL2_CLK
			100 - SYSTEM_PLL1_DIV5
			101 - VIDEO_PLL1_CLK
			110 - SYSTEM_PLL3_CLK
			111 - AUDIO_PLL2_CLK
69	DISPLAY_DC8000_CLK_ROOT	600	000 - 25M_REF_CLK
			001 - VIDEO_PLL2_CLK
			010 - SYSTEM_PLL1_CLK
			011 - SYSTEM_PLL2_CLK
			100 - SYSTEM_PLL1_DIV5
			101 - VIDEO_PLL1_CLK
			110 - SYSTEM_PLL3_CLK
			111 - AUDIO_PLL2_CLK
70	PCIE1_CTRL_CLK_ROOT	250	000 - 25M_REF_CLK
			001 - SYSTEM_PLL2_DIV4
			010 - SYSTEM_PLL2_DIV5
			011 - SYSTEM_PLL1_DIV3
			100 - SYSTEM_PLL1_CLK
			101 - SYSTEM_PLL2_DIV2
			110 - SYSTEM_PLL2_DIV3
			111 - SYSTEM_PLL3_CLK
71	PCIE1_PHY_CLK_ROOT	100	000 - 25M_REF_CLK
			001 - SYSTEM_PLL2_DIV10
			010 - SYSTEM_PLL2_DIV2
			011 - EXT_CLK_1
			100 - EXT_CLK_2
			101 - EXT_CLK_3
			110 - EXT_CLK_4
			111 - SYSTEM_PLL1_DIV2
72	PCIE1_AUX_CLK_ROOT	10	000 - 25M_REF_CLK
			001 - SYSTEM_PLL2_DIV5
			010 - SYSTEM_PLL2_DIV20

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_DIV10 101 - SYSTEM_PLL1_DIV10 110 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV4
73	DC_PIXEL_CLK_ROOT	594	000 - 25M_REF_CLK 001 - VIDEO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - AUDIO_PLL1_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL3_CLK 111 - EXT_CLK_4
74	LCDIF_PIXEL_CLK_ROOT	148.5	000 - 25M_REF_CLK 001 - VIDEO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - AUDIO_PLL1_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL3_CLK 111 - EXT_CLK_4
75	SAI1_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_1 111 - EXT_CLK_2
76	SAI2_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_2 111 - EXT_CLK_3
77	SAI3_CLK_ROOT	66	000 - 25M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4
78	SAI4_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_1 111 - EXT_CLK_2
79	SAI5_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_2 111 - EXT_CLK_3
80	SAI6_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4
81	SPDIF1_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_2

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			111 - EXT_CLK_3
82	SPDIF2_CLK_ROOT	66	000 - 25M_REF_CLK 001 - AUDIO_PLL1_CLK 010 - AUDIO_PLL2_CLK 011 - VIDEO_PLL1_CLK 100 - SYSTEM_PLL1_DIV6 101 - 27M_REF_CLK 110 - EXT_CLK_3 111 - EXT_CLK_4
83	ENET_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV8 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL1_DIV5 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - EXT_CLK_4
84	ENET_TIMER_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - AUDIO_PLL1_CLK 011 - EXT_CLK_1 100 - EXT_CLK_2 101 - EXT_CLK_3 110 - EXT_CLK_4 111 - VIDEO_PLL1_CLK
85	ENET_PHY_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV20 010 - SYSTEM_PLL2_DIV8 011 - SYSTEM_PLL2_DIV5 100 - SYSTEM_PLL2_DIV2 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - AUDIO_PLL2_CLK
86	NAND_CLK_ROOT	500	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV2 010 - AUDIO_PLL1_CLK 011 - SYSTEM_PLL1_DIV2 100 - AUDIO_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - VIDEO_PLL1_CLK
87	QSPI_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV2 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV2 100 - AUDIO_PLL2_CLK 101 - SYSTEM_PLL1_DIV3 110 - SYSTEM_PLL3_CLK 111 - SYSTEM_PLL1_DIV8
88	USDHC1_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV2 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL3_CLK 101 - SYSTEM_PLL1_DIV3 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV8
89	USDHC2_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV2 010 - SYSTEM_PLL1_CLK 011 - SYSTEM_PLL2_DIV2 100 - SYSTEM_PLL3_CLK 101 - SYSTEM_PLL1_DIV3 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV8
90	I2C1_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 101 - VIDEO_PLL1_CLK 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV6
91	I2C2_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL2_DIV20

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 101 - VIDEO_PLL1_CLK 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV6
92	I2C3_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 101 - VIDEO_PLL1_CLK 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV6
93	I2C4_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV5 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - AUDIO_PLL1_CLK 101 - VIDEO_PLL1_CLK 110 - AUDIO_PLL2_CLK 111 - SYSTEM_PLL1_DIV6
94	UART1_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4 111 - AUDIO_PLL2_CLK
95	UART2_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_2 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
96	UART3_CLK_ROOT	80	000 - 25M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4 111 - AUDIO_PLL2_CLK
97	UART4_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV10 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_2 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
98	USB_CORE_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV8 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL2_DIV5 101 - EXT_CLK_2 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
99	USB_PHY_REF_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV8 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL2_DIV5 101 - EXT_CLK_2 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
100	GIC_CLK_ROOT	400	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL2_DIV10 100 - SYSTEM_PLL1_CLK 101 - EXT_CLK_2 110 - EXT_CLK_4

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			111 - AUDIO_PLL2_CLK
101	ECSPI1_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL1_DIV5 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - AUDIO_PLL2_CLK
102	ECSPI2_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL1_DIV5 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - AUDIO_PLL2_CLK
103	PWM1_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV5 011 - SYSTEM_PLL1_DIV20 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_1 110 - SYSTEM_PLL1_DIV10 111 - VIDEO_PLL1_CLK
104	PWM2_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV5 011 - SYSTEM_PLL1_DIV20 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_1 110 - SYSTEM_PLL1_DIV10 111 - VIDEO_PLL1_CLK
105	PWM3_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV5 011 - SYSTEM_PLL1_DIV20 100 - SYSTEM_PLL3_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			101 - EXT_CLK_2 110 - SYSTEM_PLL1_DIV10 111 - VIDEO_PLL1_CLK
106	PWM4_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV5 011 - SYSTEM_PLL1_DIV20 100 - SYSTEM_PLL3_CLK 101 - EXT_CLK_2 110 - SYSTEM_PLL1_DIV10 111 - VIDEO_PLL1_CLK
107	GPT1_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_1
108	GPT2_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_2
109	GPT3_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_3
110	GPT4_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_1
111	GPT5_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_2
112	GPT6_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL1_DIV20 100 - VIDEO_PLL1_CLK 101 - SYSTEM_PLL1_DIV10 110 - AUDIO_PLL1_CLK 111 - EXT_CLK_3
113	TRACE_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL1_DIV5 011 - VPU_PLL_CLK 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_1 111 - EXT_CLK_3
114	WDOG_CLK_ROOT	66	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV6 010 - SYSTEM_PLL1_DIV5 011 - VPU_PLL_CLK 100 - SYSTEM_PLL2_DIV8 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL1_DIV10 111 - SYSTEM_PLL2_DIV6
115	WRCLK_CLK_ROOT	40	000 - 25M_REF_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			001 - SYSTEM_PLL1_DIV20 010 - VPU_PLL_CLK 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_DIV5 101 - SYSTEM_PLL1_DIV3 110 - SYSTEM_PLL2_DIV2 111 - SYSTEM_PLL1_DIV8
116	IPP_DO_CLKO1	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_CLK 010 - 27M_REF_CLK 011 - SYSTEM_PLL1_DIV4 100 - AUDIO_PLL2_CLK 101 - SYSTEM_PLL2_DIV2 110 - VPU_PLL_CLK 111 - SYSTEM_PLL1_DIV10
117	IPP_DO_CLKO2	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV2 011 - SYSTEM_PLL2_DIV6 100 - SYSTEM_PLL3_CLK 101 - AUDIO_PLL1_CLK 110 - VIDEO_PLL1_CLK 111 - 32K_REF_CLK
118	MIPI_DSI_CORE_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL2_DIV4 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
119	MIPI_DSI_PHY_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV8 010 - SYSTEM_PLL2_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - EXT_CLK_2 110 - AUDIO_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			111 - VIDEO_PLL1_CLK
120	MIPI_DSI_DBI_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL2_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
121	OLD_MIPI_DSI_ESC_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
122	MIPI_CSI1_CORE_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL2_DIV4 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
123	MIPI_CSI1_PHY_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV3 010 - SYSTEM_PLL2_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - EXT_CLK_2 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
124	MIPI_CSI1_ESC_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
125	MIPI_CSI2_CORE_CLK_ROOT	266	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV3 010 - SYSTEM_PLL2_DIV4 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
126	MIPI_CSI2_PHY_REF_CLK_ROOT	125	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV3 010 - SYSTEM_PLL2_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - EXT_CLK_2 110 - AUDIO_PLL2_CLK 111 - VIDEO_PLL1_CLK
127	MIPI_CSI2_ESC_CLK_ROOT	133	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
128	PCIE2_CTRL_CLK_ROOT	250	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV4 010 - SYSTEM_PLL2_DIV5 011 - SYSTEM_PLL1_DIV3 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_DIV2 110 - SYSTEM_PLL2_DIV3 111 - SYSTEM_PLL3_CLK
129	PCIE2_PHY_CLK_ROOT	100	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL2_DIV2

Table continues on the next page...

Table 5-1. Clock Root Table (continued)

Clock Slice	Clock Root	Max Frequency (MHz)	Source Select (CCM_TARGET_ROOT[MUX])
			011 - EXT_CLK_1 100 - EXT_CLK_2 101 - EXT_CLK_3 110 - EXT_CLK_4 111 - SYSTEM_PLL1_DIV2
130	PCIE2_AUX_CLK_ROOT	10	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL2_DIV20 011 - SYSTEM_PLL3_CLK 100 - SYSTEM_PLL2_DIV10 101 - SYSTEM_PLL1_DIV10 110 - SYSTEM_PLL1_DIV5 111 - SYSTEM_PLL1_DIV4
131	ECSPI3_CLK_ROOT	80	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV5 010 - SYSTEM_PLL1_DIV20 011 - SYSTEM_PLL1_DIV5 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL3_CLK 110 - SYSTEM_PLL2_DIV4 111 - AUDIO_PLL2_CLK
132	OLD_MIPI_DSI_ESC_RX_CLK_ROOT	60	000 - 25M_REF_CLK 001 - SYSTEM_PLL2_DIV10 010 - SYSTEM_PLL1_DIV10 011 - SYSTEM_PLL1_CLK 100 - SYSTEM_PLL2_CLK 101 - SYSTEM_PLL3_CLK 110 - EXT_CLK_3 111 - AUDIO_PLL2_CLK
133	DISPLAY_HDMI_CLK_ROOT	200	000 - 25M_REF_CLK 001 - SYSTEM_PLL1_DIV4 010 - SYSTEM_PLL2_DIV5 011 - VPU_PLL_CLK 100 - SYSTEM_PLL1_CLK 101 - SYSTEM_PLL2_CLK 110 - SYSTEM_PLL3_CLK 111 - EXT_CLK_4

5.1.3 Clock Tree

The figure below illustrates the clock sources from the PLLs.

NOTE

Some clock gates illustrated below are symbolic of distributed clock gates (multiple) and not a sole clock gate. These clock slices typically source multiple IP (e.g. bus clocks).

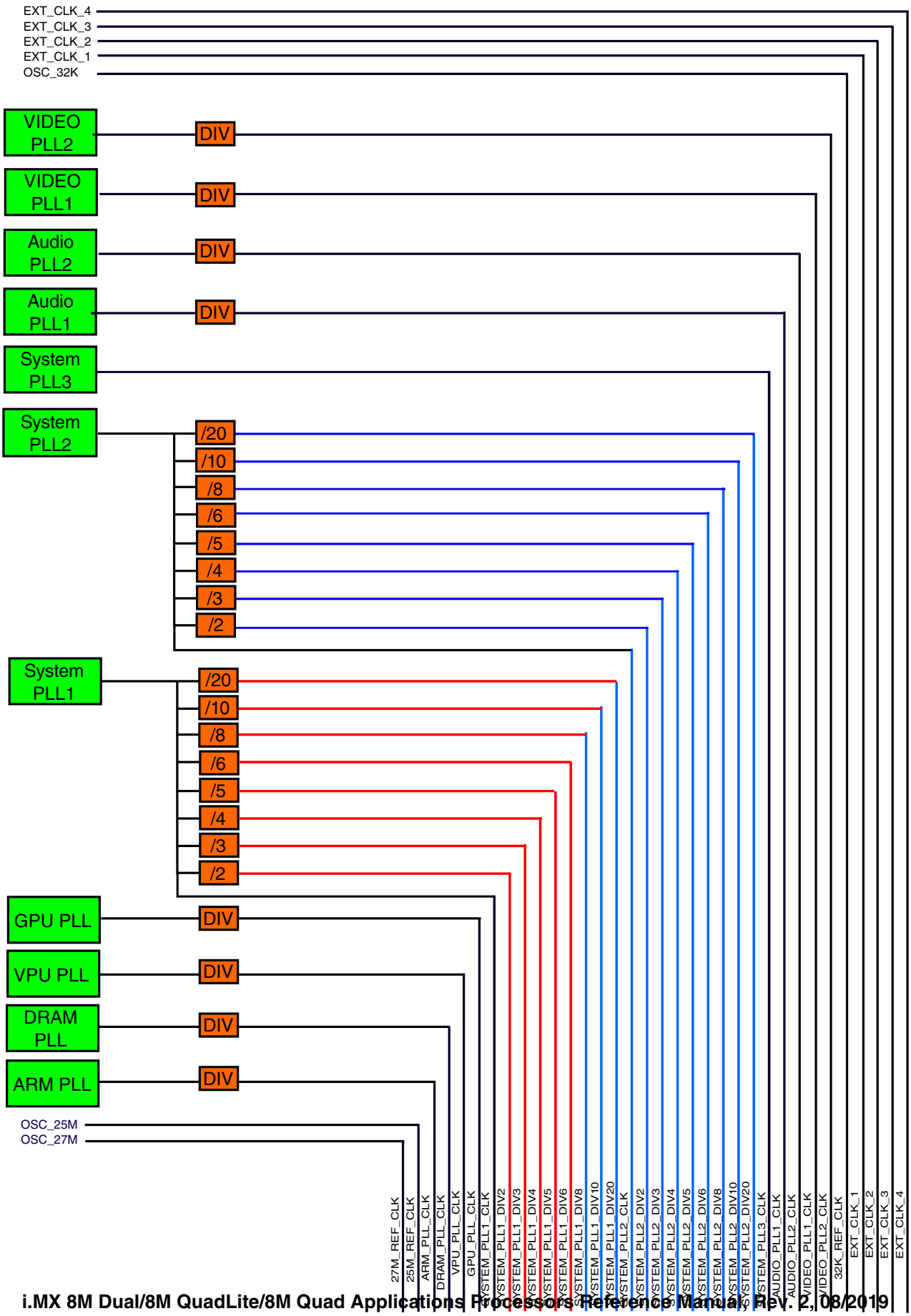
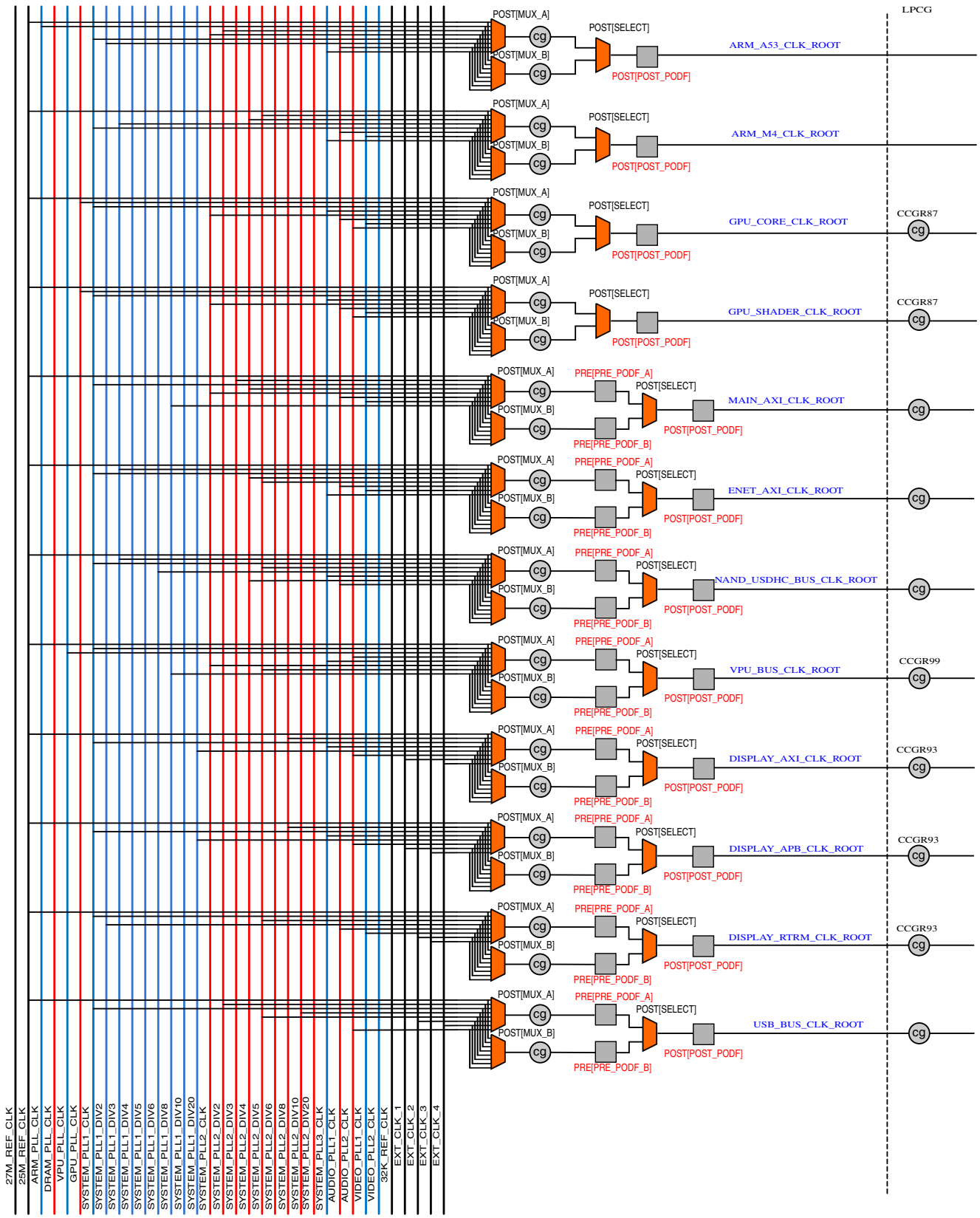
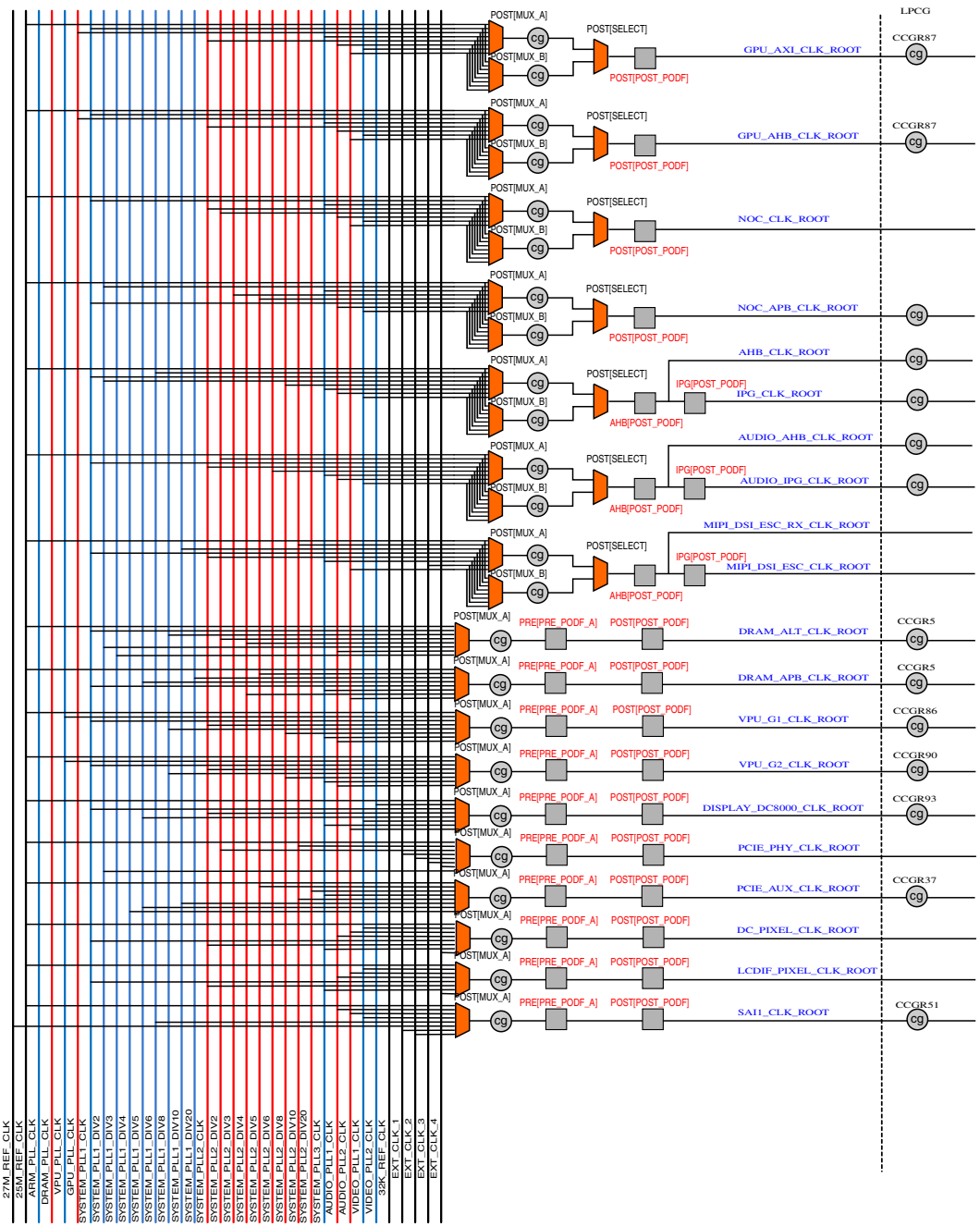
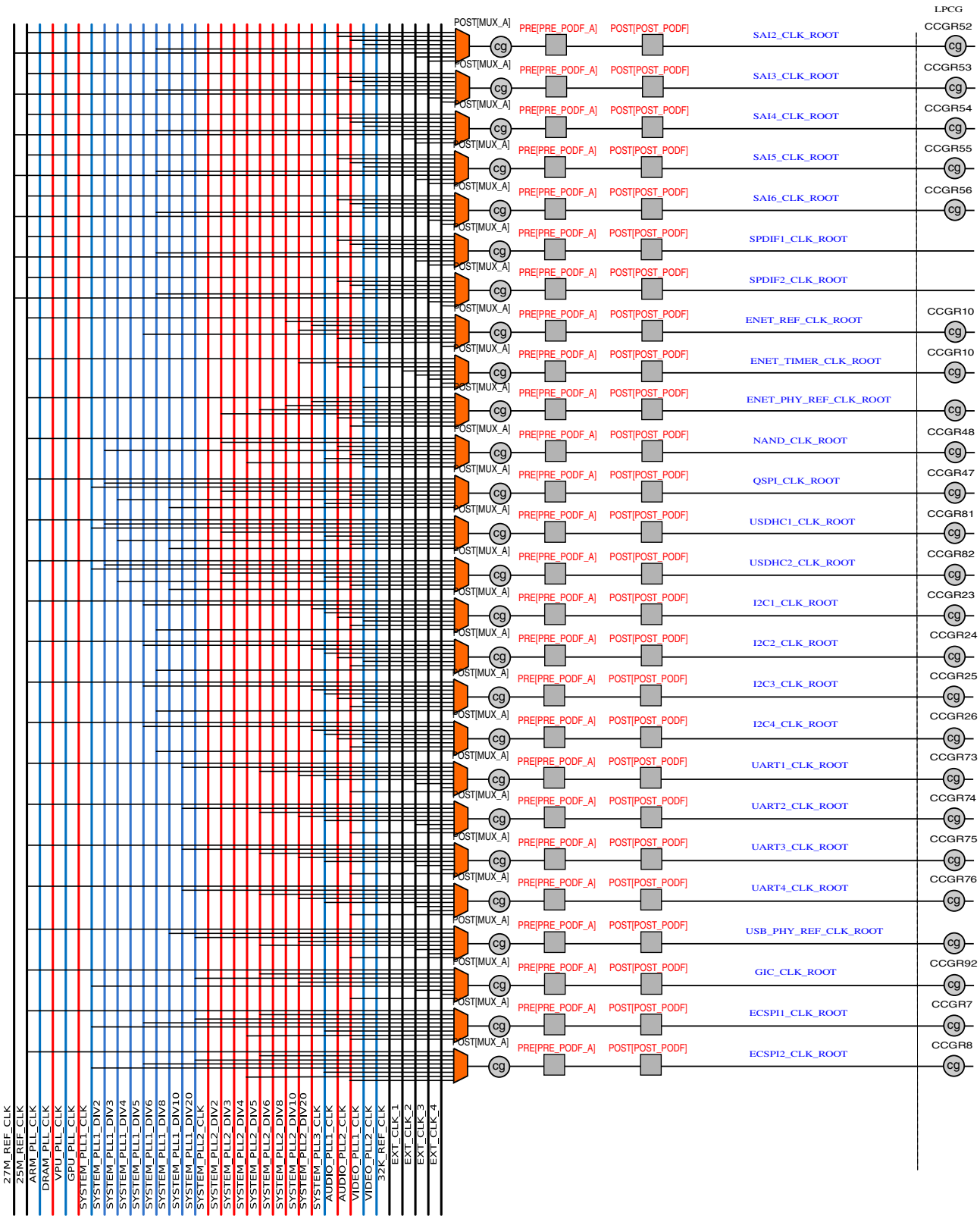


Figure 5-2. CCM input clock sources

The figure below illustrates the clock slices of CCM and clock root generation.




LPCG
27M_REF_CLK
25M_REF_CLK
ARM_PLL_CLK
DRAM_PLL_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL1_CLK
VIDEO_PLL2_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4
POST[MUX_A]
POST[MUX_B]
POST[SELECT]
POST[POST_PODF]
GPU_AXI_CLK_ROOT
CCGR87
GPU_AHB_CLK_ROOT
NOC_CLK_ROOT
NOC_APB_CLK_ROOT
AHB_CLK_ROOT
IPG[POST_PODF]
IPG_CLK_ROOT
AHB[POST_PODF]
AUDIO_AHB_CLK_ROOT
AUDIO_IPG_CLK_ROOT
MIPI_DSI_ESC_RX_CLK_ROOT
MIPI_DSI_ESC_CLK_ROOT
PRE[PRE_PODF_A]
DRAM_ALT_CLK_ROOT
CCGR5
DRAM_APB_CLK_ROOT
VPU_G1_CLK_ROOT
CCGR86
VPU_G2_CLK_ROOT
CCGR90
DISPLAY_DC8000_CLK_ROOT
CCGR93
PCIE_PHY_CLK_ROOT
PCIE_AUX_CLK_ROOT
CCGR37
DC_PIXEL_CLK_ROOT
LCDIF_PIXEL_CLK_ROOT
SAI1_CLK_ROOT
CCGR51
cg


LPCG
27M_REF_CLK
25M_REF_CLK
ARM_PLL_CLK
DRAM_PLL_CLK
VPU_PLL_CLK
GPU_PLL_CLK
SYSTEM_PLL1_CLK
SYSTEM_PLL1_DIV2
SYSTEM_PLL1_DIV3
SYSTEM_PLL1_DIV4
SYSTEM_PLL1_DIV5
SYSTEM_PLL1_DIV6
SYSTEM_PLL1_DIV8
SYSTEM_PLL1_DIV10
SYSTEM_PLL1_DIV20
SYSTEM_PLL2_CLK
SYSTEM_PLL2_DIV2
SYSTEM_PLL2_DIV3
SYSTEM_PLL2_DIV4
SYSTEM_PLL2_DIV5
SYSTEM_PLL2_DIV6
SYSTEM_PLL2_DIV8
SYSTEM_PLL2_DIV10
SYSTEM_PLL2_DIV20
SYSTEM_PLL3_CLK
AUDIO_PLL1_CLK
AUDIO_PLL2_CLK
VIDEO_PLL1_CLK
VIDEO_PLL2_CLK
32K_REF_CLK
EXT_CLK_1
EXT_CLK_2
EXT_CLK_3
EXT_CLK_4
POST[MUX_A]
cg
PRE[PRE_PODF_A]
POST[POST_PODF]
SAI2_CLK_ROOT
SAI3_CLK_ROOT
SAI4_CLK_ROOT
SAI5_CLK_ROOT
SAI6_CLK_ROOT
SPDIF1_CLK_ROOT
SPDIF2_CLK_ROOT
ENET_REF_CLK_ROOT
ENET_TIMER_CLK_ROOT
ENET_PHY_REF_CLK_ROOT
NAND_CLK_ROOT
QSPI_CLK_ROOT
USDHC1_CLK_ROOT
USDHC2_CLK_ROOT
I2C1_CLK_ROOT
I2C2_CLK_ROOT
I2C3_CLK_ROOT
I2C4_CLK_ROOT
UART1_CLK_ROOT
UART2_CLK_ROOT
UART3_CLK_ROOT
UART4_CLK_ROOT
USB_PHY_REF_CLK_ROOT
GIC_CLK_ROOT
ECSPI1_CLK_ROOT
ECSPI2_CLK_ROOT
CCGR52
CCGR53
CCGR54
CCGR55
CCGR56
CCGR10
CCGR10
CCGR48
CCGR47
CCGR81
CCGR82
CCGR23
CCGR24
CCGR25
CCGR26
CCGR73
CCGR74
CCGR75
CCGR76
CCGR92
CCGR7
CCGR8

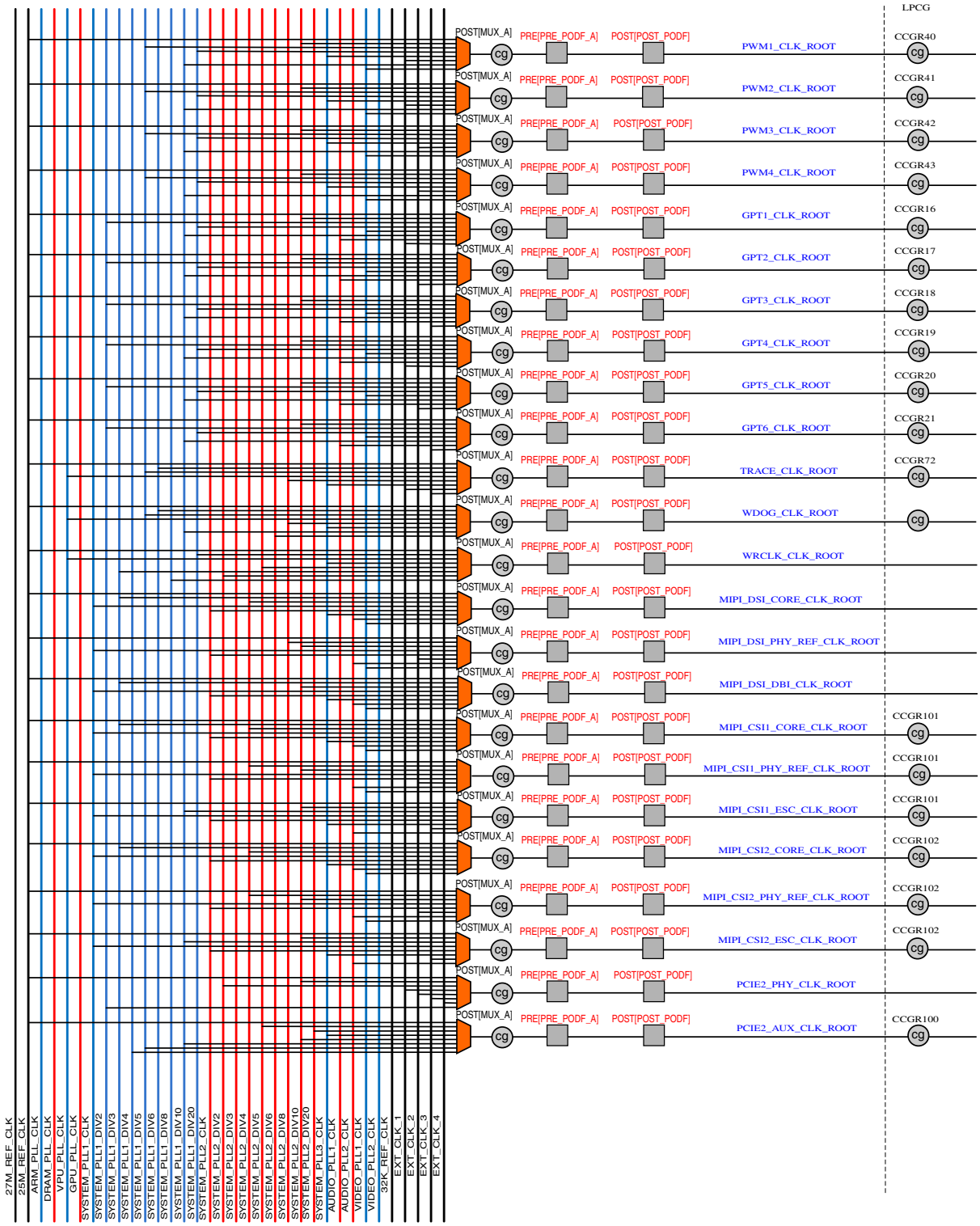


Figure 5-3. CCM Clock Tree Root Slices

5.1.4 System Clocks

The table below shows the CCM output module clock connectivity and gating.

Table 5-2. System Clocks and Gating

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
AIPS_TZ	aips_tz1.hclk	ccm_ahb_clk_root	clk_enable_ipmux1 (CCGR28)
	ipmux1.master_clk master_clk	ccm_ahb_clk_root	clk_enable_ipmux1 (CCGR28)
	ipmux1.slave_clk	ccm_ipg_clk_root	clk_enable_ipmux1 (CCGR28)
	aips_tz2.hclk	ccm_ahb_clk_root	clk_enable_ipmux2 (CCGR29)
	Ipmux2.master_clk	ccm_ahb_clk_root	clk_enable_ipmux2 (CCGR29)
	Ipmux2.slave_clk	ccm_ipg_clk_root	clk_enable_ipmux2 (CCGR29)
	aips_tz3.hclk	ccm_ahb_clk_root	clk_enable_ipmux3 (CCGR30)
	Ipmux3.master_clk master_clk	ccm_ahb_clk_root	clk_enable_ipmux3 (CCGR30)
	Ipmux3.slave_clk	ccm_ipg_clk_root	clk_enable_ipmux3 (CCGR30)
APBHDMA	apbhdma.hclk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
	apbhdma_sec.mst_hclk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
	rawnand.u_bch_input_apb_clk u_bch_input_apb_clk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
	rawnand.u_gpmi_bch_input_bch_clk u_gpmi_bch_input_bch_clk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
	rawnand.u_gpmi_bch_input_gpmi_io_clk	ccm_nand_clk_root	clk_enable_rawnand (CCGR48)
	rawnand.U_gpmi_input_apb_clk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
CAAM	caam.aclk	ccm_ahb_clk_root	1
	caam.ipg_clk	ccm_ipg_clk_root	1
	caam.ipg_clk_s	ccm_ipg_clk_root	1
	caam_exsc.aclk_exsc	ccm_ahb_clk_root	1
	caam_mem.clk	ccm_ahb_clk_root	1
CM4	cm4.cm4_cti_clk	ccm_arm_m4_clk_root	GPC controlled
	cm4.cm4_fclk	ccm_arm_m4_clk_root	GPC controlled

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	cm4.cm4_hclk	ccm_arm_m4_clk_root	GPC controlled
	cm4.dap_clk	ccm_ahb_clk_root	GPC controlled
	cm4.ipg_clk_nic	ccm_arm_m4_clk_root	GPC controlled
	cm4.tcmc_hclk	ccm_arm_m4_clk_root	GPC controlled
	cm4_mem.tcmc_hclk	ccm_arm_m4_clk_root	GPC controlled
	cm4_sec.ipg_clk	ccm_ipg_clk_root	GPC controlled
	cm4_sec.ipg_clk_s	ccm_ipg_clk_root	GPC controlled
	cm4_sec.mst_hclk	ccm_arm_m4_clk_root	GPC controlled
CSI	csi2_1.clk_vid	ccm_mipi_csi1_phy_ref_clk_root	1
	csi2_1.clk	ccm_mipi_csi1_core_clk_root	clk_enable_mipi_csi1 (CCGR101)
	csi2_1.clk_esc	ccm_mipi_csi1_esc_clk_root	clk_enable_mipi_csi1 (CCGR101)
	csi2_1.pclk	ccm_mipi_csi1_core_clk_root	1
	csi2_1.clk_ui	ccm_mipi_csi1_phy_ref_clk_root	clk_enable_mipi_csi1 (CCGR101)
	csi2_2.clk_vid	ccm_mipi_csi2_phy_ref_clk_root	1
	csi2_2.clk	ccm_mipi_csi2_core_clk_root	clk_enable_mipi_csi2 (CCGR102)
	csi2_2.clk_esc	ccm_mipi_csi2_esc_clk_root	clk_enable_mipi_csi2 (CCGR102)
	csi2_2.pclk	ccm_mipi_csi2_core_clk_root	1
	csi2_2.clk_ui	ccm_mipi_csi2_phy_ref_clk_root	clk_enable_mipi_csi2 (CCGR102)
CSU	csu.ipg_clk_s	ccm_ipg_clk_root	clk_enable_csu (CCGR3)
DAP	dap.dapclk_2_2	ccm_ahb_clk_root	clk_enable_debug (CCGR4)
ECSPI	ecspi1.ipg_clk	ccm_ipg_clk_root	clk_enable_ecspi1 (CCGR7)
	ecspi1.ipg_clk_per	ccm_ecspi1_clk_root	clk_enable_ecspi1 (CCGR7)
	ecspi1.ipg_clk_s	ccm_ipg_clk_root	clk_enable_ecspi1 (CCGR7)
	ecspi2.ipg_clk	ccm_ipg_clk_root	clk_enable_ecspi2 (CCGR8)
	ecspi2.ipg_clk_per	ccm_ecspi2_clk_root	clk_enable_ecspi2 (CCGR8)
	ecspi2.Ipg_clk_s	ccm_ipg_clk_root	clk_enable_ecspi2 (CCGR8)
	ecspi3.ipg_clk	ccm_ipg_clk_root	clk_enable_ecspi3 (CCGR9)
	ecspi3.ipg_clk_per	ccm_ecspi3_clk_root	clk_enable_ecspi3 (CCGR9)
	ecspi3.Ipg_clk_s	ccm_ipg_clk_root	clk_enable_ecspi3 (CCGR9)
ENET	enet1.ipp_ind_mac0_txclk	ccm_enet_ref_clk_root	clk_enable_enet1 (CCGR10)
	enet1.ipg_clk	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_mac0	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_mac0_s	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	enet1.ipg_clk_s	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)
	enet1.ipg_clk_time	ccm_enet_timer_clk_root	clk_enable_enet1 (CCGR10)
	enet1_mem.mac0_rxmem_clk	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)
	enet1_mem.mac0_txmem_clk	ccm_enet_axi_clk_root	clk_enable_enet1 (CCGR10)
GPIO	gpio1.ipg_clk_s	ccm_ipg_clk_root	clk_enable_gpio1 (CCGR11)
	gpio2.ipg_clk_s	ccm_ipg_clk_root	clk_enable_gpio2 (CCGR12)
	gpio3.ipg_clk_s	ccm_ipg_clk_root	clk_enable_gpio3 (CCGR13)
	gpio4.ipg_clk_s	ccm_ipg_clk_root	clk_enable_gpio4 (CCGR14)
	gpio5.ipg_clk_s	ccm_ipg_clk_root	clk_enable_gpio5 (CCGR15)
GPT	gpt1.ipg_clk	ccm_gpt1_clk_root	clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_highfreq	ccm_gpt1_clk_root	clk_enable_gpt1 (CCGR16)
	gpt1.ipg_clk_s	ccm_gpt1_clk_root	clk_enable_gpt1 (CCGR16)
	gpt2.ipg_clk	ccm_gpt2_clk_root	clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_highfreq	ccm_gpt2_clk_root	clk_enable_gpt2 (CCGR17)
	gpt2.ipg_clk_s	ccm_gpt2_clk_root	clk_enable_gpt2 (CCGR17)
	gpt3.ipg_clk	ccm_gpt3_clk_root	clk_enable_gpt3 (CCGR18)
	gpt3.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt3 (CCGR18)
	gpt3.ipg_clk_highfreq	ccm_gpt3_clk_root	clk_enable_gpt3 (CCGR18)
	gpt3.ipg_clk_s	ccm_gpt3_clk_root	clk_enable_gpt3 (CCGR18)
	gpt4.ipg_clk	ccm_gpt4_clk_root	clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_highfreq	ccm_gpt4_clk_root	clk_enable_gpt4 (CCGR19)
	gpt4.ipg_clk_s	ccm_gpt4_clk_root	clk_enable_gpt4 (CCGR19)
	gpt5.ipg_clk	ccm_gpt5_clk_root	clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_highfreq	ccm_gpt5_clk_root	clk_enable_gpt5 (CCGR20)
	gpt5.ipg_clk_s	ccm_gpt5_clk_root	clk_enable_gpt5 (CCGR20)
	gpt6.ipg_clk	ccm_gpt6_clk_root	clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_24m	anamix_osc_25m_clk	clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_highfreq	ccm_gpt6_clk_root	clk_enable_gpt6 (CCGR21)
	gpt6.ipg_clk_s	ccm_gpt6_clk_root	clk_enable_gpt6 (CCGR21)
I2C	i2c1.ipg_clk_patref	ccm_i2c1_clk_root	clk_enable_i2c1 (CCGR23)
	i2c1.ipg_clk_s	ccm_i2c1_clk_root	clk_enable_i2c1 (CCGR23)
	i2c2.ipg_clk_patref	ccm_i2c2_clk_root	clk_enable_i2c2 (CCGR24)
	i2c2.ipg_clk_s	ccm_i2c2_clk_root	clk_enable_i2c2 (CCGR24)
	i2c3.ipg_clk_patref	ccm_i2c3_clk_root	clk_enable_i2c3 (CCGR25)
	i2c3.ipg_clk_s	ccm_i2c3_clk_root	clk_enable_i2c3 (CCGR25)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	i2c4.ipg_clk_patref	ccm_i2c4_clk_root	clk_enable_i2c4 (CCGR26)
	i2c4.ipg_clk_s	ccm_i2c4_clk_root	clk_enable_i2c4 (CCGR26)
IOMUXC	iomuxc.ipg_clk_s	ccm_ipg_clk_root	clk_enable_iomux (CCGR27)
	iomuxc_gpr.ipg_clk_s	ccm_ipg_clk_root	clk_enable_iomux (CCGR27)
	iomux.ipt_clk_io	ccm_ipg_clk_root	clk_enable_iomux (CCGR27)
LCD	lcdif.pix_clk	ccm_lcdif_pixel_clk_root	1
	lcdif.apb_clk	ccm_main_axi_clk_root	1
DSI	mipi.CLKREF	ccm_mipi_dsi_phy_ref_clk_root	1
	mipi.pclk	ccm_main_axi_clk_root	1
	mipi.RxClkEsc	ccm_mipi_dsi_esc_rx_clk_root	1
	mipi.TxClkEsc	ccm_mipi_dsi_esc_clk_root	1
MU	mu.ipg_clk_dsp	ccm_ipg_clk_root	clk_enable_mu (CCGR33)
	mu.ipg_clk_mcu	ccm_ipg_clk_root	clk_enable_mu (CCGR33)
	mu.ipg_clk_s_dsp	ccm_ipg_clk_root	clk_enable_mu (CCGR33)
	mu.ipg_clk_s_mcu	ccm_ipg_clk_root	clk_enable_mu (CCGR33)
OCOTP	ocotp.ipg_clk	ccm_ipg_clk_root	clk_enable_ocotp (CCGR34)
	ocotp.ipg_clk_s	ccm_ipg_clk_root	clk_enable_ocotp (CCGR34)
OCRAM	ocram_ctrl.clk	ccm_main_axi_clk_root	clk_enable_ocram (CCGR35)
	ocram_ctrl_s.clk	ccm_ahb_clk_root	clk_enable_ocram_s (CCGR36)
	ocram_exsc.aclk_exsc	ccm_main_axi_clk_root	clk_enable_ocram (CCGR35)
	ocram_exsc.ipg_clk	ccm_ipg_clk_root	clk_enable_ocram (CCGR35)
	ocram_mem.clk	ccm_main_axi_clk_root	clk_enable_ocram (CCGR35)
	ocram_s_exsc.aclk_exsc	ccm_ahb_clk_root	clk_enable_ocram_s (CCGR36)
	ocram_s_exsc.ipg_clk	ccm_ipg_clk_root	clk_enable_ocram_s (CCGR36)
	ocram_s_mem.clk	ccm_ahb_clk_root	clk_enable_ocram_s (CCGR36)
PCIE	pcie_clk_rst.auxclk	ccm_pcie_aux_clk_root	clk_enable_pcie (CCGR37)
	pcie_clk_rst.mstr_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_clk_rst.slv_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_ctrl.mstr_aclk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_ctrl.slv_aclk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_exsc.aclk_exsc	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_exsc.ipg_clk	ccm_ipg_clk_root	clk_enable_pcie (CCGR37)
	pcie_mem.mstr_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)
	pcie_mem.slv_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie (CCGR37)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	pcie2_clk_rst.auxclk	ccm_pcie_aux_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_clk_rst.mstr_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_clk_rst.slv_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_ctrl.mstr_aclk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_ctrl.slv_aclk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_exsc.aclk_exsc	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_exsc.ipg_clk	ccm_ipg_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_mem.mstr_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
	pcie2_mem.slv_axi_clk	ccm_main_axi_clk_root	clk_enable_pcie2 (CCGR100)
PCIE_PHY	pcie_phy.ref_alt_clk_p	ccm_pcie_phy_clk_root	1
	pcie2_phy.ref_alt_clk_p	ccm_pcie2_phy_clk_root	1
PERFMON	perfmon1.apb_clk	ccm_ipg_clk_root	clk_enable_perfmon1 (CCGR38)
	perfmon1.axi0_ACLK	ccm_main_axi_clk_root	clk_enable_perfmon1 (CCGR38)
	perfmon2.apb_clk	ccm_ipg_clk_root	clk_enable_perfmon2 (CCGR39)
	perfmon2.axi0_ACLK	ccm_main_axi_clk_root	clk_enable_perfmon2 (CCGR39)
PWM	pwm1.ipg_clk	ccm_pwm1_clk_root	clk_enable_pwm1 (CCGR40)
	pwm1.ipg_clk_high_freq	ccm_pwm1_clk_root	clk_enable_pwm1 (CCGR40)
	pwm1.ipg_clk_s	ccm_pwm1_clk_root	clk_enable_pwm1 (CCGR40)
	pwm2.ipg_clk	ccm_pwm2_clk_root	clk_enable_pwm2 (CCGR41)
	pwm2.ipg_clk_high_freq	ccm_pwm2_clk_root	clk_enable_pwm2 (CCGR41)
	pwm2.ipg_clk_s	ccm_pwm2_clk_root	clk_enable_pwm2 (CCGR41)
	pwm3.ipg_clk	ccm_pwm3_clk_root	clk_enable_pwm3 (CCGR42)
	pwm3.ipg_clk_high_freq	ccm_pwm3_clk_root	clk_enable_pwm3 (CCGR42)
	pwm3.ipg_clk_s	ccm_pwm3_clk_root	clk_enable_pwm3 (CCGR42)
	pwm4.ipg_clk	ccm_pwm4_clk_root	clk_enable_pwm4 (CCGR43)
	pwm4.ipg_clk_high_freq	ccm_pwm4_clk_root	clk_enable_pwm4 (CCGR43)
	pwm4.ipg_clk_s	ccm_pwm4_clk_root	clk_enable_pwm4 (CCGR43)
QSPI	qspi.ahb_clk	ccm_ahb_clk_root	clk_enable_qspi (CCGR47)
	qspi.ipg_clk	ccm_ipg_clk_root	clk_enable_qspi (CCGR47)
	qspi.ipg_clk_4xsfif	ccm_qspi_clk_root	clk_enable_qspi (CCGR47)
	qspi.ipg_clk_s	ccm_ipg_clk_root	clk_enable_qspi (CCGR47)
	qspi_sec.ipg_clk	ccm_ipg_clk_root	clk_enable_qspi (CCGR47)
	qspi_sec.ipg_clk_s	ccm_ipg_clk_root	clk_enable_qspi (CCGR47)
	qspi_sec.mst_hclk	ccm_ahb_clk_root	clk_enable_qspi (CCGR47)
RDC	rdc.ipg_clk_s	ccm_ipg_clk_root	clk_enable_rdc (CCGR49)
	rdc.ipg_clk	ccm_ipg_clk_root	clk_enable_rdc (CCGR49)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	rdc_mem.ipg_clk	ccm_ipg_clk_root	clk_enable_rdc (CCGR49)
ROMCP	romcp.hclk	ccm_ahb_clk_root	clk_enable_rom (CCGR50)
	romcp.hclk_reg	ccm_ipg_clk_root	clk_enable_rom (CCGR50)
	romcp_mem.rom_CLK	ccm_ahb_clk_root	clk_enable_rom (CCGR50)
	romcp_sec.mst_hclk	ccm_ahb_clk_root	clk_enable_rom (CCGR50)
SAI	sai1.ipg_clk	ccm_audio_ipg_clk_root	clk_enable_sai1 (CCGR51)
	sai1.ipg_clk_s	ccm_audio_ipg_clk_root	clk_enable_sai1 (CCGR51)
	sai1.ipg_clk_sai_mclk_1	ccm_sai1_clk_root	clk_enable_sai1 (CCGR51)
	sai1.ipt_clk_sai_bclk	ccm_sai1_clk_root	clk_enable_sai1 (CCGR51)
	sai1.ipt_clk_sai_bclk_b	ccm_sai1_clk_root	clk_enable_sai1 (CCGR51)
	sai2.ipg_clk	ccm _ipg_clk_root	clk_enable_sai2 (CCGR52)
	sai2.ipg_clk_s	ccm _ipg_clk_root	clk_enable_sai2 (CCGR52)
	sai2.ipg_clk_sai_mclk_1	ccm_sai2_clk_root	clk_enable_sai2 (CCGR52)
	sai2.ipt_clk_sai_bclk	ccm_sai2_clk_root	clk_enable_sai2 (CCGR52)
	sai2.ipt_clk_sai_bclk_b	ccm_sai2_clk_root	clk_enable_sai2 (CCGR52)
	sai3.ipg_clk	ccm _ipg_clk_root	clk_enable_sai3 (CCGR53)
	sai3.ipg_clk_s	ccm _ipg_clk_root	clk_enable_sai3 (CCGR53)
	sai3.ipg_clk_sai_mclk_1	ccm_sai3_clk_root	clk_enable_sai3 (CCGR53)
	sai3.ipt_clk_sai_bclk	ccm_sa3_clk_root	clk_enable_sai3 (CCGR53)
	sai3.ipt_clk_sai_bclk_b	ccm_sa3_clk_root	clk_enable_sai3 (CCGR53)
	sai4.ipg_clk	ccm_audio_ipg_clk_root	clk_enable_sai4 (CCGR54)
	sai4.ipg_clk_s	ccm_audio_ipg_clk_root	clk_enable_sai4 (CCGR54)
	sai4.ipg_clk_sai_mclk_1	ccm_sai4_clk_root	clk_enable_sai4 (CCGR54)
	sai4.ipt_clk_sai_bclk	ccm_sai4_clk_root	clk_enable_sai4 (CCGR54)
	sai4.ipt_clk_sai_bclk_b	ccm_sai4_clk_root	clk_enable_sai4 (CCGR54)
	sai5.ipg_clk	ccm_audio_ipg_clk_root	clk_enable_sai5 (CCGR55)
	sai5.ipg_clk_s	ccm_audio_ipg_clk_root	clk_enable_sai5 (CCGR55)
	sai5.ipg_clk_sai_mclk_1	ccm_sai5_clk_root	clk_enable_sai5 (CCGR55)
	sai5.ipt_clk_sai_bclk	ccm_sai5_clk_root	clk_enable_sai5 (CCGR55)
	sai5.ipt_clk_sai_bclk_b	ccm_sai5_clk_root	clk_enable_sai5 (CCGR55)
	sai6.ipg_clk	ccm_audio_ipg_clk_root	clk_enable_sai6 (CCGR56)
	sai6.ipg_clk_s	ccm_audio_ipg_clk_root	clk_enable_sai6 (CCGR56)
	sai6.ipg_clk_sai_mclk_1	ccm_sai6_clk_root	clk_enable_sai6 (CCGR56)
	sai6.ipt_clk_sai_bclk	ccm_sai6_clk_root	clk_enable_sai6 (CCGR56)
	sai6.ipt_clk_sai_bclk_b	ccm_sai6_clk_root	clk_enable_sai6 (CCGR56)
SCTR	sctr.ipg_clk	ccm _ipg_clk_root	clk_enable_sctr (CCGR57)
	sctr.ipg_clk_s	ccm _ipg_clk_root	clk_enable_sctr (CCGR57)
	sctr.scan_clk	anamix_osc_25m_clk	clk_enable_sctr (CCGR57)
	sctr.sys_ctr_base_clk	anamix_osc_25m_clk	clk_enable_sctr (CCGR57)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
SDMA	sdma1.ips_hostctrl_clk	ccm_ipg_clk_root	clk_enable_sdma1 (CCGR58)
	sdma1.sdma_ap_ahb_clk	ccm_ahb_clk_root	clk_enable_sdma1 (CCGR58)
	sdma1.sdma_core_clk	ccm_ipg_clk_root	clk_enable_sdma1 (CCGR58)
	sdma2.ips_hostctrl_clk	ccm_audio_ipg_clk_root	clk_enable_sdma2 (CCGR59)
	sdma2.sdma_ap_ahb_clk	ccm_audio_ahb_clk_root	clk_enable_sdma2 (CCGR59)
	sdma2.sdma_core_clk	ccm_audio_ipg_clk_root	clk_enable_sdma2 (CCGR59)
SEC	sec_wrapper.clk	ccm_ipg_clk_root	clk_enable_sec_debug (CCGR60)
SEMA42	sema1.clk	ccm_ipg_clk_root	clk_enable_sema1 (CCGR61)
	sema2.clk	ccm_ipg_clk_root	clk_enable_sema2 (CCGR62)
SIM	sim_display.cm4clk	ccm_arm_m4_clk_root	GPC controlled
	sim_display.mainclk	ccm_main_axi_clk_root	clk_enable_sim_display (CCGR63)
	sim_display.mainclk_r	ccm_main_axi_clk_root	clk_enable_sim_display (CCGR63)
	sim_enet.mainclk	ccm_enet_axi_clk_root	clk_enable_sim_enet (CCGR64)
	sim_enet.mainclk_r	ccm_enet_axi_clk_root	clk_enable_sim_enet (CCGR64)
	sim_m.mainclk	ccm_ahb_clk_root	clk_enable_sim_m (CCGR65)
	sim_m.mainclk_r	ccm_ahb_clk_root	clk_enable_sim_m (CCGR65)
	sim_m.usdhcclk	ccm_nand_usdhc_bus_clk_root	clk_enable_sim_m (CCGR65)
	sim_m.usdhcclk_r	ccm_nand_usdhc_bus_clk_root	clk_enable_sim_m (CCGR65)
	sim_main.cm4clk	ccm_arm_m4_clk_root	GPC controlled
	sim_main.enetclk	ccm_enet_axi_clk_root	clk_enable_sim_enet (CCGR64)
	sim_main.mainclk	ccm_main_axi_clk_root	clk_enable_sim_main (CCGR66)
	sim_main.mainclk_r	ccm_main_axi_clk_root	clk_enable_sim_main (CCGR66)
	sim_main.per_mclk	ccm_ahb_clk_root	clk_enable_sim_m (CCGR65)
	sim_main.per_sclk	ccm_ahb_clk_root	clk_enable_sim_s (CCGR67)
	sim_main.usdhcclk	ccm_nand_usdhc_bus_clk_root	clk_enable_sim_m (CCGR65)
	sim_main.wakeupclk	ccm_ahb_clk_root	clk_enable_sim_wakeup (CCGR68)
	sim_s.apbhdmaclk	ccm_nand_usdhc_bus_clk_root	clk_enable_rawnand (CCGR48)
	sim_s.gpv4clk	ccm_enet_axi_clk_root	clk_enable_sim_enet (CCGR64)
	sim_s.mainclk	ccm_ahb_clk_root	clk_enable_sim_s (CCGR67)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	sim_s.mainclk_r	ccm_ahb_clk_root	clk_enable_sim_s (CCGR67)
	sim_s.weimclk	ccm_ahb_clk_root	1
	sim_wakeup.mainclk	ccm_ahb_clk_root	clk_enable_sim_wakeup (CCGR68)
	sim_wakeup.mainclk_r	ccm_ahb_clk_root	clk_enable_sim_wakeup (CCGR68)
	pl301_audio.displayclk	ccm_main_axi_clk_root	clk_enable_sim_display (CCGR63)
SNVS	snvs_hs_wrapper.ipg_clk	ccm_ipg_clk_root	clk_enable_snvs (CCGR71)
	snvs_hs_wrapper.ipg_clk_s	ccm_ipg_clk_root	clk_enable_snvs (CCGR71)
	snvsmix.ipg_clk_root	ccm_ipg_clk_root	1
SPBA	spba1.ipg_clk	ccm_ipg_clk_root	clk_enable_ipmux3 (CCGR30)
	spba1.ipg_clk_s	ccm_ipg_clk_root	clk_enable_ipmux3 (CCGR30)
	spba2.ipg_clk	ccm_audio_ipg_clk_root	1
	spba2.ipg_clk_s	ccm_audio_ipg_clk_root	1
SPDIF	spdif1.extal_clk	anamix_osc_25m_clk	1
	spdif1.gclkw_t0	ccm_spdif1_clk_root	1
	spdif1.ipg_clk_s	ccm_ipg_clk_root	1
	spdif1.tx_clk	ccm_spdif1_clk_root	1
	spdif1.tx_clk1	ccm_spdif1_clk_root	1
	spdif1.tx_clk2	spdif1_ext_clk_root	1
	spdif1.tx_clk3	ccm_spdif1_clk_root	1
	spdif1.tx_clk4	ccm_spdif1_clk_root	1
	spdif1.tx_clk5	ccm_spdif1_clk_root	1
	spdif2.extal_clk	anamix_osc_25m_clk	1
	spdif2.gclkw_t0	ccm_spdif2_clk_root	1
	spdif2.ipg_clk_s	ccm_ipg_clk_root	1
	spdif2.tx_clk	ccm_spdif2_clk_root	1
	spdif2.tx_clk1	ccm_spdif2_clk_root	1
	spdif2.tx_clk2	ccm_spdif2_clk_root	1
	spdif2.tx_clk3	ccm_spdif2_clk_root	1
	spdif2.tx_clk4	ccm_spdif2_clk_root	1
	spdif2.tx_clk5	ccm_spdif2_clk_root	1
TRACE	coresight.DBGCLK	main_axi_clk_root	clk_enable_trace (CCGR72)
	coresight.traceclkin	ccm_trace_clk_root	clk_enable_trace (CCGR72)
	coresight_mem.cs_etf_clk	main_axi_clk_root	clk_enable_trace (CCGR72)
UART	uart1.ipg_clk	ccm_ipg_clk_root	clk_enable_uart1 (CCGR73)
	uart1.ipg_clk_s	ccm_ipg_clk_root	clk_enable_uart1 (CCGR73)

Table continues on the next page...

Table 5-2. System Clocks and Gating (continued)

Module	Module Clock (instance.clock)	Clock Root	Module Clock Gating Enable (CCGR)
	uart1.ipg_perclk	ccm_uart1_clk_root	clk_enable_uart1 (CCGR73)
	uart2.ipg_clk	ccm_ipg_clk_root	clk_enable_uart2 (CCGR74)
	uart2.ipg_clk_s	ccm_ipg_clk_root	clk_enable_uart2 (CCGR74)
	uart2.ipg_perclk	ccm_uart2_clk_root	clk_enable_uart2 (CCGR74)
	uart3.ipg_clk	ccm_ipg_clk_root	clk_enable_uart3 (CCGR75)
	uart3.ipg_clk_s	ccm_ipg_clk_root	clk_enable_uart3 (CCGR75)
	uart3.ipg_perclk	ccm_uart3_clk_root	clk_enable_uart3 (CCGR75)
	uart4.ipg_clk	ccm_ipg_clk_root	clk_enable_uart4 (CCGR76)
	uart4.ipg_clk_s	ccm_ipg_clk_root	clk_enable_uart4 (CCGR76)
	uart4.ipg_perclk	ccm_uart4_clk_root	clk_enable_uart4 (CCGR76)
USB	usb.clk	ccm_ipg_clk_root	clk_enable_usb (CCGR22)
USDHC	usdhc1.hclk	ccm_nand_usdhc_bus_clk_root	clk_enable_usdhc1 (CCGR81)
	usdhc1.ipg_clk	ccm_ipg_clk_root	clk_enable_usdhc1 (CCGR81)
	usdhc1.ipg_clk_perclk	ccm_usdhc1_clk_root	clk_enable_usdhc1 (CCGR81)
	usdhc1.ipg_clk_s	ccm_ipg_clk_root	clk_enable_usdhc1 (CCGR81)
	usdhc2.hclk	ccm_nand_usdhc_bus_clk_root	clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk	ccm_ipg_clk_root	clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk_perclk	ccm_usdhc2_clk_root	clk_enable_usdhc2 (CCGR82)
	usdhc2.ipg_clk_s	ccm_ipg_clk_root	clk_enable_usdhc2 (CCGR82)
WDOG	wdog1.ipg_clk	ccm_wdog_clk_root	clk_enable_wdog1 (CCGR83)
	wdog1.ipg_clk_s	ccm_wdog_clk_root	clk_enable_wdog1 (CCGR83)
	wdog2.ipg_clk	ccm_wdog_clk_root	clk_enable_wdog2 (CCGR84)
	wdog2.ipg_clk_s	ccm_wdog_clk_root	clk_enable_wdog2 (CCGR84)
	wdog3.ipg_clk	ccm_wdog_clk_root	clk_enable_wdog3 (CCGR85)
	wdog3.ipg_clk_s	ccm_wdog_clk_root	clk_enable_wdog3 (CCGR85)

5.1.5 Functional Description

The following sections describe the functional details of CCM.

5.1.5.1 Input Clocks

The table below describes the input clock sources that supply the muxes to the clock slices in the clock root generator. Please see PLL Interface for PLL programming information.

Control Register (CCM_PLL_CTRLn)	Input Clock	Frequency (MHz)	Description
12	ARM_PLL_CLK	1600	Arm PLL
13	GPU_PLL_CLK	1600	GPU PLL clock output
14	VPU_PLL_CLK	800	VPU PLL clock output
15	DRAM_PLL1_CLK	800	DDR PLL
16	SYSTEM_PLL1_CLK	800	System PLL1 output clock
17	SYSTEM_PLL1_DIV2	400	System PLL1 divided 2 clock output
18	SYSTEM_PLL1_DIV3	266	System PLL1 divided 3 clock output
19	SYSTEM_PLL1_DIV4	200	System PLL1 divided 4 clock output
20	SYSTEM_PLL1_DIV5	160	System PLL1 divided 5 clock output
21	SYSTEM_PLL1_DIV6	133	System PLL1 divided 6 clock output
22	SYSTEM_PLL1_DIV8	100	System PLL1 divided 8 clock output
23	SYSTEM_PLL1_DIV10	80	System PLL1 divided 10 clock output
24	SYSTEM_PLL1_DIV20	40	System PLL divided 20 clock output
25	SYSTEM_PLL2_CLK	1000	System PLL2 output clock
26	SYSTEM_PLL2_DIV2	500	System PLL2 divided 2 clock output
27	SYSTEM_PLL2_DIV3	333	System PLL2 divided 3 clock output
28	SYSTEM_PLL2_DIV4	250	System PLL2 divided 4 clock output
29	SYSTEM_PLL2_DIV5	200	System PLL2 divided 5 clock output
30	SYSTEM_PLL2_DIV6	166	System PLL2 divided 6 clock output
31	SYSTEM_PLL2_DIV8	125	System PLL2 divided 8 clock output
32	SYSTEM_PLL2_DIV10	100	System PLL2 divided 10 clock output
33	SYSTEM_PLL2_DIV20	50	System PLL2 divided 20 clock output
34	SYSTEM_PLL3_CLK	1000	System PLL3 output clock
35	AUDIO_PLL1_CLK	650	Audio PLL1 clock output
36	AUDIO_PLL2_CLK	650	Audio PLL2 clock output
37	VIDEO_PLL1_CLK	650	Video PLL1 clock output
38	VIDEO_PLL2_CLK	600	Video PLL2 clock output
external source	32K_REF_CLK	0.032	32K oscillator clock output
external source	25M_REF_CLK	25	25M oscillator clock output
external source	27M_REF_CLK	27	27M oscillator clock output
external source	EXT_CLK_1	133	Clock input from external IO
external source	EXT_CLK_2	133	Clock input from external IO
external source	EXT_CLK_3	133	Clock input from external IO
external source	EXT_CLK_4	133	Clock input from external IO

NOTE

CKIL_SYNC needs to be configured the same as the PLL for ipg_clk

5.1.5.2 CKIL Synchronizer

CCM provides a synchronized version of the 32K clock called CKIL Synchronizer (CKIL_SYNC). The CKIL_SYNC clock is generated and synchronized by the IPG_CLK_ROOT when IPG_CLK_ROOT is active.

When the system enters low-power mode that requires the shutdown of IPG_CLK_ROOT, the CKIL Synchonrizer needs to be bypassed and fed directly from the XTALOSC 32K source.

The control for the CKIL_SYNC bypass comes from source control. The control for CKIL_SYNC bypass needs to be configured exactly the same as the contols for the PLL output that is used to generate IPG_CLK_ROOT.

In cases where a module ipg_clk is not tied to IPG_CLK_ROOT, the bypass signal to the ipg_clk is controlled by LPCG. To avoid issues on their 32K input, the LPCG must also be shutdown at the same time as the PLL clock output for that particular ipg_clk generation.

5.1.5.3 Clock Components

The details of the CCM clock components are detailed below.

5.1.5.3.1 Clock Divider

Synchronized clock dividers can be used to perform integer division on the source clock frequency. The divider guarantees a clean clock signal on its output during the change of the divide factor. Dividers perform a 1/(N+1) divide. A glitch can cause a sync-divider to go into an unrecoverable state, therefore a clean clock signal must be provided to a sync-divider.

When updating the divider value, a read enable signal (read_en) deasserts to stop the related control registers from fetching the divider value. To change a divider factor value, the bus enters a transition state and the divide logic latches the divider factor value. After this value is latched, the read_en is reasserted. The divider will begin to update its internal divide logic. After it finishes updating, the divider sends an acknowledge signal (sw_ack) to indicate a new value is in effect.

5.1.5.3.2 Clock Switching Multiplexer

Clock switching multiplexers can guarantee a clean clock signal when switching between 2 clock sources. Both clock inputs must be active when switching. Glitches in the selected source may be transferred to the output clock while the de-selected source is blocked. Glitches that occur during switching may cause an unpredictable state, and will recover in 2~3 cycles.

Clock switching multiplexers first shutdown the current active clock, then switch. After receiving an acknowledge that the current clock stopped. The multiplexer turns on the new selected clock source. As the acknowledge of turning on the new clock source is received, the source switch finishes.

5.1.5.3.3 Clock Gate

A clock gate cell is used to gate clocks. When gated, the clock will stop at 0. A clock gate cell can accept glitches on its clock inputs. If the gate cell is off, it will block or absorb glitches. If the gate cell is on, it may pass glitches to its output. A gate cell needs 2~3 clock cycles to change states. The state during this period is either on or off and will recover in 2~3 cycles, but cannot be predicted which state it is in during this period.

5.1.5.3.4 8 to 1 Multiplexer

The 8-to-1 multiplexer is a combinational multiplexer that can switch anytime. The multiplexer output does not guarantee a clean clock signal. During switching, the output must be gated or all 8 inputs will require the same value.

5.1.5.4 Clock Slices

There are several types of clock generation slices in CCM. The slices are categorized as Core, Bus, Peripheral (IP), and DRAM.

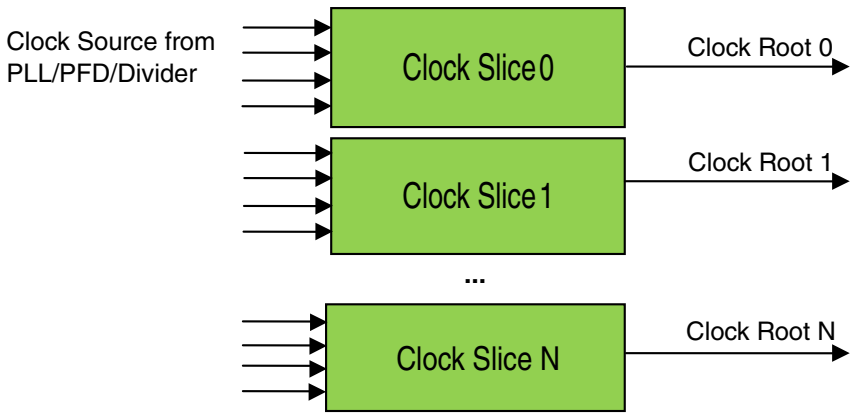


Figure 5-4. Clock Slices

5.1.5.4.1 Core clock slice

Core clock slices are designed for high speed, non-stop clock generation, typically for an Arm core. A core clock slice is comprised of a post divider and clock switching multiplexer. Each has a clock gate and a clock multiplexer inside. To run at high frequency, post divider is 3 bits, which is half the bit width of other slices. The two 8-to-1 multiplexers are not glitch-less, so switching them should only be done when their output is not routed to a clock output.

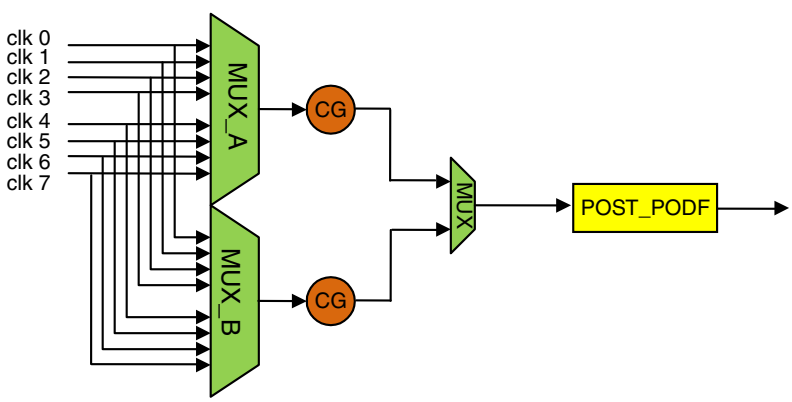


Figure 5-5. Core clock slice

5.1.5.4.2 Bus clock slice

Bus clock slices are comprised of a post divider and a clock switching multiplexer. Each has a pre-divider, clock gate and a clock multiplexer inside. The pre-divider is three bits and provides a maximum division factor of 8. The post-divider is six bits and can divide

down the clock input to 1MHz on clock output. The two 8-to-1 multiplexers are not glitch-less, so switching should only be done when their output is not routed to clock output.

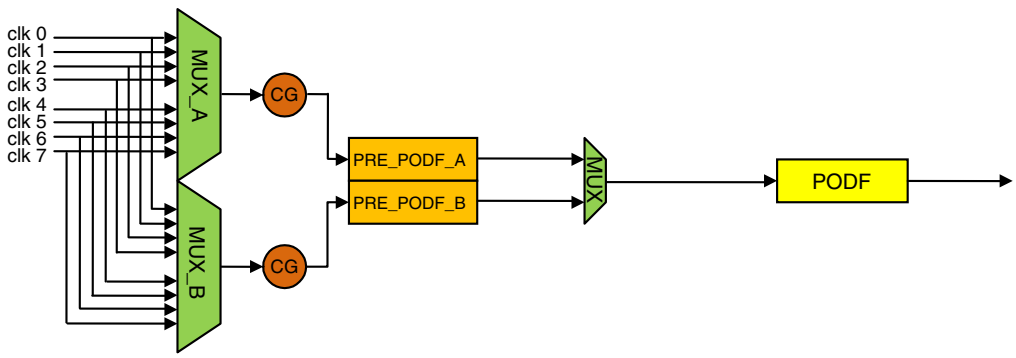


Figure 5-6. Bus clock slice

5.1.5.4.3 Peripheral clock slice

IP peripheral clock slices are comprised of a post-divider, pre-divider, clock gate and a clock multiplexer. The pre-divider is three bit and can divide down by a factor of 8. The post-divider is six bit and can divide down the input clock to 1MHz on clock output. The 8-to-1 multiplexer is not glitch-less, so switching should only be done when their output is not routed to clock output.

IP clock slices must be stopped to change the clock source.

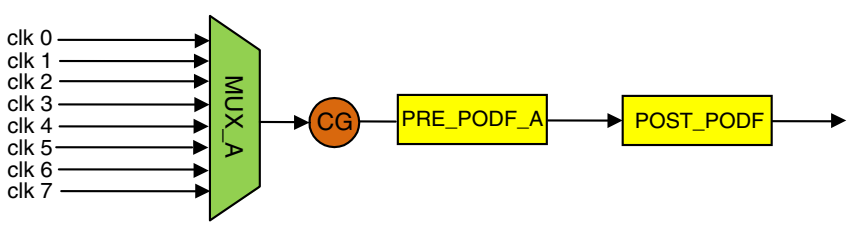


Figure 5-7. Peripheral clock slice

5.1.5.4.4 SSCG and Fractional PLLs

The SSCG and Fractional-N PLLs are detailed below. The SSCG block diagram is shown below. Please see CCM ANALOG for more information.

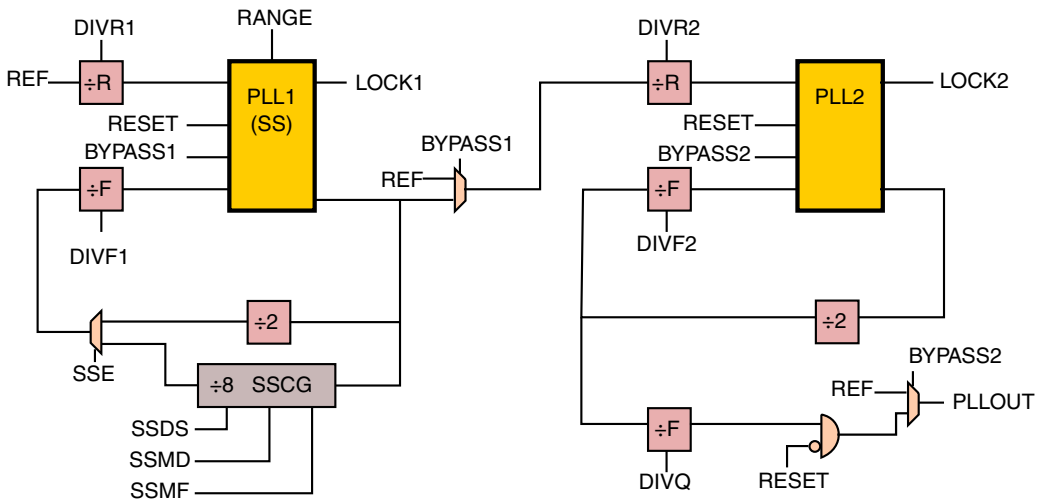


Figure 5-8. SSCG PLL Block Diagram

Formula for SSCG PLLOUT:

- SSE=0: PLLOUT = REF/DIVR1 * 2 * DIVF1/DIVR2 * DIVF2/DIVQ
- SSE=1: PLLOUT = REF/DIVR1 * 8 * DIVF1/DIVR2 * DIVF2/DIVQ

The ARM PLL, GPU PLL, VPU PLL, Audio PLL1/2, and Video PLL1 are fractional PLLs. The frequency on these can be tuned to be very accurate to meet audio and video interface requirements.

The figure below shows the Fractional PLL block diagram.

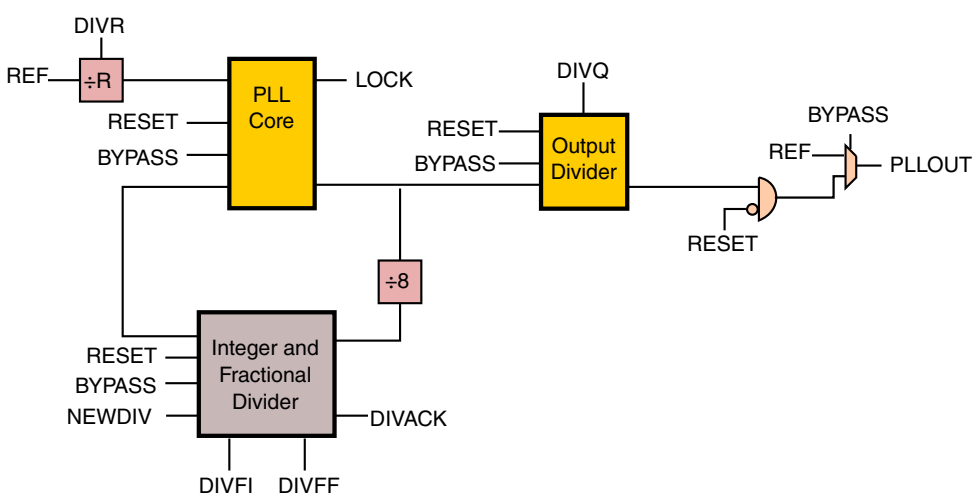


Figure 5-9. Fractional PLL Block Diagram

Formula for Fraction PLLOUT:

- PLLOUT = REF/DIVR_VAL * 8 * DIVF_VAL/DIVQ_VAL
- Where, DIVF_VAL = 1 + DIVFI + (DIVFF/2^24)

5.1.5.5 Clock gate control

CCM can perform automatic clock shutdown of on-chip peripherals according to system low power mode. Each logic domain can respectively declare its dependency level on each clock. If a clock is detected that is not dependant on any domain, it will be shutdown to save power.

Clock generation inside the CCM creates a clock root for on-chip peripherals. Before the clock root goes into peripherals via low power clock gating cells. By controlling these LPCGs, CCM can manage on-chip peripheral clocks.

Clock gate controls use active clock gating, which means the low power clock gating (LPCG) requires an active clock root. The clock generation module only performs multiplexing, gating and dividing on clock sources. Therefore, the clock root from the generation module will stop when the corresponding clock source stops.

The ENABLE bit must be set for the clock root that LPCG is actively gating.

5.1.5.6 Clock source control

CCM can perform automatic PLL shutdown of CCM_ANALOG according to system low power mode. Each logic domain can respectively declare its dependency level on each clock. If a PLL is detected not depended by any domain, it will be shutdown to save power.

PLL in ANATOP can be set be controlled by CCM via setting some CCM override bits. Also there is control on 32K IPG_CLK_ROOT synchronized version inside clock source control module.

For PLLs, we have channel on every PLL, every PFD and every divider. PLL is the source of PFDs and dividers. For any clock, its source must be left on when it is kept on. Behavior is undefined if this rule is violated.

For a shutdown clock source, if it is declared depended by writing clock source control registers, the controlling logic will turn on the source immediately, while the setting goes into a shadow register. After clock source get ready, the setting will be accepted by source control logic, and copied from shadow register to setting register.

5.1.5.7 Access control

CCM can implement its own access control based on domain to provide more precise control on shared resources. Access controls are implemented on clock root generation, clock gate control, and clock source control. Access control logic does not impact read access, but blocks unauthentic write access.

Access controls on clock root generation are independent between every clock root. A sticky authentic fail flag is set when a domain writes to a register and authentication fails. The access control logic contains a whitelist and a semaphore. By default, each clock root's access control logic is disabled after power-on reset. Software can enable access control anytime after reset.

NOTE

Once access logic is enabled, it cannot be disabled until the next power-on reset.

Table 5-3. Whitelist

Enabled	Write access will be authenticated before being performed.
Disabled	every access on protected item will be performed.

NOTE

Only domains that are on the whitelist can perform write access to this clock root when access control is enabled.

Table 5-4. Semaphore

Enabled	A domain must obtain the semaphore’s ownership before its write access can be authenticated. Only a domain on the whitelist can obtain the ownership and the ownership will last until the domain explicitly releases the ownership. Semaphore obtain will fail if it is already fetched by some other domain.
Disabled	Authentic check will check only on whitelist.

NOTE

Semaphore is intended to help software keep the clock root from unexpectedly changing.

Access control of clock gate and clock source control is performed in a simple operation. Every domain can only write on the bits for it's own setting. Any write to irrespective domain will be ignored.

5.1.5.8 System level considerations

Clock shutdown strategy

Any clock shutdown should first shutdown the LPCG, then the PLL. If the LPCG is configured not to shutdown, the clock root for the LPCG should not stop either. Violating this rule leads the system into an unpredictable state.

Core clock root frequency

If the core clock is set lower that one third of the IPG clock, SRC needs to generate a longer reset signal to match the requirement from the Arm core. This typically happens when the Arm core runs at some divided value of the XTAL 24M while IPG clock is supplied by the PLL.

DRAM clock

The DRAM PHYM clock needs a clock frequency faster that 400MHz.

USB OTG CLOCK

The USB clock may not be a reliable clock source in some applications. This clock may stop when USB cable is disconnected.

5.1.6 Programming Guide

5.1.6.1 Set, Clear, and Toggle register features

Every register of the CCM has set, clear, and toggle features.

The set feature for a given register is located at:
Base Address of the register + 0x04

The clear feature for a given register is located at:
Base Address of the regsister + 0x08

The toggle feature is located at:
Base Address of the register + 0x0c

Read from all 4 locations to get the current register value. Writing to the base register bits sets the register to the write value. Writing 1 to the set register bits sets them to 1, while writing 0 has no effect. Writing 1 to the clear register bits clears them to 0, while writing 0 has no effect. Writing 1 to the toggle register bits sets them to invert the value, while writing 0 has no effect.

5.1.6.2 PLL Interface

CCM can control PLLs inside CCM_ANALOG when entering or leaving low-power mode. Software must set the PLL override inside CCM_ANALOG before entering low-power mode after power-on reset (POR).

There are four levels of low-power modes in a logic domain:

- Not needed
- Needed in RUN
- Needed in RUN and WAIT
- Needed in RUN, WAIT, and STOP

CCM only takes action while domain status are switching between STOP (DEEP SLEEP mode is considered the same as STOP). There are 4 domains that can be assigned. Any CPU platform can be assigned to any domain by RDC. If a domain is empty, the domain is considered as STOP.

Each domain can declare its dependency to CCM. The use of any clock, without declaring it in its own domain, is not permitted. A domain declares its dependency on a clock by writing the dependency level. Settings against behavior in low-power mode are as follows:

Table 5-5. Domain Dependency

Domain Level	Run	Wait	Stop / Deep sleep
0			
1	Required		
2	Required	Required	
3	Required	Required	Required

Table 5-6. CCGR Program Interface

CCGR		Domain3				Domain2				Domain1				Domain0			
	31-16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Level1	Level0			Level1	Level0			Level1	Level0			Level1	Level0

Each domain can change only bits assigned to control access. Any irrelevant write to it will be ignored. For example, Domain 0 can only write to bits [1:0]. Any bits written to, other than bits [1:0], will be ignored. Other domains can read all of the other domain settings. The default value for domain 0 is 2, and will enter STOP mode after shutting down. When the default value of the other domain setting is 0, it will not be required.

When setting clock source, the settings will not take effect immediately. The setting will enter the shadow register first. If a PLL shutdown or new setting enters the shadow register to declare dependency on the PLL, the PLL will turn on immediately. When the PLL is ready, the setting in shadow register will be updated to the new setting. During this period, the pending bit will be set and cleared. Then CCM will send the PLL control signal as a shadow register and inform GPC the PLL status according to the setting register. In other cases, the setting will be updated from the shadow register immediately. Clock sources have dependency on each other.

NOTE

Do not shutdown the parent clock when the required child clock is active. Attempting to do so will lead to unpredicable and unrecoverable behavior. It is recommended to shutdown the parent clock and child clock together.

5.1.6.3 CCGR Interface

Before a clock root goes to on–chip peripherals, the clock root is distributed through low power clock gates (LPCG). These LPCG are implemented to automatically perform clock shutdown when a domain enters and leaves a low-power state.

There are four levels of low-power modes in a logic domain:

- Not needed
- Needed in RUN
- Needed in RUN and WAIT
- Needed in RUN, WAIT, and STOP

CCM only takes action while domain status are switching between STOP (DEEP SLEEP mode is considered the same as STOP). There are 4 domains that can be assigned. Any CPU platform can be assigned to any domain by RDC. If a domain is empty, the domain is considered as STOP.

Each domain can declare its dependency to CCM. The use of any clock, without declaring it in its own domain, is not permitted. A domain declares its dependency on a clock by writing the dependency level. Settings against behavior in low-power mode are as follows:

Table 5-7. Domain Dependency

Domain Level	RUN	WAIT	STOP/ DEEP SLEEP
0			
1	Required		
2	Required	Required	

Table continues on the next page...

Table 5-7. Domain Dependency (continued)

3	Required	Required	Required

Table 5-8. CCGR Program Interface

CCGR		Domain3				Domain2				Domain1				Domain0			
	31-16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
				Level1	Level0			Level1	Level0			Level1	Level0			Level1	Level0

Each domain can change only bits assigned to control access. Any irrelevant write to it will be ignored. For example, Domain 0 can only write to bits [1:0]. Any bits written to, other than bits [1:0], will be ignored. Other domains can read all of the other domain settings. The default value for domain 0 is 2, and will enter STOP mode after shutting down. When the default value of the other domain setting is 0, it will not be required.

The table below lists the CCM Clock Gating Register (CCGR) and associated offset for each LPCG enable.

NOTE

Not all CCGRs are mapped.

Table 5-9. CCGR Mapping Table

Gating Register	LPCG Enable	Offset
CCM_CCGR0	DVFS	0x4000
CCM_CCGR1	Anamix	0x4010
CCM_CCGR2	CPU	0x4020
CCM_CCGR3	CSU	0x4030
CCM_CCGR4	Debug	0x4040
CCM_CCGR5	DDR1	0x4050
CCM_CCGR6	Reserved	0x4060
CCM_CCGR7	ECSPI1	0x4070
CCM_CCGR8	ECSPI2	0x4080
CCM_CCGR9	ECSPI3	0x4090
CCM_CCGR10	ENET1	0x40A0
CCM_CCGR11	GPIO1	0x40B0
CCM_CCGR12	GPIO2	0x40C0
CCM_CCGR13	GPIO3	0x40D0
CCM_CCGR14	GPIO4	0x40E0
CCM_CCGR15	GPIO5	0x40F0
CCM_CCGR16	GPT1	0x4100

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR17	GPT2	0x4110
CCM_CCGR18	GPT3	0x4120
CCM_CCGR19	GPT4	0x4130
CCM_CCGR20	GPT5	0x4140
CCM_CCGR21	GPT6	0x4150
CCM_CCGR22	HS	0x4160
CCM_CCGR23	I2C1	0x4170
CCM_CCGR24	I2C2	0x4180
CCM_CCGR25	I2C3	0x4190
CCM_CCGR26	I2C4	0x41A0
CCM_CCGR27	IOMUX	0x41B0
CCM_CCGR28	IOMUX1	0x41C0
CCM_CCGR29	IOMUX2	0x41D0
CCM_CCGR30	IOMUX3	0x41E0
CCM_CCGR31	IOMUX4	0x41F0
CCM_CCGR32	M4	0x4200
CCM_CCGR33	MU	0x4210
CCM_CCGR34	OCOTP	0x4220
CCM_CCGR35	OCRAM	0x4230
CCM_CCGR36	OCRAM_s	0x4240
CCM_CCGR37	PCIE	0x4250
CCM_CCGR38	PERFMON1	0x4260
CCM_CCGR39	PERFMON2	0x4270
CCM_CCGR40	PWM1	0x4280
CCM_CCGR41	PWM2	0x4290
CCM_CCGR42	PWM3	0x42A0
CCM_CCGR43	PWM4	0x42B0
CCM_CCGR44	QoS	0x42C0
CCM_CCGR45	Dismix	0x42D0
CCM_CCGR46	Megamix	0x42E0
CCM_CCGR47	QSPI	0x42F0
CCM_CCGR48	NAND (APBHDMA, GPMI, BCH)	0x4300
CCM_CCGR49	RDC	0x4310
CCM_CCGR50	ROM	0x4320
CCM_CCGR51	SAI1	0x4330
CCM_CCGR52	SAI2	0x4340
CCM_CCGR53	SAI3	0x4350
CCM_CCGR54	SAI4	0x4360
CCM_CCGR55	SAI5	0x4370

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR56	SAI6	0x4380
CCM_CCGR57	SCTR	0x4390
CCM_CCGR58	SDMA1	0x43A0
CCM_CCGR59	SDMA2	0x43B0
CCM_CCGR60	SEC_DEBUG	0x43C0
CCM_CCGR61	SEMA1	0x43D0
CCM_CCGR62	SEMA2	0x43E0
CCM_CCGR63	SIM_display	0x43F0
CCM_CCGR64	SIM_ENET	0x4400
CCM_CCGR65	SIM_m	0x4410
CCM_CCGR66	SIM_main	0x4420
CCM_CCGR67	SIM_s	0x4430
CCM_CCGR68	SIM_wakeup	0x4440
CCM_CCGR69	SIM_USB	0x4450
CCM_CCGR70	SIM_VPU	0x4460
CCM_CCGR71	SNVS	0x4470
CCM_CCGR72	Trace	0x4480
CCM_CCGR73	UART1	0x4490
CCM_CCGR74	UART2	0x44A0
CCM_CCGR75	UART3	0x44B0
CCM_CCGR76	UART4	0x44C0
CCM_CCGR77	USB_CTRL1	0x44D0
CCM_CCGR78	USB_CTRL2	0x44E0
CCM_CCGR79	USB_PHY1	0x44F0
CCM_CCGR80	USB_PHY2	0x4500
CCM_CCGR81	USDHC1	0x4510
CCM_CCGR82	USDHC2	0x4520
CCM_CCGR83	WDOG1	0x4530
CCM_CCGR84	WDOG2	0x4540
CCM_CCGR85	WDOG3	0x4550
CCM_CCGR86	VA53	0x4560
CCM_CCGR87	GPU	0x4570
CCM_CCGR88	HEVC	0x4580
CCM_CCGR89	AVC	0x4590
CCM_CCGR90	VP9	0x45A0
CCM_CCGR91	HEVC_inter	0x45B0
CCM_CCGR92	GIC	0x45C0
CCM_CCGR93	Display	0x45D0
CCM_CCGR94	HDMI	0x45E0

Table continues on the next page...

Table 5-9. CCGR Mapping Table (continued)

Gating Register	LPCG Enable	Offset
CCM_CCGR95	HDMI_phy	0x45F0
CCM_CCGR96	XTALOSC	0x4600
CCM_CCGR97	PLL	0x4610
CCM_CCGR98	TEMPSENSOR	0x4620
CCM_CCGR99	VPU_DEC	0x4630
CCM_CCGR100	PCIE2	0x4640
CCM_CCGR101	MIPI_CSI1	0x4650
CCM_CCGR102	MIPI_CSI2	0x4660

5.1.6.4 Target Interface

The Target Interface is optimized to simplify software operation. Using this interface, all clock roots are in the same program model with the same register bit field mapping. The software does not handle the details of the clock slice and clock slice types. Software writes the desired settings to the register, and the internal hardware logic generates a required sequence to achieve the desired settings.

The Target Interface requires the software to provide whether a clock is active, the clock source number to be selected, pre-divide value, and post-divide value. If a clock slice does not support a setting, that setting is simply ignored, and will not effect the supported fields.

Freq = (clock source freq)/(pre_div+1)/(post_div+1)

The internal logic sequence of the Target Interface guarantees a clean clock on output without frequency overshoot. A requirement of the Target Interface's software is that the target clock source is active.

The Target Interface sequence begins by opening all clocks, applying highest divider value, switching to the new clock source, then decreasing divider value to the target frequency. If Shutdown is requested, it will be performed last.

The clock output is always active when using the Target Interface. For intermediate frequency requests, the Target Interface choses the lowest frequency source to avoid frequency overshoot on the Peripheral clock slices. For Core and Bus clock slices, the clock switching multiplexer is used to guarantee smooth clock switching.

A write operation on a target interface completes once the output clock is running at the desired setting. Software polling is not necessary to determine clock stability.

STEP	STATE	OPERATION
0	SMART_IDLE	Idle state, no write operation is pending
1	SMART_WAIT_READY	State occurs when a write access is received, wait for every field to be ready
2	SMART_APPLY_GATE1	Open all branches, all gates inside clock slices
3	SMART_WAIT_GATE1	Wait for gate applied
4	SMART_APPLY_PODF1	apply post divider and post divider for if new value generate slower clock
5	SMART_WAIT_PODF1	Wait for divider accept new value
6	SMART_APPLY_GATE2	shutdown spare branch if exist, else shutdown working branch
7	SMART_WAIT_GATE2	Wait for shutdown operation complete
8	SMART_APPLY_MUX	Change multiplexer to new source on spare branch if exist, else switch working one
9	SMART_APPLY_GATE3	open gates on all branches
10	SMART_WAIT_GATE3	Wait for gates opened
11	SMART_APPLY_SWITCH	Switch clock switching multiplexer if there is one
12	SMART_WAIT_SWITCH	Wait for clock switching multiplexer switch
13	SMART_APPLY_PODF2	apply post divider and post divider for if new value generate faster clock
14	SMART_WAIT_PODF2	Wait for divider accept new value
15	SMART_APPLY_GATE4	apply clock gate setting, shutdown spare one if there is
16	SMART_WAIT_GATE4	Wait for gate applied
17	SMART_APPLY_AUTO	apply auto and auto divider
18	SMART_DONE	Finish, wait for bus operation complete

5.1.6.5 Normal Interface

Normal interface provide more controllable thing that target interface. And also provide protections against dangerous operation.

Normal interface provides safe sequences to handle each clock component, divider, gate, multiplexer. But it is software that needs to care the order and relationship between updating components.

Writing to this interface will complete immediately, and internal logic will continue try to apply written values to clock components. A busy flag will be assert during applying.

Field access rule:

1. Only one field can be modified a time
2. No field can be modified when any field pending
3. Not violate change condition in following table

FIELD	CHANGE CONDITION	FINISH CONDITION
Auto		immediate
Auto-divider		immediate
Bypass	Gatea active and gateb active	Bypass switch complete
Post-divider		New divider value applied
Gate B	Bypass disable	New gate value active
Pre-divider B	Bypass disable and gateb not gated	New divider value applied
MUX B	Bypass disable and gateb gating	immediate
Gate A	Bypass	New gate value active
Pre-divider A	Bypass and gatea not gated	New divider value applied
MUX A	Bypass and gatea gating	immediate

- Error will be reported if access rules violated.
- Unsafe or ambiguous access will be ignored.

If a write access as blocked by normal interface, the write operation will be ignored. And a sticky bit “violate” will be set. The bit will last until software clears it explicitly. The violate bit is 4 bits inside CCM, each for a logic domain. Each domain can read and clear the bit for itself, the bits for other domain is neither visible nor clearable.

5.1.7 CCM Memory Map/Register Definition

The Memory Map below represents the full array for CCM.

NOTE

Not all mapped Clock Slices and CCGRs are tied to functional components.

Please see the following for the functional mapping tables and information:

- CCM_PLL_CTRL - Input Clocks
- CCM_TARGET_ROOT - Clock Root Selects
- CCM_CCGR - CCGR Interface

CCM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0000	General Purpose Register (CCM_GPR0)	32	R/W	0000_0000h	5.1.7.1/501
3038_0004	General Purpose Register (CCM_GPR0_SET)	32	R/W	0000_0000h	5.1.7.1/501
3038_0008	General Purpose Register (CCM_GPR0_CLR)	32	R/W	0000_0000h	5.1.7.1/501
3038_000C	General Purpose Register (CCM_GPR0_TOG)	32	R/W	0000_0000h	5.1.7.1/501

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0800	CCM PLL Control Register (CCM_PLL_CTRL0)	32	R/W	0000_0002h	5.1.7.2/502
3038_0804	CCM PLL Control Register (CCM_PLL_CTRL0_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0808	CCM PLL Control Register (CCM_PLL_CTRL0_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_080C	CCM PLL Control Register (CCM_PLL_CTRL0_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0810	CCM PLL Control Register (CCM_PLL_CTRL1)	32	R/W	0000_0002h	5.1.7.2/502
3038_0814	CCM PLL Control Register (CCM_PLL_CTRL1_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0818	CCM PLL Control Register (CCM_PLL_CTRL1_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_081C	CCM PLL Control Register (CCM_PLL_CTRL1_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0820	CCM PLL Control Register (CCM_PLL_CTRL2)	32	R/W	0000_0002h	5.1.7.2/502
3038_0824	CCM PLL Control Register (CCM_PLL_CTRL2_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0828	CCM PLL Control Register (CCM_PLL_CTRL2_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_082C	CCM PLL Control Register (CCM_PLL_CTRL2_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0830	CCM PLL Control Register (CCM_PLL_CTRL3)	32	R/W	0000_0002h	5.1.7.2/502
3038_0834	CCM PLL Control Register (CCM_PLL_CTRL3_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0838	CCM PLL Control Register (CCM_PLL_CTRL3_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_083C	CCM PLL Control Register (CCM_PLL_CTRL3_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0840	CCM PLL Control Register (CCM_PLL_CTRL4)	32	R/W	0000_0002h	5.1.7.2/502
3038_0844	CCM PLL Control Register (CCM_PLL_CTRL4_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0848	CCM PLL Control Register (CCM_PLL_CTRL4_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_084C	CCM PLL Control Register (CCM_PLL_CTRL4_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0850	CCM PLL Control Register (CCM_PLL_CTRL5)	32	R/W	0000_0002h	5.1.7.2/502
3038_0854	CCM PLL Control Register (CCM_PLL_CTRL5_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0858	CCM PLL Control Register (CCM_PLL_CTRL5_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_085C	CCM PLL Control Register (CCM_PLL_CTRL5_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0860	CCM PLL Control Register (CCM_PLL_CTRL6)	32	R/W	0000_0002h	5.1.7.2/502
3038_0864	CCM PLL Control Register (CCM_PLL_CTRL6_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0868	CCM PLL Control Register (CCM_PLL_CTRL6_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_086C	CCM PLL Control Register (CCM_PLL_CTRL6_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0870	CCM PLL Control Register (CCM_PLL_CTRL7)	32	R/W	0000_0002h	5.1.7.2/502
3038_0874	CCM PLL Control Register (CCM_PLL_CTRL7_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0878	CCM PLL Control Register (CCM_PLL_CTRL7_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_087C	CCM PLL Control Register (CCM_PLL_CTRL7_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0880	CCM PLL Control Register (CCM_PLL_CTRL8)	32	R/W	0000_0002h	5.1.7.2/502
3038_0884	CCM PLL Control Register (CCM_PLL_CTRL8_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0888	CCM PLL Control Register (CCM_PLL_CTRL8_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_088C	CCM PLL Control Register (CCM_PLL_CTRL8_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0890	CCM PLL Control Register (CCM_PLL_CTRL9)	32	R/W	0000_0002h	5.1.7.2/502
3038_0894	CCM PLL Control Register (CCM_PLL_CTRL9_SET)	32	R/W	0000_0002h	5.1.7.3/504

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0898	CCM PLL Control Register (CCM_PLL_CTRL9_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_089C	CCM PLL Control Register (CCM_PLL_CTRL9_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08A0	CCM PLL Control Register (CCM_PLL_CTRL10)	32	R/W	0000_0002h	5.1.7.2/502
3038_08A4	CCM PLL Control Register (CCM_PLL_CTRL10_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08A8	CCM PLL Control Register (CCM_PLL_CTRL10_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08AC	CCM PLL Control Register (CCM_PLL_CTRL10_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08B0	CCM PLL Control Register (CCM_PLL_CTRL11)	32	R/W	0000_0002h	5.1.7.2/502
3038_08B4	CCM PLL Control Register (CCM_PLL_CTRL11_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08B8	CCM PLL Control Register (CCM_PLL_CTRL11_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08BC	CCM PLL Control Register (CCM_PLL_CTRL11_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08C0	CCM PLL Control Register (CCM_PLL_CTRL12)	32	R/W	0000_0002h	5.1.7.2/502
3038_08C4	CCM PLL Control Register (CCM_PLL_CTRL12_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08C8	CCM PLL Control Register (CCM_PLL_CTRL12_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08CC	CCM PLL Control Register (CCM_PLL_CTRL12_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08D0	CCM PLL Control Register (CCM_PLL_CTRL13)	32	R/W	0000_0002h	5.1.7.2/502
3038_08D4	CCM PLL Control Register (CCM_PLL_CTRL13_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08D8	CCM PLL Control Register (CCM_PLL_CTRL13_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08DC	CCM PLL Control Register (CCM_PLL_CTRL13_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08E0	CCM PLL Control Register (CCM_PLL_CTRL14)	32	R/W	0000_0002h	5.1.7.2/502
3038_08E4	CCM PLL Control Register (CCM_PLL_CTRL14_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08E8	CCM PLL Control Register (CCM_PLL_CTRL14_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08EC	CCM PLL Control Register (CCM_PLL_CTRL14_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_08F0	CCM PLL Control Register (CCM_PLL_CTRL15)	32	R/W	0000_0002h	5.1.7.2/502
3038_08F4	CCM PLL Control Register (CCM_PLL_CTRL15_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_08F8	CCM PLL Control Register (CCM_PLL_CTRL15_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_08FC	CCM PLL Control Register (CCM_PLL_CTRL15_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0900	CCM PLL Control Register (CCM_PLL_CTRL16)	32	R/W	0000_0002h	5.1.7.2/502
3038_0904	CCM PLL Control Register (CCM_PLL_CTRL16_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0908	CCM PLL Control Register (CCM_PLL_CTRL16_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_090C	CCM PLL Control Register (CCM_PLL_CTRL16_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0910	CCM PLL Control Register (CCM_PLL_CTRL17)	32	R/W	0000_0002h	5.1.7.2/502
3038_0914	CCM PLL Control Register (CCM_PLL_CTRL17_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0918	CCM PLL Control Register (CCM_PLL_CTRL17_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_091C	CCM PLL Control Register (CCM_PLL_CTRL17_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0920	CCM PLL Control Register (CCM_PLL_CTRL18)	32	R/W	0000_0002h	5.1.7.2/502
3038_0924	CCM PLL Control Register (CCM_PLL_CTRL18_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0928	CCM PLL Control Register (CCM_PLL_CTRL18_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_092C	CCM PLL Control Register (CCM_PLL_CTRL18_TOG)	32	R/W	0000_0002h	5.1.7.5/508

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0930	CCM PLL Control Register (CCM_PLL_CTRL19)	32	R/W	0000_0002h	5.1.7.2/502
3038_0934	CCM PLL Control Register (CCM_PLL_CTRL19_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0938	CCM PLL Control Register (CCM_PLL_CTRL19_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_093C	CCM PLL Control Register (CCM_PLL_CTRL19_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0940	CCM PLL Control Register (CCM_PLL_CTRL20)	32	R/W	0000_0002h	5.1.7.2/502
3038_0944	CCM PLL Control Register (CCM_PLL_CTRL20_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0948	CCM PLL Control Register (CCM_PLL_CTRL20_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_094C	CCM PLL Control Register (CCM_PLL_CTRL20_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0950	CCM PLL Control Register (CCM_PLL_CTRL21)	32	R/W	0000_0002h	5.1.7.2/502
3038_0954	CCM PLL Control Register (CCM_PLL_CTRL21_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0958	CCM PLL Control Register (CCM_PLL_CTRL21_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_095C	CCM PLL Control Register (CCM_PLL_CTRL21_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0960	CCM PLL Control Register (CCM_PLL_CTRL22)	32	R/W	0000_0002h	5.1.7.2/502
3038_0964	CCM PLL Control Register (CCM_PLL_CTRL22_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0968	CCM PLL Control Register (CCM_PLL_CTRL22_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_096C	CCM PLL Control Register (CCM_PLL_CTRL22_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0970	CCM PLL Control Register (CCM_PLL_CTRL23)	32	R/W	0000_0002h	5.1.7.2/502
3038_0974	CCM PLL Control Register (CCM_PLL_CTRL23_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0978	CCM PLL Control Register (CCM_PLL_CTRL23_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_097C	CCM PLL Control Register (CCM_PLL_CTRL23_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0980	CCM PLL Control Register (CCM_PLL_CTRL24)	32	R/W	0000_0002h	5.1.7.2/502
3038_0984	CCM PLL Control Register (CCM_PLL_CTRL24_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0988	CCM PLL Control Register (CCM_PLL_CTRL24_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_098C	CCM PLL Control Register (CCM_PLL_CTRL24_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0990	CCM PLL Control Register (CCM_PLL_CTRL25)	32	R/W	0000_0002h	5.1.7.2/502
3038_0994	CCM PLL Control Register (CCM_PLL_CTRL25_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0998	CCM PLL Control Register (CCM_PLL_CTRL25_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_099C	CCM PLL Control Register (CCM_PLL_CTRL25_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09A0	CCM PLL Control Register (CCM_PLL_CTRL26)	32	R/W	0000_0002h	5.1.7.2/502
3038_09A4	CCM PLL Control Register (CCM_PLL_CTRL26_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_09A8	CCM PLL Control Register (CCM_PLL_CTRL26_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09AC	CCM PLL Control Register (CCM_PLL_CTRL26_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09B0	CCM PLL Control Register (CCM_PLL_CTRL27)	32	R/W	0000_0002h	5.1.7.2/502
3038_09B4	CCM PLL Control Register (CCM_PLL_CTRL27_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_09B8	CCM PLL Control Register (CCM_PLL_CTRL27_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09BC	CCM PLL Control Register (CCM_PLL_CTRL27_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09C0	CCM PLL Control Register (CCM_PLL_CTRL28)	32	R/W	0000_0002h	5.1.7.2/502
3038_09C4	CCM PLL Control Register (CCM_PLL_CTRL28_SET)	32	R/W	0000_0002h	5.1.7.3/504

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_09C8	CCM PLL Control Register (CCM_PLL_CTRL28_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09CC	CCM PLL Control Register (CCM_PLL_CTRL28_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09D0	CCM PLL Control Register (CCM_PLL_CTRL29)	32	R/W	0000_0002h	5.1.7.2/502
3038_09D4	CCM PLL Control Register (CCM_PLL_CTRL29_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_09D8	CCM PLL Control Register (CCM_PLL_CTRL29_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09DC	CCM PLL Control Register (CCM_PLL_CTRL29_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09E0	CCM PLL Control Register (CCM_PLL_CTRL30)	32	R/W	0000_0002h	5.1.7.2/502
3038_09E4	CCM PLL Control Register (CCM_PLL_CTRL30_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_09E8	CCM PLL Control Register (CCM_PLL_CTRL30_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09EC	CCM PLL Control Register (CCM_PLL_CTRL30_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_09F0	CCM PLL Control Register (CCM_PLL_CTRL31)	32	R/W	0000_0002h	5.1.7.2/502
3038_09F4	CCM PLL Control Register (CCM_PLL_CTRL31_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_09F8	CCM PLL Control Register (CCM_PLL_CTRL31_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_09FC	CCM PLL Control Register (CCM_PLL_CTRL31_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A00	CCM PLL Control Register (CCM_PLL_CTRL32)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A04	CCM PLL Control Register (CCM_PLL_CTRL32_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A08	CCM PLL Control Register (CCM_PLL_CTRL32_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A0C	CCM PLL Control Register (CCM_PLL_CTRL32_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A10	CCM PLL Control Register (CCM_PLL_CTRL33)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A14	CCM PLL Control Register (CCM_PLL_CTRL33_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A18	CCM PLL Control Register (CCM_PLL_CTRL33_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A1C	CCM PLL Control Register (CCM_PLL_CTRL33_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A20	CCM PLL Control Register (CCM_PLL_CTRL34)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A24	CCM PLL Control Register (CCM_PLL_CTRL34_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A28	CCM PLL Control Register (CCM_PLL_CTRL34_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A2C	CCM PLL Control Register (CCM_PLL_CTRL34_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A30	CCM PLL Control Register (CCM_PLL_CTRL35)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A34	CCM PLL Control Register (CCM_PLL_CTRL35_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A38	CCM PLL Control Register (CCM_PLL_CTRL35_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A3C	CCM PLL Control Register (CCM_PLL_CTRL35_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A40	CCM PLL Control Register (CCM_PLL_CTRL36)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A44	CCM PLL Control Register (CCM_PLL_CTRL36_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A48	CCM PLL Control Register (CCM_PLL_CTRL36_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A4C	CCM PLL Control Register (CCM_PLL_CTRL36_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_0A50	CCM PLL Control Register (CCM_PLL_CTRL37)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A54	CCM PLL Control Register (CCM_PLL_CTRL37_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A58	CCM PLL Control Register (CCM_PLL_CTRL37_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A5C	CCM PLL Control Register (CCM_PLL_CTRL37_TOG)	32	R/W	0000_0002h	5.1.7.5/508

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_0A60	CCM PLL Control Register (CCM_PLL_CTRL38)	32	R/W	0000_0002h	5.1.7.2/502
3038_0A64	CCM PLL Control Register (CCM_PLL_CTRL38_SET)	32	R/W	0000_0002h	5.1.7.3/504
3038_0A68	CCM PLL Control Register (CCM_PLL_CTRL38_CLR)	32	R/W	0000_0002h	5.1.7.4/506
3038_0A6C	CCM PLL Control Register (CCM_PLL_CTRL38_TOG)	32	R/W	0000_0002h	5.1.7.5/508
3038_4000	CCM Clock Gating Register (CCM_CCGR0)	32	R/W	0000_0002h	5.1.7.6/510
3038_4004	CCM Clock Gating Register (CCM_CCGR0_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4008	CCM Clock Gating Register (CCM_CCGR0_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_400C	CCM Clock Gating Register (CCM_CCGR0_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4010	CCM Clock Gating Register (CCM_CCGR1)	32	R/W	0000_0002h	5.1.7.6/510
3038_4014	CCM Clock Gating Register (CCM_CCGR1_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4018	CCM Clock Gating Register (CCM_CCGR1_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_401C	CCM Clock Gating Register (CCM_CCGR1_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4020	CCM Clock Gating Register (CCM_CCGR2)	32	R/W	0000_0002h	5.1.7.6/510
3038_4024	CCM Clock Gating Register (CCM_CCGR2_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4028	CCM Clock Gating Register (CCM_CCGR2_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_402C	CCM Clock Gating Register (CCM_CCGR2_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4030	CCM Clock Gating Register (CCM_CCGR3)	32	R/W	0000_0002h	5.1.7.6/510
3038_4034	CCM Clock Gating Register (CCM_CCGR3_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4038	CCM Clock Gating Register (CCM_CCGR3_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_403C	CCM Clock Gating Register (CCM_CCGR3_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4040	CCM Clock Gating Register (CCM_CCGR4)	32	R/W	0000_0002h	5.1.7.6/510
3038_4044	CCM Clock Gating Register (CCM_CCGR4_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4048	CCM Clock Gating Register (CCM_CCGR4_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_404C	CCM Clock Gating Register (CCM_CCGR4_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4050	CCM Clock Gating Register (CCM_CCGR5)	32	R/W	0000_0002h	5.1.7.6/510
3038_4054	CCM Clock Gating Register (CCM_CCGR5_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4058	CCM Clock Gating Register (CCM_CCGR5_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_405C	CCM Clock Gating Register (CCM_CCGR5_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4060	CCM Clock Gating Register (CCM_CCGR6)	32	R/W	0000_0002h	5.1.7.6/510
3038_4064	CCM Clock Gating Register (CCM_CCGR6_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4068	CCM Clock Gating Register (CCM_CCGR6_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_406C	CCM Clock Gating Register (CCM_CCGR6_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4070	CCM Clock Gating Register (CCM_CCGR7)	32	R/W	0000_0002h	5.1.7.6/510
3038_4074	CCM Clock Gating Register (CCM_CCGR7_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4078	CCM Clock Gating Register (CCM_CCGR7_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_407C	CCM Clock Gating Register (CCM_CCGR7_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4080	CCM Clock Gating Register (CCM_CCGR8)	32	R/W	0000_0002h	5.1.7.6/510
3038_4084	CCM Clock Gating Register (CCM_CCGR8_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4088	CCM Clock Gating Register (CCM_CCGR8_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_408C	CCM Clock Gating Register (CCM_CCGR8_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4090	CCM Clock Gating Register (CCM_CCGR9)	32	R/W	0000_0002h	5.1.7.6/510
3038_4094	CCM Clock Gating Register (CCM_CCGR9_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4098	CCM Clock Gating Register (CCM_CCGR9_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_409C	CCM Clock Gating Register (CCM_CCGR9_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40A0	CCM Clock Gating Register (CCM_CCGR10)	32	R/W	0000_0002h	5.1.7.6/510
3038_40A4	CCM Clock Gating Register (CCM_CCGR10_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40A8	CCM Clock Gating Register (CCM_CCGR10_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40AC	CCM Clock Gating Register (CCM_CCGR10_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40B0	CCM Clock Gating Register (CCM_CCGR11)	32	R/W	0000_0002h	5.1.7.6/510
3038_40B4	CCM Clock Gating Register (CCM_CCGR11_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40B8	CCM Clock Gating Register (CCM_CCGR11_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40BC	CCM Clock Gating Register (CCM_CCGR11_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40C0	CCM Clock Gating Register (CCM_CCGR12)	32	R/W	0000_0002h	5.1.7.6/510
3038_40C4	CCM Clock Gating Register (CCM_CCGR12_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40C8	CCM Clock Gating Register (CCM_CCGR12_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40CC	CCM Clock Gating Register (CCM_CCGR12_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40D0	CCM Clock Gating Register (CCM_CCGR13)	32	R/W	0000_0002h	5.1.7.6/510
3038_40D4	CCM Clock Gating Register (CCM_CCGR13_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40D8	CCM Clock Gating Register (CCM_CCGR13_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40DC	CCM Clock Gating Register (CCM_CCGR13_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40E0	CCM Clock Gating Register (CCM_CCGR14)	32	R/W	0000_0002h	5.1.7.6/510
3038_40E4	CCM Clock Gating Register (CCM_CCGR14_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40E8	CCM Clock Gating Register (CCM_CCGR14_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40EC	CCM Clock Gating Register (CCM_CCGR14_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_40F0	CCM Clock Gating Register (CCM_CCGR15)	32	R/W	0000_0002h	5.1.7.6/510
3038_40F4	CCM Clock Gating Register (CCM_CCGR15_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_40F8	CCM Clock Gating Register (CCM_CCGR15_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_40FC	CCM Clock Gating Register (CCM_CCGR15_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4100	CCM Clock Gating Register (CCM_CCGR16)	32	R/W	0000_0002h	5.1.7.6/510
3038_4104	CCM Clock Gating Register (CCM_CCGR16_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4108	CCM Clock Gating Register (CCM_CCGR16_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_410C	CCM Clock Gating Register (CCM_CCGR16_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4110	CCM Clock Gating Register (CCM_CCGR17)	32	R/W	0000_0002h	5.1.7.6/510
3038_4114	CCM Clock Gating Register (CCM_CCGR17_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4118	CCM Clock Gating Register (CCM_CCGR17_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_411C	CCM Clock Gating Register (CCM_CCGR17_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4120	CCM Clock Gating Register (CCM_CCGR18)	32	R/W	0000_0002h	5.1.7.6/510
3038_4124	CCM Clock Gating Register (CCM_CCGR18_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4128	CCM Clock Gating Register (CCM_CCGR18_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_412C	CCM Clock Gating Register (CCM_CCGR18_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4130	CCM Clock Gating Register (CCM_CCGR19)	32	R/W	0000_0002h	5.1.7.6/510
3038_4134	CCM Clock Gating Register (CCM_CCGR19_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4138	CCM Clock Gating Register (CCM_CCGR19_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_413C	CCM Clock Gating Register (CCM_CCGR19_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4140	CCM Clock Gating Register (CCM_CCGR20)	32	R/W	0000_0002h	5.1.7.6/510
3038_4144	CCM Clock Gating Register (CCM_CCGR20_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4148	CCM Clock Gating Register (CCM_CCGR20_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_414C	CCM Clock Gating Register (CCM_CCGR20_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4150	CCM Clock Gating Register (CCM_CCGR21)	32	R/W	0000_0002h	5.1.7.6/510
3038_4154	CCM Clock Gating Register (CCM_CCGR21_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4158	CCM Clock Gating Register (CCM_CCGR21_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_415C	CCM Clock Gating Register (CCM_CCGR21_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4160	CCM Clock Gating Register (CCM_CCGR22)	32	R/W	0000_0002h	5.1.7.6/510
3038_4164	CCM Clock Gating Register (CCM_CCGR22_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4168	CCM Clock Gating Register (CCM_CCGR22_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_416C	CCM Clock Gating Register (CCM_CCGR22_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4170	CCM Clock Gating Register (CCM_CCGR23)	32	R/W	0000_0002h	5.1.7.6/510
3038_4174	CCM Clock Gating Register (CCM_CCGR23_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4178	CCM Clock Gating Register (CCM_CCGR23_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_417C	CCM Clock Gating Register (CCM_CCGR23_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4180	CCM Clock Gating Register (CCM_CCGR24)	32	R/W	0000_0002h	5.1.7.6/510
3038_4184	CCM Clock Gating Register (CCM_CCGR24_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4188	CCM Clock Gating Register (CCM_CCGR24_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_418C	CCM Clock Gating Register (CCM_CCGR24_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4190	CCM Clock Gating Register (CCM_CCGR25)	32	R/W	0000_0002h	5.1.7.6/510
3038_4194	CCM Clock Gating Register (CCM_CCGR25_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4198	CCM Clock Gating Register (CCM_CCGR25_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_419C	CCM Clock Gating Register (CCM_CCGR25_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41A0	CCM Clock Gating Register (CCM_CCGR26)	32	R/W	0000_0002h	5.1.7.6/510
3038_41A4	CCM Clock Gating Register (CCM_CCGR26_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_41A8	CCM Clock Gating Register (CCM_CCGR26_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41AC	CCM Clock Gating Register (CCM_CCGR26_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41B0	CCM Clock Gating Register (CCM_CCGR27)	32	R/W	0000_0002h	5.1.7.6/510
3038_41B4	CCM Clock Gating Register (CCM_CCGR27_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_41B8	CCM Clock Gating Register (CCM_CCGR27_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41BC	CCM Clock Gating Register (CCM_CCGR27_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41C0	CCM Clock Gating Register (CCM_CCGR28)	32	R/W	0000_0002h	5.1.7.6/510
3038_41C4	CCM Clock Gating Register (CCM_CCGR28_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_41C8	CCM Clock Gating Register (CCM_CCGR28_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41CC	CCM Clock Gating Register (CCM_CCGR28_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41D0	CCM Clock Gating Register (CCM_CCGR29)	32	R/W	0000_0002h	5.1.7.6/510
3038_41D4	CCM Clock Gating Register (CCM_CCGR29_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_41D8	CCM Clock Gating Register (CCM_CCGR29_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41DC	CCM Clock Gating Register (CCM_CCGR29_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41E0	CCM Clock Gating Register (CCM_CCGR30)	32	R/W	0000_0002h	5.1.7.6/510
3038_41E4	CCM Clock Gating Register (CCM_CCGR30_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_41E8	CCM Clock Gating Register (CCM_CCGR30_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41EC	CCM Clock Gating Register (CCM_CCGR30_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_41F0	CCM Clock Gating Register (CCM_CCGR31)	32	R/W	0000_0002h	5.1.7.6/510
3038_41F4	CCM Clock Gating Register (CCM_CCGR31_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_41F8	CCM Clock Gating Register (CCM_CCGR31_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_41FC	CCM Clock Gating Register (CCM_CCGR31_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4200	CCM Clock Gating Register (CCM_CCGR32)	32	R/W	0000_0002h	5.1.7.6/510
3038_4204	CCM Clock Gating Register (CCM_CCGR32_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4208	CCM Clock Gating Register (CCM_CCGR32_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_420C	CCM Clock Gating Register (CCM_CCGR32_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4210	CCM Clock Gating Register (CCM_CCGR33)	32	R/W	0000_0002h	5.1.7.6/510
3038_4214	CCM Clock Gating Register (CCM_CCGR33_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4218	CCM Clock Gating Register (CCM_CCGR33_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_421C	CCM Clock Gating Register (CCM_CCGR33_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4220	CCM Clock Gating Register (CCM_CCGR34)	32	R/W	0000_0002h	5.1.7.6/510
3038_4224	CCM Clock Gating Register (CCM_CCGR34_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4228	CCM Clock Gating Register (CCM_CCGR34_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_422C	CCM Clock Gating Register (CCM_CCGR34_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4230	CCM Clock Gating Register (CCM_CCGR35)	32	R/W	0000_0002h	5.1.7.6/510
3038_4234	CCM Clock Gating Register (CCM_CCGR35_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4238	CCM Clock Gating Register (CCM_CCGR35_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_423C	CCM Clock Gating Register (CCM_CCGR35_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4240	CCM Clock Gating Register (CCM_CCGR36)	32	R/W	0000_0002h	5.1.7.6/510
3038_4244	CCM Clock Gating Register (CCM_CCGR36_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4248	CCM Clock Gating Register (CCM_CCGR36_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_424C	CCM Clock Gating Register (CCM_CCGR36_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4250	CCM Clock Gating Register (CCM_CCGR37)	32	R/W	0000_0002h	5.1.7.6/510
3038_4254	CCM Clock Gating Register (CCM_CCGR37_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4258	CCM Clock Gating Register (CCM_CCGR37_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_425C	CCM Clock Gating Register (CCM_CCGR37_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4260	CCM Clock Gating Register (CCM_CCGR38)	32	R/W	0000_0002h	5.1.7.6/510
3038_4264	CCM Clock Gating Register (CCM_CCGR38_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4268	CCM Clock Gating Register (CCM_CCGR38_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_426C	CCM Clock Gating Register (CCM_CCGR38_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4270	CCM Clock Gating Register (CCM_CCGR39)	32	R/W	0000_0002h	5.1.7.6/510
3038_4274	CCM Clock Gating Register (CCM_CCGR39_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4278	CCM Clock Gating Register (CCM_CCGR39_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_427C	CCM Clock Gating Register (CCM_CCGR39_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4280	CCM Clock Gating Register (CCM_CCGR40)	32	R/W	0000_0002h	5.1.7.6/510
3038_4284	CCM Clock Gating Register (CCM_CCGR40_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4288	CCM Clock Gating Register (CCM_CCGR40_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_428C	CCM Clock Gating Register (CCM_CCGR40_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4290	CCM Clock Gating Register (CCM_CCGR41)	32	R/W	0000_0002h	5.1.7.6/510
3038_4294	CCM Clock Gating Register (CCM_CCGR41_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4298	CCM Clock Gating Register (CCM_CCGR41_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_429C	CCM Clock Gating Register (CCM_CCGR41_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42A0	CCM Clock Gating Register (CCM_CCGR42)	32	R/W	0000_0002h	5.1.7.6/510
3038_42A4	CCM Clock Gating Register (CCM_CCGR42_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_42A8	CCM Clock Gating Register (CCM_CCGR42_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42AC	CCM Clock Gating Register (CCM_CCGR42_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42B0	CCM Clock Gating Register (CCM_CCGR43)	32	R/W	0000_0002h	5.1.7.6/510
3038_42B4	CCM Clock Gating Register (CCM_CCGR43_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_42B8	CCM Clock Gating Register (CCM_CCGR43_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42BC	CCM Clock Gating Register (CCM_CCGR43_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42C0	CCM Clock Gating Register (CCM_CCGR44)	32	R/W	0000_0002h	5.1.7.6/510
3038_42C4	CCM Clock Gating Register (CCM_CCGR44_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_42C8	CCM Clock Gating Register (CCM_CCGR44_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42CC	CCM Clock Gating Register (CCM_CCGR44_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42D0	CCM Clock Gating Register (CCM_CCGR45)	32	R/W	0000_0002h	5.1.7.6/510
3038_42D4	CCM Clock Gating Register (CCM_CCGR45_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_42D8	CCM Clock Gating Register (CCM_CCGR45_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42DC	CCM Clock Gating Register (CCM_CCGR45_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42E0	CCM Clock Gating Register (CCM_CCGR46)	32	R/W	0000_0002h	5.1.7.6/510
3038_42E4	CCM Clock Gating Register (CCM_CCGR46_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_42E8	CCM Clock Gating Register (CCM_CCGR46_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42EC	CCM Clock Gating Register (CCM_CCGR46_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_42F0	CCM Clock Gating Register (CCM_CCGR47)	32	R/W	0000_0002h	5.1.7.6/510
3038_42F4	CCM Clock Gating Register (CCM_CCGR47_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_42F8	CCM Clock Gating Register (CCM_CCGR47_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_42FC	CCM Clock Gating Register (CCM_CCGR47_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4300	CCM Clock Gating Register (CCM_CCGR48)	32	R/W	0000_0002h	5.1.7.6/510
3038_4304	CCM Clock Gating Register (CCM_CCGR48_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4308	CCM Clock Gating Register (CCM_CCGR48_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_430C	CCM Clock Gating Register (CCM_CCGR48_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4310	CCM Clock Gating Register (CCM_CCGR49)	32	R/W	0000_0002h	5.1.7.6/510
3038_4314	CCM Clock Gating Register (CCM_CCGR49_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4318	CCM Clock Gating Register (CCM_CCGR49_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_431C	CCM Clock Gating Register (CCM_CCGR49_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4320	CCM Clock Gating Register (CCM_CCGR50)	32	R/W	0000_0002h	5.1.7.6/510
3038_4324	CCM Clock Gating Register (CCM_CCGR50_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4328	CCM Clock Gating Register (CCM_CCGR50_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_432C	CCM Clock Gating Register (CCM_CCGR50_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4330	CCM Clock Gating Register (CCM_CCGR51)	32	R/W	0000_0002h	5.1.7.6/510
3038_4334	CCM Clock Gating Register (CCM_CCGR51_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4338	CCM Clock Gating Register (CCM_CCGR51_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_433C	CCM Clock Gating Register (CCM_CCGR51_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4340	CCM Clock Gating Register (CCM_CCGR52)	32	R/W	0000_0002h	5.1.7.6/510
3038_4344	CCM Clock Gating Register (CCM_CCGR52_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4348	CCM Clock Gating Register (CCM_CCGR52_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_434C	CCM Clock Gating Register (CCM_CCGR52_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4350	CCM Clock Gating Register (CCM_CCGR53)	32	R/W	0000_0002h	5.1.7.6/510
3038_4354	CCM Clock Gating Register (CCM_CCGR53_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4358	CCM Clock Gating Register (CCM_CCGR53_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_435C	CCM Clock Gating Register (CCM_CCGR53_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4360	CCM Clock Gating Register (CCM_CCGR54)	32	R/W	0000_0002h	5.1.7.6/510
3038_4364	CCM Clock Gating Register (CCM_CCGR54_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4368	CCM Clock Gating Register (CCM_CCGR54_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_436C	CCM Clock Gating Register (CCM_CCGR54_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4370	CCM Clock Gating Register (CCM_CCGR55)	32	R/W	0000_0002h	5.1.7.6/510
3038_4374	CCM Clock Gating Register (CCM_CCGR55_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4378	CCM Clock Gating Register (CCM_CCGR55_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_437C	CCM Clock Gating Register (CCM_CCGR55_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4380	CCM Clock Gating Register (CCM_CCGR56)	32	R/W	0000_0002h	5.1.7.6/510
3038_4384	CCM Clock Gating Register (CCM_CCGR56_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4388	CCM Clock Gating Register (CCM_CCGR56_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_438C	CCM Clock Gating Register (CCM_CCGR56_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4390	CCM Clock Gating Register (CCM_CCGR57)	32	R/W	0000_0002h	5.1.7.6/510
3038_4394	CCM Clock Gating Register (CCM_CCGR57_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4398	CCM Clock Gating Register (CCM_CCGR57_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_439C	CCM Clock Gating Register (CCM_CCGR57_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43A0	CCM Clock Gating Register (CCM_CCGR58)	32	R/W	0000_0002h	5.1.7.6/510
3038_43A4	CCM Clock Gating Register (CCM_CCGR58_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43A8	CCM Clock Gating Register (CCM_CCGR58_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43AC	CCM Clock Gating Register (CCM_CCGR58_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43B0	CCM Clock Gating Register (CCM_CCGR59)	32	R/W	0000_0002h	5.1.7.6/510
3038_43B4	CCM Clock Gating Register (CCM_CCGR59_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43B8	CCM Clock Gating Register (CCM_CCGR59_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43BC	CCM Clock Gating Register (CCM_CCGR59_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43C0	CCM Clock Gating Register (CCM_CCGR60)	32	R/W	0000_0002h	5.1.7.6/510
3038_43C4	CCM Clock Gating Register (CCM_CCGR60_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43C8	CCM Clock Gating Register (CCM_CCGR60_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43CC	CCM Clock Gating Register (CCM_CCGR60_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43D0	CCM Clock Gating Register (CCM_CCGR61)	32	R/W	0000_0002h	5.1.7.6/510
3038_43D4	CCM Clock Gating Register (CCM_CCGR61_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43D8	CCM Clock Gating Register (CCM_CCGR61_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43DC	CCM Clock Gating Register (CCM_CCGR61_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43E0	CCM Clock Gating Register (CCM_CCGR62)	32	R/W	0000_0002h	5.1.7.6/510
3038_43E4	CCM Clock Gating Register (CCM_CCGR62_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43E8	CCM Clock Gating Register (CCM_CCGR62_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43EC	CCM Clock Gating Register (CCM_CCGR62_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_43F0	CCM Clock Gating Register (CCM_CCGR63)	32	R/W	0000_0002h	5.1.7.6/510
3038_43F4	CCM Clock Gating Register (CCM_CCGR63_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_43F8	CCM Clock Gating Register (CCM_CCGR63_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_43FC	CCM Clock Gating Register (CCM_CCGR63_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4400	CCM Clock Gating Register (CCM_CCGR64)	32	R/W	0000_0002h	5.1.7.6/510
3038_4404	CCM Clock Gating Register (CCM_CCGR64_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4408	CCM Clock Gating Register (CCM_CCGR64_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_440C	CCM Clock Gating Register (CCM_CCGR64_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4410	CCM Clock Gating Register (CCM_CCGR65)	32	R/W	0000_0002h	5.1.7.6/510
3038_4414	CCM Clock Gating Register (CCM_CCGR65_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4418	CCM Clock Gating Register (CCM_CCGR65_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_441C	CCM Clock Gating Register (CCM_CCGR65_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4420	CCM Clock Gating Register (CCM_CCGR66)	32	R/W	0000_0002h	5.1.7.6/510
3038_4424	CCM Clock Gating Register (CCM_CCGR66_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4428	CCM Clock Gating Register (CCM_CCGR66_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_442C	CCM Clock Gating Register (CCM_CCGR66_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4430	CCM Clock Gating Register (CCM_CCGR67)	32	R/W	0000_0002h	5.1.7.6/510
3038_4434	CCM Clock Gating Register (CCM_CCGR67_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4438	CCM Clock Gating Register (CCM_CCGR67_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_443C	CCM Clock Gating Register (CCM_CCGR67_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4440	CCM Clock Gating Register (CCM_CCGR68)	32	R/W	0000_0002h	5.1.7.6/510
3038_4444	CCM Clock Gating Register (CCM_CCGR68_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4448	CCM Clock Gating Register (CCM_CCGR68_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_444C	CCM Clock Gating Register (CCM_CCGR68_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4450	CCM Clock Gating Register (CCM_CCGR69)	32	R/W	0000_0002h	5.1.7.6/510
3038_4454	CCM Clock Gating Register (CCM_CCGR69_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4458	CCM Clock Gating Register (CCM_CCGR69_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_445C	CCM Clock Gating Register (CCM_CCGR69_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4460	CCM Clock Gating Register (CCM_CCGR70)	32	R/W	0000_0002h	5.1.7.6/510
3038_4464	CCM Clock Gating Register (CCM_CCGR70_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4468	CCM Clock Gating Register (CCM_CCGR70_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_446C	CCM Clock Gating Register (CCM_CCGR70_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4470	CCM Clock Gating Register (CCM_CCGR71)	32	R/W	0000_0002h	5.1.7.6/510
3038_4474	CCM Clock Gating Register (CCM_CCGR71_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4478	CCM Clock Gating Register (CCM_CCGR71_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_447C	CCM Clock Gating Register (CCM_CCGR71_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4480	CCM Clock Gating Register (CCM_CCGR72)	32	R/W	0000_0002h	5.1.7.6/510
3038_4484	CCM Clock Gating Register (CCM_CCGR72_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4488	CCM Clock Gating Register (CCM_CCGR72_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_448C	CCM Clock Gating Register (CCM_CCGR72_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4490	CCM Clock Gating Register (CCM_CCGR73)	32	R/W	0000_0002h	5.1.7.6/510
3038_4494	CCM Clock Gating Register (CCM_CCGR73_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4498	CCM Clock Gating Register (CCM_CCGR73_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_449C	CCM Clock Gating Register (CCM_CCGR73_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_44A0	CCM Clock Gating Register (CCM_CCGR74)	32	R/W	0000_0002h	5.1.7.6/510
3038_44A4	CCM Clock Gating Register (CCM_CCGR74_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44A8	CCM Clock Gating Register (CCM_CCGR74_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44AC	CCM Clock Gating Register (CCM_CCGR74_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_44B0	CCM Clock Gating Register (CCM_CCGR75)	32	R/W	0000_0002h	5.1.7.6/510
3038_44B4	CCM Clock Gating Register (CCM_CCGR75_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44B8	CCM Clock Gating Register (CCM_CCGR75_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44BC	CCM Clock Gating Register (CCM_CCGR75_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_44C0	CCM Clock Gating Register (CCM_CCGR76)	32	R/W	0000_0002h	5.1.7.6/510
3038_44C4	CCM Clock Gating Register (CCM_CCGR76_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44C8	CCM Clock Gating Register (CCM_CCGR76_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44CC	CCM Clock Gating Register (CCM_CCGR76_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_44D0	CCM Clock Gating Register (CCM_CCGR77)	32	R/W	0000_0002h	5.1.7.6/510
3038_44D4	CCM Clock Gating Register (CCM_CCGR77_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44D8	CCM Clock Gating Register (CCM_CCGR77_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44DC	CCM Clock Gating Register (CCM_CCGR77_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_44E0	CCM Clock Gating Register (CCM_CCGR78)	32	R/W	0000_0002h	5.1.7.6/510
3038_44E4	CCM Clock Gating Register (CCM_CCGR78_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44E8	CCM Clock Gating Register (CCM_CCGR78_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44EC	CCM Clock Gating Register (CCM_CCGR78_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_44F0	CCM Clock Gating Register (CCM_CCGR79)	32	R/W	0000_0002h	5.1.7.6/510
3038_44F4	CCM Clock Gating Register (CCM_CCGR79_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_44F8	CCM Clock Gating Register (CCM_CCGR79_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_44FC	CCM Clock Gating Register (CCM_CCGR79_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4500	CCM Clock Gating Register (CCM_CCGR80)	32	R/W	0000_0002h	5.1.7.6/510
3038_4504	CCM Clock Gating Register (CCM_CCGR80_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4508	CCM Clock Gating Register (CCM_CCGR80_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_450C	CCM Clock Gating Register (CCM_CCGR80_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4510	CCM Clock Gating Register (CCM_CCGR81)	32	R/W	0000_0002h	5.1.7.6/510
3038_4514	CCM Clock Gating Register (CCM_CCGR81_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4518	CCM Clock Gating Register (CCM_CCGR81_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_451C	CCM Clock Gating Register (CCM_CCGR81_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4520	CCM Clock Gating Register (CCM_CCGR82)	32	R/W	0000_0002h	5.1.7.6/510
3038_4524	CCM Clock Gating Register (CCM_CCGR82_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4528	CCM Clock Gating Register (CCM_CCGR82_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_452C	CCM Clock Gating Register (CCM_CCGR82_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4530	CCM Clock Gating Register (CCM_CCGR83)	32	R/W	0000_0002h	5.1.7.6/510
3038_4534	CCM Clock Gating Register (CCM_CCGR83_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4538	CCM Clock Gating Register (CCM_CCGR83_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_453C	CCM Clock Gating Register (CCM_CCGR83_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4540	CCM Clock Gating Register (CCM_CCGR84)	32	R/W	0000_0002h	5.1.7.6/510
3038_4544	CCM Clock Gating Register (CCM_CCGR84_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4548	CCM Clock Gating Register (CCM_CCGR84_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_454C	CCM Clock Gating Register (CCM_CCGR84_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4550	CCM Clock Gating Register (CCM_CCGR85)	32	R/W	0000_0002h	5.1.7.6/510
3038_4554	CCM Clock Gating Register (CCM_CCGR85_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4558	CCM Clock Gating Register (CCM_CCGR85_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_455C	CCM Clock Gating Register (CCM_CCGR85_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4560	CCM Clock Gating Register (CCM_CCGR86)	32	R/W	0000_0002h	5.1.7.6/510
3038_4564	CCM Clock Gating Register (CCM_CCGR86_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4568	CCM Clock Gating Register (CCM_CCGR86_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_456C	CCM Clock Gating Register (CCM_CCGR86_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4570	CCM Clock Gating Register (CCM_CCGR87)	32	R/W	0000_0002h	5.1.7.6/510
3038_4574	CCM Clock Gating Register (CCM_CCGR87_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4578	CCM Clock Gating Register (CCM_CCGR87_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_457C	CCM Clock Gating Register (CCM_CCGR87_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4580	CCM Clock Gating Register (CCM_CCGR88)	32	R/W	0000_0002h	5.1.7.6/510
3038_4584	CCM Clock Gating Register (CCM_CCGR88_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4588	CCM Clock Gating Register (CCM_CCGR88_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_458C	CCM Clock Gating Register (CCM_CCGR88_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4590	CCM Clock Gating Register (CCM_CCGR89)	32	R/W	0000_0002h	5.1.7.6/510
3038_4594	CCM Clock Gating Register (CCM_CCGR89_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4598	CCM Clock Gating Register (CCM_CCGR89_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_459C	CCM Clock Gating Register (CCM_CCGR89_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_45A0	CCM Clock Gating Register (CCM_CCGR90)	32	R/W	0000_0002h	5.1.7.6/510
3038_45A4	CCM Clock Gating Register (CCM_CCGR90_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45A8	CCM Clock Gating Register (CCM_CCGR90_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45AC	CCM Clock Gating Register (CCM_CCGR90_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_45B0	CCM Clock Gating Register (CCM_CCGR91)	32	R/W	0000_0002h	5.1.7.6/510
3038_45B4	CCM Clock Gating Register (CCM_CCGR91_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45B8	CCM Clock Gating Register (CCM_CCGR91_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45BC	CCM Clock Gating Register (CCM_CCGR91_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_45C0	CCM Clock Gating Register (CCM_CCGR92)	32	R/W	0000_0002h	5.1.7.6/510
3038_45C4	CCM Clock Gating Register (CCM_CCGR92_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45C8	CCM Clock Gating Register (CCM_CCGR92_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45CC	CCM Clock Gating Register (CCM_CCGR92_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_45D0	CCM Clock Gating Register (CCM_CCGR93)	32	R/W	0000_0002h	5.1.7.6/510
3038_45D4	CCM Clock Gating Register (CCM_CCGR93_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45D8	CCM Clock Gating Register (CCM_CCGR93_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45DC	CCM Clock Gating Register (CCM_CCGR93_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_45E0	CCM Clock Gating Register (CCM_CCGR94)	32	R/W	0000_0002h	5.1.7.6/510
3038_45E4	CCM Clock Gating Register (CCM_CCGR94_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45E8	CCM Clock Gating Register (CCM_CCGR94_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45EC	CCM Clock Gating Register (CCM_CCGR94_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_45F0	CCM Clock Gating Register (CCM_CCGR95)	32	R/W	0000_0002h	5.1.7.6/510
3038_45F4	CCM Clock Gating Register (CCM_CCGR95_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_45F8	CCM Clock Gating Register (CCM_CCGR95_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_45FC	CCM Clock Gating Register (CCM_CCGR95_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4600	CCM Clock Gating Register (CCM_CCGR96)	32	R/W	0000_0002h	5.1.7.6/510
3038_4604	CCM Clock Gating Register (CCM_CCGR96_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4608	CCM Clock Gating Register (CCM_CCGR96_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_460C	CCM Clock Gating Register (CCM_CCGR96_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4610	CCM Clock Gating Register (CCM_CCGR97)	32	R/W	0000_0002h	5.1.7.6/510
3038_4614	CCM Clock Gating Register (CCM_CCGR97_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4618	CCM Clock Gating Register (CCM_CCGR97_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_461C	CCM Clock Gating Register (CCM_CCGR97_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4620	CCM Clock Gating Register (CCM_CCGR98)	32	R/W	0000_0002h	5.1.7.6/510
3038_4624	CCM Clock Gating Register (CCM_CCGR98_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4628	CCM Clock Gating Register (CCM_CCGR98_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_462C	CCM Clock Gating Register (CCM_CCGR98_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4630	CCM Clock Gating Register (CCM_CCGR99)	32	R/W	0000_0002h	5.1.7.6/510
3038_4634	CCM Clock Gating Register (CCM_CCGR99_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4638	CCM Clock Gating Register (CCM_CCGR99_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_463C	CCM Clock Gating Register (CCM_CCGR99_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4640	CCM Clock Gating Register (CCM_CCGR100)	32	R/W	0000_0002h	5.1.7.6/510
3038_4644	CCM Clock Gating Register (CCM_CCGR100_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4648	CCM Clock Gating Register (CCM_CCGR100_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_464C	CCM Clock Gating Register (CCM_CCGR100_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4650	CCM Clock Gating Register (CCM_CCGR101)	32	R/W	0000_0002h	5.1.7.6/510
3038_4654	CCM Clock Gating Register (CCM_CCGR101_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4658	CCM Clock Gating Register (CCM_CCGR101_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_465C	CCM Clock Gating Register (CCM_CCGR101_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4660	CCM Clock Gating Register (CCM_CCGR102)	32	R/W	0000_0002h	5.1.7.6/510
3038_4664	CCM Clock Gating Register (CCM_CCGR102_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4668	CCM Clock Gating Register (CCM_CCGR102_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_466C	CCM Clock Gating Register (CCM_CCGR102_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4670	CCM Clock Gating Register (CCM_CCGR103)	32	R/W	0000_0002h	5.1.7.6/510
3038_4674	CCM Clock Gating Register (CCM_CCGR103_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4678	CCM Clock Gating Register (CCM_CCGR103_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_467C	CCM Clock Gating Register (CCM_CCGR103_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4680	CCM Clock Gating Register (CCM_CCGR104)	32	R/W	0000_0002h	5.1.7.6/510
3038_4684	CCM Clock Gating Register (CCM_CCGR104_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4688	CCM Clock Gating Register (CCM_CCGR104_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_468C	CCM Clock Gating Register (CCM_CCGR104_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4690	CCM Clock Gating Register (CCM_CCGR105)	32	R/W	0000_0002h	5.1.7.6/510
3038_4694	CCM Clock Gating Register (CCM_CCGR105_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4698	CCM Clock Gating Register (CCM_CCGR105_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_469C	CCM Clock Gating Register (CCM_CCGR105_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46A0	CCM Clock Gating Register (CCM_CCGR106)	32	R/W	0000_0002h	5.1.7.6/510
3038_46A4	CCM Clock Gating Register (CCM_CCGR106_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46A8	CCM Clock Gating Register (CCM_CCGR106_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46AC	CCM Clock Gating Register (CCM_CCGR106_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46B0	CCM Clock Gating Register (CCM_CCGR107)	32	R/W	0000_0002h	5.1.7.6/510
3038_46B4	CCM Clock Gating Register (CCM_CCGR107_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46B8	CCM Clock Gating Register (CCM_CCGR107_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46BC	CCM Clock Gating Register (CCM_CCGR107_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46C0	CCM Clock Gating Register (CCM_CCGR108)	32	R/W	0000_0002h	5.1.7.6/510
3038_46C4	CCM Clock Gating Register (CCM_CCGR108_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46C8	CCM Clock Gating Register (CCM_CCGR108_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46CC	CCM Clock Gating Register (CCM_CCGR108_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46D0	CCM Clock Gating Register (CCM_CCGR109)	32	R/W	0000_0002h	5.1.7.6/510
3038_46D4	CCM Clock Gating Register (CCM_CCGR109_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46D8	CCM Clock Gating Register (CCM_CCGR109_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46DC	CCM Clock Gating Register (CCM_CCGR109_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46E0	CCM Clock Gating Register (CCM_CCGR110)	32	R/W	0000_0002h	5.1.7.6/510
3038_46E4	CCM Clock Gating Register (CCM_CCGR110_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46E8	CCM Clock Gating Register (CCM_CCGR110_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46EC	CCM Clock Gating Register (CCM_CCGR110_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_46F0	CCM Clock Gating Register (CCM_CCGR111)	32	R/W	0000_0002h	5.1.7.6/510
3038_46F4	CCM Clock Gating Register (CCM_CCGR111_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_46F8	CCM Clock Gating Register (CCM_CCGR111_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_46FC	CCM Clock Gating Register (CCM_CCGR111_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4700	CCM Clock Gating Register (CCM_CCGR112)	32	R/W	0000_0002h	5.1.7.6/510
3038_4704	CCM Clock Gating Register (CCM_CCGR112_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4708	CCM Clock Gating Register (CCM_CCGR112_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_470C	CCM Clock Gating Register (CCM_CCGR112_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4710	CCM Clock Gating Register (CCM_CCGR113)	32	R/W	0000_0002h	5.1.7.6/510
3038_4714	CCM Clock Gating Register (CCM_CCGR113_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4718	CCM Clock Gating Register (CCM_CCGR113_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_471C	CCM Clock Gating Register (CCM_CCGR113_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4720	CCM Clock Gating Register (CCM_CCGR114)	32	R/W	0000_0002h	5.1.7.6/510
3038_4724	CCM Clock Gating Register (CCM_CCGR114_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4728	CCM Clock Gating Register (CCM_CCGR114_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_472C	CCM Clock Gating Register (CCM_CCGR114_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4730	CCM Clock Gating Register (CCM_CCGR115)	32	R/W	0000_0002h	5.1.7.6/510
3038_4734	CCM Clock Gating Register (CCM_CCGR115_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4738	CCM Clock Gating Register (CCM_CCGR115_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_473C	CCM Clock Gating Register (CCM_CCGR115_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4740	CCM Clock Gating Register (CCM_CCGR116)	32	R/W	0000_0002h	5.1.7.6/510
3038_4744	CCM Clock Gating Register (CCM_CCGR116_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4748	CCM Clock Gating Register (CCM_CCGR116_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_474C	CCM Clock Gating Register (CCM_CCGR116_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4750	CCM Clock Gating Register (CCM_CCGR117)	32	R/W	0000_0002h	5.1.7.6/510
3038_4754	CCM Clock Gating Register (CCM_CCGR117_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4758	CCM Clock Gating Register (CCM_CCGR117_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_475C	CCM Clock Gating Register (CCM_CCGR117_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4760	CCM Clock Gating Register (CCM_CCGR118)	32	R/W	0000_0002h	5.1.7.6/510
3038_4764	CCM Clock Gating Register (CCM_CCGR118_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4768	CCM Clock Gating Register (CCM_CCGR118_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_476C	CCM Clock Gating Register (CCM_CCGR118_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4770	CCM Clock Gating Register (CCM_CCGR119)	32	R/W	0000_0002h	5.1.7.6/510
3038_4774	CCM Clock Gating Register (CCM_CCGR119_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4778	CCM Clock Gating Register (CCM_CCGR119_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_477C	CCM Clock Gating Register (CCM_CCGR119_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4780	CCM Clock Gating Register (CCM_CCGR120)	32	R/W	0000_0002h	5.1.7.6/510
3038_4784	CCM Clock Gating Register (CCM_CCGR120_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4788	CCM Clock Gating Register (CCM_CCGR120_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_478C	CCM Clock Gating Register (CCM_CCGR120_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4790	CCM Clock Gating Register (CCM_CCGR121)	32	R/W	0000_0002h	5.1.7.6/510
3038_4794	CCM Clock Gating Register (CCM_CCGR121_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4798	CCM Clock Gating Register (CCM_CCGR121_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_479C	CCM Clock Gating Register (CCM_CCGR121_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47A0	CCM Clock Gating Register (CCM_CCGR122)	32	R/W	0000_0002h	5.1.7.6/510
3038_47A4	CCM Clock Gating Register (CCM_CCGR122_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_47A8	CCM Clock Gating Register (CCM_CCGR122_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47AC	CCM Clock Gating Register (CCM_CCGR122_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47B0	CCM Clock Gating Register (CCM_CCGR123)	32	R/W	0000_0002h	5.1.7.6/510
3038_47B4	CCM Clock Gating Register (CCM_CCGR123_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_47B8	CCM Clock Gating Register (CCM_CCGR123_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47BC	CCM Clock Gating Register (CCM_CCGR123_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47C0	CCM Clock Gating Register (CCM_CCGR124)	32	R/W	0000_0002h	5.1.7.6/510
3038_47C4	CCM Clock Gating Register (CCM_CCGR124_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_47C8	CCM Clock Gating Register (CCM_CCGR124_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47CC	CCM Clock Gating Register (CCM_CCGR124_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47D0	CCM Clock Gating Register (CCM_CCGR125)	32	R/W	0000_0002h	5.1.7.6/510
3038_47D4	CCM Clock Gating Register (CCM_CCGR125_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_47D8	CCM Clock Gating Register (CCM_CCGR125_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47DC	CCM Clock Gating Register (CCM_CCGR125_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47E0	CCM Clock Gating Register (CCM_CCGR126)	32	R/W	0000_0002h	5.1.7.6/510
3038_47E4	CCM Clock Gating Register (CCM_CCGR126_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_47E8	CCM Clock Gating Register (CCM_CCGR126_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47EC	CCM Clock Gating Register (CCM_CCGR126_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_47F0	CCM Clock Gating Register (CCM_CCGR127)	32	R/W	0000_0002h	5.1.7.6/510
3038_47F4	CCM Clock Gating Register (CCM_CCGR127_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_47F8	CCM Clock Gating Register (CCM_CCGR127_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_47FC	CCM Clock Gating Register (CCM_CCGR127_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4800	CCM Clock Gating Register (CCM_CCGR128)	32	R/W	0000_0002h	5.1.7.6/510
3038_4804	CCM Clock Gating Register (CCM_CCGR128_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4808	CCM Clock Gating Register (CCM_CCGR128_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_480C	CCM Clock Gating Register (CCM_CCGR128_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4810	CCM Clock Gating Register (CCM_CCGR129)	32	R/W	0000_0002h	5.1.7.6/510
3038_4814	CCM Clock Gating Register (CCM_CCGR129_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4818	CCM Clock Gating Register (CCM_CCGR129_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_481C	CCM Clock Gating Register (CCM_CCGR129_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4820	CCM Clock Gating Register (CCM_CCGR130)	32	R/W	0000_0002h	5.1.7.6/510
3038_4824	CCM Clock Gating Register (CCM_CCGR130_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4828	CCM Clock Gating Register (CCM_CCGR130_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_482C	CCM Clock Gating Register (CCM_CCGR130_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4830	CCM Clock Gating Register (CCM_CCGR131)	32	R/W	0000_0002h	5.1.7.6/510
3038_4834	CCM Clock Gating Register (CCM_CCGR131_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4838	CCM Clock Gating Register (CCM_CCGR131_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_483C	CCM Clock Gating Register (CCM_CCGR131_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4840	CCM Clock Gating Register (CCM_CCGR132)	32	R/W	0000_0002h	5.1.7.6/510
3038_4844	CCM Clock Gating Register (CCM_CCGR132_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4848	CCM Clock Gating Register (CCM_CCGR132_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_484C	CCM Clock Gating Register (CCM_CCGR132_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4850	CCM Clock Gating Register (CCM_CCGR133)	32	R/W	0000_0002h	5.1.7.6/510
3038_4854	CCM Clock Gating Register (CCM_CCGR133_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4858	CCM Clock Gating Register (CCM_CCGR133_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_485C	CCM Clock Gating Register (CCM_CCGR133_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4860	CCM Clock Gating Register (CCM_CCGR134)	32	R/W	0000_0002h	5.1.7.6/510
3038_4864	CCM Clock Gating Register (CCM_CCGR134_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4868	CCM Clock Gating Register (CCM_CCGR134_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_486C	CCM Clock Gating Register (CCM_CCGR134_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4870	CCM Clock Gating Register (CCM_CCGR135)	32	R/W	0000_0002h	5.1.7.6/510
3038_4874	CCM Clock Gating Register (CCM_CCGR135_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4878	CCM Clock Gating Register (CCM_CCGR135_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_487C	CCM Clock Gating Register (CCM_CCGR135_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4880	CCM Clock Gating Register (CCM_CCGR136)	32	R/W	0000_0002h	5.1.7.6/510
3038_4884	CCM Clock Gating Register (CCM_CCGR136_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4888	CCM Clock Gating Register (CCM_CCGR136_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_488C	CCM Clock Gating Register (CCM_CCGR136_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4890	CCM Clock Gating Register (CCM_CCGR137)	32	R/W	0000_0002h	5.1.7.6/510
3038_4894	CCM Clock Gating Register (CCM_CCGR137_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4898	CCM Clock Gating Register (CCM_CCGR137_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_489C	CCM Clock Gating Register (CCM_CCGR137_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48A0	CCM Clock Gating Register (CCM_CCGR138)	32	R/W	0000_0002h	5.1.7.6/510
3038_48A4	CCM Clock Gating Register (CCM_CCGR138_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_48A8	CCM Clock Gating Register (CCM_CCGR138_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48AC	CCM Clock Gating Register (CCM_CCGR138_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48B0	CCM Clock Gating Register (CCM_CCGR139)	32	R/W	0000_0002h	5.1.7.6/510
3038_48B4	CCM Clock Gating Register (CCM_CCGR139_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_48B8	CCM Clock Gating Register (CCM_CCGR139_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48BC	CCM Clock Gating Register (CCM_CCGR139_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48C0	CCM Clock Gating Register (CCM_CCGR140)	32	R/W	0000_0002h	5.1.7.6/510
3038_48C4	CCM Clock Gating Register (CCM_CCGR140_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_48C8	CCM Clock Gating Register (CCM_CCGR140_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48CC	CCM Clock Gating Register (CCM_CCGR140_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48D0	CCM Clock Gating Register (CCM_CCGR141)	32	R/W	0000_0002h	5.1.7.6/510
3038_48D4	CCM Clock Gating Register (CCM_CCGR141_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_48D8	CCM Clock Gating Register (CCM_CCGR141_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48DC	CCM Clock Gating Register (CCM_CCGR141_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48E0	CCM Clock Gating Register (CCM_CCGR142)	32	R/W	0000_0002h	5.1.7.6/510
3038_48E4	CCM Clock Gating Register (CCM_CCGR142_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_48E8	CCM Clock Gating Register (CCM_CCGR142_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48EC	CCM Clock Gating Register (CCM_CCGR142_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_48F0	CCM Clock Gating Register (CCM_CCGR143)	32	R/W	0000_0002h	5.1.7.6/510
3038_48F4	CCM Clock Gating Register (CCM_CCGR143_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_48F8	CCM Clock Gating Register (CCM_CCGR143_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_48FC	CCM Clock Gating Register (CCM_CCGR143_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4900	CCM Clock Gating Register (CCM_CCGR144)	32	R/W	0000_0002h	5.1.7.6/510
3038_4904	CCM Clock Gating Register (CCM_CCGR144_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4908	CCM Clock Gating Register (CCM_CCGR144_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_490C	CCM Clock Gating Register (CCM_CCGR144_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4910	CCM Clock Gating Register (CCM_CCGR145)	32	R/W	0000_0002h	5.1.7.6/510
3038_4914	CCM Clock Gating Register (CCM_CCGR145_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4918	CCM Clock Gating Register (CCM_CCGR145_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_491C	CCM Clock Gating Register (CCM_CCGR145_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4920	CCM Clock Gating Register (CCM_CCGR146)	32	R/W	0000_0002h	5.1.7.6/510
3038_4924	CCM Clock Gating Register (CCM_CCGR146_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4928	CCM Clock Gating Register (CCM_CCGR146_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_492C	CCM Clock Gating Register (CCM_CCGR146_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4930	CCM Clock Gating Register (CCM_CCGR147)	32	R/W	0000_0002h	5.1.7.6/510
3038_4934	CCM Clock Gating Register (CCM_CCGR147_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4938	CCM Clock Gating Register (CCM_CCGR147_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_493C	CCM Clock Gating Register (CCM_CCGR147_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4940	CCM Clock Gating Register (CCM_CCGR148)	32	R/W	0000_0002h	5.1.7.6/510
3038_4944	CCM Clock Gating Register (CCM_CCGR148_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4948	CCM Clock Gating Register (CCM_CCGR148_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_494C	CCM Clock Gating Register (CCM_CCGR148_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4950	CCM Clock Gating Register (CCM_CCGR149)	32	R/W	0000_0002h	5.1.7.6/510
3038_4954	CCM Clock Gating Register (CCM_CCGR149_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4958	CCM Clock Gating Register (CCM_CCGR149_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_495C	CCM Clock Gating Register (CCM_CCGR149_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4960	CCM Clock Gating Register (CCM_CCGR150)	32	R/W	0000_0002h	5.1.7.6/510
3038_4964	CCM Clock Gating Register (CCM_CCGR150_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4968	CCM Clock Gating Register (CCM_CCGR150_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_496C	CCM Clock Gating Register (CCM_CCGR150_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4970	CCM Clock Gating Register (CCM_CCGR151)	32	R/W	0000_0002h	5.1.7.6/510
3038_4974	CCM Clock Gating Register (CCM_CCGR151_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4978	CCM Clock Gating Register (CCM_CCGR151_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_497C	CCM Clock Gating Register (CCM_CCGR151_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4980	CCM Clock Gating Register (CCM_CCGR152)	32	R/W	0000_0002h	5.1.7.6/510
3038_4984	CCM Clock Gating Register (CCM_CCGR152_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4988	CCM Clock Gating Register (CCM_CCGR152_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_498C	CCM Clock Gating Register (CCM_CCGR152_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4990	CCM Clock Gating Register (CCM_CCGR153)	32	R/W	0000_0002h	5.1.7.6/510
3038_4994	CCM Clock Gating Register (CCM_CCGR153_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4998	CCM Clock Gating Register (CCM_CCGR153_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_499C	CCM Clock Gating Register (CCM_CCGR153_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49A0	CCM Clock Gating Register (CCM_CCGR154)	32	R/W	0000_0002h	5.1.7.6/510
3038_49A4	CCM Clock Gating Register (CCM_CCGR154_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49A8	CCM Clock Gating Register (CCM_CCGR154_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49AC	CCM Clock Gating Register (CCM_CCGR154_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49B0	CCM Clock Gating Register (CCM_CCGR155)	32	R/W	0000_0002h	5.1.7.6/510
3038_49B4	CCM Clock Gating Register (CCM_CCGR155_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49B8	CCM Clock Gating Register (CCM_CCGR155_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49BC	CCM Clock Gating Register (CCM_CCGR155_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49C0	CCM Clock Gating Register (CCM_CCGR156)	32	R/W	0000_0002h	5.1.7.6/510
3038_49C4	CCM Clock Gating Register (CCM_CCGR156_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49C8	CCM Clock Gating Register (CCM_CCGR156_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49CC	CCM Clock Gating Register (CCM_CCGR156_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49D0	CCM Clock Gating Register (CCM_CCGR157)	32	R/W	0000_0002h	5.1.7.6/510
3038_49D4	CCM Clock Gating Register (CCM_CCGR157_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49D8	CCM Clock Gating Register (CCM_CCGR157_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49DC	CCM Clock Gating Register (CCM_CCGR157_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49E0	CCM Clock Gating Register (CCM_CCGR158)	32	R/W	0000_0002h	5.1.7.6/510
3038_49E4	CCM Clock Gating Register (CCM_CCGR158_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49E8	CCM Clock Gating Register (CCM_CCGR158_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49EC	CCM Clock Gating Register (CCM_CCGR158_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_49F0	CCM Clock Gating Register (CCM_CCGR159)	32	R/W	0000_0002h	5.1.7.6/510
3038_49F4	CCM Clock Gating Register (CCM_CCGR159_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_49F8	CCM Clock Gating Register (CCM_CCGR159_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_49FC	CCM Clock Gating Register (CCM_CCGR159_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A00	CCM Clock Gating Register (CCM_CCGR160)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A04	CCM Clock Gating Register (CCM_CCGR160_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4A08	CCM Clock Gating Register (CCM_CCGR160_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A0C	CCM Clock Gating Register (CCM_CCGR160_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A10	CCM Clock Gating Register (CCM_CCGR161)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A14	CCM Clock Gating Register (CCM_CCGR161_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A18	CCM Clock Gating Register (CCM_CCGR161_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A1C	CCM Clock Gating Register (CCM_CCGR161_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A20	CCM Clock Gating Register (CCM_CCGR162)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A24	CCM Clock Gating Register (CCM_CCGR162_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A28	CCM Clock Gating Register (CCM_CCGR162_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A2C	CCM Clock Gating Register (CCM_CCGR162_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A30	CCM Clock Gating Register (CCM_CCGR163)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A34	CCM Clock Gating Register (CCM_CCGR163_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A38	CCM Clock Gating Register (CCM_CCGR163_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A3C	CCM Clock Gating Register (CCM_CCGR163_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A40	CCM Clock Gating Register (CCM_CCGR164)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A44	CCM Clock Gating Register (CCM_CCGR164_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A48	CCM Clock Gating Register (CCM_CCGR164_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A4C	CCM Clock Gating Register (CCM_CCGR164_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A50	CCM Clock Gating Register (CCM_CCGR165)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A54	CCM Clock Gating Register (CCM_CCGR165_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A58	CCM Clock Gating Register (CCM_CCGR165_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A5C	CCM Clock Gating Register (CCM_CCGR165_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A60	CCM Clock Gating Register (CCM_CCGR166)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A64	CCM Clock Gating Register (CCM_CCGR166_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A68	CCM Clock Gating Register (CCM_CCGR166_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A6C	CCM Clock Gating Register (CCM_CCGR166_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A70	CCM Clock Gating Register (CCM_CCGR167)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A74	CCM Clock Gating Register (CCM_CCGR167_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A78	CCM Clock Gating Register (CCM_CCGR167_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A7C	CCM Clock Gating Register (CCM_CCGR167_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A80	CCM Clock Gating Register (CCM_CCGR168)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A84	CCM Clock Gating Register (CCM_CCGR168_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A88	CCM Clock Gating Register (CCM_CCGR168_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A8C	CCM Clock Gating Register (CCM_CCGR168_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4A90	CCM Clock Gating Register (CCM_CCGR169)	32	R/W	0000_0002h	5.1.7.6/510
3038_4A94	CCM Clock Gating Register (CCM_CCGR169_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4A98	CCM Clock Gating Register (CCM_CCGR169_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4A9C	CCM Clock Gating Register (CCM_CCGR169_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4AA0	CCM Clock Gating Register (CCM_CCGR170)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AA4	CCM Clock Gating Register (CCM_CCGR170_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AA8	CCM Clock Gating Register (CCM_CCGR170_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4AAC	CCM Clock Gating Register (CCM_CCGR170_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4AB0	CCM Clock Gating Register (CCM_CCGR171)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AB4	CCM Clock Gating Register (CCM_CCGR171_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AB8	CCM Clock Gating Register (CCM_CCGR171_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4ABC	CCM Clock Gating Register (CCM_CCGR171_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4AC0	CCM Clock Gating Register (CCM_CCGR172)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AC4	CCM Clock Gating Register (CCM_CCGR172_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AC8	CCM Clock Gating Register (CCM_CCGR172_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4ACC	CCM Clock Gating Register (CCM_CCGR172_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4AD0	CCM Clock Gating Register (CCM_CCGR173)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AD4	CCM Clock Gating Register (CCM_CCGR173_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AD8	CCM Clock Gating Register (CCM_CCGR173_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4ADC	CCM Clock Gating Register (CCM_CCGR173_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4AE0	CCM Clock Gating Register (CCM_CCGR174)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AE4	CCM Clock Gating Register (CCM_CCGR174_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AE8	CCM Clock Gating Register (CCM_CCGR174_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4AEC	CCM Clock Gating Register (CCM_CCGR174_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4AF0	CCM Clock Gating Register (CCM_CCGR175)	32	R/W	0000_0002h	5.1.7.6/510
3038_4AF4	CCM Clock Gating Register (CCM_CCGR175_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4AF8	CCM Clock Gating Register (CCM_CCGR175_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4AFC	CCM Clock Gating Register (CCM_CCGR175_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B00	CCM Clock Gating Register (CCM_CCGR176)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B04	CCM Clock Gating Register (CCM_CCGR176_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B08	CCM Clock Gating Register (CCM_CCGR176_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B0C	CCM Clock Gating Register (CCM_CCGR176_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B10	CCM Clock Gating Register (CCM_CCGR177)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B14	CCM Clock Gating Register (CCM_CCGR177_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B18	CCM Clock Gating Register (CCM_CCGR177_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B1C	CCM Clock Gating Register (CCM_CCGR177_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B20	CCM Clock Gating Register (CCM_CCGR178)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B24	CCM Clock Gating Register (CCM_CCGR178_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B28	CCM Clock Gating Register (CCM_CCGR178_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B2C	CCM Clock Gating Register (CCM_CCGR178_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B30	CCM Clock Gating Register (CCM_CCGR179)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B34	CCM Clock Gating Register (CCM_CCGR179_SET)	32	R/W	0000_0002h	5.1.7.7/512

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4B38	CCM Clock Gating Register (CCM_CCGR179_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B3C	CCM Clock Gating Register (CCM_CCGR179_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B40	CCM Clock Gating Register (CCM_CCGR180)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B44	CCM Clock Gating Register (CCM_CCGR180_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B48	CCM Clock Gating Register (CCM_CCGR180_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B4C	CCM Clock Gating Register (CCM_CCGR180_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B50	CCM Clock Gating Register (CCM_CCGR181)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B54	CCM Clock Gating Register (CCM_CCGR181_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B58	CCM Clock Gating Register (CCM_CCGR181_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B5C	CCM Clock Gating Register (CCM_CCGR181_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B60	CCM Clock Gating Register (CCM_CCGR182)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B64	CCM Clock Gating Register (CCM_CCGR182_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B68	CCM Clock Gating Register (CCM_CCGR182_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B6C	CCM Clock Gating Register (CCM_CCGR182_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B70	CCM Clock Gating Register (CCM_CCGR183)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B74	CCM Clock Gating Register (CCM_CCGR183_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B78	CCM Clock Gating Register (CCM_CCGR183_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B7C	CCM Clock Gating Register (CCM_CCGR183_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B80	CCM Clock Gating Register (CCM_CCGR184)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B84	CCM Clock Gating Register (CCM_CCGR184_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B88	CCM Clock Gating Register (CCM_CCGR184_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B8C	CCM Clock Gating Register (CCM_CCGR184_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4B90	CCM Clock Gating Register (CCM_CCGR185)	32	R/W	0000_0002h	5.1.7.6/510
3038_4B94	CCM Clock Gating Register (CCM_CCGR185_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4B98	CCM Clock Gating Register (CCM_CCGR185_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4B9C	CCM Clock Gating Register (CCM_CCGR185_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4BA0	CCM Clock Gating Register (CCM_CCGR186)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BA4	CCM Clock Gating Register (CCM_CCGR186_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BA8	CCM Clock Gating Register (CCM_CCGR186_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BAC	CCM Clock Gating Register (CCM_CCGR186_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4BB0	CCM Clock Gating Register (CCM_CCGR187)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BB4	CCM Clock Gating Register (CCM_CCGR187_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BB8	CCM Clock Gating Register (CCM_CCGR187_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BBC	CCM Clock Gating Register (CCM_CCGR187_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4BC0	CCM Clock Gating Register (CCM_CCGR188)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BC4	CCM Clock Gating Register (CCM_CCGR188_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BC8	CCM Clock Gating Register (CCM_CCGR188_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BCC	CCM Clock Gating Register (CCM_CCGR188_TOG)	32	R/W	0000_0002h	5.1.7.9/516

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_4BD0	CCM Clock Gating Register (CCM_CCGR189)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BD4	CCM Clock Gating Register (CCM_CCGR189_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BD8	CCM Clock Gating Register (CCM_CCGR189_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BDC	CCM Clock Gating Register (CCM_CCGR189_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4BE0	CCM Clock Gating Register (CCM_CCGR190)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BE4	CCM Clock Gating Register (CCM_CCGR190_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BE8	CCM Clock Gating Register (CCM_CCGR190_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BEC	CCM Clock Gating Register (CCM_CCGR190_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_4BF0	CCM Clock Gating Register (CCM_CCGR191)	32	R/W	0000_0002h	5.1.7.6/510
3038_4BF4	CCM Clock Gating Register (CCM_CCGR191_SET)	32	R/W	0000_0002h	5.1.7.7/512
3038_4BF8	CCM Clock Gating Register (CCM_CCGR191_CLR)	32	R/W	0000_0002h	5.1.7.8/514
3038_4BFC	CCM Clock Gating Register (CCM_CCGR191_TOG)	32	R/W	0000_0002h	5.1.7.9/516
3038_8000	Target Register (CCM_TARGET_ROOT0)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8004	Target Register (CCM_TARGET_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8008	Target Register (CCM_TARGET_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_800C	Target Register (CCM_TARGET_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8010	Miscellaneous Register (CCM_MISC0)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8014	Miscellaneous Register (CCM_MISC_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8018	Miscellaneous Register (CCM_MISC_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_801C	Miscellaneous Register (CCM_MISC_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8020	Post Divider Register (CCM_POST0)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8024	Post Divider Register (CCM_POST_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8028	Post Divider Register (CCM_POST_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_802C	Post Divider Register (CCM_POST_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8030	Pre Divider Register (CCM_PRE0)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8034	Pre Divider Register (CCM_PRE_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8038	Pre Divider Register (CCM_PRE_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_803C	Pre Divider Register (CCM_PRE_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8070	Access Control Register (CCM_ACCESS_CTRL0)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8074	Access Control Register (CCM_ACCESS_CTRL_ROOT0_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8078	Access Control Register (CCM_ACCESS_CTRL_ROOT0_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_807C	Access Control Register (CCM_ACCESS_CTRL_ROOT0_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8080	Target Register (CCM_TARGET_ROOT1)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8084	Target Register (CCM_TARGET_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8088	Target Register (CCM_TARGET_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_808C	Target Register (CCM_TARGET_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8090	Miscellaneous Register (CCM_MISC1)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8094	Miscellaneous Register (CCM_MISC_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8098	Miscellaneous Register (CCM_MISC_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_809C	Miscellaneous Register (CCM_MISC_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_80A0	Post Divider Register (CCM_POST1)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_80A4	Post Divider Register (CCM_POST_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_80A8	Post Divider Register (CCM_POST_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_80AC	Post Divider Register (CCM_POST_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_80B0	Pre Divider Register (CCM_PRE1)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_80B4	Pre Divider Register (CCM_PRE_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_80B8	Pre Divider Register (CCM_PRE_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_80BC	Pre Divider Register (CCM_PRE_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_80F0	Access Control Register (CCM_ACCESS_CTRL1)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_80F4	Access Control Register (CCM_ACCESS_CTRL_ROOT1_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_80F8	Access Control Register (CCM_ACCESS_CTRL_ROOT1_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_80FC	Access Control Register (CCM_ACCESS_CTRL_ROOT1_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8100	Target Register (CCM_TARGET_ROOT2)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8104	Target Register (CCM_TARGET_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8108	Target Register (CCM_TARGET_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_810C	Target Register (CCM_TARGET_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8110	Miscellaneous Register (CCM_MISC2)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8114	Miscellaneous Register (CCM_MISC_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8118	Miscellaneous Register (CCM_MISC_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_811C	Miscellaneous Register (CCM_MISC_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8120	Post Divider Register (CCM_POST2)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8124	Post Divider Register (CCM_POST_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8128	Post Divider Register (CCM_POST_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_812C	Post Divider Register (CCM_POST_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8130	Pre Divider Register (CCM_PRE2)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8134	Pre Divider Register (CCM_PRE_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8138	Pre Divider Register (CCM_PRE_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_813C	Pre Divider Register (CCM_PRE_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8170	Access Control Register (CCM_ACCESS_CTRL2)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8174	Access Control Register (CCM_ACCESS_CTRL_ROOT2_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8178	Access Control Register (CCM_ACCESS_CTRL_ROOT2_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_817C	Access Control Register (CCM_ACCESS_CTRL_ROOT2_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8180	Target Register (CCM_TARGET_ROOT3)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8184	Target Register (CCM_TARGET_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8188	Target Register (CCM_TARGET_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_818C	Target Register (CCM_TARGET_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8190	Miscellaneous Register (CCM_MISC3)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8194	Miscellaneous Register (CCM_MISC_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8198	Miscellaneous Register (CCM_MISC_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_819C	Miscellaneous Register (CCM_MISC_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_81A0	Post Divider Register (CCM_POST3)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_81A4	Post Divider Register (CCM_POST_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_81A8	Post Divider Register (CCM_POST_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_81AC	Post Divider Register (CCM_POST_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_81B0	Pre Divider Register (CCM_PRE3)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_81B4	Pre Divider Register (CCM_PRE_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_81B8	Pre Divider Register (CCM_PRE_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_81BC	Pre Divider Register (CCM_PRE_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_81F0	Access Control Register (CCM_ACCESS_CTRL3)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_81F4	Access Control Register (CCM_ACCESS_CTRL_ROOT3_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_81F8	Access Control Register (CCM_ACCESS_CTRL_ROOT3_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_81FC	Access Control Register (CCM_ACCESS_CTRL_ROOT3_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8200	Target Register (CCM_TARGET_ROOT4)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8204	Target Register (CCM_TARGET_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8208	Target Register (CCM_TARGET_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_820C	Target Register (CCM_TARGET_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8210	Miscellaneous Register (CCM_MISC4)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8214	Miscellaneous Register (CCM_MISC_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8218	Miscellaneous Register (CCM_MISC_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_821C	Miscellaneous Register (CCM_MISC_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8220	Post Divider Register (CCM_POST4)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8224	Post Divider Register (CCM_POST_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8228	Post Divider Register (CCM_POST_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_822C	Post Divider Register (CCM_POST_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8230	Pre Divider Register (CCM_PRE4)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8234	Pre Divider Register (CCM_PRE_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8238	Pre Divider Register (CCM_PRE_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_823C	Pre Divider Register (CCM_PRE_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8270	Access Control Register (CCM_ACCESS_CTRL4)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8274	Access Control Register (CCM_ACCESS_CTRL_ROOT4_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8278	Access Control Register (CCM_ACCESS_CTRL_ROOT4_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_827C	Access Control Register (CCM_ACCESS_CTRL_ROOT4_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8280	Target Register (CCM_TARGET_ROOT5)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8284	Target Register (CCM_TARGET_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8288	Target Register (CCM_TARGET_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_828C	Target Register (CCM_TARGET_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8290	Miscellaneous Register (CCM_MISC5)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8294	Miscellaneous Register (CCM_MISC_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8298	Miscellaneous Register (CCM_MISC_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_829C	Miscellaneous Register (CCM_MISC_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_82A0	Post Divider Register (CCM_POST5)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_82A4	Post Divider Register (CCM_POST_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_82A8	Post Divider Register (CCM_POST_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_82AC	Post Divider Register (CCM_POST_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_82B0	Pre Divider Register (CCM_PRE5)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_82B4	Pre Divider Register (CCM_PRE_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_82B8	Pre Divider Register (CCM_PRE_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_82BC	Pre Divider Register (CCM_PRE_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_82F0	Access Control Register (CCM_ACCESS_CTRL5)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_82F4	Access Control Register (CCM_ACCESS_CTRL_ROOT5_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_82F8	Access Control Register (CCM_ACCESS_CTRL_ROOT5_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_82FC	Access Control Register (CCM_ACCESS_CTRL_ROOT5_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8300	Target Register (CCM_TARGET_ROOT6)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8304	Target Register (CCM_TARGET_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8308	Target Register (CCM_TARGET_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_830C	Target Register (CCM_TARGET_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8310	Miscellaneous Register (CCM_MISC6)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8314	Miscellaneous Register (CCM_MISC_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8318	Miscellaneous Register (CCM_MISC_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_831C	Miscellaneous Register (CCM_MISC_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8320	Post Divider Register (CCM_POST6)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8324	Post Divider Register (CCM_POST_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8328	Post Divider Register (CCM_POST_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_832C	Post Divider Register (CCM_POST_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8330	Pre Divider Register (CCM_PRE6)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8334	Pre Divider Register (CCM_PRE_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8338	Pre Divider Register (CCM_PRE_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_833C	Pre Divider Register (CCM_PRE_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8370	Access Control Register (CCM_ACCESS_CTRL6)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8374	Access Control Register (CCM_ACCESS_CTRL_ROOT6_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8378	Access Control Register (CCM_ACCESS_CTRL_ROOT6_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_837C	Access Control Register (CCM_ACCESS_CTRL_ROOT6_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8380	Target Register (CCM_TARGET_ROOT7)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8384	Target Register (CCM_TARGET_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8388	Target Register (CCM_TARGET_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_838C	Target Register (CCM_TARGET_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8390	Miscellaneous Register (CCM_MISC7)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8394	Miscellaneous Register (CCM_MISC_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8398	Miscellaneous Register (CCM_MISC_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_839C	Miscellaneous Register (CCM_MISC_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_83A0	Post Divider Register (CCM_POST7)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_83A4	Post Divider Register (CCM_POST_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_83A8	Post Divider Register (CCM_POST_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_83AC	Post Divider Register (CCM_POST_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_83B0	Pre Divider Register (CCM_PRE7)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_83B4	Pre Divider Register (CCM_PRE_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_83B8	Pre Divider Register (CCM_PRE_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_83BC	Pre Divider Register (CCM_PRE_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_83F0	Access Control Register (CCM_ACCESS_CTRL7)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_83F4	Access Control Register (CCM_ACCESS_CTRL_ROOT7_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_83F8	Access Control Register (CCM_ACCESS_CTRL_ROOT7_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_83FC	Access Control Register (CCM_ACCESS_CTRL_ROOT7_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8400	Target Register (CCM_TARGET_ROOT8)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8404	Target Register (CCM_TARGET_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8408	Target Register (CCM_TARGET_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_840C	Target Register (CCM_TARGET_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8410	Miscellaneous Register (CCM_MISC8)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8414	Miscellaneous Register (CCM_MISC_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8418	Miscellaneous Register (CCM_MISC_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_841C	Miscellaneous Register (CCM_MISC_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8420	Post Divider Register (CCM_POST8)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8424	Post Divider Register (CCM_POST_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8428	Post Divider Register (CCM_POST_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_842C	Post Divider Register (CCM_POST_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8430	Pre Divider Register (CCM_PRE8)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8434	Pre Divider Register (CCM_PRE_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8438	Pre Divider Register (CCM_PRE_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_843C	Pre Divider Register (CCM_PRE_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8470	Access Control Register (CCM_ACCESS_CTRL8)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8474	Access Control Register (CCM_ACCESS_CTRL_ROOT8_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8478	Access Control Register (CCM_ACCESS_CTRL_ROOT8_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_847C	Access Control Register (CCM_ACCESS_CTRL_ROOT8_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8480	Target Register (CCM_TARGET_ROOT9)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8484	Target Register (CCM_TARGET_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8488	Target Register (CCM_TARGET_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_848C	Target Register (CCM_TARGET_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8490	Miscellaneous Register (CCM_MISC9)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8494	Miscellaneous Register (CCM_MISC_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8498	Miscellaneous Register (CCM_MISC_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_849C	Miscellaneous Register (CCM_MISC_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_84A0	Post Divider Register (CCM_POST9)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_84A4	Post Divider Register (CCM_POST_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_84A8	Post Divider Register (CCM_POST_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_84AC	Post Divider Register (CCM_POST_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_84B0	Pre Divider Register (CCM_PRE9)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_84B4	Pre Divider Register (CCM_PRE_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_84B8	Pre Divider Register (CCM_PRE_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_84BC	Pre Divider Register (CCM_PRE_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_84F0	Access Control Register (CCM_ACCESS_CTRL9)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_84F4	Access Control Register (CCM_ACCESS_CTRL_ROOT9_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_84F8	Access Control Register (CCM_ACCESS_CTRL_ROOT9_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_84FC	Access Control Register (CCM_ACCESS_CTRL_ROOT9_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8500	Target Register (CCM_TARGET_ROOT10)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8504	Target Register (CCM_TARGET_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8508	Target Register (CCM_TARGET_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_850C	Target Register (CCM_TARGET_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8510	Miscellaneous Register (CCM_MISC10)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8514	Miscellaneous Register (CCM_MISC_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8518	Miscellaneous Register (CCM_MISC_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_851C	Miscellaneous Register (CCM_MISC_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8520	Post Divider Register (CCM_POST10)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8524	Post Divider Register (CCM_POST_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8528	Post Divider Register (CCM_POST_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_852C	Post Divider Register (CCM_POST_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8530	Pre Divider Register (CCM_PRE10)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8534	Pre Divider Register (CCM_PRE_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8538	Pre Divider Register (CCM_PRE_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_853C	Pre Divider Register (CCM_PRE_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8570	Access Control Register (CCM_ACCESS_CTRL10)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8574	Access Control Register (CCM_ACCESS_CTRL_ROOT10_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8578	Access Control Register (CCM_ACCESS_CTRL_ROOT10_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_857C	Access Control Register (CCM_ACCESS_CTRL_ROOT10_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8580	Target Register (CCM_TARGET_ROOT11)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8584	Target Register (CCM_TARGET_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8588	Target Register (CCM_TARGET_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_858C	Target Register (CCM_TARGET_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8590	Miscellaneous Register (CCM_MISC11)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8594	Miscellaneous Register (CCM_MISC_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8598	Miscellaneous Register (CCM_MISC_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_859C	Miscellaneous Register (CCM_MISC_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_85A0	Post Divider Register (CCM_POST11)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_85A4	Post Divider Register (CCM_POST_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_85A8	Post Divider Register (CCM_POST_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_85AC	Post Divider Register (CCM_POST_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_85B0	Pre Divider Register (CCM_PRE11)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_85B4	Pre Divider Register (CCM_PRE_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_85B8	Pre Divider Register (CCM_PRE_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_85BC	Pre Divider Register (CCM_PRE_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_85F0	Access Control Register (CCM_ACCESS_CTRL11)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_85F4	Access Control Register (CCM_ACCESS_CTRL_ROOT11_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_85F8	Access Control Register (CCM_ACCESS_CTRL_ROOT11_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_85FC	Access Control Register (CCM_ACCESS_CTRL_ROOT11_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8600	Target Register (CCM_TARGET_ROOT12)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8604	Target Register (CCM_TARGET_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8608	Target Register (CCM_TARGET_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_860C	Target Register (CCM_TARGET_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8610	Miscellaneous Register (CCM_MISC12)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8614	Miscellaneous Register (CCM_MISC_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8618	Miscellaneous Register (CCM_MISC_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_861C	Miscellaneous Register (CCM_MISC_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8620	Post Divider Register (CCM_POST12)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8624	Post Divider Register (CCM_POST_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8628	Post Divider Register (CCM_POST_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_862C	Post Divider Register (CCM_POST_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8630	Pre Divider Register (CCM_PRE12)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8634	Pre Divider Register (CCM_PRE_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8638	Pre Divider Register (CCM_PRE_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_863C	Pre Divider Register (CCM_PRE_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8670	Access Control Register (CCM_ACCESS_CTRL12)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8674	Access Control Register (CCM_ACCESS_CTRL_ROOT12_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8678	Access Control Register (CCM_ACCESS_CTRL_ROOT12_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_867C	Access Control Register (CCM_ACCESS_CTRL_ROOT12_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8680	Target Register (CCM_TARGET_ROOT13)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8684	Target Register (CCM_TARGET_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8688	Target Register (CCM_TARGET_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_868C	Target Register (CCM_TARGET_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8690	Miscellaneous Register (CCM_MISC13)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8694	Miscellaneous Register (CCM_MISC_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8698	Miscellaneous Register (CCM_MISC_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_869C	Miscellaneous Register (CCM_MISC_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_86A0	Post Divider Register (CCM_POST13)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_86A4	Post Divider Register (CCM_POST_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_86A8	Post Divider Register (CCM_POST_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_86AC	Post Divider Register (CCM_POST_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_86B0	Pre Divider Register (CCM_PRE13)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_86B4	Pre Divider Register (CCM_PRE_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_86B8	Pre Divider Register (CCM_PRE_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_86BC	Pre Divider Register (CCM_PRE_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_86F0	Access Control Register (CCM_ACCESS_CTRL13)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_86F4	Access Control Register (CCM_ACCESS_CTRL_ROOT13_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_86F8	Access Control Register (CCM_ACCESS_CTRL_ROOT13_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_86FC	Access Control Register (CCM_ACCESS_CTRL_ROOT13_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8700	Target Register (CCM_TARGET_ROOT14)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8704	Target Register (CCM_TARGET_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8708	Target Register (CCM_TARGET_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_870C	Target Register (CCM_TARGET_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8710	Miscellaneous Register (CCM_MISC14)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8714	Miscellaneous Register (CCM_MISC_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8718	Miscellaneous Register (CCM_MISC_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_871C	Miscellaneous Register (CCM_MISC_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8720	Post Divider Register (CCM_POST14)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8724	Post Divider Register (CCM_POST_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8728	Post Divider Register (CCM_POST_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_872C	Post Divider Register (CCM_POST_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8730	Pre Divider Register (CCM_PRE14)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8734	Pre Divider Register (CCM_PRE_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8738	Pre Divider Register (CCM_PRE_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_873C	Pre Divider Register (CCM_PRE_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8770	Access Control Register (CCM_ACCESS_CTRL14)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8774	Access Control Register (CCM_ACCESS_CTRL_ROOT14_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8778	Access Control Register (CCM_ACCESS_CTRL_ROOT14_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_877C	Access Control Register (CCM_ACCESS_CTRL_ROOT14_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8780	Target Register (CCM_TARGET_ROOT15)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8784	Target Register (CCM_TARGET_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8788	Target Register (CCM_TARGET_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_878C	Target Register (CCM_TARGET_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8790	Miscellaneous Register (CCM_MISC15)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8794	Miscellaneous Register (CCM_MISC_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8798	Miscellaneous Register (CCM_MISC_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_879C	Miscellaneous Register (CCM_MISC_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_87A0	Post Divider Register (CCM_POST15)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_87A4	Post Divider Register (CCM_POST_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_87A8	Post Divider Register (CCM_POST_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_87AC	Post Divider Register (CCM_POST_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_87B0	Pre Divider Register (CCM_PRE15)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_87B4	Pre Divider Register (CCM_PRE_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_87B8	Pre Divider Register (CCM_PRE_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_87BC	Pre Divider Register (CCM_PRE_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_87F0	Access Control Register (CCM_ACCESS_CTRL15)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_87F4	Access Control Register (CCM_ACCESS_CTRL_ROOT15_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_87F8	Access Control Register (CCM_ACCESS_CTRL_ROOT15_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_87FC	Access Control Register (CCM_ACCESS_CTRL_ROOT15_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8800	Target Register (CCM_TARGET_ROOT16)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8804	Target Register (CCM_TARGET_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8808	Target Register (CCM_TARGET_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_880C	Target Register (CCM_TARGET_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8810	Miscellaneous Register (CCM_MISC16)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8814	Miscellaneous Register (CCM_MISC_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8818	Miscellaneous Register (CCM_MISC_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_881C	Miscellaneous Register (CCM_MISC_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8820	Post Divider Register (CCM_POST16)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8824	Post Divider Register (CCM_POST_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8828	Post Divider Register (CCM_POST_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_882C	Post Divider Register (CCM_POST_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8830	Pre Divider Register (CCM_PRE16)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8834	Pre Divider Register (CCM_PRE_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8838	Pre Divider Register (CCM_PRE_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_883C	Pre Divider Register (CCM_PRE_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8870	Access Control Register (CCM_ACCESS_CTRL16)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8874	Access Control Register (CCM_ACCESS_CTRL_ROOT16_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8878	Access Control Register (CCM_ACCESS_CTRL_ROOT16_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_887C	Access Control Register (CCM_ACCESS_CTRL_ROOT16_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8880	Target Register (CCM_TARGET_ROOT17)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8884	Target Register (CCM_TARGET_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8888	Target Register (CCM_TARGET_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_888C	Target Register (CCM_TARGET_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8890	Miscellaneous Register (CCM_MISC17)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8894	Miscellaneous Register (CCM_MISC_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8898	Miscellaneous Register (CCM_MISC_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_889C	Miscellaneous Register (CCM_MISC_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_88A0	Post Divider Register (CCM_POST17)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_88A4	Post Divider Register (CCM_POST_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_88A8	Post Divider Register (CCM_POST_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_88AC	Post Divider Register (CCM_POST_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_88B0	Pre Divider Register (CCM_PRE17)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_88B4	Pre Divider Register (CCM_PRE_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_88B8	Pre Divider Register (CCM_PRE_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_88BC	Pre Divider Register (CCM_PRE_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_88F0	Access Control Register (CCM_ACCESS_CTRL17)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_88F4	Access Control Register (CCM_ACCESS_CTRL_ROOT17_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_88F8	Access Control Register (CCM_ACCESS_CTRL_ROOT17_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_88FC	Access Control Register (CCM_ACCESS_CTRL_ROOT17_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8900	Target Register (CCM_TARGET_ROOT18)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8904	Target Register (CCM_TARGET_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8908	Target Register (CCM_TARGET_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_890C	Target Register (CCM_TARGET_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8910	Miscellaneous Register (CCM_MISC18)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8914	Miscellaneous Register (CCM_MISC_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8918	Miscellaneous Register (CCM_MISC_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_891C	Miscellaneous Register (CCM_MISC_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8920	Post Divider Register (CCM_POST18)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8924	Post Divider Register (CCM_POST_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8928	Post Divider Register (CCM_POST_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_892C	Post Divider Register (CCM_POST_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8930	Pre Divider Register (CCM_PRE18)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8934	Pre Divider Register (CCM_PRE_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8938	Pre Divider Register (CCM_PRE_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_893C	Pre Divider Register (CCM_PRE_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8970	Access Control Register (CCM_ACCESS_CTRL18)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8974	Access Control Register (CCM_ACCESS_CTRL_ROOT18_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8978	Access Control Register (CCM_ACCESS_CTRL_ROOT18_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_897C	Access Control Register (CCM_ACCESS_CTRL_ROOT18_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8980	Target Register (CCM_TARGET_ROOT19)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8984	Target Register (CCM_TARGET_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8988	Target Register (CCM_TARGET_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_898C	Target Register (CCM_TARGET_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8990	Miscellaneous Register (CCM_MISC19)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8994	Miscellaneous Register (CCM_MISC_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8998	Miscellaneous Register (CCM_MISC_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_899C	Miscellaneous Register (CCM_MISC_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_89A0	Post Divider Register (CCM_POST19)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_89A4	Post Divider Register (CCM_POST_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_89A8	Post Divider Register (CCM_POST_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_89AC	Post Divider Register (CCM_POST_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_89B0	Pre Divider Register (CCM_PRE19)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_89B4	Pre Divider Register (CCM_PRE_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_89B8	Pre Divider Register (CCM_PRE_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_89BC	Pre Divider Register (CCM_PRE_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_89F0	Access Control Register (CCM_ACCESS_CTRL19)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_89F4	Access Control Register (CCM_ACCESS_CTRL_ROOT19_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_89F8	Access Control Register (CCM_ACCESS_CTRL_ROOT19_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_89FC	Access Control Register (CCM_ACCESS_CTRL_ROOT19_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8A00	Target Register (CCM_TARGET_ROOT20)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8A04	Target Register (CCM_TARGET_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8A08	Target Register (CCM_TARGET_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8A0C	Target Register (CCM_TARGET_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8A10	Miscellaneous Register (CCM_MISC20)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8A14	Miscellaneous Register (CCM_MISC_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8A18	Miscellaneous Register (CCM_MISC_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8A1C	Miscellaneous Register (CCM_MISC_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8A20	Post Divider Register (CCM_POST20)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8A24	Post Divider Register (CCM_POST_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8A28	Post Divider Register (CCM_POST_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8A2C	Post Divider Register (CCM_POST_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8A30	Pre Divider Register (CCM_PRE20)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8A34	Pre Divider Register (CCM_PRE_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8A38	Pre Divider Register (CCM_PRE_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8A3C	Pre Divider Register (CCM_PRE_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8A70	Access Control Register (CCM_ACCESS_CTRL20)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8A74	Access Control Register (CCM_ACCESS_CTRL_ROOT20_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8A78	Access Control Register (CCM_ACCESS_CTRL_ROOT20_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8A7C	Access Control Register (CCM_ACCESS_CTRL_ROOT20_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8A80	Target Register (CCM_TARGET_ROOT21)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8A84	Target Register (CCM_TARGET_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8A88	Target Register (CCM_TARGET_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8A8C	Target Register (CCM_TARGET_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8A90	Miscellaneous Register (CCM_MISC21)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8A94	Miscellaneous Register (CCM_MISC_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8A98	Miscellaneous Register (CCM_MISC_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8A9C	Miscellaneous Register (CCM_MISC_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8AA0	Post Divider Register (CCM_POST21)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8AA4	Post Divider Register (CCM_POST_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8AA8	Post Divider Register (CCM_POST_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8AAC	Post Divider Register (CCM_POST_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8AB0	Pre Divider Register (CCM_PRE21)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8AB4	Pre Divider Register (CCM_PRE_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8AB8	Pre Divider Register (CCM_PRE_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8ABC	Pre Divider Register (CCM_PRE_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8AF0	Access Control Register (CCM_ACCESS_CTRL21)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8AF4	Access Control Register (CCM_ACCESS_CTRL_ROOT21_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8AF8	Access Control Register (CCM_ACCESS_CTRL_ROOT21_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8AFC	Access Control Register (CCM_ACCESS_CTRL_ROOT21_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8B00	Target Register (CCM_TARGET_ROOT22)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8B04	Target Register (CCM_TARGET_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8B08	Target Register (CCM_TARGET_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8B0C	Target Register (CCM_TARGET_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8B10	Miscellaneous Register (CCM_MISC22)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8B14	Miscellaneous Register (CCM_MISC_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8B18	Miscellaneous Register (CCM_MISC_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8B1C	Miscellaneous Register (CCM_MISC_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8B20	Post Divider Register (CCM_POST22)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8B24	Post Divider Register (CCM_POST_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8B28	Post Divider Register (CCM_POST_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8B2C	Post Divider Register (CCM_POST_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8B30	Pre Divider Register (CCM_PRE22)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8B34	Pre Divider Register (CCM_PRE_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8B38	Pre Divider Register (CCM_PRE_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8B3C	Pre Divider Register (CCM_PRE_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8B70	Access Control Register (CCM_ACCESS_CTRL22)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8B74	Access Control Register (CCM_ACCESS_CTRL_ROOT22_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8B78	Access Control Register (CCM_ACCESS_CTRL_ROOT22_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8B7C	Access Control Register (CCM_ACCESS_CTRL_ROOT22_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8B80	Target Register (CCM_TARGET_ROOT23)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8B84	Target Register (CCM_TARGET_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8B88	Target Register (CCM_TARGET_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8B8C	Target Register (CCM_TARGET_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8B90	Miscellaneous Register (CCM_MISC23)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8B94	Miscellaneous Register (CCM_MISC_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8B98	Miscellaneous Register (CCM_MISC_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8B9C	Miscellaneous Register (CCM_MISC_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8BA0	Post Divider Register (CCM_POST23)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8BA4	Post Divider Register (CCM_POST_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8BA8	Post Divider Register (CCM_POST_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8BAC	Post Divider Register (CCM_POST_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8BB0	Pre Divider Register (CCM_PRE23)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8BB4	Pre Divider Register (CCM_PRE_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8BB8	Pre Divider Register (CCM_PRE_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8BBC	Pre Divider Register (CCM_PRE_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8BF0	Access Control Register (CCM_ACCESS_CTRL23)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8BF4	Access Control Register (CCM_ACCESS_CTRL_ROOT23_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8BF8	Access Control Register (CCM_ACCESS_CTRL_ROOT23_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8BFC	Access Control Register (CCM_ACCESS_CTRL_ROOT23_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8C00	Target Register (CCM_TARGET_ROOT24)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8C04	Target Register (CCM_TARGET_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8C08	Target Register (CCM_TARGET_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8C0C	Target Register (CCM_TARGET_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8C10	Miscellaneous Register (CCM_MISC24)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8C14	Miscellaneous Register (CCM_MISC_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8C18	Miscellaneous Register (CCM_MISC_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8C1C	Miscellaneous Register (CCM_MISC_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8C20	Post Divider Register (CCM_POST24)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8C24	Post Divider Register (CCM_POST_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8C28	Post Divider Register (CCM_POST_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8C2C	Post Divider Register (CCM_POST_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8C30	Pre Divider Register (CCM_PRE24)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8C34	Pre Divider Register (CCM_PRE_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8C38	Pre Divider Register (CCM_PRE_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8C3C	Pre Divider Register (CCM_PRE_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8C70	Access Control Register (CCM_ACCESS_CTRL24)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8C74	Access Control Register (CCM_ACCESS_CTRL_ROOT24_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8C78	Access Control Register (CCM_ACCESS_CTRL_ROOT24_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8C7C	Access Control Register (CCM_ACCESS_CTRL_ROOT24_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8C80	Target Register (CCM_TARGET_ROOT25)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8C84	Target Register (CCM_TARGET_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8C88	Target Register (CCM_TARGET_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8C8C	Target Register (CCM_TARGET_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8C90	Miscellaneous Register (CCM_MISC25)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8C94	Miscellaneous Register (CCM_MISC_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8C98	Miscellaneous Register (CCM_MISC_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8C9C	Miscellaneous Register (CCM_MISC_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8CA0	Post Divider Register (CCM_POST25)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8CA4	Post Divider Register (CCM_POST_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8CA8	Post Divider Register (CCM_POST_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8CAC	Post Divider Register (CCM_POST_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8CB0	Pre Divider Register (CCM_PRE25)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8CB4	Pre Divider Register (CCM_PRE_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8CB8	Pre Divider Register (CCM_PRE_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8CBC	Pre Divider Register (CCM_PRE_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8CF0	Access Control Register (CCM_ACCESS_CTRL25)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8CF4	Access Control Register (CCM_ACCESS_CTRL_ROOT25_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8CF8	Access Control Register (CCM_ACCESS_CTRL_ROOT25_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8CFC	Access Control Register (CCM_ACCESS_CTRL_ROOT25_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8D00	Target Register (CCM_TARGET_ROOT26)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8D04	Target Register (CCM_TARGET_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8D08	Target Register (CCM_TARGET_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8D0C	Target Register (CCM_TARGET_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8D10	Miscellaneous Register (CCM_MISC26)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8D14	Miscellaneous Register (CCM_MISC_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8D18	Miscellaneous Register (CCM_MISC_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8D1C	Miscellaneous Register (CCM_MISC_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8D20	Post Divider Register (CCM_POST26)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8D24	Post Divider Register (CCM_POST_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8D28	Post Divider Register (CCM_POST_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8D2C	Post Divider Register (CCM_POST_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8D30	Pre Divider Register (CCM_PRE26)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8D34	Pre Divider Register (CCM_PRE_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8D38	Pre Divider Register (CCM_PRE_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8D3C	Pre Divider Register (CCM_PRE_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8D70	Access Control Register (CCM_ACCESS_CTRL26)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8D74	Access Control Register (CCM_ACCESS_CTRL_ROOT26_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8D78	Access Control Register (CCM_ACCESS_CTRL_ROOT26_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8D7C	Access Control Register (CCM_ACCESS_CTRL_ROOT26_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8D80	Target Register (CCM_TARGET_ROOT27)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8D84	Target Register (CCM_TARGET_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8D88	Target Register (CCM_TARGET_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8D8C	Target Register (CCM_TARGET_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8D90	Miscellaneous Register (CCM_MISC27)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8D94	Miscellaneous Register (CCM_MISC_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8D98	Miscellaneous Register (CCM_MISC_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8D9C	Miscellaneous Register (CCM_MISC_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8DA0	Post Divider Register (CCM_POST27)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8DA4	Post Divider Register (CCM_POST_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8DA8	Post Divider Register (CCM_POST_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8DAC	Post Divider Register (CCM_POST_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8DB0	Pre Divider Register (CCM_PRE27)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8DB4	Pre Divider Register (CCM_PRE_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8DB8	Pre Divider Register (CCM_PRE_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8DBC	Pre Divider Register (CCM_PRE_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8DF0	Access Control Register (CCM_ACCESS_CTRL27)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8DF4	Access Control Register (CCM_ACCESS_CTRL_ROOT27_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8DF8	Access Control Register (CCM_ACCESS_CTRL_ROOT27_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8DFC	Access Control Register (CCM_ACCESS_CTRL_ROOT27_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8E00	Target Register (CCM_TARGET_ROOT28)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8E04	Target Register (CCM_TARGET_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8E08	Target Register (CCM_TARGET_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8E0C	Target Register (CCM_TARGET_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8E10	Miscellaneous Register (CCM_MISC28)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8E14	Miscellaneous Register (CCM_MISC_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8E18	Miscellaneous Register (CCM_MISC_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8E1C	Miscellaneous Register (CCM_MISC_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8E20	Post Divider Register (CCM_POST28)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8E24	Post Divider Register (CCM_POST_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8E28	Post Divider Register (CCM_POST_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8E2C	Post Divider Register (CCM_POST_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8E30	Pre Divider Register (CCM_PRE28)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8E34	Pre Divider Register (CCM_PRE_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8E38	Pre Divider Register (CCM_PRE_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8E3C	Pre Divider Register (CCM_PRE_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8E70	Access Control Register (CCM_ACCESS_CTRL28)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8E74	Access Control Register (CCM_ACCESS_CTRL_ROOT28_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8E78	Access Control Register (CCM_ACCESS_CTRL_ROOT28_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8E7C	Access Control Register (CCM_ACCESS_CTRL_ROOT28_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8E80	Target Register (CCM_TARGET_ROOT29)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8E84	Target Register (CCM_TARGET_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8E88	Target Register (CCM_TARGET_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8E8C	Target Register (CCM_TARGET_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8E90	Miscellaneous Register (CCM_MISC29)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8E94	Miscellaneous Register (CCM_MISC_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8E98	Miscellaneous Register (CCM_MISC_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8E9C	Miscellaneous Register (CCM_MISC_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8EA0	Post Divider Register (CCM_POST29)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8EA4	Post Divider Register (CCM_POST_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8EA8	Post Divider Register (CCM_POST_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8EAC	Post Divider Register (CCM_POST_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8EB0	Pre Divider Register (CCM_PRE29)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8EB4	Pre Divider Register (CCM_PRE_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8EB8	Pre Divider Register (CCM_PRE_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8EBC	Pre Divider Register (CCM_PRE_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8EF0	Access Control Register (CCM_ACCESS_CTRL29)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8EF4	Access Control Register (CCM_ACCESS_CTRL_ROOT29_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8EF8	Access Control Register (CCM_ACCESS_CTRL_ROOT29_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8EFC	Access Control Register (CCM_ACCESS_CTRL_ROOT29_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8F00	Target Register (CCM_TARGET_ROOT30)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8F04	Target Register (CCM_TARGET_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8F08	Target Register (CCM_TARGET_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8F0C	Target Register (CCM_TARGET_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8F10	Miscellaneous Register (CCM_MISC30)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8F14	Miscellaneous Register (CCM_MISC_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8F18	Miscellaneous Register (CCM_MISC_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8F1C	Miscellaneous Register (CCM_MISC_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8F20	Post Divider Register (CCM_POST30)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8F24	Post Divider Register (CCM_POST_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8F28	Post Divider Register (CCM_POST_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_8F2C	Post Divider Register (CCM_POST_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8F30	Pre Divider Register (CCM_PRE30)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8F34	Pre Divider Register (CCM_PRE_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8F38	Pre Divider Register (CCM_PRE_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8F3C	Pre Divider Register (CCM_PRE_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8F70	Access Control Register (CCM_ACCESS_CTRL30)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8F74	Access Control Register (CCM_ACCESS_CTRL_ROOT30_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8F78	Access Control Register (CCM_ACCESS_CTRL_ROOT30_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8F7C	Access Control Register (CCM_ACCESS_CTRL_ROOT30_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_8F80	Target Register (CCM_TARGET_ROOT31)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_8F84	Target Register (CCM_TARGET_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_8F88	Target Register (CCM_TARGET_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_8F8C	Target Register (CCM_TARGET_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_8F90	Miscellaneous Register (CCM_MISC31)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_8F94	Miscellaneous Register (CCM_MISC_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_8F98	Miscellaneous Register (CCM_MISC_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_8F9C	Miscellaneous Register (CCM_MISC_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_8FA0	Post Divider Register (CCM_POST31)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_8FA4	Post Divider Register (CCM_POST_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_8FA8	Post Divider Register (CCM_POST_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_8FAC	Post Divider Register (CCM_POST_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_8FB0	Pre Divider Register (CCM_PRE31)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_8FB4	Pre Divider Register (CCM_PRE_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_8FB8	Pre Divider Register (CCM_PRE_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_8FBC	Pre Divider Register (CCM_PRE_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_8FF0	Access Control Register (CCM_ACCESS_CTRL31)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_8FF4	Access Control Register (CCM_ACCESS_CTRL_ROOT31_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_8FF8	Access Control Register (CCM_ACCESS_CTRL_ROOT31_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_8FFC	Access Control Register (CCM_ACCESS_CTRL_ROOT31_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9000	Target Register (CCM_TARGET_ROOT32)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9004	Target Register (CCM_TARGET_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9008	Target Register (CCM_TARGET_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_900C	Target Register (CCM_TARGET_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9010	Miscellaneous Register (CCM_MISC32)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9014	Miscellaneous Register (CCM_MISC_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9018	Miscellaneous Register (CCM_MISC_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_901C	Miscellaneous Register (CCM_MISC_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9020	Post Divider Register (CCM_POST32)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9024	Post Divider Register (CCM_POST_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9028	Post Divider Register (CCM_POST_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_902C	Post Divider Register (CCM_POST_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9030	Pre Divider Register (CCM_PRE32)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9034	Pre Divider Register (CCM_PRE_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9038	Pre Divider Register (CCM_PRE_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_903C	Pre Divider Register (CCM_PRE_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9070	Access Control Register (CCM_ACCESS_CTRL32)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9074	Access Control Register (CCM_ACCESS_CTRL_ROOT32_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9078	Access Control Register (CCM_ACCESS_CTRL_ROOT32_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_907C	Access Control Register (CCM_ACCESS_CTRL_ROOT32_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9080	Target Register (CCM_TARGET_ROOT33)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9084	Target Register (CCM_TARGET_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9088	Target Register (CCM_TARGET_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_908C	Target Register (CCM_TARGET_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9090	Miscellaneous Register (CCM_MISC33)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9094	Miscellaneous Register (CCM_MISC_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9098	Miscellaneous Register (CCM_MISC_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_909C	Miscellaneous Register (CCM_MISC_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_90A0	Post Divider Register (CCM_POST33)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_90A4	Post Divider Register (CCM_POST_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_90A8	Post Divider Register (CCM_POST_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_90AC	Post Divider Register (CCM_POST_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_90B0	Pre Divider Register (CCM_PRE33)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_90B4	Pre Divider Register (CCM_PRE_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_90B8	Pre Divider Register (CCM_PRE_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_90BC	Pre Divider Register (CCM_PRE_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_90F0	Access Control Register (CCM_ACCESS_CTRL33)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_90F4	Access Control Register (CCM_ACCESS_CTRL_ROOT33_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_90F8	Access Control Register (CCM_ACCESS_CTRL_ROOT33_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_90FC	Access Control Register (CCM_ACCESS_CTRL_ROOT33_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9100	Target Register (CCM_TARGET_ROOT34)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9104	Target Register (CCM_TARGET_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9108	Target Register (CCM_TARGET_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_910C	Target Register (CCM_TARGET_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9110	Miscellaneous Register (CCM_MISC34)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9114	Miscellaneous Register (CCM_MISC_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9118	Miscellaneous Register (CCM_MISC_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_911C	Miscellaneous Register (CCM_MISC_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9120	Post Divider Register (CCM_POST34)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9124	Post Divider Register (CCM_POST_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9128	Post Divider Register (CCM_POST_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_912C	Post Divider Register (CCM_POST_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9130	Pre Divider Register (CCM_PRE34)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9134	Pre Divider Register (CCM_PRE_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9138	Pre Divider Register (CCM_PRE_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_913C	Pre Divider Register (CCM_PRE_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9170	Access Control Register (CCM_ACCESS_CTRL34)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9174	Access Control Register (CCM_ACCESS_CTRL_ROOT34_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9178	Access Control Register (CCM_ACCESS_CTRL_ROOT34_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_917C	Access Control Register (CCM_ACCESS_CTRL_ROOT34_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9180	Target Register (CCM_TARGET_ROOT35)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9184	Target Register (CCM_TARGET_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9188	Target Register (CCM_TARGET_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_918C	Target Register (CCM_TARGET_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9190	Miscellaneous Register (CCM_MISC35)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9194	Miscellaneous Register (CCM_MISC_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9198	Miscellaneous Register (CCM_MISC_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_919C	Miscellaneous Register (CCM_MISC_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_91A0	Post Divider Register (CCM_POST35)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_91A4	Post Divider Register (CCM_POST_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_91A8	Post Divider Register (CCM_POST_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_91AC	Post Divider Register (CCM_POST_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_91B0	Pre Divider Register (CCM_PRE35)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_91B4	Pre Divider Register (CCM_PRE_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_91B8	Pre Divider Register (CCM_PRE_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_91BC	Pre Divider Register (CCM_PRE_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_91F0	Access Control Register (CCM_ACCESS_CTRL35)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_91F4	Access Control Register (CCM_ACCESS_CTRL_ROOT35_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_91F8	Access Control Register (CCM_ACCESS_CTRL_ROOT35_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_91FC	Access Control Register (CCM_ACCESS_CTRL_ROOT35_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9200	Target Register (CCM_TARGET_ROOT36)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9204	Target Register (CCM_TARGET_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9208	Target Register (CCM_TARGET_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_920C	Target Register (CCM_TARGET_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9210	Miscellaneous Register (CCM_MISC36)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9214	Miscellaneous Register (CCM_MISC_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9218	Miscellaneous Register (CCM_MISC_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_921C	Miscellaneous Register (CCM_MISC_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9220	Post Divider Register (CCM_POST36)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9224	Post Divider Register (CCM_POST_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9228	Post Divider Register (CCM_POST_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_922C	Post Divider Register (CCM_POST_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9230	Pre Divider Register (CCM_PRE36)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9234	Pre Divider Register (CCM_PRE_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9238	Pre Divider Register (CCM_PRE_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_923C	Pre Divider Register (CCM_PRE_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9270	Access Control Register (CCM_ACCESS_CTRL36)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9274	Access Control Register (CCM_ACCESS_CTRL_ROOT36_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9278	Access Control Register (CCM_ACCESS_CTRL_ROOT36_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_927C	Access Control Register (CCM_ACCESS_CTRL_ROOT36_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9280	Target Register (CCM_TARGET_ROOT37)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9284	Target Register (CCM_TARGET_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9288	Target Register (CCM_TARGET_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_928C	Target Register (CCM_TARGET_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9290	Miscellaneous Register (CCM_MISC37)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9294	Miscellaneous Register (CCM_MISC_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9298	Miscellaneous Register (CCM_MISC_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_929C	Miscellaneous Register (CCM_MISC_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_92A0	Post Divider Register (CCM_POST37)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_92A4	Post Divider Register (CCM_POST_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_92A8	Post Divider Register (CCM_POST_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_92AC	Post Divider Register (CCM_POST_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_92B0	Pre Divider Register (CCM_PRE37)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_92B4	Pre Divider Register (CCM_PRE_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_92B8	Pre Divider Register (CCM_PRE_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_92BC	Pre Divider Register (CCM_PRE_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_92F0	Access Control Register (CCM_ACCESS_CTRL37)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_92F4	Access Control Register (CCM_ACCESS_CTRL_ROOT37_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_92F8	Access Control Register (CCM_ACCESS_CTRL_ROOT37_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_92FC	Access Control Register (CCM_ACCESS_CTRL_ROOT37_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9300	Target Register (CCM_TARGET_ROOT38)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9304	Target Register (CCM_TARGET_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9308	Target Register (CCM_TARGET_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_930C	Target Register (CCM_TARGET_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9310	Miscellaneous Register (CCM_MISC38)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9314	Miscellaneous Register (CCM_MISC_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9318	Miscellaneous Register (CCM_MISC_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_931C	Miscellaneous Register (CCM_MISC_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9320	Post Divider Register (CCM_POST38)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9324	Post Divider Register (CCM_POST_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9328	Post Divider Register (CCM_POST_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_932C	Post Divider Register (CCM_POST_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9330	Pre Divider Register (CCM_PRE38)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9334	Pre Divider Register (CCM_PRE_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9338	Pre Divider Register (CCM_PRE_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_933C	Pre Divider Register (CCM_PRE_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9370	Access Control Register (CCM_ACCESS_CTRL38)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9374	Access Control Register (CCM_ACCESS_CTRL_ROOT38_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9378	Access Control Register (CCM_ACCESS_CTRL_ROOT38_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_937C	Access Control Register (CCM_ACCESS_CTRL_ROOT38_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9380	Target Register (CCM_TARGET_ROOT39)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9384	Target Register (CCM_TARGET_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9388	Target Register (CCM_TARGET_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_938C	Target Register (CCM_TARGET_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9390	Miscellaneous Register (CCM_MISC39)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9394	Miscellaneous Register (CCM_MISC_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9398	Miscellaneous Register (CCM_MISC_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_939C	Miscellaneous Register (CCM_MISC_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_93A0	Post Divider Register (CCM_POST39)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_93A4	Post Divider Register (CCM_POST_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_93A8	Post Divider Register (CCM_POST_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_93AC	Post Divider Register (CCM_POST_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_93B0	Pre Divider Register (CCM_PRE39)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_93B4	Pre Divider Register (CCM_PRE_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_93B8	Pre Divider Register (CCM_PRE_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_93BC	Pre Divider Register (CCM_PRE_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_93F0	Access Control Register (CCM_ACCESS_CTRL39)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_93F4	Access Control Register (CCM_ACCESS_CTRL_ROOT39_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_93F8	Access Control Register (CCM_ACCESS_CTRL_ROOT39_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_93FC	Access Control Register (CCM_ACCESS_CTRL_ROOT39_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9400	Target Register (CCM_TARGET_ROOT40)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9404	Target Register (CCM_TARGET_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9408	Target Register (CCM_TARGET_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_940C	Target Register (CCM_TARGET_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9410	Miscellaneous Register (CCM_MISC40)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9414	Miscellaneous Register (CCM_MISC_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9418	Miscellaneous Register (CCM_MISC_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_941C	Miscellaneous Register (CCM_MISC_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9420	Post Divider Register (CCM_POST40)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9424	Post Divider Register (CCM_POST_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9428	Post Divider Register (CCM_POST_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_942C	Post Divider Register (CCM_POST_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9430	Pre Divider Register (CCM_PRE40)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9434	Pre Divider Register (CCM_PRE_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9438	Pre Divider Register (CCM_PRE_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_943C	Pre Divider Register (CCM_PRE_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9470	Access Control Register (CCM_ACCESS_CTRL40)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9474	Access Control Register (CCM_ACCESS_CTRL_ROOT40_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9478	Access Control Register (CCM_ACCESS_CTRL_ROOT40_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_947C	Access Control Register (CCM_ACCESS_CTRL_ROOT40_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9480	Target Register (CCM_TARGET_ROOT41)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9484	Target Register (CCM_TARGET_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9488	Target Register (CCM_TARGET_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_948C	Target Register (CCM_TARGET_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9490	Miscellaneous Register (CCM_MISC41)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9494	Miscellaneous Register (CCM_MISC_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9498	Miscellaneous Register (CCM_MISC_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_949C	Miscellaneous Register (CCM_MISC_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_94A0	Post Divider Register (CCM_POST41)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_94A4	Post Divider Register (CCM_POST_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_94A8	Post Divider Register (CCM_POST_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_94AC	Post Divider Register (CCM_POST_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_94B0	Pre Divider Register (CCM_PRE41)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_94B4	Pre Divider Register (CCM_PRE_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_94B8	Pre Divider Register (CCM_PRE_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_94BC	Pre Divider Register (CCM_PRE_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_94F0	Access Control Register (CCM_ACCESS_CTRL41)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_94F4	Access Control Register (CCM_ACCESS_CTRL_ROOT41_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_94F8	Access Control Register (CCM_ACCESS_CTRL_ROOT41_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_94FC	Access Control Register (CCM_ACCESS_CTRL_ROOT41_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9500	Target Register (CCM_TARGET_ROOT42)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9504	Target Register (CCM_TARGET_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9508	Target Register (CCM_TARGET_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_950C	Target Register (CCM_TARGET_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9510	Miscellaneous Register (CCM_MISC42)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9514	Miscellaneous Register (CCM_MISC_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9518	Miscellaneous Register (CCM_MISC_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_951C	Miscellaneous Register (CCM_MISC_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9520	Post Divider Register (CCM_POST42)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9524	Post Divider Register (CCM_POST_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9528	Post Divider Register (CCM_POST_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_952C	Post Divider Register (CCM_POST_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9530	Pre Divider Register (CCM_PRE42)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9534	Pre Divider Register (CCM_PRE_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9538	Pre Divider Register (CCM_PRE_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_953C	Pre Divider Register (CCM_PRE_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9570	Access Control Register (CCM_ACCESS_CTRL42)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9574	Access Control Register (CCM_ACCESS_CTRL_ROOT42_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9578	Access Control Register (CCM_ACCESS_CTRL_ROOT42_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_957C	Access Control Register (CCM_ACCESS_CTRL_ROOT42_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9580	Target Register (CCM_TARGET_ROOT43)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9584	Target Register (CCM_TARGET_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9588	Target Register (CCM_TARGET_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_958C	Target Register (CCM_TARGET_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9590	Miscellaneous Register (CCM_MISC43)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9594	Miscellaneous Register (CCM_MISC_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9598	Miscellaneous Register (CCM_MISC_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_959C	Miscellaneous Register (CCM_MISC_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_95A0	Post Divider Register (CCM_POST43)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_95A4	Post Divider Register (CCM_POST_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_95A8	Post Divider Register (CCM_POST_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_95AC	Post Divider Register (CCM_POST_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_95B0	Pre Divider Register (CCM_PRE43)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_95B4	Pre Divider Register (CCM_PRE_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_95B8	Pre Divider Register (CCM_PRE_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_95BC	Pre Divider Register (CCM_PRE_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_95F0	Access Control Register (CCM_ACCESS_CTRL43)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_95F4	Access Control Register (CCM_ACCESS_CTRL_ROOT43_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_95F8	Access Control Register (CCM_ACCESS_CTRL_ROOT43_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_95FC	Access Control Register (CCM_ACCESS_CTRL_ROOT43_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9600	Target Register (CCM_TARGET_ROOT44)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9604	Target Register (CCM_TARGET_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9608	Target Register (CCM_TARGET_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_960C	Target Register (CCM_TARGET_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9610	Miscellaneous Register (CCM_MISC44)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9614	Miscellaneous Register (CCM_MISC_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9618	Miscellaneous Register (CCM_MISC_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_961C	Miscellaneous Register (CCM_MISC_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9620	Post Divider Register (CCM_POST44)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9624	Post Divider Register (CCM_POST_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9628	Post Divider Register (CCM_POST_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_962C	Post Divider Register (CCM_POST_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9630	Pre Divider Register (CCM_PRE44)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9634	Pre Divider Register (CCM_PRE_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9638	Pre Divider Register (CCM_PRE_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_963C	Pre Divider Register (CCM_PRE_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9670	Access Control Register (CCM_ACCESS_CTRL44)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9674	Access Control Register (CCM_ACCESS_CTRL_ROOT44_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9678	Access Control Register (CCM_ACCESS_CTRL_ROOT44_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_967C	Access Control Register (CCM_ACCESS_CTRL_ROOT44_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9680	Target Register (CCM_TARGET_ROOT45)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9684	Target Register (CCM_TARGET_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9688	Target Register (CCM_TARGET_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_968C	Target Register (CCM_TARGET_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9690	Miscellaneous Register (CCM_MISC45)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9694	Miscellaneous Register (CCM_MISC_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9698	Miscellaneous Register (CCM_MISC_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_969C	Miscellaneous Register (CCM_MISC_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_96A0	Post Divider Register (CCM_POST45)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_96A4	Post Divider Register (CCM_POST_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_96A8	Post Divider Register (CCM_POST_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_96AC	Post Divider Register (CCM_POST_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_96B0	Pre Divider Register (CCM_PRE45)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_96B4	Pre Divider Register (CCM_PRE_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_96B8	Pre Divider Register (CCM_PRE_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_96BC	Pre Divider Register (CCM_PRE_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_96F0	Access Control Register (CCM_ACCESS_CTRL45)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_96F4	Access Control Register (CCM_ACCESS_CTRL_ROOT45_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_96F8	Access Control Register (CCM_ACCESS_CTRL_ROOT45_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_96FC	Access Control Register (CCM_ACCESS_CTRL_ROOT45_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9700	Target Register (CCM_TARGET_ROOT46)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9704	Target Register (CCM_TARGET_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9708	Target Register (CCM_TARGET_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_970C	Target Register (CCM_TARGET_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9710	Miscellaneous Register (CCM_MISC46)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9714	Miscellaneous Register (CCM_MISC_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9718	Miscellaneous Register (CCM_MISC_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_971C	Miscellaneous Register (CCM_MISC_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9720	Post Divider Register (CCM_POST46)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9724	Post Divider Register (CCM_POST_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9728	Post Divider Register (CCM_POST_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_972C	Post Divider Register (CCM_POST_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9730	Pre Divider Register (CCM_PRE46)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9734	Pre Divider Register (CCM_PRE_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9738	Pre Divider Register (CCM_PRE_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_973C	Pre Divider Register (CCM_PRE_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9770	Access Control Register (CCM_ACCESS_CTRL46)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9774	Access Control Register (CCM_ACCESS_CTRL_ROOT46_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9778	Access Control Register (CCM_ACCESS_CTRL_ROOT46_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_977C	Access Control Register (CCM_ACCESS_CTRL_ROOT46_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9780	Target Register (CCM_TARGET_ROOT47)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9784	Target Register (CCM_TARGET_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9788	Target Register (CCM_TARGET_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_978C	Target Register (CCM_TARGET_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9790	Miscellaneous Register (CCM_MISC47)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9794	Miscellaneous Register (CCM_MISC_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9798	Miscellaneous Register (CCM_MISC_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_979C	Miscellaneous Register (CCM_MISC_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_97A0	Post Divider Register (CCM_POST47)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_97A4	Post Divider Register (CCM_POST_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_97A8	Post Divider Register (CCM_POST_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_97AC	Post Divider Register (CCM_POST_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_97B0	Pre Divider Register (CCM_PRE47)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_97B4	Pre Divider Register (CCM_PRE_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_97B8	Pre Divider Register (CCM_PRE_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_97BC	Pre Divider Register (CCM_PRE_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_97F0	Access Control Register (CCM_ACCESS_CTRL47)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_97F4	Access Control Register (CCM_ACCESS_CTRL_ROOT47_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_97F8	Access Control Register (CCM_ACCESS_CTRL_ROOT47_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_97FC	Access Control Register (CCM_ACCESS_CTRL_ROOT47_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9800	Target Register (CCM_TARGET_ROOT48)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9804	Target Register (CCM_TARGET_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9808	Target Register (CCM_TARGET_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_980C	Target Register (CCM_TARGET_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9810	Miscellaneous Register (CCM_MISC48)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9814	Miscellaneous Register (CCM_MISC_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9818	Miscellaneous Register (CCM_MISC_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_981C	Miscellaneous Register (CCM_MISC_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9820	Post Divider Register (CCM_POST48)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9824	Post Divider Register (CCM_POST_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9828	Post Divider Register (CCM_POST_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_982C	Post Divider Register (CCM_POST_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9830	Pre Divider Register (CCM_PRE48)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9834	Pre Divider Register (CCM_PRE_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9838	Pre Divider Register (CCM_PRE_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_983C	Pre Divider Register (CCM_PRE_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9870	Access Control Register (CCM_ACCESS_CTRL48)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9874	Access Control Register (CCM_ACCESS_CTRL_ROOT48_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9878	Access Control Register (CCM_ACCESS_CTRL_ROOT48_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_987C	Access Control Register (CCM_ACCESS_CTRL_ROOT48_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9880	Target Register (CCM_TARGET_ROOT49)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9884	Target Register (CCM_TARGET_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9888	Target Register (CCM_TARGET_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_988C	Target Register (CCM_TARGET_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9890	Miscellaneous Register (CCM_MISC49)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9894	Miscellaneous Register (CCM_MISC_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9898	Miscellaneous Register (CCM_MISC_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_989C	Miscellaneous Register (CCM_MISC_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_98A0	Post Divider Register (CCM_POST49)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_98A4	Post Divider Register (CCM_POST_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_98A8	Post Divider Register (CCM_POST_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_98AC	Post Divider Register (CCM_POST_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_98B0	Pre Divider Register (CCM_PRE49)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_98B4	Pre Divider Register (CCM_PRE_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_98B8	Pre Divider Register (CCM_PRE_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_98BC	Pre Divider Register (CCM_PRE_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_98F0	Access Control Register (CCM_ACCESS_CTRL49)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_98F4	Access Control Register (CCM_ACCESS_CTRL_ROOT49_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_98F8	Access Control Register (CCM_ACCESS_CTRL_ROOT49_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_98FC	Access Control Register (CCM_ACCESS_CTRL_ROOT49_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9900	Target Register (CCM_TARGET_ROOT50)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9904	Target Register (CCM_TARGET_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9908	Target Register (CCM_TARGET_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_990C	Target Register (CCM_TARGET_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9910	Miscellaneous Register (CCM_MISC50)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9914	Miscellaneous Register (CCM_MISC_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9918	Miscellaneous Register (CCM_MISC_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_991C	Miscellaneous Register (CCM_MISC_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9920	Post Divider Register (CCM_POST50)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9924	Post Divider Register (CCM_POST_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9928	Post Divider Register (CCM_POST_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_992C	Post Divider Register (CCM_POST_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9930	Pre Divider Register (CCM_PRE50)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9934	Pre Divider Register (CCM_PRE_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9938	Pre Divider Register (CCM_PRE_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_993C	Pre Divider Register (CCM_PRE_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9970	Access Control Register (CCM_ACCESS_CTRL50)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9974	Access Control Register (CCM_ACCESS_CTRL_ROOT50_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9978	Access Control Register (CCM_ACCESS_CTRL_ROOT50_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_997C	Access Control Register (CCM_ACCESS_CTRL_ROOT50_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9980	Target Register (CCM_TARGET_ROOT51)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9984	Target Register (CCM_TARGET_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9988	Target Register (CCM_TARGET_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_998C	Target Register (CCM_TARGET_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9990	Miscellaneous Register (CCM_MISC51)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9994	Miscellaneous Register (CCM_MISC_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9998	Miscellaneous Register (CCM_MISC_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_999C	Miscellaneous Register (CCM_MISC_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_99A0	Post Divider Register (CCM_POST51)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_99A4	Post Divider Register (CCM_POST_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_99A8	Post Divider Register (CCM_POST_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_99AC	Post Divider Register (CCM_POST_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_99B0	Pre Divider Register (CCM_PRE51)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_99B4	Pre Divider Register (CCM_PRE_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_99B8	Pre Divider Register (CCM_PRE_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_99BC	Pre Divider Register (CCM_PRE_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_99F0	Access Control Register (CCM_ACCESS_CTRL51)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_99F4	Access Control Register (CCM_ACCESS_CTRL_ROOT51_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_99F8	Access Control Register (CCM_ACCESS_CTRL_ROOT51_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_99FC	Access Control Register (CCM_ACCESS_CTRL_ROOT51_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9A00	Target Register (CCM_TARGET_ROOT52)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9A04	Target Register (CCM_TARGET_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9A08	Target Register (CCM_TARGET_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9A0C	Target Register (CCM_TARGET_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9A10	Miscellaneous Register (CCM_MISC52)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9A14	Miscellaneous Register (CCM_MISC_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9A18	Miscellaneous Register (CCM_MISC_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9A1C	Miscellaneous Register (CCM_MISC_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9A20	Post Divider Register (CCM_POST52)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9A24	Post Divider Register (CCM_POST_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9A28	Post Divider Register (CCM_POST_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9A2C	Post Divider Register (CCM_POST_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9A30	Pre Divider Register (CCM_PRE52)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9A34	Pre Divider Register (CCM_PRE_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9A38	Pre Divider Register (CCM_PRE_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9A3C	Pre Divider Register (CCM_PRE_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9A70	Access Control Register (CCM_ACCESS_CTRL52)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9A74	Access Control Register (CCM_ACCESS_CTRL_ROOT52_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9A78	Access Control Register (CCM_ACCESS_CTRL_ROOT52_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9A7C	Access Control Register (CCM_ACCESS_CTRL_ROOT52_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9A80	Target Register (CCM_TARGET_ROOT53)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9A84	Target Register (CCM_TARGET_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9A88	Target Register (CCM_TARGET_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9A8C	Target Register (CCM_TARGET_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9A90	Miscellaneous Register (CCM_MISC53)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9A94	Miscellaneous Register (CCM_MISC_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9A98	Miscellaneous Register (CCM_MISC_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9A9C	Miscellaneous Register (CCM_MISC_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9AA0	Post Divider Register (CCM_POST53)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9AA4	Post Divider Register (CCM_POST_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9AA8	Post Divider Register (CCM_POST_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9AAC	Post Divider Register (CCM_POST_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9AB0	Pre Divider Register (CCM_PRE53)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9AB4	Pre Divider Register (CCM_PRE_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9AB8	Pre Divider Register (CCM_PRE_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9ABC	Pre Divider Register (CCM_PRE_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9AF0	Access Control Register (CCM_ACCESS_CTRL53)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9AF4	Access Control Register (CCM_ACCESS_CTRL_ROOT53_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9AF8	Access Control Register (CCM_ACCESS_CTRL_ROOT53_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9AFC	Access Control Register (CCM_ACCESS_CTRL_ROOT53_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9B00	Target Register (CCM_TARGET_ROOT54)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9B04	Target Register (CCM_TARGET_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9B08	Target Register (CCM_TARGET_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9B0C	Target Register (CCM_TARGET_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9B10	Miscellaneous Register (CCM_MISC54)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9B14	Miscellaneous Register (CCM_MISC_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9B18	Miscellaneous Register (CCM_MISC_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9B1C	Miscellaneous Register (CCM_MISC_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9B20	Post Divider Register (CCM_POST54)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9B24	Post Divider Register (CCM_POST_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9B28	Post Divider Register (CCM_POST_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9B2C	Post Divider Register (CCM_POST_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9B30	Pre Divider Register (CCM_PRE54)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9B34	Pre Divider Register (CCM_PRE_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9B38	Pre Divider Register (CCM_PRE_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9B3C	Pre Divider Register (CCM_PRE_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9B70	Access Control Register (CCM_ACCESS_CTRL54)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9B74	Access Control Register (CCM_ACCESS_CTRL_ROOT54_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9B78	Access Control Register (CCM_ACCESS_CTRL_ROOT54_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9B7C	Access Control Register (CCM_ACCESS_CTRL_ROOT54_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9B80	Target Register (CCM_TARGET_ROOT55)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9B84	Target Register (CCM_TARGET_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9B88	Target Register (CCM_TARGET_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9B8C	Target Register (CCM_TARGET_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9B90	Miscellaneous Register (CCM_MISC55)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9B94	Miscellaneous Register (CCM_MISC_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9B98	Miscellaneous Register (CCM_MISC_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9B9C	Miscellaneous Register (CCM_MISC_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9BA0	Post Divider Register (CCM_POST55)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9BA4	Post Divider Register (CCM_POST_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9BA8	Post Divider Register (CCM_POST_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9BAC	Post Divider Register (CCM_POST_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9BB0	Pre Divider Register (CCM_PRE55)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9BB4	Pre Divider Register (CCM_PRE_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9BB8	Pre Divider Register (CCM_PRE_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9BBC	Pre Divider Register (CCM_PRE_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9BF0	Access Control Register (CCM_ACCESS_CTRL55)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9BF4	Access Control Register (CCM_ACCESS_CTRL_ROOT55_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9BF8	Access Control Register (CCM_ACCESS_CTRL_ROOT55_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9BFC	Access Control Register (CCM_ACCESS_CTRL_ROOT55_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9C00	Target Register (CCM_TARGET_ROOT56)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9C04	Target Register (CCM_TARGET_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9C08	Target Register (CCM_TARGET_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9C0C	Target Register (CCM_TARGET_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9C10	Miscellaneous Register (CCM_MISC56)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9C14	Miscellaneous Register (CCM_MISC_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9C18	Miscellaneous Register (CCM_MISC_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9C1C	Miscellaneous Register (CCM_MISC_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9C20	Post Divider Register (CCM_POST56)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9C24	Post Divider Register (CCM_POST_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9C28	Post Divider Register (CCM_POST_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9C2C	Post Divider Register (CCM_POST_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9C30	Pre Divider Register (CCM_PRE56)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9C34	Pre Divider Register (CCM_PRE_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9C38	Pre Divider Register (CCM_PRE_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9C3C	Pre Divider Register (CCM_PRE_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9C70	Access Control Register (CCM_ACCESS_CTRL56)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9C74	Access Control Register (CCM_ACCESS_CTRL_ROOT56_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9C78	Access Control Register (CCM_ACCESS_CTRL_ROOT56_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9C7C	Access Control Register (CCM_ACCESS_CTRL_ROOT56_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9C80	Target Register (CCM_TARGET_ROOT57)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9C84	Target Register (CCM_TARGET_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9C88	Target Register (CCM_TARGET_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9C8C	Target Register (CCM_TARGET_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9C90	Miscellaneous Register (CCM_MISC57)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9C94	Miscellaneous Register (CCM_MISC_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9C98	Miscellaneous Register (CCM_MISC_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9C9C	Miscellaneous Register (CCM_MISC_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9CA0	Post Divider Register (CCM_POST57)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9CA4	Post Divider Register (CCM_POST_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9CA8	Post Divider Register (CCM_POST_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9CAC	Post Divider Register (CCM_POST_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9CB0	Pre Divider Register (CCM_PRE57)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9CB4	Pre Divider Register (CCM_PRE_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9CB8	Pre Divider Register (CCM_PRE_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9CBC	Pre Divider Register (CCM_PRE_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9CF0	Access Control Register (CCM_ACCESS_CTRL57)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9CF4	Access Control Register (CCM_ACCESS_CTRL_ROOT57_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9CF8	Access Control Register (CCM_ACCESS_CTRL_ROOT57_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9CFC	Access Control Register (CCM_ACCESS_CTRL_ROOT57_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9D00	Target Register (CCM_TARGET_ROOT58)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9D04	Target Register (CCM_TARGET_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9D08	Target Register (CCM_TARGET_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9D0C	Target Register (CCM_TARGET_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9D10	Miscellaneous Register (CCM_MISC58)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9D14	Miscellaneous Register (CCM_MISC_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9D18	Miscellaneous Register (CCM_MISC_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9D1C	Miscellaneous Register (CCM_MISC_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9D20	Post Divider Register (CCM_POST58)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9D24	Post Divider Register (CCM_POST_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9D28	Post Divider Register (CCM_POST_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9D2C	Post Divider Register (CCM_POST_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9D30	Pre Divider Register (CCM_PRE58)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9D34	Pre Divider Register (CCM_PRE_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9D38	Pre Divider Register (CCM_PRE_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9D3C	Pre Divider Register (CCM_PRE_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9D70	Access Control Register (CCM_ACCESS_CTRL58)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9D74	Access Control Register (CCM_ACCESS_CTRL_ROOT58_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9D78	Access Control Register (CCM_ACCESS_CTRL_ROOT58_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9D7C	Access Control Register (CCM_ACCESS_CTRL_ROOT58_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9D80	Target Register (CCM_TARGET_ROOT59)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9D84	Target Register (CCM_TARGET_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9D88	Target Register (CCM_TARGET_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9D8C	Target Register (CCM_TARGET_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9D90	Miscellaneous Register (CCM_MISC59)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9D94	Miscellaneous Register (CCM_MISC_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9D98	Miscellaneous Register (CCM_MISC_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9D9C	Miscellaneous Register (CCM_MISC_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9DA0	Post Divider Register (CCM_POST59)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9DA4	Post Divider Register (CCM_POST_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9DA8	Post Divider Register (CCM_POST_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9DAC	Post Divider Register (CCM_POST_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9DB0	Pre Divider Register (CCM_PRE59)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9DB4	Pre Divider Register (CCM_PRE_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9DB8	Pre Divider Register (CCM_PRE_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9DBC	Pre Divider Register (CCM_PRE_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9DF0	Access Control Register (CCM_ACCESS_CTRL59)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9DF4	Access Control Register (CCM_ACCESS_CTRL_ROOT59_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9DF8	Access Control Register (CCM_ACCESS_CTRL_ROOT59_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9DFC	Access Control Register (CCM_ACCESS_CTRL_ROOT59_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9E00	Target Register (CCM_TARGET_ROOT60)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9E04	Target Register (CCM_TARGET_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9E08	Target Register (CCM_TARGET_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9E0C	Target Register (CCM_TARGET_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9E10	Miscellaneous Register (CCM_MISC60)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9E14	Miscellaneous Register (CCM_MISC_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9E18	Miscellaneous Register (CCM_MISC_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9E1C	Miscellaneous Register (CCM_MISC_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9E20	Post Divider Register (CCM_POST60)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9E24	Post Divider Register (CCM_POST_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9E28	Post Divider Register (CCM_POST_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9E2C	Post Divider Register (CCM_POST_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9E30	Pre Divider Register (CCM_PRE60)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9E34	Pre Divider Register (CCM_PRE_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9E38	Pre Divider Register (CCM_PRE_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9E3C	Pre Divider Register (CCM_PRE_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9E70	Access Control Register (CCM_ACCESS_CTRL60)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9E74	Access Control Register (CCM_ACCESS_CTRL_ROOT60_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9E78	Access Control Register (CCM_ACCESS_CTRL_ROOT60_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9E7C	Access Control Register (CCM_ACCESS_CTRL_ROOT60_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9E80	Target Register (CCM_TARGET_ROOT61)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9E84	Target Register (CCM_TARGET_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9E88	Target Register (CCM_TARGET_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9E8C	Target Register (CCM_TARGET_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9E90	Miscellaneous Register (CCM_MISC61)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9E94	Miscellaneous Register (CCM_MISC_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9E98	Miscellaneous Register (CCM_MISC_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9E9C	Miscellaneous Register (CCM_MISC_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9EA0	Post Divider Register (CCM_POST61)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9EA4	Post Divider Register (CCM_POST_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9EA8	Post Divider Register (CCM_POST_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9EAC	Post Divider Register (CCM_POST_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9EB0	Pre Divider Register (CCM_PRE61)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9EB4	Pre Divider Register (CCM_PRE_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9EB8	Pre Divider Register (CCM_PRE_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9EBC	Pre Divider Register (CCM_PRE_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9EF0	Access Control Register (CCM_ACCESS_CTRL61)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9EF4	Access Control Register (CCM_ACCESS_CTRL_ROOT61_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9EF8	Access Control Register (CCM_ACCESS_CTRL_ROOT61_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9EFC	Access Control Register (CCM_ACCESS_CTRL_ROOT61_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9F00	Target Register (CCM_TARGET_ROOT62)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9F04	Target Register (CCM_TARGET_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9F08	Target Register (CCM_TARGET_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9F0C	Target Register (CCM_TARGET_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9F10	Miscellaneous Register (CCM_MISC62)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9F14	Miscellaneous Register (CCM_MISC_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9F18	Miscellaneous Register (CCM_MISC_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9F1C	Miscellaneous Register (CCM_MISC_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9F20	Post Divider Register (CCM_POST62)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_9F24	Post Divider Register (CCM_POST_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9F28	Post Divider Register (CCM_POST_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9F2C	Post Divider Register (CCM_POST_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9F30	Pre Divider Register (CCM_PRE62)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9F34	Pre Divider Register (CCM_PRE_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9F38	Pre Divider Register (CCM_PRE_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9F3C	Pre Divider Register (CCM_PRE_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9F70	Access Control Register (CCM_ACCESS_CTRL62)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9F74	Access Control Register (CCM_ACCESS_CTRL_ROOT62_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9F78	Access Control Register (CCM_ACCESS_CTRL_ROOT62_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9F7C	Access Control Register (CCM_ACCESS_CTRL_ROOT62_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_9F80	Target Register (CCM_TARGET_ROOT63)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_9F84	Target Register (CCM_TARGET_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_9F88	Target Register (CCM_TARGET_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_9F8C	Target Register (CCM_TARGET_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_9F90	Miscellaneous Register (CCM_MISC63)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_9F94	Miscellaneous Register (CCM_MISC_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_9F98	Miscellaneous Register (CCM_MISC_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_9F9C	Miscellaneous Register (CCM_MISC_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_9FA0	Post Divider Register (CCM_POST63)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_9FA4	Post Divider Register (CCM_POST_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_9FA8	Post Divider Register (CCM_POST_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_9FAC	Post Divider Register (CCM_POST_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_9FB0	Pre Divider Register (CCM_PRE63)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_9FB4	Pre Divider Register (CCM_PRE_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_9FB8	Pre Divider Register (CCM_PRE_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_9FBC	Pre Divider Register (CCM_PRE_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_9FF0	Access Control Register (CCM_ACCESS_CTRL63)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_9FF4	Access Control Register (CCM_ACCESS_CTRL_ROOT63_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_9FF8	Access Control Register (CCM_ACCESS_CTRL_ROOT63_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_9FFC	Access Control Register (CCM_ACCESS_CTRL_ROOT63_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A000	Target Register (CCM_TARGET_ROOT64)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A004	Target Register (CCM_TARGET_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A008	Target Register (CCM_TARGET_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A00C	Target Register (CCM_TARGET_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A010	Miscellaneous Register (CCM_MISC64)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A014	Miscellaneous Register (CCM_MISC_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A018	Miscellaneous Register (CCM_MISC_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A01C	Miscellaneous Register (CCM_MISC_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A020	Post Divider Register (CCM_POST64)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A024	Post Divider Register (CCM_POST_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A028	Post Divider Register (CCM_POST_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A02C	Post Divider Register (CCM_POST_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A030	Pre Divider Register (CCM_PRE64)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A034	Pre Divider Register (CCM_PRE_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A038	Pre Divider Register (CCM_PRE_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A03C	Pre Divider Register (CCM_PRE_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A070	Access Control Register (CCM_ACCESS_CTRL64)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A074	Access Control Register (CCM_ACCESS_CTRL_ROOT64_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A078	Access Control Register (CCM_ACCESS_CTRL_ROOT64_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A07C	Access Control Register (CCM_ACCESS_CTRL_ROOT64_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A080	Target Register (CCM_TARGET_ROOT65)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A084	Target Register (CCM_TARGET_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A088	Target Register (CCM_TARGET_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A08C	Target Register (CCM_TARGET_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A090	Miscellaneous Register (CCM_MISC65)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A094	Miscellaneous Register (CCM_MISC_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A098	Miscellaneous Register (CCM_MISC_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A09C	Miscellaneous Register (CCM_MISC_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A0A0	Post Divider Register (CCM_POST65)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A0A4	Post Divider Register (CCM_POST_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A0A8	Post Divider Register (CCM_POST_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A0AC	Post Divider Register (CCM_POST_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A0B0	Pre Divider Register (CCM_PRE65)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A0B4	Pre Divider Register (CCM_PRE_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A0B8	Pre Divider Register (CCM_PRE_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A0BC	Pre Divider Register (CCM_PRE_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A0F0	Access Control Register (CCM_ACCESS_CTRL65)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT65_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT65_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT65_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A100	Target Register (CCM_TARGET_ROOT66)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A104	Target Register (CCM_TARGET_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A108	Target Register (CCM_TARGET_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A10C	Target Register (CCM_TARGET_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A110	Miscellaneous Register (CCM_MISC66)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A114	Miscellaneous Register (CCM_MISC_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A118	Miscellaneous Register (CCM_MISC_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A11C	Miscellaneous Register (CCM_MISC_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A120	Post Divider Register (CCM_POST66)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A124	Post Divider Register (CCM_POST_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A128	Post Divider Register (CCM_POST_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A12C	Post Divider Register (CCM_POST_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A130	Pre Divider Register (CCM_PRE66)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A134	Pre Divider Register (CCM_PRE_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A138	Pre Divider Register (CCM_PRE_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A13C	Pre Divider Register (CCM_PRE_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A170	Access Control Register (CCM_ACCESS_CTRL66)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A174	Access Control Register (CCM_ACCESS_CTRL_ROOT66_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A178	Access Control Register (CCM_ACCESS_CTRL_ROOT66_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A17C	Access Control Register (CCM_ACCESS_CTRL_ROOT66_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A180	Target Register (CCM_TARGET_ROOT67)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A184	Target Register (CCM_TARGET_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A188	Target Register (CCM_TARGET_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A18C	Target Register (CCM_TARGET_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A190	Miscellaneous Register (CCM_MISC67)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A194	Miscellaneous Register (CCM_MISC_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A198	Miscellaneous Register (CCM_MISC_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A19C	Miscellaneous Register (CCM_MISC_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A1A0	Post Divider Register (CCM_POST67)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A1A4	Post Divider Register (CCM_POST_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A1A8	Post Divider Register (CCM_POST_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A1AC	Post Divider Register (CCM_POST_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A1B0	Pre Divider Register (CCM_PRE67)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A1B4	Pre Divider Register (CCM_PRE_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A1B8	Pre Divider Register (CCM_PRE_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A1BC	Pre Divider Register (CCM_PRE_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A1F0	Access Control Register (CCM_ACCESS_CTRL67)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT67_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT67_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT67_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A200	Target Register (CCM_TARGET_ROOT68)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A204	Target Register (CCM_TARGET_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A208	Target Register (CCM_TARGET_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A20C	Target Register (CCM_TARGET_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A210	Miscellaneous Register (CCM_MISC68)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A214	Miscellaneous Register (CCM_MISC_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A218	Miscellaneous Register (CCM_MISC_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A21C	Miscellaneous Register (CCM_MISC_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A220	Post Divider Register (CCM_POST68)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A224	Post Divider Register (CCM_POST_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A228	Post Divider Register (CCM_POST_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A22C	Post Divider Register (CCM_POST_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A230	Pre Divider Register (CCM_PRE68)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A234	Pre Divider Register (CCM_PRE_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A238	Pre Divider Register (CCM_PRE_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A23C	Pre Divider Register (CCM_PRE_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A270	Access Control Register (CCM_ACCESS_CTRL68)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A274	Access Control Register (CCM_ACCESS_CTRL_ROOT68_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A278	Access Control Register (CCM_ACCESS_CTRL_ROOT68_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A27C	Access Control Register (CCM_ACCESS_CTRL_ROOT68_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A280	Target Register (CCM_TARGET_ROOT69)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A284	Target Register (CCM_TARGET_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A288	Target Register (CCM_TARGET_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A28C	Target Register (CCM_TARGET_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A290	Miscellaneous Register (CCM_MISC69)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A294	Miscellaneous Register (CCM_MISC_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A298	Miscellaneous Register (CCM_MISC_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A29C	Miscellaneous Register (CCM_MISC_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A2A0	Post Divider Register (CCM_POST69)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A2A4	Post Divider Register (CCM_POST_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A2A8	Post Divider Register (CCM_POST_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A2AC	Post Divider Register (CCM_POST_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A2B0	Pre Divider Register (CCM_PRE69)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A2B4	Pre Divider Register (CCM_PRE_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A2B8	Pre Divider Register (CCM_PRE_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A2BC	Pre Divider Register (CCM_PRE_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A2F0	Access Control Register (CCM_ACCESS_CTRL69)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT69_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT69_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT69_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A300	Target Register (CCM_TARGET_ROOT70)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A304	Target Register (CCM_TARGET_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A308	Target Register (CCM_TARGET_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A30C	Target Register (CCM_TARGET_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A310	Miscellaneous Register (CCM_MISC70)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A314	Miscellaneous Register (CCM_MISC_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A318	Miscellaneous Register (CCM_MISC_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A31C	Miscellaneous Register (CCM_MISC_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A320	Post Divider Register (CCM_POST70)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A324	Post Divider Register (CCM_POST_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A328	Post Divider Register (CCM_POST_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A32C	Post Divider Register (CCM_POST_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A330	Pre Divider Register (CCM_PRE70)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A334	Pre Divider Register (CCM_PRE_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A338	Pre Divider Register (CCM_PRE_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A33C	Pre Divider Register (CCM_PRE_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A370	Access Control Register (CCM_ACCESS_CTRL70)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A374	Access Control Register (CCM_ACCESS_CTRL_ROOT70_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A378	Access Control Register (CCM_ACCESS_CTRL_ROOT70_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A37C	Access Control Register (CCM_ACCESS_CTRL_ROOT70_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A380	Target Register (CCM_TARGET_ROOT71)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A384	Target Register (CCM_TARGET_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A388	Target Register (CCM_TARGET_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A38C	Target Register (CCM_TARGET_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A390	Miscellaneous Register (CCM_MISC71)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A394	Miscellaneous Register (CCM_MISC_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A398	Miscellaneous Register (CCM_MISC_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A39C	Miscellaneous Register (CCM_MISC_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A3A0	Post Divider Register (CCM_POST71)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A3A4	Post Divider Register (CCM_POST_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A3A8	Post Divider Register (CCM_POST_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A3AC	Post Divider Register (CCM_POST_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A3B0	Pre Divider Register (CCM_PRE71)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A3B4	Pre Divider Register (CCM_PRE_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A3B8	Pre Divider Register (CCM_PRE_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A3BC	Pre Divider Register (CCM_PRE_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A3F0	Access Control Register (CCM_ACCESS_CTRL71)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT71_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT71_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT71_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A400	Target Register (CCM_TARGET_ROOT72)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A404	Target Register (CCM_TARGET_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A408	Target Register (CCM_TARGET_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A40C	Target Register (CCM_TARGET_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A410	Miscellaneous Register (CCM_MISC72)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A414	Miscellaneous Register (CCM_MISC_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A418	Miscellaneous Register (CCM_MISC_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A41C	Miscellaneous Register (CCM_MISC_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A420	Post Divider Register (CCM_POST72)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A424	Post Divider Register (CCM_POST_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A428	Post Divider Register (CCM_POST_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A42C	Post Divider Register (CCM_POST_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A430	Pre Divider Register (CCM_PRE72)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A434	Pre Divider Register (CCM_PRE_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A438	Pre Divider Register (CCM_PRE_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A43C	Pre Divider Register (CCM_PRE_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A470	Access Control Register (CCM_ACCESS_CTRL72)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A474	Access Control Register (CCM_ACCESS_CTRL_ROOT72_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A478	Access Control Register (CCM_ACCESS_CTRL_ROOT72_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A47C	Access Control Register (CCM_ACCESS_CTRL_ROOT72_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A480	Target Register (CCM_TARGET_ROOT73)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A484	Target Register (CCM_TARGET_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A488	Target Register (CCM_TARGET_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A48C	Target Register (CCM_TARGET_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A490	Miscellaneous Register (CCM_MISC73)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A494	Miscellaneous Register (CCM_MISC_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A498	Miscellaneous Register (CCM_MISC_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A49C	Miscellaneous Register (CCM_MISC_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A4A0	Post Divider Register (CCM_POST73)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A4A4	Post Divider Register (CCM_POST_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A4A8	Post Divider Register (CCM_POST_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A4AC	Post Divider Register (CCM_POST_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A4B0	Pre Divider Register (CCM_PRE73)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A4B4	Pre Divider Register (CCM_PRE_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A4B8	Pre Divider Register (CCM_PRE_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A4BC	Pre Divider Register (CCM_PRE_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A4F0	Access Control Register (CCM_ACCESS_CTRL73)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT73_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT73_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT73_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A500	Target Register (CCM_TARGET_ROOT74)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A504	Target Register (CCM_TARGET_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A508	Target Register (CCM_TARGET_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A50C	Target Register (CCM_TARGET_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A510	Miscellaneous Register (CCM_MISC74)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A514	Miscellaneous Register (CCM_MISC_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A518	Miscellaneous Register (CCM_MISC_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A51C	Miscellaneous Register (CCM_MISC_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A520	Post Divider Register (CCM_POST74)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A524	Post Divider Register (CCM_POST_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A528	Post Divider Register (CCM_POST_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A52C	Post Divider Register (CCM_POST_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A530	Pre Divider Register (CCM_PRE74)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A534	Pre Divider Register (CCM_PRE_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A538	Pre Divider Register (CCM_PRE_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A53C	Pre Divider Register (CCM_PRE_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A570	Access Control Register (CCM_ACCESS_CTRL74)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A574	Access Control Register (CCM_ACCESS_CTRL_ROOT74_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A578	Access Control Register (CCM_ACCESS_CTRL_ROOT74_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A57C	Access Control Register (CCM_ACCESS_CTRL_ROOT74_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A580	Target Register (CCM_TARGET_ROOT75)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A584	Target Register (CCM_TARGET_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A588	Target Register (CCM_TARGET_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A58C	Target Register (CCM_TARGET_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A590	Miscellaneous Register (CCM_MISC75)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A594	Miscellaneous Register (CCM_MISC_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A598	Miscellaneous Register (CCM_MISC_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A59C	Miscellaneous Register (CCM_MISC_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A5A0	Post Divider Register (CCM_POST75)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A5A4	Post Divider Register (CCM_POST_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A5A8	Post Divider Register (CCM_POST_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A5AC	Post Divider Register (CCM_POST_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A5B0	Pre Divider Register (CCM_PRE75)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A5B4	Pre Divider Register (CCM_PRE_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A5B8	Pre Divider Register (CCM_PRE_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A5BC	Pre Divider Register (CCM_PRE_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A5F0	Access Control Register (CCM_ACCESS_CTRL75)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT75_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT75_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT75_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A600	Target Register (CCM_TARGET_ROOT76)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A604	Target Register (CCM_TARGET_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A608	Target Register (CCM_TARGET_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A60C	Target Register (CCM_TARGET_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A610	Miscellaneous Register (CCM_MISC76)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A614	Miscellaneous Register (CCM_MISC_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A618	Miscellaneous Register (CCM_MISC_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A61C	Miscellaneous Register (CCM_MISC_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A620	Post Divider Register (CCM_POST76)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A624	Post Divider Register (CCM_POST_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A628	Post Divider Register (CCM_POST_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A62C	Post Divider Register (CCM_POST_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A630	Pre Divider Register (CCM_PRE76)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A634	Pre Divider Register (CCM_PRE_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A638	Pre Divider Register (CCM_PRE_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A63C	Pre Divider Register (CCM_PRE_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A670	Access Control Register (CCM_ACCESS_CTRL76)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A674	Access Control Register (CCM_ACCESS_CTRL_ROOT76_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A678	Access Control Register (CCM_ACCESS_CTRL_ROOT76_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A67C	Access Control Register (CCM_ACCESS_CTRL_ROOT76_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A680	Target Register (CCM_TARGET_ROOT77)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A684	Target Register (CCM_TARGET_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A688	Target Register (CCM_TARGET_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A68C	Target Register (CCM_TARGET_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A690	Miscellaneous Register (CCM_MISC77)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A694	Miscellaneous Register (CCM_MISC_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A698	Miscellaneous Register (CCM_MISC_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A69C	Miscellaneous Register (CCM_MISC_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A6A0	Post Divider Register (CCM_POST77)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A6A4	Post Divider Register (CCM_POST_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A6A8	Post Divider Register (CCM_POST_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A6AC	Post Divider Register (CCM_POST_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A6B0	Pre Divider Register (CCM_PRE77)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A6B4	Pre Divider Register (CCM_PRE_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A6B8	Pre Divider Register (CCM_PRE_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A6BC	Pre Divider Register (CCM_PRE_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A6F0	Access Control Register (CCM_ACCESS_CTRL77)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT77_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT77_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT77_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A700	Target Register (CCM_TARGET_ROOT78)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A704	Target Register (CCM_TARGET_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A708	Target Register (CCM_TARGET_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A70C	Target Register (CCM_TARGET_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A710	Miscellaneous Register (CCM_MISC78)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A714	Miscellaneous Register (CCM_MISC_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A718	Miscellaneous Register (CCM_MISC_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A71C	Miscellaneous Register (CCM_MISC_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A720	Post Divider Register (CCM_POST78)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A724	Post Divider Register (CCM_POST_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A728	Post Divider Register (CCM_POST_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A72C	Post Divider Register (CCM_POST_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A730	Pre Divider Register (CCM_PRE78)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A734	Pre Divider Register (CCM_PRE_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A738	Pre Divider Register (CCM_PRE_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A73C	Pre Divider Register (CCM_PRE_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A770	Access Control Register (CCM_ACCESS_CTRL78)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A774	Access Control Register (CCM_ACCESS_CTRL_ROOT78_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A778	Access Control Register (CCM_ACCESS_CTRL_ROOT78_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A77C	Access Control Register (CCM_ACCESS_CTRL_ROOT78_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A780	Target Register (CCM_TARGET_ROOT79)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A784	Target Register (CCM_TARGET_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A788	Target Register (CCM_TARGET_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A78C	Target Register (CCM_TARGET_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A790	Miscellaneous Register (CCM_MISC79)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A794	Miscellaneous Register (CCM_MISC_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A798	Miscellaneous Register (CCM_MISC_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A79C	Miscellaneous Register (CCM_MISC_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A7A0	Post Divider Register (CCM_POST79)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A7A4	Post Divider Register (CCM_POST_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A7A8	Post Divider Register (CCM_POST_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A7AC	Post Divider Register (CCM_POST_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A7B0	Pre Divider Register (CCM_PRE79)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A7B4	Pre Divider Register (CCM_PRE_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A7B8	Pre Divider Register (CCM_PRE_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A7BC	Pre Divider Register (CCM_PRE_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A7F0	Access Control Register (CCM_ACCESS_CTRL79)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A7F4	Access Control Register (CCM_ACCESS_CTRL_ROOT79_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A7F8	Access Control Register (CCM_ACCESS_CTRL_ROOT79_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A7FC	Access Control Register (CCM_ACCESS_CTRL_ROOT79_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A800	Target Register (CCM_TARGET_ROOT80)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A804	Target Register (CCM_TARGET_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A808	Target Register (CCM_TARGET_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A80C	Target Register (CCM_TARGET_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A810	Miscellaneous Register (CCM_MISC80)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A814	Miscellaneous Register (CCM_MISC_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A818	Miscellaneous Register (CCM_MISC_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A81C	Miscellaneous Register (CCM_MISC_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A820	Post Divider Register (CCM_POST80)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A824	Post Divider Register (CCM_POST_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A828	Post Divider Register (CCM_POST_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A82C	Post Divider Register (CCM_POST_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A830	Pre Divider Register (CCM_PRE80)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A834	Pre Divider Register (CCM_PRE_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A838	Pre Divider Register (CCM_PRE_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A83C	Pre Divider Register (CCM_PRE_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A870	Access Control Register (CCM_ACCESS_CTRL80)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A874	Access Control Register (CCM_ACCESS_CTRL_ROOT80_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A878	Access Control Register (CCM_ACCESS_CTRL_ROOT80_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A87C	Access Control Register (CCM_ACCESS_CTRL_ROOT80_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A880	Target Register (CCM_TARGET_ROOT81)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A884	Target Register (CCM_TARGET_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A888	Target Register (CCM_TARGET_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A88C	Target Register (CCM_TARGET_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A890	Miscellaneous Register (CCM_MISC81)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A894	Miscellaneous Register (CCM_MISC_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A898	Miscellaneous Register (CCM_MISC_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A89C	Miscellaneous Register (CCM_MISC_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A8A0	Post Divider Register (CCM_POST81)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A8A4	Post Divider Register (CCM_POST_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A8A8	Post Divider Register (CCM_POST_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A8AC	Post Divider Register (CCM_POST_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A8B0	Pre Divider Register (CCM_PRE81)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A8B4	Pre Divider Register (CCM_PRE_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A8B8	Pre Divider Register (CCM_PRE_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A8BC	Pre Divider Register (CCM_PRE_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A8F0	Access Control Register (CCM_ACCESS_CTRL81)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A8F4	Access Control Register (CCM_ACCESS_CTRL_ROOT81_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A8F8	Access Control Register (CCM_ACCESS_CTRL_ROOT81_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A8FC	Access Control Register (CCM_ACCESS_CTRL_ROOT81_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A900	Target Register (CCM_TARGET_ROOT82)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A904	Target Register (CCM_TARGET_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A908	Target Register (CCM_TARGET_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A90C	Target Register (CCM_TARGET_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A910	Miscellaneous Register (CCM_MISC82)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_A914	Miscellaneous Register (CCM_MISC_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A918	Miscellaneous Register (CCM_MISC_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A91C	Miscellaneous Register (CCM_MISC_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A920	Post Divider Register (CCM_POST82)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A924	Post Divider Register (CCM_POST_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A928	Post Divider Register (CCM_POST_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A92C	Post Divider Register (CCM_POST_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A930	Pre Divider Register (CCM_PRE82)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A934	Pre Divider Register (CCM_PRE_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A938	Pre Divider Register (CCM_PRE_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A93C	Pre Divider Register (CCM_PRE_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A970	Access Control Register (CCM_ACCESS_CTRL82)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A974	Access Control Register (CCM_ACCESS_CTRL_ROOT82_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A978	Access Control Register (CCM_ACCESS_CTRL_ROOT82_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A97C	Access Control Register (CCM_ACCESS_CTRL_ROOT82_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_A980	Target Register (CCM_TARGET_ROOT83)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_A984	Target Register (CCM_TARGET_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_A988	Target Register (CCM_TARGET_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_A98C	Target Register (CCM_TARGET_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_A990	Miscellaneous Register (CCM_MISC83)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_A994	Miscellaneous Register (CCM_MISC_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_A998	Miscellaneous Register (CCM_MISC_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_A99C	Miscellaneous Register (CCM_MISC_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_A9A0	Post Divider Register (CCM_POST83)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_A9A4	Post Divider Register (CCM_POST_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_A9A8	Post Divider Register (CCM_POST_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_A9AC	Post Divider Register (CCM_POST_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_A9B0	Pre Divider Register (CCM_PRE83)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_A9B4	Pre Divider Register (CCM_PRE_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_A9B8	Pre Divider Register (CCM_PRE_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_A9BC	Pre Divider Register (CCM_PRE_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_A9F0	Access Control Register (CCM_ACCESS_CTRL83)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_A9F4	Access Control Register (CCM_ACCESS_CTRL_ROOT83_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_A9F8	Access Control Register (CCM_ACCESS_CTRL_ROOT83_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_A9FC	Access Control Register (CCM_ACCESS_CTRL_ROOT83_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AA00	Target Register (CCM_TARGET_ROOT84)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AA04	Target Register (CCM_TARGET_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AA08	Target Register (CCM_TARGET_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AA0C	Target Register (CCM_TARGET_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AA10	Miscellaneous Register (CCM_MISC84)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AA14	Miscellaneous Register (CCM_MISC_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AA18	Miscellaneous Register (CCM_MISC_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AA1C	Miscellaneous Register (CCM_MISC_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AA20	Post Divider Register (CCM_POST84)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AA24	Post Divider Register (CCM_POST_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AA28	Post Divider Register (CCM_POST_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AA2C	Post Divider Register (CCM_POST_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AA30	Pre Divider Register (CCM_PRE84)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AA34	Pre Divider Register (CCM_PRE_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AA38	Pre Divider Register (CCM_PRE_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AA3C	Pre Divider Register (CCM_PRE_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AA70	Access Control Register (CCM_ACCESS_CTRL84)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AA74	Access Control Register (CCM_ACCESS_CTRL_ROOT84_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AA78	Access Control Register (CCM_ACCESS_CTRL_ROOT84_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AA7C	Access Control Register (CCM_ACCESS_CTRL_ROOT84_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AA80	Target Register (CCM_TARGET_ROOT85)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AA84	Target Register (CCM_TARGET_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AA88	Target Register (CCM_TARGET_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AA8C	Target Register (CCM_TARGET_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AA90	Miscellaneous Register (CCM_MISC85)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AA94	Miscellaneous Register (CCM_MISC_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AA98	Miscellaneous Register (CCM_MISC_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AA9C	Miscellaneous Register (CCM_MISC_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AAA0	Post Divider Register (CCM_POST85)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AAA4	Post Divider Register (CCM_POST_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AAA8	Post Divider Register (CCM_POST_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AAAC	Post Divider Register (CCM_POST_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AAB0	Pre Divider Register (CCM_PRE85)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AAB4	Pre Divider Register (CCM_PRE_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AAB8	Pre Divider Register (CCM_PRE_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AABC	Pre Divider Register (CCM_PRE_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AAF0	Access Control Register (CCM_ACCESS_CTRL85)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AAF4	Access Control Register (CCM_ACCESS_CTRL_ROOT85_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AAF8	Access Control Register (CCM_ACCESS_CTRL_ROOT85_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AAFC	Access Control Register (CCM_ACCESS_CTRL_ROOT85_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AB00	Target Register (CCM_TARGET_ROOT86)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AB04	Target Register (CCM_TARGET_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AB08	Target Register (CCM_TARGET_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AB0C	Target Register (CCM_TARGET_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AB10	Miscellaneous Register (CCM_MISC86)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AB14	Miscellaneous Register (CCM_MISC_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AB18	Miscellaneous Register (CCM_MISC_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AB1C	Miscellaneous Register (CCM_MISC_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AB20	Post Divider Register (CCM_POST86)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AB24	Post Divider Register (CCM_POST_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AB28	Post Divider Register (CCM_POST_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AB2C	Post Divider Register (CCM_POST_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AB30	Pre Divider Register (CCM_PRE86)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AB34	Pre Divider Register (CCM_PRE_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AB38	Pre Divider Register (CCM_PRE_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AB3C	Pre Divider Register (CCM_PRE_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AB70	Access Control Register (CCM_ACCESS_CTRL86)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AB74	Access Control Register (CCM_ACCESS_CTRL_ROOT86_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AB78	Access Control Register (CCM_ACCESS_CTRL_ROOT86_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AB7C	Access Control Register (CCM_ACCESS_CTRL_ROOT86_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AB80	Target Register (CCM_TARGET_ROOT87)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AB84	Target Register (CCM_TARGET_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AB88	Target Register (CCM_TARGET_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AB8C	Target Register (CCM_TARGET_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AB90	Miscellaneous Register (CCM_MISC87)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AB94	Miscellaneous Register (CCM_MISC_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AB98	Miscellaneous Register (CCM_MISC_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AB9C	Miscellaneous Register (CCM_MISC_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_ABA0	Post Divider Register (CCM_POST87)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_ABA4	Post Divider Register (CCM_POST_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_ABA8	Post Divider Register (CCM_POST_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_ABAC	Post Divider Register (CCM_POST_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_ABB0	Pre Divider Register (CCM_PRE87)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_ABB4	Pre Divider Register (CCM_PRE_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_ABB8	Pre Divider Register (CCM_PRE_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_ABBC	Pre Divider Register (CCM_PRE_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_ABF0	Access Control Register (CCM_ACCESS_CTRL87)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_ABF4	Access Control Register (CCM_ACCESS_CTRL_ROOT87_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_ABF8	Access Control Register (CCM_ACCESS_CTRL_ROOT87_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_ABFC	Access Control Register (CCM_ACCESS_CTRL_ROOT87_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AC00	Target Register (CCM_TARGET_ROOT88)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AC04	Target Register (CCM_TARGET_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AC08	Target Register (CCM_TARGET_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AC0C	Target Register (CCM_TARGET_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AC10	Miscellaneous Register (CCM_MISC88)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AC14	Miscellaneous Register (CCM_MISC_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AC18	Miscellaneous Register (CCM_MISC_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AC1C	Miscellaneous Register (CCM_MISC_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AC20	Post Divider Register (CCM_POST88)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AC24	Post Divider Register (CCM_POST_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AC28	Post Divider Register (CCM_POST_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AC2C	Post Divider Register (CCM_POST_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AC30	Pre Divider Register (CCM_PRE88)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AC34	Pre Divider Register (CCM_PRE_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AC38	Pre Divider Register (CCM_PRE_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AC3C	Pre Divider Register (CCM_PRE_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AC70	Access Control Register (CCM_ACCESS_CTRL88)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AC74	Access Control Register (CCM_ACCESS_CTRL_ROOT88_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AC78	Access Control Register (CCM_ACCESS_CTRL_ROOT88_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AC7C	Access Control Register (CCM_ACCESS_CTRL_ROOT88_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AC80	Target Register (CCM_TARGET_ROOT89)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AC84	Target Register (CCM_TARGET_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AC88	Target Register (CCM_TARGET_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AC8C	Target Register (CCM_TARGET_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AC90	Miscellaneous Register (CCM_MISC89)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AC94	Miscellaneous Register (CCM_MISC_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AC98	Miscellaneous Register (CCM_MISC_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AC9C	Miscellaneous Register (CCM_MISC_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_ACA0	Post Divider Register (CCM_POST89)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_ACA4	Post Divider Register (CCM_POST_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_ACA8	Post Divider Register (CCM_POST_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_ACAC	Post Divider Register (CCM_POST_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_ACB0	Pre Divider Register (CCM_PRE89)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_ACB4	Pre Divider Register (CCM_PRE_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_ACB8	Pre Divider Register (CCM_PRE_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_ACBC	Pre Divider Register (CCM_PRE_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_ACF0	Access Control Register (CCM_ACCESS_CTRL89)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_ACF4	Access Control Register (CCM_ACCESS_CTRL_ROOT89_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_ACF8	Access Control Register (CCM_ACCESS_CTRL_ROOT89_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_ACFC	Access Control Register (CCM_ACCESS_CTRL_ROOT89_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AD00	Target Register (CCM_TARGET_ROOT90)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AD04	Target Register (CCM_TARGET_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AD08	Target Register (CCM_TARGET_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AD0C	Target Register (CCM_TARGET_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AD10	Miscellaneous Register (CCM_MISC90)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AD14	Miscellaneous Register (CCM_MISC_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AD18	Miscellaneous Register (CCM_MISC_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AD1C	Miscellaneous Register (CCM_MISC_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AD20	Post Divider Register (CCM_POST90)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AD24	Post Divider Register (CCM_POST_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AD28	Post Divider Register (CCM_POST_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AD2C	Post Divider Register (CCM_POST_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AD30	Pre Divider Register (CCM_PRE90)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AD34	Pre Divider Register (CCM_PRE_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AD38	Pre Divider Register (CCM_PRE_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AD3C	Pre Divider Register (CCM_PRE_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AD70	Access Control Register (CCM_ACCESS_CTRL90)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AD74	Access Control Register (CCM_ACCESS_CTRL_ROOT90_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AD78	Access Control Register (CCM_ACCESS_CTRL_ROOT90_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AD7C	Access Control Register (CCM_ACCESS_CTRL_ROOT90_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AD80	Target Register (CCM_TARGET_ROOT91)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AD84	Target Register (CCM_TARGET_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AD88	Target Register (CCM_TARGET_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AD8C	Target Register (CCM_TARGET_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AD90	Miscellaneous Register (CCM_MISC91)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AD94	Miscellaneous Register (CCM_MISC_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AD98	Miscellaneous Register (CCM_MISC_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AD9C	Miscellaneous Register (CCM_MISC_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_ADA0	Post Divider Register (CCM_POST91)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_ADA4	Post Divider Register (CCM_POST_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_ADA8	Post Divider Register (CCM_POST_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_ADAC	Post Divider Register (CCM_POST_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_ADB0	Pre Divider Register (CCM_PRE91)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_ADB4	Pre Divider Register (CCM_PRE_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_ADB8	Pre Divider Register (CCM_PRE_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_ADBC	Pre Divider Register (CCM_PRE_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_ADF0	Access Control Register (CCM_ACCESS_CTRL91)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_ADF4	Access Control Register (CCM_ACCESS_CTRL_ROOT91_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_ADF8	Access Control Register (CCM_ACCESS_CTRL_ROOT91_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_ADFC	Access Control Register (CCM_ACCESS_CTRL_ROOT91_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AE00	Target Register (CCM_TARGET_ROOT92)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AE04	Target Register (CCM_TARGET_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AE08	Target Register (CCM_TARGET_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AE0C	Target Register (CCM_TARGET_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AE10	Miscellaneous Register (CCM_MISC92)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AE14	Miscellaneous Register (CCM_MISC_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AE18	Miscellaneous Register (CCM_MISC_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AE1C	Miscellaneous Register (CCM_MISC_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AE20	Post Divider Register (CCM_POST92)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AE24	Post Divider Register (CCM_POST_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AE28	Post Divider Register (CCM_POST_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AE2C	Post Divider Register (CCM_POST_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AE30	Pre Divider Register (CCM_PRE92)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AE34	Pre Divider Register (CCM_PRE_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AE38	Pre Divider Register (CCM_PRE_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AE3C	Pre Divider Register (CCM_PRE_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AE70	Access Control Register (CCM_ACCESS_CTRL92)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AE74	Access Control Register (CCM_ACCESS_CTRL_ROOT92_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AE78	Access Control Register (CCM_ACCESS_CTRL_ROOT92_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AE7C	Access Control Register (CCM_ACCESS_CTRL_ROOT92_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AE80	Target Register (CCM_TARGET_ROOT93)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AE84	Target Register (CCM_TARGET_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AE88	Target Register (CCM_TARGET_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AE8C	Target Register (CCM_TARGET_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AE90	Miscellaneous Register (CCM_MISC93)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AE94	Miscellaneous Register (CCM_MISC_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AE98	Miscellaneous Register (CCM_MISC_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AE9C	Miscellaneous Register (CCM_MISC_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AEA0	Post Divider Register (CCM_POST93)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AEA4	Post Divider Register (CCM_POST_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AEA8	Post Divider Register (CCM_POST_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AEAC	Post Divider Register (CCM_POST_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AEB0	Pre Divider Register (CCM_PRE93)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AEB4	Pre Divider Register (CCM_PRE_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AEB8	Pre Divider Register (CCM_PRE_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AEBC	Pre Divider Register (CCM_PRE_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AEF0	Access Control Register (CCM_ACCESS_CTRL93)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AEF4	Access Control Register (CCM_ACCESS_CTRL_ROOT93_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AEF8	Access Control Register (CCM_ACCESS_CTRL_ROOT93_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AEFC	Access Control Register (CCM_ACCESS_CTRL_ROOT93_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AF00	Target Register (CCM_TARGET_ROOT94)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AF04	Target Register (CCM_TARGET_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AF08	Target Register (CCM_TARGET_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AF0C	Target Register (CCM_TARGET_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AF10	Miscellaneous Register (CCM_MISC94)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AF14	Miscellaneous Register (CCM_MISC_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AF18	Miscellaneous Register (CCM_MISC_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_AF1C	Miscellaneous Register (CCM_MISC_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AF20	Post Divider Register (CCM_POST94)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AF24	Post Divider Register (CCM_POST_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AF28	Post Divider Register (CCM_POST_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AF2C	Post Divider Register (CCM_POST_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AF30	Pre Divider Register (CCM_PRE94)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AF34	Pre Divider Register (CCM_PRE_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AF38	Pre Divider Register (CCM_PRE_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AF3C	Pre Divider Register (CCM_PRE_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AF70	Access Control Register (CCM_ACCESS_CTRL94)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AF74	Access Control Register (CCM_ACCESS_CTRL_ROOT94_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AF78	Access Control Register (CCM_ACCESS_CTRL_ROOT94_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AF7C	Access Control Register (CCM_ACCESS_CTRL_ROOT94_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_AF80	Target Register (CCM_TARGET_ROOT95)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_AF84	Target Register (CCM_TARGET_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_AF88	Target Register (CCM_TARGET_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_AF8C	Target Register (CCM_TARGET_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_AF90	Miscellaneous Register (CCM_MISC95)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_AF94	Miscellaneous Register (CCM_MISC_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_AF98	Miscellaneous Register (CCM_MISC_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_AF9C	Miscellaneous Register (CCM_MISC_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_AFA0	Post Divider Register (CCM_POST95)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_AFA4	Post Divider Register (CCM_POST_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_AFA8	Post Divider Register (CCM_POST_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_AFAC	Post Divider Register (CCM_POST_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_AFB0	Pre Divider Register (CCM_PRE95)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_AFB4	Pre Divider Register (CCM_PRE_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_AFB8	Pre Divider Register (CCM_PRE_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_AFBC	Pre Divider Register (CCM_PRE_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_AFF0	Access Control Register (CCM_ACCESS_CTRL95)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_AFF4	Access Control Register (CCM_ACCESS_CTRL_ROOT95_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_AFF8	Access Control Register (CCM_ACCESS_CTRL_ROOT95_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_AFFC	Access Control Register (CCM_ACCESS_CTRL_ROOT95_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B000	Target Register (CCM_TARGET_ROOT96)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B004	Target Register (CCM_TARGET_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B008	Target Register (CCM_TARGET_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B00C	Target Register (CCM_TARGET_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B010	Miscellaneous Register (CCM_MISC96)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B014	Miscellaneous Register (CCM_MISC_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B018	Miscellaneous Register (CCM_MISC_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B01C	Miscellaneous Register (CCM_MISC_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B020	Post Divider Register (CCM_POST96)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B024	Post Divider Register (CCM_POST_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B028	Post Divider Register (CCM_POST_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B02C	Post Divider Register (CCM_POST_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B030	Pre Divider Register (CCM_PRE96)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B034	Pre Divider Register (CCM_PRE_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B038	Pre Divider Register (CCM_PRE_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B03C	Pre Divider Register (CCM_PRE_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B070	Access Control Register (CCM_ACCESS_CTRL96)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B074	Access Control Register (CCM_ACCESS_CTRL_ROOT96_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B078	Access Control Register (CCM_ACCESS_CTRL_ROOT96_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B07C	Access Control Register (CCM_ACCESS_CTRL_ROOT96_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B080	Target Register (CCM_TARGET_ROOT97)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B084	Target Register (CCM_TARGET_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B088	Target Register (CCM_TARGET_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B08C	Target Register (CCM_TARGET_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B090	Miscellaneous Register (CCM_MISC97)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B094	Miscellaneous Register (CCM_MISC_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B098	Miscellaneous Register (CCM_MISC_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B09C	Miscellaneous Register (CCM_MISC_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B0A0	Post Divider Register (CCM_POST97)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B0A4	Post Divider Register (CCM_POST_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B0A8	Post Divider Register (CCM_POST_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B0AC	Post Divider Register (CCM_POST_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B0B0	Pre Divider Register (CCM_PRE97)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B0B4	Pre Divider Register (CCM_PRE_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B0B8	Pre Divider Register (CCM_PRE_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B0BC	Pre Divider Register (CCM_PRE_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B0F0	Access Control Register (CCM_ACCESS_CTRL97)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT97_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT97_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT97_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B100	Target Register (CCM_TARGET_ROOT98)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B104	Target Register (CCM_TARGET_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B108	Target Register (CCM_TARGET_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B10C	Target Register (CCM_TARGET_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B110	Miscellaneous Register (CCM_MISC98)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B114	Miscellaneous Register (CCM_MISC_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B118	Miscellaneous Register (CCM_MISC_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B11C	Miscellaneous Register (CCM_MISC_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B120	Post Divider Register (CCM_POST98)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B124	Post Divider Register (CCM_POST_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B128	Post Divider Register (CCM_POST_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B12C	Post Divider Register (CCM_POST_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B130	Pre Divider Register (CCM_PRE98)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B134	Pre Divider Register (CCM_PRE_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B138	Pre Divider Register (CCM_PRE_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B13C	Pre Divider Register (CCM_PRE_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B170	Access Control Register (CCM_ACCESS_CTRL98)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B174	Access Control Register (CCM_ACCESS_CTRL_ROOT98_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B178	Access Control Register (CCM_ACCESS_CTRL_ROOT98_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B17C	Access Control Register (CCM_ACCESS_CTRL_ROOT98_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B180	Target Register (CCM_TARGET_ROOT99)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B184	Target Register (CCM_TARGET_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B188	Target Register (CCM_TARGET_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B18C	Target Register (CCM_TARGET_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B190	Miscellaneous Register (CCM_MISC99)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B194	Miscellaneous Register (CCM_MISC_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B198	Miscellaneous Register (CCM_MISC_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B19C	Miscellaneous Register (CCM_MISC_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B1A0	Post Divider Register (CCM_POST99)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B1A4	Post Divider Register (CCM_POST_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B1A8	Post Divider Register (CCM_POST_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B1AC	Post Divider Register (CCM_POST_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B1B0	Pre Divider Register (CCM_PRE99)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B1B4	Pre Divider Register (CCM_PRE_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B1B8	Pre Divider Register (CCM_PRE_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B1BC	Pre Divider Register (CCM_PRE_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B1F0	Access Control Register (CCM_ACCESS_CTRL99)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT99_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT99_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT99_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B200	Target Register (CCM_TARGET_ROOT100)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B204	Target Register (CCM_TARGET_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B208	Target Register (CCM_TARGET_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B20C	Target Register (CCM_TARGET_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B210	Miscellaneous Register (CCM_MISC100)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B214	Miscellaneous Register (CCM_MISC_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B218	Miscellaneous Register (CCM_MISC_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B21C	Miscellaneous Register (CCM_MISC_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B220	Post Divider Register (CCM_POST100)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B224	Post Divider Register (CCM_POST_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B228	Post Divider Register (CCM_POST_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B22C	Post Divider Register (CCM_POST_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B230	Pre Divider Register (CCM_PRE100)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B234	Pre Divider Register (CCM_PRE_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B238	Pre Divider Register (CCM_PRE_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B23C	Pre Divider Register (CCM_PRE_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B270	Access Control Register (CCM_ACCESS_CTRL100)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B274	Access Control Register (CCM_ACCESS_CTRL_ROOT100_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B278	Access Control Register (CCM_ACCESS_CTRL_ROOT100_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B27C	Access Control Register (CCM_ACCESS_CTRL_ROOT100_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B280	Target Register (CCM_TARGET_ROOT101)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B284	Target Register (CCM_TARGET_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B288	Target Register (CCM_TARGET_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B28C	Target Register (CCM_TARGET_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B290	Miscellaneous Register (CCM_MISC101)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B294	Miscellaneous Register (CCM_MISC_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B298	Miscellaneous Register (CCM_MISC_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B29C	Miscellaneous Register (CCM_MISC_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B2A0	Post Divider Register (CCM_POST101)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B2A4	Post Divider Register (CCM_POST_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B2A8	Post Divider Register (CCM_POST_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B2AC	Post Divider Register (CCM_POST_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B2B0	Pre Divider Register (CCM_PRE101)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B2B4	Pre Divider Register (CCM_PRE_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B2B8	Pre Divider Register (CCM_PRE_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B2BC	Pre Divider Register (CCM_PRE_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B2F0	Access Control Register (CCM_ACCESS_CTRL101)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT101_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT101_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT101_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B300	Target Register (CCM_TARGET_ROOT102)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B304	Target Register (CCM_TARGET_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B308	Target Register (CCM_TARGET_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B30C	Target Register (CCM_TARGET_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B310	Miscellaneous Register (CCM_MISC102)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B314	Miscellaneous Register (CCM_MISC_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B318	Miscellaneous Register (CCM_MISC_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B31C	Miscellaneous Register (CCM_MISC_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B320	Post Divider Register (CCM_POST102)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B324	Post Divider Register (CCM_POST_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B328	Post Divider Register (CCM_POST_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B32C	Post Divider Register (CCM_POST_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B330	Pre Divider Register (CCM_PRE102)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B334	Pre Divider Register (CCM_PRE_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B338	Pre Divider Register (CCM_PRE_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B33C	Pre Divider Register (CCM_PRE_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B370	Access Control Register (CCM_ACCESS_CTRL102)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B374	Access Control Register (CCM_ACCESS_CTRL_ROOT102_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B378	Access Control Register (CCM_ACCESS_CTRL_ROOT102_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B37C	Access Control Register (CCM_ACCESS_CTRL_ROOT102_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B380	Target Register (CCM_TARGET_ROOT103)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B384	Target Register (CCM_TARGET_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B388	Target Register (CCM_TARGET_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B38C	Target Register (CCM_TARGET_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B390	Miscellaneous Register (CCM_MISC103)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B394	Miscellaneous Register (CCM_MISC_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B398	Miscellaneous Register (CCM_MISC_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B39C	Miscellaneous Register (CCM_MISC_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B3A0	Post Divider Register (CCM_POST103)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B3A4	Post Divider Register (CCM_POST_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B3A8	Post Divider Register (CCM_POST_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B3AC	Post Divider Register (CCM_POST_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B3B0	Pre Divider Register (CCM_PRE103)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B3B4	Pre Divider Register (CCM_PRE_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B3B8	Pre Divider Register (CCM_PRE_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B3BC	Pre Divider Register (CCM_PRE_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B3F0	Access Control Register (CCM_ACCESS_CTRL103)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT103_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT103_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT103_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B400	Target Register (CCM_TARGET_ROOT104)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B404	Target Register (CCM_TARGET_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B408	Target Register (CCM_TARGET_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B40C	Target Register (CCM_TARGET_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B410	Miscellaneous Register (CCM_MISC104)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B414	Miscellaneous Register (CCM_MISC_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B418	Miscellaneous Register (CCM_MISC_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B41C	Miscellaneous Register (CCM_MISC_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B420	Post Divider Register (CCM_POST104)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B424	Post Divider Register (CCM_POST_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B428	Post Divider Register (CCM_POST_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B42C	Post Divider Register (CCM_POST_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B430	Pre Divider Register (CCM_PRE104)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B434	Pre Divider Register (CCM_PRE_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B438	Pre Divider Register (CCM_PRE_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B43C	Pre Divider Register (CCM_PRE_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B470	Access Control Register (CCM_ACCESS_CTRL104)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B474	Access Control Register (CCM_ACCESS_CTRL_ROOT104_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B478	Access Control Register (CCM_ACCESS_CTRL_ROOT104_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B47C	Access Control Register (CCM_ACCESS_CTRL_ROOT104_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B480	Target Register (CCM_TARGET_ROOT105)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B484	Target Register (CCM_TARGET_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B488	Target Register (CCM_TARGET_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B48C	Target Register (CCM_TARGET_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B490	Miscellaneous Register (CCM_MISC105)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B494	Miscellaneous Register (CCM_MISC_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B498	Miscellaneous Register (CCM_MISC_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B49C	Miscellaneous Register (CCM_MISC_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B4A0	Post Divider Register (CCM_POST105)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B4A4	Post Divider Register (CCM_POST_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B4A8	Post Divider Register (CCM_POST_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B4AC	Post Divider Register (CCM_POST_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B4B0	Pre Divider Register (CCM_PRE105)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B4B4	Pre Divider Register (CCM_PRE_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B4B8	Pre Divider Register (CCM_PRE_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B4BC	Pre Divider Register (CCM_PRE_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B4F0	Access Control Register (CCM_ACCESS_CTRL105)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT105_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT105_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT105_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B500	Target Register (CCM_TARGET_ROOT106)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B504	Target Register (CCM_TARGET_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B508	Target Register (CCM_TARGET_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B50C	Target Register (CCM_TARGET_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B510	Miscellaneous Register (CCM_MISC106)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B514	Miscellaneous Register (CCM_MISC_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B518	Miscellaneous Register (CCM_MISC_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B51C	Miscellaneous Register (CCM_MISC_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B520	Post Divider Register (CCM_POST106)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B524	Post Divider Register (CCM_POST_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B528	Post Divider Register (CCM_POST_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B52C	Post Divider Register (CCM_POST_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B530	Pre Divider Register (CCM_PRE106)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B534	Pre Divider Register (CCM_PRE_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B538	Pre Divider Register (CCM_PRE_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B53C	Pre Divider Register (CCM_PRE_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B570	Access Control Register (CCM_ACCESS_CTRL106)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B574	Access Control Register (CCM_ACCESS_CTRL_ROOT106_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B578	Access Control Register (CCM_ACCESS_CTRL_ROOT106_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B57C	Access Control Register (CCM_ACCESS_CTRL_ROOT106_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B580	Target Register (CCM_TARGET_ROOT107)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B584	Target Register (CCM_TARGET_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B588	Target Register (CCM_TARGET_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B58C	Target Register (CCM_TARGET_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B590	Miscellaneous Register (CCM_MISC107)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B594	Miscellaneous Register (CCM_MISC_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B598	Miscellaneous Register (CCM_MISC_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B59C	Miscellaneous Register (CCM_MISC_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B5A0	Post Divider Register (CCM_POST107)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B5A4	Post Divider Register (CCM_POST_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B5A8	Post Divider Register (CCM_POST_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B5AC	Post Divider Register (CCM_POST_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B5B0	Pre Divider Register (CCM_PRE107)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B5B4	Pre Divider Register (CCM_PRE_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B5B8	Pre Divider Register (CCM_PRE_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B5BC	Pre Divider Register (CCM_PRE_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B5F0	Access Control Register (CCM_ACCESS_CTRL107)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT107_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT107_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT107_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B600	Target Register (CCM_TARGET_ROOT108)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B604	Target Register (CCM_TARGET_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B608	Target Register (CCM_TARGET_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B60C	Target Register (CCM_TARGET_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B610	Miscellaneous Register (CCM_MISC108)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B614	Miscellaneous Register (CCM_MISC_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B618	Miscellaneous Register (CCM_MISC_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B61C	Miscellaneous Register (CCM_MISC_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B620	Post Divider Register (CCM_POST108)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B624	Post Divider Register (CCM_POST_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B628	Post Divider Register (CCM_POST_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B62C	Post Divider Register (CCM_POST_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B630	Pre Divider Register (CCM_PRE108)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B634	Pre Divider Register (CCM_PRE_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B638	Pre Divider Register (CCM_PRE_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B63C	Pre Divider Register (CCM_PRE_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B670	Access Control Register (CCM_ACCESS_CTRL108)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B674	Access Control Register (CCM_ACCESS_CTRL_ROOT108_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B678	Access Control Register (CCM_ACCESS_CTRL_ROOT108_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B67C	Access Control Register (CCM_ACCESS_CTRL_ROOT108_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B680	Target Register (CCM_TARGET_ROOT109)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B684	Target Register (CCM_TARGET_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B688	Target Register (CCM_TARGET_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B68C	Target Register (CCM_TARGET_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B690	Miscellaneous Register (CCM_MISC109)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B694	Miscellaneous Register (CCM_MISC_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B698	Miscellaneous Register (CCM_MISC_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B69C	Miscellaneous Register (CCM_MISC_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B6A0	Post Divider Register (CCM_POST109)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B6A4	Post Divider Register (CCM_POST_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B6A8	Post Divider Register (CCM_POST_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B6AC	Post Divider Register (CCM_POST_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B6B0	Pre Divider Register (CCM_PRE109)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B6B4	Pre Divider Register (CCM_PRE_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B6B8	Pre Divider Register (CCM_PRE_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B6BC	Pre Divider Register (CCM_PRE_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B6F0	Access Control Register (CCM_ACCESS_CTRL109)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT109_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT109_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT109_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B700	Target Register (CCM_TARGET_ROOT110)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B704	Target Register (CCM_TARGET_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B708	Target Register (CCM_TARGET_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B70C	Target Register (CCM_TARGET_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B710	Miscellaneous Register (CCM_MISC110)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B714	Miscellaneous Register (CCM_MISC_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B718	Miscellaneous Register (CCM_MISC_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B71C	Miscellaneous Register (CCM_MISC_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B720	Post Divider Register (CCM_POST110)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B724	Post Divider Register (CCM_POST_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B728	Post Divider Register (CCM_POST_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B72C	Post Divider Register (CCM_POST_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B730	Pre Divider Register (CCM_PRE110)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B734	Pre Divider Register (CCM_PRE_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B738	Pre Divider Register (CCM_PRE_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B73C	Pre Divider Register (CCM_PRE_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B770	Access Control Register (CCM_ACCESS_CTRL110)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B774	Access Control Register (CCM_ACCESS_CTRL_ROOT110_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B778	Access Control Register (CCM_ACCESS_CTRL_ROOT110_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B77C	Access Control Register (CCM_ACCESS_CTRL_ROOT110_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B780	Target Register (CCM_TARGET_ROOT111)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B784	Target Register (CCM_TARGET_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B788	Target Register (CCM_TARGET_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B78C	Target Register (CCM_TARGET_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B790	Miscellaneous Register (CCM_MISC111)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B794	Miscellaneous Register (CCM_MISC_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B798	Miscellaneous Register (CCM_MISC_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B79C	Miscellaneous Register (CCM_MISC_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B7A0	Post Divider Register (CCM_POST111)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B7A4	Post Divider Register (CCM_POST_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B7A8	Post Divider Register (CCM_POST_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B7AC	Post Divider Register (CCM_POST_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B7B0	Pre Divider Register (CCM_PRE111)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B7B4	Pre Divider Register (CCM_PRE_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B7B8	Pre Divider Register (CCM_PRE_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B7BC	Pre Divider Register (CCM_PRE_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B7F0	Access Control Register (CCM_ACCESS_CTRL111)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B7F4	Access Control Register (CCM_ACCESS_CTRL_ROOT111_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B7F8	Access Control Register (CCM_ACCESS_CTRL_ROOT111_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B7FC	Access Control Register (CCM_ACCESS_CTRL_ROOT111_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B800	Target Register (CCM_TARGET_ROOT112)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B804	Target Register (CCM_TARGET_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B808	Target Register (CCM_TARGET_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B80C	Target Register (CCM_TARGET_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B810	Miscellaneous Register (CCM_MISC112)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B814	Miscellaneous Register (CCM_MISC_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B818	Miscellaneous Register (CCM_MISC_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B81C	Miscellaneous Register (CCM_MISC_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B820	Post Divider Register (CCM_POST112)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B824	Post Divider Register (CCM_POST_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B828	Post Divider Register (CCM_POST_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B82C	Post Divider Register (CCM_POST_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B830	Pre Divider Register (CCM_PRE112)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B834	Pre Divider Register (CCM_PRE_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B838	Pre Divider Register (CCM_PRE_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B83C	Pre Divider Register (CCM_PRE_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B870	Access Control Register (CCM_ACCESS_CTRL112)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B874	Access Control Register (CCM_ACCESS_CTRL_ROOT112_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B878	Access Control Register (CCM_ACCESS_CTRL_ROOT112_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B87C	Access Control Register (CCM_ACCESS_CTRL_ROOT112_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B880	Target Register (CCM_TARGET_ROOT113)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B884	Target Register (CCM_TARGET_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B888	Target Register (CCM_TARGET_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B88C	Target Register (CCM_TARGET_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B890	Miscellaneous Register (CCM_MISC113)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B894	Miscellaneous Register (CCM_MISC_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B898	Miscellaneous Register (CCM_MISC_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B89C	Miscellaneous Register (CCM_MISC_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B8A0	Post Divider Register (CCM_POST113)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B8A4	Post Divider Register (CCM_POST_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B8A8	Post Divider Register (CCM_POST_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B8AC	Post Divider Register (CCM_POST_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B8B0	Pre Divider Register (CCM_PRE113)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B8B4	Pre Divider Register (CCM_PRE_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B8B8	Pre Divider Register (CCM_PRE_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B8BC	Pre Divider Register (CCM_PRE_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B8F0	Access Control Register (CCM_ACCESS_CTRL113)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B8F4	Access Control Register (CCM_ACCESS_CTRL_ROOT113_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B8F8	Access Control Register (CCM_ACCESS_CTRL_ROOT113_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B8FC	Access Control Register (CCM_ACCESS_CTRL_ROOT113_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B900	Target Register (CCM_TARGET_ROOT114)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B904	Target Register (CCM_TARGET_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B908	Target Register (CCM_TARGET_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_B90C	Target Register (CCM_TARGET_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B910	Miscellaneous Register (CCM_MISC114)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B914	Miscellaneous Register (CCM_MISC_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B918	Miscellaneous Register (CCM_MISC_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B91C	Miscellaneous Register (CCM_MISC_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B920	Post Divider Register (CCM_POST114)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B924	Post Divider Register (CCM_POST_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B928	Post Divider Register (CCM_POST_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B92C	Post Divider Register (CCM_POST_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B930	Pre Divider Register (CCM_PRE114)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B934	Pre Divider Register (CCM_PRE_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B938	Pre Divider Register (CCM_PRE_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B93C	Pre Divider Register (CCM_PRE_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B970	Access Control Register (CCM_ACCESS_CTRL114)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B974	Access Control Register (CCM_ACCESS_CTRL_ROOT114_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B978	Access Control Register (CCM_ACCESS_CTRL_ROOT114_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B97C	Access Control Register (CCM_ACCESS_CTRL_ROOT114_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_B980	Target Register (CCM_TARGET_ROOT115)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_B984	Target Register (CCM_TARGET_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_B988	Target Register (CCM_TARGET_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_B98C	Target Register (CCM_TARGET_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_B990	Miscellaneous Register (CCM_MISC115)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_B994	Miscellaneous Register (CCM_MISC_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_B998	Miscellaneous Register (CCM_MISC_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_B99C	Miscellaneous Register (CCM_MISC_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_B9A0	Post Divider Register (CCM_POST115)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_B9A4	Post Divider Register (CCM_POST_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_B9A8	Post Divider Register (CCM_POST_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_B9AC	Post Divider Register (CCM_POST_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_B9B0	Pre Divider Register (CCM_PRE115)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_B9B4	Pre Divider Register (CCM_PRE_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_B9B8	Pre Divider Register (CCM_PRE_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_B9BC	Pre Divider Register (CCM_PRE_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_B9F0	Access Control Register (CCM_ACCESS_CTRL115)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_B9F4	Access Control Register (CCM_ACCESS_CTRL_ROOT115_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_B9F8	Access Control Register (CCM_ACCESS_CTRL_ROOT115_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_B9FC	Access Control Register (CCM_ACCESS_CTRL_ROOT115_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BA00	Target Register (CCM_TARGET_ROOT116)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BA04	Target Register (CCM_TARGET_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BA08	Target Register (CCM_TARGET_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BA0C	Target Register (CCM_TARGET_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BA10	Miscellaneous Register (CCM_MISC116)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BA14	Miscellaneous Register (CCM_MISC_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BA18	Miscellaneous Register (CCM_MISC_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BA1C	Miscellaneous Register (CCM_MISC_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BA20	Post Divider Register (CCM_POST116)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BA24	Post Divider Register (CCM_POST_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BA28	Post Divider Register (CCM_POST_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BA2C	Post Divider Register (CCM_POST_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BA30	Pre Divider Register (CCM_PRE116)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BA34	Pre Divider Register (CCM_PRE_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BA38	Pre Divider Register (CCM_PRE_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BA3C	Pre Divider Register (CCM_PRE_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BA70	Access Control Register (CCM_ACCESS_CTRL116)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BA74	Access Control Register (CCM_ACCESS_CTRL_ROOT116_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BA78	Access Control Register (CCM_ACCESS_CTRL_ROOT116_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BA7C	Access Control Register (CCM_ACCESS_CTRL_ROOT116_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BA80	Target Register (CCM_TARGET_ROOT117)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BA84	Target Register (CCM_TARGET_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BA88	Target Register (CCM_TARGET_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BA8C	Target Register (CCM_TARGET_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BA90	Miscellaneous Register (CCM_MISC117)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BA94	Miscellaneous Register (CCM_MISC_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BA98	Miscellaneous Register (CCM_MISC_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BA9C	Miscellaneous Register (CCM_MISC_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BAA0	Post Divider Register (CCM_POST117)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BAA4	Post Divider Register (CCM_POST_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BAA8	Post Divider Register (CCM_POST_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BAAC	Post Divider Register (CCM_POST_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BAB0	Pre Divider Register (CCM_PRE117)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BAB4	Pre Divider Register (CCM_PRE_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BAB8	Pre Divider Register (CCM_PRE_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BABC	Pre Divider Register (CCM_PRE_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BAF0	Access Control Register (CCM_ACCESS_CTRL117)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BAF4	Access Control Register (CCM_ACCESS_CTRL_ROOT117_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BAF8	Access Control Register (CCM_ACCESS_CTRL_ROOT117_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BAFC	Access Control Register (CCM_ACCESS_CTRL_ROOT117_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BB00	Target Register (CCM_TARGET_ROOT118)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BB04	Target Register (CCM_TARGET_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BB08	Target Register (CCM_TARGET_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BB0C	Target Register (CCM_TARGET_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BB10	Miscellaneous Register (CCM_MISC118)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BB14	Miscellaneous Register (CCM_MISC_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BB18	Miscellaneous Register (CCM_MISC_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BB1C	Miscellaneous Register (CCM_MISC_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BB20	Post Divider Register (CCM_POST118)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BB24	Post Divider Register (CCM_POST_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BB28	Post Divider Register (CCM_POST_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BB2C	Post Divider Register (CCM_POST_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BB30	Pre Divider Register (CCM_PRE118)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BB34	Pre Divider Register (CCM_PRE_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BB38	Pre Divider Register (CCM_PRE_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BB3C	Pre Divider Register (CCM_PRE_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BB70	Access Control Register (CCM_ACCESS_CTRL118)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BB74	Access Control Register (CCM_ACCESS_CTRL_ROOT118_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BB78	Access Control Register (CCM_ACCESS_CTRL_ROOT118_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BB7C	Access Control Register (CCM_ACCESS_CTRL_ROOT118_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BB80	Target Register (CCM_TARGET_ROOT119)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BB84	Target Register (CCM_TARGET_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BB88	Target Register (CCM_TARGET_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BB8C	Target Register (CCM_TARGET_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BB90	Miscellaneous Register (CCM_MISC119)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BB94	Miscellaneous Register (CCM_MISC_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BB98	Miscellaneous Register (CCM_MISC_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BB9C	Miscellaneous Register (CCM_MISC_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BBA0	Post Divider Register (CCM_POST119)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BBA4	Post Divider Register (CCM_POST_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BBA8	Post Divider Register (CCM_POST_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BBAC	Post Divider Register (CCM_POST_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BBB0	Pre Divider Register (CCM_PRE119)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BBB4	Pre Divider Register (CCM_PRE_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BBB8	Pre Divider Register (CCM_PRE_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BBBC	Pre Divider Register (CCM_PRE_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BBF0	Access Control Register (CCM_ACCESS_CTRL119)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BBF4	Access Control Register (CCM_ACCESS_CTRL_ROOT119_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BBF8	Access Control Register (CCM_ACCESS_CTRL_ROOT119_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BBFC	Access Control Register (CCM_ACCESS_CTRL_ROOT119_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BC00	Target Register (CCM_TARGET_ROOT120)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BC04	Target Register (CCM_TARGET_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BC08	Target Register (CCM_TARGET_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BC0C	Target Register (CCM_TARGET_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BC10	Miscellaneous Register (CCM_MISC120)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BC14	Miscellaneous Register (CCM_MISC_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BC18	Miscellaneous Register (CCM_MISC_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BC1C	Miscellaneous Register (CCM_MISC_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BC20	Post Divider Register (CCM_POST120)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BC24	Post Divider Register (CCM_POST_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BC28	Post Divider Register (CCM_POST_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BC2C	Post Divider Register (CCM_POST_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BC30	Pre Divider Register (CCM_PRE120)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BC34	Pre Divider Register (CCM_PRE_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BC38	Pre Divider Register (CCM_PRE_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BC3C	Pre Divider Register (CCM_PRE_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BC70	Access Control Register (CCM_ACCESS_CTRL120)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BC74	Access Control Register (CCM_ACCESS_CTRL_ROOT120_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BC78	Access Control Register (CCM_ACCESS_CTRL_ROOT120_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BC7C	Access Control Register (CCM_ACCESS_CTRL_ROOT120_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BC80	Target Register (CCM_TARGET_ROOT121)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BC84	Target Register (CCM_TARGET_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BC88	Target Register (CCM_TARGET_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BC8C	Target Register (CCM_TARGET_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BC90	Miscellaneous Register (CCM_MISC121)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BC94	Miscellaneous Register (CCM_MISC_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BC98	Miscellaneous Register (CCM_MISC_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BC9C	Miscellaneous Register (CCM_MISC_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BCA0	Post Divider Register (CCM_POST121)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BCA4	Post Divider Register (CCM_POST_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BCA8	Post Divider Register (CCM_POST_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BCAC	Post Divider Register (CCM_POST_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BCB0	Pre Divider Register (CCM_PRE121)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BCB4	Pre Divider Register (CCM_PRE_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BCB8	Pre Divider Register (CCM_PRE_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BCBC	Pre Divider Register (CCM_PRE_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BCF0	Access Control Register (CCM_ACCESS_CTRL121)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BCF4	Access Control Register (CCM_ACCESS_CTRL_ROOT121_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BCF8	Access Control Register (CCM_ACCESS_CTRL_ROOT121_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BCFC	Access Control Register (CCM_ACCESS_CTRL_ROOT121_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BD00	Target Register (CCM_TARGET_ROOT122)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BD04	Target Register (CCM_TARGET_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BD08	Target Register (CCM_TARGET_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BD0C	Target Register (CCM_TARGET_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BD10	Miscellaneous Register (CCM_MISC122)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BD14	Miscellaneous Register (CCM_MISC_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BD18	Miscellaneous Register (CCM_MISC_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BD1C	Miscellaneous Register (CCM_MISC_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BD20	Post Divider Register (CCM_POST122)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BD24	Post Divider Register (CCM_POST_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BD28	Post Divider Register (CCM_POST_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BD2C	Post Divider Register (CCM_POST_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BD30	Pre Divider Register (CCM_PRE122)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BD34	Pre Divider Register (CCM_PRE_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BD38	Pre Divider Register (CCM_PRE_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BD3C	Pre Divider Register (CCM_PRE_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BD70	Access Control Register (CCM_ACCESS_CTRL122)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BD74	Access Control Register (CCM_ACCESS_CTRL_ROOT122_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BD78	Access Control Register (CCM_ACCESS_CTRL_ROOT122_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BD7C	Access Control Register (CCM_ACCESS_CTRL_ROOT122_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BD80	Target Register (CCM_TARGET_ROOT123)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BD84	Target Register (CCM_TARGET_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BD88	Target Register (CCM_TARGET_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BD8C	Target Register (CCM_TARGET_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BD90	Miscellaneous Register (CCM_MISC123)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BD94	Miscellaneous Register (CCM_MISC_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BD98	Miscellaneous Register (CCM_MISC_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BD9C	Miscellaneous Register (CCM_MISC_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BDA0	Post Divider Register (CCM_POST123)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BDA4	Post Divider Register (CCM_POST_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BDA8	Post Divider Register (CCM_POST_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BDAC	Post Divider Register (CCM_POST_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BDB0	Pre Divider Register (CCM_PRE123)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BDB4	Pre Divider Register (CCM_PRE_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BDB8	Pre Divider Register (CCM_PRE_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BDBC	Pre Divider Register (CCM_PRE_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BDF0	Access Control Register (CCM_ACCESS_CTRL123)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BDF4	Access Control Register (CCM_ACCESS_CTRL_ROOT123_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BDF8	Access Control Register (CCM_ACCESS_CTRL_ROOT123_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BDFC	Access Control Register (CCM_ACCESS_CTRL_ROOT123_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BE00	Target Register (CCM_TARGET_ROOT124)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BE04	Target Register (CCM_TARGET_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BE08	Target Register (CCM_TARGET_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BE0C	Target Register (CCM_TARGET_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BE10	Miscellaneous Register (CCM_MISC124)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BE14	Miscellaneous Register (CCM_MISC_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BE18	Miscellaneous Register (CCM_MISC_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BE1C	Miscellaneous Register (CCM_MISC_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BE20	Post Divider Register (CCM_POST124)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BE24	Post Divider Register (CCM_POST_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BE28	Post Divider Register (CCM_POST_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BE2C	Post Divider Register (CCM_POST_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BE30	Pre Divider Register (CCM_PRE124)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BE34	Pre Divider Register (CCM_PRE_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BE38	Pre Divider Register (CCM_PRE_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BE3C	Pre Divider Register (CCM_PRE_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BE70	Access Control Register (CCM_ACCESS_CTRL124)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BE74	Access Control Register (CCM_ACCESS_CTRL_ROOT124_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BE78	Access Control Register (CCM_ACCESS_CTRL_ROOT124_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BE7C	Access Control Register (CCM_ACCESS_CTRL_ROOT124_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BE80	Target Register (CCM_TARGET_ROOT125)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BE84	Target Register (CCM_TARGET_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BE88	Target Register (CCM_TARGET_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BE8C	Target Register (CCM_TARGET_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BE90	Miscellaneous Register (CCM_MISC125)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BE94	Miscellaneous Register (CCM_MISC_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BE98	Miscellaneous Register (CCM_MISC_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BE9C	Miscellaneous Register (CCM_MISC_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BEA0	Post Divider Register (CCM_POST125)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BEA4	Post Divider Register (CCM_POST_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BEA8	Post Divider Register (CCM_POST_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BEAC	Post Divider Register (CCM_POST_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BEB0	Pre Divider Register (CCM_PRE125)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BEB4	Pre Divider Register (CCM_PRE_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BEB8	Pre Divider Register (CCM_PRE_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BEBC	Pre Divider Register (CCM_PRE_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BEF0	Access Control Register (CCM_ACCESS_CTRL125)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BEF4	Access Control Register (CCM_ACCESS_CTRL_ROOT125_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BEF8	Access Control Register (CCM_ACCESS_CTRL_ROOT125_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BEFC	Access Control Register (CCM_ACCESS_CTRL_ROOT125_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BF00	Target Register (CCM_TARGET_ROOT126)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BF04	Target Register (CCM_TARGET_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BF08	Target Register (CCM_TARGET_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BF0C	Target Register (CCM_TARGET_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BF10	Miscellaneous Register (CCM_MISC126)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_BF14	Miscellaneous Register (CCM_MISC_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BF18	Miscellaneous Register (CCM_MISC_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BF1C	Miscellaneous Register (CCM_MISC_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BF20	Post Divider Register (CCM_POST126)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BF24	Post Divider Register (CCM_POST_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BF28	Post Divider Register (CCM_POST_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BF2C	Post Divider Register (CCM_POST_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BF30	Pre Divider Register (CCM_PRE126)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BF34	Pre Divider Register (CCM_PRE_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BF38	Pre Divider Register (CCM_PRE_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BF3C	Pre Divider Register (CCM_PRE_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BF70	Access Control Register (CCM_ACCESS_CTRL126)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BF74	Access Control Register (CCM_ACCESS_CTRL_ROOT126_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BF78	Access Control Register (CCM_ACCESS_CTRL_ROOT126_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BF7C	Access Control Register (CCM_ACCESS_CTRL_ROOT126_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_BF80	Target Register (CCM_TARGET_ROOT127)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_BF84	Target Register (CCM_TARGET_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_BF88	Target Register (CCM_TARGET_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_BF8C	Target Register (CCM_TARGET_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_BF90	Miscellaneous Register (CCM_MISC127)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_BF94	Miscellaneous Register (CCM_MISC_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_BF98	Miscellaneous Register (CCM_MISC_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_BF9C	Miscellaneous Register (CCM_MISC_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_BFA0	Post Divider Register (CCM_POST127)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_BFA4	Post Divider Register (CCM_POST_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_BFA8	Post Divider Register (CCM_POST_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_BFAC	Post Divider Register (CCM_POST_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_BFB0	Pre Divider Register (CCM_PRE127)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_BFB4	Pre Divider Register (CCM_PRE_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_BFB8	Pre Divider Register (CCM_PRE_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_BFBC	Pre Divider Register (CCM_PRE_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_BFF0	Access Control Register (CCM_ACCESS_CTRL127)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_BFF4	Access Control Register (CCM_ACCESS_CTRL_ROOT127_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_BFF8	Access Control Register (CCM_ACCESS_CTRL_ROOT127_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_BFFC	Access Control Register (CCM_ACCESS_CTRL_ROOT127_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C000	Target Register (CCM_TARGET_ROOT128)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C004	Target Register (CCM_TARGET_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C008	Target Register (CCM_TARGET_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C00C	Target Register (CCM_TARGET_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C010	Miscellaneous Register (CCM_MISC128)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C014	Miscellaneous Register (CCM_MISC_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C018	Miscellaneous Register (CCM_MISC_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C01C	Miscellaneous Register (CCM_MISC_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C020	Post Divider Register (CCM_POST128)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C024	Post Divider Register (CCM_POST_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C028	Post Divider Register (CCM_POST_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C02C	Post Divider Register (CCM_POST_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C030	Pre Divider Register (CCM_PRE128)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C034	Pre Divider Register (CCM_PRE_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C038	Pre Divider Register (CCM_PRE_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C03C	Pre Divider Register (CCM_PRE_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C070	Access Control Register (CCM_ACCESS_CTRL128)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C074	Access Control Register (CCM_ACCESS_CTRL_ROOT128_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C078	Access Control Register (CCM_ACCESS_CTRL_ROOT128_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C07C	Access Control Register (CCM_ACCESS_CTRL_ROOT128_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C080	Target Register (CCM_TARGET_ROOT129)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C084	Target Register (CCM_TARGET_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C088	Target Register (CCM_TARGET_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C08C	Target Register (CCM_TARGET_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C090	Miscellaneous Register (CCM_MISC129)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C094	Miscellaneous Register (CCM_MISC_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C098	Miscellaneous Register (CCM_MISC_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C09C	Miscellaneous Register (CCM_MISC_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C0A0	Post Divider Register (CCM_POST129)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C0A4	Post Divider Register (CCM_POST_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C0A8	Post Divider Register (CCM_POST_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C0AC	Post Divider Register (CCM_POST_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C0B0	Pre Divider Register (CCM_PRE129)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C0B4	Pre Divider Register (CCM_PRE_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C0B8	Pre Divider Register (CCM_PRE_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C0BC	Pre Divider Register (CCM_PRE_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C0F0	Access Control Register (CCM_ACCESS_CTRL129)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C0F4	Access Control Register (CCM_ACCESS_CTRL_ROOT129_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C0F8	Access Control Register (CCM_ACCESS_CTRL_ROOT129_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C0FC	Access Control Register (CCM_ACCESS_CTRL_ROOT129_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C100	Target Register (CCM_TARGET_ROOT130)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C104	Target Register (CCM_TARGET_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C108	Target Register (CCM_TARGET_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C10C	Target Register (CCM_TARGET_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C110	Miscellaneous Register (CCM_MISC130)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C114	Miscellaneous Register (CCM_MISC_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C118	Miscellaneous Register (CCM_MISC_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C11C	Miscellaneous Register (CCM_MISC_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C120	Post Divider Register (CCM_POST130)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C124	Post Divider Register (CCM_POST_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C128	Post Divider Register (CCM_POST_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C12C	Post Divider Register (CCM_POST_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C130	Pre Divider Register (CCM_PRE130)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C134	Pre Divider Register (CCM_PRE_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C138	Pre Divider Register (CCM_PRE_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C13C	Pre Divider Register (CCM_PRE_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C170	Access Control Register (CCM_ACCESS_CTRL130)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C174	Access Control Register (CCM_ACCESS_CTRL_ROOT130_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C178	Access Control Register (CCM_ACCESS_CTRL_ROOT130_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C17C	Access Control Register (CCM_ACCESS_CTRL_ROOT130_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C180	Target Register (CCM_TARGET_ROOT131)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C184	Target Register (CCM_TARGET_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C188	Target Register (CCM_TARGET_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C18C	Target Register (CCM_TARGET_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C190	Miscellaneous Register (CCM_MISC131)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C194	Miscellaneous Register (CCM_MISC_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C198	Miscellaneous Register (CCM_MISC_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C19C	Miscellaneous Register (CCM_MISC_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C1A0	Post Divider Register (CCM_POST131)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C1A4	Post Divider Register (CCM_POST_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C1A8	Post Divider Register (CCM_POST_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C1AC	Post Divider Register (CCM_POST_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C1B0	Pre Divider Register (CCM_PRE131)	32	R/W	1000_0000h	5.1.7.22/ 542

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C1B4	Pre Divider Register (CCM_PRE_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C1B8	Pre Divider Register (CCM_PRE_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C1BC	Pre Divider Register (CCM_PRE_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C1F0	Access Control Register (CCM_ACCESS_CTRL131)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C1F4	Access Control Register (CCM_ACCESS_CTRL_ROOT131_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C1F8	Access Control Register (CCM_ACCESS_CTRL_ROOT131_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C1FC	Access Control Register (CCM_ACCESS_CTRL_ROOT131_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C200	Target Register (CCM_TARGET_ROOT132)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C204	Target Register (CCM_TARGET_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C208	Target Register (CCM_TARGET_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C20C	Target Register (CCM_TARGET_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C210	Miscellaneous Register (CCM_MISC132)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C214	Miscellaneous Register (CCM_MISC_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C218	Miscellaneous Register (CCM_MISC_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C21C	Miscellaneous Register (CCM_MISC_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C220	Post Divider Register (CCM_POST132)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C224	Post Divider Register (CCM_POST_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C228	Post Divider Register (CCM_POST_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C22C	Post Divider Register (CCM_POST_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C230	Pre Divider Register (CCM_PRE132)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C234	Pre Divider Register (CCM_PRE_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C238	Pre Divider Register (CCM_PRE_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C23C	Pre Divider Register (CCM_PRE_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C270	Access Control Register (CCM_ACCESS_CTRL132)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C274	Access Control Register (CCM_ACCESS_CTRL_ROOT132_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C278	Access Control Register (CCM_ACCESS_CTRL_ROOT132_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C27C	Access Control Register (CCM_ACCESS_CTRL_ROOT132_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C280	Target Register (CCM_TARGET_ROOT133)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C284	Target Register (CCM_TARGET_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C288	Target Register (CCM_TARGET_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C28C	Target Register (CCM_TARGET_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C290	Miscellaneous Register (CCM_MISC133)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C294	Miscellaneous Register (CCM_MISC_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C298	Miscellaneous Register (CCM_MISC_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C29C	Miscellaneous Register (CCM_MISC_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C2A0	Post Divider Register (CCM_POST133)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C2A4	Post Divider Register (CCM_POST_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C2A8	Post Divider Register (CCM_POST_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C2AC	Post Divider Register (CCM_POST_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C2B0	Pre Divider Register (CCM_PRE133)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C2B4	Pre Divider Register (CCM_PRE_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C2B8	Pre Divider Register (CCM_PRE_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C2BC	Pre Divider Register (CCM_PRE_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C2F0	Access Control Register (CCM_ACCESS_CTRL133)	32	R/W	0000_0000h	5.1.7.26/ 554

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C2F4	Access Control Register (CCM_ACCESS_CTRL_ROOT133_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C2F8	Access Control Register (CCM_ACCESS_CTRL_ROOT133_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C2FC	Access Control Register (CCM_ACCESS_CTRL_ROOT133_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C300	Target Register (CCM_TARGET_ROOT134)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C304	Target Register (CCM_TARGET_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C308	Target Register (CCM_TARGET_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C30C	Target Register (CCM_TARGET_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C310	Miscellaneous Register (CCM_MISC134)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C314	Miscellaneous Register (CCM_MISC_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C318	Miscellaneous Register (CCM_MISC_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C31C	Miscellaneous Register (CCM_MISC_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C320	Post Divider Register (CCM_POST134)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C324	Post Divider Register (CCM_POST_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C328	Post Divider Register (CCM_POST_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C32C	Post Divider Register (CCM_POST_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C330	Pre Divider Register (CCM_PRE134)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C334	Pre Divider Register (CCM_PRE_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C338	Pre Divider Register (CCM_PRE_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C33C	Pre Divider Register (CCM_PRE_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C370	Access Control Register (CCM_ACCESS_CTRL134)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C374	Access Control Register (CCM_ACCESS_CTRL_ROOT134_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C378	Access Control Register (CCM_ACCESS_CTRL_ROOT134_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C37C	Access Control Register (CCM_ACCESS_CTRL_ROOT134_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C380	Target Register (CCM_TARGET_ROOT135)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C384	Target Register (CCM_TARGET_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C388	Target Register (CCM_TARGET_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C38C	Target Register (CCM_TARGET_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C390	Miscellaneous Register (CCM_MISC135)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C394	Miscellaneous Register (CCM_MISC_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C398	Miscellaneous Register (CCM_MISC_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C39C	Miscellaneous Register (CCM_MISC_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C3A0	Post Divider Register (CCM_POST135)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C3A4	Post Divider Register (CCM_POST_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C3A8	Post Divider Register (CCM_POST_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C3AC	Post Divider Register (CCM_POST_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C3B0	Pre Divider Register (CCM_PRE135)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C3B4	Pre Divider Register (CCM_PRE_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C3B8	Pre Divider Register (CCM_PRE_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C3BC	Pre Divider Register (CCM_PRE_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C3F0	Access Control Register (CCM_ACCESS_CTRL135)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C3F4	Access Control Register (CCM_ACCESS_CTRL_ROOT135_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C3F8	Access Control Register (CCM_ACCESS_CTRL_ROOT135_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C3FC	Access Control Register (CCM_ACCESS_CTRL_ROOT135_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C400	Target Register (CCM_TARGET_ROOT136)	32	R/W	1000_0000h	5.1.7.10/ 518

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C404	Target Register (CCM_TARGET_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C408	Target Register (CCM_TARGET_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C40C	Target Register (CCM_TARGET_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C410	Miscellaneous Register (CCM_MISC136)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C414	Miscellaneous Register (CCM_MISC_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C418	Miscellaneous Register (CCM_MISC_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C41C	Miscellaneous Register (CCM_MISC_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C420	Post Divider Register (CCM_POST136)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C424	Post Divider Register (CCM_POST_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C428	Post Divider Register (CCM_POST_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C42C	Post Divider Register (CCM_POST_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C430	Pre Divider Register (CCM_PRE136)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C434	Pre Divider Register (CCM_PRE_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C438	Pre Divider Register (CCM_PRE_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C43C	Pre Divider Register (CCM_PRE_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C470	Access Control Register (CCM_ACCESS_CTRL136)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C474	Access Control Register (CCM_ACCESS_CTRL_ROOT136_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C478	Access Control Register (CCM_ACCESS_CTRL_ROOT136_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C47C	Access Control Register (CCM_ACCESS_CTRL_ROOT136_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C480	Target Register (CCM_TARGET_ROOT137)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C484	Target Register (CCM_TARGET_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C488	Target Register (CCM_TARGET_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C48C	Target Register (CCM_TARGET_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C490	Miscellaneous Register (CCM_MISC137)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C494	Miscellaneous Register (CCM_MISC_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C498	Miscellaneous Register (CCM_MISC_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C49C	Miscellaneous Register (CCM_MISC_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C4A0	Post Divider Register (CCM_POST137)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C4A4	Post Divider Register (CCM_POST_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C4A8	Post Divider Register (CCM_POST_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C4AC	Post Divider Register (CCM_POST_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C4B0	Pre Divider Register (CCM_PRE137)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C4B4	Pre Divider Register (CCM_PRE_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C4B8	Pre Divider Register (CCM_PRE_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C4BC	Pre Divider Register (CCM_PRE_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C4F0	Access Control Register (CCM_ACCESS_CTRL137)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C4F4	Access Control Register (CCM_ACCESS_CTRL_ROOT137_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C4F8	Access Control Register (CCM_ACCESS_CTRL_ROOT137_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C4FC	Access Control Register (CCM_ACCESS_CTRL_ROOT137_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C500	Target Register (CCM_TARGET_ROOT138)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C504	Target Register (CCM_TARGET_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C508	Target Register (CCM_TARGET_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C50C	Target Register (CCM_TARGET_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C510	Miscellaneous Register (CCM_MISC138)	32	R/W	0000_0000h	5.1.7.14/ 526

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C514	Miscellaneous Register (CCM_MISC_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C518	Miscellaneous Register (CCM_MISC_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C51C	Miscellaneous Register (CCM_MISC_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C520	Post Divider Register (CCM_POST138)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C524	Post Divider Register (CCM_POST_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C528	Post Divider Register (CCM_POST_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C52C	Post Divider Register (CCM_POST_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C530	Pre Divider Register (CCM_PRE138)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C534	Pre Divider Register (CCM_PRE_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C538	Pre Divider Register (CCM_PRE_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C53C	Pre Divider Register (CCM_PRE_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C570	Access Control Register (CCM_ACCESS_CTRL138)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C574	Access Control Register (CCM_ACCESS_CTRL_ROOT138_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C578	Access Control Register (CCM_ACCESS_CTRL_ROOT138_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C57C	Access Control Register (CCM_ACCESS_CTRL_ROOT138_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C580	Target Register (CCM_TARGET_ROOT139)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C584	Target Register (CCM_TARGET_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C588	Target Register (CCM_TARGET_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C58C	Target Register (CCM_TARGET_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C590	Miscellaneous Register (CCM_MISC139)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C594	Miscellaneous Register (CCM_MISC_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C598	Miscellaneous Register (CCM_MISC_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C59C	Miscellaneous Register (CCM_MISC_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C5A0	Post Divider Register (CCM_POST139)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C5A4	Post Divider Register (CCM_POST_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C5A8	Post Divider Register (CCM_POST_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C5AC	Post Divider Register (CCM_POST_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C5B0	Pre Divider Register (CCM_PRE139)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C5B4	Pre Divider Register (CCM_PRE_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C5B8	Pre Divider Register (CCM_PRE_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C5BC	Pre Divider Register (CCM_PRE_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C5F0	Access Control Register (CCM_ACCESS_CTRL139)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C5F4	Access Control Register (CCM_ACCESS_CTRL_ROOT139_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C5F8	Access Control Register (CCM_ACCESS_CTRL_ROOT139_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C5FC	Access Control Register (CCM_ACCESS_CTRL_ROOT139_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C600	Target Register (CCM_TARGET_ROOT140)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C604	Target Register (CCM_TARGET_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C608	Target Register (CCM_TARGET_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C60C	Target Register (CCM_TARGET_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C610	Miscellaneous Register (CCM_MISC140)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C614	Miscellaneous Register (CCM_MISC_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C618	Miscellaneous Register (CCM_MISC_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C61C	Miscellaneous Register (CCM_MISC_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C620	Post Divider Register (CCM_POST140)	32	R/W	0000_0000h	5.1.7.18/ 530

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C624	Post Divider Register (CCM_POST_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C628	Post Divider Register (CCM_POST_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536
3038_C62C	Post Divider Register (CCM_POST_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C630	Pre Divider Register (CCM_PRE140)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C634	Pre Divider Register (CCM_PRE_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C638	Pre Divider Register (CCM_PRE_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C63C	Pre Divider Register (CCM_PRE_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C670	Access Control Register (CCM_ACCESS_CTRL140)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C674	Access Control Register (CCM_ACCESS_CTRL_ROOT140_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C678	Access Control Register (CCM_ACCESS_CTRL_ROOT140_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C67C	Access Control Register (CCM_ACCESS_CTRL_ROOT140_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561
3038_C680	Target Register (CCM_TARGET_ROOT141)	32	R/W	1000_0000h	5.1.7.10/ 518
3038_C684	Target Register (CCM_TARGET_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.11/ 520
3038_C688	Target Register (CCM_TARGET_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.12/ 522
3038_C68C	Target Register (CCM_TARGET_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.13/ 524
3038_C690	Miscellaneous Register (CCM_MISC141)	32	R/W	0000_0000h	5.1.7.14/ 526
3038_C694	Miscellaneous Register (CCM_MISC_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.15/ 527
3038_C698	Miscellaneous Register (CCM_MISC_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.16/ 528
3038_C69C	Miscellaneous Register (CCM_MISC_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.17/ 529
3038_C6A0	Post Divider Register (CCM_POST141)	32	R/W	0000_0000h	5.1.7.18/ 530
3038_C6A4	Post Divider Register (CCM_POST_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.19/ 533
3038_C6A8	Post Divider Register (CCM_POST_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.20/ 536

Table continues on the next page...

CCM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3038_C6AC	Post Divider Register (CCM_POST_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.21/ 539
3038_C6B0	Pre Divider Register (CCM_PRE141)	32	R/W	1000_0000h	5.1.7.22/ 542
3038_C6B4	Pre Divider Register (CCM_PRE_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.23/ 545
3038_C6B8	Pre Divider Register (CCM_PRE_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.24/ 548
3038_C6BC	Pre Divider Register (CCM_PRE_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.25/ 551
3038_C6F0	Access Control Register (CCM_ACCESS_CTRL141)	32	R/W	0000_0000h	5.1.7.26/ 554
3038_C6F4	Access Control Register (CCM_ACCESS_CTRL_ROOT141_SET)	32	R/W	0000_0000h	5.1.7.27/ 556
3038_C6F8	Access Control Register (CCM_ACCESS_CTRL_ROOT141_CLR)	32	R/W	0000_0000h	5.1.7.28/ 559
3038_C6FC	Access Control Register (CCM_ACCESS_CTRL_ROOT141_TOG)	32	R/W	0000_0000h	5.1.7.29/ 561

5.1.7.1 General Purpose Register (CCM_GPR0*n*)

GPR0

Address: 3038_0000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GP0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_GPR0*n* field descriptions

Field	Description
GP0	Timeout cycle count of ipg_clk, when perform read and write.

5.1.7.2 CCM PLL Control Register (CCM_PLL_CTRL*n*)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 800h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n* field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n* field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.3 CCM PLL Control Register (CCM_PLL_CTRL*n*_SET)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 804h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_SET field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_SET field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.4 CCM PLL Control Register (CCM_PLL_CTRL*n*_CLR)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 808h offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_CLR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_CLR field descriptions (continued)

Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.5 CCM PLL Control Register (CCM_PLL_CTRL*n*_TOG)

See Input Clocks for PLL control mapping.

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

Address: 3038_0000h base + 80Ch offset + (16d × i), where i=0d to 38d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_PLL_CTRL*n*_TOG field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed

Table continues on the next page...

CCM_PLL_CTRL*n*_TOG field descriptions (continued)

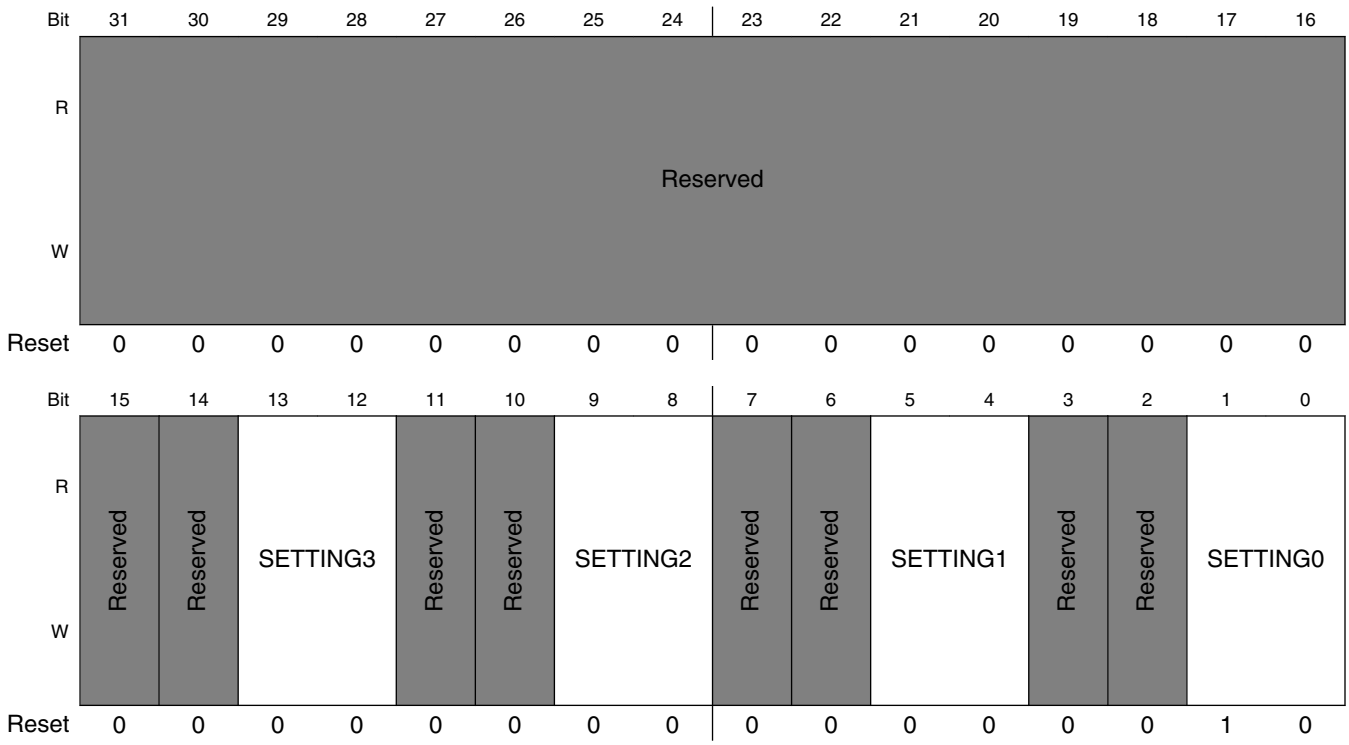
Field	Description
	01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.6 CCM Clock Gating Register (CCM_CCGR*n*)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 4000h offset + (16d × i), where i=0d to 191d



CCM_CCGR*n* field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n* field descriptions (continued)

Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.7 CCM Clock Gating Register (CCM_CCGR*n*_SET)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 4004h offset + (16d × i), where i=0d to 191d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved	Reserved	SETTING3		Reserved	Reserved	SETTING2		Reserved	Reserved	SETTING1		Reserved	Reserved	SETTING0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

CCM_CCGR*n*_SET field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_SET field descriptions (continued)

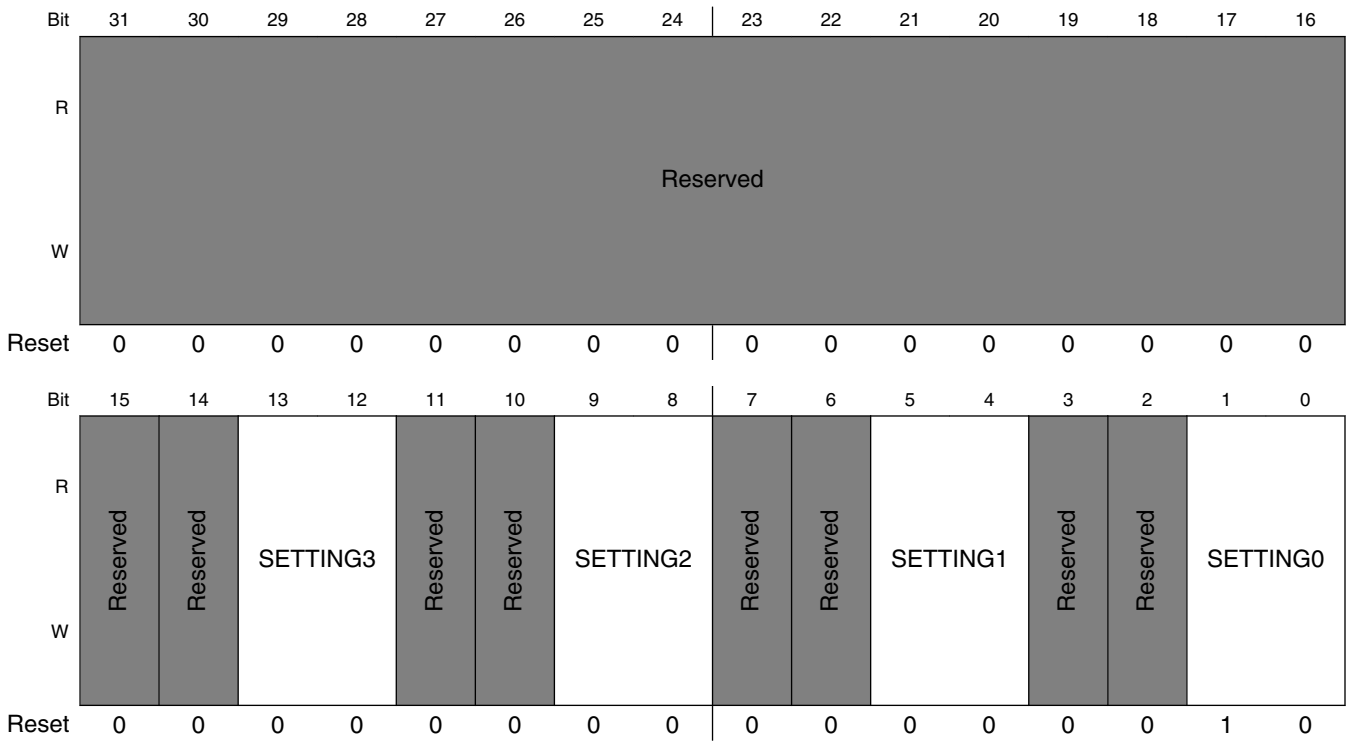
Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.8 CCM Clock Gating Register (CCM_CCGR*n*_CLR)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 4008h offset + (16d × i), where i=0d to 191d



CCM_CCGR*n*_CLR field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_CLR field descriptions (continued)

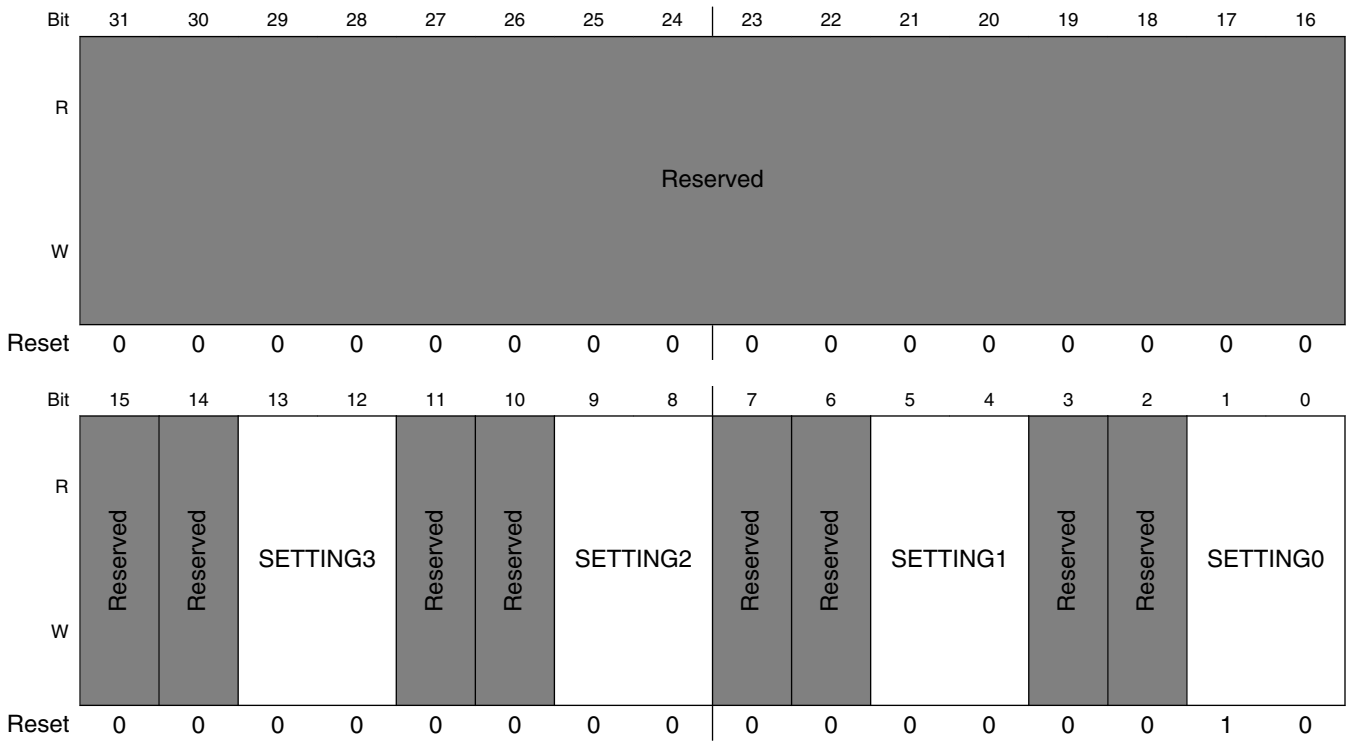
Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

5.1.7.9 CCM Clock Gating Register (CCM_CCGR*n*_TOG)

NOTE

Not all CCGRs are mapped. See CCGR Interface for CCGR mapping and clock gating information.

Address: 3038_0000h base + 400Ch offset + (16d × i), where i=0d to 191d



CCM_CCGR*n*_TOG field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 -	This field is reserved. Reserved
14 -	This field is reserved. Reserved
13–12 SETTING3	Clock gate control setting for domain 3. This field can only be written by domain 3 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

Table continues on the next page...

CCM_CCGR*n*_TOG field descriptions (continued)

Field	Description
11 -	This field is reserved. Reserved
10 -	This field is reserved. Reserved
9–8 SETTING2	Clock gate control setting for domain 2. This field can only be written by domain 2 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
7 -	This field is reserved. Reserved
6 -	This field is reserved. Reserved
5–4 SETTING1	Clock gate control setting for domain 1. This field can only be written by domain 1. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time
3 -	This field is reserved. Reserved
2 -	This field is reserved. Reserved
SETTING0	Clock gate control setting for domain 0. This field can only be written by domain 0. 00 Domain clocks not needed 01 Domain clocks needed when in RUN 10 Domain clocks needed when in RUN and WAIT 11 Domain clocks needed all the time

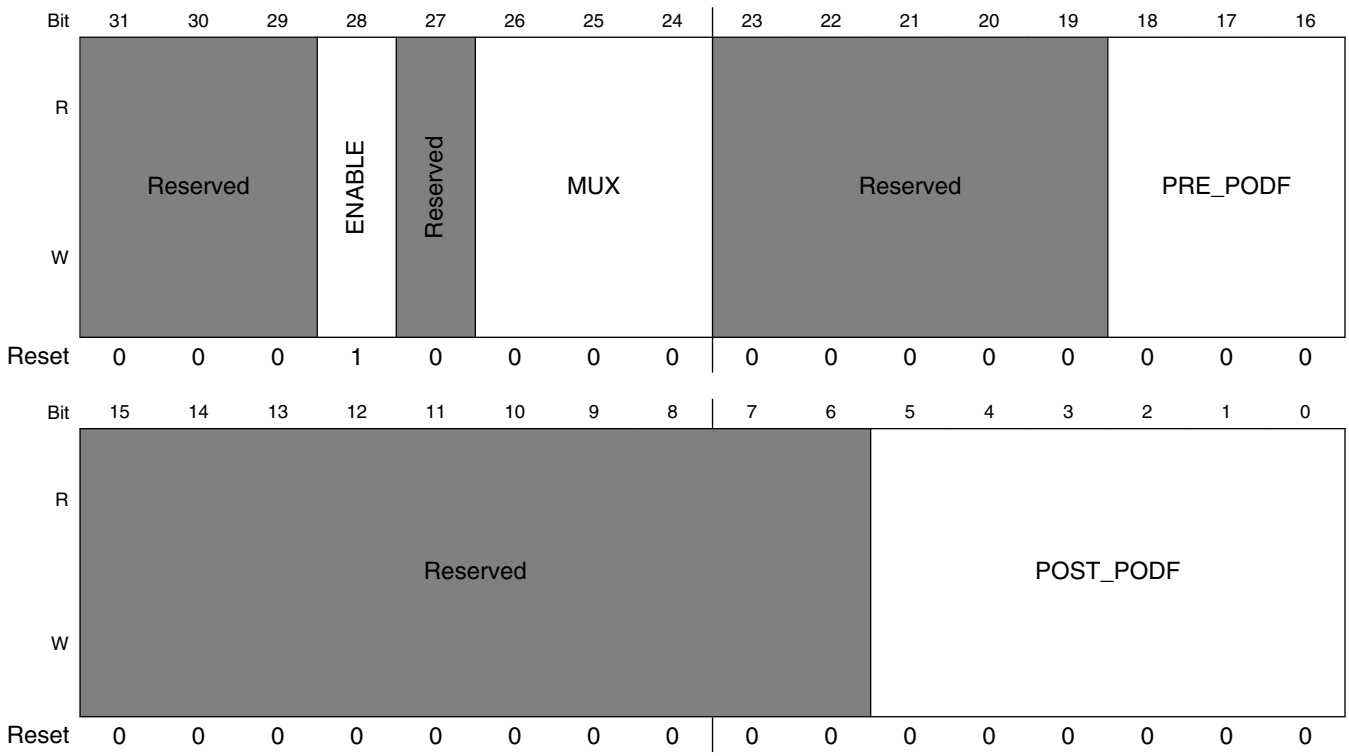
5.1.7.10 Target Register (CCM_TARGET_ROOT*n*)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8000h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n* field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n* field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1. For CORE, this field is 3 bit long. For IPG, this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

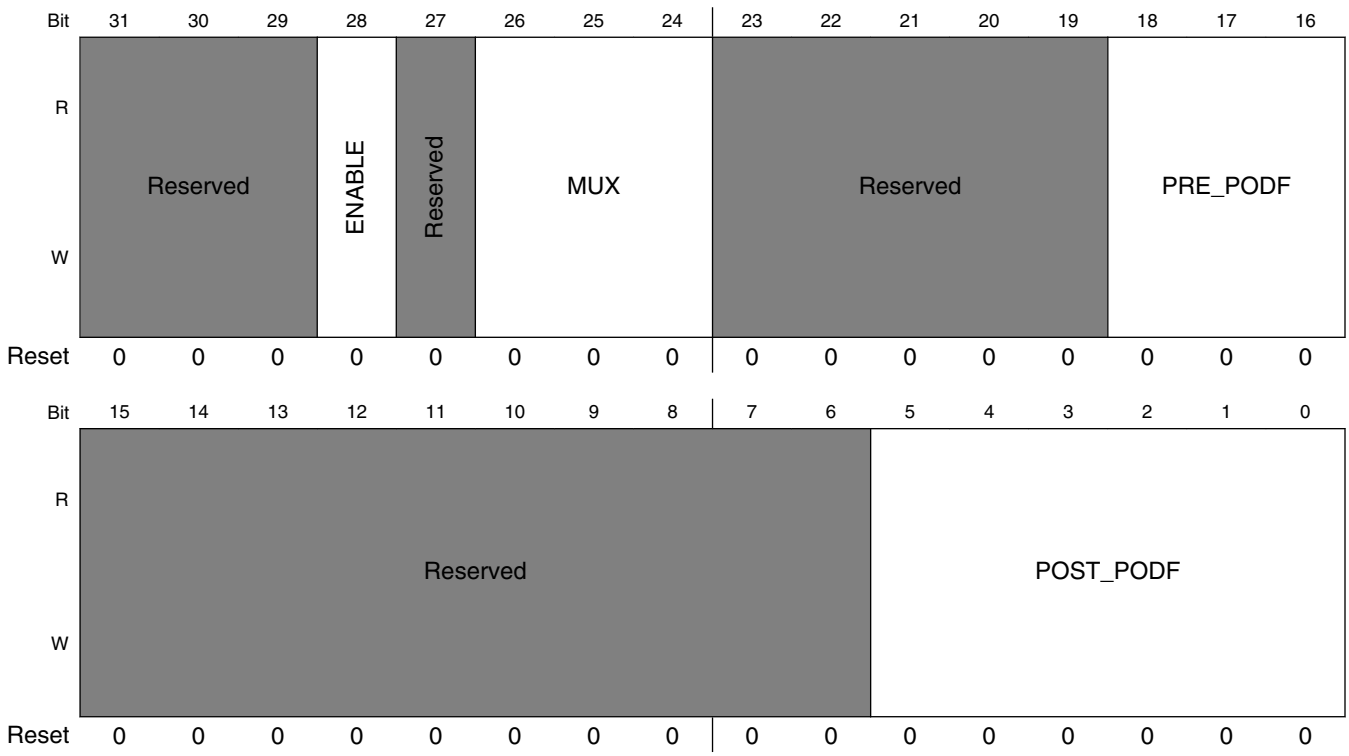
5.1.7.11 Target Register (CCM_TARGET_ROOT*n*_SET)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8004h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_SET field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n*_SET field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1. For CORE, this field is 3 bit long. For IPG, this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

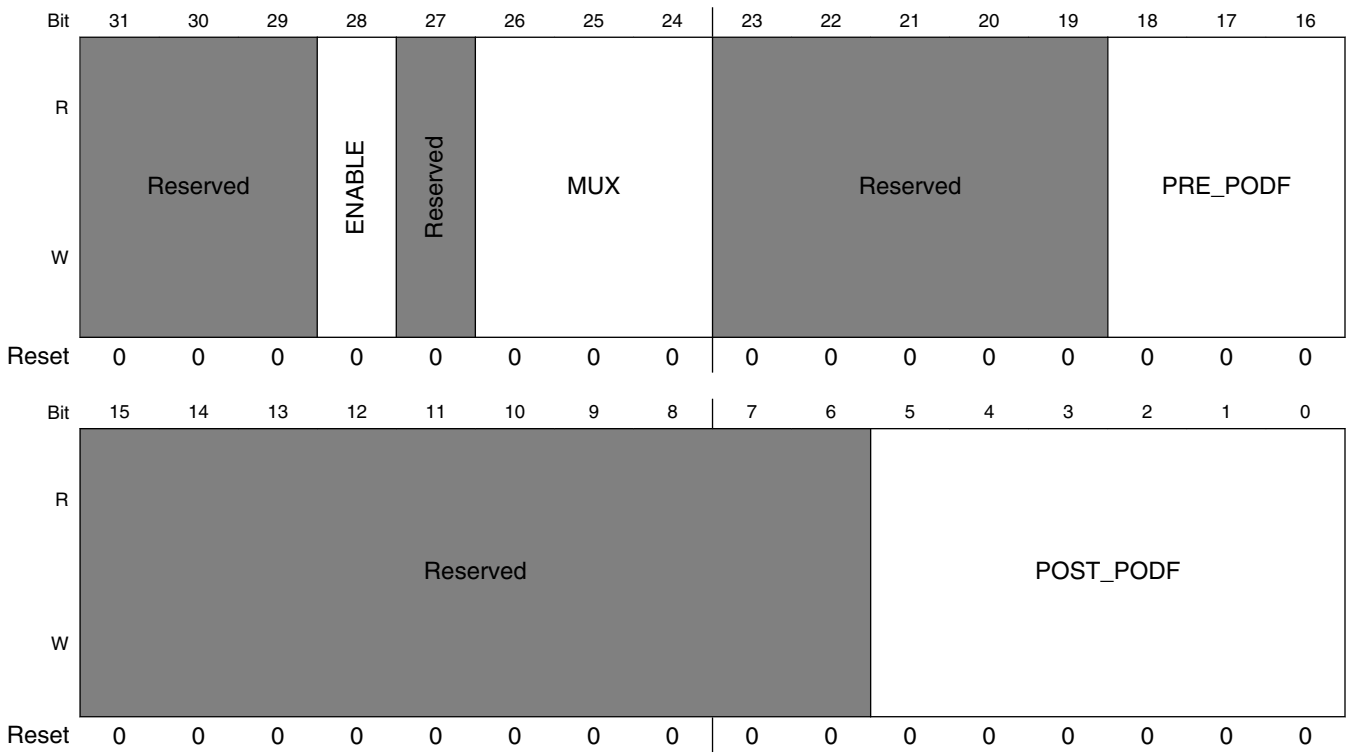
5.1.7.12 Target Register (CCM_TARGET_ROOT*n*_CLR)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 8008h offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_CLR field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources This field is 1 bit long for DRAM and CORE

Table continues on the next page...

CCM_TARGET_ROOT*n*_CLR field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divider divide the number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1. For CORE, this field is 3 bit long. For IPG, this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

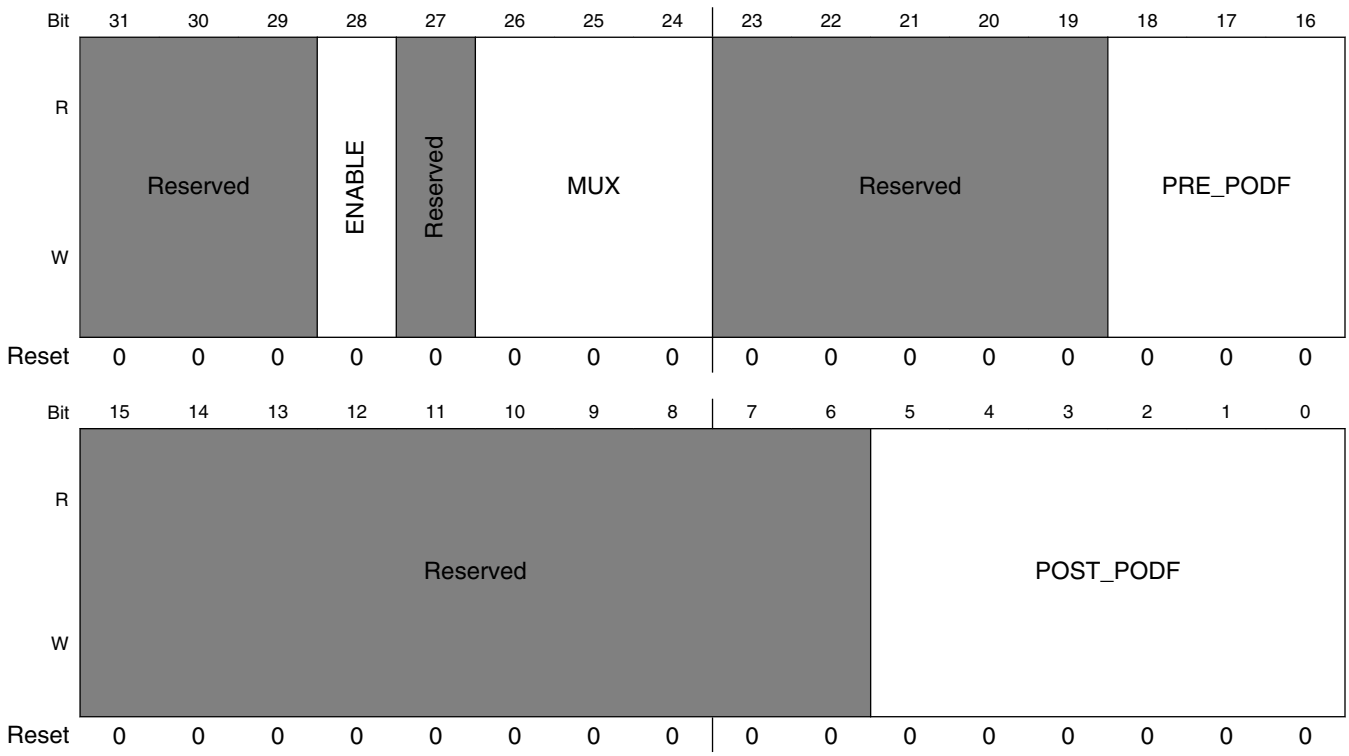
5.1.7.13 Target Register (CCM_TARGET_ROOT*n*_TOG)

See Target Interface for more information.

NOTE

See Clock Root Selects for clock root offsets and muxing information.

Address: 3038_0000h base + 800Ch offset + (128d × i), where i=0d to 141d



CCM_TARGET_ROOT*n*_TOG field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28 ENABLE	Enable this clock 0 clock root is OFF 1 clock root is ON
27 -	This field is reserved. Reserved
26–24 MUX	Selection of clock sources This field is 1 bit long for DRAM and CORE

Table continues on the next page...

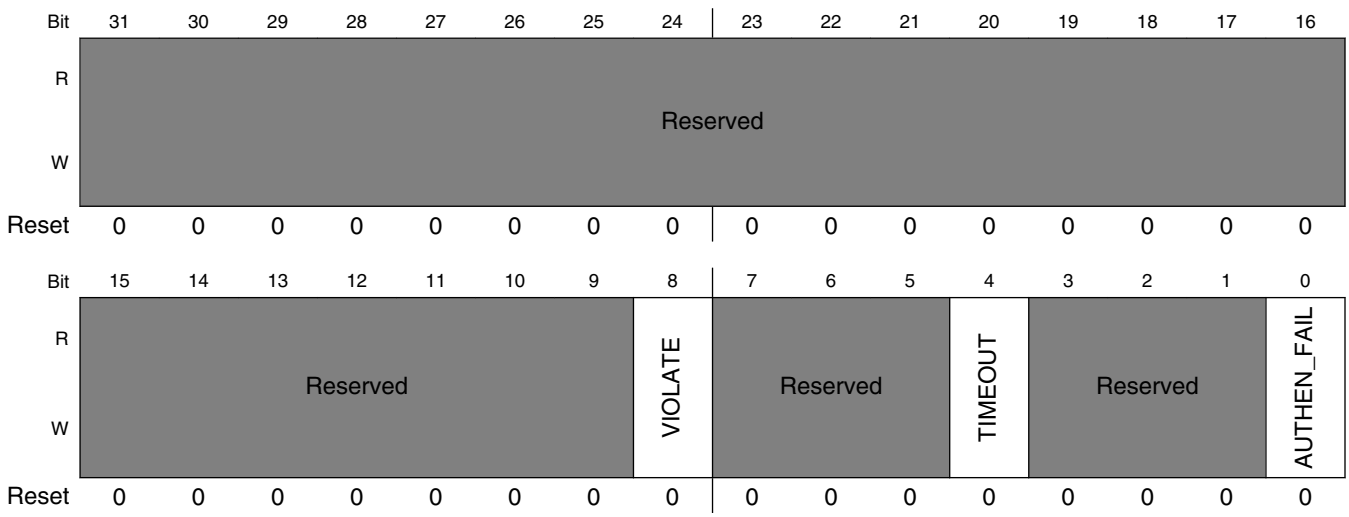
CCM_TARGET_ROOT*n*_TOG field descriptions (continued)

Field	Description
23–19 -	This field is reserved. Reserved
18–16 PRE_PODF	Pre divide divide number Divider value is n+1 This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15–6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1. For CORE, this field is 3 bit long. For IPG, this field is 1 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.14 Miscellaneous Register (CCM_MISC*n*)

MISC

Address: 3038_0000h base + 8010h offset + (128d × i), where i=0d to 141d



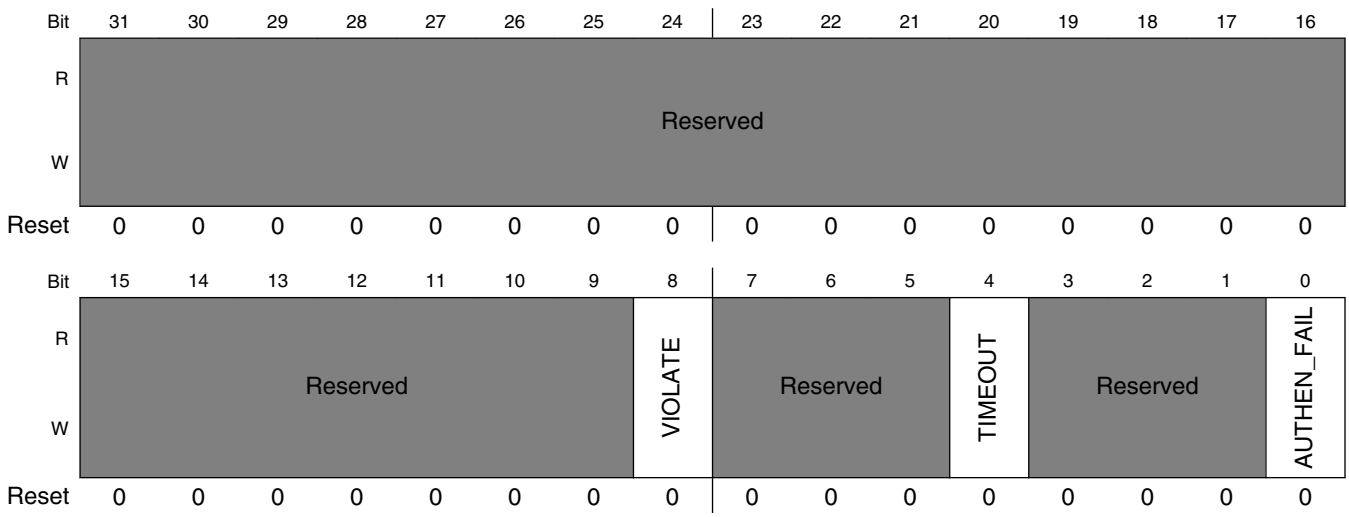
CCM_MISC*n* field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.15 Miscellaneous Register (CCM_MISC_ROOT*n*_SET)

Misc

Address: 3038_0000h base + 8014h offset + (128d × i), where i=0d to 141d



CCM_MISC_ROOT*n*_SET field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.16 Miscellaneous Register (CCM_MISC_ROOT*n*_CLR)

MISC

Address: 3038_0000h base + 8018h offset + (128d × i), where i=0d to 141d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							VIOLATE	Reserved			TIMEOUT	Reserved			AUTHEN_FAIL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CCM_MISC_ROOT*n*_CLR field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.17 Miscellaneous Register (CCM_MISC_ROOT*n*_TOG)

MISC

Address: 3038_0000h base + 801Ch offset + (128d × i), where i=0d to 141d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							VIOLATE	Reserved			TIMEOUT	Reserved			AUTHEN_FAIL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

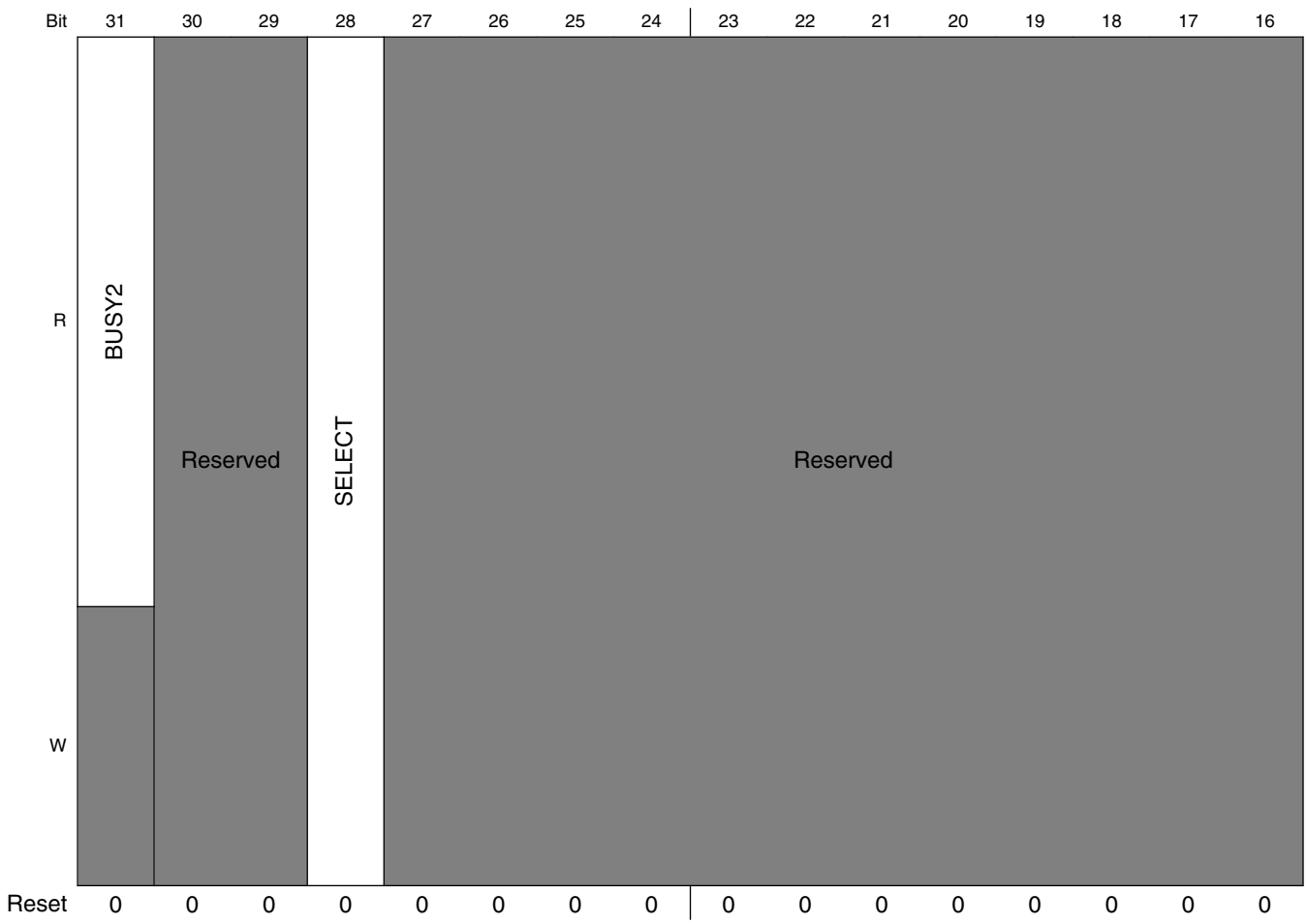
CCM_MISC_ROOT*n*_TOG field descriptions

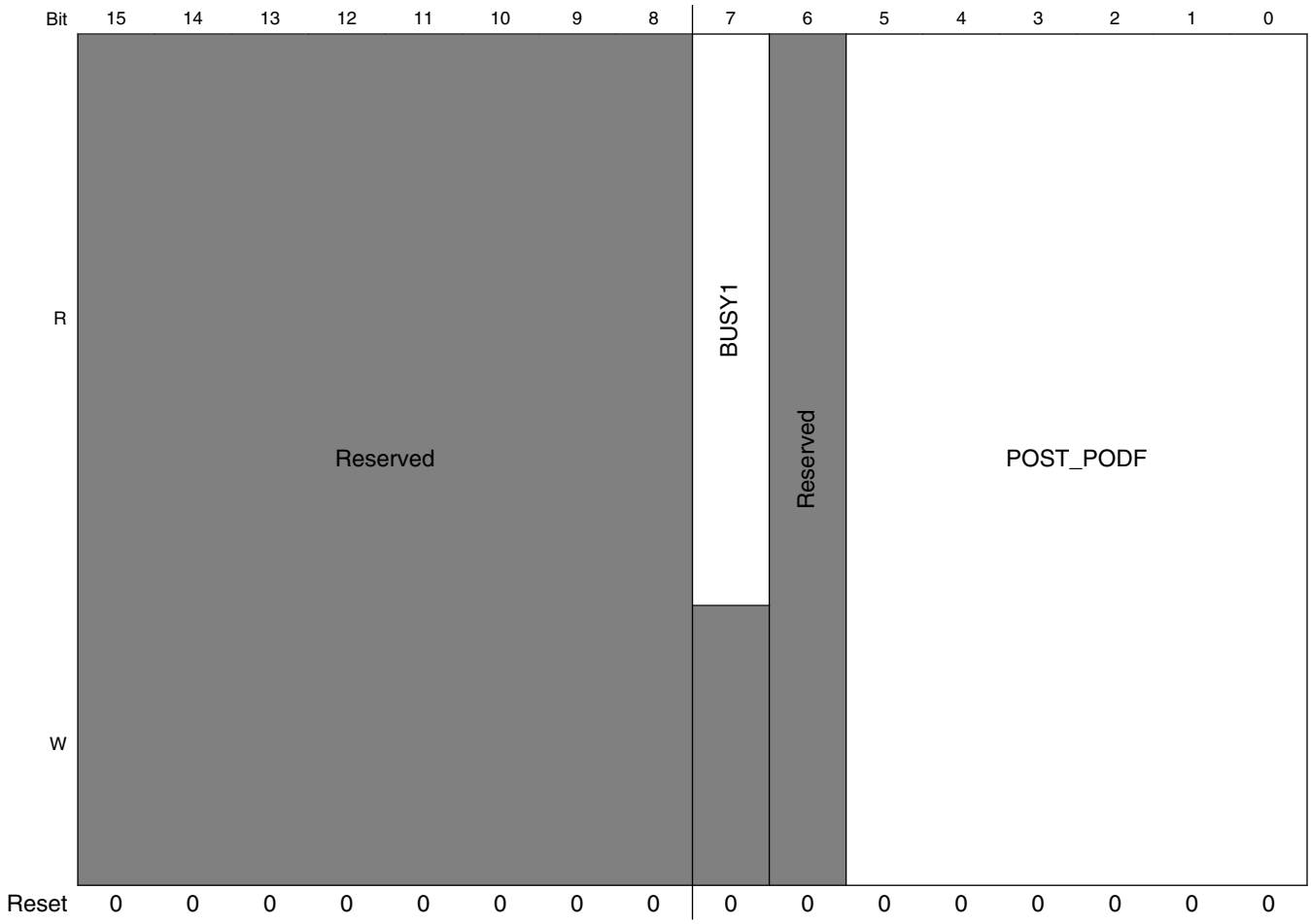
Field	Description
31–9 -	This field is reserved. Reserved
8 VIOLATE	This sticky bit reflects access violation in normal interface of this clock. This bit has internal 4 bits, one for each domain. Violation from other domain is not visible or clearable. This file is cleared to 0 while write 1.
7–5 -	This field is reserved. Reserved
4 TIMEOUT	This sticky bit reflects time out happened during accessing this clock. This bit has internal 4 bits, one for each domain. Timeout from other domain is not visible or clearable. This file is cleared to 0 while write 1.
3–1 -	This field is reserved. Reserved
0 AUTHEN_FAIL	This sticky bit reflects access restricted by access control of this clock. This bit has internal 4 bits, one for each domain. Authentic fail from other domain is not visible or clearable. This file is cleared to 0 while write 1

5.1.7.18 Post Divider Register (CCM_POST*n*)

Post Register

Address: 3038_0000h base + 8020h offset + (128d × i), where i=0d to 141d





CCM_POST*n* field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of pre clock branches This field is not applied in IP 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1

Table continues on the next page...

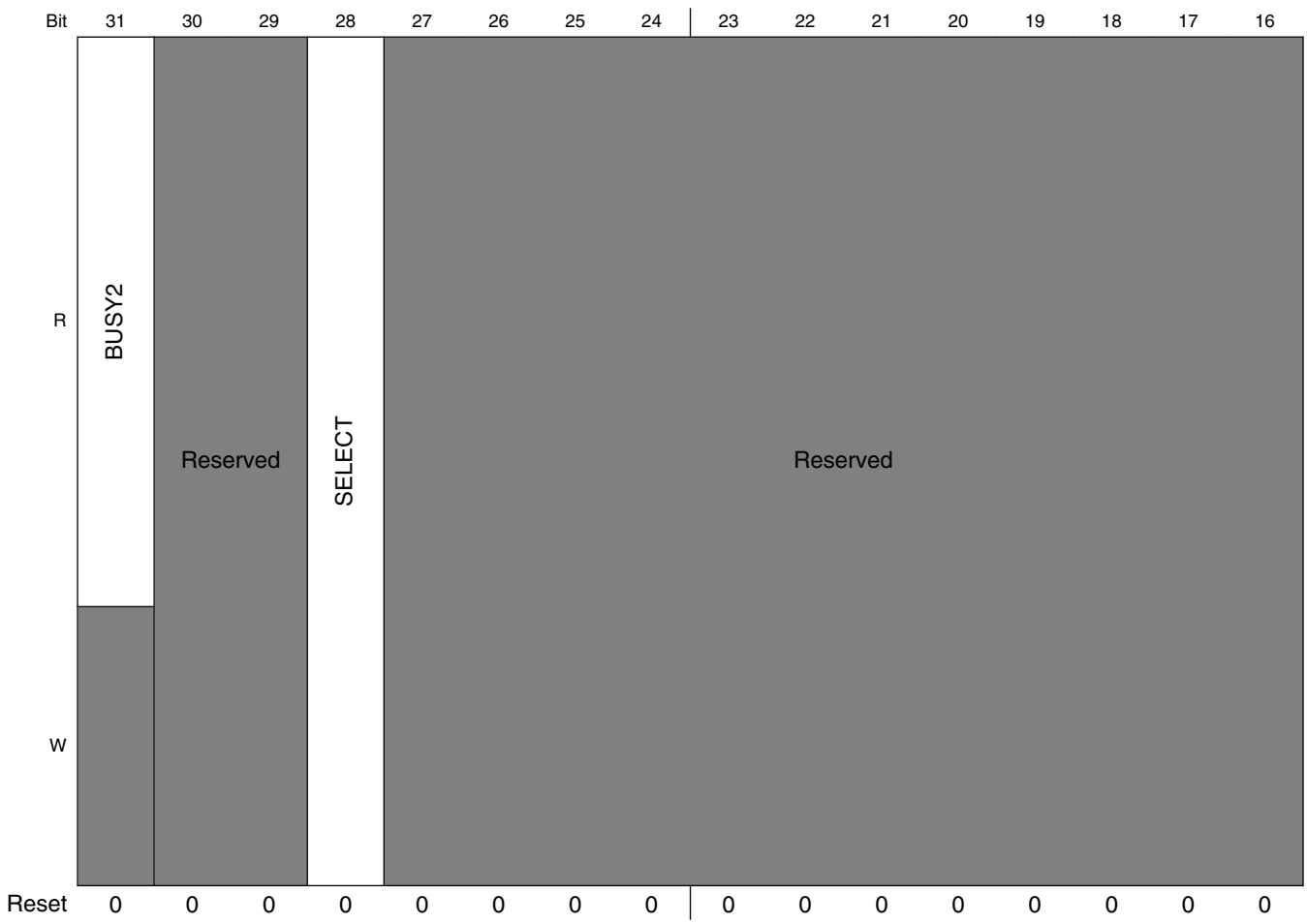
CCM_POST*n* field descriptions (continued)

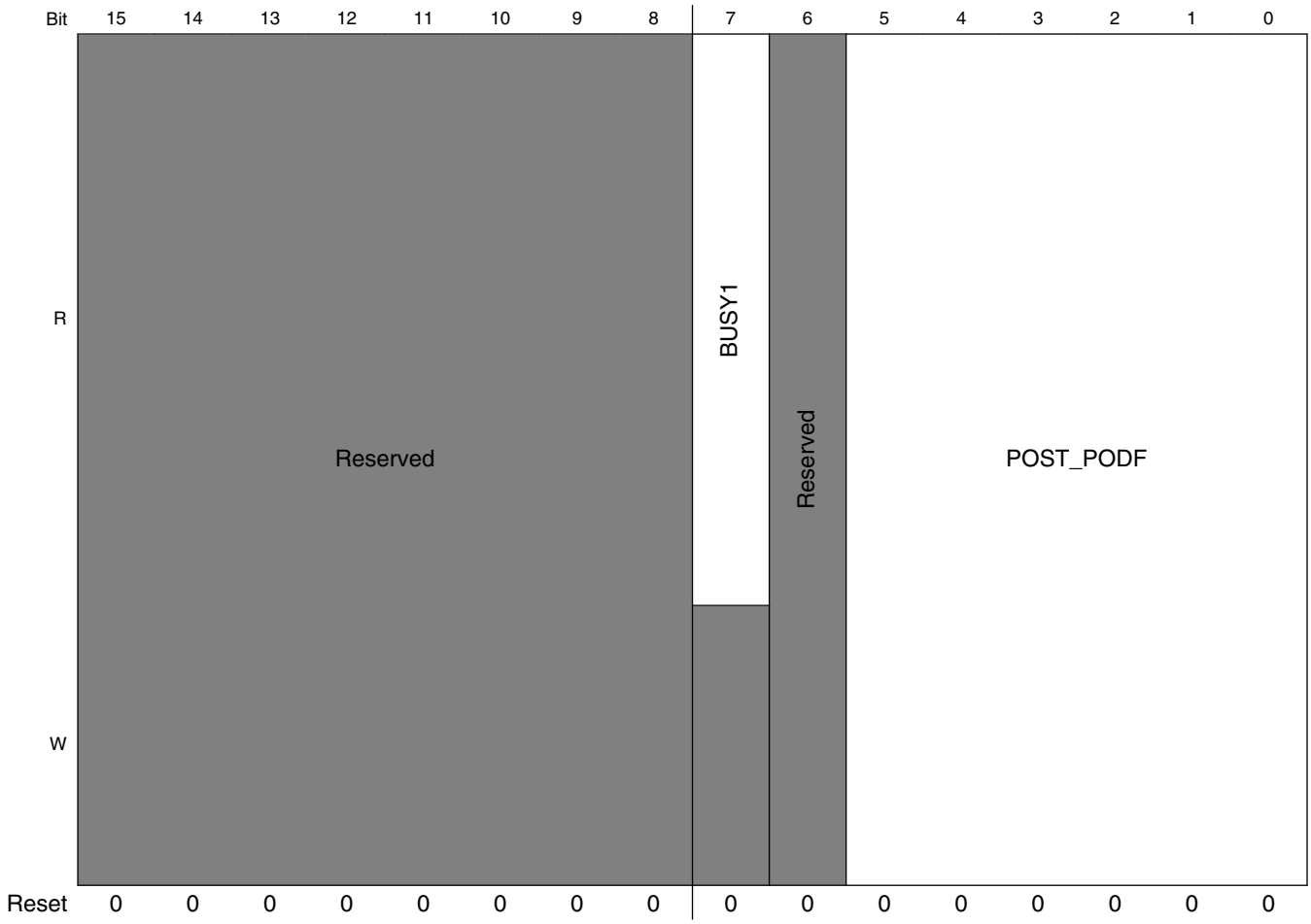
Field	Description
	For CORE, this field is 3 bit long. For IPG, this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.19 Post Divider Register (CCM_POST_ROOT*n*_SET)

Post Divider Register

Address: 3038_0000h base + 8024h offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_SET field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of pre clock branches This field is not applied in IP 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1

Table continues on the next page...

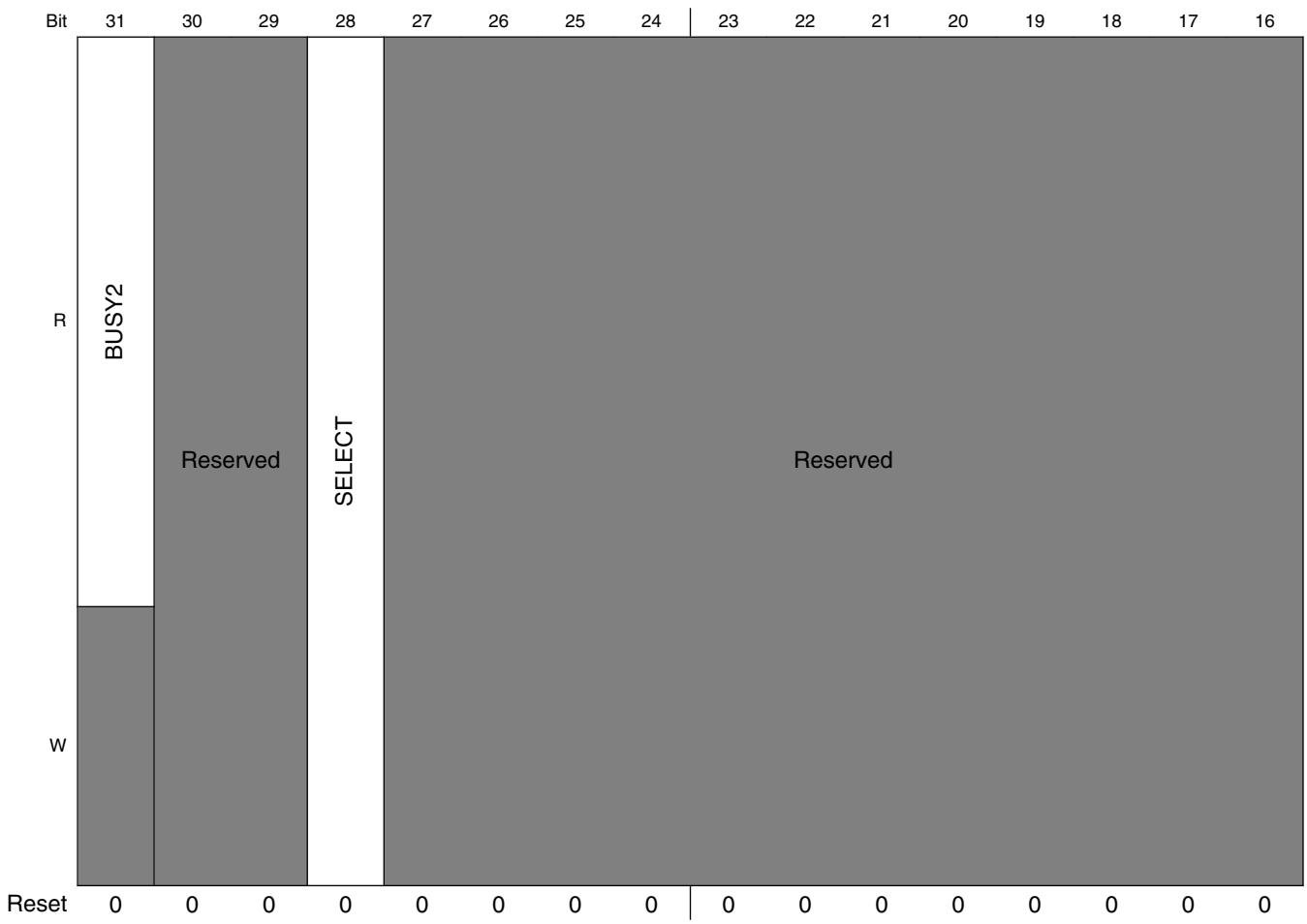
CCM_POST_ROOT*n*_SET field descriptions (continued)

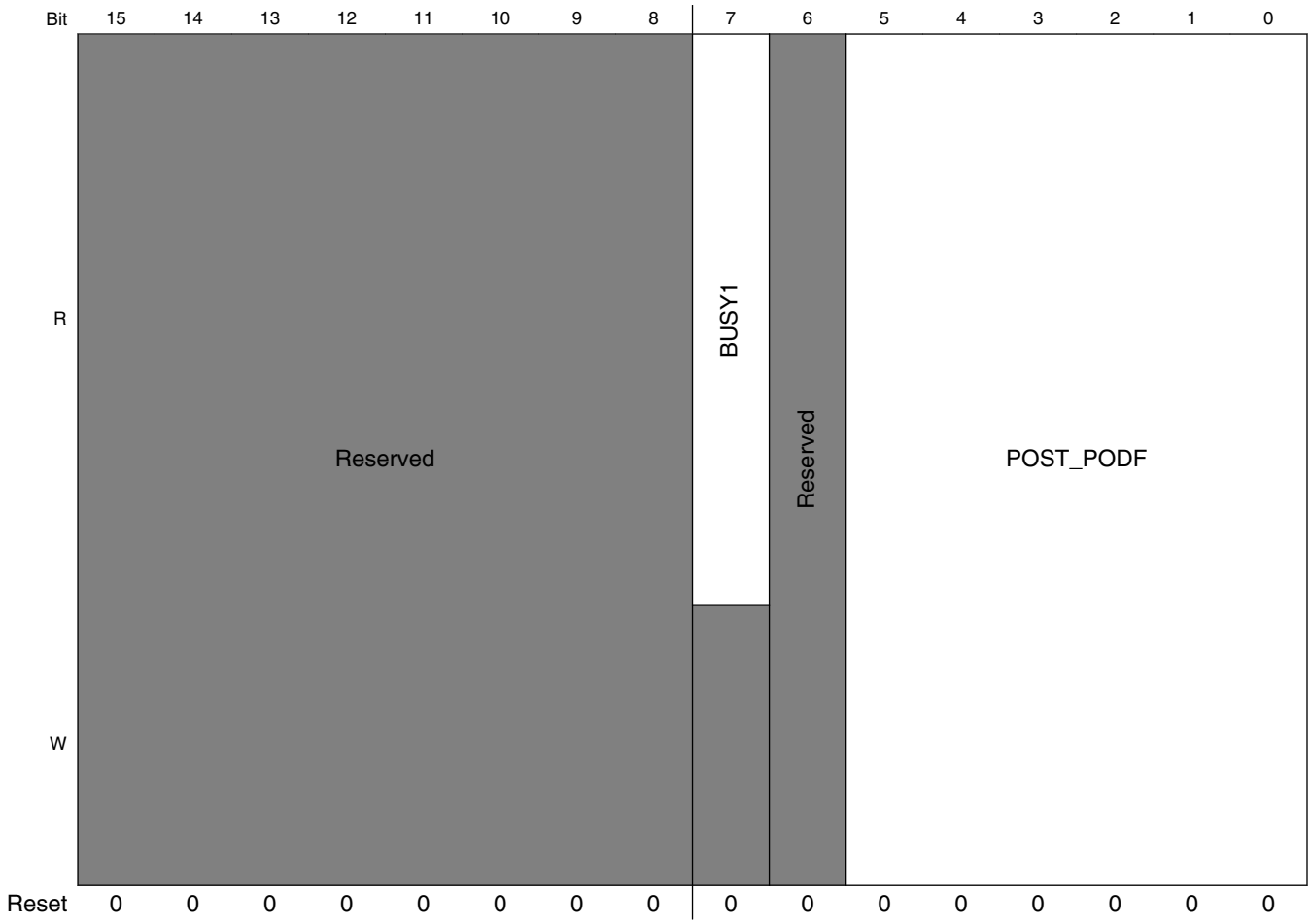
Field	Description
	For CORE, this field is 3 bit long. For IPG, this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.20 Post Divider Register (CCM_POST_ROOT*n*_CLR)

Post Root Register

Address: 3038_0000h base + 8028h offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_CLR field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of pre clock branches This field is not applied in IP 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide the number Divider value is n + 1

Table continues on the next page...

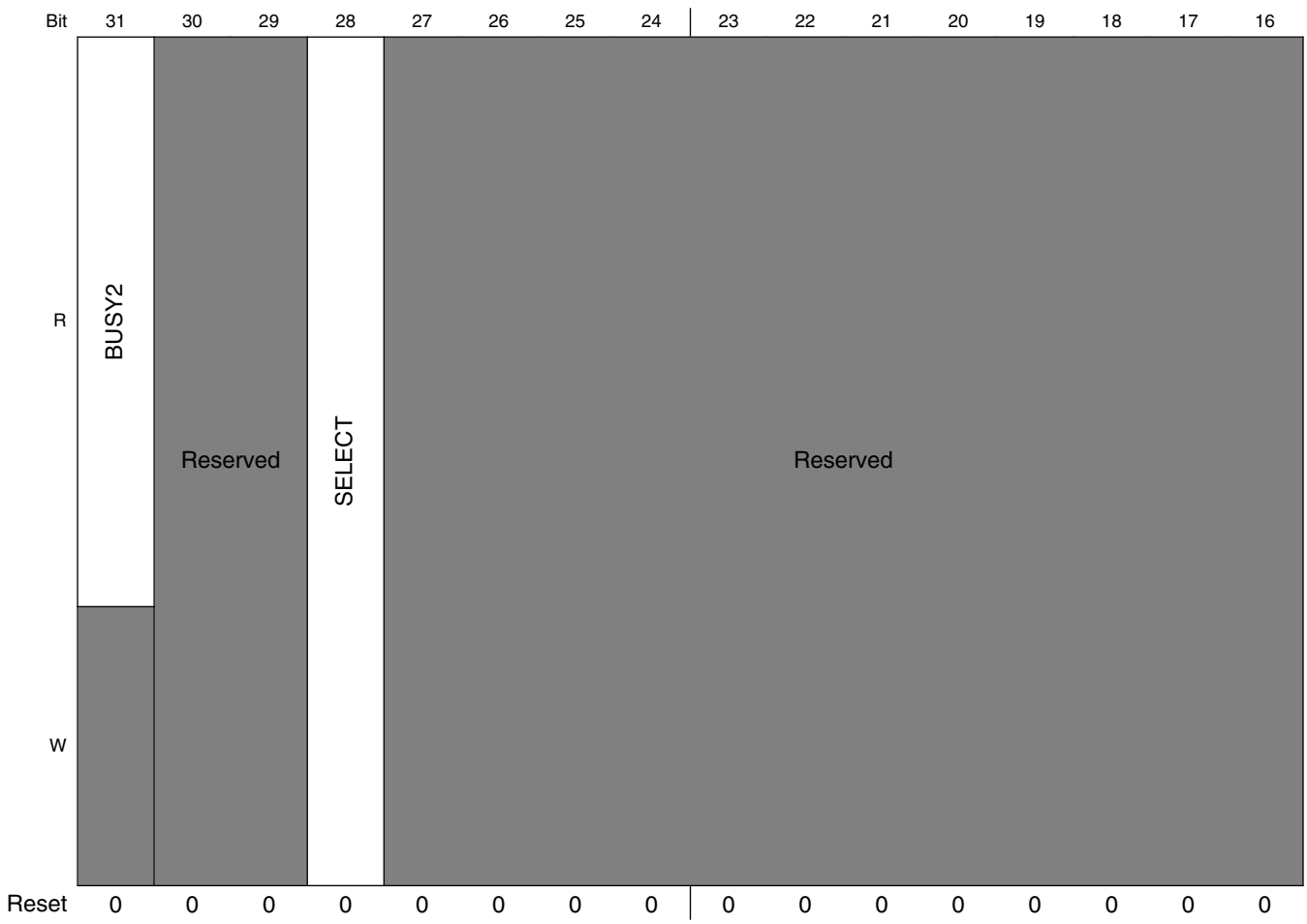
CCM_POST_ROOT*n*_CLR field descriptions (continued)

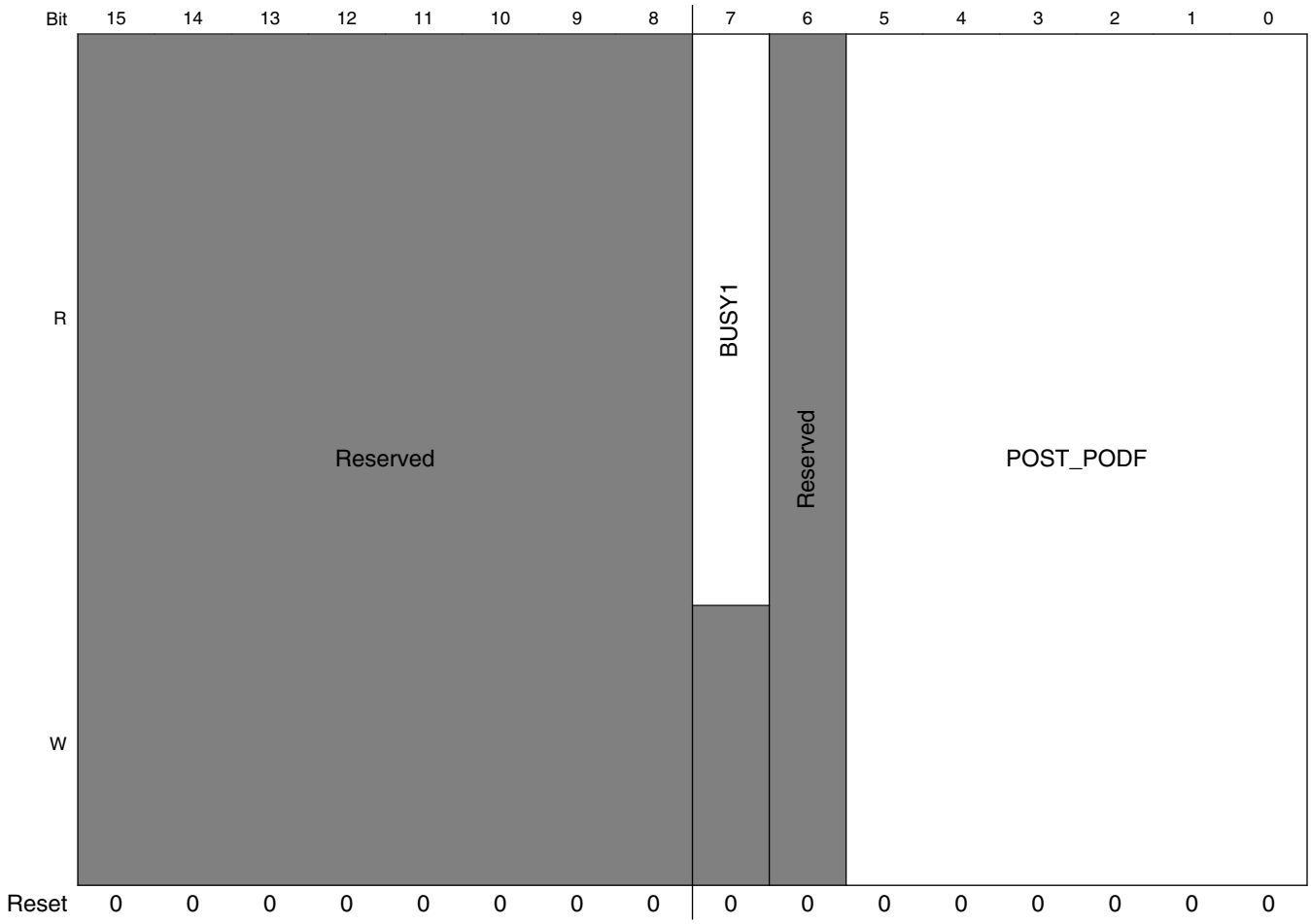
Field	Description
	For CORE, this field is 3 bit long. For IPG, this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.21 Post Divider Register (CCM_POST_ROOT*n*_TOG)

Post Root Register

Address: 3038_0000h base + 802Ch offset + (128d × i), where i=0d to 141d





CCM_POST_ROOT*n*_TOG field descriptions

Field	Description
31 BUSY2	Clock switching multiplexer is applying new setting
30–29 -	This field is reserved. Reserved
28 SELECT	Selection of pre clock branches This field is not applied in IP 0 select branch A 1 select branch B
27–8 -	This field is reserved. Reserved
7 BUSY1	Post divider is applying new set value
6 -	This field is reserved. Reserved
POST_PODF	Post divider divide number Divider value is n + 1

Table continues on the next page...

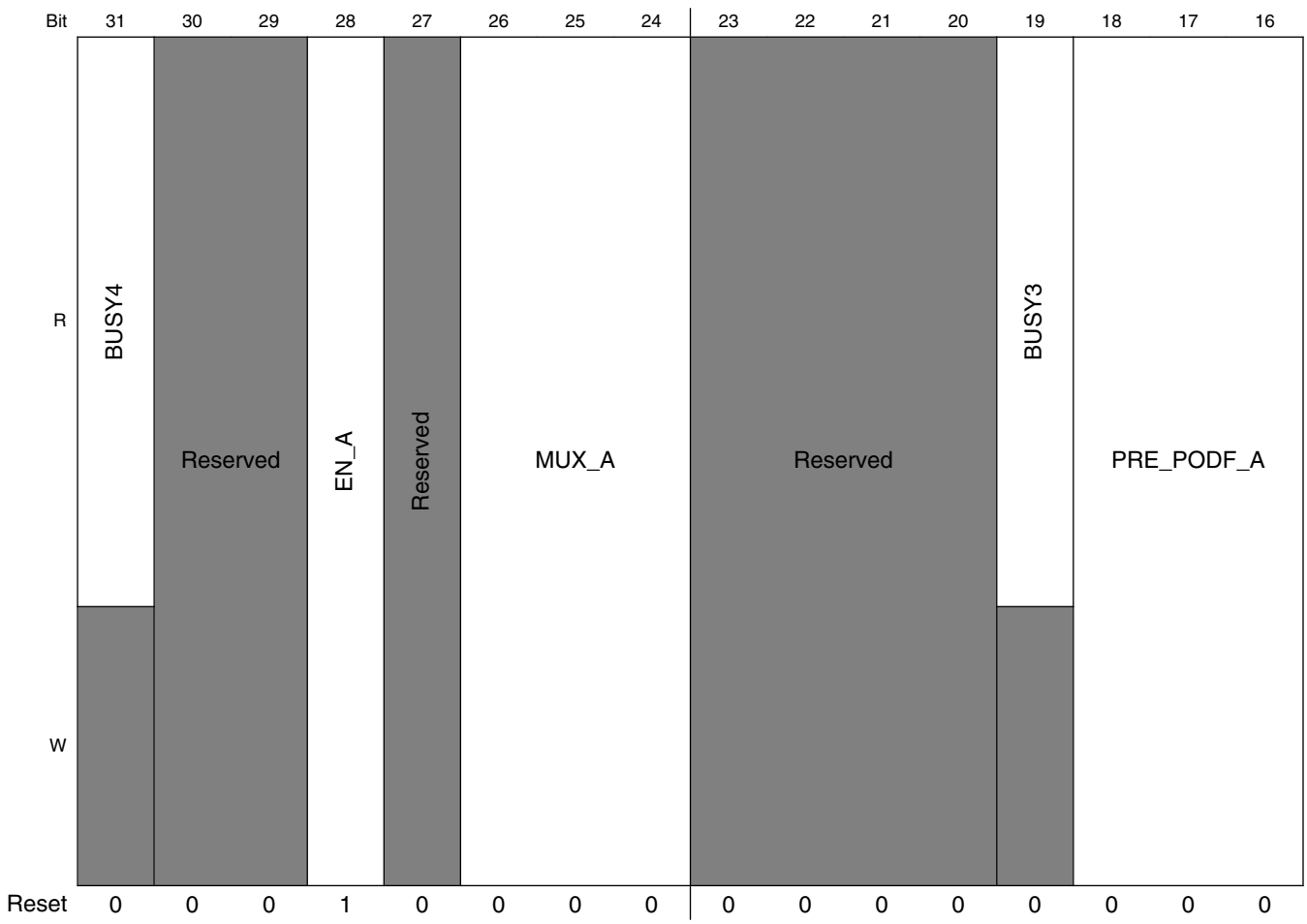
CCM_POST_ROOT*n*_TOG field descriptions (continued)

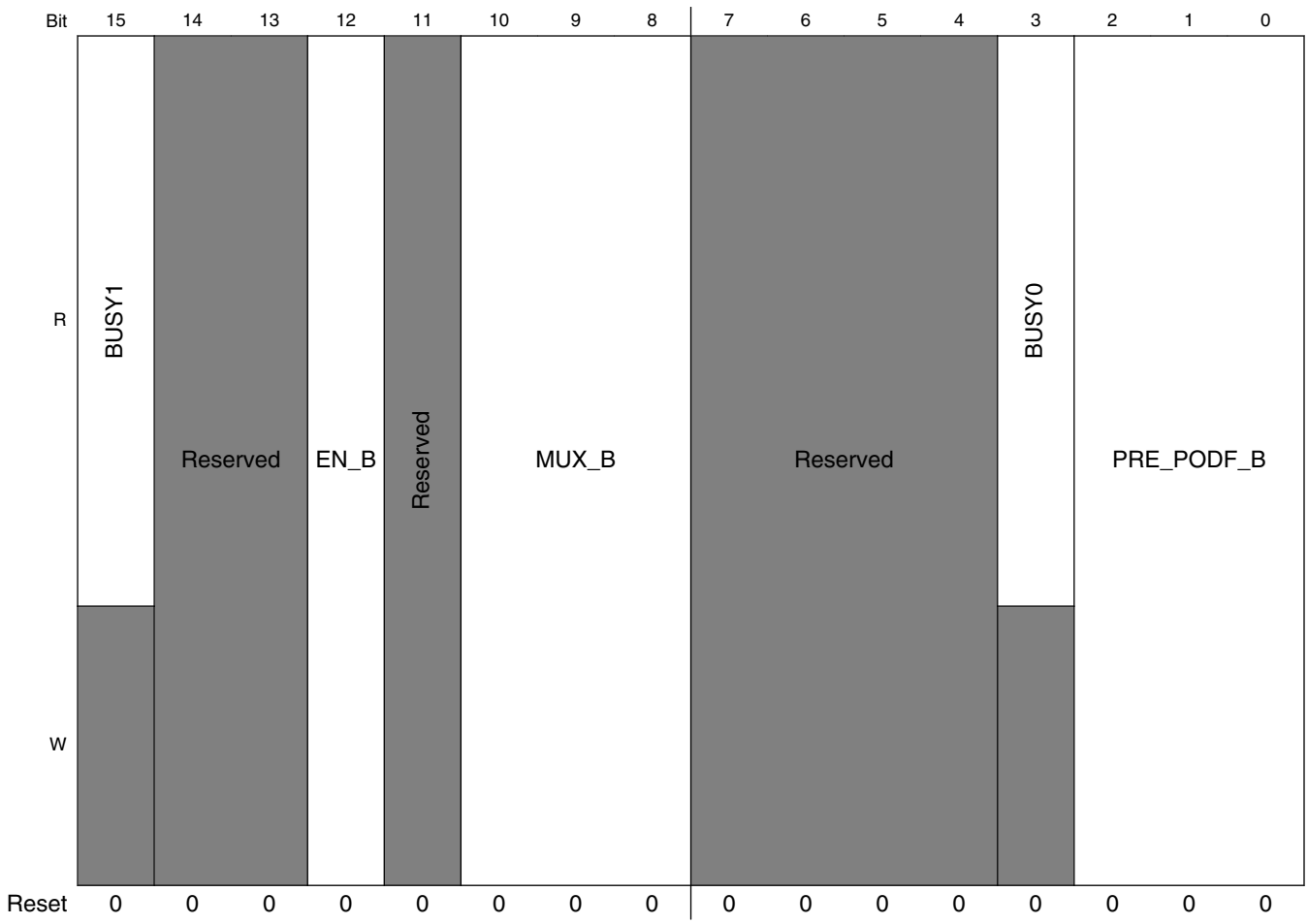
Field	Description
	For CORE, this field is 3 bit long. For IPG, this field is 2 bit long. This field does not apply to DRAM_PHYM 000000 Divide by 1 000001 Divide by 2 000010 Divide by 3 000011 Divide by 4 000100 Divide by 5 000101 Divide by 6 : 111111 Divide by 64

5.1.7.22 Pre Divider Register (CCM_PRE*n*)

Pre Register

Address: 3038_0000h base + 8030h offset + (128d × i), where i=0d to 141d





CCM_PRE*n* field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

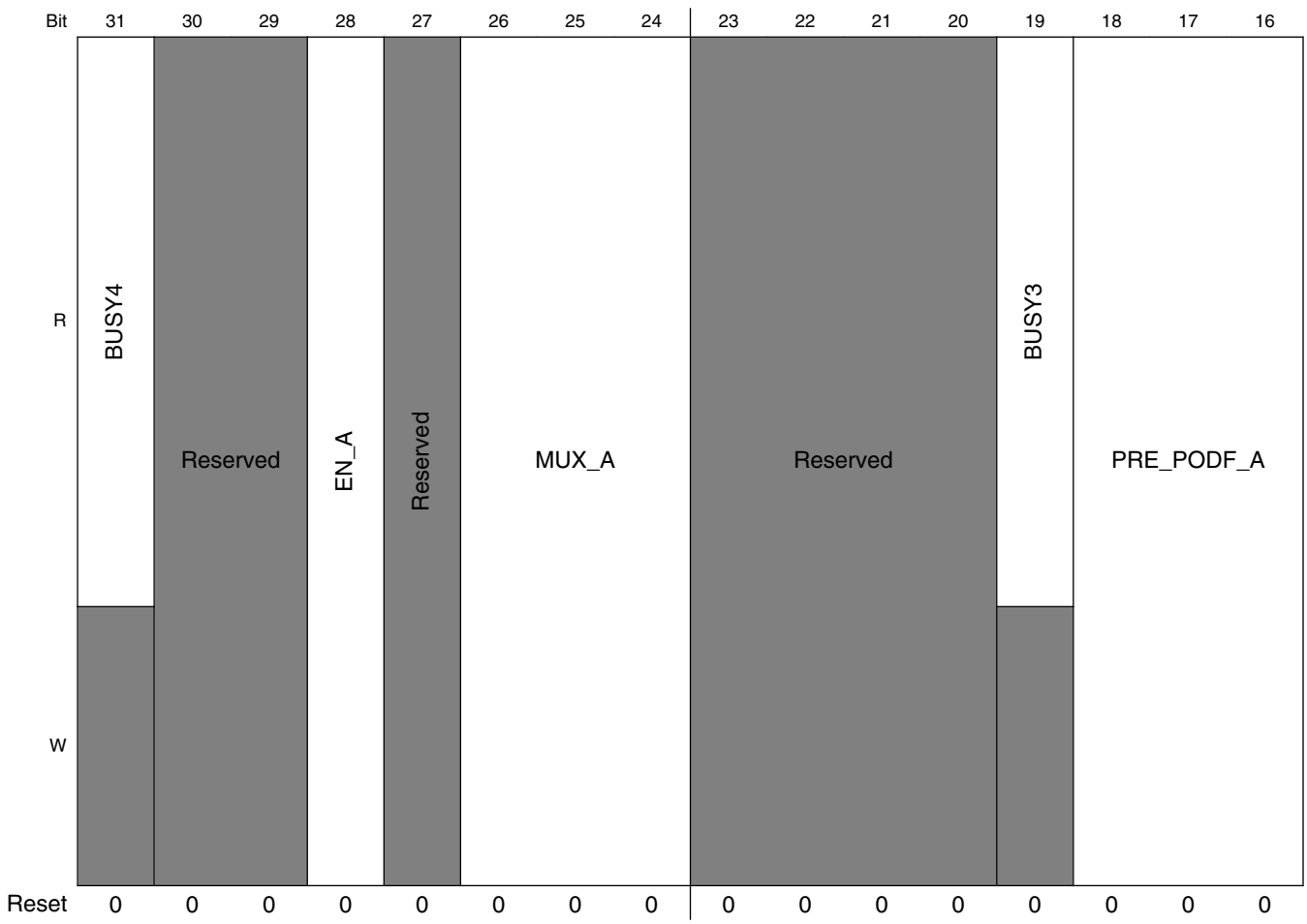
CCM_PRE*n* field descriptions (continued)

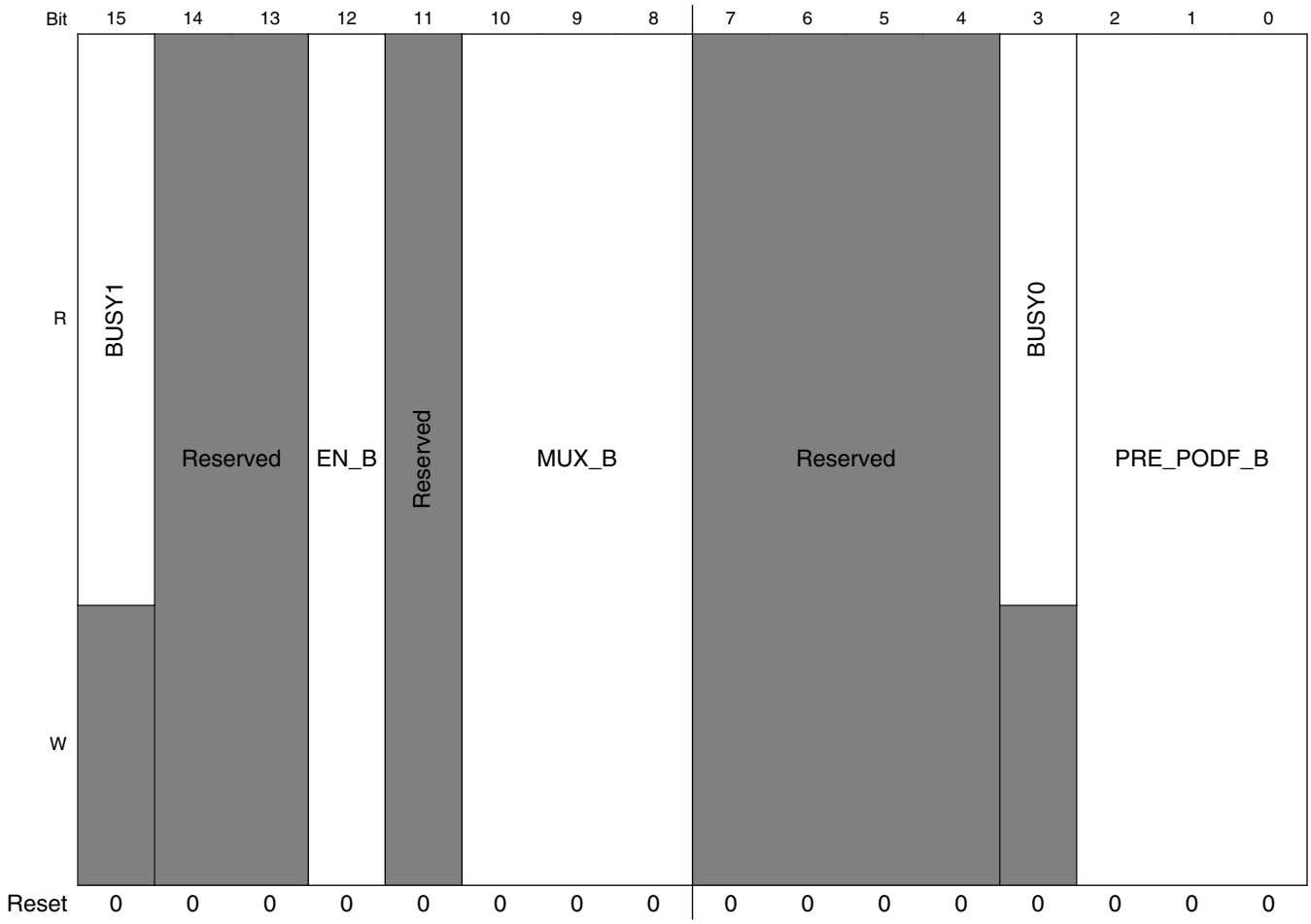
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch a is applying field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.23 Pre Divider Register (CCM_PRE_ROOT*n*_SET)

Pre Divider Register

Address: 3038_0000h base + 8034h offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_SET field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

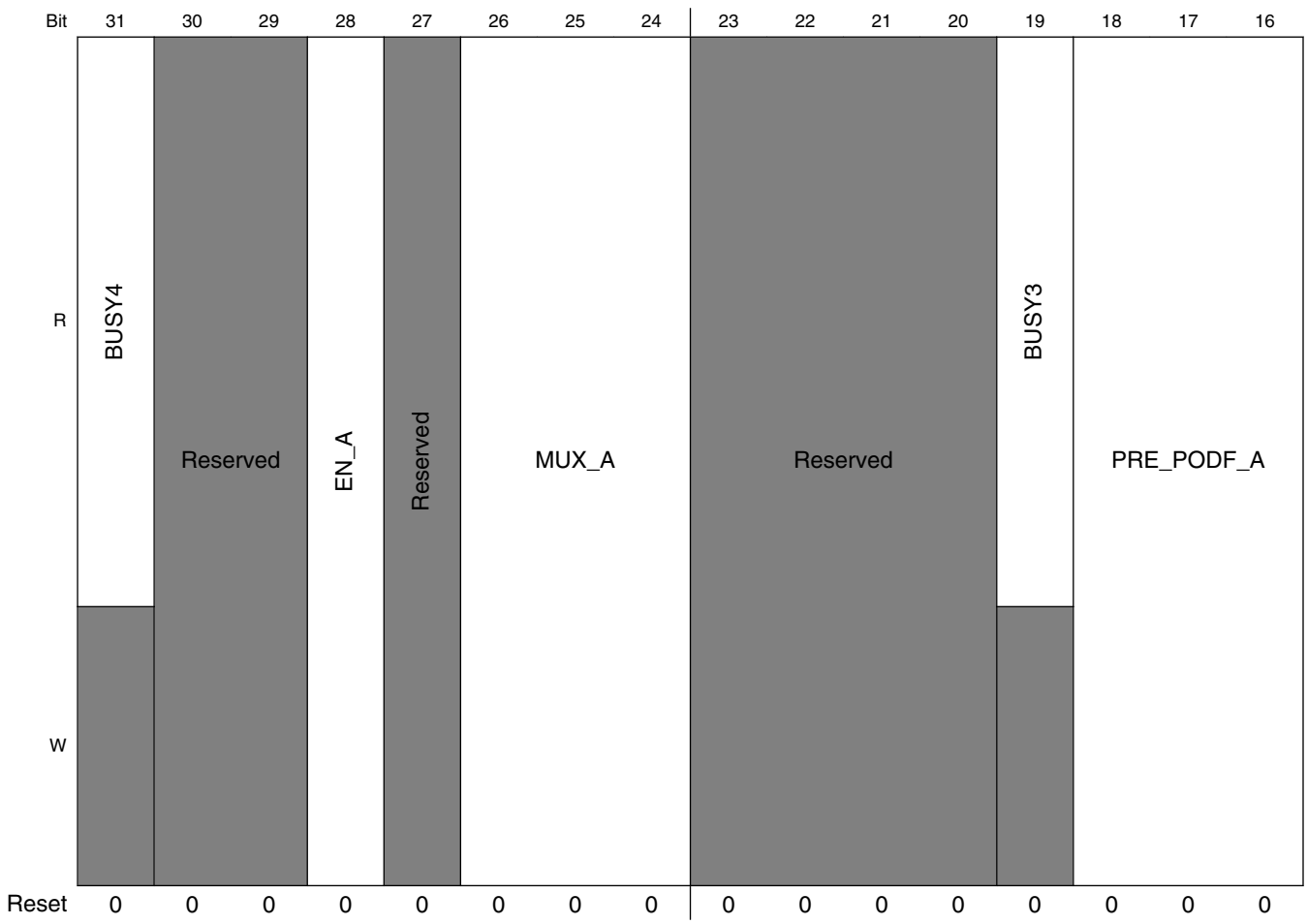
CCM_PRE_ROOT*n*_SET field descriptions (continued)

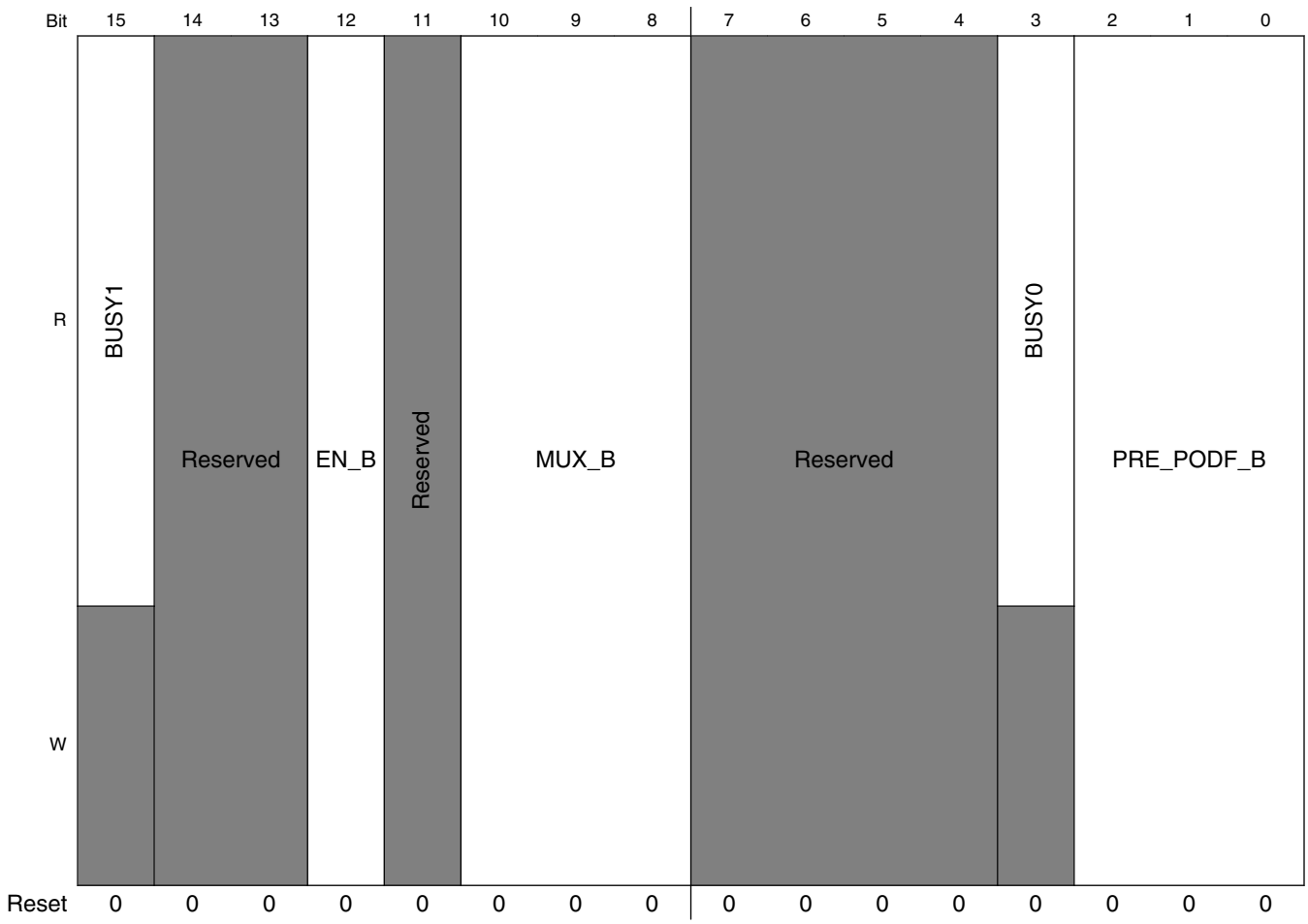
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch A is applying field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.24 Pre Divider Register (CCM_PRE_ROOT*n*_CLR)

Pree Root Register

Address: 3038_0000h base + 8038h offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_CLR field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

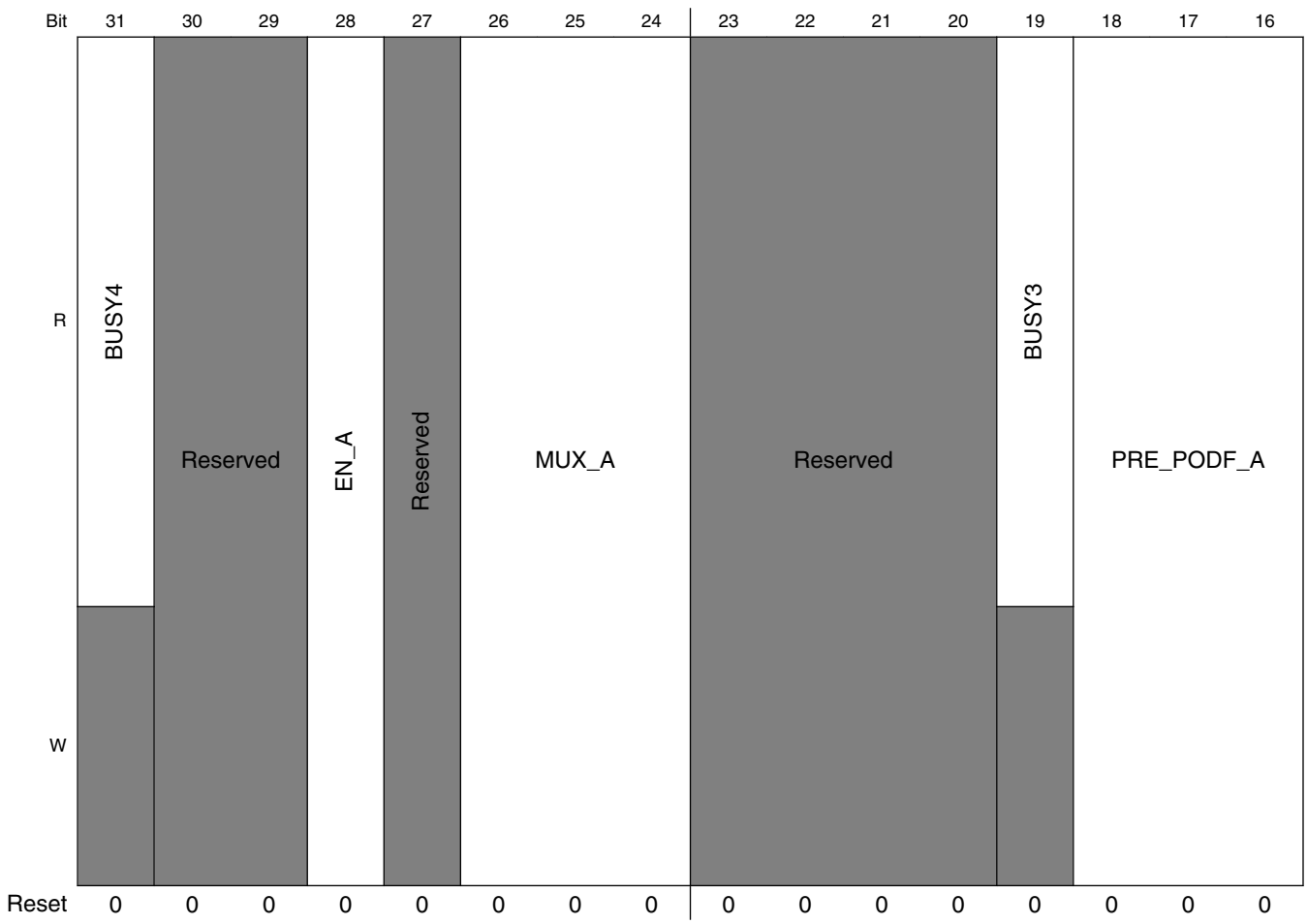
CCM_PRE_ROOT*n*_CLR field descriptions (continued)

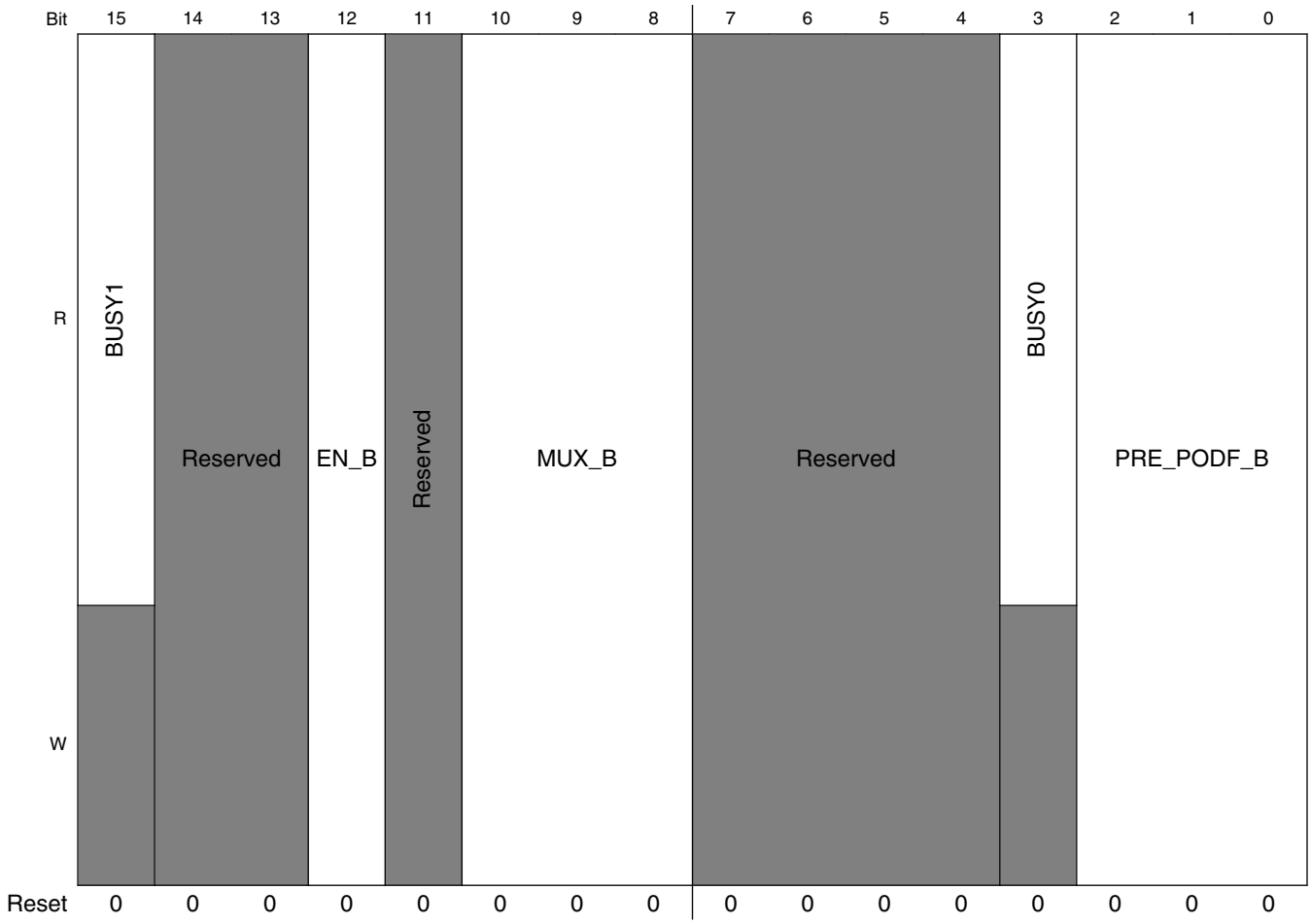
Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch A is applied This field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.25 Pre Divider Register (CCM_PRE_ROOT*n*_TOG)

Pre Root Register

Address: 3038_0000h base + 803Ch offset + (128d × i), where i=0d to 141d





CCM_PRE_ROOT*n*_TOG field descriptions

Field	Description
31 BUSY4	EN_A field is applied to field This field applies to DRAM and DRAM_PHYM
30–29 -	This field is reserved. Reserved
28 EN_A	Branch A clock gate control This field applies to DRAM and DRAM_PHYM 0 Clock shutdown 1 clock ON
27 -	This field is reserved. Reserved
26–24 MUX_A	Selection control of multiplexer of branch A This field applies to DRAM and DRAM_PHYM
23–20 -	This field is reserved. Reserved
19 BUSY3	Pre divider value for branch A is applied This field applies to DRAM and DRAM_PHYM

Table continues on the next page...

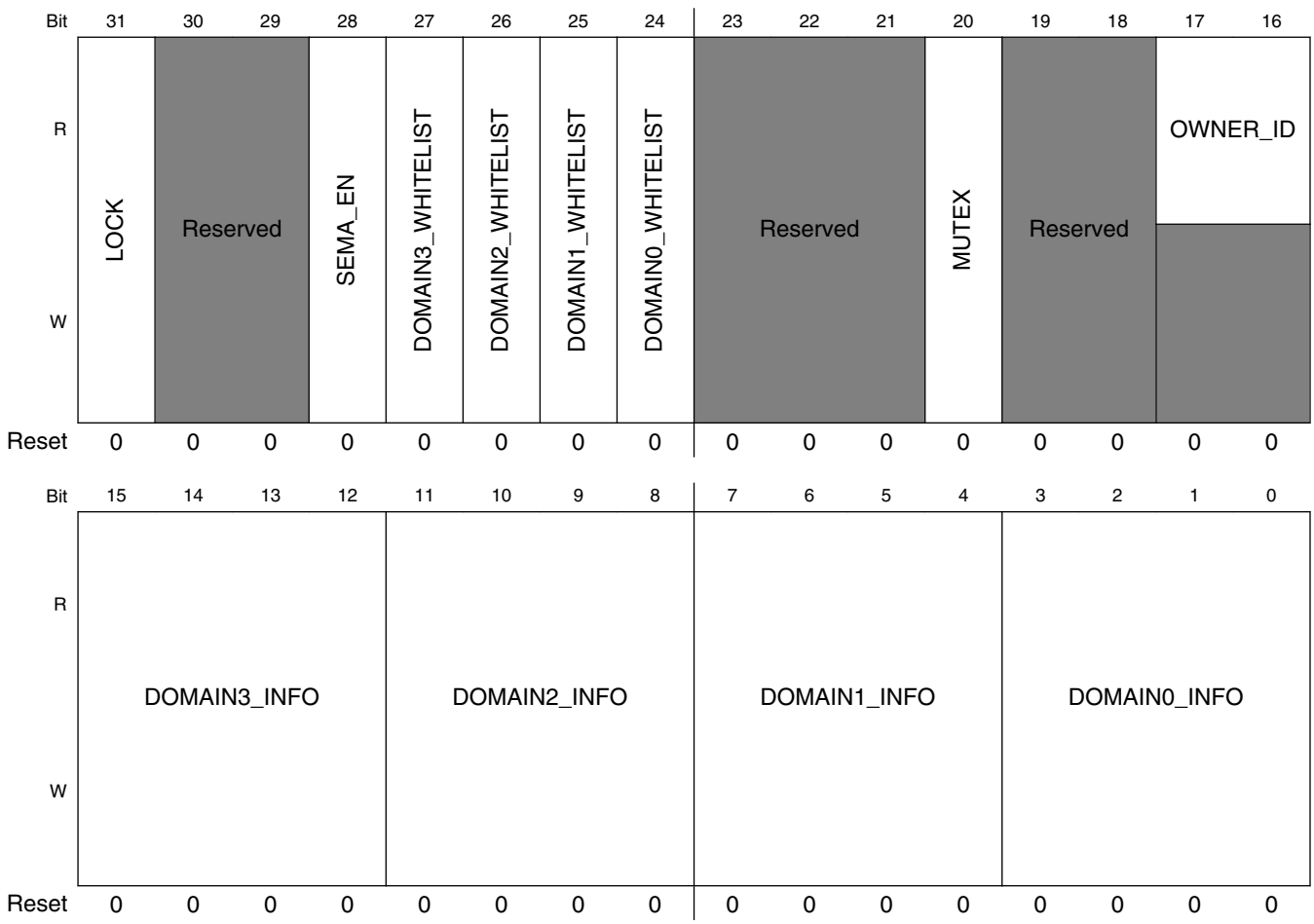
CCM_PRE_ROOT*n*_TOG field descriptions (continued)

Field	Description
18–16 PRE_PODF_A	Pre divider divide number for branch A Divider value is n + 1. This field does not apply for CORE, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 BUSY1	EN_B is applied to field This field does not apply for CORE, IP,DRAM, DRAM_PHYM
14–13 -	This field is reserved. Reserved
12 EN_B	Branch B clock gate control This field does not apply for CORE, IP,DRAM, DRAM_PHYM 0 Clock shutdown 1 Clock ON
11 -	This field is reserved. Reserved
10–8 MUX_B	Selection control of multiplexer of branch B This field does not apply for CORE, IP,DRAM, DRAM_PHYM
7–4 -	This field is reserved. Reserved
3 BUSY0	Pre divider value for branch a is applied field does not apply for CORE, IP,DRAM, DRAM_PHYM
PRE_PODF_B	Pre divider divide number for branch B Divider value is n + 1. This field does not apply for CORE, IP, DRAM, DRAM_PHYM 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.7.26 Access Control Register (CCM_ACCESS_CTRL*n*)

Access Control Register

Address: 3038_0000h base + 8070h offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL*n* field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1

Table continues on the next page...

CCM_ACCESS_CTRL*n* field descriptions (continued)

Field	Description
	0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1

Table continues on the next page...

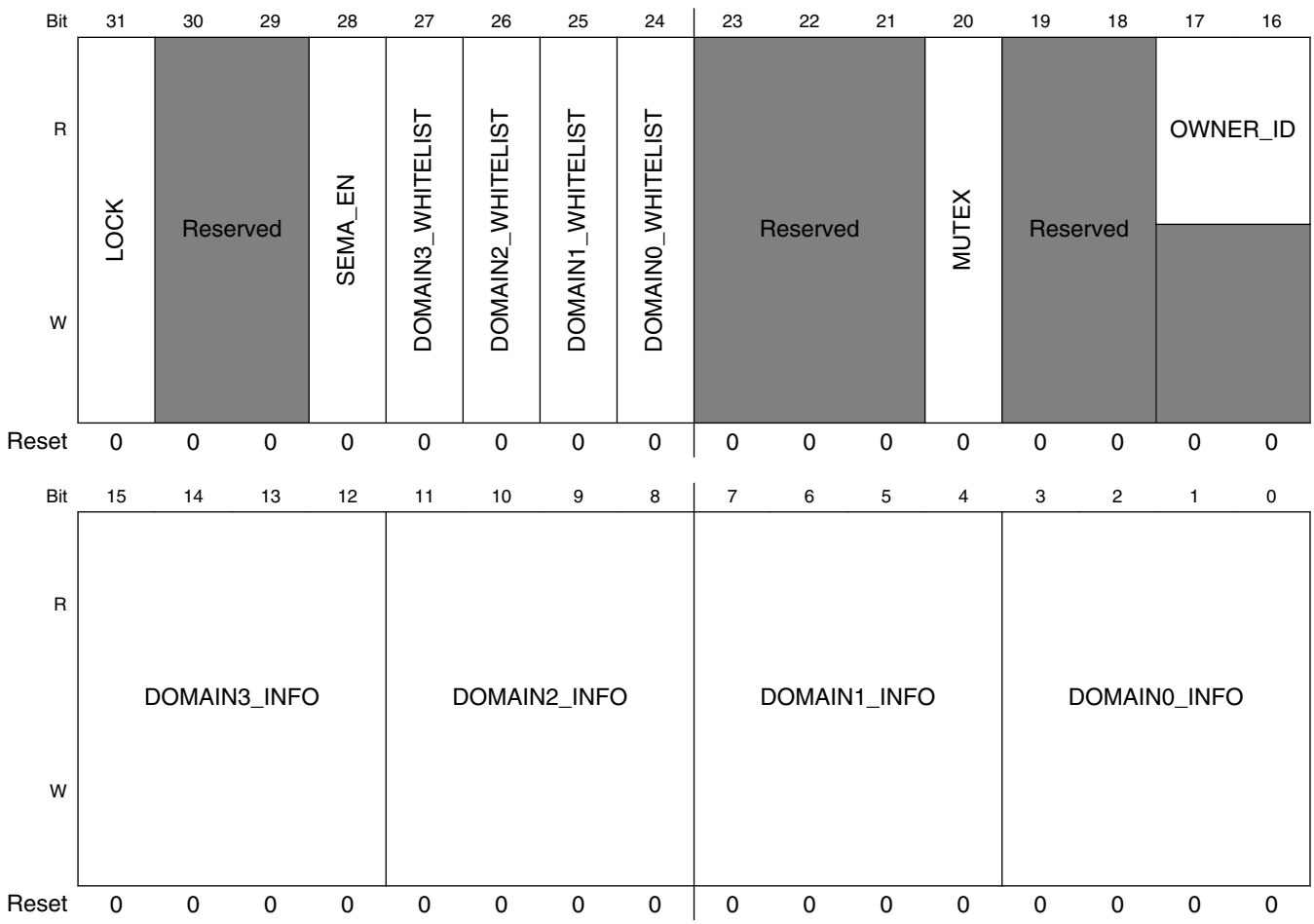
CCM_ACCESS_CTRL*n* field descriptions (continued)

Field	Description
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.27 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_SET)

Access Control Register

Address: 3038_0000h base + 8074h offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL_ROOT*n*_SET field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset.

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_SET field descriptions (continued)

Field	Description
	0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3

Table continues on the next page...

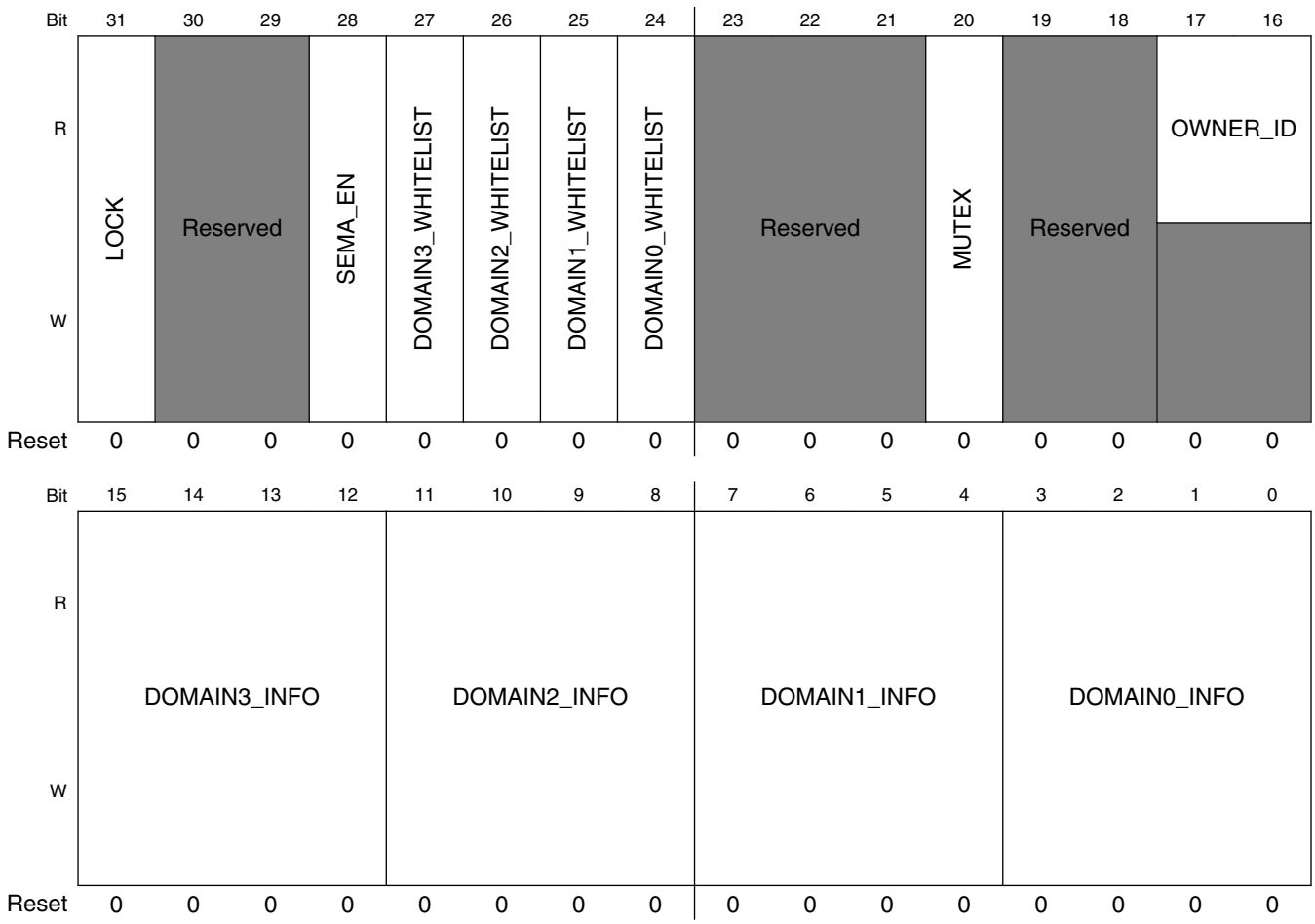
CCM_ACCESS_CTRL_ROOT*n*_SET field descriptions (continued)

Field	Description
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.28 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_CLR)

Access Control Register

Address: 3038_0000h base + 8078h offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions (continued)

Field	Description
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1 2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2

Table continues on the next page...

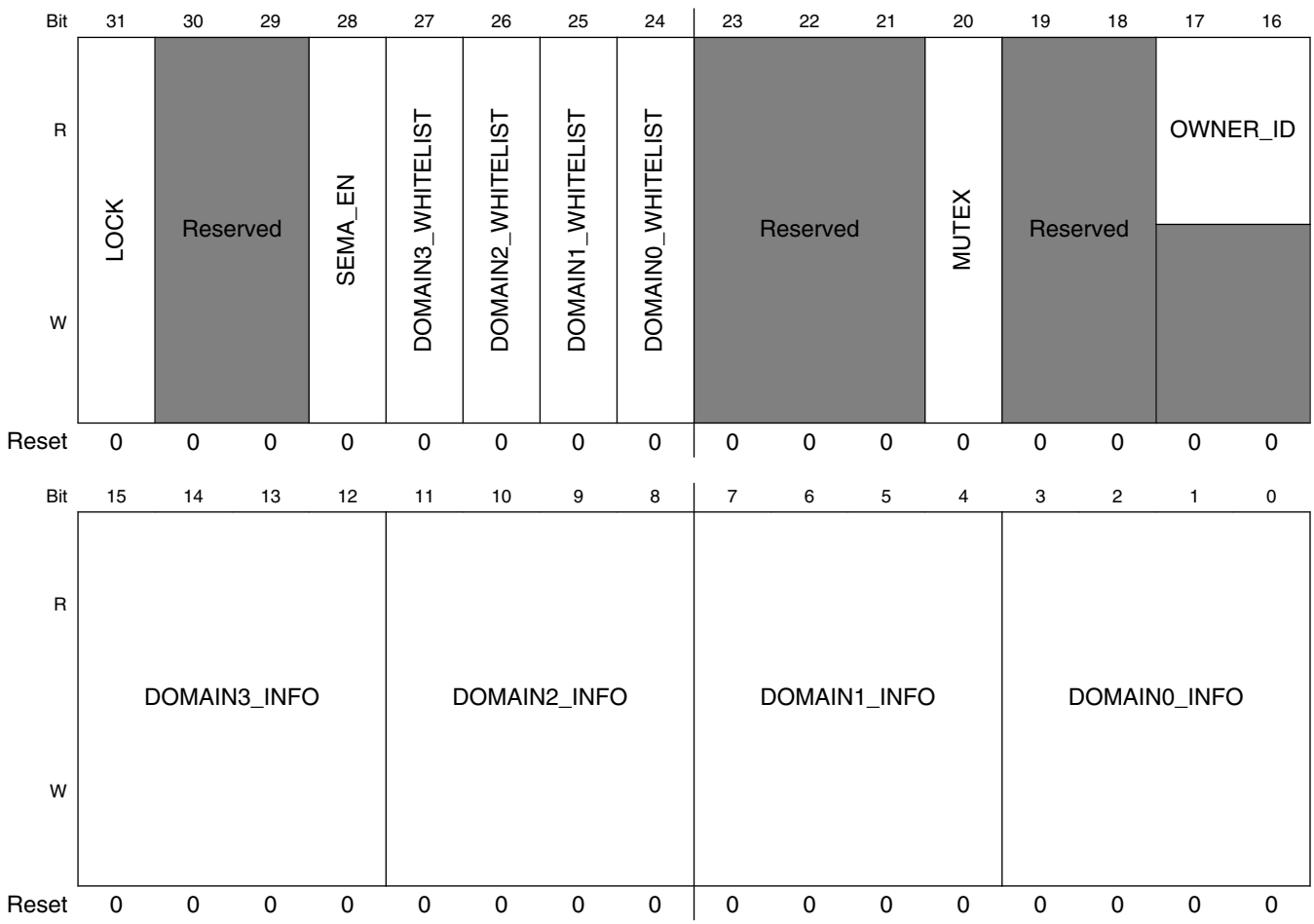
CCM_ACCESS_CTRL_ROOT*n*_CLR field descriptions (continued)

Field	Description
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.7.29 Access Control Register (CCM_ACCESS_CTRL_ROOT*n*_TOG)

Access Control Register

Address: 3038_0000h base + 807Ch offset + (128d × i), where i=0d to 141d



CCM_ACCESS_CTRL_ROOT*n*_TOG field descriptions

Field	Description
31 LOCK	Lock this clock root to use access control This bit can be set to 1 by software, and can be cleared only by system reset. 0 Access control inactive 1 Access control active
30–29 -	This field is reserved. Reserved
28 SEMA_EN	Enable internal semaphore This field cannot be changed when lock bit is 1 0 Disable 1 Enable
27 DOMAIN3_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
26 DOMAIN2_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
25 DOMAIN1_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
24 DOMAIN0_ WHITELIST	White list of domains that can change setting of this clock root. Each domain has a corresponding bit. 0 Domain cannot change the setting 1 Domain can change the setting
23–21 -	This field is reserved. Reserved
20 MUTEX	Semaphore to control access 0 Semaphore is free to take 1 Semaphore is taken Write 0 Release semaphore Write 1 Acquire semaphore
19–18 -	This field is reserved. Reserved
17–16 OWNER_ID	Current domain that owns semaphore This field is meaningless when MUTEX is 0 0 domaino 1 domain1

Table continues on the next page...

CCM_ACCESS_CTRL_ROOT*n*_TOG field descriptions (continued)

Field	Description
	2 domain2 3 domain3
15–12 DOMAIN3_INFO	Information from domain 3 to pass to others This field can only be changed by domain 3
11–8 DOMAIN2_INFO	Information from domain 2 to pass to others This field can only be changed by domain 2
7–4 DOMAIN1_INFO	Information from domain 1 to pass to others This field can only be changed by domain 1
DOMAIN0_INFO	Information from domain 0 to pass to others This field can only be changed by domain 0

5.1.8 CCM Analog Memory Map/Register Definition

CCM_ANALOG memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3036_0000	AUDIO PLL1 Configuration 0 Register (CCM_ANALOG_AUDIO_PLL1_CFG0)	32	R/W	0020_C081h	5.1.8.1/ 565
3036_0004	AUDIO PLL1 Configuration 1 Register (CCM_ANALOG_AUDIO_PLL1_CFG1)	32	R/W	0000_0040h	5.1.8.2/ 568
3036_0008	AUDIO PLL2 Configuration 0 Register (CCM_ANALOG_AUDIO_PLL2_CFG0)	32	R/W	0020_C081h	5.1.8.3/ 569
3036_000C	AUDIO PLL2 Configuration 1 Register (CCM_ANALOG_AUDIO_PLL2_CFG1)	32	R/W	0000_0040h	5.1.8.4/ 572
3036_0010	VIDEO PLL Configuration 0 Register (CCM_ANALOG_VIDEO_PLL1_CFG0)	32	R/W	0020_C081h	5.1.8.5/ 573
3036_0014	VIDEO PLL Configuration 1 Register (CCM_ANALOG_VIDEO_PLL1_CFG1)	32	R/W	0000_0040h	5.1.8.6/ 576
3036_0018	GPU PLL Configuration 0 Register (CCM_ANALOG_GPU_PLL_CFG0)	32	R/W	0020_C081h	5.1.8.7/ 577
3036_001C	GPU PLL Configuration 1 Register (CCM_ANALOG_GPU_PLL_CFG1)	32	R/W	0000_004Fh	5.1.8.8/ 580
3036_0020	VPU PLL Configuration 0 Register (CCM_ANALOG_VPU_PLL_CFG0)	32	R/W	0020_C081h	5.1.8.9/ 581
3036_0024	VPU PLL Configuration 1 Register (CCM_ANALOG_VPU_PLL_CFG1)	32	R/W	0000_003Bh	5.1.8.10/ 584
3036_0028	ARM PLL Configuration 0 Register (CCM_ANALOG_ARM_PLL_CFG0)	32	R/W	0020_C080h	5.1.8.11/ 585
3036_002C	ARM PLL Configuration 1 Register (CCM_ANALOG_ARM_PLL_CFG1)	32	R/W	0000_004Fh	5.1.8.12/ 588

Table continues on the next page...

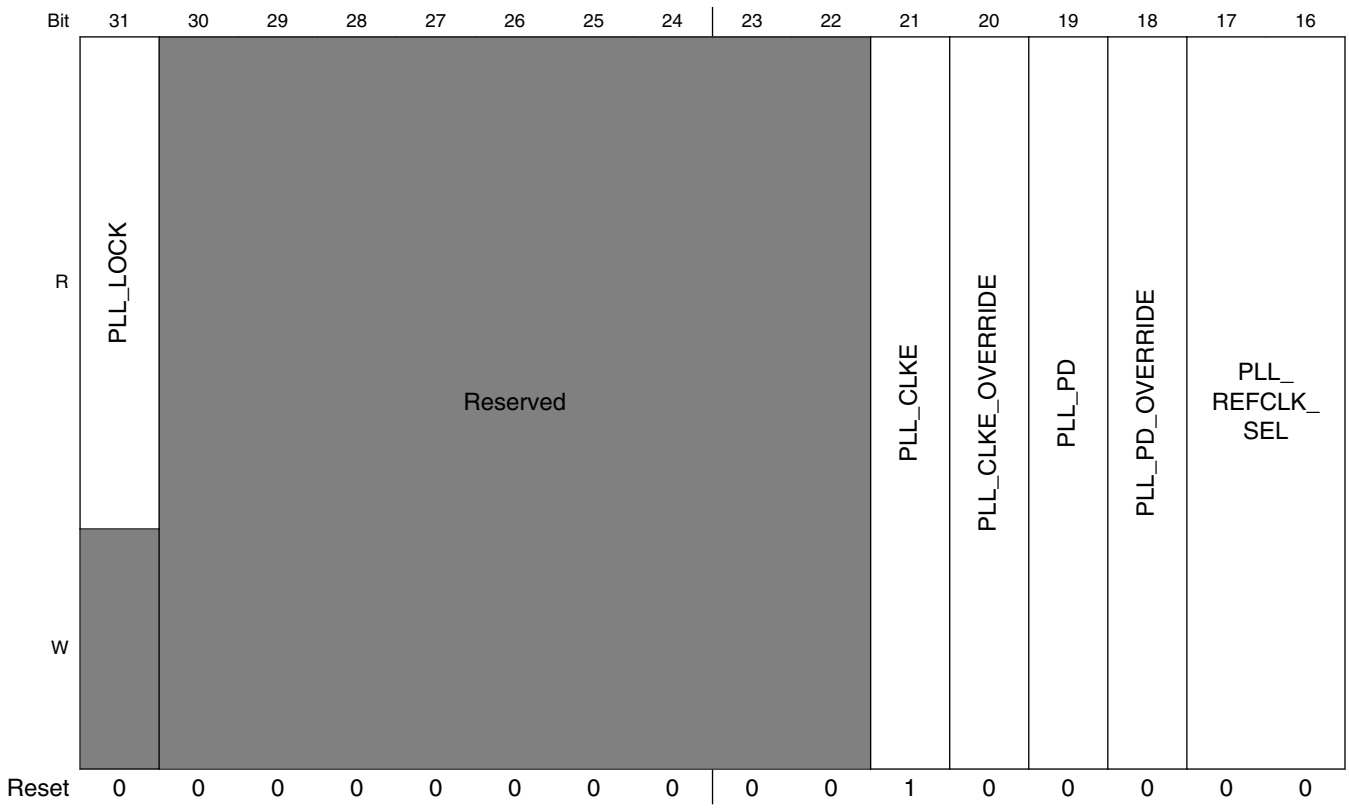
CCM_ANALOG memory map (continued)

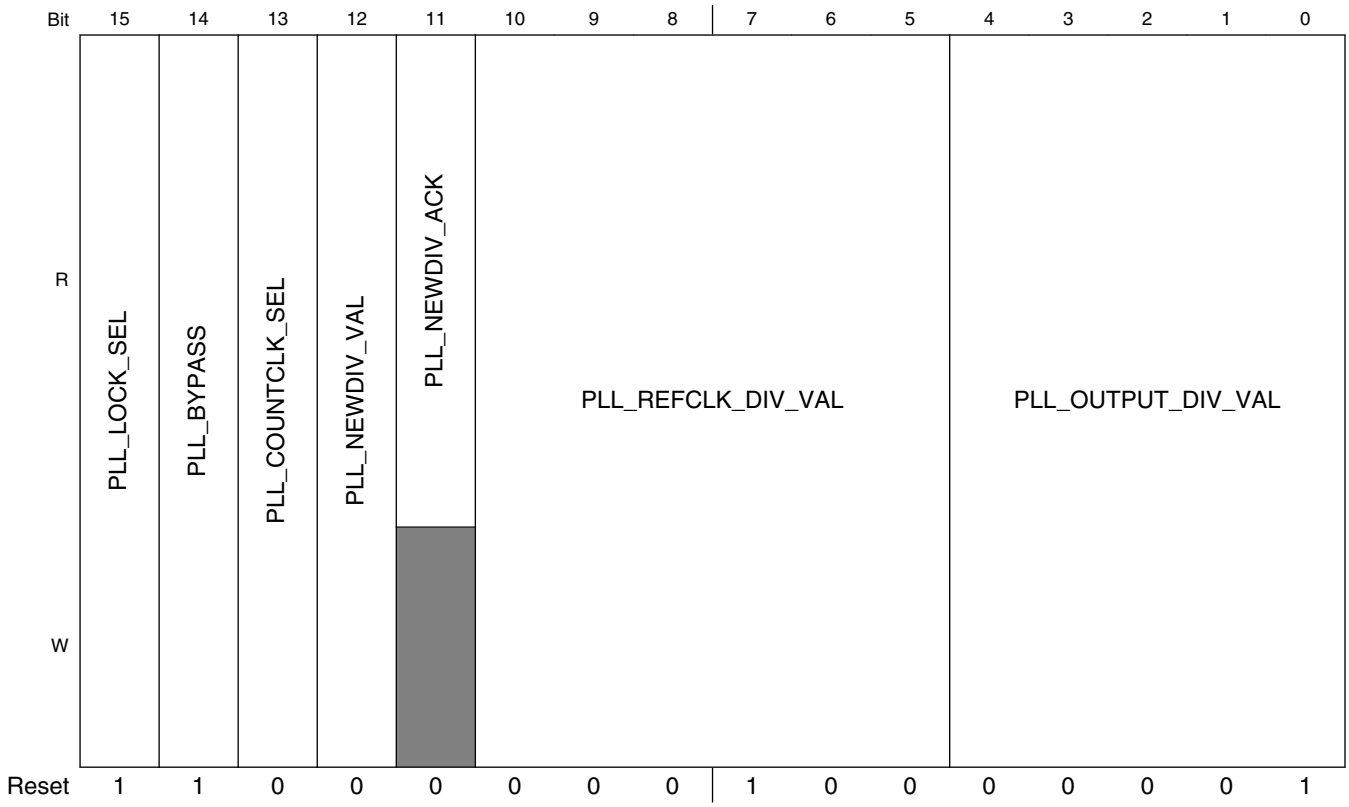
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3036_0030	System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL1_CFG0)	32	R/W	02AA_AA38h	5.1.8.13/ 589
3036_0034	System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL1_CFG1)	32	R/W	0000_0000h	5.1.8.14/ 592
3036_0038	System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL1_CFG2)	32	R/W	00BB_E580h	5.1.8.15/ 593
3036_003C	System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL2_CFG0)	32	R/W	02AA_AA38h	5.1.8.16/ 594
3036_0040	System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL2_CFG1)	32	R/W	0000_0000h	5.1.8.17/ 597
3036_0044	System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL2_CFG2)	32	R/W	00BB_E700h	5.1.8.18/ 598
3036_0048	System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL3_CFG0)	32	R/W	0000_0238h	5.1.8.19/ 599
3036_004C	System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL3_CFG1)	32	R/W	0000_0000h	5.1.8.20/ 601
3036_0050	System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL3_CFG2)	32	R/W	00BB_E700h	5.1.8.21/ 602
3036_0054	VIDEO PLL2 Configuration 0 Register (CCM_ANALOG_VIDEO_PLL2_CFG0)	32	R/W	0000_0238h	5.1.8.22/ 603
3036_0058	VIDEO PLL2 Configuration 1 Register (CCM_ANALOG_VIDEO_PLL2_CFG1)	32	R/W	0000_0000h	5.1.8.23/ 605
3036_005C	VIDEO PLL2 Configuration 2 Register (CCM_ANALOG_VIDEO_PLL2_CFG2)	32	R/W	00EC_6480h	5.1.8.24/ 606
3036_0060	DRAM PLL Configuration 0 Register (CCM_ANALOG_DRAM_PLL_CFG0)	32	R/W	0000_0238h	5.1.8.25/ 607
3036_0064	DRAM PLL Configuration 1 Register (CCM_ANALOG_DRAM_PLL_CFG1)	32	R/W	0000_0000h	5.1.8.26/ 609
3036_0068	DRAM PLL Configuration 2 Register (CCM_ANALOG_DRAM_PLL_CFG2)	32	R/W	00BB_E580h	5.1.8.27/ 610
3036_006C	DIGPROG Register (CCM_ANALOG_DIGPROG)	32	R	0082_4010h	5.1.8.28/ 611
3036_0070	Osc Misc Configuration Register (CCM_ANALOG_OSC_MISC_CFG)	32	R/W	0000_0014h	5.1.8.29/ 612
3036_0074	PLLOUT Monitor Configuration Register (CCM_ANALOG_PLLOUT_MONITOR_CFG)	32	R/W	0000_0000h	5.1.8.30/ 613
3036_0078	Fractional PLLOUT Divider Configuration Register (CCM_ANALOG_FRAC_PLLOUT_DIV_CFG)	32	R/W	0000_0000h	5.1.8.31/ 614
3036_007C	SCCG PLLOUT Divider Configuration Register (CCM_ANALOG_SCCG_PLLOUT_DIV_CFG)	32	R/W	0000_0000h	5.1.8.32/ 617

5.1.8.1 AUDIO PLL1 Configuration 0 Register (CCM_ANALOG_AUDIO_PLL1_CFG0)

AUDIO PLL1 Configuration 0 Register

Address: 3036_0000h base + 0h offset = 3036_0000h





CCM_ANALOG_AUDIO_PLL1_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

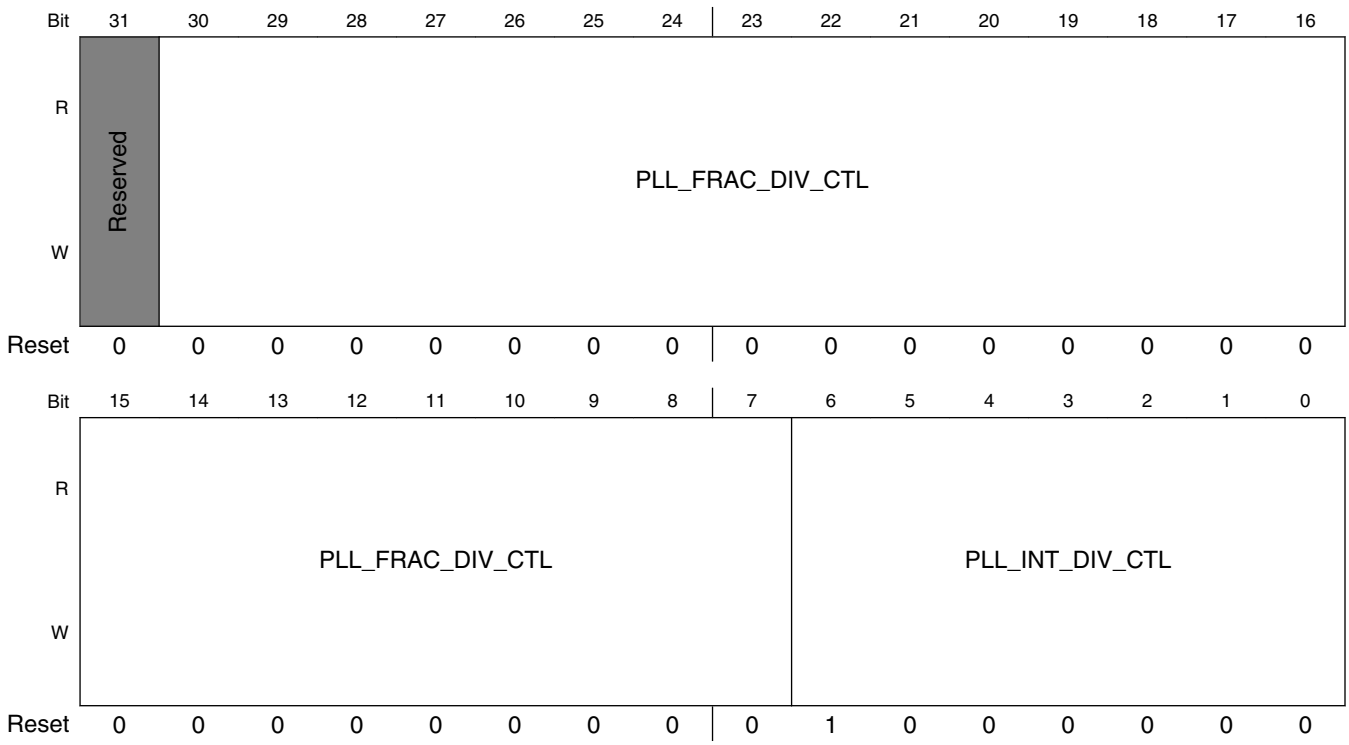
CCM_ANALOG_AUDIO_PLL1_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.2 AUDIO PLL1 Configuration 1 Register (CCM_ANALOG_AUDIO_PLL1_CFG1)

AUDIO PLL1 Configuration 1 Register

Address: 3036_0000h base + 4h offset = 3036_0004h



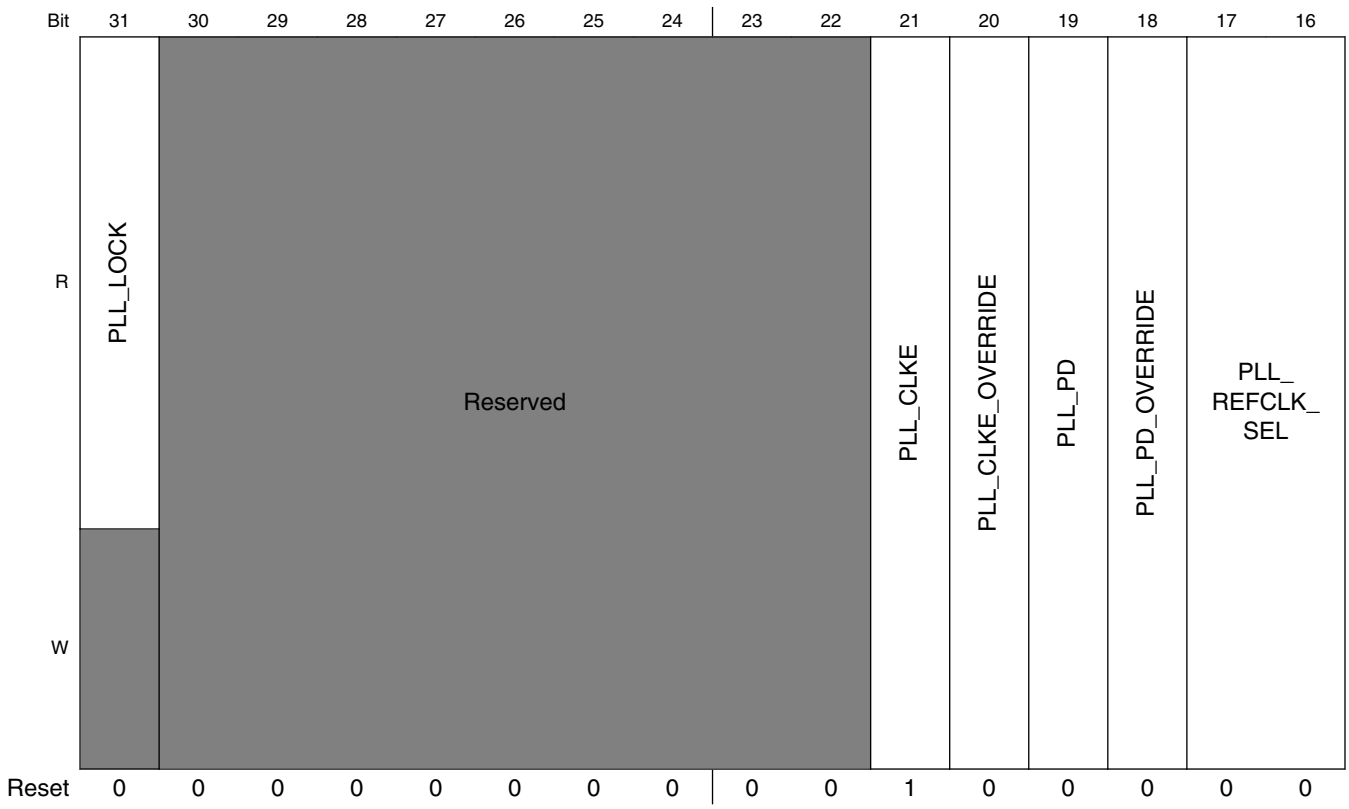
CCM_ANALOG_AUDIO_PLL1_CFG1 field descriptions

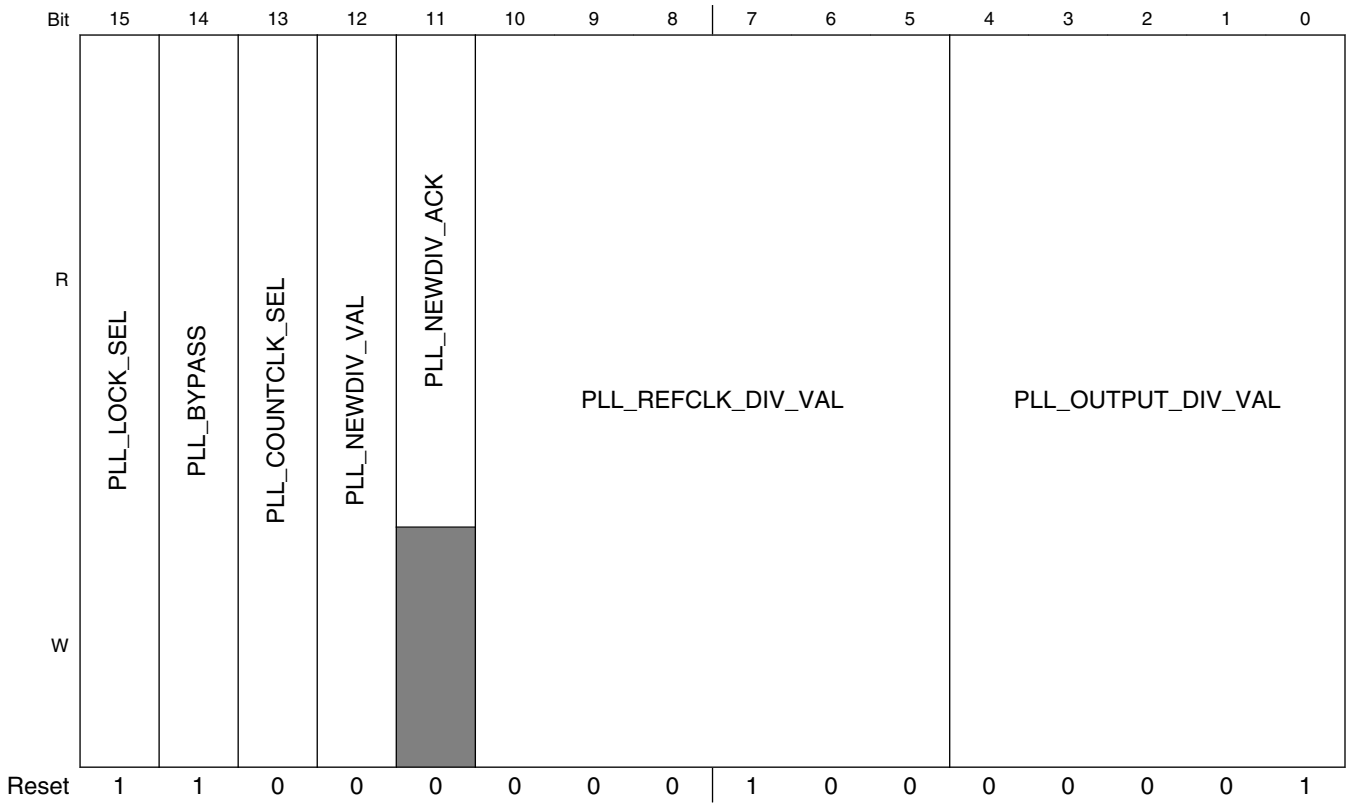
Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.3 AUDIO PLL2 Configuration 0 Register (CCM_ANALOG_AUDIO_PLL2_CFG0)

AUDIO PLL2 Configuration 0 Register

Address: 3036_0000h base + 8h offset = 3036_0008h





CCM_ANALOG_AUDIO_PLL2_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

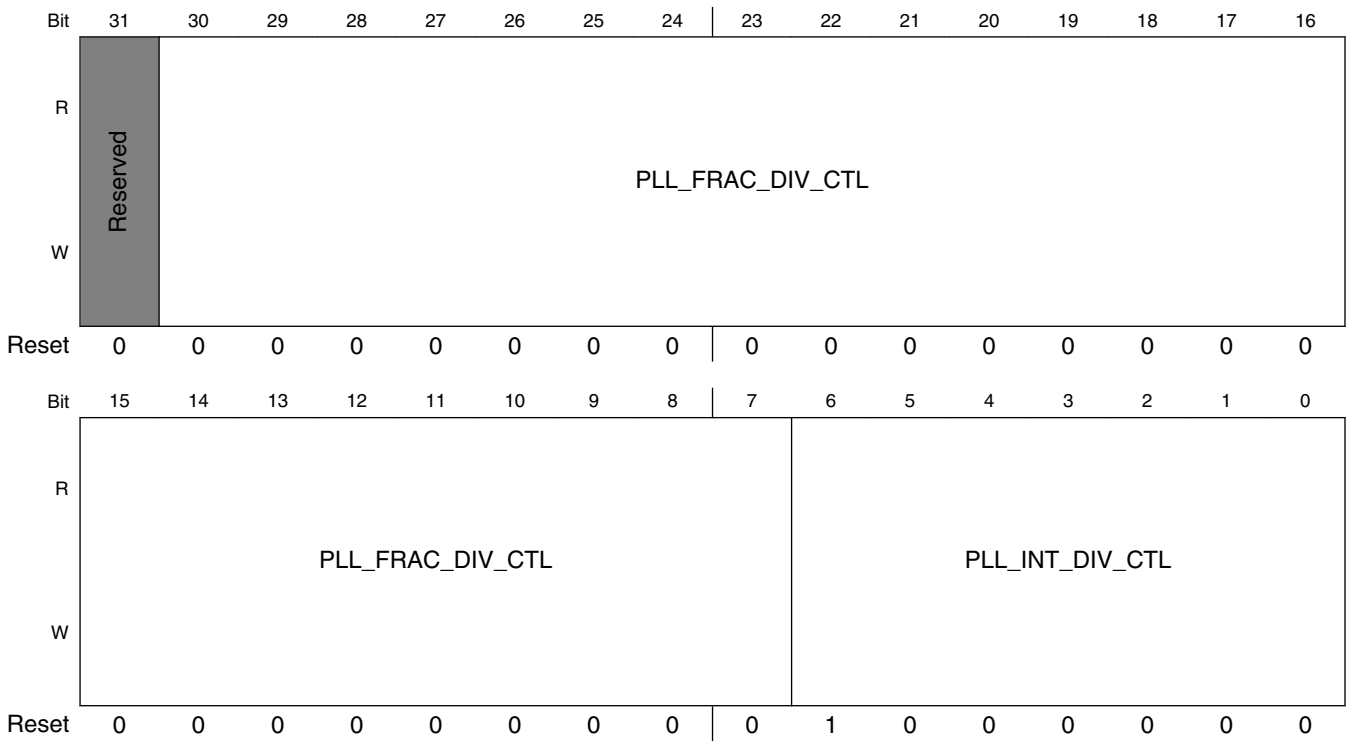
CCM_ANALOG_AUDIO_PLL2_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.4 AUDIO PLL2 Configuration 1 Register (CCM_ANALOG_AUDIO_PLL2_CFG1)

AUDIO PLL2 Configuration 1 Register

Address: 3036_0000h base + Ch offset = 3036_000Ch



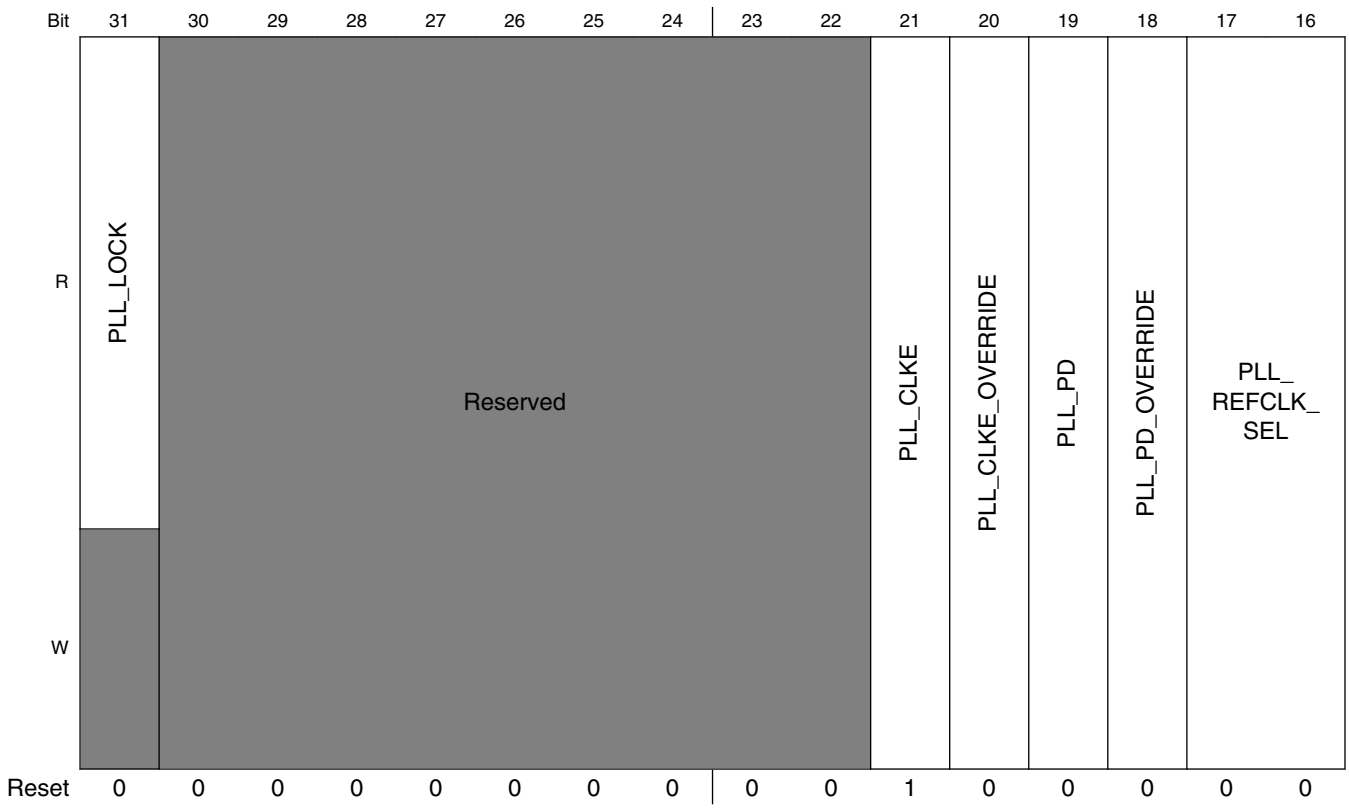
CCM_ANALOG_AUDIO_PLL2_CFG1 field descriptions

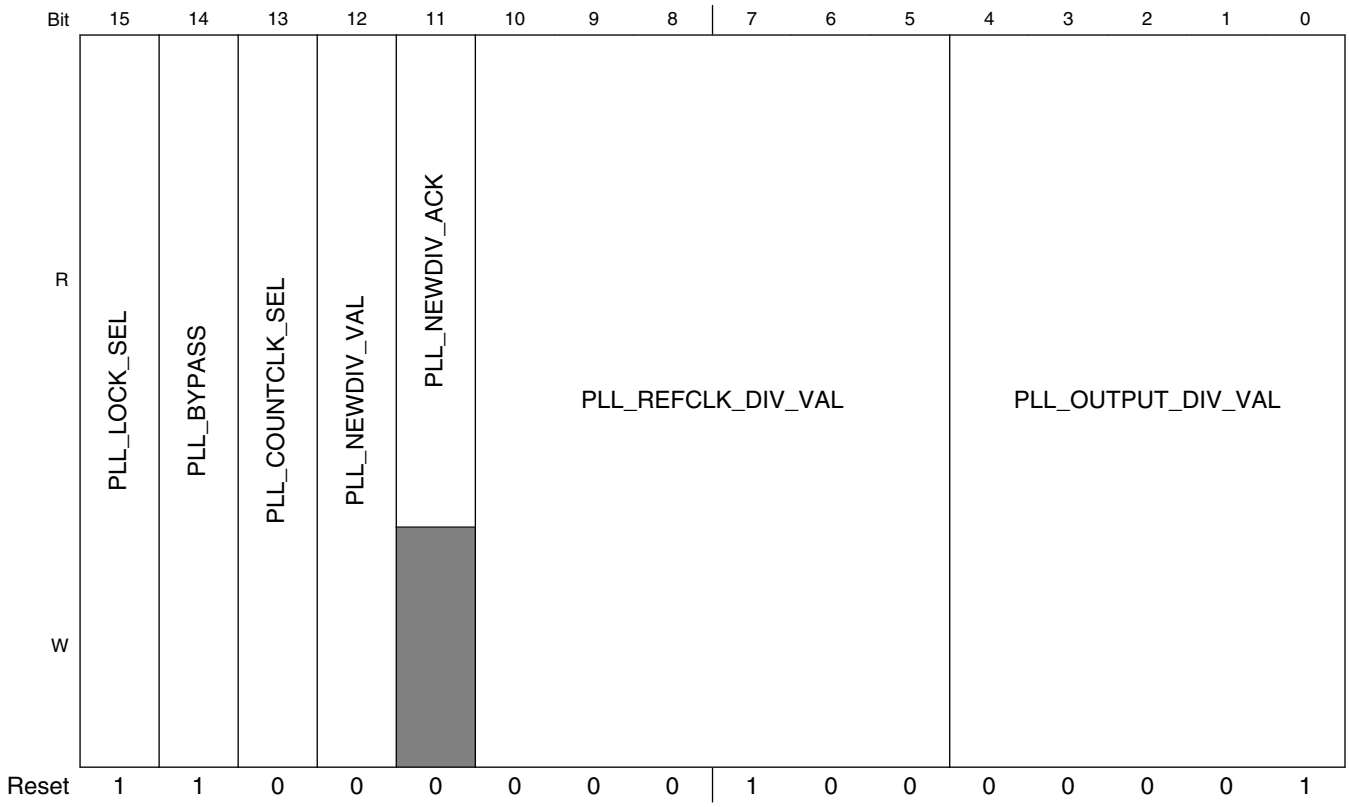
Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.5 VIDEO PLL Configuration 0 Register (CCM_ANALOG_VIDEO_PLL1_CFG0)

VIDEO PLL Configuration 0 Register

Address: 3036_0000h base + 10h offset = 3036_0010h





CCM_ANALOG_VIDEO_PLL1_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

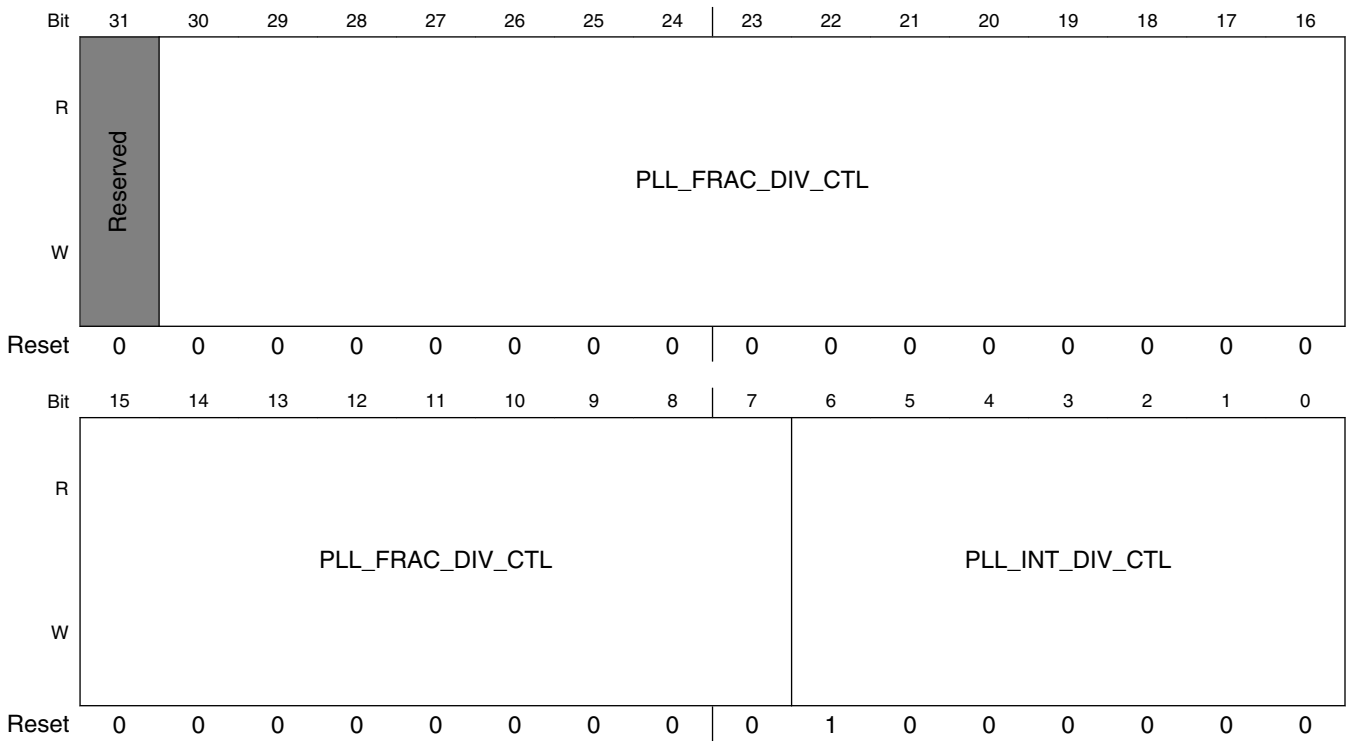
CCM_ANALOG_VIDEO_PLL1_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.6 VIDEO PLL Configuration 1 Register (CCM_ANALOG_VIDEO_PLL1_CFG1)

VIDEO PLL Configuration 1 Register

Address: 3036_0000h base + 14h offset = 3036_0014h



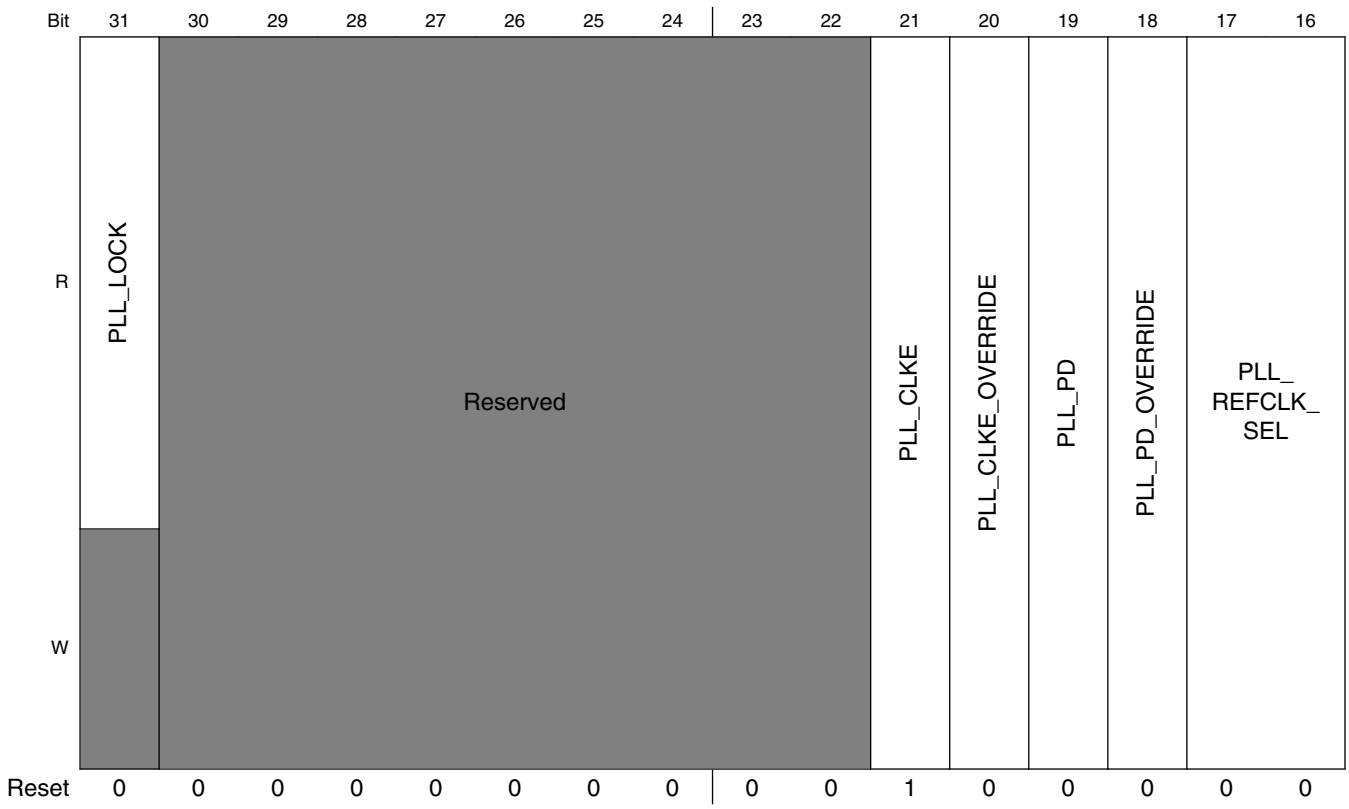
CCM_ANALOG_VIDEO_PLL1_CFG1 field descriptions

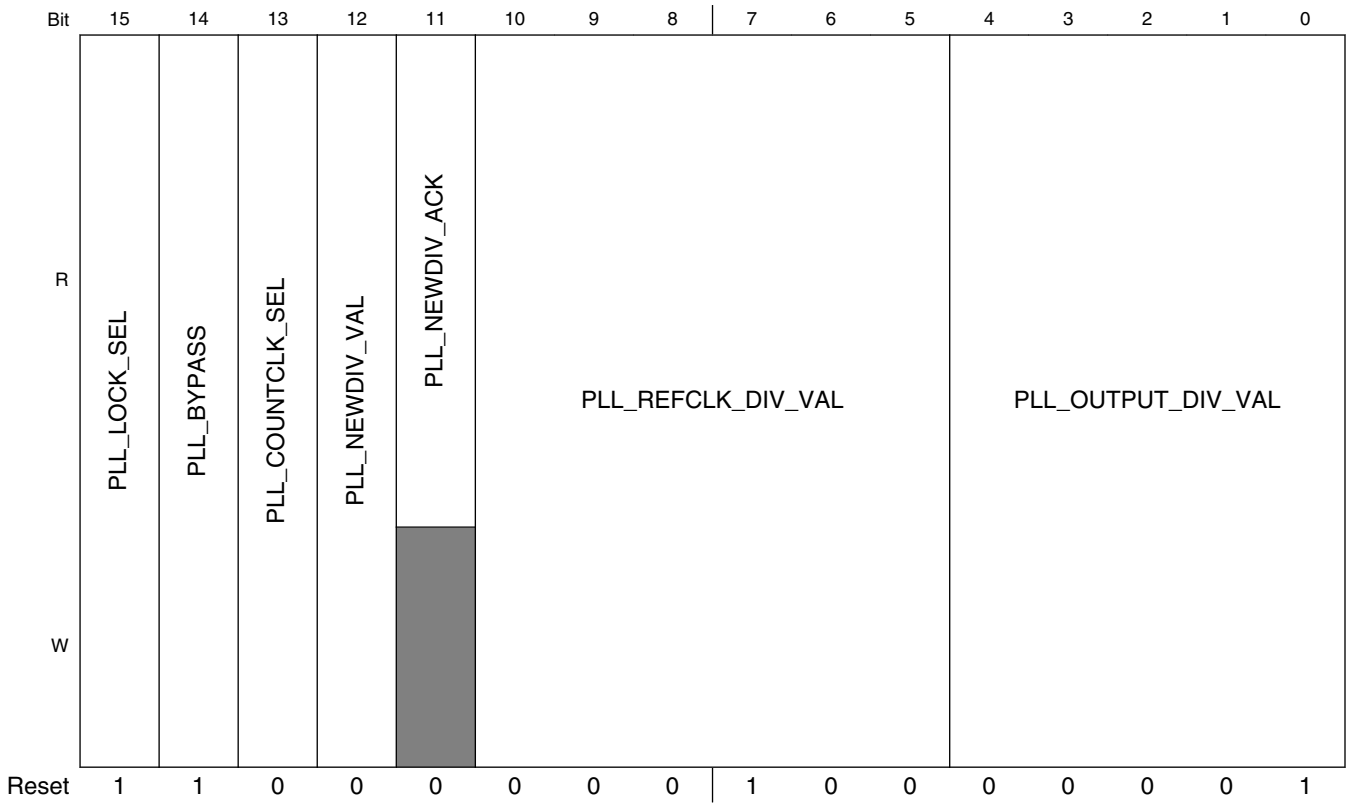
Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.7 GPU PLL Configuration 0 Register (CCM_ANALOG_GPU_PLL_CFG0)

GPU PLL Configuration 0 Register

Address: 3036_0000h base + 18h offset = 3036_0018h





CCM_ANALOG_GPU_PLL_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

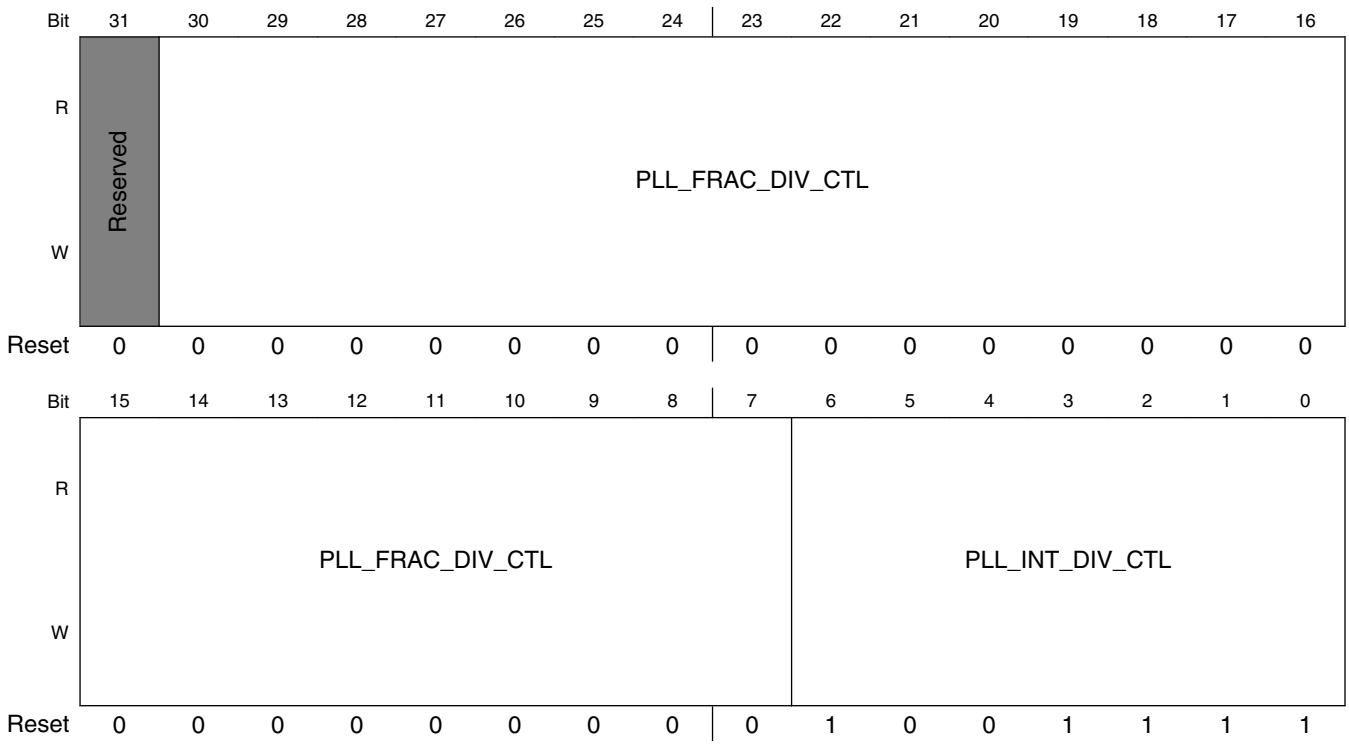
CCM_ANALOG_GPU_PLL_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.8 GPU PLL Configuration 1 Register (CCM_ANALOG_GPU_PLL_CFG1)

GPU PLL Configuration 1 Register

Address: 3036_0000h base + 1Ch offset = 3036_001Ch



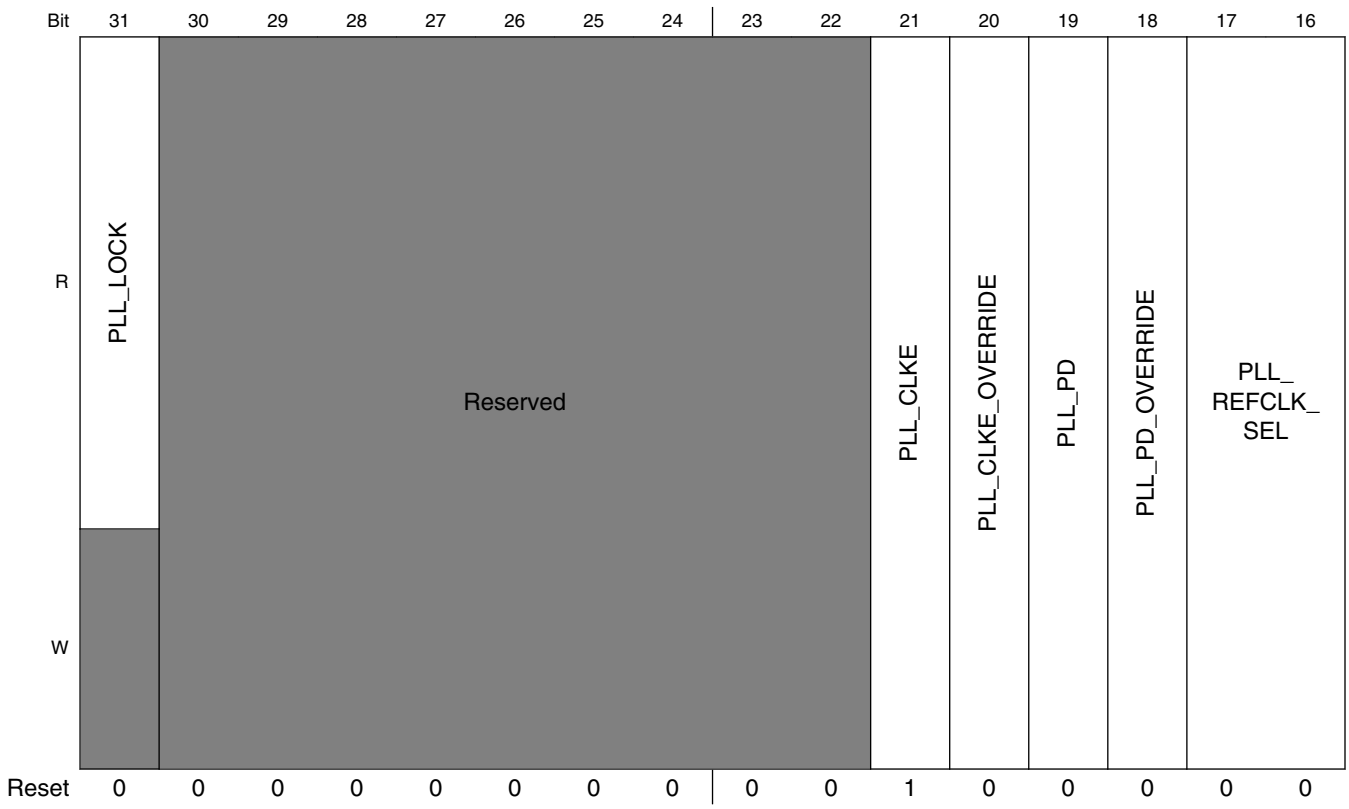
CCM_ANALOG_GPU_PLL_CFG1 field descriptions

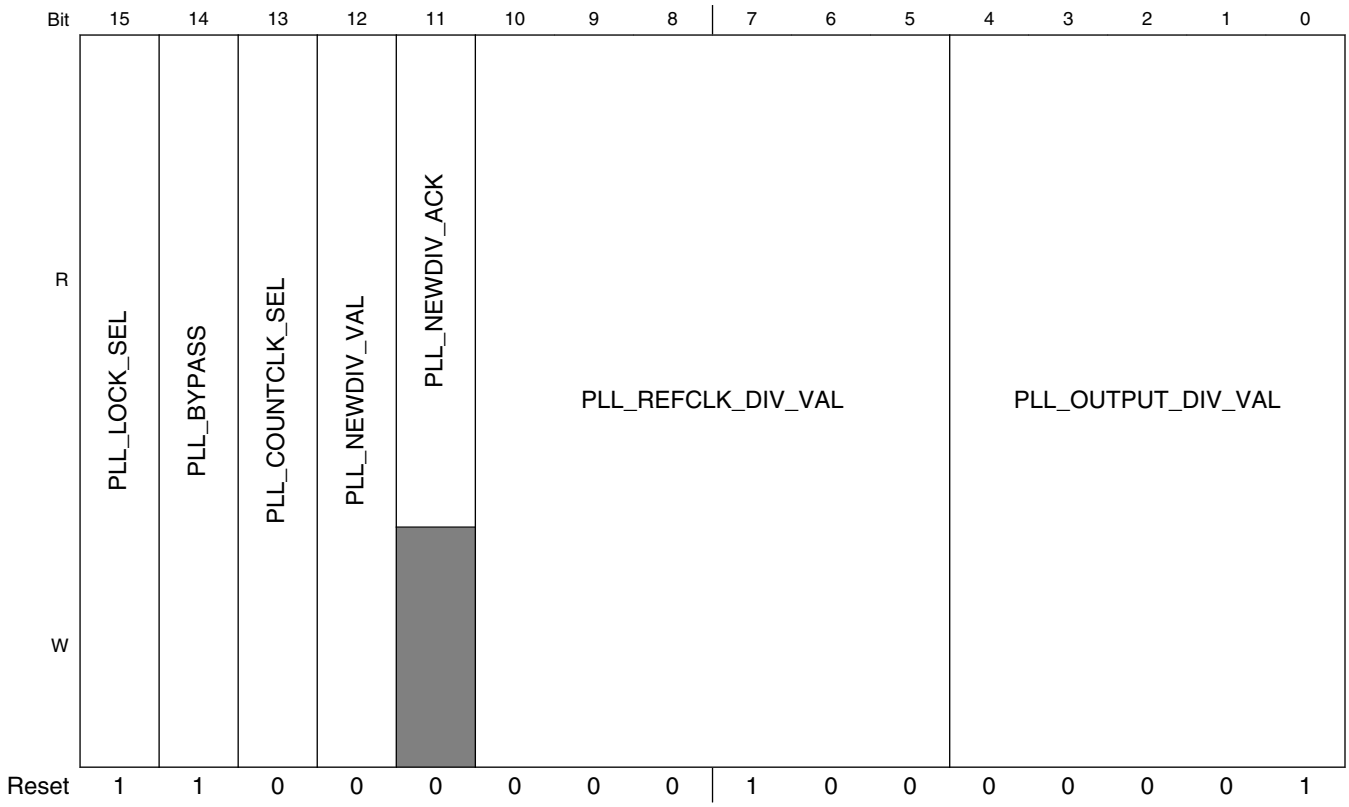
Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.9 VPU PLL Configuration 0 Register (CCM_ANALOG_VPU_PLL_CFG0)

VPU PLL Configuration 0 Register

Address: 3036_0000h base + 20h offset = 3036_0020h





CCM_ANALOG_VPU_PLL_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

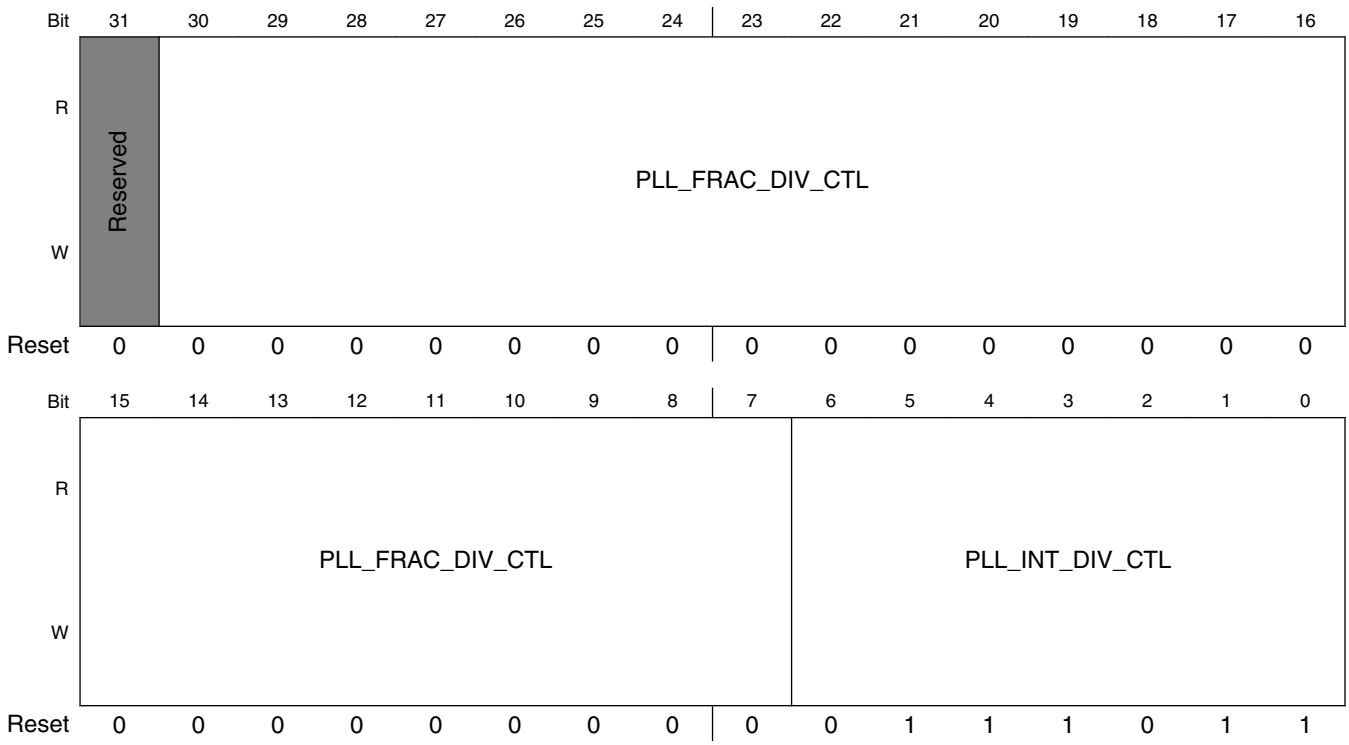
CCM_ANALOG_VPU_PLL_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.10 VPU PLL Configuration 1 Register (CCM_ANALOG_VPU_PLL_CFG1)

VPU PLL Configuration 1 Register

Address: 3036_0000h base + 24h offset = 3036_0024h



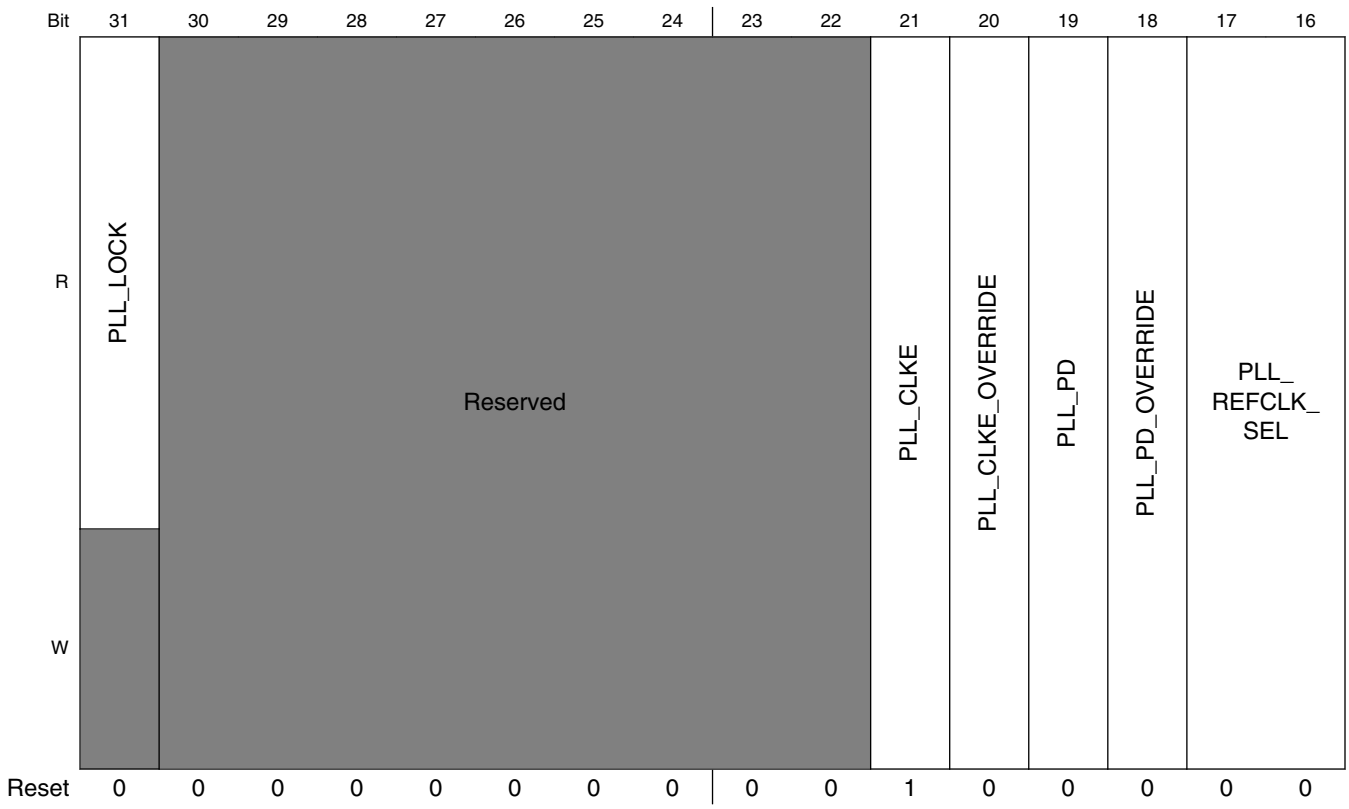
CCM_ANALOG_VPU_PLL_CFG1 field descriptions

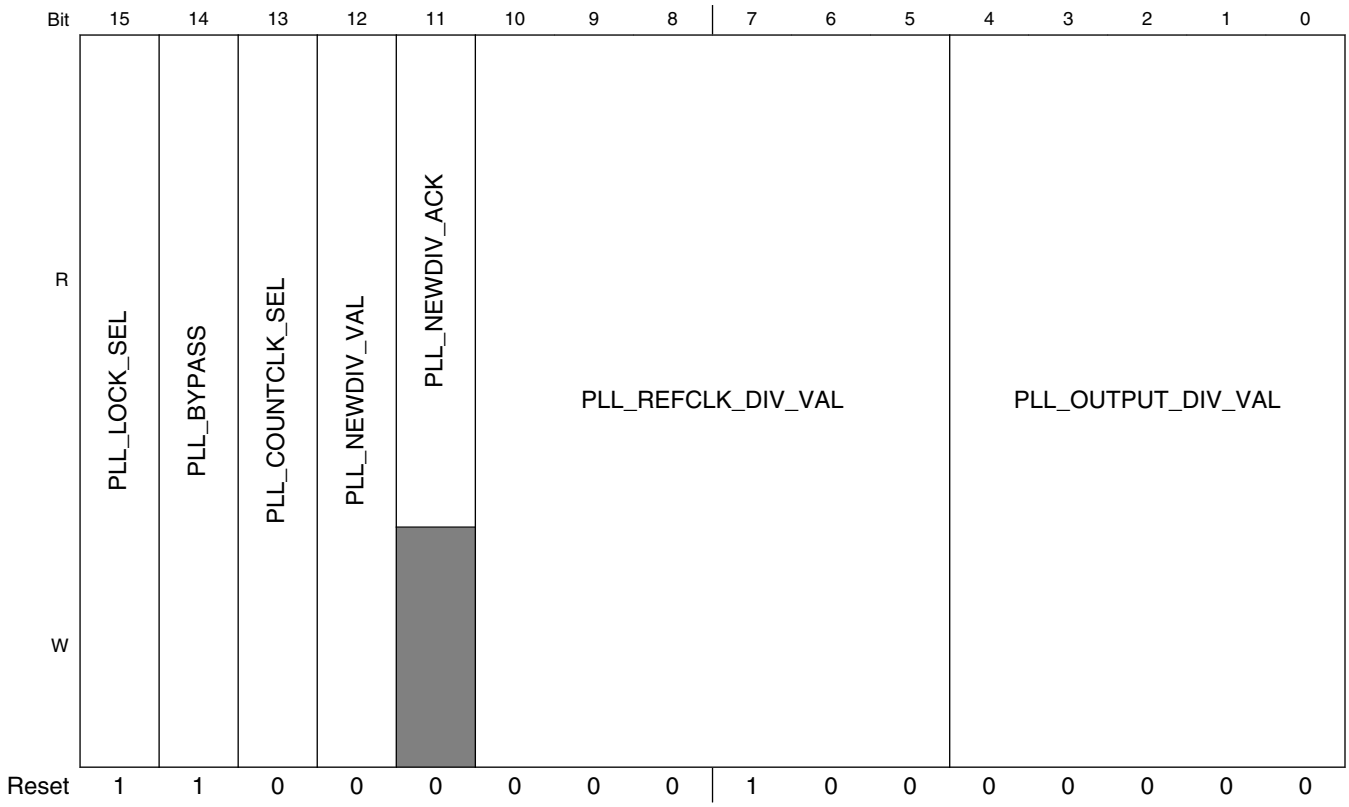
Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.11 ARM PLL Configuration 0 Register (CCM_ANALOG_ARM_PLL_CFG0)

ARM PLL Configuration 0 Register

Address: 3036_0000h base + 28h offset = 3036_0028h





CCM_ANALOG_ARM_PLL_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–22 -	This field is reserved. Reserved
21 PLL_CLKE	PLL output clock clock gating enable active high
20 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
19 PLL_PD	PLL output clock clock gating enable active high
18 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
17–16 PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N
15 PLL_LOCK_SEL	PLL Lock signal select

Table continues on the next page...

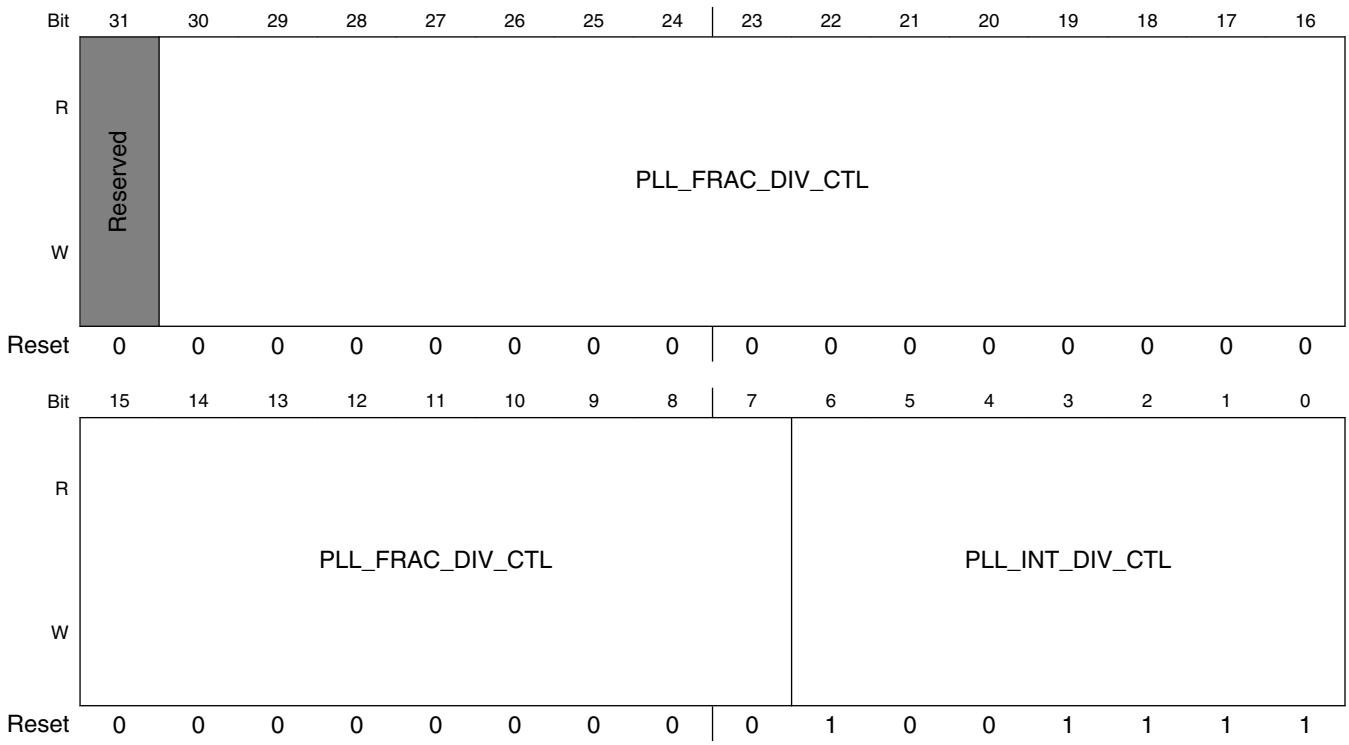
CCM_ANALOG_ARM_PLL_CFG0 field descriptions (continued)

Field	Description
	00 Select PLL lock output 01 Select maximum lock time counter output
14 PLL_BYPASS	PLL bypass control active high
13 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
12 PLL_NEWDIV_ VAL	PLL new fraction divide input control active high
11 PLL_NEWDIV_ ACK	PLL new fraction divide handshake signal
10–5 PLL_REFCLK_ DIV_VAL	PLL reference clock divide value. Both REF and post_divide REF must be within the range 10MHz to 300MHz. Valid divider values are 1 to 64
PLL_OUTPUT_ DIV_VAL	PLL output clock divide value Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are even divides from 2 to 64 (binary value + 1) * 2 00000 - 2 00001 - 4 00010 - 6 . . . 11111 - 64

5.1.8.12 ARM PLL Configuration 1 Register (CCM_ANALOG_ARM_PLL_CFG1)

ARM PLL Configuration 1 Register

Address: 3036_0000h base + 2Ch offset = 3036_002Ch



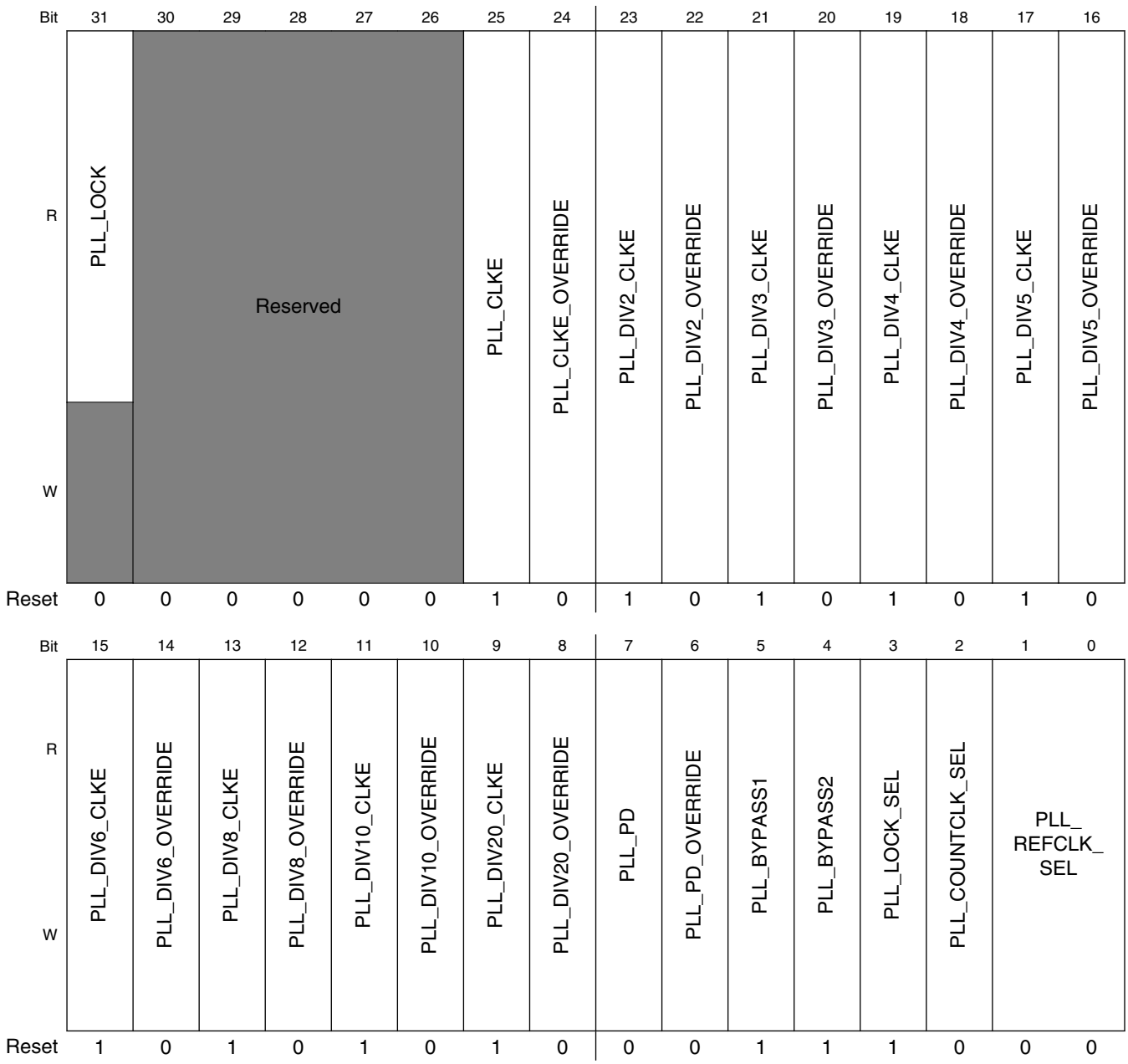
CCM_ANALOG_ARM_PLL_CFG1 field descriptions

Field	Description
31 -	This field is reserved. Reserved
30–7 PLL_FRAC_DIV_CTL	PLL fraction divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz Valid divider values are 1 to 2^24
PLL_INT_DIV_CTL	PLL Integer divide control Settings must maintain the PLL operational range. Valid output clock range 30MHz to 2000MHz. Valid divider values are 1 to 32

5.1.8.13 System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL1_CFG0)

System PLL Configuration 0 Register

Address: 3036_0000h base + 30h offset = 3036_0030h



CCM_ANALOG_SYS_PLL1_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–26 -	This field is reserved. Reserved
25 PLL_CLKE	PLL output clock clock gating enable active high
24 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
23 PLL_DIV2_CLKE	PLL output clock divide by 2 clock gating enable active high
22 PLL_DIV2_ OVERRIDE	Override the PLL_DIV2_CLKE, clock gating enable signal from CCM block active high
21 PLL_DIV3_CLKE	PLL output clock divide by 3 clock gating enable active high
20 PLL_DIV3_ OVERRIDE	Override the PLL_DIV3_CLKE, clock gating enable signal from CCM block active high
19 PLL_DIV4_CLKE	PLL output clock divide by 4 clock gating enable active high
18 PLL_DIV4_ OVERRIDE	Override the PLL_DIV4_CLKE, clock gating enable signal from CCM block active high
17 PLL_DIV5_CLKE	PLL output clock divide by 5 clock gating enable active high
16 PLL_DIV5_ OVERRIDE	Override the PLL_DIV5_CLKE, clock gating enable signal from CCM block active high
15 PLL_DIV6_CLKE	PLL output clock divide by 6 clock gating enable active high
14 PLL_DIV6_ OVERRIDE	Override the PLL_DIV6_CLKE, clock gating enable signal from CCM block active high
13 PLL_DIV8_CLKE	PLL output clock divide by 8 clock gating enable active high
12 PLL_DIV8_ OVERRIDE	Override the PLL_DIV8_CLKE, clock gating enable signal from CCM block active high
11 PLL_DIV10_ CLKE	PLL output clock divide by 10 clock gating enable active high
10 PLL_DIV10_ OVERRIDE	Override the PLL_DIV10_CLKE, clock gating enable signal from CCM block active high
9 PLL_DIV20_ CLKE	PLL output clock divide by 20 clock gating enable active high

Table continues on the next page...

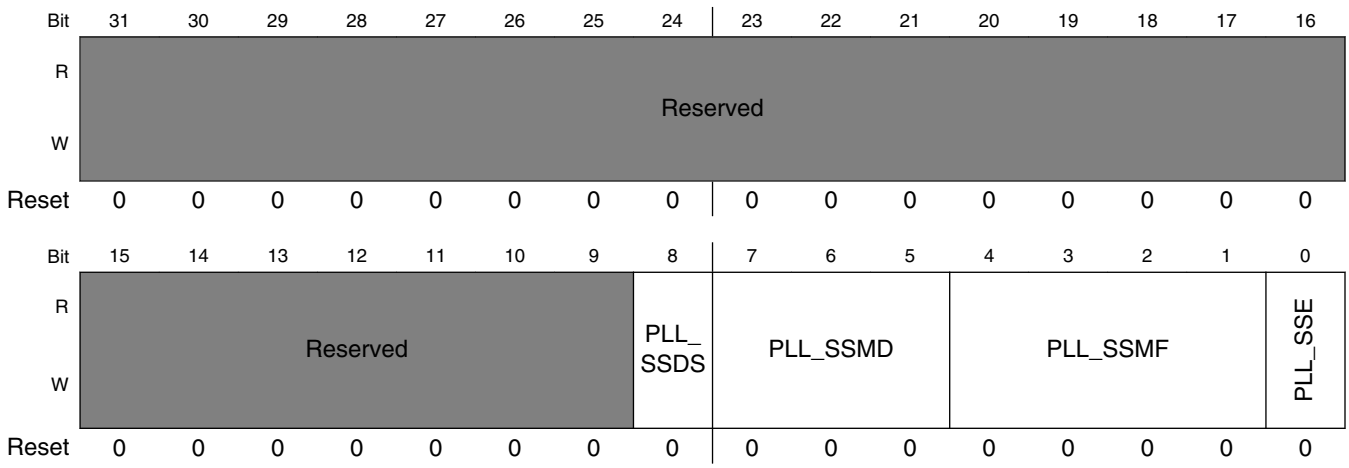
CCM_ANALOG_SYS_PLL1_CFG0 field descriptions (continued)

Field	Description
8 PLL_DIV20_ OVERRIDE	Override the PLL_DIV20_CLKE, clock gating enable signal from CCM block active high
7 PLL_PD	PLL output clock clock gating enable active high
6 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
5 PLL_BYPASS1	Internal PLL1 bypass control active high
4 PLL_BYPASS2	Internal PLL2 bypass control active high
3 PLL_LOCK_SEL	PLL Lock signal select 00 Select PLL lock output 01 Select maximum lock time counter output
2 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N

5.1.8.14 System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL1_CFG1)

System PLL Configuration 1 Register

Address: 3036_0000h base + 34h offset = 3036_0034h



CCM_ANALOG_SYS_PLL1_CFG1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PLL_SSDS	Selects between Spread Spectrum Center Spread and Down Spread Modes. 0 Center Spread 1 Down Spread
7–5 PLL_SSMD	Controls Spread Spectrum modulation depth 000 0.25 001 0.5 010 0.75 011 1.0 100 1.5 101 2.0 110 3.0 111 4.0
4–1 PLL_SSMF	Controls Spread Spectrum Modulation Frequency Fmod = SSMF[3:0]*2*Fvco1/251658
0 PLL_SSE	Enables Spread Spectrum Mode active high

5.1.8.15 System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL1_CFG2)

System PLL Configuration 2 Register

Address: 3036_0000h base + 38h offset = 3036_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				PLL_REF_DIVR1			PLL_REF_DIVR2						PLL_FEEDBACK_DIVF1		
Reset	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_FEEDBACK_DIVF1			PLL_FEEDBACK_DIVF2					PLL_OUTPUT_DIV_VAL							PLL_FILTER_RANGE
Reset	1	1	1	0	0	1	0	1	1	0	0	0	0	0	0	0

CCM_ANALOG_SYS_PLL1_CFG2 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–25 PLL_REF_DIVR1	Internal PLL1 reference clock divider value. REF must be within the range 25MHz to 235MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 8
24–19 PLL_REF_DIVR2	Internal PLL2 reference clock divider value. REF must be within the range 54MHz to 75MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 64
18–13 PLL_ FEEDBACK_ DIVF1	Internal PLL1 reference clock divider value. Fvco1 must be within the range 1600MHz to 2400MHz. Valid divider values are 1 to 64
12–7 PLL_ FEEDBACK_ DIVF2	Internal PLL2 reference clock divider value. REF must be within the range 20MHz to 1200MHz. Valid divider values are 1 to 64
6–1 PLL_OUTPUT_ DIV_VAL	Internal PLL2 output clock divider value Fout must be within the range 20MHz ~ 1200MHz. Valid divider values are 1 ~ 64
0 PLL_FILTER_ RANGE	This sets the internal PLL1 loop filter to work with the post-reference divider frequency. 0 25 to 35 MHz 1 35 to 54 MHz

5.1.8.16 System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL2_CFG0)

System PLL Configuration 0 Register

Address: 3036_0000h base + 3Ch offset = 3036_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PLL_LOCK	Reserved					PLL_CLKE	PLL_CLKE_OVERRIDE	PLL_DIV2_CLKE	PLL_DIV2_OVERRIDE	PLL_DIV3_CLKE	PLL_DIV3_OVERRIDE	PLL_DIV4_CLKE	PLL_DIV4_OVERRIDE	PLL_DIV5_CLKE	PLL_DIV5_OVERRIDE
W																
Reset	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PLL_DIV6_CLKE	PLL_DIV6_OVERRIDE	PLL_DIV8_CLKE	PLL_DIV8_OVERRIDE	PLL_DIV10_CLKE	PLL_DIV10_OVERRIDE	PLL_DIV20_CLKE	PLL_DIV20_OVERRIDE	PLL_PD	PLL_PD_OVERRIDE	PLL_BYPASS1	PLL_BYPASS2	PLL_LOCK_SEL	PLL_COUNTCLK_SEL	PLL_REFCLK_SEL	
W																
Reset	1	0	1	0	1	0	1	0	0	0	1	1	1	0	0	0

CCM_ANALOG_SYS_PLL2_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high
30–26 -	This field is reserved. Reserved
25 PLL_CLKE	PLL output clock clock gating enable active high
24 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
23 PLL_DIV2_CLKE	PLL output clock divide by 2 clock gating enable active high
22 PLL_DIV2_ OVERRIDE	Override the PLL_DIV2_CLKE, clock gating enable signal from CCM block active high
21 PLL_DIV3_CLKE	PLL output clock divide by 3 clock gating enable active high
20 PLL_DIV3_ OVERRIDE	Override the PLL_DIV3_CLKE, clock gating enable signal from CCM block active high
19 PLL_DIV4_CLKE	PLL output clock divide by 4 clock gating enable active high
18 PLL_DIV4_ OVERRIDE	Override the PLL_DIV4_CLKE, clock gating enable signal from CCM block active high
17 PLL_DIV5_CLKE	PLL output clock divide by 5 clock gating enable active high
16 PLL_DIV5_ OVERRIDE	Override the PLL_DIV5_CLKE, clock gating enable signal from CCM block active high
15 PLL_DIV6_CLKE	PLL output clock divide by 6 clock gating enable active high
14 PLL_DIV6_ OVERRIDE	Override the PLL_DIV6_CLKE, clock gating enable signal from CCM block active high
13 PLL_DIV8_CLKE	PLL output clock divide by 8 clock gating enable active high
12 PLL_DIV8_ OVERRIDE	Override the PLL_DIV8_CLKE, clock gating enable signal from CCM block active high
11 PLL_DIV10_ CLKE	PLL output clock divide by 10 clock gating enable active high
10 PLL_DIV10_ OVERRIDE	Override the PLL_DIV10_CLKE, clock gating enable signal from CCM block active high
9 PLL_DIV20_ CLKE	PLL output clock divide by 20 clock gating enable active high

Table continues on the next page...

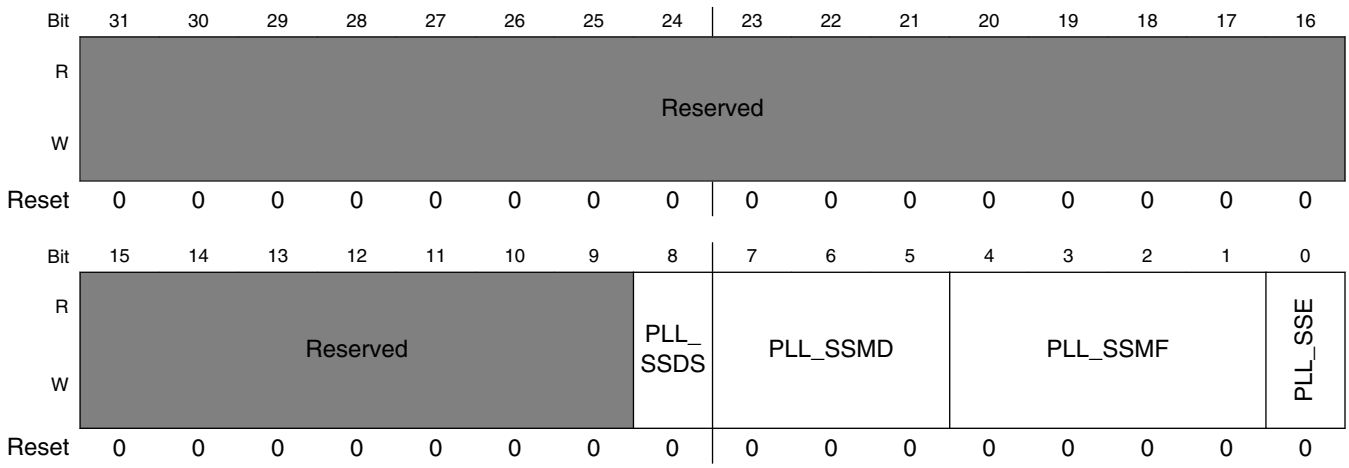
CCM_ANALOG_SYS_PLL2_CFG0 field descriptions (continued)

Field	Description
8 PLL_DIV20_ OVERRIDE	Override the PLL_DIV20_CLKE, clock gating enable signal from CCM block active high
7 PLL_PD	PLL output clock clock gating enable active high
6 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
5 PLL_BYPASS1	Internal PLL1 bypass control active high
4 PLL_BYPASS2	Internal PLL2 bypass control active high
3 PLL_LOCK_SEL	PLL Lock signal select 00 Select PLL lock output 01 Select maximum lock time counter output
2 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N

5.1.8.17 System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL2_CFG1)

System PLL Configuration 1 Register

Address: 3036_0000h base + 40h offset = 3036_0040h



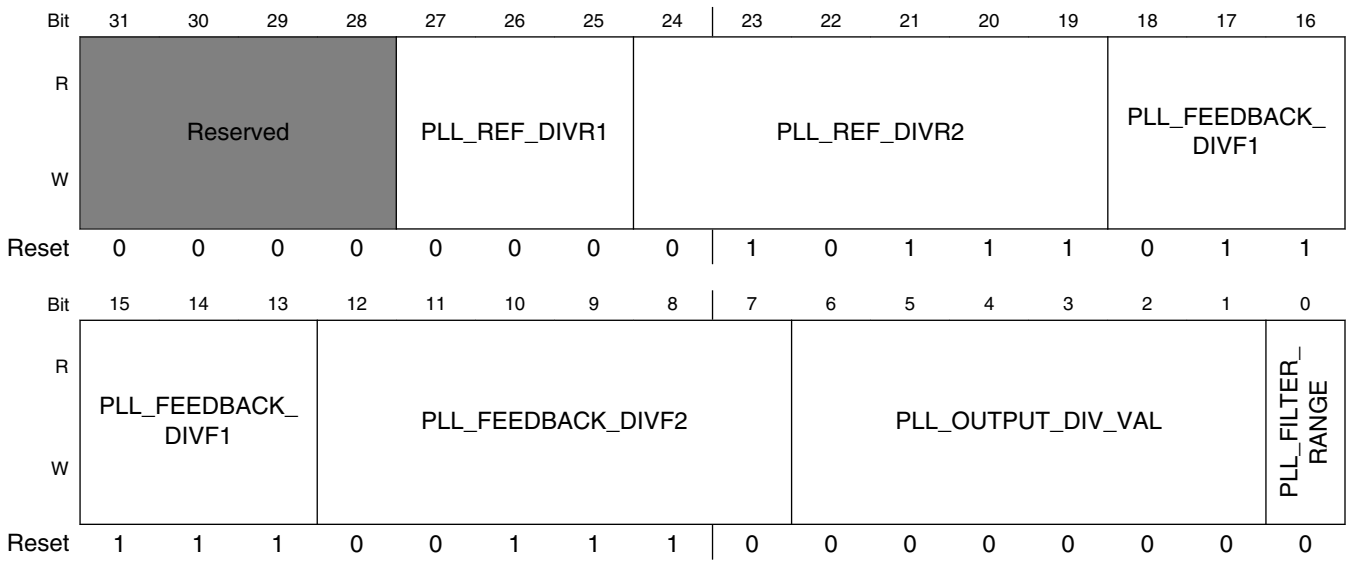
CCM_ANALOG_SYS_PLL2_CFG1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PLL_SSDS	Selects between Spread Spectrum Center Spread and Down Spread Modes. 0 Center Spread 1 Down Spread
7–5 PLL_SSMD	Controls Spread Spectrum modulation depth 000 0.25 001 0.5 010 0.75 011 1.0 100 1.5 101 2.0 110 3.0 111 4.0
4–1 PLL_SSMF	Controls Spread Spectrum Modulation Frequency Fmod = SSMF[3:0]*2*Fvco1/251658
0 PLL_SSE	Enables Spread Spectrum Mode active high

5.1.8.18 System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL2_CFG2)

System PLL Configuration 2 Register

Address: 3036_0000h base + 44h offset = 3036_0044h



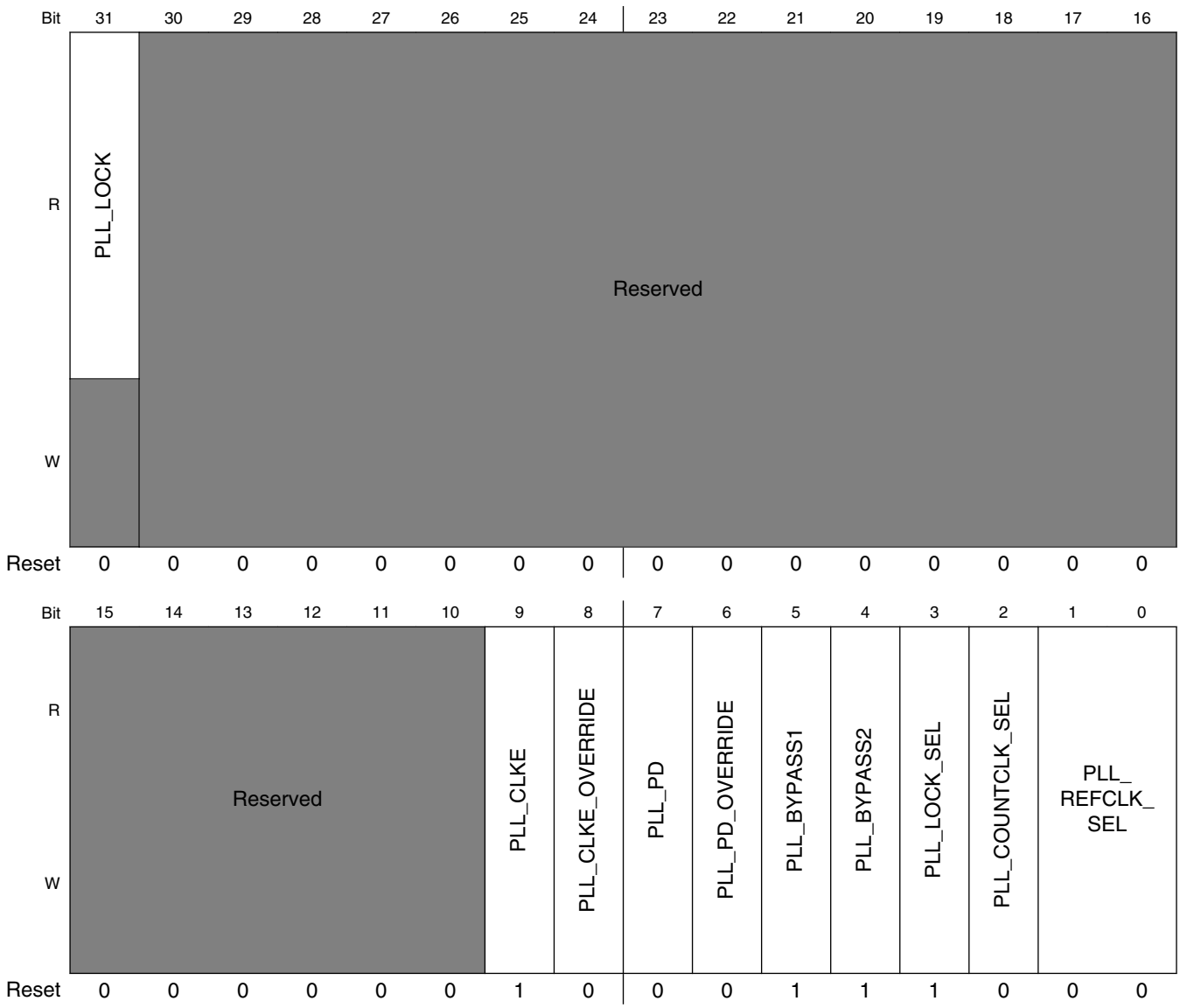
CCM_ANALOG_SYS_PLL2_CFG2 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–25 PLL_REF_DIVR1	Internal PLL1 reference clock divider value. REF must be within the range 25MHz to 235MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 8
24–19 PLL_REF_DIVR2	Internal PLL2 reference clock divider value. REF must be within the range 54MHz to 75MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 64
18–13 PLL_ FEEDBACK_ DIVF1	Internal PLL1 reference clock divider value. Fvco1 must be within the range 1600MHz to 2400MHz. Valid divider values are 1 to 64
12–7 PLL_ FEEDBACK_ DIVF2	Internal PLL2 reference clock divider value. REF must be within the range 20MHz to 1200MHz. Valid divider values are 1 to 64
6–1 PLL_OUTPUT_ DIV_VAL	Internal PLL2 output clock divider value Fout must be within the range 20MHz ~ 1200MHz. Valid divider values are 1 ~ 64
0 PLL_FILTER_ RANGE	This sets the internal PLL1 loop filter to work with the post-reference divider frequency. 0 25 to 35 MHz 1 35 to 54 MHz

5.1.8.19 System PLL Configuration 0 Register (CCM_ANALOG_SYS_PLL3_CFG0)

System PLL Configuration 0 Register

Address: 3036_0000h base + 48h offset = 3036_0048h



CCM_ANALOG_SYS_PLL3_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high

Table continues on the next page...

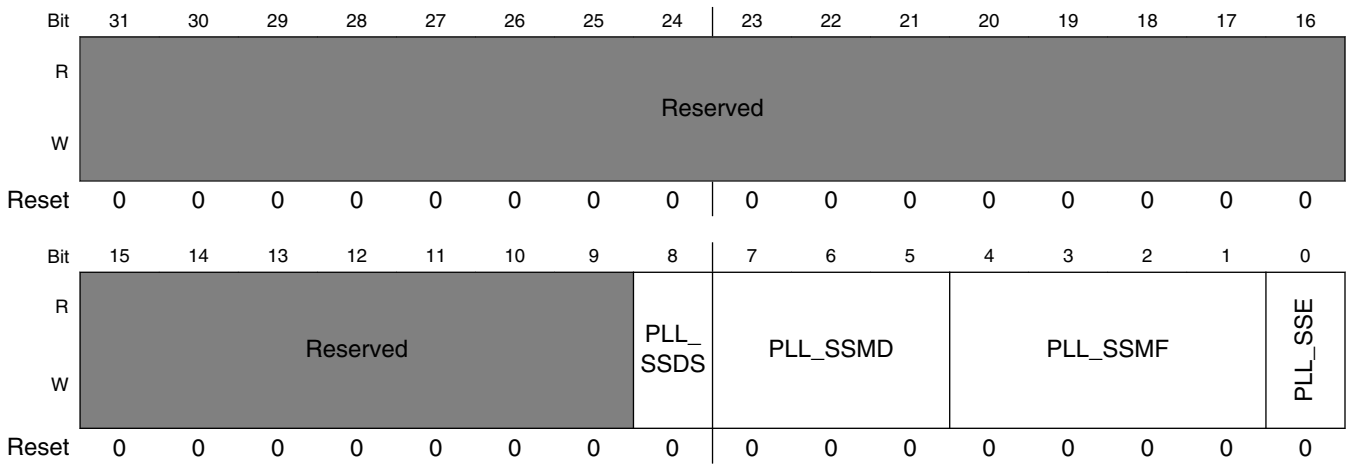
CCM_ANALOG_SYS_PLL3_CFG0 field descriptions (continued)

Field	Description
30–10 -	This field is reserved. Reserved
9 PLL_CLKE	PLL output clock divide by 20 clock gating enable active high
8 PLL_CLKE_ OVERRIDE	Override the PLL_DIV20_CLKE, clock gating enable signal from CCM block active high
7 PLL_PD	PLL output clock clock gating enable active high
6 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
5 PLL_BYPASS1	Internal PLL1 bypass control active high
4 PLL_BYPASS2	Internal PLL2 bypass control active high
3 PLL_LOCK_SEL	PLL Lock signal select 00 Select PLL lock output 01 Select maximum lock time counter output
2 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N

5.1.8.20 System_PLL Configuration 1 Register (CCM_ANALOG_SYS_PLL3_CFG1)

System PLL Configuration 1 Register

Address: 3036_0000h base + 4Ch offset = 3036_004Ch



CCM_ANALOG_SYS_PLL3_CFG1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PLL_SSDS	Selects between Spread Spectrum Center Spread and Down Spread Modes. 0 Center Spread 1 Down Spread
7–5 PLL_SSMD	Controls Spread Spectrum modulation depth 000 0.25 001 0.5 010 0.75 011 1.0 100 1.5 101 2.0 110 3.0 111 4.0
4–1 PLL_SSMF	Controls Spread Spectrum Modulation Frequency Fmod = SSMF[3:0]*2*Fvco1/251658
0 PLL_SSE	Enables Spread Spectrum Mode active high

5.1.8.21 System_PLL Configuration 2 Register (CCM_ANALOG_SYS_PLL3_CFG2)

System PLL Configuration 2 Register

Address: 3036_0000h base + 50h offset = 3036_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				PLL_REF_DIVR1				PLL_REF_DIVR2					PLL_FEEDBACK_DIVF1		
Reset	0	0	0	0	0	0	0	0	1	0	1	1	1	0	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PLL_FEEDBACK_DIVF1			PLL_FEEDBACK_DIVF2					PLL_OUTPUT_DIV_VAL							PLL_FILTER_RANGE
Reset	1	1	1	0	0	1	1	1	0	0	0	0	0	0	0	0

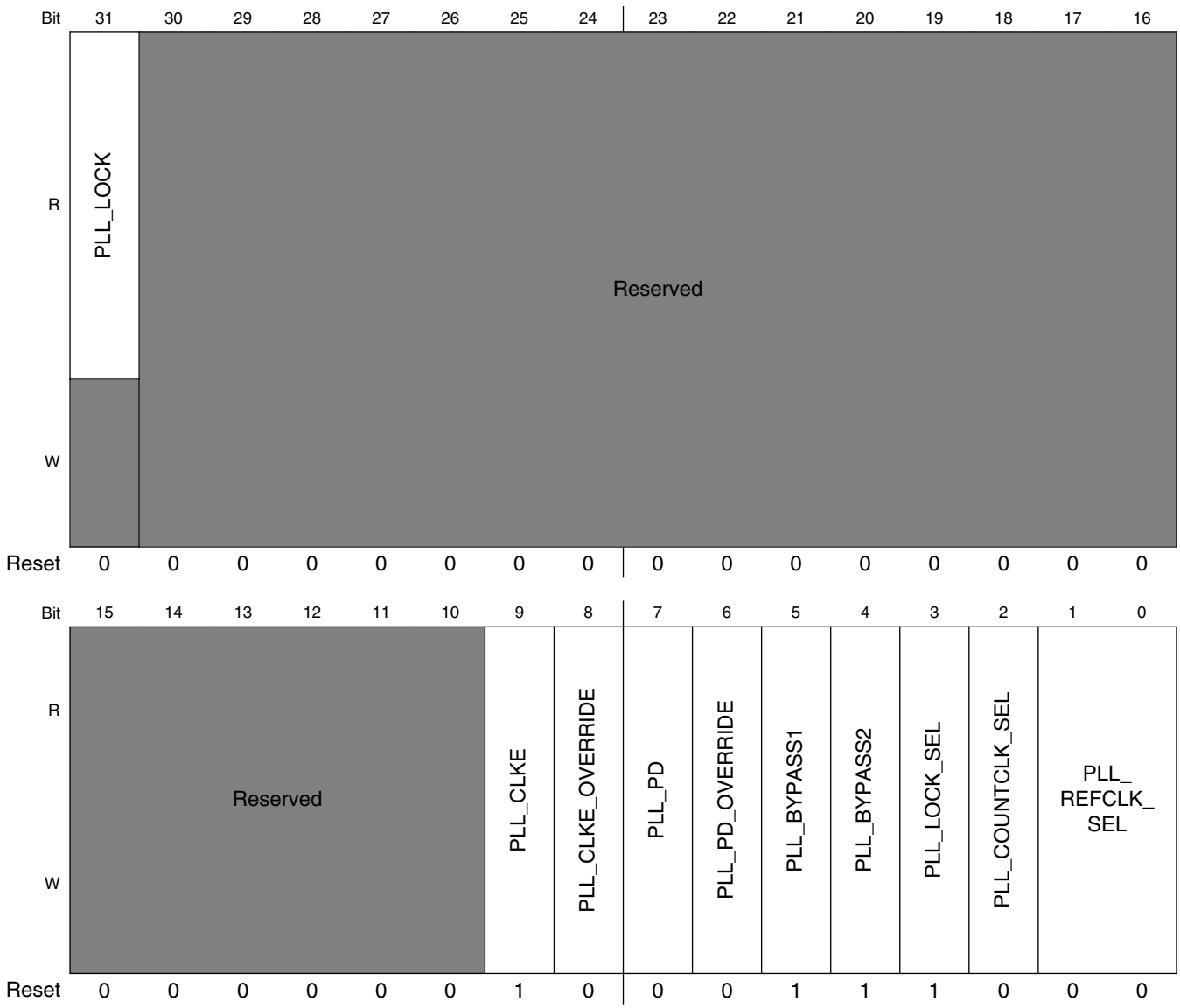
CCM_ANALOG_SYS_PLL3_CFG2 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–25 PLL_REF_DIVR1	Internal PLL1 reference clock divider value. REF must be within the range 25MHz to 235MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 8
24–19 PLL_REF_DIVR2	Internal PLL2 reference clock divider value. REF must be within the range 54MHz to 75MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 64
18–13 PLL_ FEEDBACK_ DIVF1	Internal PLL1 reference clock divider value. Fvco1 must be within the range 1600MHz to 2400MHz. Valid divider values are 1 to 64
12–7 PLL_ FEEDBACK_ DIVF2	Internal PLL2 reference clock divider value. REF must be within the range 20MHz to 1200MHz. Valid divider values are 1 to 64
6–1 PLL_OUTPUT_ DIV_VAL	Internal PLL2 output clock divider value Fout must be within the range 20MHz ~ 1200MHz. Valid divider values are 1 ~ 64
0 PLL_FILTER_ RANGE	This sets the internal PLL1 loop filter to work with the post-reference divider frequency. 0 25 to 35 MHz 1 35 to 54 MHz

5.1.8.22 VIDEO PLL2 Configuration 0 Register (CCM_ANALOG_VIDEO_PLL2_CFG0)

VIDEO PLL2 Configuration 0 Register

Address: 3036_0000h base + 54h offset = 3036_0054h



CCM_ANALOG_VIDEO_PLL2_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high

Table continues on the next page...

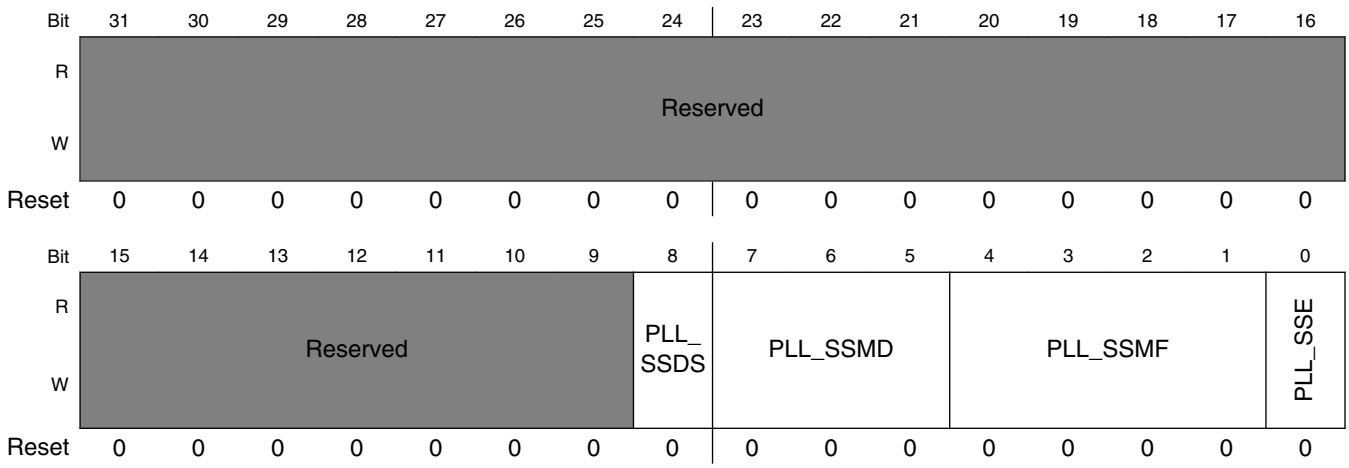
CCM_ANALOG_VIDEO_PLL2_CFG0 field descriptions (continued)

Field	Description
30–10 -	This field is reserved. Reserved
9 PLL_CLKE	PLL output clock clock gating enable active high
8 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
7 PLL_PD	PLL output clock clock gating enable active high
6 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
5 PLL_BYPASS1	Internal PLL1 bypass control active high
4 PLL_BYPASS2	Internal PLL2 bypass control active high
3 PLL_LOCK_SEL	PLL Lock signal select 00 Select PLL lock output 01 Select maximum lock time counter output
2 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N

5.1.8.23 VIDEO PLL2 Configuration 1 Register (CCM_ANALOG_VIDEO_PLL2_CFG1)

VIDEO PLL2 Configuration 1 Register

Address: 3036_0000h base + 58h offset = 3036_0058h



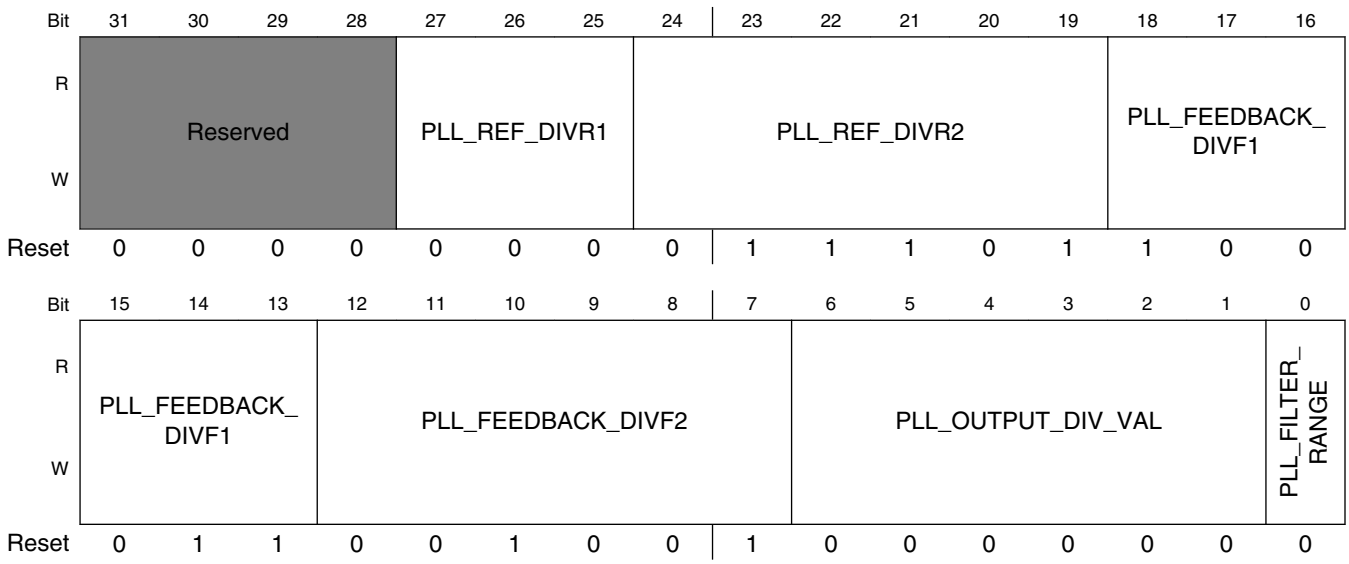
CCM_ANALOG_VIDEO_PLL2_CFG1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PLL_SSDS	Selects between Spread Spectrum Center Spread and Down Spread Modes. 0 Center Spread 1 Down Spread
7–5 PLL_SSMD	Controls Spread Spectrum modulation depth 000 0.25 001 0.5 010 0.75 011 1.0 100 1.5 101 2.0 110 3.0 111 4.0
4–1 PLL_SSMF	Controls Spread Spectrum Modulation Frequency Fmod = SSMF[3:0]*2*Fvco1/251658
0 PLL_SSE	Enables Spread Spectrum Mode active high

5.1.8.24 VIDEO PLL2 Configuration 2 Register (CCM_ANALOG_VIDEO_PLL2_CFG2)

VIDEO PLL2 Configuration 2 Register

Address: 3036_0000h base + 5Ch offset = 3036_005Ch



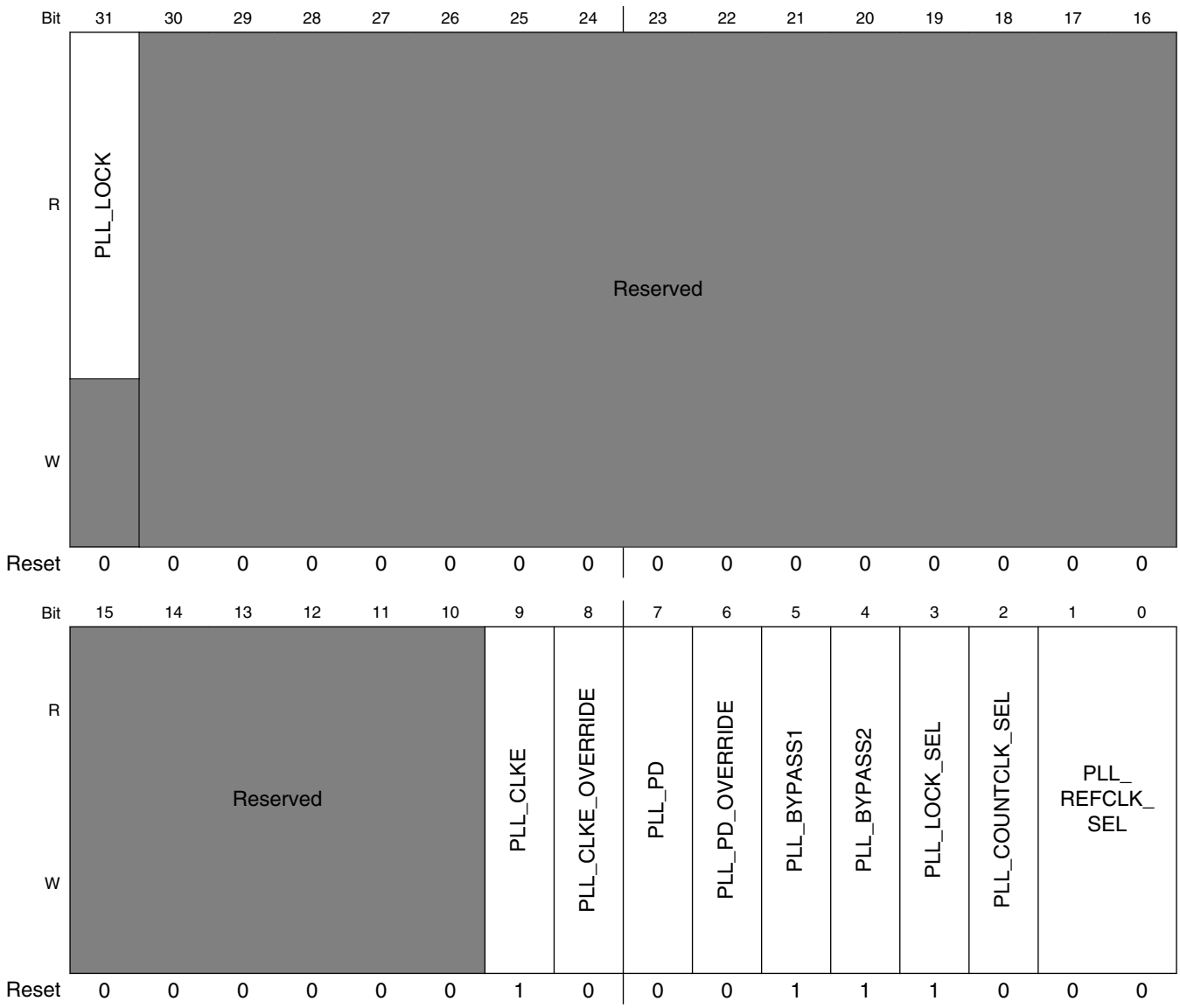
CCM_ANALOG_VIDEO_PLL2_CFG2 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–25 PLL_REF_DIVR1	Internal PLL1 reference clock divider value. REF must be within the range 25MHz to 235MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 8
24–19 PLL_REF_DIVR2	Internal PLL2 reference clock divider value. REF must be within the range 54MHz to 75MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 64
18–13 PLL_ FEEDBACK_ DIVF1	Internal PLL1 reference clock divider value. Fvco1 must be within the range 1600MHz to 2400MHz. Valid divider values are 1 to 64
12–7 PLL_ FEEDBACK_ DIVF2	Internal PLL2 reference clock divider value. REF must be within the range 20MHz to 1200MHz. Valid divider values are 1 to 64
6–1 PLL_OUTPUT_ DIV_VAL	Internal PLL2 output clock divider value Fout must be within the range 20MHz ~ 1200MHz. Valid divider values are 1 ~ 64
0 PLL_FILTER_ RANGE	This sets the internal PLL1 loop filter to work with the post-reference divider frequency. 0 25 to 35 MHz 1 35 to 54 MHz

5.1.8.25 DRAM PLL Configuration 0 Register (CCM_ANALOG_DRAM_PLL_CFG0)

DRAM PLL Configuration 0 Register

Address: 3036_0000h base + 60h offset = 3036_0060h



CCM_ANALOG_DRAM_PLL_CFG0 field descriptions

Field	Description
31 PLL_LOCK	PLL lock status active high

Table continues on the next page...

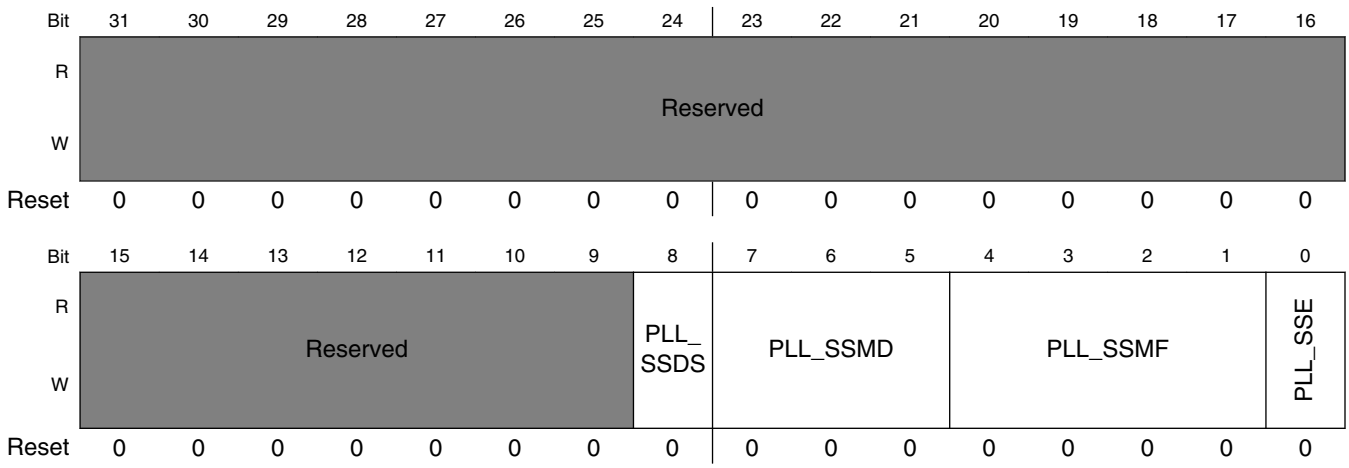
CCM_ANALOG_DRAM_PLL_CFG0 field descriptions (continued)

Field	Description
30–10 -	This field is reserved. Reserved
9 PLL_CLKE	PLL output clock clock gating enable active high
8 PLL_CLKE_ OVERRIDE	Override the PLL_CLKE, clock gating enable signal from CCM block active high
7 PLL_PD	PLL output clock clock gating enable active high
6 PLL_PD_ OVERRIDE	Override the PLL_PD, clock gating enable signal from CCM block active high
5 PLL_BYPASS1	Internal PLL1 bypass control active high
4 PLL_BYPASS2	Internal PLL2 bypass control active high
3 PLL_LOCK_SEL	PLL Lock signal select 00 Select PLL lock output 01 Select maximum lock time counter output
2 PLL_ COUNTCLK_ SEL	PLL maximum lock time counter clock select 0 25M_REF_CLK 1 27M_REF_CLK
PLL_REFCLK_ SEL	PLL reference clocks select 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK_P_N

5.1.8.26 DRAM PLL Configuration 1 Register (CCM_ANALOG_DRAM_PLL_CFG1)

DRAM PLL Configuration 1 Register

Address: 3036_0000h base + 64h offset = 3036_0064h



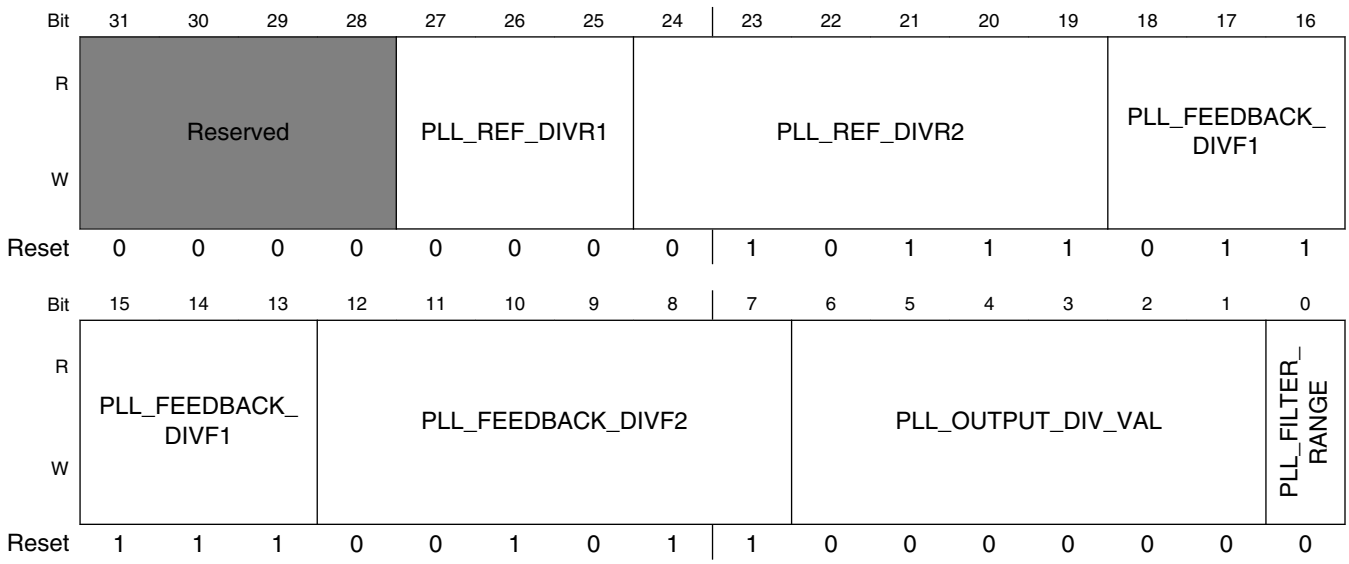
CCM_ANALOG_DRAM_PLL_CFG1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 PLL_SSDS	Selects between Spread Spectrum Center Spread and Down Spread Modes. 0 Center Spread 1 Down Spread
7–5 PLL_SSMD	Controls Spread Spectrum modulation depth 000 0.25 001 0.5 010 0.75 011 1.0 100 1.5 101 2.0 110 3.0 111 4.0
4–1 PLL_SSMF	Controls Spread Spectrum Modulation Frequency Fmod = SSMF[3:0]*2*Fvco1/251658
0 PLL_SSE	Enables Spread Spectrum Mode active high

5.1.8.27 DRAM PLL Configuration 2 Register (CCM_ANALOG_DRAM_PLL_CFG2)

DRAM PLL Configuration 2 Register

Address: 3036_0000h base + 68h offset = 3036_0068h



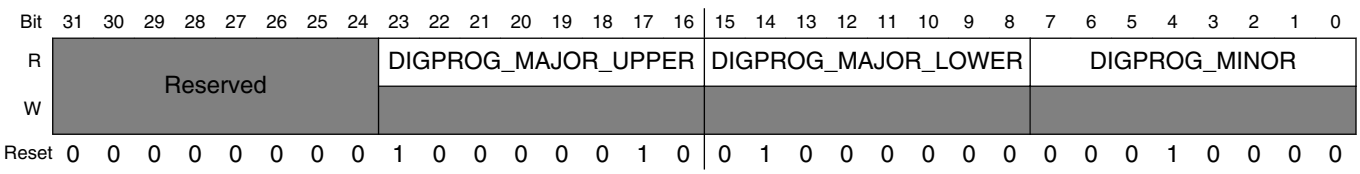
CCM_ANALOG_DRAM_PLL_CFG2 field descriptions

Field	Description
31–28 -	This field is reserved. Reserved
27–25 PLL_REF_DIVR1	Internal PLL1 reference clock divider value. REF must be within the range 25MHz to 235MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 8
24–19 PLL_REF_DIVR2	Internal PLL2 reference clock divider value. REF must be within the range 54MHz to 75MHz. post_divide REF must be within the range 25MHz to 54MHz. Valid divider values are 1 to 64
18–13 PLL_ FEEDBACK_ DIVF1	Internal PLL1 reference clock divider value. Fvco1 must be within the range 1600MHz to 2400MHz. Valid divider values are 1 to 64
12–7 PLL_ FEEDBACK_ DIVF2	Internal PLL2 reference clock divider value. REF must be within the range 20MHz to 1200MHz. Valid divider values are 1 to 64
6–1 PLL_OUTPUT_ DIV_VAL	Internal PLL2 output clock divider value Fout must be within the range 20MHz ~ 1200MHz. Valid divider values are 1 ~ 64
0 PLL_FILTER_ RANGE	This sets the internal PLL1 loop filter to work with the post-reference divider frequency. 0 25 to 35 MHz 1 35 to 54 MHz

5.1.8.28 DIGPROG Register (CCM_ANALOG_DIGPROG)

DIGPROG Register

Address: 3036_0000h base + 6Ch offset = 3036_006Ch



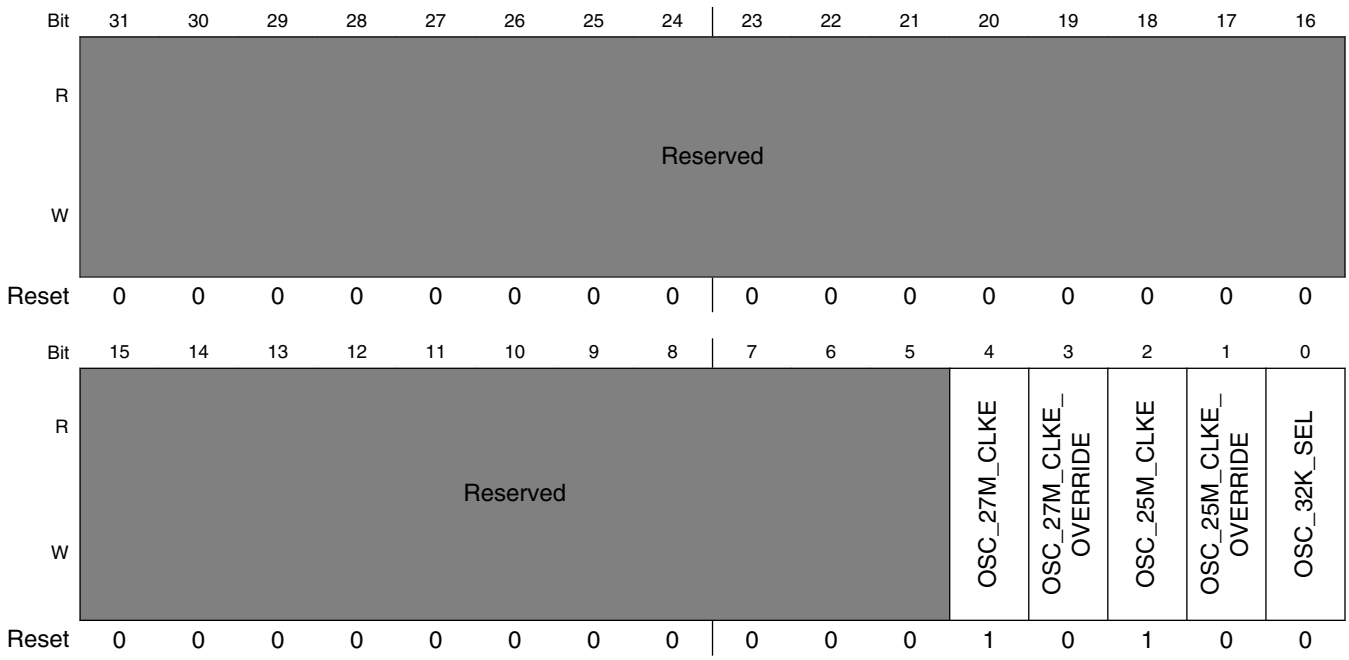
CCM_ANALOG_DIGPROG field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–16 DIGPROG_ MAJOR_UPPER	Bit[7:4] is 0x8, stands for “i.MX8” Bit[3:0] is 0x2, stands for ”M”
15–8 DIGPROG_ MAJOR_LOWER	Bit[7:4] is 0x4, stands for “Quad” Bit[3:0] is 0x0, stands for “Lite”
DIGPROG_ MINOR	Bit[7:4] is the base layer revision, Bit[3:0] is the metal layer revision 0x10 stands for Tapeout 1.0

5.1.8.29 Osc Misc Configuration Register (CCM_ANALOG_OSC_MISC_CFG)

Osc Misc Register

Address: 3036_0000h base + 70h offset = 3036_0070h



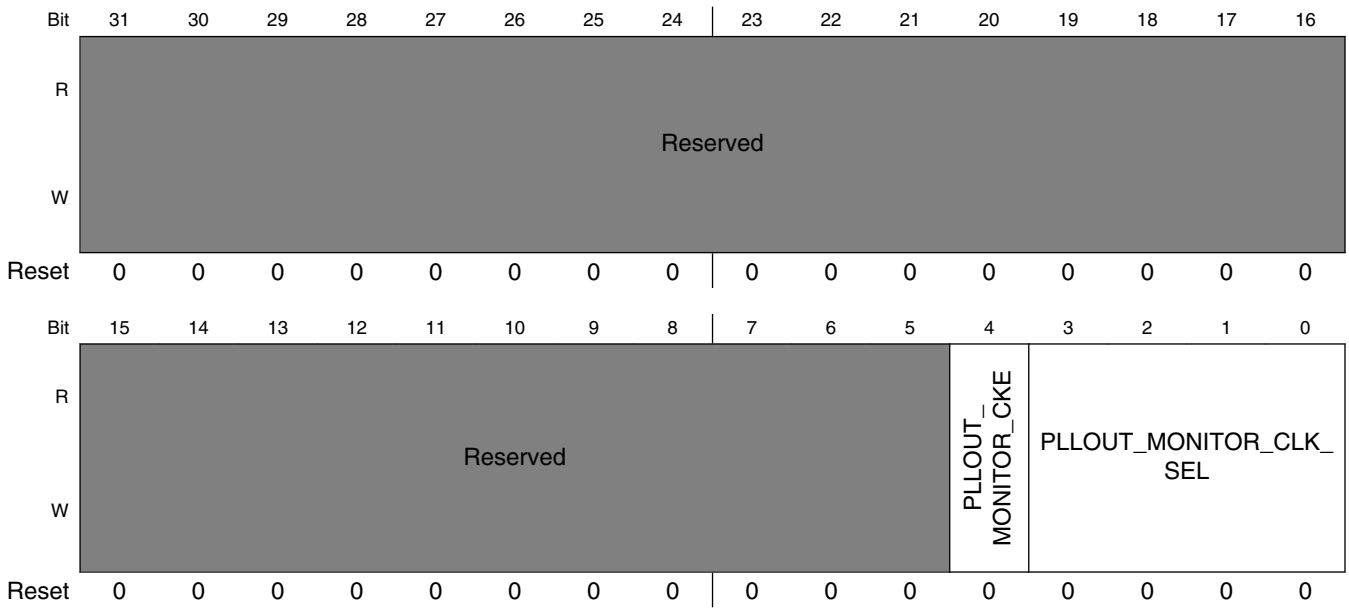
CCM_ANALOG_OSC_MISC_CFG field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 OSC_27M_CLKE	27MHz OSC output clock gating enable active high
3 OSC_27M_ CLKE_ OVERRIDE	Override the OSC_27M_CLKE, clock gating enable signal from CCM block active high
2 OSC_25M_CLKE	25MHz OSC output clock gating enable active high
1 OSC_25M_ CLKE_ OVERRIDE	Override the OSC_27M_CLKE, clock gating enable signal from CCM block active high
0 OSC_32K_SEL	32KHz OSC input select 0 25M_REF_CLK_DIV800 1 RTC

5.1.8.30 PLLOUT Monitor Configuration Register (CCM_ANALOG_PLLOUT_MONITOR_CFG)

PLLOUT Monitor Configuration Register contains bits to control the clock that will be generated on the CCM clock mapped to CLK2_P/N.

Address: 3036_0000h base + 74h offset = 3036_0074h



CCM_ANALOG_PLLOUT_MONITOR_CFG field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 PLLOUT_ MONITOR_CKE	Clock monitor output clock gating enable active high
PLLOUT_ MONITOR_CLK_ SEL	Clock monitor output clock select: 00 25M_REF_CLK 01 27M_REF_CLK 10 HDMI_PHY_27M_CLK 11 CLK1_P_N 100 OSC_32K_CLK 101 AUDIO_PLL1_CLK 110 AUDIO_PLL2_CLK 111 GPU_PLL_CLK 1000 VPU_PLL_CLK 1001 VIDEO_PLL1_CLK

Table continues on the next page...

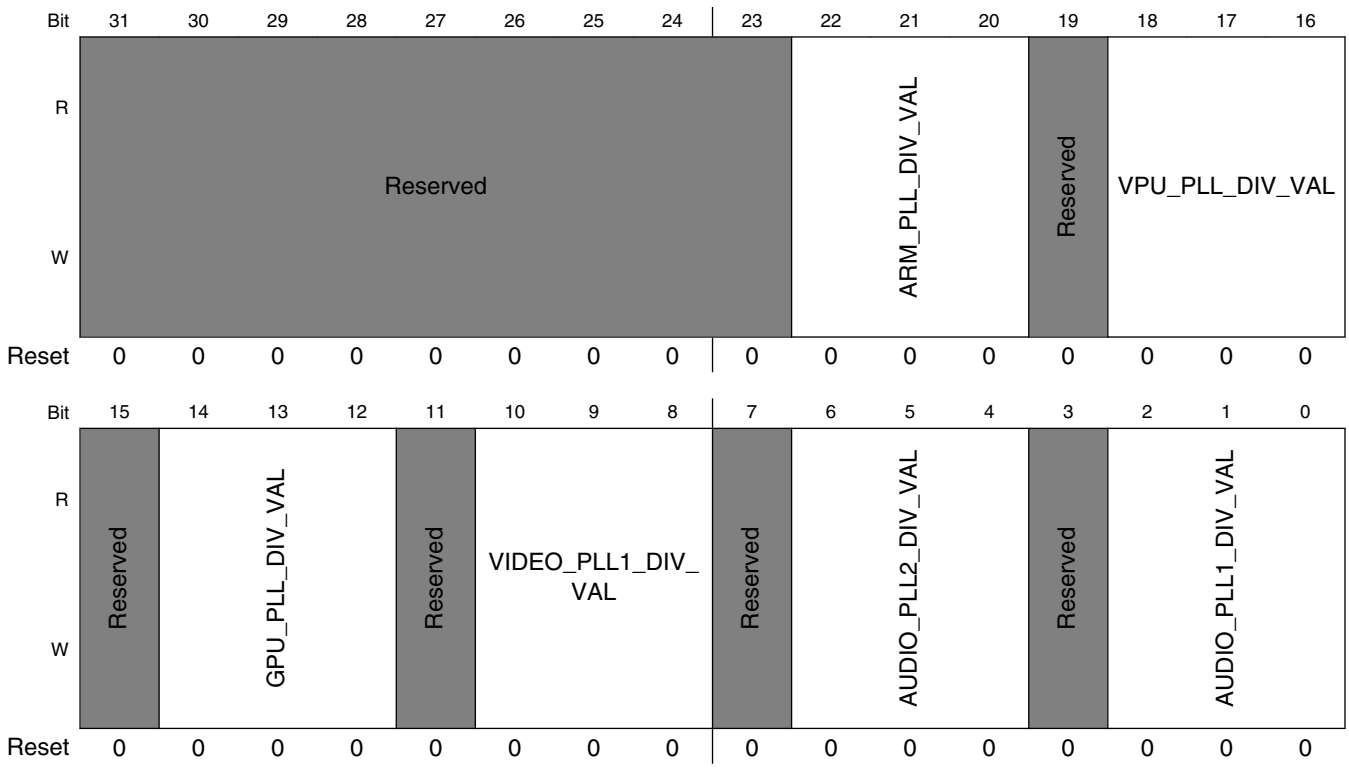
CCM_ANALOG_PLLOUT_MONITOR_CFG field descriptions (continued)

Field	Description
	1010 ARM_PLL_CLK 1011 SYSTEM_PLL1_CLK 1100 SYSTEM_PLL2_CLK 1101 SYSTEM_PLL3_CLK 1110 VIDEO_PLL2_CLK 1111 DRAM_PLL_CLK

5.1.8.31 Fractional PLLOUT Divider Configuration Register (CCM_ANALOG_FRAC_PLLOUT_DIV_CFG)

Fractional PLLOUT Divider Configuration Register

Address: 3036_0000h base + 78h offset = 3036_0078h



CCM_ANALOG_FRAC_PLLOUT_DIV_CFG field descriptions

Field	Description
31–23 -	This field is reserved. Reserved

Table continues on the next page...

CCM_ANALOG_FRAC_PLLOUT_DIV_CFG field descriptions (continued)

Field	Description
22–20 ARM_PLL_DIV_VAL	ARM PLL clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
19 -	This field is reserved. Reserved
18–16 VPU_PLL_DIV_VAL	VPU PLL clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 -	This field is reserved. Reserved
14–12 GPU_PLL_DIV_VAL	GPU PLL clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
11 -	This field is reserved. Reserved
10–8 VIDEO_PLL1_DIV_VAL	VIDEO PLL1 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
7 -	This field is reserved. Reserved

Table continues on the next page...

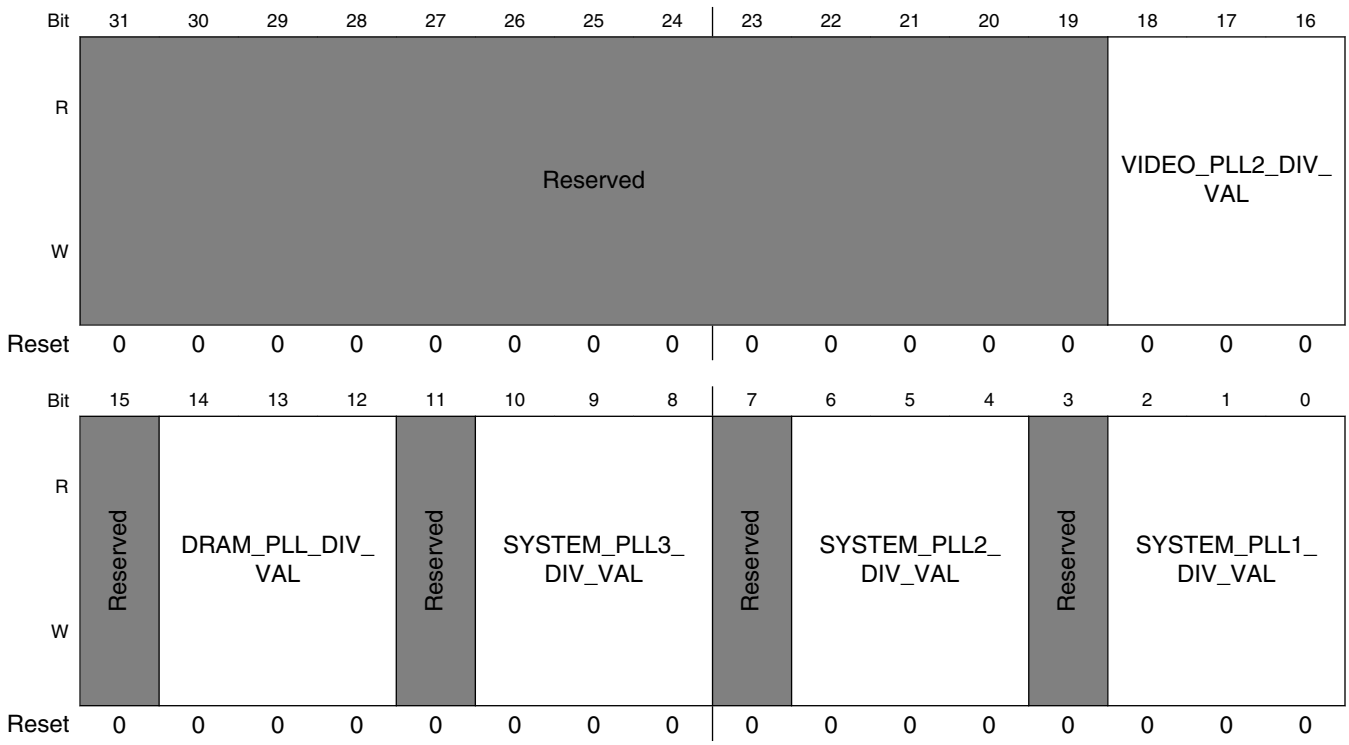
CCM_ANALOG_FRAC_PLLOUT_DIV_CFG field descriptions (continued)

Field	Description
6–4 AUDIO_PLL2_DIV_VAL	AUDIO PLL2 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
3 -	This field is reserved. Reserved
AUDIO_PLL1_DIV_VAL	AUDIO PLL1 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.1.8.32 SCCG PLLOUT Divider Configuration Register (CCM_ANALOG_SCCG_PLLOUT_DIV_CFG)

SCCG PLLOUT Divider Configuration Register

Address: 3036_0000h base + 7Ch offset = 3036_007Ch



CCM_ANALOG_SCCG_PLLOUT_DIV_CFG field descriptions

Field	Description
31–19 -	This field is reserved. Reserved
18–16 VIDEO_PLL2_DIV_VAL	VIDEO PLL2 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
15 -	This field is reserved. Reserved

Table continues on the next page...

CCM_ANALOG_SCCG_PLLOUT_DIV_CFG field descriptions (continued)

Field	Description
14–12 DRAM_PLL_ DIV_VAL	DRAM PLL clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
11 -	This field is reserved. Reserved
10–8 SYSTEM_PLL3_ DIV_VAL	System PLL3 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
7 -	This field is reserved. Reserved
6–4 SYSTEM_PLL2_ DIV_VAL	System PLL2 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8
3 -	This field is reserved. Reserved
SYSTEM_PLL1_ DIV_VAL	System PLL1 clock divider value, for test purpose. 000 Divide by 1 001 Divide by 2 010 Divide by 3 011 Divide by 4 100 Divide by 5 101 Divide by 6 110 Divide by 7 111 Divide by 8

5.2 General Power Controller (GPC)

5.2.1 Overview

The General Power Controller (GPC) module controls the following functions:

- Provide low power mode control for A53 and M4 platform
- Provide Power domain management all Arm and SOC power domain
- Provide domain control mechanism based on A53 and M4 CPU domain
- Provide handshake with CCM for clock management in low power mode
- Provide handshake with SRC for power down and power up sequence
- Provide handshake with Analog for Deep Sleep Mode control

5.2.2 Features

The General Power Controller (GPC) module controls the following functions:

- Support programmable feature for WAIT/STOP/DSM low power mode
- Support time slot based power domain control
- Support flexible sleep and wakeup condition
- Support domain control for multi CPU platforms system
- All register accessed by IP bus
- Interface for the following IPs:
    - CCM – clock controller module
    - SRC – system reset controller
    - ANALOG – miscellaneous analog control

5.2.3 Block Diagram

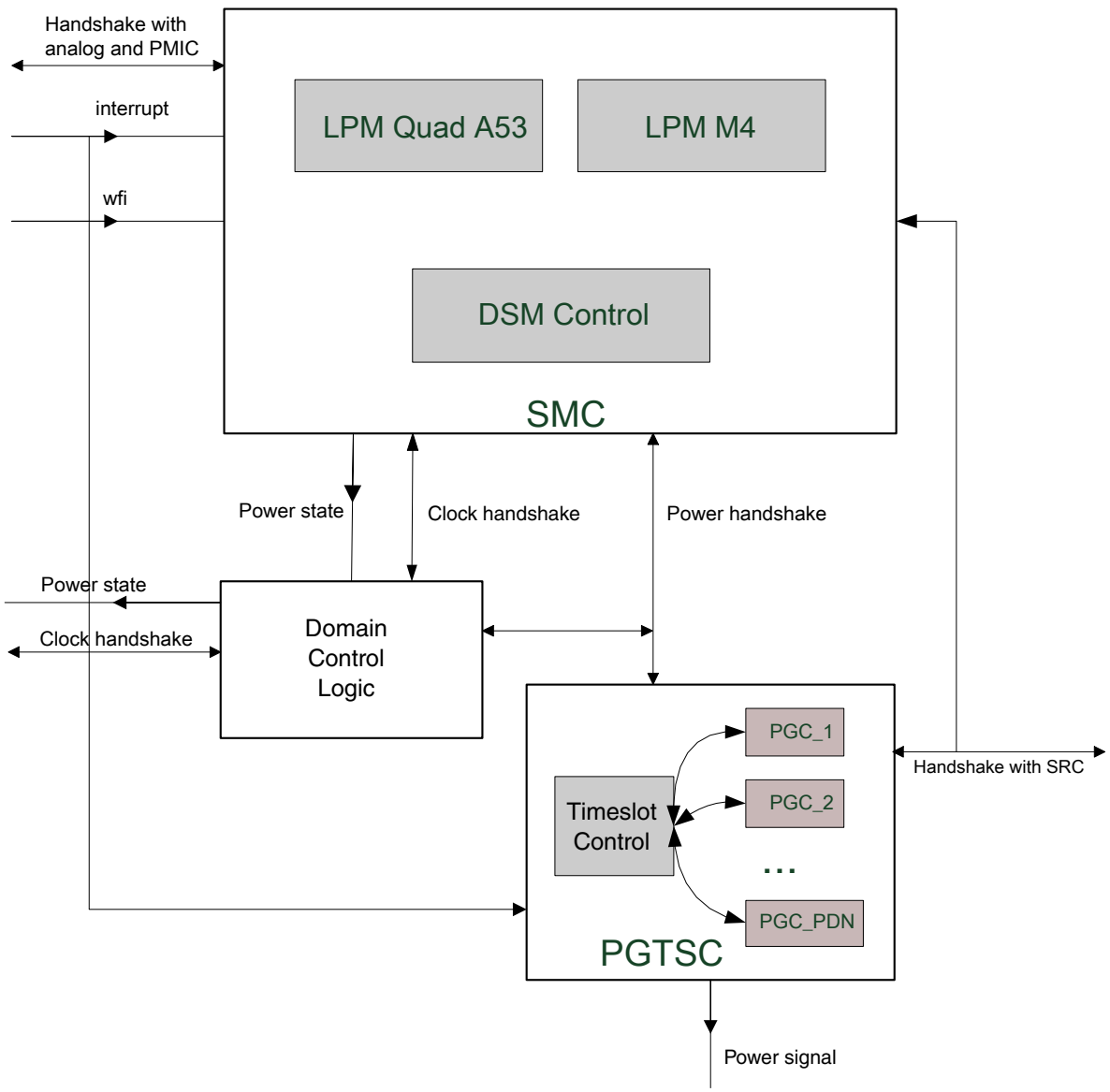


Figure 5-10. GPC Block Diagram

The GPC module contains two sub-modules: System Mode Controller (SMC) and Power Gating Time Slot Control (PGTSC):

- GPC Top: the top level GPC. It also includes the top memory map and registers, domain control information, and memory low power control.
- System Mode Controller (SMC):
  - The SMC supports two low power modes (LPM), WAIT and STOP. Each LPM corresponds to one mode for A53 platform and one mode for M4 platform.

- SMC controls the power sequence in Deep Sleep Mode (DSM)
- SMC support the power up and power down of A53 core0/core1/core2/core3 by IRQ/WFI signals without LPM triggered
- SMC can translate the LPM request for A53 and M4 platform to power up and power down request to PGTSC.
- Power Gating Time Slot Control (PGTSC):
  - The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems. For subsystems to be completely powered down in low power modes, a specific sequence of power control signals must be followed. The sequence timing is programmable using the PGC control registers.
  - There are 20 PGCs in the chip, all of them can be power down/up with a software trigger and all of them can be mapped to 20 timing slots and power-up and power down by request from SMC.

5.2.4 Functional Description

5.2.4.1 RUN mode

This is the normal/functional operating mode. In this mode, the CPU runs in its normal operational mode.

5.2.4.2 Low power mode

There are two CPU platforms (each of them represents a CPU domain): Quad core Cortex A53 platform and Cortex M4 platform. Each platform supports two low power modes: WAIT mode and STOP mode.

5.2.4.2.1 WAIT mode

In this low power mode:

- LPCG can be defined to be shut off or not in wait mode for each CPU domain
- PLL can be defined to be shut off or not in wait mode for each CPU domain

NOTE

The PLLs will only been closed in non-fast wake-up mode, relevant bit are

GPC_SLPCR[EN_A53_FASTWUP_WAIT_MODE] and
GPC_SLPCR[EN_M4_FASTWUP_WAIT_MODE]

- CPU clock can be defined been shut off or not in wait for each CPU platform. (GPC_LPCR_A53_BSC[CPU_CLK_ON_LPM] and GPC_LPCR_M4[CPU_CLK_ON_LPM])
- Power of different power domain can be defined be shut off or not in wait mode for each platform domain
- Some peripherals may go to wait mode along with A53 or M4 platform.

5.2.4.2.2 STOP mode

In this low power mode:

- LPCG can be defined been shut off or not in stop mode for each CPU domain
- PLL can be defined been shut off or not in stop mode for each CPU domain

NOTE

The PLLs will only been closed in non-fast wake-up mode, relevant bit are
GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] and
GPC_SLPCR[EN_M4_FASTWUP_STOP_MODE]

- CPU clock can be defined been shut off or not in stop for each CPU (GPC_LPCR_A53_BSC[CPU_CLK_ON_LPM] and GPC_LPCR_M4[CPU_CLK_ON_LPM])
- Power of different power domain can be defined be shut off or not in stop mode for each platform domain
- Some peripherals may go to stop mode along with A53 or M4 platform.

5.2.4.3 Deep Sleep Mode

The Deep Sleep Mode (DSM) is a system low power mode.

In this mode:

- On-chip OSC can be defined to be shut off or not in DSM (GPC_SLPCR[SBYOS])
- PMIC can be defined to be stand-by mode or not in DSM (GPC_SLPCR[VSTBY])
- Regulator can be defined to be BYPASS mode or not in DSM (GPC_SLPCR[RBC_EN])
- Memory can be defined to go to retention mode or not in DSM(GPC_MLPCR[MEMLP_CTL_DIS])
- A53 platform power (VDD_ARM) can be defined to be shut off or not in DSM.

NOTE

CCM configuration must make sure close all PLLs before system goes to DSM

NOTE

For the SoC to correctly power up after entering DSM, CCM_PLL_CTRLx must not be set to 0x0 or 0x3 for any domain in use.

5.2.4.4 LPM Sleep Process

CPU platform will go to WAIT/STOP under the following conditions:

- LPM registers (GPC_LPCR_A53_BSC[LPM0], GPC_LPCR_A53_BSC[LPM1], GPC_LPCR_A53_BSC2[LPM2], GPC_LPCR_A53_BSC2[LPM3], GPC_LPCR_M4[LPM]) are set to WAIT or STOP.

  NOTE

  Since A53 platform has four cores, the so each core has its own LPM register: GPC_LPCR_A53_BSC[LPM0] and GPC_LPCR_A53_BSC[LPM1], GPC_LPCR_A53_BSC2[LPM2] and GPC_LPCR_A53_BSC2[LPM3]. The unified LPM of A53 will be generated with the lower LPM of the cores.

- Asserting the WFI signal will trigger CPU sleep process. There are five WFIs that come from A53 platform: WFI_core0, WFI_core1, WFI_core2, WFI_core3, and WFI_scu. The A53 platform will go to LPM when all WFIs are asserted. If the GPC_LPCR_A53_AD[EN_C0_WFI_PDN] or GPC_LPCR_A53_AD[EN_C1_WFI_PDN] or GPC_LPCR_A53_AD[EN_C2_WFI_PDN] or GPC_LPCR_A53_AD[EN_C3_WFI_PDN] bit is set, the LPM trigger condition will be a little different. See Power control for A53 Platform for more information. Only one WFI comes from M4 platform. The M4 platform will go to LPM when WFI_M4 asserted.

  NOTE

  WFI condition can be masked by register bits GPC_LPCR_A53_BSC[MASK_n_WFI] and GPC_LPCR_M4[MASK_M4_WFI]

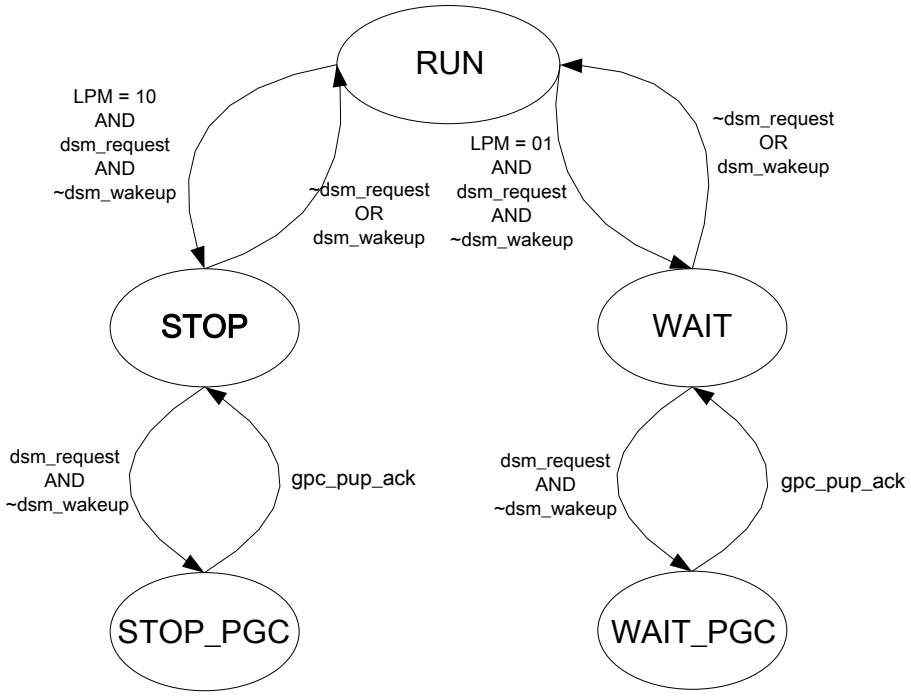


Figure 5-11. LPM transition inside CPU platform

System will go to DSM under the following conditions:

- Both A53 and M4 are STOP mode.
- Both GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] and GPC_SLPCR[EN_M4_FASTWUP_STOP_MODE] are not set.
- GPC_SLPCR[EN_DSM] is set.

NOTE

If GPC_LPCR_M4[MASK_DSM_TRIGGER] is set, the system will go to DSM when A53 goes STOP mode and GPC_SLPCR[EN_A53_FASTWUP_STOP_MODE] is not set. If GPC_LPCR_A53_BSC[MASK_DSM_TRIGGER] is set, the system will go to DSM when M4 goes to STOP mode and GPC_SLPCR[EN_M4_FASTWUP_STOP_MODE] not set. GPC_LPCR_M4[MASK_DSM_TRIGGER] and GPC_LPCR_A53_BSC[MASK_DSM_TRIGGER] cannot be set at the same time.

5.2.4.5 LPM Wake Up Process

DSM, STOP, WAIT mode will be woken up by interrupts:

- The CPU platforms share the same IRQ sources in this chip. Software can use GPC_IMRn_CORE0_A53, GPC_IMRn_CORE1_A53, GPC_IMRn_CORE2_A53, GPC_IMRn_CORE3_A53, and GPC_IMRn_M4 to separate the 128 bits IRQ sources to A53 core0, core1, core2, and core3, and M4 platform.
- The A53 core0, core1, core2, and core3 IRQ can also be from GIC source (defined by GPC_LPCR_A53_BSC[IRQ_SRC_C3], GPC_LPCR_A53_BSC[IRQ_SRC_C2], GPC_LPCR_A53_BSC[IRQ_SRC_C1], and GPC_LPCR_A53_BSC[IRQ_SRC_C0]) and if it is chosen from GIC source the GPC_IMRn_x_A53 will lose its function. See Power control for A53 Platform for more information.
- Interrupts for both A53 and M4 will cause the system wake up from DSM, A53 interrupt will wake up A53 from LPM, M4 interrupt will wake up M4 from LPM.

5.2.5 Power Gating Controller (PGC) Overview

The Power Gating Controller (PGC) is a power management component that controls the power-down and power-up sequencing of individual subsystems. For subsystems to be completely powered down in low power modes, a specific sequence of power control signals must be followed. The sequence timing is programmable using the PGC control registers.

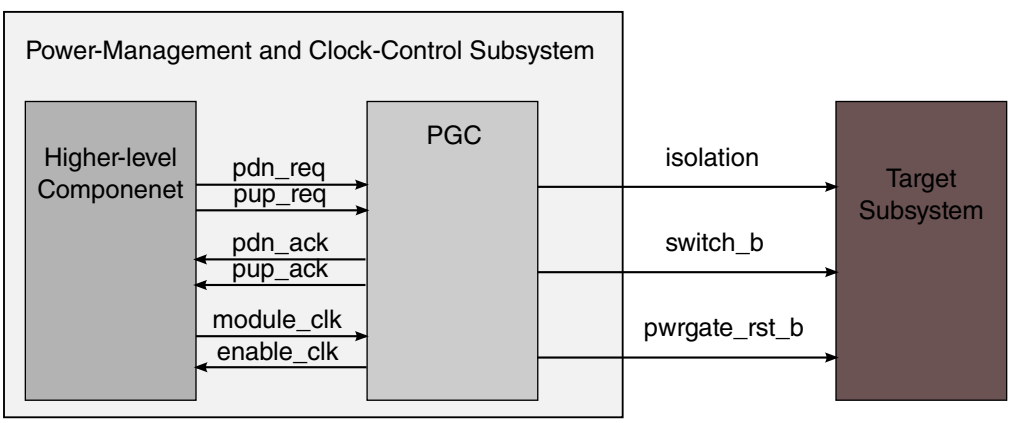


Figure 5-12. Power Gating Controller (PGC)

5.2.5.1 PGC power domains

The following table lists the PGCs in the chip and the corresponding power domain. There are three types of PGC - CPU, MIX, and PU. The power for the PUs (Power Units) can be controlled by the GPC.

PGC type	Type	Power domain
PGC_C0	CPU	Core0 of A53 platform
PGC_C1	CPU	Core1 of A53 platform
PGC_C2	CPU	Core2 of A53 platform
PGC_C3	CPU	Core3 of A53 platform
PGC_SCU	CPU	SCU/L2 cache RAM of A53 platform
PGC_MF	MIX	Fastmix and Megamix
PGC_OTG1	PU	USB OTG1 PHY
PGC_OTG2	PU	USB OTG2 PHY
PGC_PCIE	PU	PCIE1 PHY
PGC_MIPI	PU	MIPI DSI PHY
PGC_DDR1	PU	DDR1
PGC_DDR2	PU	DDR2
PGC_VPU	PU	VPU
PGC_GPU	PU	GPU
PGC_HDMI	PU	HDMI
PGC_DISP	PU	DISP
PGC_MIPI_CSI1	PU	MIPI CSI1 PHY
PGC_MIPI_CSI2	PU	MIPI CSI2 PHY
PGC_PCIE2	PU	PCIE2 PHY

5.2.5.2 Trigger to PGC: Hardware and Software Requests

All PGCs (except fastmix/megamix PGC) can be power up/down by hardware or software request. Fastmix/megamix PGC can only be power up/down by hardware request.

The LPM controller for each platform can generate hardware power down or power up request. All hardware requests will be mapped to “timeslot controller” before they goes to relevant PGC (see “Time slot control for PGCs” for more information).

The CPU can also generate software power up or power down request to relevant PGCs. The software trigger will not be mapped to timeslot control. If there are PGCs in software PDN/PUP sequence the request from LPM will be masked. The software trigger will also be failed if the timeslot control is in “busy” state.

All power up/down request to PGCs will be mapped to domain control module (see “Domain control for PGCs ”).

PGC_C0, PGC_C1, PGC_C2, and PGC_C3 can be triggered by its own “WFI/IRQ” without LPM trigger and time slot. See Power control for A53 Platform for more information.

5.2.5.3 Time slot control mechanism for PGCs

GPC uses a time slot controller to control the PGC sub-systems, such as the PGC in CPU0, CPU1,CPU2, CPU3, MIX, PCIE PHY, etc. We use it for below reasons when system wakes from low power mode.

1. Support flexible power down/up sequence for different sub-system
2. Sub-system can be power up in different slot to avoid large ramping up current in case they start ramping up at the same time

There are a total of 21 time slots used in the chip, one or more PGCs are used for power up or power down in each of these slots. The time slot controller will sample power up/down requests at slot0. If there are power up/down requests from SMC, it will scan from slot0 to slot20. When the scan process comes to one slot which has a defined power up or power down for one or more PGCs (defined by “SLTn_CFG”), the power up/down sequence of relevant PGCs will happen in that slot. Otherwise, the relevant slot will be skipped.

The next slot will not begin until the all PGCs finish their power up or power down process in current time slot. When all the 20 slots are finished, slot controller will jump to IDLE state and monitor new request from SMC.

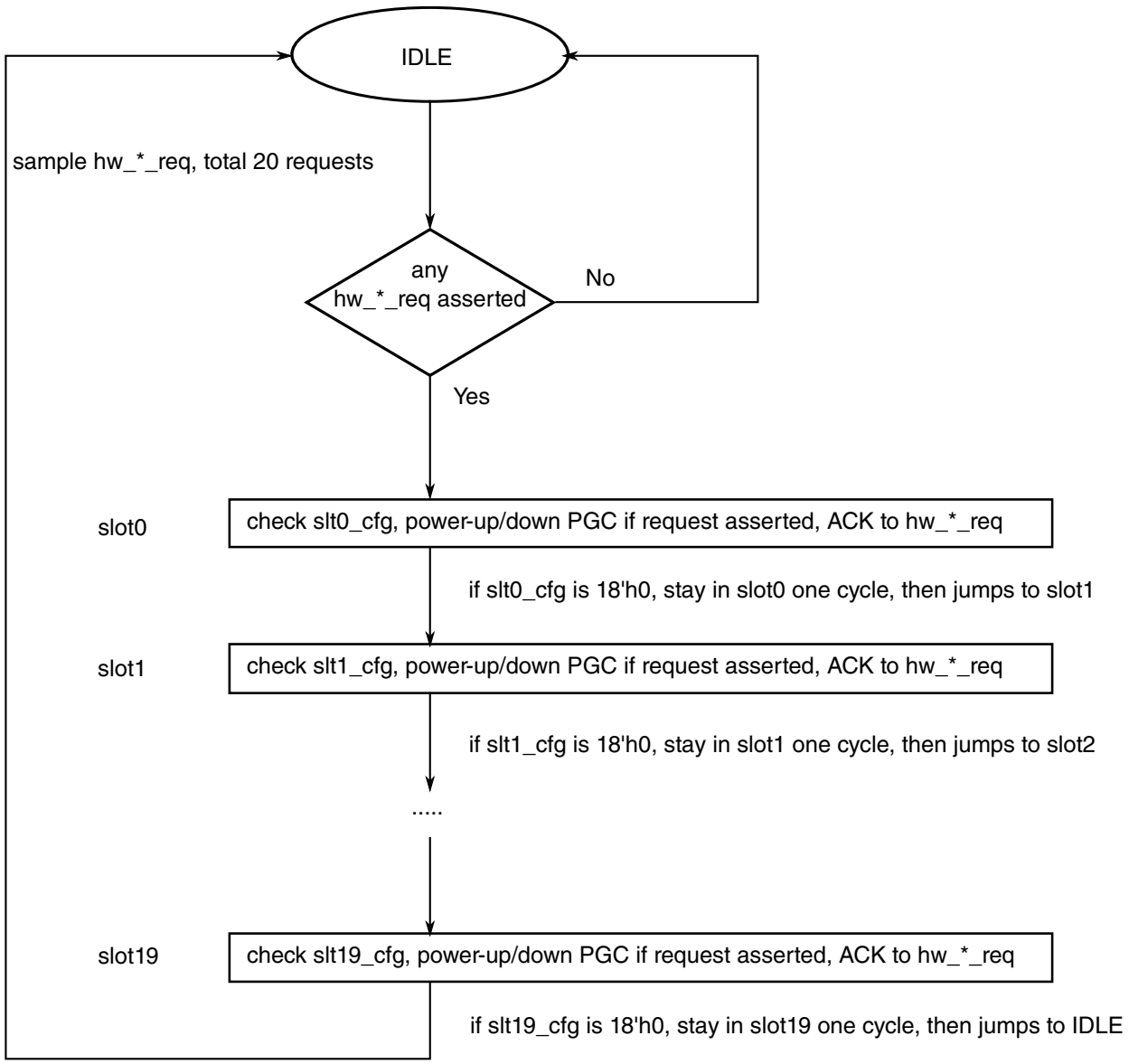


Figure 5-13. Slot controller processing flow

NOTE

PGC_SCU should be “always-on” to PGC_C0 PGC_C1, PGC_C2, and PGC_C3. This means PGC_SCU should be power up earlier than PGC_C0/PGC_C1/PGC_C2/PGC_C3 and should be power down later than PGC_C0/PGC_C1/ PGC_C2/PGC_C3 (see example code 1 and 2). If we arrange A53 Cx/A53 SCU power down/up in same slot, special setting is required (see example code 2).

NOTE

When the system enters/exists ALL_OFF or L2_RETENTION mode, PGC_MF should be power up earlier than PGC_C0/ PGC_C1/PGC_C2/PGC_C3/PGC_SCU. We can arrange MIX PGC power up in earlier slot than A53 Cx/SCU power up slot (See example code 1 and 2).

5.2.5.4 Handshake between LPM controller and time slot controller

The figure below shows an example of A53 “into” LPM sequence. We want to power up fastmix/megamix,A53 SCU and A53 core0 in time slot0 /slot1/slot2 respectively. Request from low power mode controller will be mapped to relevant PGCs according to the rules listed above. We choose acknowledge from PGC_core0 as the acknowledge for A53 LPM power down request (relevant register bits are defined in “PGC_ACK_SEL_A53”). The A53 LPM will regard the three PGCs as a virtual big PGC and it will cancel all power up request when it receive the acknowledge signals from PGC_core0.

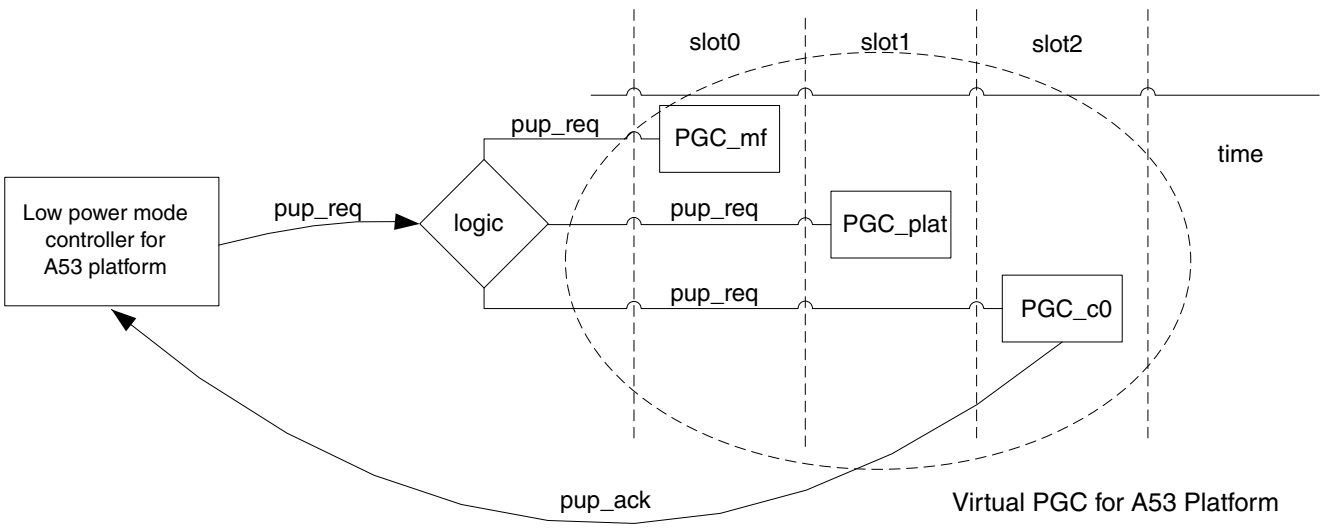


Figure 5-14. A53 into LPM sequence

NOTE

If a PGC is mapped to two CPU domain (refer to “Domain control for PGCs ”for more information), it cannot be selected as the power down acknowledge for both of the CPU platform. “PGC_ACK_SEL_A53”, "PGC_ACK_SEL_A53_PU", and “PGC_ACK_SEL_M4” are should be chosen for the last PGC in power up or power down sequence in the time slot. If there is no PGC be power up/power down with LPM sequence, the

"dummy" acknowledge should be selected. Only one PGC should be selected for power down or power up acknowledge for one CPU platform.

5.2.6 Power control for A53 Platform

5.2.6.1 A53 Platform power domains and power modes

There are six power domains inside SEC dual core Cortex A53 platform: Core0, Core1, Core2, Core3, SCU, and L2 RAM. There are six power states in A53 platform:

Power State	PDCPU0	PDCPU1	PDCPU2	PDCPU3	PDPLAT	PDL2	VDD_ARM
ALL_ON	ON	ON	ON	ON	ON	ON	ON
THREE_CPU_ON	3 CPUs are ON, 1 CPU is OFF				ON	ON	ON
TWO_CPU_ON	2 CPUs are ON, 2 CPUs are OFF				ON	ON	ON
ONE_CPU_ON	1 CPU is ON, 3 CPUs are OFF				ON	ON	ON
ALL_CPU_OFF	OFF	OFF	OFF	OFF	ON	ON	ON
L2_RETENTION	OFF	OFF	OFF	OFF	OFF	RET	ON
ALL_OFF	OFF	OFF	OFF	OFF	OFF	OFF	ON
POWER_DOWN	OFF	OFF	OFF	OFF	OFF	OFF	OFF

NOTE

In all six power states, "ALL_ON", "THREE_CPU_ON", "TWO_CPU_ON", and "ONE_CPU_ON" can exist in all RUN, WAIT or STOP mode of A53 platform. "L2_RETENTION" and "ALL_OFF" can only exist in WAIT or STOP mode of A53 platform.

5.2.6.2 Power down process for the A53 Platform

5.2.6.2.1 Power down of Core0, Core1, Core2, and Core3 in the A53 Platform

The power of core0, core1, core2, and core3 can be shut off along with the LPM process, as show in the following figure:

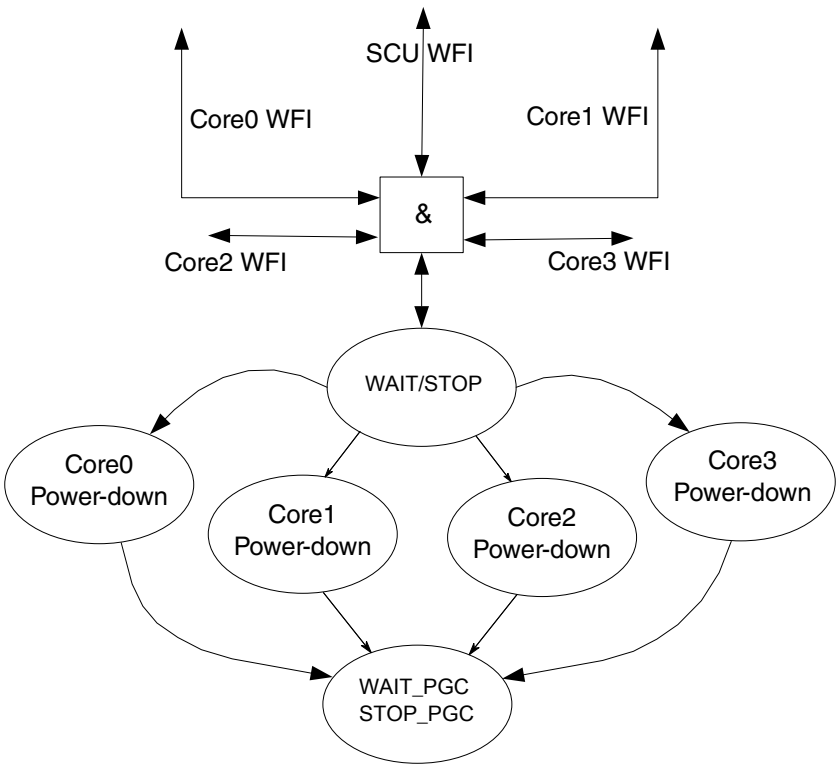


Figure 5-15. Power down of Core0, Core1, Core2, and Core3

WFIs from A53 platform will trigger the A53 platform LPM and the power of core0, core1, core2, or core3 will be shut off when “lpcr_a53_ad.en_c0_pdn”, “lpcr_a53_ad.en_c1_pdn”, “lpcr_a53_ad.en_c2_pdn”, or “lpcr_a53_ad.en_c3_pdn” enabled in this process. This mode should be used when core0 is used as the leading core of A53 platform.

The power of core0, core1, core2, and core3 can also be shut off in RUN mode: in this mode “LPCR_A53_AD.en_c0_wfi_pdn”, “LPCR_A53_AD.en_c1_wfi_pdn”, “LPCR_A53_AD.en_c2_wfi_pdn”, and “LPCR_A53_AD.en_c3_wfi_pdn” should be set and the condition to trigger A53 LPM will be some different:

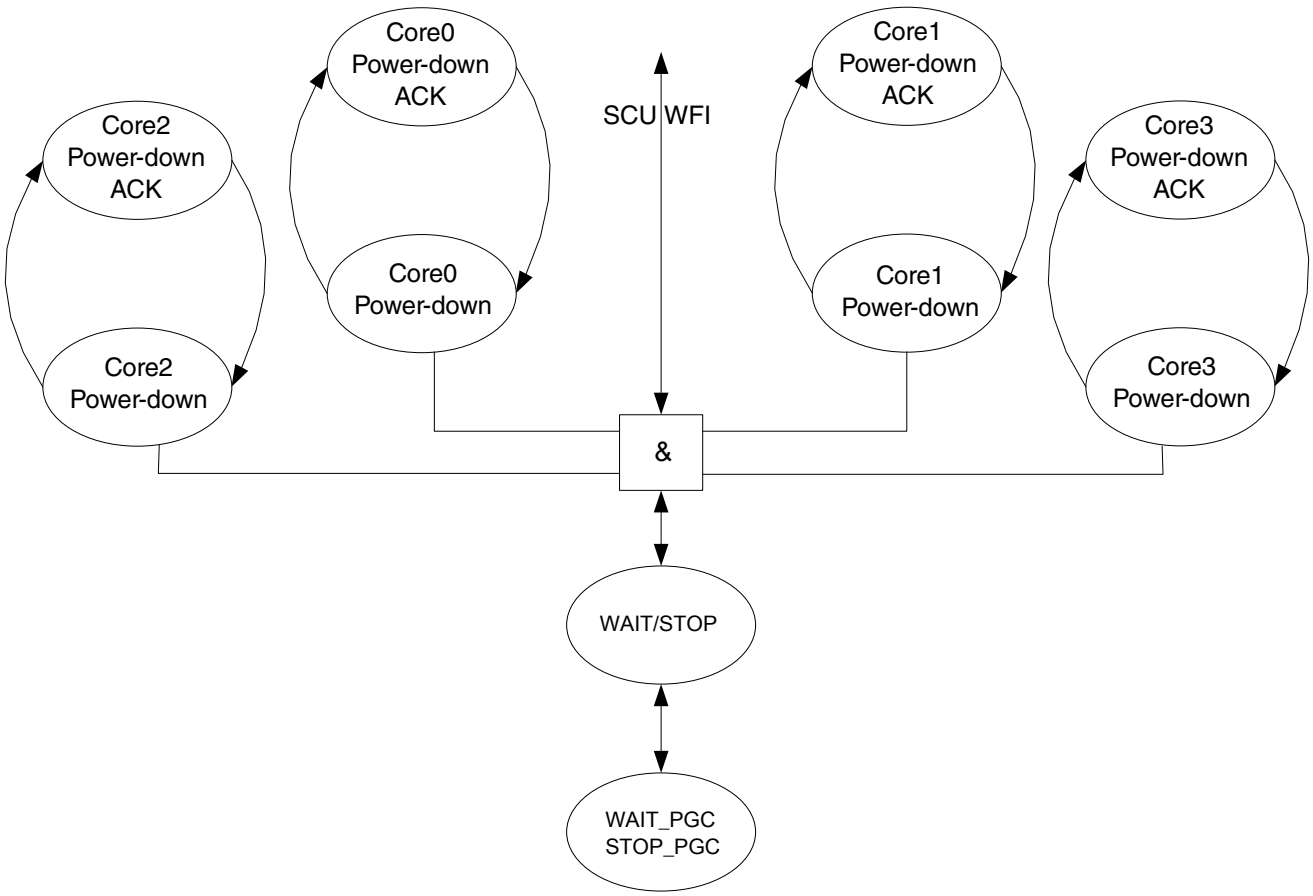


Figure 5-16. Power down of Core0, Core1, Core2, and Core3 in RUN mode

In this mode, core0/core1/core2/core3 power down process should not be disturbed by IRQs.

5.2.6.2.2 Power down of SCU and L2 Cache RAM

Power domain SCU and L2 cache RAM is controlled by one PGC – "PGC_PLAT" and they can only be power down in LPM process (when "LPCR_A53_AD.en_plat_pdn" is set). There is another bit "LPCR_A53_AD.l2pge" which will decide if L2 cache RAM need to be in retention mode when SCU domain is power down.

5.2.6.3 Power up process for the A53 Platform

- The core0, core1, core2, and core3 can be powered up with the exit of A53 LPM. The relevant bits are "LPCR_A53_AD.en_c0_pup", "LPCR_A53_AD.en_c1_pup", "LPCR_A53_AD.en_c2_pup", and "LPCR_A53_AD.en_c3_pup".
- The core0, core1, core2, and core3 can also be powered up by interrupt signal in RUN mode.

- The relevant bit are “LPCR_A53_AD.en_c0_irq_pup”, “LPCR_A53_AD.en_c1_irq_pup”, “LPCR_A53_AD.en_c2_irq_pup”, and “LPCR_A53_AD.en_c3_irq_pup”.
- The interrupt signal can be chosen from GIC or directly from IRQ with mask in GPCv2.
- The relevant select bits are “LPCR_A53_BSC.irq_src_c0”, “LPCR_A53_BSC.irq_src_c1”, “LPCR_A53_BSC.irq_src_c2”, “LPCR_A53_BSC.irq_src_c3”, and “LPCR_A53_BSC.irq_src_a53_wup”. (LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC.)

{LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC[29:28]}	Usage	Restriction
5'b00000	Use IRQ trigger A53 LPM and use IRQ to power up core0 to core3	None
5'b01111	Use GIC trigger A53 LPM and GIC to power up core0 to core3	SCU cannot power down in LPM, CPU clock cannot stop in LPM
5'b11111	Use IRQ trigger A53 LPM and GIC to power up core0 to core3	SCU cannot power down in LPM

As show in the table above, core0/core1/core2/core3 can only be power up by its own interrupt in RUN mode of A53 platform.

There are three combination of {LPCR_A53_BSC[30],LPCR_A53_BSC[23:22],LPCR_A53_BSC[29:28]}:

1. In the first case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53“ are used to separate the 128 bits interrupts for core0, core1, core2, and core3 of A53 platform and also used as the interrupt mask for A53 LPM.
2. In the second case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53” are not used, GIC setting are used to separate interrupts for core0, core1, core2, and core3 of A53 platform and also used as the interrupt mask for A53 LPM.
3. In the third case, “IMRn_CORE0_A53, IMRn_CORE1_A53, IMRn_CORE2_A53, IMRn_CORE3_A53” is used as the mask for interrupt for A53 LPM, GIC setting are used to separate interrupts for core0, core1, core2, and core3.

5.2.7 Power control for the M4 Platform

M4 LPM is same as A53 LPM. M4 platform doesn’t have its own power domain and exist as a part of Megamix. M4 LPM power down request cannot power down the fastmix/megamix directly. There is a virtual PGC reserved for M4, the virtual PGC will

do nothing else except generating an acknowledge signal and this signal can be chosen as the acknowledge signal for M4 platform. Make sure virtual PGC power up/down slot same with fastmix/megamix PGC power up/down slot (See example code 3).

5.2.8 Domain control for PGCs

The following rules are used for PGC power up/down with domain mapping control:

1. For PGCs inside CPU platform (since M4 doesn't have its own power domain, only PGCs in A53 platform are referred): for hardware trigger (including both power up and power down) only the LPM request from its own platform will take effect; for software trigger (including both power up and power down) (the relevant register bit are "CPU_PGC_SW_PUP_REQ" and "CPU_PGC_SW_PDN_REQ"), only the software running in its own platform will take effect.
2. For PGCs outside CPU platform(MIX and PU PGCs), register bits "PGC_CPU_MAPPING" will map MIX and PU PGCs to A53 or M4 CPU domain:
   - One PGC can be mapped to one or both of A53 and M4 domain;
   - For hardware power up request, if a PGC is mapped to any CPU domain, the PGC will be powered up when the corresponding CPU platform sends out its power up request.
   - For software power up, if a PGC is mapped any CPU platform, the PGC can be powered up by software running in corresponding CPU.(the relevant register bits are "MIX_PGC_SW_PUP_REQ" and "PU_PGC_SW_PUP_REQ")
   - For hardware power down, if a power domain is mapped to only one CPU domain, the relevant PGC can be powered down when the corresponding CPU platform sends out its power down request; If a power domain is mapped to both two CPU domain, when one CPU platform sends out its hardware power down request, the relevant PGC will power down with any of the following condition satisfied: the other CPU already in LPM; the other CPU want to power down relevant PGC (the relevant register are "A53_MIX_PDN_FLG, M4_MIX_PDN_FLG, A53_PU_PDN_FLG, M4_PU_PDN_FLG")
   - For software power down, if a power domain is mapped to only one CPU domain, the PGC can be powered down by software running in corresponding CPU. (the relevant register bits are "MIX_PGC_SW_PUP_REQ" and "PU_PGC_SW_PUP_REQ"). If a power domain is mapped to both two CPU domain, when one CPU platform sends out its software power down request, the relevant PGC will power down with any of the following condition satisfied: the other CPU already in LPM; the other CPU want to power down relevant PGC (the relevant register are "A53_MIX_PDN_FLG, M4_MIX_PDN_FLG, A53_PU_PDN_FLG, M4_PU_PDN_FLG")
   - The access of "PGC_CPU_MAPPING" is controlled by domain information from RDC

5.2.9 Example Code

Below are code examples for entering specific power scenarios

5.2.9.1 Example Code 1

```
        //ARM enters into ALL_OFF(STOP) mode and enable DSM :
        //after "wfi", MIX/C0/C1/C2/C3 power down in SLOT0, SCU power down in SLOT1 when
        //after "GPT1_INT" arrived, MIX power up in SLOT2, SCU power up in SLOT3, C0 power
up in SLOT4
        //IMRx_CORE0_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF); //[23] : GPT1 used as wakeup
source
        reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
        //IMRx_CORE1_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
        //IMRx_CORE2_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
        //IMRx_CORE3_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
        //LPCR_A53_BSC
        reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
        //[30],[23:22],[29:28] : A53_CO/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
        //[14] : CLOCK OFF during STOP mode
        //[3:0] : STOP mode
        //LPCR_A53_BSC2
        reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
        //[3:0] : STOP mode
        //LPCR_A53_AD
        reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0B0A1A) ;
        //[16] : 1(ALL_OFF mode); 0(L2 retention mode)
        //[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
        //[4] : A53_SCU power down with A53 LPM PDN REQ
        //[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
        //LPCR_M4
        reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x80000000) ;
        //[31] : DSM ingore to check M4 low power state
        //[1:0] : M4 LPM run mode
        //SLPCR
        reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
        //[31] : enable DSM
        //[30] : eneable regulator bypass
        //[5:3] : wait 64 ckil clock cycles
        //[2] : enable PMIC standby
        //[1] : enable OSC power down
        //[0] : bypass PMIC ready handshake
        //PGC_ACK_SEL_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
        //[2] : A53_SCU PGC as LPM power down ack
        //[16] : A53_C0 PGC as LPM power up ack
```

```
        //SLT_CFG0
        reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000055) ;
        //[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
        //SLT_CFG0_PU
        reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0000001) ;
        //[0] : fastmix/megamix power down in SLOT0
        //SLT_CFG1
        reg32_write(GPC_IPS_BASE_ADDR + 0xB4,0x00000100) ;
        //[8] : A53_SCU power down in SLOT1
        //SLT_CFG2_PU
        reg32_write(GPC_IPS_BASE_ADDR + 0x208,0x00000002) ;
        //[1] : fastmix/megamix power up in SLOT2
        //SLT_CFG3
        reg32_write(GPC_IPS_BASE_ADDR + 0xBC,0x00000200) ;
        //[9] : A53_SCU power up in SLOT3
        //SLT_CFG4
        reg32_write(GPC_IPS_BASE_ADDR + 0xC0,0x00000002) ;
        //[1] : A53_C0 power up in SLOT4, A53_C1/ A53_C2/A53_C3 not power up
        //PGC_CPU_MAPPING
        reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00000001) ;
        //[0] : MIX PGC mapping to A53 LPM
        //A53 PGC
        reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) |
0x00000001) ;
        // enable A53_C0 PGC power down
        reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) |
0x00000001) ;
        // enable A53_C1 PGC power down
        reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x880) |
0x00000001) ;
        // enable A53_C2 PGC power down
        reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x8C0) |
0x00000001) ;
        // enable A53_C3 PGC power down
        reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x900) |
0x00000001) ;
        // enable A53_SCU PGC power down
        reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 << 10) | 0x5B | (0x51 << 20) ) ;
        // change nL2retn/mempwr/dftram to meet SCU power up timing
        //fastmix/megamix PGC
        reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) |
0x00000001) ;
        // enable MIX PGC power down
```

5.2.9.2 Example Code 2

```
        //ARM enters into L2_RETENTION(STOP) mode and enable DSM :
        //after "wfi", MIX/C0/C1/C2/C3/SCU power down in SLOT0
        //after "GPT1_INT" arrived, MIX power up in SLOT1, SCU/C0 power up in SLOT2
        //IMRx_CORE0_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF);
        //[23] : GPT1 used as wakeup source
        reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
        //IMRx_CORE1_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
        //IMRx_CORE2_A53
        reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
        reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
```

```
            reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
            //IMRx_CORE3_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
            //LPCR_A53_BSC
            reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
            //[30],[23:22],[29:28] : A53_C0/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
            //[14] : CLOCK OFF during STOP mode
            //[3:0] : STOP mode
            //LPCR_A53_BSC2
            reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
            //[3:0] : STOP mode
            //LPCR_A53_AD
            reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0A0A1A) ;
            //[16] : 1(ALL_OFF mode); 0(L2 retention mode)
            //[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
            //[4] : A53_SCU power down with A53 LPM PDN REQ
            //[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
            //LPCR_M4
            reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x80000000) ;
            //[31] : DSM ingore to check M4 low power state
            //[1:0] : M4 LPM run mode
            //SLPCR
            reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
            //[31] : enable DSM
            //[30] : eneable regulator bypass
            //[5:3] : wait 64 ckil clock cycles
            //[2] : enable PMIC standby
            //[1] : enable OSC power down
            //[0] : bypass PMIC ready handshake
            //PGC_ACK_SEL_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
            //[2] : A53_SCU PGC as LPM power down ack
            //[16] : A53_C0 PGC as LPM power up ack
            //SLT_CFG0
            reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000155) ;
            //[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
            //[8] : A53_SCU power down in SLOT0
            //SLT_CFG0_PU
            reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0000001) ;
            //[0] : fastmix/megamix power down in SLOT0
            //A53_Cx/SCU are power down in same slot. Special setting is required( see below
PGC setting #A )
            //SLT_CFG1_PU
            reg32_write(GPC_IPS_BASE_ADDR + 0x204,0x00000002) ;
            //[1] : fastmix/megamix power up in SLOT1
            //SLT_CFG2
            reg32_write(GPC_IPS_BASE_ADDR + 0xB8,0x00000202) ;
            //[9] : A53_SCU power up in SLOT1
            //[1] : A53_C0 power up in SLOT1, A53_C1/ A53_C2/ A53_C3 not power up
            //A53_Cx/SCU are power up in same slot. Special setting is required( see below
PGC setting #B )
            //PGC_CPU_MAPPING
            reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00000001) ;
            //[0] : MIX PGC mapping to A53 LPM
            //Special PGC setting #A for C0/C1/C2/C3/SCU power down in same slot(SCU should
be always ON comparing to C0/C1/C2/C3)
            reg32_write(GPC_IPS_BASE_ADDR +0x808, (reg32_read(GPC_IPS_BASE_ADDR +0x808) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C0.ISO2SW = 8 ; C0.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C1.ISO2SW = 8 ; C1.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x888, (reg32_read(GPC_IPS_BASE_ADDR +0x888) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C2.ISO2SW = 8 ; C2.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x8C8, (reg32_read(GPC_IPS_BASE_ADDR +0x8C8) &
```

```
0xFFFFC0C0) | 0x0801 ) ;
            // set C3.ISO2SW = 8 ; C3.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x908, (reg32_read(GPC_IPS_BASE_ADDR +0x908) &
0xFFFFC0C0) | 0x1001 ) ;
            // set SCU.ISO2SW = 16 ; SCU.ISO = 1;
            //Special PGC setting #B for A53_Cx/SCU power up in same slot(SCU should be
always ON comparing to C0/C1/C2/C3)
            reg32_write(GPC_IPS_BASE_ADDR +0x804, (reg32_read(GPC_IPS_BASE_ADDR +0x804) &
0xFF800040) | 0x11 | (0x20 << 7) ) ;
            // set C0.SW = 0x11 ; C0.SW2ISO = 0x20 ;
            reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) &
0xFF800040) | 0x11 | (0x20 << 7) ) ;
            // set C1.SW = 0x11 ; C1.SW2ISO = 0x20 ;
            reg32_write(GPC_IPS_BASE_ADDR +0x884, (reg32_read(GPC_IPS_BASE_ADDR +0x884) &
0xFF800040) | 0x11 | (0x20 << 7) ) ;
            // set C2.SW = 0x11 ; C2.SW2ISO = 0x20 ;
            reg32_write(GPC_IPS_BASE_ADDR +0x8C4, (reg32_read(GPC_IPS_BASE_ADDR +0x8C4) &
0xFF800040) | 0x11 | (0x20 << 7) ) ;
            // set C3.SW = 0x11 ; C3.SW2ISO = 0x20 ;
            reg32_write(GPC_IPS_BASE_ADDR +0x904, (reg32_read(GPC_IPS_BASE_ADDR +0x904) &
0xFF800040) | 0x1 | (0x0f << 7) ) ;
            // set SCU.SW = 0x1 ; SCU.SW2ISO = 0x0f ;
            //A53 PGC
            reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) |
0x00000001) ;
            // enable A53_C0 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) |
0x00000001) ;
            // enable A53_C1 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x880) |
0x00000001) ;
            // enable A53_C2 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x8C0, reg32_read(GPC_IPS_BASE_ADDR +0x8C0) |
0x00000001) ;
            // enable A53_C3 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x900, reg32_read(GPC_IPS_BASE_ADDR +0x900) |
0x00000001) ;
            // enable A53_SCU PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 << 10) | 0x5B | (0x51 << 20) ) ;
            // change nL2retn/mempwr/dftram to meet SCU power up timing
            //fastmix/megamix PGC
            reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) |
0x00000001) ;
            // enable MIX PGC power down
```

5.2.9.3 Example Code 3

```
            //A53/M4 both enters into low power mode. A53/M4 are in different master domain.
            //MIX are mapping to both A53 and M4
            //after A53/M4 enters into low power mode, MIX will be also power down.A53/M4
enters into low power mode any time
            //either A53 or M4 exists from low power mode, MIX will be also power up
            //IMRx_CORE0_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x30, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x34, 0xFF7FFFFF);
            //[23] : GPT1 used as ARM wakeup source
            reg32_write(GPC_IPS_BASE_ADDR + 0x38, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x3C, 0xFFFFFFFF);
            //IMRx_CORE1_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x40, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x44, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x48, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x4C, 0xFFFFFFFF);
```

```
            //IMRx_CORE2_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x1C0, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1C4, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1C8, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1CC, 0xFFFFFFFF);
            //IMRx_CORE3_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D0, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D4, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1D8, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x1DC, 0xFFFFFFFF);
            //IMRx_M4
            reg32_write(GPC_IPS_BASE_ADDR + 0x50, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x54, 0xFFBFFFFF);
            //[22] : GPT2 used as M4 wakeup source
            reg32_write(GPC_IPS_BASE_ADDR + 0x58, 0xFFFFFFFF);
            reg32_write(GPC_IPS_BASE_ADDR + 0x5C, 0xFFFFFFFF);
            //LPCR_A53_BSC
            reg32_write(GPC_IPS_BASE_ADDR, 0x0000000A) ;
            //[30],[23:22],[29:28] : A53_C0/A53_C1/A53_C2/A53_C3/LPM wakeup from external INT
            //[14] : CLOCK OFF during STOP mode
            //[3:0] : STOP mode
            //LPCR_A53_BSC2
            reg32_write(GPC_IPS_BASE_ADDR + 0x108, 0x0000000A) ;
            //[3:0] : STOP mode
            //LPCR_A53_AD
            reg32_write(GPC_IPS_BASE_ADDR + 0x4, 0x0A0B0A1A) ;
            //[16] : 1(ALL_OFF mode); 0(L2 retention mode)
            //[27]/[25]/[11]/[9]: A53_C0/A53_C1/A53_C2/A53_C3 power up with A53 LPM PUP REQ
            //[4] : A53_SCU power down with A53 LPM PDN REQ
            //[19]/[17]/[3]/[1] : A53_C0/A53_C1/A53_C2/A53_C3 powr down with A53 LPM PDN REQ
            //LPCR_M4
            reg32_write(GPC_IPS_BASE_ADDR + 0x8, 0x00003FFE) ;
            //[31] : 0(check M4 low power state to enter into DSM)
            //[14] : 0(M4 clock OFF during low power mode)
            //[3:2] : enable M4 virtual PGC power up/down with M4 LPM
            //[1:0] : M4 LPM STOP mode
            //SLPCR
            reg32_write(GPC_IPS_BASE_ADDR + 0x14,0xe000ffA7) ;
            //[31] : enable DSM
            //[30] : eneable regulator bypass
            //[5:3] : wait 64 ckil clock cycles
            //[2] : enable PMIC standby
            //[1] : enable OSC power down
            //[0] : bypass PMIC ready handshake
            //PGC_ACK_SEL_A53
            reg32_write(GPC_IPS_BASE_ADDR + 0x24,0x00010004) ;
            //[2] : A53_SCU PGC as LPM power down ack
            //[16] : A53_C0 PGC as LPM power up ack
            //PGC_ACK_SEL_M4
            reg32_write(GPC_IPS_BASE_ADDR + 0x28,0x00010001) ;
            //[0] : M4 virtual PGC as M4 LPM power down ack
            //[16] : M4 virtual PGC as M4 LPM power up ack
            //GPC_MISC
            reg32_write(GPC_IPS_BASE_ADDR + 0x2C,reg32_read(GPC_IPS_BASE_ADDR + 0x2C) |
0x100) ;
            //[8] : not mask M4 power down request to M4 virtual PGC
            //SLT_CFG0
            reg32_write(GPC_IPS_BASE_ADDR + 0xB0,0x00000155) ;
            //[6]/[4]/[2]/[0] : A53_C0/A53_C1/ A53_C2/A53_C3 power down in SLOT0
            //[8] : A53_SCU power down in SLOT0
            //SLT_CFG0_PU
            reg32_write(GPC_IPS_BASE_ADDR + 0x200,0x0001001) ;
            //[0] : fastmix/megamix power down in SLOT0
            //[12] : M4 virtual PGC power down in SLOT0, same with fastmix/megamix PGC power
down slot
            //A53_Cx/SCU are power down in same slot. Special setting is required( see below
PGC setting #A )
            //SLT_CFG1_PU
            reg32_write(GPC_IPS_BASE_ADDR + 0x204,0x00002002) ;
            //[2] : fastmix/megamix power up in SLOT1
```

```
            //[13] : M4 virtual PGC power up in SLOT1, same with fastmix/megamix PGC power
up slot
            //SLT_CFG2
            reg32_write(GPC_IPS_BASE_ADDR + 0xB8,0x00000202) ;
            //[9] : A53_SCU power up in SLOT1
            //[1] : A53_C0 power up in SLOT1, A53_C1/ A53_C2/ A53_C3 not power up
            //A53_Cx/SCU are power up in same slot. Special setting is required( see below
PGC setting #B )
            //PGC_CPU_MAPPING
            reg32_write(GPC_IPS_BASE_ADDR + 0xEC,0x00010001) ;
            //[0] : MIX PGC mapping to A53 LPM
            //[16] : MIX PGC mapping to M4 LPM
            //Special PGC setting #A for C0/C1/SCU power down in same slot(SCU should be
always ON comparing to C0/C1/C2/C3)
            reg32_write(GPC_IPS_BASE_ADDR +0x808, (reg32_read(GPC_IPS_BASE_ADDR +0x808) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C0.ISO2SW = 8 ; C0.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C1.ISO2SW = 8 ; C1.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x848, (reg32_read(GPC_IPS_BASE_ADDR +0x848) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C2.ISO2SW = 8 ; C2.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x888, (reg32_read(GPC_IPS_BASE_ADDR +0x888) &
0xFFFFC0C0) | 0x0801 ) ;
            // set C3.ISO2SW = 8 ; C3.ISO = 1;
            reg32_write(GPC_IPS_BASE_ADDR +0x908, (reg32_read(GPC_IPS_BASE_ADDR +0x908) &
0xFFFFC0C0) | 0x1001 ) ;
            // set SCU.ISO2SW = 16 ; SCU.ISO = 1;
            //Special PGC setting #B for A53_Cx/SCU power up in same slot(SCU should be
always ON comparing to C0/C1/C2/C3)
            reg32_write(GPC_IPS_BASE_ADDR +0x804, (reg32_read(GPC_IPS_BASE_ADDR +0x804) &
0xFF800040) | 0x10 | (0x1f << 7) ) ;
            // set C0.SW = 0x10 ; C0.SW2ISO = 0x1f ;
            reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) &
0xFF800040) | 0x10 | (0x1f << 7) ) ;
            // set C1.SW = 0x10 ; C1.SW2ISO = 0x1f ;
            reg32_write(GPC_IPS_BASE_ADDR +0x844, (reg32_read(GPC_IPS_BASE_ADDR +0x844) &
0xFF800040) | 0x10 | (0x1f << 7) ) ;
            // set C2.SW = 0x10 ; C2.SW2ISO = 0x1f ;
            reg32_write(GPC_IPS_BASE_ADDR +0x884, (reg32_read(GPC_IPS_BASE_ADDR +0x884) &
0xFF800040) | 0x10 | (0x1f << 7) ) ;
            // set C3.SW = 0x10 ; C3.SW2ISO = 0x1f ;
            reg32_write(GPC_IPS_BASE_ADDR +0x904, (reg32_read(GPC_IPS_BASE_ADDR +0x904) &
0xFF800040) | 0x1 | (0x0f << 7) ) ;
            // set SCU.SW = 0x1 ; SCU.SW2ISO = 0x0f ;
            //A53 PGC
            reg32_write(GPC_IPS_BASE_ADDR +0x800, reg32_read(GPC_IPS_BASE_ADDR +0x800) |
0x00000001) ;
            // enable A53_C0 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) |
0x00000001) ;
            // enable A53_C1 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x840, reg32_read(GPC_IPS_BASE_ADDR +0x840) |
0x00000001) ;
            // enable A53_C2 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x880, reg32_read(GPC_IPS_BASE_ADDR +0x880) |
0x00000001) ;
            // enable A53_C3 PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x900, reg32_read(GPC_IPS_BASE_ADDR +0x900) |
0x00000001) ;
            // enable A53_SCU PGC power down
            reg32_write(GPC_IPS_BASE_ADDR +0x910, (0x59 << 10) | 0x5B | (0x51 << 20) ) ;
            // change nL2retn/mempwr/dftram to meet SCU power up timing
            //fastmix/megamix PGC
            reg32_write(GPC_IPS_BASE_ADDR +0xA00, reg32_read(GPC_IPS_BASE_ADDR +0xA00) |
0x00000001) ;
            // enable MIX PGC power down
```

5.2.9.4 Example Code 4

```
            // software power up/down PCIE PHY
            //power up PCIE PHY
            reg32_write ( GPC_IPS_BASE_ADDR + 0xEC, reg32_read(GPC_IPS_BASE_ADDR + 0x0EC) |
0x8 );
            //map PCIE PGC to A53
            reg32_write ( GPC_IPS_BASE_ADDR + 0xF8 , reg32_read(GPC_IPS_BASE_ADDR + 0xF8) |
0x2 );
            //trigger sw Power up Request
            while( read(GPC_IPS_BASE_ADDR + 0xF8) & 0x2 );
            //wait software power up request self clear
            //power down PCIE PHY
            reg32_write ( GPC_IPS_BASE_ADDR + 0xEC, reg32_read(GPC_IPS_BASE_ADDR + 0x0EC) |
0x8 );
            //map PCIE PGC to A53
            reg32_write ( GPC_IPS_BASE_ADDR + 0xC40, reg32_read(GPC_IPS_BASE_ADDR + 0xC40) |
0x1 );
            // enable PCIE PGC power down
            reg32_write ( GPC_IPS_BASE_ADDR + 0x104 , reg32_read(GPC_IPS_BASE_ADDR + 0x104)
| 0x2 );
            //trigger sw Power Down Request
            while( read(GPC_IPS_BASE_ADDR + 0x104) & 0x2 );
            //wait software power down request self clear
```

5.2.10 GPC Memory Map/Register Definition

Detailed descriptions of each register can be found below.

The total GPC memory map is 4KB

Table 5-10. Memory Regions

Address Range(offset)	Region
0x000 - 0x3FF	GPC configuration register
0x400 - 0x7FF	Reserved
0x800 - 0x9FF	CPU and SCU type PGC register base address
0xA00 - 0xBFF	MIX type PGC register base address
0xC00 - 0xFFF	PU type PGC register base address

Each PGC (CPU type, MIX type, PU type) will occupy 64 Bytes address space, the specific base address of each PGC are listed as below.

- 0x800 ~ 0x83F : PGC for A53 core0
- 0x840 ~ 0x87F: PGC for A53 core1
- 0x880 ~ 0x8BF: PGC for A53 core2

- 0x8C0 ~ 0x8FF: PGC for A53 core3
- 0x900 ~ 0x93F: PGC for A53 SCU
- 0xA00 ~ 0xA3F: PGC for fastmix/megamix
- 0xC00 ~ 0xC3F: PGC for MIPI DSI PHY
- 0xC40 ~ 0xC7F: PGC for PCIE1 PHY
- 0xC80 ~ 0xCBF: USB_OTG1
- 0xCC0 ~ 0xCFF: USB_OTG2
- 0xD00 ~ 0xD3F: Reserved
- 0xD40 ~ 0xD7F: DDR1
- 0xD80 ~ 0xDBF: Reserved
- 0xDC0 ~ 0xDFF: GPUMIX
- 0xE00 ~ 0xE3F: VPUMIX
- 0xE40 ~ 0xE7F: HDMI
- 0xE80 ~ 0xEBF: DISPMIX
- 0xEC0 ~ 0xEFF: MIPI CSI1 PHY
- 0xF00 ~ 0xF3F: MIPI CSI2 PHY
- 0xF40 ~ 0xF7F: PCIE2 PHY

For more specific information about PGC register definition, please see the register definition for each PGC.

GPC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0000	Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC)	32	R/W	0000_3FF0h	5.2.10.1/ 648
303A_0004	Advanced Low power control register of A53 platform (GPC_LPCR_A53_AD)	32	R/W	0000_0020h	5.2.10.2/ 651
303A_0008	Low power control register of CPU1 (GPC_LPCR_M4)	32	R/W	0000_3FF0h	5.2.10.3/ 653
303A_0014	System low power control register (GPC_SLPCR)	32	R/W	E000_FF82h	5.2.10.4/ 655
303A_0018	MASTER LPM Handshake (GPC_MST_CPU_MAPPING)	32	R/W	0000_00FFh	5.2.10.5/ 657
303A_0020	Memory low power control register (GPC_MLPCR)	32	R/W	0101_0100h	5.2.10.6/ 658
303A_0024	PGC acknowledge signal selection of A53 platform (GPC_PGC_ACK_SEL_A53)	32	R/W	8000_8000h	5.2.10.7/ 659
303A_0028	PGC acknowledge signal selection of M4 platform (GPC_PGC_ACK_SEL_M4)	32	R/W	8000_8000h	5.2.10.8/ 661
303A_002C	GPC Miscellaneous register (GPC_MISC)	32	R/W	0000_0021h	5.2.10.9/ 662
303A_0030	IRQ masking register 1 of A53 core0 (GPC_IMR1_CORE0_A53)	32	R/W	0000_0000h	5.2.10.10/ 663

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0034	IRQ masking register 2 of A53 core0 (GPC_IMR2_CORE0_A53)	32	R/W	0000_0000h	5.2.10.11/ 664
303A_0038	IRQ masking register 3 of A53 core0 (GPC_IMR3_CORE0_A53)	32	R/W	0000_0000h	5.2.10.12/ 664
303A_003C	IRQ masking register 4 of A53 core0 (GPC_IMR4_CORE0_A53)	32	R/W	0000_0000h	5.2.10.13/ 665
303A_0040	IRQ masking register 1 of A53 core1 (GPC_IMR1_CORE1_A53)	32	R/W	0000_0000h	5.2.10.14/ 665
303A_0044	IRQ masking register 2 of A53 core1 (GPC_IMR2_CORE1_A53)	32	R/W	0000_0000h	5.2.10.15/ 665
303A_0048	IRQ masking register 3 of A53 core1 (GPC_IMR3_CORE1_A53)	32	R/W	0000_0000h	5.2.10.16/ 666
303A_004C	IRQ masking register 4 of A53 core1 (GPC_IMR4_CORE1_A53)	32	R/W	0000_0000h	5.2.10.17/ 666
303A_0050	IRQ masking register 1 of M4 (GPC_IMR1_M4)	32	R/W	0000_0000h	5.2.10.18/ 667
303A_0054	IRQ masking register 2 of M4 (GPC_IMR2_M4)	32	R/W	0000_0000h	5.2.10.19/ 667
303A_0058	IRQ masking register 3 of M4 (GPC_IMR3_M4)	32	R/W	0000_0000h	5.2.10.20/ 667
303A_005C	IRQ masking register 4 of M4 (GPC_IMR4_M4)	32	R/W	0000_0000h	5.2.10.21/ 668
303A_0070	IRQ status register 1 of A53 (GPC_ISR1_A53)	32	R	0000_0000h	5.2.10.22/ 668
303A_0074	IRQ status register 2 of A53 (GPC_ISR2_A53)	32	R	0000_0000h	5.2.10.23/ 669
303A_0078	IRQ status register 3 of A53 (GPC_ISR3_A53)	32	R	0000_0000h	5.2.10.24/ 669
303A_007C	IRQ status register 4 of A53 (GPC_ISR4_A53)	32	R	0000_0000h	5.2.10.25/ 669
303A_0080	IRQ status register 1 of M4 (GPC_ISR1_M4)	32	R	0000_0000h	5.2.10.26/ 670
303A_0084	IRQ status register 2 of M4 (GPC_ISR2_M4)	32	R	0000_0000h	5.2.10.27/ 670
303A_0088	IRQ status register 3 of M4 (GPC_ISR3_M4)	32	R	0000_0000h	5.2.10.28/ 671
303A_008C	IRQ status register 4 of M4 (GPC_ISR4_M4)	32	R	0000_0000h	5.2.10.29/ 671
303A_00B0	Slot configure register for CPUs (GPC_SLT0_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00B4	Slot configure register for CPUs (GPC_SLT1_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00B8	Slot configure register for CPUs (GPC_SLT2_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_00BC	Slot configure register for CPUs (GPC_SLT3_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00C0	Slot configure register for CPUs (GPC_SLT4_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00C4	Slot configure register for CPUs (GPC_SLT5_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00C8	Slot configure register for CPUs (GPC_SLT6_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00CC	Slot configure register for CPUs (GPC_SLT7_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00D0	Slot configure register for CPUs (GPC_SLT8_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00D4	Slot configure register for CPUs (GPC_SLT9_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00D8	Slot configure register for CPUs (GPC_SLT10_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00DC	Slot configure register for CPUs (GPC_SLT11_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00E0	Slot configure register for CPUs (GPC_SLT12_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00E4	Slot configure register for CPUs (GPC_SLT13_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00E8	Slot configure register for CPUs (GPC_SLT14_CFG)	32	R/W	0000_0000h	5.2.10.30/ 671
303A_00EC	PGC CPU mapping (GPC_PGC_CPU_0_1_MAPPING)	32	R/W	0000_0000h	5.2.10.31/ 674
303A_00F0	CPU PGC software power up trigger (GPC_CPU_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.32/ 676
303A_00F4	MIX PGC software power up trigger (GPC_MIX_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.33/ 677
303A_00F8	PU PGC software up trigger (GPC_PU_PGC_SW_PUP_REQ)	32	R/W	0000_0000h	5.2.10.34/ 678
303A_00FC	CPU PGC software down trigger (GPC_CPU_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.35/ 679
303A_0100	MIX PGC software power down trigger (GPC_MIX_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.36/ 680
303A_0104	PU PGC software down trigger (GPC_PU_PGC_SW_PDN_REQ)	32	R/W	0000_0000h	5.2.10.37/ 681
303A_0108	Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC2)	32	R/W	0000_0000h	5.2.10.38/ 683
303A_0130	CPU PGC software up trigger status1 (GPC_CPU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.39/ 684
303A_0134	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.40/ 686

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0138	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.40/ 686
303A_013C	A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.40/ 686
303A_0140	M4 MIX PGC software up trigger status register (GPC_M4_MIX_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.41/ 687
303A_0144	M4 MIX PGC software up trigger status register (GPC_M4_MIX_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.41/ 687
303A_0148	M4 MIX PGC software up trigger status register (GPC_M4_MIX_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.41/ 687
303A_014C	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.42/ 689
303A_0150	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.42/ 689
303A_0154	A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.42/ 689
303A_0158	M4 PU PGC software up trigger status register (GPC_M4_PU_PGC_PUP_STATUS0)	32	R	0000_0000h	5.2.10.43/ 692
303A_015C	M4 PU PGC software up trigger status register (GPC_M4_PU_PGC_PUP_STATUS1)	32	R	0000_0000h	5.2.10.43/ 692
303A_0160	M4 PU PGC software up trigger status register (GPC_M4_PU_PGC_PUP_STATUS2)	32	R	0000_0000h	5.2.10.43/ 692
303A_0170	CPU PGC software dn trigger status1 (GPC_CPU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.44/ 695
303A_0174	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.45/ 697
303A_0178	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.45/ 697
303A_017C	A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.45/ 697
303A_0180	M4 MIX PGC software power down trigger status register (GPC_M4_MIX_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.46/ 699
303A_0184	M4 MIX PGC software power down trigger status register (GPC_M4_MIX_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.46/ 699
303A_0188	M4 MIX PGC software power down trigger status register (GPC_M4_MIX_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.46/ 699
303A_018C	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.47/ 700
303A_0190	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.47/ 700
303A_0194	A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.47/ 700
303A_0198	M4 PU PGC software down trigger status (GPC_M4_PU_PGC_PDN_STATUS0)	32	R	0000_0000h	5.2.10.48/ 703

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_019C	M4 PU PGC software down trigger status (GPC_M4_PU_PGC_PDN_STATUS1)	32	R	0000_0000h	5.2.10.48/ 703
303A_01A0	M4 PU PGC software down trigger status (GPC_M4_PU_PGC_PDN_STATUS2)	32	R	0000_0000h	5.2.10.48/ 703
303A_01B0	A53 MIX PDN FLG (GPC_A53_MIX_PDN_FLG)	32	R/W	0000_0000h	5.2.10.49/ 707
303A_01B4	A53 PU PDN FLG (GPC_A53_PU_PDN_FLG)	32	R/W	0000_0000h	5.2.10.50/ 708
303A_01B8	M4 MIX PDN FLG (GPC_M4_MIX_PDN_FLG)	32	R/W	0000_0000h	5.2.10.51/ 709
303A_01BC	M4 PU PDN FLG (GPC_M4_PU_PDN_FLG)	32	R/W	0000_0000h	5.2.10.52/ 710
303A_01C0	IRQ masking register 1 of A53 core2 (GPC_IMR1_CORE2_A53)	32	R/W	0000_0000h	5.2.10.53/ 710
303A_01C4	IRQ masking register 2 of A53 core2 (GPC_IMR2_CORE2_A53)	32	R/W	0000_0000h	5.2.10.54/ 711
303A_01C8	IRQ masking register 3 of A53 core2 (GPC_IMR3_CORE2_A53)	32	R/W	0000_0000h	5.2.10.55/ 711
303A_01CC	IRQ masking register 4 of A53 core2 (GPC_IMR4_CORE2_A53)	32	R/W	0000_0000h	5.2.10.56/ 712
303A_01D0	IRQ masking register 1 of A53 core3 (GPC_IMR1_CORE3_A53)	32	R/W	0000_0000h	5.2.10.57/ 712
303A_01D4	IRQ masking register 2 of A53 core3 (GPC_IMR2_CORE3_A53)	32	R/W	0000_0000h	5.2.10.58/ 713
303A_01D8	IRQ masking register 3 of A53 core3 (GPC_IMR3_CORE3_A53)	32	R/W	0000_0000h	5.2.10.59/ 713
303A_01DC	IRQ masking register 4 of A53 core3 (GPC_IMR4_CORE3_A53)	32	R/W	0000_0000h	5.2.10.60/ 714
303A_01E0	PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU)	32	R/W	0000_0000h	5.2.10.61/ 714
303A_01E4	PGC acknowledge signal selection of M4 platform for PUs (GPC_ACK_SEL_M4_PU)	32	R/W	0000_0000h	5.2.10.62/ 717
303A_01E8	Slot configure register for PGC CPUs (GPC_SLT15_CFG)	32	R/W	0000_0000h	5.2.10.63/ 719
303A_01EC	Slot configure register for PGC CPUs (GPC_SLT16_CFG)	32	R/W	0000_0000h	5.2.10.63/ 719
303A_01F0	Slot configure register for PGC CPUs (GPC_SLT17_CFG)	32	R/W	0000_0000h	5.2.10.63/ 719
303A_01F4	Slot configure register for PGC CPUs (GPC_SLT18_CFG)	32	R/W	0000_0000h	5.2.10.63/ 719
303A_01F8	Slot configure register for PGC CPUs (GPC_SLT19_CFG)	32	R/W	0000_0000h	5.2.10.63/ 719
303A_01FC	Power handshake register (GPC_PU_PWRHSK)	32	R/W	0000_FFFFh	5.2.10.64/ 722

Table continues on the next page...

GPC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0200	Slot configure register for PGC PUs (GPC_SLT0_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0204	Slot configure register for PGC PUs (GPC_SLT1_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0208	Slot configure register for PGC PUs (GPC_SLT2_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_020C	Slot configure register for PGC PUs (GPC_SLT3_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0210	Slot configure register for PGC PUs (GPC_SLT4_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0214	Slot configure register for PGC PUs (GPC_SLT5_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0218	Slot configure register for PGC PUs (GPC_SLT6_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_021C	Slot configure register for PGC PUs (GPC_SLT7_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0220	Slot configure register for PGC PUs (GPC_SLT8_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0224	Slot configure register for PGC PUs (GPC_SLT9_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0228	Slot configure register for PGC PUs (GPC_SLT10_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_022C	Slot configure register for PGC PUs (GPC_SLT11_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0230	Slot configure register for PGC PUs (GPC_SLT12_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0234	Slot configure register for PGC PUs (GPC_SLT13_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0238	Slot configure register for PGC PUs (GPC_SLT14_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_023C	Slot configure register for PGC PUs (GPC_SLT15_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0240	Slot configure register for PGC PUs (GPC_SLT16_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0244	Slot configure register for PGC PUs (GPC_SLT17_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_0248	Slot configure register for PGC PUs (GPC_SLT18_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724
303A_024C	Slot configure register for PGC PUs (GPC_SLT19_CFG_PU)	32	R/W	0000_0000h	5.2.10.65/ 724

5.2.10.1 Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC)

NOTE

LPCR_A53_BSC[CPU_CLK_ON_LPM] should be set 1'b1 when using A53 low power debug feature

NOTE

Always set LPM1/LPM0 with same value

Address: 303A_0000h base + 0h offset = 303A_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MASK_DSM_TRIGGER	IRQ_SRC_A53_WUP	IRQ_SRC_C1	IRQ_SRC_C0	Reserved	MASK_L2CC_WFI	Reserved	MASK_SCU_WFI	IRQ_SRC_C3	IRQ_SRC_C2	Reserved		MASK_CORE3_WFI	MASK_CORE2_WFI	MASK_CORE1_WFI	MASK_CORE0_WFI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	CPU_CLK_ON_LPM	Reserved							MST0_LPM_HSK_MASK	Reserved		LPM1		LPM0	
W																
Reset	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0

GPC_LPCR_A53_BSC field descriptions

Field	Description
31 MASK_DSM_TRIGGER	DSM Trigger Mask 0 DSM trigger of A53 platform will not be masked 1 DSM trigger of A53 platform will be masked
30 IRQ_SRC_A53_WUP	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See "Power up process for A53 platform" for more specific information.

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
	0 LPM wakeup source be “OR” result of LPCR_A53_BSC[IRQ_SRC_C0]/ LPCR_A53_BSC[IRQ_SRC_C1]/LPCR_A53_BSC[IRQ_SRC_C2]/LPCR_A53_BSC[IRQ_SRC_C3] setting 1 LPM wakeup source from external INT[127:0], masked by IMR0
29 IRQ_SRC_C1	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See “Power up process for A53 platform” for more specific information. 0 core1 wakeup source from external INT[127:0], masked by IMR1 refer to “Power up process for A53 platform” for more specific information 1 core1 wakeup source from GIC(nFIQ[1]/nIRQ[1]), SCU should not be power down during low power mode when this bit is set to 1’b1
28 IRQ_SRC_C0	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. See “Power up process for A53 platform” for more specific information. 0 core0 wakeup source from external INT[127:0], masked by IMR0 refer to “Power up process for A53 platform” for more specific information 1 core0 wakeup source from GIC(nFIQ[0]/nIRQ[0]), SCU should not be power down during low power mode when this bit is set to 1’b1
27 -	This field is reserved. Reserved
26 MASK_L2CC_ WFI	L2 cache controller Wait For Interrupt Mask Register 0 WFI for L2 cache controller is not masked 1 WFI for L2 cache controller is masked
25 -	This field is reserved. Reserved
24 MASK_SCU_WFI	SCU Wait For Interrupt Mask Register 0 WFI for SCU is not masked 1 WFI for SCU is masked
23 IRQ_SRC_C3	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. 0 core3 wakeup source from external INT[127:0], masked by IMR1. See Power Up Process for A53 Platform for more specific information. 1 core3 wakeup source from external GIC(nFIQ[1]/nIRQ[1]), SCU should not be powered down during low power mode when this bit is set to 1'b1.
22 IRQ_SRC_C2	LPCR_A53_BSC[IRQ_SRC_C0], LPCR_A53_BSC[IRQ_SRC_C1], LPCR_A53_BSC[IRQ_SRC_C2], LPCR_A53_BSC[IRQ_SRC_C3], and LPCR_A53_BSC[IRQ_SRC_A53_WUP] work together to decide the wake up source for A53 LPM and core0/core1/core2/core3 power. 0 core2 wakeup source from external INT[127:0], masked by IMR1. See Power Up Process for A53 Platform for more specific information. 1 core2 wakeup source from external GIC(nFIQ[1]/nIRQ[1]), SCU should not be powered down during low power mode when this bit is set to 1'b1.

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
21–20 -	This field is reserved.
19 MASK_CORE3_WFI	CORE3 Wait For Interrupt Mask 0 WFI for CORE3 is not masked 1 WFI for CORE3 is masked
18 MASK_CORE2_WFI	CORE2 Wait For Interrupt Mask 0 WFI for CORE2 is not masked 1 WFI for CORE2 is masked
17 MASK_CORE1_WFI	CORE1 Wait For Interrupt Mask 0 WFI for CORE1 is not masked 1 WFI for CORE1 is masked
16 MASK_CORE0_WFI	CORE0 Wait For Interrupt Mask 0 WFI for CORE0 is not masked 1 WFI for CORE0 is masked
15 -	This field is reserved. Reserved
14 CPU_CLK_ON_LPM	Define if A53 clocks will be disabled on wait/stop mode. 0 A53 clock disabled on wait/stop mode 1 A53 clock enabled on wait/stop mode
13–7 -	This field is reserved. Reserved
6 MST0_LPM_HSK_MASK	MASTER0 LPM handshake mask MASTER0(SCU) will handshake with GPC in LPM, follow this when you want this master power off. This bit should be used together with MST_CPU_MAPPING[0]. If you want power of SCU, use this setting: LPCR_A53_BSC[6]=0; MST_CPU_MAPPING[0]=1 Otherwise use: LPCR_A53_BSC[6]=1; MST_CPU_MAPPING[0]=0 0 enable MASTER0 LPM handshake, wait ACK from MASTER0 1 disable MASTER0 LPM handshake, mask ACK from MASTER0
5–4 -	This field is reserved. Reserved
3–2 LPM1	CORE1 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved
LPM0	CORE0 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode

Table continues on the next page...

GPC_LPCR_A53_BSC field descriptions (continued)

Field	Description
	10 Transfer to STOP mode 11 Reserved

5.2.10.2 Advanced Low power control register of A53 platform (GPC_LPCR_A53_AD)

Address: 303A_0000h base + 4h offset = 303A_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	L2PGE	Reserved			EN_C3_PUP	EN_C3_IRQ_PUP	EN_C2_PUP	EN_C2_IRQ_PUP	EN_C3_WFI_PDN_DIS	EN_C2_WFI_PDN_DIS	EN_C1_WFI_PDN_DIS	EN_C0_WFI_PDN_DIS	EN_C3_PDN	EN_C3_WFI_PDN	EN_C2_PDN	EN_C2_WFI_PDN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				EN_C1_PUP	EN_C1_IRQ_PUP	EN_C0_PUP	EN_C0_IRQ_PUP	Reserved		EN_L2_WFI_PDN	EN_PLAT_PDN	EN_C1_PDN	EN_C1_WFI_PDN	EN_C0_PDN	EN_C0_WFI_PDN
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

GPC_LPCR_A53_AD field descriptions

Field	Description
31 L2PGE	0 L2 cache RAM will power down with SCU power domain in A53 platform (used for ALL_OFF mode) 1 L2 cache RAM will not power down with SCU power domain in A53 platform (used for ALL_OFF mode)
30–28 -	This field is reserved. Reserved
27 EN_C3_PUP	0 CORE3 will not power up with lower power mode request 1 CORE3 will power up with low power mode request (only used wake up from CPU_OFF)
26 EN_C3_IRQ_PUP	0 CORE3 will not power up with IRQ request 1 CORE3 will power up with IRQ request
25 EN_C2_PUP	0 CORE2 will not power up with lower power mode request 1 CORE2 will power up with low power mode request (only used wake up from CPU_OFF)
24 EN_C2_IRQ_PUP	0 CORE2 will not power up with IRQ request 1 CORE2 will power up with IRQ request

Table continues on the next page...

GPC_LPCR_A53_AD field descriptions (continued)

Field	Description
23 EN_C3_WFI_PDN_DIS	0 Disable WFI power down core3 1 Enable WFI power down core3
22 EN_C2_WFI_PDN_DIS	0 Disable WIFI power down core2 1 Enable WIFI power down core2
21 EN_C1_WFI_PDN_DIS	0 Disable WIFI power down core1 1 Enable WIFI power down core1
20 EN_C0_WFI_PDN_DIS	0 Disable WIFI power down core0 1 Enable WIFI power down core0
19 EN_C3_PDN	0 CORE3 will not be power down with low power mode request 1 CORE3 will be power down with low power mode request
18 EN_C3_WFI_PDN	0 CORE3 will not be power down with WFI request 1 CORE3 will be power down with WFI request
17 EN_C2_PDN	0 CORE2 will not be power down with low power mode request 1 CORE2 will be power down with low power mode request
16 EN_C2_WFI_PDN	0 CORE2 will not be power down with WFI request 1 CORE2 will be power down with WFI request
15–12 -	This field is reserved. Reserved
11 EN_C1_PUP	0 CORE1 will not power up with low power mode request (only used wake up from CPU01_OFF mode) 1 CORE1 will power up with low power mode request
10 EN_C1_IRQ_PUP	0 CORE1 will not power up with IRQ request 1 CORE1 will power up with IRQ request
9 EN_C0_PUP	(only used wake up from CPU01_OFF mode) 0 CORE0 will not power up with low power mode request 1 CORE0 will power up with low power mode request
8 EN_C0_IRQ_PUP	0 CORE0 will not power up with IRQ request 1 CORE0 will power up with IRQ request
7–6 -	This field is reserved. Reserved
5 EN_L2_WFI_PDN	**NOTE:** Before reset, L2 WFI is 1 and make GPC generate an error DSM request. This bit is used to mask the L2 WFI before reset. After reset, L2 WFI change to 0, and functions are OK, SW must clear this bit at the beginning of code. 0 SCU and L2 will not be power down with WFI request 1 SCU and L2 will be power down with WFI request (default)
4 EN_PLAT_PDN	0 SCU and L2 cache RAM will not be power down with low power mode request 1 SCU and L2 cache RAM will be power down with low power mode request

Table continues on the next page...

GPC_LPCR_A53_AD field descriptions (continued)

Field	Description
3 EN_C1_PDN	0 CORE1 will not be power down with low power mode request 1 CORE1 will be power down with low power mode request
2 EN_C1_WFI_PDN	0 CORE1 will not be power down with WFI request 1 CORE1 will be power down with WFI request
1 EN_C0_PDN	0 CORE0 will not be power down with low power mode request 1 CORE0 will be power down with low power mode request
0 EN_C0_WFI_PDN	0 CORE0 will not be power down with WFI request 1 CORE0 will be power down with WFI request

5.2.10.3 Low power control register of CPU1 (GPC_LPCR_M4)

NOTE

LPCR_M4[CPU_CLK_ON_LPM] should be set 1'b0 if M4 goes to LPM without trigger power down of related domains

Address: 303A_0000h base + 8h offset = 303A_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MASK_DSM_TRIGGER	Reserved														MASK_M4_WFI
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	CPU_CLK_ON_LPM	Reserved										EN_M4_PUP	EN_M4_PDN	LPM0	
W																
Reset	0	0	1	1	1	1	1	1	1	1	1	1	0	0	0	0

GPC_LPCR_M4 field descriptions

Field	Description
31 MASK_DSM_TRIGGER	M4 WFI Mask 0 DSM trigger of M4 platform will not be masked 1 DSM trigger of M4 platform will be masked
30–17 -	This field is reserved. Reserved
16 MASK_M4_WFI	M4 WFI Mask 0 WFI for M4 is not masked 1 WFI for M4 is masked
15 -	This field is reserved. Reserved
14 CPU_CLK_ON_LPM	Define if M4 clocks will be disabled on wait/stop mode. 0 M4 clock disabled on wait/stop mode. 1 M4 clock enabled on wait/stop mode.
13–4 -	This field is reserved. Reserved
3 EN_M4_PUP	Enable m4 virtual PGC power up with LPM enter
2 EN_M4_PDN	Enable m4 virtual PGC power down with LPM enter
LPM0	Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

5.2.10.4 System low power control register (GPC_SLPCR)

NOTE

SLPCR[VSTBY] must be set to 1'b1 if SLPCR[RBC_EN] is set to 1'b1; SLPCR[SBYOS] must be set to 1'b1 if SLPCR[VSTBY] is set to 1'b1.

Address: 303A_0000h base + 14h offset = 303A_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN_DSM	RBC_EN	REG_BYPASS_COUNT						DISABLE_A53_IS_DSM	Reserved			EN_M4_FASTWUP_STOP_MODE	EN_M4_FASTWUP_WAIT_MODE	EN_A53_FASTWUP_STOP_MODE	EN_A53_FASTWUP_WAIT_MODE
Reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OSCCNT								COSC_EN	COSC_PWRDOWN	STBY_COUNT			VSTBY	SBYOS	BYPASS_PMIC_READY
Reset	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	0

GPC_SLPCR field descriptions

Field	Description
31 EN_DSM	DSM enable 0 DSM disabled 1 DSM enabled
30 RBC_EN	Enable for REG_BYPASS_COUNTER. If enabled, REG_BYPASS signal will be asserted after REG_BYPASS_COUNT clocks of CKIL, after standby voltage is requested. If standby voltage is not requested REG_BYPASS won't be asserted, even if counter is enabled. 0 REG_BYPASS_COUNTER disabled 1 REG_BYPASS_COUNTER enabled
29–24 REG_BYPASS_COUNT	Counter for REG_BYPASS signal assertion after standby voltage request by PMIC_STBY_REQ. 000000 no delay 000001 1 CKIL clock period delay 111111 63 CKIL clock period delay

Table continues on the next page...

GPC_SLPCR field descriptions (continued)

Field	Description
23 DISABLE_A53_ISO_DSM	0 Enable A53 isolation signal in DSM 1 Disable A53 isolation signal in DSM
22–20 -	This field is reserved. Reserved
19 EN_M4_FASTWUP_STOP_MODE	Enable M4 fast wake up stop mode, relevant PLLs will not be closed in this mode.
18 EN_M4_FASTWUP_WAIT_MODE	Enable M4 fast wake up wait mode, relevant PLLs will not be closed in this mode.
17 EN_A53_FASTWUP_STOP_MODE	Enable A53 fast wake up stop mode, relevant PLLs will not be closed in this mode.
16 EN_A53_FASTWUP_WAIT_MODE	Enable A53 fast wake up wait mode, relevant PLLs will not be closed in this mode.
15–8 OSCCNT	Oscillator ready counter value. These bits define value of 32KHz counter, that serve as counter for oscillator lock time. This is used for oscillator lock time. Current estimation is ~5ms. This counter will be used in sequence out of DSM and if sbyos bit was defined. GPC will wait the “OSCCNT” number of cycles before it notify CCM to open the relevant PLLs. 00000000 count 1 ckil 11111111 count 256 ckils
7 COSC_EN	On-chip oscillator enable bit - this bit value is reflected on the output cosc_en. The system will start with on-chip oscillator enabled to supply source for the PLLs. Software can change this bit if a transition to the bypass PLL clocks was performed for all the PLLs. In cases that this bit is changed from ‘0’ to ‘1’ then GPC will enable the on-chip oscillator and after counting oscnt ckil clock cycles before it notify CCM to open the relevant PLLs . The cosc_en bit should be changed only when on-chip oscillator is not chosen as the clock source. 0 Disable on-chip oscillator 1 Enable on-chip oscillator
6 COSC_PWRDOWN	In run mode, software can manually control powering down of on chip oscillator, i.e. generating ‘1’ on cosc_pwrdown signal. If software manually powered down the on chip oscillator, then sbyos functionality for on-chip oscillator will be bypassed. The manual closing of on-chip oscillator should be performed only in case the reference oscillator is not the source of all the clocks generation. 0 On-chip oscillator will not be powered down, i.e. cosc_pwrdown = 0 1 On-chip oscillator will be powered down, i.e. cosc_pwrdown = 1
5–3 STBY_COUNT	Standby counter definition. These two bits define, in the case of stop exit (if VSTBY bit was set), the amount of time GPC will wait between PMIC_STBY_REQ negation and the check of assertion of PMIC_READY. 000 GPC will wait 4 ckil clock cycles 001 GPC will wait 8 ckil clock cycles

Table continues on the next page...

GPC_SLPCR field descriptions (continued)

Field	Description
	010 GPC will wait 16 ckil clock cycles 011 GPC will wait 32 ckil clock cycles 100 GPC will wait 64 ckil clock cycles 101 GPC will wait 128 ckil clock cycles 110 GPC will wait 256 ckil clock cycles 111 GPC will wait 512 ckil clock cycles
2 VSTBY	Voltage standby request bit. This bit defines if PMIC_STBY_REQ pin, which notifies external power management IC to move from functional voltage to standby voltage, will be asserted in stop mode. 0 Voltage will not be changed to standby voltage after next entrance to stop mode. (PMIC_STBY_REQ will remain negated - '0') 1 Voltage will be changed to standby voltage after next entrance to stop mode.
1 SBYOS	Standby clock oscillator bit. This bit defines if cosc_pwrdown, which power down the on chip oscillator, will be asserted in DSM. 0 On chip oscillator will not be powered down, after next entrance to DSM. 1 On chip oscillator will be powered down, after next entrance to DSM. When returning from DSM, external oscillator will be enabled again, on chip oscillator will return to oscillator mode , and after oscnt count GPC will continue with the exit from DSM process.
0 BYPASS_PMIC_READY	By asserting this bit GPC will bypass waiting for PMIC_READY signal when coming out of DSM. This should be used for PMIC's that don't support the PMIC_READY signal. 0 Don't bypass the PMIC_READY signal - GPC will wait for its assertion during exit of low power mode if standby voltage was enabled 1 Bypass the PMIC_READY signal - GPC will not wait for its assertion during exit of low power mode if standby voltage was enabled

5.2.10.5 MASTER LPM Handshake (GPC_MST_CPU_MAPPING)

Address: 303A_0000h base + 18h offset = 303A_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEMLP_RET_PGEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MEMLP_RET_PGEN															MST0_CPU_MAPPING
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

GPC_MST_CPU_MAPPING field descriptions

Field	Description
31–1 MEMLP_RET_PGEN	Delay counter for "retnx" and "pgen"
0 MST0_CPU_MAPPING	MASTER0 CPU Mapping SCU LPM handshake mask. This bit should be used together with LPCR_A7_BSC[6]. If you want power of SCU, use this setting: LPCR_A7_BSC[6]=0; MST_CPU_MAPPING[0]=1 Otherwise, use this: LPCR_A7_BSC[6]=1; MST_CPU_MAPPING[0]=0 0 GPC will not send out power off requirement 1 GPC will send out power off requirement

5.2.10.6 Memory low power control register (GPC_MLPCR)

Address: 303A_0000h base + 20h offset = 303A_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MEMLP_RET_PGEN								MEM_EXT_CNT							
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MEMLP_ENT_CNT								Reserved					ROMLP_PDN_DIS	MEMLP_RET_SEL	MEMLP_CTL_DIS
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

GPC_MLPCR field descriptions

Field	Description
31–24 MEMLP_RET_PGEN	Delay counter for “retnx” and “pgen”
23–16 MEM_EXT_CNT	Delay counter to start existing from memory low power
15–8 MEMLP_ENT_CNT	Delay counter to make sure all clock off after pll_dis_req is issued by smc

Table continues on the next page...

GPC_MLPCR field descriptions (continued)

Field	Description
7–3 -	This field is reserved. Reserved
2 ROMLP_PDN_DIS	ROM shut down control 0 Enable ROM shut down control(should also enable RAM low power control); 1 Disable ROM shut down control
1 MEMLP_RET_SEL	Retention select 0 retention mode 2 1 retention mode 1
0 MEMLP_CTL_DIS	RAM low-power control 0 Enable RAM low power control 1 Disable RAM low power control

5.2.10.7 PGC acknowledge signal selection of A53 platform (GPC_PGC_ACK_SEL_A53)

The register can only be accessed by A53 platform

Address: 303A_0000h base + 24h offset = 303A_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	A53_PGC_PUP_ACK	A53_C3_PGC_PUP_ACK	A53_C2_PGC_PUP_ACK	Reserved										A53_PLAT_PGC_PUP_ACK	A53_C1_PGC_PUP_ACK	A53_C0_PGC_PUP_ACK
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	A53_PGC_PDN_ACK	A53_C3_PGC_PDN_ACK	A53_C2_PGC_PDN_ACK	Reserved										A53_PLAT_PGC_PDN_ACK	A53_C1_PGC_PDN_ACK	A53_C0_PGC_PDN_ACK
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_ACK_SEL_A53 field descriptions

Field	Description
31 A53_PGC_PUP_ACK	Select power up acknowledge signal of A53 (dummy) PGC as the power up acknowledge for A53 LPM.
30 A53_C3_PGC_PUP_ACK	Select power up acknowledge signal of A53 CORE3 PGC as the power up acknowledge for A53 LPM.
29 A53_C2_PGC_PUP_ACK	Select power up acknowledge signal of A53 CORE2 PGC as the power up acknowledge for A53 LPM.
28–19 -	This field is reserved. Reserved
18 A53_PLAT_PGC_PUP_ACK	Select power up acknowledge signal of A53 PLATFORM PGC as the power up acknowledge for A53 LPM.
17 A53_C1_PGC_PUP_ACK	Select power up acknowledge signal of A53 CORE1 PGC as the power up acknowledge for A53 LPM.
16 A53_C0_PGC_PUP_ACK	Select power up acknowledge signal of A53 CORE0 PGC as the power up acknowledge for A53 LPM.
15 A53_PGC_PDN_ACK	Select power down acknowledge signal of A53 (dummy) PGC as the power down acknowledge for A53 LPM.
14 A53_C3_PGC_PDN_ACK	Select power down acknowledge signal of A53 CORE3 PGC as the power down acknowledge for A53 LPM.
13 A53_C2_PGC_PDN_ACK	Select power down acknowledge signal of A53 CORE2 PGC as the power down acknowledge for A53 LPM.
12–3 -	This field is reserved. Reserved
2 A53_PLAT_PGC_PDN_ACK	Select power down acknowledge signal of A53 PLATFORM PGC as the power down acknowledge for A53 LPM.
1 A53_C1_PGC_PDN_ACK	Select power down acknowledge signal of A53 CORE1 PGC as the power down acknowledge for A53 LPM.
0 A53_C0_PGC_PDN_ACK	Select power down acknowledge signal of A53 CORE0 PGC as the power down acknowledge for A53 LPM.

5.2.10.8 PGC acknowledge signal selection of M4 platform (GPC_PGC_ACK_SEL_M4)

This register can only be accessed by the M4 platform.

NOTE

"dummy" PGC cannot be mapped to time slot control. "virtual" PGC can be mapped to time slot control. When virtual PGC is used, below setting is required - GPC_MISC[M4_PDN_REQ_MASK] should be set to 1'b1 and arrange virtual GPC in same slot with MIX. power/up slot (See example code 3). MIX PGC may possibly power down later than A53 platform power down when virtual PGC is used.

Address: 303A_0000h base + 28h offset = 303A_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	M4_DUMMY_PGC_PUP_ACK	Reserved														M4_VIRTUAL_PGC_PUP_ACK
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	M4_DUMMY_PGC_PDN_ACK	Reserved														M4_VIRTUAL_PGC_PDN_ACK
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_ACK_SEL_M4 field descriptions

Field	Description
31 M4_DUMMY_PGC_PUP_ACK	Select power up acknowledge signal of M4 (dummy) PGC as the power up acknowledge for M4 LPM.
30–17 -	This field is reserved. Reserved

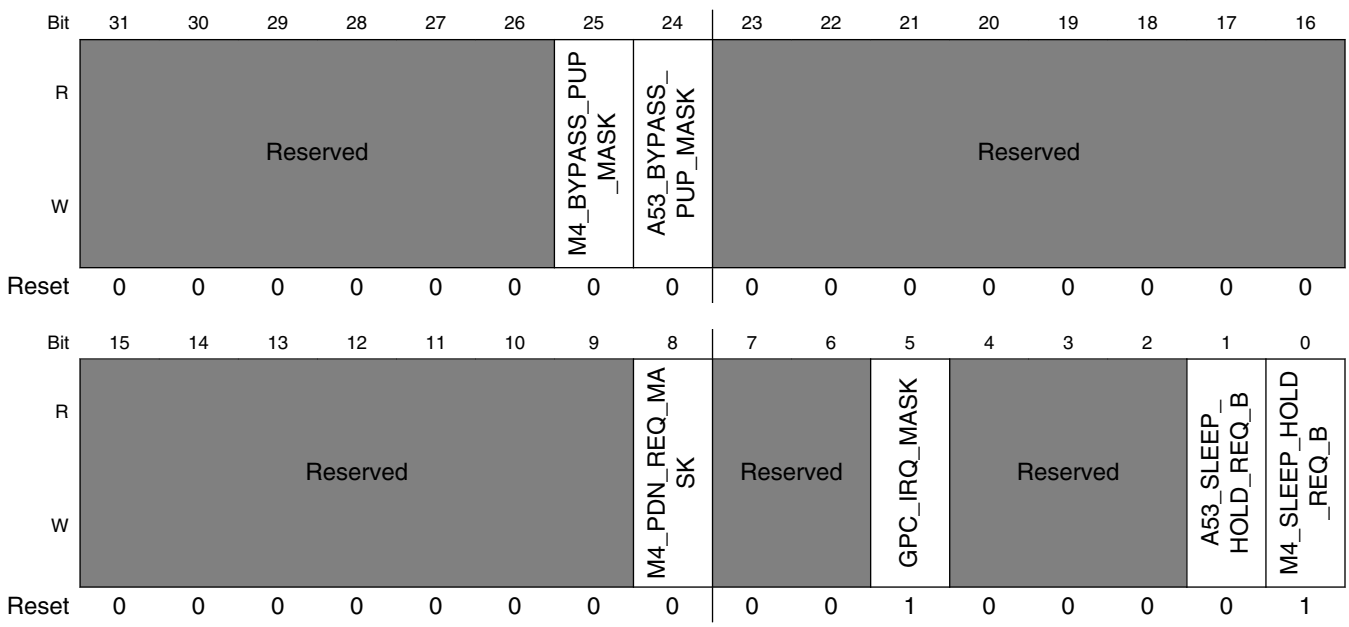
Table continues on the next page...

GPC_PGC_ACK_SEL_M4 field descriptions (continued)

Field	Description
16 M4_VIRTUAL_ PGC_PUP_ACK	Select power up acknowledge signal of M4 virtual PGC as the power up acknowledge for M4 LPM. M4 virtual PGC only acknowledge power up request in the end of current slot time
15 M4_DUMMY_ PGC_PDN_ACK	Select power down acknowledge signal of M4 (dummy) PGC as the power down acknowledge for M4 LPM.
14–1 -	This field is reserved. Reserved
0 M4_VIRTUAL_ PGC_PDN_ACK	Select power down acknowledge signal of M4 virtual PGC as the power down acknowledge for M4 LPM. M4 virtual PGC only acknowledge power down request in the end of current slot time

5.2.10.9 GPC Miscellaneous register (GPC_MISC)

Address: 303A_0000h base + 2Ch offset = 303A_002Ch



GPC_MISC field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25 M4_BYPASS_PUP_MASK	

Table continues on the next page...

GPC_MISC field descriptions (continued)

Field	Description
24 A53_BYPASS_PUP_MASK	
23–9 -	This field is reserved. Reserved
8 M4_PDN_REQ_MASK	M4 power-down mask 0 M4 power down request to virtual M4 PGC will be masked. 1 M4 power down request to virtual M4 PGC will not be masked. Set this bit to 1'b1 when M4 virtual PGC is used.
7–6 -	This field is reserved. Reserved
5 GPC_IRQ_MASK	GPC interrupt/event masking 0 Not masked 1 Interrupt / event is masked
4–2 -	This field is reserved. Reserved
1 A53_SLEEP_HOLD_REQ_B	A53 sleep hold 0 Hold A53 platform in sleep mode. This bit is a software control bit to A53 platform. 1 Don't hold A53 platform in sleep mode.
0 M4_SLEEP_HOLD_REQ_B	M4 sleep hold 0 Hold M4 platform in sleep mode. This bit is a software control bit to M4 platform. 1 Don't hold M4 platform in sleep mode.

5.2.10.10 IRQ masking register 1 of A53 core0 (GPC_IMR1_CORE0_A53)

The four IMRn_CORE0_A53 (n = 1,2,3,4) registers are used as interrupt mask for A53 core0.

Address: 303A_0000h base + 30h offset = 303A_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE0_A53 field descriptions

Field	Description
IMR1_CORE0_A53	A53 core0 IRQ[31:0] masking bits:

GPC_IMR1_CORE0_A53 field descriptions (continued)

Field	Description
	0 IRQ not masked 1 IRQ masked

5.2.10.11 IRQ masking register 2 of A53 core0 (GPC_IMR2_CORE0_A53)

Address: 303A_0000h base + 34h offset = 303A_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE0_A53 field descriptions

Field	Description
IMR2_CORE0_A53	A53 core0 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.12 IRQ masking register 3 of A53 core0 (GPC_IMR3_CORE0_A53)

Address: 303A_0000h base + 38h offset = 303A_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE0_A53 field descriptions

Field	Description
IMR3_CORE0_A53	A53 core0 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.13 IRQ masking register 4 of A53 core0 (GPC_IMR4_CORE0_A53)

Address: 303A_0000h base + 3Ch offset = 303A_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE0_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE0_A53 field descriptions

Field	Description
IMR4_CORE0_A53	A53 core0 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.14 IRQ masking register 1 of A53 core1 (GPC_IMR1_CORE1_A53)

The four IMRn_CORE1_A53 (n = 1,2,3,4) registers are used as interrupt mask for A53 core1.

Address: 303A_0000h base + 40h offset = 303A_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE1_A53 field descriptions

Field	Description
IMR1_CORE1_A53	A53 core1 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.15 IRQ masking register 2 of A53 core1 (GPC_IMR2_CORE1_A53)

Address: 303A_0000h base + 44h offset = 303A_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE1_A53 field descriptions

Field	Description
IMR2_CORE1_A53	A53 core1 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.16 IRQ masking register 3 of A53 core1 (GPC_IMR3_CORE1_A53)

Address: 303A_0000h base + 48h offset = 303A_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE1_A53 field descriptions

Field	Description
IMR3_CORE1_A53	A53 core1 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.17 IRQ masking register 4 of A53 core1 (GPC_IMR4_CORE1_A53)

Address: 303A_0000h base + 4Ch offset = 303A_004Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE1_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE1_A53 field descriptions

Field	Description
IMR4_CORE1_A53	A53 core1 IRQ[127:96] masking bits:

GPC_IMR4_CORE1_A53 field descriptions (continued)

Field	Description
	0 IRQ not masked 1 IRQ masked

5.2.10.18 IRQ masking register 1 of M4 (GPC_IMR1_M4)

The four IMRn_M4 (n = 1,2,3,4) registers are used as interrupt mask for M4.

Address: 303A_0000h base + 50h offset = 303A_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_M4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_M4 field descriptions

Field	Description
IMR1_M4	M4 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.19 IRQ masking register 2 of M4 (GPC_IMR2_M4)

Address: 303A_0000h base + 54h offset = 303A_0054h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_M4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_M4 field descriptions

Field	Description
IMR2_M4	M4 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.20 IRQ masking register 3 of M4 (GPC_IMR3_M4)

Address: 303A_0000h base + 58h offset = 303A_0058h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_M4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_M4 field descriptions

Field	Description
IMR3_M4	M4 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.21 IRQ masking register 4 of M4 (GPC_IMR4_M4)

Address: 303A_0000h base + 5Ch offset = 303A_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_M4																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_M4 field descriptions

Field	Description
IMR4_M4	M4 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.22 IRQ status register 1 of A53 (GPC_ISR1_A53)

The four ISRn_A53 (n = 1,2,3,4) registers, all of them are read only register

Address: 303A_0000h base + 70h offset = 303A_0070h

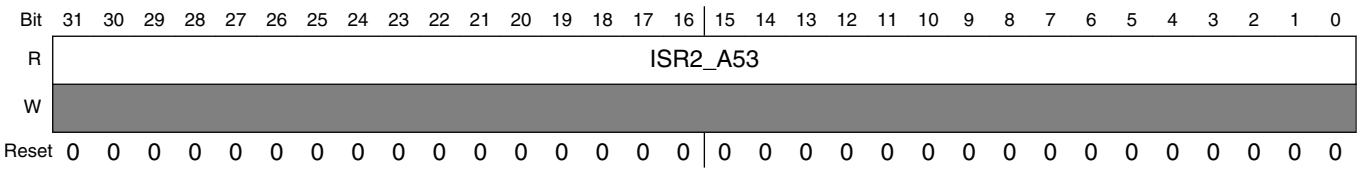
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR1_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR1_A53 field descriptions

Field	Description
ISR1_A53	A53 IRQ[31:0] status

5.2.10.23 IRQ status register 2 of A53 (GPC_ISR2_A53)

Address: 303A_0000h base + 74h offset = 303A_0074h

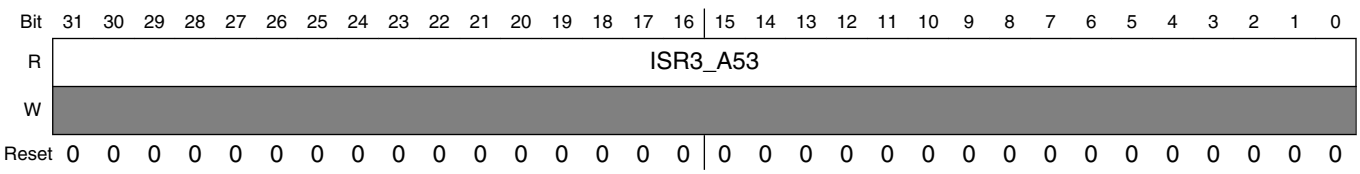


GPC_ISR2_A53 field descriptions

Field	Description
ISR2_A53	A53 IRQ[63:32] status

5.2.10.24 IRQ status register 3 of A53 (GPC_ISR3_A53)

Address: 303A_0000h base + 78h offset = 303A_0078h



GPC_ISR3_A53 field descriptions

Field	Description
ISR3_A53	A53 IRQ[95:64] status

5.2.10.25 IRQ status register 4 of A53 (GPC_ISR4_A53)

Address: 303A_0000h base + 7Ch offset = 303A_007Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR4_A53																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR4_A53 field descriptions

Field	Description
ISR4_A53	A53 IRQ[127:96] status

5.2.10.26 IRQ status register 1 of M4 (GPC_ISR1_M4)

The four ISRn_M4 (n = 1,2,3,4) registers, all of them are read only register

Address: 303A_0000h base + 80h offset = 303A_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR1_M4																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR1_M4 field descriptions

Field	Description
ISR1_M4	M4 IRQ[31:0] status

5.2.10.27 IRQ status register 2 of M4 (GPC_ISR2_M4)

Address: 303A_0000h base + 84h offset = 303A_0084h

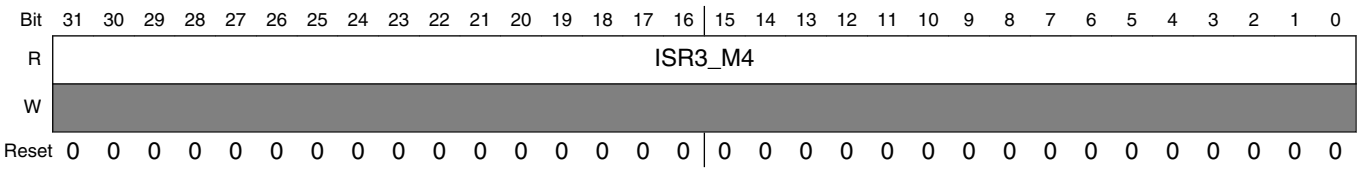
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR2_M4																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ISR2_M4 field descriptions

Field	Description
ISR2_M4	M4 IRQ[63:32] status

5.2.10.28 IRQ status register 3 of M4 (GPC_ISR3_M4)

Address: 303A_0000h base + 88h offset = 303A_0088h

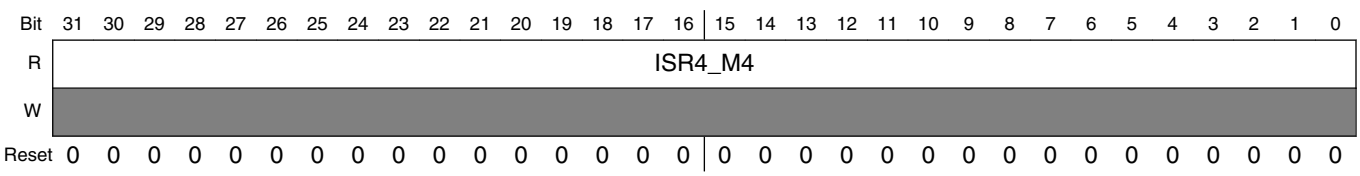


GPC_ISR3_M4 field descriptions

Field	Description
ISR3_M4	M4 IRQ[95:64] status

5.2.10.29 IRQ status register 4 of M4 (GPC_ISR4_M4)

Address: 303A_0000h base + 8Ch offset = 303A_008Ch



GPC_ISR4_M4 field descriptions

Field	Description
ISR4_M4	M4 IRQ[127:96] status

5.2.10.30 Slot configure register for CPUs (GPC_SLT*n*_CFG)

There are 20 slots in each SLTn_CFG(n = 0~19) that define the power up or power down behavior of one or more A53 core, NOC, or SCU PGC in each slot. This array contains slots 0 to 14, see Memory Map for slots 15 to 19.

In each “SLTn_cfg”, 2 bits (slt_cfg[1:0])are reserved for each PGC:

- 2’b01 (slot controller will power down relevant PGC in corresponding slot if hardware power down request asserted)
- 2’b10 (slot controller will power up relevant PGC in corresponding slot if hardware power up request asserted)
- 2’b00 or 2’b11 (not power down or power up behavior in relevant slot)

The specific bits assignment for each PGC is shown in the table below.

	PGCx	PGCx-1	..	PGC2	PGC1	PGC0
SLT0_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLT1_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
:	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLTn_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]

Address: 303A_0000h base + B0h offset + (4d × i), where i=0d to 14d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						SCU_PUP_SLOT_CONTROL	SCU_PDN_SLOT_CONTROL	CORE3_A53_PUP_SLOT_CONTROL	CORE3_A53_PDN_SLOT_CONTROL	CORE2_A53_PUP_SLOT_CONTROL	CORE2_A53_PDN_SLOT_CONTROL	CORE1_A53_PUP_SLOT_CONTROL	CORE1_A53_PDN_SLOT_CONTROL	CORE0_A53_PUP_SLOT_CONTROL	CORE0_A53_PDN_SLOT_CONTROL
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG field descriptions

Field	Description
31–10 -	This field is reserved.
9 SCU_PUP_ SLOT_ CONTROL	SCU Power-up slot control
8 SCU_PDN_ SLOT_ CONTROL	SCU Power-down slot control
7 CORE3_A53_ PUP_SLOT_ CONTROL	CORE3 A53 Power-up slot control

Table continues on the next page...

GPC_SLT*n*_CFG field descriptions (continued)

Field	Description
6 CORE3_A53_ PDN_SLOT_ CONTROL	CORE3 A53 Power-down slot control
5 CORE2_A53_ PUP_SLOT_ CONTROL	CORE2 A53 Power-up slot control
4 CORE2_A53_ PDN_SLOT_ CONTROL	CORE2 A53 Power-down slot control
3 CORE1_A53_ PUP_SLOT_ CONTROL	CORE1 A53 Power-up slot control
2 CORE1_A53_ PDN_SLOT_ CONTROL	CORE1 A53 Power-down slot control
1 CORE0_A53_ PUP_SLOT_ CONTROL	CORE0 A53 Power-up slot control
0 CORE0_A53_ PDN_SLOT_ CONTROL	CORE0 A53 Power-down slot control

5.2.10.31 PGC CPU mapping (GPC_PGC_CPU_0_1_MAPPING)

Address: 303A_0000h base + ECh offset = 303A_00ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCIE2_M4_DOMAIN	MIPI_CSI2_M4_DOMAIN	MIPI_CSI1_M4_DOMAIN	DISP_M4_DOMAIN	HDMI_M4_DOMAIN	VPU_M4_DOMAIN	GPU_M4_DOMAIN	DDR2_M4_DOMAIN	DDR1_M4_DOMAIN	Reserved	OTG2_M4_DOMAIN	OTG1_M4_DOMAIN	PCIE_M4_DOMAIN	MIPI_M4_DOMAIN	Reserved	MF_M4_DOMAIN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCIE2_A53_DOMAIN	MIPI_CSI2_A53_DOMAIN	MIPI_CSI1_A53_DOMAIN	DISP_A53_DOMAIN	HDMI_A53_DOMAIN	VPU_A53_DOMAIN	GPU_A53_DOMAIN	DDR2_A53_DOMAIN	DDR1_A53_DOMAIN	Reserved	OTG2_A53_DOMAIN	OTG1_A53_DOMAIN	PCIE_A53_DOMAIN	MIPI_A53_DOMAIN	Reserved	MF_A53_DOMAIN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PGC_CPU_0_1_MAPPING field descriptions

Field	Description
31 PCIE2_M4_DOMAIN	PCIE2_M4_DOMAIN
30 MIPI_CSI2_M4_DOMAIN	MIPI_CSI2_M4_DOMAIN
29 MIPI_CSI1_M4_DOMAIN	MIPI_CSI1_M4_DOMAIN
28 DISP_M4_DOMAIN	DISP_M4_DOMAIN
27 HDMI_M4_DOMAIN	HDMI_M4_DOMAIN

Table continues on the next page...

GPC_PGC_CPU_0_1_MAPPING field descriptions (continued)

Field	Description
26 VPU_M4_ DOMAIN	VPU_M4_DOMAIN
25 GPU_M4_DOMA IN	GPU_M4_DOMAIN
24 DDR2_M4_DOM AIN	DDR2_M4_DOMAIN
23 DDR1_M4_DOM AIN	DDR1_M4_DOMAIN
22 -	This field is reserved.
21 OTG2_M4_ DOMAIN	OTG2_M4_DOMAIN
20 OTG1_M4_DOM AIN	OTG1_M4_DOMAIN
19 PCIE_M4_ DOMAIN	PCIE_M4_DOMAIN
18 MIPI_M4_DOMAI N	MIPI_M4_DOMAIN
17 -	This field is reserved. Reserved
16 MF_M4_ DOMAIN	MF_M4_DOMAIN
15 PCIE2_A53_DO MAIN	PCIE2_A53_DOMAIN
14 MIPI_CSI2_A53_ DOMAIN	MIPI_CSI2_A53_DOMAIN
13 MIPI_CSI1_A53_ DOMAIN	MIPI_CSI1_A53_DOMAIN
12 DISP_A53_ DOMAIN	DISP_A53_DOMAIN
11 HDMI_A53_DOM AIN	HDMI_A53_DOMAIN
10 VPU_A53_ DOMAIN	VPU_A53_DOMAIN

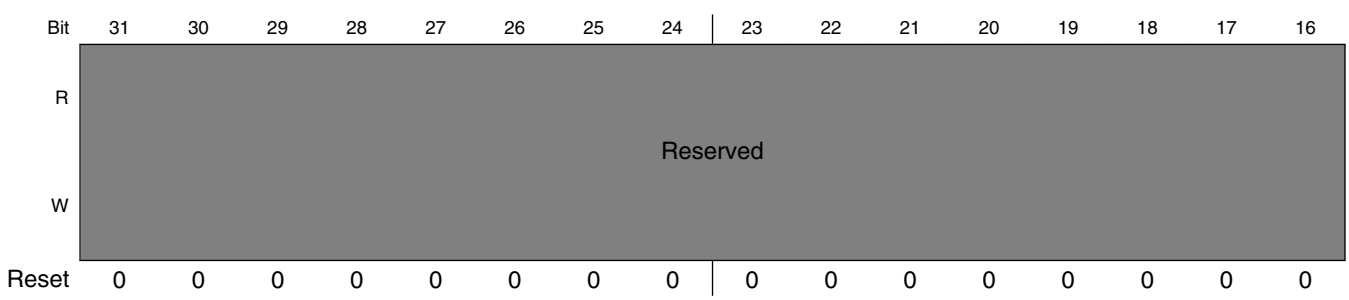
Table continues on the next page...

GPC_PGC_CPU_0_1_MAPPING field descriptions (continued)

Field	Description
9 GPU_A53_DOMAIN	GPU_A53_DOMAIN
8 DDR2_A53_DOMAIN	DDR2_A53_DOMAIN
7 DDR1_A53_DOMAIN	DDR1_A53_DOMAIN
6 -	This field is reserved.
5 OTG2_A53_DOMAIN	OTG2_A53_DOMAIN
4 OTG1_A53_DOMAIN	OTG1_A53_DOMAIN
3 PCIE_A53_DOMAIN	PCIE_A53_DOMAIN
2 MIPI_A53_DOMAIN	MIPI A53 DOMAIN
1 -	This field is reserved. Reserved
0 MF_A53_DOMAIN	MF_A53_DOMAIN

5.2.10.32 CPU PGC software power up trigger (GPC_CPU_PGC_SW_PUP_REQ)

Address: 303A_0000h base + F0h offset = 303A_00F0h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											CORE3_A53_SW_PUP_REQ	CORE2_A53_SW_PUP_REQ	SCU_A53_SW_PUP_REQ	CORE1_A53_SW_PUP_REQ	CORE0_A53_SW_PUP_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_CPU_PGC_SW_PUP_REQ field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 CORE3_A53_SW_PUP_REQ	Software power up trigger for Core3 A53 PGC
3 CORE2_A53_SW_PUP_REQ	Software power up trigger for Core2 A53 PGC
2 SCU_A53_SW_PUP_REQ	Software power up trigger for SCU A53
1 CORE1_A53_SW_PUP_REQ	Software power up trigger for Core1 A53 PGC
0 CORE0_A53_SW_PUP_REQ	Software power up trigger for Core0 A53 PGC

5.2.10.33 MIX PGC software power up trigger (GPC_MIX_PGC_SW_PUP_REQ)

Address: 303A_0000h base + F4h offset = 303A_00F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															MIX_SW_PUP_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_MIX_PGC_SW_PUP_REQ field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 MIX_SW_PUP_REQ	Software power up trigger for MIX PGC

5.2.10.34 PU PGC software up trigger (GPC_PU_PGC_SW_PUP_REQ)

Address: 303A_0000h base + F8h offset = 303A_00F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		PCIE2_SW_PUP_REQ	MIPI_CSI2_SW_PUP_REQ	MIPI_CSI1_SW_PUP_REQ	DISP_SW_PUP_REQ	HDMI_SW_PUP_REQ	VPU_SW_PUP_REQ	GPU_SW_PUP_REQ	DDR2_SW_PUP_REQ	DDR1_SW_PUP_REQ	Reserved	USB_OTG2_SW_PUP_REQ	USB_OTG1_SW_PUP_REQ	PCIE_SW_PUP_REQ	MIPI_SW_PUP_REQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PU_PGC_SW_PUP_REQ field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 PCIE2_SW_PUP_REQ	Software power up trigger for PCIE2
12 MIPI_CSI2_SW_PUP_REQ	Software power up trigger for MIPI CSI2

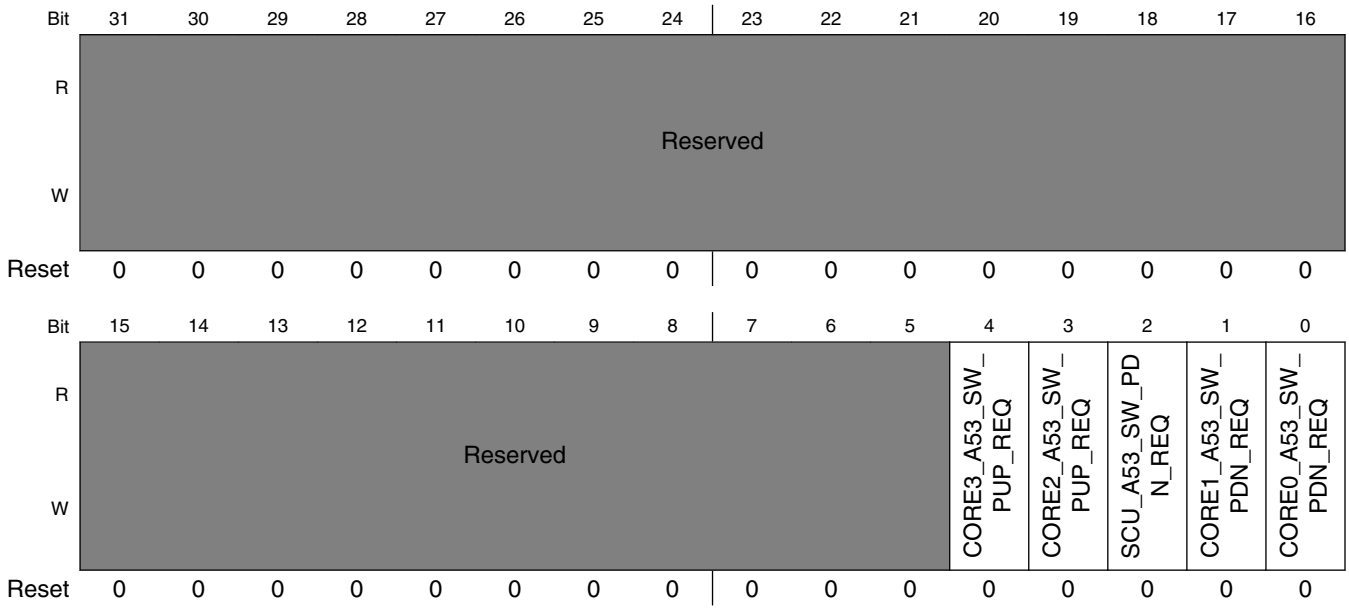
Table continues on the next page...

GPC_PU_PGC_SW_PUP_REQ field descriptions (continued)

Field	Description
11 MIPI_CSI1_SW_PUP_REQ	Software power up trigger for MIPI_CSI1
10 DISP_SW_PUP_REQ	Software power up trigger for DISP
9 HDMI_SW_PUP_REQ	Software power up trigger for HDMI
8 VPU_SW_PUP_REQ	Software power up trigger for VPU
7 GPU_SW_PUP_REQ	Software power up trigger for GPU
6 DDR2_SW_PUP_REQ	Software power up trigger for DDR2
5 DDR1_SW_PUP_REQ	Software power up trigger for DDR1
4 -	This field is reserved.
3 USB_OTG2_SW_PUP_REQ	Software power up trigger for USB_OTG2
2 USB_OTG1_SW_PUP_REQ	Software power up trigger for USB_OTG1
1 PCIE_SW_PUP_REQ	Software power up trigger for PCIE
0 MIPI_SW_PUP_REQ	Software power up trigger for MIPI

5.2.10.35 CPU PGC software down trigger (GPC_CPU_PGC_SW_PDN_REQ)

Address: 303A_0000h base + FCh offset = 303A_00FCh

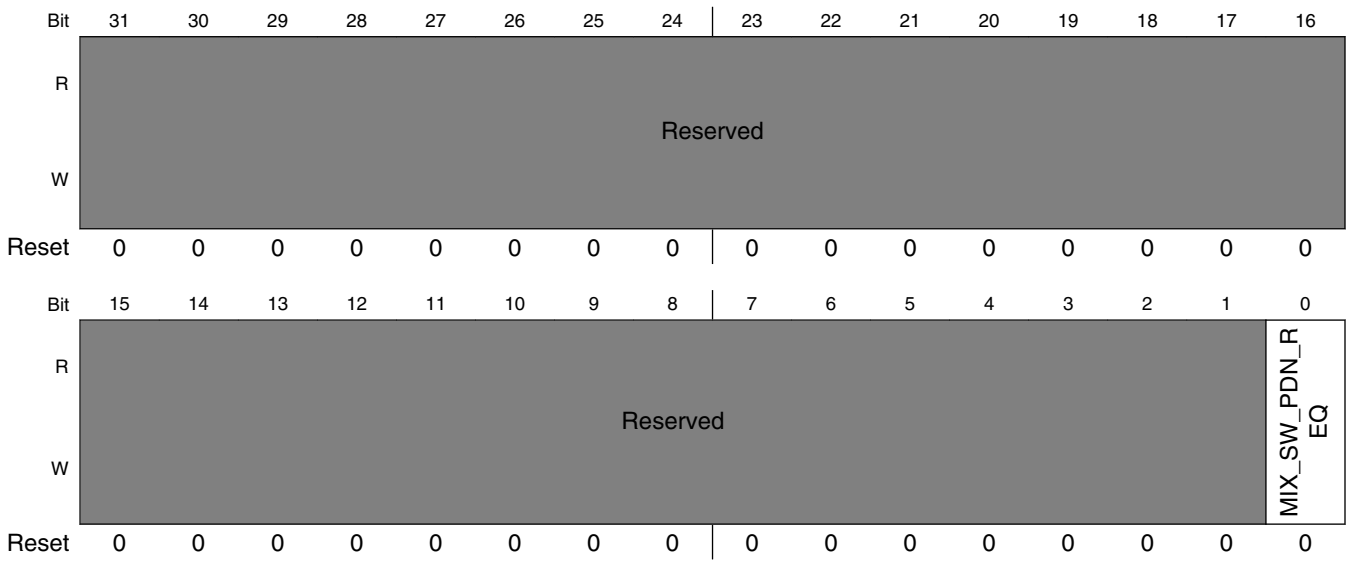


GPC_CPU_PGC_SW_PDN_REQ field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 CORE3_A53_SW_PUP_REQ	Software power up trigger for Core3 A53 PGC
3 CORE2_A53_SW_PUP_REQ	Software power up trigger for Core2 A53 PGC
2 SCU_A53_SW_PDN_REQ	Software power down trigger for SCU A53
1 CORE1_A53_SW_PDN_REQ	Software power down trigger for Core1 A53 PGC
0 CORE0_A53_SW_PDN_REQ	Software power down trigger for Core0 A53 PGC

5.2.10.36 MIX PGC software power down trigger (GPC_MIX_PGC_SW_PDN_REQ)

Address: 303A_0000h base + 100h offset = 303A_0100h

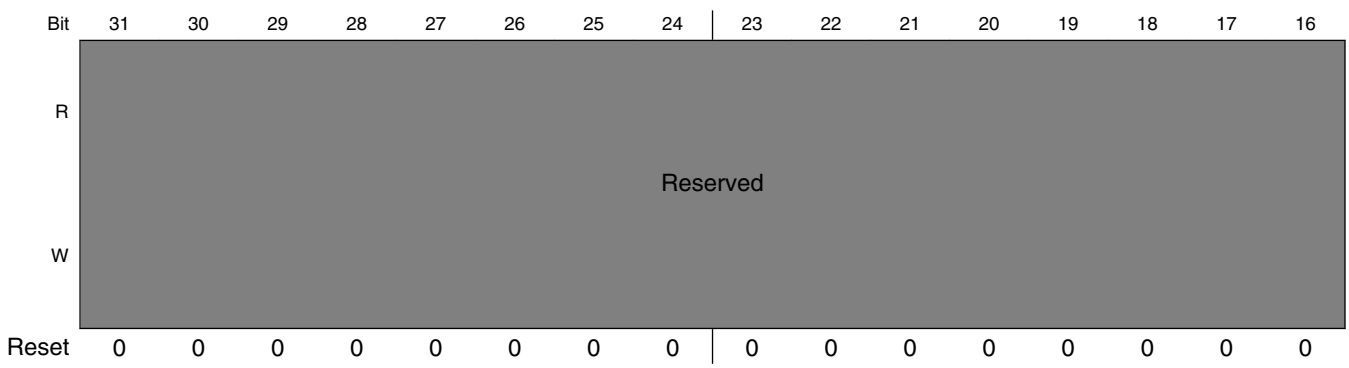


GPC_MIX_PGC_SW_PDN_REQ field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 MIX_SW_PDN_REQ	Software power down trigger for MIX PGC

5.2.10.37 PU PGC software down trigger (GPC_PU_PGC_SW_PDN_REQ)

Address: 303A_0000h base + 104h offset = 303A_0104h



Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		PCIE2_SW_PDN_REQ	MIPI_CSI2_SW_PDN_REQ	MIPI_CSI1_SW_PDN_REQ	DISP_SW_PDN_REQ	HDMI_SW_PDN_REQ	VPU_SW_PDN_REQ	GPU_SW_PDN_REQ	DDR2_SW_PDN_REQ	DDR1_SW_PDN_REQ	Reserved	USB_OTG2_SW_PDN_REQ	USB_OTG1_SW_PDN_REQ	PCIE_SW_PDN_REQ	MIPI_SW_PDN_REQ
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_PU_PGC_SW_PDN_REQ field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 PCIE2_SW_PDN_REQ	Software power down trigger for PCIE2
12 MIPI_CSI2_SW_PDN_REQ	Software power down trigger for MIPI_CSI2
11 MIPI_CSI1_SW_PDN_REQ	Software power down trigger for MIPI_CSI1
10 DISP_SW_PDN_REQ	Software power down trigger for DISP
9 HDMI_SW_PDN_REQ	Software power down trigger for HDMI
8 VPU_SW_PDN_REQ	Software power down trigger for VPU
7 GPU_SW_PDN_REQ	Software power down trigger for GPU
6 DDR2_SW_PDN_REQ	Software power down trigger for DDR2
5 DDR1_SW_PDN_REQ	Software power down trigger for DDR1
4 -	This field is reserved.
3 USB_OTG2_SW_PDN_REQ	Software power down trigger for USB_OTG2

Table continues on the next page...

GPC_PU_PGC_SW_PDN_REQ field descriptions (continued)

Field	Description
2 USB_OTG1_ SW_PDN_REQ	Software power down trigger for USB_OTG1
1 PCIE_SW_PDN_ REQ	Software power down trigger for PCIE
0 MIPI_SW_PDN_ REQ	Software power down trigger for MIPI

5.2.10.38 Basic Low power control register of A53 platform (GPC_LPCR_A53_BSC2)

Address: 303A_0000h base + 108h offset = 303A_0108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												LPM3		LPM2	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_LPCR_A53_BSC2 field descriptions

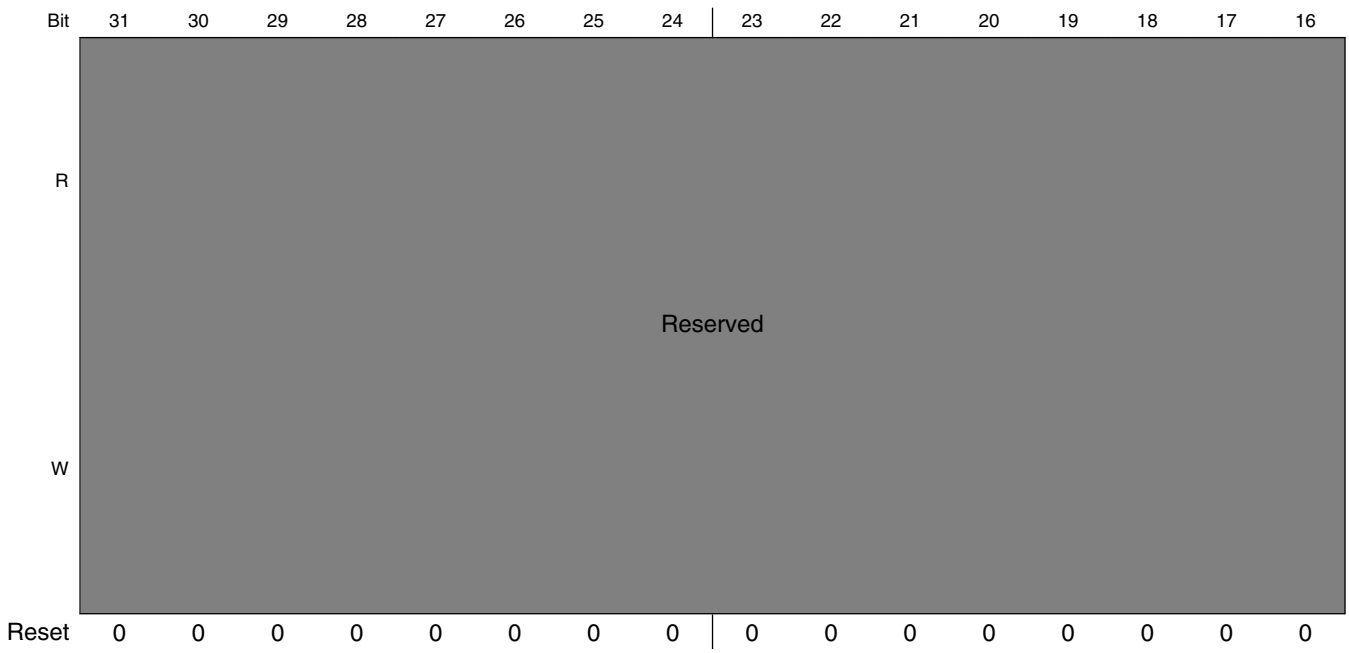
Field	Description
31–4 -	This field is reserved. Reserved
3–2 LPM3	CORE3 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved
LPM2	CORE2 Setting the low power mode that system will enter on next assertion of dsm_request signal. 00 Remain in RUN mode 01 Transfer to WAIT mode 10 Transfer to STOP mode 11 Reserved

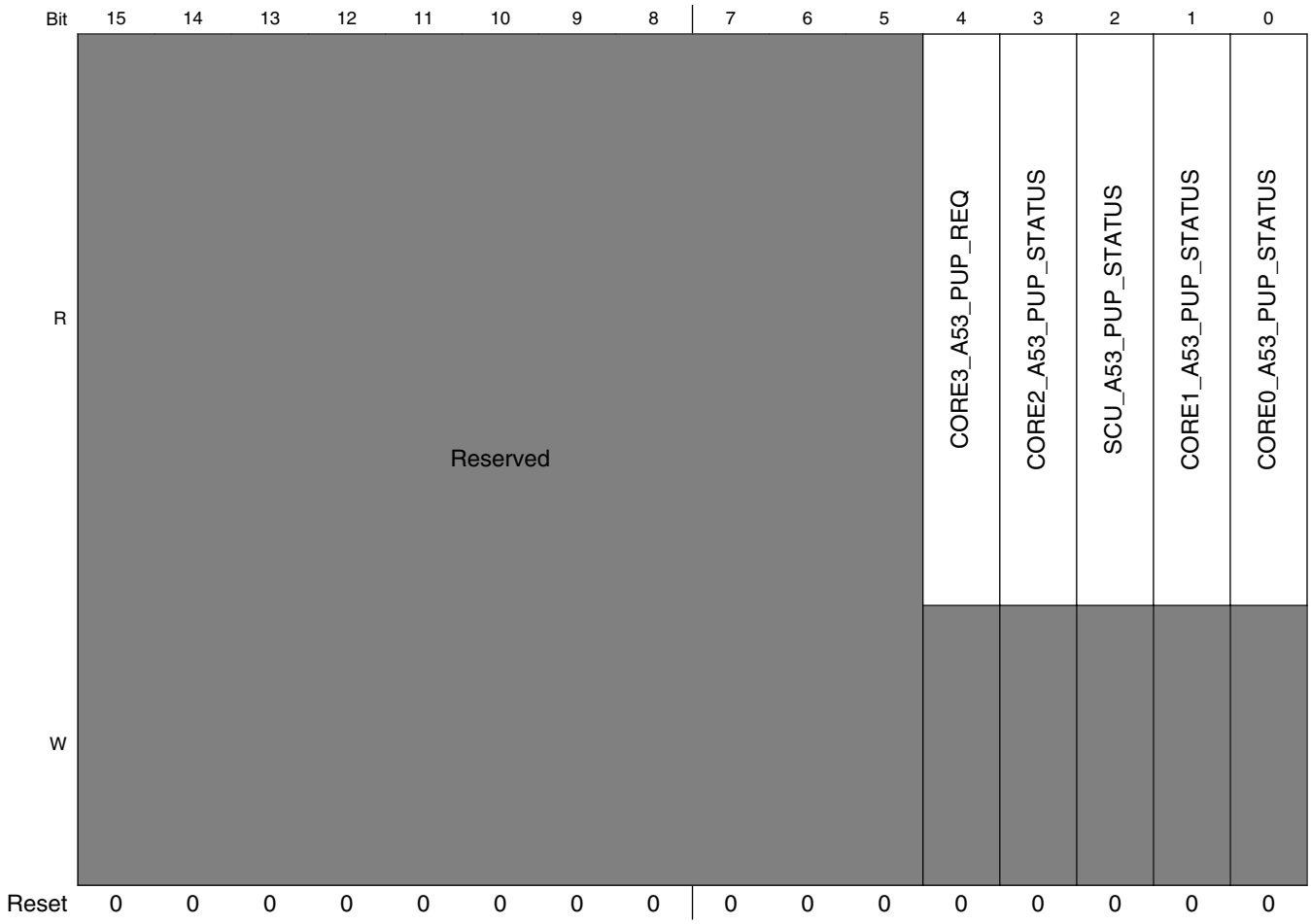
5.2.10.39 CPU PGC software up trigger status1 (GPC_CPU_PGC_PUP_STATUS1)

CPU_PGC_PUP_STATUS1 is a read only register, represents the results for power up software trigger for CPU type PGCs.

The field description is show in table below, the value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power down process. The relevant bit will be cleared after a success operation of power up software trigger for CPU type PGCs.

Address: 303A_0000h base + 130h offset = 303A_0130h





GPC_CPU_PGC_PUP_STATUS1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 CORE3_A53_PUP_REQ	
3 CORE2_A53_PUP_STATUS	
2 SCU_A53_PUP_STATUS	
1 CORE1_A53_PUP_STATUS	
0 CORE0_A53_PUP_STATUS	

5.2.10.40 A53 MIX software up trigger status register (GPC_A53_MIX_PGC_PUP_STATUS*n*)

A53_MIX_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from A53 platform to MIX type PGCs.

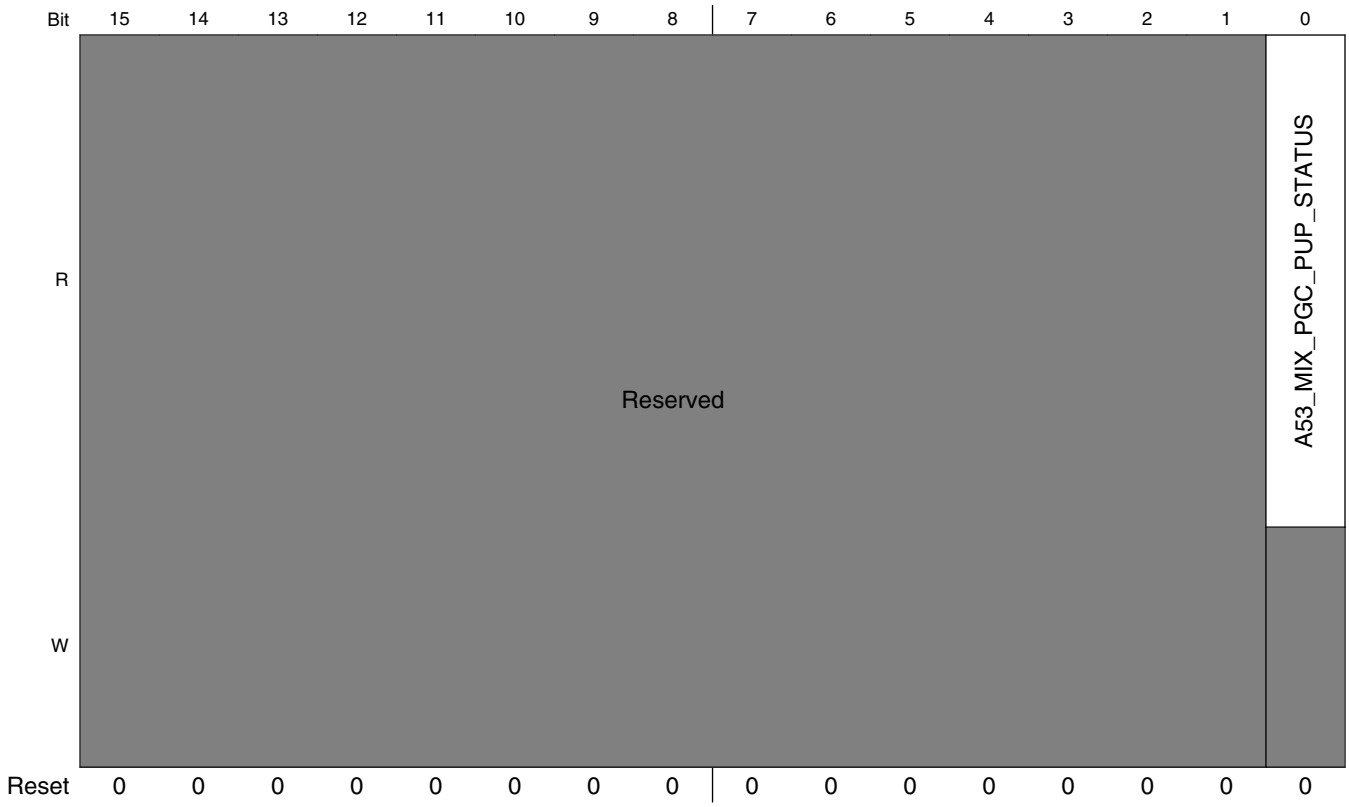
A53_MIX_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

A53_MIX_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

A53_MIX_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

Address: 303A_0000h base + 134h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_A53_MIX_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PGC_PUP_STATUS	

5.2.10.41 M4 MIX PGC software up trigger status register (GPC_M4_MIX_PGC_PUP_STATUS*n*)

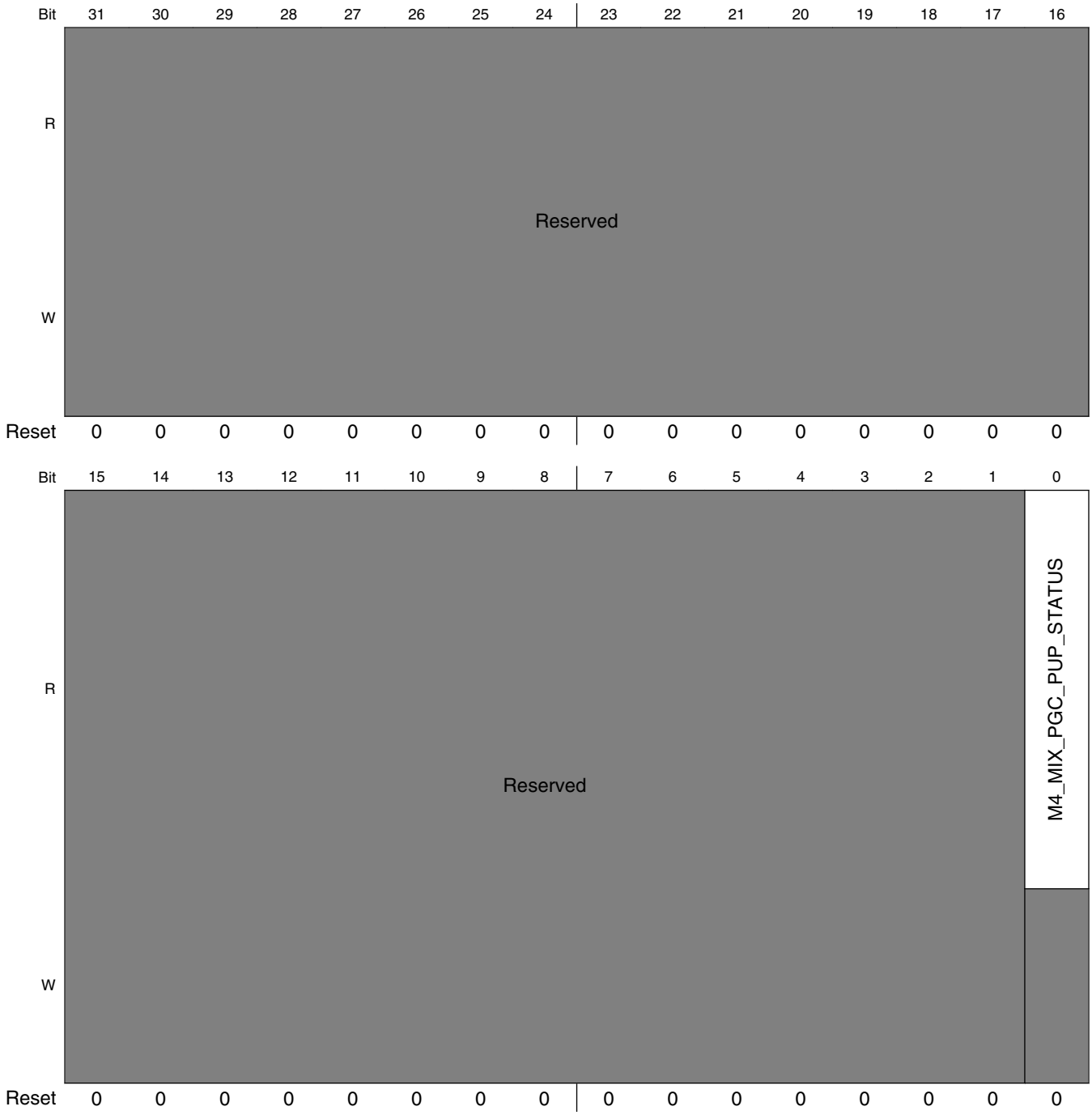
M4_MIX_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from M4 platform to MIX type PGCs.

M4_MIX_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

M4_MIX_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

M4_MIX_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for MIX type PGCs.

Address: 303A_0000h base + 140h offset + (4d × i), where i=0d to 2d



GPC_M4_MIX_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M4_MIX_PGC_ PUP_STATUS	

5.2.10.42 A53 PU software up trigger status register (GPC_A53_PU_PGC_PUP_STATUS*n*)

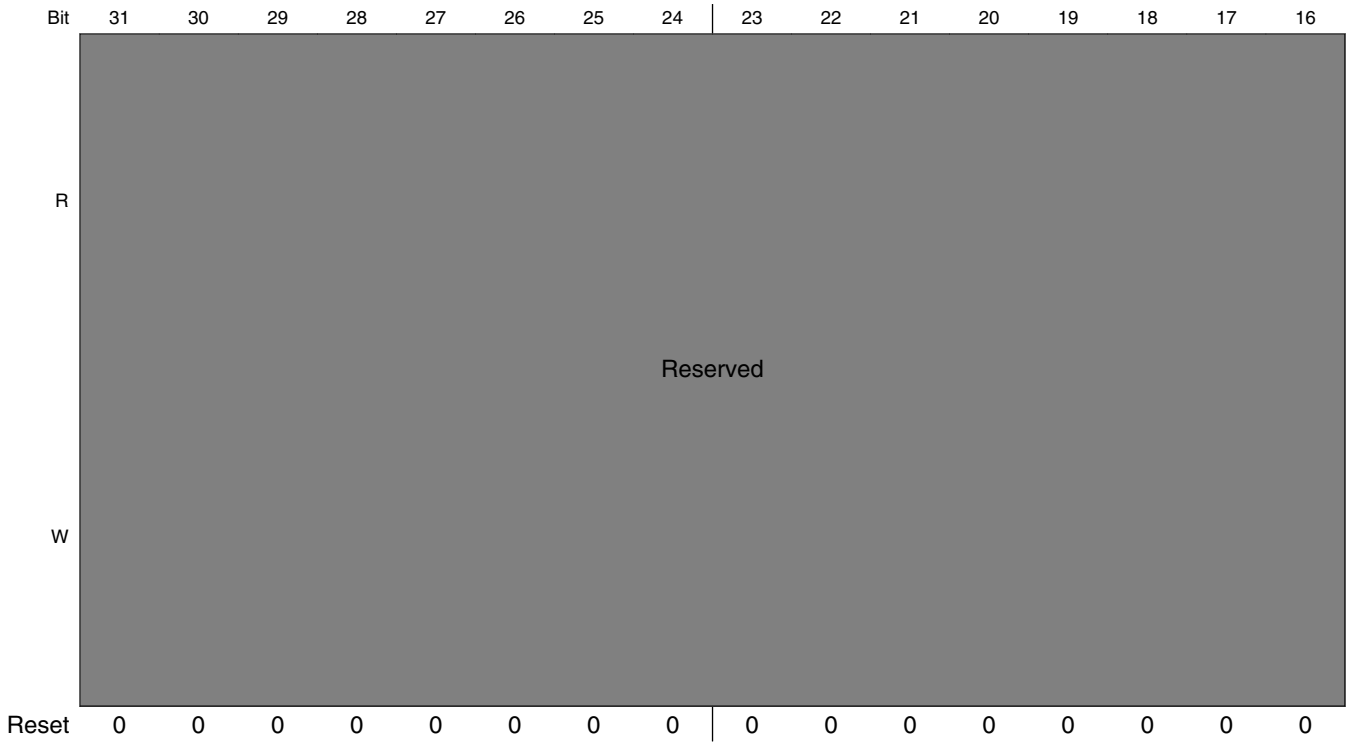
A53_PU_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from A53 platform to PU type PGCs.

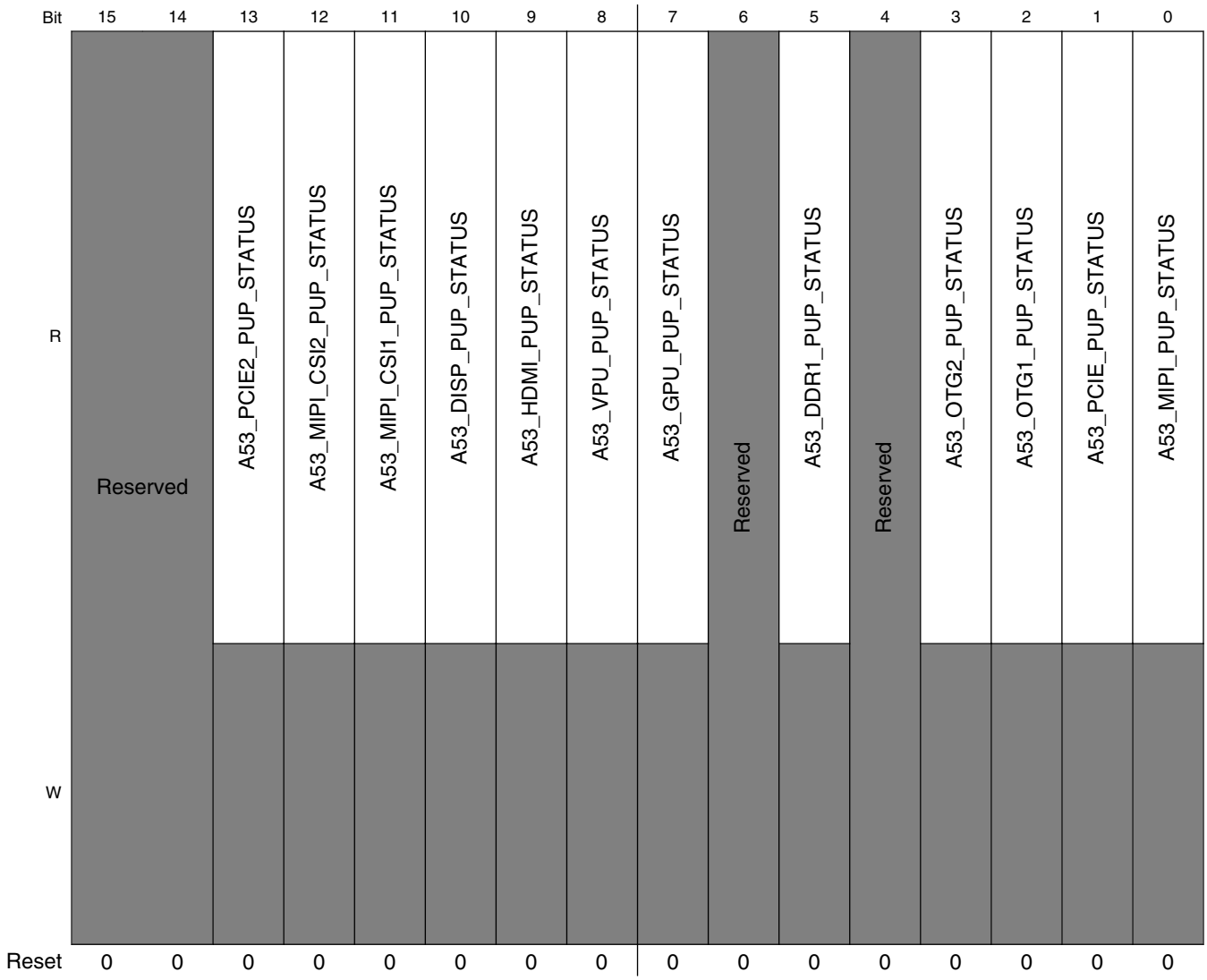
A53_PU_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

A53_PU_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

A53_PU_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

Address: 303A_0000h base + 14Ch offset + (4d × i), where i=0d to 2d





GPC_A53_PU_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 A53_PCIE2_PUP_STATUS	
12 A53_MIPI_CSI2_PUP_STATUS	
11 A53_MIPI_CSI1_PUP_STATUS	
10 A53_DISP_PUP_STATUS	

Table continues on the next page...

GPC_A53_PU_PGC_PUP_STATUS*n* field descriptions (continued)

Field	Description
9 A53_HDMI_PUP_STATUS	
8 A53_VPU_PUP_STATUS	
7 A53_GPU_PUP_STATUS	
6 -	This field is reserved.
5 A53_DDR1_PUP_STATUS	
4 -	This field is reserved.
3 A53_OTG2_PUP_STATUS	
2 A53_OTG1_PUP_STATUS	
1 A53_PCIE_PUP_STATUS	
0 A53_MIPI_PUP_STATUS	

5.2.10.43 M4 PU PGC software up trigger status register (GPC_M4_PU_PGC_PUP_STATUS*n*)

M4_PU_PGC_PUP_STATUSn (n = 0,1,2) are a read only register, represents the results for power up software trigger from M4 platform to PU type PGCs.

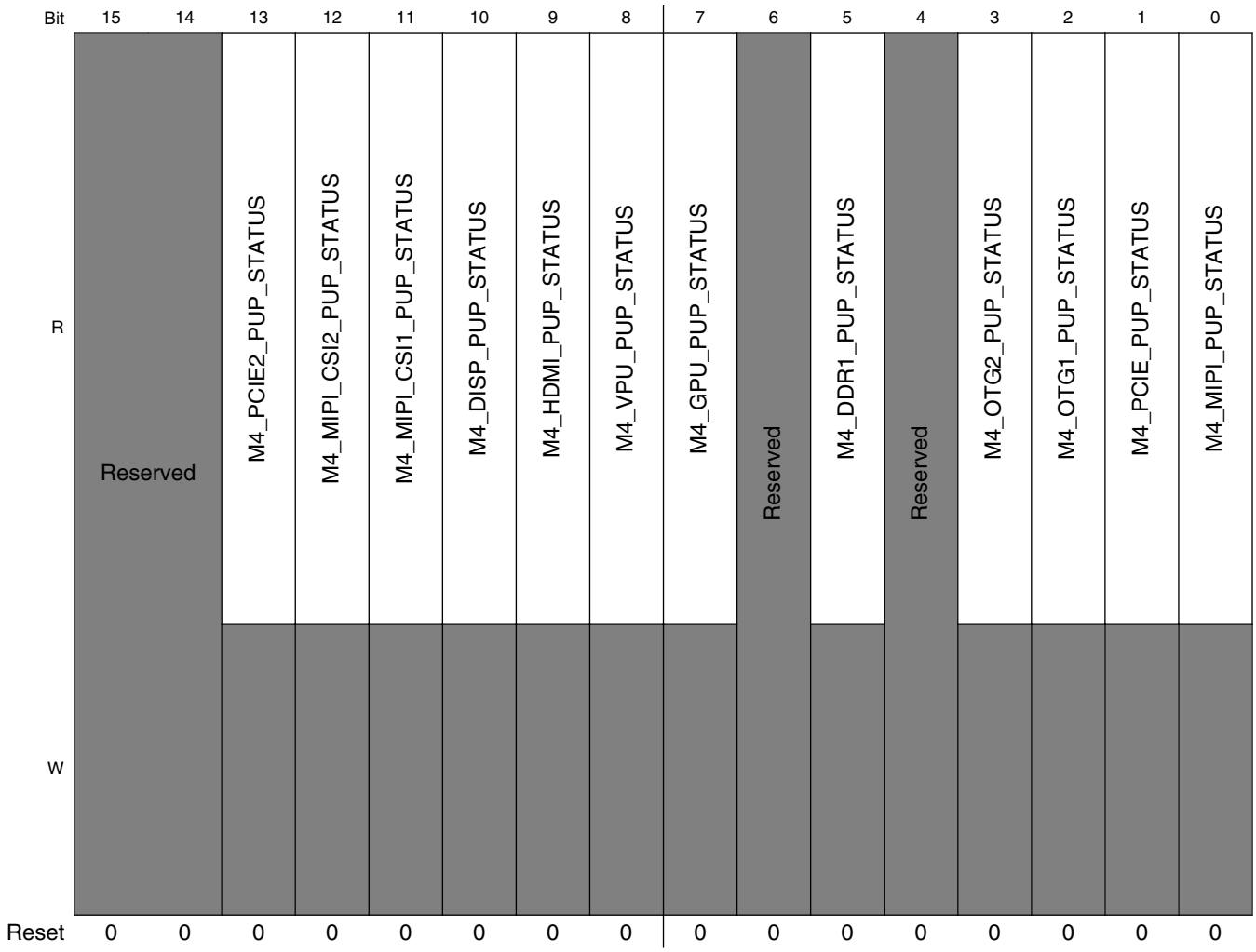
M4_PU_PGC_PUP_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

M4_PU_PGC_PUP_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

M4_PU_PGC_PUP_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power up software trigger for PU type PGCs.

Address: 303A_0000h base + 158h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_M4_PU_PGC_PUP_STATUS*n* field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 M4_PCIE2_PUP_STATUS	
12 M4_MIPI_CSI2_PUP_STATUS	
11 M4_MIPI_CSI1_PUP_STATUS	
10 M4_DISP_PUP_STATUS	
9 M4_HDMI_PUP_STATUS	

Table continues on the next page...

GPC_M4_PU_PGC_PUP_STATUS*n* field descriptions (continued)

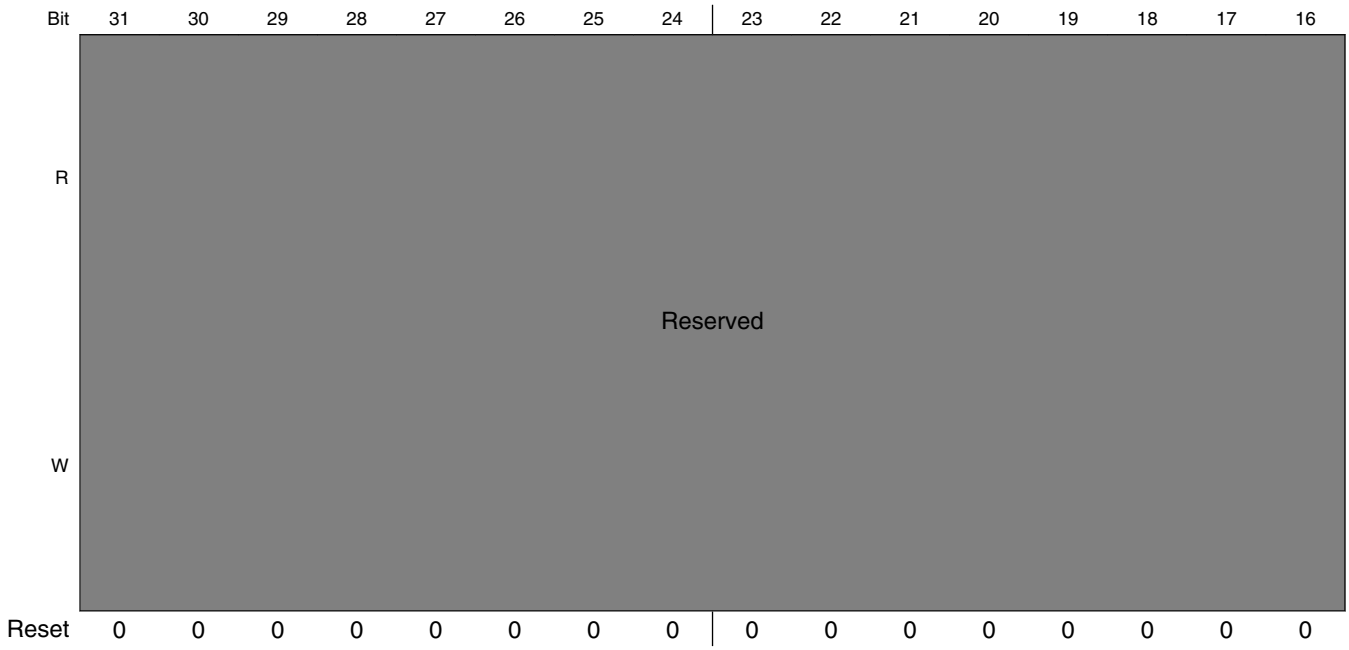
Field	Description
8 M4_VPU_PUP_STATUS	
7 M4_GPU_PUP_STATUS	
6 -	This field is reserved.
5 M4_DDR1_PUP_STATUS	
4 -	This field is reserved.
3 M4_OTG2_PUP_STATUS	
2 M4_OTG1_PUP_STATUS	
1 M4_PCIE_PUP_STATUS	
0 M4_MIPI_PUP_STATUS	

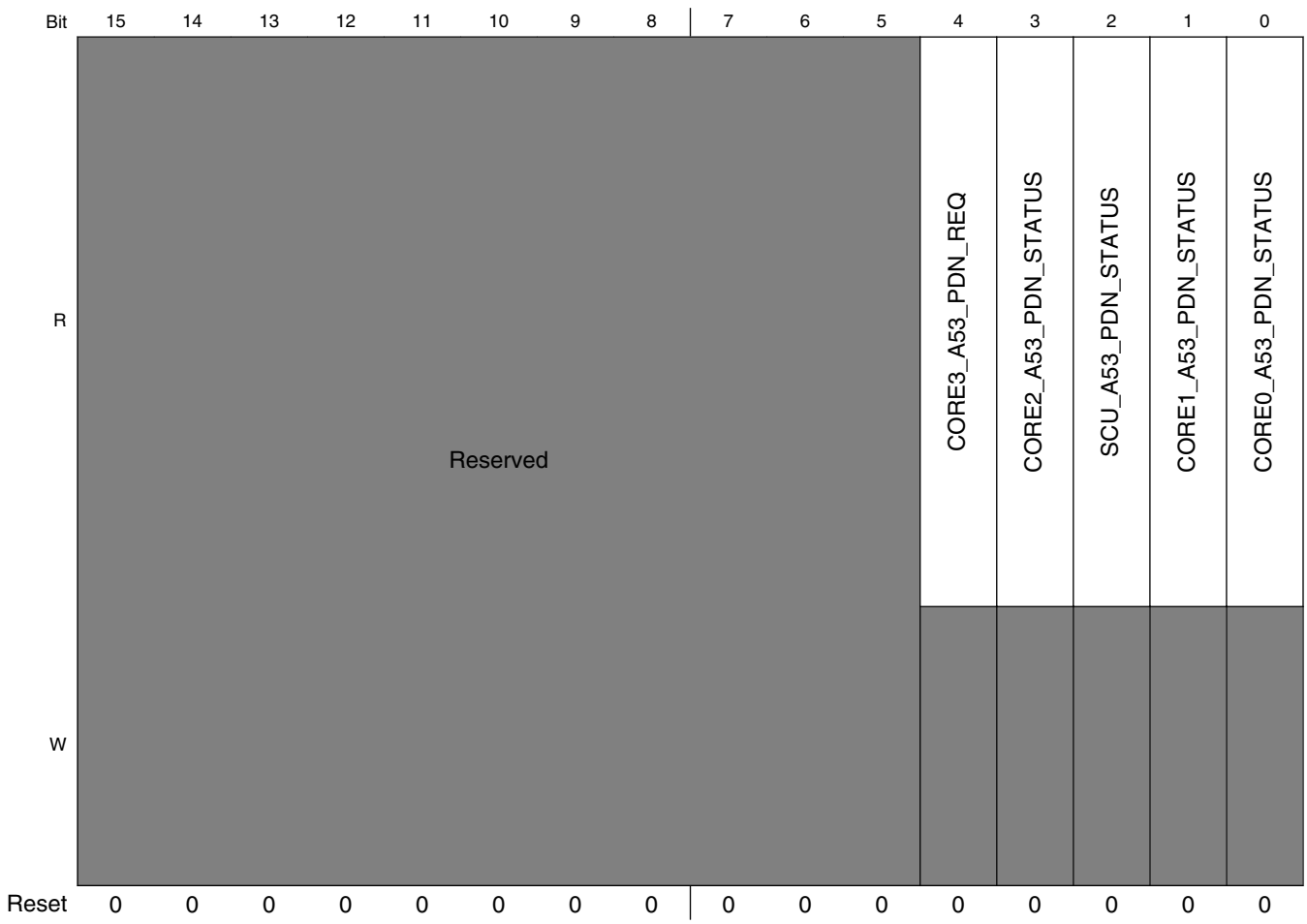
5.2.10.44 CPU PGC software dn trigger status1 (GPC_CPU_PGC_PDN_STATUS1)

CPU_PGC_PDN_STATUS1 is a read only register, represents the results for power DN software trigger for CPU type PGCs.

The field description is show in table below, the value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power down process. The relevant bit will be cleared after a success operation of power DN software trigger for CPU type PGCs.

Address: 303A_0000h base + 170h offset = 303A_0170h





GPC_CPU_PGC_PDN_STATUS1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 CORE3_A53_PDN_REQ	
3 CORE2_A53_PDN_STATUS	
2 SCU_A53_PDN_STATUS	
1 CORE1_A53_PDN_STATUS	
0 CORE0_A53_PDN_STATUS	

5.2.10.45 A53 MIX software down trigger status register (GPC_A53_MIX_PGC_PDN_STATUS*n*)

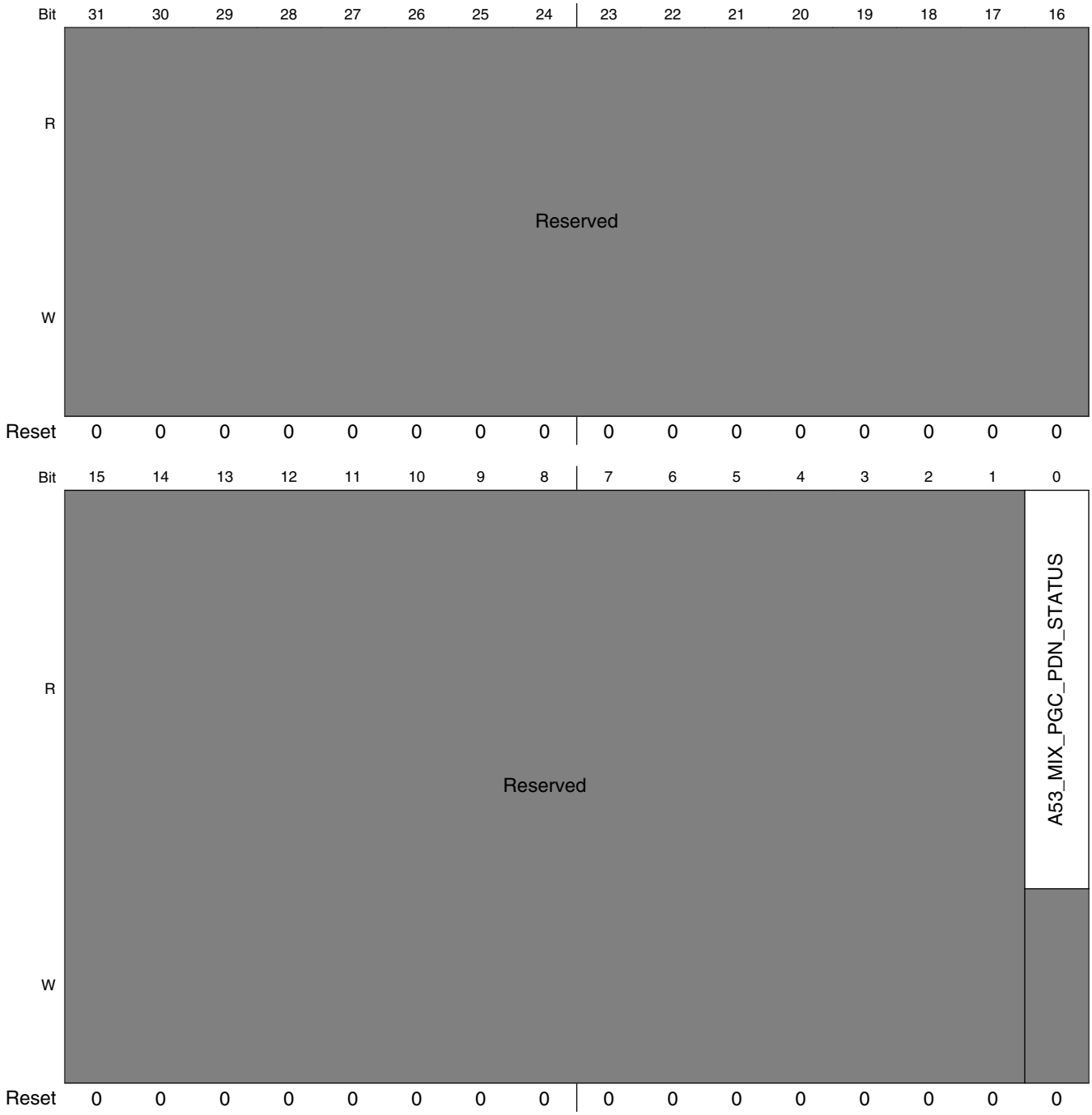
A53_MIX_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power down software trigger from A53 platform to MIX type PGCs.

A53_MIX_PGC_PDN_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

A53_MIX_PGC_PDN_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

A53_MIX_PGC_PDN_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

Address: 303A_0000h base + 174h offset + (4d × i), where i=0d to 2d



GPC_A53_MIX_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PGC_PDN_STATUS	

5.2.10.46 M4 MIX PGC software power down trigger status register (GPC_M4_MIX_PGC_PDN_STATUS*n*)

M4_MIX_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power down software trigger from M4 platform to MIX type PGCs.

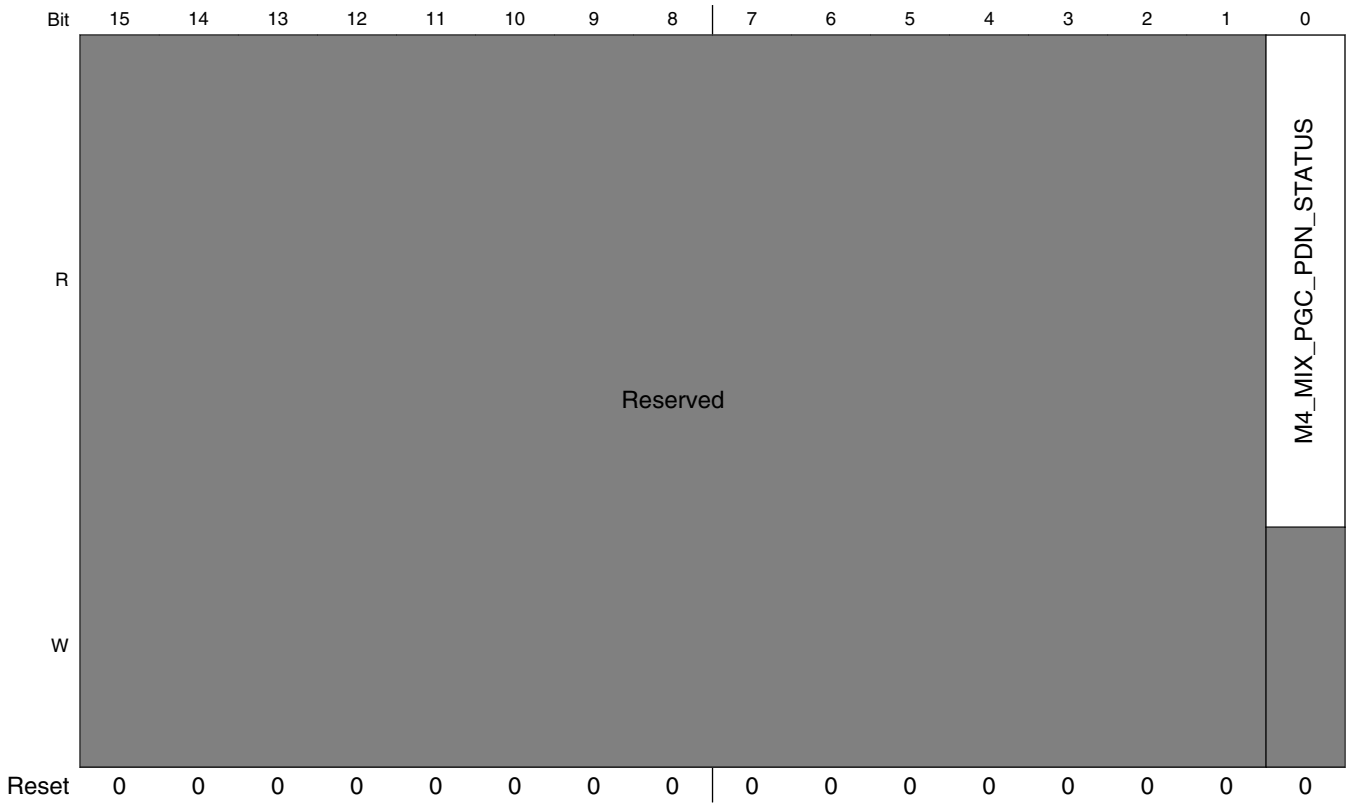
M4_MIX_PGC_PDN_STATUS0: value of “1’b1” represent the software power up trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

M4_MIX_PGC_PDN_STATUS1: value of “1’b1” represent the software power up trigger failed because the relevant PGC is in a power up process. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

M4_MIX_PGC_PDN_STATUS2: value of “1’b1” represent the software power up trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power down software trigger for MIX type PGCs.

Address: 303A_0000h base + 180h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_M4_MIX_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M4_MIX_PGC_ PDN_STATUS	

5.2.10.47 A53 PU PGC software down trigger status (GPC_A53_PU_PGC_PDN_STATUS*n*)

A53_PU_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power DN software trigger from A53 platform to PU type PGCs.

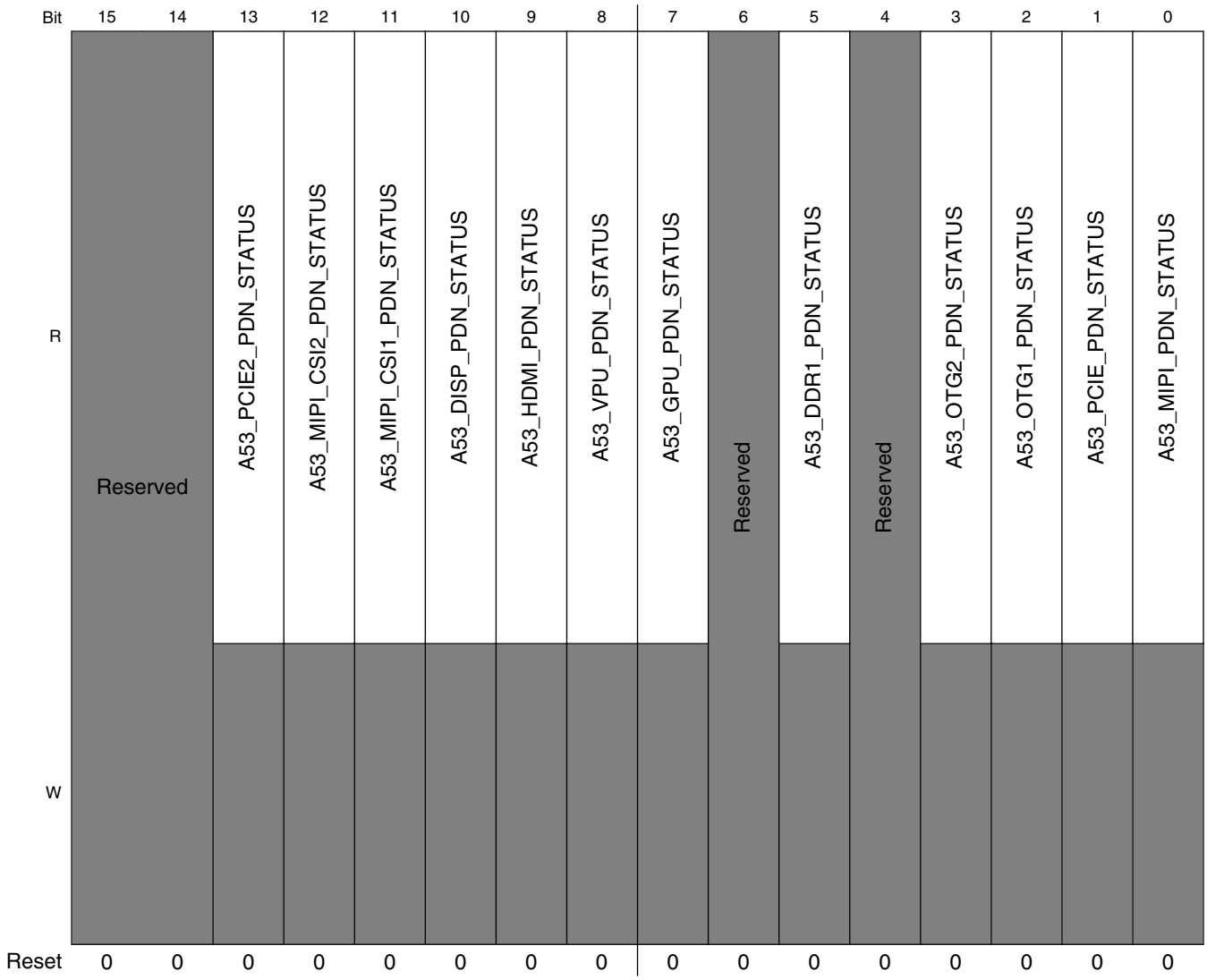
A53_PU_PGC_PDN_STATUS0: value of “1’b1” represent the software power DN trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

A53_PU_PGC_PDN_STATUS1: value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power DN process. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

A53_PU_PGC_PDN_STATUS2: value of “1’b1” represent the software power DN trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

Address: 303A_0000h base + 18Ch offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_A53_PU_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 A53_PCIE2_PDN_STATUS	
12 A53_MIPI_CSI2_PDN_STATUS	
11 A53_MIPI_CSI1_PDN_STATUS	
10 A53_DISP_PDN_STATUS	

Table continues on the next page...

GPC_A53_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
9 A53_HDMI_PDN_STATUS	
8 A53_VPU_PDN_STATUS	
7 A53_GPU_PDN_STATUS	
6 -	This field is reserved.
5 A53_DDR1_PDN_STATUS	
4 -	This field is reserved.
3 A53_OTG2_PDN_STATUS	
2 A53_OTG1_PDN_STATUS	
1 A53_PCIE_PDN_STATUS	
0 A53_MIPI_PDN_STATUS	

5.2.10.48 M4 PU PGC software down trigger status (GPC_M4_PU_PGC_PDN_STATUS*n*)

M4_PU_PGC_PDN_STATUSn (n = 0,1,2) are a read only register, represents the results for power DN software trigger from M4 platform to PU type PGCs.

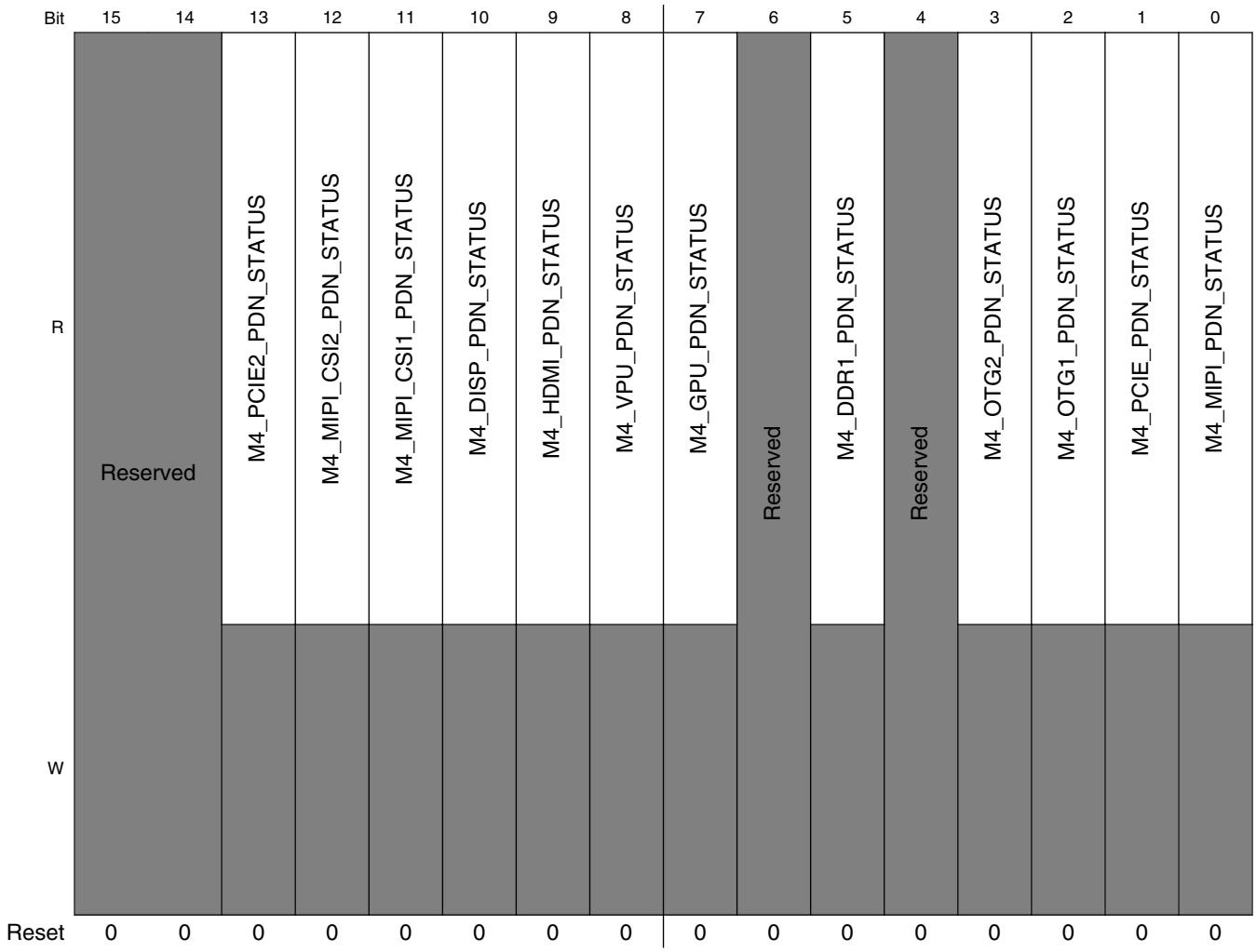
M4_PU_PGC_PDN_STATUS0: value of “1’b1” represent the software power DN trigger failed because domain control condition. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

M4_PU_PGC_PDN_STATUS1: value of “1’b1” represent the software power DN trigger failed because the relevant PGC is in a power DN process. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

M4_PU_PGC_PDN_STATUS2: value of “1’b1” represent the software power DN trigger failed because time slot control is busy. The relevant bit will be cleared after a success operation of power DN software trigger for PU type PGCs.

Address: 303A_0000h base + 198h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_M4_PU_PGC_PDN_STATUS*n* field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 M4_PCIE2_PDN_STATUS	
12 M4_MIPI_CSI2_PDN_STATUS	
11 M4_MIPI_CSI1_PDN_STATUS	
10 M4_DISP_PDN_STATUS	
9 M4_HDMI_PDN_STATUS	

Table continues on the next page...

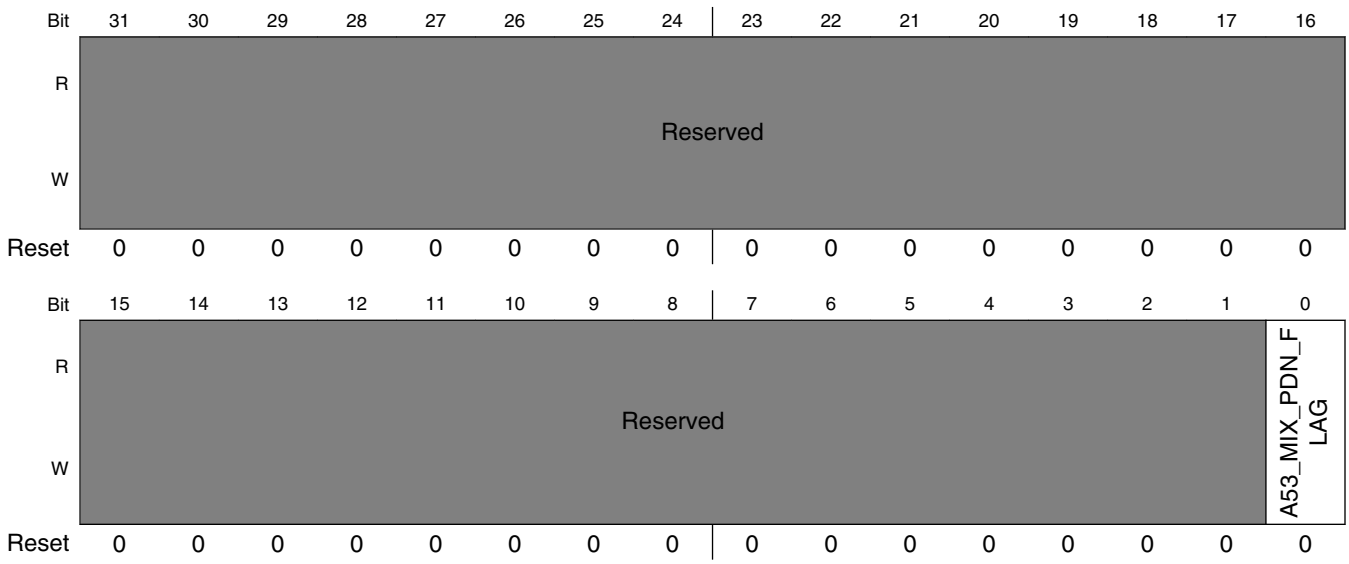
GPC_M4_PU_PGC_PDN_STATUS*n* field descriptions (continued)

Field	Description
8 M4_VPU_PDN_STATUS	
7 M4_GPU_PDN_STATUS	
6 -	This field is reserved.
5 M4_DDR1_PDN_STATUS	
4 -	This field is reserved.
3 M4_OTG2_PDN_STATUS	
2 M4_OTG1_PDN_STATUS	
1 M4_PCIE_PDN_STATUS	
0 M4_MIPI_PDN_STATUS	

5.2.10.49 A53 MIX PDN FLG (GPC_A53_MIX_PDN_FLG)

This is flag bit relevant domain control, represents A53 CPU platform wants to power down MIX PGC. The register can only be accessed by A53 platform.

Address: 303A_0000h base + 1B0h offset = 303A_01B0h



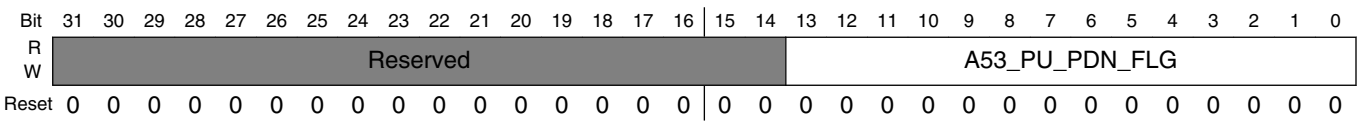
GPC_A53_MIX_PDN_FLG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 A53_MIX_PDN_FLAG	A53 MIX power-down flag

5.2.10.50 A53 PU PDN FLG (GPC_A53_PU_PDN_FLG)

The register field is show in the table below. The 1’b1 represents A53 CPU platform wants to power down certain PU PGC. The register is a read only register. The register bits will be set when corresponding A53 software power down trigger happens and will be clear when corresponding A53 software power up trigger happens.

Address: 303A_0000h base + 1B4h offset = 303A_01B4h



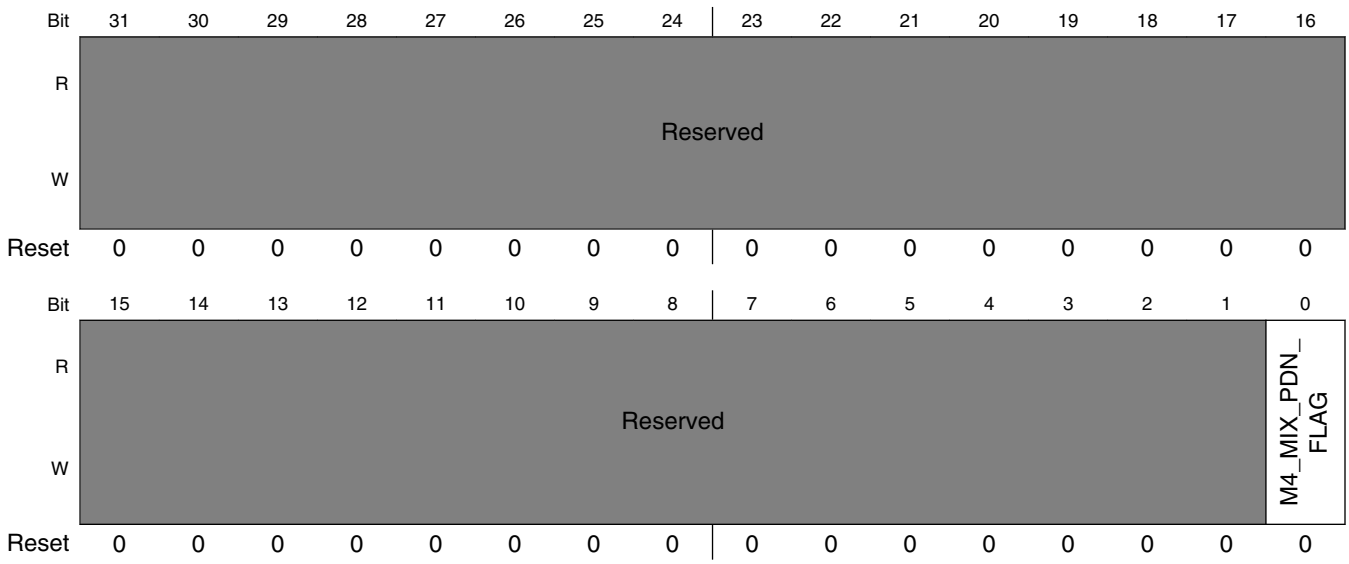
GPC_A53_PU_PDN_FLG field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
A53_PU_PDN_FLG	A53 PGC power-down flag

5.2.10.51 M4 MIX PDN FLG (GPC_M4_MIX_PDN_FLG)

This is flag bit relevant domain control, represents M4 CPU platform wants to power down MIX PGC. The register can only be accessed by M4 platform.

Address: 303A_0000h base + 1B8h offset = 303A_01B8h



GPC_M4_MIX_PDN_FLG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 M4_MIX_PDN_FLAG	M4_MIX power-down flag

5.2.10.52 M4 PU PDN FLG (GPC_M4_PU_PDN_FLG)

The register field is show in the table below. The 1'b1 represents M4 CPU platform wants to power down certain PU PGC. The register is a read only register. The register bits will be set when corresponding M4 software power down trigger happens and will be clear when corresponding M4 software power up trigger happens.

Address: 303A_0000h base + 1BCh offset = 303A_01BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																M4_PU_PDN_FLG																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_M4_PU_PDN_FLG field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
M4_PU_PDN_FLG	M4 power-down flag

5.2.10.53 IRQ masking register 1 of A53 core2 (GPC_IMR1_CORE2_A53)

The four IMRn_CORE2_A53 (n = 1,2,3,4) registers are used as interrupt mask for A53 core2.

Address: 303A_0000h base + 1C0h offset = 303A_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE2_A53 field descriptions

Field	Description
IMR1_CORE2_A53	A53 core2 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.54 IRQ masking register 2 of A53 core2 (GPC_IMR2_CORE2_A53)

Address: 303A_0000h base + 1C4h offset = 303A_01C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE2_A53 field descriptions

Field	Description
IMR2_CORE2_A53	A53 core2 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.55 IRQ masking register 3 of A53 core2 (GPC_IMR3_CORE2_A53)

Address: 303A_0000h base + 1C8h offset = 303A_01C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE2_A53 field descriptions

Field	Description
IMR3_CORE2_A53	A53 core2 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.56 IRQ masking register 4 of A53 core2 (GPC_IMR4_CORE2_A53)

Address: 303A_0000h base + 1CCh offset = 303A_01CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE2_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE2_A53 field descriptions

Field	Description
IMR4_CORE2_A53	A53 core2 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.57 IRQ masking register 1 of A53 core3 (GPC_IMR1_CORE3_A53)

The four IMRn_CORE2_A53 (n = 1,2,3,4) registers are used as interrupt mask for A53 core3.

Address: 303A_0000h base + 1D0h offset = 303A_01D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR1_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR1_CORE3_A53 field descriptions

Field	Description
IMR1_CORE3_A53	A53 core3 IRQ[31:0] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.58 IRQ masking register 2 of A53 core3 (GPC_IMR2_CORE3_A53)

Address: 303A_0000h base + 1D4h offset = 303A_01D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR2_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR2_CORE3_A53 field descriptions

Field	Description
IMR2_CORE3_A53	A53 core3 IRQ[63:32] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.59 IRQ masking register 3 of A53 core3 (GPC_IMR3_CORE3_A53)

Address: 303A_0000h base + 1D8h offset = 303A_01D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR3_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR3_CORE3_A53 field descriptions

Field	Description
IMR3_CORE3_A53	A53 core3 IRQ[95:64] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.60 IRQ masking register 4 of A53 core3 (GPC_IMR4_CORE3_A53)

Address: 303A_0000h base + 1DCh offset = 303A_01DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR4_CORE3_A53																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_IMR4_CORE3_A53 field descriptions

Field	Description
IMR4_CORE3_A53	A53 core3 IRQ[127:96] masking bits: 0 IRQ not masked 1 IRQ masked

5.2.10.61 PGC acknowledge signal selection of A53 platform for PUs (GPC_ACK_SEL_A53_PU)

Address: 303A_0000h base + 1E0h offset = 303A_01E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCIE2_PGC_PUP_ACK	MIPI_CSI2_PGC_PUP_ACK	MIPI_CSI1_PGC_PUP_ACK	DISP_PGC_PUP_ACK	HDMI_PGC_PUP_ACK	VPU_PGC_PUP_ACK	GPU_PGC_PUP_ACK	DDR2_PGC_PUP_ACK	DDR1_PGC_PUP_ACK	Reserved	USB_OTG2_PGC_PUP_ACK	USB_OTG1_PGC_PUP_ACK	PCIE_PGC_PUP_ACK	MIPI_PGC_PUP_ACK	Reserved	MF_PGC_PUP_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE2_PGC_PDN_ACK	MIPI_CSI2_PGC_PDN_ACK	MIPI_CSI1_PGC_PDN_ACK	DISP_PGC_PDN_ACK	HDMI_PGC_PDN_ACK	VPU_PGC_PDN_ACK	GPU_PGC_PDN_ACK	DDR2_PGC_PDN_ACK	DDR1_PGC_PDN_ACK	Reserved	USB_OTG2_PGC_PDN_ACK	USB_OTG1_PGC_PDN_ACK	PCIE_PGC_PDN_ACK	MIPI_PGC_PDN_ACK	Reserved	MF_PGC_PDN_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ACK_SEL_A53_PU field descriptions

Field	Description
31 PCIE2_PGC_PUP_ACK	Select power down acknowledge signal of PCIE2 PGC as the power up acknowledge for A53 LPM.
30 MIPI_CSI2_PGC_PUP_ACK	Select power down acknowledge signal of MIPI_CSI2 PGC as the power up acknowledge for A53 LPM.
29 MIPI_CSI1_PGC_PUP_ACK	Select power down acknowledge signal of MIPI_CSI1 PGC as the power up acknowledge for A53 LPM.
28 DISP_PGC_PUP_ACK	Select power down acknowledge signal of DISP PGC as the power up acknowledge for A53 LPM.
27 HDMI_PGC_PUP_ACK	Select power down acknowledge signal of HDMI PGC as the power up acknowledge for A53 LPM.
26 VPU_PGC_PUP_ACK	Select power down acknowledge signal of VPU PGC as the power up acknowledge for A53 LPM.
25 GPU_PGC_PUP_ACK	Select power down acknowledge signal of GPU PGC as the power up acknowledge for A53 LPM.
24 DDR2_PGC_PUP_ACK	Select power down acknowledge signal of DDR2 PGC as the power up acknowledge for A53 LPM.
23 DDR1_PGC_PUP_ACK	Select power down acknowledge signal of DDR1 PGC as the power up acknowledge for A53 LPM.
22 -	This field is reserved.
21 USB_OTG2_PGC_PUP_ACK	Select power down acknowledge signal of USB_OTG2 PGC as the power up acknowledge for A53 LPM.
20 USB_OTG1_PGC_PUP_ACK	Select power down acknowledge signal of USB_OTG1 PGC as the power up acknowledge for A53 LPM.

Table continues on the next page...

GPC_ACK_SEL_A53_PU field descriptions (continued)

Field	Description
19 PCIE_PGC_ PUP_ACK	Select power down acknowledge signal of PCIE PGC as the power up acknowledge for A53 LPM.
18 MIPI_PGC_PUP_ ACK	Select power down acknowledge signal of MIPI PGC as the power up acknowledge for A53 LPM.
17 -	This field is reserved.
16 MF_PGC_PUP_ ACK	Select power down acknowledge signal of MIX PGC as the power up acknowledge for A53 LPM.
15 PCIE2_PGC_PD N_ACK	Select power down acknowledge signal of PCIE2 PGC as the power down acknowledge for A53 LPM.
14 MIPI_CSI2_PGC _PDN_ACK	Select power down acknowledge signal of MIPI_CSI2 PGC as the power down acknowledge for A53 LPM.
13 MIPI_CSI1_PGC _PDN_ACK	Select power down acknowledge signal of MIPI_CSI1 PGC as the power down acknowledge for A53 LPM.
12 DISP_PGC_ PDN_ACK	Select power down acknowledge signal of DISP PGC as the power down acknowledge for A53 LPM.
11 HDMI_PGC_PDN _ACK	Select power down acknowledge signal of HDMI PGC as the power down acknowledge for A53 LPM.
10 VPU_PGC_ PDN_ACK	Select power down acknowledge signal of VPU PGC as the power down acknowledge for A53 LPM.
9 GPU_PGC_ PDN_ACK	Select power down acknowledge signal of GPU PGC as the power down acknowledge for A53 LPM.
8 DDR2_PGC_PD N_ACK	Select power down acknowledge signal of DDR2 PGC as the power down acknowledge for A53 LPM.
7 DDR1_PGC_ PDN_ACK	Select power down acknowledge signal of DDR1 PGC as the power down acknowledge for A53 LPM.
6 -	This field is reserved. Reserved
5 USB_OTG2_ PGC_PDN_ACK	Select power down acknowledge signal of USB_OTG2 PGC as the power down acknowledge for A53 LPM.
4 USB_OTG1_ PGC_PDN_ACK	Select power down acknowledge signal of USB_OTG1 PGC as the power down acknowledge for A53 LPM.
3 PCIE_PGC_ PDN_ACK	Select power down acknowledge signal of PCIE PGC as the power down acknowledge for A53 LPM.

Table continues on the next page...

GPC_ACK_SEL_A53_PU field descriptions (continued)

Field	Description
2 MIPI_PGC_PDN_ACK	Select power down acknowledge signal of MIPI PGC as the power down acknowledge for A53 LPM.
1 -	This field is reserved. Reserved
0 MF_PGC_PDN_ACK	Select power down acknowledge signal of MIX PGC as the power down acknowledge for A53 LPM.

5.2.10.62 PGC acknowledge signal selection of M4 platform for PUs (GPC_ACK_SEL_M4_PU)

Address: 303A_0000h base + 1E4h offset = 303A_01E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCIE2_PGC_PUP_ACK	MIPI_CSI2_PGC_PUP_ACK	MIPI_CSI1_PGC_PUP_ACK	DISP_PGC_PUP_ACK	HDMI_PGC_PUP_ACK	VPU_PGC_PUP_ACK	GPU_PGC_PUP_ACK	DDR2_PGC_PUP_ACK	DDR1_PGC_PUP_ACK	Reserved	USB_OTG2_PGC_PUP_ACK	USB_OTG1_PGC_PUP_ACK	PCIE_PGC_PUP_ACK	MIPI_PGC_PUP_ACK	Reserved	MF_PGC_PUP_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE2_PGC_PDN_ACK	MIPI_CSI2_PGC_PDN_ACK	MIPI_CSI1_PGC_PDN_ACK	DISP_PGC_PDN_ACK	HDMI_PGC_PDN_ACK	VPU_PGC_PDN_ACK	GPU_PGC_PDN_ACK	DDR2_PGC_PDN_ACK	DDR1_PGC_PDN_ACK	Reserved	USB_OTG2_PGC_PDN_ACK	USB_OTG1_PGC_PDN_ACK	PCIE_PGC_PDN_ACK	MIPI_PGC_PDN_ACK	Reserved	MF_PGC_PDN_ACK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_ACK_SEL_M4_PU field descriptions

Field	Description
31 PCIE2_PGC_PUP_ACK	Select power down acknowledge signal of PCIE2 PGC as the power up acknowledge for M4 LPM.

Table continues on the next page...

GPC_ACK_SEL_M4_PU field descriptions (continued)

Field	Description
30 MIPI_CSI2_PGC_PUP_ACK	Select power down acknowledge signal of MIPI_CSI2 PGC as the power up acknowledge for M4 LPM.
29 MIPI_CSI1_PGC_PUP_ACK	Select power down acknowledge signal of MIPI_CSI1 PGC as the power up acknowledge for M4 LPM.
28 DISP_PGC_PUP_ACK	Select power down acknowledge signal of DISP PGC as the power up acknowledge for M4 LPM.
27 HDMI_PGC_PUP_ACK	Select power down acknowledge signal of HDMI PGC as the power up acknowledge for M4 LPM.
26 VPU_PGC_PUP_ACK	Select power down acknowledge signal of VPU PGC as the power up acknowledge for M4 LPM.
25 GPU_PGC_PUP_ACK	Select power down acknowledge signal of GPU PGC as the power up acknowledge for M4 LPM.
24 DDR2_PGC_PUP_ACK	Select power down acknowledge signal of DDR2 PGC as the power up acknowledge for M4 LPM.
23 DDR1_PGC_PUP_ACK	Select power down acknowledge signal of DDR1 PGC as the power up acknowledge for M4 LPM.
22 -	This field is reserved.
21 USB_OTG2_PGC_PUP_ACK	Select power down acknowledge signal of USB_OTG2 PGC as the power up acknowledge for M4 LPM.
20 USB_OTG1_PGC_PUP_ACK	Select power down acknowledge signal of USB_OTG1 PGC as the power up acknowledge for M4 LPM.
19 PCIE_PGC_PUP_ACK	Select power down acknowledge signal of PCIE PGC as the power up acknowledge for M4 LPM.
18 MIPI_PGC_PUP_ACK	Select power down acknowledge signal of MIPI PGC as the power up acknowledge for M4 LPM.
17 -	This field is reserved.
16 MF_PGC_PUP_ACK	Select power down acknowledge signal of MIX PGC as the power up acknowledge for M4 LPM.
15 PCIE2_PGC_PDN_ACK	Select power down acknowledge signal of PCIE2 PGC as the power down acknowledge for M4 LPM.
14 MIPI_CSI2_PGC_PDN_ACK	Select power down acknowledge signal of MIPI_CSI2 PGC as the power down acknowledge for M4 LPM.

Table continues on the next page...

GPC_ACK_SEL_M4_PU field descriptions (continued)

Field	Description
13 MIPI_CSI1_PGC_PDN_ACK	Select power down acknowledge signal of MIPI_CSI1 PGC as the power down acknowledge for M4 LPM.
12 DISP_PGC_PDN_ACK	Select power down acknowledge signal of DISP PGC as the power down acknowledge for M4 LPM.
11 HDMI_PGC_PDN_ACK	Select power down acknowledge signal of HDMI PGC as the power down acknowledge for M4 LPM.
10 VPU_PGC_PDN_ACK	Select power down acknowledge signal of VPU PGC as the power down acknowledge for M4 LPM.
9 GPU_PGC_PDN_ACK	Select power down acknowledge signal of GPU PGC as the power down acknowledge for M4 LPM.
8 DDR2_PGC_PDN_ACK	Select power down acknowledge signal of DDR2 PGC as the power down acknowledge for M4 LPM.
7 DDR1_PGC_PDN_ACK	Select power down acknowledge signal of DDR1 PGC as the power down acknowledge for M4 LPM.
6 -	This field is reserved.
5 USB_OTG2_PGC_PDN_ACK	Select power down acknowledge signal of USB_OTG2 PGC as the power down acknowledge for M4 LPM.
4 USB_OTG1_PGC_PDN_ACK	Select power down acknowledge signal of USB_OTG1 PGC as the power down acknowledge for M4 LPM.
3 PCIE_PGC_PDN_ACK	Select power down acknowledge signal of PCIE PGC as the power down acknowledge for M4 LPM.
2 MIPI_PGC_PDN_ACK	Select power down acknowledge signal of MIPI PGC as the power down acknowledge for M4 LPM.
1 -	This field is reserved.
0 MF_PGC_PDN_ACK	Select power down acknowledge signal of MIX PGC as the power down acknowledge for M4 LPM.

5.2.10.63 Slot configure register for PGC CPUs (GPC_SLT*n*_CFG)

There are 20 slots in each SLTn_CFG(n = 0~19) that define the power up or power down behavior of one or more A53 cores, NOC, or SCU PGC in each slot. This array contains slots 15 to 19, see Memory Map for slots 0 to 14.

In each “SLTn_cfg”, 2 bits (slt_cfg[1:0])are reserved for each PGC:

- 2’b01 (slot controller will power down relevant PGC in corresponding slot if hardware power down request asserted)
- 2’b10 (slot controller will power up relevant PGC in corresponding slot if hardware power up request asserted)
- 2’b00 or 2’b11 (not power down or power up behavior in relevant slot)

The specific bits assignment for each PGC is shown in the table below.

	PGCx	PGCx-1	..	PGC2	PGC1	PGC0
SLT0_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLT1_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
:	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]
SLTn_CFG	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]	slt_cfg[1:0]

Address: 303A_0000h base + 1E8h offset + (4d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						SCU_PUP_SLOT_CONTROL	SCU_PDN_SLOT_CONTROL	CORE3_A53_PUP_SLOT_CONTROL	CORE3_A53_PDN_SLOT_CONTROL	CORE2_A53_PUP_SLOT_CONTROL	CORE2_A53_PDN_SLOT_CONTROL	CORE1_A53_PUP_SLOT_CONTROL	CORE1_A53_PDN_SLOT_CONTROL	CORE0_A53_PUP_SLOT_CONTROL	CORE0_A53_PDN_SLOT_CONTROL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG field descriptions

Field	Description
31–10 -	This field is reserved.
9 SCU_PUP_SLOT_CONTROL	SCU Power-up slot control

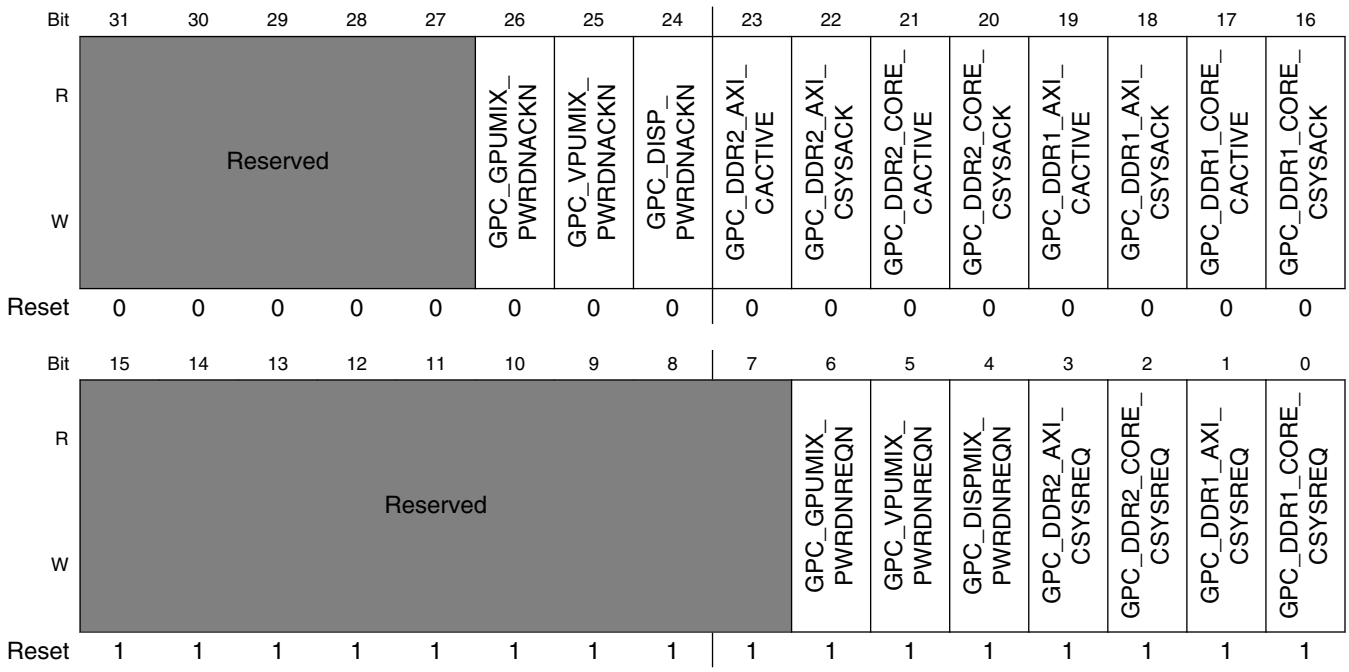
Table continues on the next page...

GPC_SLT*n*_CFG field descriptions (continued)

Field	Description
8 SCU_PDN_ SLOT_ CONTROL	SCU Power-down slot control
7 CORE3_A53_ PUP_SLOT_ CONTROL	CORE3 A53 Power-up slot control
6 CORE3_A53_ PDN_SLOT_ CONTROL	CORE3 A53 Power-down slot control
5 CORE2_A53_ PUP_SLOT_ CONTROL	CORE2 A53 Power-up slot control
4 CORE2_A53_ PDN_SLOT_ CONTROL	CORE2 A53 Power-down slot control
3 CORE1_A53_ PUP_SLOT_ CONTROL	CORE1 A53 Power-up slot control
2 CORE1_A53_ PDN_SLOT_ CONTROL	CORE1 A53 Power-down slot control
1 CORE0_A53_ PUP_SLOT_ CONTROL	CORE0 A53 Power-up slot control
0 CORE0_A53_ PDN_SLOT_ CONTROL	CORE0 A53 Power-down slot control

5.2.10.64 Power handshake register (GPC_PU_PWRHSK)

Address: 303A_0000h base + 1FCh offset = 303A_01FCh



GPC_PU_PWRHSK field descriptions

Field	Description
31–27 -	This field is reserved.
26 GPC_GPUMIX_PWRDNACKN	GPU ADB400 power down ack. Active 0
25 GPC_VPUMIX_PWRDNACKN	VPU ADB400 power down ack. Active 0
24 GPC_DISP_PWRDNACKN	DISP ADB400 power down ack. Active 0
23 GPC_DDR2_AXI_CACTIVE	DDR2 AXI Clock Active
22 GPC_DDR2_AXI_CSYSACK	DDR2 AXI Low-Power Request ack

Table continues on the next page...

GPC_PU_PWRHSK field descriptions (continued)

Field	Description
21 GPC_DDR2_ CORE_CACTIVE	DDR2 controller Hardware Low-Power Clock active
20 GPC_DDR2_ CORE_ CSYSACK	DDR2 controller Hardware Low_Power ack
19 GPC_DDR1_ AXI_CACTIVE	DDR1 AXI Clock Active
18 GPC_DDR1_ AXI_CSYSACK	DDR1 AXI Low-Power Request ack
17 GPC_DDR1_ CORE_CACTIVE	DDR1 controller Hardware Low-Power Clock active
16 GPC_DDR1_ CORE_ CSYSACK	DDR1 controller Hardware Low_Power ack
15–7 -	This field is reserved.
6 GPC_GPUMIX_ PWRDNREQN	GPU ADB400 power down request. Active 0
5 GPC_VPUMIX_ PWRDNREQN	VPU ADB400 power down request. Active 0
4 GPC_DISPMIX_ PWRDNREQN	DISPMIX ADB400 power down request. Active 0
3 GPC_DDR2_ AXI_CSYSREQ	DDR2 AXI Low-Power Request
2 GPC_DDR2_ CORE_ CSYSREQ	DDR2 controller Hardware Low-Power Request
1 GPC_DDR1_ AXI_CSYSREQ	DDR1 AXI Low-Power Request
0 GPC_DDR1_ CORE_ CSYSREQ	DDR1 controller Hardware Low-Power Request

5.2.10.65 Slot configure register for PGC PUs (GPC_SLT*n*_CFG_PU)

There are 20 slots in each SLTn_CFG_PU (n = 0~19) that define the power up or power down behavior of PU PGC in each slot. See PGC power domains section for list of PGC PUs.

Address: 303A_0000h base + 200h offset + (4d × i), where i=0d to 19d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCIE2_PUP_SLOT_CONTROL	PCIE2_PDN_SLOT_CONTROL	MIPI_CSI2_PUP_SLOT_CONTROL	MIPI_CSI2_PDN_SLOT_CONTROL	MIPI_CSI1_PUP_SLOT_CONTROL	MIPI_CSI1_PDN_SLOT_CONTROL	DISP_PUP_SLOT_CONTROL	DISP_PDN_SLOT_CONTROL	HDMI_PUP_SLOT_CONTROL	HDMI_PDN_SLOT_CONTROL	VPU_PUP_SLOT_CONTROL	VPU_PDN_SLOT_CONTROL	GPU_PUP_SLOT_CONTROL	GPU_PDN_SLOT_CONTROL	DDR2_PUP_SLOT_CONTROL	DDR2_PDN_SLOT_CONTROL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DDR1_PUP_SLOT_CONTROL	DDR1_PDN_SLOT_CONTROL	M4_PUP_SLOT_CONTROL	M4_PDN_SLOT_CONTROL	Reserved		OTG2_PUP_SLOT_CONTROL	OTG2_PDN_SLOT_CONTROL	OTG1_PUP_SLOT_CONTROL	OTG1_PDN_SLOT_CONTROL	PCIE_PUP_SLOT_CONTROL	PCIE_PDN_SLOT_CONTROL	MIPI_PUP_SLOT_CONTROL	MIPI_PDN_SLOT_CONTROL	MF_PUP_SLOT_CONTROL	MF_PDN_SLOT_CONTROL
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPC_SLT*n*_CFG_PU field descriptions

Field	Description
31 PCIE2_PUP_SLOT_CONTROL	PCIE2 Power-up slot control
30 PCIE2_PDN_SLOT_CONTROL	PCIE2 Power-down slot control
29 MIPI_CSI2_PUP_SLOT_CONTROL	MIPI_CSI2 Power-up slot control
28 MIPI_CSI2_PDN_SLOT_CONTROL	MIPI_CSI2 Power-down slot control

Table continues on the next page...

GPC_SLT*n*_CFG_PU field descriptions (continued)

Field	Description
27 MIPI_CSI1_PUP_SLOT_CONTROL	MIPI_CSI1 Power-up slot control
26 MIPI_CSI1_PDN_SLOT_CONTROL	MIPI_CSI1 Power-down slot control
25 DISP_PUP_SLOT_CONTROL	DISP Power-up slot control
24 DISP_PDN_SLOT_CONTROL	DISP Power-down slot control
23 HDMI_PUP_SLOT_CONTROL	HDMI Power-up slot control
22 HDMI_PDN_SLOT_CONTROL	HDMI Power-down slot control
21 VPU_PUP_SLOT_CONTROL	VPU Power-up slot control
20 VPU_PDN_SLOT_CONTROL	VPU Power-down slot control
19 GPU_PUP_SLOT_CONTROL	GPU Power-up slot control
18 GPU_PDN_SLOT_CONTROL	GPU Power-down slot control
17 DDR2_PUP_SLOT_CONTROL	DDR2 Power-up slot control
16 DDR2_PDN_SLOT_CONTROL	DDR2 Power-down slot control
15 DDR1_PUP_SLOT_CONTROL	DDR1 Power-up slot control

Table continues on the next page...

GPC_SLT*n*_CFG_PU field descriptions (continued)

Field	Description
14 DDR1_PDN_ SLOT_ CONTROL	DDR1 Power-down slot control
13 M4_PUP_SLOT_ CONTROL	M4 Power-up slot control
12 M4_PDN_SLOT_ CONTROL	M4 Power-down slot control
11–10 -	This field is reserved.
9 OTG2_PUP_ SLOT_ CONTROL	OTG2 Power-up slot control
8 OTG2_PDN_ SLOT_ CONTROL	OTG2 Power-down slot control
7 OTG1_PUP_ SLOT_ CONTROL	OTG1 Power-up slot control
6 OTG1_PDN_ SLOT_ CONTROL	OTG1 Power-down slot control
5 PCIE_PUP_ SLOT_ CONTROL	PCIE Power-up slot control
4 PCIE_PDN_ SLOT_ CONTROL	SCU Power-down slot control
3 MIPI_PUP_ SLOT_ CONTROL	MIPI Power-up slot control
2 MIPI_PDN_ SLOT_ CONTROL	MIPI Power-down slot control
1 MF_PUP_SLOT_ CONTROL	MF Power-up slot control
0 MF_PDN_SLOT_ CONTROL	MF Power-down slot control

5.2.11 GPC PGC Memory Map/Register Definition

There are 14 PGC inside GPCv2, with 4 different types: CPU/SCU/MIX/PU. PCIE/ MIPI/USB OTGx/USB HSIC PGC belongs to PU type PGC. Each type PGC has 4 different control words PGC_CTRL,PGC_PUPSCR,PGC_PDNSCR and PGC_SR. Different types PGC may have different field definition in these four registers. There is another extra control word PGC_AUXSW which has different field definition for PCIE/ MIPI PGC and SCU type PGC.

The total GPC memory map is 4KB

Table 5-11. Memory Regions

Address Range(offset)	Region
0x000 - 0x3FF	GPC configuration register
0x400 - 0x7FF	Reserved
0x800 - 0x9FF	CPU and SCU type PGC register base address
0xA00 - 0xBFF	MIX type PGC register base address
0xC00 - 0xFFF	PU type PGC register base address

Each PGC (CPU type, MIX type, PU type) will occupy 64 Bytes address space, the specific base address of each PGC are listed as below.

- 0x800 ~ 0x83F: PGC for A53 core0
- 0x840 ~ 0x87F: PGC for A53 core1
- 0x880 ~ 0x8BF: PGC for A53 core2
- 0x8C0 ~ 0x8FF: PGC for A53 core3
- 0x900 ~ 0x93F: PGC for A53 SCU
- 0xC00 ~ 0xC3F: PGC for MIPI DSI PHY (PU0)
- 0xC40 ~ 0xC7F: PGC for PCIE1 PHY (PU1)
- 0xC80 ~ 0xCBF: USB_OTG1 (PU2)
- 0xCC0 ~ 0xCFF: USB_OTG2 (PU3)
- 0xD00 ~ 0xD3F: Reserved (PU4)
- 0xD40 ~ 0xD7F: DDR1 (PU5)
- 0xD80 ~ 0xDBF: Reserved (PU6)
- 0xDC0 ~ 0xDFF: GPUMIX (PU7)
- 0xE00 ~ 0xE3F: VPUMIX (PU8)
- 0xE40 ~ 0xE7F: HDMI (PU9)
- 0xE80 ~ 0xEBF: DISPMIX (PU10)
- 0xEC0 ~ 0xEFF: MIPI CSI1 PHY (PU11)

- 0xF00 ~ 0xF3F: MIPI CSI2 PHY (PU12)
- 0xF40 ~ 0xF7F: PCIE2 PHY (PU13)

GPC_PGC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0800	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE0_CTRL)	32	R/W	0604_0202h	5.2.11.1/ 733
303A_0804	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE0_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/ 734
303A_0808	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE0_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/ 735
303A_080C	GPC PGC Status Register (GPC_PGC_A53CORE0_SR)	32	R/W	0000_1000h	5.2.11.4/ 737
303A_0810	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_A53CORE0_AUXSW)	32	R/W	0000_0131h	5.2.11.5/ 740
303A_0840	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE1_CTRL)	32	R/W	0604_0202h	5.2.11.1/ 733
303A_0844	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE1_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/ 734
303A_0848	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE1_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/ 735
303A_084C	GPC PGC Status Register (GPC_PGC_A53CORE1_SR)	32	R/W	0000_1000h	5.2.11.4/ 737
303A_0850	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_A53CORE1_AUXSW)	32	R/W	0000_0131h	5.2.11.5/ 740
303A_0880	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE2_CTRL)	32	R/W	0604_0202h	5.2.11.1/ 733
303A_0884	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE2_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/ 734
303A_0888	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE2_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/ 735
303A_088C	GPC PGC Status Register (GPC_PGC_A53CORE2_SR)	32	R/W	0000_1000h	5.2.11.4/ 737
303A_0890	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_A53CORE2_AUXSW)	32	R/W	0000_0131h	5.2.11.5/ 740
303A_08C0	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53CORE3_CTRL)	32	R/W	0604_0202h	5.2.11.1/ 733
303A_08C4	GPC PGC Up Sequence Control Register (GPC_PGC_A53CORE3_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/ 734
303A_08C8	GPC PGC Down Sequence Control Register (GPC_PGC_A53CORE3_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/ 735
303A_08CC	GPC PGC Status Register (GPC_PGC_A53CORE3_SR)	32	R/W	0000_1000h	5.2.11.4/ 737
303A_08D0	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_A53CORE3_AUXSW)	32	R/W	0000_0131h	5.2.11.5/ 740

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0900	GPC PGC Control Register for PGC CPUs (GPC_PGC_A53SCU_CTRL)	32	R/W	0604_0202h	5.2.11.1/ 733
303A_0904	GPC PGC Up Sequence Control Register (GPC_PGC_A53SCU_PUPSCR)	32	R/W	0009_97C1h	5.2.11.2/ 734
303A_0908	GPC PGC Down Sequence Control Register (GPC_PGC_A53SCU_PDNSCR)	32	R/W	2100_0801h	5.2.11.3/ 735
303A_090C	GPC PGC Status Register (GPC_PGC_A53SCU_SR)	32	R/W	0000_1000h	5.2.11.4/ 737
303A_0910	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_A53SCU_AUXSW)	32	R/W	0000_0131h	5.2.11.5/ 740
303A_0A10	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_MIX_AUXSW)	32	R/W	0000_0131h	5.2.11.6/ 741
303A_0A40	GPC PGC Control Register for PGC MIX. (GPC_PGC_NOC_MIX_CTRL)	32	R/W	0604_0202h	5.2.11.7/ 743
303A_0A44	GPC PGC Up Sequence Control Register (GPC_PGC_NOC_MIX_PUPSCR)	32	R/W	0009_97C1h	5.2.11.8/ 744
303A_0A48	GPC PGC Down Sequence Control Register (GPC_PGC_NOC_MIX_PDNSCR)	32	R/W	2100_0801h	5.2.11.9/ 745
303A_0A4C	GPC PGC Status Register (GPC_PGC_NOC_MIX_SR)	32	R/W	0000_1000h	5.2.11.10/ 747
303A_0C00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU0_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0C04	GPC PGC Up Sequence Control Register (GPC_PGC_PU0_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0C08	GPC PGC Down Sequence Control Register (GPC_PGC_PU0_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0C0C	GPC PGC Status Register (GPC_PGC_PU0_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0C10	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU0_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0C40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU1_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0C44	GPC PGC Up Sequence Control Register (GPC_PGC_PU1_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0C48	GPC PGC Down Sequence Control Register (GPC_PGC_PU1_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0C4C	GPC PGC Status Register (GPC_PGC_PU1_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0C50	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU1_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0C80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU2_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0C84	GPC PGC Up Sequence Control Register (GPC_PGC_PU2_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0C88	GPC PGC Down Sequence Control Register (GPC_PGC_PU2_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0C8C	GPC PGC Status Register (GPC_PGC_PU2_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0C90	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU2_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0CC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU3_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0CC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU3_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0CC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU3_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0CCC	GPC PGC Status Register (GPC_PGC_PU3_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0CD0	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU3_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0D00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU4_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0D04	GPC PGC Up Sequence Control Register (GPC_PGC_PU4_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0D08	GPC PGC Down Sequence Control Register (GPC_PGC_PU4_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0D0C	GPC PGC Status Register (GPC_PGC_PU4_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0D10	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU4_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0D40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU5_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0D44	GPC PGC Up Sequence Control Register (GPC_PGC_PU5_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0D48	GPC PGC Down Sequence Control Register (GPC_PGC_PU5_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0D4C	GPC PGC Status Register (GPC_PGC_PU5_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0D50	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU5_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0D80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU6_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0D84	GPC PGC Up Sequence Control Register (GPC_PGC_PU6_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0D88	GPC PGC Down Sequence Control Register (GPC_PGC_PU6_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0D8C	GPC PGC Status Register (GPC_PGC_PU6_SR)	32	R/W	0000_1000h	5.2.11.14/ 754

Table continues on the next page...

GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0D90	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU6_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0DC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU7_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0DC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU7_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0DC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU7_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0DCC	GPC PGC Status Register (GPC_PGC_PU7_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0DD0	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU7_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0E00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU8_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0E04	GPC PGC Up Sequence Control Register (GPC_PGC_PU8_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0E08	GPC PGC Down Sequence Control Register (GPC_PGC_PU8_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0E0C	GPC PGC Status Register (GPC_PGC_PU8_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0E10	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU8_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0E40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU9_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0E44	GPC PGC Up Sequence Control Register (GPC_PGC_PU9_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0E48	GPC PGC Down Sequence Control Register (GPC_PGC_PU9_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0E4C	GPC PGC Status Register (GPC_PGC_PU9_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0E50	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU9_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0E80	GPC PGC Control Register for PGC PUs (GPC_PGC_PU10_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0E84	GPC PGC Up Sequence Control Register (GPC_PGC_PU10_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0E88	GPC PGC Down Sequence Control Register (GPC_PGC_PU10_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0E8C	GPC PGC Status Register (GPC_PGC_PU10_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0E90	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU10_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0EC0	GPC PGC Control Register for PGC PUs (GPC_PGC_PU11_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750

Table continues on the next page...

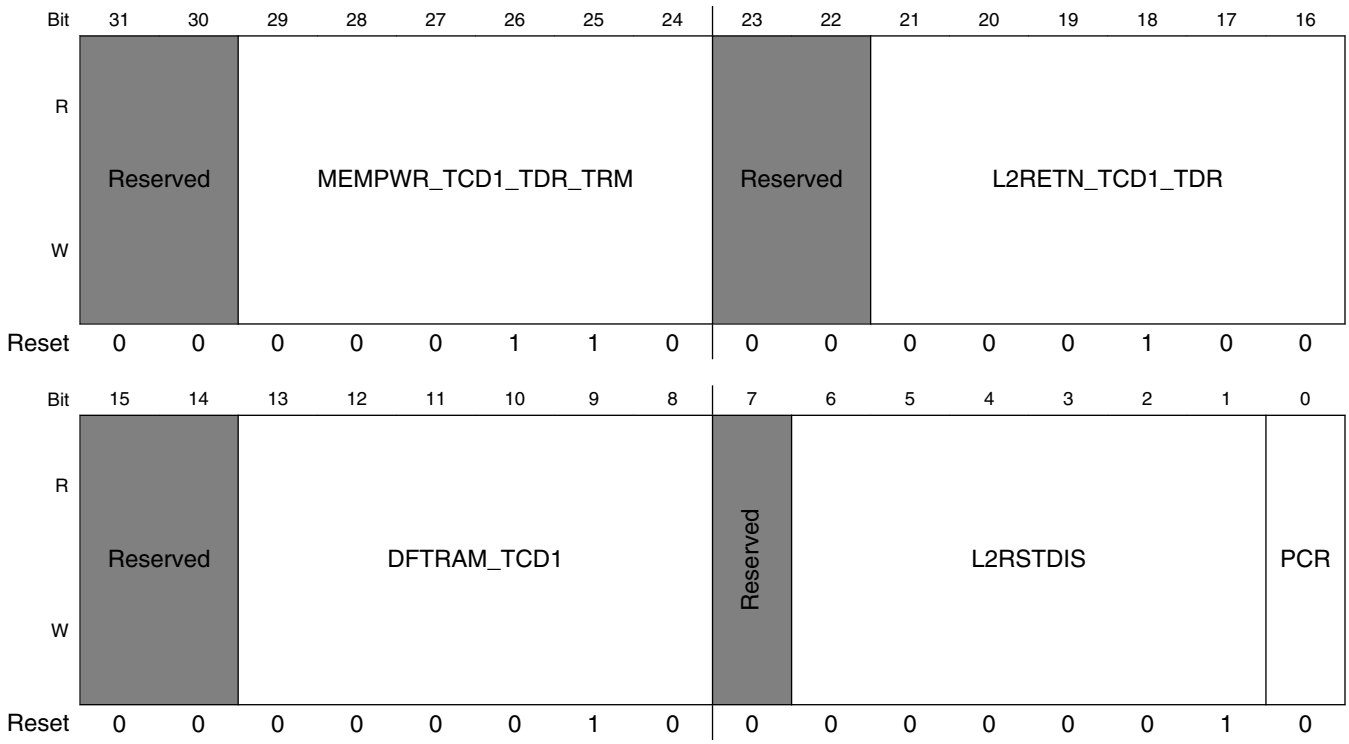
GPC_PGC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
303A_0EC4	GPC PGC Up Sequence Control Register (GPC_PGC_PU11_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0EC8	GPC PGC Down Sequence Control Register (GPC_PGC_PU11_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0ECC	GPC PGC Status Register (GPC_PGC_PU11_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0ED0	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU11_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0F00	GPC PGC Control Register for PGC PUs (GPC_PGC_PU12_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0F04	GPC PGC Up Sequence Control Register (GPC_PGC_PU12_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0F08	GPC PGC Down Sequence Control Register (GPC_PGC_PU12_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0F0C	GPC PGC Status Register (GPC_PGC_PU12_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0F10	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU12_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757
303A_0F40	GPC PGC Control Register for PGC PUs (GPC_PGC_PU13_CTRL)	32	R/W	0604_0202h	5.2.11.11/ 750
303A_0F44	GPC PGC Up Sequence Control Register (GPC_PGC_PU13_PUPSCR)	32	R/W	0009_97C1h	5.2.11.12/ 751
303A_0F48	GPC PGC Down Sequence Control Register (GPC_PGC_PU13_PDNSCR)	32	R/W	2100_0801h	5.2.11.13/ 752
303A_0F4C	GPC PGC Status Register (GPC_PGC_PU13_SR)	32	R/W	0000_1000h	5.2.11.14/ 754
303A_0F50	GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_PU13_AUXSW)	32	R/W	0000_0131h	5.2.11.15/ 757

5.2.11.1 GPC PGC Control Register for PGC CPUs (GPC_PGC_*n*CTRL)

GPC PGC Control Register for the PGC CPUs. See the PGC Memory Map for the assignments.

Address: 303A_0000h base + 800h offset + (64d × i), where i=0d to 4d



GPC_PGC_*n*CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_ TCD1_TDR_ TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_ TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved

Table continues on the next page...

GPC_PGC_*n*CTRL field descriptions (continued)

Field	Description
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.2 GPC PGC Up Sequence Control Register (GPC_PGC_*n*PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + 804h offset + (64d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCALL_SCALLOUT_CNT									SW2ISO						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW2ISO								PUP_WAIT_SCALL_OUT	SW						
Reset	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1

GPC_PGC_*n*PUPSCR field descriptions

Field	Description
31–23 PUP_SCALL_ SCALLOUT_CNT	After SCALL asserting to 1'b0, count this value to complete switch power up **NOTE:** Only valid when pup_wait_scall_out is set to 1'b0. Can't be programmed to zero (This register control only for MIX Type PGC)
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 PUP_WAIT_ SCALL_OUT	After SCALL asserting to 1'b0, wait handshake signal SCALL_OUT to return to 1'b0 (This register control only for MIX Type PGC)
SW	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW before asserting switch_b **NOTE:** SW must not be programmed to zero.

5.2.11.3 GPC PGC Down Sequence Control Register (GPC_PGC_*n*PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + 808h offset + (64d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCPRE_SCALL_CNT								PDN_SCALL_SCALLOUT_CNT							
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						PDN_WAIT_SCALL_OUT	Reserved	ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

GPC_PGC_*n*PDNSCR field descriptions

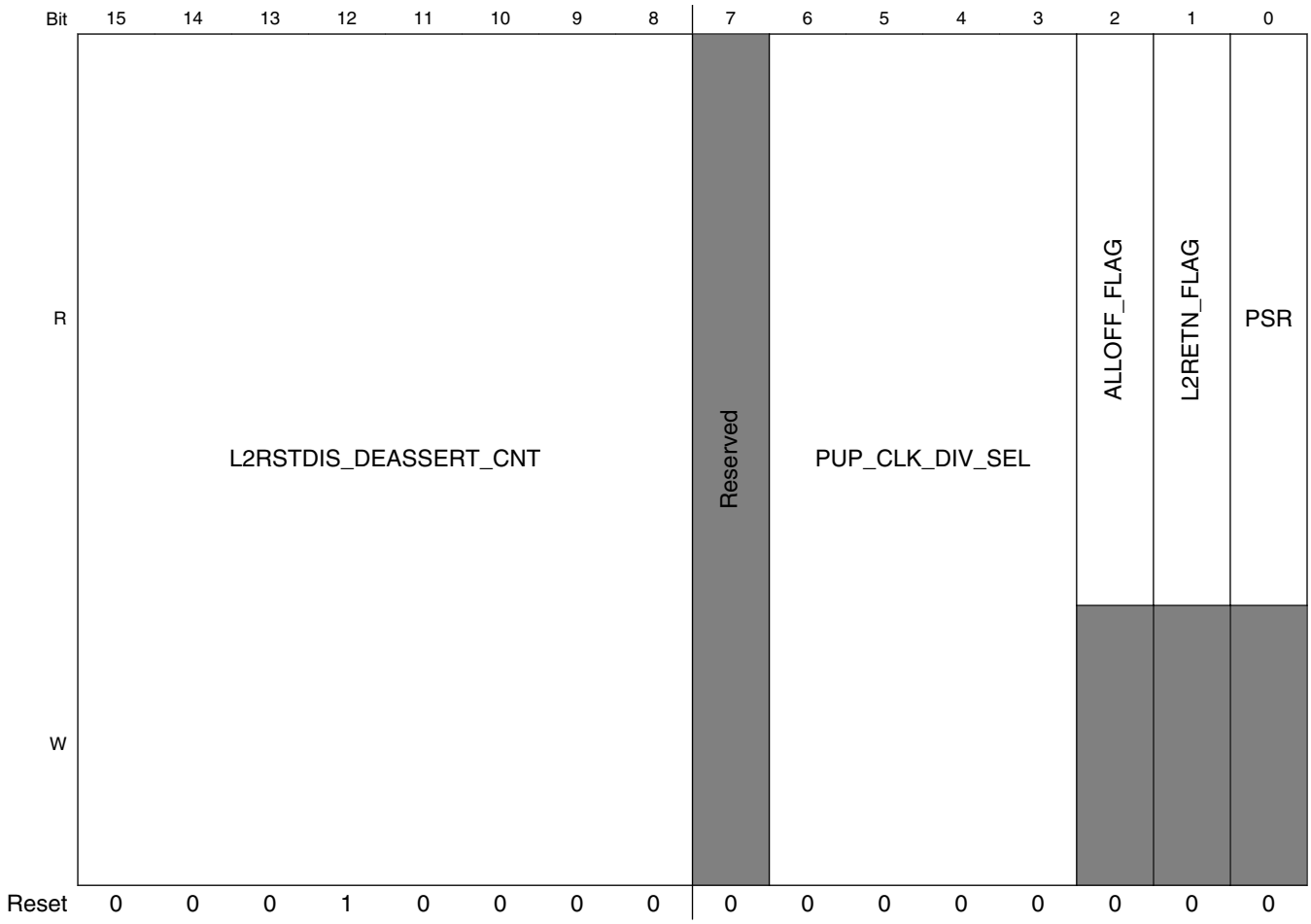
Field	Description
31–24 PUP_SCPRE_ SCALL_CNT	After SCPRE asserting to 1’b0, count this value to assert SCALL to 1’b0 (This register control only for MIX Type PGC)
23–16 PDN_SCALL_ SCALLOUT_CNT	Default 8'h0. After SCALL asserting to 1’b1, count this value to complete switch power down. **NOTE:** This control is only valid when pdn_wait_scall_out is set to 1’b0. It can be programmed to zero. (This register control only for MIX Type PGC)
15–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7 PDN_WAIT_ SCALL_OUT	Default 1'b0 If set to 1'b1, after SCALL asserting to 1'b1, wait handshake signal SCALL_OUT to change to 1'b1. **NOTE:** This register control is only for MIX Type PGC.
6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.4 GPC PGC Status Register (GPC_PGC_*n*SR)

GPC PGC Status Register

Address: 303A_0000h base + 80Ch offset + (64d × i), where i=0d to 4d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved														L2RSTDIS_DEASSERT_CNT	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



GPC_PGC_*n*SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

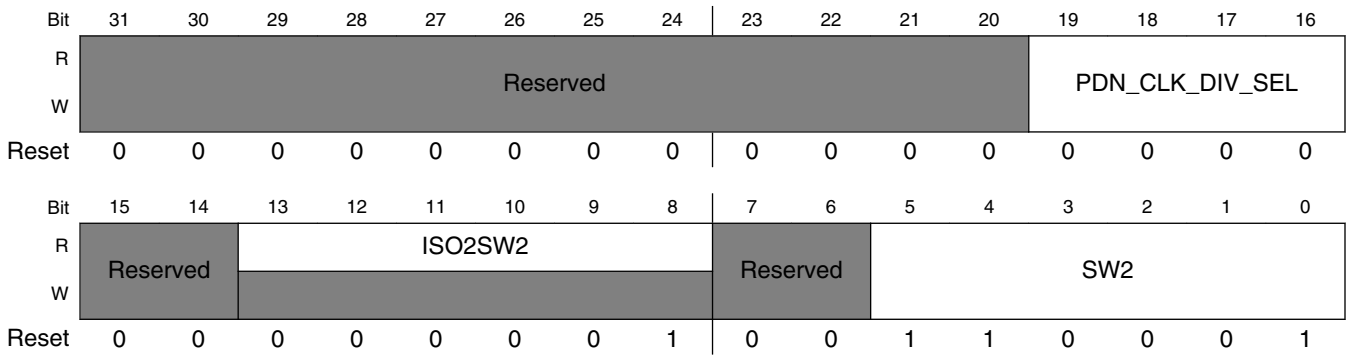
GPC_PGC_*n*SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.2.11.5 GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_*n*AUXSW)

GPC PGC Auxiliary Power Switch Control Register. The register has different field definition for PCIE/MIPI PGC and SCU Type PGC.

Address: 303A_0000h base + 810h offset + (64d × i), where i=0d to 4d



GPC_PGC_*n*AUXSW field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19–16 PDN_CLK_DIV_SEL	Clock divider select for the clock of power down counter 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk 1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
15–14 -	This field is reserved. Reserved
13–8 ISO2SW2	after asserting isolation by power-down request(pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW2 before negating switch2_b(1P8 Power)

Table continues on the next page...

GPC_PGC_*n*AUXSW field descriptions (continued)

Field	Description
	NOTE: ISO2SW2 must not be programmed to zero. 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
7–6 -	This field is reserved. Reserved
SW2	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW2 before asserting switch2_b(1P8 Power) **NOTE:** SW2 must not be programmed to zero.

5.2.11.6 GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_MIX_AUXSW)

GPC PGC Auxiliary Power Switch Control Register. The register has different field definition for PCIE/MIPI PGC and SCU Type PGC.

Address: 303A_0000h base + A10h offset = 303A_0A10h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												PDN_CLK_DIV_SEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW2						Reserved		SW2					
Reset	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1

GPC_PGC_MIX_AUXSW field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19–16 PDN_CLK_DIV_SEL	Clock divider select for the clock of power down counter 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk

Table continues on the next page...

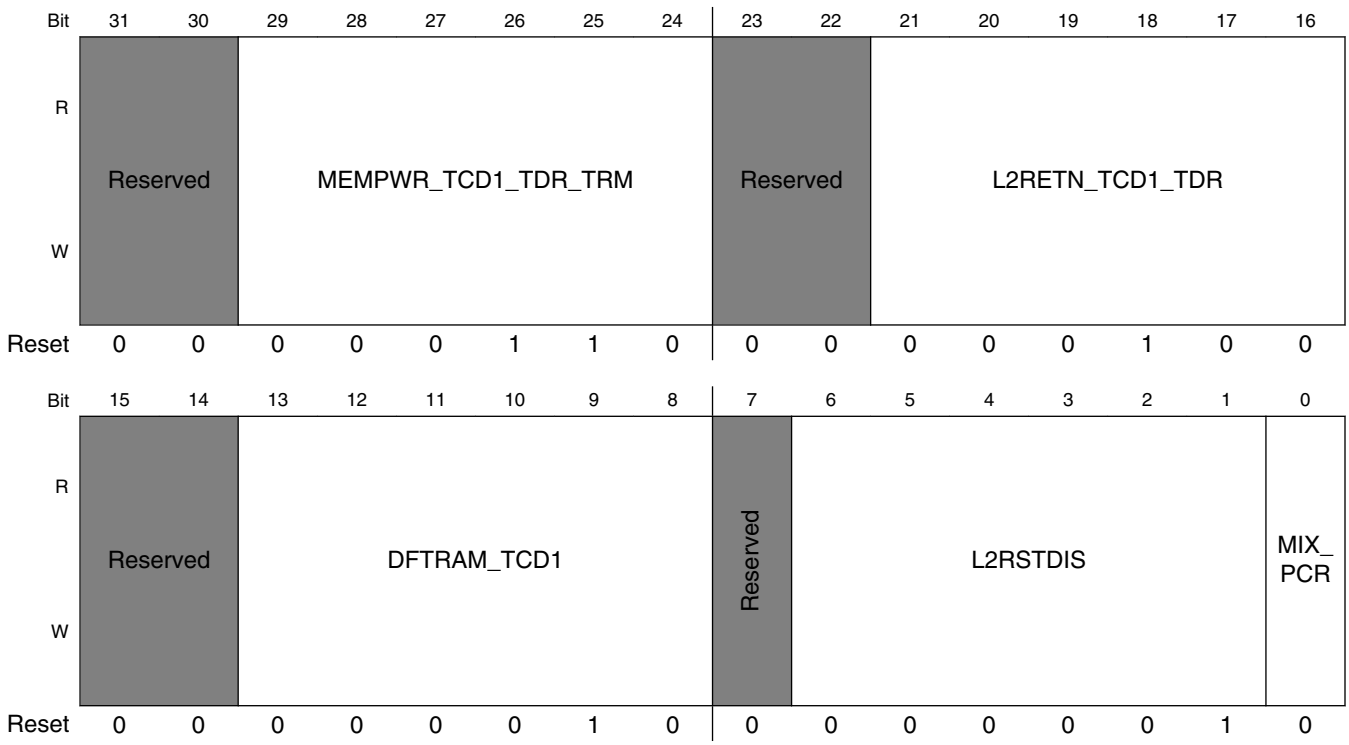
GPC_PGC_MIX_AUXSW field descriptions (continued)

Field	Description
	0110 1/64 count_clk 0111 1/128 count_clk 1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
15–14 -	This field is reserved. Reserved
13–8 ISO2SW2	after asserting isolation by power-down request(pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW2 before negating switch2_b(1P8 Power) **NOTE:** ISO2SW2 must not be programmed to zero. 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
7–6 -	This field is reserved. Reserved
SW2	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW2 before asserting switch2_b(1P8 Power) **NOTE:** SW2 must not be programmed to zero.

5.2.11.7 GPC PGC Control Register for PGC MIX. (GPC_PGC_NOC_MIX_CTRL)

GPC PGC Control Register.

Address: 303A_0000h base + A40h offset = 303A_0A40h



GPC_PGC_NOC_MIX_CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_TCD1_TDR_TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved

Table continues on the next page...

GPC_PGC_NOC_MIX_CTRL field descriptions (continued)

Field	Description
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 MIX_PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.8 GPC PGC Up Sequence Control Register (GPC_PGC_NOC_MIX_PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + A44h offset = 303A_0A44h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCALL_SCALLOUT_CNT									SW2ISO						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW2ISO									PUP_WAIT_SCALL_OUT	SW					
Reset	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1

GPC_PGC_NOC_MIX_PUPSCR field descriptions

Field	Description
31–23 PUP_SCALL_ SCALLOUT_CNT	After SCALL asserting to 1'b0, count this value to complete switch power up **NOTE:** Only valid when pup_wait_scall_out is set to 1'b0. Can't be programmed to zero (This register control only for MIX Type PGC)
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 PUP_WAIT_ SCALL_OUT	After SCALL asserting to 1'b0, wait handshake signal SCALL_OUT to return to 1'b0 (This register control only for MIX Type PGC)
SW	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW before asserting switch_b **NOTE:** SW must not be programmed to zero.

5.2.11.9 GPC PGC Down Sequence Control Register (GPC_PGC_NOC_MIX_PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + A48h offset = 303A_0A48h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCPRE_SCALL_CNT								PDN_SCALL_SCALLOUT_CNT							
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						PDN_WAIT_SCALL_OUT	Reserved	ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

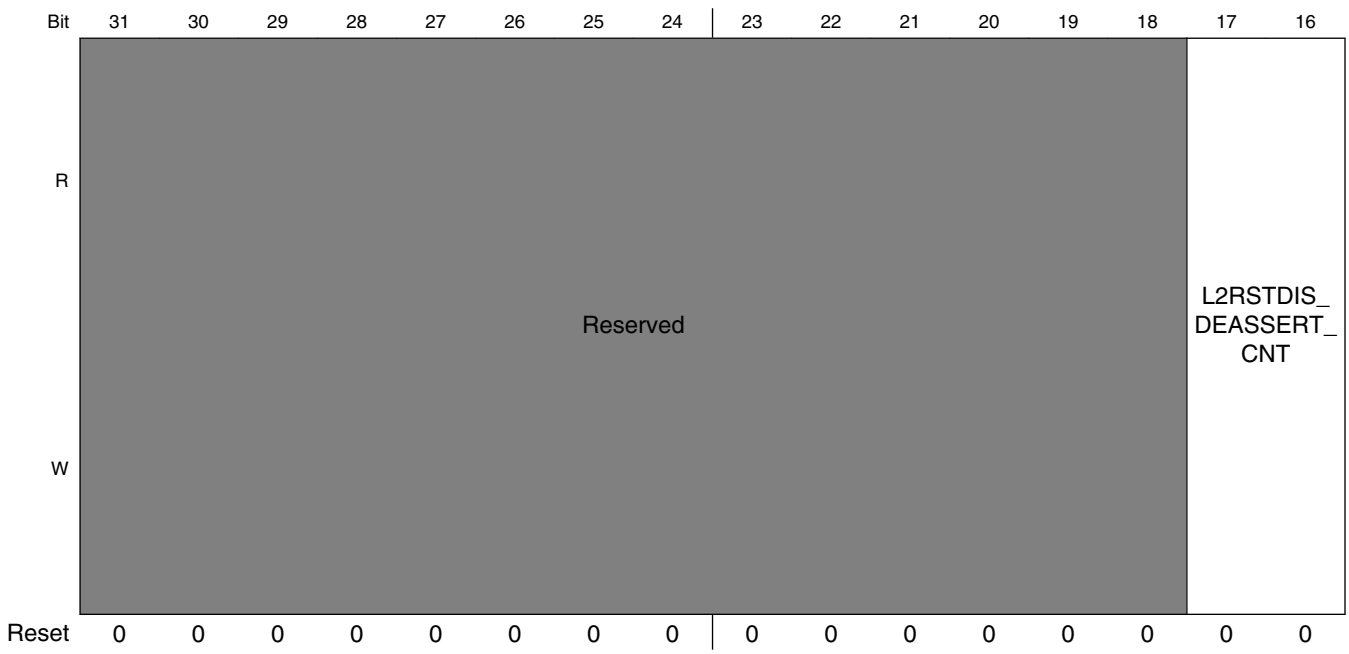
GPC_PGC_NOC_MIX_PDNSCR field descriptions

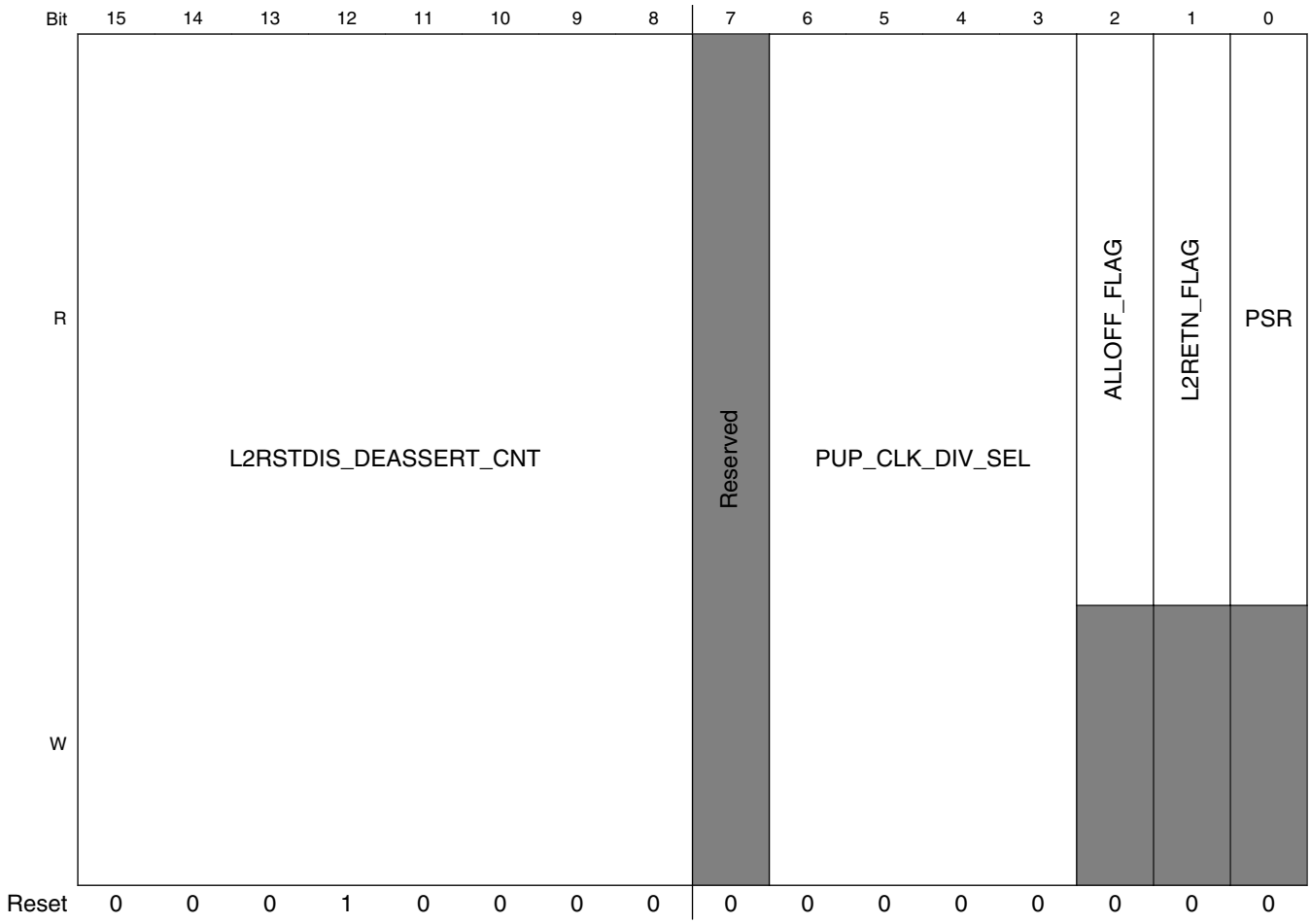
Field	Description
31–24 PUP_SCPRE_ SCALL_CNT	After SCPRE asserting to 1’b0, count this value to assert SCALL to 1’b0 (This register control only for MIX Type PGC)
23–16 PDN_SCALL_ SCALLOUT_CNT	Default 8'h0. After SCALL asserting to 1’b1, count this value to complete switch power down. **NOTE:** This control is only valid when pdn_wait_scall_out is set to 1’b0. It can be programmed to zero. (This register control only for MIX Type PGC)
15–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7 PDN_WAIT_ SCALL_OUT	Default 1'b0 If set to 1'b1, after SCALL asserting to 1'b1, wait handshake signal SCALL_OUT to change to 1'b1. **NOTE:** This register control is only for MIX Type PGC.
6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.10 GPC PGC Status Register (GPC_PGC_NOC_MIX_SR)

GPC PGC Status Register

Address: 303A_0000h base + A4Ch offset = 303A_0A4Ch





GPC_PGC_NOC_MIX_SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

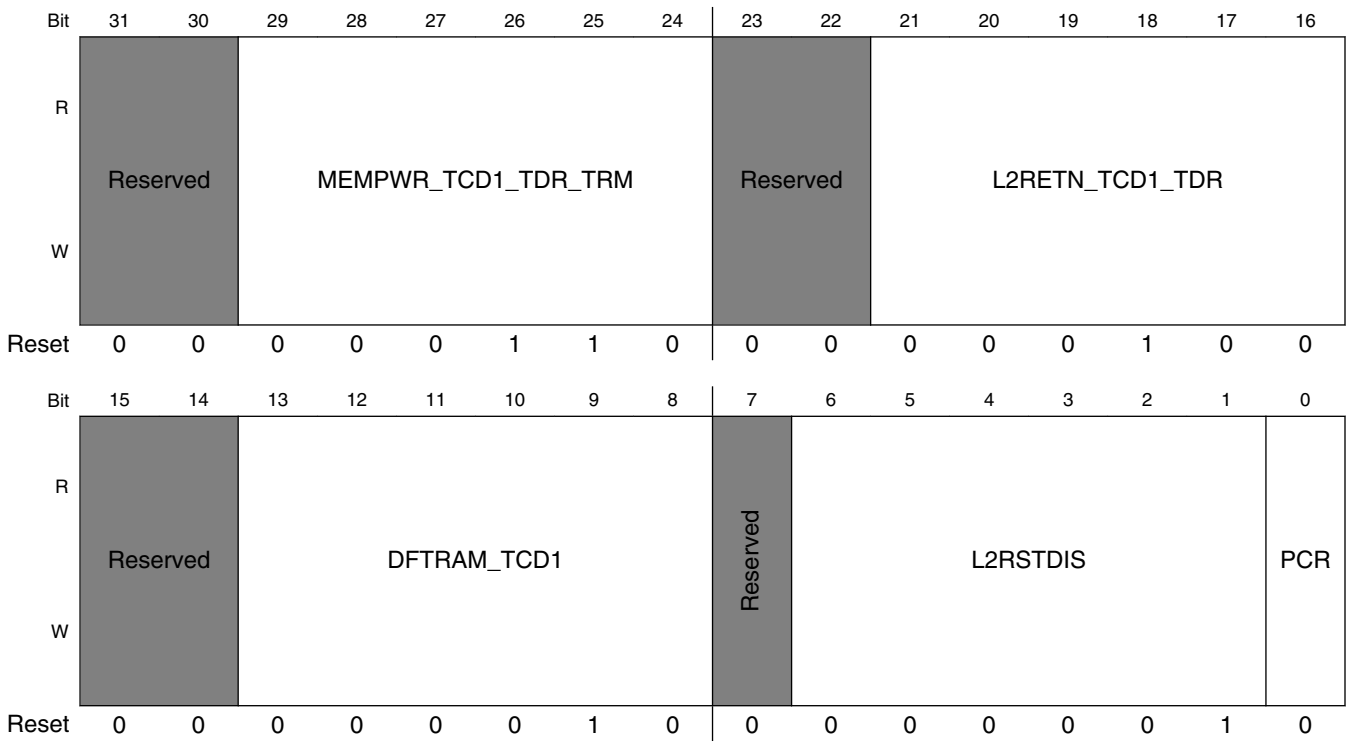
GPC_PGC_NOC_MIX_SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.2.11.11 GPC PGC Control Register for PGC PUs (GPC_PGC_*n*CTRL)

GPC PGC Control Register for the PUs. See the PGC Memory Map for the assignments.

Address: 303A_0000h base + C00h offset + (64d × i), where i=0d to 13d



GPC_PGC_*n*CTRL field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–24 MEMPWR_ TCD1_TDR_ TRM	After scu pdn_req, count this value to assert A53 mempwr to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
23–22 -	This field is reserved. Reserved
21–16 L2RETN_TCD1_ TDR	After scu pdn_req, count this value to assert A53 l2retn to 1'b0 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
15–14 -	This field is reserved. Reserved

Table continues on the next page...

GPC_PGC_*n*CTRL field descriptions (continued)

Field	Description
13–8 DFTRAM_TCD1	After scu pdn_req, count this value to assert A53 dftram to 1'b1 **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–1 L2RSTDIS	After scu pdn_req, count this value to assert A53 l2rstdis to 1'b1, it will be clear automatically once any of A53 core0/core1/core2/core3 is wakeup **NOTE:** Can't be programmed to zero (This register control only for SCU Type PGC)
0 PCR	Power Control **NOTE:** PCR must not change from power-down request (pdn_req) assertion until the target subsystem is completely powered up. 0 Do not switch off power even if pdn_req is asserted. 1 Switch off power when pdn_req is asserted.

5.2.11.12 GPC PGC Up Sequence Control Register (GPC_PGC_*n*PUPSCR)

GPC PGC Up Sequence Control Register

Address: 303A_0000h base + C04h offset + (64d × i), where i=0d to 13d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCALL_SCALLOUT_CNT									SW2ISO						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW2ISO									PUP_WAIT_SCALL_OUT	SW					
Reset	1	0	0	1	0	1	1	1	1	1	0	0	0	0	0	1

GPC_PGC_*n*PUPSCR field descriptions

Field	Description
31–23 PUP_SCALL_ SCALLOUT_CNT	After SCALL asserting to 1'b0, count this value to complete switch power up **NOTE:** Only valid when pup_wait_scall_out is set to 1'b0. Can't be programmed to zero (This register control only for MIX Type PGC)
22–7 SW2ISO	After asserting switch_b, the PGC waits a number of clocks equal to the value of SW2ISO before negating isolation.
6 PUP_WAIT_ SCALL_OUT	After SCALL asserting to 1'b0, wait handshake signal SCALL_OUT to return to 1'b0 (This register control only for MIX Type PGC)
SW	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW before asserting switch_b **NOTE:** SW must not be programmed to zero.

5.2.11.13 GPC PGC Down Sequence Control Register (GPC_PGC_*n*PDNSCR)

GPC PGC Down Sequence Control Register

Address: 303A_0000h base + C08h offset + (64d × i), where i=0d to 13d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PUP_SCPRE_SCALL_CNT								PDN_SCALL_SCALLOUT_CNT							
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		ISO2SW						PDN_WAIT_SCALL_OUT	Reserved	ISO					
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	1

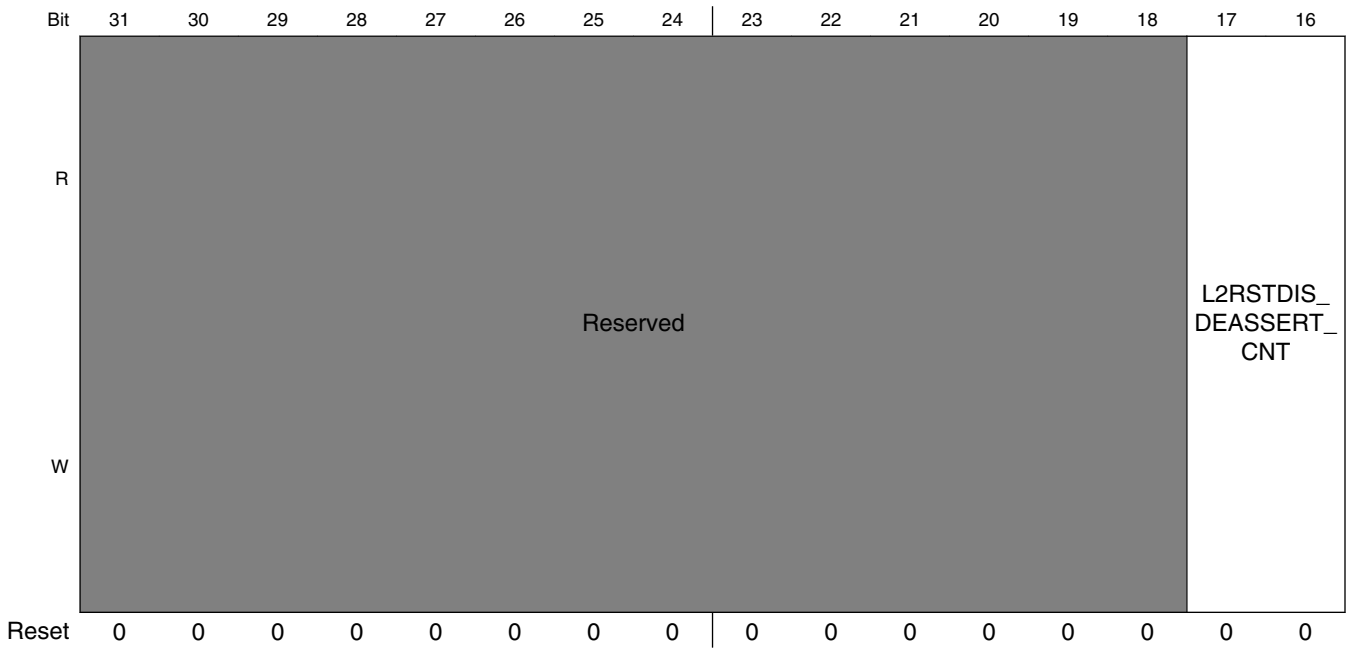
GPC_PGC_*n*PDNSCR field descriptions

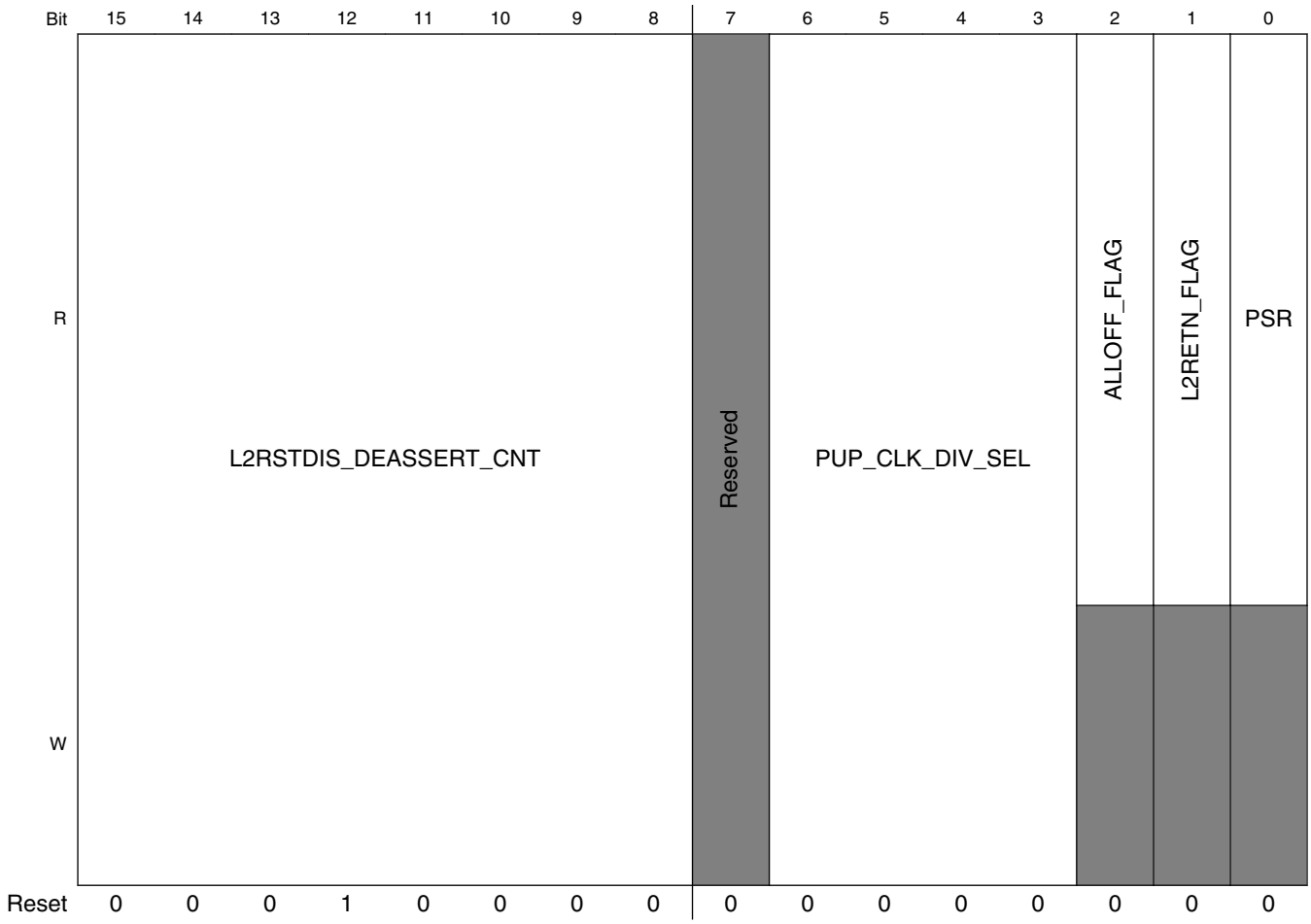
Field	Description
31–24 PUP_SCPRE_ SCALL_CNT	After SCPRE asserting to 1’b0, count this value to assert SCALL to 1’b0 (This register control only for MIX Type PGC)
23–16 PDN_SCALL_ SCALLOUT_CNT	Default 8'h0. After SCALL asserting to 1’b1, count this value to complete switch power down. **NOTE:** This control is only valid when pdn_wait_scall_out is set to 1’b0. It can be programmed to zero. (This register control only for MIX Type PGC)
15–14 -	This field is reserved. Reserved
13–8 ISO2SW	After asserting isolation(by pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW before negating switch_b **NOTE:** ISO2SW must not be programmed to zero.
7 PDN_WAIT_ SCALL_OUT	Default 1'b0 If set to 1'b1, after SCALL asserting to 1'b1, wait handshake signal SCALL_OUT to change to 1'b1. **NOTE:** This register control is only for MIX Type PGC.
6 -	This field is reserved. Reserved
ISO	After a power-down request (pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO before asserting isolation **NOTE:** ISO must not be programmed to zero.

5.2.11.14 GPC PGC Status Register (GPC_PGC_*n*SR)

GPC PGC Status Register

Address: 303A_0000h base + C0Ch offset + (64d × i), where i=0d to 13d





GPC_PGC_*n*SR field descriptions

Field	Description
31–18 -	This field is reserved. Reserved
17–8 L2RSTDIS_ DEASSERT_ CNT	Count this value to de-assert L2RSTDISABLE to LOW after CPU0 or CPU1 power up **NOTE:** This value can't be programmed to zero (This register control only for SCU Type PGC)
7 -	This field is reserved. Reserved
6–3 PUP_CLK_DIV_ SEL	Clock divider select for the clock of power up counter(count_clk is 32KHz for CPU/SCU type PGC, ipg_clk(66MHz) for MIX/PU Type PGC) 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk

Table continues on the next page...

GPC_PGC_*n*SR field descriptions (continued)

Field	Description
	1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
2 ALLOFF_FLAG	All-off flag. **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from ALL_OFF mode, otherwise, it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
1 L2RETN_FLAG	L2 Retention Flag **NOTE:** Software should write “1” to clear this flag after A53 is wakeup from L2 retention mode, otherwise it will always keep to 1 (This register control only for SCU Type PGC) 0 A53 is not wakeup from L2 retention mode. 1 A53 is wakeup from L2 retention mode.
0 PSR	Power status. When in functional (or software-controlled debug) mode, PGC hardware sets PSR as soon as any of the power control output changes its state to one. Write one to clear this bit. Software should clear this bit after power up; otherwise, PSR continues to reflect the power status of the initial power down. 0 The target subsystem was not powered down for the previous power-down request. 1 The target subsystem was powered down for the previous power-down request.

5.2.11.15 GPC PGC Auxiliary Power Switch Control Register (GPC_PGC_*n*AUXSW)

GPC PGC Auxiliary Power Switch Control Register. The register has different field definition for PCIE/MIPI PGC and SCU Type PGC.

Address: 303A_0000h base + C10h offset + (64d × i), where i=0d to 13d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												PDN_CLK_DIV_SEL			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved		ISO2SW2						Reserved		SW2					
W																
Reset	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	1

GPC_PGC_*n*AUXSW field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19–16 PDN_CLK_DIV_SEL	Clock divider select for the clock of power down counter 0000 1 0001 1/2 count_clk 0010 1/4 count_clk 0011 1/8 count_clk 0100 1/16 count_clk 0101 1/32 count_clk 0110 1/64 count_clk 0111 1/128 count_clk 1000 1/256 count_clk 1001 1/512 count_clk 1010 1/1024 count_clk 1011 1/2056 count_clk 1100 1/4096 count_clk 1101 1/8192 count_clk 1110 1/16384 count_clk 1111 1/32768 count_clk
15–14 -	This field is reserved. Reserved
13–8 ISO2SW2	

Table continues on the next page...

GPC_PGC_*n*AUXSW field descriptions (continued)

Field	Description
	after asserting isolation by power-down request(pdn_req assertion), the PGC waits a number of clocks equal to the value of ISO2SW2 before negating switch2_b(1P8 Power) **NOTE:** ISO2SW2 must not be programmed to zero. 0 A53 is not wakeup from ALL_OFF mode. 1 A53 is wakeup from ALL_OFF mode.
7–6 -	This field is reserved. Reserved
SW2	After a power-up request (pup_req assertion), the PGC waits a number of clocks equal to the value of SW2 before asserting switch2_b(1P8 Power) **NOTE:** SW2 must not be programmed to zero.

5.3 Crystal Oscillator (XTALOSC)

5.3.1 Overview

The chip has three XTAL modules, 25MHz XTAL module, 27MHz XTAL module and 32KHz XTAL Module. The 25MHz and 27MHz XTAL modules are instantiated from the same XTAL IP, which includes:

- 25/27MHz crystal oscillator to generate reference clock
- Digital control logics for the XTAL

The 32KHz XTAL module uses a different IP and it is used as the clock source for the RTC, located in the SNVS.

The 25MHz oscillator will be used as the primary clock source for the PLLs to generate the clock for CPU, BUS, and high-speed interfaces. For fractional PLLs, the 25MHz clock from the oscillator can be used as the PLL reference clock directly.

The 27MHz oscillator will be used as the reference clock for HDMI PHY. It can also be used as the alternative source for the PLLs.

Each XTAL module supports the following modes through register configuration:

- Normal oscillator mode - In normal mode, the XTAL IP generates stable square wave based on the crystal oscillator input.
- Bypass mode - In bypass mode, an external clock can be input through the XTAL pad. The IP supports up to 50MHz external clock in bypass mode.

5.3.2 XTALOSC Memory Map/Register Definition

XTALOSC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3027_0000	25M Oscillator Control Configuration Register (XTALOSC_OSC25M_CTL_CFG)	32	R/W	0180_2070h	5.3.2.1/ 759
3027_8000	27M Oscillator Control Configuration Register (XTALOSC_OSC27M_CTL_CFG)	32	R/W	0180_2070h	5.3.2.2/ 761

5.3.2.1 25M Oscillator Control Configuration Register (XTALOSC_OSC25M_CTL_CFG)

25M Oscillator Control Configuration Register

Address: 3027_0000h base + 0h offset = 3027_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	OSC_BYPSS	OSC_GM_TST_SEL	Reserved						OSC_EOCV							
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	OSC_INT_MASK	Reserved	OSC_OK_BYPASS	OSC_DIV					OSC_INT_STU	OSC_GM_SEL			OSC_HYST_CTL	OSC_ALC_CTL	Reserved	
W																
Reset	0	0	1	0	0	0	0	0	0	1	1	1	0	0	0	0

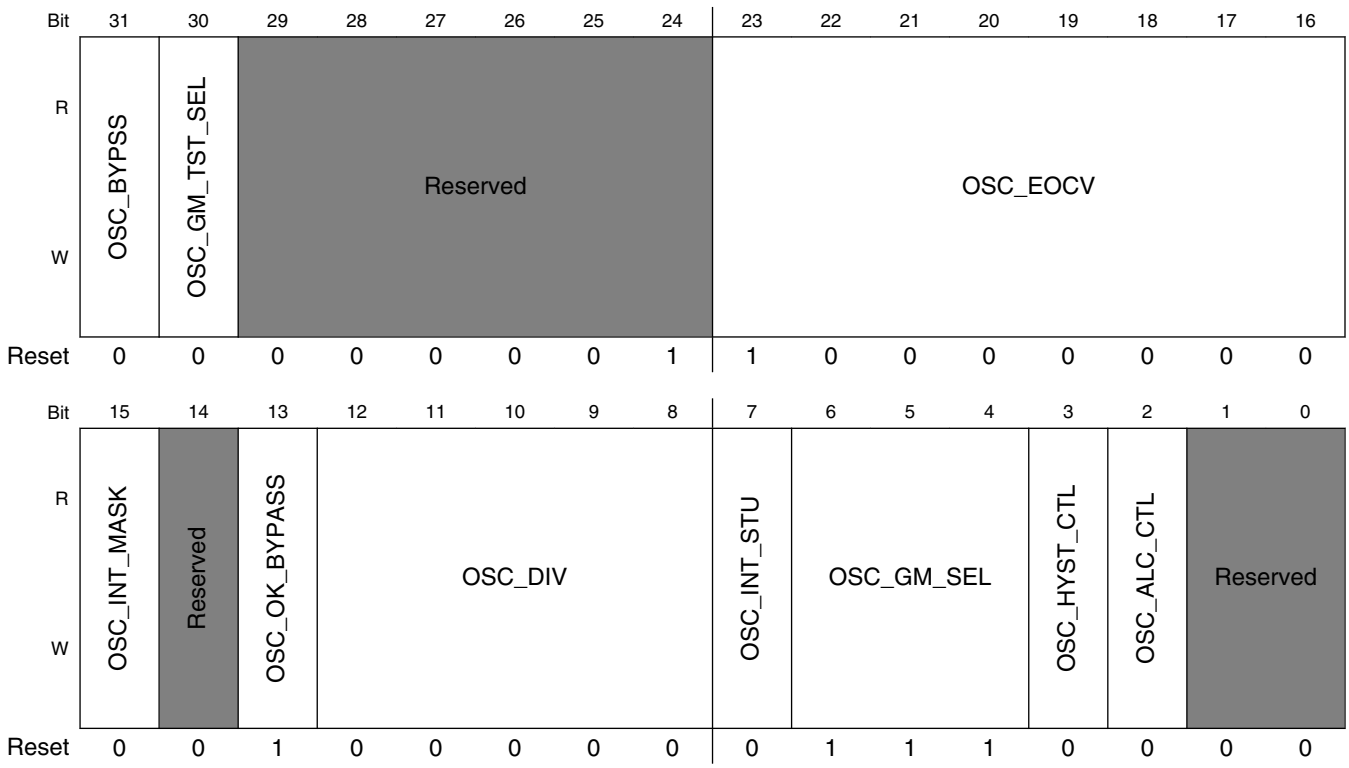
XTALOSC_OSC25M_CTL_CFG field descriptions

Field	Description
31 OSC_BYPSS	Crystal Oscillator bypass This bit specifies whether the oscillator should be bypassed or not. Software can only set this bit. System reset is needed to reset this bit. 0 Oscillator output is used as root clock. 1 EXTAL is used as root clock
30 OSC_GM_TST_SEL	Test mode GM measurement 0 Normal run mode 1 Enable test mode measurement of GM
29–24 -	This field is reserved. Reserved
23–16 OSC_EOCV	End of Count Value These bits specify the end of count value to be used for comparison by the oscillator stabilization counter OSCCNT after reset or whenever it is switched on from the off state. This counting period ensures that external oscillator clock signal is stable before it can be selected by the system. When oscillator counter reaches the value EOCV 512, oscillator available interrupt request is generated. The OSCCNT counter is kept under reset if oscillator bypass mode is selected
15 OSC_INT_MASK	Crystal oscillator clock interrupt mask This bit masks the I_OSC interrupt bit. 0 Crystal oscillator clock interrupt is masked 1 Crystal oscillator clock interrupt is enabled
14 -	This field is reserved. Reserved
13 OSC_OK_BYPASS	OSC ok output bypass
12–8 OSC_DIV	Crystal oscillator clock division factor These bits specify the crystal oscillator output clock division factor. The output clock is divided by the factor OSCDIV+1
7 OSC_INT_STU	Crystal oscillator clock interrupt This bit is set by hardware when OSCCNT counter reaches the count value EOCV x 512. It is cleared by software by writing 1. 0 No oscillator clock interrupt occurred 1 Oscillator clock interrupt pending
6–4 OSC_GM_SEL	Crystal overdrive protection
3 OSC_HYST_CTL	Hysteresis Control 0 Enable hysteresis control 1 Disable hysteresis control
2 OSC_ALC_CTL	Automatic Level Controller Enable 0 Enable automatic level controller 1 Disable automatic level controller
-	This field is reserved. Reserved

5.3.2.2 27M Oscillator Control Configuration Register (XTALOSC_OSC27M_CTL_CFG)

27M Oscillator Control Configuration Register

Address: 3027_0000h base + 8000h offset = 3027_8000h



XTALOSC_OSC27M_CTL_CFG field descriptions

Field	Description
31 OSC_BYPSS	Crystal Oscillator bypass This bit specifies whether the oscillator should be bypassed or not. Software can only set this bit. System reset is needed to reset this bit. 0 Oscillator output is used as root clock. 1 EXTAL is used as root clock
30 OSC_GM_TST_SEL	Test mode GM measurement 0 Normal run mode 1 Enable test mode measurement of GM
29–24 -	This field is reserved. Reserved
23–16 OSC_EOCV	End of Count Value These bits specify the end of count value to be used for comparison by the oscillator stabilization counter OSCCNT after reset or whenever it is switched on from the off state. This counting period ensures that external oscillator clock signal is stable before it can be selected by the system. When

Table continues on the next page...

XTALOSC_OSC27M_CTL_CFG field descriptions (continued)

Field	Description
	oscillator counter reaches the value EOCV 512, oscillator available interrupt request is generated. The OSCCNT counter is kept under reset if oscillator bypass mode is selected
15 OSC_INT_MASK	Crystal oscillator clock interrupt mask This bit masks the I_OSC interrupt bit. 0 Crystal oscillator clock interrupt is masked 1 Crystal oscillator clock interrupt is enabled
14 -	This field is reserved. Reserved
13 OSC_OK_ BYPASS	OSC ok output bypass
12–8 OSC_DIV	Crystal oscillator clock division factor These bits specify the crystal oscillator output clock division factor. The output clock is divided by the factor OSCDIV+1
7 OSC_INT_STU	Crystal oscillator clock interrupt This bit is set by hardware when OSCCNT counter reaches the count value EOCV x 512. It is cleared by software by writing 1. 0 No oscillator clock interrupt occurred 1 Oscillator clock interrupt pending
6–4 OSC_GM_SEL	Crystal overdrive protection
3 OSC_HYST_CTL	Hysteresis Control 0 Enable hysteresis control 1 Disable hysteresis control
2 OSC_ALC_CTL	Automatic Level Controller Enable 0 Enable automatic level controller 1 Disable automatic level controller
-	This field is reserved. Reserved

5.4 Thermal Management Unit (TMU)

5.4.1 TMU as implemented on the chip

The local sensors for each channel are put at different locations on the chip to have more accurate temperature monitoring for certain IP. Since the die size is not big, only 3 local sensors are used on this chip. The sensors are put next to high-power modules such as CPU, GPU. The table below shows the local sensor placement.

Table 5-12. Local Sensor Placement

Temperature	Sensor
0	ARM
1	GPU
2	VPU
3-15	Reserved

5.4.2 Thermal Monitoring Unit Introduction

The Thermal Monitoring Unit (TMU) monitors and reports the temperature from one or more remote temperature measurement sites located on chip.

5.4.2.1 TMU Overview

The TMU has access to multiple temperature measurement sites strategically located on the chip. It monitors these sites and can signal an alarm if a programmed threshold is ever exceeded. The upper and lower temperature range is continuously captured. A set of reporting registers allow for reading the current temperature at monitored sites.

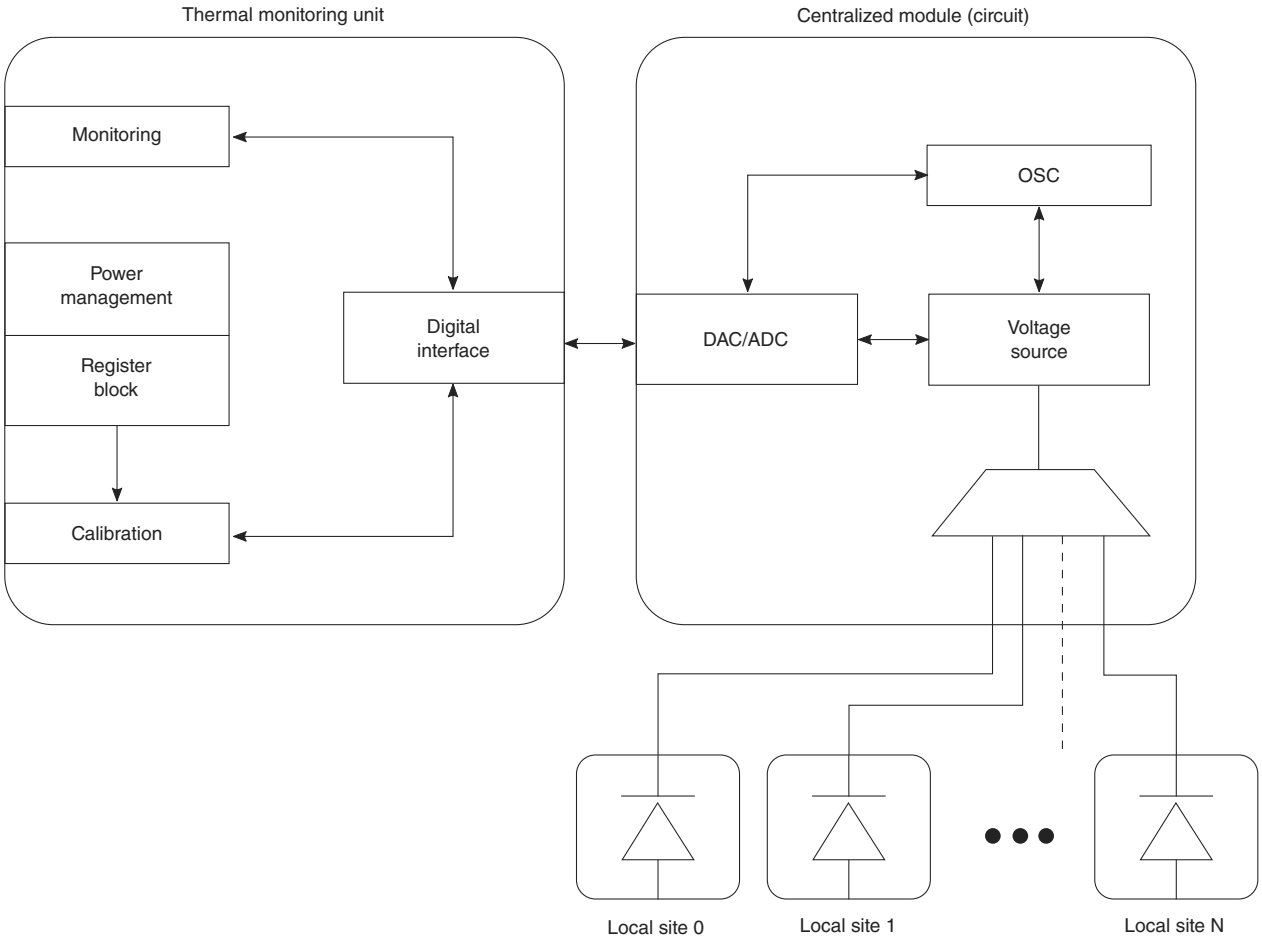


Figure 5-17. Thermal Monitoring Unit Block Diagram

5.4.2.2 Features

The temperature management unit features:

- Temperature measurement range 0-85°C.
- Calibration
  - Calibration table loaded from boot code ROM using pre-boot loader *or* initialization of calibration table through software
- Monitoring
  - Single-, or multi-site monitoring
  - Programmable monitoring interval
  - Out-of-range indication
  - High/low temperature range monitoring
  - Immediate and average temperature monitoring

- Average temperature monitoring programmable low-pass filtering
- Programmable monitoring thresholds for normal and critical alarm

- Reporting
  - Immediate and average temperature reporting for all monitor sites

5.4.2.3 Modes of Operation

The TMU has one mode of operation:

- Monitoring

The mode register monitoring enable bit, TMR[ME], determines if the unit is in active monitoring mode or in power saving mode.

The table below describes bit settings required for each TMU mode of operation.

Table 5-13. TMU Mode Bit Setting

Modes with Features	TMR[ME]
Monitoring mode disabled	0
Monitoring mode enabled	1

5.4.3 TMU register descriptions

The table shows the memory map for management of the TMU resources.

5.4.3.1 TMU Memory map

TMU base address: 3026_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	TMU mode register (TMR)	32	RW	0000_0000h
4h	TMU status register (TSR)	32	RO	0000_0000h
8h	TMU monitor temperature measurement interval register (TMTMIR)	32	RW	0000_0000h
20h	TMU interrupt enable register (TIER)	32	RW	0000_0000h
24h	TMU interrupt detect register (TIDR)	32	W1C	0000_0000h
28h	TMU interrupt site capture register (TISCR)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2Ch	TMU interrupt critical site capture register (TICSCR)	32	RW	0000_0000h
40h	TMU monitor high temperature capture register (TMHTCR)	32	RO	0000_0000h
44h	TMU monitor low temperature capture register (TMLTCR)	32	RO	0000_0000h
50h	TMU monitor high temperature immediate threshold register (TMHTITR)	32	RW	0000_0000h
54h	TMU monitor high temperature average threshold register (TMHTATR)	32	RW	0000_0000h
58h	TMU monitor high temperature average critical threshold register (TMHTACTR)	32	RW	0000_0000h
80h	TMU temperature configuration register (TTCFGR)	32	RW	0000_0000h
84h	TMU sensor configuration register (TSCFGR)	32	RW	0000_0000h
100h	TMU report immediate temperature site register 0 (TRITSR0)	32	RO	0000_0000h
104h	TMU report average temperature site register 0 (TRATSR0)	32	RO	0000_0000h
110h	TMU report immediate temperature site register 1 (TRITSR1)	32	RO	0000_0000h
114h	TMU report average temperature site register 1 (TRATSR1)	32	RO	0000_0000h
120h	TMU report immediate temperature site register 2 (TRITSR2)	32	RO	0000_0000h
124h	TMU report average temperature site register 2 (TRATSR2)	32	RO	0000_0000h
F10h	TMU temperature range 0 control register (TTR0CR)	32	RW	000B_0000h
F14h	TMU temperature range 1 control register (TTR1CR)	32	RW	000A_0026h
F18h	TMU temperature range 2 control register (TTR2CR)	32	RW	0008_0048h
F1Ch	TMU temperature range 3 control register (TTR3CR)	32	RW	0007_0061h

5.4.3.2 TMU mode register (TMR)

5.4.3.2.1 Offset

Register	Offset
TMR	0h

5.4.3.2.2 Function

The TMU mode register allows software to control the operation of the thermal monitoring.

5.4.3.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ME	Reserved			ALPF		Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSITE			Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.2.4 Fields

Field	Function
31 ME	Monitoring mode enable. 0 No monitoring. 1 Monitoring of sites as defined by field MSITE. Before enabling the TMU for monitoring, the TMU must be configured, see section Initialization Information. Failure to properly initialize the configuration table may result in boundedly undefined behavior.
30-28 —	Reserved
27-26 ALPF	Average low pass filter setting. 00 1.0 01 0.5 10 0.25 11 0.125 The average temperature is calculated as: ALPF x Current_Temp + (1 - ALPF) x Average_Temp. If no previous (average) temperature is valid, current temperature is used. For proper operation, this field should only change when monitoring is disabled.
25-16 —	Reserved
15-13 MSITE	Monitoring site select 0 - 2. By setting the select bit for a temperature sensor site, it is enabled and included in all monitoring functions. For proper operation, this field should only change when monitoring is disabled. If no site is selected, site 0 is monitored by default.
12-0 —	Reserved

5.4.3.3 TMU status register (TSR)

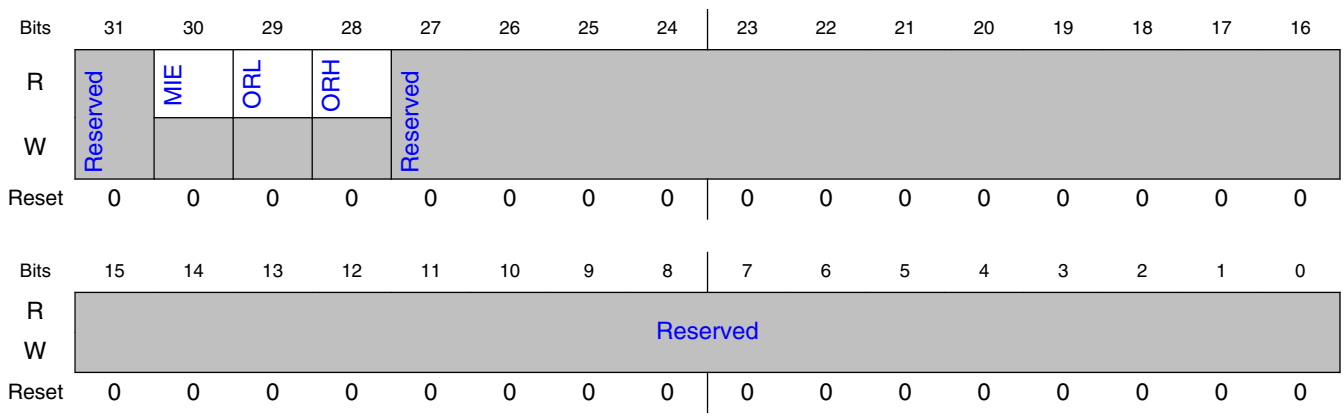
5.4.3.3.1 Offset

Register	Offset
TSR	4h

5.4.3.3.2 Function

The TMU status register reports the monitoring and calibration status during operation.

5.4.3.3.3 Diagram



5.4.3.3.4 Fields

Field	Function
31 —	Reserved
30 MIE	Monitoring interval exceeded. 0 Monitoring interval not exceeded. 1 Monitoring interval exceeded. The time required to perform measurement of all monitored sites has exceeded the monitoring interval as defined by TMTMIR. This bit will clear automatically when TMU monitoring is (re-)enabled or the monitoring interval register, TMTMIR, is written.
29 ORL	Out-of-range low temperature measurement detected. A temperature sensor detected a temperature reading below the lowest measurable temperature of 0 degrees Celsius. This bit will clear automatically when TMU monitoring is (re-)enabled.

Table continues on the next page...

Field	Function
28 ORH	Out-of-range high temperature measurement detected. A temperature sensor detected a temperature reading above the highest measurable temperature of 85 °C. This bit will clear automatically when TMU monitoring is (re-)enabled.
27-0 —	Reserved

5.4.3.4 TMU monitor temperature measurement interval register (TMTMIR)

5.4.3.4.1 Offset

Register	Offset
TMTMIR	8h

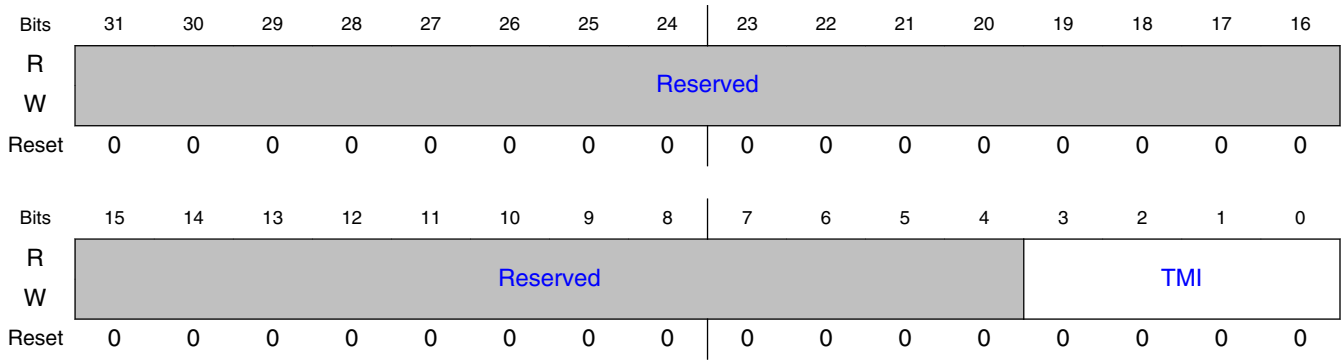
5.4.3.4.2 Function

The TMU monitor temperature measurement interval register determines at what frequency temperature sensors are read. All enabled monitored sites are read once in the duration of the time interval. The status bit TSR[MIE] will be set if the temperature measument takes longer than the set interval. Software should consider increasing the interval or reducing the number of active sites if the interval is exceeded. Disabling the interval allows for continuous monitoring.

NOTE

The time it takes for one temperature measurement is dependent on the temperatures measured and mode settings, thus there is no fixed time advertised for a single temperature measurements.

5.4.3.4.3 Diagram



5.4.3.4.4 Fields

Field	Function
31-4 —	Reserved
3-0 TMI	Temperature monitoring interval in seconds. For proper operation, this field should only change when monitoring is disabled, TMR[ME]=0. For lower platform speeds, the time increases proportionally, while the opposite is true for higher platform speeds.

Setting	Platform Clock Frequency			
	333 MHz	400 MHz	667 MHz	800 MHz
0000	0.03 s	0.02 s	0.015 s	0.01 s
0001	0.06 s	0.04 s	0.03 s	0.02 s
0010	0.10 s	0.08 s	0.05 s	0.04 s
0011	0.20 s	0.17 s	0.10 s	0.08 s
0100	0.40 s	0.34 s	0.20 s	0.17 s
0101	0.80 s	0.67 s	0.40 s	0.34 s
0110	1.60 s	1.34 s	0.80 s	0.67 s
0111	3.2 s	2.7 s	1.6 s	1.34 s
1000	6.4 s	5.4 s	3.2 s	2.7 s
1001	12.8 s	10.7 s	6.4 s	5.4 s
1010	25.8 s	21.5 s	12.9 s	10.7 s
1011	51.6 s	42.9 s	25.8 s	21.5 s
1100	103 s	85.9 s	51.5 s	42.9 s
1101	206.0 s	171.8 s	103.0 s	85.9 s
1110	412.2 s	343.6 s	206.1 s	171.8 s
1111	Disabled			

5.4.3.5 TMU interrupt enable register (TIER)

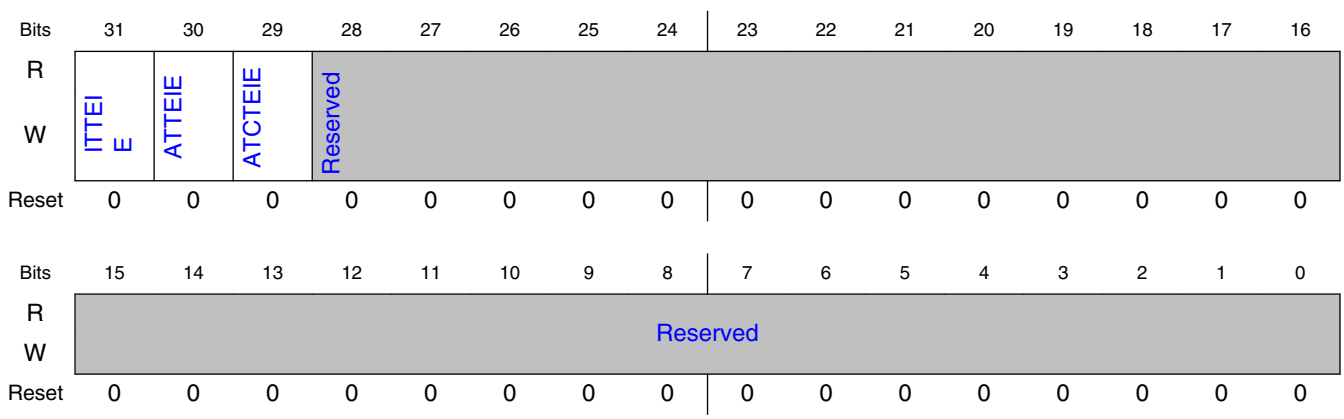
5.4.3.5.1 Offset

Register	Offset
TIER	20h

5.4.3.5.2 Function

The TMU interrupt enable register determines if a detected status condition should cause a system interrupt. A system interrupt occurs if a bit in this register is set and the corresponding bit in the interrupt detect register is also set. To clear the interrupt, write a 1 to the interrupt detect register.

5.4.3.5.3 Diagram



5.4.3.5.4 Fields

Field	Function
31 ITTEIE	Immediate temperature threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ITTE] is set.
30 ATTEIE	Average temperature threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATTE] is set.

Table continues on the next page...

Field	Function
29 ATCTEIE	Average temperature critical threshold exceeded interrupt enable. 0 Disabled. 1 Interrupt enabled. Generate an interrupt if TIDR[ATCTE] is set.
28-0 —	Reserved

5.4.3.6 TMU interrupt detect register (TIDR)

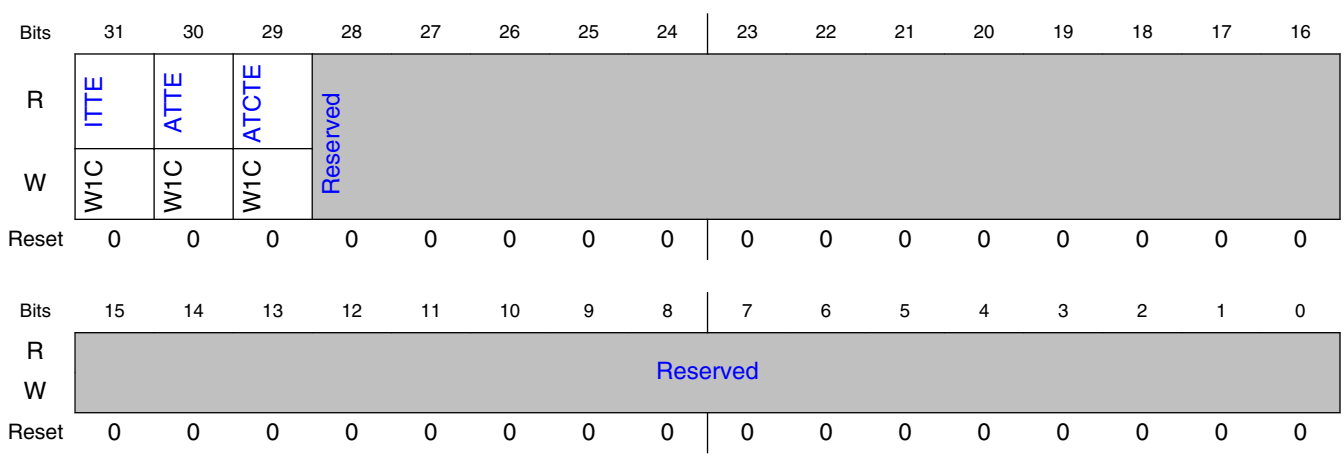
5.4.3.6.1 Offset

Register	Offset
TIDR	24h

5.4.3.6.2 Function

The TMU interrupt detect register indicates if an status condition was detected that could generate an interrupt. Write 1 to clear the detected condition and the interrupt, if enabled.

5.4.3.6.3 Diagram



5.4.3.6.4 Fields

Field	Function
31	Immediate temperature threshold exceeded. Write 1 to clear.

Table continues on the next page...

Field	Function
ITTE	0 No threshold exceeded. 1 Immediate temperature threshold, as defined by TMHTITR, has been exceeded by one or more monitored sites. This includes an out-of-range measured temperature above 85 °C. The sites which has exceeded the threshold are captured in TISCR[ISITE].
30 ATTE	Average temperature threshold exceeded. Write 1 to clear. 0 No threshold exceeded. 1 Average temperature threshold, as defined by TMHTATR, has been exceeded by one or more monitored sites. The sites which has exceeded the threshold are captured in TISCR[ASITE].
29 ATCTE	Average temperature critical threshold exceeded. Write 1 to clear. 0 No threshold exceeded. 1 Average temperature critical threshold, as defined by TMHTACTR, has been exceeded by one or more monitored sites. The sites which has exceeded the threshold are captured in TICSCR[CASITE].
28-0 —	Reserved

5.4.3.7 TMU interrupt site capture register (TISCR)

5.4.3.7.1 Offset

Register	Offset
TISCR	28h

5.4.3.7.2 Function

The TMU interrupt site capture register holds information about the temperature sensor site associated with a detected interrupt event.

5.4.3.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ISITE			Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ASITE			Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.7.4 Fields

Field	Function
31-29 ISITE	Temperature sensor site associated with the setting of TIDR[ITTE]. This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected interrupt events ITTE.
28-16 —	Reserved
15-13 ASITE	Temperature sensor site associated with the setting of TIDR[ATTE] . This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected interrupt event ATTE.
12-0 —	Reserved

5.4.3.8 TMU interrupt critical site capture register (TICSCR)

5.4.3.8.1 Offset

Register	Offset
TICSCR	2Ch

5.4.3.8.2 Function

The TMU interrupt critical site capture register holds information about the temperature sensor site associated with a detected critical interrupt event.

5.4.3.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CASITE			Reserved												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.8.4 Fields

Field	Function
31-16 —	Reserved
15-13 CASITE	Temperature sensor site associated with the setting of TIDR[ATCTE] . This field has the same bit representation as TMR[MSITE]. Software should clear this field after handling the detected critical interrupt event ATCTE.
12-0 —	Reserved

5.4.3.9 TMU monitor high temperature capture register (TMHTCR)

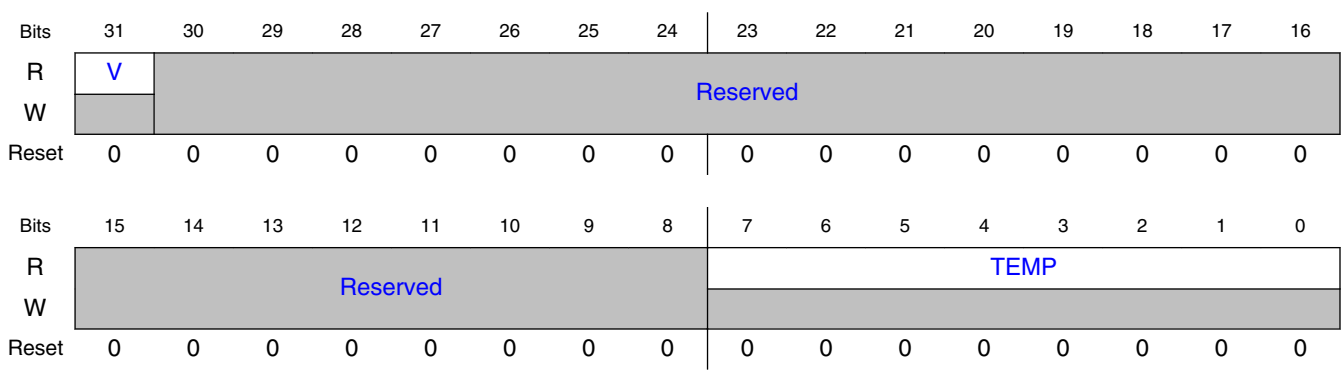
5.4.3.9.1 Offset

Register	Offset
TMHTCR	40h

5.4.3.9.2 Function

This TMU monitor register captures and record the highest temperature reached for any one enabled monitored site within temperature sensor range.

5.4.3.9.3 Diagram



5.4.3.9.4 Fields

Field	Function
31 V	Valid reading. 0 Temperature reading is not valid due to no measured temperature within the sensor range of 0-85 °C for an enabled monitored site. 1 Temperature reading is valid. (Re-)enabling the TMU will automatically clear this bit and start a new search.
30-8 —	Reserved
7-0 TEMP	Highest temperature recorded in degrees Celcius by any enabled monitored site. Valid when V=1. 0-85 °C Sensor range 86-255 °C Reserved

5.4.3.10 TMU monitor low temperature capture register (TMLTCR)

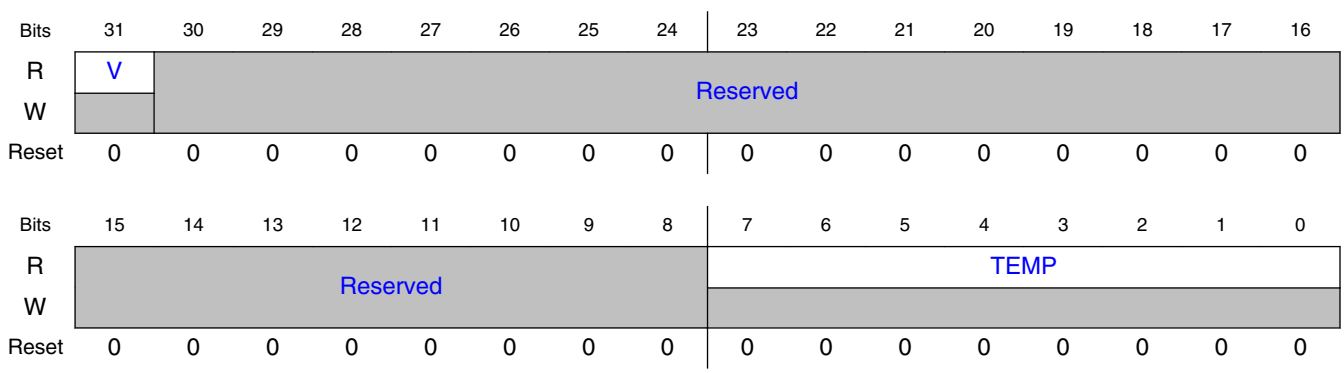
5.4.3.10.1 Offset

Register	Offset
TMLTCR	44h

5.4.3.10.2 Function

This TMU monitor register captures and record the lowest temperature reached for any one enabled monitored site within temperature sensor range.

5.4.3.10.3 Diagram



5.4.3.10.4 Fields

Field	Function
31 V	Valid reading. 0 Temperature reading is not valid due to no measured temperature within the sensor range of 0-85 °C for an enabled monitored site. 1 Temperature reading is valid. (Re-)enabling the TMU will automatically clear this bit and start a new search.
30-8 —	Reserved
7-0 TEMP	Lowest temperature recorded in degrees Celcius by any enabled monitored site. Valid when V=1. 0-85 °C Sensor range 86-255 °C Reserved

5.4.3.11 TMU monitor high temperature immediate threshold register (TMHTITR)

5.4.3.11.1 Offset

Register	Offset
TMHTITR	50h

5.4.3.11.2 Function

This TMU monitor register determines the high current temperature threshold for generating the TIDR[ITTE] event.

5.4.3.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.11.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved
7-0 TEMP	High temperature immediate threshold value. Determines the current upper temperature threshold, for any enabled monitored site, that if exceeded will cause TIDR[ITTE] to be set when EN=1. 0-85 °C Sensor range 86-255 °C Reserved

5.4.3.12 TMU monitor high temperature average threshold register (TMHTATR)

5.4.3.12.1 Offset

Register	Offset
TMHTATR	54h

5.4.3.12.2 Function

This TMU monitor register determines the high average temperature threshold for generating the TIDR[ATTE] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating average temperature.

5.4.3.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.12.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved
7-0 TEMP	High temperature average threshold value. Determines the average upper temperature threshold, for any enabled monitor site, that if exceeded will cause TIDR[ATTE] to be set when EN=1. 0-85 °C Sensor range 86-255 °C Reserved

5.4.3.13 TMU monitor high temperature average critical threshold register (TMHTACTR)

5.4.3.13.1 Offset

Register	Offset
TMHTACTR	58h

5.4.3.13.2 Function

This TMU monitor register determines the high average critical temperature threshold for generating the TIDR[ATCTE] event. The low-pass filter setting, TMR[ALPF], determines the function for calculating average temperature.

5.4.3.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.13.4 Fields

Field	Function
31 EN	Enable threshold. 0 Disabled. 1 Threshold enabled.
30-8 —	Reserved
7-0 TEMP	High temperature average critical threshold value. Determines the average upper critical temperature threshold, for any enabled monitored site, that if exceeded will cause TIDR[ATCTE] to be set when EN=1. 0-85 °C Sensor range 86-255 °C Reserved

5.4.3.14 TMU temperature configuration register (TTCFGR)

5.4.3.14.1 Offset

Register	Offset
TTCFGR	80h

5.4.3.14.2 Function

The TMU temperature configuration register, in conjunction with the sensor configuration register, is used to initialize the internal sensor translation table used during monitoring. This register pair defines indirect access to the table. See the section Initialization Information.

5.4.3.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.14.4 Fields

Field	Function
31-0 DATA	Sensor data.

5.4.3.15 TMU sensor configuration register (TSCFGR)

5.4.3.15.1 Offset

Register	Offset
TSCFGR	84h

5.4.3.15.2 Function

The TMU sensor configuration register, in conjunction with the temperature configuration register, is used to initialize the internal sensor translation table used during monitoring. This register pair defines indirect access to the table. Reading this register will return the data from the translation table as defined by TTCFGR. See the section Initialization Information.

5.4.3.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.15.4 Fields

Field	Function
31-0 DATA	Sensor data.

5.4.3.16 TMU report immediate temperature site register a (TRITSR0 - TRITSR2)

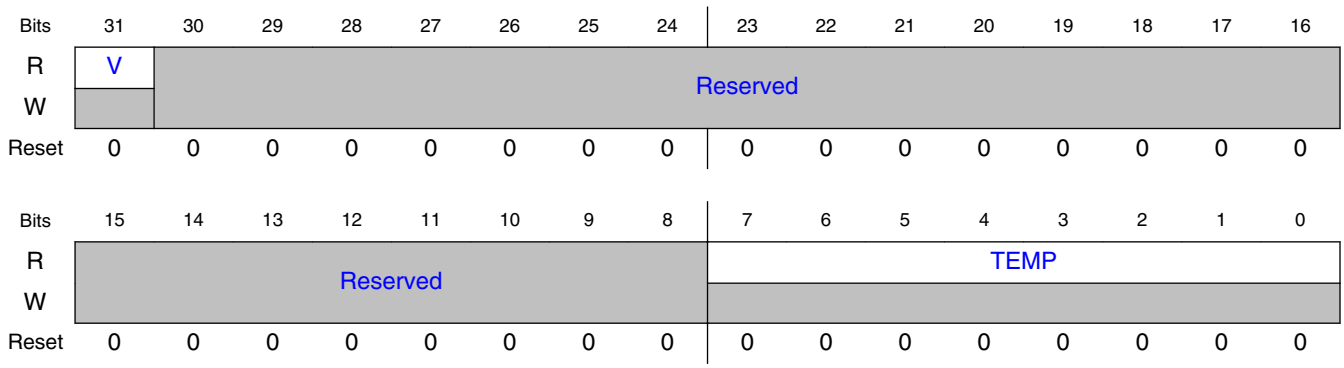
5.4.3.16.1 Offset

Register	Offset
TRITSR0	100h
TRITSR1	110h
TRITSR2	120h

5.4.3.16.2 Function

This TMU report register returns the last measured temperature at site. The site must be part of the list of enabled monitored sites as defined by TMR[MSITE].

5.4.3.16.3 Diagram



5.4.3.16.4 Fields

Field	Function
31 V	Valid measured temperature. 0 Not valid. Temperature out of sensor range or first measurement still pending. 1 Valid.
30-8 —	Reserved
7-0 TEMP	Last temperature reading at site when V=1.

5.4.3.17 TMU report average temperature site register a (TRATSR0 - TRATSR2)

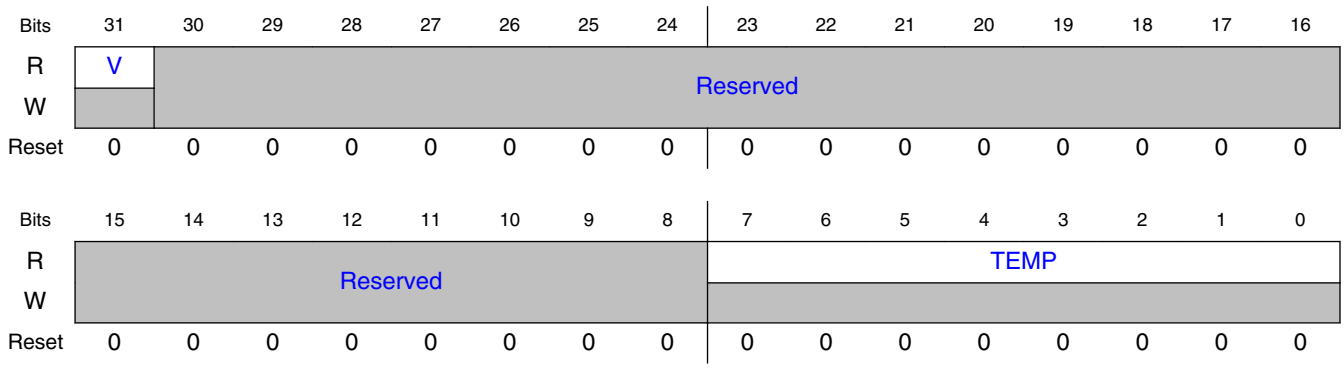
5.4.3.17.1 Offset

Register	Offset
TRATSR0	104h
TRATSR1	114h
TRATSR2	124h

5.4.3.17.2 Function

This TMU report register returns the average measured temperature at site. The site must be part of the list of enabled monitored sites as defined by TMR[MSITE].

5.4.3.17.3 Diagram



5.4.3.17.4 Fields

Field	Function
31 V	Valid measured temperature. 0 Not valid. Temperature out of sensor range or first measurement still pending. 1 Valid.
30-8 —	Reserved
7-0 TEMP	Average temperature reading at site when V=1.

5.4.3.18 TMU temperature range 0 control register (TTR0CR)

5.4.3.18.1 Offset

Register	Offset
TTR0CR	F10h

5.4.3.18.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 5-14. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

5.4.3.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

5.4.3.18.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-0 TEMP	Starting temperature in Celsius for range.

5.4.3.19 TMU temperature range 1 control register (TTR1CR)

5.4.3.19.1 Offset

Register	Offset
TTR1CR	F14h

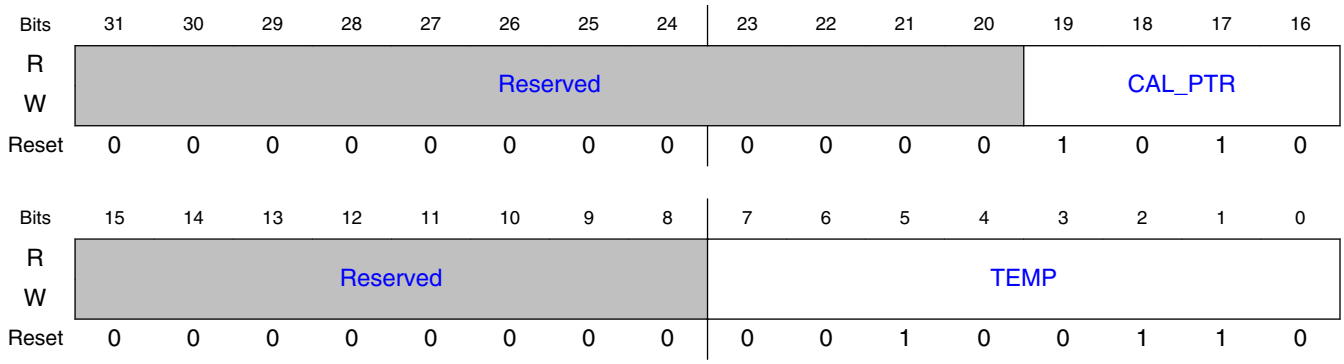
5.4.3.19.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 5-15. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

5.4.3.19.3 Diagram



5.4.3.19.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

5.4.3.20 TMU temperature range 2 control register (TTR2CR)

5.4.3.20.1 Offset

Register	Offset
TTR2CR	F18h

5.4.3.20.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 5-16. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

5.4.3.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

5.4.3.20.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8	Reserved

Table continues on the next page...

Field	Function
—	
7-0 TEMP	Starting temperature in Celsius for range.

5.4.3.21 TMU temperature range 3 control register (TTR3CR)

5.4.3.21.1 Offset

Register	Offset
TTR3CR	F1Ch

5.4.3.21.2 Function

The TMU temperature range control registers allow for defining the four temperature ranges that each sensor covers. A starting temperature range and number of calibration points, up to 16, is given. The temperature ranges may overlap to remove possible sample error, when searching for an accurate sensor reading. The default temperature calibration points are shown below.

Table 5-17. Default Temperature Configuration Points

Configuration Range	Starting Temperature (Celsius)	Temperature Configuration Points (Celsius)	Number of Points
0	0	0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44	12
1	38	38, 42, 46, 50, 54, 58, 62, 66, 70, 74, 78	11
2	72	72, 76, 80, 84, 88, 92, 96, 100, 104	9
3	97	97, 101, 105, 109, 113, 117, 121, 125	8

5.4.3.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												CAL_PTR			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TEMP							
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1

5.4.3.21.4 Fields

Field	Function
31-20 —	Reserved
19-16 CAL_PTR	Temperature configuration points. 0000 Reserved 0001 2 points 0010 3 points ... 1111 16 points
15-8 —	Reserved
7-0 TEMP	Starting temperature in Celsius for range.

5.4.4 Functional Description

The following sections describe the functionality of the TMU in details.

5.4.4.1 Monitoring

Monitoring is the process of reading enabled temperature sensor sites on-chip at regular intervals and taking appropriate actions, such as alarming the user when the temperature exceeds a programmed temperature threshold.

During monitoring, all enabled temperature sensor sites are periodically measured for temperature starting with site 0 and ending with site 2. The interval at which the sensors are read is set in TMTMIR and should reflect the maximum interval time required to accurately capture temperature changes. If the measurement interval has not expired after the last active site has been read, the sensor logic enters low-power mode. If the interval has expired when the last active site is read, the measurement interval exceeded bit, TSR[MIE], is set and the next active site is read immediately. If the interval has been exceeded, user may opt to reduce number of sites monitored or increase the interval, if possible.

For each site the current and average temperature is logged. The average temperature is calculated based on the low-pass filter function in the mode register. If any of the set temperature threshold registers are exceeded, the corresponding interrupt detect bit is set in TIDR. Interrupts are enabled through the interrupt enable register, TIER.

Process for enabling monitoring mode:

1. Clear the interrupt detect register, TIDR.
2. Clear interrupt site capture register (TISCR) and interrupt critical site capture register (TICSCR).
3. Enable interrupt handling by setting the appropriate bits in TIER.
4. Set the temperature threshold registers TMHTITR, TMHTATR and TMHTACTR.
5. Set the monitoring interval register, TMTMIR.
6. Enable monitor mode by setting TMR[ME]=1. Sites to monitor are controlled by setting TMR[MSITE]. There should be at least one active site enabled. Set other mode control bits as needed.
7. If the monitoring interval is too short as indicated by TSR[MIE], the interval may need to be increased or number of sites reduced.

5.4.4.2 Reporting

The TMU can directly report the current and average temperature for a particular temperature sensor site during monitoring mode by reading one of the report registers per site. The report uses the last measurement done by the monitoring process and requires a site to be actively monitored for accurate reading. Reading a site which has last measured an invalid temperature outside the sensor range of 0-85 degrees Celsius, will have the valid bit cleared.

If monitoring is disabled, the last temperature measurement remains for the site(s) previously monitored and can still be read using the report registers knowing that the temperature reported is no longer accurate. This method can be used to capture the temperature at multiple sites in time, but does not allow for continuous monitoring.

Chapter 6
SNVS, Reset, Fuse, and Boot

6.1 System Boot

6.1.1 Overview

The boot process begins at the Power-On Reset (POR) where the hardware reset logic forces the Arm core to begin the execution starting from the on-chip boot ROM.

The boot ROM code uses the state of the internal register BOOT_MODE[1:0] as well as the state of various eFUSEs and/or GPIO settings to determine the boot flow behavior of the device.

The main features of the ROM include:

- Support for booting from various boot devices
- Serial downloader support (USB OTG)
- Device Configuration Data (DCD) and plugin
- Wake-up from the low-power modes

The boot ROM supports these boot devices:

- NAND flash
- SD/MMC

The boot ROM uses the state of the BOOT_MODE and eFUSEs to determine the boot device. For development purposes, the eFUSEs used to determine the boot device may be overridden using the GPIO pin inputs.

The boot ROM code also allows to download the programs to be run on the device. The example is a provisioning program that can make further use of the serial connection to provide a boot device with a new image. Typically, the provisioning program is downloaded to the internal RAM and allows to program the boot devices, such as the SD/MMC flash. The ROM serial downloader uses a high-speed USB in a non-stream mode connection.

The Device Configuration Data (DCD) feature allows the boot ROM code to obtain the SOC configuration data from an external program image residing on the boot device. As an example, the DCD can be used to program the DDR controller for optimal settings, improving the boot performance. The DCD is restricted to the memory areas and peripheral addresses that are considered essential for the boot purposes (see Write data command).

A key feature of the boot ROM is the ability to perform a secure boot, also known as a High-Assurance Boot (HAB). This is supported by the HAB security library which is a subcomponent of the ROM code. The HAB uses a combination of hardware and software together with the Public Key Infrastructure (PKI) protocol to protect the system from executing unauthorized programs. Before the HAB allows the user image to execute, the image must be signed. The signing process is done during the image build process by the private key holder and the signatures are then included as a part of the final program image. If configured to do so, the ROM verifies the signatures using the public keys included in the program image. A secure boot with HAB can be performed on all boot devices supported on the chip in addition to the serial downloader. The HAB library in the boot ROM also provides the API functions, allowing the additional boot chain components (bootloaders) to extend the secure boot chain. The out-of-fab setting for the SEC_CONFIG is the open configuration, in which the ROM/HAB performs the image authentication, but all authentication errors are ignored and the image is still allowed to execute.

6.1.2 Boot modes

During reset, the chip checks the power gating controller status register.

During boot, the core's behavior is defined by the boot mode pin settings, as described in Boot mode pin settings. When waking up from the low-power boot mode, the core skips the clock settings. The boot ROM checks that the PERSISTENT_ENTRY0 (see Persistent bits) is a pointer to a valid address space (OCRAM, DDR, or EIM). If the PERSISTENT_ENTRY0 is a pointer to a valid range, it starts the execution using the entry point from the PERSISTENT_ENTRY0 register. If the PERSISTENT_ENTRY0 is a pointer to an invalid range, the core performs the system reset.

6.1.2.1 Boot mode pin settings

The device has four boot modes (one is reserved for NXP use). The boot mode is selected based on the binary value stored in the internal BOOT_MODE register.

The BOOT_MODE is initialized by sampling the BOOT_MODE0 and BOOT_MODE1 inputs on the rising edge of the POR_B. After these inputs are sampled, their subsequent state does not affect the contents of the BOOT_MODE internal register. The state of the internal BOOT_MODE register may be read from the BMOD[1:0] field of the SRC Boot Mode Register (SRC_SBMR2). The available boot modes are: Boot From Fuses, serial boot via USB, and Internal Boot. See this table for settings:

Table 6-1. Boot MODE pin settings

BOOT_MODE[1:0]	Boot Type
00	Boot From Fuses
01	Serial Downloader
10	Internal Boot
11	Reserved

6.1.2.2 High-level boot sequence

The figure found here show the high-level boot ROM code flow.

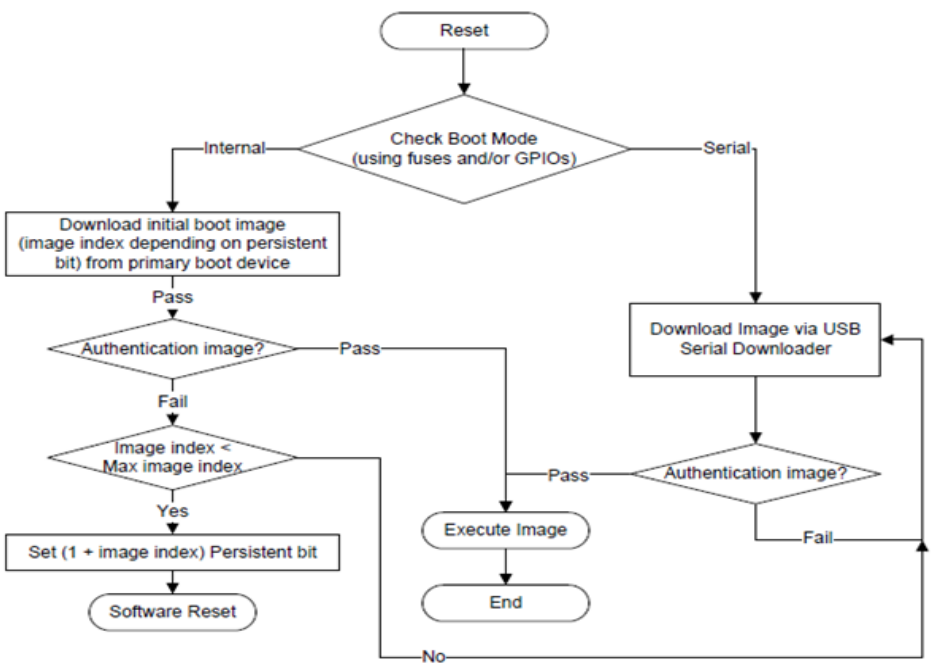


Figure 6-1. Boot flow

6.1.2.3 Boot From Fuses mode (BOOT_MODE[1:0] = 00b)

A value of 00b in the BOOT_MODE[1:0] register selects the Boot From Fuses mode.

This mode is similar to the Internal Boot mode described in Internal Boot mode (BOOT_MODE[1:0] = 0b10) with one difference. In this mode, the GPIO boot override pins are ignored. The boot ROM code uses the boot eFUSE settings only. This mode also supports a secure boot using HAB.

If set to Boot From Fuses, the boot flow is controlled by the BT_FUSE_SEL eFUSE value. If BT_FUSE_SEL = 0, indicating that the boot device (for example, flash, SD/MMC) was not programmed yet, the boot flow jumps directly to the Serial Downloader. If BT_FUSE_SEL = 1, the normal boot flow is followed, where the ROM attempts to boot from the selected boot device.

The first time a board is used, the default eFUSEs may be configured incorrectly for the hardware on the platform. In such case, the Boot ROM code may try to boot from a device that does not exist. This may cause an electrical/logic violation on some pads. Using the Boot From Fuses mode addresses this problem.

Setting the BT_FUSE_SEL=0 forces the ROM code to jump directly to the Serial Downloader. This allows a bootloader to be downloaded which can then provision the boot device with a program image and blow the BT_FUSE_SEL and the other boot configuration eFUSEs. After the reset, the boot ROM code determines that the BT_FUSE_SEL is blown (BT_FUSE_SEL = 1) and the ROM code performs an internal boot according to the new eFUSE settings. This allows the user to set BOOT_MODE[1:0]=00b on a production device and burn the fuses on the same device (by forcing the entry to the Serial Downloader), without changing the value of the BOOT_MODE[1:0] or the pullups/pulldowns on the BOOT_MODE pins.

6.1.2.4 Serial Downloader

The Serial Downloader provides a means to download a program image to the chip over the USB serial connection.

In this mode, the ROM programs the WDOG1 for a time-out specified by the fuse WDOG Time-out Select (See fusemap for details) if the WDOG_ENABLE eFuse is 1 and continuously polls for the USB connection. If no activity is found on the USB OTG1 and the watchdog timer expires, the Arm core is reset.

NOTE

After the downloaded image is loaded, it is responsible for managing the watchdog resets properly.

This figure shows the USB boot flow:

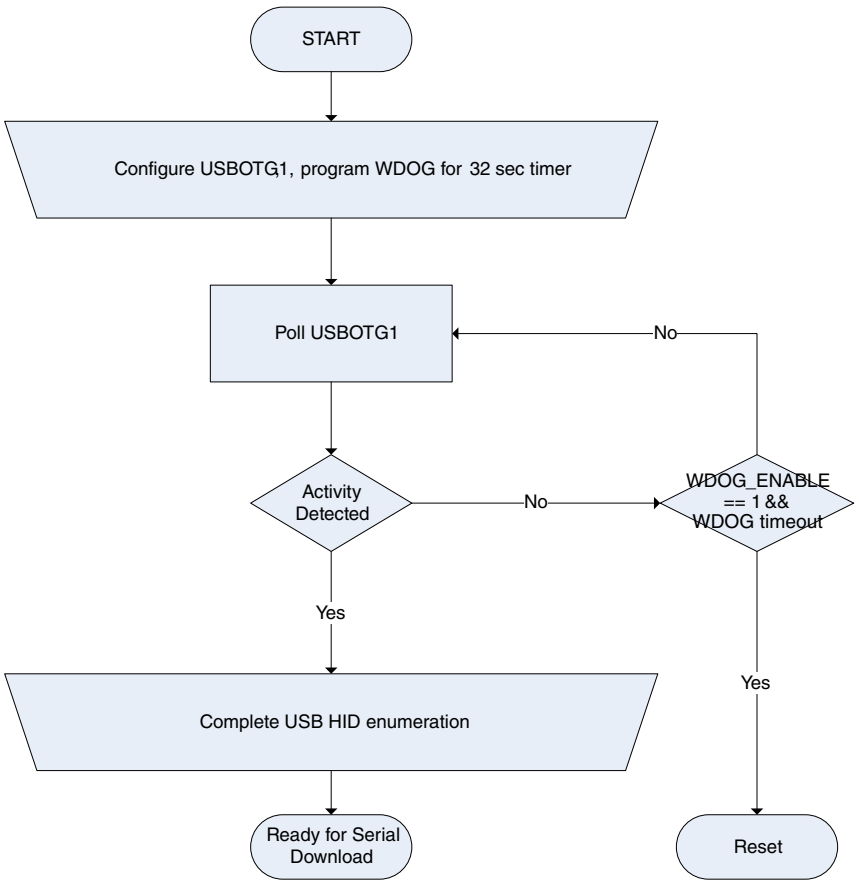


Figure 6-2. Serial Downloader boot flow

NOTE

Before going into USB serial mode, Boot ROM detect SD/MMC card on USDHC2 port. If a card is inserted, ROM will try to boot from it. This is the so-called Manufacture SD/MMC boot. This feature can be disabled by blowing fuse "Disable SD/MMC Manufacture Mode". See SD/MMC manufacture mode for details.

6.1.2.5 Internal Boot mode (BOOT_MODE[1:0] = 0b10)

A value of 0b10 in the BOOT_MODE[1:0] register selects the Internal Boot mode. In this mode, the processor continues to execute the boot code from the internal boot ROM.

The boot code performs the hardware initialization, loads the program image from the chosen boot device, performs the image validation using the HAB library (see Boot security settings), and then jumps to an address derived from the program image. If an error occurs during the internal boot, the boot code jumps to the Serial Downloader (see Serial Downloader). A secure boot using the HAB is possible in all the three boot modes.

When set to the Internal Boot, the boot flow may be controlled by a combination of eFUSE settings with an option of overriding the fuse settings using the General Purpose I/O (GPIO) pins. The GPIO Boot Select FUSE (BT_FUSE_SEL) determines whether the ROM uses the GPIO pins for a selected number of configuration parameters or eFUSEs in this mode.

- If BT_FUSE_SEL = 1, all boot options are controlled by the eFUSEs described in Table 6-2.
- If BT_FUSE_SEL = 0, the specific boot configuration parameters may be set using the GPIO pins rather than eFUSEs. The fuses that can be overridden when in this mode are indicated in the GPIO column of Table 6-2. Table 6-3 provides the details of the GPIO pins.

The use of the GPIO overrides is intended for development since these pads are used for other purposes in the deployed products. NXP recommends controlling the boot configuration by the eFUSEs in the deployed products and reserving the use of the GPIO mode for the development and testing purposes only.

6.1.2.6 Boot security settings

The internal boot modes use one of three security configurations.

- Closed: This level is intended for use with shipping-secure products. All HAB functions are executed and the security hardware is initialized (the Security Controller or SNVS enters the Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, and the boot flow is aborted with the control being passed to the serial downloader. At this level, the execution does not leave the internal ROM unless the target executable image is authenticated.
- Open: This level is intended for use in non-secure products or during the development phases of a secure product. All HAB functions are executed as for a closed device. The security hardware is initialized (except for the SNVS which is left in the Non-Secure state), the DCD is processed if present, and the program image is authenticated by the HAB before its execution. All detected errors are logged, but have no influence on the boot flow which continues as if the errors did not occur. This configuration is useful for a secure product development because the program

image runs even if the authentication data is missing or incorrect, and the error log can be examined to determine the cause of the authentication failure.
- Field Return: This level is intended for the parts returned from the shipped products.

NOTE

The DIR_BT_DIS eFuse must be programmed prior to shipping a device in a security enabled configuration. If the this eFuse is not blown, the system is not secure.

6.1.3 Device configuration

This section describes the external inputs that control the behavior of the Boot ROM code.

This includes the boot device selection (SD, MMC, and so on), boot device configuration (SD bus width, speed, and so on), and other. In general, the source for this configuration comes from the eFUSEs embedded inside the chip. However, certain configuration parameters can be sourced from the GPIO pins, allowing further flexibility during the development process.

6.1.3.1 Boot eFUSE descriptions

This table is a comprehensive list of the configuration parameters that the ROM uses.

Table 6-2. Boot eFUSE descriptions

Fuse Address	Configuration	Definition	GPIO[1]	Shipped value	Settings[2]
DIR_BT_DIS 0x470[27]	OEM	Disables the NXP reserved modes.	NA	0	0—The reserved NXP modes are enabled. 1—The reserved NXP modes are disabled. This fuse must be blown to 1 for normal operation.
BT_FUSE_SEL 0x470[28]	OEM	In the Internal Boot mode BOOT_MODE[1:0] = 10, the BT_FUSE_SEL fuse determines whether the boot settings indicated by a Yes in the	NA	0	If BOOT_MODE[1:0] = 0b10: 0—The bits of the SBMR are overridden by the GPIO pins.

Table continues on the next page...

Table 6-2. Boot eFUSE descriptions (continued)

		GPIO column are controlled by the GPIO pins or the eFUSE settings in the On-Chip OTP Controller (OCOTP). In the Boot From Fuse mode BOOT_MODE[1:0] = 00, the BT_FUSE_SEL fuse indicates whether the bit configuration eFuses are programmed.			1—The specific bits of the SBMR are controlled by the eFUSE settings. If BOOT_MODE[1:0] = 0b00 0—The BOOT configuration eFuses are not programmed yet. The boot flow jumps to the serial downloader. 1—The BOOT configuration eFuses are programmed. The regular boot flow is performed.
SEC_CONFIG[1:0] 0 - 0x450[17] 1 - 0x470[25]	SEC_CONFIG [0] - NXP SEC_CONFIG [1] - OEM	Security Configuration, as defined in Boot security settings	NA	01	00—Reserved 01—Open (allows any program image, even if the authentication fails) 1x—Closed (The program image executes only if authenticated)
FIELD_RETURN	OEM	Enables the NXP reserved modes.	NA	0	0—The NXP reserved modes are enabled/ disabled based on the DIR_BT_DIS value. 1—The NXP reserved modes are enabled.
SRK_HASH[255:0] 0x580 - 0x5F0	OEM	256-bit hash value of the super root key (SRK_HASH)	NA	0	Settings vary—used by HAB
BT_DCACHE_DISABLE	OEM	The D Cache disable bit used by the boot ROM for fast HAB processing.	No	0	0— D Cache is enabled by the ROM during the boot. 1— D Cache is disabled by the ROM during the boot.

Table continues on the next page...

Table 6-2. Boot eFUSE descriptions (continued)

L1 I-Cache DISABLE 0x480[12]	OEM	L1 I Cache disable bit used by the boot during the entire execution.	No	0	0—L1 I Cache is enabled by the ROM during the boot. 1—L1 I Cache is disabled by the ROM during the boot.
BT_FREQ 0x480[9]	OEM	Boot frequency selection	No	0	0—Arm—792 MHz, DDR—396 MHz, AXI— 396 MHz 1—Arm—396 MHz, DDR—307 MHz, AXI— 307 MHz. The Arm/DDR/AXI frequency may be lower if the LP_BOOT is blown.
LPB_BOOT 0x480[15:14]	OEM	USB Low-Power Boot	No	00	0x—LPB Disable 10—Divide by 2 11—Divide by 4
BT_LPB_POLARITY 0x480[13]	OEM	USB Low-Power Boot GPIO polarity	No	0	0—Low on the GPIO pad indicates the lowpower condition. 1—High on the GPIO pad indicates the low-power condition.
WDOG_ENABLE 0x480[10]	OEM	Watchdog reset counter enable	No	0	0—The watchdog reset counter is disabled during the serial downloader. 1—The watchdog reset counter is enabled during the serial downloader.
MMC_DLL_DLY[6:0] 0x490[14:8]	OEM	uSDHC Delay Line settings	No	0000000	uSDHC Delay Line settings
SRK_REVOKE[2:0] 0x670[2:0]	OEM	SRK revocation mask	No	000	SRK revocation mask
DISABLE_SDMMC_MFG 0x490[23]	OEM	Disable the SDMMC manufacture mode	Yes	0	0—enable the SD/MMC MFG mode 1—disable the SD/MMC MFG mode

Table continues on the next page...

Table 6-2. Boot eFUSE descriptions (continued)

USDHC_PAD_SETTINGS 0x490[31:24] NAND_PAD_SETTIN GS 0x4A0[31:24]	OEM	Override values for the SD/MMC and NAND boot modes	No	00000000	Override the following IO PAD settings: [1:0] Driver Strength [2] Slew Rate [3] Hysteresis [4] Pull/Keeper select [6:5] Pull up/down config [7] Reserved.
USDHC_OVERRIDE_PAD_SETTINGS 0x490[17]	OEM	Overrides the HYS bit for the SD pads	No	0	Override the IO PAD setting HYS to 1 for the SD pads.
NAND_OVERRIDE_PAD_SETTINGS 0x4B0[7]	OEM	Overrides the HYS bit for the SD pads	No	0	Override the IO PAD setting HYS to 1 for the SD pads.
eMMC_4.4_RESET_TO_PRE-IDLE_STATE 0x490[21]	OEM	ROM resets the boot device in the pre-idle state using the eMMC 4.4 feature, CMD0 with the argument value 0xf0f0f0f0.	No	0	Applicable for booting from the eMMC 4.4 spec or greater version devices. The fuse must not be blown for the eMMC 4.3 or lesser spec version devices.

1. This setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO Boot Overrides for the corresponding GPIO pin.
2. 0 = intact fuse and 1= blown fuse

6.1.3.2 GPIO boot overrides

This table provides a list of the GPIO boot overrides:

Table 6-3. GPIO override contact assignments

Package pin	Direction on reset	eFuse
BOOT_MODE1	Input	Boot mode selection
BOOT_MODE0	Input	
SAI1_RXD0	Input	BOOT_CFG[0]
SAI1_RXD1	Input	BOOT_CFG[1]
SAI1_RXD2	Input	BOOT_CFG[2]

Table continues on the next page...

Table 6-3. GPIO override contact assignments (continued)

Package pin	Direction on reset	eFuse
SAI1_RXD3	Input	BOOT_CFG[3]
SAI1_RXD4	Input	BOOT_CFG[4]
SAI1_RXD5	Input	BOOT_CFG[5]
SAI1_RXD6	Input	BOOT_CFG[6]
SAI1_RXD7	Input	BOOT_CFG[7]
SAI1_TXD0	Input	BOOT_CFG[8]
SAI1_TXD1	Input	BOOT_CFG[9]
SAI1_TXD2	Input	BOOT_CFG[10]
SAI1_TXD3	Input	BOOT_CFG[11]
SAI1_TXD4	Input	BOOT_CFG[12]
SAI1_TXD5	Input	BOOT_CFG[13]
SAI1_TXD6	Input	BOOT_CFG[14]
SAI1_TXD7	Input	BOOT_CFG[15]

The input pins provided are sampled at boot, and can be used to override the corresponding eFUSE values, depending on the setting of the BT_FUSE_SEL fuse.

6.1.3.3 Device Configuration Data (DCD)

The DCD is the configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various on-chip peripherals. See Device Configuration Data (DCD) for more details on DCD.

6.1.4 Device initialization

This section describes the details of the ROM and provides the initialization details.

This includes details on:

- The ROM memory map
- The RAM memory map
- On-chip blocks that the ROM must use or change the POR register default values
- Clock initialization
- Enabling the MMU/L2 cache
- Exception handling and interrupt handling

6.1.4.1 Internal ROM/RAM memory map

These figures show the iROM memory map:

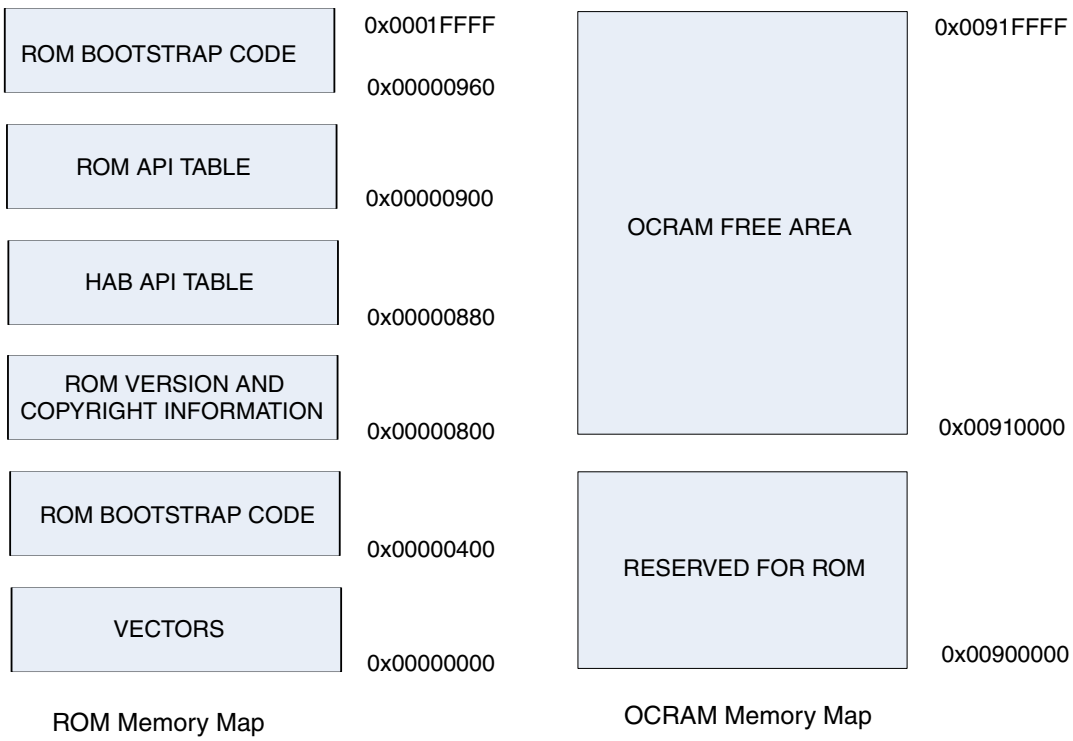


Figure 6-3. Internal ROM and RAM memory map

NOTE

The entire OCRAM region can be used freely after the boot.

6.1.4.2 Boot block activation

The boot ROM affects a number of different hardware blocks which are activated and play a vital role in the boot flow.

The ROM configures and uses the following blocks (listed in an alphabetical order) during the boot process. Note that the blocks actually used depend on the boot mode and the boot device selection:

- APBH—the DMA engine to drive the GPMI module
- BCH—62-bit error correction hardware engine with the AXI bus master and a private connection to the GPMI
- CCM—Clock Control Module
- GPMI—NAND controller pin interface

- OCOTP_CTRL—On-Chip OTP Controller; the OCOTP contains the eFUSEs
- IOMUXC—I/O Multiplexer Control which allows the GPIO use to override the eFUSE boot settings;
- IOMUXC GPR—I/O Multiplexer Control General-Purpose Registers
- CAAM—Cryptographic Acceleration and Assurance Module
- SNVS—Secure Non-Volatile Storage
- SRC—System Reset Controller
- USB—used for the serial download of a boot device provisioning program
- USDHC—Ultra-Secure Digital Host Controller
- WDOG-1—Watchdog timer

6.1.4.3 Clocks at boot time

The table below show the various clocks and their sources used by the ROM.

After the reset, each Arm core has access to all peripherals. The ROM code disables the clocks listed in the following table, except for the boot devices listed in the second column.

Table 6-4. PLL setting by ROM

PLL name	Frequency	Comment
ARM_PLL	1000 MHz	
SYS_PLL1	800 MHz	
SYS_PLL2	1000 MHz	
SYS_PLL3	1000 MHz	

NOTE

All other PLLs are in the default status.

Table 6-5. Clock root setting by ROM

Clock Name	Frequency (MHz)	Source	Enable
ARM_A53_ROOT	1000	arm_pll_clk	Yes
ARM_M4_CLK_ROOT	200	system_pll2_200m_clk	
AHB_CLK_ROOT	133	system_pll1_133m_clk	Yes
MAIN_AXI_CLK_ROOT	333	system_pll2_333m_clk	Yes
VPU_A53_CLK_ROOT	800	arm_pll_clk	No
DRAM_ALT_CLK_ROOT	800	system_pll1_800m_clk	yes
NAND_CLK_ROOT	500	system_pll2_500m_clk	Enabled by driver
NAND_USDHC_BUS_CLK_ROOT	266	system_pll1_266m_clk	Enabled by driver

Table continues on the next page...

Table 6-5. Clock root setting by ROM (continued)

Clock Name	Frequency (MHz)	Source	Enable
USB_BUS_CLK_ROOT		system_pll2_500m_clk	Enabled by driver
NOC_CLK_ROOT	400	system_pll1_800m_clk	Yes
USDHC1_CLK_ROOT	200	system_pll1_400m_clk	Enabled by driver
USDHC2_CLK_ROOT	200	system_pll1_400m_clk	Enabled by driver
USB_PHY_REF_CLK_ROOT		system_pll1_100m_clk	Enabled by driver
ECSPI1_CLK_ROOT	200	system_pll2_200m_clk	No
ECSPI2_CLK_ROOT	200	system_pll2_200m_clk	No
ECSPI3_CLK_ROOT	200	system_pll2_200m_clk	No
WRCLK_CLK_ROOT		system_pll1_40m_clk	No

NOTE

All other clock roots are in the default status.

Table 6-6. NAND_CLK_ROOT setting

NAND data rate	NAND_CLK_ROOT source	Frequency
Async/Legacy NAND	system_pll1_400m_clk	25 MHz
Sync 40M	system_pll1_400m_clk	40 MHz
Toggle/Sync 66M	system_pll1_400m_clk	66 MHz
Toggle 80M	system_pll1_400m_clk	80 MHz
Sync 100M	system_pll1_400m_clk	100 MHz
Toggle/Sync 133M	system_pll1_400m_clk	133 MHz
Sync 160M	system_pll1_400m_clk	133 MHz
Toggle/Sync 200M	system_pll1_400m_clk	200 MHz

NOTE

The NAND_CLK_ROOT source depends on the NAND data rate.

The ROM code disables the clocks listed in the following table, except for the boot devices listed in the "Enabled for boot device" column below.

Table 6-7. CCGR setting by ROM

CCGR Register	LPCG Enable	Enabled for boot device
CCM_CCGR0	Dvfs	
CCM_CCGR1	Anamix	
CCM_CCGR2	Cpu	
CCM_CCGR3	Csu	Security related
CCM_CCGR4	debug	

Table continues on the next page...

Table 6-7. CCGR setting by ROM (continued)

CCGR Register	LPCG Enable	Enabled for boot device
CCM_CCGR5	Dram1	
CCM_CCGR6	reserved	
CCM_CCGR7	Ecspi1	
CCM_CCGR8	Ecspi2	
CCM_CCGR9	Ecspi3	
CCM_CCGR10	Enet1	
CCM_CCGR11	Gpio1	Never gate GPIO clock. uSDHC and test mode use GPIO.
CCM_CCGR12	Gpio2	
CCM_CCGR13	Gpio3	
CCM_CCGR14	Gpio4	
CCM_CCGR15	Gpio5	
CCM_CCGR16	Gpt1	Used by ROM as tick. Keep no changed so it is 25MHz.
CCM_CCGR17	Gpt2	Can be used in DCD. Keep no changed so it is 25MHz.
CCM_CCGR18	Gpt3	
CCM_CCGR19	Gpt4	
CCM_CCGR20	Gpt5	
CCM_CCGR21	Gpt6	
CCM_CCGR22	Hs	
CCM_CCGR23	I2c1	No I2C to be enabled
CCM_CCGR24	I2c2	
CCM_CCGR25	I2c3	
CCM_CCGR26	I2c4	
CCM_CCGR27	Iomux	
CCM_CCGR28	Iomux1	
CCM_CCGR29	Iomux2	
CCM_CCGR30	Iomux3	
CCM_CCGR31	Iomux4	
CCM_CCGR32	M4	
CCM_CCGR33	Mu	
CCM_CCGR34	Ocotp	
CCM_CCGR35	Ocram	
CCM_CCGR36	Ocram_s	
CCM_CCGR37	Pcie	
CCM_CCGR38	Perfmon1	
CCM_CCGR39	Perfmon2	
CCM_CCGR40	Pwm1	
CCM_CCGR41	Pwm2	

Table continues on the next page...

Table 6-7. CCGR setting by ROM (continued)

CCGR Register	LPCG Enable	Enabled for boot device
CCM_CCGR42	Pwm3	
CCM_CCGR43	Pwm4	
CCM_CCGR44	Qos	
CCM_CCGR45	Dismix	
CCM_CCGR46	Megamix	
CCM_CCGR47		
CCM_CCGR48	Rawnand	Will be gated on if boot from NAND is issued.
CCM_CCGR49	Rdc	Never used by ROM. Gate it off.
CCM_CCGR50	Rom	
CCM_CCGR51	Sai1	
CCM_CCGR52	Sai2	
CCM_CCGR53	Sai3	
CCM_CCGR54	Sai4	
CCM_CCGR55	Sai5	
CCM_CCGR56	Sai6	
CCM_CCGR57	Sctr	System counter. Do not gate off.
CCM_CCGR58	Sdma1	Not used by ROM. Gate it off.
CCM_CCGR59	Sdma2	
CCM_CCGR60	Sec_debug	
CCM_CCGR61	Sema1	Not used by ROM. Gate it off.
CCM_CCGR62	Sema2	
CCM_CCGR63	Sim_display	Bus clock. Do not gate off.
CCM_CCGR64	Sim_enet	Bus clock. Do not gate off.
CCM_CCGR65	Sim_m	Bus clock. Do not gate off.
CCM_CCGR66	Sim_main	Bus clock. Do not gate off.
CCM_CCGR67	Sim_s	Bus clock. Do not gate off.
CCM_CCGR68	Sim_wakeup	Bus clock. Do not gate off.
CCM_CCGR69	Sim_usb	Bus clock. Do not gate off.
CCM_CCGR70	Sim_vpu	Bus clock. Do not gate off.
CCM_CCGR71	Snvs	Secure Non-Volatile Storage
CCM_CCGR72	Trace	
CCM_CCGR73	Uart1	
CCM_CCGR74	Uart2	
CCM_CCGR75	Uart3	
CCM_CCGR76	Uart4	
CCM_CCGR77	Usb_ctrl1	Used by ROM USB driver
CCM_CCGR78	Usb_ctrl2	Not used by ROM. Gate it off.
CCM_CCGR79	Usb_phy1	Used by ROM USB driver. Keep it gated on so the PHY can work earlier.

Table continues on the next page...

Table 6-7. CCGR setting by ROM (continued)

CCGR Register	LPCG Enable	Enabled for boot device
CCM_CCGR80	Usb_phy2	Not used by ROM. Keep it on, otherwise the PHY PLL's reference clock (24MHz) will be gated off when PLL is still enabled, and the PHY may not work normally.
CCM_CCGR81	Usdhc1	Used by USDHC driver
CCM_CCGR82	Usdhc2	Used by USDHC driver
CCM_CCGR83	Wdog1	WDOG1 used by ROM.
CCM_CCGR84	Wdog2	
CCM_CCGR85	Wdog3	
CCM_CCGR86	Va53	
CCM_CCGR87	Gpu	
CCM_CCGR88	Hevc	
CCM_CCGR89	Avc	
CCM_CCGR90	Vp9	
CCM_CCGR91	Hevc_inter	
CCM_CCGR92	Gic	Leave on for Software.
CCM_CCGR93	Display	
CCM_CCGR94	Hdmi	
CCM_CCGR95	Hdmi_phy	
CCM_CCGR96	Xtal	
CCM_CCGR97	Pll	
CCM_CCGR98	Tsensor	
CCM_CCGR99	Vpu_dec	
CCM_CCGR100	Pcie2	
CCM_CCGR101	Mipi_csi1	
CCM_CCGR102	Mipi_csi2	

6.1.4.5 Exception handling

The exception vectors located at the start of the ROM are used to map all the Arm exceptions (except the reset exception) to a duplicate exception vector table in the internal RAM.

During the boot phase of CPU0, the RAM vectors point to the serial downloader in the ROM.

After the boot, the program image can overwrite the vectors as required. The code shown below is used to map the ROM exception vector table to the duplicate exception vector table in the RAM.

Mapping ROM Exception Vector Table

```
;; Define linker area for ROM exception vector table
AREA IROM_VECTORS, CODE, READONLY
LDR     PC, Reset_Addr
LDR     PC, Undefined_Addr
LDR     PC, SWI_Addr
LDR     PC, Prefetch_Addr
LDR     PC, Abort_Addr
NOP                          ; Reserved vector
LDR     PC, IRQ_Addr
LDR     PC, FIQ_Addr

;; Define exception vector table
Reset_Addr      DCD     start_address
Undefined_Addr  DCD     iRAM_Undefined_Handler
SWI_Addr        DCD     iRAM_SWI_Handler
Prefetch_Addr   DCD     iRAM_Prefetch_Handler
Abort_Addr      DCD     iRAM_Abort_Handler
                DCD     0               ; Reserved vector
IRQ_Addr        DCD     iRAM_IRQ_Handler
FIQ_Addr        DCD     iRAM_FIQ_Handler

start_address DCD start ;reset handler vector
```

6.1.4.6 Interrupt handling during boot

No special interrupt-handling routines are required during the boot process. The interrupts are disabled during the boot ROM execution and may be enabled in a later boot stage.

6.1.4.7 Persistent bits

Some modes of the boot ROM require the registers that keep their values after a warm reset. The SRC General-Purpose registers are used for this purpose.

See this table for persistent bits list and description:

Table 6-8. Persistent bits

Bit name	Bit location	Description
PERSIST_SECONDARY_BOOT	SRC_GPR10[30]	This bit identifies which image must be used—primary and secondary. Used only for the boot modes that support redundant boot.
PERSIST_BLOCK_REWRITE	SRC_GPR10[29]	This bit is used as a warning. It identifies that there are errors in the NAND blocks that hold the application image. See NAND flash for more details.
PERSISTENT_ENTRY0[31:0]	SRC_GPR1[31:0]	Holds the entry function for the CPU0 to wake up from the low-power mode.
PERSISTENT_ARG0[31:0]	SRC_GPR2[31:0]	Holds the argument of entry function for the CPU0 to wake up from the low-power mode.
PERSISTENT_ENTRY1[31:0]	SRC_GPR3[31:0]	Holds the entry function for the CPU1 to wake up from the low-power mode.
PERSISTENT_ARG1[31:0]	SRC_GPR4[31:0]	Holds the argument of the entry function for the CPU1 to wake up from the low-power mode.

6.1.5 Boot devices (internal boot)

The chip supports these boot flash devices:

- Raw NAND (MLC and SLC), and Toggle-mode NAND flash through GPMI-2 interface, located at CS0. Page sizes of 2 KB, 4 KB, and 8 KB. The bus widths of 8-bit with 2 through 62-bit BCH hardware ECC (Error Correction) are supported.
- SD/MMC/eSD/SDXC/eMMC4.4 via USDHC interface, supporting high capacity cards.

The selection of the external boot device type is controlled by the BOOT_CFG eFUSEs. See this table for more details:

Table 6-9. Boot device selection

BOOT_CFG[14:12]	Boot device
001	SD/eSD
010	MMC/eMMC
011	NAND

6.1.5.1 NAND flash

The boot ROM supports a number of MLC/SLC NAND flash devices from different vendors and LBA NAND flash devices. The Error Correction and Control (ECC) subblock (BCH) is used to detect the errors.

6.1.5.1.1 NAND eFUSE configuration

The boot ROM determines the configuration of the external NAND flash by parameters, either provided by the eFUSE, or sampled on the GPIO pins during boot. See Table 6-10 for parameters details.

NOTE

BOOT_CFGx sampled on the GPIO pins depends on the BT_FUSE_SEL setting. See Boot Fusemap for details.

6.1.5.1.2 NAND flash boot flow and Boot Control Blocks (BCB)

There are two BCB data structures:

- FCB
- DBBT

As a part of the NAND media initialization, the ROM driver uses safe NAND timings to search for the Firmware Configuration Block (FCB) that contains the optimum NAND timings, the page address of the Discovered Bad Block Table (DBBT) Search Area, and the start page address of the primary and secondary firmware.

The hardware ECC level to use is embedded inside the FCB block. The FCB data structure is also protected using the ECC. The driver reads raw 2112 bytes of the first sector and runs through the software ECC engine that determines whether the FCB data is valid or not.

If the FCB is found, the optimum NAND timings are loaded for further reads. If the ECC fails, or the fingerprints do not match, the Block Search state machine increments the page number to the Search Stride number of pages to read for the next BCB until the SearchCount pages have been read.

If the search fails to find a valid FCB, the NAND driver responds with an error and the boot ROM enters the serial download mode.

The FCB contains the page address of the DBBT Search Area, and the page address for primary and secondary boot images. The DBBT is searched in the DBBT Search Area, just like the FCB is searched. After the FCB is read, the DBBT is loaded, and the primary or secondary boot image is loaded using the starting page address from the FCB.

This figure shows the state diagram of the FCB search:

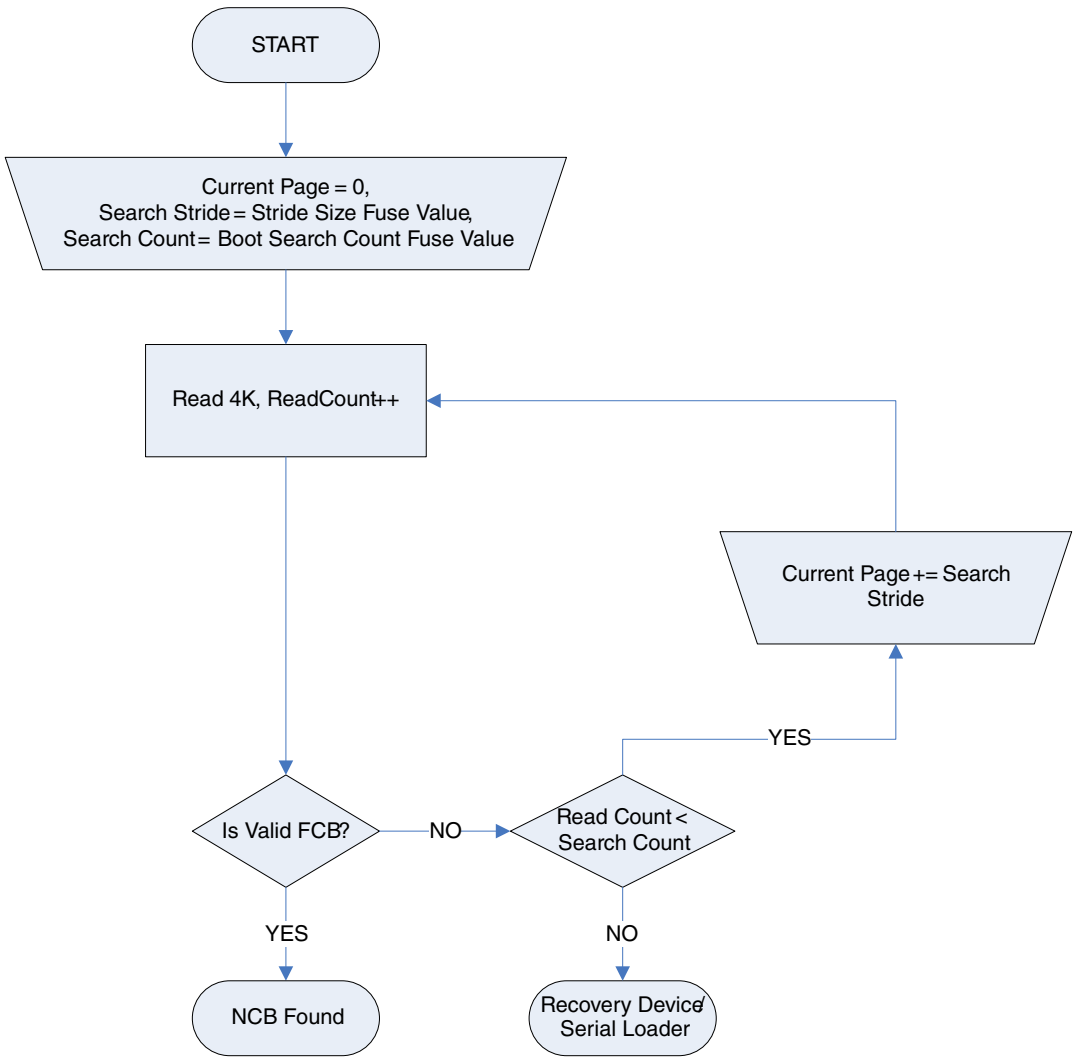


Figure 6-4. FCB search flow

When the FCB is found, the boot ROM searches for the Discovered Bad Blocks Table (DBBT). If the DBBT Search Area is 0 in the FCB, the ROM assumes that there are no bad blocks on the NAND device boot area. See this figure for the DBBT search flow:

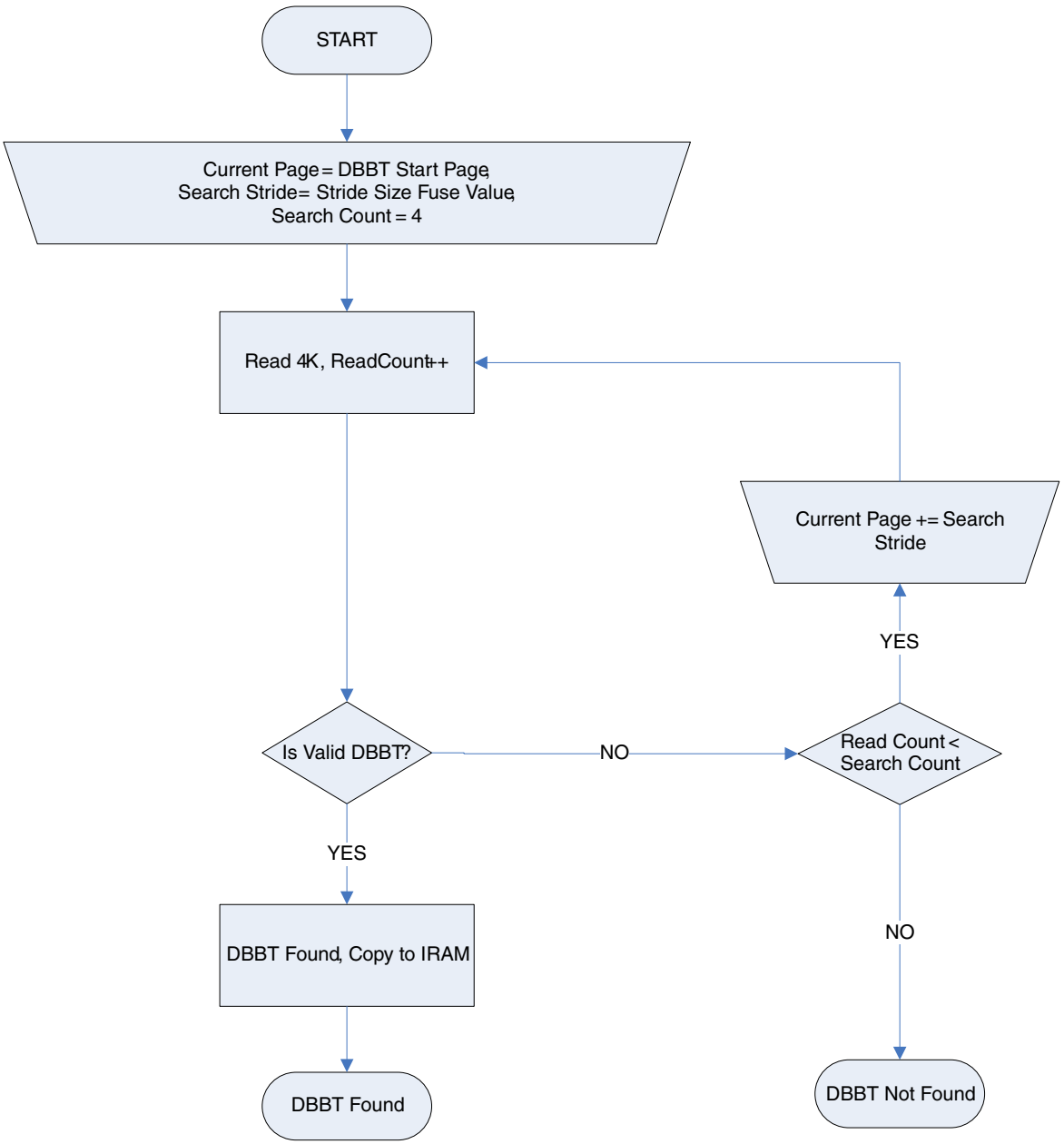


Figure 6-5. DBBT search flow

The BCB search and load function also monitors the ECC correction threshold and sets the PERSIST_BLOCK_REWRITE persistent bit if the threshold exceeds the maximum ECC correction ability.

If there is a page with a number of errors higher than ECC can correct during the primary image read, the boot ROM turns on the PERSIST_SECONDARY_BOOT bit and performs the software reset (After the software reset, the secondary image is used).

If there is a page with number of errors higher than ECC can correct during secondary image read, the boot ROM goes to the serial loader.

6.1.5.1.3 Firmware configuration block

The FCB is the first sector in the first good block. The FCB must be present at each search stride of the search area.

The search area contains copies of the FCB at each stride distance, so, in case the first NAND block becomes corrupted, the ROM finds its copy in the next NAND block. The search area must span over at least two NAND blocks. The location information for the DBBT search area, FW1, and FW2 are all specified in the FCB. This table shows the flash control block structure:

Table 6-10. Flash control block structure

Name	Start byte	Size in bytes	Description
Reserved	0	4	Reserved for Fingerprint #1(Checksum)
FingerPrint	4	4	32-bit word with a value of 0x20424346, in ascii "FCB"
Version	8	4	32-bit version number; this version of FCB is 0x00000001
m_NANDTiming	12	8	8 B of data for eight NAND timing parameters from the NAND datasheet. The eight parameters are: m_NandTiming[0]=data_setup, m_NandTiming[1]=data_hold, m_NandTiming[2]=address_setup, m_NandTiming[3]=dsample_time, m_NandTiming[4]=nand_timing_state, m_NandTiming[5]=REA, m_NandTiming[6]=RLOH, m_NandTiming[7]=RHOH. The ROM only uses the first four parameters, but the FCB provides space for other four parameters to be used by the bootloader or other applications.
PageDataSize	20	4	The number of bytes of data in a page. Typically, this is 2048 bytes for 2112 bytes page size or 4096 bytes for 4314/4224 bytes page size or 8192 for 8568 bytes page size.
TotalPageSize	24	4	The total number of bytes in a page. Typically, 2112 for 2-KB page or 4224 or 4314 for 4-KB page or 8568 for 8-KB page.
SectorsPerBlock	28	4	The number of pages per block. Typically 64 or 128 or depending on the NAND device type.

Table continues on the next page...

Table 6-10. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
NumberOfNANDs	32	4	Not used by ROM
TotalInternalDie	36	4	Not used by ROM
CellType	40	4	Not used by ROM
EccBlockNEccType	44	4	Value from 0 to is used to set the BCH Error Corrrection level 0, 2, 4, .. or 62 for Block BN of ECC page, used in configuring the BCH62 page layout registers.
EccBlock0Size	48	4	Size of block B0 used in configuring the BCH62 page-layout registers.
EccBlockNSize	52	4	Size of block BN used in configuring the BCH62 page-layout registers.
EccBlock0EccType	56	4	Value from 0 to used to set the BCH Error Corrrection level 0, 2, 4, .. or 62 for Block BN of ECC page, used in configuring the BCH62 page layout registers.
MetadataBytes	60	4	Size of metadata bytes used in configuring the BCH62 page-layout registers.
NumEccBlocksPerPage	64	4	Number of the ECC blocks BN not including B0. This value is used in configuring the BCH62 page-layout registers.
EccBlockNEccLevelSDK	68	4	Not used by ROM
EccBlock0SizeSDK	72	4	Not used by ROM
EccBlockNSizeSDK	76	4	Not used by ROM
EccBlock0EccLevelSDK	80	4	Not used by ROM
NumEccBlocksPerPageSDK	84	4	Not used by ROM
MetadataBytesSDK	88	4	Not used by ROM
EraseThreshold	92	4	Not used by ROM
Firmware1_startingPage	104	4	Page number address where the first copy of bootable firmware is located.
Firmware2_startingPage	108	4	Page number address where the second copy of bootable firmware is located.
PagesInFirmware1	112	4	Size of the first copy of firmware in pages.
PagesInFirmware2	116	4	Size of the second copy of firmware in pages.
DBBTSearchAreaStartAddress	120	4	Page address for the bad block table search area.
BadBlockMarkerByte	124	4	This is an input offset in the BCH page for the ROM to swap with the first byte of metadata after reading a page using the BCH62. The ROM supports the restoration of manufacturer-marked bad block markers in the page and this offset is the bad block marker offset location.
BadBlockMarkerStartBit	128	4	This is an input bit offset in the BadBlockMarkerByte for the ROM to use when swapping eight bits with the first byte of metadata.

Table continues on the next page...

Table 6-10. Flash control block structure (continued)

Name	Start byte	Size in bytes	Description
BBMarkerPhysicalOffset	132	4	This is the offset where the manufacturer leaves the bad block marker on a page.
BCHType	136	4	0 for BCH20 and 1 for BCH62. The chip is backwards compatible to BCH20 and this field tells the ROM to use the BCH20 or BCH62 block.
TMTiming2_ReadLatency	140	4	Toggle mode NAND timing parameter read latency, the ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_PreambleDelay	144	4	Toggle mode NAND timing parameter Preamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CEDelay	148	4	Toggle mode NAND timing parameter CE Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_PostambleDelay	152	4	Toggle mode NAND timing parameter Postamble Delay. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_CmdAddPause	156	4	Toggle mode NAND timing parameter Cmd Add Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMTiming2_DataPause	160	4	Toggle mode NAND timing parameter Data Pause. The ROM uses this value to configure the timing2 register of the GPMI.
TMSpeed	164	4	This is the toggle mode speed for the ROM to configure the gpmi clock. 0 for 33 MHz, 1 for 40 MHz, and 2 for 66 MHz.
TMTiming1_BusyTimeout	168	4	Toggle mode NAND timing parameter Busy Timeout. The ROM uses this value to configure the timing1 register of the GPMI.
DISBBM	172	4	If 0, the ROM swaps the BadBlockMarkerByte with metadata[0] after reading a page using the BCH62. If the value is 1, the ROM does not swap.
BBMark_spare_offset	176	4	The offset in the metadata place which stores the data in the bad block marker place.
Onfi_sync_enable	180	4	Enable the Onfi nand sync mode support.
Onfi_sync_speed	184	4	Speed for the Onfi nand sync mode: 0 - 24 MHz, 1 - 33 MHz, 2 - 40 MHz, 3 - 50 MHz, 4 - 66 MHz, 5 - 80 MHz, 6 - 100 MHz, 7 - 133 MHz, 8 - 160 MHz, 9 - 200 MHz
Onfi_syncNANDData	188	28	The parameters for the Onfi nand sync mode timing. They are read latency, ce_delay, preamble_delay, postamble_delay, cmdadd_pause, data_pause, and busy_timeout.
DISBB_Search	216	4	Disable the bad block search function when reading the firmware, only using DBBT.

The FCB data structure is protected using a 62-bit ECC. The layout of the FCB page is illustrated in this figure:

Figure 6-6. Layout of the FCB page

The detailed parameters of the FCB pages are listed in this table:

Table 6-11. Parameters setting for FCB page

Parameter	Value
TotalPageSize	2048+64=2112
MetadataBytes	32
EccBlock0Size	128
EccBlock0EccType	31
BCHType	0
EccBlockNSize	128
EccBlockNEccType	31
NumEccBlocksPerPage	7

To reduce the disturbances caused by a neighboring cell in the FCB page in the NAND chip, a randomizer is enabled when reading the FCB page. BCH ECC has a Randomizer module that is interfaced through the GPMI APBHDMA chain. The Randomizer can generate random data based on BCH ECC encoded/decoded data. It can be employed to reduce the disturbances caused by a neighboring cell in the NAND chip, thus reducing bit errors. The randomizer is used to reduce the bit errors in the FCB. Ensure that the randomizer is enabled when burning the FCB pages in the NAND flash. To control the randomizer for the pages (except for FCB), a new field called Randomizer_Enable is added into the FCB structure. If the Randomizer_Enable field is set to 0, the randomizer is disabled. Reading the pages (except for FCB) being set to a non-zero value enables the randomizer. For detailed randomizer information, see Randomizer.

6.1.5.1.4 Discovered Bad Block Table (DBBT)

See this table for the DBBT format:

Table 6-12. DBBT structure

Name	Start byte	Size in bytes	Description
reserved	0	4	-
FingerPrint	4	4	32-bit word with a value of 0x44424254,in ascii "DBBT"
Version	8	4	32-bit version number; this version of DBBT is 0x00000001
reserved	12	4	-
DBBT_NUM_OF_PAGES	16	4	Size of the DBBT in pages
reserved	20	4*PageSize-20	-
reserved	4*PageSize	4	-
Number of Entries	4*PageSize + 4	4	Number of bad blocks
Bad Block Number	4*PageSize + 8	4	First bad block number
Bad Block Number	4*PageSize + 12	4	Second bad block number
-	-	-	Next bad block number
-	-	-	-
Last bad block number	-	-	Last bad block number

6.1.5.1.5 Bad block handling in ROM

During the firmware boot, at the block boundary, the Bad Block table is searched for a match to the next block.

If no match is found, the next block can be loaded. If a match is found, the block must be skipped and the next block checked.

If the Bad Block table start page is null, check the manufactory made Bad Block marker. The location of the Bad Block maker is at the first three or last three pages in every block of the NAND flash. The NAND manufacturers normally use one byte in the spare area of certain pages within a block to mark that a block is bad or not. A value of 0xFF means good block, non-FF means bad block.

To preserve the BI (bad block information), the flash updater or gang programmer applications must swap the Bad Block Information (BI) data to byte 0 of the metadata area for every page before programming the NAND flash. When the ROM loads the firmware, it copies back the value at metadata[0] to the BI offset in the page data. This figure shows how the factory bad block marker is preserved:

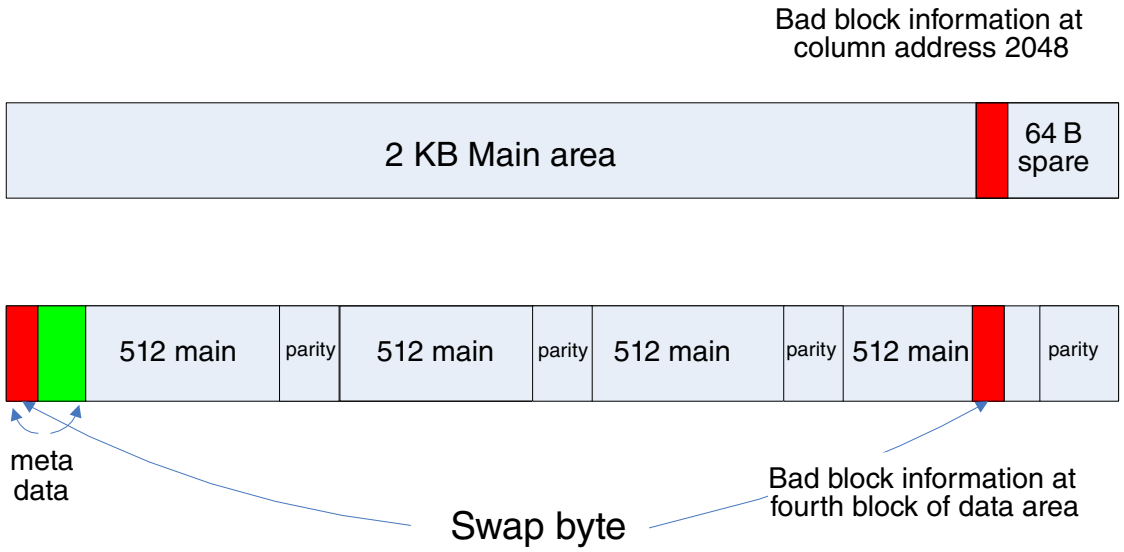


Figure 6-7. Factory bad block marker preservation

In the FCB structure, there are two elements (m_u32BadBlockMarkerByte and m_u32BadBlockMarkerStartBit) to indicate the byte and bit place in the page data that the manufacturer marked the bad block marker.

6.1.5.1.6 Toggle mode DDR NAND boot

If the BT_TOGGLEMODE efuse is blown, the ROM does the following to boot from the Samsung's toggle mode DDR NAND.

6.1.5.1.6.1 GPMI and BCH clocks configuration

The ROM sets the clock source and the dividers in the CCM registers.

If the BOOT_CFG is set (toggle mode), the GPMI/BCH CLK source is PLL2PFD4, and running at 66 MHz, otherwise the GPMI/ BCH CLK souce is PLL3, running at 24 MHz. The ROM sets the default values to timing0, timing1, and timing2 gpmi registers for 24 MHz clock speed. It uses the BOOT_CFG fuse to configure the GPMI timing2 register parameters preamble delay and read latency. The default value for these parameters is 2 when the fuses are not blown.

The default timing parameter values used by the ROM for the toggle-mode device are:

- Timing0.ADDRESS_SETUP = 5
- Timing0.DATA_SETUP = 10
- Timing0.DATA_HOLD = 10
- Timing1.DEVICE_BUSY_TIMEOUT = 0 x 500

- Timing2.READ_LATENCY = BOOT_CFG if blown, otherwise 2
- Timing2.CE_DELAY = 2
- Timing2.PREAMBLE_DELAY = BOOT_CFG if blown, otherwise 2
- Timing2.POSTAMBLE_DELAY = 3
- Timing2.CMDADD_PAUSE = 4
- Timing2.DATA_PAUSE = 6

The default timing parameters can be overriden by the TMTiming2_ReadLatency, TMTiming2_PreambleDelay, TMTiming2_CEDelay, TMTiming2_PostambleDelay, TMTiming2_CmdAddPause, and TMTiming2_DataPause parameters of the FCB.

6.1.5.1.6.2 Setup DMA for DDR transfers

In the DMA descriptors, the GPMI is configured to read the page data at a double data rate, the word length is set to 16, and the transfer count to a half of the page size.

6.1.5.1.6.3 Reconfigure timing and speed using values in FCB

After reading the FCB page with the GPMI set to default timings and a speed of 33 MHz, the ROM reconfigures the CCM dividers to run the gpmi/bch clks to a desired speed specified in the FCB for the rest of the boot process. The GPMI timing registers are also reconfigured to the values specified in the FCB.

The GPMI speed can be configured using the FCB parameter TMSpeed:

- 0—25 MHz
- 1—33 MHz
- 2—40 MHz
- 3—50 MHz
- 4—66 MHz
- 5—80 MHz
- 6—100 MHz
- 7—133 MHz
- 8—133 MHz
- 9—200 MHz

The GPMI timing0 register fields data_setup, data_hold, and address_setup are set to the values specified for the data_setup and data_hold and address_setup in the FCB member m_NANDTiming.

The GPMI timing1.DEVICE_BUSY_TIMEOUT is set to the value specified in the FCB member TMTiming1_BusyTimeout.

The GPMI timing2 register values are set using the FCB members TMTiming2.READ_LATENCY, CE_DELAY, PREAMBLE_DELAY, POSTAMBLE_DELAY, CMDADD_PAUSE, and DATA_PAUSE.

6.1.5.1.7 Typical NAND page organization

6.1.5.1.7.1 BCH ECC page organization

The first data block is called block 0 and the rest of the blocks are called block N. A separate ECC level scan is used for block 0 and block N.

The metadata bytes must be located at the beginning of a page, starting at byte 0, followed by the data block 0, the ECC bytes for data block 0, the block 1 and its ECC bytes, and so on, up until the N data blocks. The ECC level for the block 0 can be different from the ECC level for the rest of the blocks.

For the NAND boot with page-size restrictions and the data block size restricted to 512 B, only few combinations of the ECC for block 0 and block N are possible.

This figure shows the valid layout for 2112-byte sized page.

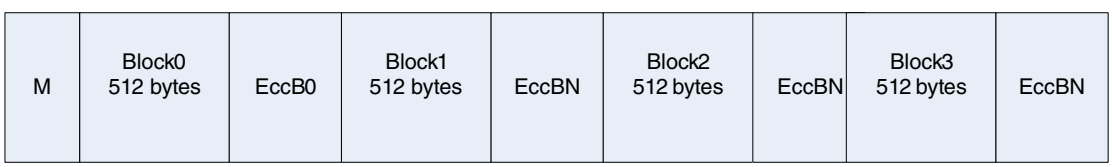


Figure 6-8. Valid layout for 2112-byte sized page

The example below is for 13 bits of parity (GF13). The number of ECC bits required for a data block is calculated using the (ECC_Correction_Level * 13) bits.

In the above layout, the ECC size for EccB0 and EccBN must be selected to not exceed a total page size of 2112 bytes. The EccB0 and EccBN can be one of the 2, 4, 6, 8, 10, 12, 14, 16, 18, and 20 bits on the ECC correction level. The total bytes are:

[M + (data_block_size x 4) + ([EccB0 + (EccBN x 3)] x 13) / 8] <= 2112;

M = metadata bytes and data_block_size is 512.

There are four data blocks of 512 bytes each in a page of 2-KB page sized NAND. The values of EccB0 and EccBN must be such that the above calculation does not result in a value greater than 2112 bytes.

M	Block0 512 bytes	EccB0	Block1 512 bytes	EccBN	Block2 512 bytes	EccBN	Block3 512 bytes	EccBN
	Block4 512 bytes	EccBN	Block5 512 bytes	EccBN	Block6 512 bytes	EccBN	Block7 512 bytes	EccBN

Figure 6-9. Valid layout for 4-KB sized page

Different NAND manufacturers have different sizes for a 4-KB page; 4314 bytes is typical.

[M + (data_block_size x 8) + ([EccB0 + (EccBN x 7)] x 13) / 8] <= 4314;

M= metadata bytes and data_block_size is 512.

There are eight data blocks of 512 bytes each in a page of a 4-KB page sized NAND. The values of the EccB0 and EccBN must be such that the above calculation does not result in a value greater than the size of a page in a 4-KB page NAND.

6.1.5.1.7.2 Metadata

The number of bytes used for the metadata is specified in the FCB. The metadata for the BCH encoded pages is placed at the beginning of a page. The ROM only cares about the first byte of metadata to swap it with a bad block marker byte in the page data after each page read; it is important to have at least one byte for the metadata bytes field in the FCB data structure.

6.1.5.1.8 IOMUX configuration for NAND

The following table shows the RawNAND IOMUX pin configuration.

Table 6-13. NAND IOMUX pin configuration

Signal	Pad name
NAND_CLE	SD3_CLK.alt1
NAND_ALE	SD3_CMD.alt1
NAND_WP_B	SAI1_MCLK.alt1
NAND_RE_B	SD3_STROBE.alt1
NAND_WE_B	SD3_RESET_B.alt1
NAND_READY_B	SAI1_TXD.alt1
NAND_DQS	SAI1_TXFS.alt1

Table continues on the next page...

Table 6-13. NAND IOMUX pin configuration (continued)

NAND_CE0_B	SAI1_TXC.alt1
NAND_DATA00	SD3_DATA0.alt1
NAND_DATA01	SD3_DATA1.alt1
NAND_DATA02	SD3_DATA2.alt2
NAND_DATA03	SD3_DATA3.alt3
NAND_DATA04	SD3_DATA4.alt4
NAND_DATA05	SD3_DATA5.alt5
NAND_DATA06	SD3_DATA6.alt6
NAND_DATA07	SD3_DATA7.alt7

6.1.5.2 Expansion device

The ROM supports booting from the MMC/eMMC and SD/eSD compliant devices.

6.1.5.2.1 Expansion device eFUSE configuration

The SD/MMC/eSD/eMMC/SDXC boot can be performed using either the USDHC ports, based on the setting of the BOOT_CFG[11:10] (Port Select) fuse or it is associated to the GPIO input value at the boot.

All USDHC ports support the fast boot. See this table for details:

Table 6-14. USDHC boot eFUSE descriptions

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
BOOT_CFG[7]	OEM	Fast boot support	Yes	000	0 - Normal boot 1 - Fast boot
BOOT_CFG[6:4]	OEM	Bus width	Yes	000	0 - SD/eSD/SDXC 1 - MMC/eMMC
BOOT_CFG[3:1]	OEM	SD speed mode	Yes	000	SD speed selection speed 000 - Normal/SDR12 001 - High/SDR25 010 - SDR50 011 - SDR104 else - Reserved
		MMC Speed Mode (CFG[3:2])	Yes	00	MMC speed selection 00 - Normal 01 - High

Table continues on the next page...

Table 6-14. USDHC boot eFUSE descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
					else - Reserved
		USDHC1 IO Voltage Selection (CFG[1])	Yes	0	USDHC1 IO VOLTAGE SELECTION (only for MMC/eMMC boot) 0 - 3.3 V 1 - 1.8 V
BOOT_CFG[0]	OEM	USDHC2 IO VOLTAGE	Yes	0	USDHC2 IO VOLTAGE SELECTION (only for MMC/eMMC boot) 0 - 3.3 V 1 - 1.8 V
BOOT_CFG[15:12]	OEM	Boot device selection	Yes	0000	0001 - Boot from SD/eSD 0010 - Boot from MMC/eMMC
BOOT_CFG[11:10]	OEM	USDHC port selection	Yes	00	00 - USDHC-1 01 - USDHC-2 10 - USDHC-3 else - reserved
BOOT_CFG[9]	OEM	SD power cycle enable/ eMMC reset enable	Yes	0	SD power cycle/eMMC reset 0 - Disabled 1 - Enabled
BOOT_CFG[8]	OEM	USDHC loopback clock selection	Yes	0	USDHC loopback clock source selection 0 - Through SD pad 1 - Direct
0x490[14:8]	OEM	SD/MMC DLL DLY config	No	0	Delay target for USDHC DLL, it is applied to the slave mode target delay or overrides the mode target delay, depending on the DLL override fuse bit value.
0x490[15]	OEM	USDHC DLL override enabled	No	0	0 - No override 1 - Override
0x490[16]	OEM	USDHC DLL enabled	No	0	0 - Disable the DLL for SD/eMMC 1 - Enable the DLL for SD/eMMC
0x490[17]	OEM	USDHC override pad settings selection			0 - Use default pad settings 1 - Override the USDHC pad settings by using the PAD_SETTINGS value
0x490[18]	OEM	USDHC_IOMUX_SION_BIT_ENABLE	No	0	0 - Disable 1 - Enable

Table continues on the next page...

Table 6-14. USDHC boot eFUSE descriptions (continued)

Fuse	Config	Definition	GPIO[1]	Shipped value	Settings
0x490[19]	OEM	ENABLE_EMMC_5K_PULLUP	No	0	0 - 47 K pullup 1 - 5 K pullup
0x490[20]	OEM	USDHC_PAD_PULL_DOWN	No	0	0 - No action 1 - Pull down
0x490[21]	OEM	Issue pre-idle command enabled (for eMMC4.4)	No	0	0 - Enable 1 - Disable
0x490[23]	OEM	Disable SDMMC manufacture mode	No	0	0 - Enable 1 - Disable
0X490[31:24]	OEM	USDHC pad setting override	No	0	Override pad settings default if 0X490[17] is set
0x4A0[0]	OEM	Fast boot acknowledge enable	No	0	0 - Boot Ack disabled 1 - Boot Ack enabled
0x4A0[1]	OEM	USDHC3 IO voltage selection	No	0	0 - 3.3 V 1 - 1.8 V
0x4A0[2]	OEM	uSDHC power-off polarity selection	No	0	0 - Low 1 - High
0x4A0[3]	OEM	uSDHC power cycle delay selection	No	0	0 - 5 ms 1 - 2.5 ms
0x4A0[5:4]	OEM	uSDHC power cycle interval	No	0	00 - 20 ms 01 - 10 ms 10 - 5 ms 11 - 2.5 ms

1. The setting can be overridden by the GPIO settings when the BT_FUSE_SEL fuse is intact. See GPIO boot overrides for the corresponding GPIO pin.

The boot code supports these standards:

- MMCv4.4 or less
- eMMCv4.4 or less
- SDv2.0 or less
- eSDv2.10 rev-0.9, with or without FAST_BOOT
- SDXCv3.0

The MMC/SD/eSD/SDXC/eMMC can be connected to any of the USDHC blocks and can be booted by copying 4 KB of data from the MMC/SD/eSD/eMMC device to the internal RAM. After checking the Image Vector Table header value (0xD1) from program image, the ROM code performs a DCD check. After a successful DCD

extraction, the ROM code extracts from the Boot Data Structure the destination pointer and length of image to be copied to the RAM device from where the code execution occurs.

The maximum image size to load into the SD/MMC boot is 32 MB. This is due to a limited number of uSDHC ADMA Buffer Descriptors allocated by the ROM.

NOTE

The initial 4 KB of the program image must contain the IVT, DCD, and the Boot Data structures.

Table 6-15. SD/MMC frequencies

	SD	MMC	MMC (DDR mode)
Identification (KHz)	347.22		
Normal-speed mode (MHz)	25	20	25
High-speed mode (MHz)	50	40	50
UHSI SDR50 (MHz)	100		
UHSI SDR104 (MHz)	200		

NOTE

The boot ROM code reads the application image length and the application destination pointer from the image.

6.1.5.2.2 MMC and eMMC boot

This table provides the MMC and eMMC boot details.

Table 6-16. MMC and eMMC boot details

Normal boot mode	During the initialization (normal boot mode), the MMC frequency is set to 347.22 KHz. When the MMC card enters the identification portion of the initialization, the voltage validation is performed, and the ROM boot code checks the high-voltage settings and the card capacity. The ROM boot code supports both the high-capacity and low-capacity MMC/eMMC cards. After the initialization phase is complete, the ROM boot code switches to a higher frequency (20 MHz in the normal boot mode or 40 MHz in the high-speed mode). The eMMC is also interfaced via the USDHC and follows the same flow as the MMC. The boot partition can be selected for an MMC4.x card after the card initialization is complete. The ROM code reads the BOOT_PARTITION_ENABLE field in the Ext_CSD[179] to get the boot partition to be set. If there is no boot partition mentioned in the BOOT_PARTITION_ENABLE field or the user partition was mentioned, the ROM boots from the user partition.

Table continues on the next page...

Table 6-16. MMC and eMMC boot details (continued)

eMMC4.3 or eMMC4.4 device supporting special boot mode	If using an eMMC4.3 or eMMC4.4 device that supports the special boot mode, it can be initiated by pulling the CMD line low. If the BOOT ACK is enabled, the eMMC4.3/eMMC4.4 device sends the BOOT ACK via the DATA lines and the ROM can read the BOOT ACK [S010E] to identify the eMMC4.3/eMMC4.4 device. If the BOOT ACK is enabled, the ROM waits 50 ms to get the BOOT ACK and if the BOOT ACK is received by the ROM. If BOOT ACK is disabled ROM waits 1 second for data. If the BOOT ACK or data was received, the eMMC4.3/eMMC4.4 is booted in the "boot mode", otherwise the eMMC4.3/eMMC4.4 boots as a normal MMC card from the selected boot partition. This boot mode can be selected by the BOOT_CFG (fast boot) fuse. The BOOT ACK is selected by the .
eMMC4.4 device	If using the eMMC4.4 device, the Double Data Rate (DDR) mode can be used. This mode can be selected by the BOOT_CFG2[7:5] (bus width) fuse.

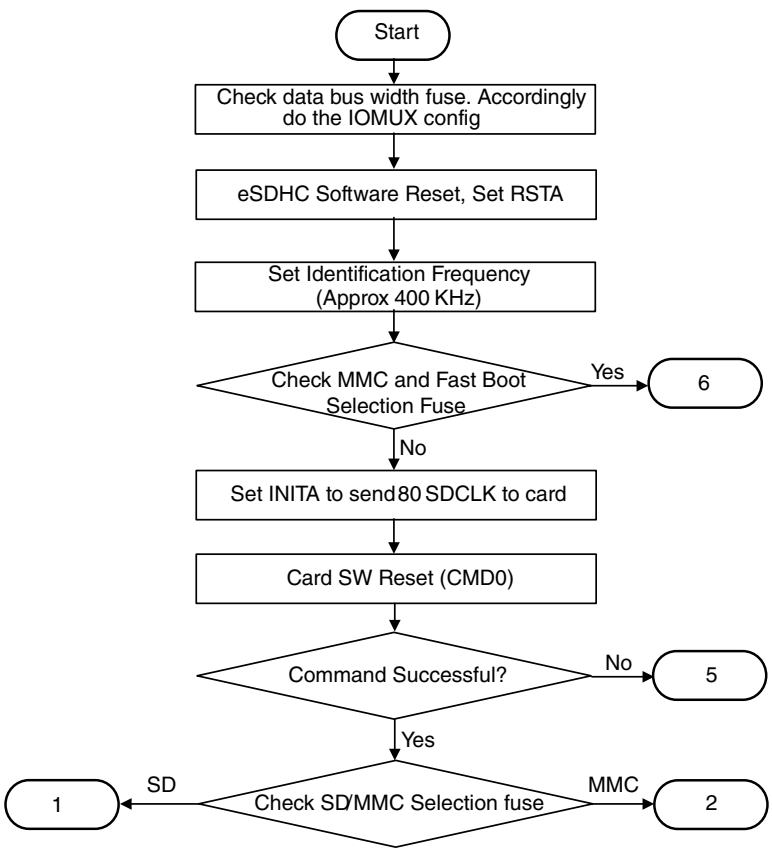


Figure 6-10. Expansion device boot flow (1 of 6)

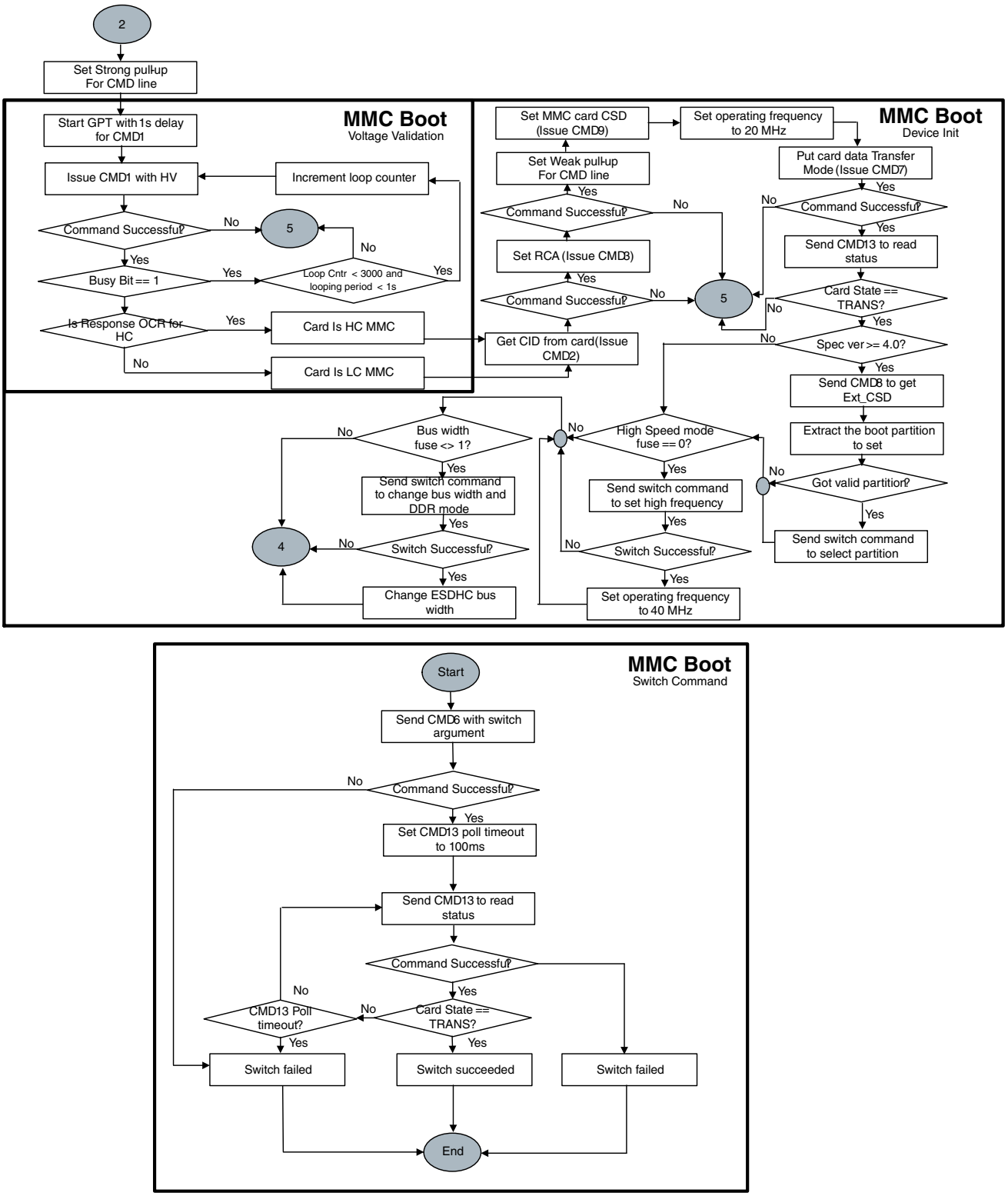


Figure 6-11. Expansion device (MMC) boot flow (2 of 6)

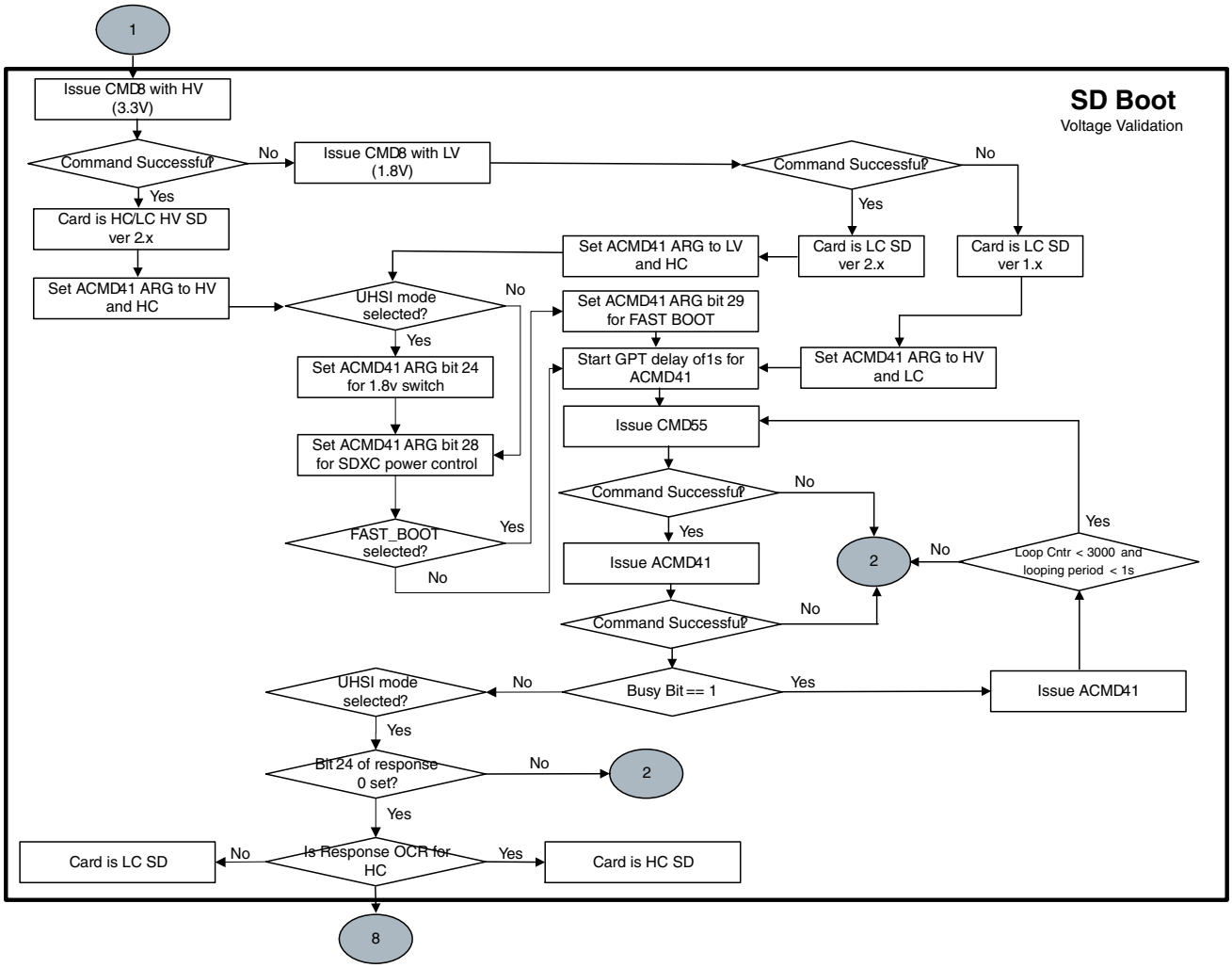


Figure 6-12. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 1

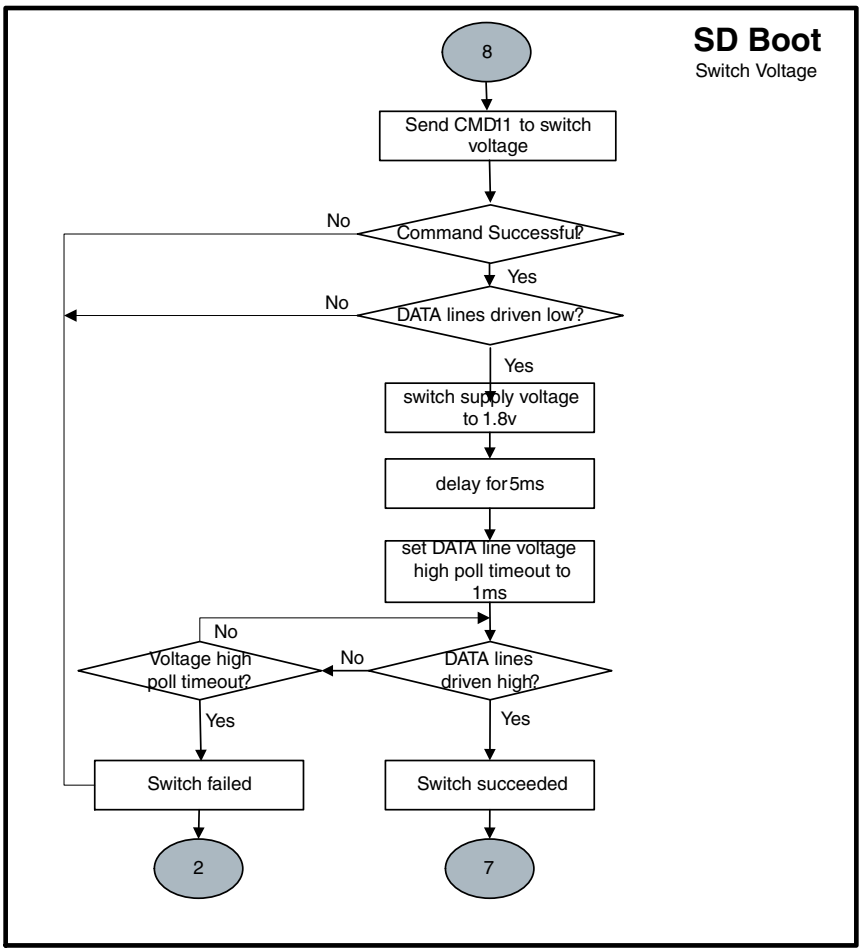


Figure 6-13. Expansion device (SD/eSD/SDXC) boot flow (3 of 6) part 2

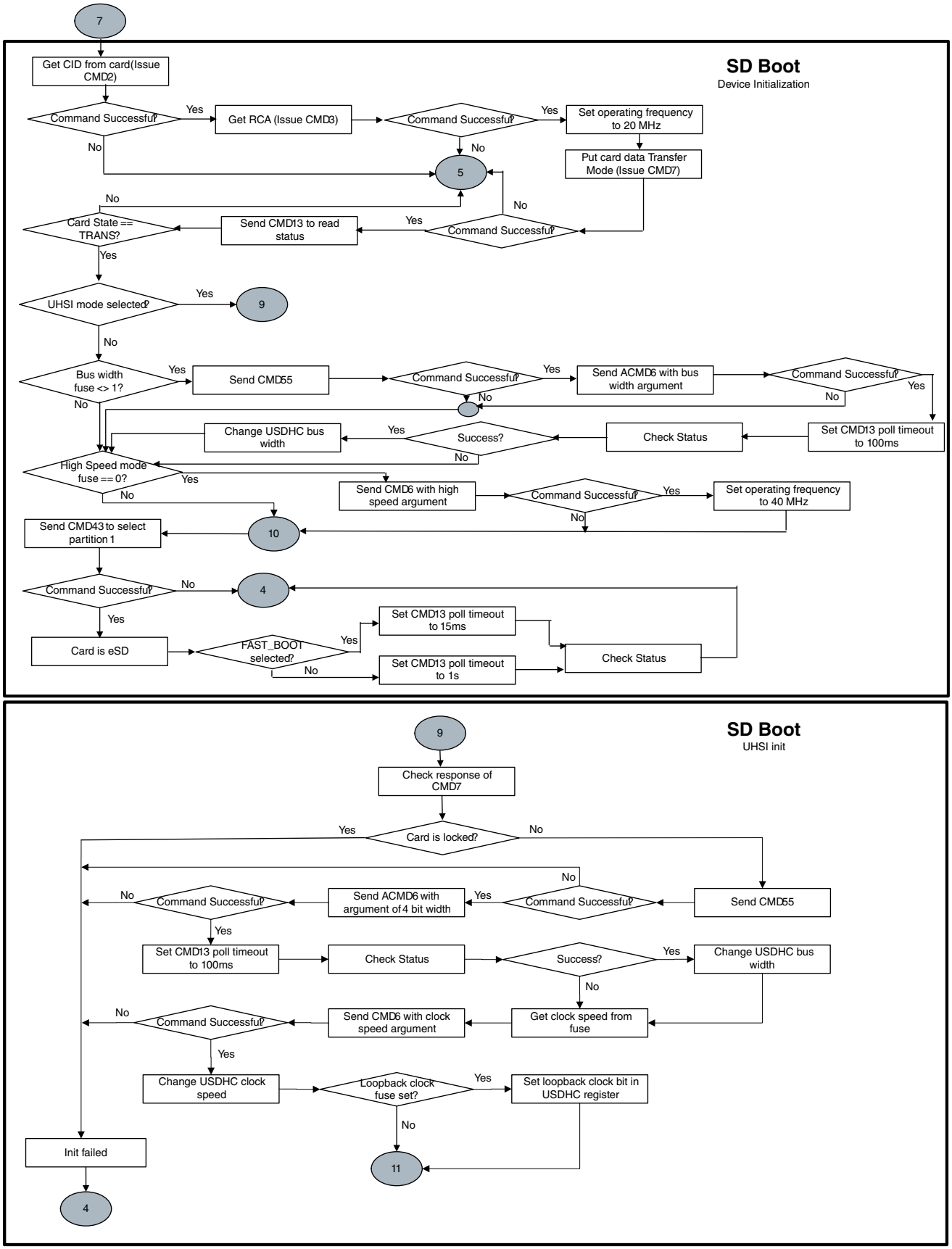


Figure 6-14. Expansion device (MMCSD/eSD/SDXC) boot flow (4 of 6)

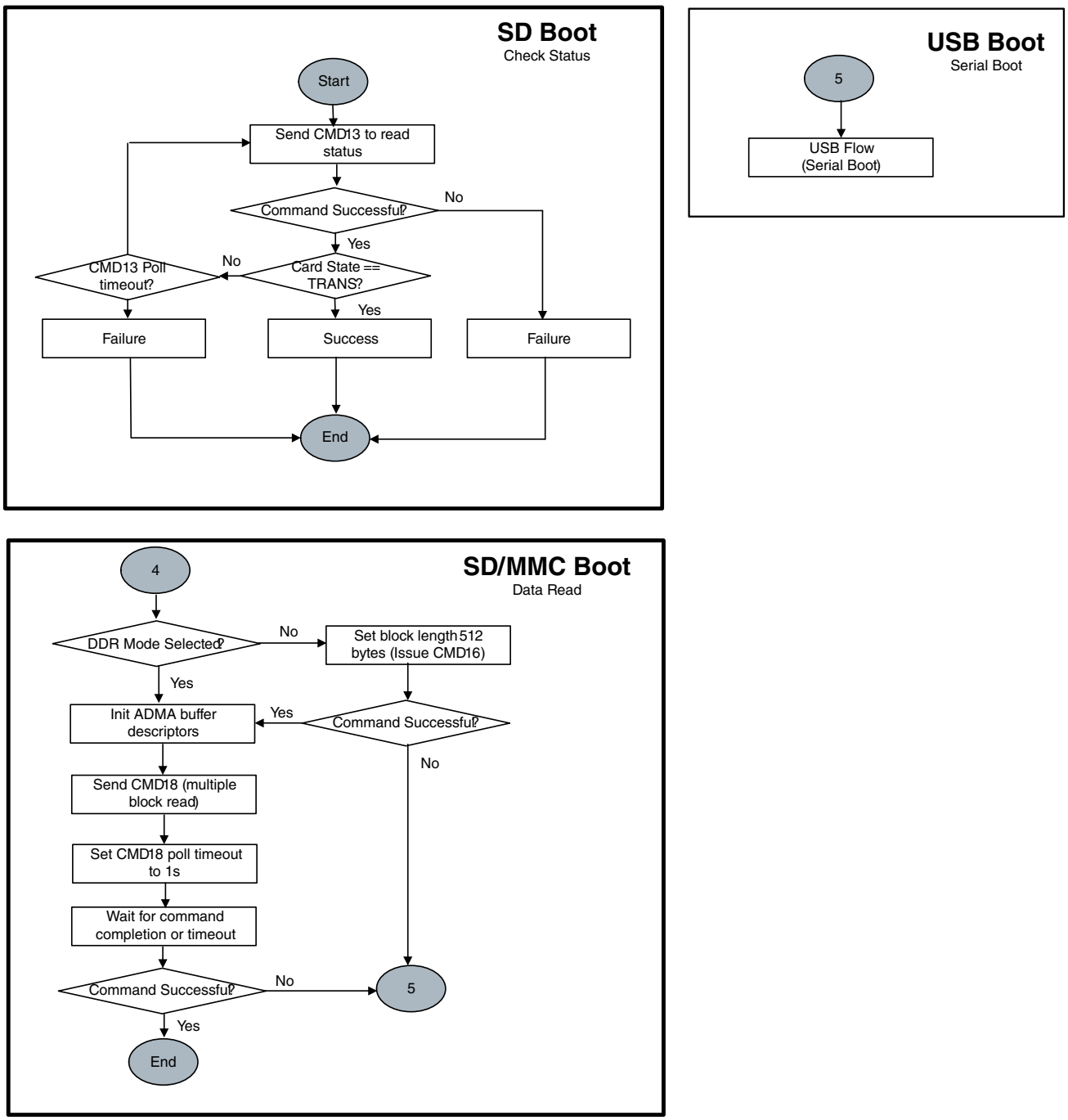


Figure 6-15. Expansion device (SD/eSD) boot flow (5 of 6)

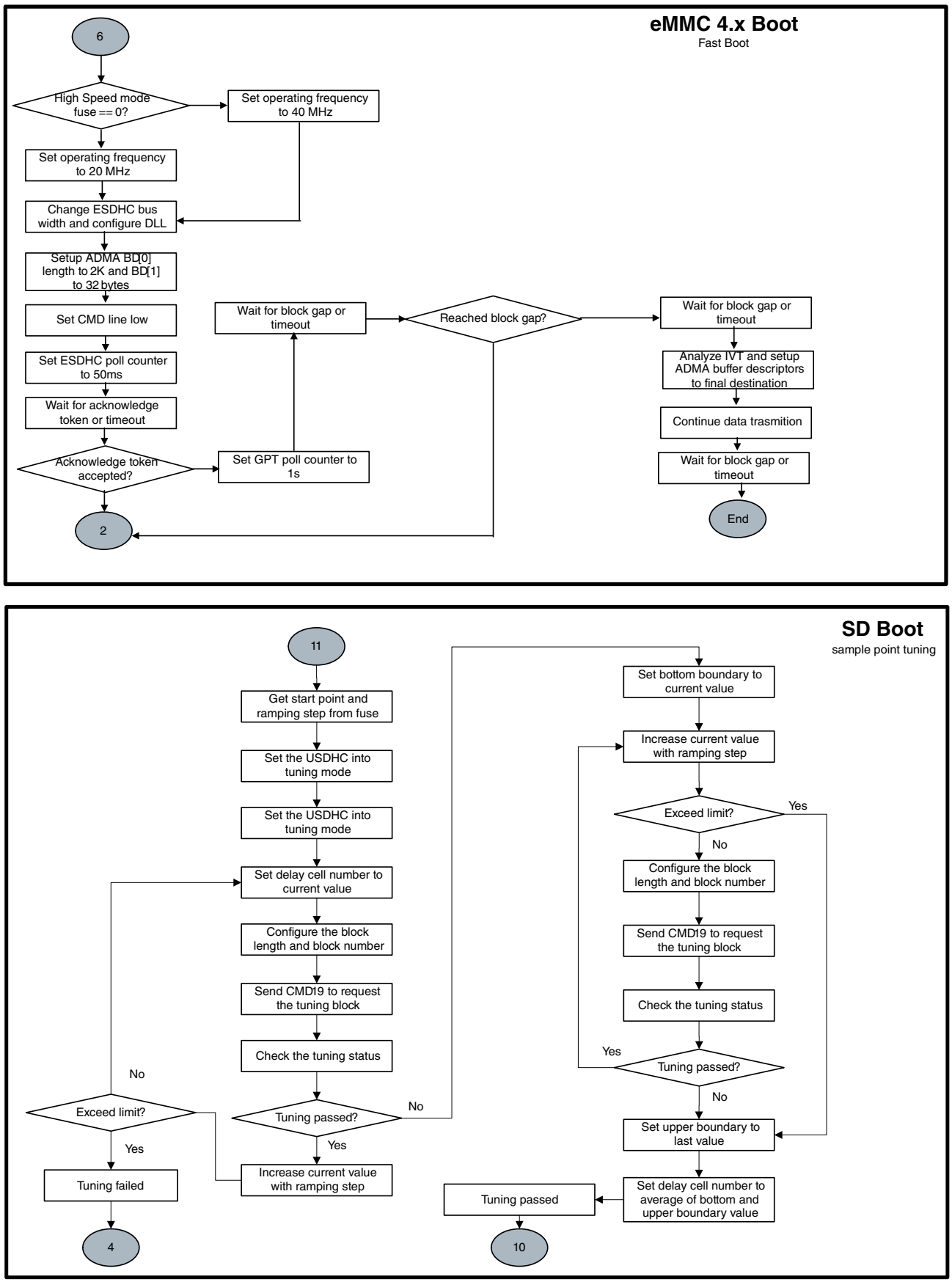


Figure 6-16. Expansion device boot flow (6 of 6)

6.1.5.2.3 SD, eSD, and SDXC

After the normal boot mode initialization begins, the SD/eSD/SDXC frequency is set to 347.22 kHz. During the identification phase, the SD/eSD/SDXC card voltage validation is performed. During the voltage validation, the boot code first checks with the high-voltage settings; if that fails, it checks with the low-voltage settings.

The capacity of the card is also checked. The boot code supports the high-capacity and low-capacity SD/eSD/SDXC cards after the voltage validation card initialization is done.

During the card initialization, the ROM boot code attempts to set the boot partition for all SD, eSD, and SDXC devices. If this fails, the boot code assumes that the card is a normal SD or SDXC card. If it does not fail, the boot code assumes it is an eSD card. After the initialization phase is over, the boot code switches to a higher frequency (25 MHz in the normal-speed mode or 50 MHz in the high-speed mode). The ROM also supports the FAST_BOOT mode booting from the eSD card. This mode can be selected by the BOOT_CFG[4] (Fast Boot) fuse described in .

For the UHSI cards, the clock speed fuses can be set to SDR50 or SDR104 on ports. This enables the voltage switch process to set the signaling voltage to 1.8 V during the voltage validation. The bus width is fixed at a 4-bit width and a sampling point tuning process is needed to calibrate the number of the delay cells. If the SD Loopback Clock eFuse is set, the feedback clock comes directly from the loopback SD clock, instead of the card clock (by default). The SD clock speed can be selected by the BOOT_CFG[3:2], and the SD Loopback Clock is selected by the BOOT_CFG[0].

The UHSI calibration start value (MMC_DLL_DLY[6:0]) and the step value can be set to optimize the sample point tuning process.

If the SD Power Cycle Enable eFuse is 1, the ROM sets the SD_RST pad low, waits for 5 ms, and then sets the SD_RST pad high. If the SD_RST pad is connected to the SD power supply enable logic on board, it enables the power cycle of the SD card. This may be crucial in case the SD logic is in the 1.8 V states and must be reset to the 3.3 V states.

6.1.5.2.4 IOMUX configuration for SD/MMC

Table 6-17. SD/MMC IOMUX pin configuration

Signal	USDHC1	USDHC2
CLK	SD1_CLK.alt0	SD2_CLK.alt0
CMD	SD1_CMD.alt0	SD2_CMD.alt0
DATA0	SD1_DATA0.alt0	SD2_DATA0.alt0
DATA1	SD1_DATA1.alt0	SD2_DATA1.alt0
DATA2	SD1_DATA2.alt0	SD2_DATA2.alt0

Table continues on the next page...

Table 6-17. SD/MMC IOMUX pin configuration (continued)

Signal	USDHC1	USDHC2
DATA3	SD1_DATA3.alt0	SD2_DATA3.alt0
DATA4	SD1_DATA4.alt0	-
DATA5	SD1_DATA5.alt0	-
DATA6	SD1_DATA6.alt0	-
DATA7	SD1_DATA7.alt0	-
VSELECT	-	GPIO1_IO04.alt1
RESET_B	SD1_RESET_B.alt0	SD2_RESET_B.alt0
CD_B	-	SD2_CD_B.alt0
WP	-	SD2_WP.alt0

6.1.5.2.5 Redundant boot support for expansion device

The ROM supports the redundant boot for an expansion device. The primary or secondary image is selected, depending on the PERSIST_SECONDARY_BOOT setting. (see Table 6-8).

If the PERSIST_SECONDARY_BOOT is 0, the boot ROM uses address 0x8400 for the primary image.

If the PERSIST_SECONDARY_BOOT is 1, the boot ROM reads the secondary image table from address 0x8200 on the boot media and uses the address specified in the table for the secondary image.

Table 6-18. Secondary image table format

Reserved (chipNum)
Reserved (driveType)
tag
firstSectorNumber
Reserved (sectorCount)

Where:

- The tag is used as an indication of the valid secondary image table. It must be 0x00112233.
- The firstSectorNumber is the first 512-byte sector number of the secondary image.

For the secondary image support, the primary image must reserve the space for the secondary image table. See this figure for the typical structures layout on an expansion device.

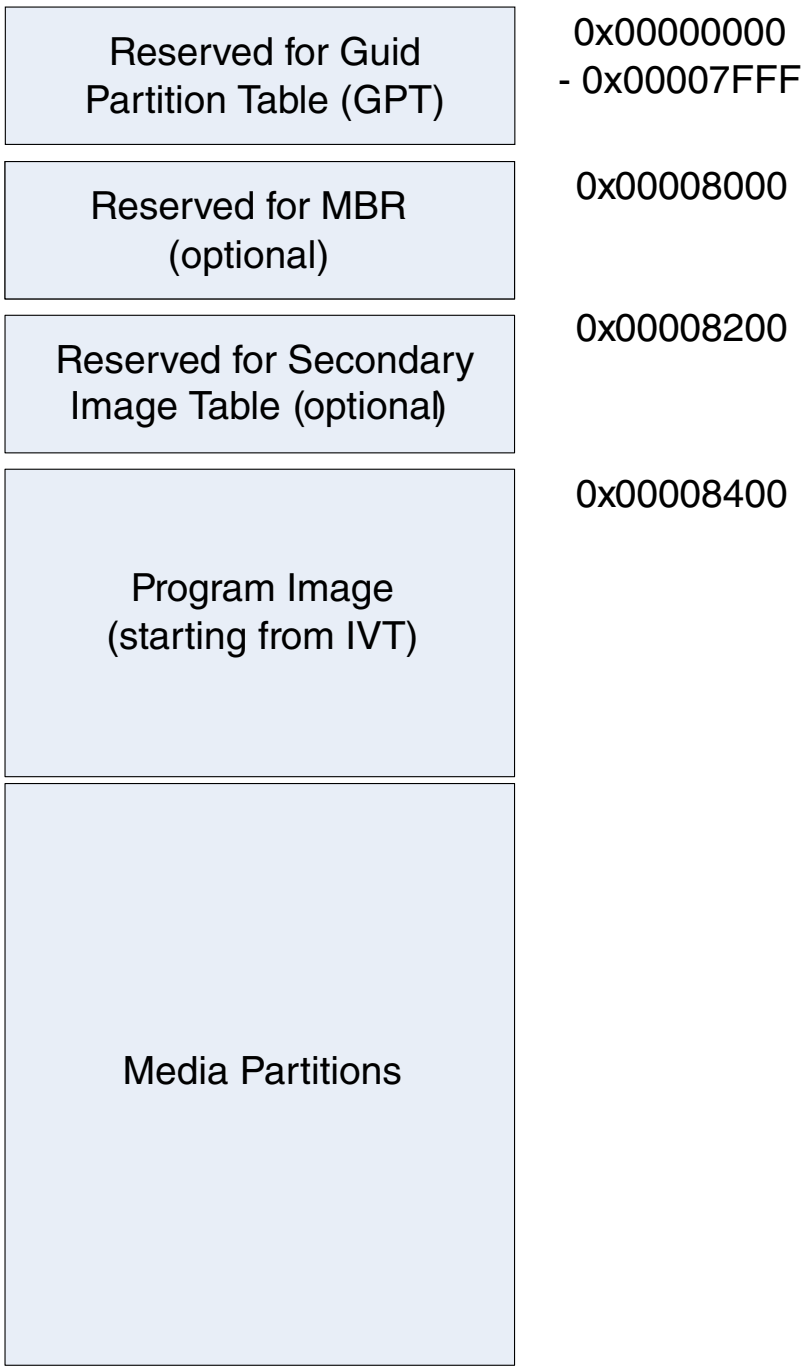


Figure 6-17. Expansion device structures layout

For the Closed mode, if there are failures during primary image authentication, the boot ROM turns on the PERSIST_SECONDARY_BOOT bit (see Table 6-8) and performs the software reset. (After the software reset, the secondary image is used.)

6.1.6 Program image

This section describes the data structures that are required to be included in the user's program image. The program image consists of:

- Image vector table—a list of pointers located at a fixed address that the ROM examines to determine where the other components of the program image are located.
- Boot data—a table that indicates the program image location, program image size in bytes, and the plugin flag.
- Device configuration data—IC configuration data.
- User code and data.

6.1.6.1 Image Vector Table and Boot Data

The Image Vector Table (IVT) is the data structure that the ROM reads from the boot device supplying the program image containing the required data components to perform a successful boot.

The IVT includes the program image entry point, a pointer to Device Configuration Data (DCD) and other pointers used by the ROM during the boot process.The ROM locates the IVT at a fixed address that is determined by the boot device connected to the Chip. The IVT offset from the base address and initial load region size for each boot device type is defined in the table below. The location of the IVT is the only fixed requirement by the ROM. The remainder or the image memory map is flexible and is determined by the contents of the IVT.

Table 6-19. Image Vector Table Offset and Initial Load Region Size

Boot Device Type	Image Vector Table Offset	Initial Load Region Size
NAND	1 Kbyte = 0x4000 bytes	8 Kbyte
SD/eSD/MMC/eMMC normal boot	33 Kbyte = 0x8400 bytes	8 Kbyte
eMMC Fast boot	1 Kbyte = 0x400 bytes	8 Kbyte

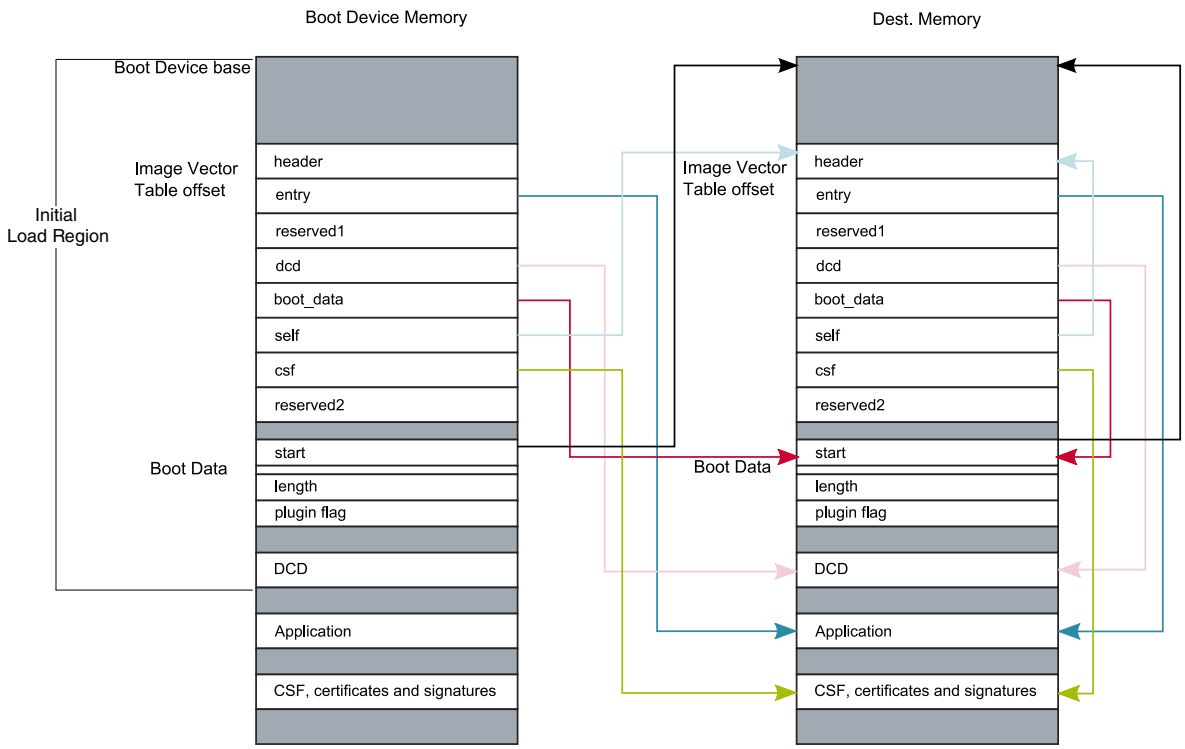


Figure 6-18. Image Vector Table

6.1.6.1.1 Image vector table structure

The IVT has the following format where each entry is a 32-bit word:

Table 6-20. IVT format

header
entry: Absolute address of the first instruction to execute from the image
reserved1: Reserved and should be zero
dcd: Absolute address of the image DCD. The DCD is optional so this field may be set to NULL if no DCD is required. See Device Configuration Data (DCD) for further details on the DCD.
boot data: Absolute address of the boot data
self: Absolute address of the IVT. Used internally by the ROM.
csf: Absolute address of the Command Sequence File (CSF) used by the HAB library. See High-Assurance Boot (HAB) for details on the secure boot using HAB. This field must be set to NULL when not performing a secure boot
reserved2: Reserved and should be zero

Figure 6-19 shows the IVT header format:

Figure 6-19. IVT header format

where:

```
Tag: A single byte field set to 0xD1
Length: a two byte field in big endian format containing the overall length of the IVT,
in bytes, including the header. (the length is fixed and must have a value of
32 bytes)
Version: A single byte field set to 0x40 or 0x41
```

6.1.6.1.2 Boot data structure

The boot data must follow the format defined in the table found here, each entry is a 32-bit word.

Table 6-21. Boot data format

start	Absolute address of the image
length	Size of the program image
plugin	Plugin flag (see Plugin image)

6.1.6.2 HDMI image boot up

The chip also supports HDMI image boot up for HDMI firmware. In addition to the standard image, there is a special image flagged as HDMI image (see Boot data structure). And normally it will come before the standard image.

ROM will check the HDMI image viability first. If the HDMI image is present and the HDMI device is not disabled by the eFuse, it will load the image into HDMI RAM and do the authentication. The image will include both the HDMI firmware and authentication information in it.

If the authentication is succeeded, then the firmware will be started and verified. ROM will also load the HDMI/HDCP related keys and then lock it.

If the authentication is failed, then the HDMI memory will be cleared and locked. If the HDMI is disabled, ROM will release the JTAG access at early boot up stage otherwise it will be released after HDMI image processing.

After processing HDMI image, the ROM process will move to standard image boot up sequence.

The HDMI image is generated by NXP CDT tool and firmware will be provided by customer. For more details about generating HDMI image, please refer to NXP CDT user guide.

The following figure shows the HDMI image boot flow. See High-level boot sequence for standard image boot flow.

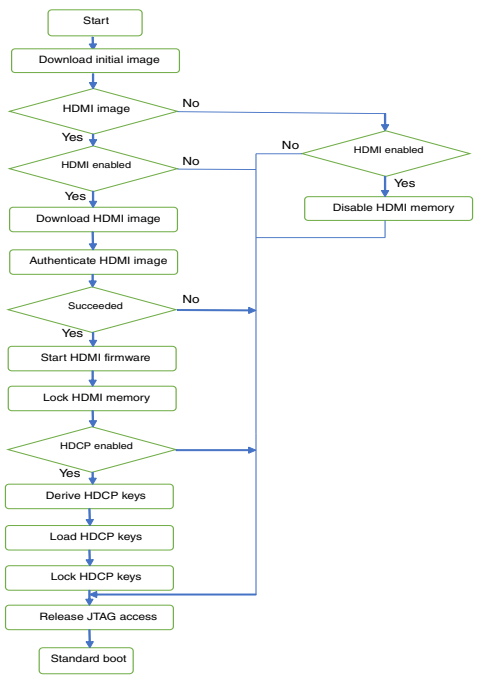


Figure 6-20. HDMI image boot flow

6.1.6.2.1 HDMI and A53 boot image location on boot device

The new requirement for i.MX 8M Dual/8M QuadLite/8M Quad ROM is to locate HDMI image on the boot device, load it into the HDMI block's i-ram and d-ram memories, authenticate it, and then finally locate, load, and authenticate the A53 boot image.

The chip can boot from the following boot media:

1. eMMC/SD
2. NAND

On the boot media, the HDMI image will be placed first followed by the A53 image. The distance between the two images is different for different types of boot media.

For eMMC/SD, the A53 image should start on the next 512-byte sector where HDMI image ends. Typically, here is the layout for eMMC/SD device for Normal and fast boot methods:

Table 6-22. eMMC/SD BOOT layout

eMMC/SD BOOT			
NORMAL BOOT	**SECTOR/LBA #**	**BYTE OFFSET**	**FAST BOOT**
32K SPACE FOR GUID PARTITION TABLE	0	0	START OF HDMI IMAGE
	1	512	
	2	1024	HDMI IMAGE IVT AND BOOT DATA
	…		
	…		…
	16	8192	96K HDMI IMAGE
	.	.	64K HDMI IMEM
	…		28K HDMI DMEM FOLLOWED BY COPY OF IVT AND BOOT DATA CSF DATA
START OF HDMI IMAGE	64	32768	
SECONDARY IMAGE TABLE	65		
HDMI IMAGE IVT AND BOOT DATA	66		
	…		
…	…		
92K HDMI IMAGE DATA	80	40960	
64K HDMI IMEM	…	…	
28K HDMI DMEM	208	106496	START OF A53 IMAGE
	210	107520	A53 IMAGE IVT AND BOOT DATA
	…	…	DCD
	…	…	…
	…	…	CSF DATA
	…	…	…
	…	…	
	…	…	A53 IMAGE DATA
COPY OF IVT AND BOOT DATA	264		
CSF DATA	264 Sector, Byte 64		
…	…		
START OF A53 IMAGE	272	139264	
SECONDARY IMAGE TABLE	273		
A53 IMAGE IVT AND BOOT DATA	274	140288	
DCD			
…			

Table continues on the next page...

Table 6-22. eMMC/SD BOOT layout (continued)

eMMC/SD BOOT			
NORMAL BOOT	**SECTOR/LBA #**	**BYTE OFFSET**	**FAST BOOT**
CSF DATA			
…	.		
	.		
A53 IMAGE DATA			
…			
SECOND COPY OF HDMI IMAGE			
SECOND COPY OF BOOT IMAGE			

For NAND device, the A53 image should start on the next new block where HDMI image ends. Typically, here is the layout for NAND device:

Table 6-23. NAND layout

NAND BOOT	
BLOCK #	**NAND BOOT**
0	FCB SEARCH AREA
…	…
4	DBBT SEARCH AREA
…	...
8	START OF HDMI IMAGE
	SECONDARY IMAGE TABLE
	HDMI IMAGE IVT AND BOOT DATA
	CSF DATA
	…
	96K HDMI IMAGE 64K HDMI IMEM 28K HDMI DMEM FOLLOWED BY COPY OF IVT AND BOOT DATA CSF DATA
9	START OF A53 IMAGE
	SECONDARY IMAGE TABLE
	A53 IMAGE IVT AND BOOT DATA
	DCD
	…
	CSF DATA
	…
	.
	A53 IMAGE DATA
	.
	.

Table continues on the next page...

Table 6-23. NAND layout (continued)

NAND BOOT	
BLOCK #	**NAND BOOT**
	.
	.
10	START OF HDMI IMAGE COPY 2
	.
	HDMI IMAGE IVT AND BOOT DATA
	CSF DATA
	…
	96K HDMI IMAGE 64K HDMI IMEM 28K HDMI DMEM FOLLOWED BY COPY OF IVT AND BOOT DATA CSF DATA
	.
	.
	.
	.
	.
	.
11	START OF A53 IMAGE COPY 2
	A53 IMAGE IVT AND BOOT DATA
	DCD
	…
	CSF DATA
	…
	…
	A53 IMAGE DATA
	.
	.
	.
	.

6.1.6.3 Device Configuration Data (DCD)

Upon reset, the chip uses the default register values for all peripherals in the system. However, these settings typically are not ideal for achieving the optimal system performance and there are even some peripherals that must be configured before they can be used.

The DCD is a configuration information contained in the program image (external to the ROM) that the ROM interprets to configure various peripherals on the chip.

For example, the EIM default settings allow the core to interface to a NOR flash device immediately after the reset. This allows the chip to interface with any NOR flash device, but has the disadvantage of slow performance. Additionally, some components (such as DDR) require some sequence of register programming as a part of the configuration before it is ready to be used. The DCD feature can be used to program the EIM registers and the DDR Controller registers to the optimal settings.

The ROM determines the location of the DCD table based on the information located in the Image Vector Table (IVT). See Image Vector Table and Boot Data for more details. The DCD table shown below is a big-endian byte array of the allowable DCD commands. The maximum size of the DCD is limited to 1768 B.

Figure 6-21. DCD data format

The DCD header is 4 B with the following format:

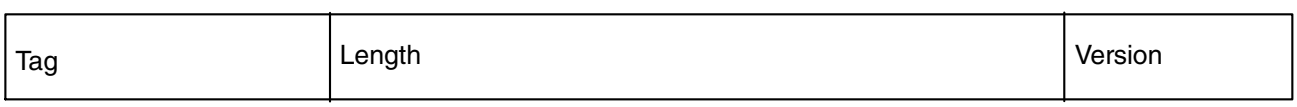


Figure 6-22. DCD header format

where:

```
Tag: A single-byte field set to 0xD2
Length: a two-byte field in the big-endian format containing the overall length of the DCD
(in bytes) including the header
Version: A single-byte field set to 0x41
```

6.1.6.3.1 Write data command

The write data command is used to write a list of given 1-, 2- or 4-byte values (or bitmasks) to a corresponding list of target addresses.

The format of the write data command (in a big-endian byte array) is shown in this table:

Table 6-24. Write data command format

Tag	Length	Parameter
Address		
Value/Mask		
[Address]		
[Value/Mask]		
...		
[Address]		
[Value/Mask]		

where:

```
Tag: a single-byte field set to 0xCC
Length: a two-byte field in a big-endian format, containing the length of the Write Data
Command (in bytes) including the header
Address: the target address to which the data must be written
Value/Mask: the data value (or bitmask) to be written to the preceding address
```

The parameter field is a single byte divided into the bitfields, as follows:

Table 6-25. Write data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of the target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only specific bits may be overwritten at the target address
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

One or more target address and value/bitmask pairs can be specified. The same bytes' and flags' parameters apply to all locations in the command.

When successful, this command writes to each target address in accordance with the flags as follows:

Table 6-26. Interpretation of write data command flags

"Mask"	"Set"	Action	Interpretation
0	0	*address = val_msk	Write value
0	1	*address = val_msk	Write value
1	0	*address &= ~val_msk	Clear bitmask
1	1	*address |= val_msk	Set bitmask

NOTE

If any of the target addresses does not have the same alignment as the data width indicated in the parameter field, none of the values are written.

If any of the values are larger or any of the bitmasks are wider than permitted by the data width indicated in the parameter field, none of the values are written.

If any of the target addresses do not lie within the allowed region, none of the values are written. The list of allowable blocks and target addresses for the chip are provided below.

6.1.6.3.2 Check data command

The check data command is used to test for a given 1-, 2-, or 4-byte bitmasks from a source address.

The check data command is a big-endian byte array with the format shown in this table:

Table 6-27. Check data command format

Tag	Length	Parameter
Address		
Mask		
[Count]		

where:

```
Tag: a single-byte field set to 0xCF
Length: a two-byte field in the big-endian format containing the length of the check data
command (in bytes) including the header
Address: the source address to test
Mask: the bit mask to test
Count: an optional poll count; If the count is not specified, this command polls
indefinitely
until the exit condition is met. If count = 0, this command behaves as for the NOP.
```

The parameter field is a single byte divided into bitfields, as follows:

Table 6-28. Check data command parameter field

7	6	5	4	3	2	1	0
flags					bytes		

where

```
bytes: the width of target locations in bytes (either 1, 2, or 4)
flags: control flags for the command behavior
Data Mask = bit 3: if set, only the specific bits may be overwritten at a target address
```

```
(otherwise all bits may be overwritten)
Data Set = bit 4: if set, the bits at the target address are overwritten with this flag
(otherwise it is ignored)
```

This command polls the source address until either the exit condition is satisfied, or the poll count is reached. The exit condition is determined by the flags as follows:

Table 6-29. Interpretation of check data command flags

"Mask"	"Set"	Action	Interpretation
0	0	(*address & mask) == 0	All bits clear
0	1	(*address & mask) == mask	All bits set
1	0	(*address & mask)!= mask	Any bit clear
1	1	(*address & mask)!= 0	Any bit set

NOTE

If the source address does not have the same alignment as the data width indicated in the parameter field, the value is not read.

If the bitmask is wider than permitted by the data width indicated in the parameter field, the value is not read.

6.1.6.3.3 NOP command

This command has no effect.

The format of the NOP command is a big-endian four-byte array, as shown in this table:

Table 6-30. NOP command format

Tag	Length	Undefined

where:

```
Tag: a single-byte field set to 0xC0
Length: a two-byte field in big endian containing the length of the NOP command in bytes
(fixed to a
value of 4)
Undefined: this byte is ignored and can be set to any value.
```

6.1.6.3.4 Unlock command

The unlock command is used to prevent specific engine features from being locked when exiting the ROM.

The format of the unlock command (in a big-endian byte array) is shown in this table:

Table 6-31. Unlock command format

Tag	Length	Eng
Value		
Value		
...		
Value		

where:

NOTE

This command may not be used in the DCD structure if the SEC_CONFIG is configured as closed.

6.1.7 Plugin image

The ROM supports a limited number of boot devices. When using other devices as a boot source (for example, Ethernet, CDROM, or USB), the supported boot device must be used (typically serial ROM) as a firmware to provide the missing boot drivers. Additionally, the plugin can be customized to support boot drivers, which is more flexible when performing the device initialization, such as condition judging, delay assertion, or to apply custom settings to the boot device and memory system.

In addition to the standard images, the chip also supports plugin images. The plugin images return the execution to the ROM whereas the standard image does not.

The boot ROM detects the image type using the plugin flag of the boot data structure (see Boot data structure). If the plugin flag is 1, then the ROM uses the image as a plugin function. The function must initialize the boot device and copy the program image to the final location. At the end, the plugin function must return with the program image parameters. (See High-level boot sequence for details about the boot flow).

The boot ROM authenticates the plugin image before running the plugin function and then authenticates the program image.

The plugin function must follow the API described below:

typedef BOOLEAN (*plugin_download_f)(void **start, size_t *bytes, UINT32 *ivt_offset);

ARGUMENTS PASSED:

- start - the image load address on exit.

- bytes - the image size on exit.
- ivt_offset - the offset (in bytes) of the IVT from the image start address on exit.

RETURN VALUE:

- 1 - success
- 0 - failure

6.1.8 Serial Downloader

The Serial Downloader provides a means to download a program image to the chip over the USB serial connection.

In this mode, the ROM programs the WDOG1 for a time-out specified by the fuse WDOG Time-out Select (See fusemap for details) if the WDOG_ENABLE eFuse is 1 and continuously polls for the USB connection. If no activity is found on the USB OTG1 and the watchdog timer expires, the Arm core is reset.

NOTE

After the downloaded image is loaded, it is responsible for managing the watchdog resets properly.

This figure shows the USB boot flow:

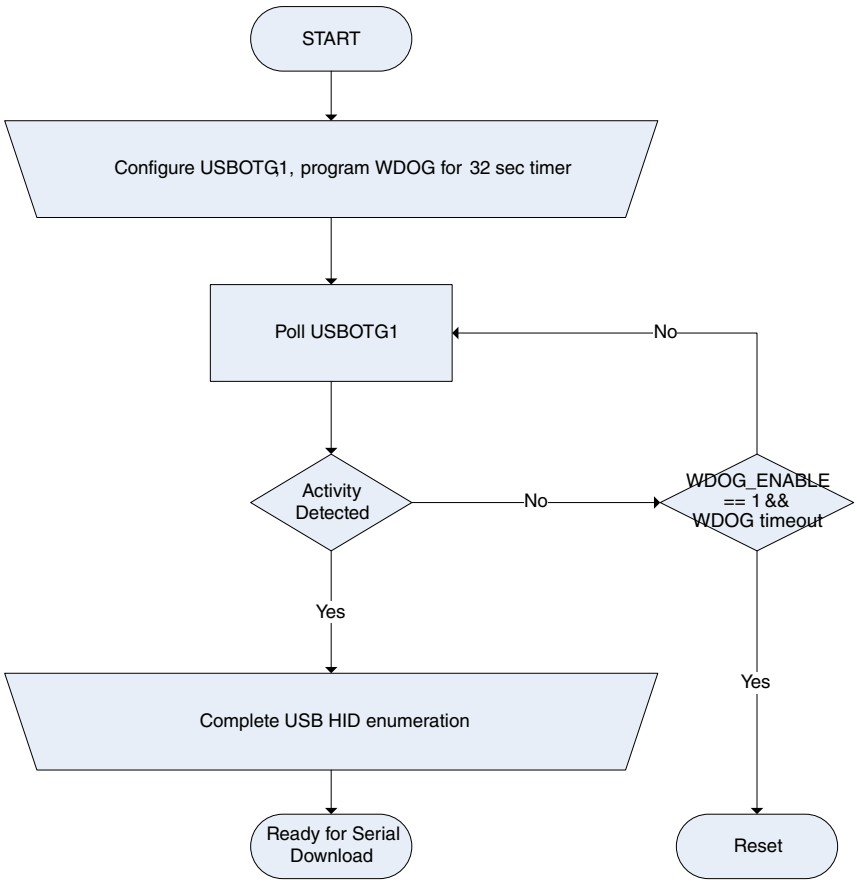


Figure 6-23. Serial Downloader boot flow

NOTE

Before going into USB serial mode, Boot ROM detect SD/MMC card on USDHC2 port. If a card is inserted, ROM will try to boot from it. This is the so-called Manufacture SD/MMC boot. This feature can be disabled by blowing fuse "Disable SD/MMC Manufacture Mode". See SD/MMC manufacture mode for details.

6.1.8.1 USB

The USB support is composed of the USB (core controller, compliant with the USB 3.0 specification) and the USBPHY (HS USB transceiver).

The ROM supports the USB OTG port for boot purposes. The other USB ports on the chip are not supported for boot purposes.

The USB Driver is implemented as a USB HID class. A collection of four HID reports are used to implement the SDP protocol for data transfers, as described in Table 6-32.

Table 6-32. USB HID reports

Report ID (first byte)	Transfer endpoint	Direction	Length	Description
1	control OUT	Host to device	17 B	SDP command from the host to the device.
2	control OUT	Host to device	Up to 1025 B	Data associated with the report 1 SDP command.
3	interrupt	Device to host	5 B	HAB security configuration. The device sends 0x12343412 in the closed mode and 0x56787856 in the open mode.
4	interrupt	Device to host	Up to 65 B	Data in response to the SDP command in report 1.

6.1.8.1.1 USB configuration details

The USB OTG function device driver supports a high speed (HS for UTMI) non-stream mode with a maximal packet size of 512 B and a low-level USB OTG function.

The VID/PID and strings for the are listed in the following table.

Table 6-33. VID/PID and strings for USB device driver

Descriptor	Value
VID	0x1FC9 (NXP vendor ID)
PID[1]	0x012B
String Descriptor1 (manufacturer)	NXP Semiconductors
String Descriptor2 (product)	SE Blank M850 SP Blank M850 NS Blank M850 FR Blank M850
String Descriptor4	NXP Flash
String Descriptor5	NXP Flash

1. Allocation based on the BPN (Before Part Number)

6.1.8.1.2 IOMUX configuration for USB

The interface signals of the UTMI PHY are not configured in the IOMUX,, except for the USB_OTGn_ID pins. The USB ID pin function is configured using the USBNC_n_CTRL2[ID_DIG_SEL] and the IOMUXC_USB_OTGn_ID_SELECT_INPUT register. The remaining pins of the UTMI PHY interface use the dedicated contacts on the IC. See the chip data sheet for details.

6.1.8.2 Serial Download Protocol (SDP)

The 16-byte SDP command from the host to device is sent using the HID report 1.

This table describes the 16-byte SDP command data structure:

Table 6-34. 16-byte SDP command data structure

BYTE offset	Size	Name	Description
0	2	COMMAND TYPE	These commands are supported for the ROM: • 0x0101 READ_REGISTER • 0x0202 WRITE_REGISTER • 0x0404 WRITE_FILE • 0x0505 ERROR_STATUS • 0x0A0A DCD_WRITE • 0x0B0B JUMP_ADDRESS
2	4	ADDRESS	Only relevant for these commands: READ_REGISTER, WRITE_REGISTER, WRITE_FILE, DCD_WRITE, and JUMP_ADDRESS. For the READ_REGISTER and WRITE_REGISTER commands, this field is the address to a register. For the WRITE_FILE and JUMP_ADDRESS commands, this field is an address to the internal or external memory address.
6	1	FORMAT	Format of access, 0x8 for an 8-bit access, 0x10 for a 16-bit access, and 0x20 for a 32-bit access. Only relevant for the READ_REGISTER and WRITE_REGISTER commands.
7	4	DATA COUNT	Size of the data to read or write. Only relevant for the WRITE_FILE, READ_REGISTER, WRITE_REGISTER, and DCD_WRITE commands. For the WRITE_FILE and DCD_WRITE commands, the DATA COUNT is in the byte units.
11	4	DATA	The value to write. Only relevant for the WRITE_REGISTER command.
15	1	RESERVED	Reserved

6.1.8.2.1 SDP commands

The SDP commands are described in the following sections.

6.1.8.2.1.1 READ_REGISTER

The transaction for the READ_REGISTER command consists of these reports: Report1 for the command, Report3 for the security configuration, and Report4 for the response or the register value.

The register to read is specified in the ADDRESS field of the SDP command. The first device sends Report3 with the security configuration followed by the Report4 with the bytes read at a given address. If the count is greater than 64, multiple reports with the report id 4 are sent until the entire data requested by the host is sent. The STATUS is either 0x12343412 for the closed parts and 0x56787856 for the open or field return parts.

Report1, Command, Host to Device:

1	Valid values for the READ_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT

ID 16-byte SDP command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host: first response report

4	Register value

ID 4 bytes of data containing the register value. If the number of bytes requested is less than 4, the remaining bytes must be ignored by the host.

Multiple reports of the report id 4 are sent until the entire requested data is sent.

Report4, Response, Device to Host: last response report

4	Register value

ID 64 bytes of data containing the register value. If the number of bytes requested is less than 64, the remaining bytes must be ignored by the host.

6.1.8.2.1.2 WRITE_REGISTER

The transaction for the WRITE_REGISTER command consists of these reports: Report1 for the command, Report3 for the security configuration and Report4 for the write status.

The host sends Report1 with the WRITE_REGISTER command. The register to write is specified in the ADDRESS field of the SDP command of Report1, with the FORMAT field set to the data type (number of bits to write, either 8, 16, or 32) and the value to write in the DATA field of the SDP command. The device writes the DATA to the register address and returns the WRITE_COMPLETE code using Report4 and the security configuration using Report3 to complete the transaction.

Report1, Command, Host to Device:

1	Valid values for WRITE_REGISTER COMMAND, ADDRESS, FORMAT, DATA_COUNT and DATA

ID 16-byte SDP command

Report3, Response, Device to Host:

3	4 bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	WRITE_COMPLETE (0x128A8A12) status

ID 64 bytes data with the first 4 bytes to indicate that the write is completed with code 0x128A8A12. On failure, the device reports the HAB error status.

6.1.8.2.1.3 WRITE_FILE

The transaction for the WRITE_FILE command consists of these reports: Report1 for the command phase, Report2 for the data phase, Report3 for the HAB mode, and Report4 to indicate that the data are received in full.

The size of each Report2 is limited to 1024 bytes (limitation of the USB HID protocol). Hence, multiple Report2 packets are sent by the host in the data phase until the entire data is transferred to the device. When the entire data (DATA_COUNT bytes) is received, the device sends Report3 with the HAB mode and Report4 with 0x88888888, indicating that the file download completed.

Report1, Host to Device:

1	Valid values for WRITE_FILE COMMAND, ADDRESS, DATA_COUNT

ID 16-byte SDP command

==========Optional Begin==========

Host sends the ERROR_STATUS command to query if the HAB rejected the address

========== Optional End==========

Report2, Host to Device:

2	File data

ID Max 1024 bytes data per report

Report2, Host to Device:

2	File data

ID Max 1024 bytes data per report

Report3, Device to Host:

3	4 bytes indicating security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	COMPLETE (0x88888888) status

ID 64 bytes data with the first four bytes to indicate that the file download completed with code 0x88888888. On failure, the device reports the HAB error status.

6.1.8.2.1.4 ERROR_STATUS

The transaction for the SDP command ERROR_STATUS consists of three reports.

Report1 is used by the host to send the command; the device sends global error status in four bytes of Report4 after returning the security configuration in Report3. When the device receives the ERROR_STATUS command, it returns the global error status that is updated for each command. This command is useful to find out whether the last command resulted in a device error or succeeded.

Report1, Command, Host to Device:

1	ERROR_STATUS COMMAND

ID 16-byte SDP Command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	Four bytes Error status

ID first 4 bytes status in 64 bytes Report4

6.1.8.2.1.5 DCD_WRITE

The SDP command DCD_WRITE is used by the host to send multiple register writes in one shot. This command is provided to speed up the process of programming the register writes (such as to configure an external RAM device).

The command goes with Report1 from the host with COMMAND TYPE set to DCD_WRITE, ADDRESS which is used as a temporary location of the DCD data, and DATA_COUNT to the number of bytes sent in the data out phase. In the data phase, the host sends the data for a number of registers using Report2. The device completes the transaction with Report3 indicating the security configuration and Report4 with the WRITE_COMPLETE code 0x12828212.

Report1, Command, Host to Device:

1	DCD_WRITE COMMAND, ADDRESS, DATA_COUNT

ID 16-byte SDP Command

Report2, Data, Host to Device:

2	DCD binary data

ID Max 1024 bytes per report

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	WRITE_COMPLETE (0x128A8A12) status

ID 64 bytes report with the first four bytes to indicate that the write completed with the code 0x128A8A12. On failure, the device reports the HAB error status.

See Device Configuration Data (DCD) for the DCD format description.

6.1.8.2.1.6 SKIP_DCD_HEADER

The SDP command SKIP_DCD_HEADER is used by the host to inform the device to skip the DCD configuration within the download image.

If the download image must be run on the DDR, the DCD configuration data must be built into the image. In case the host issued DCD_WRITE to push the DCD configuration data to the device for the DDR initialization, the image with the DCD information causes the ROM to initialize the DDR twice, and may cause the initialization processing to hang.

The SKIP_DCD_HEADER command informs the device to skip the DCD configuration within the download image and avoid this issue.

This command is typically sent after JUMP_ADDRESS. This command is sent by the host in the command-phase of the transaction using Report1, there is no data phase for this command. The device completes the transaction with Report3 indicating the security configuration and Report4 with the OK_ACK code 0x900DD009.

Report1, Command, Host to Device:

1	SKIP_DCD_HEADER

ID 16-byte SDP Command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

Report4, Response, Device to Host:

4	OK_ACK (0x900DD009)

6.1.8.2.1.7 JUMP_ADDRESS

The SDP command JUMP_ADDRESS is the last command that the host can send to the device. After this command, the device jumps to the address specified in the ADDRESS field of the SDP command and starts to execute.

This command usually follows after the WRITE_FILE command. The command is sent by the host in the command-phase of the transaction using Report1. There is no data phase for this command, but the device sends the status Report3 to complete the transaction. If the authentication fails, it also sends Report4 with the HAB error status.

Report1, Command, Host to Device:

1	JUMP_ADDRESS COMMAND, ADDRESS

ID 16-byte SDP Command

Report3, Response, Device to Host:

3	Four bytes indicating the security configuration

ID 4 bytes status

This report is sent by the device only in case of an error jumping to the given address, or if the device reports error in Report4, Response, Device to Host:

4	Four bytes HAB error status

ID 4 bytes status, 64 bytes report length

6.1.9 Low-power boot

The ROM supports the low-power boot. If the LPB_BOOT fuses are blown, the chip checks if there is a low-power condition via the pad. If there is a low-power boot condition, the ROM applies division factors on the ARM, DDR, AXI, and AHB root clocks based on the LPB_BOOT fuses value (see the table below). The polarity of the low-power boot condition on the pad is set by the BT_LPB_POLARITY fuse (see the following figure).

Table 6-35. Low-power boot frequencies

LPB_BOOT	Boot Frequencies=0	Boot Frequencies=1
00	ARM_A53_CLK_ROOT= 1000 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 333 MHz	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 166 MHz
01	ARM_A53_CLK_ROOT= 1000 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 333 MHz	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 166 MHz
10	ARM_A53_CLK_ROOT= 500 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 166 MHz	ARM_A53_CLK_ROOT= 250 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 83.3 MHz
11	ARM_A53_CLK_ROOT= 250 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 83.3 MHz	ARM_A53_CLK_ROOT= 250 MHz AHB_CLK_ROOT= 133 MHz MAIN_AXI_CLK_ROOT= 41.67 MHz

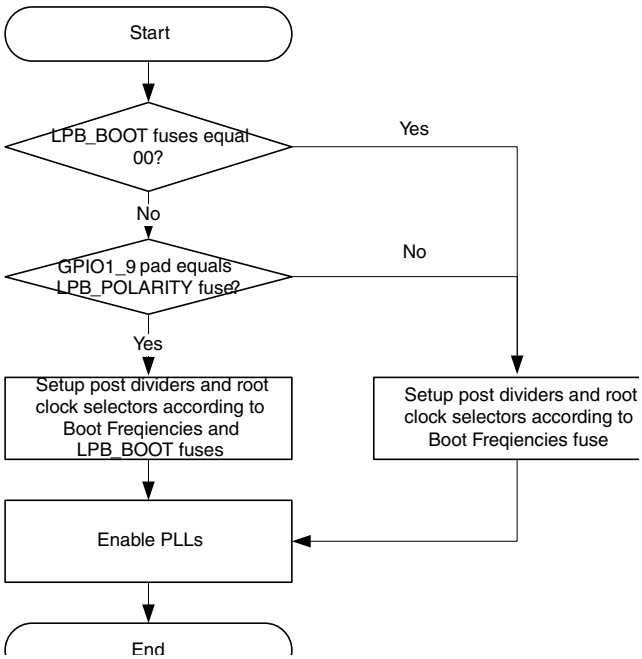


Figure 6-24. Low-power boot flow

6.1.10 SD/MMC manufacture mode

When the internal boot and recover boot (if enabled) failed, the boot goes to the SD/MMC manufacture mode before the serial download mode. In the manufacture mode, one bit bus width is used despite of the fuse setting.

By default, the SD/MMC manufacture mode is enabled. Blow the fuse of the DISABLE_SDMMC_MFG to disable it.

NOTE

A secondary boot is not supported in the SD/MMC manufacture mode.

NOTE

ROM does not support loading of HDMI image in SD/MMC manufacturing mode.

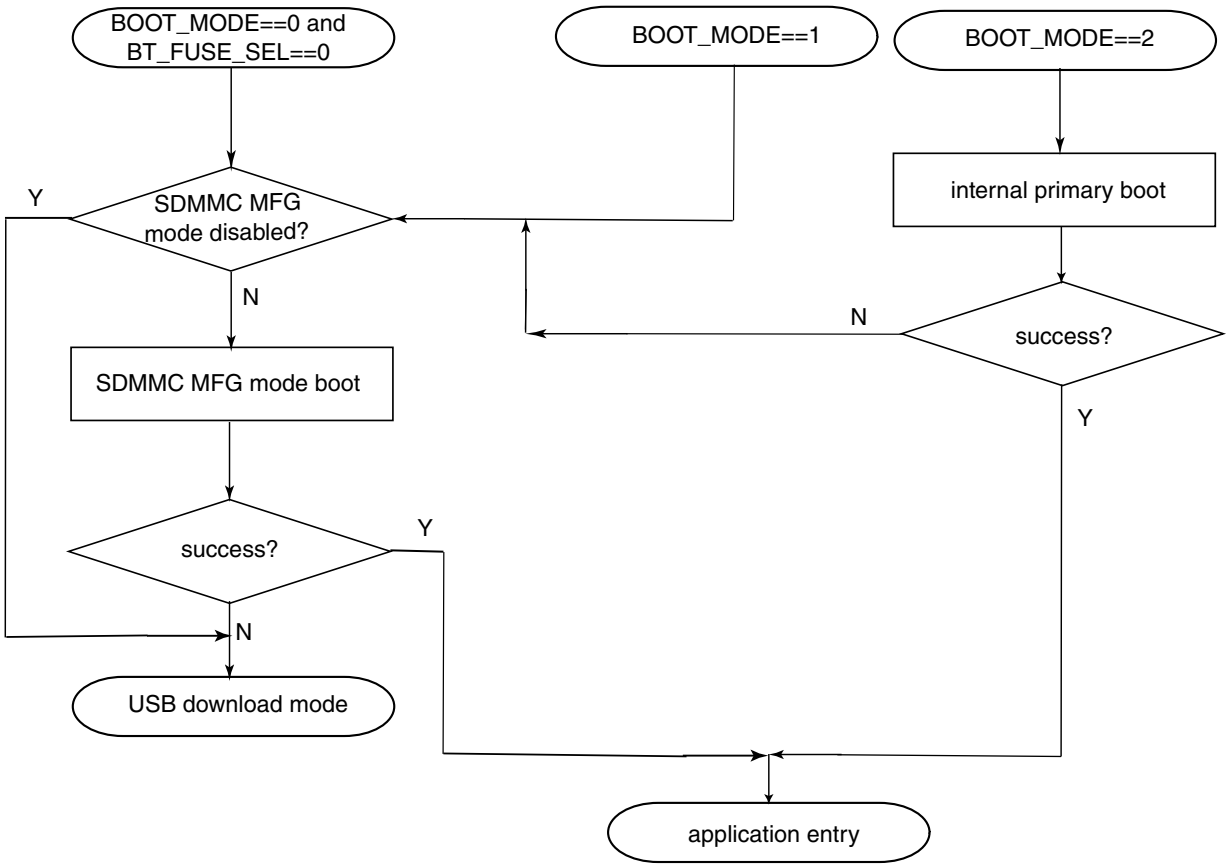


Figure 6-25. SD/MMC manufacture boot flow

NOTE

The EEPROM recovery mode is not supported on this chip.

6.1.11 High-Assurance Boot (HAB)

The High Assurance Boot (HAB) component of the ROM protects against the potential threat of attackers modifying the areas of code or data in the programmable memory to make it behave in an incorrect manner. The HAB also prevents the attempts to gain access to features which must not be available.

The integration of the HAB feature with the ROM code ensures that the chip does not enter an operational state if the existing hardware security blocks detected a condition that may be a security threat or if the areas of memory deemed to be important were modified. The HAB uses the RSA digital signatures to enforce these policies.

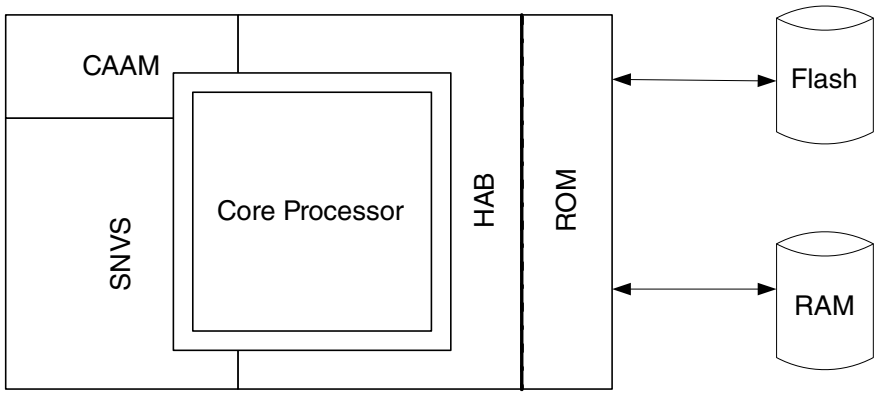


Figure 6-26. Secure boot components

The figure above illustrates the components used during a secure boot using HAB. The HAB interfaces with the SNVS to make sure that the system security state is as expected. The HAB also uses the CAAM hardware block to accelerate the SHA-256 message digest operations performed during the signature verifications and AES-128 operations for the encrypted boot operations. The HAB also includes a software implementation of SHA-256 for cases where a hardware accelerator can't be used. The RSA key sizes supported are 1024, 2048, 3072, and 4096 bits. The RSA signature verification operations are performed by a software implementation contained in the HAB library. The main features supported by the HAB are:

- X.509 public key certificate support
- CMS signature format support
- Proprietary encrypted boot support. Note that the encrypted boot depends on the CAAM hardware module. When the CAAM is disabled (when the EXPORT_CONTROL fuse is blown), the encrypted boot is not available.

NOTE

NXP provides the reference Code Signing Tool (CST) for key generation, certificate generation, and code signing for use with the HAB library. The CST can be found by searching for "IMX_CST_TOOL" at http://www.nxp.com.

NOTE

For further details on using the secure boot feature using HAB, refer to *Secure Boot on i.MX 50, i.MX 53, i.MX 6 and i.MX 7 Series using HABv4 (AN4581).*

6.1.11.1 HAB API vector table addresses

For devices that perform a secure boot, the HAB library may be called by the boot stages that execute after the ROM code.

NOTE

For additional information on the secure boot including the HAB API, refer to *HABv4 RVT Guidelines and Recommendations (AN12263).*

6.1.12 Boot information for software

To address the requirement that the boot image may need to get the basic boot information when getting out of the boot process, the boot software information (Boot_SW_Info) is provisioned by the ROM.

The base address of the Boot_SW_Info (the SW information pointer is recorded at 0x9e8 for ROM A0 and 0x968 for ROM B0). The software must read 0x9e8 for ROM A0 or 0x968 for ROM B0 to get the base address of the Boot_SW_Info, and parse the Boot_SW_Info content to get the boot information.

Table 6-36. Boot_SW_Info structure

Offset	Byte3	Byte2	Byte1	Byte0
0x0	Reserved	Boot Device Type	Boot Device Instance	Reserved
0x4	Arm core frequency (in Hz)			
0x8	AXI bus frequency (in Hz)			
0xC	DDR frequency (in Hz)			
0x10	GPT1 input clock frequency (in Hz)			
0x14	Reserved			
0x18				
0x1C				

NOTE

The boot ROM sets the GPT1 in a free-running mode with a 32-kHz input clock.

Boot device type mapping:

- 0x1 - SD card or eSD chip
- 0x2 - MMC card or eMMC chip
- 0x3 - NAND chip

Boot device instance: The instance index of the boot device, starting from 0.

6.2 Fusemap

6.2.1 Boot Fusemap

The following section details the various modes and selection of the required boot devices. A separate map is given for each and every boot device. The device select is specified by BOOT_CFG[14:12] fuses listed below.

Table 6-37. Boot Device Select

Boot Device	BOOT_CFG[14]	BOOT_CFG[13]	BOOT_CFG[12]
SD/eSD	0	0	1
MMC/eMMC	0	1	0
NAND	0	1	1

NOTE

Fuses marked as “Reserved” are reserved for NXP internal (and future) use only. Customers should not attempt to burn these, as the IC behavior may be unpredictable. The reserved fuses can be read as either 0 or 1.

Table 6-38. SD/eSD Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x470[15:8]	Reserved	Boot Device Select			Port Select: 00 - SD1 01 - SD2		Power Cycle Enable 0 - Disable 1 - Enable	SD Loopback Clock Source SEL (SDR50 and SDR104 Only) 0 - Disable 1 - Enable
0x470[7:0]	Fast Boot: 0 - Regular 1 - Fast Boot	Reserved		Bus Width: 0 - 1-bit 1 - 4-bit	Speed 000 - Normal/SDR12 001 - High/SDR25 010 - SDR50 011 - SDR104 101 - Reserved for DDR50 Others - Reserved			Reserved

Table 6-39. MMC/eMMC Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x470[15:8]	Reserved	Boot Device Select			Port Select: 00 - SD1 01 - SD2		Power Cycle Enable 0 - Disable 1 - Enable	SD Loopback Clock Source SEL (SDR50 and SDR104 Only) 0 - Disable 1 - Enable
0x470[7:0]	Fast Boot: 0 - Regular 1 - Fast Boot	Bus Width: 000 - 1-bit 001 - 4-bit 010 - 8-bit 101 - 4-bit DDR (MMC 4.4) 110 - 8-bit DDR (MMC 4.4) Else - reserved			Speed 00 - Normal 01 - High 10 - Reserved for HS200 11 - Reserved		USDHC1 IO VOLTAGE SELECTION 0 - 3.3V 1 - 1.8V	USDHC2 IO VOLTAGE SELECTION 0 - 3.3V 1 - 1.8V

Table 6-40. NAND Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x470[15:8]	Reserved	Boot Device Select			Pages in Block: 00 - 128 01 - 64 10 - 32 11 - 256		NAND Row Address Bytes: 00 - 3 01 - 2 10 - 4 11 - 5	
0x470[7:0]	BT_TOGGLEMODE 0 - Raw NAND 1 - Toggle mode	BOOT_SEARCH_COUNT 00 - 2 01 - 2 10 - 4 11 - 8	Toggle Mode 33MHz Preamble Delay, Read Latency 000 - 16 GPMICLK cycles 001 - 1 GPMICLK cycles 010 - 2 GPMICLK cycles 011 - 3 GPMICLK cycles 100 - 4 GPMICLK cycles 101 - 5 GPMICLK cycles 110 - 6 GPMICLK cycles 111 - 7 GPMICLK cycles 1111- 15 GPMICLK cycles					Reserved

Table 6-41. Boot Fusemap

Addr	7	6	5	4	3	2	1	0
0x480[7:0]	Reserved							
0x480[15:8]	LPB_BOOT (Core/DDR/ Bus) 00/01 - LPB Disable 10 - Div by2 11 - Div by 4		BT_LPB_POLARITY (GPIO polarity)	L1 I-Cache DISABLE	TZASC_ENABLE	WDOG_ENABLE 0 - Disabled 1 - Enabled	Boot Frequencies (ARM/DDR) 0 - 800 / 800 MHz 1 - 400 / 400 MHz	Reserved
0x480[23:16]	NOC_ID_REMAP_BYPASS	SDP_READ_DISABLE	SDP_DISABLE	FORCE_INTERNAL_BOOT	Reserved	WDOG Timeout Select 000 - 64s 001 - 32s 010 - 16s 011 - 8s 100 - 4s Others - Reserved		
0x480[31:24]	Reserved							UART Serial Download Disable 0 - Enabled 1 - Disabled
0x490[7:0]	Reserved for uSDHC future use							
0x490[15:8]	USDHC DLL Select 0 - DLL Slave Mode for 1 - DLL Override Mode	MMC_DLL_DLY[6:0] Delay target for USDHC DLL, it is applied to slave mode target delay or override mode target delay depends on DLL Override fuse bit value.						
0x490[23:16]	Disable SDMMC Manufacture mode 0 - Enable 1 - Disable	USDHC_CMD_OE_PRE_EN (SD/MMC debug)	eMMC 4.4 - RESET TO PRE-IDLE STATE 0 - enable 1 - disable	USDHC_PAD_PULL_DOWN 0 - no action 1 - pull down	ENABLE_EMMC_22K_PULLUP 0 - 47K pullup 1 - 22K pullup	USDHC_IOMUX_SION_BIT_ENABLE 0 - Disable 1 - Enable	USDHC Override Pad Settings (using PAD_SETTINGS value)	USDHC DLL_ENABLE 0 - Disable DLL for SD/ eMMC 1 - Enable DLL for SD/ eMMC
0x490[31:24]	USDHCPAD_SETTINGS[7:0]							
0x4A0[7:0]	SD Calibration Step 00 - 1		uSDHC Power Cycle Interval 00 - 20ms 01 - 10ms		uSDHC Power Cycle Delay 0 - 5ms	uSDHC Power On Polarity 0 - Low	DCD Size 0 - Default Setting	Fast Boot Acknowledge Disable 0 - Boot Ack Disabled

Table continues on the next page...

Table 6-41. Boot Fusemap (continued)

Addr	7	6	5	4	3	2	1	0
			10 - 5ms 11 - 2.5		1 - 2.5ms	1 - High	1 - Force to 100K	1 - Boot Ack Enabled
0x4A0[15:8]	NAND_READ_CMD_CODE1[7:0]							
0x4A0[23:16]	NAND_READ_CMD_CODE2[7:0]							
0x4A0[31:24]	NAND_PAD_SETTINGS[7:0]							
0x4B0[7:0]	Override NAND Pad Settings (using PAD_SETTINGS value)	GPMI Read DDR DLL Target Value 0000 - 7 0001 - 1 0111 - 0 1111 - 15				0x4B0[1:0] NAND Number of Devices: 00 - 1 01 - 2 10 - 4 11 - Reserved		
0x4B0[15:8]	READ_RETRY_SEQ_ID[3:0] 0000 - do not use read retry (RR) sequence embedded in ROM 0001 - Micron 20nm RR sequence 0010 - Toshiba A19nm RR sequence 0011 - Toshiba 19nm RR sequence 0100 - SanDisk 19nm RR sequence 0101 - SanDisk 1ynmRR sequence 0110 - Hynix 20nm A Die RR sequence 0111 - Hynix 26nm RR sequence 1000 - Hynix 20nm B Die RR sequence 1001 - Hynix 20nm C Die RR sequence Others - Reserved				Reserved			
0x4B0[23:16]	Reserved							
0x4B0[31:24]	RNG_TRIM[7:0]							

6.2.2 Lock Fusemap

Table 6-42 describes the functions of various lock fuses.

Table 6-42. Lock Fuses

Addr	7	6	5	4	3	2	1	0
0x400[7:0]	Reserved		Reserved		BOOT_CFG_LOCK 1x - OP x1 - WP		TESTER_LOCK 1x - OP x1 - WP	

Table continues on the next page...

Table 6-42. Lock Fuses (continued)

Addr	7	6	5	4	3	2	1	0
0x400[15:8]	MAC_ADDR_LOCK 1x - OP x1 - WP		USB_ID_LOCK 1x - OP x1 - WP		Reserved	SJC_RESP_LOCK WRP,OP,RDP	SRK_LOCK	Reserved
0x400[23:16]	GP2_LOCK 1x - OP x1 - WP		GP1_LOCK 1x - OP x1 - WP		Reserved		Reserved	Reserved
0x400[31:24]	Reserved							

NOTE

TESTER_LOCK programmed by NXP / set at factory

6.2.3 Fusemap Descriptions Table

NOTE

Definitions for fuse settings are as follows:

- Unlock - The controlled field can be read, sensed, burned, or overridden in the corresponding OCOTP shadow register.
- Override Project (OP) - The controlled field can be read, sensed, or burned in the corresponding OCOTP shadow register.
- Write Project (WP) - The controlled field can be read, sensed, or overridden in the corresponding OCOTP shadow register.
- OP + WP - The controlled field can only be read or sensed in the corresponding OCOTP shadow register. It cannot be burned or overridden.
- Lock - The controlled field cannot be read, burned, or overridden.

Table 6-43. Fusemap Descriptions

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x400[1:0]	TESTER_LOCK	2	Lock for tester related fuses at 0x400-0x460.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the	OCOTP

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	
0x400[3:2]	BOOT_CFG_LOCK	2	Lock for BOOT related fuses at 0x470-4B0.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	OCOTP
0x400[8:4]	Reserved	5	Reserved	Reserved	Reserved
0x400[9]	SRK_LOCK	1	Lock for SRK_HASH[255:0] fuses.	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register)	OCOTP

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				1 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	
0x400[10]	SJC_RESP_LOCK	1	Lock for SJC_RESP[55:0] fuses.	0 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 1 - Lock (The controlled field can't be read, overridden nor burned)	OCOTP
0x400[11]	Reserved	1	Reserved	Reserved	Reserved
0x400[13:12]	USB_ID_LOCK	2	Lock for USB_PID and USB_VID fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	OCOTP
0x400[15:14]	MAC_ADDR_LOCK	2	Lock for MAC_ADDR fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register)	OCOTP

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	
0x400[19:16]	Reserved	4	Reserved	Reserved	Reserved
0x400[21:20]	GP1_LOCK	2	Lock for GP1[63:0] fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register) 01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned) 11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)	OCOTP
0x400[23:22]	GP2_LOCK	2	Lock for GP2[63:0] fuses.	00 - Unlock (The controlled field can be read, sensed, burned or overridden in the corresponded OCOTP shadow register) 10 - OP (Override Protect, the controlled field can be read, sensed or burned, but can't be overridden in the corresponded OCOTP shadow register)	OCOTP

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

<table>
<tr><th>Fuse Address</th><th>Fuses Name</th><th>Number of Fuses</th><th>Fuses Function</th><th>Setting</th><th>Used by</th></tr>
<tr><td></td><td></td><td></td><td></td><td>01 - WP (Write Protect, the controlled field can be read, sensed or overridden in the corresponded OCOTP shadow register, but can't be burned)
11 - OP + WP (The controlled field can be read or sensed only, but can't be burned or overridden in the corresponded OCOTP shadow register)</td><td></td></tr>
<tr><td>0x400[31:24]</td><td>Reserved</td><td>8</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x420[31:0]</td><td>SJC_CHALL[63:0] / UNIQUE_ID[63:0]</td><td>32</td><td>The SJC CHALLENGE / Unique ID</td><td>-</td><td>SJC, SW</td></tr>
<tr><td>0x430</td><td>Reserved</td><td>32</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[7:0]</td><td>Reserved</td><td>8</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x440[9:8]</td><td>SPEED_GRADING[1:0]</td><td>2</td><td>Burned by tester program, for indicating IC core speed. (Hot burn may not be used).</td><td><table><tr><th>FCA[5:4]</th><th>FHA[3:2]</th><th>FRAL[1:0]</th><th>MHz</th><th>P/N</th></tr><tr><td>xx</td><td>xx</td><td>00</td><td>800</td><td>08</td></tr><tr><td>xx</td><td>xx</td><td>01</td><td>1000</td><td>10</td></tr><tr><td>xx</td><td>xx</td><td>10</td><td>1300</td><td>13</td></tr><tr><td>xx</td><td>xx</td><td>11</td><td>1500</td><td>15</td></tr></table></td><td>PROD / SW</td></tr>
<tr><td>0x440[31:10]</td><td>Reserved</td><td>22</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x450[1:0]</td><td>NUM_A53_CORES</td><td>2</td><td>Number of A53 CPU cores available.</td><td>00 - 4 cores
01 - 1 cores
10 - 2 cores
11 - Reserved</td><td>SRC, SJC, SW</td></tr>
<tr><td>0x450[7:2]</td><td>Reserved</td><td>6</td><td>Reserved</td><td>Reserved</td><td>Reserved</td></tr>
<tr><td>0x450[8]</td><td>M4_DISABLE</td><td>1</td><td>Disable M4 Core.</td><td>0 - enabled
1 - disabled</td><td>M4</td></tr>
<tr><td>0x450[9]</td><td>M4_MPU_DISABLE</td><td>1</td><td>Disable M4 MPU IP.</td><td>0 - enabled
1 - disabled</td><td>M4</td></tr>
<tr><td>0x450[10]</td><td>M4_FPU_DISABLE</td><td>1</td><td>Disable M4 FPU IP.</td><td>0 - enabled
1 - disabled</td><td>M4</td></tr>
</table>

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x450[11]	USB_OTG1_DISABLE	1	Disable USB OTG1 IP.	0 - enabled 1 - disabled	USB OTG1
0x450[12]	USB_OTG2_DISABLE	1	Disable USB OTG2 IP.	0 - enabled 1 - disabled	USB OTG2
0x450[13]	DOLBY_DISABLE	1	Disable Dolby HDR IP.	0 - enabled 1 - disabled	Display
0x450[15:14]	Reserved	2	Reserved	Reserved	Reserved
0x450[16]	EXPORT_CONTROL	1	Used for disabling CAAM and SNVS encryption	0 - Secure part 1 - Security disable (CAAM encryption disabled)	CAAM, SW(ROM)
0x450[17]	SEC_CONFIG[0]	1	Security Configuration Mode, and block of debugging of security HW, by JTAG.	Combined with SEC_CONFIG[1], provide FAB/Open/Close security states: 00 - FAB (Open) 01 - Open - allows any code to be flashed and executed, even if it has no valid signature. 1x - Closed (Security On) Also - used for 'blocking' debug of security modules, when burned.	SJC, SNVS, , SRC, tpsmp
0x450[18]	VPU_VP9_DISABLE	1	Disable VP9 Decoder in VPU	0 - enabled 1 - disabled	VPU
0x450[19]	VPU_HEVC_DISABLE	1	Disable HEVC Decoder in VPU	0 - enabled 1 - disabled	VPU
0x450[20]	VPU_AVC_DISABLE	1	Disable AVC Decoder in VPU	0 - enabled 1 - disabled	VPU
0x450[21]	VPU_DISABLE	1	Disable VPU IP	0 - enabled 1 - disabled	VPU
0x450[22]	PCIE1_DISABLE	1	Disable PCIe-1 IP.	0 - enabled 1 - disabled	PCIE
0x450[23]	PCIE2_DISABLE	1	Disable PCIe-2 IP.	0 - enabled 1 - disabled	PCIE
0x450[24]	GPU_DISABLE	1	Disable GPU IP.	0 - enabled 1 - disabled	GPU
0x450[25]	HDMI_DISABLE	1	Disable HDMI IP.	0 - enabled 1 - disabled	HDMI

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x450[26]	DC_DISABLE	1	Disable Display Controller IP.	0 - enabled 1 - disabled	Display
0x450[27]	HDCP_DISABLE	1	Disable HDCP Function	0 - enabled 1 - disabled	HDMI
0x450[28]	MIPI_DSI_DISABLE	1	Disable MIPI DSI IP.	0 - enabled 1 - disabled	MIPI DSI
0x450[29]	ENET_DISABLE	1	Disable ENET IP.	0 - enabled 1 - disabled	ENET
0x450[30]	MIPI_CSI1_DISABLE	1	Disable MIPI CSI1 IP.	0 - enabled 1 - disabled	MIPI CSI
0x450[31]	MIPI_CSI2_DISABLE	1	Disable MIPI CSI2 IP.	0 - enabled 1 - disabled	MIPI CSI
0x460	Reserved	32	Reserved	Reserved	Reserved
0x470[15:0]	BOOT_CFG	16	BOOT configuration register, Usage varies, depending on selected boot device.	See boot fusemap for details.	SRC SW(ROM)
0x470[24:16]	Reserved	9	Reserved	Reserved	Reserved
0x470[25]	SEC_CONFIG[1]	1	Security Configuration (with SEC_CONFIG[0])	00 - FAB (Open) 01 - Open - allows any code to be flashed and executed, even if it has no valid signature. 1x - Closed (Security On)	SW (ROM), SRC, SNVS, TPSMP
0x470[26]	Reserved	1	Reserved	Reserved	Reserved
0x470[28]	BT_FUSE_SEL	1	Determines, whether using fuses for boot configuration, or GPIO /Serial loader.	If boot_mode="00" (Development) 0=Boot mode configuration is taken from GPIOs. 1=Boot mode configuration is taken from fuses. If boot_mode="10" (Production) 0 - Boot using Serial Loader (USB) 1- Boot mode configuration is taken from fuses.	SRC SW(ROM)
0x470[29]	FORCE_COLD_BOOT(SBMR)	1	Force cold boot when A53 core come out of reset. Reflected in SBMR reg of SRC	Fuse Function: 0 – Default behavior allowing a fast recovery from low power modes. That is, the ROM is allowed to jump to	SRC SW(ROM)

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
				the address previously programmed in the SRC persistent register. 1 – Fast recovery path in the ROM is not allowed and a cold boot is always performed. Customers wanting a higher level of security should burn this fuse.	
0x470[31:30]	Reserved	2	Reserved	Reserved	Reserved
0x480[31:0]	BOOT_CFG_PARAMETER	32	BOOT configuration parameters, Usage varies, depending on selected boot device.	See boot fusemap for details.	SW (ROM)
0x490[31:0]	BOOT_CFG_PARAMETER	32	BOOT configuration parameters, Usage varies, depending on selected boot device.	See boot fusemap for details.	SW (ROM)
0x4A0[31:0]	BOOT_CFG_PARAMETER	32	BOOT configuration parameters, Usage varies, depending on selected boot device.	See boot fusemap for details.	SW (ROM)
0x4B0[31:0]	BOOT_CFG_PARAMETER	32	BOOT configuration parameters, Usage varies, depending on selected boot device.	See boot fusemap for details.	SW (ROM)
0x4C0-0x500	Reserved	384	Reserved	Reserved	Reserved
0x580[31:0]	SRK_HASH[255:0]	256	SRK key, no HW visible lines. NO HW Visible signals available	-	SW (HAB)
0x600[23:0]	SJC_RESP[55:0]	56	Response reference value for the secure JTAG controller	-	SJC
0x610[31:24]	Reserved	8	Reserved	Reserved	Reserved
0x620[15:0]	USB_VID[31:0]	16	USB VID	-	SW
0x620[31:16]	USB_PID[31:0]	16	USB PID	-	SW
0x630	Reserved	32	Reserved	Reserved	Reserved
0x640[15:0]	MAC_ADDR[47:0]	48	Reserved for customers/SW	-	SW
0x650[31:16]	Reserved	48	Reserved	Reserved	Reserved
0x670-0x700	Reserved	544	Reserved	Reserved	Reserved
0x780[31:0]	GP1[63:0]	64	General Purpose fuse register #1	-	SW
0x7A0[31:0]	GP2[63:0]	64	General Purpose fuse register #2	-	SW
0x7C0-0x7E0	Reserved	96	Reserved	Reserved	Reserved

Table continues on the next page...

Table 6-43. Fusemap Descriptions (continued)

Fuse Address	Fuses Name	Number of Fuses	Fuses Function	Setting	Used by
0x7F0[23:0]	Reserved	24	Reserved	Reserved	Reserved
0x7F0[31:24]	HDCP_EXPONENT[7:0]	8	HDCP exponent	-	HDCP
0x800-0x13F0	Reserved	6144	Reserved	Reserved	Reserved

6.3 On-Chip OTP Controller (OCOTP_CTRL)

6.3.1 Overview

This section contains information describing the requirements for the on-chip eFuse OTP controller along with details about the block functionality and implementation.

In this document, the words "eFuse" and "OTP" are interchangeable. OCOTP refers to the hardware block itself.

6.3.1.1 Features

The OCOTP provides the following features:

- Loading and housing of fuse content into shadow registers.
- Generation of HWV_FUSE (hardware visible fuse bus) and the HWV_REG bus which is made up of volatile PIO register based "fuses". The HWV_REG bits come from the SCS (Software Controllable Signals) register.
- Generation of STICKY_REG which is consist of sticky register bits.
- Provide program-protect and read-protect eFuse.
- Provide override and read protection of shadow register.

6.3.2 Clocks

The table found here describes the clock sources for OCOTP.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 6-44. OCOTP Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock

6.3.3 Top-Level Symbol and Functional Overview

The figure found here shows the OCOTP system level diagram.

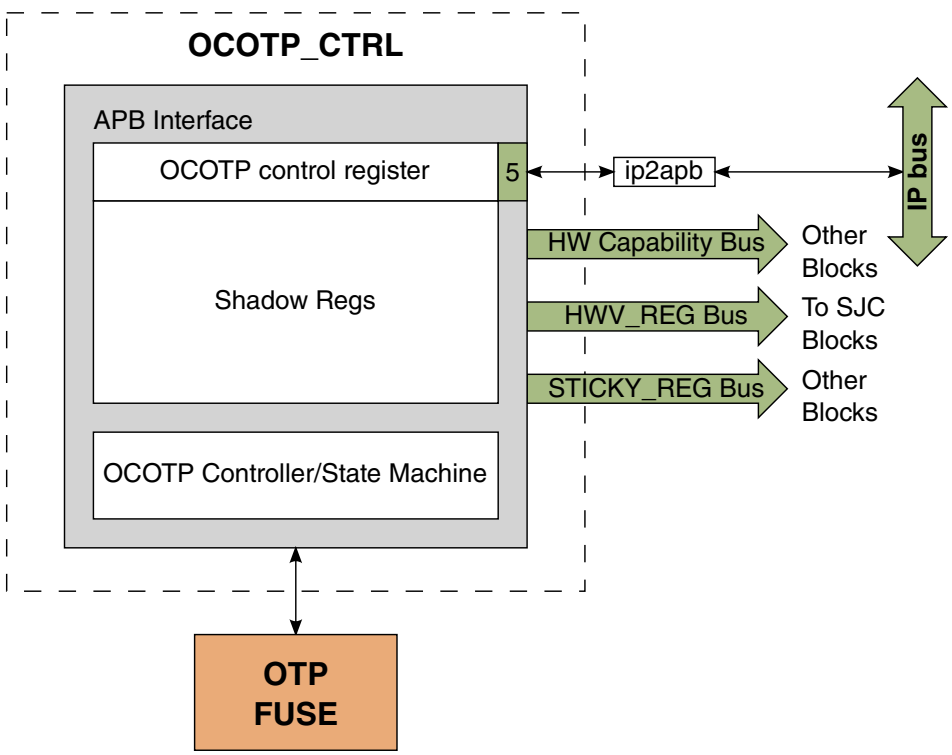


Figure 6-27. OCOTP System Level Diagram

6.3.3.1 Operation

The IP bus interface of the OCOTP provides two functions.

- Configure control registers for programming and reading fuse .
- Override and read shadow registers.

6.3.3.1.1 Shadow Register Reload

All fuse words in are shadowed. Therefore, fuse information is available through memory mapped shadow registers. If fuses are subsequently programmed, the shadow registers should be reloaded to keep them coherent with the fuse bank arrays.

The "reload shadows" feature allows the user to force a reload of the shadow registers (including HW_OCOTP_LOCK) without having to reset the device. To force a reload, complete the following steps:

1. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write , read or reload must be completed before a new access can be requested.
2. Set the HW_OCOTP_CTRL[RELOAD_SHADOWS] bit. OCOTP will read all the fuse one by one and put it into corresponding shadow register.
3. Wait for HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[RELOAD_SHADOWS] to be cleared by the controller.

The controller will automatically clear the HW_OCOTP_CTRL[RELOAD_SHADOWS] bit after the successful completion of the operation.

6.3.3.1.2 Fuse and Shadow Register Read

All shadow registers are always readable through the APB bus except some secret keys regions. When their corresponding fuse lock bits are set, the shadow registers also become read locked. After read locking, reading from these registers will return 0xBADABADA.

In addition HW_OCOTP_CTRL[ERROR] will be set. It must be cleared by software before any new write , read or reload access can be issued. Subsequent reads to unlocked shadow locations will still work successfully however.

To read fuse words directly from correctly complete the following steps:

1. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write, read or reload must be completed before a read access can be requested.
2. Write the requested to HW_OCOTP_CTRL[ADDR].

6.3.3.1.3 Fuse and Shadow Register Writes

Shadow register bits can be overridden by software until the corresponding fuse lock bit for the region is set. When the lock shadow bit is set, the shadow registers for that lock region become write locked. The LOCK shadow register also has no shadow or fuse lock bits but it is always read only.

In order to avoid "rogue" code performing erroneous writes to OTP, a special unlocking sequence is required for writes to the fuse banks. To program fuse bank correctly complete the following steps:

1. Program HW_OCOTP_TIMING[STROBE_PROG] and fields with timing values to match the current frequency of the ipg_clk. OTP writes will work at maximum bus frequencies as long as the parameters are set correctly.
2. Check that HW_OCOTP_CTRL[BUSY] and HW_OCOTP_CTRL[ERROR] are clear. Overlapped accesses are not supported by the controller. Any pending write or reload must be completed before a write access can be requested.
3. Write the requested to HW_OCOTP_CTRL[ADDR] and program the unlock code into HW_OCOTP_CTRL[WR_UNLOCK]. This must be programmed for each write access. The lock code is documented in the register description. Both the unlock code and address can be written in the same operation.

It should be noted that write latencies to OTP are numbers of 10 micro-seconds. Write latencies is based on amount of bit filed which is 1. For example : program half fuse bits in one word need 10 us x 16.

For further details of OTP read/write operations see [eFUSE].

HW_OCOTP_CTRL[ERROR] will be set under the following conditions:

- A write is performed to a shadow register during a shadow reload (essentially, while HW_OCOTP_CTRL[RELOAD_SHADOWS] is set. In addition, the contents of the shadow register shall not be updated.
- A write is performed to a shadow register which has been locked.
- A read is performed to from a shadow register which has been read locked.
- A program is performed to a fuse which has been .
- A read is performed to from a fuse which has been read locked.

6.3.3.1.4 Write Postamble

Due to internal electrical characteristics of the OTP during writes, all OTP operations following a write must be separated by 2 us after the clearing of HW_OCOTP_CTRL_BUSY following the write. This guarantees programming voltages on-chip to reach a steady state when exiting a write sequence. This includes reads, shadow reloads, or other writes.

A recommended software sequence to meet the postamble requirements is as follows:

- Issue the write and poll for BUSY (as per Fuse Shadow Memory Footprint).
- After BUSY is clear, wait an additional 2 us.
- Perform the next OTP operation.

6.3.3.2 Fuse Shadow Memory Footprint

The OTP memory footprint shows in the following figure. The registers are grouped by lock region. Their names correspond to the PIO register and fusemap names.

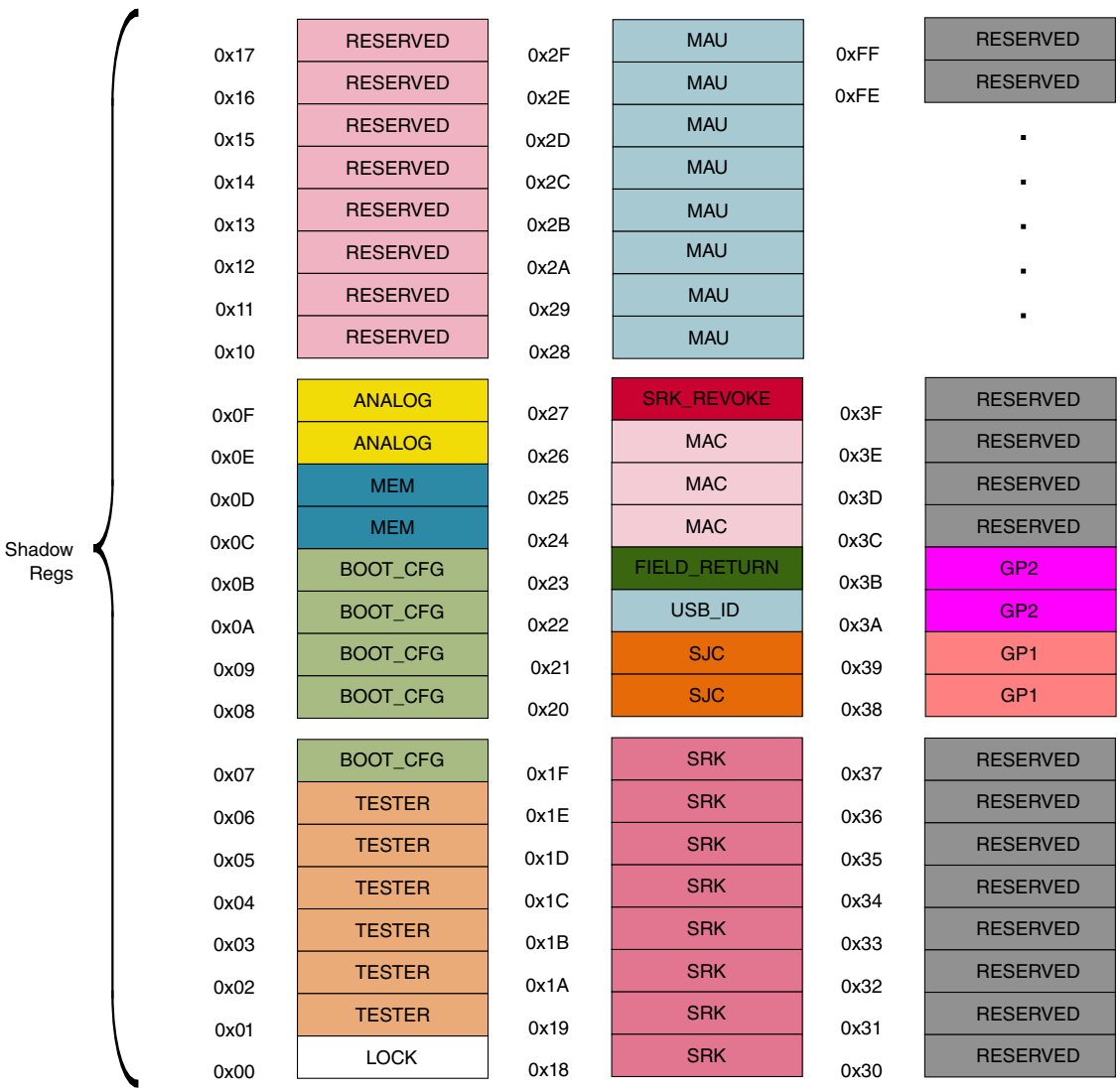


Figure 6-28. OTP Memory Footprint

6.3.3.3 OTP Read/Write Timing Parameters

There are timing fields contained in the HW_OCOTP_TIMING register that specify counter limit values, which are used to specify the signal timing.

Both two timing parameters are specified in ipg_clk cycles. Since the ipg_clk frequency can be set to a range of values, these parameters must be adjusted with the clock to yield the appropriate delay.

6.3.3.4 Hardware Visible Fuses

The hwv_fuse bus emanates from the OCOTP block and goes to various other blocks inside the chip. This bus is made up of the shadow register bits for .

Only a subset of these fuse bits are currently used by the hardware. The fuse bits are initially copied from the banks after reset is deasserted. When all fuse bits are loaded into their shadow registers, the OCOTP asserts the fuse_latched output signal.

The hwv_reg bus also comes from the OCOTP. Its source is the HW_OCOTP_SCS register. This register has 1 defined bit, the HAB_JDE bit, that is connected to the SJC block. The SCS bits are intended to be used as volatile fuse bits under software control. Additional bits will be defined as needed in future implementations.

The system-wide reset sequence must be coordinated by the system reset controller, so that the hwv_fuse and hwv_reg buses are stable and reflect the values of the fuses before they are used by the rest of the system.

6.3.3.5 Behavior During Reset

The OCOTP is always active. The shadow registers automatically load the appropriate OTP contents after reset is deasserted. During this load-time HW_OCOTP_CTRL_BUSY is set. The load time is similar to that of a "reload shadow" operation.

6.3.3.6 Secure JTAG control

The JTAG control fuses are used to allow or disallow JTAG access to secured resources.

Three JTAG security levels are envisioned, as shown in the table below.

Table 6-45. JTAG Security Level Control Bits

Security Mode	JTAG_SMODE	Description
No Debug	2'b11	The highest security level.
Secure JTAG	2'b01	Limit the JTAG access by using key based authentication mechanism.
JTAG Enable	2'b00	Low Security, all JTAG features are enabled.

6.3.4 Fuse Map

See the Fusemap chapter of this reference manual for more information.

6.3.5 OCOTP Memory Map/Register Definition

NOTE

When write/read unimplemented register address in ocotp_ctrl, ocotp_ctrl will not send error and read data will be 0.

OCOTP Hardware Register Format Summary

OCOTP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3035_0000	OTP Controller Control Register (OCOTP_HW_OCOTP_CTRL)	32	R/W	0000_0000h	6.3.5.1/ 887
3035_0004	OTP Controller Control Register (OCOTP_HW_OCOTP_CTRL_SET)	32	R/W	0000_0000h	6.3.5.1/ 887
3035_0008	OTP Controller Control Register (OCOTP_HW_OCOTP_CTRL_CLR)	32	R/W	0000_0000h	6.3.5.1/ 887
3035_000C	OTP Controller Control Register (OCOTP_HW_OCOTP_CTRL_TOG)	32	R/W	0000_0000h	6.3.5.1/ 887
3035_0010	OTP Controller Timing Register (OCOTP_HW_OCOTP_TIMING)	32	R/W	0400_0000h	6.3.5.2/ 888
3035_0020	OTP Controller Write Data Register (OCOTP_HW_OCOTP_DATA)	32	R/W	0000_0000h	6.3.5.3/ 889
3035_0030	OTP Controller Write Data Register (OCOTP_HW_OCOTP_READ_CTRL)	32	R/W	0000_0000h	6.3.5.4/ 890
3035_0040	OTP Controller Read Data Register (OCOTP_HW_OCOTP_READ_FUSE_DATA)	32	R/W	0000_0000h	6.3.5.5/ 891
3035_0050	Sticky bit Register (OCOTP_HW_OCOTP_SW_STICKY)	32	R/W	0000_0000h	6.3.5.6/ 892
3035_0060	Software Controllable Signals Register (OCOTP_HW_OCOTP_SCS)	32	R/W	0000_0000h	6.3.5.7/ 895
3035_0064	Software Controllable Signals Register (OCOTP_HW_OCOTP_SCS_SET)	32	R/W	0000_0000h	6.3.5.7/ 895
3035_0068	Software Controllable Signals Register (OCOTP_HW_OCOTP_SCS_CLR)	32	R/W	0000_0000h	6.3.5.7/ 895
3035_006C	Software Controllable Signals Register (OCOTP_HW_OCOTP_SCS_TOG)	32	R/W	0000_0000h	6.3.5.7/ 895

Table continues on the next page...

OCOTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3035_0090	OTP Controller Version Register (OCOTP_HW_OCOTP_VERSION)	32	R/W	0148_1299h	6.3.5.8/ 896
3035_0400	Value of OTP Bank0 Word0 (Lock controls) (OCOTP_HW_OCOTP_LOCK)	32	R/W	0000_0000h	6.3.5.9/ 897
3035_0410	Value of OTP Bank0 Word1 (Tester Info.) (OCOTP_HW_OCOTP_TESTER0)	32	R/W	0000_0000h	6.3.5.10/ 899
3035_0420	Value of OTP Bank0 Word2 (tester Info.) (OCOTP_HW_OCOTP_TESTER1)	32	R/W	0000_0000h	6.3.5.11/ 899
3035_0430	Value of OTP Bank0 Word3 (Tester Info.) (OCOTP_HW_OCOTP_TESTER2)	32	R/W	0000_0000h	6.3.5.12/ 900
3035_0440	Value of OTP Bank1 Word0 (Tester Info.) (OCOTP_HW_OCOTP_TESTER3)	32	R/W	0000_0000h	6.3.5.13/ 900
3035_0450	Value of OTP Bank1 Word1 (Tester Info.) (OCOTP_HW_OCOTP_TESTER4)	32	R/W	0000_0000h	6.3.5.14/ 901
3035_0460	Value of OTP Bank1 Word2 (Tester Info.) (OCOTP_HW_OCOTP_TESTER5)	32	R/W	0000_0000h	6.3.5.15/ 901
3035_0470	Value of OTP Bank1 Word3 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG0)	32	R/W	0000_0000h	6.3.5.16/ 902
3035_0480	Value of OTP Bank2 Word0 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG1)	32	R/W	0000_0000h	6.3.5.17/ 902
3035_0490	Value of OTP Bank2 Word1 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG2)	32	R/W	0000_0000h	6.3.5.18/ 903
3035_04A0	Value of OTP Bank2 Word2 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG3)	32	R/W	0000_0000h	6.3.5.19/ 903
3035_04B0	Value of OTP Bank2 Word3 (BOOT Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG4)	32	R/W	0000_0000h	6.3.5.20/ 904
3035_04C0	Value of OTP Bank3 Word0 (Memory Related Info.) (OCOTP_HW_OCOTP_MEM_TRIM0)	32	R/W	0000_0000h	6.3.5.21/ 904
3035_04D0	Value of OTP Bank3 Word1 (Memory Related Info.) (OCOTP_HW_OCOTP_MEM_TRIM1)	32	R/W	0000_0000h	6.3.5.22/ 905
3035_04E0	Value of OTP Bank3 Word2 (Analog Info.) (OCOTP_HW_OCOTP_ANA0)	32	R/W	0000_0000h	6.3.5.23/ 905
3035_04F0	Value of OTP Bank3 Word3 (Analog Info.) (OCOTP_HW_OCOTP_ANA1)	32	R/W	0000_0000h	6.3.5.24/ 906
3035_0580	Shadow Register for OTP Bank6 Word0 (SRK Hash) (OCOTP_HW_OCOTP_SRK0)	32	R/W	0000_0000h	6.3.5.25/ 906
3035_0590	Shadow Register for OTP Bank6 Word1 (SRK Hash) (OCOTP_HW_OCOTP_SRK1)	32	R/W	0000_0000h	6.3.5.26/ 907
3035_05A0	Shadow Register for OTP Bank6 Word2 (SRK Hash) (OCOTP_HW_OCOTP_SRK2)	32	R/W	0000_0000h	6.3.5.27/ 907
3035_05B0	Shadow Register for OTP Bank6 Word3 (SRK Hash) (OCOTP_HW_OCOTP_SRK3)	32	R/W	0000_0000h	6.3.5.28/ 908
3035_05C0	Shadow Register for OTP Bank7 Word0 (SRK Hash) (OCOTP_HW_OCOTP_SRK4)	32	R/W	0000_0000h	6.3.5.29/ 908

Table continues on the next page...

OCOTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3035_05D0	Shadow Register for OTP Bank7 Word1 (SRK Hash) (OCOTP_HW_OCOTP_SRK5)	32	R/W	0000_0000h	6.3.5.30/ 909
3035_05E0	Shadow Register for OTP Bank7 Word2 (SRK Hash) (OCOTP_HW_OCOTP_SRK6)	32	R/W	0000_0000h	6.3.5.31/ 909
3035_05F0	Shadow Register for OTP Bank7 Word3 (SRK Hash) (OCOTP_HW_OCOTP_SRK7)	32	R/W	0000_0000h	6.3.5.32/ 910
3035_0600	Value of OTP Bank8 Word0 (Secure JTAG Response Field) (OCOTP_HW_OCOTP_SJC_RESP0)	32	R/W	0000_0000h	6.3.5.33/ 910
3035_0610	Value of OTP Bank8 Word1 (Secure JTAG Response Field) (OCOTP_HW_OCOTP_SJC_RESP1)	32	R/W	0000_0000h	6.3.5.34/ 911
3035_0620	Value of OTP Bank8 Word2 (USB ID info) (OCOTP_HW_OCOTP_USB_ID)	32	R/W	0000_0000h	6.3.5.35/ 911
3035_0630	Value of OTP Bank8 Word3 (Field Return) (OCOTP_HW_OCOTP_FIELD_RETURN)	32	R/W	0000_0000h	6.3.5.36/ 912
3035_0640	Value of OTP Bank9 Word0 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR0)	32	R/W	0000_0000h	6.3.5.37/ 912
3035_0650	Value of OTP Bank9 Word1 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR1)	32	R/W	0000_0000h	6.3.5.38/ 913
3035_0660	Value of OTP Bank9 Word2 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR2)	32	R/W	0000_0000h	6.3.5.39/ 913
3035_0670	Value of OTP Bank9 Word3 (SRK Revoke) (OCOTP_HW_OCOTP_SRK_REVOKE)	32	R/W	0000_0000h	6.3.5.40/ 914
3035_0680	Shadow Register for OTP Bank10 Word0 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY0)	32	R/W	0000_0000h	6.3.5.41/ 914
3035_0690	Shadow Register for OTP Bank10 Word1 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY1)	32	R/W	0000_0000h	6.3.5.42/ 915
3035_06A0	Shadow Register for OTP Bank10 Word2 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY2)	32	R/W	0000_0000h	6.3.5.43/ 915
3035_06B0	Shadow Register for OTP Bank10 Word3 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY3)	32	R/W	0000_0000h	6.3.5.44/ 916
3035_06C0	Shadow Register for OTP Bank11 Word0 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY4)	32	R/W	0000_0000h	6.3.5.45/ 916
3035_06D0	Shadow Register for OTP Bank11 Word1 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY5)	32	R/W	0000_0000h	6.3.5.46/ 917
3035_06E0	Shadow Register for OTP Bank11 Word2 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY6)	32	R/W	0000_0000h	6.3.5.47/ 917
3035_06F0	Shadow Register for OTP Bank11 Word3 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY7)	32	R/W	0000_0000h	6.3.5.48/ 918
3035_0780	Value of OTP Bank14 Word0 () (OCOTP_HW_OCOTP_GP10)	32	R/W	0000_0000h	6.3.5.49/ 918
3035_0790	Value of OTP Bank14 Word1 () (OCOTP_HW_OCOTP_GP11)	32	R/W	0000_0000h	6.3.5.50/ 919
3035_07A0	Value of OTP Bank14 Word2 () (OCOTP_HW_OCOTP_GP20)	32	R/W	0000_0000h	6.3.5.51/ 919

Table continues on the next page...

OCOTP memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3035_07B0	Value of OTP Bank14 Word3 () (OCOTP_HW_OCOTP_GP21)	32	R/W	0000_0000h	6.3.5.52/ 919

6.3.5.1 OTP Controller Control Register (OCOTP_HW_OCOTP_CTRL*n*)

The OCOTP Control and Status Register provides the necessary software interface for performing read and write operations to the On-Chip OTP (One-Time Programmable ROM). The control fields such as WR_UNLOCK, ADDR and BUSY/ERROR may be used in conjuction with the HW_OCOTP_DATA register to perform write operations. Read operations to the On-Chip OTP are involving ADDR, BUSY/ERROR bit field and HW_OCOTP_READ_CTRL register. Read value is saved in HW_OCOTP_READ_FUSE_DATA register.

Address: 3035_0000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WR_UNLOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					RELOAD_SHADOWS	ERROR	BUSY	ADDR							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_CTRL*n* field descriptions

Field	Description
31–16 WR_UNLOCK	Write 0x3E77 to enable OTP write accesses. NOTE: This register must be unlocked on a write-by-write basis (a write is initiated when HW_OCOTP_DATA is written), so the UNLOCK bitfield must contain the correct key value during all writes to HW_OCOTP_DATA, otherwise a write shall not be initiated. This field is automatically cleared after a successful write completion (clearing of BUSY).
15–11 -	This field is reserved. Reserved
10 RELOAD_ SHADOWS	Set to force re-loading the shadow registers (HW/SW capability and LOCK). This operation will automatically set BUSY. Once the shadow registers have been re-loaded, BUSY and RELOAD_SHADOWS are automatically cleared by the controller.
9 ERROR	Set by the controller when an access to a locked region(OTP or shadow register) is requested. Must be cleared before any further access can be performed. This bit can only be set by the controller. This bit is also set if the Pin interface is active and software requests an access to the OTP. In this instance, the ERROR bit cannot be cleared until the Pin interface access has completed. Reset this bit by writing a one to the SCT clear address space and not by a general write.
8 BUSY	OTP controller status bit. When active, no new write access or read access to OTP(including RELOAD_SHADOWS) can be performed. Cleared by controller when access complete. After reset (or after setting RELOAD_SHADOWS), this bit is set by the controller until the HW/SW and LOCK registers are successfully copied, after which time it is automatically cleared by the controller.
ADDR	OTP write and read access address register. Specifies one of 128 word address locations (0x00 - 0x7f). If a valid access is accepted by the controller, the controller makes an internal copy of this value. This internal copy will not update until the access is complete.

6.3.5.2 OTP Controller Timing Register (OCOTP_HW_OCOTP_TIMING)

This register specifies timing parameters for programming and reading the OCOTP fuse array.

Address: 3035_0000h base + 10h offset = 3035_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSRVD0				WAIT						STROBE_READ						RELAX				STROBE_PROG											
W																																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TIMING field descriptions

Field	Description
31–28 RSRVD0	These bits always read back zero.
27–22 WAIT	This count value specifies time interval between auto read and write access in one time program. It is given in number of ipg_clk periods.

Table continues on the next page...

OCOTP_HW_OCOTP_TIMING field descriptions (continued)

Field	Description
21–16 STROBE_READ	This count value specifies the strobe period in one time read OTP. Trd = ((STROBE_READ+1)- 2*(RELAX +1)) /ipg_clk_freq. It is given in number of ipg_clk periods.
15–12 RELAX	This count value specifies the time to add to all default timing parameters other than the Tpgm and Trd. It is given in number of ipg_clk periods.
STROBE_PROG	This count value specifies the strobe period in one time write OTP. Tpgm = ((STROBE_PROG+1)- 2*(RELAX+1)) /ipg_clk_freq. It is given in number of ipg_clk periods.

6.3.5.3 OTP Controller Write Data Register (OCOTP_HW_OCOTP_DATA)

This register is used in conjuction with HW_OCOTP_CTRL to perform one-time writes to the OTP.

Address: 3035_0000h base + 20h offset = 3035_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

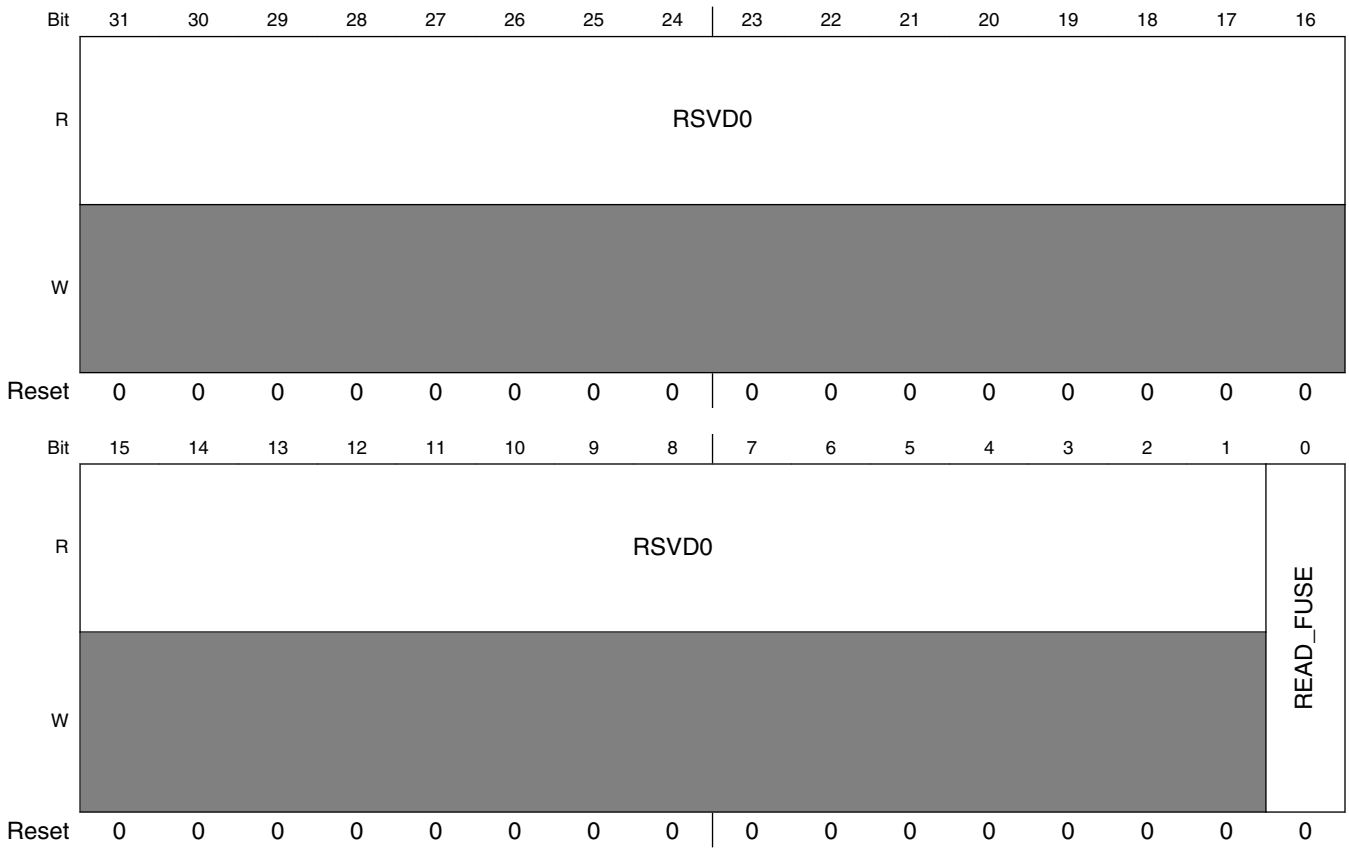
OCOTP_HW_OCOTP_DATA field descriptions

Field	Description
DATA	Used to initiate a write to OTP.

6.3.5.4 OTP Controller Write Data Register (OCOTP_HW_OCOTP_READ_CTRL)

This register is used in conjuction with HW_OCOTP_CTRL to perform one time read to the OTP.

Address: 3035_0000h base + 30h offset = 3035_0030h



OCOTP_HW_OCOTP_READ_CTRL field descriptions

Field	Description
31–1 RSVD0	Reserved
0 READ_FUSE	Used to initiate a read to OTP.

6.3.5.5 OTP Controller Read Data Register (OCOTP_HW_OCOTP_READ_FUSE_DATA)

The data read from OTP

Address: 3035_0000h base + 40h offset = 3035_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

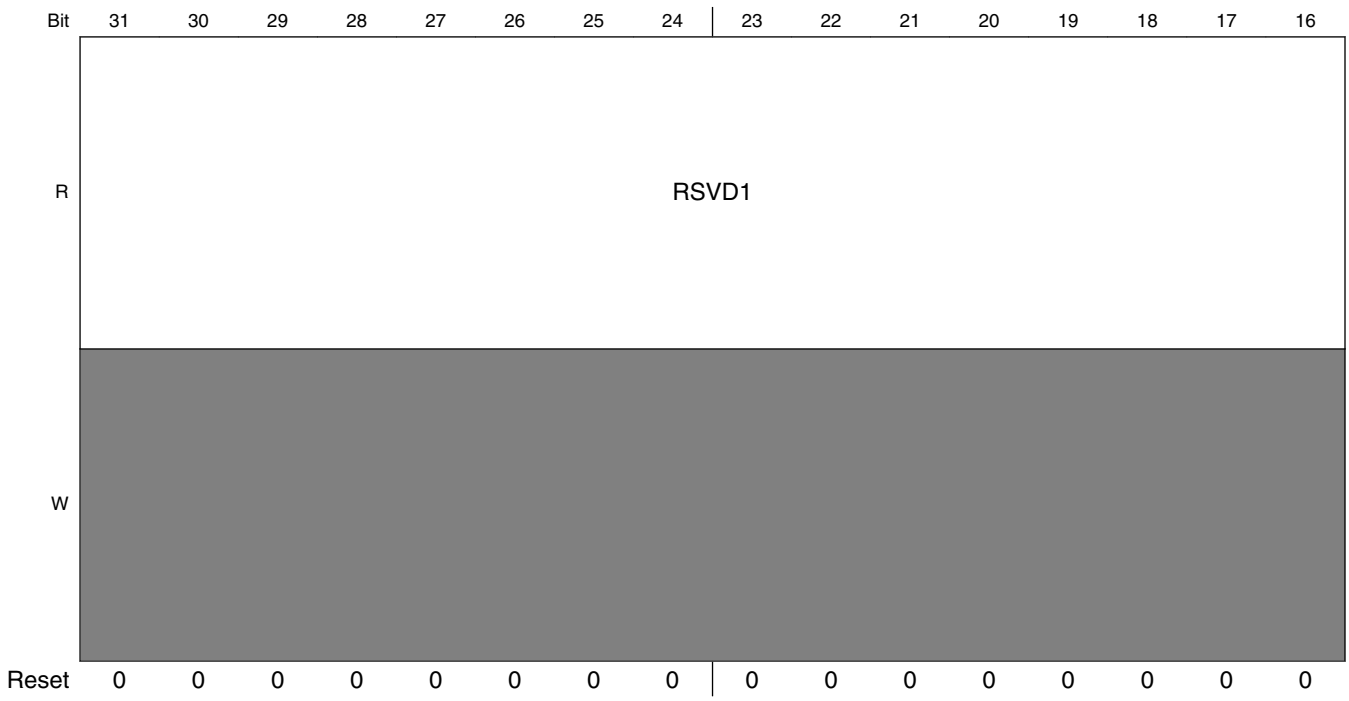
OCOTP_HW_OCOTP_READ_FUSE_DATA field descriptions

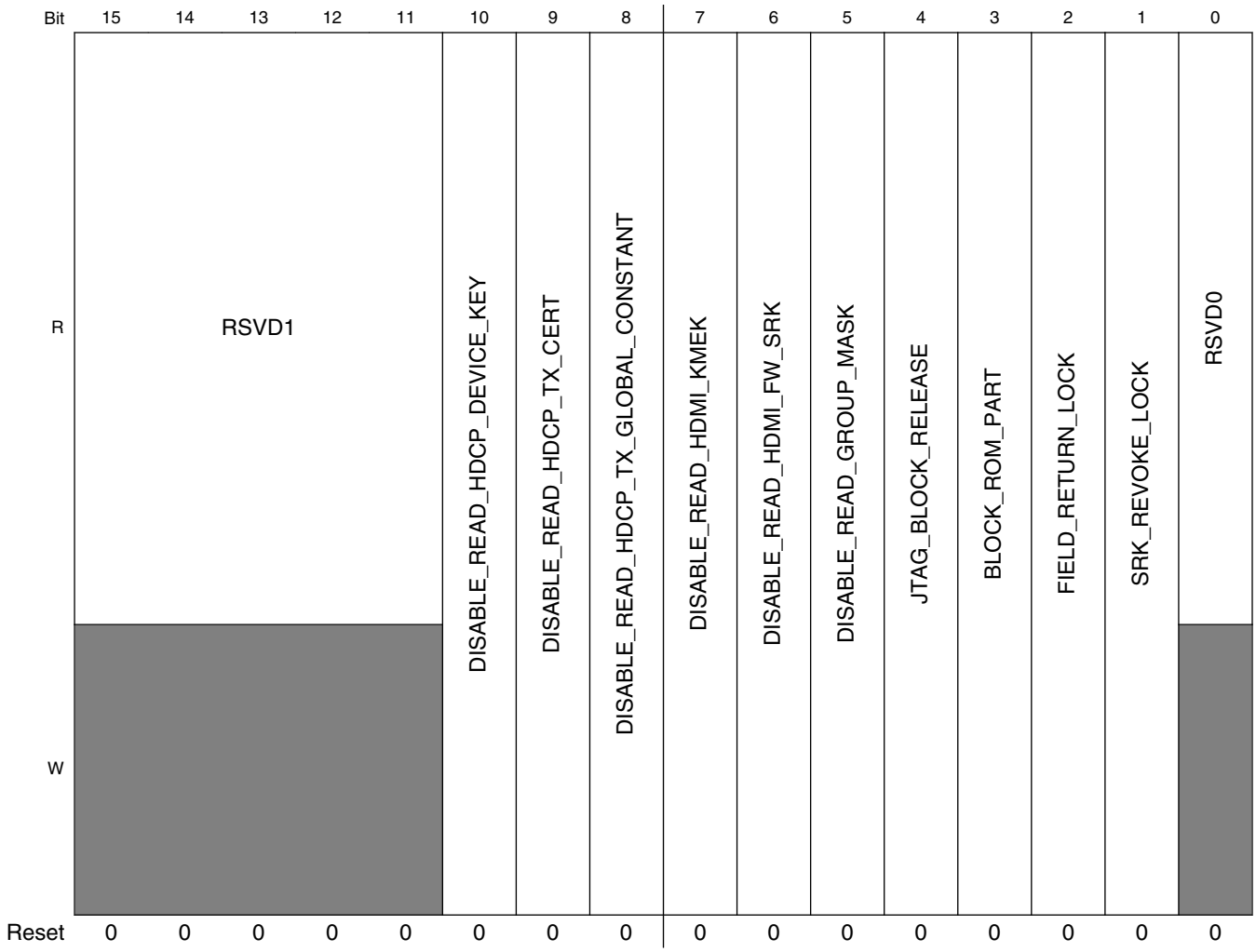
Field	Description
DATA	The data read from OTP

6.3.5.6 Sticky bit Register (OCOTP_HW_OCOTP_SW_STICKY)

Some sticky bits are used by SW to lock some fuse area , shadow registers and other features.

Address: 3035_0000h base + 50h offset = 3035_0050h





OCOTP_HW_OCOTP_SW_STICKY field descriptions

Field	Description
31–11 RSVD1	Reserved
10 DISABLE_ READ_HDCP_ DEVICE_KEY	Shadow register write and OTP write lock for HDCP_DEVICE_HDCP region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
9 DISABLE_ READ_HDCP_ TX_CERT	Shadow register write and OTP write lock for HDCP_TX_CERT region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
8 DISABLE_ READ_HDCP_ TX_GLOBAL_ CONSTANT	Shadow register write and OTP write lock for HDCP_TX_GLOBAL_CONSTANT region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.

Table continues on the next page...

OCOTP_HW_OCOTP_SW_STICKY field descriptions (continued)

Field	Description
7 DISABLE_ READ_HDMI_ KMEK	Shadow register write and OTP write lock for HDMI_KMEK region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
6 DISABLE_ READ_HDMI_ FW_SRK	Shadow register write and OTP write lock for HDMI_FW_SRK region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
5 DISABLE_ READ_GROUP_ MASK	Shadow register write and OTP write lock for GROUP_MASK region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
4 JTAG_BLOCK_ RELEASE	Set by Arm during Boot after DTCP is initialized and before test mode entry. * 0 (Default) - JTAG is blocked (subject to other conditions).* 1 - JTAG block is released (subject to other controls). Once this bit is set, it is always high unless a POR is issued.
3 BLOCK_ROM_ PART	Set by Arm during Boot after DTCP is initialized and before test mode entry, if ROM_PART_LOCK=1.* 0 (Default) - Secret part of Boot ROM is not hidden (subject to other conditions).* 1 - Secret part of Boot ROM is hidden.Once this bit is set, it is always high unless a POR is issued.
2 FIELD_ RETURN_LOCK	Shadow register write and OTP write lock for FIELD_RETURN region. When set, the writing of this region's shadow register and OTP fuse word are blocked.Once this bit is set, it is always high unless a POR is issued.
1 SRK_REVOKE_ LOCK	Shadow register write and OTP write lock for SRK_REVOKE region. When set, the writing of this region's shadow register and OTP fuse word are blocked. Once this bit is set, it is always high unless a POR is issued.
0 RSVD0	Reserved

6.3.5.7 Software Controllable Signals Register (OCOTP_HW_OCOTP_SCS*n*)

This register holds volatile configuration values that can be set and locked by trusted software. All values are returned to their defualt values after POR.

Address: 3035_0000h base + 60h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LOCK	SPARE														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE															HAB_JDE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_SCS*n* field descriptions

Field	Description
31 LOCK	When set, all of the bits in this register are locked and can not be changed through SW programming. This bit is only reset after a POR is issued.
30–1 SPARE	Unallocated read/write bits for implementation specific software use.
0 HAB_JDE	HAB JTAG Debug Enable. This bit is used by the HAB to enable JTAG debugging, assuming that a properlay signed command to do so is found and validated by the HAB. The HAB must lock the register before passing control to the OS whether or not JTAG debugging has been enabled. Once JTAG is enabled by this bit, it can not be disabled unless the system is reset by POR. 0: JTAG debugging is not enabled by the HAB (it may still be enabled by other mechanisms). 1: JTAG debugging is enabled by the HAB (though this signal may be gated off).

6.3.5.8 OTP Controller Version Register (OCOTP_HW_OCOTP_VERSION)

This register indicates the RTL version in use.

Address: 3035_0000h base + 90h offset = 3035_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	0	1	0	1	0	0	1	0	0	0	0	0	0	1	0	0	1	0	1	0	0	1	1	0	0	1

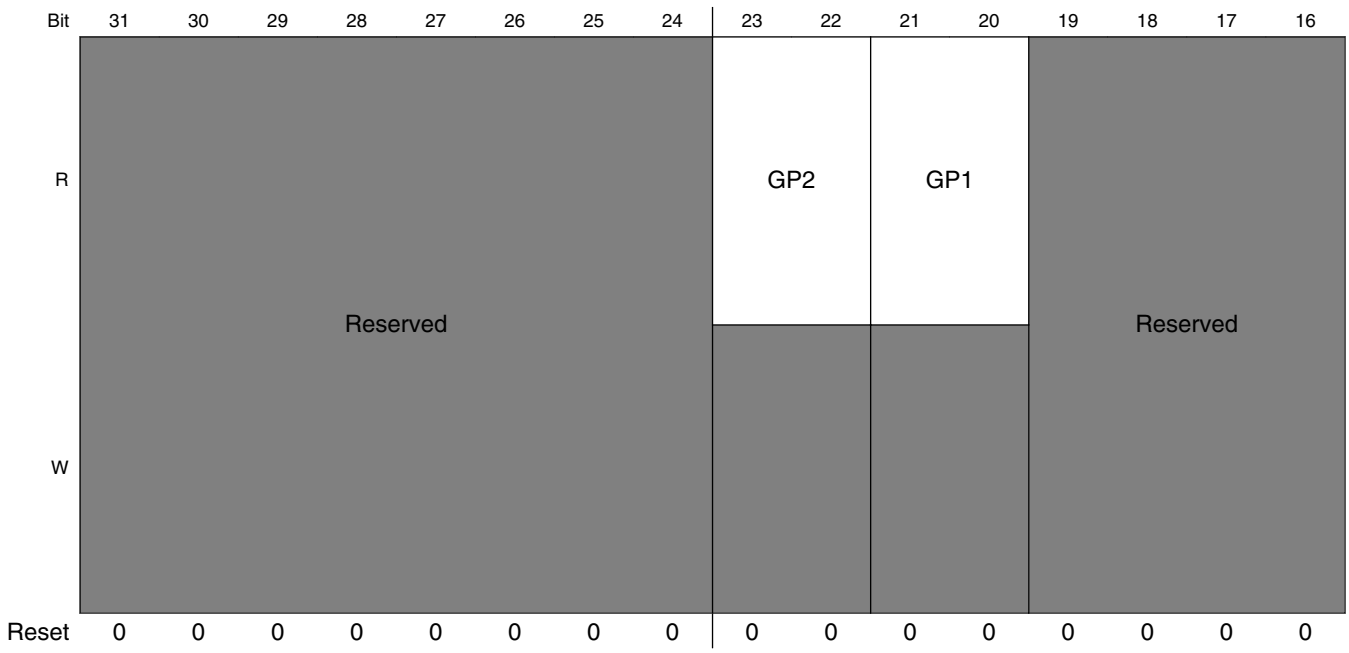
OCOTP_HW_OCOTP_VERSION field descriptions

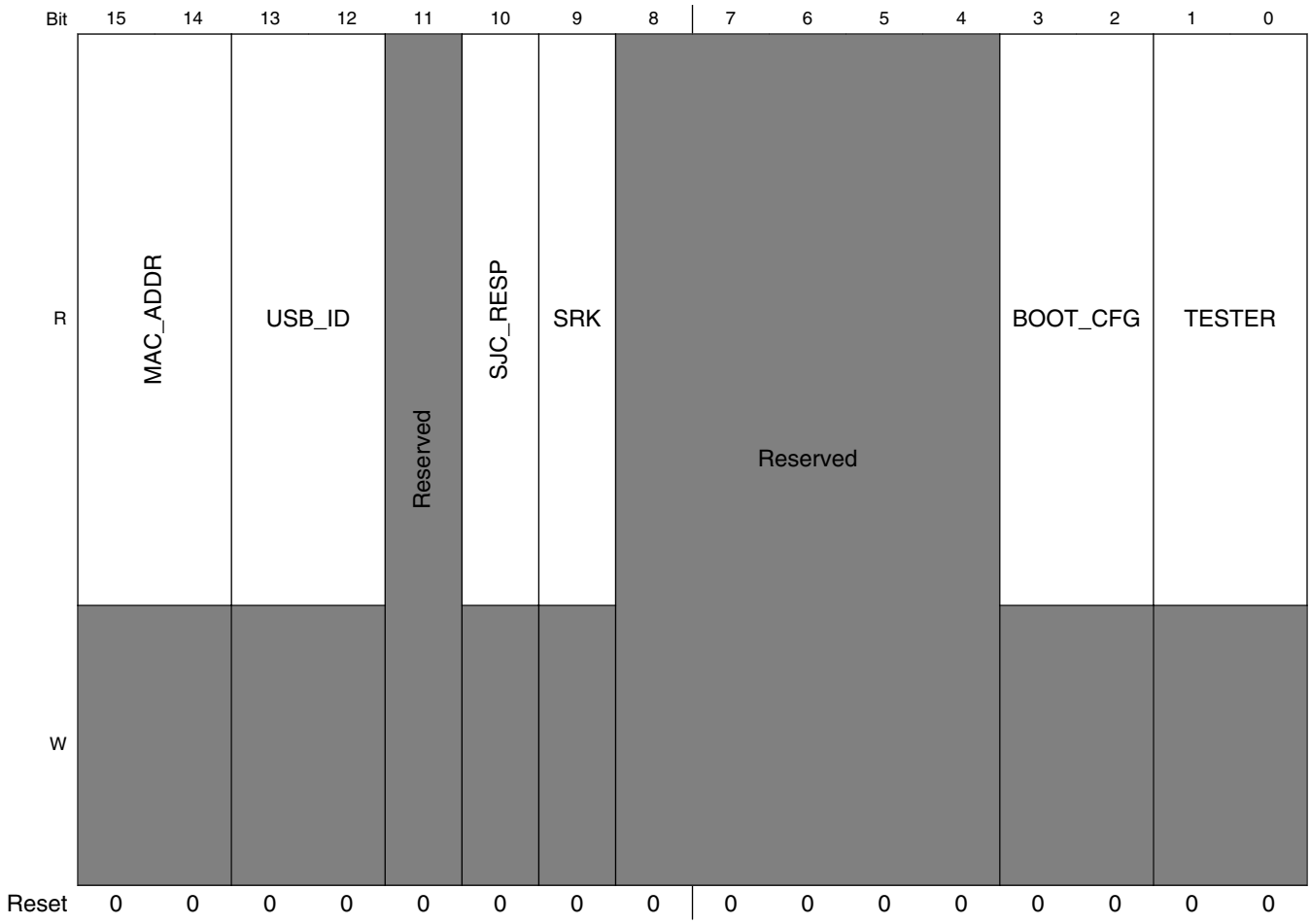
Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

6.3.5.9 Value of OTP Bank0 Word0 (Lock controls) (OCOTP_HW_OCOTP_LOCK)

Shadowed memory mapped access to OTP Bank 0, word 0.

Address: 3035_0000h base + 400h offset = 3035_0400h





OCOTP_HW_OCOTP_LOCK field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–22 GP2	Status of shadow register and OTP write lock for gp2 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
21–20 GP1	Status of shadow register and OTP write lock for gp1 region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
19–16 -	This field is reserved. Reserved
15–14 MAC_ADDR	Status of shadow register and OTP write lock for mac_addr region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
13–12 USB_ID	Status of shadow register and OTP write lock for usb_id region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
11 -	This field is reserved. Reserved

Table continues on the next page...

OCOTP_HW_OCOTP_LOCK field descriptions (continued)

Field	Description
10 SJC_RESP	Status of shadow register read and write, OTP read and write lock for sjc_resp region. When set, the writing of this region's shadow register and OTP fuse word are blocked. The read of this region's shadow register and OTP fuse word are also blocked.
9 SRK	Status of shadow register and OTP write lock for srk region. When set, the writing of this region's shadow register and OTP fuse word are blocked.
8–4 -	This field is reserved. Reserved
3–2 BOOT_CFG	Status of shadow register and OTP write lock for boot_cfg region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.
TESTER	Status of shadow register and OTP write lock for tester region. When bit 1 is set, the writing of this region's shadow register is blocked. When bit 0 is set, the writing of this region's OTP fuse word is blocked.

6.3.5.10 Value of OTP Bank0 Word1 (Tester Info.) (OCOTP_HW_OCOTP_TESTER0)

Shadowed memory mapped access to OTP Bank 0, word 1.

Address: 3035_0000h base + 410h offset = 3035_0410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER0 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 1. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.11 Value of OTP Bank0 Word2 (tester Info.) (OCOTP_HW_OCOTP_TESTER1)

shadowed memory mapped access to OTP Bank 0, word 2.

Address: 3035_0000h base + 420h offset = 3035_0420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER1 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 2. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.12 Value of OTP Bank0 Word3 (Tester Info.) (OCOTP_HW_OCOTP_TESTER2)

Shadowed memory mapped access to OTP Bank 0, word 3.

Address: 3035_0000h base + 430h offset = 3035_0430h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER2 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 0, word 3. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.13 Value of OTP Bank1 Word0 (Tester Info.) (OCOTP_HW_OCOTP_TESTER3)

Non-shadowed memory mapped access to OTP Bank 1, word 0.

Address: 3035_0000h base + 440h offset = 3035_0440h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER3 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 1, word 0. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.14 Value of OTP Bank1 Word1 (Tester Info.) (OCOTP_HW_OCOTP_TESTER4)

Shadowed memory mapped access to OTP Bank 1, word 1.

Address: 3035_0000h base + 450h offset = 3035_0450h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER4 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 1, word 1. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.15 Value of OTP Bank1 Word2 (Tester Info.) (OCOTP_HW_OCOTP_TESTER5)

Shadowed memory mapped access to OTP Bank 1, word 2.

Address: 3035_0000h base + 460h offset = 3035_0460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_TESTER5 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 1, word 2. These bits become read-only after the HW_OCOTP_LOCK_TESTER[1] bit is set.

6.3.5.16 Value of OTP Bank1 Word3 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG0)

Shadowed memory mapped access to OTP Bank 1, word 3.

Address: 3035_0000h base + 470h offset = 3035_0470h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_BOOT_CFG0 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 1, word 3. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.17 Value of OTP Bank2 Word0 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG1)

Shadowed memory mapped access to OTP bank 2, word 0.

Address: 3035_0000h base + 480h offset = 3035_0480h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_BOOT_CFG1 field descriptions

Field	Description
BITS	Reflects value of OTP bank 2, word 0. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.18 Value of OTP Bank2 Word1 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG2)

Shadowed memory mapped access to OTP bank 2, word 1.

Address: 3035_0000h base + 490h offset = 3035_0490h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_BOOT_CFG2 field descriptions

Field	Description
BITS	Reflects value of OTP bank 2, word 1. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.19 Value of OTP Bank2 Word2 (Boot Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG3)

Shadowed memory mapped access to OTP bank 2, word 2.

Address: 3035_0000h base + 4A0h offset = 3035_04A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_BOOT_CFG3 field descriptions

Field	Description
BITS	Reflects value of OTP bank 2, word 2. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.20 Value of OTP Bank2 Word3 (BOOT Configuration Info.) (OCOTP_HW_OCOTP_BOOT_CFG4)

Shadowed memory mapped access to OTP bank 2, word 3.

Address: 3035_0000h base + 4B0h offset = 3035_04B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_BOOT_CFG4 field descriptions

Field	Description
BITS	Reflects value of OTP bank 2, word 3. These bits become read-only after the HW_OCOTP_LOCK_BOOT_CFG[1] bit is set.

6.3.5.21 Value of OTP Bank3 Word0 (Memory Related Info.) (OCOTP_HW_OCOTP_MEM_TRIM0)

Shadowed memory mapped access to OTP bank 3, word 0.

Address: 3035_0000h base + 4C0h offset = 3035_04C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MEM_TRIM0 field descriptions

Field	Description
BITS	Reflects value of OTP bank 3, word 0. These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

6.3.5.22 Value of OTP Bank3 Word1 (Memory Related Info.) (OCOTP_HW_OCOTP_MEM_TRIM1)

Shadowed memory mapped access to OTP bank 3, word 1.

Address: 3035_0000h base + 4D0h offset = 3035_04D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MEM_TRIM1 field descriptions

Field	Description
BITS	Reflects value of OTP bank 3, word 1. These bits become read-only after the HW_OCOTP_LOCK_MEM_TRIM[1] bit is set.

6.3.5.23 Value of OTP Bank3 Word2 (Analog Info.) (OCOTP_HW_OCOTP_ANA0)

Shadowed memory mapped access to OTP bank 3, word 2.

Address: 3035_0000h base + 4E0h offset = 3035_04E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_ANA0 field descriptions

Field	Description
BITS	Reflects value of OTP bank 3, word 2. These bits become read-only after the HW_OCOTP_LOCK_ANALOG[1] bit is set.

6.3.5.24 Value of OTP Bank3 Word3 (Analog Info.) (OCOTP_HW_OCOTP_ANA1)

Shadowed memory mapped access to OTP bank 3, word 3.

Address: 3035_0000h base + 4F0h offset = 3035_04F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_ANA1 field descriptions

Field	Description
BITS	Reflects value of OTP bank 3, word 3. These bits become read-only after the HW_OCOTP_LOCK_ANALOG[1] bit is set.

6.3.5.25 Shadow Register for OTP Bank6 Word0 (SRK Hash) (OCOTP_HW_OCOTP_SRK0)

Shadowed memory mapped access to OTP Bank 6, word 0.

Address: 3035_0000h base + 580h offset = 3035_0580h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

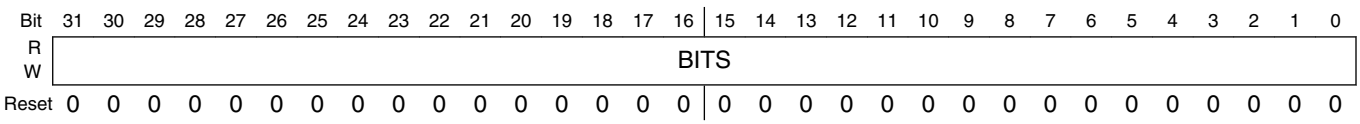
OCOTP_HW_OCOTP_SRK0 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word0 (Copy of OTP Bank 6, word 0). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.26 Shadow Register for OTP Bank6 Word1 (SRK Hash) (OCOTP_HW_OCOTP_SRK1)

Shadowed memory mapped access to OTP Bank 6, word 1.

Address: 3035_0000h base + 590h offset = 3035_0590h



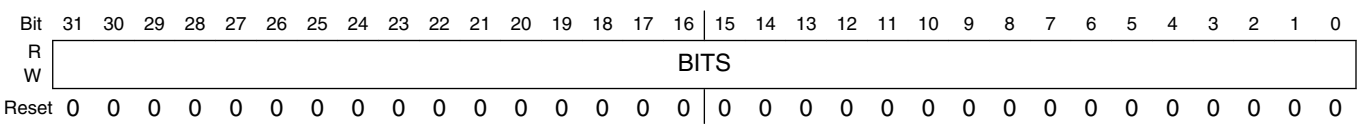
OCOTP_HW_OCOTP_SRK1 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word1 (Copy of OTP Bank 6, word 1). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.27 Shadow Register for OTP Bank6 Word2 (SRK Hash) (OCOTP_HW_OCOTP_SRK2)

Shadowed memory mapped access to OTP Bank 6, word 2.

Address: 3035_0000h base + 5A0h offset = 3035_05A0h



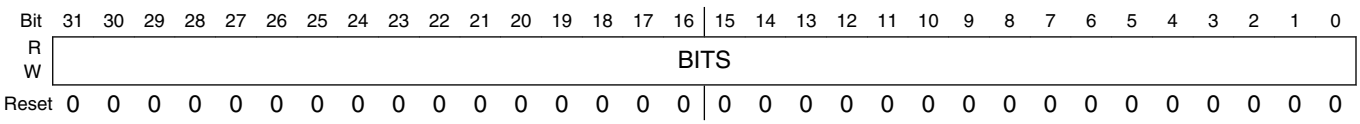
OCOTP_HW_OCOTP_SRK2 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word2 (Copy of OTP Bank 6, word 2). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.28 Shadow Register for OTP Bank6 Word3 (SRK Hash) (OCOTP_HW_OCOTP_SRK3)

Shadowed memory mapped access to OTP Bank 6, word 3.

Address: 3035_0000h base + 5B0h offset = 3035_05B0h



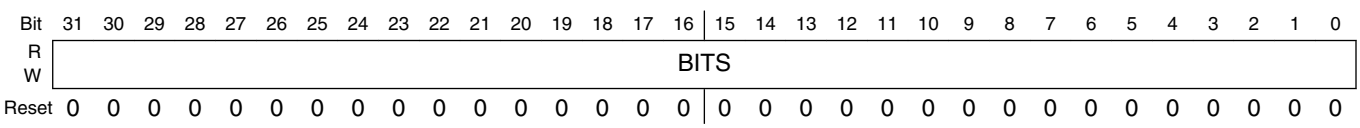
OCOTP_HW_OCOTP_SRK3 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word3 (Copy of OTP Bank 6, word 3). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.29 Shadow Register for OTP Bank7 Word0 (SRK Hash) (OCOTP_HW_OCOTP_SRK4)

Shadowed memory mapped access to OTP Bank 7, word 0.

Address: 3035_0000h base + 5C0h offset = 3035_05C0h



OCOTP_HW_OCOTP_SRK4 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word4 (Copy of OTP Bank 7, word 0). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.30 Shadow Register for OTP Bank7 Word1 (SRK Hash) (OCOTP_HW_OCOTP_SRK5)

Shadowed memory mapped access to OTP Bank 7, word 1.

Address: 3035_0000h base + 5D0h offset = 3035_05D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_SRK5 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word5 (Copy of OTP Bank 7, word 1). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.31 Shadow Register for OTP Bank7 Word2 (SRK Hash) (OCOTP_HW_OCOTP_SRK6)

Shadowed memory mapped access to OTP Bank 7, word 2.

Address: 3035_0000h base + 5E0h offset = 3035_05E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

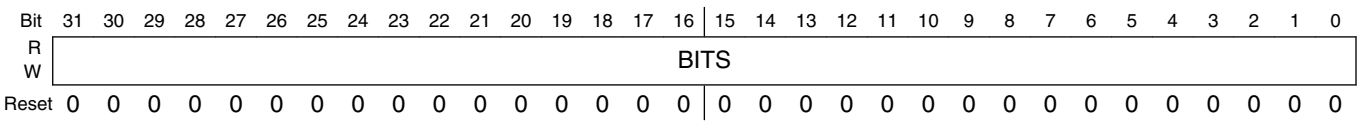
OCOTP_HW_OCOTP_SRK6 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word6 (Copy of OTP Bank 7, word 2). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.32 Shadow Register for OTP Bank7 Word3 (SRK Hash) (OCOTP_HW_OCOTP_SRK7)

Shadowed memory mapped access to OTP Bank 7, word 3.

Address: 3035_0000h base + 5F0h offset = 3035_05F0h



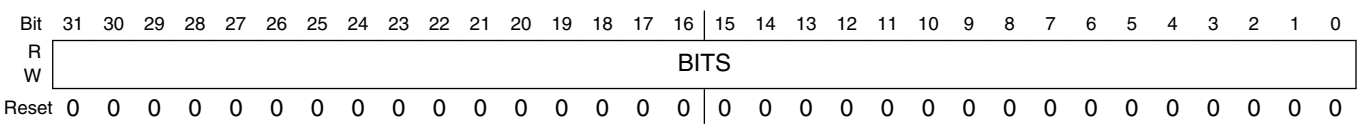
OCOTP_HW_OCOTP_SRK7 field descriptions

Field	Description
BITS	Shadow register for the hash of the Super Root Key word7 (Copy of OTP Bank 7, word 3). These bits become read-only after the HW_OCOTP_LOCK_SRK bit is set.

6.3.5.33 Value of OTP Bank8 Word0 (Secure JTAG Response Field) (OCOTP_HW_OCOTP_SJC_RESP0)

Shadowed memory mapped access to OTP Bank 8, word 0.

Address: 3035_0000h base + 600h offset = 3035_0600h



OCOTP_HW_OCOTP_SJC_RESP0 field descriptions

Field	Description
BITS	Shadow register for the SJC_RESP Key word0 (Copy of OTP Bank 8, word 0). These bits can be not read and wrotten after the HW_OCOTP_LOCK_SJC_RESP bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.34 Value of OTP Bank8 Word1 (Secure JTAG Response Field) (OCOTP_HW_OCOTP_SJC_RESP1)

Shadowed memory mapped access to OTP Bank 8, word 1.

Address: 3035_0000h base + 610h offset = 3035_0610h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_SJC_RESP1 field descriptions

Field	Description
BITS	Shadow register for the SJC_RESP Key word1 (Copy of OTP Bank 8, word 1). These bits can be not read and wrotten after the HW_OCOTP_LOCK_SJC_RESP bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.35 Value of OTP Bank8 Word2 (USB ID info) (OCOTP_HW_OCOTP_USB_ID)

Shadowed memory mapped access to OTP Bank 8, word 2.

Address: 3035_0000h base + 620h offset = 3035_0620h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_USB_ID field descriptions

Field	Description
BITS	Reflects value of OTP Bank 8, word 2.

6.3.5.36 Value of OTP Bank8 Word3 (Field Return) (OCOTP_HW_OCOTP_FIELD_RETURN)

Shadowed memory mapped access to OTP Bank 8, word 3.

Address: 3035_0000h base + 630h offset = 3035_0630h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_FIELD_RETURN field descriptions

Field	Description
BITS	Reflects value of OTP Bank 8, word 3.

6.3.5.37 Value of OTP Bank9 Word0 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR0)

Shadowed memory mapped access to OTP Bank 9, word 0.

Address: 3035_0000h base + 640h offset = 3035_0640h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAC_ADDR0 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 9, word 0.

6.3.5.38 Value of OTP Bank9 Word1 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR1)

Shadowed memory mapped access to OTP Bank 9, word 1.

Address: 3035_0000h base + 650h offset = 3035_0650h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAC_ADDR1 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 9, word 1.

6.3.5.39 Value of OTP Bank9 Word2 (MAC Address) (OCOTP_HW_OCOTP_MAC_ADDR2)

Shadowed memory mapped access to OTP Bank 9, word 2.

Address: 3035_0000h base + 660h offset = 3035_0660h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAC_ADDR2 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 9, word 2.

6.3.5.40 Value of OTP Bank9 Word3 (SRK Revoke) (OCOTP_HW_OCOTP_SRK_REVOKE)

Shadowed memory mapped access to OTP Bank 9, word 3.

Address: 3035_0000h base + 670h offset = 3035_0670h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_SRK_REVOKE field descriptions

Field	Description
BITS	Reflects value of OTP Bank 9, word 3.

6.3.5.41 Shadow Register for OTP Bank10 Word0 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY0)

Shadowed memory mapped access to OTP Bank 10, word 0.

Address: 3035_0000h base + 680h offset = 3035_0680h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAU_KEY0 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word0 (Copy of OTP Bank 10, word 0). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.42 Shadow Register for OTP Bank10 Word1 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY1)

Shadowed memory mapped access to OTP Bank 10, word 1.

Address: 3035_0000h base + 690h offset = 3035_0690h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAU_KEY1 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word1 (Copy of OTP Bank 10, word 1). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.43 Shadow Register for OTP Bank10 Word2 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY2)

Shadowed memory mapped access to OTP Bank 10, word 2.

Address: 3035_0000h base + 6A0h offset = 3035_06A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAU_KEY2 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word2 (Copy of OTP Bank 10, word 2). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.44 Shadow Register for OTP Bank10 Word3 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY3)

Shadowed memory mapped access to OTP Bank 10, word 3.

Address: 3035_0000h base + 6B0h offset = 3035_06B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_MAU_KEY3 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word3 (Copy of OTP Bank 10, word 3). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.45 Shadow Register for OTP Bank11 Word0 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY4)

Shadowed memory mapped access to OTP Bank 11, word 0.

Address: 3035_0000h base + 6C0h offset = 3035_06C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

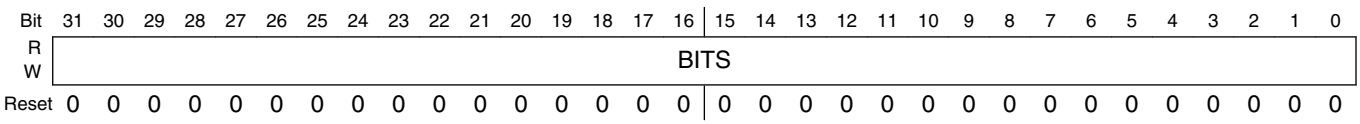
OCOTP_HW_OCOTP_MAU_KEY4 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word4 (Copy of OTP Bank 11, word 0). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.46 Shadow Register for OTP Bank11 Word1 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY5)

Shadowed memory mapped access to OTP Bank 11, word 1.

Address: 3035_0000h base + 6D0h offset = 3035_06D0h



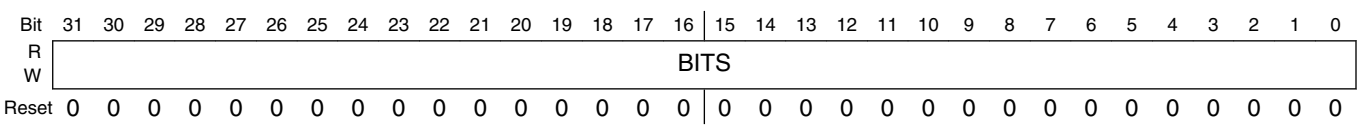
OCOTP_HW_OCOTP_MAU_KEY5 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word5 (Copy of OTP Bank 11, word 1). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.47 Shadow Register for OTP Bank11 Word2 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY6)

Shadowed memory mapped access to OTP Bank 11, word 2.

Address: 3035_0000h base + 6E0h offset = 3035_06E0h



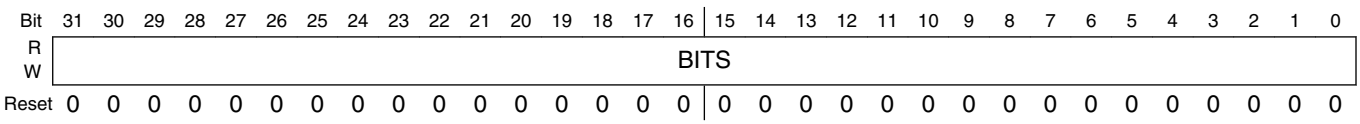
OCOTP_HW_OCOTP_MAU_KEY6 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word6 (Copy of OTP Bank 11, word 2). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.48 Shadow Register for OTP Bank11 Word3 (MAU Key) (OCOTP_HW_OCOTP_MAU_KEY7)

Shadowed memory mapped access to OTP Bank 11, word 3.

Address: 3035_0000h base + 6F0h offset = 3035_06F0h



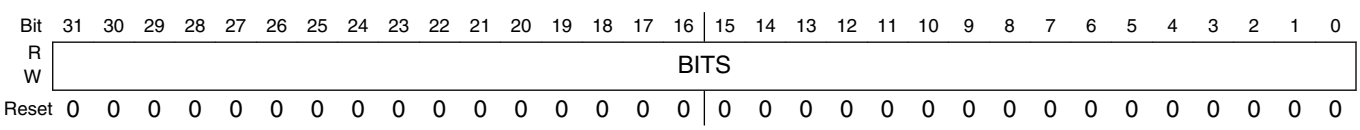
OCOTP_HW_OCOTP_MAU_KEY7 field descriptions

Field	Description
BITS	Shadow register for the MAU Key word7 (Copy of OTP Bank 11, word 3). These bits can be not read and wrotten after the HW_OCOTP_LOCK_MAU_KEY bit is set. If read, returns 0xBADA_BADA and sets HW_OCOTP_CTRL[ERROR].

6.3.5.49 Value of OTP Bank14 Word0 () (OCOTP_HW_OCOTP_GP10)

Shadowed memory mapped access to OTP Bank 14, word 0.

Address: 3035_0000h base + 780h offset = 3035_0780h



OCOTP_HW_OCOTP_GP10 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 14, word 0.

6.3.5.50 Value of OTP Bank14 Word1 () (OCOTP_HW_OCOTP_GP11)

Shadowed memory mapped access to OTP Bank 14, word 1.

Address: 3035_0000h base + 790h offset = 3035_0790h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_GP11 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 14, word 1.

6.3.5.51 Value of OTP Bank14 Word2 () (OCOTP_HW_OCOTP_GP20)

Shadowed memory mapped access to OTP Bank 14, word 2.

Address: 3035_0000h base + 7A0h offset = 3035_07A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_GP20 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 14, word 2.

6.3.5.52 Value of OTP Bank14 Word3 () (OCOTP_HW_OCOTP_GP21)

Shadowed memory mapped access to OTP Bank 14, word 3.

Address: 3035_0000h base + 7B0h offset = 3035_07B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BITS																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

OCOTP_HW_OCOTP_GP21 field descriptions

Field	Description
BITS	Reflects value of OTP Bank 14, word 3.

6.4 Secure Non-Volatile Storage (SNVS)

6.4.1 SNVS introduction

SNVS is a companion module to the CAAM module.

SNVS incorporates both security and non-security functionality. The SNVS non-security functionality is described in this document, but the SNVS security functionality is described only in the Security Reference Manual.

SNVS non-security functions:

- Realtime Counter (RTC) - a software accessible realtime counter
  - RTC can be set to the value in the SRTC
- General Purpose Register - a set of registers used to hold 128 bits of data specified by software
  - If the SNVS_LP power input is connected to an uninterrupted power supply, the GPR value is maintained when main SoC is powered off
- Chip power-on/power-off - If the SNVS_LP power input is connected to an uninterrupted power supply and the Power On button input signal is connected to a power button external to the chip, logic within SNVS_LP can be used to wake the chip from a power down.

6.4.1.1 SNVS feature list

The following table summarizes the features of SNVS:

Table 6-46. SNVS feature list

Feature	Description	Links for Further Information
Real time counter (RTC)	• The RTC is driven by a dedicated clock, which is off when the system power is down. • Programmable time alarm interrupt	SNVS_HP Real Time Counter
General-purpose register	• The general-purpose register is available to software to store 128 bits of data.	Using the General-Purpose Register

Table continues on the next page...

Table 6-46. SNVS feature list (continued)

Feature	Description	Links for Further Information
	• The general-purpose register is zeroized when a security violation is detected. • If the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains), the general-purpose register value is retained even if the main chip is powered down.	
Register access restrictions	• Some registers/values can be written only once per boot cycle.	privileged and non-privileged registers
Wakeup from power off	• Input signal from off chip requests SNVS_LP to power on the main SoC (Assuming that the SNVS_LP power input is connected to an uninterrupted power supply (see SNVS power domains). • Hardware debounces the input signal using software-specified signal bounce characteristics	LP Wake-Up Interrupt Enable

6.4.1.2 SNVS functional description

SNVS implements several non-security features that involve software interaction:

- reading or writing the Realtime Counter (RTC) (This is a non-privileged operation.) - software can also instruct SNVS to load the current SRTC value into the RTC
- reading or writing the General Purpose Register (GPR) (Note that there may be a significant delay when reading or writing registers in the LP section if the LP clock is different from the HP clock.)

The following sections describe in more detail the operation of SNVS.

6.4.2 SNVS Structure

SNVS is organized as two major sub-modules:

- Low-Power Section of SNVS (SNVS_LP)

  The SNVS_LP section provides hardware that enables secure storage and protection of sensitive data. The SNVS module is designed to safely hold security-related data such as cryptographic key, time counter, monotonic counter, and general purpose security information.

  The SNVS_LP block implements the following functional units:
  - Control and Status Registers
  - General Purpose Registers

  When the LP section is connected to an uninterrupted power supply the state of these registers is maintained even when the main chip power is off. (see SNVS power domains)

- High-Power Section of SNVS (SNVS_HP)

  The SNVS_HP section contains all SNVS status and configuration registers. It implements all features that enable system communication and provisioning of the SNVS_LP section.

  The SNVS_HP provides an interface between SNVS_LP and the rest of the system.

  The SNVS_HP block implements the following functional units:
  - IP Bus Interface
  - SNVS_LP Interface
  - Zeroizable Master Key Programming Mechanism
  - Real Time Counter with Alarm Control and Status Registers
  - Control and status registers

  SNVS_HP is in the chip's power supply domain and thus receives power along with the rest of the chip.

The following figure illustrates the structure of SNVS.

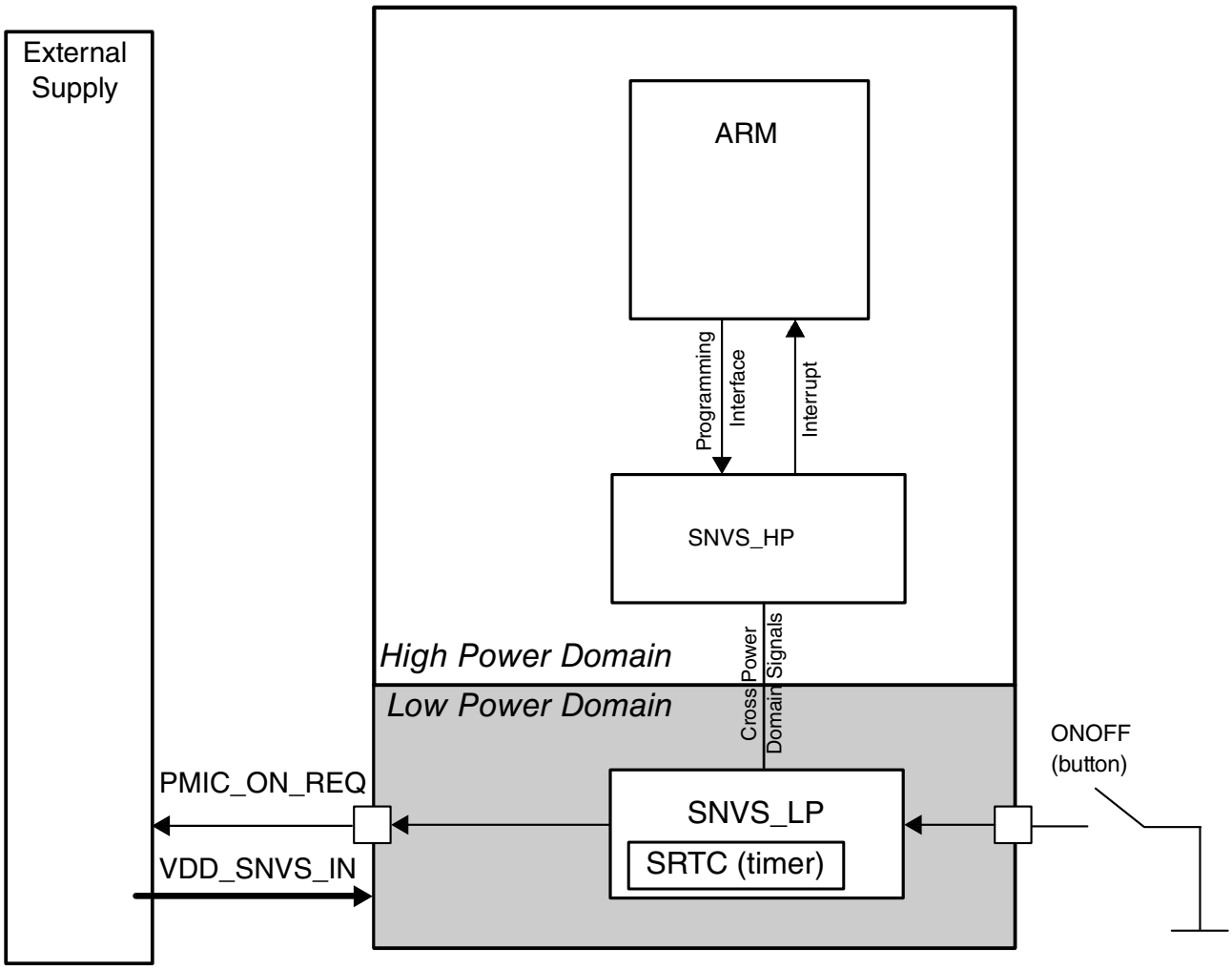


Figure 6-29. SNVS Block Diagram

6.4.2.1 SNVS power domains

In some versions of SNVS (including this version), the LP (Low Power) section is implemented in an independent power domain from the HP (High Power) section, and most other logic on the chip. Throughout the SNVS documentation whenever mention is made of "always-on" logic, this assumes a version of SNVS that implements an independent power domain for the LP section, and that the power for this section is supplied by an uninterrupted power supply. The purpose for the independent power domain is so that data can be retained and certain logic can remain functional even when the main chip logic is powered down. But this is possible only if the LP domain remains powered via an uninterrupted power supply when the main chip power domain is powered off. Usually this uninterrupted power supply would be a battery, with possibly some power management logic to power the LP section from main power (and perhaps

recharge the battery) when main power is on, and switch to battery power when the main power is off. In versions of SNVS with an independent LP power domain the LP section can be electrically isolated from the rest of the chip logic to ensure that its logic does not get corrupted when the main chip is powered down. If the battery runs down or is removed, an LP POR will occur when the LP section next powers up. Note that some OEMs may choose to connect LP power to HP/main chip power and dispense with a battery. In that case the SNVS will operate the same as an SNVS without an independent LP power domain. No state will be retained in the LP section when the chip is powered down, and an LP POR will occur whenever there is an HP POR.

6.4.2.2 Power glitch detector (PGD)

SNVS_LP incorporates a mechanism to detect glitches on the SNVS_LP power supply that might cause the LP control, status, and secure counter values to change. The mechanism works as follows:

1. The PGD register (LPPGDR) is loaded with the known specific value 4173_6166h as part of the SNVS initialization process.
2. Subsequently, this register's value is continuously compared to the hardwired value 4173_6166h.
3. If the comparison indicates that any bit has changed, a power glitch violation is asserted.

Power glitch detection is always enabled and cannot be disabled. At LP POR this register is reset to all 0's, so the hardwired comparison fails and a power glitch violation is reported. Therefore, before programming any feature in the SNVS the power glitch violation should be cleared. The initialization software should write the proper value (4173_6166h) into LPPGDR (see SNVS_LP Power Glitch Detector Register (LPPGDR)) and should then clear the power glitch record in the LP status register (see SNVS_LP Status Register (LPSR)).

The following figure shows the PGD mechanism.

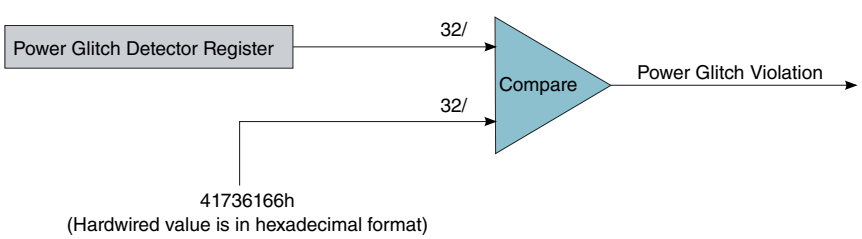


Figure 6-30. Power glitch detector

6.4.2.3 SNVS clock sources

The SNVS has the following clock sources:

- System peripheral clock input. This clock is used by the SNVS's internal logic, for example, the Security State Machine. This clock can be gated outside of the module when the SNVS indicates that it is not in use.
- HP RTC clock. This clock is used by SNVS_HP real-time counter. This clock does not need to be synchronous with other clocks.

6.4.3 Runtime Procedures

SNVS implements a number of features that are intended to be accessed by software at runtime (as opposed to accessed at boot time). These features include:

- Real Time Clock (see SNVS_HP Real Time Counter)
- General Purpose Register (see Using the General-Purpose Register)

Procedures for using these features are described in the following sections.

6.4.3.1 Using SNVS Timer Facilities

SNVS incorporates timer facilities that can optionally generate an interrupt at a specified time. As described in the following sections, SNVS_HP incorporates a Real Time Counter that is available for general use, and SNVS_LP incorporates a Secure Real Time Counter intended for security applications.

6.4.3.1.1 SNVS_HP Real Time Counter

SNVS_HP implements a real time counter that can be read or written by any application; it has no privileged software access restrictions. When the chip is powered down the RTC is not active and it is reset at chip POR. The RTC can be used to generate a functional interrupt request either at a specific time, or at a specific frequency, or both. To generate an interrupt request at a specific time HPTA_EN is set to 0, the desired time is written to HPTA_MS and HPTA_LS and then HPTA_EN is set to 1. HPTA_EN, HPTA_MS and HPTA_LS can be written by any software that has access to SNVS registers; there are no privileged access restrictions. The counter can be synchronized to the SNVS_LP SRTC by writing to the HP_TS bit of SNVS_HP Control Register. This is particularly useful if the SNVS_LP is powered from an uninterrupted power source because the RTC can then be set from a chip-internal time source.

6.4.3.1.2 RTC/SRTC control bits setting

All SNVS registers are programmed from the register bus, consequently any software-initiated changes are synchronized with the IP clock. Several registers can also change synchronously with the RTC/SRTC clock after they are programmed. To avoid IP clock and RTC/SRTC clock synchronization issues, the following values can be changed only when the corresponding function is disabled.

Table 6-47. RTC/SRTC synchronized values list

Function	Value/register	Control bit setting
HP section		
HP Real Time Counter	HPRTCMR and HPRTCLR Registers	RTC_EN = 0 : HPRTCMR/HPRTCLR **can** be programmed RTC_EN = 1 : HPRTCMR/HPRTCLR **cannot** be programmed
HP Time Alarm	HPTAMR and HPTALR Registers	HPTA_EN = 0 : HPTAMR/HPTALR **can** be programmed HPTA_EN = 1 : HPTAMR/HPTALR **cannot** be programmed
LP section		
LP Secure Real Time Counter	LPRTCMR and LPRTCLR Registers	SRTC_ENV = 0 : LPRTCMR/LPRTCLR **can** be programmed SRTC_ENV = 1 : LPRTCMR/LPRTCLR **cannot** be programmed
LP Time Alarm	LPTAR Register	LPTA_EN = 0 : LPTAR **can** be programmed LPTA_EN = 1 : LPTAR **cannot** be programmed

Use the following steps to program synchronized values:

1. Check the enable bit value. If set, clear it.

2. Verify that the enable bit is cleared. There are two reasons to verify the enable bit's setting:

- Enable bit clearing does not happen immediately; it takes three IP clock cycles and two RTC/SRTC clock cycles to change the enable bit's value.
- If the enable bit is locked for programming, it cannot be cleared.

3. Program the desired value.

4. Set the enable bit; it takes three IP clock cycles and two RTC/SRTC clock cycles for the bit to set.

NOTE

Incrementing the value programmed into RTC/SRTC registers by two compensates for the two RTC/SRTC clock cycle delay that is required to enable the counter.

6.4.3.1.3 Reading RTC and SRTC values

Software should follow the following procedure to ensure that it has read correct data from the RTC (HPRTCMR and HPRTCLR) and SRTC (LPSRTCMR and LPSRTCLR) registers:

- Read the most-significant half and the least-significant half of the RTC/SRTC and then read both halves again. If the values read are the same both times, the value is correct.
- If the two consecutive pairs of reads yield different results, perform two more reads.

The worst case scenario may require three sessions of two consecutive pairs of reads. There are several reasons that the values may be incorrectly read initially:

- Synchronization issues between the RTC/SRTC clock and the system clock
- Since the counter continues to increment, there may be a carry from the least-significant 32-bits to the most-significant bits in between reading the two halves of the counter

6.4.3.2 Using Other SNVS Registers

The sections below describe how to use the General Purpose Register.Monotonic Counter.The sections below describe how to use the General Purpose Register and the Monotonic Counter.

6.4.3.2.1 Using the General-Purpose Register

SNVS implements a 128-bit general-purpose register allows software to store a small amount of data. To maintain backward compatibility with versions of SNVS that implement only a 32-bit general purpose register, the most-significant word of the general purpose register is aliased to the original legacy address, and to maintain backward compatibility with versions of snvs_module_name that implement a 128-bit general purpose register, the most-significant half of the general purpose register is aliased to the previous legacy address address. The data in the GPR will be retained during system power-down mode as long as the SNVS_LP remains powered by an uninterrupted power source.

6.4.4 Reset and Initialization of SNVS

SNVS is implemented in two sections (HP and LP) that both must be initialized by software. If the SNVS_LP is powered by an uninterrupted power source that is separate from main SoC power, then SNVS can operate in either of two modes, depending upon whether the main SoC power is on or off. During main SoC power-down SNVS_HP is powered-down, but SNVS_LP is powered from the backup power supply and is electrically isolated from the rest of the chip. In this mode SNVS_LP keeps its registers' values but the LP registers cannot be read or written. During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Both LP and HP registers can be read and written (locks and privilege modes permitting). Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational.

Since the HP and LP sections reside in different power domains, the POR for the two sections can occur at different times. If the SNVS_LP section remains powered by an uninterrupted power source when the main SoC power is off, SNVS_LP is initialized rarely, typically once when the device is first powered on and again whenever the battery is replaced. During main SoC power-up the isolation of SNVS_LP is disabled and both SNVS_HP and SNVS_LP are powered from the main SoC power. Signals between the SNVS_HP and SNVS_LP sections are enabled and all SNVS functions are operational. The SNVS_HP section is powered from the main SoC power, so it must be initialized after the device is powered on.

6.4.5 SNVS register descriptions

This section contains detailed register descriptions for the SNVS registers. Each description includes a standard register diagram and register table. The register table provides detailed descriptions of the register bit and field functions, in bit order.

SNVS registers consist of two types:

- Privileged read/write accessible
- Non-privileged read/write accessible

Privileged read/write accessible registers can only be accessed for read/write by privileged software. Unauthorized write accesses are ignored, and unauthorized read accesses return zero. Non-privileged software can access privileged access registers when the non-privileged software access enable bit is set in the SNVS_HP Command Register.

- Non-Secure
- Trusted
- Secure

Non-privileged read/write accessible registers are read/write accessible by any software.

The LP register values are set only on LP POR and are unaffected by System (HP) POR. The HP registers are set only on System POR and are unaffected by LP POR.

The following table shows the SNVS main memory map.

NOTE

For more information on security-related bitfields, see the Security Reference Manual.

6.4.5.1 SNVS Memory map

SNVS base address: 3037_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	SNVS_HP Command Register (HPCOMR)	32	RW	0000_0000h
8h	SNVS_HP Control Register (HPCR)	32	RW	0000_0000h
14h	SNVS_HP Status Register (HPSR)	32	RW	8000_0000h
24h	SNVS_HP Real Time Counter MSB Register (HPRTCMR)	32	RW	0000_0000h
28h	SNVS_HP Real Time Counter LSB Register (HPRTCLR)	32	RW	0000_0000h
2Ch	SNVS_HP Time Alarm MSB Register (HPTAMR)	32	RW	0000_0000h
30h	SNVS_HP Time Alarm LSB Register (HPTALR)	32	RW	0000_0000h
34h	SNVS_LP Lock Register (LPLR)	32	RW	0000_0000h
38h	SNVS_LP Control Register (LPCR)	32	RW	0000_0020h
4Ch	SNVS_LP Status Register (LPSR)	32	RW	0000_0008h
5Ch	SNVS_LP Secure Monotonic Counter MSB Register (LPSMCMR)	32	RW	0000_0000h
60h	SNVS_LP Secure Monotonic Counter LSB Register (LPSMCLR)	32	RW	0000_0000h
64h	SNVS_LP Power Glitch Detector Register (LPPGDR)	32	RW	0000_0000h
68h	SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)	32	RW	0000_0000h
90h - 9Ch	SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0_alias - LPGPR3_alias)	32	RW	0000_0000h
100h - 10Ch	SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)	32	RW	0000_0000h
BF8h	SNVS_HP Version ID Register 1 (HPVIDR1)	32	RO	003E_0103h
BFCh	SNVS_HP Version ID Register 2 (HPVIDR2)	32	RO	0600_0300h

6.4.5.2 SNVS_HP Command Register (HPCOMR)

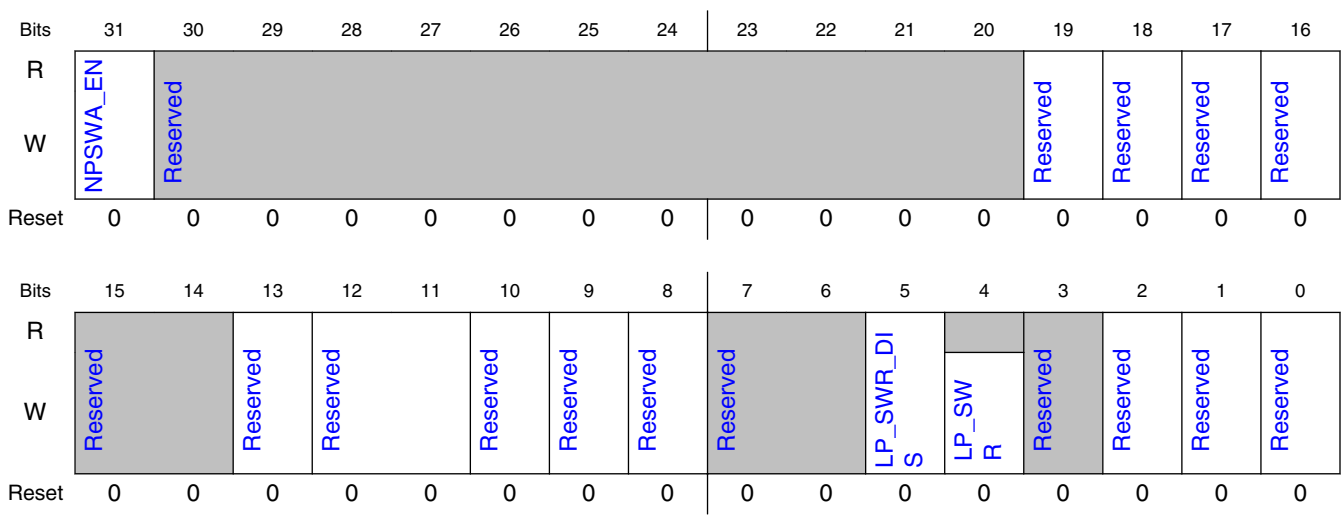
6.4.5.2.1 Offset

Register	Offset
HPCOMR	4h

6.4.5.2.2 Function

The SNVS_HP Command Register contains the command, configuration, and control bits for the SNVS block. This is a privileged write register.

6.4.5.2.3 Diagram



6.4.5.2.4 Fields

Field	Function
31 NPSWA_EN	Non-Privileged Software Access Enable When set, allows non-privileged software to access all SNVS registers, including those that are privileged software read/write access only. 0 Only privileged software can access privileged registers 1 Any software can access privileged registers
30-20	Reserved

Table continues on the next page...

Field	Function
—	
19 —	Reserved
18 —	Reserved
17 —	Reserved
16 —	Reserved
15-14 —	Reserved
13 —	Reserved
12-11 —	Reserved
10 —	Reserved
9 —	Reserved
8 —	Reserved
7-6 —	Reserved
5 LP_SWR_DIS	LP Software Reset Disable When set, disables the LP software reset. Once set, this bit can only be reset by the system reset. 0b - LP software reset is enabled 1b - LP software reset is disabled
4 LP_SWR	LP Software Reset When set to 1, the registers in the SNVS_LP section are reset. 0b - No Action 1b - Reset LP section
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

6.4.5.3 SNVS_HP Control Register (HPCR)

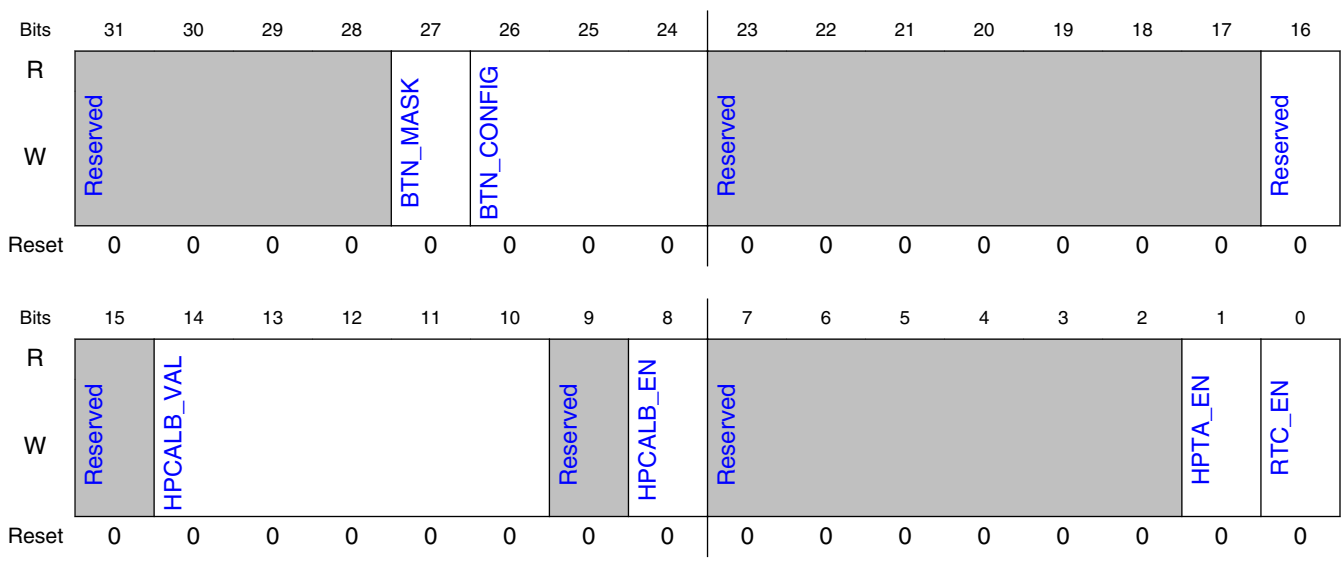
6.4.5.3.1 Offset

Register	Offset
HPCR	8h

6.4.5.3.2 Function

The SNVS_HP Control Register contains various control bits of the HP section of SNVS. This is *not* a privileged write register.

6.4.5.3.3 Diagram



6.4.5.3.4 Fields

Field	Function
31-28 —	Reserved
27 BTN_MASK	Button interrupt mask. This bit is used to mask the ipi_snvs_btn_int_b (button) interrupt request. 0: Interrupt disabled

Table continues on the next page...

Field	Function
	1: Interrupt enabled
26-24 BTN_CONFIG	Button Configuration. This field is used to configure which feature of the button (BTN) input signal constitutes "active". 000: Button signal is active high 001: Button signal is active low 010: Button signal is active on the falling edge 011: Button signal is active on the rising edge 100: Button signal is active on any edge All other patterns are Reserved
23-17 —	Reserved
16 —	Reserved
15 —	Reserved
14-10 HPCALB_VAL	HP Calibration Value Defines signed calibration value for the HP Real Time Counter. This field can be programmed only when RTC Calibration is disabled (HPCALB_EN is not set). This is a 5-bit 2's complement value, hence the allowable calibration values are in the range from -16 to +15 counts per 32768 ticks of the counter. 00000b - +0 counts per each 32768 ticks of the counter 00001b - +1 counts per each 32768 ticks of the counter 00010b - +2 counts per each 32768 ticks of the counter 01111b - +15 counts per each 32768 ticks of the counter 10000b - -16 counts per each 32768 ticks of the counter 10001b - -15 counts per each 32768 ticks of the counter 11110b - -2 counts per each 32768 ticks of the counter 11111b - -1 counts per each 32768 ticks of the counter
9 —	Reserved
8 HPCALB_EN	HP Real Time Counter Calibration Enabled Indicates that the time calibration mechanism is enabled. 0b - HP Timer calibration disabled 1b - HP Timer calibration enabled
7-2 —	Reserved
1 HPTA_EN	HP Time Alarm Enable When set, the time alarm interrupt is generated if the value in the HP Time Alarm Registers is equal to the value of the HP Real Time Counter. 0b - HP Time Alarm Interrupt is disabled 1b - HP Time Alarm Interrupt is enabled
0 RTC_EN	HP Real Time Counter Enable. This bit syncs with the 32KHz clock. It won't update with the bus clock. 0b - RTC is disabled 1b - RTC is enabled

6.4.5.4 SNVS_HP Status Register (HPSR)

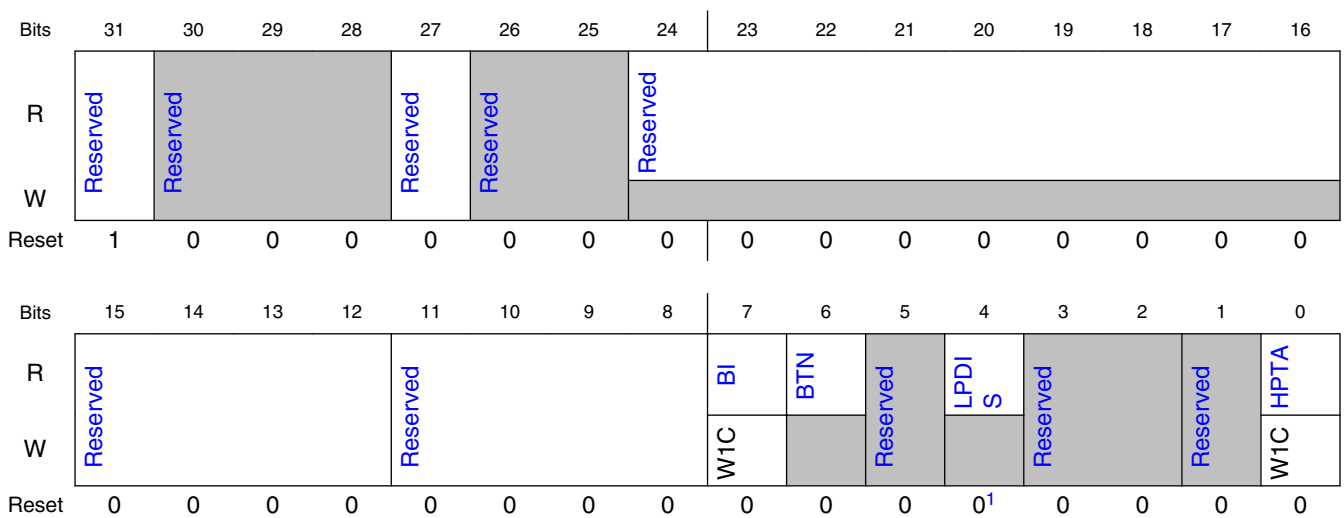
6.4.5.4.1 Offset

Register	Offset
HPSR	14h

6.4.5.4.2 Function

The HP Status Register reflects the internal state of the SNVS. This is *not* a privileged write register.

6.4.5.4.3 Diagram



1. The value of Low Power Disable is determined by the *no_battery* input signal to SNVS.

6.4.5.4.4 Fields

Field	Function
31 —	Reserved
30-28 —	Reserved
27	Reserved

Table continues on the next page...

Field	Function
—	
26-25 —	Reserved
24-16 —	Reserved
15-12 —	Reserved
11-8 —	Reserved
7 BI	Button Interrupt Signal ipi_snvs_btn_int_b was asserted.
6 BTN	Button Value of the BTN input. This is the external button used for PMIC control. 0: BTN not pressed 1: BTN pressed
5 —	Reserved
4 LPDIS	Low Power Disable If 1, the low power section has been disabled by means of an input signal to SNVS.
3-2 —	Reserved
1 —	Reserved
0 HPTA	HP Time Alarm Indicates that the HP Time Alarm has occurred since this bit was last cleared. 0b - No time alarm interrupt occurred. 1b - A time alarm interrupt occurred.

6.4.5.5 SNVS_HP Real Time Counter MSB Register (HPRTCMR)

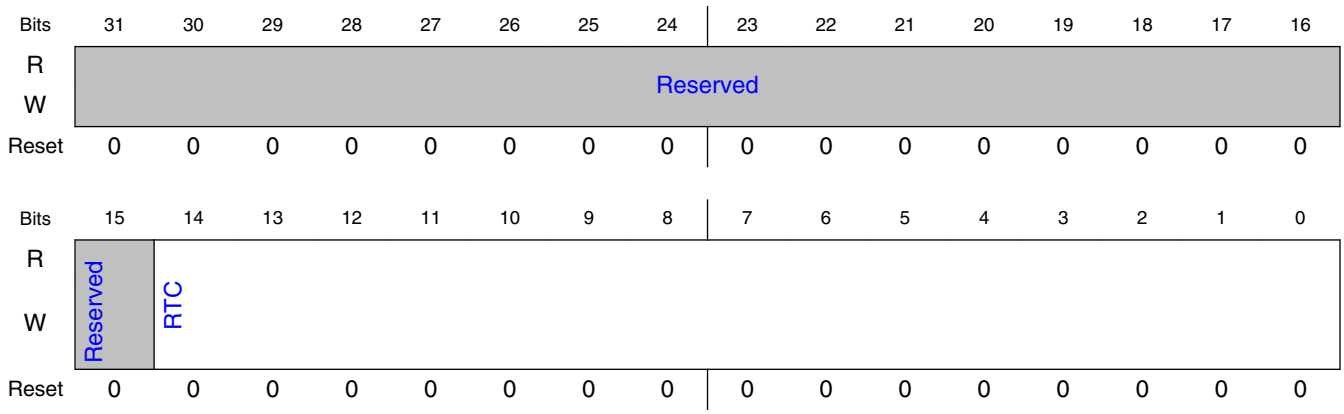
6.4.5.5.1 Offset

Register	Offset
HPRTCMR	24h

6.4.5.5.2 Function

The SNVS_HP Real Time Counter MSB register contains the 15 most-significant bits of the HP Real Time Counter. This is *not* a privileged write register.

6.4.5.5.3 Diagram



6.4.5.5.4 Fields

Field	Function
31-15 —	Reserved
14-0 RTC	HP Real Time Counter The most-significant 15 bits of the RTC. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

6.4.5.6 SNVS_HP Real Time Counter LSB Register (HPRTCLR)

6.4.5.6.1 Offset

Register	Offset
HPRTCLR	28h

6.4.5.6.2 Function

The SNVS_HP Real Time Counter LSB register contains the 32 least-significant bits of the HP real time counter. This is *not* a privileged write register.

6.4.5.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.6.4 Fields

Field	Function
31-0 RTC	HP Real Time Counter least-significant 32 bits. This register can be programmed only when RTC is not active (RTC_EN bit is not set).

6.4.5.7 SNVS_HP Time Alarm MSB Register (HPTAMR)

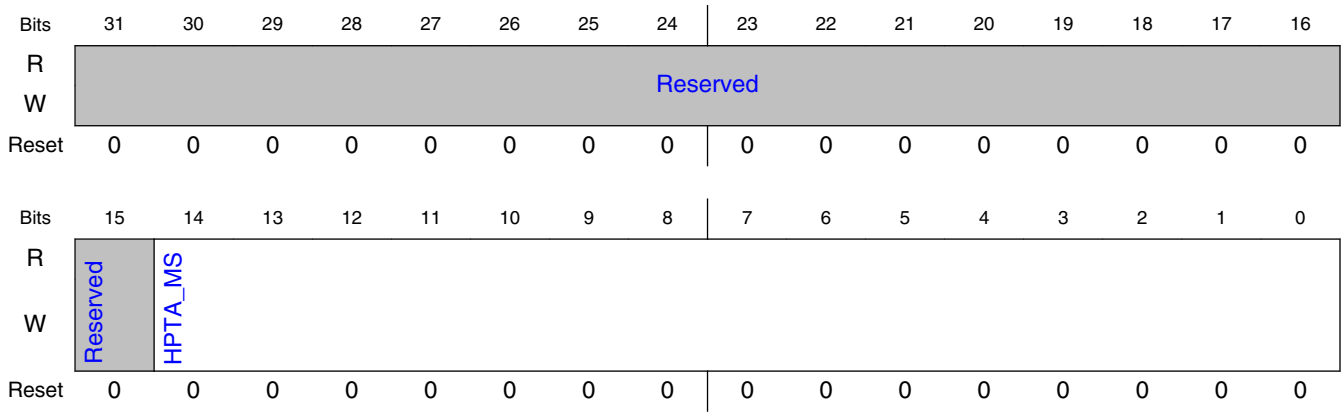
6.4.5.7.1 Offset

Register	Offset
HPTAMR	2Ch

6.4.5.7.2 Function

The SNVS_HP Time Alarm MSB register contains the most-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

6.4.5.7.3 Diagram



6.4.5.7.4 Fields

Field	Function
31-15 —	Reserved
14-0 HPTA_MS	HP Time Alarm, most-significant 15 bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

6.4.5.8 SNVS_HP Time Alarm LSB Register (HPTALR)

6.4.5.8.1 Offset

Register	Offset
HPTALR	30h

6.4.5.8.2 Function

The SNVS_HP Time Alarm LSB register contains the 32 least-significant bits of the SNVS_HP Time Alarm value. This is *not* a privileged write register.

6.4.5.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HPTA_LS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.8.4 Fields

Field	Function
31-0 HPTA_LS	HP Time Alarm, 32 least-significant bits. This register can be programmed only when HP time alarm is disabled (HPTA_EN bit is not set).

6.4.5.9 SNVS_LP Lock Register (LPLR)

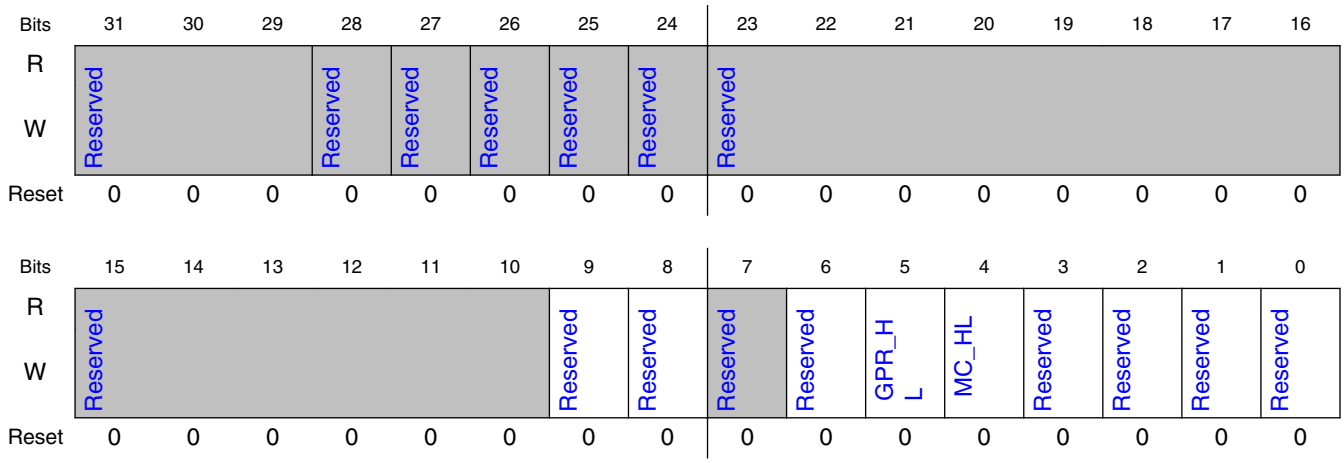
6.4.5.9.1 Offset

Register	Offset
LPLR	34h

6.4.5.9.2 Function

The SNVS_LP Lock Register contains lock bits for the SNVS_LP registers. This is a privileged write register.

6.4.5.9.3 Diagram



6.4.5.9.4 Fields

Field	Function
31-29 —	Reserved
28 —	Reserved
27 —	Reserved
26 —	Reserved
25 —	Reserved
24 —	Reserved
23-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 —	Reserved
6 —	Reserved
5	General Purpose Register Hard Lock

Table continues on the next page...

Field	Function
GPR_HL	When set, prevents any writes to the GPR. Once set, this bit can only be reset by the LP POR. 0b - Write access is allowed. 1b - Write access is not allowed.
4 MC_HL	Monotonic Counter Hard Lock When set, prevents any writes (increments) to the MC Registers and MC_ENV bit. Once set, this bit can only be reset by the LP POR. 0b - Write access (increment) is allowed. 1b - Write access (increment) is not allowed.
3 —	Reserved
2 —	Reserved
1 —	Reserved
0 —	Reserved

6.4.5.10 SNVS_LP Control Register (LPCR)

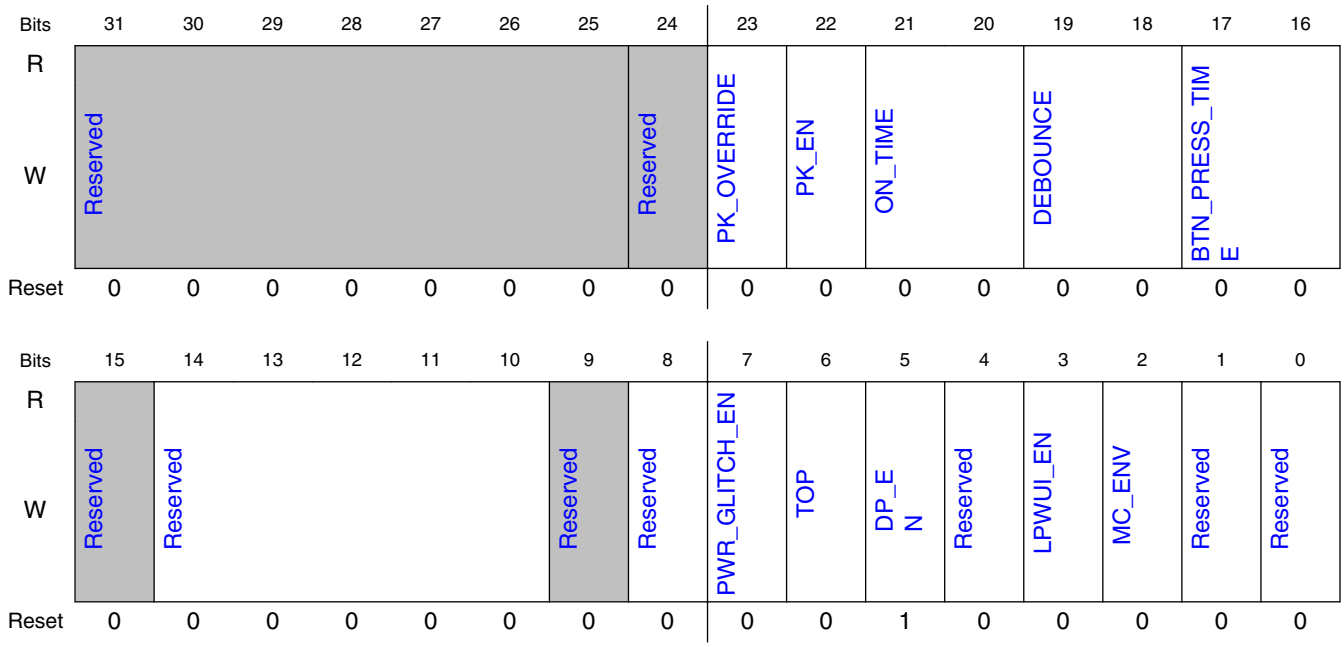
6.4.5.10.1 Offset

Register	Offset
LPCR	38h

6.4.5.10.2 Function

The SNVS_LP Control Register contains various control bits of the LP section of SNVS. This is a privileged write register.

6.4.5.10.3 Diagram



6.4.5.10.4 Fields

Field	Function
31-25 —	Reserved
24 —	Reserved
23 PK_OVERRIDE	PMIC On Request Override The value written to PK_OVERRIDE will be asserted on output signal snvs_lp_pk_override. That signal is used to override the IOMUX control for the PMIC I/O pad.
22 PK_EN	PMIC On Request Enable The value written to PK_EN will be asserted on output signal snvs_lp_pk_en. That signal is used to turn off the pullup/pulldown circuitry in the PMIC I/O pad.
21-20 ON_TIME	The ON_TIME field is used to configure the period of time after BTN is asserted before pmic_en_b is asserted to turn on the SoC power. 00: 500msec off->on transition time 01: 50msec off->on transition time 10: 100msec off->on transition time 11: 0msec off->on transition time
19-18 DEBOUNCE	This field configures the amount of debounce time for the BTN input signal. 00: 50msec debounce 01: 100msec debounce

Table continues on the next page...

Field	Function
	10: 500msec debounce 11: 0msec debounce
17-16 BTN_PRESS_TIME	This field configures the button press time out values for the PMIC Logic. 00 : 5 secs 01 : 10 secs 10 : 15 secs 11 : long press disabled (pmic_en_b will not be asserted regardlessof how long BTN is asserted)
15 —	Reserved
14-10 —	Reserved
9 —	Reserved
8 —	Reserved
7 PWR_GLITCH_EN	Power Glitch Enable By default the detection of a power glitch does not cause the pmic_en_b signal to be asserted. Setting the Power Glitch Enable bit to 1 enables the power glitch event for the PMIC. 0 - disabled 1 - enabled
6 TOP	Turn off System Power Asserting this bit causes a signal to be sent to the Power Management IC to turn off the system power. This bit will clear once power is off. This bit is only valid when the Dumb PMIC is enabled. 0b - Leave system power on. 1b - Turn off system power.
5 DP_EN	Dumb PMIC Enabled When set, software can control the system power. When cleared, the system requires a Smart PMIC to automatically turn power off. 0b - Smart PMIC enabled. 1b - Dumb PMIC enabled.
4 —	Reserved
3 LPWUI_EN	LP Wake-Up Interrupt Enable This interrupt line should be connected to the external pin and is intended to inform the external chip about an SNVS_LP event (MC rollover, SRTC rollover, or time alarm). This wake-up signal can be asserted only when the chip (HP section) is powered down, and the LP section is isolated. 0 LP wake-up interrupt is disabled. 1 LP wake-up interrupt is enabled.
2 MC_ENV	Monotonic Counter Enabled and Valid When set, the MC can be incremented (by write transaction to the LPSMCMR or LPSMCLR). Once MC_SL or MC_HL bit is set this bit can be changed only by LP software reset or LP POR. 0b - MC is disabled or invalid.

Table continues on the next page...

Field	Function
	1b - MC is enabled and valid.
1 —	Reserved
0 —	Reserved

6.4.5.11 SNVS_LP Status Register (LPSR)

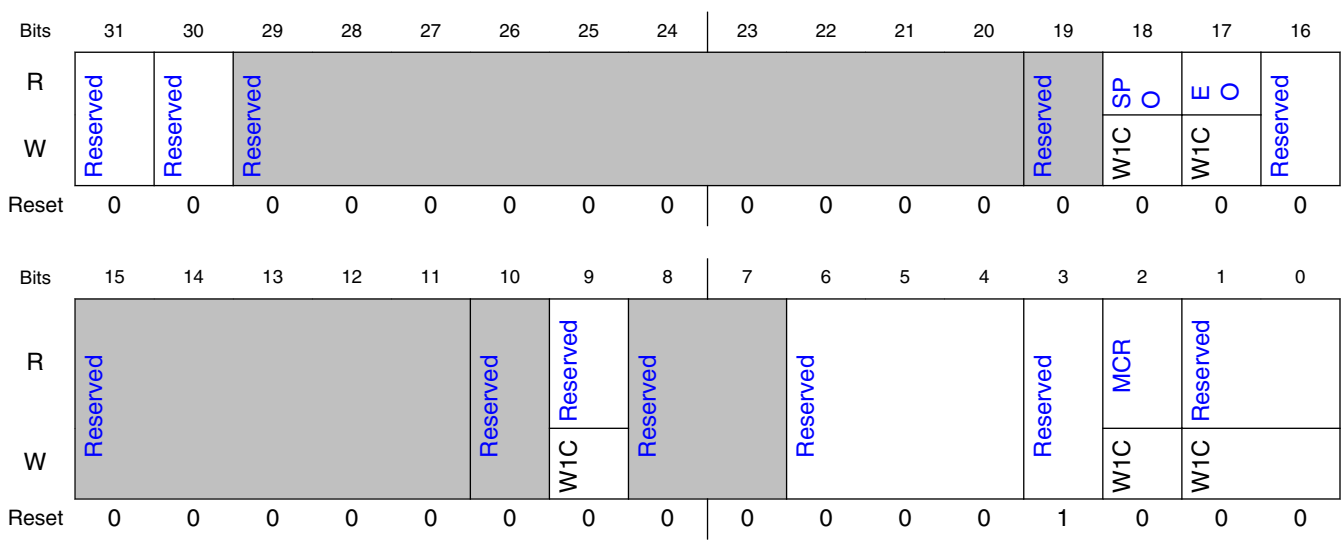
6.4.5.11.1 Offset

Register	Offset
LPSR	4Ch

6.4.5.11.2 Function

The SNVS_LP Status Register reflects the internal state and behavior of the SNVS_LP. This is a privileged write register.

6.4.5.11.3 Diagram



6.4.5.11.4 Fields

Field	Function
31 —	Reserved
30 —	Reserved
29-20 —	Reserved
19 —	Reserved
18 SPO	Set Power Off The SPO bit is set when the power button is pressed longer than the configured debounce time. Writing to the SPO bit will clear the set_pwr_off_irq interrupt. 0b - Set Power Off was not detected. 1b - Set Power Off was detected.
17 EO	Emergency Off This bit is set when a power off is requested. 0b - Emergency off was not detected. 1b - Emergency off was detected.
16 —	Reserved
15-11 —	Reserved
10 —	Reserved
9 —	Reserved
8-7 —	Reserved
6-4 —	Reserved
3 —	Reserved
2 MCR	Monotonic Counter Rollover 0b - MC has not reached its maximum value. 1b - MC has reached its maximum value.
1-0 —	Reserved

6.4.5.12 SNVS_LP Secure Monotonic Counter MSB Register (LPSMCMR)

6.4.5.12.1 Offset

Register	Offset
LPSMCMR	5Ch

6.4.5.12.2 Function

The SNVS_LP Secure Monotonic Counter MSB Register contains the monotonic counter era bits and the most-significant 16 bits of the monotonic counter. The monotonic counter is incremented by one if there is a write command to the LPSMCMR or LPSMCLR register. This is a non-privileged read-only register.

6.4.5.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MC_ERA_BITS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MON_COUNTER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.12.4 Fields

Field	Function
31-16 MC_ERA_BITS	Monotonic Counter Era Bits These bits are inputs to the module and typically connect to fuses. When the Monotonic Counter is in use (i.e. enabled and valid and powered by an uninterrupted power source), and the boot software detects that the Monotonic Counter most-significant 16 Bits and Monotonic Counter LSB Register have been reset (MC_ENV=0), the boot software can take action to ensure that the value in the monotonic counter remains monotonic (i.e. never decreasing). The action is to blow an additional MC_ERA_BITS fuse. Since the MC_ERA_BITS field forms the most-significant field of the monotonic counter, blowing an additional fuse guarantees that the new monotonic counter value is higher than any previous value. Since the Monotonic Counter is reset on an LP Software Reset, an excessive number of MC_ERA_BITS fusez may be consumed if LP Software Reset is used repeatedly.
15-0	Monotonic Counter most-significant 16 Bits

Field	Function
MON_COUNTER	**Note that writing to this register does *not* change the value of this field to the value that was written.** **The 48-bit monotonic counter value (consisting of LPSMCMR[MON_COUNTER] prepended to LPSMCLR[MON_COUNTER]) is incremented by one when:** • A write transaction to the LPSMCMR or LPSMCLR register is detected. • The MC_ENV bit is set. • MC_SL and MC_HL bits are not set. This value can be reset only by LP software reset or LP POR.

6.4.5.13 SNVS_LP Secure Monotonic Counter LSB Register (LPSMCLR)

6.4.5.13.1 Offset

Register	Offset
LPSMCLR	60h

6.4.5.13.2 Function

The SNVS_LP Secure Monotonic Counter LSB Register contains the 32 least-significant bits of the monotonic counter. The MC is incremented by one if there is a write command to the LPSMCMR or LPSMCLR register. This is a non-privileged read-only register.

6.4.5.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MON_COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MON_COUNTER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.13.4 Fields

Field	Function
31-0 MON_COUNTER	Monotonic Counter bits **Note that writing to this register does *not* change the value of this field to the value that was written.** **The 48-bit monotonic counter value (consisting of LPSMCMR[MON_COUNTER] prepended to LPSMCLR[MON_COUNTER]) is incremented by one when:** • A write transaction to the LPSMCMR or LPSMCLR register is detected. • The MC_ENV bit is set. • MC_SL and MC_HL bits are not set. This value can be reset only by LP software reset or LP POR.

6.4.5.14 SNVS_LP Power Glitch Detector Register (LPPGDR)

6.4.5.14.1 Offset

Register	Offset
LPPGDR	64h

6.4.5.14.2 Function

The SNVS_LP Power Glitch Detector Register is a 32-bit read/write register that is used for storing the power glitch detector value, as described in Power glitch detector (PGD). This is a privileged write register.

6.4.5.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PGD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PGD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.14.4 Fields

Field	Function
31-0 PGD	Power Glitch Detector Value

6.4.5.15 SNVS_LP General Purpose Register 0 (legacy alias) (LPGPR0_legacy_alias)

6.4.5.15.1 Offset

Register	Offset
LPGPR0_legacy_alias	68h

6.4.5.15.2 Function

See register SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3).

6.4.5.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.15.4 Fields

Field	Function
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

6.4.5.16 SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0_alias - LPGPR3_alias)

6.4.5.16.1 Offset

Register	Offset
LPGPR0_alias	90h
LPGPR1_alias	94h
LPGPR2_alias	98h
LPGPR3_alias	9Ch

6.4.5.16.2 Function

See register SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3).

6.4.5.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.16.4 Fields

Field	Function
31-0 GPR	General Purpose Register When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

6.4.5.17 SNVS_LP General Purpose Registers 0 .. 3 (LPGPR0 - LPGPR3)

6.4.5.17.1 Offset

Register	Offset
LPGPR0	100h
LPGPR1	104h
LPGPR2	108h
LPGPR3	10Ch

6.4.5.17.2 Function

The SNVS_LP General Purpose Register is a 128-bit read/write register located in SNVS_LP, which can be used by any application for retaining data during an SoC power-down mode. This is a privileged read/write register. The full GPR register is accessed as 4 32-bit registers located in successive word addresses starting at offset 100h. For backward compatibility with earlier versions of SNVS, LPGPR0..LPGPR3 are aliased at the earlier offset of 90h and LPGPR0 is also aliased at its original offset of 68h. New software should access the GPR register at the preferred offset of 100h. The GPR will be automatically zeroized when an enabled security event occurs, unless GPR zeroization is disabled via the GPR_Z_DIS bit in the LP Control Register.

6.4.5.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

6.4.5.17.4 Fields

Field	Function
31-0	General Purpose Register

Field	Function
GPR	When GPR_SL or GPR_HL bit is set, the register cannot be programmed.

6.4.5.18 SNVS_HP Version ID Register 1 (HPVIDR1)

6.4.5.18.1 Offset

Register	Offset
HPVIDR1	BF8h

6.4.5.18.2 Function

The SNVS_HP Version ID Register 1 is a non-privileged read-only register that contains the current version of the SNVS. The version consists of a module ID, a major version number, and a minor version number.

6.4.5.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR_REV								MINOR_REV							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	1

6.4.5.18.4 Fields

Field	Function
31-16 IP_ID	SNVS block ID
15-8 MAJOR_REV	SNVS block major version number
7-0 MINOR_REV	SNVS block minor version number

6.4.5.19 SNVS_HP Version ID Register 2 (HPVIDR2)

6.4.5.19.1 Offset

Register	Offset
HPVIDR2	BFCh

6.4.5.19.2 Function

The SNVS_HP Version ID Register 2 is a non-privileged read-only register that indicates the current version of the SNVS. Version ID register 2 consists of the following fields: integration options, ECO revision, and configuration options.

6.4.5.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	IP_ERA								INTG_OPT							
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECO_REV								CONFIG_OPT							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

6.4.5.19.4 Fields

Field	Function
31-24 IP_ERA	IP Era 00h - Era 1 or 2 03h - Era 3 04h - Era 4 05h - Era 5
23-16 INTG_OPT	SNVS Integration Options
15-8	SNVS ECO Revision

Table continues on the next page...

Field	Function
ECO_REV	
7-0 CONFIG_OPT	SNVS Configuration Options

6.5 System Reset Controller (SRC)

6.5.1 SRC Overview

The System Reset Controller (SRC) controls the reset and boot operation of the SoC.

It is responsible for the generation of all reset signals and boot decoding.

The reset controller determines the source and the type of reset, such as POR, COLD, and performs the necessary reset qualification and stretching sequences. Based on the type of reset, the reset logic generates the reset sequence for the entire IC. Whenever the chip is powered on, the reset is issued through SRC_ONOFF signal and the entire chip is reset.

6.5.1.1 Features

The SRC includes the following features.

- Receives and handles the resets from all the reset sources
- Resets the appropriate domains based upon the resets sources and the nature of the reset
- Latches the SRC_BOOT_MODE pins and common configuration signals from the internal fuse

6.5.2 Clocks

The table found here describes the clock sources for SRC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 6-48. SRC Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

6.5.3 Top-level resets, power-up sequence and external supply integration

Information found here defines chip resets, power-up sequence, and external supply integration.

6.5.3.1 Reset and Power-up Flow

The chip presumes the following reset and power-up flow:

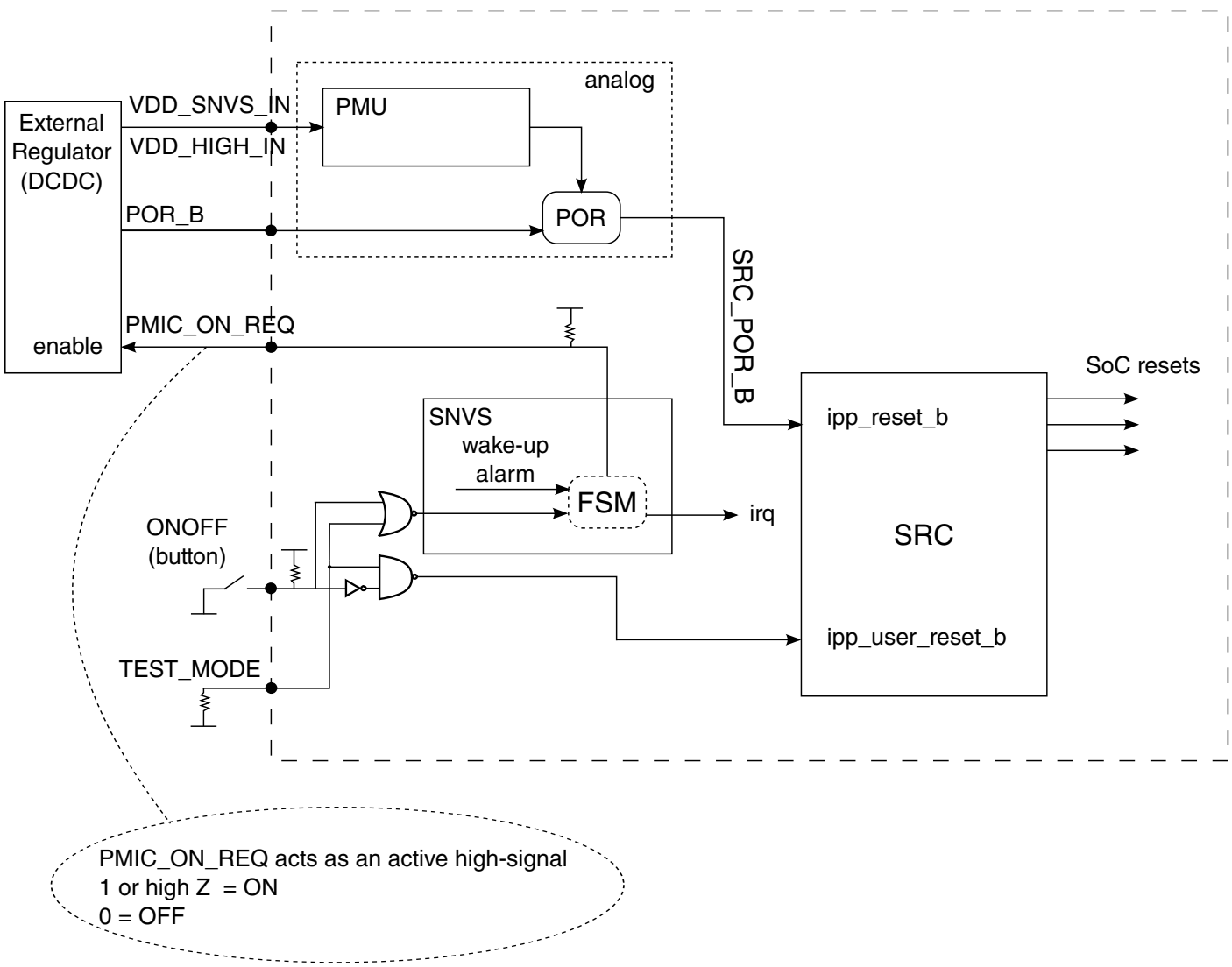


Figure 6-31. Chip reset scheme under PMU control

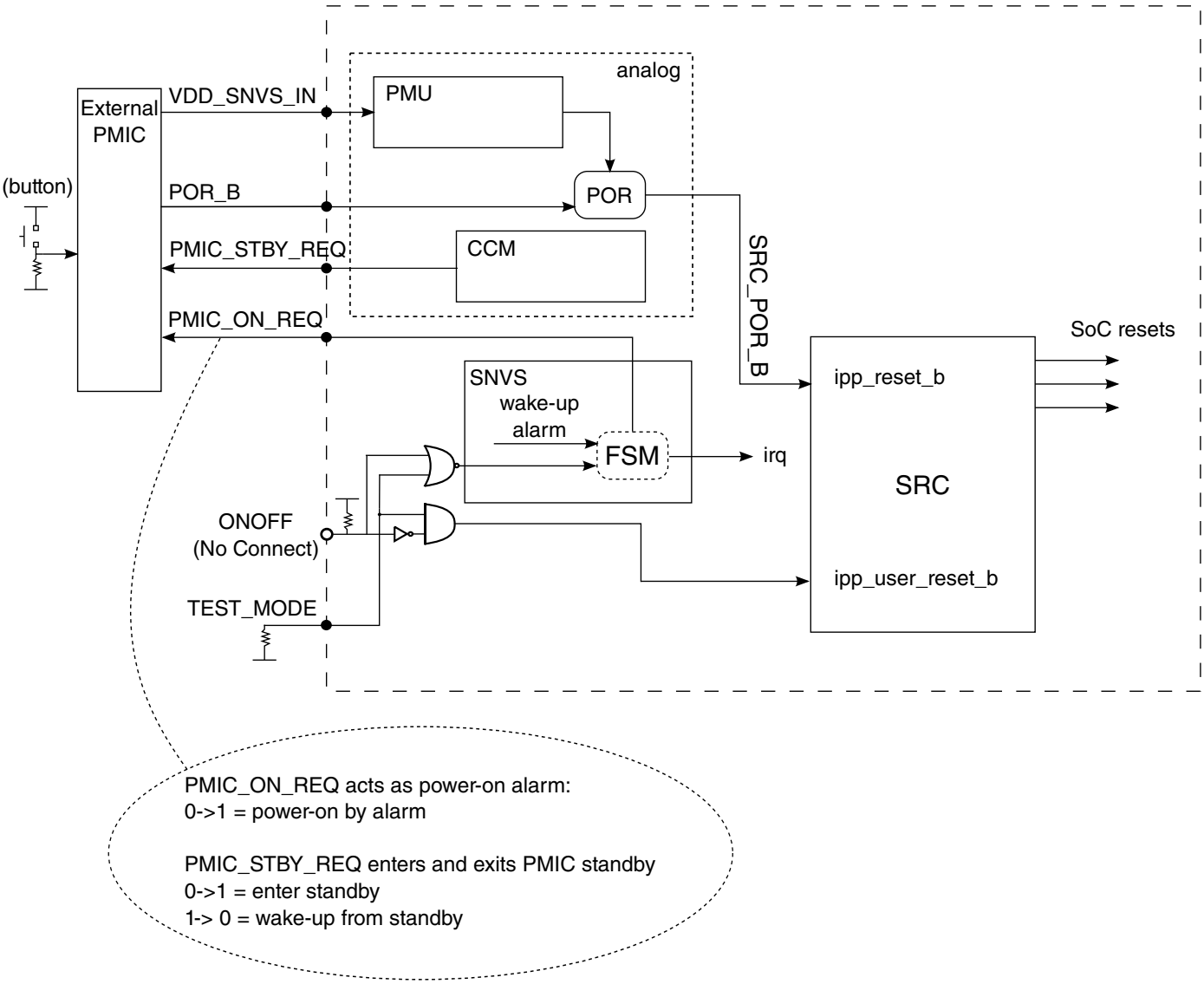


Figure 6-32. Chip reset scheme under external PMIC control

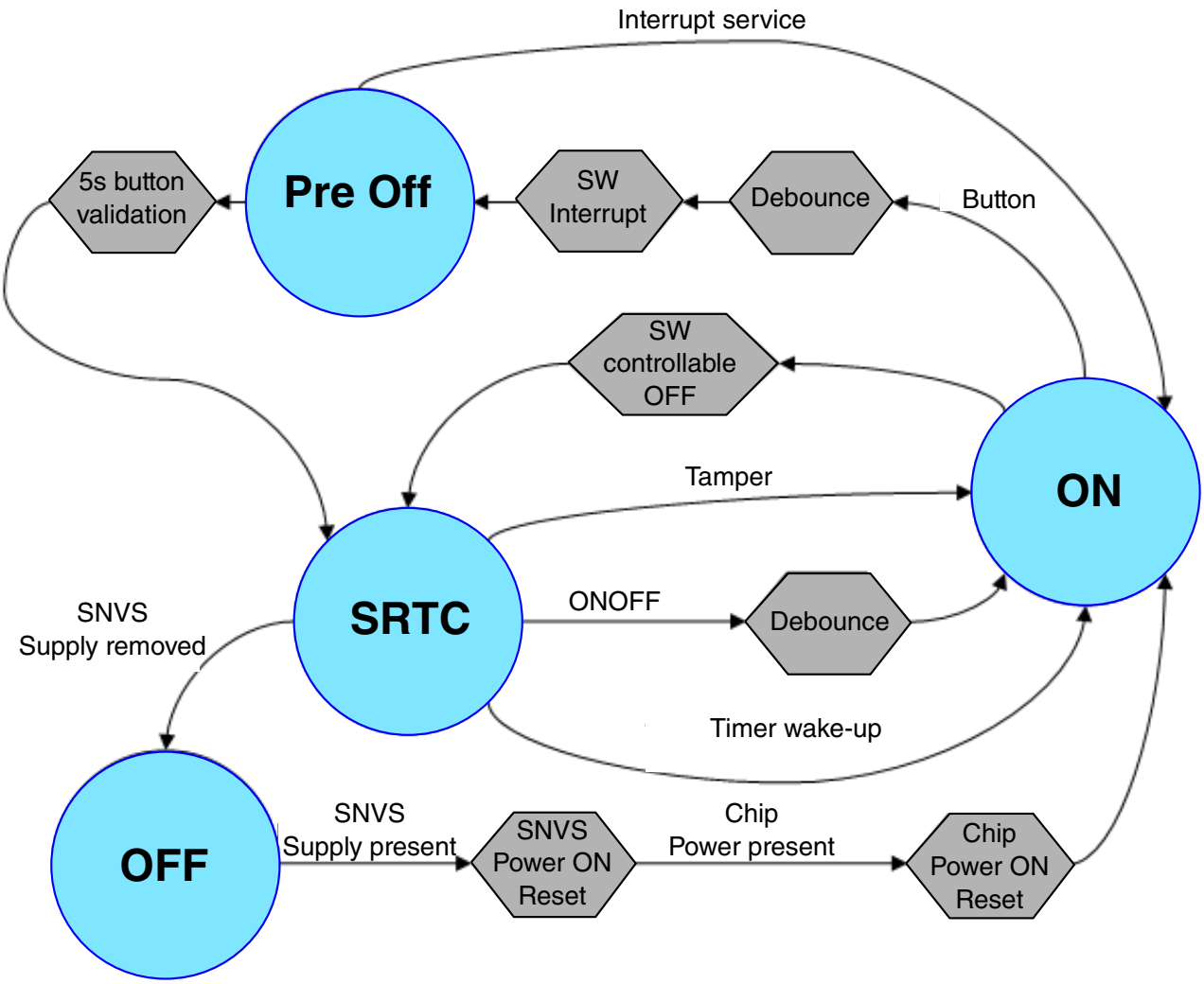


Figure 6-33. Chip on/off state flow diagram

6.5.3.2 Finite-State Machine (FSM)

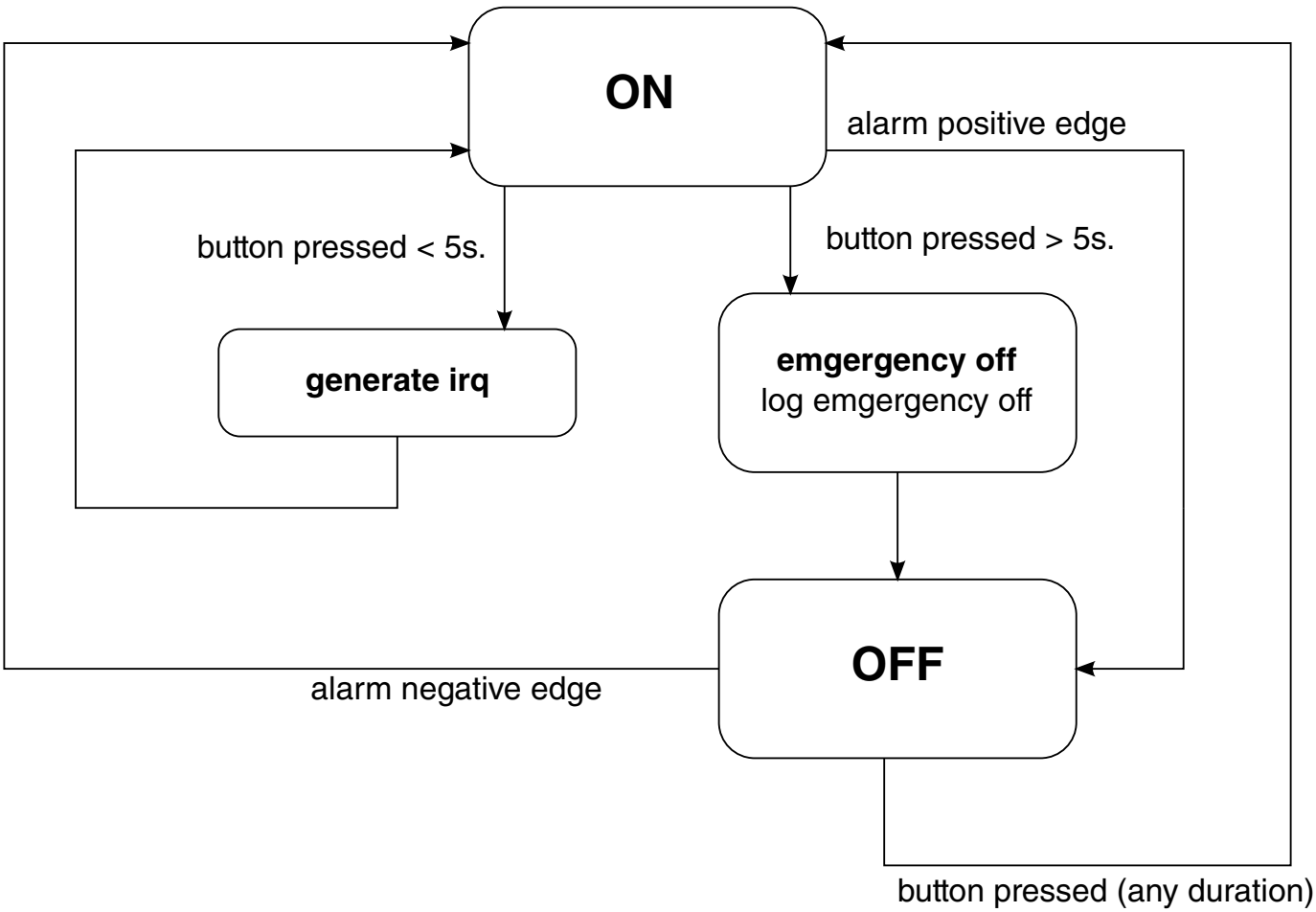


Figure 6-34. FSM

6.5.3.3 Power mode transitions

Table 6-49. Power mode transitions

Power mode	Configuration with external PMIC	Configuration with internal PMIC
ON, first time	1. SoC power supply is connected to SNVS. 2. When button is pressed, PMIC powers on.	1. SoC power supply is connected to SNVS. 2. When button is pressed, 'state' goes ON, PMIC_ON_REQ goes '1'. 3. External regulator is enabled.
Normal ON to OFF, by button	1. Button is pressed for a short duration on the external PMIC. 2. Interrupt request (irq) is sent to SoC from external PMIC. 3. SoC is programming PMIC for power off when standby is asserted.	1. SoC button is pressed for a short duration. 2. Interrupt request (irq) is sent to SoC from FSM. 3. Alarm timer is set up by software routine and started. 4. Upon alarm_in assertion to '1', PMIC_ON_REQ goes '0'. 5. External regulator goes OFF.

Table continues on the next page...

Table 6-49. Power mode transitions (continued)

Power mode	Configuration with external PMIC	Configuration with internal PMIC
	4. In CCM STOP mode, Standby is asserted, PMIC gates SoC supplies.	
Emergency ON to OFF, by button	1. Button is pressed for an extended time on the external PMIC. 2. PMIC is powering off.	1. Button is pressed for longer than 5 seconds on the SoC. 2. FSM validates button pressed for 5 seconds. 3. Emergency power off is logged, PMIC_ON_REQ goes '0', alarm_mask goes '1'. 4. External regulator goes OFF.
OFF to ON, by button	1. Button is pressed on the external PMIC. 2. PMIC powers ON.	1. Button is pressed on the SoC. 2. PMIC_ON_REQ goes '1', alarm_mask goes '0'. 3. External regulator powers ON.
OFF to ON, by timer alarm	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. PMIC receives assertion of PMIC_ON_REQ and wakes up.	1. Timer alarm in SNVS is programmed by software before SoC goes OFF. 2. SoC enters OFF mode. 3. Upon timer limit, wake up alarm goes '0'. PMIC_ON_REQ goes '1'. 4. External regulator is enabled by PMIC_ON_REQ = 1.

6.5.4 Power-On Reset and power sequencing

This module generates an internal POR_B signal that is logically AND'ed with any externally applied SRC_POR_B signal. The internal POR_B signal will be held low until all of the following conditions are met:

- 4ms after the external power supply VDDHIGH_IN is valid
- 1ms after the VDD_SOC_CAP supply is valid

The 4ms and 1ms delays are derived from counting the 32 kHz RTC clock cycles; the accuracy depends on the accuracy of the RTC. When the RTC crystal is either absent or in the process of powering up, an internal ring oscillator will be the source of RTC, which is not as accurate as the crystal.

6.5.4.1 External POR using SRC_POR_B

If the external SRC_POR_B signal is used to control the processor POR, SRC_POR_B must remain low (asserted) until the VDD_ARM_CAP and VDD_SOC_CAP supplies are stable.

6.5.4.2 Internal POR

If the external SRC_POR_B signal is not used (always held high or left unconnected), the processor defaults to the internal POR function (PMU controls generation of the POR based on the power supplies).

If the internal POR function is used, the following power supply requirements must be met:

- VDD_ARM_IN and VDD_SOC_IN may be supplied from the same source, or
- VDD_SOC_IN can be supplied before VDD_ARM_IN with a maximum delay of 1 ms.

6.5.5 Functional Description

6.5.5.1 Reset Control

This section details the reset control of this device.

6.5.5.1.1 Reset inputs and outputs

The reset control logic receives reset requests from all potential reset sources. All the immediate sources of reset are directly passed to the reset stretching block, whereas the resets requiring qualification are passed on to the reset qualification logic before they are sent to the reset stretching block.

All reset inputs and outputs are described in the following figure:

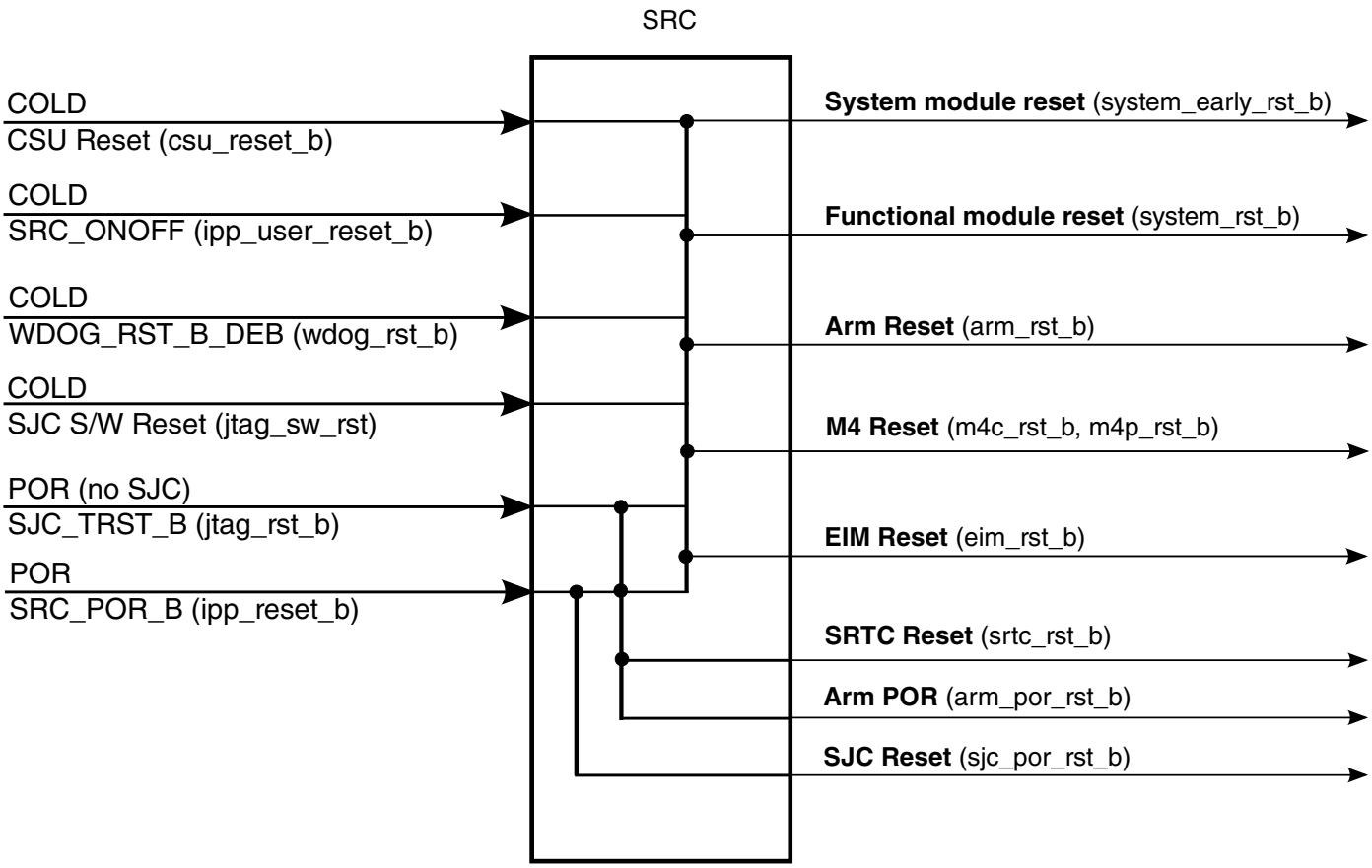


Figure 6-35. SRC inputs and outputs

The reset types and modules they affect are shown in Table 6-50. As there is no chip POR, the POR_B is used to reset the entire chip including test logic and JTAG modules.

NOTE

All resets are expected to be active low except jtag_sw_rst.

Table 6-50. SRC reset functionality

SoC Modules	POR	COLD
System modules (PLLs, fuses, etc)	yes	yes
Functional modules	yes	yes
Arm	yes	yes
Arm SoC	yes	yes
M4 Core	yes	yes
M4 Platform	yes	yes
Arm POR	yes	no
Arm debug	yes	no
SJC	yes	no
SRTC	yes	no

The reset priorities are POR (strongest) and COLD (weakest). If a stronger reset is asserted during the sequence of a weaker reset, then the weaker sequence will be overridden, and the stronger reset sequence will commence. There is no priority within a reset type (POR, etc). If a reset is asserted during the reset sequence of the same type, the reset sequence will be interrupted and restarted.

The following lists the functionality of each of these reset outputs:

- system_early_rst_b - Resets the system modules that need to start first as CCM, OCOTP_CTRL, FUSEBOX, etc.
- system_rst_b - Resets functional modules
- arm_rst_b - Resets Arm module (on regular system reset)
- arm_por_rst_b - Resets Arm POR input
- arm_soc_rst_b - Reset for Arm SOC
- m4c_rst_b - Reset for M4 core
- m4p_rst_b - Reset for M4 platform
- arm_dbg_rst_b - Reset debug logic of Arm
- test_logic_rst_b - Reset test logic (IOMUXC, DAP)
- sjc_por_rst_b - Reset to SJC
- srtc_rst_b - Resets SRTC

NOTE

It is assumed that each reset source will deassert after its assertion, either due to reset generated to the system from SRC, or by negation of the reset source (if it came from an external source to the chip). In the latter case, the reset source is assumed to be held for at least 2 XTALI clocks so it can be sampled by SRC.

6.5.5.1.2 Reset Handling

6.5.5.1.2.1 POR (SRC_POR_B)

SRC_POR_B is an external reset signal. When the chip is powered up, the reset signal is passed through the POR_B pin indicating power-up sequence. The SRC resets the entire chip including the JTAG (SJC) module. All SRC registers will be reset during the POR sequence.

As soon as SRC_POR_B occurs, all resets are asserted and the entire chip is reset by SRC. The SRC_POR_B is stretched for 2 XTALI cycles and the stretching sequence takes place after 2 XTALI clocks of POR_B pin deassertion.

The srtc_rst_b signal is deasserted together with SRC_POR_B signal. The output are also deasserted after the stretching of SRC_POR_B has deasserted.

The sjc_por_rst_b signal is deasserted together with SRC_POR_B signal. The output is also deasserted after the stretching of SRC_POR_B has deasserted.

After the above resets deassert, system_early_rst_b reset is deasserted after 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock ouputs and the system root clocks.

When the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation.

SRC then enables OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL.

- SRC will prepare the boot information
- After 8 ipg cycles, resets to all modules will be de-asserted
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

6.5.5.1.2.2 COLD RESET

The sequence is similar to SRC_POR_B except the memory repair operation is not performed.

After the reset source deasserts, system_early_rst_b reset is deasserted after at least 2 XTALI clocks. The system_early_rst_b is used for the CCM and PLL-IPs to start generating PLL clock outputs and the system root clocks.

After the system root clocks are ready, the CCM will assert system_clk_ready signal. This signal is generated during the start sequence in the CCM and it involves the preparation of the PLLs to generate clock roots for functional operation. See CCM for more information.

After system_clk_ready arrives at the SRC, it will enable OCOTP_CTRL and fusebox clocks, so that fuses can be loaded to OCOTP_CTRL. OCOTP_CTRL will notify with iim_ready_flag when the fusebox loading finishes.

- SRC will prepare the boot information
- After 8 ipg cycles resets to all modules will be deasserted
- After 8 ipg cycles, system clocks will be enabled (en_system_clk).

6.5.5.2 Parallel Reset Requests

SRC will follow the following rules in the case of parallel reset requests:

1. The order of strength of resets is POR - strongest, COLD - weakest
2. If a stronger reset is asserted during weaker reset sequence, then the stronger reset will take over and the stronger reset process will commence. The following cases fall into this category:
   - POR reset request in the middle of cold reset process - the cold will be stopped and the POR sequence will start.
3. If a weaker reset is asserted during stronger reset sequence, then the stronger reset sequence will continue without interference. If at the end of the stronger reset process the weaker request is still asserted then the weaker sequence will commence. The following cases fall into this category:
   - COLD reset requests in the middle of POR reset process - the POR process will continue without interference.
4. If a similar reset request is asserted during the process of reset handling, then the process of reset handling will start over (with the same process). The following cases fall into this category:
   - POR reset request in the middle of POR reset process - the POR process will start over.
   - COLD reset request in the middle of COLD reset process - the COLD process will start over.

6.5.5.3 Boot Mode Control

6.5.5.3.1 BOOT_MODE Pin Latching

The exact boot sequence is controlled by the values of the BOOT_MODE pins on this device.

The value of the BOOT_MODE pins will be latched after the OCOTP_CTRL asserts the fuse read completion flag. After latching, the values of the BOOT_MODE pins are used to determine the booting options of the core as described in the SRC_SBMRx registers.

The boot mode general purpose bits can be provided to the SRC from either e-fuses or GPIO signals. The gpio_bt_sel e-fuse defines the source to be used to derive the boot information. When gpio_bt_sel is set, e-fuses are used. When cleared, GPIO signals are used.

The boot information is provided in SRC_SBMR1 register. The figure below shows the selection of boot mode information.

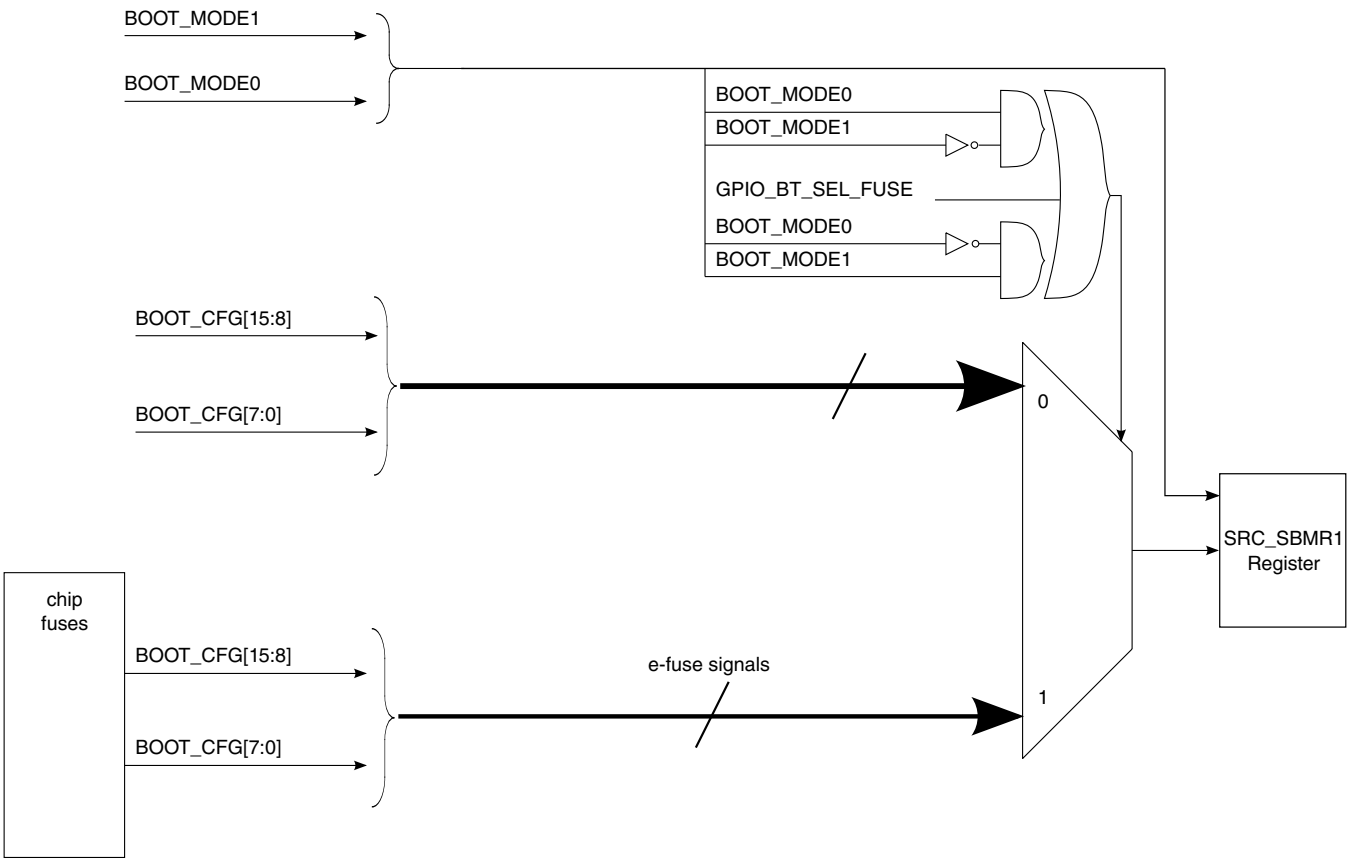


Figure 6-36. Boot mode information

6.5.6 SRC Memory Map/Register Definition

SRC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3039_0000	SRC Reset Control Register (SRC_SCR)	32	R/W	0000_00A0h	6.5.6.1/ 967
3039_0004	A53 Reset Control Register (SRC_A53RCR0)	32	R/W	000A_0000h	6.5.6.2/ 969
3039_0008	A53 Reset Control Register (SRC_A53RCR1)	32	R/W	0000_0001h	6.5.6.3/ 974
3039_000C	M4 Reset Control Register (SRC_M4RCR)	32	R/W	0000_00A8h	6.5.6.4/ 976
3039_0020	USB OTG PHY1 Reset Control Register (SRC_USBOPHY1_RCR)	32	R/W	0000_0001h	6.5.6.5/ 979
3039_0024	USB OTG PHY2 Reset Control Register (SRC_USBOPHY2_RCR)	32	R/W	0000_0001h	6.5.6.6/ 980

Table continues on the next page...

SRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3039_0028	MIPI PHY Reset Control Register (SRC_MIPIPHY_RCR)	32	R/W	0000_0000h	6.5.6.7/ 982
3039_002C	PCIE PHY Reset Control Register (SRC_PCIEPHY_RCR)	32	R/W	0000_000Ah	6.5.6.8/ 984
3039_0030	HDMI Reset Control Register (SRC_HDMI_RCR)	32	R/W	0000_0000h	6.5.6.9/ 987
3039_0034	DISPLAY Reset Control Register (SRC_DISP_RCR)	32	R/W	0000_0000h	6.5.6.10/ 988
3039_0040	GPU Reset Control Register (SRC_GPU_RCR)	32	R/W	0000_0000h	6.5.6.11/ 990
3039_0044	VPU Reset Control Register (SRC_VPU_RCR)	32	R/W	0000_0000h	6.5.6.12/ 991
3039_0048	PCIE2 Reset Control Register (SRC_PCIE2_RCR)	32	R/W	0000_000Ah	6.5.6.13/ 993
3039_004C	MIPI CSI1 PHY Reset Control Register (SRC_MIPIPHY1_RCR)	32	R/W	0000_0000h	6.5.6.14/ 996
3039_0050	MIPI CSI2 PHY Reset Control Register (SRC_MIPIPHY2_RCR)	32	R/W	0000_0000h	6.5.6.15/ 998
3039_0058	SRC Boot Mode Register 1 (SRC_SBMR1)	32	R	0000_0000h	6.5.6.16/ 999
3039_005C	SRC Reset Status Register (SRC_SRSR)	32	R/W	0000_0001h	6.5.6.17/ 1000
3039_0068	SRC Interrupt Status Register (SRC_SISR)	32	R/W	0000_0000h	6.5.6.18/ 1003
3039_006C	SRC Interrupt Mask Register (SRC_SIMR)	32	R/W	0000_03FFh	6.5.6.19/ 1006
3039_0070	SRC Boot Mode Register 2 (SRC_SBMR2)	32	R	0000_0000h	6.5.6.20/ 1010
3039_0074	SRC General Purpose Register 1 (SRC_GPR1)	32	R/W	0000_0000h	6.5.6.21/ 1011
3039_0078	SRC General Purpose Register 2 (SRC_GPR2)	32	R/W	0000_0000h	6.5.6.22/ 1011
3039_007C	SRC General Purpose Register 3 (SRC_GPR3)	32	R/W	0000_0000h	6.5.6.23/ 1012
3039_0080	SRC General Purpose Register 4 (SRC_GPR4)	32	R/W	0000_0000h	6.5.6.24/ 1012
3039_0084	SRC General Purpose Register 5 (SRC_GPR5)	32	R/W	0000_0000h	6.5.6.25/ 1012
3039_0088	SRC General Purpose Register 6 (SRC_GPR6)	32	R/W	0000_0000h	6.5.6.26/ 1013
3039_008C	SRC General Purpose Register 7 (SRC_GPR7)	32	R/W	0000_0000h	6.5.6.27/ 1013
3039_0090	SRC General Purpose Register 8 (SRC_GPR8)	32	R/W	0000_0000h	6.5.6.28/ 1013

Table continues on the next page...

SRC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3039_0094	SRC General Purpose Register 9 (SRC_GPR9)	32	R/W	0000_0000h	6.5.6.29/ 1014
3039_0098	SRC General Purpose Register 10 (SRC_GPR10)	32	R/W	0000_0000h	6.5.6.30/ 1015
3039_1000	SRC DDR Controller Reset Control Register (SRC_DDRC_RCR)	32	R/W	0000_000Fh	6.5.6.31/ 1016
3039_1004	SRC DDRC2 Controller Reset Control Register (SRC_DDRC2_RCR)	32	R/W	0000_000Fh	6.5.6.32/ 1018

6.5.6.1 SRC Reset Control Register (SRC_SCR)

The reset control register (SCR), contains bits that control operation of the reset controller.

Address: 3039_0000h base + 0h offset = 3039_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								MASK_TEMPSENSE_ RESET				Reserved			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

SRC_SCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock

Table continues on the next page...

SRC_SCR field descriptions (continued)

Field	Description
	NOTE: Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–8 -	This field is reserved. Reserved
7–4 MASK_ TEMPSENSE_ RESET	Mask tempsense_reset source. If these 4 bits are coded from A to 5 then, the tempsense_reset input to SRC will be masked and the tempsense_reset will not create a reset to the chip. 0101 tempsense_reset is masked 1010 tempsense_reset is not masked
-	This field is reserved. Reserved

6.5.6.2 A53 Reset Control Register (SRC_A53RCR0)

The A53 Reset Control Register (A53RCR), contains bits that control the A53 reset generation.

Address: 3039_0000h base + 4h offset = 3039_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved		A53_L2RESET	A53_SOC_DBG_RESET	MASK_WDOG1_RST			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	A53_ETM_RESET3	A53_ETM_RESET2	A53_ETM_RESET1	A53_ETM_RESET0	A53_DBG_RESET3	A53_DBG_RESET2	A53_DBG_RESET1	A53_DBG_RESET0	A53_CORE_RESET3	A53_CORE_RESET2	A53_CORE_RESET1	A53_CORE_RESET0	A53_CORE_POR_RESET3	A53_CORE_POR_RESET2	A53_CORE_POR_RESET1	A53_CORE_POR_RESET0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_A53RCR0 field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–22 -	This field is reserved. Reserved
21 A53_L2RESET	Software reset for A53 Snoop Control Unit (SCU). **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert SCU reset 1 assert SCU reset
20 A53_SOC_DBG_ RESET	Software reset for system level debug reset. It initializes the shared Debug APB, the CTI, and the CTM. It also causes: • A53_dbgreset[3:0] and A53_etmreset[3:0] to be asserted • debug logic in the processor power domain and in the debug power domain to be reset **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert system level debug reset 1 assert system level debug reset
19–16 MASK_WDOG1_ RST	Mask wdog1_rst_b source. If these 4 bits are coded from A to 5 then, the wdog1_rst_b input to SRC will be masked and the wdog1_rst_b will not create a reset to the chip. **NOTE:** During the time the WDOG event is masked using SRC logic, it is likely that the WDOG Reset Status Register (WRSR) bit 1 (which indicates a WDOG timeout event) will get asserted. software / OS developer must prepare for this case. Re-enabling the WDOG is possible, by un-masking it in SRC, though it must be preceded by servicing the WDOG. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG module. (Hardware reset is the only way to cause the de-assertion of that bit). Any other code will be coded to 1010 i.e. wdog1_rst_b is not masked 0101 wdog1_rst_b is masked 1010 wdog1_rst_b is not masked

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

Field	Description
15 A53_ETM_ RESET3	Software reset for core3 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the rest process will begin, and once it finished, this bit will be self-cleared. 0 do not assert core3 ETM reset 1 assert core3 ETM reset
14 A53_ETM_ RESET2	Software reset for core2 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the rest process will begin, and once it finished, this bit will be self-cleared. 0 do not assert core2 ETM reset 1 assert core2 ETM reset
13 A53_ETM_ RESET1	Software reset for core1 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 ETM reset 1 assert core1 ETM reset
12 A53_ETM_ RESET0	Software reset for core0 ETM only. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 ETM reset 1 assert core0 ETM reset
11 A53_DBG_ RESET3	Software reset for core3 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core3 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 debug reset 1 assert core3 debug reset
10 A53_DBG_ RESET2	Software reset for core2 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core2 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 debug reset 1 assert core2 debug reset
9 A53_DBG_ RESET1	Software reset for core1 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core1 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain.

Table continues on the next page...

SRC_A53RCR0 field descriptions (continued)

Field	Description
	NOTE: This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 debug reset 1 assert core1 debug reset
8 A53_DBG_RESET0	Software reset for core0 debug only. It initialize the debug, and breakpoint and watchpoint logic in the core1 processor power domain. It also reset the debug logic for core1 processor, which is in the debug power domain. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 debug reset 1 assert core0 debug reset
7 A53_CORE_RESET3	Software reset for core3 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 reset 1 assert core3 reset
6 A53_CORE_RESET2	Software reset for core2 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 reset 1 assert core2 reset
5 A53_CORE_RESET1	Software reset for core1 only. It initializes the processor logic in the core1 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 reset 1 assert core1 reset
4 A53_CORE_RESET0	Software reset for core0 only. It initializes the processor logic in the core0 processor power domains, not including the debug, breakpoint and watchpoint logic. **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 reset 1 assert core0 reset
3 A53_CORE_POR_RESET3	POR reset for A53 core3 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core3 processor power domains

Table continues on the next page...

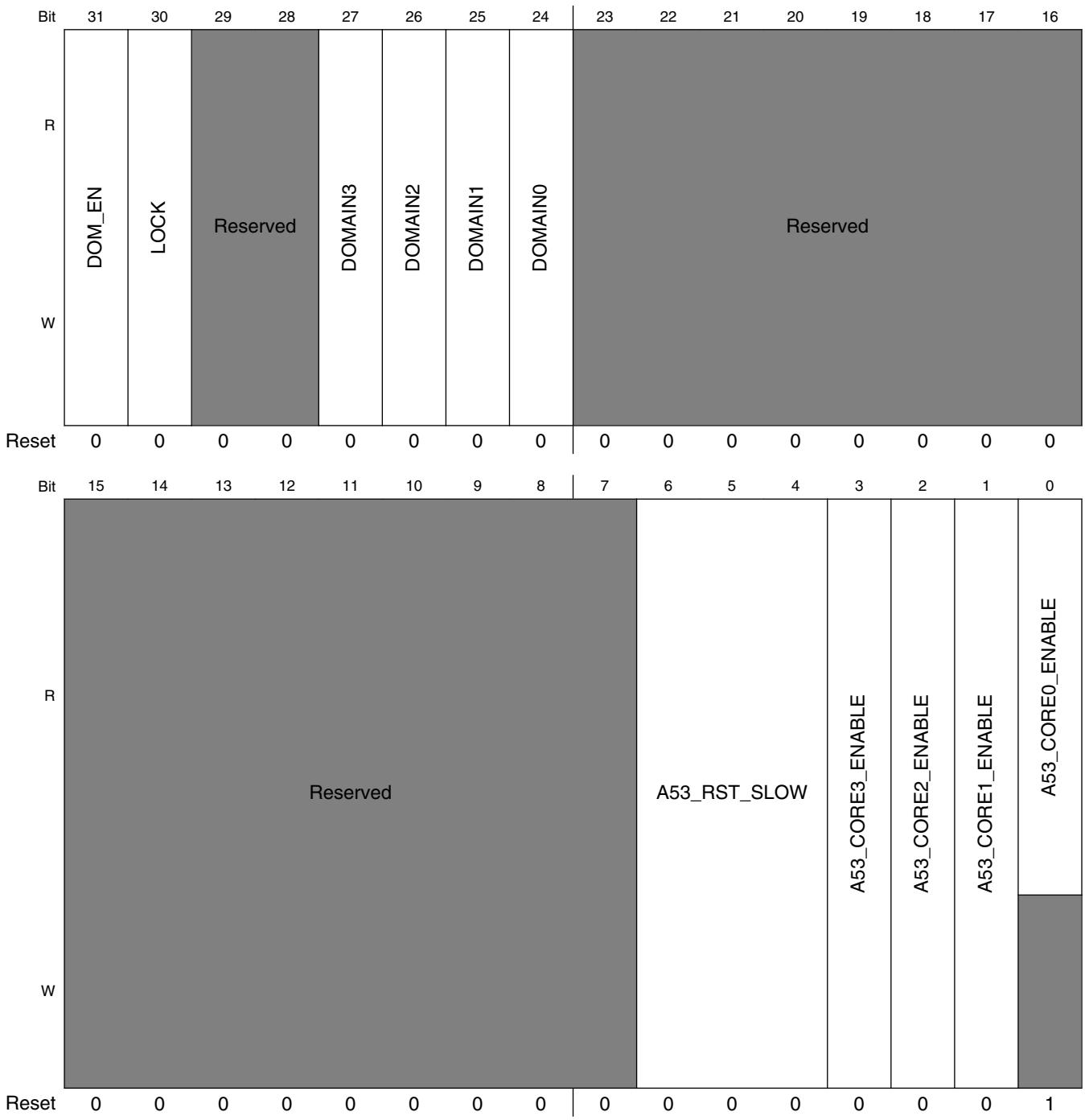
SRC_A53RCR0 field descriptions (continued)

Field	Description
	NOTE: This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core3 reset 1 assert core3 reset
2 A53_CORE_ POR_RESET2	POR reset for A53 core2 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core2 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core2 reset 1 assert core2 reset
1 A53_CORE_ POR_RESET1	POR reset for A53 core1 only. It initializes all the core1 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core1 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core1 reset 1 assert core1 reset
0 A53_CORE_ POR_RESET0	POR reset for A53 core0 only. It initializes all the core0 processor logic, including CPU Debug, and breakpoint and watchpoint logic in the core0 processor power domains **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. 0 do not assert core0 reset 1 assert core0 reset

6.5.6.3 A53 Reset Control Register (SRC_A53RCR1)

The A53 Reset Control Register (A53RCR), contains bits that control the A53 reset generation.

Address: 3039_0000h base + 8h offset = 3039_0008h



SRC_A53RCR1 field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–7 -	This field is reserved. Reserved
6–4 A53_RST_SLOW	A53_RST_SLOW
3 A53_CORE3_ENABLE	core 3 enable 0 core3 is disabled 1 core3 is enabled
2 A53_CORE2_ENABLE	core 2 enable 0 core2 is disabled 1 core2 is enabled
1 A53_CORE1_ENABLE	core 1 enable

Table continues on the next page...

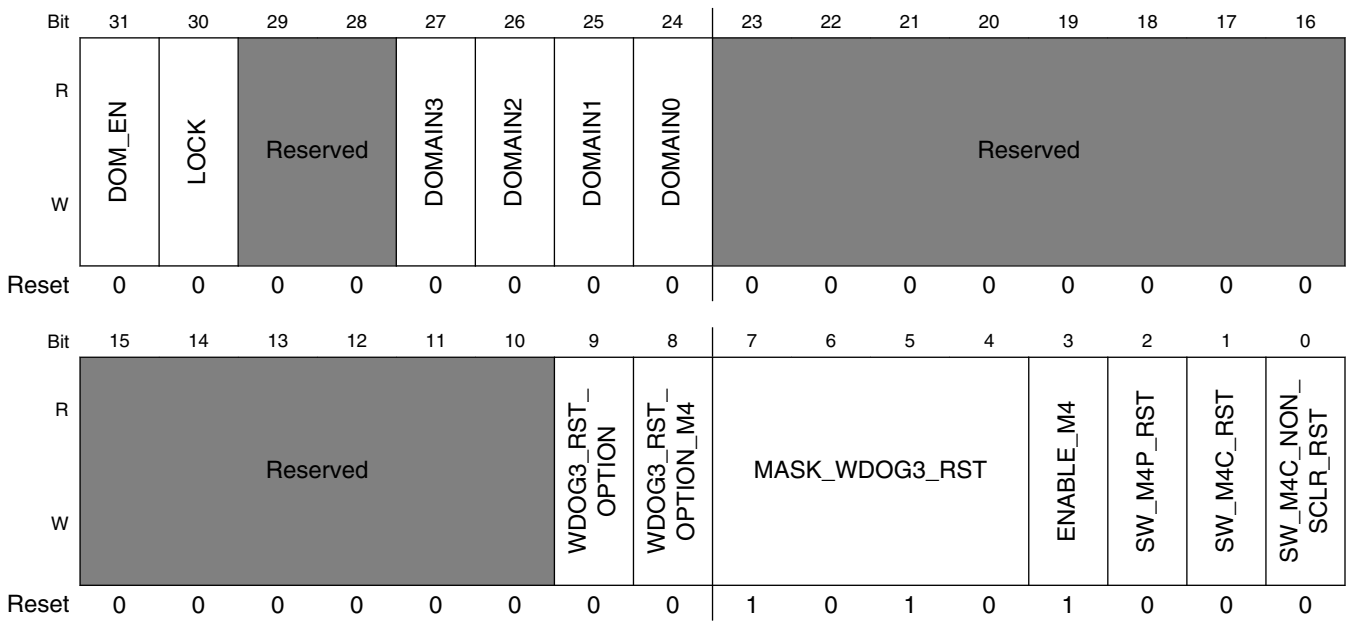
SRC_A53RCR1 field descriptions (continued)

Field	Description
	0 core1 is disabled 1 core1 is enabled
0 A53_CORE0_ENABLE	Always 1, can't be changed.

6.5.6.4 M4 Reset Control Register (SRC_M4RCR)

The M4 Reset Control Register (M4RCR), contains bits that control the M4 reset generation.

Address: 3039_0000h base + Ch offset = 3039_000Ch



SRC_M4RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock

Table continues on the next page...

SRC_M4RCR field descriptions (continued)

Field	Description
	NOTE: Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–10 -	This field is reserved. Reserved
9 WDOG3_RST_ OPTION	Wdog3_rst_b option 0 Wdog3_rst_b asserts M4 reset 1 Wdog3_rst_b asserts global reset
8 WDOG3_RST_ OPTION_M4	Wdog3_rst_b option for M4. This bit is only effective when wdog3_rst_option is set to 1. 0 wdgo3_rst_b Reset M4 core only 1 Reset both M4 core and platform
7–4 MASK_WDOG3_ RST	Mask wdog3_rst_b source. If these 4 bits are coded from A to 5 then, the wdog3_rst_b input to SRC will be masked and the wdog3_rst_b will not create a reset to the chip. **NOTE:** During the time the WDOG3 event is masked using SRC logic, it is likely that the WDOG3 Reset Status Register (WRSR) bit 1 (which indicates a WDOG3 timeout event) will get asserted. Software / OS developer must prepare for this case. Re-enabling the WDOG3 is possible, by un-masking it in SRC, though it must be preceded by servicing the WDOG3. However, for the case that the event has been asserted, the status bit (WRSR bit-1) will remain asserted, regardless of servicing the WDOG3 module. (Hardware reset is the only way to cause the de-assertion of that bit). Any other code will be coded to 1010 i.e. wdog3_rst_b is not masked 0101 wdog3_rst_b is masked 1010 wdog3_rst_b is not masked
3 ENABLE_M4	Enable M4 0 M4 is disabled 1 M4 is enabled

Table continues on the next page...

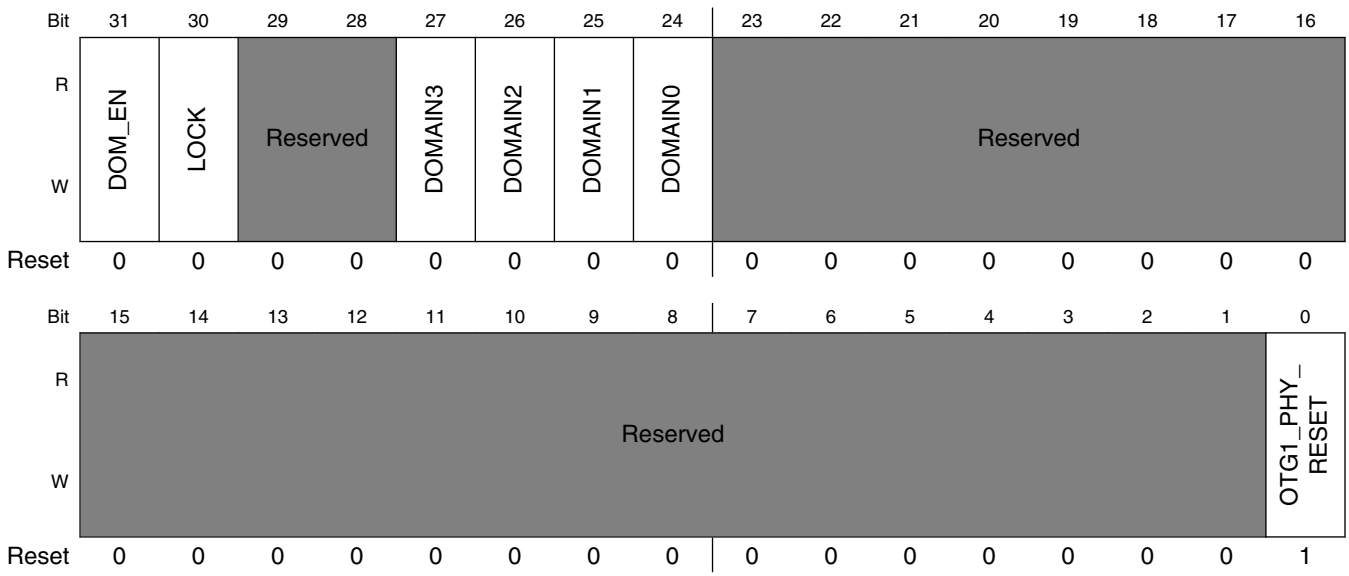
SRC_M4RCR field descriptions (continued)

Field	Description
2 SW_M4P_RST	Self-clearing SW reset for M4 platform **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SRC_SISR register for details. 0 do not assert M4 platform reset 1 assert M4 platform reset
1 SW_M4C_RST	Self-clearing SW reset for M4 core **NOTE:** This is a self clearing bit. Once it is set to 1, the reset process will begin, and once it finishes, this bit will be self cleared. Software can determine that the reset has finished once this bit is cleared. Software can also configure SRC to generate interrupt once the software has finished. Please refer to SRC_SISR register for details. 0 do not assert M4 core reset 1 assert M4 core reset
0 SW_M4C_NON_SCLR_RST	Non-self-clearing SW reset for M4 core 0 do not assert M4 core reset 1 assert M4 core reset

6.5.6.5 USB OTG PHY1 Reset Control Register (SRC_USBOPHY1_RCR)

The USB OTG PHY1 Reset Control Register (SRC_IP_RCR2), contains bits that control the USB OTG PHY1 reset generation.

Address: 3039_0000h base + 20h offset = 3039_0020h



SRC_USBOPHY1_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

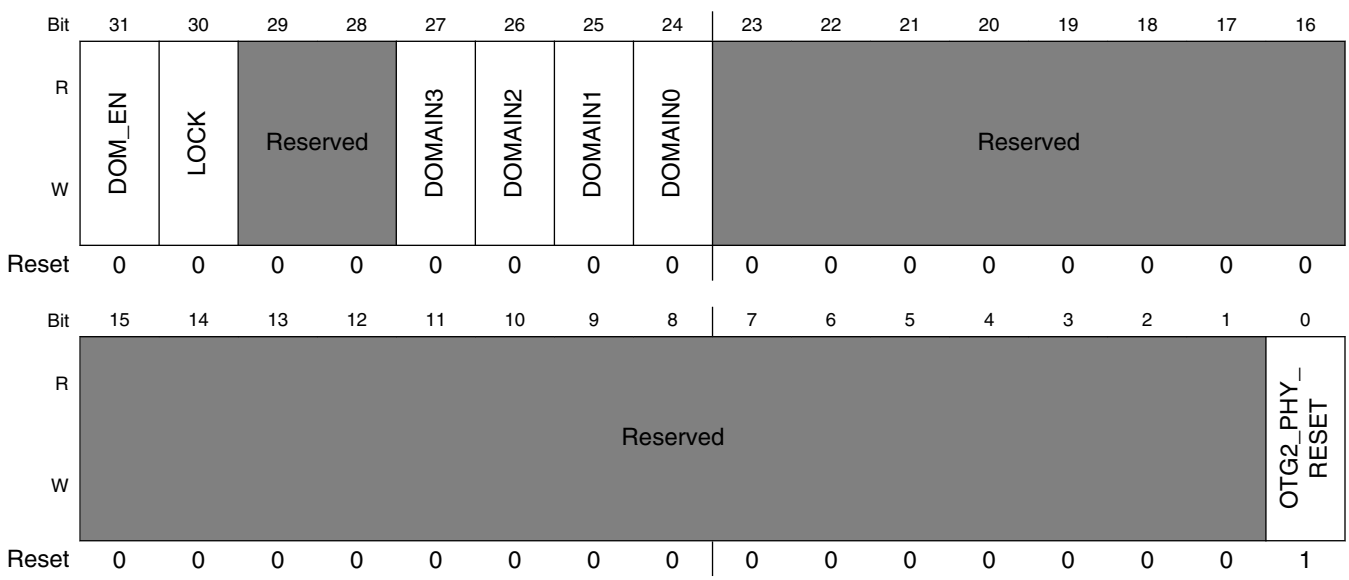
SRC_USBOPHY1_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 OTG1_PHY_RESET	0 Don't reset USB OTG1 PHY 1 Reset USB OTG1 PHY

6.5.6.6 USB OTG PHY2 Reset Control Register (SRC_USBOPHY2_RCR)

The USB OTG PHY2 Reset Control Register (SRC_IP_RCR2), contains bits that control the USB OTG PHY2 reset generation.

Address: 3039_0000h base + 24h offset = 3039_0024h



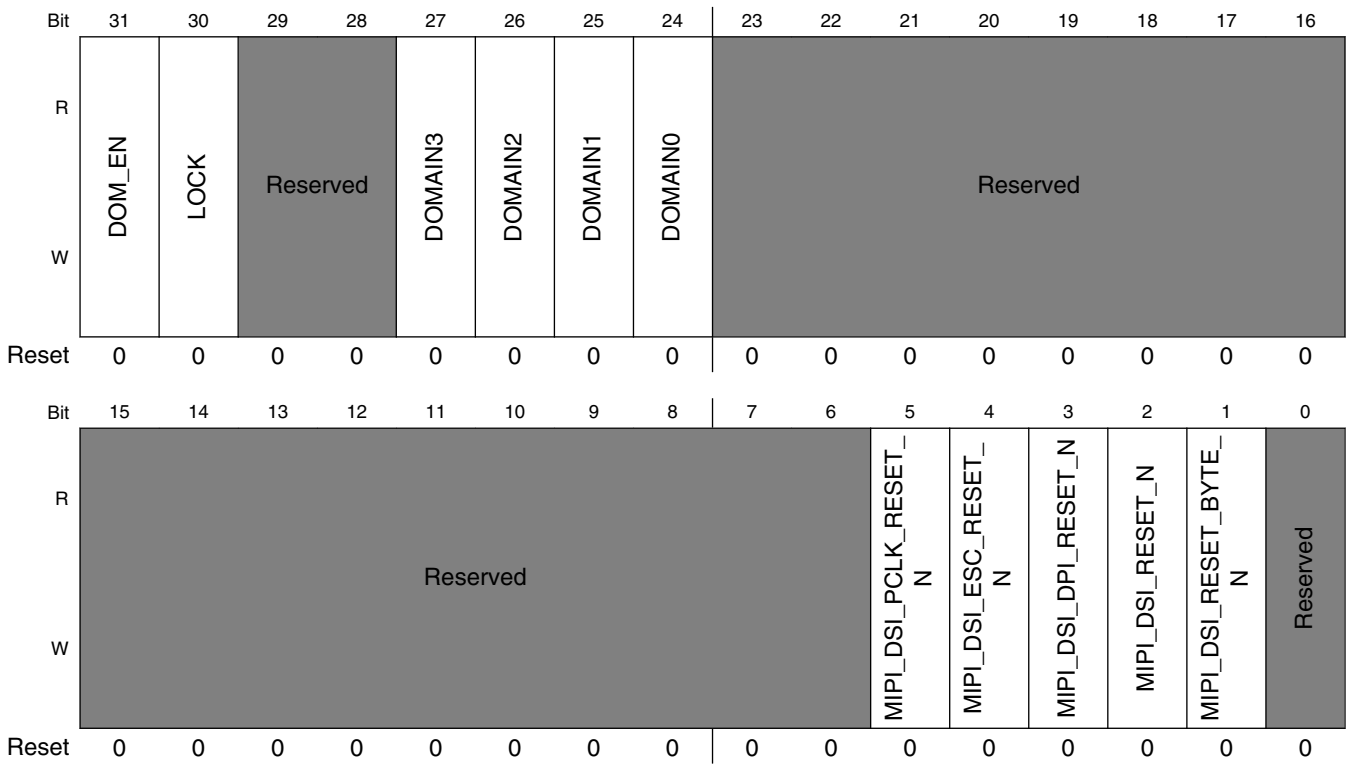
SRC_USBOPHY2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 OTG2_PHY_RESET	0 Don't reset USB OTG2 PHY 1 Reset USB OTG2 PHY

6.5.6.7 MIPI PHY Reset Control Register (SRC_MIPIPHY_RCR)

The MIPI PHY Reset Control Register (SRC_MIPIPHY_RCR), contains bits that control the MIPI PHYreset generation.

Address: 3039_0000h base + 28h offset = 3039_0028h



SRC_MIPIPHY_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified

Table continues on the next page...

SRC_MIPIPHY_RCR field descriptions (continued)

Field	Description
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–6 -	This field is reserved. Reserved
5 MIPI_DSI_ PCLK_RESET_N	0 Reset 1 Don't reset
4 MIPI_DSI_ESC_ RESET_N	0 Reset 1 Don't reset
3 MIPI_DSI_DPI_ RESET_N	0 Reset 1 Don't reset
2 MIPI_DSI_ RESET_N	0 Reset 1 Don't reset
1 MIPI_DSI_ RESET_BYTE_N	0 Reset 1 Don't reset
0 -	This field is reserved. Reserved

6.5.6.8 PCIE PHY Reset Control Register (SRC_PCIEPHY_RCR)

The PCIE PHY Control Register (SRC_PCIEPHY_RCR), contains bits that control the PCIE PHY reset generation.

Address: 3039_0000h base + 2Ch offset = 3039_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							PCIE_CTRL_APP_XFER_PENDING
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCIE_CTRL_APP_UNLOCK_MSG	PCIE_CTRL_SYS_INT	Reserved	PCIE_CTRL_CFG_L1_AUX	PCIE_CTRL_APPS_TURNOFF	PCIE_CTRL_APPS_PME	PCIE_CTRL_APPS_EXIT	PCIE_CTRL_APPS_ENTER	PCIE_CTRL_APPS_READY	PCIE_CTRL_APPS_EN	PCIE_CTRL_APPS_RST	PCIE_CTRL_APPS_CLK_REQ	PCIEPHY_PERST	PCIEPHY_BTN	PCIEPHY_G_RST	PCIE_PHY_POWER_ON_RESET_N
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

SRC_PCIEPHY_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified

Table continues on the next page...

SRC_PCIEPHY_RCR field descriptions (continued)

Field	Description
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–17 -	This field is reserved. Reserved
16 PCIE_CTRL_ APP_XFER_ PENDING	PCIE_CTRL_APP_XFER_PENDING
15 PCIE_CTRL_ APP_UNLOCK_ MSG	PCIE_CTRL_APP_UNLOCK_MSG
14 PCIE_CTRL_ SYS_INT	PCIE_CTRL_SYS_INT
13 -	This field is reserved.
12 PCIE_CTRL_ CFG_L1_AUX	Pcie_ctrl_cfg_l1_aux_clk_switch_core_clk_gate_en
11 PCIE_CTRL_ APPS_ TURNOFF	Pcie_ctrl_apps_pm_xmt_turnoff
10 PCIE_CTRL_ APPS_PME	Pcie_ctrl_apps_pm_xmt_pme
9 PCIE_CTRL_ APPS_EXIT	Pcie_ctrl_app_req_exit_l1
8 PCIE_CTRL_ APPS_ENTER	Pcie_ctrl_app_req_entr_l1

Table continues on the next page...

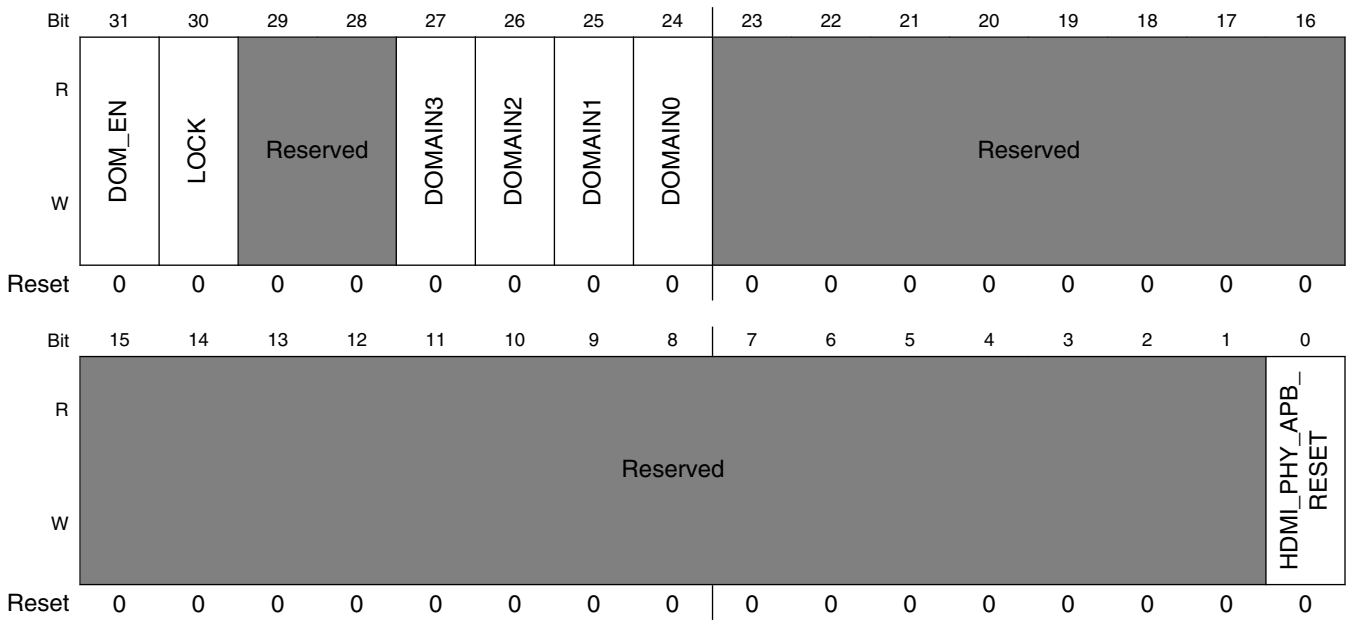
SRC_PCIEPHY_RCR field descriptions (continued)

Field	Description
7 PCIE_CTRL_ APPS_READY	Pcie_ctrl_app_ready_entr_l23
6 PCIE_CTRL_ APPS_EN	Pcie_ctrl_app_ltssm_enable
5 PCIE_CTRL_ APPS_RST	Pcie_ctrl_app_init_rst
4 PCIE_CTRL_ APPS_CLK_REQ	Pcie_ctrl_app_clk_req_n
3 PCIEPHY_ PERST	Pciephy_perst
2 PCIEPHY_BTN	PCIE PHY button
1 PCIEPHY_G_ RST	PCIE PHY Global Reset
0 PCIE_PHY_POW ER_ON_RESET_ N	PCIE_PHY_POWER_ON_RESET_N

6.5.6.9 HDMI Reset Control Register (SRC_HDMI_RCR)

The HDMI Control Register contains bits that control the HDMI reset generation.

Address: 3039_0000h base + 30h offset = 3039_0030h



SRC_HDMI_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register

Table continues on the next page...

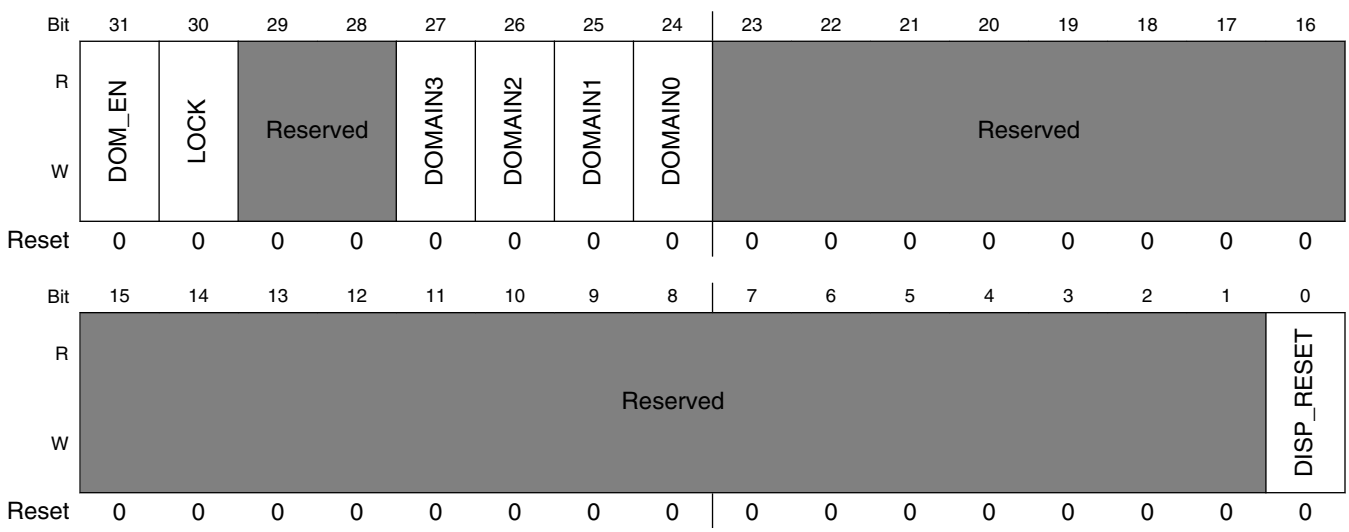
SRC_HDMI_RCR field descriptions (continued)

Field	Description
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 HDMI_PHY_ APB_RESET	Active 1

6.5.6.10 DISPLAY Reset Control Register (SRC_DISP_RCR)

The DISPLAY Control Register contains bits that control the DISPLAY reset generation.

Address: 3039_0000h base + 34h offset = 3039_0034h



SRC_DISP_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set.

Table continues on the next page...

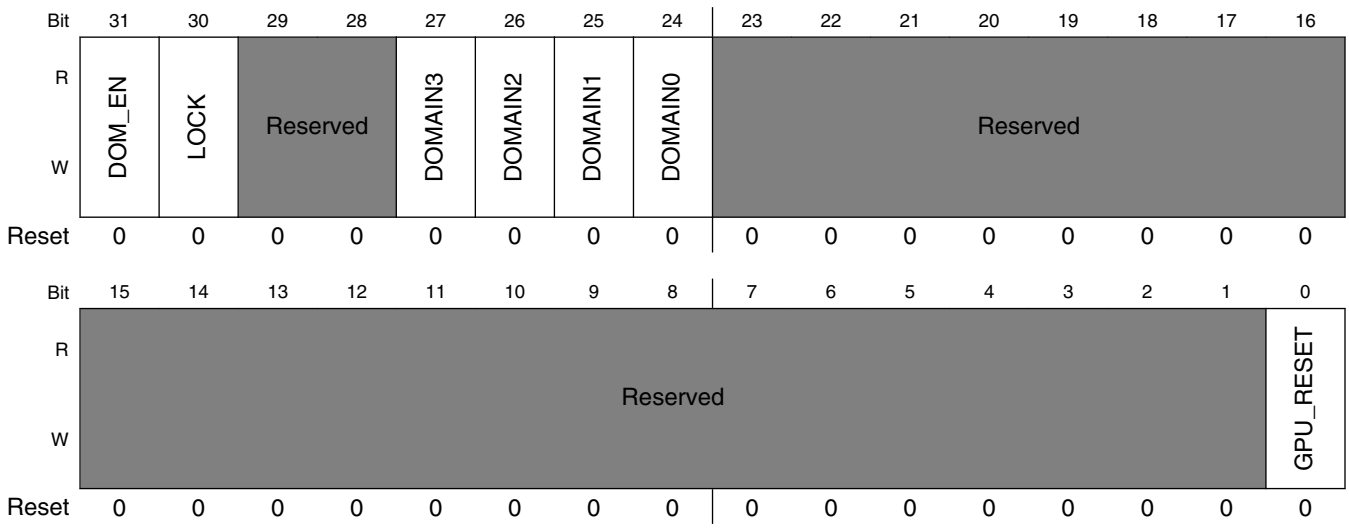
SRC_DISP_RCR field descriptions (continued)

Field	Description
	0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 DISP_RESET	0 Don't reset dispmix 1 Reset dispmix

6.5.6.11 GPU Reset Control Register (SRC_GPU_RCR)

The GPU Control Register contains bits that control the GPU reset generation.

Address: 3039_0000h base + 40h offset = 3039_0040h



SRC_GPU_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

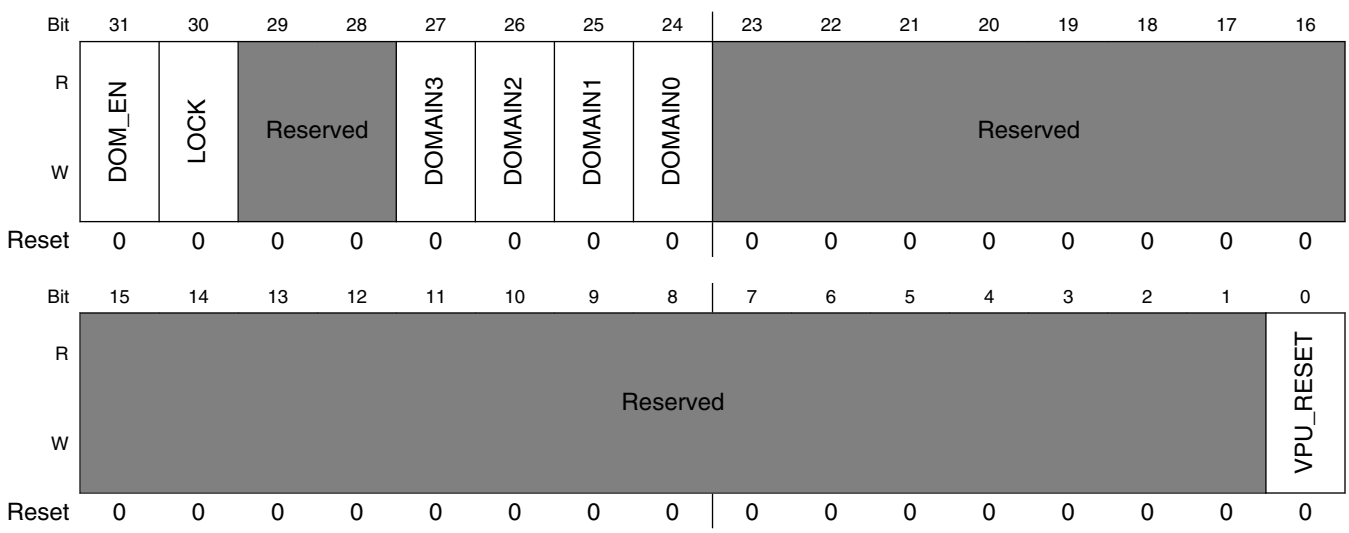
SRC_GPU_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 GPU_RESET	GPU_RESET

6.5.6.12 VPU Reset Control Register (SRC_VPU_RCR)

The VPU Control Register contains bits that control the VPU reset generation.

Address: 3039_0000h base + 44h offset = 3039_0044h



SRC_VPU_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set.

Table continues on the next page...

SRC_VPU_RCR field descriptions (continued)

Field	Description
	0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–1 -	This field is reserved. Reserved
0 VPU_RESET	VPU_RESET

6.5.6.13 PCIE2 Reset Control Register (SRC_PCIE2_RCR)

The PCIE2 Control Register contains bits that control the PCIE2 reset generation.

Address: 3039_0000h base + 48h offset = 3039_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DOM_EN	LOCK	Reserved		DOMAIN3	DOMAIN2	DOMAIN1	DOMAIN0	Reserved							PCIE_CTRL_APP_XFER_PENDING
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCIE_CTRL_APP_UNLOCK_MSG	PCIE_CTRL_SYS_INT	Reserved	PCIE_CTRL_CFG_L1_AUX	PCIE_CTRL_APPS_TURNOFF	PCIE_CTRL_APPS_PME	PCIE_CTRL_APPS_EXIT	PCIE_CTRL_APPS_ENTER	PCIE_CTRL_APPS_READY	PCIE_CTRL_APPS_EN	PCIE_CTRL_APPS_RST	PCIE_CTRL_APPS_CLK_REQ	PCIE_PERST	PCIE_BTN	PCIE_G_RST	PCIE_PHY_POWER_ON_RESET_N
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

SRC_PCIE2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved

Table continues on the next page...

SRC_PCIE2_RCR field descriptions (continued)

Field	Description
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–17 -	This field is reserved. Reserved
16 PCIE_CTRL_ APP_XFER_ PENDING	PCIE_CTRL_APP_XFER_PENDING
15 PCIE_CTRL_ APP_UNLOCK_ MSG	PCIE_CTRL_APP_UNLOCK_MSG
14 PCIE_CTRL_ SYS_INT	PCIE_CTRL_SYS_INT
13 -	This field is reserved.
12 PCIE_CTRL_ CFG_L1_AUX	Pcie_ctrl_cfg_l1_aux_clk_switch_core_clk_gate_en
11 PCIE_CTRL_ APPS_ TURNOFF	Pcie_ctrl_apps_pm_xmt_turnoff
10 PCIE_CTRL_ APPS_PME	Pcie_ctrl_apps_pm_xmt_pme
9 PCIE_CTRL_ APPS_EXIT	Pcie_ctrl_app_req_exit_l1
8 PCIE_CTRL_ APPS_ENTER	Pcie_ctrl_app_req_entr_l1

Table continues on the next page...

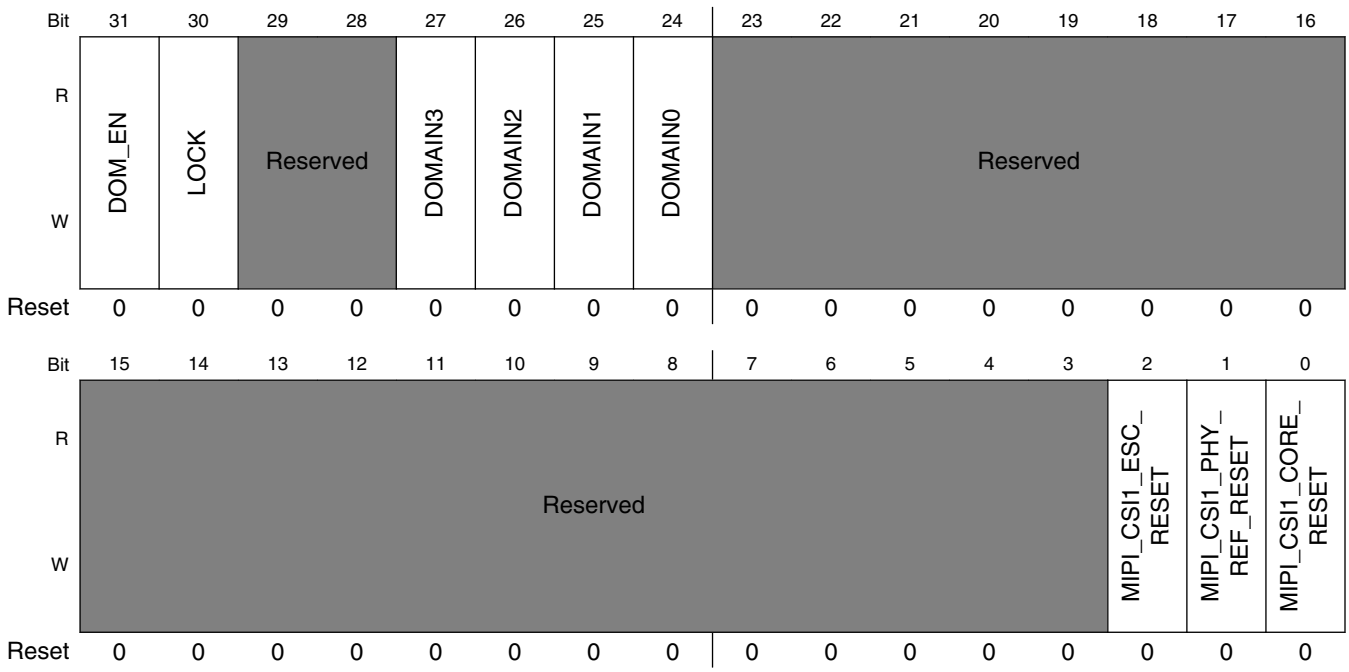
SRC_PCIE2_RCR field descriptions (continued)

Field	Description
7 PCIE_CTRL_ APPS_READY	Pcie_ctrl_app_ready_entr_l23
6 PCIE_CTRL_ APPS_EN	Pcie_ctrl_app_ltssm_enable
5 PCIE_CTRL_ APPS_RST	Pcie_ctrl_app_init_rst
4 PCIE_CTRL_ APPS_CLK_REQ	Pcie_ctrl_app_clk_req_n
3 PCIE_PERST	Pcie_perst
2 PCIE_BTN	PCIE2 button
1 PCIE_G_RST	PCIE Global Reset
0 PCIE_PHY_ POWER_ON_ RESET_N	PCIE_PHY_POWER_ON_RESET_N

6.5.6.14 MIPI CSI1 PHY Reset Control Register (SRC_MIPIPHY1_RCR)

The MIPI CSI1 PHY Control Register contains bits that control the MIPI CSI1 PHY reset generation.

Address: 3039_0000h base + 4Ch offset = 3039_004Ch



SRC_MIPIPHY1_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved

Table continues on the next page...

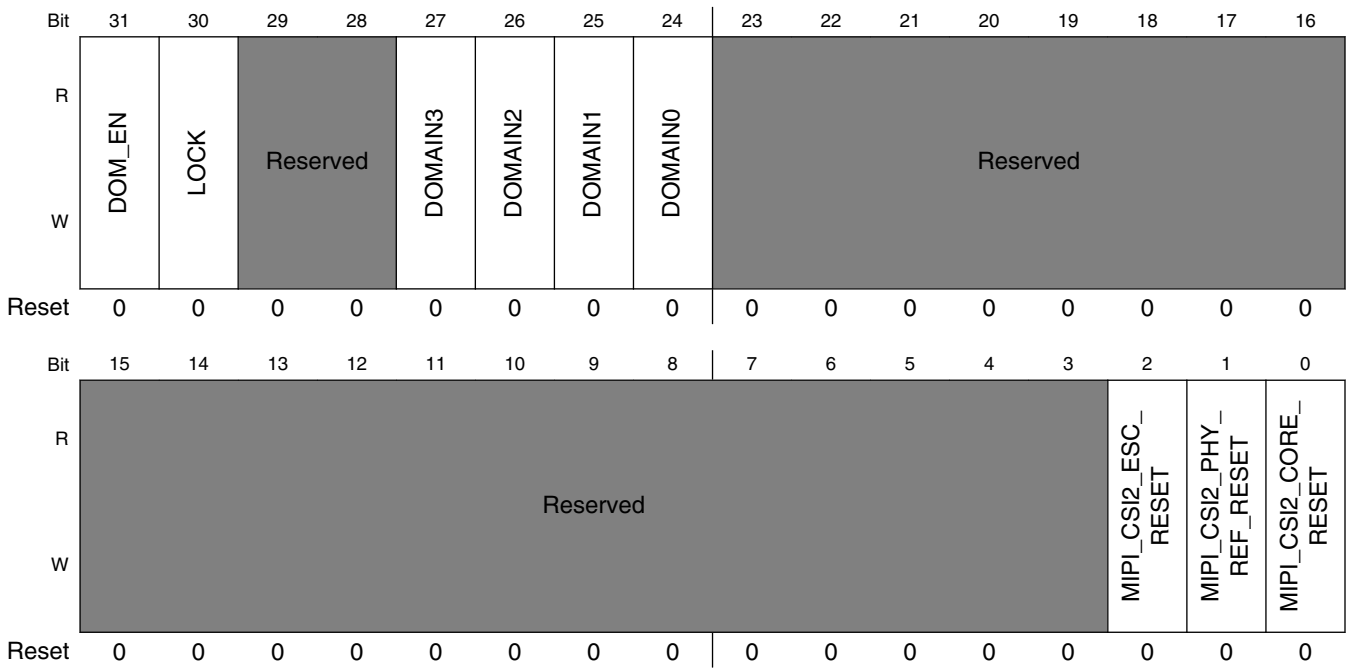
SRC_MIPIPHY1_RCR field descriptions (continued)

Field	Description
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–3 -	This field is reserved. Reserved
2 MIPI_CSI1_ ESC_RESET	MIPI_CSI1_ESC_RESET
1 MIPI_CSI1_ PHY_REF_ RESET	MIPI_CSI1_PHY_REF_RESET
0 MIPI_CSI1_ CORE_RESET	MIPI_CSI1_CORE_RESET

6.5.6.15 MIPI CSI2 PHY Reset Control Register (SRC_MIPIPHY2_RCR)

The MIPI CSI2 PHY Control Register contains bits that control the MIPI CSI2 PHY reset generation.

Address: 3039_0000h base + 50h offset = 3039_0050h



SRC_MIPIPHY2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved

Table continues on the next page...

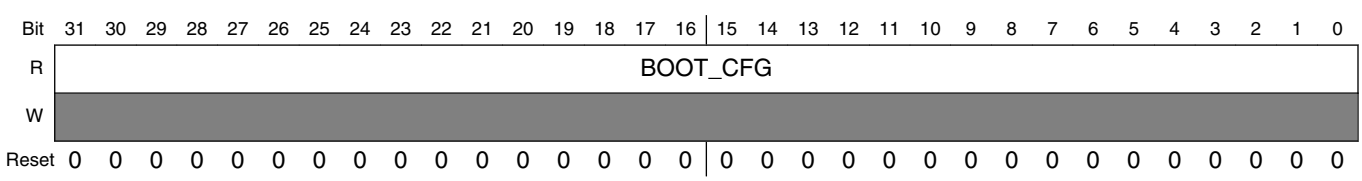
SRC_MIPIPHY2_RCR field descriptions (continued)

Field	Description
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–3 -	This field is reserved. Reserved
2 MIPI_CSI2_ ESC_RESET	MIPI_CSI2_ESC_RESET
1 MIPI_CSI2_ PHY_REF_ RESET	MIPI_CSI2_PHY_REF_RESET
0 MIPI_CSI2_ CORE_RESET	MIPI_CSI2_CORE_RESET

6.5.6.16 SRC Boot Mode Register 1 (SRC_SBMR1)

The Boot Mode register (SBMR) contains bits that reflect the status of Boot Mode Pins of the chip. The reset value is configuration dependent (depending on boot/fuses/IO pads).

Address: 3039_0000h base + 58h offset = 3039_0058h



SRC_SBMR1 field descriptions

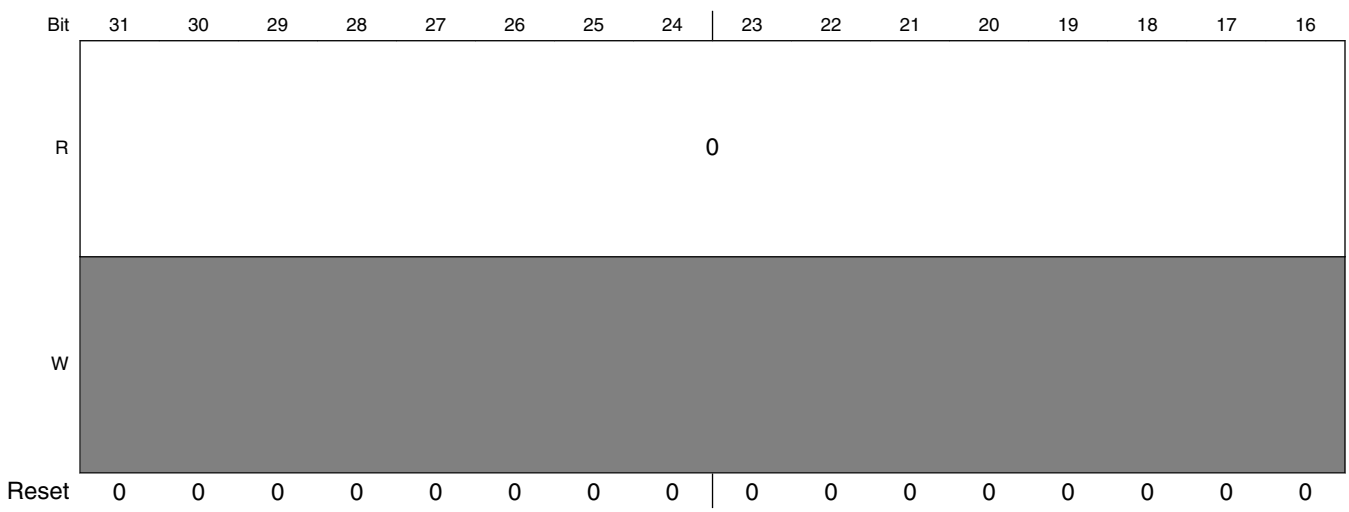
Field	Description
BOOT_CFG	Refer to fusemap.

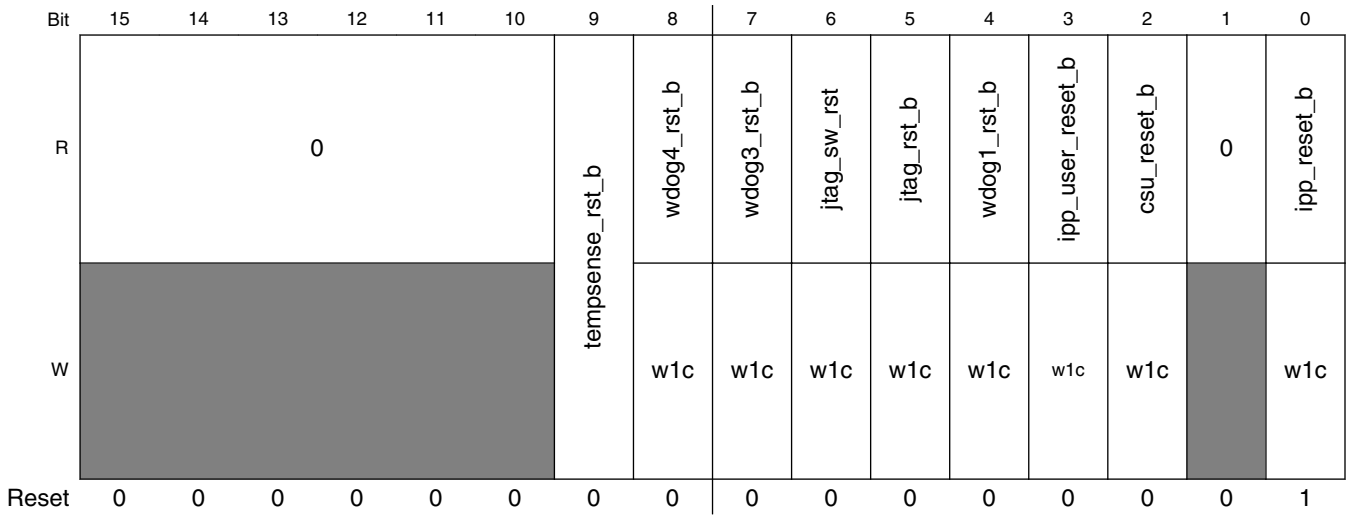
6.5.6.17 SRC Reset Status Register (SRC_SRSR)

The SRSR is a write to one clear register which records the source of the reset events for the chip. The SRC reset status register will capture all the reset sources that have occurred. This register is reset on ipp_reset_b. This is a read-write register.

For bit[9-0] - writing zero does not have any effect. Writing one will clear the corresponding bit. The individual bits can be cleared by writing one to that bit. When the system comes out of reset, this register will have bits set corresponding to all the reset sources that occurred during system reset. Software has to take care to clear this register by writing one after every reset that occurs so that the register will contain the information of recently occurred reset.

Address: 3039_0000h base + 5Ch offset = 3039_005Ch





SRC_SRSR field descriptions

Field	Description
31–10 Reserved	This read-only field is reserved and always has the value 0.
9 tempsense_rst_b	Temper Sensor software reset. Indicates whether the reset was the result of software reset from on-chip Temperature Sensor. 0 Reset is not a result of software reset from Temperature Sensor. 1 Reset is a result of software reset from Temperature Sensor.
8 wdog4_rst_b	IC Watchdog4 Time-out reset. Indicates whether the reset was the result of the watchdog4 time-out event. 0 Reset is not a result of the watchdog4 time-out event. 1 Reset is a result of the watchdog4 time-out event.
7 wdog3_rst_b	IC Watchdog3 Time-out reset. Indicates whether the reset was the result of the watchdog3 time-out event. 0 Reset is not a result of the watchdog3 time-out event. 1 Reset is a result of the watchdog3 time-out event.
6 jtag_sw_rst	JTAG software reset. Indicates whether the reset was the result of software reset from JTAG. 0 Reset is not a result of software reset from JTAG. 1 Reset is a result of software reset from JTAG.
5 jtag_rst_b	HIGH - Z JTAG reset. Indicates whether the reset was the result of HIGH-Z reset from JTAG. 0 Reset is not a result of HIGH-Z reset from JTAG. 1 Reset is a result of HIGH-Z reset from JTAG.
4 wdog1_rst_b	IC Watchdog1 Time-out reset. Indicates whether the reset was the result of the watchdog time-out event. 0 Reset is not a result of the watchdog1 time-out event. 1 Reset is a result of the watchdog1 time-out event.
3 ipp_user_reset_b	Indicates whether the reset was the result of the ipp_user_reset_b qualified reset. 0 Reset is not a result of the ipp_user_reset_b qualified as COLD reset event. 1 Reset is a result of the ipp_user_reset_b qualified as COLD reset event.

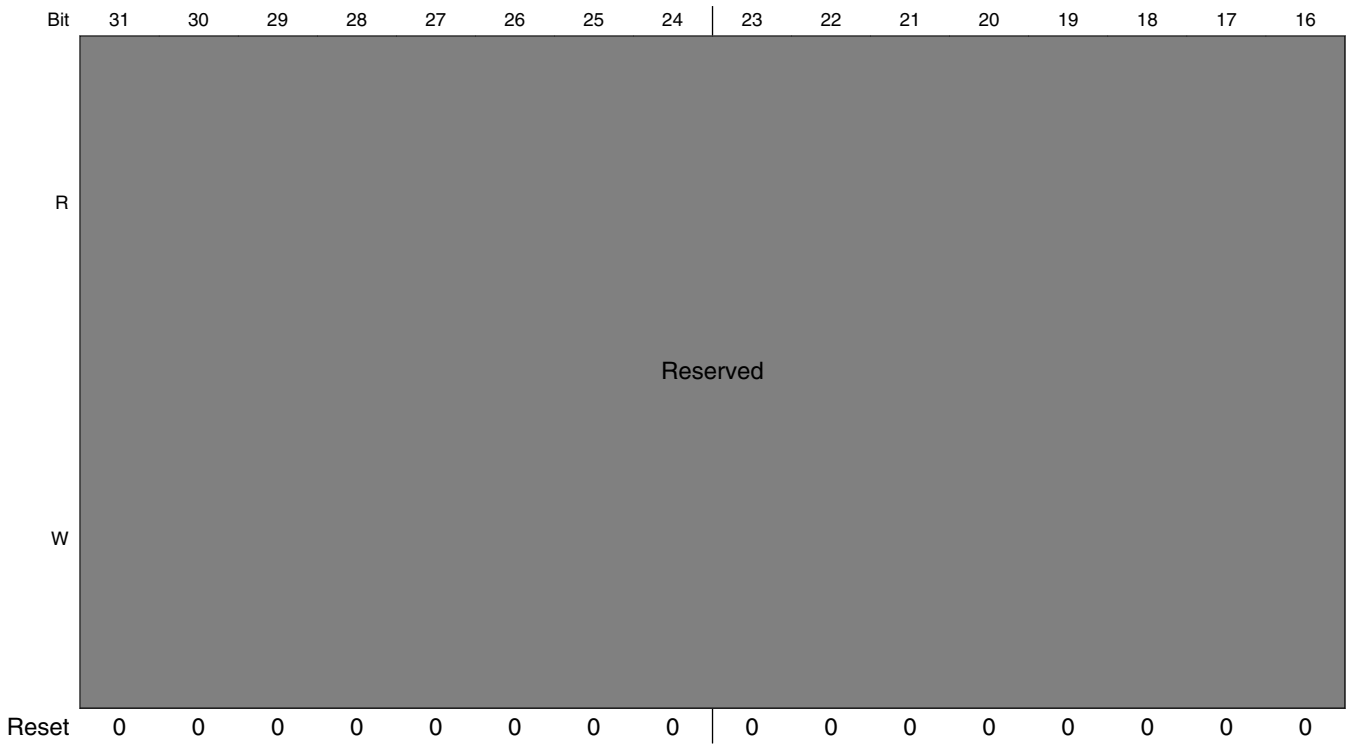
Table continues on the next page...

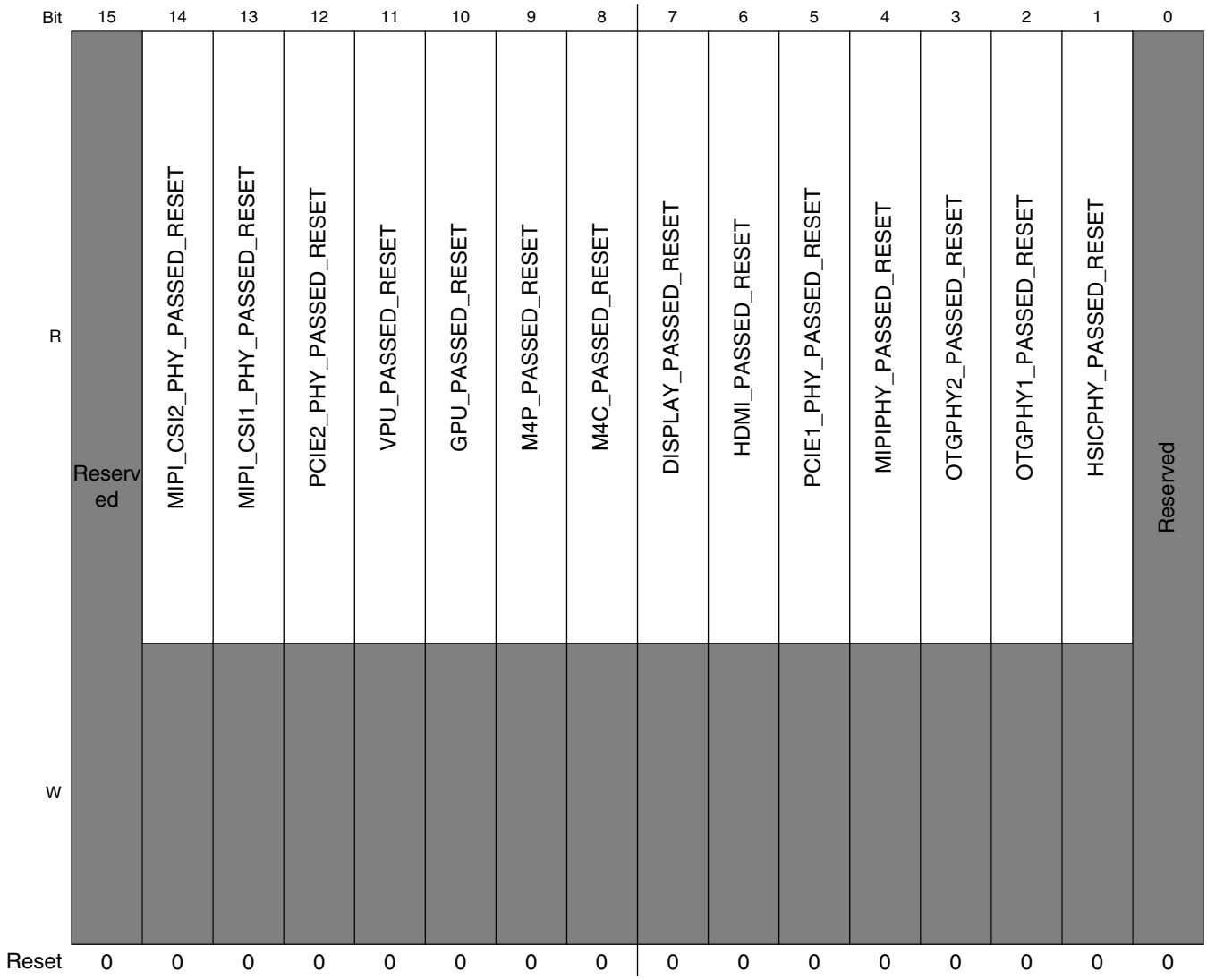
SRC_SRSR field descriptions (continued)

Field	Description
2 csu_reset_b	Indicates whether the reset was the result of the csu_reset_b input. **NOTE:** If case the csu_reset_b occurred during a WARM reset process, during the phase that ipg_clk is not available yet, then the occurrence of CSU reset will not be reflected in this bit. 0 Reset is not a result of the csu_reset_b event. 1 Reset is a result of the csu_reset_b event.
1 Reserved	This read-only field is reserved and always has the value 0.
0 ipp_reset_b	Indicates whether reset was the result of ipp_reset_b pin (Power-up sequence) 0 Reset is not a result of ipp_reset_b pin. 1 Reset is a result of ipp_reset_b pin.

6.5.6.18 SRC Interrupt Status Register (SRC_SISR)

Address: 3039_0000h base + 68h offset = 3039_0068h





SRC_SISR field descriptions

Field	Description
31–15 -	This field is reserved. Reserved
14 MIPI_CSI2_ PHY_PASSED_ RESET	Interrupt generated to indicate that MIPI CSI2 PHY passed software reset and is ready to be used 0 interrupt generated not due to MIPI CSI2 PHY reset 1 interrupt generated due to MIPI CSI2 PHY reset
13 MIPI_CSI1_ PHY_PASSED_ RESET	Interrupt generated to indicate that MIPI CSI1 PHY passed software reset and is ready to be used 0 interrupt generated not due to MIPI CSI1 PHY reset 1 interrupt generated due to MIPI CSI1 PHY reset
12 PCIE2_PHY_ PASSED_ RESET	Interrupt generated to indicate that PCIE2 PHY passed software reset and is ready to be used 0 interrupt generated not due to PCIE2 PHY reset 1 interrupt generated due to PCIE2 PHY reset

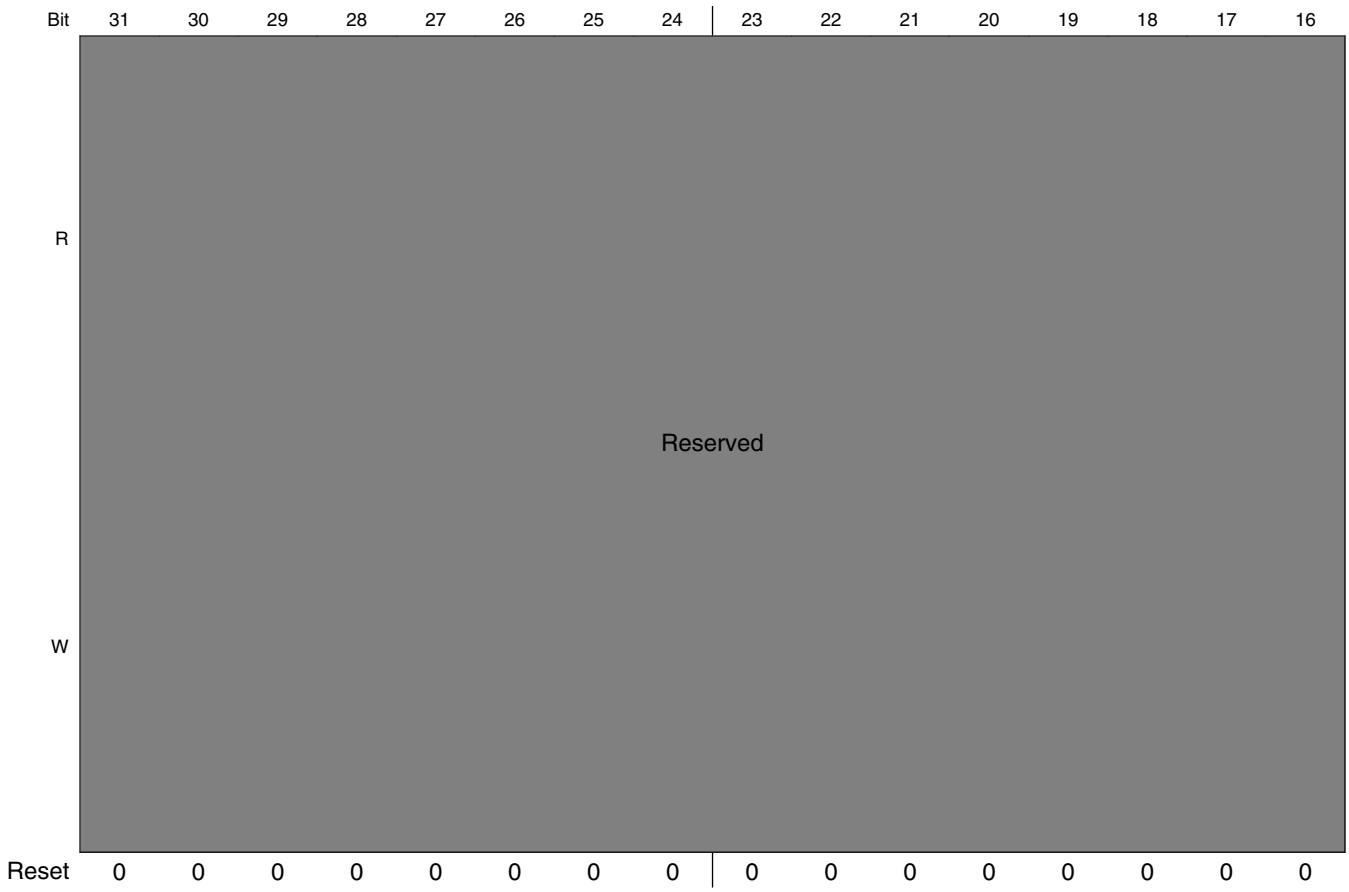
Table continues on the next page...

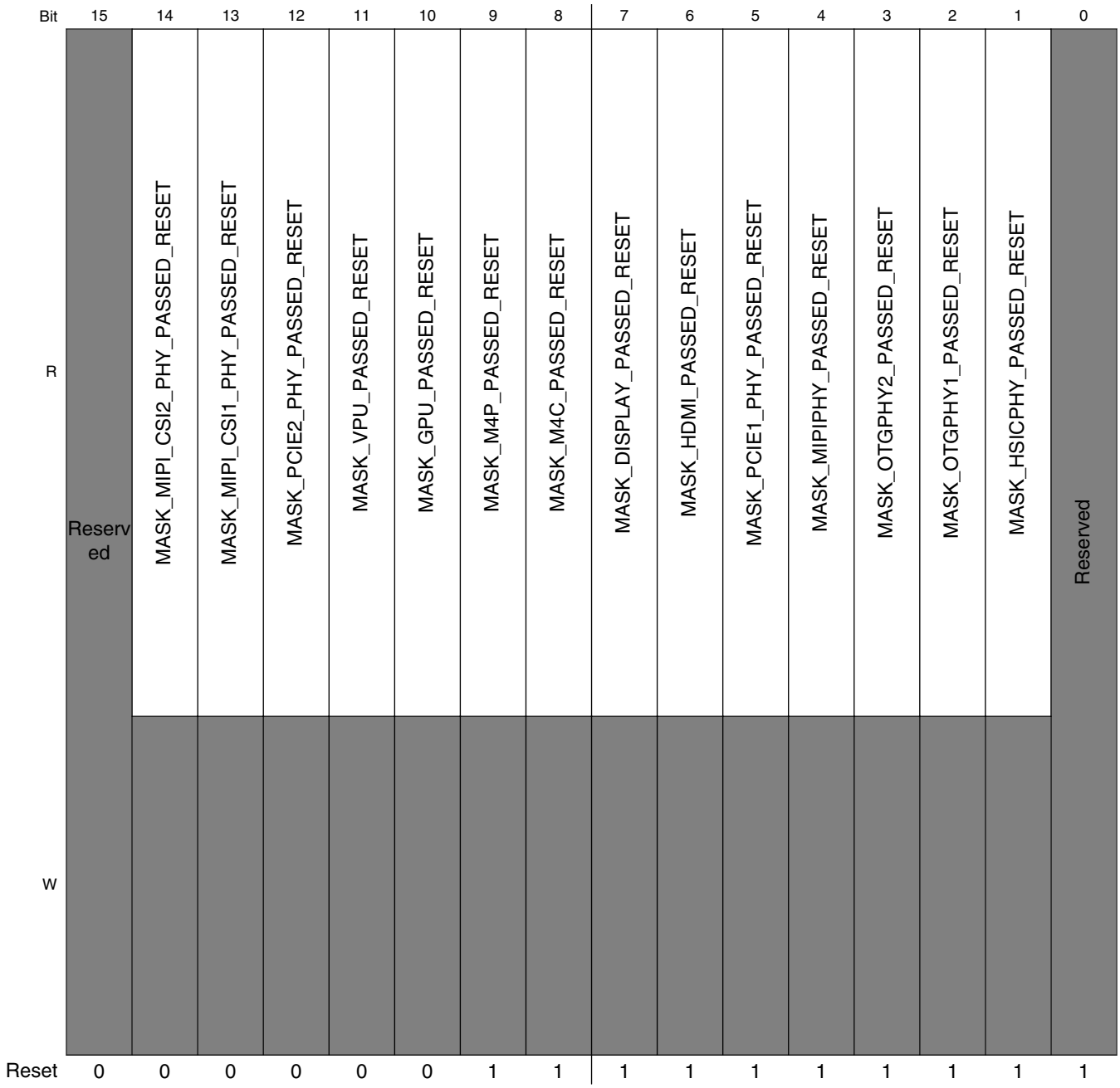
SRC_SISR field descriptions (continued)

Field	Description
11 VPU_PASSED_ RESET	Interrupt generated to indicate that VPU passed software reset and is ready to be used 0 interrupt generated not due to VPU reset 1 interrupt generated due to VPU reset
10 GPU_PASSED_ RESET	Interrupt generated to indicate that GPU passed software reset and is ready to be used 0 interrupt generated not due to GPU reset 1 interrupt generated due to GPU reset
9 M4P_PASSED_ RESET	Interrupt generated to indicate that m4 platform passed software reset and is ready to be used 0 interrupt generated not due to m4 platform reset 1 interrupt generated due to m4 platform reset
8 M4C_PASSED_ RESET	Interrupt generated to indicate that m4 core passed software reset and is ready to be used 0 interrupt generated not due to m4 core reset 1 interrupt generated due to m4 core reset
7 DISPLAY_ PASSED_ RESET	Interrupt generated to indicate that DISPLAY passed software reset and is ready to be used 0 Interrupt generated not due to DISPLAY passed reset 1 Interrupt generated due to DISPLAY passed reset
6 HDMI_PASSED_ RESET	Interrupt generated to indicate that HDMI passed software reset and is ready to be used 0 Interrupt generated not due to HDMI passed reset 1 Interrupt generated due to HDMI passed reset
5 PCIE1_PHY_ PASSED_ RESET	Interrupt generated to indicate that PCIE1 PHY passed software reset and is ready to be used 0 Interrupt generated not due to PCIE1 PHY passed reset 1 Interrupt generated due to PCIE1 PHY passed reset
4 MIPIPHY_ PASSED_ RESET	Interrupt generated to indicate that MIPI PHY passed software reset and is ready to be used 0 Interrupt generated not due to MIPI PHY passed reset 1 Interrupt generated due to MIPI PHY passed reset
3 OTGPHY2_ PASSED_ RESET	Interrupt generated to indicate that OTG PHY2 passed software reset and is ready to be used 0 Interrupt generated not due to OTG PHY2 passed reset 1 Interrupt generated due to OTG PHY2 passed reset
2 OTGPHY1_ PASSED_ RESET	Interrupt generated to indicate that OTG PHY1 passed software reset and is ready to be used 0 Interrupt generated not due to OTG PHY1 passed reset 1 Interrupt generated due to OTG PHY1 passed reset
1 HSICPHY_ PASSED_ RESET	Interrupt generated to indicate that HSIC PHY passed software reset and is ready to be used 0 Interrupt generated not due to HSIC PHY passed reset 1 Interrupt generated due to HSIC PHY passed reset
0 -	This field is reserved. Reserved

6.5.6.19 SRC Interrupt Mask Register (SRC_SIMR)

Address: 3039_0000h base + 6Ch offset = 3039_006Ch





SRC_SIMR field descriptions

Field	Description
31–15 -	This field is reserved. Reserved
14 MASK_MIPI_CSI2_PHY_PASSED_RESET	Mask interrupt generation due to MIPI CSI2 PHY passed reset 0 do not mask interrupt due to MIPI CSI2 PHY passed reset - interrupt will be created 1 mask interrupt due to MIPI CSI2 PHY passed reset

Table continues on the next page...

SRC_SIMR field descriptions (continued)

Field	Description
13 MASK_MIPI_ CSI1_PHY_ PASSED_ RESET	Mask interrupt generation due to MIPI CSI1 PHY passed reset 0 do not mask interrupt due to MIPI CSI1 PHY passed reset - interrupt will be created 1 mask interrupt due to MIPI CSI1 PHY passed reset
12 MASK_PCIE2_ PHY_PASSED_ RESET	Mask interrupt generation due to PCIE2 PHY passed reset 0 do not mask interrupt due to PCIE2 PHY passed reset - interrupt will be created 1 mask interrupt due to PCIE2 PHY passed reset
11 MASK_VPU_ PASSED_ RESET	Mask interrupt generation due to VPU passed reset 0 do not mask interrupt due to VPU passed reset - interrupt will be created 1 mask interrupt due to VPU passed reset
10 MASK_GPU_ PASSED_ RESET	Mask interrupt generation due to GPU passed reset 0 do not mask interrupt due to GPU passed reset - interrupt will be created 1 mask interrupt due to GPU passed reset
9 MASK_M4P_ PASSED_ RESET	mask interrupt generation due to m4 platform passed reset 0 do not mask interrupt due to m4 platform passed reset - interrupt will be created 1 mask interrupt due to m4platform passed reset
8 MASK_M4C_ PASSED_ RESET	mask interrupt generation due to m4 core passed reset 0 do not mask interrupt due to m4 core passed reset - interrupt will be created 1 mask interrupt due to m4 core passed reset
7 MASK_ DISPLAY_ PASSED_ RESET	Mask interrupt generation due to display passed reset 0 do not mask interrupt due to display passed reset - interrupt will be created 1 mask interrupt due to display passed reset
6 MASK_HDMI_ PASSED_ RESET	Mask interrupt generation due to HDMI passed reset 0 do not mask interrupt due to HDMI passed reset - interrupt will be created 1 mask interrupt due to HDMI passed reset
5 MASK_PCIE1_ PHY_PASSED_ RESET	Mask interrupt generation due to PCIE1 PHY passed reset 0 do not mask interrupt due to PCIE1 PHY passed reset - interrupt will be created 1 mask interrupt due to PCIE1 PHY passed reset
4 MASK_ MIPIPHY_ PASSED_ RESET	mask interrupt generation due to MIPI PHY passed reset 0 do not mask interrupt due to MIPI PHY passed reset - interrupt will be created 1 mask interrupt due to MIPI PHY passed reset
3 MASK_ OTGPHY2_ PASSED_ RESET	mask interrupt generation due to OTG PHY2 passed reset 0 do not mask interrupt due to OTG PHY2 passed reset - interrupt will be created 1 mask interrupt due to OTG PHY2 passed reset

Table continues on the next page...

SRC_SIMR field descriptions (continued)

Field	Description
2 MASK_ OTGPHY1_ PASSED_ RESET	mask interrupt generation due to OTG PHY1 passed reset 0 do not mask interrupt due to OTG PHY1 passed reset - interrupt will be created 1 mask interrupt due to OTG PHY1 passed reset
1 MASK_ HSICPHY_ PASSED_ RESET	mask interrupt generation due to HSIC PHY passed reset 0 do not mask interrupt due to HSIC PHY passed reset - interrupt will be created 1 mask interrupt due to HSIC PHY passed reset
0 -	This field is reserved. Reserved

6.5.6.20 SRC Boot Mode Register 2 (SRC_SBMR2)

The Boot Mode register (SBMR), contains bits that reflect the status of Boot Mode Pins of the chip. The default values for those bits depends on the values of pins/fuses during reset sequence.

Address: 3039_0000h base + 70h offset = 3039_0070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0						BMOD		0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										FUSE_FORCE_COLD_BOOT	BT_FUSE_SEL	DIR_BT_DIS	0	SEC_ CONFIG[1:0]	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_SBMR2 field descriptions

Field	Description
31–26 Reserved	This read-only field is reserved and always has the value 0.
25–24 BMOD	BMOD shows the latched state of the BOOT_MODE1 and BOOT_MODE0 signals on the rising edge of POR_B. See the Boot mode pin settings section of System Boot.
23–6 Reserved	This read-only field is reserved and always has the value 0.
5 FUSE_FORCE_COLD_BOOT	See Fusemap for additional information.
4 BT_FUSE_SEL	BT_FUSE_SEL (connected to gpio bt_fuse_sel) shows the state of the BT_FUSE_SEL fuse. See Fusemap for additional information on this fuse.
3 DIR_BT_DIS	DIR_BT_DIS shows the state of the DIR_BT_DIS fuse. See the fusemap for additional information on this fuse.
2 Reserved	This read-only field is reserved and always has the value 0.
SEC_CONFIG[1:0]	SEC_CONFIG[1] shows the state of the SEC_CONFIG[1] fuse. See Fusemap for additional information on this fuse. SEC_CONFIG[0] shows the state of the SEC_CONFIG[0] fuse. This fuse is shown as reserved in Fusemap (address 0x440[1]) because it does not have a user-relevant function.

6.5.6.21 SRC General Purpose Register 1 (SRC_GPR1)

Address: 3039_0000h base + 74h offset = 3039_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ENTRY0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR1 field descriptions

Field	Description
PERSISTENT_ENTRY0	Holds entry function for core0 for waking-up from low power mode. The SRC ensures that the register value will persist across system resets.

6.5.6.22 SRC General Purpose Register 2 (SRC_GPR2)

Address: 3039_0000h base + 78h offset = 3039_0078h

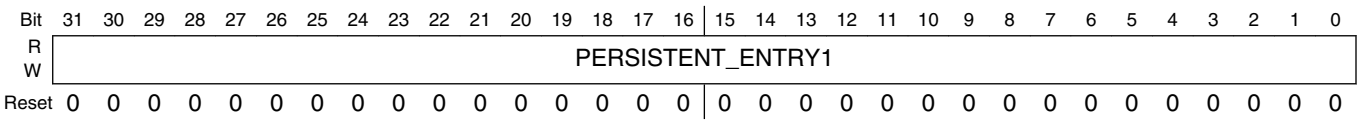
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ARG0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR2 field descriptions

Field	Description
PERSISTENT_ARG0	Holds argument of entry function for core0 for waking-up from low power mode. The SRC ensures that the register value will persist across system resets.

6.5.6.23 SRC General Purpose Register 3 (SRC_GPR3)

Address: 3039_0000h base + 7Ch offset = 3039_007Ch

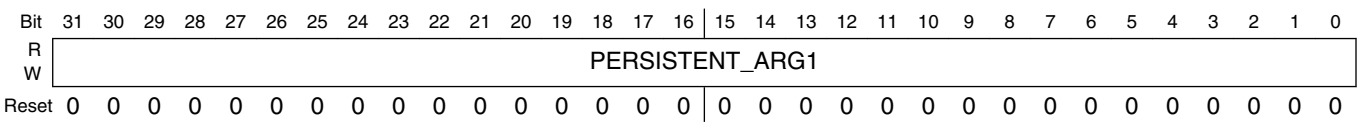


SRC_GPR3 field descriptions

Field	Description
PERSISTENT_ENTRY1	Holds entry function for core1. The SRC ensures that the register value will persist across system resets.

6.5.6.24 SRC General Purpose Register 4 (SRC_GPR4)

Address: 3039_0000h base + 80h offset = 3039_0080h

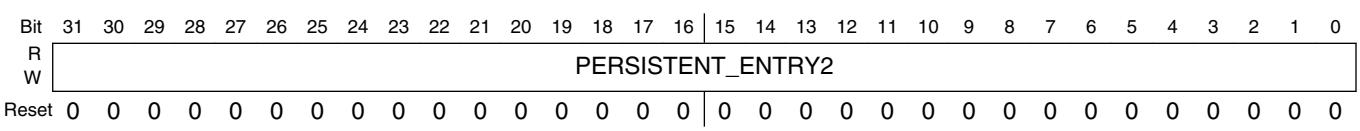


SRC_GPR4 field descriptions

Field	Description
PERSISTENT_ARG1	Holds argument of entry function for core1. The SRC ensures that the register value will persist across system resets.

6.5.6.25 SRC General Purpose Register 5 (SRC_GPR5)

Address: 3039_0000h base + 84h offset = 3039_0084h



SRC_GPR5 field descriptions

Field	Description
PERSISTENT_ ENTRY2	Holds entry function for core2. The SRC ensures that the register value will persist across system resets.

6.5.6.26 SRC General Purpose Register 6 (SRC_GPR6)

Address: 3039_0000h base + 88h offset = 3039_0088h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ARG2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR6 field descriptions

Field	Description
PERSISTENT_ ARG2	Holds argument of entry function for core2. The SRC ensures that the register value will persist across system resets.

6.5.6.27 SRC General Purpose Register 7 (SRC_GPR7)

Address: 3039_0000h base + 8Ch offset = 3039_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ENTRY3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SRC_GPR7 field descriptions

Field	Description
PERSISTENT_ ENTRY3	Holds entry function for core3. The SRC ensures that the register value will persist across system resets.

6.5.6.28 SRC General Purpose Register 8 (SRC_GPR8)

Address: 3039_0000h base + 90h offset = 3039_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERSISTENT_ARG3																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

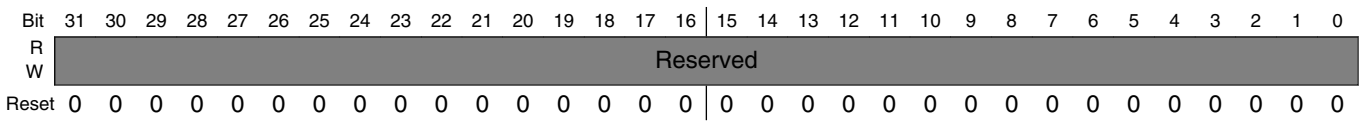
SRC_GPR8 field descriptions

Field	Description
PERSISTENT_ARG3	Holds argument of entry function for core3. The SRC ensures that the register value will persist across system resets.

6.5.6.29 SRC General Purpose Register 9 (SRC_GPR9)

Reserved for Internal Use.

Address: 3039_0000h base + 94h offset = 3039_0094h

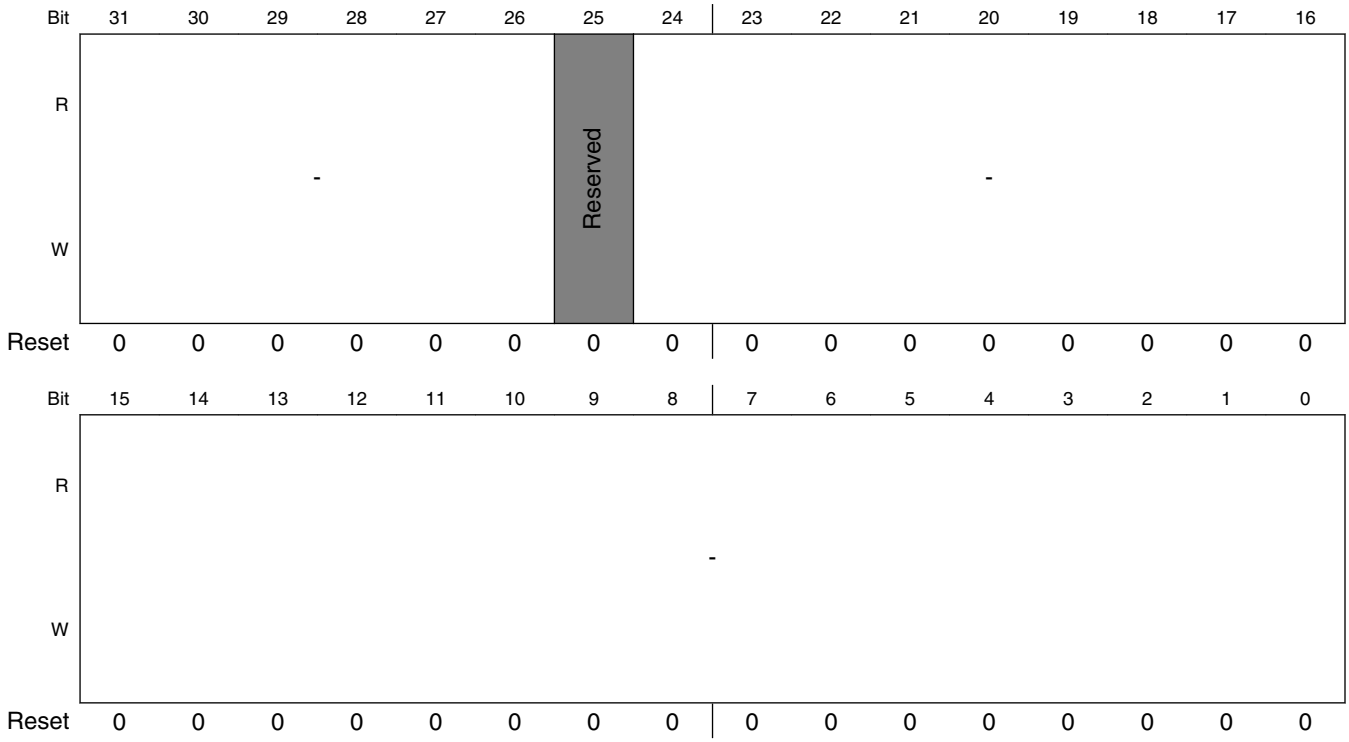


SRC_GPR9 field descriptions

Field	Description
-	This field is reserved. Reserved.

6.5.6.30 SRC General Purpose Register 10 (SRC_GPR10)

Address: 3039_0000h base + 98h offset = 3039_0098h



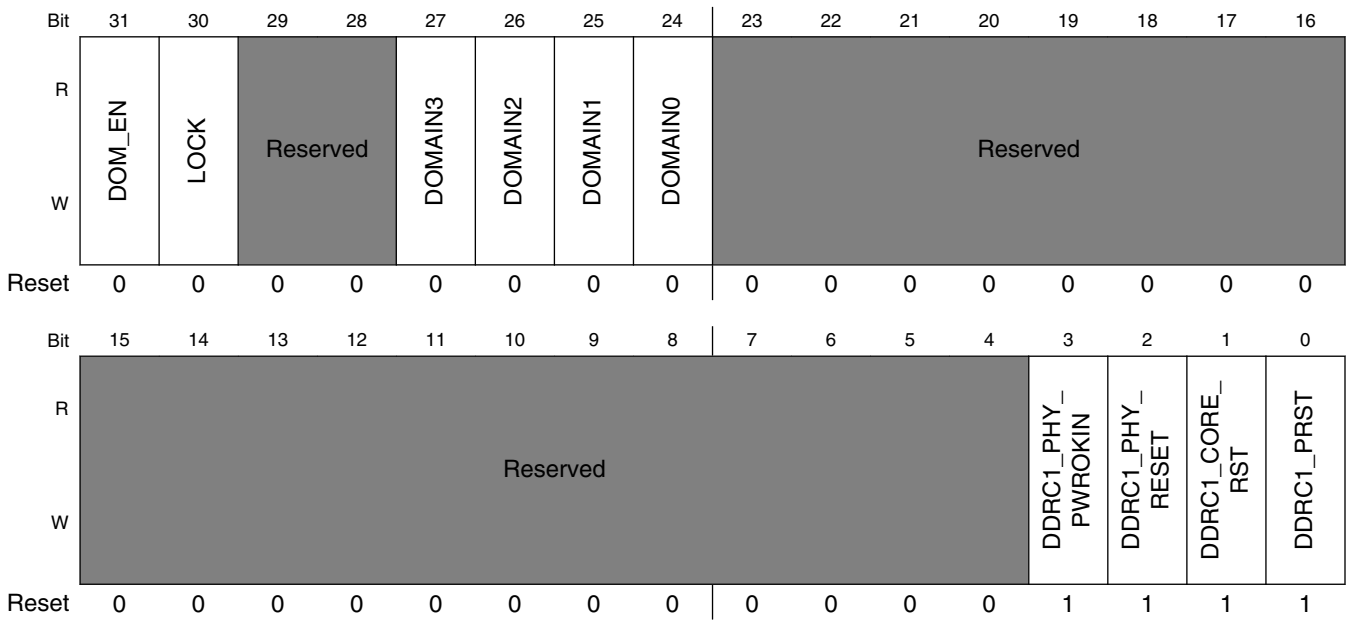
SRC_GPR10 field descriptions

Field	Description
31–26 -	Read/write bits, for general purpose **NOTE:** Reset only by POR
25 -	This field is reserved. Reserved.
-	Read/write bits, for general purpose **NOTE:** Reset only by POR

6.5.6.31 SRC DDR Controller Reset Control Register (SRC_DDRC_RCR)

DDR Controller Reset Control Register

Address: 3039_0000h base + 1000h offset = 3039_1000h



SRC_DDRC_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

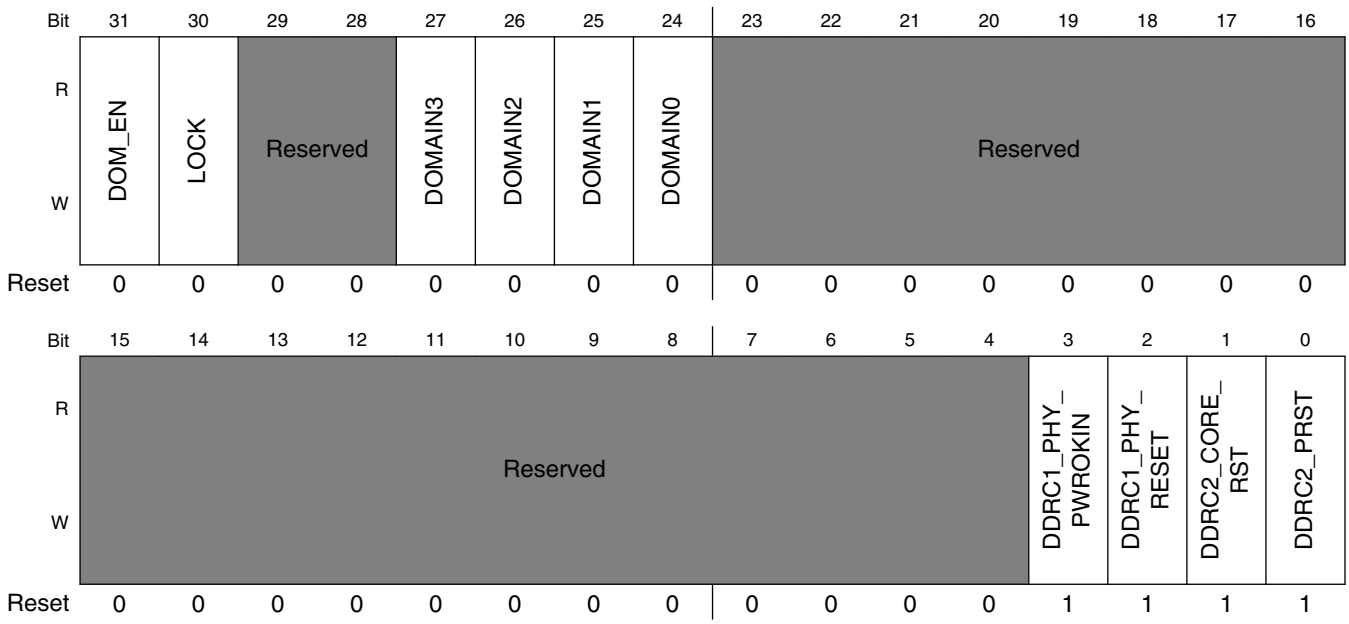
SRC_DDRC_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain0 cannot write to this register. 1 This register is assigned to domain0. The master from domain0 can write to this register
23–4 -	This field is reserved. Reserved
3 DDRC1_PHY_PWROKIN	0 De-assert DDR controller 1 Assert DDR Controller
2 DDRC1_PHY_RESET	0 De-assert DDR controller 1 Assert DDR Controller
1 DDRC1_CORE_RST	DDR Controller core_ddrc_rstn and aresetn. This bit is set to 1 automatically when fastmix is powered off. 0 De-assert DDR controller aresetn and core_ddrc_rstn 1 Assert DDR Controller preset and DDR PHY reset
0 DDRC1_PRST	DDR Controller preset and DDR PHY reset. This bit is set to 1 automatically when fastmix is powered off. **NOTE:** This reset can only be released when DDR Controller clock inputs are active and stable for 30 cycles 0 De-assert DDR Controller preset and DDR PHY reset reset 1 Assert DDR Controller preset and DDR PHY reset

6.5.6.32 SRC DDRC2 Controller Reset Control Register (SRC_DDRC2_RCR)

DDRC2 Controller Reset Control Register

Address: 3039_0000h base + 1004h offset = 3039_1004h



SRC_DDRC2_RCR field descriptions

Field	Description
31 DOM_EN	Domain Control enable for this register. **NOTE:** [31:30] and [27:24] areas are not controlled by this bit. [31:30] and [27:24] area can be modified by any masters from any domains when Lock bit is not set. 0 Disables domain control. All of this register's bits except [31:30] and [27:24] can be modified by any masters 1 Enables domain control. All of this register's bits except [31:30] and [27:24] can only be modified by the masters from the domains specified in [27:24] area.
30 LOCK	Domain control bits lock **NOTE:** Lock bit is a write-once register, once it is set to 1, it can't be write to 0 0 [31] and [27:24] bits can be modified 1 [31] and [27:24] bits cannot be modified
29–28 -	This field is reserved. Reserved
27 DOMAIN3	Domain3 assignment control. Effective when dom_en is set to 1.

Table continues on the next page...

SRC_DDRC2_RCR field descriptions (continued)

Field	Description
	0 This register is not assigned to domain3. The master from domain3 cannot write to this register. 1 This register is assigned to domain3. The master from domain3 can write to this register
26 DOMAIN2	Domain2 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain2. The master from domain2 cannot write to this register. 1 This register is assigned to domain2. The master from domain2 can write to this register
25 DOMAIN1	Domain1 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain1. The master from domain1 cannot write to this register. 1 This register is assigned to domain1. The master from domain1 can write to this register
24 DOMAIN0	Domain0 assignment control. Effective when dom_en is set to 1. 0 This register is not assigned to domain0. The master from domain3 cannot write to this register. 1 This register is assigned to domain0. The master from domain3 can write to this register
23–4 -	This field is reserved. Reserved
3 DDRC1_PHY_PWROKIN	0 De-assert DDR controller 1 Assert DDR Controller
2 DDRC1_PHY_RESET	0 De-assert DDR controller 1 Assert DDR Controller
1 DDRC2_CORE_RST	DDRC2 Controller core_ddrc_rstn and aresetn. This bit is set to 1 automatically when fastmix is powered off. 0 De-assert DDR controller aresetn and core_ddrc_rstn 1 Assert DDR Controller preset and DDR PHY reset
0 DDRC2_PRST	DDRC2 Controller preset and DDR PHY reset. This bit is set to 1 automatically when fastmix is powered off. This bit is used as src_system_rst_b of DDR1. **NOTE:** This reset can only be released when DDR Controller clock inputs are active and stable for 30 cycles 0 De-assert DDRC2 Controller preset and DDR PHY reset reset 1 Assert DDRC2 Controller preset and DDR PHY reset

Chapter 7
Interrupts and DMA

7.1 Interrupts and DMA Events

7.1.1 Overview

This chapter provides the interrupt assignments of the ARM domain in A53 Interrupts, CM4 Interrupts, and the DMA events in SDMA event mapping

7.1.2 A53 Interrupts

The Global Interrupt Controller (GIC) collects up to 128 interrupt requests from all the chip sources and provides an interface to the Cortex A53 CPU.

Each interrupt can be configured as a normal or a secure interrupt. Software force registers and software priority masking are also supported. The following table describes the A53 interrupt sources.

Table 7-1. ARM Domain Interrupt Summary

IRQ	Module	Logic	Interrupt Description
IRQ	MODULE	LOGIC	Description
0	boot	-	Used to notify cores on exception condition while boot
1	dap	-	DAP Interrupt
2	sdma1	-	AND of all 48 SDMA interrupts (events) from all the channels
3	gpu	-	GPU Interrupt
4	snvs_lp_wrapper	OR	ON-OFF button press shorter than 5 secs (pulse event)
4	snvs_hp_wrapper	OR	ON-OFF button press shorter than 5 secs (pulse event)
5	lcdif	-	LCDIF Interrupt

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
6	spdif1	OR	SPDIF Rx interrupt
6	spdif1	OR	SPDIF Tx interrupt
7	h264dec	-	h264 Decoder Interrupt
8	vpudma	-	VPU DMA Interrupt
9	qos	-	QOS Interrupt
10	wdog3	-	Watchdog Timer reset
11	hs	-	HS Interrupt Request
12	apbhdma	OR	GPMI operation channel 0 description complete interrupt
12	apbhdma	OR	GPMI operation channel 1 description complete interrupt
12	apbhdma	OR	GPMI operation channel 2 description complete interrupt
12	apbhdma	OR	GPMI operation channel 3 description complete interrupt
13	spdif2	OR	SPDIF Rx interrupt
13	spdif2	OR	SPDIF Tx interrupt
14	rawnand	-	BCH operation complete interrupt
15	rawnand	-	GPMI operation TIMEOUT ERROR interrupt
16	hdmi_ips	-	HDMI IRQ
17	hdmi_ips	-	HDMI IRQ
18	hdmi_ips	-	HDMI IRQ
19	snvs_hp_wrapper	-	SRTC Consolidated Interrupt. Non TZ.
20	snvs_hp_wrapper	-	SRTC Security Interrupt. TZ.
21	csu	-	CSU Interrupt Request. Indicates to the processor that one or more alarm inputs were asserted
22	usdhc1	-	uSDHC1 Enhanced SDHC Interrupt Request
23	usdhc2	-	uSDHC2 Enhanced SDHC Interrupt Request
24	dc8000_control	-	DC8000 Display Controller IRQ
25	DTRC_wrapper	-	DTRC IRQ
26	uart1	-	UART-1 ORed interrupt
27	uart2	-	UART-2 ORed interrupt
28	uart3	-	UART-3 ORed interrupt
29	uart4	-	UART-4 ORed interrupt
30	vp9dec	-	VP9 Decoder Interrupt
31	ecspi1	-	eCSPI1 interrupt request line to the core.
32	ecspi2	-	eCSPI2 interrupt request line to the core.
33	ecspi3	-	eCSPI3 interrupt request line to the core.
34	mipi_dsi	-	MIPI-DSI interrupt
35	i2c1	-	I2C-1 Interrupt

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
36	i2c2	-	I2C-2 Interrupt
37	i2c3	-	I2C-3 Interrupt
38	i2c4	-	I2C-4 Interrupt
39	rdc	-	RDC interrupt
40	usb1	-	USB-1 Interrupt
41	usb2	-	USB-2 Interrrupt
42	csi1	-	CSI-1 Interrupt
43	csi2	-	CSI-2 Interrupt
44	mipi_csi1	-	MIPI-CSI-1 Interrupt
45	mipi_csi2	-	MIPI-CSI-2 Interrupt
46	gpt6	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
47	sctr	-	System Counter Interrupt [0]
48	sctr	-	System Counter Interrupt [1]
49	anamix	OR	TempSensor (Temperature alarm).
49	anamix	OR	TempSensor (Temperature critical alarm).
49	Reserved	-	Reserved
50	sai3	OR	SAI3 Receive Interrupt
50	sai3	OR	SAI3 Receive Async Interrupt
50	sai3	OR	SAI3 Transmit Interrupt
50	sai3	OR	SAI3 Transmit Async Interrupt
51	gpt5	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
52	gpt4	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
53	gpt3	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
54	gpt2	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
55	gpt1	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
56	gpio1	-	Active HIGH Interrupt from INT7 from GPIO
57	gpio1	-	Active HIGH Interrupt from INT6 from GPIO
58	gpio1	-	Active HIGH Interrupt from INT5 from GPIO
59	gpio1	-	Active HIGH Interrupt from INT4 from GPIO
60	gpio1	-	Active HIGH Interrupt from INT3 from GPIO
61	gpio1	-	Active HIGH Interrupt from INT2 from GPIO

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
62	gpio1	-	Active HIGH Interrupt from INT1 from GPIO
63	gpio1	-	Active HIGH Interrupt from INT0 from GPIO
64	gpio1	-	Combined interrupt indication for GPIO1 signal 0 throughout 15
65	gpio1	-	Combined interrupt indication for GPIO1 signal 16 throughout 31
66	gpio2	-	Combined interrupt indication for GPIO2 signal 0 throughout 15
67	gpio2	-	Combined interrupt indication for GPIO2 signal 16 throughout 31
68	gpio3	-	Combined interrupt indication for GPIO3 signal 0 throughout 15
69	gpio3	-	Combined interrupt indication for GPIO3 signal 16 throughout 31
70	gpio4	-	Combined interrupt indication for GPIO4 signal 0 throughout 15
71	gpio4	-	Combined interrupt indication for GPIO4 signal 16 throughout 31
72	gpio5	-	Combined interrupt indication for GPIO5 signal 0 throughout 15
73	gpio5	-	Combined interrupt indication for GPIO5 signal 16 throughout 31
74	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
75	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
76	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
77	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
78	wdog1	-	Watchdog 1 Timer reset
79	wdog2	-	Watchdog 2 Timer reset
80	pcie_ctrl2	-	Channels [63:32] interrupts requests
81	pwm1	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
82	pwm2	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
83	pwm3	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
84	pwm4	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
85	ccmsrcgpcmix	-	CCM, Interrupt Request 1

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
86	ccmsrcgpcmix	-	CCM, Interrupt Request 2
87	ccmsrcgpcmix	-	GPC, Interrupt Request 1
88	mu	-	Interrupt to A53
89	ccmsrcgpcmix	-	SRC interrupt request
90	sai5	OR	SAI5 Receive Interrupt
90	sai5	OR	SAI5 Receive Async Interrupt
90	sai5	OR	SAI5 Transmit Interrupt
90	sai5	OR	SAI5 Transmit Async Interrupt
90	sai6	OR	SAI6 Receive Interrupt
90	sai6	OR	SAI6 Receive Async Interrupt
90	sai6	OR	SAI6 Transmit Interrupt
90	sai6	OR	SAI6 Transmit Async Interrupt
91	rtic	-	RTIC Interrupt
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[0])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[1])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[2])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[3])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[0])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[1])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[2])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[3])
94	ccmsrcgpcmix	-	Combined CPU wdog interrupts (4x) out of SRC.
95	sai1	OR	SAI1 Receive Interrupt
95	sai1	OR	SAI1 Receive Async Interrupt
95	sai1	OR	SAI1 Transmit Interrupt
95	sai1	OR	SAI1 Transmit Async Interrupt
96	sai2	OR	SAI2 Receive Interrupt
96	sai2	OR	SAI2 Receive Async Interrupt
96	sai2	OR	SAI2 Transmit Interrupt
96	sai2	OR	SAI2 Transmit Async Interrupt
97	mu	-	Interrupt to M4
98	ddr	-	Interrupt for performance monitor in DRAM controller
99	ddr	-	-

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
100	sai4	OR	SAI4 Receive Interrupt
100	sai4	OR	SAI4 Receive Async Interrupt
100	sai4	OR	SAI4 Transmit Interrupt
100	sai4	OR	SAI4 Transmit Async Interrupt
101	cpu	-	Error indicator for AXI transaction with a write response error condition
102	cpu	-	Error indicator for L2 RAM double-bit ECC error
103	sdma2	-	AND of all 48 SDMA interrupts (events) from all the channels
104	Reserved	-	Reserved
105	caam_wrapper	-	CAAM interrupt queue for JQ
106	caam_wrapper	-	CAAM interrupt queue for JQ
107	qspi	-	QSPI Interrupt
108	tzasc	-	TZASC (PL380) interrupt
109	Reserved	-	Reserved
110	Reserved	-	Reserved
111	Reserved	-	Reserved
112	perfmon1	-	General interrupt
113	perfmon2	-	General interrupt
114	caam_wrapper	-	CAAM interrupt queue for JQ
115	caam_wrapper	-	Recoverable error interrupt
116	hs	-	HS Interrupt Request
117	hevcdec	-	HEVC Decoder Interrupt
118	enet1	OR	MAC 0 Receive Buffer Done
118	enet1	OR	MAC 0 Receive Frame Done
118	enet1	OR	MAC 0 Transmit Buffer Done
118	enet1	OR	MAC 0 Transmit Frame Done
119	enet1	OR	MAC 0 Receive Buffer Done
119	enet1	OR	MAC 0 Receive Frame Done
119	enet1	OR	MAC 0 Transmit Buffer Done
119	enet1	OR	MAC 0 Transmit Frame Done
120	enet1	OR	MAC 0 Periodic Timer Overflow
120	enet1	OR	MAC 0 Time Stamp Available
120	enet1	OR	MAC 0 Payload Receive Error
120	enet1	OR	MAC 0 Transmit FIFO Underrun
120	enet1	OR	MAC 0 Collision Retry Limit
120	enet1	OR	MAC 0 Late Collision
120	enet1	OR	MAC 0 Ethernet Bus Error
120	enet1	OR	MAC 0 MII Data Transfer Done
120	enet1	OR	MAC 0 Receive Buffer Done

Table continues on the next page...

Table 7-1. ARM Domain Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
120	enet1	OR	MAC 0 Receive Frame Done
120	enet1	OR	MAC 0 Transmit Buffer Done
120	enet1	OR	MAC 0 Transmit Frame Done
120	enet1	OR	MAC 0 Graceful Stop
120	enet1	OR	MAC 0 Babbling Transmit Error
120	enet1	OR	MAC 0 Babbling Receive Error
120	enet1	OR	MAC 0 Receive Flush Frame0
120	enet1	OR	MAC 0 Receive Flush Frame1
120	enet1	OR	MAC 0 Receive Flush Frame2
120	enet1	OR	MAC 0 Wakeup Request (sync)
120	enet1	OR	MAC 0 Babbling Receive Error
120	enet1	OR	MAC 0 Wakeup Request (sync)
121	enet1	-	MAC 0 1588 Timer Interrupt – synchronous
122	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
123	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
124	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
125	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
126	Reserved	-	Reserved
127	pcie_ctrl1	-	Channels [63:32] interrupts requests

7.1.3 CM4 Interrupts

The Nested Vectored Interrupt Controller (NVIC) collects up to 128 interrupt requests from all chip sources and provides an interface to the Cortex M4 Core.

The following table describes the M4 interrupt sources.

Table 7-2. CM4 Interrupt Summary

IRQ	Module	Logic	Interrupt Description
IRQ	MODULE	LOGIC	Description
0	boot	-	Used to notify cores on exception condition while boot
1	dap	-	DAP Interrupt
2	sdma1	-	AND of all 48 SDMA interrupts (events) from all the channels

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
3	gpu	-	GPU Interrupt
4	snvs_lp_wrapper	OR	ON-OFF button press shorter than 5 secs (pulse event)
4	snvs_hp_wrapper	OR	ON-OFF button press shorter than 5 secs (pulse event)
5	lcdif	-	LCDIF Interrupt
6	spdif1	OR	SPDIF Rx interrupt
6	spdif1	OR	SPDIF Tx interrupt
7	h264dec	-	h264 Decoder Interrupt
8	vpudma	-	VPU DMA Interrupt
9	qos	-	QOS Interrupt
10	wdog3	-	Watchdog Timer reset
11	hs	-	HS Interrupt Request
12	apbhdma	OR	GPMI operation channel 0 description complete interrupt
12	apbhdma	OR	GPMI operation channel 1 description complete interrupt
12	apbhdma	OR	GPMI operation channel 2 description complete interrupt
12	apbhdma	OR	GPMI operation channel 3 description complete interrupt
13	spdif2	OR	SPDIF Rx interrupt
13	spdif2	OR	SPDIF Tx interrupt
14	rawnand	-	BCH operation complete interrupt
15	rawnand	-	GPMI operation TIMEOUT ERROR interrupt
16	hdmi_ips	-	HDMI IRQ
17	hdmi_ips	-	HDMI IRQ
18	hdmi_ips	-	HDMI IRQ
19	snvs_hp_wrapper	-	SRTC Consolidated Interrupt. Non TZ.
20	snvs_hp_wrapper	-	SRTC Security Interrupt. TZ.
21	csu	-	CSU Interrupt Request. Indicates to the processor that one or more alarm inputs were asserted
22	usdhc1	-	uSDHC1 Enhanced SDHC Interrupt Request
23	usdhc2	-	uSDHC2 Enhanced SDHC Interrupt Request
24	dc8000_control	-	DC8000 Display Controller IRQ
25	DTRC_wrapper	-	DTRC IRQ
26	uart1	-	UART-1 ORed interrupt
27	uart2	-	UART-2 ORed interrupt
28	uart3	-	UART-3 ORed interrupt
29	uart4	-	UART-4 ORed interrupt
30	vp9dec	-	VP9 Decoder Interrupt

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
31	ecspi1	-	eCSPI1 interrupt request line to the core.
32	ecspi2	-	eCSPI2 interrupt request line to the core.
33	ecspi3	-	eCSPI3 interrupt request line to the core.
34	mipi_dsi	-	MIPI-DSI interrupt
35	i2c1	-	I2C-1 Interrupt
36	i2c2	-	I2C-2 Interrupt
37	i2c3	-	I2C-3 Interrupt
38	i2c4	-	I2C-4 Interrupt
39	rdc	-	RDC interrupt
40	usb1	-	USB-1 Interrupt
41	usb2	-	USB-2 Interrrupt
42	csi1	-	CSI-1 Interrupt
43	csi2	-	CSI-2 Interrupt
44	mipi_csi1	-	MIPI-CSI-1 Interrupt
45	mipi_csi2	-	MIPI-CSI-2 Interrupt
46	gpt6	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
47	sctr	-	System Counter Interrupt [0]
48	sctr	-	System Counter Interrupt [1]
49	anamix	OR	TempSensor (Temperature alarm).
49	anamix	OR	TempSensor (Temperature critical alarm).
49	Reserved	-	Reserved
50	sai3	OR	SAI3 Receive Interrupt
50	sai3	OR	SAI3 Receive Async Interrupt
50	sai3	OR	SAI3 Transmit Interrupt
50	sai3	OR	SAI3 Transmit Async Interrupt
51	gpt5	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
52	gpt4	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
53	gpt3	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
54	gpt2	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
55	gpt1	-	OR of GPT Rollover interrupt line, Input Capture 1 & 2 lines, Output Compare 1,2 &3 Interrupt lines
56	gpio1	-	Active HIGH Interrupt from INT7 from GPIO

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
57	gpio1	-	Active HIGH Interrupt from INT6 from GPIO
58	gpio1	-	Active HIGH Interrupt from INT5 from GPIO
59	gpio1	-	Active HIGH Interrupt from INT4 from GPIO
60	gpio1	-	Active HIGH Interrupt from INT3 from GPIO
61	gpio1	-	Active HIGH Interrupt from INT2 from GPIO
62	gpio1	-	Active HIGH Interrupt from INT1 from GPIO
63	gpio1	-	Active HIGH Interrupt from INT0 from GPIO
64	gpio1	-	Combined interrupt indication for GPIO1 signal 0 throughout 15
65	gpio1	-	Combined interrupt indication for GPIO1 signal 16 throughout 31
66	gpio2	-	Combined interrupt indication for GPIO2 signal 0 throughout 15
67	gpio2	-	Combined interrupt indication for GPIO2 signal 16 throughout 31
68	gpio3	-	Combined interrupt indication for GPIO3 signal 0 throughout 15
69	gpio3	-	Combined interrupt indication for GPIO3 signal 16 throughout 31
70	gpio4	-	Combined interrupt indication for GPIO4 signal 0 throughout 15
71	gpio4	-	Combined interrupt indication for GPIO4 signal 16 throughout 31
72	gpio5	-	Combined interrupt indication for GPIO5 signal 0 throughout 15
73	gpio5	-	Combined interrupt indication for GPIO5 signal 16 throughout 31
74	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
75	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
76	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
77	pcie_ctrl2	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
78	wdog1	-	Watchdog 1 Timer reset
79	wdog2	-	Watchdog 2 Timer reset
80	pcie_ctrl2	-	Channels [63:32] interrupts requests
81	pwm1	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
82	pwm2	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
83	pwm3	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
84	pwm4	-	Cumulative interrupt line. OR of Rollover Interrupt line, Compare Interrupt line and FIFO Waterlevel crossing interrupt line
85	ccmsrcgpcmix	-	CCM, Interrupt Request 1
86	ccmsrcgpcmix	-	CCM, Interrupt Request 2
87	ccmsrcgpcmix	-	GPC, Interrupt Request 1
88	mu	-	Interrupt to A53
89	ccmsrcgpcmix	-	SRC interrupt request
90	sai5	OR	SAI5 Receive Interrupt
90	sai5	OR	SAI5 Receive Async Interrupt
90	sai5	OR	SAI5 Transmit Interrupt
90	sai5	OR	SAI5 Transmit Async Interrupt
90	sai6	OR	SAI6 Receive Interrupt
90	sai6	OR	SAI6 Receive Async Interrupt
90	sai6	OR	SAI6 Transmit Interrupt
90	sai6	OR	SAI6 Transmit Async Interrupt
91	rtic	-	RTIC Interrupt
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[0])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[1])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[2])
92	cpu	OR	Performance Unit Interrupts from Quad-A53 platform (interrnally: PMUIRQ[3])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[0])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[1])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[2])
93	cpu	OR	CTI trigger outputs from Quad-A53 platform (internal: nCTIIRQ[3])
94	ccmsrcgpcmix	-	Combined CPU wdog interrupts (4x) out of SRC.
95	sai1	OR	SAI1 Receive Interrupt
95	sai1	OR	SAI1 Receive Async Interrupt
95	sai1	OR	SAI1 Transmit Interrupt
95	sai1	OR	SAI1 Transmit Async Interrupt
96	sai2	OR	SAI2 Receive Interrupt
96	sai2	OR	SAI2 Receive Async Interrupt

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
96	sai2	OR	SAI2 Transmit Interrupt
96	sai2	OR	SAI2 Transmit Async Interrupt
97	mu	-	Interrupt to M4
98	ddr	-	Interrupt for performance monitor in DRAM controller
99	ddr	-	-
100	sai4	OR	SAI4 Receive Interrupt
100	sai4	OR	SAI4 Receive Async Interrupt
100	sai4	OR	SAI4 Transmit Interrupt
100	sai4	OR	SAI4 Transmit Async Interrupt
101	cpu	-	Error indicator for AXI transaction with a write response error condition
102	cpu	-	Error indicator for L2 RAM double-bit ECC error
103	sdma2	-	AND of all 48 SDMA interrupts (events) from all the channels
104	Reserved	-	Reserved
105	caam_wrapper	-	CAAM interrupt queue for JQ
106	caam_wrapper	-	CAAM interrupt queue for JQ
107	qspi	-	QSPI Interrupt
108	tzasc	-	TZASC (PL380) interrupt
109	Reserved	-	Reserved
110	Reserved	-	Reserved
111	Reserved	-	Reserved
112	perfmon1	-	General interrupt
113	perfmon2	-	General interrupt
114	caam_wrapper	-	CAAM interrupt queue for JQ
115	caam_wrapper	-	Recoverable error interrupt
116	hs	-	HS Interrupt Request
117	hevcdec	-	HEVC Decoder Interrupt
118	enet1	OR	MAC 0 Receive Buffer Done
118	enet1	OR	MAC 0 Receive Frame Done
118	enet1	OR	MAC 0 Transmit Buffer Done
118	enet1	OR	MAC 0 Transmit Frame Done
119	enet1	OR	MAC 0 Receive Buffer Done
119	enet1	OR	MAC 0 Receive Frame Done
119	enet1	OR	MAC 0 Transmit Buffer Done
119	enet1	OR	MAC 0 Transmit Frame Done
120	enet1	OR	MAC 0 Periodic Timer Overflow
120	enet1	OR	MAC 0 Time Stamp Available
120	enet1	OR	MAC 0 Payload Receive Error

Table continues on the next page...

Table 7-2. CM4 Interrupt Summary (continued)

IRQ	Module	Logic	Interrupt Description
120	enet1	OR	MAC 0 Transmit FIFO Underrun
120	enet1	OR	MAC 0 Collision Retry Limit
120	enet1	OR	MAC 0 Late Collision
120	enet1	OR	MAC 0 Ethernet Bus Error
120	enet1	OR	MAC 0 MII Data Transfer Done
120	enet1	OR	MAC 0 Receive Buffer Done
120	enet1	OR	MAC 0 Receive Frame Done
120	enet1	OR	MAC 0 Transmit Buffer Done
120	enet1	OR	MAC 0 Transmit Frame Done
120	enet1	OR	MAC 0 Graceful Stop
120	enet1	OR	MAC 0 Babbling Transmit Error
120	enet1	OR	MAC 0 Babbling Receive Error
120	enet1	OR	MAC 0 Receive Flush Frame0
120	enet1	OR	MAC 0 Receive Flush Frame1
120	enet1	OR	MAC 0 Receive Flush Frame2
120	enet1	OR	MAC 0 Wakeup Request (sync)
120	enet1	OR	MAC 0 Babbling Receive Error
120	enet1	OR	MAC 0 Wakeup Request (sync)
121	enet1	-	MAC 0 1588 Timer Interrupt – synchronous
122	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
123	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
124	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
125	pcie_ctrl1	-	Coming from GLUE logic, of set/reset FF, driven by PCIE signals.
126	Reserved	-	Reserved
127	pcie_ctrl1	-	Channels [63:32] interrupts requests

7.1.4 SDMA event mapping

The following table shows the DMA request signals for peripherals in the chip.

Table 7-3. SDMA1 event mapping

SDMA	Module	Description
0	ecspi1	eCSPI1 Rx request
1	ecspi1	eCSPI1 Tx request

Table continues on the next page...

Table 7-3. SDMA1 event mapping (continued)

SDMA	Module	Description
2	ecspi2	eCSPI2 Rx request
3	ecspi2	eCSPI2 Tx request
4	-	Reserved
5	-	Reserved
6	-	Reserved
7	-	Reserved
8	spdif1	SPDIF1 Rx DMA request
9	spdif1	SPDIF1 Tx DMA request
10	sai2	SAI-2 receive DMA request
11	sai2	SAI-2 transmit DMA request
12	sai3	SAI-3 receive DMA request
13	sai3	SAI-3 transmit DMA request
14	iomux	external DMA from pad through IOMUX #1
15	iomux	external DMA from pad through IOMUX #2
16	spdif2	SPDIF2 Rx DMA request
17	spdif2	SPDIF2 Tx DMA request
18	i2c1	I2C1 DMA event
19	i2c2	I2C2 DMA event
20	i2c3	I2C3 DMA event
21	i2c4	I2C4 DMA event
22	uart1	Rx FIFO
23	uart1	Tx FIFO
24	uart2	Rx FIFO
25	uart2	Tx FIFO
26	uart3	Rx FIFO
27	uart3	Tx FIFO
28	uart4	Rx FIFO
29	uart4	Tx FIFO
30	-	Reserved
31	-	Reserved
32	-	Reserved
33	-	Reserved
34	-	Reserved
35	-	Reserved
36	qspi1	QSPI DMA TX request
37	qspi1	QSPI DMA RX request
38	gpt1	GPT1 counter event
39	gpt2	GPT2 counter event
40	gpt3	GPT3 counter event

Table continues on the next page...

Table 7-3. SDMA1 event mapping (continued)

SDMA	Module	Description
41	-	Reserved
42	-	Reserved
43	-	Reserved
44	enet1	ENET1 1588 Event2 out
45	enet1	ENET1 1588 Event0 out
46	enet1	ENET1 1588 Event3 out
47	enet1	ENET1 1588 Event1 out

Table 7-4. SDMA2 event mapping

SDMA	Module	Description
SDMA	MODULE	Description
0	sai4	SAI-4 receive DMA request
1	sai4	SAI-4 transmit DMA request
2	sai5	SAI-5 receive DMA request
3	sai5	SAI-5 transmit DMA request
4	sai6	SAI-6 receive DMA request
5	sai6	SAI-6 transmit DMA request
6	-	Reserved
7	-	Reserved
8	sai1	SAI-1 receive DMA request
9	sai1	SAI-1 transmit DMA request
10	-	Reserved
11	-	Reserved
12	-	Reserved
13	-	Reserved
14	iomux	external DMA from pad through IOMUX #1
15	iomux	external DMA from pad through IOMUX #2
16	-	Reserved
17	-	Reserved
18	-	Reserved
19	-	Reserved
20	-	Reserved
21	-	Reserved
22	-	Reserved
23	-	Reserved
24	-	Reserved
25	-	Reserved
26	-	Reserved

Table continues on the next page...

Table 7-4. SDMA2 event mapping (continued)

SDMA	Module	Description
27	-	Reserved
28	-	Reserved
29	-	Reserved
30	-	Reserved
31	-	Reserved
32	-	Reserved
33	-	Reserved
34	-	Reserved
35	-	Reserved
36	-	Reserved
37	-	Reserved
38	gpt4	GPT4 counter event
39	gpt5	GPT5 counter event
40	gpt6	GPT6 counter event
41	-	Reserved
42	-	Reserved
43	-	Reserved
44	-	Reserved
45	-	Reserved
46	-	Reserved
47	-	Reserved

7.2 Smart Direct Memory Access Controller (SDMA)

7.2.1 Overview

The Smart Direct Memory Access (SDMA) controller offers highly-competitive DMA features combined with software-based virtual-DMA flexibility. It enables data transfers between peripheral I/O devices and internal/external memories.

The SDMA controller helps maximize system performance by off-loading the Arm core in dynamic data routing.

7.2.1.1 Block Diagram

The figure below shows a block diagram of the SDMA controller. It includes the custom RISC core along with its RAM, ROM, DMA units, and the scheduler.

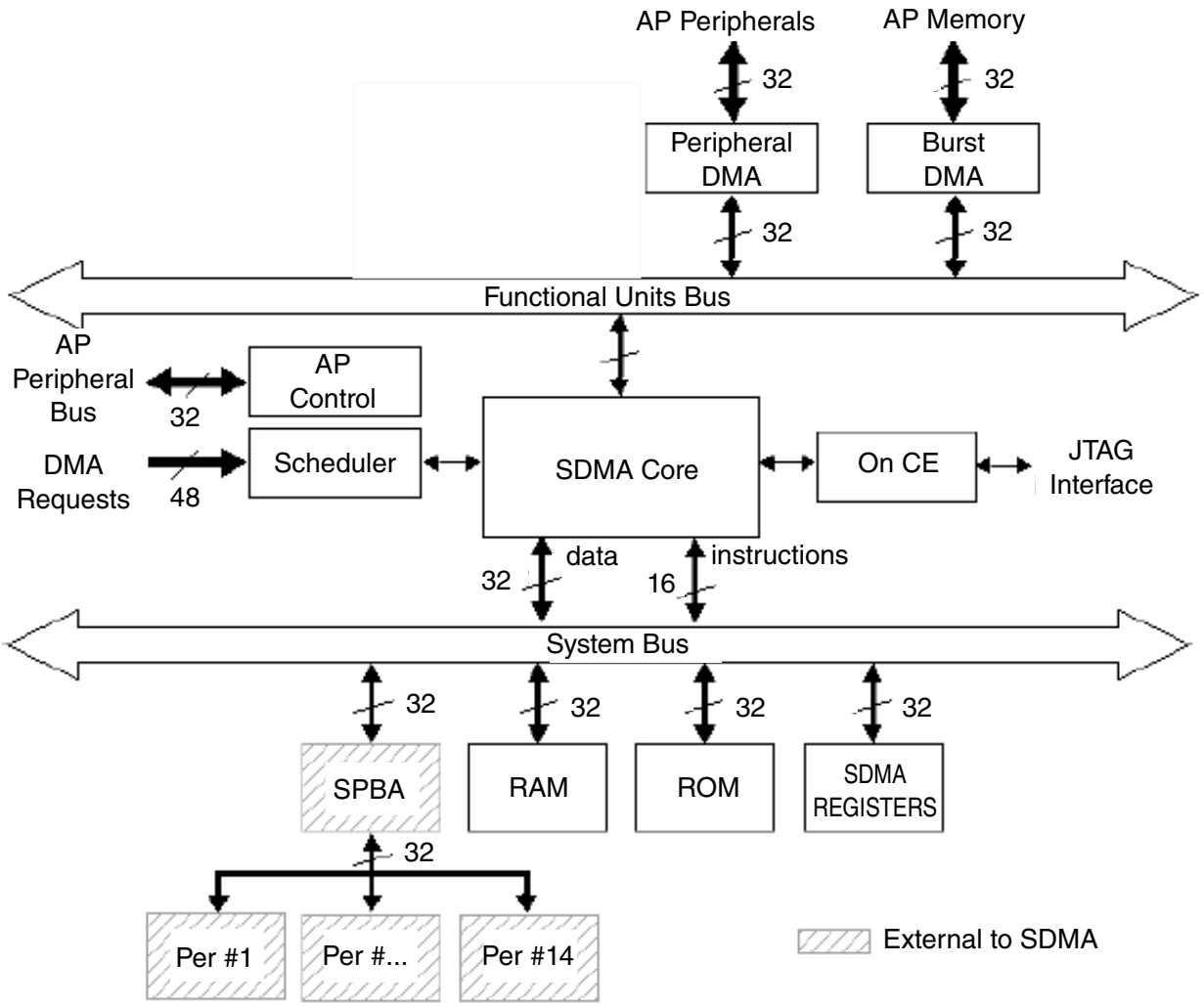


Figure 7-1. SDMA Block Diagram

The SDMA core executes short routines that perform DMA transfers; these routines are called *scripts*. The SDMA core interfaces to its own memory via the SDMA system bus. The SDMA system bus supports a 32-bit data path and a 16-bit address bus. The system bus datapath is used for both 16-bit instruction (program) memory access and 32-bit data access. DMA units interface to the core via the Functional Unit Bus and use dedicated registers to perform DMA transfers.

The SDMA memory contains a ROM and a RAM. The ROM contains startup scripts (for example, boot code) and other common utilities, which are referenced by the scripts that reside in the RAM. The internal RAM is divided into a context area and a script area (more details about this mapping are available in Instruction Memory Map and Data Memory Map).

Every transfer channel requires one context area to keep the contents of all the core and unit registers while inactive. Channel scripts are downloaded into the internal RAM by the SDMA using a dedicated channel that is started during the boot sequence. Downloads are invoked using commands and pointers provided by the Arm platform. Every channel contains a corresponding channel script located in RAM and/or ROM that can be reconfigured independently as-needed. Channel scripts can be stored in an external memory and downloaded when needed. The SDMA can be configured with any mixture of scripts to enable an endless combination of supported services.

The scheduler monitors and detects DMA requests, mapping them to channels, and mapping individual channels to a pre-configured priority. At any given point, the scheduler presents the highest priority channel that requires service to the SDMA core. A special SDMA core instruction is used to "conditionally yield" the current channel being executed to an eligible channel that requires service. If (and only if) there is an eligible channel pending, will the current channel execution be preempted.

There are two yield instructions that differently determine the eligible channels: In the first version, eligible channels are pending channels with a strictly higher priority than the current channel priority. In the second version (yieldge), eligible channels are pending channels with a priority that is greater or equal to the current channel priority. The scheduler detects devices that need service through its 48 DMA request inputs. After a request is detected, the scheduler determines the channel(s) that is (are) triggered by this request and marks it (them) as pending in the "Channel Pending (EP)" register. The priorities of all the pending channels are continuously evaluated in order to update the highest pending priority. The channel pending flag is cleared by the channel script when the transfer has completed.

The Arm platform control block contains the control registers used to configure the 32 individual channels. There are 48 Channel Enable registers, and every register maps one DMA request to any desired combination of channels. The 32 Priority registers are used to assign a programmable 1-of-7 level priority to every possible channel. This block also contains all other control registers that the Arm platform can access.

The 48 DMA requests that are connected to the scheduler come from a variety of sources. The "receive register full" and "transmit register empty" signals found in the UART and USB ports are typical examples of DMA requests that can be connected to the SDMA. These requests can be used to trigger a specific SDMA channel, or several channels.

There is an OnCE compatible debug port for product development. The OnCE includes support for setting breakpoints, single-step and trace, and register dump capability. In addition, all memory locations are accessible from the debug port.

7.2.1.2 Features

The following are the SDMA features:

- Multi-channel DMA supporting up to 32 time-division multiplexed DMA channels
- Hardware or software driven triggers for each channel
- 48 hardware driven triggers that can be mapped to any channel.
- Memory accesses including linear addressing, FIFO addressing and 2D addressing
- Fast context-switching with two-level, priority-based preemptive multi-tasking
- 16-bit instruction-set micro-RISC engine (the SDMA core)
- Two DMA units with some or all the following features:
  - Auto-flush and prefetch capability
  - Flexible address management (increment, decrement, and no address changes on source and destination address)
  - Misaligned data-transfer support
  - Uni-directional and bi-directional flows (copy mode)
  - Up to eight-word buffers for configurable burst transfers
- Support of byte-swapping
- An available API and library of scripts
- Little-Endian and Big-Endian modes
- Hardware handshakes for low-power entry sequence
- Security support to lock contents of the SDMA script RAM.
- 4-Kbyte ROM containing startup scripts (for example, boot code) and other common utilities that can be referenced by RAM-located scripts
- 8-Kbyte RAM area is divided into a processor context area and a code space area used to store channel scripts that are downloaded from the system memory
- Debug support, including a OnCE port, real-time monitors, and embedded cross-trigger events
- Supported clock frequencies in process:
  - Configurable clock options for the SDMA core and the Arm platform DMA units
    - 1:2 ratio with maximum of SDMA core running at Arm platform Peripheral Bus speed and DMA running at max DMA frequency.
    - 1:1 ratio when both SDMA core and Arm platform DMA clocks are set to the Arm platform Peripheral Bus speed.
- Peripheral bus interface for configuration register programming by the Arm platform

- The SDMA RISC engine (arithmetic and logic operations), which is referred to as the "SDMA core."
- An internal peripheral bus connected to the Shared Peripherals Bus Interface (SPBA) that enables access to up to 14 shared peripherals. SDMA supports 32-bit accesses to word peripherals and 16-bit accesses to half-word peripherals.
- The peripheral DMA unit that is hooked-up to the Arm platform Crossbar Switch to service Arm peripherals
- The burst DMA unit is able to perform burst accesses to the external memory
- All the DMA units are 32-bit AHB masters. They are connected to different buses, thus allowing concurrent accesses.

7.2.2 Clocks

The following table describes the clock sources for SDMA. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information. For functional information regarding module clocks, see SDMA Clocks and Low Power Modes.

Table 7-5. SDMA Clocks

Clock name	Clock Root	Description
events_sync_clk (clk)	ahb_clk_root	Arm peripheral / events clock
ips_hostctrl_clk	ipg_clk_root	Host control clock
ap_ahb_clk	ahb_clk_root	Arm platform bus clock
core_clk	ipg_clk_root	Module / Core clock
tck	-	JTAG access clock

7.2.3 Functional Description

The figure below shows the SDMA topology, and is composed of the following components:

- SDMA Core (SDMA Core)
- SDMA Scheduler (Scheduler)
- Functional Units:
  - Burst DMA (Burst DMA Unit)
  - Peripheral DMA (Peripheral DMA Unit)
- Arm platform Control for Arm control register access.
- Internal RAM and ROM Memory (SDMA Programming Model)
- OnCE debug Port (The OnCE Controller)

The functional unit bus provides access by the SDMA core to the DMA units. The system bus provides access to SDMA internal memory and also supports up to 14 peripherals.

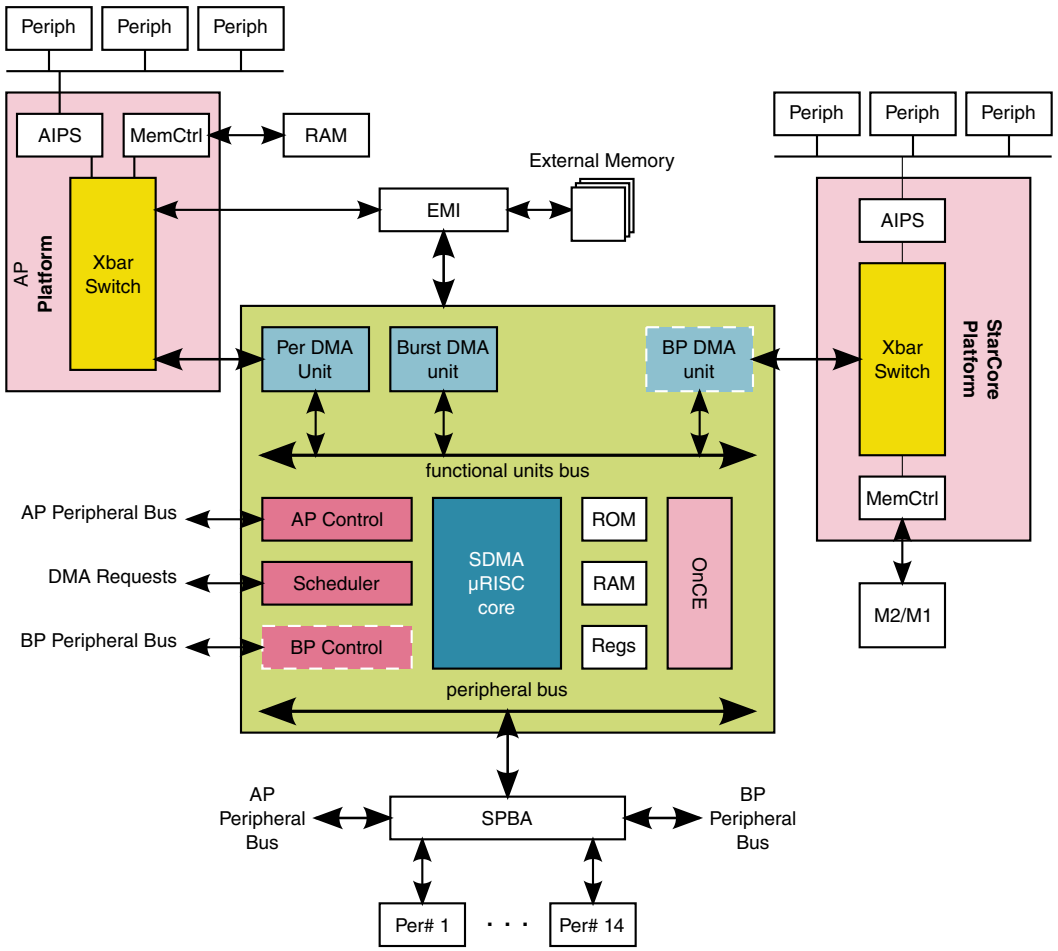


Figure 7-2. SDMA Connections

7.2.3.1 SDMA Core

The SDMA core is a customized RISC-like processor that is specifically developed to control DMA units and perform L1 tasks like byte-stuffing or framing.

The SDMA core incorporates on-chip debug capability using the OnCE.

The SDMA core is based on a 32-bit register architecture with 16-bit instructions. There are eight general purpose 32-bit registers, four flags (T, LM, SF, and DF), and four PCU registers (PC, RPC, SPC, and EPC) that can address 16,384 16-bit instructions.

7.2.3.1.1 SDMA Core Structure

The figure found here shows the structure of the SDMA core. It also shows the different registers, calculation resources, and possible data movements.

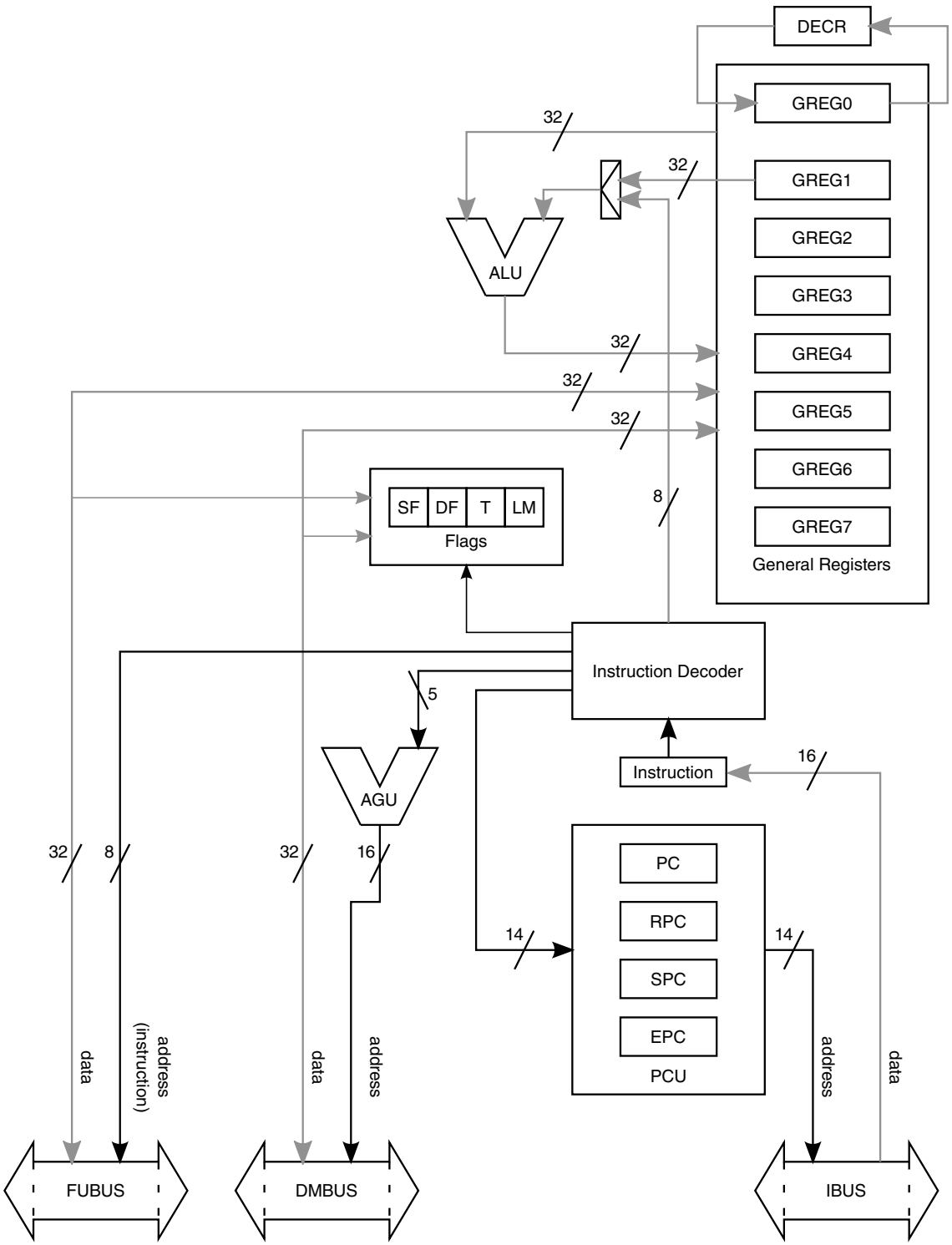


Figure 7-3. SDMA Core

- The Program Control Unit (PCU) is described in Program Control Unit (PCU). It handles the state of the core and generates the instruction fetch addresses. Instructions are retrieved from the Instruction Bus (IBUS) and stored in the SDMA core instruction register prior to their decoding. The PCU contains the following registers:
  - The Program Counter (PC) contains the address of the current instruction.
  - The Return Program Counter (RPC) contains the address of the instruction that follows a jump to the subroutine.
  - The Start Program Counter (SPC) contains the address of the first instruction of the current hardware loop.
  - End Program Counter (EPC) contains the address of the last instruction of the current hardware loop.
- The other core registers are the general purpose registers (GREGn) and the flags.
  - The general purpose registers can be used to hold data and addresses. They can be loaded with immediate values (for example, 8-bit data that are encoded in the instruction), results of calculations that were performed with the ALU, 32-bit data that comes from the memory or peripherals via the Data Memory Bus (DMBUS), 32-bit data that comes from the DMAs via the Functional Units Bus (FUBUS) or another general purpose register. Their content can be the operands of the ALU, the data to send on either bus (DMBUS or FUBUS), or a pointer to memory (DMBUS address).
  - The general register 0 (GREG0) is also the hardware loop counter. In hardware loops, it cannot be used for any other purpose. This register uses a dedicated decrement unit (DECR) shown in Figure 7-3.
  - The flags reflect the status of operations:
    - SF and DF are set when the last load or store on either bus (FUBUS or DMBUS) received an error response.
    - LM is set when the core is executing instructions inside a hardware loop.
    - T is set when the ALU operation result was 0 or the loop counter reaches 0 (the latter is preponderant when an ALU operation is the last instruction of a hardware loop).
- The ALU has two operands: any general register and either a second general register or an immediate value. The result is always stored into the first general register. A NOP function can be utilized by moving a register's contents into itself (For example, the instruction: mov R0,R0).
- The 16-bit instructions are fetched via the instruction bus (IBUS) whose address is driven by the PC. The SDMA RAM and ROM are visible to the core as 16-bit devices through this interface.
- The memory (RAM and ROM), memory mapped registers, and external peripherals are accessed via the DMBUS. The address is always taken from a general register whose content is added to a 5-bit immediate value. This is the only available

addressing mode. The DMBUS is a 32-bit data bus. Except for the peripherals that are external to the SDMA, the address accuracy is the 32-bit word (for example, adding 1 to an address points to the next word, not the next byte).

- The functional units are accessed via the FUBUS connection. The data is exchanged with any general register, but the address (which in fact is the instruction and the selector of the functional unit) comes from an 8-bit field of the corresponding load or store.

7.2.3.1.2 Program Control Unit (PCU)

This part of the SDMA core is dedicated to the control of the RISC engine, as implied by the instructions that are executed. Its behavior is determined by the instruction type and the inputs of the SDMA.

It contains the PC, RPC, SPC, and EPC registers that are described in SDMA Core Structure.

7.2.3.1.2.1 Instruction Types

The state sequence and the delay of execution vary according to the type of the instruction. There are six possible categories of instructions, as follows:

1. Standard: Most of the instructions belong to this category, and always last 1 cycle.
2. ldf/stf: These are respectively the load and store instructions that access the functional units. They last 1+n cycles where n is the number of wait-states of the targeted functional unit.
3. ld/st: These are the load and store instructions that access the memory and peripherals. They last 1+n cycles where n is the number of wait-states of the targeted device (1 for the ROM, RAM, and memory mapped registers, 1 + the external peripheral wait-states). These instructions always last at least two cycles, but the core is able to handle them in one cycle. The first wait-state is inserted outside the core.
4. Branch: These are all the instructions that cause the Program Counter to point to another instruction other than the following one (for example, one that breaks the sequential flow). There are the absolute jumps, the conditional branches, the jump to the sub-routines, and the return from the sub-routine.
5. Loop,Modified Load or Store: The hardware loop instruction modifies the potential behavior of any load or store inside the loop (for example, when the LM flag is set). A jump may be implied after any such load or store if it received an error. The error causes an early exit of the loop, which means a jump to the instruction that follows the one that is pointed to by EPC. An additional cycle is required by the PCU to perform the jump (+1 to the ld/st/ldf/stf original execution delay). Although there is

usually an implicit jump after the last instruction of the loop when the PC goes back to SPC, this is performed at no cycle cost.

6. Done: The done, yield, or yieldge instructions are used to control channel switching. When no channel switching is performed, these instructions last a single cycle. When there is a change of channel or context switch, the delay is variable and depends on many factors (as detailed in Context Switching).

7.2.3.1.2.2 PCU States

The PCU state is visible through outputs of the SDMA (see Real-Time Debug Outputs) or the OnCE status register(see OnCE Status Register (OSTAT)).

The PCU state is a four-bit field that can take the values shown in the following table. Figure 7-4 shows the possible state transitions and the corresponding conditions.

Table 7-6. PCU States

Value	State	Description
0	Program	This is the usual instruction cycle.
1	Data	This state is inserted when there are wait-states during a load or a store on the data bus (ld/st type).
2	Change of Flow	This is the second cycle of any instruction that breaks the sequence of instructions (branch and done types). This state lasts only a single cycle; it is always followed by the Program state.
3	Error in Loop	This state is used when an error causes a hardware loop exit (loop-modified load or store type). This state only lasts a single cycle; it is always followed by the Program state.
4	Debug	The SDMA is stopped in debug mode.
5	Functional Unit	This state is inserted when there are wait-states during a load or a store on the functional units bus (ldf/stf type).
6	Sleep	No script is running: The core is idle after saving the last channel context.
7	Save	The context switch FSM is saving the current channel.
8	Program in Sleep	Same as Program except there is no associated channel, this state is used when instructions are executed after entering debug mode, whereas the core was in either Sleep mode.
9	Data in Sleep	This is the same as Data except there is no associated channel.
10	Change of Flow in Sleep	This is the same as Change of Flow except there is no associated channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
11	Error in Loop in Sleep	This is the same as Error in Loop except there is no associated. channel. This state only lasts a single cycle, and is always followed by the Program in Sleep state.
12	Debug in Sleep	This is the same as Debug except the core was put in debug mode when no channel was active.
13	Functional Unit in Sleep	This is the same as Functional Unit except there is no associated channel.

Table continues on the next page...

Table 7-6. PCU States (continued)

Value	State	Description
14	Sleep after Reset	This shows that no script is running, and the core is idle after a reset. When a channel becomes active, no context is restored but the core starts its boot program located at address 0 (or the address available in register in Channel 0 Boot Address (SDMAARM_CHN0ADDR)).
15	Restore	The context switch FSM is restoring the next channel context.

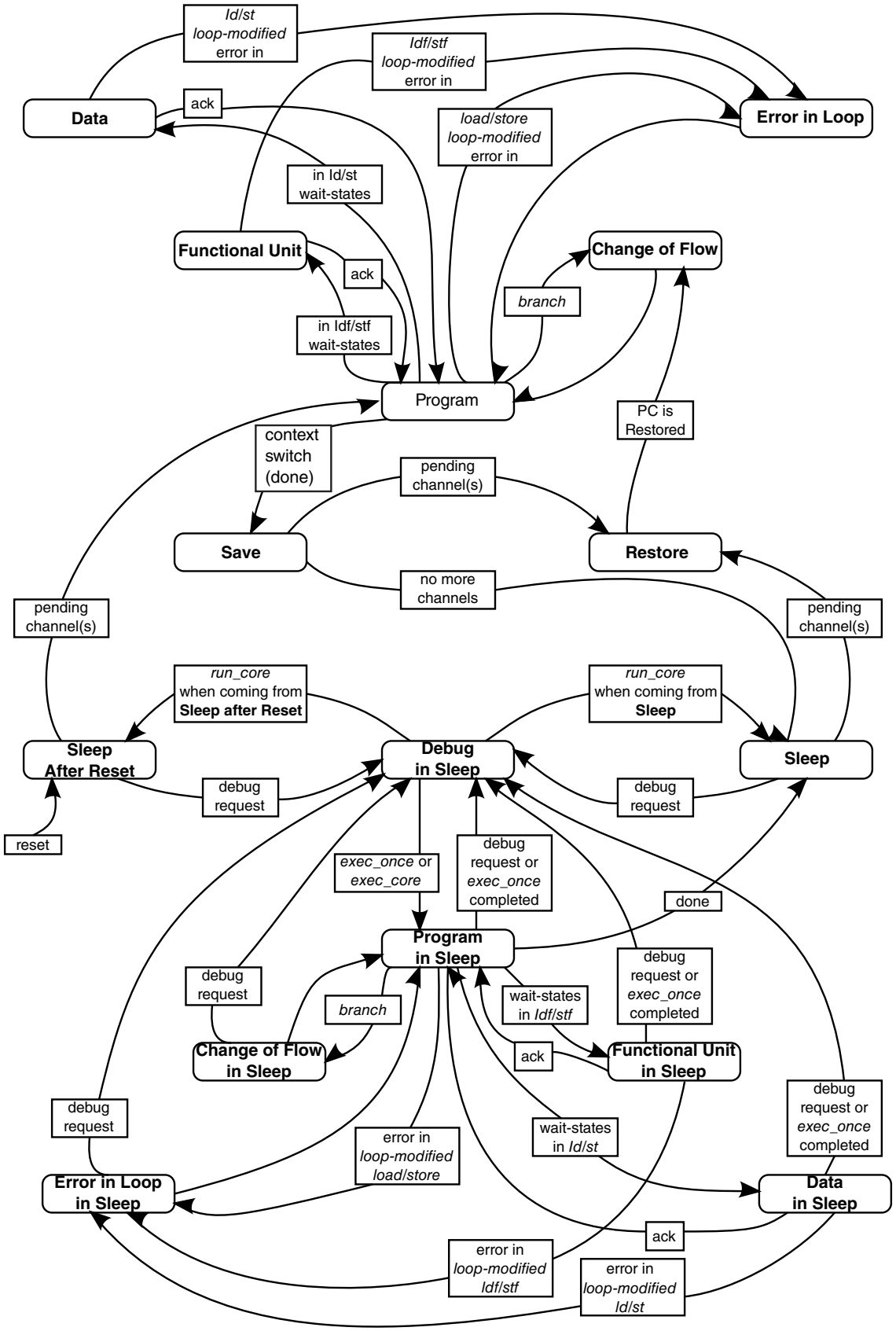


Figure 7-4. PCU State Diagram

7.2.3.1.3 SDMA Core Memory

The SDMA has two memory spaces: one for the instructions and one for the data. As both spaces share the same resources (ROM and RAM devices), the system bus manages possible conflicts when the core accesses the same resource for both an instruction read and a data read or write.

Program and data memory is further described in Address Space.

Instructions of 16-bit width are stored in 32-bit wide devices and can be accessed as data. The mapping is Big Endian: an even instruction address (terminated by 0) accesses the most significant part of the 32-bit data (bits [31:16]), and an odd instruction address (terminated by 1) accesses the least significant part of the 32-bit data (bits [15:0]). Instructions can be fetched out of internal ROM or RAM.

Data can be read from ROM, RAM, memory mapped registers, and external peripherals, and written to the same devices (except the ROM).

The ROM contains bootload scripts, channel scripts, and common subroutines which may be referenced by channel scripts elsewhere in the ROM or RAM.

The RAM is divided into a context area and a code space area which may be used to store channel scripts. The RAM contains undefined values after a hardware reset. Channel scripts and initial context values are downloaded into RAM using channel 0 which is reserved for bootload functions.

7.2.3.2 Scheduler

All channel scheduling hardware is included in the Scheduler.

7.2.3.2.1 Primary Functions

The scheduler is a hardware-based design used to coordinate the timely execution of 32 virtual DMA channels by the SDMA core on the basis of channel status and priority.

The scheduler performs the following functions:

- Monitors, detects, and registers the occurrence of any one of the 48 DMA requests
- Links a specific request to a channel or group of channels (channel mapping)
- Ignores requests that are not mapped to a previously configured channel
- Maintains a list of all the channels that are requesting service
- Assigns a pre-programmed priority level (1 of 7) to every channel requesting service
- Detects and flags overrun/underrun conditions

7.2.3.2.2 Channels and DMA Requests

7.2.3.2.2.1 Channels

A Virtual Channel (hereafter simply called a channel) manages a flow of data through the SDMA. Flows are typically unidirectional.

The SDMA can have up to 32 simultaneously operating channels, numbered from 0 to 31. Channel 0 is usually dedicated to control the SDMA script downloading. All the channels can be assigned by the Arm platform software.

7.2.3.2.2.2 DMA Requests

A DMA request is caused by externally (for example, external to the SDMA) controlled conditions (for example, UART receive FIFO reaches a threshold). The SDMA currently supports up to 48 DMA requests.

7.2.3.2.2.3 Mapping from DMA Requests to Channels and Priorities

A channel can stall waiting on a single DMA request. A single DMA request can awake more than one channel (in fact, any request can awake any combination of channels).

The mapping between DMA requests and channels is program-controlled. There is a storage element assigned for each of the 48 requests that contains a bitmap table of the channels that are awakened by the event.

Every channel also has a three-bit register that indicates its priority.

7.2.3.2.3 Scheduler Functional Description

Scheduler Overview describes the behavior of the SDMA scheduler-from the channel enabling conditions to the highest priority pending channel selection.

7.2.3.2.3.1 Scheduler Overview

The scheduler algorithm is built in hardware. It is provided with possibilities for the Arm platform to control its behavior.

The scheduler processes incoming DMA requests, maps detected requests to 0, one, or several channels, maintains a list of channels that are requesting service (pending channels), identifies the top priority and its associated channel, and selects the next active channel when the current channel yields.

The following figure shows a functional overview.

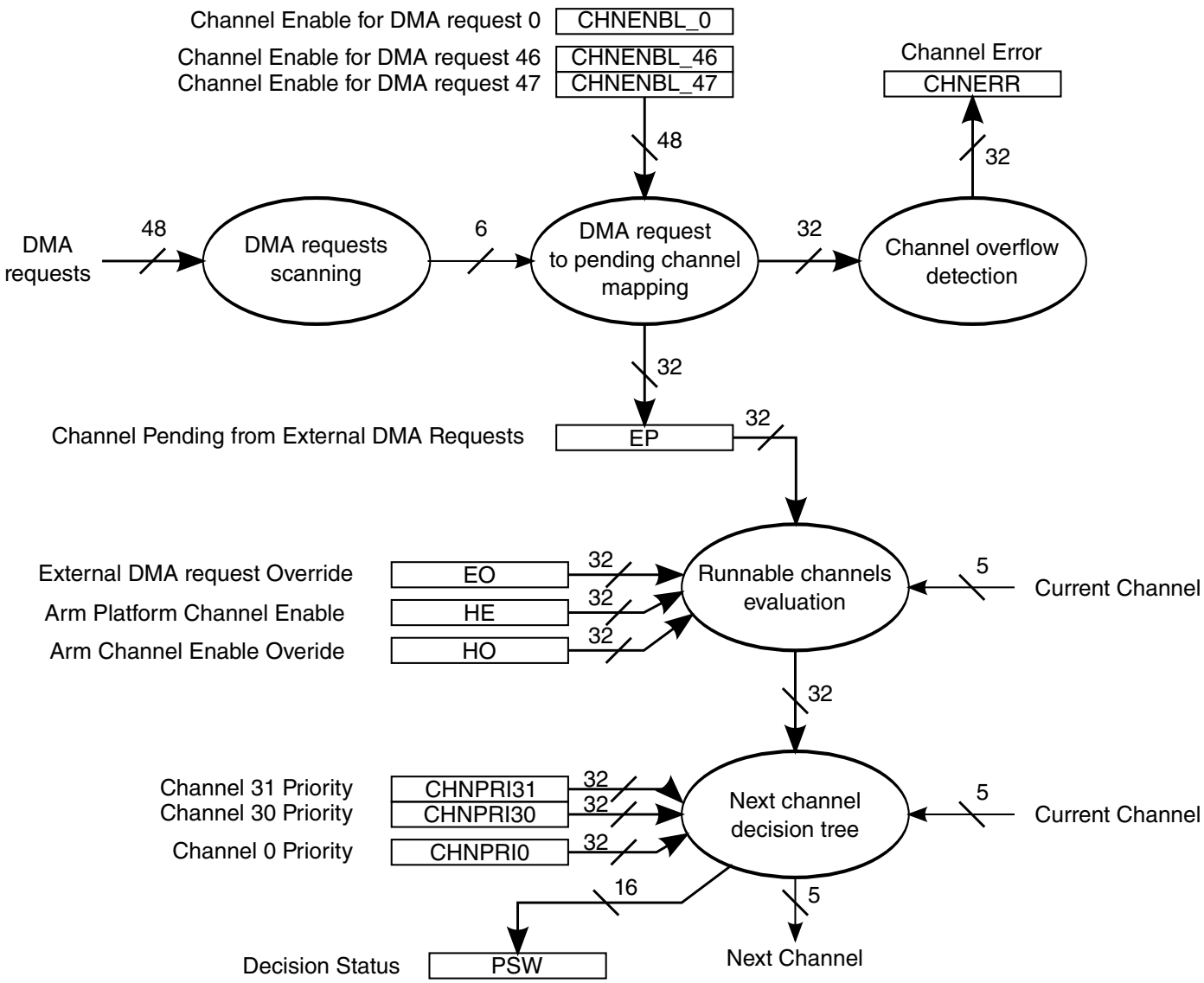


Figure 7-5. SDMA Hardware Scheduler

7.2.3.2.3.2 DMA Requests Scanning

The scheduler contains a 48-bit edge detection device that detects the rising edge of every DMA request and transmits the request number to the next stage.

The DMA requests are assumed to be generated on the same reference clock as the SDMA core clock; they are detected as soon as the signal goes from a 1-to-n-cycles low state to a 1-to-m-cycles high state.

This system is able to detect single-cycle pulses as well as level-based DMA requests such as a FIFO threshold crossing. In this case, the SDMA provides a memory mapped register that can be used by the channel script to monitor the DMA requests lines, and thus determines whether the data transfer is done or not done, and then continues with the transfer or closes the channel.

When several DMA requests are detected at the same time, they are forwarded to the next scheduler stage at the rate of one request per cycle. No request is lost.

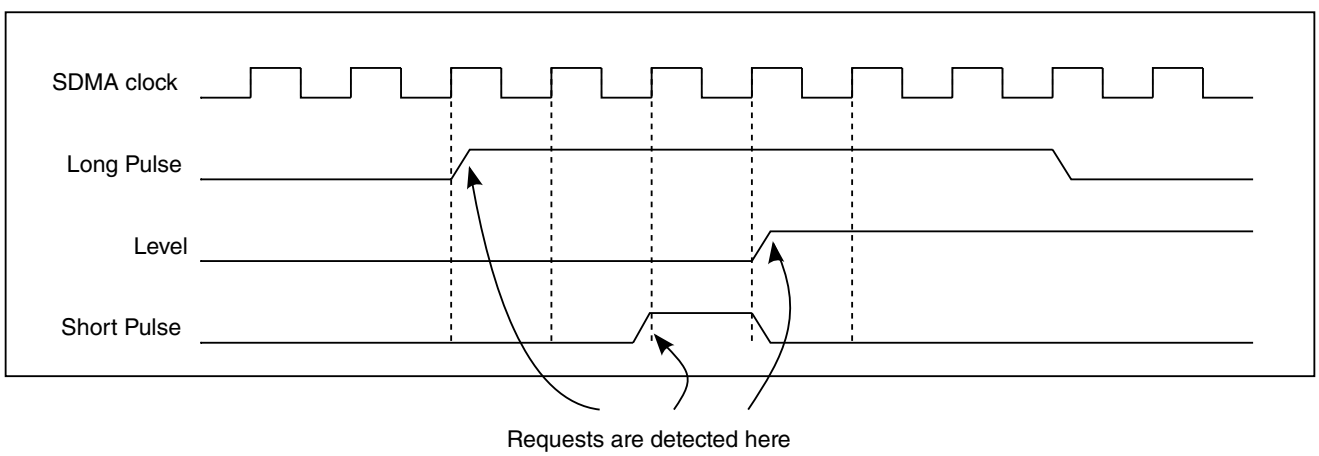


Figure 7-6. Examples of Valid DMA Requests

The DMA request inputs are connected to various sources that depend on the SoC. The exact list of DMA request inputs and their associated number is available in each respective project-specific chapter.

7.2.3.2.3.3 Mapping DMA Requests to Pending Channels

Whenever a DMA request is detected by the first stage, its number is used in the second stage to determine the channels that have to be activated.

This is performed with an array of 48 registers that are 32 bits wide: There are 48 Channel Enable Registers (CHNENBLn), one register per DMA request. The DMA request number selects the Channel Enable Registers, and every bit of this 32-bit register indicates that the corresponding channel must be activated when it is a 1.

This information is passed on the EP register. For every bit of the Channel Enable Register that is set, the corresponding bit of the EP register is also set, and the remaining bits of EP are left unchanged. The transformation of EP is summarized by the following equation:

EP = EP or CHNENBLn

The EP register is used to know which channels require service because they received a DMA request.

Typical contents of the CHNENBLn registers are all 0s, except for a single bit set. For example, a DMA request triggers one channel, but all 0s or several 1s are possible. One DMA request could activate several channels, and the channel execution sequence can be controlled by the channel priorities and numbers, as explained in the next sections. The following table illustrates an example configuration.

NOTE

From the table, the DMA request 0 is programmed to simultaneously trigger channels 0, 1, and 31. Also, DMA requests 30-47 are not used in this example. The remaining channels 2 to 30, are configured to be triggered by DMA requests 29 to 1, respectively.

Table 7-7. Channel Enable RAM Programming Example

DMA Request Number	Channel																															
	31																															0
0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1
1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 7-7. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																														
	31																														0
7	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0
21	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
28	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
29	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
30	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
37	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Table continues on the next page...

Table 7-7. Channel Enable RAM Programming Example (continued)

DMA Request Number	Channel																														
	31																														0
40	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
41	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
42	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
43	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
44	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
45	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7.2.3.2.3.4 Channel Overflow

A channel overflow occurs when a DMA request requires service from channel *n* by setting bit *n* of the register EP, but this bit is already set, meaning channel *n* is already pending. This can come from an overrun/underrun condition.

This detection is possible only when the DMA requests are pulses, because a level-based DMA request stays high until it is serviced, even though an underrun or overrun condition occurs, thus preventing another edge detection of the DMA request.

The channel overflow information is saved in the 32-bit CHNERR register (1 bit per channel). You can configure the SDMA to trigger an interrupt to the Arm platform when there are 1s in CHNERR. Every bit of CHNERR is masked with the corresponding bit of INTRMASK and if it gives a 1, the corresponding bit of INTR is set, triggering the interrupt.

7.2.3.2.3.5 Runnable Channels Evaluation

The EP register is used in conjunction with several other 32-bit registers to determine the channels that are runnable.

Registers EO, DO, HO and HE, are controlled by the Arm platform. EP is controlled by the DMA requests and their mapping to channels.

Several channels may be runnable at any given time. The i^{th} channel is runnable if (and only if) the condition below is true:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

After reset, the HE[i], HO[i], EP[i], and EO[i] bits are all cleared whereas the DO[i] bits are all set. The functions associated with DO are not available for this device. When DO[i] is set, the scheduler condition becomes:

(HE[i] or HO[i]) and (EP[i] or EO[i])

The registers in these equations are controlled as follows:

- Arm platform (host) channel enable flag HE[i] may be set or cleared by the Arm platform with the HSTART and STOP_STAT registers. It can also be cleared by the i^{th} channel script.

  Typical usage is for the Arm platform to set this flag to activate the channel. The flag is cleared by the SDMA core when the transfer is done.

- Externally triggered channel pending flag EP[i] is set by the scheduler when the channel was activated by a DMA request. It can be cleared by the i^{th} channel script.
- The Arm platform channel override flag HO[i] may be set or cleared by the Arm platform. When set, it enables the i^{th} channel to run without the involvement of the Arm platform.

  Typical usage is for the Arm platform to set this flag for channels that do not need Arm platform supervision such as channels that are controlled by DMA request events (EP).

- DO should always be set to 1 so that the runnable channel evaluation considers only HO, HE, EP, and EO.
- Externally triggered channel override flag EO[i] may be set or cleared by the Arm platform. When set, it prevents the i^{th} channel from stopping and stalling on incoming peripheral DMA requests. This is the case when the channel is not handling data transfers with peripherals (for example, a memory to memory transfer).

The SDMA can clear the HE[i], and EP[i] bits by means of a done or notify instruction. The done instruction causes a reschedule; thus, enabling another channel to preempt the current one, while the notify instruction does not. The done and notify instructions can clear either HE[i] or EP[i] (never more than one at a time).

Table 7-8. Runnable Channel Selection Control

Register	Set by	Cleared By
HO	Write to HOSTOVR register	Write to HOSTOVR register

Table continues on the next page...

Table 7-8. Runnable Channel Selection Control (continued)

Register	Set by	Cleared By
HE	Write to HSTART register	Write to STOP_STAT register or by the channel script with the done or notify instructions.
DO	Write to DSPOVR register	Write to DSPOVR register
EO	Write to EVTOVER register	Write to EVTOVER register
EP	Set by external DMA request event input.	By the channel script with the done or notify instructions

7.2.3.2.3.6 Next Channel Decision Tree

The next channel number is computed from the runnable channels list, the current channel number, and their respective priorities.

It is re-evaluated every cycle, but is only used when the current channel yields or terminates by executing a yield, yieldge, or done instruction.

The decision tree is based on the selection of the runnable channel that has the highest priority.

The highest priority channel is selected according to the following rules:

- Runnable channels are sorted by priority.
- If one of the channels with the highest priority had been preempted by a channel with a higher priority, but did not want to yield to a channel of the same priority (for example, it executed a yield, not a yieldge), it is elected as the next channel.
- The channels that belong to the highest priority group are sorted by their number and the channel that has the highest number in this group becomes the next channel. For example, if priorities are the same, channel 31 will be selected before channel 30.

When the current channel requires a reschedule with a yield(ge) or a done instruction, the context switch decision is based on the instruction parameter, the current channel number and priority, and the next channel number and priority. The possible cases are all listed in the following table. The grayed cells correspond to unusual cases that should not occur with a typical usage of the SDMA.

Table 7-9. Channel Switching Decision with a yield, yield(ge), or done

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
yield (done 0)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Current
			Current < Next	Next [1]

Table continues on the next page...

Table 7-9. Channel Switching Decision with a yield, yield(ge), or done (continued)

Instruction	Current Channel	Next Channel	Priorities Comparison	New Running Channel/Comments
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
yieldge (done 1)	Runnable	Not runnable	none	Current
	Runnable	Runnable	Current > Next	Current
			Current = Next	Next[1]
			Current < Next	Next[1]
	Not runnable	Not runnable	none	none[2] (occurs when the channel was disabled by the Arm platform)
	Not runnable	Runnable	none	Next[1] (occurs when the channel was disabled by the Arm platform)
done (done>1)	Not runnable	Not runnable	none	none[2]
	Runnable	Not runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)
	Not runnable	Runnable	none	Next[1]
	Runnable	Runnable	none	Current[3] (occurs when the done instruction does not disable the channel runnable condition)

1. Current channel script execution is stopped, its context is saved; the next channel context is restored and its script execution resumes
2. Current channel context is saved and SDMA enters IDLE mode
3. Current channel context is saved, then restored, and the current channel script resumes execution

Finally, when the SDMA is in IDLE mode and a runnable channel is elected as the next channel, its context is immediately restored and the script execution resumes.

The *combinatorial-decision* tree supports dynamic modifications of the EP, EO, HE, HO, and DO flags as well as dynamic modifications of the channel priorities. The propagation times are detailed in Scheduler Pipeline Timing Diagram.

The decision tree status is available in the PSW register, which is continuously updated. It contains the next channel priority, the next channel number, the current channel priority, and the current channel number. When a priority is read as 0, it means the channel is not runnable.

A few examples of decisions are presented below:

- Channel 31 is running with priority 5, channels 13 and 24 are pending with the same priority 5; channel 24 is eligible as the next channel since 24 > 13.
- Channel 31 is running with priority 7, channels 13 and 24 are pending with priority 5; channel 31 is the next channel because its priority is greater than the other pending channels.
- Channels 7, 23, and 29 are pending with the same priority. Channel 7 is active and runs a yieldge; it is preempted by channel 29. After a period of time, channel 29 runs a yieldge, it is then preempted by channel 23 that is the selected channel since channel 29 is the current channel. Later, channel 23 runs a yieldge and is preempted by channel 29. Channels 23 and 29 will go on switching after every yieldge until one of them terminates. It is only at that point that channel 7 becomes eligible again.
- Channel 11 is running with priority 3, and channel 15 is pending with priority 4. When the channel 31 script executes a yield instruction, it gets preempted by channel 15; then channels 6 and 18 with priority 3 become pending. Because channel 11 was preempted after executing a yield and there is no pending channel with a strictly greater priority, it is eligible as the next channel (although its number 11 < 18).

7.2.3.2.3.7 Scheduler State Diagram

The Figure 7-7 summarizes the behavior of the SDMA scheduler with details about the exact mechanism of the priority decision tree. It is important to understand the scheduler is a hardwired pipeline, which means all the stages are performed simultaneously every cycle, but a change on any given stage is reflected on the next stage after the delays presented in Scheduler Pipeline Timing Diagram.

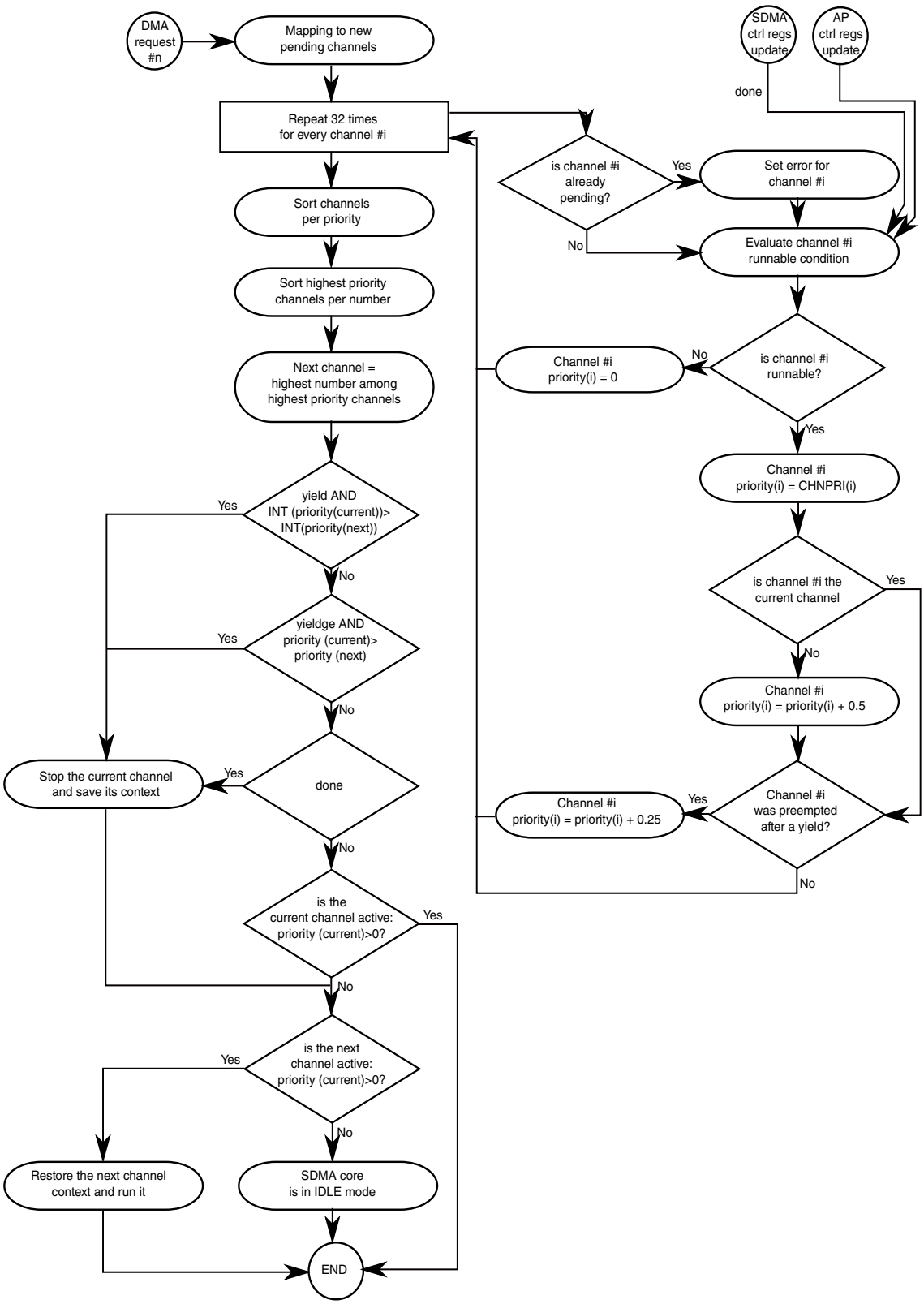


Figure 7-7. Scheduler State Diagram

7.2.3.2.3.8 Scheduler Pipeline Timing Diagram

The SDMA scheduler process of DMA-request and control-register modifications is not immediate.

The figure below shows the exact delays of all the tasks. The reference clock is the SDMA core clock.

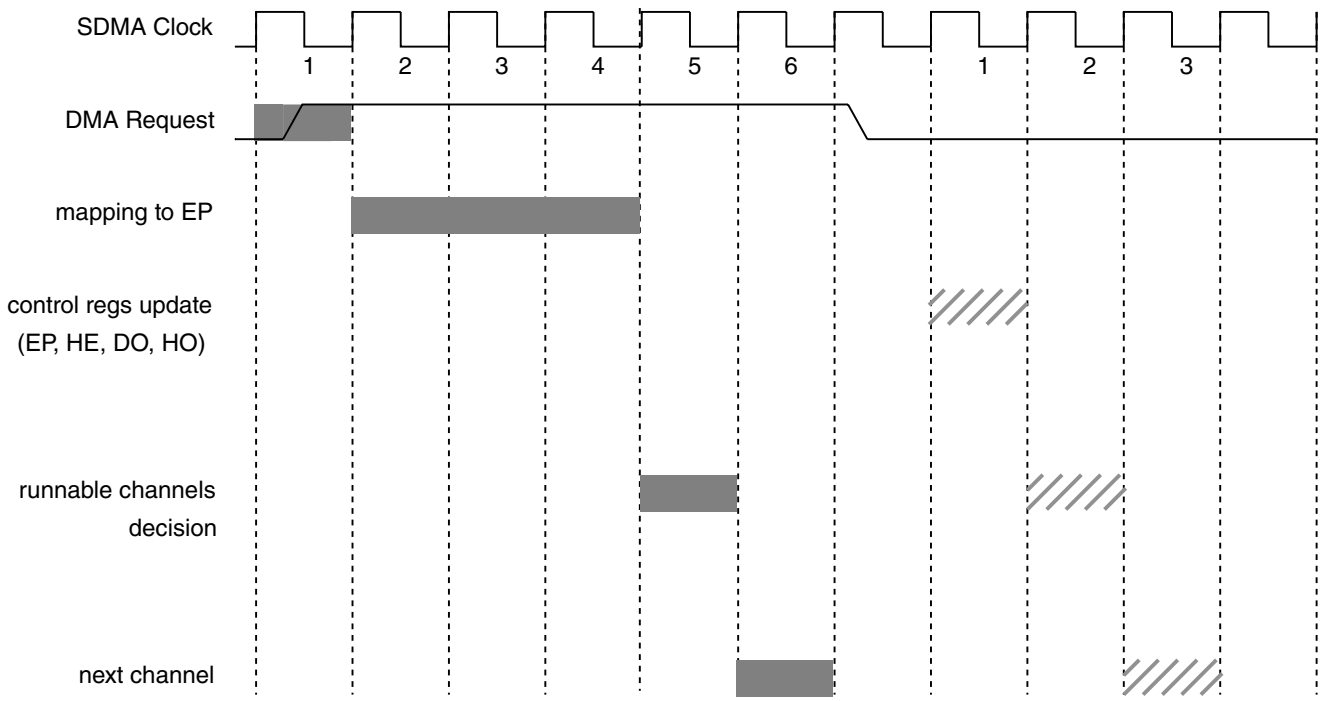


Figure 7-8. Scheduler Timing Diagram

Two numbers can be inferred from this timing diagram. First, it takes six SDMA core clock cycles to update the next channel from a DMA request. Second, it takes three SDMA core clock cycles to update the next channel from a direct modification of the condition registers (EP, DO, HE, or HO) by any processor. The processors that can modify these bits include SDMA with a done instruction or the Arm platform with a write access through the corresponding control port on their respective peripheral bus).

7.2.3.2.3.9 Channel-DMA Request Mapping

The 48 DMA request inputs to the SDMA scheduler are listed in project-specific chapters. Refer to the respective chapters for this information.

7.2.3.2.3.10 Examples: How to Start a Channel

A channel can be started when the following equation is true for channel *i*:

(HE[i] or HO[i]) and (DO[i]) and (EP[i] or EO[i])

Once this equation is true, the scheduler can start this channel according to the priority of all pending channels. Several examples of configuration are listed below:

1. To start a channel triggered by Arm platform software:
   - Initially, configure HO[i]=0, DO[i]=1, and EO[i]=1 using registers indicated in Table 7-8.
   - Arm platform software triggers the channel by writing to the HSTART register to set HE[i]=1, thereby setting the above equation true.
2. To start a channel triggered by DMA request event.
   - Initially, configure HO[i]=1, DO[i]=1, and EO[i]=0 using registers indicated in Table 7-8.
   - The DMA request is asserted to trigger the channel by setting EP[i]=1, which makes the above equation true.

7.2.3.2.4 Context Switching

On execution of a done or yield(ge) instruction, the current channel may be changed either because it has finished (which necessarily happens when the done instruction is executed), or it was preempted by a higher priority channel (which is possible but not systematic when the yield(ge) is executed).

Upon a channel change the SDMA goes through a context switch procedure.

When the current channel yields or ends, the context for that channel is saved into the context RAM locations for that channel. When the next channel starts running, its context is first restored from RAM.

Since context RAM is not yet initialized by reset, there will be no context restore at the beginning of the first channel (bootload channel) run after reset. It is expected that the bootload channel will be used to initialize the context for all other channels. When the bootload channel finishes running or yields, SDMA will enter its SAVE state and save that channel's context into RAM. Then, if the bootload channel is called again later, the context will be restored from RAM when the channel starts again.

The context structure for each channel is defined in Context Switching-Programming and Table 7-14. There will be one context area reserved for each channel. When a channel ends or yields, the SDMA core registers are automatically saved into the context RAM and later restored from the context RAM when the channel is next run. The total RAM

space reserved for 32-channel contexts is either 3K or 4K depending on whether the SMSZ bit is set in the CHN0ADDR register, which enables an additional 8 words of scratch RAM for each context.

7.2.3.2.4.1 Context Switch Modes

The exact procedure to save the context of the old channel, and to restore the context of the new channel depends on the context switch mode selected by the Arm platform in the CONFIG control register.

The following are the context switch modes:

- By default, the "dynamic" context switch is set. This mode provides the most efficient context switch for an average of eight cycles to stop the current channel, save its context, restore the next channel context, and resume its execution. It consists of saving modified registers of the current channel in the background (for example, during the channel execution)-which leaves very few registers to save when the switch is decided-resuming execution of the next channel as soon as possible (for example, when the minimal set of registers is restored), and continuing the restore phase during this execution.
- In "dynamic with no loop" mode, the same principle is followed except the modified registers are only saved in the background when the loop flag is not set. This mode offers almost the same effectiveness as the previous one, but it prevents the system from accessing the RAM during loops to save power. This is the recommended mode for an efficient context-switch when the loop bodies are short.
- In "dynamic power" mode, no background saving is performed, which reduces power consumption to the minimum. The modified registers are only saved when the context switch starts. The restore phase is the same as before. This is the mode that achieves the optimal power consumption at the cost of a slower context-switch.
- In a "static" context switch, all the registers are saved when a context switch is decided, and all the registers are restored before starting the execution of the new channel. This mode enables a predictable behavior of the context switch since all the registers are restored prior to the channel start and all registers are saved after the channel termination.

NOTE

Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized during the context SAVE phase when the channel is done or yields. Subsequent calls to the same channel or different channels may use any of the dynamic context modes. This will ensure that all context locations for the bootload

channel are initialized, and prevent undefined values in context RAM from being loaded during the context restore if the channel is re-started later.

7.2.3.2.4.2 Context Switch Procedure

The Program Control Unit goes into the *save* state, the current context is spilled into memory, and the next channel context is restored according to the context-switch mode that was selected by the Arm platform.

The context switch procedure is as follows:

1. Load the current context's spill base address.
2. Spill the modified registers of the current channel to memory according to the selected context switch mode while the channel is running.

   On a done or yield(ge) that causes the channel preemption, the PCU goes into the *save* state. In *static* mode, all the registers are saved; whereas, in either *dynamic* mode, the registers that were modified but not yet saved are then saved, and the PCU registers and flags are finally saved.

3. Put the SDMA core into *sleep* and wait for new channels to be serviced. This step is skipped if there are pending channels when the current channel is saved.

   As soon as there is at least one pending channel, the PCU goes into its *restore* state to restore the context of the channel that was elected by the scheduler.

   Once a channel is elected, it remains the current channel until its script requests a rescheduling operation with a done or yield(ge) instruction. That means the current channel cannot be modified by the Arm platform, even if it is no more runnable or if its priority is modified.

   The Arm platform can however force a reschedule by writing the corresponding bit in the CONFIG register, which has the same effect as if the script had executed a done instruction. That feature should only be used to stop the SDMA in emergency cases.

4. Load the context base-address of the new channel.

   In "static" mode, all the registers are restored. In either "dynamic" modes, only the PCU registers are restored.

   The new channel is running. In "static" mode, no more activity regarding context restoring or saving is performed. In either "dynamic" modes, the registers are restored in the background every time an access to the context RAM is possible, and

priority is given to restoring the registers that are required by the next instruction to be executed. When a register has not been restored and the next instruction needs it, this instruction gets stalled until the register was restored.

In "dynamic" and "dynamic with no loop" modes, background saving of dirty registers is performed every time an access to the context RAM is possible and allowed by the context switch mode.

NOTE

The contents of a channel context space in the context RAM depends on the selected context switch mode. In "dynamic" and "dynamic with no loop" modes, the contents of the context RAM tend to match the contents of the SDMA registers (except for the PCU registers and flags that are never saved in the background). In "dynamic power" and "static" modes, the contents of the context RAM remain unchanged until the channel terminates with a done or gets preempted.

7.2.3.2.4.3 Context Map in Memory

Refer to Context Switching-Programming.

7.2.3.3 Functional Units

The functional units are small systems that are used by the SDMA core to handle data transfers between the core and a bus domain external to the SDMA.

The SDMA core is able to control and exchange data with these systems by sending instructions and reading or writing data from/to the functional units' registers via the FUBUS. This is done with the ldf and stf instructions.

The following sections provide introductions to the available functional units. Functional Units Programming Model provides descriptions the functional units' behaviors.

7.2.3.3.1 Burst DMA Unit

The burst DMA unit enables the SDMA core to perform data transfers to and from the Arm platform memory.

It is optimized for accessing SDRAM-like devices. It does not provide control to assign a privilege level to the DMA access. The burst DMA unit provides the SDMA with means to do the following:

- Perform up to 8-beat read and write bursts to the Arm platform memory, which optimizes throughput when accessing SDRAM-type devices because of an internal, 36-byte FIFO
- Access the Arm platform memory at once or twice the SDMA core frequency
- Copy data from one Arm platform memory location to another Arm platform memory location at the Arm platform bus speed, which provides a very high throughput
- Control the method for addressing the Arm platform memory (automatic increment of addresses or frozen addresses-the former aimed at accessing RAM-like memory and the latter aimed at accessing single-address FIFOs)
- Enable or disable automatic prefetch when reading data from the Arm platform memory. When the prefetch mode is selected, the burst DMA automatically triggers external bursts to fill its FIFO without waiting for the SDMA core to request the corresponding data, greatly improving throughput.
- Rely on the DMA to automatically flush its FIFO content when there is enough data to generate an 8-beat burst to the Arm platform memory. Or, it forces a flush when a data transfer must terminate.
- In the former case, the SDMA core may only be stalled when it tries writing data and there is not enough room left in the FIFO. In the latter case, the core is stalled until the data is effectively written to the Arm platform memory.

  In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory. This error status is retrieved by a later access to the burst DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory is caught.

- Handle address alignment issues between the Arm platform memory map and the SDMA core data. This enables the core to read or write 32-bit data from the burst DMA, whereas the corresponding Arm platform address is not 32-bit aligned. This drastically improves the SDMA scripts' efficiency since the same loop that transfers 32 bits at a time can be used regardless of the start and end addresses in the Arm platform memory space.

This unit structure and registers are described in Burst DMA Structure and Burst DMA Registers.

7.2.3.3.1.1 Burst DMA Structure

The burst DMA is essentially made up of a 36-byte FIFO, address registers, and a controlling state-machine. The 36-byte FIFO enables eight-word buffering with address alignment, and the state-machine manages clock adaptation when required.

The burst DMA is depicted in the figure below.

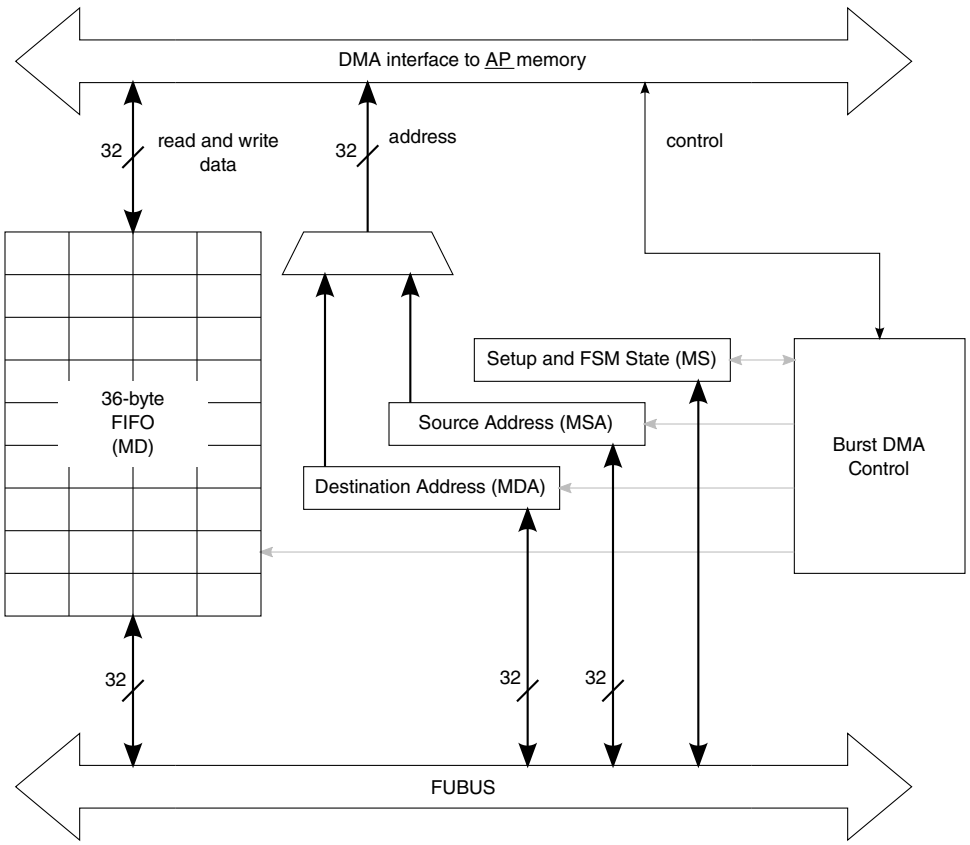


Figure 7-9. Burst DMA Structure

7.2.3.3.1.2 Burst DMA Registers

There are four registers, as follows, that may be accessed from the SDMA core:

- MSA (Memory Source Address) - Holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads new data from the FIFO.
- MDA (Memory Destination Address) - Holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes new data into the FIFO.
- MD (Memory Data) - Labels the 36-byte FIFO access point: Reading a byte, halfword, or word from MD respectively retrieves the first 1, 2, or 4 bytes of the

FIFO (for example, the bytes that were stored first by the DMA state-machine when transferring data from the Arm platform memory).

- When the FIFO does not hold as many bytes as required by the SDMA core, the core is stalled until the missing bytes are read from the Arm platform memory. In the case of prefetch mode, the DMA controller decides when it should start a burst to Arm platform memory in order to reduce the risk to not have the required data for the future accesses of the core. When there is no prefetching, a burst is triggered when the required data is not available in the FIFO.

  Writing a byte, halfword, or word to MD stores 1, 2, or 4 bytes, respectively, at the end of the FIFO (for example, these bytes are transmitted to the Arm platform memory after all the other bytes that were previously stored in the FIFO). When the FIFO does not have enough room left to hold the written data, the SDMA core is stalled until a sufficient amount of FIFO contents are flushed out to the Arm platform memory. Flushing is decided by the DMA controller when there are enough bytes in the FIFO to perform the largest allowed burst to Arm platform memory (the exact size depends on the burst start address and the AHB 1 Kbyte boundary rule). However, the SDMA core has the ability to force the flushing operation at any time, for example, when at the end of the data transfer, prior to channel closure.

- MS (Memory Setup) - Contains the state of the burst DMA control, the two flags that define whether each address register is incremented after every access to the external memory, and another flag that is set when a bus error occurred.

7.2.3.3.1.3 Burst DMA Data Transfers

Three typical usages have been identified that involve the burst DMA: the data transfer startpoint, the endpoint, or both.

Every case requires a different procedure, as listed in the following sections:

7.2.3.3.1.3.1 Data Retrieval from the Arm platform Memory

The following steps retrieve data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the source address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the source address register itself (MSA).
- Read data from the FIFO using the *ldf MD* instruction as many times as needed. If an error occurred during the fetch from Arm platform memory, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from the FIFO.

7.2.3.3.1.3.2 Storing Data Into the Arm platform Memory

The following steps store data from Arm platform memory using the burst DMA unit:

- Set up the MS flags to reflect the mode for the destination address (incremented or frozen according to the type of accessed device: memory or peripheral FIFO), then initialize the destination address register itself (MDA).
- Store data into the FIFO using the *stf MD* instruction as many times as needed.
- When the transfer is finished and if the DMA worked in automatic flush mode, force the flush of the FIFO. This instruction is stalled until all the FIFO data is effectively sent to the Arm platform memory and the error status of the transfer is available in the DF flag.

7.2.3.3.1.3.3 Transferring Data Between Two Arm platform Memory Locations-Burst DMA Unit

The following steps copy data between two Arm platform memory locations using the burst DMA unit:

- Set up the MS flags to reflect the modes for the source and destination addresses (all the combinations are possible), then initialize the source address register (MSA) and the destination address register (MDA). Both addresses must be word-aligned.
- Use as many *stf MD* instructions with the *COPY* flag as needed. Every instruction triggers a burst read of a given number of words from the source address (this number is provided to the burst DMA via the SDMA core general purpose register, which is referenced in the *stf* instruction). Once all the data is loaded into the FIFO, the DMA empties it with a write burst of the same count to the destination address. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the burst DMA to check the error status.

7.2.3.3.2 Peripheral DMA Unit

The peripheral DMA unit is the second functional unit that connects the SDMA to the Arm platform memory.

Unlike the burst DMA, it does not support burst transfers and is optimized for accessing peripherals. It does not provide control to assign a privilege level to the DMA access. Its feature list comprises the following:

- Access to the Arm platform peripherals or memory at once or twice the SDMA core frequency

- Data copy from one Arm platform memory location to another Arm platform memory location at memory bus speed, improving throughput
- Control of the method for addressing the Arm platform memory (automatic increment or decrement of addresses or frozen addresses, the first ones aimed at accessing RAM-like memory and the last one aimed at accessing single-address FIFOs)
- Selectable automatic prefetch when reading data from the Arm platform memory. In prefetch mode, the peripheral DMA automatically fetches another data-without waiting for the SDMA core to request it-when its data register is empty, which improves the throughput
- Selectable automatic flush. In this mode, the SDMA core may only be stalled when it tries writing data and the previous write operation is not finished yet; whereas, in forced flush mode, the core is stalled until the data is effectively written to the Arm platform memory.
- In automatic flush mode, the core receives an acknowledge that does not reflect the actual error status when the data is effectively written into the Arm platform memory or the peripheral. This error status is retrieved by a later access to the peripheral DMA. Terminating a write data transfer with a forced flush command guarantees that any bus error to the Arm platform memory has been caught.

This unit structure and registers are described in Peripheral DMA Structure and Peripheral DMA Registers.

7.2.3.3.2.1 Peripheral DMA Structure

The peripheral DMA is made up of a 32-bit data register, two address registers, and a controlling state-machine. The state-machine manages clock adaptation, when required.

It is shown in the following figure.

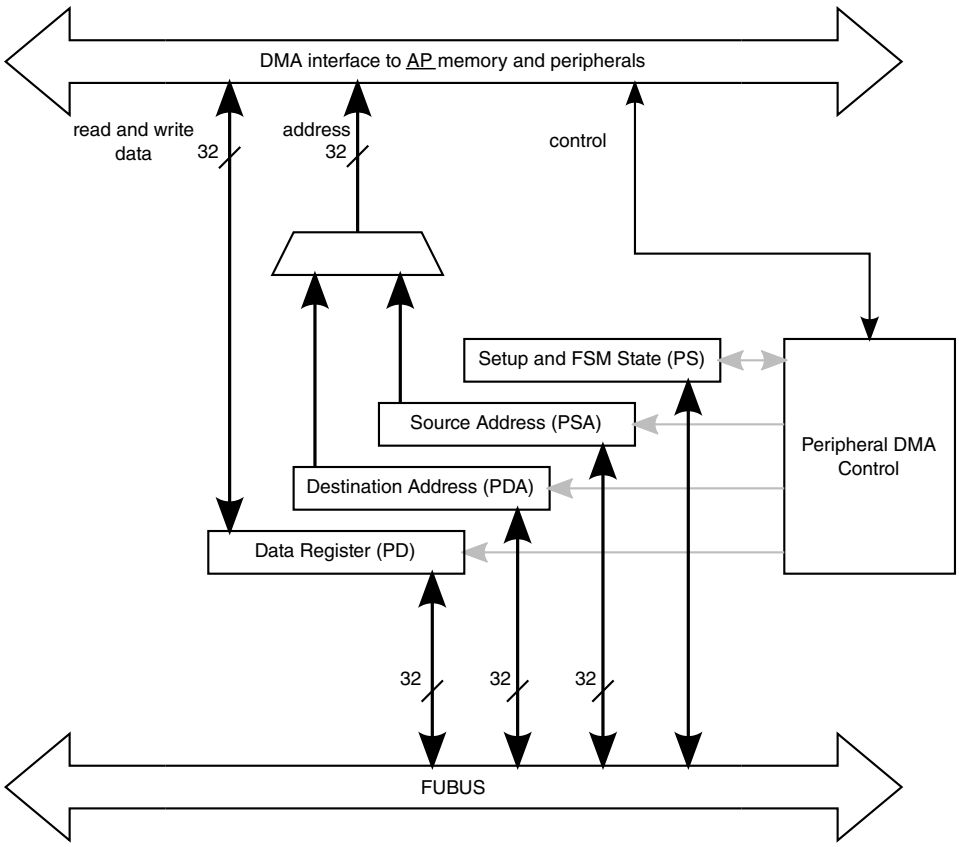


Figure 7-10. Peripheral DMA structure

7.2.3.3.2.2 Peripheral DMA Registers

According to Figure 7-10, the peripheral DMA has four registers that may be read or written by the SDMA core:

- *PD (Peripheral Data)* is the DMA 32-bit data register.
- *PSA (Peripheral Source Address)* holds the source byte address in the Arm platform memory map for reading data from this location. This register is automatically modified every time the core reads a new data from PD.

- *PDA (Peripheral Destination Address)* holds the destination byte address in the Arm platform memory map for writing data to this location. This register is automatically modified every time the core writes a new data into PD.
- *PS (Peripheral Setup)* contains the state of the peripheral DMA control, two configuration fields that define the way address registers are modified after every data access, two additional configuration fields that define the data size to access the source and destination devices, and another field that contains the latest transfer error status.

7.2.3.3.2.3 Peripheral DMA Data Transfers

There are three typical usages that involve the peripheral DMA, whether it is the data transfer start-point, endpoint, or both.

Every case requires a different procedure, as described in Data Retrieval from the Arm platform Memory or Peripheral, Storing Data into the Arm platform Memory or Peripheral, and Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit.

7.2.3.3.2.3.1 Data Retrieval from the Arm platform Memory or Peripheral

The following steps retrieve data from Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the source (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the source address register itself (PSA) with an address that is aligned to the programmed data size.
- Read data from PD using the ldf PD instruction as many times as needed. If an error occurs during the fetch from the Arm platform memory or peripheral, the DMA control tags the error status on the data and the SDMA core SF flag is set when reading this data from PD.

7.2.3.3.2.3.2 Storing Data into the Arm platform Memory or Peripheral

The following steps store data to Arm platform memory using the peripheral DMA unit:

- Set up the PS fields to reflect the mode and data size for the destination (incremented, decremented, or frozen address register; 8-bit, 16-bit, or 32-bit data transfers), then initialize the destination address register itself (PDA) with an address that is aligned to the programmed data size.

- Store data into PD using the *stf PD* instruction as many times as needed.
- When the transfer is finished and if the peripheral DMA worked in automatic flush mode, force the flush of PD. This instruction is stalled until PD contents are effectively sent to the Arm platform memory or peripheral, and the error status of the transfer is available in the DF flag.

7.2.3.3.2.3.3 Transferring Data Between Two Arm platform Memory Locations-Peripheral DMA Unit

The following steps copy data between two Arm platform memory locations using the peripheral DMA unit:

- Set up the PS fields to reflect the modes and data size for the source and destination addresses (all the combinations of addressing modes are possible, but both data sizes must be identical), then initialize the source address register (PSA) and the destination address register (PDA). Both addresses must be aligned with the programmed data size.
- Use as many *stf PD* instructions with the *COPY* flag as needed. Every instruction triggers a single read from the source address; a single write of the received data immediately follows. The DMA acknowledges prior to instruction completion, which frees the SDMA core for other tasks at no delay cost.
- Once the transfer is done, there should be a final access to the peripheral DMA to check the error status.

7.2.3.4 SDMA Security Support

The SDMA provides support to SDMA software to block unauthorized updates to the scripts in RAM.

SDMA supports the following Security modes:

- Open Mode: has full control to load scripts and context into SDMA RAM. This is the default mode.
- Locked Mode: The Arm platform loads scripts and channel contexts at startup when it is still executing known safe software. When finished, it locks the SDMA to prevent further updates to RAM and selected registers. More details described in Locked Mode.

7.2.3.4.1 Locked Mode

The LOCK bit in the SDMA_LOCK register provides support for SDMA scripts to freeze RAM contents after the initial bootload routine to prevent future unauthorized updates to SDMA RAM.

After initial RAM contents are uploaded, Arm platform software can set the LOCK bit to secure the RAM contents to prevent future updates by an unauthorized. After the LOCK bit is written with a '1', the SDMA is "locked" until reset.

The LOCK bit can be read in the SDMA's internal memory map in the LOCK register (see Section SDMA LOCK (SDMAARM_SDMA_LOCK)). SDMA scripts which load information into RAM can check the value of the LOCK bit to determine if an upload to RAM is allowed. If not allowed, the script can refuse to allow the request to copy data into the RAM to continue. The exact use of the LOCK bit in SDMA scripts for security control will be described in SDMA software documentation (see SDMA Scripts).

While SDMA is locked, attempts to write to the SDMA_LOCK, CHN0ADR, ILLINSTADDR, and ONCE_ENB registers will be ignored. All registers remain readable. Writes to other registers are still allowed.

Once the SDMA is locked, the LOCK bit can only be cleared by a reset. A hardware reset will always clear the LOCK bit. A software reset initiated by writing to the RESET register will only clear the LOCK bit if the SRESET_LOCK_CLR bit in the SDMA_LOCK register is set. Since SDMA_LOCK register cannot be updated if SDMA is locked, the SRESET_LOCK_CLR bit must be configured before setting the LOCK bit. The SREST_LOCK_CLR bit will also be cleared by resets that clear the LOCK bit.

The SDMA RISC core uses the ILLINST and CHN0ADDR registers as pointers to determine where to jump to after an illegal instruction or upon boot after a reset. The LOCK bit prevents updates to these registers to protect against unauthorized changes to these pointers.

While SDMA is locked, the ONCE_ENB register cannot be written to prevent the OnCE under Arm platform control from being used to gain access to SDMA internal memory. If Arm platform control of the OnCE is enabled before setting the LOCK bit, the Arm platform can use the ONCE for debug purpose after LOCK is set.

7.2.3.5 OnCE and PCU Debug States

The SDMA has two different debug modes in which the OnCE performs debug instructions.

Refer to Figure 7-4 for an example of the PCU states in debug. The following are the two debug states:

- When a channel is running (that is, when CCR and CCPRI are different from 0, which can be read in the PSW register), SDMA can execute a SoftBkpt instruction from the channel script or receive a debug request. When either happens, the SDMA enters its "Classical" *Debug* state, which is described in OnCE and Real-Time Debug.
- When a channel is not running, the SDMA can be in *Sleep* state or in *Sleep after Reset* state. If a debug request is sent to the core, it enters its *Debug in Sleep* state. This debug mode works similarly to the "Classical" *Debug* state, except it returns to the original state (*Sleep* or *Sleep after Reset*) when the debug mode is left via the exec_core instruction of the OnCE. From this *Debug in Sleep* state, the SDMA can execute a program whereas no channel is running. If a new debug request is sent to the core or if a SoftBkpt is executed, it comes back to this *Debug in Sleep* state.

The OnCE is provided with several instructions that can be executed when the core is in either debug state. The following table summarizes the behavior of these OnCE debug instructions. There exists other secondary OnCE instructions that are described in OnCE and Real-Time Debug.

Table 7-10. SDMA in Debug Mode

Instruction	Debug	Debug in Sleep
exec_once	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.	exec_once <instruction> SDMA executes the <instruction> and returns to the *Debug in Sleep* state. The Program Counter (PC) is not incremented. This command must not be used with an instruction that modifies the PC value.
run_core	run_core <instruction> SDMA executes the <instruction>, leaves the *Debug* state and continues executing the channel script from the position where it stopped. This command must not be used with an instruction that modifies the PC value.	run_core <instruction> SDMA executes the <instruction> and returns to its *Sleep* or *Sleep after Reset* initial state. This command must not be used with an instruction that modifies the PC value.
exec_core	exec_core <instruction> It is similar to run_core except it requires an instruction that changes the PC value (jump, branch...): the SDMA jumps to the new PC value, leaves the *Debug* state and starts executing instructions from this new PC value.	exec_core <instruction> If the previous state was *Sleep after Reset*, the SDMA returns to this state, and Chn0Addr value overrides the PC value. Otherwise, the SDMA jumps to the new PC value and starts executing instructions from this new PC.

NOTE

The feature exec_core in *Debug in Sleep* after *Sleep after Reset* was added for the Channel boot (channel 0) to allow the debugger to return to *Sleep after Reset* state with a new PC

value. The SDMA will be ready to boot at the Chn0Addr address.

7.2.3.6 SDMA Clocks and Low Power Modes

The SDMA receives several root clocks from the SoC clock controller block and performs adaptive clock gating to optimize its power consumption. From a user standpoint, clock gating and power mode selection are fully automatized inside the SDMA.

Root clock control is available from the SoC clock controller block.

There are numerous clock sources that are used in the SDMA. They belong to one of two possible clock domains listed in the following table, and have frequency constraints within each domain. Clocks are considered asynchronous between domains.

Within the Arm platform/SDMA clock domain, all clocks must come from the same DPLL. The Arm platform DMA interfaces (peripheral DMA and burst DMA) receive their clock from the Arm platform DMA clock source whose frequency can be once or twice the frequency of the SDMA core clock. The DMA interfaces are designed to work at the Arm platform DMA frequency, but the SDMA core is physically limited to a maximum 104 MHz frequency. Since this is lower than the maximum Arm platform DMA frequency, the SDMA core clock is tied to the Arm platform peripheral clock frequency.

The Arm platform Peripheral Bus Clock source must be an exact sub-frequency of the SDMA Core clock source (any integer value greater or equal to 1).

Table 7-11. Clocking Scheme

Clock Domain	Source Clock	Comments
Arm platform	SDMA core (SDMA main core)	Source clock for the core and all its operations; this clock is thus used by most of the SDMA sub-blocks.
	Arm platform DMA	DMA interface for the peripheral DMA and the burst DMA. It is balanced with the main clock source, and its frequency is either once or twice the main clock frequency.
	Arm platform peripheral	Connection to the Arm platform peripheral bus. It is a sub-frequency of the main clock frequency.
JTAG	TCK	Clock for JTAG access, limited to maximum of 1/8 of the SDMA core clock frequency.

The JTAG clock is sampled by the SDMA main clock to determine its rising edge. This simplifies design and clock management, but it also adds a ratio constraint between those two clocks. It is guaranteed the JTAG interface works properly when the frequency of TCK is lower than $1/8^{th}$ of the frequency of the SDMA main clock (which is about 8 MHz when the SDMA core clock frequency is 66 MHz).

7.2.3.6.1 Clock Gating and Low Power Modes

The SDMA automatically performs power saving without requiring user involvement. It implements two levels of automatic clock gating.

7.2.3.6.1.1 Coarse Clock Gating

Every sub-block clock comes from one of the five available sources, and is gated with the sub-block specific enabling condition.

The following table displays the sub-block clocks and their source. It also indicates the relationships that may exist between different sub-blocks clock enables.

Table 7-12. Sub-blocks Clocks

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
Core	SDMA Main Core	The core sub-block clock is running when the core is not in one of its sleep states (Sleep or Sleep after Reset) or there is a pending channel. Typically, the core sub-block clock is stopped once all the channels are processed and the core enters its sleep state. A new pending channel awakes the core sub-block clock.	None
Memories	SDMA Main Core	The clock activation only occurs during a core access.	Disabled when Core sub-block clock is disabled or no memory access in progress
Scheduler	SDMA Main Core	Its clock only runs when scheduling is needed: for example, when there are pending channels, upon reception of a DMA request, and anytime the Arm platform modifies the channel running conditions.	None
Arm platform Control	SDMA Main Core & Arm platform peripheral	The Arm platform peripheral clock is solely used to determine the frequency ratio with the SDMA main clock. The control registers' clock is based on *SDMA main clock*; it is active when the Arm platform or the SDMA modifies the contents of one of these registers.	None
Burst DMA	SDMA Main Core & Arm platform DMA	The burst DMA has two clocks: The first clock is derived from the SDMA main core clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled	Disabled when Core sub-block clock is disabled

Table continues on the next page...

Table 7-12. Sub-blocks Clocks (continued)

Sub-block	Source Clocks	Enabling Condition and Comments	Related Enabling Conditions
		during active phases of data transfers (for example, these clocks are turned off when the burst DMA is not used by the running channel script).	
Peripheral DMA	SDMA Main Core & Arm platform DMA	The peripheral DMA has two clocks: The first clock is derived from SDMA main clock and drives registers that are connected to the FUBUS. The second clock is derived from the Arm platform DMA clock and drives registers that are connected to the Arm platform DMA bus outside the SDMA. Both clocks are enabled during active phases of data transfers (for example, these clocks are turned off when the peripheral DMA is not used by the running channel script).	Disabled when Core sub-block clock is disabled
OnCE	SDMA Main Core	The OnCE clock is derived from main source clock. It is disabled by default. In order to use the OnCE, its clock must be explicitly turned on, either by enabling the OnCE access from the Arm platform peripheral bus (register ONCE_ENB), or by driving the clk_gating_off input pin high. This is a SDMA input whose driver depends on the SoC implementation (typically a JTAG controller). The OnCE also receives the TCK input, which is the JTAG clock. It does not use it as a functional clock; the TCK input is sampled instead. Refer to Synchronization Implementation.	When enabled, all other clocks are systematically on (clock gating is off)

7.2.3.6.1.2 Refined Clock Gating

The SDMA implements a second level of clock gating on a register-per-register basis.

Unlike the first level that covers all the SDMA flip-flops, except the synchronizers (only five flip-flops are always running), the second level is only available for eligible registers, which amounts to about 90% of the SDMA flip-flops.

These gated registers are only clocked when the hardware logic detects a new data loading. This additional gating further reduces dynamic power consumption.

7.2.3.6.1.3 Low Power Modes and User Control

Power savings are automatically managed by the SDMA hardware without any user involvement; however, one can distinguish three different power modes: SLEEP, RUN, and DEBUG.

The following table describes these modes, and shows how to switch from one mode to another.

Table 7-13. Power Modes

Power Mode	Sub-blocks							Comments
	Core	Memories	Scheduler	Arm platform Control	Burst DMA	Peripheral DMA	OnCE	
SLEEP	off[1]	off	wait[2]	wait	off	off	off	Set when the PCU state is either *Sleep* or *Sleep after Reset* and the SDMA is not in DEBUG mode. This is the default mode after reset.
RUN	on[3]	wait	wait	wait	wait	wait	off	Set for the other PCU states that are reachable out of debug: *Program*, *Data*, *Change of Flow*, *Error in Loop*, *Debug*, *Functional Unit*, *Save,* or *Restore*.
DEBUG	on	on	on	on	on	on	on	Set regardless of the PCU state when clock gating is turned off to use the OnCE features (either *clk_gating_off* pin high or ONCE_ENB[0] set).

1. *off*: no clock
2. *wait*: only clocked when accessed or stimulated
3. *on*: clock is always running

It is possible to control the SDMA power mode. The procedures to force the SDMA into either mode are described in SLEEP Mode.

7.2.3.6.1.3.1 SLEEP Mode

This is the default mode after reset; therefore, resetting the SDMA forces this mode.

However, the common procedure is as follows:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Disable all channels (via the STOP_STAT control register, and the HO, DO, EO if necessary).
- Wait for the active channels to complete or force a reschedule via the reschedule bit in the RESET register.
- The SDMA is in SLEEP mode making it possible to completely shut off its clock from the chip level clock controller using the procedure described in Stop Mode Response.

7.2.3.6.1.3.2 RUN Mode

This is the default mode when a channel is running:

- Ensure the *clk_gating_off* pin is low and ONCE_ENB[0] is cleared.
- Activate at least one channel (via the HSTART control registers, a DMA request, and/or the HO, DO, EO register bits).

7.2.3.6.1.3.3 DEBUG Mode

The DEBUG mode must be set when one needs to use the debugging facilities of the SDMA.

- Ensure the SDMA clocks are running from the CCM.
- Set the *clk_gating_off* pin high or use the SDMA to set ONCE_ENB[0].

7.2.3.6.1.4 Stop Mode Response

The SDMA receives a stop request from the chip level clock controller. This request may be asserted when the chip enters the stop low power mode.

If the SDMA is running when the request is received, then the SDMA will complete all pending channels before returning to the SLEEP state. The SDMA sends an acknowledgement to the clock controller when the SLEEP state is entered indicating that the SDMAs clocks can be turned off.

7.2.3.6.2 Reset

After reset (either received from the reset block or a software reset required by the Arm platform), the SDMA is in IDLE mode. It will start its boot code located at address 0 once a channel is activated.

Activating a channel can be done by the Arm platform after programming a positive priority and setting the channel bit in the EVTPEND register.

There will not be a context RESTORE for the first channel (bootload channel) called after a reset because the context data in RAM has not been initialized. Static context mode should be used for the first channel called after reset to ensure that the all context RAM for that channel is initialized. Subsequent calls to the same channel or different channels may use any of the dynamic context modes

7.2.3.7 Software Interface

Appendix A fully describes the SDMA Application Programming Interface (API).

7.2.3.8 Initialization Information

This section discusses the following:

- Hardware Reset
- Channel Script Execution
- Initialization and Script Execution Setup Sequence

7.2.3.8.1 Hardware Reset

After reset, the program RAM, context RAM, data RAM, and RAM containing the channel enable registers (CHNENBLn) have unpredictable contents.

The active register set is assigned to channel 0 and the PC is initialized to all zeros. However, since the channel enable register is all zeros, there are no active channels and the SDMA is halted waiting for the boot channel to start.

The Arm platform will have to setup the SDMA in order to boot it. The CONFIG register must be initialized to determine the DMA/core clock ratio (1 or 2). Channel Enable Registers must also be initialized.

To start up the SDMA, the Arm platform first creates some channel control blocks (CCB) and buffer descriptors (BD) in Arm platform memory for the boot channel (channel 0) and then initializes the channel 0 pointer register (SDMA_MC0PTR) to the address of the first control block. Data Structures for Boot Code and Channel Scripts provides an overview of the data structure for the CCB and BD's. The SDMA_HSTART, SDMA_HOSTOVR and SDMA_EVTOVR registers are then configured according to Runnable Channels Evaluation to allow channel 0 to run.

Upon being enabled, the SDMA begins executing the script located at the address indicated by the Channel 0 Boot Address register (SDMA_CHN0ADDR) in the program memory. The reset value of SDMA_CHN0ADDR points to the default bootload script in ROM. This ROM script will read the channel 0 pointer register (SDMA_MC0PTR) to determine the location of the Channel Control Block (SDMA_CCB) in Arm platform memory. The script will then begin fetching by DMA the first channel control block which contains a pointer to the location channel 0 Buffer Descriptor chain which is also fetched via DMA. If the buffer descriptor contains a valid command, the script interprets the command in each buffer descriptor and proceeds to implement the command and move on to the next buffer descriptor control block. The buffer descriptor commands for channel zero are typically set up to load SDMA's program RAM, Data RAM, and initial values for the channel contexts. Some channel scripts expect particular parameters to be passed

There are two ways to make the SDMA boot on a user-defined script. The OnCE (either via its JTAG interface or its Arm platform Control interface) can be used to download any code in the SDMA RAM and force the SDMA to boot on that code. Also, the SDMA_CHN0ADDR register in the Arm platform programming model can be modified to point to user code in RAM which would need to either have been loaded via the ONCE or default bootload routine (ex before a S/W reset).

7.2.3.8.2 Channel Script Execution

The execution of an SDMA script depends on both the instructions that make up the script, the data context upon which it operates, and commands or parameters allowed to the buffer. All these items must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate context, but may share scripts and locations in data RAM.

Channels are initialized by the Arm platform by using channel 0 to download any required scripts and data values and the channels initial context. The context contains all the initial values of the SDMA core registers. This includes the Program Counter (PC) which is set to the start of the desired script in SDMA program memory.

The Arm platform selects which trigger conditions that must occur for the channel to start by configuring the SDMA_CHNENBL, SDMA_HOSTOVR and SDMA_EVTOVR registers. The trigger events include Arm platform setting HE (SDMA_HSTART) or a hardware DMA request asserts an event input to SDMA. The channel can become active according to its priority compared with other runnable channels when the selected trigger(s) cause the condition described in Runnable Channels Evaluation to evaluate as true.

The specific parameters to be passed to each script in the buffer descriptor or context are documented in the software documentation for each script. Please refer to SDMA Scripts for complete script documentation. Buffer Descriptor Format provides an overview of the buffer descriptor format.

7.2.3.8.3 Initialization and Script Execution Setup Sequence

To summarize, the following steps are minimally required to setup SDMA and run channel scripts.

- Perform Hardware Reset. The program RAM, context RAM, data RAM and SDMA_CHNENBLn registers have unpredictable contents after this reset.
- Initialize SDMA_CHNENBLn registers to map DMA request events to desired channels.

- Configure SDMA_CHNPRIn registers to select priority for runnable channels. A non-zero priority is required for the channel to run.
- Configure the SDMA_CONFIG register to select DMA to SDMA core clock ratio .
- Set up channel control blocks and buffer descriptors in Arm platform to specify the loading of SDMA program RAM and channel contexts for each SDMA channel to be used. Reference Data Structures for Boot Code and Channel Scripts.
- Configure SDMA_MC0PTR register with base address of Arm platform Channel Control Block base address.
- Initialize SDMA_CHNENBLn registers to map DMA request events to associated channel. Reference Mapping DMA Requests to Pending Channels.
- Configure SDMA_CHNPRIn registers to set priority for each channel to be run.
- For each channel to be run, configure SDMA_HOSTOVR (HO) and SDMA_EVTOVR (EO) registers to select which events (hardware and/or software trigger events) must occur for the channel to be runnable. Reference Runnable Channels Evaluation.
- Set bit 0 of the SDMA_HSTART register to set HE[0] and allow Channel 0 to run (assumes EO[0] andDO[0] were both set in previous step). This will cause SDMA to load the program RAM and channel contexts configured previously.
- Wait for Channel 0 to finish running. This is indicated by HI[0]=1 in the SDMA_SDMA_INTR register, or by optional interrupt to the Arm platform.
- Set the LOCK bit in the SDMA_SDMA_LOCK register to prevent un-authorized uploads of data to SDMA RAM.
- Additional channel scripts can now be run by enabling the selected software or hardware trigger event according to Runnable Channels Evaluation.

7.2.3.9 SDMA Programming Model

This section describes the programming model for the SDMA RISC engine, including its processor, memory, and internal control registers.

All addresses are related to the internal SDMA memory map, which is completely different from the Arm platform memory maps. The Arm platform processor has no access to any hardware resource described, except when those resources are described in Arm Platform Memory Map and Control Register Summary. .

7.2.3.9.1 State and Registers Per Channel

The SDMA can be seen as a set of 32 identical devices that are able to perform one data transfer channel each. Only one channel can work at a time, but every channel state is available at any time.

This chapter lists the components of every channel state.

7.2.3.9.2 General Purpose Registers

Each channel has eight general purpose registers of 32 bits for use by scripts. General register 0 has a dedicated function for the loop instruction, but otherwise can be used for any purpose.

7.2.3.9.3 Functional Unit State

Each channel context has some state that is part of the functional units.

The specific allocation of this state is part of the functional unit definition that is described in Burst DMA Unit Programming, Peripheral DMA Unit Programming .

This state must be saved/restored on context switches.

7.2.3.9.3.1 Program Counter Register (PC)

The PC is 14 bits. Since instructions are 16 bits in width and all memory in the SDMA is 32 bits in width, the low order bit of the PC selects which half of the 32-bit word contains the current instruction.

A low order bit of zero selects the most significant half of the word.[1]

7.2.3.9.3.2 Flags

Each channel has the following four flags:

- The T bit reflects the status of some arithmetic and test instructions. It is set when the result of an addition or a subtraction is zero and cleared otherwise. It is also the copy of the tested bits. Finally, it can also be set when the loop counter (GReg0) reaches zero. When the last instruction of the hardware loop is an operation that can modify the T flag, its effect on T is discarded and replaced by the GReg0 status.
- Two additional bits, SF and DF, are used to indicate error conditions resulting from loading data sources and storing to destinations, respectively. Access errors set these bits, and successful transactions clear them. They can also be cleared by specific instructions (CLRF and loop). The source fault (SF) is updated by the loads LD and LDF; the destination fault (DF) is updated by the stores ST and STF.

1. For example, big-Endian.

- Access errors are caused by several conditions including writing to the ROM, writing to a read-only memory mapped register, accessing an unmapped address, or any transfer error received by a peripheral when it is accessed.

  The SF and DF flags have a major impact on the behavior of the hardware loop: If SF or DF is set when starting a hardware loop and it is not masked by the loop instruction, the loop body will not be executed. Inside the loop body, if a load or store sets the corresponding SF or DF flag, the loop exits immediately. Testing the status of the T flag at the end of the loop (as well as testing both SF and DF) tells if the loop exited abnormally as any anticipated exit prevents GReg0 from reaching the zero value and thus setting the T flag. This is also valid if the fault occurs at the last instruction of the last loop.

- The last flag is the loop mode flag, LM, which is composed of two bits. The most significant bit indicates when the processor is currently operating in loop mode. It is set by the loop instruction and is cleared after execution of the last instruction of the last loop. The least significant bit is set when the program counter points to the last instruction of a loop on the last path. It is used for a channel that is restored with this configuration to know that the next program counter is EPC. As with the dynamic context switch Greg0, which indicates when the program must get out of the loop, it can be restored only on the last instruction of the loop. This, however, is too late to fetch the next instruction after the loop.

7.2.3.9.3.3 Return Program Counter (RPC)

The RPC is 14 bits. It is set by the jump to the subroutine instructions and used by the return from the subroutine instructions.

Instructions are available to transfer its contents to and from a general register.

7.2.3.9.3.4 Loop Mode Start Program Counter (SPC)

The SPC is 14 bits. It is set by the loop instruction to the location immediately following it.

7.2.3.9.3.5 Loop Mode End Program Counter (EPC)

The EPC is 14 bits. It is set by the loop instruction to the location of the next instruction after the loop.

7.2.3.9.4 Context Switching-Programming

Each channel has a separate context consisting of the eight general purpose registers and additional registers representing the state of the functional units.

The active registers and functional units contain the context of the active channel. The context of inactive channels is stored in SDMA RAM, which is part of the SDMA address space.

In a function of the selected context switching mode (Context Switching), modified registers by the program can be saved in the channel RAM space while the program is going on. In every cycle, a write access to the RAM is possible.

On a done or yield(ge) instruction, SDMA goes into "real" context switching. In one of the dynamic modes, modified registers not previously saved, as well as the PC-Loop registers, are stored into the context area of the channel that will be closed. The new PC-Loop registers are loaded from the context area of the new channel. All other registers are restored while the program is executed, giving priority to registers used by the decoded instruction. Therefore, in the best case, only the PC and Loop registers should be saved and restored during this context-switching phase, which only requires five SDMA cycles.

In static mode, the context switch stores all registers in the old channel RAM space, and restores all registers from the new channel RAM space. It requires 26 SDMA cycles.

The address of the context memory for channel *i* is *CONTEXT_BASE* + 24**i* or *CONTEXT_BASE* + *32*i w*here CONTEXT_BASE equals 0x0800. The table below presents the layout of a channel context in memory:

Table 7-14. Layout of a Channel Context in Memory for SDMA

OFFSET	31	30	29-16	15	14	13-0
0	SF	-	RPC	T	-	PC
1	LM		EPC	DF	-	SPC
2	GR0					
3	GR1					
4	GR2					
5	GR3					
6	GR4					
7	GR5					
8	GR6					
9	GR7					
10	MDA (burst DMA)					
11	MSA (burst DMA)					
12	MS (burst DMA)					
13	MD (burst DMA)					

Table continues on the next page...

Table 7-14. Layout of a Channel Context in Memory for SDMA (continued)

14	PDA (peripheral DMA)
15	PSA (peripheral DMA)
16	PS (peripheral DMA)
17	PD (peripheral DMA)
18	
19	
20	Reserved[1]
21	Reserved[1]
22	Reserved[1]
23	Reserved[1]
24	Scratch RAM (optional)
25	Scratch RAM (optional)
26	Scratch RAM (optional)
27	Scratch RAM (optional)
28	Scratch RAM (optional)
29	Scratch RAM (optional)
30	Scratch RAM (optional)
31	Scratch RAM (optional)

7.2.3.9.5 Address Space

The SDMA has four internal buses which are listed here.

- The Instruction bus reads instructions from the memory. Its address map is described in Instruction Memory Map.
- The Data bus (DMBUS) accesses the same memories as those visible on the Instruction bus, some memory-mapped registers (scheduler status and OnCE registers), and up to 14 peripherals. Its address map is described in Data Memory Map.
- The Functional Units bus (FUBUS) accesses the , Burst DMA, Peripheral DMA . The addressing mechanism is further detailed in Functional Units Programming Model.
- The Context Switch bus reads/writes registers into context-switch RAM space. It is a 64-bit bus dedicated for accessing this RAM space for updating the context of the running channel. While the program is going on, this bus has the lowest priority compared to the Instruction and Data buses, except for restoring a register needed for the decoded instruction to be executed. On the save part of a context switch (when the PCU is in its slave state), this is the only one used. On the restore part, the Instruction bus has the priority to read the next instruction at the restored PC and

otherwise the Context Switch bus is used. It is not possible to control the actual data transfers that occur on this bus.

7.2.3.9.5.1 Instruction Memory Map

The instruction memory map is based on a 14-bit address bus and a 16-bit data (instruction) bus.

Instructions are fetched from either program ROM or program RAM. An SDMA script is able to change the contents of the program RAM, which is also visible from the data bus.

The first two instruction locations (at 0 and 1) are special. Location 0 is where the PC is set on reset. Location 1 is where the PC is set upon the execution of an illegal instruction. It is expected that both of these locations will contain a jmp to handle routines.

Table 7-15. SDMA Instruction Memory Space

Device	SDMA Address (Hex)	Base Address Label	Block Name	WS	Description
ROM	0x0000 ↓ 0x07FF	SDMA_IBUS_ROM_ADDR	-	0	4 Kbyte internal ROM with boot code and standard routines.
RAM	0x1000 ↓ 0x1FFF	SDMA_IBUS_RAM_ADDR	-	0	8 Kbyte internal RAM with channels context and user data/routines.

7.2.3.9.5.2 Data Memory Map

All of the data accessible to SDMA scripts make up the data memory space of the SDMA.

This address space has several components:

- ROM (also visible on the Instruction bus)
- RAM (also visible on the Instruction bus)
- Shared Peripherals Registers
- SDMA Internal Registers (scheduler, OnCE, and registers that are also accessible by the Arm platform)

SDMA scripts can read and write to the context RAM, data RAM, shared peripheral registers, and internal registers.

The address range is 16 bits and the data width is 32 bits. When accessing peripheral registers (USB and so on), the data width may be different. The exact address map for the peripherals depends on the project (as presented in each respective chapter).

Data access is performed with *ld* and *st* instructions that take the address from a general purpose register in the core (GReg*n*). The mapping between the general purpose register contents and the address bus is given in the following table:

Table 7-16. GReg*n* to DMBUS Address Mapping

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sz	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
address															

Grayed bits are simply discarded but they must be cleared to ensure forward-script compatibility.

- sz (bit 31) indicates the peripheral data width: 0 is used for a 32-bit peripheral and 1 is used for a 16-bit peripheral.
- address (bits 15 down to 0) is the address of the accessed resource (internal memory, internal register, or shared peripheral).

Table 7-17. SDMA Data Memory Space

Device	SDMA Address (Hex)	Size	Description
ROM	0x0000 → 0x03FF	4 Kbyte	4 Kbyte internal ROM with boot code and standard routines
Reserved	0x0400 → 0x07FF	4 Kbyte	4 Kbyte Reserved
RAM	0x0800 → 0x0FFF	8 Kbyte	8 Kbyte internal RAM with channels contexts and user data/routines
per1	0x1000 → 0x1FFF	16 Kbyte	*peripheral 1* memory space (4 Kbyte peripheral's address space)
per2	0x2000 → 0x2FFF	16 Kbyte	*peripheral 2* memory space (4 Kbyte peripheral's address space)
per3	0x3000 → 0x3FFF	16 Kbyte	*peripheral 3* memory space (4 Kbyte peripheral's address space)
per4	0x4000 → 0x4FFF	16 Kbyte	*peripheral 4* memory space (4 Kbyte peripheral's address space)
per5	0x5000 → 0x5FFF	16 Kbyte	*peripheral 5* memory space (4 Kbyte peripheral's address space)
per6	0x6000 → 0x6FFF	16 Kbyte	*peripheral 6* memory space (4 Kbyte peripheral's address space)
Registers	0x7000 → 0x7FFF	16 Kbyte	Memory mapped registers
per7	0x8000 → 0x8FFF	16 Kbyte	*peripheral 7* memory space (4 Kbyte peripheral's address space)
per8	0x9000 → 0x9FFF	16 Kbyte	*peripheral 8* memory space (4 Kbyte peripheral's address space)
per9	0xA000 → 0xAFFF	16 Kbyte	*peripheral 9* memory space (4 Kbyte peripheral's address space)
per10	0xB000 → 0xBFFF	16 Kbyte	*peripheral 10* memory space (4 Kbyte peripheral's address space)
per11	0xC000 → 0xCFFF	16 Kbyte	*peripheral 11* memory space (4 Kbyte peripheral's address space)
per12	0xD000 → 0xDFFF	16 Kbyte	*peripheral 12* memory space (4 Kbyte peripheral's address space)
per13	0xE000 → 0xEFFF	16 Kbyte	*peripheral 13* memory space (4 Kbyte peripheral's address space)
per14	0xF000 → 0xFFFF	16 Kbyte	*peripheral 14* memory space (4 Kbyte peripheral's address space)

7.2.3.10 SDMA Initialization

Appendix A describes the setup of the SDMA . This section provides a quick description of several initialization procedures.

NOTE

There may be differences with the actual implementation in the API.

7.2.3.10.1 Hardware Reset-SDMA

After reset, the RAM that holds contexts, data, scripts, and the DMA request-channels matrix has unpredictable content.

The core registers are all reset to 0, including the PC; the PCU state is *Sleep after Reset*. No channel can be activated because all of the priorities are also reset to 0.

7.2.3.10.2 Standard Boot Sequence

The following is the standard boot sequence:

1. Initialize the CONFIG register-detailed in Configuration Register (SDMAARM_CONFIG)-to determine the Arm platform DMA/core clock ratio (1 or 2)
2. Initialize the DMA request-channels matrix (seeChannel Enable RAM (SDMAARM_CHNENBL*n*)).
3. Program the channel control registers-Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel BP Override (SDMA_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND)-according to the channel allocation.
4. Perform any necessary setup as required by the standard boot script in ROM (this is described in Appendix A).
5. Trigger channel 0 with the Channel Start (SDMAARM_HSTART) register, which starts the execution of the ROM script starting at address 0. This boot downloads channel scripts and contexts in RAM.

7.2.3.10.3 User-Defined Boot Sequence

The following is a user-defined boot sequence:

1. Initialize the Configuration Register (SDMAARM_CONFIG)Channel Enable RAM (SDMAARM_CHNENBL*n*), Channel Event Override (SDMAARM_EVTOVR), Channel BP Override (SDMAARM_DSPOVR), Channel Arm platform Override (SDMAARM_HOSTOVR), and Channel Event Pending (SDMAARM_EVTPEND).

2. Use the OnCE (either via its JTAG interface or its Arm platform control registers) to download any code in the SDMA RAM. Accessing the Memory describes how to write data to the RAM via the OnCE.
3. Use the OnCE instructions to make the PC default value point to the new boot script start address, or rely on the ROM startup script, which first jumps to the address in Channel 0 Boot Address (SDMAARM_CHN0ADDR). (This register default address points to the standard boot script.)

7.2.3.10.4 Script Loading and Context Initialization

The execution of an SDMA script depends on both the instructions that make up the script and the data context upon which it operates. Both must be initialized before the script is allowed to execute.

Each of the 32 channels has a separate data context, but may share scripts and locations in the data RAM.

The Arm platform manages the space in program RAM and data RAM. It also manages the assignment of SDMA channels to the device drivers that need them. Channels are initialized by the Arm platform via the channel 0 boot script. The boot channel downloads any required scripts with their data and the channels' initial contexts. Every context contains all the initial values of the registers, including the PC. Then the Arm platform can enable any channel that becomes active and begins fetching and executing instructions from its script.

7.2.3.11 Instruction Description

The following sections introduce the instruction of the SDMA.

Instruction set details are available in Instruction Set.

7.2.3.11.1 Scheduling Instructions

The following are scheduling instructions:

- done-The instruction causes certain scheduling or interrupt bits to be set or cleared, which may cause a change in the schedule-ability of the running channel. Then the instruction causes the SDMA to evaluate the current scheduling priorities and to choose the highest priority ready channel. If this channel is not the current channel, a context switch will take place. If there are no runnable channels, the SDMA will enter the stopped mode. The done 5 has a special usage reserved for debug, as explained in Debug Instructions.

- yield-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater than the current channel priority.
- yieldge-These instructions are special cases of the done instruction. They do not modify the scheduling bits, but allow the highest pending channel (if it exists) to preempt the current channel if the pending channel priority is strictly greater or equal to the current channel priority.
- notify-The notify instruction affects the scheduling bits, but does not cause rescheduling.

7.2.3.11.2 Conditional Branch Instructions

The conditional branch instructions of an 8-bit displacement, which is sign-extended and added to the current PC (which points to the next instruction) if the condition is satisfied.

Otherwise, control passes to the next sequential instruction.

- BF-Branch if False. The branch is taken if the T bit in the processor status is zero (false).
- BT-Branch if True. The branch is taken if the T bit in the processor status is one (true).
- BSF-Branch if Source Fault. The branch is taken if the SF bit in the processor status is one.
- BDF-Branch if Destination Fault. The branch is taken if the DF bit in the processor status is one.

7.2.3.11.3 Unconditional Jump Instructions

There are two varieties of unconditional control transfers: an absolute transfer and a through-register transfer.

Absolute transfers have a 14-bit address field that replaces the current PC.

- JMP-Jump. Causes the processor to jump to an absolute address encoded in the instruction itself.
- JSR-Jump to Subroutine. Causes the processor to jump to a subroutine, the address of which is encoded in the instruction itself.

- JMPR-Jump through Register. Causes the processor to jump to an absolute address contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.
- JSRR-Jump to Subroutine through Register. Causes the processor to jump to a subroutine, the address of which is contained in a General register. This instruction is meant to be used when more than one level of subroutines are required.

7.2.3.11.4 Subroutine Return Instructions

The following are subroutine return instructions:

- RET-Return from Subroutine. The RET restores the contents of RPC to PC.
- LDRPC-Load from RPC to Register. THe LDRPC instruction is meant to be used when more than one level of subroutines are required. It stores the contents of RPC in any General register.

7.2.3.11.5 Loop Instruction

The following is a loop instruction:

LOOP-Enters Loop Mode. Before entering loop mode, the loop instruction can optionally clear the fault flags (SF and/or DF) based on a 2-bit field in the instruction. This feature is linked to the fact that setting SF or DF in loop mode will cause an immediate exit of the loop.

7.2.3.11.6 Miscellaneous Instructions

The following are miscellaneous instructions:

- CLRF-Clear Fault Flags. This instruction clears any combination of SF and DF.
- MOV r,s-This moves data from GReg[s] to GReg[r].
- LDI r,immediate-This loads GReg[r] with a zero-extended immediate value.

7.2.3.11.7 Logic Instructions

The following are logic instructions:

- XORr,s-This performs an exclusive or between GReg[r] and GReg[s], and stores the result in GReg[r].
- XORIr,immediate-This performs an exclusive or between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- ORr,s-This performs an or between GReg[r] and GReg[s], and stores the result in GReg[r].

- ORIr,immediate-This performs an or between GReg[r] and a zero-extended immediate value and, stores the result in GReg[r].
- ANDNr,s-This performs an and between GReg[r] and the negated GReg[s], and stores the result in GReg[r].
- ANDNIr,immediate-This performs an and between GReg[r] and the negated zero-extended immediate value, and stores the result in GReg[r].
- ANDr,s-This performs an and between GReg[r] and GReg[s], and stores the result in GReg[r].
- ANDIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].

7.2.3.11.8 Arithmetic Instructions

Arithmetic instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the result is zero, otherwise it is cleared.

- ADD r,s-This performs the addition of GReg[r] and GReg[s], and stores the result in GReg[r].
- ADDI r,immediate-This performs the addition of GReg[r] and a zero-extended immediate value, and stores the result in GReg[r].
- SUB r,s-This performs the subtraction of GReg[s] from GReg[r], and stores the result in GReg[r].
- SUBIr,immediate-This performs the subtraction of a zero-extended immediate value from GReg[r], and stores the result in GReg[r].

7.2.3.11.9 Compare Instructions

Compare instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if the comparison is true, otherwise it is cleared.

NOTE

Only one version of the immediate form is implemented. Non-equality comparisons to immediate values will require two instructions.

- CMPEQ r,s-This sets T when registers GReg[r] and GReg[s] are equal.
- CMPEQIr,immediate-This sets T when register GReg[r] and the zero-extended immediate value are equal.
- CMPLTr,s-This sets T when register GReg[r] is less than and not equal to GReg[s]. The comparison is signed.
- CMPHS r,s-This sets T when register GReg[r] is greater than or equal to GReg[s]. The comparison is signed.

7.2.3.11.10 Test Instructions

Test instructions modify the T bit in the processor status according to the result of the operation. The T bit is set if any bit in the result is one, otherwise it is cleared.

- TSTr,s-This performs an and between GReg[r] and GReg[s], and sets T if the result is not zero.
- TSTIr,immediate-This performs an and between GReg[r] and a zero-extended immediate value, and sets T if the result is not zero.

7.2.3.11.11 Byte Permutation Instructions

These instructions shuffle the bytes in a register. For the purpose of describing these instructions, have the bytes in a register be numbered from the most significant as b_3, b_2, b_1, b_0.

- RORBr-The rotate right byte. The result is b_0, b_3, b_2, b_1.
- REVBr-The reverse bytes in word. The result is b_0, b_1, b_2, b_3.
- REVBLOr-The reverse, two low-order bytes. The result is b_3, b_2, b_0, b_1.

7.2.3.11.12 Bit Shift Instructions

The following are bit shift instructions:

- ROR1r-The rotate right 1 bit. This instruction does a circular right shift of 1 bit.
- LSR1r-The logical shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by a 0.
- ASR1r-The arithmetic shift right 1 bit. This instruction shifts all bits to the right by 1. The high order bit is replaced by itself.
- LSL1r-The logical shift left 1 bit. This instruction shifts all bits to the left by 1. The low order bit is replaced by zero.

7.2.3.11.13 Bit Manipulation Instructions

- BCLRIr,n-The bit clear is immediate; clears bit number *i* in register *r*.
- BSETIr,n-The bit set is immediate; sets bit number *i* in register *r*.
- BTSTIr,n-The bit test is immediate; tests bit number *i* in register *r* (T becomes equal to the selected register bit).

7.2.3.11.14 SDMA Memory Access Instructions

All memory accesses are 32 bits.

Any memory location that is implemented with less than 32 bits (for example, peripheral registers) causes unimplemented bits to be read as 0s.

All memory accesses will cause either the SF or DF flags in the processor status to be set if they cause a fault.

What constitutes a fault, especially when accessing peripheral registers, is a property of the memory location.

- LDr,(b,d)-The load instruction creates an address by adding the displacement field (d) to the contents of the base register (b). The SDMA location at the resulting address is read and placed in the destination register (r).
- STr,(b,d)-The store instruction creates an address in the same manner as the load instruction. The register (r) is stored in the SDMA location at the resulting address.

7.2.3.11.15 Functional Unit Instructions

The functional unit instructions have an 8-bit field that is placed on the functional unit bus.

Some of these bits are used to select which functional unit should be involved in the transfer. The remaining bits are decoded by the selected functional unit so their specific use depends on the functional unit. See Functional Units Programming Model.

There are two functional unit instructions, as follows:

- LDFr,fub-The 8-bit field is placed on the functional unit bus and a read is issued to the selected functional unit. As a result of this instruction, the SF may be set in the processor status.
- STFr,fub-The 8-bit field is placed on the functional unit bus and a write is issued to the selected functional unit. As a result of this instruction, the DF may be set in the processor status.

7.2.3.11.16 Illegal Instructions

All instruction encodings that are illegal cause the following actions:

- The current PC (which points to one beyond the offending instruction) is put in the EPC register.
- The loop mode bit is cleared.
- The PC is set to the value stored in the Illegal Instruction Trap Address (SDMAARM_ILLINSTADDR) register (the default value is 0x0001).

ILLEGAL-Although any instruction other than those indicated in the SDMA specification will trigger the illegal instruction mechanism, the ILLEGAL instruction code is preferred as it will always be kept as *illegal* in the possible future versions of the SDMA core.

7.2.3.11.17 Debug Instructions

The following are debug instructions:

- SOFTBKPT-The software breakpoint instruction causes the core to stop and enter debug mode. The core can then be accessed and started by the OnCE debug block only.
- done 5-This instruction is used for debugging, as it copies the contents of the PCU registers and flags to the context memory. Information on this instruction is described in Saving the Context.
- CpShReg-This instruction copies the context memory into the PCU registers and flags. Modifying the corresponding memory location before executing this instruction enables you to have the channel continue from a new instruction address. This instruction is described in Restoring the Context.

7.2.3.12 Functional Units Programming Model

The functional unit instructions cause an 8-bit code, found in the low eight bits of the instruction, to be asserted on the functional unit control bus.

Some of these bits are used to select one of several functional units. Functional units which can be selected include SDMA registers such as MSA and MSD which are not mapped in the SDMA memory map, and are accessible only through the functional unit bus. These Functional Unit Registers are listed in the following table. In order to establish a programming convention, assume the selection bits are some number of the most significant bits of the 8-bit code. Furthermore, some number of the least significant bits is decoded by a given functional unit to establish the type of operation to perform.

Table 7-18. Functional Unit Registers

Functional Unit	Register	Register Name	Section/Page
Burst DMA Unit Programming	SDMSA	Memory Source Address Register	Memory Source Address Register (MSA)
	MDA	Memory Destination Address Register	Memory Destination Address Register (MDA)
	MD	Memory Data Buffer Register	Memory Data Buffer Register (MD)

Table continues on the next page...

Table 7-18. Functional Unit Registers (continued)

Functional Unit	Register	Register Name	Section/Page
			(Write) Burst DMA Write (stf) (Read) Burst DMA Read (ldf)
	MS	Memory State Register	State Register (MS)
Peripheral DMA Unit Programming	PSA	Peripheral Source Address Register	Peripheral Source Address Register (PSA)
	PDA	Peripheral Destination Address Register	Peripheral Destination Address Register (PDA)
	PD	Peripheral Data Buffer Register	Peripheral Data Register (PD) (Write) Peripheral DMA Write (stf)-Write Mode (Read) Peripheral DMA Read (ldf)-Read Mode
	PS	Peripheral State Register	Peripheral State Register (PS)

More information regarding the functional units can be found in Peripheral DMA Unit, and Burst DMA Unit.

7.2.3.12.1 Burst DMA Unit Programming

The DMA instructions control the DMA state machine and may cause a DMA cycle on the associated memory bus.

There are four registers associated with the burst DMA unit: a Memory Source Address register (MSA), a Memory Destination Address register (MDA), a Memory Data buffer (MD), and a state register (MS). The burst DMA has two different uses:

- A data transfer between External Memory Interface and SDMA general register
- A data transfer in copy mode where blocks of data are transferred from the source address to the destination address

7.2.3.12.1.1 Memory Source Address Register (MSA)

The source address register contains the pointer into EXTMC memory associated with the next read data transfer. It has byte granularity.

Reading the register with the ldf instruction has no side effects, and gives the address value in the EXTMC memory of the next data that is read by the SDMA during an ldf MD instruction.

Writing the source address register has two side effects: If the prefetch bit is set, a DMA read cycle (8-word read access) is issued with the new address. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MSA to guarantee all the data is effectively written to memory.

The MSA register has two modes of programming:

- Frozen-In frozen mode, the MSA register is not modified after DMA accesses.
- Incremented (default mode)-In incremental mode, MSA is incremented by the number of bytes transferred during read cycles.

7.2.3.12.1.2 Memory Destination Address Register (MDA)

The destination address register contains the pointer into EXTMC memory associated with the next write data transfer. It has byte granularity.

Reading the MDA register with the ldf instruction has no side effects. It gives the address value in the EXTMC memory where the next SDMA data (stf r,MD instruction) is stored when MD FIFO is flushed.

Writing the destination address register has one side effect. Any data still located in the buffer is lost. If there is valid write data in the buffer, it is necessary to force the DMA to completely flush it out before modifying MDA to guarantee all the data is effectively written to memory.

The MDA register has two modes of programming:

- Frozen-In frozen mode, the MDA register is not modified after DMA accesses.
- Incremented (default mode)-The MDA register is incremented by the number of bytes transferred during write cycles.

7.2.3.12.1.3 Memory Data Buffer Register (MD)

The data buffer register consists of a bank of 36 bytes that behave like FIFO.

This FIFO stores the eight words received when a read burst is triggered by the DMA (DMA is in read mode).

The MD register is in write mode after a writing in MDA or after an stf MD instruction.

In that case, a burst write access is automatically triggered when there are more than eight words in MD. For bandwidth optimization, any transfers between DMA and the EXTMC controller are based on burst accesses.

An ldf r,MD|SIZE instruction that reads the data buffer may cause a DMA cycle, as follows:

- If there are less bytes in the FIFO than the size parameter of the instruction. For instance, if only two bytes are available in MD and a 4-byte read is requested, a burst read access is executed to complete the two bytes.
- If the prefetch bit is set, and after reading there is enough space in the FIFO to store a full burst, a burst read access is triggered.

An stf r,MD|SIZE instruction that writes to the data buffer may cause a DMA cycle if the number of written bytes in MD is higher than 32 (eight words) or if the flush bit is set.

When DMA is used for data transfer between SDMA and EXTMC (reading or writing), no immediate error is possible because the block manages a data misalignment issue; therefore, it is allowed to read/write a word to/from a half-word address. However, the addresses (source or destination) must belong to the EXTMC memory mapping. The only potential error, in this mode, would be the error sent back by the EXTMC controller when an access to a super-user page is detected. The whole transfer on the DMA associated bus will be considered successful when there are no errors seen on the bus during the transfer. In copy mode, an immediate error could be returned to SDMA as described in Burst DMA Unit Error Management.

7.2.3.12.1.4 State Register (MS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer. MS is also accessed to set-up the conditional yielding feature.

The initialization value of this register is 0 and it consists of the following:

Table 7-19. SDMA_MS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	0	0	spriv	stype	0	0	dpriv	dtype
W																
R	0	0	0	0	y	d	e		0	0	n					
W																

Table 7-20. SDMA_MS Field Descriptions

Field	Description
31-22	Reserved

Table continues on the next page...

Table 7-20. SDMA_MS Field Descriptions (continued)

Field	Description
21 spriv	The spriv value is ignored for this device. 0 = valid value 1 = Reserved
20 stype	Source Mode. Indicates if MSA has to be incremented (or not) during accesses. 0 Frozen-MSA is not modified. 1 Incremented-MSA is incremented by the number of transferred bytes during read access.
19-18	Reserved
17 dpriv	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved
16 dtype	Destination Mode. Indicates if MDA has to be incremented (or not) during accesses. 0 Frozen-MDA is not modified. 1 Incremented-MDA is incremented by the number of transferred bytes during write access.
15-12	Reserved
11 y	Conditional Yielding selector. When selected, theyield/yieldge instructions will not switch channels if the Burst DMA is in Write Mode, and it has less than four bytes in its FIFO. This is aimed at reducing the number of inefficient FIFO flushes due to context switches. 0 Always yields 1 Yields conditionally (when there are less than four bytes in the FIFO in write mode)
10 d	Access Direction or DMA Mode. DMA is in write mode when data was written into MD by stf MD instructions, or if a previous DMA cycle on the external bus was a write access. Writing MDA or MSA changes the DMA mode to the respective value. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by SDMA with an ldf MD instruction. Reading MDA or MSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 error read burst
7-6	Reserved
5-0 n	Number of bytes in the MD FIFO.

7.2.3.12.1.5 Burst DMA Write (stf)

When received from a stf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 7-21. STF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s		p	freeze	r			spriv
MDA								dpriv
MD			f	cpy			sz	
MS								

Table 7-22. STF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MSA)	Prefetch Flag 0 No prefetch 1 Prefetch required from new MSA
5 f (MD)	Forced Flush Flag 0 Automatic flush 1 FIFO contents are flushed (including the new written data).
4 freeze (MSA/MDA)	Address Freeze Mode 0 Address is normally incremented. 1 Address is frozen.
4 cpy (MD)	Copy Mode selection 0 Write Mode 1 Copy Mode
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD/MS)	Transfer Size 00 size 0 (no data stored in the FIFO) 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
0 spriv (MSA)	The spriv value is ignored for this device. 0 = valid value

Table continues on the next page...

Table 7-22. STF Code Bit Field Descriptions (continued)

Field	Description
	1 = Reserved
0 dpriv (MDA)	The dpriv value is ignored for this device. 0 = valid value 1 = Reserved

The possible write instructions are listed in the table below (unused bits should always be cleared).

Table 7-23. Burst DMA STF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	stf r,MSA	Writes content of the SDMA general register (r) to the source address register. MSA is in incremented mode.
00_0_1_00_00	stf r,MSA|FR	Writes content of the SDMA general register (r) to the source address register. MSA is in frozen mode.
00_1_0_00_00	stf r,MSA|PF	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access. MSA is in incremented mode.
00_1_1_00_00	stf r,MSA|PF|FR	Writes content of the SDMA general register (r) to the source address register, and starts a read burst access.
00_0_0_01_00	stf r,MDA	Writes content of the SDMA general register (r) to the destination address register. MDA is in incremented mode.
00_0_1_01_00	stf r,MDA|FR	Writes content of the SDMA general register (r) to the destination address register. MDA is in frozen mode.
00_1_0_10_00	stf r,MD|SZ0|FL	No data transfers between the SDMA and MD, but all valid written data of the MD is flushed to the memory. An acknowledge or error is sent back to the SDMA core on transfer completion.
00_0_0_10_01	stf r,MD|SZ8	8-bit (byte) transfer to write buffer MD
00_1_0_10_01	stf r,MD|SZ8|FL	8-bit (byte) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_10	stf r,MD|SZ16	16-bit (half-word) transfer to write buffer MD
00_1_0_10_10	stf r,MD|SZ16|FL	16-bit (half-word) transfer to write buffer MD and flush after transfer. All valid written data of the MD is flushed to memory.
00_0_0_10_11	stf r,MD|SZ32	32-bit (word) transfer to write buffer MD
00_1_0_10_11	stf r,MD|SZ32|FL	32-bit (word) transfer to write buffer MD and flush after transfer. All valid written data of MD is flushed to memory.
00_0_1_10_00	stf r,MD|CPY	No data transfer between SDMA and MD but starts a copy transfer whose length is given by the 4 LSB of *r* register. (Maximum burst length is eight words.)
00_0_0_11_11	stf r,MS	32-bit (word) transfer to status register MS
00_0_0_11_00	stf r,MS|SZ0	Clears the error flag (if set). Other MS bits are unchanged; this instruction is also known as clref MS.

NOTE

When a flush bit is set, the SDMA flushes the FIFO including the newly written data. An acknowledge is sent to the core before the flush completes (except if size 0 is used). The goal of this flush bit is to force a flush, but it is recommended to use it only when needed (for example, when finishing a row of pixels during 2D data transfers). Indeed, if this bit is omitted and if there are more than 32 bytes in the FIFO, a burst write access is automatically triggered.

Since all the stf r,MD instructions (including the copy mode) acknowledge the SDMA core before the store is effective (except if size 0 is used), it is recommended to perform an ldf from MS before terminating a channel in order to check the final error status. (The ldf from MS will stall the core until all the data was flushed out and the transfer status is known.)

After every stf MD instruction, the MDA is incremented by the number of bytes that are written in MD, except when it is programmed in frozen mode.

7.2.3.12.1.6 Burst DMA Read (ldf)

When received from an ldf instruction, the function code bits are interpreted as follows, depending on the addressed register:

Table 7-24. LDF Code Bits

Register	7	6	5	4	3	2	1	0
MSA	s				r			
MDA								
MD			p				sz	
MS								

Table 7-25. LDF Code Bit Field Descriptions

Field	Description
7-6 s	Functional Unit selector 00 for Burst DMA
5 p (MD)	Prefetch Flag 0 no prefetch 1 automatic prefetch

Table continues on the next page...

Table 7-25. LDF Code Bit Field Descriptions (continued)

Field	Description
3-2 r	Register selection 00 MSA 01 MDA 10 MD 11 MS
1-0 sz (MD)	Transfer Size 00 reserved 01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)

The table below lists the possible write instructions (unused bits should always be cleared).

Table 7-26. Burst DMA LDF Instruction List

Binary	Assembly	Comments
00_0_0_00_00	ldf r,MSA	Copies the source address register value into an SDMA general register. It gives the memory address of the next data that will be read with an ldf MD instruction.
00_0_0_01_00	ldf r,MDA	Copies the destination address register value into an SDMA general register. It gives the memory address where the next incoming data will be flushed.
00_0_0_10_01	ldf r,MD|SZ8	8-bit (byte) read
00_1_0_10_01	ldf r,MD|SZ8|PF	8-bit (byte) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_10	ldf r,MD|SZ16	16-bit (half-word) read
00_1_0_10_10	ldf r,MD|SZ16|PF	16-bit (half-word) read. If after this reading, and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_10_11	ldf r,MD|SZ32	32-bit (word) read
00_1_0_10_11	ldf r,MD|SZ32|PF	32-bit (word) read. If after this reading and the MD FIFO is empty, a burst read access at the MSA address is triggered.
00_0_0_11_00	ldf r,MS	Copy the status register value into an SDMA general register.

NOTE

Read data is 0-extended before writing in the SDMA general registers. When reading the MD register, the DMA takes data from the FIFO if it is available. If part or whole data is not in the FIFO, an external burst read access is performed to provide the missing data. The SDMA is stalled as long as the required read data is not complete.

After every reading, MSA is incremented by the number of read bytes from MD FIFO, except when MSA is programmed in frozen mode.

7.2.3.12.1.7 Prefetch/Flush and Auto-Flush Management-Burst DMA Unit

The prefetch and auto-flush management enables the SDMA RISC machine to go on while a DMA access is performed.

When the RISC core requires a prefetch (p = 1) to the Burst DMA, it will receive an immediate transfer acknowledge before the DMA has finished the external access. This enables the RISC core to do other things like accessing another DMA machine.

The basic principle in prefetch mode is for the DMA to anticipate data reads from the SDMA RISC engine by fetching external bursts of data as soon as there is enough space in the DMA FIFO to store it. If ever the RISC engine required data that is not available in the FIFO, the read acknowledge is delayed until the data is available, but it does not have to wait until the burst completes.

The auto-flush basic principle is similar: An automatic flush is triggered every time there are eight words to be written in the FIFO. If the FIFO is full and the RISC engine requires another write, it is stalled until the burst has started and enough space was freed in the FIFO to store that new data. This means the SDMA RISC engine does not have to wait for the completion of a burst to receive its acknowledge and continue its processing.

In particular, an auto-flush is executed when DMA is in write mode and if the following is true:

- If the FIFO is empty and the first write is to a word-aligned address of any size (ex: the 2 LSB of MDA[1:0]= 0x0), the auto-flush is triggered immediately after the write of the 32'nd byte.
- If the FIFO is empty, and if MDA is an odd byte address (1, 3, 5, 7,...) and an stf MD|SZ8 is executed, the byte is flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is a half-word address (2, 6, 0xA,...) and an stf MD|SZ16 is executed, the two bytes of the incoming data are flushed to memory. Once MDA increments to a word aligned address, the auto-flush will be triggered every 32 bytes.
- If the FIFO is empty, and if MDA is not a word-aligned address (ex 1, 2, 3, 5, 6, 7, 9,...), and an stf MD|SZ32 is executed, the first 1 to 3 bytes will be flushed up to the next word aligned address. Afterwards, an auto-flush will be triggered each time the FIFO receives 32-bytes.

- Therefore, if an stf MD|SZ32 is executed with MDA equal to 0x1 and with an empty MD FIFO, the bytes located at addresses 1, 2, and 3 are flushed, and the byte located at address 4 remains in MD FIFO. This solves the misalignment issue. Additionally, the next write instructions (stf) complete the FIFO until it contains eight words; then a burst write is executed by the DMA to empty the FIFO. Protocol on the external bus does not support bursts of different data types (byte, half-word, or word).

  For example, consider the case where data is written using a byte access, stf MD| SZ8. The value of MDA during the very first byte write determines when the auto-flush will occur as follows:

- If MDA=0x0, the flush occurs following the write of byte 32
- If MDA=0x1, the flush occurs following the write of byte 1, byte 3 and byte 35.
- If MDA=0x2, the flush occurs following the write of byte 2 and byte 34.
- If MDA=0x3, the flush occurs following the write of byte 1 and byte 33.
- If MDA=0x4, the flush occurs following the write of byte 32

The flush command forces the DMA to flush all MD valid bytes to the EXTMC controller. An acknowledge is sent immediately to the SDMA, and any potential error is reported on a future access. It is thus essential to conclude a transfer with a last read from MS, which will stall the core until all data was flushed out and returned to the transfer status (acknowledge or error).

NOTE

During this kind of auto-flush (which occurs only at the beginning of a misaligned write transfer) no acknowledge is sent back to the SDMA, which is stalled until a flush is completed.

7.2.3.12.1.8 Data Alignment and Endianness-Burst DMA Unit

7.2.3.12.1.8.1 Burst DMA in Read Mode

For every read access to MD, the data returned to the SDMA core and the new FIFO state depends on the MSA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is fetched externally on a 32-bit bus, but the valid bytes only are stored in the FIFO and left-aligned (for a transfer of consecutive words, it is only the first word that may be truncated). The following table shows the FIFO byte alignment strategy and the corresponding MSA, the returned data, and the new FIFO state for any access size of an internal read from MD.

Table 7-27. FIFO Read Configuration

Before read		Internal read access size	Read data	After read	
MSA[1:0]	FIFO state			MSA[1:0]	FIFO state
00	x0 x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 and so on...	sz8	00 00 00 x0	01	x1 x2 x3 y0 y1 y2 y3 z0
		sz16	00 00 x0 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz32	x0 x1 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
01	x1 x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 and so on...	sz8	00 00 00 x1	10	x2 x3 y0 y1 y2 y3 z0 z1
		sz16	00 00 x1 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz32	x1 x2 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
10	x2 x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 and so on...	sz8	00 00 00 x2	11	x3 y0 y1 y2 y3 z0 z1 z2
		sz16	00 00 x2 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz32	x2 x3 y0 y1	10	y2 y3 z0 z1 z2 z3 t0 t1
11	x3 y0 y1 y2 y3 z0 z1 z2 z3 t0 t1 t2 and so on...	sz8	00 00 00 x3	00	y0 y1 y2 y3 z0 z1 z2 z3
		sz16	00 00 x3 y0	01	y1 y2 y3 z0 z1 z2 z3 t0
		sz32	x3 y0 y1 y2	11	y3 z0 z1 z2 z3 t0 t1 t2

7.2.3.12.1.8.2 Burst DMA in Write Mode

For every write access to the MD, the new FIFO state depends on the MDA status and the access size.

The FIFO is considered as a stack of 36 bytes: Data is stored in the FIFO according to the internal access size and the former MDA value. The following table shows the FIFO byte alignment strategy corresponding to MDA, as well as the new FIFO state for any access size of an internal write to MD.

Table 7-28. FIFO Write Configuration

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
00	tt uu vv ww ?? ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	01	tt uu vv ww x0 ?? ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	10	tt uu vv ww x0 x1 ?? ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	00	tt uu vv ww x0 x1 x2 x3 ?? ?? ?? ??
01	tt uu vv ww xx ?? ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	10	tt uu vv ww xx x0 ?? ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	11	tt uu vv ww xx x0 x1 ?? ?? ?? ?? ??
		sz32	x0 x1 x2 x3	01	tt uu vv ww xx x0 x1 x2 x3 ?? ?? ??
10	tt uu vv ww xx yy ?? ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	11	tt uu vv ww xx yy x0 ?? ?? ?? ?? ??
		sz16	?? ?? x0 x1	00	tt uu vv ww xx yy x0 x1 ?? ?? ?? ??
		sz32	x0 x1 x2 x3	10	tt uu vv ww xx yy x0 x1 x2 x3 ?? ??
11	tt uu vv ww xx yy zz ?? ?? ?? ?? ?? and so on...	sz8	?? ?? ?? x0	00	tt uu vv ww xx yy zz x0 ?? ?? ?? ??
		sz16	?? ?? x0 x1	01	tt uu vv ww xx yy zz x0

Table continues on the next page...

Table 7-28. FIFO Write Configuration (continued)

Before write		Internal write access size	Written data	After write	
MDA[1:0]	FIFO state			MDA[1:0]	FIFO state
					x1 ?? ?? ??
		sz32	x0 x1 x2 x3	11	tt uu vv ww xx yy zz x0 x1 x2 x3 ??

NOTE

If the FIFO mode changes from a write to a read mode, all remaining written bytes in MD are lost but no error is returned. Typically, this happens if an ldf MD is executed after stf MD instructions. Before a mode change, it is recommended to force the flush of a potential remaining byte by a stfMD|SZ0|FL instruction. In the same way, if a FIFO mode changes from a read to a write mode, all prefetched data present in the FIFO is lost and no error is returned.

7.2.3.12.1.8.3 Endianness-Burst DMA Unit

Big and Little Endian are supported by the Burst DMA, but data is always stored in MD in Big Endian.

Byte manipulation is performed when data is exchanged with an Burst controller (for example, during read or write burst accesses).

7.2.3.12.1.9 Burst DMA Unit Copy Mode

A mechanism is available to perform fast Arm-to-Arm transfers.

Data does not flow through the SDMA core: It is kept in the DMA FIFO. This mechanism is selected when writing MD with a special option in the instruction code (copy flag).

It is possible to transfer up to eight words in one SDMA instruction (this does not mean in one cycle). In this mode, every time an stf MD|CPY is executed, a read burst is executed and directly followed by a write burst transfer. Burst transfers are limited to eight words. The size of the transfer (in words)-given by the SDMA general register (4 LSB)-is also limited to eight. The following SDMA code shows how 100 bytes could be copied from the MSA address to the MDA address. This is sample code only.

Burst DMA copy mode example

```
          ldi r0,@src
          stf r0,MSA                          // Source address setup
          ldi r1,@dst
          stf r1,MSA                          // Destination address setup
          ldi r0,0x64                         // data transfer counter
          ldi r1,0x8
MAIN_XFER:
          cmphs r0,r1                         // Is r0 >= 0x8
          bf LAST_XFER                      // If not, jump to last transfer label
          stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.
          subi r0,0x8                         // Decrement counter
          jmp MAIN_XFER                     // return to main transfer loop
LAST_XFER:
          stf r0,MD|CPY
```

The main transfer loop is executed 12 times; then r0 equals 4 and the last transfer loop is run.

In this mode, an acknowledge is transmitted to the core as soon as the read burst can start; thus, a first copy instruction returns an immediate acknowledge and subsequent copy instructions will be acknowledged as soon as the previous copy has finished.

7.2.3.12.1.10 Burst DMA Unit Error Management

Another point to consider is the management of errors.

Because the DMA immediately sends an acknowledge to the RISC core (except for the stf MS|SZ0|FLS instruction), it assumes no error will occur. If an error occurs, it is flagged (transfer error acknowledge) for the following DMA access.

This should not be a problem if the DMA is used properly. The MD accesses are meant to stall the SDMA as little as possible to optimize throughput and hide calculation time. Therefore, final access to MS should be performed before closing a channel. This access waits until any pending operation is finished in the burst DMA and gather any remaining error.

In copy mode, an error could be immediately returned to the SDMA on execution of the ldf copy or stf copy instruction. It happens when MSA or MDA are not word addresses (for example, 0[4]). This is because copy mode must only be used for transferring a large packet of aligned data.

When an error is received during a *read* transfer to the external bus, which may occur during the burst accesses, the MD FIFO contains the valid beats of the burst, and the error flag of MS is set to 2'b11 (error read burst). It is possible to read MS ("n" field) to know how much valid data remains in MD and when MD is empty (after ldf instructions). The next read MD instruction sets the MS error flag to 2'b10 (error mode), and an error is sent back to the SDMA core. In error mode, it is possible to read MSA, which gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, gives rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

In "error read burst" mode, writing MDA, MSA, or MD, or starting a copy transfer by a stf MD|COPY instruction will cancel the error mode. The following table shows when an immediate error is sent back according to the executed instruction.

Table 7-29. Possibilities in ERROR READ BURST Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA (IU IPF) stf rn, MDA stf rn,MD|COPY	NO	Error mode is reset. MSA, MDA, or MD are updated and a DMA cycle may start. For the stf MD|COPY, a copy loop is executed.
stf rn, MS	NO	MS is updated.
ldf rn, MS ldf rn, MSA ldf rn, MDA	NO	MS, MSA, and MDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	YES/NO	Immediate error if there is no more data available for read in the FIFO.

When an error is received during a *write* transfer, the error is reported to the next DMA access. In this case, an error is sent to the SDMA core and the DMA goes to its error mode. Reading MS gives the number of bytes that remain in MD; reading MDA gives the address of the error data. Any attempt to read or write MD, or to modify MDA or MSA in error mode, give rise to an error; therefore, an error flag must be reset by clearing MS at the end of the SDMA code section responsible for error management.

Table 7-30. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, MD stf rn, MSA stf rn, MDA	Yes	Any attempt to modify MD, MSA, MDA will raise an immediate error and burst DMA remains in error mode. When address registers are write-accessed, an error is returned.
stf rn, MS	No	This is the only way to exit error mode. MS[9:8] must be reset by an stf MS|SZ0 instruction.
ldf rn, MS ldf rn, MSA ldf rn, MDA	No	MS, MSA, and MDA could be read in error mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, MD	Yes	Whatever the DMA direction (read or write), an ldf rn triggers an immediate error.

7.2.3.12.1.11 Conditional Yielding-Burst DMA Unit

The standard SDMA transfer is based upon a hardware loop that has the following structure:

Hardware Loop

```
loop
load Rn,source            // can be ldf or ld
<computation>            // can be done through functional units
store Rn,dest              // can be st or stf
done 0               // yield
```

This structure needs to be kept independent of the functional units' particularities regarding the context switch. However, there can be variations in the context switch's efficiency, which can depend on the number of data received up to that point, and on the data itself.

The DMA, with its 8-word burst capability, has a preferable context switch period when its address register is 8-word aligned: It is the only moment that occurs once every eight loops when the succession of bursts is not broken by the context switch. When this is not the case, a context switch requires the storing (or loading) of less than eight words, which requires separate accesses and is far less efficient. The rest of the 8-word packet is stored (or loaded) after the context restore, and this is done as separate accesses.

The proposed solution is a conditional yielding, which occurs only when the DMA is in an optimum state. It does not require any modification to the scripts. The condition is decided at the DMA level.

The DMA can be programmed in two modes-conditional or always-true-for every channel, which provides complete flexibility. By default, the DMA is not in conditional mode.

The DMA condition is computed from the FIFO fill level and the various modes, as follows:

- When copy mode is selected, regardless of the transfer direction ('read' or 'write'), the condition is always true.
- In read mode, the condition is always true.
- In write mode, the condition is true when there are four bytes or less in the FIFO; it is false when there are more than four bytes. The 4-byte limit comes from the possibility of saving those bytes as MD with absolutely no impact on the bus accesses.

The aim at conditional yielding is to avoid splitting bus accesses (especially bursts).

7.2.3.12.2 Peripheral DMA Unit Programming

The peripheral DMA unit is connected to the Multi-Layer DMA Crossbar Switch of the Arm platform.

Its goal is to perform data transfers between any blocks connected to the DMA bus of this platform. These blocks are either peripherals or memories. The peripheral DMA could be seen as the Arm platform DMA controller.

The DMA performs data transfers in three modes:

- Read mode, where data is read from peripherals or from memory connected to the Arm platform and copied in a SDMA general register.
- Write mode, where data of a general register has to be written in a peripheral or a memory.
- Copy mode, where data is read from a peripheral (or memory) at a source address (PSA) and automatically written to a peripheral (or memory) at a destination address (PDA).

In copy mode, no SDMA general register is involved as transferred data only goes through the data register of the DMA.

The peripheral DMA has three addressing modes: frozen, incremented, and decremented, as follows:

- Frozen mode-When source or destination addresses are frozen, their value is not modified after a transfer. This mode is typically used for addressing peripheral FIFOs located at a fixed address.
- Incremented mode-When source or destination addresses are in incremented mode, after every transfer they are incremented by the number of bytes transferred.
- Decremented mode-In decremented mode, addresses are decremented by the number of bytes transferred.

The peripheral DMA registers are as follows:

- Two, 32-bit address registers (PSA and PDA) that respectively contain the source address for a read access and the destination address for a write access
- A 32-bit status register (PS) that contains information on the peripheral DMA configuration, such as the number of valid bytes in the data register, the error flag, the source and destination address mode, and so on.
- A 32-bit data register (PD) that stores data involved in a data transfer

7.2.3.12.2.1 Peripheral Source Address Register (PSA)

The source address register contains a pointer to a source peripheral or a memory associated with the next read data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PSA) to store the address value
- A 2-bit register (stype) to store the source address mode (frozen, incremented, or decremented)
- A 2-bit register (ssize) to store the source target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects and gives the address value of the next data that will be read by the SDMA during an ldf MD instruction. Writing the source address register may have side effects. If there is valid write data in the data register and the source address is changed, the write data is discarded. If the prefetch bit is set, a DMA read cycle is issued with the new address.

When PSA is to be written, you must specify the source target address mode, providing its size (byte, half-word, or word). This enables omission of the size field in all ldf MD instructions. When DMA performs a read cycle, its size is given by the value of the PSA source size register (ssize). If source is a memory in incremented mode, first programmed in word mode (stf PSA|SZ32|I), and if an SDMA script needs to read bytes from this memory, the size of the source target must be updated before executing new accesses. The source address mode and its size are given by labels added to the stf PSA instruction as described in the write section. The ssize and stype registers are part of the DMA status register (PS).

Writing to PSA may issue an immediate error if the source size is not compatible with the value to be written into the PSA register. For instance, writing a 2 in PSA and specifying that it is memory-accessed in word mode creates an immediate error.

7.2.3.12.2.2 Peripheral Destination Address Register (PDA)

The destination address register contains a pointer to a source peripheral or a memory associated with the next write data transfer. It has byte granularity.

It is based on the following:

- A 32-bit register (PDA) to store the address value
- A 2-bit register (dtype) to store the destination address mode (frozen, incremented, or decremented)
- A 2-bit register (dsize) to store the destination target data path size (byte, half-word, or word)

Reading the register with the ldf instruction has no side effects, and gives the address value of the next data that will be written by SDMA during an stfMD instruction. Writing the destination register has no side effect. Similar to the PSA register, the destination address mode and source are specified in the stf PDA instruction and may also generate an error in case of incorrect programming.

7.2.3.12.2.3 Peripheral Data Register (PD)

The data register of the peripheral DMA is a 32-bit register. When the destination address is correctly set up, any writing to PD will automatically flush the new input data.

The number of SDMA bytes that will be transferred is given by the PDA size register. Unlike other SDMA DMAs, PD is not a FIFO: It is not used to accumulate bytes that from the SDMA and must be packed before being sent to external memories. In read mode, and if the source address is correctly set up, an ldf instruction will empty PD. If a prefetch is required along with the instruction, the DMA will initiate a new read transfer.

Reading PD in prefetch mode only stalls the SDMA when the prefetched data is not yet available. Writing PD only stalls the SDMA if the previous write operation was not completed. As soon as the previous operation is over, the acknowledge is sent back to the SDMA RISC engine.

An error flag-part of PS-is set when an external access fails. The error is thus reported to the next SDMA instruction that involves the peripheral DMA.

7.2.3.12.2.4 Peripheral State Register (PS)

The state register contains the DMA state-machine value. It can be accessed in case of an error received during a transfer.

Although all PS fields can be written by an stf instruction, it is recommended to access only the error bit (to reset it). Modifying other PS fields will provide an un-guaranteed DMA behavior.

The initialization value of PS is 0, and it consists of the following structure:

Table 7-31. PS Structure

	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	0	0	0	0	0	0	0	ssize		stype		dsize		dtype	
W																
R	0	0	0	0	0	d	e		0	0	0	0	0	n		
W																

Table 7-32. PS Field Descriptions

Field	Description
31-24	Reserved
23-22 ssize	Source Target Size. Determines the size of the read transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
21-20 stype	Source address Mode. Determines whether PSA is incremented, decremented, or kept unmodified after every read from the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
19-18 dsize	Destination Target Size. Determines the size of the write transfers on the external bus. It should match the accessed device characteristics. 00 *reserved* 01 Byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
17-16 dtype	Destination address Mode. Determines whether PDA is incremented, decremented, or kept unmodified after every write on the external bus. 00 Frozen Mode 01 Incremented Mode 10 Decremented Mode 11 *reserved*
15-11	Reserved
10 d	Direction Flag or DMA Mode. DMA is in write mode when data was written into PD by stf PD instructions, or if a previous DMA cycle on the external bus was a write access. Writing PDA or PSA does not change the DMA mode. DMA is in read mode when a previous DMA cycle was a read access, and DMA stays in read mode when data is read by the SDMA with an ldf PD instruction. Reading PDA or PSA does not change the DMA mode. 0 Read Mode 1 Write Mode
9-8 e	Error. Indicates if the previous access was acknowledged with a bus error. 00 No error was received. 01 *reserved* 10 Error mode 11 Error read
7-3	Reserved

Table continues on the next page...

Table 7-32. PS Field Descriptions (continued)

Field	Description
2-0 n	number of bytes in PD

NOTE

dtype, dsize, stype, and ssize are updated when PSA and PDA are written.

7.2.3.12.2.5 Peripheral DMA Write (stf)-Write Mode

When written by an stf instruction, the function code bits are interpreted as follows:

Table 7-33. STF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s		p	ar	am		sz	
PDA								
PD			pdsel					
PS			pssel					

Table 7-34. STF Code Bits Field Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PSA)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
3-2 am (PSA/PDA)	Address Mode. Determines how PSA or PDA is modified after every read or write access to the PD. 00 Frozen-Address registers are not modified after the transfer. 01 Incremented-Address registers are incremented by the number of transferred bytes. 10 Decremented-Address registers are decremented by the number of transferred bytes. 11 Updated-PSA and PDA are not modified. Either address mode is not modified, but the width of the data path is updated by the sz field.
1-0 sz	Transfer Size 00 *reserved*

Table continues on the next page...

Table 7-34. STF Code Bits Field Descriptions (continued)

Field	Description
	01 byte (8 bits) 10 half-word (16 bits) 11 word (32 bits)
5-0 pdsel	PD access selector 001000 is the only valid option
5-0 pssel	PS access selector 111111 writes to PS 001100 only clears the error flag in PS

Due to the large number of possible stf instructions, the following table provides only a short list of all the possible write instructions:

Table 7-35. Peripheral DMA STF Instruction List

Binary	Assembly	Comments
11_00_00_01 11_00_00_10 11_00_00_11	stf Rn, PSA|SZ8 |F stf Rn, PSA|SZ16|F stf Rn, PSA|SZ32|F	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is frozen.
11_10_00_01 11_10_00_10 11_10_00_11	stf Rn,PSA|SZ8 |F|PF stf Rn,PSA |SZ16|F|PF stf Rn,PSA |SZ32|F|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • 1, 2, or 4 bytes are *fetched* from the peripheral source. • Source address is frozen.
11_00_01_01 11_00_01_10 11_00_01_11	stf Rn, PSA|SZ8 |I stf Rn, PSA|SZ16|I stf Rn, PSA| SZ32|I	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1,2 or 4 after read PD.
11_10_01_01 11_10_01_10 11_10_01_11	stf Rn, PSA|SZ8 |I|PF stf Rn, PSA|SZ16|I|PF stf Rn, PSA|SZ32|I|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA + 1, 2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.
11_00_10_01 11_00_10_10 11_00_10_11	stf Rn, PSA|SZ8 |D stf Rn, PSA|SZ16|D stf Rn, PSA|SZ32|D	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD.
11_10_10_01 11_10_10_10 11_10_10_11	stf Rn, PSA|SZ8 |D|PF stf Rn, PSA|SZ16|D|PF stf Rn, PSA|SZ32|D|PF	• Source is a byte, half-word, or word target at the Rn address. Any further PD read instructions will trigger a byte, half-word, or word access to the source. • Source address is in incremented mode: PSA = PSA-1,2, or 4 after read PD. • 1, 2, or 4 bytes are *fetched* from the peripheral source.

Table continues on the next page...

Table 7-35. Peripheral DMA STF Instruction List (continued)

Binary	Assembly	Comments
11_00_11_01 11_00_11_10 11_00_11_11	stf Rn, PSA|SZ8 |U stf Rn, PSA|SZ16 |U stf Rn, PSA|SZ32 |U	• *Update* source pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost.
11_10_11_01 11_10_11_10 11_10_11_11	stf Rn, PSA|SZ8 |PF|U stf Rn, PSA|SZ16 |PF|U stf Rn, PSA|SZ32 |PF|U	• *Update* source pointer, which becomes a pointer to a target accessed in byte, half-word, or word. • PSA value is not modified by Rn. • Bytes present in PD are lost. • 1, 2, or 4 bytes are *fetched* from the memory source.
11_01_00_01 11_01_00_10 11_01_00_11	stf Rn, PDA|SZ8 |F stf Rn, PDA|SZ16|F stf Rn, PDA|SZ32|F	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is frozen.
11_01_01_01 11_01_01_10 11_01_01_11	stf Rn, PDA|SZ8 |I stf Rn, PDA|SZ16|I stf Rn, PDA|SZ32|I	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA + 1, 2, or 4 after write PD.
11_01_10_01 11_01_10_10 11_01_10_11	stf Rn, PDA|SZ8 |D stf Rn, PDA|SZ16|D stf Rn, PDA|SZ32|D	• Destination is a byte, half-word, or word target at the Rn address, and any further PD write instructions will trigger byte, half-word, or word access to the destination. • Destination address is in incremented mode: PDA = PDA-1, 2, or 4 after write PD.
11_01_11_01 11_01_11_10 11_01_11_11	stf Rn, PDA|SZ8 |U stf Rn, PDA|SZ16 |U stf Rn, PDA|SZ32 |U	• *Update* destination pointer to memory, which becomes a pointer to a memory accessed in byte, half-word, or word. • PDA value is not modified by Rn • bytes present in PD are lost
11_00_10_00	stf Rn, PD	• Write "dsize" bytes of Rn in PD and automatically flush to destination target
11_11_11_11	stf Rn, PS	• Write status register
11_00_11_00	stf Rn,clrefPS	• Clear error flag if set

NOTE

When writing PD, size information is not important: It is embedded in the dsize field of PDA register. If dsize is 1, 2, or 4, then one, two, or four bytes from Rn is written to the PD register, and automatically flushed out to the destination target.

7.2.3.12.2.6 Peripheral DMA Read (ldf)-Read Mode

When received from an ldf instruction, the function code bits are interpreted as follows.

Table 7-36. LDF Code Bits

Register	7	6	5	4	3	2	1	0
PSA	s			ar	a			
PDA								
PD			p	cpy				
PS			pssel					

Table 7-37. LDF Code Bits Descriptions

Field	Description
7-6 s	Functional Unit selector 11 for Peripheral DMA
5 p (PD)	Prefetch Flag 0 no prefetch 1 automatic prefetch
4 ar (PSA/PDA)	Address Register Selector 0 PSA 1 PDA
4 cpy (PD)	Copy Mode 0 standard access 1 copy mode access
3 a	Register Set selection 0 PSA or PDA 1 PD or PS
5-0 pssel	PS access selector 111111 is the only valid option to read PS

Table 7-38. Peripheral DMA LDF Instruction List

Binary	Assembly	Comments
11_0_0_0_000	ldf Rn, PSA	Reads 32-bit of PSA value
11_0_1_0_000	ldf Rn, PDA	Reads 32-bit of PDA value
11_0_0_1_000	ldf Rn, PD	Reads programmed source size bytes of PD (0-extended)
11_1_0_1_000	ldf Rn, PD|PF	Reads programmed source size bytes of PD (0-extended), and starts a prefetch at PSA address.
11_0_1_1_000	ldf Rn, PD|COPY	Starts a copy transfer from the source target at the PSA address to the destination target at the PDA address. No data transmits through Rn, but Rn contents are lost (Rn is loaded with PD temporary contents that are *not* the copied data).
11_111111	ldf Rn, PS	Reads 32-bit of PS value

NOTE

When reading PD, size information is not important: It is embedded in the ssize field of the PSA register. If ssize is 1, 2, or 4, the one, two, or four bytes is transferred from PD to Rn. Read data is 0-extended.

7.2.3.12.2.7 Peripheral DMA Unit Copy Mode

Like burst DMA, the peripheral DMA unit has a copy mode that is used when data transfers do not involve SDMA general registers.

Data is read from the source target at a PSA address, stored in PD, and then automatically flushed to the destination target at the PDA address. Copy mode is only available for transfers that involve two targets of the same data path width.

Since copy mode is invoked with an ldf instruction, the *loaded* general purpose register loses its previous contents. (However, the new contents are unpredictable as they depend on temporary values that are seen on the external DMA bus.)

7.2.3.12.2.8 Error Management

Peripheral DMA generates two kinds of errors: the immediate error that sanctioned incorrect register programming; and the error triggered by the previous access and stored in the error flag of PS until a DMA instruction is executed.

7.2.3.12.2.8.1 Immediate Errors

The following table lists all incorrect DMA register setups.

Table 7-39. Immediate Errors with Peripheral DMA

Rn[1:0] values	DMA instruction	Comments
0x01 0x11	stf Rn, PSA|SZ16|F stf Rn, PSA|SZ16|I stf Rn, PDA|SZ16|F stf Rn, PDA|SZ16|I	If PSA points to a half-word peripheral or to a half-word address in memory, its value must be 0 modulo 2.
0x01 0x10 0x11	stf Rn, PSA|SZ32|F stf Rn, PSA|SZ32|I stf Rn, PDA|SZ32|F stf Rn, PDA|SZ32|I	If PSA points to a word peripheral or to a word address in memory, its value must be 0 modulo 4.
PSA[1:0]-PDA[1:0]	DMA instruction	Comments

Table continues on the next page...

Table 7-39. Immediate Errors with Peripheral DMA (continued)

Rn[1:0] values	DMA instruction	Comments
0x01 0x10 0x11	stf Rn, PSA|SZ32|U stf Rn, PDA|SZ32|U	When PDA or PSA is updated and becomes a pointer to a word address in memory, its content must be 0 modulo 4.
0x01 0x11	stf Rn, PSA|SZ16|U stf Rn, PDA|SZ16|U	When PDA or PSA is updated and becomes a pointer to a half-word address in memory, its content must be 0 modulo 2.
Read/Write PD instruction	Comments	
stf Rn,PD ldf Rn,PD	If PDA size (dsize) has never been set up before an stf PD instruction (dsize=0) If PSA size (ssize) has never been set up before an ldf PD instruction (ssize=0)	
ldf Rn,PD|CPY	Copy mode is possible only between two targets whose data path width is identical. It is P8↔P8, P16↔P16, or P32↔P32 regardless of the way the address registers are incremented.	

7.2.3.12.2.8.2 Data Transfer Errors

When PSA and PDA are correctly set up, the only error that may arise for an ldf PD or stf PD instruction would be the error of the previous DMA cycle.

Error handling is driven by a single consideration: When an error occurred during a data read on the DMA interface, this error should appear as a transfer error to the core when the core attempts to retrieve the data that was not successfully read from the accessed device (memory or peripheral).

When an error occurred during a write access to the DMA interface, the data is still available in PD and should not be destroyed by subsequent core accesses: The core must be warned about the error issue.

There are three error handling mechanisms for each case: Read Error (First Phase), Write Error and Read Error (Second Phase), and Copy Mode Errors handling.

7.2.3.12.2.8.3 Read Error (First Phase)

If an error occurred during a prefetch command, the peripheral DMA enters its ERROR READ mode (PS[9:8]=11). In this mode, the error is reported on the next ldf PD instruction and writing PSA, PDA, or PD will cancel the error flag.

The block returns no error mode and instructions are normally executed (a DMA cycle may be triggered). Similarly, initiating a copy transfer will reset the error flag and start a copy transfer. The following table details which instructions can be executed in this mode.

Table 7-40. Possibilities in ERROR READ Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA (IU IPF) stf rn, PDA ldf rn,PD|COPY	NO	Error mode is reset, PSA or PDA are updated, or a write cycle is started. For the ldf PD|COPY, a copy loop is executed.
stf rn, PS	NO	PS is updated.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR READ mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Error of the previous read access is reported here and the peripheral DMA enters its ERROR mode.

7.2.3.12.2.8.4 Write Error and Read Error (Second Phase)

The peripheral DMA enters its ERROR mode (PS[9:8]=10) when the previous DMA write cycle failed, or, as explained in Read Error (First Phase), when an ldf PD is executed while the block is in ERROR READ mode. When a DMA cycle failed, address registers (PSA, PDA) are not modified and continue to point to the problematic address. In ERROR mode, stf instructions may raise an immediate error, and ldf instructions will not (as detailed in the table below).

Table 7-41. Possibilities in ERROR Mode

DMA Instruction	Immediate Error	Comments
stf rn, PD stf rn, PSA stf rn, PDA	YES	Any attempt to modify PD, PSA, or PDA will raise an immediate error, and the peripheral DMA stays in ERROR mode. When address registers are write accessed, an error is returned.
stf rn, PS	NO	This is the only way to exit the ERROR mode. PS[3] must be reset by an stf PS instruction.
ldf rn, PS ldf rn, PSA ldf rn, PDA	NO	PS, PSA, and PDA could be read in ERROR mode without any side effects (for example, no DMA cycle is triggered).
ldf rn, PD	YES	Whatever the DMA direction (read or write), an ldf rn, PD instruction will show an immediate error.

7.2.3.12.2.8.5 Copy Mode Errors

Because copy mode is a write access that follows a read access, there are two possible cases of bus error.

When the read access incurs a bus error, the peripheral DMA behaves exactly as described in Read Error (First Phase) and Write Error and Read Error (Second Phase) : It enters its ERROR READ mode, and so on.

When the error occurred during the write access of the copy transfer, the DMA enables the core to retrieve the data that was read because it is assumed the read from the peripheral removed the data from its source device. Therefore, the data to be flushed is still in PD. Any subsequent access to PD triggers an error to the core, which should execute its error handling procedure.

Once the ERROR mode is left (after writing to PS), it is possible for the core to retrieve the data in PD with an ldf instruction or try to flush PD contents once again (for example, when the error was due to a full FIFO and the script waited for the FIFO to be emptied) with another ldf instruction in copy mode. This latter instruction detects that there is valid data in PD, tries to flush it, and thus skips the read phase of the copy instruction. This is a different behavior from the usual stf PD instruction that overwrites PD with the selected General Purpose register contents. The same mechanism can be used any time PD holds data that is not written because of a bus error on the DMA interface; when the data was written via a copy instruction, or via the usual stf PD instruction.

7.2.3.12.2.8.6 Error Check Example

The following code illustrates an example checking for both immediate and data transfer errors on a store to the PD register. The first bdf instruction checks for an immediate error, but if a data transfer error occurred it is reported until the next instruction to access the Peripheral DMA. A second check of the error flags is done after the ldf PS instruction. The value of PS here can be ignored. The act of reading any register in Peripheral DMA while it is in an error mode that returns the error to the core to set either the SF or DF flag. Any error returned on an ldf command sets the SF flag and any error returned on an stf instruction sets the DF flag. This can create a situation as shown in the example where a bus error during a DMA write which would normally be considered as a destination fault is reported as a source fault because the error was reported to the SDMA core during an ldf instruction.

Table 7-42. Peripheral DMA Error Check

Function	Instruction	Comment
clrf	0	Clear SF and DF flags
stf	R4, PD	Write data to memory
bdf	error_routine	Check for immediate error from write to PD.
ldf	r3, PS	Read PS (PS value in R3 can be ignored)

Table continues on the next page...

Table 7-42. Peripheral DMA Error Check (continued)

Function	Instruction	Comment
bsf	error_routine	Check for bus error from "stf R4,PD". SF is set because it is a ldf instruction, even though the original error was a destination fault.

7.2.3.12.2.9 Peripheral DMA Unit Prefetch/Flush Management

There is no flush bit because every time data is stored in PD by a stf PD instruction-assuming PDA is correctly programmed-it is automatically flushed to the destination.

An acknowledge is returned in the cycle of the DMA instruction, and the SDMA is only stalled by an instruction that addresses the peripheral DMA when the previous DMA access is not over.

7.2.3.12.3 OnCE and Real-Time Debug

The On-Chip Emulation block (OnCE) is the debug interface to the SDMA.

It supports the access to all core internal devices (registers, memory, and so on), and provides a set of mechanisms that control the core. The OnCE is accessed by JTAG ports at the chip's board level, or by the host via its peripheral bus.

To reduce the size of the hardware material involved, all tasks supported by the OnCE are performed on the SDMA core. The architecture of the SDMA OnCE is relatively simple and very flexible.

The commands supported by the SDMA OnCE are listed in the following sections.

7.2.3.12.3.1 Memory and Register Access

A set of mechanisms is provided to access SDMA memory and register locations. Both reading and writing are allowed. The access is supported if the processor is in debug mode.

Those registers can also be accessed through the Arm platform Control interface when the OnCE is controlled by the Arm platform, as described in the "Using BP" section.

7.2.3.12.3.2 Hardware Breakpoints

An event detection unit is implemented to support memory breakpoints. The unit watches the data exchanged between the SDMA memory bus and the core.

A debug request is sent to the core when matching conditions occur. The unit supports mixed conditions based on address range, access type, and data value. Event detection unit configuration registers are memory mapped in the SDMA space (see Arm platform Channel 0 Pointer (SDMAARM_MC0PTR)): You can modify them through a regular memory access or the Arm platform control interface.

7.2.3.12.3.3 Watchpoints

One output pin is provided to monitor matching trigger conditions that are defined in the event detection unit.

7.2.3.12.3.4 Software Breakpoints

The SDMA instruction set contains a software breakpoint. Upon executing a software breakpoint instruction, the core suspends normal execution and enters debug mode.

No hardware step execution mode is implemented in the OnCE, but this feature may be implemented at the software level with this instruction.

7.2.3.12.3.5 Core Control

Commands are provided to monitor and control processor activity. You can halt the core, rerun the core from another address location, and get processor status.

Any hardware breakpoint on the instruction bus is not supported, but this feature may be implemented by inserting a software breakpoints program.

7.2.3.13 The OnCE Controller

The OnCE controller receives commands from the Arm platform or from the JTAG controller. Each command is interpreted before being sent to the core.

7.2.3.13.1 OnCE Commands

A small set of commands supports the communication between the OnCE and the external world.

This command set enables you to perform any of the following tasks: control processor activity, save core context, and execute an SDMA instruction from the OnCE. Combined together, these tasks perform more complex commands.

A full OnCE command contains a 4-bit instruction (the OnCE command opcode) and a variable length data field (the OnCE data). During command execution, the OnCE data is transferred in a OnCE internal register before being exchanged with the SDMA. Some data values are also exported. This mechanism creates a link between the processor and the external world. Nine commands are defined: The following table presents their formats.

Table 7-43. OnCE Command Opcode Values

Instruction Opcode	Name	Action	Register	Data Field Size	Mode
0000	rstatus	Reads the OnCE status register	STATUS	16-bit	normal/debug
0001	dmov	Updates general register GReg1	GREG1	32-bit	debug
0010	exec_once	Runs the instruction from the SDMA instruction register	INSTRUCTION	16-bit	debug
0011	run_core	Returns to normal execution	BYPASS	1-bit	debug
0100	exec_core	Returns to normal execution via a jump instruction that specifies the new address	INSTRUCTION	16-bit	debug
0101	debug_rqst	Stops the core after execution of current instruction	BYPASS	1-bit	normal
0110	rbuffer	Reads the real time buffer	RTB	32-bit	normal/debug
0111-1110	reserved	Reserved	BYPASS	1-bit	normal/debug
1111	bypass	Bypasses TARM platform controller	BYPASS	1-bit	normal/debug

Each instruction corresponds to a specific action performed on the OnCE. The nature of the associated data field is clearly identified. The dmov command is followed by a 32-bit data value (which is a data value for the SDMA); the exec_once and the exec_core commands are followed by a 16-bit data value (which is an instruction for the SDMA); the rstatus command is followed by a 16-bit control value (which is the content of the OnCE status register); the rbuffer command is followed by a 32-bit data value. The debug_rqst and the run_core commands are followed by a single bit data field (this is a bypass value). Finally, the bypass instruction enables the SDMA JTAG TAP controller to be daisy-chained with another JTAG TAP controller. This is a JTAG-only feature. The set of commands is simple, but enables you to perform any possible task on the SDMA during a debug process.

7.2.3.13.2 Sending Commands to the OnCE Controller

The JTAG access is the standard access to the OnCE, but sometimes the JTAG is not available to fix some bugs (if the chip is in production for instance), an additional access is then required. Therefore, one Arm platform access to the OnCE is provided.

7.2.3.13.2.1 Using the JTAG Interface

A serial access is performed through the five JTAG pins TCK, TRST, TMS, TDI, and TDO. A Test Access Port controller is provided to decode the TMS control signal.

It produces shift-enable signals (shift_ir and shift_dr), and updates enable signals (update_ir and update_dr). It is fully compliant with the IEEE 1149.1 testability (JTAG) standard.

During the shift_ir state, the command opcode is shifted into the OnCE controller (for example, the signal from the TDI pin is shifted into the command register and the TDO pin receives the signal shifted out). After transferring the four bits of the command, an update_ir signal is asserted and the command is decoded. The target data register is now clearly identified and the corresponding control signal is produced, as follows: bypass enable signal (bp_en), instruction enable signal (inst_en), data enable (data_en), and status enable signal (stat_en).

During the shift_dr state, the TDI signal is shifted into one of the following target registers: bypass register (1 bit), SDMA instruction register (16 bits), SDMA data register (32 bits), or OnCE status register (16 bits). The TDO pin is connected to the output of the selected register to receive the signals shifted out.

The JTAG access is disabled when the Arm platform access is enabled.

7.2.3.13.2.2 Using the Arm platform

The Arm platform access to the OnCE is not the standard access, but it is required if the JTAG is not available.

For example, if the SDMA ROM is out of use on a chip in production, and the Arm platform needs to download new code and restart the SDMA, the OnCE can easily perform this operation. This type of debug operation justifies the use of an Arm platform access to the OnCE.

To drive the OnCE, the Arm platform uses some registers contained in the Arm platform Control block of the SDMA. These registers are accessed through the Arm platform peripheral bus. Most of these registers are connected to another register in the OnCE controller. Thus, accessing one of these registers is equivalent to accessing the associated register in the OnCE controller.

The set of registers in the Arm platform Control block is listed below:

- ONCE_ENB register (1 bit, read/write)-This 1-bit register enables the Arm platform access to the OnCE. When this bit is set, the signals from the JTAG are ignored. When it is cleared, all writing operations to the following registers through the Host Control interface are ignored. This register is reset on a JTAG reset.
- ONCE_CMD register (4 bits, read/write)-This 4-bit register receives the command opcode. It is connected to the command register in the controller. A write access to this register causes the associated command to be executed on the OnCE. For example, after writing "0001" in this register, a dmov command is executed.

NOTE

On the Arm platform side, the rstatus and bypass commands are not supported. This register is reset on a JTAG reset.

- ONCE_DATA register (32 bits, read/write)-This 32-bit register is connected to the SDMA data register. This register is used when executing a dmov or rbuffer command.

NOTE

Before requesting a dmov command, the 32-bit data to transfer must be written in the ONCE_DATA register. At the end of the execution, the register is updated with GReg1 former value. This register is reset on a JTAG reset.

- ONCE_INSTR register (16 bits, read/write)-This 16-bit register is connected to the SDMA instruction register. This register is used when executing an exec_core or an exec_once command.

NOTE

Before requesting an exec_core or an exec_once command, the appropriate instruction must be written in the ONCE_INSTR register. This register is reset on a JTAG reset.

- ONCE_STAT register (16 bits, read only)-A read access to the ONCE_STAT register returns the content of the OnCE status register (OSTAT). This register is read only.
- The bypass register is not useful when the Arm platform controls the OnCE, therefore no register is defined in the Arm platform Control block to access the bypass register.

7.2.3.13.2.3 Conflicts Between the JTAG and the Arm platform Accesses

When Arm platform access to the SDMA OnCE is enabled (that is, when the bit in the ONCE_ENB register is set), the JTAG access is disabled. This guarantees that the block is not accessed at the same time on both sides.

It is possible to check whether the JTAG access to the SDMA OnCE is enabled from the JTAG port. When the JTAG access is disabled, the SDMA TDO always returns 1. The check requires the following steps:

- Execute a dmov command from debug mode (with neither 0xffffffff nor 0x0 as dmov value: 0x5a5a5a5a is good).
- Execute another dmov command (the value here is not important).
- The returned value from the latter dmov command should be the original one if the JTAG access is enabled; if it is 0xffffffff instead of the original input value, this means the JTAG access is disabled.

7.2.3.13.3 Executing a Command from the OnCE

All the commands defined in OnCE Commands can be accessed through the JTAG. The Arm platform can access all these commands except the rstatus command.

On the Arm platform side, the OnCE status is directly accessed by reading the ONCE_STAT register.

7.2.3.13.3.1 Nature of the Commands

Two types of commands may be distinguished. First, there are two commands that do not interact with the core: rstatus and rbuffer. Those commands may be requested at any time: They do not depend on the core status.

NOTE

Each of these commands exports a data value or a status value from the SDMA.

There are also commands that interact with the core: dmov, run_core, exec_core, exec_once, and debug_rqst. These commands are core status dependent, as follows:

- During user mode only the debug_rqst is taken into account.
- During debug mode, all these commands are taken into account except the debug_rqst. For example, an exec_once command requested while not in debug mode has no effect.

7.2.3.13.3.2 Execution Request

The SDMA starts executing a task in debug mode when requested by the OnCE controller. The execution starting time depends on the type of access used to communicate with the OnCE.

If the JTAG is used, the request is send after decoding the update_dr state in the TAP controller. Therefore, always cross this state when sending a command through the JTAG. If the OnCE is driven from the Arm platform side, the request is sent after detecting a write access to the ONCE_CMD register. All the registers involved in this operation must be loaded first.

The following is an example of an exec_core command execution from the Arm platform side: After writing '010' in the ONCE_CMD register, the OnCE controller asks the SDMA to execute the instruction contained in the ONCE_INSTR register. The instruction involved should be available in the ONCE_INSTR register before the beginning of the execution.

7.2.3.13.3.3 Command Execution

The following list shows the commands and details how each command is executed:

- rstatus command execution-The rstatus command exports the content of the OnCE status register (OSR). If the JTAG is used, the status information is captured in the OnCE status register during the capture_dr state, and shifted out after 16 TCK clock cycles in the shift_dr state. The rstatus command is not supported on the Arm platform side, but a status register is provided instead. The rstatus may be performed in both debug and user modes.
- dmov command execution-The dmov command accesses SDMA internal registers. Executing a dmov instruction exchanges the 32-bit data values between the SDMA data register and the general register GReg[1].
- If the JTAG is used, the content of GReg1 is captured in the SDMA data register during the capture_dr state, then it is shifted out after 32 TCK clock cycles in the shift_dr state. During the update_dr state, GReg1 is updated with the new, shifted-in

32-bit data value. If the OnCE is driven from the Arm platform side, the data values contained in GReg1 and the SDMA data register are exchanged after detecting a write access to the ONCE_CMD register. The ONCE_DATA register must therefore be loaded first.

- exec_once command execution-The exec_once command executes the instruction loaded in the SDMA instruction register. The command may only be requested from debug mode. The SDMA returns to debug mode at the end of the execution.
- Change of flow instructions as well as instructions that may cause a context switch are not supported: The comprehensive list comprises done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions.

  No other command should be requested before the SDMA returns to debug mode. The SDMA status (for example, whether it is in debug mode or not) can be detected by polling with the rstatus OnCE command, monitoring the debug_mode pin, or checking the OnCE Status Register (SDMAARM_ONCE_STAT) register via the Arm platform control interface.

NOTE

Most of the instructions are single-cycle, which omits the step of polling the status. Loads and stores to DMA units are typical instructions that might require this polling.

If the JTAG is used, the 16-bit instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. A request is sent to the core when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the request is sent to the SDMA when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must be therefore be loaded first.

- run_core command execution-The run_core command leaves debug mode and resume normal program execution. The next instruction executed is the last instruction decoded before entering debug mode. Be sure to restore core context before re-running the core. This procedure is detailed in Restoring the Context.
- If the JTAG is used, a 1-bit bypass value is shifted in the bypass register in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the core is rerun when detecting a write access to the ONCE_CMD register.

- exec_core command execution-The exec_core command resumes program execution from any address. The 16-bit instruction provided with the exec_core overwrites the last instruction decoded before entering debug mode. This command is designed to support change of flow instructions, so that a program execution can be restarted

from any address. After executing an exec_core command, the SDMA leaves debug mode. The exec_core command is usually used with a jmp instruction.

- If the JTAG is used, the 16-bit branch instruction is shifted in the SDMA instruction register after 16 TCK clock cycles in the shift_dr state. The SDMA is rerun when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the SDMA reruns when detecting a write access to the ONCE_CMD register. The ONCE_INSTR register must therefore be loaded first. For example, to restart the SDMA from the program address 0x100, the instruction loaded should be a jump to address 0x100 instruction.
- debug_rqst command execution-The debug_rqst command puts the SDMA in debug mode. If the JTAG is used, a 1-bit bypass value is shifted in the bypass register during the shift_dr state. A debug request is sent to the SDMA when the update_dr state is decoded in the TAP controller. If the OnCE is driven from the Arm platform side, the debug request is sent when detecting a write access to the ONCE_CMD register. When the SDMA is already in debug mode, this command is simply ignored.
- rbuffer command execution-The rbuffer command exports the content of the real time buffer (RTB). If the JTAG is used, the content of the real time buffer (RTB) is captured in the SDMA data register during the capture_dr state. The register is completely shifted out after maintaining the shift_dr state during 32 TCK clock cycles. If the OnCE is driven from the Arm platform side, the content of the RTB is captured in the ONCE_DATA register after detecting a write access to the ONCE_CMD register.
- bypass command execution-This command is only available from the JTAG interface. It enables daisy-chaining of the SDMA JTAG TAP controller with other JTAG TAP controllers. This command does not change the SDMA state and can be executed in any mode (run, debug, or sleep). It selects the bypass register of the TAP controller.

7.2.3.13.4 Registers Descriptions

See SDMACORE, and SDMAARM, for detailed information on each register.

7.2.3.13.4.1 Event Cell Counter Register (ECOUNT)

The event cell counter register is a 16-bit register that contains the number of times minus one that an event detection occurs before generating a debug request.

This register should be written before attempting to use the event detection counter during an event detection process. The event cell counter register is cleared on a JTAG reset.

7.2.3.13.4.2 Event Cell Address Registers (EAA or EAB)

The event cell contains two address registers-the event cell address register (a), called EAA, and the event cell address register (b), called EAB. Every address register is a 16-bit register that stores a user-defined address value. This value computes one of the following address conditions: addra_cond or addrb_cond. Every address register is cleared on a JTAG reset.

7.2.3.13.4.3 Event Cell Address Mask Register (EAM)

The event cell address mask register is a 16-bit register that contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before comparing addresses.

NOTE

There is a common address mask value for the two address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The event cell address mask register is cleared on a JTAG reset.

7.2.3.13.4.4 Event Cell Data Register (ED)

The event cell data register is a 32-bit register that contains a user-defined data value. This data value is an input for the data comparator, which generates the data_cond condition.

The event cell data register is cleared on a JTAG reset.

7.2.3.13.4.5 Event Cell Data Mask Register (EDM)

The event cell data mask register is a 32-bit register that contains a user-defined data mask value. This mask is applied to the data value latched from the memory bus before comparing data.

Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. The event cell data mask register is cleared on a JTAG reset.

7.2.3.13.4.6 Real Time Buffer Register (RTB)

The real Time Buffer register is a 32-bit register that stores and retrieves run-time information without putting the SDMA in debug mode.

Refer to Real Time Buffer for more details.

7.2.3.13.4.7 Event Control Register (ECTL)

The event cell control register is a 16-bit register that defines cell event occurrence conditions.

The event cell control register is cleared on a JTAG reset. See also OnCE Event Detection Unit for more details.

7.2.3.13.4.8 Trace Buffer (TB)

The Trace Buffer register retrieves the information in the Trace Buffer.

See Trace Buffer for more details.

7.2.3.13.4.9 OnCE Status Register (OSTAT)

The OnCE status register is a 16-bit register that contains processor and event detection unit status. The OSTAT is a read-only register.

Refer to OnCE Status Register (SDMAARM_ONCE_STAT) for detailed description of the individual fields in the OSTAT register.

The following figure shows the OSTAT structure.

Table 7-44. OnCE Status Register (OnCE)

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PST[3:0]				RCV	EDR	ODR	SWB	MST					ECDR[2:0]		

Where PST[3:0] is the SDMA core state, RCV is set when the real-time buffer (RTB) is modified. EDR, ODR, and SWB are set, respectively, when the SDMA has entered debug mode because of an external debug request, a OnCE debug_rqst command, or a software breakpoint. MST is set when the OnCE is controlled from the Arm platform control interface, and when ECDR is a three-flag set that shows the event cell condition(s) that put the core in debug mode. The OSTAT never provides more than one reason for entering debug mode.

There are two ways of accessing OSTAT content, as follows:

1. Send an rstatus command to the OnCE controller through the JTAG, or read the ONCE_STAT register through the Arm platform access. Executing the rstatus command through the JTAG can be performed in both user and debug modes.
2. Perform an SDMA read access to the location in the SDMA core memory map (OSTAT register) debug mode using the exec_once command. With this method of access, the SDMA state reflected by the PST (processor status bit) is always DATA.

The register may also be accessed by a running application.

7.2.3.13.5 JTAG Interface Requirements

Because the signals received from the JTAG (running on TCK) are transferred to the OnCE controller (running on the SDMA clock), a synchronization mechanism is required.

7.2.3.13.5.1 TCK Speed Limitation

In the JTAG top-level layer, the TDO signal is always captured on a TCK falling edge. To guarantee a stable TDO signal from the SDMA during this operation, a falling edge detection is performed on TCK.

Before being latched in the *I* flip-flop (see Figure 7-11) on TCK falling edge, the TDO signal must be stable at the input of the flip-flop. This condition is verified if the TCK period is superior to the following delay:

worst-case edge detection delay + negative-edge signal propagation delay + JTAG top-level logic propagation delay

The frequency relationship, TCK < CLK/8, limitation guarantees that all operations are performed as expected.

7.2.3.13.5.2 Synchronization Implementation

The figure found here shows the synchronization mechanism.

Flip-flops tck0, tck1, and tck2 perform falling- and rising-edge detections on TCK. They generate the posedge_detected and negedge_detected nets that are used to sample the TDI and TMS inputs into the respective tdi and tms flip-flops, and update the tdo flip-flop to yield the TDO output. In the design, the only signal that might go metastable is the output of the tck0 flip-flop. This signal is captured in the tck1 flip-flop and no logical operation is performed on it to minimize a metastability propagation risk.

The TDI and TMS flip-flops also cannot go metastable: The propagation time of the rising-edge detection signal through tck0, tck1, and tck2 guarantees that the TDI and TMS inputs are stable when captured in the TDI and TMS flip-flops.

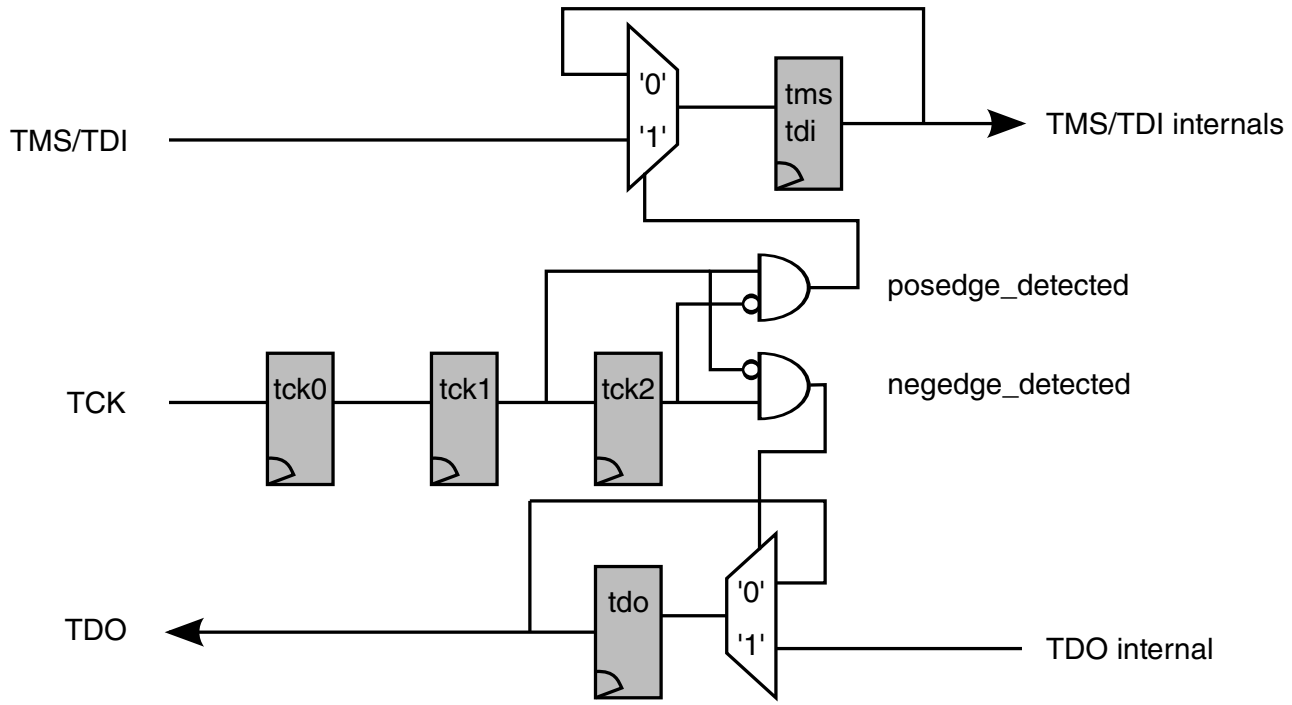


Figure 7-11. OnCE Synchronization Layer

The following figure shows synchronization timings. It takes three CLK clock cycles to synchronize TDI on the SDMA clock.

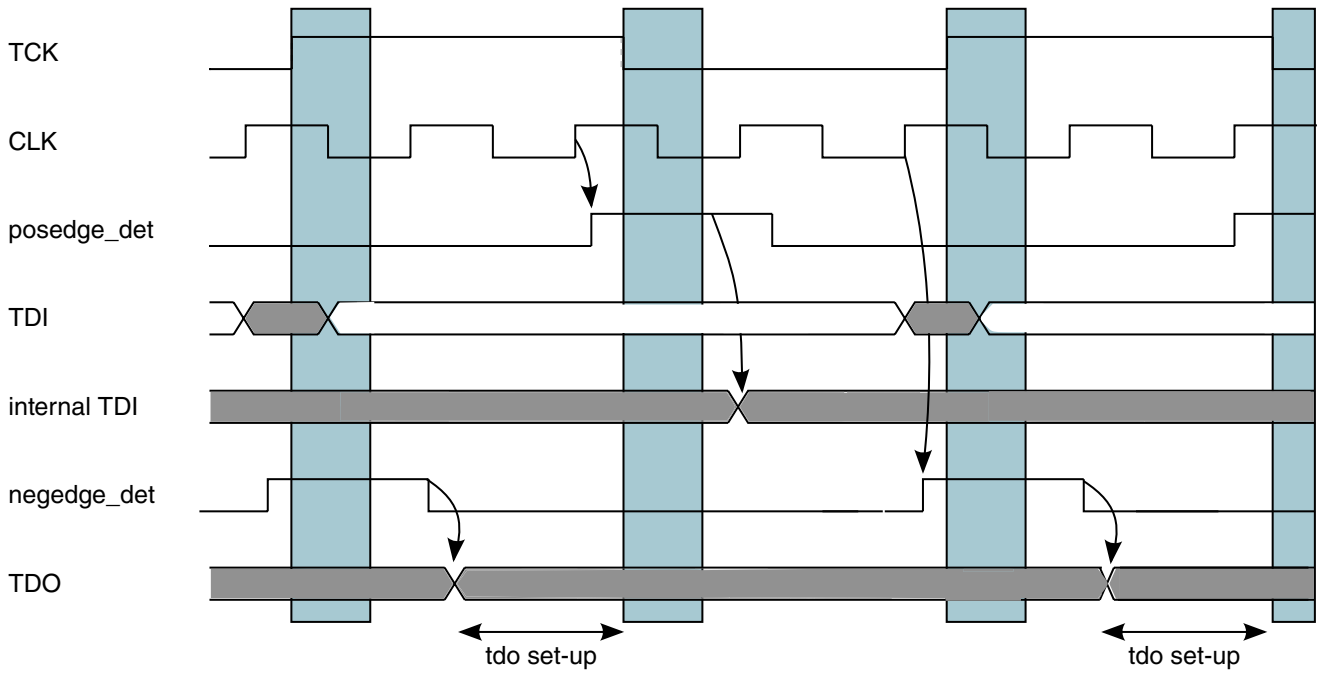


Figure 7-12. Synchronization Timings

7.2.3.13.5.3 JTAG Controller Start-Up Recommended Procedure

To ensure correct TAP controller initialization, it is recommended to use the following procedure:

1. Assert JTAG reset TRSTB (for example, set low).
2. Set TMS low.
3. Wait for 1 TCK clock.
4. Release JTAG reset TRSTB (for example, set high).
5. Wait for a minimum of five TCK cycles.

7.2.3.14 Using the OnCE

This section provides the elements necessary to run the OnCE during a debug process.

In addition to the basic set of commandsdescribed in OnCE Commands, more complex commands can be built to meet users' requirements.

7.2.3.14.1 Activating Clocks in Debug Mode

For power consumption issues, some clocks in the SDMA are disabled when not needed.

This is the case for instances when the SDMA is in sleep mode. Clock gating management depends on the interface used to control the OnCE.

- For the JTAG access, the SDMA clock gating must be turned off via the clk_gating_off input.
- For the Arm platform access, the SDMA clock gating is automatically turned off when the Arm platform access is enabled (see OnCE Enable (SDMAARM_ONCE_ENB)).

7.2.3.14.2 Getting the Current Status

Most of the commands the OnCE supports have an impact on the status of the SDMA.

It is not permissible to request the execution of an instruction on the SDMA from the OnCE while the SDMA is not in debug mode. Such a violation may cause unpredictable behavior, and it might be necessary to reset the SDMA.

Therefore, the value of the PST bits provided in the OnCE status register should always be checked before sending any request to the SDMA.

7.2.3.14.3 Methods of Entering Debug Mode

A debug request may be asserted at any time, but it is not always taken into account immediately. Debug mode cannot be entered in the middle of an instruction, or during the save or restore states of a context switch.

The request is ignored when the core is already in debug mode. Refer to Figure 7-4, which shows all possible transitions to the debug state, as there are several ways to enter debug mode.

7.2.3.14.3.1 External Debug Request During Reset

To enter debug mode after exiting reset, the external debug line has to be maintained high. This line is handled by the JTAG top-level block.

NOTE

The SDMA detects the debug requests only if the SDMA clock is running (see Activating Clocks in Debug Mode). The debug request line should be not be maintained high when the SDMA is in debug mode.

NOTE

The debug_rqst command (from the OnCE command set) is not supported during system reset.

7.2.3.14.3.2 Debug Request During Normal Activity

During normal activity, the SDMA enters debug mode when the following is true:

1. If the debug request line from the JTAG top-level is asserted, or
2. If the OnCE controller receives a debug_rqst command.

   The debug_rqst command can be sent by the JTAG access or by an access on the Arm platform side (if the Arm platform access is enabled).

7.2.3.14.3.3 Software Breakpoint Instruction

The SDMA enters debug mode at the end of the execution of a software breakpoint instruction. This instruction must be inserted in program flow executed by the core.

7.2.3.14.3.4 Event Detection Unit Matching Condition

If the event detection is enabled, a debug request is sent to the core after detecting a matching condition on the SDMA memory bus.

See OnCE Event Detection Unit for more details.

7.2.3.14.4 Executing Instructions in Debug Mode

The OnCE supports a mechanism to execute instructions in debug mode. If the SDMA is in debug mode, then the exec_once command can be used to execute an SDMA instruction from the OnCE controller. The SDMA returns to debug mode at the end of each execution.

Some instructions are not supported by the exec_once command: done/yield/yiedge (except done 5), BF, BT, BSF, BDF, JMP, JSR, JMPR, JSRR, RET, and LOOP, as well as all the illegal instructions are not supported.

NOTE

While instructions are executed in debug mode from the OnCE, the program counter of the SDMA is not incremented.

7.2.3.14.5 Command Sequences Examples

This section provides examples of command sequences that run the SDMA in debug mode. These sequences are available for both the Arm platform and JTAG accesses.

The following presents the syntax used in this section. The data field provided with each command is put in parenthesis with the command name. A '-' is used if the data field provided is a *don't care* value.

```
my_command(data_field);                 // executing my_command with a data field
my_command(-);                        // executing my_command with a don't care data field
```

The value returned by the command (if there is one) is referred by an assignment. In case the value returned by the command is not used, the assignment is omitted. For an Arm platform access, the value returned (it is always a data value) is obtained by reading back into the SDMA data register.

```
data_out = my_command(data_in); // returning a data value
```

To clarify the syntax, the instructions' opcodes are referred to by their names. In practice, use the corresponding 16-bit encoding.

7.2.3.14.5.1 Getting the SDMA Status

NOTE

Before executing any command that affects the SDMA (like dmov or exec_once), check that the SDMA is in debug mode.

Use the following snippet:

```
rstatus();        // read SDMA status until the SDMA is in debug mode
...
rstatus();
```

If the SDMA is not in debug mode, then a debug request must be generated. In this case, the SDMA enters debug mode at the end of the execution of the current instruction. Use this snippet:

```
debug_rqst(-);   // debug request
```

In the following sections, it is assumed that the SDMA was successfully put into debug mode.

7.2.3.14.5.2 Saving the Context

The first debug task is to save the SDMA context, which is the content of the eight general-purpose registers, the loop and PC-related registers, and the flags.

Use the general register GReg[1] as an intermediate register to export the entire context of the SDMA.

The following example shows how to save GReg[0], GReg[1], GReg[2] and GReg[3]. The sequence of commands used to export additional general registers is very similar to this.

Save GReg[0], GReg[1], GReg[2], and GReg[3]

```
GReg1_data = dmov(-);                          // the value exported is the content of
GReg[1]
exec_once("mov GReg1,GReg0");                      // puts the content of GReg[0] into
GReg[1]
GReg0_data = dmov(-);                          // the value exported is the content of
GReg[0]
exec_once("mov GReg1, GReg2");                     // puts the content of GReg[2] into
GReg[1]
GReg2_data = dmov(-);                          // the value exported is the content of
GReg[2]
exec_once("mov GReg1, GReg3");                     // puts the content of GReg[3] into
GReg[1]
GReg3_data = dmov(-);                          // the value exported is the content of
GReg[3]
```

Get the value of the internal flags (SF, DF, T, and LM), of the loop related registers (EPC and SPC), and of the PC-related registers (PC and RPC). Use a done 5, which is the formatting instruction dedicated to the debug. This instruction formats the flags and the values contained in the registers. It also writes the resulting values into the channel context memory. It should not be used when entering debug from the IDLE state (for example, with no active channel script running on the SDMA), because it will update a channel context that may belong to any channel.

```
exec_once("done 5");                    // formatting the value of flags and registers
```

At this point, the channel context should be up-to-date in memory, and debug operations should now be possible. However, the context can be exported with the following instructions:

Exporting the Context

```
dmov(ctx_base_addr);                              // loading GReg[1] with the channel
context base address
exec_once("ld GReg0,(GReg1,0)");                      // get RPC-PC into GReg0
exec_once("ld GReg1, (GReg1,1)");                     // get SPC-EPC into GReg1
Loop_data = dmov(-);                           // read back the value of Loop registers
exec_once("mov GReg1, GReg0");                     // puts the PC info into GReg1
PC_data = dmov(-);                             // reads back the content of the PC registers
```

After this sequence of operations, the entire SDMA context is exported via the OnCE.

7.2.3.14.5.3 Restoring the Context

At this point in the operation, restore the context of the SDMA. It can be different from the original context located in memory, and the content previously saved into the debugging application via the OnCE.

The example found hereshows how it is possible to modify the current channel context.

Modifying the Current Channel Context

```
dmov(Loop_data);                                  // put Loop former value into GReg[1]
exec_once("mov GReg0, GReg1");                        // copy to GReg[0]
dmov(PC_data);                                    // put PC former value into GReg[1]
exec_once("mov GReg2, GReg1");                        // copy to GReg[2]
dmov(ctx_base_addr);                                 // put channel context base address into
GReg[1]
exec_once("st GReg0, (GReg1,1)");                          // restore Loop context
exec_once("st GReg2, (GReg1,0)");                          // restore PC context
```

Once the context in memory is the desired context (with or without applying the previous instruction sequence), it can be restored to the *real* PC and loop registers in the SDMA core:

```
exec_once("cpShReg");                    // restore flags and PC & loop related registers
```

After this command, the SDMA core PC, RPC, SPC, EPC registers, as well as the flags contain the same data as what is stored in the context RAM for the current channel.

The following example shows how to restore the context of general registers GReg[0], GReg[1], GReg[2] and GReg[3].

Restoring the General Register Context

```
dmov(GReg3_data);                    // put GReg[3] restore value in GReg[1]
exec_once("mov GReg3, GReg1"); // restore GReg[3]
dmov(GReg2_data);                    // put GReg[2] restore value in GReg[1]
exec_once("mov GReg2, GReg1"); // restore GReg[2]
dmov(GReg0_data);                    // put GReg[0] restore value in GReg[1]
exec_once("mov GReg0, GReg1"); // restore GReg[0]
dmov(GReg1_data);                    // restore GReg[1]
```

At this point, it is possible to restart the normal program execution.

NOTE

Every SDMA core general register value can be modified by a mov instruction, which makes modification of these registers easy during debug. Unfortunately, there is no such instruction as a mov to directly modify the contents of either PCU register or flag (PC, RPC, SPC, EPC, T, LM, SF, or DF). The cpShReg instruction is meant to provide a means for changing these register contents via the context memory.

7.2.3.14.5.4 Accessing the Memory

In the example shown here, it is assumed that the SDMA context is entirely saved. If true, it is permissible to modify the general purpose registers during debugging activity.

To perform a memory read access, the target address is stored via the OnCE in GReg[1], then the load instruction is executed on the SDMA (the data loaded from the memory overwrites the address contained in GReg[1]), and then the result value is read back via the OnCE.

```
macro READ:            dmov(target_addr);                                  // put the target
address in GReg[1]
                    exec_once("ld GReg1,(GReg1,0)");                        // execute the
load instruction
                    res_data = dmov(-);                                  // exports the result
data value
```

For a memory write access, the target address is written in GReg[0], and the value to store is written in GReg[1]. Then the store instruction is executed on the SDMA.

```
macro WRITE:            dmov(target_addr);                                  // puts the
target address in GReg[1]
                    exec_once("mov GReg0,GReg1");                        // puts the target
address in GReg[0]
                    dmov(target_data);                                   // puts the target
data in GReg[1]
                    exec_once("st GReg1,(GReg0,0)");                        // performs the
store operation
```

This sequence is shown as an example; however, many other sequences are possible.

NOTE

This sequence of commands can also be applied to memory-mapped registers.

7.2.3.14.5.5 Resuming Program Execution

Before resuming program execution, it is assumed that the SDMA context is properly restored. There are two ways to restart the SDMA.

Start by executing the last instruction fetched before entering debug mode, as follows.

```
run_core(-);                          // resume execution from where we stopped before
```

If necessary, restart the execution from a different address. In this case, use the exec_core command. The data field provided with this command must be the encoding of a jump instruction.

```
exec_core("jmp start_addr"); // rerun the SDMA from another address
```

In these two examples, the SDMA exits debug mode and keeps executing the code fetched from the memory.

7.2.3.14.5.6 Single Stepping in RAM

To execute a program step-by-step from the RAM, insert software breakpoints in the program flow at appropriate places so that the SDMA only executes one instruction before returning to debug mode.

First, read the next instruction to execute in the RAM. Then, depending on the value of this instruction, compute the address where a software breakpoint instruction should be inserted. The instruction at the corresponding address must be saved, and, the software breakpoint instruction is inserted. After restarting the SDMA, there is only one instruction executed before meeting the software breakpoint.

The following example shows the macro functions READ and WRITE, which correspond to the sequence of commands (described above) used to access the memory.

NOTE

The data read from the memory are 32-bit values, while the instructions are 16-bit values only. This is why it is best to only use addresses divided by two when accessing the memory.

READ and WRITE Macro Functions

```
next_instr = READ(run_addr/2);                          // read the next instruction to execute
// the tool now has to compute the address where the breakpoint
// instruction should be inserted, this address is the "bkpt_addr"
instr_save = READ(bkpt_addr/2);                          // save the instruction before
overwriting
STORE("bkpt instruction",bkpt_addr/2);                        // store the bkpt instruction
in memory
exec_core("jmp run_addr");                              // rerun the SDMA
rstatus(-);                                          // wait for the SDMA to enter debug mode
...
rstatus(-);
STORE(instr_save,bpkt_addr/2);                           // restore the instruction
overwritten
```

In case of branched conditional instructions, a breakpoint instruction should be written at the two possible target addresses.

7.2.3.14.5.7 Single Stepping in ROM

No single-step mechanism is supported in ROM. The program code can be loaded in the RAM, where the single-step mechanism can be executed.

7.2.3.14.6 OnCE Event Detection Unit

The event detection unit watches signals from the data memory bus (DMBUS), which the SDMA core uses to access its RAM, ROM, and memory mapped registers.

A debug request is sent to the OnCE controller when user-defined conditions on address and/or data values are true.

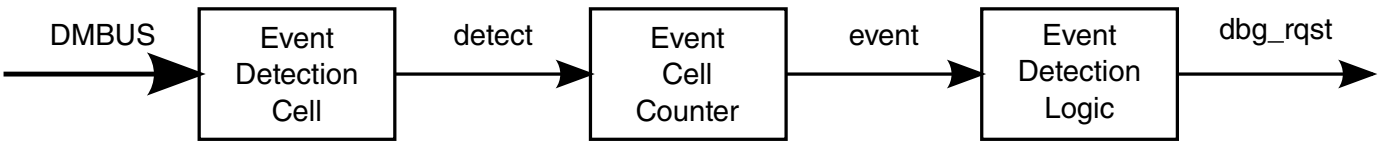


Figure 7-13. Event Detection Unit

A counter, provided with the detection cell, is decreased after an event detection. A debug request is sent to the core only when the counter reaches the value of 0. It is possible to disable the use of the counter if a debug request has to be generated after each event detection.

The event cell is the basic block that supports hardware breakpoints on an address value and/or data values coming from the SDMA memory bus. The trigger condition that generates the debug request is a mixed condition based on those values.

The following figure shows the event cell architecture. The event cell contains the address (stored in the memory address register) and the data (stored in the memory data register) used during the last memory access. There are some user-defined reference values located in memory mapped registers-the event cell addresses, the event cell address mask, the event cell data, and the event cell data mask. These registers are accessed by standard load/store instructions just like regular memory locations.

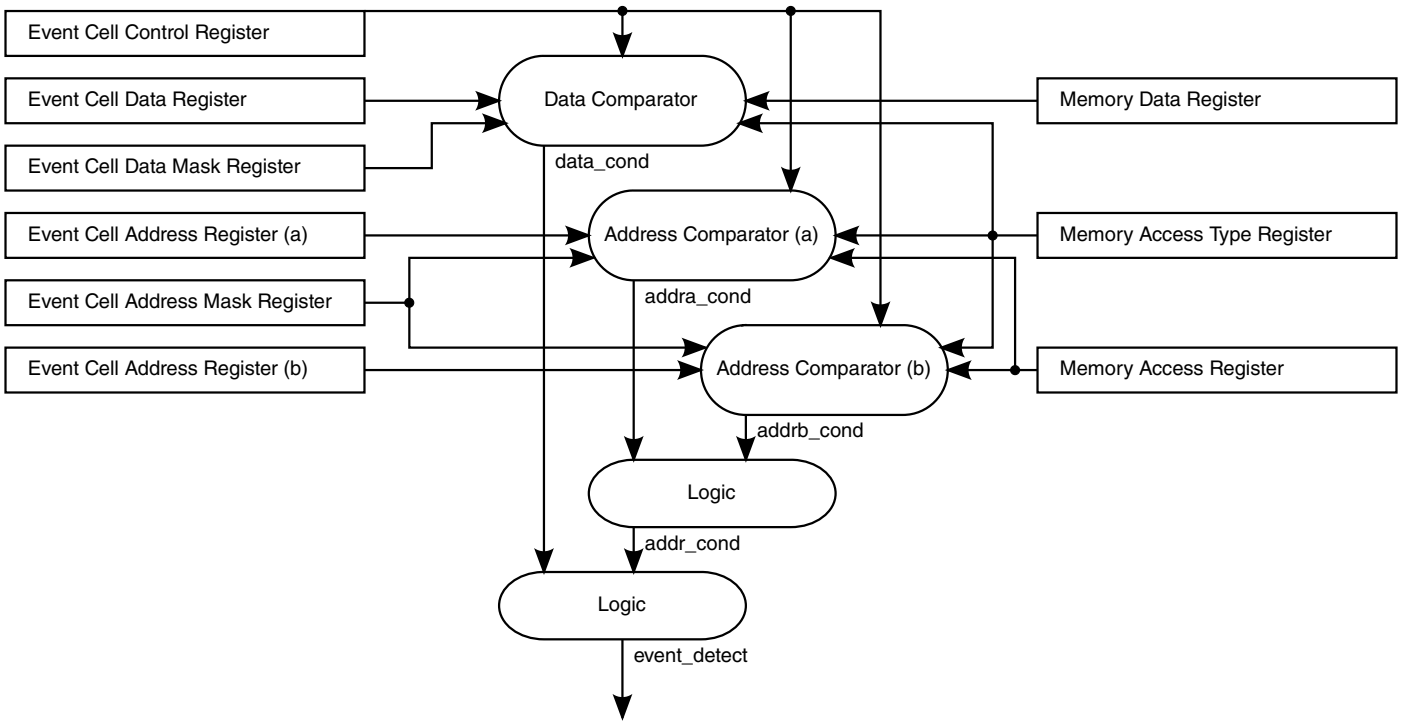


Figure 7-14. Event Cell Architecture

To define a memory breakpoint, three conditions are taken into account: The first two conditions are comparisons of the current memory address with user-defined reference addresses (these conditions are called addressA and addressB). The third condition consists of a comparison between the data received on the DMBUS and a user-defined reference data (this condition is called data). An intermediate address condition is set to express a dependency between addressA and addressB conditions.

7.2.3.14.7 Clock Gating and Reset

This section details how to use the clocks and handle the reset signals.

7.2.3.14.7.1 Clocks

Because the SDMA uses clock gating to save power, it is necessary to disable the clock gating and force the clocks to be enabled when using the OnCE.

When the OnCE is accessed through its JTAG interface, clock gating must be disabled outside the SDMA via a dedicated SDMA input port clk_gating_off. The reason why detection is not performed automatically by the SDMA internal hardware is that it would cost power to monitor activity on the JTAG interface.

When the OnCE is accessed through the Arm platform Control interface, clock gating is automatically turned off. This is done when bit 0 of the ONCE_ENB register (see OnCE Enable (SDMAARM_ONCE_ENB)) is set. A write access to this register is possible even when the OnCE clock is not running. If the Arm platform access is used, the bit in the ONCE_ENB register must be set before any attempt to access any other OnCE register.

7.2.3.14.7.2 Resets

The OnCE reset is different from the SDMA main reset.

Normally, activating the SDMA reset while keeping the OnCE reset inactive (when possible) enables you to reset the core without having to reprogram the OnCE.

7.2.3.14.8 Real Time Features

To rebuild the skeleton of a program execution, it is necessary to store the addresses of the program instructions where jumps are taken: A trace buffer is therefore provided. A real time buffer has also been added to receive data values written during a program execution.

The content of this register may be exported through JTAG ports without stopping the core.

7.2.3.14.8.1 Trace Buffer

The Trace Buffer is a 32-stage buffer that contains appropriate information to identify the 32 last changes of flow detected during a program execution.

The following figure shows an overview of the Trace Buffer.

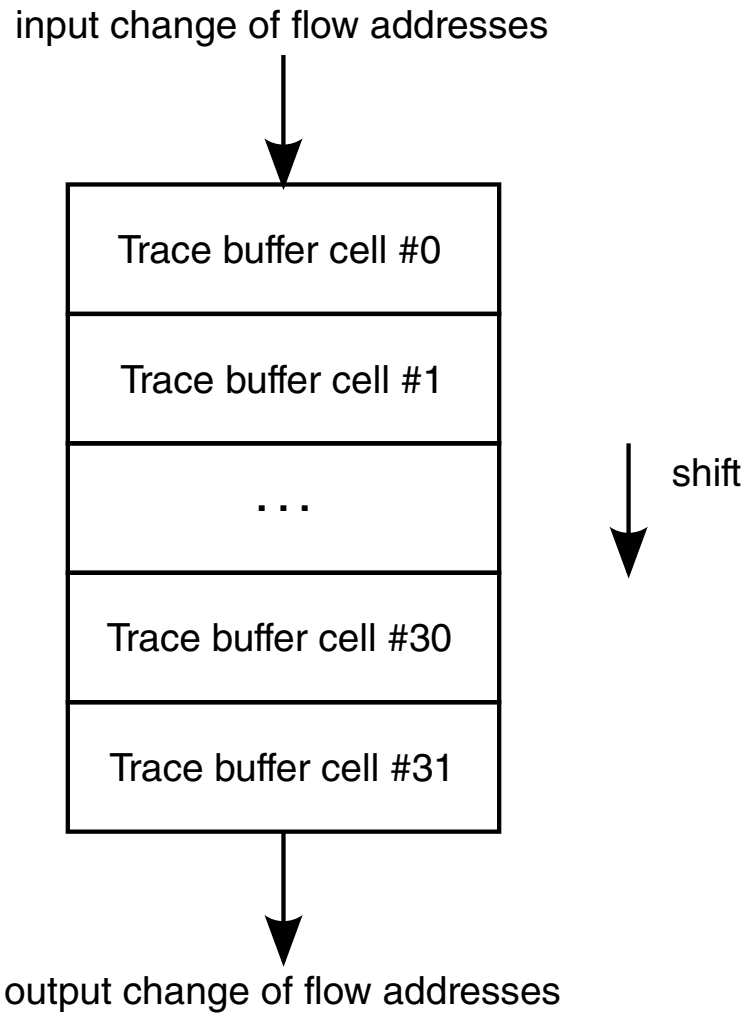


Figure 7-15. Trace Buffer

Each cell of the trace buffer contains two reference addresses and a flag. The flag is set when the addresses stored in the cell correspond to a valid change of flow; otherwise, the flag is cleared. The three most significant bits are unused.

After every change of flow detection, the address of current instruction and the address of the target instruction are stored at the top of the Trace Buffer (cell #0). The flag in the cell is set to indicate that a valid change of flow was detected. Former cell values are shifted one level down. The Trace Buffer contains the 32 last changes of flow. All the flags are reset on a software or a hardware reset, and after each transition from debug mode to user mode.

A memory mapped register of SDMA core, the Trace Buffer register (TB), is provided to read the content of the Trace Buffer. This operation should be done in debug mode. Performing a read access to the Trace Buffer register returns the content of the bottom of the Trace Buffer (cell #31). After every read access, the trace buffer is shifted one level down, and the flag at the top of the trace buffer is cleared.

A typical OnCE command sequence that retrieves the oldest change-of-flow information is a follows:

```
exec_once("mov r1, TB");                              // stores the oldest change-of-flow in
GReg1
dmov(-);                                          // retrieves GReg1 contents
```

This sequence requires the SDMA to be put in debug mode.

7.2.3.14.8.2 Real Time Buffer

The Real Time Buffer register (RTB) is a memory mapped register that can be accessed as a regular memory location by the SDMA core during program execution. This register is located in the OnCE.

Executing ar rbuffer command (see The OnCE Controller for further details) exports the content of this register through JTAG ports.

When a write access is performed at the memory location corresponding to the RTB, the receive flag (for example, the RCV bit) is set in the OnCE Status Register (OSR). This flag is cleared at the end of the execution of a rbuffer command.

NOTE

Every write access to the RTB memory location updates the RTB register even if the RCV flag is set. The RTB is cleared on a JTAG reset.

7.2.3.14.8.3 Emulation Pin

The debug_matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit.

Since it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

7.2.3.14.8.4 Real-Time Debug Outputs

The table found here shows the debug signals that are available at the SDMA boundaries. Their availability at chip boundaries depends on the project.

Table 7-45. Real-Time Debug Output Pins

Pin	Description
debug_core_state[3:0]	The core_state bits reflect the state of the SDMA core. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the core idle state); the "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 is executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: they are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Context Switch Saving Channel 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change of Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Context Switch Restoring Channel
debug_yield	Pulse that is active when a yield (done 0) or a yieldge (done 1) instruction is executed. 0 - 1 yield/yieldge executed
debug_core_run	Active when the SDMA core is executing instructions. 0 Debug or sleep mode

Table continues on the next page...

Table 7-45. Real-Time Debug Output Pins (continued)

Pin	Description
	1 Run mode
debug_event_channel_sel	Indicates if debug_event_channel displays current channel or last received event 0- debug_event_channel[5:0] gives the number of the current channel 1- debug_event_channel[5:0] gives the number of the last received event
debug_event_channel[5:0]	Gives the number of any DMA request as soon as it is received or the number of the current channel. The value of debug_event_channel_sel indicates if debug_event_channel displays the current channel or last received event. The signal debug_event_channel_sel must be observed to determine what information is provided on debug_event_chanel at any given time.
debug_pc[13:0]	Program Counter value; it has a meaning when the core is in run mode.
debug_mode	Set when the core is in debug. 0 - 1 Core is in debug
debug_bus_error	Set when an error was received during a load or a store (ld, st, ldf, or stf instruction) and registered in SF or DF flag. 0 No error during last load/store 1 Error during last load/store
debug_bus_device[4:0]	Indicates the device or functional unit that is accessed by the current instruction. The debug_bus_device output is always valid when in sleep mode, debug mode, or executing any instruction that does not access the functional units or the memory mapped devices, "no access" is output. 0 No access 1 MSA 2 MDA 3 MD 4 MS 5 PSA 6 PDA 7 PD 8 PS 9 RESERVED 10 RESERVED 11 RESERVED 12 RESERVED 13 CA 14 CS 15 Reserved 16 Memory (RAM or ROM) 17 Memory mapped register

Table continues on the next page...

Table 7-45. Real-Time Debug Output Pins (continued)

Pin	Description
	18 Peripheral #1 19 Peripheral #2 20 Peripheral #3 21 Peripheral #4 22 Peripheral #5 23 Peripheral #6 24 Peripheral #7 25 Peripheral #8 26 Peripheral #9 27 Peripheral #10 28 Peripheral #11 29 Peripheral #12 30 Peripheral #13 31 Peripheral #14
debug_bus_rwb	Indicates the direction of the access given by debug_bus_device 0 Write access (st or stf) 1 Read access (ld or ldf)
debug_matched_dmbus	Pulse indicating the OnCE event detection unit has detected a match on the data bus during an access to memory (RAM or ROM), a memory mapped register or a peripheral that is hooked to the SDMA. 0 - 1 data bus match detected
debug_rtbuffer_write	Pulse indicating when the real-time buffer is written by the core. 0 - 1 RTB was modified
debug_evt_chn_lines[7:0]	Eight lines that generate short pulses when DMA requests are received or channels are (re)started. Every line is controlled through two parameters defined in registers Cross-Trigger Events Configuration Register 1 (SDMAARM_XTRIG_CONF1) (as described in SDMAARM). The following two parameters are available for every line: • CNF-Indicates what is monitored on the line: 0 for a channel start, 1 for a DMA request reception • NUM[5:0]-Gives the number of the DMA request or channel to monitor

The matched_event emulation pin reflects the matching condition status detected by the Event Detection Unit. Because it can be necessary to detect conditions without triggering debug requests, it is possible to disable the generation of debug requests by the Event Detection Unit and still have the matching condition available on the emulation pin. This can be done by clearing the EN flag in the ECTL register.

All real-time debug outputs are disabled by default (for example, they are stuck to 0) to avoid power consumption when they are not used. They are enabled when bit 11 (RTDOBS) of the Configuration Register (SDMAARM_CONFIG) is set. Signals provided to the system JTAG controller for SDMA debug mode status will also be enabled when the *clk_gating_off* input is asserted.

7.2.4 Instruction Set

7.2.4.1 Instruction Encoding

This section presents a short summary of the instruction codes. All context switch instructions are listed for information only; they cannot function properly out of the context switch routine.

```
x...x - don't care

rrr - destination/source general register

sss - additional source general register

bbb - general register used as address base register

ddddd - address displacement

nnnnn - bit number
uuuuuuuu - function unit command bits

pppppppp - branch displacement (signed)

iiiiiiii - 8-bit immediate

jjj - control bit to clear

ff - flag to clear
00000jjj00000000        - done (done,yield,wait)
00000jjj00000001        - notify
00000xxx00000010        - reserved
00000xxx00000011        - reserved
00000xxx00000100        - reserved
0000000000000101     - softBkpt
0000000100000101     - reserved
0000001000000101     - reserved
0000001100000101     - reserved
0000010000000101     - reserved
0000010100000101     - reserved
0000011000000101     - reserved
0000011100000101     - reserved
0000000000000110     - ret
0000000100000110     - reserved
0000001000000110     - reserved
0000001100000110     - reserved
0000010000000110     - reserved
0000010100000110     - reserved
0000011000000110     - reserved
0000011100000110     - reserved
```

```
000000ff00000111       - clrf ff
0000010000000111     - reserved
0000010100000111     - reserved
0000011000000111     - reserved
0000011100000111     - illegal
00000rrr00001000        - jmpr r
00000rrr00001001        - jsrr
00000rrr00001010        - ldrpc r
00000rrr00001011        - reserved
00000rrr000011xx          - reserved
00000rrr00010000        - revb
00000rrr00010001        - revblo
00000rrr00010010        - rorb
00000rrr00010011        - reserved
00000rrr00010100        - ror1
00000rrr00010101        - lsr1
00000rrr00010110        - asr1
00000rrr00010111        - lsl1
00000rrr001nnnnn        - bclri r,n
00000rrr010nnnnn        - bseti r,n
00000rrr011nnnnn        - btsti r,n
00000xxx10000xxx           - reserved
00000rrr10001sss           - mov
00000rrr10010sss           - xor
00000rrr10011sss           - add
00000rrr10100sss           - sub
00000rrr10101sss           - or
00000rrr10110sss           - andn
00000rrr10111sss           - and
00000rrr11000sss           - tst
00000rrr11001sss           - cmpeq
00000rrr11010sss           - cmplt
00000rrr11011sss           - cmphs
0000011011100000     - reserved
0000011011100001     - reserved
0000011011100010     - cpShReg
0000011011100011     - reserved
0000011011100100     - reserved
0000011011100101     - reserved
0000011011100110     - reserved
0000011011100111     - reserved
00000xxx11101xxx           - reserved
00000xxx11110xxx           - reserved
00000xxx11111xxx           - reserved
00001rrriiiiiiii                - ldi r,i
00010rrriiiiiiii                - xori r,i
00011rrriiiiiiii                - addi r,i
00100rrriiiiiiii                - subi r,i
00101rrriiiiiiii                - ori r,i
00110rrriiiiiiii                - andni r,i
00111rrriiiiiiii                - andi r,i
01000rrriiiiiiii                - tsti r,i
01001rrriiiiiiii                - cmpeqi r,i
01010rrrdddddbbb        - ld r,(d,b)
01011rrrdddddbbb        - st r,u
01100rrruuuuuuuu        - ldf r,u
01101rrruuuuuuuu        - stf r,u
011100xxxxxxxxxx              - reserved
011101xxxxxxxxxx              - reserved
011110ffnnnnnnnn       - Loop ff flags are reset
01111100pppppppp     - bf pc=pc+signed(pppppppp)+1
01111101pppppppp     - bt pc=pc+signed(pppppppp)+1
01111110pppppppp     - bsf pc=pc+signed(pppppppp)+1
01111111pppppppp     - bdf pc=pc+signed(pppppppp)+1
10aaaaaaaaaaaaaa     - jmp absolute
11aaaaaaaaaaaaaa     - jsr absolute
```

7.2.4.2 SDMA Instruction Set

This section describes all the useful instructions from the SDMA set.

Table 7-46. SDMA Instruction List

Instruction	Description	Page
ADD	Addition	ADD (Addition)
ADDI	Add with Immediate Value	ADDI (Add with Immediate Value)
AND	Logical AND	AND (Logical AND)
ANDI	Logical AND with Immediate Value	ANDI (Logical AND with Immediate Value)
ANDN	Logical AND NOT	ANDN (Logical AND NOT)
ANDNI	Logical AND with Negated Immediate Value	ANDNI (Logical AND with Negated Immediate Value)
ASR1	Arithmetic Shift Right by 1 Bit	ASR1 (Arithmetic Shift Right by 1 Bit)
BCLRI	Bit Clear Immediate	BCLRI1 (Bit Clear Immediate)
BDF	Conditional Branch if Destination Fault	BDF (Conditional Branch if Destination Fault)
BF	Conditional Branch if False	Functional Units Programming Model
BSETI	Bit Set Immediate	BSETI (Bit Set Immediate)
BSF	Conditional Branch if Source Fault	BSF (Conditional Branch if Source Fault)
BT	Conditional Branch if True	BT (Conditional Branch if True)
BTSTI	Bit Test immediate	BTSTI (Bit Test immediate)
CLRF	Clear Arm platform flags	CLRF (Clear Arm platform flags)
CMPEQ	Compare for Equal	CMPEQ (Compare for Equal)
CMPEQI	Compare with Immediate for Equal	CMPEQI (Compare with Immediate for Equal)
CMPHS	Compare for Higher or Same	CMPHS (Compare for Higher or Same)
CMPLT	Compare for Less Than	CMPLT (Compare for Less Than)
cpShReg	Update Context of PCU Registers and Flags	cpShReg (Update Context of PCU Registers and Flag)
DONE	DONE, Yield	DONE (DONE, Yield)
ILLEGAL	ILLEGAL Instruction	ILLEGAL (ILLEGAL Instruction)
JMP	Unconditional Jump Immediate	JMP (Unconditional Jump Immediate)
JMPR	Unconditional Jump	JMPR (Unconditional Jump)
JSR	Unconditional Jump to Subroutine Immediate	JSR (Unconditional Jump to Subroutine Immediate)
JSRR	Unconditional Jump to Subroutine	JSRR (Unconditional Jump to Subroutine)
LD	Load Register	LD (Load Register)
LDF	Load Register from Functional Unit	LDF (Load Register from Functional Unit)
LDI	Load Register with Immediate Value	LDI (Load Register with Immediate Value)
LDRPC	Load from RPC to Register	LDRPC (Load from RPC to Register)

Table continues on the next page...

Table 7-46. SDMA Instruction List (continued)

Instruction	Description	Page
LOOP	Hardware Loop	LOOP (Hardware Loop)
LSL1	Logical Shift Left by 1 Bit	LSL1 (Logical Shift Left by 1 Bit)
LSR1	Logical Shift Right by 1 Bit	LSR1 (Logical Shift Right by 1 Bit)
MOV	Logical Move	MOV (Logical Move)
NOTIFY	Notify to Arm platform	NOTIFY (Notify to Arm platform)
OR	Logical OR	OR (Logical OR)
ORI	Logical OR with Immediate Value	ORI (Logical OR with Immediate Value)
RET	Return from Subroutine	RET (Return from Subroutine)
REVB	Reverse Byte Order	REVB (Reverse Byte Order)
REVBLO	Reverse Low Order Bytes	Reverse Low Order Bytes(REVBLO)
ROR1	Rotate Right by 1 Bit	ROR1 (Rotate Right by 1 Bit)
RORB	Rotate Right by 1 Byte	RORB (Rotate Right by 1 Byte)
SOFTBKPT	Software Breakpoint	SOFTBKPT (Software Breakpoint)
ST	Store Register	ST (Store Register)
STF	Store Register in Functional Unit	STF (Store Register in Functional Unit)
SUB	Subtract	SUB (Subtract)
SUBI	Subtract with Immediate	SUBI (Subtract with Immediate)
TST	Test with Zero	TST (Test with Zero)
TSTI	Test Immediate	TSTI (Test Immediate)
XOR	Logical Exclusive OR	XOR (Logical Exclusive OR)
XORI	Exclusive OR with Immediate	XORI (Exclusive OR with Immediate)

7.2.4.2.1 ADD (Addition)

Operation:

```
GReg[r] ← GReg[s] + GReg[r]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: add r,s

Example: add 0,3
```

ADD GReg[3] and GReg[0] and store the result in GReg[0]

CPU Flags: T

Cycles: 1

Description: Performs the ADDition of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*. The T flag is set if the result of the operation is 0. It is cleared if the result is not 0.

Instruction Format:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.2 ADDI (Add with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] + immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: addi r,immediate

Example: add 6,112
```

ADD GReg[6] and decimal value 112 and store the result in GReg[6]

CPU Flags: T

Cycles: 1

Description: Adds a 0-extended immediate value to a general register; stores the result in the general register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

 ...

11111110 - 254

11111111 - 255
```

7.2.4.2.3 AND (Logical AND)

Operation:

```
GReg[r] ← GReg[s] & GReg[r]
```

Assembler:

```
Syntax: and r,s

Example: and 1,2
```

AND GReg[1] and GReg[2] and store the result in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.4 ANDI (Logical AND with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & immediate
```

Assembler:

```
Syntax: andi r,immediate

Example: andi 7,45
```

AND GReg[7] and decimal value 45 and store the result in GReg[7]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between a 0-extended immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.5 ANDN (Logical AND NOT)

Operation:

```
GReg[r] ← ~GReg[s] & GReg[r]
```

Assembler:

```
Syntax:andn r,s

Example: andn 3,4
```

AND GReg[3] and NOT GReg[4] (bit inverted) and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the AND of the negation of the source general register *s* and the destination general register *r*, and stores the result in the destination general register *r*.

Instruction Format:

Table 7-47. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	1	0	s	s	s

Instruction Fields:

rrr /sss - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.6 ANDNI (Logical AND with Negated Immediate Value)

Operation:

```
GReg[r] ← GReg[r] & ~immediate
```

Assembler:

```
Syntax: andni r,immediate

Example: andni 0,2
```

AND GReg[0] and decimal value -3 (inverted 32-bit value 2) and store the result in GReg[0]

CPU Flags: unaffected

Cycles: 1

Description: Performs an AND between the negation of a 0-extended 8-bit immediate value and a general register; stores the result in the general register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.7 ASR1 (Arithmetic Shift Right by 1 Bit)

Operation:

```
GReg[r]:{b31,b30,...,b1,b0} ← GReg[r]:{b31,b31,b30,...,b1}
```

Assembler:

```
Syntax: asr1 r

Example: asr1 3
```

divide by 2 the signed value of GReg[3] and store the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any general register to the right and keep the same sign: The left bit (bit 31) is kept untouched.

Instruction Format:

Table 7-48. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.8 BCLRI1 (Bit Clear Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),0,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bclri r,i

Example: bclri 1,12
```

clear bit 12 in GReg[1]

CPU Flags: Unaffected

Cycles: 1

Description: Clear the bit of register r specified by the 5-bit immediate field

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	1	i	i	i	i	i

rrr - register field:

```
000 - GReg[0]
```

```
001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - immediate value:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.9 BDF (Conditional Branch if Destination Fault)

Operation:

```
if (DF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax:bdf label

Example: bdf LLL
```

Jump to LLL if DF is set, or go to the next instruction if DF is cleared; the displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: If flag DF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag DF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	1	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.10 BF (Conditional Branch if False)

Operation:

```
if (T == 0)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler:

```
Syntax: bf label

Example: bf LLL
```

Jump to LLL if T is cleared, or go to the next instruction if T is set. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is cleared, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is set, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	0	p	p	p	p	p	p	p	p

Instruction Fields:

ppppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.11 BSETI (Bit Set Immediate)

Operation:

```
GReg[r]:{b31,...,b(i+1),1,b(i-1),...,b0} ← GReg[r]:{b31,...,b(i+1),b(i),b(i-1),...,b0}
```

Assembler:

```
Syntax: bseti r,i

Example: bseti 6,5
```

Set bit 5 in GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Sets bit number i in the selected General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	0	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
00000 - 0

00001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.12 BSF (Conditional Branch if Source Fault)

Operation:

```
if (SF == 1) PC ← PC + 1 + displacement else PC ← PC + 1
```

Assembler:

```
Syntax: bsf label

Example: bsf LLL
```

Jump to LLL if SF is set, or go to the next instruction if SF is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag SF is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag SF is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	1	0	p	p	p	p	p	p	p	p

Instruction Fields:

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110 - 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 - (-2)

11111111 - (-1)
```

7.2.4.2.13 BT (Conditional Branch if True)

Operation

```
if (T == 1)

PC ← PC + 1 + displacement

else

PC ← PC + 1
```

Assembler

```
Syntax: bt label

bt LLL
```

Jump to LLL if T is set, or go to the next instruction if T is cleared. The displacement value is calculated by the assembler.

CPU Flags: Unaffected

Cycles: 2 when the branch is done, 1 otherwise

Description: Conditional branch: If flag T is set, jump to the new address that is calculated by adding the sign-extended 8-bit displacement to the next PC address. If flag T is cleared, no jump is performed: The next instruction is located at the next PC address.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	1	0	1	p	p	p	p	p	p	p	p

pppppppp - signed displacement field:

```
00000000 - 0

00000001 - 1

...

01111110- 126

01111111 - 127

10000000 - (-128)

10000001 - (-127)

...

11111110 -  (-2)

11111111 - (-1)
```

7.2.4.2.14 BTSTI (Bit Test immediate)

Operation:

```
T ← GReg[r]:b(i)
```

Assembler:

```
Syntax: btsti r,i

Example: btsti 2,29
```

Test bit 29 in GReg[2] and copy its value in flag T

CPU flags: T

Cycles: 1

Description: T is loaded with the value of bit number i from the selected general register.

Instruction Format:

Table 7-49. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	1	1	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiii - bit number field:

```
0000 - 0

0001 - 1

...

11110 - 30

11111 - 31
```

7.2.4.2.15 CLRF (Clear Arm platform flags)

Operation:

```
 if (ff%2 == 0)

SF ← 0

if (ff/2 == 0)

DF ← 0
```

Assembler:

```
Syntax: clrf ff

Example: clrf 2
```

Clear flag SF and keep flag DF unchanged

CPU Flags: SF, DF

Cycles: 1

Description: Clears a selection of the Arm platform fault flags: SF, DF, both SF and DF or none can be cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	f	f	0	0	0	0	0	1	1	1

Instruction Fields:

ff - flags field:

```
00 - clear SF and clear DF

01 - clear DF

10 - clear

SF 11 - no clear
```

7.2.4.2.16 CMPEQ (Compare for Equal)

Operation:

```
T ← (GReg[s] == GReg[r])
```

Assembler:

```
Syntax: cmpeq r,s

Example: cmpeq 7,5
```

Compare GReg[7] and GReg[5] and set flag T if they are equal

CPU flags: T

Cycles: 1

Description: Subtracts the destination general register *r* from the source general register *s*, and sets T if the result is 0, clears T if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.17 CMPEQI (Compare with Immediate for Equal)

Operation:

```
T ← (GReg[r] == immediate)
```

Assembler:

```
Syntax: cmpeqi r,immediate

Example: cmpeqi 2,13
```

Compare GReg[2] and decimal value 13 and set flag T if they are equal

CPU Flags: T

Cycles: 1

Description: Subtracts the 0-extended 8-bit immediate value from the general register, and sets T if the result is 0, clears T if the result is not 0. The immediate value is the low-order byte of the instruction.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254
```

```
11111111 - 255
```

7.2.4.2.18 CMPHS (Compare for Higher or Same)

Operation:

```
T ← (GReg[r] ≥ GReg[s])
```

Assembler:

```
Syntax: cmphs r,s

Example: cmphs 0,1
```

Compare GReg[0] and GReg[1] and set flag T if GReg[0] is higher than or equal to GReg[1]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is higher than or equal to the source general register *s*, clears T otherwise. The comparison is unsigned.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.19 CMPLT (Compare for Less Than)

Operation:

```
T ← (GReg[r] < GReg[s])
```

Assembler:

```
Syntax: cmplt r,s

Example: cmplt 7,4
```

Compare GReg[7] and GReg[4] and set flag T if GReg[7] is lower than GReg[4]

CPU Flags: T

Cycles: 1

Description: Compares the destination general register *r* and the source general register *s*, and sets T if the destination general register *r* is lower than the source general register *s*, clears T otherwise. The comparison is signed.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	1	0	s	s	s

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.20 cpShReg (Update Context of PCU Registers and Flag)

Assembler:

```
Syntax: cpShReg
```

CPU Flags: none

Cycles: 1

Description: SF, RPC, T, PC,LM, EPC, DF, and SPC registers are updated according to the value of their corresponding bits in the context memory. This instruction must only be used in debug mode via the OnCE. It reverses the done *5* operation.

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	0	1	1	1	0	0	0	1	0

7.2.4.2.21 DONE (DONE, Yield)

Operation:

```
if (jjj&6 == 2) HE[CCR] ← 0

if (jjj == 3) HI[CCR] ← 1

if (jjj == 4) EP[CCR] ← 0


if ((jjj == 0) && (NCP > CCP)) CCR ← NCR

else if ((jjj == 1) && (NCP >= CCP))

CCR ← NCR

else

CCR ← NCR
```

(CCR stands for Current Channel Register; NCR stands for Next Channel Register)

Assembler:

```
Syntax: done jjj

Example: done 3
```

Clear HE bit for the current channel, send an interrupt to the Arm platform for the current channel and reschedule.

CPU Flags: Unaffected

Cycles: Variable if a context switch is done, 1 otherwise

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required. Sends an interrupt to the corresponding Arm platform by setting the appropriate flag, if required (HI for the corresponding channel number). Reschedules according to the mode and the NCP (Next Channel Priority) and CCP (Current Channel Priority) values. According to the scheduling decision, the NCR (Next Channel Register) is copied to the CCR (Current Channel Register) and channel contexts are switched. If several channels with the same highest priority are pending, they are ordered by their number from 31 down to 0. The higher number is selected (for example, channel 26 is selected if channels 3, 12, 14, and 26 with the same highest priority are pending). If no flag is modified, the reschedule can allow the replacement of the current

channel by another channel with a priority strictly greater than the current channel priority (yield). Or, it can allow the replacement of the current channel by another channel with a priority greater than or equal to the current channel priority (yieldge). In the latter case, the selected channel will always be the first one with the same priority, starting from channel number 31 down to channel 0 (the current channel does not belong to the set of selectable channels).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	0

jjj - Channel Flags field:

000 - No channel flags affected: Reschedule only if the next channel priority is greater than current channel priority (yield)

001 - No channel flags affected: Reschedule only if the next channel priority is greater than or equal to the current channel priority (yieldge)

010 - Clear HE for the current channel and reschedule 011 - Clear HE, set HI for the current channel and reschedule 100 - Clear EP for the current channel and reschedule

101 - Reserved for debug to copy relevant registers into context memory

110 - RESERVED

111 - RESERVED

For the scheduling rules, refer to Scheduler Functional Description. Every possible done instruction is further described as follows:

- done 0/yield is executed by a channel script when it accepts preemption by a higher priority channel;
- done 1/yieldge is executed by a channel script when it accepts preemption by a higher priority channel and it also accepts a roll-up with other channels that have the same priority;
- done 2 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed and it requires termination;
- done 3 is executed by a channel script that was triggered by a Arm platform start via the Channel Start (SDMAARM_HSTART) register, when its task is completed, it requires termination and it needs to trigger an interrupt to the Arm platform upon closure;

- done 4 is executed by a channel script that was triggered by a DMA request, when its task is completed and it requires termination;
- done 5 is used in debug mode only; it copies the PCU registers and flags to the context memory of the current channel;

7.2.4.2.22 ILLEGAL (ILLEGAL Instruction)

Operation:

```
PC ← 0001
```

Assembler:

```
Syntax: illegal
```

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the Illegal instruction routine located at address 0001. All unauthorized instructions result in an Illegal instruction behavior; however, the ILLEGAL instruction must be used to guarantee software compatibility with future versions of the SDMA.

Instruction Format

Table 7-50. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	1	1	1	0	0	0	0	0	1	1	1

7.2.4.2.23 JMP (Unconditional Jump Immediate)

Operation:

```
PC ← absolute_address
```

Assembler:

```
Syntax: jmp label

Example: jmp LLL
```

The assembler translates the label to the exact address

CPU Flags:Unaffected

Cycles: 2

Description: Jumps to the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

7.2.4.2.24 JMPR (Unconditional Jump)

Operation:

```
PC ← GReg[r]
```

Assembler:

```
Syntax: jmpr r

Example: jmpr 0
```

Jump to address stored in GReg[0]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the absolute address contained in a General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.25 JSR (Unconditional Jump to Subroutine Immediate)

Operation:

```
RPC ← PC + 1

PC ← absolute_address
```

Assembler:

```
Syntax: jsr r

Example:jsr LLL
```

Jumps to subroutine starting at LLL; the assembler translates the label to exact address

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine located at the absolute address contained the lower 14 bits of the instruction (the PC is a 14-bit register).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	a	a	a	a	a	a	a	a	a	a	a	a	a	a

aaaaaaaaaaaaaa - address field:

```
00000000000000 - 0

00000000000001 - 1

...

11111111111110 - 16382

11111111111111 - 16383
```

7.2.4.2.26 JSRR (Unconditional Jump to Subroutine)

Operation:

```
RPC ← PC + 1

PC ← GReg[r]
```

Assembler:

```
Syntax: jsrr r

Example:jsrr 5
```

Jumps to subroutine located at address stored in GReg[5]

CPU Flags: Unaffected

Cycles: 2

Description: Jumps to the subroutine at address contained in a General Register

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.27 LD (Load Register)

Operation:

```
GReg[r] ← [GReg[b] + displacement]

if (transfer_error)

SF ← 1

else

SF ← 0
```

Assembler:

```
Syntax: ld r,(b,displacement)

Example: ld 1,(2,23)
```

Loads data into GReg[1]; the data is located at address obtained by adding decimal value 23 to GReg[2]

CPU Flags: SF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to fetch on the DM bus. The data received from the bus is stored in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	0	r	r	r	d	d	d	d	d	b	b	b

rrr / bbb - register field:

```
 000 - GReg[0]

 001 - GReg[1]

 ...

 111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0

00001 - 1

...

11111 - 31
```

7.2.4.2.28 LDF (Load Register from Functional Unit)

Operation:

```
GReg[r] ← [fu_address]

if (transfer_error)

SF ← 1
```

```
else

SF ←0
```

fu_address is an 8-bit field and depends on addressed functional unit

Assembler:

```
Syntax: ldf r,fu_address

Example: ldf 0,13
```

Loads data coming from the Burst DMA register MD into GReg[0]; it is a 32-bit access with no prefetch

CPU Flags: SF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and stores the data received from the bus in the destination General Register r. If an error occurs during the transfer, the flag SF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	0	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the LDF instruction usage with each functional unit:

- Burst DMA Read (ldf) for Burst DMA
- Peripheral DMA Read (ldf)-Read Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit source register and action (unspecified values are reserved):

```
00000000 - MSA

00000100 - MDA

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - MS

00101001 - MD byte - prefetch

00101010 - MD halfword - prefetch

00101011 - MD word - prefetch

01000000 - DSA



11000000 - PSA

11001000 - PD

11010000 - PDA

11011000 - PD in copy mode (rrr contents are lost)

11101000 - PD - prefetch next data

11111111 - PS
```

7.2.4.2.29 LDI (Load Register with Immediate Value)

Operation:

```
GReg[r] ← immediate
```

Assembler:

```
Syntax: ldi r,immediate

Example: ldi 6,1
```

loads decimal value 1 into GReg[6]

CPU Flags: Unaffected

Cycles: 1

Description: Stores a 0-extended immediate value in a General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.30 LDRPC (Load from RPC to Register)

Operation:

```
GReg[r] ← RPC
```

Assembler:

```
Syntax: ldrpc r

Example: ldrpc 3
```

copies RPC to GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Stores the contents of the RPC in a General Register. That instruction may be used to have more than one level of subroutines.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	0	1	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.31 LOOP (Hardware Loop)

Operation:

```
if (ff%2 == 0)
   SF ← 0
if (ff/2 == 0)
  DF ← 0
if ((GReg[0] == 0) || (SF == 1) || (DF == 1))
   PC ← PC + loop_size + 1
else
   {
      SPC ← PC + 1
      EPC ← PC + loop_size + 1
      LM ← 1
      PC ← PC + 1
   }
```

during every instruction execution in the loop:

```
if ((SF == 1) || (DF == 1))
 {
```

```
    LM ← 0
    PC ← EPC
  }
else if ((PC + 1) == EPC)
 {
    GReg[0] ← GReg[0] - 1
    if (GReg[0] == 0)
      {
         LM ← 0
         PC ← EPC
       }
     else
         PC ← SPC
   }
else
  PC ← nextPC(instruction)
```

after the execution of the last instruction of the loop body:

```
if (GReg[0] == 0)
   T ← 1
else
   T ← 0
```

Assembler:

```
Syntax: loop n{,ff}
Example: loop 3,1
```

Executes GReg[0] times the instructions comprised between PC+1 and PC+3 (included); ff=1 clears the DF flag before starting the loop. When omitted, the ff field is set to 0 (clearing both SF and DF).

CPU Flags: LM[1:0], T

Cycles: 2 when the loop count (GReg[0]) is 0 or SF or DF is set at loop start, 1+1 when the loop starts but exits abnormally (SF or DF set inside the loop which adds 1 cycle to the offending load or store to jump to EPC), 1 when the loop is executed normally

Description: The loop instruction executes a sequence of instructions several times. The number of times is given by the contents of GReg[0], the loop counter. SDMA will jump to the first instruction after the end of the loop if the value in GReg[0] is 0. Otherwise the SDMA enters loop mode. It sets the most significant bit of the LM flag that will only be reset once the last instruction of the last loop is executed. The instructions in the loop are executed GReg[0] times. The management of fault flags (SF and DF) is as follows. When entering the hardware loop, SF and DF can be cleared according to the ff field of the instruction. After that operation, if any flag is still set the loop will not be executed. The SDMA will jump to the first instruction after the end of the loop without entering loop mode. During the execution of the loop, if any fault flag is set by a LD, LDF, ST, or STF instruction, the SDMA will immediately exit loop mode and jump to the first instruction after the end of the loop. In that case, GReg0 is not decremented for that last piece of the loop body execution (even if the SF or DF flag is set at the last instruction of the loop body). The T flag reflects the state of GReg[0] after the end of the loop, which is an indicator of the complete execution of the loop. If the loop exited because of an error (SF or DF set), GReg[0] will not be 0 at the end of the loop, hence T will be cleared. If the loop executes without fault, GReg[0] will be 0 at the end of the loop, hence T will be set. The boundary case when a source or destination fault occurs at the last instruction of the last loop is considered as an anticipated exit of the loop, which causes the T flag to be cleared. If the last instruction executed before leaving the hardware loop also tries to modify the T flag, the flag is updated according to the value of GReg[0], NOT according to the result of the last executed instruction.

Limitations:

1. 1. Jump instructions (JMP, JMPR, JSR, JSRR, BF, BT, BSF, BDF) are not allowed inside the hardware loop.
2. 2. GReg[0] cannot be written to inside the hardware loop (it can be read).
3. 3. The empty loop (0 instruction in the body) is forbidden.
4. 4. If GReg[0] == 0 at the start of the loop, which causes a jump to EPC, the T flag is not updated.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	1	1	0	f	f	n	n	n	n	n	n	n	n

Instruction Fields:

```
ff - flags field:

00 - clear SF and clear DF

01 - clear DF

10 - clear SF
```

```
11 - no clear
```

nnnnnnnn - loop size

```
00000000 - empty loop: forbidden value

00000001 - 1 instruction in the loop

00000010 - 2 instructions in the loop

...

11111111 - 255 instructions in the loop
```

7.2.4.2.32 LSL1 (Logical Shift Left by 1 Bit)

Operation:

```
GReg[r]:{b30,...,b1,b0,0} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsl1 r

Example: lsl1 2
```

multiplies by 2 the value in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the left. The right bit (bit 0) is set to 0. No overflow is detected by the hardware.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	1	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]
```

```
111 - GReg[7]
```

7.2.4.2.33 LSR1 (Logical Shift Right by 1 Bit)

Operation:

```
GReg[r]:{0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: lsr1 r

Example: lsr1 4
```

divides by 2 the unsigned value contained in GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Shift the bits of any General Register to the right. The left bit (bit 31) is set to 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	1

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.34 MOV (Logical Move)

Operation:

```
GReg[r] ← GReg[s]
```

Assembler:

```
Syntax: mov r,s

Example: mov 4,0
```

copies GReg[0] to GReg[4]

CPU Flags: Unaffected

Cycles: 1

Description: Move the contents of the source General Register s to the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.35 NOTIFY (Notify to Arm platform)

Operation:

```
 if (jjj & 4 == 0)

 {

   if (jjj&2 == 2)

      HE[CCR] ← 0

   if (jjj&1== 1)

       HI[CCR] ← 1

  }

else if (jjj == 4)
```

```
   EP[CCR] ← 0

else
```

(CCR stands for Current Channel Register)

Assembler:

```
Syntax: notify jjj

Example: notify 3
```

clears the HE bit for the current channel and sends an interrupt to the Host for the current channel

CPU Flags: Unaffected

Cycles: 1

Description: Clears one of the channel enabling bits (HE or EP for the corresponding channel number) if required, sends an interrupt to the corresponding Arm platform by setting the appropriate flag if required (HI for the corresponding channel number).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	j	j	j	0	0	0	0	0	0	0	1

jjj - Channel Flags field:

```
000 - unused

001 - set HI for the current channel

010 - clear HE for the current channel

011 - clear HE, set HI for the current channel

100 - clear EP for the current channel

101 - RESERVED

110 - RESERVED

111 - RESERVED
```

7.2.4.2.36 OR (Logical OR)

Operation:

```
GReg[r] ← GReg[s] | GReg[r]
```

Assembler:

```
Syntax: or r,s
```

```
Example: or 3,6
```

ORs GReg[3] and GReg[6] and stores the result in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	1	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.37 ORI (Logical OR with Immediate Value)

Operation:

```
GReg[r] ← GReg[r] | immediate
```

Assembler:

```
Syntax: ori r,immediate

Example: ori 1,56
```

ORs GReg[1] and the decimal value 56 and stores the result in GReg[1]

CPU Flags: unaffected

Cycles: 1

Description: Performs an OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	1	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.38 RET (Return from Subroutine)

Operation:

```
PC ← RPC
```

Assembler:

```
Syntax: ret
```

CPU Flags: Unaffected

Cycles: 2

Description: Return from subroutine.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

7.2.4.2.39 REVB (Reverse Byte Order)

Operation:

```
GReg[r]:{B3,B2,B1,B0} ← GReg[r]:{B0,B1,B2,B3}
```

Assembler:

```
Syntax: revb r

Example:  revb 5
```

reverses bytes order in GReg[5]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse the byte order of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.40 Reverse Low Order Bytes(REVBLO)

Operation:

```
GReg[r]:{B3,B2,B0,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: revblo r

Example: revblo 0
```

reverses low order bytes in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Reverse both low order bytes of any General Register.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	0	1

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.41 ROR1 (Rotate Right by 1 Bit)

Operation:

```
GReg[r]:{b0,b31,b30,...,b1} ← GReg[r]:{b31,b30,...,b1,b0}
```

Assembler:

```
Syntax: ror1 r

Example: ror1 3
```

rotates bits to the right in GReg[3]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bits of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	1	0	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.42 RORB (Rotate Right by 1 Byte)

Operation:

```
GReg[r]:{B0,B3,B2,B1} ← GReg[r]:{B3,B2,B1,B0}
```

Assembler:

```
Syntax: rorb r

Example: rorb 2
```

rotates bytes to the right in GReg[2]

CPU Flags: Unaffected

Cycles: 1

Description: Rotate the bytes of any General Register to the right.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	0	0	0	1	0	0	1	0

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.43 SOFTBKPT (Software Breakpoint)

Operation:

Stops the current script and enters debug mode

Assembler:

```
softbkpt
```

CPU Flags: Unaffected

Description: When the core executes this instruction, it has the same effect as receiving a debug request from the OnCE or via the external debug request input: the script execution halts, the PCU enters its debug state and waits for the OnCE commands that are described in OnCE and Real-Time Debug.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

7.2.4.2.44 ST (Store Register)

Operation:

```
[GReg[b] + displacement] ← GReg[r]
```

```
if (transfer_error)

 DF ← 1

 else

DF ← 0
```

Assembler:

```
Syntax: st r,(b,displacement)

Example: st 7,(0,9)
```

stores the value from GReg[7] into memory at address obtained by adding decimal value 9 to GReg[0]

CPU Flags: DF

Cycles: 2+n where n is 0 for ROM, RAM or memory mapped registers, and n is the number of wait-states of the peripheral for a peripheral access

Description: Adds a 5-bit 0-extended displacement to a base address in General Register b; the result is the address of the data to store on the DM bus. The data sent on the bus comes from the source General Register r. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	1	1	r	r	r	d	d	d	d	d	b	b	b

Instruction Fields:

rrr / bbb - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ddddd - displacement value:

```
00000 - 0
```

```
00001 - 1

...

11111 - 31
```

7.2.4.2.45 STF (Store Register in Functional Unit)

Operation:

```
[fu_address] ← GReg[r] 0

if (transfer_error) 0

DF ← 1 0

else 0

DF ← 0
```

fu_address is an 8-bit field

Assembler:

```
Syntax: stf r,fu_address

Example: stf 3,0x2B
```

stores the 32-bit contents of GReg[3] to the Burst DMA register MD; waits until the flush to external memory is completed

CPU Flags: DF

Cycles: 1+n where n is the number of wait-states that may be inserted by the functional unit

Description: Sends an 8-bit address on the Functional Unit Bus (FU bus) and sends the contents of the source General Register r on the bus. If an error occurs during the transfer, the flag DF is set, else it is cleared.

Table 7-51. Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	1	0	1	r	r	r	f	f	f	f	f	f	f	f

See the following sections for more details of the STF instruction usage with each functional unit:

- Burst DMA Write (stf) for Burst DMA
- Peripheral DMA Write (stf)-Write Mode for Peripheral DMA

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

ffffffff - functional unit destination register and action (unspecified values are reserved):

00000000 - MSA in incremented mode

00000100 - MDA in incremented mode

00001001 - MD byte

00001010 - MD halfword

00001011 - MD word

00001100 - clear MS error flag

00001111 - MS

00010000 - MSA in frozen mode

00010100 - MDA in frozen mode

00011000 - MD in copy mode - number of words in rrr

00100000 - MSA in incremented mode - start prefetch

00101000 - MD no data - flush

00101001 - MD byte - flush

00101010 - MD halfword - flush

00101011 - MD word - flush

00110000 - MSA in frozen mode - start prefetch

11000001 - PSA in frozen mode - 8-bit data width

11000010 - PSA in frozen mode - 16-bit data width

11000011 - PSA in frozen mode - 32-bit data width

11000101 - PSA in incremented mode - 8-bit data width

11000110 - PSA in incremented mode - 16-bit data width

11000111 - PSA in incremented mode - 32-bit data width

11001000 - PD

11001001 - PSA in decremented mode - 8-bit data width

11001010 - PSA in decremented mode - 16-bit data width

11001011 - PSA in decremented mode - 32-bit data width

11001100 - clear PS error flag

11001101 - PSA data width becomes 8-bit

11001110 - PSA data width becomes 16-bit

11001111 - PSA data width becomes 32-bit

11010001 - PDA in frozen mode - 8-bit data width

11010010 - PDA in frozen mode - 16-bit data width

11010011 - PDA in frozen mode - 32-bit data width

11010101 - PDA in incremented mode - 8-bit data width

11010110 - PDA in incremented mode - 16-bit data width

11010111 - PDA in incremented mode - 32-bit data width

11011001 - PDA in decremented mode - 8-bit data width

11011010 - PDA in decremented mode - 16-bit data width

11011011 - PDA in decremented mode - 32-bit data width

11011101 - PDA data width becomes 8-bit

11011110 - PDA data width becomes 16-bit

11011111 - PDA data width becomes 32-bit

11100001 - PSA in frozen mode - 8-bit data width - prefetch data

11100010 - PSA in frozen mode - 16-bit data width - prefetch data

11100011 - PSA in frozen mode - 32-bit data width - prefetch data

11100101 - PSA in incremented mode - 8-bit data width - prefetch data

11100110 - PSA in incremented mode - 16-bit data width - prefetch data

11100111 - PSA in incremented mode - 32-bit data width - prefetch data

11101001 - PSA in decremented mode - 8-bit data width - prefetch data

11101010 - PSA in decremented mode - 16-bit data width - prefetch data

11101011 - PSA in decremented mode - 32-bit data width - prefetch data

11101101 - PSA data width becomes 8-bit - prefetch data

11101110 - PSA data width becomes 16-bit - prefetch data

11101111 - PSA data width becomes 32-bit - prefetch data

11111111- PS

7.2.4.2.46 SUB (Subtract)

Operation:

```
GReg[r] ← GReg[r] - GReg[s]

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,s

Example: sub 4,7
```

SUBtracts GReg[7] from GReg[4] and stores the result in GReg[4]

CPU Flags: T

Cycles: 1

Description: Subtracts the source General Register s from the destination General Register r, and stores the result in the destination General Register r. The T flag is set if the result of the operation is 0; it is cleared if the result is not 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	1	0	0	s	s	s

Instruction Fields:

rrr / sss - register fields:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.47 SUBI (Subtract with Immediate)

Operation:

```
GReg[r] ← GReg[r] - immediate

T ← (GReg[r] == 0)
```

Assembler:

```
Syntax: sub r,immediate

Example: sub 1,255
```

SUBtracts decimal value 255 from GReg[1] and stores the result in GReg[1]

CPU Flags: T

Cycles: 1

Description: Subtracts a 0-extended 8-bit immediate value from a General Register; stores the result in the General Register. The flag T is set when the result of the operation is 0; otherwise, it is cleared. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	1	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.48 TST (Test with Zero)

Operation:

```
T ← ((GReg[s] & GReg[r]) != 0)
```

Assembler:

```
Syntax: tst r,s

Example: tst 2,3
```

ANDs GReg[2] and GReg[3] and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of the source General Register s and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	1	0	0	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.49 TSTI (Test Immediate)

Operation:

```
T ← ((GReg[r] & immediate) != 0)
```

Assembler:

```
Syntax: tsti r,immediate

Example: tsti 5,13
```

ANDs GReg[5] and decimal value 13 and sets T if the result is non-null

CPU Flags: T

Cycles: 1

Description: Performs the AND of a 0-extended 8-bit immediate value and the destination General Register r, and sets T if the result is not 0, clears T if the result is 0. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	0	0	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - destination register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.50 XOR (Logical Exclusive OR)

Operation:

```
GReg[r] ← GReg[s] ^ GReg[r]
```

Assembler:

```
Syntax: xor r,s

Example: xor 0,3
```

XORs GReg[0] and GReg[3] and stores the result in GReg[0]

CPU Flags: Unaffected

Cycles: 1

Description: Performs the eXclusive OR of the source General Register s and the destination General Register r, and stores the result in the destination General Register r.

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	0	r	r	r	1	0	0	1	0	s	s	s

Instruction Fields:

rrr / sss - register field:

```
000 - GReg[0]

001 - GReg[1]
```

```
010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]

111 - GReg[7]
```

7.2.4.2.51 XORI (Exclusive OR with Immediate)

Operation:

```
GReg[r] ← GReg[r] ^ immediate
```

Assembler:

```
Syntax: xori r,immediate

Example: xor 7,5
```

XORs GReg[5] and decimal value 5 and stores the result in GReg[7]

CPU Flags: Unaffected

Cycles: 1

Description: Performs an eXclusive OR between a 0-extended 8-bit immediate value and a General Register; stores the result in the General Register. The immediate value is the low-order byte of the instruction and has a maximum value of 255 (0xFF).

Instruction Format

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	1	0	r	r	r	i	i	i	i	i	i	i	i

Instruction Fields:

rrr - register field:

```
000 - GReg[0]

001 - GReg[1]

010 - GReg[2]

011 - GReg[3]

100 - GReg[4]

101 - GReg[5]

110 - GReg[6]
```

```
111 - GReg[7]
```

iiiiiiii - immediate value:

```
00000000 - 0

00000001 - 1

...

11111110 - 254

11111111 - 255
```

7.2.4.2.52 YIELD, YIELDGE (DONE, Yield)

By default, unsupported assembler syntax. Can be aliased to the corresponding done instructions (yield = done 0; yieldge = done 1). Refer to the done instruction description DONE (DONE, Yield) .

7.2.5 Software Restrictions

7.2.5.1 Unsupported Burst DMA Access Sequence

The SDMA does not support triggering a pre-fetch followed by a flush of the Burst DMA without reading or writing any data. If the flush occurs while the background pre-fetch DMA operation is still in progress, it could result in un-defined behavior.

An example of the sequence which could result in undefined results is shown in the following example:

Instruction sequence not supported

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch in the
                        ; background
mov R0,R0               ; Execute multiple assembly instructions, none of which
                        ; read
mov R0,R0               ; or write data to/from MD
stf MD|SZ0|FL           ; Flush FIFO without writing data. If the pre-fetch is still
                        ; in progress when this instruction is executed, there
                        ; could be undefined operation
```

A work-around to avoid any undesirable results is to first read MD to ensure the pre-fetch is complete before the flush is attempted.

Work-Around to previous example

```
stf r1, MSA|PF          ; Update source address, triggers data pre-fetch.
```

```
mov R0,R0              ; Execute multiple assembly instructions, none of which
                       ; read
mov R0,R0              ; or write data to/from MD
ldf r2, MD             ; dummy read of MD to ensure pre-fetch is complete
                       ; before the next instruction
stf MD|SZ0|FL          ; Flush FIFO without writing data
```

7.2.6 Application Notes

7.2.6.1 Data Structures for Boot Code and Channel Scripts

SDMA boot code downloads the different channel contexts and the scripts that will be executed on SDMA channels during the application.

The boot code is run after reset when channel 0 is started by the Arm platform. The boot code is also known as channel 0 script.

The boot code is based on the Channel Control Block (CCB) and Buffer Descriptor (BD) mechanisms that are data structures located into the Arm platform memory space. With these data structures, it is possible to instruct SDMA to download scripts and contexts but also to dump a context or a script to a destination data buffer. Channel scripts also use the CCB and BD data structures to pass instructions and/or pointers to data to be copied.

The format, processing, and field definition of the CCB and BD are defined and performed entirely by the software script rather than the SDMA hardware. An overview of the format and structure is provided here, but for complete details refer to the SDMA software documentation (see SDMA Scripts).

The CCB and BD data structures are accessed by SDMA using DMA and processed by the SDMA scripts. The ROM contains common sub-routines for processing these data structures which may be called by the bootload and channel scripts.

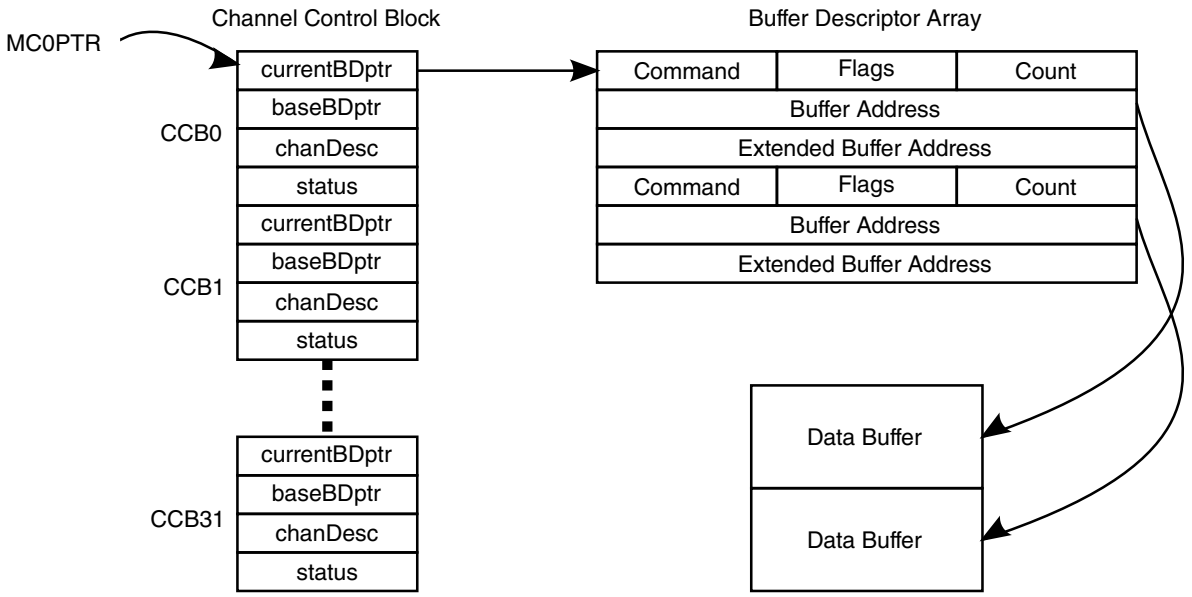


Figure 7-16. Data Structures Layout

The previous figure shows an example how these data structures are linked to pass command and pointers to data buffers. The SDMA's MC0PTR register holds the base address of the Channel 0 Control Block (CCB0). The Channel 0 control block holds a pointer to the array of buffer descriptors. The buffer descriptors are used to tell the channel 0 (boot channel) what to do as described Buffer Descriptor Format.

7.2.6.1.1 Buffer Descriptor Format

Buffer descriptors are three longs (32-bit words) in size as, shown in the figure found here.

A buffer descriptor describes the properties of the data buffer it points to. The buffer descriptors can be used for linear or circular data buffers in the Arm platform processor memory. The CCB contains a pointer to the base BD as well as the current BD.

Table 7-52. Buffer Descriptor

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Command								-	-	L	R	I	C	W	D	Count															
Buffer Address																															
Extended Buffer Address																															

Table 7-53. Buffer Descriptor Field Descriptions

Field	Description
31-24 Command	Command. The command field is used to differentiate operations performed within a script when the script accesses this particular buffer descriptor. The use of this field can be defined by the script. The command values defined for the bootload script are defined in Buffer Descriptor Commands for Bootload scripts. Refer to the individual script definition in script library documents in SDMA Scripts for command field definitions for other scripts.
23	Reserved
22	Reserved
21 L	Last Buffer Descriptor: This bit is set in SDMA IPC scripts to indicate to the receiving Core that the transfer has ended. Whenever the source finishes transferring the count it wanted to transfer, it sets LAST_BIT in the destination BD, to let the destination know that transfer is over.This bit also tells the destination software that when it processes the destination BDs, they need not process any BD after the BD with the LAST_BIT set.For example, when the DSP prepares a single buffer descriptor with count equals to 25 and Arm platform prepares a single buffer descriptor with count equals 100. When 25 bytes have been transferred from DSP to Arm platform, the DSP buffer descriptor is normally closed while the Arm platform buffer descriptor will have the L bit set and the byte count updated to 25.
20 R	erroR. Indicates an error occurred on the channel's buffer descriptor requested command. Some scripts may overwrite the command field with an error code indicating the source of the error. 0 No Error 1 Error
19 I	Interrupt. When SDMA has finished to process data transfer attached to this buffer descriptor, send an interrupt to the Arm platform. 0 No Interrupt 1 Interrupt the processor when BD is complete
18 C	Continuous. This buffer is allowed to receive multiple transmit buffers or is allowed to transmit to multiple receive buffers.The Continuous bit is decoded at the end of the processing of a BD to determine if the SDMA script must open a new BD to potentially continue the data transfer. 0 No further buffer descriptors 1 SDMA should move to the next Buffer descriptor after this one
17 W	Wrap. Indicates if this buffer descriptor is the last one for the channel control block. When encountering this bit set, the SDMA scripts updates the CurrentBD pointer to point to the first Buffer Descriptor of the array. This bit is set if the Arm platform wants to organize the array of BD in a circular way (like a ring). When all BD have been processed and if Wrap bit and CONtinuous bit are set in the last BD, the SDMA script will wrap around and it will try to re-open the first BD. 0 No Error 1 Wrap to first buffer descriptor after this one is processed.
16 D	D - "Done": bit 16: indicates the "ownership" of the buffer descriptor. When D=0 the host owns the buffer descriptor; when D=1 SDMA owns the buffer descriptor. In the case of the channel 0, D=1 indicates the SDMA has not yet processed this buffer, D=0 indicates the SDMA has processed this buffer. 0 Arm platform owns the buffer. 1 SDMA owns the buffer
15-0 Count	Count. the count field (bit 15-0) indicates the size of the data to be transmitted, the size of the data buffer pointed to by the buffer descriptor. The SDMA memory structure is different for program memory (16-bits shorts/half-words) and data memory (32-bits long). For channel 0 buffer descriptors, Count is expressed in 16-bit half-words when PM is addressed and in 32-bit words when DM is addressed. Count is typically expressed in bytes for other channel scripts, but the unit is dependant on the script.
31-0	Buffer address. Address pointer to the data buffer.
31-0	Extended buffer address. Additional pointer or other information required by some scripts.

The buffer descriptors form an array of programmable size. If the last buffer descriptor is marked by the Wrap flag-bit W=1, the array of buffer descriptor is treated as a ring with some logically continuous portion owned by the Arm platform with D=0, and the remainder owned by the SDMA with D=1. The count field of the buffer descriptor indicates how much data has been transmitted.

If Arm platform has prepared 3 buffers to be filled by the SDMA script, it has also prepared 3 BD, one for each buffer. The *Cont* and *Wrap* bits are used to organize the buffers in a circular way. For example, *CONT*inous bit is set to 1 in the 2 first BDs and *Wrap* is set in the 3rd BD. The SDMA script opens and processes BD#1. Since *CONT*inous bit is set for this BD, the SDMA will open the second BD and it will process it. Each time a BD is processed, its *Done* bit is reset by the SDMA. After the 3rd BD, if *CONT*inous is not set but if *Wrap* is set, the SDMA script stops here and the next time the channel will be triggered, the script will open the BD pointed by the currentBDptr pointer of the CCB and it will correspond to the first buffer descriptor.

If the *CONT*inous bit and *Wrap* bits are both set in the 3rd BD, the script will close it and it will try to open the first BD. An error may occur at this point if the BD#1 has already been processed and its *Done* bit is 0. The SDMA script cannot process a BD with a *Done* bit to 0. It means the BD is not ready to be processed. To avoid this situation, the *CONT*inous bit should not be set for the last BD if *Wrap* is set, and the Interrupt flag must set for the last BD. It will warn the owner of the BD that all the BDs have been processed and it has to re-set to 1 the *Done* bit of all the BD's if it desires the SDMA to fill them again. Basically, if the Arm platform expects the SDMA to fill up the buffers in a circular fashion, then it's the responsibility of the Arm platform to set the *Done* bit of a buffer descriptor at an appropriate time.

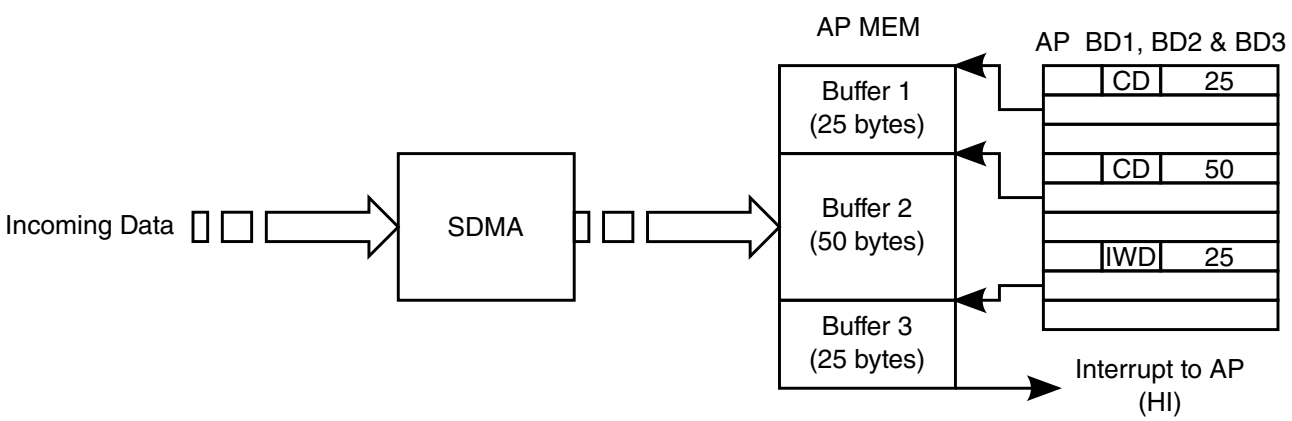


Figure 7-17. Buffer Descriptor Flow

The previous figure shows an example buffer descriptor flow. When the incoming data is stored and fills the first buffer of 25 bytes, the SDMA script opens the second BD because the CONTinuous bit was set. Then next incoming data is put in the second buffer. After receiving 50 bytes, the second buffer descriptor is also closed. The Done bit is reset and the third BD is opened. After receiving another 25 bytes, the third buffer is full and an interrupt is sent to the Arm platform because the Interrupt flag is set in the 3rd BD. The CONTinuous flag is not present the transfer is over. The next time the script will be triggered, the BD to be opened will be the first buffer descriptor since the Wrap flag was set in the 3rd BD. It is the Arm platform responsibility to set the Done bit of all the BD if it wants to use the same buffers.

7.2.6.1.2 Buffer Descriptor Commands for Bootload scripts

The command field of the buffer descriptor is defined separately for each script.

The following table lists the buffer descriptor commands defined for the channel 0 bootload script.

Table 7-54. Channel Zero Buffer Descriptor Commands

Command Field (binary)	Command	Description	Buffer Address	Extended Buffer Address
0000_0001 (0x01)	C0_SET_DM	Load SDMA data memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0010 (0x02)	C0_GET_DM	Copy SDMA data memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
0000_0100 (0x04)	C0_SET_PM	Load SDMA program memory (RAM) from Arm platform memory buffer	Arm platform memory source address	SDMA memory destination address
0000_0110 (0x06)	C0_GET_PM	Copy SDMA program memory (RAM) to Arm platform memory buffer	Arm platform memory destination address	SDMA memory source address
cccc_c111 (0x07 | CHN)	C0_SETCTX	Load Context for channel cccc into SDMA RAM from Arm platform memory buffer	Arm Platform memory source address	-
cccc_c011 (0x03 | CHN)	C0_GETCTXT	Copy Context for channel ccccc from SDMA RAM to Arm platform memory buffer	Arm platform memory destination address	-

The Channel 0 bootload commands are summarized as follows:

- C0_SET_[PM-DM]: load the buffer descriptor data in the SDMA local memory at the address pointed to by the "extended buffer address" field. The SDMA RAM can be seen as a Program Memory (PM, 16-bit address) or Data Memory (DM 32-bit address). When C0_SET_PM is used, the count field is expressed in "shorts" (16-bit half words), this command can be used to download scripts. When C0_SET_DM is

used, the count field is expressed in "long" (32-bit words), this command can be used to download channel contexts to the context channel area in RAM.

- C0_GET_[PM-DM]: write to the buffer descriptor's data buffer the content of the SDMA local memory from the address pointed to by the "extended buffer address" field for the length defined by the count in the buffer descriptor. C0_GET_PM is used to dump some part of the Program Memory (may be used to dump context of a channel), therefore count is expressed in "shorts"; while C0_GET_DM is used to dump to the buffer descriptor's data buffer, so the count field is in "longs."
- C0_SETCTX: load a context into the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using the channel number the script computes the offset of the context data pointer for the channel relative to the context page base to use as the destination address in SDMA memory. Then the C0_SET_DM command explained above is invoked to load SDMA RAM from memory. The counter indicates the size in words of the context structure.
- Command value: (in binary) cccc c111, where ccccc is the channel number (5 bits). For instance, 0x0F means set context for channel 1, 0xFF means set context for channel 31.

- C0_GETCTX: write to the buffer descriptor's data buffer the content of the SDMA context page area. The handling script decodes the channel number from the 5 MSB of the command field of the buffer descriptor. Using this channel number, the script computes the offset of the context data pointer for the channel relative to the context page base to use as the source address for the copy. Then the C0_GET_DM command explained above is invoked to copy the context to memory. The counter indicates the size in words of the context structure.
- Command value: (in binary): cccc c011, where ccccc is the channel number (5 bits). For instance, 0x03 means get context of channel 1, 0xFB means get context of channel 31.

NOTE

To download channel context, C0_SETDM and C0_SETCTXT command can be used but the second one is easier because the channel number is embedded into the command field, whereas with the C0_SETDM, the pointer to the channel context area must be written into the extended buffer address field of the buffer descriptor.

7.2.6.1.3 Example of Buffer Descriptors for Channel 0.

Figure 7-19 illustrates the buffer descriptors that must be set in Arm platform memory space, before execution of boot code, to download contexts and scripts of channels 1, 4, and 10. After boot code execution, SDMA memory will be populated with the different contexts and scripts as presented in the following figure.

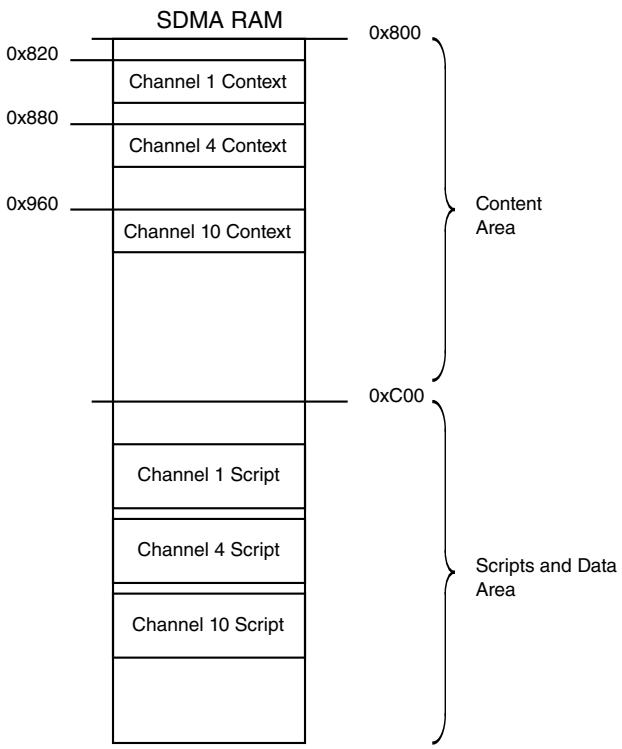
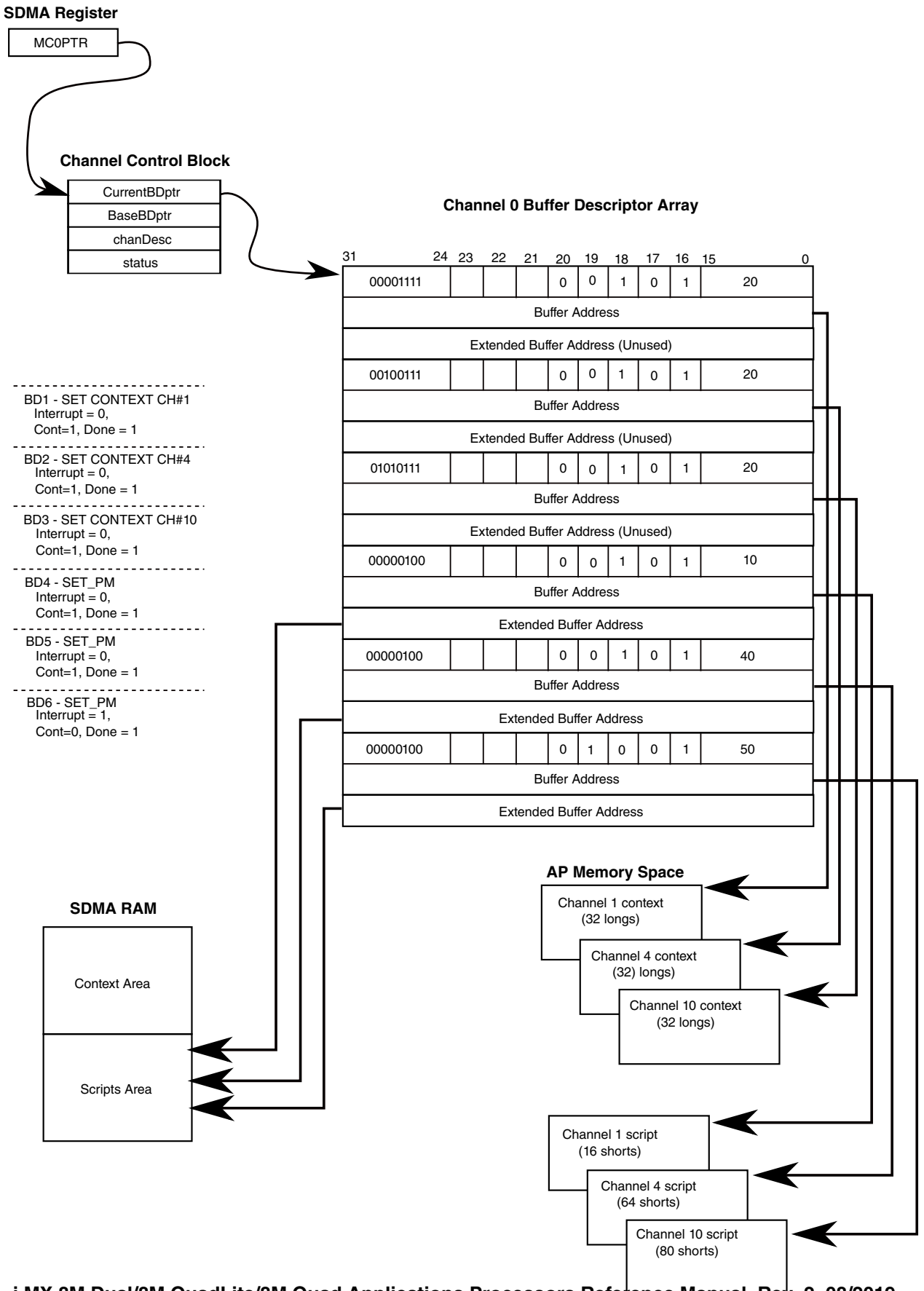


Figure 7-18. Example of SDMA RAM After Boot Session


SDMA Register
MC0PTR
Channel Control Block
CurrentBDptr
BaseBDptr
chanDesc
status
Channel 0 Buffer Descriptor Array
31 24 23 22 21 20 19 18 17 16 15 0
00001111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00100111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
01010111 0 0 1 0 1 20
Buffer Address
Extended Buffer Address (Unused)
00000100 0 0 1 0 1 10
Buffer Address
Extended Buffer Address
00000100 0 0 1 0 1 40
Buffer Address
Extended Buffer Address
00000100 0 1 0 0 1 50
Buffer Address
Extended Buffer Address
BD1 - SET CONTEXT CH#1
Interrupt = 0,
Cont=1, Done = 1
BD2 - SET CONTEXT CH#4
Interrupt = 0,
Cont=1, Done = 1
BD3 - SET CONTEXT CH#10
Interrupt = 0,
Cont=1, Done = 1
BD4 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD5 - SET_PM
Interrupt = 0,
Cont=1, Done = 1
BD6 - SET_PM
Interrupt = 1,
Cont=0, Done = 1
AP Memory Space
Channel 1 context
(32 longs)
Channel 4 context
(32) longs)
Channel 10 context
(32 longs)
SDMA RAM
Context Area
Scripts Area
Channel 1 script
(16 shorts)
Channel 4 script
(64 shorts)
Channel 10 script
(80 shorts)

7.2.6.1.4 Channel Context

There are 32 channel context memory structures pointed to by the local save area pointer. These channel context memory structures are fixed.

The script in the SDMA computes the memory offset for a given channel based on the structure length and channel number. Figure below shows the structure of the channel context as it is saved in the SDMA local memory (RAM).

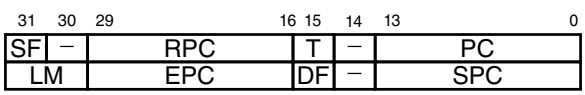
A channel context consists in 24 words, one per register. A total of 32 words are reserved for every channel. The additional 8 words are called scratch ram and they are dedicated to each channel. This memory area is commonly used for stack management.

The structure is divided in 4 areas:

- Channel status registers
- General purpose registers
- Functional units state registers reflecting the state of the Arm platform DMAs (Burst and Peripheral DMA).
- Scratch RAM

The details of the channel context status registers are described in the following figure.

The PC field of the first long register must point to the SDMA RAM address where the script that will be executed on the channel is located and this value equals the one stored in the extended buffer address of the buffer descriptor with C0_SETPM command.



SF: Source fault while loading data
RPC: Return program counter
T: Test bit: status of arithmetic and test instructions
PC: Program counter
LM: Loop mode
EPC: Loop end program counter
DF: Destination fault while storing data
SPC: Loop Start program counter

Figure 7-20. SDMA State Registers (ShPC, ShLoop)

7.2.6.2 Typical Data Transfer Supported by SDMA DMA Units

This section presents a library of SDMA scripts that perform data transfers through the peripheral DMA and the burst DMA units.

The Arm platform memory and peripherals are devices that either the peripheral DMA or the burst DMA can access. The scripts are given for a peripheral DMA whose address registers are programmed in incremented mode when internal memory is involved. See the following table for the summary.

Table 7-55. Typical Data Transfers Summary

Data Transfer	Peripheral DMA	Burst DMA	Comments
Arm platform External Memory ↔ Arm platform External Memory		3	Copy mode Script example, see Burst DMA Unit Copy Mode and External Memory to External Memory.
Arm platform Peripheral ↔ Arm platform Peripheral	3		Copy mode if same data path width Script example, see Peripheral to Peripheral Transfer.
Arm platform External Memory ↔ Arm platform Peripheral	3	3	Data transit through SDMA Script example, see Transfer Between Peripheral and External Memory.
Arm platform External Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Transfer Between External Memory and Internal Memory.
Arm platform Internal Memory ↔ Arm platform Internal Memory		3	Copy mode Script example, see Internal Memory to Internal Memory.
Arm platform Internal memory ↔ Arm platform Peripheral	3		Data transit through SDMA Script example, see Transfer Between Peripheral and Internal Memory.

NOTE

These scripts are provided as examples of how to use DMA blocks to perform required data transfers: They are not "official" programs.

7.2.6.2.1 External Memory to External Memory

This section describes the SDMA script that performs data moves in external memory.

For this particular data transfer, only the burst DMA is used. It is programmed in copy mode, so no data transmits through an SDMA general register.

The SDMA core only monitors data transfer status. It is assumed source and destination address values are already present in two SDMA general registers (r1 and r2). For this example, it is also assumed that a 32-bit word-to-move for source-to-destination address is present in r0 and equals 64.

Data Moves in External Memory

```
1         stf r1,MSA                          // Source address setup

2         stf r2,MDA                          // Destination address setup

3         ldi r0,0x64                         // 64 words must be transferred from MSA to
MDA

4         ldi r1,0x8

MAIN_XFER:

5         cmphs r0,r1                         // Is r0 >= 0x8

6         bf LAST_XFER                      // If not, jump to last transfer label

7         stf r1,MD|CPY                        // Copy 8 words from MSA to MDA address.

8         subi r0,0x8                         // Decrement counter

9         jmp MAIN_XFER                     // return to main transfer loop

LAST_XFER:

10        stf r0,MD|CPY                        // perform last transfer
```

All instructions are performed in one cycle (jumps excepted). Instruction 7 triggers a copy transfer: A read burst access of 8-word starts, data is staged in MD and then a write burst of 8 words is executed. Instruction 8, 9, 5, and 6 are executed while the burst access is in progress. If this access is not complete when instruction 7 is executed a second time, SDMA stalls on this instruction as long as the previous copy transfer is not over. In this case, the instruction is no longer a one-cycle instruction.

During the main loop (MAIN_XFER), r1 always equals 8, so burst lengths are 8 words. On the last ldf |CPY instruction (10), r1 equals the reminder of r0 divided by 8; therefore, the length of bursts triggered in copy mode equal r1 value, which is between 1 and 7.

7.2.6.2.2 Peripheral to Peripheral Transfer

For this data transfer, only the peripheral DMA is used.

It is programmed in copy mode, so no data will transmit through the SDMA general register used in the ldf instruction, but the contents of the general register are lost. The SDMA core only monitors the transfer.

7.2.6.2.2.1 Source and Destination Target Have the Same Data Path Width

When the source and destination target have the same data path width, the following is true:

- Source target is a *half-word* (16-bit) peripheral located at address 0x1002.
- Destination is a *half-word* (16-bit) peripheral located at address 0x2006.

It is assumed the address values are already present in two SDMA general registers (r1, r2). The script for a transfer of 10 half-word is as follows:

Same Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ16|F                    //r1=0x1002 Source address register setup
2         stf r2, PDA|SZ16|F                    //r2=0x2006 Destination address register
setup
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                           //loop counter is 10
//MAIN LOOP TRANFER
copy_loop:

5         loop 2,0
6           ldf r7,PD|CPY                      //Reads 1 half-word from src and writes to
dest.
7           yield
8         bdf ERROR_DURING_XFER
ERROR_ADDR_SETUP:
                    //correction of PSA/PDA setup and jumps to main loop transfer
ERROR_DURING_XFER:
//flag error is set,
//PS can be read to know if error occurs during read or write access.
```

If a data transfer must occur between two word peripherals, only the setup section should be updated. The transfer itself is always performed by the hardware loop instruction.

All instructions are executed in one cycle (change of flow excepted). On instruction 6, a single read access is triggered, read data is staged in PD, and a write-to-destination is executed. When the transfers are in progress, the SDMA can execute he next instructions in parallel. If instruction 6, which performs the copy transfer, is executed while the previous access is not over, SDMA is stalled and instruction ldf is a multi-cycle instruction.

7.2.6.2.2.2 Source and Destination Target Have a Different Data Path Width

When the source and destination target have a different data path width, copy mode cannot be used, and any attempt to initiate a copy transfer immediately raises an error, which is stored in the SF flag.

The following example shows the SDMA code that could transfer 10 words from a *word* (32-bit) peripheral to a *half-word* peripheral whose addresses are preliminary and stored in r1 and r2.

Different Data Path Width for Source and Destination

```
//SETUP SECTION
1         stf r1, PSA|SZ32|F|PF                    //r1=0x1000 and prefetch data
2         stf r2, PDA|SZ16|F                      //r2=0x2006
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0xa                             //loop counter is 10
//MAIN LOOP TRANFER
main_loop_xfer_16_16:
5         loop 6,0
6           ldf r7,PD                           //copy 32-bit of PD in r7
7           stf r7,PD                            //store 16 LSB of r7 in PD and a flush is
executed
8           rorb r7
9           rorb r7                            //16 MSB --> 16 LSB
10          stf r7,PD                            //store 16 LSB of r6 in PD and a flush.
11          yield
```

On instruction 1, when the source address register is programmed and a data prefetch is required, a read access is executed. In parallel, the SDMA executes instructions 2 to 5. On instruction 6, the SDMA tries to read data that was fetched by instruction 1. If data is ready, the ldf will be a one cycle instruction; otherwise, the SDMA is stalled as long as the read access is not finished. Then, the 16 LSB of the read data is stored in PD and automatically flushed to the destination peripheral. In parallel, the SDMA executes the rotation instructions (8, 9), and stores the 16 MSB of the read data into PD. If a previous write access is finished, instruction 10 will be a one-cycle instruction.

The main loop transfer may appear inefficient, but due to wait states imposed to the peripheral DMA each time an external access is performed, a software pipeline is in place. During the time needed to flush PD, the SDMA executes the move and rotation operations. SDMA executes instructions in parallel with DMA accesses.

7.2.6.2.3 Transfer Between Peripheral and External Memory

7.2.6.2.3.1 Peripheral to External Memory Transfer

A transfer from a peripheral to the external memory controller involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from word peripheral to the external memory would be as follows:

Peripheral to External Memory Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, PSA|SZ16|F|PF                     //r1=0x1000 and prefetch 32-bit data
2         stf r2, MDA                             //r2=0x2000, setup burst DMA destination
address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64                             //loop counter is 100
5
          //MAIN LOOP TRANFER
6         loop 3,0
7           ldf r1,PD|PF                          // read 32 bits of PD and initiate a new read
```

```
access.
8           stf r1,MD|32                    // store 32 bits of r1 in the MD fifo.
9           yield
10        ldf r1,PD                       // last word data is read
11        stf r1,MD|32|FL                   // to flush all remaining bytes of MD
```

On instruction 1, the source address register of the peripheral DMA is programmed and data is fetched. This data is stored in PD and the SDMA reads PD during instruction 7, which is a one-cycle instruction that is read-access finished. On the same instruction (7), a data prefetch is required and a read access to the source peripheral is executed. In parallel, the SDMA stored the previous read data into the data register of MD. When MD (which is an eight-word FIFO) is full, a burst write access is executed to empty the FIFO. As long as the next SDMA instructions do not access the burst DMA, they will be one-cycle instructions. The following figures show how the peripheral DMA and burst DMA work in parallel.

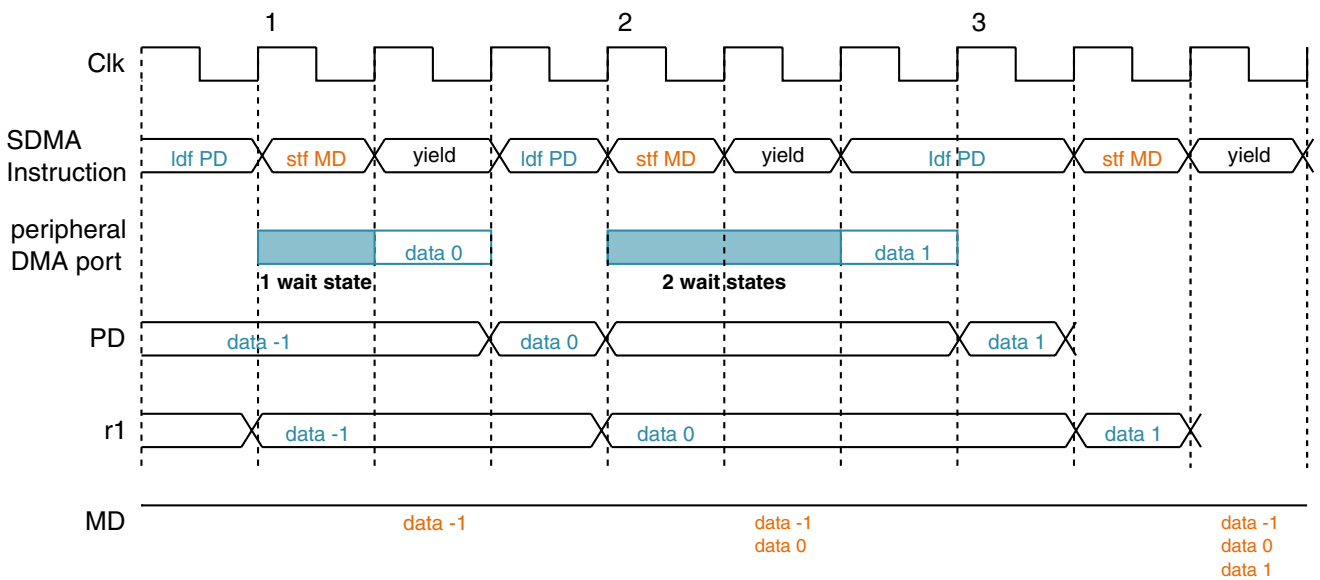


Figure 7-21. Peripheral to External Memory Example (1)

As seen in the figure above, the read access triggered by the ldf PD instruction is symbolized by the blue bar when in progress. After wait states, the read data (data 0, data 1) is stored in PD on the clk rising edge. On edge 2, data 0 is available in PD so it can be transferred to the SDMA general register r1, and then stored in MD FIFO. On edge 3, data 1 is not in PD; therefore, SDMA is stalled on the ldf instruction, which lasts two cycles. The figure below shows an example of when MD FIFO is full with data.

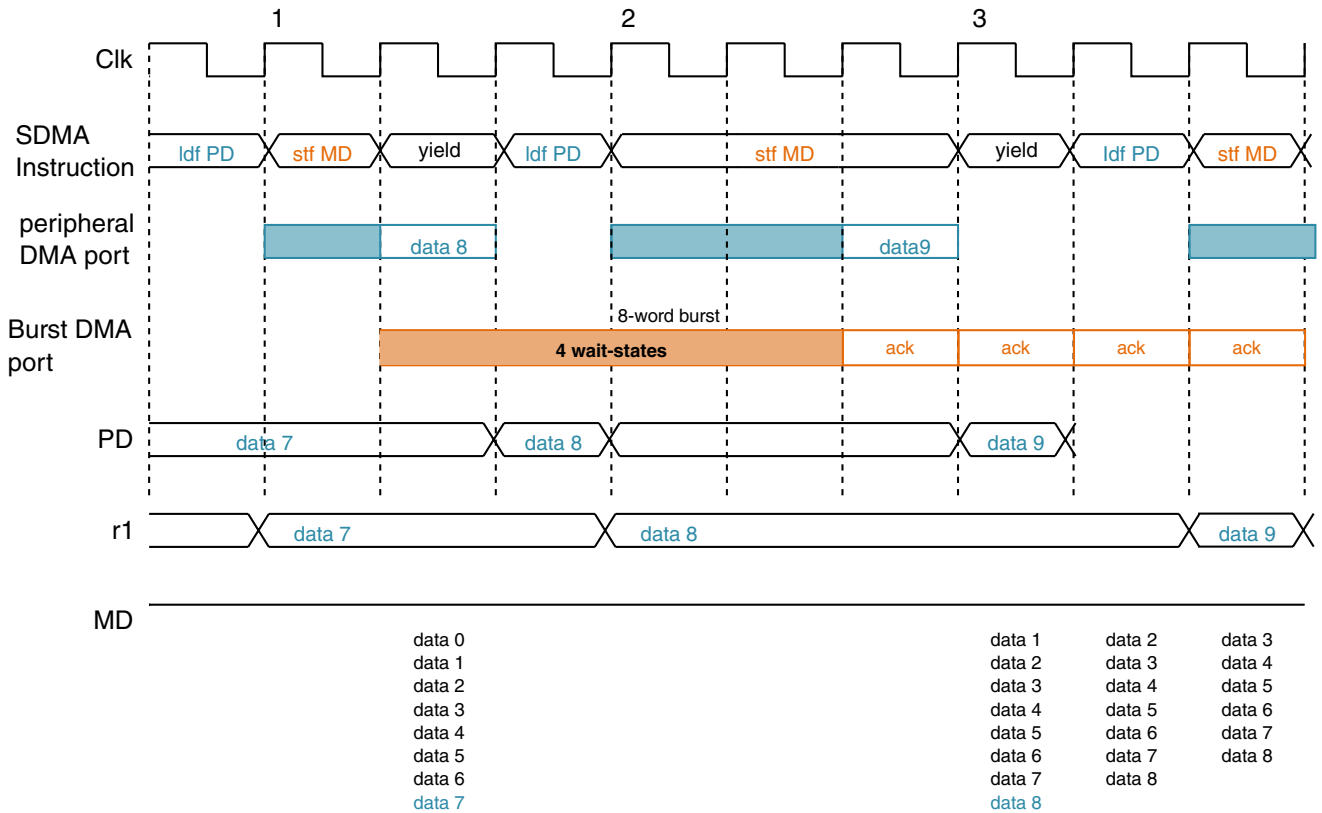


Figure 7-22. Peripheral to External Memory Example (2)

In the previous figure, the write bar means the burst DMA is performing a write burst access. The latency to have the first write acknowledge is four cycles. SDMA is stalled on instruction stf because no acknowledge was received, MD FIFO is full, and there is no empty slot to store data 9. When an acknowledge is sampled by the burst DMA, FIFO is shifted and data 8 is written. As long as there is at least one empty slot in MD FIFO, the stf MD instruction lasts one cycle.

7.2.6.2.3.2 External Memory to Peripheral Transfer

A transfer from the external memory to a peripheral involves the peripheral DMA and the burst DMA.

The code for transferring 100 word from external memory to a word peripheral would be as follows:

External Memory to Peripheral Transfer

```
//SETUP SECTION source and destination addresses are already in r1 and r2
1         stf r1, MSA|PF             //r1=0x1000 and starts a 8-word read burst
2         stf r2, PDA|SZ32|P          //r2=0x2010, setup peripheral DMA destination address
3         bdf ERROR_ADDR_SETUP
4         ldi r0,0x64               //loop counter is 100
//MAIN LOOP TRANFER
```

```
6          loop 3,0
7            ldf r1,MD|32|PF          // read 32 bits of MD and initiate a new read access
                                   // if MD is empty after this reading.
8            stf r1,PD              // store 32 bits of r1 in the PD.
9            yield
           10 ldf r1,MD|32           // last word data is read
11         stf r1,PD                // last write access
```

On instruction 1, a read burst of 8 words begins. Read data is staged into MD. On instruction 7 (and if data is available in MD), 32 bits are copied into r1. Then instruction 8 writes them into PD and an automatic flush is executed. The SDMA core, peripheral DMA, and burst DMA can work in parallel as long as no SDMA instruction tries to start a new write access on the peripheral DMA while the previous access is still in progress, or as long as there is data in MD when the SDMA tries to read it.

7.2.6.2.4 Transfer Between External Memory and Internal Memory

Since the internal memory (Arm platform RAM) is accessed via the peripheral DMA and the external memory is accessed via the burst DMA, the SDMA scripts that are described in Transfer Between Peripheral and External Memory can be reused. The exception is that the peripheral DMA address registers (PSA or PDA, depending on the script) should be programmed in incremented mode rather than frozen mode.

7.2.6.2.4.1 Internal Memory to Internal Memory

The internal memory can only be accessed via the peripheral DMA, so the script described in Peripheral to Peripheral Transfer can be reused with a different programming of the peripheral DMA address registers.

7.2.6.2.4.2 Transfer Between Peripheral and Internal Memory

For this transfer, the peripheral DMA is also used in copy mode.

The SDMA script is very similar to the one described in Peripheral to Peripheral Transfer, except for the peripheral DMA address registers programming.

7.2.7 Arm Platform Memory Map and Control Register Definitions

The Arm platform controls the SDMA by means of several interface registers. Those registers are described in the current section.

All registers are clocked with the SDMA clock (which means the Arm platform must ensure that the SDMA clock is running when it wants to access any register).

SDMAARM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0000	Arm platform Channel 0 Pointer (SDMAARM2_MC0PTR)	32	R/W	0000_0000h	7.2.7.1/ 1236
302C_0004	Channel Interrupts (SDMAARM2_INTR)	32	w1c	0000_0000h	7.2.7.2/ 1236
302C_0008	Channel Stop/Channel Status (SDMAARM2_STOP_STAT)	32	w1c	0000_0000h	7.2.7.3/ 1237
302C_000C	Channel Start (SDMAARM2_HSTART)	32	R/W	0000_0000h	7.2.7.4/ 1237
302C_0010	Channel Event Override (SDMAARM2_EVTOVR)	32	R/W	0000_0000h	7.2.7.5/ 1238
302C_0014	Channel BP Override (SDMAARM2_DSPOVR)	32	R/W	FFFF_FFFFh	7.2.7.6/ 1238
302C_0018	Channel Arm platform Override (SDMAARM2_HOSTOVR)	32	R/W	0000_0000h	7.2.7.7/ 1238
302C_001C	Channel Event Pending (SDMAARM2_EVTPEND)	32	w1c	0000_0000h	7.2.7.8/ 1239
302C_0024	Reset Register (SDMAARM2_RESET)	32	R	0000_0000h	7.2.7.9/ 1240
302C_0028	DMA Request Error Register (SDMAARM2_EVTERR)	32	R	0000_0000h	7.2.7.10/ 1241
302C_002C	Channel Arm platform Interrupt Mask (SDMAARM2_INTRMASK)	32	R/W	0000_0000h	7.2.7.11/ 1241
302C_0030	Schedule Status (SDMAARM2_PSW)	32	R	0000_0000h	7.2.7.12/ 1242
302C_0034	DMA Request Error Register (SDMAARM2_EVTERRDBG)	32	R	0000_0000h	7.2.7.13/ 1242
302C_0038	Configuration Register (SDMAARM2_CONFIG)	32	R/W	0000_0003h	7.2.7.14/ 1243
302C_003C	SDMA LOCK (SDMAARM2_SDMA_LOCK)	32	R/W	0000_0000h	7.2.7.15/ 1244
302C_0040	OnCE Enable (SDMAARM2_ONCE_ENB)	32	R/W	0000_0000h	7.2.7.16/ 1245

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0044	OnCE Data Register (SDMAARM2_ONCE_DATA)	32	R/W	0000_0000h	7.2.7.17/ 1245
302C_0048	OnCE Instruction Register (SDMAARM2_ONCE_INSTR)	32	R/W	0000_0000h	7.2.7.18/ 1246
302C_004C	OnCE Status Register (SDMAARM2_ONCE_STAT)	32	R	0000_E000h	7.2.7.19/ 1246
302C_0050	OnCE Command Register (SDMAARM2_ONCE_CMD)	32	R/W	0000_0000h	7.2.7.20/ 1248
302C_0058	Illegal Instruction Trap Address (SDMAARM2_ILLINSTADDR)	32	R/W	0000_0001h	7.2.7.21/ 1248
302C_005C	Channel 0 Boot Address (SDMAARM2_CHN0ADDR)	32	R/W	0000_0050h	7.2.7.22/ 1249
302C_0060	DMA Requests (SDMAARM2_EVT_MIRROR)	32	R	0000_0000h	7.2.7.23/ 1250
302C_0064	DMA Requests 2 (SDMAARM2_EVT_MIRROR2)	32	R	0000_0000h	7.2.7.24/ 1250
302C_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM2_XTRIG_CONF1)	32	R/W	0000_0000h	7.2.7.25/ 1251
302C_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM2_XTRIG_CONF2)	32	R/W	0000_0000h	7.2.7.26/ 1252
302C_0100	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI0)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0104	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI1)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0108	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI2)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_010C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI3)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0110	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI4)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0114	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI5)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0118	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI6)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_011C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI7)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0120	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI8)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0124	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI9)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0128	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI10)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_012C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI11)	32	R/W	0000_0000h	7.2.7.27/ 1253

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0130	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI12)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0134	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI13)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0138	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI14)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_013C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI15)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0140	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI16)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0144	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI17)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0148	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI18)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_014C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI19)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0150	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI20)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0154	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI21)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0158	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI22)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_015C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI23)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0160	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI24)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0164	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI25)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0168	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI26)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_016C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI27)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0170	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI28)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0174	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI29)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0178	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI30)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_017C	Channel Priority Registers (SDMAARM2_SDMA_CHNPRI31)	32	R/W	0000_0000h	7.2.7.27/ 1253
302C_0200	Channel Enable RAM (SDMAARM2_CHNENBL0)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0204	Channel Enable RAM (SDMAARM2_CHNENBL1)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0208	Channel Enable RAM (SDMAARM2_CHNENBL2)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_020C	Channel Enable RAM (SDMAARM2_CHNENBL3)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0210	Channel Enable RAM (SDMAARM2_CHNENBL4)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0214	Channel Enable RAM (SDMAARM2_CHNENBL5)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0218	Channel Enable RAM (SDMAARM2_CHNENBL6)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_021C	Channel Enable RAM (SDMAARM2_CHNENBL7)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0220	Channel Enable RAM (SDMAARM2_CHNENBL8)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0224	Channel Enable RAM (SDMAARM2_CHNENBL9)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0228	Channel Enable RAM (SDMAARM2_CHNENBL10)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_022C	Channel Enable RAM (SDMAARM2_CHNENBL11)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0230	Channel Enable RAM (SDMAARM2_CHNENBL12)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0234	Channel Enable RAM (SDMAARM2_CHNENBL13)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0238	Channel Enable RAM (SDMAARM2_CHNENBL14)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_023C	Channel Enable RAM (SDMAARM2_CHNENBL15)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0240	Channel Enable RAM (SDMAARM2_CHNENBL16)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0244	Channel Enable RAM (SDMAARM2_CHNENBL17)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0248	Channel Enable RAM (SDMAARM2_CHNENBL18)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_024C	Channel Enable RAM (SDMAARM2_CHNENBL19)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0250	Channel Enable RAM (SDMAARM2_CHNENBL20)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0254	Channel Enable RAM (SDMAARM2_CHNENBL21)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0258	Channel Enable RAM (SDMAARM2_CHNENBL22)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_025C	Channel Enable RAM (SDMAARM2_CHNENBL23)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0260	Channel Enable RAM (SDMAARM2_CHNENBL24)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0264	Channel Enable RAM (SDMAARM2_CHNENBL25)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0268	Channel Enable RAM (SDMAARM2_CHNENBL26)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_026C	Channel Enable RAM (SDMAARM2_CHNENBL27)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0270	Channel Enable RAM (SDMAARM2_CHNENBL28)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0274	Channel Enable RAM (SDMAARM2_CHNENBL29)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0278	Channel Enable RAM (SDMAARM2_CHNENBL30)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_027C	Channel Enable RAM (SDMAARM2_CHNENBL31)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0280	Channel Enable RAM (SDMAARM2_CHNENBL32)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0284	Channel Enable RAM (SDMAARM2_CHNENBL33)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0288	Channel Enable RAM (SDMAARM2_CHNENBL34)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_028C	Channel Enable RAM (SDMAARM2_CHNENBL35)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0290	Channel Enable RAM (SDMAARM2_CHNENBL36)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0294	Channel Enable RAM (SDMAARM2_CHNENBL37)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_0298	Channel Enable RAM (SDMAARM2_CHNENBL38)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_029C	Channel Enable RAM (SDMAARM2_CHNENBL39)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02A0	Channel Enable RAM (SDMAARM2_CHNENBL40)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02A4	Channel Enable RAM (SDMAARM2_CHNENBL41)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02A8	Channel Enable RAM (SDMAARM2_CHNENBL42)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02AC	Channel Enable RAM (SDMAARM2_CHNENBL43)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02B0	Channel Enable RAM (SDMAARM2_CHNENBL44)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02B4	Channel Enable RAM (SDMAARM2_CHNENBL45)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_02B8	Channel Enable RAM (SDMAARM2_CHNENBL46)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_02BC	Channel Enable RAM (SDMAARM2_CHNENBL47)	32	R/W	0000_0000h	7.2.7.28/ 1254
302C_1000	SDMA DONE0 Configuration (SDMAARM2_DONE0_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.29/ 1254
302C_1004	SDMA DONE1 Configuration (SDMAARM2_DONE1_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.30/ 1256
30BD_0000	Arm platform Channel 0 Pointer (SDMAARM1_MC0PTR)	32	R/W	0000_0000h	7.2.7.1/ 1236
30BD_0004	Channel Interrupts (SDMAARM1_INTR)	32	w1c	0000_0000h	7.2.7.2/ 1236
30BD_0008	Channel Stop/Channel Status (SDMAARM1_STOP_STAT)	32	w1c	0000_0000h	7.2.7.3/ 1237
30BD_000C	Channel Start (SDMAARM1_HSTART)	32	R/W	0000_0000h	7.2.7.4/ 1237
30BD_0010	Channel Event Override (SDMAARM1_EVTOVR)	32	R/W	0000_0000h	7.2.7.5/ 1238
30BD_0014	Channel BP Override (SDMAARM1_DSPOVR)	32	R/W	FFFF_FFFFh	7.2.7.6/ 1238
30BD_0018	Channel Arm platform Override (SDMAARM1_HOSTOVR)	32	R/W	0000_0000h	7.2.7.7/ 1238
30BD_001C	Channel Event Pending (SDMAARM1_EVTPEND)	32	w1c	0000_0000h	7.2.7.8/ 1239
30BD_0024	Reset Register (SDMAARM1_RESET)	32	R	0000_0000h	7.2.7.9/ 1240
30BD_0028	DMA Request Error Register (SDMAARM1_EVTERR)	32	R	0000_0000h	7.2.7.10/ 1241
30BD_002C	Channel Arm platform Interrupt Mask (SDMAARM1_INTRMASK)	32	R/W	0000_0000h	7.2.7.11/ 1241
30BD_0030	Schedule Status (SDMAARM1_PSW)	32	R	0000_0000h	7.2.7.12/ 1242
30BD_0034	DMA Request Error Register (SDMAARM1_EVTERRDBG)	32	R	0000_0000h	7.2.7.13/ 1242
30BD_0038	Configuration Register (SDMAARM1_CONFIG)	32	R/W	0000_0003h	7.2.7.14/ 1243
30BD_003C	SDMA LOCK (SDMAARM1_SDMA_LOCK)	32	R/W	0000_0000h	7.2.7.15/ 1244
30BD_0040	OnCE Enable (SDMAARM1_ONCE_ENB)	32	R/W	0000_0000h	7.2.7.16/ 1245
30BD_0044	OnCE Data Register (SDMAARM1_ONCE_DATA)	32	R/W	0000_0000h	7.2.7.17/ 1245
30BD_0048	OnCE Instruction Register (SDMAARM1_ONCE_INSTR)	32	R/W	0000_0000h	7.2.7.18/ 1246

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_004C	OnCE Status Register (SDMAARM1_ONCE_STAT)	32	R	0000_E000h	7.2.7.19/ 1246
30BD_0050	OnCE Command Register (SDMAARM1_ONCE_CMD)	32	R/W	0000_0000h	7.2.7.20/ 1248
30BD_0058	Illegal Instruction Trap Address (SDMAARM1_ILLINSTADDR)	32	R/W	0000_0001h	7.2.7.21/ 1248
30BD_005C	Channel 0 Boot Address (SDMAARM1_CHN0ADDR)	32	R/W	0000_0050h	7.2.7.22/ 1249
30BD_0060	DMA Requests (SDMAARM1_EVT_MIRROR)	32	R	0000_0000h	7.2.7.23/ 1250
30BD_0064	DMA Requests 2 (SDMAARM1_EVT_MIRROR2)	32	R	0000_0000h	7.2.7.24/ 1250
30BD_0070	Cross-Trigger Events Configuration Register 1 (SDMAARM1_XTRIG_CONF1)	32	R/W	0000_0000h	7.2.7.25/ 1251
30BD_0074	Cross-Trigger Events Configuration Register 2 (SDMAARM1_XTRIG_CONF2)	32	R/W	0000_0000h	7.2.7.26/ 1252
30BD_0100	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI0)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0104	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI1)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0108	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI2)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_010C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI3)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0110	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI4)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0114	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI5)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0118	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI6)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_011C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI7)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0120	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI8)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0124	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI9)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0128	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI10)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_012C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI11)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0130	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI12)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0134	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI13)	32	R/W	0000_0000h	7.2.7.27/ 1253

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0138	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI14)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_013C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI15)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0140	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI16)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0144	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI17)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0148	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI18)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_014C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI19)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0150	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI20)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0154	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI21)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0158	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI22)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_015C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI23)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0160	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI24)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0164	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI25)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0168	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI26)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_016C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI27)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0170	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI28)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0174	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI29)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0178	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI30)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_017C	Channel Priority Registers (SDMAARM1_SDMA_CHNPRI31)	32	R/W	0000_0000h	7.2.7.27/ 1253
30BD_0200	Channel Enable RAM (SDMAARM1_CHNENBL0)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0204	Channel Enable RAM (SDMAARM1_CHNENBL1)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0208	Channel Enable RAM (SDMAARM1_CHNENBL2)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_020C	Channel Enable RAM (SDMAARM1_CHNENBL3)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0210	Channel Enable RAM (SDMAARM1_CHNENBL4)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0214	Channel Enable RAM (SDMAARM1_CHNENBL5)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0218	Channel Enable RAM (SDMAARM1_CHNENBL6)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_021C	Channel Enable RAM (SDMAARM1_CHNENBL7)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0220	Channel Enable RAM (SDMAARM1_CHNENBL8)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0224	Channel Enable RAM (SDMAARM1_CHNENBL9)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0228	Channel Enable RAM (SDMAARM1_CHNENBL10)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_022C	Channel Enable RAM (SDMAARM1_CHNENBL11)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0230	Channel Enable RAM (SDMAARM1_CHNENBL12)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0234	Channel Enable RAM (SDMAARM1_CHNENBL13)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0238	Channel Enable RAM (SDMAARM1_CHNENBL14)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_023C	Channel Enable RAM (SDMAARM1_CHNENBL15)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0240	Channel Enable RAM (SDMAARM1_CHNENBL16)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0244	Channel Enable RAM (SDMAARM1_CHNENBL17)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0248	Channel Enable RAM (SDMAARM1_CHNENBL18)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_024C	Channel Enable RAM (SDMAARM1_CHNENBL19)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0250	Channel Enable RAM (SDMAARM1_CHNENBL20)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0254	Channel Enable RAM (SDMAARM1_CHNENBL21)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0258	Channel Enable RAM (SDMAARM1_CHNENBL22)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_025C	Channel Enable RAM (SDMAARM1_CHNENBL23)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0260	Channel Enable RAM (SDMAARM1_CHNENBL24)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0264	Channel Enable RAM (SDMAARM1_CHNENBL25)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0268	Channel Enable RAM (SDMAARM1_CHNENBL26)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_026C	Channel Enable RAM (SDMAARM1_CHNENBL27)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0270	Channel Enable RAM (SDMAARM1_CHNENBL28)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0274	Channel Enable RAM (SDMAARM1_CHNENBL29)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0278	Channel Enable RAM (SDMAARM1_CHNENBL30)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_027C	Channel Enable RAM (SDMAARM1_CHNENBL31)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0280	Channel Enable RAM (SDMAARM1_CHNENBL32)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0284	Channel Enable RAM (SDMAARM1_CHNENBL33)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0288	Channel Enable RAM (SDMAARM1_CHNENBL34)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_028C	Channel Enable RAM (SDMAARM1_CHNENBL35)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0290	Channel Enable RAM (SDMAARM1_CHNENBL36)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0294	Channel Enable RAM (SDMAARM1_CHNENBL37)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_0298	Channel Enable RAM (SDMAARM1_CHNENBL38)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_029C	Channel Enable RAM (SDMAARM1_CHNENBL39)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02A0	Channel Enable RAM (SDMAARM1_CHNENBL40)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02A4	Channel Enable RAM (SDMAARM1_CHNENBL41)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02A8	Channel Enable RAM (SDMAARM1_CHNENBL42)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02AC	Channel Enable RAM (SDMAARM1_CHNENBL43)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02B0	Channel Enable RAM (SDMAARM1_CHNENBL44)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02B4	Channel Enable RAM (SDMAARM1_CHNENBL45)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02B8	Channel Enable RAM (SDMAARM1_CHNENBL46)	32	R/W	0000_0000h	7.2.7.28/ 1254
30BD_02BC	Channel Enable RAM (SDMAARM1_CHNENBL47)	32	R/W	0000_0000h	7.2.7.28/ 1254

Table continues on the next page...

SDMAARM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_1000	SDMA DONE0 Configuration (SDMAARM1_DONE0_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.29/ 1254
30BD_1004	SDMA DONE1 Configuration (SDMAARM1_DONE1_CONFIG)	32	R/W	1F1F_1F1Fh	7.2.7.30/ 1256

7.2.7.1 Arm platform Channel 0 Pointer (SDMAARMx_MC0PTR)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MC0PTR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_MC0PTR field descriptions

Field	Description
MC0PTR	**Channel 0 Pointer** contains the 32-bit address, in Arm platform memory, of channel 0 control block (the boot channel). Appendix A fully describes the SDMA Application Programming Interface (API). The Arm platform has a read/write access and the SDMA has a read-only access.

7.2.7.2 Channel Interrupts (SDMAARMx_INTR)

Address: Base address + 4h offset

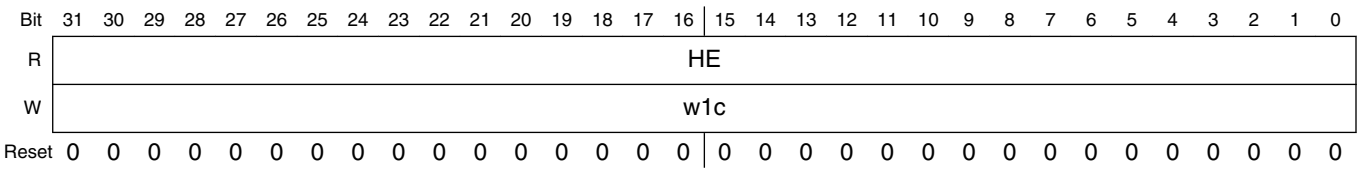
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	HI[31:0]																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_INTR field descriptions

Field	Description
HI[31:0]	The Arm platform Interrupts register contains the 32 HI[i] bits. If any bit is set, it will cause an interrupt to the Arm platform. This register is a "write-ones" register to the Arm platform. When the Arm platform sets a bit in this register the corresponding HI[i] bit is cleared. The interrupt service routine should clear individual channel bits when their interrupts are serviced, failure to do so will cause continuous interrupts. The SDMA is responsible for setting the HI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

7.2.7.3 Channel Stop/Channel Status (SDMAARMx_STOP_STAT)

Address: Base address + 8h offset

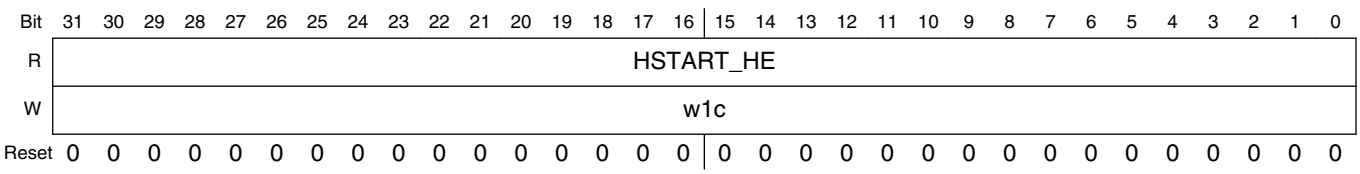


SDMAARMx_STOP_STAT field descriptions

Field	Description
HE	This 32-bit register gives access to the Arm platform Enable bits. There is one bit for every channel. This register is a "write-ones" register to the Arm platform. When the Arm platform writes 1 in bit i of this register, it clears the HE[i] and HSTART[i] bits. Reading this register yields the current state of the HE[i] bits.

7.2.7.4 Channel Start (SDMAARMx_HSTART)

Address: Base address + Ch offset



SDMAARMx_HSTART field descriptions

Field	Description
HSTART_HE	The HSTART_HE registers are 32 bits wide with one bit for every channel. When a bit is written to 1, it enables the corresponding channel. Two physical registers are accessed with that address (HSTART and HE), which enables the Arm platform to trigger a channel a second time before the first trigger is processed. • This register is a "write-ones" register to the Arm platform. Neither HSTART[i] bit can be set while the corresponding HE[i] bit is cleared. • When the Arm platform tries to set the HSTART[i] bit by writing a one (if the corresponding HE[i] bit is clear), the bit in the HSTART[i] register will remain cleared and the HE[i] bit will be set. • If the corresponding HE[i] bit was already set, the HSTART[i] bit will be set. The next time the SDMA channel *i* attempts to clear the HE[i] bit by means of a `done` instruction, the bit in the HSTART[i] register will be cleared and the HE[i] bit will take the old value of the HSTART[i] bit. • Reading this register yields the current state of the HSTART[i] bits. This mechanism enables the Arm platform to pipeline two HSTART commands per channel.

7.2.7.5 Channel Event Override (SDMAARMx_EVTOVR)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVTOVR field descriptions

Field	Description
EO	The Channel Event Override register contains the 32 EO[i] bits. A bit set in this register causes the SDMA to ignore DMA requests when scheduling the corresponding channel.

7.2.7.6 Channel BP Override (SDMAARMx_DSPOVR)

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DO																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

SDMAARMx_DSPOVR field descriptions

Field	Description
DO	This register is reserved. All DO bits should be set to the reset value of 1. A setting of 0 will prevent SDMA channels from starting according to the condition described in Runnable Channels Evaluation. 0 - Reserved 1 - Reset value.

7.2.7.7 Channel Arm platform Override (SDMAARMx_HOSTOVR)

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HO																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_HOSTOVR field descriptions

Field	Description
HO	The Channel Arm platform Override register contains the 32 HO[i] bits. A bit set in this register causes the SDMA to ignore the Arm platform enable bit (HE) when scheduling the corresponding channel.

7.2.7.8 Channel Event Pending (SDMAARMx_EVTPEND)

Address: Base address + 1Ch offset

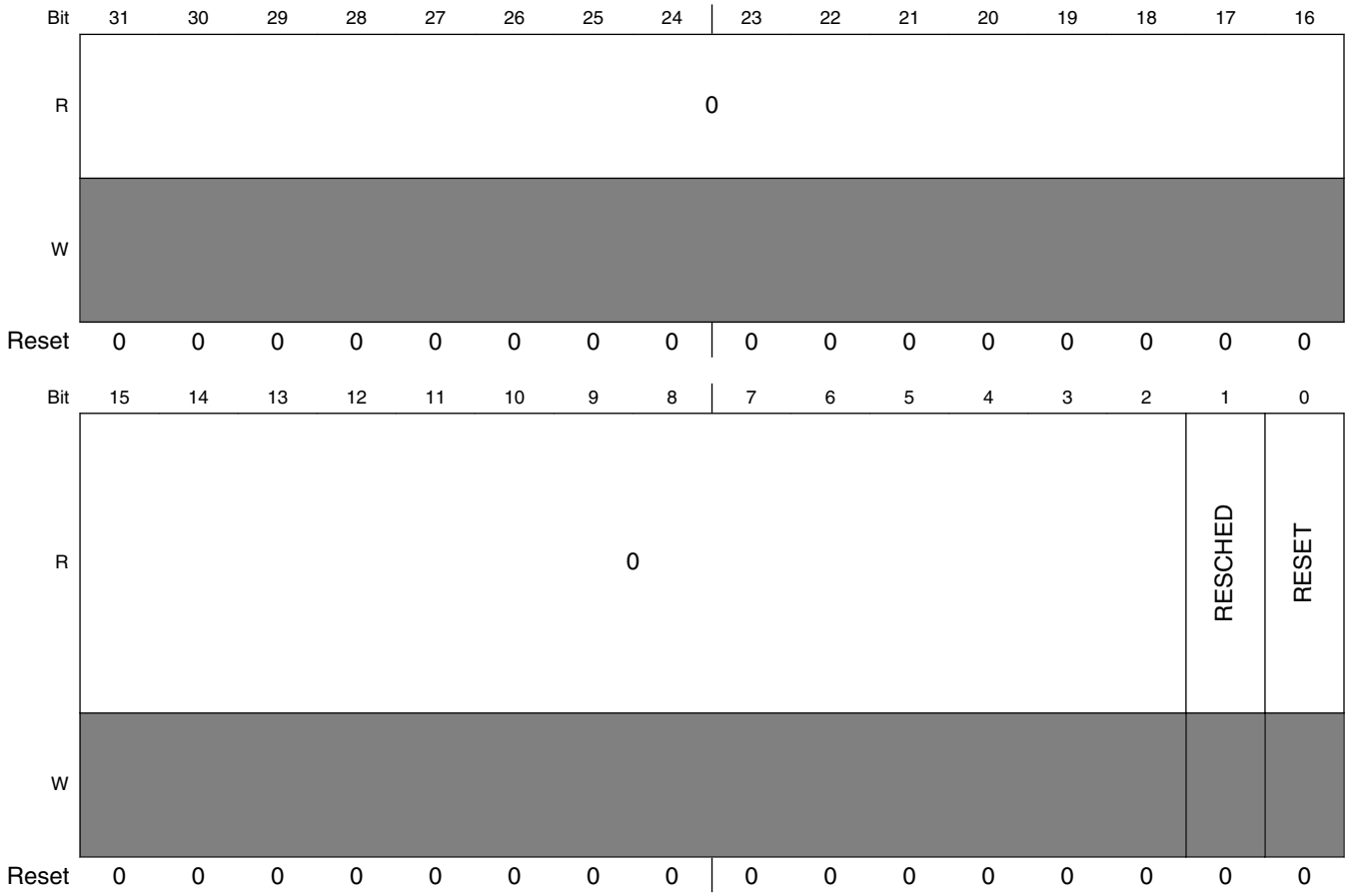
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EP																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVTPEND field descriptions

Field	Description
EP	The Channel Event Pending register contains the 32 EP[i] bits. Reading this register enables the Arm platform to determine what channels are pending after the reception of a DMA request. • Setting a bit in this register causes the SDMA to reevaluate scheduling as if a DMA request mapped on this channel had occurred. This is useful for starting up channels, so that initialization is done before awaiting the first request. The scheduler can also set bits in the EVTPEND register according to the received DMA requests. • The EP[i] bit may be cleared by the `done` instruction when running the channel *i* script. This a "write-ones" mechanism: Writing a '0' does not clear the corresponding bit.

7.2.7.9 Reset Register (SDMAARMx_RESET)

Address: Base address + 24h offset

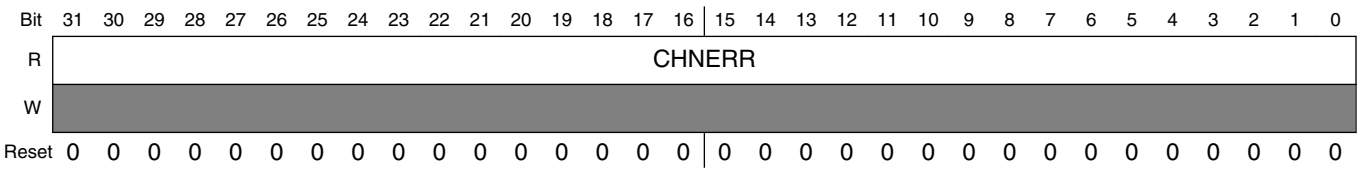


SDMAARMx_RESET field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 RESCHED	When set, this bit forces the SDMA to reschedule as if a script had executed a `done` instruction. This enables the Arm platform to recover from a runaway script on a channel by clearing its HE[i] bit via the STOP register, and then forcing a reschedule via the RESCHED bit. The RESCHED bit is cleared when the context switch starts.
0 RESET	When set, this bit causes the SDMA to be held in a software reset. The internal reset signal is held low 16 cycles; the RESET bit is automatically cleared when the internal reset signal rises.

7.2.7.10 DMA Request Error Register (SDMAARMx_EVTERR)

Address: Base address + 28h offset

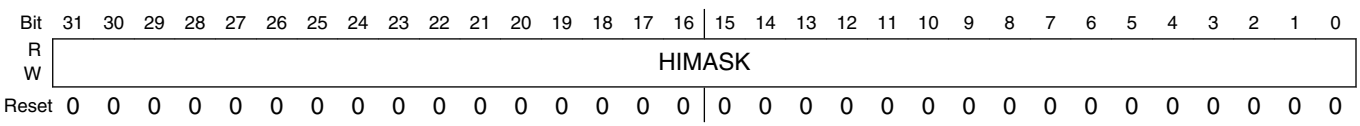


SDMAARMx_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the Arm platform when an incoming DMA request was detected and it triggers a channel that is already pending or being serviced. This probably means there is an overflow of data for that channel. • An interrupt is sent to the Arm platform if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the Arm platform or during SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set; the EVTERR[i] bit is unaffected if the Arm platform tries to set the EP[i] bit, whereas, that EP[i] bit is already set.

7.2.7.11 Channel Arm platform Interrupt Mask (SDMAARMx_INTRMASK)

Address: Base address + 2Ch offset



SDMAARMx_INTRMASK field descriptions

Field	Description
HIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit HIMASK[i] is set, the HI[i] bit is set and an interrupt is sent to the Arm platform when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

7.2.7.12 Schedule Status (SDMAARMx_PSW)

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																NCP[2:0]			NCR[4:0]					CCP[2:0]				CCR[4:0]			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_PSW field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–13 NCP[2:0]	The Next Channel Priority gives the next pending channel priority. When the priority is 0, it means there is no pending channel and the NCR value has no meaning. 0 No running channel 1 Active channel priority
12–8 NCR[4:0]	The Next Channel Register indicates the number of the next scheduled pending channel with the highest priority.
7–4 CCP[2:0]	The Current Channel Priority indicates the priority of the current active channel. When the priority is 0, no channel is running: The SDMA is idle and the CCR value has no meaning. In the case that the SDMA has finished running the channel and has entered sleep state, CCP will indicate the priority of previous running channel. 0 No running channel 1 Active channel priority
CCR[4:0]	The Current Channel Register indicates the number of the channel that is being executed by the SDMA. SDMA. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel.

7.2.7.13 DMA Request Error Register (SDMAARMx_EVTERRDBG)

Address: Base address + 34h offset

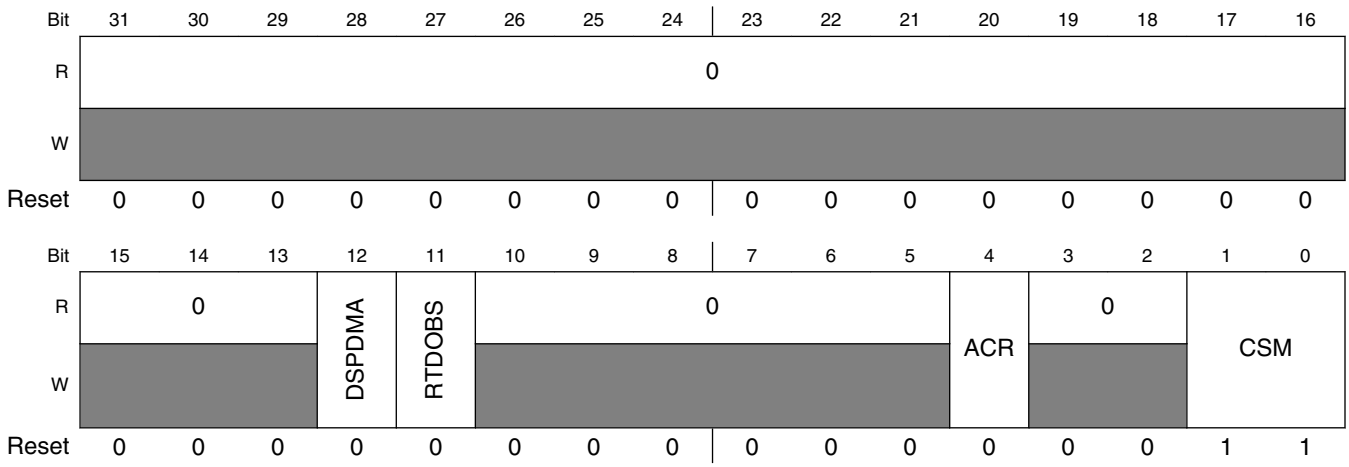
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR, except reading it does not clear its contents. This address is meant to be used in debug mode. The Arm platform OnCE may check this register value without modifying it.

7.2.7.14 Configuration Register (SDMAARMx_CONFIG)

Address: Base address + 38h offset



SDMAARMx_CONFIG field descriptions

Field	Description
31–13 Reserved	This read-only field is reserved and always has the value 0.
12 DSPDMA	This bit's function is reserved and should be configured as zero. 0 - Reset Value 1 - Reserved
11 RTDOBS	Indicates if Real-Time Debug pins are used: They do not toggle by default in order to reduce power consumption. 0 RTD pins disabled 1 RTD pins enabled
10–5 Reserved	This read-only field is reserved and always has the value 0.
4 ACR	Arm platform DMA / SDMA Core Clock Ratio. Selects the clock ratio between Arm platform DMA interfaces (burst DMA and peripheral DMA) and the internal SDMA core clock. The frequency selection is determined separately by the chip clock controller. This bit has to match the configuration of the chip clock controller that generates the clocks used in the SDMA. 0 Arm platform DMA interface frequency equals twice core frequency 1 Arm platform DMA interface frequency equals core frequency
3–2 Reserved	This read-only field is reserved and always has the value 0.
CSM	Selects the Context Switch Mode. The Arm platform has a read/write access. The SDMA cannot modify that register. The value at reset is 3, which selects the dynamic context switch by default. That register can be modified at anytime but the new context switch configuration will only be taken into account at the start of the next restore phase. NOTE: The first call to SDMA's channel 0 Bootload script after reset should use static context switch mode to ensure the context RAM for channel 0 is initialized in the channel SAVE Phase. After Channel 0 is run once, then any of the dynamic context modes can be used.

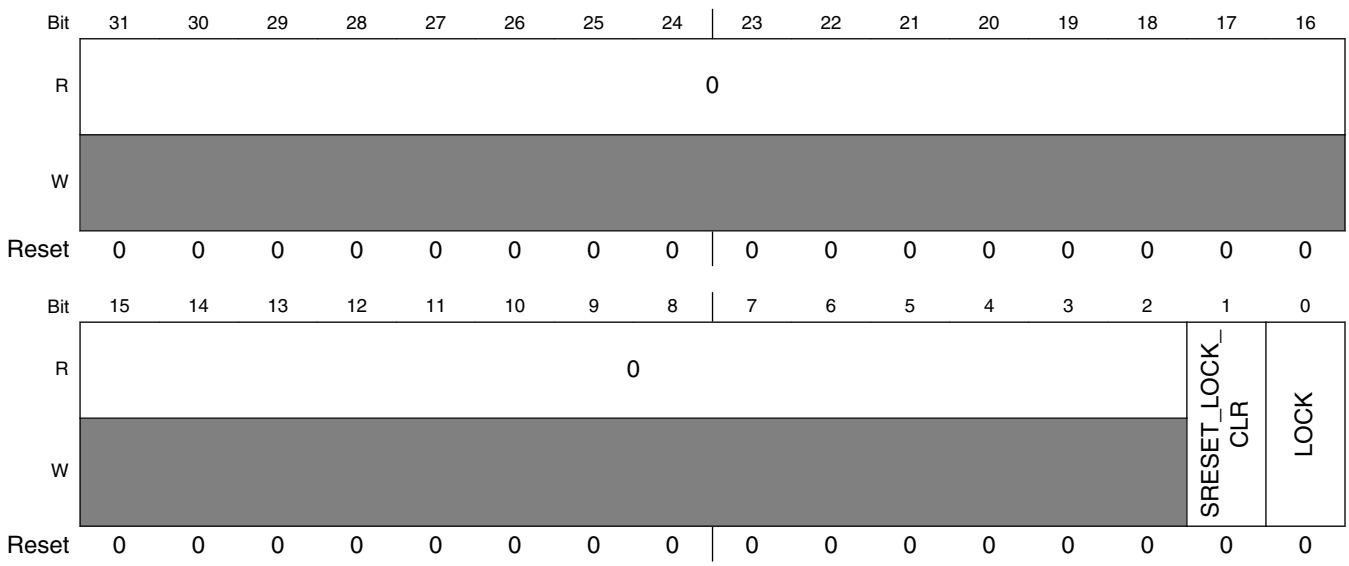
Table continues on the next page...

SDMAARMx_CONFIG field descriptions (continued)

Field	Description
	0 static 1 dynamic low power 2 dynamic with no loop 3 dynamic

7.2.7.15 SDMA LOCK (SDMAARMx_SDMA_LOCK)

Address: Base address + 3Ch offset



SDMAARMx_SDMA_LOCK field descriptions

Field	Description
31–2 Reserved	This read-only field is reserved and always has the value 0.
1 SRESET_LOCK_CLR	The SRESET_LOCK_CLR bit determine if the LOCK bit is cleared on a software reset triggered by writing to the RESET register. This bit cannot be changed if LOCK=1. SREST_LOCK_CLR is cleared by conditions that clear the LOCK bit. 0 Software Reset does not clear the LOCK bit. 1 Software Reset clears the LOCK bit.
0 LOCK	The LOCK bit is used to restrict access to update SDMA script memory through ROM channel zero scripts and through the OnCE interface under Arm platform control. The LOCK bit is set: • The SDMA_LOCK, ONCE_ENB,CH0ADDR, and ILLINSTADDR registers cannot be written. These registers can be read, but writes are ignored. • SDMA software executing out of ROM or RAM may check the LOCK bit in the LOCK register Lock Status Register (SDMACORE_SDMA_LOCK) to determine if certain operations are allowed, such as up-loading new scripts.

Table continues on the next page...

SDMAARMx_SDMA_LOCK field descriptions (continued)

Field	Description
	Once the LOCK bit is set to 1, only a reset can clear it. The LOCK bit is cleared by a hardware reset. LOCK is cleared by a software reset only if SRESET_LOCK_CLR is set. 0 LOCK disengaged. 1 LOCK enabled.

7.2.7.16 OnCE Enable (SDMAARMx_ONCE_ENB)

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															ENB
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_ONCE_ENB field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 ENB	The OnCE Enable register selects the OnCE control source: When cleared (0), the OnCE registers are accessed through the JTAG interface; when set (1), the OnCE registers may be accessed by the Arm platform through the addresses described, as follows. • After reset, the OnCE registers are accessed through the JTAG interface. • Writing a 1 to ENB enables the Arm platform to access the ONCE_* as any other SDMA control register. • When cleared (0), all the ONCE_xxx registers cannot be written. The value of ENB cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.17 OnCE Data Register (SDMAARMx_ONCE_DATA)

Address: Base address + 44h offset

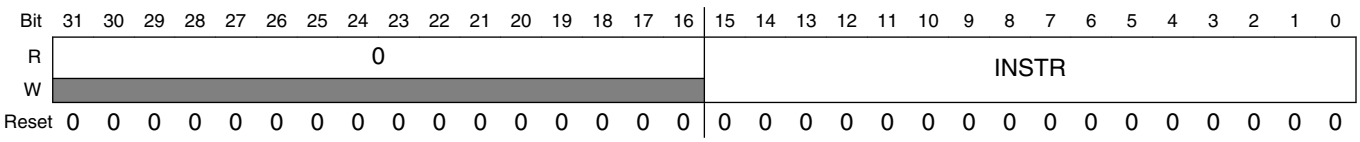
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_ONCE_DATA field descriptions

Field	Description
DATA	Data register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

7.2.7.18 OnCE Instruction Register (SDMAARMx_ONCE_INSTR)

Address: Base address + 48h offset

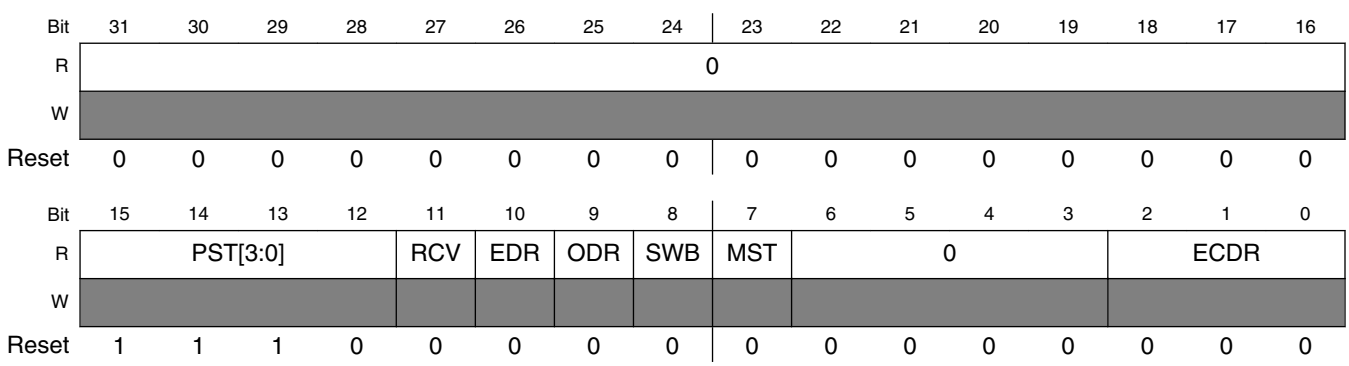


SDMAARMx_ONCE_INSTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INSTR	Instruction register of the OnCE JTAG controller. Refer to OnCE and Real-Time Debug for information on this register.

7.2.7.19 OnCE Status Register (SDMAARMx_ONCE_STAT)

Address: Base address + 4Ch offset



SDMAARMx_ONCE_STAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. Its states are as follows: • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st).

Table continues on the next page...

SDMAARMx_ONCE_STAT field descriptions (continued)

Field	Description
	• The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel: They are used when entering debug mode after reset. The reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow in Loop in Sleep 12 Debug in Sleep 13 Functional Unit in Sleep 14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 The JTAG interface controls the OnCE. 1 The Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.
ECDR	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one of the EDR bits is set (the meaning of the encoding is given below). The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addra_cond, addrb_cond, and data_cond conditions. The value of those fields is given by the EDR bits.

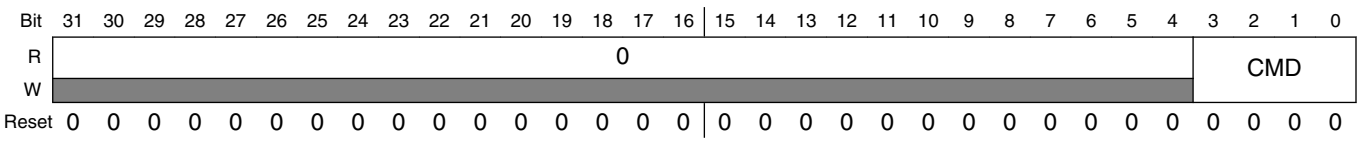
Table continues on the next page...

SDMAARMx_ONCE_STAT field descriptions (continued)

Field	Description
	0 1 matched addra_cond 1 1 matched addrb_cond 2 1 matched data_cond

7.2.7.20 OnCE Command Register (SDMAARMx_ONCE_CMD)

Address: Base address + 50h offset

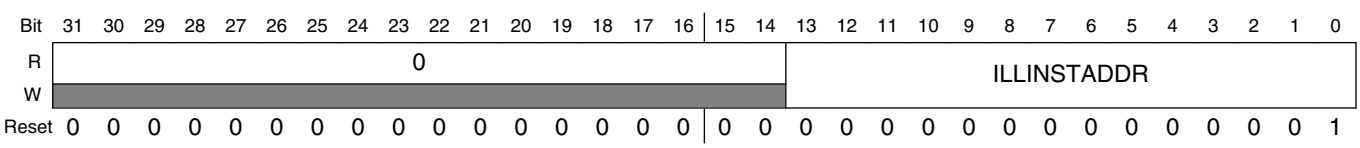


SDMAARMx_ONCE_CMD field descriptions

Field	Description
31–4 Reserved	This read-only field is reserved and always has the value 0.
CMD	Writing to this register will cause the OnCE to execute the command that is written. When needed, the ONCE_DATA and ONCE_INSTR registers should be loaded with the correct value before writing the command to that register. For a list of the OnCE commands and their usage, see OnCE and Real-Time Debug. **NOTE:** 7-15 *reserved* 0 rstatus 1 dmov 2 exec_once 3 run_core 4 exec_core 5 debug_rqst 6 rbuffer

7.2.7.21 Illegal Instruction Trap Address (SDMAARMx_ILLINSTADDR)

Address: Base address + 58h offset

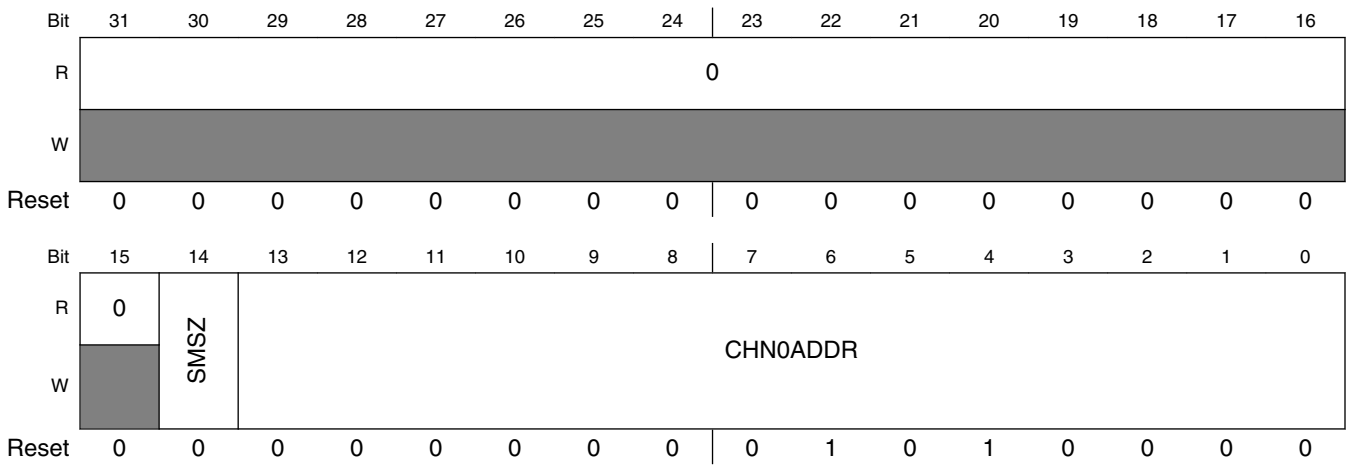


SDMAARMx_ILLINSTADDR field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
ILLINSTADDR	The Illegal Instruction Trap Address is the address where the SDMA jumps when an illegal instruction is executed. It is 0x0001 after reset. The value of ILLINSTADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.22 Channel 0 Boot Address (SDMAARMx_CHN0ADDR)

Address: Base address + 5Ch offset

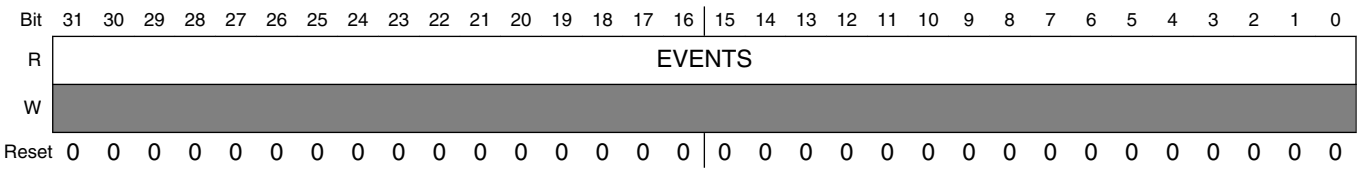


SDMAARMx_CHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. The value of SMSZ cannot be changed if the LOCK bit in the SDMA_LOCK register is set. 0 24 words per context 1 32 words per context
CHN0ADDR	This 14-bit register is used by the boot code of the SDMA. After reset, it points to the standard boot routine in ROM (channel 0 routine). By changing this address, you can perform a boot sequence with your own routine. The very first instructions of the boot code fetch the contents of this register (it is also mapped in the SDMA memory space) and jump to the given address. The reset value is 0x0050 (decimal 80). The value of CHN0ADDR cannot be changed if the LOCK bit in the SDMA_LOCK register is set.

7.2.7.23 DMA Requests (SDMAARMx_EVT_MIRROR)

Address: Base address + 60h offset

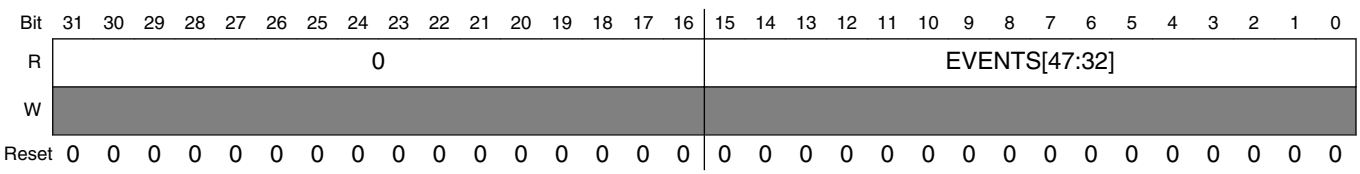


SDMAARMx_EVT_MIRROR field descriptions

Field	Description
EVENTS	This register reflects the DMA requests received by the SDMA for events 31-0. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of 32 DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR register is cleared following read access. 0 DMA request event not pending 1 DMA request event pending

7.2.7.24 DMA Requests 2 (SDMAARMx_EVT_MIRROR2)

Address: Base address + 64h offset

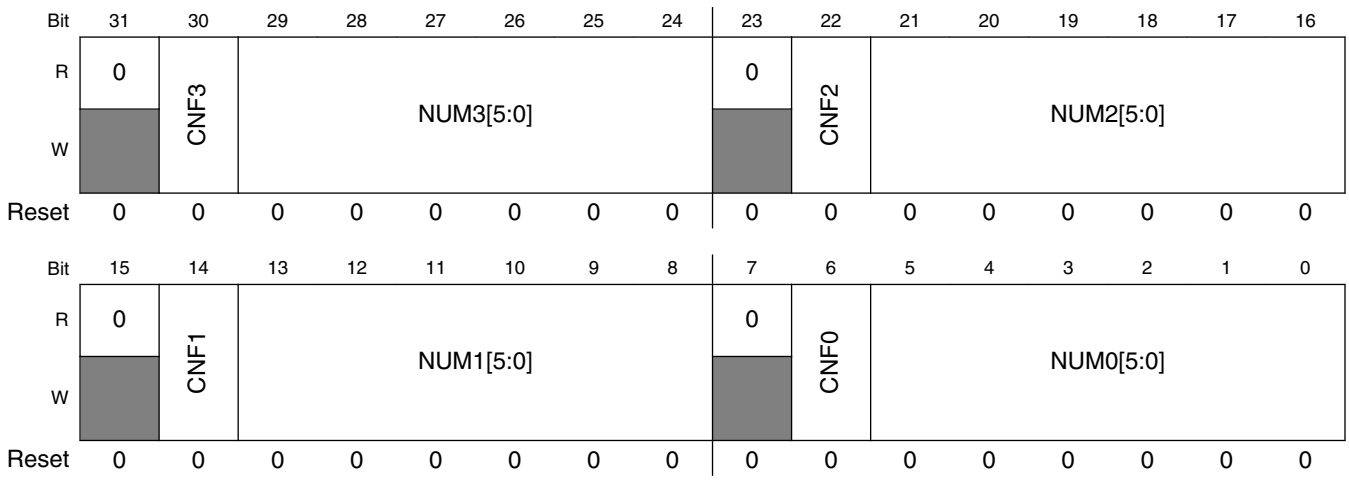


SDMAARMx_EVT_MIRROR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS[47:32]	This register reflects the DMA requests received by the SDMA for events 47-32. The Arm platform and the SDMA have a read-only access. There is one bit associated with each of DMA request events. This information may be useful during debug of the blocks that generate the DMA requests. The EVT_MIRROR2 register is cleared following read access. 0 - DMA request event not pending 1- DMA request event pending

7.2.7.25 Cross-Trigger Events Configuration Register 1 (SDMAARMx_XTRIG_CONF1)

Address: Base address + 70h offset



SDMAARMx_XTRIG_CONF1 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF3	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by the reception of a DMA request or by the starting of a channel script execution. 0 channel 1 DMA request
29–24 NUM3[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.
22 CNF2	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM2[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF1	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution.

Table continues on the next page...

SDMAARMx_XTRIG_CONF1 field descriptions (continued)

Field	Description
	0 channel 1 DMA request
13–8 NUM1[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF0	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM0[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

7.2.7.26 Cross-Trigger Events Configuration Register 2 (SDMAARMx_XTRIG_CONF2)

Address: Base address + 74h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	CNF7	NUM7[5:0]						0	CNF6	NUM6[5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0	CNF5	NUM5[5:0]						0	CNF4	NUM4[5:0]					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_XTRIG_CONF2 field descriptions

Field	Description
31 Reserved	This read-only field is reserved and always has the value 0.
30 CNF7	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request

Table continues on the next page...

SDMAARMx_XTRIG_CONF2 field descriptions (continued)

Field	Description
29–24 NUM7[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
23 Reserved	This read-only field is reserved and always has the value 0.
22 CNF6	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
21–16 NUM6[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
15 Reserved	This read-only field is reserved and always has the value 0.
14 CNF5	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution 0 channel 1 DMA request
13–8 NUM5[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.
7 Reserved	This read-only field is reserved and always has the value 0.
6 CNF4	Configuration of the SDMA event line number *i* that is connected to the cross-trigger. It determines whether the event line pulse is generated by receiving a DMA request or by starting a channel script execution. 0 channel 1 DMA request
NUM4[5:0]	Contains the number of the DMA request or channel that triggers the pulse on the cross-trigger event line number *i*.

7.2.7.27 Channel Priority Registers (SDMAARMx_SDMA_CHNPRI*n*)

Address: Base address + 100h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													CHNPRIn		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_SDMA_CHNPRI*n* field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CHNPRIn	This contains the priority of channel number *n*. Useful values are between 1 and 7; 0 is reserved by the SDMA hardware to determine when there is no pending channel. Reset value is 0, which prevents the channels from starting.

7.2.7.28 Channel Enable RAM (SDMAARMx_CHNENBL*n*)

Address: Base address + 200h offset + (4d × i), where i=0d to 47d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ENBLn																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMAARMx_CHNENBL*n* field descriptions

Field	Description
ENBLn	This 32-bit value selects the channels that are triggered by the DMA request number *n*. If ENBLn[i] is set to 1, bit EP[i] will be set when the DMA request *n* is received. These 48 32-bit registers are physically located in a RAM, with no known reset value. It is thus essential for the Arm platform to program them before any DMA request is triggered to the SDMA, otherwise an unpredictable combination of channels may be started.

7.2.7.29 SDMA DONE0 Configuration (SDMAARMx_DONE0_CONFIG)

Address: Base address + 1000h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DONE_SEL3	SW_DONE_DIS3	0	CH_SEL3					DONE_SEL2	SW_DONE_DIS2	0	CH_SEL2				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DONE_SEL1	SW_DONE_DIS1	0	CH_SEL1					DONE_SEL0	SW_DONE_DIS0	0	CH_SEL0				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

SDMAARMx_DONE0_CONFIG field descriptions

Field	Description
31 DONE_SEL3	Select Done from SW or HW for channel 3 0 HW 1 SW
30 SW_DONE_DIS3	Disable SW Done for channel 3 0 Enable 1 Disable
29 Reserved	This read-only field is reserved and always has the value 0.
28–24 CH_SEL3	Select event for channel 3 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
23 DONE_SEL2	Select Done from SW or HW for channel 2 0 HW 1 SW
22 SW_DONE_DIS2	Disable SW Done for channel 2 0 Enable 1 Disable
21 Reserved	This read-only field is reserved and always has the value 0.
20–16 CH_SEL2	Select event for channel 2 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
15 DONE_SEL1	Select Done from SW or HW for channel 1 0 HW 1 SW
14 SW_DONE_DIS1	Disable SW Done for channel 1 0 Enable 1 Disable
13 Reserved	This read-only field is reserved and always has the value 0.
12–8 CH_SEL1	Select event for channel 1 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected.

Table continues on the next page...

SDMAARMx_DONE0_CONFIG field descriptions (continued)

Field	Description
	00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
7 DONE_SEL0	Select Done from SW or HW for channel 0 0 HW 1 SW
6 SW_DONE_DIS0	Disable SW Done for channel 0 0 Enable 1 Disable
5 Reserved	This read-only field is reserved and always has the value 0.
CH_SEL0	Select event for channel 0 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31

7.2.7.30 SDMA DONE1 Configuration (SDMAARMx_DONE1_CONFIG)

Address: Base address + 1004h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DONE_SEL7	SW_DONE_DIS7	0	CH_SEL7					DONE_SEL6	SW_DONE_DIS6	0	CH_SEL6				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DONE_SEL5	SW_DONE_DIS5	0	CH_SEL5					DONE_SEL4	SW_DONE_DIS4	0	CH_SEL4				
W																
Reset	0	0	0	1	1	1	1	1	0	0	0	1	1	1	1	1

SDMAARMx_DONE1_CONFIG field descriptions

Field	Description
31 DONE_SEL7	Select Done from SW or HW for channel 7 0 HW 1 SW
30 SW_DONE_DIS7	Disable SW Done for channel 7 0 Enable 1 Disable
29 Reserved	This read-only field is reserved and always has the value 0.
28–24 CH_SEL7	Select event for channel 7 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
23 DONE_SEL6	Select Done from SW or HW for channel 6 0 HW 1 SW
22 SW_DONE_DIS6	Disable SW Done for channel 6 0 Enable 1 Disable
21 Reserved	This read-only field is reserved and always has the value 0.
20–16 CH_SEL6	Select event for channel 6 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
15 DONE_SEL5	Select Done from SW or HW for channel 5 0 HW 1 SW
14 SW_DONE_DIS5	Disable SW Done for channel 5 0 Enable 1 Disable
13 Reserved	This read-only field is reserved and always has the value 0.
12–8 CH_SEL5	Select event for channel 5 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected.

Table continues on the next page...

SDMAARMx_DONE1_CONFIG field descriptions (continued)

Field	Description
	00000 - Event 0 00001 - Event 1 ... 11111 - Event 31
7 DONE_SEL4	Select Done from SW or HW for channel 4 0 HW 1 SW
6 SW_DONE_DIS4	Disable SW Done for channel 4 0 Enable 1 Disable
5 Reserved	This read-only field is reserved and always has the value 0.
CH_SEL4	Select event for channel 4 when Done is selected from HW. HW Done will be asserted when the negative edge of selected event's Event Pending (EP) is detected. 00000 - Event 0 00001 - Event 1 ... 11111 - Event 31

7.2.8 BP Memory Map and Control Register Definitions

The following section describes SDMA control registers available to the BP.

NOTE

These registers are physically implemented in all platforms, but are not accessible when the SDMA BP control port is not connected. Reset values are calculated to allow the system to work when those registers cannot be accessed.

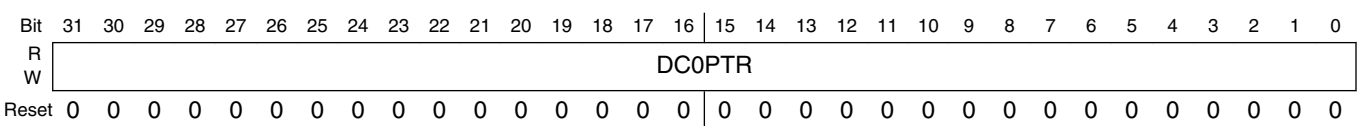
All registers are clocked with the SDMA clock (which means the SDMA clock must be running when the BP wants to access any register).

SDMABP memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0000	Channel 0 Pointer (SDMABP2_DC0PTR)	32	R/W	0000_0000h	7.2.8.1/ 1259
302C_0004	Channel Interrupts (SDMABP2_INTR)	32	w1c	0000_0000h	7.2.8.2/ 1260
302C_0008	Channel Stop/Channel Status (SDMABP2_STOP_STAT)	32	R/W	0000_0000h	7.2.8.3/ 1260
302C_000C	Channel Start (SDMABP2_DSTART)	32	R	0000_0000h	7.2.8.4/ 1260
302C_0028	DMA Request Error Register (SDMABP2_EVTERR)	32	R	0000_0000h	7.2.8.5/ 1261
302C_002C	Channel DSP Interrupt Mask (SDMABP2_INTRMASK)	32	R/W	0000_0000h	7.2.8.6/ 1261
302C_0034	DMA Request Error Register (SDMABP2_EVTERRDBG)	32	R	0000_0000h	7.2.8.7/ 1262
30BD_0000	Channel 0 Pointer (SDMABP1_DC0PTR)	32	R/W	0000_0000h	7.2.8.1/ 1259
30BD_0004	Channel Interrupts (SDMABP1_INTR)	32	w1c	0000_0000h	7.2.8.2/ 1260
30BD_0008	Channel Stop/Channel Status (SDMABP1_STOP_STAT)	32	R/W	0000_0000h	7.2.8.3/ 1260
30BD_000C	Channel Start (SDMABP1_DSTART)	32	R	0000_0000h	7.2.8.4/ 1260
30BD_0028	DMA Request Error Register (SDMABP1_EVTERR)	32	R	0000_0000h	7.2.8.5/ 1261
30BD_002C	Channel DSP Interrupt Mask (SDMABP1_INTRMASK)	32	R/W	0000_0000h	7.2.8.6/ 1261
30BD_0034	DMA Request Error Register (SDMABP1_EVTERRDBG)	32	R	0000_0000h	7.2.8.7/ 1262

7.2.8.1 Channel 0 Pointer (SDMABPx_DC0PTR)

Address: Base address + 0h offset



SDMABPx_DC0PTR field descriptions

Field	Description
DC0PTR	**Channel 0 Pointer** contains the 32-bit address, in BP memory, of the array of channel control blocks starting with the one for channel 0 (the control channel). This register should be initialized by the BP before it enables a channel (for example, channel 0). See the API document SDMA Scripts User Manual for the use of this register. The BP has a read/write access and the SDMA has a read-only access.

7.2.8.2 Channel Interrupts (SDMABPx_INTR)

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DI																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_INTR field descriptions

Field	Description
DI	The BP Interrupts register contains the 32 DI[i] bits. If any bit is set, it will cause an interrupt to the BP. • This register is a "write-ones" register to the BP. When the BP sets a bit in this register, the corresponding DI[i] bit is cleared. • The interrupt service routine should clear individual channel bits when their interrupts are serviced; failure to do so will cause continuous interrupts. • The SDMA is responsible for setting the DI[i] bit corresponding to the current channel when the corresponding `done` instruction is executed.

7.2.8.3 Channel Stop/Channel Status (SDMABPx_STOP_STAT)

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DE																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_STOP_STAT field descriptions

Field	Description
DE	This 32-bit register gives access to the BP (DSP) Enable bits, DE. There is one bit for every channel. • This register is a "write-ones" register to the BP. • When the BP writes 1 in bit *i* of this register, it clears the DE[i] and DSTART[i] bits. • Reading this register yields the current state of the DE[i] bits.

7.2.8.4 Channel Start (SDMABPx_DSTART)

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DSTART_DE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_DSTART field descriptions

Field	Description
DSTART_DE	The DSTART_DE registers are 32 bits wide with one bit for every channel. • When a bit is written to 1, it enables the corresponding channel. • Two physical registers are accessed with that address (DSTART and DE), which enables the BP to trigger a channel a second time before the first trigger was processed. • This register is a "write-ones" register to the BP. Neither DSTART[i] bit can be set while the corresponding DE[i] bit is cleared. • When the BP tries to set the DSTART[i] bit by writing a one (if the corresponding DE[i] bit is clear), the bit in the DSTART[i] register will remain cleared and the DE[i] bit will be set. If the corresponding DE[i] bit was already set, the DSTART[i] bit will be set. • The next time the SDMA channel *i* attempts to clear the DE[i] bit by means of a `done` instruction, the bit in the DSTART[i] register will be cleared and the DE[i] bit will take the old value of the DSTART[i] bit. • Reading this register yields the current state of the DSTART[i] bits. This mechanism enables the BP to pipeline two DSTART commands per channel.

7.2.8.5 DMA Request Error Register (SDMABPx_EVTERR)

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_EVTERR field descriptions

Field	Description
CHNERR	This register is used by the SDMA to warn the BP when an incoming DMA request was detected; it then triggers a channel that is already pending or being serviced, which may mean there is an overflow of data for that channel. An interrupt is sent to the BP if the corresponding channel bit is set in the INTRMASK register. • This is a "write-ones" register for the scheduler. It is only able to set the flags. The flags are cleared when the register is read by the BP or during an SDMA reset. • The CHNERR[i] bit is set when a DMA request that triggers channel *i* is received through the corresponding input pins and the EP[i] bit is already set. The EVTERR[i] bit is unaffected if the BP tries to set the EP[i] bit when that EP[i] bit is already set.

7.2.8.6 Channel DSP Interrupt Mask (SDMABPx_INTRMASK)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DIMASK																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_INTRMASK field descriptions

Field	Description
DIMASK	The Interrupt Mask Register contains 32 interrupt generation mask bits. If bit DIMASK[i] is set, the DI[i] bit is set and an interrupt is sent to the BP when a DMA request error is detected on channel *i* (for example, EVTERR[i] is set).

7.2.8.7 DMA Request Error Register (SDMABPx_EVTERRDBG)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CHNERR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMABPx_EVTERRDBG field descriptions

Field	Description
CHNERR	This register is the same as EVTERR except reading it does not clear its contents. This address is meant to be used in debug mode. The BP OnCE may check this register value without modifying it.

7.2.9 SDMA Internal (Core) Memory Map and Internal Register Definitions

The actual SDMA memory mapped registers are summarized in the following sections; for peripherals' memory maps, refer to the respective chapters.

The following definitions serve as a key for the SDMA internal register summary.

SDMACORE memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0000	Arm platform Channel 0 Pointer (SDMACORE2_MC0PTR)	32	R	0000_0000h	7.2.9.1/ 1264
302C_0008	Current Channel Pointer (SDMACORE2_CCPTR)	32	R	0000_0000h	7.2.9.2/ 1265
302C_000C	Current Channel Register (SDMACORE2_CCR)	32	R	0000_0000h	7.2.9.3/ 1265
302C_0010	Highest Pending Channel Register (SDMACORE2_NCR)	32	R	0000_0000h	7.2.9.4/ 1266

Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302C_0014	External DMA Requests Mirror (SDMACORE2_EVENTS)	32	R	0000_0000h	7.2.9.5/ 1267
302C_0018	Current Channel Priority (SDMACORE2_CCPRI)	32	R	0000_0000h	7.2.9.6/ 1268
302C_001C	Next Channel Priority (SDMACORE2_NCPRI)	32	R	0000_0000h	7.2.9.7/ 1268
302C_0020	OnCE Event Cell Counter (SDMACORE2_ECOUNT)	32	R/W	0000_0000h	7.2.9.8/ 1269
302C_0024	OnCE Event Cell Control Register (SDMACORE2_ECTL)	32	R/W	0000_0000h	7.2.9.9/ 1269
302C_0028	OnCE Event Address Register A (SDMACORE2_EAA)	32	R/W	0000_0000h	7.2.9.10/ 1271
302C_002C	OnCE Event Cell Address Register B (SDMACORE2_EAB)	32	R/W	0000_0000h	7.2.9.11/ 1271
302C_0030	OnCE Event Cell Address Mask (SDMACORE2_EAM)	32	R/W	0000_0000h	7.2.9.12/ 1271
302C_0034	OnCE Event Cell Data Register (SDMACORE2_ED)	32	R/W	0000_0000h	7.2.9.13/ 1272
302C_0038	OnCE Event Cell Data Mask (SDMACORE2_EDM)	32	R/W	0000_0000h	7.2.9.14/ 1272
302C_003C	OnCE Real-Time Buffer (SDMACORE2_RTB)	32	R/W	0000_0000h	7.2.9.15/ 1273
302C_0040	OnCE Trace Buffer (SDMACORE2_TB)	32	R	0000_0000h	7.2.9.16/ 1273
302C_0044	OnCE Status (SDMACORE2_OSTAT)	32	R	0000_0000h	7.2.9.17/ 1274
302C_0048	Channel 0 Boot Address (SDMACORE2_MCHN0ADDR)	32	R	0000_0000h	7.2.9.18/ 1276
302C_004C	ENDIAN Status Register (SDMACORE2_ENDIANNESS)	32	R	0000_0001h	7.2.9.19/ 1277
302C_0054	Lock Status Register (SDMACORE2_SDMA_LOCK)	32	R	0000_0000h	7.2.9.20/ 1278
302C_0058	External DMA Requests Mirror #2 (SDMACORE2_EVENTS2)	32	R	0000_0000h	7.2.9.21/ 1278
30BD_0000	Arm platform Channel 0 Pointer (SDMACORE1_MC0PTR)	32	R	0000_0000h	7.2.9.1/ 1264
30BD_0008	Current Channel Pointer (SDMACORE1_CCPTR)	32	R	0000_0000h	7.2.9.2/ 1265
30BD_000C	Current Channel Register (SDMACORE1_CCR)	32	R	0000_0000h	7.2.9.3/ 1265
30BD_0010	Highest Pending Channel Register (SDMACORE1_NCR)	32	R	0000_0000h	7.2.9.4/ 1266
30BD_0014	External DMA Requests Mirror (SDMACORE1_EVENTS)	32	R	0000_0000h	7.2.9.5/ 1267

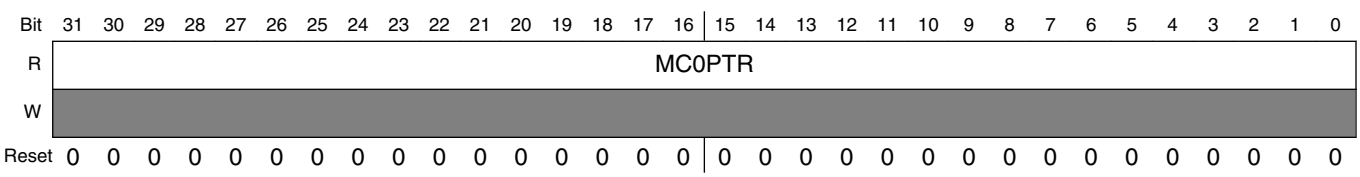
Table continues on the next page...

SDMACORE memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BD_0018	Current Channel Priority (SDMACORE1_CCPRI)	32	R	0000_0000h	7.2.9.6/ 1268
30BD_001C	Next Channel Priority (SDMACORE1_NCPRI)	32	R	0000_0000h	7.2.9.7/ 1268
30BD_0020	OnCE Event Cell Counter (SDMACORE1_ECOUNT)	32	R/W	0000_0000h	7.2.9.8/ 1269
30BD_0024	OnCE Event Cell Control Register (SDMACORE1_ECTL)	32	R/W	0000_0000h	7.2.9.9/ 1269
30BD_0028	OnCE Event Address Register A (SDMACORE1_EAA)	32	R/W	0000_0000h	7.2.9.10/ 1271
30BD_002C	OnCE Event Cell Address Register B (SDMACORE1_EAB)	32	R/W	0000_0000h	7.2.9.11/ 1271
30BD_0030	OnCE Event Cell Address Mask (SDMACORE1_EAM)	32	R/W	0000_0000h	7.2.9.12/ 1271
30BD_0034	OnCE Event Cell Data Register (SDMACORE1_ED)	32	R/W	0000_0000h	7.2.9.13/ 1272
30BD_0038	OnCE Event Cell Data Mask (SDMACORE1_EDM)	32	R/W	0000_0000h	7.2.9.14/ 1272
30BD_003C	OnCE Real-Time Buffer (SDMACORE1_RTB)	32	R/W	0000_0000h	7.2.9.15/ 1273
30BD_0040	OnCE Trace Buffer (SDMACORE1_TB)	32	R	0000_0000h	7.2.9.16/ 1273
30BD_0044	OnCE Status (SDMACORE1_OSTAT)	32	R	0000_0000h	7.2.9.17/ 1274
30BD_0048	Channel 0 Boot Address (SDMACORE1_MCHN0ADDR)	32	R	0000_0000h	7.2.9.18/ 1276
30BD_004C	ENDIAN Status Register (SDMACORE1_ENDIANNESS)	32	R	0000_0001h	7.2.9.19/ 1277
30BD_0054	Lock Status Register (SDMACORE1_SDMA_LOCK)	32	R	0000_0000h	7.2.9.20/ 1278
30BD_0058	External DMA Requests Mirror #2 (SDMACORE1_EVENTS2)	32	R	0000_0000h	7.2.9.21/ 1278

7.2.9.1 Arm platform Channel 0 Pointer (SDMACOREx_MC0PTR)

Address: Base address + 0h offset

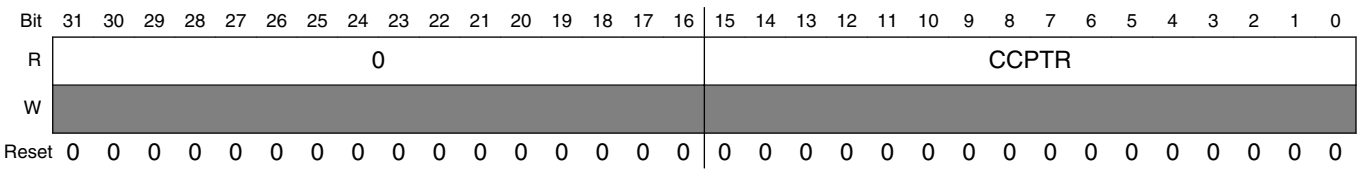


SDMACOREx_MC0PTR field descriptions

Field	Description
MC0PTR	Contains the address-in the Arm platform memory space-of the initial SDMA context and scripts that are loaded by the SDMA boot script running on channel 0.

7.2.9.2 Current Channel Pointer (SDMACOREx_CCPTR)

Address: Base address + 8h offset

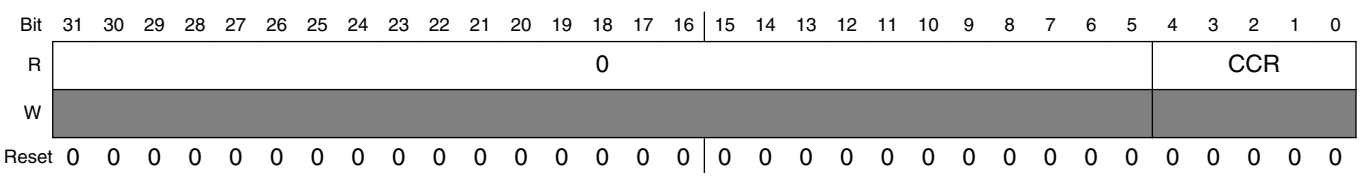


SDMACOREx_CCPTR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
CCPTR	Contains the start address of the context data for the current channel: Its value is *CONTEXT_BASE* + 24* *CCR* or *CONTEXT_BASE* + 32* *CCR* where CONTEXT_BASE = 0x0800. The value 24 or 32 is selected according to the programmed channel scratch RAM size in the register shown in Channel 0 Boot Address (SDMAARM_CHN0ADDR) .

7.2.9.3 Current Channel Register (SDMACOREx_CCR)

Address: Base address + Ch offset



SDMACOREx_CCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
CCR	Contains the number of the current running channel whose context is installed. In the case that the SDMA has finished running the channel and has entered sleep state, CCR will indicate the previous running channel. The PST bits in the OSTAT register indicate when the SDMA is in sleep state.

7.2.9.4 Highest Pending Channel Register (SDMACOREx_NCR)

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																											NCR				
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_NCR field descriptions

Field	Description
31–5 Reserved	This read-only field is reserved and always has the value 0.
NCR	Contains the number of the pending channel that the scheduler has selected to run next.

7.2.9.5 External DMA Requests Mirror (SDMACOREx_EVENTS)

NOTE

This register is very useful in the case of DMA requests that are active when a peripheral FIFO level is above the programmed watermark. The activation of the DMA request (rising edge) is detected by the SDMA logic and it can enable one or several channels. One of the channels accesses the peripheral and reads or writes a number of data that matches the watermark level (for example, if the watermark is four words, the channel reads or writes four words).

If the channel is effectively executed long after the DMA request was received, reading or writing the watermark number of data may not be sufficient to reset the DMA request (for example, if the FIFO watermark is four and at the channel execution it already contains nine pieces of data). This means no new rising edge may be detected by the SDMA, although there still remains transfers to perform. Therefore, if the channel were terminated at that time, it would not be restarted, causing potential overrun or underrun of the peripheral.

The proposed mechanism is for the channel to check this register after it has performed the "watermark" number of accesses to the peripheral. If the bit for the DMA request that triggers this channel is set, it means there is still another watermark number of data to transfer. This goes on until the bit is cleared. The same script can be used for multiple channels that require this behavior. The script can determine its channel number from the CCR register and infer the corresponding DMA request bit to check. It needs a reference table that is coherent with the request-channel matrix that the Arm platform programmed.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	EVENTS																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_EVENTS field descriptions

Field	Description
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 0-31. The EVENTS2 register displays events 32-47.

7.2.9.6 Current Channel Priority (SDMACOREx_CCPRI)

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													CCPRI		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_CCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
CCPRI	Contains the 3-bit priority of the channel whose context is installed. It is 0 when no channel is running. **NOTE:** 1-7 current channel priority 0 no running channel

7.2.9.7 Next Channel Priority (SDMACOREx_NCPRI)

Address: Base address + 1Ch offset

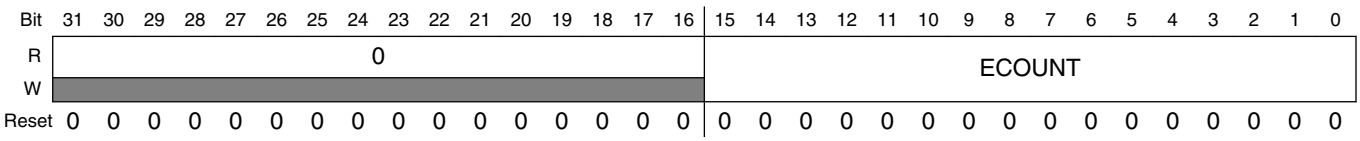
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																													NCPRI		
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_NCPRI field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
NCPRI	Contains the 3-bit priority of the channel the scheduler has selected to run next. It is 0 when no other channel is pending.

7.2.9.8 OnCE Event Cell Counter (SDMACOREx_ECOUNT)

Address: Base address + 20h offset

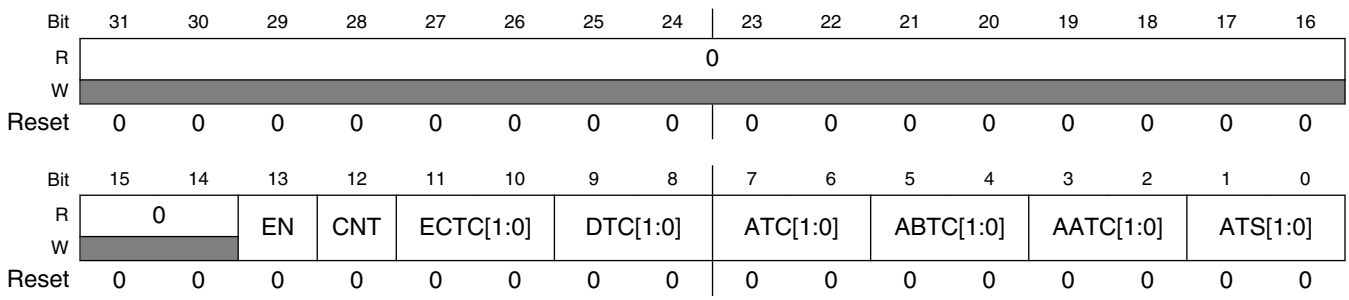


SDMACOREx_ECOUNT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
ECOUNT	The event cell counter contains the number of times minus one that an event detection must occur before generating a debug request. • This register should be written before any attempt to use the event detection counter during an event detection process. • The counter is cleared on a JTAG reset.

7.2.9.9 OnCE Event Cell Control Register (SDMACOREx_ECTL)

Address: Base address + 24h offset



SDMACOREx_ECTL field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 EN	Event Cell Enable. If the EN bit is set, the event cell is allowed to generate debug requests (the cell is awakened). If it is cleared, the event detection unit is disabled and no hardware breakpoint is generated, but matching conditions are still reflected on the emulation pin. 0 Cell is disabled. 1 Cell is enabled.
12 CNT	Event Counter Enable. The event counter enable bit determines if the cell counter is used during the event detection. In order to use the event counter during an event detection process, the event cell counter register should be loaded with a value equal to the number of times minus one that an event occurs before a debug request is sent. After every event detection, the counter is decreased. When the counter reaches

Table continues on the next page...

SDMACOREx_ECTL field descriptions (continued)

Field	Description
	the value 0, the event detection cell sends a debug request to the core. The event counter register should be written and the EN bit should be set before each new event detection process uses the event counter. 0 Counter is disabled. 1 Counter is enabled.
11–10 ECTC[1:0]	The event cell trigger condition bits select the combination of address and data matching conditions that generate the final address/data condition. During program execution, if this event cell trigger condition goes to 1, a debug request is sent to the SDMA. The EN bit must be set to enable the debug request generation. 00 address ONLY 01 data ONLY 10 address AND data 11 address OR data
9–8 DTC[1:0]	The data trigger condition bits define when data is considered matching after comparison with the data register of the event detection unit. The operations are performed on unsigned values. 00 equal 01 not equal 10 greater than 11 less than
7–6 ATC[1:0]	The address trigger condition bits select how the two address conditions (addressA and addressB) are combined to define the global address matching condition. The supported combinations are described, as follows. 00 addressA ONLY 01 addrA AND addrB 10 addrA OR addrB 11 reserved
5–4 ABTC[1:0]	The Address B Trigger Condition (ABTC) controls the operations performed by address comparator B. All operations are performed on unsigned values. This comparator B outputs the addressB condition. 00 equal 01 not equal 10 greater than 11 less than
3–2 AATC[1:0]	The Address A Trigger Condition (AATC) controls the operations performed by address comparator A. All operations are performed on unsigned values. This comparator A outputs the addressA condition. 00 equal 01 not equal 10 greater than 11 less than
ATS[1:0]	The access type select bits define the memory access type required on the SDMA memory bus. 00 read ONLY 01 write ONLY 10 read or write 11 -

7.2.9.10 OnCE Event Address Register A (SDMACORE*x*_EAA)

Address: Base address + 28h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAA															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAA field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAA	Event Cell Address Register A computes an address A condition. It is cleared on a JTAG reset.

7.2.9.11 OnCE Event Cell Address Register B (SDMACORE*x*_EAB)

Address: Base address + 2Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAB															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAB field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EAB	Event Cell Address Register B computes an address B condition. It is cleared on a JTAG reset.

7.2.9.12 OnCE Event Cell Address Mask (SDMACORE*x*_EAM)

Address: Base address + 30h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																EAM															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACORE*x*_EAM field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.

Table continues on the next page...

SDMACOREx_EAM field descriptions (continued)

Field	Description
EAM	The Event Cell Address Mask contains a user-defined address mask value. This mask is applied to the address value latched from the memory address bus before performing the address comparison. **NOTE:** There is a common address mask value for both address comparators. If bit *i* of this register is set, then bit *i* of the address value latched from the memory bus does not influence the result of the address comparison. The register is cleared on a JTAG reset.

7.2.9.13 OnCE Event Cell Data Register (SDMACOREx_ED)

Address: Base address + 34h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ED																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_ED field descriptions

Field	Description
ED	The event cell data register contains a user defined data value. This data value is an input for the data comparator which generates the data condition. It is cleared on a JTAG reset.

7.2.9.14 OnCE Event Cell Data Mask (SDMACOREx_EDM)

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EDM																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_EDM field descriptions

Field	Description
EDM	The event cell data mask register contains the user-defined data mask value. • This mask is applied to the data value latched from the memory bus before performing the data comparison. • Setting bit *i* of the event cell data mask register means that bit *i* of the data value latched from the address bus does not influence the result of the data comparison. • The data mask is cleared on a JTAG reset.

7.2.9.15 OnCE Real-Time Buffer (SDMACOREx_RTB)

Address: Base address + 3Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RTB																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_RTB field descriptions

Field	Description
RTB	The Real Time Buffer register stores and retrieves run time information without putting the SDMA in debug mode. Writing to that register triggers a pulse on a specific real-time debug pin whose connection depends on the chip implementation. The RTB value can be accessed by the OnCE under Arm platform or JTAG control using the rbuffer command.

7.2.9.16 OnCE Trace Buffer (SDMACOREx_TB)

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			TBF	TADDR											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TADDR		CHFADDR													
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_TB field descriptions

Field	Description
31–29 Reserved	This read-only field is reserved and always has the value 0.
28 TBF	The Trace Buffer Flag is set when the buffer contains the addresses of a valid change of flow. The contents of the buffer should be ignored otherwise. 0 Invalid information 1 Valid information
27–14 TADDR	The target address is the address taken after the execution of the change of flow instruction.
CHFADDR	The change of flow address is the address where the change of flow is taken when executing a change of flow instruction.

7.2.9.17 OnCE Status (SDMACOREx_OSTAT)

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PST[3:0]				RCV	EDR	ODR	SWB	MST	0				ECDR[2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SDMACOREx_OSTAT field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PST[3:0]	The Processor Status bits reflect the state of the SDMA RISC engine. • The "Program" state is the usual instruction execution cycle. • The "Data" state is inserted when there are wait-states during a load or a store on the data bus (ld or st). • The "Change of Flow" state is the second cycle of any instruction that breaks the sequence of instructions (jumps and channel-switching instructions). • The "Change of Flow in Loop" state is used when an error causes a hardware loop exit. • The "Debug" state means the SDMA is in debug mode. • The "Functional Unit" state is inserted when there are wait-states during a load or a store on the functional units bus (ldf or stf). • In "Sleep" modes, no script is running (this is the RISC engine idle state). The "after Reset" is slightly different because no context restoring phase will happen when a channel is triggered: The script located at address 0 will be executed (boot operation). • The "in Sleep" states are the same as above except they do not have any corresponding channel. They are used when entering debug mode after reset; the reason is that it is necessary to return to the "Sleep after Reset" state when leaving debug mode. 0 Program 1 Data 2 Change of Flow 3 Change of Flow in Loop 4 Debug 5 Functional Unit 6 Sleep 7 Save 8 Program in Sleep 9 Data in Sleep 10 Change of Flow in Sleep 11 Change Flow Loop Sleep 12 Debug in Sleep 13 Functional Unit in Sleep

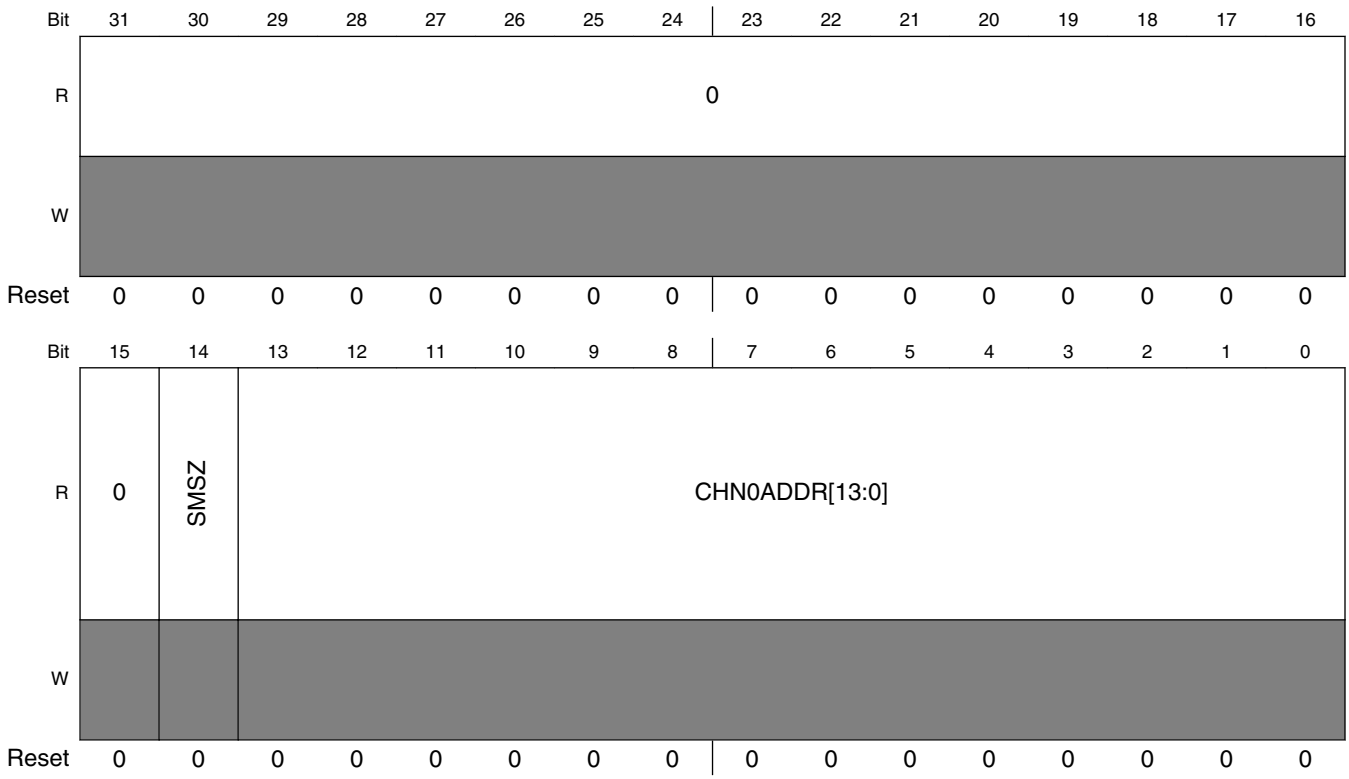
Table continues on the next page...

SDMACOREx_OSTAT field descriptions (continued)

Field	Description
	14 Sleep after Reset 15 Restore
11 RCV	After each write access to the real time buffer (RTB), the RCV bit is set. This bit is cleared after execution of an `rbuffer` command and on a JTAG reset.
10 EDR	This flag is raised when the SDMA has entered debug mode after an external debug request.
9 ODR	This flag is raised when the SDMA has entered debug mode after a OnCE debug request.
8 SWB	This flag is raised when the SDMA has entered debug mode after a software breakpoint.
7 MST	This flag is raised when the OnCE is controlled from the Arm platform peripheral interface. 0 JTAG interface controls the OnCE. 1 Arm platform peripheral interface controls the OnCE.
6–3 Reserved	This read-only field is reserved and always has the value 0.
ECDR[2:0]	Event Cell Debug Request. If the debug request comes from the event cell, the reason for entering debug mode is given by the EDR bits. The encoding of the EDR bits is useful to find out more precisely why the debug request was generated. A debug request from an event cell is generated for a specific combination of the addressA, addressB, and data conditions; the value of those fields is given by the EDR bits. If all three bits of the EDR are reset, then it did not generate any debug request. If the cell did generate a debug request, then at least one EDR bit is set; the meaning of the encoding is as follows: 0 1 matched addressA condition 1 1 matched addressB condition 2 1 matched data condition

7.2.9.18 Channel 0 Boot Address (SDMACOREx_MCHN0ADDR)

Address: Base address + 48h offset

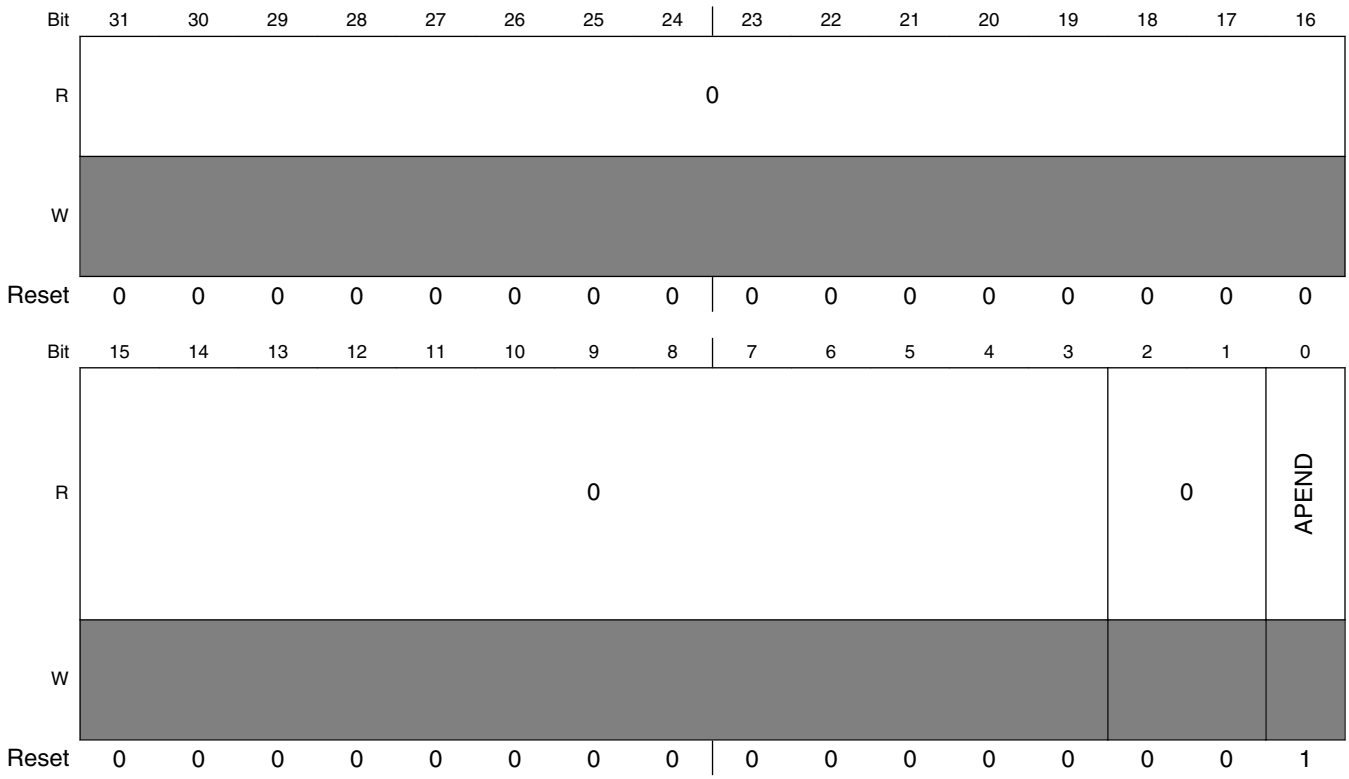


SDMACOREx_MCHN0ADDR field descriptions

Field	Description
31–15 Reserved	This read-only field is reserved and always has the value 0.
14 SMSZ	The bit 14 (Scratch Memory Size) determines if scratch memory must be available after every channel context. After reset, it is equal to 0, which defines a RAM space of 24 words for each channel. All of this area stores the channel context. By setting this bit, 32 words are reserved for every channel context, which gives eight additional words that can be used by the channel script to store any type of data. Those words are never erased by the context switching mechanism. 0 24 words per context 1 32 words per context
CHN0ADDR[13:0]	Contains the address of the channel 0 routine programmed by the Arm platform; it is loaded into a general register at the very start of the boot and the SDMA jumps to the address it contains. By default, it points to the standard boot routine in ROM.

7.2.9.19 ENDIAN Status Register (SDMACOREx_ENDIANNESS)

Address: Base address + 4Ch offset

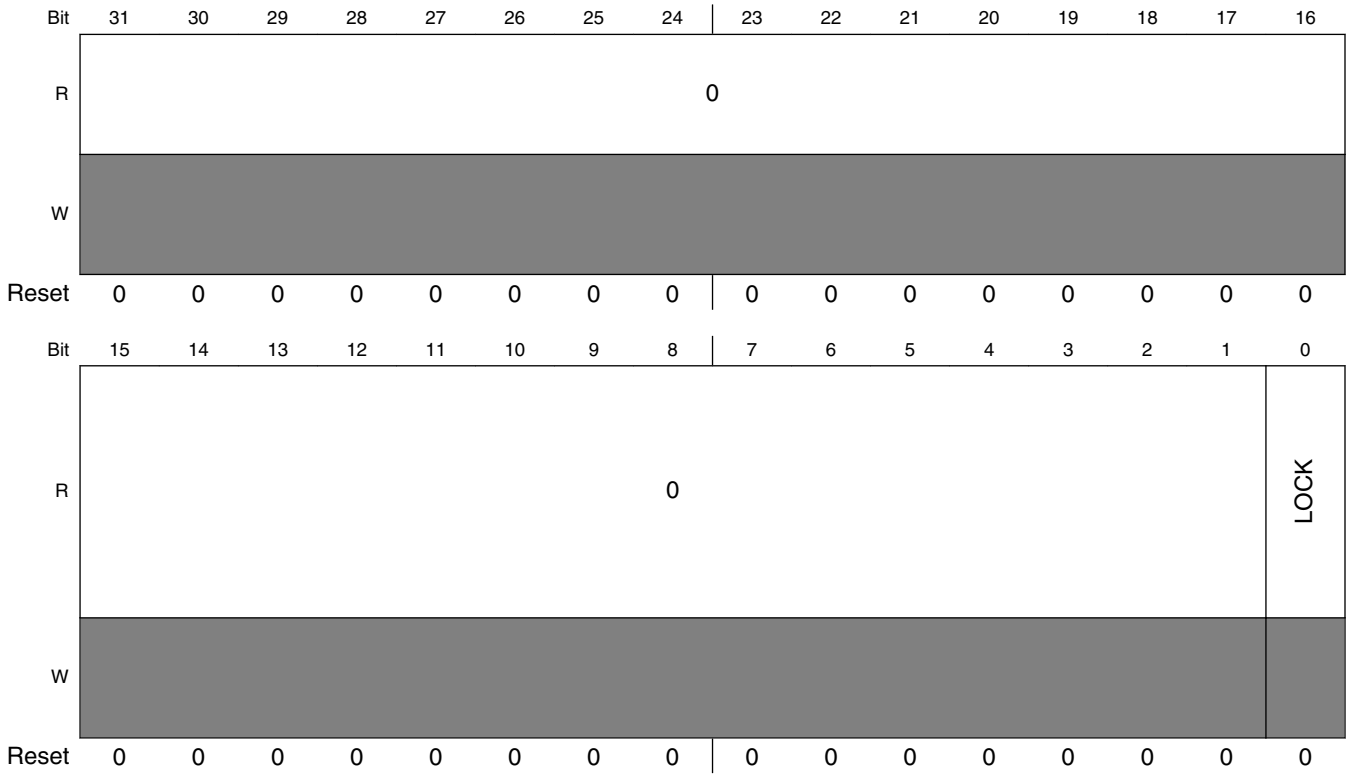


SDMACOREx_ENDIANNESS field descriptions

Field	Description
31–3 Reserved	This read-only field is reserved and always has the value 0.
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 APEND	APEND indicates the endian mode of the Peripheral and Burst DMA interfaces. This bit is tied to logic '1' indicating little-endian mode. 0 - Arm platform is in big-endian mode 1 - Arm platform is in little-endian mode

7.2.9.20 Lock Status Register (SDMACOREx_SDMA_LOCK)

Address: Base address + 54h offset

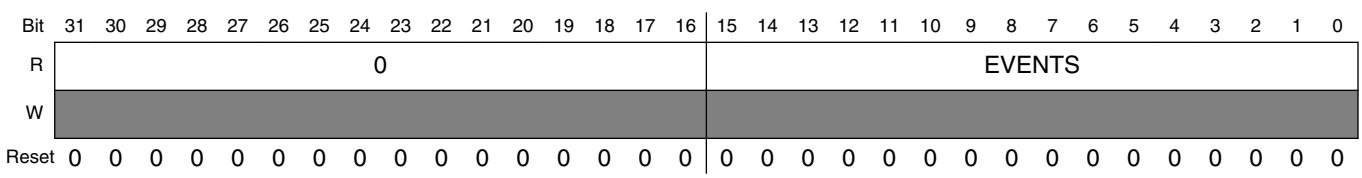


SDMACOREx_SDMA_LOCK field descriptions

Field	Description
31–1 Reserved	This read-only field is reserved and always has the value 0.
0 LOCK	The LOCK bit reports the value of the LOCK bit in the SDMA_LOCK status register. SDMA software may use this value to determine if certain operations such as loading of new scripts is allowed. 0 - LOCK bit clear 1 - LOCK bit set

7.2.9.21 External DMA Requests Mirror #2 (SDMACOREx_EVENTS2)

Address: Base address + 58h offset



SDMACOREx_EVENTS2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
EVENTS	Reflects the status of the SDMA's external DMA requests. It is meant to allow any channel to monitor the states of these SDMA inputs. This register displays EVENTS 32-47. The separate EVENTS register displays events 0-31.

7.2.10 SDMA Peripheral Registers

Refer to the respective peripherals' chapters for more information.

Chapter 8
Chip IO and Pinmux

8.1 External Signals and Pin Multiplexing

8.1.1 Overview

The chip contains a limited number of pins, most of which have multiple signal options. These signal-to-pin and pin-to-signal options are selected by the input-output multiplexer called IOMUX. The IOMUX is also used to configure other pin characteristics, such as voltage level, drive strength, and hysteresis.

The muxing options table lists the external signals grouped by the module instance, the muxing options for each signal, and the registers used to route the signal to the chosen pad.

8.1.1.1 Muxing Options

Instance	Port	Pad	Mode
ARM_PLATFORM	ARM_PLATFORM_EVENTI	SAI1_TXC	ALT4
	ARM_PLATFORM_EVENTO	SAI1_TXFS	ALT4
	ARM_PLATFORM_TRACE00	SAI1_RXD0	ALT4
	ARM_PLATFORM_TRACE01	SAI1_RXD1	ALT4
	ARM_PLATFORM_TRACE10	SAI1_TXD2	ALT4
	ARM_PLATFORM_TRACE11	SAI1_TXD3	ALT4
	ARM_PLATFORM_TRACE12	SAI1_TXD4	ALT4
	ARM_PLATFORM_TRACE13	SAI1_TXD5	ALT4
	ARM_PLATFORM_TRACE14	SAI1_TXD6	ALT4
	ARM_PLATFORM_TRACE15	SAI1_TXD7	ALT4
	ARM_PLATFORM_TRACE02	SAI1_RXD2	ALT4
	ARM_PLATFORM_TRACE03	SAI1_RXD3	ALT4

Table continues on the next page...

Instance	Port	Pad	Mode
	ARM_PLATFORM_TRACE04	SAI1_RXD4	ALT4
	ARM_PLATFORM_TRACE05	SAI1_RXD5	ALT4
	ARM_PLATFORM_TRACE06	SAI1_RXD6	ALT4
	ARM_PLATFORM_TRACE07	SAI1_RXD7	ALT4
	ARM_PLATFORM_TRACE08	SAI1_TXD0	ALT4
	ARM_PLATFORM_TRACE09	SAI1_TXD1	ALT4
	ARM_PLATFORM_TRACE_CLK	SAI1_RXFS	ALT4
	ARM_PLATFORM_TRACE_CTL	SAI1_RXC	ALT4
ANAMIX	REF_CLK_25M	GPIO1_IO01	ALT5
	REF_CLK_32K	GPIO1_IO00	ALT5
CCM	CCM_CLKO1	GPIO1_IO14	ALT6
	CCM_CLKO2	GPIO1_IO15	ALT6
	CCM_ENET_PHY_REF_CLK_ROOT	GPIO1_IO00	ALT1
	CCM_EXT_CLK1	GPIO1_IO00	ALT6
	CCM_EXT_CLK2	GPIO1_IO01	ALT6
	CCM_EXT_CLK3	GPIO1_IO06	ALT6
	CCM_EXT_CLK4	GPIO1_IO07	ALT6
	CCM_PMIC_READY	GPIO1_IO05	ALT5
	CCM_PMIC_READY	GPIO1_IO11	ALT5
	CCM_PMIC_STBY_REQ	PMIC_STBY_REQ	ALT0
	CCM_ENET_REF_CLK_ROOT (output)	ENET_TD2	ALT1
	CLK1_N	CLK1_N	No Muxing
	CLK1_P	CLK1_P	No Muxing
	CLK2_N	CLK2_N	No Muxing
	CLK2_P	CLK2_P	No Muxing
CJTAG	CJTAG_WRAPPER_MODE	JTAG_MOD	ALT0
	CJTAG_WRAPPER_TCK	JTAG_TCK	ALT0
	CJTAG_WRAPPER_TDI	JTAG_TDI	ALT0
	CJTAG_WRAPPER_TDO	JTAG_TDO	ALT0
	CJTAG_WRAPPER_TMS	JTAG_TMS	ALT0
	CJTAG_WRAPPER_TRST_B	JTAG_TRST_B	ALT0
DDR	DRAM_AC00	DRAM_AC00	No Muxing
	DRAM_AC01	DRAM_AC01	No Muxing
	DRAM_AC02	DRAM_AC02	No Muxing
	DRAM_AC03	DRAM_AC03	No Muxing
	DRAM_AC04	DRAM_AC04	No Muxing
	DRAM_AC05	DRAM_AC05	No Muxing
	DRAM_AC06	DRAM_AC06	No Muxing

Table continues on the next page...

Instance	Port	Pad	Mode
	DRAM_AC07p	DRAM_AC07	No Muxing
	DRAM_AC08	DRAM_AC08	No Muxing
	DRAM_AC09	DRAM_AC09	No Muxing
	DRAM_AC10	DRAM_AC10	No Muxing
	DRAM_AC11	DRAM_AC11	No Muxing
	DRAM_AC12	DRAM_AC12	No Muxing
	DRAM_AC13	DRAM_AC13	No Muxing
	DRAM_AC14	DRAM_AC14	No Muxing
	DRAM_AC15	DRAM_AC15	No Muxing
	DRAM_AC16	DRAM_AC16	No Muxing
	DRAM_AC17	DRAM_AC17	No Muxing
	DRAM_AC19	DRAM_AC19	No Muxing
	DRAM_AC20	DRAM_AC20	No Muxing
	DRAM_AC21	DRAM_AC21	No Muxing
	DRAM_AC22	DRAM_AC22	No Muxing
	DRAM_AC23	DRAM_AC23	No Muxing
	DRAM_AC24	DRAM_AC24	No Muxing
	DRAM_AC25	DRAM_AC25	No Muxing
	DRAM_AC26	DRAM_AC26	No Muxing
	DRAM_AC27	DRAM_AC27	No Muxing
	DRAM_AC28	DRAM_AC28	No Muxing
	DRAM_AC29	DRAM_AC29	No Muxing
	DRAM_AC30	DRAM_AC30	No Muxing
	DRAM_AC31	DRAM_AC31	No Muxing
	DRAM_AC32	DRAM_AC32	No Muxing
	DRAM_AC33	DRAM_AC33	No Muxing
	DRAM_AC34	DRAM_AC34	No Muxing
	DRAM_AC35	DRAM_AC35	No Muxing
	DRAM_AC36	DRAM_AC36	No Muxing
	DRAM_AC37	DRAM_AC37	No Muxing
	DRAM_AC38	DRAM_AC38	No Muxing
	DRAM_ALERT_N	DRAM_ALERT_N	No Muxing
	DRAM_DM0	DRAM_DM0	No Muxing
	DRAM_DM1	DRAM_DM1	No Muxing
	DRAM_DM2	DRAM_DM2	No Muxing
	DRAM_DM3	DRAM_DM3	No Muxing
	DRAM_DQ00	DRAM_DQ00	No Muxing
	DRAM_DQ01	DRAM_DQ01	No Muxing
	DRAM_DQ02	DRAM_DQ02	No Muxing
	DRAM_DQ03	DRAM_DQ03	No Muxing
	DRAM_DQ04	DRAM_DQ04	No Muxing

Table continues on the next page...

Instance	Port	Pad	Mode
	DRAM_DQ05	DRAM_DQ05	No Muxing
	DRAM_DQ06	DRAM_DQ06	No Muxing
	DRAM_DQ07	DRAM_DQ07	No Muxing
	DRAM_DQ08	DRAM_DQ08	No Muxing
	DRAM_DQ09	DRAM_DQ09	No Muxing
	DRAM_DQ10	DRAM_DQ10	No Muxing
	DRAM_DQ11	DRAM_DQ11	No Muxing
	DRAM_DQ12	DRAM_DQ12	No Muxing
	DRAM_DQ13	DRAM_DQ13	No Muxing
	DRAM_DQ14	DRAM_DQ14	No Muxing
	DRAM_DQ15	DRAM_DQ15	No Muxing
	DRAM_DQ16	DRAM_DQ16	No Muxing
	DRAM_DQ17	DRAM_DQ17	No Muxing
	DRAM_DQ18	DRAM_DQ18	No Muxing
	DRAM_DQ19	DRAM_DQ19	No Muxing
	DRAM_DQ20	DRAM_DQ20	No Muxing
	DRAM_DQ21	DRAM_DQ21	No Muxing
	DRAM_DQ22	DRAM_DQ22	No Muxing
	DRAM_DQ23	DRAM_DQ23	No Muxing
	DRAM_DQ24	DRAM_DQ24	No Muxing
	DRAM_DQ25	DRAM_DQ25	No Muxing
	DRAM_DQ26	DRAM_DQ26	No Muxing
	DRAM_DQ27	DRAM_DQ27	No Muxing
	DRAM_DQ28	DRAM_DQ28	No Muxing
	DRAM_DQ29	DRAM_DQ29	No Muxing
	DRAM_DQ30	DRAM_DQ30	No Muxing
	DRAM_DQ31	DRAM_DQ31	No Muxing
	DRAM_DQS0_N	DRAM_DQS0_N	No Muxing
	DRAM_DQS0_P	DRAM_DQS0_P	No Muxing
	DRAM_DQS1_N	DRAM_DQS1_N	No Muxing
	DRAM_DQS1_P	DRAM_DQS1_P	No Muxing
	DRAM_DQS2_N	DRAM_DQS2_N	No Muxing
	DRAM_DQS2_P	DRAM_DQS2_P	No Muxing
	DRAM_DQS3_N	DRAM_DQS3_N	No Muxing
	DRAM_DQS3_P	DRAM_DQS3_P	No Muxing
	DRAM_RESET_N	DRAM_RESET_N	No Muxing
	DRAM_VREF	DRAM_VREF	No Muxing
	DRAM_ZN	DRAM_ZN	No Muxing
ECSPI1	ECSPI1_MISO	ECSPI1_MISO	ALT0
	ECSPI1_MOSI	ECSPI1_MOSI	ALT0
	ECSPI1_SCLK	ECSPI1_SCLK	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	ECSPI1_SS0	ECSPI1_SS0	ALT0
ECSPI2	ECSPI2_MISO	ECSPI2_MISO	ALT0
	ECSPI2_MOSI	ECSPI2_MOSI	ALT0
	ECSPI2_SCLK	ECSPI2_SCLK	ALT0
	ECSPI2_SS0	ECSPI2_SS0	ALT0
ECSPI3	ECSPI3_MISO	UART2_RXD	ALT1
	ECSPI3_MOSI	UART1_TXD	ALT1
	ECSPI3_SCLK	UART1_RXD	ALT1
	ECSPI3_SS0	UART2_TXD	ALT1
ENET1	ENET1_1588_EVENT0_IN	GPIO1_IO08	ALT1
	ENET1_1588_EVENT0_OUT	GPIO1_IO09	ALT1
	ENET1_1588_EVENT1_IN	I2C2_SCL	ALT1
	ENET1_1588_EVENT1_OUT	I2C2_SDA	ALT1
	ENET1_MDC	ENET_MDC	ALT0
	ENET1_MDC	GPIO1_IO06	ALT1
	ENET1_MDC	I2C1_SCL	ALT1
	ENET1_MDIO	ENET_MDIO	ALT0
	ENET1_MDIO	GPIO1_IO07	ALT1
	ENET1_MDIO	I2C1_SDA	ALT1
	ENET1_RGMII_RD0	ENET_RD0	ALT0
	ENET1_RGMII_RD1	ENET_RD1	ALT0
	ENET1_RGMII_RD2	ENET_RD2	ALT0
	ENET1_RGMII_RD3	ENET_RD3	ALT0
	ENET1_RGMII_RX_CTL	ENET_RX_CTL	ALT0
	ENET1_RGMII_RXC	ENET_RXC	ALT0
	ENET1_RGMII_TD0	ENET_TD0	ALT0
	ENET1_RGMII_TD1	ENET_TD1	ALT0
	ENET1_RGMII_TD2	ENET_TD2	ALT0
	ENET1_RGMII_TD3	ENET_TD3	ALT0
	ENET1_RGMII_TX_CTL	ENET_TX_CTL	ALT0
	ENET1_RGMII_TXC	ENET_TXC	ALT0
	ENET1_RX_ER	ENET_RXC	ALT1
	ENET1_TX_ER	ENET_TXC	ALT1
	ENET1_TX_CLK (input)	ENET_TD2	ALT1
GPIO1	GPIO1_IO00	GPIO1_IO00	ALT0
	GPIO1_IO01	GPIO1_IO01	ALT0
	GPIO1_IO10	GPIO1_IO10	ALT0
	GPIO1_IO11	GPIO1_IO11	ALT0
	GPIO1_IO12	GPIO1_IO12	ALT0
	GPIO1_IO13	GPIO1_IO13	ALT0
	GPIO1_IO14	GPIO1_IO14	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO1_IO15	GPIO1_IO15	ALT0
	GPIO1_IO16	ENET_MDC	ALT5
	GPIO1_IO17	ENET_MDIO	ALT5
	GPIO1_IO18	ENET_TD3	ALT5
	GPIO1_IO19	ENET_TD2	ALT5
	GPIO1_IO02	GPIO1_IO02	ALT0
	GPIO1_IO20	ENET_TD1	ALT5
	GPIO1_IO21	ENET_TD0	ALT5
	GPIO1_IO22	ENET_TX_CTL	ALT5
	GPIO1_IO23	ENET_TXC	ALT5
	GPIO1_IO24	ENET_RX_CTL	ALT5
	GPIO1_IO25	ENET_RXC	ALT5
	GPIO1_IO26	ENET_RD0	ALT5
	GPIO1_IO27	ENET_RD1	ALT5
	GPIO1_IO28	ENET_RD2	ALT5
	GPIO1_IO29	ENET_RD3	ALT5
	GPIO1_IO03	GPIO1_IO03	ALT0
	GPIO1_IO04	GPIO1_IO04	ALT0
	GPIO1_IO05	GPIO1_IO05	ALT0
	GPIO1_IO06	GPIO1_IO06	ALT0
	GPIO1_IO07	GPIO1_IO07	ALT0
	GPIO1_IO08	GPIO1_IO08	ALT0
	GPIO1_IO09	GPIO1_IO09	ALT0
GPIO2	GPIO2_IO00	SD1_CLK	ALT5
	GPIO2_IO01	SD1_CMD	ALT5
	GPIO2_IO10	SD1_RESET_B	ALT5
	GPIO2_IO11	SD1_STROBE	ALT5
	GPIO2_IO12	SD2_CD_B	ALT5
	GPIO2_IO13	SD2_CLK	ALT5
	GPIO2_IO14	SD2_CMD	ALT5
	GPIO2_IO15	SD2_DATA0	ALT5
	GPIO2_IO16	SD2_DATA1	ALT5
	GPIO2_IO17	SD2_DATA2	ALT5
	GPIO2_IO18	SD2_DATA3	ALT5
	GPIO2_IO19	SD2_RESET_B	ALT5
	GPIO2_IO02	SD1_DATA0	ALT5
	GPIO2_IO20	SD2_WP	ALT5
	GPIO2_IO03	SD1_DATA1	ALT5
	GPIO2_IO04	SD1_DATA2	ALT5
	GPIO2_IO05	SD1_DATA3	ALT5
	GPIO2_IO06	SD1_DATA4	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO2_IO07	SD1_DATA5	ALT5
	GPIO2_IO08	SD1_DATA6	ALT5
	GPIO2_IO09	SD1_DATA7	ALT5
GPIO3	GPIO3_IO00	NAND_ALE	ALT5
	GPIO3_IO01	NAND_CE0_B	ALT5
	GPIO3_IO10	NAND_DATA04	ALT5
	GPIO3_IO11	NAND_DATA05	ALT5
	GPIO3_IO12	NAND_DATA06	ALT5
	GPIO3_IO13	NAND_DATA07	ALT5
	GPIO3_IO14	NAND_DQS	ALT5
	GPIO3_IO15	NAND_RE_B	ALT5
	GPIO3_IO16	NAND_READY_B	ALT5
	GPIO3_IO17	NAND_WE_B	ALT5
	GPIO3_IO18	NAND_WP_B	ALT5
	GPIO3_IO19	SAI5_RXFS	ALT5
	GPIO3_IO02	NAND_CE1_B	ALT5
	GPIO3_IO20	SAI5_RXC	ALT5
	GPIO3_IO21	SAI5_RXD0	ALT5
	GPIO3_IO22	SAI5_RXD1	ALT5
	GPIO3_IO23	SAI5_RXD2	ALT5
	GPIO3_IO24	SAI5_RXD3	ALT5
	GPIO3_IO25	SAI5_MCLK	ALT5
	GPIO3_IO03	NAND_CE2_B	ALT5
	GPIO3_IO04	NAND_CE3_B	ALT5
	GPIO3_IO05	NAND_CLE	ALT5
	GPIO3_IO06	NAND_DATA00	ALT5
	GPIO3_IO07	NAND_DATA01	ALT5
	GPIO3_IO08	NAND_DATA02	ALT5
	GPIO3_IO09	NAND_DATA03	ALT5
GPIO4	GPIO4_IO00	SAI1_RXFS	ALT5
	GPIO4_IO01	SAI1_RXC	ALT5
	GPIO4_IO10	SAI1_TXFS	ALT5
	GPIO4_IO11	SAI1_TXC	ALT5
	GPIO4_IO12	SAI1_TXD0	ALT5
	GPIO4_IO13	SAI1_TXD1	ALT5
	GPIO4_IO14	SAI1_TXD2	ALT5
	GPIO4_IO15	SAI1_TXD3	ALT5
	GPIO4_IO16	SAI1_TXD4	ALT5
	GPIO4_IO17	SAI1_TXD5	ALT5
	GPIO4_IO18	SAI1_TXD6	ALT5
	GPIO4_IO19	SAI1_TXD7	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO4_IO02	SAI1_RXD0	ALT5
	GPIO4_IO20	SAI1_MCLK	ALT5
	GPIO4_IO21	SAI2_RXFS	ALT5
	GPIO4_IO22	SAI2_RXC	ALT5
	GPIO4_IO23	SAI2_RXD0	ALT5
	GPIO4_IO24	SAI2_TXFS	ALT5
	GPIO4_IO25	SAI2_TXC	ALT5
	GPIO4_IO26	SAI2_TXD0	ALT5
	GPIO4_IO27	SAI2_MCLK	ALT5
	GPIO4_IO28	SAI3_RXFS	ALT5
	GPIO4_IO29	SAI3_RXC	ALT5
	GPIO4_IO03	SAI1_RXD1	ALT5
	GPIO4_IO30	SAI3_RXD	ALT5
	GPIO4_IO31	SAI3_TXFS	ALT5
	GPIO4_IO04	SAI1_RXD2	ALT5
	GPIO4_IO05	SAI1_RXD3	ALT5
	GPIO4_IO06	SAI1_RXD4	ALT5
	GPIO4_IO07	SAI1_RXD5	ALT5
	GPIO4_IO08	SAI1_RXD6	ALT5
	GPIO4_IO09	SAI1_RXD7	ALT5
GPIO5	GPIO5_IO00	SAI3_TXC	ALT5
	GPIO5_IO01	SAI3_TXD	ALT5
	GPIO5_IO10	ECSPI2_SCLK	ALT5
	GPIO5_IO11	ECSPI2_MOSI	ALT5
	GPIO5_IO12	ECSPI2_MISO	ALT5
	GPIO5_IO13	ECSPI2_SS0	ALT5
	GPIO5_IO14	I2C1_SCL	ALT5
	GPIO5_IO15	I2C1_SDA	ALT5
	GPIO5_IO16	I2C2_SCL	ALT5
	GPIO5_IO17	I2C2_SDA	ALT5
	GPIO5_IO18	I2C3_SCL	ALT5
	GPIO5_IO19	I2C3_SDA	ALT5
	GPIO5_IO02	SAI3_MCLK	ALT5
	GPIO5_IO20	I2C4_SCL	ALT5
	GPIO5_IO21	I2C4_SDA	ALT5
	GPIO5_IO22	UART1_RXD	ALT5
	GPIO5_IO23	UART1_TXD	ALT5
	GPIO5_IO24	UART2_RXD	ALT5
	GPIO5_IO25	UART2_TXD	ALT5
	GPIO5_IO26	UART3_RXD	ALT5
	GPIO5_IO27	UART3_TXD	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
	GPIO5_IO28	UART4_RXD	ALT5
	GPIO5_IO29	UART4_TXD	ALT5
	GPIO5_IO03	SPDIF_TX	ALT5
	GPIO5_IO04	SPDIF_RX	ALT5
	GPIO5_IO05	SPDIF_EXT_CLK	ALT5
	GPIO5_IO06	ECSPI1_SCLK	ALT5
	GPIO5_IO07	ECSPI1_MOSI	ALT5
	GPIO5_IO08	ECSPI1_MISO	ALT5
	GPIO5_IO09	ECSPI1_SS0	ALT5
GPT1	GPT1_CAPTURE1	SAI3_RXFS	ALT1
	GPT1_CAPTURE2	SAI3_RXC	ALT1
	GPT1_CLK	SAI3_TXFS	ALT1
	GPT1_COMPARE1	SAI3_RXD	ALT1
	GPT1_COMPARE2	SAI3_TXC	ALT1
	GPT1_COMPARE3	SAI3_TXD	ALT1
GPT2	GPT2_CLK	I2C3_SCL	ALT2
GPT3	GPT3_CLK	I2C3_SDA	ALT2
HDMI	HDMI_AUX_N	HDMI_AUX_N	No Muxing
	HDMI_AUX_P	HDMI_AUX_P	No Muxing
	HDMI_CEC	HDMI_CEC	No Muxing
	HDMI_DDC_SCL	HDMI_DDC_SCL	No Muxing
	HDMI_DDC_SDA	HDMI_DDC_SDA	No Muxing
	HDMI_HPD	HDMI_HPD	No Muxing
	HDMI_REFCLK_N	HDMI_REFCLK_N	No Muxing
	HDMI_REFCLK_P	HDMI_REFCLK_P	No Muxing
	HDMI_REXT	HDMI_REXT	No Muxing
	HDMI_TX_N_LN_0	HDMI_TX_N_LN_0	No Muxing
	HDMI_TX_N_LN_1	HDMI_TX_N_LN_1	No Muxing
	HDMI_TX_N_LN_2	HDMI_TX_N_LN_2	No Muxing
	HDMI_TX_N_LN_3	HDMI_TX_N_LN_3	No Muxing
	HDMI_TX_P_LN_0	HDMI_TX_P_LN_0	No Muxing
	HDMI_TX_P_LN_1	HDMI_TX_P_LN_1	No Muxing
	HDMI_TX_P_LN_2	HDMI_TX_P_LN_2	No Muxing
	HDMI_TX_P_LN_3	HDMI_TX_P_LN_3	No Muxing
I2C1	I2C1_SDA	I2C1_SDA	ALT0
	I2C1_SCL	I2C1_SCL	ALT0
I2C2	I2C2_SDA	I2C2_SDA	ALT0
	I2C2_SCL	I2C2_SCL	ALT0
I2C3	I2C3_SDA	I2C3_SDA	ALT0
	I2C3_SCL	I2C3_SCL	ALT0
I2C4	I2C4_SDA	I2C4_SDA	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	I2C4_SCL	I2C4_SCL	ALT0
M4	M4_NMI	GPIO1_IO05	ALT1
MIPI_CSI1	MIPI_CSI1_CLK_N	MIPI_CSI1_CLK_N	No Muxing
	MIPI_CSI1_CLK_P	MIPI_CSI1_CLK_P	No Muxing
	MIPI_CSI1_DATA0_N	MIPI_CSI1_D0_N	No Muxing
	MIPI_CSI1_DATA0_P	MIPI_CSI1_D0_P	No Muxing
	MIPI_CSI1_DATA1_N	MIPI_CSI1_D1_N	No Muxing
	MIPI_CSI1_DATA1_P	MIPI_CSI1_D1_P	No Muxing
	MIPI_CSI1_DATA2_N	MIPI_CSI1_D2_N	No Muxing
	MIPI_CSI1_DATA2_P	MIPI_CSI1_D2_P	No Muxing
	MIPI_CSI1_DATA3_N	MIPI_CSI1_D3_N	No Muxing
	MIPI_CSI1_DATA3_P	MIPI_CSI1_D3_P	No Muxing
MIPI_CSI2	MIPI_CSI2_CLK_N	MIPI_CSI2_CLK_N	No Muxing
	MIPI_CSI2_CLK_P	MIPI_CSI2_CLK_P	No Muxing
	MIPI_CSI2_DATA0_N	MIPI_CSI2_D0_N	No Muxing
	MIPI_CSI2_DATA0_P	MIPI_CSI2_D0_P	No Muxing
	MIPI_CSI2_DATA1_N	MIPI_CSI2_D1_N	No Muxing
	MIPI_CSI2_DATA1_P	MIPI_CSI2_D1_P	No Muxing
	MIPI_CSI2_DATA2_N	MIPI_CSI2_D2_N	No Muxing
	MIPI_CSI2_DATA2_P	MIPI_CSI2_D2_P	No Muxing
	MIPI_CSI2_DATA3_N	MIPI_CSI2_D3_N	No Muxing
	MIPI_CSI2_DATA3_P	MIPI_CSI2_D3_P	No Muxing
MIPI_DSI	MIPI_DSI_CLK_N	MIPI_DSI_CLK_N	No Muxing
	MIPI_DSI_CLK_P	MIPI_DSI_CLK_P	No Muxing
	MIPI_DSI_DATA0_N	MIPI_DSI_D0_N	No Muxing
	MIPI_DSI_DATA0_P	MIPI_DSI_D0_P	No Muxing
	MIPI_DSI_DATA1_N	MIPI_DSI_D1_N	No Muxing
	MIPI_DSI_DATA1_P	MIPI_DSI_D1_P	No Muxing
	MIPI_DSI_DATA2_N	MIPI_DSI_D2_N	No Muxing
	MIPI_DSI_DATA2_P	MIPI_DSI_D2_P	No Muxing
	MIPI_DSI_DATA3_N	MIPI_DSI_D3_N	No Muxing
	MIPI_DSI_DATA3_P	MIPI_DSI_D3_P	No Muxing
	MIPI_DSI_REXT	MIPI_DSI_REXT	No Muxing
NAND	NAND_ALE	NAND_ALE	ALT0
	NAND_CE0_B	NAND_CE0_B	ALT0
	NAND_CE1_B	NAND_CE1_B	ALT0
	NAND_CE2_B	NAND_CE2_B	ALT0
	NAND_CE3_B	NAND_CE3_B	ALT0
	NAND_CLE	NAND_CLE	ALT0
	NAND_DATA00	NAND_DATA00	ALT0
	NAND_DATA01	NAND_DATA01	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	NAND_DATA02	NAND_DATA02	ALT0
	NAND_DATA03	NAND_DATA03	ALT0
	NAND_DATA04	NAND_DATA04	ALT0
	NAND_DATA05	NAND_DATA05	ALT0
	NAND_DATA06	NAND_DATA06	ALT0
	NAND_DATA07	NAND_DATA07	ALT0
	NAND_DQS	NAND_DQS	ALT0
	NAND_RE_B	NAND_RE_B	ALT0
	NAND_READY_B	NAND_READY_B	ALT0
	NAND_WE_B	NAND_WE_B	ALT0
	NAND_WP_B	NAND_WP_B	ALT0
PCIE1	PCIE1_CLKREQ_B	I2C4_SCL	ALT2
	PCIE1_CLKREQ_B	UART4_RXD	ALT2
	PCIE1_REF_PAD_CLK_N	PCIE1_REF_PAD_CLK_N	No Muxing
	PCIE1_REF_PAD_CLK_P	PCIE1_REF_PAD_CLK_P	No Muxing
	PCIE1_RESREF	PCIE1_RESREF	No Muxing
	PCIE1_RXN_N	PCIE1_RXN_N	No Muxing
	PCIE1_RXN_P	PCIE1_RXN_P	No Muxing
	PCIE1_TXN_N	PCIE1_TXN_N	No Muxing
	PCIE1_TXN_P	PCIE1_TXN_P	No Muxing
PCIE2	PCIE2_CLKREQ_B	I2C4_SDA	ALT2
	PCIE2_CLKREQ_B	UART4_TXD	ALT2
	PCIE2_REF_PAD_CLK_N	PCIE2_REF_PAD_CLK_N	No Muxing
	PCIE2_REF_PAD_CLK_P	PCIE2_REF_PAD_CLK_P	No Muxing
	PCIE2_RESREF	PCIE2_RESREF	No Muxing
	PCIE2_RXN_N	PCIE2_RXN_N	No Muxing
	PCIE2_RXN_P	PCIE2_RXN_P	No Muxing
	PCIE2_TXN_N	PCIE2_TXN_N	No Muxing
	PCIE2_TXN_P	PCIE2_TXN_P	No Muxing
PWM1	PWM1_OUT	GPIO1_IO01	ALT1
	PWM1_OUT	I2C4_SDA	ALT1
	PWM1_OUT	SPDIF_EXT_CLK	ALT1
PWM2	PWM2_OUT	I2C4_SCL	ALT1
	PWM2_OUT	SPDIF_RX	ALT1
	PWM2_OUT	GPIO1_IO13	ALT5
PWM3	PWM3_OUT	I2C3_SDA	ALT1
	PWM3_OUT	SPDIF_TX	ALT1
	PWM3_OUT	GPIO1_IO14	ALT5
PWM4	PWM4_OUT	I2C3_SCL	ALT1
	PWM4_OUT	SAI3_MCLK	ALT1
	PWM4_OUT	GPIO1_IO15	ALT5

Table continues on the next page...

Instance	Port	Pad	Mode
QSPI	QSPI_A_DATA00	NAND_DATA00	ALT1
	QSPI_A_DATA01	NAND_DATA01	ALT1
	QSPI_A_DATA02	NAND_DATA02	ALT1
	QSPI_A_DATA03	NAND_DATA03	ALT1
	QSPI_A_DQS	NAND_DQS	ALT1
	QSPI_A_SCLK	NAND_ALE	ALT1
	QSPI_A_SS0_B	NAND_CE0_B	ALT1
	QSPI_A_SS1_B	NAND_CE1_B	ALT1
	QSPI_B_DATA00	NAND_DATA04	ALT1
	QSPI_B_DATA01	NAND_DATA05	ALT1
	QSPI_B_DATA02	NAND_DATA06	ALT1
	QSPI_B_DATA03	NAND_DATA07	ALT1
	QSPI_B_DQS	NAND_RE_B	ALT1
	QSPI_B_SCLK	NAND_CLE	ALT1
	QSPI_B_SS0_B	NAND_CE2_B	ALT1
	QSPI_B_SS1_B	NAND_CE3_B	ALT1
SAI1	SAI1_MCLK	SAI1_MCLK	ALT0
	SAI1_RX_BCLK	SAI1_RXC	ALT0
	SAI1_RX_DATA00	SAI1_RXD0	ALT0
	SAI1_RX_DATA01	SAI1_RXD1	ALT0
	SAI1_RX_DATA02	SAI1_RXD2	ALT0
	SAI1_RX_DATA03	SAI1_RXD3	ALT0
	SAI1_RX_DATA04	SAI1_RXD4	ALT0
	SAI1_RX_DATA05	SAI1_RXD5	ALT0
	SAI1_RX_DATA06	SAI1_RXD6	ALT0
	SAI1_RX_DATA07	SAI1_RXD7	ALT0
	SAI1_RX_SYNC	SAI1_RXFS	ALT0
	SAI1_RX_SYNC	SAI1_RXD5	ALT3
	SAI1_TX_BCLK	SAI1_TXC	ALT0
	SAI1_TX_BCLK	SAI5_MCLK	ALT1
	SAI1_TX_BCLK	SAI1_MCLK	ALT2
	SAI1_TX_DATA00	SAI1_TXD0	ALT0
	SAI1_TX_DATA00	SAI5_RXFS	ALT1
	SAI1_TX_DATA01	SAI1_TXD1	ALT0
	SAI1_TX_DATA01	SAI5_RXC	ALT1
	SAI1_TX_DATA02	SAI1_TXD2	ALT0
	SAI1_TX_DATA02	SAI5_RXD0	ALT1
	SAI1_TX_DATA03	SAI1_TXD3	ALT0
	SAI1_TX_DATA03	SAI5_RXD1	ALT1
	SAI1_TX_DATA04	SAI1_TXD4	ALT0
	SAI1_TX_DATA04	SAI5_RXD2	ALT1

Table continues on the next page...

Instance	Port	Pad	Mode
	SAI1_TX_DATA04	SAI1_RXD7	ALT3
	SAI1_TX_DATA05	SAI1_TXD5	ALT0
	SAI1_TX_DATA05	SAI5_RXD3	ALT1
	SAI1_TX_DATA06	SAI1_TXD6	ALT0
	SAI1_TX_DATA07	SAI1_TXD7	ALT0
	SAI1_TX_SYNC	SAI1_TXFS	ALT0
	SAI1_TX_SYNC	SAI5_RXD1	ALT2
	SAI1_TX_SYNC	SAI5_RXD2	ALT2
	SAI1_TX_SYNC	SAI5_RXD3	ALT2
	SAI1_TX_SYNC	SAI1_RXD7	ALT2
SAI2	SAI2_MCLK	SAI2_MCLK	ALT0
	SAI2_RX_BCLK	SAI2_RXC	ALT0
	SAI2_RX_DATA00	SAI2_RXD0	ALT0
	SAI2_RX_SYNC	SAI2_RXFS	ALT0
	SAI2_TX_BCLK	SAI2_TXC	ALT0
	SAI2_TX_DATA00	SAI2_TXD0	ALT0
	SAI2_TX_SYNC	SAI2_TXFS	ALT0
SAI3	SAI3_MCLK	SAI3_MCLK	ALT0
	SAI3_RX_BCLK	SAI3_RXC	ALT0
	SAI3_RX_DATA00	SAI3_RXD	ALT0
	SAI3_RX_SYNC	SAI3_RXFS	ALT0
	SAI3_TX_BCLK	SAI3_TXC	ALT0
	SAI3_TX_DATA00	SAI3_TXD	ALT0
	SAI3_TX_SYNC	SAI3_TXFS	ALT0
SAI4	SAI4_MCLK	SAI5_MCLK	ALT2
SAI5	SAI5_MCLK	SAI1_MCLK	ALT1
	SAI5_MCLK	SAI2_MCLK	ALT1
	SAI5_MCLK	SAI5_MCLK	ALT0
	SAI5_MCLK	SAI3_MCLK	ALT2
	SAI5_RX_BCLK	SAI1_RXC	ALT1
	SAI5_RX_BCLK	SAI5_RXC	ALT0
	SAI5_RX_BCLK	SAI3_RXC	ALT2
	SAI5_RX_DATA00	SAI1_RXD0	ALT1
	SAI5_RX_DATA00	SAI5_RXD0	ALT0
	SAI5_RX_DATA00	SAI3_RXD	ALT2
	SAI5_RX_DATA01	SAI1_RXD1	ALT1
	SAI5_RX_DATA01	SAI5_RXD1	ALT0
	SAI5_RX_DATA01	SAI3_TXFS	ALT2
	SAI5_RX_DATA02	SAI1_RXD2	ALT1
	SAI5_RX_DATA02	SAI5_RXD2	ALT0
	SAI5_RX_DATA02	SAI3_TXC	ALT2

Table continues on the next page...

Instance	Port	Pad	Mode
	SAI5_RX_DATA03	SAI1_RXD3	ALT1
	SAI5_RX_DATA03	SAI5_RXD3	ALT0
	SAI5_RX_DATA03	SAI3_TXD	ALT2
	SAI5_RX_SYNC	SAI1_RXFS	ALT1
	SAI5_RX_SYNC	SAI5_RXFS	ALT0
	SAI5_RX_SYNC	SAI3_RXFS	ALT2
	SAI5_TX_BCLK	SAI1_TXC	ALT1
	SAI5_TX_BCLK	SAI2_RXC	ALT1
	SAI5_TX_BCLK	SAI5_RXD2	ALT3
	SAI5_TX_DATA00	SAI1_TXD0	ALT1
	SAI5_TX_DATA00	SAI2_RXD0	ALT1
	SAI5_TX_DATA00	SAI5_RXD3	ALT3
	SAI5_TX_DATA01	SAI1_TXD1	ALT1
	SAI5_TX_DATA01	SAI2_TXFS	ALT1
	SAI5_TX_DATA02	SAI1_TXD2	ALT1
	SAI5_TX_DATA02	SAI2_TXC	ALT1
	SAI5_TX_DATA03	SAI1_TXD3	ALT1
	SAI5_TX_DATA03	SAI2_TXD0	ALT1
	SAI5_TX_SYNC	SAI1_TXFS	ALT1
	SAI5_TX_SYNC	SAI2_RXFS	ALT1
	SAI5_TX_SYNC	SAI5_RXD1	ALT3
SAI6	SAI6_MCLK	SAI1_RXD7	ALT1
	SAI6_MCLK	SAI1_TXD7	ALT1
	SAI6_RX_BCLK	SAI1_TXD4	ALT1
	SAI6_RX_BCLK	SAI1_RXD4	ALT2
	SAI6_RX_DATA00	SAI1_TXD5	ALT1
	SAI6_RX_DATA00	SAI1_RXD5	ALT2
	SAI6_RX_SYNC	SAI1_TXD6	ALT1
	SAI6_RX_SYNC	SAI1_RXD6	ALT2
	SAI6_TX_BCLK	SAI1_RXD4	ALT1
	SAI6_TX_BCLK	SAI1_TXD4	ALT2
	SAI6_TX_DATA00	SAI1_RXD5	ALT1
	SAI6_TX_DATA00	SAI1_TXD5	ALT2
	SAI6_TX_SYNC	SAI1_RXD6	ALT1
	SAI6_TX_SYNC	SAI1_TXD6	ALT2
SDMA1	SDMA1_EXT_EVENT00	GPIO1_IO03	ALT5
	SDMA1_EXT_EVENT01	GPIO1_IO04	ALT5
SJC	SJC_DE_B	GPIO1_IO02	ALT7
SDMA2	SDMA2_EXT_EVENT00	GPIO1_IO09	ALT5
	SDMA2_EXT_EVENT01	GPIO1_IO12	ALT5
SNVS	SNVS_ONOFF	ONOFF	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	SNVS_PMIC_ON_REQ	PMIC_ON_REQ	ALT0
	SNVS_POR_B	POR_B	ALT0
	SNVS_RTC	RTC	ALT0
	SNVS_RTC_RESET_B	RTC_RESET_B	ALT0
SPDIF1	SPDIF1_EXT_CLK	SPDIF_EXT_CLK	ALT0
	SPDIF1_IN	SPDIF_RX	ALT0
	SPDIF1_OUT	SPDIF_TX	ALT0
SRC	SRC_BOOT_CFG00	SAI1_RXD0	ALT6
	SRC_BOOT_CFG01	SAI1_RXD1	ALT6
	SRC_BOOT_CFG10	SAI1_TXD2	ALT6
	SRC_BOOT_CFG11	SAI1_TXD3	ALT6
	SRC_BOOT_CFG12	SAI1_TXD4	ALT6
	SRC_BOOT_CFG13	SAI1_TXD5	ALT6
	SRC_BOOT_CFG14	SAI1_TXD6	ALT6
	SRC_BOOT_CFG15	SAI1_TXD7	ALT6
	SRC_BOOT_CFG02	SAI1_RXD2	ALT6
	SRC_BOOT_CFG03	SAI1_RXD3	ALT6
	SRC_BOOT_CFG04	SAI1_RXD4	ALT6
	SRC_BOOT_CFG05	SAI1_RXD5	ALT6
	SRC_BOOT_CFG06	SAI1_RXD6	ALT6
	SRC_BOOT_CFG07	SAI1_RXD7	ALT6
	SRC_BOOT_CFG08	SAI1_TXD0	ALT6
	SRC_BOOT_CFG09	SAI1_TXD1	ALT6
	SRC_BOOT_MODE0	BOOT_MODE0	ALT0
	SRC_BOOT_MODE1	BOOT_MODE1	ALT0
UART1	UART1_CTS_B	UART3_RXD	ALT1
	UART1_RTS_B	UART3_TXD	ALT1
	UART1_RX	UART1_RXD	ALT0
	UART1_TX	UART1_TXD	ALT0
UART2	UART2_CTS_B	UART4_RXD	ALT1
	UART2_RTS_B	UART4_TXD	ALT1
	UART2_RX	UART2_RXD	ALT0
	UART2_TX	UART2_TXD	ALT0
UART3	UART3_CTS_B	ECSPI1_MISO	ALT1
	UART3_RTS_B	ECSPI1_SS0	ALT1
	UART3_RX	ECSPI1_SCLK	ALT1
	UART3_RX	UART3_RXD	ALT0
	UART3_TX	ECSPI1_MOSI	ALT1
	UART3_TX	UART3_TXD	ALT0
UART4	UART4_CTS_B	ECSPI2_MISO	ALT1
	UART4_RTS_B	ECSPI2_SS0	ALT1

Table continues on the next page...

Instance	Port	Pad	Mode
	UART4_RX	ECSPI2_SCLK	ALT1
	UART4_RX	UART4_RXD	ALT0
	UART4_TX	ECSPI2_MOSI	ALT1
	UART4_TX	UART4_TXD	ALT0
USB1	USB1_DN	USB1_DN	No Muxing
	USB1_DP	USB1_DP	No Muxing
	USB1_VBUS	USB1_VBUS	No Muxing
	USB1_ID	USB1_ID	No Muxing
	USB1_OTG_ID	GPIO1_IO10	ALT1
	USB1_OTG_OC	GPIO1_IO13	ALT1
	USB1_OTG_PWR	GPIO1_IO12	ALT1
	USB1_RESREF	USB1_RESREF	No Muxing
	USB1_RX_N	USB1_RX_N	No Muxing
	USB1_RX_P	USB1_RX_P	No Muxing
	USB1_TX_N	USB1_TX_N	No Muxing
	USB1_TX_P	USB1_TX_P	No Muxing
USB2	USB2_DN	USB2_DN	No Muxing
	USB2_DP	USB2_DP	No Muxing
	USB2_ID	USB2_ID	No Muxing
	USB2_OTG_ID	GPIO1_IO11	ALT1
	USB2_OTG_OC	GPIO1_IO15	ALT1
	USB2_OTG_PWR	GPIO1_IO14	ALT1
	USB2_RESREF	USB2_RESREF	No Muxing
	USB2_RX_N	USB2_RX_N	No Muxing
	USB2_RX_P	USB2_RX_P	No Muxing
	USB2_TX_N	USB2_TX_N	No Muxing
	USB2_TX_P	USB2_TX_P	No Muxing
	USB2_VBUS	USB2_VBUS	No Muxing
USDHC1	USDHC1_CD_B	GPIO1_IO06	ALT5
	USDHC1_CLK	SD1_CLK	ALT0
	USDHC1_CMD	SD1_CMD	ALT0
	USDHC1_DATA0	SD1_DATA0	ALT0
	USDHC1_DATA1	SD1_DATA1	ALT0
	USDHC1_DATA2	SD1_DATA2	ALT0
	USDHC1_DATA3	SD1_DATA3	ALT0
	USDHC1_DATA4	SD1_DATA4	ALT0
	USDHC1_DATA5	SD1_DATA5	ALT0
	USDHC1_DATA6	SD1_DATA6	ALT0
	USDHC1_DATA7	SD1_DATA7	ALT0
	USDHC1_RESET_B	SD1_RESET_B	ALT0
	USDHC1_STROBE	SD1_STROBE	ALT0

Table continues on the next page...

Instance	Port	Pad	Mode
	USDHC1_VSELECT	GPIO1_IO03	ALT1
	USDHC1_WP	GPIO1_IO07	ALT5
USDHC2	USDHC2_CD_B	SD2_CD_B	ALT0
	USDHC2_CLK	SD2_CLK	ALT0
	USDHC2_CMD	SD2_CMD	ALT0
	USDHC2_DATA0	SD2_DATA0	ALT0
	USDHC2_DATA1	SD2_DATA1	ALT0
	USDHC2_DATA2	SD2_DATA2	ALT0
	USDHC2_DATA3	SD2_DATA3	ALT0
	USDHC2_RESET_B	SD2_RESET_B	ALT0
	USDHC2_RESET_B	GPIO1_IO08	ALT5
	USDHC2_VSELECT	GPIO1_IO04	ALT1
	USDHC2_WP	SD2_WP	ALT0
WDOG1	WDOG1_WDOG_ANY	GPIO1_IO02	ALT5
	WDOG1_WDOG_B	GPIO1_IO02	ALT1
XTALOSC	XTALI_25M	XTALI_25M	No Muxing
	XTALI_27M	XTALI_27M	No Muxing
	XTALO_25M	XTALO_25M	No Muxing
	XTALO_27M	XTALO_27M	No Muxing

8.2 IOMUX Controller (IOMUXC)

8.2.1 Overview

The IOMUX Controller (IOMUXC), together with the IOMUX, enables the IC to share one pad to several functional blocks. This sharing is done by multiplexing the pad's input and output signals.

Every module requires a specific pad setting (such as pull up or keeper), and for each pad, there are up to 8 muxing options (called ALT modes). The pad settings parameters are controlled by the IOMUXC.

The IOMUX consists only of combinatorial logic combined from several basic IOMUX cells. Each basic IOMUX cell handles only one pad signal's muxing.

Figure 8-1 illustrates the IOMUX/IOMUXC connectivity in the system.

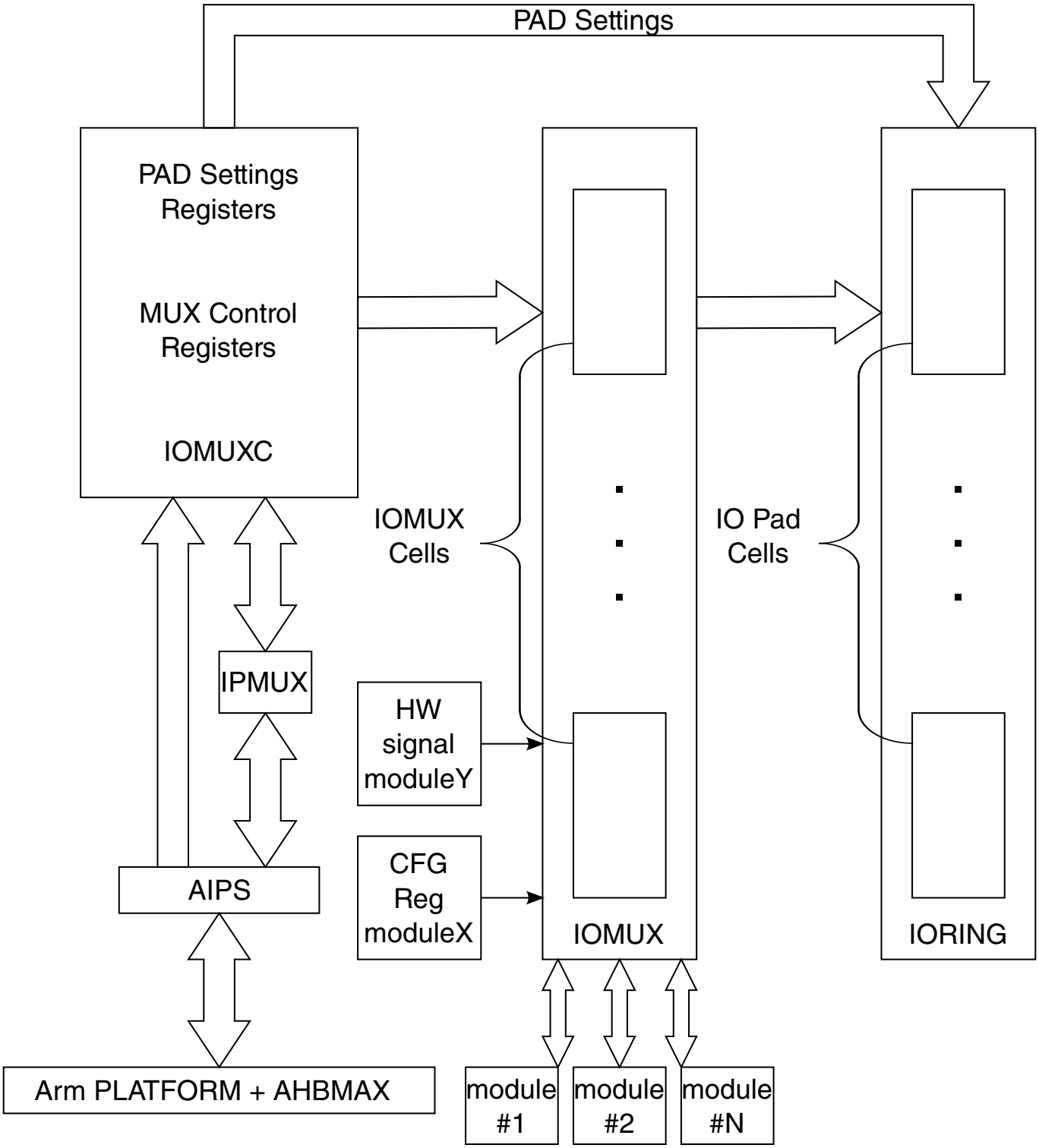


Figure 8-1. IOMUX SoC Level Block Diagram

8.2.1.1 Features

The IOMUXC features are:

- 32-bit software mux control registers (IOMUXC_SW_MUX_CTL_PAD_<PAD NAME> or IOMUXC_SW_MUX_CTL_GRP_<GROUP NAME>) to configure 1 of 8 alternate (ALT) MUX_MODE feilds of each pad or a predefined group of pads and to enable the forcing of an input path of the pad(s) (SION bit).
- 32-bit software pad control registers (IOMUXC_SW_PAD_CTL_PAD_<PAD_NAME> or IOMUXC_SW_PAD_CTL_GRP_<GROUP NAME>) to configure specific pad settings of each pad, or a predefined group of pads.
- 32-bit general purpose registers - several (GPR0 to GPR*n*) 32-bit registers according to SoC requirements for any usage.
- 32-bit input select control registers to control the input path to a module when more than one pad drives this module input.

Each SW MUX/PAD CTL IOMUXC register handles only one pad or one pad's group.

Only the minimum number of registers required by software are implemented by hardware. For example, if only ALT0 and ALT1 modes are used on Pad x then only one bit register will be generated as the MUX_MODE control field in the software mux control register of Pad x.

The software mux control registers may allow the forcing of pads to become input (input path enabled) regardless of the functional direction driven. This may be useful for loopback and GPIO data capture.

8.2.2 Clocks

The table found here describes the clock sources for IOMUXC.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 8-1. IOMUXC Clocks

Clock name	Clock Root	Description
ipt_clk_io		IO clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

8.2.3 Functional description

This section provides a complete functional description of the block.

The IOMUXC consists of two sub-blocks:

- IOMUXC_REGISTERS includes all of the IOMUXC registers (see Features).
- IOMUXC_LOGIC includes all of the IOMUXC combinatorial logic (IP interface controls, address decoder, observability muxes).

The IOMUX consists of a number (about the number of pads in the SoC) of basic iomux_cell units. If only one functional mode is required for a specific pad, there is no need for IOMUX and the signals can be connected directly from the module to the I/O. The IOMUX cell is required whenever two or more functional modes are required for a specific pad or when one functional mode and the one test mode are required.

The basic iomux_cell design, which allows two levels of HW signal control (in ALT6 and ALT7 modes - ALT7 gets highest priority) is shown in Figure 8-2.

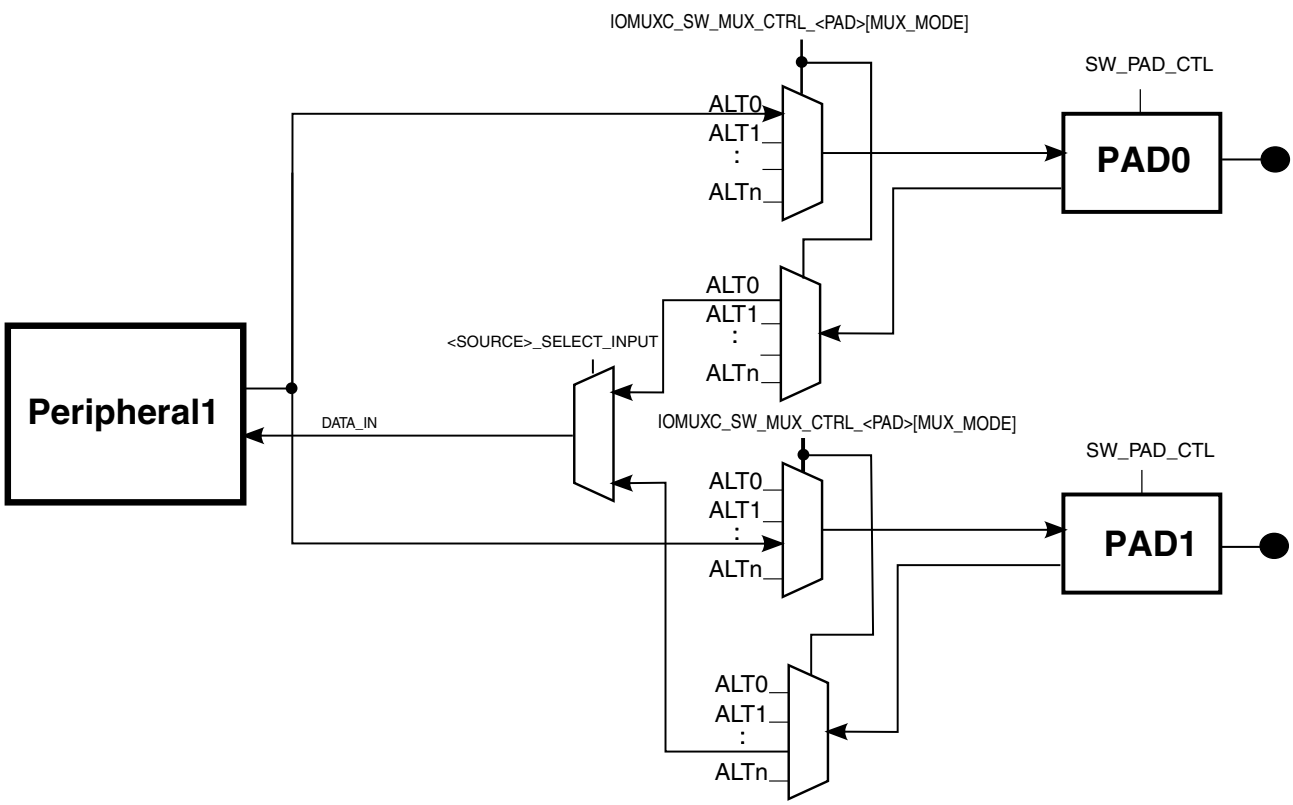


Figure 8-2. IOMUX Cell Block Diagram

8.2.3.1 ALT6 and ALT7 extended muxing modes

The ALT7 and ALT6 extended muxing modes allow any signal in the system (such as fuse, pad input, JTAG, or software register) to override any software configuration and to force the ALT6/ALT7 muxing mode.

It also allows an IOMUX software register to control a group of pads.

8.2.3.2 SW Loopback through SION bit

A limited option exists to override the default pad functionality and force the input path to be active (ipp_ibe==1'b1) regardless of the value driven by the corresponding module. This can be done by setting the SION (Software Input On) bit in the IOMUXC_SW_MUX_CTL register (when available) to "1".

Uses include:

- LoopBack - Module x drives the pad and also receives pad value as an input.
- GPIO Capture - Module x drives the pad and the value is captured by GPIO.

8.2.3.3 Daisy chain - multi pads driving same module input pin

In some cases, more than one pad may drive a single module input pin. Such cases require the addition of one more level of IOMUXing; all of these input signals are muxed, and a dedicated software controlled register controls the mux in order to select the required input path.

A module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers).

This means that a module port involved in "daisy chain" requires two software configuration commands, one for selecting the mode for this pad (programable via the IOMUXC_SW_MUX_CTL_<PAD> registers) and one for defining it as the input path (via the daisy chain registers). The daisy chain is illustrated in the figure below.

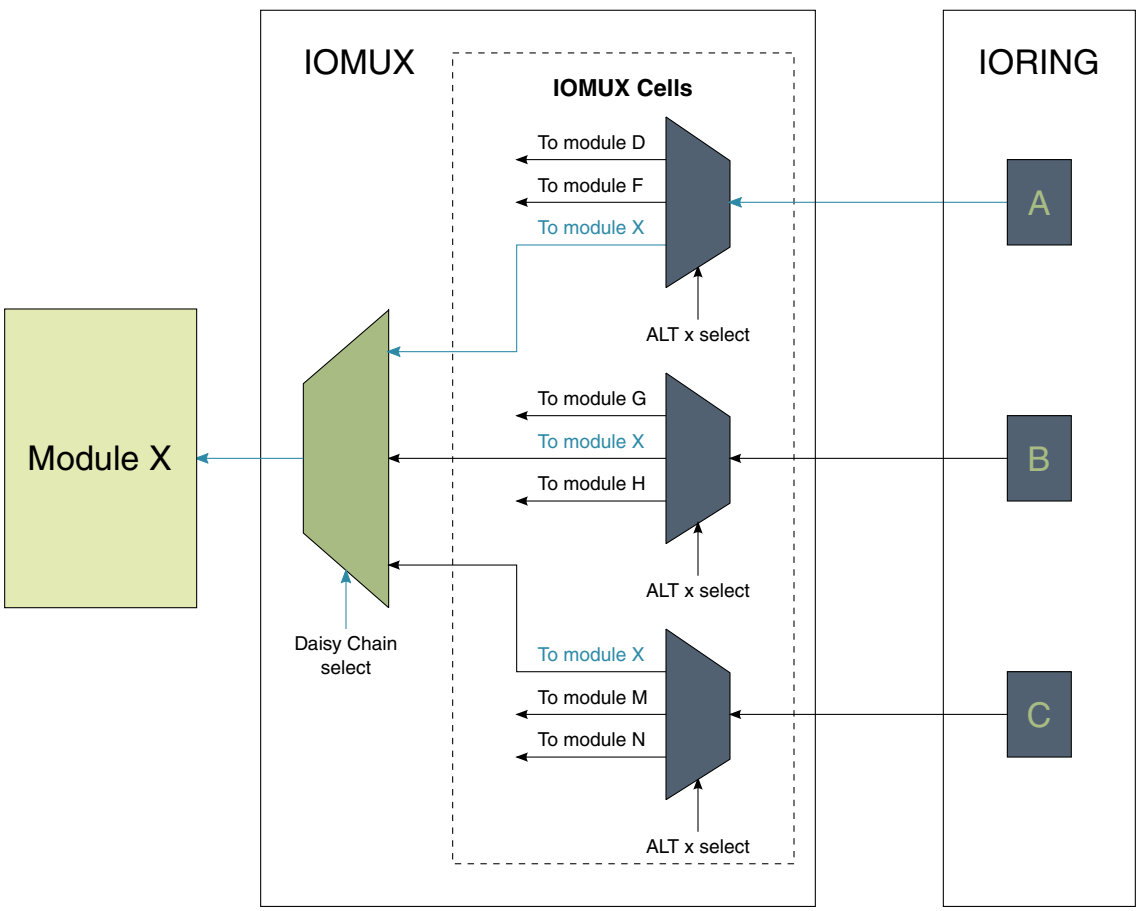


Figure 8-3. Daisy chain illustration

8.2.4 IOMUXC GPR Memory Map/Register Definition

IOMUXC_GPR memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3034_0000	GPR0 General Purpose Register (IOMUXC_GPR_GPR0)	32	R/W	0000_0000h	8.2.4.1/ 1305
3034_0004	GPR1 General Purpose Register (IOMUXC_GPR_GPR1)	32	R/W	0000_0000h	8.2.4.2/ 1306
3034_0008	GPR2 General Purpose Register (IOMUXC_GPR_GPR2)	32	R/W	0000_0000h	8.2.4.3/ 1307
3034_000C	GPR3 General Purpose Register (IOMUXC_GPR_GPR3)	32	R/W	0000_00FFh	8.2.4.4/ 1309
3034_0010	GPR4 General Purpose Register (IOMUXC_GPR_GPR4)	32	R/W	0000_0000h	8.2.4.5/ 1313
3034_0014	GPR5 General Purpose Register (IOMUXC_GPR_GPR5)	32	R/W	0000_0000h	8.2.4.6/ 1316
3034_0018	GPR6 General Purpose Register (IOMUXC_GPR_GPR6)	32	R/W	0000_0000h	8.2.4.7/ 1317
3034_001C	GPR7 General Purpose Register (IOMUXC_GPR_GPR7)	32	R/W	0000_0000h	8.2.4.8/ 1317
3034_0020	GPR8 General Purpose Register (IOMUXC_GPR_GPR8)	32	R/W	0000_0000h	8.2.4.9/ 1318
3034_0024	GPR9 General Purpose Register (IOMUXC_GPR_GPR9)	32	R/W	0000_0000h	8.2.4.10/ 1318
3034_0028	GPR10 General Purpose Register (IOMUXC_GPR_GPR10)	32	R/W	0000_0008h	8.2.4.11/ 1319
3034_002C	GPR11 General Purpose Register (IOMUXC_GPR_GPR11)	32	R/W	0000_0000h	8.2.4.12/ 1320
3034_0030	GPR12 General Purpose Register (IOMUXC_GPR_GPR12)	32	R/W	0000_4400h	8.2.4.13/ 1322
3034_0034	GPR13 General Purpose Register (IOMUXC_GPR_GPR13)	32	R/W	0000_0000h	8.2.4.14/ 1324
3034_0038	GPR14 General Purpose Register (IOMUXC_GPR_GPR14)	32	R/W	4940_9100h	8.2.4.15/ 1326
3034_003C	GPR15 General Purpose Register (IOMUXC_GPR_GPR15)	32	R/W	6188_FFFFh	8.2.4.16/ 1327
3034_0040	GPR16 General Purpose Register (IOMUXC_GPR_GPR16)	32	R/W	4940_9100h	8.2.4.17/ 1328
3034_0044	GPR17 General Purpose Register (IOMUXC_GPR_GPR17)	32	R/W	6188_FFFFh	8.2.4.18/ 1330
3034_0048	GPR18 General Purpose Register (IOMUXC_GPR_GPR18)	32	R/W	0000_0000h	8.2.4.19/ 1330
3034_004C	GPR19 General Purpose Register (IOMUXC_GPR_GPR19)	32	R	0000_0000h	8.2.4.20/ 1331

Table continues on the next page...

IOMUXC_GPR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3034_0050	GPR20 General Purpose Register (IOMUXC_GPR_GPR20)	32	R/W	0000_0000h	8.2.4.21/ 1331
3034_0054	GPR21 General Purpose Register (IOMUXC_GPR_GPR21)	32	R/W	0000_0000h	8.2.4.22/ 1332
3034_0058	GPR22 General Purpose Register (IOMUXC_GPR_GPR22)	32	R	0000_0000h	8.2.4.23/ 1332
3034_005C	GPR23 General Purpose Register (IOMUXC_GPR_GPR23)	32	R	0000_0000h	8.2.4.24/ 1333
3034_0060	GPR24 General Purpose Register (IOMUXC_GPR_GPR24)	32	R	0000_0000h	8.2.4.25/ 1335
3034_0064	GPR25 General Purpose Register (IOMUXC_GPR_GPR25)	32	R	0000_0000h	8.2.4.26/ 1338
3034_0068	GPR26 General Purpose Register (IOMUXC_GPR_GPR26)	32	R	0000_0000h	8.2.4.27/ 1338
3034_006C	GPR27 General Purpose Register (IOMUXC_GPR_GPR27)	32	R	0000_0000h	8.2.4.28/ 1339
3034_0070	GPR28 General Purpose Register (IOMUXC_GPR_GPR28)	32	R	0000_0000h	8.2.4.29/ 1339
3034_0074	GPR29 General Purpose Register (IOMUXC_GPR_GPR29)	32	R/W	0000_0000h	8.2.4.30/ 1340
3034_0078	GPR30 General Purpose Register (IOMUXC_GPR_GPR30)	32	R	0000_0000h	8.2.4.31/ 1341
3034_007C	GPR31 General Purpose Register (IOMUXC_GPR_GPR31)	32	R	0000_0000h	8.2.4.32/ 1342
3034_0080	GPR32 General Purpose Register (IOMUXC_GPR_GPR32)	32	R	0000_0000h	8.2.4.33/ 1344
3034_0084	GPR33 General Purpose Register (IOMUXC_GPR_GPR33)	32	R	0000_0000h	8.2.4.34/ 1345
3034_0088	GPR34 General Purpose Register (IOMUXC_GPR_GPR34)	32	R/W	0000_0800h	8.2.4.35/ 1346
3034_008C	GPR35 General Purpose Register (IOMUXC_GPR_GPR35)	32	R	0000_0000h	8.2.4.36/ 1348
3034_0090	GPR36 General Purpose Register (IOMUXC_GPR_GPR36)	32	R/W	0000_0000h	8.2.4.37/ 1351
3034_0094	GPR37 General Purpose Register (IOMUXC_GPR_GPR37)	32	R	0000_0000h	8.2.4.38/ 1352
3034_0098	GPR38 General Purpose Register (IOMUXC_GPR_GPR38)	32	R	0000_0000h	8.2.4.39/ 1354
3034_009C	GPR39 General Purpose Register (IOMUXC_GPR_GPR39)	32	R	0000_0000h	8.2.4.40/ 1356
3034_00A0	GPR40 General Purpose Register (IOMUXC_GPR_GPR40)	32	R	0000_0000h	8.2.4.41/ 1358
3034_00A4	GPR41 General Purpose Register (IOMUXC_GPR_GPR41)	32	R/W	0000_0800h	8.2.4.42/ 1360

Table continues on the next page...

IOMUXC_GPR memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3034_00A8	GPR42 General Purpose Register (IOMUXC_GPR_GPR42)	32	R	0000_0000h	8.2.4.43/ 1362
3034_00AC	GPR43 General Purpose Register (IOMUXC_GPR_GPR43)	32	R/W	0000_0000h	8.2.4.44/ 1365
3034_00B0	GPR44 General Purpose Register (IOMUXC_GPR_GPR44)	32	R	0000_0000h	8.2.4.45/ 1366
3034_00B4	GPR45 General Purpose Register (IOMUXC_GPR_GPR45)	32	R	0000_0000h	8.2.4.46/ 1368
3034_00B8	GPR46 General Purpose Register (IOMUXC_GPR_GPR46)	32	R	0000_0000h	8.2.4.47/ 1370
3034_00BC	GPR47 General Purpose Register (IOMUXC_GPR_GPR47)	32	R	0000_0000h	8.2.4.48/ 1372

8.2.4.1 GPR0 General Purpose Register (IOMUXC_GPR_GPR0)

GPR Register

Address: 3034_0000h base + 0h offset = 3034_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

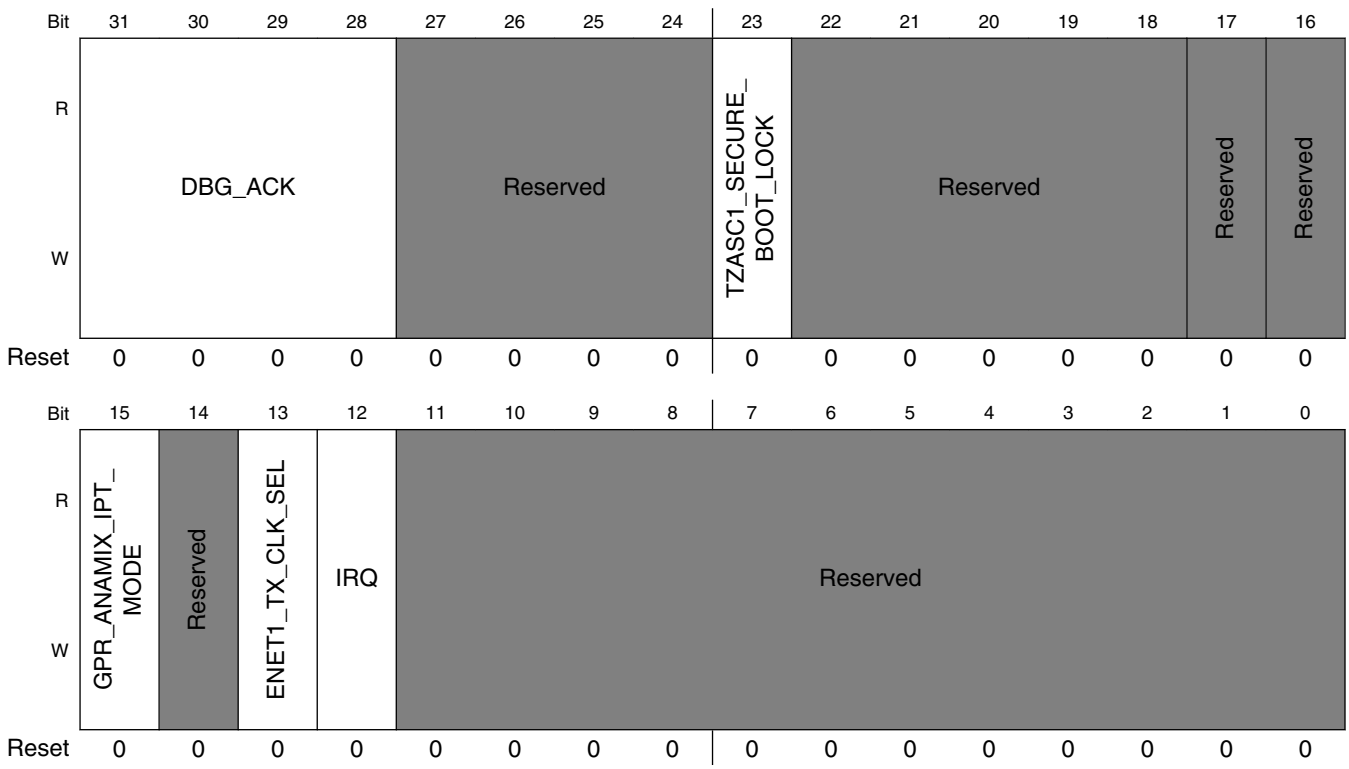
IOMUXC_GPR_GPR0 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.2 GPR1 General Purpose Register (IOMUXC_GPR_GPR1)

GPR Register

Address: 3034_0000h base + 4h offset = 3034_0004h



IOMUXC_GPR_GPR1 field descriptions

Field	Description
31–28 DBG_ACK	Debug Acknowledge
27–24 -	This field is reserved. Reserved
23 TZASC1_ SECURE_ BOOT_LOCK	TZASC-1 Secure Boot Lock 0 Secure boot lock is disabled 1 Secure boot lock is enabled
22–18 -	This field is reserved. Reserved
17 -	This field is reserved.
16 -	This field is reserved. Reserved

Table continues on the next page...

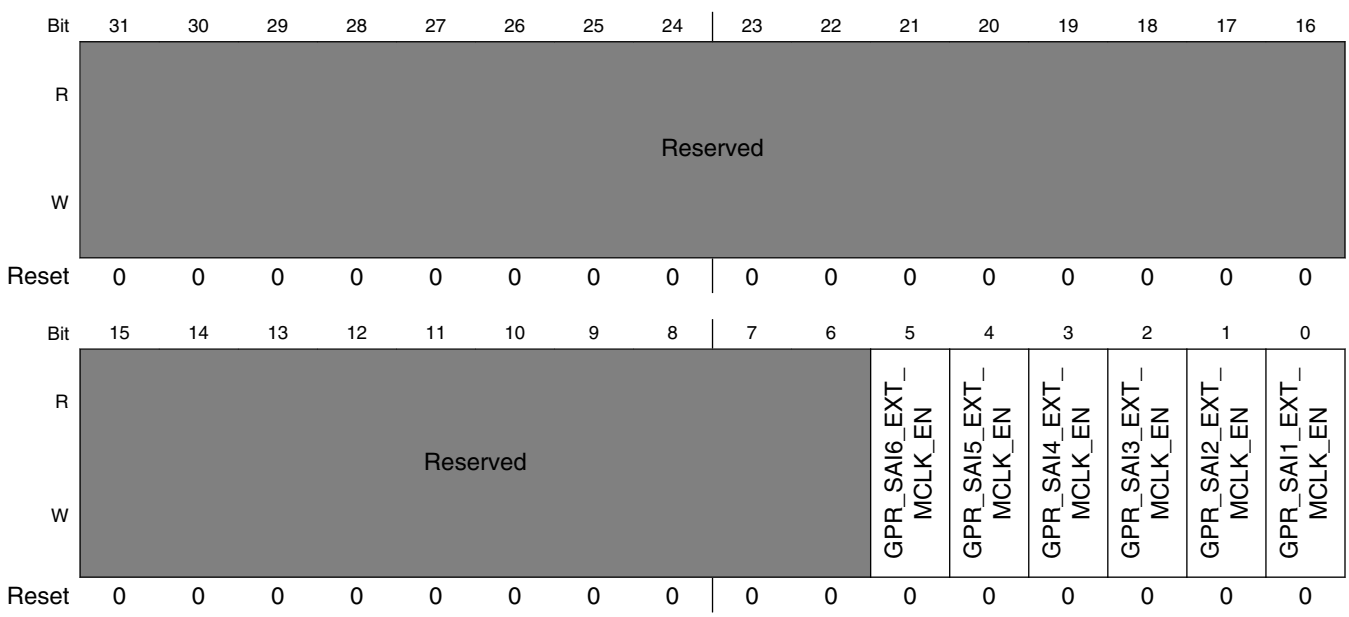
IOMUXC_GPR_GPR1 field descriptions (continued)

Field	Description
15 GPR_ANAMIX_IPT_MODE	
14 -	This field is reserved. Reserved
13 ENET1_TX_CLK_SEL	ENET1 reference clock mode select 0 Gets ENET1 TX reference clk. This clock is also output to pins via the IOMUX. ENET_REF_CLK1 function. 1 Gets ENET1 TX reference clk from the ENET1_TX_CLK pin. In this use case, an external OSC provides the clock for both the external PHY and the internal controller
12 IRQ	Interrupt signal which is connected to CPU IRQS[0]. Used to notify cores on exception condition while boot.
-	This field is reserved. Reserved

8.2.4.3 GPR2 General Purpose Register (IOMUXC_GPR_GPR2)

GPR Register

Address: 3034_0000h base + 8h offset = 3034_0008h



IOMUXC_GPR_GPR2 field descriptions

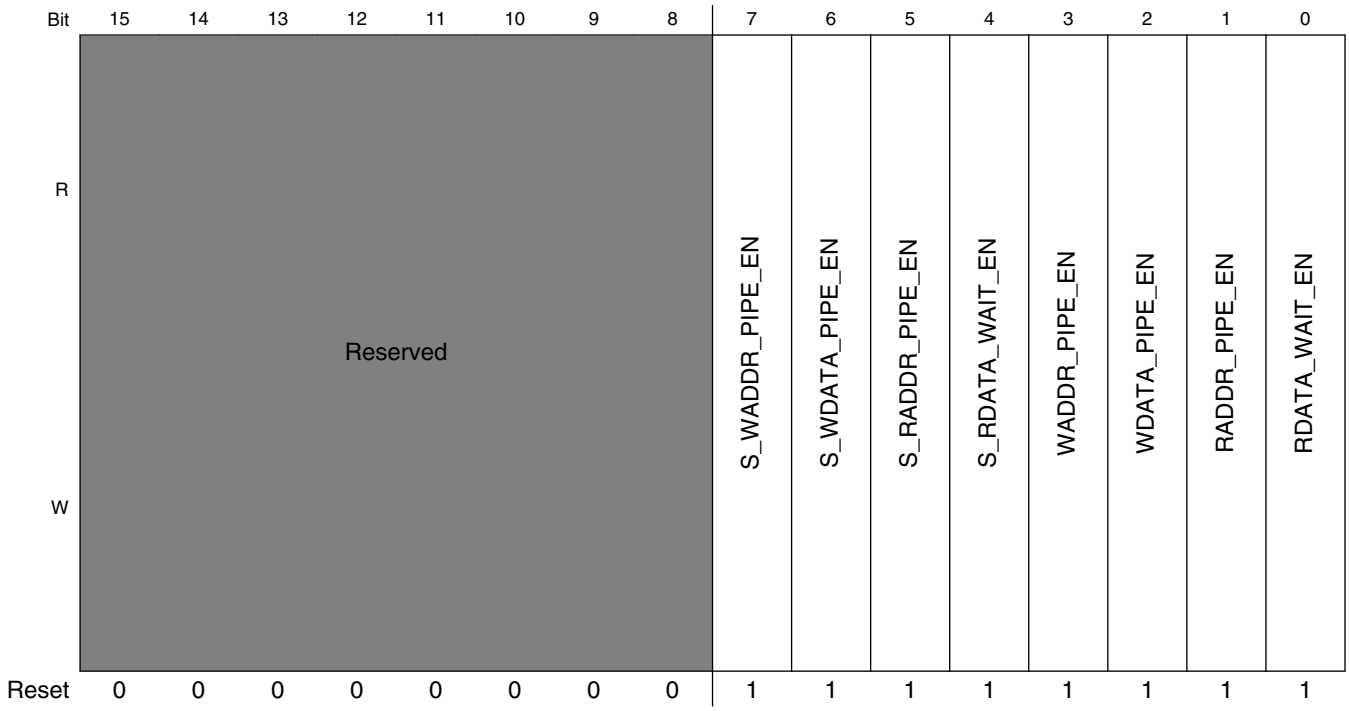
Field	Description
31–6 -	This field is reserved. Reserved
5 GPR_SAI6_ EXT_MCLK_EN	SAI6 External MCLK Enable
4 GPR_SAI5_ EXT_MCLK_EN	SAI5 External MCLK Enable
3 GPR_SAI4_ EXT_MCLK_EN	SAI4 External MCLK Enable
2 GPR_SAI3_ EXT_MCLK_EN	SAI3 External MCLK Enable
1 GPR_SAI2_ EXT_MCLK_EN	SAI2 External MCLK Enable
0 GPR_SAI1_ EXT_MCLK_EN	SAI1 External MCLK Enable

8.2.4.4 GPR3 General Purpose Register (IOMUXC_GPR_GPR3)

GPR Register

Address: 3034_0000h base + Ch offset = 3034_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								S_WADDR_PIPE_EN_PNDG	S_WDATA_PIPE_EN_PNDG	S_RADDR_PIPE_EN_PNDG	S_RDATA_WAIT_EN_PNDG	WADDR_PIPE_EN_PNDG	WDATA_PIPE_EN_PDG	RADDR_PIPE_EN_PDG	RDATA_WAIT_EN_PDG
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IOMUXC_GPR_GPR3 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23 S_WADDR_ PIPE_EN_PNDG	State Retention On-chip RAM write address pipeline enable update is pending 0 write address pipeline enable configuration valid 1 write address pipeline enable bit changed
22 S_WDATA_ PIPE_EN_PNDG	State Retention On-chip RAM write data pipeline enable update is pending 0 write data pipeline enable configuration valid 1 write data pipeline enable bit changed
21 S_RADDR_ PIPE_EN_PNDG	State Retention On-chip RAM read address pipeline enable update is pending 0 read address pipeline enable configuration valid 1 read address pipeline enable bit changed
20 S_RDATA_ WAIT_EN_PNDG	State Retention On-chip RAM read data wait state control update is pending 0 read data wait state control configuration valid 1 read data wait state control bit changed
19 WADDR_PIPE_ EN_PNDG	On-chip RAM write address pipeline enable update is pending 0 write address pipeline enable configuration valid 1 write address pipeline enable bit changed
18 WDATA_PIPE_ EN_PDG	On-chip RAM write data pipeline enable update is pending 0 write data pipeline enable configuration valid 1 write data pipeline enable bit changed

Table continues on the next page...

IOMUXC_GPR_GPR3 field descriptions (continued)

Field	Description
17 RADDR_PIPE_EN_PDG	On-chip RAM read address pipeline enable update is pending 0 read address pipeline enable configuration valid 1 read address pipeline enable bit changed
16 RDATA_WAIT_EN_PDG	On-chip RAM read data wait state control update is pending 0 read data wait state control configuration valid 1 read data wait state control bit changed
15–8 -	This field is reserved. Reserved
7 S_WADDR_PIPE_EN	State Retention On-chip RAM write address pipeline enable For description, please refer to WADDR_PIPE_EN bit
6 S_WDATA_PIPE_EN	State Retention On-chip RAM write data pipeline enable For description, please refer to WDATA_PIPE_EN bit
5 S_RADDR_PIPE_EN	State Retention On-chip RAM read address pipeline enable For description, please refer to RADDR_PIPE_EN bit
4 S_RDATA_WAIT_EN	State Retention On-chip RAM read data wait state control For description, please refer to RDATA_WAIT_EN bit
3 WADDR_PIPE_EN	On-chip RAM write address pipeline enable When this feature is enabled, the write address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, i.e., at most 1 more clock cycle for each write burst with multiple beats of data. When this feature is disabled, the write address from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write data is also ready at this cycle). 0 write address pipeline is disabled 0 write address pipeline is disabled 1 write address pipeline is enabled
2 WDATA_PIPE_EN	On-chip RAM write data pipeline enable When this feature is enabled, the write data from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the write access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI write transaction, i.e., at most 1 more clock cycle for each write burst with multiple beats of data. When this feature is disabled, the write data from the AXI master can be accepted by the on-chip RAM without delay, and data can be written to memory at this cycle (if no other access and write address is also ready at this cycle). 0 write data pipeline is disabled 1 write data pipeline is enabled
1 RADDR_PIPE_EN	On-chip RAM read address pipeline enable When this feature is enabled, the read address from the AXI master would be delayed 1 cycle before it can be accepted by the on-chip RAM. This can avoid setup time issue for the read access on the memory cell at high frequency. Enable this feature would cost at most 1 more clock cycle for each AXI read

Table continues on the next page...

IOMUXC_GPR_GPR3 field descriptions (continued)

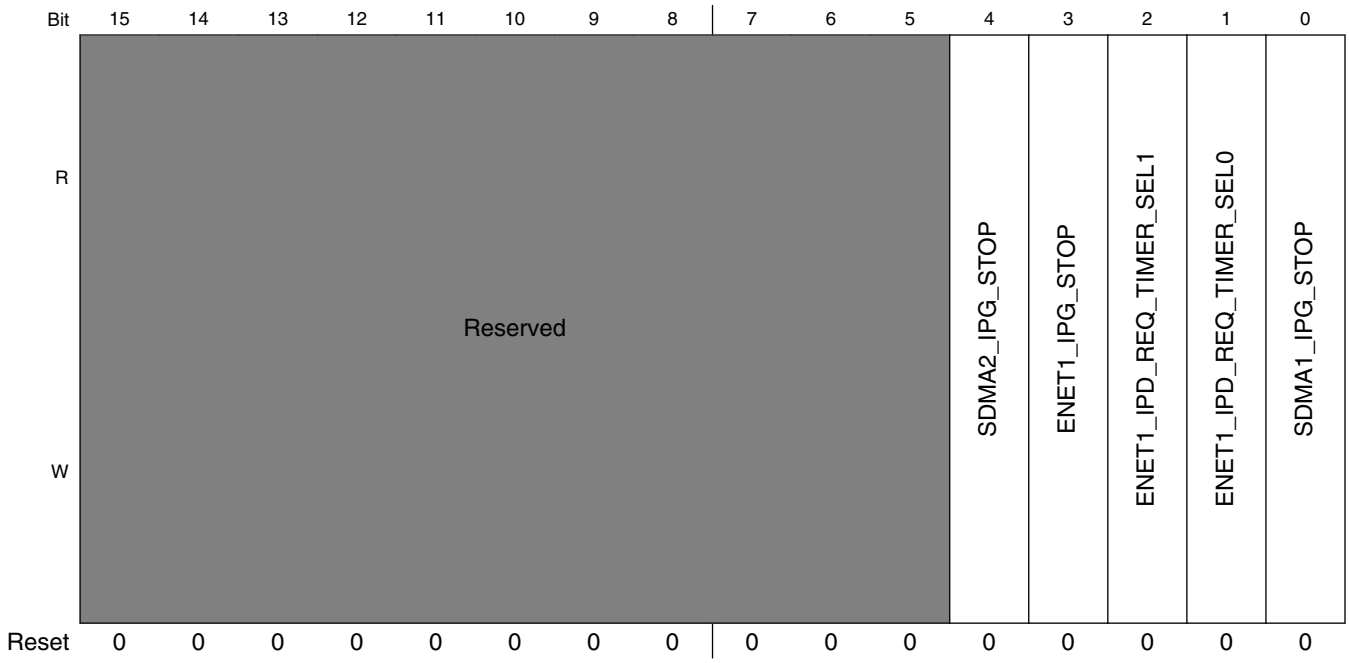
Field	Description
	transaction, i.e., at most 1 more clock cycle for each read burst with multiple beats of data. When this feature is disabled, the read address from the AXI master can be accepted by the on-chip RAM without delay, and data can become ready for master at next clock cycle (if no other access and no read data wait). 0 read address pipeline is disabled 1 read address pipeline is enabled
0 RDATA_WAIT_EN	On-chip RAM read data wait state control When the read data wait state is enabled, it will cost 2 cycles for each read access, (each beat of a read burst). This can avoid the potential timing problem caused by the relatively longer memory access time at higher frequency. When this feature is disabled, it only costs 1 clock cycle to finish a read transaction, i.e., get read data back in the next cycle of read request becomes valid on the bus. 0 read data wait state disabled 1 read data wait state enabled

8.2.4.5 GPR4 General Purpose Register (IOMUXC_GPR_GPR4)

GPR Register

Address: 3034_0000h base + 10h offset = 3034_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0					SAI6_IPG_STOP_ACK	SAI5_IPG_STOP_ACK	SAI4_IPG_STOP_ACK	SAI3_IPG_STOP_ACK	SAI2_IPG_STOP_ACK	SAI1_IPG_STOP_ACK	SDMA2_IPG_STOP_ACK	ENET1_IPG_STOP_ACK	Reserved		SDMA1_IPG_STOP_ACK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



IOMUXC_GPR_GPR4 field descriptions

Field	Description
31–27 Reserved	This read-only field is reserved and always has the value 0.
26 SAI6_IPG_ STOP_ACK	SAI6 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
25 SAI5_IPG_ STOP_ACK	SAI5 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
24 SAI4_IPG_ STOP_ACK	SAI4 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
23 SAI3_IPG_ STOP_ACK	SAI3 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
22 SAI2_IPG_ STOP_ACK	SAI2 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
21 SAI1_IPG_ STOP_ACK	SAI1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode

Table continues on the next page...

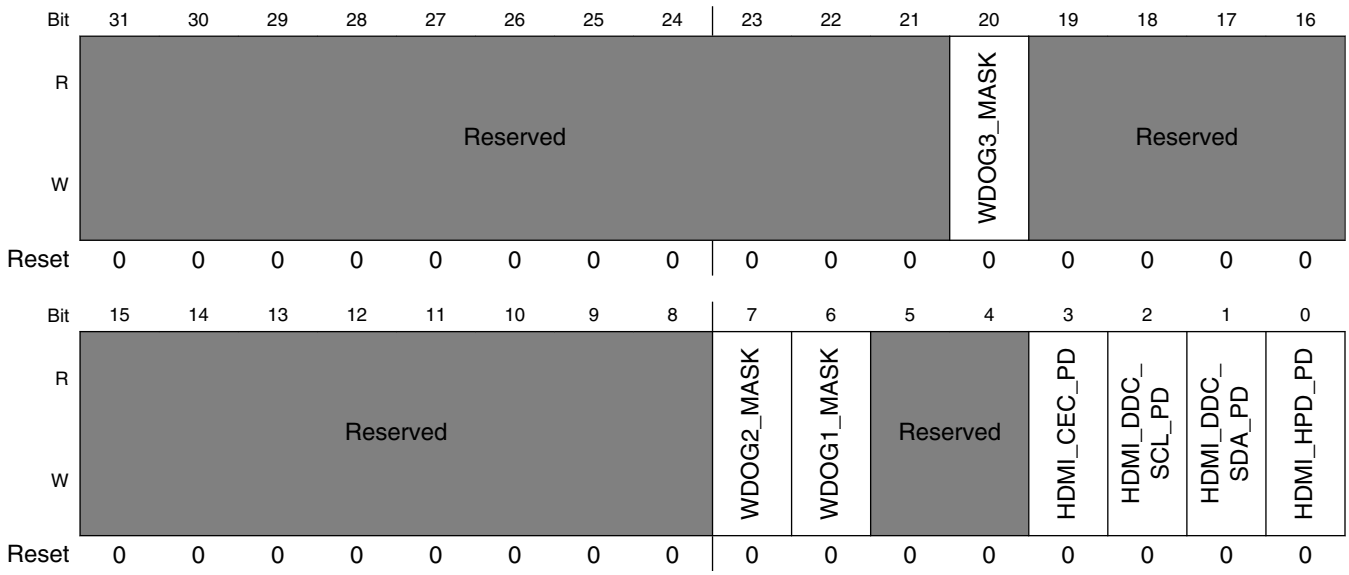
IOMUXC_GPR_GPR4 field descriptions (continued)

Field	Description
20 SDMA2_IPG_ STOP_ACK	SDMA2 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
19 ENET1_IPG_ STOP_ACK	ENET1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
18–17 -	This field is reserved. Reserved
16 SDMA1_IPG_ STOP_ACK	SDMA1 stop acknowledge 0 stop acknowledge is not asserted 1 stop acknowledge is asserted, peripheral is in STOP mode
15–5 -	This field is reserved. Reserved
4 SDMA2_IPG_ STOP	SDMA2 stop request 0 stop request off 1 stop request on
3 ENET1_IPG_ STOP	ENET1 stop request 0 stop request off 1 stop request on
2 ENET1_IPD_ REQ_TIMER_ SEL1	ENET1 IPD_REQ Timer Select 1 0 Select ipd_req_mac0_timer3 to SDMA IRQ 47 1 Select ipd_req_mac0_timer1 to SDMA IRQ 47
1 ENET1_IPD_ REQ_TIMER_ SEL0	ENET1 IPD_REQ Timer Select 0 0 Select ipd_req_mac0_timer2 to SDMA IRQ 45 1 Select ipd_req_mac0_timer0 to SDMA IRQ 45
0 SDMA1_IPG_ STOP	SDMA1 stop request 0 stop request off 1 stop request on

8.2.4.6 GPR5 General Purpose Register (IOMUXC_GPR_GPR5)

GPR Register

Address: 3034_0000h base + 14h offset = 3034_0014h



IOMUXC_GPR_GPR5 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 WDOG3_MASK	WDOG3 Timeout Mask 0 WDOG3 Timeout behaves normally 1 WDOG3 Timeout is masked
19–8 -	This field is reserved. Reserved
7 WDOG2_MASK	WDOG2 Timeout Mask 0 WDOG2 Timeout behaves normally 1 WDOG2 Timeout is masked
6 WDOG1_MASK	WDOG1 Timeout Mask 0 WDOG1 Timeout behaves normally 1 WDOG1 Timeout is masked
5–4 -	This field is reserved. Reserved
3 HDMI_CEC_PD	Connect to hdmi_cec_pd_pad

Table continues on the next page...

IOMUXC_GPR_GPR5 field descriptions (continued)

Field	Description
2 HDMI_DDC_ SCL_PD	Connect to hdmi_ddc_scl_pd_pad
1 HDMI_DDC_ SDA_PD	Connect to hdmi_ddc_sda_pd_pad
0 HDMI_HPD_PD	Connect to hdmi_hpd_pd_pad

8.2.4.7 GPR6 General Purpose Register (IOMUXC_GPR_GPR6)

GPR Register

Address: 3034_0000h base + 18h offset = 3034_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR6 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.8 GPR7 General Purpose Register (IOMUXC_GPR_GPR7)

GPR Register

Address: 3034_0000h base + 1Ch offset = 3034_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

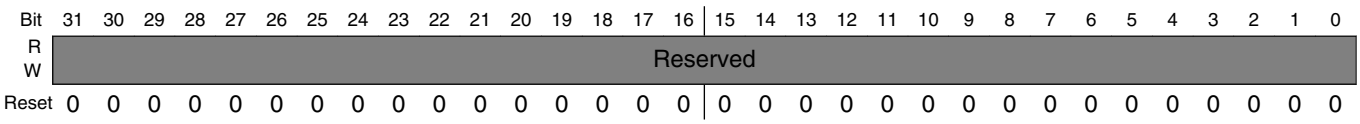
IOMUXC_GPR_GPR7 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.9 GPR8 General Purpose Register (IOMUXC_GPR_GPR8)

GPR Register

Address: 3034_0000h base + 20h offset = 3034_0020h



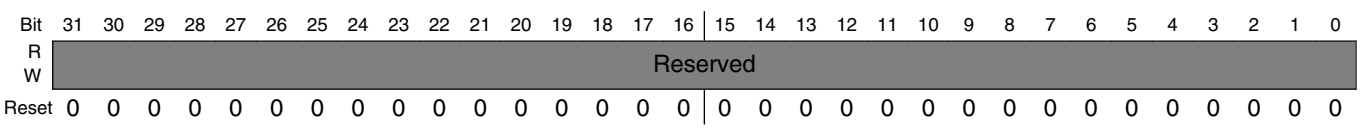
IOMUXC_GPR_GPR8 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.10 GPR9 General Purpose Register (IOMUXC_GPR_GPR9)

GPR Register

Address: 3034_0000h base + 24h offset = 3034_0024h



IOMUXC_GPR_GPR9 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.11 GPR10 General Purpose Register (IOMUXC_GPR_GPR10)

GPR Register

Address: 3034_0000h base + 28h offset = 3034_0028h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved												EXC_ERR_RESP_EN_LOCK	SEC_ERR_RESP_EN_LOCK	TZASC_ID_SWAP_BYPASS_LOCK	TZASC_EN_LOCK
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												EXC_ERR_RESP_EN	SEC_ERR_RESP_EN	TZASC_ID_SWAP_BYPASS	TZASC_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

IOMUXC_GPR_GPR10 field descriptions

Field	Description
31–20 -	This field is reserved. Reserved
19 EXC_ERR_RESP_EN_LOCK	Lock bit for EXC_ERR_RESP_EN
18 SEC_ERR_RESP_EN_LOCK	Lock bit for SEC_ERR_RESP_EN
17 TZASC_ID_SWAP_BYPASS_LOCK	Lock bit for TZASC_ID_SWAP_BYPASS
16 TZASC_EN_LOCK	Lock bit for TZASC_EN

Table continues on the next page...

IOMUXC_GPR_GPR10 field descriptions (continued)

Field	Description
15–4 -	This field is reserved. Reserved
3 EXC_ERR_ RESP_EN	Security exclusive access error response enable for all security gaskets (on both AHB and AXI busses). Enables an ERR response on the AXI vs an OK response for an exclusive access error. 0 OK response on the AXI for an exclusive access error 1 ERR response on the AXI for an exclusive access error
2 SEC_ERR_ RESP_EN	Security error response enable for all security gaskets (on both AHB and AXI busses) This is a "lock" type bit 0 OKAY response 1 SLVERR response
1 TZASC_ID_ SWAP_BYPASS	Connect to id_swap_bypass input on tzasc_id_wrap
0 TZASC_EN	Connect to tzasc_en input on tzasc_id_wrap

8.2.4.12 GPR11 General Purpose Register (IOMUXC_GPR_GPR11)

GPR Register

Address: 3034_0000h base + 2Ch offset = 3034_002Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		OCRAM_S_TZ_ADDR_LOCK			OCRAM_S_TZ_EN_LOCK	Reserved				OCRAM_TZ_ADDR_LOCK					OCRAM_TZ_EN_LOCK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		OCRAM_S_TZ_ADDR			OCRAM_S_TZ_EN	Reserved				OCRAM_TZ_ADDR					OCRAM_TZ_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

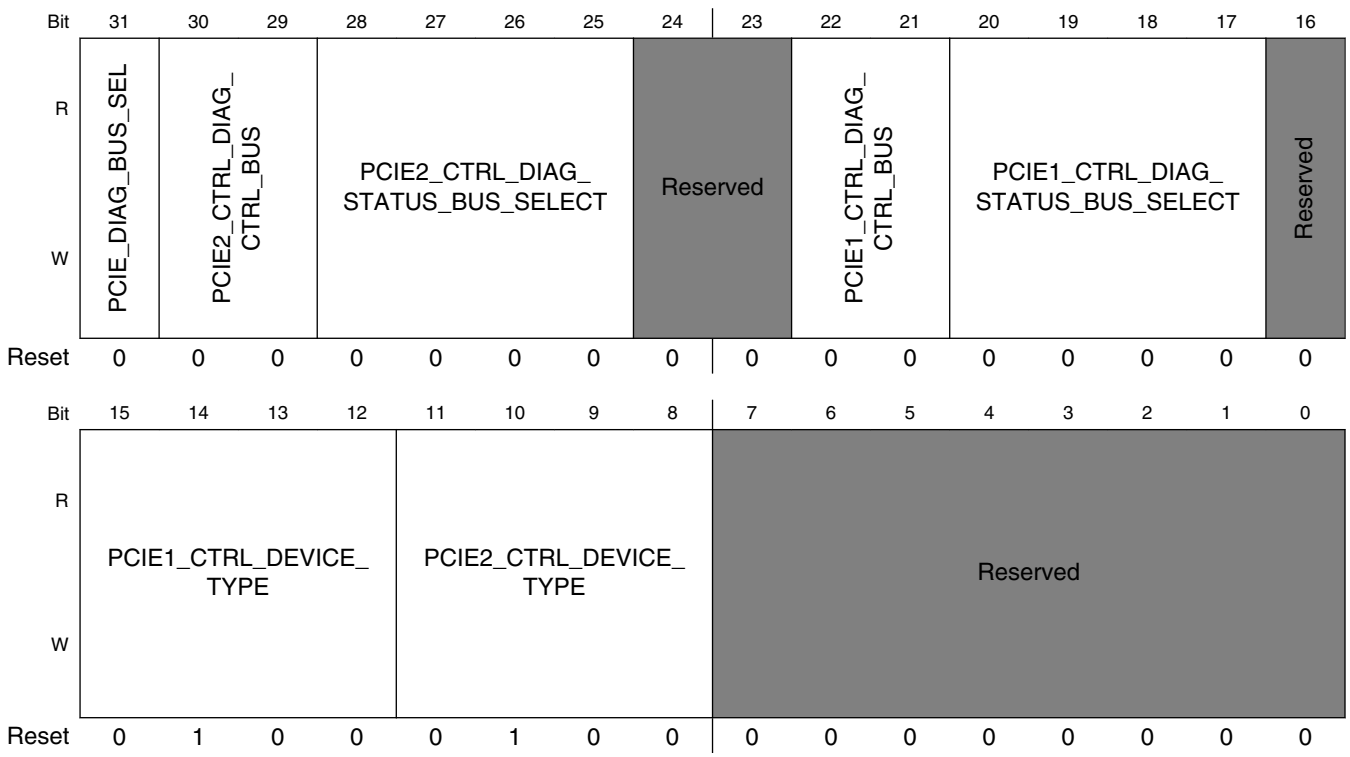
IOMUXC_GPR_GPR11 field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
29–27 OCRAM_S_TZ_ADDR_LOCK	Lock Bits
26 OCRAM_S_TZ_EN_LOCK	Lock Bit
25–22 -	This field is reserved. Reserved
21–17 OCRAM_TZ_ADDR_LOCK	Lock Bits
16 OCRAM_TZ_EN_LOCK	Lock Bit
15–14 -	This field is reserved. Reserved
13–11 OCRAM_S_TZ_ADDR	State Retention OCRAM TrustZone (TZ) start address. This is the start address of the secure memory region within the State Retention OCRAM memory space is 4KB granularity. The start address affects the State Retention OCRAM transactions only if OCRAM_S_TZ_EN bit is set. The State Retention OCRAM TZ ENDADDR is not configurable and is set to the end of State Retention OCRAM memory space. These are "lock" type bits
10 OCRAM_S_TZ_EN	State Retention OCRAM TrustZone (TZ) enable This is a "lock" type bit 0 The TrustZone feature is disabled. Entire State Retention OCRAM space is available for all access types (secure/non-secure/user/supervisor). 1 The TrustZone feature is enabled. Access to address in the range specified by [ENDADDR:STARTADDR] follows the execution mode access policy described in CSU chapter.
9–6 -	This field is reserved. Reserved
5–1 OCRAM_TZ_ADDR	OCRAM TrustZone (TZ) start address. This is the start address of the secure memory region within the OCRAM memory space is 4KB granularity. The start address affects the OCRAM transactions only if OCRAM_TZ_EN bit is set. The OCRAM TZ ENDADDR is not configurable and is set to the end of OCRAM memory space. These are "lock" type bits
0 OCRAM_TZ_EN	OCRAM TrustZone (TZ) enable This is a "lock" type bit 0 The TrustZone feature is disabled. Entire OCRAM space is available for all access types (secure/non-secure/user/supervisor). 1 The TrustZone feature is enabled. Access to address in the range specified by [ENDADDR:STARTADDR] follows the execution mode access policy described in CSU chapter.

8.2.4.13 GPR12 General Purpose Register (IOMUXC_GPR_GPR12)

GPR Register

Address: 3034_0000h base + 30h offset = 3034_0030h



IOMUXC_GPR_GPR12 field descriptions

Field	Description
31 PCIE_DIAG_BUS_SEL	Control the source of the PCIE DIAG STATUS bus, PCIe1 or PCIe2.
30–29 PCIE2_CTRL_DIAG_CTRL_BUS	PCI Express Diagnostic Control Bus
28–25 PCIE2_CTRL_DIAG_STATUS_BUS_SELECT	PCI Express Diagnostic Status Bus Select
24–23 -	This field is reserved. Reserved

Table continues on the next page...

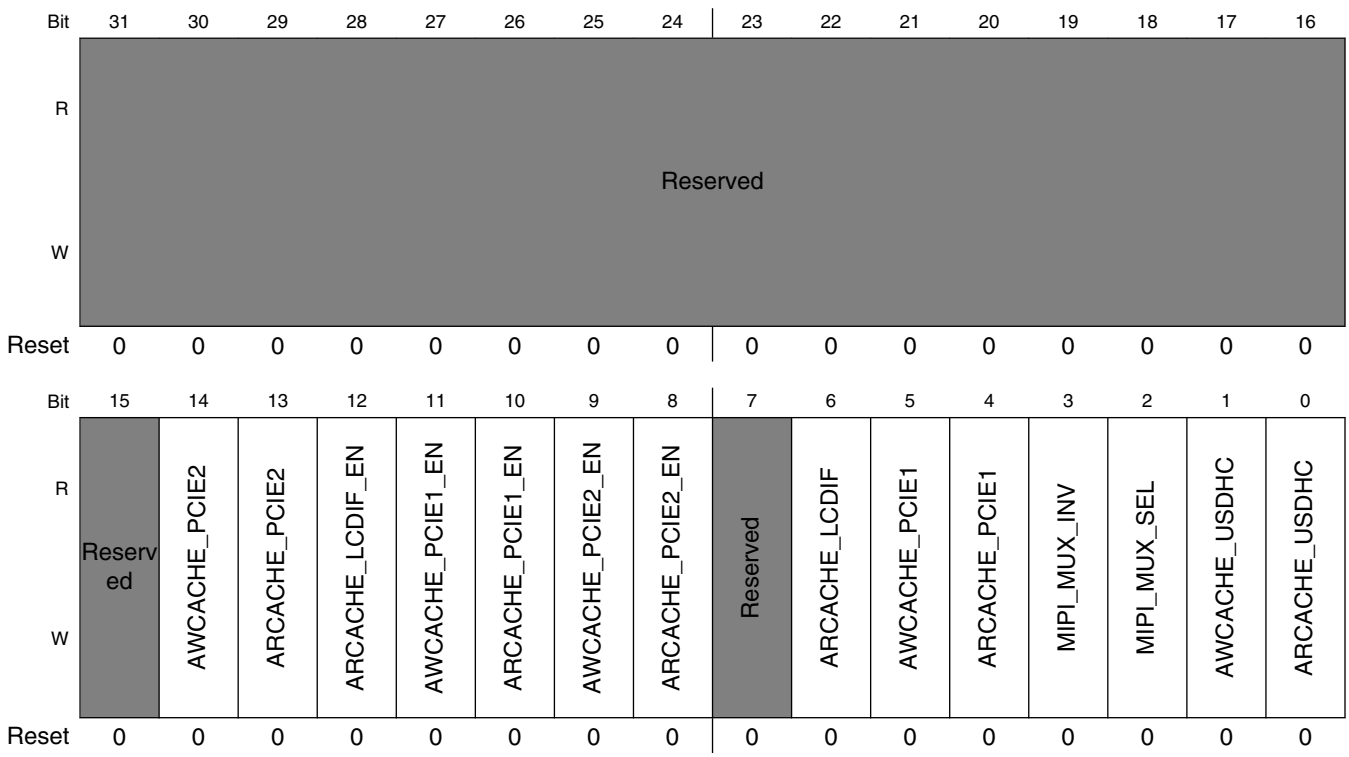
IOMUXC_GPR_GPR12 field descriptions (continued)

Field	Description
22–21 PCIE1_CTRL_ DIAG_CTRL_ BUS	PCI Express Diagnostic Control Bus
20–17 PCIE1_CTRL_ DIAG_STATUS_ BUS_SELECT	PCI Express Diagnostic Status Bus Select
16 -	This field is reserved. Reserved
15–12 PCIE1_CTRL_ DEVICE_TYPE	PCI Express device/port type 0000 PCI Express endpoint 0001 Legacy PCI Express endpoint 0100 Root port of PCI Express root complex
11–8 PCIE2_CTRL_ DEVICE_TYPE	PCI Express device/port type 0000 PCI Express endpoint 0001 Legacy PCI Express endpoint 0100 Root port of PCI Express root complex
-	This field is reserved. Reserved

8.2.4.14 GPR13 General Purpose Register (IOMUXC_GPR_GPR13)

GPR Register

Address: 3034_0000h base + 34h offset = 3034_0034h



IOMUXC_GPR_GPR13 field descriptions

Field	Description
31–15 -	This field is reserved. Reserved
14 AWCACHE_PCIE2	PCIe AXI Master Port AWCACHE Override Value Note: this bit only takes effect when AWCACHE_PCIE_EN = 1 0 Drive PCIe AXI Master Port AWCACHE[1] to 0 1 Drive PCIe AXI Master Port AWCACHE[1] to 1
13 ARCACHE_PCIE2	PCIe AXI Master Port ARCACHE Override Value Note: this bit only takes effect when ARCACHE_PCIE_EN = 1 0 Drive PCIe AXI Master Port ARCACHE[1] to 0 1 Drive PCIe AXI Master Port ARCACHE[1] to 1
12 ARCACHE_LCDIF_EN	LCDIF AXI Master Port ARCACHE Override Enable

Table continues on the next page...

IOMUXC_GPR_GPR13 field descriptions (continued)

Field	Description
	0 LCDIF AXI Master Port ARCACHE[1] driven by LCDIF 1 LCDIF AXI Master Port ARCACHE[1] driven to constant value specified by the ARCACHE_LCDIF bit
11 AWCACHE_PCIE1_EN	PCIe AXI Master Port AWCACHE Override Enable 0 PCIe AXI Master Port AWCACHE[1] driven by PCIe 1 PCIe AXI Master Port AWCACHE[1] driven to constant value specified by the AWCACHE_PCIE1 bit
10 ARCACHE_PCIE1_EN	PCIe AXI Master Port ARCACHE Override Enable 0 PCIe AXI Master Port ARCACHE[1] driven by PCIe 1 PCIe AXI Master Port ARCACHE[1] driven to constant value specified by the ARCACHE_PCIe1 bit
9 AWCACHE_PCIE2_EN	PCIe AXI Master Port AWCACHE Override Enable 0 PCIE Primary AXI Master Port AWCACHE[1] driven by PCIE 1 PCIe Primary AXI Master Port AWCACHE[1] driven to constant value specified by the AWCACHE_PCIE2 bit
8 ARCACHE_PCIE2_EN	PCIe AXI Master Port ARCACHE Override Enable 0 PCIE Primary AXI Master Port ARCACHE[1] driven by PCIE 1 PCIE Primary AXI Master Port ARCACHE[1] driven to constant value specified by the ARCACHE_PCIE2 bit
7 -	This field is reserved. Reserved
6 ARCACHE_LCDIF	LCDIF AXI Master Port ARCACHE Override Value Note: this bit only takes effect when ARCACHE_LCDIF_EN = 1 0 Drive LCDIF AXI Master Port ARCACHE[1] to 0 1 Drive LCDIF AXI Master Port ARCACHE[1] to 1
5 AWCACHE_PCIE1	PCIe AXI Master Port AWCACHE Override Value Note: this bit only takes effect when AWCACHE_PCIE_EN = 1 0 Drive PCIe AXI Master Port AWCACHE[1] to 0 1 Drive PCIe AXI Master Port AWCACHE[1] to 1
4 ARCACHE_PCIE1	PCIe AXI Master Port ARCACHE Override Value Note: this bit only takes effect when ARCACHE_PCIE_EN = 1 0 Drive PCIe AXI Master Port ARCACHE[1] to 0 1 Drive PCIe AXI Master Port ARCACHE[1] to 1
3 MIPI_MUX_INV	MIPI MUX INV
2 MIPI_MUX_SEL	MIPI MUX SEL
1 AWCACHE_USDHC	USDHC 1-3 AXI Master AWCACHE Override Value Note: this bit always overrides 0 Drive USDHC AXI Master AWCACHE[1] to 0 1 Drive USDHC AXI Master AWCACHE[1] to 1

Table continues on the next page...

IOMUXC_GPR_GPR13 field descriptions (continued)

Field	Description
0 ARCACHE_ USDHC	USDHC 1-3 AXI Master ARCACHE Override Value Note: this bit always overrides 0 Drive USDHC AXI Master ARCACHE[1] to 0 1 Drive USDHC AXI Master ARCACHE[1] to 1

8.2.4.15 GPR14 General Purpose Register (IOMUXC_GPR_GPR14)

GPR Register

Address: 3034_0000h base + 38h offset = 3034_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCIE1_PHY_LOS_BIAS			PCIE1_PHY_LOS_LEVEL					PCIE1_PHY_RX0_EQ			PCIE1_PHY_TX0_TERM_OFFSET				
Reset	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE1_PHY_TX_VBOOST_LVL			PCIE1_VREG_BYPASS	PCIE1_CLKREQ_B_OVERRIDE	PCIE1_CLKREQ_B_OVERRIDE_EN	PCIE1_REF_USE_PAD	PCIE1_APP_CLK_PM_EN	Reserved							
Reset	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR14 field descriptions

Field	Description
31–29 PCIE1_PHY_ LOS_BIAS	To PCIe PHY
28–24 PCIE1_PHY_ LOS_LEVEL	To PCIe PHY

Table continues on the next page...

IOMUXC_GPR_GPR14 field descriptions (continued)

Field	Description
23–21 PCIE1_PHY_ RX0_EQ	To PCIe PHY
20–16 PCIE1_PHY_ TX0_TERM_ OFFSET	To PCIe PHY
15–13 PCIE1_PHY_TX_ VBOOST_LVL	To PCIe PHY
12 PCIE1_VREG_ BYPASS	To PCIe PHY
11 PCIE1_ CLKREQ_B_ OVERRIDE	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller
10 PCIE1_ CLKREQ_B_ OVERRIDE_EN	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller
9 PCIE1_REF_ USE_PAD	To PCIe PHY
8 PCIE1_APP_ CLK_PM_EN	To PCIe PHY
-	This field is reserved. Reserved

8.2.4.16 GPR15 General Purpose Register (IOMUXC_GPR_GPR15)

GPR Register

Address: 3034_0000h base + 3Ch offset = 3034_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE1_PCS_TX_ DEEMPH_GEN1						PCIE1_PCS_TX_ DEEMPH_GEN2_ 3P5DB						PCIE1_PCS_TX_ DEEMPH_GEN2_ 6DB				PCIE1_PCS_TX_ SWING_FULL									PCIE1_PCS_TX_ SWING_LOW							
Reset	0	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IOMUXC_GPR_GPR15 field descriptions

Field	Description
31–26 PCIE1_PCS_TX_DEEMPH_GEN1	To PCIe PHY
25–20 PCIE1_PCS_TX_DEEMPH_GEN2_3P5DB	To PCIe PHY
19–14 PCIE1_PCS_TX_DEEMPH_GEN2_6DB	To PCIe PHY
13–7 PCIE1_PCS_TX_SWING_FULL	To PCIe PHY
PCIE1_PCS_TX_SWING_LOW	To PCIe PHY

8.2.4.17 GPR16 General Purpose Register (IOMUXC_GPR_GPR16)

GPR Register

Address: 3034_0000h base + 40h offset = 3034_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCIE2_PHY_LOS_BIAS			PCIE2_PHY_LOS_LEVEL					PCIE2_PHY_RX0_EQ			PCIE2_PHY_TX0_TERM_OFFSET				
Reset	0	1	0	0	1	0	0	1	0	1	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE2_PHY_TX_VBOOST_LVL			PCIE2_VREG_BYPASS	PCIE2_CLKREQ_B_OVERRIDE	PCIE2_CLKREQ_B_OVERRIDE_EN	PCIE2_REF_USE_PAD	PCIE2_APP_CLK_PM_EN	Reserved							
Reset	1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR16 field descriptions

Field	Description
31–29 PCIE2_PHY_ LOS_BIAS	To PCIe PHY
28–24 PCIE2_PHY_ LOS_LEVEL	To PCIe PHY
23–21 PCIE2_PHY_ RX0_EQ	To PCIe PHY
20–16 PCIE2_PHY_ TX0_TERM_ OFFSET	To PCIe PHY
15–13 PCIE2_PHY_TX_ VBOOST_LVL	To PCIe PHY
12 PCIE2_VREG_ BYPASS	To PCIe PHY
11 PCIE2_ CLKREQ_B_ OVERRIDE	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller
10 PCIE2_ CLKREQ_B_ OVERRIDE_EN	Control the PCIE_CLKREQ_B to the pad together with CLKREQ_B from controller
9 PCIE2_REF_ USE_PAD	To PCIe PHY
8 PCIE2_APP_ CLK_PM_EN	To PCIe PHY
-	This field is reserved. Reserved

8.2.4.18 GPR17 General Purpose Register (IOMUXC_GPR_GPR17)

GPR Register

Address: 3034_0000h base + 44h offset = 3034_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCIE2_PCS_TX_DEEMPH_GEN1						PCIE2_PCS_TX_DEEMPH_GEN2_3P5DB						PCIE2_PCS_TX_DEEMPH_GEN2_6DB				PCIE2_PCS_TX_SWING_FULL									PCIE2_PCS_TX_SWING_LOW					
Reset	0	1	1	0	0	0	0	1	1	0	0	0	1	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

IOMUXC_GPR_GPR17 field descriptions

Field	Description
31–26 PCIE2_PCS_TX_DEEMPH_GEN1	To PCIe PHY
25–20 PCIE2_PCS_TX_DEEMPH_GEN2_3P5DB	To PCIe PHY
19–14 PCIE2_PCS_TX_DEEMPH_GEN2_6DB	To PCIe PHY
13–7 PCIE2_PCS_TX_SWING_FULL	To PCIe PHY
PCIE2_PCS_TX_SWING_LOW	To PCIe PHY

8.2.4.19 GPR18 General Purpose Register (IOMUXC_GPR_GPR18)

GPR Register

Address: 3034_0000h base + 48h offset = 3034_0048h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

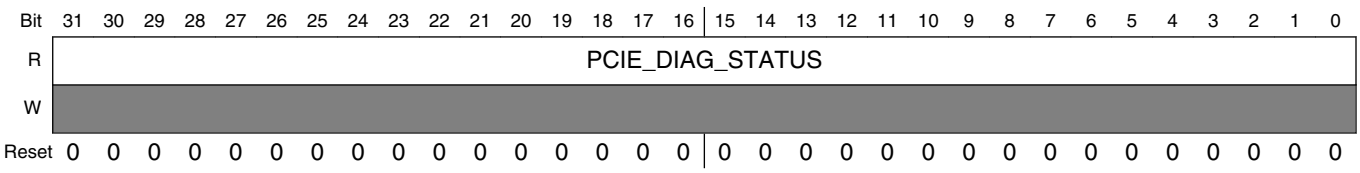
IOMUXC_GPR_GPR18 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.20 GPR19 General Purpose Register (IOMUXC_GPR_GPR19)

GPR Register

Address: 3034_0000h base + 4Ch offset = 3034_004Ch



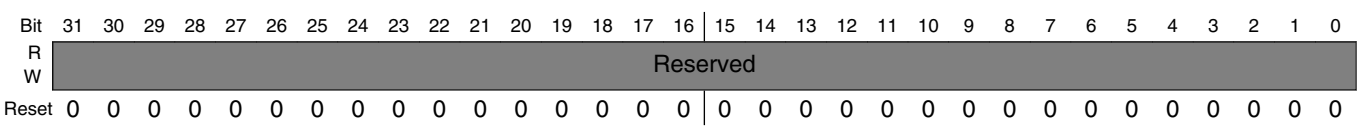
IOMUXC_GPR_GPR19 field descriptions

Field	Description
PCIE_DIAG_STATUS	PCIe DIAG Status Bus

8.2.4.21 GPR20 General Purpose Register (IOMUXC_GPR_GPR20)

GPR Register

Address: 3034_0000h base + 50h offset = 3034_0050h



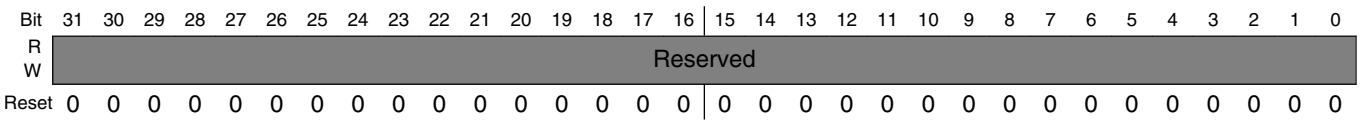
IOMUXC_GPR_GPR20 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.22 GPR21 General Purpose Register (IOMUXC_GPR_GPR21)

GPR Register

Address: 3034_0000h base + 54h offset = 3034_0054h



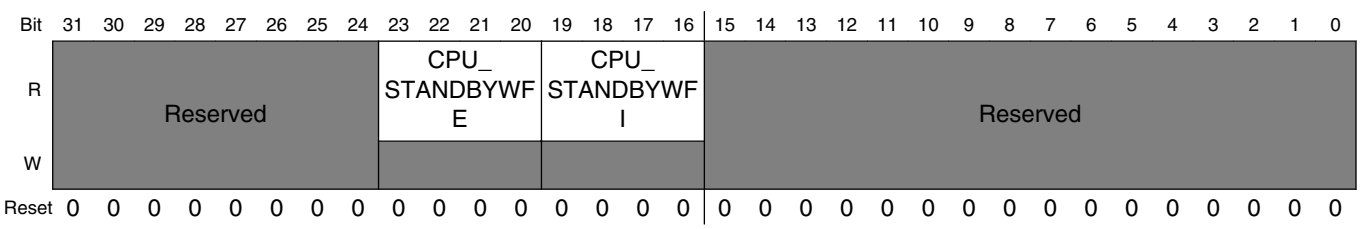
IOMUXC_GPR_GPR21 field descriptions

Field	Description
-	This field is reserved. Reserved

8.2.4.23 GPR22 General Purpose Register (IOMUXC_GPR_GPR22)

GPR Register

Address: 3034_0000h base + 58h offset = 3034_0058h



IOMUXC_GPR_GPR22 field descriptions

Field	Description
31–24 -	This field is reserved. Reserved
23–20 CPU_ STANDBYWFE	Status of CPU STANDBYWFE low power states MSB: status of core 3 STANDBYWFE low power state LSB: status of core 0 STANDBYWFE low power state
19–16 CPU_ STANDBYWFI	Status of CPU STANDBYWFI low power states MSB: status of core 3 STANDBYWFI low power state LSB: status of core 0 STANDBYWFI low power state
-	This field is reserved. Reserved

8.2.4.24 GPR23 General Purpose Register (IOMUXC_GPR_GPR23)

GPR Register

Address: 3034_0000h base + 5Ch offset = 3034_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DDSI_DPHY_TURNAROUND	DSI_TRIGGER_REQ	DSI_TRIGGER_SEND		DSI_TX_ULPS_ENABLE				DSI_HSEL	DSI_NOCAL	DSI_RCALT			DSI_RTERM_SEL	DSI_RX_RCAL	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_GPR_GPR23 field descriptions

Field	Description
31–16 -	This field is reserved. Reserved
15 DDSI_DPHY_TURNAROUND	For DSI Control
14 DSI_TRIGGER_REQ	For DSI Control
13–12 DSI_TRIGGER_SEND	For DSI Control
11–7 DSI_TX_ULPS_ENABLE	For DSI Control
6 DSI_HSEL	For DSI Control
5 DSI_NOCAL	For DSI Control
4–3 DSI_RCALT	For DSI Control

Table continues on the next page...

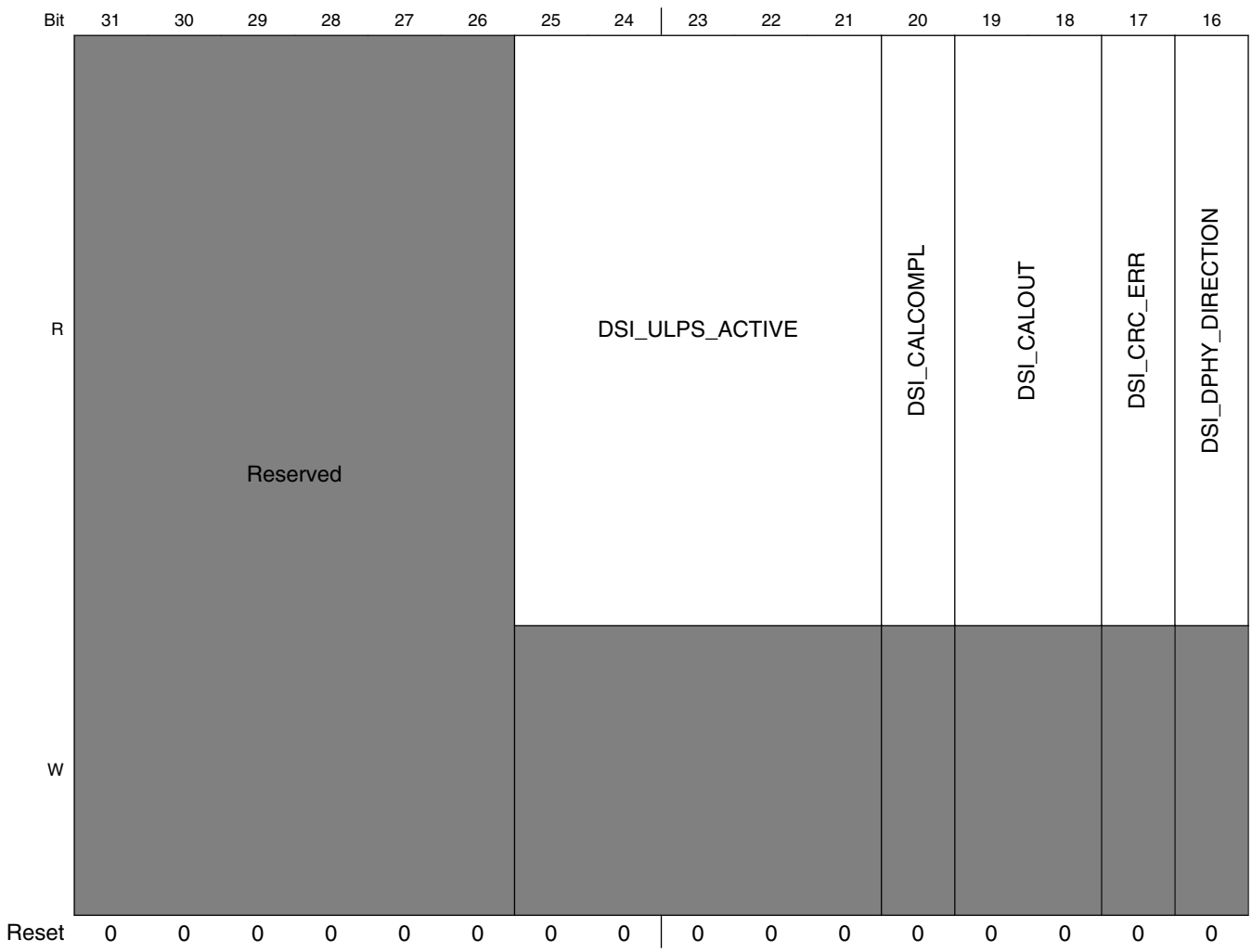
IOMUXC_GPR_GPR23 field descriptions (continued)

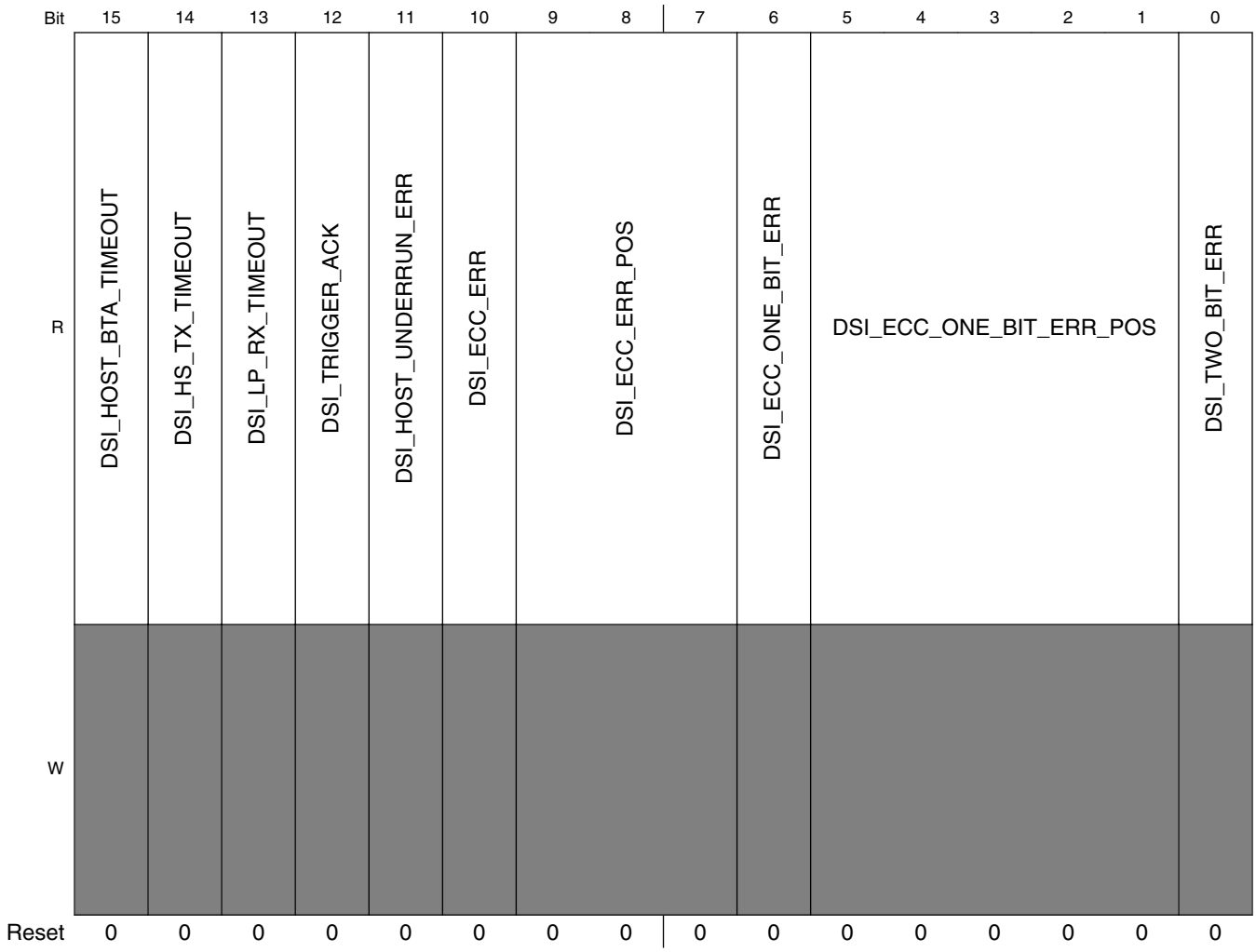
Field	Description
2 DSI_RTERM_ SEL	For DSI Control
DSI_RX_RCAL	For DSI Control

8.2.4.25 GPR24 General Purpose Register (IOMUXC_GPR_GPR24)

GPR Register

Address: 3034_0000h base + 60h offset = 3034_0060h





IOMUXC_GPR_GPR24 field descriptions

Field	Description
31–26 -	This field is reserved. Reserved
25–21 DSI_ULPS_ACTIVE	for DSI status
20 DSI_CALCOMPL	for DSI status
19–18 DSI_CALOUT	for DSI status
17 DSI_CRC_ERR	for DSI status
16 DSI_DPHY_DIRECTION	for DSI status

Table continues on the next page...

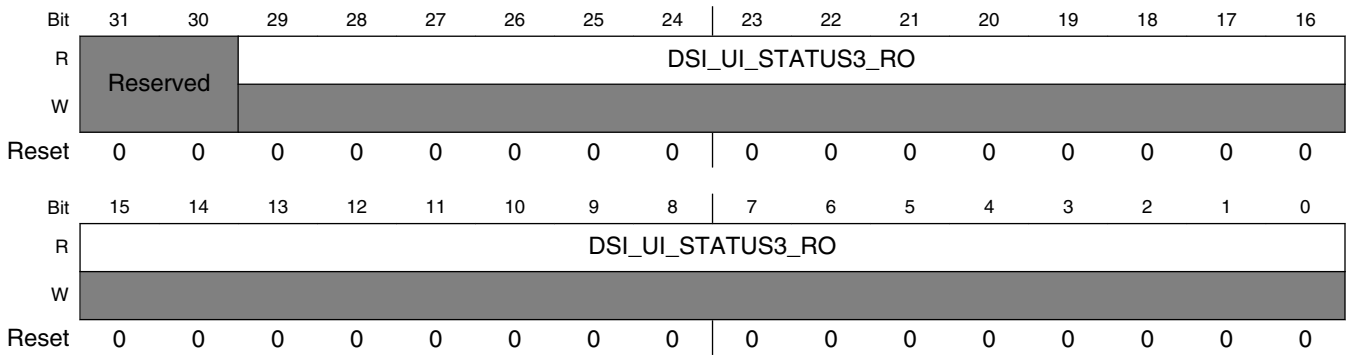
IOMUXC_GPR_GPR24 field descriptions (continued)

Field	Description
15 DSI_HOST_ BTA_TIMEOUT	for DSI status
14 DSI_HS_TX_ TIMEOUT	for DSI status
13 DSI_LP_RX_ TIMEOUT	for DSI status
12 DSI_TRIGGER_ ACK	for DSI status
11 DSI_HOST_ UNDERRUN_ ERR	for DSI status
10 DSI_ECC_ERR	for DSI status
9–7 DSI_ECC_ERR_ POS	for DSI status
6 DSI_ECC_ONE_ BIT_ERR	for DSI status
5–1 DSI_ECC_ONE_ BIT_ERR_POS	for DSI status
0 DSI_TWO_BIT_ ERR	for DSI status

8.2.4.26 GPR25 General Purpose Register (IOMUXC_GPR_GPR25)

GPR Register

Address: 3034_0000h base + 64h offset = 3034_0064h



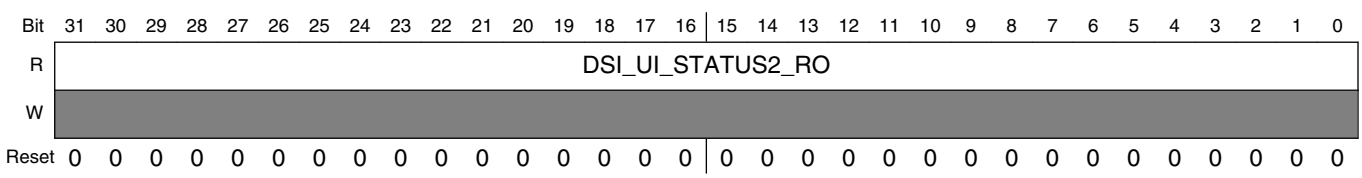
IOMUXC_GPR_GPR25 field descriptions

Field	Description
31–30 -	This field is reserved. Reserved
DSI_UI_ STATUS3_RO	for DSI status: host_ui_status[125:96]

8.2.4.27 GPR26 General Purpose Register (IOMUXC_GPR_GPR26)

GPR Register

Address: 3034_0000h base + 68h offset = 3034_0068h



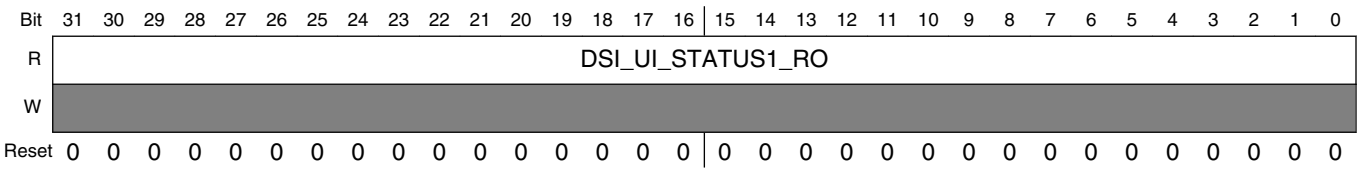
IOMUXC_GPR_GPR26 field descriptions

Field	Description
DSI_UI_ STATUS2_RO	for DSI status: host_ui_status[95:64]

8.2.4.28 GPR27 General Purpose Register (IOMUXC_GPR_GPR27)

GPR Register

Address: 3034_0000h base + 6Ch offset = 3034_006Ch



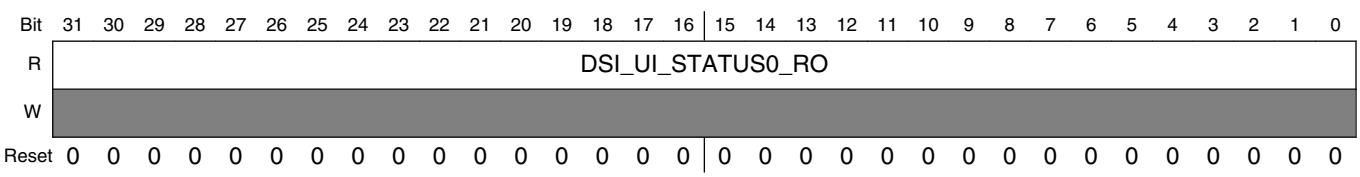
IOMUXC_GPR_GPR27 field descriptions

Field	Description
DSI_UI_ STATUS1_RO	for DSI status: host_ui_status[63:32]

8.2.4.29 GPR28 General Purpose Register (IOMUXC_GPR_GPR28)

GPR Register

Address: 3034_0000h base + 70h offset = 3034_0070h



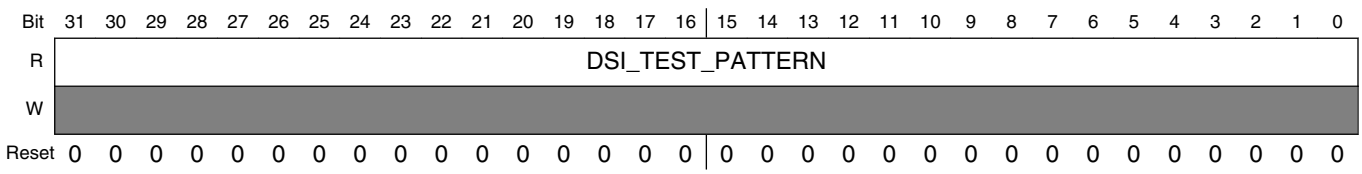
IOMUXC_GPR_GPR28 field descriptions

Field	Description
DSI_UI_ STATUS0_RO	for DSI status: host_ui_status[31:0]

8.2.4.30 GPR29 General Purpose Register (IOMUXC_GPR_GPR29)

GPR Register

Address: 3034_0000h base + 74h offset = 3034_0074h



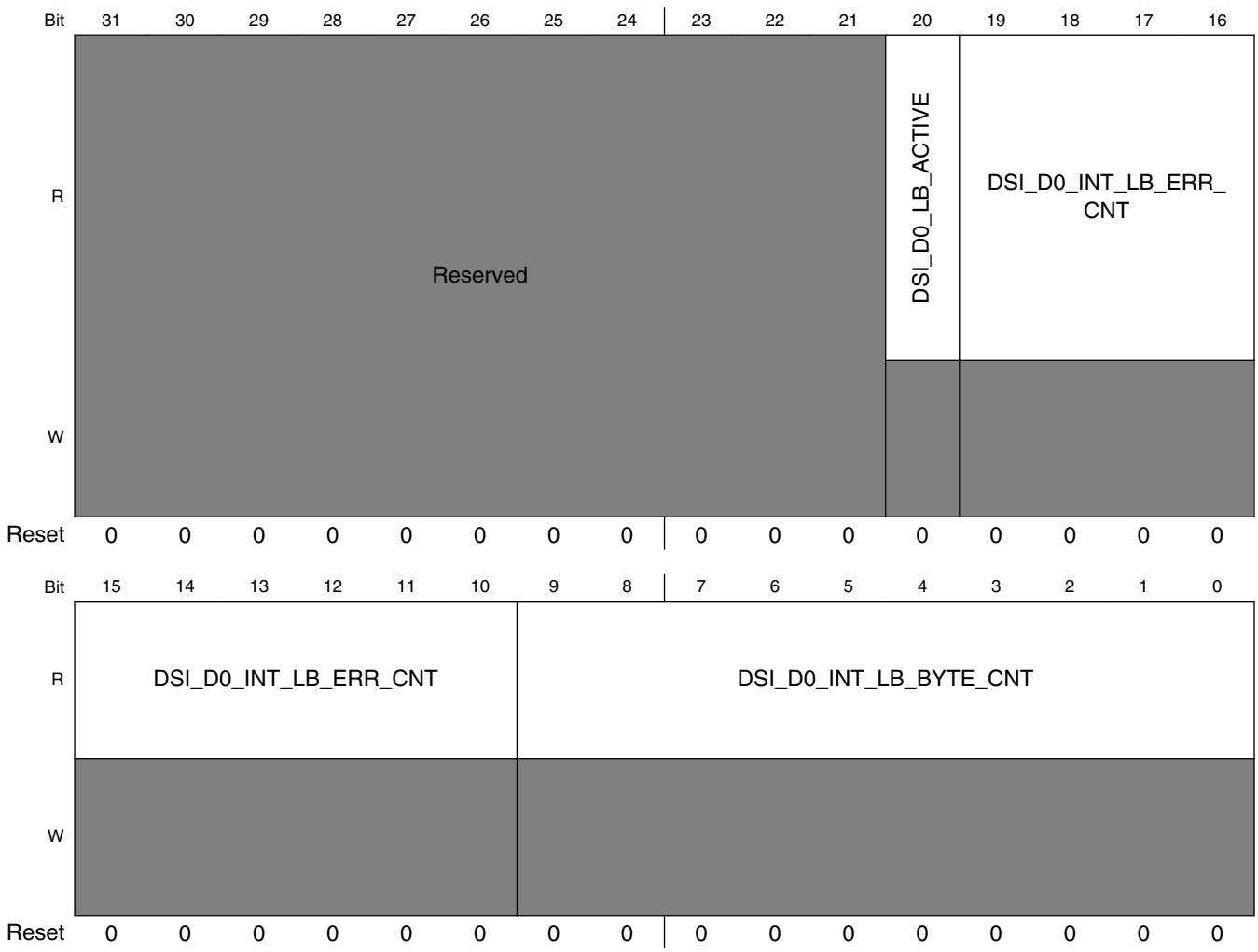
IOMUXC_GPR_GPR29 field descriptions

Field	Description
DSI_TEST_PATTERN	for DSI test control

8.2.4.31 GPR30 General Purpose Register (IOMUXC_GPR_GPR30)

GPR Register

Address: 3034_0000h base + 78h offset = 3034_0078h



IOMUXC_GPR_GPR30 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 DSI_D0_LB_ACTIVE	for DSI test status
19–10 DSI_D0_INT_LB_ERR_CNT	for DSI test status

Table continues on the next page...

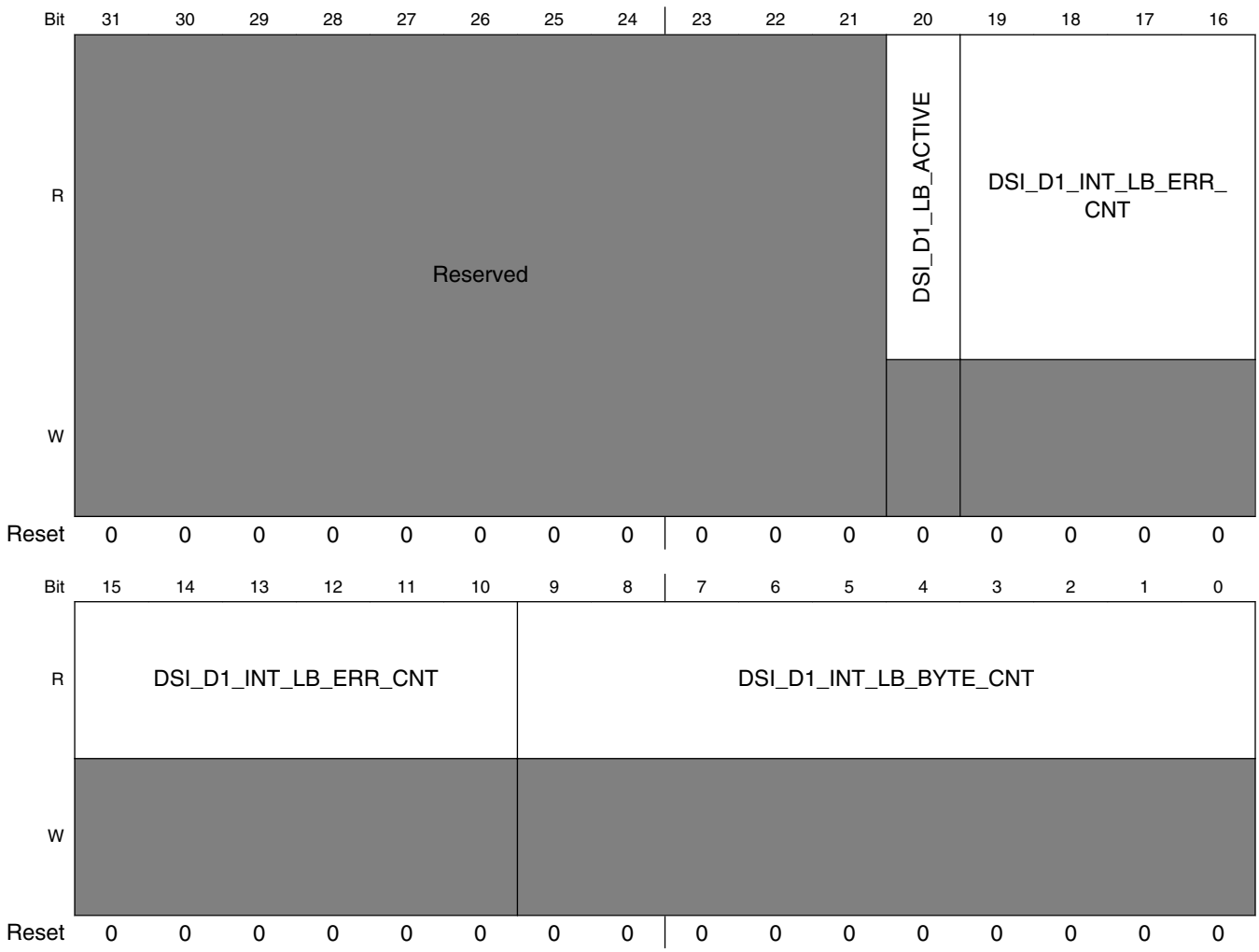
IOMUXC_GPR_GPR30 field descriptions (continued)

Field	Description
DSI_D0_INT_ LB_BYTE_CNT	for DSI test status

8.2.4.32 GPR31 General Purpose Register (IOMUXC_GPR_GPR31)

GPR Register

Address: 3034_0000h base + 7Ch offset = 3034_007Ch



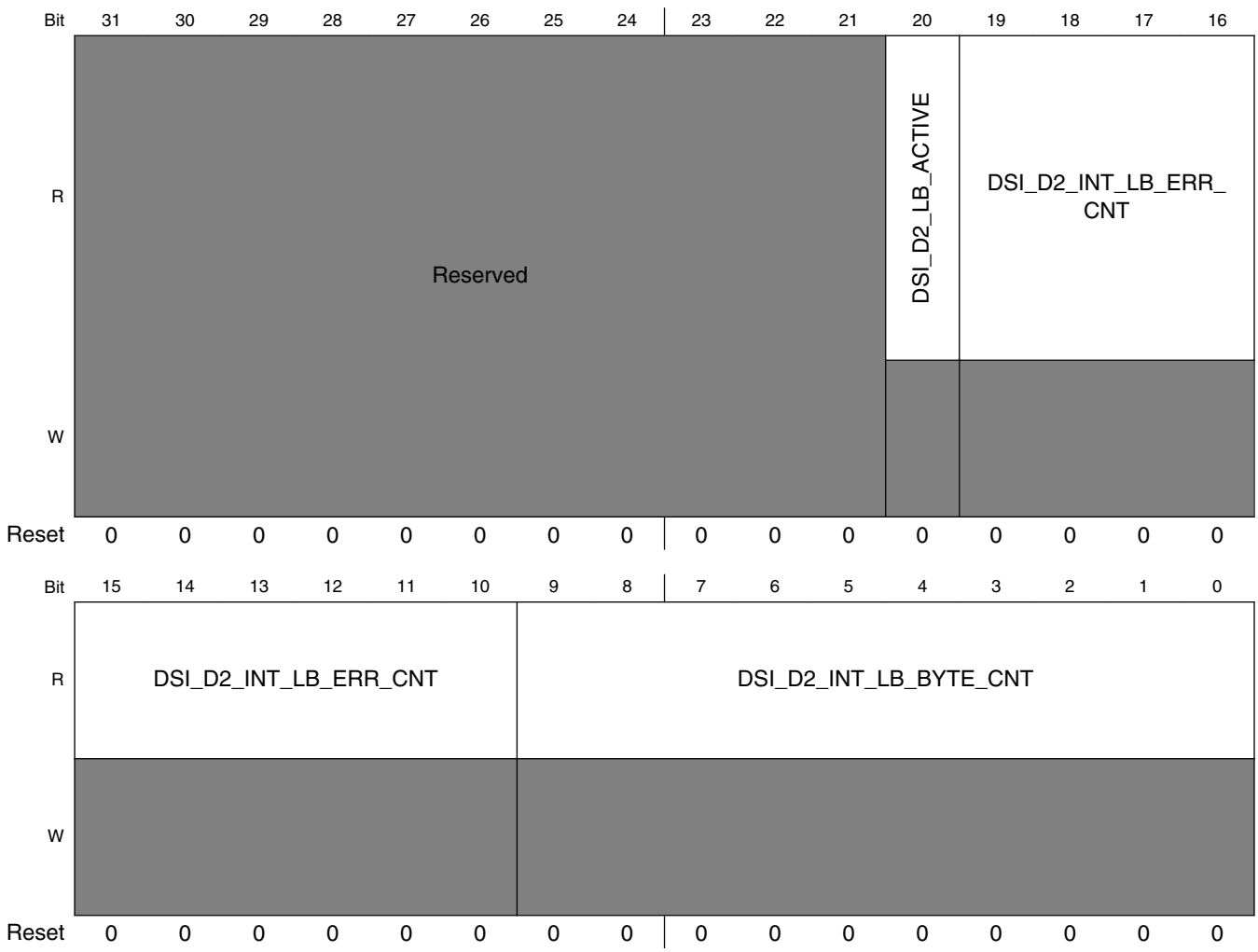
IOMUXC_GPR_GPR31 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 DSI_D1_LB_ACTIVE	for DSI test status
19–10 DSI_D1_INT_LB_ERR_CNT	for DSI test status
DSI_D1_INT_LB_BYTE_CNT	for DSI test status

8.2.4.33 GPR32 General Purpose Register (IOMUXC_GPR_GPR32)

GPR Register

Address: 3034_0000h base + 80h offset = 3034_0080h



IOMUXC_GPR_GPR32 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 DSI_D2_LB_ACTIVE	for DSI test status
19–10 DSI_D2_INT_LB_ERR_CNT	for DSI test status

Table continues on the next page...

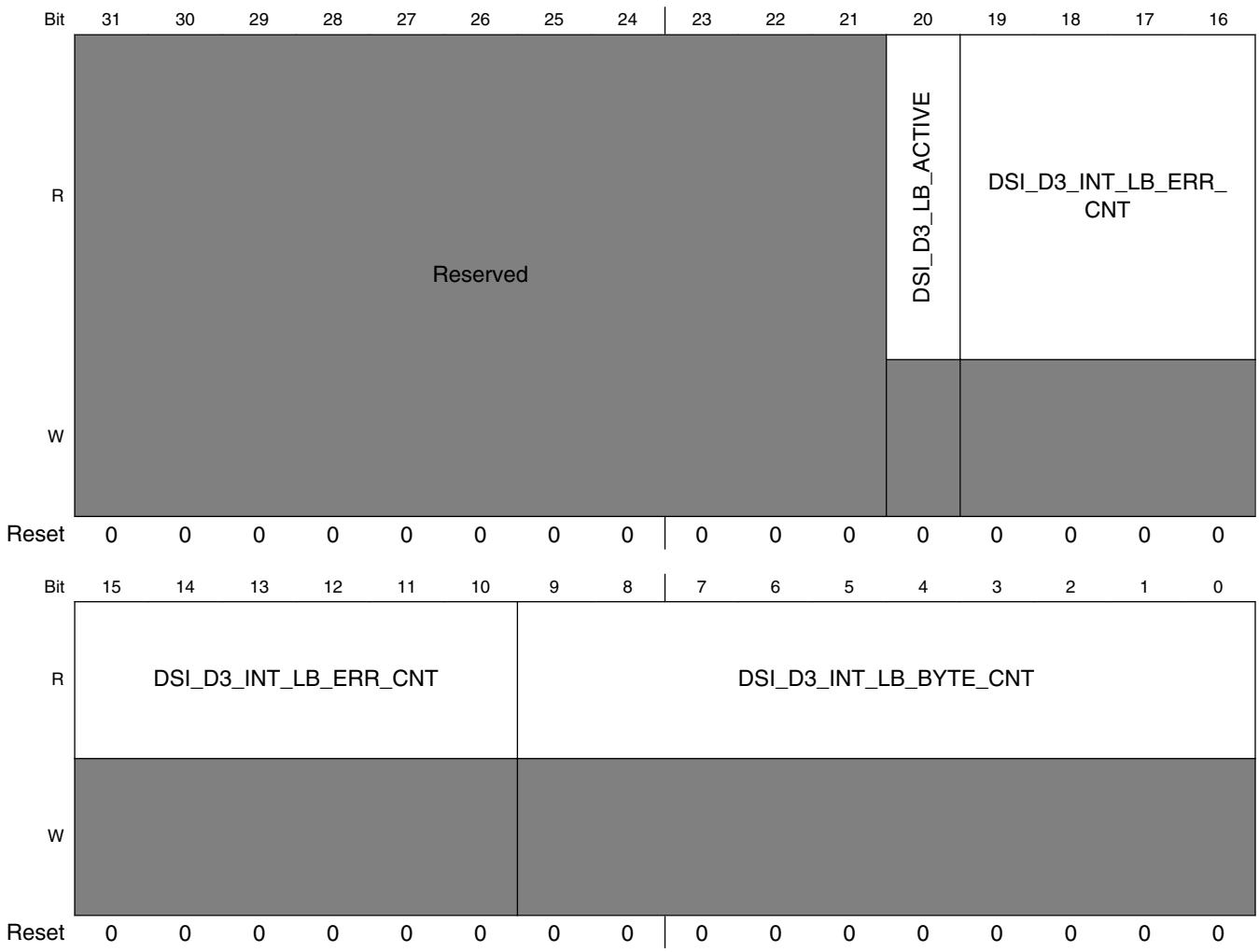
IOMUXC_GPR_GPR32 field descriptions (continued)

Field	Description
DSI_D2_INT_ LB_BYTE_CNT	for DSI test status

8.2.4.34 GPR33 General Purpose Register (IOMUXC_GPR_GPR33)

GPR Register

Address: 3034_0000h base + 84h offset = 3034_0084h



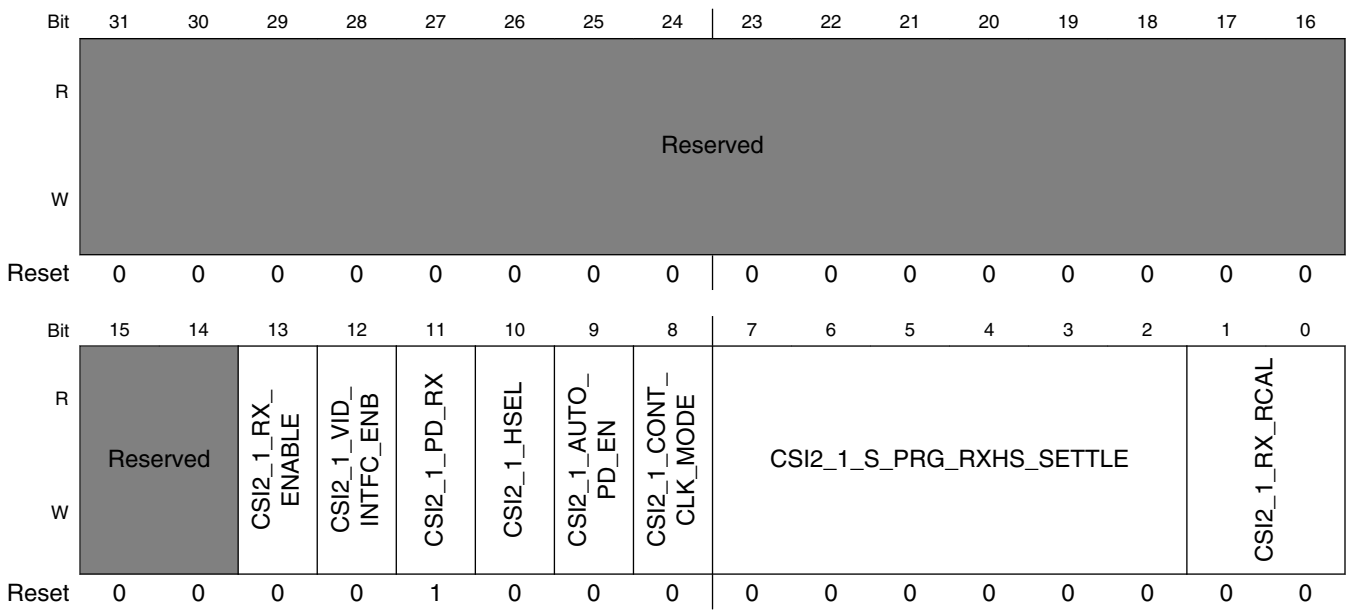
IOMUXC_GPR_GPR33 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 DSI_D3_LB_ACTIVE	for DSI test status
19–10 DSI_D3_INT_LB_ERR_CNT	for DSI test status
DSI_D3_INT_LB_BYTE_CNT	for DSI test status

8.2.4.35 GPR34 General Purpose Register (IOMUXC_GPR_GPR34)

GPR Register

Address: 3034_0000h base + 88h offset = 3034_0088h



IOMUXC_GPR_GPR34 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 CSI2_1_RX_ENABLE	MIPI CSI2_1 Controller Receive Enable, active high. When deasserted, the RX controller will pause data reception at the next packet boundary. Since the CSI-2 protocol does not allow the Receiver to pause data, when rx_enable is deasserted the RX Controller still receives data from the RX DPHY but it does not forward the receive packets to the user interface but instead discards the received data. When rx_enable

Table continues on the next page...

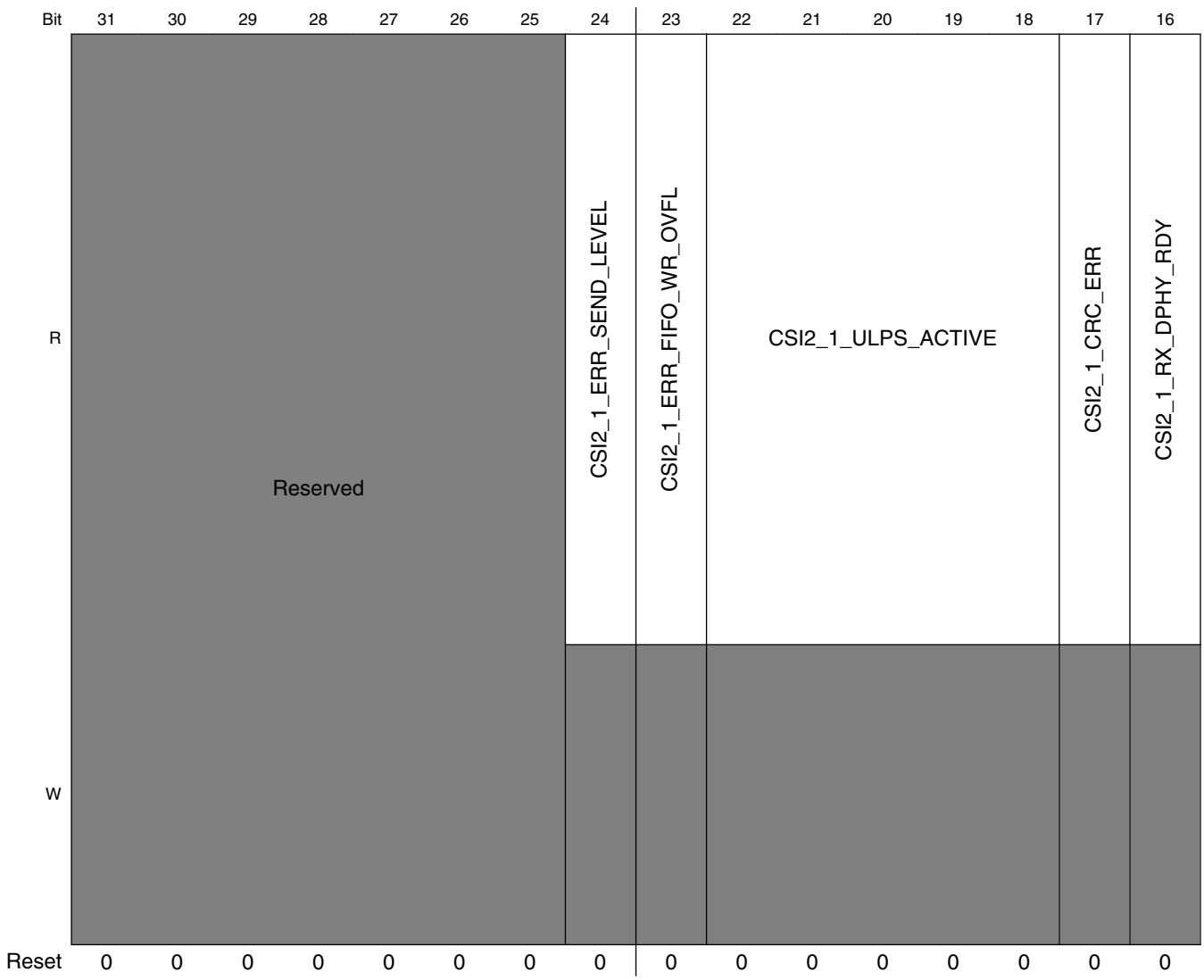
IOMUXC_GPR_GPR34 field descriptions (continued)

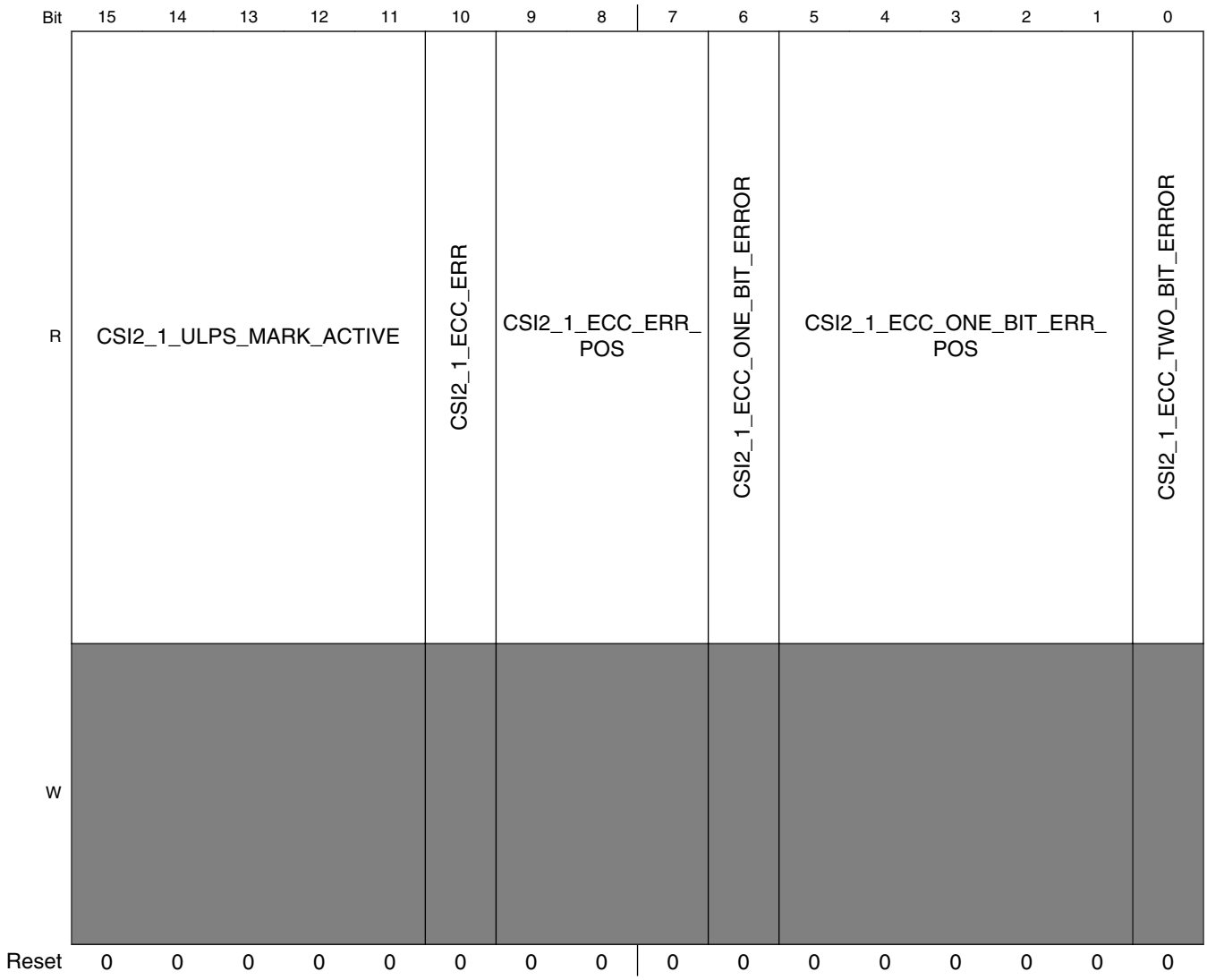
Field	Description
	is asserted, the RX controller will wait for the start of the next packet before allowing data to be sent out over the user interface.
12 CSI2_1_VID_INTFC_ENB	MIPI CSI2_1 controller video interface enable. 0 Disable 1 Enable
11 CSI2_1_PD_RX	MIPI CSI2_1 D-PHY Power Down input.When high, all blocks are powered down.
10 CSI2_1_HSEL	MIPI CSI2_1 D-PHY High Speed Select 0 1.0 Gbps operation 1 1.2 Gbps operation
9 CSI2_1_AUTO_PD_EN	MIPI CSI2_1 D-PHY Power down inactive lanes reported by CFG_NUM_LANES input bus. 0 Inactive lanes are powered up and driving LP11. 1 Inactive lanes are powered down.
8 CSI2_1_CONT_CLK_MODE	MIPI CSI2_1 D-PHY slave clock lane feature enable to maintain HS reception state during continuous clock mode operation. 0 Feature disable 1 Feature enabled
7–2 CSI2_1_S_PRG_RXHS_SETTLE	MIPI CSI2_1 D-PHY program T_HS_SETTLE bits. HS-RX waits for Time-out T_HS_SETTLE in order to neglect transition effects.
CSI2_1_RX_RCAL	MIPI CSI2_1 D-PHY On-chip termination control bits for manual calibration of HS-RX. 00 25% higher than mid-range. Highest impedance setting. 01 15% higher than mid-range. 10 Mid-range impedance setting. 11 25% lower than mid-range. Lowest impedance setting.

8.2.4.36 GPR35 General Purpose Register (IOMUXC_GPR_GPR35)

GPR Register

Address: 3034_0000h base + 8Ch offset = 3034_008Ch





IOMUXC_GPR_GPR35 field descriptions

Field	Description
31–25 -	This field is reserved. Reserved
24 CSI2_1_ERR_SEND_LEVEL	For MIPI CSI2_1 Controller, this flag indicates the video interface pixel fifo did not accumulate enough sample to trigger its send level setting before the end of an image data packet was detected. This flag would generally mean that the cfg_vid_p_fifo_send_level is set to high.
23 CSI2_1_ERR_FIFO_WR_OVFL	For MIPI CSI2_1 Controller, this flag indicates the video interface pixel fifo overflowed on a pixel write. This means the output video port is not removing pixels quickly enough to keep up with the input data rate. This flag would generally mean the clk_vid rate is too low.
22–18 CSI2_1_ULPS_ACTIVE	MIPI CSI2_1 Controller Receive Ultra Low Power State active. Bits assert high when corresponding clock or data lane is in ULPS mode. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1

Table continues on the next page...

IOMUXC_GPR_GPR35 field descriptions (continued)

Field	Description
	Bit [3] – Data Lane 2 Bit [4] – Data Lane 3
17 CSI2_1_CRC_ERR	MIPI CSI2_1 Controller CRC status. Asserts high when the CRC calculated on the received data does not match the CRC the transmitter sent at the end of the packet. Valid when the eop_out output asserts.
16 CSI2_1_RX_DPHY_RDY	MIPI CSI2_1 D-PHY ready status 0 Not ready 1 Ready
15–11 CSI2_1_ULPS_MARK_ACTIVE	MIPI CSI2_1 Controller Receive Ultra Low Power State Mark status. Bits assert high when corresponding clock or data lane has exited ULPS and entered Mark-1 state. Lanes will remain in Mark-1 state for a minimum of 1ms as per the MIPI DPHY specification. While in Mark-1 state, no other active, like High Speed data transmission, is allowed. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1 Bit [3] – Data Lane 2 Bit [4] – Data Lane 3
10 CSI2_1_ECC_ERR	MIPI CSI2_1: Error detected in the ECC bits. This signal is no longer supported.
9–7 CSI2_1_ECC_ERR_POS	MIPI CSI2_1: Position of the corrected single bit error in the packet header. Valid when pkt_hdr_valid is asserted high.
6 CSI2_1_ECC_ONE_BIT_ERROR	MIPI CSI2_1: Two packet header bit errors were detected and not corrected, active high and is valid when pkt_hdr_is high.
5–1 CSI2_1_ECC_ONE_BIT_ERR_POS	MIPI CSI2_1: Position of the corrected single bit error in the packet header. Valid when pkt_hdr_valid is asserted high.
0 CSI2_1_ECC_TWO_BIT_ERROR	MIPI CSI2_1: Two packet header bit errors were detected and not corrected, active high and is valid when pkt_hdr_is high.

8.2.4.37 GPR36 General Purpose Register (IOMUXC_GPR_GPR36)

GPR Register

Address: 3034_0000h base + 90h offset = 3034_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI2_1_TEST_PATTERN																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

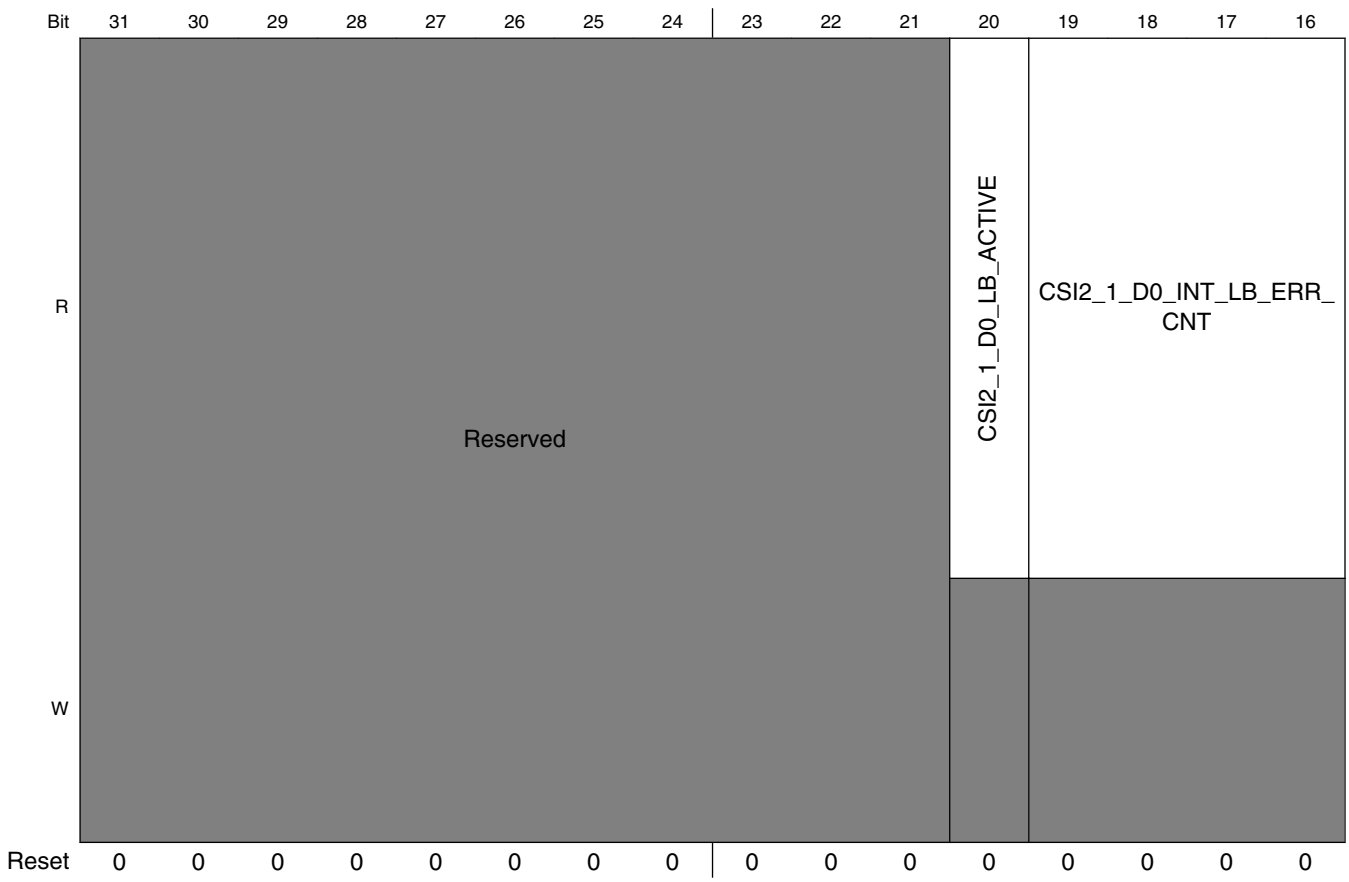
IOMUXC_GPR_GPR36 field descriptions

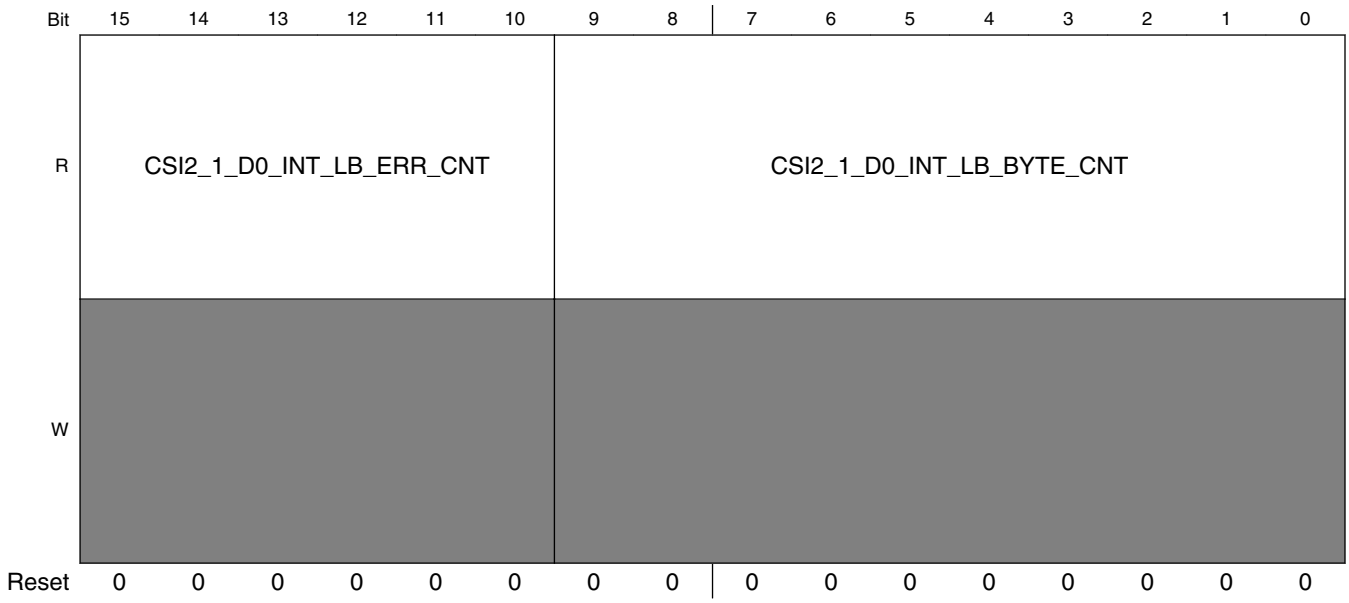
Field	Description
CSI2_1_TEST_PATTERN	for CSI2_1 test control

8.2.4.38 GPR37 General Purpose Register (IOMUXC_GPR_GPR37)

GPR Register

Address: 3034_0000h base + 94h offset = 3034_0094h





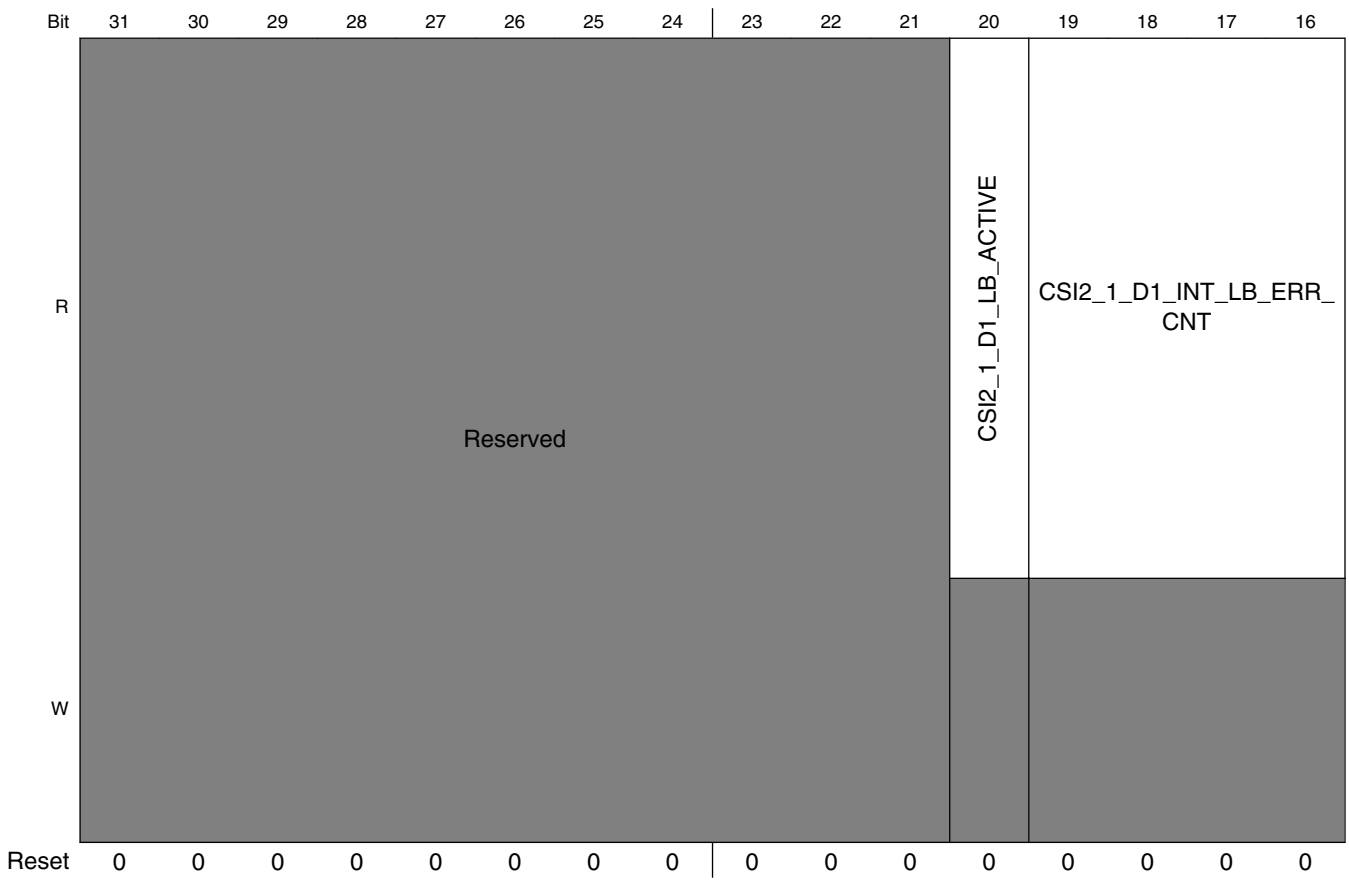
IOMUXC_GPR_GPR37 field descriptions

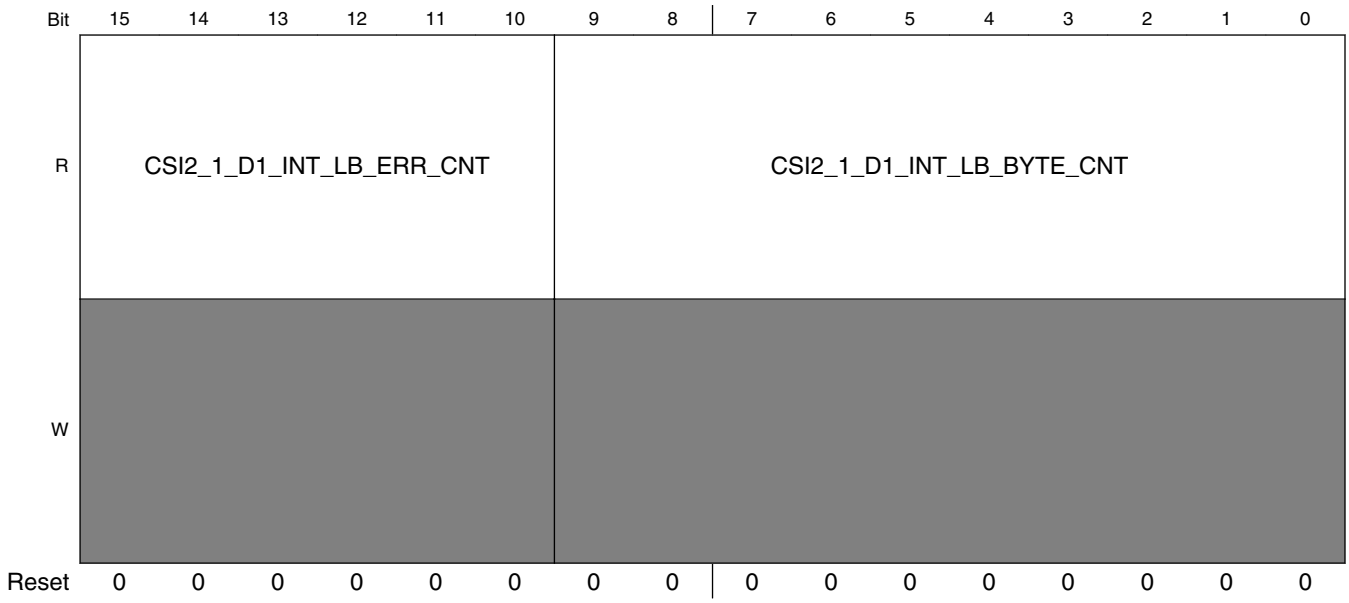
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_1_D0_LB_ACTIVE	for DSI test status
19–10 CSI2_1_D0_INT_LB_ERR_CNT	for DSI test status
CSI2_1_D0_INT_LB_BYTE_CNT	for DSI test status

8.2.4.39 GPR38 General Purpose Register (IOMUXC_GPR_GPR38)

GPR Register

Address: 3034_0000h base + 98h offset = 3034_0098h





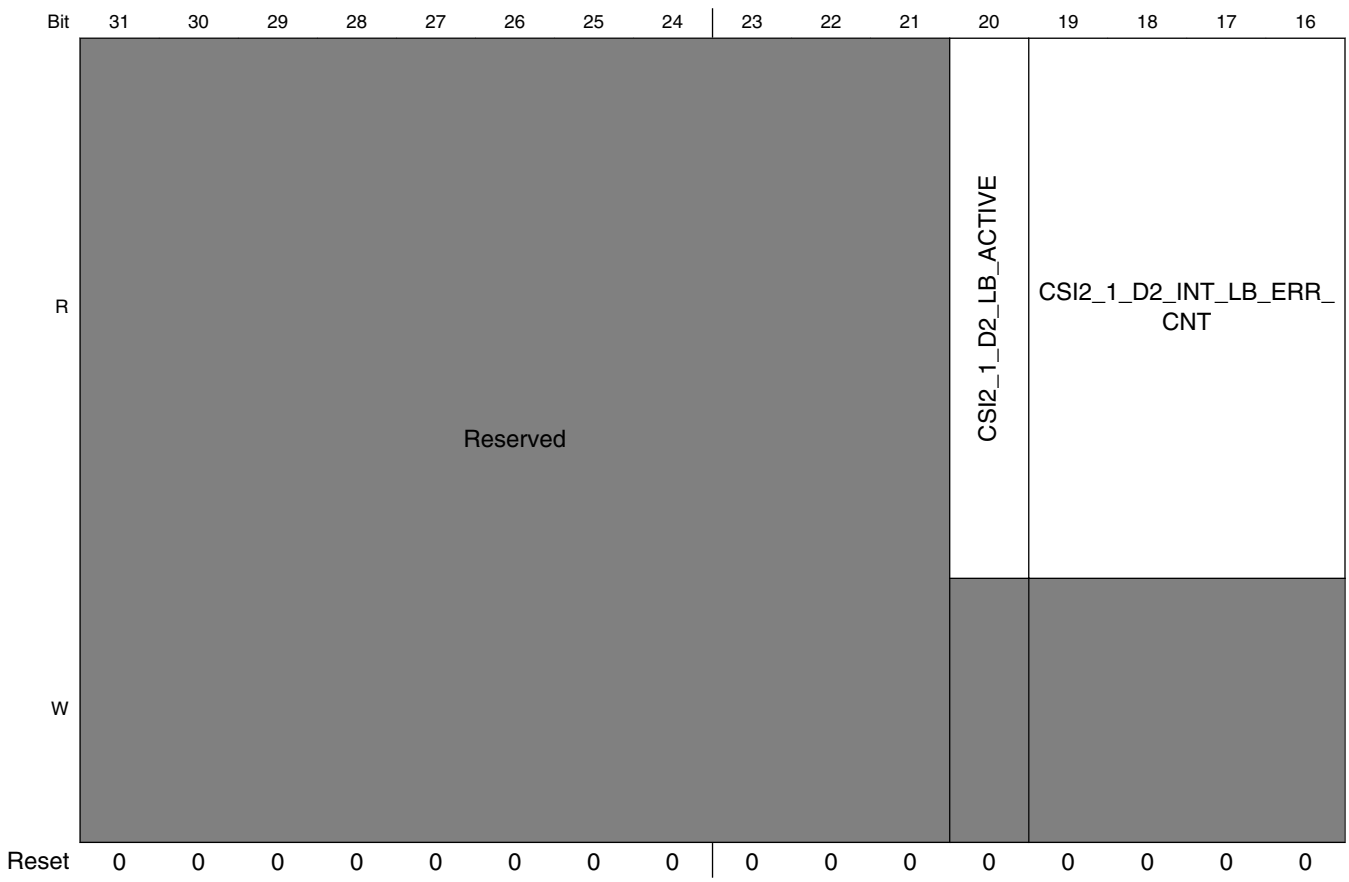
IOMUXC_GPR_GPR38 field descriptions

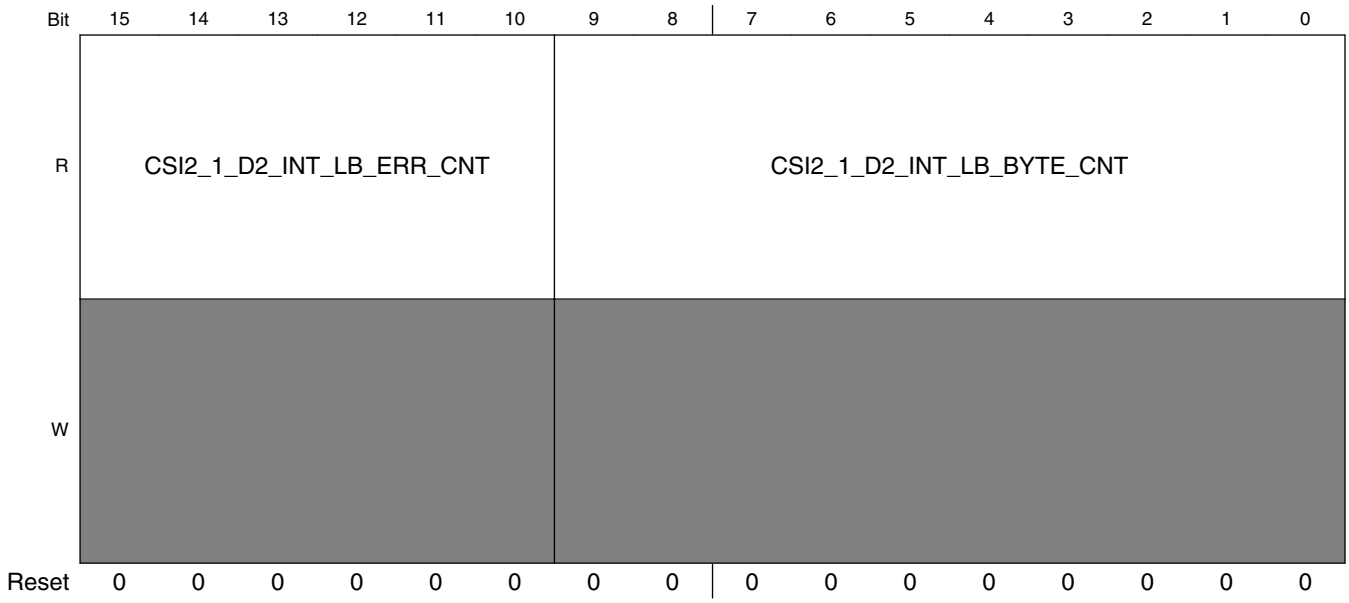
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_1_D1_LB_ACTIVE	for DSI test status
19–10 CSI2_1_D1_INT_LB_ERR_CNT	for DSI test status
CSI2_1_D1_INT_LB_BYTE_CNT	for DSI test status

8.2.4.40 GPR39 General Purpose Register (IOMUXC_GPR_GPR39)

GPR Register

Address: 3034_0000h base + 9Ch offset = 3034_009Ch





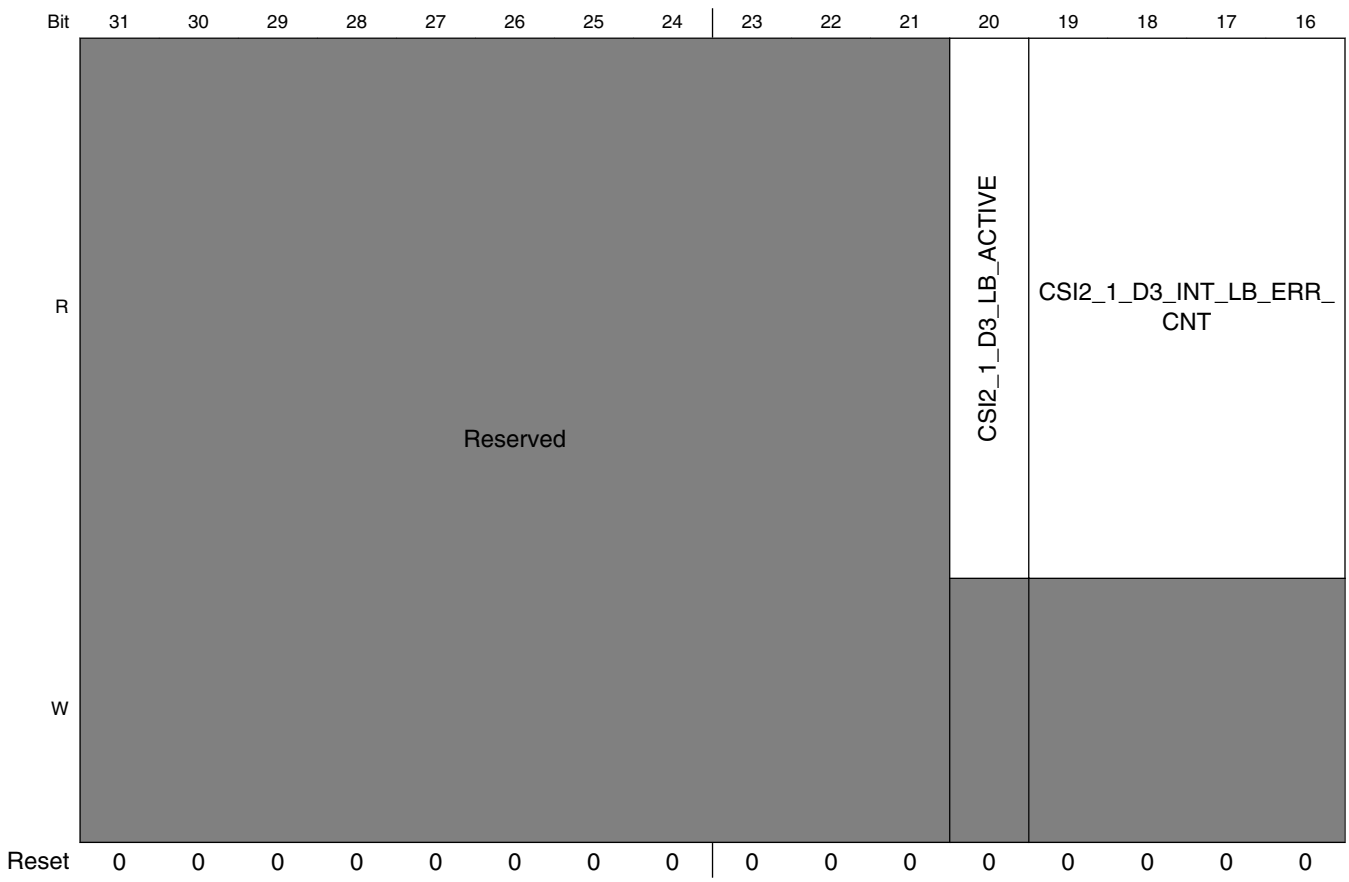
IOMUXC_GPR_GPR39 field descriptions

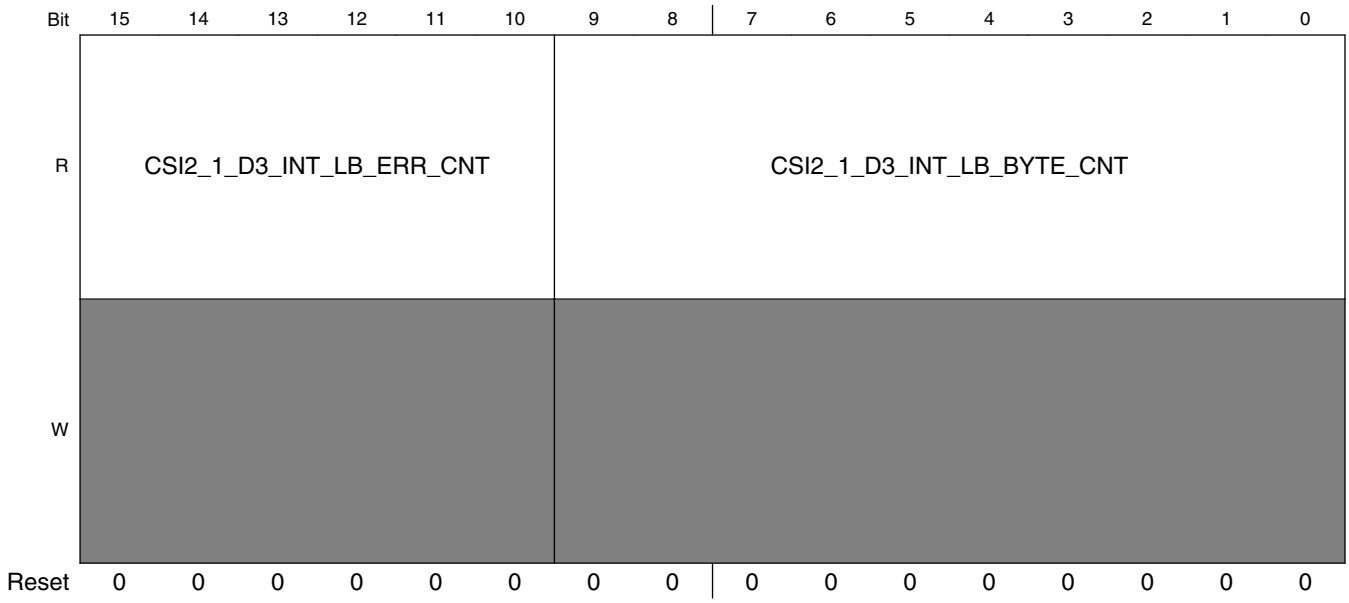
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_1_D2_LB_ACTIVE	for DSI test status
19–10 CSI2_1_D2_INT_LB_ERR_CNT	for DSI test status
CSI2_1_D2_INT_LB_BYTE_CNT	for DSI test status

8.2.4.41 GPR40 General Purpose Register (IOMUXC_GPR_GPR40)

GPR Register

Address: 3034_0000h base + A0h offset = 3034_00A0h





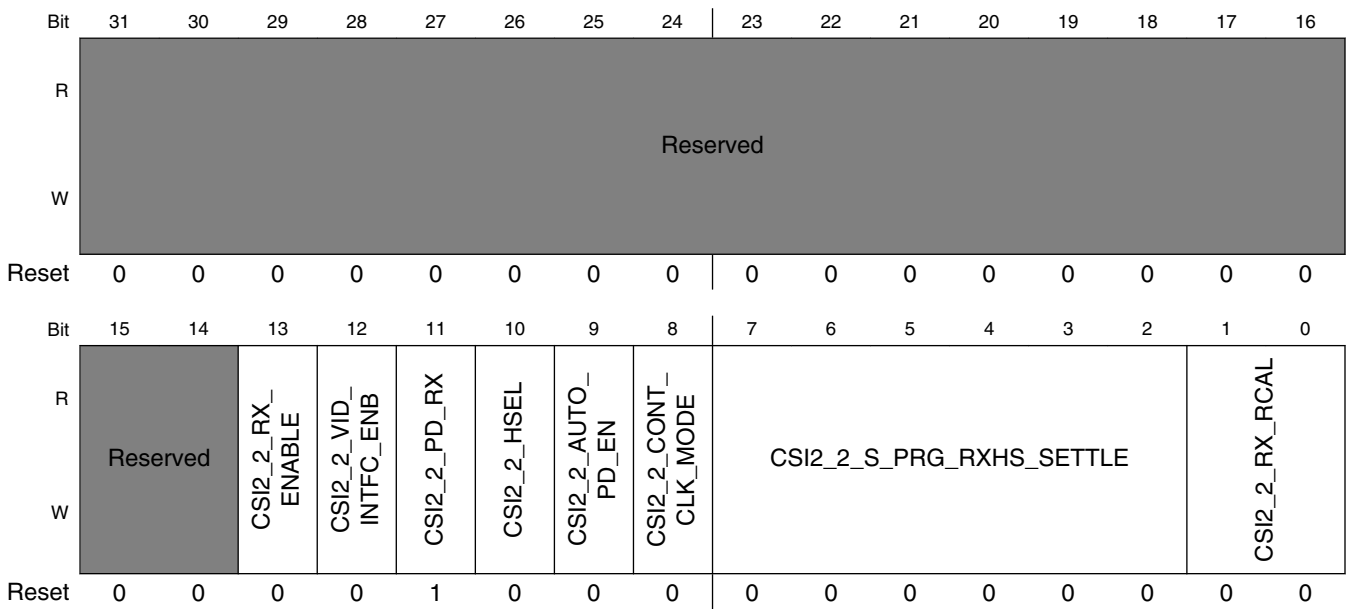
IOMUXC_GPR_GPR40 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_1_D3_LB_ACTIVE	for DSI test status
19–10 CSI2_1_D3_INT_LB_ERR_CNT	for DSI test status
CSI2_1_D3_INT_LB_BYTE_CNT	for DSI test status

8.2.4.42 GPR41 General Purpose Register (IOMUXC_GPR_GPR41)

GPR Register

Address: 3034_0000h base + A4h offset = 3034_00A4h



IOMUXC_GPR_GPR41 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13 CSI2_2_RX_ENABLE	MIPI CSI2_2 Controller Receive Enable, active high. When deasserted, the RX controller will pause data reception at the next packet boundary. Since the CSI-2 protocol does not allow the Receiver to pause data, when rx_enable is deasserted the RX Controller still receives data from the RX DPHY but it does not forward the receive packets to the user interface but instead discards the received data. When rx_enable is asserted, the RX controller will wait for the start of the next packet before allowing data to be sent out over the user interface.
12 CSI2_2_VID_INTFC_ENB	MIPI CSI2_2 controller video interface enable. 0 Disable 1 Enable
11 CSI2_2_PD_RX	MIPI CSI2_2 D-PHY Power Down input.When high, all blocks are powered down.
10 CSI2_2_HSEL	MIPI CSI2_2 D-PHY High Speed Select 0 1.0 Gbps operation 1 1.2 Gbps operation

Table continues on the next page...

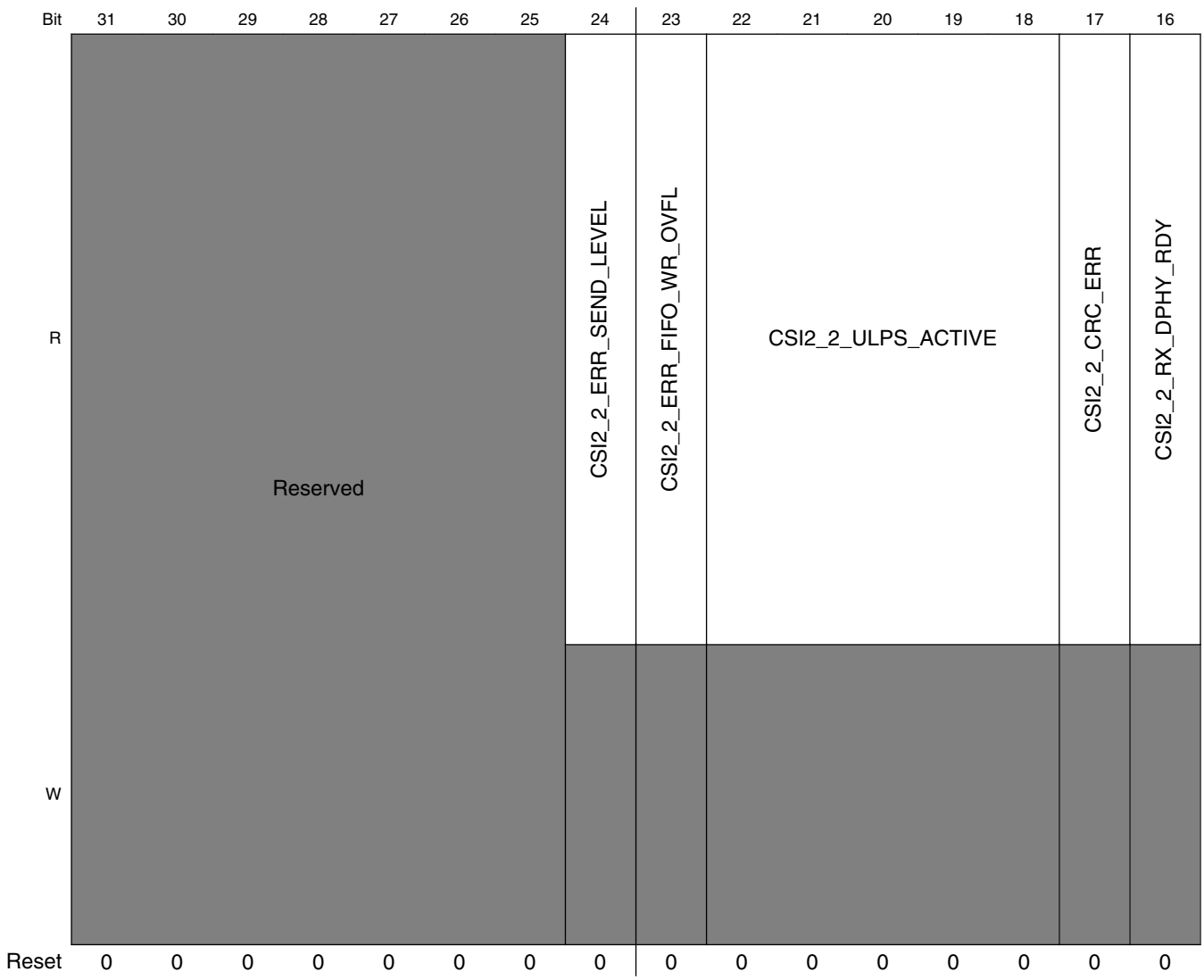
IOMUXC_GPR_GPR41 field descriptions (continued)

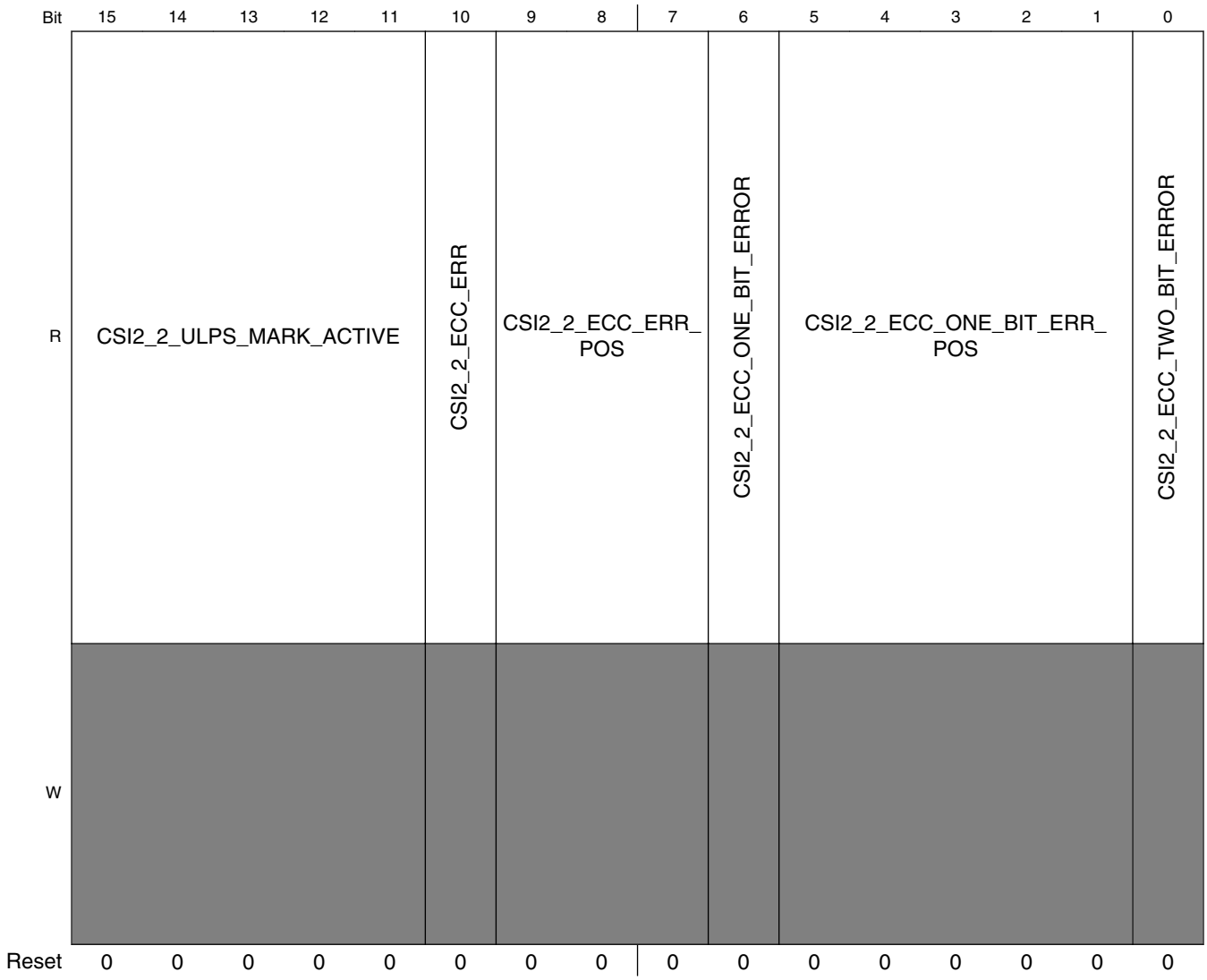
Field	Description
9 CSI2_2_AUTO_PD_EN	MIPI CSI2_2 D-PHY Power down inactive lanes reported by CFG_NUM_LANES input bus. 0 Inactive lanes are powered up and driving LP11. 1 Inactive lanes are powered down.
8 CSI2_2_CONT_CLK_MODE	MIPI CSI2_2 D-PHY slave clock lane feature enable to maintain HS reception state during continuous clock mode operation. 0 Feature disable 1 Feature enable
7–2 CSI2_2_S_PRG_RXHS_SETTLE	MIPI CSI2_2 D-PHY program T_HS_SETTLE bits. HS-RX waits for Time-out T_HS_SETTLE in order to neglect transition effects.
CSI2_2_RX_RCAL	MIPI CSI2_2 D-PHY On-chip termination control bits for manual calibration of HS-RX. 00 25% higher than mid-range. Highest impedance setting. 01 15% higher than mid-range. 10 Mid-range impedance setting. 11 25% lower than mid-range. Lowest impedance setting.

8.2.4.43 GPR42 General Purpose Register (IOMUXC_GPR_GPR42)

GPR Register

Address: 3034_0000h base + A8h offset = 3034_00A8h





IOMUXC_GPR_GPR42 field descriptions

Field	Description
31–25 -	This field is reserved. Reserved
24 CSI2_2_ERR_SEND_LEVEL	For MIPI CSI2_2 Controller, this flag indicates the video interface pixel fifo did not accumulate enough sample to trigger its send level setting before the end of an image data packet was detected. This flag would generally mean that the cfg_vid_p_fifo_send_level is set to high.
23 CSI2_2_ERR_FIFO_WR_OVFL	For MIPI CSI2_2 Controller, this flag indicates the video interface pixel fifo overflowed on a pixel write. This means the output video port is not removing pixels quickly enough to keep up with the input data rate. This flag would generally mean the clk_vid rate is too low.
22–18 CSI2_2_ULPS_ACTIVE	MIPI CSI2_2 Controller Receive Ultra Low Power State active. Bits assert high when corresponding clock or data lane is in ULPS mode. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1

Table continues on the next page...

IOMUXC_GPR_GPR42 field descriptions (continued)

Field	Description
	Bit [3] – Data Lane 2 Bit [4] – Data Lane 3
17 CSI2_2_CRC_ERR	MIPI CSI2_2 Controller CRC status. Asserts high when the CRC calculated on the received data does not match the CRC the transmitter sent at the end of the packet. Valid when the eop_out output asserts.
16 CSI2_2_RX_DPHY_RDY	MIPI CSI2_2 D-PHY ready status 0 Not ready 1 Ready
15–11 CSI2_2_ULPS_MARK_ACTIVE	MIPI CSI2_2 Controller Receive Ultra Low Power State Mark status. Bits assert high when corresponding clock or data lane has exited ULPS and entered Mark-1 state. Lanes will remain in Mark-1 state for a minimum of 1ms as per the MIPI DPHY specification. While in Mark-1 state, no other active, like High Speed data transmission, is allowed. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1 Bit [3] – Data Lane 2 Bit [4] – Data Lane 3
10 CSI2_2_ECC_ERR	MIPI CSI2_2: Error detected in the ECC bits. This signal is no longer supported.
9–7 CSI2_2_ECC_ERR_POS	MIPI CSI2_2:Position of the corrected single bit error in the packet header. Valid when pkt_hdr_valid is asserted high.
6 CSI2_2_ECC_ONE_BIT_ERROR	MIPI CSI2_2:Two packet header bit errors were detected and not corrected, active high and is valid when pkt_hdr_is high.
5–1 CSI2_2_ECC_ONE_BIT_ERR_POS	MIPI CSI2_2:Position of the corrected single bit error in the packet header. Valid when pkt_hdr_valid is asserted high.
0 CSI2_2_ECC_TWO_BIT_ERROR	MIPI CSI2_2:Two packet header bit errors were detected and not corrected, active high and is valid when pkt_hdr_is high.

8.2.4.44 GPR43 General Purpose Register (IOMUXC_GPR_GPR43)

GPR Register

Address: 3034_0000h base + ACh offset = 3034_00ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSI2_2_TEST_PATTERN																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

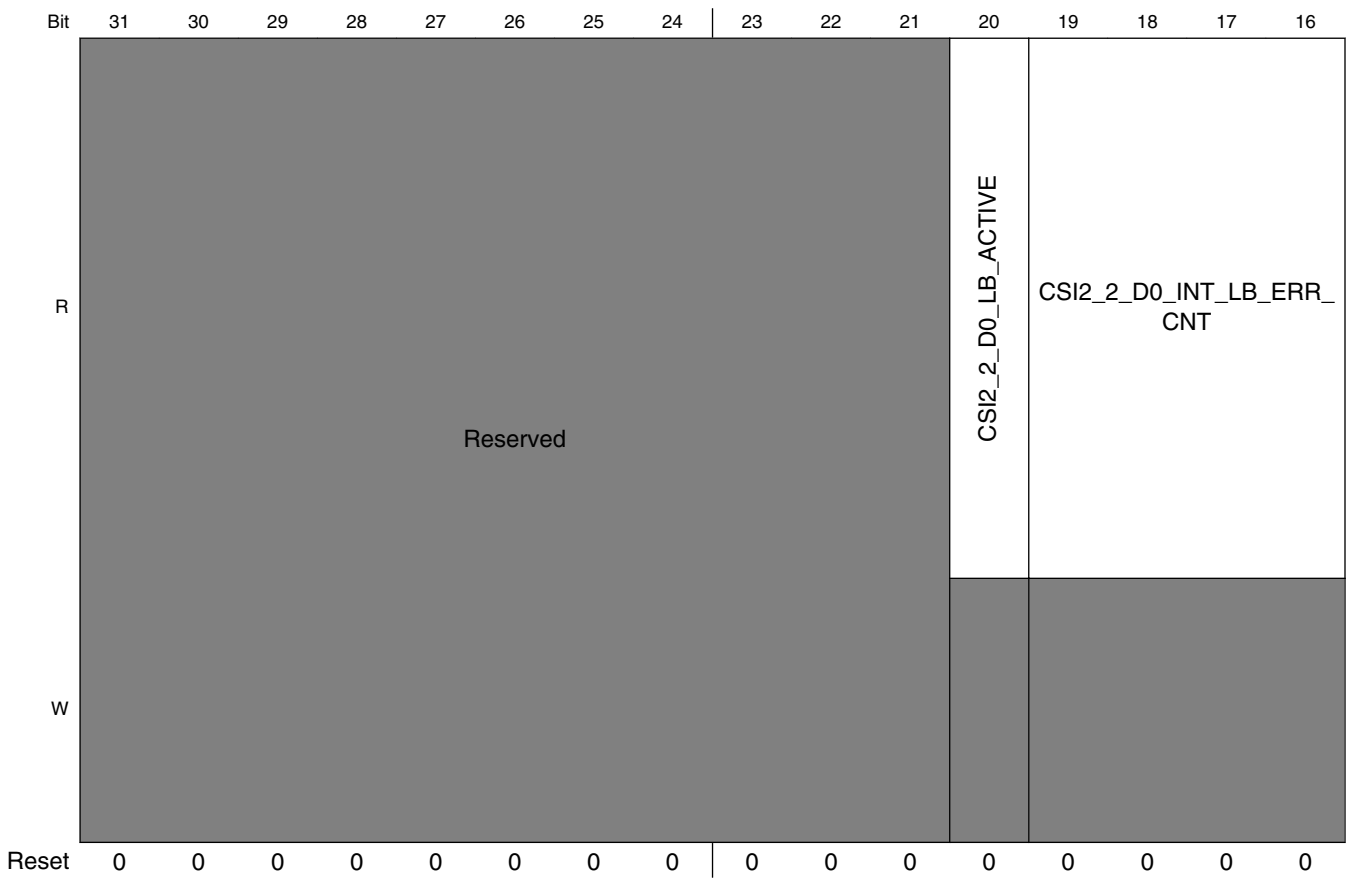
IOMUXC_GPR_GPR43 field descriptions

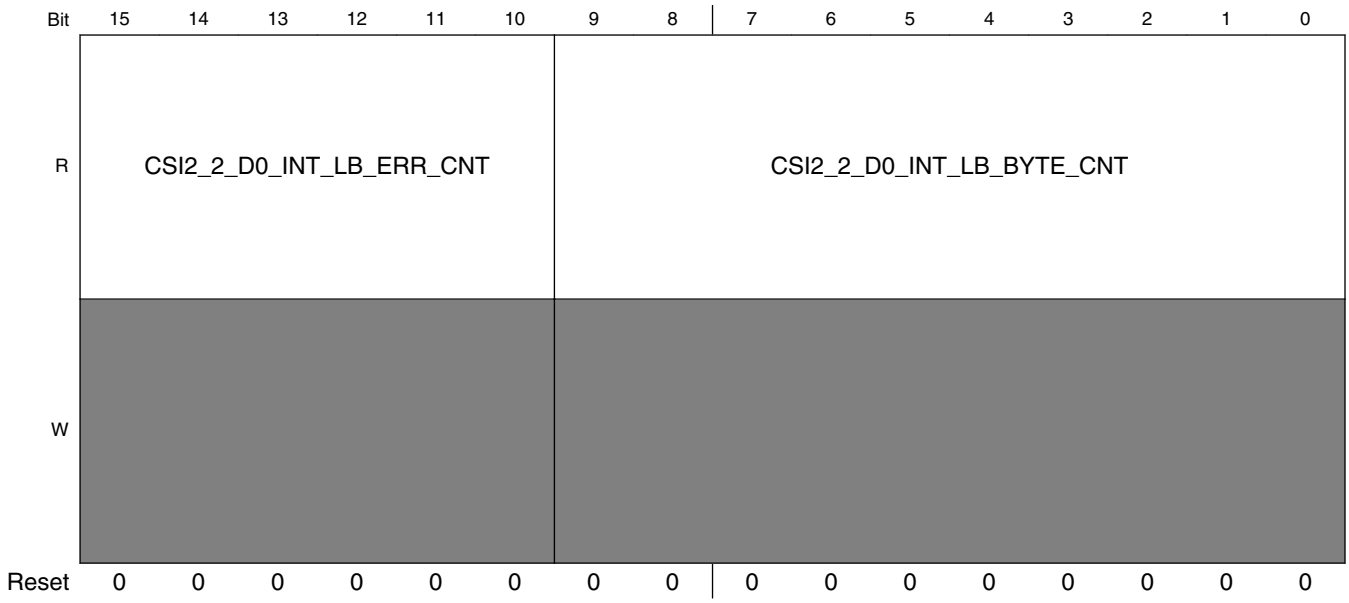
Field	Description
CSI2_2_TEST_PATTERN	for CSI2_2 test control

8.2.4.45 GPR44 General Purpose Register (IOMUXC_GPR_GPR44)

GPR Register

Address: 3034_0000h base + B0h offset = 3034_00B0h





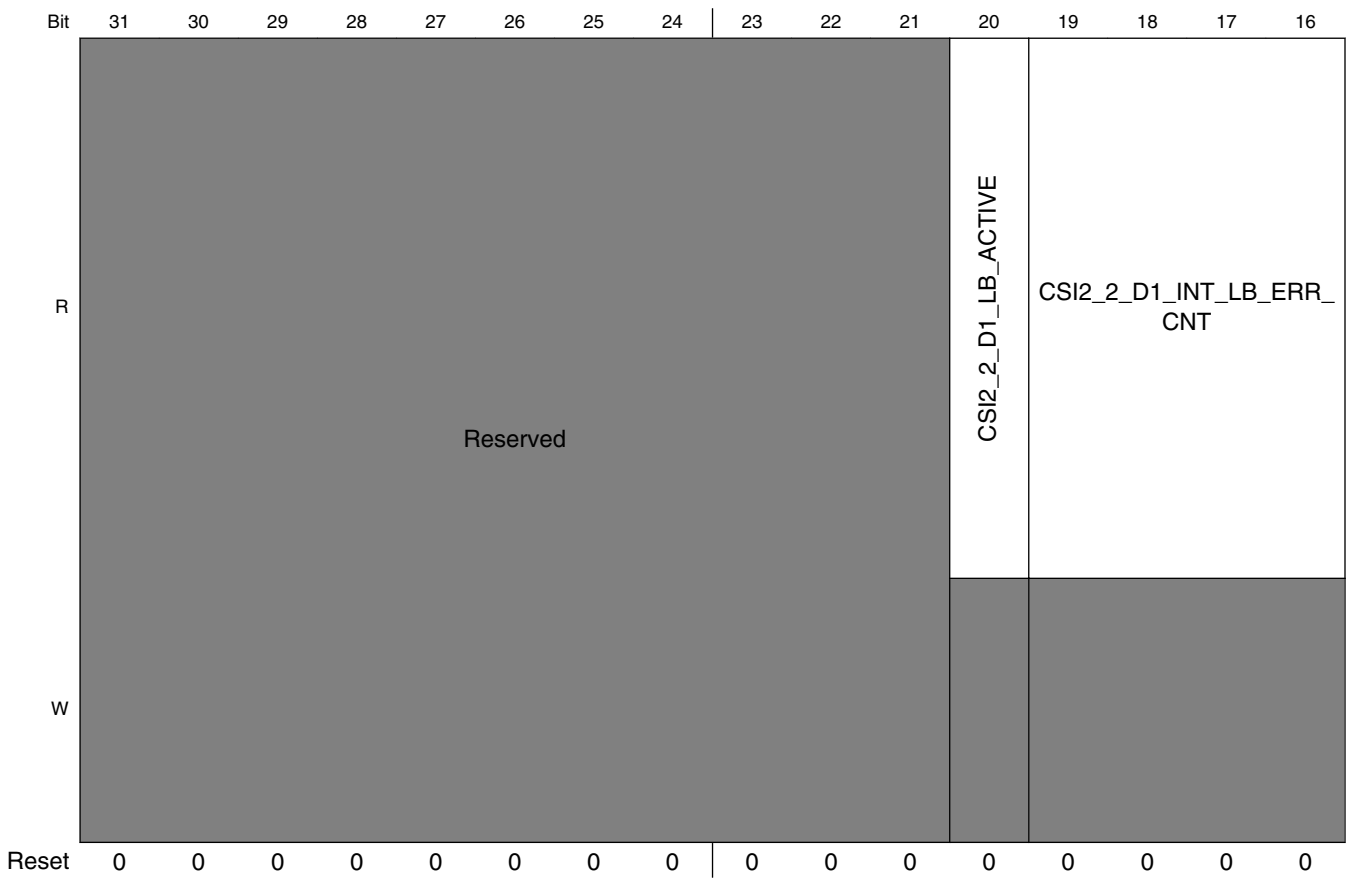
IOMUXC_GPR_GPR44 field descriptions

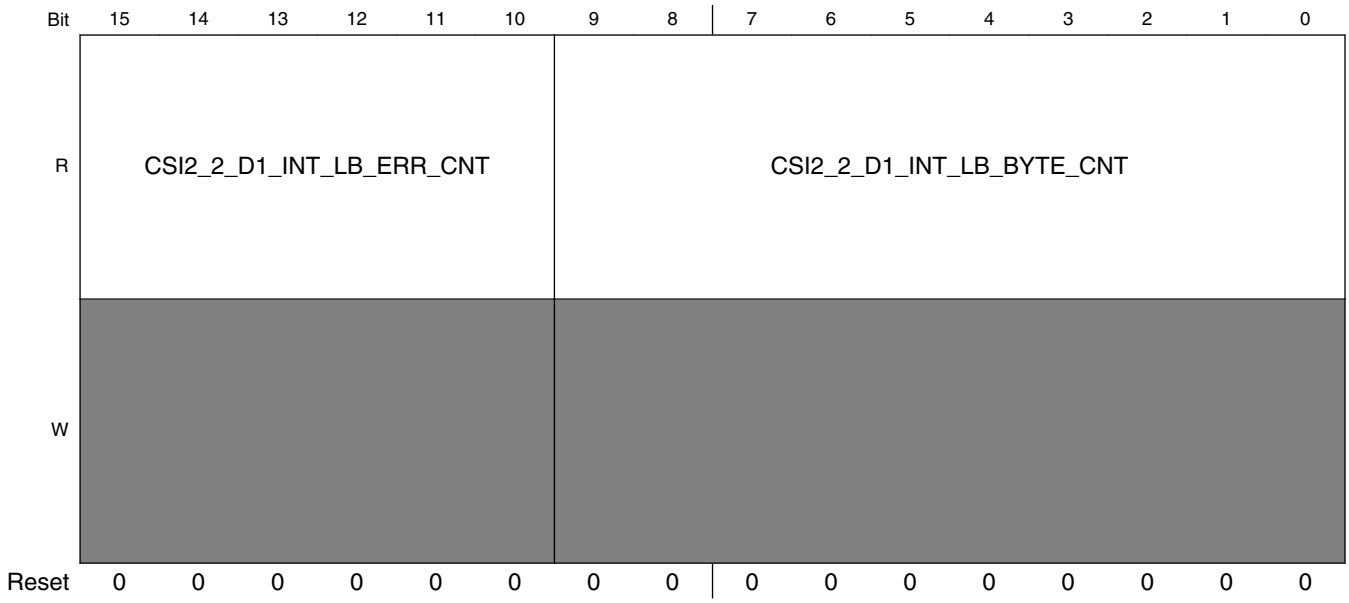
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_2_D0_LB_ACTIVE	for DSI test status
19–10 CSI2_2_D0_INT_LB_ERR_CNT	for DSI test status
CSI2_2_D0_INT_LB_BYTE_CNT	for DSI test status

8.2.4.46 GPR45 General Purpose Register (IOMUXC_GPR_GPR45)

GPR Register

Address: 3034_0000h base + B4h offset = 3034_00B4h





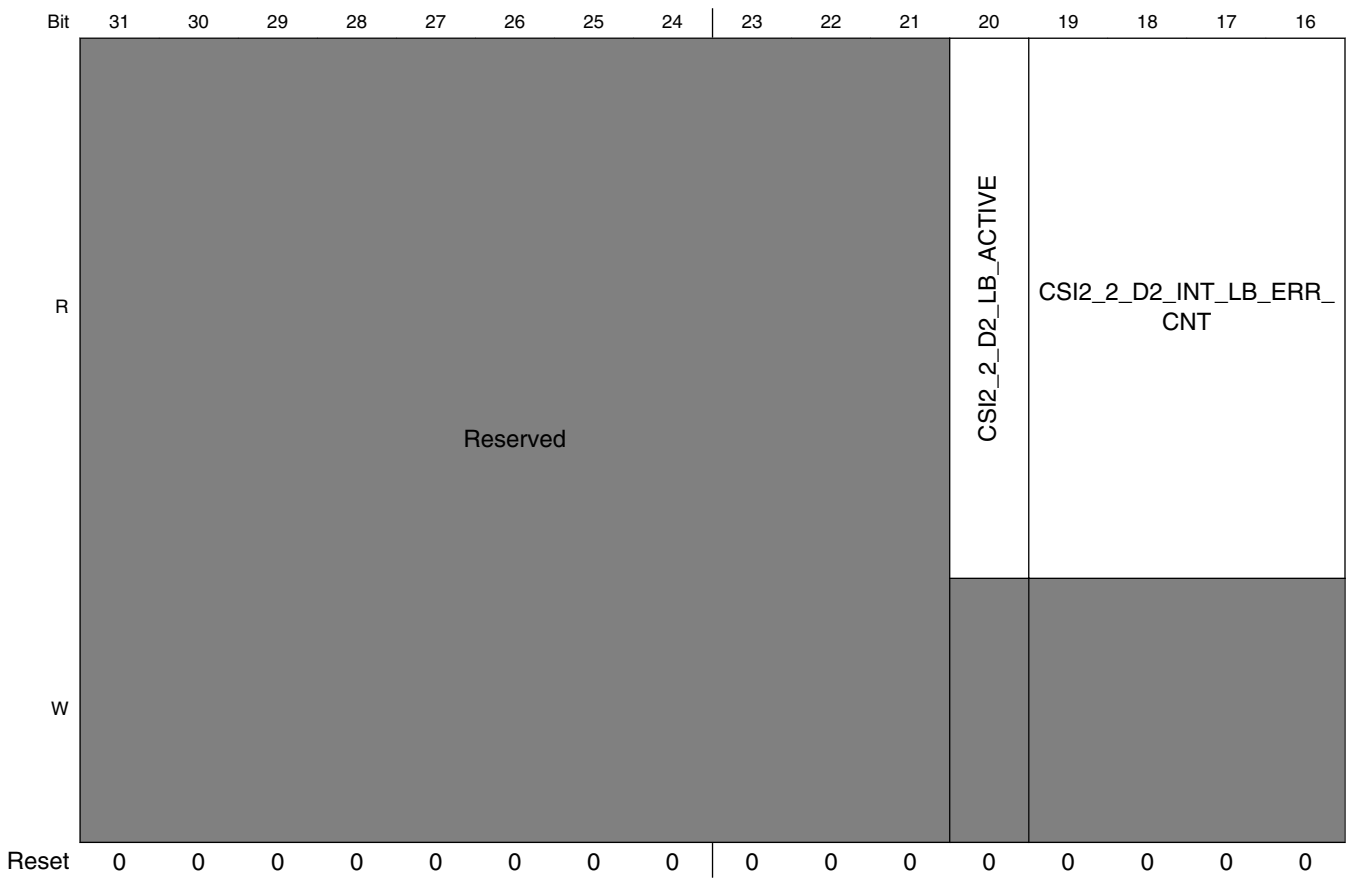
IOMUXC_GPR_GPR45 field descriptions

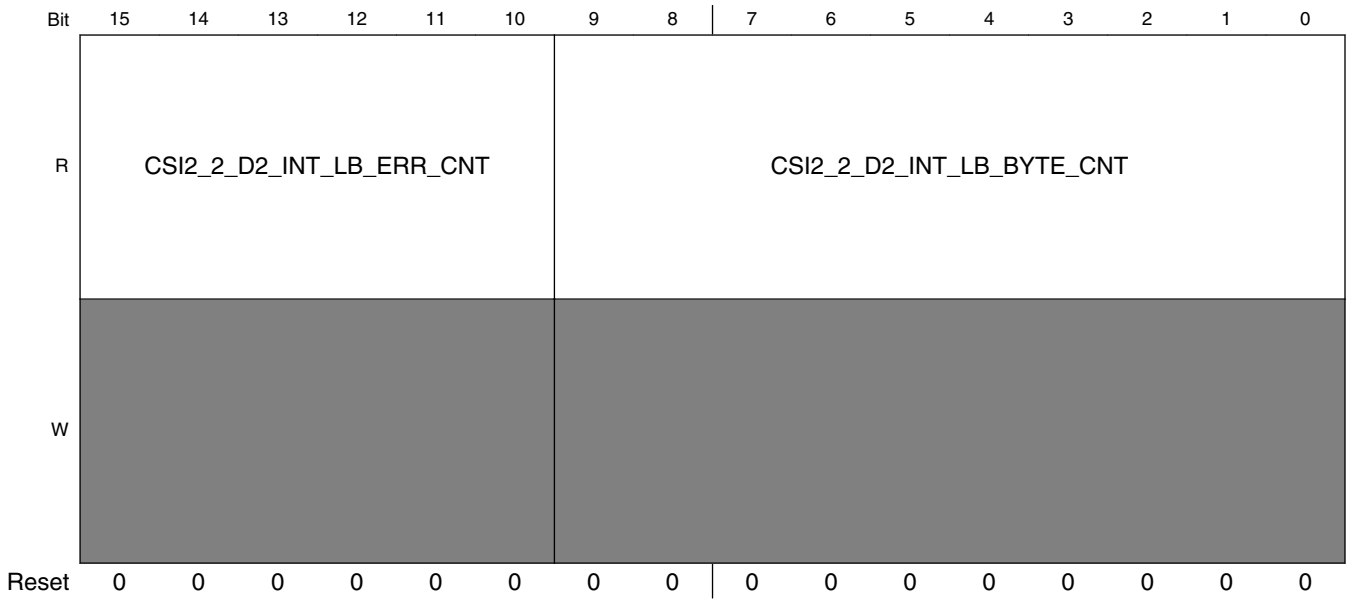
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_2_D1_LB_ACTIVE	for DSI test status
19–10 CSI2_2_D1_INT_LB_ERR_CNT	for DSI test status
CSI2_2_D1_INT_LB_BYTE_CNT	for DSI test status

8.2.4.47 GPR46 General Purpose Register (IOMUXC_GPR_GPR46)

GPR Register

Address: 3034_0000h base + B8h offset = 3034_00B8h





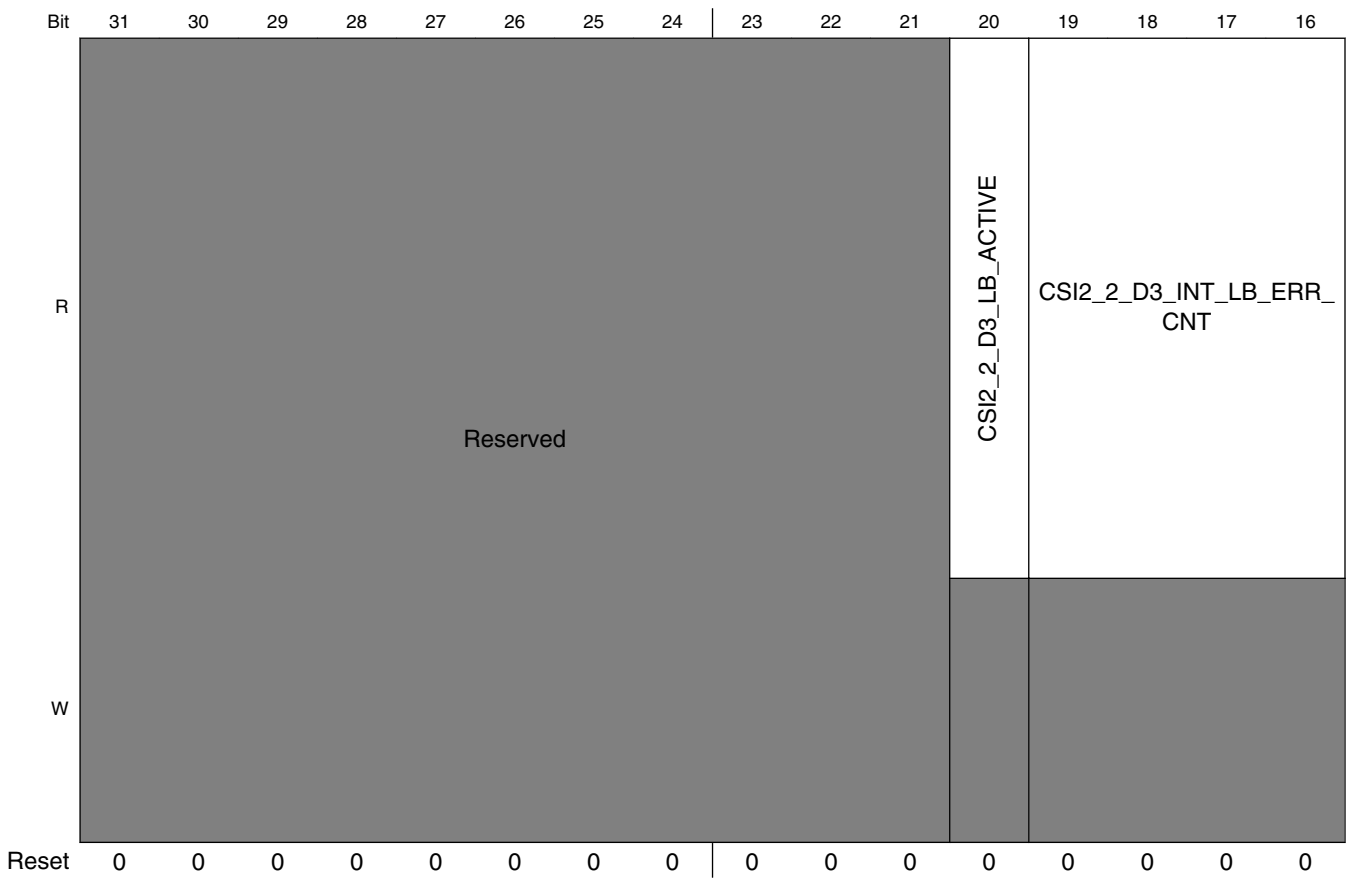
IOMUXC_GPR_GPR46 field descriptions

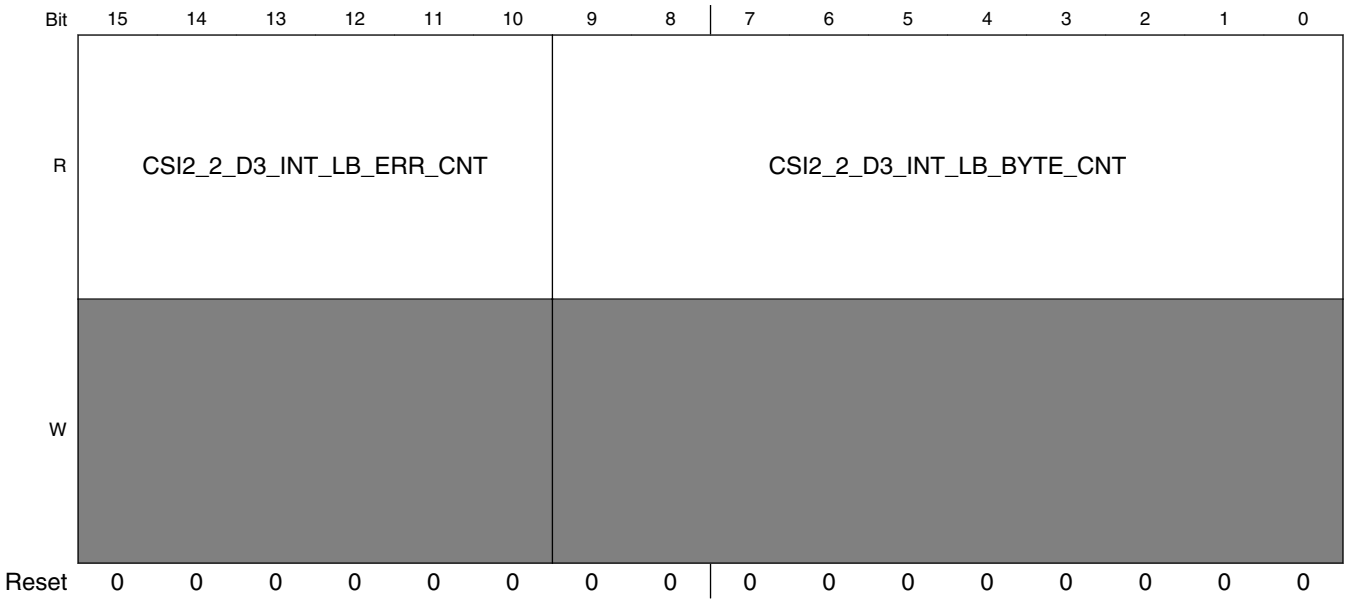
Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_2_D2_LB_ACTIVE	for DSI test status
19–10 CSI2_2_D2_INT_LB_ERR_CNT	for DSI test status
CSI2_2_D2_INT_LB_BYTE_CNT	for DSI test status

8.2.4.48 GPR47 General Purpose Register (IOMUXC_GPR_GPR47)

GPR Register

Address: 3034_0000h base + BCh offset = 3034_00BCh





IOMUXC_GPR_GPR47 field descriptions

Field	Description
31–21 -	This field is reserved. Reserved
20 CSI2_2_D3_LB_ACTIVE	for DSI test status
19–10 CSI2_2_D3_INT_LB_ERR_CNT	for DSI test status
CSI2_2_D3_INT_LB_BYTE_CNT	for DSI test status

8.2.5 IOMUXC Memory Map/Register Definition

IOMUXC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0014	SW_MUX_CTL_PAD_PMIC_STBY_REQ SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_PMIC_STBY_REQ)	32	R/W	0000_0000h	8.2.5.1/ 1389
3033_0018	SW_MUX_CTL_PAD_PMIC_ON_REQ SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_PMIC_ON_REQ)	32	R/W	0000_0000h	8.2.5.2/ 1389
3033_001C	SW_MUX_CTL_PAD_ONOFF SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ONOFF)	32	R/W	0000_0000h	8.2.5.3/ 1390

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0020	SW_MUX_CTL_PAD_POR_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_POR_B)	32	R/W	0000_0000h	8.2.5.4/ 1391
3033_0024	SW_MUX_CTL_PAD_RTC_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_RTC_RESET_B)	32	R/W	0000_0000h	8.2.5.5/ 1391
3033_0028	SW_MUX_CTL_PAD_GPIO1_IO00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00)	32	R/W	0000_0000h	8.2.5.6/ 1392
3033_002C	SW_MUX_CTL_PAD_GPIO1_IO01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01)	32	R/W	0000_0000h	8.2.5.7/ 1393
3033_0030	SW_MUX_CTL_PAD_GPIO1_IO02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02)	32	R/W	0000_0000h	8.2.5.8/ 1395
3033_0034	SW_MUX_CTL_PAD_GPIO1_IO03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03)	32	R/W	0000_0000h	8.2.5.9/ 1396
3033_0038	SW_MUX_CTL_PAD_GPIO1_IO04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04)	32	R/W	0000_0000h	8.2.5.10/ 1397
3033_003C	SW_MUX_CTL_PAD_GPIO1_IO05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05)	32	R/W	0000_0000h	8.2.5.11/ 1398
3033_0040	SW_MUX_CTL_PAD_GPIO1_IO06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06)	32	R/W	0000_0000h	8.2.5.12/ 1399
3033_0044	SW_MUX_CTL_PAD_GPIO1_IO07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07)	32	R/W	0000_0000h	8.2.5.13/ 1400
3033_0048	SW_MUX_CTL_PAD_GPIO1_IO08 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08)	32	R/W	0000_0000h	8.2.5.14/ 1401
3033_004C	SW_MUX_CTL_PAD_GPIO1_IO09 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09)	32	R/W	0000_0000h	8.2.5.15/ 1402
3033_0050	SW_MUX_CTL_PAD_GPIO1_IO10 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10)	32	R/W	0000_0000h	8.2.5.16/ 1404
3033_0054	SW_MUX_CTL_PAD_GPIO1_IO11 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11)	32	R/W	0000_0000h	8.2.5.17/ 1405
3033_0058	SW_MUX_CTL_PAD_GPIO1_IO12 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12)	32	R/W	0000_0000h	8.2.5.18/ 1406
3033_005C	SW_MUX_CTL_PAD_GPIO1_IO13 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13)	32	R/W	0000_0000h	8.2.5.19/ 1407
3033_0060	SW_MUX_CTL_PAD_GPIO1_IO14 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14)	32	R/W	0000_0000h	8.2.5.20/ 1408
3033_0064	SW_MUX_CTL_PAD_GPIO1_IO15 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15)	32	R/W	0000_0000h	8.2.5.21/ 1409
3033_0068	SW_MUX_CTL_PAD_ENET_MDC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)	32	R/W	0000_0005h	8.2.5.22/ 1410
3033_006C	SW_MUX_CTL_PAD_ENET_MDIO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)	32	R/W	0000_0005h	8.2.5.23/ 1411
3033_0070	SW_MUX_CTL_PAD_ENET_TD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD3)	32	R/W	0000_0005h	8.2.5.24/ 1412
3033_0074	SW_MUX_CTL_PAD_ENET_TD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD2)	32	R/W	0000_0005h	8.2.5.25/ 1413

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0078	SW_MUX_CTL_PAD_ENET_TD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD1)	32	R/W	0000_0005h	8.2.5.26/ 1414
3033_007C	SW_MUX_CTL_PAD_ENET_TD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD0)	32	R/W	0000_0005h	8.2.5.27/ 1415
3033_0080	SW_MUX_CTL_PAD_ENET_TX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL)	32	R/W	0000_0005h	8.2.5.28/ 1416
3033_0084	SW_MUX_CTL_PAD_ENET_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TXC)	32	R/W	0000_0005h	8.2.5.29/ 1417
3033_0088	SW_MUX_CTL_PAD_ENET_RX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL)	32	R/W	0000_0005h	8.2.5.30/ 1418
3033_008C	SW_MUX_CTL_PAD_ENET_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RXC)	32	R/W	0000_0005h	8.2.5.31/ 1419
3033_0090	SW_MUX_CTL_PAD_ENET_RD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD0)	32	R/W	0000_0005h	8.2.5.32/ 1420
3033_0094	SW_MUX_CTL_PAD_ENET_RD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD1)	32	R/W	0000_0005h	8.2.5.33/ 1421
3033_0098	SW_MUX_CTL_PAD_ENET_RD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD2)	32	R/W	0000_0005h	8.2.5.34/ 1422
3033_009C	SW_MUX_CTL_PAD_ENET_RD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD3)	32	R/W	0000_0005h	8.2.5.35/ 1423
3033_00A0	SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)	32	R/W	0000_0005h	8.2.5.36/ 1424
3033_00A4	SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)	32	R/W	0000_0005h	8.2.5.37/ 1425
3033_00A8	SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)	32	R/W	0000_0005h	8.2.5.38/ 1426
3033_00AC	SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)	32	R/W	0000_0005h	8.2.5.39/ 1427
3033_00B0	SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)	32	R/W	0000_0005h	8.2.5.40/ 1428
3033_00B4	SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)	32	R/W	0000_0005h	8.2.5.41/ 1429
3033_00B8	SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4)	32	R/W	0000_0005h	8.2.5.42/ 1430
3033_00BC	SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5)	32	R/W	0000_0005h	8.2.5.43/ 1431
3033_00C0	SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6)	32	R/W	0000_0005h	8.2.5.44/ 1432
3033_00C4	SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7)	32	R/W	0000_0005h	8.2.5.45/ 1433
3033_00C8	SW_MUX_CTL_PAD_SD1_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B)	32	R/W	0000_0005h	8.2.5.46/ 1434
3033_00CC	SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE)	32	R/W	0000_0005h	8.2.5.47/ 1435

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_00D0	SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B)	32	R/W	0000_0005h	8.2.5.48/ 1436
3033_00D4	SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)	32	R/W	0000_0005h	8.2.5.49/ 1437
3033_00D8	SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)	32	R/W	0000_0005h	8.2.5.50/ 1438
3033_00DC	SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)	32	R/W	0000_0005h	8.2.5.51/ 1439
3033_00E0	SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)	32	R/W	0000_0005h	8.2.5.52/ 1440
3033_00E4	SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)	32	R/W	0000_0005h	8.2.5.53/ 1441
3033_00E8	SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)	32	R/W	0000_0005h	8.2.5.54/ 1442
3033_00EC	SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B)	32	R/W	0000_0005h	8.2.5.55/ 1443
3033_00F0	SW_MUX_CTL_PAD_SD2_WP SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_WP)	32	R/W	0000_0005h	8.2.5.56/ 1444
3033_00F4	SW_MUX_CTL_PAD_NAND_ALE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)	32	R/W	0000_0005h	8.2.5.57/ 1445
3033_00F8	SW_MUX_CTL_PAD_NAND_CE0_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B)	32	R/W	0000_0005h	8.2.5.58/ 1446
3033_00FC	SW_MUX_CTL_PAD_NAND_CE1_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B)	32	R/W	0000_0005h	8.2.5.59/ 1447
3033_0100	SW_MUX_CTL_PAD_NAND_CE2_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B)	32	R/W	0000_0005h	8.2.5.60/ 1448
3033_0104	SW_MUX_CTL_PAD_NAND_CE3_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B)	32	R/W	0000_0005h	8.2.5.61/ 1449
3033_0108	SW_MUX_CTL_PAD_NAND_CLE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)	32	R/W	0000_0005h	8.2.5.62/ 1450
3033_010C	SW_MUX_CTL_PAD_NAND_DATA00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)	32	R/W	0000_0005h	8.2.5.63/ 1451
3033_0110	SW_MUX_CTL_PAD_NAND_DATA01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)	32	R/W	0000_0005h	8.2.5.64/ 1452
3033_0114	SW_MUX_CTL_PAD_NAND_DATA02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)	32	R/W	0000_0005h	8.2.5.65/ 1453
3033_0118	SW_MUX_CTL_PAD_NAND_DATA03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)	32	R/W	0000_0005h	8.2.5.66/ 1454
3033_011C	SW_MUX_CTL_PAD_NAND_DATA04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)	32	R/W	0000_0005h	8.2.5.67/ 1455
3033_0120	SW_MUX_CTL_PAD_NAND_DATA05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)	32	R/W	0000_0005h	8.2.5.68/ 1456
3033_0124	SW_MUX_CTL_PAD_NAND_DATA06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)	32	R/W	0000_0005h	8.2.5.69/ 1457

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0128	SW_MUX_CTL_PAD_NAND_DATA07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)	32	R/W	0000_0005h	8.2.5.70/ 1458
3033_012C	SW_MUX_CTL_PAD_NAND_DQS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DQS)	32	R/W	0000_0005h	8.2.5.71/ 1459
3033_0130	SW_MUX_CTL_PAD_NAND_RE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B)	32	R/W	0000_0005h	8.2.5.72/ 1460
3033_0134	SW_MUX_CTL_PAD_NAND_READY_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)	32	R/W	0000_0005h	8.2.5.73/ 1461
3033_0138	SW_MUX_CTL_PAD_NAND_WE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B)	32	R/W	0000_0005h	8.2.5.74/ 1462
3033_013C	SW_MUX_CTL_PAD_NAND_WP_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)	32	R/W	0000_0005h	8.2.5.75/ 1463
3033_0140	SW_MUX_CTL_PAD_SAI5_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS)	32	R/W	0000_0005h	8.2.5.76/ 1464
3033_0144	SW_MUX_CTL_PAD_SAI5_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC)	32	R/W	0000_0005h	8.2.5.77/ 1465
3033_0148	SW_MUX_CTL_PAD_SAI5_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0)	32	R/W	0000_0005h	8.2.5.78/ 1466
3033_014C	SW_MUX_CTL_PAD_SAI5_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1)	32	R/W	0000_0005h	8.2.5.79/ 1467
3033_0150	SW_MUX_CTL_PAD_SAI5_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2)	32	R/W	0000_0005h	8.2.5.80/ 1468
3033_0154	SW_MUX_CTL_PAD_SAI5_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3)	32	R/W	0000_0005h	8.2.5.81/ 1469
3033_0158	SW_MUX_CTL_PAD_SAI5_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK)	32	R/W	0000_0005h	8.2.5.82/ 1470
3033_015C	SW_MUX_CTL_PAD_SAI1_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS)	32	R/W	0000_0005h	8.2.5.83/ 1471
3033_0160	SW_MUX_CTL_PAD_SAI1_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC)	32	R/W	0000_0005h	8.2.5.84/ 1473
3033_0164	SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0)	32	R/W	0000_0005h	8.2.5.85/ 1474
3033_0168	SW_MUX_CTL_PAD_SAI1_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1)	32	R/W	0000_0005h	8.2.5.86/ 1475
3033_016C	SW_MUX_CTL_PAD_SAI1_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2)	32	R/W	0000_0005h	8.2.5.87/ 1477
3033_0170	SW_MUX_CTL_PAD_SAI1_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3)	32	R/W	0000_0005h	8.2.5.88/ 1478
3033_0174	SW_MUX_CTL_PAD_SAI1_RXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4)	32	R/W	0000_0005h	8.2.5.89/ 1479
3033_0178	SW_MUX_CTL_PAD_SAI1_RXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5)	32	R/W	0000_0005h	8.2.5.90/ 1481
3033_017C	SW_MUX_CTL_PAD_SAI1_RXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6)	32	R/W	0000_0005h	8.2.5.91/ 1482

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0180	SW_MUX_CTL_PAD_SAI1_RXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7)	32	R/W	0000_0005h	8.2.5.92/ 1483
3033_0184	SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS)	32	R/W	0000_0005h	8.2.5.93/ 1485
3033_0188	SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC)	32	R/W	0000_0005h	8.2.5.94/ 1486
3033_018C	SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0)	32	R/W	0000_0005h	8.2.5.95/ 1487
3033_0190	SW_MUX_CTL_PAD_SAI1_TXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1)	32	R/W	0000_0005h	8.2.5.96/ 1488
3033_0194	SW_MUX_CTL_PAD_SAI1_TXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2)	32	R/W	0000_0005h	8.2.5.97/ 1490
3033_0198	SW_MUX_CTL_PAD_SAI1_TXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3)	32	R/W	0000_0005h	8.2.5.98/ 1491
3033_019C	SW_MUX_CTL_PAD_SAI1_TXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4)	32	R/W	0000_0005h	8.2.5.99/ 1492
3033_01A0	SW_MUX_CTL_PAD_SAI1_TXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5)	32	R/W	0000_0005h	8.2.5.100/ 1494
3033_01A4	SW_MUX_CTL_PAD_SAI1_TXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6)	32	R/W	0000_0005h	8.2.5.101/ 1495
3033_01A8	SW_MUX_CTL_PAD_SAI1_TXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7)	32	R/W	0000_0005h	8.2.5.102/ 1496
3033_01AC	SW_MUX_CTL_PAD_SAI1_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK)	32	R/W	0000_0005h	8.2.5.103/ 1498
3033_01B0	SW_MUX_CTL_PAD_SAI2_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS)	32	R/W	0000_0005h	8.2.5.104/ 1499
3033_01B4	SW_MUX_CTL_PAD_SAI2_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC)	32	R/W	0000_0005h	8.2.5.105/ 1500
3033_01B8	SW_MUX_CTL_PAD_SAI2_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0)	32	R/W	0000_0005h	8.2.5.106/ 1501
3033_01BC	SW_MUX_CTL_PAD_SAI2_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS)	32	R/W	0000_0005h	8.2.5.107/ 1502
3033_01C0	SW_MUX_CTL_PAD_SAI2_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC)	32	R/W	0000_0005h	8.2.5.108/ 1503
3033_01C4	SW_MUX_CTL_PAD_SAI2_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0)	32	R/W	0000_0005h	8.2.5.109/ 1504
3033_01C8	SW_MUX_CTL_PAD_SAI2_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK)	32	R/W	0000_0005h	8.2.5.110/ 1505
3033_01CC	SW_MUX_CTL_PAD_SAI3_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS)	32	R/W	0000_0005h	8.2.5.111/ 1506
3033_01D0	SW_MUX_CTL_PAD_SAI3_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC)	32	R/W	0000_0005h	8.2.5.112/ 1507
3033_01D4	SW_MUX_CTL_PAD_SAI3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD)	32	R/W	0000_0005h	8.2.5.113/ 1508

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_01D8	SW_MUX_CTL_PAD_SAI3_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS)	32	R/W	0000_0005h	8.2.5.114/ 1509
3033_01DC	SW_MUX_CTL_PAD_SAI3_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC)	32	R/W	0000_0005h	8.2.5.115/ 1510
3033_01E0	SW_MUX_CTL_PAD_SAI3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD)	32	R/W	0000_0005h	8.2.5.116/ 1511
3033_01E4	SW_MUX_CTL_PAD_SAI3_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK)	32	R/W	0000_0005h	8.2.5.117/ 1512
3033_01E8	SW_MUX_CTL_PAD_SPDIF_TX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX)	32	R/W	0000_0005h	8.2.5.118/ 1513
3033_01EC	SW_MUX_CTL_PAD_SPDIF_RX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX)	32	R/W	0000_0005h	8.2.5.119/ 1514
3033_01F0	SW_MUX_CTL_PAD_SPDIF_EXT_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK)	32	R/W	0000_0005h	8.2.5.120/ 1515
3033_01F4	SW_MUX_CTL_PAD_ECSPI1_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK)	32	R/W	0000_0005h	8.2.5.121/ 1516
3033_01F8	SW_MUX_CTL_PAD_ECSPI1_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI)	32	R/W	0000_0005h	8.2.5.122/ 1517
3033_01FC	SW_MUX_CTL_PAD_ECSPI1_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO)	32	R/W	0000_0005h	8.2.5.123/ 1518
3033_0200	SW_MUX_CTL_PAD_ECSPI1_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0)	32	R/W	0000_0005h	8.2.5.124/ 1519
3033_0204	SW_MUX_CTL_PAD_ECSPI2_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK)	32	R/W	0000_0005h	8.2.5.125/ 1520
3033_0208	SW_MUX_CTL_PAD_ECSPI2_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI)	32	R/W	0000_0005h	8.2.5.126/ 1521
3033_020C	SW_MUX_CTL_PAD_ECSPI2_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO)	32	R/W	0000_0005h	8.2.5.127/ 1522
3033_0210	SW_MUX_CTL_PAD_ECSPI2_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0)	32	R/W	0000_0005h	8.2.5.128/ 1523
3033_0214	SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL)	32	R/W	0000_0005h	8.2.5.129/ 1524
3033_0218	SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA)	32	R/W	0000_0005h	8.2.5.130/ 1525
3033_021C	SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL)	32	R/W	0000_0005h	8.2.5.131/ 1526
3033_0220	SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA)	32	R/W	0000_0005h	8.2.5.132/ 1527
3033_0224	SW_MUX_CTL_PAD_I2C3_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL)	32	R/W	0000_0005h	8.2.5.133/ 1528
3033_0228	SW_MUX_CTL_PAD_I2C3_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA)	32	R/W	0000_0005h	8.2.5.134/ 1529
3033_022C	SW_MUX_CTL_PAD_I2C4_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL)	32	R/W	0000_0005h	8.2.5.135/ 1530

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0230	SW_MUX_CTL_PAD_I2C4_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA)	32	R/W	0000_0005h	8.2.5.136/ 1531
3033_0234	SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_RXD)	32	R/W	0000_0005h	8.2.5.137/ 1532
3033_0238	SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_TXD)	32	R/W	0000_0005h	8.2.5.138/ 1533
3033_023C	SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_RXD)	32	R/W	0000_0005h	8.2.5.139/ 1534
3033_0240	SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_TXD)	32	R/W	0000_0005h	8.2.5.140/ 1535
3033_0244	SW_MUX_CTL_PAD_UART3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_RXD)	32	R/W	0000_0005h	8.2.5.141/ 1536
3033_0248	SW_MUX_CTL_PAD_UART3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_TXD)	32	R/W	0000_0005h	8.2.5.142/ 1537
3033_024C	SW_MUX_CTL_PAD_UART4_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_RXD)	32	R/W	0000_0005h	8.2.5.143/ 1538
3033_0250	SW_MUX_CTL_PAD_UART4_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_TXD)	32	R/W	0000_0005h	8.2.5.144/ 1539
3033_0254	SW_PAD_CTL_PAD_TEST_MODE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_TEST_MODE)	32	R/W	0000_0001h	8.2.5.145/ 1540
3033_0258	SW_PAD_CTL_PAD_BOOT_MODE0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)	32	R/W	0000_0081h	8.2.5.146/ 1541
3033_025C	SW_PAD_CTL_PAD_BOOT_MODE1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)	32	R/W	0000_0081h	8.2.5.147/ 1543
3033_0260	SW_PAD_CTL_PAD_JTAG_MOD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)	32	R/W	0000_1801h	8.2.5.148/ 1544
3033_0264	SW_PAD_CTL_PAD_JTAG_TRST_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRST_B)	32	R/W	0000_0041h	8.2.5.149/ 1546
3033_0268	SW_PAD_CTL_PAD_JTAG_TDI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)	32	R/W	0000_0041h	8.2.5.150/ 1547
3033_026C	SW_PAD_CTL_PAD_JTAG_TMS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)	32	R/W	0000_0041h	8.2.5.151/ 1548
3033_0270	SW_PAD_CTL_PAD_JTAG_TCK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)	32	R/W	0000_0041h	8.2.5.152/ 1549
3033_0274	SW_PAD_CTL_PAD_JTAG_TDO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)	32	R/W	0000_0041h	8.2.5.153/ 1550
3033_0278	SW_PAD_CTL_PAD_RTC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_RTC)	32	R/W	0000_0001h	8.2.5.154/ 1552
3033_027C	SW_PAD_CTL_PAD_PMIC_STBY_REQ SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ)	32	R/W	0000_000Ch	8.2.5.155/ 1553
3033_0280	SW_PAD_CTL_PAD_PMIC_ON_REQ SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ)	32	R/W	0000_186Ch	8.2.5.156/ 1555
3033_0284	SW_PAD_CTL_PAD_ONOFF SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ONOFF)	32	R/W	0000_00CCh	8.2.5.157/ 1557

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0288	SW_PAD_CTL_PAD_POR_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_POR_B)	32	R/W	0000_00CCh	8.2.5.158/ 1558
3033_028C	SW_PAD_CTL_PAD_RTC_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_RTC_RESET_B)	32	R/W	0000_00CCh	8.2.5.159/ 1560
3033_0290	SW_PAD_CTL_PAD_GPIO1_IO00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00)	32	R/W	0000_0014h	8.2.5.160/ 1561
3033_0294	SW_PAD_CTL_PAD_GPIO1_IO01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01)	32	R/W	0000_0016h	8.2.5.161/ 1563
3033_0298	SW_PAD_CTL_PAD_GPIO1_IO02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02)	32	R/W	0000_0056h	8.2.5.162/ 1564
3033_029C	SW_PAD_CTL_PAD_GPIO1_IO03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03)	32	R/W	0000_0016h	8.2.5.163/ 1566
3033_02A0	SW_PAD_CTL_PAD_GPIO1_IO04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04)	32	R/W	0000_0016h	8.2.5.164/ 1567
3033_02A4	SW_PAD_CTL_PAD_GPIO1_IO05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05)	32	R/W	0000_0056h	8.2.5.165/ 1569
3033_02A8	SW_PAD_CTL_PAD_GPIO1_IO06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06)	32	R/W	0000_0016h	8.2.5.166/ 1570
3033_02AC	SW_PAD_CTL_PAD_GPIO1_IO07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07)	32	R/W	0000_1816h	8.2.5.167/ 1572
3033_02B0	SW_PAD_CTL_PAD_GPIO1_IO08 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08)	32	R/W	0000_0016h	8.2.5.168/ 1574
3033_02B4	SW_PAD_CTL_PAD_GPIO1_IO09 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09)	32	R/W	0000_0016h	8.2.5.169/ 1575
3033_02B8	SW_PAD_CTL_PAD_GPIO1_IO10 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10)	32	R/W	0000_0016h	8.2.5.170/ 1577
3033_02BC	SW_PAD_CTL_PAD_GPIO1_IO11 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11)	32	R/W	0000_0016h	8.2.5.171/ 1578
3033_02C0	SW_PAD_CTL_PAD_GPIO1_IO12 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12)	32	R/W	0000_0016h	8.2.5.172/ 1580
3033_02C4	SW_PAD_CTL_PAD_GPIO1_IO13 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13)	32	R/W	0000_0016h	8.2.5.173/ 1581
3033_02C8	SW_PAD_CTL_PAD_GPIO1_IO14 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14)	32	R/W	0000_0016h	8.2.5.174/ 1583
3033_02CC	SW_PAD_CTL_PAD_GPIO1_IO15 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15)	32	R/W	0000_0016h	8.2.5.175/ 1584
3033_02D0	SW_PAD_CTL_PAD_ENET_MDC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)	32	R/W	0000_0016h	8.2.5.176/ 1586
3033_02D4	SW_PAD_CTL_PAD_ENET_MDIO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)	32	R/W	0000_0016h	8.2.5.177/ 1587
3033_02D8	SW_PAD_CTL_PAD_ENET_TD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD3)	32	R/W	0000_0016h	8.2.5.178/ 1589
3033_02DC	SW_PAD_CTL_PAD_ENET_TD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD2)	32	R/W	0000_0016h	8.2.5.179/ 1590

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_02E0	SW_PAD_CTL_PAD_ENET_TD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD1)	32	R/W	0000_0016h	8.2.5.180/ 1592
3033_02E4	SW_PAD_CTL_PAD_ENET_TD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD0)	32	R/W	0000_0016h	8.2.5.181/ 1593
3033_02E8	SW_PAD_CTL_PAD_ENET_TX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL)	32	R/W	0000_1816h	8.2.5.182/ 1595
3033_02EC	SW_PAD_CTL_PAD_ENET_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TXC)	32	R/W	0000_0016h	8.2.5.183/ 1597
3033_02F0	SW_PAD_CTL_PAD_ENET_RX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL)	32	R/W	0000_0016h	8.2.5.184/ 1598
3033_02F4	SW_PAD_CTL_PAD_ENET_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RXC)	32	R/W	0000_0016h	8.2.5.185/ 1600
3033_02F8	SW_PAD_CTL_PAD_ENET_RD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD0)	32	R/W	0000_0016h	8.2.5.186/ 1601
3033_02FC	SW_PAD_CTL_PAD_ENET_RD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD1)	32	R/W	0000_0016h	8.2.5.187/ 1603
3033_0300	SW_PAD_CTL_PAD_ENET_RD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD2)	32	R/W	0000_0016h	8.2.5.188/ 1604
3033_0304	SW_PAD_CTL_PAD_ENET_RD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD3)	32	R/W	0000_0016h	8.2.5.189/ 1606
3033_0308	SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)	32	R/W	0000_0016h	8.2.5.190/ 1607
3033_030C	SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)	32	R/W	0000_0016h	8.2.5.191/ 1609
3033_0310	SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)	32	R/W	0000_0016h	8.2.5.192/ 1610
3033_0314	SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)	32	R/W	0000_0016h	8.2.5.193/ 1612
3033_0318	SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)	32	R/W	0000_0016h	8.2.5.194/ 1613
3033_031C	SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)	32	R/W	0000_1816h	8.2.5.195/ 1615
3033_0320	SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4)	32	R/W	0000_0016h	8.2.5.196/ 1617
3033_0324	SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5)	32	R/W	0000_0016h	8.2.5.197/ 1618
3033_0328	SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6)	32	R/W	0000_0016h	8.2.5.198/ 1620
3033_032C	SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7)	32	R/W	0000_0016h	8.2.5.199/ 1621
3033_0330	SW_PAD_CTL_PAD_SD1_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B)	32	R/W	0000_0016h	8.2.5.200/ 1623
3033_0334	SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE)	32	R/W	0000_0016h	8.2.5.201/ 1624

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0338	SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B)	32	R/W	0000_0016h	8.2.5.202/ 1626
3033_033C	SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)	32	R/W	0000_0016h	8.2.5.203/ 1627
3033_0340	SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)	32	R/W	0000_0016h	8.2.5.204/ 1629
3033_0344	SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)	32	R/W	0000_1816h	8.2.5.205/ 1630
3033_0348	SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)	32	R/W	0000_0016h	8.2.5.206/ 1632
3033_034C	SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)	32	R/W	0000_0016h	8.2.5.207/ 1634
3033_0350	SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)	32	R/W	0000_0016h	8.2.5.208/ 1635
3033_0354	SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B)	32	R/W	0000_0016h	8.2.5.209/ 1637
3033_0358	SW_PAD_CTL_PAD_SD2_WP SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_WP)	32	R/W	0000_0016h	8.2.5.210/ 1638
3033_035C	SW_PAD_CTL_PAD_NAND_ALE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)	32	R/W	0000_0016h	8.2.5.211/ 1640
3033_0360	SW_PAD_CTL_PAD_NAND_CE0_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B)	32	R/W	0000_0016h	8.2.5.212/ 1641
3033_0364	SW_PAD_CTL_PAD_NAND_CE1_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B)	32	R/W	0000_0016h	8.2.5.213/ 1643
3033_0368	SW_PAD_CTL_PAD_NAND_CE2_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B)	32	R/W	0000_0016h	8.2.5.214/ 1644
3033_036C	SW_PAD_CTL_PAD_NAND_CE3_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B)	32	R/W	0000_0016h	8.2.5.215/ 1646
3033_0370	SW_PAD_CTL_PAD_NAND_CLE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)	32	R/W	0000_0016h	8.2.5.216/ 1647
3033_0374	SW_PAD_CTL_PAD_NAND_DATA00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)	32	R/W	0000_0016h	8.2.5.217/ 1649
3033_0378	SW_PAD_CTL_PAD_NAND_DATA01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)	32	R/W	0000_0016h	8.2.5.218/ 1650
3033_037C	SW_PAD_CTL_PAD_NAND_DATA02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)	32	R/W	0000_0016h	8.2.5.219/ 1652
3033_0380	SW_PAD_CTL_PAD_NAND_DATA03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)	32	R/W	0000_1816h	8.2.5.220/ 1653
3033_0384	SW_PAD_CTL_PAD_NAND_DATA04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)	32	R/W	0000_0016h	8.2.5.221/ 1655
3033_0388	SW_PAD_CTL_PAD_NAND_DATA05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)	32	R/W	0000_0016h	8.2.5.222/ 1657
3033_038C	SW_PAD_CTL_PAD_NAND_DATA06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)	32	R/W	0000_0016h	8.2.5.223/ 1658

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0390	SW_PAD_CTL_PAD_NAND_DATA07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)	32	R/W	0000_0016h	8.2.5.224/ 1660
3033_0394	SW_PAD_CTL_PAD_NAND_DQS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DQS)	32	R/W	0000_0016h	8.2.5.225/ 1661
3033_0398	SW_PAD_CTL_PAD_NAND_RE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B)	32	R/W	0000_0016h	8.2.5.226/ 1663
3033_039C	SW_PAD_CTL_PAD_NAND_READY_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)	32	R/W	0000_0016h	8.2.5.227/ 1664
3033_03A0	SW_PAD_CTL_PAD_NAND_WE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B)	32	R/W	0000_0016h	8.2.5.228/ 1666
3033_03A4	SW_PAD_CTL_PAD_NAND_WP_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)	32	R/W	0000_0016h	8.2.5.229/ 1667
3033_03A8	SW_PAD_CTL_PAD_SAI5_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS)	32	R/W	0000_0016h	8.2.5.230/ 1669
3033_03AC	SW_PAD_CTL_PAD_SAI5_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC)	32	R/W	0000_0016h	8.2.5.231/ 1670
3033_03B0	SW_PAD_CTL_PAD_SAI5_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0)	32	R/W	0000_0016h	8.2.5.232/ 1672
3033_03B4	SW_PAD_CTL_PAD_SAI5_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1)	32	R/W	0000_1816h	8.2.5.233/ 1673
3033_03B8	SW_PAD_CTL_PAD_SAI5_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2)	32	R/W	0000_0016h	8.2.5.234/ 1675
3033_03BC	SW_PAD_CTL_PAD_SAI5_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3)	32	R/W	0000_0016h	8.2.5.235/ 1677
3033_03C0	SW_PAD_CTL_PAD_SAI5_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK)	32	R/W	0000_0016h	8.2.5.236/ 1678
3033_03C4	SW_PAD_CTL_PAD_SAI1_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS)	32	R/W	0000_0016h	8.2.5.237/ 1680
3033_03C8	SW_PAD_CTL_PAD_SAI1_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC)	32	R/W	0000_0016h	8.2.5.238/ 1681
3033_03CC	SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0)	32	R/W	0000_0016h	8.2.5.239/ 1683
3033_03D0	SW_PAD_CTL_PAD_SAI1_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1)	32	R/W	0000_0016h	8.2.5.240/ 1684
3033_03D4	SW_PAD_CTL_PAD_SAI1_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2)	32	R/W	0000_0016h	8.2.5.241/ 1686
3033_03D8	SW_PAD_CTL_PAD_SAI1_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3)	32	R/W	0000_0016h	8.2.5.242/ 1687
3033_03DC	SW_PAD_CTL_PAD_SAI1_RXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4)	32	R/W	0000_0016h	8.2.5.243/ 1689
3033_03E0	SW_PAD_CTL_PAD_SAI1_RXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5)	32	R/W	0000_0016h	8.2.5.244/ 1690
3033_03E4	SW_PAD_CTL_PAD_SAI1_RXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6)	32	R/W	0000_0016h	8.2.5.245/ 1692

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_03E8	SW_PAD_CTL_PAD_SAI1_RXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7)	32	R/W	0000_0016h	8.2.5.246/ 1693
3033_03EC	SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS)	32	R/W	0000_1816h	8.2.5.247/ 1695
3033_03F0	SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC)	32	R/W	0000_0016h	8.2.5.248/ 1697
3033_03F4	SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0)	32	R/W	0000_0016h	8.2.5.249/ 1698
3033_03F8	SW_PAD_CTL_PAD_SAI1_TXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1)	32	R/W	0000_0016h	8.2.5.250/ 1700
3033_03FC	SW_PAD_CTL_PAD_SAI1_TXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2)	32	R/W	0000_0016h	8.2.5.251/ 1701
3033_0400	SW_PAD_CTL_PAD_SAI1_TXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3)	32	R/W	0000_0016h	8.2.5.252/ 1703
3033_0404	SW_PAD_CTL_PAD_SAI1_TXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4)	32	R/W	0000_0016h	8.2.5.253/ 1704
3033_0408	SW_PAD_CTL_PAD_SAI1_TXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5)	32	R/W	0000_0016h	8.2.5.254/ 1706
3033_040C	SW_PAD_CTL_PAD_SAI1_TXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6)	32	R/W	0000_0016h	8.2.5.255/ 1707
3033_0410	SW_PAD_CTL_PAD_SAI1_TXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7)	32	R/W	0000_0016h	8.2.5.256/ 1709
3033_0414	SW_PAD_CTL_PAD_SAI1_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK)	32	R/W	0000_0016h	8.2.5.257/ 1710
3033_0418	SW_PAD_CTL_PAD_SAI2_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS)	32	R/W	0000_0016h	8.2.5.258/ 1712
3033_041C	SW_PAD_CTL_PAD_SAI2_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC)	32	R/W	0000_0016h	8.2.5.259/ 1713
3033_0420	SW_PAD_CTL_PAD_SAI2_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0)	32	R/W	0000_0016h	8.2.5.260/ 1715
3033_0424	SW_PAD_CTL_PAD_SAI2_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS)	32	R/W	0000_1816h	8.2.5.261/ 1716
3033_0428	SW_PAD_CTL_PAD_SAI2_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC)	32	R/W	0000_0016h	8.2.5.262/ 1718
3033_042C	SW_PAD_CTL_PAD_SAI2_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0)	32	R/W	0000_0016h	8.2.5.263/ 1720
3033_0430	SW_PAD_CTL_PAD_SAI2_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK)	32	R/W	0000_0016h	8.2.5.264/ 1721
3033_0434	SW_PAD_CTL_PAD_SAI3_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS)	32	R/W	0000_0016h	8.2.5.265/ 1723
3033_0438	SW_PAD_CTL_PAD_SAI3_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC)	32	R/W	0000_0016h	8.2.5.266/ 1724
3033_043C	SW_PAD_CTL_PAD_SAI3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD)	32	R/W	0000_0016h	8.2.5.267/ 1726

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0440	SW_PAD_CTL_PAD_SAI3_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS)	32	R/W	0000_0016h	8.2.5.268/ 1727
3033_0444	SW_PAD_CTL_PAD_SAI3_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC)	32	R/W	0000_1816h	8.2.5.269/ 1729
3033_0448	SW_PAD_CTL_PAD_SAI3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD)	32	R/W	0000_0016h	8.2.5.270/ 1731
3033_044C	SW_PAD_CTL_PAD_SAI3_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK)	32	R/W	0000_0016h	8.2.5.271/ 1732
3033_0450	SW_PAD_CTL_PAD_SPDIF_TX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX)	32	R/W	0000_0016h	8.2.5.272/ 1734
3033_0454	SW_PAD_CTL_PAD_SPDIF_RX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX)	32	R/W	0000_0016h	8.2.5.273/ 1735
3033_0458	SW_PAD_CTL_PAD_SPDIF_EXT_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK)	32	R/W	0000_0016h	8.2.5.274/ 1737
3033_045C	SW_PAD_CTL_PAD_ECSPI1_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK)	32	R/W	0000_0016h	8.2.5.275/ 1738
3033_0460	SW_PAD_CTL_PAD_ECSPI1_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI)	32	R/W	0000_0016h	8.2.5.276/ 1740
3033_0464	SW_PAD_CTL_PAD_ECSPI1_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO)	32	R/W	0000_0016h	8.2.5.277/ 1741
3033_0468	SW_PAD_CTL_PAD_ECSPI1_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0)	32	R/W	0000_0016h	8.2.5.278/ 1743
3033_046C	SW_PAD_CTL_PAD_ECSPI2_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK)	32	R/W	0000_1816h	8.2.5.279/ 1744
3033_0470	SW_PAD_CTL_PAD_ECSPI2_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI)	32	R/W	0000_0016h	8.2.5.280/ 1746
3033_0474	SW_PAD_CTL_PAD_ECSPI2_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO)	32	R/W	0000_0016h	8.2.5.281/ 1748
3033_0478	SW_PAD_CTL_PAD_ECSPI2_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0)	32	R/W	0000_0016h	8.2.5.282/ 1749
3033_047C	SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL)	32	R/W	0000_0016h	8.2.5.283/ 1751
3033_0480	SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA)	32	R/W	0000_0016h	8.2.5.284/ 1752
3033_0484	SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL)	32	R/W	0000_0016h	8.2.5.285/ 1754
3033_0488	SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA)	32	R/W	0000_0016h	8.2.5.286/ 1755
3033_048C	SW_PAD_CTL_PAD_I2C3_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL)	32	R/W	0000_1816h	8.2.5.287/ 1757
3033_0490	SW_PAD_CTL_PAD_I2C3_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA)	32	R/W	0000_0016h	8.2.5.288/ 1759
3033_0494	SW_PAD_CTL_PAD_I2C4_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL)	32	R/W	0000_0016h	8.2.5.289/ 1760

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_0498	SW_PAD_CTL_PAD_I2C4_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA)	32	R/W	0000_0016h	8.2.5.290/ 1762
3033_049C	SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_RXD)	32	R/W	0000_0016h	8.2.5.291/ 1763
3033_04A0	SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_TXD)	32	R/W	0000_0016h	8.2.5.292/ 1765
3033_04A4	SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_RXD)	32	R/W	0000_0016h	8.2.5.293/ 1766
3033_04A8	SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_TXD)	32	R/W	0000_0016h	8.2.5.294/ 1768
3033_04AC	SW_PAD_CTL_PAD_UART3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_RXD)	32	R/W	0000_1816h	8.2.5.295/ 1769
3033_04B0	SW_PAD_CTL_PAD_UART3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_TXD)	32	R/W	0000_0016h	8.2.5.296/ 1771
3033_04B4	SW_PAD_CTL_PAD_UART4_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_RXD)	32	R/W	0000_0016h	8.2.5.297/ 1773
3033_04B8	SW_PAD_CTL_PAD_UART4_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_TXD)	32	R/W	0000_0016h	8.2.5.298/ 1774
3033_04BC	CCM_PMIC_READY_SELECT_INPUT DAISY Register (IOMUXC_CCM_PMIC_READY_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.299/ 1776
3033_04C0	ENET1_MDIO_SELECT_INPUT DAISY Register (IOMUXC_ENET1_MDIO_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.300/ 1776
3033_04C4	SAI1_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI1_RX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.301/ 1777
3033_04C8	SAI1_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI1_TX_BCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.302/ 1778
3033_04CC	SAI1_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI1_TX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.303/ 1778
3033_04D0	SAI5_RX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RX_BCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.304/ 1779
3033_04D4	SAI5_RXD0_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD0_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.305/ 1780
3033_04D8	SAI5_RXD1_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD1_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.306/ 1780
3033_04DC	SAI5_RXD2_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD2_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.307/ 1781
3033_04E0	SAI5_RXD3_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD3_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.308/ 1782
3033_04E4	SAI5_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.309/ 1782
3033_04E8	SAI5_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_TX_BCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.310/ 1783
3033_04EC	SAI5_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI5_TX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.311/ 1784

Table continues on the next page...

IOMUXC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3033_04F0	UART1_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART1_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.312/ 1785
3033_04F4	UART1_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART1_RXD_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.313/ 1786
3033_04F8	UART2_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART2_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.314/ 1787
3033_04FC	UART2_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART2_RXD_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.315/ 1788
3033_0500	UART3_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART3_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.316/ 1789
3033_0504	UART3_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART3_RXD_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.317/ 1789
3033_0508	UART4_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART4_RTS_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.318/ 1790
3033_050C	UART4_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART4_RXD_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.319/ 1791
3033_0510	SAI6_RX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RX_BCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.320/ 1792
3033_0514	SAI6_RXD0_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RXD0_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.321/ 1793
3033_0518	SAI6_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.322/ 1794
3033_051C	SAI6_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_TX_BCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.323/ 1795
3033_0520	SAI6_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI6_TX_SYNC_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.324/ 1796
3033_0524	PCIE1_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE1_CLKREQ_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.325/ 1797
3033_0528	PCIE2_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE2_CLKREQ_B_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.326/ 1798
3033_052C	SAI5_MCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.327/ 1798
3033_0530	SAI6_MCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_MCLK_SELECT_INPUT)	32	R/W	0000_0000h	8.2.5.328/ 1799

8.2.5.1 SW_MUX_CTL_PAD_PMIC_STBY_REQ SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_PMIC_STBY_REQ)

SW_MUX_CTL Register

Address: 3033_0000h base + 14h offset = 3033_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									SION	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_PMIC_STBY_REQ field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad PMIC_STBY_REQ is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad PMIC_STBY_REQ
-	This field is reserved. Reserved

8.2.5.2 SW_MUX_CTL_PAD_PMIC_ON_REQ SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_PMIC_ON_REQ)

SW_MUX_CTL Register

Address: 3033_0000h base + 18h offset = 3033_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									SION	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_PMIC_ON_REQ field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad PMIC_ON_REQ is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad PMIC_ON_REQ
-	This field is reserved. Reserved

8.2.5.3 SW_MUX_CTL_PAD_ONOFF SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ONOFF)

SW_MUX_CTL Register

Address: 3033_0000h base + 1Ch offset = 3033_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved									SION	Reserved					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_ONOFF field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ONOFF is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ONOFF
-	This field is reserved. Reserved

8.2.5.4 SW_MUX_CTL_PAD_POR_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_POR_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 20h offset = 3033_0020h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								SION	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SW_MUX_CTL_PAD_POR_B field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad POR_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad POR_B
-	This field is reserved. Reserved

8.2.5.5 SW_MUX_CTL_PAD_RTC_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_RTC_RESET_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 24h offset = 3033_0024h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								SION	Reserved						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

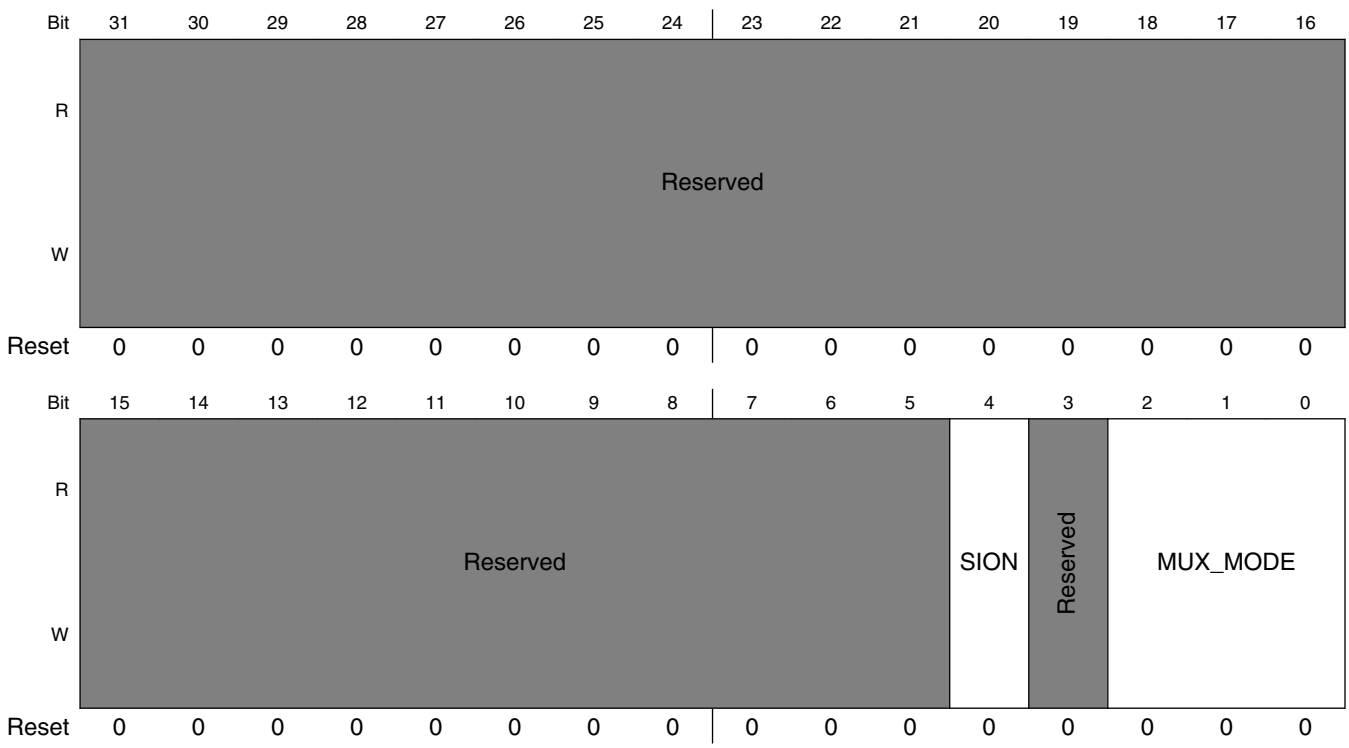
IOMUXC_SW_MUX_CTL_PAD_RTC_RESET_B field descriptions

Field	Description
31–7 -	This field is reserved. Reserved
6 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad RTC_RESET_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad RTC_RESET_B
-	This field is reserved. Reserved

8.2.5.6 SW_MUX_CTL_PAD_GPIO1_IO00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00)

SW_MUX_CTL Register

Address: 3033_0000h base + 28h offset = 3033_0028h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

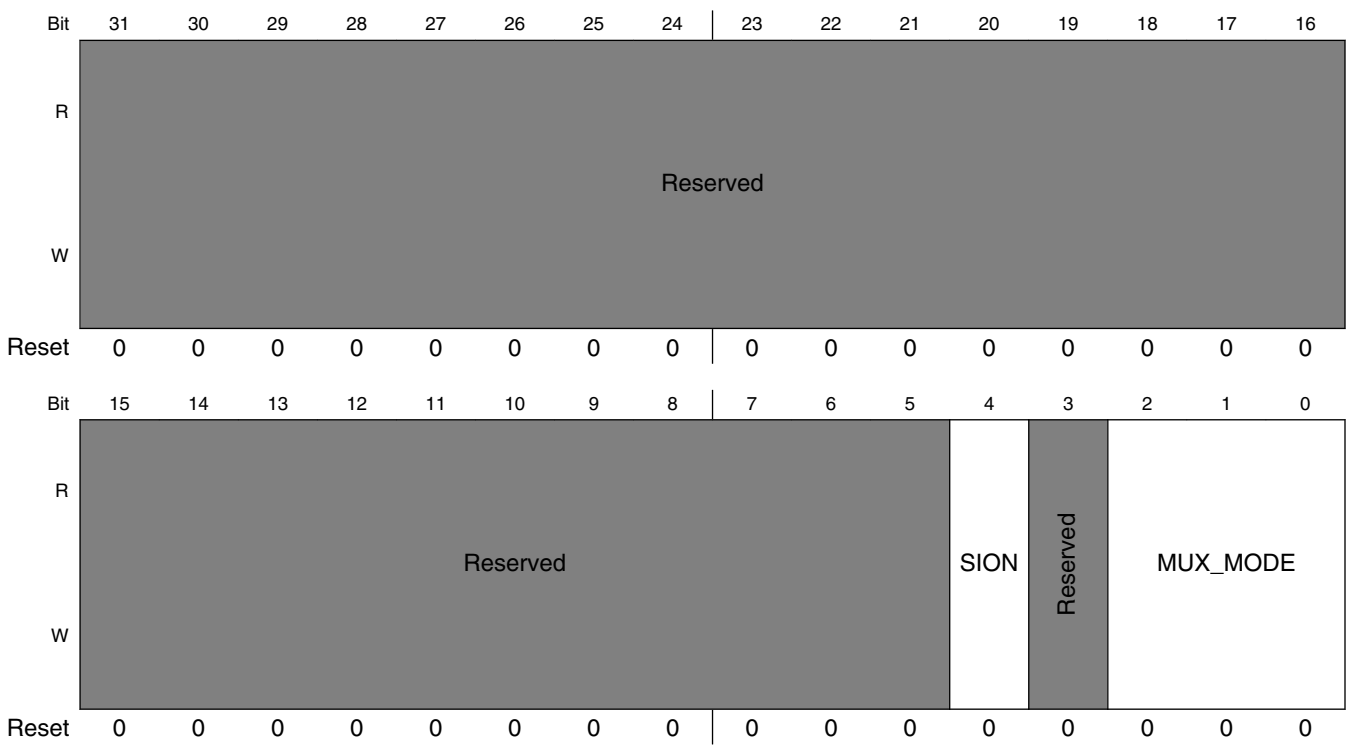
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO00 field descriptions (continued)

Field	Description
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO00 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO00
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO00 000 **ALT0_GPIO1_IO00** — Select mux mode: ALT0 mux port: IO00 of instance: GPIO1 001 **ALT1_CCM_ENET_PHY_REF_CLK_ROOT** — Select mux mode: ALT1 mux port: ENET_PHY_REF_CLK_ROOT of instance: CCM 101 **ALT5_ANAMIX_REF_CLK_32K** — Select mux mode: ALT5 mux port: REF_CLK_32K of instance: ANAMIX 110 **ALT6_CCM_EXT_CLK1** — Select mux mode: ALT6 mux port: EXT_CLK1 of instance: CCM

8.2.5.7 SW_MUX_CTL_PAD_GPIO1_IO01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01)

SW_MUX_CTL Register

Address: 3033_0000h base + 2Ch offset = 3033_002Ch



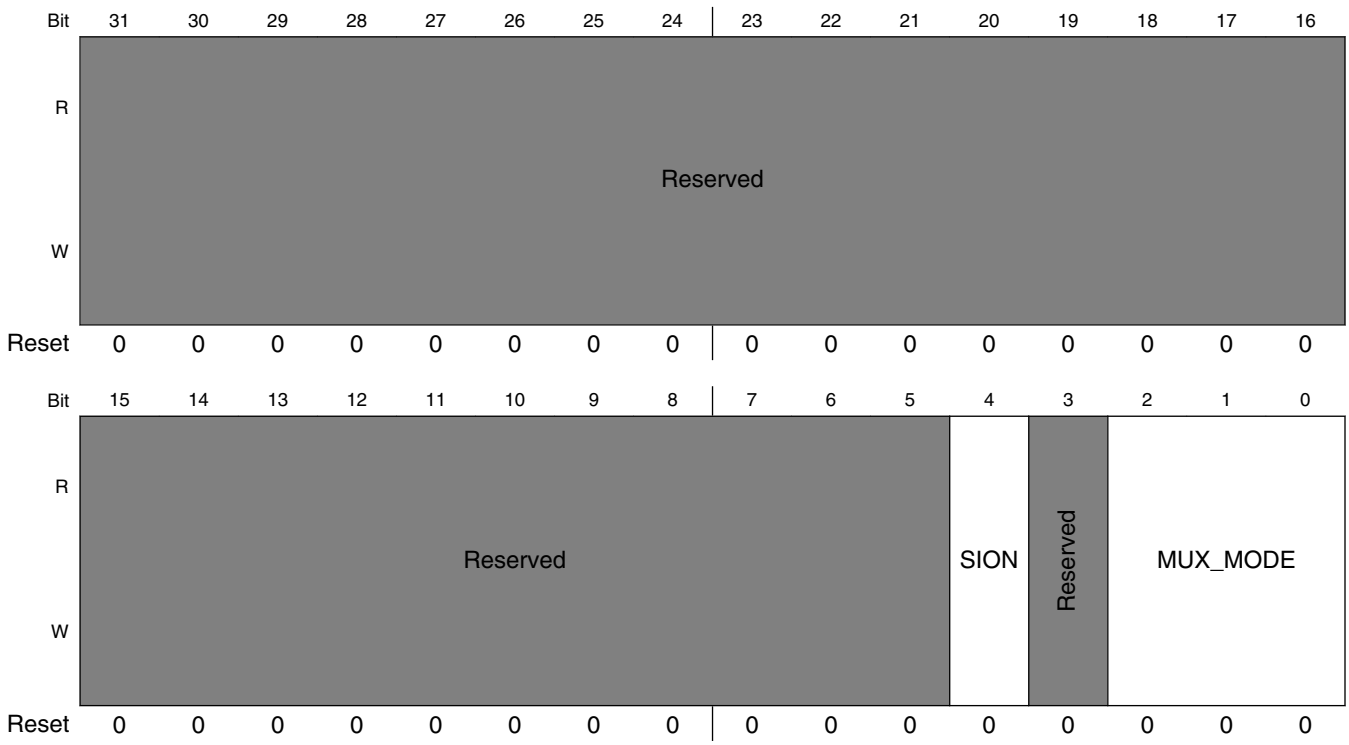
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO01 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO01 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO01
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO01 000 **ALT0_GPIO1_IO01** — Select mux mode: ALT0 mux port: IO01 of instance: GPIO1 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM1 101 **ALT5_ANAMIX_REF_CLK_25M** — Select mux mode: ALT5 mux port: REF_CLK_25M of instance: ANAMIX 110 **ALT6_CCM_EXT_CLK2** — Select mux mode: ALT6 mux port: EXT_CLK2 of instance: CCM

8.2.5.8 SW_MUX_CTL_PAD_GPIO1_IO02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02)

SW_MUX_CTL Register

Address: 3033_0000h base + 30h offset = 3033_0030h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO02 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO02
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO02 000 **ALT0_GPIO1_IO02** — Select mux mode: ALT0 mux port: IO02 of instance: GPIO1 001 **ALT1_WDOG1_WDOG_B** — Select mux mode: ALT1 mux port: WDOG_B of instance: WDOG1

Table continues on the next page...

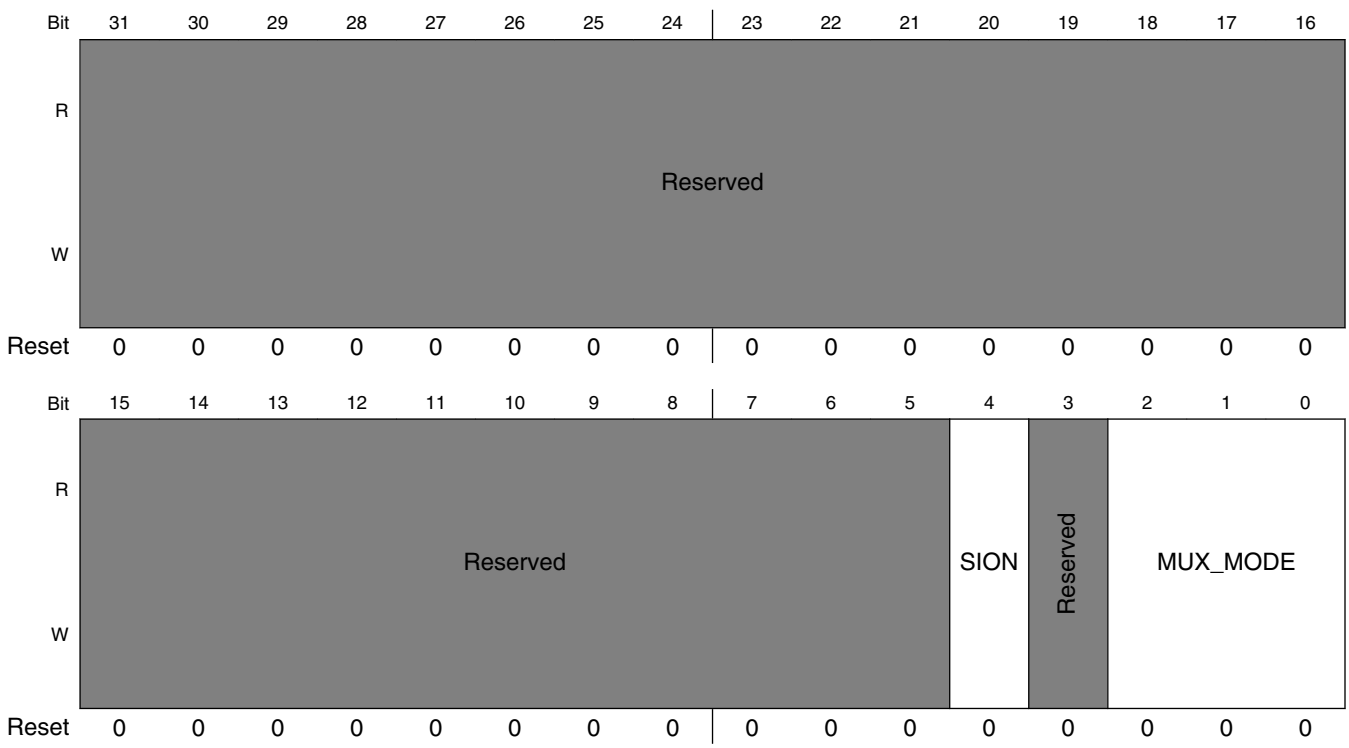
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO02 field descriptions (continued)

Field	Description
	101 **ALT5_WDOG1_WDOG_ANY** — Select mux mode: ALT5 mux port: WDOG_ANY of instance: WDOG1 111 **ALT7_SJC_DE_B** — Select mux mode: ALT7 mux port: DE_B of instance: SJC

8.2.5.9 SW_MUX_CTL_PAD_GPIO1_IO03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03)

SW_MUX_CTL Register

Address: 3033_0000h base + 34h offset = 3033_0034h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO03 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO03

Table continues on the next page...

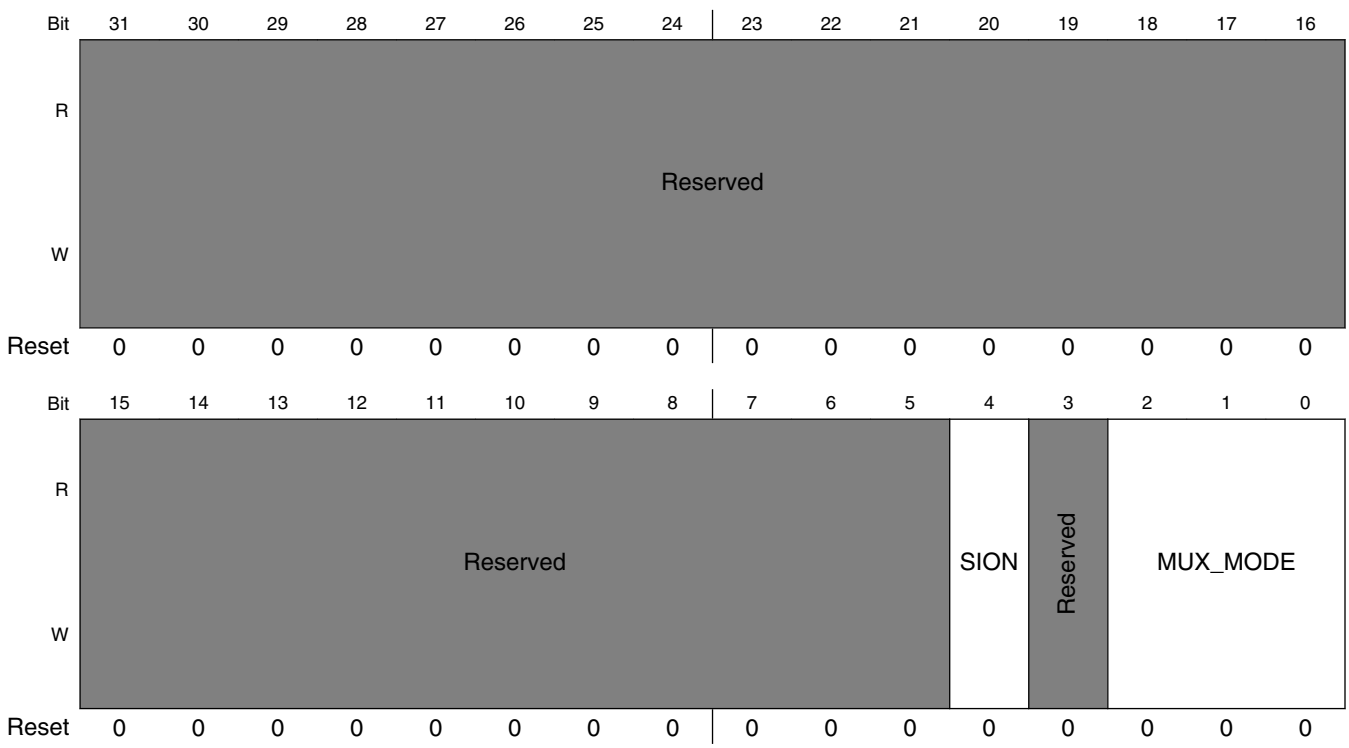
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO03 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO03 000 **ALT0_GPIO1_IO03** — Select mux mode: ALT0 mux port: IO03 of instance: GPIO1 001 **ALT1_USDHC1_VSELECT** — Select mux mode: ALT1 mux port: VSELECT of instance: USDHC1 101 **ALT5_SDMA1_EXT_EVENT0** — Select mux mode: ALT5 mux port: EXT_EVENT0 of instance: SDMA1

8.2.5.10 SW_MUX_CTL_PAD_GPIO1_IO04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04)

SW_MUX_CTL Register

Address: 3033_0000h base + 38h offset = 3033_0038h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO04 field descriptions (continued)

Field	Description
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO04 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO04
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO04 000 **ALT0_GPIO1_IO04** — Select mux mode: ALT0 mux port: IO04 of instance: GPIO1 001 **ALT1_USDHC2_VSELECT** — Select mux mode: ALT1 mux port: VSELECT of instance: USDHC2 101 **ALT5_SDMA1_EXT_EVENT1** — Select mux mode: ALT5 mux port: EXT_EVENT1 of instance: SDMA1

8.2.5.11 SW_MUX_CTL_PAD_GPIO1_IO05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05)

SW_MUX_CTL Register

Address: 3033_0000h base + 3Ch offset = 3033_003Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

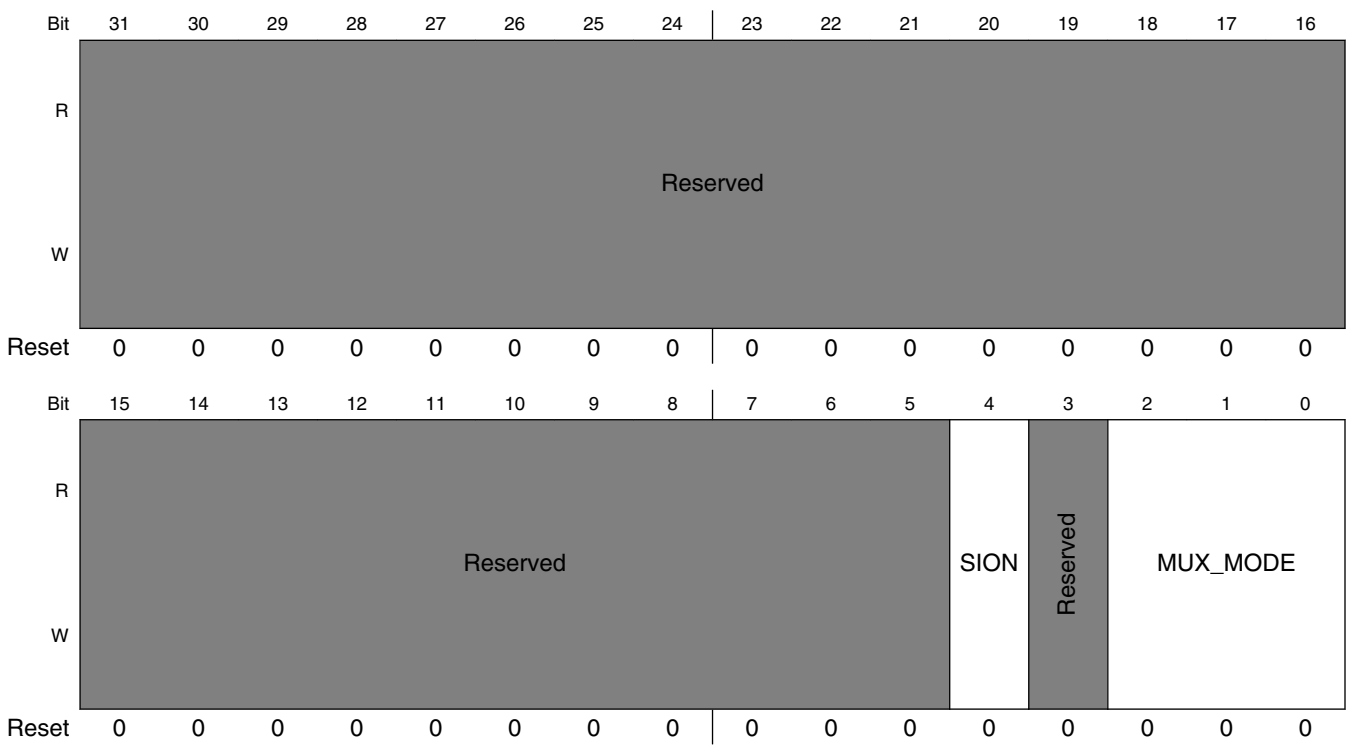
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO05 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO05 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO05
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO05 000 **ALT0_GPIO1_IO05** — Select mux mode: ALT0 mux port: IO05 of instance: GPIO1 001 **ALT1_M4_NMI** — Select mux mode: ALT1 mux port: NMI of instance: M4 101 **ALT5_CCM_PMIC_READY** — Select mux mode: ALT5 mux port: PMIC_READY of instance: CCM

8.2.5.12 SW_MUX_CTL_PAD_GPIO1_IO06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06)

SW_MUX_CTL Register

Address: 3033_0000h base + 40h offset = 3033_0040h



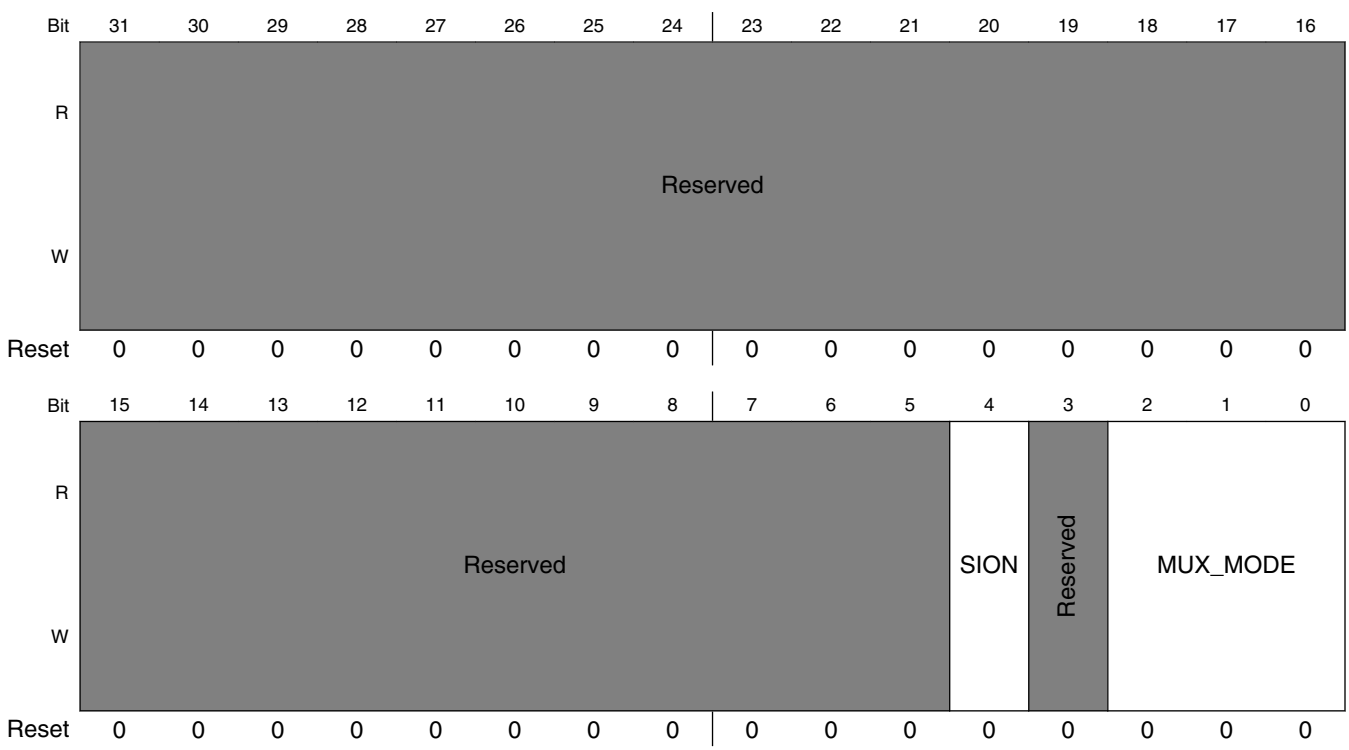
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO06 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO06 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO06
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO06 000 **ALT0_GPIO1_IO06** — Select mux mode: ALT0 mux port: IO06 of instance: GPIO1 001 **ALT1_ENET1_MDC** — Select mux mode: ALT1 mux port: MDC of instance: ENET1 101 **ALT5_USDHC1_CD_B** — Select mux mode: ALT5 mux port: CD_B of instance: USDHC1 110 **ALT6_CCM_EXT_CLK3** — Select mux mode: ALT6 mux port: EXT_CLK3 of instance: CCM

8.2.5.13 SW_MUX_CTL_PAD_GPIO1_IO07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07)

SW_MUX_CTL Register

Address: 3033_0000h base + 44h offset = 3033_0044h



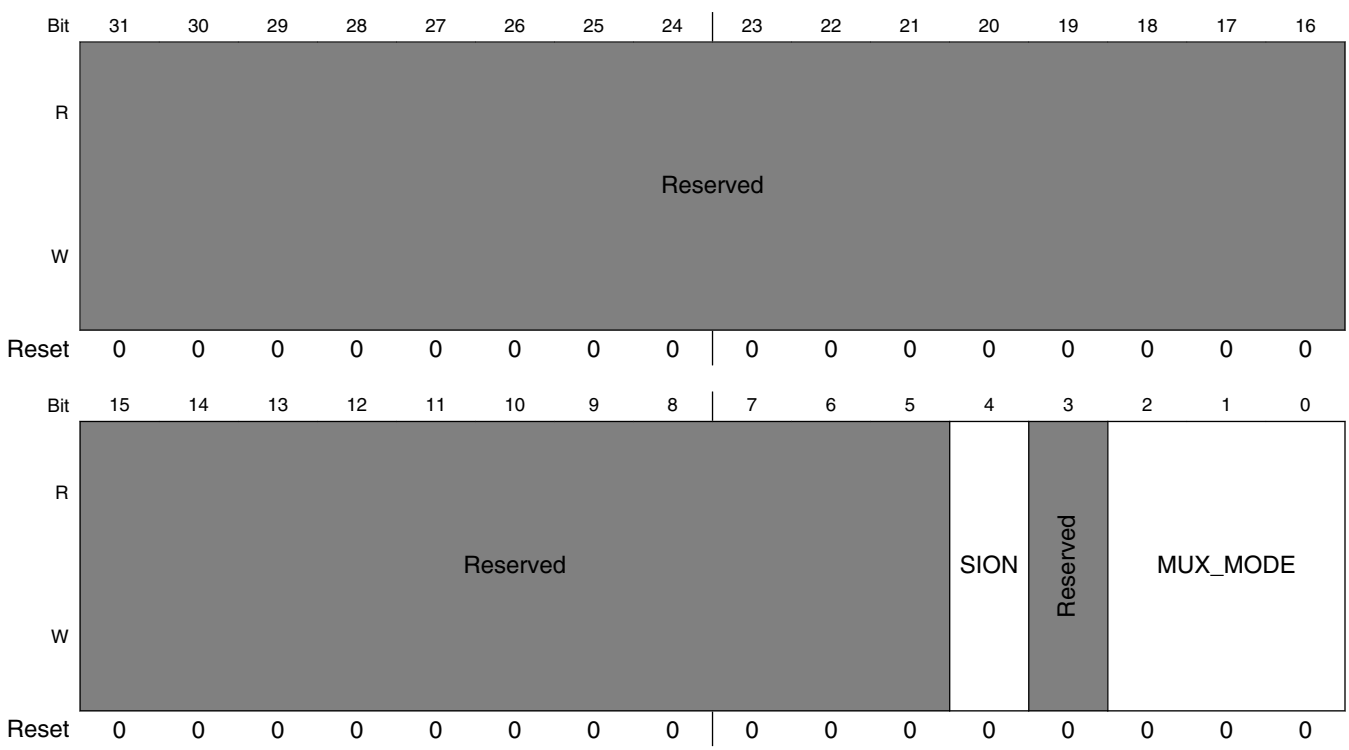
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO07 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO07 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO07
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO07 000 **ALT0_GPIO1_IO07** — Select mux mode: ALT0 mux port: IO07 of instance: GPIO1 001 **ALT1_ENET1_MDIO** — Select mux mode: ALT1 mux port: MDIO of instance: ENET1 101 **ALT5_USDHC1_WP** — Select mux mode: ALT5 mux port: WP of instance: USDHC1 110 **ALT6_CCM_EXT_CLK4** — Select mux mode: ALT6 mux port: EXT_CLK4 of instance: CCM

8.2.5.14 SW_MUX_CTL_PAD_GPIO1_IO08 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08)

SW_MUX_CTL Register

Address: 3033_0000h base + 48h offset = 3033_0048h



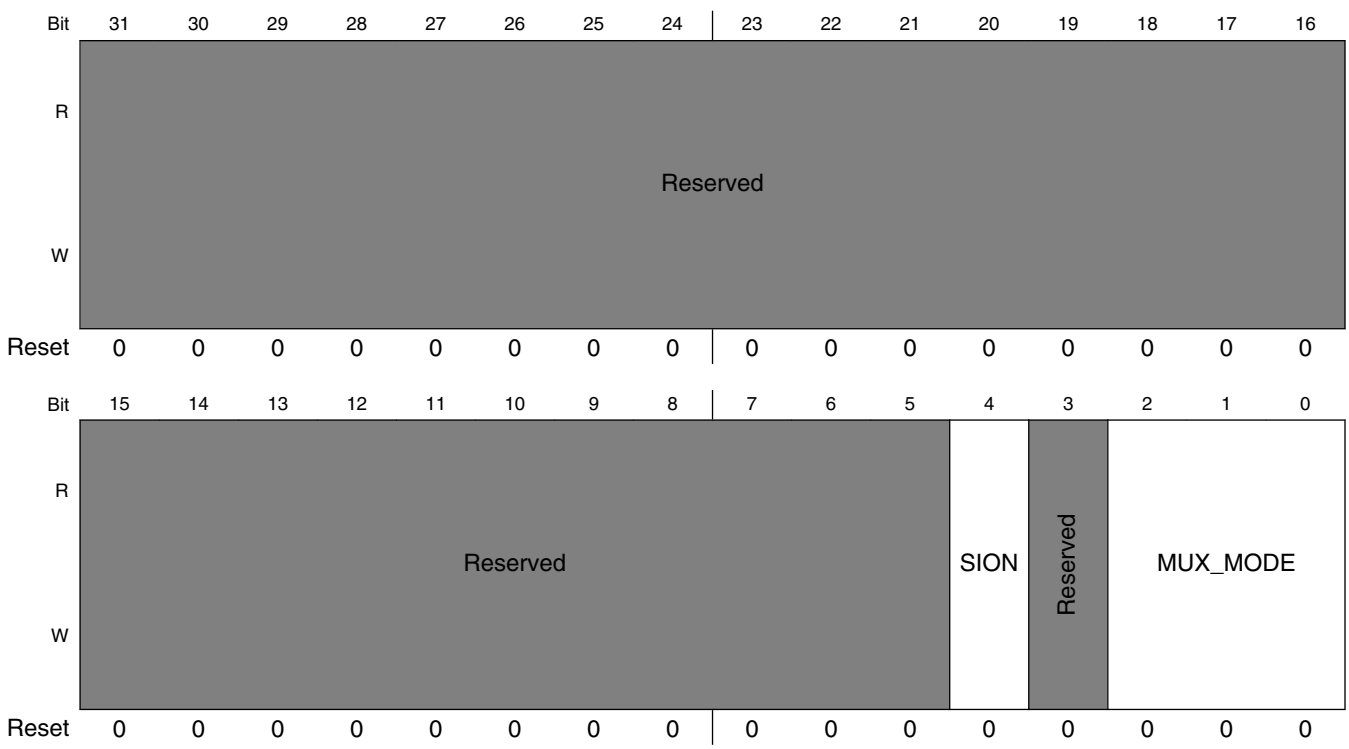
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO08 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO08 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO08
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO08 000 **ALT0_GPIO1_IO08** — Select mux mode: ALT0 mux port: IO08 of instance: GPIO1 001 **ALT1_ENET1_1588_EVENT0_IN** — Select mux mode: ALT1 mux port: 1588_EVENT0_IN of instance: ENET1 101 **ALT5_USDHC2_RESET_B** — Select mux mode: ALT5 mux port: RESET_B of instance: USDHC2

8.2.5.15 SW_MUX_CTL_PAD_GPIO1_IO09 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09)

SW_MUX_CTL Register

Address: 3033_0000h base + 4Ch offset = 3033_004Ch



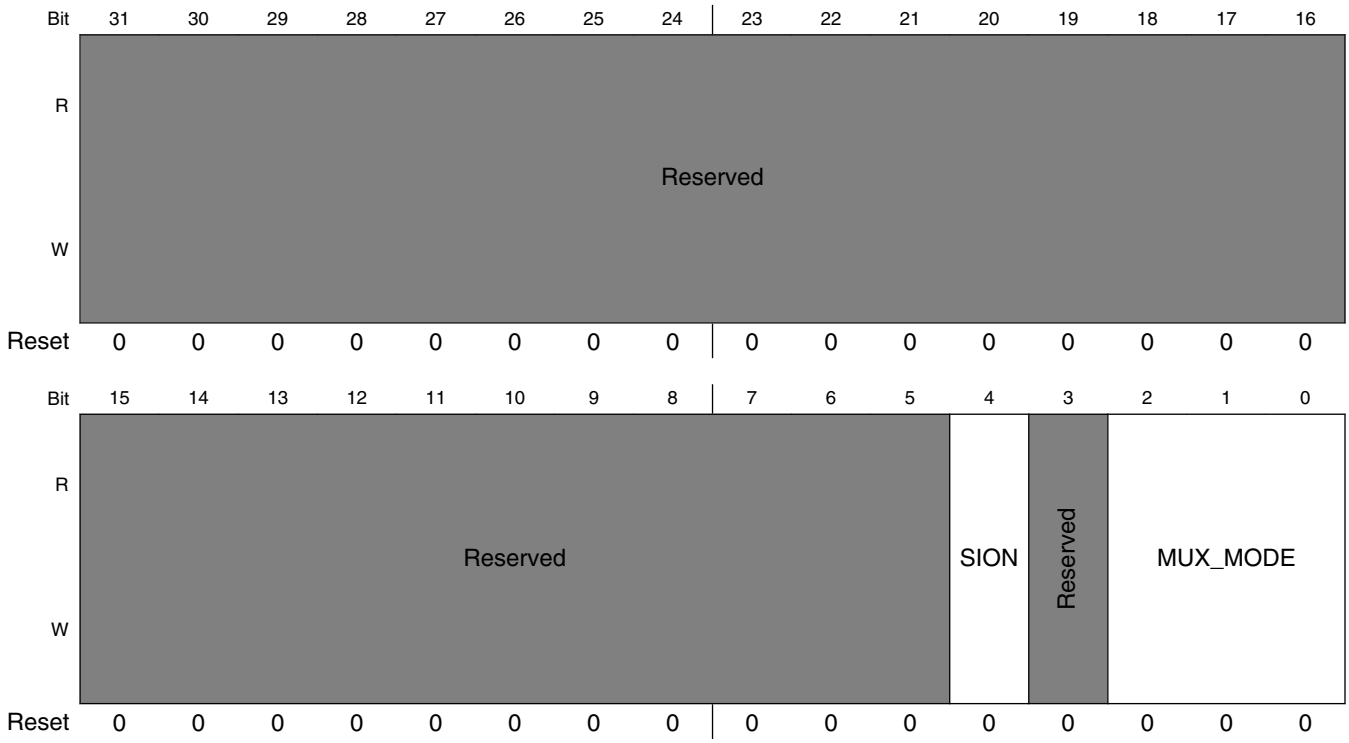
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO09 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO09 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO09
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO09 000 **ALT0_GPIO1_IO09** — Select mux mode: ALT0 mux port: IO09 of instance: GPIO1 001 **ALT1_ENET1_1588_EVENT0_OUT** — Select mux mode: ALT1 mux port: 1588_EVENT0_OUT of instance: ENET1 101 **ALT5_SDMA2_EXT_EVENT0** — Select mux mode: ALT5 mux port: EXT_EVENT0 of instance: SDMA2

8.2.5.16 SW_MUX_CTL_PAD_GPIO1_IO10 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10)

SW_MUX_CTL Register

Address: 3033_0000h base + 50h offset = 3033_0050h



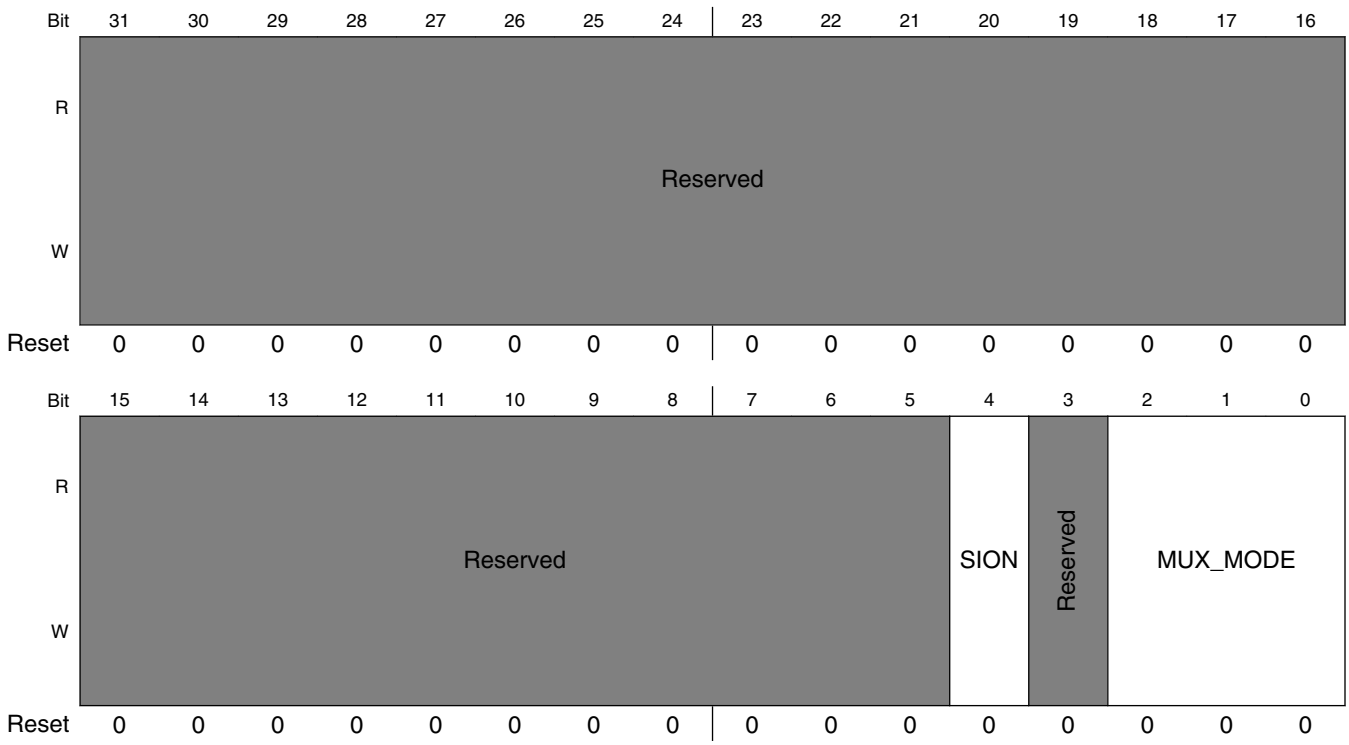
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO10 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO10 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO10
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: GPIO1_IO10 000 **ALT0_GPIO1_IO10** — Select mux mode: ALT0 mux port: IO10 of instance: GPIO1 001 **ALT1_USB1_OTG_ID** — Select mux mode: ALT1 mux port: OTG_ID of instance: USB1

8.2.5.17 SW_MUX_CTL_PAD_GPIO1_IO11 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11)

SW_MUX_CTL Register

Address: 3033_0000h base + 54h offset = 3033_0054h



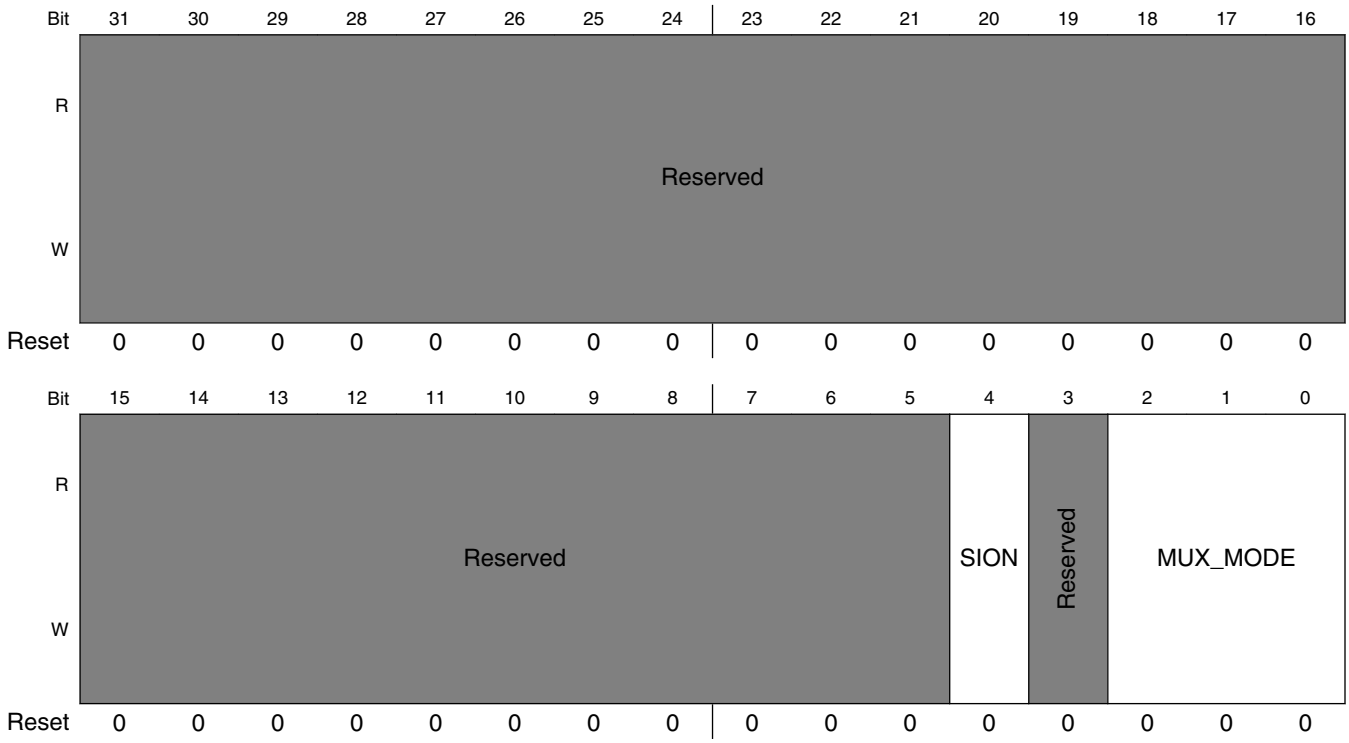
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO11 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO11 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO11
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO11 000 **ALT0_GPIO1_IO11** — Select mux mode: ALT0 mux port: IO11 of instance: GPIO1 001 **ALT1_USB2_OTG_ID** — Select mux mode: ALT1 mux port: OTG_ID of instance: USB2 101 **ALT5_CCM_PMIC_READY** — Select mux mode: ALT5 mux port: PMIC_READY of instance: CCM

8.2.5.18 SW_MUX_CTL_PAD_GPIO1_IO12 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12)

SW_MUX_CTL Register

Address: 3033_0000h base + 58h offset = 3033_0058h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO12 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO12 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO12
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO12 000 **ALT0_GPIO1_IO12** — Select mux mode: ALT0 mux port: IO12 of instance: GPIO1 001 **ALT1_USB1_OTG_PWR** — Select mux mode: ALT1 mux port: OTG_PWR of instance: USB1 101 **ALT5_SDMA2_EXT_EVENT1** — Select mux mode: ALT5 mux port: EXT_EVENT1 of instance: SDMA2

8.2.5.19 SW_MUX_CTL_PAD_GPIO1_IO13 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13)

SW_MUX_CTL Register

Address: 3033_0000h base + 5Ch offset = 3033_005Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

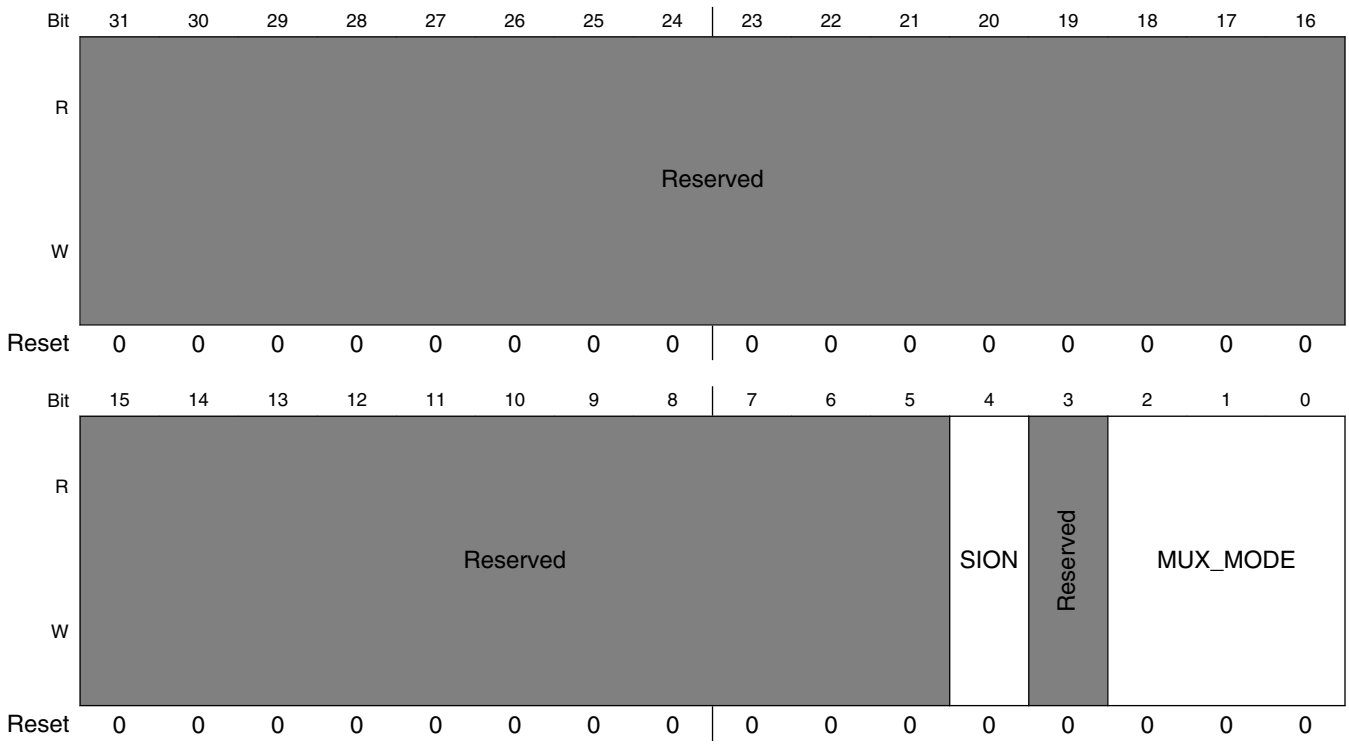
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO13 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO13 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO13
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: GPIO1_IO13 000 **ALT0_GPIO1_IO13** — Select mux mode: ALT0 mux port: IO13 of instance: GPIO1 001 **ALT1_USB1_OTG_OC** — Select mux mode: ALT1 mux port: OTG_OC of instance: USB1 101 **ALT5_PWM2_OUT** — Select mux mode: ALT5 mux port: OUT of instance: PWM2

8.2.5.20 SW_MUX_CTL_PAD_GPIO1_IO14 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14)

SW_MUX_CTL Register

Address: 3033_0000h base + 60h offset = 3033_0060h



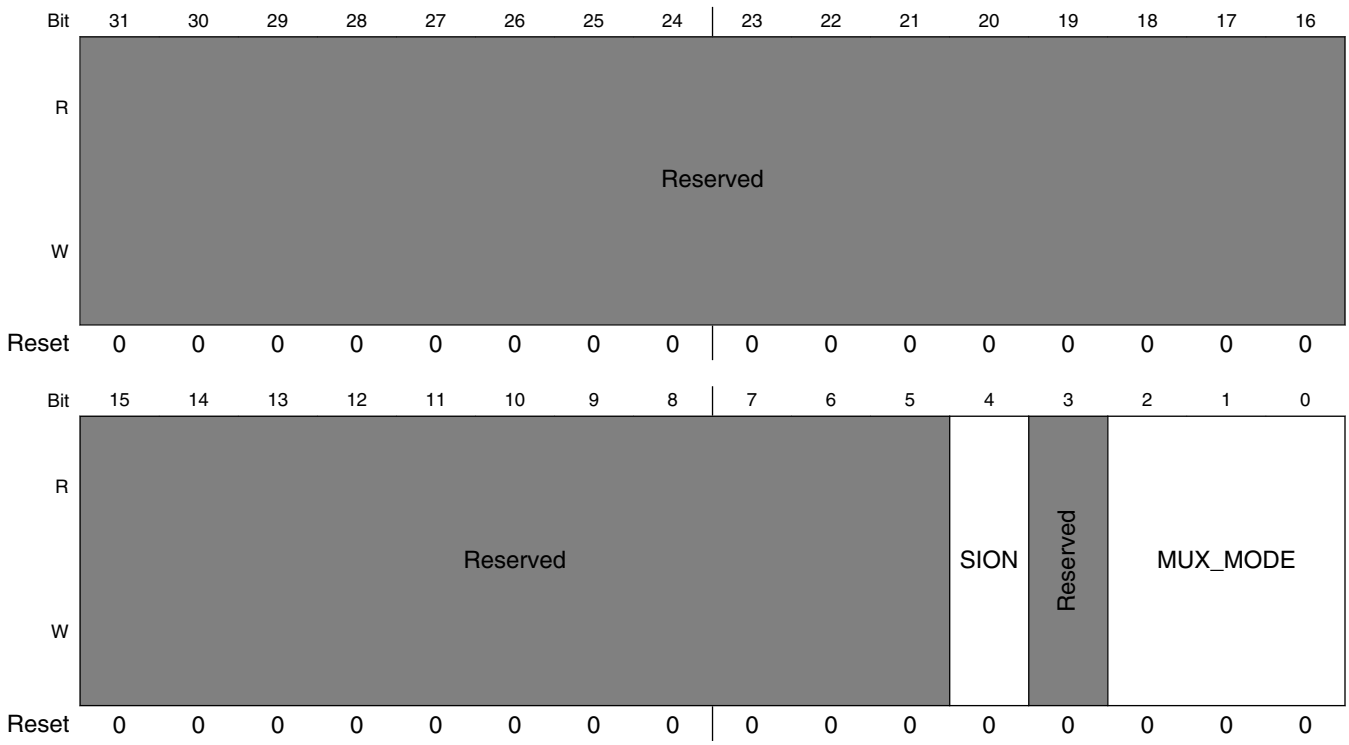
IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO14 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO14 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO14
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO14 000 **ALT0_GPIO1_IO14** — Select mux mode: ALT0 mux port: IO14 of instance: GPIO1 001 **ALT1_USB2_OTG_PWR** — Select mux mode: ALT1 mux port: OTG_PWR of instance: USB2 101 **ALT5_PWM3_OUT** — Select mux mode: ALT5 mux port: OUT of instance: PWM3 110 **ALT6_CCM_CLKO1** — Select mux mode: ALT6 mux port: CLKO1 of instance: CCM

8.2.5.21 SW_MUX_CTL_PAD_GPIO1_IO15 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15)

SW_MUX_CTL Register

Address: 3033_0000h base + 64h offset = 3033_0064h



IOMUXC_SW_MUX_CTL_PAD_GPIO1_IO15 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad GPIO1_IO15 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad GPIO1_IO15
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: GPIO1_IO15 000 **ALT0_GPIO1_IO15** — Select mux mode: ALT0 mux port: IO15 of instance: GPIO1 001 **ALT1_USB2_OTG_OC** — Select mux mode: ALT1 mux port: OTG_OC of instance: USB2 101 **ALT5_PWM4_OUT** — Select mux mode: ALT5 mux port: OUT of instance: PWM4 110 **ALT6_CCM_CLKO2** — Select mux mode: ALT6 mux port: CLKO2 of instance: CCM

8.2.5.22 SW_MUX_CTL_PAD_ENET_MDC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDC)

SW_MUX_CTL Register

Address: 3033_0000h base + 68h offset = 3033_0068h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_MDC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_MDC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_MDC 000 **ALT0_ENET1_MDC** — Select mux mode: ALT0 mux port: MDC of instance: ENET1 101 **ALT5_GPIO1_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO1

8.2.5.23 SW_MUX_CTL_PAD_ENET_MDIO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO)

SW_MUX_CTL Register

Address: 3033_0000h base + 6Ch offset = 3033_006Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

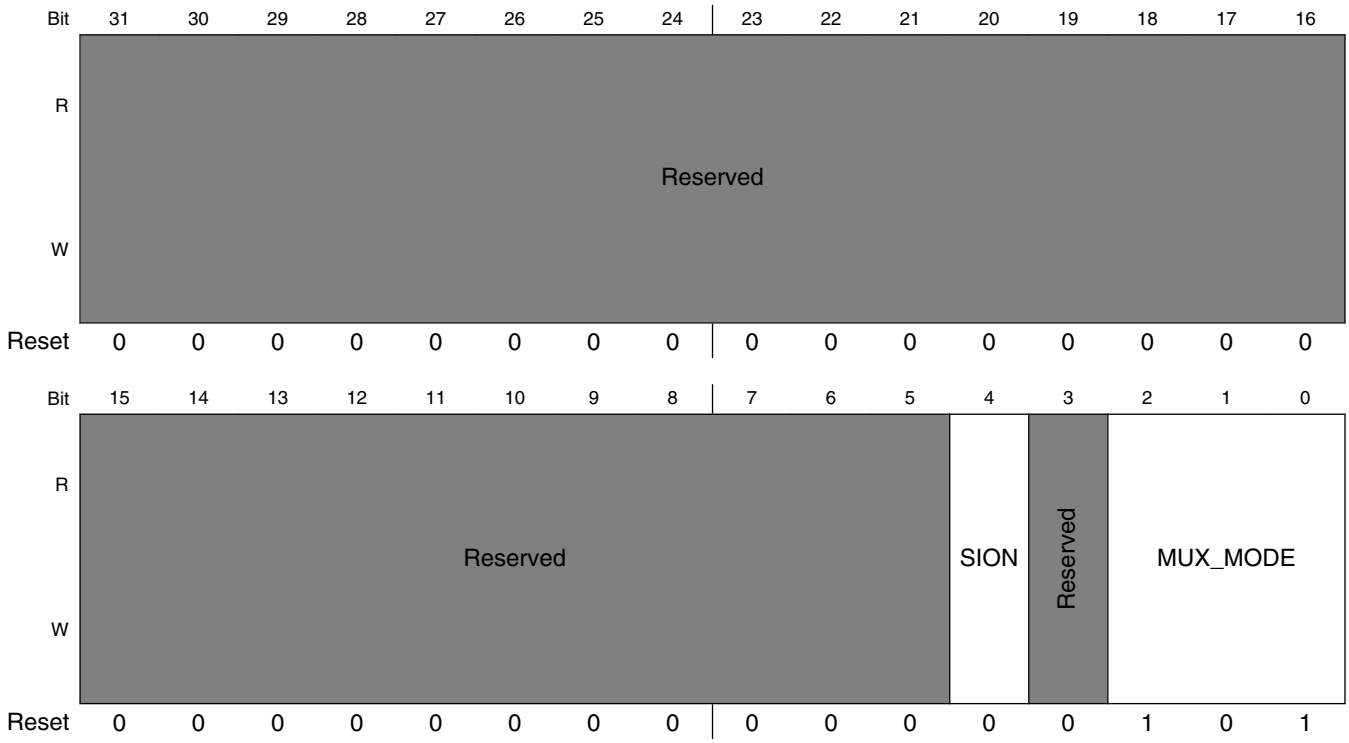
IOMUXC_SW_MUX_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_MDIO is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_MDIO
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_MDIO 000 **ALT0_ENET1_MDIO** — Select mux mode: ALT0 mux port: MDIO of instance: ENET1 101 **ALT5_GPIO1_IO17** — Select mux mode: ALT5 mux port: IO17 of instance: GPIO1

8.2.5.24 SW_MUX_CTL_PAD_ENET_TD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 70h offset = 3033_0070h



IOMUXC_SW_MUX_CTL_PAD_ENET_TD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TD3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TD3
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_TD3 000 **ALT0_ENET1_RGMII_TD3** — Select mux mode: ALT0 mux port: RGMII_TD3 of instance: ENET1 101 **ALT5_GPIO1_IO18** — Select mux mode: ALT5 mux port: IO18 of instance: GPIO1

8.2.5.25 SW_MUX_CTL_PAD_ENET_TD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 74h offset = 3033_0074h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

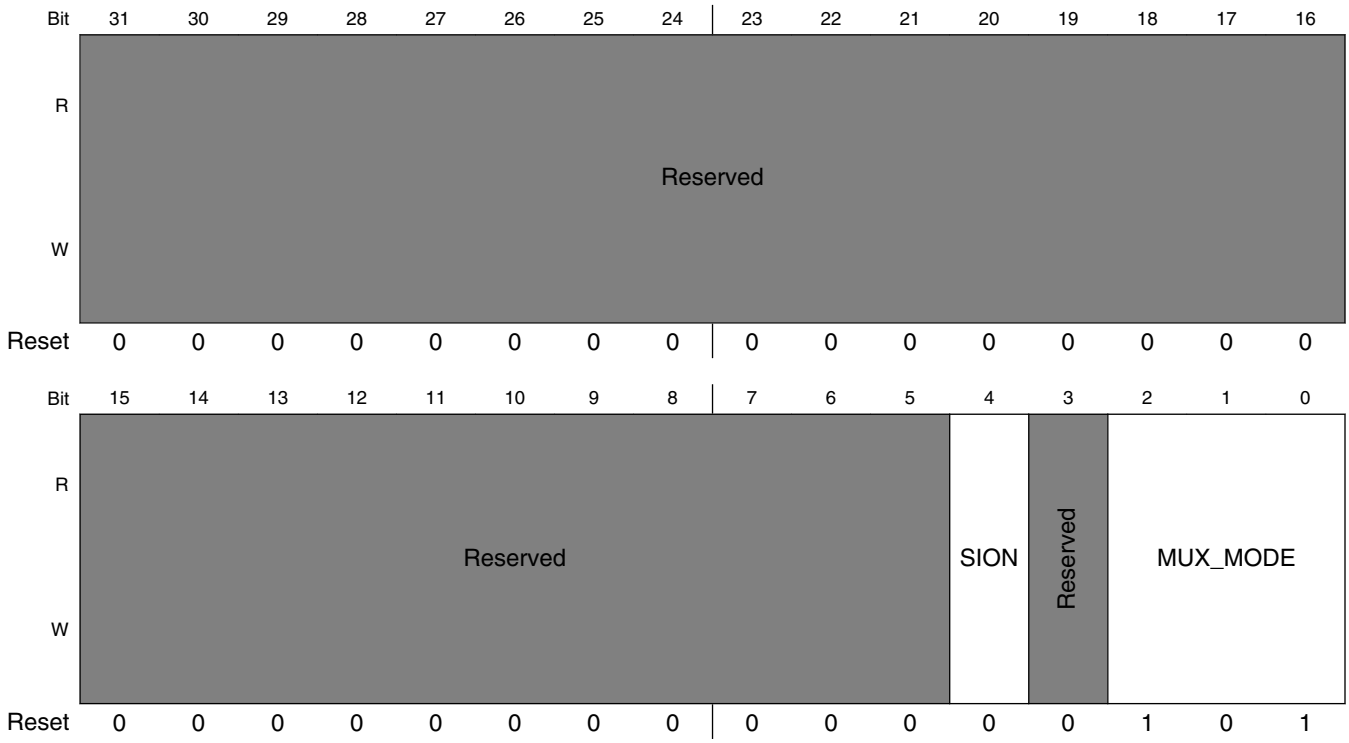
IOMUXC_SW_MUX_CTL_PAD_ENET_TD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TD2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TD2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ENET_TD2 000 **ALT0_ENET1_RGMII_TD2** — Select mux mode: ALT0 mux port: RGMII_TD2 of instance: ENET1 001 **ALT1_ENET1_TX_CLK** — Select mux mode: ALT1 mux port: TX_CLK of instance: ENET1 (input), mux port: ENET_REF_CLK_ROOT of CCM (output) 101 **ALT5_GPIO1_IO19** — Select mux mode: ALT5 mux port: IO19 of instance: GPIO1

8.2.5.26 SW_MUX_CTL_PAD_ENET_TD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 78h offset = 3033_0078h



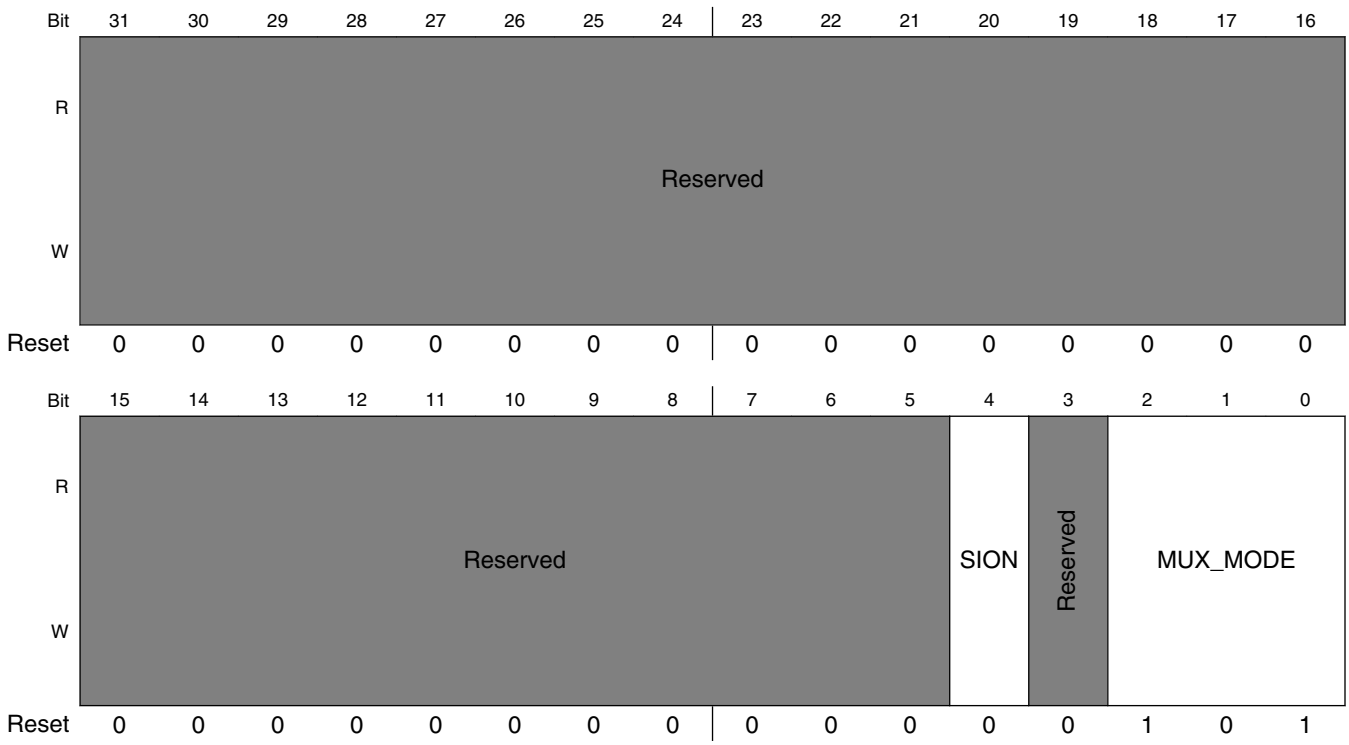
IOMUXC_SW_MUX_CTL_PAD_ENET_TD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TD1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TD1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_TD1 000 **ALT0_ENET1_RGMII_TD1** — Select mux mode: ALT0 mux port: RGMII_TD1 of instance: ENET1 101 **ALT5_GPIO1_IO20** — Select mux mode: ALT5 mux port: IO20 of instance: GPIO1

8.2.5.27 SW_MUX_CTL_PAD_ENET_TD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 7Ch offset = 3033_007Ch



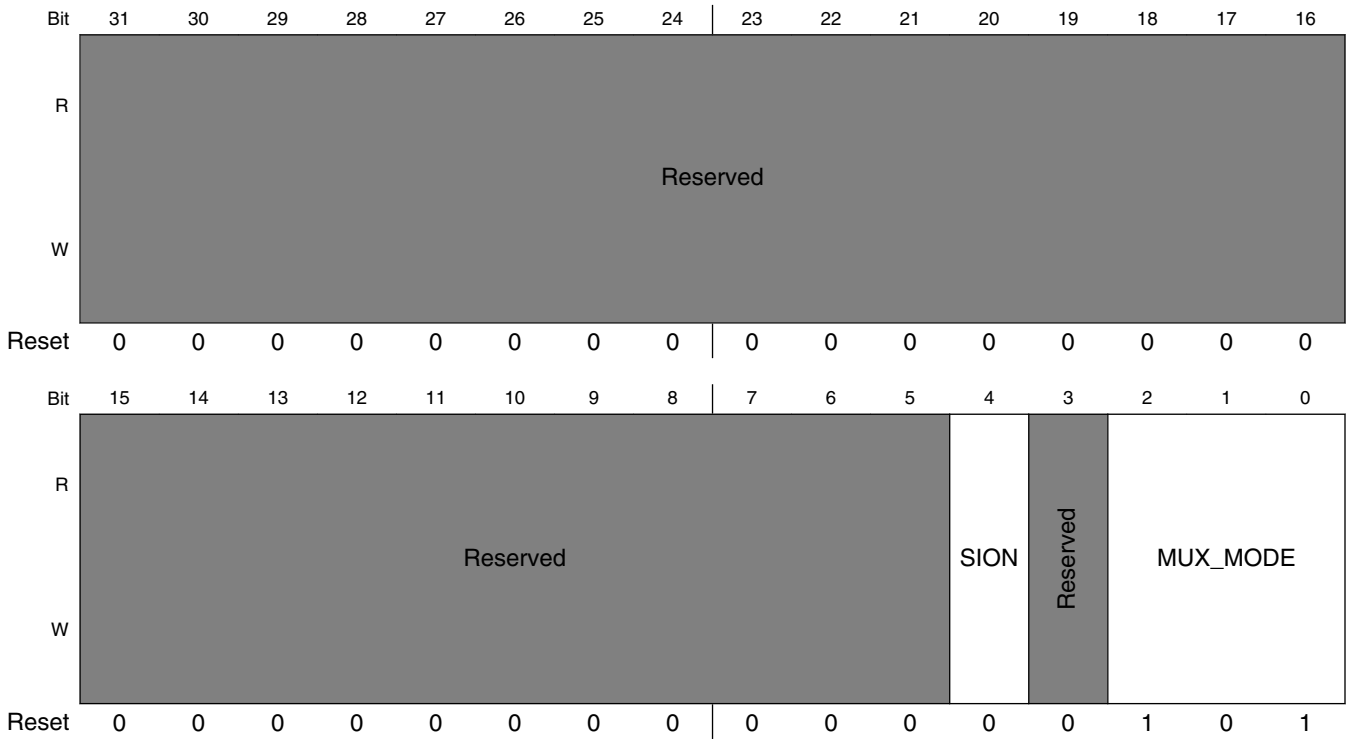
IOMUXC_SW_MUX_CTL_PAD_ENET_TD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_TD0 000 **ALT0_ENET1_RGMII_TD0** — Select mux mode: ALT0 mux port: RGMII_TD0 of instance: ENET1 101 **ALT5_GPIO1_IO21** — Select mux mode: ALT5 mux port: IO21 of instance: GPIO1

8.2.5.28 SW_MUX_CTL_PAD_ENET_TX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL)

SW_MUX_CTL Register

Address: 3033_0000h base + 80h offset = 3033_0080h



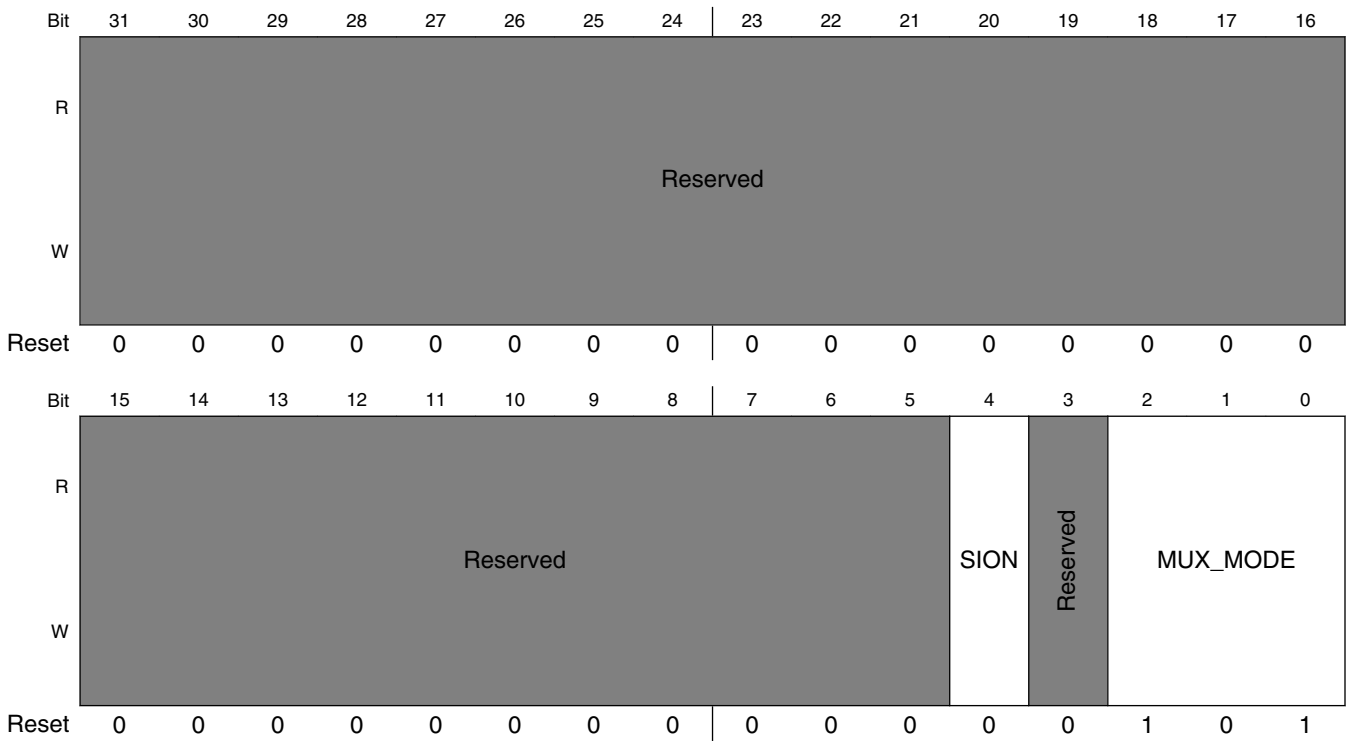
IOMUXC_SW_MUX_CTL_PAD_ENET_TX_CTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TX_CTL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TX_CTL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_TX_CTL 000 **ALT0_ENET1_RGMII_TX_CTL** — Select mux mode: ALT0 mux port: RGMII_TX_CTL of instance: ENET1 101 **ALT5_GPIO1_IO22** — Select mux mode: ALT5 mux port: IO22 of instance: GPIO1

8.2.5.29 SW_MUX_CTL_PAD_ENET_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 84h offset = 3033_0084h



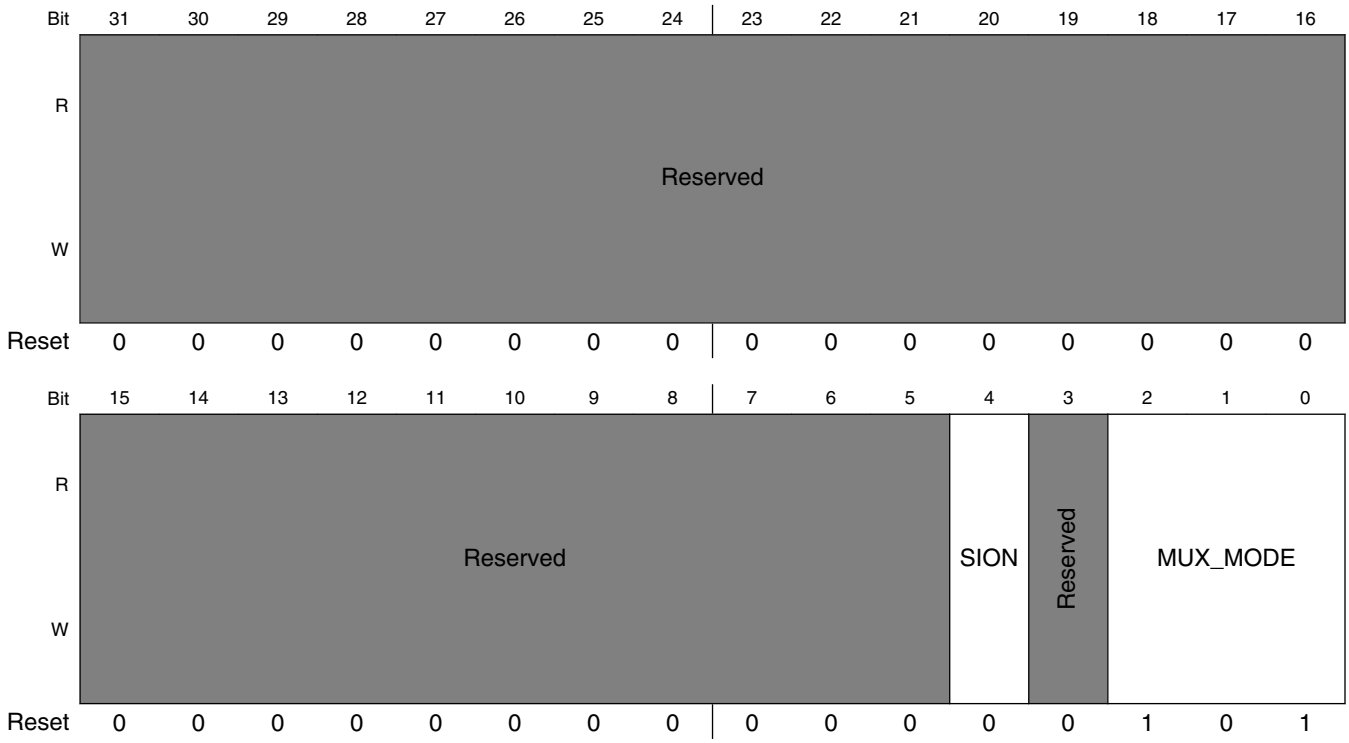
IOMUXC_SW_MUX_CTL_PAD_ENET_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_TXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_TXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ENET_TXC 000 **ALT0_ENET1_RGMII_TXC** — Select mux mode: ALT0 mux port: RGMII_TXC of instance: ENET1 001 **ALT1_ENET1_TX_ER** — Select mux mode: ALT1 mux port: TX_ER of instance: ENET1 101 **ALT5_GPIO1_IO23** — Select mux mode: ALT5 mux port: IO23 of instance: GPIO1

8.2.5.30 SW_MUX_CTL_PAD_ENET_RX_CTL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL)

SW_MUX_CTL Register

Address: 3033_0000h base + 88h offset = 3033_0088h



IOMUXC_SW_MUX_CTL_PAD_ENET_RX_CTL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RX_CTL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RX_CTL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_RX_CTL 000 **ALT0_ENET1_RGMII_RX_CTL** — Select mux mode: ALT0 mux port: RGMII_RX_CTL of instance: ENET1 101 **ALT5_GPIO1_IO24** — Select mux mode: ALT5 mux port: IO24 of instance: GPIO1

8.2.5.31 SW_MUX_CTL_PAD_ENET_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 8Ch offset = 3033_008Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

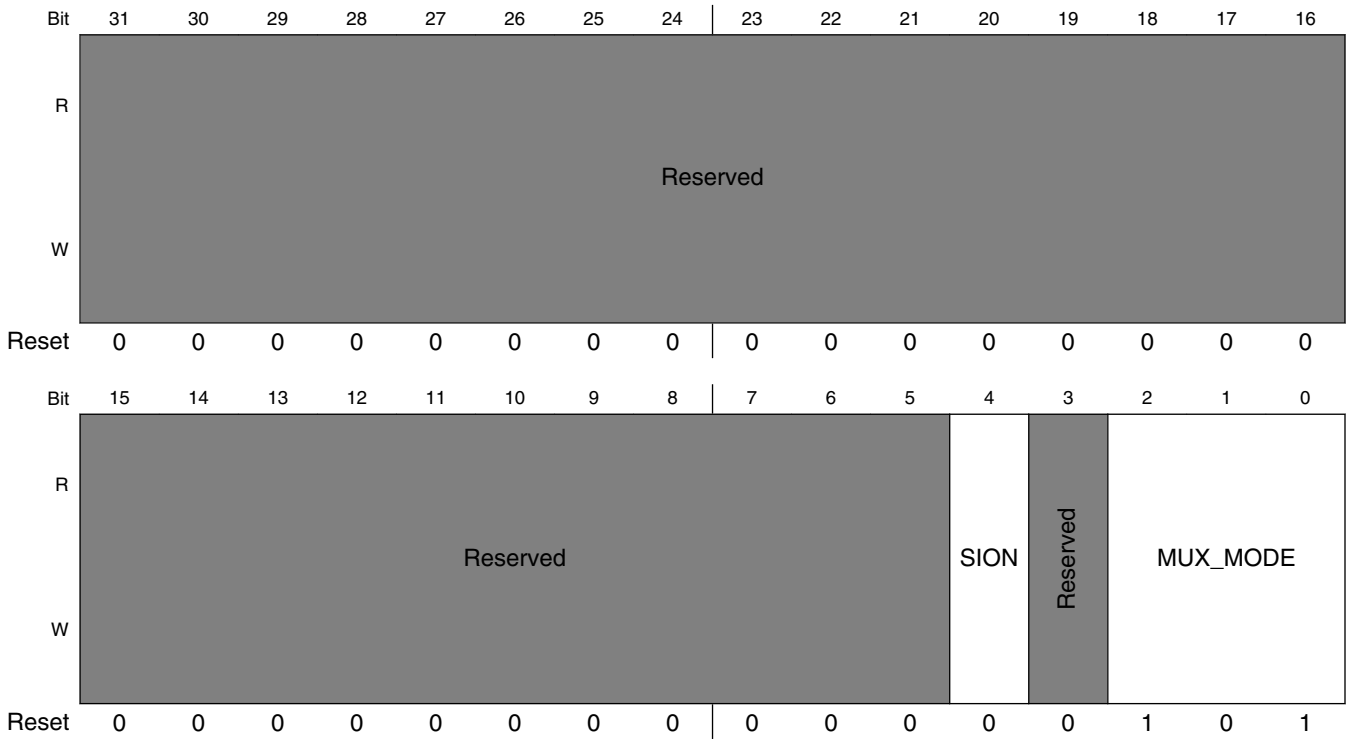
IOMUXC_SW_MUX_CTL_PAD_ENET_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ENET_RXC 000 **ALT0_ENET1_RGMII_RXC** — Select mux mode: ALT0 mux port: RGMII_RXC of instance: ENET1 001 **ALT1_ENET1_RX_ER** — Select mux mode: ALT1 mux port: RX_ER of instance: ENET1 101 **ALT5_GPIO1_IO25** — Select mux mode: ALT5 mux port: IO25 of instance: GPIO1

8.2.5.32 SW_MUX_CTL_PAD_ENET_RD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 90h offset = 3033_0090h



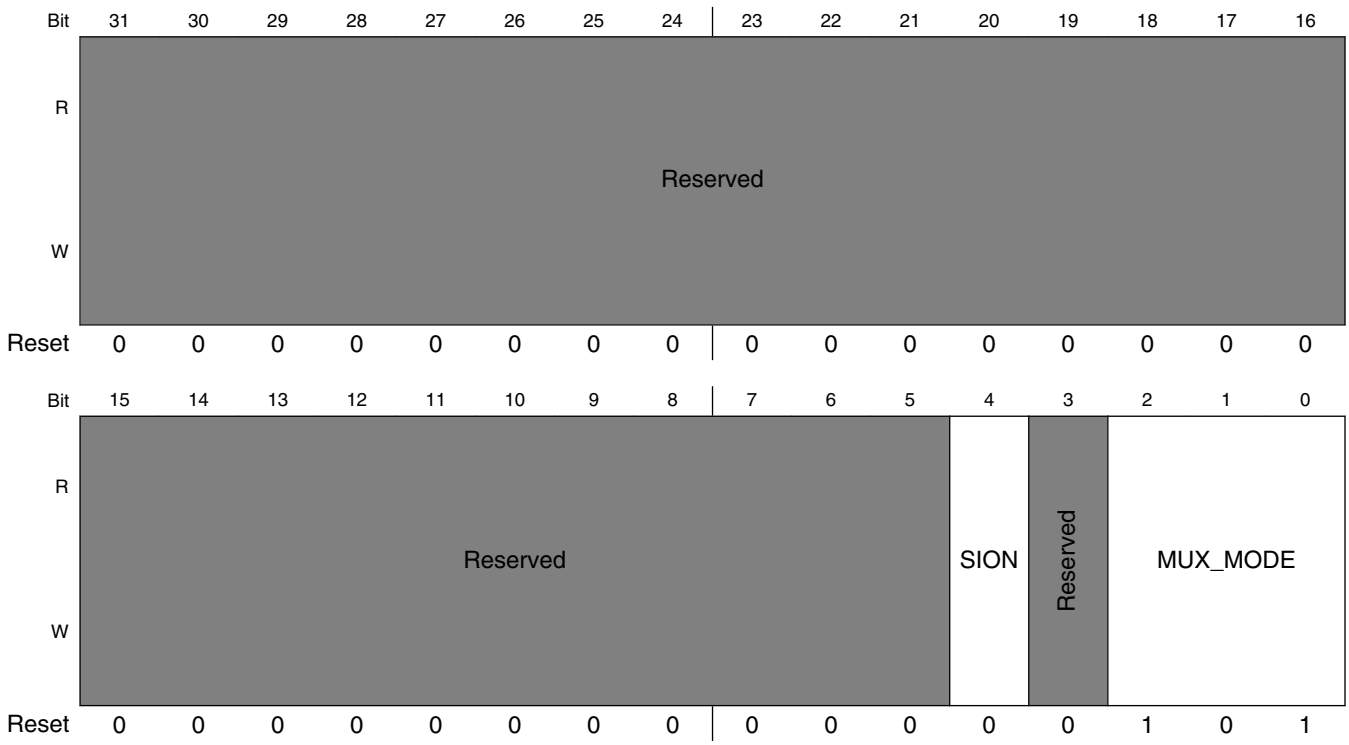
IOMUXC_SW_MUX_CTL_PAD_ENET_RD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_RD0 000 **ALT0_ENET1_RGMII_RD0** — Select mux mode: ALT0 mux port: RGMII_RD0 of instance: ENET1 101 **ALT5_GPIO1_IO26** — Select mux mode: ALT5 mux port: IO26 of instance: GPIO1

8.2.5.33 SW_MUX_CTL_PAD_ENET_RD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 94h offset = 3033_0094h



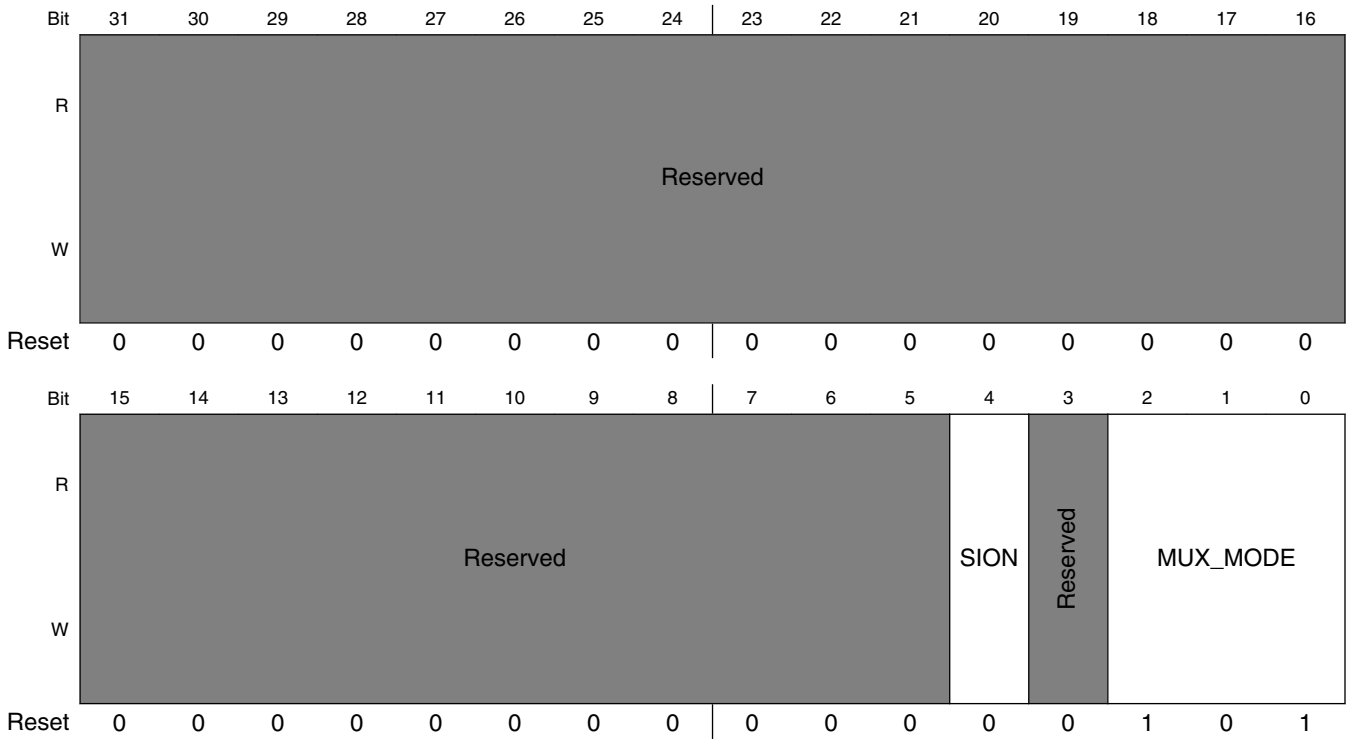
IOMUXC_SW_MUX_CTL_PAD_ENET_RD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RD1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RD1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_RD1 000 **ALT0_ENET1_RGMII_RD1** — Select mux mode: ALT0 mux port: RGMII_RD1 of instance: ENET1 101 **ALT5_GPIO1_IO27** — Select mux mode: ALT5 mux port: IO27 of instance: GPIO1

8.2.5.34 SW_MUX_CTL_PAD_ENET_RD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 98h offset = 3033_0098h



IOMUXC_SW_MUX_CTL_PAD_ENET_RD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RD2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RD2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_RD2 000 **ALT0_ENET1_RGMII_RD2** — Select mux mode: ALT0 mux port: RGMII_RD2 of instance: ENET1 101 **ALT5_GPIO1_IO28** — Select mux mode: ALT5 mux port: IO28 of instance: GPIO1

8.2.5.35 SW_MUX_CTL_PAD_ENET_RD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ENET_RD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 9Ch offset = 3033_009Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

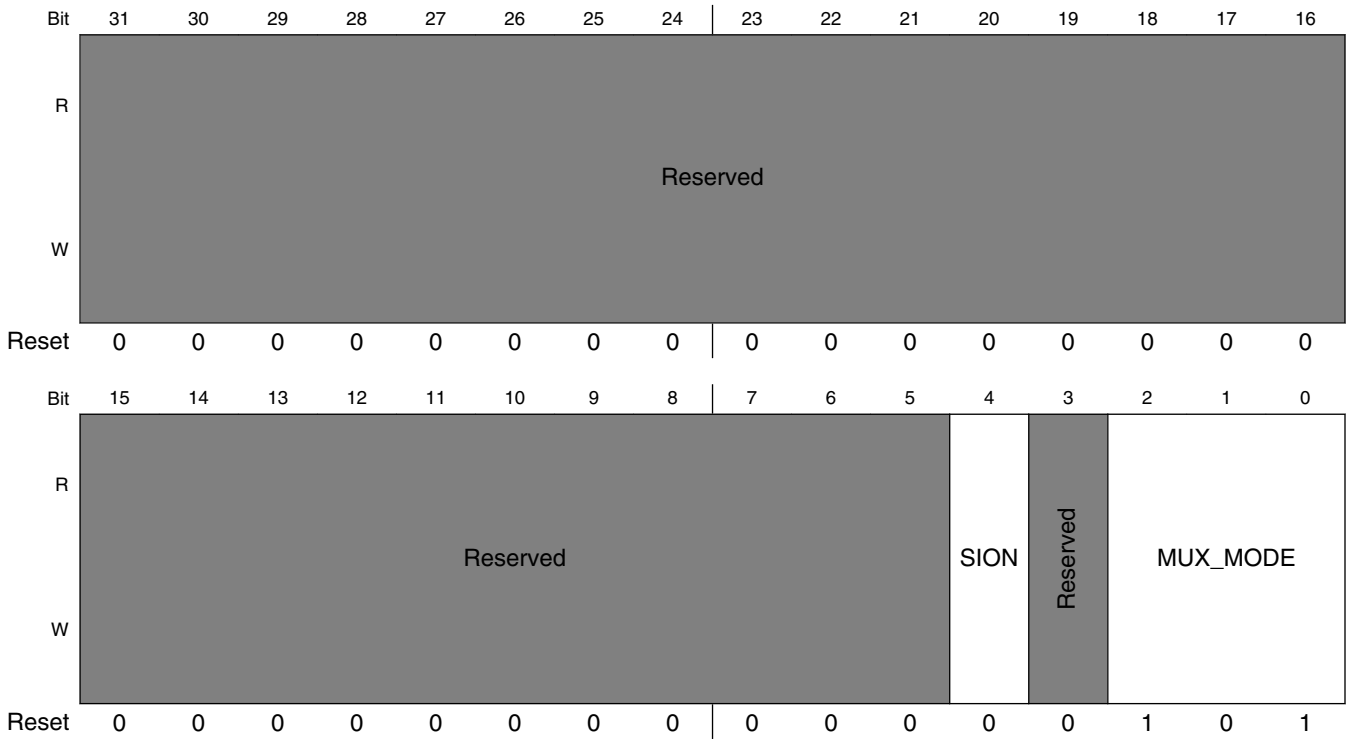
IOMUXC_SW_MUX_CTL_PAD_ENET_RD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ENET_RD3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ENET_RD3
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: ENET_RD3 000 **ALT0_ENET1_RGMII_RD3** — Select mux mode: ALT0 mux port: RGMII_RD3 of instance: ENET1 101 **ALT5_GPIO1_IO29** — Select mux mode: ALT5 mux port: IO29 of instance: GPIO1

8.2.5.36 SW_MUX_CTL_PAD_SD1_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + A0h offset = 3033_00A0h



IOMUXC_SW_MUX_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_CLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_CLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_CLK 000 **ALT0_USDHC1_CLK** — Select mux mode: ALT0 mux port: CLK of instance: USDHC1 101 **ALT5_GPIO2_IO00** — Select mux mode: ALT5 mux port: IO00 of instance: GPIO2

8.2.5.37 SW_MUX_CTL_PAD_SD1_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_CMD)

SW_MUX_CTL Register

Address: 3033_0000h base + A4h offset = 3033_00A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

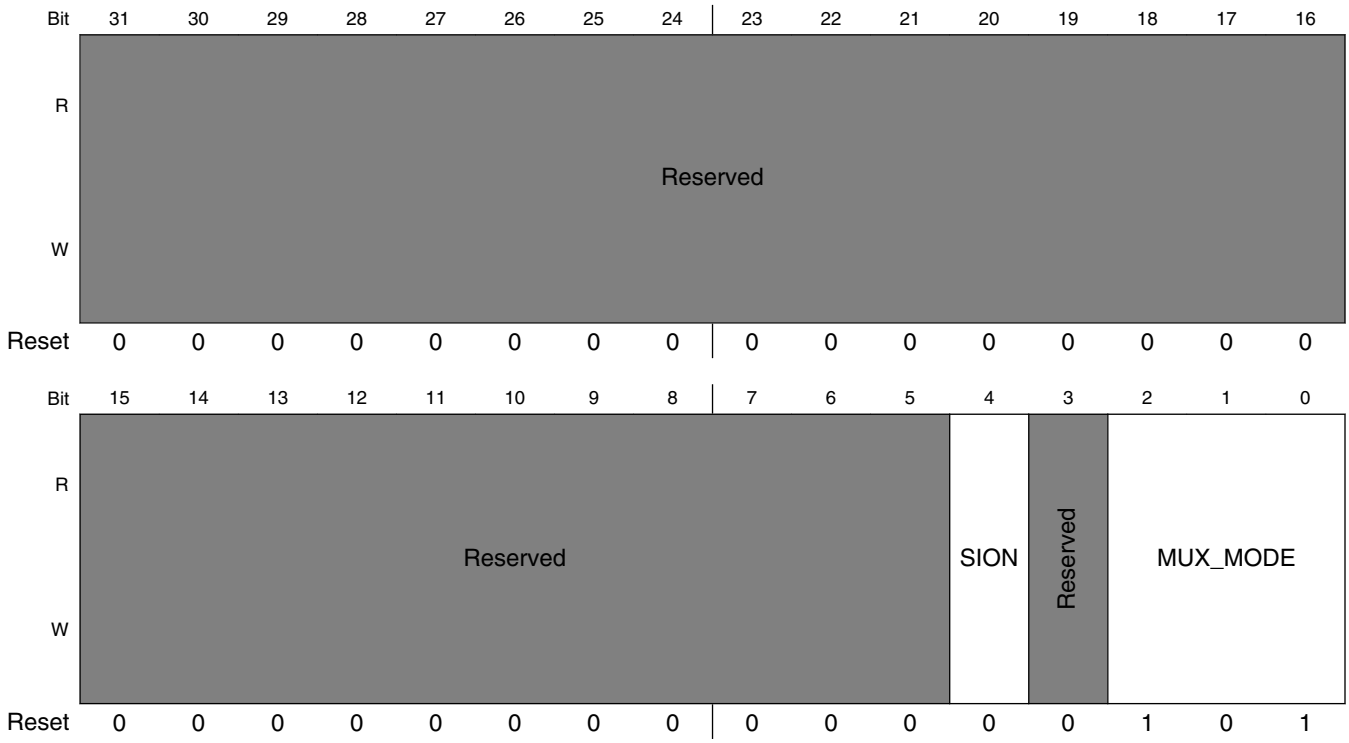
IOMUXC_SW_MUX_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_CMD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_CMD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_CMD 000 **ALT0_USDHC1_CMD** — Select mux mode: ALT0 mux port: CMD of instance: USDHC1 101 **ALT5_GPIO2_IO01** — Select mux mode: ALT5 mux port: IO01 of instance: GPIO2

8.2.5.38 SW_MUX_CTL_PAD_SD1_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0)

SW_MUX_CTL Register

Address: 3033_0000h base + A8h offset = 3033_00A8h



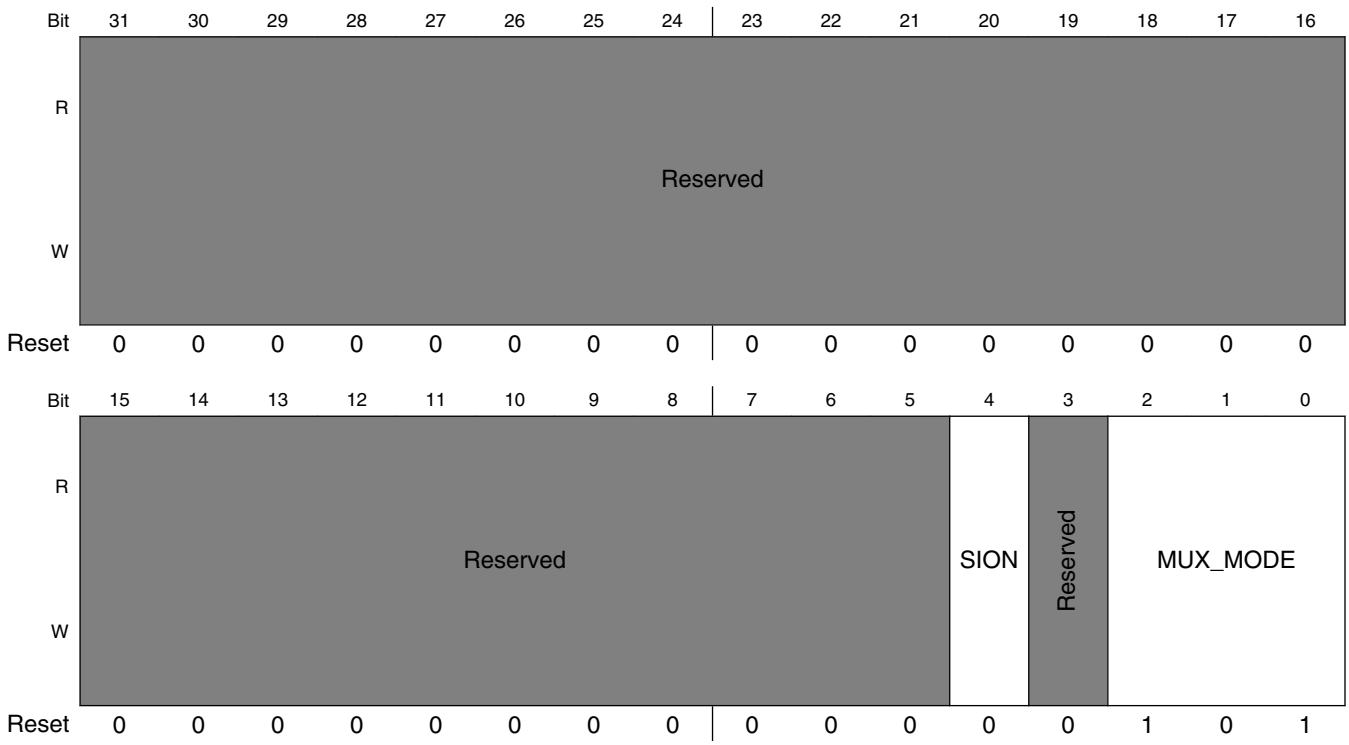
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA0 000 **ALT0_USDHC1_DATA0** — Select mux mode: ALT0 mux port: DATA0 of instance: USDHC1 101 **ALT5_GPIO2_IO02** — Select mux mode: ALT5 mux port: IO02 of instance: GPIO2

8.2.5.39 SW_MUX_CTL_PAD_SD1_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1)

SW_MUX_CTL Register

Address: 3033_0000h base + ACh offset = 3033_00ACh



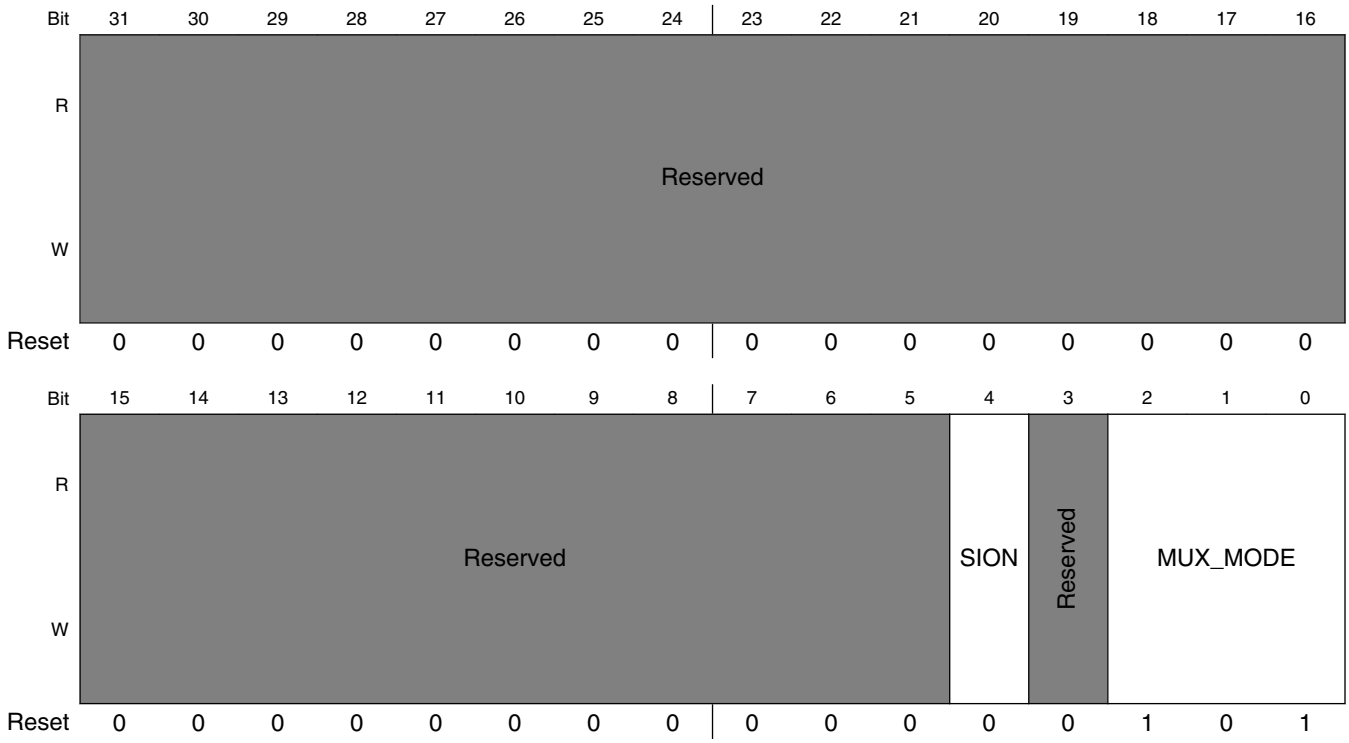
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA1 000 **ALT0_USDHC1_DATA1** — Select mux mode: ALT0 mux port: DATA1 of instance: USDHC1 101 **ALT5_GPIO2_IO03** — Select mux mode: ALT5 mux port: IO03 of instance: GPIO2

8.2.5.40 SW_MUX_CTL_PAD_SD1_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2)

SW_MUX_CTL Register

Address: 3033_0000h base + B0h offset = 3033_00B0h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA2 000 **ALT0_USDHC1_DATA2** — Select mux mode: ALT0 mux port: DATA2 of instance: USDHC1 101 **ALT5_GPIO2_IO04** — Select mux mode: ALT5 mux port: IO04 of instance: GPIO2

8.2.5.41 SW_MUX_CTL_PAD_SD1_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3)

SW_MUX_CTL Register

Address: 3033_0000h base + B4h offset = 3033_00B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

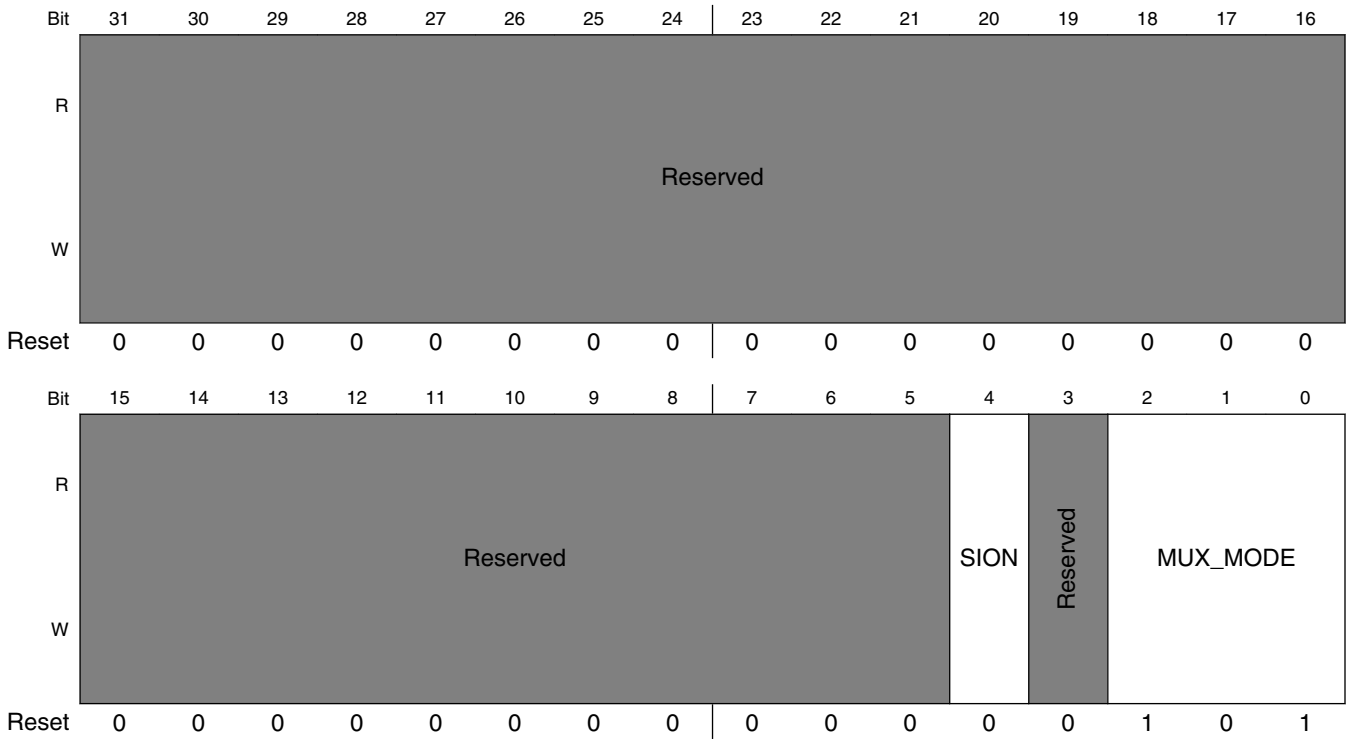
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA3
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA3 000 **ALT0_USDHC1_DATA3** — Select mux mode: ALT0 mux port: DATA3 of instance: USDHC1 101 **ALT5_GPIO2_IO05** — Select mux mode: ALT5 mux port: IO05 of instance: GPIO2

8.2.5.42 SW_MUX_CTL_PAD_SD1_DATA4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4)

SW_MUX_CTL Register

Address: 3033_0000h base + B8h offset = 3033_00B8h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA4 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA4
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA4 000 **ALT0_USDHC1_DATA4** — Select mux mode: ALT0 mux port: DATA4 of instance: USDHC1 101 **ALT5_GPIO2_IO06** — Select mux mode: ALT5 mux port: IO06 of instance: GPIO2

8.2.5.43 SW_MUX_CTL_PAD_SD1_DATA5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5)

SW_MUX_CTL Register

Address: 3033_0000h base + BCh offset = 3033_00BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

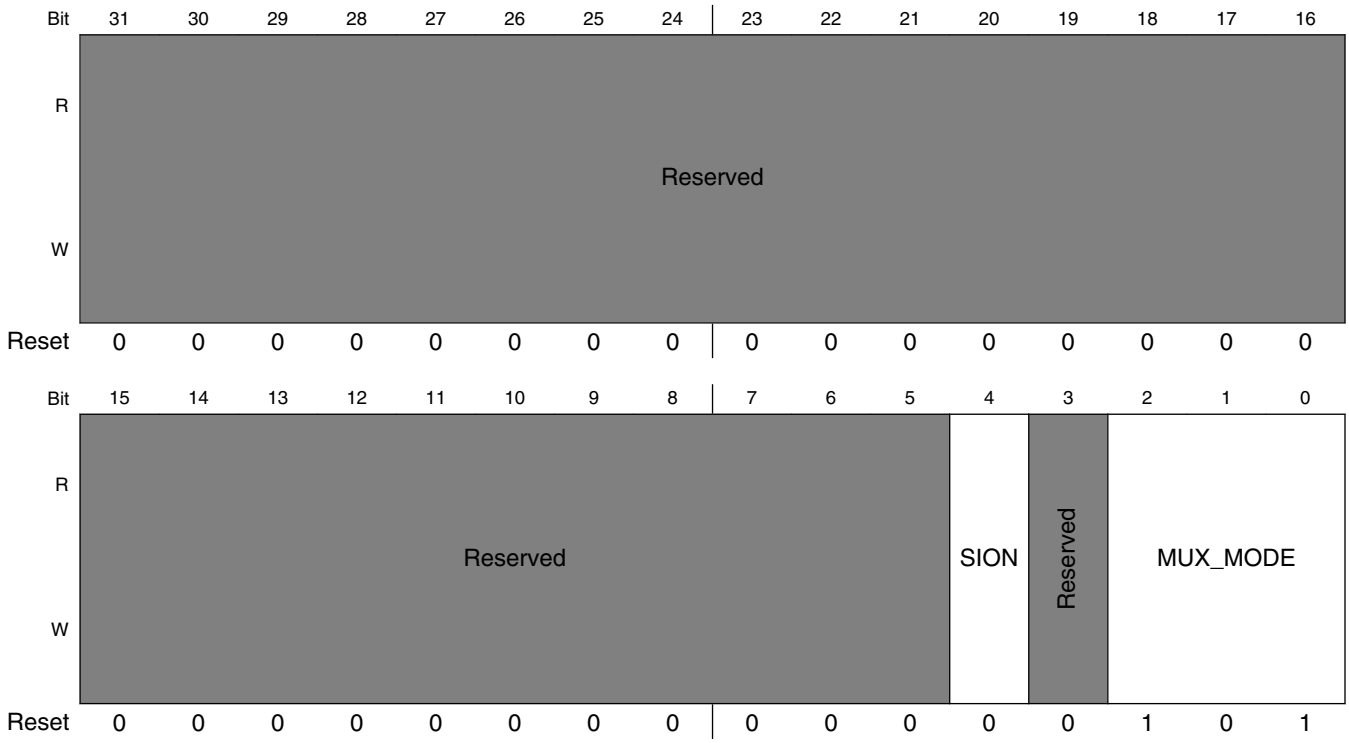
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA5 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA5
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA5 000 **ALT0_USDHC1_DATA5** — Select mux mode: ALT0 mux port: DATA5 of instance: USDHC1 101 **ALT5_GPIO2_IO07** — Select mux mode: ALT5 mux port: IO07 of instance: GPIO2

8.2.5.44 SW_MUX_CTL_PAD_SD1_DATA6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6)

SW_MUX_CTL Register

Address: 3033_0000h base + C0h offset = 3033_00C0h



IOMUXC_SW_MUX_CTL_PAD_SD1_DATA6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA6 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA6
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA6 000 **ALT0_USDHC1_DATA6** — Select mux mode: ALT0 mux port: DATA6 of instance: USDHC1 101 **ALT5_GPIO2_IO08** — Select mux mode: ALT5 mux port: IO08 of instance: GPIO2

8.2.5.45 SW_MUX_CTL_PAD_SD1_DATA7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7)

SW_MUX_CTL Register

Address: 3033_0000h base + C4h offset = 3033_00C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

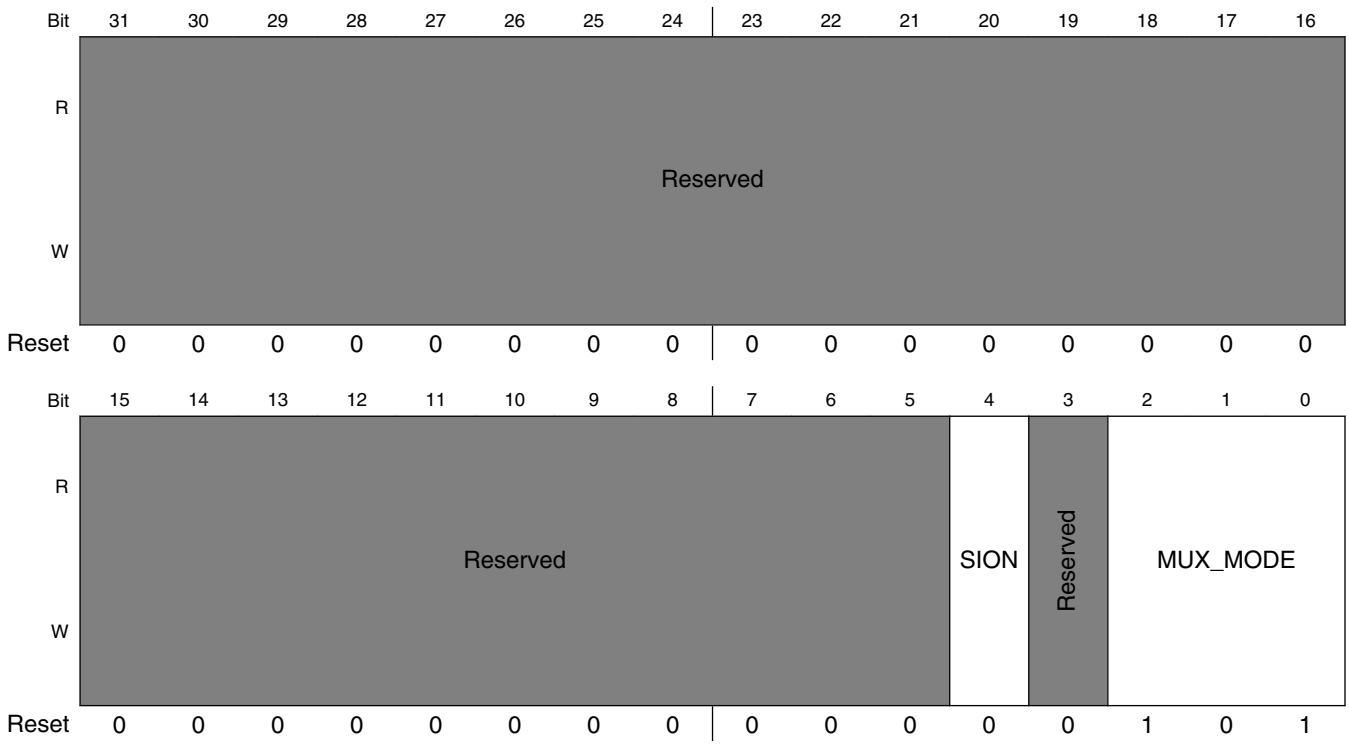
IOMUXC_SW_MUX_CTL_PAD_SD1_DATA7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_DATA7 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_DATA7
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_DATA7 000 **ALT0_USDHC1_DATA7** — Select mux mode: ALT0 mux port: DATA7 of instance: USDHC1 101 **ALT5_GPIO2_IO09** — Select mux mode: ALT5 mux port: IO09 of instance: GPIO2

8.2.5.46 SW_MUX_CTL_PAD_SD1_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B)

SW_MUX_CTL Register

Address: 3033_0000h base + C8h offset = 3033_00C8h



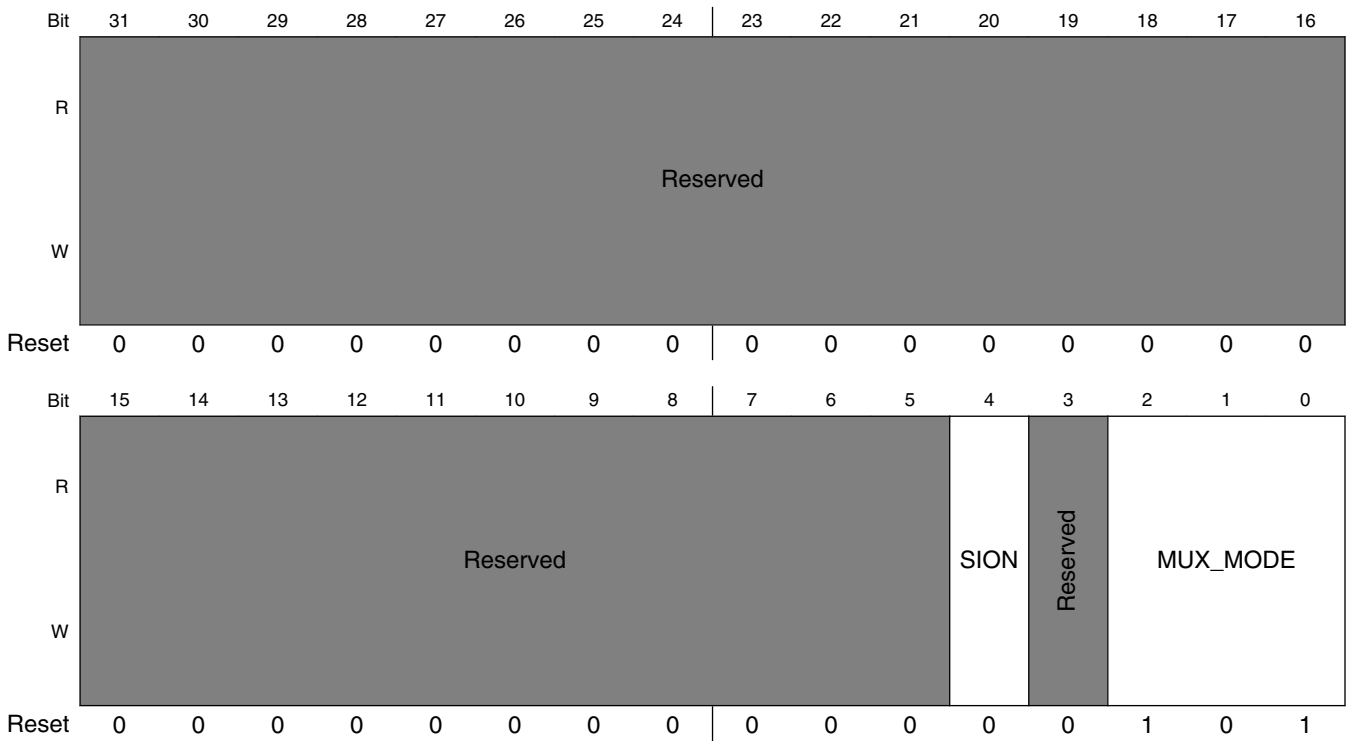
IOMUXC_SW_MUX_CTL_PAD_SD1_RESET_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_RESET_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_RESET_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_RESET_B 000 **ALT0_USDHC1_RESET_B** — Select mux mode: ALT0 mux port: RESET_B of instance: USDHC1 101 **ALT5_GPIO2_IO10** — Select mux mode: ALT5 mux port: IO10 of instance: GPIO2

8.2.5.47 SW_MUX_CTL_PAD_SD1_STROBE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE)

SW_MUX_CTL Register

Address: 3033_0000h base + CCh offset = 3033_00CCh



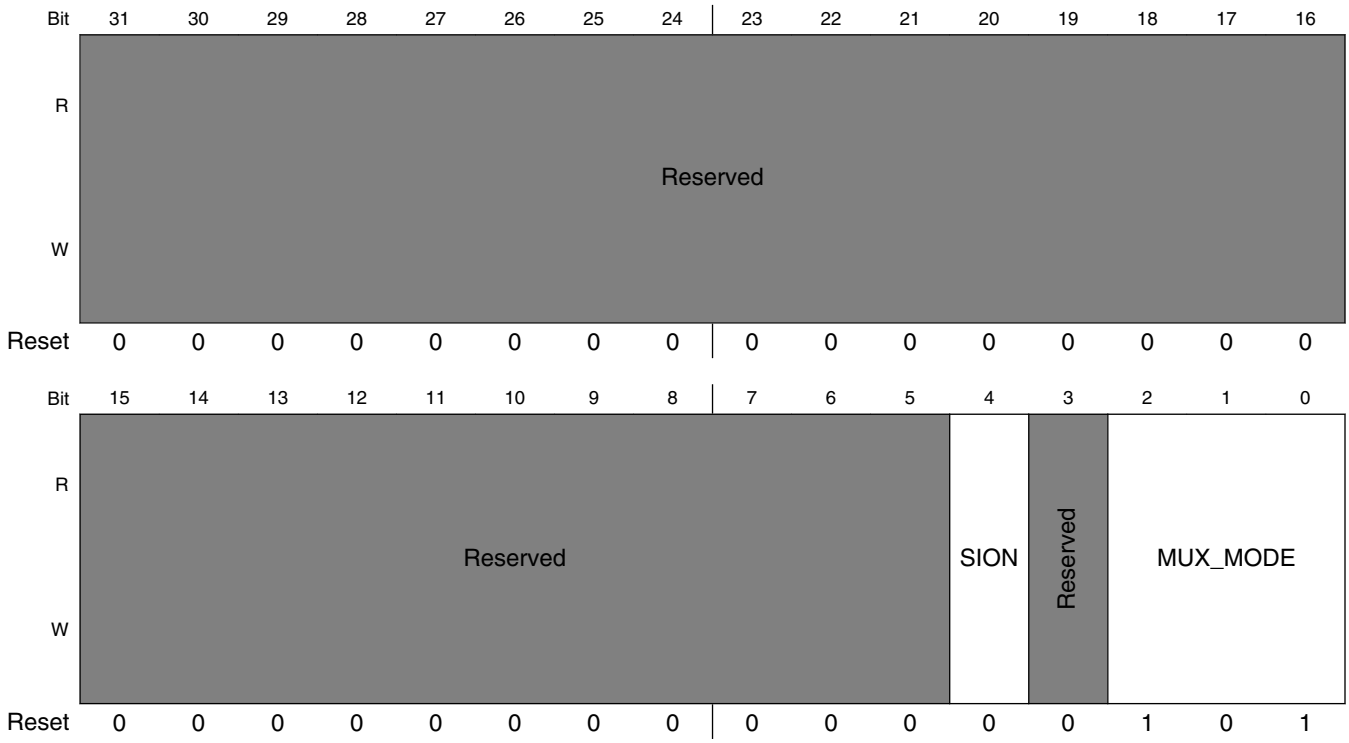
IOMUXC_SW_MUX_CTL_PAD_SD1_STROBE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD1_STROBE is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD1_STROBE
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD1_STROBE 000 **ALT0_USDHC1_STROBE** — Select mux mode: ALT0 mux port: STROBE of instance: USDHC1 101 **ALT5_GPIO2_IO11** — Select mux mode: ALT5 mux port: IO11 of instance: GPIO2

8.2.5.48 SW_MUX_CTL_PAD_SD2_CD_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B)

SW_MUX_CTL Register

Address: 3033_0000h base + D0h offset = 3033_00D0h



IOMUXC_SW_MUX_CTL_PAD_SD2_CD_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_CD_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_CD_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_CD_B 000 **ALT0_USDHC2_CD_B** — Select mux mode: ALT0 mux port: CD_B of instance: USDHC2 101 **ALT5_GPIO2_IO12** — Select mux mode: ALT5 mux port: IO12 of instance: GPIO2

8.2.5.49 SW_MUX_CTL_PAD_SD2_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + D4h offset = 3033_00D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_CLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_CLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_CLK 000 **ALT0_USDHC2_CLK** — Select mux mode: ALT0 mux port: CLK of instance: USDHC2 101 **ALT5_GPIO2_IO13** — Select mux mode: ALT5 mux port: IO13 of instance: GPIO2

8.2.5.50 SW_MUX_CTL_PAD_SD2_CMD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_CMD)

SW_MUX_CTL Register

Address: 3033_0000h base + D8h offset = 3033_00D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_CMD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_CMD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_CMD 000 **ALT0_USDHC2_CMD** — Select mux mode: ALT0 mux port: CMD of instance: USDHC2 101 **ALT5_GPIO2_IO14** — Select mux mode: ALT5 mux port: IO14 of instance: GPIO2

8.2.5.51 SW_MUX_CTL_PAD_SD2_DATA0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0)

SW_MUX_CTL Register

Address: 3033_0000h base + DCh offset = 3033_00DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

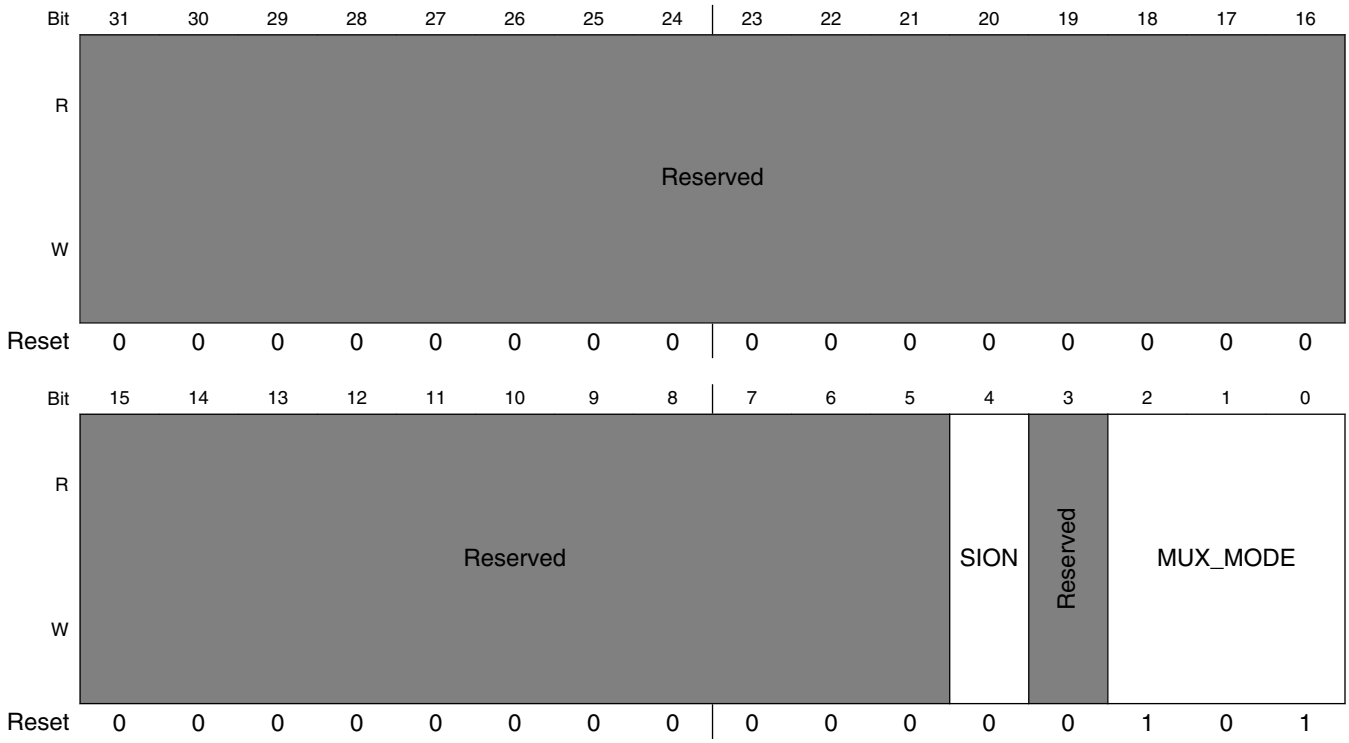
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_DATA0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_DATA0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_DATA0 000 **ALT0_USDHC2_DATA0** — Select mux mode: ALT0 mux port: DATA0 of instance: USDHC2 101 **ALT5_GPIO2_IO15** — Select mux mode: ALT5 mux port: IO15 of instance: GPIO2

8.2.5.52 SW_MUX_CTL_PAD_SD2_DATA1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1)

SW_MUX_CTL Register

Address: 3033_0000h base + E0h offset = 3033_00E0h



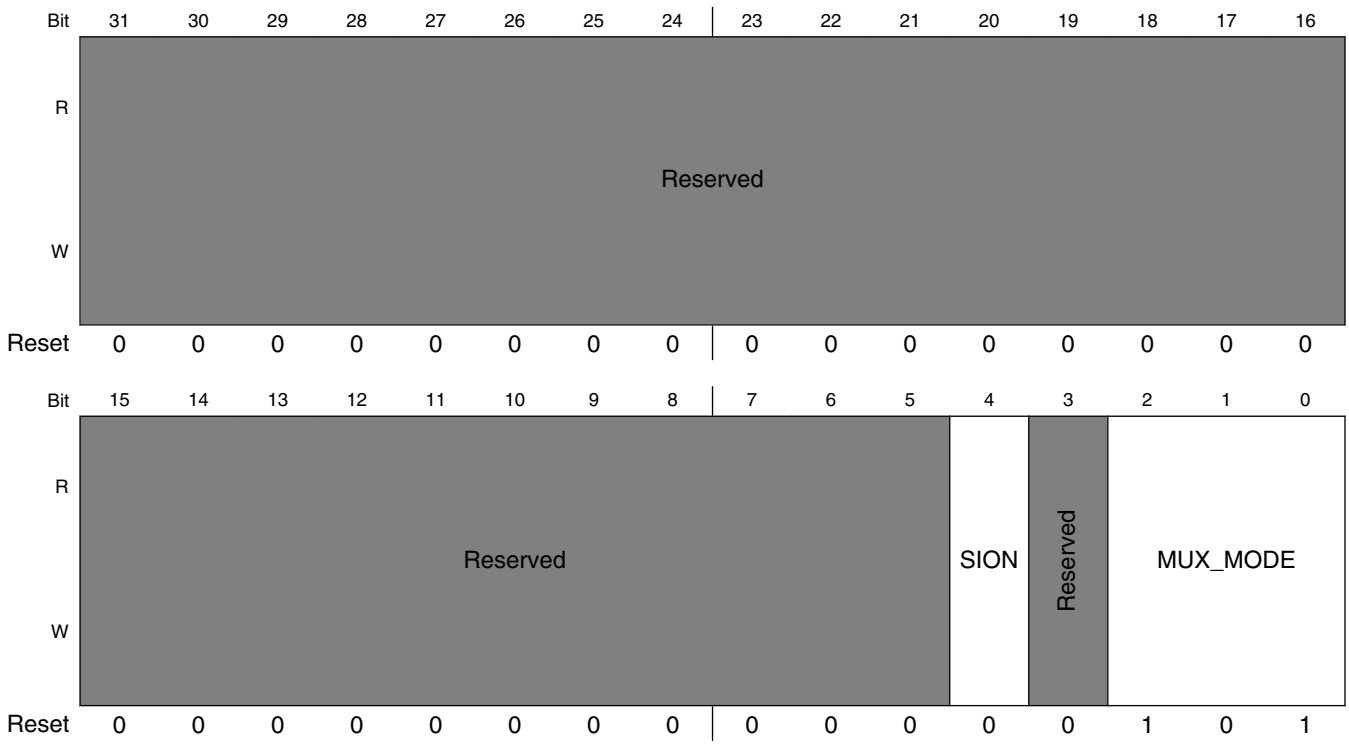
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_DATA1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_DATA1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_DATA1 000 **ALT0_USDHC2_DATA1** — Select mux mode: ALT0 mux port: DATA1 of instance: USDHC2 101 **ALT5_GPIO2_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO2

8.2.5.53 SW_MUX_CTL_PAD_SD2_DATA2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2)

SW_MUX_CTL Register

Address: 3033_0000h base + E4h offset = 3033_00E4h



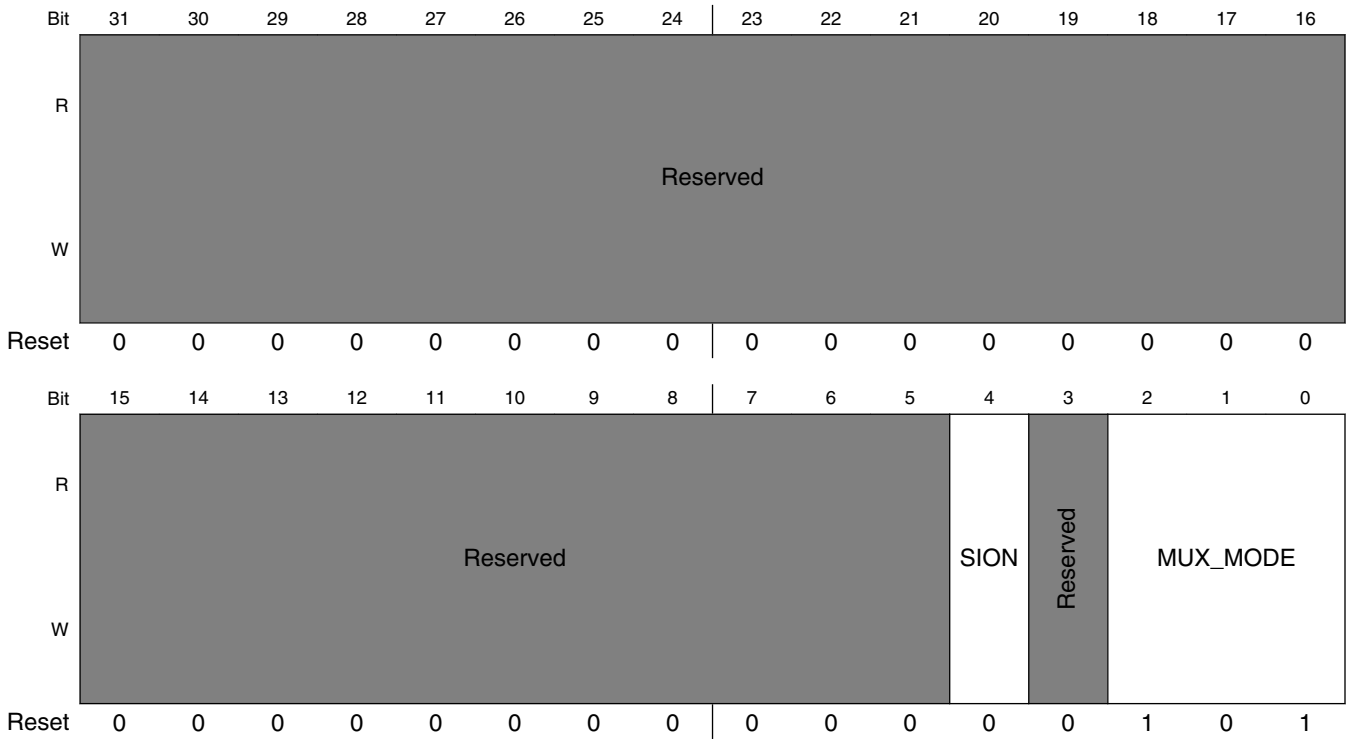
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_DATA2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_DATA2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_DATA2 000 **ALT0_USDHC2_DATA2** — Select mux mode: ALT0 mux port: DATA2 of instance: USDHC2 101 **ALT5_GPIO2_IO17** — Select mux mode: ALT5 mux port: IO17 of instance: GPIO2

8.2.5.54 SW_MUX_CTL_PAD_SD2_DATA3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3)

SW_MUX_CTL Register

Address: 3033_0000h base + E8h offset = 3033_00E8h



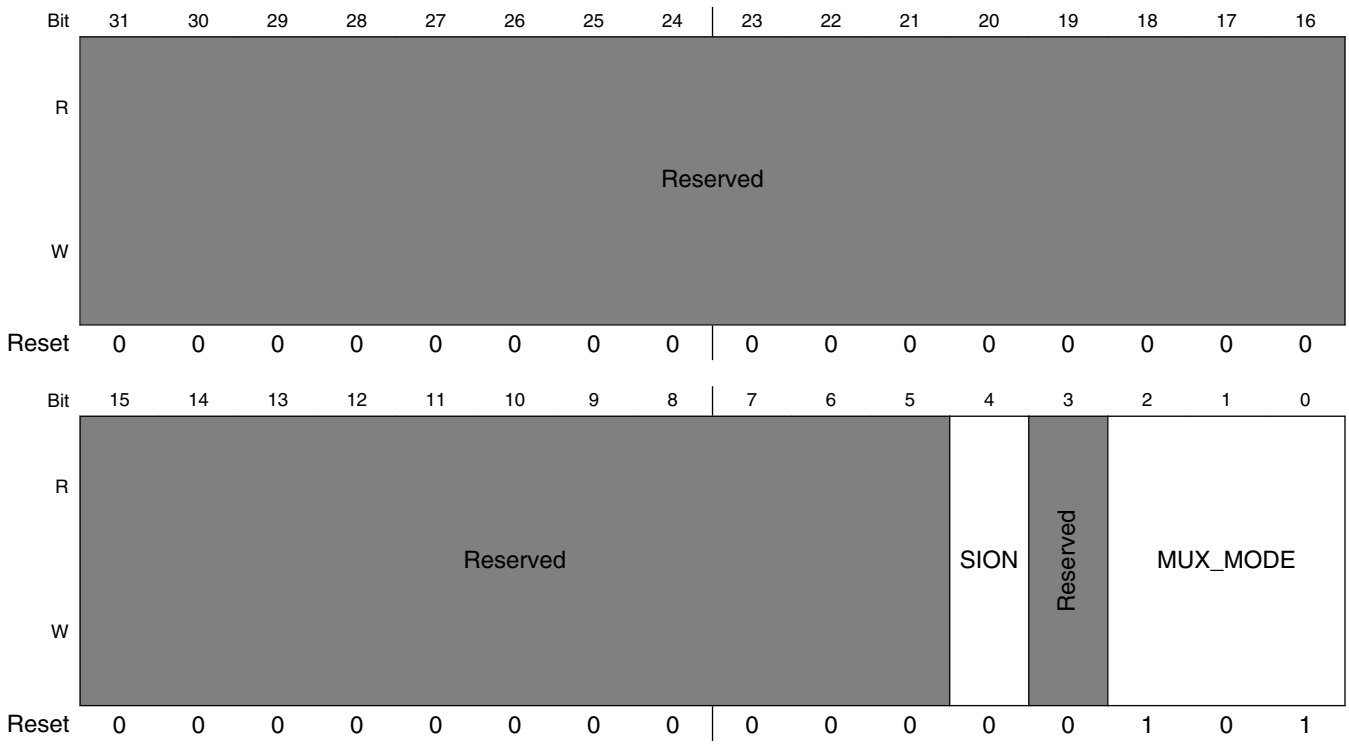
IOMUXC_SW_MUX_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_DATA3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_DATA3
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_DATA3 000 **ALT0_USDHC2_DATA3** — Select mux mode: ALT0 mux port: DATA3 of instance: USDHC2 101 **ALT5_GPIO2_IO18** — Select mux mode: ALT5 mux port: IO18 of instance: GPIO2

8.2.5.55 SW_MUX_CTL_PAD_SD2_RESET_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B)

SW_MUX_CTL Register

Address: 3033_0000h base + ECh offset = 3033_00ECh



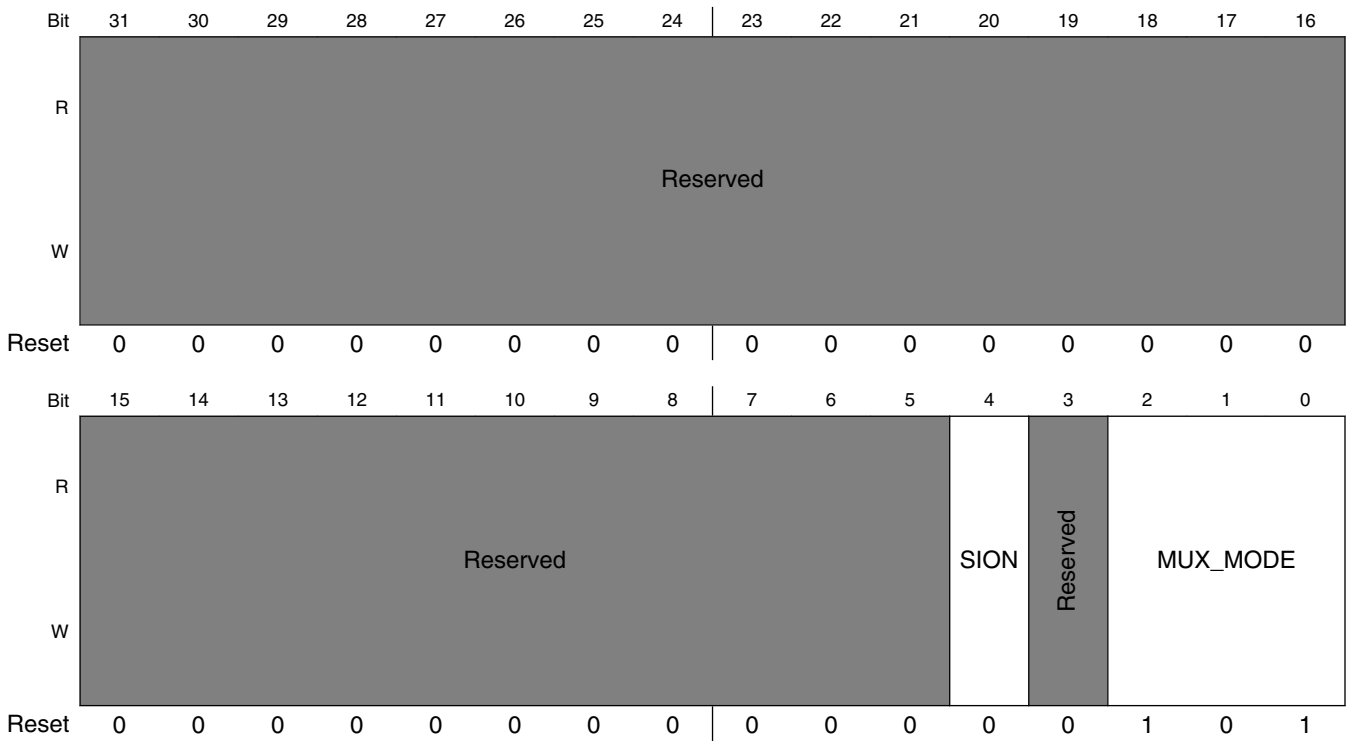
IOMUXC_SW_MUX_CTL_PAD_SD2_RESET_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_RESET_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_RESET_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_RESET_B 000 **ALT0_USDHC2_RESET_B** — Select mux mode: ALT0 mux port: RESET_B of instance: USDHC2 101 **ALT5_GPIO2_IO19** — Select mux mode: ALT5 mux port: IO19 of instance: GPIO2

8.2.5.56 SW_MUX_CTL_PAD_SD2_WP SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SD2_WP)

SW_MUX_CTL Register

Address: 3033_0000h base + F0h offset = 3033_00F0h



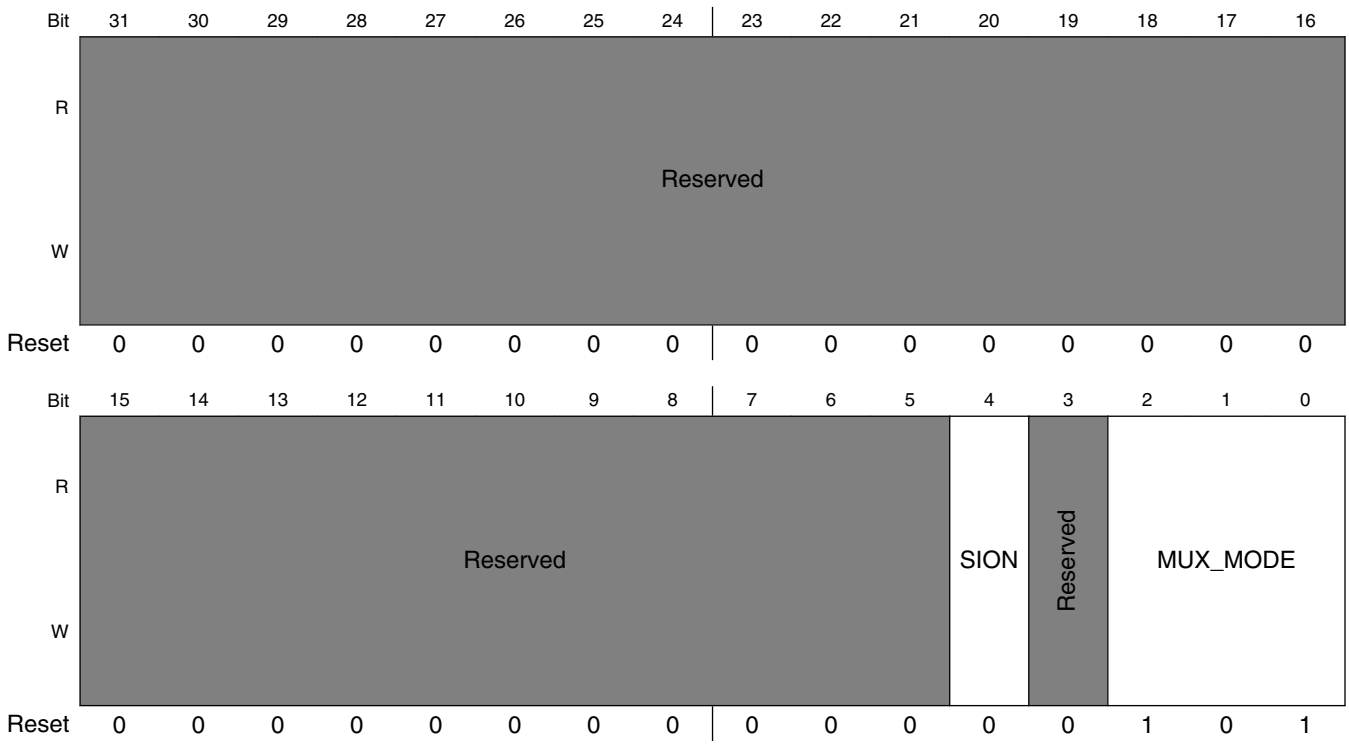
IOMUXC_SW_MUX_CTL_PAD_SD2_WP field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SD2_WP is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SD2_WP
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: SD2_WP 000 **ALT0_USDHC2_WP** — Select mux mode: ALT0 mux port: WP of instance: USDHC2 101 **ALT5_GPIO2_IO20** — Select mux mode: ALT5 mux port: IO20 of instance: GPIO2

8.2.5.57 SW_MUX_CTL_PAD_NAND_ALE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_ALE)

SW_MUX_CTL Register

Address: 3033_0000h base + F4h offset = 3033_00F4h



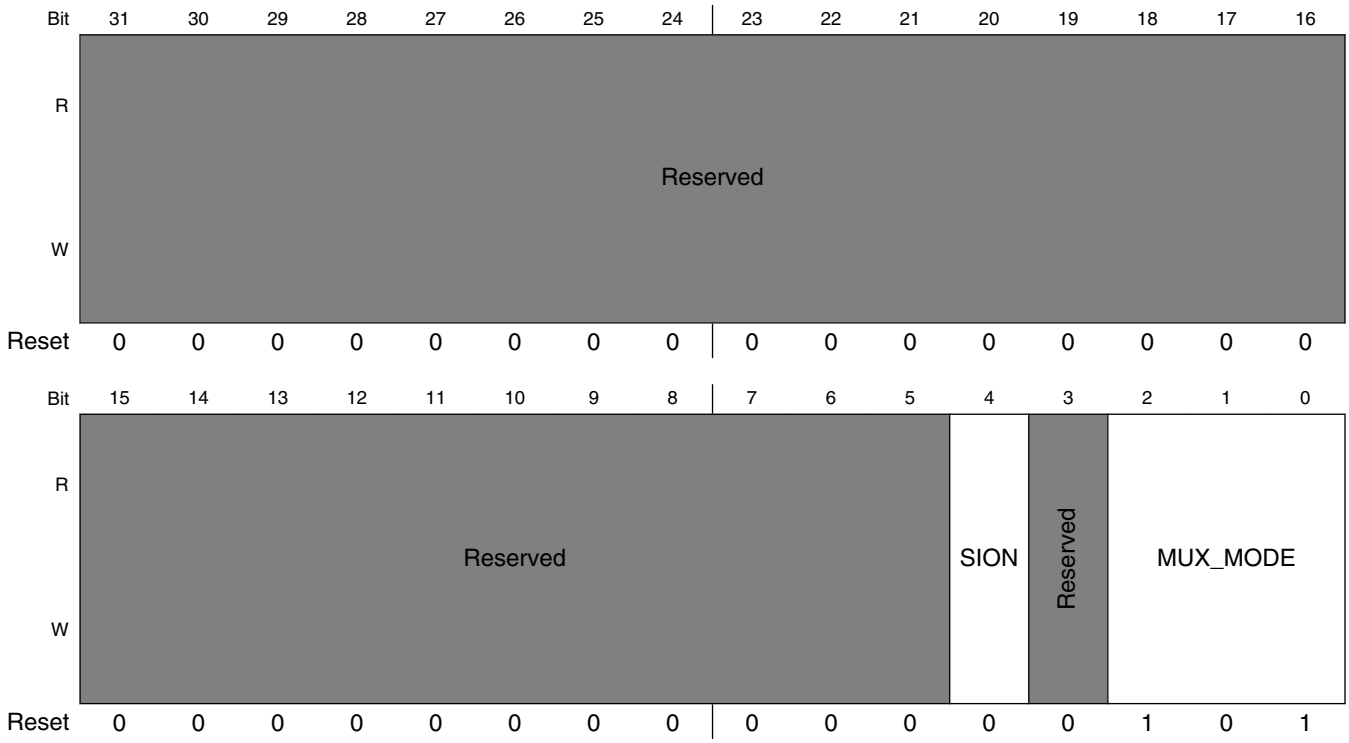
IOMUXC_SW_MUX_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_ALE is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_ALE
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_ALE 000 **ALT0_RAWNAND_ALE** — Select mux mode: ALT0 mux port: ALE of instance: RAWNAND 001 **ALT1_QSPI_A_SCLK** — Select mux mode: ALT1 mux port: A_SCLK of instance: QSPI 101 **ALT5_GPIO3_IO00** — Select mux mode: ALT5 mux port: IO00 of instance: GPIO3

8.2.5.58 SW_MUX_CTL_PAD_NAND_CE0_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B)

SW_MUX_CTL Register

Address: 3033_0000h base + F8h offset = 3033_00F8h



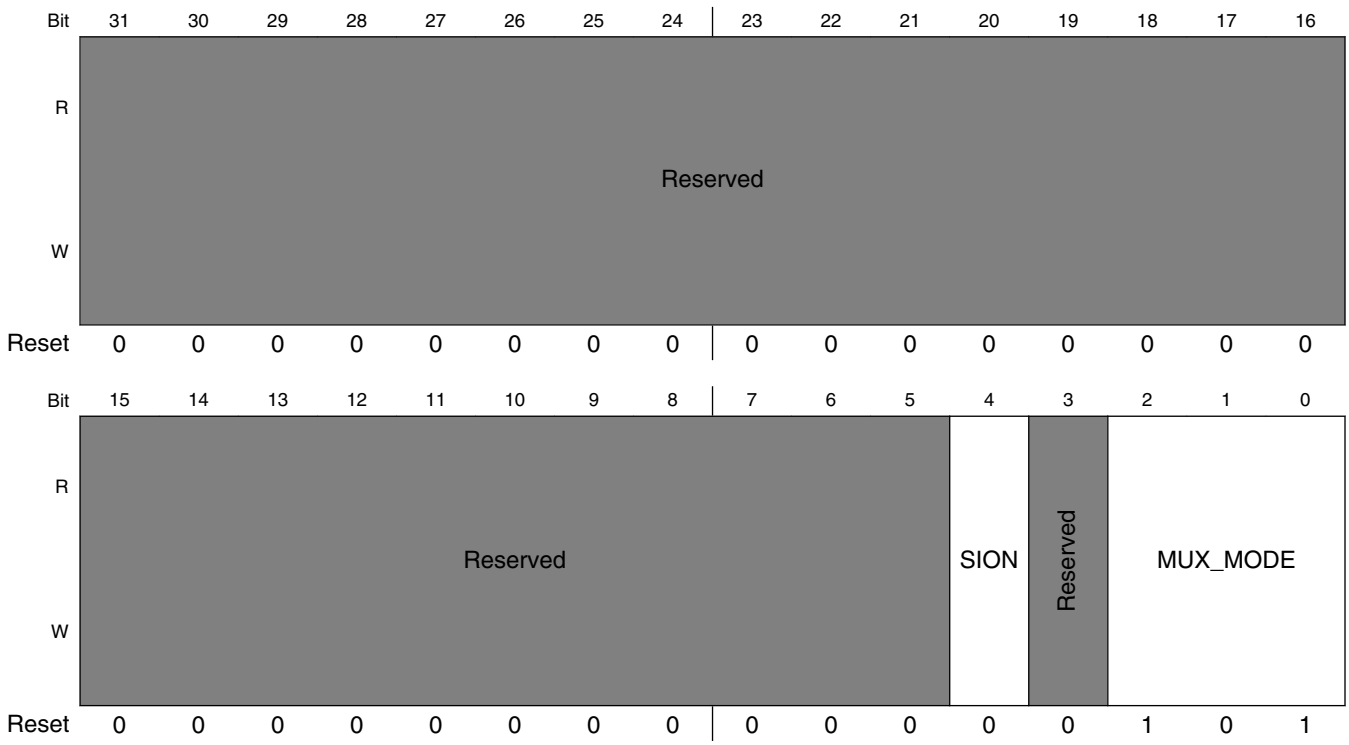
IOMUXC_SW_MUX_CTL_PAD_NAND_CE0_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_CE0_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_CE0_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_CE0_B 000 **ALT0_RAWNAND_CE0_B** — Select mux mode: ALT0 mux port: CE0_B of instance: RAWNAND 001 **ALT1_QSPI_A_SS0_B** — Select mux mode: ALT1 mux port: A_SS0_B of instance: QSPI 101 **ALT5_GPIO3_IO01** — Select mux mode: ALT5 mux port: IO01 of instance: GPIO3

8.2.5.59 SW_MUX_CTL_PAD_NAND_CE1_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B)

SW_MUX_CTL Register

Address: 3033_0000h base + FCh offset = 3033_00FCh



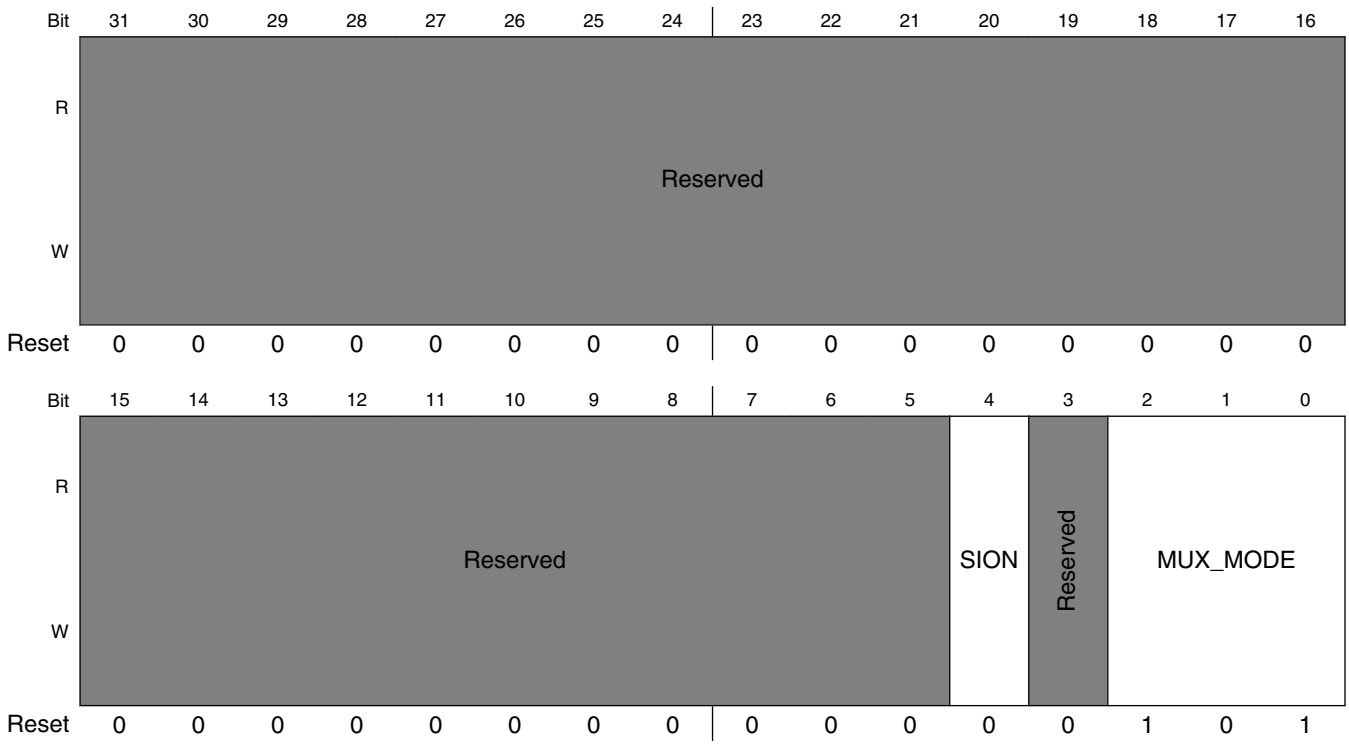
IOMUXC_SW_MUX_CTL_PAD_NAND_CE1_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_CE1_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_CE1_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_CE1_B 000 **ALT0_RAWNAND_CE1_B** — Select mux mode: ALT0 mux port: CE1_B of instance: RAWNAND 001 **ALT1_QSPI_A_SS1_B** — Select mux mode: ALT1 mux port: A_SS1_B of instance: QSPI 101 **ALT5_GPIO3_IO02** — Select mux mode: ALT5 mux port: IO02 of instance: GPIO3

8.2.5.60 SW_MUX_CTL_PAD_NAND_CE2_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 100h offset = 3033_0100h



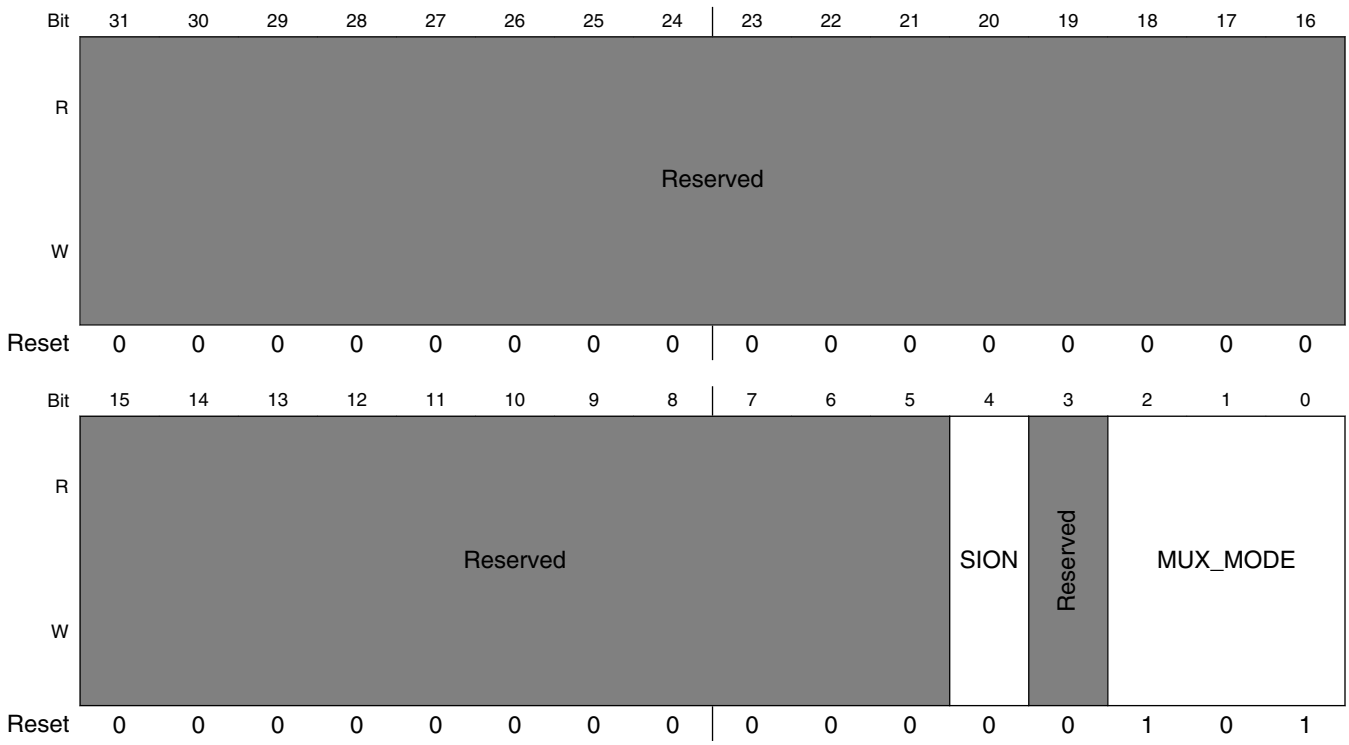
IOMUXC_SW_MUX_CTL_PAD_NAND_CE2_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_CE2_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_CE2_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_CE2_B 000 **ALT0_RAWNAND_CE2_B** — Select mux mode: ALT0 mux port: CE2_B of instance: RAWNAND 001 **ALT1_QSPI_B_SS0_B** — Select mux mode: ALT1 mux port: B_SS0_B of instance: QSPI 101 **ALT5_GPIO3_IO03** — Select mux mode: ALT5 mux port: IO03 of instance: GPIO3

8.2.5.61 SW_MUX_CTL_PAD_NAND_CE3_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 104h offset = 3033_0104h



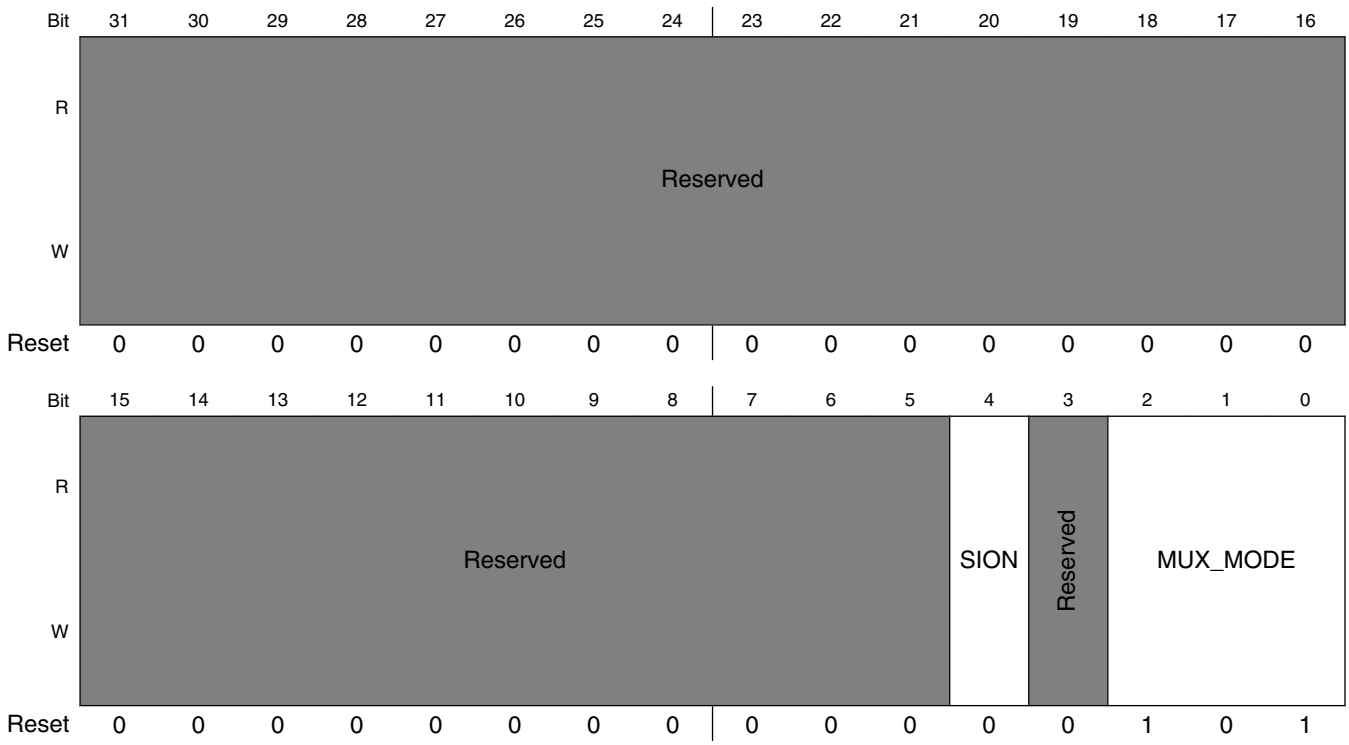
IOMUXC_SW_MUX_CTL_PAD_NAND_CE3_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_CE3_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_CE3_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_CE3_B 000 **ALT0_RAWNAND_CE3_B** — Select mux mode: ALT0 mux port: CE3_B of instance: RAWNAND 001 **ALT1_QSPI_B_SS1_B** — Select mux mode: ALT1 mux port: B_SS1_B of instance: QSPI 101 **ALT5_GPIO3_IO04** — Select mux mode: ALT5 mux port: IO04 of instance: GPIO3

8.2.5.62 SW_MUX_CTL_PAD_NAND_CLE SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_CLE)

SW_MUX_CTL Register

Address: 3033_0000h base + 108h offset = 3033_0108h



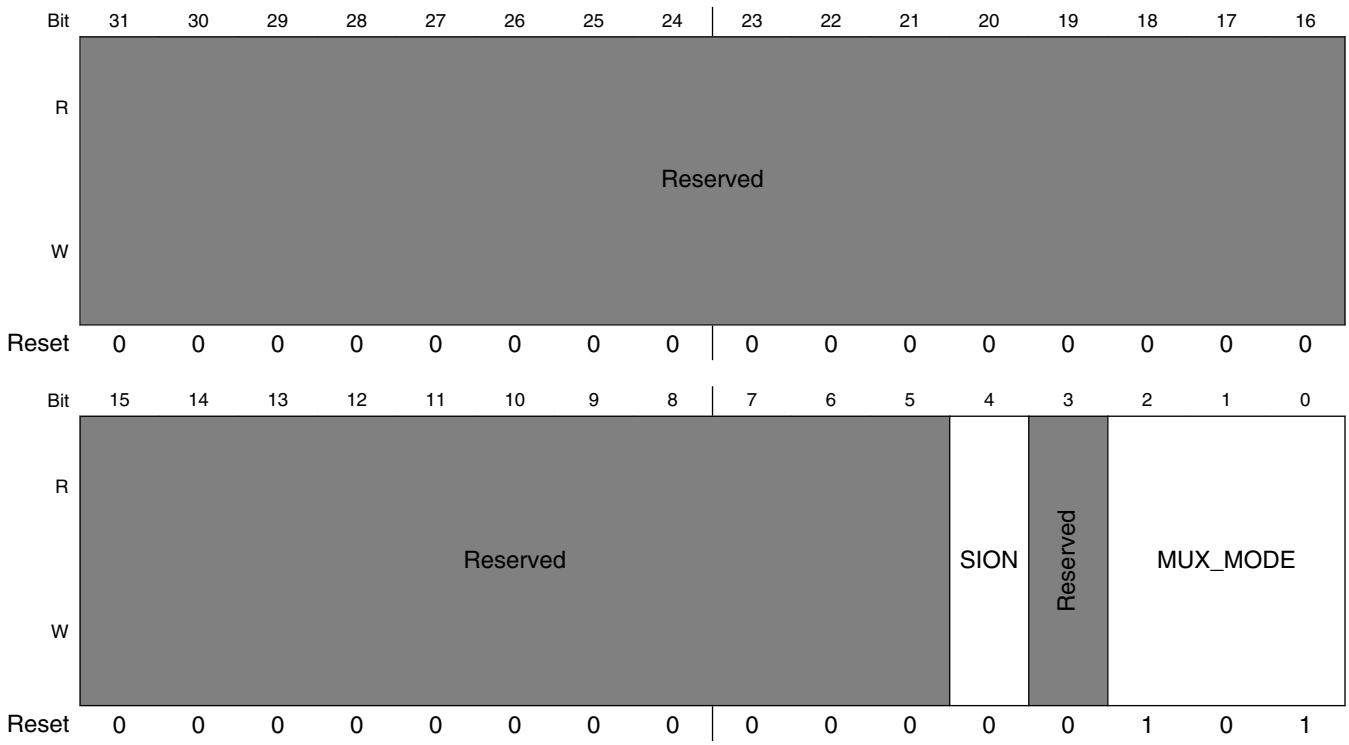
IOMUXC_SW_MUX_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_CLE is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_CLE
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_CLE 000 **ALT0_RAWNAND_CLE** — Select mux mode: ALT0 mux port: CLE of instance: RAWNAND 001 **ALT1_QSPI_B_SCLK** — Select mux mode: ALT1 mux port: B_SCLK of instance: QSPI 101 **ALT5_GPIO3_IO05** — Select mux mode: ALT5 mux port: IO05 of instance: GPIO3

8.2.5.63 SW_MUX_CTL_PAD_NAND_DATA00 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00)

SW_MUX_CTL Register

Address: 3033_0000h base + 10Ch offset = 3033_010Ch



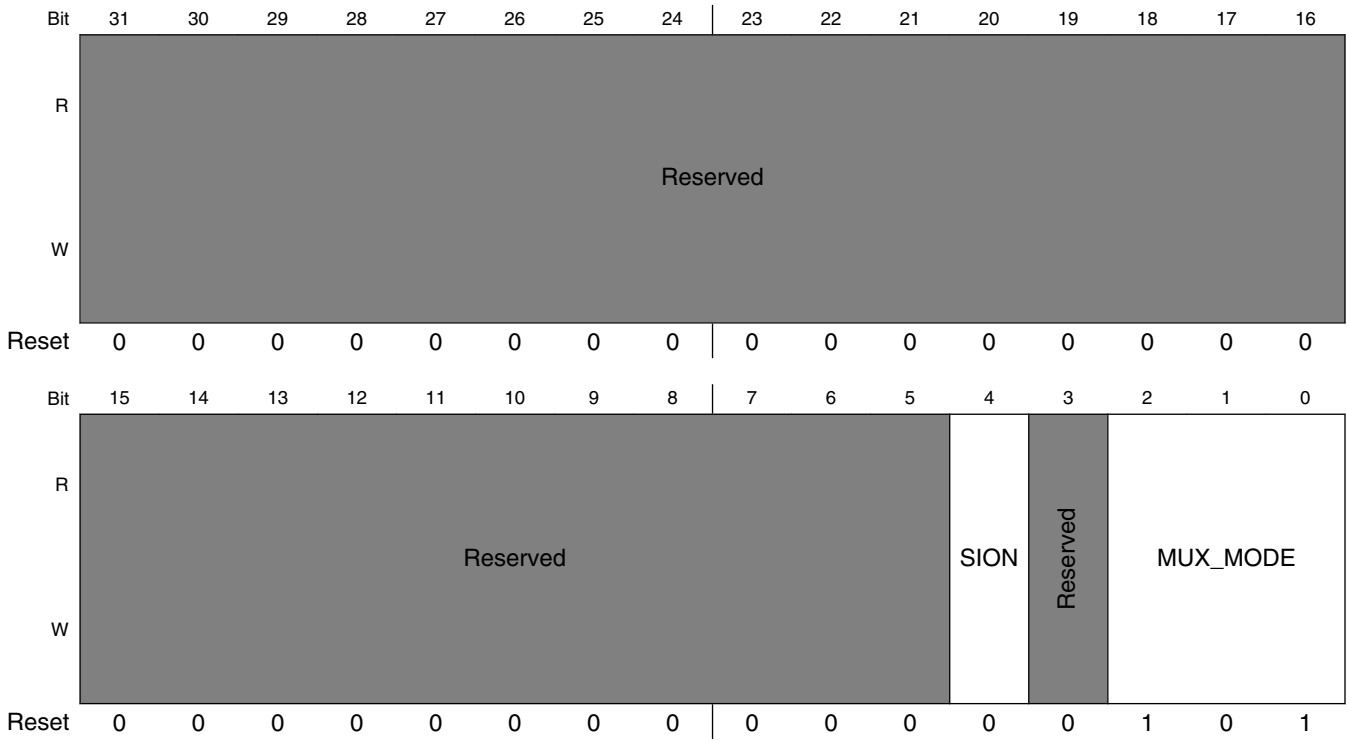
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA00 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA00
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA00 000 **ALT0_RAWNAND_DATA00** — Select mux mode: ALT0 mux port: DATA00 of instance: RAWNAND 001 **ALT1_QSPI_A_DATA0** — Select mux mode: ALT1 mux port: A_DATA0 of instance: QSPI 101 **ALT5_GPIO3_IO06** — Select mux mode: ALT5 mux port: IO06 of instance: GPIO3

8.2.5.64 SW_MUX_CTL_PAD_NAND_DATA01 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01)

SW_MUX_CTL Register

Address: 3033_0000h base + 110h offset = 3033_0110h



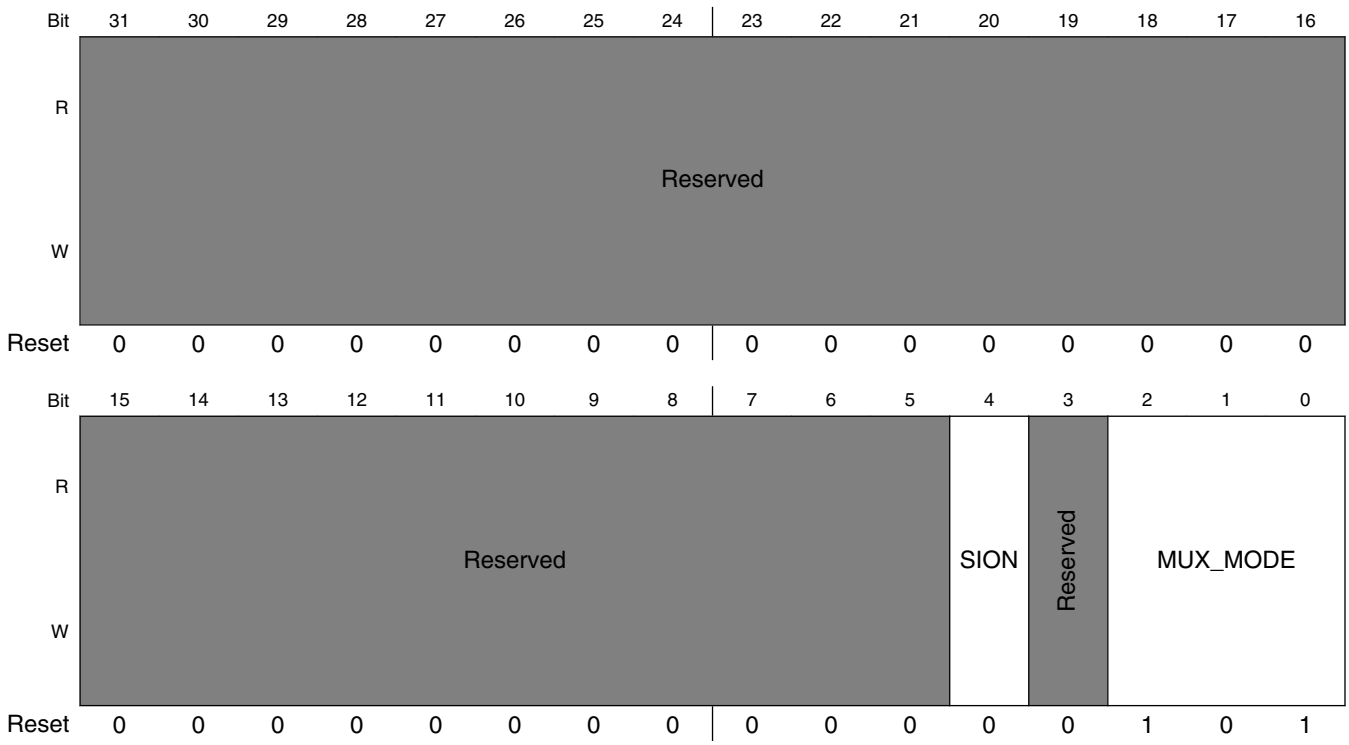
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA01 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA01
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA01 000 **ALT0_RAWNAND_DATA01** — Select mux mode: ALT0 mux port: DATA01 of instance: RAWNAND 001 **ALT1_QSPI_A_DATA1** — Select mux mode: ALT1 mux port: A_DATA1 of instance: QSPI 101 **ALT5_GPIO3_IO07** — Select mux mode: ALT5 mux port: IO07 of instance: GPIO3

8.2.5.65 SW_MUX_CTL_PAD_NAND_DATA02 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02)

SW_MUX_CTL Register

Address: 3033_0000h base + 114h offset = 3033_0114h



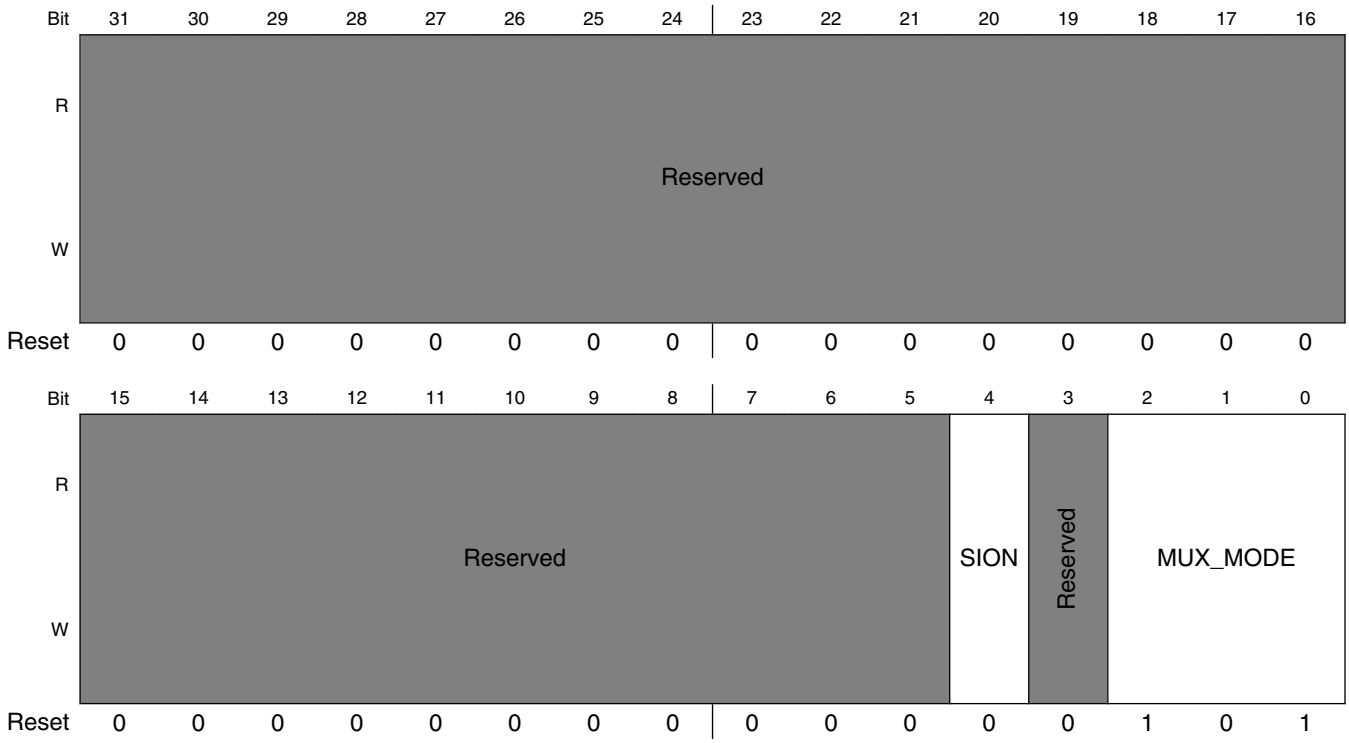
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA02 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA02
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA02 000 **ALT0_RAWNAND_DATA02** — Select mux mode: ALT0 mux port: DATA02 of instance: RAWNAND 001 **ALT1_QSPI_A_DATA2** — Select mux mode: ALT1 mux port: A_DATA2 of instance: QSPI 101 **ALT5_GPIO3_IO08** — Select mux mode: ALT5 mux port: IO08 of instance: GPIO3

8.2.5.66 SW_MUX_CTL_PAD_NAND_DATA03 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03)

SW_MUX_CTL Register

Address: 3033_0000h base + 118h offset = 3033_0118h



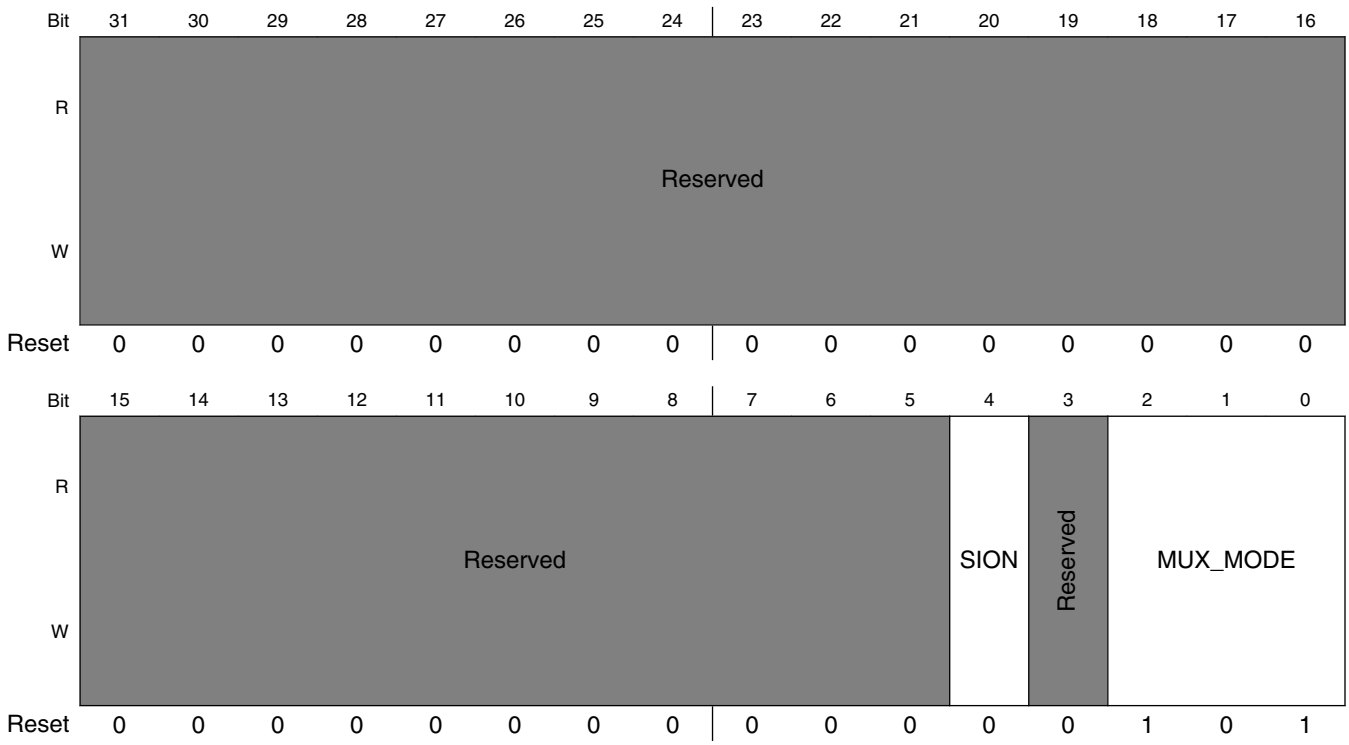
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA03 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA03
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA03 000 **ALT0_RAWNAND_DATA03** — Select mux mode: ALT0 mux port: DATA03 of instance: RAWNAND 001 **ALT1_QSPI_A_DATA3** — Select mux mode: ALT1 mux port: A_DATA3 of instance: QSPI 101 **ALT5_GPIO3_IO09** — Select mux mode: ALT5 mux port: IO09 of instance: GPIO3

8.2.5.67 SW_MUX_CTL_PAD_NAND_DATA04 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04)

SW_MUX_CTL Register

Address: 3033_0000h base + 11Ch offset = 3033_011Ch



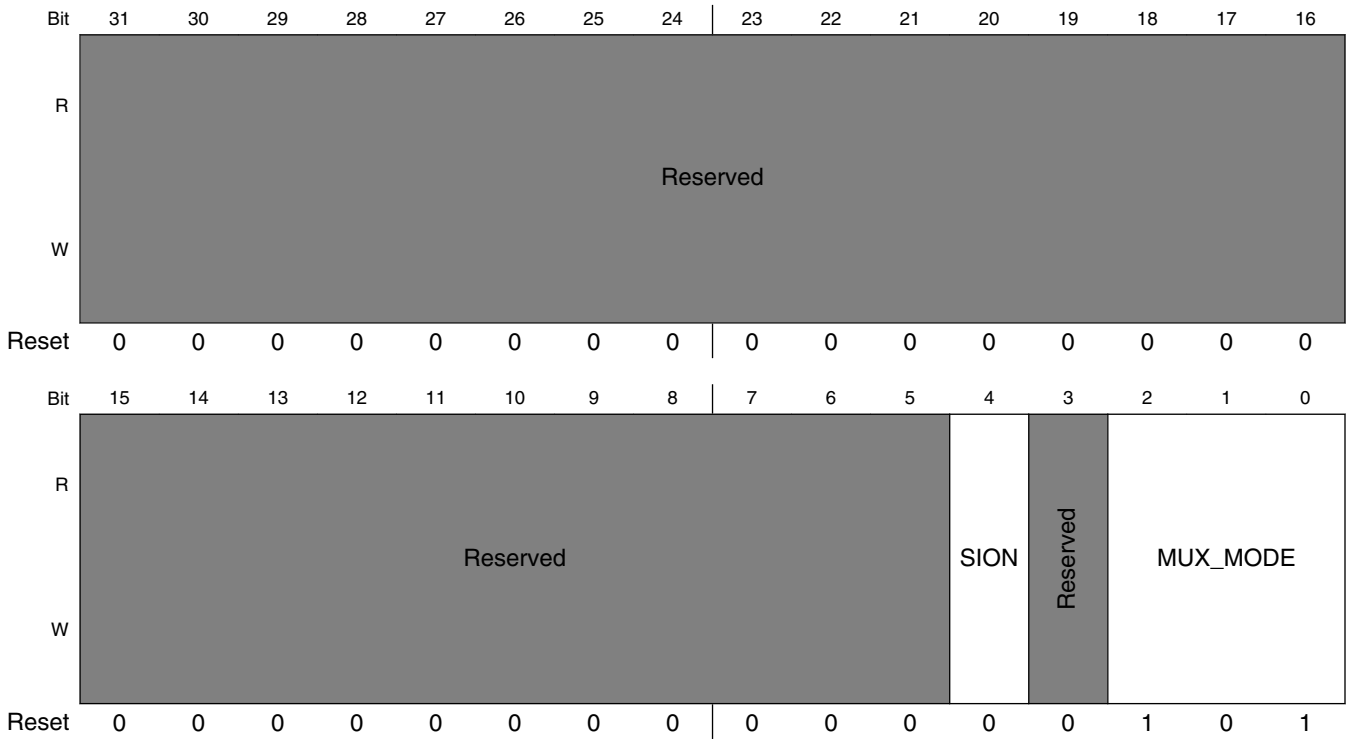
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA04 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA04
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA04 000 **ALT0_RAWNAND_DATA04** — Select mux mode: ALT0 mux port: DATA04 of instance: RAWNAND 001 **ALT1_QSPI_B_DATA0** — Select mux mode: ALT1 mux port: B_DATA0 of instance: QSPI 101 **ALT5_GPIO3_IO10** — Select mux mode: ALT5 mux port: IO10 of instance: GPIO3

8.2.5.68 SW_MUX_CTL_PAD_NAND_DATA05 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05)

SW_MUX_CTL Register

Address: 3033_0000h base + 120h offset = 3033_0120h



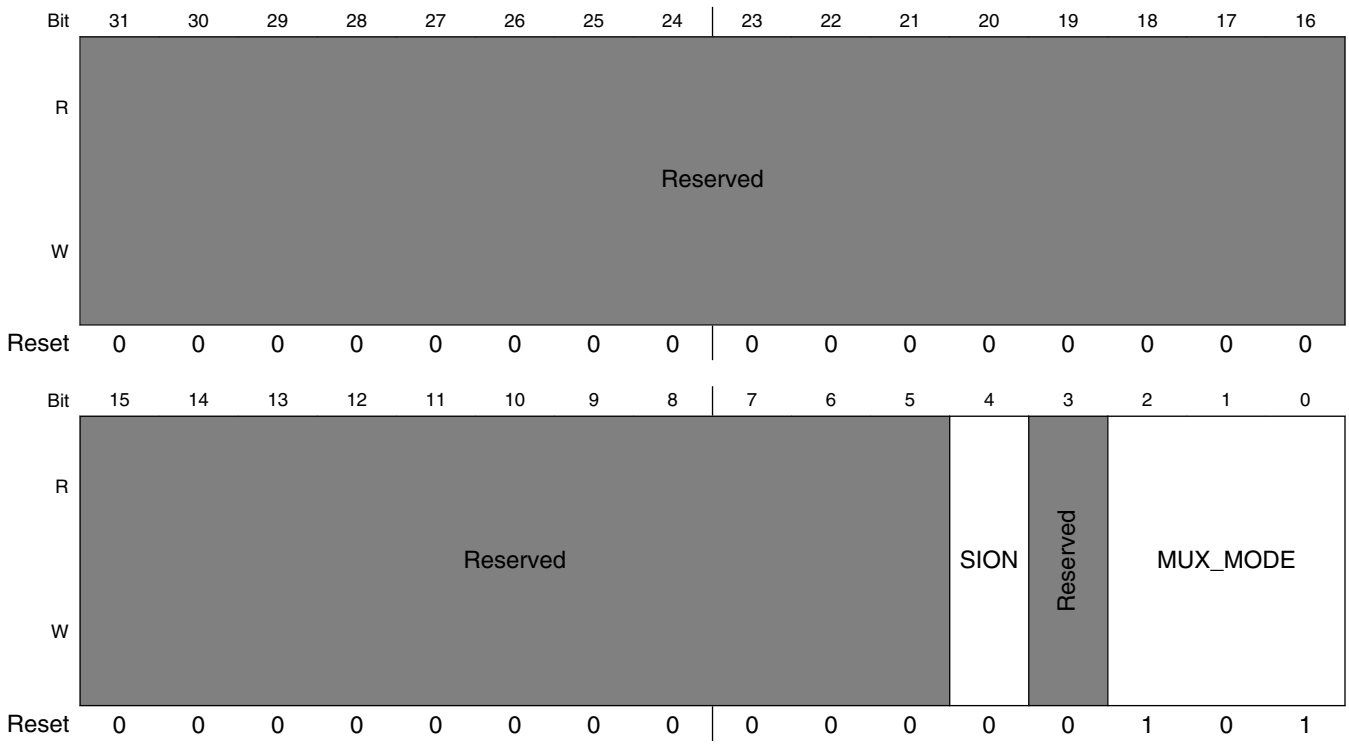
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA05 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA05
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA05 000 **ALT0_RAWNAND_DATA05** — Select mux mode: ALT0 mux port: DATA05 of instance: RAWNAND 001 **ALT1_QSPI_B_DATA1** — Select mux mode: ALT1 mux port: B_DATA1 of instance: QSPI 101 **ALT5_GPIO3_IO11** — Select mux mode: ALT5 mux port: IO11 of instance: GPIO3

8.2.5.69 SW_MUX_CTL_PAD_NAND_DATA06 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06)

SW_MUX_CTL Register

Address: 3033_0000h base + 124h offset = 3033_0124h



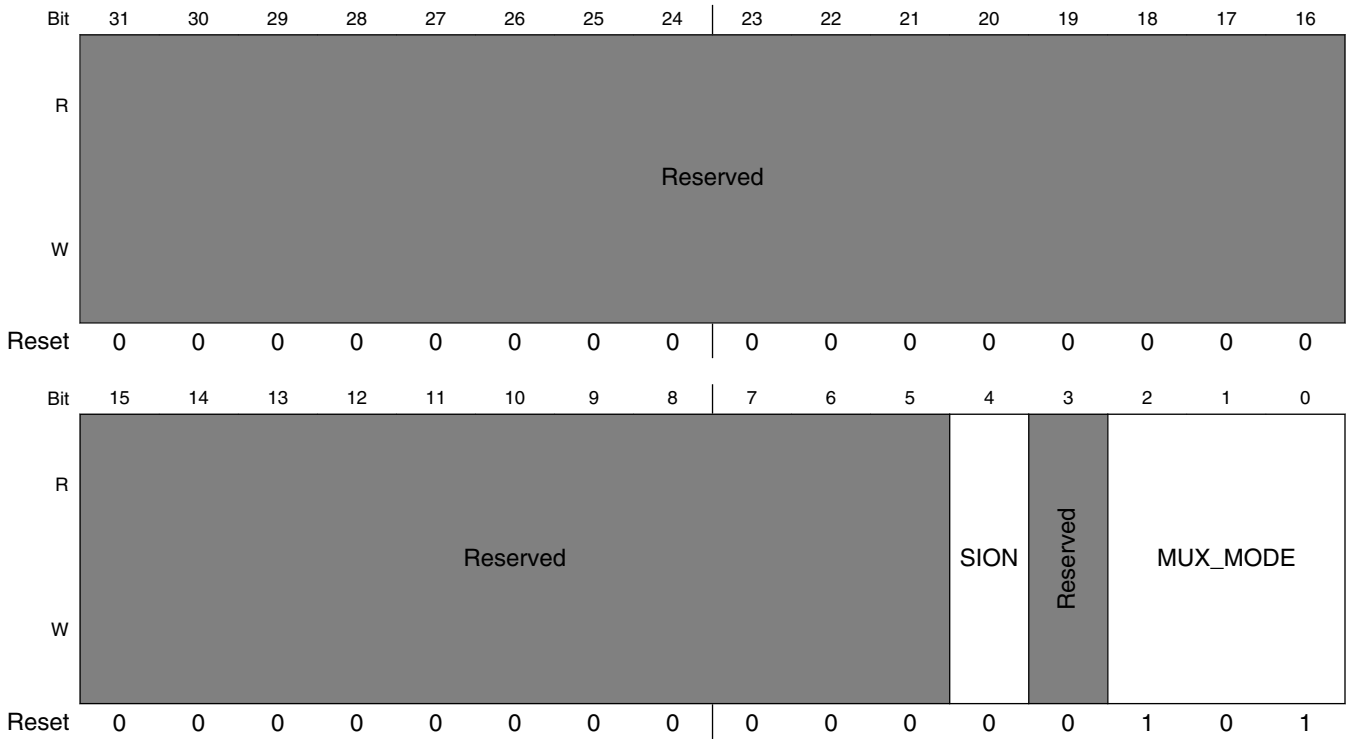
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA06 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA06
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA06 000 **ALT0_RAWNAND_DATA06** — Select mux mode: ALT0 mux port: DATA06 of instance: RAWNAND 001 **ALT1_QSPI_B_DATA2** — Select mux mode: ALT1 mux port: B_DATA2 of instance: QSPI 101 **ALT5_GPIO3_IO12** — Select mux mode: ALT5 mux port: IO12 of instance: GPIO3

8.2.5.70 SW_MUX_CTL_PAD_NAND_DATA07 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07)

SW_MUX_CTL Register

Address: 3033_0000h base + 128h offset = 3033_0128h



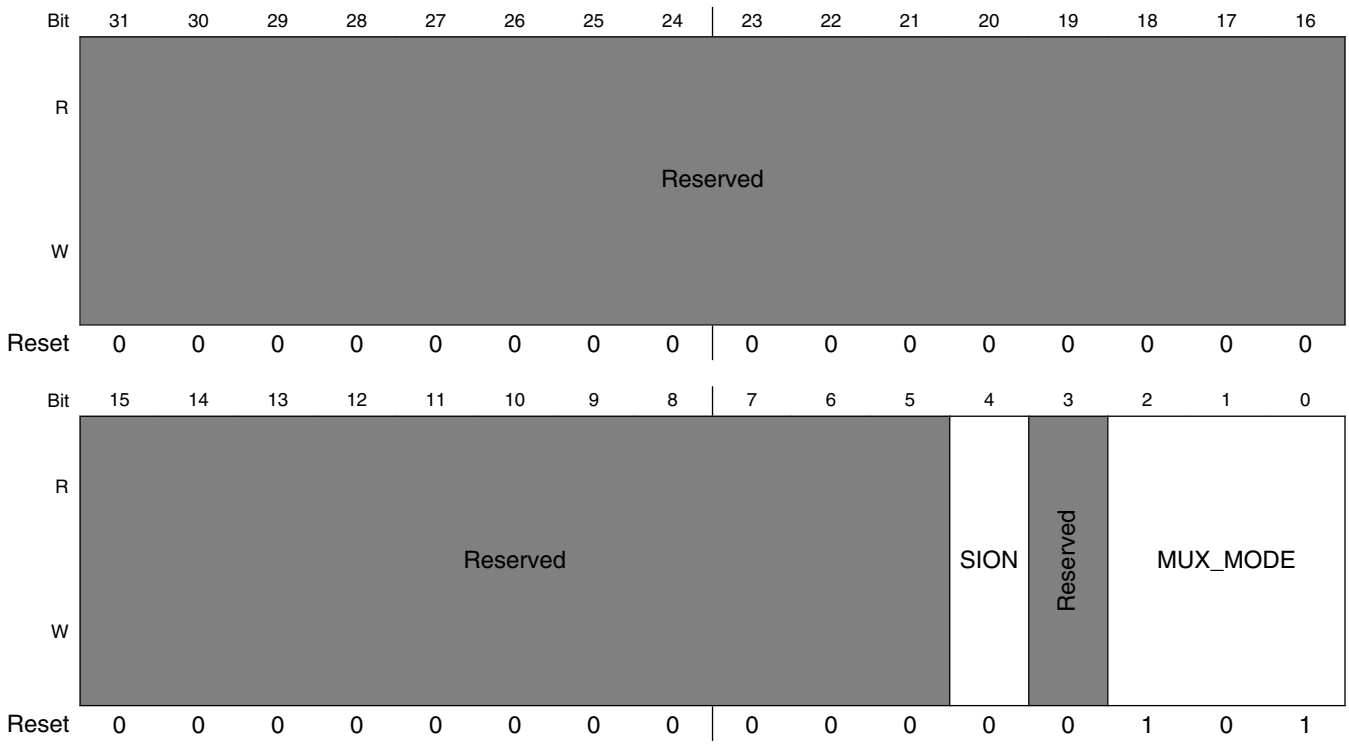
IOMUXC_SW_MUX_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DATA07 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DATA07
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DATA07 000 **ALT0_RAWNAND_DATA07** — Select mux mode: ALT0 mux port: DATA07 of instance: RAWNAND 001 **ALT1_QSPI_B_DATA3** — Select mux mode: ALT1 mux port: B_DATA3 of instance: QSPI 101 **ALT5_GPIO3_IO13** — Select mux mode: ALT5 mux port: IO13 of instance: GPIO3

8.2.5.71 SW_MUX_CTL_PAD_NAND_DQS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_DQS)

SW_MUX_CTL Register

Address: 3033_0000h base + 12Ch offset = 3033_012Ch



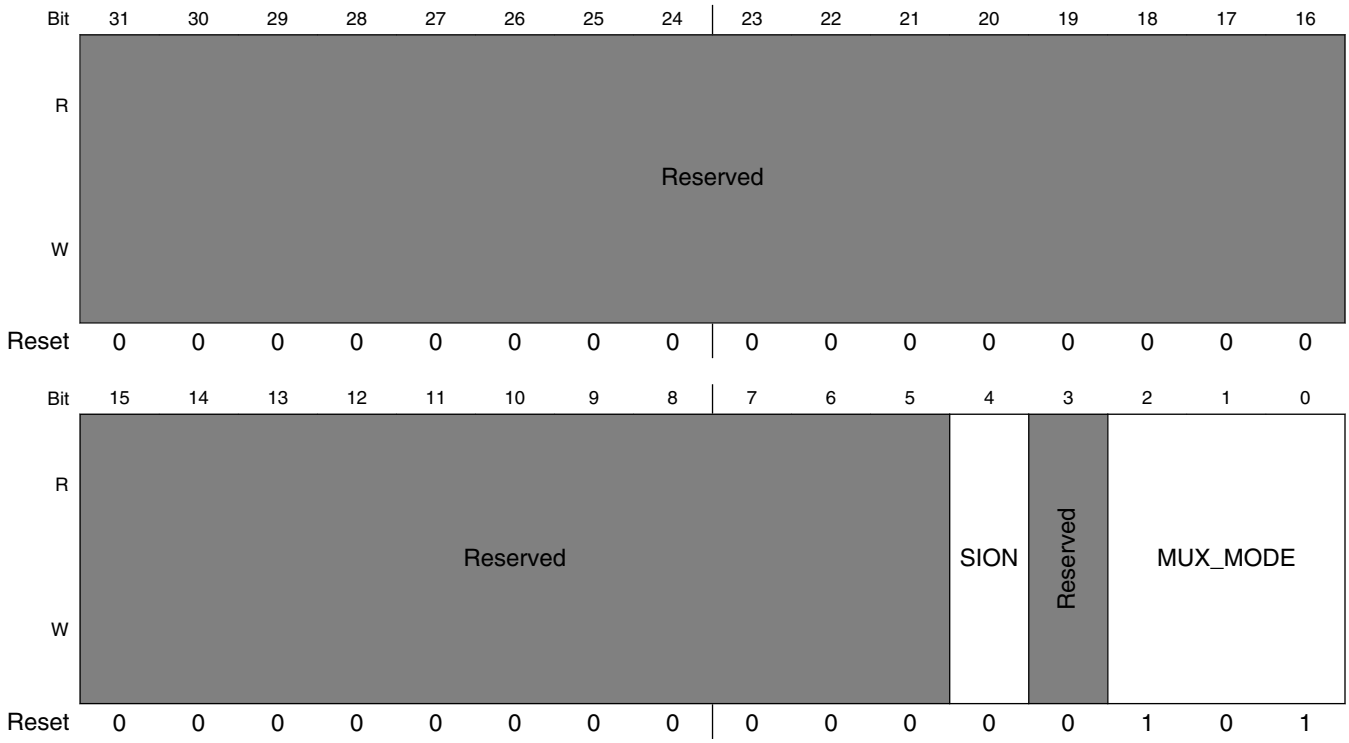
IOMUXC_SW_MUX_CTL_PAD_NAND_DQS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_DQS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_DQS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_DQS 000 **ALT0_RAWNAND_DQS** — Select mux mode: ALT0 mux port: DQS of instance: RAWNAND 001 **ALT1_QSPI_A_DQS** — Select mux mode: ALT1 mux port: A_DQS of instance: QSPI 101 **ALT5_GPIO3_IO14** — Select mux mode: ALT5 mux port: IO14 of instance: GPIO3

8.2.5.72 SW_MUX_CTL_PAD_NAND_RE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 130h offset = 3033_0130h



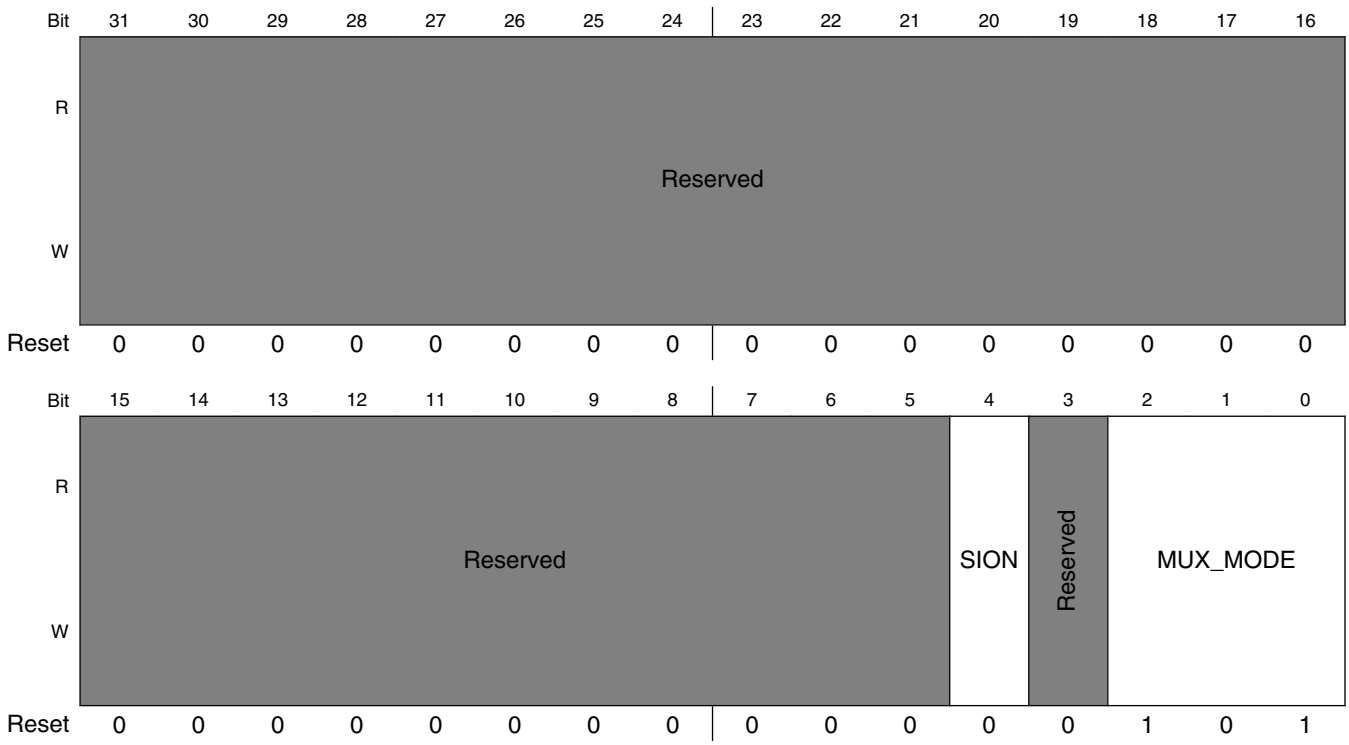
IOMUXC_SW_MUX_CTL_PAD_NAND_RE_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_RE_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_RE_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: NAND_RE_B 000 **ALT0_RAWNAND_RE_B** — Select mux mode: ALT0 mux port: RE_B of instance: RAWNAND 001 **ALT1_QSPI_B_DQS** — Select mux mode: ALT1 mux port: B_DQS of instance: QSPI 101 **ALT5_GPIO3_IO15** — Select mux mode: ALT5 mux port: IO15 of instance: GPIO3

8.2.5.73 SW_MUX_CTL_PAD_NAND_READY_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 134h offset = 3033_0134h



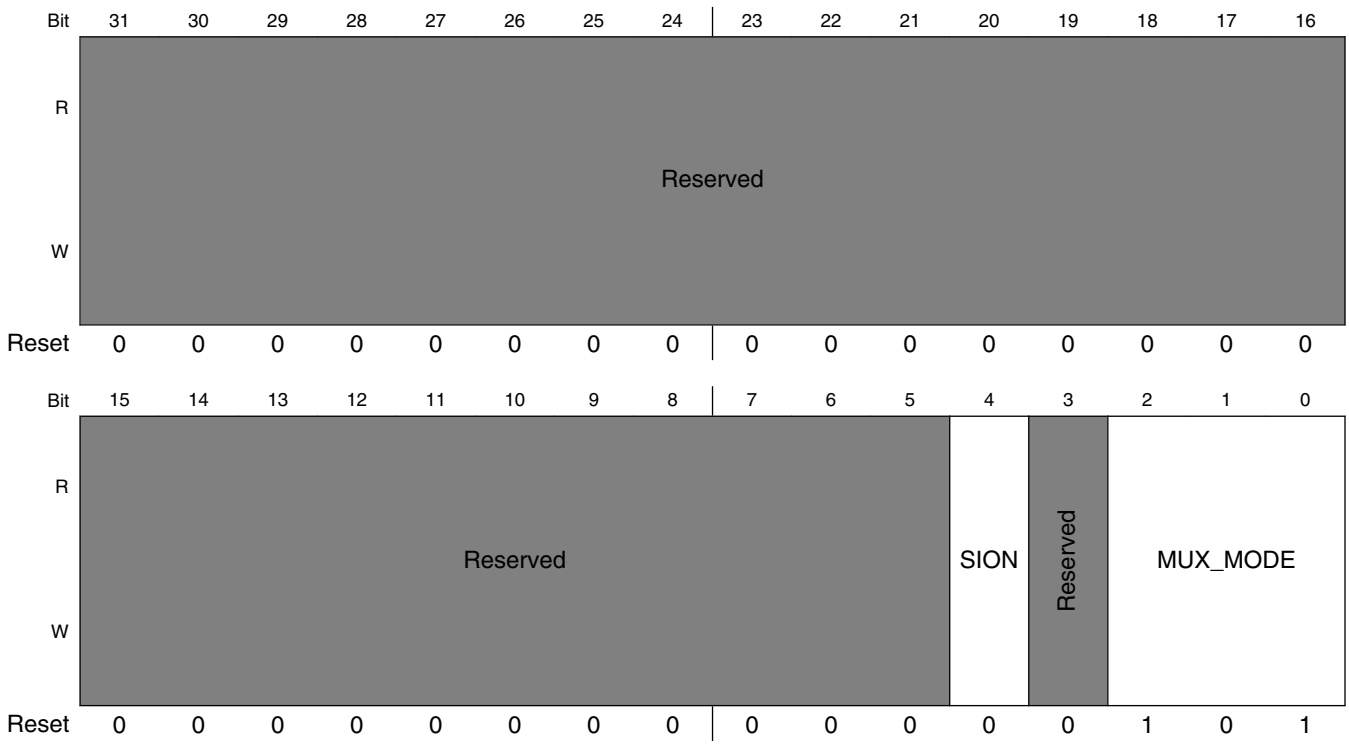
IOMUXC_SW_MUX_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_READY_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_READY_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: NAND_READY_B 000 **ALT0_RAWNAND_READY_B** — Select mux mode: ALT0 mux port: READY_B of instance: RAWNAND 101 **ALT5_GPIO3_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO3

8.2.5.74 SW_MUX_CTL_PAD_NAND_WE_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 138h offset = 3033_0138h



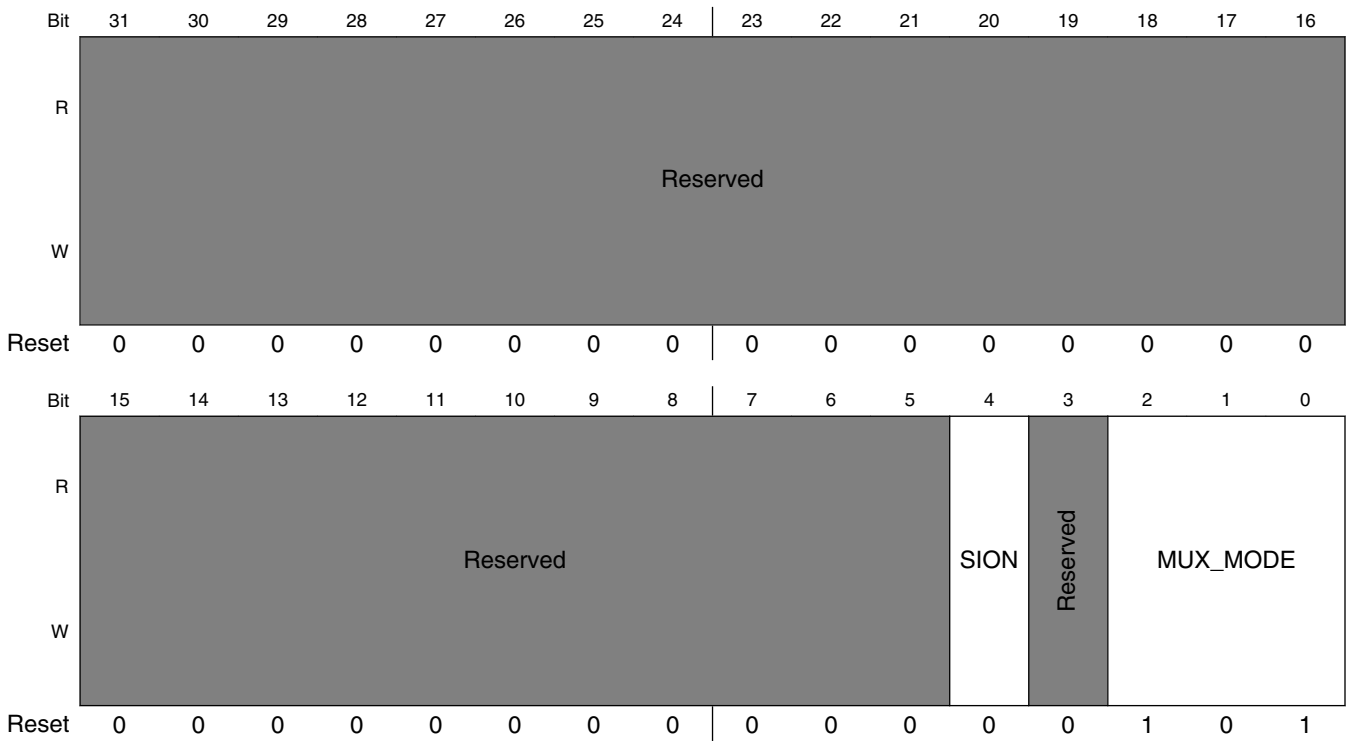
IOMUXC_SW_MUX_CTL_PAD_NAND_WE_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_WE_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_WE_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: NAND_WE_B 000 **ALT0_RAWNAND_WE_B** — Select mux mode: ALT0 mux port: WE_B of instance: RAWNAND 101 **ALT5_GPIO3_IO17** — Select mux mode: ALT5 mux port: IO17 of instance: GPIO3

8.2.5.75 SW_MUX_CTL_PAD_NAND_WP_B SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B)

SW_MUX_CTL Register

Address: 3033_0000h base + 13Ch offset = 3033_013Ch



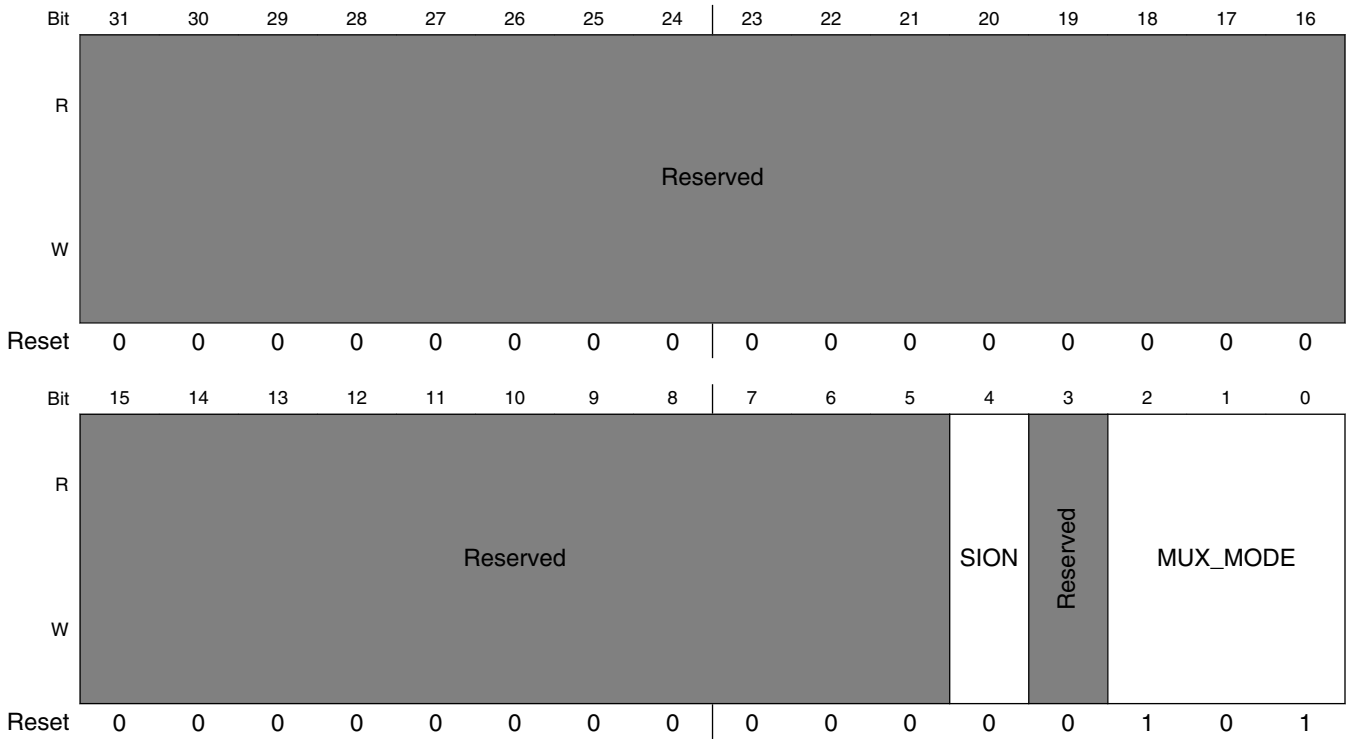
IOMUXC_SW_MUX_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad NAND_WP_B is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad NAND_WP_B
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 2 iomux modes to be used for pad: NAND_WP_B 000 **ALT0_RAWNAND_WP_B** — Select mux mode: ALT0 mux port: WP_B of instance: RAWNAND 101 **ALT5_GPIO3_IO18** — Select mux mode: ALT5 mux port: IO18 of instance: GPIO3

8.2.5.76 SW_MUX_CTL_PAD_SAI5_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 140h offset = 3033_0140h



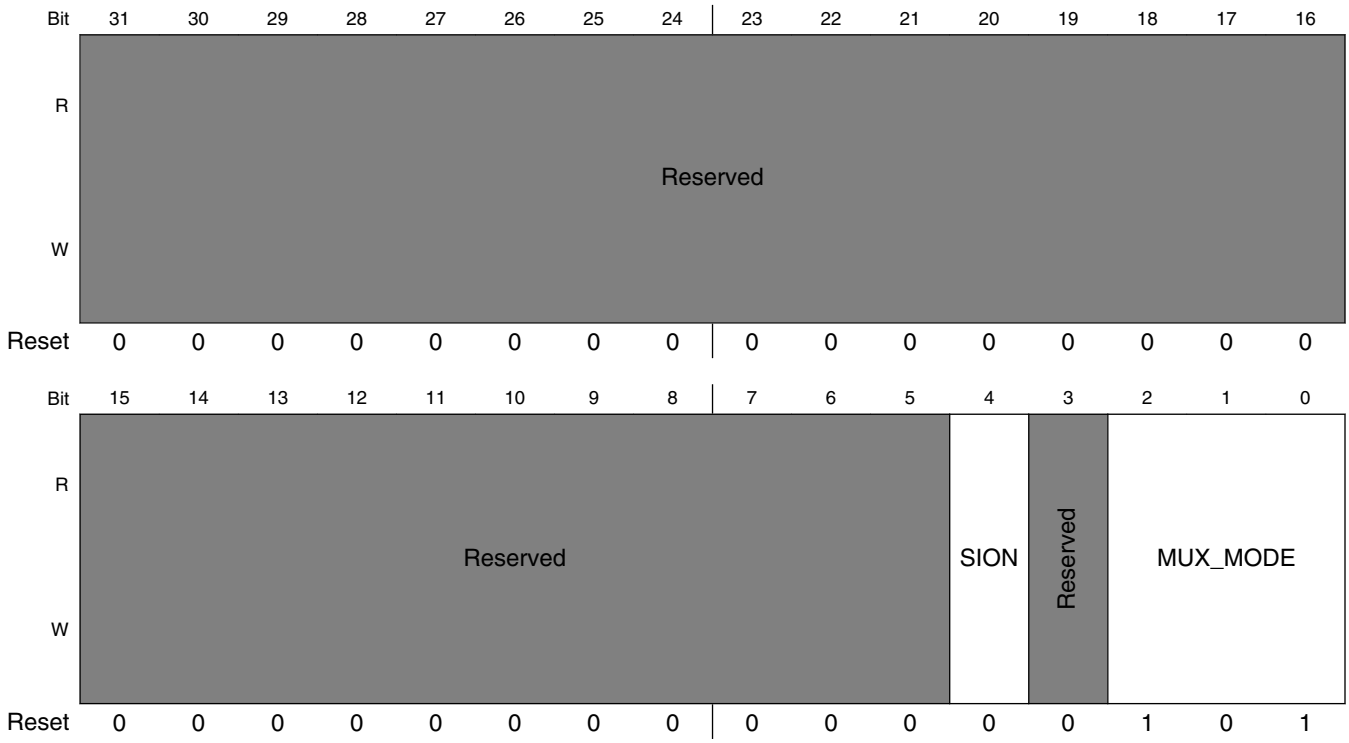
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI5_RXFS 000 **ALT0_SAI5_RX_SYNC** — Select mux mode: ALT0 mux port: RX_SYNC of instance: SAI5 001 **ALT1_SAI1_TX_DATA0** — Select mux mode: ALT1 mux port: TX_DATA0 of instance: SAI1 101 **ALT5_GPIO3_IO19** — Select mux mode: ALT5 mux port: IO19 of instance: GPIO3

8.2.5.77 SW_MUX_CTL_PAD_SAI5_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 144h offset = 3033_0144h



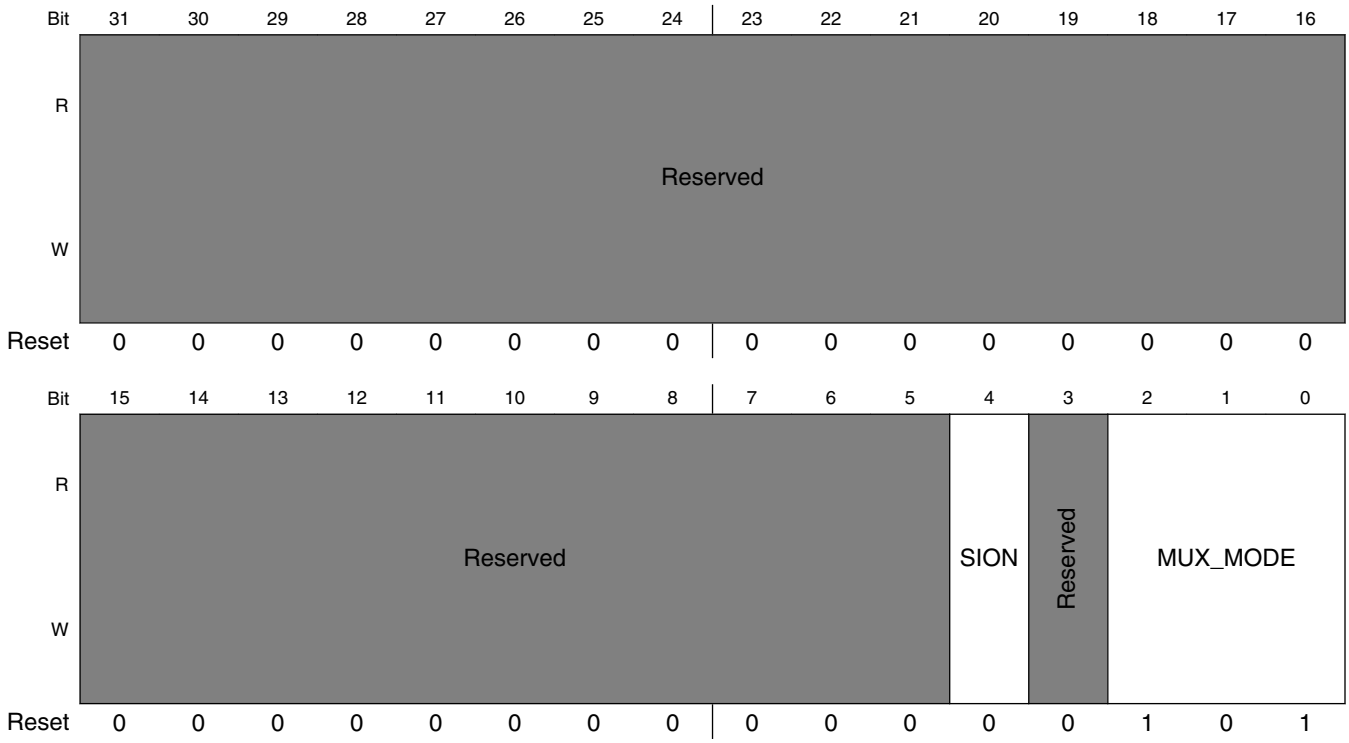
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI5_RXC 000 **ALT0_SAI5_RX_BCLK** — Select mux mode: ALT0 mux port: RX_BCLK of instance: SAI5 001 **ALT1_SAI1_TX_DATA1** — Select mux mode: ALT1 mux port: TX_DATA1 of instance: SAI1 101 **ALT5_GPIO3_IO20** — Select mux mode: ALT5 mux port: IO20 of instance: GPIO3

8.2.5.78 SW_MUX_CTL_PAD_SAI5_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 148h offset = 3033_0148h



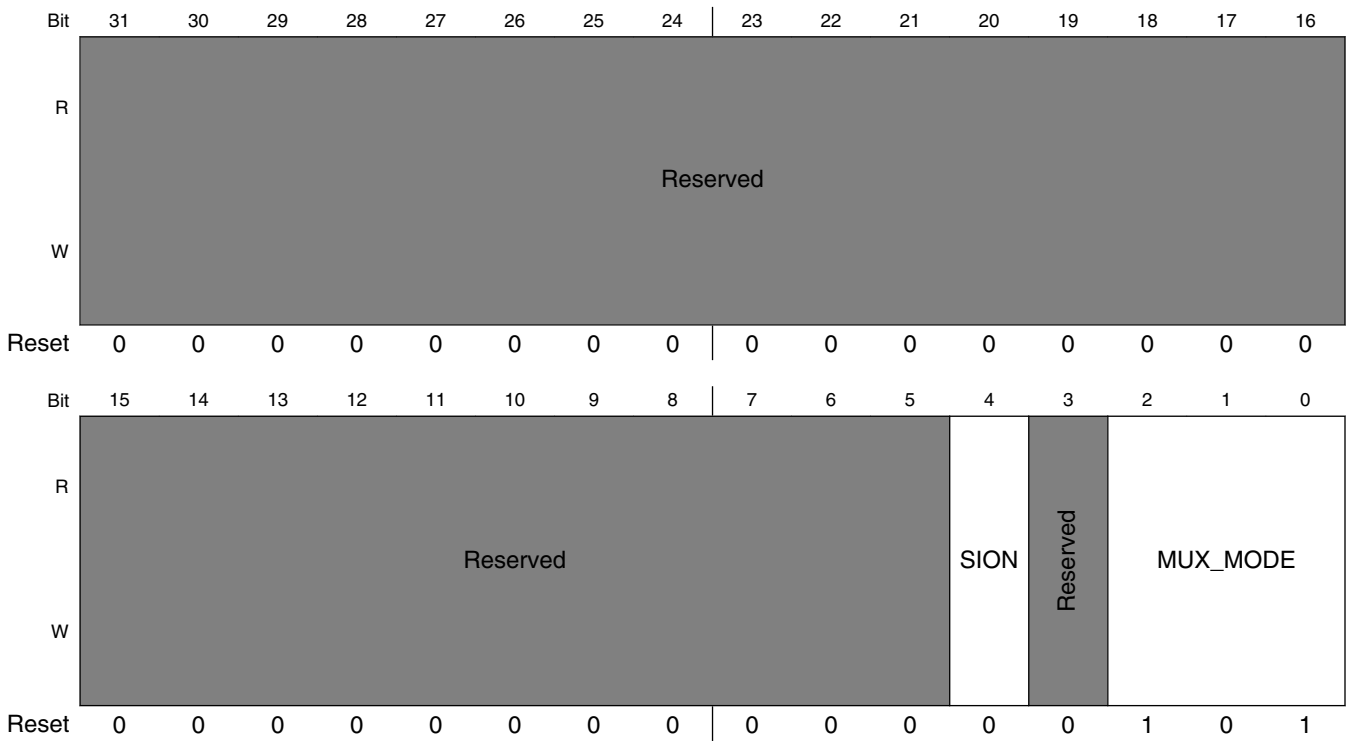
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI5_RXD0 000 **ALT0_SAI5_RX_DATA0** — Select mux mode: ALT0 mux port: RX_DATA0 of instance: SAI5 001 **ALT1_SAI1_TX_DATA2** — Select mux mode: ALT1 mux port: TX_DATA2 of instance: SAI1 101 **ALT5_GPIO3_IO21** — Select mux mode: ALT5 mux port: IO21 of instance: GPIO3

8.2.5.79 SW_MUX_CTL_PAD_SAI5_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 14Ch offset = 3033_014Ch



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXD1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXD1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI5_RXD1 000 **ALT0_SAI5_RX_DATA1** — Select mux mode: ALT0 mux port: RX_DATA1 of instance: SAI5 001 **ALT1_SAI1_TX_DATA3** — Select mux mode: ALT1 mux port: TX_DATA3 of instance: SAI1 010 **ALT2_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: TX_SYNC of instance: SAI1

Table continues on the next page...

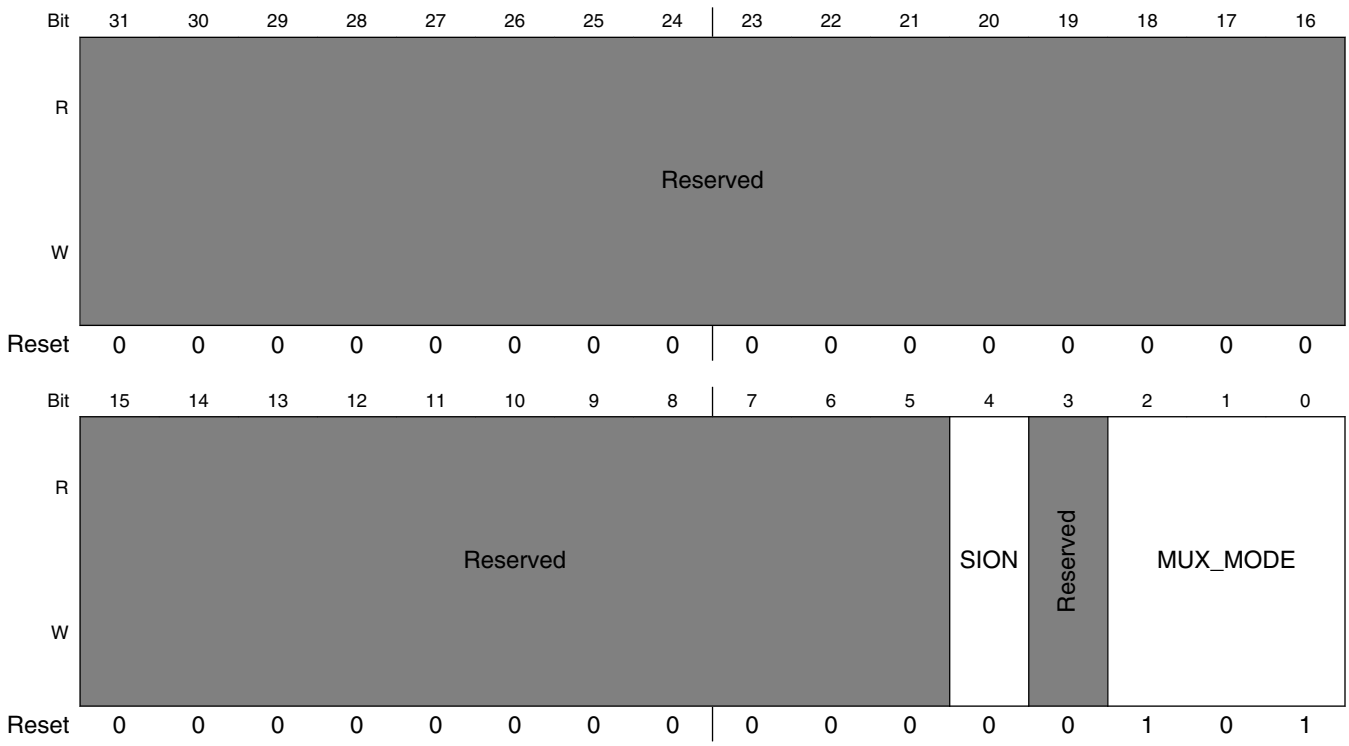
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD1 field descriptions (continued)

Field	Description
	011 **ALT3_SAI5_TX_SYNC** — Select mux mode: ALT3 mux port: TX_SYNC of instance: SAI5 101 **ALT5_GPIO3_IO22** — Select mux mode: ALT5 mux port: IO22 of instance: GPIO3

8.2.5.80 SW_MUX_CTL_PAD_SAI5_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 150h offset = 3033_0150h



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXD2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXD2
3 -	This field is reserved. Reserved

Table continues on the next page...

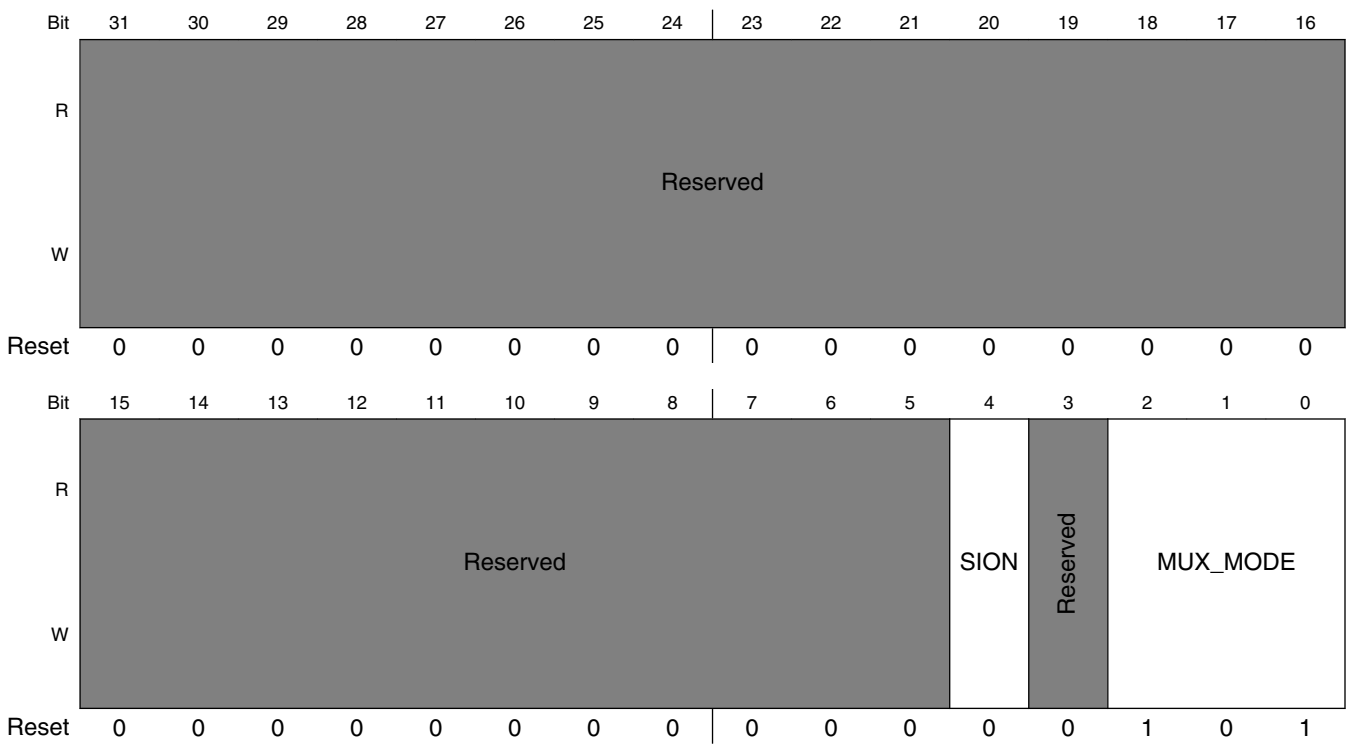
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD2 field descriptions (continued)

Field	Description
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI5_RXD2 000 **ALT0_SAI5_RX_DATA2** — Select mux mode: ALT0 mux port: RX_DATA2 of instance: SAI5 001 **ALT1_SAI1_TX_DATA4** — Select mux mode: ALT1 mux port: TX_DATA4 of instance: SAI1 010 **ALT2_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: TX_SYNC of instance: SAI1 011 **ALT3_SAI5_TX_BCLK** — Select mux mode: ALT3 mux port: TX_BCLK of instance: SAI5 101 **ALT5_GPIO3_IO23** — Select mux mode: ALT5 mux port: IO23 of instance: GPIO3

8.2.5.81 SW_MUX_CTL_PAD_SAI5_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 154h offset = 3033_0154h



IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

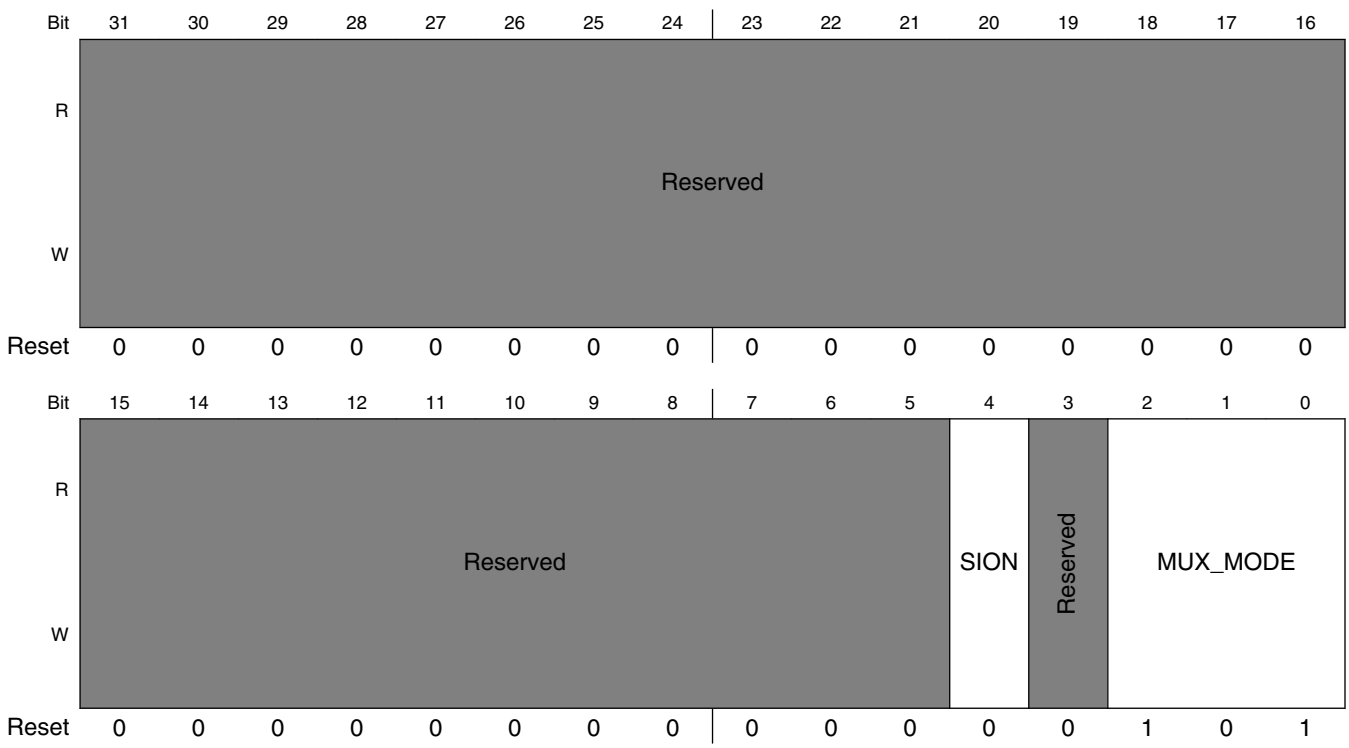
IOMUXC_SW_MUX_CTL_PAD_SAI5_RXD3 field descriptions (continued)

Field	Description
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_RXD3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_RXD3
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI5_RXD3 000 **ALT0_SAI5_RX_DATA3** — Select mux mode: ALT0 mux port: RX_DATA3 of instance: SAI5 001 **ALT1_SAI1_TX_DATA5** — Select mux mode: ALT1 mux port: TX_DATA5 of instance: SAI1 010 **ALT2_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: TX_SYNC of instance: SAI1 011 **ALT3_SAI5_TX_DATA0** — Select mux mode: ALT3 mux port: TX_DATA0 of instance: SAI5 101 **ALT5_GPIO3_IO24** — Select mux mode: ALT5 mux port: IO24 of instance: GPIO3

8.2.5.82 SW_MUX_CTL_PAD_SAI5_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 158h offset = 3033_0158h



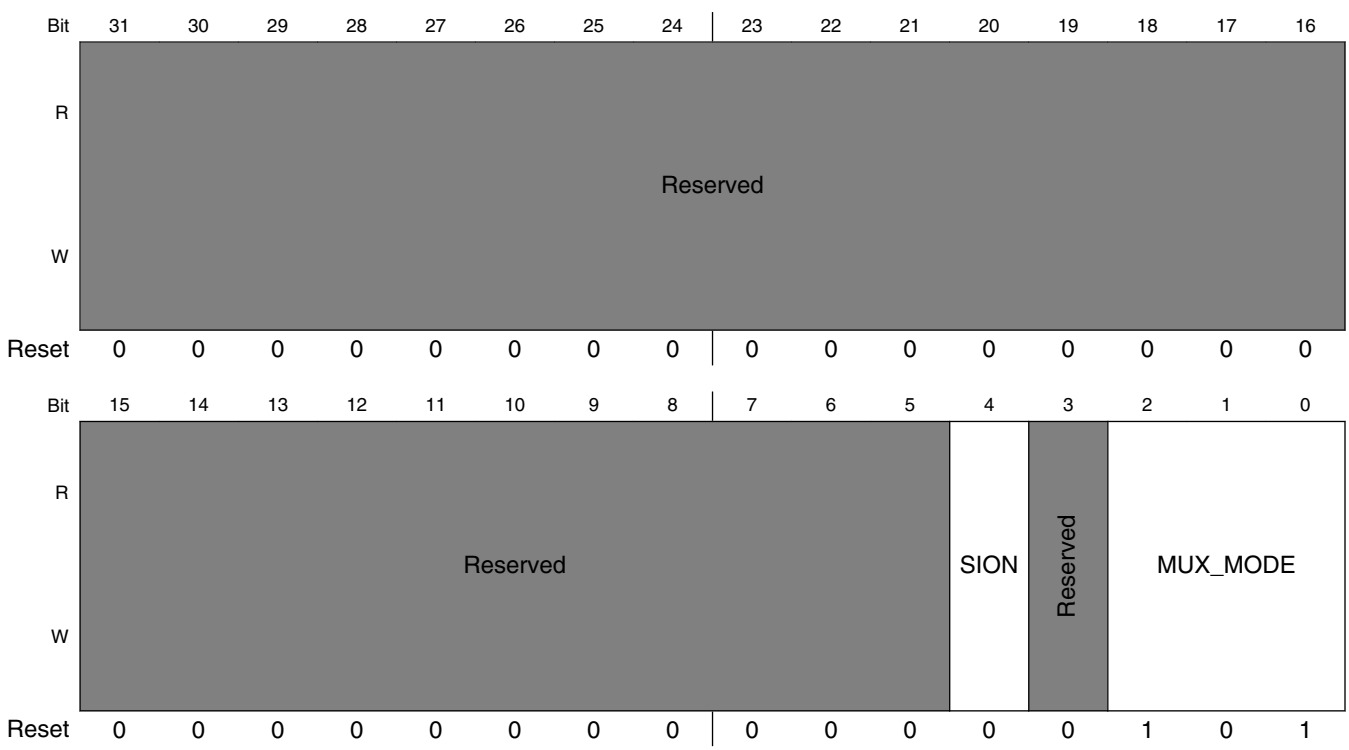
IOMUXC_SW_MUX_CTL_PAD_SAI5_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI5_MCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI5_MCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI5_MCLK 000 **ALT0_SAI5_MCLK** — Select mux mode: ALT0 mux port: MCLK of instance: SAI5 001 **ALT1_SAI1_TX_BCLK** — Select mux mode: ALT1 mux port: TX_BCLK of instance: SAI1 010 **ALT2_SAI4_MCLK** — Select mux mode: ALT2 mux port: MCLK of instance: SAI4 101 **ALT5_GPIO3_IO25** — Select mux mode: ALT5 mux port: IO25 of instance: GPIO3

8.2.5.83 SW_MUX_CTL_PAD_SAI1_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 15Ch offset = 3033_015Ch



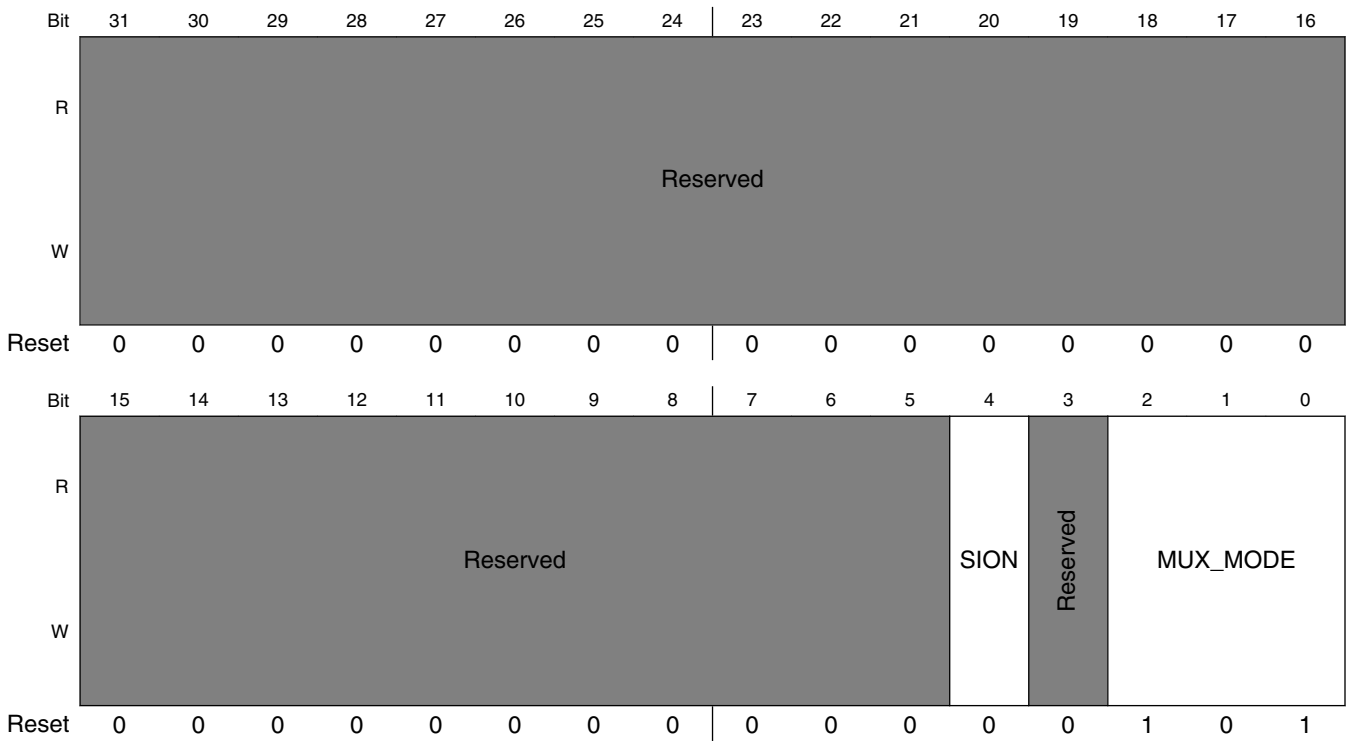
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI1_RXFS 000 **ALT0_SAI1_RX_SYNC** — Select mux mode: ALT0 mux port: RX_SYNC of instance: SAI1 001 **ALT1_SAI5_RX_SYNC** — Select mux mode: ALT1 mux port: RX_SYNC of instance: SAI5 100 **ALT4_CORESIGHT_TRACE_CLK** — Select mux mode: ALT4 mux port: TRACE_CLK of instance: CORESIGHT 101 **ALT5_GPIO4_IO00** — Select mux mode: ALT5 mux port: IO00 of instance: GPIO4

8.2.5.84 SW_MUX_CTL_PAD_SAI1_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 160h offset = 3033_0160h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI1_RXC 000 **ALT0_SAI1_RX_BCLK** — Select mux mode: ALT0 mux port: RX_BCLK of instance: SAI1 001 **ALT1_SAI5_RX_BCLK** — Select mux mode: ALT1 mux port: RX_BCLK of instance: SAI5

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_RXC field descriptions (continued)

Field	Description
	100 **ALT4_CORESIGHT_TRACE_CTL** — Select mux mode: ALT4 mux port: TRACE_CTL of instance: CORESIGHT 101 **ALT5_GPIO4_IO01** — Select mux mode: ALT5 mux port: IO01 of instance: GPIO4

8.2.5.85 SW_MUX_CTL_PAD_SAI1_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 164h offset = 3033_0164h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD0

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD0 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_RXD0 000 **ALT0_SAI1_RX_DATA0** — Select mux mode: ALT0 mux port: RX_DATA0 of instance: SAI1 001 **ALT1_SAI5_RX_DATA0** — Select mux mode: ALT1 mux port: RX_DATA0 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE0** — Select mux mode: ALT4 mux port: TRACE0 of instance: CORESIGHT 101 **ALT5_GPIO4_IO02** — Select mux mode: ALT5 mux port: IO02 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG0** — Select mux mode: ALT6 mux port: BOOT_CFG0 of instance: SRC

8.2.5.86 SW_MUX_CTL_PAD_SAI1_RXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 168h offset = 3033_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

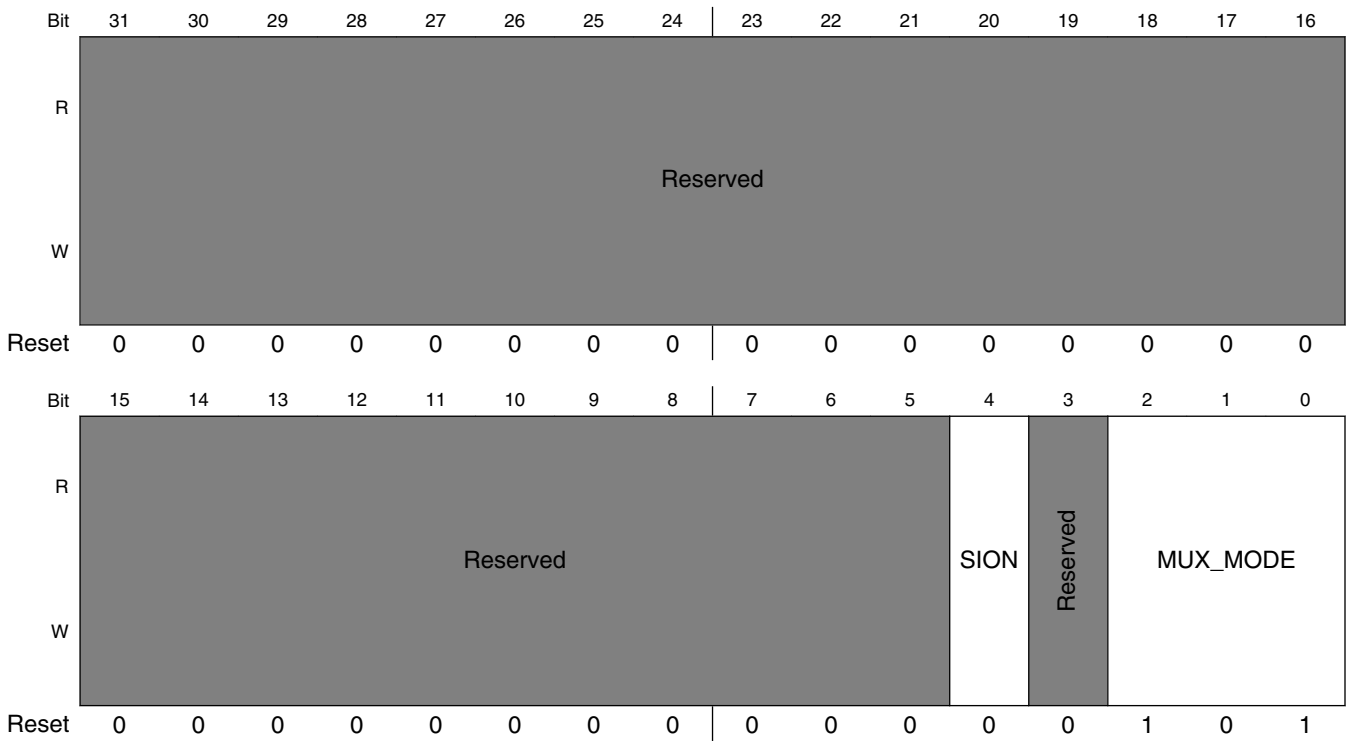
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_RXD1 000 **ALT0_SAI1_RX_DATA1** — Select mux mode: ALT0 mux port: RX_DATA1 of instance: SAI1 001 **ALT1_SAI5_RX_DATA1** — Select mux mode: ALT1 mux port: RX_DATA1 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE1** — Select mux mode: ALT4 mux port: TRACE1 of instance: CORESIGHT 101 **ALT5_GPIO4_IO03** — Select mux mode: ALT5 mux port: IO03 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG1** — Select mux mode: ALT6 mux port: BOOT_CFG1 of instance: SRC

8.2.5.87 SW_MUX_CTL_PAD_SAI1_RXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 16Ch offset = 3033_016Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_RXD2 000 **ALT0_SAI1_RX_DATA2** — Select mux mode: ALT0 mux port: RX_DATA2 of instance: SAI1 001 **ALT1_SAI5_RX_DATA2** — Select mux mode: ALT1 mux port: RX_DATA2 of instance: SAI5

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD2 field descriptions (continued)

Field	Description
	100 **ALT4_CORESIGHT_TRACE2** — Select mux mode: ALT4 mux port: TRACE2 of instance: CORESIGHT 101 **ALT5_GPIO4_IO04** — Select mux mode: ALT5 mux port: IO04 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG2** — Select mux mode: ALT6 mux port: BOOT_CFG2 of instance: SRC

8.2.5.88 SW_MUX_CTL_PAD_SAI1_RXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 170h offset = 3033_0170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD3

Table continues on the next page...

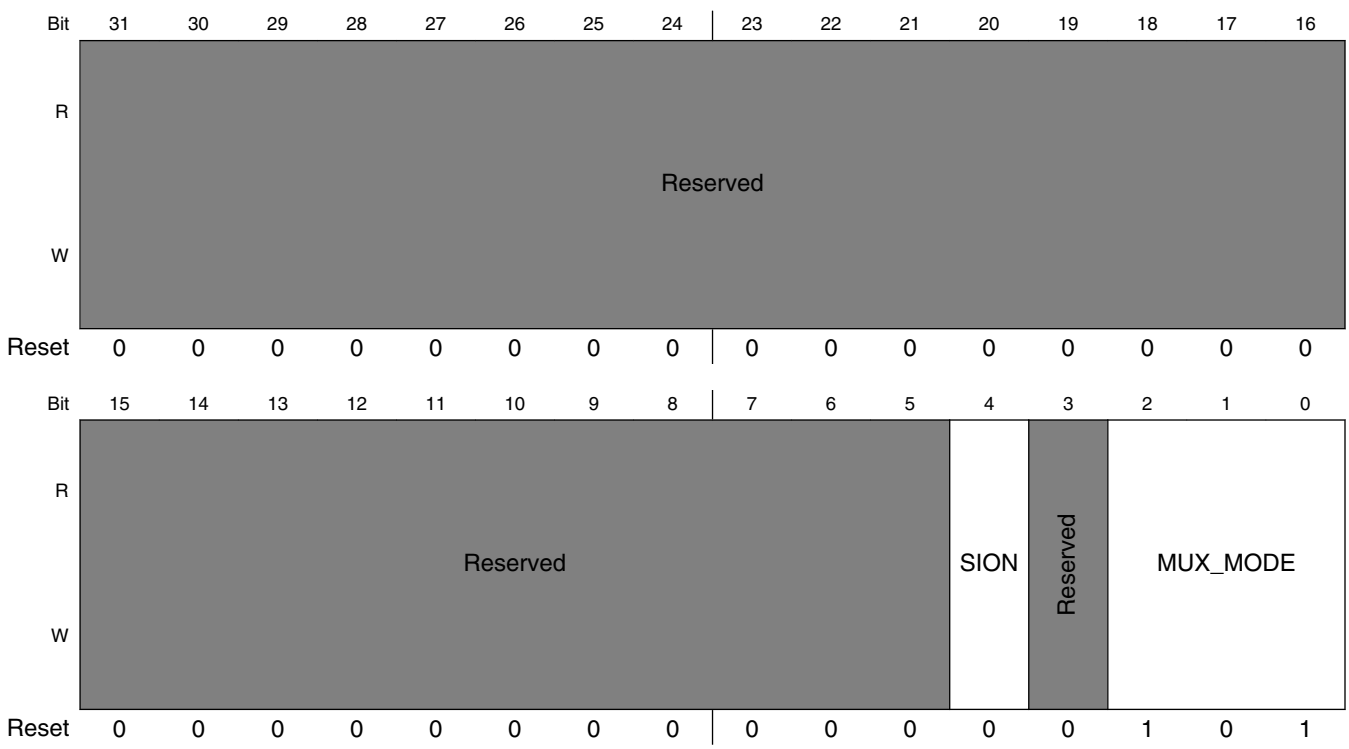
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD3 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_RXD3 000 **ALT0_SAI1_RX_DATA3** — Select mux mode: ALT0 mux port: RX_DATA3 of instance: SAI1 001 **ALT1_SAI5_RX_DATA3** — Select mux mode: ALT1 mux port: RX_DATA3 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE3** — Select mux mode: ALT4 mux port: TRACE3 of instance: CORESIGHT 101 **ALT5_GPIO4_IO05** — Select mux mode: ALT5 mux port: IO05 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG3** — Select mux mode: ALT6 mux port: BOOT_CFG3 of instance: SRC

8.2.5.89 SW_MUX_CTL_PAD_SAI1_RXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4)

SW_MUX_CTL Register

Address: 3033_0000h base + 174h offset = 3033_0174h



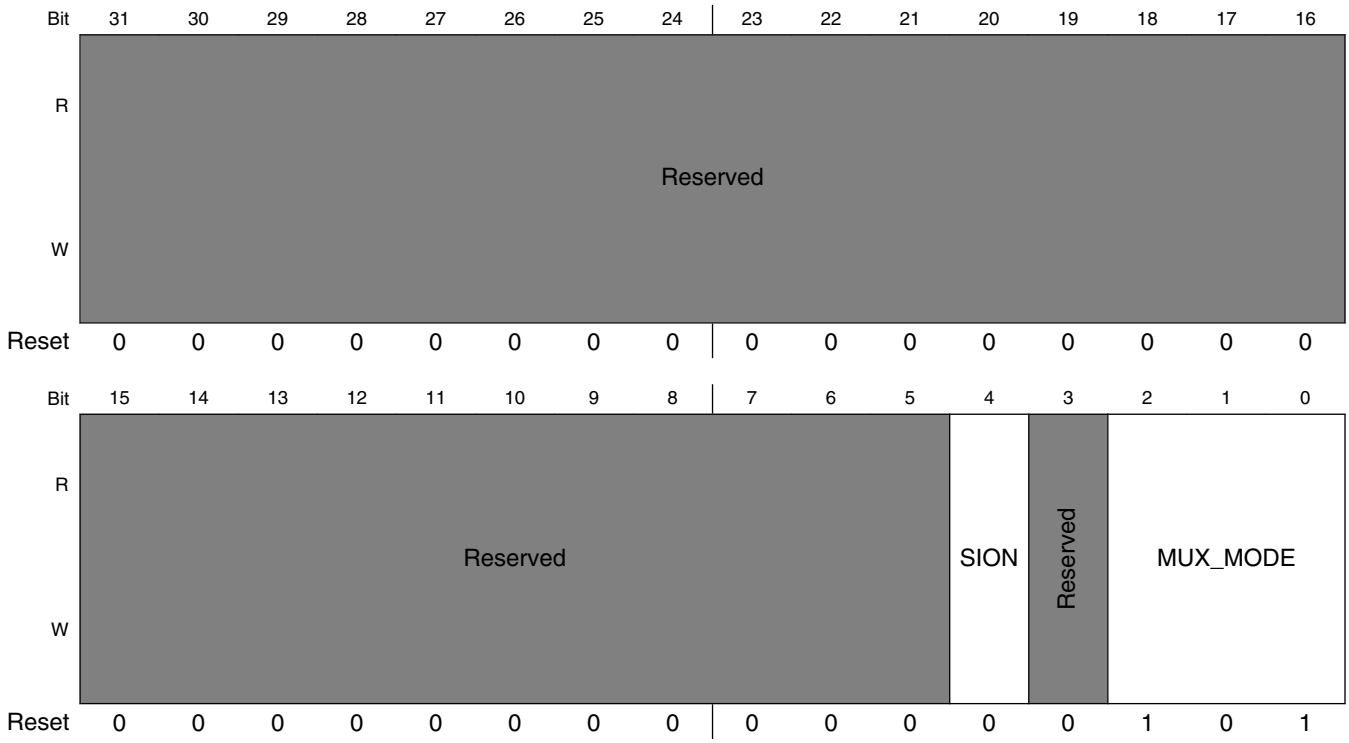
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD4 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD4
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 6 iomux modes to be used for pad: SAI1_RXD4 000 **ALT0_SAI1_RX_DATA4** — Select mux mode: ALT0 mux port: RX_DATA4 of instance: SAI1 001 **ALT1_SAI6_TX_BCLK** — Select mux mode: ALT1 mux port: TX_BCLK of instance: SAI6 010 **ALT2_SAI6_RX_BCLK** — Select mux mode: ALT2 mux port: RX_BCLK of instance: SAI6 100 **ALT4_CORESIGHT_TRACE4** — Select mux mode: ALT4 mux port: TRACE4 of instance: CORESIGHT 101 **ALT5_GPIO4_IO06** — Select mux mode: ALT5 mux port: IO06 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG4** — Select mux mode: ALT6 mux port: BOOT_CFG4 of instance: SRC

8.2.5.90 SW_MUX_CTL_PAD_SAI1_RXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5)

SW_MUX_CTL Register

Address: 3033_0000h base + 178h offset = 3033_0178h



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD5 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD5
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 7 iomux modes to be used for pad: SAI1_RXD5 000 **ALT0_SAI1_RX_DATA5** — Select mux mode: ALT0 mux port: RX_DATA5 of instance: SAI1 001 **ALT1_SAI6_TX_DATA0** — Select mux mode: ALT1 mux port: TX_DATA0 of instance: SAI6 010 **ALT2_SAI6_RX_DATA0** — Select mux mode: ALT2 mux port: RX_DATA0 of instance: SAI6

Table continues on the next page...

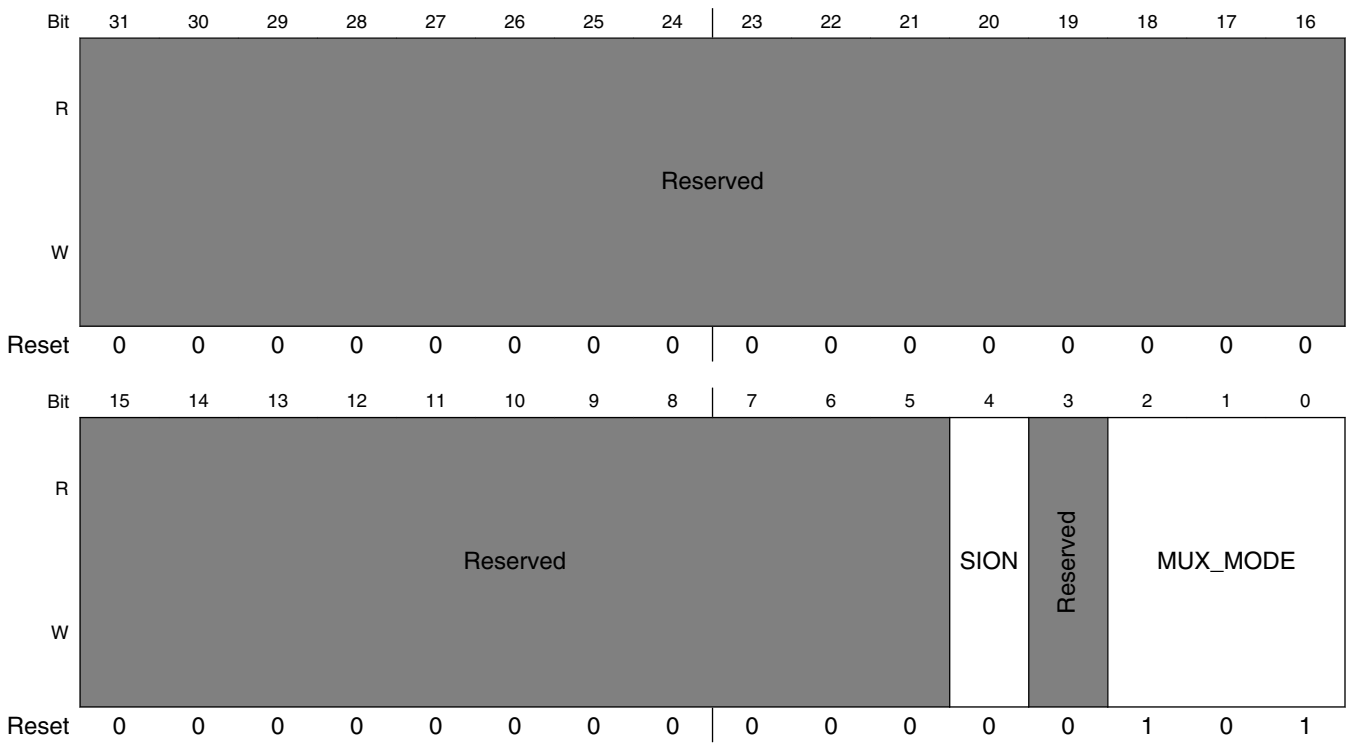
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD5 field descriptions (continued)

Field	Description
	011 **ALT3_SAI1_RX_SYNC** — Select mux mode: ALT3 mux port: RX_SYNC of instance: SAI1 100 **ALT4_CORESIGHT_TRACE5** — Select mux mode: ALT4 mux port: TRACE5 of instance: CORESIGHT 101 **ALT5_GPIO4_IO07** — Select mux mode: ALT5 mux port: IO07 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG5** — Select mux mode: ALT6 mux port: BOOT_CFG5 of instance: SRC

8.2.5.91 SW_MUX_CTL_PAD_SAI1_RXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6)

SW_MUX_CTL Register

Address: 3033_0000h base + 17Ch offset = 3033_017Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality

Table continues on the next page...

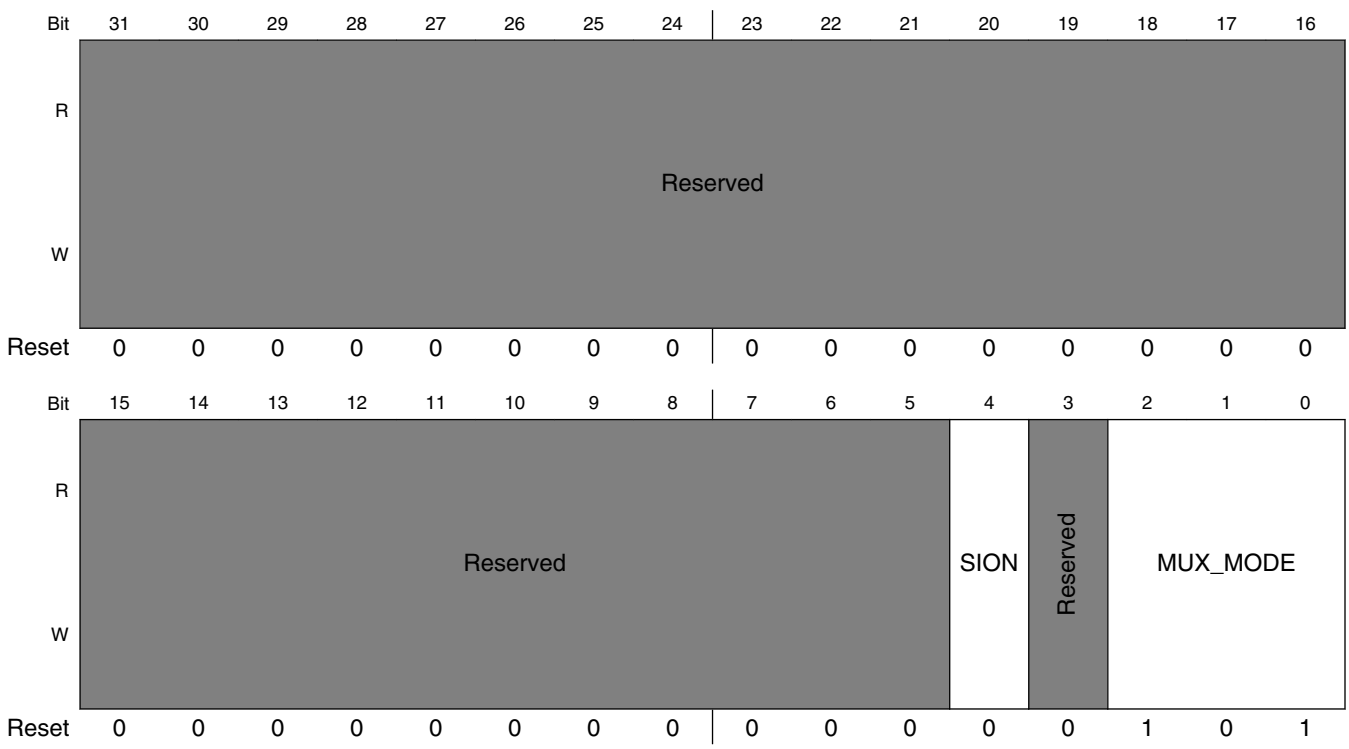
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD6 field descriptions (continued)

Field	Description
	0 **SION_DISABLED** — Input Path of pad SAI1_RXD6 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD6
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 6 iomux modes to be used for pad: SAI1_RXD6 000 **ALT0_SAI1_RX_DATA6** — Select mux mode: ALT0 mux port: RX_DATA6 of instance: SAI1 001 **ALT1_SAI6_TX_SYNC** — Select mux mode: ALT1 mux port: TX_SYNC of instance: SAI6 010 **ALT2_SAI6_RX_SYNC** — Select mux mode: ALT2 mux port: RX_SYNC of instance: SAI6 100 **ALT4_CORESIGHT_TRACE6** — Select mux mode: ALT4 mux port: TRACE6 of instance: CORESIGHT 101 **ALT5_GPIO4_IO08** — Select mux mode: ALT5 mux port: IO08 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG6** — Select mux mode: ALT6 mux port: BOOT_CFG6 of instance: SRC

8.2.5.92 SW_MUX_CTL_PAD_SAI1_RXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7)

SW_MUX_CTL Register

Address: 3033_0000h base + 180h offset = 3033_0180h



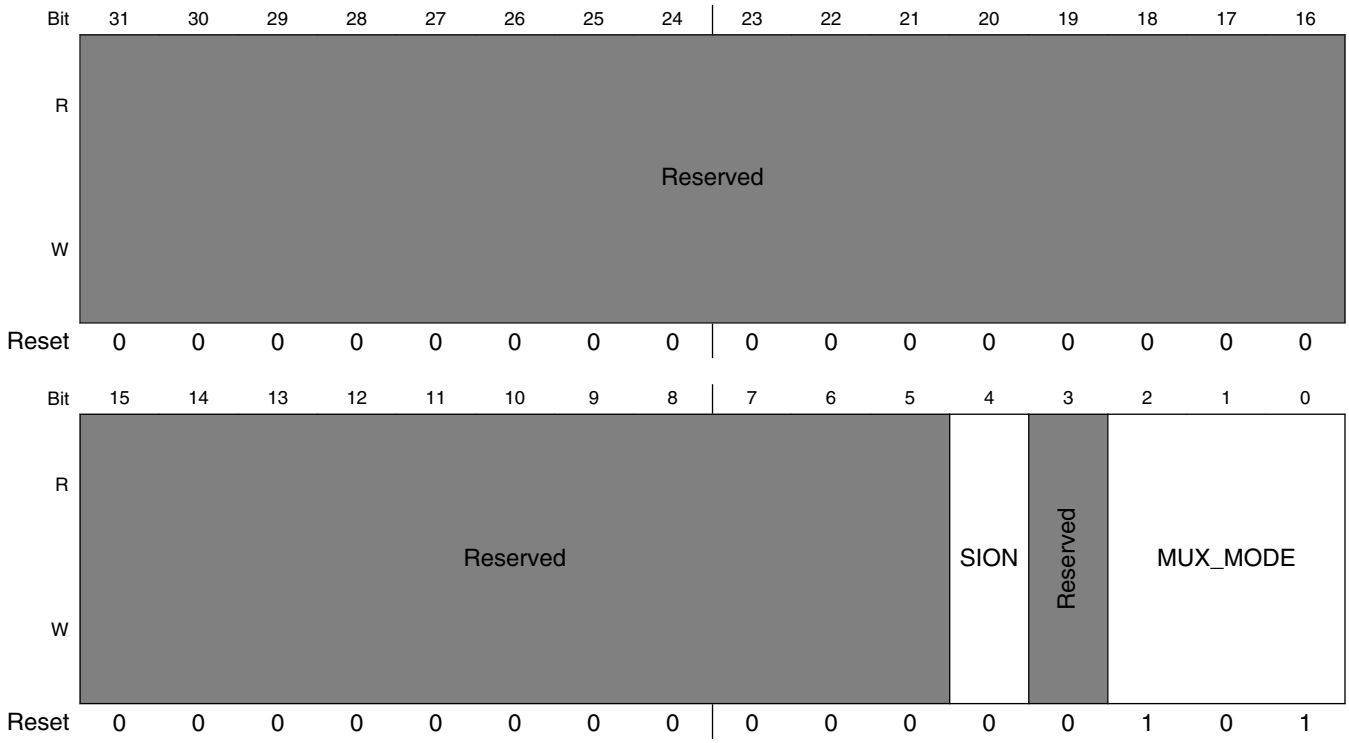
IOMUXC_SW_MUX_CTL_PAD_SAI1_RXD7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_RXD7 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_RXD7
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 7 iomux modes to be used for pad: SAI1_RXD7 000 **ALT0_SAI1_RX_DATA7** — Select mux mode: ALT0 mux port: RX_DATA7 of instance: SAI1 001 **ALT1_SAI6_MCLK** — Select mux mode: ALT1 mux port: MCLK of instance: SAI6 010 **ALT2_SAI1_TX_SYNC** — Select mux mode: ALT2 mux port: TX_SYNC of instance: SAI1 011 **ALT3_SAI1_TX_DATA4** — Select mux mode: ALT3 mux port: TX_DATA4 of instance: SAI1 100 **ALT4_CORESIGHT_TRACE7** — Select mux mode: ALT4 mux port: TRACE7 of instance: CORESIGHT 101 **ALT5_GPIO4_IO09** — Select mux mode: ALT5 mux port: IO09 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG7** — Select mux mode: ALT6 mux port: BOOT_CFG7 of instance: SRC

8.2.5.93 SW_MUX_CTL_PAD_SAI1_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 184h offset = 3033_0184h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI1_TXFS 000 **ALT0_SAI1_TX_SYNC** — Select mux mode: ALT0 mux port: TX_SYNC of instance: SAI1 001 **ALT1_SAI5_TX_SYNC** — Select mux mode: ALT1 mux port: TX_SYNC of instance: SAI5

Table continues on the next page...

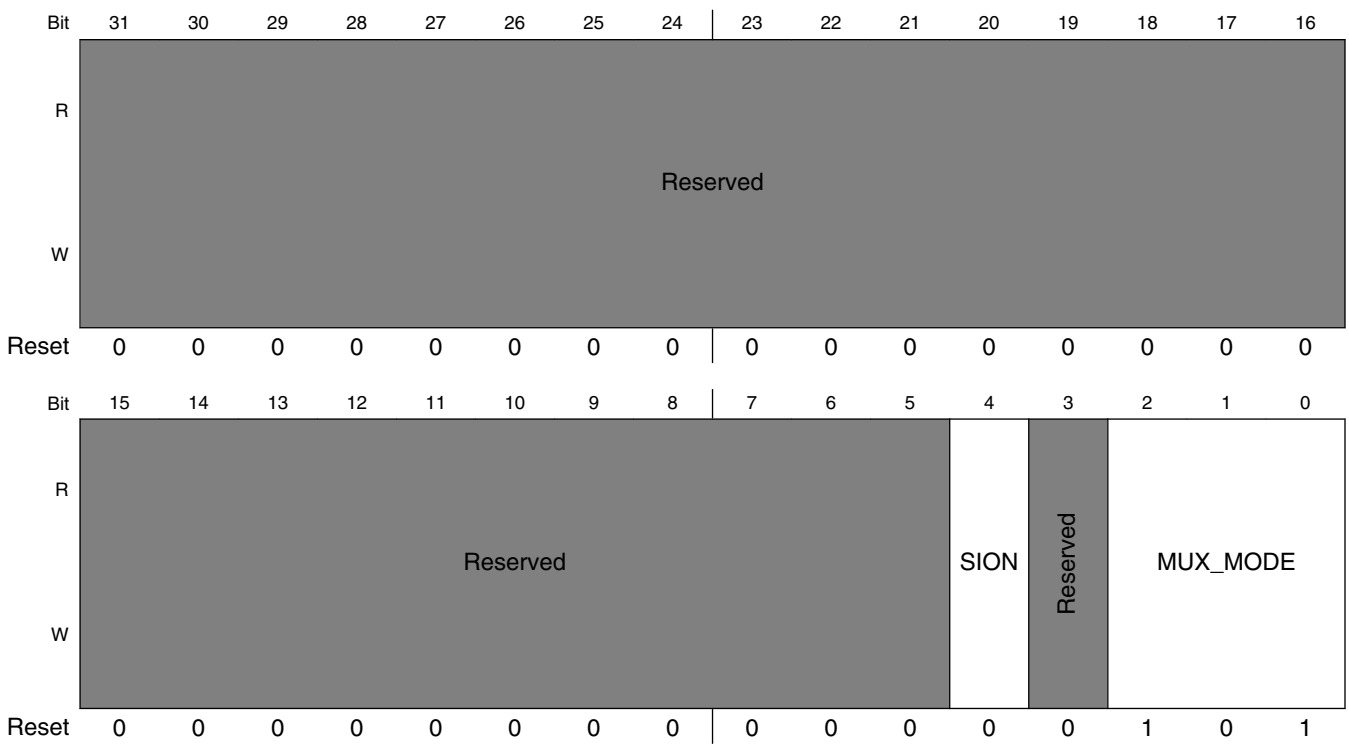
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXFS field descriptions (continued)

Field	Description
	100 **ALT4_CORESIGHT_EVENTO** — Select mux mode: ALT4 mux port: EVENTO of instance: CORESIGHT 101 **ALT5_GPIO4_IO10** — Select mux mode: ALT5 mux port: IO10 of instance: GPIO4

8.2.5.94 SW_MUX_CTL_PAD_SAI1_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 188h offset = 3033_0188h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXC

Table continues on the next page...

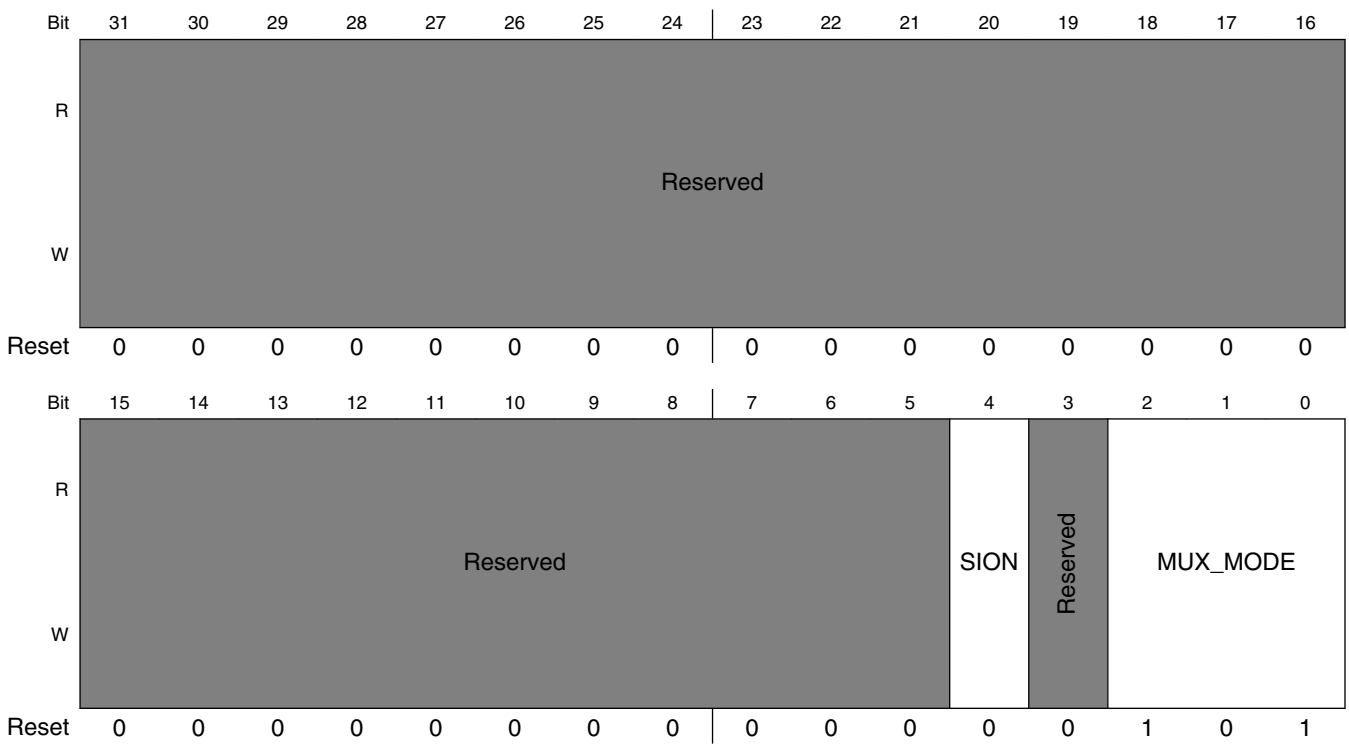
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXC field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI1_TXC 000 **ALT0_SAI1_TX_BCLK** — Select mux mode: ALT0 mux port: TX_BCLK of instance: SAI1 001 **ALT1_SAI5_TX_BCLK** — Select mux mode: ALT1 mux port: TX_BCLK of instance: SAI5 100 **ALT4_CORESIGHT_EVENTI** — Select mux mode: ALT4 mux port: EVENTI of instance: CORESIGHT 101 **ALT5_GPIO4_IO11** — Select mux mode: ALT5 mux port: IO11 of instance: GPIO4

8.2.5.95 SW_MUX_CTL_PAD_SAI1_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 18Ch offset = 3033_018Ch



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD0 field descriptions (continued)

Field	Description
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_TXD0 000 **ALT0_SAI1_TX_DATA0** — Select mux mode: ALT0 mux port: TX_DATA0 of instance: SAI1 001 **ALT1_SAI5_TX_DATA0** — Select mux mode: ALT1 mux port: TX_DATA0 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE8** — Select mux mode: ALT4 mux port: TRACE8 of instance: CORESIGHT 101 **ALT5_GPIO4_IO12** — Select mux mode: ALT5 mux port: IO12 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG8** — Select mux mode: ALT6 mux port: BOOT_CFG8 of instance: SRC

8.2.5.96 SW_MUX_CTL_PAD_SAI1_TXD1 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1)

SW_MUX_CTL Register

Address: 3033_0000h base + 190h offset = 3033_0190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

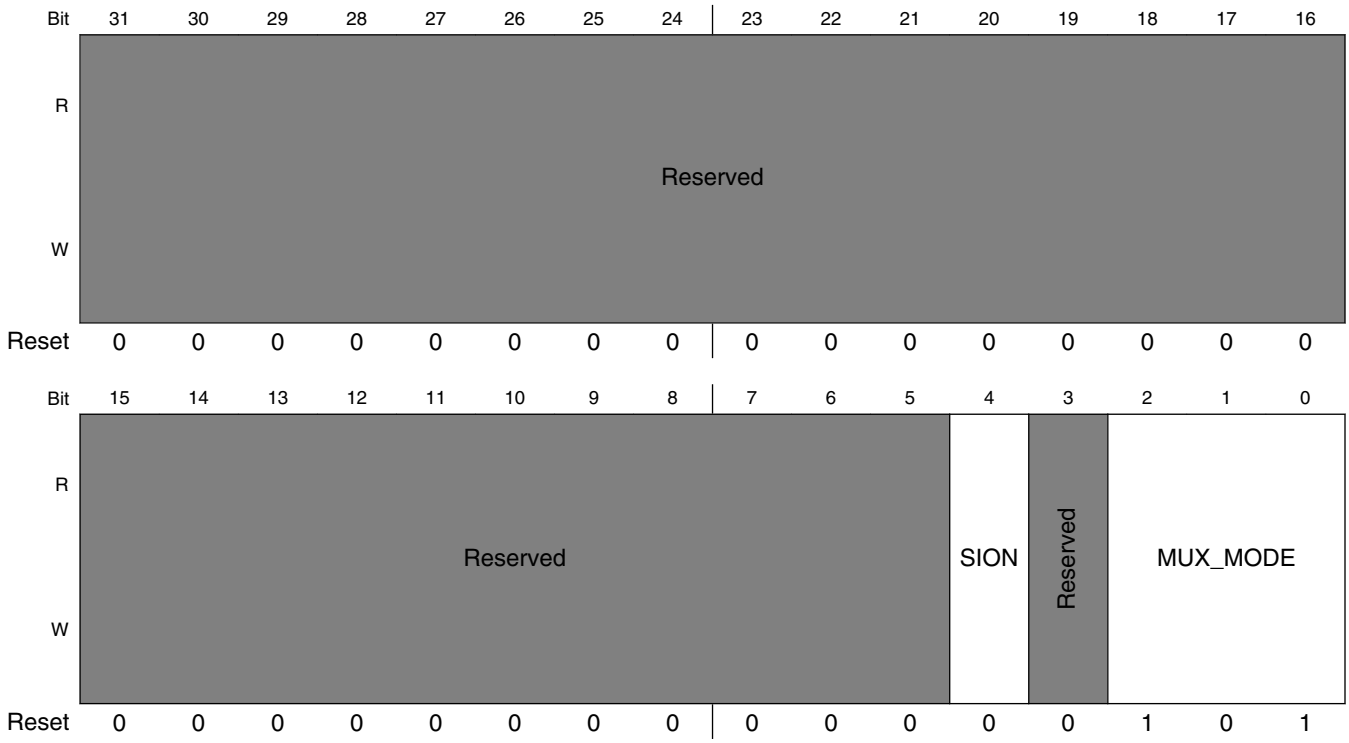
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD1 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD1 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD1
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_TXD1 000 **ALT0_SAI1_TX_DATA1** — Select mux mode: ALT0 mux port: TX_DATA1 of instance: SAI1 001 **ALT1_SAI5_TX_DATA1** — Select mux mode: ALT1 mux port: TX_DATA1 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE9** — Select mux mode: ALT4 mux port: TRACE9 of instance: CORESIGHT 101 **ALT5_GPIO4_IO13** — Select mux mode: ALT5 mux port: IO13 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG9** — Select mux mode: ALT6 mux port: BOOT_CFG9 of instance: SRC

8.2.5.97 SW_MUX_CTL_PAD_SAI1_TXD2 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2)

SW_MUX_CTL Register

Address: 3033_0000h base + 194h offset = 3033_0194h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD2 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD2
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_TXD2 000 **ALT0_SAI1_TX_DATA2** — Select mux mode: ALT0 mux port: TX_DATA2 of instance: SAI1 001 **ALT1_SAI5_TX_DATA2** — Select mux mode: ALT1 mux port: TX_DATA2 of instance: SAI5

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD2 field descriptions (continued)

Field	Description
	100 **ALT4_CORESIGHT_TRACE10** — Select mux mode: ALT4 mux port: TRACE10 of instance: CORESIGHT 101 **ALT5_GPIO4_IO14** — Select mux mode: ALT5 mux port: IO14 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG10** — Select mux mode: ALT6 mux port: BOOT_CFG10 of instance: SRC

8.2.5.98 SW_MUX_CTL_PAD_SAI1_TXD3 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3)

SW_MUX_CTL Register

Address: 3033_0000h base + 198h offset = 3033_0198h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD3 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD3

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD3 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_TXD3 000 **ALT0_SAI1_TX_DATA3** — Select mux mode: ALT0 mux port: TX_DATA3 of instance: SAI1 001 **ALT1_SAI5_TX_DATA3** — Select mux mode: ALT1 mux port: TX_DATA3 of instance: SAI5 100 **ALT4_CORESIGHT_TRACE11** — Select mux mode: ALT4 mux port: TRACE11 of instance: CORESIGHT 101 **ALT5_GPIO4_IO15** — Select mux mode: ALT5 mux port: IO15 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG11** — Select mux mode: ALT6 mux port: BOOT_CFG11 of instance: SRC

8.2.5.99 SW_MUX_CTL_PAD_SAI1_TXD4 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4)

SW_MUX_CTL Register

Address: 3033_0000h base + 19Ch offset = 3033_019Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

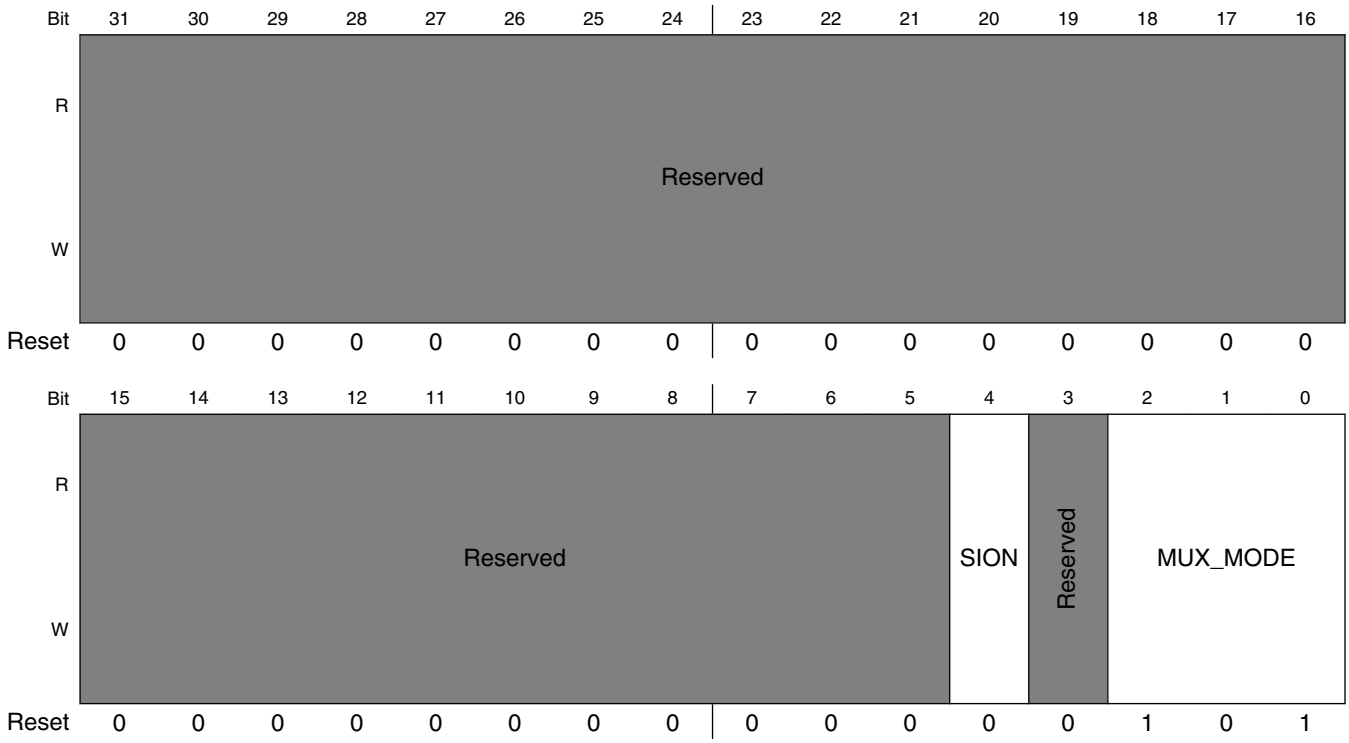
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD4 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD4 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD4
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 6 iomux modes to be used for pad: SAI1_TXD4 000 **ALT0_SAI1_TX_DATA4** — Select mux mode: ALT0 mux port: TX_DATA4 of instance: SAI1 001 **ALT1_SAI6_RX_BCLK** — Select mux mode: ALT1 mux port: RX_BCLK of instance: SAI6 010 **ALT2_SAI6_TX_BCLK** — Select mux mode: ALT2 mux port: TX_BCLK of instance: SAI6 100 **ALT4_CORESIGHT_TRACE12** — Select mux mode: ALT4 mux port: TRACE12 of instance: CORESIGHT 101 **ALT5_GPIO4_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG12** — Select mux mode: ALT6 mux port: BOOT_CFG12 of instance: SRC

8.2.5.100 SW_MUX_CTL_PAD_SAI1_TXD5 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A0h offset = 3033_01A0h



IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD5 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD5
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 6 iomux modes to be used for pad: SAI1_TXD5 000 **ALT0_SAI1_TX_DATA5** — Select mux mode: ALT0 mux port: TX_DATA5 of instance: SAI1 001 **ALT1_SAI6_RX_DATA0** — Select mux mode: ALT1 mux port: RX_DATA0 of instance: SAI6 010 **ALT2_SAI6_TX_DATA0** — Select mux mode: ALT2 mux port: TX_DATA0 of instance: SAI6

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD5 field descriptions (continued)

Field	Description
	100 **ALT4_CORESIGHT_TRACE13** — Select mux mode: ALT4 mux port: TRACE13 of instance: CORESIGHT 101 **ALT5_GPIO4_IO17** — Select mux mode: ALT5 mux port: IO17 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG13** — Select mux mode: ALT6 mux port: BOOT_CFG13 of instance: SRC

8.2.5.101 SW_MUX_CTL_PAD_SAI1_TXD6 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A4h offset = 3033_01A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD6 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD6

Table continues on the next page...

IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD6 field descriptions (continued)

Field	Description
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 6 iomux modes to be used for pad: SAI1_TXD6 000 **ALT0_SAI1_TX_DATA6** — Select mux mode: ALT0 mux port: TX_DATA6 of instance: SAI1 001 **ALT1_SAI6_RX_SYNC** — Select mux mode: ALT1 mux port: RX_SYNC of instance: SAI6 010 **ALT2_SAI6_TX_SYNC** — Select mux mode: ALT2 mux port: TX_SYNC of instance: SAI6 100 **ALT4_CORESIGHT_TRACE14** — Select mux mode: ALT4 mux port: TRACE14 of instance: CORESIGHT 101 **ALT5_GPIO4_IO18** — Select mux mode: ALT5 mux port: IO18 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG14** — Select mux mode: ALT6 mux port: BOOT_CFG14 of instance: SRC

8.2.5.102 SW_MUX_CTL_PAD_SAI1_TXD7 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7)

SW_MUX_CTL Register

Address: 3033_0000h base + 1A8h offset = 3033_01A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

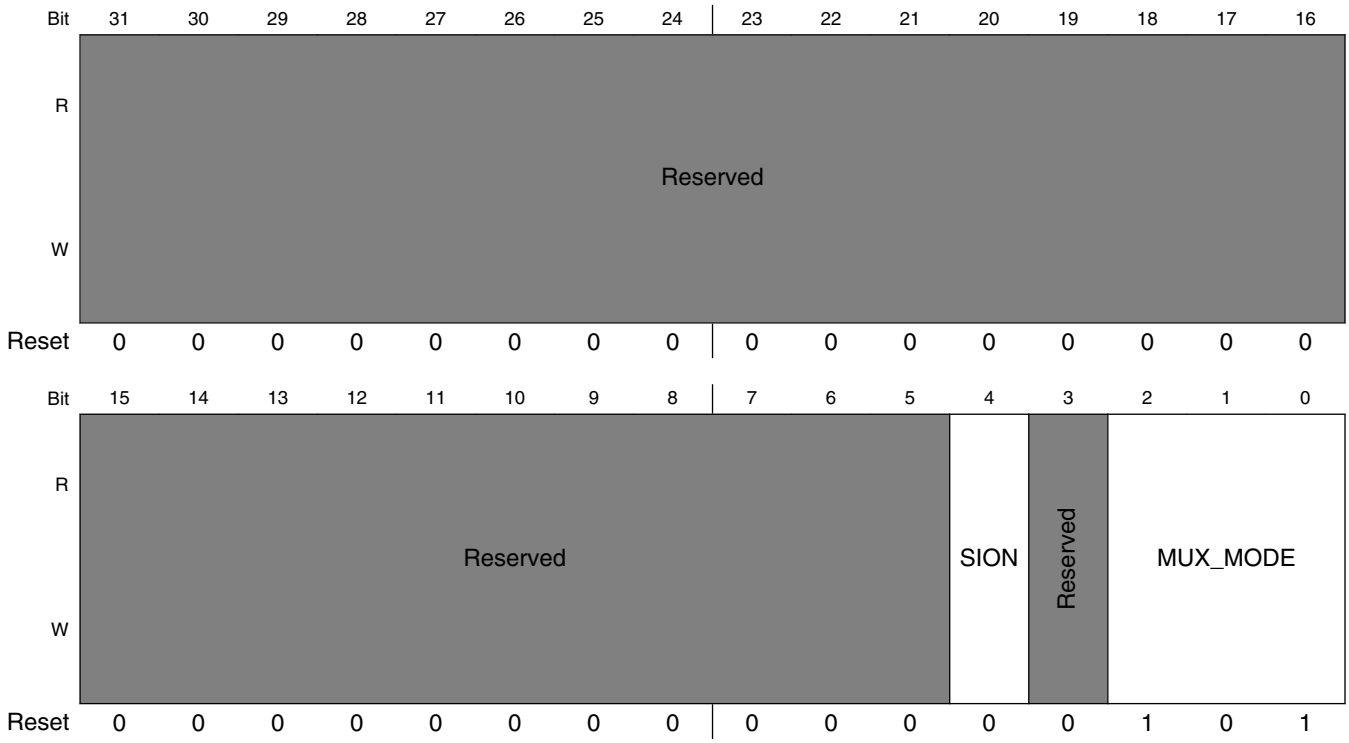
IOMUXC_SW_MUX_CTL_PAD_SAI1_TXD7 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_TXD7 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_TXD7
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 5 iomux modes to be used for pad: SAI1_TXD7 000 **ALT0_SAI1_TX_DATA7** — Select mux mode: ALT0 mux port: TX_DATA7 of instance: SAI1 001 **ALT1_SAI6_MCLK** — Select mux mode: ALT1 mux port: MCLK of instance: SAI6 100 **ALT4_CORESIGHT_TRACE15** — Select mux mode: ALT4 mux port: TRACE15 of instance: CORESIGHT 101 **ALT5_GPIO4_IO19** — Select mux mode: ALT5 mux port: IO19 of instance: GPIO4 110 **ALT6_SRC_BOOT_CFG15** — Select mux mode: ALT6 mux port: BOOT_CFG15 of instance: SRC

8.2.5.103 SW_MUX_CTL_PAD_SAI1_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1ACh offset = 3033_01ACh



IOMUXC_SW_MUX_CTL_PAD_SAI1_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI1_MCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI1_MCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI1_MCLK 000 **ALT0_SAI1_MCLK** — Select mux mode: ALT0 mux port: MCLK of instance: SAI1 001 **ALT1_SAI5_MCLK** — Select mux mode: ALT1 mux port: MCLK of instance: SAI5 010 **ALT2_SAI1_TX_BCLK** — Select mux mode: ALT2 mux port: TX_BCLK of instance: SAI1 101 **ALT5_GPIO4_IO20** — Select mux mode: ALT5 mux port: IO20 of instance: GPIO4

8.2.5.104 SW_MUX_CTL_PAD_SAI2_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B0h offset = 3033_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

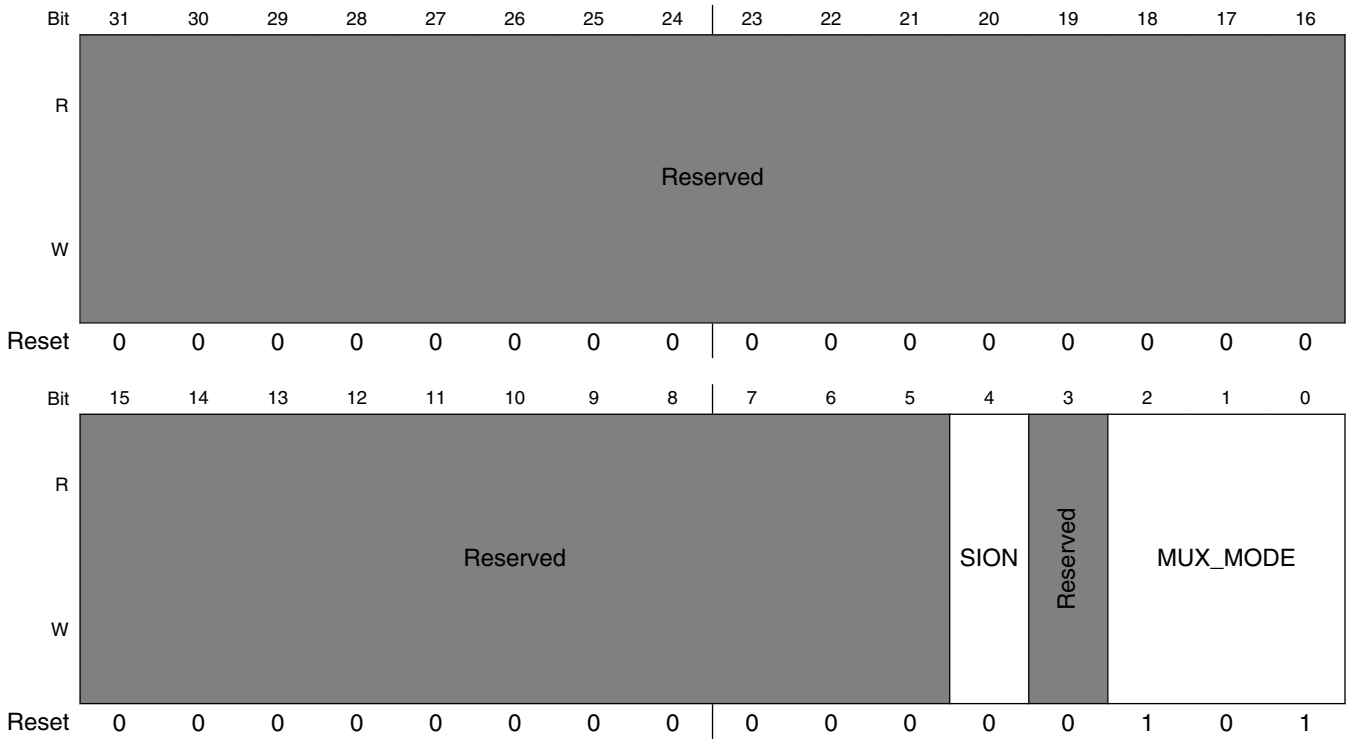
IOMUXC_SW_MUX_CTL_PAD_SAI2_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_RXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_RXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_RXFS 000 **ALT0_SAI2_RX_SYNC** — Select mux mode: ALT0 mux port: RX_SYNC of instance: SAI2 001 **ALT1_SAI5_TX_SYNC** — Select mux mode: ALT1 mux port: TX_SYNC of instance: SAI5 101 **ALT5_GPIO4_IO21** — Select mux mode: ALT5 mux port: IO21 of instance: GPIO4

8.2.5.105 SW_MUX_CTL_PAD_SAI2_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B4h offset = 3033_01B4h



IOMUXC_SW_MUX_CTL_PAD_SAI2_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_RXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_RXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_RXC 000 **ALT0_SAI2_RX_BCLK** — Select mux mode: ALT0 mux port: RX_BCLK of instance: SAI2 001 **ALT1_SAI5_TX_BCLK** — Select mux mode: ALT1 mux port: TX_BCLK of instance: SAI5 101 **ALT5_GPIO4_IO22** — Select mux mode: ALT5 mux port: IO22 of instance: GPIO4

8.2.5.106 SW_MUX_CTL_PAD_SAI2_RXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1B8h offset = 3033_01B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

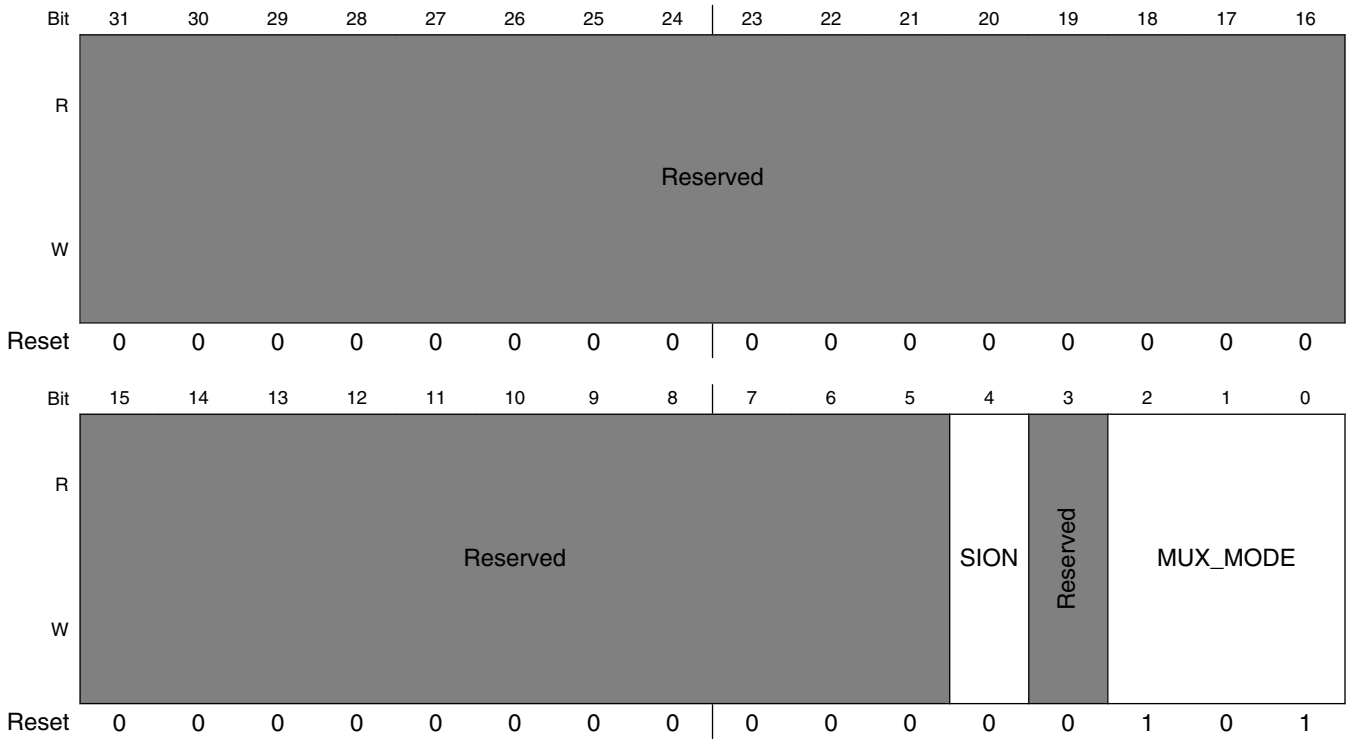
IOMUXC_SW_MUX_CTL_PAD_SAI2_RXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_RXD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_RXD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_RXD0 000 **ALT0_SAI2_RX_DATA0** — Select mux mode: ALT0 mux port: RX_DATA0 of instance: SAI2 001 **ALT1_SAI5_TX_DATA0** — Select mux mode: ALT1 mux port: TX_DATA0 of instance: SAI5 101 **ALT5_GPIO4_IO23** — Select mux mode: ALT5 mux port: IO23 of instance: GPIO4

8.2.5.107 SW_MUX_CTL_PAD_SAI2_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1BCh offset = 3033_01BCh



IOMUXC_SW_MUX_CTL_PAD_SAI2_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_TXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_TXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_TXFS 000 **ALT0_SAI2_TX_SYNC** — Select mux mode: ALT0 mux port: TX_SYNC of instance: SAI2 001 **ALT1_SAI5_TX_DATA1** — Select mux mode: ALT1 mux port: TX_DATA1 of instance: SAI5 101 **ALT5_GPIO4_IO24** — Select mux mode: ALT5 mux port: IO24 of instance: GPIO4

8.2.5.108 SW_MUX_CTL_PAD_SAI2_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C0h offset = 3033_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

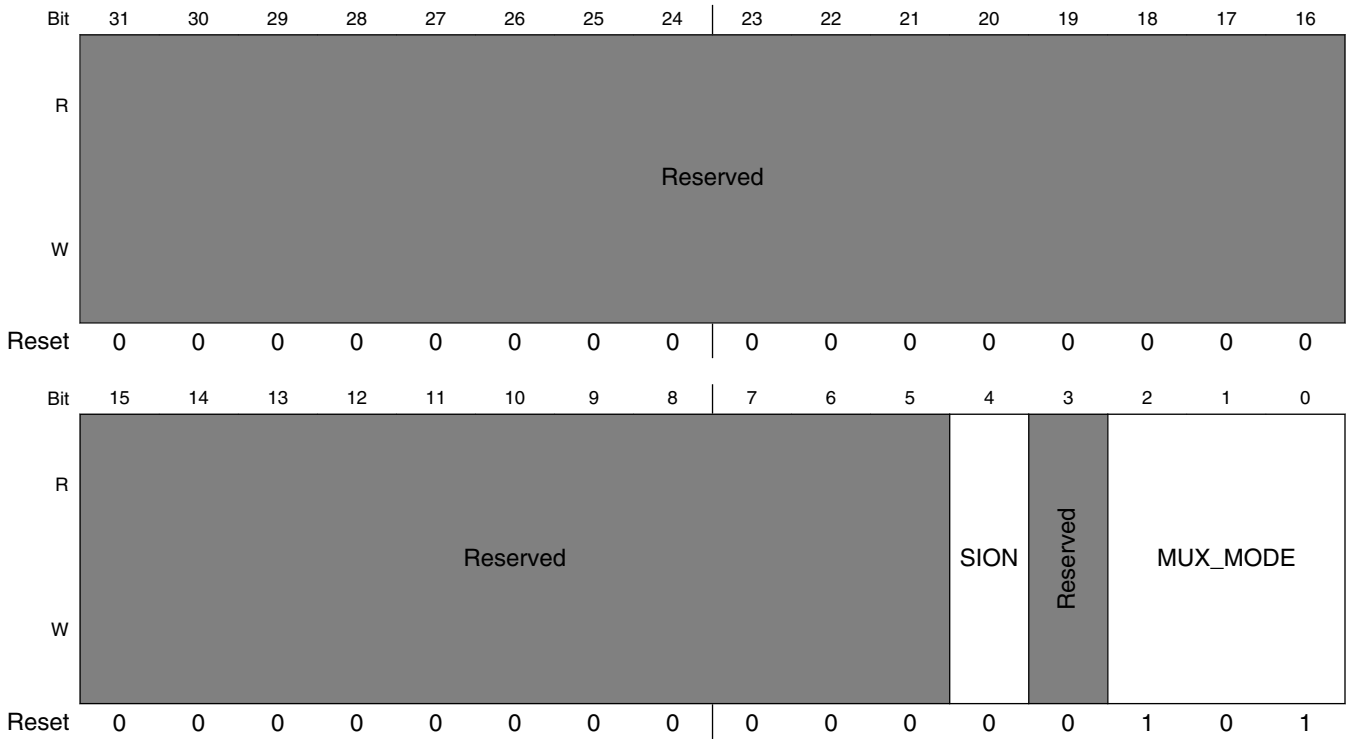
IOMUXC_SW_MUX_CTL_PAD_SAI2_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_TXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_TXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_TXC 000 **ALT0_SAI2_TX_BCLK** — Select mux mode: ALT0 mux port: TX_BCLK of instance: SAI2 001 **ALT1_SAI5_TX_DATA2** — Select mux mode: ALT1 mux port: TX_DATA2 of instance: SAI5 101 **ALT5_GPIO4_IO25** — Select mux mode: ALT5 mux port: IO25 of instance: GPIO4

8.2.5.109 SW_MUX_CTL_PAD_SAI2_TXD0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C4h offset = 3033_01C4h



IOMUXC_SW_MUX_CTL_PAD_SAI2_TXD0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_TXD0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_TXD0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_TXD0 000 **ALT0_SAI2_TX_DATA0** — Select mux mode: ALT0 mux port: TX_DATA0 of instance: SAI2 001 **ALT1_SAI5_TX_DATA3** — Select mux mode: ALT1 mux port: TX_DATA3 of instance: SAI5 101 **ALT5_GPIO4_IO26** — Select mux mode: ALT5 mux port: IO26 of instance: GPIO4

8.2.5.110 SW_MUX_CTL_PAD_SAI2_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1C8h offset = 3033_01C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

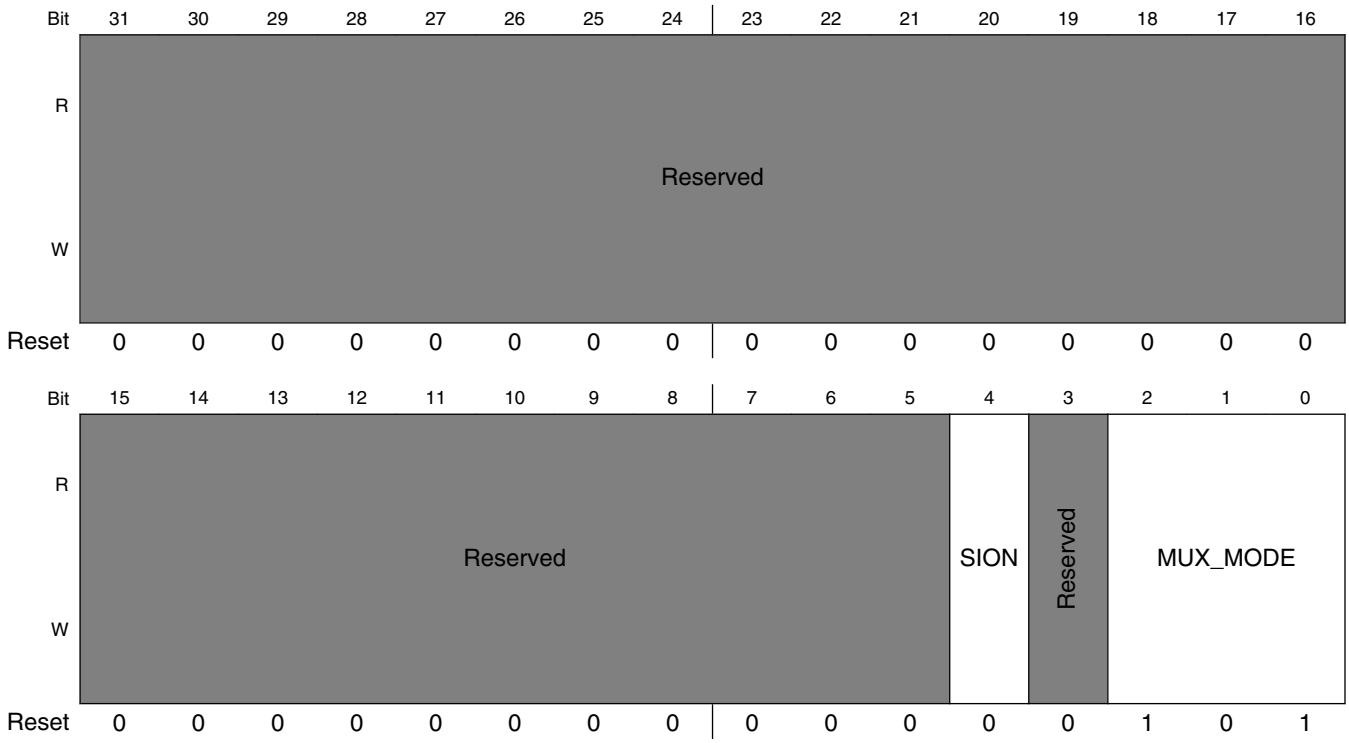
IOMUXC_SW_MUX_CTL_PAD_SAI2_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI2_MCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI2_MCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SAI2_MCLK 000 **ALT0_SAI2_MCLK** — Select mux mode: ALT0 mux port: MCLK of instance: SAI2 001 **ALT1_SAI5_MCLK** — Select mux mode: ALT1 mux port: MCLK of instance: SAI5 101 **ALT5_GPIO4_IO27** — Select mux mode: ALT5 mux port: IO27 of instance: GPIO4

8.2.5.111 SW_MUX_CTL_PAD_SAI3_RXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1CCh offset = 3033_01CCh



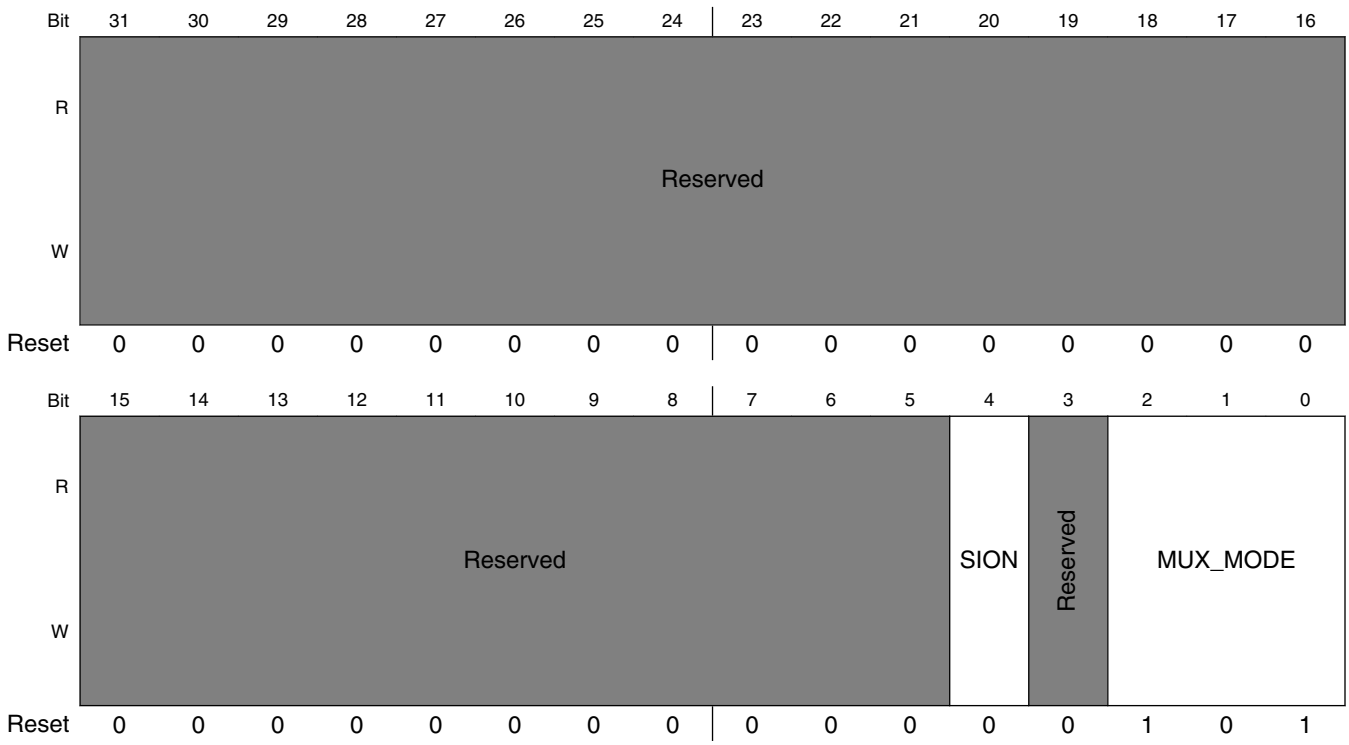
IOMUXC_SW_MUX_CTL_PAD_SAI3_RXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_RXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_RXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_RXFS 000 **ALT0_SAI3_RX_SYNC** — Select mux mode: ALT0 mux port: RX_SYNC of instance: SAI3 001 **ALT1_GPT1_CAPTURE1** — Select mux mode: ALT1 mux port: CAPTURE1 of instance: GPT1 010 **ALT2_SAI5_RX_SYNC** — Select mux mode: ALT2 mux port: RX_SYNC of instance: SAI5 101 **ALT5_GPIO4_IO28** — Select mux mode: ALT5 mux port: IO28 of instance: GPIO4

8.2.5.112 SW_MUX_CTL_PAD_SAI3_RXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D0h offset = 3033_01D0h



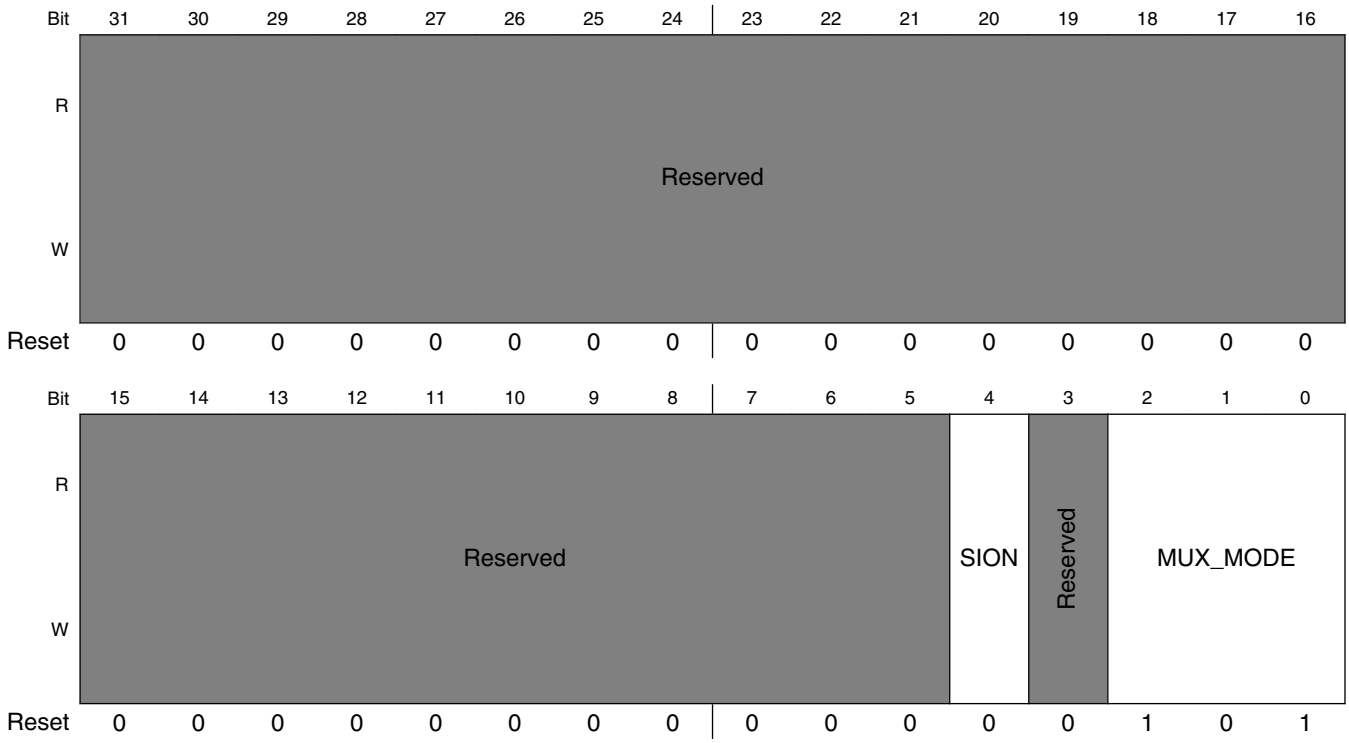
IOMUXC_SW_MUX_CTL_PAD_SAI3_RXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_RXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_RXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_RXC 000 **ALT0_SAI3_RX_BCLK** — Select mux mode: ALT0 mux port: RX_BCLK of instance: SAI3 001 **ALT1_GPT1_CAPTURE2** — Select mux mode: ALT1 mux port: CAPTURE2 of instance: GPT1 010 **ALT2_SAI5_RX_BCLK** — Select mux mode: ALT2 mux port: RX_BCLK of instance: SAI5 101 **ALT5_GPIO4_IO29** — Select mux mode: ALT5 mux port: IO29 of instance: GPIO4

8.2.5.113 SW_MUX_CTL_PAD_SAI3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D4h offset = 3033_01D4h



IOMUXC_SW_MUX_CTL_PAD_SAI3_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_RXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_RXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_RXD 000 **ALT0_SAI3_RX_DATA0** — Select mux mode: ALT0 mux port: RX_DATA0 of instance: SAI3 001 **ALT1_GPT1_COMPARE1** — Select mux mode: ALT1 mux port: COMPARE1 of instance: GPT1 010 **ALT2_SAI5_RX_DATA0** — Select mux mode: ALT2 mux port: RX_DATA0 of instance: SAI5 101 **ALT5_GPIO4_IO30** — Select mux mode: ALT5 mux port: IO30 of instance: GPIO4

8.2.5.114 SW_MUX_CTL_PAD_SAI3_TXFS SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS)

SW_MUX_CTL Register

Address: 3033_0000h base + 1D8h offset = 3033_01D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

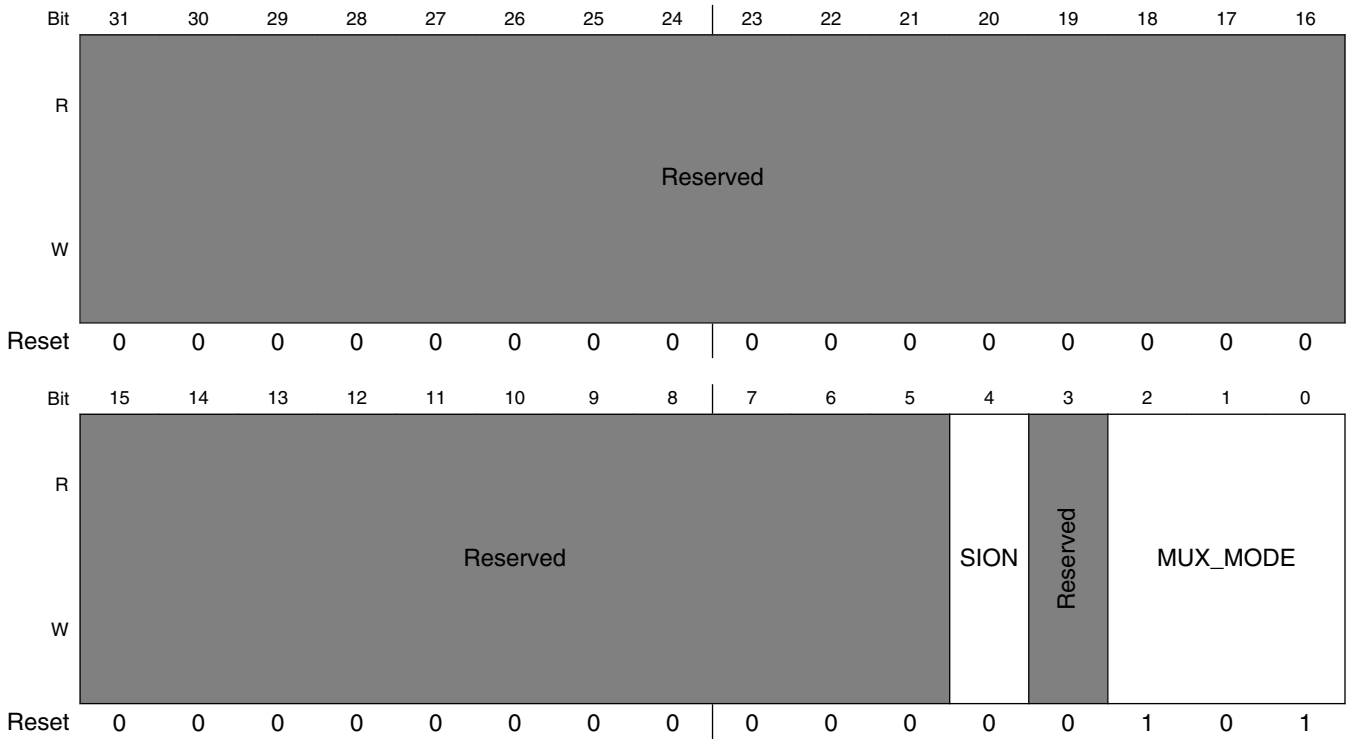
IOMUXC_SW_MUX_CTL_PAD_SAI3_TXFS field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_TXFS is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_TXFS
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_TXFS 000 **ALT0_SAI3_TX_SYNC** — Select mux mode: ALT0 mux port: TX_SYNC of instance: SAI3 001 **ALT1_GPT1_CLK** — Select mux mode: ALT1 mux port: CLK of instance: GPT1 010 **ALT2_SAI5_RX_DATA1** — Select mux mode: ALT2 mux port: RX_DATA1 of instance: SAI5 101 **ALT5_GPIO4_IO31** — Select mux mode: ALT5 mux port: IO31 of instance: GPIO4

8.2.5.115 SW_MUX_CTL_PAD_SAI3_TXC SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC)

SW_MUX_CTL Register

Address: 3033_0000h base + 1DCh offset = 3033_01DCh



IOMUXC_SW_MUX_CTL_PAD_SAI3_TXC field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_TXC is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_TXC
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_TXC 000 **ALT0_SAI3_TX_BCLK** — Select mux mode: ALT0 mux port: TX_BCLK of instance: SAI3 001 **ALT1_GPT1_COMPARE2** — Select mux mode: ALT1 mux port: COMPARE2 of instance: GPT1 010 **ALT2_SAI5_RX_DATA2** — Select mux mode: ALT2 mux port: RX_DATA2 of instance: SAI5 101 **ALT5_GPIO5_IO00** — Select mux mode: ALT5 mux port: IO00 of instance: GPIO5

8.2.5.116 SW_MUX_CTL_PAD_SAI3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E0h offset = 3033_01E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

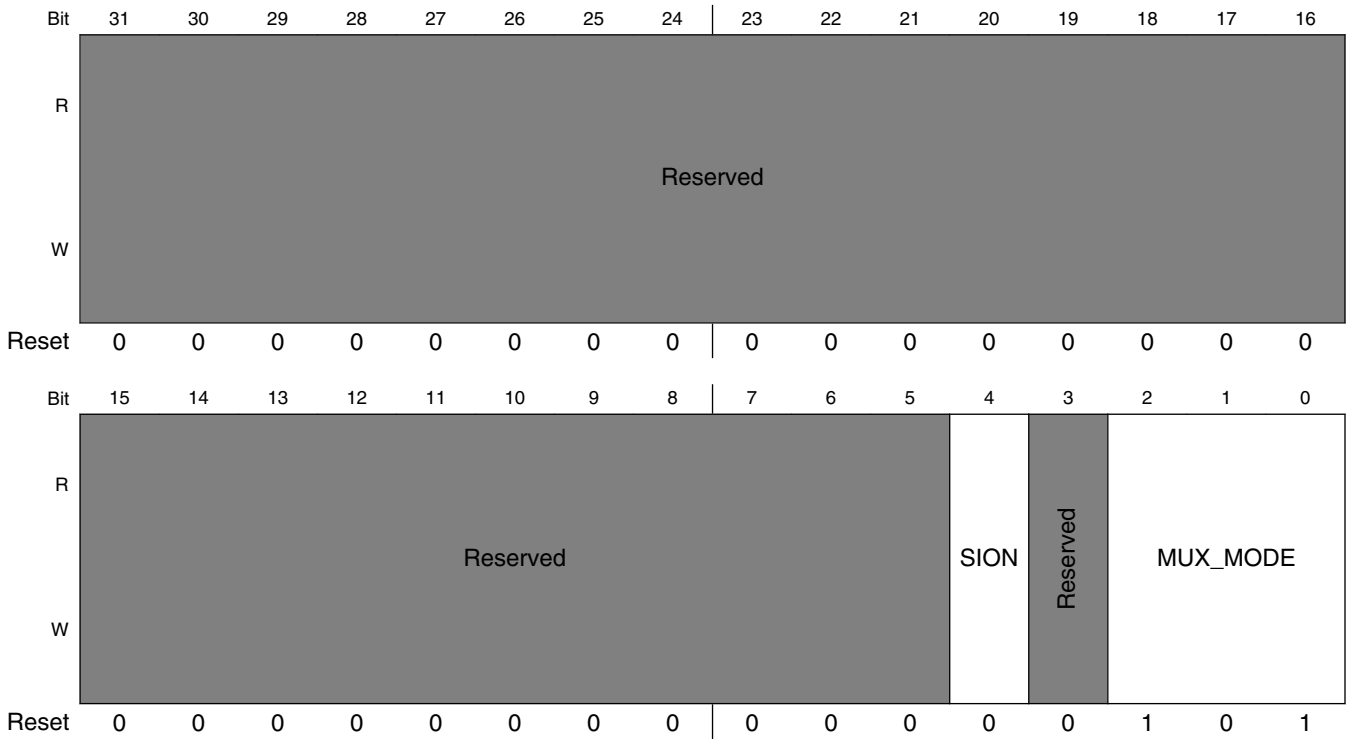
IOMUXC_SW_MUX_CTL_PAD_SAI3_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_TXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_TXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_TXD 000 **ALT0_SAI3_TX_DATA0** — Select mux mode: ALT0 mux port: TX_DATA0 of instance: SAI3 001 **ALT1_GPT1_COMPARE3** — Select mux mode: ALT1 mux port: COMPARE3 of instance: GPT1 010 **ALT2_SAI5_RX_DATA3** — Select mux mode: ALT2 mux port: RX_DATA3 of instance: SAI5 101 **ALT5_GPIO5_IO01** — Select mux mode: ALT5 mux port: IO01 of instance: GPIO5

8.2.5.117 SW_MUX_CTL_PAD_SAI3_MCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E4h offset = 3033_01E4h



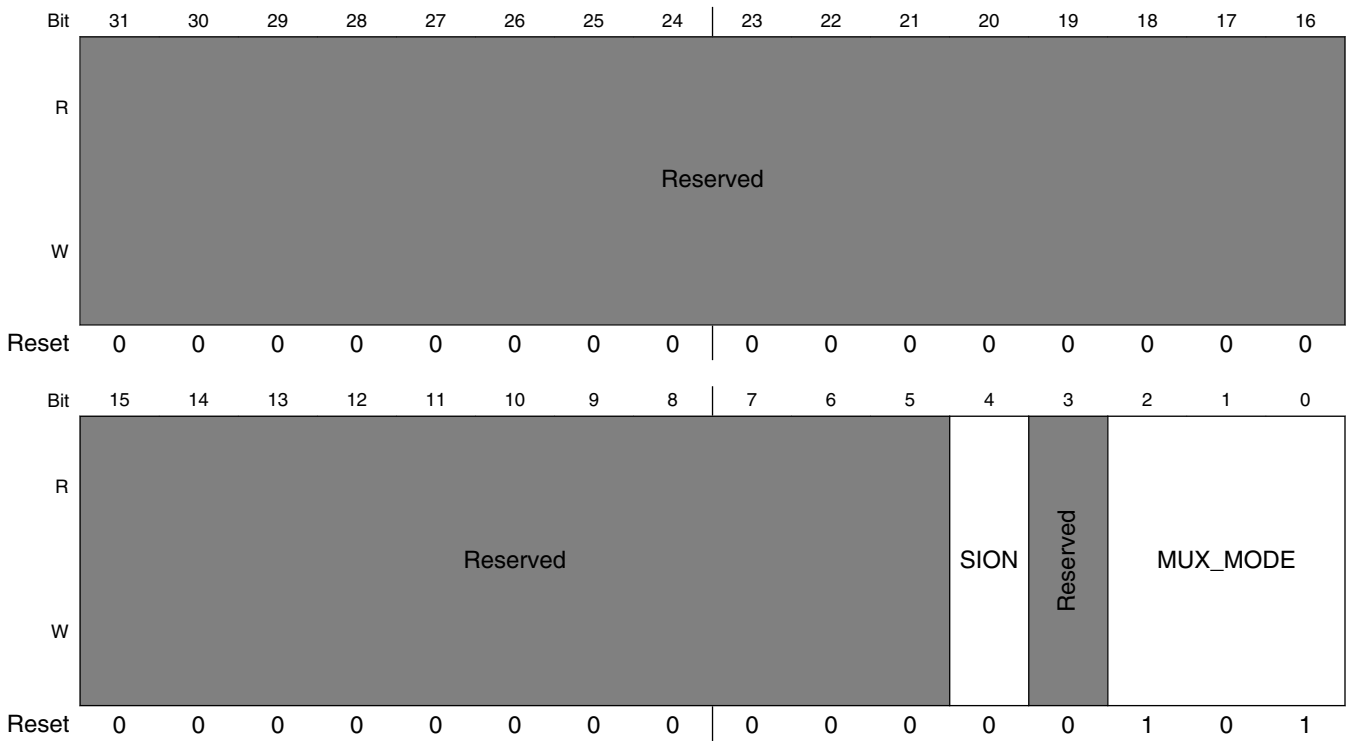
IOMUXC_SW_MUX_CTL_PAD_SAI3_MCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SAI3_MCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SAI3_MCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: SAI3_MCLK 000 **ALT0_SAI3_MCLK** — Select mux mode: ALT0 mux port: MCLK of instance: SAI3 001 **ALT1_PWM4_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM4 010 **ALT2_SAI5_MCLK** — Select mux mode: ALT2 mux port: MCLK of instance: SAI5 101 **ALT5_GPIO5_IO02** — Select mux mode: ALT5 mux port: IO02 of instance: GPIO5

8.2.5.118 SW_MUX_CTL_PAD_SPDIF_TX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX)

SW_MUX_CTL Register

Address: 3033_0000h base + 1E8h offset = 3033_01E8h



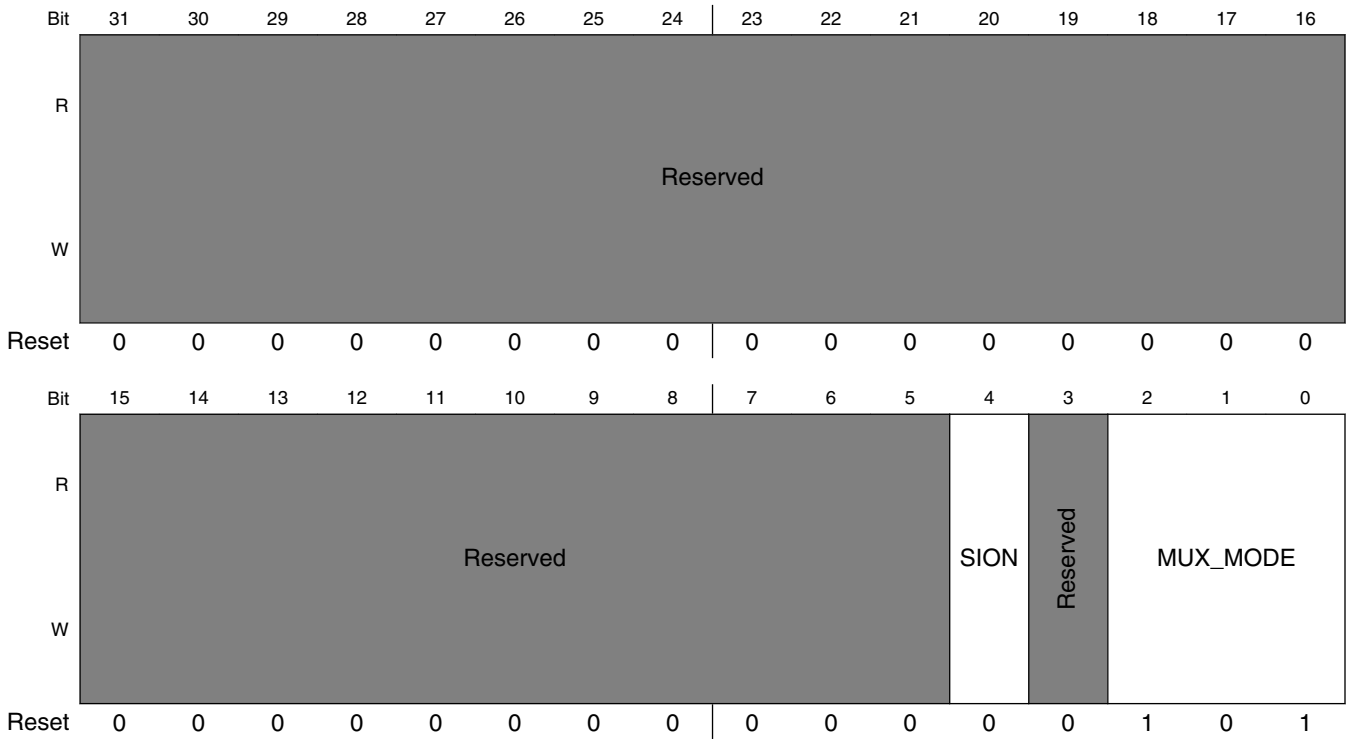
IOMUXC_SW_MUX_CTL_PAD_SPDIF_TX field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SPDIF_TX is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SPDIF_TX
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SPDIF_TX 000 **ALT0_SPDIF1_OUT** — Select mux mode: ALT0 mux port: OUT of instance: SPDIF1 001 **ALT1_PWM3_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM3 101 **ALT5_GPIO5_IO03** — Select mux mode: ALT5 mux port: IO03 of instance: GPIO5

8.2.5.119 SW_MUX_CTL_PAD_SPDIF_RX SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX)

SW_MUX_CTL Register

Address: 3033_0000h base + 1ECh offset = 3033_01ECh



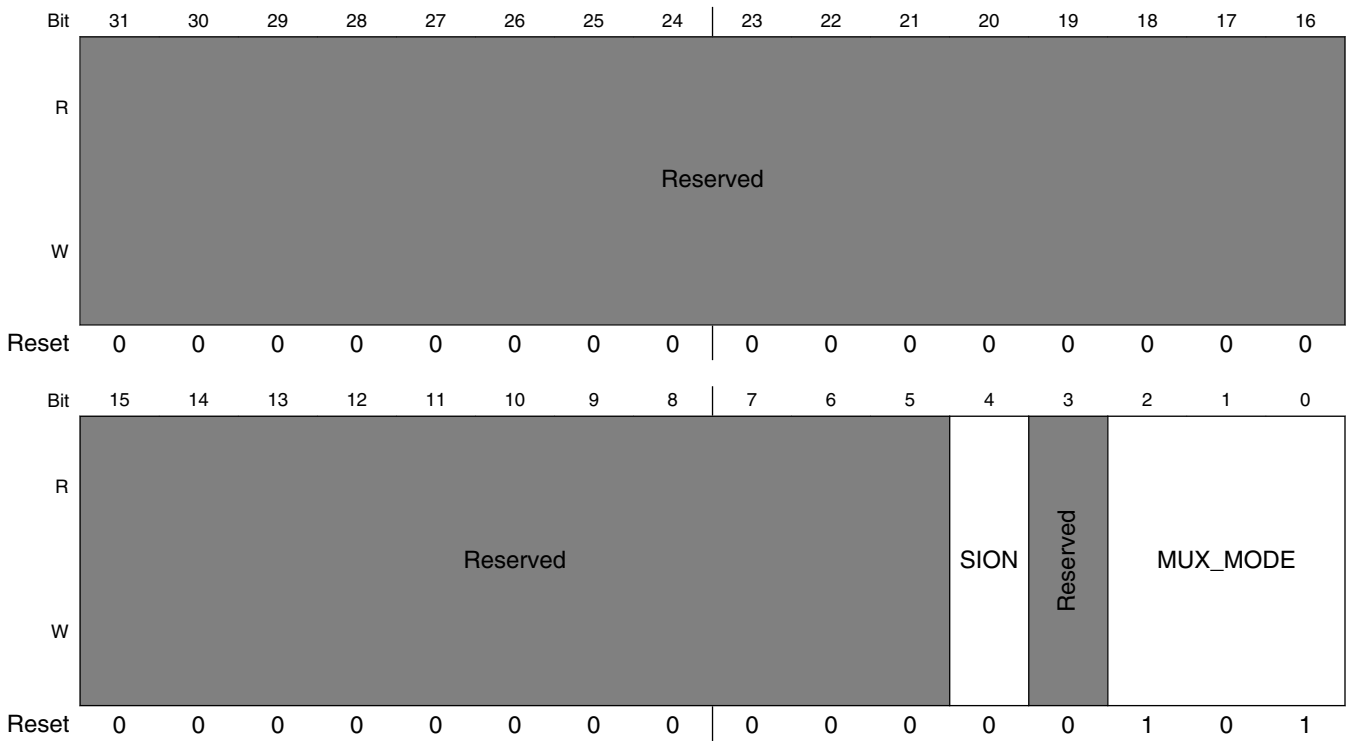
IOMUXC_SW_MUX_CTL_PAD_SPDIF_RX field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SPDIF_RX is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SPDIF_RX
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SPDIF_RX 000 **ALT0_SPDIF1_IN** — Select mux mode: ALT0 mux port: IN of instance: SPDIF1 001 **ALT1_PWM2_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM2 101 **ALT5_GPIO5_IO04** — Select mux mode: ALT5 mux port: IO04 of instance: GPIO5

8.2.5.120 SW_MUX_CTL_PAD_SPDIF_EXT_CLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F0h offset = 3033_01F0h



IOMUXC_SW_MUX_CTL_PAD_SPDIF_EXT_CLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad SPDIF_EXT_CLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad SPDIF_EXT_CLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: SPDIF_EXT_CLK 000 **ALT0_SPDIF1_EXT_CLK** — Select mux mode: ALT0 mux port: EXT_CLK of instance: SPDIF1 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM1 101 **ALT5_GPIO5_IO05** — Select mux mode: ALT5 mux port: IO05 of instance: GPIO5

8.2.5.121 SW_MUX_CTL_PAD_ECSPI1_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F4h offset = 3033_01F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI1_SCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI1_SCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI1_SCLK 000 **ALT0_ECSPI1_SCLK** — Select mux mode: ALT0 mux port: SCLK of instance: ECSPI1 001 **ALT1_UART3_RX** — Select mux mode: ALT1 mux port: RX of instance: UART3 101 **ALT5_GPIO5_IO06** — Select mux mode: ALT5 mux port: IO06 of instance: GPIO5

8.2.5.122 SW_MUX_CTL_PAD_ECSPI1_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI)

SW_MUX_CTL Register

Address: 3033_0000h base + 1F8h offset = 3033_01F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											SION	Reserved	MUX_MODE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MOSI field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI1_MOSI is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI1_MOSI
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI1_MOSI 000 **ALT0_ECSPI1_MOSI** — Select mux mode: ALT0 mux port: MOSI of instance: ECSPI1 001 **ALT1_UART3_TX** — Select mux mode: ALT1 mux port: TX of instance: UART3 101 **ALT5_GPIO5_IO07** — Select mux mode: ALT5 mux port: IO07 of instance: GPIO5

8.2.5.123 SW_MUX_CTL_PAD_ECSPI1_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO)

SW_MUX_CTL Register

Address: 3033_0000h base + 1FCh offset = 3033_01FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

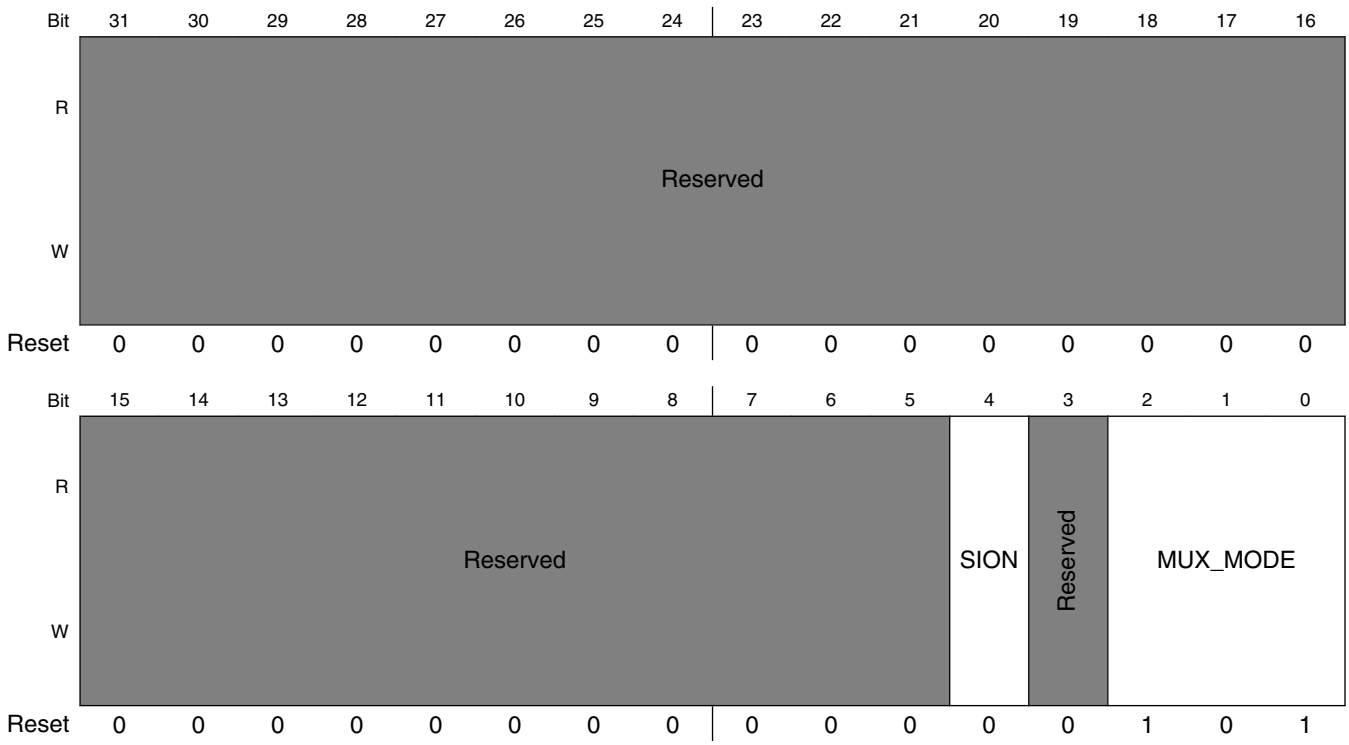
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_MISO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI1_MISO is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI1_MISO
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI1_MISO 000 **ALT0_ECSPI1_MISO** — Select mux mode: ALT0 mux port: MISO of instance: ECSPI1 001 **ALT1_UART3_CTS_B** — Select mux mode: ALT1 mux port: CTS_B of instance: UART3 101 **ALT5_GPIO5_IO08** — Select mux mode: ALT5 mux port: IO08 of instance: GPIO5

8.2.5.124 SW_MUX_CTL_PAD_ECSPI1_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0)

SW_MUX_CTL Register

Address: 3033_0000h base + 200h offset = 3033_0200h



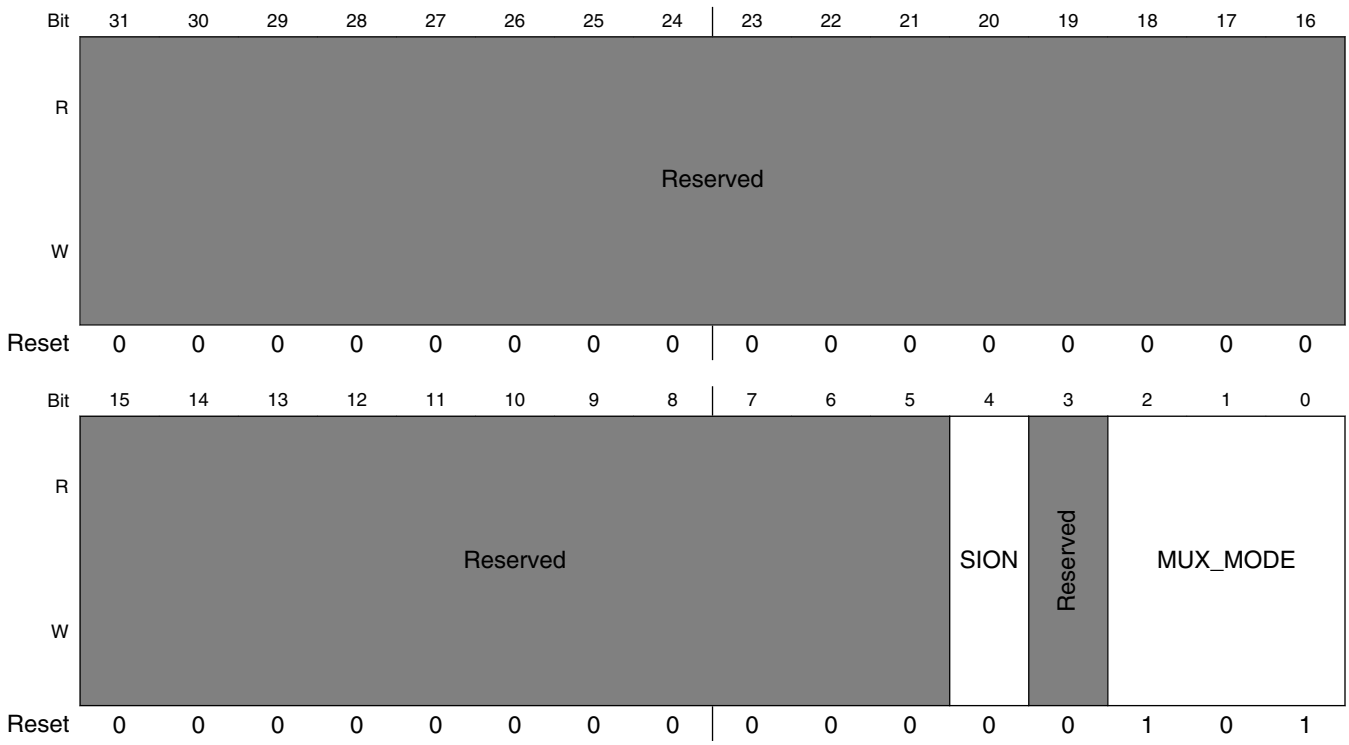
IOMUXC_SW_MUX_CTL_PAD_ECSPI1_SS0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI1_SS0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI1_SS0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI1_SS0 000 **ALT0_ECSPI1_SS0** — Select mux mode: ALT0 mux port: SS0 of instance: ECSPI1 001 **ALT1_UART3_RTS_B** — Select mux mode: ALT1 mux port: RTS_B of instance: UART3 101 **ALT5_GPIO5_IO09** — Select mux mode: ALT5 mux port: IO09 of instance: GPIO5

8.2.5.125 SW_MUX_CTL_PAD_ECSPI2_SCLK SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK)

SW_MUX_CTL Register

Address: 3033_0000h base + 204h offset = 3033_0204h



IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SCLK field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI2_SCLK is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI2_SCLK
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI2_SCLK 000 **ALT0_ECSPI2_SCLK** — Select mux mode: ALT0 mux port: SCLK of instance: ECSPI2 001 **ALT1_UART4_RX** — Select mux mode: ALT1 mux port: RX of instance: UART4 101 **ALT5_GPIO5_IO10** — Select mux mode: ALT5 mux port: IO10 of instance: GPIO5

8.2.5.126 SW_MUX_CTL_PAD_ECSPI2_MOSI SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI)

SW_MUX_CTL Register

Address: 3033_0000h base + 208h offset = 3033_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MOSI field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI2_MOSI is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI2_MOSI
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI2_MOSI 000 **ALT0_ECSPI2_MOSI** — Select mux mode: ALT0 mux port: MOSI of instance: ECSPI2 001 **ALT1_UART4_TX** — Select mux mode: ALT1 mux port: TX of instance: UART4 101 **ALT5_GPIO5_IO11** — Select mux mode: ALT5 mux port: IO11 of instance: GPIO5

8.2.5.127 SW_MUX_CTL_PAD_ECSPI2_MISO SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO)

SW_MUX_CTL Register

Address: 3033_0000h base + 20Ch offset = 3033_020Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

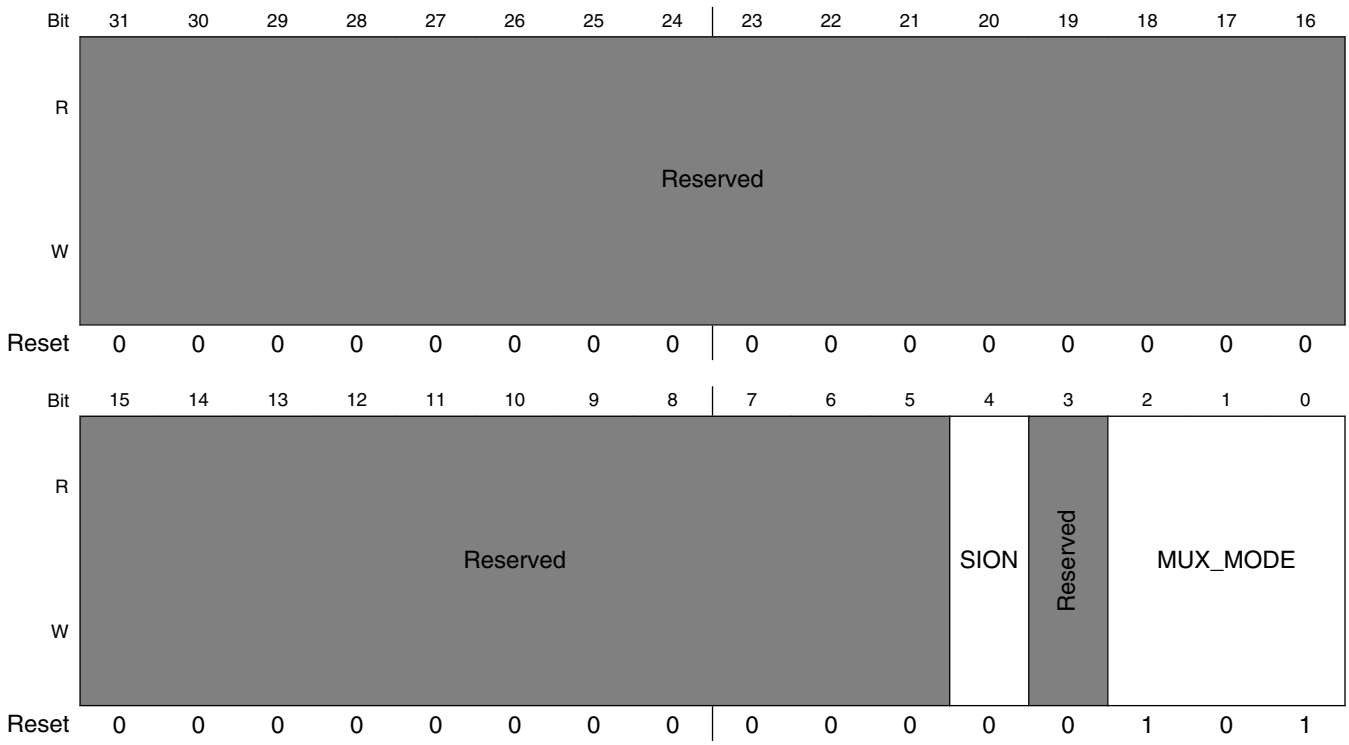
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_MISO field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI2_MISO is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI2_MISO
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI2_MISO 000 **ALT0_ECSPI2_MISO** — Select mux mode: ALT0 mux port: MISO of instance: ECSPI2 001 **ALT1_UART4_CTS_B** — Select mux mode: ALT1 mux port: CTS_B of instance: UART4 101 **ALT5_GPIO5_IO12** — Select mux mode: ALT5 mux port: IO12 of instance: GPIO5

8.2.5.128 SW_MUX_CTL_PAD_ECSPI2_SS0 SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0)

SW_MUX_CTL Register

Address: 3033_0000h base + 210h offset = 3033_0210h



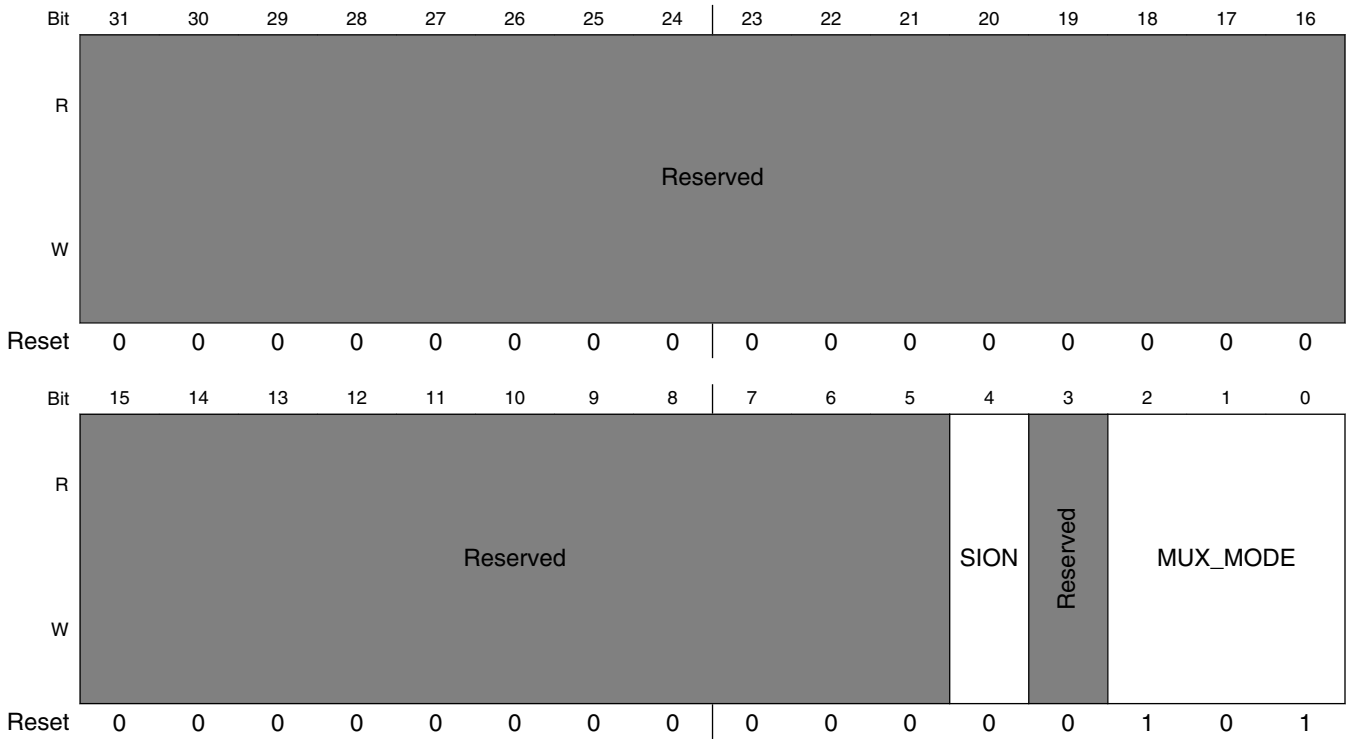
IOMUXC_SW_MUX_CTL_PAD_ECSPI2_SS0 field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad ECSPI2_SS0 is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad ECSPI2_SS0
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: ECSPI2_SS0 000 **ALT0_ECSPI2_SS0** — Select mux mode: ALT0 mux port: SS0 of instance: ECSPI2 001 **ALT1_UART4_RTS_B** — Select mux mode: ALT1 mux port: RTS_B of instance: UART4 101 **ALT5_GPIO5_IO13** — Select mux mode: ALT5 mux port: IO13 of instance: GPIO5

8.2.5.129 SW_MUX_CTL_PAD_I2C1_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 214h offset = 3033_0214h



IOMUXC_SW_MUX_CTL_PAD_I2C1_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C1_SCL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C1_SCL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: I2C1_SCL 000 **ALT0_I2C1_SCL** — Select mux mode: ALT0 mux port: SCL of instance: I2C1 001 **ALT1_ENET1_MDC** — Select mux mode: ALT1 mux port: MDC of instance: ENET1 101 **ALT5_GPIO5_IO14** — Select mux mode: ALT5 mux port: IO14 of instance: GPIO5

8.2.5.130 SW_MUX_CTL_PAD_I2C1_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 218h offset = 3033_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

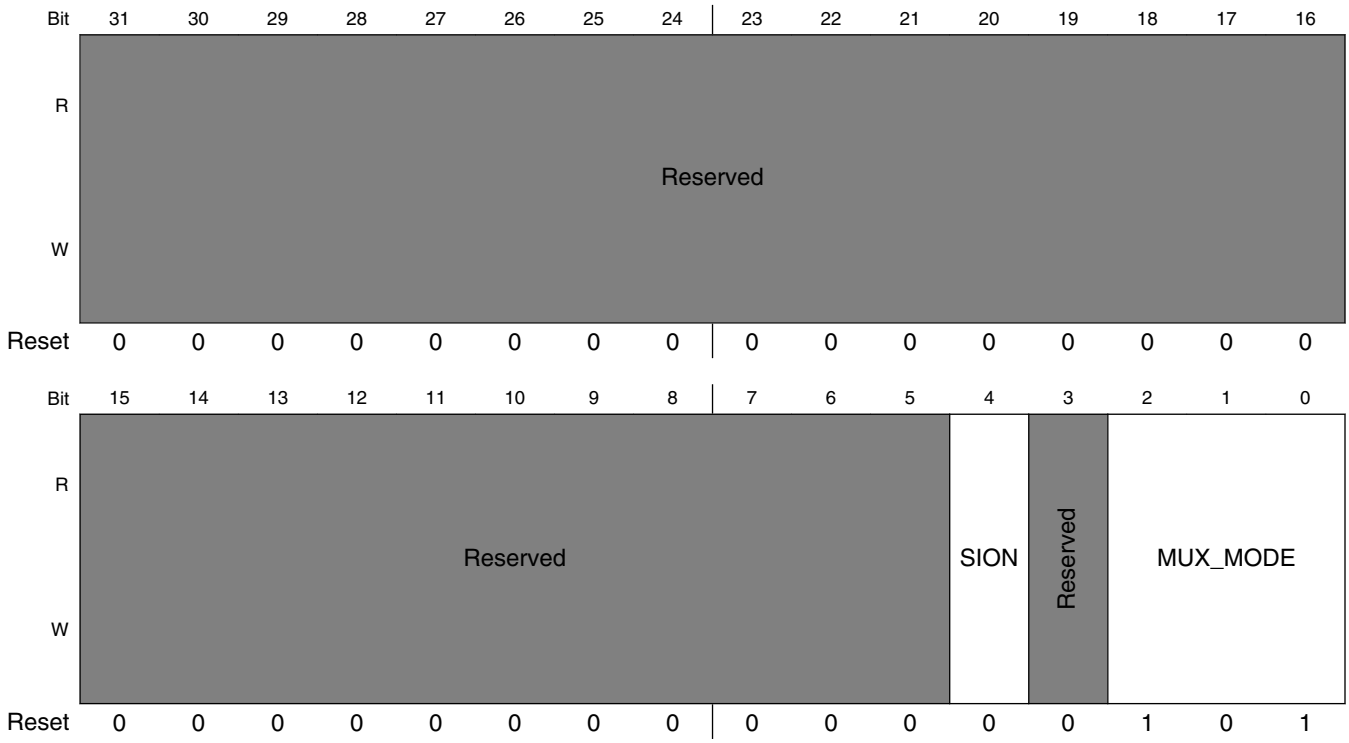
IOMUXC_SW_MUX_CTL_PAD_I2C1_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C1_SDA is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C1_SDA
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: I2C1_SDA 000 **ALT0_I2C1_SDA** — Select mux mode: ALT0 mux port: SDA of instance: I2C1 001 **ALT1_ENET1_MDIO** — Select mux mode: ALT1 mux port: MDIO of instance: ENET1 101 **ALT5_GPIO5_IO15** — Select mux mode: ALT5 mux port: IO15 of instance: GPIO5

8.2.5.131 SW_MUX_CTL_PAD_I2C2_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 21Ch offset = 3033_021Ch



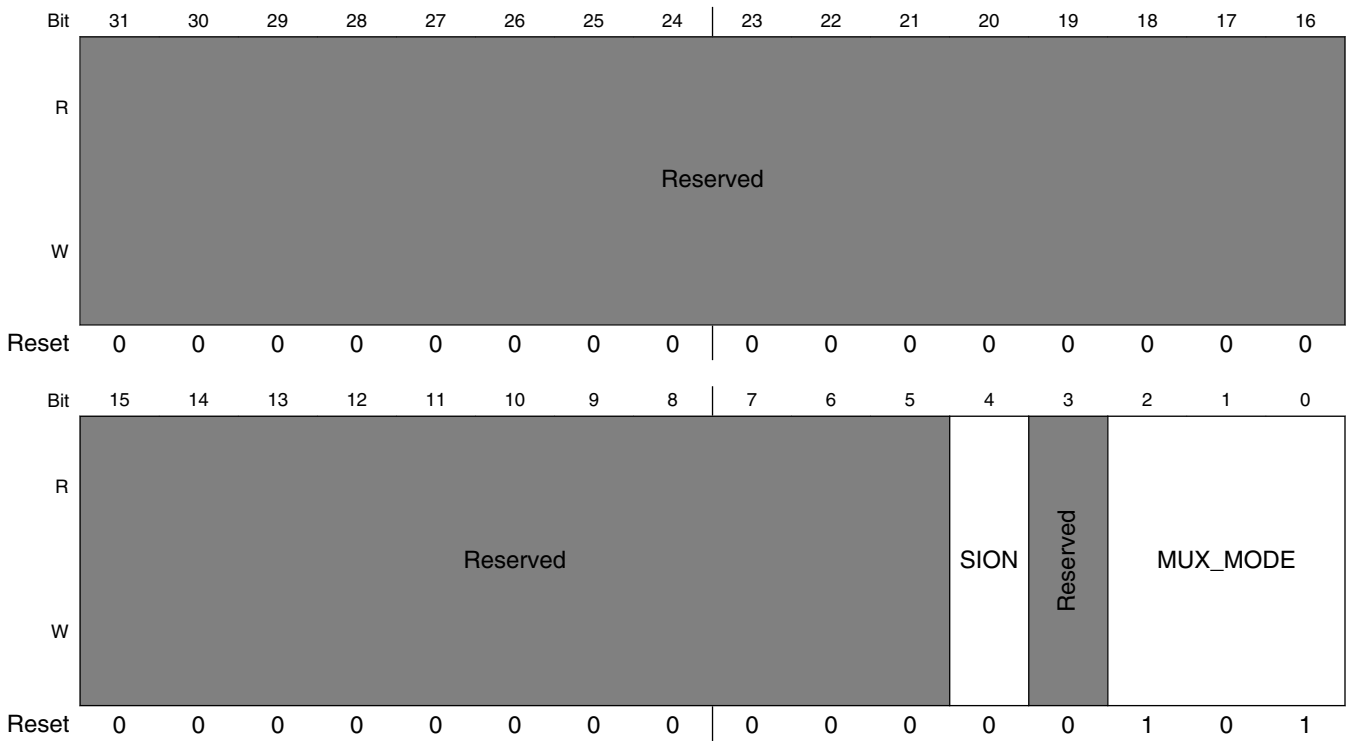
IOMUXC_SW_MUX_CTL_PAD_I2C2_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C2_SCL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C2_SCL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: I2C2_SCL 000 **ALT0_I2C2_SCL** — Select mux mode: ALT0 mux port: SCL of instance: I2C2 001 **ALT1_ENET1_1588_EVENT1_IN** — Select mux mode: ALT1 mux port: 1588_EVENT1_IN of instance: ENET1 101 **ALT5_GPIO5_IO16** — Select mux mode: ALT5 mux port: IO16 of instance: GPIO5

8.2.5.132 SW_MUX_CTL_PAD_I2C2_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 220h offset = 3033_0220h



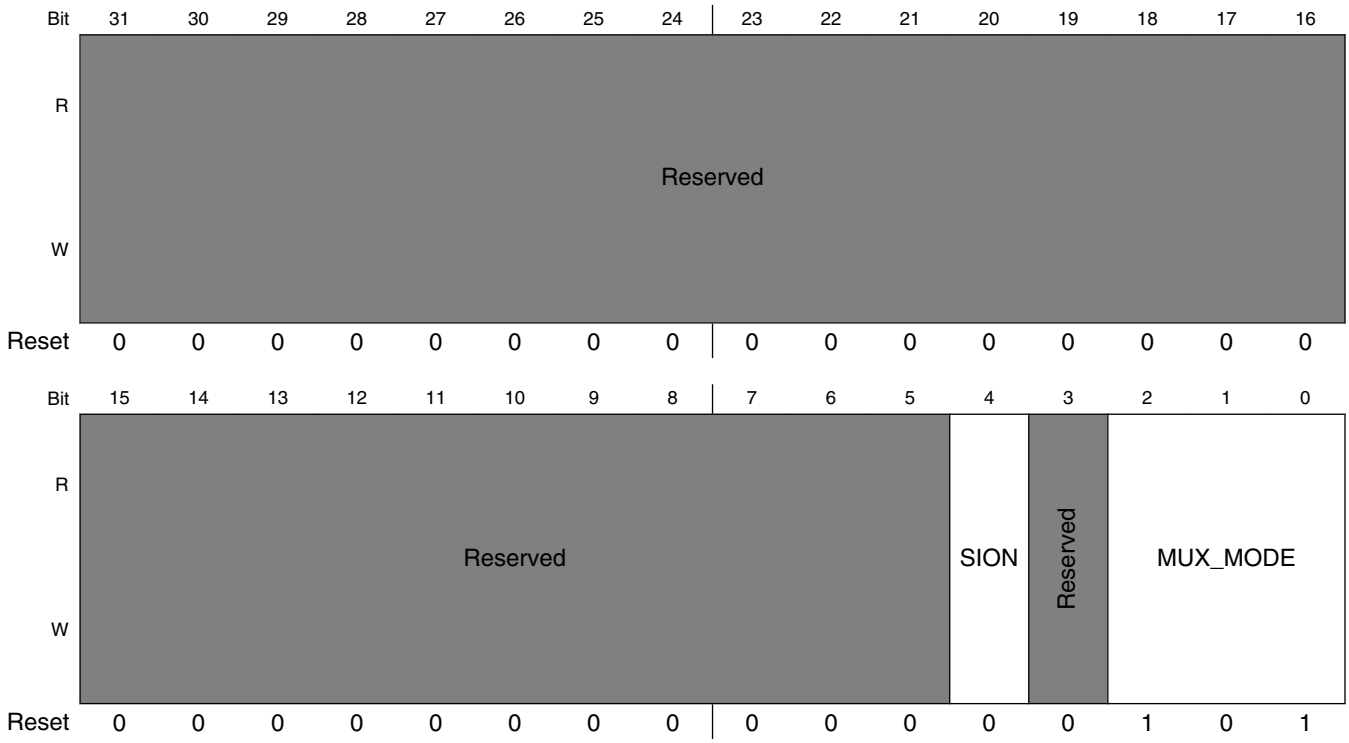
IOMUXC_SW_MUX_CTL_PAD_I2C2_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C2_SDA is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C2_SDA
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: I2C2_SDA 000 **ALT0_I2C2_SDA** — Select mux mode: ALT0 mux port: SDA of instance: I2C2 001 **ALT1_ENET1_1588_EVENT1_OUT** — Select mux mode: ALT1 mux port: 1588_EVENT1_OUT of instance: ENET1 101 **ALT5_GPIO5_IO17** — Select mux mode: ALT5 mux port: IO17 of instance: GPIO5

8.2.5.133 SW_MUX_CTL_PAD_I2C3_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 224h offset = 3033_0224h



IOMUXC_SW_MUX_CTL_PAD_I2C3_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C3_SCL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C3_SCL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: I2C3_SCL 000 **ALT0_I2C3_SCL** — Select mux mode: ALT0 mux port: SCL of instance: I2C3 001 **ALT1_PWM4_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM4 010 **ALT2_GPT2_CLK** — Select mux mode: ALT2 mux port: CLK of instance: GPT2 101 **ALT5_GPIO5_IO18** — Select mux mode: ALT5 mux port: IO18 of instance: GPIO5

8.2.5.134 SW_MUX_CTL_PAD_I2C3_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 228h offset = 3033_0228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

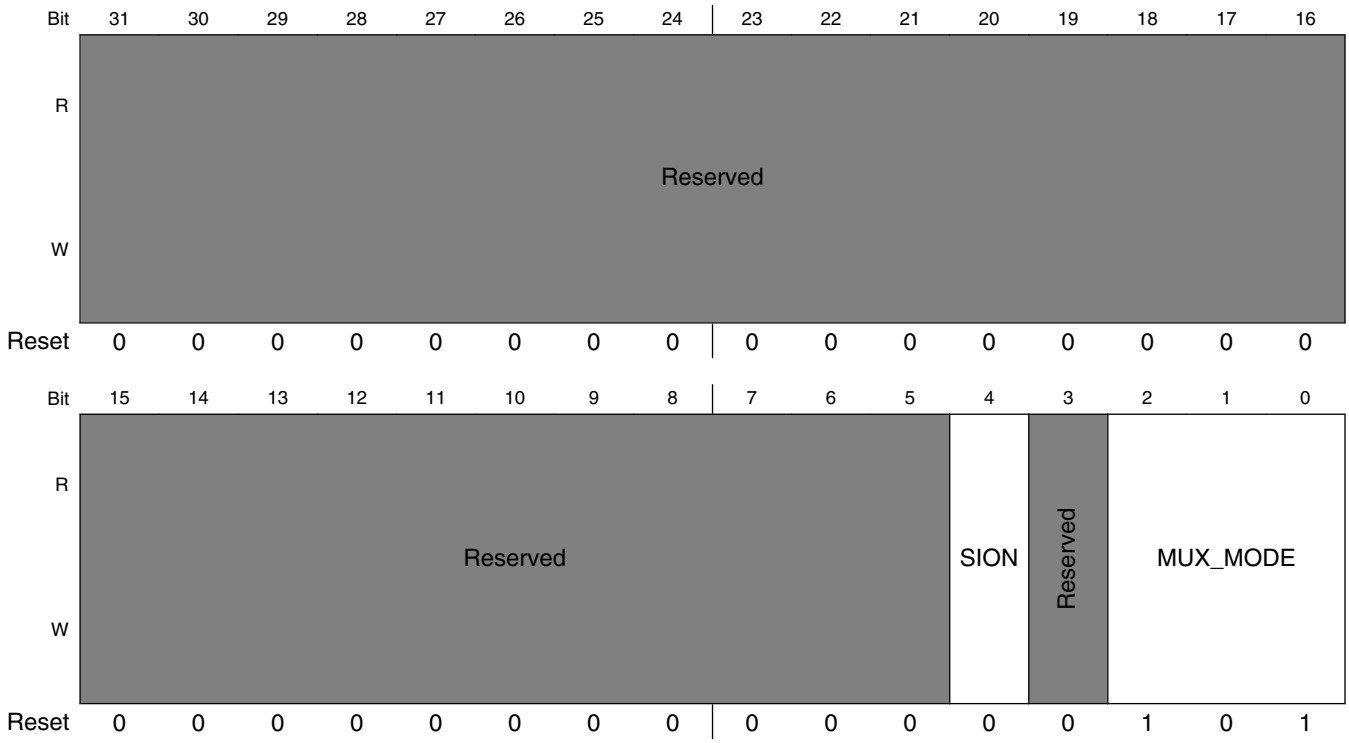
IOMUXC_SW_MUX_CTL_PAD_I2C3_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C3_SDA is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C3_SDA
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: I2C3_SDA 000 **ALT0_I2C3_SDA** — Select mux mode: ALT0 mux port: SDA of instance: I2C3 001 **ALT1_PWM3_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM3 010 **ALT2_GPT3_CLK** — Select mux mode: ALT2 mux port: CLK of instance: GPT3 101 **ALT5_GPIO5_IO19** — Select mux mode: ALT5 mux port: IO19 of instance: GPIO5

8.2.5.135 SW_MUX_CTL_PAD_I2C4_SCL SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL)

SW_MUX_CTL Register

Address: 3033_0000h base + 22Ch offset = 3033_022Ch



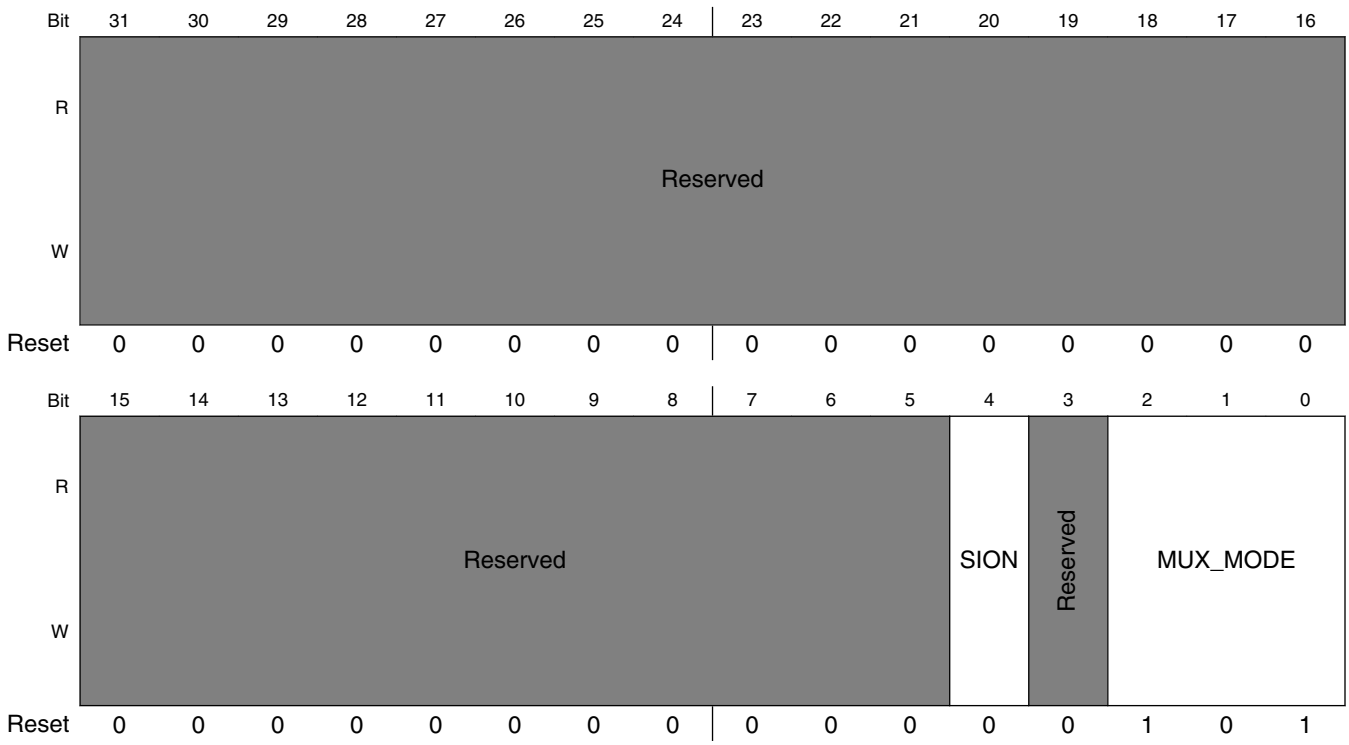
IOMUXC_SW_MUX_CTL_PAD_I2C4_SCL field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C4_SCL is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C4_SCL
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: I2C4_SCL 000 **ALT0_I2C4_SCL** — Select mux mode: ALT0 mux port: SCL of instance: I2C4 001 **ALT1_PWM2_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM2 010 **ALT2_PCIE1_CLKREQ_B** — Select mux mode: ALT2 mux port: CLKREQ_B of instance: PCIE1 101 **ALT5_GPIO5_IO20** — Select mux mode: ALT5 mux port: IO20 of instance: GPIO5

8.2.5.136 SW_MUX_CTL_PAD_I2C4_SDA SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA)

SW_MUX_CTL Register

Address: 3033_0000h base + 230h offset = 3033_0230h



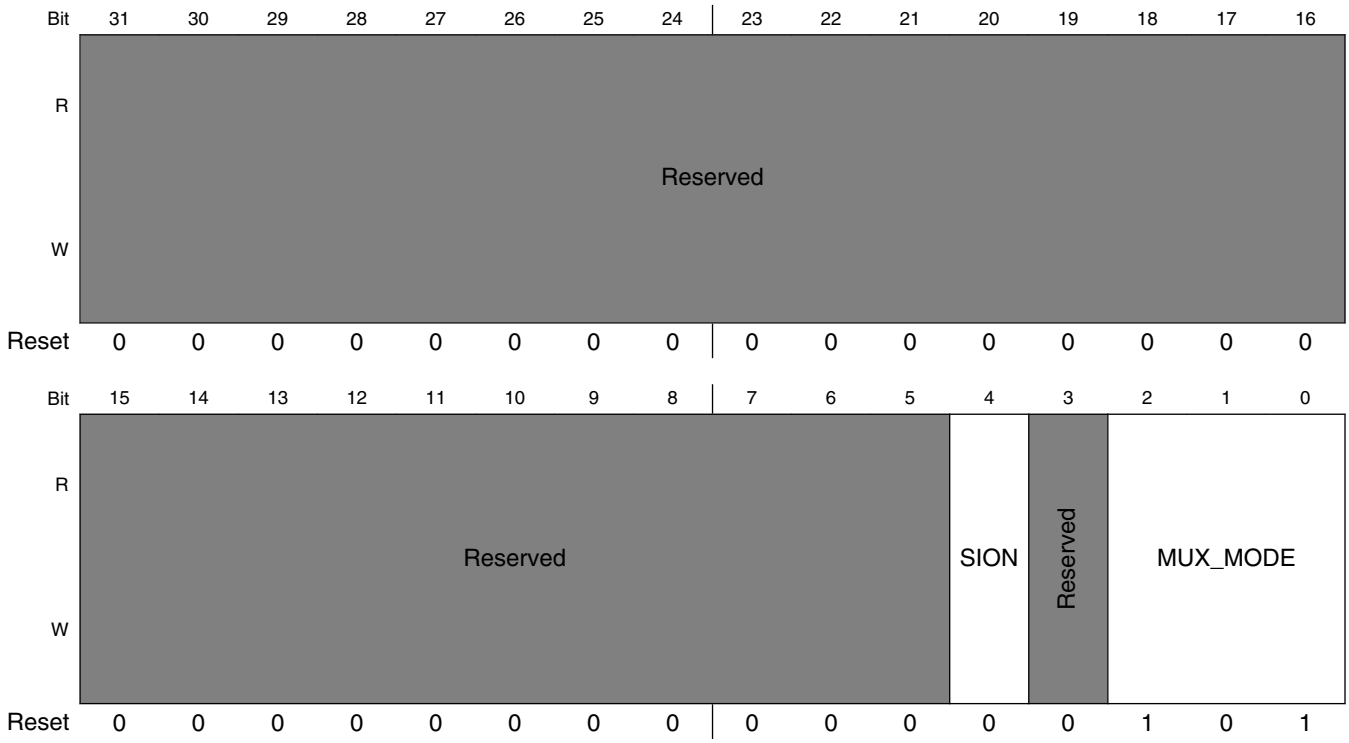
IOMUXC_SW_MUX_CTL_PAD_I2C4_SDA field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad I2C4_SDA is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad I2C4_SDA
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: I2C4_SDA 000 **ALT0_I2C4_SDA** — Select mux mode: ALT0 mux port: SDA of instance: I2C4 001 **ALT1_PWM1_OUT** — Select mux mode: ALT1 mux port: OUT of instance: PWM1 010 **ALT2_PCIE2_CLKREQ_B** — Select mux mode: ALT2 mux port: CLKREQ_B of instance: PCIE2 101 **ALT5_GPIO5_IO21** — Select mux mode: ALT5 mux port: IO21 of instance: GPIO5

8.2.5.137 SW_MUX_CTL_PAD_UART1_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 234h offset = 3033_0234h



IOMUXC_SW_MUX_CTL_PAD_UART1_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART1_RXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART1_RXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART1_RXD 000 **ALT0_UART1_RX** — Select mux mode: ALT0 mux port: RX of instance: UART1 001 **ALT1_ECSPI3_SCLK** — Select mux mode: ALT1 mux port: SCLK of instance: ECSPI3 101 **ALT5_GPIO5_IO22** — Select mux mode: ALT5 mux port: IO22 of instance: GPIO5

8.2.5.138 SW_MUX_CTL_PAD_UART1_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART1_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 238h offset = 3033_0238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											SION	Reserved	MUX_MODE		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

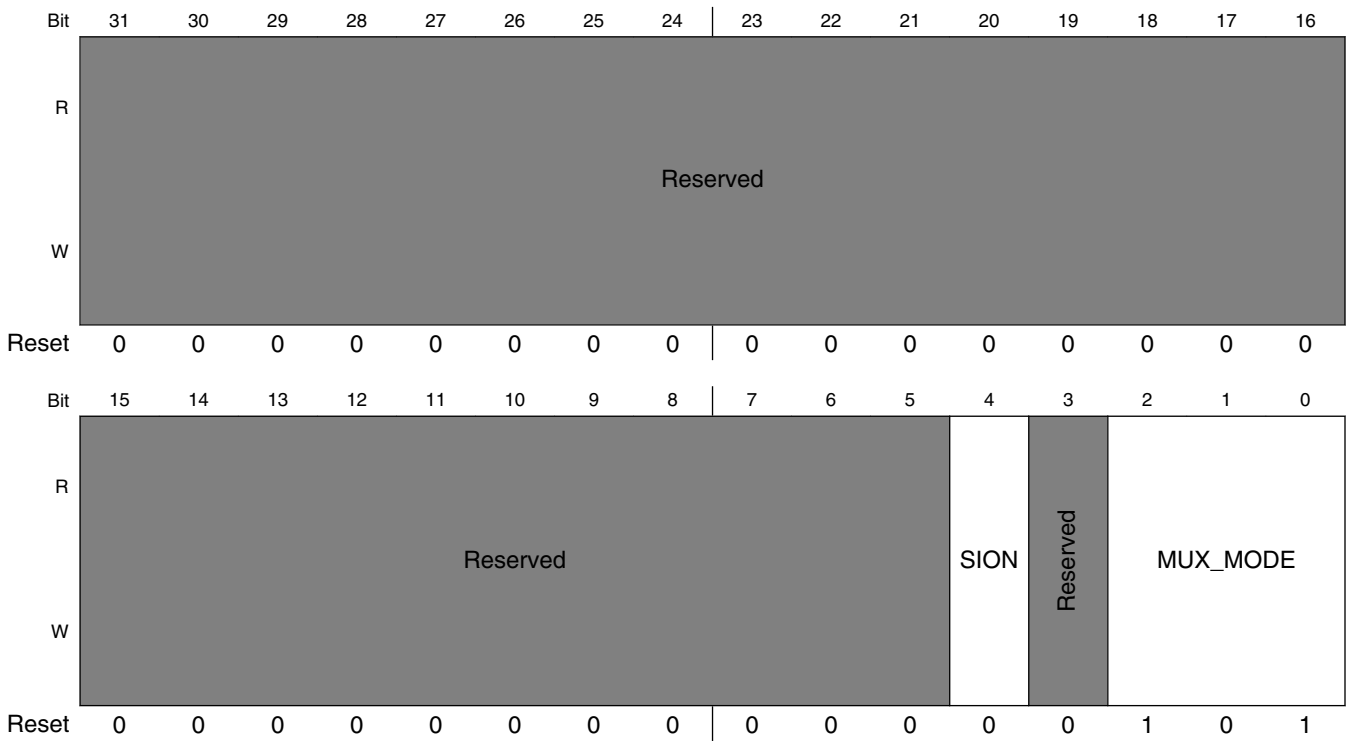
IOMUXC_SW_MUX_CTL_PAD_UART1_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART1_TXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART1_TXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART1_TXD 000 **ALT0_UART1_TX** — Select mux mode: ALT0 mux port: TX of instance: UART1 001 **ALT1_ECSPI3_MOSI** — Select mux mode: ALT1 mux port: MOSI of instance: ECSPI3 101 **ALT5_GPIO5_IO23** — Select mux mode: ALT5 mux port: IO23 of instance: GPIO5

8.2.5.139 SW_MUX_CTL_PAD_UART2_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 23Ch offset = 3033_023Ch



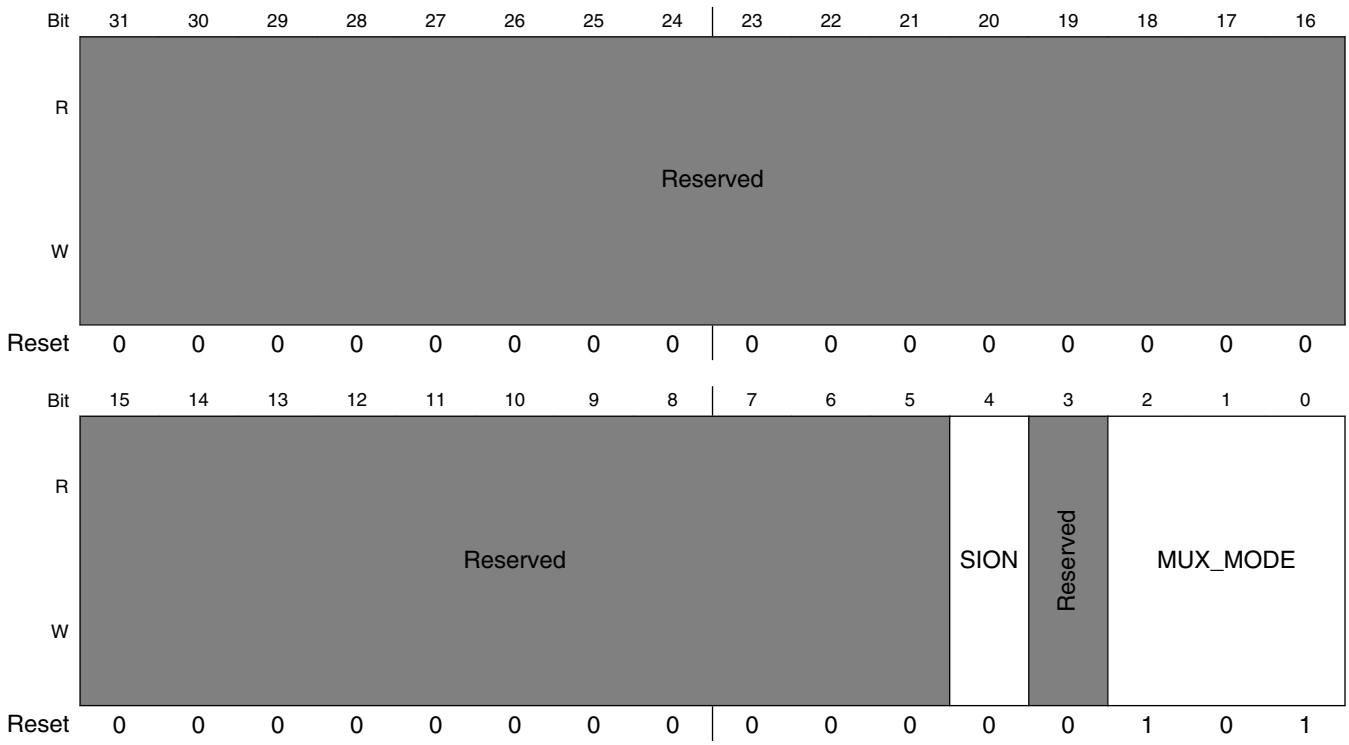
IOMUXC_SW_MUX_CTL_PAD_UART2_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART2_RXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART2_RXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART2_RXD 000 **ALT0_UART2_RX** — Select mux mode: ALT0 mux port: RX of instance: UART2 001 **ALT1_ECSPI3_MISO** — Select mux mode: ALT1 mux port: MISO of instance: ECSPI3 101 **ALT5_GPIO5_IO24** — Select mux mode: ALT5 mux port: IO24 of instance: GPIO5

8.2.5.140 SW_MUX_CTL_PAD_UART2_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART2_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 240h offset = 3033_0240h



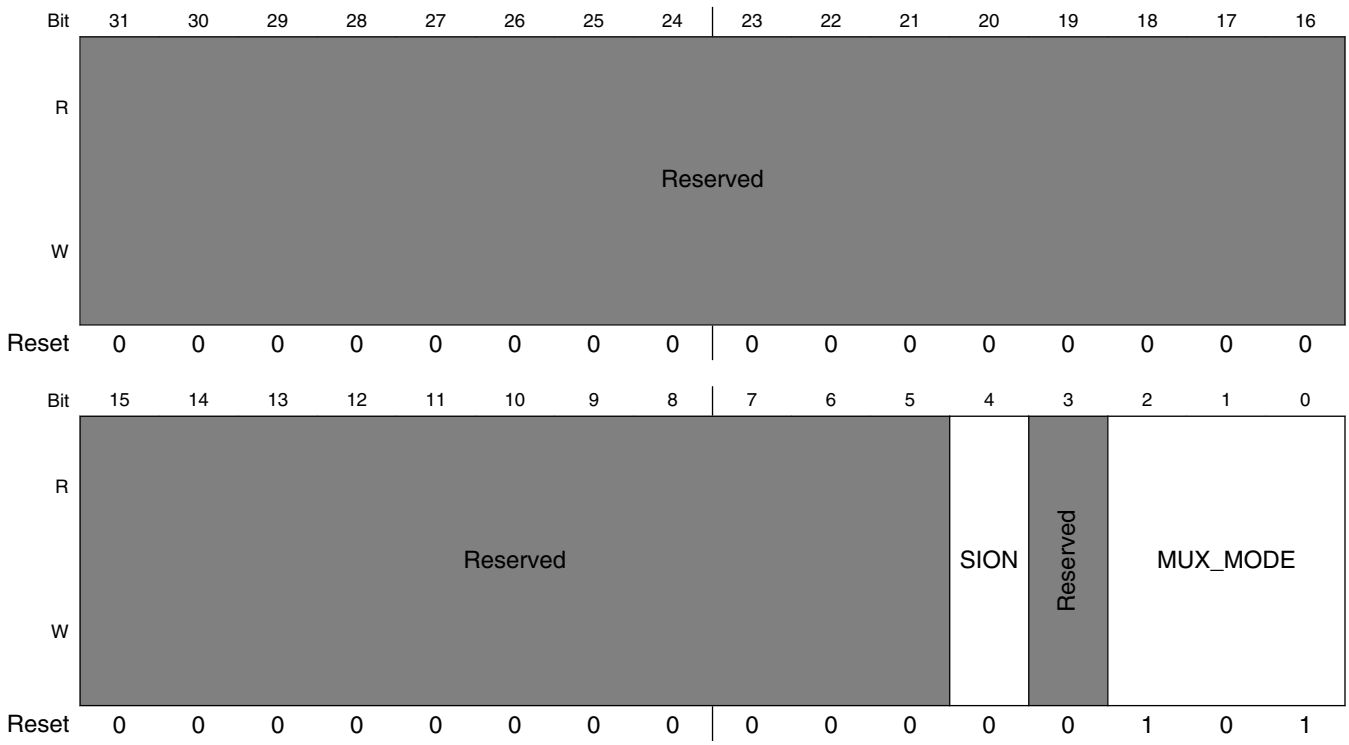
IOMUXC_SW_MUX_CTL_PAD_UART2_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART2_TXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART2_TXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART2_TXD 000 **ALT0_UART2_TX** — Select mux mode: ALT0 mux port: TX of instance: UART2 001 **ALT1_ECSPI3_SS0** — Select mux mode: ALT1 mux port: SS0 of instance: ECSPI3 101 **ALT5_GPIO5_IO25** — Select mux mode: ALT5 mux port: IO25 of instance: GPIO5

8.2.5.141 SW_MUX_CTL_PAD_UART3_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 244h offset = 3033_0244h



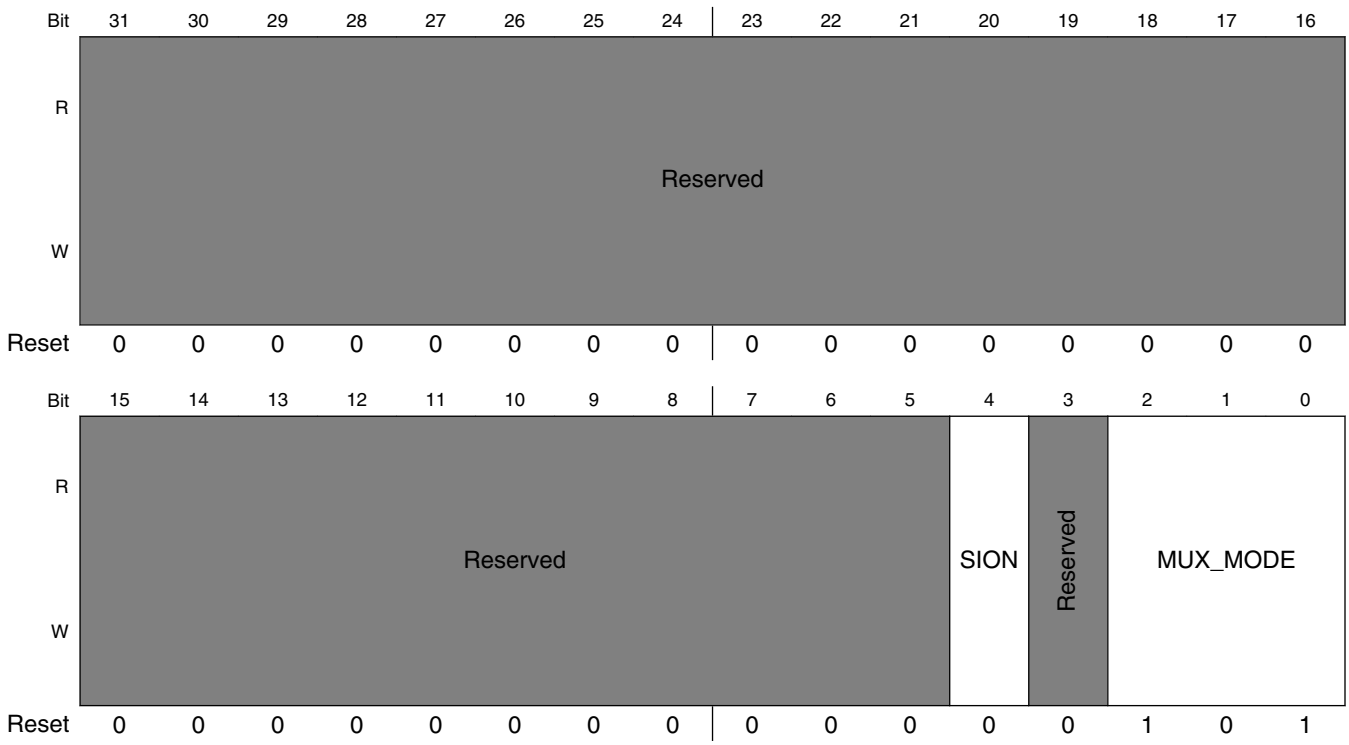
IOMUXC_SW_MUX_CTL_PAD_UART3_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART3_RXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART3_RXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART3_RXD 000 **ALT0_UART3_RX** — Select mux mode: ALT0 mux port: RX of instance: UART3 001 **ALT1_UART1_CTS_B** — Select mux mode: ALT1 mux port: CTS_B of instance: UART1 101 **ALT5_GPIO5_IO26** — Select mux mode: ALT5 mux port: IO26 of instance: GPIO5

8.2.5.142 SW_MUX_CTL_PAD_UART3_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART3_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 248h offset = 3033_0248h



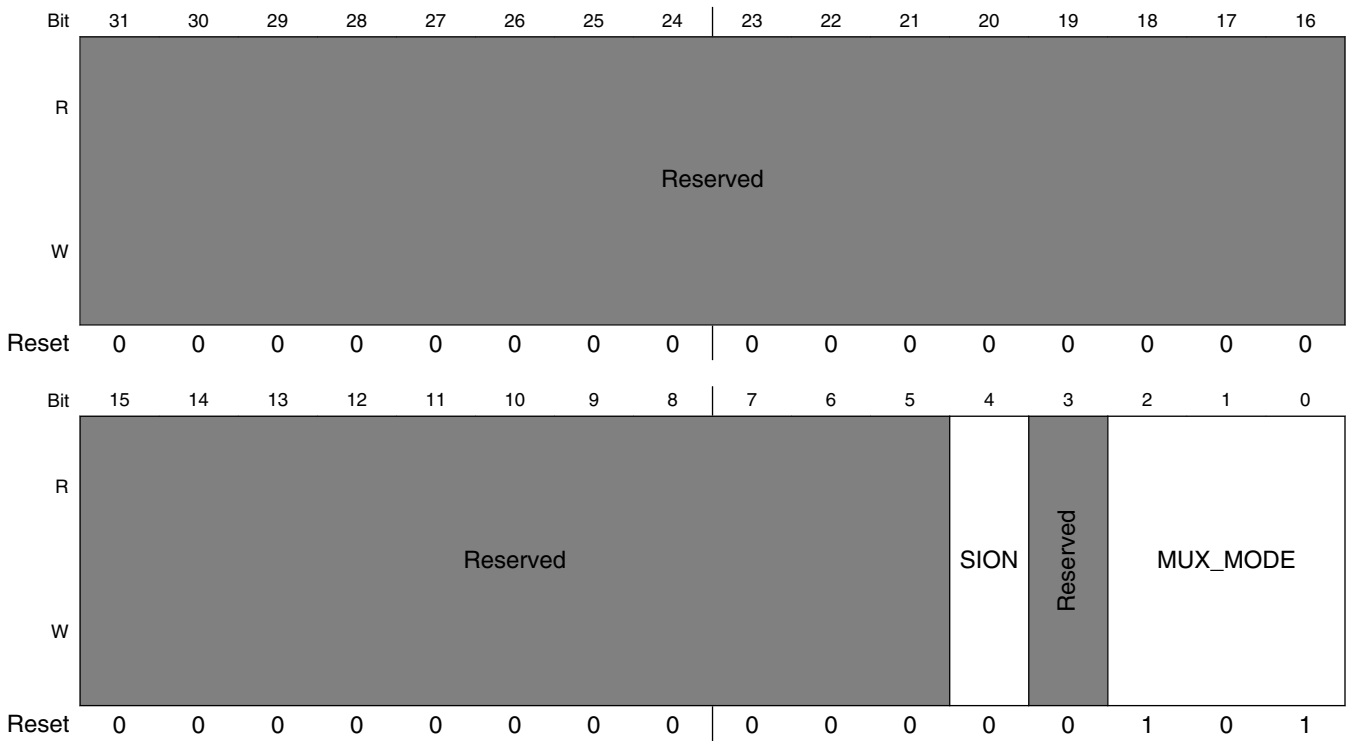
IOMUXC_SW_MUX_CTL_PAD_UART3_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART3_TXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART3_TXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 3 iomux modes to be used for pad: UART3_TXD 000 **ALT0_UART3_TX** — Select mux mode: ALT0 mux port: TX of instance: UART3 001 **ALT1_UART1_RTS_B** — Select mux mode: ALT1 mux port: RTS_B of instance: UART1 101 **ALT5_GPIO5_IO27** — Select mux mode: ALT5 mux port: IO27 of instance: GPIO5

8.2.5.143 SW_MUX_CTL_PAD_UART4_RXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_RXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 24Ch offset = 3033_024Ch



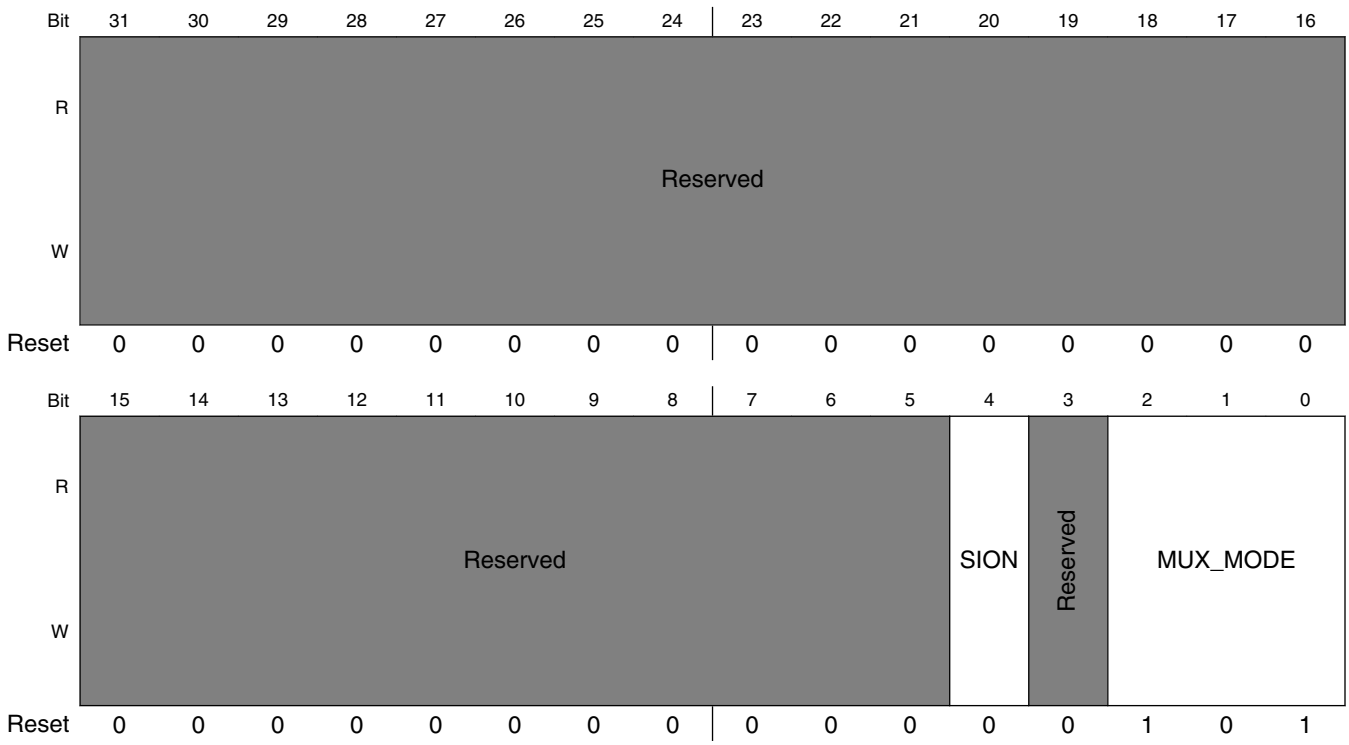
IOMUXC_SW_MUX_CTL_PAD_UART4_RXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART4_RXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART4_RXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: UART4_RXD 000 **ALT0_UART4_RX** — Select mux mode: ALT0 mux port: RX of instance: UART4 001 **ALT1_UART2_CTS_B** — Select mux mode: ALT1 mux port: CTS_B of instance: UART2 010 **ALT2_PCIE1_CLKREQ_B** — Select mux mode: ALT2 mux port: CLKREQ_B of instance: PCIE1 101 **ALT5_GPIO5_IO28** — Select mux mode: ALT5 mux port: IO28 of instance: GPIO5

8.2.5.144 SW_MUX_CTL_PAD_UART4_TXD SW MUX Control Register (IOMUXC_SW_MUX_CTL_PAD_UART4_TXD)

SW_MUX_CTL Register

Address: 3033_0000h base + 250h offset = 3033_0250h



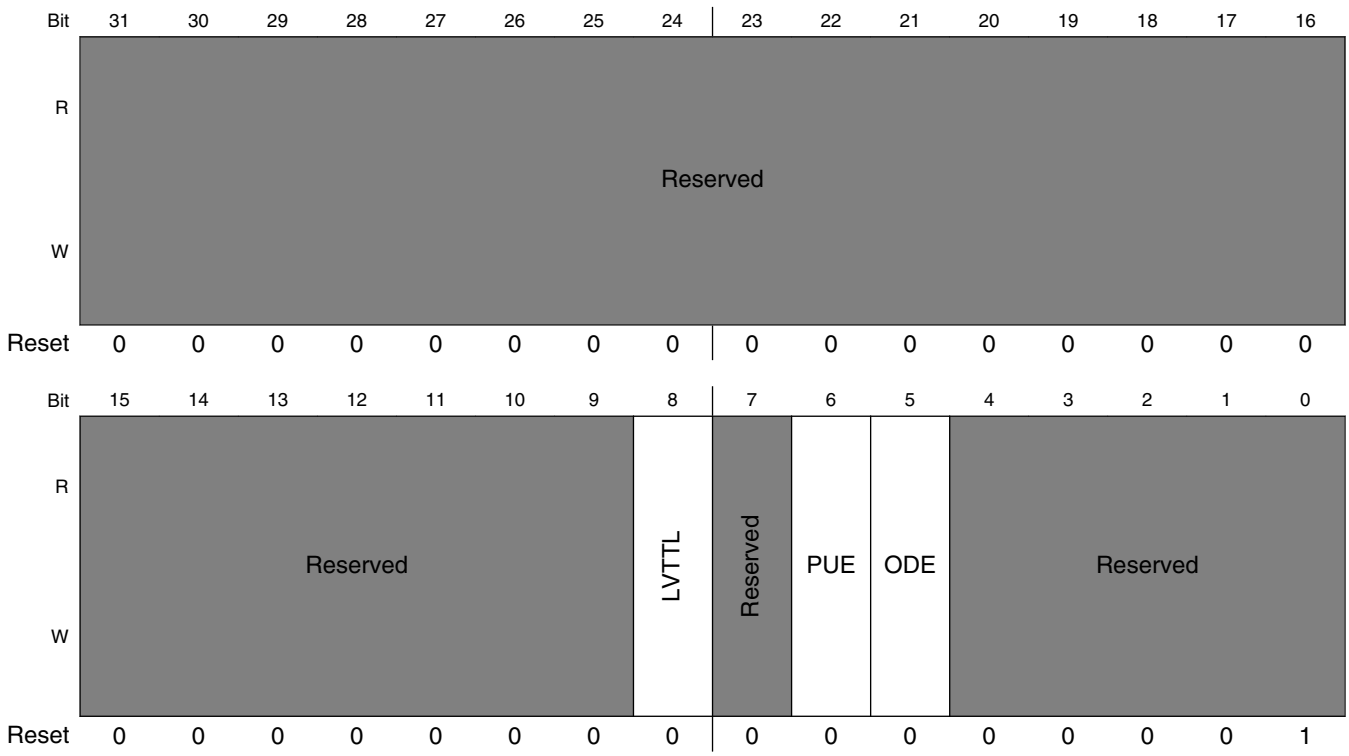
IOMUXC_SW_MUX_CTL_PAD_UART4_TXD field descriptions

Field	Description
31–5 -	This field is reserved. Reserved
4 SION	Software Input On Field Force the select mux mode Input path no matter of MUX_MODE functionality 0 **SION_DISABLED** — Input Path of pad UART4_TXD is determined by functionality 1 **SION_ENABLED** — Force Input Path of pad UART4_TXD
3 -	This field is reserved. Reserved
MUX_MODE	MUX Mode Select Field Select 1 of 4 iomux modes to be used for pad: UART4_TXD 000 **ALT0_UART4_TX** — Select mux mode: ALT0 mux port: TX of instance: UART4 001 **ALT1_UART2_RTS_B** — Select mux mode: ALT1 mux port: RTS_B of instance: UART2 010 **ALT2_PCIE2_CLKREQ_B** — Select mux mode: ALT2 mux port: CLKREQ_B of instance: PCIE2 101 **ALT5_GPIO5_IO29** — Select mux mode: ALT5 mux port: IO29 of instance: GPIO5

8.2.5.145 SW_PAD_CTL_PAD_TEST_MODE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_TEST_MODE)

SW_PAD_CTL Register

Address: 3033_0000h base + 254h offset = 3033_0254h



IOMUXC_SW_PAD_CTL_PAD_TEST_MODE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 -	This field is reserved. Reserved
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_TEST_MODE field descriptions (continued)

Field	Description
5 ODE	Open Drain Enable Field Select one out of next values for pad: TEST_MODE 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.146 SW_PAD_CTL_PAD_BOOT_MODE0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0)

SW_PAD_CTL Register

Address: 3033_0000h base + 258h offset = 3033_0258h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	Reserved	PUE	ODE	Reserved				
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field

Table continues on the next page...

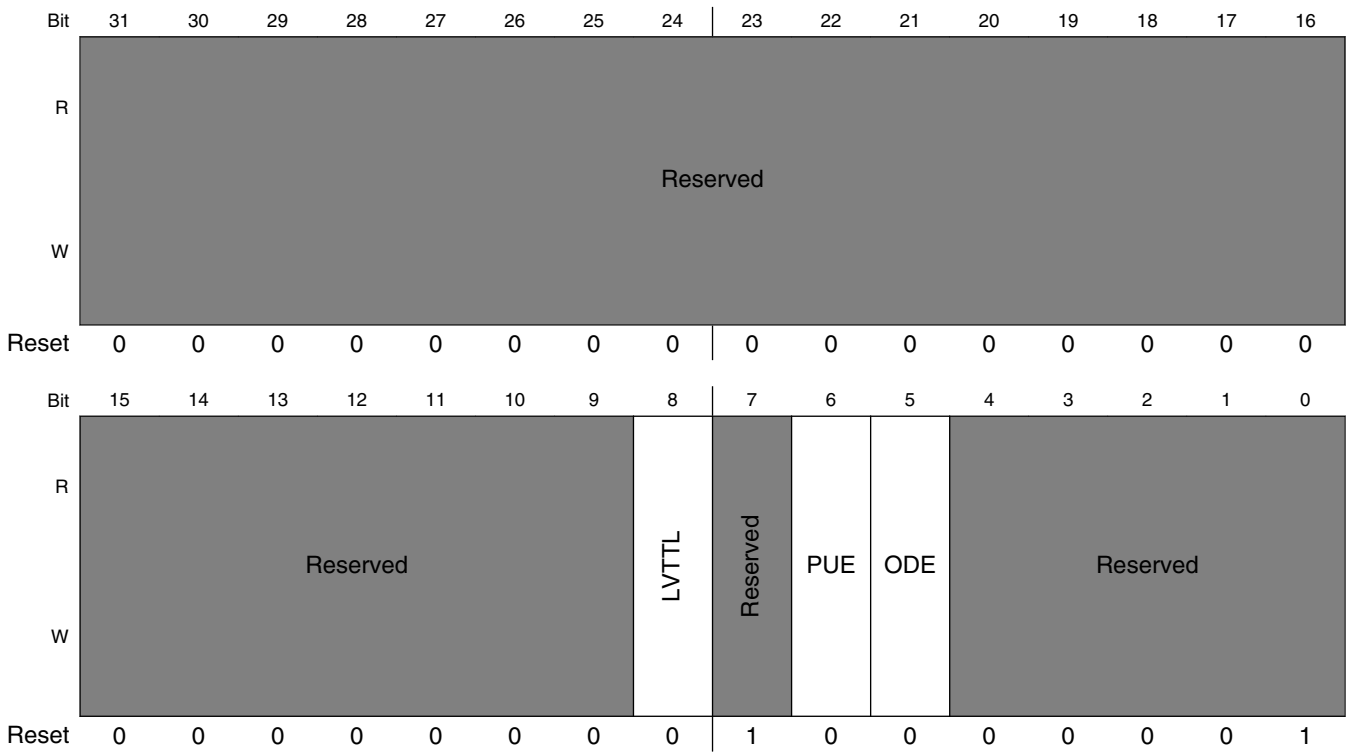
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE0 field descriptions (continued)

Field	Description
	Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 -	This field is reserved. Reserved
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: BOOT_MODE0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.147 SW_PAD_CTL_PAD_BOOT_MODE1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1)

SW_PAD_CTL Register

Address: 3033_0000h base + 25Ch offset = 3033_025Ch



IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 -	This field is reserved. Reserved
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled

Table continues on the next page...

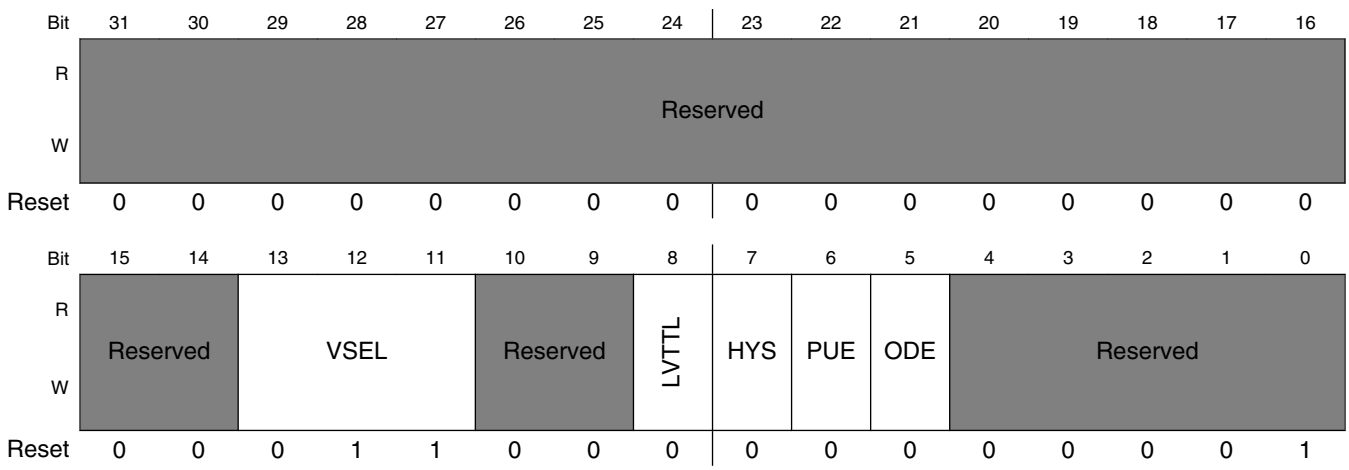
IOMUXC_SW_PAD_CTL_PAD_BOOT_MODE1 field descriptions (continued)

Field	Description
5 ODE	Open Drain Enable Field Select one out of next values for pad: BOOT_MODE1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.148 SW_PAD_CTL_PAD_JTAG_MOD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD)

SW_PAD_CTL Register

Address: 3033_0000h base + 260h offset = 3033_0260h



IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: JTAG_MOD 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_MOD field descriptions (continued)

Field	Description
	110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_MOD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.149 SW_PAD_CTL_PAD_JTAG_TRST_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TRST_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 264h offset = 3033_0264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	Reserved				
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_JTAG_TRST_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_TRST_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

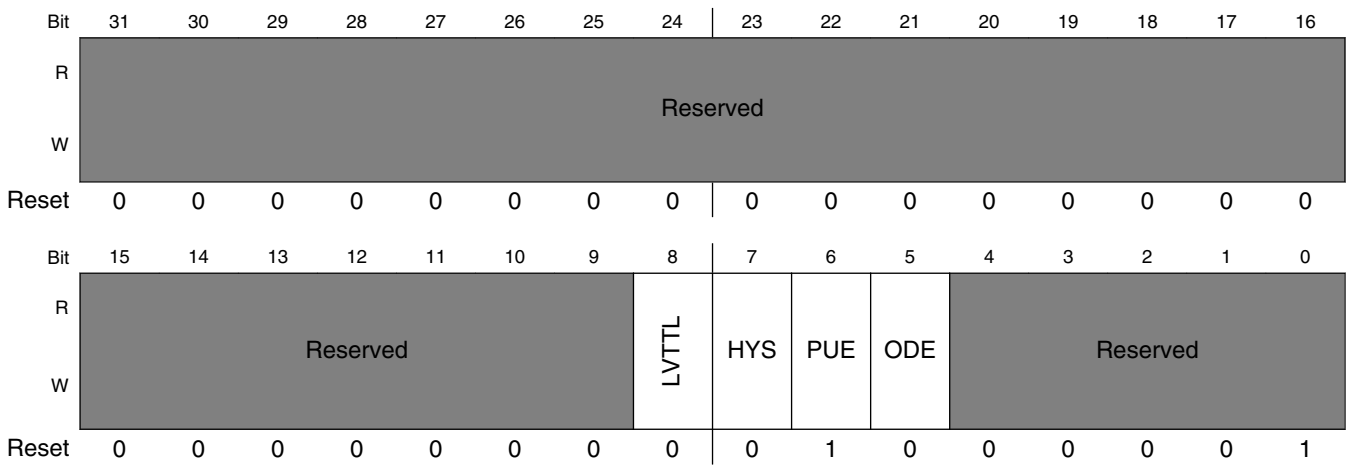
IOMUXC_SW_PAD_CTL_PAD_JTAG_TRST_B field descriptions (continued)

Field	Description
-	This field is reserved. Reserved

8.2.5.150 SW_PAD_CTL_PAD_JTAG_TDI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI)

SW_PAD_CTL Register

Address: 3033_0000h base + 268h offset = 3033_0268h



IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled

Table continues on the next page...

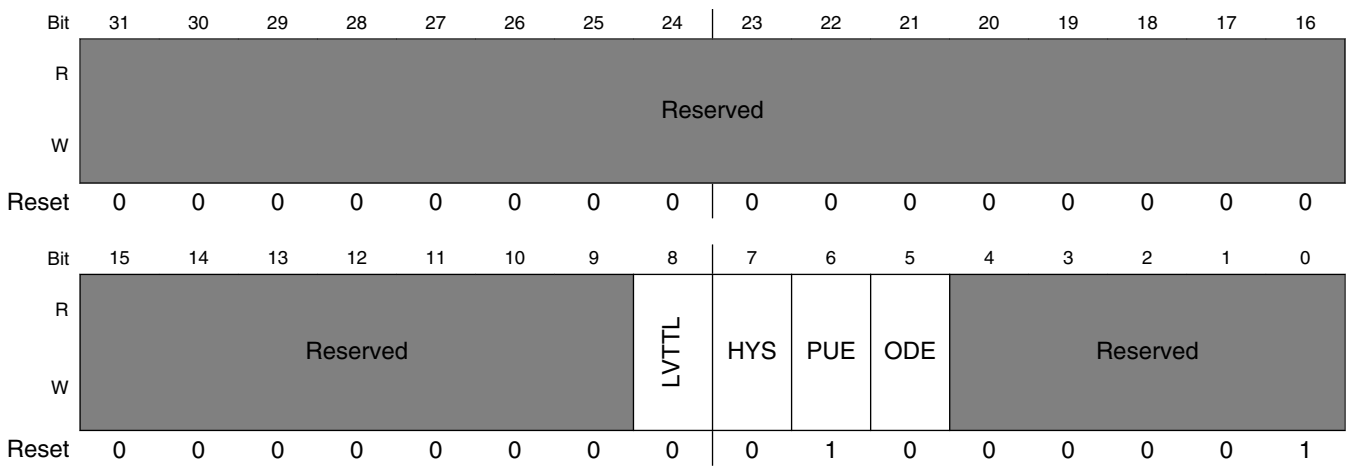
IOMUXC_SW_PAD_CTL_PAD_JTAG_TDI field descriptions (continued)

Field	Description
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_TDI 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.151 SW_PAD_CTL_PAD_JTAG_TMS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS)

SW_PAD_CTL Register

Address: 3033_0000h base + 26Ch offset = 3033_026Ch



IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TMS field descriptions (continued)

Field	Description
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_TMS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.152 SW_PAD_CTL_PAD_JTAG_TCK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK)

SW_PAD_CTL Register

Address: 3033_0000h base + 270h offset = 3033_0270h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	Reserved				
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_JTAG_TCK field descriptions (continued)

Field	Description
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_TCK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.153 SW_PAD_CTL_PAD_JTAG_TDO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO)

SW_PAD_CTL Register

Address: 3033_0000h base + 274h offset = 3033_0274h

IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved

Table continues on the next page...

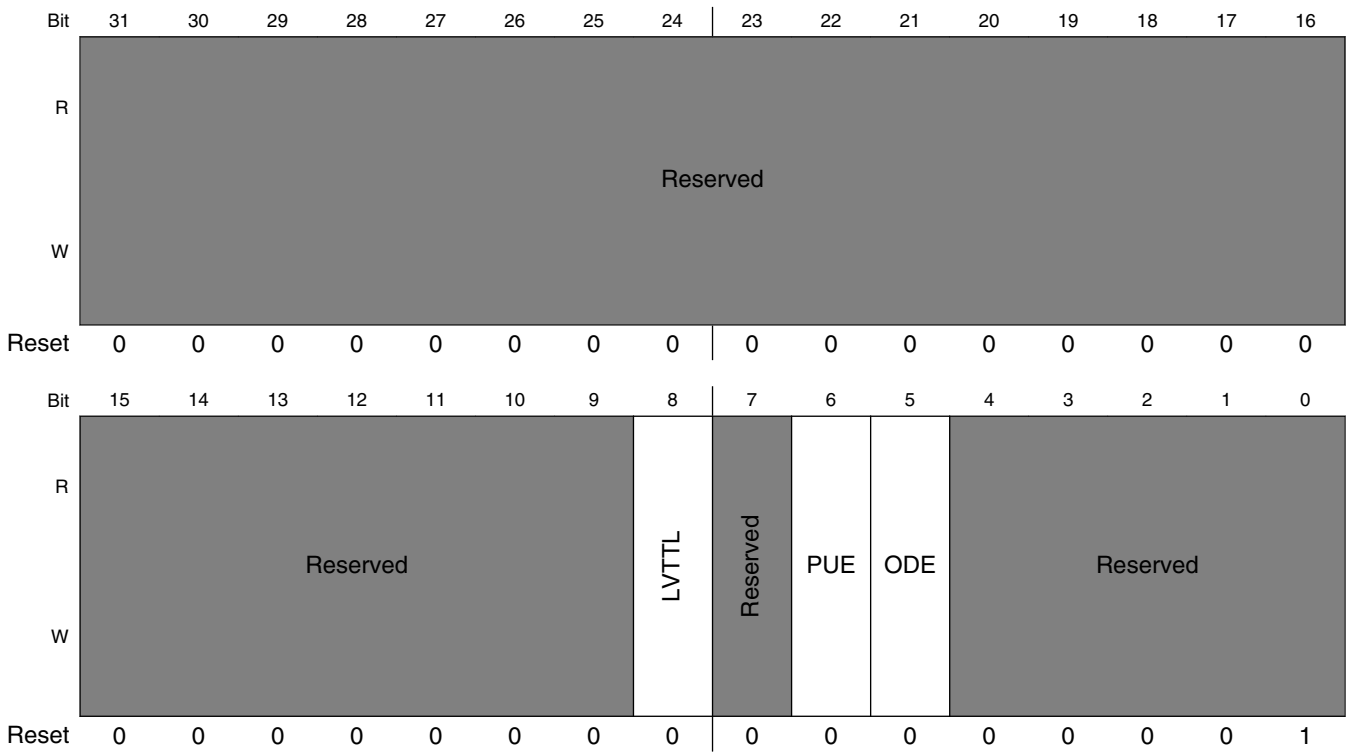
IOMUXC_SW_PAD_CTL_PAD_JTAG_TDO field descriptions (continued)

Field	Description
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: JTAG_TDO 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.154 SW_PAD_CTL_PAD_RTC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_RTC)

SW_PAD_CTL Register

Address: 3033_0000h base + 278h offset = 3033_0278h



IOMUXC_SW_PAD_CTL_PAD_RTC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 -	This field is reserved. Reserved
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RTC field descriptions (continued)

Field	Description
5 ODE	Open Drain Enable Field Select one out of next values for pad: RTC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
-	This field is reserved. Reserved

8.2.5.155 SW_PAD_CTL_PAD_PMIC_STBY_REQ SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ)

SW_PAD_CTL Register

Address: 3033_0000h base + 27Ch offset = 3033_027Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PMIC_STBY_REQ field descriptions (continued)

Field	Description
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: PMIC_STBY_REQ 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: PMIC_STBY_REQ 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: PMIC_STBY_REQ 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.156 SW_PAD_CTL_PAD_PMIC_ON_REQ SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ)

SW_PAD_CTL Register

Address: 3033_0000h base + 280h offset = 3033_0280h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	1	1	0	1	1	0	0

IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: PMIC_ON_REQ 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_PMIC_ON_REQ field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: PMIC_ON_REQ 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: PMIC_ON_REQ 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: PMIC_ON_REQ 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.157 SW_PAD_CTL_PAD_ONOFF SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ONOFF)

SW_PAD_CTL Register

Address: 3033_0000h base + 284h offset = 3033_0284h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0

IOMUXC_SW_PAD_CTL_PAD_ONOFF field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ONOFF 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ONOFF field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ONOFF 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ONOFF 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.158 SW_PAD_CTL_PAD_POR_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_POR_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 288h offset = 3033_0288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0

IOMUXC_SW_PAD_CTL_PAD_POR_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: POR_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: POR_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: POR_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.159 SW_PAD_CTL_PAD_RTC_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_RTC_RESET_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 28Ch offset = 3033_028Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	1	1	0	0	1	1	0	0

IOMUXC_SW_PAD_CTL_PAD_RTC_RESET_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: RTC_RESET_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_RTC_RESET_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: RTC_RESET_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: RTC_RESET_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.160 SW_PAD_CTL_PAD_GPIO1_IO00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00)

SW_PAD_CTL Register

Address: 3033_0000h base + 290h offset = 3033_0290h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO00 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO00 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO00 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO00 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.161 SW_PAD_CTL_PAD_GPIO1_IO01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01)

SW_PAD_CTL Register

Address: 3033_0000h base + 294h offset = 3033_0294h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO01 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO01 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO01 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO01 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.162 SW_PAD_CTL_PAD_GPIO1_IO02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02)

SW_PAD_CTL Register

Address: 3033_0000h base + 298h offset = 3033_0298h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO02 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO02 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO02 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO02 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.163 SW_PAD_CTL_PAD_GPIO1_IO03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03)

SW_PAD_CTL Register

Address: 3033_0000h base + 29Ch offset = 3033_029Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO03 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO03 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO03 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO03 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.164 SW_PAD_CTL_PAD_GPIO1_IO04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A0h offset = 3033_02A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO04 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO04 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO04 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO04 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.165 SW_PAD_CTL_PAD_GPIO1_IO05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A4h offset = 3033_02A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO05 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO05 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO05 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO05 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.166 SW_PAD_CTL_PAD_GPIO1_IO06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06)

SW_PAD_CTL Register

Address: 3033_0000h base + 2A8h offset = 3033_02A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO06 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO06 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO06 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO06 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.167 SW_PAD_CTL_PAD_GPIO1_IO07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07)

SW_PAD_CTL Register

Address: 3033_0000h base + 2ACh offset = 3033_02ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: GPIO1_IO07 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO07 field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO07 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO07 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO07 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.168 SW_PAD_CTL_PAD_GPIO1_IO08 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B0h offset = 3033_02B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO08 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO08 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO08 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO08 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.169 SW_PAD_CTL_PAD_GPIO1_IO09 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B4h offset = 3033_02B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO09 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO09 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO09 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO09 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.170 SW_PAD_CTL_PAD_GPIO1_IO10 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10)

SW_PAD_CTL Register

Address: 3033_0000h base + 2B8h offset = 3033_02B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO10 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO10 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO10 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO10 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.171 SW_PAD_CTL_PAD_GPIO1_IO11 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11)

SW_PAD_CTL Register

Address: 3033_0000h base + 2BCh offset = 3033_02BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO11 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO11 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO11 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO11 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.172 SW_PAD_CTL_PAD_GPIO1_IO12 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C0h offset = 3033_02C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO12 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO12 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO12 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO12 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.173 SW_PAD_CTL_PAD_GPIO1_IO13 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C4h offset = 3033_02C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO13 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO13 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO13 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO13 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.174 SW_PAD_CTL_PAD_GPIO1_IO14 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14)

SW_PAD_CTL Register

Address: 3033_0000h base + 2C8h offset = 3033_02C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO14 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO14 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO14 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO14 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.175 SW_PAD_CTL_PAD_GPIO1_IO15 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15)

SW_PAD_CTL Register

Address: 3033_0000h base + 2CCh offset = 3033_02CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_GPIO1_IO15 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: GPIO1_IO15 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: GPIO1_IO15 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: GPIO1_IO15 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.176 SW_PAD_CTL_PAD_ENET_MDC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D0h offset = 3033_02D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_MDC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_MDC field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_MDC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_MDC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.177 SW_PAD_CTL_PAD_ENET_MDIO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D4h offset = 3033_02D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_MDIO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_MDIO 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_MDIO 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_MDIO 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.178 SW_PAD_CTL_PAD_ENET_TD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 2D8h offset = 3033_02D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TD3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TD3 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TD3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TD3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.179 SW_PAD_CTL_PAD_ENET_TD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 2DCh offset = 3033_02DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TD2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TD2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TD2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.180 SW_PAD_CTL_PAD_ENET_TD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E0h offset = 3033_02E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TD1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TD1 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TD1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TD1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.181 SW_PAD_CTL_PAD_ENET_TD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E4h offset = 3033_02E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.182 SW_PAD_CTL_PAD_ENET_TX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL)

SW_PAD_CTL Register

Address: 3033_0000h base + 2E8h offset = 3033_02E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: ENET_TX_CTL 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TX_CTL field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TX_CTL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TX_CTL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TX_CTL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.183 SW_PAD_CTL_PAD_ENET_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2ECh offset = 3033_02ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_TXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_TXC field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_TXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_TXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.184 SW_PAD_CTL_PAD_ENET_RX_CTL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F0h offset = 3033_02F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RX_CTL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RX_CTL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RX_CTL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RX_CTL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.185 SW_PAD_CTL_PAD_ENET_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F4h offset = 3033_02F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RXC field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.186 SW_PAD_CTL_PAD_ENET_RD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 2F8h offset = 3033_02F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.187 SW_PAD_CTL_PAD_ENET_RD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 2FCh offset = 3033_02FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RD1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RD1 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RD1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RD1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.188 SW_PAD_CTL_PAD_ENET_RD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 300h offset = 3033_0300h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RD2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RD2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RD2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.189 SW_PAD_CTL_PAD_ENET_RD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ENET_RD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 304h offset = 3033_0304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ENET_RD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ENET_RD3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ENET_RD3 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ENET_RD3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ENET_RD3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.190 SW_PAD_CTL_PAD_SD1_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 308h offset = 3033_0308h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_CLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_CLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_CLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.191 SW_PAD_CTL_PAD_SD1_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_CMD)

SW_PAD_CTL Register

Address: 3033_0000h base + 30Ch offset = 3033_030Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_CMD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_CMD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_CMD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_CMD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.192 SW_PAD_CTL_PAD_SD1_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0)

SW_PAD_CTL Register

Address: 3033_0000h base + 310h offset = 3033_0310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.193 SW_PAD_CTL_PAD_SD1_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1)

SW_PAD_CTL Register

Address: 3033_0000h base + 314h offset = 3033_0314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA1 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.194 SW_PAD_CTL_PAD_SD1_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2)

SW_PAD_CTL Register

Address: 3033_0000h base + 318h offset = 3033_0318h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.195 SW_PAD_CTL_PAD_SD1_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3)

SW_PAD_CTL Register

Address: 3033_0000h base + 31Ch offset = 3033_031Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SD1_DATA3 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA3 field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.196 SW_PAD_CTL_PAD_SD1_DATA4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4)

SW_PAD_CTL Register

Address: 3033_0000h base + 320h offset = 3033_0320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA4 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA4 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA4 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA4 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.197 SW_PAD_CTL_PAD_SD1_DATA5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5)

SW_PAD_CTL Register

Address: 3033_0000h base + 324h offset = 3033_0324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA5 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA5 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA5 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.198 SW_PAD_CTL_PAD_SD1_DATA6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6)

SW_PAD_CTL Register

Address: 3033_0000h base + 328h offset = 3033_0328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA6 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA6 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA6 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA6 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.199 SW_PAD_CTL_PAD_SD1_DATA7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7)

SW_PAD_CTL Register

Address: 3033_0000h base + 32Ch offset = 3033_032Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_DATA7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_DATA7 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_DATA7 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_DATA7 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.200 SW_PAD_CTL_PAD_SD1_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 330h offset = 3033_0330h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_RESET_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD1_RESET_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_RESET_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_RESET_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.201 SW_PAD_CTL_PAD_SD1_STROBE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE)

SW_PAD_CTL Register

Address: 3033_0000h base + 334h offset = 3033_0334h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD1_STROBE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD1_STROBE 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD1_STROBE 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD1_STROBE 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.202 SW_PAD_CTL_PAD_SD2_CD_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 338h offset = 3033_0338h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_CD_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CD_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_CD_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_CD_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.203 SW_PAD_CTL_PAD_SD2_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 33Ch offset = 3033_033Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_CLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_CLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_CLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.204 SW_PAD_CTL_PAD_SD2_CMD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_CMD)

SW_PAD_CTL Register

Address: 3033_0000h base + 340h offset = 3033_0340h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_CMD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_CMD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_CMD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_CMD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.205 SW_PAD_CTL_PAD_SD2_DATA0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0)

SW_PAD_CTL Register

Address: 3033_0000h base + 344h offset = 3033_0344h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SD2_DATA0 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_DATA0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA0 field descriptions (continued)

Field	Description
	Select one out of next values for pad: SD2_DATA0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.206 SW_PAD_CTL_PAD_SD2_DATA1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1)

SW_PAD_CTL Register

Address: 3033_0000h base + 348h offset = 3033_0348h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA1 field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_DATA1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.207 SW_PAD_CTL_PAD_SD2_DATA2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2)

SW_PAD_CTL Register

Address: 3033_0000h base + 34Ch offset = 3033_034Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA2 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_DATA2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.208 SW_PAD_CTL_PAD_SD2_DATA3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3)

SW_PAD_CTL Register

Address: 3033_0000h base + 350h offset = 3033_0350h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_DATA3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_DATA3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_DATA3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_DATA3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.209 SW_PAD_CTL_PAD_SD2_RESET_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 354h offset = 3033_0354h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_RESET_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SD2_RESET_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_RESET_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_RESET_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.210 SW_PAD_CTL_PAD_SD2_WP SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SD2_WP)

SW_PAD_CTL Register

Address: 3033_0000h base + 358h offset = 3033_0358h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

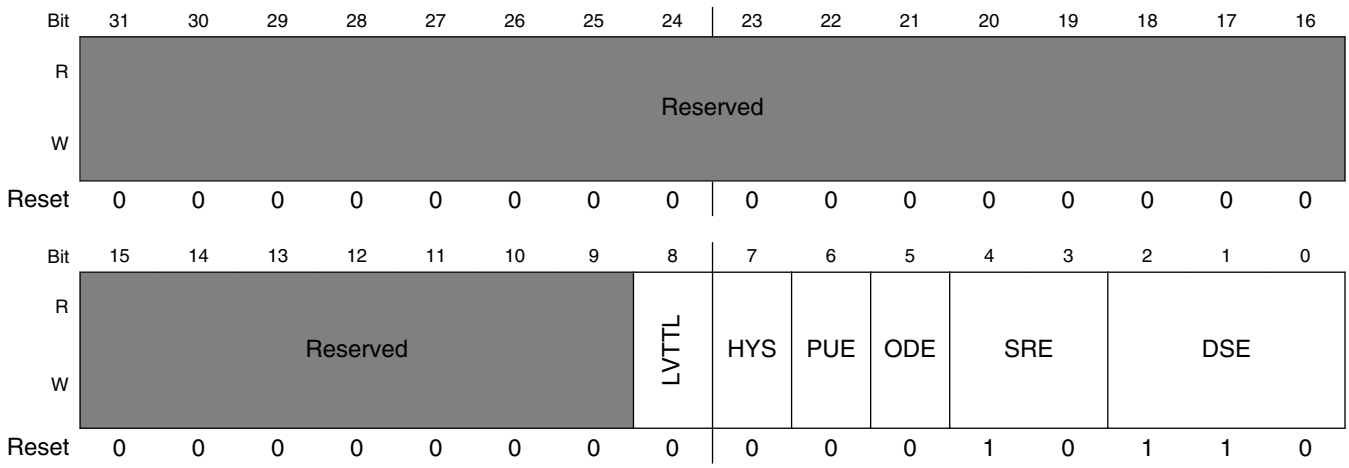
IOMUXC_SW_PAD_CTL_PAD_SD2_WP field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SD2_WP 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SD2_WP 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SD2_WP 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.211 SW_PAD_CTL_PAD_NAND_ALE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_ALE)

SW_PAD_CTL Register

Address: 3033_0000h base + 35Ch offset = 3033_035Ch



IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_ALE 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_ALE field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_ALE 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_ALE 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.212 SW_PAD_CTL_PAD_NAND_CE0_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 360h offset = 3033_0360h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CE0_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_CE0_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_CE0_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_CE0_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.213 SW_PAD_CTL_PAD_NAND_CE1_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 364h offset = 3033_0364h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_CE1_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CE1_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_CE1_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_CE1_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.214 SW_PAD_CTL_PAD_NAND_CE2_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 368h offset = 3033_0368h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CE2_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_CE2_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_CE2_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_CE2_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.215 SW_PAD_CTL_PAD_NAND_CE3_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 36Ch offset = 3033_036Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_CE3_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_CE3_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_CE3_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_CE3_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.216 SW_PAD_CTL_PAD_NAND_CLE SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_CLE)

SW_PAD_CTL Register

Address: 3033_0000h base + 370h offset = 3033_0370h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_CLE field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_CLE 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_CLE 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_CLE 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.217 SW_PAD_CTL_PAD_NAND_DATA00 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00)

SW_PAD_CTL Register

Address: 3033_0000h base + 374h offset = 3033_0374h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA00 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA00 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA00 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA00 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.218 SW_PAD_CTL_PAD_NAND_DATA01 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01)

SW_PAD_CTL Register

Address: 3033_0000h base + 378h offset = 3033_0378h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA01 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA01 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA01 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA01 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.219 SW_PAD_CTL_PAD_NAND_DATA02 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02)

SW_PAD_CTL Register

Address: 3033_0000h base + 37Ch offset = 3033_037Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA02 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA02 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA02 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA02 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.220 SW_PAD_CTL_PAD_NAND_DATA03 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03)

SW_PAD_CTL Register

Address: 3033_0000h base + 380h offset = 3033_0380h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: NAND_DATA03 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA03 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA03 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA03 field descriptions (continued)

Field	Description
	Select one out of next values for pad: NAND_DATA03 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.221 SW_PAD_CTL_PAD_NAND_DATA04 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04)

SW_PAD_CTL Register

Address: 3033_0000h base + 384h offset = 3033_0384h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA04 field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA04 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA04 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA04 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.222 SW_PAD_CTL_PAD_NAND_DATA05 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05)

SW_PAD_CTL Register

Address: 3033_0000h base + 388h offset = 3033_0388h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA05 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA05 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA05 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA05 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.223 SW_PAD_CTL_PAD_NAND_DATA06 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06)

SW_PAD_CTL Register

Address: 3033_0000h base + 38Ch offset = 3033_038Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA06 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA06 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA06 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA06 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.224 SW_PAD_CTL_PAD_NAND_DATA07 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07)

SW_PAD_CTL Register

Address: 3033_0000h base + 390h offset = 3033_0390h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DATA07 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_DATA07 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DATA07 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DATA07 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.225 SW_PAD_CTL_PAD_NAND_DQS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_DQS)

SW_PAD_CTL Register

Address: 3033_0000h base + 394h offset = 3033_0394h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_DQS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_DQS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_DQS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_DQS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.226 SW_PAD_CTL_PAD_NAND_RE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 398h offset = 3033_0398h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_RE_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_RE_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_RE_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_RE_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.227 SW_PAD_CTL_PAD_NAND_READY_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 39Ch offset = 3033_039Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_READY_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_READY_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_READY_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_READY_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.228 SW_PAD_CTL_PAD_NAND_WE_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A0h offset = 3033_03A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_WE_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_NAND_WE_B field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_WE_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_WE_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.229 SW_PAD_CTL_PAD_NAND_WP_B SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A4h offset = 3033_03A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_NAND_WP_B field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: NAND_WP_B 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: NAND_WP_B 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: NAND_WP_B 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.230 SW_PAD_CTL_PAD_SAI5_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3A8h offset = 3033_03A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXFS field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.231 SW_PAD_CTL_PAD_SAI5_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 3ACh offset = 3033_03ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

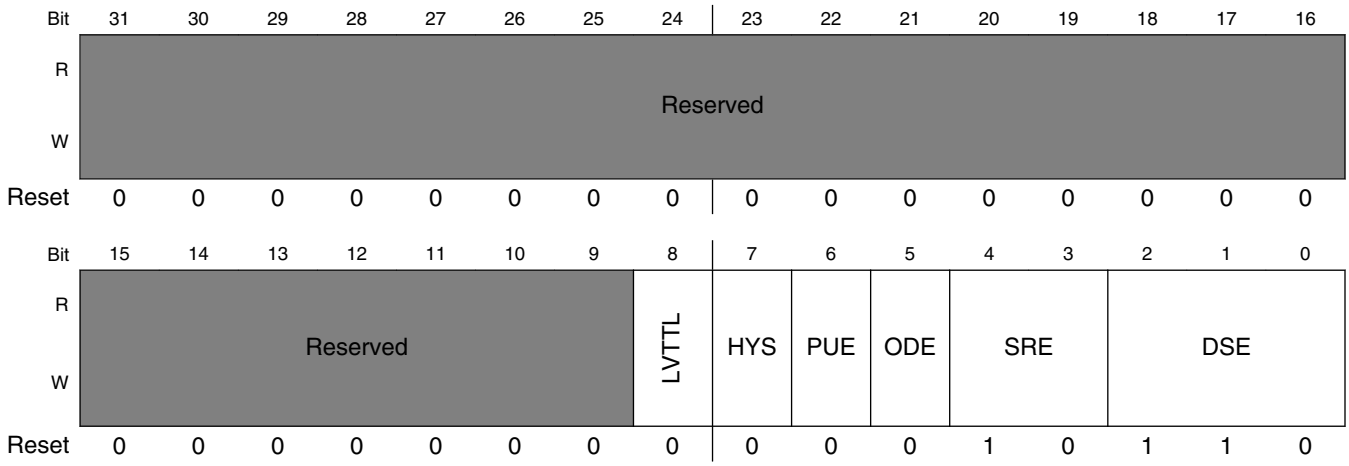
IOMUXC_SW_PAD_CTL_PAD_SAI5_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.232 SW_PAD_CTL_PAD_SAI5_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B0h offset = 3033_03B0h



IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD0 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.233 SW_PAD_CTL_PAD_SAI5_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B4h offset = 3033_03B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1 field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SAI5_RXD1 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXD1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXD1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD1 field descriptions (continued)

Field	Description
	Select one out of next values for pad: SAI5_RXD1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.234 SW_PAD_CTL_PAD_SAI5_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 3B8h offset = 3033_03B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD2 field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXD2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXD2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.235 SW_PAD_CTL_PAD_SAI5_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 3BCh offset = 3033_03BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_RXD3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI5_RXD3 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_RXD3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_RXD3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.236 SW_PAD_CTL_PAD_SAI5_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C0h offset = 3033_03C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI5_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI5_MCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI5_MCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI5_MCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.237 SW_PAD_CTL_PAD_SAI1_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C4h offset = 3033_03C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXFS field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.238 SW_PAD_CTL_PAD_SAI1_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 3C8h offset = 3033_03C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.239 SW_PAD_CTL_PAD_SAI1_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 3CCh offset = 3033_03CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD0 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.240 SW_PAD_CTL_PAD_SAI1_RXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D0h offset = 3033_03D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.241 SW_PAD_CTL_PAD_SAI1_RXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D4h offset = 3033_03D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD2 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.242 SW_PAD_CTL_PAD_SAI1_RXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 3D8h offset = 3033_03D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.243 SW_PAD_CTL_PAD_SAI1_RXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4)

SW_PAD_CTL Register

Address: 3033_0000h base + 3DCh offset = 3033_03DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD4 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD4 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD4 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD4 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.244 SW_PAD_CTL_PAD_SAI1_RXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E0h offset = 3033_03E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD5 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD5 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD5 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.245 SW_PAD_CTL_PAD_SAI1_RXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E4h offset = 3033_03E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD6 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD6 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD6 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD6 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.246 SW_PAD_CTL_PAD_SAI1_RXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7)

SW_PAD_CTL Register

Address: 3033_0000h base + 3E8h offset = 3033_03E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_RXD7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_RXD7 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_RXD7 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_RXD7 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.247 SW_PAD_CTL_PAD_SAI1_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 3ECh offset = 3033_03ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SAI1_TXFS 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXFS field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.248 SW_PAD_CTL_PAD_SAI1_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F0h offset = 3033_03F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXC field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.249 SW_PAD_CTL_PAD_SAI1_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F4h offset = 3033_03F4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.250 SW_PAD_CTL_PAD_SAI1_TXD1 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1)

SW_PAD_CTL Register

Address: 3033_0000h base + 3F8h offset = 3033_03F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD1 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD1 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD1 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD1 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.251 SW_PAD_CTL_PAD_SAI1_TXD2 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2)

SW_PAD_CTL Register

Address: 3033_0000h base + 3FCh offset = 3033_03FCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD2 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD2 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD2 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD2 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.252 SW_PAD_CTL_PAD_SAI1_TXD3 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3)

SW_PAD_CTL Register

Address: 3033_0000h base + 400h offset = 3033_0400h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD3 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD3 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD3 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD3 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.253 SW_PAD_CTL_PAD_SAI1_TXD4 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4)

SW_PAD_CTL Register

Address: 3033_0000h base + 404h offset = 3033_0404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD4 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD4 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD4 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD4 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.254 SW_PAD_CTL_PAD_SAI1_TXD5 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5)

SW_PAD_CTL Register

Address: 3033_0000h base + 408h offset = 3033_0408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD5 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD5 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD5 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD5 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.255 SW_PAD_CTL_PAD_SAI1_TXD6 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6)

SW_PAD_CTL Register

Address: 3033_0000h base + 40Ch offset = 3033_040Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD6 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD6 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD6 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD6 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.256 SW_PAD_CTL_PAD_SAI1_TXD7 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7)

SW_PAD_CTL Register

Address: 3033_0000h base + 410h offset = 3033_0410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_TXD7 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI1_TXD7 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_TXD7 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_TXD7 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.257 SW_PAD_CTL_PAD_SAI1_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 414h offset = 3033_0414h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI1_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI1_MCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI1_MCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI1_MCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.258 SW_PAD_CTL_PAD_SAI2_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 418h offset = 3033_0418h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_RXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI2_RXFS field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_RXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.259 SW_PAD_CTL_PAD_SAI2_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 41Ch offset = 3033_041Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_RXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_RXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.260 SW_PAD_CTL_PAD_SAI2_RXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 420h offset = 3033_0420h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_RXD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI2_RXD0 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_RXD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_RXD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.261 SW_PAD_CTL_PAD_SAI2_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 424h offset = 3033_0424h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SAI2_TXFS 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_TXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_TXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXFS field descriptions (continued)

Field	Description
	Select one out of next values for pad: SAI2_TXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.262 SW_PAD_CTL_PAD_SAI2_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 428h offset = 3033_0428h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXC field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_TXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_TXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_TXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.263 SW_PAD_CTL_PAD_SAI2_TXD0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0)

SW_PAD_CTL Register

Address: 3033_0000h base + 42Ch offset = 3033_042Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_TXD0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI2_TXD0 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_TXD0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_TXD0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.264 SW_PAD_CTL_PAD_SAI2_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 430h offset = 3033_0430h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI2_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI2_MCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI2_MCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI2_MCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.265 SW_PAD_CTL_PAD_SAI3_RXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 434h offset = 3033_0434h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_RXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI3_RXFS field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_RXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.266 SW_PAD_CTL_PAD_SAI3_RXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 438h offset = 3033_0438h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_RXC field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_RXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_RXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.267 SW_PAD_CTL_PAD_SAI3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 43Ch offset = 3033_043Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_RXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI3_RXD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_RXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_RXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.268 SW_PAD_CTL_PAD_SAI3_TXFS SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS)

SW_PAD_CTL Register

Address: 3033_0000h base + 440h offset = 3033_0440h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_TXFS field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_TXFS 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_TXFS 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXFS 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.269 SW_PAD_CTL_PAD_SAI3_TXC SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC)

SW_PAD_CTL Register

Address: 3033_0000h base + 444h offset = 3033_0444h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: SAI3_TXC 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI3_TXC field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_TXC 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_TXC 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXC 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.270 SW_PAD_CTL_PAD_SAI3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 448h offset = 3033_0448h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_TXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SAI3_TXD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_TXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_TXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.271 SW_PAD_CTL_PAD_SAI3_MCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 44Ch offset = 3033_044Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SAI3_MCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SAI3_MCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SAI3_MCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SAI3_MCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.272 SW_PAD_CTL_PAD_SPDIF_TX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX)

SW_PAD_CTL Register

Address: 3033_0000h base + 450h offset = 3033_0450h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SPDIF_TX 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SPDIF_TX field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SPDIF_TX 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SPDIF_TX 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.273 SW_PAD_CTL_PAD_SPDIF_RX SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX)

SW_PAD_CTL Register

Address: 3033_0000h base + 454h offset = 3033_0454h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SPDIF_RX field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SPDIF_RX 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SPDIF_RX 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SPDIF_RX 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.274 SW_PAD_CTL_PAD_SPDIF_EXT_CLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 458h offset = 3033_0458h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: SPDIF_EXT_CLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_SPDIF_EXT_CLK field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: SPDIF_EXT_CLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: SPDIF_EXT_CLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.275 SW_PAD_CTL_PAD_ECSPI1_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 45Ch offset = 3033_045Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SCLK field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI1_SCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI1_SCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_SCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.276 SW_PAD_CTL_PAD_ECSPI1_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI)

SW_PAD_CTL Register

Address: 3033_0000h base + 460h offset = 3033_0460h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI1_MOSI 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MOSI field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI1_MOSI 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_MOSI 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.277 SW_PAD_CTL_PAD_ECSPI1_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO)

SW_PAD_CTL Register

Address: 3033_0000h base + 464h offset = 3033_0464h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_MISO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI1_MISO 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI1_MISO 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_MISO 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.278 SW_PAD_CTL_PAD_ECSPI1_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0)

SW_PAD_CTL Register

Address: 3033_0000h base + 468h offset = 3033_0468h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI1_SS0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ECSPI1_SS0 field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI1_SS0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI1_SS0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.279 SW_PAD_CTL_PAD_ECSPI2_SCLK SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK)

SW_PAD_CTL Register

Address: 3033_0000h base + 46Ch offset = 3033_046Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: ECSPI2_SCLK 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI2_SCLK 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI2_SCLK 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SCLK field descriptions (continued)

Field	Description
	Select one out of next values for pad: ECSPI2_SCLK 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.280 SW_PAD_CTL_PAD_ECSPI2_MOSI SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI)

SW_PAD_CTL Register

Address: 3033_0000h base + 470h offset = 3033_0470h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MOSI field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI2_MOSI 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI2_MOSI 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_MOSI 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.281 SW_PAD_CTL_PAD_ECSPI2_MISO SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO)

SW_PAD_CTL Register

Address: 3033_0000h base + 474h offset = 3033_0474h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI2_MISO 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_MISO field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI2_MISO 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_MISO 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.282 SW_PAD_CTL_PAD_ECSPI2_SS0 SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0)

SW_PAD_CTL Register

Address: 3033_0000h base + 478h offset = 3033_0478h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_ECSPI2_SS0 field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: ECSPI2_SS0 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: ECSPI2_SS0 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: ECSPI2_SS0 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.283 SW_PAD_CTL_PAD_I2C1_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 47Ch offset = 3033_047Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C1_SCL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_I2C1_SCL field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C1_SCL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C1_SCL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.284 SW_PAD_CTL_PAD_I2C1_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 480h offset = 3033_0480h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C1_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C1_SDA 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C1_SDA 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C1_SDA 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.285 SW_PAD_CTL_PAD_I2C2_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 484h offset = 3033_0484h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C2_SCL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_I2C2_SCL field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C2_SCL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C2_SCL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.286 SW_PAD_CTL_PAD_I2C2_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 488h offset = 3033_0488h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C2_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C2_SDA 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C2_SDA 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C2_SDA 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.287 SW_PAD_CTL_PAD_I2C3_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 48Ch offset = 3033_048Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: I2C3_SCL 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_I2C3_SCL field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C3_SCL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C3_SCL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C3_SCL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.288 SW_PAD_CTL_PAD_I2C3_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 490h offset = 3033_0490h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C3_SDA 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_I2C3_SDA field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C3_SDA 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C3_SDA 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.289 SW_PAD_CTL_PAD_I2C4_SCL SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL)

SW_PAD_CTL Register

Address: 3033_0000h base + 494h offset = 3033_0494h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C4_SCL field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C4_SCL 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C4_SCL 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C4_SCL 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.290 SW_PAD_CTL_PAD_I2C4_SDA SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA)

SW_PAD_CTL Register

Address: 3033_0000h base + 498h offset = 3033_0498h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: I2C4_SDA 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_I2C4_SDA field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: I2C4_SDA 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: I2C4_SDA 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.291 SW_PAD_CTL_PAD_UART1_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 49Ch offset = 3033_049Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART1_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART1_RXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART1_RXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART1_RXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.292 SW_PAD_CTL_PAD_UART1_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART1_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A0h offset = 3033_04A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART1_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART1_TXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_UART1_TXD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART1_TXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART1_TXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.293 SW_PAD_CTL_PAD_UART2_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A4h offset = 3033_04A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART2_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART2_RXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART2_RXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART2_RXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.294 SW_PAD_CTL_PAD_UART2_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART2_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4A8h offset = 3033_04A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART2_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART2_TXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_UART2_TXD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART2_TXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART2_TXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.295 SW_PAD_CTL_PAD_UART3_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4ACh offset = 3033_04ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		VSEL			Reserved		LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	1	1	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART3_RXD field descriptions

Field	Description
31–14 -	This field is reserved. Reserved
13–11 VSEL	Voltage Select Field Select one out of next values for pad: UART3_RXD 000 **VSEL_0_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 001 **VSEL_1_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 010 **VSEL_2_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 011 **VSEL_3_Auto_Detct_Mode** — Auto Detect 3.3/2.5/1.2/1.8 V mode 100 **VSEL_4_Manual_3p3V_Mode** — Manually Set 3.3V mode 101 **VSEL_5_Manual_2p5V_Mode** — Manually Set 2.5V mode 110 **VSEL_6_Manual_2p5V_Mode** — Manually Set 2.5V mode 111 **VSEL_7_Manual_1p2_1p8V_Mode** — Manually Set 1.2V/1.8V mode
10–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART3_RXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART3_RXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_UART3_RXD field descriptions (continued)

Field	Description
	Select one out of next values for pad: UART3_RXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.296 SW_PAD_CTL_PAD_UART3_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART3_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B0h offset = 3033_04B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART3_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_UART3_TXD field descriptions (continued)

Field	Description
	0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART3_TXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART3_TXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART3_TXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.297 SW_PAD_CTL_PAD_UART4_RXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_RXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B4h offset = 3033_04B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

IOMUXC_SW_PAD_CTL_PAD_UART4_RXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART4_RXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled

Table continues on the next page...

IOMUXC_SW_PAD_CTL_PAD_UART4_RXD field descriptions (continued)

Field	Description
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART4_RXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART4_RXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.298 SW_PAD_CTL_PAD_UART4_TXD SW PAD Control Register (IOMUXC_SW_PAD_CTL_PAD_UART4_TXD)

SW_PAD_CTL Register

Address: 3033_0000h base + 4B8h offset = 3033_04B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							LVTTL	HYS	PUE	ODE	SRE		DSE		
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0

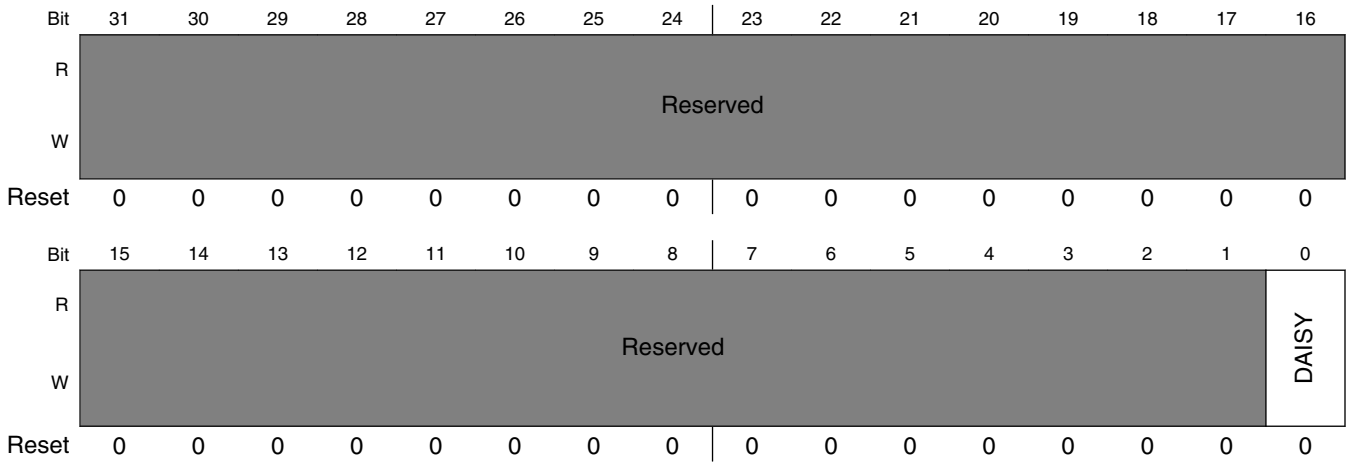
IOMUXC_SW_PAD_CTL_PAD_UART4_TXD field descriptions

Field	Description
31–9 -	This field is reserved. Reserved
8 LVTTL	Lvttl Enable Field Control signal to select LVTTL input 0 **Disabled** — LVTTL Disabled 1 **Enabled** — LVTTL Enabled
7 HYS	Schmitt trigger Enable Field Control signal to select Schymitt trigger 0 **Disabled** — Schmitt Trigger Disabled 1 **Enabled** — Schmitt Trigger Enabled
6 PUE	Pull Up Enable Field Control signal to select internal pullup resistor 0 **Disabled** — Pull Up Resistor Disabled 1 **Enabled** — Pull Up Resistor Enabled
5 ODE	Open Drain Enable Field Select one out of next values for pad: UART4_TXD 0 **Disabled** — Open Drain Disabled 1 **Enabled** — Open Drain Enabled
4–3 SRE	Slew Rate Field 2-bit slew rate control signals to select between 50,100 and 200MHz I/O cell operation frequency range with reduced switching noise. Select one out of next values for pad: UART4_TXD 00 **SLOW** — Slow Frequency Slew Rate (50Mhz) 01 **MEDIUM** — Medium Frequency Slew Rate (100Mhz) 10 **FAST** — Fast Frequency Slew Rate (150Mhz) 11 **MAX** — Max Frequency Slew Rate (200Mhz)
DSE	Drive Strength Field Select one out of next values for pad: UART4_TXD 000 **HI-Z** — Output driver is disabled (Hi-Z State) 001 **255_OHM** — 255 Ohm @3.3V, 240 Ohm @2.5V, 230 Ohm @1.8V, 265 Ohm @1.2V 010 **105_OHM** — 105 Ohm @3.3V, 100 Ohm @2.5V, 85 Ohm @1.8V, 110 Ohm @1.2V 011 **75_OHM** — 75 Ohm @3.3V, 70 Ohm @2.5V, 60 Ohm @1.8V, 80 Ohm @1.2V 100 **85_OHM** — 85 Ohm @3.3V, 80 Ohm @2.5V, 75 Ohm @1.8V, 90 Ohm @1.2V 101 **65_OHM** — 65 Ohm @3.3V, 60 Ohm @2.5V, 55 Ohm @1.8V, 65 Ohm @1.2V 110 **45_OHM** — 45 Ohm @3.3V, 45 Ohm @2.5V, 40 Ohm @1.8V, 50 Ohm @1.2V 111 **40_OHM** — 40 Ohm @3.3V, 40 Ohm @2.5V, 33 Ohm @1.8V, 40 Ohm @1.2V

8.2.5.299 CCM_PMIC_READY_SELECT_INPUT DAISY Register (IOMUXC_CCM_PMIC_READY_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4BCh offset = 3033_04BCh



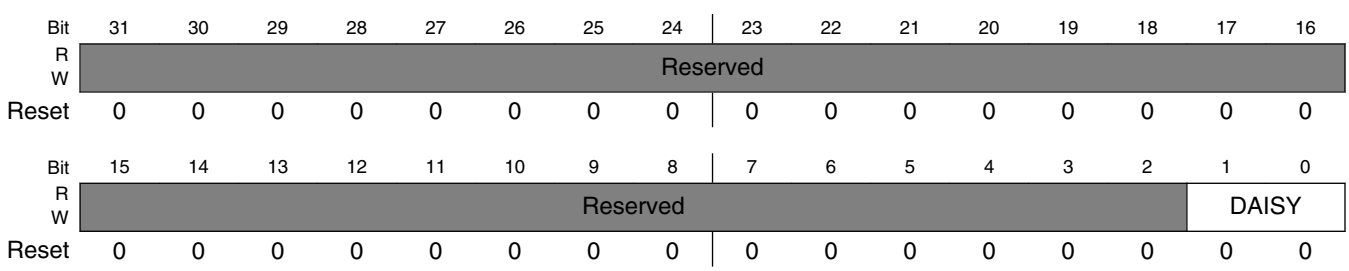
IOMUXC_CCM_PMIC_READY_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **GPIO1_IO05_ALT5** — Selecting Pad: GPIO1_IO05 Mode: ALT5 for CCM_PMIC_READY 1 **GPIO1_IO11_ALT5** — Selecting Pad: GPIO1_IO11 Mode: ALT5 for CCM_PMIC_READY

8.2.5.300 ENET1_MDIO_SELECT_INPUT DAISY Register (IOMUXC_ENET1_MDIO_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4C0h offset = 3033_04C0h



IOMUXC_ENET1_MDIO_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **GPIO01_IO07_ALT1** — Selecting Pad: GPIO01_IO07 Mode: ALT1 for ENET1_MDIO 01 **ENET_MDIO_ALT0** — Selecting Pad: ENET_MDIO Mode: ALT0 for ENET1_MDIO 10 **I2C1_SDA_ALT1** — Selecting Pad: I2C1_SDA Mode: ALT1 for ENET1_MDIO 11 **GPIO01_IO07_ALT1** — Selecting Pad: GPIO01_IO07 Mode: ALT1 for ENET1_MDIO

8.2.5.301 SAI1_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI1_RX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4C4h offset = 3033_04C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI1_RX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **SAI1_RXFS_ALT0** — Selecting Pad: SAI1_RXFS Mode: ALT0 for SAI1_RX_SYNC 1 **SAI1_RXD5_ALT3** — Selecting Pad: SAI1_RXD5 Mode: ALT3 for SAI1_RX_SYNC

8.2.5.302 SAI1_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI1_TX_BCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4C8h offset = 3033_04C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI1_TX_BCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_MCLK_ALT1** — Selecting Pad: SAI5_MCLK Mode: ALT1 for SAI1_TX_BCLK 01 **SAI1_TXC_ALT0** — Selecting Pad: SAI1_TXC Mode: ALT0 for SAI1_TX_BCLK 10 **SAI1_MCLK_ALT2** — Selecting Pad: SAI1_MCLK Mode: ALT2 for SAI1_TX_BCLK 11 **SAI5_MCLK_ALT1** — Selecting Pad: SAI5_MCLK Mode: ALT1 for SAI1_TX_BCLK

8.2.5.303 SAI1_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI1_TX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4CCh offset = 3033_04CCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI1_TX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–3 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field

Table continues on the next page...

IOMUXC_SAI1_TX_SYNC_SELECT_INPUT field descriptions (continued)

Field	Description
	Selecting Pads Involved in Daisy Chain 000 **SAI5_RXD1_ALT2** — Selecting Pad: SAI5_RXD1 Mode: ALT2 for SAI1_TX_SYNC 001 **SAI5_RXD2_ALT2** — Selecting Pad: SAI5_RXD2 Mode: ALT2 for SAI1_TX_SYNC 010 **SAI5_RXD3_ALT2** — Selecting Pad: SAI5_RXD3 Mode: ALT2 for SAI1_TX_SYNC 011 **SAI1_TXFS_ALT0** — Selecting Pad: SAI1_TXFS Mode: ALT0 for SAI1_TX_SYNC 100 **SAI1_RXD7_ALT2** — Selecting Pad: SAI1_RXD7 Mode: ALT2 for SAI1_TX_SYNC 101 **SAI5_RXD1_ALT2** — Selecting Pad: SAI5_RXD1 Mode: ALT2 for SAI1_TX_SYNC 110 **SAI5_RXD1_ALT2** — Selecting Pad: SAI5_RXD1 Mode: ALT2 for SAI1_TX_SYNC 111 **SAI5_RXD1_ALT2** — Selecting Pad: SAI5_RXD1 Mode: ALT2 for SAI1_TX_SYNC

8.2.5.304 SAI5_RX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RX_BCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D0h offset = 3033_04D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RX_BCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXC_ALT0** — Selecting Pad: SAI5_RXC Mode: ALT0 for SAI5_RX_BCLK 01 **SAI1_RXC_ALT1** — Selecting Pad: SAI1_RXC Mode: ALT1 for SAI5_RX_BCLK 10 **SAI3_RXC_ALT2** — Selecting Pad: SAI3_RXC Mode: ALT2 for SAI5_RX_BCLK 11 **SAI5_RXC_ALT0** — Selecting Pad: SAI5_RXC Mode: ALT0 for SAI5_RX_BCLK

8.2.5.305 SAI5_RXD0_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD0_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D4h offset = 3033_04D4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RXD0_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD0_ALT0** — Selecting Pad: SAI5_RXD0 Mode: ALT0 for SAI5_RXD0 01 **SAI1_RXD0_ALT1** — Selecting Pad: SAI1_RXD0 Mode: ALT1 for SAI5_RXD0 10 **SAI3_RXD_ALT2** — Selecting Pad: SAI3_RXD Mode: ALT2 for SAI5_RXD0 11 **SAI5_RXD0_ALT0** — Selecting Pad: SAI5_RXD0 Mode: ALT0 for SAI5_RXD0

8.2.5.306 SAI5_RXD1_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD1_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4D8h offset = 3033_04D8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RXD1_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD1_ALT0** — Selecting Pad: SAI5_RXD1 Mode: ALT0 for SAI5_RXD1 01 **SAI1_RXD1_ALT1** — Selecting Pad: SAI1_RXD1 Mode: ALT1 for SAI5_RXD1 10 **SAI3_TXFS_ALT1** — Selecting Pad: SAI3_TXFS Mode: ALT1 for SAI5_RXD1 11 **SAI5_RXD1_ALT0** — Selecting Pad: SAI5_RXD1 Mode: ALT0 for SAI5_RXD1

8.2.5.307 SAI5_RXD2_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD2_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4DCh offset = 3033_04DCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RXD2_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD2_ALT0** — Selecting Pad: SAI5_RXD2 Mode: ALT0 for SAI5_RXD2 01 **SAI1_RXD2_ALT1** — Selecting Pad: SAI1_RXD2 Mode: ALT1 for SAI5_RXD2 10 **SAI3_TXC_ALT2** — Selecting Pad: SAI3_TXC Mode: ALT2 for SAI5_RXD2 11 **SAI5_RXD2_ALT0** — Selecting Pad: SAI5_RXD2 Mode: ALT0 for SAI5_RXD2

8.2.5.308 SAI5_RXD3_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RXD3_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4E0h offset = 3033_04E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RXD3_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD3_ALT0** — Selecting Pad: SAI5_RXD3 Mode: ALT0 for SAI5_RXD3 01 **SAI1_RXD3_ALT1** — Selecting Pad: SAI1_RXD3 Mode: ALT1 for SAI5_RXD3 10 **SAI3_TXD_ALT2** — Selecting Pad: SAI3_TXD Mode: ALT2 for SAI5_RXD3 11 **SAI5_RXD3_ALT0** — Selecting Pad: SAI5_RXD3 Mode: ALT0 for SAI5_RXD3

8.2.5.309 SAI5_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI5_RX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4E4h offset = 3033_04E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_RX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXFS_ALT0** — Selecting Pad: SAI5_RXFS Mode: ALT0 for SAI5_RX_SYNC 01 **SAI1_RXFS_ALT1** — Selecting Pad: SAI1_RXFS Mode: ALT1 for SAI5_RX_SYNC 10 **SAI3_RXFS_ALT2** — Selecting Pad: SAI3_RXFS Mode: ALT2 for SAI5_RX_SYNC 11 **SAI5_RXFS_ALT0** — Selecting Pad: SAI5_RXFS Mode: ALT0 for SAI5_RX_SYNC

8.2.5.310 SAI5_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_TX_BCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4E8h offset = 3033_04E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI5_TX_BCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD2_ALT3** — Selecting Pad: SAI5_RXD2 Mode: ALT3 for SAI5_TX_BCLK 01 **SAI1_TXC_ALT1** — Selecting Pad: SAI1_TXC Mode: ALT1 for SAI5_TX_BCLK 10 **SAI2_RXC_ALT1** — Selecting Pad: SAI2_RXC Mode: ALT1 for SAI5_TX_BCLK 11 **SAI5_RXD2_ALT3** — Selecting Pad: SAI5_RXD2 Mode: ALT3 for SAI5_TX_BCLK

8.2.5.311 SAI5_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI5_TX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4ECh offset = 3033_04ECh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

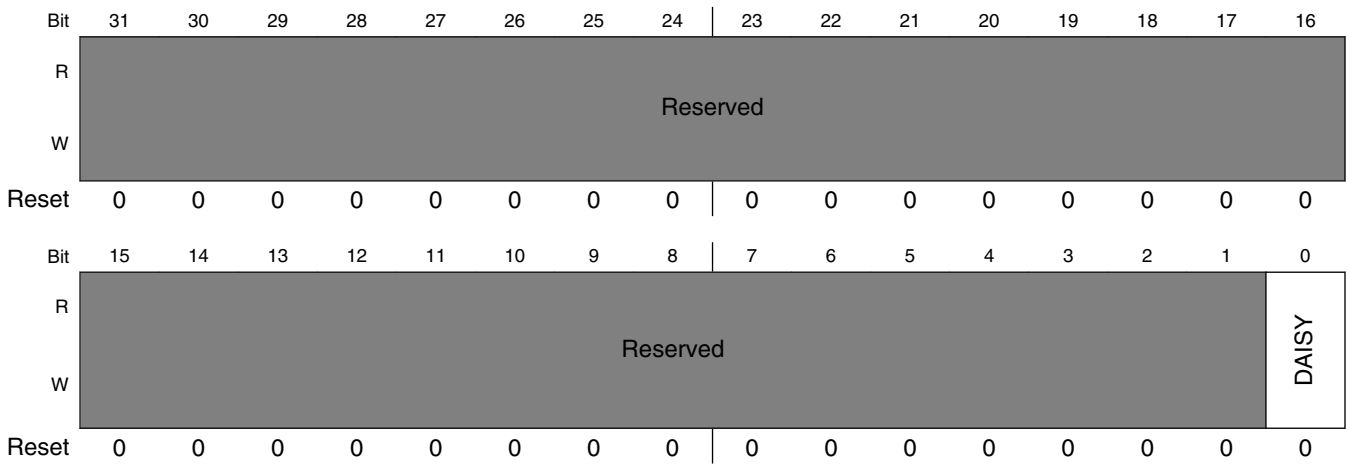
IOMUXC_SAI5_TX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_RXD1_ALT3** — Selecting Pad: SAI5_RXD1 Mode: ALT3 for SAI5_TX_SYNC 01 **SAI1_TXFS_ALT1** — Selecting Pad: SAI1_TXFS Mode: ALT1 for SAI5_TX_SYNC 10 **SAI2_RXFS_ALT1** — Selecting Pad: SAI2_RXFS Mode: ALT1 for SAI5_TX_SYNC 11 **SAI5_RXD1_ALT3** — Selecting Pad: SAI5_RXD1 Mode: ALT3 for SAI5_TX_SYNC

8.2.5.312 UART1_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART1_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4F0h offset = 3033_04F0h



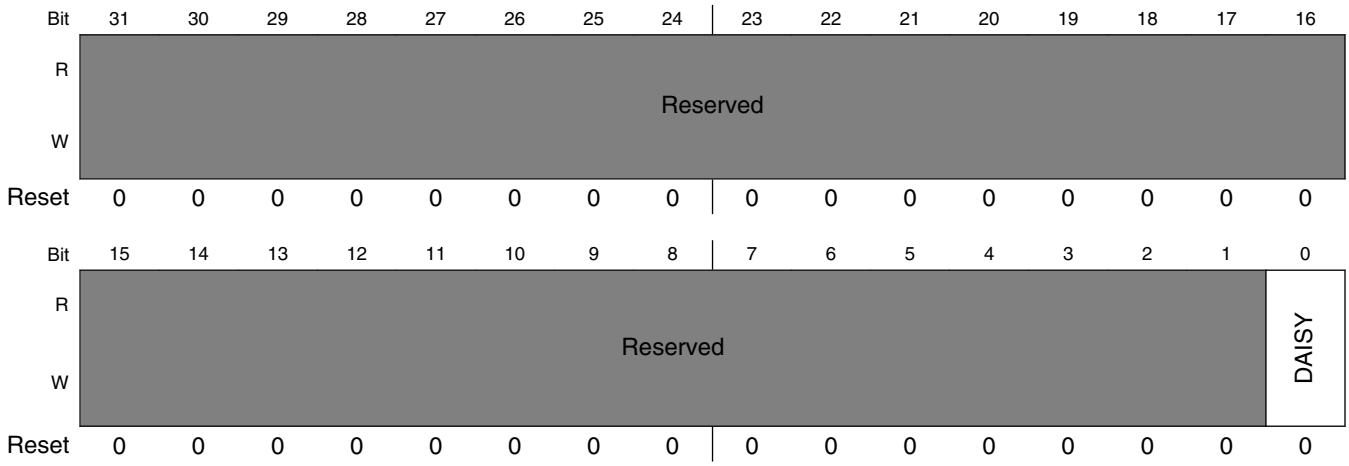
IOMUXC_UART1_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **UART3_RXD_ALT1** — Selecting Pad: UART3_RXD Mode: ALT1 for UART1_RTS_B 1 **UART3_TXD_ALT1** — Selecting Pad: UART3_TXD Mode: ALT1 for UART1_RTS_B

8.2.5.313 UART1_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART1_RXD_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4F4h offset = 3033_04F4h



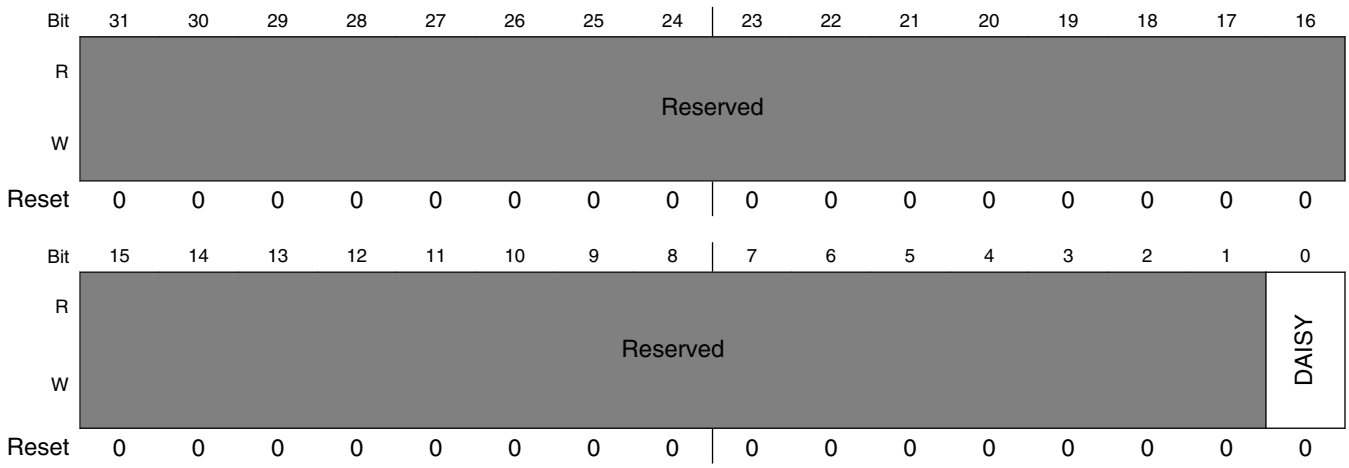
IOMUXC_UART1_RXD_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **UART1_RXD_ALT0** — Selecting Pad: UART1_RXD Mode: ALT0 for UART1_RXD 1 **UART1_TXD_ALT0** — Selecting Pad: UART1_TXD Mode: ALT0 for UART1_RXD

8.2.5.314 UART2_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART2_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4F8h offset = 3033_04F8h



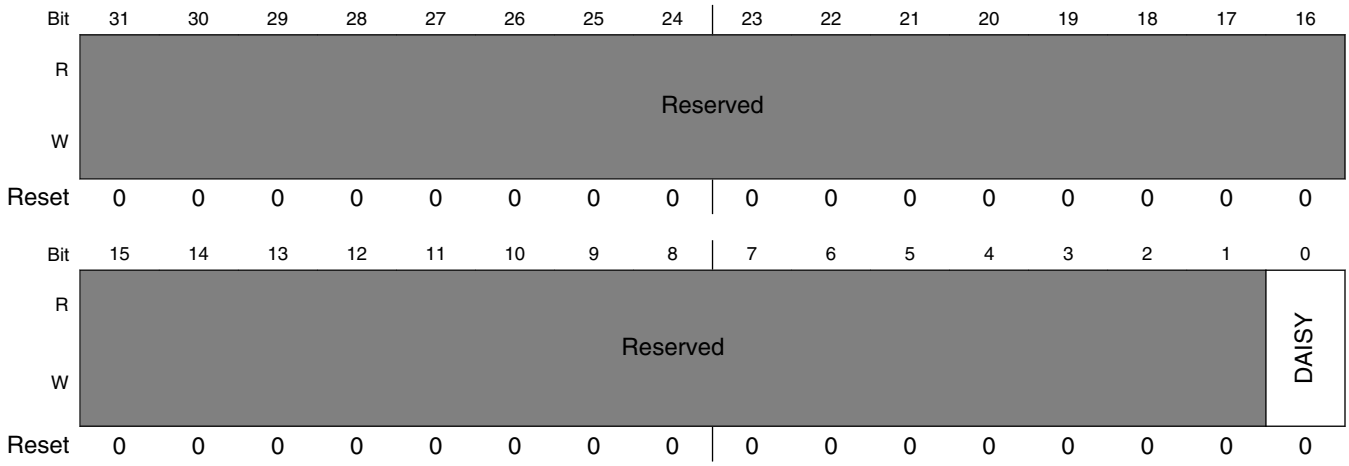
IOMUXC_UART2_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **UART4_RXD_ALT1** — Selecting Pad: UART4_RXD Mode: ALT1 for UART2_RTS_B 1 **UART4_TXD_ALT1** — Selecting Pad: UART4_TXD Mode: ALT1 for UART2_RTS_B

8.2.5.315 UART2_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART2_RXD_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 4FCh offset = 3033_04FCh



IOMUXC_UART2_RXD_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **UART2_RXD_ALT0** — Selecting Pad: UART2_RXD Mode: ALT0 for UART2_RXD 1 **UART2_TXD_ALT0** — Selecting Pad: UART2_TXD Mode: ALT0 for UART2_RXD

8.2.5.316 UART3_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART3_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 500h offset = 3033_0500h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART3_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **ECSPI1_MISO_ALT1** — Selecting Pad: ECSPI1_MISO Mode: ALT1 for UART3_RTS_B 1 **ECSPI1_SS0_ALT1** — Selecting Pad: ECSPI1_SS0 Mode: ALT1 for UART3_RTS_B

8.2.5.317 UART3_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART3_RXD_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 504h offset = 3033_0504h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART3_RXD_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **ECSPI1_SCLK_ALT1** — Selecting Pad: ECSPI1_SCLK Mode: ALT1 for UART3_RXD 01 **ECSPI1_MOSI_ALT1** — Selecting Pad: ECSPI1_MOSI Mode: ALT1 for UART3_RXD 10 **UART3_RXD_ALT0** — Selecting Pad: UART3_RXD Mode: ALT0 for UART3_RXD 11 **UART3_TXD_ALT0** — Selecting Pad: UART3_TXD Mode: ALT0 for UART3_RXD

8.2.5.318 UART4_RTS_B_SELECT_INPUT DAISY Register (IOMUXC_UART4_RTS_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 508h offset = 3033_0508h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_UART4_RTS_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **ECSPI2_MISO_ALT1** — Selecting Pad: ECSPI2_MISO Mode: ALT1 for UART4_RTS_B 1 **ECSPI2_SS0_ALT1** — Selecting Pad: ECSPI2_SS0 Mode: ALT1 for UART4_RTS_B

8.2.5.319 UART4_RXD_SELECT_INPUT DAISY Register (IOMUXC_UART4_RXD_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 50Ch offset = 3033_050Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														DAISY	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

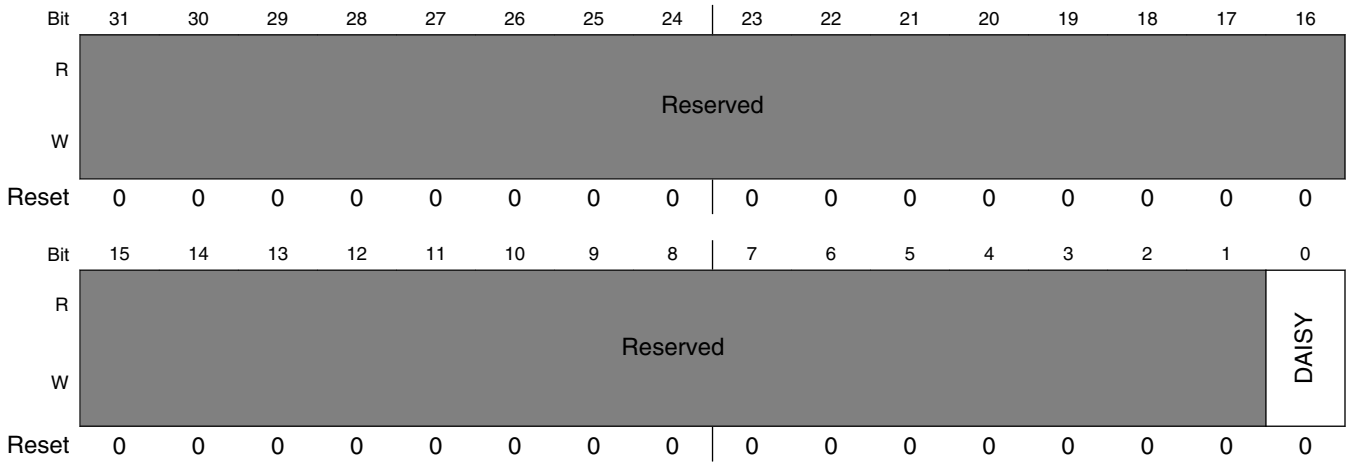
IOMUXC_UART4_RXD_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **ECSPI2_SCLK_ALT1** — Selecting Pad: ECSPI2_SCLK Mode: ALT1 for UART4_RXD 01 **ECSPI2_MOSI_ALT1** — Selecting Pad: ECSPI2_MOSI Mode: ALT1 for UART4_RXD 10 **UART4_RXD_ALT0** — Selecting Pad: UART4_RXD Mode: ALT0 for UART4_RXD 11 **UART4_TXD_ALT0** — Selecting Pad: UART4_RXD Mode: ALT0 for UART4_RXD

8.2.5.320 SAI6_RX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RX_BCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 510h offset = 3033_0510h



IOMUXC_SAI6_RX_BCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI1_RXD4_ALT2** — Selecting Pad: SAI1_RXD4 Mode: ALT2 for SAI6_RX_BCLK 01 **SAI1_TXD4_ALT1** — Selecting Pad: SAI1_TXD4 Mode: ALT1 for SAI6_RX_BCLK

8.2.5.321 SAI6_RXD0_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RXD0_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 514h offset = 3033_0514h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

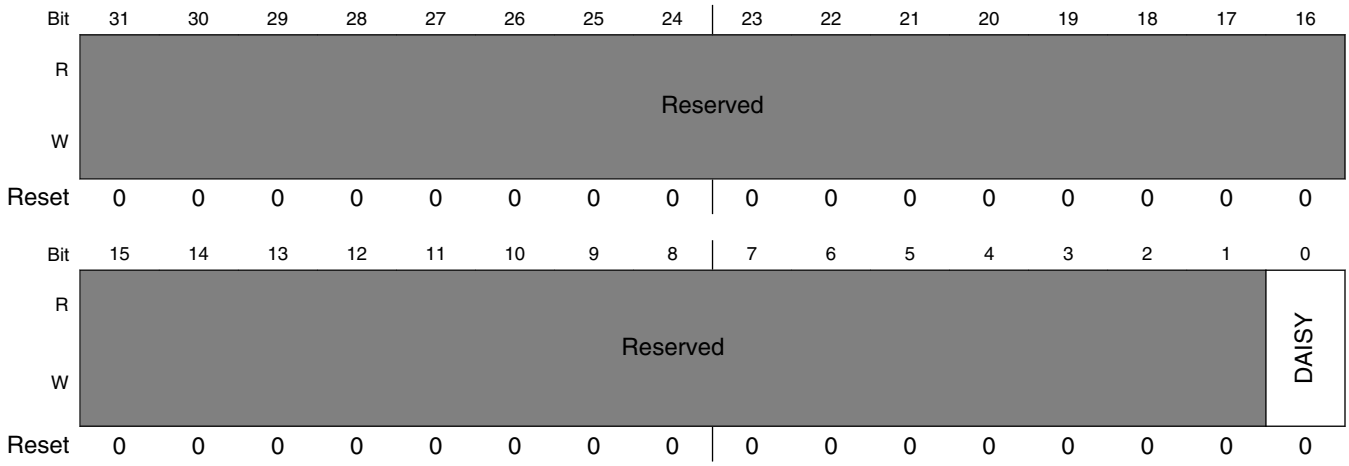
IOMUXC_SAI6_RXD0_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI1_RXD5_ALT2** — Selecting Pad: SAI1_RXD5 Mode: ALT2 for SAI6_RXD0 01 **SAI1_TXD5_ALT1** — Selecting Pad: SAI1_TXD5 Mode: ALT1 for SAI6_RXD0

8.2.5.322 SAI6_RX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI6_RX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 518h offset = 3033_0518h



IOMUXC_SAI6_RX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI1_RXD6_ALT2** — Selecting Pad: SAI1_RXD6 Mode: ALT2 for SAI6_RX_SYNC 01 **SAI1_TXD6_ALT1** — Selecting Pad: SAI1_TXD6 Mode: ALT1 for SAI6_RX_SYNC

8.2.5.323 SAI6_TX_BCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_TX_BCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 51Ch offset = 3033_051Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI6_TX_BCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **SAI1_RXD4_ALT1** — Selecting Pad: SAI1_RXD4 Mode: ALT1 for SAI6_TX_BCLK 1 **SAI1_TXD4_ALT2** — Selecting Pad: SAI1_TXD4 Mode: ALT2 for SAI6_TX_BCLK

8.2.5.324 SAI6_TX_SYNC_SELECT_INPUT DAISY Register (IOMUXC_SAI6_TX_SYNC_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 520h offset = 3033_0520h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_SAI6_TX_SYNC_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **SAI1_RXD6_ALT1** — Selecting Pad: SAI1_RXD6 Mode: ALT1 for SAI6_TX_SYNC 1 **SAI1_TXD6_ALT2** — Selecting Pad: SAI1_TXD6 Mode: ALT2 for SAI6_TX_SYNC

8.2.5.325 PCIE1_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE1_CLKREQ_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 524h offset = 3033_0524h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_PCIE1_CLKREQ_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **I2C4_SCL_ALT2** — Selecting Pad: I2C4_SCL Mode: ALT2 for PCIE1_CLKREQ_B 1 **UART4_RXD_ALT2** — Selecting Pad: UART4_RXD Mode: ALT2 for PCIE1_CLKREQ_B

8.2.5.326 PCIE2_CLKREQ_B_SELECT_INPUT DAISY Register (IOMUXC_PCIE2_CLKREQ_B_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 528h offset = 3033_0528h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

IOMUXC_PCIE2_CLKREQ_B_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **I2C4_SDA_ALT2** — Selecting Pad: I2C4_SDA Mode: ALT2 for PCIE2_CLKREQ_B 1 **UART4_TXD_ALT2** — Selecting Pad: UART4_TXD Mode: ALT2 for PCIE2_CLKREQ_B

8.2.5.327 SAI5_MCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI5_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 52Ch offset = 3033_052Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															DAISY
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

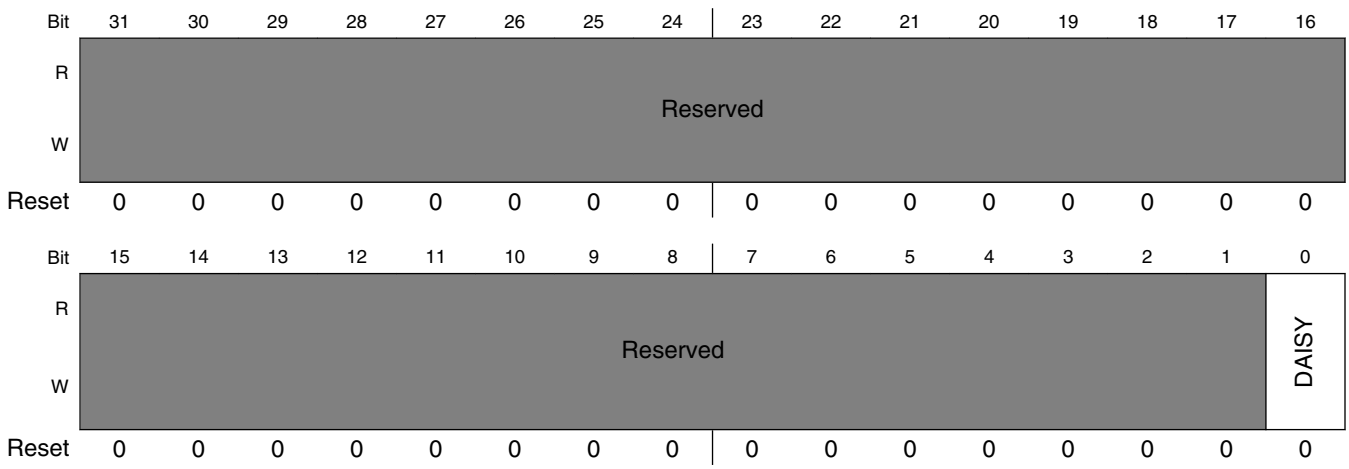
IOMUXC_SAI5_MCLK_SELECT_INPUT field descriptions

Field	Description
31–2 -	This field is reserved. Reserved
DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 00 **SAI5_MCLK_ALT0** — Selecting Pad: SAI5_MCLK Mode: ALT0 for SAI5_MCLK 01 **SAI1_MCLK_ALT1** — Selecting Pad: SAI1_MCLK Mode: ALT1 for SAI5_MCLK 10 **SAI2_MCLK_ALT1** — Selecting Pad: SAI2_MCLK Mode: ALT1 for SAI5_MCLK 11 **SAI3_MCLK_ALT2** — Selecting Pad: SAI3_MCLK Mode: ALT2 for SAI5_MCLK

8.2.5.328 SAI6_MCLK_SELECT_INPUT DAISY Register (IOMUXC_SAI6_MCLK_SELECT_INPUT)

DAISY Register

Address: 3033_0000h base + 530h offset = 3033_0530h



IOMUXC_SAI6_MCLK_SELECT_INPUT field descriptions

Field	Description
31–1 -	This field is reserved. Reserved
0 DAISY	Input Select (DAISY) Field Selecting Pads Involved in Daisy Chain 0 **SAI1_RXD7_ALT1** — Selecting Pad: SAI1_RXD7 Mode: ALT1 for SAI6_MCLK 1 **SAI1_TXD7_ALT1** — Selecting Pad: SAI1_TXD7 Mode: ALT1 for SAI6_MCLK

8.3 General Purpose Input/Output (GPIO)

8.3.1 Overview

The GPIO general-purpose input/output peripheral provides dedicated general-purpose pins that can be configured as either inputs or outputs.

When configured as an output, it is possible to write to an internal register to control the state driven on the output pin. When configured as an input, it is possible to detect the state of the input by reading the state of an internal register. In addition, the GPIO peripheral can produce CORE interrupts.

The GPIO is one of the blocks controlling the IOMUX of the chip.

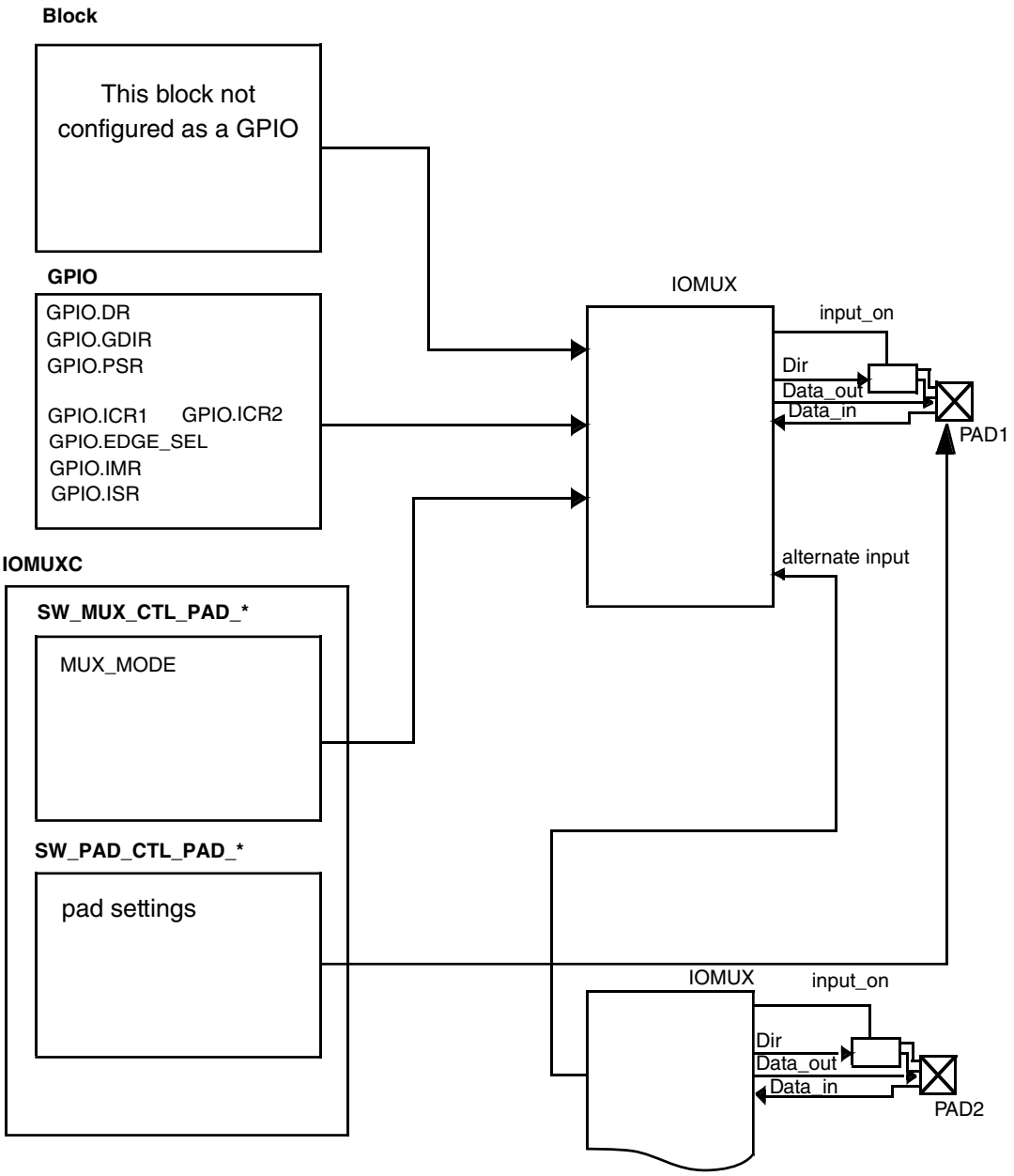


Figure 8-4. Chip IOMUX Scheme

The GPIO functionality is provided through eight registers, an edge-detect circuit, and interrupt generation logic.

The eight registers are:

- Data register (GPIO_DR)
- GPIO direction register (GPIO_GDIR)
- Pad sample register (GPIO_PSR)
- Interrupt control registers (GPIO_ICR1, GPIO_ICR2)

- Edge select register (GPIO_EDGE_SEL)
- Interrupt mask register (GPIO_IMR)
- Interrupt status register (GPIO_ISR)

These registers are described in detail in GPIO Memory Map/Register Definition.

Each GPIO input has a dedicated edge-detect circuit which can be configured through software to detect rising edges, falling edges, logic low-levels or logic high-levels on the input signals. The outputs of the edge detect circuits are optionally masked by setting the corresponding bit in the interrupt mask register (GPIO_IMR). These qualified outputs are OR'ed together to generate two one-bit interrupt lines:

- Combined interrupt indication for GPIOx signals 0 - 15
- Combined interrupt indication for GPIOx signals 16 - 31

In addition, GPIO1 provides visibility to each of its 8 low order interrupt sources (i.e. GPIO1 interrupt n, for n = 0 – 7). However, individual interrupt indications from other GPIOx are not available.

The GPIO edge detection is described further in Interrupt Control Unit.

The GPIO's overall functionality is described further in GPIO Functional Description.

8.3.1.1 Block Diagram

The GPIO subsystem contains multiple GPIO blocks, which can generate and control up to 32 signals for general purpose.

A block diagram of the GPIO is shown in Figure 8-5

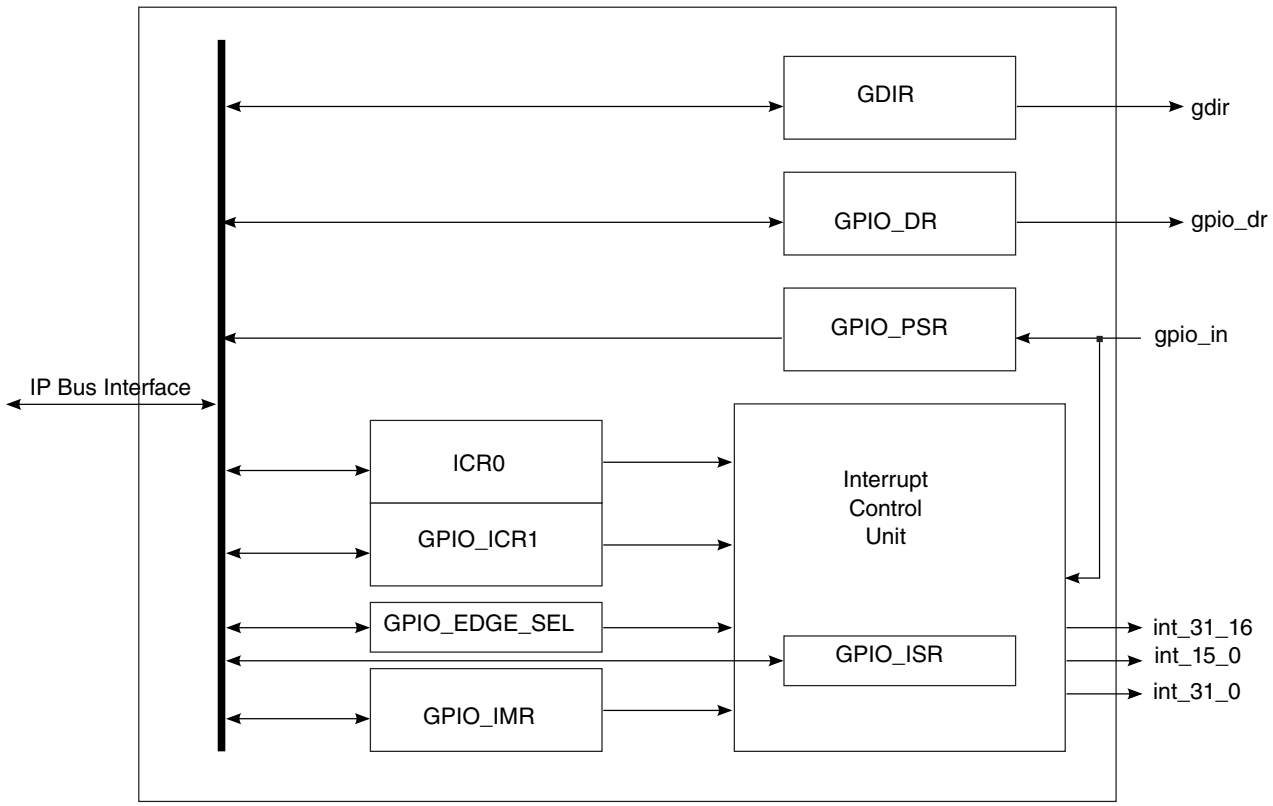


Figure 8-5. GPIO Block Diagram

8.3.1.2 Features

The GPIO includes the following features:

- General purpose input/output logic capabilities:
  - Drives specific data to output using the data register (GPIO_DR)
  - Controls the direction of the signal using the GPIO direction register (GPIO_GDIR)
  - Enables the core to sample the status of the corresponding inputs by reading the pad sample register (GPIO_PSR).
- GPIO interrupt capabilities:
  - Supports up to 32 interrupts
  - Identifies interrupt edges
  - Generates three active-high interrupts to the SoC interrupt controller

8.3.2 Clocks

The table found here describes the clock sources for GPIO.

Please see the clock controller Module for clock setting, configuration and gating information.

Table 8-2. GPIO Clocks

Clock name	Clock Root	Description
ipg_clk_s	ipg_clk_root	Peripheral access clock

8.3.3 GPIO Functional Description

This section provides a complete functional description of the block.

8.3.3.1 GPIO Function

A GPIO signal can operate as a general-purpose input/output when the IOMUX is set to GPIO mode. Each GPIO signal may be independently configured as either an input or an output using the GPIO direction register (GPIO_GDIR).

When configured as an output (GPIO_GDIR bit = 1), the value in the data bit in the GPIO data register (GPIO_DR) is driven on the corresponding GPIO line. When a signal is configured as an input (GPIO_GDIR bit = 0), the state of the input can be read from the corresponding GPIO_PSR bit.

8.3.3.2 GPIO Programming

8.3.3.2.1 GPIO Read Mode

The programming sequence for reading input signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUX Controller (IOMUXC)).
2. Configure GPIO direction register to input (GPIO_GDIR[GDIR] set to 0b).
3. Read value from data reg*ister*/pad status register.

A pseudocode description to read [input3:input0] values is as follows:

```
// SET INPUTS TO GPIO MODE.
write sw_mux_ctl_<input0>_<input1>_<input2>_<input3>, 32'h00000000
// SET GDIR TO INPUT.
write GDIR[31:4,input3_bit, input2_bit, input1_bit, input0_bit,] 32'hxxxxxxx0
// READ INPUT VALUE FROM DR.
read DR
// READ INPUT VALUE FROM PSR.
read PSR
```

NOTE

While the GPIO direction is set to input (GPIO_GDIR = 0), a read access to GPIO_DR does not return GPIO_DR data. Instead, it returns the GPIO_PSR data, which is the corresponding input signal value.

8.3.3.2.2 GPIO Write Mode

The programming sequence for driving output signals should be as follows:

1. Configure IOMUX to select GPIO mode (Via IOMUXC), also enable SION if need to read loopback pad value through PSR
2. Configure GPIO direction register to output (GPIO_GDIR[GDIR] set to 1b).
3. Write value to data register (GPIO_DR).

A pseudocode description to drive 4'b0101 on [output3:output0] is as follows:

```
// SET PADS TO GPIO MODE VIA IOMUX.
write sw_mux_ctl_pad_<output[0-3]>.mux_mode, <GPIO_MUX_MODE>
// Enable loopback so we can capture pad value into PSR in output mode
write sw_mux_ctl_pad_<output[0-3]>.sion, 1
// SET GDIR=1 TO OUTPUT BITS.
write GDIR[31:4,output3_bit,output2_bit, output1_bit, output0_bit,] 32'hxxxxxxxF
// WRITE OUTPUT VALUE=4'b0101 TO DR.
write DR, 32'hxxxxxxx5
// READ OUTPUT VALUE FROM PSR ONLY.
read_cmp PSR, 32'hxxxxxxx5
```

8.3.3.3 Interrupt Control Unit

In addition to the general-purpose input/output function, the edge-detect logic in the GPIO peripheral reflects whether a transition has occurred on a given GPIO signal that is configured as an input (GDIR bit = 0). The interrupt control registers (GPIO_ICR1 and GPIO_ICR2) may be used to independently configure the interrupt condition of each input signal (low-to-high transition, high-to-low transition, low, or high). For information about GPIO_ICR1 and GPIO_ICR2 settings, see GPIO Memory Map/Register Definition.

The interrupt control unit is built of 32 interrupt control subunits, where each subunit handles a single interrupt line.

8.3.4 GPIO Memory Map/Register Definition

There are eight 32-bit GPIO registers. All registers are accessible from the IP interface. Only 32-bit access is supported.

The GPIO memory map is shown in the following table.

GPIO memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3020_0000	GPIO data register (GPIO1_DR)	32	R/W	0000_0000h	8.3.4.1/ 1808
3020_0004	GPIO direction register (GPIO1_GDIR)	32	R/W	0000_0000h	8.3.4.2/ 1809
3020_0008	GPIO pad status register (GPIO1_PSR)	32	R	0000_0000h	8.3.4.3/ 1809
3020_000C	GPIO interrupt configuration register1 (GPIO1_ICR1)	32	R/W	0000_0000h	8.3.4.4/ 1810
3020_0010	GPIO interrupt configuration register2 (GPIO1_ICR2)	32	R/W	0000_0000h	8.3.4.5/ 1814
3020_0014	GPIO interrupt mask register (GPIO1_IMR)	32	R/W	0000_0000h	8.3.4.6/ 1817
3020_0018	GPIO interrupt status register (GPIO1_ISR)	32	w1c	0000_0000h	8.3.4.7/ 1818
3020_001C	GPIO edge select register (GPIO1_EDGE_SEL)	32	R/W	0000_0000h	8.3.4.8/ 1819
3021_0000	GPIO data register (GPIO2_DR)	32	R/W	0000_0000h	8.3.4.1/ 1808
3021_0004	GPIO direction register (GPIO2_GDIR)	32	R/W	0000_0000h	8.3.4.2/ 1809
3021_0008	GPIO pad status register (GPIO2_PSR)	32	R	0000_0000h	8.3.4.3/ 1809
3021_000C	GPIO interrupt configuration register1 (GPIO2_ICR1)	32	R/W	0000_0000h	8.3.4.4/ 1810
3021_0010	GPIO interrupt configuration register2 (GPIO2_ICR2)	32	R/W	0000_0000h	8.3.4.5/ 1814
3021_0014	GPIO interrupt mask register (GPIO2_IMR)	32	R/W	0000_0000h	8.3.4.6/ 1817
3021_0018	GPIO interrupt status register (GPIO2_ISR)	32	w1c	0000_0000h	8.3.4.7/ 1818
3021_001C	GPIO edge select register (GPIO2_EDGE_SEL)	32	R/W	0000_0000h	8.3.4.8/ 1819

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3022_0000	GPIO data register (GPIO3_DR)	32	R/W	0000_0000h	8.3.4.1/ 1808
3022_0004	GPIO direction register (GPIO3_GDIR)	32	R/W	0000_0000h	8.3.4.2/ 1809
3022_0008	GPIO pad status register (GPIO3_PSR)	32	R	0000_0000h	8.3.4.3/ 1809
3022_000C	GPIO interrupt configuration register1 (GPIO3_ICR1)	32	R/W	0000_0000h	8.3.4.4/ 1810
3022_0010	GPIO interrupt configuration register2 (GPIO3_ICR2)	32	R/W	0000_0000h	8.3.4.5/ 1814
3022_0014	GPIO interrupt mask register (GPIO3_IMR)	32	R/W	0000_0000h	8.3.4.6/ 1817
3022_0018	GPIO interrupt status register (GPIO3_ISR)	32	w1c	0000_0000h	8.3.4.7/ 1818
3022_001C	GPIO edge select register (GPIO3_EDGE_SEL)	32	R/W	0000_0000h	8.3.4.8/ 1819
3023_0000	GPIO data register (GPIO4_DR)	32	R/W	0000_0000h	8.3.4.1/ 1808
3023_0004	GPIO direction register (GPIO4_GDIR)	32	R/W	0000_0000h	8.3.4.2/ 1809
3023_0008	GPIO pad status register (GPIO4_PSR)	32	R	0000_0000h	8.3.4.3/ 1809
3023_000C	GPIO interrupt configuration register1 (GPIO4_ICR1)	32	R/W	0000_0000h	8.3.4.4/ 1810
3023_0010	GPIO interrupt configuration register2 (GPIO4_ICR2)	32	R/W	0000_0000h	8.3.4.5/ 1814
3023_0014	GPIO interrupt mask register (GPIO4_IMR)	32	R/W	0000_0000h	8.3.4.6/ 1817
3023_0018	GPIO interrupt status register (GPIO4_ISR)	32	w1c	0000_0000h	8.3.4.7/ 1818
3023_001C	GPIO edge select register (GPIO4_EDGE_SEL)	32	R/W	0000_0000h	8.3.4.8/ 1819
3024_0000	GPIO data register (GPIO5_DR)	32	R/W	0000_0000h	8.3.4.1/ 1808
3024_0004	GPIO direction register (GPIO5_GDIR)	32	R/W	0000_0000h	8.3.4.2/ 1809
3024_0008	GPIO pad status register (GPIO5_PSR)	32	R	0000_0000h	8.3.4.3/ 1809
3024_000C	GPIO interrupt configuration register1 (GPIO5_ICR1)	32	R/W	0000_0000h	8.3.4.4/ 1810
3024_0010	GPIO interrupt configuration register2 (GPIO5_ICR2)	32	R/W	0000_0000h	8.3.4.5/ 1814
3024_0014	GPIO interrupt mask register (GPIO5_IMR)	32	R/W	0000_0000h	8.3.4.6/ 1817

Table continues on the next page...

GPIO memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3024_0018	GPIO interrupt status register (GPIO5_ISR)	32	w1c	0000_0000h	8.3.4.7/ 1818
3024_001C	GPIO edge select register (GPIO5_EDGE_SEL)	32	R/W	0000_0000h	8.3.4.8/ 1819

8.3.4.1 GPIO data register (GPIOx_DR)

The 32-bit GPIO_DR register stores data that is ready to be driven to the output lines. If the IOMUXC is in GPIO mode and a given GPIO direction bit is set, then the corresponding DR bit is driven to the output. If a given GPIO direction bit is cleared, then a read of GPIO_DR reflects the value of the corresponding signal.Two wait states are required in read access for synchronization.

The results of a read of a DR bit depends on the IOMUXC input mode settings and the corresponding GDIR bit as follows:

- If GDIR[*n*] is set and IOMUXC input mode is GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is GPIO, then reading DR[*n*] returns the corresponding input signal's value.
- If GDIR[*n*] is set and IOMUXC input mode is not GPIO, then reading DR[*n*] returns the contents of DR[*n*].
- If GDIR[*n*] is cleared and IOMUXC input mode is not GPIO, then reading DR[*n*] always returns zero.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_DR field descriptions

Field	Description
DR	Data bits. This register defines the value of the GPIO output when the signal is configured as an output (GDIR[n]=1). Writes to this register are stored in a register. Reading GPIO_DR returns the value stored in the register if the signal is configured as an output (GDIR[n]=1), or the input signal's value if configured as an input (GDIR[n]=0). **NOTE:** The I/O multiplexer must be configured to GPIO mode for the GPIO_DR value to connect with the signal. Reading the data register with the input path disabled always returns a zero value.

8.3.4.2 GPIO direction register (GPIOx_GDIR)

GPIO_GDIR functions as direction control when the IOMUXC is in GPIO mode. Each bit specifies the direction of a one-bit signal. The mapping of each DIR bit to a corresponding SoC signal is determined by the SoC's pin assignment and the IOMUX table. For more details consult the IOMUXC chapter.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GDIR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_GDIR field descriptions

Field	Description
GDIR	GPIO direction bits. Bit n of this register defines the direction of the GPIO[n] signal. **NOTE:** GPIO_GDIR affects only the direction of the I/O signal when the corresponding bit in the I/O MUX is configured for GPIO. 0 **INPUT** — GPIO is configured as input. 1 **OUTPUT** — GPIO is configured as output.

8.3.4.3 GPIO pad status register (GPIOx_PSR)

GPIO_PSR is a read-only register. Each bit stores the value of the corresponding input signal (as configured in the IOMUX). This register is clocked with the ipg_clk_s clock, meaning that the input signal is sampled only when accessing this location. Two wait states are required any time this register is accessed for synchronization.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PSR																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_PSR field descriptions

Field	Description
PSR	GPIO pad status bits (status bits). Reading GPIO_PSR returns the state of the corresponding input signal. Settings:

GPIOx_PSR field descriptions (continued)

Field	Description
	NOTE: The IOMUXC must be configured to GPIO mode for GPIO_PSR to reflect the state of the corresponding signal.

8.3.4.4 GPIO interrupt configuration register1 (GPIOx_ICR1)

GPIO_ICR1 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR15		ICR14		ICR13		ICR12		ICR11		ICR10		ICR9		ICR8	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR7		ICR6		ICR5		ICR4		ICR3		ICR2		ICR1		ICR0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ICR1 field descriptions

Field	Description
31–30 ICR15	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 15. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
29–28 ICR14	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 14. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
27–26 ICR13	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 13. Settings:

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
25–24 ICR12	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 12. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
23–22 ICR11	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 11. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
21–20 ICR10	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 10. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
19–18 ICR9	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 9. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
17–16 ICR8	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 8. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows:

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
15–14 ICR7	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 7. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
13–12 ICR6	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 6. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
11–10 ICR5	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 5. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
9–8 ICR4	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 4. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
7–6 ICR3	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 3. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive.

Table continues on the next page...

GPIOx_ICR1 field descriptions (continued)

Field	Description
	01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
5–4 ICR2	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 2. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
3–2 ICR1	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 1. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
ICR0	Interrupt configuration 1 fields. This register controls the active condition of the interrupt function for GPIO interrupt 0. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

8.3.4.5 GPIO interrupt configuration register2 (GPIOx_ICR2)

GPIO_ICR2 contains 16 two-bit fields, where each field specifies the interrupt configuration for a different input signal.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICR31		ICR30		ICR29		ICR28		ICR27		ICR26		ICR25		ICR24	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICR23		ICR22		ICR21		ICR20		ICR19		ICR18		ICR17		ICR16	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ICR2 field descriptions

Field	Description
31–30 ICR31	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 31. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
29–28 ICR30	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 30. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
27–26 ICR29	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 29. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
25–24 ICR28	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 28. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
23–22 ICR27	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 27. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
21–20 ICR26	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 26. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
19–18 ICR25	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 25. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
17–16 ICR24	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 24. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
15–14 ICR23	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 23. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
13–12 ICR22	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 22. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
11–10 ICR21	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 21. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
9–8 ICR20	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 20. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
7–6 ICR19	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 19. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

Table continues on the next page...

GPIOx_ICR2 field descriptions (continued)

Field	Description
5–4 ICR18	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 18. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
3–2 ICR17	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 17. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.
ICR16	Interrupt configuration 2 fields. This register controls the active condition of the interrupt function for GPIO interrupt 16. Settings: Bits ICRn[1:0] determine the interrupt condition for signal n as follows: 00 **LOW_LEVEL** — Interrupt n is low-level sensitive. 01 **HIGH_LEVEL** — Interrupt n is high-level sensitive. 10 **RISING_EDGE** — Interrupt n is rising-edge sensitive. 11 **FALLING_EDGE** — Interrupt n is falling-edge sensitive.

8.3.4.6 GPIO interrupt mask register (GPIOx_IMR)

GPIO_IMR contains masking bits for each interrupt line.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_IMR field descriptions

Field	Description
IMR	Interrupt Mask bits. This register is used to enable or disable the interrupt function on each of the 32 GPIO signals. Settings:

GPIOx_IMR field descriptions (continued)

Field	Description
	Bit IMR[n] (n=0...31) controls interrupt n as follows: 0 **MASKED** — Interrupt n is disabled. 1 **UNMASKED** — Interrupt n is enabled.

8.3.4.7 GPIO interrupt status register (GPIOx_ISR)

The GPIO_ISR functions as an interrupt status indicator. Each bit indicates whether an interrupt condition has been met for the corresponding input signal. When an interrupt condition is met (as determined by the corresponding interrupt condition register field), the corresponding bit in this register is set. Two wait states are required in read access for synchronization. One wait state is required for reset.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ISR																															
W	w1c																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_ISR field descriptions

Field	Description
ISR	Interrupt status bits - Bit n of this register is asserted (active high) when the active condition (as determined by the corresponding ICR bit) is detected on the GPIO input and is waiting for service. The value of this register is independent of the value in GPIO_IMR. When the active condition has been detected, the corresponding bit remains set until cleared by software. Status flags are cleared by writing a 1 to the corresponding bit position.

8.3.4.8 GPIO edge select register (GPIOx_EDGE_SEL)

GPIO_EDGE_SEL may be used to override the ICR registers' configuration. If the GPIO_EDGE_SEL bit is set, then a rising edge or falling edge in the corresponding signal generates an interrupt. This register provides backward compatibility. On reset all bits are cleared (ICR is not overridden).

Address: Base address + 1Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GPIO_EDGE_SEL																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPIOx_EDGE_SEL field descriptions

Field	Description
GPIO_EDGE_SEL	Edge select. When GPIO_EDGE_SEL[*n*] is set, the GPIO disregards the ICR[*n*] setting, and detects any edge on the corresponding input signal.

Chapter 9
External Memory

9.1 External Memory Overview

9.1.1 External Memory Overview

9.1.1.1 DRAM Interface

The chip has a single 16/32-bit DRAM controller.

The DRAM controller and PHY are licensed from SNPS. The key features of the DRAM controller and PHY include:

- LPDDR4-3200 in POP BGA package
- LPDDR4-3200, DDR4-2400, and DDR3L-1600 in Non-POP BGA package
- 32/16-bit DRAM interface
- 800MHz 128-bit AXI bus
- Up to 8GB of memory capacity
- Support various low power modes, clock and power gated operations, that are defined for i.MX 8M Quad. In addition, it shall be able to place the external DRAM into and out of its self-refresh mode as requested by different low-power operating modes

9.1.1.2 eSD/eMMC/SDIO Support

The chip has two Ultra Secured Digital Host Controller (uSDHC) modules for the SD/eMMC interface. It provides the interface between the host system and SD/SDIO/MMC cards. The key features include:

- Supports SD/SDIO standard, up to version 3.0
- Supports MMC standard, up to version 5.1
- Supports 3.3V and 1.8V operation, but does not support 1.2V operation

- Supports 1-bit / 4-bit SD and SDIO modes, 1-bit / 4-bit / 8-bit MMC modes
  - Up to 400 Mbps of data transfer for SDIO cards using 4 parallel data lines in SDR mode
  - Up to 800 Mbps of data transfer for SDIO card using 4 parallel data lines in DDR mode
  - Up to 400 Mbps of data transfer for SDXC cards using 4 parallel data lines in SDR mode
  - Up to 800 Mbps of data transfer for SDXC card using 4 parallel data lines in DDR mode
  - Up to 1600 Mbps of data transfer for MMC cards using 8 parallel data lines in SDR mode
  - Up to 3200 Mbps of data transfer for MMC cards using 8 parallel data lines in DDR mode
- One uSDHC controller (SD1) can support up to 8-bit interface, the other controller (SD2) can only support up to 4-bit interface.

9.1.1.3 Raw NAND Support

The Raw NAND Flash controller consists of three components:

- GPMI as the NAND controller pin interface
- APBH_DMA as the DMA engine that drives the GPMI module
- BCH as the 62-bit error correction hardware engine with an AXI bus master and a private connection to GPMI.

The Raw NAND Flash controller support following key features:

- 8-bit NAND FLASH, up to 4 devices supported by 4 chip-selects and 1 ganged ready/busy
- ONFI 2.x complaint, synchronous clock rate of up to 100 MHz with data rate of up to 200 MB/s
- Support ONFI NAND for Micron and Hynix and Toggle NAND for Toshiba and Samsung
- BCH62 for ECC, up to 200MB/s

9.1.1.4 Quad SPI Interface

There is one Quad Serial Peripheral Interface (QuadSPI) block that acts as an interface to one or two external serial flash devices, each with up to four bidirectional data lines. The key features include:

- Each channel can be configured as 1/2/4-bit operation
- Support both dual-channel or single-channel operation

- Support both SDR mode and DDR mode
- Support up to 80MHz SDR Mode and 50MHz DDR Mode

9.2 DRC Performance Monitor (DRC_PERF_MON)

9.2.1 Introduction

This section provides an introduction to the DDR performance monitor.

9.2.2 Functional description

The main functions of the DDR Performance Monitor include:

- Monitor the performance signals provided by DDR controller
- Sample the MRR read data (LPDDR4) or MPR read data (DDR4) after DDR controller issues an Mode register read command

9.2.2.1 Performance Counter

This module contain a number of counters. One of them is a special counter which counts "time". The rest of them are used for signal monitoring.

The special counter (COUNTER0) will be "cleared" when COUNTER0_CTRL[counter_0_clear] is set. It will start to increment if COUNTER0_CTRL[counter_0_enable] is set to high. This counter will overflow when the value reaches 0xFFFF_FFFF. Once this happens, all the counters will lock and an interrupt will be generated.

The counters 1-3 used for signal monitoring operates the same way as the special counter. They will be cleared when counter_n_clear is high, and become active when both counter_0 and counter_n are enabled and not overflowing. When active, counter_n will increase if the monitored signal is high.

The signal being monitored can be selected from the below list via counter_n_csv:

- WAQ_COUNT_0[3:0]: the number of used positions in the write queue FIFO
- WAQ_POP_0: one write granted by port arbiter
- RAQ_WCOUNT_0[3:0]: number of used positions in the read queue
- RAQ_POP_0: one read granted by port arbiter
- LPR_CREDIT_CNT[6:0]: number of available low priority read CAM slots free

- HPR_CREDIT_CNT[6:0]: number of available high priority read CAM slots free
- WR_CREDIT_CNT[6:0]: number of available write CAM slots free
- stat_ddrc_reg_selfref_type[1:0] self-refresh status and type:
  - 00 - not in self refresh
  - 11 - self-refresh by automatic
  - 10 - caused by SW/HW interface
- perf_hif_rd: asserts for every read command to the DDRC
- perf_hif_wr: asserts for every write command to the DDRC
- perf_hif_rmw: asserts for every read modify write to the DDRC
- perf_hif_hi_pri_rd: asserts for every high priority read
- perf_hpr_req_with_nocredit: asserts when there is a high priority read request not served due to no available credit
- perf_hpr_xact_when_critical: asserts for every high priority read transaction that is scheduled when the high priority queue is in critical state
- perf_lpr_req_with_nocredit: asserts when there is a low priority read request not served due to no available credit
- perf_lpr_xact_when_critical: asserts for every low priority read transaction that is scheduled when the low priority queue is in critical state
- perf_wr_xact_when_critical: asserts for every write transaction that is scheduled when the write queue is in critical state
- perf_dfi_rd_data_cycles: assert for every read data cycle
- perf_dfi_wr_data_cycles: assert for every write data cycle
- perf_rdwr_transitions: assert for every read->write or write->read transition
- perf_op_is_precharge: assert for every precharge command issued
- perf_op_is_activate: assert for every active command issued
- perf_op_is_load_mode: assert for every MRW or MRR command issued
- perf_op_is_mwr: assert for every masked write command issued
- perf_op_is_rd: assert for every read command issued
- perf_op_is_rd_activate: assert for every read active issued
- perf_op_is_refresh: assert for every refresh command issued
- perf_op_is_wr: assert for every write command issued
- perf_raw_hazard: asserts for every read-after-write collision that happens in the controller

9.2.2.2 Mode register read

DDR performance monitor will sample MRR read data (LPDDR4) or MPR read data (DDR4) after DDRC issues a mode register command.

Serial mode and parallel mode are supported for DDR4 MPR read and it can be selected via MRR_0_DATA[mrr_mode_sel].

9.2.3 Memory Map and register definition

This section includes the DRC_PERF_MON module memory map and detailed descriptions of all registers.

9.2.3.1 DRC_PERF_MON register descriptions

9.2.3.1.1 DRC_PERF_MON Memory map

drc_perf_monitor base address: 0h

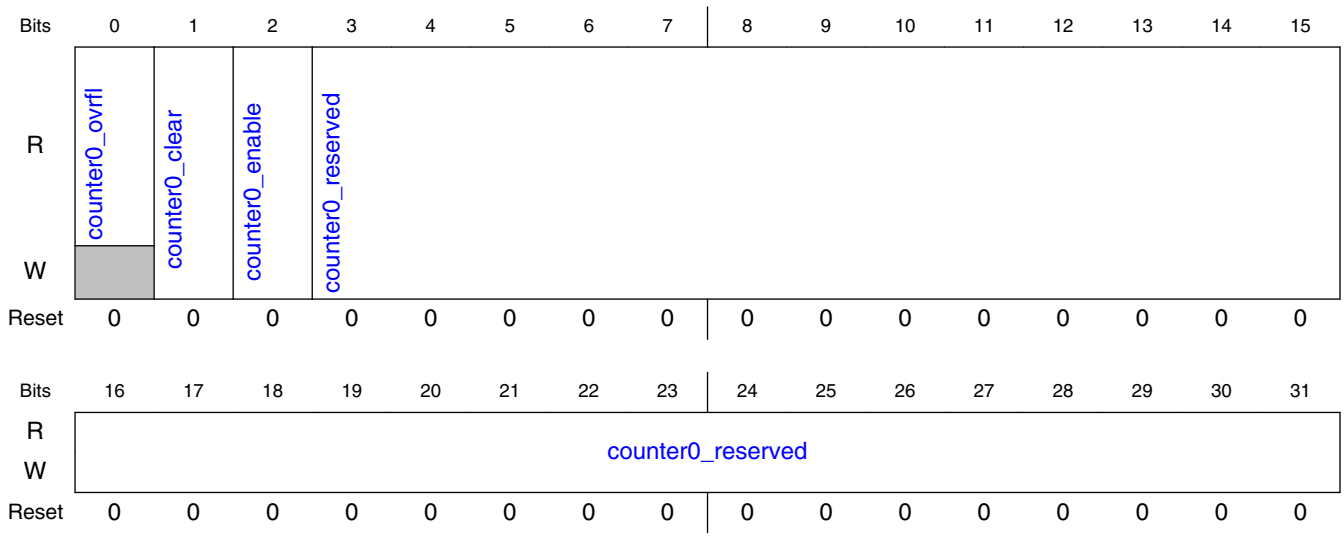
Offset	Register	Width (In bits)	Access	Reset value
0h	Counter0 control (COUNTER0_CTRL)	32	RW	0000_0000h
4h	Counter1 control (COUNTER1_CTRL)	32	RW	Table 9-
8h	Counter2 control (COUNTER2_CTRL)	32	RW	Table 9-
Ch	Counter3 control (COUNTER3_CTRL)	32	RW	Table 9-
20h	Counter0 Data (COUNTER0_DATA)	32	RO	0000_0000h
24h	Counter1 Data (COUNTER1_DATA)	32	RO	0000_0000h
28h	Counter2 Data (COUNTER2_DATA)	32	RO	0000_0000h
2Ch	Counter3 Data (COUNTER3_DATA)	32	RO	0000_0000h
40h	MRR0 data (MRR0_DATA)	32	RW	0000_0000h
44h	MRR1 data (MRR1_DATA)	32	RW	0000_0000h

9.2.3.1.2 Counter0 control (COUNTER0_CTRL)

9.2.3.1.2.1 Offset

Register	Offset
COUNTER0_CTRL	0h

9.2.3.1.2.2 Diagram



9.2.3.1.2.3 Fields

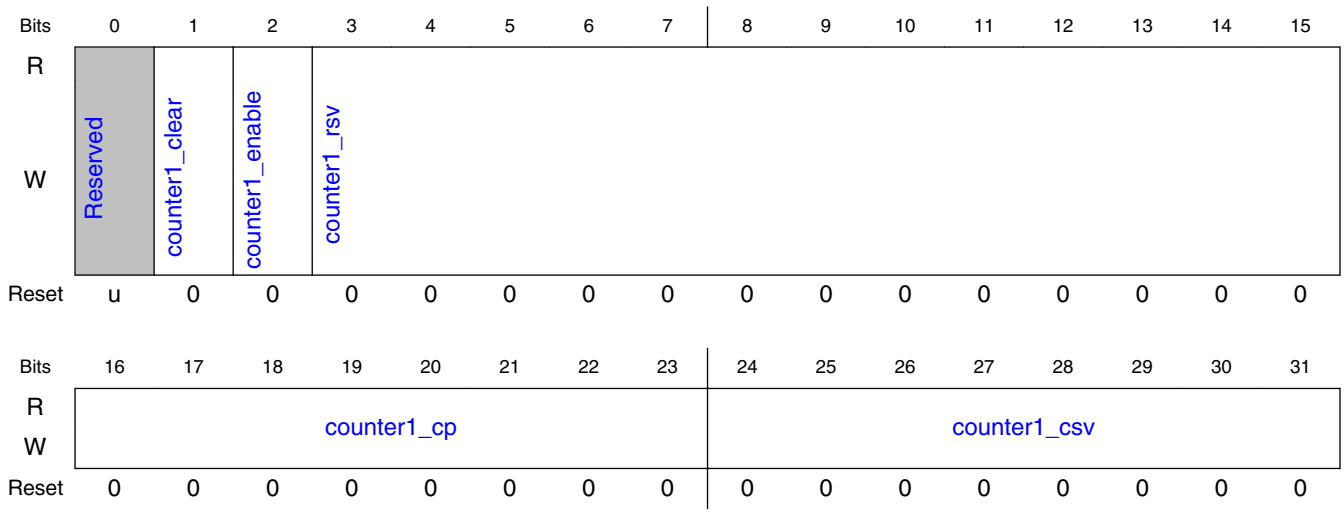
Field	Function
0 counter0_ovrfl	counter0_overflow Asserted when Counter 0 overflows
1 counter0_clear	counter0_clear Clear Counter 0 when asserted
2 counter0_enable	counter0_enable Enable Counter 0 when asserted
3-31 counter0_reserved	reserved

9.2.3.1.3 Counter1 control (COUNTER1_CTRL)

9.2.3.1.3.1 Offset

Register	Offset
COUNTER1_CTRL	4h

9.2.3.1.3.2 Diagram



9.2.3.1.3.3 Fields

Field	Function
0 —	Reserved
1 counter1_clear	counter1_clear Clear Counter 1 when asserted
2 counter1_enable	counter1_enable Enable Counter 1 when asserted
3-15 counter1_rsv	reserved
16-23 counter1_cp	counter1_conter_parameter Counter Parameter
24-31 counter1_csv	counter1_count_value_select 8'h00: use counter_1_cp for overflow calculation If configured with a non-0 value, it will select the signal to be monitored in counter 1: 8'h01: stat_ddrc_reg_selfref_type == counter_1_cp[1:0] 8'h04: raq_pop_0 8'h05: waq_pop_0 8'h06: raqb_pop_0 8'h07: raqb_wcount_0 == counter_1_cp[3:0] 8'h08: raqr_wcount_0 == counter_1_cp[3:0] 8'h09: waq_wcount_0 == counter_1_cp[3:0] 8'h10: lpr_credit_cnt == counter_1_cp[6:0] 8'h11: hpr_credit_cnt == counter_1_cp[6:0]

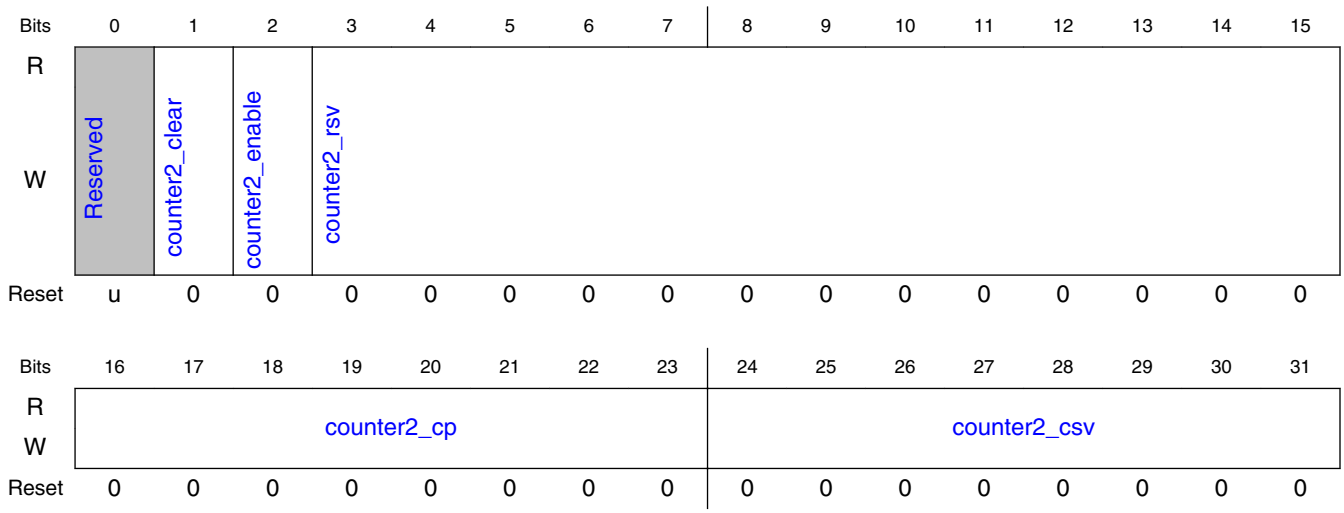
Field	Function
	8'h12: wr_credit_cnt == counter_1_cp[6:0] 8'h20: perf_hif_rd 8'h21: perf_hif_wr 8'h22: perf_hif_rmw 8'h23: perf_hif_hi_pri_rd 8'h24: perf_hpr_req_with_nocredit 8'h25: perf_hpr_xact_when_critical 8'h26: perf_lpr_req_with_nocredit 8'h27: perf_lpr_xact_when_critical 8'h29: perf_wr_xact_when_critical 8'h2a: perf_dfi_rd_data_cycles 8'h2b: perf_dfi_wr_data_cycles 8'h30: perf_rdwr_transitions 8'h31: perf_op_is_precharge 8'h32: perf_op_is_activate 8'h33: perf_op_is_load_mode 8'h34: perf_op_is_mwr 8'h35: perf_op_is_rd 8'h36: perf_op_is_rd_activate 8'h37: perf_op_is_refresh 8'h38: perf_op_is_wr 8'h39: perf_raw_hazard

9.2.3.1.4 Counter2 control (COUNTER2_CTRL)

9.2.3.1.4.1 Offset

Register	Offset
COUNTER2_CTRL	8h

9.2.3.1.4.2 Diagram



9.2.3.1.4.3 Fields

Field	Function
0 —	Reserved
1 counter2_clear	counter2_clear Clear Counter 2 when asserted
2 counter2_enable	counter2_enable Enable Counter 2 when asserted
3-15 counter2_rsv	reserved
16-23 counter2_cp	counter2 count parameter Counter Parameter
24-31 counter2_csv	counter2_count_value_select 8'h00: use counter_2_cp for overflow calculation If configured with a non-0 value, it will select the signal to be monitored in counter 2: 8'h01: stat_ddrc_reg_selfref_type == counter_2_cp[1:0] 8'h04: raq_pop_0 8'h05: waq_pop_0 8'h06: raqb_pop_0 8'h07: raqb_wcount_0 == counter_2_cp[3:0] 8'h08: raqr_wcount_0 == counter_2_cp[3:0] 8'h09: waq_wcount_0 == counter_2_cp[3:0] 8'h10: lpr_credit_cnt == counter_2_cp[6:0] 8'h11: hpr_credit_cnt == counter_2_cp[6:0]

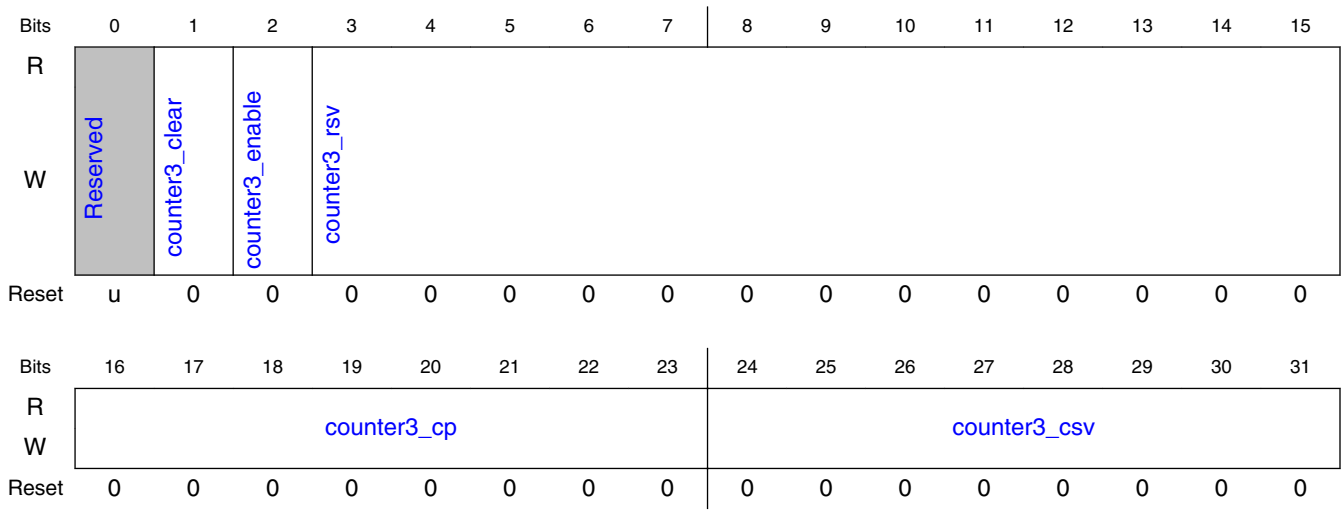
Field	Function
	8'h12: wr_credit_cnt == counter_2_cp[6:0] 8'h20: perf_hif_rd 8'h21: perf_hif_wr 8'h22: perf_hif_rmw 8'h23: perf_hif_hi_pri_rd 8'h24: perf_hpr_req_with_nocredit 8'h25: perf_hpr_xact_when_critical 8'h26: perf_lpr_req_with_nocredit 8'h27: perf_lpr_xact_when_critical 8'h29: perf_wr_xact_when_critical 8'h2a: perf_dfi_rd_data_cycles 8'h2b: perf_dfi_wr_data_cycles 8'h30: perf_rdwr_transitions 8'h31: perf_op_is_precharge 8'h32: perf_op_is_activate 8'h33: perf_op_is_load_mode 8'h34: perf_op_is_mwr 8'h35: perf_op_is_rd 8'h36: perf_op_is_rd_activate 8'h37: perf_op_is_refresh 8'h38: perf_op_is_wr 8'h39: perf_raw_hazard

9.2.3.1.5 Counter3 control (COUNTER3_CTRL)

9.2.3.1.5.1 Offset

Register	Offset
COUNTER3_CTRL	Ch

9.2.3.1.5.2 Diagram



9.2.3.1.5.3 Fields

Field	Function
0 —	Reserved
1 counter3_clear	counter3 clear Clear Counter 3 when asserted
2 counter3_enable	counter3_enable Enable Counter 3 when asserted
3-15 counter3_rsv	reserved
16-23 counter3_cp	counter3 count parameter
24-31 counter3_csv	counter3_count_value_select 8'h00: use counter_3_cp for overflow calculation If configured with a non-0 value, it will select the signal to be monitored in counter 3: 8'h01: stat_ddrc_reg_selfref_type == counter_3_cp[1:0] 8'h04: raq_pop_0 8'h05: waq_pop_0 8'h06: raqb_pop_0 8'h07: raqb_wcount_0 == counter_3_cp[3:0] 8'h08: raqr_wcount_0 == counter_3_cp[3:0] 8'h09: waq_wcount_0 == counter_3_cp[3:0] 8'h10: lpr_credit_cnt == counter_3_cp[6:0] 8'h11: hpr_credit_cnt == counter_3_cp[6:0]

Field	Function
	8'h12: wr_credit_cnt == counter_3_cp[6:0] 8'h20: perf_hif_rd 8'h21: perf_hif_wr 8'h22: perf_hif_rmw 8'h23: perf_hif_hi_pri_rd 8'h24: perf_hpr_req_with_nocredit 8'h25: perf_hpr_xact_when_critical 8'h26: perf_lpr_req_with_nocredit 8'h27: perf_lpr_xact_when_critical 8'h29: perf_wr_xact_when_critical 8'h2a: perf_dfi_rd_data_cycles 8'h2b: perf_dfi_wr_data_cycles 8'h30: perf_rdwr_transitions 8'h31: perf_op_is_precharge 8'h32: perf_op_is_activate 8'h33: perf_op_is_load_mode 8'h34: perf_op_is_mwr 8'h35: perf_op_is_rd 8'h36: perf_op_is_rd_activate 8'h37: perf_op_is_refresh 8'h38: perf_op_is_wr 8'h39: perf_raw_hazard

9.2.3.1.6 Counter0 Data (COUNTER0_DATA)

9.2.3.1.6.1 Offset

Register	Offset
COUNTER0_DATA	20h

9.2.3.1.6.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	counter0_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	counter0_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.3.1.6.3 Fields

Field	Function
0-31 counter0_data	counter0 data

9.2.3.1.7 Counter1 Data (COUNTER1_DATA)

9.2.3.1.7.1 Offset

Register	Offset
COUNTER1_DATA	24h

9.2.3.1.7.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	counter1_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	counter1_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.3.1.7.3 Fields

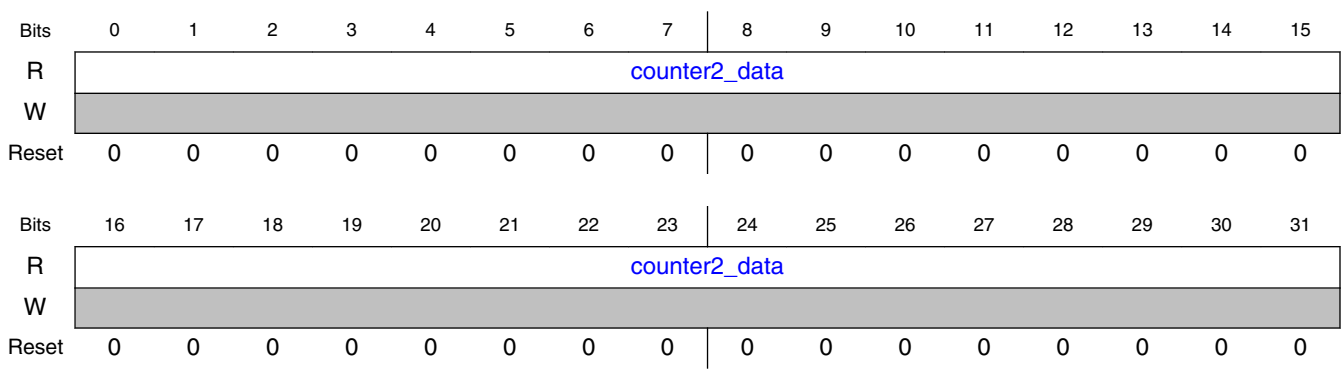
Field	Function
0-31 counter1_data	counter1 data

9.2.3.1.8 Counter2 Data (COUNTER2_DATA)

9.2.3.1.8.1 Offset

Register	Offset
COUNTER2_DATA	28h

9.2.3.1.8.2 Diagram



9.2.3.1.8.3 Fields

Field	Function
0-31 counter2_data	counter2 data

9.2.3.1.9 Counter3 Data (COUNTER3_DATA)

9.2.3.1.9.1 Offset

Register	Offset
COUNTER3_DATA	2Ch

9.2.3.1.9.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R	counter3_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R	counter3_data															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.3.1.9.3 Fields

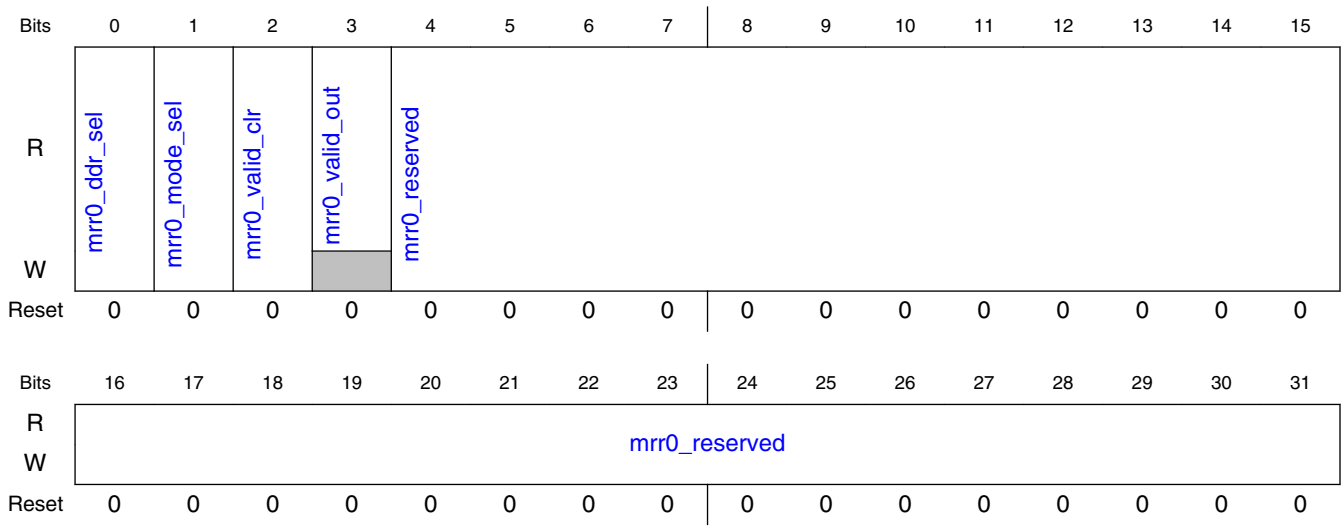
Field	Function
0-31 counter3_data	counter3 data

9.2.3.1.10 MRR0 data (MRR0_DATA)

9.2.3.1.10.1 Offset

Register	Offset
MRR0_DATA	40h

9.2.3.1.10.2 Diagram



9.2.3.1.10.3 Fields

Field	Function
0 mrr0_ddr_sel	mrr0 ddr selection 0b - DDR4 mode 1b - LPDDR4 mode
1 mrr0_mode_sel	mrr0 mode selection 0b - Parallel Mode (only valid on DDR4 mode) 1b - Serial Mode (only valid on LPDDR3/LPDDR4 mode)
2 mrr0_valid_clr	mrr0 valid clear 1b - Clear previous MRR result
3 mrr0_valid_out	mrr0 data valid 1b - MRR result is valid
4-31 mrr0_reserved	mrr0 reserved

9.2.3.1.11 MRR1 data (MRR1_DATA)

9.2.3.1.11.1 Offset

Register	Offset
MRR1_DATA	44h

9.2.3.1.11.2 Diagram

Bits	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
R W	mrr1_data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
R W	mrr1_data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.2.3.1.11.3 Fields

Field	Function
0-31 mrr1_data	mrr1 data

9.3 DDR Controller (DDRC)

9.3.1 Introduction

The DDRC (DDR Controller) is a very low power, high efficiency, low-latency and high performance memory controller for interfacing DDR based memories.

The following types of SDRAMs are supported by the DDRC:

- DDR4
- LP-DDR4

9.3.1.1 Block diagram

The following shows the block diagram for the DDR Controller.

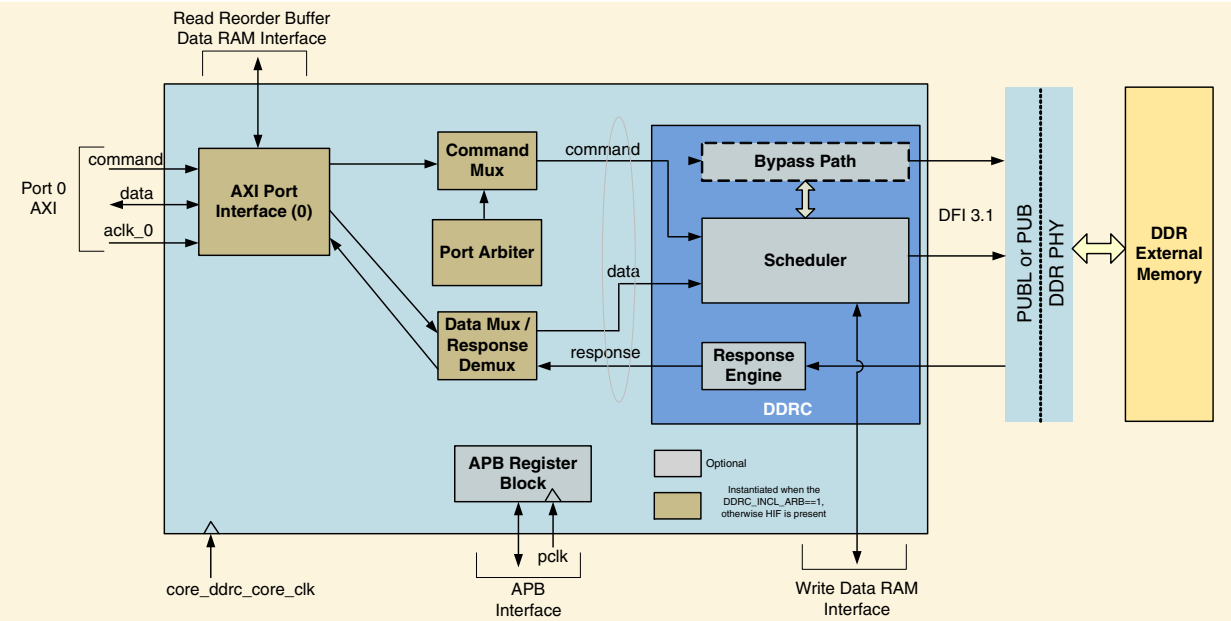


Figure 9-1. DDRC block diagram

The PUB /PUBL is the PHY Utility Block is an adaption layer between the controller and the PHY handling control function.

9.3.1.2 Features

The DDRC includes the following features:

- Direct software request control or programmable internal control for ZQ short calibration cycles
- Support for ZQ long calibration after self-refresh exit
- Direct software request control or programmable internal control for ZQ calibration cycles
- Support for ZQ Reset feature through software
- Dynamic scheduling to optimize bandwidth and latency
- Read and write buffers in fully associative CAMs, 32 each
- Delayed writes for optimum performance on SDRAM data bus
- Out of order execution of commands maximises the SDRAM efficiency
- Hardware configurable and software programmable Quality of Service (QoS) support:

    - Support for three traffic classes on read commands—high priority reads, variable priority reads and low priority reads
    - Support for two traffic classes on write commands—normal priority writes and variable priority writes
    - Support for port urgent and port throttling control
- Programmable SDRAM parameters
- Supports BL16 burst length
- Supports 1,2 memory ranks
- Control options to avoid starvation of lower priorities
- Guaranteed coherency for write-after-read (WAR) and read-after-write (RAW) hazards
- Write combine to allow multiple writes to the same address to be combined into a single write to SDRAM; supported for same starting address
- Paging policy selectable by configuration registers as any of the following:
    - Leave pages open after accesses, or
    - Close page when there are no further accesses available in the controller for that page, or
    - Auto-precharge with each access, with an optimization for page-close mode which leaves the page open after a flush for read-write and write-read collision cases
- Supports automatic SDRAM power-down entry and exit caused by lack of transaction arrival for programmable time
- Supports automatic Clock Stop entry and exit caused by lack of transaction arrival
- Supports automatic DDRC low power mode operation caused by lack of transaction arrival for programmable time via Hardware Low Power Interface
- Supports self-refresh entry and exit as follows:
    - Supports automatic self-refresh entry and exit caused by lack of transaction arrival for programmable time
    - Support for self-refresh entry and exit under software control
    - Support for self-refresh entry and exit using dedicated DDRC hardware low power interface control (similar to the AMBA 3 AXI protocol low power control interface)
- Support for dynamically changing clock frequency while in self-refresh:
    - DDR4 DLL-off mode supported
    - Shadow timing registers provided to allow fast frequency changing
- Support for explicit SDRAM mode register updates under software control
- Flexible address mapper logic to allow application specific mapping of row, column, bank, and rank bits
- Programmable support for 1T or 2T timing
- User-selectable refresh control options:
    - Controller-generated auto-refreshes at programmable average intervals

    - In multi-rank designs, an offset can be applied to each rank's refresh timer to allow rank refreshes to expire at different times (this can increase efficiency by allowing traffic to continue to other ranks while a given rank is being refreshed)
    - Ability to group up to 8 controller-generated refreshes together to be issued consecutively (this reduces the frequency of page closings, increasing overall efficiency)
    - When controller-generated refreshes are grouped, some refreshes can be issued speculatively when the controller is idle for a programmable period of time
    - Ability to disable controller-generated auto-refreshes
    - Ability to issue a refresh through direct software request
    - When LP-DDR4 is used, user-selectable ability to perform per-bank refreshes rather than all-banks refreshes
    - When DDR4 is used, fine granularity refresh can be selected
- DDR4 protocol up to DDR4-3200 speed grade (1:2 frequency ratio mode) and DDR4-2133 speed grade (1:1 frequency ratio mode) are supported by the DDRC. The following features are supported:
    - Data Bus Inversion (DBI)
    - Geardown mode may be programmed where allowed by memory and PHY specification (not supported for 1:1 frequency ratio mode)
    - Maximum power saving mode
    - Multi-purpose register (MPR) reads and writes
    - Per DRAM addressability
    - Fine granularity refresh
    - Cyclic redundancy check (CRC) with retry (not supported for 1:1 frequency ratio mode)
    - Command/address latency
    - Programmable Preamble (not supported for 1:1 frequency ratio mode)
    - 3DS with up to 8 logical ranks/DDR4-2666
- Supports Dual Data Channel, also called "Shared AC", with a common address and command bus

9.3.2 Functional description

The DDRC converts system bus transactions into memory commands on the DFI interface that are compliant with the DDR protocols. The DDRC supports both memory components and DIMMs(Dual in-line Memory Module).

Please see the block diagram at Block diagram.

The main components of the block are:

- **AXI Port Interface (XPI) block**: This block provides the interface to the application ports. It provides bus protocol handling, data buffering and reordering for read data, data bus size conversion (upsizing or downsizing), and memory burst address alignment. Read data is stored in a SRAM, read re-order buffer and returned in order, to the AXI ports. The SRAM may be instantiated as embedded memory external to the DDRC or be implemented as flops within the DDRC.
- **Port Arbiter (PA) block**: This block provides latency sensitive, priority based arbitration between the addresses issued by the XPIs (by the ports).
- **DDR Controller (DDRC) block**: This block contains a logical CAM (Content Addressable Memory), which can be synthesized using standard cells. This holds information on the commands, which is used by the scheduling algorithms to optimally schedule commands to be sent to the PHY, based on priority, bank/rank status and DDR timing constraints. A bypass path is also provided (optionally).
- **APB Register Block**: This block contains the software accessible registers.

The details on DDRC are as follows:

- Write data is stored in DDRC until its associated command is issued to the PHY.
- Read data is handled by the response engine in the DDRC and is returned in the order of scheduled read commands on the HIF.
- ECC handling is an optional function which is handled by logic modules within the DDRC in the write data path and in the response engine.

1:1 and 1:2 frequency ratios are terms used for the frequency ratio between the DDRC clock (core_ddrc_core_clk) and the memory clock.

When MEMC_PROG_FREQ_RATIO = 1 the parameter MEMC_FREQ_RATIO is tied to 2 and the HIF data bus is four times the memory data width.

The register MSTR.frequency_ratio controls whether the controller operates in 1:1 frequency ratio mode or 1:2 frequency ratio mode. When MSTR.frequency_ratio is programmed to 1 the DDRC clock rate is the same as the memory clock rate (referred to as SDR, single data rate) i.e. 1:1 frequency ratio mode. When MSTR.frequency_ratio is programmed to 0 the DDRC clock rate is one half the memory clock rate (referred to as HDR, half data rate) i.e. 1:2 frequency ratio mode.

When MEMC_PROG_FREQ_RATIO = 0 the parameter MEMC_FREQ_RATIO controls whether the controller operates in 1:1 frequency ratio mode or 1:2 frequency ratio mode. When MEMC_FREQ_RATIO=1 the DDRC clock rate is the same as the memory clock rate (referred to as SDR, single data rate) i.e. 1:1 frequency ratio mode. The HIF data bus is twice the memory data width. When MEMC_FREQ_RATIO=2 the DDRC clock rate is one half the memory clock rate (referred to as HDR, half data rate), that is, 1:2 frequency ratio mode. The HIF data bus is four times the memory data width

In the case of 1:1 frequency ratio mode, the DDRC and PHY operate at the same clock frequency and are referred to as a ‘matched frequency system’ in the DFI Specification.

For example if MEMC_PROG_FREQ_RATIO=1, in 1:1 frequency ratio mode, if the memory data width is 32 bits and the memory clock rate is 400 MHz or 800 Mbps, the HIF data bus would be 128 bits wide and only the lower half of the dfi bus is utilized. The DDRC clock rate would be 400 MHz. If the frequency ratio mode is changed to 1:2, the HIF data bus would be 128 bits wide and all of the dfi data bus is utilized. The DDRC clock rate would be 200 MHz.

For example if MEMC_PROG_FREQ_RATIO=0, in 1:1 frequency mode, if the memory data width is 32 bits and the memory clock rate is 400 MHz or 800 Mbps, the HIF data bus would be 64 bits wide and all of the dfi data bus is utilized. The DDRC clock rate would be 400 MHz. If the frequency mode is changed to 1:2, the HIF data bus would be 128 bits wide and all of the dfi data bus is utilized. The DDRC clock rate would be 200 MHz.

The terms “DFI clock” and “DFI PHY clock” are used in this document in accordance with the DFI specification:

- The DFI clock is the clock on which the DFI interface runs. It is the core_ddrc_core_clk.
- The DFI PHY clock is the SDRAM clock. In 1:1 frequency ratio, this has the same frequency as the DFI clock. In 1:2 frequency ratio, this has twice the frequency of the DFI clock.

9.3.2.1 AXI Port Interface (XPI)

The AXI protocol is burst-based with read and write request channels that specify the host ID for the request, start byte address, burst length, burst size, and burst type. This information is processed by the interface and is used subsequently by the DDRC.

The XPI interfaces the AXI application port to the DDRC and performs the following main functions:

- Read address generation
- Write address generation
- Read data generation
- Read data and response generation
- Write response generation

The XPI converts AXI bursts into read and write requests, which are forwarded to the Port Arbiter (PA). In the opposite direction, the XPI converts the responses from the DDRC into appropriate AXI responses.

The interface between XPI and PA is same as HIF (but with independent channels for read and write commands). Each AXI port can be independently configured to operate with a clock that is synchronous or asynchronous to the memory controller clock.

The AXI Specification requires that transactions must not cross a 4K address boundary. AXI compliant masters must meet this requirement. However, if a master does not comply with this AXI protocol requirement, the user can relax this restriction in the DDRC and set the boundary between 4K and 4G using the hardware parameter DDRC_AXI_ADDR_BOUNDARY.

9.3.2.1.1 Read Address Channel

The AXI read address channel has the following features:

- The read transaction can be of any length up to 256 for incremental bursts, 16 for wrapping bursts.
- The burst types supported are incremental and wrapping.
- The burst start address can be unaligned to the AXI data width boundaries
- The size of the burst can be less than the full width of the AXI data bus (also known as sub-sized transfers).

The signals on the read address channel are registered into the XPI in accordance with the AXI valid/ready handshaking protocol and are synchronous with the AXI clock (aclk).

Read requests are accepted if the request can be written into the Read Address Queue (RAQ). This is used to store all the read address requests. If DDRC_XPI_USE2RAQ_n is set to 0, there is single address queue on a given port. If DDRC_XPI_USE2RAQ_n is set to 1, there are two address queues on a given port, named Blue and Red address queues. Clock domain crossing from aclk to DDRC clock (core_ddrc_core_clk) is performed in the RAQ block. The depth of the RAQ can be configured to suit the system requirements of the user.

After registering the read address channel by the RAQ, the following operations are performed:

- Generation of new read requests is based on alignment (derived from AXI address and size), burst lengths (derived from AXI length and memory burst length), and

burst type (derived from AXI incremental or wrapping). Each AXI burst is divided into packets of length equal to the memory burst length (BL2, BL4, BL8, BL16).
- Generation of new HIF address in the case of unaligned burst and burst expansion. A burst is aligned to the memory burst boundary when:
  - A(R|W)ADDR[X]=0 if BL2
  - A(R|W)ADDR[1+X:X]=0 if BL4
  - A(R|W)ADDR[2+X:X]=0 if BL8 or
  - A(R|W)ADDR[3+X:X]=0 if BL16, Where, X is log2 of the number of data bytes in the SDRAM interface. Therefore,
  - If MEMC_DRAM_DATA_WIDTH = 8, X = 0
  - If MEMC_DRAM_DATA_WIDTH = 16, X = 1
  - If MEMC_DRAM_DATA_WIDTH = 32, X = 2 and so on

In case of an unaligned burst, the first read request is unaligned and the remaining read requests are aligned.

In general, realignment to a memory burst boundary potentially causes some data beats to be discarded (affecting bandwidth) and potentially introduces additional latency on the read data and response channel.

The arready output is defaulted to logic one and remains in logic one unless the RAQ becomes full or a WRAP burst is requested (arready is low on the next cycle after a WRAP burst is accepted). The read transaction is popped from the RAQ when it is not empty.

The XPI handles the generation of the token which is used by the DDRC for identifying the read command and corresponding data.

9.3.2.1.2 Write Address Channel

Write Address Queue (WAQ) is used to store all the addresses for write requests from a given port. There is a single queue for all AXI IDs from a given port. The write address channel behavior is similar to the read address channel.

The depth of the WAQ can be configured to suit the system requirements of the user. Write address and read address channels are independent and the ordering between the write and read requests may not be preserved.

To preserve the sequence, a higher-level protocol needs to wait for read/write response before sending the next transaction.

Transactions across the ports are independent and can be issued in any order.

The write command is not forwarded to the DDRC until the write data is collected and the strobes are evaluated.

9.3.2.1.3 Wrap Burst Expansion

A WRAP burst may be expanded by the XPI into multiple SDRAM transactions. This is also true for cases where WRAP bursts, which could be accommodated by a single transaction on the SDRAM interface. A WRAP burst is expanded into a single SDRAM transaction only in particular cases as highlighted in the tables below.

NOTE

SDRAM burst length 16 is not supported by XPI in this case. If the burst length is programmed to 16, XPI uses the value 8.

Table 9-1. MEMC_BURST_LENGTH 16 Wrap Expansion

Direction	Port Data Width	(SDRAM BurstLength8&&Quarter Burst) || (SDRAM Burst Length4&&Half Burst)|| (SDRAM BurstLength4&&Quarter Burst)	(SDRAM Burst Length16&&Quarter Burst)|| (SDRAM Burst Length8&&Half Burst)|| (SDRAM Burst Length4&& Full Burst)	(SDRAM Burst Length16&&Half Burst)|| (SDRAM Burst Length8&&Full Burst)	SDRAM Burst Length16&&Full Burst
Read	Native-Sized	N/A	Single	Single	Single
	Up-Sized	Single(onlyif araddris alignedtotheHIF burst)	Single	Single	Single
	Down-Sized	N/A	N/A	Single	Single
Write	Native-Sized	N/A	Multiple	Multiple	Single
	Up-Sized	Multiple	Multiple	Multiple	Single(onlyif awaddrisalignedtotheHIF burst)
	Down-Sized	N/A	N/A	Multiple	Single

In the tables, full burst is when:

DDR_bytes = AXI_bytes, half burst is when DDR_bytes = 2*AXI_bytes and quarter burst is when DDR_bytes = 4*AXI_bytes.

Where, DDR_bytes refers to the number of bytes transferred in a DDR burst: MSTR.burst_rdwr*2*MEMC_DRAM_DATA_WIDTH/8

AXI_bytes refers to the number of bytes transferred in an AXI burst: 2(ALEN +1)*(DDRC_PORT_DW_n/8)

As an example, in a natural-size port, with MEMC_FREQ_RATIO=1 and MEMC_BURST_LENGTH=16, when the SDRAM burst length is programmed to 8 (Burst length of 16), ALEN 7 results in a full burst.

9.3.2.1.4 Read Data and Response Channel

The XPI handles the common response interface to the DDRC to process the read data from the memory.

AXI read data and response channel has a single data storage queue (RDQ - Read Data Queue) with two clocks—AXI clock for reading and DDRC clock for writing.

Data from different IDs are stored in the same queue and are returned in the order of read address acceptance. You can configure the depth of the queue independently for each port.

The FSM handling the AXI handshake as well as the FIFO push and pop has the following features:

- Based on the information stored, filter the dummy beats
- Based on the signal hif_rdata_last, generate the rlast

The controller provides an OKAY response for the each read, except for exclusive read transactions. The DDRC provides an EXOKAY response.

SLVERR respinse may be returned for the read transactions (both normal and exclusive) for the following cases:

- ECC uncorrected error detected at the DFI
- On-chip parity address or data error
- InvalidLPDDR4 row address
- Transaction has been poisoned

NOTE

SLVERR has the highest priority.

9.3.2.1.4.1 Read Reorder Buffer

The read data can be returned from the DDRC in a different order from which the read commands are forwarded from the XPI. (due to the re-ordering of read commands in the DDRC to maximize SDRAM bandwidth).A read reorder buffer is implemented in each port to reorder the read data for that port to the same order as the order of the AXI read commands.

The read reorder buffer SRAM holds the same number of entries as the read CAM and each entry holds the read data corresponding to a DDR command. Storage for the reorder buffer can be implemented internally in the DDRC or implemented externally to the DDRC as embedded SRAM and accessed through an External RAM Interface. In either case, the control circuits for the read reorder SRAM are clocked by the DDRC clock (and not AXI clock). If implemented as embedded SRAM, a two-port SRAM is required.

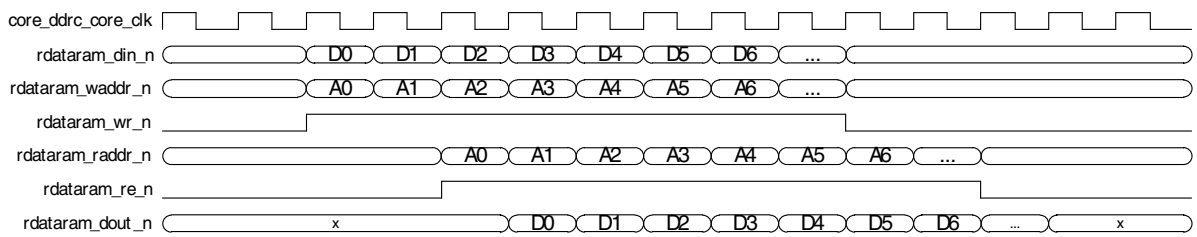


Figure 9-2. read Reorder Buffer Data RAM Interface Timing

The AXI protocol allows the read data for transactions of different IDs to be interleaved. To reduce potential delays where read data for one ID is blocked waiting for data associated with another ID, the read reorder buffer is organized as number of virtual channels (See the figure below). The Read Reorder Virtual Channel is a mechanism to allow independent read data reordering between multiple groups of AXI IDs.

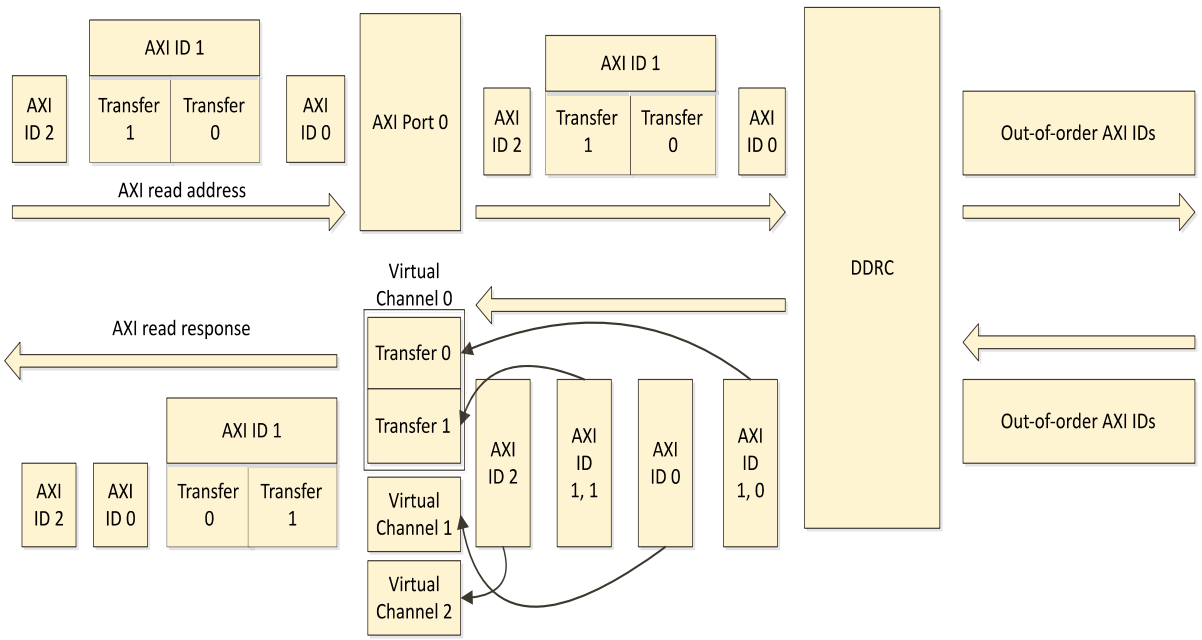


Figure 9-3. RRB Configured with Three Virtual Channels

9.3.2.1.5 Write data Channel

The AXI write data channel has a data storage queue (WDQ - Write Data Queue) with the following clocks:

- AXI clock (aclk_n) for writing and
- DDRC clock (core_ddrc_core_clk) for reading

This queue is used for the clock domain crossing between the AXI clock domain and the controller clock domain and acts as a registering layer in the case of a synchronous interface.

At the output of WDQ, some beats of a write data packet can be masked depending on alignment, burst size and burst length.

Write data from each port is forwarded to the DDRC, which forwards the data to a common data storage.

9.3.2.1.6 Data Width Conversion

The XPI performs the data width conversion between the data width at the AXI interface and internal HIF data width. The AXI data width can be set by the hardware parameter DDRC_PORT_DW.

Both data width down-conversion (where the AXI data bus is wider than the internal HIF data-bus width) and data width up-conversion (where the AXI data bus is narrower than the internal HIF data-bus width) are supported.

If no data width conversion is performed, port is referred as Native-sized:

DDRC_PORT_DW_n = 2 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 1

DDRC_PORT_DW_n = 4 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 2

When data width up conversion is performed, port is referred as Up-sized:

DDRC_PORT_DW_n < 2 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 1

DDRC_PORT_DW_n < 4 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 2

When data width down conversion is performed, port is referred as Down-sized:

DDRC_PORT_DW_n > 2 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 1

DDRC_PORT_DW_n > 4 * MEMC_DRAM_DATA_WIDTH if MEMC_FREQ_RATIO = 2

The only exception is when Shared-AC with data channel interleaving is used, where Native-sized port is equivalent to a Down-sized port (1:2 ratio) in Single Channel configuration.

The width of the queues, Write Data Queue (WDQ) and Read Data Queue (RDQ) are always set to the wider data width. In the case of data width down-conversion, these queues are set to the width of the AXI data bus. In the case of data width up-conversion, these queues are set to the width of the internal HIF data-bus width

9.3.2.1.7 Write Response Channel

The write response is generated once the last beat of write data, for a given AXI burst, is accepted by the DDRC. The write response queue can be instantiated with configurable depth which is set by the hardware parameter DDRC_AXI_WRQD_n.

The write response generation makes use of the result of the exclusive access monitor. For a write transaction, the response is always returned as OKAY. For an exclusive write transaction, the response can be returned as OKAY or EXOKAY.

SLVERR response may be returned in the following cases:

- Invalid LPDDR4 row address
- On-chip parity address or data error
- Transaction has been poisoned

NOTE

SLVERR has the highest priority

9.3.2.1.8 Exclusive Access

The DDRC supports the AXI Exclusive Access feature. This feature is enabled using the hardware parameter DDRC_EXCL_ACCESS. This parameter defines the number of addresses the DDRC can monitor.

All exclusive read transactions have an EXOKAY response (except in the case of an ECC uncorrectable error). Successful exclusive write accesses have an EXOKAY response, unsuccessful exclusive write accesses return an OKAY response. If an exclusive write fails, the data mask for the exclusive write is forced low so the data is not written.

One address range per AXI ID is monitored for exclusivity. Therefore, if a master does not complete the write portion of an exclusive operation, a subsequent exclusive read to the same ID changes the address that is being monitored for exclusivity.

Once an exclusive access monitor for a given address is enabled, all write transactions are monitored for violation, regardless of the originating port. In other words, the violation check operates across the ports.

The exclusive access monitor compares the exclusive write transaction address, size, length, ID and port number against the exclusive read transaction address, size, length, ID and port number, and only accepts an exclusive write when these parameters match. Otherwise, the exclusive write is considered as fail.

If the EXCL_ACCESS parameter is configured to support no (0) exclusive accesses, the DDRC behaves like an AXI slave that does not support exclusive accesses. Therefore, all exclusive accesses return OKAY responses.

In some cases SLVERR response may be returned for exclusive access transactions. For more information about this, see **“Read Data and Response Channel”** and **“Write Response Channel”**.

NOTE

- The read response SLVERR is generated for an exclusive read transaction for ECC configurations, when ECC is enabled and an uncorrectable error is detected. The master, upon receiving a SLVERR to an exclusive read command should not complete the exclusive operation by sending an exclusive write command. If it does, the DDRC monitors return EXOKAY if there is no violation.
- The maximum number of bytes that are monitored as a region per address cannot exceed 128 bytes. The AXI exclusive transaction length must always be a power of two.
- When all the monitors are active, further exclusive read should not be issued. If this happens, one of the active monitors is evicted using a round robin control and new exclusive read is accepted. The selection of the monitor to be evicted is based on a pointer which rotates across all the locations when a new exclusive read is received.

9.3.2.1.8.1 Dual Channel considerations

In dual channel configurations, Shared AC and LPDDR4 dual channel, a second set of monitors is instantiated for the second channel. This means that the total number of exclusive monitors is doubled.

This has the following implications:

- Maximum number of monitored locations is equal to DDRC_EXCL_ACCESS*2 (DDRC_EXCL_ACCESS locations per data channel).
- If a master issues two Exclusive Read accesses to the same ARID to different channels (for example, channel 0 first, followed by channel 1, without issuing Exclusive Write access in between), channel 0 monitor for this ID is not evicted immediately and channel 1 monitor starts monitoring for the same ID.

According to AXI protocol, the maximum transfer size for exclusive accesses is 128 bytes. For correct functionality, an exclusive access transfer must not be interleaved across two data channels, that is, one EXA transfer goes to one channel only.

The following are the possible ways to achieve this requirement:

- The channel select bit of the address map (ADDRMAP0.addrmap_dch_bit0) can be programmed at or above the 128 byte boundary. For

MEMC_DRAM_DATA_WIDTH=16, this boundary corresponds to HIF[6] or addrmap_dch_bit0=4. For MEMC_DRAM_DATA_WIDTH=32, this boundary corresponds to HIF[5] or addrmap_dch_bit0=3.

- If the required channel interleaving boundary is less than 128 bytes, then the size of all EXA transfers must be constrained to be less or equal to the interleaving boundary.

9.3.2.1.9 Software Coherency for AXI Ports

The section “Address Collision Handling” describes how the DDRC handles in-order execution of commands to the same address.

For the commands issued at the AXI port, the logic in the DDRC protects against all types of software coherency hazards when the AXI master waits for write (read) response before sending the next same address read (same address write) or vice versa.

Some special AXI masters expect an ordering relationship between the reads and writes when the opposite direction transactions are sent without waiting for the previous response. For these special cases, additional logic should be instantiated in the XPI to enforce the order of acceptance within the reads and writes between the AXI and HIF interfaces at the DDRC. This logic is included by setting of the hardware parameter DDRC_RDWR_ORDERED_n and enabled by the register PCFGR_n.rdwr_ordered_en. When this feature is enabled and read and write transactions are presented at the same cycle, the pre-arbiter gives precedence to read transactions, incrementing the write transaction order token by 1 with respect to the read transaction order token. For example, if the last order token issued was equal to 0, then the read order token will be equal to 1 and the write order token will be equal to 2.

9.3.2.1.10 Transaction Poisoning

Sideband signals arpoison/awpoison, render an AXI transaction (read or write) invalid and the DDRC signals that AXI transaction poisoning has occurred. The input signal must be asserted coinciding with the associated AXI transaction. If a write is poisoned, all of its strobes are de-asserted, making the write effectively transparent to the memory. If a read is poisoned, the command is issued to the DDR memory and all the read data beats are overwritten and returned as zeros.

The DDRC may be programmed to signal that an AXI transaction poisoning has occurred by setting an interrupt or by driving the AXI SLVERR response for that transaction response. The AXI Poison Configuration Register POISONCFG (see DDRC_REGS Registers) is used to define whether an interrupt and/or AXI SLVERR is generated when an AXI transaction poisoning has occurred. There is separate control for read and write

transactions. The port specific AXI Poison status register is asserted whenever an AXI transaction is poisoned on that port. There is a separate status for read and write transactions. The interrupt and status register is cleared by the register POISONCFG.rd_poison_intr_clr/POISONCFG.wr_poison_intr_clr. Interrupt status is stored in the AXI Poison Status register POISONSTAT (see DDRC_REGS Registers onpage 464). This register is a mirror in the APB clock domain of the actual interrupts, so status is updated with 2 or 3 pclk cycles of delay depending on the CDC logic.

An external block is expected to drive these new sideband signals depending on the access security requirements on transaction by transaction basis. Therefore, this feature replaces the need to implement ARM Trustzone.

9.3.2.1.11 AXI Data Channel Interleaving

A given AXI port (XPI) can access both data channels and low granularity interleaving between the two data channels is possible by mapping a suitable LSB address bit which is at the memory burst boundary. In that case, for example, a long AXI burst can be expanded to HIF commands so that back to back accesses are to alternating data channels. This setting achieves the highest efficiency possible from a single AXI port.

On the read data path, data order - based on AXI ordering rules - has to be preserved within each channel and between both channels. The read reorder buffer reorders the data coming from a given channel. The data ordered from both channels is then reordered by the data channel reorder logic. Thus two separate data SRAMs are instantiated, one for each channel, whose outputs are combined to provide a single wide AXI read data channel.

By setting the correct system mapping, one port can be set to access one data channel and another port can be set to access the other data channel. This configuration saves logic area in XPI block.

9.3.2.2 AHB Port Interface

The AHB slave port interface consists of an A2X block and XPI block as described in “AXI Port Interface (XPI)”. The A2X block converts the AHB slave port transactions into AXI equivalent transactions before forwarding them to the XPI, where the transactions are finally broken down into HIF packets and sent to the Port Arbiter (PA).

9.3.2.2.1 A2X

The A2X controls the following features of the AHB Slave port interface.

9.3.2.2.1.1 Split Mode

The split mode of the AHB slave port can be set by the parameter DDRC_AHB_SPLIT_MODE_n. Split response is used to control the AHB bus for efficiency and coherency on a multi-master split capable port. This is especially for AHB read transactions where the master is split once the NSEQ address is captured. When the DDRC fetches the read data for split transaction, the AHB interface slave port is free to receive a transaction for another master. The maximum number of splits (simultaneous) is equal to the number of AHB masters configured for the port. Once data is retrieved for a particular master, the master that is split is recalled.

For AHB write, the transaction may be split depending on the write response mode. For more information about this, see **"Write Response Mode"**.

In non-split mode, a single master controls the AHB interface slave port until its transaction completes, unless it is early burst terminated before completion. In the case of a read, the port output hready_resp is deasserted after the NSEQ phase of the transaction and asserted again during the data phase of each beat once the read data is available. When operated in this mode, every read transaction holds the bus for the amount of data beats requested plus at least the read idle latency time of the DDRC. For this reason, nonsplit AHB configuration has a negative impact on the performance of the port when compared with the split configurations.

9.3.2.2.1.2 Write Response mode

There are three AHB write response modes namely non-bufferable, bufferable, or dynamic write response modes. These modes can be programmed (per port) by the parameter DDRC_AHB_WRITE_RESP_MODE_n.

In non-bufferable mode, the hready_resp for the data phase of the last write beat for a defined length AHB burst is not asserted until the A2X logic receives confirmation from XPI (through bresp_n) that the last data beat is accepted by the DDRC. When the port operates in non-split mode, hready_resp remains low until this event occurs. When the port operates in split mode, the master whose write transaction is on the bus is split on either the last write data beat or if hready_resp has been driven low for DDRC_AHB_HREADY_LOW_PERIOD_n cycles (whichever occurs latter), and then recalled when that beat is accepted by the DDRC. For an undefined length INCRs, hready_resp for the data phase of each beat is not asserted until the A2X logic receives

confirmation that each beat is accepted by the DDRC. This indicates performance degradation for long undefined INCR writes when compared with the bufferable equivalent.

In bufferable mode, the logic does not wait for this last beat of write data to be accepted by the DDRC. Instead, hready_resp on the AHB port interface is asserted for the data phase of this beat once the AHB port of the DDRC can accept it. If operating in split mode, a split is generated if the hready_resp is driven low for greater than DDRC_AHB_HREADY_LOW_PERIOD_n cycles.

In dynamic mode, the transactions can be either bufferable or non-bufferable depending on the value of hprot according to the AMBA AHB Specification for the individual AHB transactions, where hprot[2]=1'b0 is non-bufferable and hprot[2]=1'b1 is bufferable.

9.3.2.2.1.3 Error Response

The AHB port only returns an error response on hresp_n for a data beat that has an uncorrectable ECC error. For this to occur, the DDRC should be configured to enable ECC logic. For more information about this, see "Read-Modify-Write (RMW) Generation".

9.3.2.2.1.4 Early Burst Termination (EBT)

A defined length write transaction that is early burst terminated by another transaction has its remaining write data beats masked during memory access. When the early burst terminated master returns, only the remaining write data beats that are masked after the initial EBT are sent to the memory. This EBT of defined length write transactions causes multiple memory accesses. The non-bufferable response for the single defined length transaction that is split into multiple memory access is given only after all memory accesses are completed.

Any defined length read transaction must return to retrieve all its data if early burst terminated. Any undefined length INCR read transaction has all remaining data beats that are not retrieved by a master, up to the read prefetch beats set by the pin hincr_arlen_n, flushed internally in the DDRC.

9.3.2.2.1.5 AHB Lite mode

The parameter DDRC_AHB_LITE_MODE_n is used to set an AHB port to operate in AHB Lite mode. This means that the port can be connected only to a single master and responds to AHB read and nonbufferable write transactions by driving hready_resp low (as opposed to split responses). This simplifies the AHB master/arbiter system connected to an AHB Lite port.

9.3.2.2.1.6 Endian conversion

The A2X provides endian mapping from the AHB port to the XPI with respect to the transaction size. This section outlines the key features of the A2X endian mapping.

The key features of endian conversion are as follows:

- Register: The functionality is enabled by PCFGC_n.ahb_endianness (see DDRC_MP Registers)
- Endian transformations supported:
  - Little Endian (LE)
  - Big Endian-32 (BE-32) invariant
  - Big Endian-A (BE-A) address invariant

9.3.2.2.2 Write Streaming

For AHB write streaming, the write data buffer parameter DDRC_AHB_WDQD should be set to a size that can support enough write data beats from the first valid AHB write beat. So, the transaction on the AHB slave interface port to time the first beat is accepted by the DDRC, signified by asserting the DDRCs hif_wdata_valid and hif_wdata_stall outputs.

9.3.2.2.3 Read Streaming

For non-split configurations, it is not possible to stream different transaction due to the nature of the AHB protocol, as we cannot get the address of the next AHB transaction to access memory until the read data for the current transaction has completed. We can stream within a transaction from the time the first read beat is available until the last read beat of the same transaction.

For split capable configurations, depending on the number of masters and arbitration scheme used accessing a port and the type of AHB transaction used, it is possible to stream reads after the initial read latency delay from the first AHB transaction.

9.3.2.2.4 XPI

The XPI used in conjunction with the A2X provides features to complete the overall AHB slave port solution. These features include data width conversion from the AHB port data width to the internal data width used by the PA and DDRC, which is twice the width of the SDRAM data interface in case of 1:1 clock frequency ratio and four times the width of the SDRAM data interface in case of 1:2 clock frequency ratio.

Each AHB port can be independently configured to operate with a clock that is synchronous or asynchronous to the memory controller clock.This clock crossing logic is done within the XPI.

ECC is supported for configurations with AHB port interfaces.

9.3.2.2.5 Software Coherency For AHB Ports

Coherency of read after write (RAW) and write after read (WAR) to the same address is always guaranteed. For more information about this, see “Software Coherency for AXI Ports” and “Address Collision Handling”.

9.3.2.3 Host Interface (HIF)

HIF is an internal interface when the DDRC is configured to include a multi port arbiter (the hardware parameter DDRC_INCL_ARB is set to 1). In this case, the AXI Port Interface and/or Port Arbiter handle all signals on this interface, effectively acting as the SoC core.

Read, write, as well as write data are received through the HIF. Following the receipt and acceptance of a write the DDRC requests write data from the interface. The interface subsequently provides the independently timed write data in the requested order (which matches the order in which requests are made to the DDRC). See “Dual HIF Functionality” for further details on Dual HIF functionality. Both read and write requests can be reordered later, before being output on the DFI interface.

Requests to the DDRC are throttled in two ways:

- A credit mechanism that ensures that buffer space is available for any request the SoC core makes to the DDRC, prior to the request being made.
- An independent stall mechanism that throttles requests when address collisions take place or if entering self-refresh through the software self-refresh or hardware low power self-refresh.

Data is handled separately. The DDRC throttles data by potentially waiting after it receives a write before requesting the associated data. If there is collision, then the throttling lasts until the collision is cleared in the DDRC. Once write data is requested, it is always accepted by the DDRC.

Read data is sent to the host via a common response interface. For more information about this, see “Common Response”.

This section describes the following about the HIF:

- “Dual HIF Functionality”
- “Credit Mechanism”
- “Read Requests”
- “Write Requests and Data”
- “Read-Modify-Write Requests”
- “Common Response”

9.3.2.3.1 Dual HIF functionality

Dual HIF functionality is as follows:

- Converts HIF single command channel into separate HIF command channel for Read and Write commands.
- Read/Write Arbitration is not performed by the PA
- Dual HIF functionality is supported only if the DDRC is configured to support a multi port arbiter (DDRC_INCL_ARB=1).
- Dual HIF functionality is supported only if the DDRC is configured to include the logic to optimize timing over scheduling efficiency (MEMC_OPT_TIMING=1).
- Dual HIF functionality continues to guarantee coherency for write-after-read (WAR) and read-afterwrite (RAW) hazards. Also, if a Read and Write/ to the same address are taken at the same time (both hif_rcmd_stall=0 and hif_wcmd_stall=0), the Write/ is performed first.

The table provides the details of the signals modified by the DDRC_DUAL_HIF parameter.

Table 9-2. Comparison of Signals Modified by DDRC_DUAL_HIF Parameter

DDRC_DUAL_HIF=0 Single HIFCommand Channel	DDRC_DUAL_HIF=1 [1] Separate HIF Command Channels One for Reads (*_rd) One for Writes/ (*_wr)
hif_cmd_valid	hif_rcmd_valid hif_wcmd_valid
hif_cmd_type	hif_rcmd_type hif_wcmd_type

Table continues on the next page...

Table 9-2. Comparison of Signals Modified by DDRC_DUAL_HIF Parameter (continued)

hif_cmd_addr	hif_rcmd_addr hif_wcmd_addr
hif_cmd_pri	hif_rcmd_prihif_wcmd_pri
hif_cmd_token	hif_rcmd_token hif_wcmd_token
hif_cmd_token	hif_rcmd_length hif_wcmd_length
hif_cmd_wdata_ptr	hif_rcmd_wdata_ptr hif_wcmd_wdata_ptr
hif_cmd_autopre	hif_rcmd_autopre hif_wcmd_autopre
hif_cmd_stall	hif_rcmd_stall hif_wcmd_stall

NOTE

1. As DDRC_DUAL_HIF=1 is only possible if DDRC_INCL_ARB=1, the right hand column refers to internal signals within the DDRC. It is provided for debug purposes only.

NOTE

The databook uses DDRC_DUAL_HIF=0 notation.

If you are using DDRC_DUAL_HIF=1, the internal signals should be interpreted in the following way:

- hif_cmd_valid refers to hif_rcmd_valid and/or hif_wcmd_valid.
- hif_cmd_stall refers to hif_rcmd_stall and/or hif_wcmd_stall and so on.

9.3.2.3.2 Credit Mechanism

The DDRC employs a credit mechanism to ensure that buffers do not overflow. The interface making the request to the DDRC, can only request commands for which it has been granted “credits” to issue.

Credits are tracked seperately for the following three command types:

- High Priority read(HPR)
- Low Prioroity rad(LPR)
- Write

Credits are counted for each command type independently according to the following rules:

- Initially the interface has zero credits.

- The interface logic must include counters to track the number of credits granted by the DDRC and decremented due to commands issued by the interface logic to the DDRC. The interface logic must have separate credit counters for each individual command type.
- Following the de-assertion of reset to the DDRC, credits are issued to the interface for each command type. A given credit count increments every time the DDRC issues a credit, indicated by the assertion of the appropriate *_credit signal on the rising edge of core_ddrc_core_clk. The credit counting logic implemented externally to the DDRC must run on the same clock.
- When the credit count is greater than zero, the interface can issue requests of that type to the DDRC. Each time a request is issued to the DDRC, the associated credit count is decremented.
- When DDRC_VPRW_EN is set to 1, the arbiter or the host can send Variable Priority Read (VPR) and Variable Priority Write (VPW) commands.
- VPR commands do not have a pre-allocated storage resource in the Read CAM (unlike HPR commands). VPR commands share the same resource with the LPR commands. As far as the credit mechanism is concerned, the VPR commands are counted in the LPR credit bucket.
- Similarly, VPW commands do not have a per-allocated storage resource in the Write CAM. VPW commands share the same resource with the Normal Write commands. As far as credit mechanism is concerned, the VPW commands are counted in the WR credit bucket.

The signals related to the credit request interface are:

- hif_cmd_valid
- hif_cmd_type
- hif_cmd_pri
- hif_lpr_credit
- hif_hpr_credit
- hif_wr_credit
- hif_cmd_stall

9.3.2.3.3 Clear Requests

When the LPR or HPR credit count is greater than zero, the interface can issue read requests to the DDRC accordingly.

A command is issued to the DDRC by asserting hif_cmd_valid on the rising edge of the clock. All read request fields must be driven to the appropriate values at the same time. These fields are:

- **hif_cmd_type**: specifies the request type as follows:

- ‘00’ indicates a write request
- ‘01’ indicates a read request
- ‘11’ is not supported

- **hif_cmd_pri**:
  - In HIF-only configurations (DDRC_INCL_ARB=0) where DDRC_VPRW_EN is set to 0, this signal is 1-bit wide. A ‘1’ indicates the read request is high priority. ‘0’ indicates the read request is low priority. This field is meaningless for write commands.
  - In Arbiter configurations (DDRC_INCL_ARB=1) or in HIF-only configurations (DDRC_INCL_ARB=0) where DDRC_VPRW_EN is set to 1, this signal is 2-bit wide.

  The encoding for Read is:

  - 2’b00 - LPR
  - 2’b01 - VPR
  - 2’b10 - HPR
  - 2’b11 - Reserved

  VPR commands are sent only in configurations with DDRC_VPRW_EN=1

- **hif_cmd_addr**: indicates the address for the read. Each word on the HIF (equivalent to 2 DDR words for 1:1 mode configurations (MEMC_FREQ_RATIO=1), or 4 DDR words for 1:2 mode configurations (MEMC_FREQ_RATIO=2)) is uniquely addressable.
- **hif_cmd_length**, is 2 bits if MEMC_BURST_LENGTH=16, 1 bit otherwise.

  Note the following:

  - For MEMC_BURST_LENGTH = 16, a normal read is 16 SDRAM words

  If MEMC_BURST_LENGTH=16:

  - 2’b11 -> Partial (Quarter) Read
  - 2’b10 -> Partial (Half) Read
  - 2'b00 -> Full Read

  Otherwise:

  - 1’b1 -> Partial (Half) Read
  - 1’b0 -> Full Read

In each case, a partial (half) read is half the length of a normal full read (where a word is the amount of data transferred on one edge of the DQS to the SDRAM in a fully configured system.) If MEMC_BURST_LENGTH=16, Partial (Quarter) Read is possible and is quarter the length of a normal full read. This field is meaningless for write or commands.

- **hif_cmd_token**: a bit field that is presented with the read command and is returned with the read response. As the responses may be presented out-of-order, the token is the identifier that indicates the read for which data is being returned on the response side. Therefore, multiple reads with the same token must never be pending in the DDRC simultaneously, as the responses would be indecipherable. Common uses for this field include identifying the requestor and serializing the response data. This field is meaningless for write or commands.

9.3.2.3.3.1 Valid And Stall

the requests by asserting hif_cmd_stall. For any cycle in which hif_cmd_stall is asserted on the rising edge of the clock, the hif_cmd_valid and all other associated request signals are ignored by the DDRC. When this happens, the interface must hold the request on the interface for another cycle and must not yet decrement the associated credit counter. When the DDRC de-asserts hif_cmd_stall on the rising edge of the clock, the request is accepted and the interface can then de-assert hif_cmd_valid or it can issue the next read, write or request (credits allowing).

The DDRC asserts hif_cmd_stall for any of the following conditions:

1. Software driven self-refresh entry requests.This is to ensure that DDRC is empty when memories are in self-refresh. For more information about this, see “Power Saving Features”.
2. Successful DDRC hardware low power entry requests to self-refresh. This is to ensure that DDRC is empty when memories are in self-refresh. For more information about this, see “Power Saving Features”.
3. Address collisions in the DDRC, where flow control is applied to prevent further commands from entering the DDRC. For more information about this, see “Address Collision Handling".
4. In the event of the Write CAM being full and no corresponding data for any of the write command has yet arrived.
5. Setting DBG1.dis_hif=1, causes hif_cmd_stall to be asserted.

NOTE

If Dual HIF functionality is enabled (DDRC_DUAL_HIF=1), hif_rcmd_stall/hif_wcmd_stall is asserted as follows:

- Conditions (1), (2), (3), and (5) cause both hif_rcmd_stall and hif_wcmd_stall to assert at the same time.
- Condition (4) only affects hif_wcmd_stall.

9.3.2.3.4 Write Requests And Data

When the WR credit count is greater than zero, the interface can issue write requests to the DDRC accordingly. A command is issued to the DDRC by asserting hif_cmd_valid on the rising edge of the clock.

All write request fields must be driven to the appropriate values at the same time. These fields are:

- **hif_cmd_type**: specifies the request type as follows:
  - '00' indicates a write request
  - '01' indicates a read request
  - '11' is not supported
- **hif_cmd_pri**: specifies the priority of the write request
  - In HIF-only configurations (DDRC_INCL_ARB=0) where DDRC_VPRW_EN is set to 0, this signal is 1-bit wide. This field is meaningless for write commands.
  - In Arbiter configurations (DDRC_INCL_ARB=1) or in HIF-only configurations (DDRC_INCL_ARB=0) where DDRC_VPRW_EN is set to 1, this signal is 2-bit wide.

  The encoding for Write is:

  - 2'b00 - NPW (Normal Priority Writes)
  - 2'b01 - VPW (Variable Priority Writes)
  - 2'b10 - Reserved
  - 2'b11 - Reserved

  VPW commands are sent only in configurations where DDRC_VPRW_EN=1
- **hif_cmd_addr**: indicates the address for the write or . Each word on the HIF (hif_wdata/hif_rdata_data) is uniquely addressable. For 1:1 frequency ratio mode configurations each 2 DDR words on the DQ bus are uniquely addressable via hif_cmd_addr. In 1:2 frequency ratio mode configurations every 4 DDR words are uniquely addressable.
- **hif_cmd_wdata_ptr**: a pointer into the requestor's own buffers that can be used to subsequently retrieve the write data. Once a write or request is accepted, this pointer is returned to the interface to retrieve the associated write data. (This field is not required, as requests are always accepted in order; the interface can choose to FIFO these pointers internally and ignore this field.)

The stall mechanism for write or requests is identical to that for read requests as explained in the section “Valid and Stall”.

Writes have three phases of operation: a write command, a write data pointer return, and finally write data provided to the DDRC. The write command phase indicates to the DDRC that a write transfer is required. The write data pointer return phase indicates to the SoC core that the DDRC is ready to receive the write data. The write data phase provides the write data to the DDRC.

In all these three phases, the transactions must follow the same order. The data pointer is returned in the same order in which the command is sent. The write data must also be sent in the same order. The DDRC can perform re-ordering to the commands later, before sending them on the DFI interface, to make efficient use of the SDRAM bandwidth.

9.3.2.3.4.1 Write Data Pointer Return

Following the acceptance of a write or request, the DDRC then returns the write data pointer to the interface logic to retrieve the associated write data. This is done by asserting hif_wdata_ptr_valid on the rising edge of clock. In the same cycle, hif_wdata_ptr indicates the value of the write data pointer. This is same as a pointer that is presented to the DDRC as hif_cmd_wdata_ptr during the write command phase.

There is no mechanism for stalling the acceptance of the write data pointer from the DDRC; the interface logic must present write data for every cycle in which hif_wdata_ptr_valid is asserted by the DDRC. The mechanism to throttle hif_wdata_ptr_valid assertions is for the interface to throttle write or requests presented to the DDRC.

There is no required timing between a write or command presented to the DDRC, and the same pointer is returned as hif_wdata_ptr. Multiple pointers can also be presented to the DDRC with multiple requests before the first is returned to the interface. The order of the pointers returned matches the order of the requests presented to the DDRC.

9.3.2.3.4.2 Write Data

After a write data pointer is returned to the interface logic, the interface logic must retrieve the associated data and present it to the DDRC. Data presented to the DDRC has no required timing relationship to the transfer of the write pointer on the interface. The data, however, must be returned in the same order that the pointers are presented to the interface (same order in which the original requests are made).

The write data can be throttled using hif_wdata_stall signal. The write data is accepted by the DDRC when hif_wdata_valid is detected high and hif_wdata_stall is detected low at the same positive edge of the clock. The signal hif_wdata_stall is asserted by the DDRC during a RMW operation. When the DDRC writes the read data for the RMW operation into the write data SRAM, it blocks the SoC core interface from sending any write data during those cycles by asserting this signal.

The write data SRAM can be implemented internally as synthesized flip-flops or implemented externally to the DDRC as embedded SRAM using DDRC_WDATA_EXTRAM parameter.

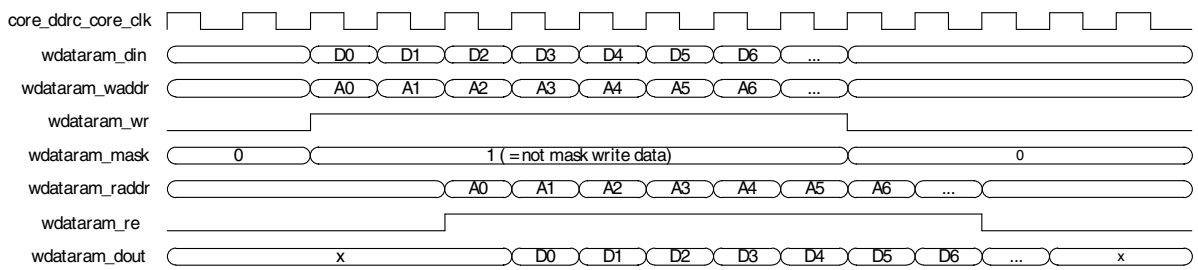


Figure 9-4. Write Data RAM Interface Timing

9.3.2.3.4.2.1 Write Data for the DDRC in BL16 Mode (MEMC_BURST_LENGTH = 16)

Write data is presented to the DDRC with the assertion of hif_wdata_valid. Each request can have one to eight data phases. hif_wdata_valid must be asserted for each data phase and hif_wdata_end must be asserted on the last data phase. hif_wdata is the data. Both hif_wdata_end and hif_wdata are valid only when hif_wdata_valid is '1'.

A normal write or RMW command has eight clocks of data associated with it. In this case, hif_wdata_end must be asserted on the eighth data phase. If the SoC core has less than 8 cycles of write data for a write command, it can choose to send 1, 2, 3, 4, 5, 6 or 7 cycles of data, and hif_wdata_end must be asserted in the appropriate last data phase - these are referred to as Partial Writes. In this case, the DDRC has all the data that it requires 7, 6, 5, 4, 3, 2 or 1 clock cycles earlier than in case of a full write. For these partial writes, the DDRC's behavior is dependent on whether DDRC_PARTIAL_WR is enabled or not:

- DDRC_PARTIAL_WR=0: DDRC still issues the number of SDRAM bursts on the DDR interface that it would issue for a normal (full) write, but masks the unused data phases.
- DDRC_PARTIAL_WR=1: DDRC issues the minimum number of SDRAM bursts on the DDRC interface, depending on the number of HIF write data beats and the HIF address alignment with respect to the SDRAM column address, because SDRAM writes must be sent aligned with BL. This may be smaller than that for a full write.

This analysis assumes 1:1 frequency ratio mode. For 1:2 frequency ratio mode, the number of data beats of the write data on the HIF is halved, so each request has one to four data phases. A Partial Write will have less than four data phases.

9.3.2.3.5 Common Response

The Common Response Interface returns data from the DDRC to the SoC core in response to read requests. The following signals are associated with this functionality:

- hif_rdata_valid
- hif_rdata_end
- hif_rdata_token
- hif_rdata_data

9.3.2.3.5.1 Response Data Ordering

Read data is returned on the HIF in the same order in which it is received from the PHY on the DFI interface. It is independent of whether the read address is aligned or unaligned. For information on how unaligned addresses are handled, see “Sequential/ Interleaved Operations”. For data lane mapping examples where the addresses are aligned, see “Data Lane Mapping Examples”.

9.3.2.3.5.2 Common Response Interface Example

The figure below shows a pair of read requests. The first request is to address A0 with token T0 and length 0. The second request is to A1 with token T1 and length 1. The second read data is returned before the first read data. The DDRC can determine that this sequence is possible due to the state of the open pages and other actions going on in the memories. This improves overall memory bandwidth utilization. The response token indicates which read's data is being provided. Once started, read data for a given token always completes for that token before the next token's read data is returned. The

hif_rdata_valid signal indicates valid data is on the bus and the hif_rdata_end signal indicates that the last data packet is being transmitted. The hif_rdata_valid can deassert in the middle of returning read data for one or more cycles, so hif_rdata_valid must be monitored to validate each cycle of returning read data.

NOTE

Block A: If hif_cmd_length is 1’b1 for this request, 1 cycle of read data comes back and the END signal is asserted in the same cycle.

Block B: If hif_cmd_length is 1’b0 for this request, 2 cycles of read data comes back and the END signal is asserted in the second cycle.

For MEMC_BURST_LENGTH=16, DFI 1:1 configurations, the following is applicable for Block A and Block B in the above figure:

Block A: If hif_cmd_length is 2’b10 for this request, 4 cycle of read data comes back and the END signal is asserted in the fourth cycle.

Block B: If hif_cmd_length is 2’b00 for this request, 8 cycles of read data comes back and the END signal is asserted in the 8th cycle.

9.3.2.3.5.3 LPDDR4 6Gb/12Gb and hif_rdata_addr_err

For LPDDR4 configurations, there is an extra output for DDRC, named hif_rdata_addr_err. This is asserted only in cases where LPDDR4 6Gb/12Gb (ADDRMAP6.lpddr4_6gb_12gb_24gb == 2'b01 or 2'b10) device density is used and the requested address for a read is not valid (when both row address bits MSB and MSB-1 are high). In these cases, DDRC performs a dummy read by changing the physical address to a valid one (row[MSB:MSB-1] changes from 2'b11 to 2'b10). The returned data is masked to 0 and hif_rdata_addr_err is asserted along with the relevant hif_rdata_valid/hif_rdata_token.

9.3.2.4 DFI Interface

The DDRC contains a DFI MC (Memory Controller) interface, which is used to connect to a DFIcompliant PHY, and to transfer address, control and data to the PHY. The DDRC interfaces to a PUB, PUBL, or PUBM2 block which acts as the DFI PHY interface. You can get more information about the DFI interface from the following specifications:

- DDR PHY Interface (DFI) Specification, Revision 3.1 (Preliminary), 21 March 2014 and
- DDR PHY Interface (DFI) Specification, Preliminary DFI 4.0 Specification - Addendum to DFI 3.1, 2 April 2014

The DFI interface is sub-divided into the following interface groups:

- Control Interface
- Write Data Interface
- Read Data Interface
- Update Interface
- Status Interface
- Training Interface (PHY-independent mode supported by the DDRC)
- Low Power Control Interface

9.3.2.4.1 1:1 Frequency Ratio Mode Considerations

The DDRC, including its DFI interface, runs at the single data rate (SDR) clock (referred to as 1:1 frequency ratio mode in the DFI Specification), so all DFI command and address signals are equal in width to the equivalent DDR SDRAM signals. DFI data signals are twice the width of the equivalent DDR SDRAM signals. The DDRC runs with an SDR clock and the PHY operates the SDR to DDR rate conversion on the data buses. In 1:1 frequency ratio mode, core_ddrc_core_clk and CK/CK# run at the same SDR clock frequency.

9.3.2.4.2 1:2 Frequency Ratio Mode Considerations

The DDRC, including its DFI interface runs at the half data rate (HDR) clock (referred to as 1:2 frequency ratio mode in the DFI Specification), so all DFI command and address signals are twice the width of the equivalent DDR SDRAM signals. DFI data signals are four times the width of the equivalent DDR SDRAM signals. The DDRC operates with an HDR clock. The PHY handles HDR-to-SDR conversion on the address bus to memory and HDR/DDR conversion on the data buses. In 1:2 frequency ratio mode, core_ddrc_core_clk runs at half the frequency of CK/CK#.

For DFI command and address signals in DFI 1:2, the DFI protocol refers to phase 0 (_p0)and phase 1 (_p1) but there are no such *_p0 and *_p1 signals in the DDRC. Instead, the lower half of the bits corresponds to phase 0 and the upper half of the bits corresponds to phase 1. For example, in a 4-rank system in DFI 1:2, dfi_cs[7:0] is the case. dfi_cs[3:0] corresponds to dfi_cs_p0 and dfi_cs[7:4] corresponds to dfi_cs_p1.

Similarly, for DFI write data signals in DFI 1:2, the DFI protocol refers to phase 0 (_p0) and phase 1 (_p1). But, there are no such *_p0 and *_p1 signals in the DDRC. Instead, the lower half of the bits correspond to phase 0 and the upper half of the bits correspond to phase 1.

Consider the following examples:

- In system that has 16 bits of SDRAM in DFI 1:2, dfi_wrdata[63:0] is the case. dfi_wrdata[31:0] corresponds to dfi_wrdata_p0 and dfi_wrdata[63:32] corresponds to dfi_wrdata_p1. Similarly, dfi_wrdata_en[3:0] is the case. dfi_wrdata_en[1:0] corresponds to dfi_wrdata_en_p0 and dfi_wrdata_en[3:2] corresponds to dfi_wrdata_en_p1. Similarly, dfi_wrdata_mask[7:0] is the case. dfi_wrdata_mask[3:0] corresponds to dfi_wrdata_mask_p0 and dfi_wrdata_mask[7:4] corresponds to dfi_wrdata_mask_p1.
- In system that has 32 bits of SDRAM in DFI 1:2, dfi_wrdata[127:0] is the case. dfi_wrdata[63:0] corresponds to dfi_wrdata_p0 and dfi_wrdata[127:64] corresponds to dfi_wrdata_p1. Similarly, dfi_wrdata_en[7:0] is the case. dfi_wrdata_en[3:0] corresponds to dfi_wrdata_en_p0 and dfi_wrdata_en[7:4] corresponds to dfi_wrdata_en_p1. Similarly, dfi_wrdata_mask[15:0] is the case. dfi_wrdata_mask[7:0] corresponds to dfi_wrdata_mask_p0 and dfi_wrdata_mask[15:8] corresponds to dfi_wrdata_mask_p1.

Similarly, for dfi_rddata_en in DFI 1:2, the DFI protocol refers to phase 0 (_p0) and phase 1 (_p1). But, there are no such *_p0 and *_p1 signals in the DDRC. Instead, the lower half of the bits correspond to phase 0 and the upper half of the bits correspond to phase 1.

Consider the following examples:

- In system that has 16 bits of SDRAM in DFI 1:2, dfi_rddata_en[3:0] is the case. dfi_rddata_en[1:0] corresponds to dfi_rddata_en_p0 and dfi_rddata_en[3:2] corresponds to dfi_rddata_en_p1.
- In system that has 32 bits of SDRAM in DFI 1:2, dfi_rddata_en[7:0] is the case. dfi_rddata_en[3:0] corresponds to dfi_rddata_en_p0 and dfi_rddata_en[7:4] corresponds to dfi_rddata_en_p1.

For dfi_rddata_valid/dfi_rddata/dfi_rddata_dbi signals in DFI 1:2, the DFI protocol refers to word 0 (_w0) and word 1 (_w1). But, there are no such *_w0 and *_w1 signals in the DDRC. Instead, the lower half of the bits correspond to word 0 and the upper half of the bits correspond to word 1.

Consider the following examples:

- In system that has 16 bits of SDRAM in DFI 1:2, dfi_rddata[63:0] is the case. dfi_rddata[31:0] corresponds to dfi_rddata_w0 and dfi_rddata[63:32] corresponds to dfi_rddata_w1. Similarly, dfi_rddata_valid[3:0] is the case. dfi_rddata_valid[1:0] corresponds to dfi_rddata_valid_w0 and dfi_rddata_valid[3:2] corresponds to dfi_rddata_valid_w1. Similarly, dfi_rddata_dbi[7:0] is the case. dfi_rddata_dbi[3:0] corresponds to dfi_rddata_dbi_w0 and dfi_rddata_dbi[7:4] corresponds to dfi_rddata_dbi_w1.
- In system that has 32 bits of SDRAM in DFI 1:2, dfi_rddata[127:0] is the case. dfi_rddata[63:0] corresponds to dfi_rddata_w0 and dfi_rddata[127:64] corresponds to dfi_rddata_w1. Similarly, dfi_rddata_valid[7:0] is the case. dfi_rddata_valid[3:0] corresponds to dfi_rddata_valid_w0 and dfi_rddata_valid[7:4] corresponds to dfi_rddata_valid_w1. Similarly, dfi_rddata_dbi[15:0] is the case. dfi_rddata_dbi[7:0] corresponds to dfi_rddata_dbi_w0 and dfi_rddata_dbi[15:8] corresponds to dfi_rddata_dbi_w1.

9.3.2.4.3 DFI Write/Read Data to SDRAM Conversion

This conversion is performed in the PHY, and therefore PHY-Specific. The following details are applicable only to Synopsys DDR PHYs; other PHYs may differ from this.

The DFI data (both read and write) should contain data for multiple byte lanes and for multiple data beats (2 in 1:1 mode, 4 in 1:2 mode). The DFI Specification does not explicitly state how this data should be packed within dfi_rddata and dfi_wrdata. The Synopsys DDR PHYs and the DDRC DFI interfaces are designed with the assumption that the data is ordered by beat and then (within each beat) by byte.

For example, for a 2-byte configuration (16 bits of SDRAM) in 1:2 mode, the DFI data order is as follows (from MSB to LSB):

[byte1-beat3, byte0-beat3, byte1-beat2, byte0-beat2, byte1-beat1, byte0-beat1, byte1-beat0, byte0-beat0]

Therefore, the PHY sends the data it receives out to the memory in the following way:

- In beat 0, the data from dfi_wrdata[15:0] are sent on DQ[15:0]
- In beat 1, the data from dfi_wrdata[31:16] are sent on DQ[15:0]

- In beat 2, the data from dfi_wrdata[47:32] are sent on DQ[15:0]
- In beat 3, the data from dfi_wrdata[63:48] are sent on DQ[15:0]

Similarly, for a 2-byte configuration (16 bits of SDRAM) in 1:1 mode, the DFI data order is as follows (from MSB to LSB):

[byte1-beat1, byte0-beat1, byte1-beat0, byte0-beat0]

Therefore, the PHY sends the data it receives out to the memory in the following way:

- In beat 0, the data from dfi_wrdata[15:0] are sent on DQ[15:0]
- In beat 1, the data from dfi_wrdata[31:16] are sent on DQ[15:0]

Similarly, for a 4-byte configuration with 32 bits of SDRAM in 1:2 mode, the DFI data order is as follows (from MSB to LSB):

[byte3-beat3, byte2-beat3, byte1-beat3, byte0-beat3, byte3-beat2, byte2-beat2, byte1-beat2, byte0- beat2, byte3-beat1, byte2-beat1, byte1-beat1, byte0-beat1, byte3-beat0, byte2-beat0, byte1-beat0, byte0-beat0]

Therefore, the PHY sends the data it receives out to the memory in the following way:

- In beat 0, the data from dfi_wrdata[31:0] are sent on DQ[31:0]
- In beat 1, the data from dfi_wrdata[63:32] are sent on DQ[31:0]
- In beat 2, the data from dfi_wrdata[95:64] are sent on DQ[31:0]
- In beat 3, the data from dfi_wrdata[127:96] are sent on DQ[31:0]

Similarly, for a 4-byte configuration (32 bits of SDRAM) in 1:1 mode, the DFI data order is as follows (from MSB to LSB):

[byte3-beat1, byte2-beat1, byte1-beat1, byte0-beat1, byte3-beat0, byte2-beat0, byte1-beat0, byte0-beat0]

Therefore, the PHY sends the data it receives out to the memory in the following way:

- In beat 0, the data from dfi_wrdata[31:0] are sent on DQ[31:0]
- In beat 1, the data from dfi_wrdata[63:32] are sent on DQ[31:0]

For more information, see the addressing example in Figure A-1. Similar ordering is true for dfi_wrdata_en/dfi_wrdata_mask and dfi_rddata_en/dfi_rddata/dfi_rddata_valid/ dfi_rddata_dbi.

NOTE

- If dfi_rddata data slice independence is enabled (DDRC_DFI_RDDATA_PER_BYTE=1), the logic assumes the DDR PHY ordering on bytes/beats for dfi_rddata_valid/dfi_rddata/dfi_rddata_dbi.

9.3.2.4.4 Control Interface

The control interface is a reflection of the SDRAM control interface including address, bank, chip select, row strobe, column strobe, write enable, clock enable and ODT control, as applicable for the memory technology. This interface consists of the following signals:

- dfi_address (width of this signal depends on configuration - 20 bits if 18 bits if MEMC_DDR4 = 1; otherwise 16 bits)
- dfi_bank
- dfi_cas_n
- dfi_cke
- dfi_cs
- dfi_odt
- dfi_ras_n
- dfi_reset_n (this signal only exists if MEMC_DDR4 = 1)
- dfi_we_n
- dfi_bg (this signal only exists if MEMC_DDR4=1)
- dfi_act_n (this signal only exists if MEMC_DDR4 = 1)

For more information about the above signals, refer to “Signal Descriptions”.

9.3.2.4.5 Write Data Interface

The write data interface handles the transmission of write data across the DFI interface. The DFI Specification defines signals and timing relationships. This interface consists of the following signals:

- dfi_wrdata
- dfi_wrdata_en
- dfi_wrdata_mask

For more information about the above signals, refer to “Signal Descriptions”. The timing of this interface is user-configurable, to support all DFI-compliant PHYs through the DFITMG0.dfi_tphy_wrlat and DFITMG0.dfi_tphy_wrdata register (see DDRC_REGS Registers). For DFI 1:2, DFITMG0.dfi_wrdata_use_dfi_phy_clk determines whether DFITMG0.dfi_tphy_wrlat and DFITMG0.dfi_tphy_wrdata are in terms of SDR or HDR clock cycles. If HDR, timing for dfi_wrdata* generation for write commands on phase 0/ phase 1 are the same and dfi_wrdata* signals are aligned to HDR clock. If SDR, timing for dfi_wrdata* generation for write commands on phase 0/phase 1 are treated separately and dfi_wrdata* signals are not guaranteed to be aligned to HDR clock. Check your PHY requirements for correct programming.

In DDR4 configurations (when MEMC_DDR4 is set) or LPDDR4 configurations (when MEMC_LPDDR4 is set), the signal dfi_wrdata_mask can act as Write DBI signal depending on the user programming of the SDRAM mode register. For more information on DBI, see “Data Bus Inversion (DBI)”.

Also, as per JEDEC Specification, the polarity of dfi_wrdata_mask is inverted when using DDR4 memories (if MSTR.ddr4 = 1). Therefore, for DDR4, if a bit of dfi_wrdata_mask is ‘0’, the corresponding byte is masked.

9.3.2.4.5.1 Support for dfi_wrdata_cs/dfi_rddata_cs Signals (DFI 3.1)

There is an optional support for dfi_wrdata_cs and dfi_rddata_cs signals by setting parameter DDRC_DFI_DATA_CS_EN=1. This is a new feature introduced in DFI 3.0, which also applies to DFI 3.1. Enable this feature only if your PHY supports it. These signals are only supported for configurations with MEMC_NUM_RANKS>1. The polarity of these signals is defined by DFIMISC.dfi_data_cs_polarity.

The signal dfi_wrdata_cs is driven as follows:

- dfi_wrdata_cs is driven according to the relevant chip select tphy_wrcslat DFI PHY clock cycles after any command that generates dfi_wrdata. This includes a write, and in DDR4, a PDA (write). It is guaranteed to remain at this value for a minimum of tphy_wrcsgap plus the time for the write data (usually MSTR.burst_rdwr).

  Similarly, for dfi_rddata_c:

- dfi_rddata_cs is driven according to the relevant chip select tphy_rdcslat DFI PHY clock cycles after any command that generates dfi_rddata. This includes a Read, in LPDDR4, a MRR and in DDR4, a MPR (read). It is guaranteed to remain at this value for a minimum of tphy_rdcsgap plus the time for the read data (usually MSTR.burst_rdwr).

The table outlines the DFI timing value and the equivalent register in the DDRC.

Table 9-3. DFI timing values for dfi_wrdata_cs/dfi_rddata_cs

DFI Timing Value	DDRC Register
$t_{phy\_wrcslat}$	DFITMG2.dfi_tphy_wrcslat
$t_{phy\_wrcsgap}$	RANKCTL.diff_rank_wr_gap [1] [2]
$t_{phy\_rdcslat}$	DFITMG2.dfi_tphy_rdcslat
$t_{phy\_rdcsgap}$	RANKCTL.diff_rank_rd_gap[1] [2]

NOTE

1. For configurations with MEMC_FREQ_RATIO=2, this register is in terms of DFI clocks (controller clocks), while tphy_wrcsgap is in terms of DFI PHY clocks.
2. If using DDR4-LRDIMM, refer to TWRWR and TRDRD timing requirements in JEDEC DDR4 Data Buffer (DDR4DB01) Specification.

9.3.2.4.6 Read Data Interface

The read data interface handles the return of read data across the DFI interface. The DFI Specification defines signals and timing relationships. This interface consists of the following signals:

- dfi_rddata
- dfi_rddata_en
- dfi_rddata_valid
- dfi_rddata_dbi (this input signal exists only if MEMC_DDR4=1)

For more information about the above signals, refer to “Signal Descriptions”. The timing of this interface is user-configurable, to support all DFI-compliant PHYs via the DFITMG0.t_rddata_en register.

For DFI 1:2, DFITMG0.dfi_rddata_use_dfi_phy_clk determines whether DFITMG0.dfi_t_rddata_en is in terms of SDR or HDR clock cycles. If HDR, timing for dfi_rddata_en generation for read commands on phase 0/phase 1 is the same and dfi_rddata* signals are aligned to HDR clock. If SDR, timing for dfi_rddata* generation for read commands on phase 0/phase 1 are treated separately and dfi_rddata* signals are not guaranteed to be aligned to HDR clock. Check your PHY requirements for correct programming.

In DDR4 configurations (when MEMC_DDR4 is set), the signal dfi_rddata_dbi is the read DBI signal. For more information on DBI, see “Data Bus Inversion (DBI)”.

The DDRC has been verified with the Synopsys PHYs. It assumes that the DFI parameter tphy_rdlat does not exceed 48 cycles. For mDDR using BL2, it assumes that the DFI parameter tphy_rdlat does not exceed 26 cycles. For both cases, it also assumes that the DFI parameter trddata_en does not exceed the SDRAM read latency (RL).

9.3.2.4.7 Update Interface

During system operation, the system may require updates to internal settings to compensate for environmental conditions. To ensure that updates do not interfere with signals on the SDRAM interface, the DFI interface supports update modes where the DFI read, write, and control interface are suspended from normal activity. The DFI Specification defines both MC-initiated and PHY-initiated updates.

9.3.2.4.7.1 MC-initiated Updates

This interface consists of the following signals:

- dfi_ctrlupd_req
- dfi_ctrlupd_ack
- dfi_ctrlupd_ack2

For more information about the above signals, refer to “Signal Descriptions”.

MC-initiated updates can be acknowledged or ignored by the PHY. DFI MC-initiated updates are performed periodically by DDRC.

9.3.2.4.7.2 PHY-initiated Updates

This interface consists of the following signals:

- dfi_phyupd_req
- dfi_phyupd_type
- dfi_phyupd_ack

For more information about the above signals, see “Signal Descriptions”.

For more details, see “DFI PHY-initiated Update Request”.

9.3.2.4.8 Status Interface

The DFI interface requires status information for initialization and clock control to the SDRAM devices. These signals are used to convey information between the MC and PHY. This interface consists of the following signals:

- dfi_init_start
- dfi_init_complete
- dfi_frequency
- dfi_dram_clk_disable
- dfi_alert_n (this signal replaces dfi_parity_error.

The DDRC does not support the following signals (defined as optional in the DFI Specification):

- dfi_data_byte_disable
- dfi_freq_ratio

9.3.2.4.8.1 Initialization

The dfi_init_start signal is used to trigger the PHY initialization by setting it to 1. It is driven by the DDRC through the APB interface with the DFIMISC.dfi_init_start read-write register field. The signal dfi_init_complete indicates that the PHY has completed its initialization. This information can be read/polled by the DDRC through the APB interface with the DFISTAT.dfi_init_complete read-only register field.

9.3.2.4.8.2 Clock Disabling

The dfi_dram_clk_disable signal depends on setting of PWRCTL.en_dfi_dram_clk_disable (see DDRC_REGS Registers). For more information about this, see “Assertion of dfi_dram_clk_disable”.

9.3.2.4.8.3 Frequency Change

The interface consists of three signals:

- dfi_init_start: Output from the controller to the PHY
- dfi_init_complete: Output from the PHY to the controller
- dfi_frequency: Output from the controller to the PHY

The procedure is as follows:

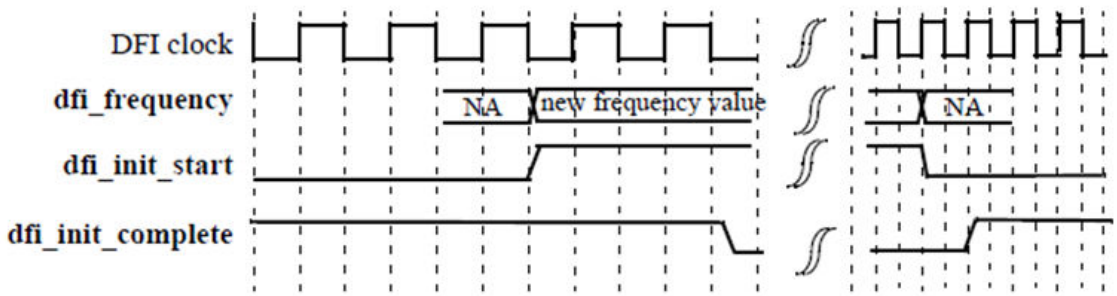


Figure 9-5. Frequency Change

dfi_frequency indicates the operating frequency of the system. This signal should change only at initialization, during a DFI frequency change operation, or other times that the system defines.This signal should be constant during normal operation.The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY, system, or both. For timing, the dfi_frequency signal must be set to a legal value and remain unchanged when dfi_init_start is asserted.The dfi_frequency signal can change any time when dfi_init_start is low and should be ignored at this time.

The behavior of the dfi_init_start signal is dependent on the dfi_init_complete signal.

If the PHY accepts the frequency change request, it must deassert the dfi_init_complete signal within tinit_start cycles of the dfi_init_start assertion. The MC continues to hold the dfi_init_start signal asserted until the clock frequency change has been completed. The de-assertion should be used by the PHY to reinitialize on the new clock frequency.

If the frequency change is not acknowledged (the dfi_init_complete signal remains asserted), the dfi_init_start signal must de-assert after tinit_start cycles. This requirement needs to be handled by the system as dfi_init_start is programmed through the software.

9.3.2.4.9 DFI Training Interface

The additional functions of SDRAM memories allow accurate alignment of critical timing signals. The DFI Specification version 2.1.1 accounts for these functions by providing a DFI training interface.

The DFI Specification defines the following operating modes for the training interface for write leveling, and read leveling:

- No Support
- MC Evaluation (not supported by the DDRC)

- PHY Evaluation
- PHY Independent

  These modes define whether the MC or PHY, or neither, maintains the responsibility for managing the programming of the delay lines and evaluating the response.

  The DFI training interface enables write and read leveling functions in a DFI-compliant PHY.

  In DDRC, the following modes are supported:

- “No Support and PHY Independent Mode”

While interfacing to the DDR PHY, the training is initiated by the PHY Utility Block (PUB) present in the PHY, and PHY independent training mode is used.

9.3.2.4.9.1 No Support and PHY Independent Mode

To enable the No Support and PHY independent mode, the PHY must drive the following signals to “11”:

- dfi_rdlvl_mode
- dfi_rdlvl_gate_mode
- dfi_wrlvl_mode

The above signals are inputs from the DFI-compliant PHY. For these modes, any other DFI Training signals which are outputs can be ignored while any other DFI Training signals that are inputs must be tied to all zeros.

Training is then performed automatically by the PHY, without any intervention from the DDRC. The PUB or PUBL block (part of the PHY) performs all required MRS commands to move the SDRAM to the correct state for each training stage.

If you are using a non-Synopsys PHY that supports PHY independent training mode, DDRC may have to perform these MRS commands.

However, DDRC has no knowledge on when the PHY is ready to execute each training stage, so this must be controlled by the user. For more information on mode register writes, see “Mode Register Write".

9.3.2.4.10 Low Power interface

In a DDR memory subsystem, it is advantageous to place the PHY in a low power state when the DDRC has knowledge that the memory subsystem remains idle for a period of time. Depending on the state of the system, the DDRC communicates state information to the PHY allowing the PHY to enter the appropriate power saving state. This interface consists of signals that are used to inform the PHY of a low power mode opportunity, as well as how quickly the DDRC requires the PHY to resume normal operation. This interface consists of the following signals:

- dfi_lp_req
- dfi_lp_wakeup
- dfi_lp_ack Software control is provided in DFILPCFG0/DFILPCFG1 registers (see DDRC_REGS Registers on page 464) for the following:
- This is an optional interface for a DFI-compliant PHY. The interface signals always exist in the DDRC. But the software can enable this interface in self-refresh and/or power-down and/or deep power-down mode and/or maximum power saving mode - DFILPCFG0.dfi_lp_en_pd, DFILPCFG0.dfi_lp_en_sr, DFILPCFG0.dfi_lp_en_dpd and DFILPCFG1.dfi_lp_en_mpsm respectively.
- Software control over what is driven on dfi_lp_wakeup signal (and its associated timing) in selfrefresh or power-down or deep power-down or maximum power saving mode - DFILPCFG0.dfi_lp_wakeup_pd, DFILPCFG0.dfi_lp_wakeup_sr, DFILPCFG0.dfi_lp_wakeup_dpd and DFILPCFG1.dfi_lp_wakeup_mpsm respectively.
- The timing of this DFI low power interface - DFILPCFG0.dfi_tlp_resp.
- ctl_idle output pin is a sideband signal (not specified in DFI) supported by certain Synopsys PHYs to trigger the PHY's Anti-Aging feature. This is driven the same as dfi_lp_req if enabled via software (DFIMISC.ctl_idle_en=1 and DFI Low Power Interface is also enabled via DFILPCFG0/DFILPCFG1). If configured for Shared AC, dfi_lp_req signal of both data channels must be asserted for ctl_idle to be asserted.

For more information, see "Power Saving Features".

9.3.2.4.11 PHY Master Interface

This interface consists of the following signals:

- dfi_phymstr_req
- dfi_phymstr_cs_state
- dfi_phymstr_state_sel
- dfi_phymstr_type
- dfi_phymstr_ack

For more information about the above signals, see "Signal Descriptions".

For more information about PHY Master interface, see “PHY Master Interface”.

9.3.2.5 APB Interface

The DDRC provides a dedicated APB 3.0 bus interface that is used to access the DDRC software programmable registers. The DDRC converts APB reads and writes into accesses to the internal register file. The APB data width is fixed to 32 bits for compatibility reasons with the DDR PHYs. The APB address width is 12 bits.

All APB interface signals (p*) are synchronous to pclk. pclk can be asynchronous to the core_ddrc_core_clk if the configuration parameter DDRC_P_ASYNC_EN is set. However, pclk frequency must be the same or less than the core_ddrc_core_clk frequency.

9.3.2.5.1 Register Classes

There are three classes of registers in the DDRC:

- Dynamic Registers
- Quasi Dynamic Registers
- Static Registers

Dynamic registers can be written at any time during operation.

Static registers can be written only when the controller is in reset.

Quasi dynamic registers can be written when the controller is in reset and some specific conditions outside reset. To write them, SWCTL.sw_done (see DDRC_REGS Registers on page 464) has to be set to 0 at the beginning of the programming sequence and set back to 1 at the end. After that, SWSTAT.sw_done_ack has to be polled for acknowledge.

9.3.3 Memory Map and register definition

This section includes the DDRC module memory map and detailed descriptions of all registers.

9.3.3.1 DDRC register descriptions

9.3.3.1.1 DDRC memory map

DDRC base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
0h	Master Register0 (MSTR)	32	RW	0304_0001h
4h	Operating Mode Status Register (STAT)	32	RO	0000_0000h
8h	Operating Mode Status Register (MSTR1)	32	RW	0000_0000h
Ch	Operating Mode Status Register (MRCTRL3)	32	RW	0000_0003h
10h	Mode Register Read/Write Control Register 0. (MRCTRL0)	32	RW	0000_0030h
14h	Mode Register Read/Write Control Register 1 (MRCTRL1)	32	RW	0000_0000h
18h	Mode Register Read/Write Status Register (MRSTAT)	32	RO	0000_0000h
1Ch	Mode Register Read/Write Control Register 2 (MRCTRL2)	32	RW	0000_0000h
20h	Temperature Derate Enable Register (DERATEEN)	32	RW	0000_0000h
24h	Temperature Derate Interval Register (DERATEINT)	32	RW	0080_0000h
30h	Low Power Control Register (PWRCTL)	32	RW	0000_0000h
34h	Low Power Timing Register (PWRTMG)	32	RW	0040_2010h
38h	Hardware Low Power Control Register (HWLPCTL)	32	RW	0000_0003h
50h	Refresh Control Register 0 (RFSHCTL0)	32	RW	0021_0000h
54h	Refresh Control Register 1 (RFSHCTL1)	32	RW	0000_0000h
60h	Refresh Control Register 3 (RFSHCTL3)	32	RW	0000_0000h
64h	Refresh Timing Register (RFSHTMG)	32	RW	0062_008Ch
D0h	SDRAM Initialization Register 0 (INIT0)	32	RW	0002_004Eh
D4h	SDRAM Initialization Register 1 (INIT1)	32	RW	0000_0000h
D8h	SDRAM Initialization Register 2 (INIT2)	32	RW	0000_0D05h
DCh	SDRAM Initialization Register 3 (INIT3)	32	RW	0000_0510h
E0h	SDRAM Initialization Register 4 (INIT4)	32	RW	0000_0000h
E4h	SDRAM Initialization Register 5 (INIT5)	32	RW	0010_0004h
E8h	SDRAM Initialization Register 6 (INIT6)	32	RW	0000_0000h
ECh	SDRAM Initialization Register 7 (INIT7)	32	RW	0000_0000h
F0h	DIMM Control Register (DIMMCTL)	32	RW	0000_0000h
F4h	Rank Control Register (RANKCTL)	32	RW	0000_066Fh
100h	SDRAM Timing Register 0 (DRAMTMG0)	32	RW	0F10_1B0Fh
104h	SDRAM Timing Register 1 (DRAMTMG1)	32	RW	0008_0414h
108h	SDRAM Timing Register 2 (DRAMTMG2)	32	RW	0305_060Dh
10Ch	SDRAM Timing Register 3 (DRAMTMG3)	32	RW	0050_400Ch
110h	SDRAM Timing Register 4 (DRAMTMG4)	32	RW	0504_0405h
114h	SDRAM Timing Register 5 (DRAMTMG5)	32	RW	0505_0403h
118h	SDRAM Timing Register 6 (DRAMTMG6)	32	RW	0202_0005h
11Ch	SDRAM Timing Register 7 (DRAMTMG7)	32	RW	0000_0202h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
120h	SDRAM Timing Register 8 (DRAMTMG8)	32	RW	0303_4405h
124h	SDRAM Timing Register 9 (DRAMTMG9)	32	RW	0004_040Dh
128h	SDRAM Timing Register 10 (DRAMTMG10)	32	RW	001C_180Ah
12Ch	SDRAM Timing Register 11 (DRAMTMG11)	32	RW	440C_021Ch
130h	SDRAM Timing Register 12 (DRAMTMG12)	32	RW	0002_0610h
134h	SDRAM Timing Register 13 (DRAMTMG13)	32	RW	1C20_0004h
138h	SDRAM Timing Register 14 (DRAMTMG14)	32	RW	0000_00A0h
13Ch	SDRAM Timing Register 15 (DRAMTMG15)	32	RW	0000_0000h
180h	ZQ Control Register 0 (ZQCTL0)	32	RW	0200_0040h
184h	ZQ Control Register 1 (ZQCTL1)	32	RW	0200_0100h
188h	ZQ Control Register 2 (ZQCTL2)	32	RW	0000_0000h
18Ch	ZQ Status Register (ZQSTAT)	32	RO	0000_0000h
190h	DFI Timing Register 0 (DFITMG0)	32	RW	0702_0002h
194h	DFI Timing Register 1 (DFITMG1)	32	RW	0000_0404h
198h	DFI Low Power Configuration Register 0 (DFILPCFG0)	32	RW	0700_0000h
19Ch	DFI Low Power Configuration Register 1 (DFILPCFG1)	32	RW	0000_0000h
1A0h	DFI Update Register 0 (DFIUPD0)	32	RW	0040_0003h
1A4h	DFI Update Register 1 (DFIUPD1)	32	RW	0001_0001h
1A8h	DFI Update Register 2 (DFIUPD2)	32	RW	8000_0000h
1B0h	DFI Miscellaneous Control Register (DFIMISC)	32	RW	0000_0001h
1B4h	DFI Timing Register 2 (DFITMG2)	32	RW	0000_0202h
1B8h	DFI Timing Register 3 (DFITMG3)	32	RW	0000_0000h
1BCh	DFI Status Register (DFISTAT)	32	RO	0000_0000h
1C0h	DM/DBI Control Register (DBICTL)	32	RW	0000_0001h
200h	Address Map Register 0 (ADDRMAP0)	32	RW	0000_0000h
204h	Address Map Register 1 (ADDRMAP1)	32	RW	0000_0000h
208h	Address Map Register 2 (ADDRMAP2)	32	RW	0000_0000h
20Ch	Address Map Register 3 (ADDRMAP3)	32	RW	0000_0000h
210h	Address Map Register 4 (ADDRMAP4)	32	RW	0000_0000h
214h	Address Map Register 5 (ADDRMAP5)	32	RW	0000_0000h
218h	Address Map Register 6 (ADDRMAP6)	32	RW	0000_0000h
21Ch	Address Map Register 7 (ADDRMAP7)	32	RW	0000_0000h
220h	Address Map Register 8 (ADDRMAP8)	32	RW	0000_0000h
224h	Address Map Register 9 (ADDRMAP9)	32	RW	0000_0000h
228h	Address Map Register 10 (ADDRMAP10)	32	RW	0000_0000h
22Ch	Address Map Register 11 (ADDRMAP11)	32	RW	0000_0000h
240h	ODT Configuration Register (ODTCFG)	32	RW	0400_0400h
244h	ODT/Rank Map Register (ODTMAP)	32	RW	0000_2211h
250h	Scheduler Control Register (SCHED)	32	RW	0000_1005h
254h	Scheduler Control Register 1 (SCHED1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
25Ch	High Priority Read CAM Register 1 (PERFHPR1)	32	RW	0F00_0001h
264h	Low Priority Read CAM Register 1 (PERFLPR1)	32	RW	0F00_007Fh
26Ch	Write CAM Register 1 (PERFWR1)	32	RW	0F00_007Fh
300h	Debug Register 0 (DBG0)	32	RW	0000_0000h
304h	Debug Register 1 (DBG1)	32	RW	0000_0000h
308h	CAM Debug Register (DBGCAM)	32	RO	0000_0000h
30Ch	Command Debug Register (DBGCMD)	32	RW	0000_0000h
310h	Status Debug Register (DBGSTAT)	32	RO	0000_0000h
320h	Software Register Programming Control Enable (SWCTL)	32	RW	0000_0001h
324h	Software Register Programming Control Status (SWSTAT)	32	RO	0000_0001h
36Ch	AXI Poison Configuration Register. (POISONCFG)	32	RW	0011_0011h
370h	AXI Poison Status Register (POISONSTAT)	32	RO	0000_0000h
3FCh	Port Status Register (PSTAT)	32	RO	0000_0000h
400h	Port Common Configuration Register (PCCFG)	32	RW	0000_0000h
404h	Port n Configuration Read Register (PCFGR_0)	32	RW	0000_0000h
408h	Port n Configuration Write Register (PCFGW_0)	32	RW	0000_4000h
490h	Port n Control Register (PCTRL_0)	32	RW	0000_0000h
494h	Port n Read QoS Configuration Register 0 (PCFGQOS0_0)	32	RW	0000_0000h
498h	Port n Read QoS Configuration Register 1 (PCFGQOS1_0)	32	RW	0000_0000h
49Ch	Port n Write QoS Configuration Register 0 (PCFGWQOS0_0)	32	RW	0000_0000h
4A0h	Port n Write QoS Configuration Register 1 (PCFGWQOS1_0)	32	RW	0000_0000h
2020h	[SHADOW] Temperature Derate Enable Register (DERATEEN_SHADOW)	32	RW	0000_0000h
2024h	[SHADOW] Temperature Derate Interval Register (DERATEINT_SHADOW)	32	RW	0080_0000h
2050h	[SHADOW] Refresh Control Register 0 (RFSHCTL0_SHADOW)	32	RW	0021_0000h
2064h	[SHADOW] Refresh Timing Register (RFSHTMG_SHADOW)	32	RW	0062_008Ch
20DCh	[SHADOW] SDRAM Initialization Register 3 (INIT3_SHADOW)	32	RW	0000_0510h
20E0h	[SHADOW] SDRAM Initialization Register 4 (INIT4_SHADOW)	32	RW	0000_0000h
20E8h	[SHADOW] SDRAM Initialization Register 6 (INIT6_SHADOW)	32	RW	0000_0000h
20ECh	[SHADOW] SDRAM Initialization Register 7 (INIT7_SHADOW)	32	RW	0000_0000h
2100h	[SHADOW] SDRAM Timing Register 0 (DRAMTMG0_SHADOW)	32	RW	0F10_1B0Fh
2104h	[SHADOW] SDRAM Timing Register 1 (DRAMTMG1_SHADOW)	32	RW	0008_0414h
2108h	[SHADOW] SDRAM Timing Register 2 (DRAMTMG2_SHADOW)	32	RW	0305_060Dh
210Ch	[SHADOW] SDRAM Timing Register 3 (DRAMTMG3_SHADOW)	32	RW	0050_400Ch
2110h	[SHADOW] SDRAM Timing Register 4 (DRAMTMG4_SHADOW)	32	RW	0504_0405h
2114h	[SHADOW] SDRAM Timing Register 5 (DRAMTMG5_SHADOW)	32	RW	0505_0403h
2118h	[SHADOW] SDRAM Timing Register 6 (DRAMTMG6_SHADOW)	32	RW	0202_0005h
211Ch	[SHADOW] SDRAM Timing Register 7 (DRAMTMG7_SHADOW)	32	RW	0000_0202h
2120h	[SHADOW] SDRAM Timing Register 8 (DRAMTMG8_SHADOW)	32	RW	0303_4405h

Table continues on the next page...

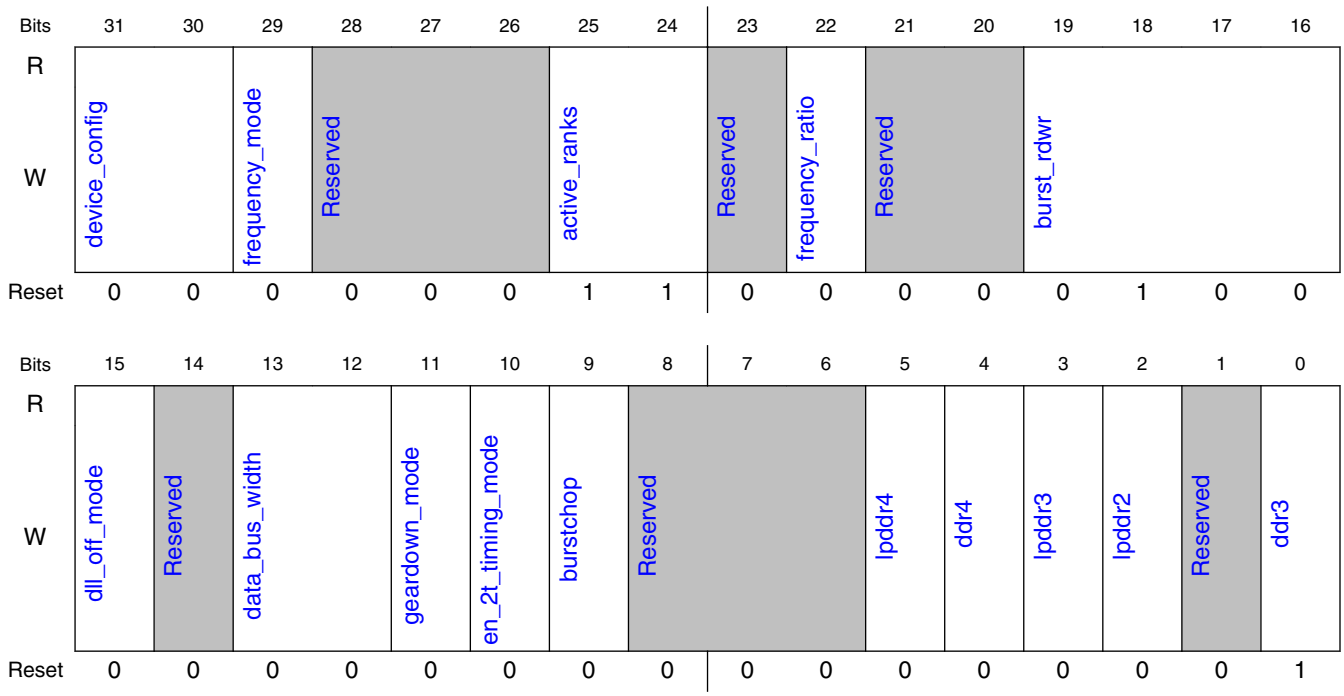
Offset	Register	Width (In bits)	Access	Reset value
2124h	[SHADOW] SDRAM Timing Register 9 (DRAMTMG9_SHADOW)	32	RW	0004_040Dh
2128h	[SHADOW] SDRAM Timing Register 10 (DRAMTMG10_SHADOW)	32	RW	001C_180Ah
212Ch	[SHADOW] SDRAM Timing Register 11 (DRAMTMG11_SHADOW)	32	RW	440C_021Ch
2130h	[SHADOW] SDRAM Timing Register 12 (DRAMTMG12_SHADOW)	32	RW	0002_0610h
2134h	[SHADOW] SDRAM Timing Register 13 (DRAMTMG13_SHADOW)	32	RW	1C20_0004h
2138h	[SHADOW] SDRAM Timing Register 14 (DRAMTMG14_SHADOW)	32	RW	0000_00A0h
213Ch	[SHADOW] SDRAM Timing Register 15 (DRAMTMG15_SHADOW)	32	RW	0000_0000h
2180h	[SHADOW] ZQ Control Register 0 (ZQCTL0_SHADOW)	32	RW	0200_0040h
2190h	[SHADOW] DFI Timing Register 0 (DFITMG0_SHADOW)	32	RW	0702_0002h
2194h	[SHADOW] DFI Timing Register 1 (DFITMG1_SHADOW)	32	RW	0000_0404h
21B4h	[SHADOW] DFI Timing Register 2 (DFITMG2_SHADOW)	32	RW	0000_0202h
21B8h	[SHADOW] DFI Timing Register 3 (DFITMG3_SHADOW)	32	RW	0000_0000h
2240h	[SHADOW] ODT Configuration Register (ODTCFG_SHADOW)	32	RW	0400_0400h

9.3.3.1.2 Master Register0 (MSTR)

9.3.3.1.2.1 Offset

Register	Offset
MSTR	0h

9.3.3.1.2.2 Diagram



9.3.3.1.2.3 Fields

Field	Function
31-30 device_config	Indicates the configuration of the device used in the system. 00b - x4 device 01b - x8 device 10b - x16 device 11b - x32 device
29 frequency_mode	Choose which registers are used. 0b - Original Registers 1b - Shadow Registers
28-26 —	Reserved
25-24 active_ranks	Only present for multi-rank configurations. Each bit represents one rank. For two-rank configurations, only bits[25:24] are present. • 1 - populated • 0 - unpopulated LSB is the lowest rank number. For 2 ranks following combinations are legal • 01 - One rank • 11 - Two ranks • Others- Reserved For 4 ranks following combinations are legal:

Table continues on the next page...

Field	Function
	• 0001 - One rank • 0011 - Two ranks • 1111 - Four ranks •
23 —	Reserved
22 frequency_ratio	Selects the Frequency Ratio 0b - 1:2 Mode 1b - 1:1 Mode
21-20 —	Reserved
19-16 burst_rdwr	SDRAM burst length used All other values are reserved. This controls the burst size used to access the SDRAM. This must match the burst length mode register setting in the SDRAM. (For BC4/8 on-the-fly mode of DDR3 and DDR4, set this field to 0x0100) Burst length of 2 is not supported with AXI ports when MEMC_BURST_LENGTH is 8. Burst length of 2 is only supported when the controller is operating in 1:1 frequency mode 0001b - Burst length of 2 (only supported for mDDR) 0010b - Burst length of 4 0100b - Burst length of 8 1000b - Burst length of 16 (only supported for mDDR, LPDDR2, and LPDDR4)
15 dll_off_mode	Set to 1 when the DDRC and DRAM has to be put in DLL-off mode for low frequency operation. Set to 0 to put DDRC and DRAM in DLL-on mode for normal frequency operation. If DDR4 CRC/parity retry is enabled (CRCPARCTL1.crc_parity_retry_enable = 1), dll_off_mode is not supported, and this bit must be set to '0'.
14 —	Reserved
13-12 data_bus_width	Selects proportion of DQ bus width that is used by the SDRAM - 00 - Full DQ bus width to SDRAM - 01 - Half DQ bus width to SDRAM - 10 - Quarter DQ bus width to SDRAM - 11 - Reserved Note that half bus width mode is only supported when the SDRAM bus width is a multiple of 16, and quarter bus width mode is only supported when the SDRAM bus width is a multiple of 32 and the configuration parameter MEMC_QBUS_SUPPORT is set. Bus width refers to DQ bus width (excluding any ECC width).
11 geardown_mode	1 indicates put the DRAM in geardown mode (2N) and 0 indicates put the DRAM in normal mode (1N). This register can be changed, only when the Controller is in self-refresh mode. This signal must be set the same value as MR3 bit A3. Note: Geardown mode is not supported if the configuration parameter MEMC_CMD_RTN2IDLE is set Note: Geardown mode is not supported if the configuration parameter DDRC_SHARED_AC is set (in Shared-AC mode) and the register value is don't care
10 en_2t_timing_mode	If 1, then DDRC uses 2T timing. Otherwise, uses 1T timing. In 2T timing, all command signals (except chip select) are held for 2 clocks on the SDRAM bus. Chip select is asserted on the second cycle of the command Note: 2T timing is not supported in LPDDR2/LPDDR3/LPDDR4 mode Note: 2T timing is not supported if the configuration parameter MEMC_CMD_RTN2IDLE is set Note: 2T timing is not supported in DDR4 geardown mode. Note: 2T timing is not supported in Shared-AC dual channel mode and the register value is don't care.
9 burstchop	When set, enable burst-chop (BC4 or 8 on-the-fly) in DDR3/DDR4. Burst Chop for Reads is exercised only in HIF configurations (DDRC_INCL_ARB not set) and if in full bus width mode (MSTR.data_bus_width = 00) and if MEMC_BURST_LENGTH=8 or 16. Burst Chop for Writes is exercised only if Partial Writes enabled (DDRC_PARTIAL_WR=1) and if CRC is disabled (CRCPARCTL1.crc_enable = 0). If DDR4 CRC/parity retry is enabled (CRCPARCTL1.crc_parity_retry_enable = 1), burst chop is not supported, and this bit must be set to '0'. BC4 (fixed) mode is not supported.
8-6	Reserved

Table continues on the next page...

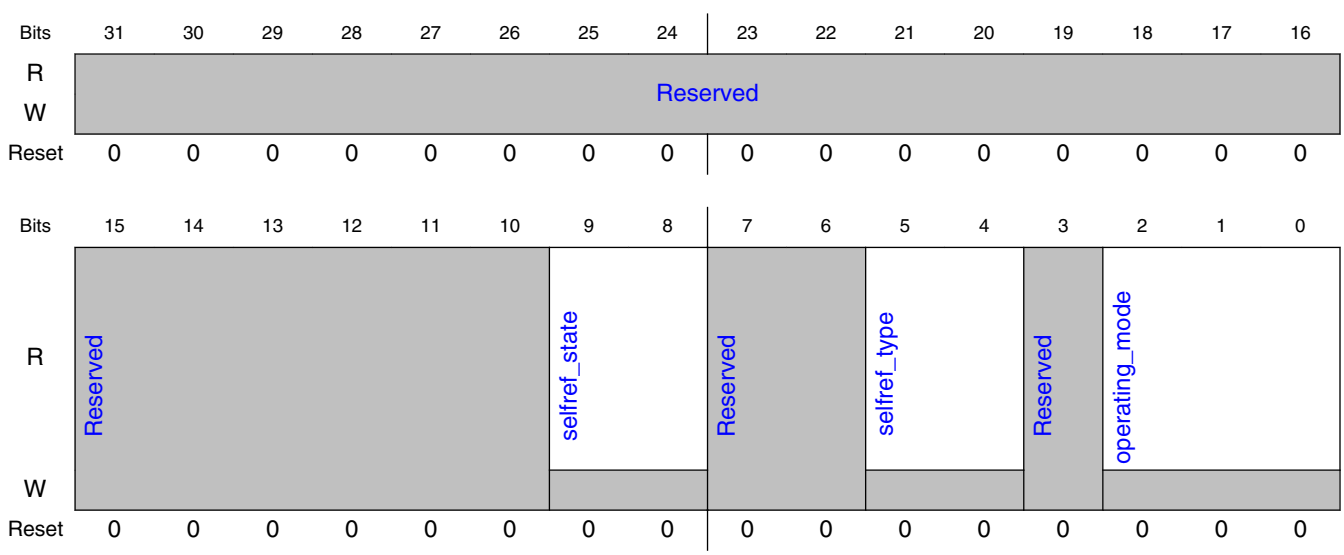
Field	Function
—	
5 lpddr4	Select LPDDR4 SDRAM - 1 - LPDDR4 SDRAM device in use. - 0 - non-LPDDR4 device in use Present only in designs configured to support LPDDR4.
4 ddr4	Select DDR4 SDRAM - 1 - DDR4 SDRAM device in use. - 0 - non-DDR4 device in use Present only in designs configured to support DDR4.
3 lpddr3	Select LPDDR3 SDRAM - 1 - LPDDR3 SDRAM device in use. - 0 - non-LPDDR3 device in use Present only in designs configured to support LPDDR3.
2 lpddr2	Select LPDDR2 SDRAM - 1 - LPDDR2 SDRAM device in use. - 0 - non-LPDDR2 device in use Present only in designs configured to support LPDDR2.
1 —	Reserved
0 ddr3	Select DDR3 SDRAM - 1 - DDR3 SDRAM device in use - 0 - non-DDR3 SDRAM device in use Only present in designs that support DDR3.

9.3.3.1.3 Operating Mode Status Register (STAT)

9.3.3.1.3.1 Offset

Register	Offset
STAT	4h

9.3.3.1.3.2 Diagram



9.3.3.1.3.3 Fields

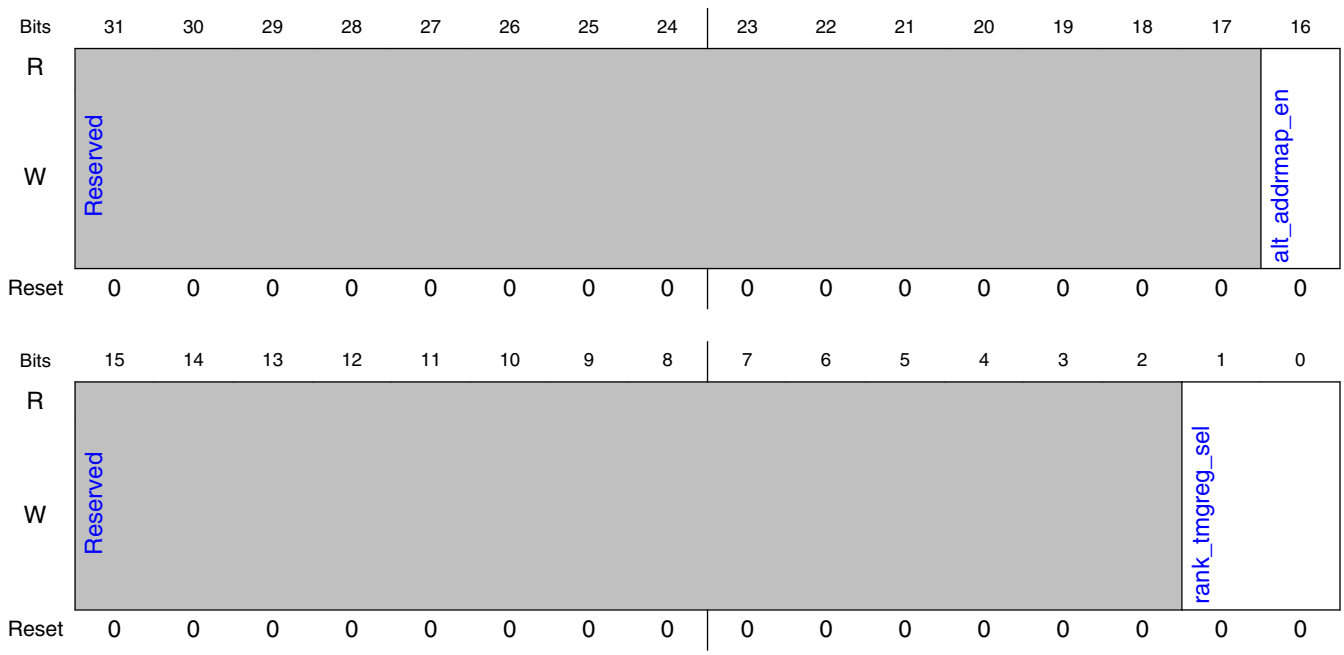
Field	Function
31-10 —	Reserved
9-8 selfref_state	Self refresh state. This indicates self refresh or self refresh power down state for LPDDR4. This register is used for frequency change and MRR/MRW access during self refresh. 00b - SDRAM is not in Self Refresh. 01b - Self refresh 1 10b - Self refresh power down 11b - Self refresh
7-6 —	Reserved
5-4 selfref_type	Flags if Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4) is entered and if it was under Automatic Self Refresh control only or not. 00b - SDRAM is not in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4). If retry is enabled by CRCPARCTRL1.crc_parity_retry_enable, this also indicates SRE command is still in parity error window or retry is in-progress. 10b - SDRAM is in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which was not caused solely under Automatic Self Refresh control. It could have been caused by Hardware Low Power Interface and/or Software (reg_ddrc_selfref_sw). If retry is enabled, this guarantees SRE command is executed correctly without parity 11b - SDRAM is in Self Refresh (except LPDDR4) or SR-Powerdown (LPDDR4), which was caused by Automatic Self Refresh only. If retry is enabled, this guarantees SRE command is executed correctly without parity error.
3 —	Reserved
2-0 operating_mode	Operating mode This is 3-bits wide in configurations with mDDR/LPDDR2/LPDDR3/LPDDR4/DDR4 support and 2-bits in all other configurations. Non-mDDR/LPDDR2/LPDDR3/LPDDR4 and non-DDR4 designs: • 00 - Init • 01 - Normal • 10 - Power-down • 11 - Self refresh mDDR/LPDDR2/LPDDR3 or DDR4 designs: • 000 - Init • 001 - Normal • 010 - Power-down • 011 - Self refresh • 1XX - Deep power-down / Maximum Power Saving Mode LPDDR4 designs: • 000 - Init • 001 - Normal • 010 - Power-down • 011 - Self refresh / Self refresh power-down

9.3.3.1.4 Operating Mode Status Register (MSTR1)

9.3.3.1.4.1 Offset

Register	Offset
MSTR1	8h

9.3.3.1.4.2 Diagram



9.3.3.1.4.3 Fields

Field	Function
31-17 —	Reserved
16 alt_addrmap_en	Enable Alternative Address Map 0b - Disable Alternative Address Map 1b - Enable Alternative Address Map
15-2 —	Reserved
1-0 rank_tmgreg_sel	rank_tmgreg_sel Indicates which register set is used for each rank. Each bit represents one rank. (LSB is the lowest rank number.)

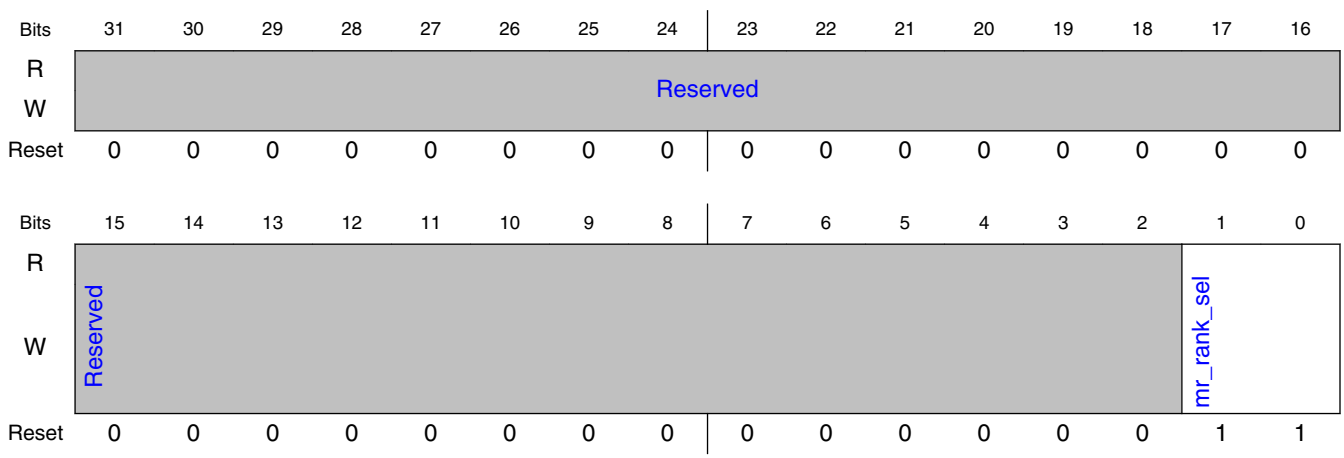
Field	Function
	NOTE: Only some of timing registers in DRAMTMGx are duplicated as MRAMTMGx for MRAM and its width is expanded for MRAM. This register can switch the duplicate registers, otherwise use DRAMTMGx. This register is only for DDR4 ST-MRAM rank, otherwise must be set to 0. This register must be set up while the Controller is in reset. 00b - USE DRAMTMGx registers for the rank 01b - USE MRAMTMGx registers for the rank

9.3.3.1.5 Operating Mode Status Register (MRCTRL3)

9.3.3.1.5.1 Offset

Register	Offset
MRCTRL3	Ch

9.3.3.1.5.2 Diagram



9.3.3.1.5.3 Fields

Field	Function
31-2 —	Reserved
1-0 mr_rank_sel	mr_rank_sel Controls which rank is accessed by MRCTRL0.mr_wr.

Field	Function
	Normally, it is desired to access all ranks, so all bits should be set to 1. However, for multi-rank UDIMMs/ RDIMMs/LRDIMMs which implement address mirroring, it may be necessary to access ranks individually. Examples (assume uMCTL2 is configured for 8 ranks): • 0x01 - select rank 0 only • 0x02 - select rank 1 only • 0x05 - select ranks 0 and 2

9.3.3.1.6 Mode Register Read/Write Control Register 0. (MRCTRL0)

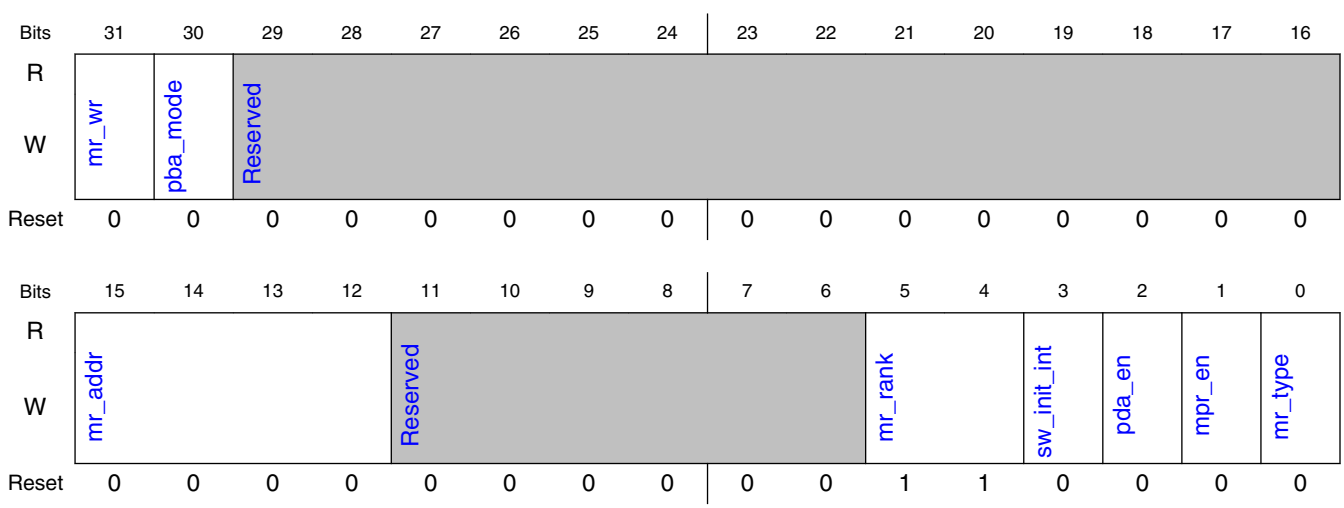
9.3.3.1.6.1 Offset

Register	Offset
MRCTRL0	10h

9.3.3.1.6.2 Function

Mode Register Read/Write Control Register 0. Note: Do not enable more than one of the following fields simultaneously: - sw_init_int - pda_en - mpr_en

9.3.3.1.6.3 Diagram



9.3.3.1.6.4 Fields

Field	Function
31 mr_wr	Setting this register bit to 1 triggers a mode register read or write operation. When the MR operation is complete, the DDRC automatically clears this bit. The other register fields of this register must be written in a separate APB transaction, before setting this mr_wr bit. It is recommended NOT to set this signal if in Init, Deep power-down or MPSM operating modes.
30 pba_mode	Indicates whether PBA access is executed. When setting this bit to 1 along with setting pda_en to 1, DDRC initiates PBA access instead of PDA access. - 0 - Per DRAM Addressability mode - 1 - Per Buffer Addressability mode The completion of PBA access is confirmed by MRSTAT.pda_done in the same way as PDA.
29-16 —	Reserved
15-12 mr_addr	Address of the mode register that is to be written to. Don't Care for LPDDR2/LPDDR3/LPDDR4 (see MRCTRL1.mr_data for mode register addressing in LPDDR2/LPDDR3/LPDDR4) This signal is also used for writing to control words of the register chip on RDIMMs/LRDIMMs. In that case, it corresponds to the bank address bits sent to the RDIMM/LRDIMM In case of DDR4, the bit[3:2] corresponds to the bank group bits. Therefore, the bit[3] as well as the bit[2:0] must be set to an appropriate value which is considered both the Address Mirroring of UDIMMs/RDIMMs/LRDIMMs and the Output Inversion of RDIMMs/LRDIMMs. 0000b - MR0 0001b - MR1 0010b - MR2 0011b - MR3 0100b - MR4 0101b - MR5 0110b - MR6 0111b - MR7
11-6 —	Reserved
5-4 mr_rank	Controls which rank is accessed by MRCTRL0.mr_wr. Normally, it is desired to access all ranks, so all bits should be set to 1. However, for multi-rank UDIMMs/RDIMMs/LRDIMMs which implement address mirroring, it may be necessary to access ranks individually. Examples (assume DDRC is configured for 4 ranks): 0x1 - select rank 0 only 0x2 - select rank 1 only 0x5 - select ranks 0 and 2 0xA - select ranks 1 and 3 0xF - select ranks 0, 1, 2 and 3
3 sw_init_int	Indicates whether Software intervention is allowed via MRCTRL0/MRCTRL1 before automatic SDRAM initialization routine or not. For DDR4, this bit can be used to initialize the DDR4 RCD (MR7) before automatic SDRAM initialization. For LPDDR4, this bit can be used to program additional mode registers before automatic SDRAM initialization if necessary. In LPDDR4 independent channel mode, note that this must be programmed to both channels beforehand. Note that this must be cleared to 0 after completing Software operation. Otherwise, SDRAM initialization routine will not re-start. 0b - Software intervention is not allowed 1b - Software intervention is allowed
2 pda_en	Indicates whether the mode register operation is MRS in PDA mode or not. Note that when pba_mode=1, PBA access is initiated instead of PDA access. 0b - MRS 1b - MRS in Per DRAM Addressability
1 mpr_en	Indicates whether the mode register operation is MRS or WR/RD for MPR (only supported for DDR4). 0b - MRS 1b - WR/RD for MPR
0 mr_type	Indicates whether the mode register operation is read or write. Only used for LPDDR2/LPDDR3/LPDDR4/DDR4.

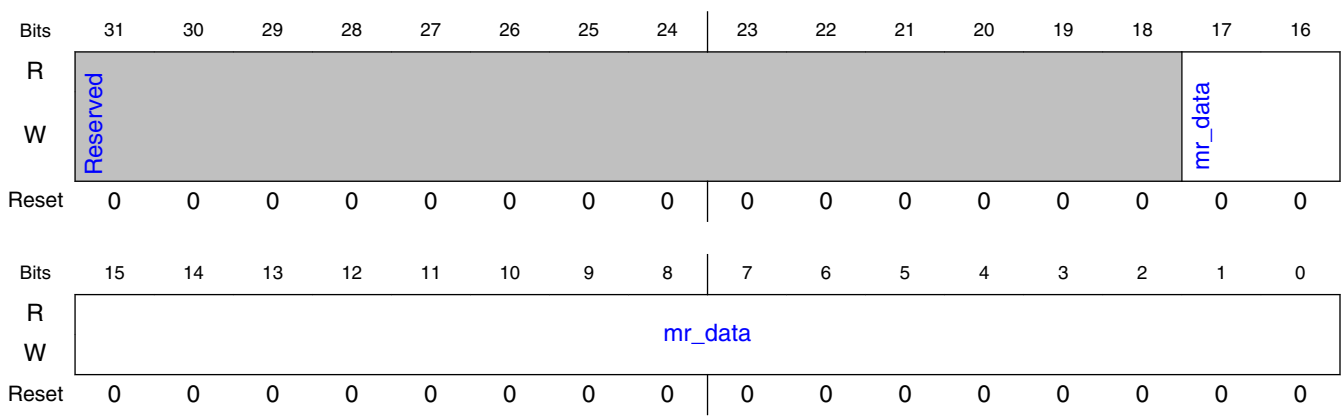
Field	Function
	0b - Write 1b - Read

9.3.3.1.7 Mode Register Read/Write Control Register 1 (MRCTRL1)

9.3.3.1.7.1 Offset

Register	Offset
MRCTRL1	14h

9.3.3.1.7.2 Diagram



9.3.3.1.7.3 Fields

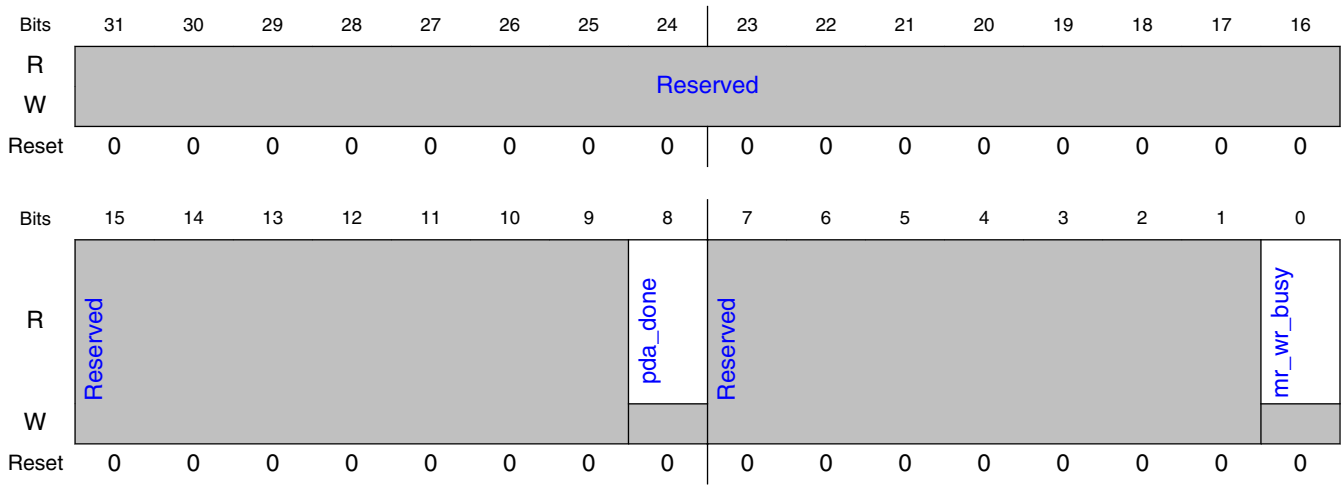
Field	Function
31-18 —	Reserved
17-0 mr_data	Mode register write data for all non-LPDDR2/non-LPDDR3/non-LPDDR4 modes. For LPDDR2/LPDDR3/LPDDR4, MRCTRL1[15:0] are interpreted as [15:8] MR Address [7:0] MR data for writes, don't care for reads. This is 18-bits wide in configurations with DDR4 support and 16-bits in all other configurations.

9.3.3.1.8 Mode Register Read/Write Status Register (MRSTAT)

9.3.3.1.8.1 Offset

Register	Offset
MRSTAT	18h

9.3.3.1.8.2 Diagram



9.3.3.1.8.3 Fields

Field	Function
31-9 —	Reserved
8 pda_done	The SoC core may initiate a MR write operation in PDA/PBA mode only if this signal is low. This signal goes high when three consecutive MRS commands related to the PDA/PBA mode are issued to the SDRAM. This signal goes low when MRCTRL0.pda_en becomes 0. Therefore, it is recommended to write MRCTRL0.pda_en to 0 after this signal goes high in order to prepare to perform PDA operation next time 0b - Indicates that mode register write operation related to PDA/PBA is in progress or has not started yet. 1b - Indicates that mode register write operation related to PDA/PBA has competed.
7-1 —	Reserved
0 mr_wr_busy	The SoC core may initiate a MR write operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the MRW/MRR request. It goes low when the MRW/MRR command is issued to the SDRAM. It is recommended not to perform MRW/MRR commands when 'MRSTAT.mr_wr_busy' is high. 0b - Indicates that the SoC core can initiate a mode register write operation 1b - Indicates that mode register write operation is in progress

9.3.3.1.9 Mode Register Read/Write Control Register 2 (MRCTRL2)

9.3.3.1.9.1 Offset

Register	Offset
MRCTRL2	1Ch

9.3.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr_device_sel															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr_device_sel															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.9.3 Fields

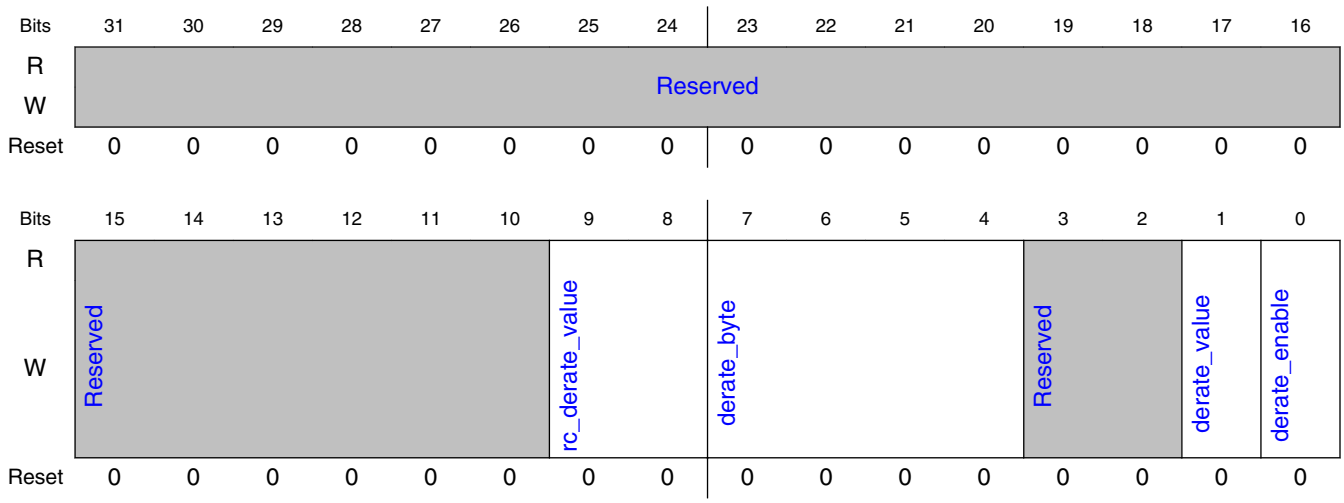
Field	Function
31-0 mr_device_sel	Indicates the device(s) to be selected during the MRS that happens in PDA mode. Each bit is associated with one device. For example, bit[0] corresponds to Device 0, bit[1] to Device 1 etc. A '1' should be programmed to indicate that the MRS command should be applied to that device. A '0' indicates that the MRS commands should be skipped for that device.

9.3.3.1.10 Temperature Derate Enable Register (DERATEEN)

9.3.3.1.10.1 Offset

Register	Offset
DERATEEN	20h

9.3.3.1.10.2 Diagram



9.3.3.1.10.3 Fields

Field	Function
31-10 —	Reserved
9-8 rc_derate_value	Derate value of tRC for LPDDR4. Present only in designs configured to support LPDDR4. The required number of cycles for derating can be determined by dividing 3.75ns by the core_ddrc_core_clk period, and rounding up the next integer. 00b - Derating uses +1 01b - Derating uses +2 10b - Derating uses +3 11b - Derating uses +4
7-4 derate_byte	Derate byte Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 Indicates which byte of the MRR data is used for derating. The maximum valid value depends on MEMC_DRAM_TOTAL_DATA_WIDTH.
3-2 —	Reserved
1 derate_value	Derate value. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 Set to 0 for all LPDDR2 speed grades as derating value of +1.875 ns is less than a core_ddrc_core_clk period. For LPDDR3/4, if the period of core_ddrc_core_clk is less than 1.875ns, this register field should be set to 1; otherwise it should be set to 0. 0b - Derating uses +1 1b - Derating uses +2
0 derate_enable	Enables derating. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4. This field must be set to '0' for non-LPDDR2/LPDDR3/LPDDR4 mode. 0b - Timing parameter derating is disabled 1b - Timing parameter derating is enabled using MR4 read value.

9.3.3.1.11 Temperature Derate Interval Register (DERATEINT)

9.3.3.1.11.1 Offset

Register	Offset
DERATEINT	24h

9.3.3.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.11.3 Fields

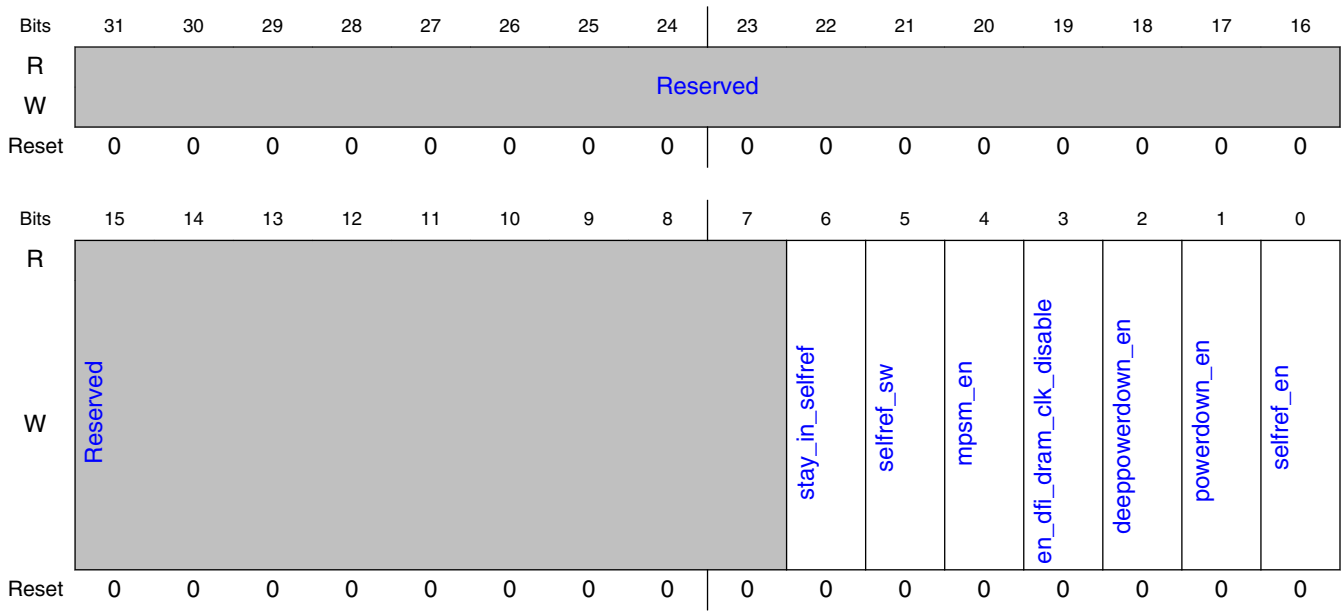
Field	Function
31-0 mr4_read_interval	Interval between two MR4 reads, used to derate the timing parameters. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4. This register must not be set to zero. Unit: DFI clock cycle.

9.3.3.1.12 Low Power Control Register (PWRCTL)

9.3.3.1.12.1 Offset

Register	Offset
PWRCTL	30h

9.3.3.1.12.2 Diagram



9.3.3.1.12.3 Fields

Field	Function
31-7 —	Reserved
6 stay_in_selfref	Self refresh state is an intermediate state to enter to Self refresh power down state or exit Self refresh power down state for LPDDR4. This register controls transition from the Self refresh state. - 1 - Prohibit transition from Self refresh state - 0 - Allow transition from Self refresh state 0b - 1b -
5 selfref_sw	A value of 1 to this register causes system to move to Self Refresh state immediately, as long as it is not in INIT or DPD/MPSM operating_mode. This is referred to as Software Entry/Exit to Self Refresh. 0b - Software Exit from Self Refresh 1b - Software Entry to Self Refresh
4 mpsm_en	When this is 1, the DDRC puts the SDRAM into maximum power saving mode when the transaction store is empty. This register must be reset to '0' to bring DDRC out of maximum power saving mode. Present only in designs configured to support DDR4. For non-DDR4, this register should not be set to 1. Note that MPSM is not supported when using a DDR PHY, if the PHY parameter DDRC_AC_CS_USE is disabled, as the MPSM exit sequence requires the chip-select signal to toggle. FOR PERFORMANCE ONLY.
3 en_dfi_dram_clk_disable	Enable the assertion of dfi_dram_clk_disable whenever a clock is not required by the SDRAM. If set to 0, dfi_dram_clk_disable is never asserted. Assertion of dfi_dram_clk_disable is as follows: In DDR2/DDR3, can only be asserted in Self Refresh. In DDR4, can be asserted in following: in Self Refresh in Maximum Power Saving Mode In mDDR/LPDDR2/LPDDR3, can be asserted in following: in Self Refresh in Power Down in Deep Power Down during Normal operation (Clock Stop) In LPDDR4, can be asserted in following: in Self Refresh Power Down in Power Down during Normal operation (Clock Stop)
2 deeppowerdown_en	When this is 1, DDRC puts the SDRAM into deep power-down mode when the transaction store is empty. This register must be reset to '0' to bring DDRC out of deep power-down mode. Controller performs automatic SDRAM initialization on deep power-down exit. Present only in designs configured to support

Table continues on the next page...

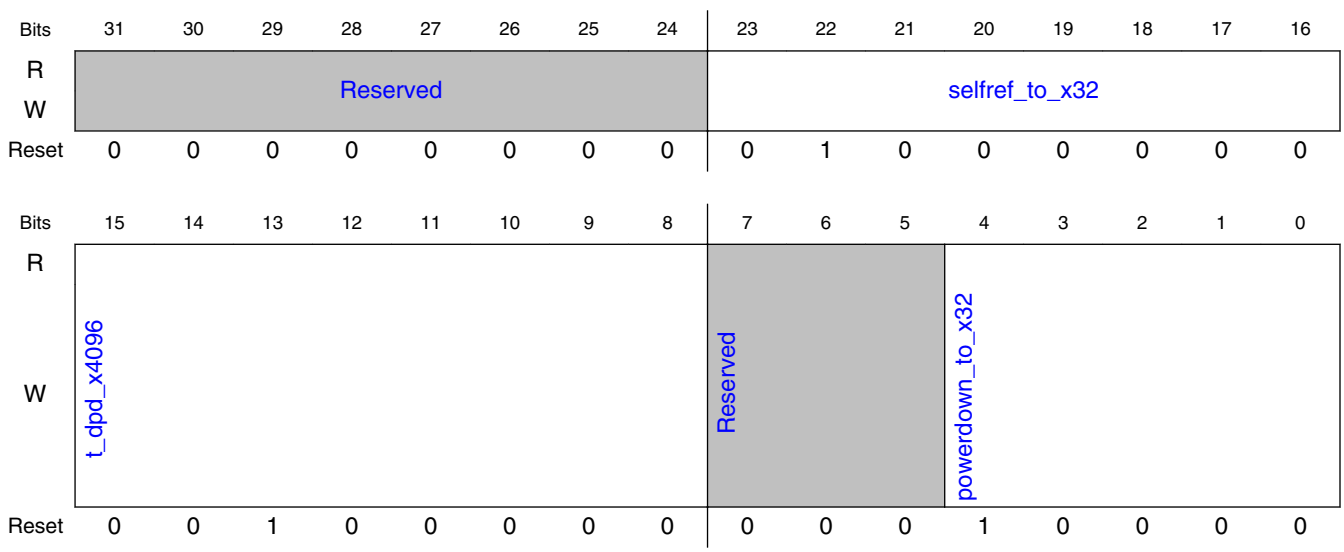
Field	Function
	mDDR or LPDDR2 or LPDDR3. For non-mDDR/non-LPDDR2/non-LPDDR3, this register should not be set to 1. FOR PERFORMANCE ONLY.
1 powerdown_en	If true then the DDRC goes into power-down after a programmable number of cycles "maximum idle clocks before power down" (PWRTMG.powerdown_to_x32). This register bit may be re-programmed during the course of normal operation.
0 selfref_en	If true then the DDRC puts the SDRAM into Self Refresh after a programmable number of cycles "maximum idle clocks before Self Refresh (PWRTMG.selfref_to_x32)". This register bit may be re-programmed during the course of normal operation.

9.3.3.1.13 Low Power Timing Register (PWRTMG)

9.3.3.1.13.1 Offset

Register	Offset
PWRTMG	34h

9.3.3.1.13.2 Diagram



9.3.3.1.13.3 Fields

Field	Function
31-24 —	Reserved

Table continues on the next page...

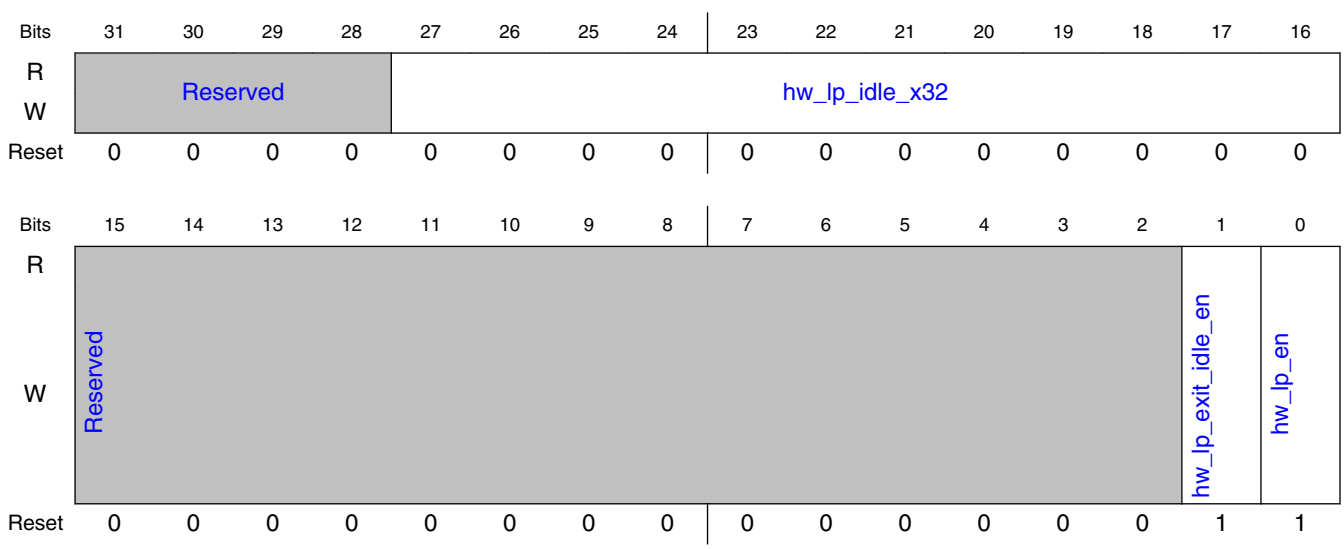
Field	Function
23-16 selfref_to_x32	After this many clocks of the DDRC command channel being idle the DDRC automatically puts the SDRAM into Self Refresh. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.selfref_en. Unit: Multiples of 32 DFI clocks. FOR PERFORMANCE ONLY.
15-8 t_dpd_x4096	Minimum deep power-down time. For mDDR, value from the JEDEC specification is 0 as mDDR exits from deep power-down mode immediately after PWRCTL.deeppowerdown_en is de-asserted. For LPDDR2/LPDDR3, value from the JEDEC specification is 500us. Unit: Multiples of 4096 DFI clocks. Present only in designs configured to support mDDR, LPDDR2 or LPDDR3. FOR PERFORMANCE ONLY.
7-5 —	Reserved
4-0 powerdown_to_x32	After this many clocks of the DDRC command channel being idle the DDRC automatically puts the SDRAM into power-down. The DDRC command channel is considered idle when there are no HIF commands outstanding. This must be enabled in the PWRCTL.powerdown_en. Unit: Multiples of 32 DFI clocks FOR PERFORMANCE ONLY.

9.3.3.1.14 Hardware Low Power Control Register (HWLPCTL)

9.3.3.1.14.1 Offset

Register	Offset
HWLPCTL	38h

9.3.3.1.14.2 Diagram



9.3.3.1.14.3 Fields

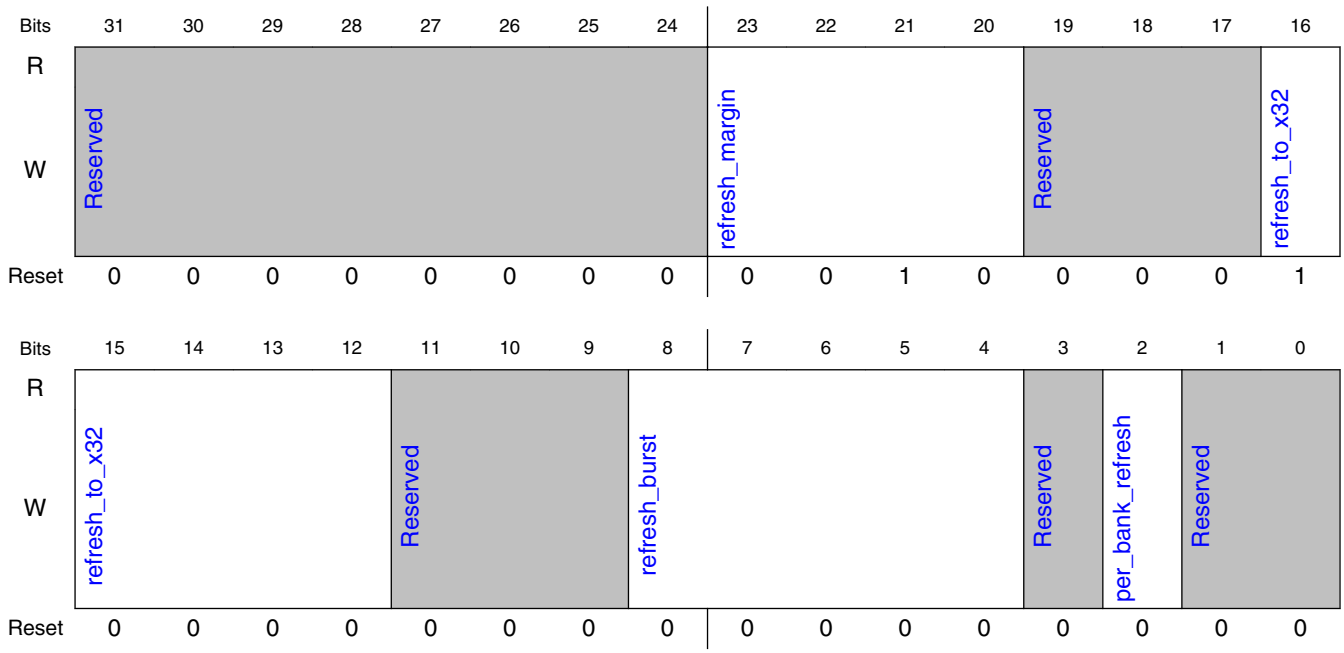
Field	Function
31-28 —	Reserved
27-16 hw_lp_idle_x32	Hardware idle period. The cactive_ddrc output is driven low if the DDRC command channel is idle for hw_lp_idle * 32 cycles if not in INIT or DPD/MPSM operating_mode. The DDRC command channel is considered idle when there are no HIF commands outstanding. The hardware idle function is disabled when hw_lp_idle_x32=0. Unit: Multiples of 32 DFI clocks. FOR PERFORMANCE ONLY.
15-2 —	Reserved
1 hw_lp_exit_idle_en	When this bit is programmed to 1 the cactive_in_ddrc pin of the DDRC can be used to exit from the automatic clock stop, automatic power down or automatic self-refresh modes. Note, it will not cause exit of Self-Refresh that was caused by Hardware Low Power Interface and/or Software (PWRCTL.selfref_sw).
0 hw_lp_en	Enable for Hardware Low Power Interface.

9.3.3.1.15 Refresh Control Register 0 (RFSHCTL0)

9.3.3.1.15.1 Offset

Register	Offset
RFSHCTL0	50h

9.3.3.1.15.2 Diagram



9.3.3.1.15.3 Fields

Field	Function
31-24 —	Reserved
23-20 refresh_margin	Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom_x32. Note that, in LPDDR2/LPDDR3/LPDDR4, internally used t_rfc_nom_x32 may be equal to RFSHTMG.t_rfc_nom_x32>>2 if derating is enabled (DERATEEN.derate_enable=1). Otherwise, internally used t_rfc_nom_x32 will be equal to RFSHTMG.t_rfc_nom_x32. Unit: Multiples of 32 DFI clocks.
19-17 —	Reserved
16-12 refresh_to_x32	If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, but it has not expired (RFSHCTL0.refresh_burst+1) times yet, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful, but before it is absolutely required. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed. Speculative refreshes continues successively until there are no refreshes pending or until new reads or writes are issued to the DDRC. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clocks.
11-9 —	Reserved
8-4 refresh_burst	The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces

Table continues on the next page...

Field	Function
	the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly increases utilization; lower numbers decreases the worst-case latency associated with refreshes. - 0 - single refresh - 1 - burst-of-2 refresh - 7 - burst-of-8 refresh For information on burst refresh feature refer to section 3.9 of DDR2 JEDEC specification - JESD79-2F.pdf. For DDR2/3, the refresh is always per-rank and not per-bank. The rank refresh can be accumulated over 8*tREFI cycles using the burst refresh feature. In DDR4 mode, according to Fine Granularity feature, 8 refreshes can be postponed in 1X mode, 16 refreshes in 2X mode and 32 refreshes in 4X mode. If using PHY-initiated updates, care must be taken in the setting of RFSHCTL0.refresh_burst, to ensure that tRFCmax is not violated due to a PHY-initiated update occurring shortly before a refresh burst was due. In this situation, the refresh burst will be delayed until the PHY-initiated update is complete.
3 —	Reserved
2 per_bank_refresh	Per bank refresh allows traffic to flow to other banks. Per bank refresh is not supported by all LPDDR2 devices but should be supported by all LPDDR3/LPDDR4 devices. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 0b - All bank refresh 1b - Per bank refresh
1-0 —	Reserved

9.3.3.1.16 Refresh Control Register 1 (RFSHCTL1)

9.3.3.1.16.1 Offset

Register	Offset
RFSHCTL1	54h

9.3.3.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				refresh_timer1_start_value_x32											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				refresh_timer0_start_value_x32											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.16.3 Fields

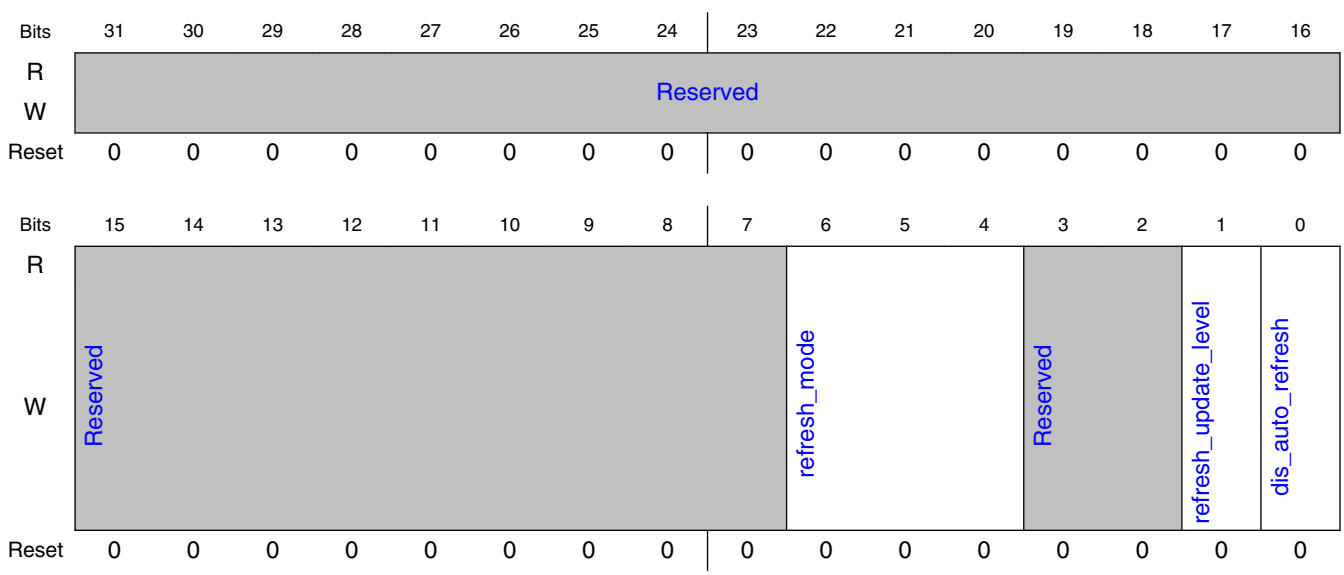
Field	Function
31-28 —	Reserved
27-16 refresh_timer1_start_value_x32	Refresh timer start for rank 1 (only present in multi-rank configurations). This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. This is explained in Refresh Controls section of architecture chapter. Unit: Multiples of 32 DFI clock cycles. FOR PERFORMANCE ONLY.
15-12 —	Reserved
11-0 refresh_timer0_start_value_x32	Refresh timer start for rank 0 (only present in multi-rank configurations). This is useful in staggering the refreshes to multiple ranks to help traffic to proceed. This is explained in Refresh Controls section of architecture chapter. Unit: Multiples of 32 DFI clock cycles. FOR PERFORMANCE ONLY.

9.3.3.1.17 Refresh Control Register 3 (RFSHCTL3)

9.3.3.1.17.1 Offset

Register	Offset
RFSHCTL3	60h

9.3.3.1.17.2 Diagram



9.3.3.1.17.3 Fields

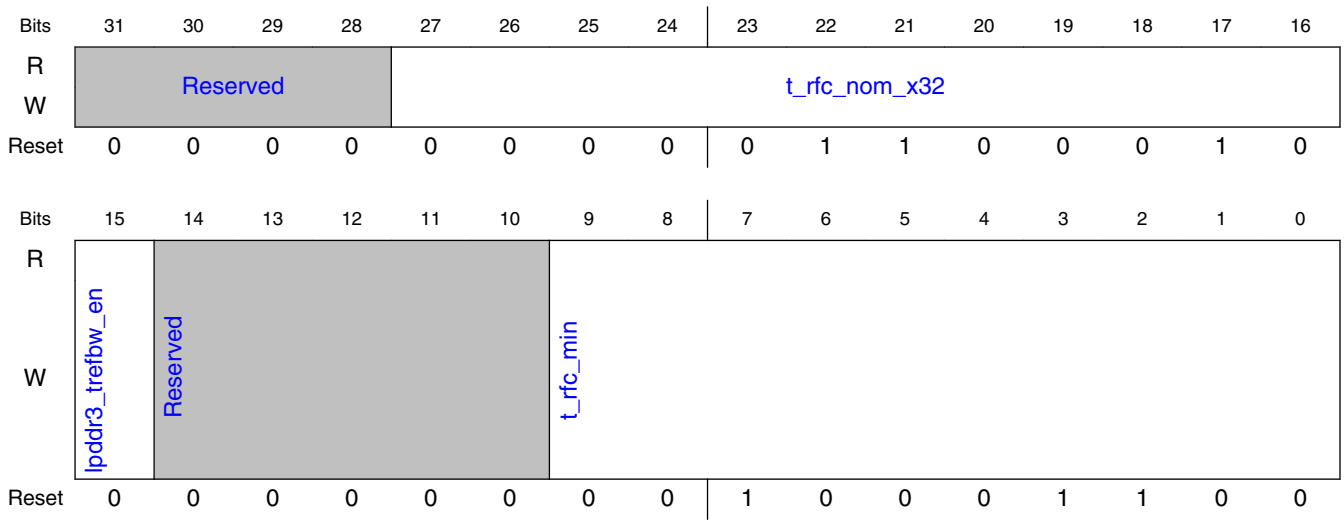
Field	Function
31-7 —	Reserved
6-4 refresh_mode	Fine Granularity Refresh Mode - 000 - Fixed 1x (Normal mode) - 001 - Fixed 2x - 010 - Fixed 4x - 101 - Enable on the fly 2x (not supported) - 110 - Enable on the fly 4x (not supported) - Everything else - reserved Note: Only Fixed 1x mode is supported if RFSHCTL3.dis_auto_refresh = 1. Note: The on-the-fly modes are not supported in this version of the DDRC. Note: This must be set up while the Controller is in reset or while the Controller is in self-refresh mode. Changing this during normal operation is not allowed. Making this a dynamic register will be supported in future version of the DDRC. Note: This register field has effect only if a DDR4 SDRAM device is in use (MSTR.ddr4 = 1).
3-2 —	Reserved
1 refresh_update_level	Toggle this signal (either from 0 to 1 or from 1 to 0) to indicate that the refresh register(s) have been updated. refresh_update_level must not be toggled when the DDRC is in reset (core_ddrc_rstn = 0). The refresh register(s) are automatically updated when exiting reset.
0 dis_auto_refresh	When '1', disable auto-refresh generated by the DDRC. When auto-refresh is disabled, the SoC core must generate refreshes using the registers reg_ddrc_rank0_refresh, reg_ddrc_rank1_refresh, reg_ddrc_rank2_refresh and reg_ddrc_rank3_refresh. When dis_auto_refresh transitions from 0 to 1, any pending refreshes are immediately scheduled by the DDRC. If DDR4 CRC/parity retry is enabled (CRCPARCTL1.crc_parity_retry_enable = 1), disable auto-refresh is not supported, and this bit must be set to '0'. (DDR4 only) If FGR mode is enabled (RFSHCTL3.refresh_mode > 0), disable auto-refresh is not supported, and this bit must be set to '0'. This register field is changeable on the fly.

9.3.3.1.18 Refresh Timing Register (RFSHTMG)

9.3.3.1.18.1 Offset

Register	Offset
RFSHTMG	64h

9.3.3.1.18.2 Diagram



9.3.3.1.18.3 Fields

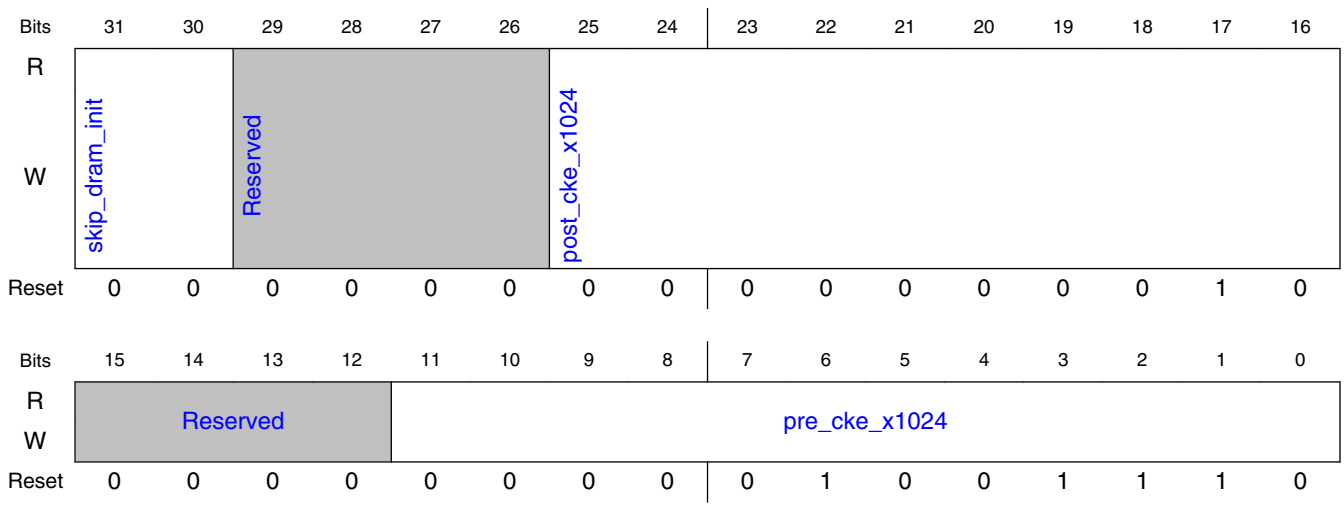
Field	Function
31-28 —	Reserved
27-16 t_rfc_nom_x32	tREFI: Average time interval between refreshes per rank (Specification: 7.8us for DDR2, DDR3 and DDR4. See JEDEC specification for mDDR, LPDDR2, LPDDR3 and LPDDR4). For LPDDR2/LPDDR3/LPDDR4: - if using all-bank refreshes (RFSHCTL0.per_bank_refresh = 0), this register should be set to tREFIab - if using per-bank refreshes (RFSHCTL0.per_bank_refresh = 1), this register should be set to tREFIpb When the controller is operating in 1:2 frequency ratio mode, program this to (tREFI/2), no rounding up. In DDR4 mode, tREFI value is different depending on the refresh mode. The user should program the appropriate value from the spec based on the value programmed in the refresh mode register. Note that RFSHTMG.t_rfc_nom_x32 * 32 must be greater than RFSHTMG.t_rfc_min, and RFSHTMG.t_rfc_nom_x32 must be greater than 0x1. - Non-DDR4 or DDR4 Fixed 1x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0xFFE. - DDR4 Fixed 2x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x7FF. - DDR4 Fixed 4x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x3FF. Unit: Multiples of 32 clocks.
15 lpddr3_trefbw_en	Used only when LPDDR3 memory type is connected. Should only be changed when DDRC is in reset. Specifies whether to use the tREFBW parameter (required by some LPDDR3 devices which comply with earlier versions of the LPDDR3 JEDEC specification) or not: - 0 - tREFBW parameter not used - 1 - tREFBW parameter used
14-10 —	Reserved
9-0 t_rfc_min	tRFC (min): Minimum time from refresh to refresh or activate. When the controller is operating in 1:1 mode, t_rfc_min should be set to RoundUp(tRFCmin/tCK). When the controller is operating in 1:2 mode, t_rfc_min should be set to RoundUp(RoundUp(tRFCmin/tCK)/2). In LPDDR2/LPDDR3/LPDDR4 mode: - if using all-bank refreshes, the tRFCmin value in the above equations is equal to tRFCab - if using per-bank refreshes, the tRFCmin value in the above equations is equal to tRFCpb In DDR4 mode, the tRFCmin value in the above equations is different depending on the refresh mode (fixed 1X,2X,4X) and the device density. The user should program the appropriate value from the spec based on the 'refresh_mode' and the device density that is used. Unit: Clocks.

9.3.3.1.19 SDRAM Initialization Register 0 (INIT0)

9.3.3.1.19.1 Offset

Register	Offset
INIT0	D0h

9.3.3.1.19.2 Diagram



9.3.3.1.19.3 Fields

Field	Function
31-30 skip_dram_init	If lower bit is enabled the SDRAM initialization routine is skipped. The upper bit decides what state the controller starts up in when reset is removed - 00 - SDRAM Intialization routine is run after power-up - 01 - SDRAM Initialization routine is skipped after power-up. Controller starts up in Normal Mode - 11 - SDRAM Initialization routine is skipped after power-up. Controller starts up in Self-refresh Mode - 10 - SDRAM Initialization routine is run after power-up. 00b - SDRAM Initialization routine is run after power-up 01b - SDRAM Initialization routine is skipped after power-up 10b - SDRAM Initialization routine is run after power-up 11b - SDRAM Initialization routine is skipped after power-up
29-26 —	Reserved
25-16 post_cke_x1024	Cycles to wait after driving CKE high to start the SDRAM initialization sequence. Unit: 1024 DFI clock cycles. DDR2 typically requires a 400 ns delay, requiring this value to be programmed to 2 at all clock speeds. LPDDR2/LPDDR3 typically requires this to be programmed for a delay of 200 us. LPDDR4 typically requires this to be programmed for a delay of 2 us. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value.

Table continues on the next page...

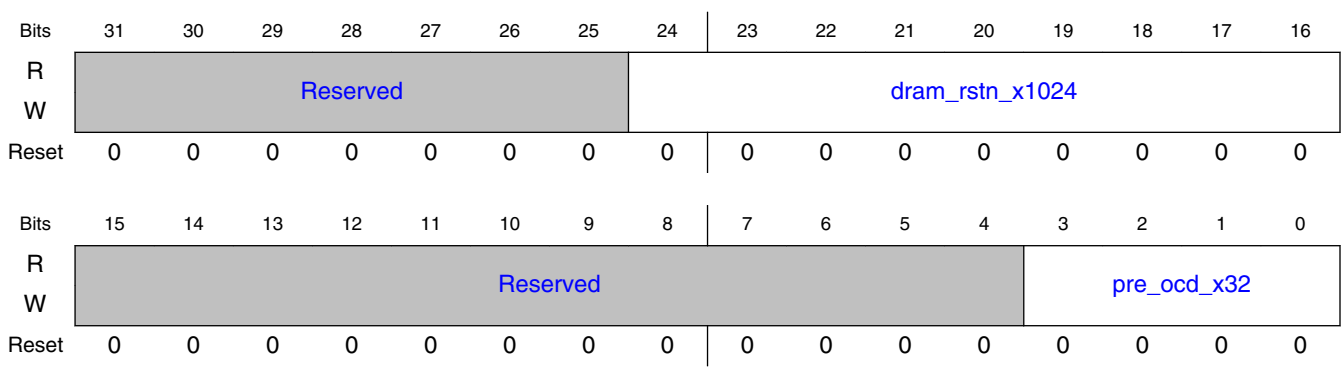
Field	Function
15-12 —	Reserved
11-0 pre_cke_x1024	Cycles to wait after reset before driving CKE high to start the SDRAM initialization sequence. Unit: 1024 DFI clock cycles. DDR2 specifications typically require this to be programmed for a delay of >= 200 us. LPDDR2/LPDDR3: tINIT1 of 100 ns (min) LPDDR4: tINIT3 of 2 ms (min) When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. For DDR3/DDR4 RDIMMs, this should include the time needed to satisfy tSTAB

9.3.3.1.20 SDRAM Initialization Register 1 (INIT1)

9.3.3.1.20.1 Offset

Register	Offset
INIT1	D4h

9.3.3.1.20.2 Diagram



9.3.3.1.20.3 Fields

Field	Function
31-25 —	Reserved
24-16 dram_rstn_x1024	Number of cycles to assert SDRAM reset signal during init sequence. This is only present for designs supporting DDR3, DDR4 or LPDDR4 devices. For use with a DDR PHY, this should be set to a minimum of 1. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: 1024 DFI clock cycles.
15-4 —	Reserved

Table continues on the next page...

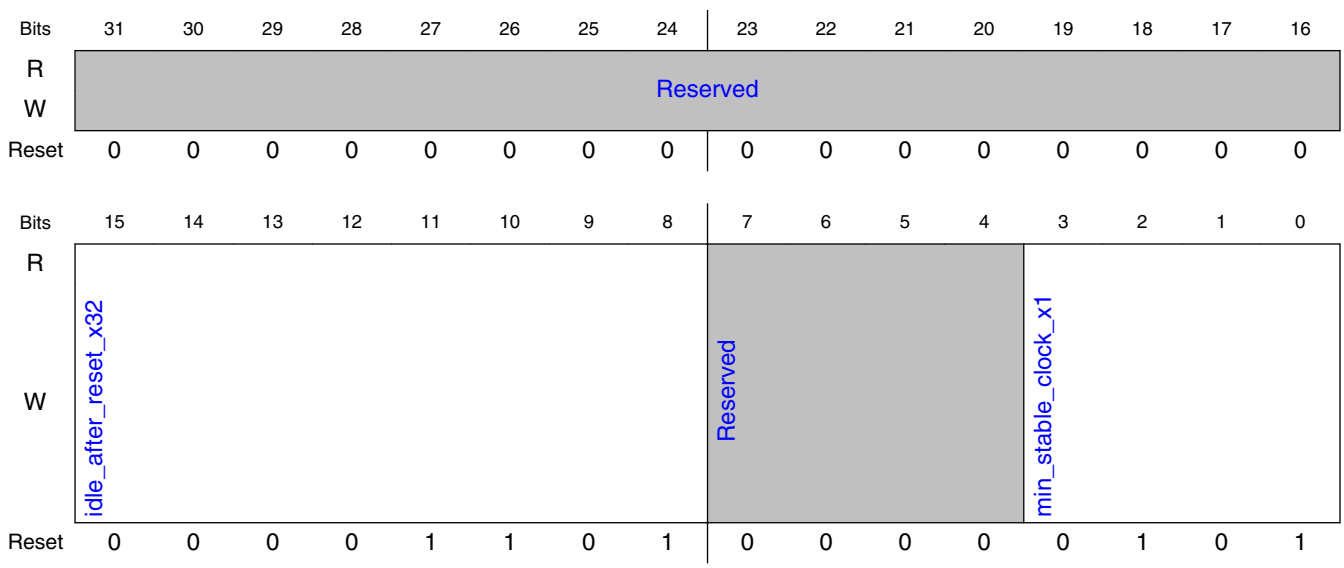
Field	Function
3-0 pre_ocd_x32	Wait period before driving the OCD complete command to SDRAM. Unit: Counts of a global timer that pulses every 32 DFI clock cycles. There is no known specific requirement for this; it may be set to zero.

9.3.3.1.21 SDRAM Initialization Register 2 (INIT2)

9.3.3.1.21.1 Offset

Register	Offset
INIT2	D8h

9.3.3.1.21.2 Diagram



9.3.3.1.21.3 Fields

Field	Function
31-16 —	Reserved
15-8 idle_after_reset_x32	Idle time after the reset command, tINIT4. Present only in designs configured to support LPDDR2. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: 32 DFI clock cycles.
7-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 min_stable_clock_x1	Time to wait after the first CKE high, tINIT2. Present only in designs configured to support LPDDR2/LPDDR3. LPDDR2/LPDDR3 typically requires 5 x tCK delay. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: DFI clock cycles.

9.3.3.1.22 SDRAM Initialization Register 3 (INIT3)

9.3.3.1.22.1 Offset

Register	Offset
INIT3	DCh

9.3.3.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr															
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

9.3.3.1.22.3 Fields

Field	Function
31-16 mr	DDR2: Value to write to MR register. Bit 8 is for DLL and the setting here is ignored. The DDRC sets this bit appropriately. DDR3/DDR4: Value loaded into MR0 register. mDDR: Value to write to MR register. LPDDR2/LPDDR3/LPDDR4 - Value to write to MR1 register
15-0 emr	DDR2: Value to write to EMR register. Bits 9:7 are for OCD and the setting in this register is ignored. The DDRC sets those bits appropriately. DDR3/DDR4: Value to write to MR1 register Set bit 7 to 0. If PHY-evaluation mode training is enabled, this bit is set appropriately by the DDRC during write leveling. mDDR: Value to write to EMR register. LPDDR2/LPDDR3/LPDDR4 - Value to write to MR2 register

9.3.3.1.23 SDRAM Initialization Register 4 (INIT4)

9.3.3.1.23.1 Offset

Register	Offset
INIT4	E0h

9.3.3.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	emr2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.23.3 Fields

Field	Function
31-16 emr2	DDR2: Value to write to EMR2 register. DDR3/DDR4: Value to write to MR2 register LPDDR2/LPDDR3/ LPDDR4: Value to write to MR3 register mDDR: Unused
15-0 emr3	DDR2: Value to write to EMR3 register. DDR3/DDR4: Value to write to MR3 register mDDR/LPDDR2/ LPDDR3: Unused LPDDR4: Value to write to MR13 register

9.3.3.1.24 SDRAM Initialization Register 5 (INIT5)

9.3.3.1.24.1 Offset

Register	Offset
INIT5	E4h

9.3.3.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								dev_zqinit_x32							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						max_auto_init_x1024									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

9.3.3.1.24.3 Fields

Field	Function
31-24 —	Reserved
23-16 dev_zqinit_x32	ZQ initial calibration, tZQINIT. Present only in designs configured to support DDR3 or DDR4 or LPDDR2/LPDDR3. DDR3 typically requires 512 SDRAM clock cycles. DDR4 requires 1024 SDRAM clock cycles. LPDDR2/LPDDR3 requires 1 us. When the controller is operating in 1:2 frequency ratio mode, program this to JEDEC spec value divided by 2, and round it up to the next integer value. Unit: 32 DFI clock cycles.
15-10 —	Reserved
9-0 max_auto_init_x1024	Maximum duration of the auto initialization, tINIT5. Present only in designs configured to support LPDDR2/LPDDR3. LPDDR2/LPDDR3 typically requires 10 us. Unit: 1024 DFI clock cycles.

9.3.3.1.25 SDRAM Initialization Register 6 (INIT6)

9.3.3.1.25.1 Offset

Register	Offset
INIT6	E8h

9.3.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.25.3 Fields

Field	Function
31-16 mr4	DDR4- Value to be loaded into SDRAM MR4 registers. Used in DDR4 designs only.
15-0 mr5	DDR4- Value to be loaded into SDRAM MR5 registers. Used in DDR4 designs only.

9.3.3.1.26 SDRAM Initialization Register 7 (INIT7)

9.3.3.1.26.1 Offset

Register	Offset
INIT7	ECh

9.3.3.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr6															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.26.3 Fields

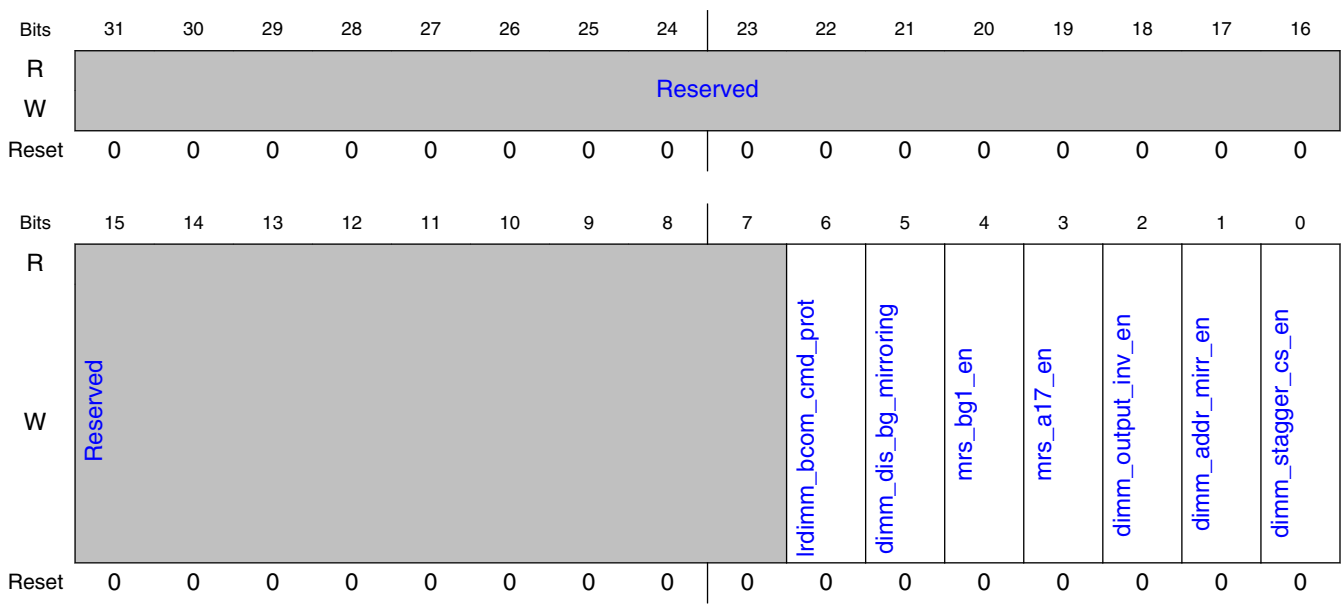
Field	Function
31-16 mr6	DDR4- Value to be loaded into SDRAM MR6 registers. Used in DDR4 designs only.
15-0 —	Reserved

9.3.3.1.27 DIMM Control Register (DIMMCTL)

9.3.3.1.27.1 Offset

Register	Offset
DIMMCTL	F0h

9.3.3.1.27.2 Diagram



9.3.3.1.27.3 Fields

Field	Function
31-7	Reserved

Table continues on the next page...

Field	Function
—	
6 lrdimm_bcom_cmd_prot	Protects the timing restrictions (tBCW/tMRC) between consecutive BCOM commands defined in the Data Buffer specification. When using DDR4 LRDIMM, this bit must be set to 1. Otherwise, this bit must be set to 0.
5 dimm_dis_bg_mirroring	Disabling Address Mirroring for BG bits. When this is set to 1, BG0 and BG1 are NOT swapped even if Address Mirroring is enabled. This will be required for DDR4 DIMMs with x16 devices. 0b - BG0 and BG1 are swapped if address mirroring is enabled. 1b - BG0 and BG1 are NOT swapped.
4 mrs_bg1_en	Enable for BG1 bit of MRS command. BG1 bit of the mode register address is specified as RFU (Reserved for Future Use) and must be programmed to 0 during MRS. In case where DRAMs which do not have BG1 are attached and both the CA parity and the Output Inversion are enabled, this must be set to 0, so that the calculation of CA parity will not include BG1 bit. Note: This has no effect on the address of any other memory accesses, or of software-driven mode register accesses. If address mirroring is enabled, this is applied to BG1 of even ranks and BG0 of odd ranks. 0b - Disabled 1b - Enabled
3 mrs_a17_en	Enable for A17 bit of MRS command. A17 bit of the mode register address is specified as RFU (Reserved for Future Use) and must be programmed to 0 during MRS. In case where DRAMs which do not have A17 are attached and the Output Inversion are enabled, this must be set to 0, so that the calculation of CA parity will not include A17 bit. Note: This has no effect on the address of any other memory accesses, or of software-driven mode register accesses. 0b - Disabled 1b - Enabled
2 dimm_output_inv_en	Output Inversion Enable (for DDR4 RDIMM/LRDIMM implementations only). DDR4 RDIMM/LRDIMM implements the Output Inversion feature by default, which means that the following address, bank address and bank group bits of B-side DRAMs are inverted: A3-A9, A11, A13, A17, BA0-BA1, BG0-BG1. Setting this bit ensures that, for mode register accesses generated by the DDRC during the automatic initialization routine and enabling of a particular DDR4 feature, separate A-side and B-side mode register accesses are generated. For B-side mode register accesses, these bits are inverted within the DDRC to compensate for this RDIMM/LRDIMM inversion. It is recommended to set this bit always, if using DDR4 RDIMMs/LRDIMMs. Note: This has no effect on the address of any other memory accesses, or of software-driven mode register accesses. 0b - Do not implement output inversion for B-side DRAMs. 1b - Implement output inversion for B-side DRAMs.
1 dimm_addr_mirr_en	Address Mirroring Enable (for multi-rank UDIMM implementations and multi-rank DDR4 RDIMM/LRDIMM implementations). Some UDIMMs and DDR4 RDIMMs/LRDIMMs implement address mirroring for odd ranks, which means that the following address, bank address and bank group bits are swapped: (A3, A4), (A5, A6), (A7, A8), (BA0, BA1) and also (A11, A13), (BG0, BG1) for the DDR4. Setting this bit ensures that, for mode register accesses during the automatic initialization routine, these bits are swapped within the DDRC to compensate for this UDIMM/RDIMM/LRDIMM swapping. In addition to the automatic initialization routine, in case of DDR4 UDIMM/RDIMM/LRDIMM, they are swapped during the automatic MRS access to enable/disable of a particular DDR4 feature. Note: This has no effect on the address of any other memory accesses, or of software-driven mode register accesses. This is not supported for mDDR, LPDDR2, LPDDR3 or LPDDR4 SDRAMs. Note: In case of x16 DDR4 DIMMs, BG1 output of MRS for the odd ranks is same as BG0 because BG1 is invalid, hence dimm_dis_bg_mirroring register must be set to 1. 0b - Do not implement address mirroring 1b - For odd ranks, implement address mirroring for MRS commands to during initialization and for any automatic DDR4 MRS commands (to be used if UDIMM/RDIMM/LRDIMM implements address mirroring)
0 dimm_stagger_cs_en	Staggering enable for multi-rank accesses (for multi-rank UDIMM, RDIMM and LRDIMM implementations only). This is not supported for mDDR, LPDDR2, LPDDR3 or LPDDR4 SDRAMs. Even if this bit is set it does not take care of software driven MR commands (via MRCTRL0/MRCTRL1), where software is responsible to send them to separate ranks as appropriate.

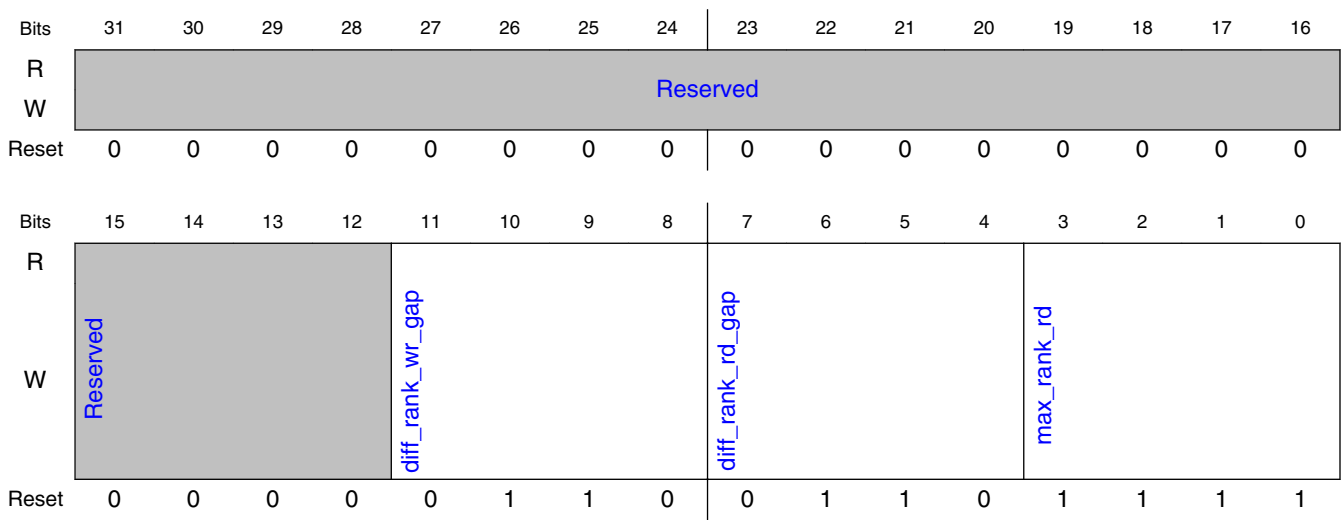
Field	Function
	0b - Do not stagger accesses 1b - (non-DDR4) Send all commands to even and odd ranks separately

9.3.3.1.28 Rank Control Register (RANKCTL)

9.3.3.1.28.1 Offset

Register	Offset
RANKCTL	F4h

9.3.3.1.28.2 Diagram



9.3.3.1.28.3 Fields

Field	Function
31-12 —	Reserved
11-8 diff_rank_wr_gap	Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive writes to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value should consider both PHY requirement and ODT requirement. - PHY requirement: tphy_wrcsgap + 1 (see PHY databook for value of tphy_wrcsgap) If CRC feature is enabled, should be increased by 1. If write preamble is set to 2tCK(DDR4/LPDDR4 only), should be increased by 1. If write postamble is set to 1.5tCK(LPDDR4 only), should be increased by 1. - ODT requirement: The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during writes. For LPDDR4, the requirement is ODTLoff - ODTLon - BL/2 + 1 When the controller is

Table continues on the next page...

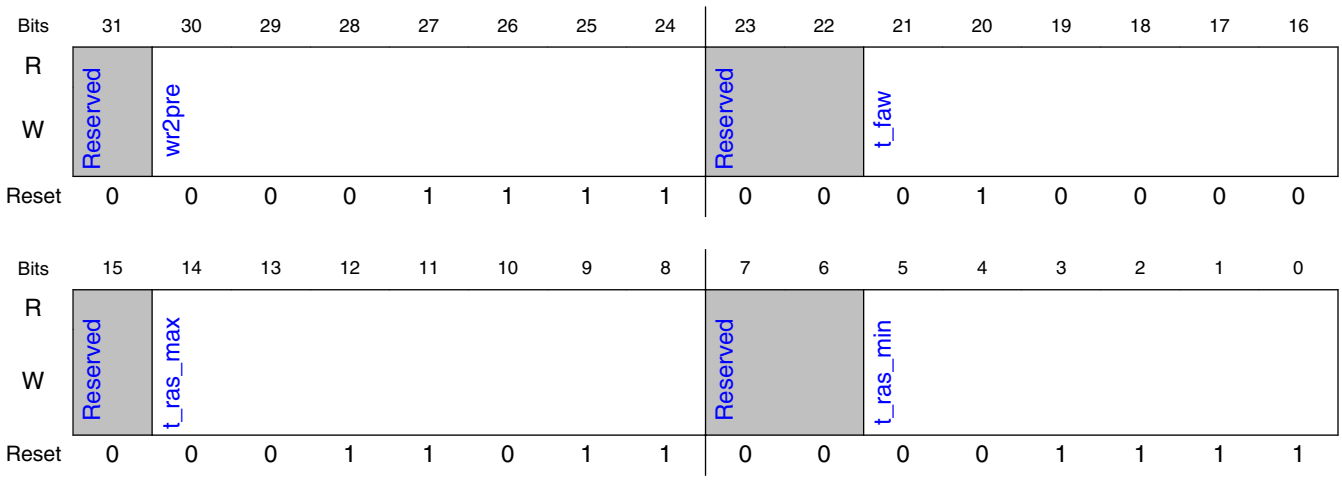
Field	Function
	operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer. Note that, if using DDR4-LRDIMM, refer to TWRWR timing requirements in JEDEC DDR4 Data Buffer (DDR4DB01) Specification.
7-4 diff_rank_rd_gap	Only present for multi-rank configurations. Indicates the number of clocks of gap in data responses when performing consecutive reads to different ranks. This is used to switch the delays in the PHY to match the rank requirements. This value should consider both PHY requirement and ODT requirement. - PHY requirement: tphy_rdcsgap + 1 (see PHY databook for value of tphy_rdcsgap) If read preamble is set to 2tCK(DDR4/LPDDR4 only), should be increased by 1. If read postamble is set to 1.5tCK(LPDDR4 only), should be increased by 1. - ODT requirement: The value programmed in this register takes care of the ODT switch off timing requirement when switching ranks during reads. When the controller is operating in 1:1 mode, program this to the larger of PHY requirement or ODT requirement. When the controller is operating in 1:2 mode, program this to the larger value divided by two and round it up to the next integer. Note that, if using DDR4-LRDIMM, refer to TRDRD timing requirements in JEDEC DDR4 Data Buffer (DDR4DB01) Specification.
3-0 max_rank_rd	Only present for multi-rank configurations. Background: Reads to the same rank can be performed back-to-back. Reads to different ranks require additional gap dictated by the register RANKCTL.diff_rank_rd_gap. This is to avoid possible data bus contention as well as to give PHY enough time to switch the delay when changing ranks. The DDRC arbitrates for bus access on a cycle-by-cycle basis; therefore after a read is scheduled, there are few clock cycles (determined by the value on RANKCTL.diff_rank_rd_gap register) in which only reads from the same rank are eligible to be scheduled. This prevents reads from other ranks from having fair access to the data bus. This parameter represents the maximum number of reads that can be scheduled consecutively to the same rank. After this number is reached, a delay equal to RANKCTL.diff_rank_rd_gap is inserted by the scheduler to allow all ranks a fair opportunity to be scheduled. Higher numbers increase bandwidth utilization, lower numbers increase fairness. This feature can be DISABLED by setting this register to 0. When set to 0, the Controller will stay on the same rank as long as commands are available for it. Minimum programmable value is 0 (feature disabled) and maximum programmable value is 0xF. FOR PERFORMANCE ONLY.

9.3.3.1.29 SDRAM Timing Register 0 (DRAMTMG0)

9.3.3.1.29.1 Offset

Register	Offset
DRAMTMG0	100h

9.3.3.1.29.2 Diagram



9.3.3.1.29.3 Fields

Field	Function
31 —	Reserved
30-24 wr2pre	Minimum time between write and precharge to same bank. Unit: Clocks Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400MHz and less for lower frequencies where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present. - tWR = Write recovery time. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR2/LPDDR3/LPDDR4 for this parameter. When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
23-22 —	Reserved
21-16 t_faw	tFAW Valid only when 8 or more banks(or banks x bank groups) are present. In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles. When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value. In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode. Unit: Clocks
15 —	Reserved
14-8 t_ras_max	tRAS(max): Maximum time between activate and precharge to same bank. This is the maximum time that a page can be kept open Minimum value of this register is 1. Zero is invalid. When the controller is operating in 1:2 frequency ratio mode, program this to (tRAS(max)-1)/2. No rounding up. Unit: Multiples of 1024 clocks.
7-6 —	Reserved
5-0 t_ras_min	tRAS(min): Minimum time between activate and precharge to the same bank. When the controller is operating in 1:2 frequency mode, 1T mode, program this to tRAS(min)/2. No rounding up. When the

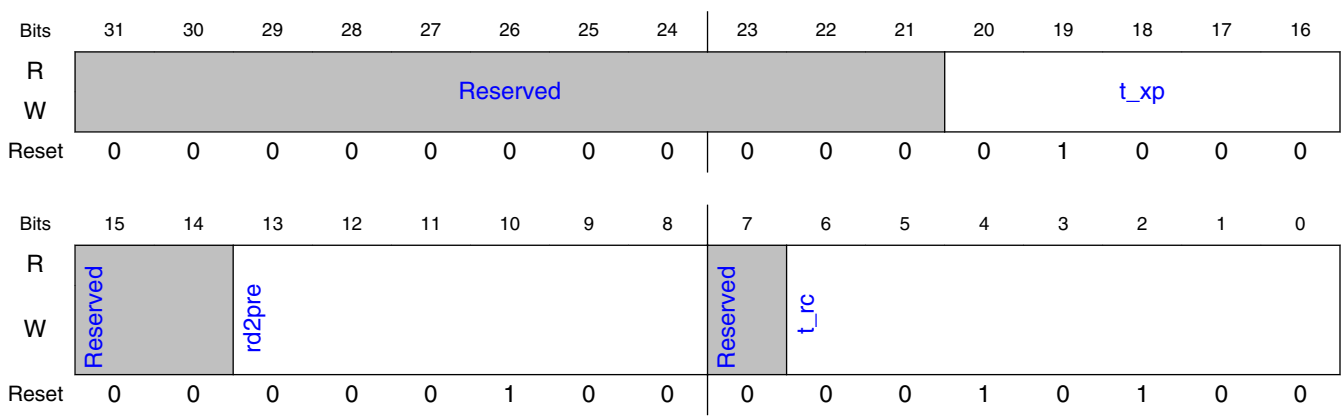
Field	Function
	controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, program this to (tRAS(min)/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.30 SDRAM Timing Register 1 (DRAMTMG1)

9.3.3.1.30.1 Offset

Register	Offset
DRAMTMG1	104h

9.3.3.1.30.2 Diagram



9.3.3.1.30.3 Fields

Field	Function
31-21 —	Reserved
20-16 t_xp	tXP: Minimum time after power-down exit to any operation. For DDR3, this should be programmed to tXPDLL if slow powerdown exit is selected in MR0[12]. If C/A parity for DDR4 is used, set to (tXP+PL) instead. When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value. Units: Clocks
15-14 —	Reserved
13-8 rd2pre	tRTP: Minimum time from read to precharge of same bank. - DDR2: tAL + BL/2 + max(tRTP, 2) - 2 - DDR3: tAL + max (tRTP, 4) - DDR4: Max of following two equations: tAL + max (tRTP, 4) or, RL + BL/2 - tRP (*). - mDDR: BL/2 - LPDDR2: Depends on if it's LPDDR2-S2 or LPDDR2-S4: LPDDR2-S2: BL/2 + tRTP - 1. LPDDR2-S4: BL/2 + max(tRTP,2) - 2. - LPDDR3: BL/2 + max(tRTP,4) - 4 - LPDDR4: BL/2 + max(tRTP,8) - 8 (*) When both DDR4 SDRAM and ST-MRAM are used simultaneously, use SDRAM's

Table continues on the next page...

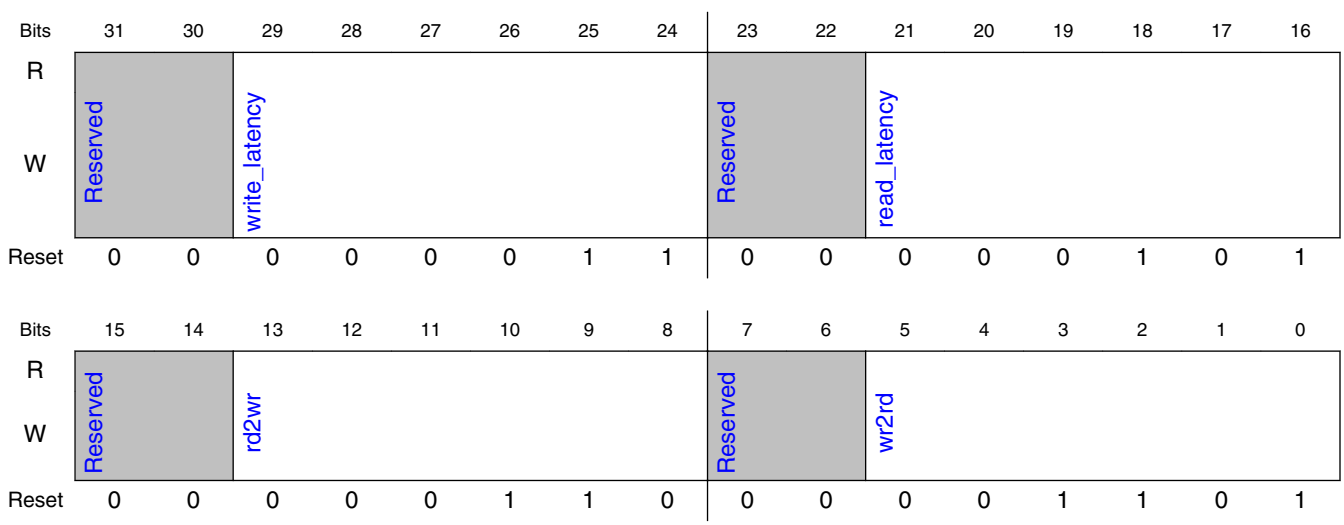
Field	Function
	tRP value for calculation. When the controller is operating in 1:2 mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Unit: Clocks.
7 —	Reserved
6-0 t_rc	tRC: Minimum time between activates to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value. Unit: Clocks.

9.3.3.1.31 SDRAM Timing Register 2 (DRAMTMG2)

9.3.3.1.31.1 Offset

Register	Offset
DRAMTMG2	108h

9.3.3.1.31.2 Diagram



9.3.3.1.31.3 Fields

Field	Function
31-30 —	Reserved

Table continues on the next page...

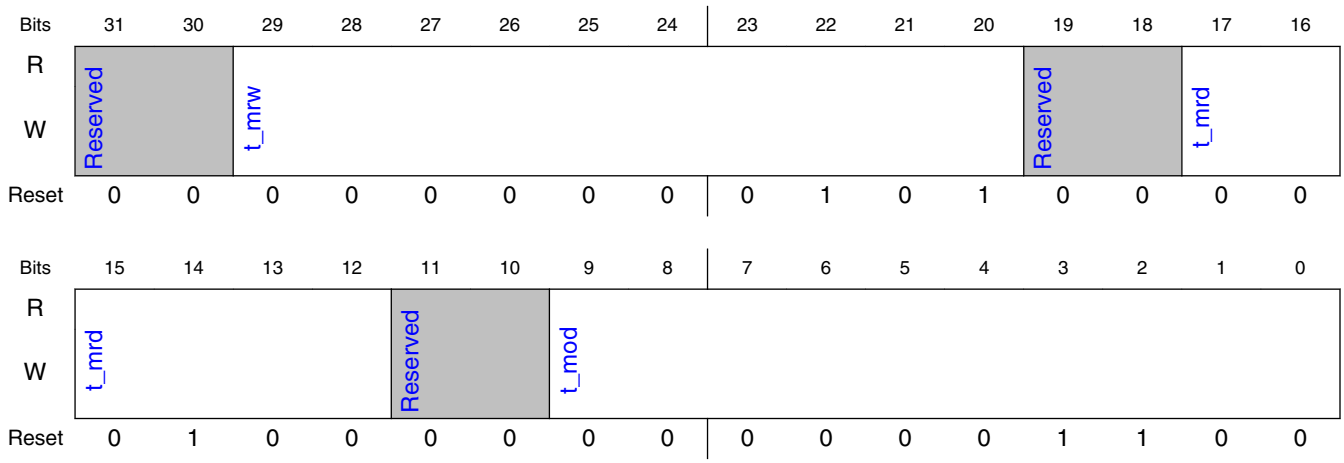
Field	Function
29-24 write_latency	Set to WL Time from write command to write data on SDRAM interface. This must be set to WL. For mDDR, it should normally be set to 1. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of WL to compensate for the extra cycle of latency through the RDIMM/ LRDIMM. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. This register field is not required for DDR2 and DDR3 (except if MEMC_TRAINING is set), as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols Unit: clocks
23-22 —	Reserved
21-16 read_latency	Set to RL Time from read command to read data on SDRAM interface. This must be set to RL. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of RL to compensate for the extra cycle of latency through the RDIMM/LRDIMM. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. This register field is not required for DDR2 and DDR3 (except if MEMC_TRAINING is set), as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols Unit: clocks
15-14 —	Reserved
13-8 rd2wr	DDR2/3/mDDR: RL + BL/2 + 2 - WL DDR4: RL + BL/2 + 1 + WR_PREAMBLE - WL LPDDR2/LPDDR3: RL + BL/2 + RU(tDQSCKmax/tCK) + 1 - WL LPDDR4(DQ ODT is Disabled): RL + BL/2 + RU(tDQSCKmax/tCK) + WR_PREAMBLE + RD_POSTAMBLE - WL LPDDR4(DQ ODT is Enabled) : RL + BL/2 + RU(tDQSCKmax/tCK) + RD_POSTAMBLE - ODTLon - RU(tODTon(min)/tCK) Minimum time from read command to write command. Include time for bus turnaround and all per-bank, per-rank, and global constraints. Please see the relevant PHY databook for details of what should be included here. Unit: Clocks. Where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - RL = read latency = CAS latency - WR_PREAMBLE = write preamble. This is unique to DDR4 and LPDDR4. - RD_POSTAMBLE = read postamble. This is unique to LPDDR4. For LPDDR2/LPDDR3/LPDDR4, if derating is enabled (DERATEEN.derate_enable=1), derated tDQSCKmax should be used. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
7-6 —	Reserved
5-0 wr2rd	DDR4: CWL + PL + BL/2 + tWTR_L Others: CWL + BL/2 + tWTR In DDR4, minimum time from write command to read command for same bank group. In others, minimum time from write command to read command. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_L = internal write to read command delay for same bank group. This comes directly from the SDRAM specification. - tWTR = internal write to read command delay. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR2/LPDDR3/LPDDR4 operation. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.32 SDRAM Timing Register 3 (DRAMTMG3)

9.3.3.1.32.1 Offset

Register	Offset
DRAMTMG3	10Ch

9.3.3.1.32.2 Diagram



9.3.3.1.32.3 Fields

Field	Function
31-30 —	Reserved
29-20 t_mrw	Time to wait after a mode register write or read (MRW or MRR). Present only in designs configured to support LPDDR2, LPDDR3 or LPDDR4. LPDDR2 typically requires value of 5. LPDDR3 typically requires value of 10. LPDDR4: Set this to the larger of tMRW and tMRWCKEL. For LPDDR2, this register is used for the time from a MRW/MRR to all other commands. When the controller is operating in 1:2 frequency ratio mode, program this to the above values divided by 2 and round it up to the next integer value. For LDPDR3, this register is used for the time from a MRW/MRR to a MRW/MRR.
19-18 —	Reserved
17-12 t_mrd	tMRD: Cycles to wait after a mode register write or read. Depending on the connected SDRAM, tMRD represents: DDR2/mDDR: Time from MRS to any command DDR3/4: Time from MRS to MRS command LPDDR2: not used LPDDR3/4: Time from MRS to non-MRS command. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value. If C/A parity for DDR4 is used, set to tMRD_PAR(tMOD+PL) instead.
11-10 —	Reserved
9-0 t_mod	tMOD: Parameter used only in DDR3 and DDR4. Cycles between load mode command and following non-load mode command. If C/A parity for DDR4 is used, set to tMOD_PAR(tMOD+PL) instead. Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode. Note that if using RDIMM/LRDIMM, depending on the

Field	Function
	PHY, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency applied to mode register writes by the RDIMM/LRDIMM chip. Also note that if using LRDIMM, the minimum value of this register is tMRD_L2 if controller is operating in 1:1 frequency ratio mode, or tMRD_L2/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode.

9.3.3.1.33 SDRAM Timing Register 4 (DRAMTMG4)

9.3.3.1.33.1 Offset

Register	Offset
DRAMTMG4	110h

9.3.3.1.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			t_rcd					Reserved				t_ccd			
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_rrd				Reserved			t_rp				
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1

9.3.3.1.33.3 Fields

Field	Function
31-29 —	Reserved
28-24 t_rcd	tRCD - tAL: Minimum time from activate to read or write command to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value. Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode. Unit: Clocks.
23-20 —	Reserved
19-16 t_ccd	DDR4: tCCD_L: This is the minimum time between two reads or two writes for same bank group. Others: tCCD: This is the minimum time between two reads or two writes. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value. Unit: clocks.

Table continues on the next page...

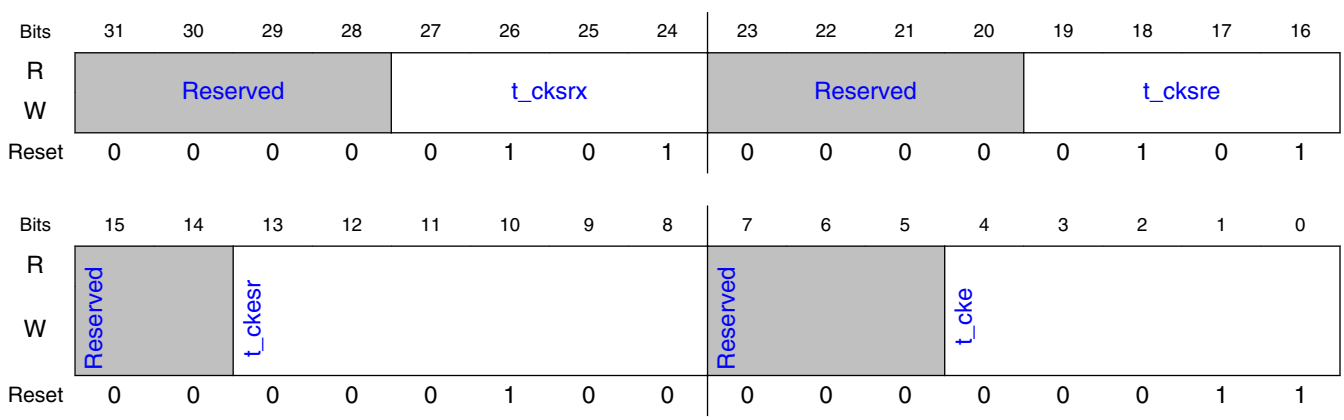
Field	Function
15-12 —	Reserved
11-8 t_rrd	DDR4: tRRD_L: Minimum time between activates from bank "a" to bank "b" for same bank group. Others: tRRD: Minimum time between activates from bank "a" to bank "b" When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_L/2 or tRRD/2) and round it up to the next integer value. Unit: Clocks.
7-5 —	Reserved
4-0 t_rp	tRP: Minimum time from precharge to activate of same bank. When the controller is operating in 1:1 frequency ratio mode, t_rp should be set to RoundUp(tRP/tCK). When the controller is operating in 1:2 frequency ratio mode, t_rp should be set to RoundDown(RoundUp(tRP/tCK)/2) + 1. When the controller is operating in 1:2 frequency ratio mode in LPDDR4, t_rp should be set to RoundUp(RoundUp(tRP/tCK)/2). Unit: Clocks.

9.3.3.1.34 SDRAM Timing Register 5 (DRAMTMG5)

9.3.3.1.34.1 Offset

Register	Offset
DRAMTMG5	114h

9.3.3.1.34.2 Diagram



9.3.3.1.34.3 Fields

Field	Function
31-28	Reserved

Table continues on the next page...

Field	Function
—	
27-24 t_cksrx	This is the time before Self Refresh Exit that CK is maintained as a valid clock before issuing SRX. Specifies the clock stable time before SRX. Recommended settings: - mDDR: 1 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEH - DDR2: 1 - DDR3: tCKSRX - DDR4: tCKSRX When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
23-20 —	Reserved
19-16 t_cksre	This is the time after Self Refresh Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after SRE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEL - DDR2: 1 - DDR3: max (10 ns, 5 tCK) - DDR4: max (10 ns, 5 tCK) (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
15-14 —	Reserved
13-8 t_ckesr	Minimum CKE low width for Self refresh or Self refresh power down entry to exit timing in memory clock cycles. Recommended settings: - mDDR: tRFC - LPDDR2: tCKESR - LPDDR3: tCKESR - LPDDR4: max(tCKELPD, tSR) - DDR2: tCKE - DDR3: tCKE + 1 - DDR4: tCKE + 1 (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
7-5 —	Reserved
4-0 t_cke	Minimum number of cycles of CKE HIGH/LOW during power-down and self refresh. - LPDDR2/LPDDR3 mode: Set this to the larger of tCKE or tCKESR - LPDDR4 mode: Set this to the larger of tCKE, tCKELPD or tSR. - Non-LPDDR2/non-LPDDR3/non-LPDDR4 designs: Set this to tCKE value. When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value. Unit: Clocks.

9.3.3.1.35 SDRAM Timing Register 6 (DRAMTMG6)

9.3.3.1.35.1 Offset

Register	Offset
DRAMTMG6	118h

9.3.3.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				t_ckdpde				Reserved				t_ckdpdx			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												t_ckcsx			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.3.3.1.35.3 Fields

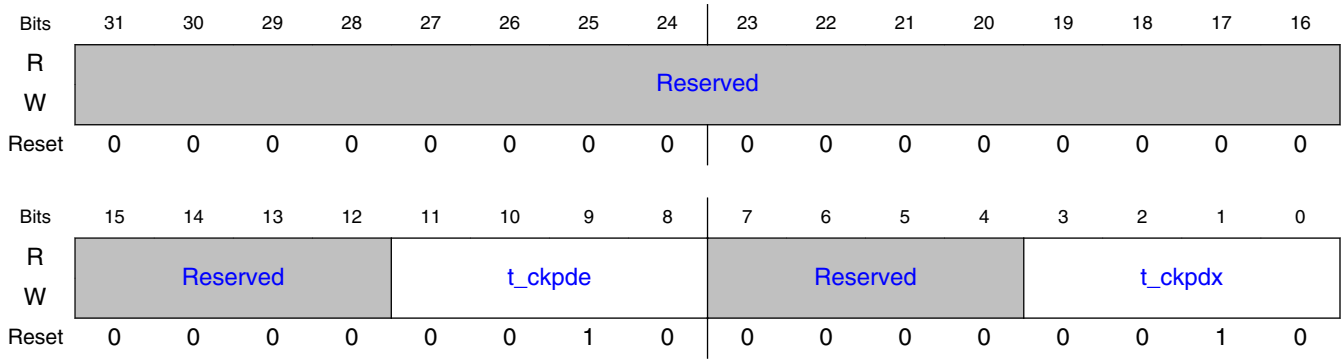
Field	Function
31-28 —	Reserved
27-24 t_ckdpde	This is the time after Deep Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after DPDE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3 devices.
23-20 —	Reserved
19-16 t_ckdpdx	This is the time before Deep Power Down Exit that CK is maintained as a valid clock before issuing DPDX. Specifies the clock stable time before DPDX. Recommended settings: - mDDR: 1 - LPDDR2: 2 - LPDDR3: 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2 devices.
15-4 —	Reserved
3-0 t_ckcsx	This is the time before Clock Stop Exit that CK is maintained as a valid clock before issuing Clock Stop Exit. Specifies the clock stable time before next command after Clock Stop Exit. Recommended settings: - mDDR: 1 - LPDDR2: tXP + 2 - LPDDR3: tXP + 2 - LPDDR4: tXP + 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.36 SDRAM Timing Register 7 (DRAMTMG7)

9.3.3.1.36.1 Offset

Register	Offset
DRAMTMG7	11Ch

9.3.3.1.36.2 Diagram



9.3.3.1.36.3 Fields

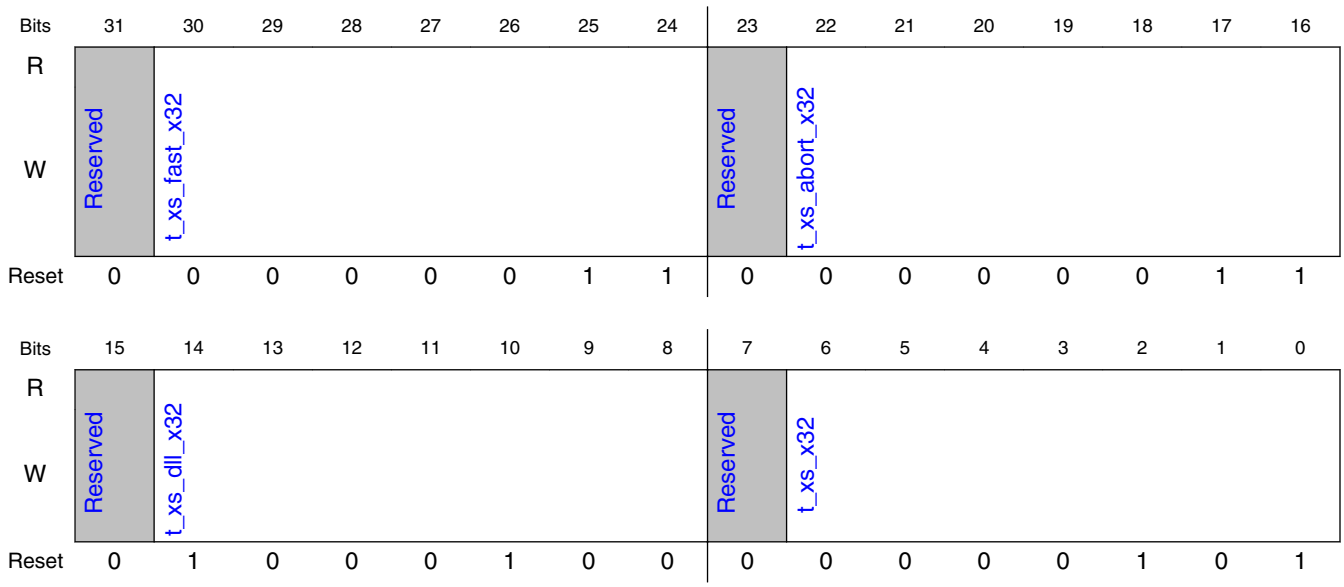
Field	Function
31-12 —	Reserved
11-8 t_ckpde	This is the time after Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after PDE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEL When using DDR2/3/4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksre. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/ LPDDR3/LPDDR4 devices.
7-4 —	Reserved
3-0 t_ckpdx	This is the time before Power Down Exit that CK is maintained as a valid clock before issuing PDX. Specifies the clock stable time before PDX. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: 2 When using DDR2/3/4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksrx. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.37 SDRAM Timing Register 8 (DRAMTMG8)

9.3.3.1.37.1 Offset

Register	Offset
DRAMTMG8	120h

9.3.3.1.37.2 Diagram



9.3.3.1.37.3 Fields

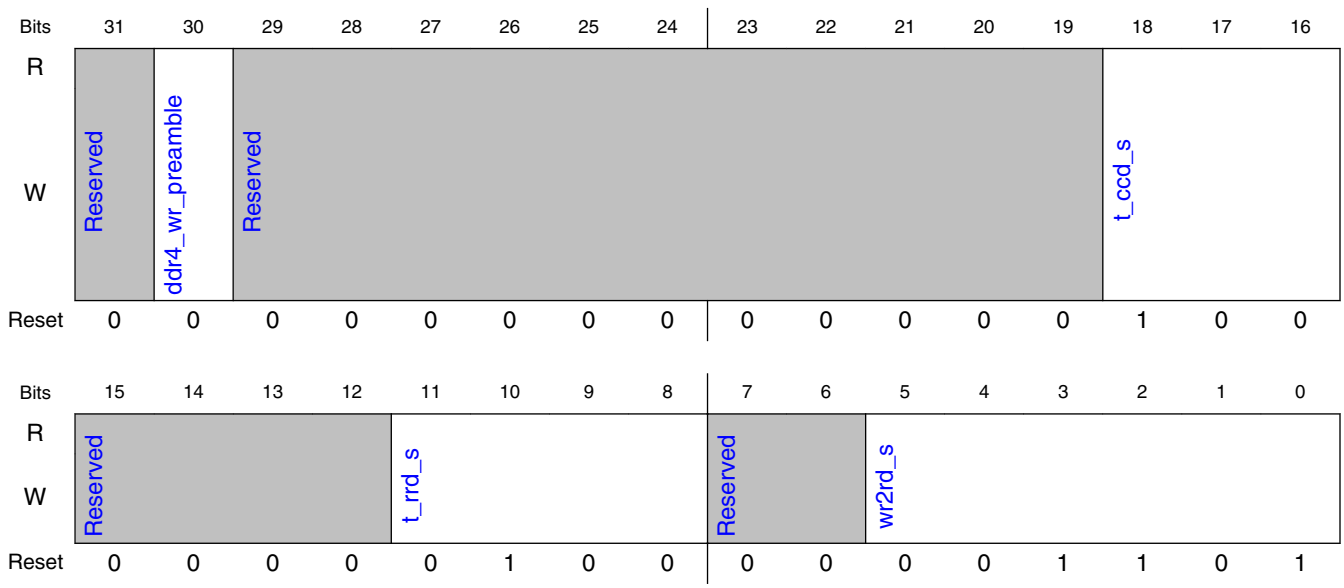
Field	Function
31 —	Reserved
30-24 t_xs_fast_x32	tXS_FAST: Exit Self Refresh to ZQCL, ZQCS and MRS (only CL, WR, RTP and Geardown mode). When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: This is applicable to only ZQCL/ZQCS commands. Note: Ensure this is less than or equal to t_xs_x32.
23 —	Reserved
22-16 t_xs_abort_x32	tXS_ABORT: Exit Self Refresh to commands not requiring a locked DLL in Self Refresh Abort. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Ensure this is less than or equal to t_xs_x32.
15 —	Reserved
14-8 t_xs_dll_x32	tXSDLL: Exit Self Refresh to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR2, DDR3 and DDR4 SDRAMs.
7 —	Reserved
6-0 t_xs_x32	tXS: Exit Self Refresh to commands not requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR2, DDR3 and DDR4 SDRAMs.

9.3.3.1.38 SDRAM Timing Register 9 (DRAMTMG9)

9.3.3.1.38.1 Offset

Register	Offset
DRAMTMG9	124h

9.3.3.1.38.2 Diagram



9.3.3.1.38.3 Fields

Field	Function
31 —	Reserved
30 ddr4_wr_preamble	DDR4 Write preamble mode - 0: 1tCK preamble - 1: 2tCK preamble Present only with MEMC_FREQ_RATIO=2
29-19 —	Reserved
18-16 t_ccd_s	tCCD_S: This is the minimum time between two reads or two writes for different bank group. For bank switching (from bank "a" to bank "b"), the minimum time is this value + 1. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: clocks.
15-12 —	Reserved

Table continues on the next page...

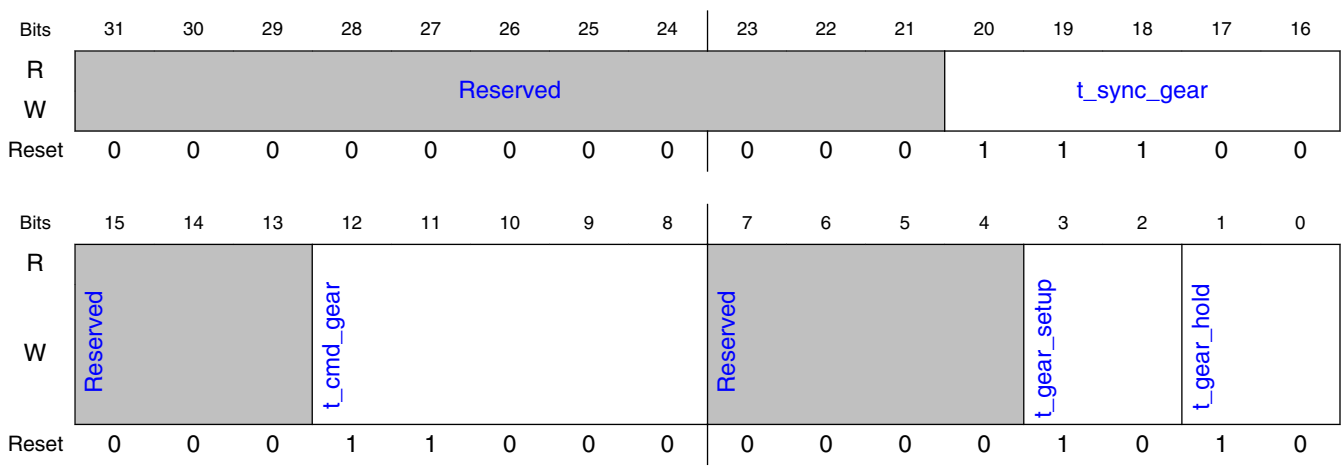
Field	Function
11-8 t_rrd_s	tRRD_S: Minimum time between activates from bank "a" to bank "b" for different bank group. When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-6 —	Reserved
5-0 wr2rd_s	CWL + PL + BL/2 + tWTR_S Minimum time from write command to read command for different bank group. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Present only in designs configured to support DDR4. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_S = internal write to read command delay for different bank group. This comes directly from the SDRAM specification. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.39 SDRAM Timing Register 10 (DRAMTMG10)

9.3.3.1.39.1 Offset

Register	Offset
DRAMTMG10	128h

9.3.3.1.39.2 Diagram



9.3.3.1.39.3 Fields

Field	Function
31-21	Reserved

Table continues on the next page...

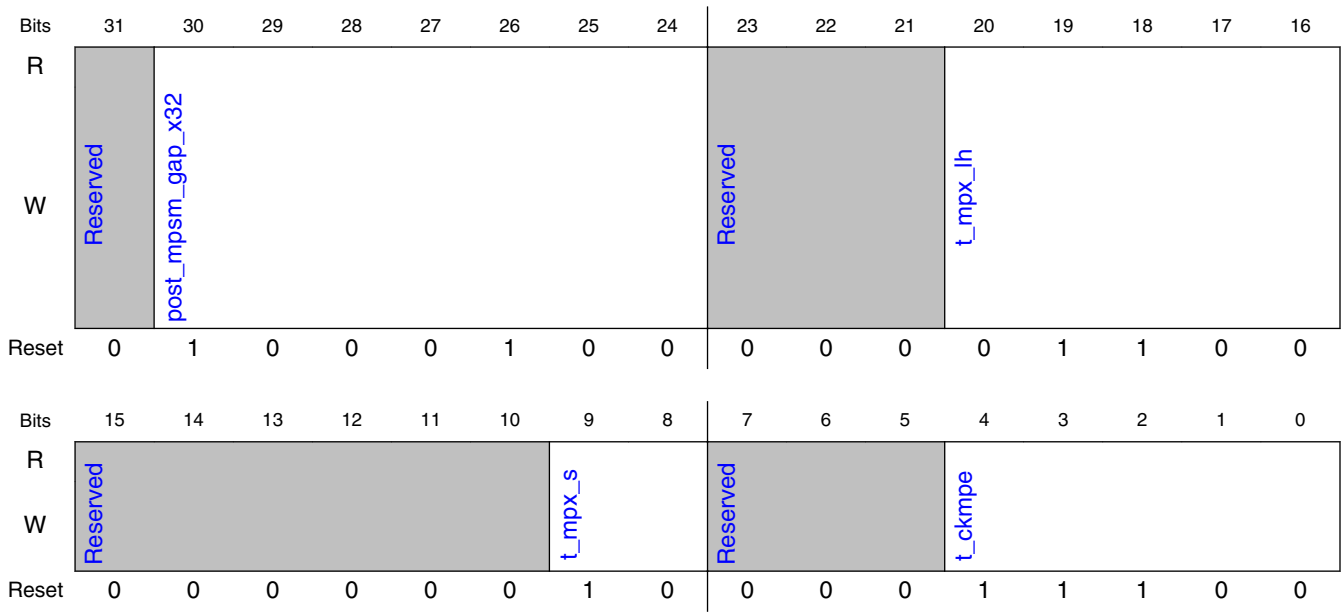
Field	Function
—	
20-16 t_sync_gear	Indicates the time between MRS command and the sync pulse time. This must be even number of clocks. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min)+4nCK tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24+4 = 28 When the controller is operating in 1:2 mode, program this to (tSYNC_GEAR/2) and round it up to the next integer value. Unit: Clocks
15-13 —	Reserved
12-8 t_cmd_gear	Sync pulse to first valid command. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min) tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24 When the controller is operating in 1:2 mode, program this to (tCMD_GEAR/2) and round it up to the next integer value. Unit: Clocks
7-4 —	Reserved
3-2 t_gear_setup	Geardown setup time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_setup/2) and round it up to the next integer value. Unit: Clocks
1-0 t_gear_hold	Geardown hold time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_hold/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.40 SDRAM Timing Register 11 (DRAMTMG11)

9.3.3.1.40.1 Offset

Register	Offset
DRAMTMG11	12Ch

9.3.3.1.40.2 Diagram



9.3.3.1.40.3 Fields

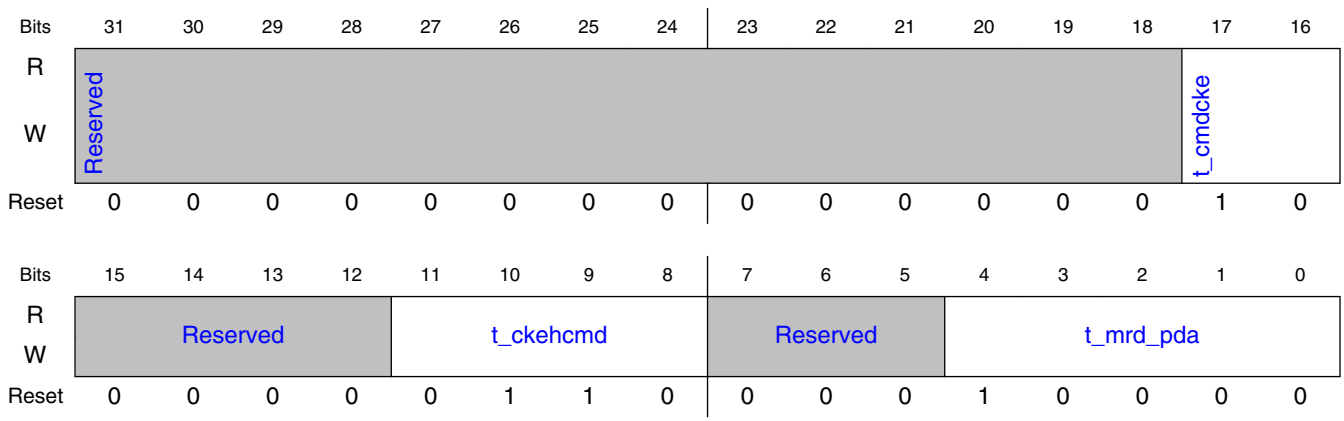
Field	Function
31 —	Reserved
30-24 post_mpsm_gap_x32	tXMPDLL: This is the minimum Exit MPSM to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to (tXMPDLL/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Multiples of 32 clocks.
23-21 —	Reserved
20-16 t_mpx_lh	tMPX_LH: This is the minimum CS_n Low hold time to CKE rising edge. When the controller is operating in 1:2 frequency ratio mode, program this to RoundUp(tMPX_LH/2)+1. Present only in designs configured to support DDR4. Unit: clocks.
15-10 —	Reserved
9-8 t_mpx_s	tMPX_S: Minimum time CS setup time to CKE. When the controller is operating in 1:2 frequency ratio mode, program this to (tMPX_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-5 —	Reserved
4-0 t_ckmpe	tCKMPE: Minimum valid clock requirement after MPSM entry. Present only in designs configured to support DDR4. Unit: Clocks. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.41 SDRAM Timing Register 12 (DRAMTMG12)

9.3.3.1.41.1 Offset

Register	Offset
DRAMTMG12	130h

9.3.3.1.41.2 Diagram



9.3.3.1.41.3 Fields

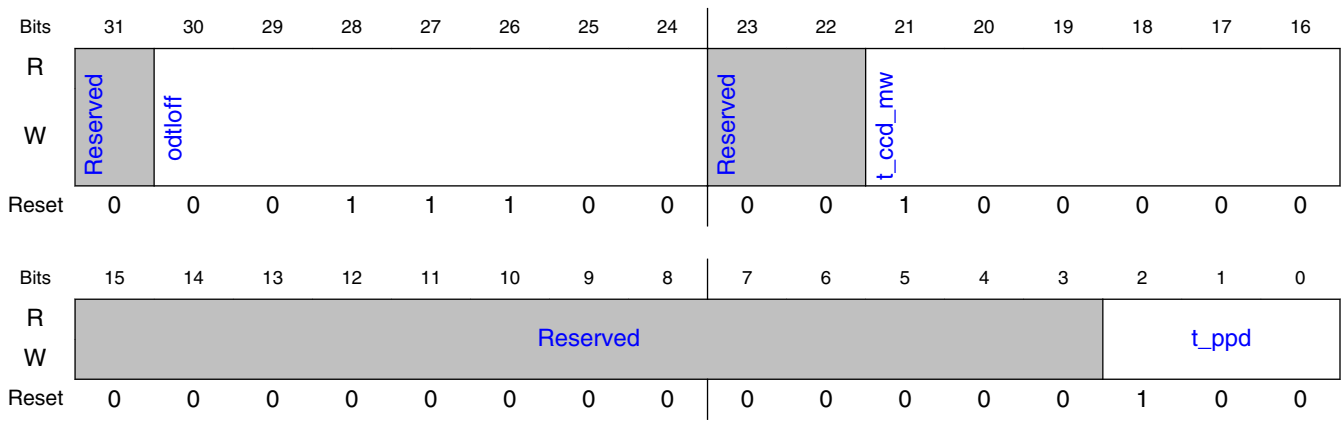
Field	Function
31-18 —	Reserved
17-16 t_cmdcke	tCMDCKE: Delay from valid command to CKE input LOW. Set this to the larger of tESCKE or tCMDCKE When the controller is operating in 1:2 frequency ratio mode, program this to (max(tESCKE, tCMDCKE)/2) and round it up to the next integer value.
15-12 —	Reserved
11-8 t_ckehcmd	tCKEHCMD: Valid command requirement after CKE input HIGH. When the controller is operating in 1:2 frequency ratio mode, program this to (tCKEHCMD/2) and round it up to the next integer value.
7-5 —	Reserved
4-0 t_mrd_pda	tMRD_PDA: This is the Mode Register Set command cycle time in PDA mode. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD_PDA/2) and round it up to the next integer value.

9.3.3.1.42 SDRAM Timing Register 13 (DRAMTMG13)

9.3.3.1.42.1 Offset

Register	Offset
DRAMTMG13	134h

9.3.3.1.42.2 Diagram



9.3.3.1.42.3 Fields

Field	Function
31 —	Reserved
30-24 odtloff	LPDDR4: tODTLoff: This is the latency from CAS-2 command to tODToff reference. When the controller is operating in 1:2 frequency ratio mode, program this to (tODTLoff/2) and round it up to the next integer value. Unit: Clocks.
23-22 —	Reserved
21-16 t_ccd_mw	LPDDR4: tCCDMW: This is the minimum time from write or masked write to masked write command for same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCDMW/2) and round it up to the next integer value. Unit: Clocks.
15-3 —	Reserved
2-0 t_ppd	LPDDR4: tPPD: This is the minimum time from precharge to precharge command. When the controller is operating in 1:2 frequency ratio mode, program this to (tPPD/2) and round it up to the next integer value. Unit: Clocks.

9.3.3.1.43 SDRAM Timing Register 14 (DRAMTMG14)

9.3.3.1.43.1 Offset

Register	Offset
DRAMTMG14	138h

9.3.3.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_xsr											
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

9.3.3.1.43.3 Fields

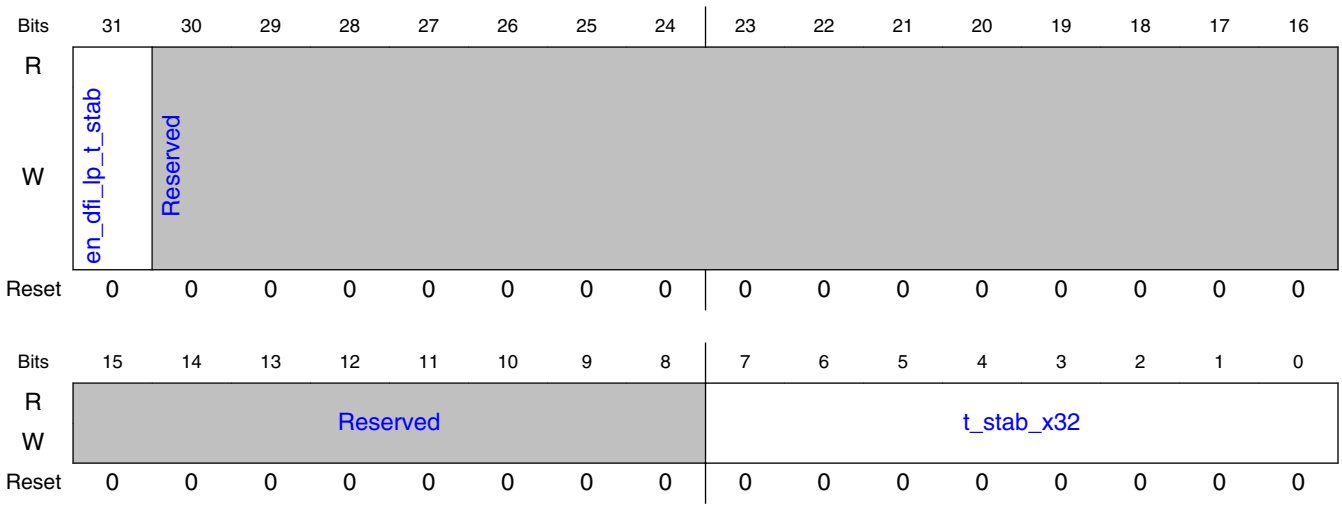
Field	Function
31-12 —	Reserved
11-0 t_xsr	tXSR: Exit Self Refresh to any command. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Note: Used only for mDDR/LPDDR2/LPDDR3/LPDDR4 mode.

9.3.3.1.44 SDRAM Timing Register 15 (DRAMTMG15)

9.3.3.1.44.1 Offset

Register	Offset
DRAMTMG15	13Ch

9.3.3.1.44.2 Diagram



9.3.3.1.44.3 Fields

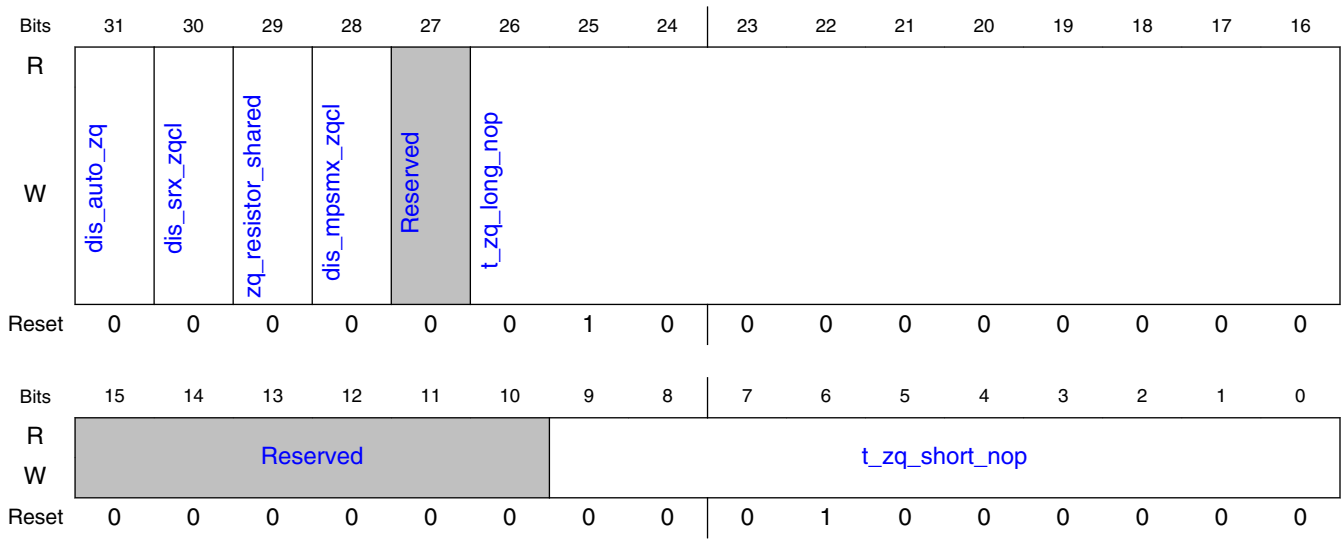
Field	Function
31 en_dfi_lp_t_stab	Enable DFI tSTAB 0b - Disable using tSTAB when exiting DFI LP 1b - Enable using tSTAB when exiting DFI LP. Needs to be set when the PHY is stopping the clock during DFI LP to save maximum power.
30-8 —	Reserved
7-0 t_stab_x32	tSTAB: Stabilization time. It is required in the following two cases for DDR3/DDR4 RDIMM : - when exiting power saving mode, if the clock was stopped, after re-enabling it the clock must be stable for a time specified by tSTAB - in the case of input clock frequency change (DDR4) - after issuing control words that refers to clock timing (Specification: 6us for DDR3, 5us for DDR4) When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: Multiples of 32 clock cycles.

9.3.3.1.45 ZQ Control Register 0 (ZQCTL0)

9.3.3.1.45.1 Offset

Register	Offset
ZQCTL0	180h

9.3.3.1.45.2 Diagram



9.3.3.1.45.3 Fields

Field	Function
31 dis_auto_zq	Disable Auto ZQCS/MPC 0b - Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024. 1b - Disable DDRC generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module.
30 dis_srx_zqcl	Disable ZQCL/MPC 0b - Enable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3 or DDR4 or LPDDR2 or LPDDR3 or LPDDR4 mode. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices. 1b - Disable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3 or DDR4 or LPDDR2 or LPDDR3 or LPDDR4 mode.
29 zq_resistor_shared	ZQ resistor sharing 0b - ZQ resistor is not shared. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices. 1b - Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap.
28 dis_mpsmx_zqcl	Do not issue ZQCL command at Maximum Power Save Mode exit if the DDRC_SHARED_AC configuration parameter is set. Program it to 1'b1. The software can send ZQCS after exiting MPSM mode. 0b - Enable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. This is only present for designs supporting DDR4 devices. 1b - Disable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode.
27 —	Reserved
26-16 t_zq_long_nop	tZQoper for DDR3/DDR4, tZQCL for LPDDR2/LPDDR3, tZQCAL for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCL (ZQ calibration long)/MPC(ZQ Start) command is issued to

Table continues on the next page...

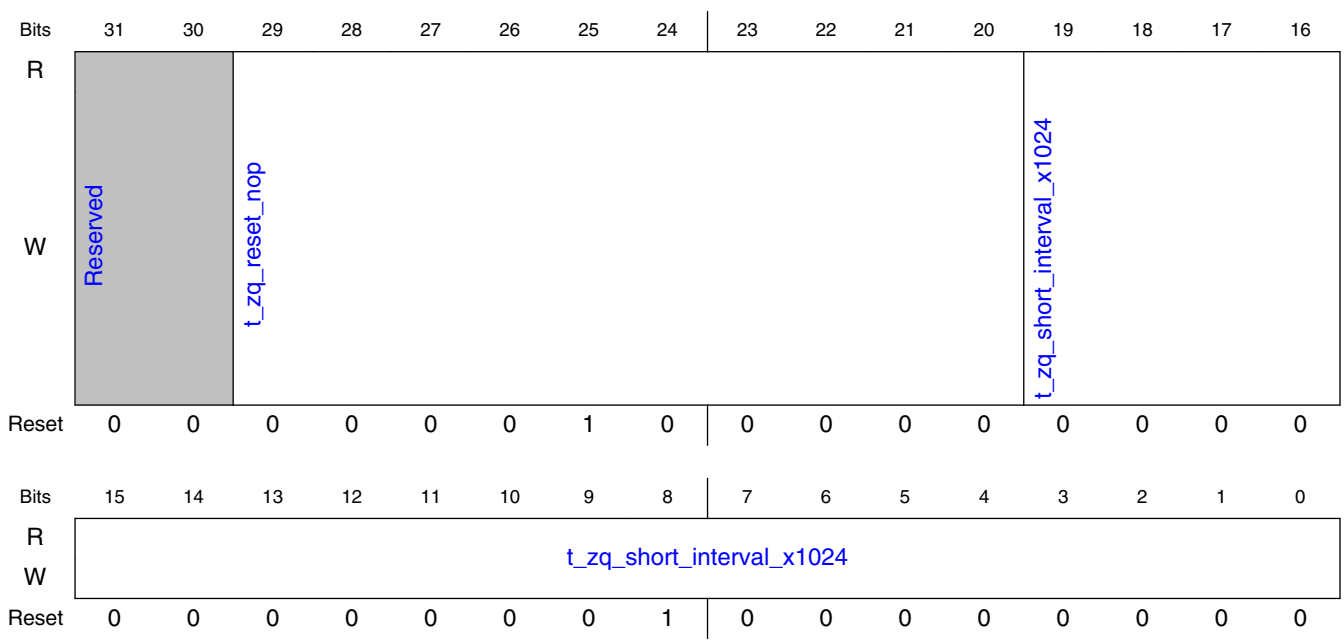
Field	Function
	SDRAM. When the controller is operating in 1:2 frequency ratio mode: DDR3/DDR4: program this to tZQoper/2 and round it up to the next integer value. LPDDR2/LPDDR3: program this to tZQCL/2 and round it up to the next integer value. LPDDR4: program this to tZQCAL/2 and round it up to the next integer value. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.
15-10 —	Reserved
9-0 t_zq_short_nop	tZQCS for DDR3/DD4/LPDDR2/LPDDR3, tZQLAT for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCS (ZQ calibration short)/MPC(ZQ Latch) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.46 ZQ Control Register 1 (ZQCTL1)

9.3.3.1.46.1 Offset

Register	Offset
ZQCTL1	184h

9.3.3.1.46.2 Diagram



9.3.3.1.46.3 Fields

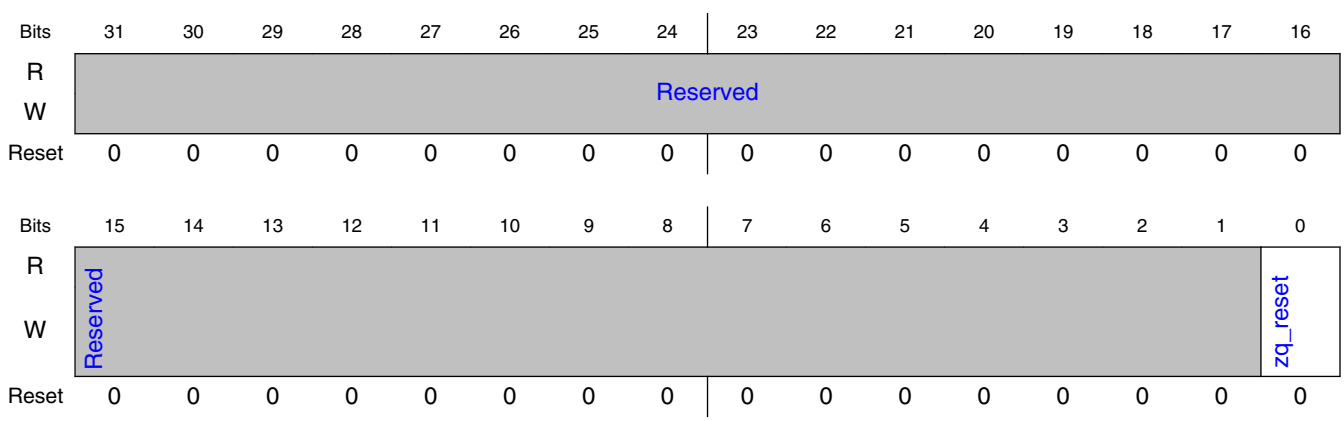
Field	Function
31-30 —	Reserved
29-20 t_zq_reset_nop	tZQReset: Number of DFI clock cycles of NOP required after a ZQReset (ZQ calibration Reset) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQReset/2 and round it up to the next integer value. This is only present for designs supporting LPDDR2/LPDDR3/LPDDR4 devices.
19-0 t_zq_short_interval_x1024	Average interval to wait between automatically issuing ZQCS (ZQ calibration short)/MPC(ZQ calibration) commands to DDR3/DDR4/LPDDR2/LPDDR3/LPDDR4 devices. Meaningless, if ZQCTL0.dis_auto_zq=1. Unit: 1024 DFI clock cycles. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.47 ZQ Control Register 2 (ZQCTL2)

9.3.3.1.47.1 Offset

Register	Offset
ZQCTL2	188h

9.3.3.1.47.2 Diagram



9.3.3.1.47.3 Fields

Field	Function
31-1 —	Reserved

Table continues on the next page...

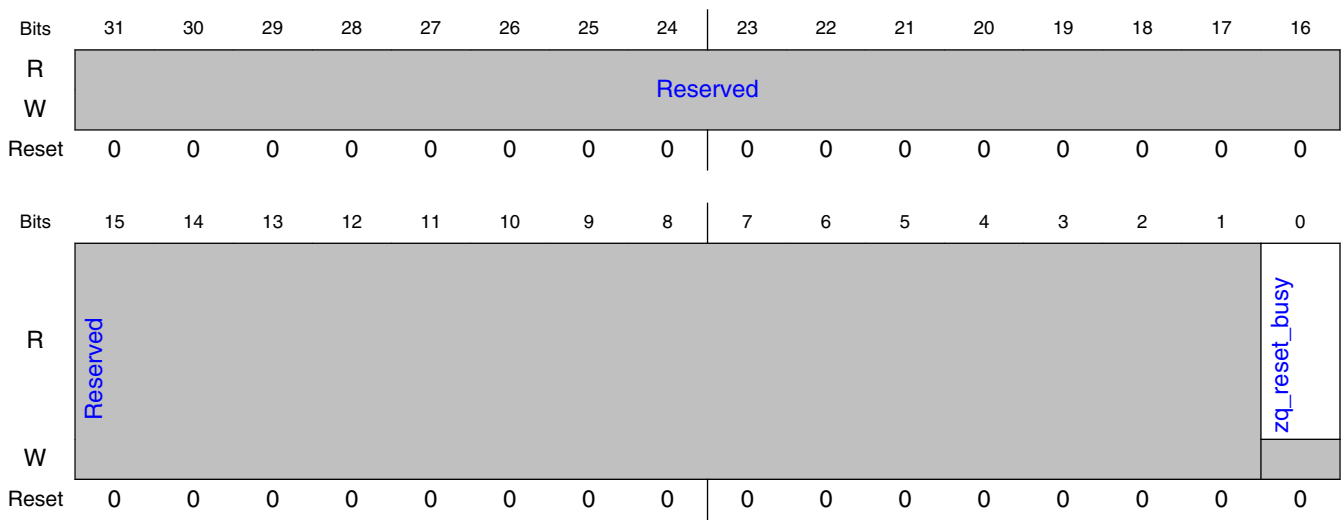
Field	Function
0 zq_reset	Setting this register bit to 1 triggers a ZQ Reset operation. When the ZQ Reset operation is complete, the DDRC automatically clears this bit. It is recommended NOT to set this signal if in Init, Self-Refresh(except LPDDR4) or SR-Powerdown(LPDDR4) or Deep power-down operating modes. This is only present for designs supporting LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.48 ZQ Status Register (ZQSTAT)

9.3.3.1.48.1 Offset

Register	Offset
ZQSTAT	18Ch

9.3.3.1.48.2 Diagram



9.3.3.1.48.3 Fields

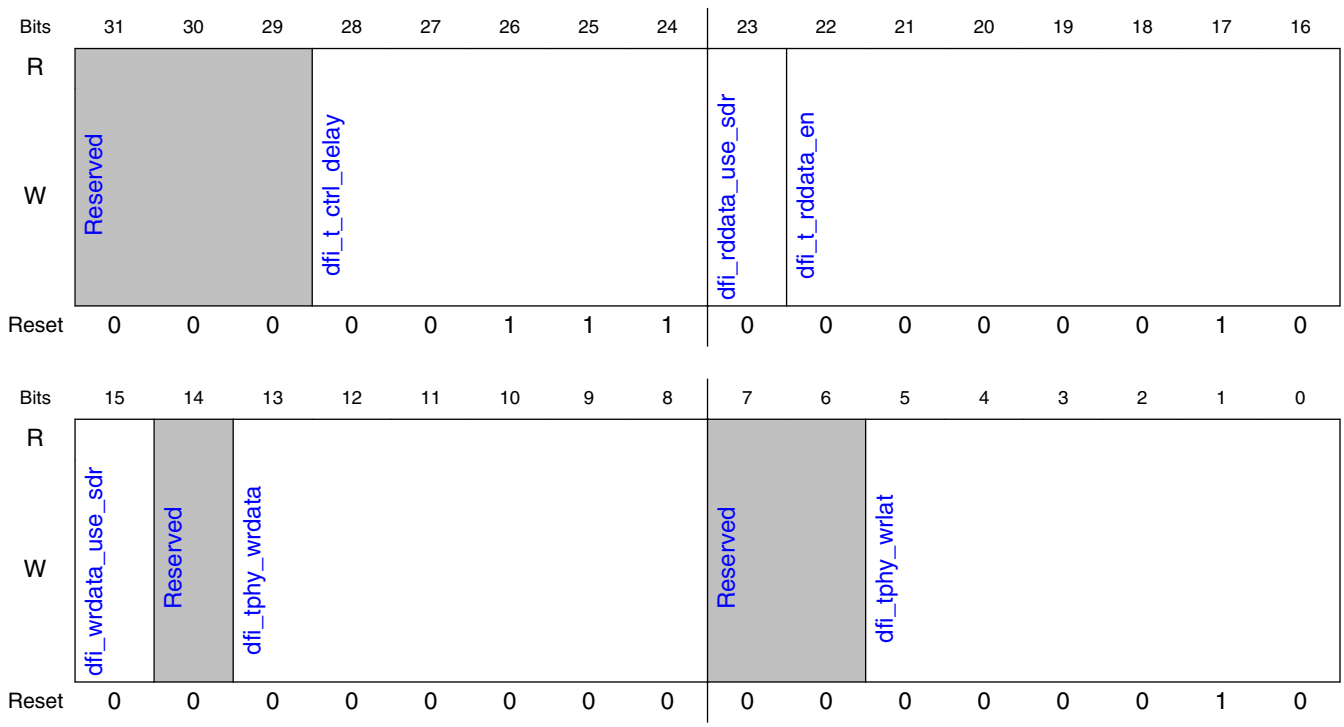
Field	Function
31-1 —	Reserved
0 zq_reset_busy	SoC core may initiate a ZQ Reset operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ZQ Reset request. It goes low when the ZQ Reset command is issued to the SDRAM and the associated NOP period is over. It is recommended not to perform ZQ Reset commands when this signal is high. 0b - Indicates that the SoC core can initiate a ZQ Reset operation 1b - Indicates that ZQ Reset operation is in progress

9.3.3.1.49 DFI Timing Register 0 (DFITMG0)

9.3.3.1.49.1 Offset

Register	Offset
DFITMG0	190h

9.3.3.1.49.2 Diagram



9.3.3.1.49.3 Fields

Field	Function
31-29 —	Reserved
28-24 dfi_t_ctrl_delay	Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value. Note that if using RDIMM/LRDIMM, it is necessary to increment this parameter by RDIMM's/ LRDIMM's extra cycle of latency in terms of DFI clock.
23	Defines whether dfi_rddata_en/dfi_rddata/dfi_rddata_valid is generated using HDR (DFI clock) or SDR (DFI PHY clock) values. Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI

Table continues on the next page...

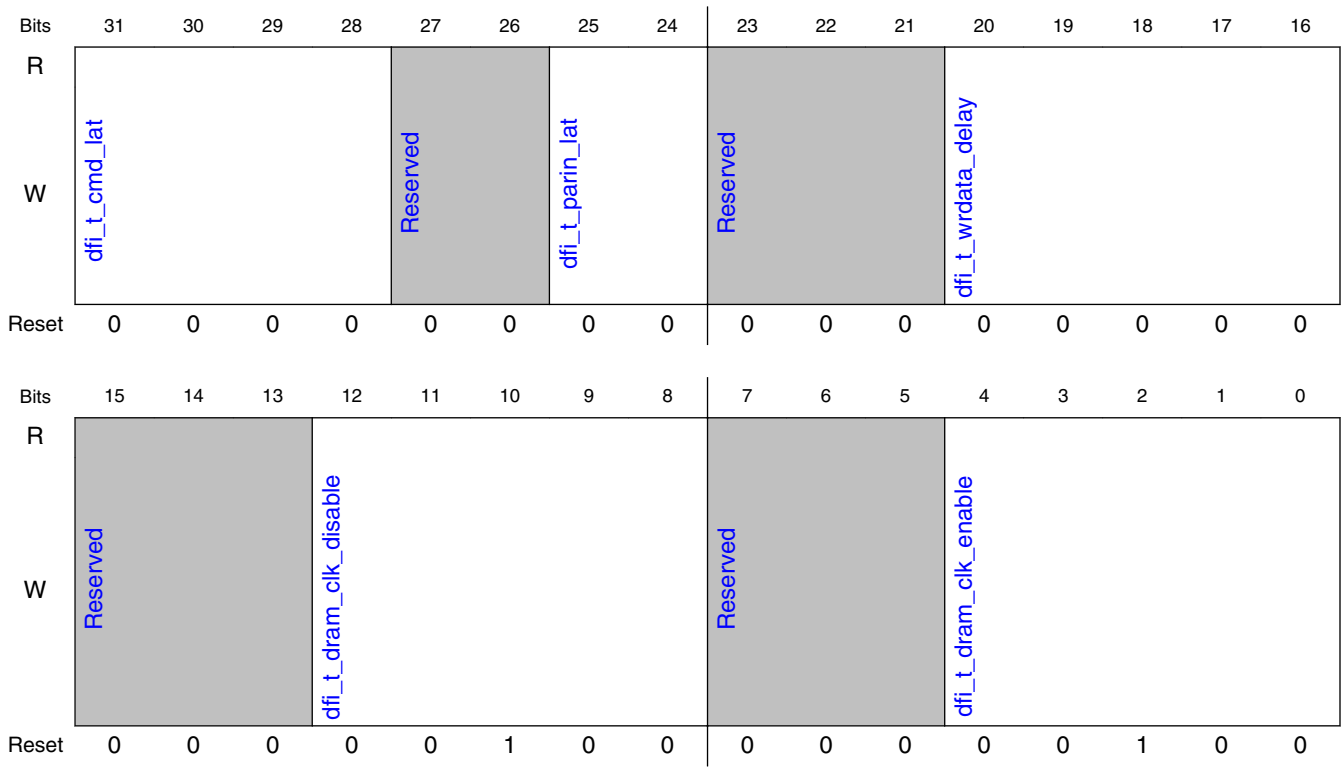
Field	Function
dfi_rddata_use_sdr	clock) or SDR (DFI PHY clock) cycles: - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
22-16 dfi_t_rddata_en	Time from the assertion of a read command on the DFI interface to the assertion of the dfi_rddata_en signal. Refer to PHY specification for correct value. This corresponds to the DFI parameter trddata_en. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of trddata_en. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_sdr.
15 dfi_wrdata_use_sdr	Defines whether dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is generated using HDR (DFI clock) or SDR (DFI PHY clock) values Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
14 —	Reserved
13-8 dfi_tphy_wrdata	Specifies the number of clock cycles between when dfi_wrdata_en is asserted to when the associated write data is driven on the dfi_wrdata signal. This corresponds to the DFI timing parameter tphy_wrdata. Refer to PHY specification for correct value. Note, max supported value is 8. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.
7-6 —	Reserved
5-0 dfi_tphy_wrlat	Write latency Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat. Refer to PHY specification for correct value.Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of tphy_wrlat. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.

9.3.3.1.50 DFI Timing Register 1 (DFITMG1)

9.3.3.1.50.1 Offset

Register	Offset
DFITMG1	194h

9.3.3.1.50.2 Diagram



9.3.3.1.50.3 Fields

Field	Function
31-28 dfi_t_cmd_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated command is driven. This field is used for CAL mode, should be set to '0' or the value which matches the CAL mode register setting in the DRAM. If the PHY can add the latency for CAL mode, this should be set to '0'. Valid Range: 0, 3, 4, 5, 6, and 8
27-26 —	Reserved
25-24 dfi_t_parin_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated dfi_parity_in signal is driven.
23-21 —	Reserved
20-16 dfi_t_wrdata_delay	Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted and when the corresponding write data transfer is completed on the DRAM bus. This corresponds to the DFI timing parameter twrdata_delay. Refer to PHY specification for correct value. For DFI 3.0 PHY, set to twrdata_delay, a new timing parameter introduced in DFI 3.0. For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM). Value to be programmed is in terms of DFI clocks, not PHY clocks. In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer. If using DFITMG0.dfi_wrdata_use_sdr=1, add 1 to the value. Unit: Clocks
15-13 —	Reserved

Table continues on the next page...

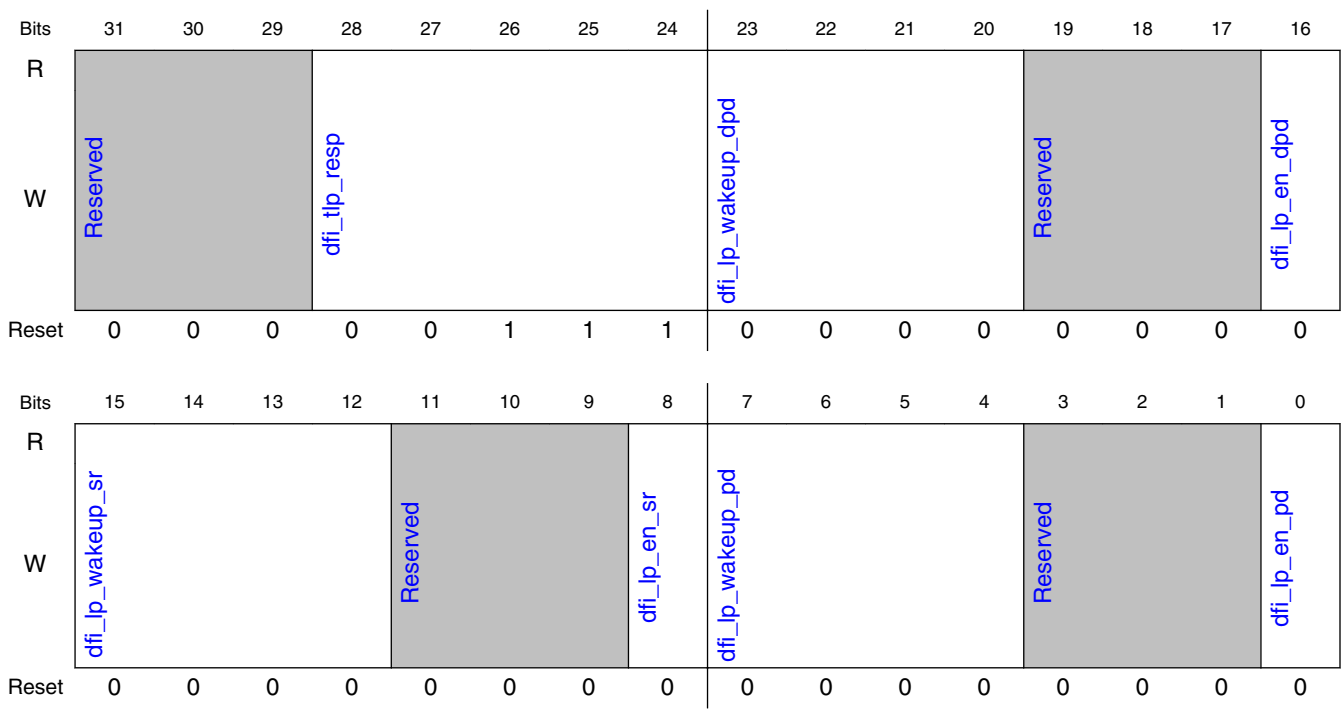
Field	Function
12-8 dfi_t_dram_clk_disable	Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
7-5 —	Reserved
4-0 dfi_t_dram_clk_enable	Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

9.3.3.1.51 DFI Low Power Configuration Register 0 (DFILPCFG0)

9.3.3.1.51.1 Offset

Register	Offset
DFILPCFG0	198h

9.3.3.1.51.2 Diagram



9.3.3.1.51.3 Fields

Field	Function
31-29 —	Reserved
28-24 dfi_tlp_resp	Setting in DFI clock cycles for DFI's tlp_resp time. Same value is used for both Power Down, Self Refresh, Deep Power Down and Maximum Power Saving modes. DFI 2.1 specification onwards, recommends using a fixed value of 7 always.
23-20 dfi_lp_wakeup_dpd	Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Deep Power Down mode is entered. Determines the DFI's tlp_wakeup time: This is only present for designs supporting mDDR or LPDDR2/ LPDDR3 devices. 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
19-17 —	Reserved
16 dfi_lp_en_dpd	Enables DFI Low Power interface handshaking during Deep Power Down Entry/Exit. - 0 - Disabled - 1 - Enabled This is only present for designs supporting mDDR or LPDDR2/LPDDR3 devices.
15-12 dfi_lp_wakeup_sr	Value in DFI clpck cycles to drive on dfi_lp_wakeup signal when Self Refresh mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
11-9 —	Reserved
8	Enables DFI Low Power interface handshaking during Self Refresh Entry/Exit. - 0 - Disabled - 1 - Enabled 0b - Disabled

Table continues on the next page...

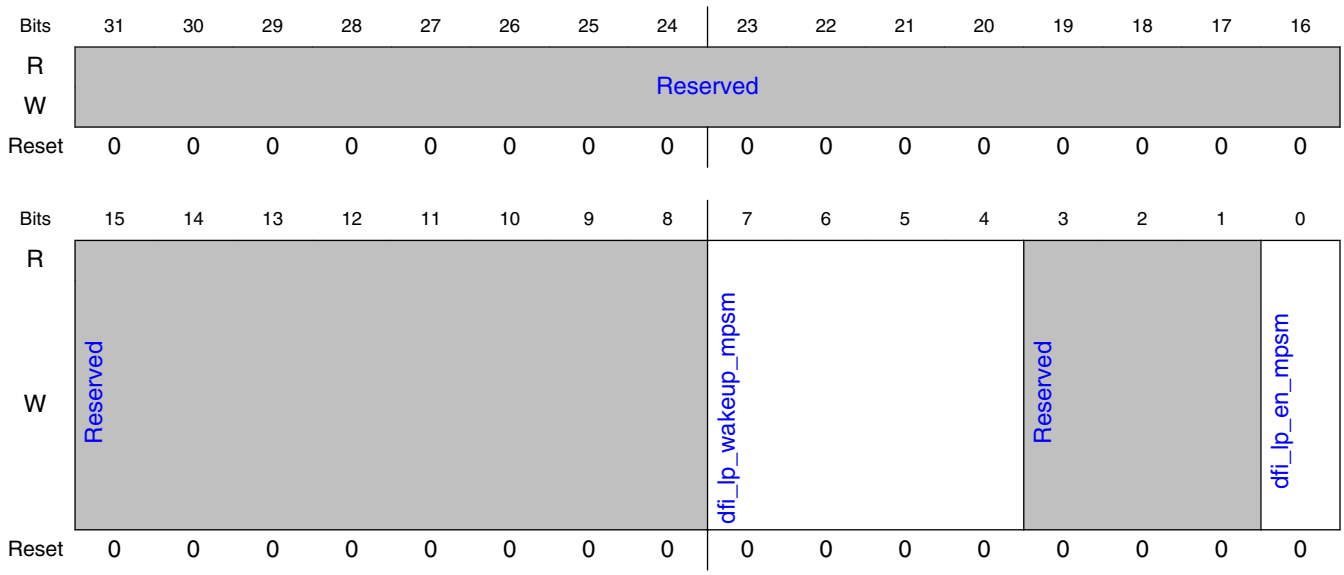
Field	Function
dfi_lp_en_sr	1b - Enabled
7-4 dfi_lp_wakeup_pd	Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Power Down mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
3-1 —	Reserved
0 dfi_lp_en_pd	Enables DFI Low Power interface handshaking during Power Down Entry/Exit. - 0 - Disabled - 1 - Enabled

9.3.3.1.52 DFI Low Power Configuration Register 1 (DFILPCFG1)

9.3.3.1.52.1 Offset

Register	Offset
DFILPCFG1	19Ch

9.3.3.1.52.2 Diagram



9.3.3.1.52.3 Fields

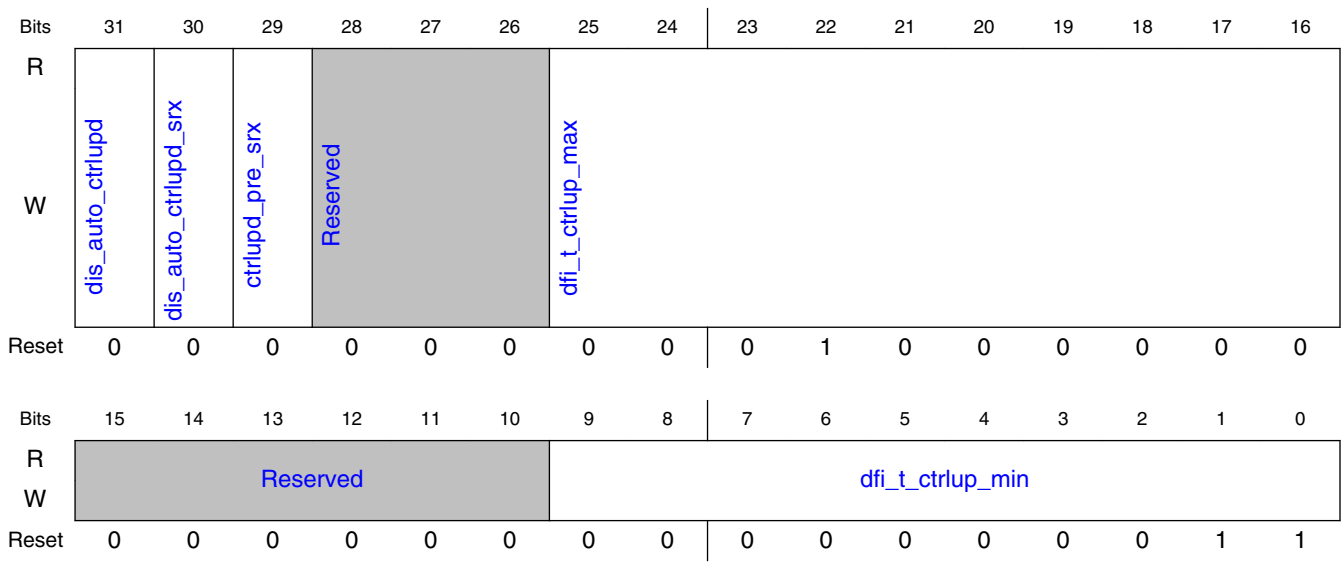
Field	Function
31-8 —	Reserved
7-4 dfi_lp_wakeup_mpsm	Value in DFI clock cycles to drive on dfi_lp_wakeup signal when Maximum Power Saving Mode is entered. Determines the DFI's tlp_wakeup time: 0000b - 16 cycles 0001b - 32 cycles 0010b - 64 cycles 0011b - 128 cycles 0100b - 256 cycles 0101b - 512 cycles 0110b - 1024 cycles 0111b - 2048 cycles 1000b - 4096 cycles 1001b - 8192 cycles 1010b - 16384 cycles 1011b - 32768 cycles 1100b - 65536 cycles 1101b - 131072 cycles 1110b - 262144 cycles 1111b - Unlimited cycles
3-1 —	Reserved
0 dfi_lp_en_mpsm	Enables DFI Low Power interface handshaking during Maximum Power Saving Mode Entry/Exit. - 0 - Disabled - 1 - Enabled This is only present for designs supporting DDR4 devices.

9.3.3.1.53 DFI Update Register 0 (DFIUPD0)

9.3.3.1.53.1 Offset

Register	Offset
DFIUPD0	1A0h

9.3.3.1.53.2 Diagram



9.3.3.1.53.3 Fields

Field	Function
31 dis_auto_ctrlupd	automatic dfi_ctrlupd_req generation by the DDRC 0b - DDRC issues dfi_ctrlupd_req periodically. 1b - disable the automatic dfi_ctrlupd_req generation by the DDRC. The core must issue the dfi_ctrlupd_req signal using register reg_ddrc_ctrlupd.
30 dis_auto_ctrlupd_srx	Auto ctrlupd request generation 0b - DDRC issues a dfi_ctrlupd_req before or after exiting self-refresh, depending on DFIUPD0.ctrlupd_pre_srx. 1b - disable the automatic dfi_ctrlupd_req generation by the DDRC at self-refresh exit.
29 ctrlupd_pre_srx	Selects dfi_ctrlupd_req requirements at SRX: - 0 : send ctrlupd after SRX - 1 : send ctrlupd before SRX If DFIUPD0.dis_auto_ctrlupd_srx=1, this register has no impact, because no dfi_ctrlupd_req will be issued when SRX. 0b - send ctrlupd after SRX 1b - send ctrlupd before SRX
28-26 —	Reserved

Table continues on the next page...

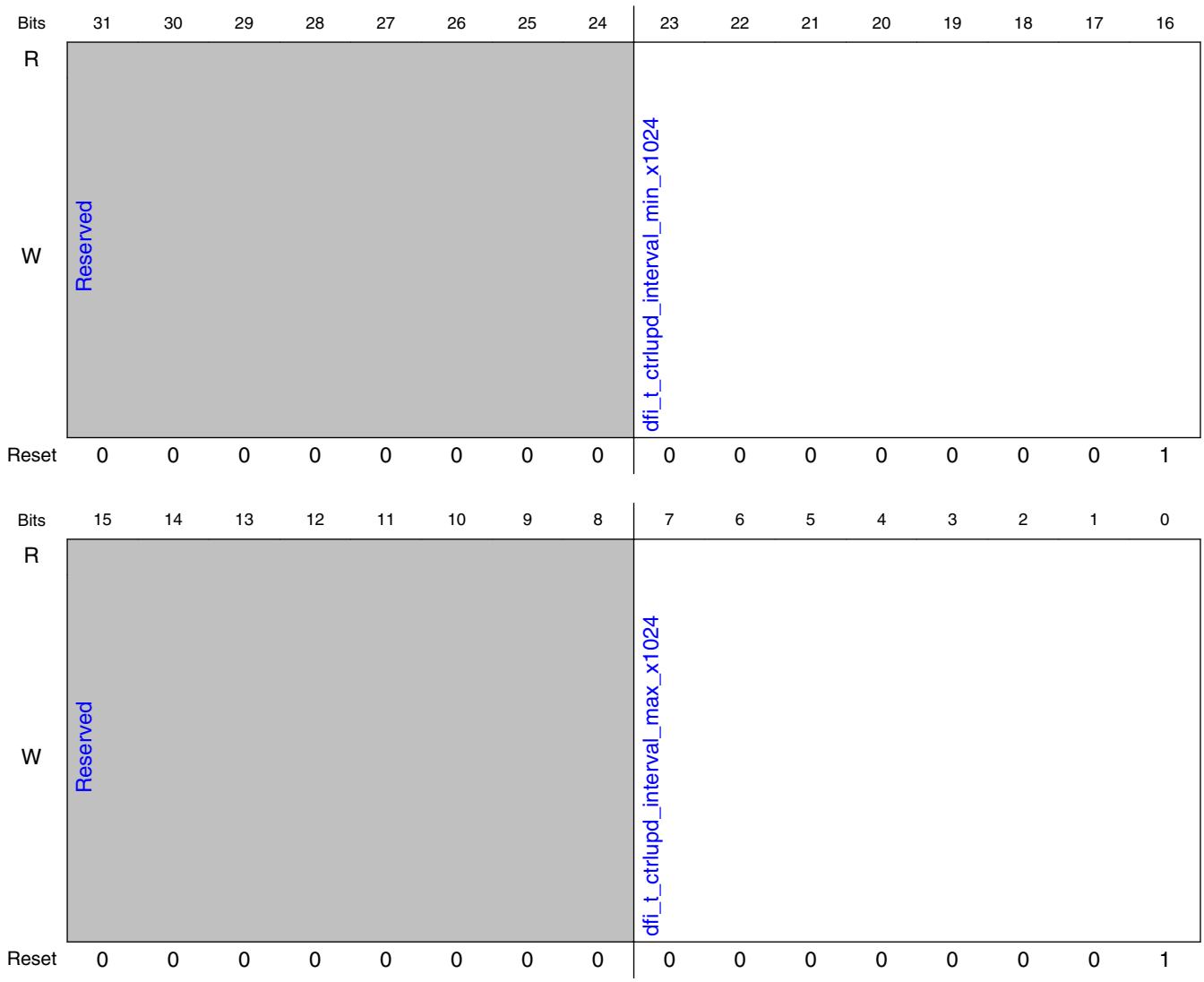
Field	Function
25-16 dfi_t_ctrlup_max	Specifies the maximum number of DFI clock cycles that the dfi_ctrlupd_req signal can assert. Lowest value to assign to this variable is 0x40.
15-10 —	Reserved
9-0 dfi_t_ctrlup_min	Specifies the minimum number of DFI clock cycles that the dfi_ctrlupd_req signal must be asserted. The DDRC expects the PHY to respond within this time. If the PHY does not respond, the DDRC will de-assert dfi_ctrlupd_req after dfi_t_ctrlup_min + 2 cycles. Lowest value to assign to this variable is 0x3.

9.3.3.1.54 DFI Update Register 1 (DFIUPD1)

9.3.3.1.54.1 Offset

Register	Offset
DFIUPD1	1A4h

9.3.3.1.54.2 Diagram



9.3.3.1.54.3 Fields

Field	Function
31-24 —	Reserved
23-16 dfi_t_ctrlupd_interval_min_x1024	This is the minimum amount of time between DDRC initiated DFI update requests (which is executed whenever the DDRC is idle). Set this number higher to reduce the frequency of update requests, which can have a small impact on the latency of the first read request when the DDRC is idle. Minimum allowed value for this field is 1. Unit: 1024 DFI clock cycles
15-8 —	Reserved

Table continues on the next page...

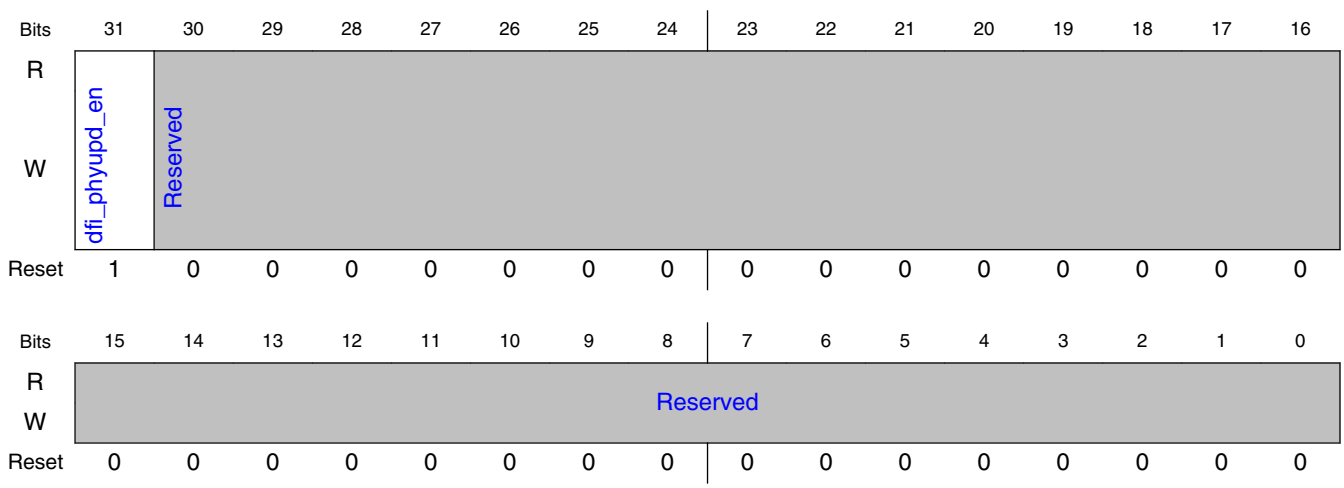
Field	Function
7-0 dfi_t_ctrlupd_interval_max_x1024	This is the maximum amount of time between DDRC initiated DFI update requests. This timer resets with each update request; when the timer expires dfi_ctrlupd_req is sent and traffic is blocked until the dfi_ctrlupd_ackx is received. PHY can use this idle time to recalibrate the delay lines to the DLLs. The DFI controller update is also used to reset PHY FIFO pointers in case of data capture errors. Updates are required to maintain calibration over PVT, but frequent updates may impact performance. Minimum allowed value for this field is 1. Note: Value programmed for DFIUPD1.dfi_t_ctrlupd_interval_max_x1024 must be greater than DFIUPD1.dfi_t_ctrlupd_interval_min_x1024. Unit: 1024 DFI clock cycles

9.3.3.1.55 DFI Update Register 2 (DFIUPD2)

9.3.3.1.55.1 Offset

Register	Offset
DFIUPD2	1A8h

9.3.3.1.55.2 Diagram



9.3.3.1.55.3 Fields

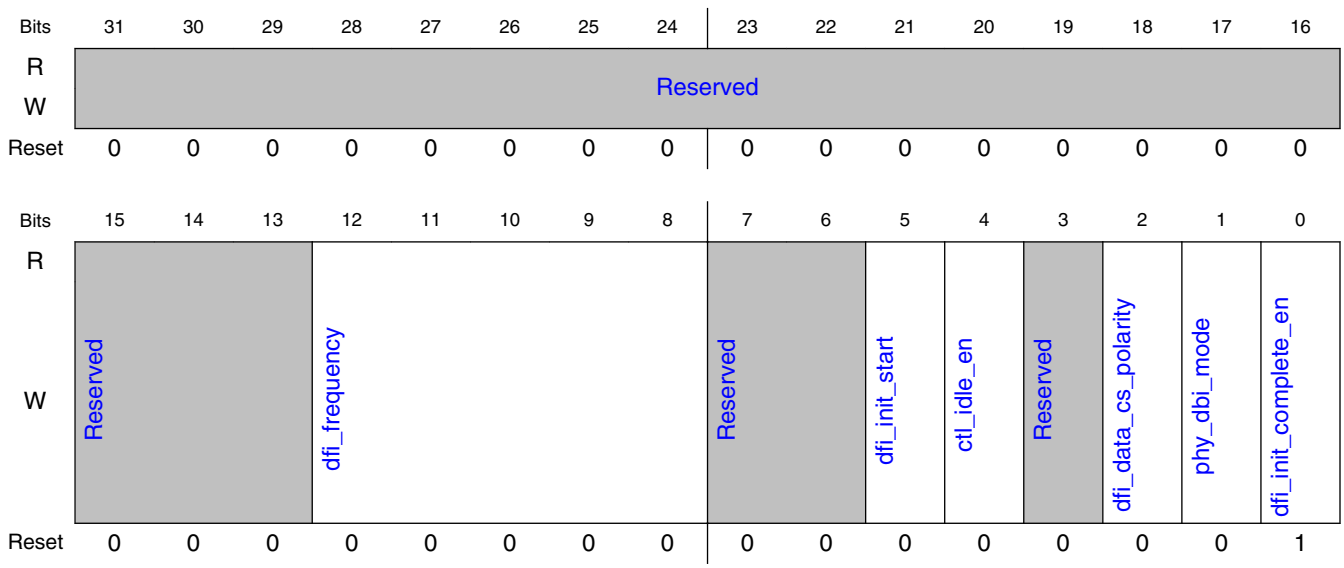
Field	Function
31 dfi_phyupd_en	Enables the support for acknowledging PHY-initiated updates: 0b - Disabled 1b - Enabled
30-0 —	Reserved

9.3.3.1.56 DFI Miscellaneous Control Register (DFIMISC)

9.3.3.1.56.1 Offset

Register	Offset
DFIMISC	1B0h

9.3.3.1.56.2 Diagram



9.3.3.1.56.3 Fields

Field	Function
31-13 —	Reserved
12-8 dfi_frequency	Indicates the operating frequency of the system. The number of supported frequencies and the mapping of signal values to clock frequencies are defined by the PHY.
7-6 —	Reserved
5 dfi_init_start	PHY init start request signal.When asserted it triggers the PHY init start request
4 ctl_idle_en	Enables support of ctl_idle signal, which is non-DFI related pin specific to certain PHYs. See signal description of ctl_idle signal for further details of ctl_idle functionality.
3	Reserved

Table continues on the next page...

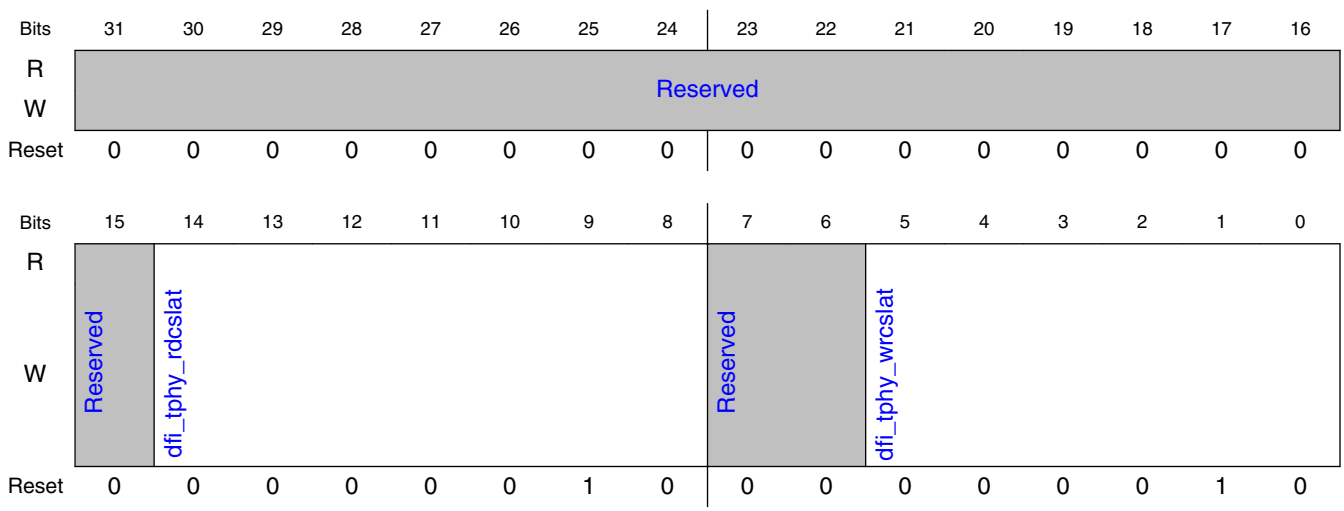
Field	Function
—	
2 dfi_data_cs_polarity	Defines polarity of dfi_wrdata_cs and dfi_rddata_cs signals. 0b - Signals are active low 1b - Signals are active high
1 phy_dbi_mode	DBI implemented in DDRC or PHY. Present only in designs configured to support DDR4 and LPDDR4. 0b - DDRC implements DBI functionality. 1b - PHY implements DBI functionality.
0 dfi_init_complete_en	PHY initialization complete enable signal. When asserted the dfi_init_complete signal can be used to trigger SDRAM initialisation

9.3.3.1.57 DFI Timing Register 2 (DFITMG2)

9.3.3.1.57.1 Offset

Register	Offset
DFITMG2	1B4h

9.3.3.1.57.2 Diagram



9.3.3.1.57.3 Fields

Field	Function
31-15	Reserved

Table continues on the next page...

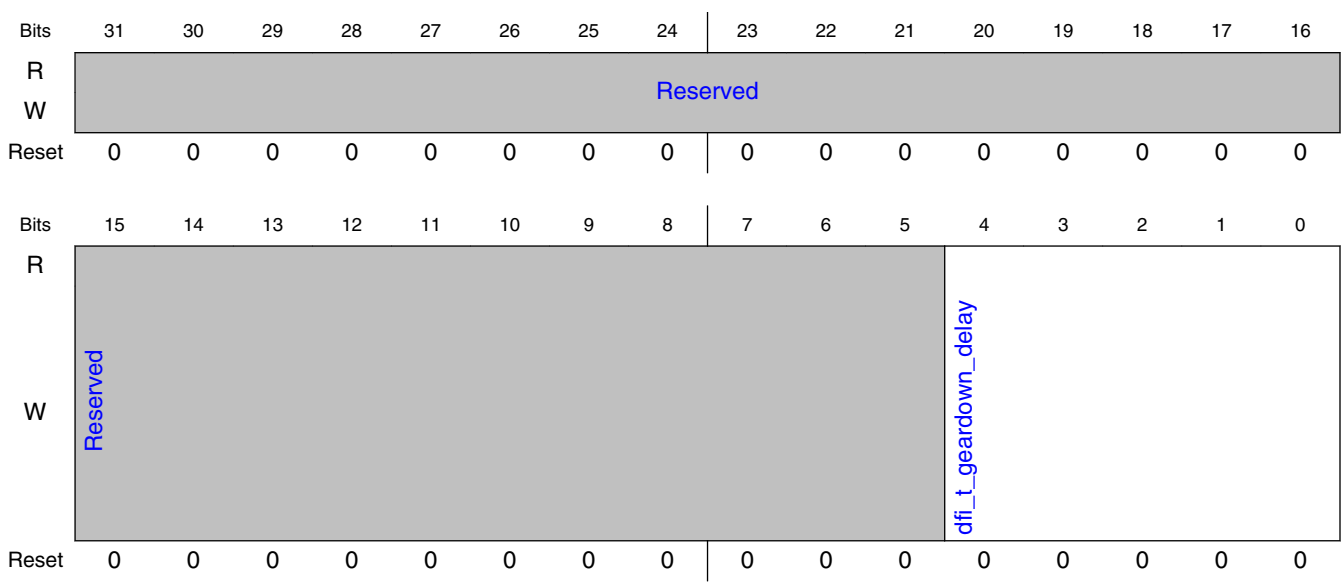
Field	Function
—	
14-8 dfi_tphy_rdcslat	Number of DFI PHY clock cycles between when a read command is sent on the DFI control interface and when the associated dfi_rddata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_rdcslat. Refer to PHY specification for correct value.
7-6 —	Reserved
5-0 dfi_tphy_wrcslat	Number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the associated dfi_wrdata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_wrcslat. Refer to PHY specification for correct value.

9.3.3.1.58 DFI Timing Register 3 (DFITMG3)

9.3.3.1.58.1 Offset

Register	Offset
DFITMG3	1B8h

9.3.3.1.58.2 Diagram



9.3.3.1.58.3 Fields

Field	Function
31-5	Reserved

Table continues on the next page...

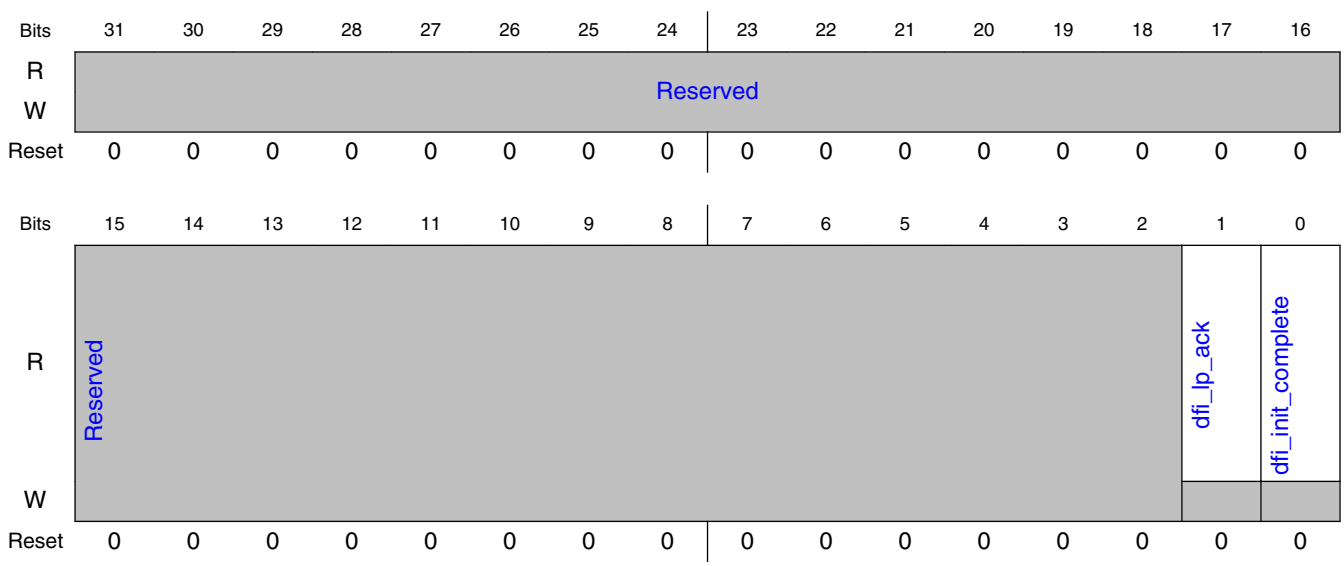
Field	Function
—	
4-0 dfi_t_geardown_delay	The delay from dfi_geardown_en assertion to the time of the PHY being ready to receive commands. Refer to PHY specification for correct value. When the controller is operating in 1:2 frequency ratio mode, program this to (tgeardown_delay/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.59 DFI Status Register (DFISTAT)

9.3.3.1.59.1 Offset

Register	Offset
DFISTAT	1BCh

9.3.3.1.59.2 Diagram



9.3.3.1.59.3 Fields

Field	Function
31-2 —	Reserved
1 dfi_lp_ack	Stores the value of the dfi_lp_ack input to the controller.

Table continues on the next page...

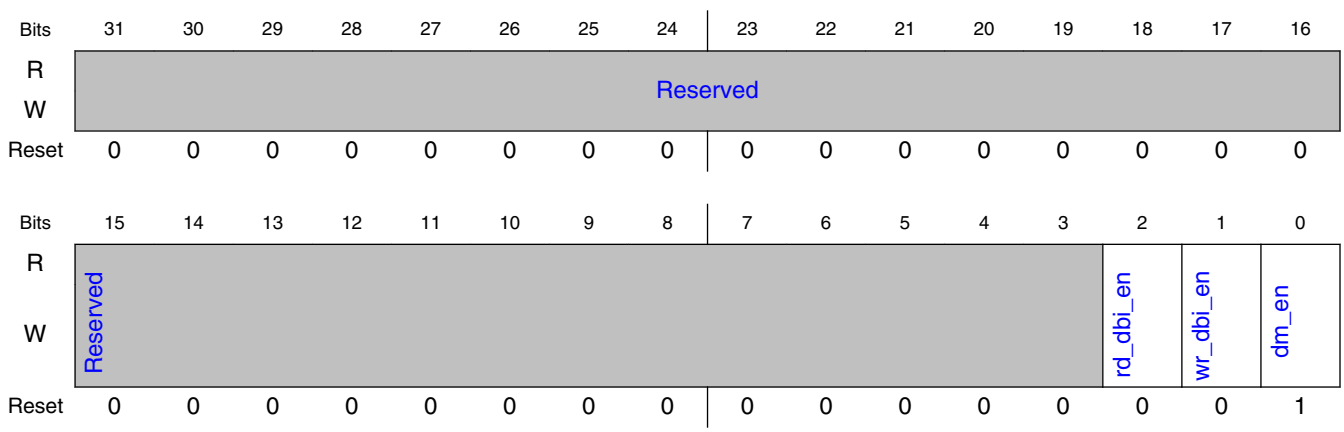
Field	Function
0 dfi_init_complete	The status flag register which announces when the DFI initialization has been completed. The DFI INIT triggered by dfi_init_start signal and then the dfi_init_complete flag is polled to know when the initialization is done.

9.3.3.1.60 DM/DBI Control Register (DBICTL)

9.3.3.1.60.1 Offset

Register	Offset
DBICTL	1C0h

9.3.3.1.60.2 Diagram



9.3.3.1.60.3 Fields

Field	Function
31-3 —	Reserved
2 rd_dbi_en	Read DBI enable signal in DDRC. - 0 - Read DBI is disabled. - 1 - Read DBI is enabled. This signal must be set the same value as DRAM's mode register. - DDR4: MR5 bit A12. When x4 devices are used, this signal must be set to 0. - LPDDR4: MR3[6]
1 wr_dbi_en	This signal must be set the same value as DRAM's mode register. - DDR4: MR5 bit A11. When x4 devices are used, this signal must be set to 0. - LPDDR4: MR3[7] 0b - Write DBI is disabled 1b - Write DBI is enabled.

Table continues on the next page...

Field	Function
0 dm_en	DM enable signal in DDRC. This signal must be set the same logical value as DRAM's mode register. - DDR4: Set this to same value as MR5 bit A10. When x4 devices are used, this signal must be set to 0. - LPDDR4: Set this to inverted value of MR13[5] which is opposite polarity from this signal 0b - DM is disabled 1b - DM is enabled

9.3.3.1.61 Address Map Register 0 (ADDRMAP0)

9.3.3.1.61.1 Offset

Register	Offset
ADDRMAP0	200h

9.3.3.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											addrmap_cs_bit0				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.61.3 Fields

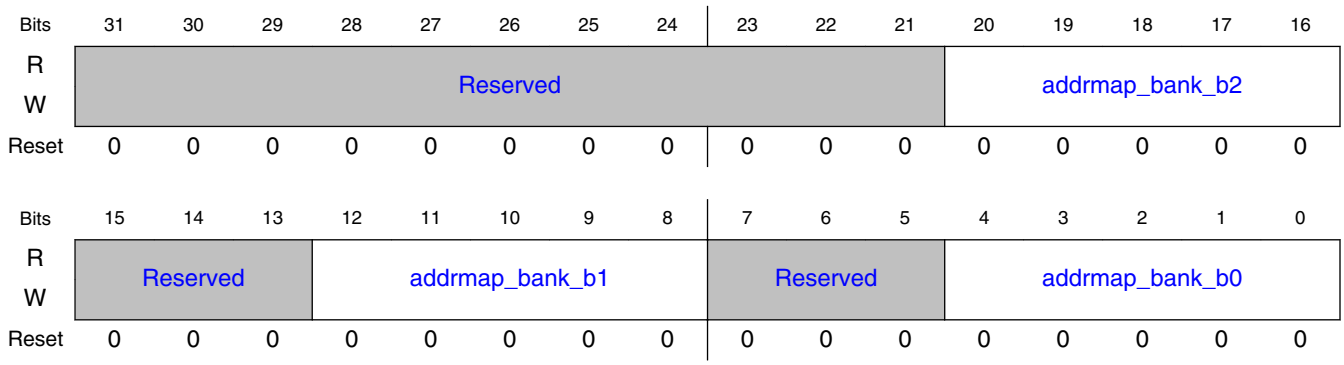
Field	Function
31-5 —	Reserved
4-0 addrmap_cs_bit0	Selects the HIF address bit used as rank address bit 0. Valid Range: 0 to 28, and 31 Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 31, rank address bit 0 is set to 0.

9.3.3.1.62 Address Map Register 1 (ADDRMAP1)

9.3.3.1.62.1 Offset

Register	Offset
ADDRMAP1	204h

9.3.3.1.62.2 Diagram



9.3.3.1.62.3 Fields

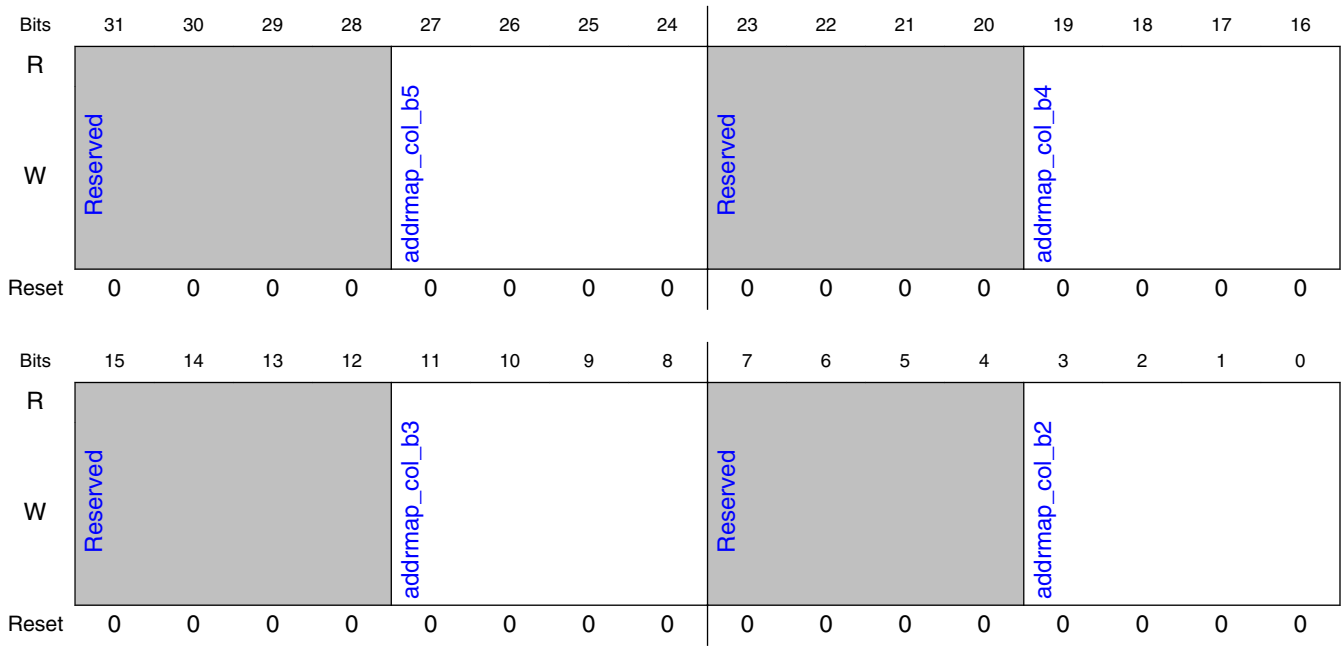
Field	Function
31-21 —	Reserved
20-16 addrmap_bank_b2	Selects the HIF address bit used as bank address bit 2. Valid Range: 0 to 30 and 31 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 31, bank address bit 2 is set to 0.
15-13 —	Reserved
12-8 addrmap_bank_b1	Selects the HIF address bits used as bank address bit 1. Valid Range: 0 to 31 Internal Base: 3 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.
7-5 —	Reserved
4-0 addrmap_bank_b0	Selects the HIF address bits used as bank address bit 0. Valid Range: 0 to 31 Internal Base: 2 The selected HIF address bit for each of the bank address bits is determined by adding the internal base to the value of this field.

9.3.3.1.63 Address Map Register 2 (ADDRMAP2)

9.3.3.1.63.1 Offset

Register	Offset
ADDRMAP2	208h

9.3.3.1.63.2 Diagram



9.3.3.1.63.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_col_b5	- Full bus width mode: Selects the HIF address bit used as column address bit 5. - Half bus width mode: Selects the HIF address bit used as column address bit 6. - Quarter bus width mode: Selects the HIF address bit used as column address bit 7 . Valid Range: 0 to 7, and 15 Internal Base: 5 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
23-20 —	Reserved
19-16 addrmap_col_b4	- Full bus width mode: Selects the HIF address bit used as column address bit 4. - Half bus width mode: Selects the HIF address bit used as column address bit 5. - Quarter bus width mode: Selects the HIF address bit used as column address bit 6. Valid Range: 0 to 7, and 15 Internal Base: 4 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
15-12	Reserved

Table continues on the next page...

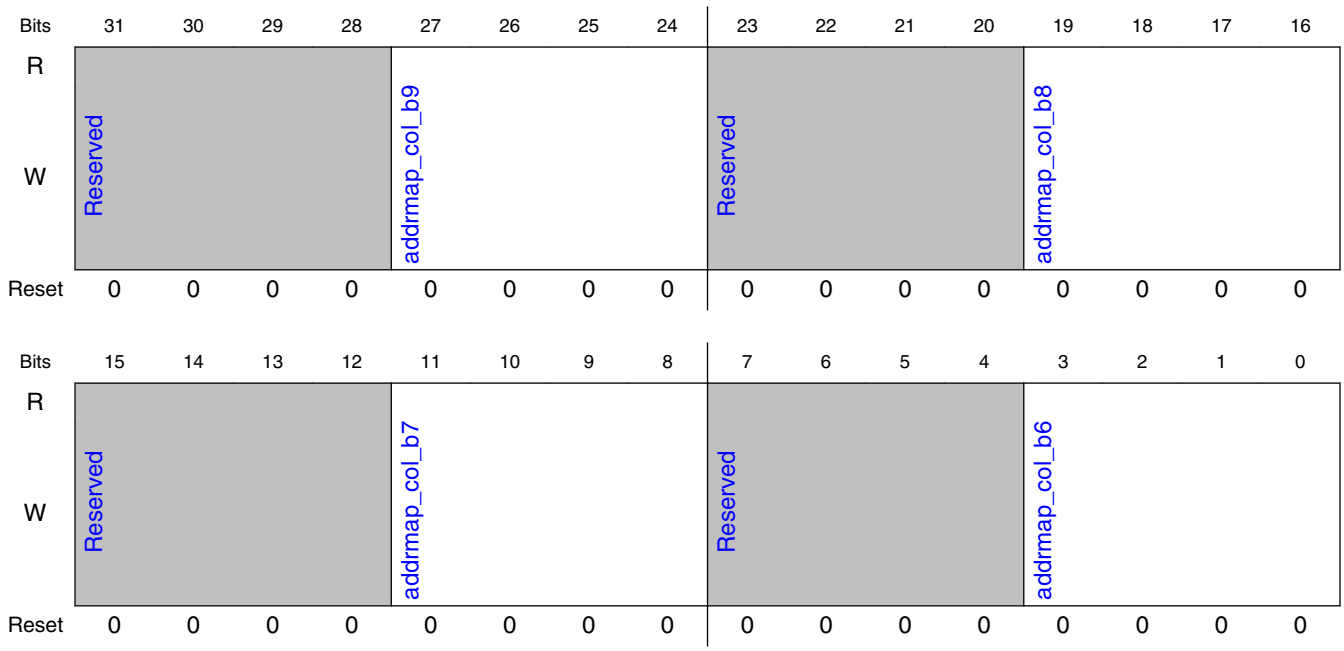
Field	Function
—	
11-8 addrmap_col_b3	- Full bus width mode: Selects the HIF address bit used as column address bit 3. - Half bus width mode: Selects the HIF address bit used as column address bit 4. - Quarter bus width mode: Selects the HIF address bit used as column address bit 5. Valid Range: 0 to 7 Internal Base: 3 The selected HIF address bit is determined by adding the internal base to the value of this field. Note, if DDRC_INCL_ARB=1, MEMC_BURST_LENGTH=16, Full bus width (MSTR.data_bus_width=00) and BL16 (MSTR.burst_rdwr=1000), it is recommended to program this to 0.
7-4 —	Reserved
3-0 addrmap_col_b2	- Full bus width mode: Selects the HIF address bit used as column address bit 2. - Half bus width mode: Selects the HIF address bit used as column address bit 3. - Quarter bus width mode: Selects the HIF address bit used as column address bit 4. Valid Range: 0 to 7 Internal Base: 2 The selected HIF address bit is determined by adding the internal base to the value of this field. Note, if DDRC_INCL_ARB=1 and MEMC_BURST_LENGTH=8, it is required to program this to 0 unless: - in Half or Quarter bus width (MSTR.data_bus_width!=00) and - PCCFG.bl_exp_mode==1 and either - In DDR4 and ADDRMAP8.addrmap_bg_b0==0 or - In LPDDR4 and ADDRMAP1.addrmap_bank_b0==0 If DDRC_INCL_ARB=1 and MEMC_BURST_LENGTH=16, it is required to program this to 0 unless: - in Half or Quarter bus width (MSTR.data_bus_width!=00) and - PCCFG.bl_exp_mode==1 and - In DDR4 and ADDRMAP8.addrmap_bg_b0==0 Otherwise, if MEMC_BURST_LENGTH=8 and Full Bus Width (MSTR.data_bus_width==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2. If MEMC_BURST_LENGTH=16 and Full Bus Width (MSTR.data_bus_width==00), it is recommended to program this to 0 so that HIF[2] maps to column address bit 2. If MEMC_BURST_LENGTH=16 and Half Bus Width (MSTR.data_bus_width==01), it is recommended to program this to 0 so that HIF[2] maps to column address bit 3.

9.3.3.1.64 Address Map Register 3 (ADDRMAP3)

9.3.3.1.64.1 Offset

Register	Offset
ADDRMAP3	20Ch

9.3.3.1.64.2 Diagram



9.3.3.1.64.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_col_b9	- Full bus width mode: Selects the HIF address bit used as column address bit 9. - Half bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode). - Quarter bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode). Valid Range: 0 to 7, and 15 Internal Base: 9 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0. Note: Per JEDEC DDR2/3/mDDR specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
23-20 —	Reserved
19-16 addrmap_col_b8	- Full bus width mode: Selects the HIF address bit used as column address bit 8. - Half bus width mode: Selects the HIF address bit used as column address bit 9. - Quarter bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode). Valid Range: 0 to 7, and 15 Internal Base: 8 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0. Note: Per JEDEC DDR2/3/mDDR specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
15-12 —	Reserved

Table continues on the next page...

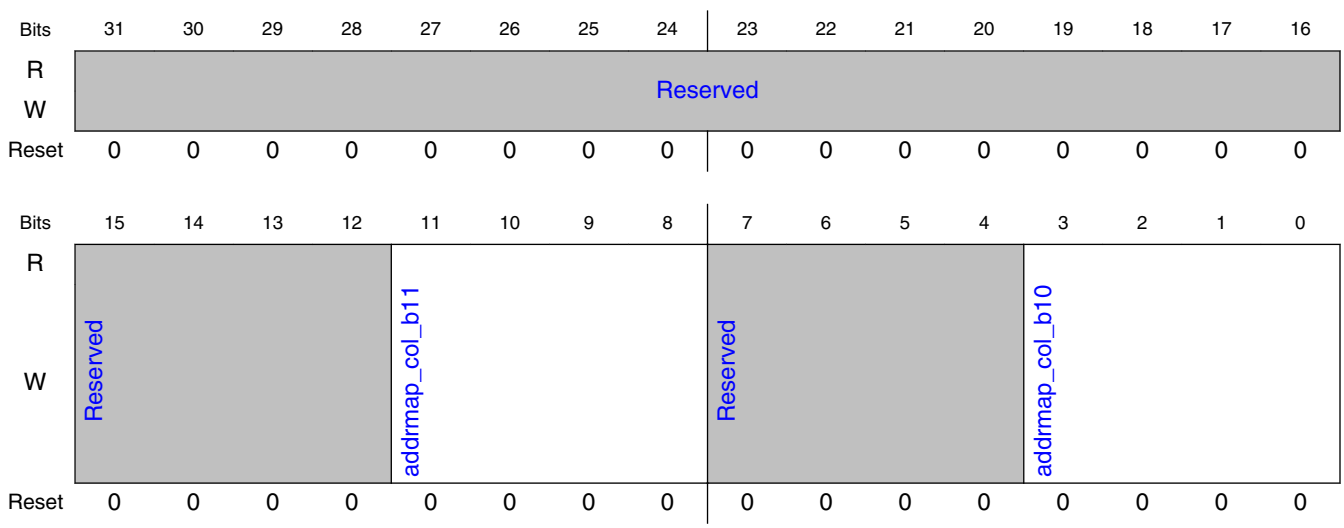
Field	Function
11-8 addrmap_col_b7	- Full bus width mode: Selects the HIF address bit used as column address bit 7. - Half bus width mode: Selects the HIF address bit used as column address bit 8. - Quarter bus width mode: Selects the HIF address bit used as column address bit 9. Valid Range: 0 to 7, and 15 Internal Base: 7 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.
7-4 —	Reserved
3-0 addrmap_col_b6	- Full bus width mode: Selects the HIF address bit used as column address bit 6. - Half bus width mode: Selects the HIF address bit used as column address bit 7. - Quarter bus width mode: Selects the HIF address bit used as column address bit 8. Valid Range: 0 to 7, and 15 Internal Base: 6 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0.

9.3.3.1.65 Address Map Register 4 (ADDRMAP4)

9.3.3.1.65.1 Offset

Register	Offset
ADDRMAP4	210h

9.3.3.1.65.2 Diagram



9.3.3.1.65.3 Fields

Field	Function
31-12	Reserved

Table continues on the next page...

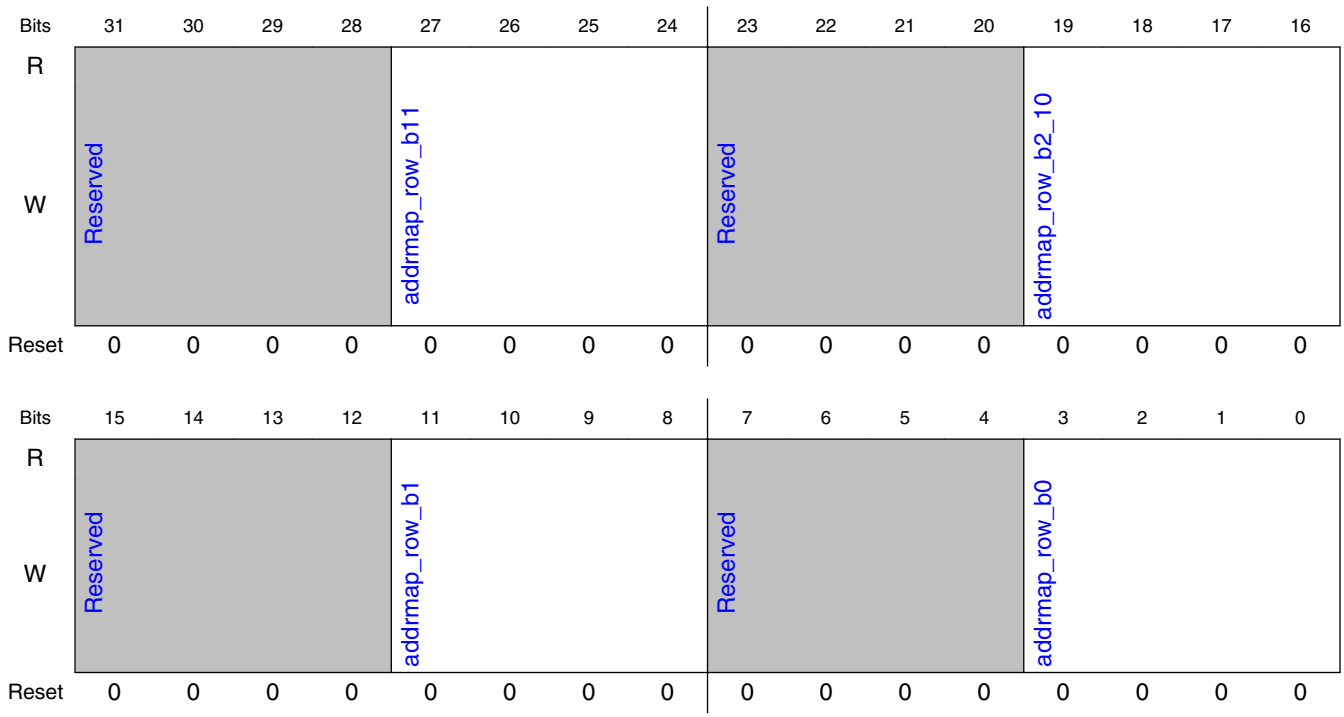
Field	Function
—	
11-8 addrmap_col_b11	- Full bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode). - Half bus width mode: Unused. To make it unused, this should be tied to 4'hF. - Quarter bus width mode: Unused. To make it unused, this must be tied to 4'hF. Valid Range: 0 to 7, and 15 Internal Base: 11 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0. Note: Per JEDEC DDR2/3/mDDR specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.
7-4 —	Reserved
3-0 addrmap_col_b10	- Full bus width mode: Selects the HIF address bit used as column address bit 11 (10 in LPDDR2/LPDDR3 mode). - Half bus width mode: Selects the HIF address bit used as column address bit 13 (11 in LPDDR2/LPDDR3 mode). - Quarter bus width mode: UNUSED. To make it unused, this must be tied to 4'hF. Valid Range: 0 to 7, and 15 Internal Base: 10 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, this column address bit is set to 0. Note: Per JEDEC DDR2/3/mDDR specification, column address bit 10 is reserved for indicating auto-precharge, and hence no source address bit can be mapped to column address bit 10. In LPDDR2/LPDDR3, there is a dedicated bit for auto-precharge in the CA bus and hence column bit 10 is used.

9.3.3.1.66 Address Map Register 5 (ADDRMAP5)

9.3.3.1.66.1 Offset

Register	Offset
ADDRMAP5	214h

9.3.3.1.66.2 Diagram



9.3.3.1.66.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_row_b11	Selects the HIF address bit used as row address bit 11. Valid Range: 0 to 11, and 15 Internal Base: 17 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 11 is set to 0.
23-20 —	Reserved
19-16 addrmap_row_b2_10	Selects the HIF address bits used as row address bits 2 to 10. Valid Range: 0 to 11, and 15 Internal Base: 8 (for row address bit 2), 9 (for row address bit 3), 10 (for row address bit 4) etc increasing to 16 (for row address bit 10) The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. When value 15 is used the values of row address bits 2 to 10 are defined by registers ADDRMAP9, ADDRMAP10, ADDRMAP11.
15-12 —	Reserved
11-8 addrmap_row_b1	Selects the HIF address bits used as row address bit 1. Valid Range: 0 to 11 Internal Base: 7 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.
7-4 —	Reserved

Table continues on the next page...

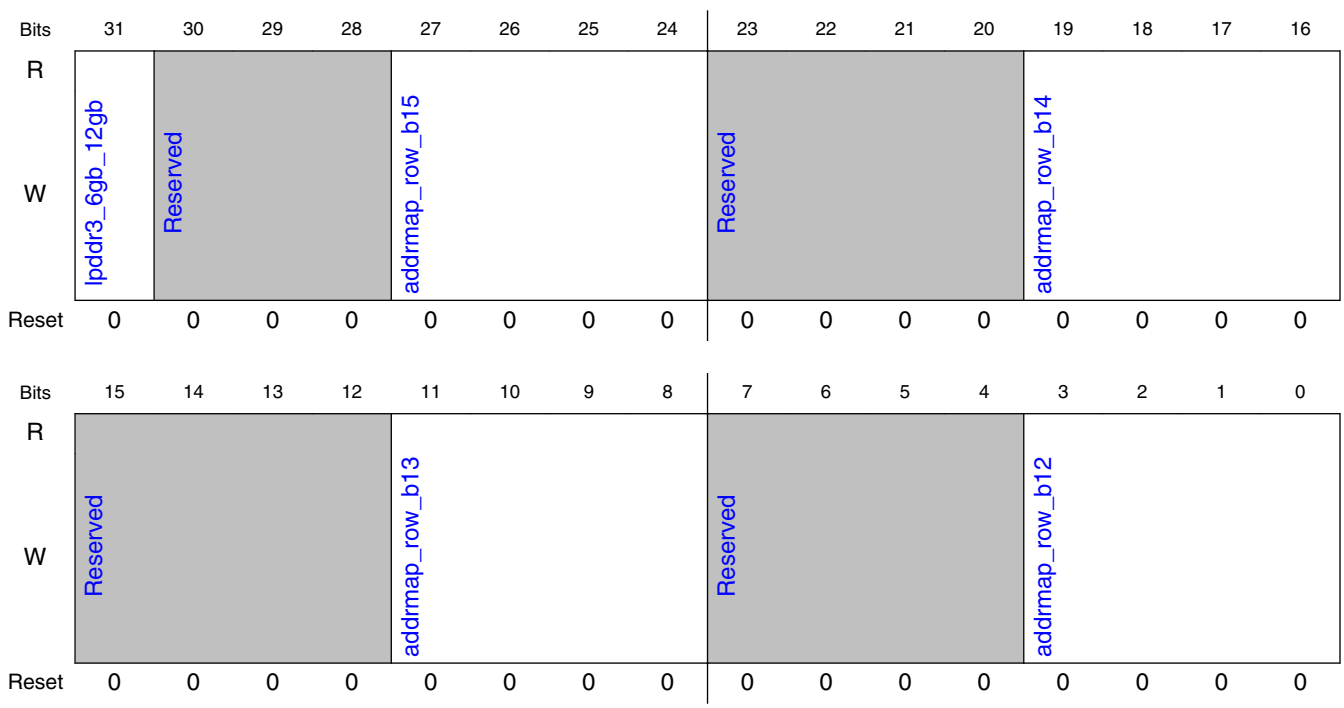
Field	Function
3-0 addrmap_row_b0	Selects the HIF address bits used as row address bit 0. Valid Range: 0 to 11 Internal Base: 6 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field.

9.3.3.1.67 Address Map Register 6 (ADDRMAP6)

9.3.3.1.67.1 Offset

Register	Offset
ADDRMAP6	218h

9.3.3.1.67.2 Diagram



9.3.3.1.67.3 Fields

Field	Function
31 lpddr3_6gb_12gb	Set this to 1 if there is an LPDDR3 SDRAM 6Gb or 12Gb device in use. - 1 - LPDDR3 SDRAM 6Gb/12Gb device in use. Every address having row[14:13]==2'b11 is considered as invalid - 0 - non-LPDDR3 6Gb/12Gb device in use. All addresses are valid Present only in designs configured to support LPDDR3.

Table continues on the next page...

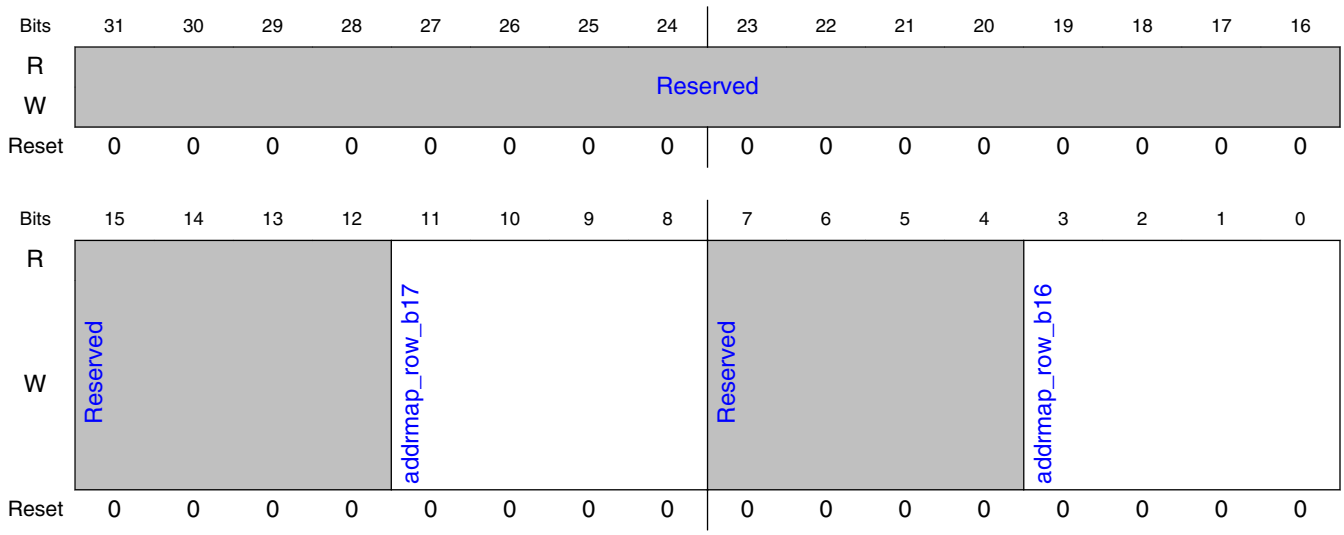
Field	Function
30-28 —	Reserved
27-24 addrmap_row_b15	Selects the HIF address bit used as row address bit 15. Valid Range: 0 to 11, and 15 Internal Base: 21 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 15 is set to 0.
23-20 —	Reserved
19-16 addrmap_row_b14	Selects the HIF address bit used as row address bit 14. Valid Range: 0 to 11, and 15 Internal Base: 20 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 14 is set to 0.
15-12 —	Reserved
11-8 addrmap_row_b13	Selects the HIF address bit used as row address bit 13. Valid Range: 0 to 11, and 15 Internal Base: 19 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 13 is set to 0.
7-4 —	Reserved
3-0 addrmap_row_b12	Selects the HIF address bit used as row address bit 12. Valid Range: 0 to 11, and 15 Internal Base: 18 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 12 is set to 0.

9.3.3.1.68 Address Map Register 7 (ADDRMAP7)

9.3.3.1.68.1 Offset

Register	Offset
ADDRMAP7	21Ch

9.3.3.1.68.2 Diagram



9.3.3.1.68.3 Fields

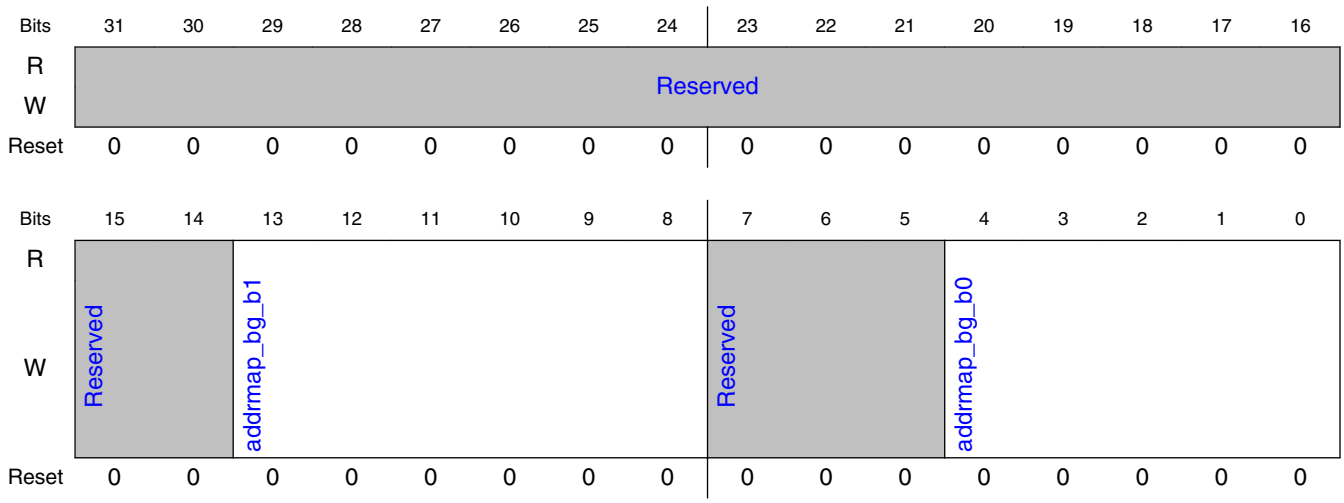
Field	Function
31-12 —	Reserved
11-8 addrmap_row_b17	Selects the HIF address bit used as row address bit 17. Valid Range: 0 to 11, and 15 Internal Base: 23 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 17 is set to 0.
7-4 —	Reserved
3-0 addrmap_row_b16	Selects the HIF address bit used as row address bit 16. Valid Range: 0 to 11, and 15 Internal Base: 22 The selected HIF address bit is determined by adding the internal base to the value of this field. If set to 15, row address bit 16 is set to 0.

9.3.3.1.69 Address Map Register 8 (ADDRMAP8)

9.3.3.1.69.1 Offset

Register	Offset
ADDRMAP8	220h

9.3.3.1.69.2 Diagram



9.3.3.1.69.3 Fields

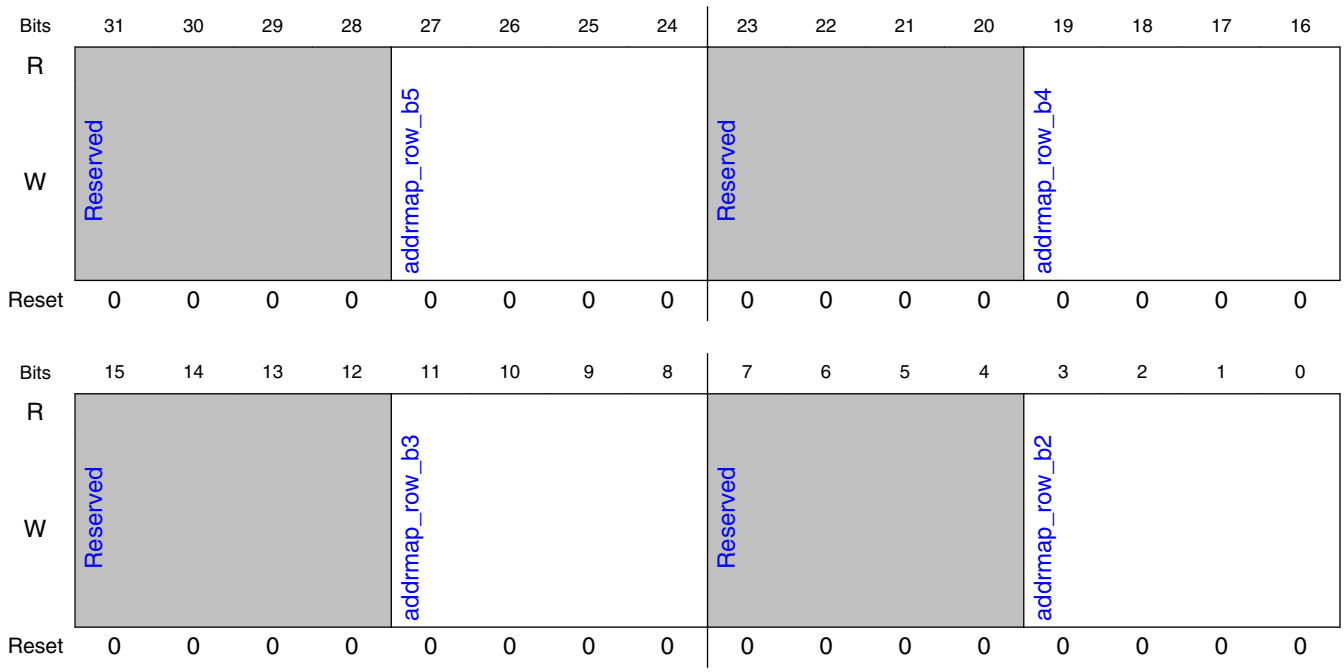
Field	Function
31-14 —	Reserved
13-8 addrmap_bg_b1	Selects the HIF address bits used as bank group address bit 1. Valid Range: 0 to 31, and 63 Internal Base: 3 The selected HIF address bit for each of the bank group address bits is determined by adding the internal base to the value of this field. If set to 63, bank group address bit 1 is set to 0.
7-5 —	Reserved
4-0 addrmap_bg_b0	Selects the HIF address bits used as bank group address bit 0. Valid Range: 0 to 31 Internal Base: 2 The selected HIF address bit for each of the bank group address bits is determined by adding the internal base to the value of this field.

9.3.3.1.70 Address Map Register 9 (ADDRMAP9)

9.3.3.1.70.1 Offset

Register	Offset
ADDRMAP9	224h

9.3.3.1.70.2 Diagram



9.3.3.1.70.3 Fields

Field	Function
31-28 —	Reserved
27-24 addrmap_row_b5	Selects the HIF address bits used as row address bit 5. Valid Range: 0 to 11 Internal Base: 11 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
23-20 —	Reserved
19-16 addrmap_row_b4	Selects the HIF address bits used as row address bit 4. Valid Range: 0 to 11 Internal Base: 10 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
15-12 —	Reserved
11-8 addrmap_row_b3	Selects the HIF address bits used as row address bit 3. Valid Range: 0 to 11 Internal Base: 9 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
7-4 —	Reserved

Table continues on the next page...

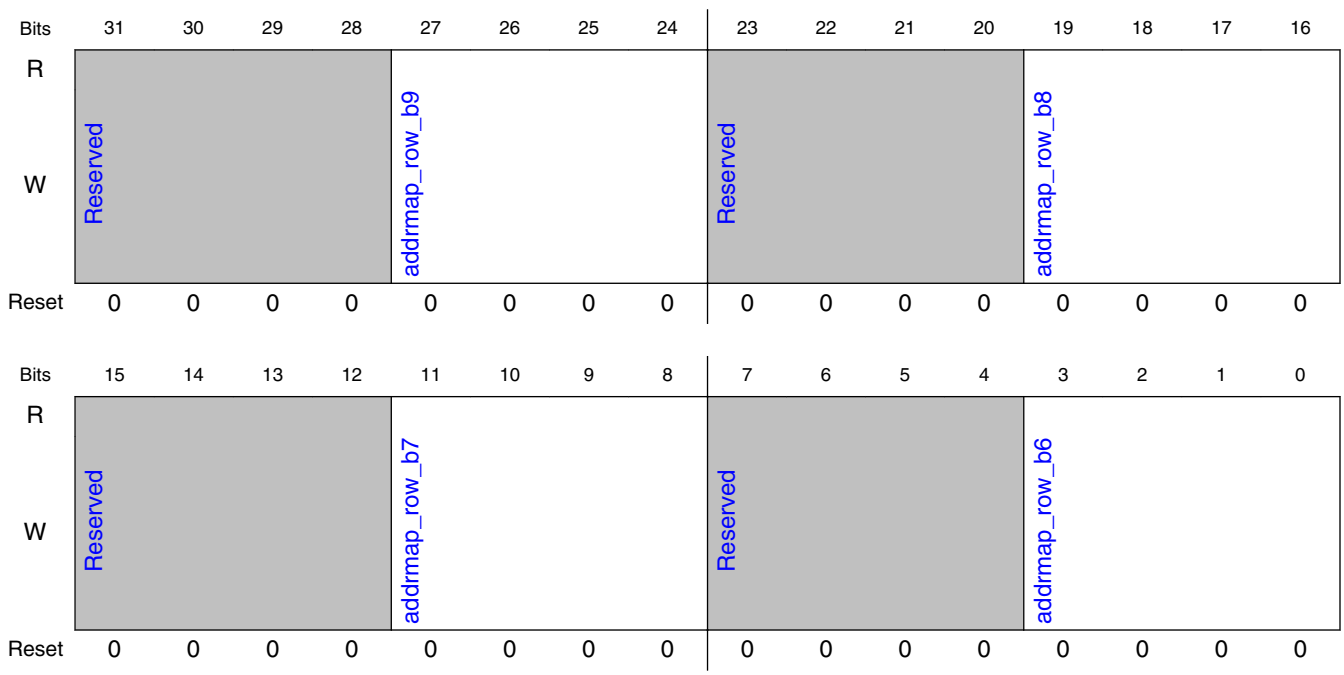
Field	Function
3-0 addrmap_row_b2	Selects the HIF address bits used as row address bit 2. Valid Range: 0 to 11 Internal Base: 8 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.3.3.1.71 Address Map Register 10 (ADDRMAP10)

9.3.3.1.71.1 Offset

Register	Offset
ADDRMAP10	228h

9.3.3.1.71.2 Diagram



9.3.3.1.71.3 Fields

Field	Function
31-28 —	Reserved

Table continues on the next page...

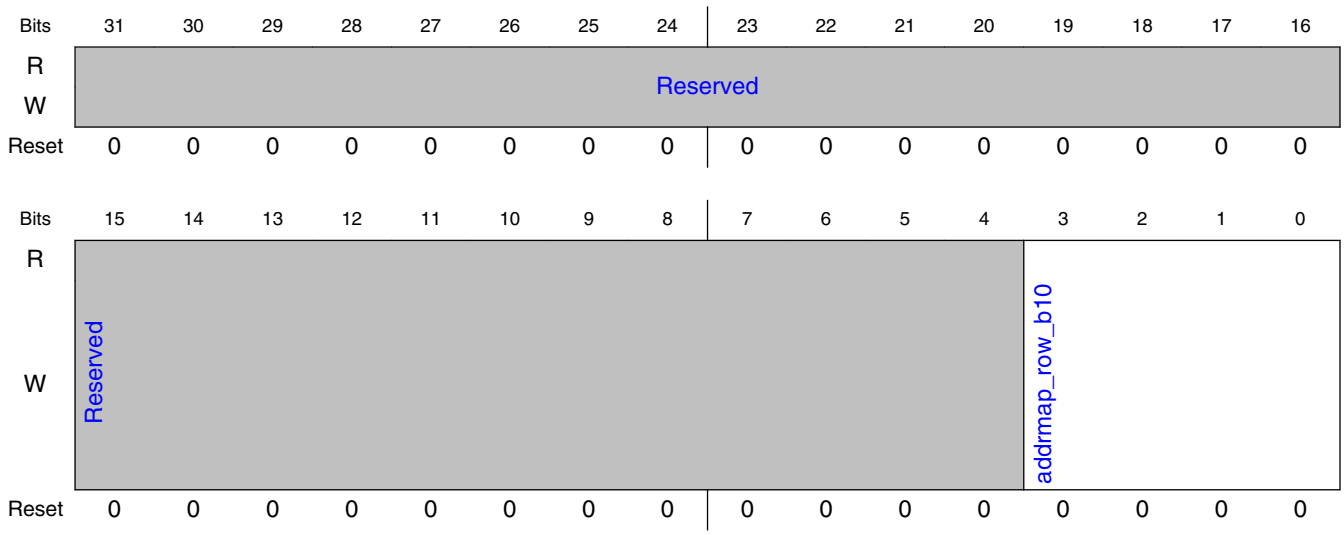
Field	Function
27-24 addrmap_row_b9	Selects the HIF address bits used as row address bit 9. Valid Range: 0 to 11 Internal Base: 15 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
23-20 —	Reserved
19-16 addrmap_row_b8	Selects the HIF address bits used as row address bit 8. Valid Range: 0 to 11 Internal Base: 14 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
15-12 —	Reserved
11-8 addrmap_row_b7	Selects the HIF address bits used as row address bit 7. Valid Range: 0 to 11 Internal Base: 13 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.
7-4 —	Reserved
3-0 addrmap_row_b6	Selects the HIF address bits used as row address bit 6. Valid Range: 0 to 11 Internal Base: 12 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.3.3.1.72 Address Map Register 11 (ADDRMAP11)

9.3.3.1.72.1 Offset

Register	Offset
ADDRMAP11	22Ch

9.3.3.1.72.2 Diagram



9.3.3.1.72.3 Fields

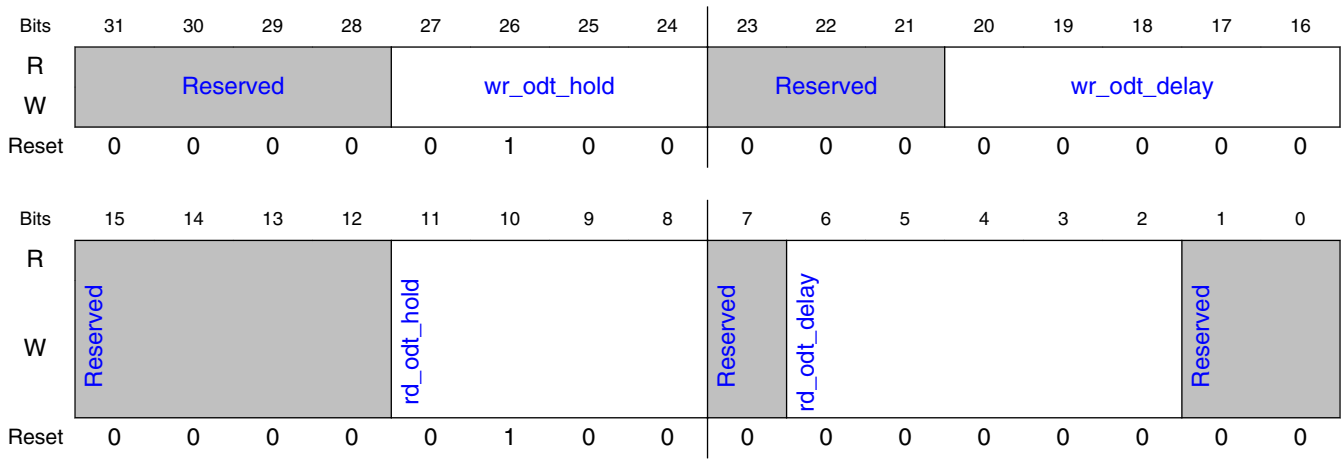
Field	Function
31-4 —	Reserved
3-0 addrmap_row_b10	Selects the HIF address bits used as row address bit 10. Valid Range: 0 to 11 Internal Base: 16 The selected HIF address bit for each of the row address bits is determined by adding the internal base to the value of this field. This register field is used only when ADDRMAP5.addrmap_row_b2_10 is set to value 15.

9.3.3.1.73 ODT Configuration Register (ODTCFG)

9.3.3.1.73.1 Offset

Register	Offset
ODTCFG	240h

9.3.3.1.73.2 Diagram



9.3.3.1.73.3 Fields

Field	Function
31-28 —	Reserved
27-24 wr_odt_hold	DFI PHY clock cycles to hold ODT for a write command. The minimum supported value is 2. Recommended values: DDR2: - BL8: 0x5 (DDR2-400/533/667), 0x6 (DDR2-800), 0x7 (DDR2-1066) - BL4: 0x3 (DDR2-400/533/667), 0x4 (DDR2-800), 0x5 (DDR2-1066) DDR3: - BL8: 0x6 DDR4: - BL8: 5 + WR_PREAMBLE + CRC_MODE WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) CRC_MODE = 0 (not CRC mode), 1 (CRC mode) LPDDR3: - BL8: 7 + RU(tODTon(max)/tCK)
23-21 —	Reserved
20-16 wr_odt_delay	The delay, in DFI PHY clock cycles, from issuing a write command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR2: - CWL + AL - 3 (DDR2-400/533/667), CWL + AL - 4 (DDR2-800), CWL + AL - 5 (DDR2-1066) If (CWL + AL - 3 < 0), DDRC does not support ODT for write operation. DDR3: - 0x0 DDR4: - DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) LPDDR3: - WL - 1 - RU(tODTon(max)/tCK))
15-12 —	Reserved
11-8 rd_odt_hold	DFI PHY clock cycles to hold ODT for a read command. The minimum supported value is 2. Recommended values: DDR2: - BL8: 0x6 (not DDR2-1066), 0x7 (DDR2-1066) - BL4: 0x4 (not DDR2-1066), 0x5 (DDR2-1066) DDR3: - BL8 - 0x6 DDR4: - BL8: 5 + RD_PREAMBLE RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) LPDDR3: - BL8: 5 + RU(tDQSCK(max)/tCK) - RD(tDQSCK(min)/tCK) + RU(tODTon(max)/tCK)
7 —	Reserved
6-2 rd_odt_delay	The delay, in DFI PHY clock cycles, from issuing a read command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR2: - CL + AL - 4 (not DDR2-1066), CL + AL - 5 (DDR2-1066) If (CL + AL - 4 < 0), DDRC does not support ODT for read operation. DDR3: - CL - CWL DDR4: - CL - CWL - RD_PREAMBLE + WR_PREAMBLE + DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) RD_PREAMBLE = 1 (1tCK write

Table continues on the next page...

Field	Function
	preamble), 2 (2tCK write preamble) If (CL - CWL - RD_PREAMBLE + WR_PREAMBLE) < 0, DDRC does not support ODT for read operation. LPDDR3: - RL + RD(tDQSCK(min)/tCK) - 1 - RU(tODTon(max)/tCK)
1-0 —	Reserved

9.3.3.1.74 ODT/Rank Map Register (ODTMAP)

9.3.3.1.74.1 Offset

Register	Offset
ODTMAP	244h

9.3.3.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		rank1_rd_odt		Reserved		rank1_wr_odt		Reserved		rank0_rd_odt		Reserved		rank0_wr_odt	
Reset	0	0	1	0	0	0	1	0	0	0	0	1	0	0	0	1

9.3.3.1.74.3 Fields

Field	Function
31-14 —	Reserved
13-12 rank1_rd_odt	Indicates which remote ODTs must be turned on during a read from rank 1. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks
11-10 —	Reserved

Table continues on the next page...

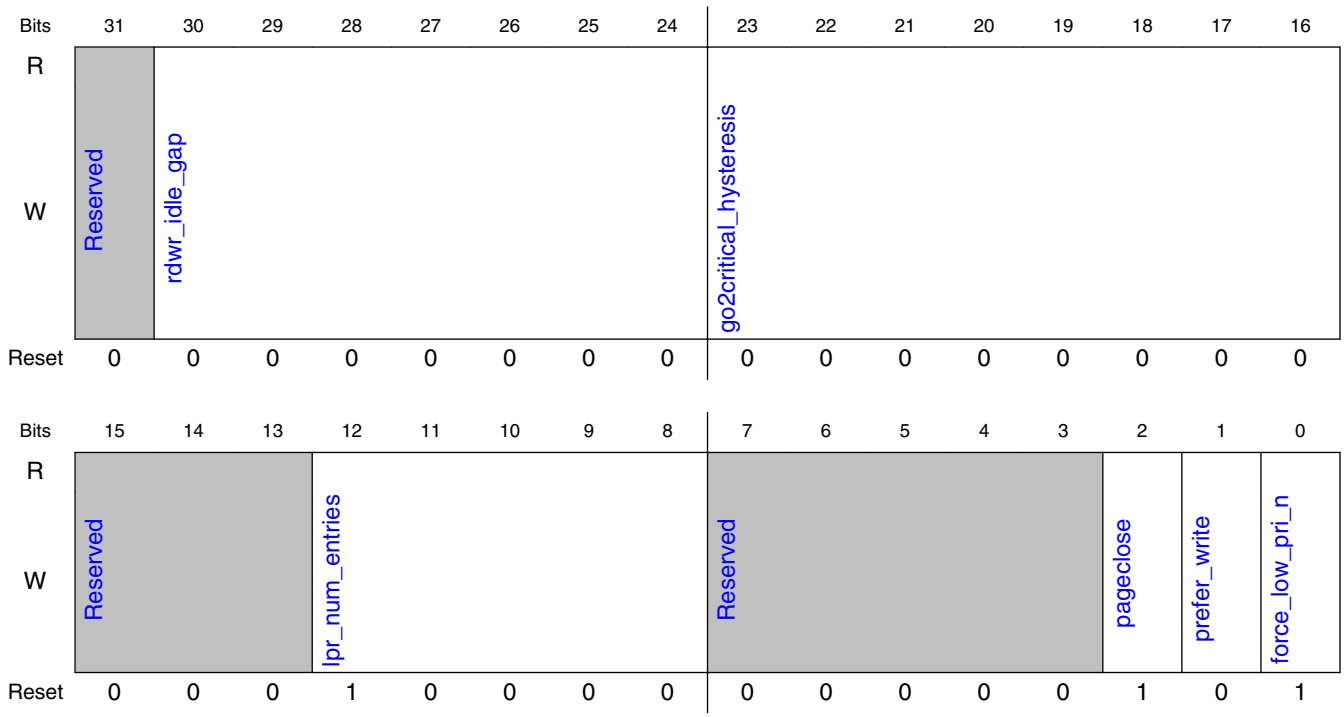
Field	Function
9-8 rank1_wr_odt	Indicates which remote ODTs must be turned on during a write to rank 1. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT. Present only in configurations that have 2 or more ranks
7-6 —	Reserved
5-4 rank0_rd_odt	Indicates which remote ODTs must be turned on during a read from rank 0. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT.
3-2 —	Reserved
1-0 rank0_wr_odt	Indicates which remote ODTs must be turned on during a write to rank 0. Each rank has a remote ODT (in the SDRAM) which can be turned on by setting the appropriate bit here. Rank 0 is controlled by the LSB; rank 1 is controlled by bit next to the LSB, etc. For each rank, set its bit to 1 to enable its ODT.

9.3.3.1.75 Scheduler Control Register (SCHED)

9.3.3.1.75.1 Offset

Register	Offset
SCHED	250h

9.3.3.1.75.2 Diagram



9.3.3.1.75.3 Fields

Field	Function
31 —	Reserved
30-24 rdwr_idle_gap	When the preferred transaction store is empty for these many clock cycles, switch to the alternate transaction store if it is non-empty. The read transaction store (both high and low priority) is the default preferred transaction store and the write transaction store is the alternative store. When prefer write over read is set this is reversed. 0x0 is a legal value for this register. When set to 0x0, the transaction store switching will happen immediately when the switching conditions become true. FOR PERFORMANCE ONLY
23-16 go2critical_hysteresis	UNUSED
15-13 —	Reserved
12-8 lpr_num_entries	Number of entries in the low priority transaction store is this value + 1. (MEMC_NO_OF_ENTRY - (SCHED.lpr_num_entries + 1)) is the number of entries available for the high priority transaction store. Setting this to maximum value allocates all entries to low priority transaction store. Setting this to 0 allocates 1 entry to low priority transaction store and the rest to high priority transaction store. Note: In ECC configurations, the numbers of write and low priority read credits issued is one less than in the non-ECC case. One entry each is reserved in the write and low-priority read CAMs for storing the RMW requests arising out of single bit error correction RMW operation.
7-3	Reserved

Table continues on the next page...

Field	Function
—	
2 pageclose	If true, bank is kept open only while there are page hit transactions available in the CAM to that bank. The last read or write command in the CAM with a bank and page hit will be executed with auto-precharge if SCHED1.pageclose_timer=0. Even if this register set to 1 and SCHED1.pageclose_timer is set to 0, explicit precharge (and not auto-precharge) may be issued in some cases where there is a mode switch between Write and Read or between LPR and HPR. The Read and Write commands that are executed as part of the ECC scrub requests are also executed without auto-precharge. If false, the bank remains open until there is a need to close it (to open a different page, or for page timeout or refresh timeout) - also known as open page policy. The open page policy can be overridden by setting the per-command-autopre bit on the HIF interface (hif_cmd_autopre). The pageclose feature provids a midway between Open and Close page policies. FOR PERFORMANCE ONLY.
1 prefer_write	If set then the bank selector prefers writes over reads. FOR DEBUG ONLY.
0 force_low_pri_n	Active low signal. When asserted ('0'), all incoming transactions are forced to low priority. This implies that all High Priority Read (HPR) and Variable Priority Read commands (VPR) will be treated as Low Priority Read (LPR) commands. On the write side, all Variable Priority Write (VPW) commands will be treated as Normal Priority Write (NPW) commands. Forcing the incoming transactions to low priority implicitly turns off Bypass path for read commands. FOR PERFORMANCE ONLY.

9.3.3.1.76 Scheduler Control Register 1 (SCHED1)

9.3.3.1.76.1 Offset

Register	Offset
SCHED1	254h

9.3.3.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								pageclose_timer							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.76.3 Fields

Field	Function
31-8 —	Reserved
7-0 pageclose_timer	This field works in conjunction with SCHED.pageclose. It only has meaning if SCHED.pageclose==1. If SCHED.pageclose==1 and pageclose_timer==0, then an auto-precharge may be scheduled for last read or write command in the CAM with a bank and page hit. Note, sometimes an explicit precharge is scheduled instead of the auto-precharge. See SCHED.pageclose for details of when this may happen. If SCHED.pageclose==1 and pageclose_timer>0, then an auto-precharge is not scheduled for last read or write command in the CAM with a bank and page hit. Instead, a timer is started, with pageclose_timer as the initial value. There is a timer on a per bank basis. The timer decrements unless the next read or write in the CAM to a bank is a page hit. It gets reset to pageclose_timer value if the next read or write in the CAM to a bank is a page hit. Once the timer has reached zero, an explcit precharge will be attempted to be scheduled.

9.3.3.1.77 High Priority Read CAM Register 1 (PERFHPR1)

9.3.3.1.77.1 Offset

Register	Offset
PERFHPR1	25Ch

9.3.3.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	hpr_xact_run_length								Reserved							
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	hpr_max_starve															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

9.3.3.1.77.3 Fields

Field	Function
31-24	Number of transactions that are serviced once the HPR queue goes critical is the smaller of: - (a) This number - (b) Number of transactions available. Unit: Transaction. FOR PERFORMANCE ONLY.

Table continues on the next page...

Field	Function
hpr_xact_run_length	
23-16 —	Reserved
15-0 hpr_max_starve	Number of DFI clocks that the HPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality; during normal operation, this function should not be disabled as it will cause excessive latencies. FOR PERFORMANCE ONLY.

9.3.3.1.78 Low Priority Read CAM Register 1 (PERFLPR1)

9.3.3.1.78.1 Offset

Register	Offset
PERFLPR1	264h

9.3.3.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	lpr_xact_run_length								Reserved							
Reset	0	0	0	0	1	1	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	lpr_max_starve															
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

9.3.3.1.78.3 Fields

Field	Function
31-24 lpr_xact_run_length	Number of transactions that are serviced once the LPR queue goes critical is the smaller of: - (a) This number - (b) Number of transactions available. Unit: Transaction. FOR PERFORMANCE ONLY.
23-16 —	Reserved
15-0 lpr_max_starve	Number of DFI clocks that the LPR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality;

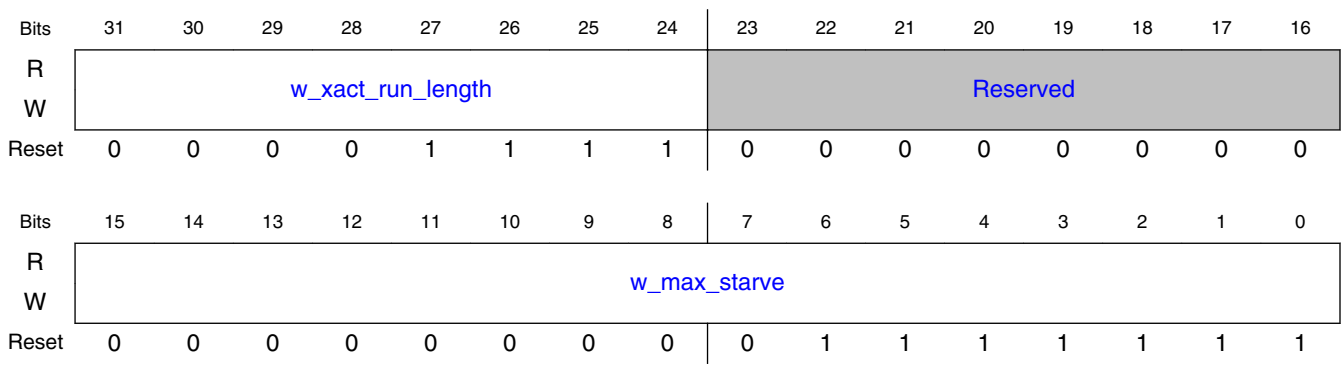
Field	Function
	during normal operation, this function should not be disabled as it will cause excessive latencies. FOR PERFORMANCE ONLY.

9.3.3.1.79 Write CAM Register 1 (PERFWR1)

9.3.3.1.79.1 Offset

Register	Offset
PERFWR1	26Ch

9.3.3.1.79.2 Diagram



9.3.3.1.79.3 Fields

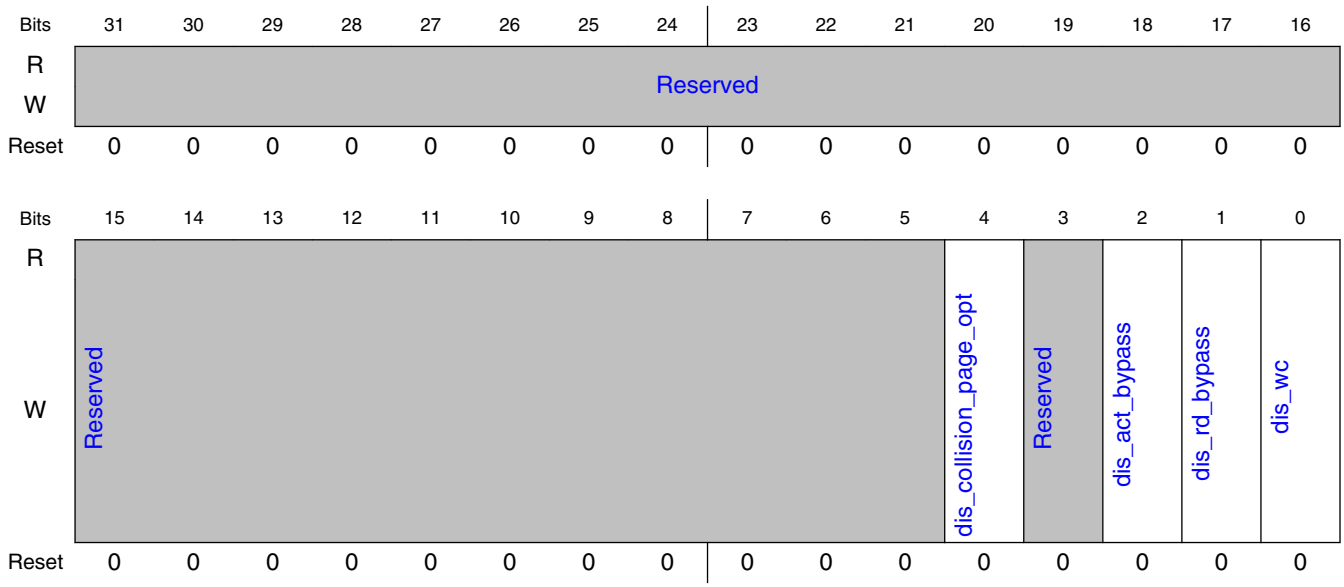
Field	Function
31-24 w_xact_run_length	Number of transactions that are serviced once the WR queue goes critical is the smaller of: - (a) This number - (b) Number of transactions available. Unit: Transaction. FOR PERFORMANCE ONLY.
23-16 —	Reserved
15-0 w_max_starve	Number of DFI clocks that the WR queue can be starved before it goes critical. The minimum valid functional value for this register is 0x1. Programming it to 0x0 will disable the starvation functionality; during normal operation, this function should not be disabled as it will cause excessive latencies. FOR PERFORMANCE ONLY.

9.3.3.1.80 Debug Register 0 (DBG0)

9.3.3.1.80.1 Offset

Register	Offset
DBG0	300h

9.3.3.1.80.2 Diagram



9.3.3.1.80.3 Fields

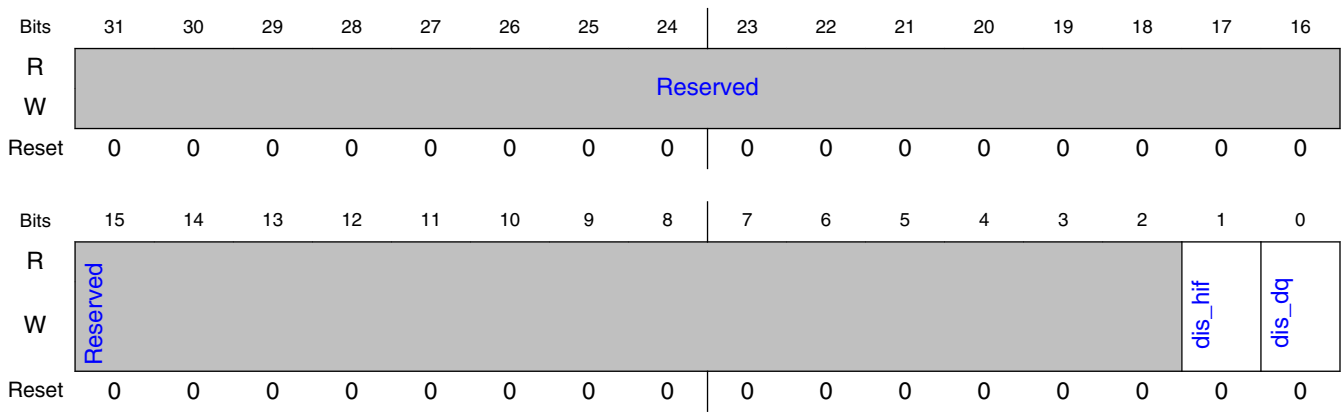
Field	Function
31-5 —	Reserved
4 dis_collision_page_opt	When this is set to '0', auto-precharge is disabled for the flushed command in a collision case. Collision cases are write followed by read to same address, read followed by write to same address, or write followed by write to same address with DBG0.dis_wc bit = 1 (where same address comparisons exclude the two address bits representing critical word). FOR DEBUG ONLY.
3 —	Reserved
2 dis_act_bypass	Only present in designs supporting activate bypass. When 1, disable bypass path for high priority read activates FOR DEBUG ONLY.
1 dis_rd_bypass	Only present in designs supporting read bypass. When 1, disable bypass path for high priority read page hits FOR DEBUG ONLY.
0 dis_wc	When 1, disable write combine. FOR DEBUG ONLY

9.3.3.1.81 Debug Register 1 (DBG1)

9.3.3.1.81.1 Offset

Register	Offset
DBG1	304h

9.3.3.1.81.2 Diagram



9.3.3.1.81.3 Fields

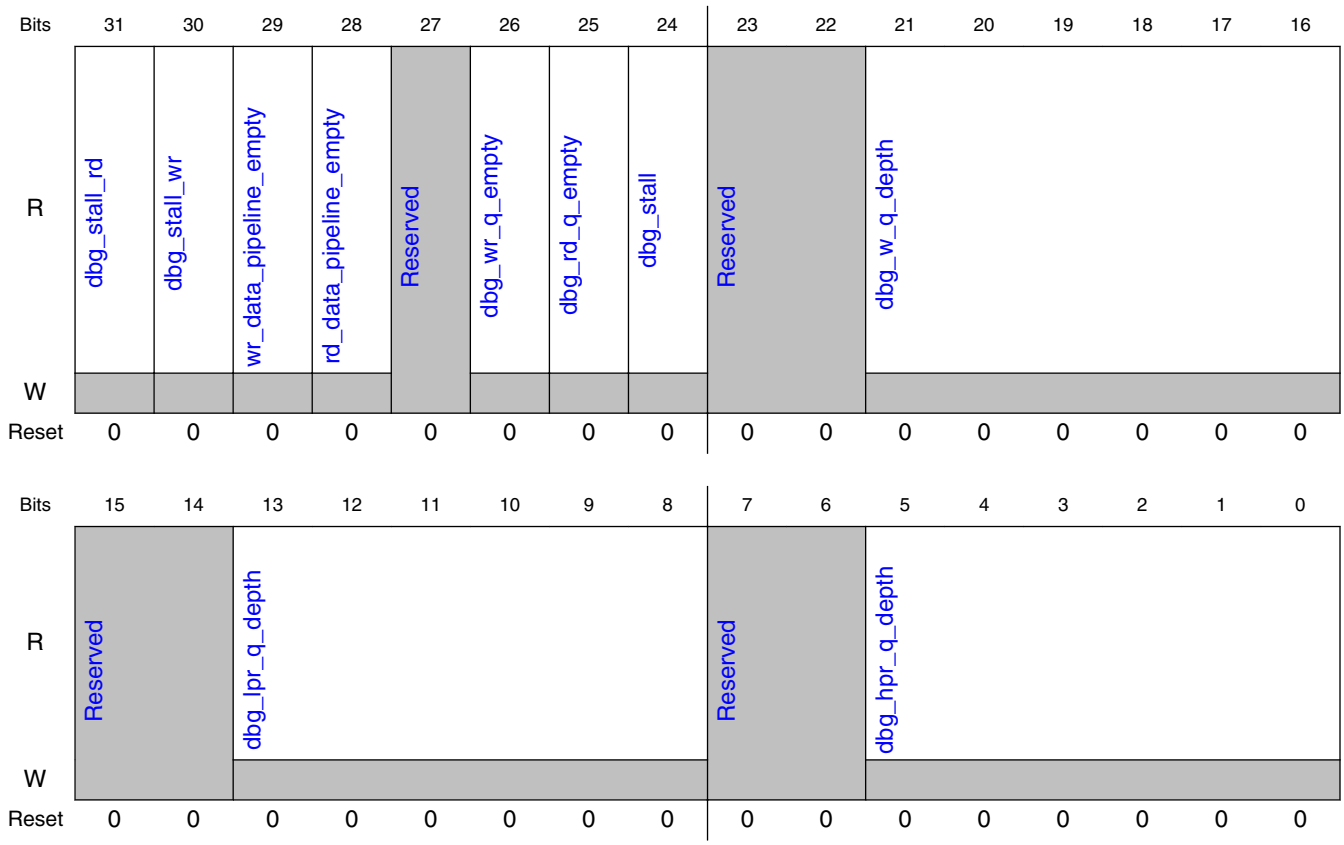
Field	Function
31-2 —	Reserved
1 dis_hif	When 1, DDRC asserts the HIF command signal hif_cmd_stall. DDRC will ignore the hif_cmd_valid and all other associated request signals. This bit is intended to be switched on-the-fly.
0 dis_dq	When 1, DDRC will not de-queue any transactions from the CAM. Bypass is also disabled. All transactions are queued in the CAM. No reads or writes are issued to SDRAM as long as this is asserted. This bit may be used to prevent reads or writes being issued by the DDRC, which makes it safe to modify certain register fields associated with reads and writes (see User Guide for details). After setting this bit, it is strongly recommended to poll DBGCAM.wr_data_pipeline_empty and DBGCAM.rd_data_pipeline_empty, before making changes to any registers which affect reads and writes. This will ensure that the relevant logic in the DDRC is idle. This bit is intended to be switched on-the-fly.

9.3.3.1.82 CAM Debug Register (DBGCAM)

9.3.3.1.82.1 Offset

Register	Offset
DBGCAM	308h

9.3.3.1.82.2 Diagram



9.3.3.1.82.3 Fields

Field	Function
31 dbg_stall_rd	Stall for Read channel FOR DEBUG ONLY
30 dbg_stall_wr	Stall for Write channel FOR DEBUG ONLY
29 wr_data_pipeline_empty	This bit indicates that the write data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed.

Table continues on the next page...

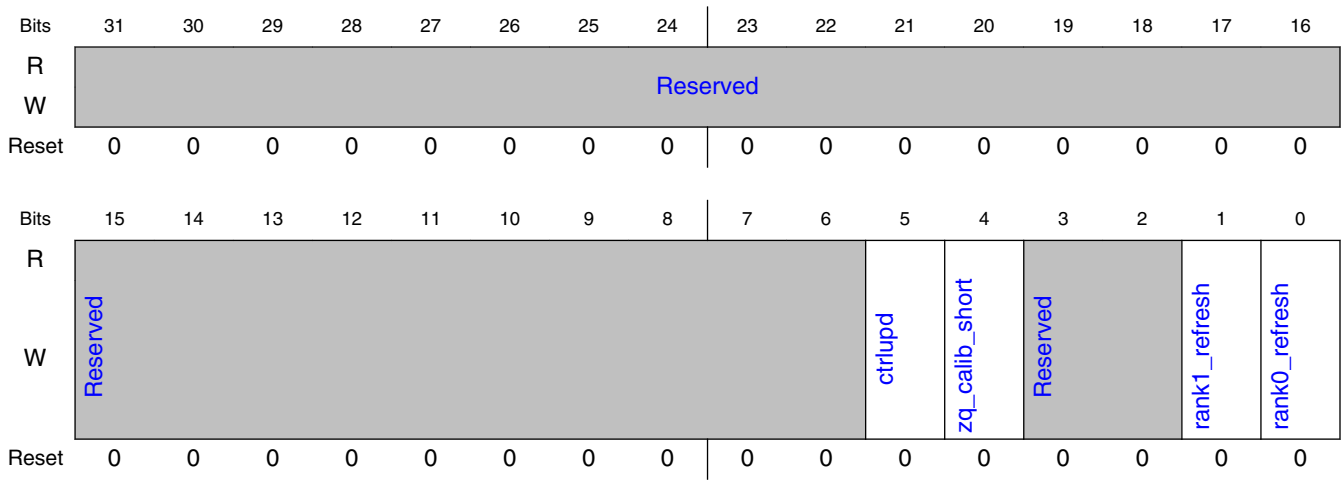
Field	Function
28 rd_data_pipeline_empty	This bit indicates that the read data pipeline on the DFI interface is empty. This register is intended to be polled at least twice after setting DBG1.dis_dq, to ensure that all remaining commands/data have completed.
27 —	Reserved
26 dbg_wr_q_empty	When 1, all the Write command queues and Write data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time. FOR DEBUG ONLY
25 dbg_rd_q_empty	When 1, all the Read command queues and Read data buffers inside DDRC are empty. This register is to be used for debug purpose. An example use-case scenario: When Controller enters Self-Refresh using the Low-Power entry sequence, Controller is expected to have executed all the commands in its queues and the write and read data drained. Hence this register should be 1 at that time. FOR DEBUG ONLY
24 dbg_stall	Stall FOR DEBUG ONLY
23-22 —	Reserved
21-16 dbg_w_q_depth	Write queue depth The last entry of WR queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY
15-14 —	Reserved
13-8 dbg_lpr_q_depth	Low priority read queue depth The last entry of Lpr queue is reserved for ECC SCRUB operation. This entry is not included in the calculation of the queue depth. FOR DEBUG ONLY
7-6 —	Reserved
5-0 dbg_hpr_q_depth	High priority read queue depth FOR DEBUG ONLY

9.3.3.1.83 Command Debug Register (DBGCMD)

9.3.3.1.83.1 Offset

Register	Offset
DBGCMD	30Ch

9.3.3.1.83.2 Diagram



9.3.3.1.83.3 Fields

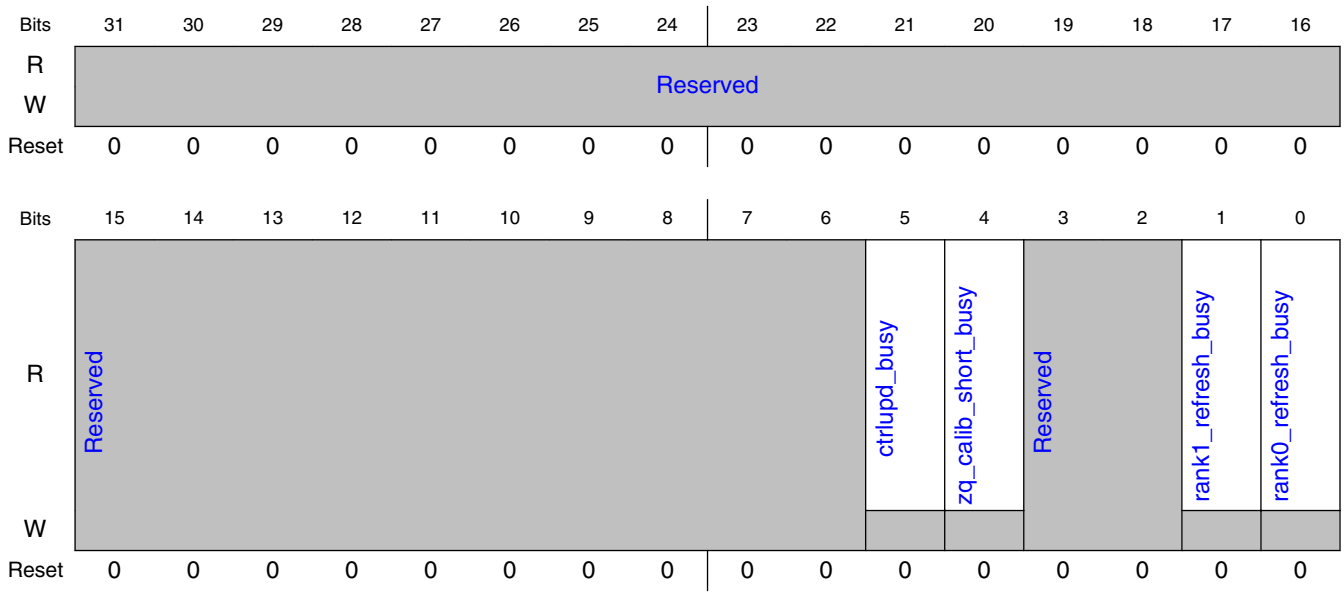
Field	Function
31-6 —	Reserved
5 ctrlupd	Setting this register bit to 1 indicates to the DDRC to issue a dfi_ctrlupd_req to the PHY. When this request is stored in the DDRC, the bit is automatically cleared. This operation must only be performed when DFIUPD0.dis_auto_ctrlupd=1.
4 zq_calib_short	Setting this register bit to 1 indicates to the DDRC to issue a ZQCS (ZQ calibration short)/MPC(ZQ calibration) command to the SDRAM. When this request is stored in the DDRC, the bit is automatically cleared. This operation can be performed only when ZQCTL0.dis_auto_zq=1. It is recommended NOT to set this register bit if in Init operating mode. This register bit is ignored when in Self-Refresh(except LPDDR4) and SR-Powerdown(LPDDR4) and Deep power-down operating modes and Maximum Power Saving Mode.
3-2 —	Reserved
1 rank1_refresh	Setting this register bit to 1 indicates to the DDRC to issue a refresh to rank 1. Writing to this bit causes DBGSTAT.rank1_refresh_busy to be set. When DBGSTAT.rank1_refresh_busy is cleared, the command has been stored in DDRC. For 3DS configuration, refresh is sent to rank index 1. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode.
0 rank0_refresh	Setting this register bit to 1 indicates to the DDRC to issue a refresh to rank 0. Writing to this bit causes DBGSTAT.rank0_refresh_busy to be set. When DBGSTAT.rank0_refresh_busy is cleared, the command has been stored in DDRC. For 3DS configuration, refresh is sent to rank index 0. This operation can be performed only when RFSHCTL3.dis_auto_refresh=1. It is recommended NOT to set this register bit if in Init or Deep power-down operating modes or Maximum Power Saving Mode.

9.3.3.1.84 Status Debug Register (DBGSTAT)

9.3.3.1.84.1 Offset

Register	Offset
DBGSTAT	310h

9.3.3.1.84.2 Diagram



9.3.3.1.84.3 Fields

Field	Function
31-6 —	Reserved
5 ctrlupd_busy	SoC core may initiate a ctrlupd operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ctrlupd request. It goes low when the ctrlupd operation is initiated in the DDRC. It is recommended not to perform ctrlupd operations when this signal is high. - 0 - Indicates that the SoC core can initiate a ctrlupd operation - 1 - Indicates that ctrlupd operation has not been initiated yet in the DDRC
4 zq_calib_short_busy	SoC core may initiate a ZQCS (ZQ calibration short) operation only if this signal is low. This signal goes high in the clock after the DDRC accepts the ZQCS request. It goes low when the ZQCS operation is initiated in the DDRC. It is recommended not to perform ZQCS operations when this signal is high. - 0 - Indicates that the SoC core can initiate a ZQCS operation - 1 - Indicates that ZQCS operation has not been initiated yet in the DDRC
3-2 —	Reserved
1 rank1_refresh_busy	SoC core may initiate a rank1_refresh operation (refresh operation to rank 1) only if this signal is low. This signal goes high in the clock after DBGCMD.rank1_refresh is set to one. It goes low when the rank1_refresh operation is stored in the DDRC. It is recommended not to perform rank1_refresh

Table continues on the next page...

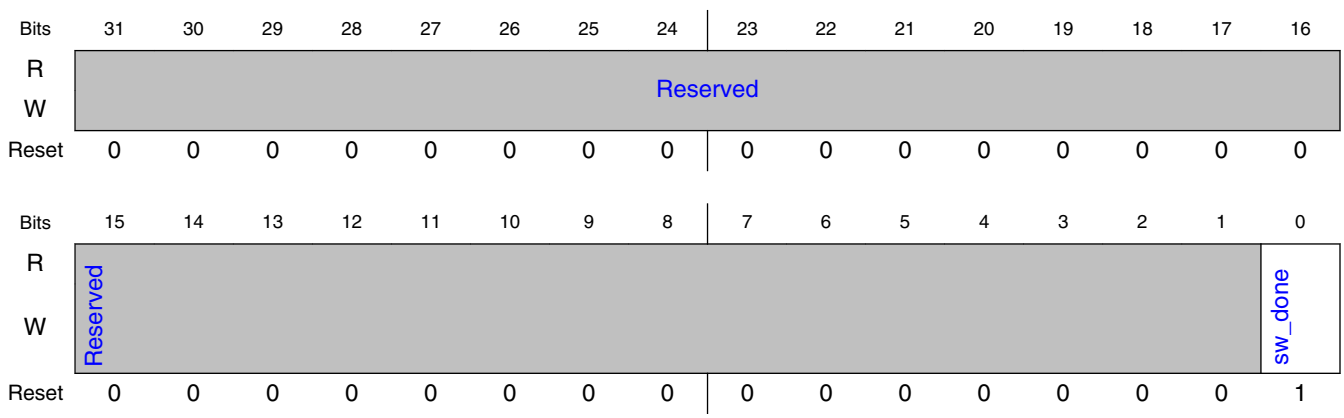
Field	Function
	operations when this signal is high. - 0 - Indicates that the SoC core can initiate a rank1_refresh operation - 1 - Indicates that rank1_refresh operation has not been stored yet in the DDRC
0 rank0_refresh_busy	SoC core may initiate a rank0_refresh operation (refresh operation to rank 0) only if this signal is low. This signal goes high in the clock after DBGCMD.rank0_refresh is set to one. It goes low when the rank0_refresh operation is stored in the DDRC. It is recommended not to perform rank0_refresh operations when this signal is high. - 0 - Indicates that the SoC core can initiate a rank0_refresh operation - 1 - Indicates that rank0_refresh operation has not been stored yet in the DDRC

9.3.3.1.85 Software Register Programming Control Enable (SWCTL)

9.3.3.1.85.1 Offset

Register	Offset
SWCTL	320h

9.3.3.1.85.2 Diagram



9.3.3.1.85.3 Fields

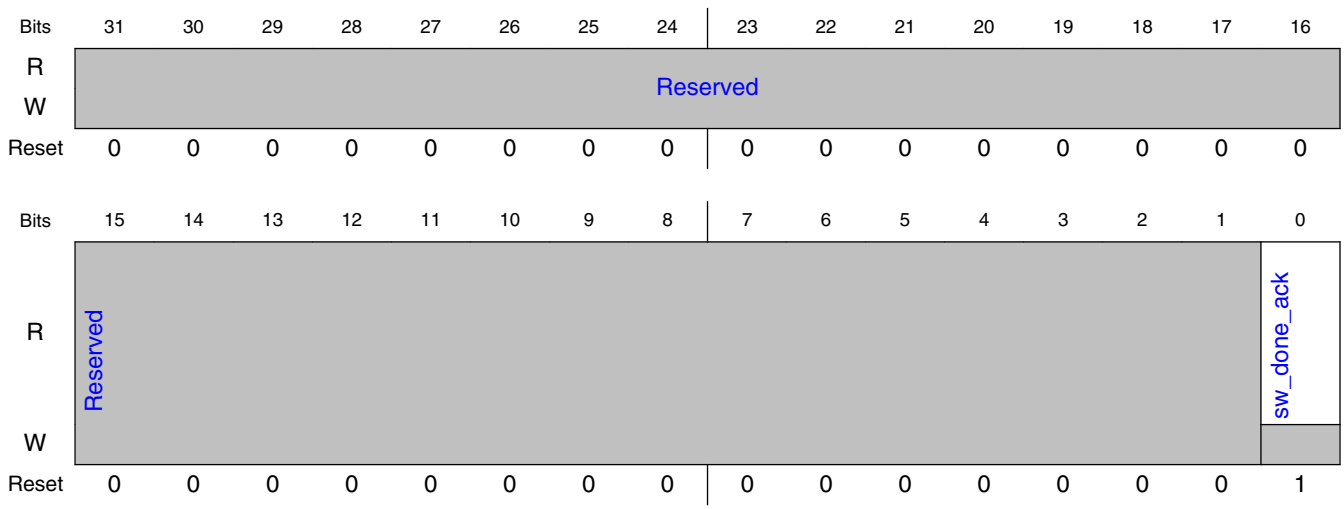
Field	Function
31-1 —	Reserved
0 sw_done	Enable quasi-dynamic register programming outside reset. Program register to 0 to enable quasi-dynamic programming. Set back register to 1 once programming is done.

9.3.3.1.86 Software Register Programming Control Status (SWSTAT)

9.3.3.1.86.1 Offset

Register	Offset
SWSTAT	324h

9.3.3.1.86.2 Diagram



9.3.3.1.86.3 Fields

Field	Function
31-1 —	Reserved
0 sw_done_ack	Register programming done. This register is the echo of SWCTL.sw_done. Wait for sw_done value 1 to propagate to sw_done_ack at the end of the programming sequence to ensure that the correct registers values are propagated to the destination clock domains.

9.3.3.1.87 AXI Poison Configuration Register. (POISONCFG)

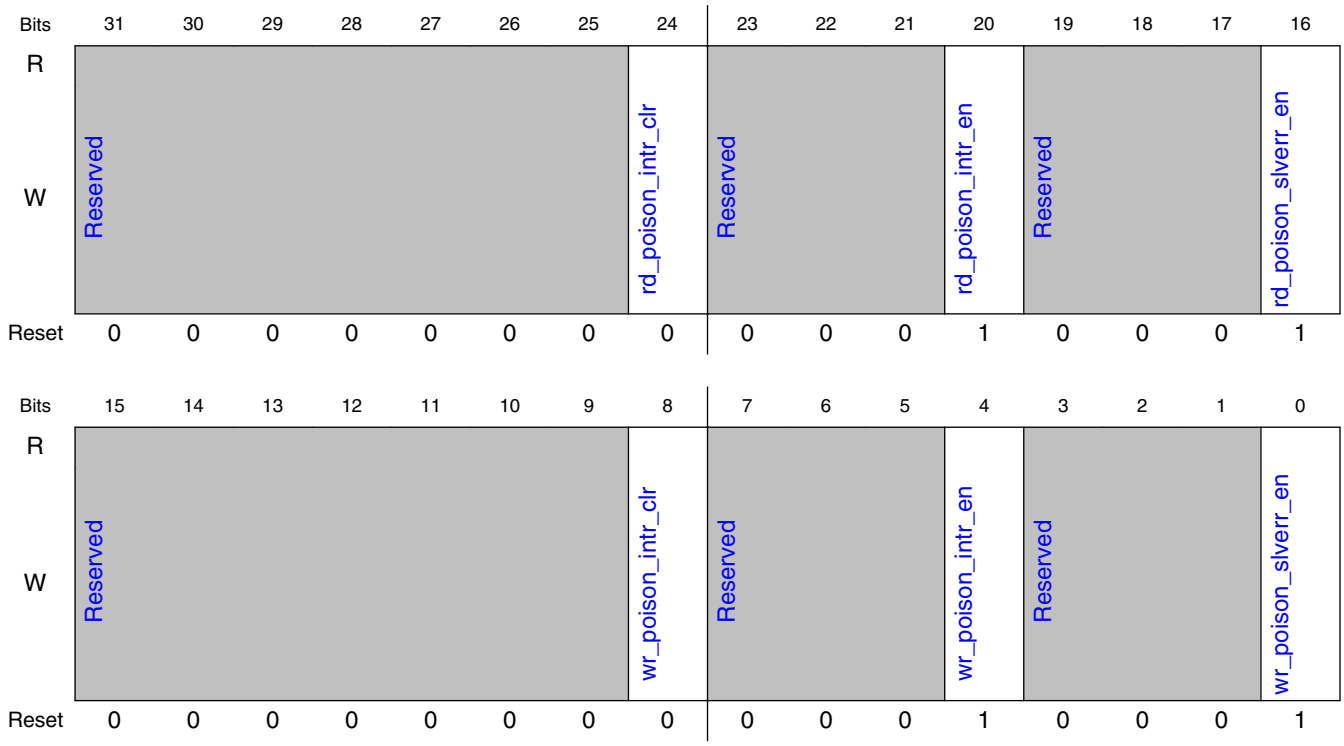
9.3.3.1.87.1 Offset

Register	Offset
POISONCFG	36Ch

9.3.3.1.87.2 Function

AXI Poison Configuration Register. Common for all AXI ports

9.3.3.1.87.3 Diagram



9.3.3.1.87.4 Fields

Field	Function
31-25 —	Reserved
24 rd_poison_intr_clr	Interrupt clear for read transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts.
23-21 —	Reserved
20 rd_poison_intr_en	If set to 1, enables interrupts for read transaction poisoning
19-17 —	Reserved
16	If set to 1, enables SLVERR response for read transaction poisoning

Table continues on the next page...

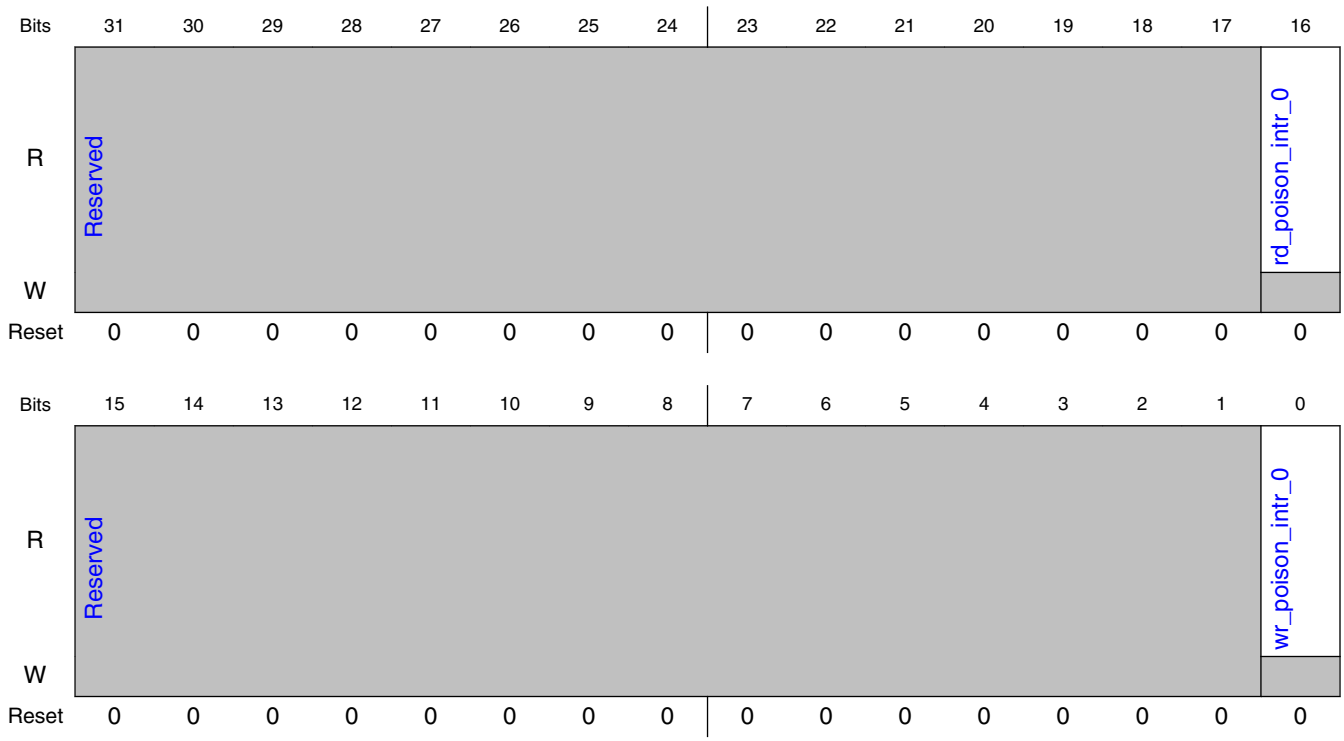
Field	Function
rd_poison_slverr_en	
15-9 —	Reserved
8 wr_poison_intr_clr	Interrupt clear for write transaction poisoning. Allow 2/3 clock cycles for correct value to propagate to core logic and clear the interrupts.
7-5 —	Reserved
4 wr_poison_intr_en	If set to 1, enables interrupts for write transaction poisoning
3-1 —	Reserved
0 wr_poison_slverr_en	If set to 1, enables SLVERR response for write transaction poisoning

9.3.3.1.88 AXI Poison Status Register (POISONSTAT)

9.3.3.1.88.1 Offset

Register	Offset
POISONSTAT	370h

9.3.3.1.88.2 Diagram



9.3.3.1.88.3 Fields

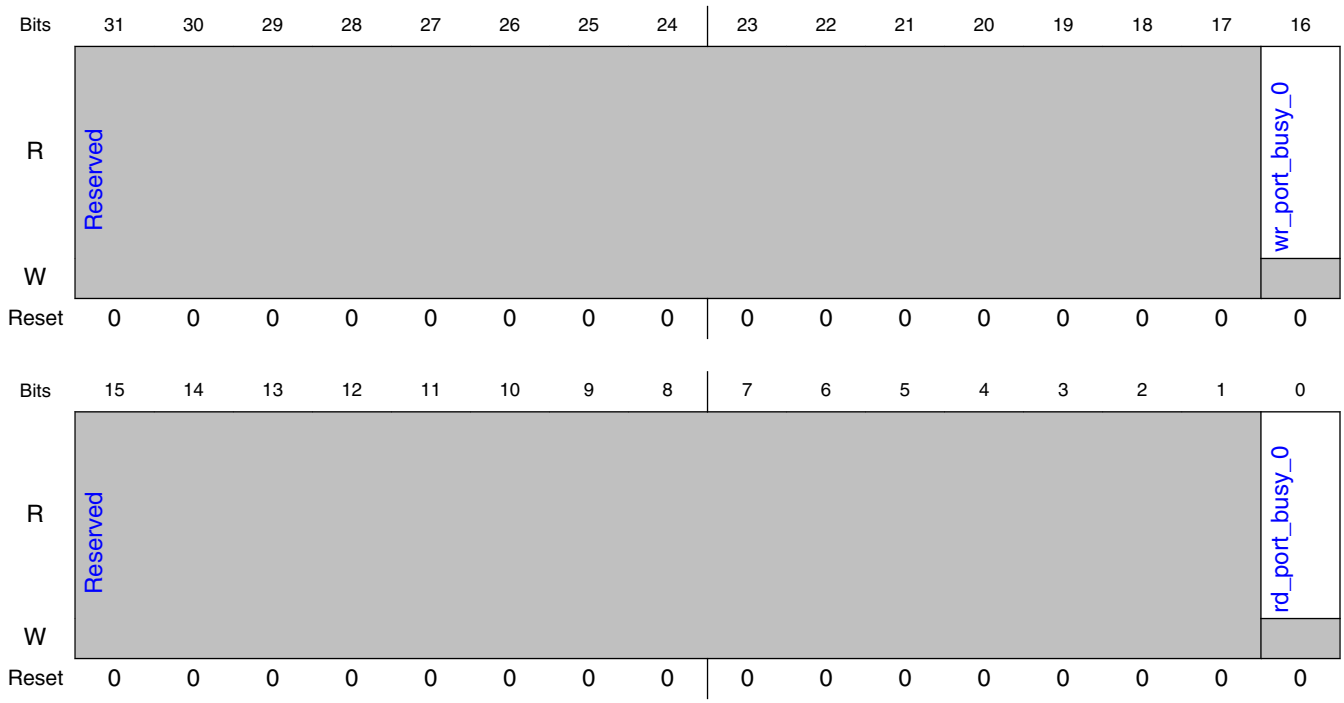
Field	Function
31-17 —	Reserved
16 rd_poison_intr_0	Read transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's read address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register rd_poison_intr_clr, then value propagated to APB clock.
15-1 —	Reserved
0 wr_poison_intr_0	Write transaction poisoning error interrupt for port 0. This register is a APB clock copy (double register synchronizer) of the interrupt asserted when a transaction is poisoned on the corresponding AXI port's write address channel. Bit 0 corresponds to Port 0, and so on. Interrupt is cleared by register wr_poison_intr_clr, then value propagated to APB clock.

9.3.3.1.89 Port Status Register (PSTAT)

9.3.3.1.89.1 Offset

Register	Offset
PSTAT	3FCh

9.3.3.1.89.2 Diagram



9.3.3.1.89.3 Fields

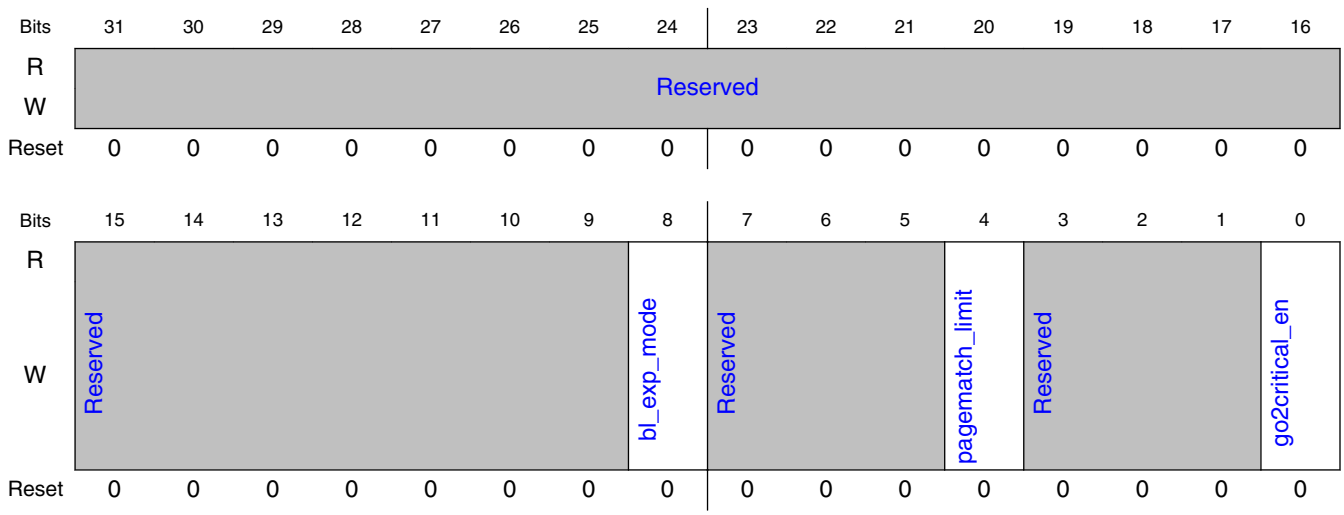
Field	Function
31-17 —	Reserved
16 wr_port_busy_0	Indicates if there are outstanding writes for AXI port 0.
15-1 —	Reserved
0 rd_port_busy_0	Indicates if there are outstanding reads for AXI port 0.

9.3.3.1.90 Port Common Configuration Register (PCCFG)

9.3.3.1.90.1 Offset

Register	Offset
PCCFG	400h

9.3.3.1.90.2 Diagram



9.3.3.1.90.3 Fields

Field	Function
31-9 —	Reserved
8 bl_exp_mode	Burst length expansion mode. By default (i.e. bl_exp_mode==0) XPI expands every AXI burst into multiple HIF commands, using the memory burst length as a unit. If set to 1, then XPI will use half of the memory burst length as a unit. This applies to both reads and writes. When MSTR.data_bus_width==00, setting bl_exp_mode to 1 has no effect. This can be used in cases where Partial Writes is enabled (DDRC_PARTIAL_WR=1), in order to avoid or minimize t_ccd_l penalty in DDR4 and t_ccd_mw penalty in LPDDR4. Hence, bl_exp_mode=1 is only recommended if DDR4 or LPDDR4. Note that if DBICTL.reg_ddrc_dm_en=0, functionality is not supported in the following cases: - DDRC_PARTIAL_WR=0 - DDRC_PARTIAL_WR=1, MSTR.data_bus_width=01, MEMC_BURST_LENGTH=8 and MSTR.burst_rdwr=1000 (LPDDR4 only) - DDRC_PARTIAL_WR=1, MSTR.data_bus_width=01, MEMC_BURST_LENGTH=4 and MSTR.burst_rdwr=0100 (DDR4 only), with either MSTR.reg_ddrc_burstchop=0 or CRCPARCTL1.reg_ddrc_crc_enable=1 Functionality is also not supported if Data Channel Interleave is enabled
7-5 —	Reserved

Table continues on the next page...

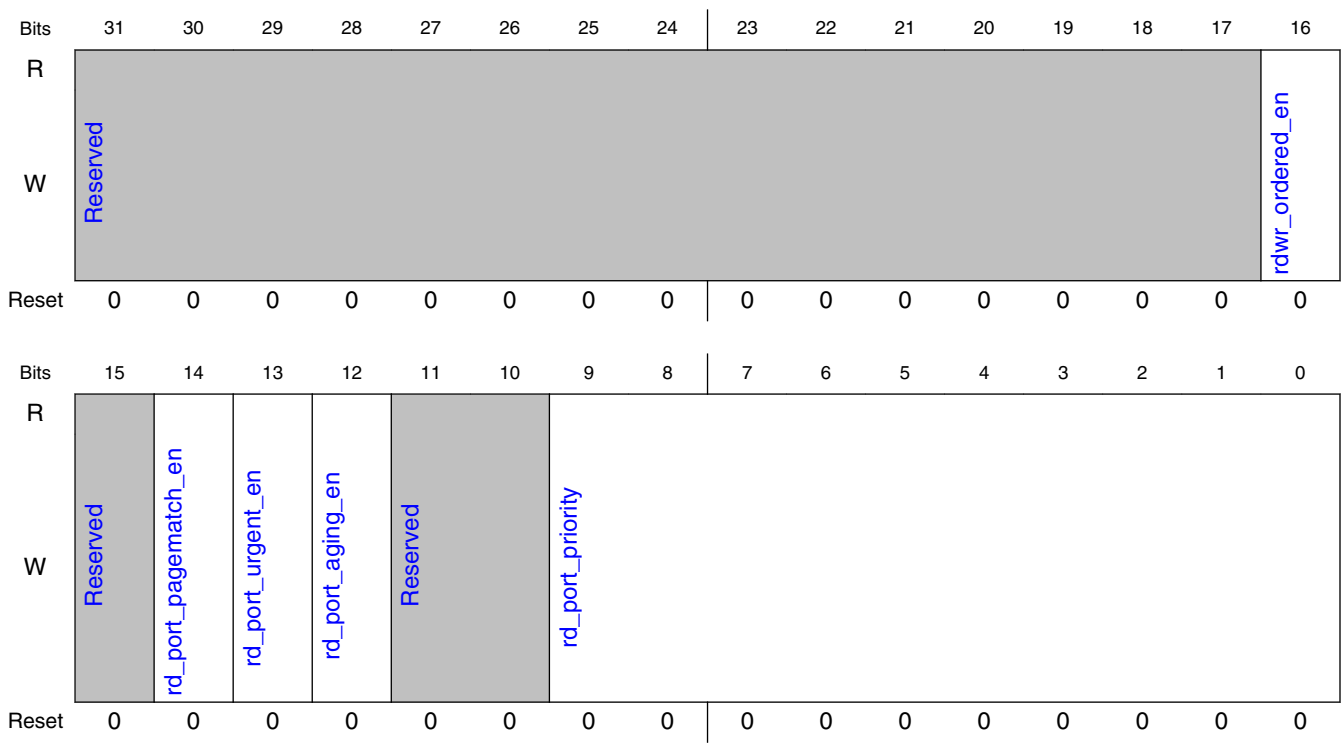
Field	Function
4 pagematch_limit	Page match four limit. If set to 1, limits the number of consecutive same page DDRC transactions that can be granted by the Port Arbiter to four when Page Match feature is enabled. If set to 0, there is no limit imposed on number of consecutive same page DDRC transactions.
3-1 —	Reserved
0 go2critical_en	If set to 1 (enabled), sets co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals going to DDRC based on urgent input (awurgent, arurgent) coming from AXI master. If set to 0 (disabled), co_gs_go2critical_wr and co_gs_go2critical_lpr/co_gs_go2critical_hpr signals at DDRC are driven to 1b'0.

9.3.3.1.91 Port n Configuration Read Register (PCFGR_0)

9.3.3.1.91.1 Offset

Register	Offset
PCFGR_0	404h

9.3.3.1.91.2 Diagram



9.3.3.1.91.3 Fields

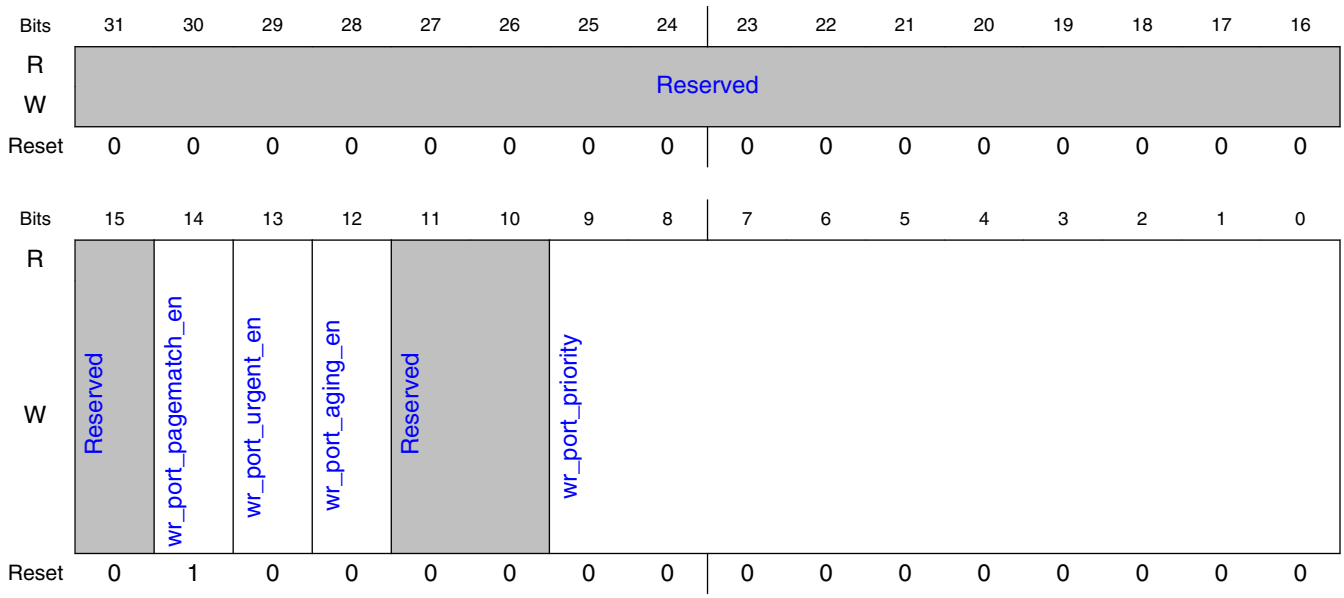
Field	Function
31-17 —	Reserved
16 rdwr_ordered_en	Enable ordered read/writes. If set to 1, preserves the ordering between read transaction and write transaction issued to the same address, on a given port. In other words, the controller ensures that all same address read and write commands from the application port interface are transported to the DFI interface in the order of acceptance. This feature is useful in cases where software coherency is desired for masters issuing back-to-back read/write transactions without waiting for write/read responses. Note that this register has an effect only if necessary logic is instantiated via the DDRC_RDWR_ORDERED_n parameter.
15 —	Reserved
14 rd_port_pagematch_en	If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also related PCCFG.pagematch_limit register.
13 rd_port_urgent_en	If set to 1, enables the AXI urgent sideband signal (arurgent). When enabled and arurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_lpr/co_gs_go2critical_hpr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that arurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
12 rd_port_aging_en	If set to 1, enables aging function for the read channel of the port.
11-10 —	Reserved
9-0 rd_port_priority	Determines the initial load value of read aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the read aging counter sets the priority of the read channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level (timeout condition - Priority0). For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (arqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become 0) to force read-write direction switching. In this case, external dynamic priority input, arqos (for reads only) can still be used to set the DDRC read priority (2 priority levels: low priority read - LPR, high priority read - HPR) on a command by command basis. Note: The two LSBs of this register field are tied internally to 2'b00.

9.3.3.1.92 Port n Configuration Write Register (PCFGW_0)

9.3.3.1.92.1 Offset

Register	Offset
PCFGW_0	408h

9.3.3.1.92.2 Diagram



9.3.3.1.92.3 Fields

Field	Function
31-15 —	Reserved
14 wr_port_pagematch_en	If set to 1, enables the Page Match feature. If enabled, once a requesting port is granted, the port is continued to be granted if the following immediate commands are to the same memory page (same bank and same row). See also related PCCFG.pagematch_limit register.
13 wr_port_urgent_en	If set to 1, enables the AXI urgent sideband signal (awurgent). When enabled and awurgent is asserted by the master, that port becomes the highest priority and co_gs_go2critical_wr signal to DDRC is asserted if enabled in PCCFG.go2critical_en register. Note that awurgent signal can be asserted anytime and as long as required which is independent of address handshaking (it is not associated with any particular command).
12 wr_port_aging_en	If set to 1, enables aging function for the write channel of the port.
11-10 —	Reserved
9-0 wr_port_priority	Determines the initial load value of write aging counters. These counters will be parallel loaded after reset, or after each grant to the corresponding port. The aging counters down-count every clock cycle where the port is requesting but not granted. The higher significant 5-bits of the write aging counter sets the initial priority of the write channel of a given port. Port's priority will increase as the higher significant 5-bits of the counter starts to decrease. When the aging counter becomes 0, the corresponding port channel will have the highest priority level. For multi-port configurations, the aging counters cannot be used to set port priorities when external dynamic priority inputs (awqos) are enabled (timeout is still applicable). For single port configurations, the aging counters are only used when they timeout (become

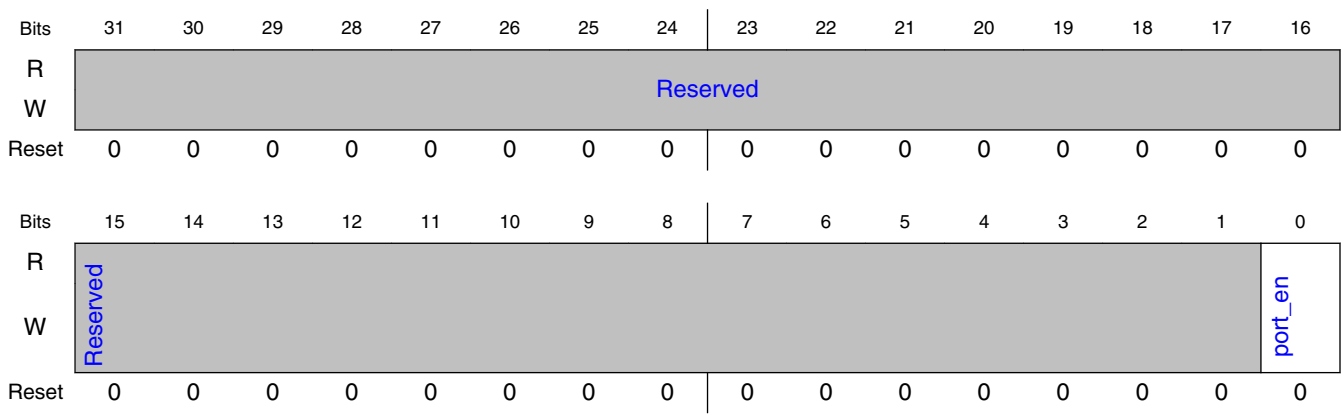
Field	Function
	0) to force read-write direction switching. Note: The two LSBs of this register field are tied internally to 2'b00.

9.3.3.1.93 Port n Control Register (PCTRL_0)

9.3.3.1.93.1 Offset

Register	Offset
PCTRL_0	490h

9.3.3.1.93.2 Diagram



9.3.3.1.93.3 Fields

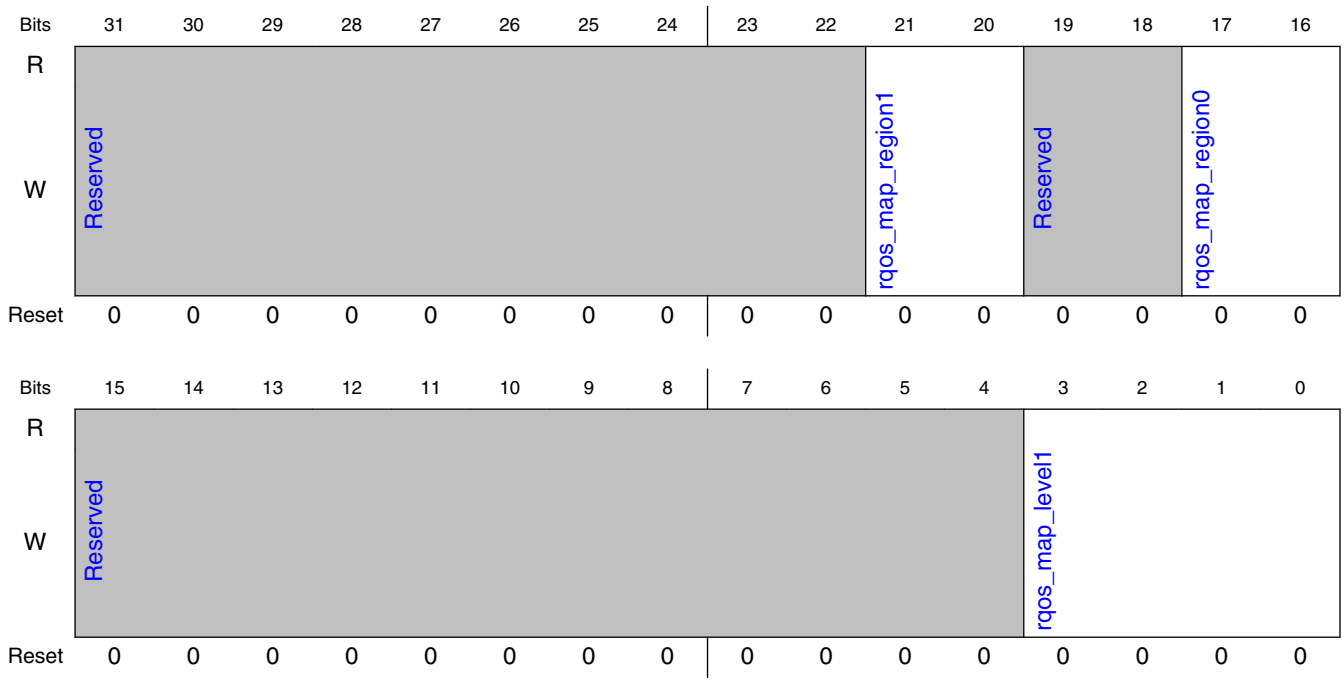
Field	Function
31-1 —	Reserved
0 port_en	Enables AXI port n.

9.3.3.1.94 Port n Read QoS Configuration Register 0 (PCFGQOS0_0)

9.3.3.1.94.1 Offset

Register	Offset
PCFGQOS0_0	494h

9.3.3.1.94.2 Diagram



9.3.3.1.94.3 Fields

Field	Function
31-22 —	Reserved
21-20 rqos_map_region1	This bitfield indicates the traffic class of region 1. Valid values are: 0 : LPR, 1: VPR, 2: HPR. For dual address queue configurations, region1 maps to the blue address queue. In this case, valid values are 0: LPR and 1: VPR only. When VPR support is disabled (DDRC_VPR_EN = 0) and traffic class of region 1 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
19-18 —	Reserved
17-16 rqos_map_region0	This bitfield indicates the traffic class of region 0. Valid values are: 0: LPR, 1: VPR, 2: HPR. For dual address queue configurations, region 0 maps to the blue address queue. In this case, valid values are: 0: LPR and 1: VPR only. When VPR support is disabled (DDRC_VPR_EN = 0) and traffic class of region0 is set to 1 (VPR), VPR traffic is aliased to LPR traffic.
15-4 —	Reserved

Table continues on the next page...

Field	Function
3-0 rqos_map_level1	Separation level1 indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 13 (for dual RAQ) or 0 to 14 (for single RAQ) which corresponds to arqos. Note that for PA, arqos values are used directly as port priorities, where the higher the value corresponds to higher port priority. All of the map_level* registers must be set to distinct values.

9.3.3.1.95 Port n Read QoS Configuration Register 1 (PCFGQOS1_0)

9.3.3.1.95.1 Offset

Register	Offset
PCFGQOS1_0	498h

9.3.3.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved					rqos_map_timeoutr										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					rqos_map_timeoutb										
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.95.3 Fields

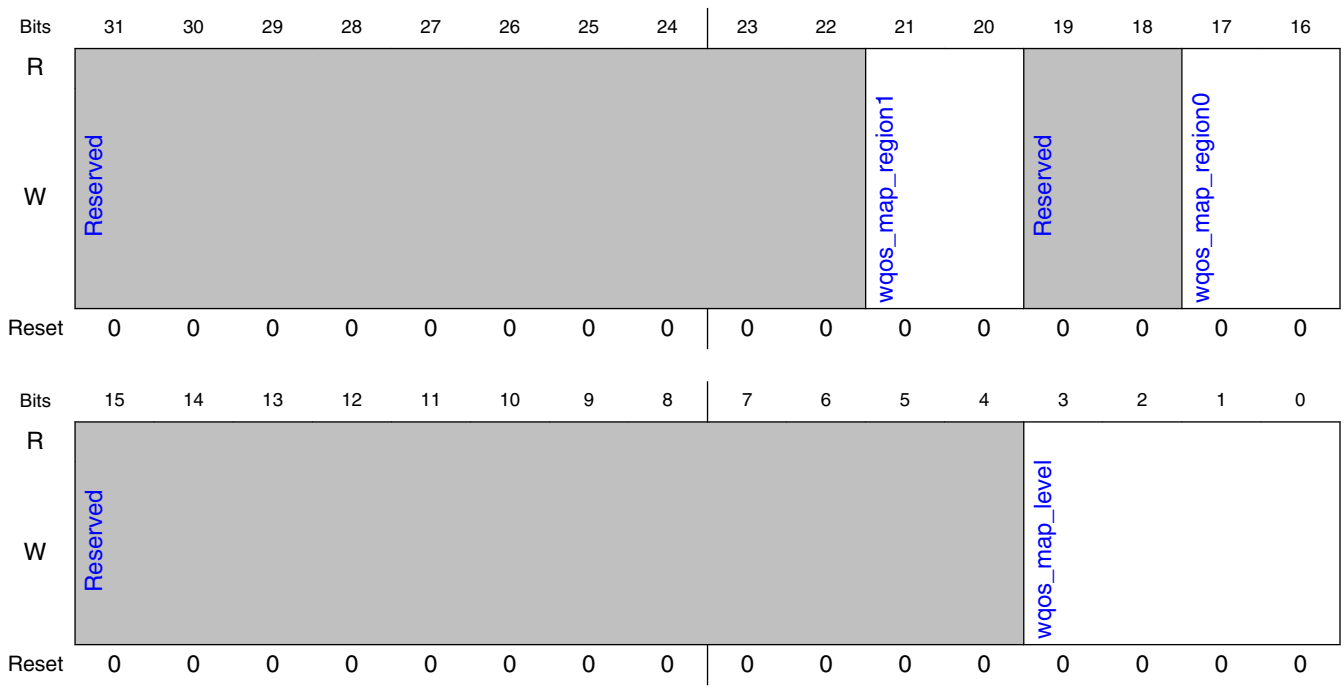
Field	Function
31-27 —	Reserved
26-16 rqos_map_timeoutr	Specifies the timeout value for transactions mapped to the red address queue.
15-11 —	Reserved
10-0 rqos_map_timeoutb	Specifies the timeout value for transactions mapped to the blue address queue.

9.3.3.1.96 Port n Write QoS Configuration Register 0 (PCFGWQOS0_0)

9.3.3.1.96.1 Offset

Register	Offset
PCFGWQOS0_0	49Ch

9.3.3.1.96.2 Diagram



9.3.3.1.96.3 Fields

Field	Function
31-22 —	Reserved
21-20 wqos_map_region1	This bitfield indicates the traffic class of region 1. Valid values are: 0: NPW, 1: VPW. When VPW support is disabled (DDRC_VPW_EN = 0) and traffic class of region 1 is set to 1 (VPW), VPW traffic is aliased to LPW traffic.
19-18 —	Reserved

Table continues on the next page...

Field	Function
17-16 wqos_map_region0	This bitfield indicates the traffic class of region 0. Valid values are: 0: NPW, 1: VPW. When VPW support is disabled (DDRC_VPW_EN = 0) and traffic class of region0 is set to 1 (VPW), VPW traffic is aliased to NPW traffic.
15-4 —	Reserved
3-0 wqos_map_level	Separation level indicating the end of region0 mapping; start of region0 is 0. Possible values for level1 are 0 to 14 which corresponds to awqos. Note that for PA, awqos values are used directly as port priorities, where the higher the value corresponds to higher port priority.

9.3.3.1.97 Port n Write QoS Configuration Register 1 (PCFGWQOS1_0)

9.3.3.1.97.1 Offset

Register	Offset
PCFGWQOS1_0	4A0h

9.3.3.1.97.2 Diagram

9.3.3.1.97.3 Fields

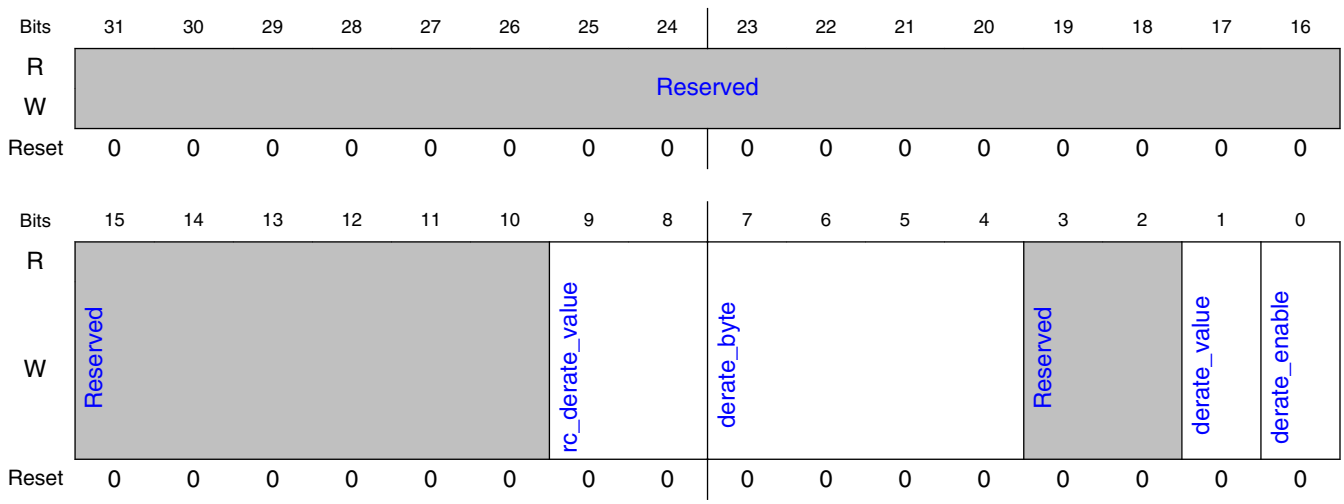
Field	Function
31-11 —	Reserved
10-0 wqos_map_timeout	Specifies the timeout value for write transactions.

9.3.3.1.98 [SHADOW] Temperature Derate Enable Register (DERATEEN_SHADOW)

9.3.3.1.98.1 Offset

Register	Offset
DERATEEN_SHADOW	2020h

9.3.3.1.98.2 Diagram



9.3.3.1.98.3 Fields

Field	Function
31-10 —	Reserved
9-8 rc_derate_value	Derate value of tRC for LPDDR4 - 0 - Derating uses +1. - 1 - Derating uses +2. - 2 - Derating uses +3. - 3 - Derating uses +4. Present only in designs configured to support LPDDR4. The required number of cycles for derating can be determined by dividing 3.75ns by the core_ddrc_core_clk period, and rounding up the next integer.
7-4 derate_byte	Derate byte Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 Indicates which byte of the MRR data is used for derating. The maximum valid value depends on MEMC_DRAM_TOTAL_DATA_WIDTH.
3-2 —	Reserved
1 derate_value	Derate value - 0 - Derating uses +1. - 1 - Derating uses +2. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 Set to 0 for all LPDDR2 speed grades as derating value of +1.875 ns is less than a core_ddrc_core_clk period. For LPDDR3/4, if the period of core_ddrc_core_clk is less than 1.875ns, this register field should be set to 1; otherwise it should be set to 0.

Table continues on the next page...

Field	Function
0 derate_enable	Enables derating - 0 - Timing parameter derating is disabled - 1 - Timing parameter derating is enabled using MR4 read value. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4 This field must be set to '0' for non-LPDDR2/LPDDR3/LPDDR4 mode.

9.3.3.1.99 [SHADOW] Temperature Derate Interval Register (DERATEINT_SHADOW)

9.3.3.1.99.1 Offset

Register	Offset
DERATEINT_SHADOW	2024h

9.3.3.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr4_read_interval															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.99.3 Fields

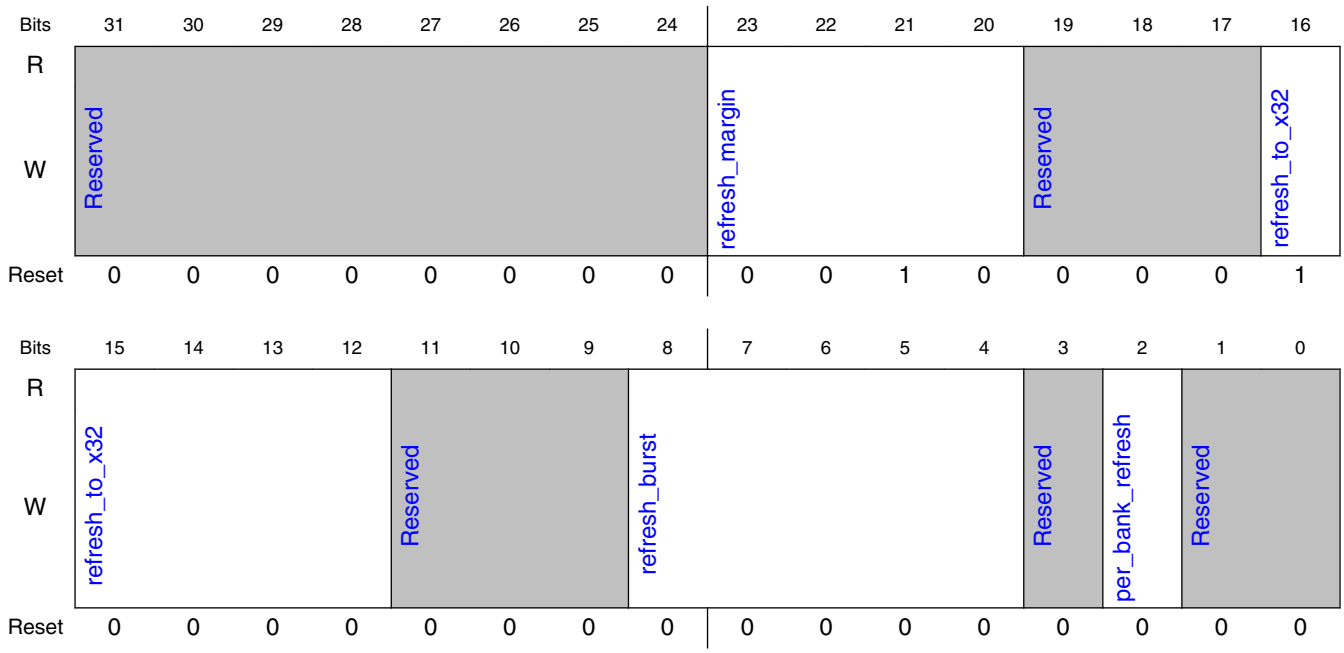
Field	Function
31-0 mr4_read_interval	Interval between two MR4 reads, used to derate the timing parameters. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4. This register must not be set to zero. Unit: DFI clock cycle.

9.3.3.1.100 [SHADOW] Refresh Control Register 0 (RFSHCTL0_SHADOW)

9.3.3.1.100.1 Offset

Register	Offset
RFSHCTL0_SHADOW	2050h

9.3.3.1.100.2 Diagram



9.3.3.1.100.3 Fields

Field	Function
31-24 —	Reserved
23-20 refresh_margin	Threshold value in number of DFI clock cycles before the critical refresh or page timer expires. A critical refresh is to be issued before this threshold is reached. It is recommended that this not be changed from the default value, currently shown as 0x2. It must always be less than internally used t_rfc_nom_x32. Note that, in LPDDR2/LPDDR3/LPDDR4, internally used t_rfc_nom_x32 may be equal to RFSHTMG.t_rfc_nom_x32>>2 if derating is enabled (DERATEEN.derate_enable=1). Otherwise, internally used t_rfc_nom_x32 will be equal to RFSHTMG.t_rfc_nom_x32. Unit: Multiples of 32 DFI clocks.
19-17 —	Reserved
16-12 refresh_to_x32	If the refresh timer (tRFCnom, also known as tREFI) has expired at least once, but it has not expired (RFSHCTL0.refresh_burst+1) times yet, then a speculative refresh may be performed. A speculative refresh is a refresh performed at a time when refresh would be useful, but before it is absolutely required. When the SDRAM bus is idle for a period of time determined by this RFSHCTL0.refresh_to_x32 and the refresh timer has expired at least once since the last refresh, then a speculative refresh is performed.

Table continues on the next page...

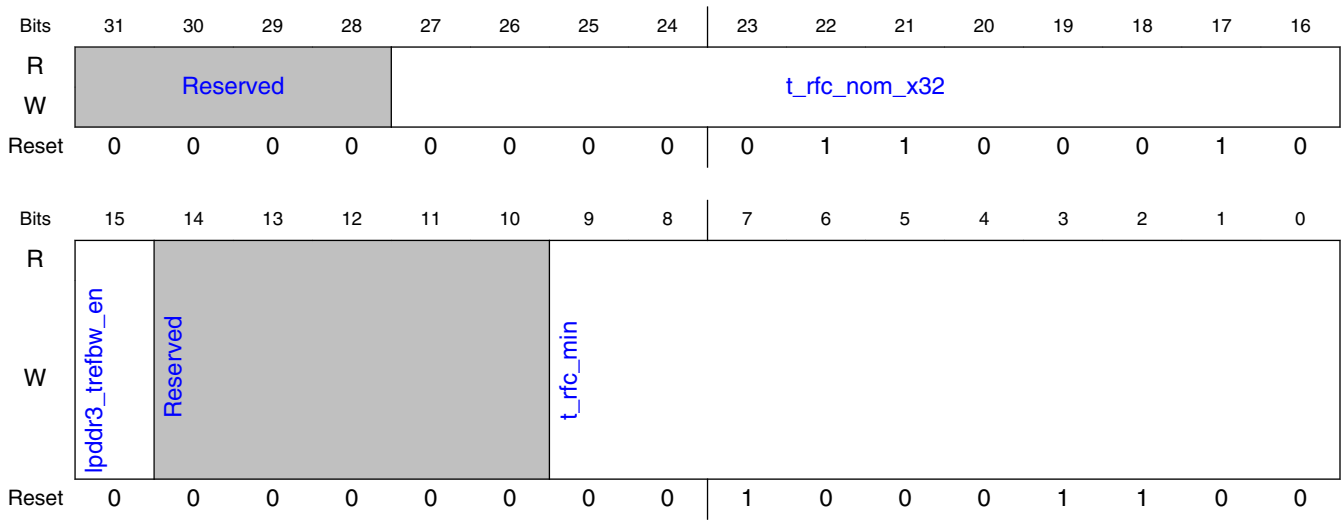
Field	Function
	Speculative refreshes continues successively until there are no refreshes pending or until new reads or writes are issued to the DDRC. FOR PERFORMANCE ONLY. Unit: Multiples of 32 DFI clocks.
11-9 —	Reserved
8-4 refresh_burst	The programmed value + 1 is the number of refresh timeouts that is allowed to accumulate before traffic is blocked and the refreshes are forced to execute. Closing pages to perform a refresh is a one-time penalty that must be paid for each group of refreshes. Therefore, performing refreshes in a burst reduces the per-refresh penalty of these page closings. Higher numbers for RFSHCTL.refresh_burst slightly increases utilization; lower numbers decreases the worst-case latency associated with refreshes. - 0 - single refresh - 1 - burst-of-2 refresh - 7 - burst-of-8 refresh For information on burst refresh feature refer to section 3.9 of DDR2 JEDEC specification - JESD79-2F.pdf. For DDR2/3, the refresh is always per-rank and not per-bank. The rank refresh can be accumulated over 8*tREFI cycles using the burst refresh feature. In DDR4 mode, according to Fine Granularity feature, 8 refreshes can be postponed in 1X mode, 16 refreshes in 2X mode and 32 refreshes in 4X mode. If using PHY-initiated updates, care must be taken in the setting of RFSHCTL0.refresh_burst, to ensure that tRFCmax is not violated due to a PHY-initiated update occurring shortly before a refresh burst was due. In this situation, the refresh burst will be delayed until the PHY-initiated update is complete.
3 —	Reserved
2 per_bank_refresh	- 1 - Per bank refresh; - 0 - All bank refresh. Per bank refresh allows traffic to flow to other banks. Per bank refresh is not supported by all LPDDR2 devices but should be supported by all LPDDR3/LPDDR4 devices. Present only in designs configured to support LPDDR2/LPDDR3/LPDDR4
1-0 —	Reserved

9.3.3.1.101 [SHADOW] Refresh Timing Register (RFSHTMG_SHADOW)

9.3.3.1.101.1 Offset

Register	Offset
RFSHTMG_SHADOW	2064h

9.3.3.1.101.2 Diagram



9.3.3.1.101.3 Fields

Field	Function
31-28 —	Reserved
27-16 t_rfc_nom_x32	tREFI: Average time interval between refreshes per rank (Specification: 7.8us for DDR2, DDR3 and DDR4. See JEDEC specification for mDDR, LPDDR2, LPDDR3 and LPDDR4). For LPDDR2/LPDDR3/LPDDR4: - if using all-bank refreshes (RFSHCTL0.per_bank_refresh = 0), this register should be set to tREFIab - if using per-bank refreshes (RFSHCTL0.per_bank_refresh = 1), this register should be set to tREFIpb When the controller is operating in 1:2 frequency ratio mode, program this to (tREFI/2), no rounding up. In DDR4 mode, tREFI value is different depending on the refresh mode. The user should program the appropriate value from the spec based on the value programmed in the refresh mode register. Note that RFSHTMG.t_rfc_nom_x32 * 32 must be greater than RFSHTMG.t_rfc_min, and RFSHTMG.t_rfc_nom_x32 must be greater than 0x1. - Non-DDR4 or DDR4 Fixed 1x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0xFFE. - DDR4 Fixed 2x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x7FF. - DDR4 Fixed 4x mode: RFSHTMG.t_rfc_nom_x32 must be less than or equal to 0x3FF. Unit: Multiples of 32 clocks.
15 lpddr3_trefbw_en	Used only when LPDDR3 memory type is connected. Should only be changed when DDRC is in reset. Specifies whether to use the tREFBW parameter (required by some LPDDR3 devices which comply with earlier versions of the LPDDR3 JEDEC specification) or not: - 0 - tREFBW parameter not used - 1 - tREFBW parameter used
14-10 —	Reserved
9-0 t_rfc_min	tRFC (min): Minimum time from refresh to refresh or activate. When the controller is operating in 1:1 mode, t_rfc_min should be set to RoundUp(tRFCmin/tCK). When the controller is operating in 1:2 mode, t_rfc_min should be set to RoundUp(RoundUp(tRFCmin/tCK)/2). In LPDDR2/LPDDR3/LPDDR4 mode: - if using all-bank refreshes, the tRFCmin value in the above equations is equal to tRFCab - if using per-bank refreshes, the tRFCmin value in the above equations is equal to tRFCpb In DDR4 mode, the tRFCmin value in the above equations is different depending on the refresh mode (fixed 1X,2X,4X) and the device density. The user should program the appropriate value from the spec based on the 'refresh_mode' and the device density that is used. Unit: Clocks.

9.3.3.1.102 [SHADOW] SDRAM Initialization Register 3 (INIT3_SHADOW)

9.3.3.1.102.1 Offset

Register	Offset
INIT3_SHADOW	20DCh

9.3.3.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr															
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

9.3.3.1.102.3 Fields

Field	Function
31-16 mr	DDR2: Value to write to MR register. Bit 8 is for DLL and the setting here is ignored. The DDRC sets this bit appropriately. DDR3/DDR4: Value loaded into MR0 register. mDDR: Value to write to MR register. LPDDR2/LPDDR3/LPDDR4 - Value to write to MR1 register
15-0 emr	DDR2: Value to write to EMR register. Bits 9:7 are for OCD and the setting in this register is ignored. The DDRC sets those bits appropriately. DDR3/DDR4: Value to write to MR1 register Set bit 7 to 0. If PHY-evaluation mode training is enabled, this bit is set appropriately by the DDRC during write leveling. mDDR: Value to write to EMR register. LPDDR2/LPDDR3/LPDDR4 - Value to write to MR2 register

9.3.3.1.103 [SHADOW] SDRAM Initialization Register 4 (INIT4_SHADOW)

9.3.3.1.103.1 Offset

Register	Offset
INIT4_SHADOW	20E0h

9.3.3.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	emr2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	emr3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.103.3 Fields

Field	Function
31-16 emr2	DDR2: Value to write to EMR2 register. DDR3/DDR4: Value to write to MR2 register LPDDR2/LPDDR3/ LPDDR4: Value to write to MR3 register mDDR: Unused
15-0 emr3	DDR2: Value to write to EMR3 register. DDR3/DDR4: Value to write to MR3 register mDDR/LPDDR2/ LPDDR3: Unused LPDDR4: Value to write to MR13 register

9.3.3.1.104 [SHADOW] SDRAM Initialization Register 6 (INIT6_SHADOW)

9.3.3.1.104.1 Offset

Register	Offset
INIT6_SHADOW	20E8h

9.3.3.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	mr4															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	mr5															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.3.3.1.104.3 Fields

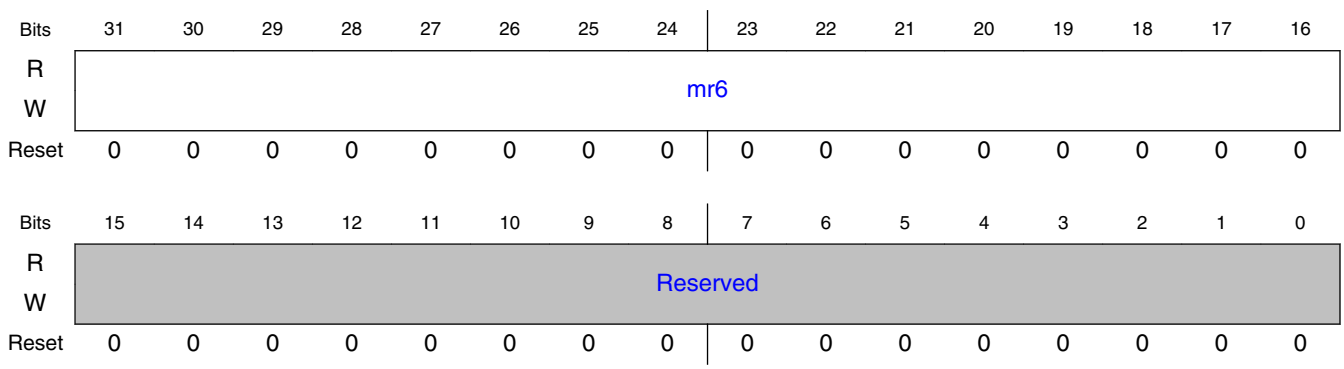
Field	Function
31-16 mr4	DDR4- Value to be loaded into SDRAM MR4 registers. Used in DDR4 designs only.
15-0 mr5	DDR4- Value to be loaded into SDRAM MR5 registers. Used in DDR4 designs only.

9.3.3.1.105 [SHADOW] SDRAM Initialization Register 7 (INIT7_SHADOW)

9.3.3.1.105.1 Offset

Register	Offset
INIT7_SHADOW	20ECh

9.3.3.1.105.2 Diagram



9.3.3.1.105.3 Fields

Field	Function
31-16 mr6	DDR4- Value to be loaded into SDRAM MR6 registers. Used in DDR4 designs only.
15-0 —	Reserved

9.3.3.1.106 [SHADOW] SDRAM Timing Register 0 (DRAMTMG0_SHADOW)

9.3.3.1.106.1 Offset

Register	Offset
DRAMTMG0_SHADOW	2100h

9.3.3.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved	wr2pre							Reserved		t_faw					
W																
Reset	0	0	0	0	1	1	1	1	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved	t_ras_max							Reserved		t_ras_min					
W																
Reset	0	0	0	1	1	0	1	1	0	0	0	0	1	1	1	1

9.3.3.1.106.3 Fields

Field	Function
31 —	Reserved
30-24 wr2pre	Minimum time between write and precharge to same bank. Unit: Clocks Specifications: WL + BL/2 + tWR = approximately 8 cycles + 15 ns = 14 clocks @400MHz and less for lower frequencies where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM. BST (burst terminate) is not supported at present. - tWR = Write recovery time. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR2/LPDDR3/LPDDR4 for this parameter. When the controller is operating in 1:2 frequency ratio mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
23-22 —	Reserved
21-16 t_faw	tFAW Valid only when 8 or more banks(or banks x bank groups) are present. In 8-bank design, at most 4 banks must be activated in a rolling window of tFAW cycles. When the controller is operating in 1:2 frequency ratio mode, program this to (tFAW/2) and round up to next integer value. In a 4-bank design, set this register to 0x1 independent of the 1:1/1:2 frequency mode. Unit: Clocks

Table continues on the next page...

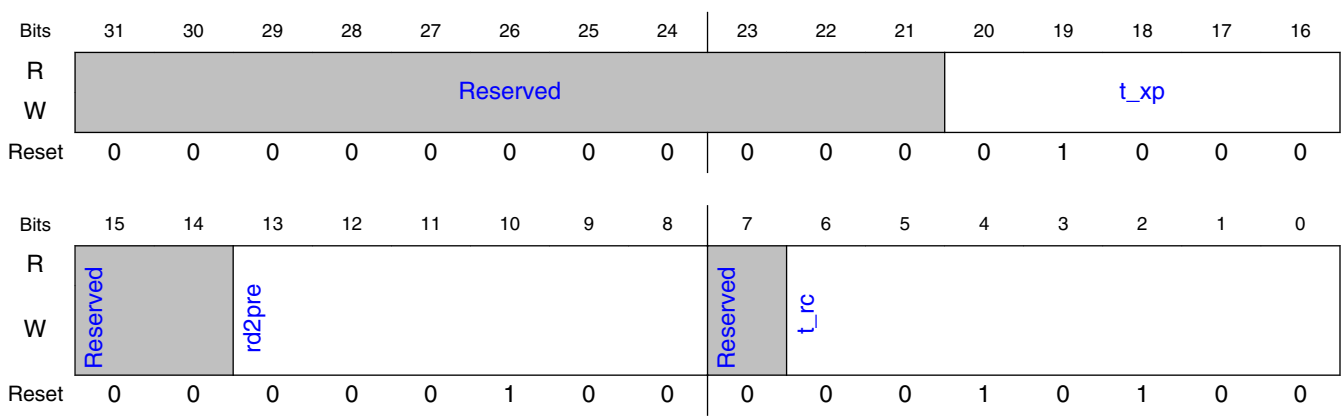
Field	Function
15 —	Reserved
14-8 t_ras_max	tRAS(max): Maximum time between activate and precharge to same bank. This is the maximum time that a page can be kept open Minimum value of this register is 1. Zero is invalid. When the controller is operating in 1:2 frequency ratio mode, program this to (tRAS(max)-1)/2. No rounding up. Unit: Multiples of 1024 clocks.
7-6 —	Reserved
5-0 t_ras_min	tRAS(min): Minimum time between activate and precharge to the same bank. When the controller is operating in 1:2 frequency mode, 1T mode, program this to tRAS(min)/2. No rounding up. When the controller is operating in 1:2 frequency ratio mode, 2T mode or LPDDR4 mode, program this to (tRAS(min)/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.107 [SHADOW] SDRAM Timing Register 1 (DRAMTMG1_SHADOW)

9.3.3.1.107.1 Offset

Register	Offset
DRAMTMG1_SHADOW	2104h

9.3.3.1.107.2 Diagram



9.3.3.1.107.3 Fields

Field	Function
31-21 —	Reserved

Table continues on the next page...

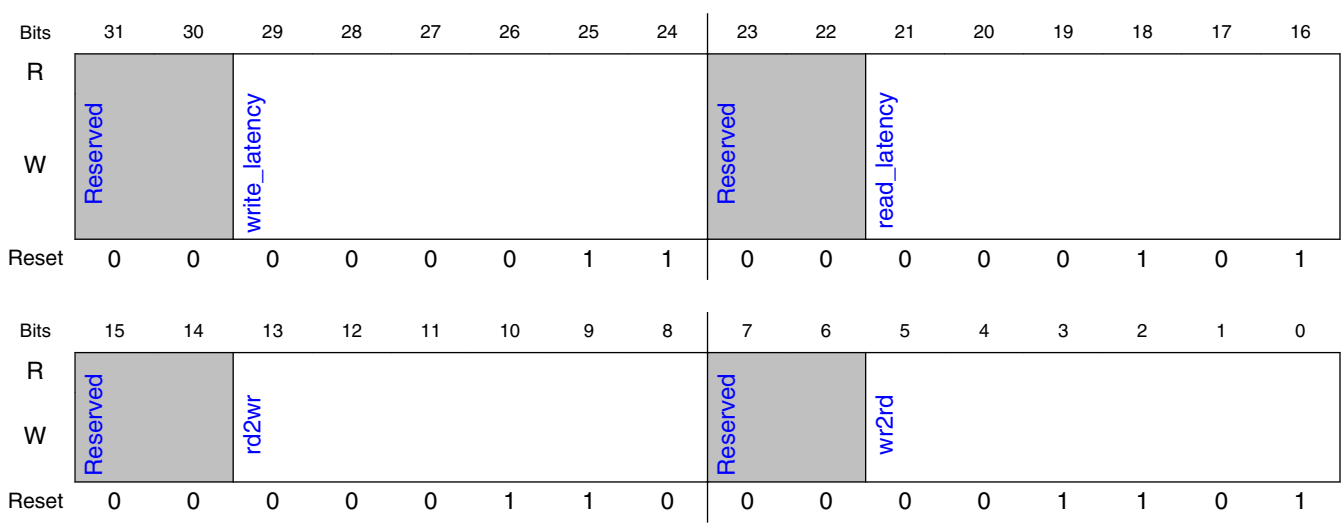
Field	Function
20-16 t_xp	tXP: Minimum time after power-down exit to any operation. For DDR3, this should be programmed to tXPDLL if slow powerdown exit is selected in MR0[12]. If C/A parity for DDR4 is used, set to (tXP+PL) instead. When the controller is operating in 1:2 frequency ratio mode, program this to (tXP/2) and round it up to the next integer value. Units: Clocks
15-14 —	Reserved
13-8 rd2pre	tRTP: Minimum time from read to precharge of same bank. - DDR2: tAL + BL/2 + max(tRTP, 2) - 2 - DDR3: tAL + max (tRTP, 4) - DDR4: Max of following two equations: tAL + max (tRTP, 4) or, RL + BL/2 - tRP (*). - mDDR: BL/2 - LPDDR2: Depends on if it's LPDDR2-S2 or LPDDR2-S4: LPDDR2-S2: BL/2 + tRTP - 1. LPDDR2-S4: BL/2 + max(tRTP,2) - 2. - LPDDR3: BL/2 + max(tRTP,4) - 4 - LPDDR4: BL/2 + max(tRTP,8) - 8 (*) When both DDR4 SDRAM and ST-MRAM are used simultaneously, use SDRAM's tRP value for calculation. When the controller is operating in 1:2 mode, 1T mode, divide the above value by 2. No rounding up. When the controller is operating in 1:2 mode, 2T mode or LPDDR4 mode, divide the above value by 2 and round it up to the next integer value. Unit: Clocks.
7 —	Reserved
6-0 t_rc	tRC: Minimum time between activates to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tRC/2) and round up to next integer value. Unit: Clocks.

9.3.3.1.108 [SHADOW] SDRAM Timing Register 2 (DRAMTMG2_SHADOW)

9.3.3.1.108.1 Offset

Register	Offset
DRAMTMG2_SHADOW	2108h

9.3.3.1.108.2 Diagram



9.3.3.1.108.3 Fields

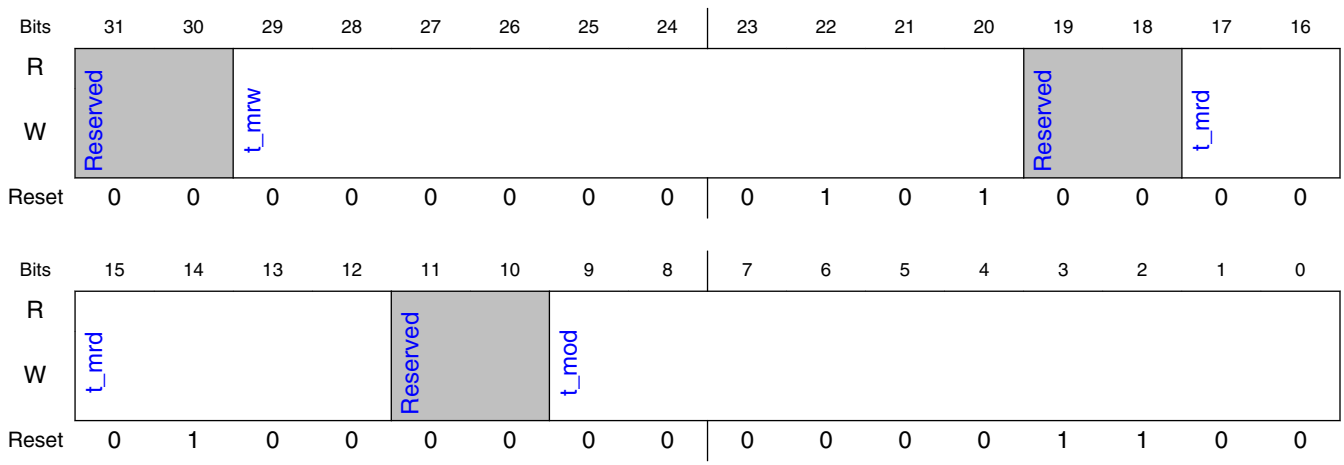
Field	Function
31-30 —	Reserved
29-24 write_latency	Set to WL Time from write command to write data on SDRAM interface. This must be set to WL. For mDDR, it should normally be set to 1. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of WL to compensate for the extra cycle of latency through the RDIMM/LRDIMM. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. This register field is not required for DDR2 and DDR3 (except if MEMC_TRAINING is set), as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols Unit: clocks
23-22 —	Reserved
21-16 read_latency	Set to RL Time from read command to read data on SDRAM interface. This must be set to RL. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to adjust the value of RL to compensate for the extra cycle of latency through the RDIMM/LRDIMM. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. This register field is not required for DDR2 and DDR3 (except if MEMC_TRAINING is set), as the DFI read and write latencies defined in DFITMG0 and DFITMG1 are sufficient for those protocols Unit: clocks
15-14 —	Reserved
13-8 rd2wr	DDR2/3/mDDR: RL + BL/2 + 2 - WL DDR4: RL + BL/2 + 1 + WR_PREAMBLE - WL LPDDR2/LPDDR3: RL + BL/2 + RU(tDQSCKmax/tCK) + 1 - WL LPDDR4(DQ ODT is Disabled): RL + BL/2 + RU(tDQSCKmax/tCK) + WR_PREAMBLE + RD_POSTAMBLE - WL LPDDR4(DQ ODT is Enabled) : RL + BL/2 + RU(tDQSCKmax/tCK) + RD_POSTAMBLE - ODTLon - RU(tODTon(min)/tCK) Minimum time from read command to write command. Include time for bus turnaround and all per-bank, per-rank, and global constraints. Please see the relevant PHY databook for details of what should be included here. Unit: Clocks. Where: - WL = write latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - RL = read latency = CAS latency - WR_PREAMBLE = write preamble. This is unique to DDR4 and LPDDR4. - RD_POSTAMBLE = read postamble. This is unique to LPDDR4. For LPDDR2/LPDDR3/LPDDR4, if derating is enabled (DERATEEN.derate_enable=1), derated tDQSCKmax should be used. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer. Note that, depending on the PHY, if using LRDIMM, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency through the LRDIMM.
7-6 —	Reserved
5-0 wr2rd	DDR4: CWL + PL + BL/2 + tWTR_L Others: CWL + BL/2 + tWTR In DDR4, minimum time from write command to read command for same bank group. In others, minimum time from write command to read command. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_L = internal write to read command delay for same bank group. This comes directly from the SDRAM specification. - tWTR = internal write to read command delay. This comes directly from the SDRAM specification. Add one extra cycle for LPDDR2/LPDDR3/LPDDR4 operation. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.109 [SHADOW] SDRAM Timing Register 3 (DRAMTMG3_SHADOW)

9.3.3.1.109.1 Offset

Register	Offset
DRAMTMG3_SHADOW	210Ch

9.3.3.1.109.2 Diagram



9.3.3.1.109.3 Fields

Field	Function
31-30 —	Reserved
29-20 t_mrw	Time to wait after a mode register write or read (MRW or MRR). Present only in designs configured to support LPDDR2, LPDDR3 or LPDDR4. LPDDR2 typically requires value of 5. LPDDR3 typically requires value of 10. LPDDR4: Set this to the larger of tMRW and tMRWCKEL. For LPDDR2, this register is used for the time from a MRW/MRR to all other commands. When the controller is operating in 1:2 frequency ratio mode, program this to the above values divided by 2 and round it up to the next integer value. For LDPDR3, this register is used for the time from a MRW/MRR to a MRW/MRR.
19-18 —	Reserved
17-12 t_mrd	tMRD: Cycles to wait after a mode register write or read. Depending on the connected SDRAM, tMRD represents: DDR2/mDDR: Time from MRS to any command DDR3/4: Time from MRS to MRS command LPDDR2: not used LPDDR3/4: Time from MRS to non-MRS command. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD/2) and round it up to the next integer value. If C/A parity for DDR4 is used, set to tMRD_PAR(tMOD+PL) instead.
11-10 —	Reserved

Table continues on the next page...

Field	Function
9-0 t_mod	tMOD: Parameter used only in DDR3 and DDR4. Cycles between load mode command and following non-load mode command. If C/A parity for DDR4 is used, set to tMOD_PAR(tMOD+PL) instead. Set to tMOD if controller is operating in 1:1 frequency ratio mode, or tMOD/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode. Note that if using RDIMM/LRDIMM, depending on the PHY, it may be necessary to adjust the value of this parameter to compensate for the extra cycle of latency applied to mode register writes by the RDIMM/LRDIMM chip. Also note that if using LRDIMM, the minimum value of this register is tMRD_L2 if controller is operating in 1:1 frequency ratio mode, or tMRD_L2/2 (rounded up to next integer) if controller is operating in 1:2 frequency ratio mode.

9.3.3.1.110 [SHADOW] SDRAM Timing Register 4 (DRAMTMG4_SHADOW)

9.3.3.1.110.1 Offset

Register	Offset
DRAMTMG4_SHADOW	2110h

9.3.3.1.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			t_rcd					Reserved				t_ccd			
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_rrd				Reserved				t_rp			
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	1

9.3.3.1.110.3 Fields

Field	Function
31-29 —	Reserved
28-24 t_rcd	tRCD - tAL: Minimum time from activate to read or write command to same bank. When the controller is operating in 1:2 frequency ratio mode, program this to ((tRCD - tAL)/2) and round it up to the next integer value. Minimum value allowed for this register is 1, which implies minimum (tRCD - tAL) value to be 2 when the controller is operating in 1:2 frequency ratio mode. Unit: Clocks.
23-20 —	Reserved

Table continues on the next page...

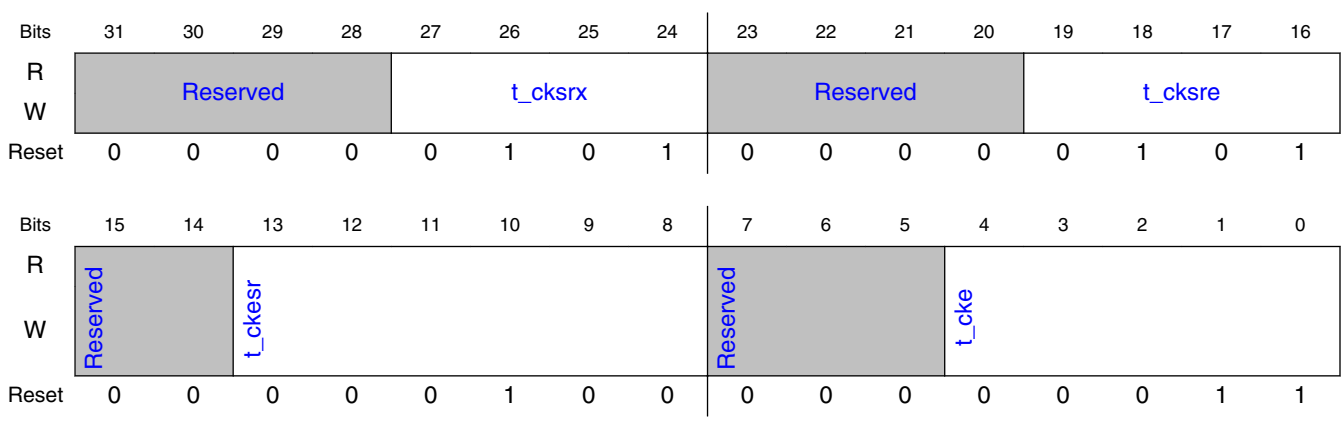
Field	Function
19-16 t_ccd	DDR4: tCCD_L: This is the minimum time between two reads or two writes for same bank group. Others: tCCD: This is the minimum time between two reads or two writes. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_L/2 or tCCD/2) and round it up to the next integer value. Unit: clocks.
15-12 —	Reserved
11-8 t_rrd	DDR4: tRRD_L: Minimum time between activates from bank "a" to bank "b" for same bank group. Others: tRRD: Minimum time between activates from bank "a" to bank "b" When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_L/2 or tRRD/2) and round it up to the next integer value. Unit: Clocks.
7-5 —	Reserved
4-0 t_rp	tRP: Minimum time from precharge to activate of same bank. When the controller is operating in 1:1 frequency ratio mode, t_rp should be set to RoundUp(tRP/tCK). When the controller is operating in 1:2 frequency ratio mode, t_rp should be set to RoundDown(RoundUp(tRP/tCK)/2) + 1. When the controller is operating in 1:2 frequency ratio mode in LPDDR4, t_rp should be set to RoundUp(RoundUp(tRP/tCK)/2). Unit: Clocks.

9.3.3.1.111 [SHADOW] SDRAM Timing Register 5 (DRAMTMG5_SHADOW)

9.3.3.1.111.1 Offset

Register	Offset
DRAMTMG5_SHADOW	2114h

9.3.3.1.111.2 Diagram



9.3.3.1.111.3 Fields

Field	Function
31-28 —	Reserved
27-24 t_cksrx	This is the time before Self Refresh Exit that CK is maintained as a valid clock before issuing SRX. Specifies the clock stable time before SRX. Recommended settings: - mDDR: 1 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEH - DDR2: 1 - DDR3: tCKSRX - DDR4: tCKSRX When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
23-20 —	Reserved
19-16 t_cksre	This is the time after Self Refresh Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after SRE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEL - DDR2: 1 - DDR3: max (10 ns, 5 tCK) - DDR4: max (10 ns, 5 tCK) (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
15-14 —	Reserved
13-8 t_ckesr	Minimum CKE low width for Self refresh or Self refresh power down entry to exit timing in memory clock cycles. Recommended settings: - mDDR: tRFC - LPDDR2: tCKESR - LPDDR3: tCKESR - LPDDR4: max(tCKELPD, tSR) - DDR2: tCKE - DDR3: tCKE + 1 - DDR4: tCKE + 1 (+ PL(parity latency)(*)) (*)Only if CRCPARCTL1.caparity_disable_before_sr=0, this register should be increased by PL. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer.
7-5 —	Reserved
4-0 t_cke	Minimum number of cycles of CKE HIGH/LOW during power-down and self refresh. - LPDDR2/LPDDR3 mode: Set this to the larger of tCKE or tCKESR - LPDDR4 mode: Set this to the larger of tCKE, tCKELPD or tSR. - Non-LPDDR2/non-LPDDR3/non-LPDDR4 designs: Set this to tCKE value. When the controller is operating in 1:2 frequency ratio mode, program this to (value described above)/2 and round it up to the next integer value. Unit: Clocks.

9.3.3.1.112 [SHADOW] SDRAM Timing Register 6 (DRAMTMG6_SHADOW)

9.3.3.1.112.1 Offset

Register	Offset
DRAMTMG6_SHADOW	2118h

9.3.3.1.112.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				t_ckdpde				Reserved				t_ckdpdx			
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												t_ckcsx			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.3.3.1.112.3 Fields

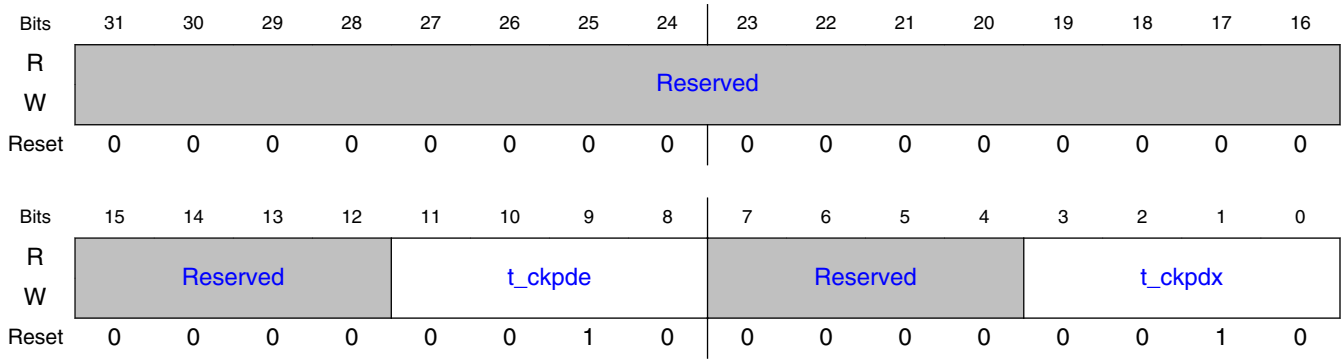
Field	Function
31-28 —	Reserved
27-24 t_ckdpde	This is the time after Deep Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after DPDE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3 devices.
23-20 —	Reserved
19-16 t_ckdpdx	This is the time before Deep Power Down Exit that CK is maintained as a valid clock before issuing DPDX. Specifies the clock stable time before DPDX. Recommended settings: - mDDR: 1 - LPDDR2: 2 - LPDDR3: 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2 devices.
15-4 —	Reserved
3-0 t_ckcsx	This is the time before Clock Stop Exit that CK is maintained as a valid clock before issuing Clock Stop Exit. Specifies the clock stable time before next command after Clock Stop Exit. Recommended settings: - mDDR: 1 - LPDDR2: tXP + 2 - LPDDR3: tXP + 2 - LPDDR4: tXP + 2 When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.113 [SHADOW] SDRAM Timing Register 7 (DRAMTMG7_SHADOW)

9.3.3.1.113.1 Offset

Register	Offset
DRAMTMG7_SHADOW	211Ch

9.3.3.1.113.2 Diagram



9.3.3.1.113.3 Fields

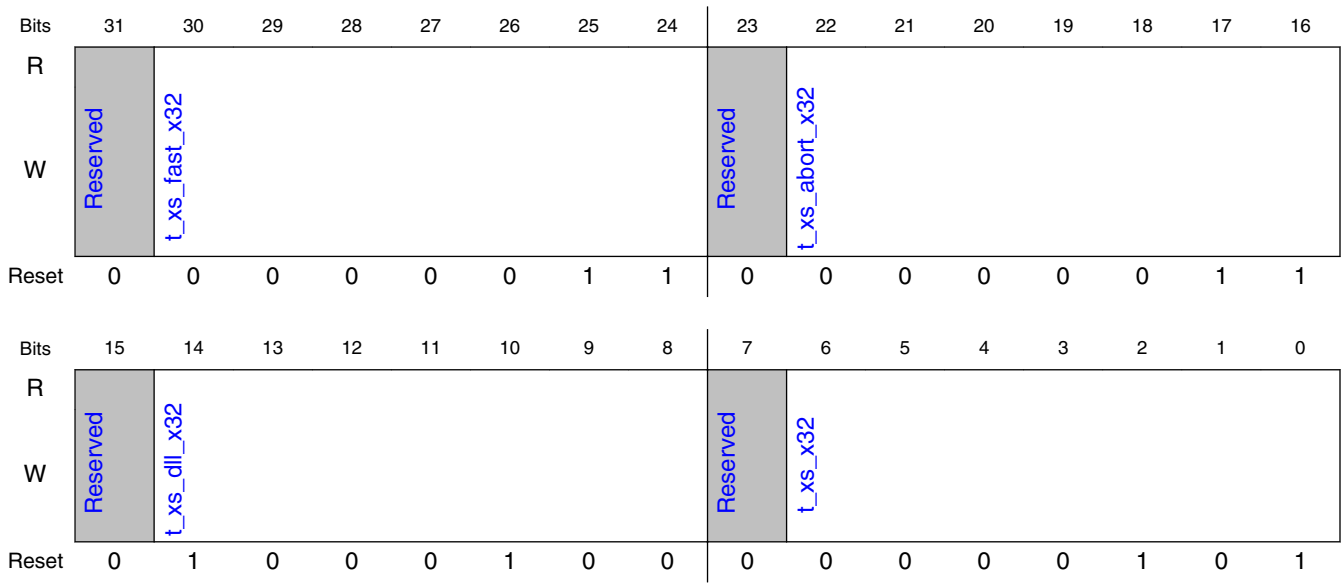
Field	Function
31-12 —	Reserved
11-8 t_ckpde	This is the time after Power Down Entry that CK is maintained as a valid clock. Specifies the clock disable delay after PDE. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: tCKCKEL When using DDR2/3/4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksre. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/ LPDDR3/LPDDR4 devices.
7-4 —	Reserved
3-0 t_ckpdx	This is the time before Power Down Exit that CK is maintained as a valid clock before issuing PDX. Specifies the clock stable time before PDX. Recommended settings: - mDDR: 0 - LPDDR2: 2 - LPDDR3: 2 - LPDDR4: 2 When using DDR2/3/4 SDRAM, this register should be set to the same value as DRAMTMG5.t_cksrx. When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. This is only present for designs supporting mDDR or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.114 [SHADOW] SDRAM Timing Register 8 (DRAMTMG8_SHADOW)

9.3.3.1.114.1 Offset

Register	Offset
DRAMTMG8_SHADOW	2120h

9.3.3.1.114.2 Diagram



9.3.3.1.114.3 Fields

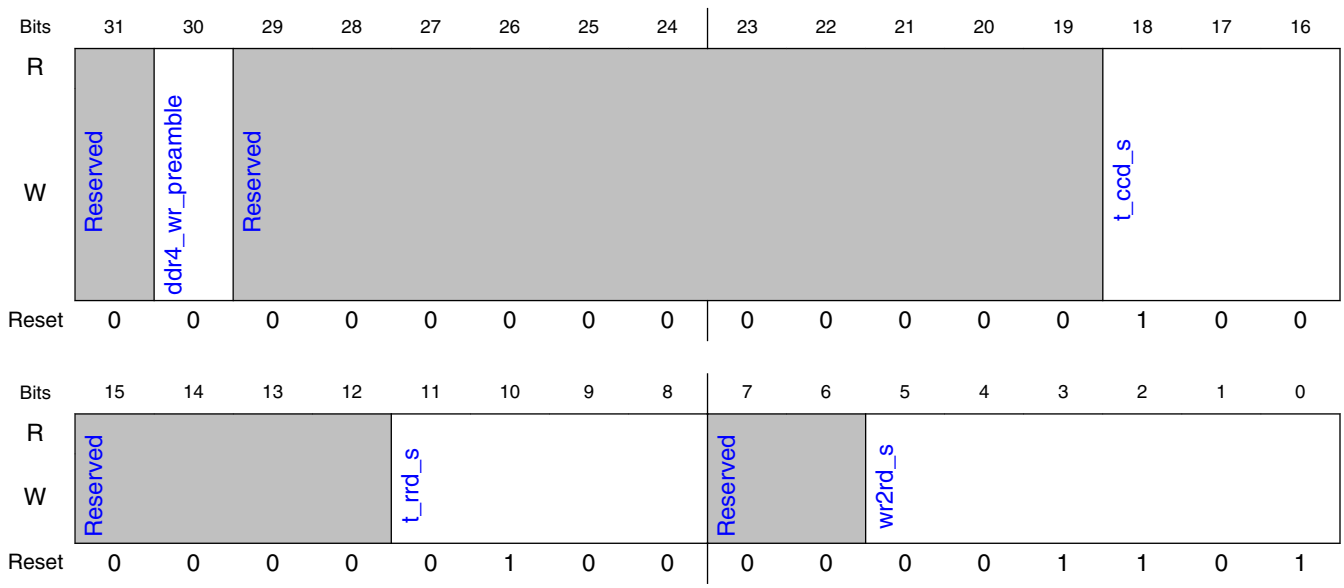
Field	Function
31 —	Reserved
30-24 t_xs_fast_x32	tXS_FAST: Exit Self Refresh to ZQCL, ZQCS and MRS (only CL, WR, RTP and Geardown mode). When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: This is applicable to only ZQCL/ZQCS commands. Note: Ensure this is less than or equal to t_xs_x32.
23 —	Reserved
22-16 t_xs_abort_x32	tXS_ABORT: Exit Self Refresh to commands not requiring a locked DLL in Self Refresh Abort. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Ensure this is less than or equal to t_xs_x32.
15 —	Reserved
14-8 t_xs_dll_x32	tXSDLL: Exit Self Refresh to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR2, DDR3 and DDR4 SDRAMs.
7 —	Reserved
6-0 t_xs_x32	tXS: Exit Self Refresh to commands not requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Unit: Multiples of 32 clocks. Note: Used only for DDR2, DDR3 and DDR4 SDRAMs.

9.3.3.1.115 [SHADOW] SDRAM Timing Register 9 (DRAMTMG9_SHADOW)

9.3.3.1.115.1 Offset

Register	Offset
DRAMTMG9_SHADOW	2124h

9.3.3.1.115.2 Diagram



9.3.3.1.115.3 Fields

Field	Function
31 —	Reserved
30 ddr4_wr_preamble	DDR4 Write preamble mode - 0: 1tCK preamble - 1: 2tCK preamble Present only with MEMC_FREQ_RATIO=2
29-19 —	Reserved
18-16 t_ccd_s	tCCD_S: This is the minimum time between two reads or two writes for different bank group. For bank switching (from bank "a" to bank "b"), the minimum time is this value + 1. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: clocks.
15-12 —	Reserved

Table continues on the next page...

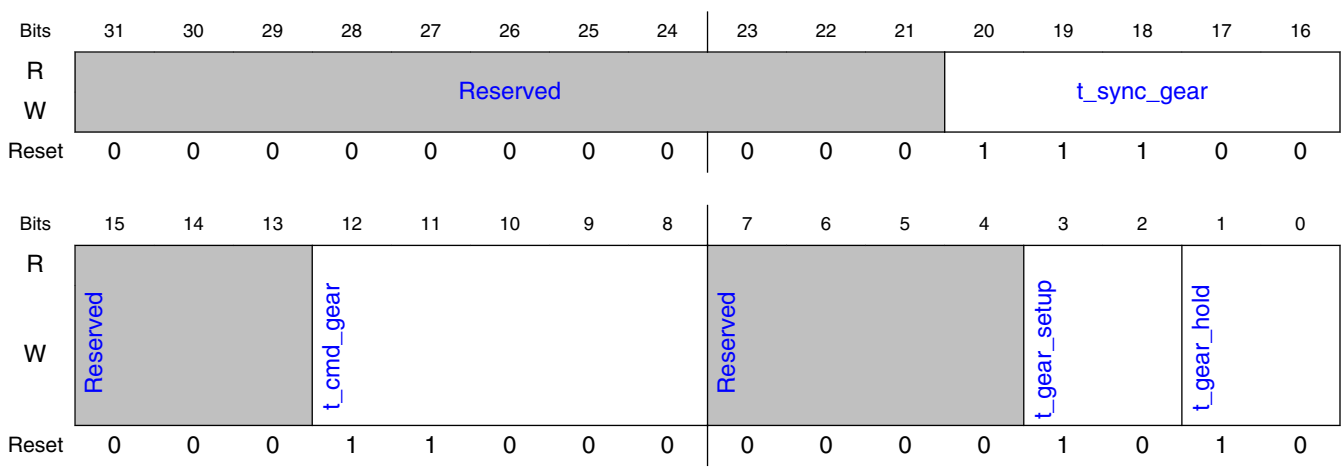
Field	Function
11-8 t_rrd_s	tRRD_S: Minimum time between activates from bank "a" to bank "b" for different bank group. When the controller is operating in 1:2 frequency ratio mode, program this to (tRRD_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-6 —	Reserved
5-0 wr2rd_s	CWL + PL + BL/2 + tWTR_S Minimum time from write command to read command for different bank group. Includes time for bus turnaround, recovery times, and all per-bank, per-rank, and global constraints. Present only in designs configured to support DDR4. Unit: Clocks. Where: - CWL = CAS write latency - PL = Parity latency - BL = burst length. This must match the value programmed in the BL bit of the mode register to the SDRAM - tWTR_S = internal write to read command delay for different bank group. This comes directly from the SDRAM specification. When the controller is operating in 1:2 mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.116 [SHADOW] SDRAM Timing Register 10 (DRAMTMG10_SHADOW)

9.3.3.1.116.1 Offset

Register	Offset
DRAMTMG10_SHADOW	2128h

9.3.3.1.116.2 Diagram



9.3.3.1.116.3 Fields

Field	Function
31-21	Reserved

Table continues on the next page...

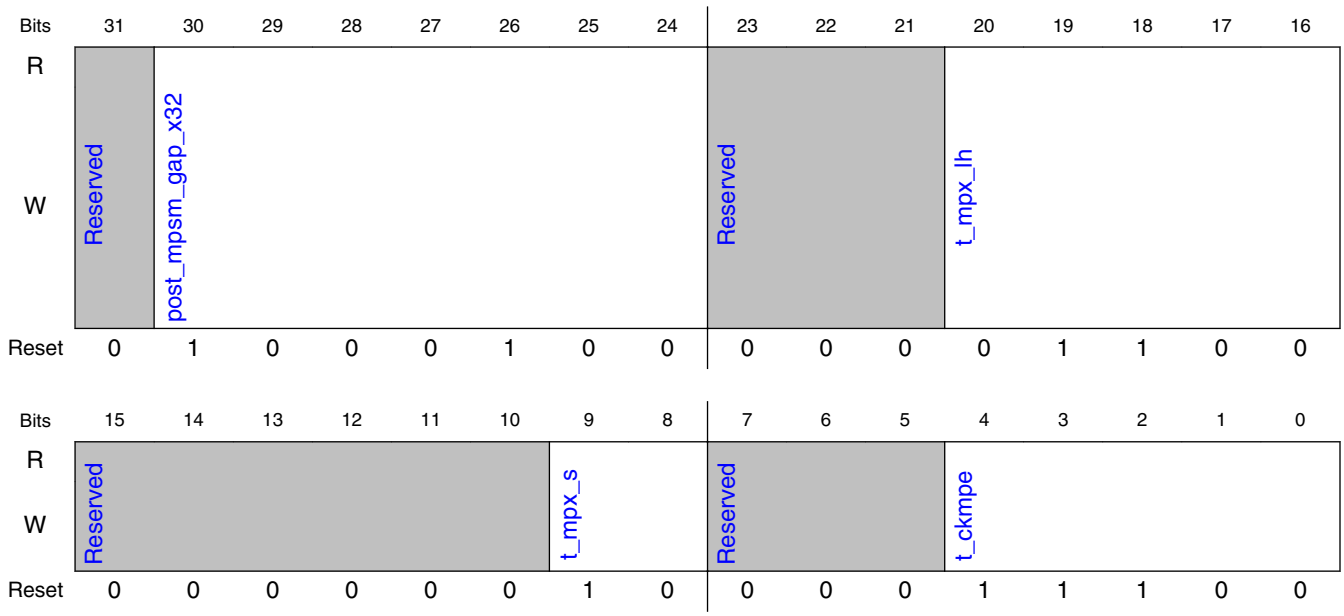
Field	Function
—	
20-16 t_sync_gear	Indicates the time between MRS command and the sync pulse time. This must be even number of clocks. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min)+4nCK tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24+4 = 28 When the controller is operating in 1:2 mode, program this to (tSYNC_GEAR/2) and round it up to the next integer value. Unit: Clocks
15-13 —	Reserved
12-8 t_cmd_gear	Sync pulse to first valid command. For DDR4-2666 and DDR4-3200, this parameter is defined as tMOD(min) tMOD(min) is greater of 24nCK or 15ns 15ns / .625ns = 24 Max value for this register is 24 When the controller is operating in 1:2 mode, program this to (tCMD_GEAR/2) and round it up to the next integer value. Unit: Clocks
7-4 —	Reserved
3-2 t_gear_setup	Geardown setup time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_setup/2) and round it up to the next integer value. Unit: Clocks
1-0 t_gear_hold	Geardown hold time. Minimum value of this register is 1. Zero is invalid. For DDR4-2666 and DDR4-3200, this parameter is defined as 2 clks When the controller is operating in 1:2 frequency ratio mode, program this to (tGEAR_hold/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.117 [SHADOW] SDRAM Timing Register 11 (DRAMTMG11_SHADOW)

9.3.3.1.117.1 Offset

Register	Offset
DRAMTMG11_SHADOW	212Ch

9.3.3.1.117.2 Diagram



9.3.3.1.117.3 Fields

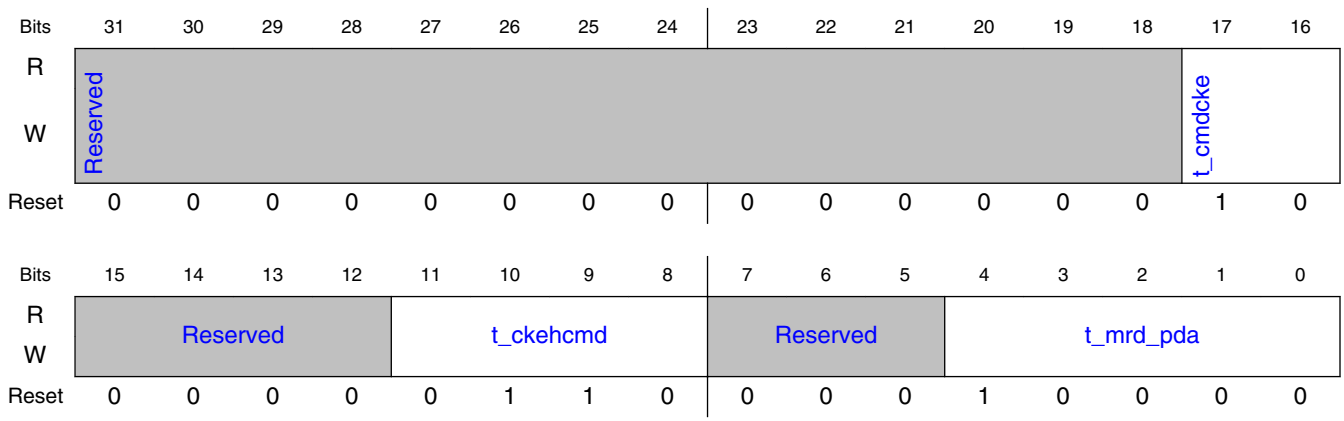
Field	Function
31 —	Reserved
30-24 post_mpsm_gap_x32	tXMPDLL: This is the minimum Exit MPSM to commands requiring a locked DLL. When the controller is operating in 1:2 frequency ratio mode, program this to (tXMPDLL/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Multiples of 32 clocks.
23-21 —	Reserved
20-16 t_mpx_lh	tMPX_LH: This is the minimum CS_n Low hold time to CKE rising edge. When the controller is operating in 1:2 frequency ratio mode, program this to RoundUp(tMPX_LH/2)+1. Present only in designs configured to support DDR4. Unit: clocks.
15-10 —	Reserved
9-8 t_mpx_s	tMPX_S: Minimum time CS setup time to CKE. When the controller is operating in 1:2 frequency ratio mode, program this to (tMPX_S/2) and round it up to the next integer value. Present only in designs configured to support DDR4. Unit: Clocks.
7-5 —	Reserved
4-0 t_ckmpe	tCKMPE: Minimum valid clock requirement after MPSM entry. Present only in designs configured to support DDR4. Unit: Clocks. When the controller is operating in 1:2 frequency ratio mode, divide the value calculated using the above equation by 2, and round it up to next integer.

9.3.3.1.118 [SHADOW] SDRAM Timing Register 12 (DRAMTMG12_SHADOW)

9.3.3.1.118.1 Offset

Register	Offset
DRAMTMG12_SHADOW	2130h

9.3.3.1.118.2 Diagram



9.3.3.1.118.3 Fields

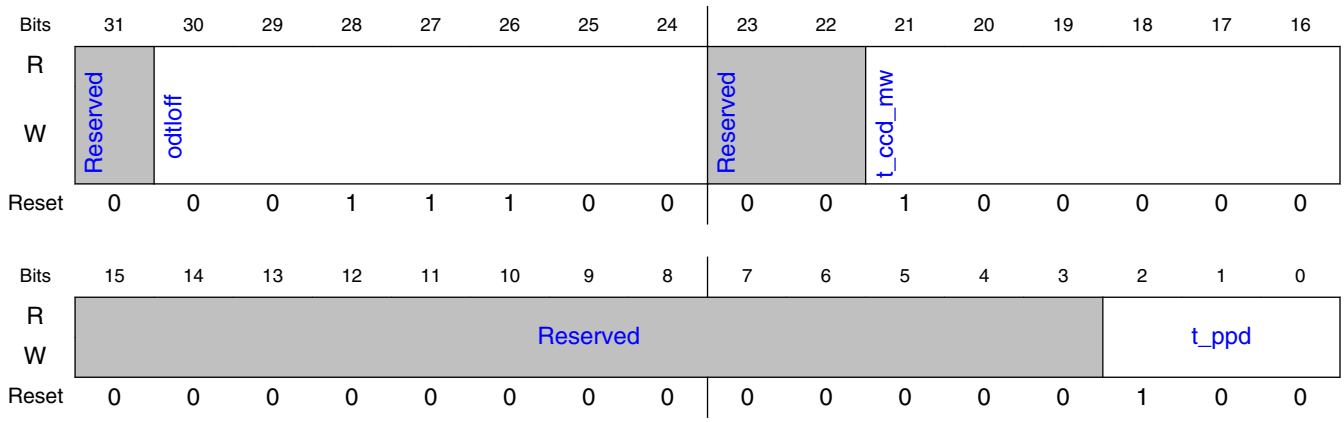
Field	Function
31-18 —	Reserved
17-16 t_cmdcke	tCMDCKE: Delay from valid command to CKE input LOW. Set this to the larger of tESCKE or tCMDCKE When the controller is operating in 1:2 frequency ratio mode, program this to (max(tESCKE, tCMDCKE)/2) and round it up to the next integer value.
15-12 —	Reserved
11-8 t_ckehcmd	tCKEHCMD: Valid command requirement after CKE input HIGH. When the controller is operating in 1:2 frequency ratio mode, program this to (tCKEHCMD/2) and round it up to the next integer value.
7-5 —	Reserved
4-0 t_mrd_pda	tMRD_PDA: This is the Mode Register Set command cycle time in PDA mode. When the controller is operating in 1:2 frequency ratio mode, program this to (tMRD_PDA/2) and round it up to the next integer value.

9.3.3.1.119 [SHADOW] SDRAM Timing Register 13 (DRAMTMG13_SHADOW)

9.3.3.1.119.1 Offset

Register	Offset
DRAMTMG13_SHADOW	2134h

9.3.3.1.119.2 Diagram



9.3.3.1.119.3 Fields

Field	Function
31 —	Reserved
30-24 odtloff	LPDDR4: tODTLoff: This is the latency from CAS-2 command to tODToff reference. When the controller is operating in 1:2 frequency ratio mode, program this to (tODTLoff/2) and round it up to the next integer value. Unit: Clocks.
23-22 —	Reserved
21-16 t_ccd_mw	LPDDR4: tCCDMW: This is the minimum time from write or masked write to masked write command for same bank. When the controller is operating in 1:2 frequency ratio mode, program this to (tCCDMW/2) and round it up to the next integer value. Unit: Clocks.
15-3 —	Reserved
2-0 t_ppd	LPDDR4: tPPD: This is the minimum time from precharge to precharge command. When the controller is operating in 1:2 frequency ratio mode, program this to (tPPD/2) and round it up to the next integer value. Unit: Clocks.

9.3.3.1.120 [SHADOW] SDRAM Timing Register 14 (DRAMTMG14_SHADOW)

9.3.3.1.120.1 Offset

Register	Offset
DRAMTMG14_SHADOW	2138h

9.3.3.1.120.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				t_xsr											
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

9.3.3.1.120.3 Fields

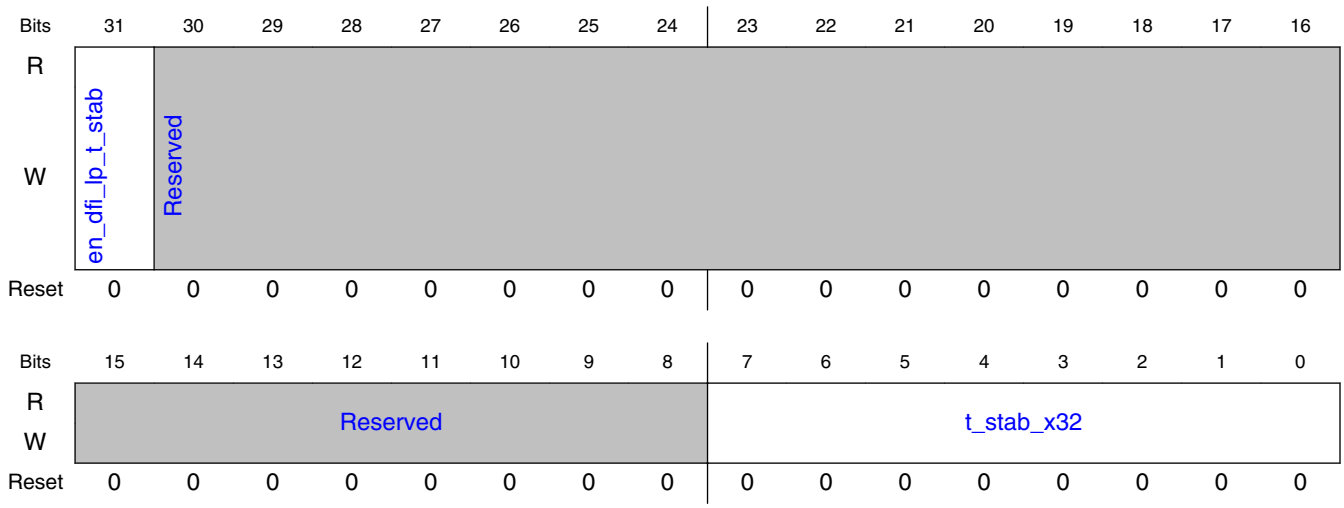
Field	Function
31-12 —	Reserved
11-0 t_xsr	tXSR: Exit Self Refresh to any command. When the controller is operating in 1:2 frequency ratio mode, program this to the above value divided by 2 and round up to next integer value. Note: Used only for mDDR/LPDDR2/LPDDR3/LPDDR4 mode.

9.3.3.1.121 [SHADOW] SDRAM Timing Register 15 (DRAMTMG15_SHADOW)

9.3.3.1.121.1 Offset

Register	Offset
DRAMTMG15_SHADOW	213Ch

9.3.3.1.121.2 Diagram



9.3.3.1.121.3 Fields

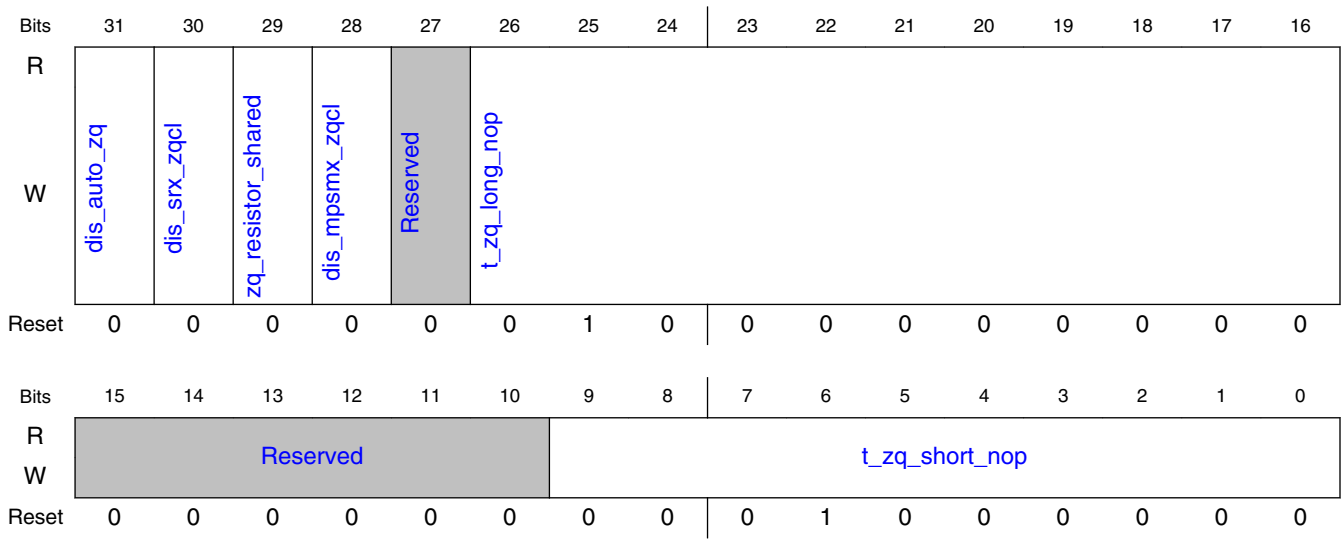
Field	Function
31 en_dfi_lp_t_stab	- 1 - Enable using tSTAB when exiting DFI LP. Needs to be set when the PHY is stopping the clock during DFI LP to save maximum power. - 0 - Disable using tSTAB when exiting DFI LP
30-8 —	Reserved
7-0 t_stab_x32	tSTAB: Stabilization time. It is required in the following two cases for DDR3/DDR4 RDIMM : - when exiting power saving mode, if the clock was stopped, after re-enabling it the clock must be stable for a time specified by tSTAB - in the case of input clock frequency change (DDR4) - after issuing control words that refers to clock timing (Specification: 6us for DDR3, 5us for DDR4) When the controller is operating in 1:2 frequency ratio mode, program this to recommended value divided by two and round it up to next integer. Unit: Multiples of 32 clock cycles.

9.3.3.1.122 [SHADOW] ZQ Control Register 0 (ZQCTL0_SHADOW)

9.3.3.1.122.1 Offset

Register	Offset
ZQCTL0_SHADOW	2180h

9.3.3.1.122.2 Diagram



9.3.3.1.122.3 Fields

Field	Function
31 dis_auto_zq	- 1 - Disable DDRC generation of ZQCS/MPC(ZQ calibration) command. Register DBGCMD.zq_calib_short can be used instead to issue ZQ calibration request from APB module. - 0 - Internally generate ZQCS/MPC(ZQ calibration) commands based on ZQCTL1.t_zq_short_interval_x1024. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.
30 dis_srx_zqcl	- 1 - Disable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3 or DDR4 or LPDDR2 or LPDDR3 or LPDDR4 mode. - 0 - Enable issuing of ZQCL/MPC(ZQ calibration) command at Self-Refresh/SR-Powerdown exit. Only applicable when run in DDR3 or DDR4 or LPDDR2 or LPDDR3 or LPDDR4 mode. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.
29 zq_resistor_shared	- 1 - Denotes that ZQ resistor is shared between ranks. Means ZQinit/ZQCL/ZQCS/MPC(ZQ calibration) commands are sent to one rank at a time with tZQinit/tZQCL/tZQCS/tZQCAL/tZQLAT timing met between commands so that commands to different ranks do not overlap. - 0 - ZQ resistor is not shared. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.
28 dis_mpsmx_zqcl	- 1 - Disable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. - 0 - Enable issuing of ZQCL command at Maximum Power Saving Mode exit. Only applicable when run in DDR4 mode. This is only present for designs supporting DDR4 devices. Note: Do not issue ZQCL command at Maximum Power Save Mode exit if the DDRC_SHARED_AC configuration parameter is set. Program it to 1'b1. The software can send ZQCS after exiting MPSM mode.
27 —	Reserved
26-16 t_zq_long_nop	tZQoper for DDR3/DDR4, tZQCL for LPDDR2/LPDDR3, tZQCAL for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCL (ZQ calibration long)/MPC(ZQ Start) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode: DDR3/DDR4: program this to tZQoper/2 and round it up to the next integer value. LPDDR2/LPDDR3: program this to tZQCL/2 and round it up to the next integer value. LPDDR4: program this to tZQCAL/2 and round it up to the next integer value. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.
15-10	Reserved

Table continues on the next page...

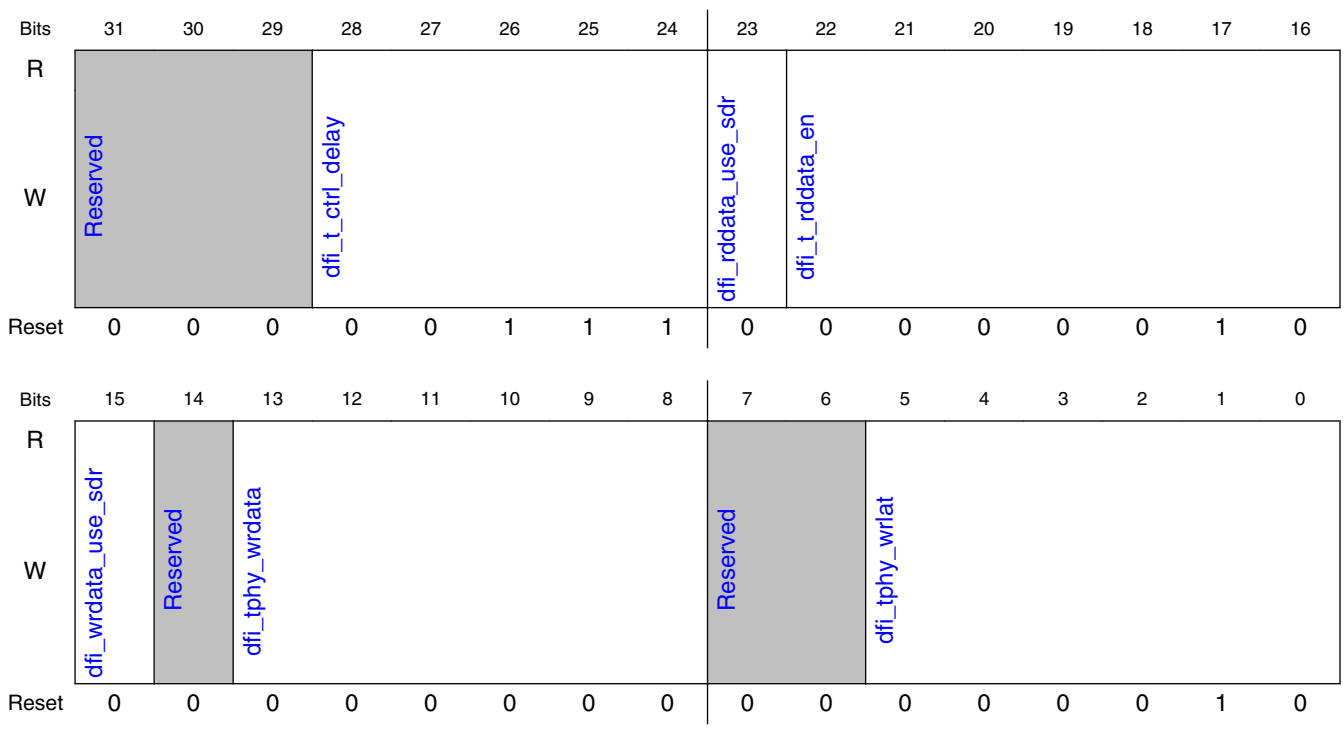
Field	Function
—	
9-0 t_zq_short_nop	tZQCS for DDR3/DD4/LPDDR2/LPDDR3, tZQLAT for LPDDR4: Number of DFI clock cycles of NOP required after a ZQCS (ZQ calibration short)/MPC(ZQ Latch) command is issued to SDRAM. When the controller is operating in 1:2 frequency ratio mode, program this to tZQCS/2 and round it up to the next integer value. This is only present for designs supporting DDR3/DDR4 or LPDDR2/LPDDR3/LPDDR4 devices.

9.3.3.1.123 [SHADOW] DFI Timing Register 0 (DFITMG0_SHADOW)

9.3.3.1.123.1 Offset

Register	Offset
DFITMG0_SHADOW	2190h

9.3.3.1.123.2 Diagram



9.3.3.1.123.3 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

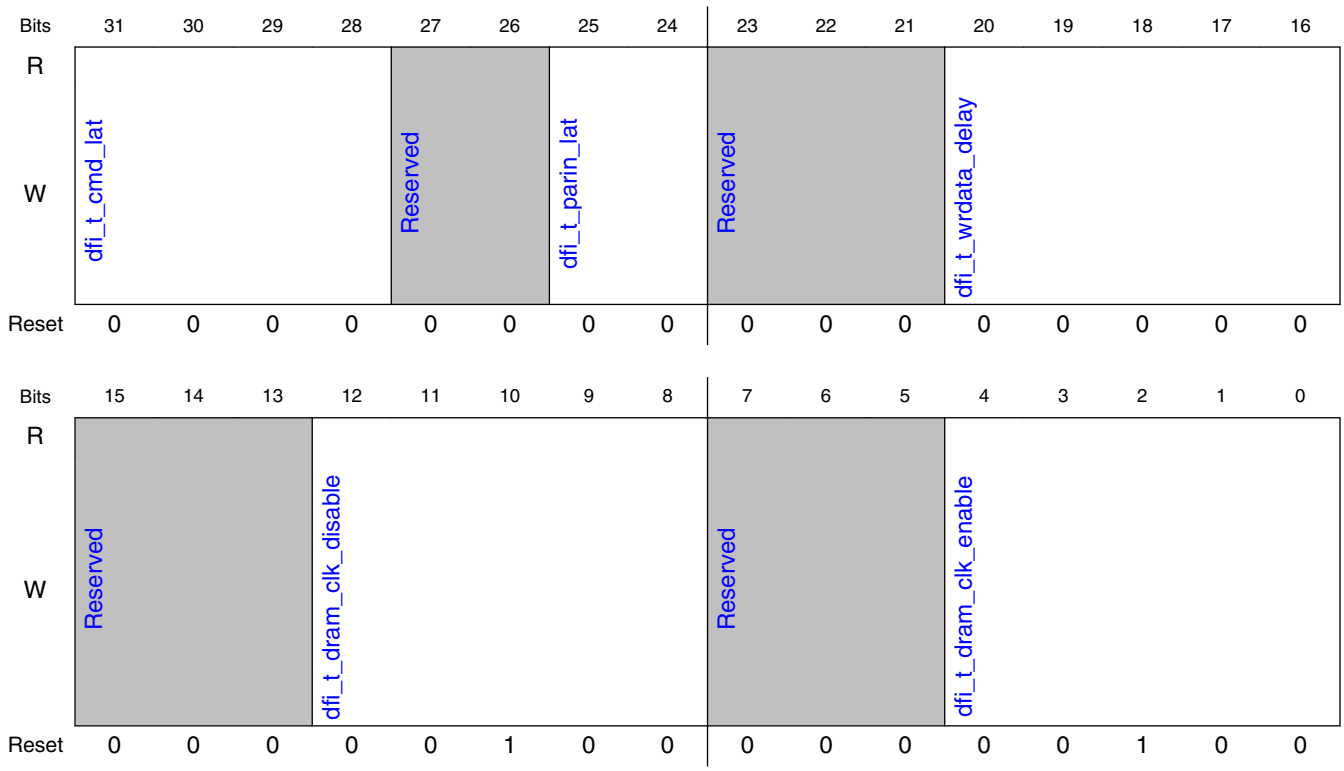
Field	Function
—	
28-24 dfi_t_ctrl_delay	Specifies the number of DFI clock cycles after an assertion or de-assertion of the DFI control signals that the control signals at the PHY-DRAM interface reflect the assertion or de-assertion. If the DFI clock and the memory clock are not phase-aligned, this timing parameter should be rounded up to the next integer value. Note that if using RDIMM/LRDIMM, it is necessary to increment this parameter by RDIMM's/ LRDIMM's extra cycle of latency in terms of DFI clock.
23 dfi_rddata_use_sdr	Defines whether dfi_rddata_en/dfi_rddata/dfi_rddata_valid is generated using HDR (DFI clock) or SDR (DFI PHY clock) values. Selects whether value in DFITMG0.dfi_t_rddata_en is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles: - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
22-16 dfi_t_rddata_en	Time from the assertion of a read command on the DFI interface to the assertion of the dfi_rddata_en signal. Refer to PHY specification for correct value. This corresponds to the DFI parameter trddata_en. Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of trddata_en. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_rddata_use_sdr.
15 dfi_wrdata_use_sdr	Defines whether dfi_wrdata_en/dfi_wrdata/dfi_wrdata_mask is generated using HDR (DFI clock) or SDR (DFI PHY clock) values Selects whether value in DFITMG0.dfi_tphy_wrlat is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles Selects whether value in DFITMG0.dfi_tphy_wrdata is in terms of HDR (DFI clock) or SDR (DFI PHY clock) cycles - 0 in terms of HDR (DFI clock) cycles - 1 in terms of SDR (DFI PHY clock) cycles Refer to PHY specification for correct value.
14 —	Reserved
13-8 dfi_tphy_wrdata	Specifies the number of clock cycles between when dfi_wrdata_en is asserted to when the associated write data is driven on the dfi_wrdata signal. This corresponds to the DFI timing parameter tphy_wrdata. Refer to PHY specification for correct value. Note, max supported value is 8. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.
7-6 —	Reserved
5-0 dfi_tphy_wrlat	Write latency Number of clocks from the write command to write data enable (dfi_wrdata_en). This corresponds to the DFI timing parameter tphy_wrlat. Refer to PHY specification for correct value.Note that, depending on the PHY, if using RDIMM/LRDIMM, it may be necessary to use the adjusted value of CL in the calculation of tphy_wrlat. This is to compensate for the extra cycle(s) of latency through the RDIMM/LRDIMM. Unit: DFI clock cycles or DFI PHY clock cycles, depending on DFITMG0.dfi_wrdata_use_sdr.

9.3.3.1.124 [SHADOW] DFI Timing Register 1 (DFITMG1_SHADOW)

9.3.3.1.124.1 Offset

Register	Offset
DFITMG1_SHADOW	2194h

9.3.3.1.124.2 Diagram



9.3.3.1.124.3 Fields

Field	Function
31-28 dfi_t_cmd_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated command is driven. This field is used for CAL mode, should be set to '0' or the value which matches the CAL mode register setting in the DRAM. If the PHY can add the latency for CAL mode, this should be set to '0'. Valid Range: 0, 3, 4, 5, 6, and 8
27-26 —	Reserved
25-24 dfi_t_parin_lat	Specifies the number of DFI PHY clock cycles between when the dfi_cs signal is asserted and when the associated dfi_parity_in signal is driven.
23-21 —	Reserved
20-16 dfi_t_wrdata_delay	Specifies the number of DFI clock cycles between when the dfi_wrdata_en signal is asserted and when the corresponding write data transfer is completed on the DRAM bus. This corresponds to the DFI timing parameter twrdata_delay. Refer to PHY specification for correct value. For DFI 3.0 PHY, set to twrdata_delay, a new timing parameter introduced in DFI 3.0. For DFI 2.1 PHY, set to tphy_wrdata + (delay of DFI write data to the DRAM). Value to be programmed is in terms of DFI clocks, not PHY clocks. In FREQ_RATIO=2, divide PHY's value by 2 and round up to next integer. If using DFITMG0.dfi_wrdata_use_sdr=1, add 1 to the value. Unit: Clocks
15-13 —	Reserved

Table continues on the next page...

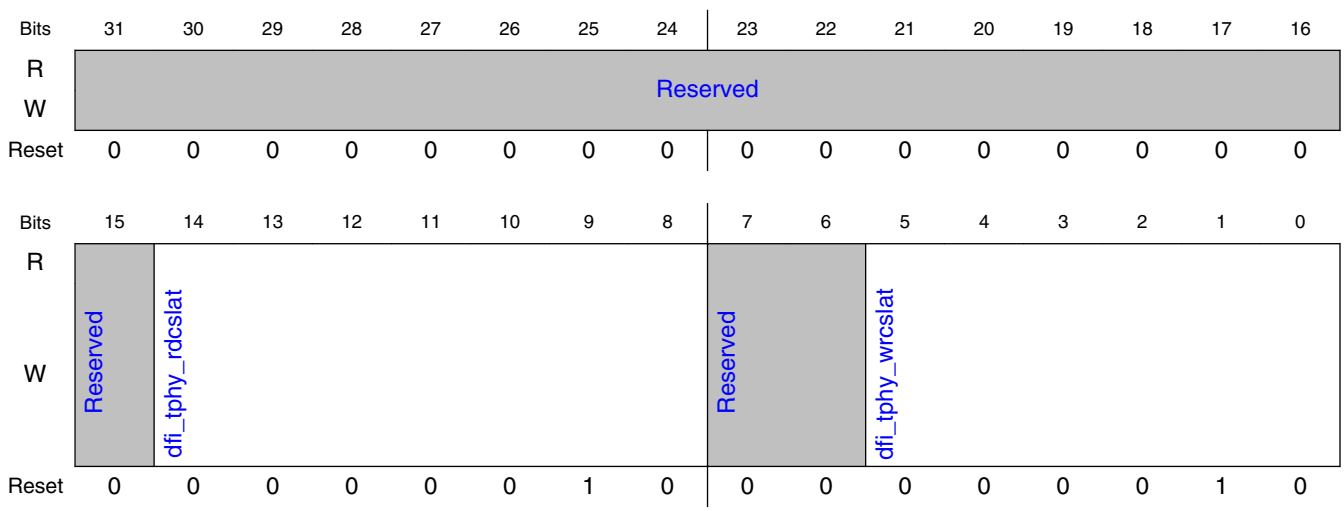
Field	Function
12-8 dfi_t_dram_clk_disable	Specifies the number of DFI clock cycles from the assertion of the dfi_dram_clk_disable signal on the DFI until the clock to the DRAM memory devices, at the PHY-DRAM boundary, maintains a low value. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.
7-5 —	Reserved
4-0 dfi_t_dram_clk_enable	Specifies the number of DFI clock cycles from the de-assertion of the dfi_dram_clk_disable signal on the DFI until the first valid rising edge of the clock to the DRAM memory devices, at the PHY-DRAM boundary. If the DFI clock and the memory clock are not phase aligned, this timing parameter should be rounded up to the next integer value.

9.3.3.1.125 [SHADOW] DFI Timing Register 2 (DFITMG2_SHADOW)

9.3.3.1.125.1 Offset

Register	Offset
DFITMG2_SHADOW	21B4h

9.3.3.1.125.2 Diagram



9.3.3.1.125.3 Fields

Field	Function
31-15 —	Reserved

Table continues on the next page...

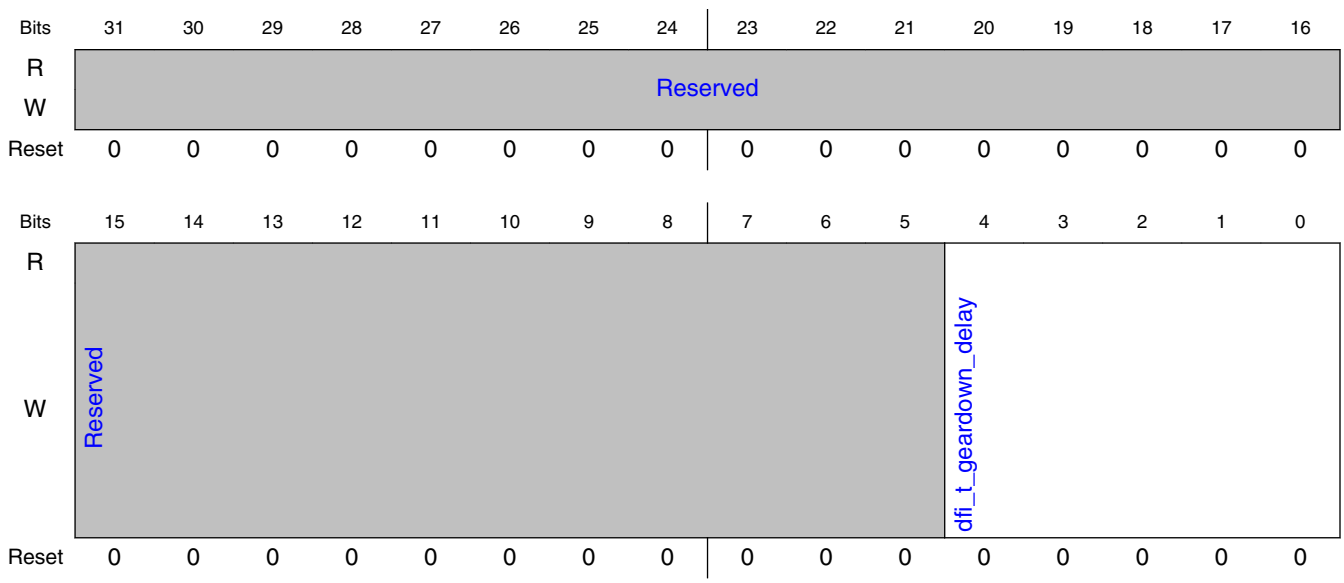
Field	Function
14-8 dfi_tphy_rdcslat	Number of DFI PHY clock cycles between when a read command is sent on the DFI control interface and when the associated dfi_rddata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_rdcslat. Refer to PHY specification for correct value.
7-6 —	Reserved
5-0 dfi_tphy_wrcslat	Number of DFI PHY clock cycles between when a write command is sent on the DFI control interface and when the associated dfi_wrdata_cs signal is asserted. This corresponds to the DFI timing parameter tphy_wrcslat. Refer to PHY specification for correct value.

9.3.3.1.126 [SHADOW] DFI Timing Register 3 (DFITMG3_SHADOW)

9.3.3.1.126.1 Offset

Register	Offset
DFITMG3_SHADOW	21B8h

9.3.3.1.126.2 Diagram



9.3.3.1.126.3 Fields

Field	Function
31-5	Reserved

Table continues on the next page...

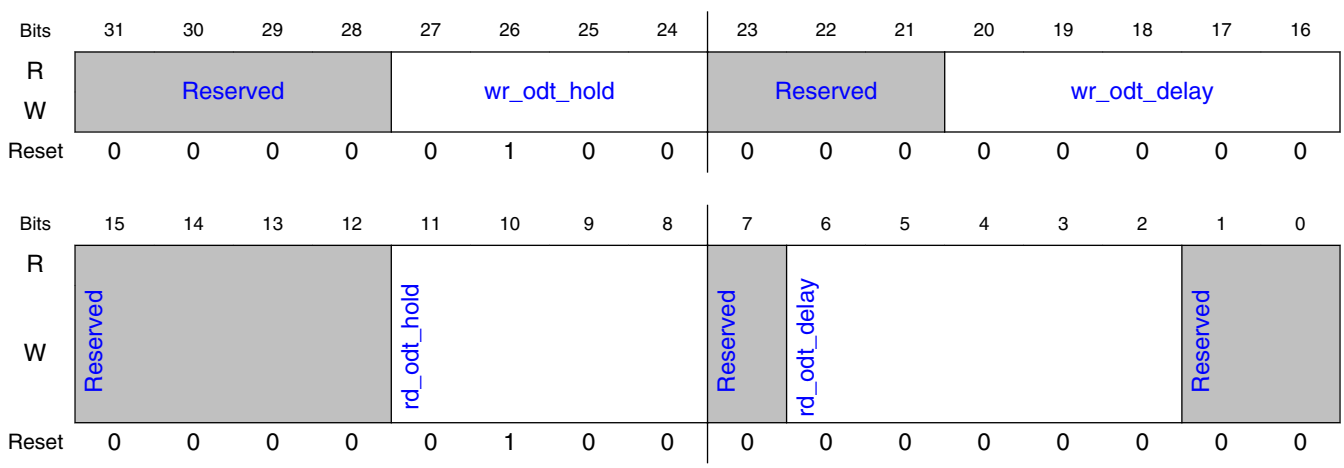
Field	Function
—	
4-0 dfi_t_geardown_delay	The delay from dfi_geardown_en assertion to the time of the PHY being ready to receive commands. Refer to PHY specification for correct value. When the controller is operating in 1:2 frequency ratio mode, program this to (tgeardown_delay/2) and round it up to the next integer value. Unit: Clocks

9.3.3.1.127 [SHADOW] ODT Configuration Register (ODTCFG_SHADOW)

9.3.3.1.127.1 Offset

Register	Offset
ODTCFG_SHADOW	2240h

9.3.3.1.127.2 Diagram



9.3.3.1.127.3 Fields

Field	Function
31-28 —	Reserved
27-24 wr_odt_hold	DFI PHY clock cycles to hold ODT for a write command. The minimum supported value is 2. Recommended values: DDR2: - BL8: 0x5 (DDR2-400/533/667), 0x6 (DDR2-800), 0x7 (DDR2-1066) - BL4: 0x3 (DDR2-400/533/667), 0x4 (DDR2-800), 0x5 (DDR2-1066) DDR3: - BL8: 0x6 DDR4: - BL8: 5 + WR_PREAMBLE + CRC_MODE WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) CRC_MODE = 0 (not CRC mode), 1 (CRC mode) LPDDR3: - BL8: 7 + RU(tODTon(max)/tCK)
23-21	Reserved

Table continues on the next page...

Field	Function
—	
20-16 wr_odt_delay	The delay, in DFI PHY clock cycles, from issuing a write command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR2: - CWL + AL - 3 (DDR2-400/533/667), CWL + AL - 4 (DDR2-800), CWL + AL - 5 (DDR2-1066) If (CWL + AL - 3 < 0), DDRC does not support ODT for write operation. DDR3: - 0x0 DDR4: - DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) LPDDR3: - WL - 1 - RU(tODTon(max)/tCK))
15-12 —	Reserved
11-8 rd_odt_hold	DFI PHY clock cycles to hold ODT for a read command. The minimum supported value is 2. Recommended values: DDR2: - BL8: 0x6 (not DDR2-1066), 0x7 (DDR2-1066) - BL4: 0x4 (not DDR2-1066), 0x5 (DDR2-1066) DDR3: - BL8 - 0x6 DDR4: - BL8: 5 + RD_PREAMBLE RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) LPDDR3: - BL8: 5 + RU(tDQSCK(max)/tCK) - RD(tDQSCK(min)/tCK) + RU(tODTon(max)/tCK)
7 —	Reserved
6-2 rd_odt_delay	The delay, in DFI PHY clock cycles, from issuing a read command to setting ODT values associated with that command. ODT setting must remain constant for the entire time that DQS is driven by the DDRC. Recommended values: DDR2: - CL + AL - 4 (not DDR2-1066), CL + AL - 5 (DDR2-1066) If (CL + AL - 4 < 0), DDRC does not support ODT for read operation. DDR3: - CL - CWL DDR4: - CL - CWL - RD_PREAMBLE + WR_PREAMBLE + DFITMG1.dfi_t_cmd_lat (to adjust for CAL mode) WR_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) RD_PREAMBLE = 1 (1tCK write preamble), 2 (2tCK write preamble) If (CL - CWL - RD_PREAMBLE + WR_PREAMBLE) < 0, DDRC does not support ODT for read operation. LPDDR3: - RL + RD(tDQSCK(min)/tCK) - 1 - RU(tODTon(max)/tCK)
1-0 —	Reserved

9.4 DDR PHY (DDR_PHY)

9.4.1 Overview

The Double Data Rate Physical Layer (DDR PHY), provides an interface between universal controllers and external DDR3/DDR3L/DDR4/LPDDR3/LPDDR4 SDRAM devices.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

The block digram of the DDR PHY is shown below.

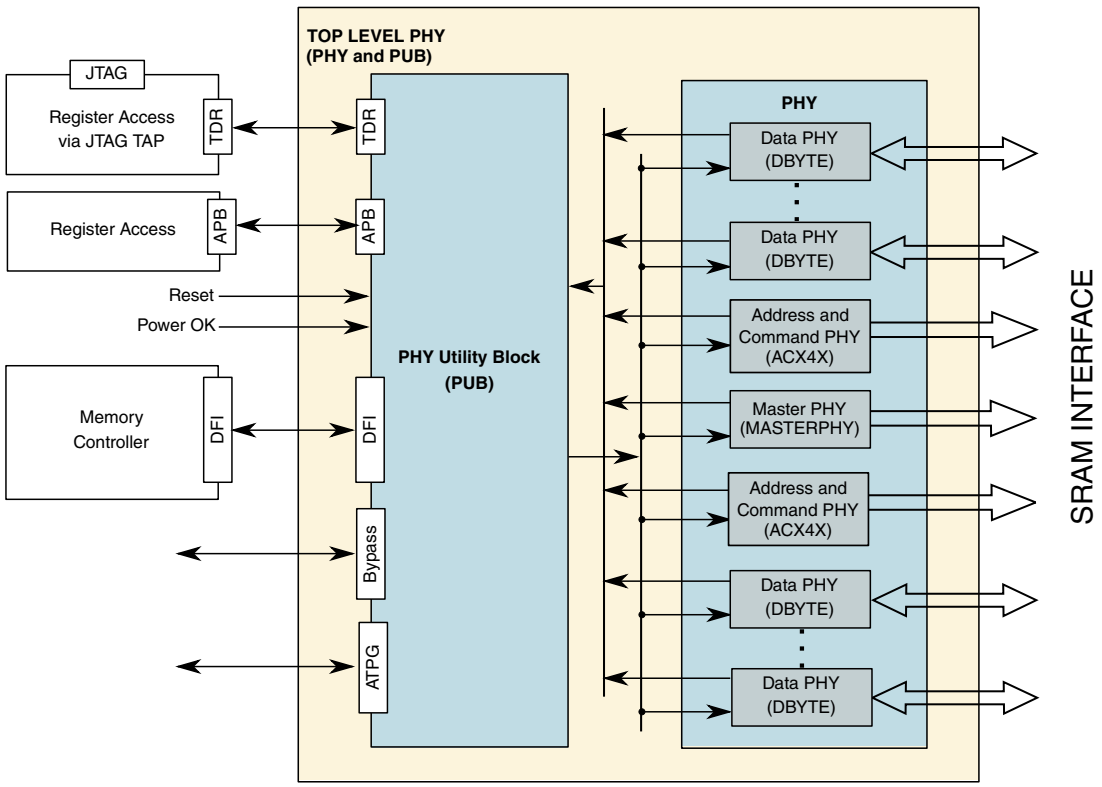


Figure 9-6. Block Diagram

9.4.1.1 Features

The following features are supported in the DDR PHY:

- LPDDR4, LPDDR3, DDR4, DDR3, and DDR3L operation
- Multiple memory rank support:
  - Four ranks supported for DDR4, DDR3, and DDR3L
  - Two ranks supported for LPDDR4 and LPDDR3
- DFI interface (DFI v4.0 Addendum 2, March 2015)
- At-speed loopback testing on both the address and data channels
- Mux-scan ATPG
- High-speed PLL used to generate the high speed clock for data transmit and high speed digital pipelines
- PHY Utility Block (PUB), which provides configuration and control
- Power on reset (POR) circuit, DRAM reset, DRAM alert and PHY debug
- A global voltage reference generation and observation circuit
- Impedance calibration circuits

9.4.2 Functional Description

The DDR PHY is composed of the following:

- DBYTE - The eight-lane data IO unit
- ACX4 - The four-lane address/command hard IP unit
- MASTERPHY - Contains the PLL, the voltage reference, thermal calibration, and reset
- PUB - PHY Utility Block

9.4.2.1 DDR Address/Command PHY (ACX4)

Info to be included at a future date.

The DDR SDRAM address/command PHY (DDRPHYACX4) provides an address and command interface to the external SDRAM memories. A memory interface typically contains a single address/command PHY and one or more data PHYs (one for each byte lane). Figure 2-1 shows the organization of the AC component.

The AC consists of multiple design slices. The address/command (AC) slice is used for all address and command and clk/cs/odt signals. Each slice contains all logic required for both the output (address/command) and input (loopback) paths of the respective signals.

All clocking inside the AC is derived from an input clock, which is generated by the PLL in the MASTERPHY. The PLL generates the higher frequency clocks that are is used inside the macro and minimizes input clock jitter.

9.4.2.1.1 ACX4 Transmit Lanes

The ACX4 component contains 4 transmit only lanes which are grouped as follows:

- Clock group: {CK_t, CK_c}
- CA group: {ras_n, cas_n, we_n, addr…}
- CKE,CS,ODT group{CKE, CS_n, ODT…}

In DDR4 systems, 12 ACs are required to control one 72b/64b channel with four independent ranks.

In LPDDR3 and LPDDR4 systems, the 12 ACs are broken up into two independently controllable groups, enabling up to two independent 32b channels (each with a maximum of two ranks.

9.4.2.1.2 ACX4 Command Path

Each SDRAM address and command signal is driven from the memory controller to the SDRAM through an address/command slice. Controlled address/command pipeline logic inside the AC slice converts the DFI-rate signals from the controller to single data rate (SDR) Address/Command signals going to the SDRAM. The last launching register inside the address/command slice is clocked by a delay-shifted DDR clock. For SDR operation, as required by DDR3 DRAMs, all outputs remain constants for two cycles at a time.

The path for the SDRAM clock (CK) is through a standard AC slice which is configured to act as a dedicated clock (CK) slice. This shifted address/command output makes the CK centered into the data eye of the address/command signals.

All AC outputs going to the SDRAM have a LCDL, which can be used to align or shift the outputs of the data PHY. This delay element can also be used to provide delay adjustment during functional or loopback modes.

9.4.2.1.3 ACX4 Loopback/DFT Paths

For every lane there will be a loopback-only receive path. The loopback path is stimulated and checked by utilizing a FirmWare image.

9.4.2.2 DDR Data PHY (DBYTE)

The DDR SDRAM Data PHY (DBYTE) provides data interface to external SDRAM memory. Features of the DBYTE include:

- High speed digital logic pipeline for transmit datapaths to the SDRAM
- High speed digital logic pipeline for receive datapaths from the SDRAM
- Drivers and receivers for the DQ and DQS signals as well as all the necessary high voltage analog macros to interface off chip
- I/O component that includes PVT-compensated on-die termination (ODT) and output impedance
- Loop-back test mode and PRBS checkers to help verify the integrity of the internal datapath of all DBYTE I/O lanes

9.4.2.2.1 DBYTE Write Path

Each SDRAM data is driven from the PUB to the SDRAM through a data (DQ) slice. Controlled data pipeline logic inside the DBYTE slice converts the DFI-rate signals from the PUB to double-data rate (DDR) signals going to the SDRAM. The last launching register inside the data slice is clocked by the DDR clock.

The path for the SDRAM write strobe is through a dedicated data strobe (DQS) slice. The output register of the DQS slice is clocked by a delayed version of PclkIn. PclkIn drives the output register in the DQ slice is shifted using the DLL in training mode. This centers the write data strobe signal in the write data eye.

All DBYTE outputs going to the SDRAM, including write data, write data strobe read DQS gate, driver power down, receiver power down, on-die termination enable, and output enable have a DLL that is used to deskew the outputs in order to maximize the write data eye.

Apart from the DQS pipeline, the DQS slice also contains a similar pipeline for output enables, on-die termination enable, driver power down, and receiver power down. This matches the I/O control signals to the corresponding data and data strobe.

9.4.2.2.2 DBYTE Read and Loopback Path

Read data from the SDRAM is written into a four-deep asynchronous FIFO that is implemented inside the DQ slice.

NOTE

The FIFO is 16 deep in terms of the DRAM DQS writing into the FIFO; it is 8 deep on the core side when a half-speed DFICLK is employed and it is 8 deep on the core side when a full speed DFICLK is employed.

The FIFO is written using the DQS clock and read using the Controller clock. A per-bit delay line at the input of the DQ slice is used to optionally de-skew read data with respect to the read strobe in order to maximize the read data eye.

The read data strobe path and a read DQS gate path is implemented inside the DQS slice. A delay line at the input of the slice shifts the read DQS by nominal 90 degrees (with the actual value arrived at through training) so that the strobe is centered into the read data eye before clocking the data into the FIFO. A Bit- Wise Delay Line (BDL) in the read DQS path is used to match a corresponding bit de-skew BDL in the DQ slice.

The DQS slice also generates the write pointers for the data FIFO. These pointers, together with the strobe, are then distributed to all DQ slices. The PUB generates a trained read-valid signal to read the data from the PHY.

The DBI uses DQS Slice 1, configured to behave as a DQ slice in x8 mode only.

9.4.2.2.2.1 DBYTE Loopback/DFT Paths

High speed loopback is implemented on all DBYTE I/Os. The DBYTE read path also doubles as the loopback path. During loopback, the PUB enables the gating of the DQS signal in the same way as it would normally gate the DQS during reads. This allows the looped back data to be clocked into the read data FIFO using the looped back DQS signal.

9.4.2.2.2.2 Memory Controller based DBYTE Loopback

This high speed PHY loopback mode uses all the DBYTE control and data paths in mission mode like operation modes. In this mode, the memory controller issues simultaneous Read/Write commands to the PHY, resulting in transmitting and receiving the same data on each DQ pad.

9.4.2.3 Master PHY (MASTERPHY)

The Master block contains the PLL, the Primary Voltage Reference, thermal calibration, reset, and I/O cells for test and control.

9.4.2.3.1 PLL

The PLL in the MASTER is a simplified PLL that has DFICLK as an input and creates an internal clock by quadrupling the DFICLK frequency. The PLL in the MASTER can only multiply the input clock frequency to produce an output clock that is four times the frequency of the input clock. The internal clock is the output of the MASTER to the ACX4 and DBYTE.

9.4.2.3.1.1 Optimal PLL Settings for Mission Mode Frequency Bins

Table 9-4. Optimal PLL Settings for Mission Mode Frequency Bins (PLL Max Bandwidth = 2 MHz)

pllin min (MHz)	pllin max (MHz)	Input div. ratio	Fbk min (MHz)	Fbk max (MHz)	Fbk div. ratio	VCO min (MHz)	VCO max (MHz)	PllCtrl2 PllFreqSel[4:0]	PLLTestMode int_half[8] prop_half[5] vco_mode[2] overflow[1]	PllCtrl1 PllCpPropCtrl[8:5]	PllCtrl1 PllCpIntCtrl[4:0]	PllCtrl4 PllCpPropGsCtrl[8:5]	PllCtrl4 PllCpIntGsCtrl[4:0]
937.5	1067	8	117.2	133.4	4	3750	4268	11000 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
625	937.5	8	78.13	117.2	4	2500	3750	11001 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b
468.75	625	4	117.2	156.3	8	3750	5000	01010 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
312.5	468.75	4	78.13	117.2	8	2500	3750	01011 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b
234.4	312.5	2	117.2	156.3	16	3750	5000	00110 ’b	1011’b	0011’b	00000 ’b	1011’b	11111 ’b
166	234.4	2	83	117.2	16	2656	3750	00111 ’b	1011’b	0100’b	00000 ’b	1011’b	11111 ’b

9.4.2.3.2 Voltage Reference

This primary voltage reference generates the VREF global out used by the receivers in the ACX4 and DBYTE.

9.4.2.3.3 Calibration

The MASTER contains the analog blocks to perform output drive and input termination impedance calibration. Through the use of a digital control state machine, located in the PUB, the full mixed-signal calibration loop matches the impedance of the driver and termination with that of an external precision resistor. The resulting digital calibration code is fed to the address/command, DQ, and DQS blocks to control their impedance.

9.4.2.3.4 Reset

Reset takes the asynchronous reset bump, the PwrOk bump, and dfi_reset_n signal and combines them and synchronizes to the DFI clock. This is used by the ACX4 and DBYTE.

9.4.2.4 PHY Utility Block (PUB)

The following is an overview of the different functionality of the PHY Utility Block (PUB):

- Registers required to configure, control, train and self-test the PHY. Also contains registers required to initialize the DRAM.
- Initialization - this includes locking of the PLL, delay line calibration and calibration of I/O impedances.
- Data Training- train the PHY for optimum timing margins. This includes finding the best positioning of the DQS gating window during reads, write leveling, data bit deskew, and optimizing the read and write data eye.
- Implements a DFI-compliant interface between the external controller and the PHY.

9.4.2.4.1 DDR PHY Control and Configuration

The DDR PHY's PUB registers provide the required configuration, control, training and self-testing of the PHY. Also contains registers required to initialize the DRAM.

Refer to the DDR PHY register descriptions for more information.

9.4.2.4.2 Initialization and Training Sequence

The following outlines the general initialization and training sequence. Unless otherwise stated, these functions occur within the DDR PHY.

1. Bring up VDD, VDDQ, and VAA. This occurs outside of the DDR PHY.
2. Start the clocks and reset the PHY. This also occurs outside of the PHY.
3. Initialize PHY configuration.
4. Load the PHY training firmware into instruction memory SRAM.
5. Set the PHY input clocks to the desired frequency. This occurs outside of the PHY.

6. Write the Message Block parameters for the training firmware.There are several locations in the data memory that must be written for the training firmware to have these parameters.
7. Execute the training firmware.
8. Once the training firmware completes, the results are returned to the data memory in the message block structure. This message block can be read. The result may be that another frequency needs to be trained (return to step 5).
9. Load the PHY Initialization Engine registers with the provided initialization sequence.
10. Initialize the PHY to mission mode by performing a DFI initialization sequence per the DFI specification.
11. The PHY is now in mission mode.

9.4.2.4.3 DFI Interface

The DDR PHY and the DDR Controller both require a standard interface to transfer address, data, and control between these two blocks. These interfaces are DFI compliant. The PUB contains the DFI interface for the PHY.

For more information regarding the DFI specification, refer to the following:

- DDR PHY Interface (DFI) Specification, Revision 3.1 (Preliminary), 21, March 2014
- DDR PHY Interface (DFI) Specification Preliminary DFI 4.0 Specification - Addendum to DFI 3.1, 2 April 2014

The DFI interface is sub-divided into the following interface groups:

- Control Interface
- Write Data Interface
- Read Data Interface
- Update Interface
- Status Interface
- Training Interface (PHY-independent mode supported by the DDRC)
- Low Power Control Interface

9.4.3 Memory Map and Register Definition

This section includes the DDRC PHY module memory map and detailed descriptions of all registers.

NOTE

Synopsys Proprietary. Used with permission.

9.4.3.1 DWC_DDRPHYA_ANIB register descriptions

9.4.3.1.1 DWC_DDRPHYA_ANIB Memory map

DDR4/3 PHY address block

DWC_DDRPHYA_ANIB0 base address: 0h

DWC_DDRPHYA_ANIB1 base address: 1000h

DWC_DDRPHYA_ANIB2 base address: 2000h

DWC_DDRPHYA_ANIB3 base address: 3000h

DWC_DDRPHYA_ANIB4 base address: 4000h

DWC_DDRPHYA_ANIB5 base address: 5000h

DWC_DDRPHYA_ANIB6 base address: 6000h

DWC_DDRPHYA_ANIB7 base address: 7000h

DWC_DDRPHYA_ANIB8 base address: 8000h

DWC_DDRPHYA_ANIB9 base address: 9000h

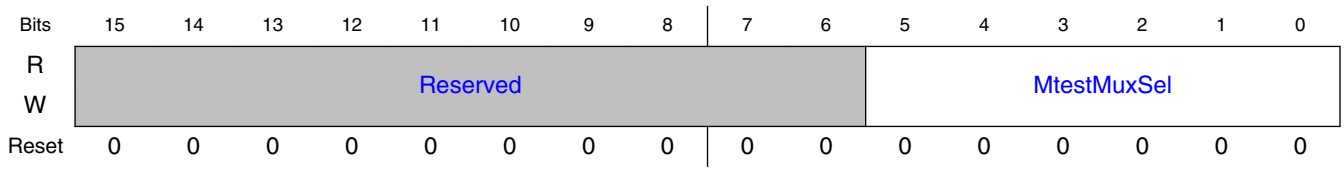
Offset	Register	Width (In bits)	Access	Reset value
34h	Digital Observation Pin control (MtestMuxSel)	16	RW	0000h
50h	Force Address/Command Tristate (Lanes A3-A0) (AForceTriCont)	16	RW	0000h
86h	Address TX impedance controls (ATxImpedance)	16	RW	03FFh
A6h	Address Loopback PRBS Error status for an entire ACX4 block (ATestPrbsErr)	16	RO	Table 9-4
AAh	Address TX slew rate and predriver controls (ATxSlewRate)	16	RW	07FFh
ACh	Address Loopback Test Result register (ATestPrbsErrCnt)	16	RO	Table 9-4
100h	Address/Command Delay, per pstate. (ATxDly_p0)	16	RW	0000h
20_0100h	Address/Command Delay, per pstate. (ATxDly_p1)	16	RW	0000h
40_0100h	Address/Command Delay, per pstate. (ATxDly_p2)	16	RW	0000h
60_0100h	Address/Command Delay, per pstate. (ATxDly_p3)	16	RW	0000h

9.4.3.1.2 Digital Observation Pin control (MtestMuxSel)

9.4.3.1.2.1 Offset

Register	Offset
MtestMuxSel	34h

9.4.3.1.2.2 Diagram



9.4.3.1.2.3 Fields

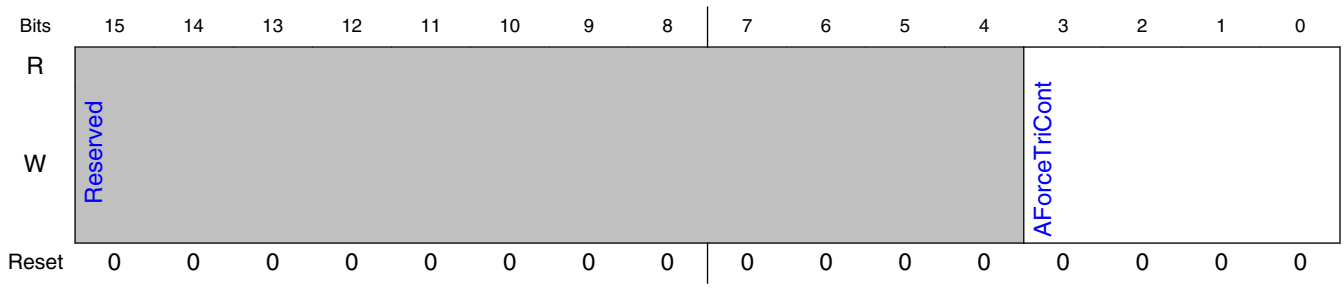
Field	Function
15-6 —	Reserved
5-0 MtestMuxSel	Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Encoding 6'h0 --> Drive 0 from this chiplet (allows flat 'OR' of pass-through information) Encoding 6'h01:1f --> Select local data from AC/DBYTE/MASTER macro Encoding 6'h20 --> Reserved (Drive 0 from macro into mux; Drive 0 from PUB synth logic path into mux) Encoding 6'h21:3f --> Select local data from PUB AC/DBYTE/MASTER synthesized logic Note: See PUB documentation for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.

9.4.3.1.3 Force Address/Command Tristate (Lanes A3-A0) (AForceTriCont)

9.4.3.1.3.1 Offset

Register	Offset
AForceTriCont	50h

9.4.3.1.3.2 Diagram



9.4.3.1.3.3 Fields

Field	Function
15-4 —	Reserved
3-0 AForceTriCont	Force tristate control, per-lane, of the ACX4 instance controlled by this register Setting this register will cause the PHY to tristate the target lane when dfi_init_complete==1 Bit [0] = controls lane 0 of the target ACX4 block Bit [1] = controls lane 1 of the target ACX4 block Bit [2] = controls lane 2 of the target ACX4 block Bit [3] = controls lane 3 of the target ACX4 block

9.4.3.1.4 Address TX impedance controls (ATxImpedance)

9.4.3.1.4.1 Offset

Register	Offset
ATxImpedance	86h

9.4.3.1.4.2 Diagram

9.4.3.1.4.3 Fields

Field	Function
15-10	Reserved

Table continues on the next page...

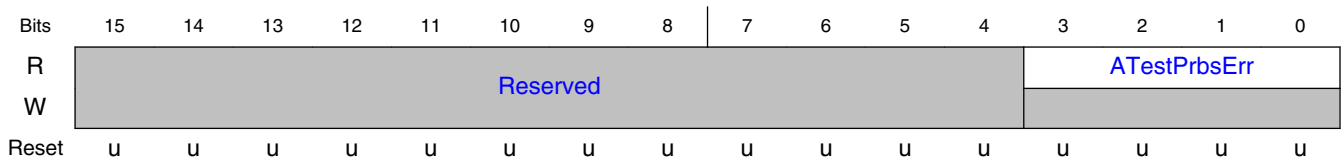
Field	Function
—	
9-5 ADrvStrenN	5 bit bus used to select the target pull down output impedance. 5 bit bus used to select the target pull down output impedance. Please Refer to Technology specific PHY DATABOOK for supported options Connects to the DrvStren pins of the driver thus: Csr_DrvStren120N[5:0] = csrADrvStrenN[4:0],1'b1 00000 = 120.0 Ohm 00001 = 60.0 Ohm 00011 = 40.0 Ohm 00111 = 30.0 Ohm 01111 = 24.0 Ohm 11111 = 20.0 Ohm
4-0 ADrvStrenP	5 bit bus used to select the target pull up output impedance. 5 bit bus used to select the target pull up output impedance. Please Refer to Technology specific PHY DATABOOK for supported options Connects to the DrvStren pins of the driver thus: Csr_DrvStren120P[5:0] = csrADrvStrenP[4:0],1'b1 00000 = 120.0 Ohm 00001 = 60.0 Ohm 00011 = 40.0 Ohm 00111 = 30.0 Ohm 01111 = 24.0 Ohm 11111 = 20.0 Ohm

9.4.3.1.5 Address Loopback PRBS Error status for an entire ACX4 block (ATestPrbsErr)

9.4.3.1.5.1 Offset

Register	Offset
ATestPrbsErr	A6h

9.4.3.1.5.2 Diagram



9.4.3.1.5.3 Fields

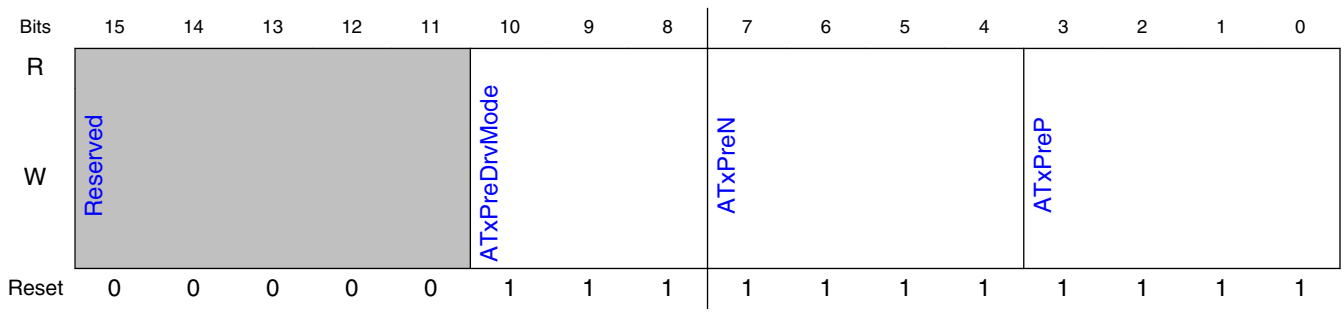
Field	Function
15-4 —	Reserved
3-0 ATestPrbsErr	Overall error indicator for each prbs bump checker. Overall error indicator for each prbs bump checker. Bit[0] = Lane0 Error Status 1 = errors found, 0 = no errors Bit[1] = Lane1 Error Status Bit[2] = Lane2 Error Status Bit[3] = Lane3 Error Status

9.4.3.1.6 Address TX slew rate and predriver controls (ATxSlewRate)

9.4.3.1.6.1 Offset

Register	Offset
ATxSlewRate	AAh

9.4.3.1.6.2 Diagram



9.4.3.1.6.3 Fields

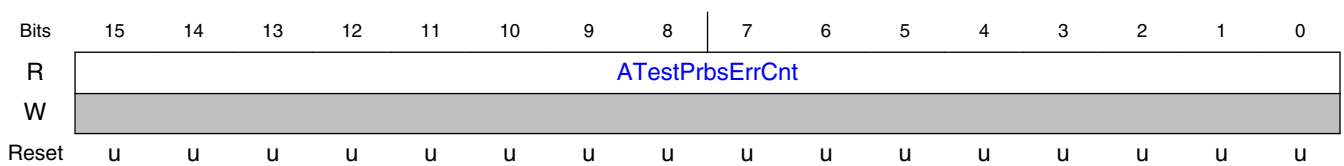
Field	Function
15-11 —	Reserved
10-8 ATxPreDrvMode	Controls predrivers to adjust timing of turn-on and turn-off of pull-up and pull-down segments.
7-4 ATxPreN	4 bit binary trim for the driver pull down slew rate. 4 bit binary trim for the driver pull down slew rate. 4'b0000 has a slower slew rate than 4'b1111
3-0 ATxPreP	4 bit binary trim for the driver pull up slew rate. 4 bit binary trim for the driver pull up slew rate. 4'b0000 has a slower slew rate than 4'b1111

9.4.3.1.7 Address Loopback Test Result register (ATestPrbsErrCnt)

9.4.3.1.7.1 Offset

Register	Offset
ATestPrbsErrCnt	ACh

9.4.3.1.7.2 Diagram



9.4.3.1.7.3 Fields

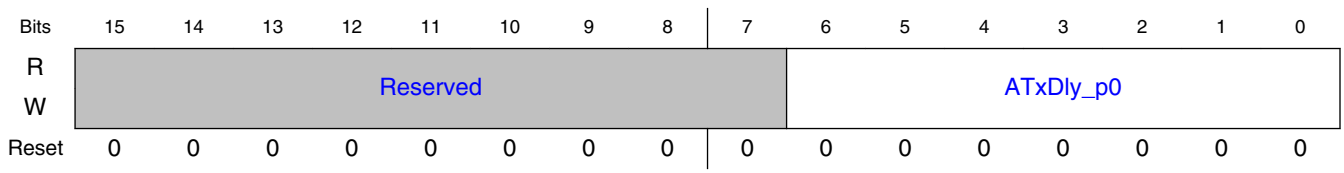
Field	Function
15-0 ATestPrbsErrCnt	Overall error indicator for each prbs bump checker. Overall error indicator for each prbs bump checker. Bit [3:0] = Lane0 Error Count Bit [7:4] = Lane1 Error Count Bit [11:8] = Lane2 Error Count Bit [15:12] = Lane3 Error Count

9.4.3.1.8 Address/Command Delay, per pstate. (ATxDly_p0)

9.4.3.1.8.1 Offset

Register	Offset
ATxDly_p0	100h

9.4.3.1.8.2 Diagram



9.4.3.1.8.3 Fields

Field	Function
15-7 —	Reserved
6-0 ATxDly_p0	Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. For DDR3/DDR4, it is recommended that address and command ACX4 have it ATxDly=0x00 The four signals generated by an ACX4 will have common timing. ATxDly[6] is the coarse delay, ie one unit of delay is 1 UI. ATxDly[5] is reserved. ATxDly[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. The target ACX4 used for memory clock generation, must have its ATxDly configured to 0x00.

9.4.3.1.9 Address/Command Delay, per pstate. (ATxDly_p1)

9.4.3.1.9.1 Offset

Register	Offset
ATxDly_p1	20_0100h

9.4.3.1.9.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ATxDly_p1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.1.9.3 Fields

Field	Function
15-7 —	Reserved
6-0 ATxDly_p1	Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. For DDR3/DDR4, it is recommended that address and command ACX4 have it ATxDly=0x00 The four signals generated by an ACX4 will have common timing. ATxDly[6] is the coarse delay, ie one unit of delay is 1 UI. ATxDly[5] is reserved. ATxDly[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. The target ACX4 used for memory clock generation, must have its ATxDly configured to 0x00.

9.4.3.1.10 Address/Command Delay, per pstate. (ATxDly_p2)

9.4.3.1.10.1 Offset

Register	Offset
ATxDly_p2	40_0100h

9.4.3.1.10.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								ATxDly_p2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.1.10.3 Fields

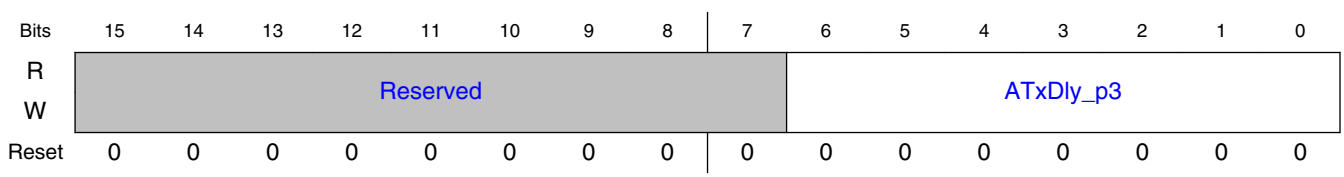
Field	Function
15-7 —	Reserved
6-0 ATxDly_p2	Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. For DDR3/DDR4, it is recommended that address and command ACX4 have it ATxDly=0x00 The four signals generated by an ACX4 will have common timing. ATxDly[6] is the coarse delay, ie one unit of delay is 1 UI. ATxDly[5] is reserved. ATxDly[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. The target ACX4 used for memory clock generation, must have its ATxDly configured to 0x00.

9.4.3.1.11 Address/Command Delay, per pstate. (ATxDly_p3)

9.4.3.1.11.1 Offset

Register	Offset
ATxDly_p3	60_0100h

9.4.3.1.11.2 Diagram



9.4.3.1.11.3 Fields

Field	Function
15-7 —	Reserved
6-0 ATxDly_p3	Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. Trained for LPDDR3/4 to generate timed address and command signals to the DRAMs, per ACX4. For DDR3/DDR4, it is recommended that address and command ACX4 have it ATxDly=0x00 The four signals generated by an ACX4 will have common timing.

Field	Function
	ATxDly[6] is the coarse delay, ie one unit of delay is 1 UI. ATxDly[5] is reserved. ATxDly[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. The target ACX4 used for memory clock generation, must have its ATxDly configured to 0x00.

9.4.3.2 DWC_DDRPHYA_APBONLY register descriptions

9.4.3.2.1 DWC_DDRPHYA_APBONLY Memory map

DDR4/3 PHY address block

DWC_DDRPHYA_APBONLY0 base address: D_0000h

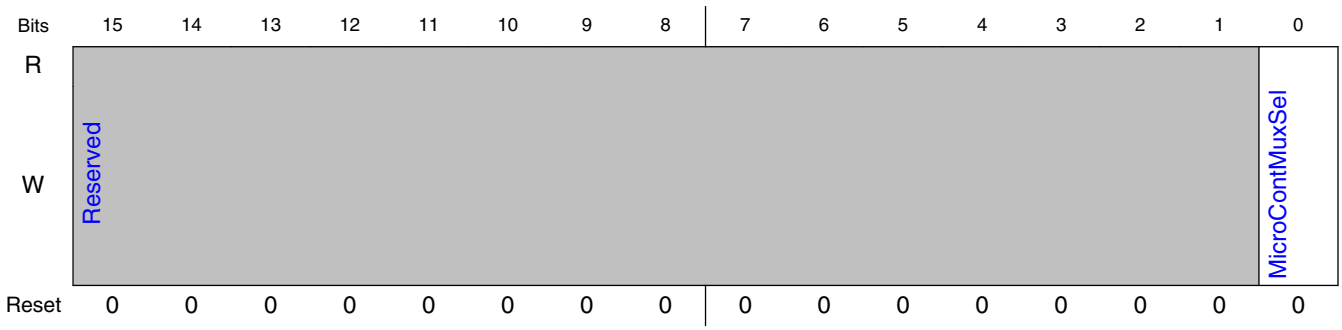
Offset	Register	Width (In bits)	Access	Reset value
0h	PMU Config Mux Select (MicroContMuxSel)	16	RW	0000h
8h	PMU/Controller Protocol - Controller Read-only Shadow (UctShadowRegs)	16	RO	Table 9-4
60h	Reserved for future use. (DctWriteOnly)	16	RW	0000h
62h	DCT downstream mailbox protocol CSR. (DctWriteProt)	16	RW	0001h
64h	Read-only view of the csr UctDatWriteOnly (UctWriteOnlyShadow)	16	RO	Table 9-4
68h	Read-only view of the csr UctDatWriteOnly (UctDatWriteOnlyShadow)	16	RO	Table 9-4
6Eh	Number of DfiClk ticks required for valid csr Rd Data. (DfiCfgRdDataValidTicks)	16	RW	0006h
132h	Controls reset and clock shutdown on the local microcontroller (MicroReset)	16	RW	0001h
1F4h	dfi_init_complete - Controller Read-only Shadow (DfiInitCompleteShadow)	16	RO	Table 9-4

9.4.3.2.2 PMU Config Mux Select (MicroContMuxSel)

9.4.3.2.2.1 Offset

Register	Offset
MicroContMuxSel	0h

9.4.3.2.2.2 Diagram



9.4.3.2.2.3 Fields

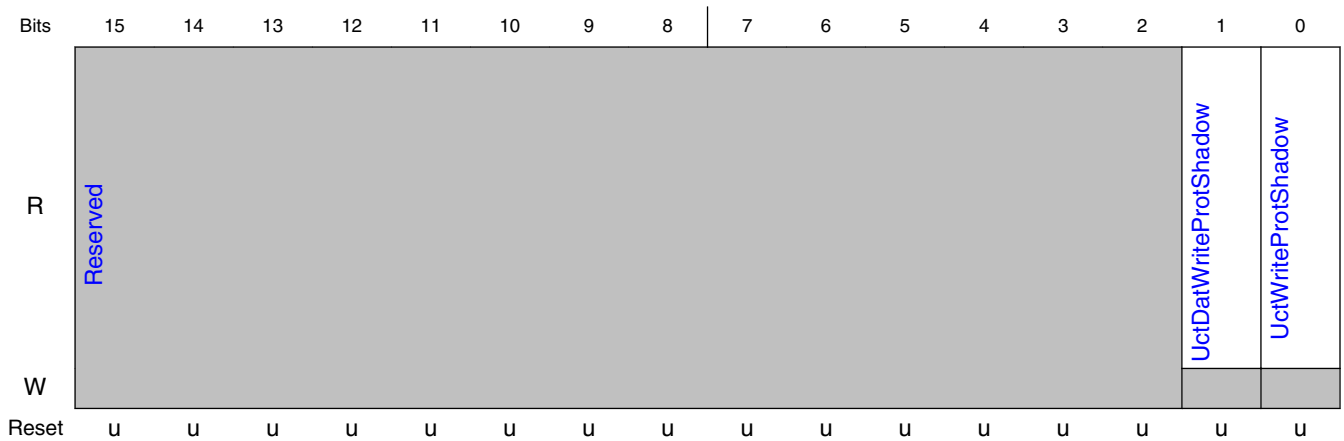
Field	Function
15-1 —	Reserved
0 MicroContMuxSel	This register controls access to the PHY configuration registers. This register controls access to the PHY configuration registers. 1 = MicroController/PHY Init Engine has control of csr bus. 0 = MicroController/PHY Init Engine csr requests are ignored.

9.4.3.2.3 PMU/Controller Protocol - Controller Read-only Shadow (UctShadowRegs)

9.4.3.2.3.1 Offset

Register	Offset
UctShadowRegs	8h

9.4.3.2.3.2 Diagram



9.4.3.2.3.3 Fields

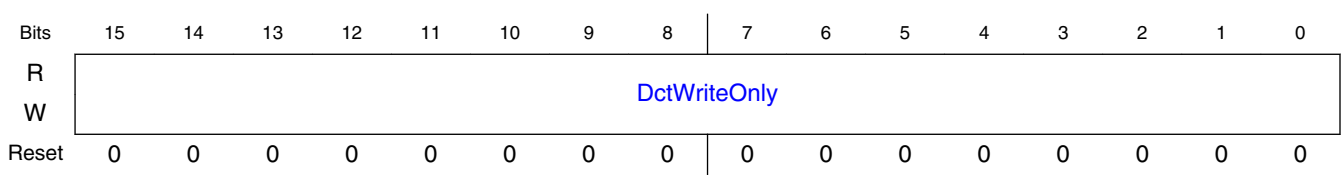
Field	Function
15-2 —	Reserved
1 UctDatWriteProtShadow	Reserved for future use.
0 UctWriteProtShadow	When set to 0, the PMU has a message for the user

9.4.3.2.4 Reserved for future use. (DctWriteOnly)

9.4.3.2.4.1 Offset

Register	Offset
DctWriteOnly	60h

9.4.3.2.4.2 Diagram



9.4.3.2.4.3 Fields

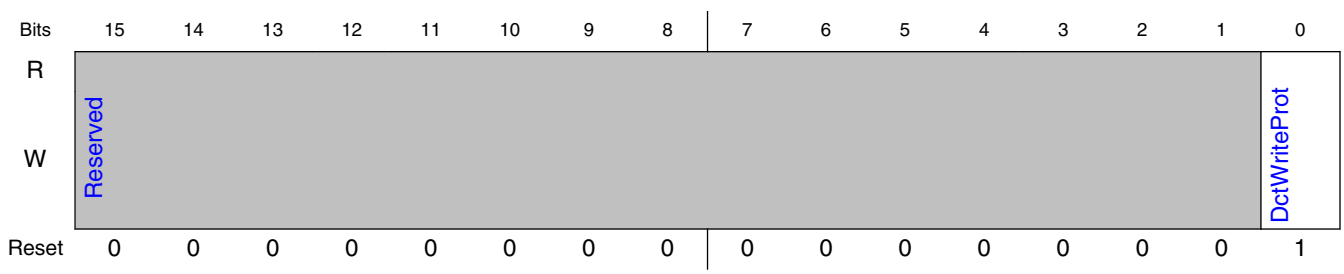
Field	Function
15-0 DctWriteOnly	Reserved for future use.

9.4.3.2.5 DCT downstream mailbox protocol CSR. (DctWriteProt)

9.4.3.2.5.1 Offset

Register	Offset
DctWriteProt	62h

9.4.3.2.5.2 Diagram



9.4.3.2.5.3 Fields

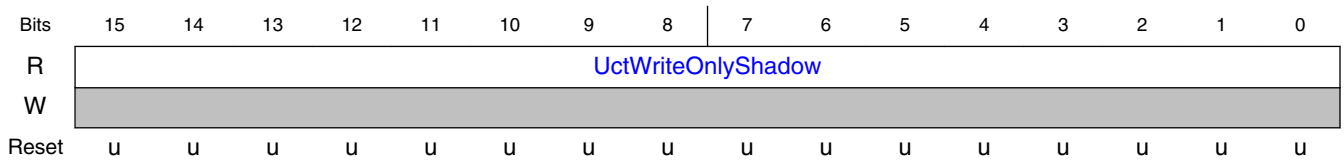
Field	Function
15-1 —	Reserved
0 DctWriteProt	By setting this register to 0, the user acknowledges the receipt of the message.

9.4.3.2.6 Read-only view of the csr UctDatWriteOnly (UctWriteOnlyShadow)

9.4.3.2.6.1 Offset

Register	Offset
UctWriteOnlyShadow	64h

9.4.3.2.6.2 Diagram



9.4.3.2.6.3 Fields

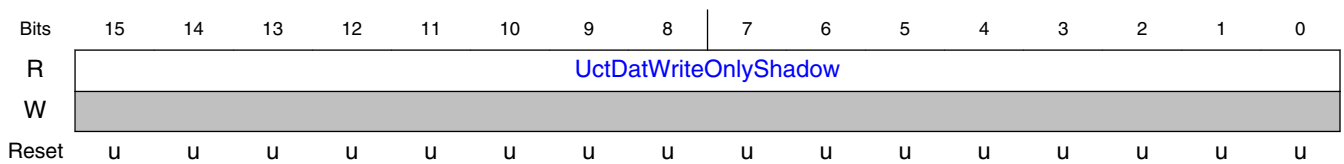
Field	Function
15-0 UctWriteOnlyShadow	Used to pass the message ID for major messages. Used to pass the message ID for major messages. Also used to pass the lower 16 bits for streaming messages.

9.4.3.2.7 Read-only view of the csr UctDatWriteOnly (UctDatWriteOnlyShadow)

9.4.3.2.7.1 Offset

Register	Offset
UctDatWriteOnlyShadow	68h

9.4.3.2.7.2 Diagram



9.4.3.2.7.3 Fields

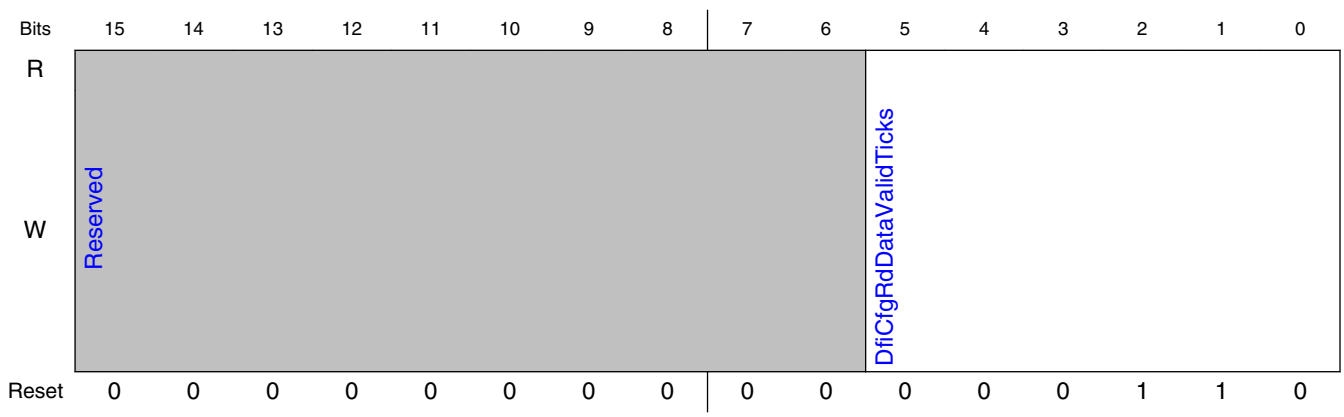
Field	Function
15-0 UctDatWriteOnlyShadow	Used to pass the upper 16 bits for streaming messages. Used to pass the upper 16 bits for streaming messages. Not used in passing major messages.

9.4.3.2.8 Number of DfiClk ticks required for valid csr Rd Data. (DfiCfgRdDataValidTicks)

9.4.3.2.8.1 Offset

Register	Offset
DfiCfgRdDataValidTicks	6Eh

9.4.3.2.8.2 Diagram



9.4.3.2.8.3 Fields

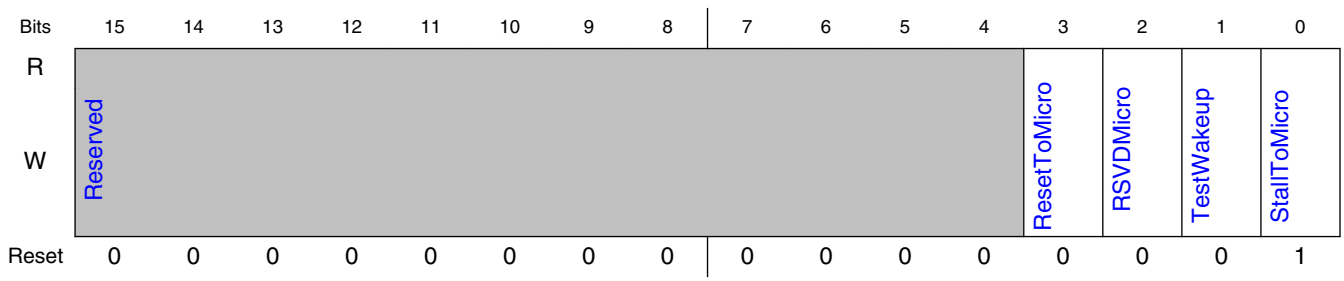
Field	Function
15-6 —	Reserved
5-0 DfiCfgRdDataValidTicks	Roundtrip delay of a register read access. Roundtrip delay of a register read access. This value must not be changed from its reset value.

9.4.3.2.9 Controls reset and clock shutdown on the local microcontroller (MicroReset)

9.4.3.2.9.1 Offset

Register	Offset
MicroReset	132h

9.4.3.2.9.2 Diagram



9.4.3.2.9.3 Fields

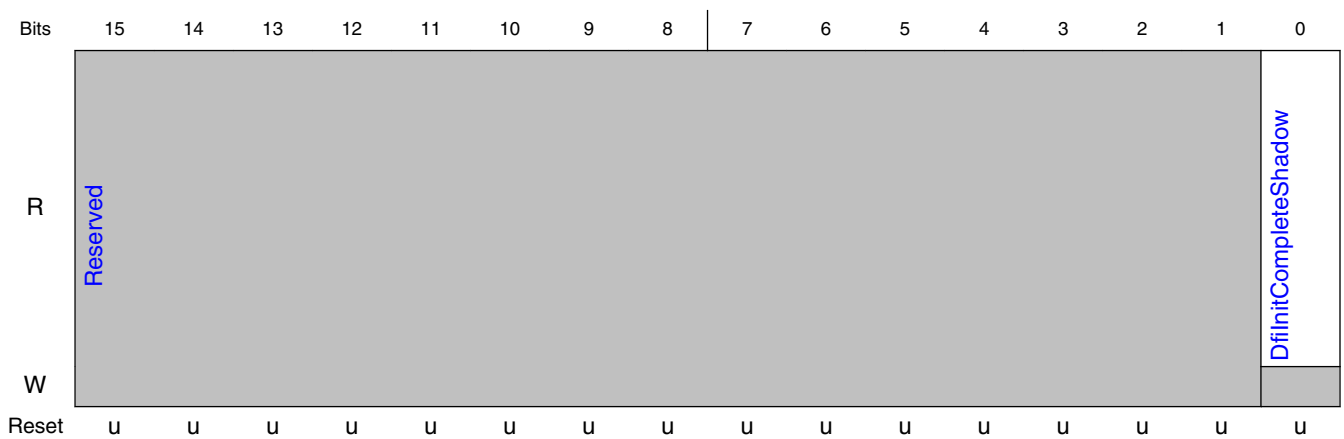
Field	Function
15-4 —	Reserved
3 ResetToMicro	Set this bit to apply synchronous reset to the microcontroller. Set this bit to apply synchronous reset to the microcontroller. Clear this bit to release reset to the microcontroller. This reset initializes the program counter to begin execution from the boot vector. This reset also clears the interrupt sticky bits.
2 RSVDMicro	RSVD
1 TestWakeup	Reserved Reserved. Must always be set to 0.
0 StallToMicro	Set this bit to stall the microcontroller by hardware. Set this bit to stall the microcontroller by hardware. Clear this bit to allow the microcontroller to continue executing from its current program counter location. Typically, this bit is used at power up to hold the program counter at the boot vector while BIOS loads the microcontroller program code. While stalled, the microcontroller clocks are gated off for power reduction.

9.4.3.2.10 dfi_init_complete - Controller Read-only Shadow (DfiInitCompleteShadow)

9.4.3.2.10.1 Offset

Register	Offset
DfiInitCompleteShadow	1F4h

9.4.3.2.10.2 Diagram



9.4.3.2.10.3 Fields

Field	Function
15-1 —	Reserved
0 DfiInitCompleteShadow	This csr presents a read-only view (a shadow) of the Register DfiInitComplete which is used by the sequencer to control the state of dfi_init_complete. This csr presents a read-only view (a shadow) of the Register DfiInitComplete which is used by the sequencer to control the state of dfi_init_complete. The value in this Register is not affected by the BlockSeq0BAck field of the SequencerOverride register. While the Register MicroContMuxSel is set, access to this Shadow register will not steal config bus bandwidth from the micro controller. That is polling won't have a performance penalty.

9.4.3.3 DWC_DDRPHYA_DBYTE register descriptions

9.4.3.3.1 DWC_DDRPHYA_DBYTE Memory map

DDR4/3 PHY address block

DWC_DDRPHYA_DBYTE0 base address: 1_0000h

DWC_DDRPHYA_DBYTE1 base address: 1_1000h

DWC_DDRPHYA_DBYTE2 base address: 1_2000h

DWC_DDRPHYA_DBYTE3 base address: 1_3000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DBYTE Module Disable (DbyteMiscMode)	16	RW	Table 9-4
34h	Digital Observation Pin control (MtestMuxSel)	16	RW	0000h
40h	DFI MaxReadLatency (DFIMRL_p0)	16	RW	0006h
60h - 1060h	VrefDAC1 control for DQ Receiver (used only when DFE is enabled in DDR4) (VrefDAC1_r0 - VrefDAC1_r8)	16	RW	0000h
80h - 1080h	VrefDAC0 control for DQ Receiver (VrefDAC0_r0 - VrefDAC0_r8)	16	RW	0000h
82h - 60_0282h	Data TX impedance controls (TxImpedanceCtrl0_b0_p0 - TxImpedanceCtrl0_b1_p3)	16	RW	0FFFh
86h - 60_0286h	Dq/Dqs receiver control (DqDqsRcvCntrl_b0_p0 - DqDqsRcvCntrl_b1_p3)	16	RW	05B0h
90h - 60_0090h	Tx dq driver equalization mode controls. (TxEqualizationMode_p0 - TxEqualizationMode_p3)	16	RW	0000h
92h - 60_0292h	TX impedance controls (TxImpedanceCtrl1_b0_p0 - TxImpedanceCtrl1_b1_p3)	16	RW	0FFFh
94h	Dq/Dqs receiver control (DqDqsRcvCntrl1)	16	RW	0400h
96h - 60_0296h	TX equalization impedance controls (TxImpedanceCtrl2_b0_p0 - TxImpedanceCtrl2_b1_p3)	16	RW	0000h
98h - 60_0098h	Dq/Dqs receiver control (DqDqsRcvCntrl2_p0 - DqDqsRcvCntrl2_p3)	16	RW	0000h
9Ah - 60_029Ah	TX ODT driver strength control (TxOdtDrvStren_b0_p0 - TxOdtDrvStren_b1_p3)	16	RW	0000h
ACh	Status of RX FIFO Consistency Checks (RxFifoCheckStatus)	16	RO	Table 9-4
AEh	Contains the captured values associated with an RxFifo consistency error (RxFifoCheckErrValues)	16	RO	Table 9-4
B0h	Data Receive FIFO Pointer Values (RxFifoInfo)	16	RO	Table 9-4
B2h	RX FIFO visibility (RxFifoVisibility)	16	RW	0000h
B4h	RX FIFO contents, lane[3:0] (RxFifoContentsDQ3210)	16	RO	Table 9-4
B6h	RX FIFO contents, lane[7:4] (RxFifoContentsDQ7654)	16	RO	Table 9-4
B8h	RX FIFO contents, dbi (RxFifoContentsDBI)	16	RO	Table 9-4

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
BEh - 60_02BEh	TX slew rate controls (TxSlewRate_b0_p0 - TxSlewRate_b1_p3)	16	RW	07FFh
D0h - 10D0h	Read DQ per-bit BDL delay (Timing Group 0). (RxPBDlyTg0_r0 - RxPBDlyTg0_r8)	16	RW	0000h
D2h - 10D2h	Read DQ per-bit BDL delay (Timing Group 1). (RxPBDlyTg1_r0 - RxPBDlyTg1_r8)	16	RW	0000h
D4h - 10D4h	Read DQ per-bit BDL delay (Timing Group 2). (RxPBDlyTg2_r0 - RxPBDlyTg2_r8)	16	RW	0000h
D6h - 10D6h	Read DQ per-bit BDL delay (Timing Group 3). (RxPBDlyTg3_r0 - RxPBDlyTg3_r8)	16	RW	0000h
100h - 60_0300h	Trained Receive Enable Delay (For Timing Group 0) (RxEnDlyTg0_u0_p0 - RxEnDlyTg0_u1_p3)	16	RW	0100h
102h - 60_0302h	Trained Receive Enable Delay (For Timing Group 1) (RxEnDlyTg1_u0_p0 - RxEnDlyTg1_u1_p3)	16	RW	0100h
104h - 60_0304h	Trained Receive Enable Delay (For Timing Group 2) (RxEnDlyTg2_u0_p0 - RxEnDlyTg2_u1_p3)	16	RW	0100h
106h - 60_0306h	Trained Receive Enable Delay (For Timing Group 3) (RxEnDlyTg3_u0_p0 - RxEnDlyTg3_u1_p3)	16	RW	0100h
118h - 60_0318h	Trained Read DQS to RxClk Delay (Timing Group DEST=0). (RxClkDlyTg0_u0_p0 - RxClkDlyTg0_u1_p3)	16	RW	0010h
11Ah - 60_031Ah	Trained Read DQS to RxClk Delay (Timing Group DEST=1). (RxClkDlyTg1_u0_p0 - RxClkDlyTg1_u1_p3)	16	RW	0010h
11Ch - 60_031Ch	Trained Read DQS to RxClk Delay (Timing Group DEST=2). (RxClkDlyTg2_u0_p0 - RxClkDlyTg2_u1_p3)	16	RW	0010h
11Eh - 60_031Eh	Trained Read DQS to RxClk Delay (Timing Group DEST=3). (RxClkDlyTg3_u0_p0 - RxClkDlyTg3_u1_p3)	16	RW	0010h
140h - 14Eh	Maps Phy DQ lane to memory DQ0 (Dq0LnSel - Dq7LnSel)	16	RW	0000h
180h - 60_1180h	Write DQ Delay (Timing Group 0). (TxDqDlyTg0_r0_p0 - TxDqDlyTg0_r8_p3)	16	RW	0010h
182h - 60_1182h	Write DQ Delay (Timing Group 1). (TxDqDlyTg1_r0_p0 - TxDqDlyTg1_r8_p3)	16	RW	0010h
184h - 60_1184h	Write DQ Delay (Timing Group 2). (TxDqDlyTg2_r0_p0 - TxDqDlyTg2_r8_p3)	16	RW	0010h
186h - 60_1186h	Write DQ Delay (Timing Group 3). (TxDqDlyTg3_r0_p0 - TxDqDlyTg3_r8_p3)	16	RW	0010h
1A0h - 60_03A0h	Write DQS Delay (Timing Group DEST=0). (TxDqsDlyTg0_u0_p0 - TxDqsDlyTg0_u1_p3)	16	RW	0100h
1A2h - 60_03A2h	Write DQS Delay (Timing Group DEST=1). (TxDqsDlyTg1_u0_p0 - TxDqsDlyTg1_u1_p3)	16	RW	0100h
1A4h - 60_03A4h	Write DQS Delay (Timing Group DEST=2). (TxDqsDlyTg2_u0_p0 - TxDqsDlyTg2_u1_p3)	16	RW	0100h
1A6h - 60_03A6h	Write DQS Delay (Timing Group DEST=3). (TxDqsDlyTg3_u0_p0 - TxDqsDlyTg3_u1_p3)	16	RW	0100h
1C8h	Debug status of the DBYTE LCDL (DxLcdlStatus)	16	RO	Table 9-4
20_0040h	DFI MaxReadLatency (DFIMRL_p1)	16	RW	0006h
40_0040h	DFI MaxReadLatency (DFIMRL_p2)	16	RW	0006h

Table continues on the next page...

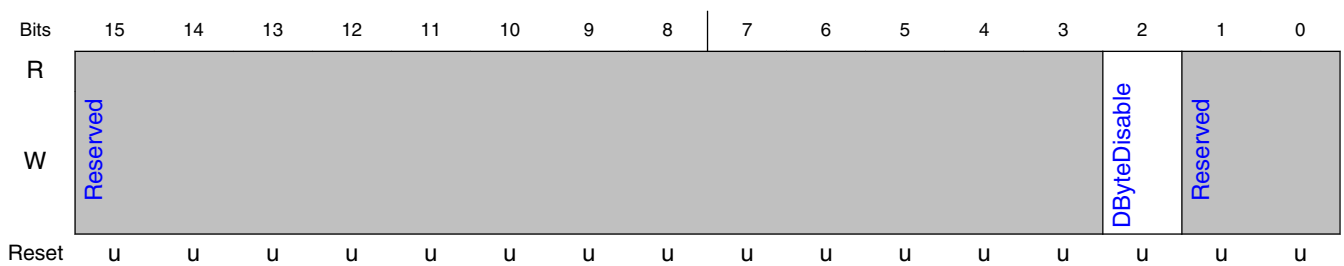
Offset	Register	Width (In bits)	Access	Reset value
40_015Ch	DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg0_p2)	16	RW	0000h
40_015Eh	DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg1_p2)	16	RW	0000h
60_0040h	DFI MaxReadLatency (DFIMRL_p3)	16	RW	0006h
60_015Ch	DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg0_p3)	16	RW	0000h
60_015Eh	DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg1_p3)	16	RW	0000h

9.4.3.3.2 DBYTE Module Disable (DbyteMiscMode)

9.4.3.3.2.1 Offset

Register	Offset
DbyteMiscMode	0h

9.4.3.3.2.2 Diagram



9.4.3.3.2.3 Fields

Field	Function
15-3 —	Reserved
2 DByteDisable	Controls whether this DBYTE module is disabled. Controls whether this DBYTE module is disabled. If this DBYTE module is not enabled, it receives no clocks and remains in reset. 0 - Enable this DBYTE module 1 - Disable this DBYTE module

Table continues on the next page...

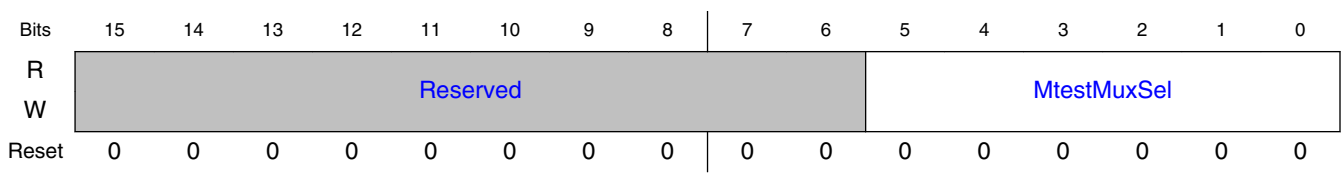
Field	Function
	This field should only be changed PHY initialization step C.
1-0 —	Reserved

9.4.3.3.3 Digital Observation Pin control (MtestMuxSel)

9.4.3.3.3.1 Offset

Register	Offset
MtestMuxSel	34h

9.4.3.3.3.2 Diagram



9.4.3.3.3.3 Fields

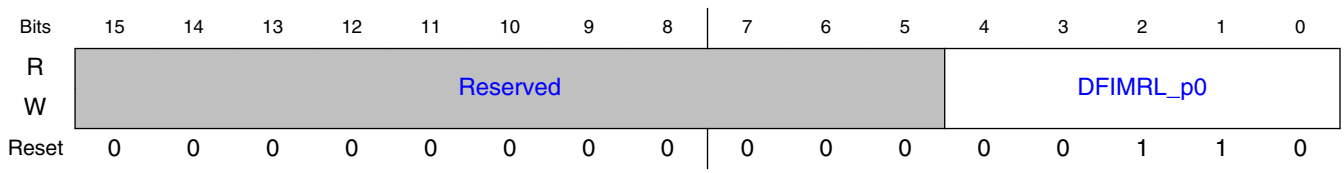
Field	Function
15-6 —	Reserved
5-0 MtestMuxSel	Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Encoding 6'h0 --> Drive 0 from this chiplet (allows flat 'OR' of pass-through information) Encoding 6'h01:1f --> Select local data from AC/DBYTE/MASTER macro Encoding 6'h20 --> Reserved (Drive 0 from macro into mux; Drive 0 from PUB synth logic path into mux) Encoding 6'h21:3f --> Select local data from PUB AC/DBYTE/MASTER synthesized logic Note: See PUB documentation for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.

9.4.3.3.4 DFI MaxReadLatency (DFIMRL_p0)

9.4.3.3.4.1 Offset

Register	Offset
DFIMRL_p0	40h

9.4.3.3.4.2 Diagram



9.4.3.3.4.3 Fields

Field	Function
15-5 —	Reserved
4-0 DFIMRL_p0	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This is the value, in units of two mem clocks, between dfi_rddata_en and dfi_rddata_valid A unit change in the LSB is a change in MRL of two mem clocks.

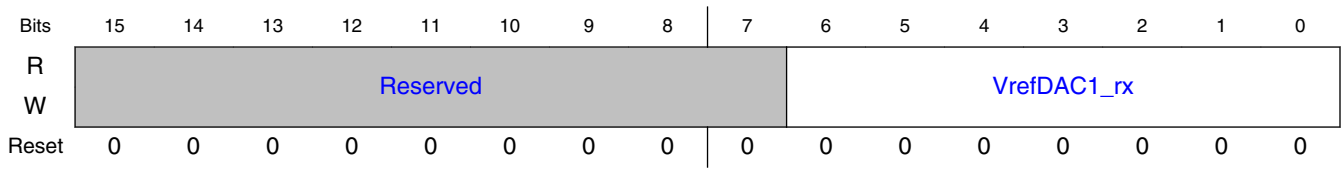
9.4.3.3.5 VrefDAC1 control for DQ Receiver (used only when DFE is enabled in DDR4) (VrefDAC1_r0 - VrefDAC1_r8)

9.4.3.3.5.1 Offset

For x = 0 to 8:

Register	Offset
VrefDAC1_rx	60h + (x × 200h)

9.4.3.3.5.2 Diagram



9.4.3.3.5.3 Fields

Field	Function
15-7 —	Reserved
6-0 VrefDAC1_rx	VrefDAC1 controls the alternate VREF setting for DFE (used only when DFE is enabled in DDR4) DAC control for rxdq cell internal VREF, trained by Firmware The VREF generators have different ranges, depending on the Mission Mode settings for DqDqsRcvCntrl::MajorMode,DqDqsRcvCntrl::ExtVrefRange 011,0 :: VREF = VDDQ*(0. VrefDAC1 controls the alternate VREF setting for DFE (used only when DFE is enabled in DDR4) DAC control for rxdq cell internal VREF, trained by Firmware The VREF generators have different ranges, depending on the Mission Mode settings for DqDqsRcvCntrl::MajorMode,DqDqsRcvCntrl::ExtVrefRange 011,0 :: VREF = VDDQ*(0.510 + VrefDAC1[6:0]*0.00345) 011,1 :: VREF = VDDQ*(0.453 + VrefDAC1[6:0]*0.00385) non-enumerated encodings are reserved Register Block Offset Address 0x030 is the VrefDAC1 for lane0. Register Block Offset Address 0x130 is the VrefDAC1 for lane1. Register Block Offset Address 0x230 is the VrefDAC1 for lane2. Register Block Offset Address 0x330 is the VrefDAC1 for lane3. Register Block Offset Address 0x430 is the VrefDAC1 for lane4. Register Block Offset Address 0x530 is the VrefDAC1 for lane5. Register Block Offset Address 0x630 is the VrefDAC1 for lane6. Register Block Offset Address 0x730 is the VrefDAC1 for lane7. Register Block Offset Address 0x830 is the VrefDAC1 for laneDBI.

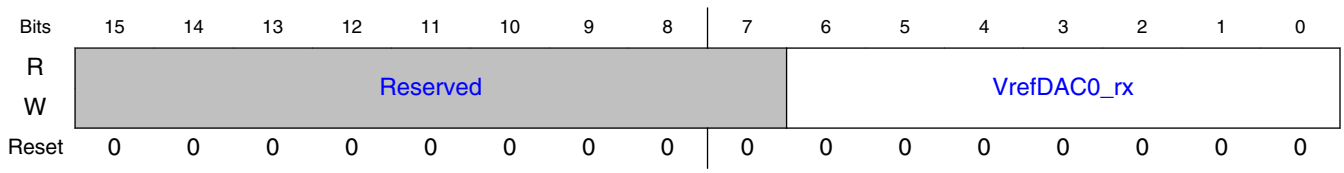
9.4.3.3.6 VrefDAC0 control for DQ Receiver (VrefDAC0_r0 - VrefDAC0_r8)

9.4.3.3.6.1 Offset

For x = 0 to 8:

Register	Offset
VrefDAC0_rx	80h + (x × 200h)

9.4.3.3.6.2 Diagram



9.4.3.3.6.3 Fields

Field	Function
15-7 —	Reserved
6-0 VrefDAC0_rx	PHY RX VREF DAC control for rxdq cell internal VREF, (used only when 2D training is enabled in LPDDR4,DDR4) DAC control for rxdq cell internal VREF, trained by Firmware The VREF generators have different ranges, depending on the Mission Mode settings for DqDqsRcvCntrl::MajorMode,DqDqsRcvCntrl::ExtVrefRange 000,0 :: VREF = VDDQ*(0. PHY RX VREF DAC control for rxdq cell internal VREF, (used only when 2D training is enabled in LPDDR4,DDR4) DAC control for rxdq cell internal VREF, trained by Firmware The VREF generators have different ranges, depending on the Mission Mode settings for DqDqsRcvCntrl::MajorMode,DqDqsRcvCntrl::ExtVrefRange 000,0 :: VREF = VDDQ*(0.287 + VrefDAC0[6:0]*0.00330) 000,1 :: VREF = VDDQ*(0.250 + VrefDAC0[6:0]*0.00385) 011,0 :: VREF = VDDQ*(0.510 + VrefDAC0[6:0]*0.00345) 011,1 :: VREF = VDDQ*(0.453 + VrefDAC0[6:0]*0.00385) 010,0 :: VREF = VDDQ*(0.047 + VrefDAC0[6:0]*0.00367) non-enumerated encodings are reserved Register Block Offset Address 0x040 is the VrefDAC0 for lane0. Register Block Offset Address 0x140 is the VrefDAC0 for lane1. Register Block Offset Address 0x240 is the VrefDAC0 for lane2. Register Block Offset Address 0x340 is the VrefDAC0 for lane3. Register Block Offset Address 0x440 is the VrefDAC0 for lane4. Register Block Offset Address 0x540 is the VrefDAC0 for lane5. Register Block Offset Address 0x640 is the VrefDAC0 for lane6. Register Block Offset Address 0x740 is the VrefDAC0 for lane7. Register Block Offset Address 0x840 is the VrefDAC0 for laneDBI.

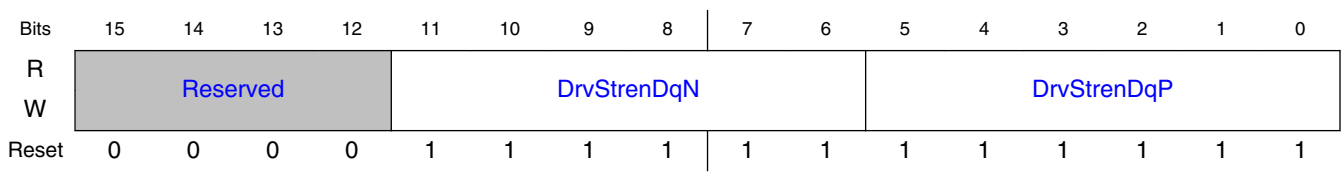
9.4.3.3.7 Data TX impedance controls (TxImpedanceCtrl0_b0_p0 - TxImpedanceCtrl0_b1_p3)

9.4.3.3.7.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxImpedanceCtrl0_bn_px	82h + (x × 20_0000h) + (n × 200h)

9.4.3.3.7.2 Diagram



9.4.3.3.7.3 Fields

Field	Function
15-12 —	Reserved
11-6 DrvStrenDqN	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance. Connects to the DrvStren pins of the driver thus: Csr_DrvStren120N[2:0],Csr_DrvStren240N[1:0],Csr_DrvStren480N[0:0] = csrDrvStrenDqN See table of csr DrvStrenDqP for list of possible impedances.
5-0 DrvStrenDqP	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance. Connects to the DrvStren pins of the driver thus: Csr_DrvStren120P[2:0], Csr_DrvStren240P[1:0],Csr_DrvStren480P[0:0] = csrDrvStrenDqP 000000 - HiZ. 000001 - 480.0 000010 - 240.0

Field	Function
	000011 - 160.0 000110 - 120.0 000111 - 96.0 001010 - 80.0 001011 - 68.6 001110 - 60.0 001111 - 53.3 011010 - 48.0 011011 - 43.6 011110 - 40.0 011111 - 36.9 111010 - 34.3 111011 - 32.0 111110 - 30.0 111111 - 28.2

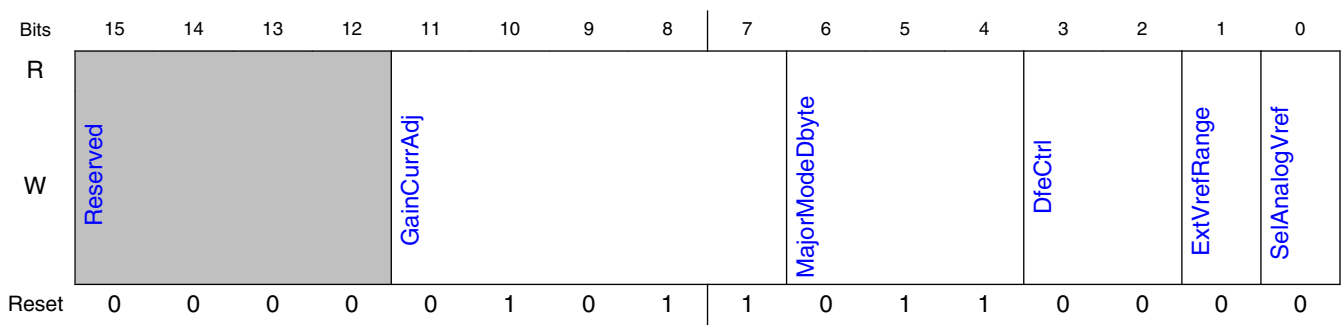
9.4.3.3.8 Dq/Dqs receiver control (DqDqsRcvCntrl_b0_p0 - DqDqsRcvCntrl_b1_p3)

9.4.3.3.8.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
DqDqsRcvCntrl_bn_px	86h + (x × 20_0000h) + (n × 200h)

9.4.3.3.8.2 Diagram



9.4.3.3.8.3 Fields

Field	Function
15-12 —	Reserved[12] unavailable - should not be set Reserved[13] for future RX circuit trim Reserved[14] Set to 1 to enable VIL/VIH testing of the DFE0 Amp Reserved[15] Set to 1 to enable VIL/VIH testing of the DFE1 Amp NOTE: Reserved[15:14] should not be asserted simultaneously
11-7 GainCurrAdj	Adjust gain current of RX amplifier stage. Adjust gain current of RX amplifier stage. It is recommended to use default values for this CSR.
6-4 MajorModeDbyte	Selects the major mode of operation for the receiver. Selects the major mode of operation for the receiver. These settings are determined by PHY Configuration based on DRAM protocol 000 - DDR3 001 - reserved (room for LPDDR3) 010 - LPDDR4 011 - DDR4 100 - DDR3 Low-power/low-speed 101 - reserved (LPDDR3 low-power/low-speed) 110 - LPDDR4 low-power/low-speed 111 - DDR4 Low-power/low-speed
3-2 DfeCtrl	DFE may be used with MajorModeDbyte=011 only 00 - DFE off 01 - DFE on 10 - Train DFE0 Amplifier 11 - Train DFE1 Amplifier These settings are determined by PHY Training FW and should not be overridden.
1 ExtVrefRange	Extends the range available in the local per-bit VREF generator.
0 SelAnalogVref	Setting this signal high will force the local per-bit VREF generator to pass the global VREFA to the samplers.

9.4.3.3.9 Tx dq driver equalization mode controls. (TxEqualizationMode_p0 - TxEqualizationMode_p3)

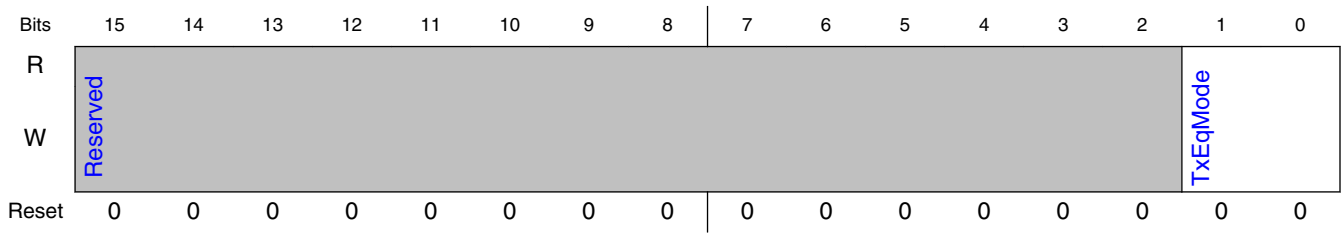
9.4.3.3.9.1 Offset

Register	Offset
TxEqualizationMode_p0	90h
TxEqualizationMode_p1	20_0090h
TxEqualizationMode_p2	40_0090h
TxEqualizationMode_p3	60_0090h

9.4.3.3.9.2 Function

Tx dq driver equalization mode controls. FFE, de-emphasis DDR43 PHY ONLY

9.4.3.3.9.3 Diagram



9.4.3.3.9.4 Fields

Field	Function
15-2 —	Reserved
1-0 TxEqMode	

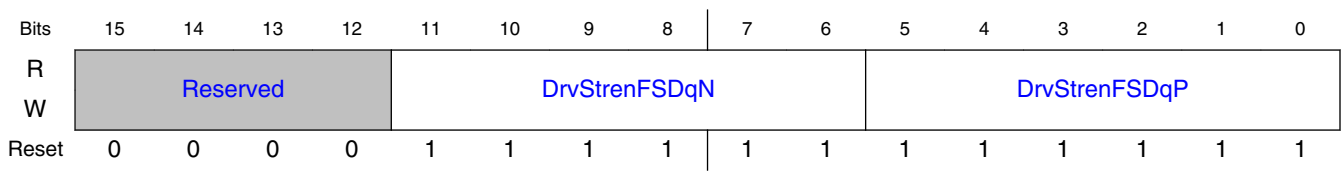
9.4.3.3.10 TX impedance controls (TxImpedanceCtrl1_b0_p0 - TxImpedanceCtrl1_b1_p3)

9.4.3.3.10.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxImpedanceCtrl1_bn_px	92h + (x × 20_0000h) + (n × 200h)

9.4.3.3.10.2 Diagram



9.4.3.3.10.3 Fields

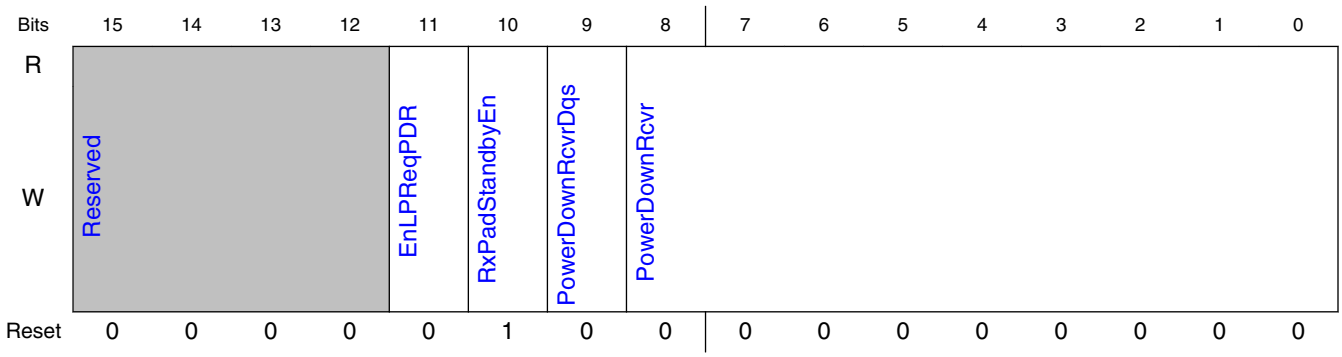
Field	Function
15-12 —	Reserved
11-6 DrvStrenFSDqN	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Connects to the DrvStren pins of the driver thus: Csr_DrvStrenFS120N[2:0], Csr_DrvStrenFS240N[1:0], Csr_DrvStrenFS480N[0:0] = csrDrvStrenDqN See table of csr DrvStrenFSDqP for list of possible impedances.
5-0 DrvStrenFSDqP	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Connects to the DrvStren pins of the driver thus: Csr_DrvStrenFS120P[2:0], Csr_DrvStrenFS240P[1:0], Csr_DrvStrenFS480P[0] = csrFSDrvStrenDq 000000 - HiZ. 000001 - 480.0 000010 - 240.0 000011 - 160.0 000110 - 120.0 000111 - 96.0 001010 - 80.0 001011 - 68.6 001110 - 60.0 001111 - 53.3 011010 - 48.0 011011 - 43.6 011110 - 40.0 011111 - 36.9 111010 - 34.3 111011 - 32.0 111110 - 30.0 111111 - 28.2

9.4.3.3.11 Dq/Dqs receiver control (DqDqsRcvCntrl1)

9.4.3.3.11.1 Offset

Register	Offset
DqDqsRcvCntrl1	94h

9.4.3.3.11.2 Diagram



9.4.3.3.11.3 Fields

Field	Function
15-12 —	Reserved
11 EnLPReqPDR	Reserved for future use
10 RxPadStandbyEn	Enables the rxdq/rxdqs StandBy power savings, per pad-group.
9 PowerDownRcvrDqs	Active high signal which powers down the receiver. Active high signal which powers down the receiver. After this pin is deasserted the receiver cannot be used for a minimum of 100 ns. This control powers down both the upper and lower DQS receivers. If Register X4TG[3:0]=0000, then the upper DQS receiver is powered down.
8-0 PowerDownRcvr	Active high signal which powers down the receiver. Active high signal which powers down the receiver. [7:0 -> dq[7:0], 8 -> dbi] After this pin is deasserted the receiver cannot be used for a minimum of 100 ns.

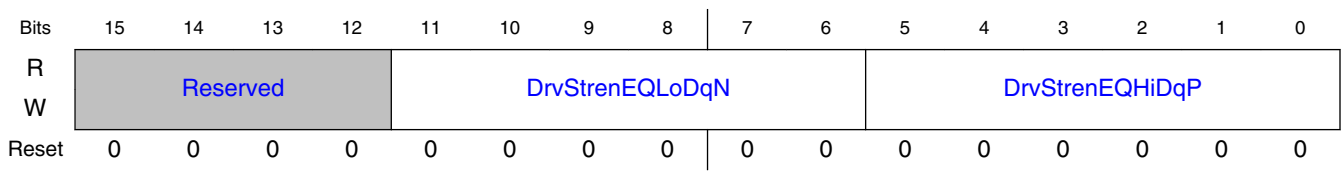
9.4.3.3.12 TX equalization impedance controls (TxImpedanceCtrl2_b0_p0 - TxImpedanceCtrl2_b1_p3)

9.4.3.3.12.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxImpedanceCtrl2_bn_px	96h + (x × 20_0000h) + (n × 200h)

9.4.3.3.12.2 Diagram



9.4.3.3.12.3 Fields

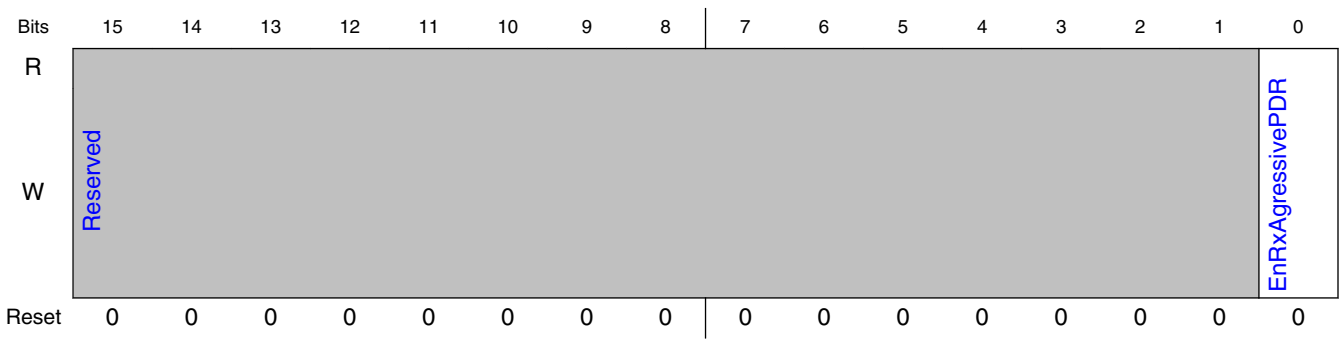
Field	Function
15-12 —	Reserved
11-6 DrvStrenEQLoDqN	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance used in equalization. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull down output impedance used in equalization. Connects to the DrvStren pins of the driver thus: Csr_DrvStrenEQLo120N[2:0], Csr_DrvStrenEQLo240N[1:0], Csr_DrvStrenEQLo480P[0] = csrDrvStrenEQLoDq
5-0 DrvStrenEQHiDqP	Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Please Refer to Technology specific PHY DATABOOK for supported options 6 bit bus used to select the target pull up output impedance used in equalization. Connects to the DrvStren pins of the driver thus: Csr_DrvStrenEQHi120P[2:0], Csr_DrvStrenEQHi240P[1:0], Csr_DrvStrenEQHi480P[0] = csrDrvStrenEQHiDq

9.4.3.3.13 Dq/Dqs receiver control (DqDqsRcvCntrl2_p0 - DqDqsRcvCntrl2_p3)

9.4.3.3.13.1 Offset

Register	Offset
DqDqsRcvCntrl2_p0	98h
DqDqsRcvCntrl2_p1	20_0098h
DqDqsRcvCntrl2_p2	40_0098h
DqDqsRcvCntrl2_p3	60_0098h

9.4.3.3.13.2 Diagram



9.4.3.3.13.3 Fields

Field	Function
15-1 —	Reserved
0 EnRxAgressivePDR	reserved

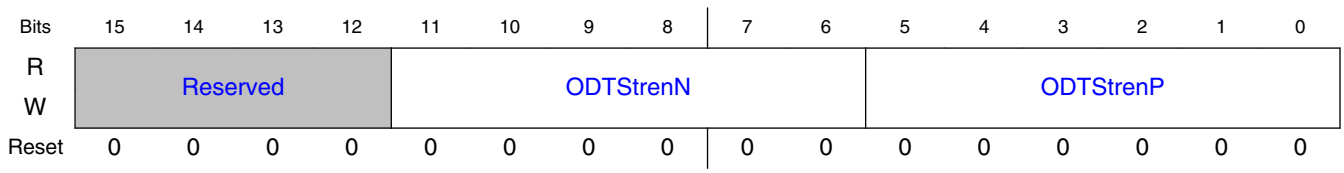
9.4.3.3.14 TX ODT driver strength control (TxOdtDrvStren_b0_p0 - TxOdtDrvStren_b1_p3)

9.4.3.3.14.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxOdtDrvStren_bn_px	9Ah + (x × 20_0000h) + (n × 200h)

9.4.3.3.14.2 Diagram



9.4.3.3.14.3 Fields

Field	Function
15-12 —	Reserved
11-6 ODTStrenN	Selects the ODT pull-down impedance. Selects the ODT pull-down impedance.. Please Refer to Technology specific PHY DATABOOK for supported options ODTStren120N[2:0], ODTStren120N[1:0],ODTStren480N[0:0] 000_00_0 = High Impedance 000_00_1 = 480 ohms 000_01_0 = 240 ohms 000_01_1 = 160 ohms 001_00_0 = 120 ohms 001_00_1 = 96.0 ohms 001_01_0 = 80.0 ohms 001_01_1 = 68.6 ohms 011_00_0 = 60.0 ohms 011_00_1 = 53.3 ohms 011_01_0 = 48.0 ohms 011_01_1 = 43.6 ohms 111_00_0 = 40.0 ohms 111_00_1 = 36.9 ohms 111_01_0 = 34.3 ohms 111_01_1 = 32.0 ohms 111_11_0 = 30.0 ohms 111_11_1 = 28.2 ohms
5-0	Selects the ODT pull-up impedance.

Field	Function
ODTStrenP	Selects the ODT pull-up impedance. Please Refer to Technology specific PHY DATABOOK for supported options ODTStren120P[2:0], ODTStren240P[1:0],ODTStren480P[0:0] 000_00_0 = High Impedance 000_00_1 = 480 ohms 000_01_0 = 240 ohms 000_01_1 = 160 ohms 001_00_0 = 120 ohms 001_00_1 = 96.0 ohms 001_01_0 = 80.0 ohms 001_01_1 = 68.6 ohms 011_00_0 = 60.0 ohms 011_00_1 = 53.3 ohms 011_01_0 = 48.0 ohms 011_01_1 = 43.6 ohms 111_00_0 = 40.0 ohms 111_00_1 = 36.9 ohms 111_01_0 = 34.3 ohms 111_01_1 = 32.0 ohms 111_11_0 = 30.0 ohms 111_11_1 = 28.2 ohms

9.4.3.3.15 Status of RX FIFO Consistency Checks (RxFifoCheckStatus)

9.4.3.3.15.1 Offset

Register	Offset
RxFifoCheckStatus	ACh

9.4.3.3.15.2 Diagram

9.4.3.3.15.3 Fields

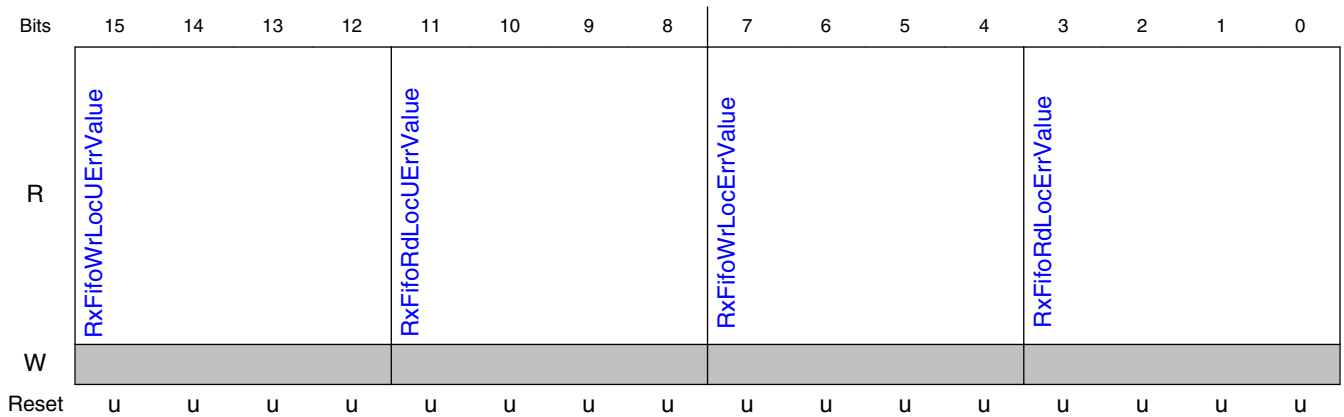
Field	Function
15-4 —	Reserved
3 RxFifoWrLocUErr	If set, the write pointer (DQS side) on the read FIFO associated with data bits [7:4] has a non-zero value at least once.
2 RxFifoRdLocUErr	If set, the read pointer (DFI side) on the read FIFO associated with data bits [7:4] has a non-zero value at least once.
1 RxFifoWrLocErr	If set, the write pointer (DQS side) on the read FIFO associated with data bits [3:0] has a non-zero value at least once.
0 RxFifoRdLocErr	If set, the read pointer (DFI side) on the read FIFO associated with data bits [3:0] had a non-zero value at least once.

9.4.3.3.16 Contains the captured values associated with an RxFifo consistency error (RxFifoCheckErrValues)

9.4.3.3.16.1 Offset

Register	Offset
RxFifoCheckErrValues	AEh

9.4.3.3.16.2 Diagram



9.4.3.3.16.3 Fields

Field	Function
15-12 RxFifoWrLocUErrValue	The first error value captured for the write pointer (DQS side) on the read FIFO associated with data bits [7:4];
11-8 RxFifoRdLocUErrValue	The first error value captured for the read pointer (DFI side) on the read FIFO associated with data bits [7:4];
7-4 RxFifoWrLocErrValue	The first error value captured for the write pointer (DQS side) on the read FIFO associated with data bits [3:0];
3-0 RxFifoRdLocErrValue	The first error value captured for the read pointer (DFI side) on the read FIFO associated with data bits [3:0];

9.4.3.3.17 Data Receive FIFO Pointer Values (RxFifoInfo)

9.4.3.3.17.1 Offset

Register	Offset
RxFifoInfo	B0h

9.4.3.3.17.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RxFifoWrLocU				RxFifoRdLocU				RxFifoWrLoc				RxFifoRdLoc			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.3.17.3 Fields

Field	Function
15-12 RxFifoWrLocU	The Mission mode write pointer of the upper-nibble Rx fifo.
11-8 RxFifoRdLocU	The Mission mode read pointer of the upper-nibble Rx fifo.
7-4 RxFifoWrLoc	The Mission mode write pointer of the lower-nibble Rx fifo.
3-0 RxFifoRdLoc	The Mission mode read pointer of the lower-nibble Rx fifo.

9.4.3.3.18 RX FIFO visibility (RxFifoVisibility)

9.4.3.3.18.1 Offset

Register	Offset
RxFifoVisibility	B2h

9.4.3.3.18.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											RxFifoRdEn	RxFifoRdPtrOvr	RxFifoRdPtr		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.18.3 Fields

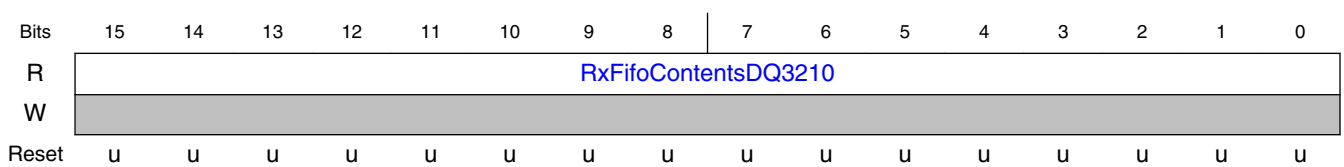
Field	Function
15-5 —	Reserved
4 RxFifoRdEn	Pulse set 0-->1-->0 this bit to capture the Fifo Contents.
3 RxFifoRdPtrOvr	0 : Normal operation - mission mode read pointer is enabled 1 : Override - Control of the rx fifo read pointer is ceded to CSR RxFifoRdPtr.
2-0 RxFifoRdPtr	If CSR RxFifoRdPtrOVr is set, then this CSR selects the rxfifo entry is visible in CSR This 3b field addresses 4b units of the 8x4b (32entry) fifo; that is, rdfifo_nibble_address[2:0]=csrRxFifoRdPtr[2:0] For example, Register RxFifoRdPtr[2:0]=2 will enable reading bit-entries 11. If CSR RxFifoRdPtrOVr is set, then this CSR selects the rxfifo entry is visible in CSR This 3b field addresses 4b units of the 8x4b (32entry) fifo; that is, rdfifo_nibble_address[2:0]=csrRxFifoRdPtr[2:0] For example, Register RxFifoRdPtr[2:0]=2 will enable reading bit-entries 11..8. The exact location of read data will depend on the prior history of reads and PHY initialization.

9.4.3.3.19 RX FIFO contents, lane[3:0] (RxFifoContentsDQ3210)

9.4.3.3.19.1 Offset

Register	Offset
RxFifoContentsDQ3210	B4h

9.4.3.3.19.2 Diagram



9.4.3.3.19.3 Fields

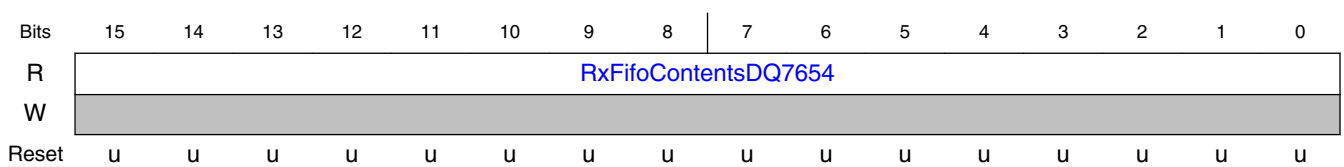
Field	Function
15-0 RxFifoContents DQ3210	A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from lane0. A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from lane0..3 from the four fifo entries addressed by rdfifo_nibble_address[2:0]=RxFifoRdPtr[2:0] Register [15:12] = lane3_ui3,lane3_ui2,lane3_ui1,lane3_ui0 where ui0 is the first occurring bit [11: 8] = lane2_ui3,lane2_ui2,lane2_ui1,lane2_ui0 [7: 4] = lane1_ui3,lane1_ui2,lane1_ui1,lane1_ui0 [3: 0] = lane0_ui3,lane0_ui2,lane0_ui1,lane0_ui0 Note that the DBYTE lane of a given index is not the same as a memory DQ of the same index unless the csr Dq<7..0>LnSel[2:0] have their default/reset value.

9.4.3.3.20 RX FIFO contents, lane[7:4] (RxFifoContentsDQ7654)

9.4.3.3.20.1 Offset

Register	Offset
RxFifoContentsDQ7654	B6h

9.4.3.3.20.2 Diagram



9.4.3.3.20.3 Fields

Field	Function
15-0 RxFifoContents DQ7654	A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from lane4. A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from lane4..7 from the four fifo entries addressed by rdfifo_nibble_address[2:0]=RxFifoRdPtr[2:0] Register [15:12] = lane7_ui3,lane7_ui2,lane7_ui1,lane7_ui0

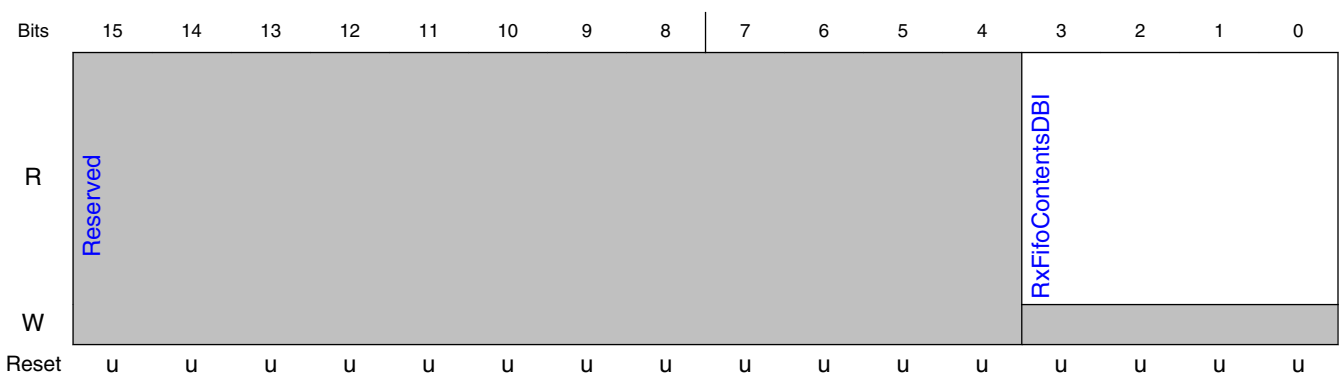
Field	Function
	[11: 8] = lane6_ui3,lane6_ui2,lane6_ui1,lane6_ui0 [7: 4] = lane5_ui3,lane5_ui2,lane5_ui1,lane5_ui0 [3: 0] = lane4_ui3,lane4_ui2,lane4_ui1,lane4_ui0 Note that the DBYTE lane of a given index is not the same as a memory DQ of the same index unless the csr Dq<7..0>LnSel[2:0] have their default/reset value.

9.4.3.3.21 RX FIFO contents, dbi (RxFifoContentsDBI)

9.4.3.3.21.1 Offset

Register	Offset
RxFifoContentsDBI	B8h

9.4.3.3.21.2 Diagram



9.4.3.3.21.3 Fields

Field	Function
15-4 —	Reserved
3-0 RxFifoContentsDBI	A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from DBI from the four fifo entries addressed by rdfifo_nibble_address[2:0]=RxFifoRdPtr[2:0] Register [3: 0] = dbi_ui3,dbi_ui2,dbi_ui1,dbi_ui0 Note that the DBYTE DBI lane is the same as the memory DBI; it is not subject to mapping using csr Dq<7. A window into the contents of the RxFifo, as controlled by CSR RxFifoVisibility This register reads 4b at a time from DBI

Field	Function
	from the four fifo entries addressed by rdfifo_nibble_address[2:0]=RxFifoRdPtr[2:0] Register [3: 0] = dbi_ui3,dbi_ui2,dbi_ui1,dbi_ui0 Note that the DBYTE DBI lane is the same as the memory DBI; it is not subject to mapping using csr Dq<7..0>LnSel[2:0] as the DQ are.

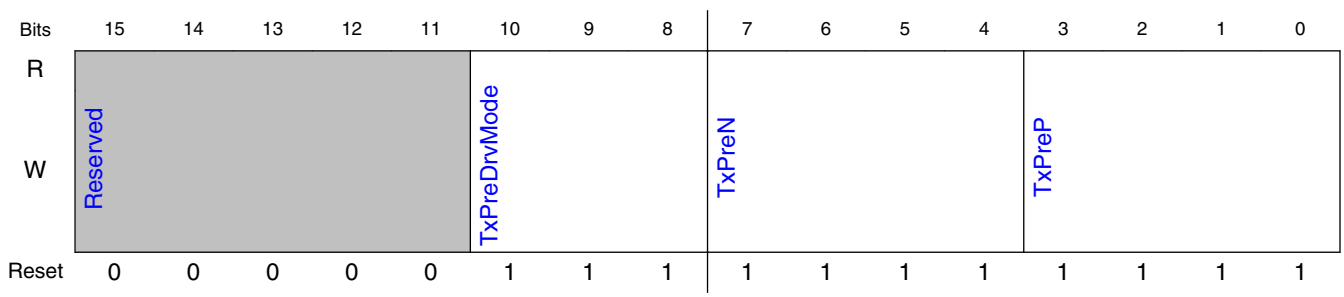
9.4.3.3.22 TX slew rate controls (TxSlewRate_b0_p0 - TxSlewRate_b1_p3)

9.4.3.3.22.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxSlewRate_bn_px	BEh + (x × 20_0000h) + (n × 200h)

9.4.3.3.22.2 Diagram



9.4.3.3.22.3 Fields

Field	Function
15-11 —	Reserved
10-8 TxPreDrvMode	Controls predrivers to adjust timing of turn-on and turn-off of pull-up and pull-down segments.
7-4 TxPreN	4 bit binary trim for the driver pull down slew rate. 4 bit binary trim for the driver pull down slew rate. 4'b0000 has a slower slew rate than 4'b1111
3-0 TxPreP	4 bit binary trim for the driver pull up slew rate. 4 bit binary trim for the driver pull up slew rate. 4'b0000 has a slower slew rate than 4'b1111

9.4.3.3.23 Read DQ per-bit BDL delay (Timing Group 0). (RxPBDlyTg0_r0 - RxPBDlyTg0_r8)

9.4.3.3.23.1 Offset

For x = 0 to 8:

Register	Offset
RxPBDlyTg0_rx	D0h + (x × 200h)

9.4.3.3.23.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RxPBDlyTg0_rx							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.23.3 Fields

Field	Function
15-7 —	Reserved
6-0 RxPBDlyTg0_rx	Read DQ per-bit BDL delay (Timing Group 0). Read DQ per-bit BDL delay (Timing Group 0). Trained to deskew the read DQ data that will be sampled by the incoming per-nibble read DQS, to improve the composite eye. The unit of change is a LSB in the control of the delay circuit, and is not denoninated in units of UI. Register Block Offset Address 0x068 controls the lane0 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x168 controls the lane1 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x268 controls the lane2 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x368 controls the lane3 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x468 controls the lane4 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x568 controls the lane5 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x668 controls the lane6 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x768 controls the lane7 read deskew timing. (For Timing Group 0) Register Block Offset Address 0x868 controls the laneDBI read deskew timing. (For Timing Group 0) Note that this register is not per-pstate. The same deskew (for channel-routing differences) is used for all pstates.

Field	Function
	. Note: Timing Group 0 = RANK0 for all systems except 3DS/LRDIMM.

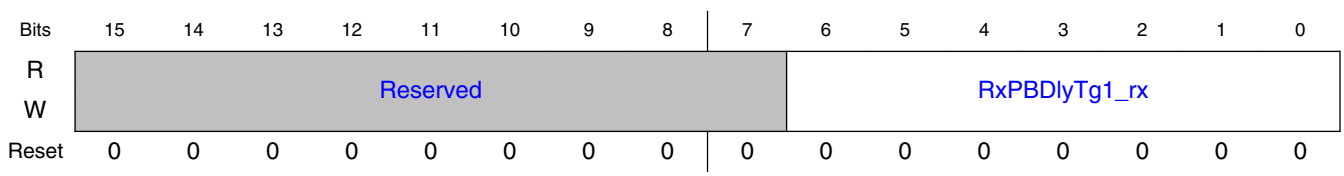
9.4.3.3.24 Read DQ per-bit BDL delay (Timing Group 1). (RxPBDlyTg1_r0 - RxPBDlyTg1_r8)

9.4.3.3.24.1 Offset

For x = 0 to 8:

Register	Offset
RxPBDlyTg1_rx	D2h + (x × 200h)

9.4.3.3.24.2 Diagram



9.4.3.3.24.3 Fields

Field	Function
15-7 —	Reserved
6-0 RxPBDlyTg1_rx	Read DQ per-bit BDL delay (Timing Group 1). Read DQ per-bit BDL delay (Timing Group 1). Trained to deskew the read DQ data that will be sampled by the incoming per-nibble read DQS, to improve the composite eye. The unit of change is a LSB in the control of the delay circuit, and is not denoninated in units of UI. Register Block Offset Address 0x069 controls the lane0 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x169 controls the lane1 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x269 controls the lane2 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x369 controls the lane3 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x469 controls the lane4 read deskew timing. (For Timing Group 1)

Field	Function
	Register Block Offset Address 0x569 controls the lane5 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x669 controls the lane6 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x769 controls the lane7 read deskew timing. (For Timing Group 1) Register Block Offset Address 0x869 controls the laneDBI read deskew timing. (For Timing Group 1) Note that this register is not per-pstate. The same deskew (for channel-routing differences) is used for all pstates. . Note: Timing Group 1 = RANK1 for all systems except 3DS/LRDIMM.

9.4.3.3.25 Read DQ per-bit BDL delay (Timing Group 2). (RxPBDlyTg2_r0 - RxPBDlyTg2_r8)

9.4.3.3.25.1 Offset

For x = 0 to 8:

Register	Offset
RxPBDlyTg2_rx	D4h + (x × 200h)

9.4.3.3.25.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								RxPBDlyTg2_rx							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.25.3 Fields

Field	Function
15-7 —	Reserved
6-0 RxPBDlyTg2_rx	Read DQ per-bit BDL delay (Timing Group 2). Read DQ per-bit BDL delay (Timing Group 2). Trained to deskew the read DQ data that will be sampled by the incoming per-nibble read DQS, to improve the composite eye. The unit of change is a LSB in the control of the delay circuit, and is not denoninated in units of UI.

Field	Function
	Register Block Offset Address 0x06a controls the lane0 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x16a controls the lane1 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x26a controls the lane2 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x36a controls the lane3 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x46a controls the lane4 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x56a controls the lane5 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x66a controls the lane6 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x76a controls the lane7 read deskew timing. (For Timing Group 2) Register Block Offset Address 0x86a controls the laneDBI read deskew timing. (For Timing Group 2) Note that this register is not per-pstate. The same deskew (for channel-routing differences) is used for all pstates. . Note: Timing Group 2 = RANK2 for all systems except 3DS/LRDIMM.

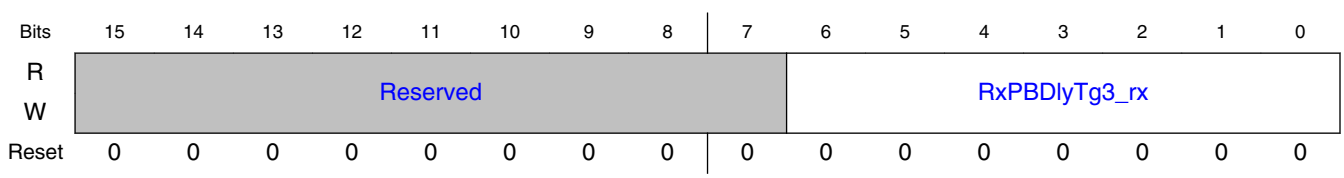
9.4.3.3.26 Read DQ per-bit BDL delay (Timing Group 3). (RxPBDlyTg3_r0 - RxPBDlyTg3_r8)

9.4.3.3.26.1 Offset

For x = 0 to 8:

Register	Offset
RxPBDlyTg3_rx	D6h + (x × 200h)

9.4.3.3.26.2 Diagram



9.4.3.3.26.3 Fields

Field	Function
15-7 —	Reserved

Table continues on the next page...

Field	Function
6-0 RxPBDlyTg3_rx	Read DQ per-bit BDL delay (Timing Group 3). Read DQ per-bit BDL delay (Timing Group 3). Trained to deskew the read DQ data that will be sampled by the incoming per-nibble read DQS, to improve the composite eye. The unit of change is a LSB in the control of the delay circuit, and is not denoninated in units of UI. Register Block Offset Address 0x06b controls the lane0 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x16b controls the lane1 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x26b controls the lane2 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x36b controls the lane3 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x46b controls the lane4 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x56b controls the lane5 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x66b controls the lane6 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x76b controls the lane7 read deskew timing. (For Timing Group 3) Register Block Offset Address 0x86b controls the laneDBI read deskew timing. (For Timing Group 3) Note that this register is not per-pstate. The same deskew (for channel-routing differences) is used for all pstates. . Note: Timing Group 3 = RANK3 for all systems except 3DS/LRDIMM.

9.4.3.3.27 Trained Receive Enable Delay (For Timing Group 0) (RxEnDlyTg0_u0_p0 - RxEnDlyTg0_u1_p3)

9.4.3.3.27.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxEnDlyTg0_un_px	100h + (x × 20_0000h) + (n × 200h)

9.4.3.3.27.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved					RxEnDlyTg0_un_px										
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.3.27.3 Fields

Field	Function
15-11 —	Reserved
10-0 RxEnDlyTg0_un_px	Trained Receive Enable Delay (For Timing Group 0) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. Trained Receive Enable Delay (For Timing Group 0) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst. RxEnDlyTg0[10:6] is the coarse delay, ie one unit of delay is 1 UI. RxEnDlyTg0[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxEnDlyTg0[5] is reserved. Register Block Offset Address 0x080 controls the lower DQ nibble DqsEn. (For Timing Group 0) Register Block Offset Address 0x180 controls the upper DQ nibble DqsEn. (For Timing Group 0) These Registers are replicated per pstate. . Note: Timing Group 0 = RANK0 for all systems except 3DS/LRDIMM.

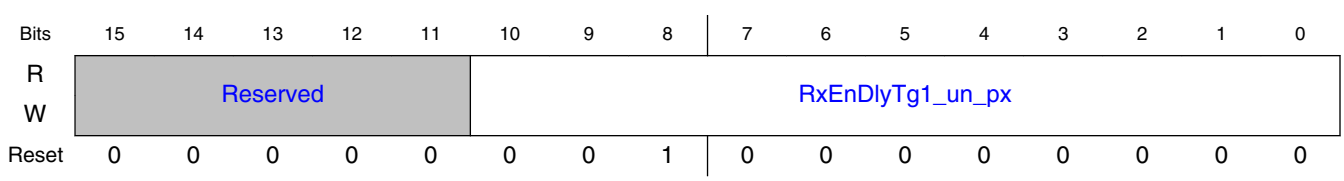
9.4.3.3.28 Trained Receive Enable Delay (For Timing Group 1) (RxEnDlyTg1_u0_p0 - RxEnDlyTg1_u1_p3)

9.4.3.3.28.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxEnDlyTg1_un_px	102h + (x × 20_0000h) + (n × 200h)

9.4.3.3.28.2 Diagram



9.4.3.3.28.3 Fields

Field	Function
15-11 —	Reserved
10-0 RxEnDlyTg1_un_px	Trained Receive Enable Delay (For Timing Group 1) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. Trained Receive Enable Delay (For Timing Group 1) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst. RxEnDlyTg1[10:6] is the coarse delay, ie one unit of delay is 1 UI. RxEnDlyTg1[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxEnDlyTg1[5] is reserved. Register Block Offset Address 0x081 controls the lower DQ nibble DqsEn. (For Timing Group 1) Register Block Offset Address 0x181 controls the upper DQ nibble DqsEn. (For Timing Group 1) These Registers are replicated per pstate. . Note: Timing Group 1 = RANK1 for all systems except 3DS/LRDIMM.

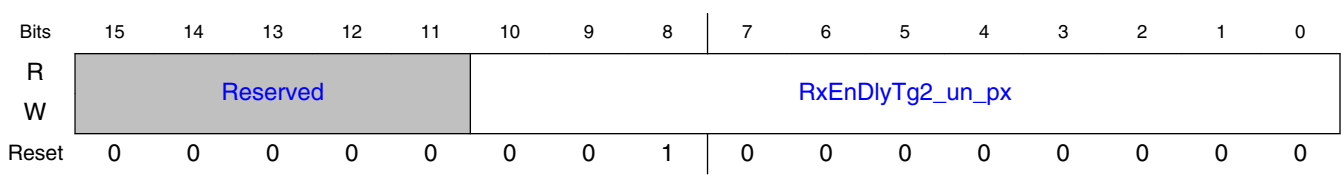
9.4.3.3.29 Trained Receive Enable Delay (For Timing Group 2) (RxEnDlyTg2_u0_p0 - RxEnDlyTg2_u1_p3)

9.4.3.3.29.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxEnDlyTg2_un_px	104h + (x × 20_0000h) + (n × 200h)

9.4.3.3.29.2 Diagram



9.4.3.3.29.3 Fields

Field	Function
15-11 —	Reserved
10-0 RxEnDlyTg2_un_px	Trained Receive Enable Delay (For Timing Group 2) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. Trained Receive Enable Delay (For Timing Group 2) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst. RxEnDlyTg2[10:6] is the coarse delay, ie one unit of delay is 1 UI. RxEnDlyTg2[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxEnDlyTg2[5] is reserved. Register Block Offset Address 0x082 controls the lower DQ nibble DqsEn. (For Timing Group 2) Register Block Offset Address 0x182 controls the upper DQ nibble DqsEn. (For Timing Group 2) These Registers are replicated per pstate. . Note: Timing Group 2 = RANK2 for all systems except 3DS/LRDIMM.

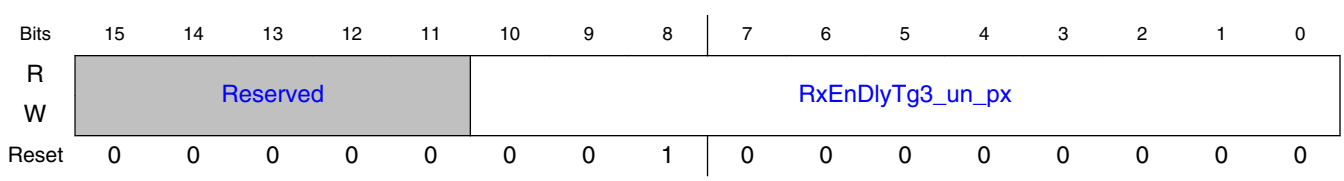
9.4.3.3.30 Trained Receive Enable Delay (For Timing Group 3) (RxEnDlyTg3_u0_p0 - RxEnDlyTg3_u1_p3)

9.4.3.3.30.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxEnDlyTg3_un_px	106h + (x × 20_0000h) + (n × 200h)

9.4.3.3.30.2 Diagram



9.4.3.3.30.3 Fields

Field	Function
15-11 —	Reserved
10-0 RxEnDlyTg3_un_px	Trained Receive Enable Delay (For Timing Group 3) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. Trained Receive Enable Delay (For Timing Group 3) Trained to set the delay from the memory-read command to the signal enabling the read DQS to generate read-data strobes. The trained read DQS enable asserts during the read DQS preamble and deasserts at the end of the read burst. RxEnDlyTg3[10:6] is the coarse delay, ie one unit of delay is 1 UI. RxEnDlyTg3[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxEnDlyTg3[5] is reserved. Register Block Offset Address 0x083 controls the lower DQ nibble DqsEn. (For Timing Group 3) Register Block Offset Address 0x183 controls the upper DQ nibble DqsEn. (For Timing Group 3) These Registers are replicated per pstate. . Note: Timing Group 3 = RANK3 for all systems except 3DS/LRDIMM.

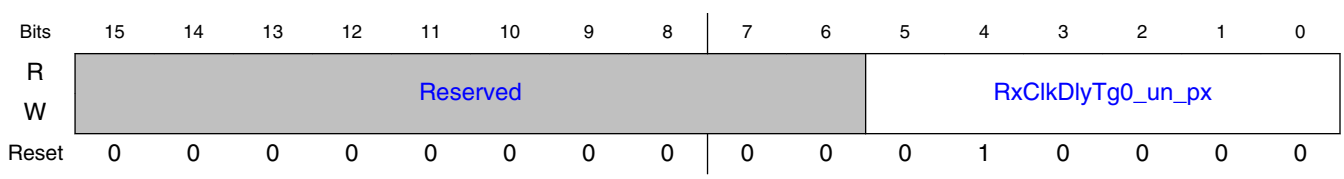
9.4.3.3.31 Trained Read DQS to RxClk Delay (Timing Group DEST=0). (RxClkDlyTg0_u0_p0 - RxClkDlyTg0_u1_p3)

9.4.3.3.31.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxClkDlyTg0_un_px	118h + (x × 20_0000h) + (n × 200h)

9.4.3.3.31.2 Diagram



9.4.3.3.31.3 Fields

Field	Function
15-6 —	Reserved
5-0 RxClkDlyTg0_un_px	Trained Read DQS to RxClk Delay (Timing Group DEST=0). Trained Read DQS to RxClk Delay (Timing Group DEST=0). Trained to generate read DQ receive clock from the received read DQS. RxClk Delay (6 bits, fine delay only); (For Timing Group 0) RxClkDlyTg0[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxClkDlyTg0[5] is reserved. There is no coarse delay field, ie the max configurable delay is 31/32 UI. Register Block Offset Address 0x08c controls the lower DQ nibble RxClk timing. (For Timing Group 0) Register Block Offset Address 0x18c controls the upper DQ nibble RxClk timing. (For Timing Group 0) These Registers are replicated per pstate. . Note: Timing Group 0 = RANK0 for all systems except 3DS/LRDIMM.

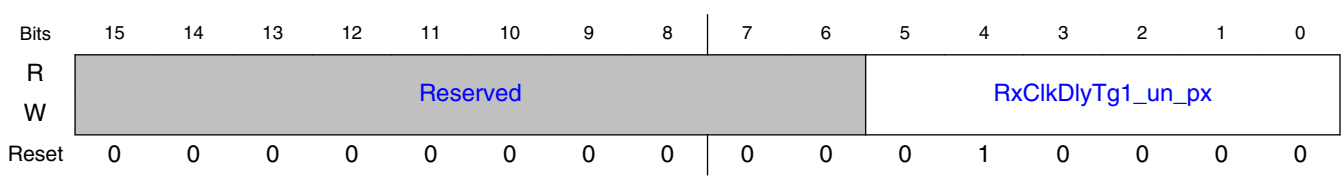
9.4.3.3.32 Trained Read DQS to RxClk Delay (Timing Group DEST=1). (RxClkDlyTg1_u0_p0 - RxClkDlyTg1_u1_p3)

9.4.3.3.32.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxClkDlyTg1_un_px	11Ah + (x × 20_0000h) + (n × 200h)

9.4.3.3.32.2 Diagram



9.4.3.3.32.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

Field	Function
—	
5-0 RxClkDlyTg1_un_px	Trained Read DQS to RxClk Delay (Timing Group DEST=1). Trained Read DQS to RxClk Delay (Timing Group DEST=1). Trained to generate read DQ receive clock from the received read DQS. RxClk Delay (6 bits, fine delay only); (For Timing Group 1) RxClkDlyTg1[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxClkDlyTg1[5] is reserved. There is no coarse delay field, ie the max configurable delay is 31/32 UI. Register Block Offset Address 0x08d controls the lower DQ nibble RxClk timing. (For Timing Group 1) Register Block Offset Address 0x18d controls the upper DQ nibble RxClk timing. (For Timing Group 1) These Registers are replicated per pstate. . Note: Timing Group 1 = RANK1 for all systems except 3DS/LRDIMM.

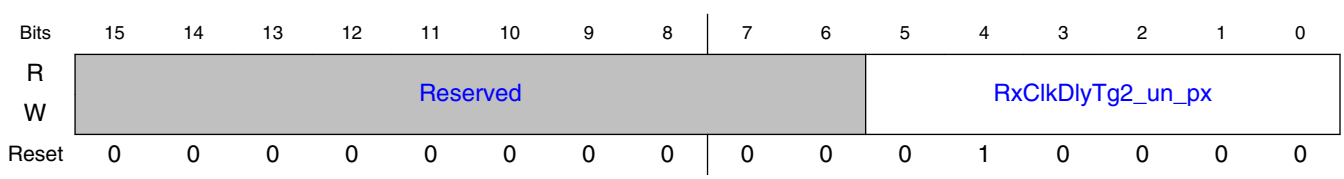
9.4.3.3.33 Trained Read DQS to RxClk Delay (Timing Group DEST=2). (RxClkDlyTg2_u0_p0 - RxClkDlyTg2_u1_p3)

9.4.3.3.33.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxClkDlyTg2_un_px	11Ch + (x × 20_0000h) + (n × 200h)

9.4.3.3.33.2 Diagram



9.4.3.3.33.3 Fields

Field	Function
15-6 —	Reserved
5-0	Trained Read DQS to RxClk Delay (Timing Group DEST=2).

Field	Function
RxClkDlyTg2_un_px	Trained Read DQS to RxClk Delay (Timing Group DEST=2). Trained to generate read DQ receive clock from the received read DQS. RxClk Delay (6 bits, fine delay only); (For Timing Group 2) RxClkDlyTg2[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxClkDlyTg2[5] is reserved. There is no coarse delay field, ie the max configurable delay is 31/32 UI. Register Block Offset Address 0x08e controls the lower DQ nibble RxClk timing. (For Timing Group 2) Register Block Offset Address 0x18e controls the upper DQ nibble RxClk timing. (For Timing Group 2) These Registers are replicated per pstate. . Note: Timing Group 2 = RANK2 for all systems except 3DS/LRDIMM.

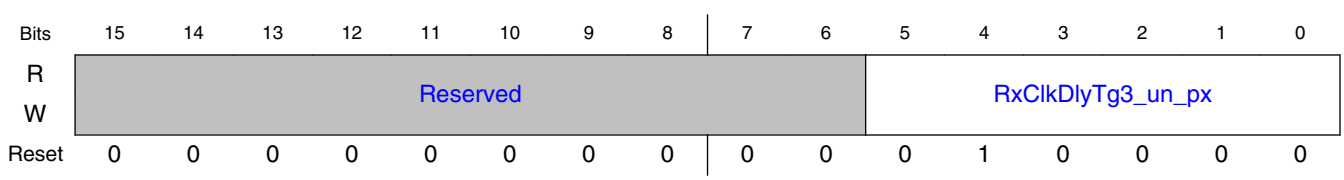
9.4.3.3.34 Trained Read DQS to RxClk Delay (Timing Group DEST=3). (RxClkDlyTg3_u0_p0 - RxClkDlyTg3_u1_p3)

9.4.3.3.34.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
RxClkDlyTg3_un_px	11Eh + (x × 20_0000h) + (n × 200h)

9.4.3.3.34.2 Diagram



9.4.3.3.34.3 Fields

Field	Function
15-6 —	Reserved
5-0 RxClkDlyTg3_un_px	Trained Read DQS to RxClk Delay (Timing Group DEST=3). Trained Read DQS to RxClk Delay (Timing Group DEST=3). Trained to generate read DQ receive clock from the received read DQS.

Field	Function
	RxClk Delay (6 bits, fine delay only); (For Timing Group 3) RxClkDlyTg3[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. RxClkDlyTg3[5] is reserved. There is no coarse delay field, ie the max configurable delay is 31/32 UI. Register Block Offset Address 0x08f controls the lower DQ nibble RxClk timing. (For Timing Group 3) Register Block Offset Address 0x18f controls the upper DQ nibble RxClk timing. (For Timing Group 3) These Registers are replicated per pstate. . Note: Timing Group 3 = RANK3 for all systems except 3DS/LRDIMM.

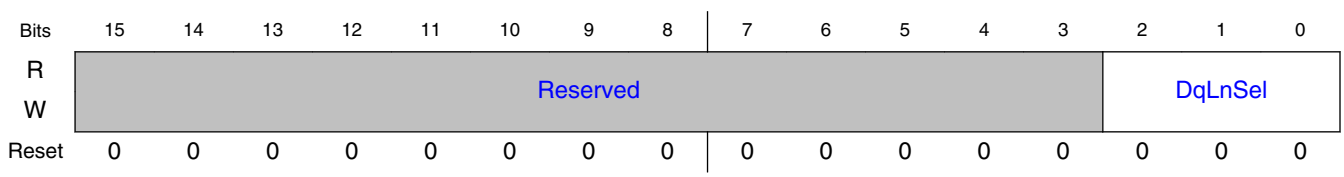
9.4.3.3.35 Maps Phy DQ lane to memory DQ0 (Dq0LnSel - Dq7LnSel)

9.4.3.3.35.1 Offset

For n = 0 to 7:

Register	Offset
DqnLnSel	140h + (n × 2h)

9.4.3.3.35.2 Diagram



9.4.3.3.35.3 Fields

Field	Function
15-3 —	Reserved
2-0 DqLnSel	Supports mapping of PHY dq to dram dq within a byte (swizzle). Supports mapping of PHY dq to dram dq within a byte (swizzle). Intended to undo the swizzle of board-level phy-to-dram connections. such that MRR operations of binary counters may be return correct values. Note that this register is per-dbyte such that the sizzle may be different per dbyte. Each register in a byte's set of DqLnSel must have a unique value within the set.

Field	Function
	For example, if, on this dbyte, PHY lane 3 is connected to memory dq0 (on this dbyte), then Register Dq0LnSel for this dbyte should be 3.

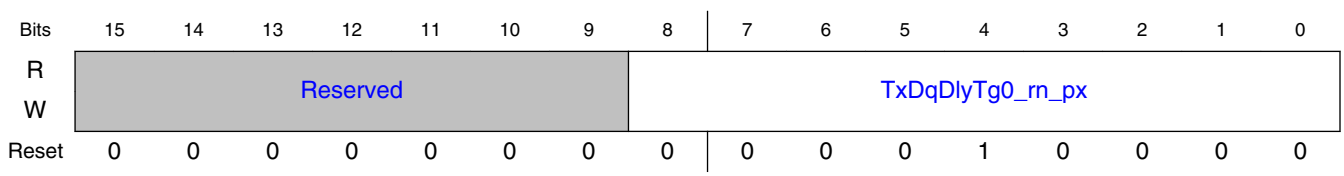
9.4.3.3.36 Write DQ Delay (Timing Group 0). (TxDqDlyTg0_r0_p0 - TxDq DlyTg0_r8_p3)

9.4.3.3.36.1 Offset

For n = 0 to 8; x = 0 to 3:

Register	Offset
TxDqDlyTg0_rn_px	180h + (x × 20_0000h) + (n × 200h)

9.4.3.3.36.2 Diagram



9.4.3.3.36.3 Fields

Field	Function
15-9 —	Reserved
8-0 TxDqDlyTg0_rn_px	Write DQ Delay (Timing Group 0). Write DQ Delay (Timing Group 0). Trained to center the delay between the write DQ to the write DQS. Tx DQ Delay (9 bits, includes 3b coarse and 6b fine delays); (For Timing Group 0) TxDqDlyTg0[8:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqDlyTg0[4:0] is the fine (fractional UI) delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqDlyTg0[5] is reserved. Register Block Offset Address 0x0C0 controls the lane0 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x1C0 controls the lane1 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x2C0 controls the lane2 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x3C0 controls the lane3 write DQ timing. (For Timing Group 0)

Field	Function
	Register Block Offset Address 0x4C0 controls the lane4 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x5C0 controls the lane5 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x6C0 controls the lane6 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x7C0 controls the lane7 write DQ timing. (For Timing Group 0) Register Block Offset Address 0x8C0 controls the laneDMDBI write DQ timing. (For Timing Group 0) These Registers are replicated per pstate. . Note: Timing Group 0 = RANK0 for all systems except 3DS/LRDIMM.

9.4.3.3.37 Write DQ Delay (Timing Group 1). (TxDqDlyTg1_r0_p0 - TxDq DlyTg1_r8_p3)

9.4.3.3.37.1 Offset

For n = 0 to 8; x = 0 to 3:

Register	Offset
TxDqDlyTg1_rn_px	182h + (x × 20_0000h) + (n × 200h)

9.4.3.3.37.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxDqDlyTg1_rn_px							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

9.4.3.3.37.3 Fields

Field	Function
15-9 —	Reserved
8-0 TxDqDlyTg1_rn_px	Write DQ Delay (Timing Group 1). Write DQ Delay (Timing Group 1). Trained to center the delay between the write DQ to the write DQS. Tx DQ Delay (9 bits, includes 3b coarse and 6b fine delays); (For Timing Group 1) TxDqDlyTg1[8:6] is the coarse delay, ie one unit of delay is 1 UI.

Field	Function
	TxDqDlyTg1[4:0] is the fine (fractional UI) delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqDlyTg1[5] is reserved. Register Block Offset Address 0x0C1 controls the lane0 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x1C1 controls the lane1 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x2C1 controls the lane2 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x3C1 controls the lane3 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x4C1 controls the lane4 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x5C1 controls the lane5 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x6C1 controls the lane6 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x7C1 controls the lane7 write DQ timing. (For Timing Group 1) Register Block Offset Address 0x8C1 controls the laneDMDBI write DQ timing. (For Timing Group 1) These Registers are replicated per pstate. . Note: Timing Group 1 = RANK1 for all systems except 3DS/LRDIMM.

9.4.3.3.38 Write DQ Delay (Timing Group 2). (TxDqDlyTg2_r0_p0 - TxDq DlyTg2_r8_p3)

9.4.3.3.38.1 Offset

For n = 0 to 8; x = 0 to 3:

Register	Offset
TxDqDlyTg2_rn_px	184h + (x × 20_0000h) + (n × 200h)

9.4.3.3.38.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxDqDlyTg2_rn_px							
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

9.4.3.3.38.3 Fields

Field	Function
15-9	Reserved

Table continues on the next page...

Field	Function
—	
8-0 TxDqDlyTg2_rn_px	Write DQ Delay (Timing Group 2). Write DQ Delay (Timing Group 2). Trained to center the delay between the write DQ to the write DQS. Tx DQ Delay (9 bits, includes 3b coarse and 6b fine delays); (For Timing Group 2) TxDqDlyTg2[8:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqDlyTg2[4:0] is the fine (fractional UI) delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqDlyTg2[5] is reserved. Register Block Offset Address 0x0C2 controls the lane0 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x1C2 controls the lane1 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x2C2 controls the lane2 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x3C2 controls the lane3 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x4C2 controls the lane4 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x5C2 controls the lane5 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x6C2 controls the lane6 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x7C2 controls the lane7 write DQ timing. (For Timing Group 2) Register Block Offset Address 0x8C2 controls the laneDMDBI write DQ timing. (For Timing Group 2) These Registers are replicated per pstate. . Note: Timing Group 2 = RANK2 for all systems except 3DS/LRDIMM.

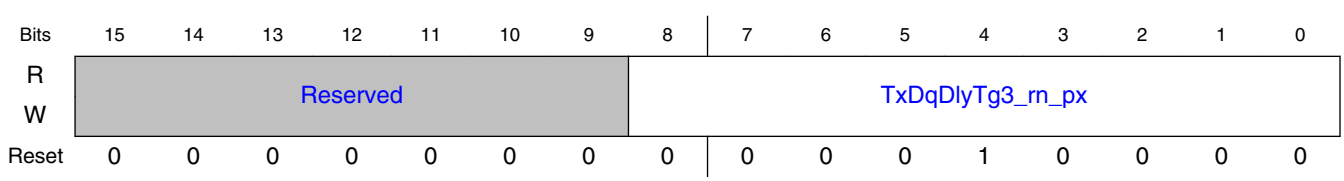
9.4.3.3.39 Write DQ Delay (Timing Group 3). (TxDqDlyTg3_r0_p0 - TxDq DlyTg3_r8_p3)

9.4.3.3.39.1 Offset

For n = 0 to 8; x = 0 to 3:

Register	Offset
TxDqDlyTg3_rn_px	186h + (x × 20_0000h) + (n × 200h)

9.4.3.3.39.2 Diagram



9.4.3.3.39.3 Fields

Field	Function
15-9 —	Reserved
8-0 TxDqDlyTg3_rn_px	Write DQ Delay (Timing Group 3). Write DQ Delay (Timing Group 3). Trained to center the delay between the write DQ to the write DQS. Tx DQ Delay (9 bits, includes 3b coarse and 6b fine delays); (For Timing Group 3) TxDqDlyTg3[8:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqDlyTg3[4:0] is the fine (fractional UI) delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqDlyTg3[5] is reserved. Register Block Offset Address 0x0C3 controls the lane0 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x1C3 controls the lane1 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x2C3 controls the lane2 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x3C3 controls the lane3 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x4C3 controls the lane4 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x5C3 controls the lane5 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x6C3 controls the lane6 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x7C3 controls the lane7 write DQ timing. (For Timing Group 3) Register Block Offset Address 0x8C3 controls the laneDMDBI write DQ timing. (For Timing Group 3) These Registers are replicated per pstate. . Note: Timing Group 3 = RANK3 for all systems except 3DS/LRDIMM.

9.4.3.3.40 Write DQS Delay (Timing Group DEST=0). (TxDqsDlyTg0_u0_p0 - TxDqsDlyTg0_u1_p3)

9.4.3.3.40.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxDqsDlyTg0_un_px	1A0h + (x × 20_0000h) + (n × 200h)

9.4.3.3.40.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxDqsDlyTg0_un_px							
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.3.40.3 Fields

Field	Function
15-10 —	Reserved
9-0 TxDqsDlyTg0_un_px	Write DQS Delay (Timing Group DEST=0). Write DQS Delay (Timing Group DEST=0). Trained to set the delay from the memory-write command to the signal driving the write DQS Tx DQS Delay (10 bits, includes coarse and fine delays); (For Timing Group 0) TxDqsDlyTg0[9:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqsDlyTg0[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqsDlyTg0[5] is reserved. Register is per nibble, per timing-group, per pstate. Register Block Offset Address 0x0D0 controls the lower DQ nibble write DQS timing. (For Timing Group 0) Register Block Offset Address 0x1D0 controls the upper DQ nibble write DQS timing. (For Timing Group 0) . Note: Timing Group 0 = RANK0 for all systems except 3DS/LRDIMM.

9.4.3.3.41 Write DQS Delay (Timing Group DEST=1). (TxDqsDlyTg1_u0_p0 - TxDqsDlyTg1_u1_p3)

9.4.3.3.41.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxDqsDlyTg1_un_px	1A2h + (x × 20_0000h) + (n × 200h)

9.4.3.3.41.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TxDqsDlyTg1_un_px									
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.3.41.3 Fields

Field	Function
15-10 —	Reserved
9-0 TxDqsDlyTg1_un_px	Write DQS Delay (Timing Group DEST=1). Write DQS Delay (Timing Group DEST=1). Trained to set the delay from the memory-write command to the signal driving the write DQS Tx DQS Delay (10 bits, includes coarse and fine delays); (For Timing Group 1) TxDqsDlyTg1[9:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqsDlyTg1[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqsDlyTg1[5] is reserved. Register is per nibble, per timing-group, per pstate. Register Block Offset Address 0x0D1 controls the lower DQ nibble write DQS timing. (For Timing Group 1) Register Block Offset Address 0x1D1 controls the upper DQ nibble write DQS timing. (For Timing Group 1) . Note: Timing Group 1 = RANK1 for all systems except 3DS/LRDIMM.

9.4.3.3.42 Write DQS Delay (Timing Group DEST=2). (TxDqsDlyTg2_u0_p0 - TxDqsDlyTg2_u1_p3)

9.4.3.3.42.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxDqsDlyTg2_un_px	1A4h + (x × 20_0000h) + (n × 200h)

9.4.3.3.42.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxDqsDlyTg2_un_px							
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.3.42.3 Fields

Field	Function
15-10 —	Reserved
9-0 TxDqsDlyTg2_un_px	Write DQS Delay (Timing Group DEST=2). Write DQS Delay (Timing Group DEST=2). Trained to set the delay from the memory-write command to the signal driving the write DQS Tx DQS Delay (10 bits, includes coarse and fine delays); (For Timing Group 2) TxDqsDlyTg2[9:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqsDlyTg2[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqsDlyTg2[5] is reserved. Register is per nibble, per timing-group, per pstate. Register Block Offset Address 0x0D2 controls the lower DQ nibble write DQS timing. (For Timing Group 2) Register Block Offset Address 0x1D2 controls the upper DQ nibble write DQS timing. (For Timing Group 2) . Note: Timing Group 2 = RANK2 for all systems except 3DS/LRDIMM.

9.4.3.3.43 Write DQS Delay (Timing Group DEST=3). (TxDqsDlyTg3_u0_p0 - TxDqsDlyTg3_u1_p3)

9.4.3.3.43.1 Offset

For n = 0 to 1; x = 0 to 3:

Register	Offset
TxDqsDlyTg3_un_px	1A6h + (x × 20_0000h) + (n × 200h)

9.4.3.3.43.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						TxDqsDlyTg3_un_px									
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.3.43.3 Fields

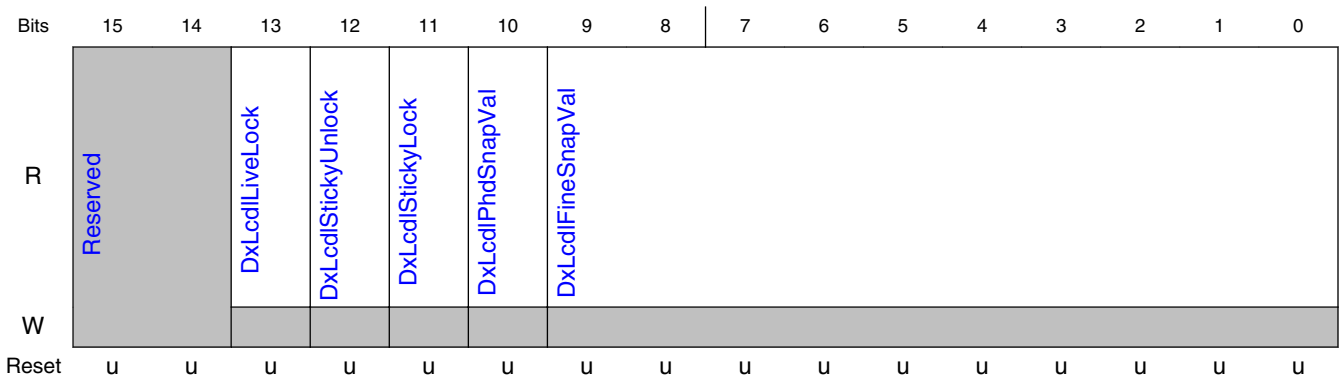
Field	Function
15-10 —	Reserved
9-0 TxDqsDlyTg3_un_px	Write DQS Delay (Timing Group DEST=3). Write DQS Delay (Timing Group DEST=3). Trained to set the delay from the memory-write command to the signal driving the write DQS Tx DQS Delay (10 bits, includes coarse and fine delays); (For Timing Group 3) TxDqsDlyTg3[9:6] is the coarse delay, ie one unit of delay is 1 UI. TxDqsDlyTg3[4:0] is the fine delay, ie one unit of delay is one-thirtysecond of a UI = UI/32. TxDqsDlyTg3[5] is reserved. Register is per nibble, per timing-group, per pstate. Register Block Offset Address 0x0D3 controls the lower DQ nibble write DQS timing. (For Timing Group 3) Register Block Offset Address 0x1D3 controls the upper DQ nibble write DQS timing. (For Timing Group 3) . Note: Timing Group 3 = RANK3 for all systems except 3DS/LRDIMM.

9.4.3.3.44 Debug status of the DBYTE LCDL (DxLcdlStatus)

9.4.3.3.44.1 Offset

Register	Offset
DxLcdlStatus	1C8h

9.4.3.3.44.2 Diagram



9.4.3.3.44.3 Fields

Field	Function
15-14 —	Reserved
13 DxLcdlLiveLock	present value of whether the LCDL is locked, valid when LcdlTstEnable=1.
12 DxLcdlStickyUnlock	latched value of whether the LCDL ever lost lock after the assertion of LcdlTstEnable.
11 DxLcdlStickyLock	latched value of whether the LCDL ever achieved lock after the assertion of LcdlTstEnable.
10 DxLcdlPhdSnapVal	Value of the LCDL phase-detector output, latched by pulse on csr LcdlFineSnap. Value of the LCDL phase-detector output, latched by pulse on csr LcdlFineSnap. while csr LcdlTstEnable=1.
9-0 DxLcdlFineSnapVal	Value of the LCDL 1UI estimate code, latched by pulse on csr LcdlFineSnap while csr LcdlTstEnable=1. Value of the LCDL 1UI estimate code, latched by pulse on csr LcdlFineSnap while csr LcdlTstEnable=1. Index 9 is reserved for growth.

9.4.3.3.45 DFI MaxReadLatency (DFIMRL_p1)

9.4.3.3.45.1 Offset

Register	Offset
DFIMRL_p1	20_0040h

9.4.3.3.45.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											DFIMRL_p1				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

9.4.3.3.45.3 Fields

Field	Function
15-5 —	Reserved
4-0 DFIMRL_p1	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This is the value, in units of two mem clocks, between dfi_rddata_en and dfi_rddata_valid A unit change in the LSB is a change in MRL of two mem clocks.

9.4.3.3.46 DFI MaxReadLatency (DFIMRL_p2)

9.4.3.3.46.1 Offset

Register	Offset
DFIMRL_p2	40_0040h

9.4.3.3.46.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											DFIMRL_p2				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

9.4.3.3.46.3 Fields

Field	Function
15-5 —	Reserved
4-0 DFIMRL_p2	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This is the value, in units of two mem clocks, between dfi_rddata_en and dfi_rddata_valid A unit change in the LSB is a change in MRL of two mem clocks.

9.4.3.3.47 DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg0_p2)

9.4.3.3.47.1 Offset

Register	Offset
PptDqsCntInvTrnTg0_p2	40_015Ch

9.4.3.3.47.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PptDqsCntInvTrnTg0_p2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.47.3 Fields

Field	Function
15-0 PptDqsCntInvTrnTg0_p2	Programmed by PHY training firmware to support LPDDR3/LPDDR4 DRAM drift compensation.

9.4.3.3.48 DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg1_p2)

9.4.3.3.48.1 Offset

Register	Offset
PptDqsCntInvTrnTg1_p2	40_015Eh

9.4.3.3.48.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PptDqsCntInvTrnTg1_p2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.48.3 Fields

Field	Function
15-0 PptDqsCntInvTrnTg1_p2	Programmed by PHY training firmware to support LPDDR3/LPDDR4 DRAM drift compensation.

9.4.3.3.49 DFI MaxReadLatency (DFIMRL_p3)

9.4.3.3.49.1 Offset

Register	Offset
DFIMRL_p3	60_0040h

9.4.3.3.49.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											DFIMRL_p3				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

9.4.3.3.49.3 Fields

Field	Function
15-5 —	Reserved
4-0 DFIMRL_p3	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This is the value, in units of two mem clocks, between dfi_rddata_en and dfi_rddata_valid A unit change in the LSB is a change in MRL of two mem clocks.

9.4.3.3.50 DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg0_p3)

9.4.3.3.50.1 Offset

Register	Offset
PptDqsCntInvTrnTg0_p3	60_015Ch

9.4.3.3.50.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PptDqsCntInvTrnTg0_p3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.50.3 Fields

Field	Function
15-0 PptDqsCntInvTrnTg0_p3	Programmed by PHY training firmware to support LPDDR3/LPDDR4 DRAM drift compensation.

9.4.3.3.51 DQS Oscillator Count inverse at time of training in LPDDR4 drift compensation (PptDqsCntInvTrnTg1_p3)

9.4.3.3.51.1 Offset

Register	Offset
PptDqsCntInvTrnTg1_p3	60_015Eh

9.4.3.3.51.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PptDqsCntInvTrnTg1_p3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.3.51.3 Fields

Field	Function
15-0 PptDqsCntInvTrnTg1_p3	Programmed by PHY training firmware to support LPDDR3/LPDDR4 DRAM drift compensation.

9.4.3.4 DWC_DDRPHYA_DRTUB register descriptions

9.4.3.4.1 DWC_DDRPHYA_DRTUB Memory map

DDR4/3 PHY address block

DWC_DDRPHYA_DRTUB0 base address: C_0000h

Offset	Register	Width (In bits)	Access	Reset value
100h	Ucclk and Hclk enables (UcclkHclkEnables)	16	RW	0003h
102h	PIE current Pstate value (CurPstate0b)	16	RW	0000h
1DAh	Customer settable by the customer (CUSTPUBREV)	16	RO	Table 9-4
1DCh	The hardware version of this PUB, excluding the PHY (PUBREV)	16	RO	1160h

9.4.3.4.2 Ucclk and Hclk enables (UcclkHclkEnables)

9.4.3.4.2.1 Offset

Register	Offset
UcclkHclkEnables	100h

9.4.3.4.2.2 Diagram

9.4.3.4.2.3 Fields

Field	Function
15-2 —	Reserved
1 HclkEn	When training has completed (and assuming no further need for the training hardware), the enable should be set to 0 to reduce power.
0 UcclkEn	When training has completed (and assuming no further need for the microcontroller), the enable should be set to 0 to reduce power.

9.4.3.4.3 PIE current Pstate value (CurPstate0b)

9.4.3.4.3.1 Offset

Register	Offset
CurPstate0b	102h

9.4.3.4.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												CurPstate0b			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.4.3.3 Fields

Field	Function
15-4 —	Reserved
3-0 CurPstate0b	PIE current Pstate value This register is used to select values for writing by the Pstate sequencer and is written in the beginning of the Pstate switch. PIE current Pstate value This register is used to select values for writing by the Pstate sequencer and is written in the beginning of the Pstate switch. It can also be used to reference which Pstate the sequencer thinks it's in.

9.4.3.4.4 Customer settable by the customer (CUSTPUBREV)

9.4.3.4.4.1 Offset

Register	Offset
CUSTPUBREV	1DAh

9.4.3.4.4.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										CUSTPUBREV					
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.4.4.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

Field	Function
—	
5-0 CUSTPUBREV	The customer settable PUB version number. The customer settable PUB version number. The value is derived from the `define DWC_DDRPHY_CUST_PUBREV

9.4.3.4.5 The hardware version of this PUB, excluding the PHY (PUBREV)

9.4.3.4.5.1 Offset

Register	Offset
PUBREV	1DCh

9.4.3.4.5.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PUBMJR								PUBMDR				PUBMNR			
W																
Reset	0	0	0	1	0	0	0	1	0	1	1	0	0	0	0	0

9.4.3.4.5.3 Fields

Field	Function
15-8 PUBMJR	Indicates major revision of the PUB.
7-4 PUBMDR	Indicates moderate revision of the PUB.
3-0 PUBMNR	Indicates minor update of the PUB.

9.4.3.5 DWC_DDRPHYA_INITENG register descriptions

9.4.3.5.1 DWC_DDRPHYA_INITENG Memory map

DesignWare Cores DDR4/3 PHY address block

DWC_DDRPHYA_INITENG0 base address: 9_0000h

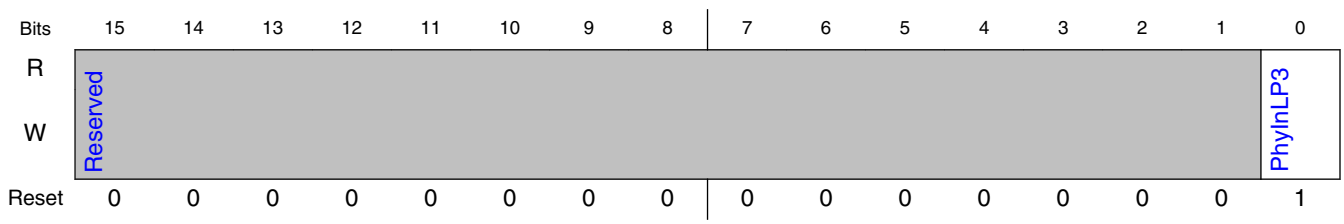
Offset	Register	Width (In bits)	Access	Reset value
50h	Indicator for PIE Lower Power 3 (LP3) Status (PhyInLP3)	16	RW	0001h

9.4.3.5.2 Indicator for PIE Lower Power 3 (LP3) Status (PhyInLP3)

9.4.3.5.2.1 Offset

Register	Offset
PhyInLP3	50h

9.4.3.5.2.2 Diagram



9.4.3.5.2.3 Fields

Field	Function
15-1 —	Reserved
0 PhyInLP3	Read Only. Read Only. Set to 1 by the PIE once completed LP3 Entry sequence; Cleared during LP3 Exit sequence. System software can read this csr for the status of PIE engine.

9.4.3.6 DWC_DDRPHYA_MASTER register descriptions

9.4.3.6.1 DWC_DDRPHYA_MASTER Memory map

DDR4/3 PHY address block

DWC_DDRPHYA_MASTER0 base address: 2_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Rx FIFO pointer initialization control (RxFifoInit)	16	RW	0000h
2h	Clock gating control (ForceClkDisable)	16	RW	0000h
6h	This Register used by Training Firmware to force an internal PHY Update Event. (ForceInternalUpdate)	16	RW	0000h
8h	Read Only displays PHY Configuration. (PhyConfig)	16	RO	Table 9-4
Eh	Test Bump Control1 (TestBumpCntrl1)	16	RW	0B03h
10h	Impedance Calibration Clock Ratio (CalUclkInfo_p0)	16	RW	0320h
14h	Test Bump Control (TestBumpCntrl)	16	RW	0000h
16h	PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p0)	16	RW	0000h
18h	PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p0)	16	RW	0000h
1Ah	PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p0)	16	RW	0000h
1Ch	PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p0)	16	RW	0000h
1Eh	PHY Alert status bit (PhyAlertStatus)	16	RO	Table 9-4
20h	Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p0)	16	RW	0000h
24h	ATestMode control (ATestMode)	16	RW	0000h
28h	TX P Impedance Calibration observation (TxCalBinP)	16	RO	Table 9-4
2Ah	TX N Impedance Calibration observation (TxCalBinN)	16	RO	Table 9-4
2Ch	TX P Impedance Calibration override (TxCalPOvr)	16	RW	0000h
2Eh	TX N Impedance Calibration override (TxCalNOvr)	16	RW	0000h
30h	Enables for update and low-power interfaces for DFI0 and DFI1 (DfiMode)	16	RW	0005h
32h	Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p0)	16	RW	0000h
34h	Digital Observation Pin control (MtestMuxSel)	16	RW	0000h
36h	Digital Observation Pin program info for debug (MtestPgmInfo)	16	RW	0000h
38h	Dynaimc Power Up/Down control (DynPwrDnUp)	16	RW	0000h
3Ch	PHY Technology ID Register (PhyTID)	16	RW	0000h
40h	HWT MaxReadLatency. (HwtMRL_p0)	16	RW	0006h
42h	DFI PhyUpdate Request time counter (in MEMCLKs) (DFIPHYUPD)	16	RW	FF07h
44h	Controls the write DQ generation for Per-Dram-Addressing of MRS (PdaMrsWriteMode)	16	RW	0000h
46h	Controls whether dfi_geardown_en will cause CS and CKE timing to change. (DFIGEARDOWNCTL)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
48h	Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p0)	16	RW	0008h
4Ah	DBYTE module controls to select X4 Dram device mode (MasterX4 Config)	16	RW	0000h
4Ch	Write level feedback DQ observability select. (WrLevBits)	16	RW	0098h
4Eh	In DDR4 Mode , this controls whether CS_N[3:2] should be multicast on CID[1:0] (EnableCsMulticast)	16	RW	0000h
50h	Drives cs_n[0] onto cs_n[1] during training (HwtLpCsMultiCast)	16	RW	0000h
58h	Disable for unused ACX Nibbles (Acx4AnibDis)	16	RW	0000h
5Ah	This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p0)	16	RW	0000h
5Ch	Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p0)	16	RW	0000h
74h	DLL Mode control CSR for DBYTEs (DbyteDllModeCntrl)	16	RW	Table 9-4
8Ah	Impedance Calibration offsets control (CalOffsets)	16	RW	0000h
8Eh	Sar Init Vals (SarInitVals)	16	RW	0127h
92h	Impedance Calibration PExt Override control (CalPExtOvr)	16	RW	0000h
94h	Impedance Calibration Cmpr 50 control (CalCmpr5Ovr)	16	RW	0000h
96h	Impedance Calibration NInt Override control (CalNIntOvr)	16	RW	0000h
A0h	Impedance Calibration driver strength control (CalDrvStr0)	16	RW	0000h
ACh	READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p0)	16	RW	0006h
B6h	This Register is used to configure the MemAlert Receiver (MemAlert Control)	16	RW	4000h
B8h	This Register is used to configure the MemAlert Receiver (MemAlert Control2)	16	RW	0000h
C0h	Protection and control of BP_MemReset_L (MemResetL)	16	RW	0000h
DAh	Drive CS_N 3:0 onto CS_N 7:4 (DriveCSLowOntoHigh)	16	RW	0000h
DCh	PUBMODE - HWT Mux Select (PUBMODE)	16	RW	0000h
DEh	Misc PHY status bits (MiscPhyStatus)	16	RO	Table 9-4
E0h	Controls whether the loopback path bypasses the final PAD node. (CoreLoopbackSel)	16	RW	0000h
E2h	DLL Various Training Parameters (DllTrainParam)	16	RW	0002h
E8h	CSn Disable Bypass for LPDDR3/4 (HwtLpCsEnBypass)	16	RW	0000h
EAh	Dfi Command/Address Mode (DfiCAMode)	16	RW	0000h
F0h	DLL Lock State machine control register (DllControl)	16	RW	0003h
F2h	DLL update phase control (PulseDllUpdatePhase)	16	RW	0000h
F8h	DLL gain control (DllGainCtl_p0)	16	RW	00A1h
110h	Impedance Calibration Control (CalRate)	16	RW	0009h
112h	Impedance Calibration Zap/Reset (CalZap)	16	RW	0000h
116h	PSTATE Selection (PState)	16	RW	000Ch
11Ah	PLL Output Control (PllOutGateControl)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
120h	PMU Power-on Reset Control (PLL/DLL Lock Done) (PorControl)	16	RW	0000h
12Eh	Impedance Calibration Busy Status (CalBusy)	16	RO	Table 9-4
130h	Miscellaneous impedance calibration controls. (CalMisc2)	16	RW	0004h
134h	Controls for disabling the impedance calibration of certain targets. (CalMisc)	16	RW	0000h
138h	Impedance Calibration Cmpr control (CalCmpr5)	16	RO	Table 9-4
13Ah	Impedance Calibration NInt control (CalNInt)	16	RO	Table 9-4
13Ch	Impedance Calibration PExt control (CalPExt)	16	RO	Table 9-4
150h	Impedance Calibration Cmp Invert control (CalCmpInvert)	16	RW	0003h
15Ch	Impedance Calibration Cmpana control (CalCmpanaCntrl)	16	RW	0112h
160h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p0)	16	RW	00E4h
164h	PHY Global Vref Controls (VrefInGlobal_p0)	16	RW	0200h
168h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p0)	16	RW	00E4h
16Ah	Counts successful PHY Master Interface Updates (PPTs) (MasUpdGoodCtr)	16	RO	Table 9-4
16Ch	Counts successful PHY-initiated DFI0 Interface Updates (PhyUpd0GoodCtr)	16	RO	Table 9-4
16Eh	Counts successful PHY-initiated DFI1 Interface Updates (PhyUpd1GoodCtr)	16	RO	Table 9-4
170h	Counts successful Memory Controller DFI0 Interface Updates (CtlUpd0GoodCtr)	16	RO	Table 9-4
172h	Counts successful Memory Controller DFI1 Interface Updates (CtlUpd1GoodCtr)	16	RO	Table 9-4
174h	Counts unsuccessful PHY Master Interface Updates (MasUpdFailCtr)	16	RO	Table 9-4
176h	Counts unsuccessful PHY-initiated DFI0 Interface Updates (PhyUpd0FailCtr)	16	RO	Table 9-4
178h	Counts unsuccessful PHY-initiated DFI1 Interface Updates (PhyUpd1FailCtr)	16	RO	Table 9-4
17Ah	Enables for Performance Counters (PhyPerfCtrEnable)	16	RW	0000h
186h	PLL Power Down (PllPwrDn)	16	RW	0001h
188h	PLL Reset (PllReset)	16	RW	0001h
18Ah	PState dependent PLL Control Register 2 (PllCtrl2_p0)	16	RW	0019h
18Ch	PLL Control Register 0 (PllCtrl0)	16	RW	2008h
18Eh	PState dependent PLL Control Register 1 (PllCtrl1_p0)	16	RW	0020h
190h	PLL Testing Control Register (PllTst)	16	RW	0000h
192h	PLL's pll_lock pin output (PllLockStatus)	16	RO	Table 9-4
194h	Additional controls for PLL CP/VCO modes of operation (PllTestMode_p0)	16	RW	0124h
196h	PLL Control Register 3 (PllCtrl3)	16	RW	01F0h
198h	PState dependent PLL Control Register 4 (PllCtrl4_p0)	16	RW	017Fh
19Ah	PLL's eoc (end of calibration) output (PllEndofCal)	16	RO	Table 9-4

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
19Ch	PLL's standby_eff (effective standby) output (PllStandbyEff)	16	RO	Table 9-4
19Eh	PLL's Dacval_out output (PllDacValOut)	16	RO	Table 9-4
1C6h	Controls for use in observing and testing the LCDLs. (LcdlDbgCntl)	16	RW	0000h
1C8h	Debug status of the DBYTE LCDL (AcLcdlStatus)	16	RO	Table 9-4
1DAh	Customer settable by the customer (CUSTPHYREV)	16	RO	Table 9-4
1DCh	The hardware version of this PHY, excluding the PUB (PHYREV)	16	RO	0100h
1DEh	Start vector value to be used for LP3-exit or Init PIE Sequence (LP3ExitSeq0BStartVector)	16	RW	00DEh
1E0h	DFI Frequency Translation Register 0 (DfiFreqXlat0)	16	RW	0000h
1E2h	DFI Frequency Translation Register 1 (DfiFreqXlat1)	16	RW	0000h
1E4h	DFI Frequency Translation Register 2 (DfiFreqXlat2)	16	RW	0000h
1E6h	DFI Frequency Translation Register 3 (DfiFreqXlat3)	16	RW	0000h
1E8h	DFI Frequency Translation Register 4 (DfiFreqXlat4)	16	RW	0000h
1EAh	DFI Frequency Translation Register 5 (DfiFreqXlat5)	16	RW	0000h
1ECh	DFI Frequency Translation Register 6 (DfiFreqXlat6)	16	RW	0000h
1EEh	DFI Frequency Translation Register 7 (DfiFreqXlat7)	16	RW	0000h
1F0h	TxRdPtrInit control register (TxRdPtrInit)	16	RW	0001h
1F2h	DFI Init Complete control (DfiInitComplete)	16	RW	0000h
1F4h	DFI Frequency Ratio (DfiFreqRatio_p0)	16	RW	0001h
1F6h	Enable more frequent consistency checks of the RX FIFOs (RxFifoChecks)	16	RW	0000h
200h	Maps PHY CAA lane 0 from dfi0_address of the index of the register contents (MapCAA0toDfi)	16	RW	0000h
202h	Maps PHY CAA lane 1 from dfi0_address of the index of the register contents (MapCAA1toDfi)	16	RW	0001h
204h	Maps PHY CAA lane 2 from dfi0_address of the index of the register contents (MapCAA2toDfi)	16	RW	0002h
206h	Maps PHY CAA lane 3 from dfi0_address of the index of the register contents (MapCAA3toDfi)	16	RW	0003h
208h	Maps PHY CAA lane 4 from dfi0_address of the index of the register contents (MapCAA4toDfi)	16	RW	0004h
20Ah	Maps PHY CAA lane 5 from dfi0_address of the index of the register contents (MapCAA5toDfi)	16	RW	0005h
20Ch	Maps PHY CAA lane 6 from dfi0_address of the index of the register contents (MapCAA6toDfi)	16	RW	0006h
20Eh	Maps PHY CAA lane 7 from dfi0_address of the index of the register contents (MapCAA7toDfi)	16	RW	0007h
210h	Maps PHY CAA lane 8 from dfi0_address of the index of the register contents (MapCAA8toDfi)	16	RW	0008h
212h	Maps PHY CAA lane 9 from dfi0_address of the index of the register contents (MapCAA9toDfi)	16	RW	0009h
220h	Maps PHY CAB lane 0 from dfi1_address of the index of the register contents (MapCAB0toDfi)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
222h	Maps PHY CAB lane 1 from dfi1_address of the index of the register contents (MapCAB1toDfi)	16	RW	0001h
224h	Maps PHY CAB lane 2 from dfi1_address of the index of the register contents (MapCAB2toDfi)	16	RW	0002h
226h	Maps PHY CAB lane 3 from dfi1_address of the index of the register contents (MapCAB3toDfi)	16	RW	0003h
228h	Maps PHY CAB lane 4 from dfi1_address of the index of the register contents (MapCAB4toDfi)	16	RW	0004h
22Ah	Maps PHY CAB lane 5 from dfi1_address of the index of the register contents (MapCAB5toDfi)	16	RW	0005h
22Ch	Maps PHY CAB lane 6 from dfi1_address of the index of the register contents (MapCAB6toDfi)	16	RW	0006h
22Eh	Maps PHY CAB lane 7 from dfi1_address of the index of the register contents (MapCAB7toDfi)	16	RW	0007h
230h	Maps PHY CAB lane 8 from dfi1_address of the index of the register contents (MapCAB8toDfi)	16	RW	0008h
232h	Maps PHY CAB lane 9 from dfi1_address of the index of the register contents (MapCAB9toDfi)	16	RW	0009h
236h	Interrupt Enable Bits (PhyInterruptEnable)	16	RW	0000h
238h	Interrupt Firmware Control Bits (PhyInterruptFWControl)	16	RW	0000h
23Ah	Interrupt Mask Bits (PhyInterruptMask)	16	RW	0000h
23Ch	Interrupt Clear Bits (PhyInterruptClear)	16	RW	0000h
23Eh	Interrupt Status Bits (PhyInterruptStatus)	16	RO	Table 9-4
240h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress0)	16	RW	0000h
242h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress1)	16	RW	0000h
244h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress2)	16	RW	0000h
246h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress3)	16	RW	0000h
248h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress4)	16	RW	0000h
24Ah	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress5)	16	RW	0000h
24Ch	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress6)	16	RW	0000h
24Eh	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress7)	16	RW	0000h
250h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress8)	16	RW	0000h
252h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress9)	16	RW	0000h
254h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress10)	16	RW	0000h
256h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress11)	16	RW	0000h
258h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress12)	16	RW	0000h
25Ah	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress13)	16	RW	0000h
25Ch	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress14)	16	RW	0000h
25Eh	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress15)	16	RW	0000h
260h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress17)	16	RW	0000h
262h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtActN)	16	RW	0000h
264h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank0)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
266h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank1)	16	RW	0000h
268h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank2)	16	RW	0000h
26Ah	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBg0)	16	RW	0000h
26Ch	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBg1)	16	RW	0000h
26Eh	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtCasN)	16	RW	0000h
270h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtRasN)	16	RW	0000h
272h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtWeN)	16	RW	0000h
274h	Signal swizzle selection for HWT swizzle (HwtSwizzleHwtParityIn)	16	RW	0000h
20_0010h	Impedance Calibration Clock Ratio (CalUclkInfo_p1)	16	RW	0320h
20_0016h	PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p1)	16	RW	0000h
20_0018h	PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p1)	16	RW	0000h
20_001Ah	PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p1)	16	RW	0000h
20_001Ch	PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p1)	16	RW	0000h
20_0020h	Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p1)	16	RW	0000h
20_0032h	Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p1)	16	RW	0000h
20_0040h	HWT MaxReadLatency. (HwtMRL_p1)	16	RW	0006h
20_0048h	Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p1)	16	RW	0008h
20_005Ah	This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p1)	16	RW	0000h
20_005Ch	Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p1)	16	RW	0000h
20_00ACh	READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p1)	16	RW	0006h
20_00F8h	DLL gain control (DllGainCtl_p1)	16	RW	00A1h
20_0160h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p1)	16	RW	00E4h
20_0164h	PHY Global Vref Controls (VrefInGlobal_p1)	16	RW	0200h
20_0168h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p1)	16	RW	00E4h
20_018Ah	PState dependent PLL Control Register 2 (PllCtrl2_p1)	16	RW	0019h
20_018Eh	PState dependent PLL Control Register 1 (PllCtrl1_p1)	16	RW	0020h
20_0194h	Additional controls for PLL CP/VCO modes of operation (PllTestMode_p1)	16	RW	0124h
20_0198h	PState dependent PLL Control Register 4 (PllCtrl4_p1)	16	RW	017Fh
20_01F4h	DFI Frequency Ratio (DfiFreqRatio_p1)	16	RW	0001h
40_0010h	Impedance Calibration Clock Ratio (CalUclkInfo_p2)	16	RW	0320h
40_0016h	PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p2)	16	RW	0000h
40_0018h	PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p2)	16	RW	0000h
40_001Ah	PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p2)	16	RW	0000h
40_001Ch	PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p2)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
40_0020h	Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p2)	16	RW	0000h
40_0032h	Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p2)	16	RW	0000h
40_0040h	HWT MaxReadLatency. (HwtMRL_p2)	16	RW	0006h
40_0048h	Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p2)	16	RW	0008h
40_005Ah	This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p2)	16	RW	0000h
40_005Ch	Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p2)	16	RW	0000h
40_00ACh	READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p2)	16	RW	0006h
40_00F8h	DLL gain control (DllGainCtl_p2)	16	RW	00A1h
40_0160h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p2)	16	RW	00E4h
40_0164h	PHY Global Vref Controls (VrefInGlobal_p2)	16	RW	0200h
40_0168h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p2)	16	RW	00E4h
40_018Ah	PState dependent PLL Control Register 2 (PllCtrl2_p2)	16	RW	0019h
40_018Eh	PState dependent PLL Control Register 1 (PllCtrl1_p2)	16	RW	0020h
40_0194h	Additional controls for PLL CP/VCO modes of operation (PllTestMode_p2)	16	RW	0124h
40_0198h	PState dependent PLL Control Register 4 (PllCtrl4_p2)	16	RW	017Fh
40_01F4h	DFI Frequency Ratio (DfiFreqRatio_p2)	16	RW	0001h
60_0010h	Impedance Calibration Clock Ratio (CalUclkInfo_p3)	16	RW	0320h
60_0016h	PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p3)	16	RW	0000h
60_0018h	PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p3)	16	RW	0000h
60_001Ah	PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p3)	16	RW	0000h
60_001Ch	PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p3)	16	RW	0000h
60_0020h	Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p3)	16	RW	0000h
60_0032h	Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p3)	16	RW	0000h
60_0040h	HWT MaxReadLatency. (HwtMRL_p3)	16	RW	0006h
60_0048h	Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p3)	16	RW	0008h
60_005Ah	This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p3)	16	RW	0000h
60_005Ch	Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p3)	16	RW	0000h
60_00ACh	READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p3)	16	RW	0006h
60_00F8h	DLL gain control (DllGainCtl_p3)	16	RW	00A1h
60_0160h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p3)	16	RW	00E4h

Table continues on the next page...

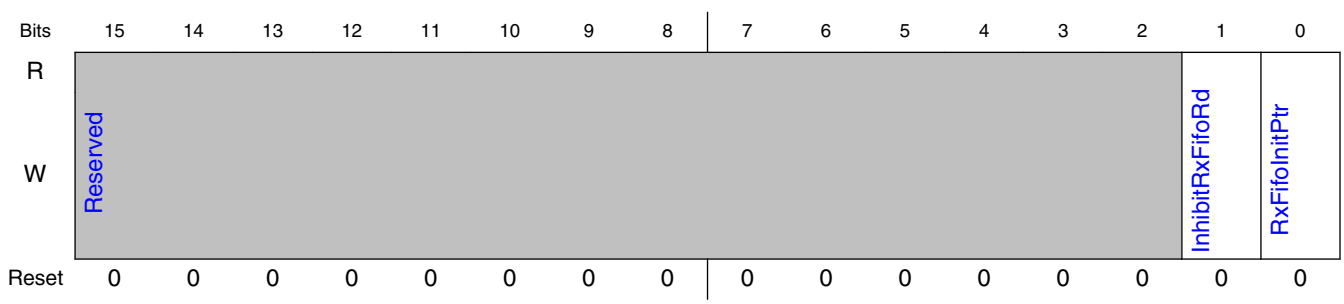
Offset	Register	Width (In bits)	Access	Reset value
60_0164h	PHY Global Vref Controls (VrefInGlobal_p3)	16	RW	0200h
60_0168h	Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p3)	16	RW	00E4h
60_018Ah	PState dependent PLL Control Register 2 (PllCtrl2_p3)	16	RW	0019h
60_018Eh	PState dependent PLL Control Register 1 (PllCtrl1_p3)	16	RW	0020h
60_0194h	Additional controls for PLL CP/VCO modes of operation (PllTestMode_p3)	16	RW	0124h
60_0198h	PState dependent PLL Control Register 4 (PllCtrl4_p3)	16	RW	017Fh
60_01F4h	DFI Frequency Ratio (DfiFreqRatio_p3)	16	RW	0001h

9.4.3.6.2 Rx FIFO pointer initialization control (RxFifoInit)

9.4.3.6.2.1 Offset

Register	Offset
RxFifoInit	0h

9.4.3.6.2.2 Diagram



9.4.3.6.2.3 Fields

Field	Function
15-2 —	Reserved
1 InhibitRxFifoRd	This field is reserved for training FW use. This field is reserved for training FW use. Setting this inhibits reads of the PHYRXDATAFIFO.
0	Setting this bit will reset the PHY RXDATAFIFO read and write pointers.

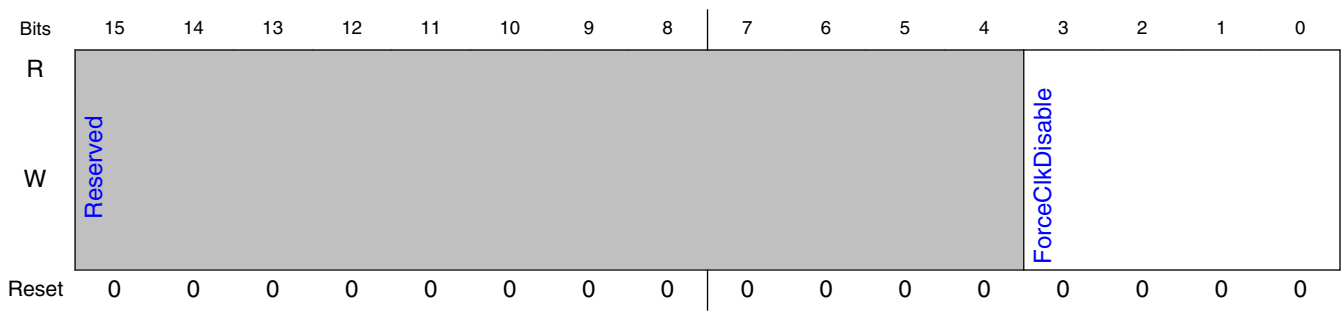
Field	Function
RxFifoInitPtr	Setting this bit will reset the PHY RXDATAFIFO read and write pointers. This bit must be Cleared to return to normal operation. This is not required in normal operation, but can be used instead of a dfi_ctrlupdate (for controllers that don't support ctrlupdate)

9.4.3.6.3 Clock gating control (ForceClkDisable)

9.4.3.6.3.1 Offset

Register	Offset
ForceClkDisable	2h

9.4.3.6.3.2 Diagram



9.4.3.6.3.3 Fields

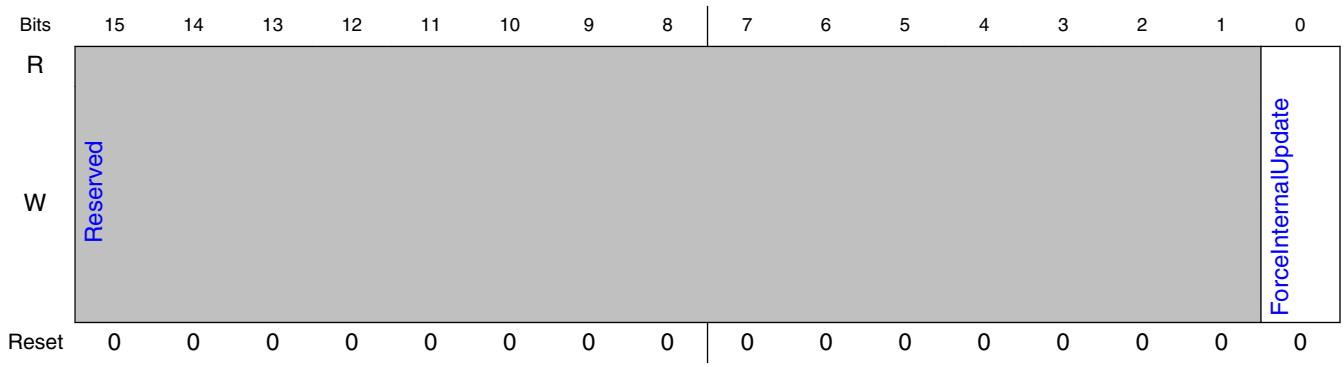
Field	Function
15-4 —	Reserved
3-0 ForceClkDisable	This CSR forces the gating of MEMCLKs driven from the PHY ForceClkDisable[0] - controls CLK_H/L0 ForceClkDisable[1] - controls CLK_H/L1 (if present) ForceClkDisable[2] - controls CLK_H/L2 (if present) ForceClkDisable[3] - controls CLK_H/L3 (if present)

9.4.3.6.4 This Register used by Training Firmware to force an internal PHY Update Event. (ForceInternalUpdate)

9.4.3.6.4.1 Offset

Register	Offset
ForceInternalUpdate	6h

9.4.3.6.4.2 Diagram



9.4.3.6.4.3 Fields

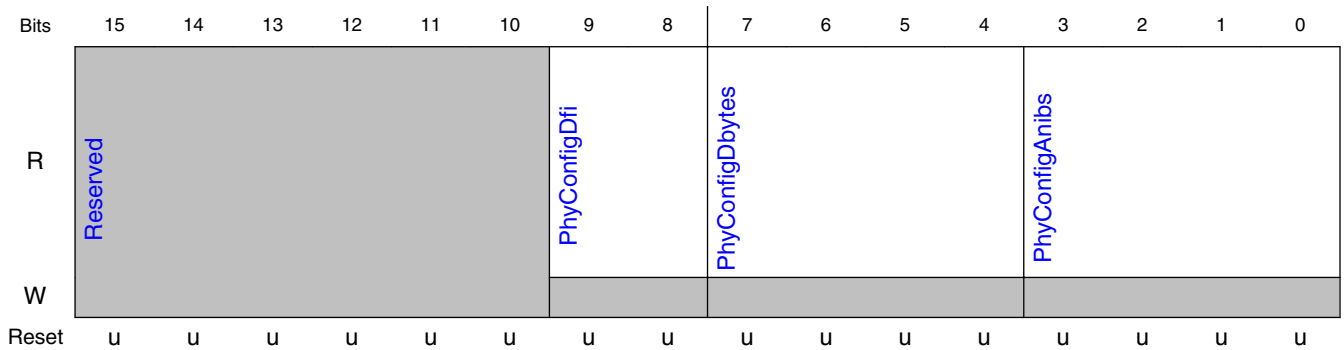
Field	Function
15-1 —	Reserved
0 ForceInternalUpdate	This Register is used by Training Firmware to force an internal PHY Update Event. This Register is used by Training Firmware to force an internal PHY Update Event. Optionally can be used by System Software to issue a PHY update Event. The register may be written to 1'b1 only when all of the following are true: [+] The DFI interface is offline. [+] The Training Hardware is sending only DES and all transactions have retired. Additionally, this Register must stay asserted for at least 32 DFICLKs before clearing. Prematurely clearing this register may result in an incomplete update.

9.4.3.6.5 Read Only displays PHY Configuration. (PhyConfig)

9.4.3.6.5.1 Offset

Register	Offset
PhyConfig	8h

9.4.3.6.5.2 Diagram



9.4.3.6.5.3 Fields

Field	Function
15-10 —	Reserved
9-8 PhyConfigDfi	Returns the following value . Returns the following value ... depending on the define "0x1" if DWC_DDRPHY_DFI1_EXISTS is not defined "0x2" if DWC_DDRPHY_DFI1_EXISTS is defined
7-4 PhyConfigDbytes	Returns the following value . Returns the following value ... depending on the chosen define "0x1" if DWC_DDRPHY_NUM_DBYTES_1 "0x2" if DWC_DDRPHY_NUM_DBYTES_2 "0x3" if DWC_DDRPHY_NUM_DBYTES_3 "0x4" if DWC_DDRPHY_NUM_DBYTES_4 "0x5" if DWC_DDRPHY_NUM_DBYTES_5 "0x6" if DWC_DDRPHY_NUM_DBYTES_6 "0x7" if DWC_DDRPHY_NUM_DBYTES_7 "0x8" if DWC_DDRPHY_NUM_DBYTES_8 "0x9" if DWC_DDRPHY_NUM_DBYTES_9 "0xA" if DWC_DDRPHY_NUM_DBYTES_10
3-0 PhyConfigAnibs	Returns the following value . Returns the following value ... depending on the chosen define "0x3" if DWC_DDRPHY_NUM_ANIBS_3 "0x6" if DWC_DDRPHY_NUM_ANIBS_6 "0xA" if DWC_DDRPHY_NUM_ANIBS_10 "0xC" if DWC_DDRPHY_NUM_ANIBS_12

9.4.3.6.6 Test Bump Control1 (TestBumpCntrl1)

9.4.3.6.6.1 Offset

Register	Offset
TestBumpCntrl1	Eh

9.4.3.6.6.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TestPowerGateEn	TestExtVrefRange	TestSelExternalVref	TestGainCurrAdj					TestAnalogOutCtrl				TestBiasBypassEn	TestMajorMode		
W																
Reset	0	0	0	0	1	0	1	1	0	0	0	0	0	0	1	1

9.4.3.6.6.3 Fields

Field	Function
15 TestPowerGateEn	Do not use, for debug only
14 TestExtVrefRange	Setting this bit will extend the VREF DAC range for debug. Setting this bit will extend the VREF DAC range for debug. MUST BE SET TO ZERO in mission mode when MemAlert is enabled
13 TestSelExternalVref	Do not use, for debug only
12-8 TestGainCurrAdj	Adjust gain and current of analog observe RX amplifier stage at analog test point Recommended mission mode default = 5'b01011
7-4 TestAnalogOutCtrl	Select receiver internal analog signals to monitor at analog test point 0xxx: AnalogTestOut=HiZ 1000: AnalogTestOut=VSS 1001: AnalogTestOut=vref_dfe0 -- observe by sweeping MALERTVrefLevel 1010: AnalogTestOut=vref_dfe1 -- observe by sweeping MALERTVrefLevel 1011: AnalogTestOut=VSS 1100: AnalogTestOut=vstg2 1101: AnalogTestOut=vcasc_cs1 1110: AnalogTestOut=vbias_cs1 Recommended mission mode default = 4'b0000
3	Do not use, for debug only

Table continues on the next page...

Field	Function
TestBiasBypassEn	
2-0 TestMajorMode	Selects the major mode of operation for the receiver. Selects the major mode of operation for the receiver. These modes are mutually exclusive. 000 - setting for DDR3-RDIMM systems to receive ERROUT_n 001 - reserved 010 - reserved 011 - setting for DDR4 systems to receive Alert_n 100 - reserved for debug 101 - reserved 110 - reserved 111 - reserved for debug

9.4.3.6.7 Impedance Calibration Clock Ratio (CalUclkInfo_p0)

9.4.3.6.7.1 Offset

Register	Offset
CalUclkInfo_p0	10h

9.4.3.6.7.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						CalUClkTicksPer1uS									
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0

9.4.3.6.7.3 Fields

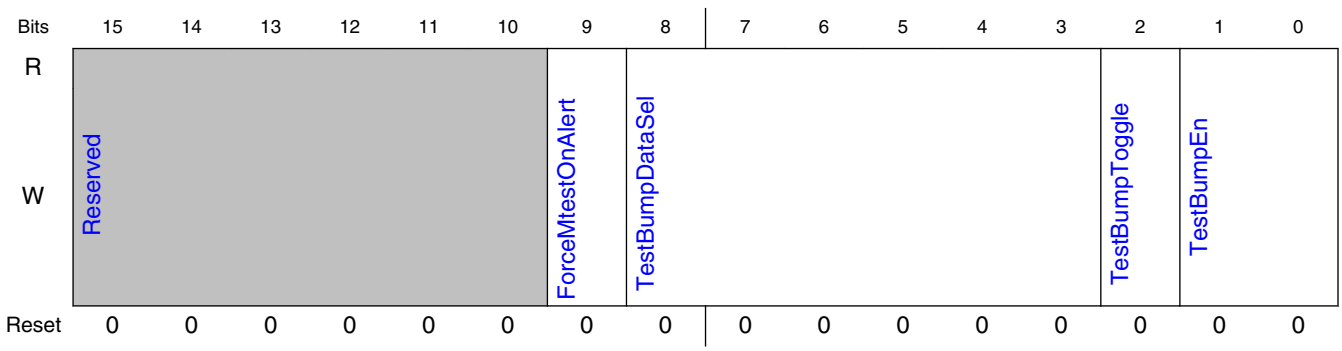
Field	Function
15-10 —	Reserved
9-0 CalUClkTicksPer1uS	Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. if (DfiClk < 24MHz) CalUclkInfo = 24 else CalUclkInfo = (number of DfiClks in 1us)

9.4.3.6.8 Test Bump Control (TestBumpCntrl)

9.4.3.6.8.1 Offset

Register	Offset
TestBumpCntrl	14h

9.4.3.6.8.2 Diagram



9.4.3.6.8.3 Fields

Field	Function
15-10 —	Reserved
9 ForceMtestOnAlert	When set, causes the Digital Observation output pin to be driven onto BP_ALERT_N
8-3 TestBumpDataSel	RVSD.
2 TestBumpToggle	This field controls the output function of the signal Digital Observation Pin, if available in the configuration of the PHY. This field controls the output function of the signal Digital Observation Pin, if available in the configuration of the PHY. When set to 1'b0 -- the output will be tristated When set to 1'b1 -- the output will be selected by MtestMuxSel
1-0 TestBumpEn	Field TestBumpEn[1:0] controls the output function of: the signal BP_ALERT_N. Field TestBumpEn[1:0] controls the output function of: the signal BP_ALERT_N.

Field	Function
	TestBumpEn Value on BP_ALERT_N output ---------- ---------- 2'b00 high-z, 2'b01 Pllout-divided-by-2, works up to full data rate 2'b10 PllDigTst[1], works up to 400MHz 2'b11 PllDigTst[1], works up to 400MHz

9.4.3.6.9 PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p0)

9.4.3.6.9.1 Offset

Register	Offset
Seq0BDLY0_p0	16h

9.4.3.6.9.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY0_p0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.9.3 Fields

Field	Function
15-0 Seq0BDLY0_p0	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: ---------- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it

Field	Function
	should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --------- ---------- ---------- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.10 PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p0)

9.4.3.6.10.1 Offset

Register	Offset
Seq0BDLY1_p0	18h

9.4.3.6.10.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY1_p0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.10.3 Fields

Field	Function
15-0 Seq0BDLY1_p0	PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --------------------------------

Field	Function
	These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --------- ---------- ---------- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.11 PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p0)

9.4.3.6.11.1 Offset

Register	Offset
Seq0BDLY2_p0	1Ah

9.4.3.6.11.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY2_p0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.11.3 Fields

Field	Function
15-0 Seq0BDLY2_p0	PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions

Field	Function
	in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.12 PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p0)

9.4.3.6.12.1 Offset

Register	Offset
Seq0BDLY3_p0	1Ch

9.4.3.6.12.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY3_p0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.12.3 Fields

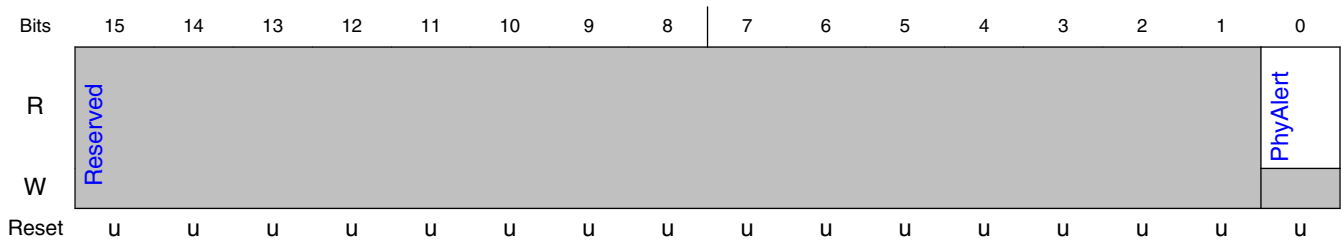
Field	Function
15-0 Seq0BDLY3_p0	PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.13 PHY Alert status bit (PhyAlertStatus)

9.4.3.6.13.1 Offset

Register	Offset
PhyAlertStatus	1Eh

9.4.3.6.13.2 Diagram



9.4.3.6.13.3 Fields

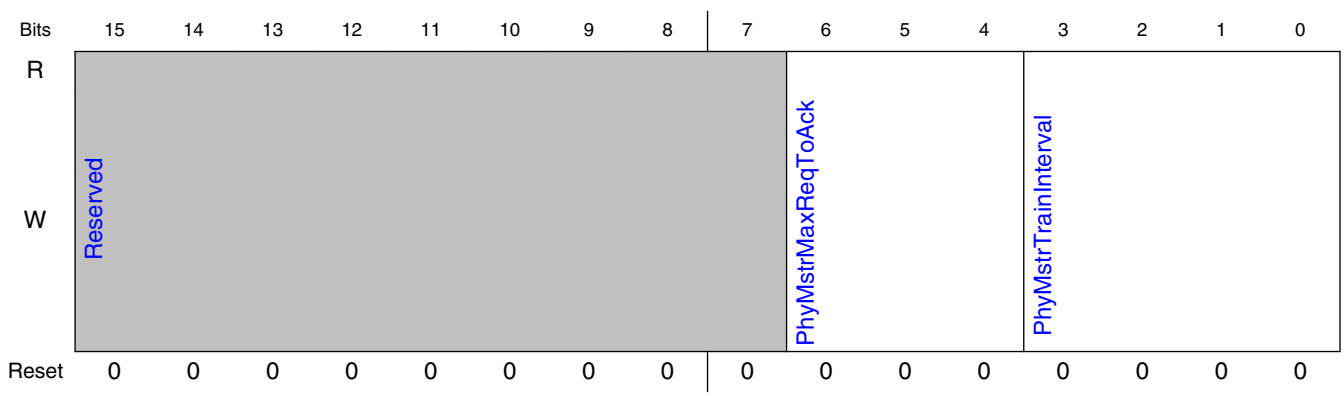
Field	Function
15-1 —	Reserved
0 PhyAlert	Current state of ALERT_N.

9.4.3.6.14 Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p0)

9.4.3.6.14.1 Offset

Register	Offset
PPTTrainSetup_p0	20h

9.4.3.6.14.2 Diagram



9.4.3.6.14.3 Fields

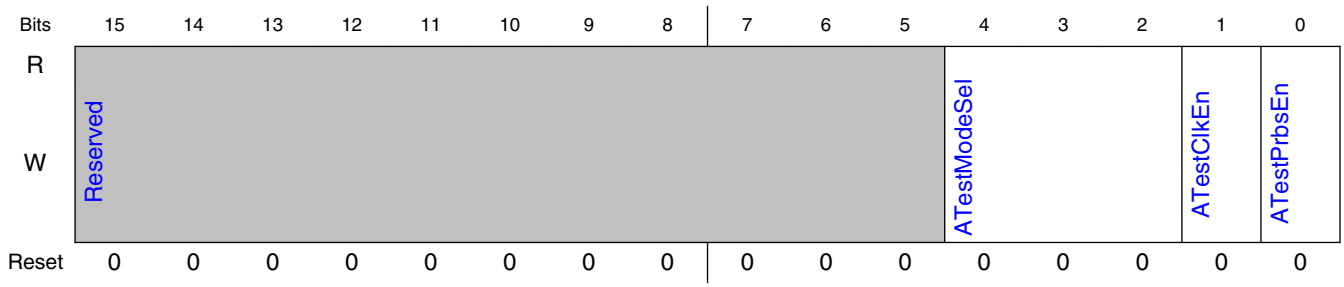
Field	Function
15-7 —	Reserved
6-4 PhyMstrMaxReqToAck	Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. 3'b000 Disable PHY Master Interface 3'b001 set tPHYMSTR_resp. = 512 MEMCLKs 3'b010 set tPHYMSTR_resp. = 1024 MEMCLKs 3'b011 set tPHYMSTR_resp. = 2048 MEMCLKs 3'b100 set tPHYMSTR_resp. = 4096 MEMCLKs 3'b101 set tPHYMSTR_resp. = 8192 MEMCLKs 3'b110 set tPHYMSTR_resp. = 32768 MEMCLKs 3'b111 set tPHYMSTR_resp. = undefined
3-0 PhyMstrTrainInterval	Bits 3:0 of this register specifies the time between the end of one training and the start of the next. Bits 3:0 of this register specifies the time between the end of one training and the start of the next. (As per section 11 of the Preliminary DFI 4.0 Specification, V2, it is the max expected time from dfi_init_complete asserted to tdfi_phymstr_ack asserted). 4'b0000 Disable PHY Master Interface 4'b0001 PHY MASTER Request Interval = 524288 MEMCLKs 4'b0010 PHY MASTER Request Interval = 1048576 MEMCLKs 4'b0011 PHY MASTER Request Interval = 2097152 MEMCLKs 4'b0100 PHY MASTER Request Interval = 4194304 MEMCLKs 4'b0101 PHY MASTER Request Interval = 8388608 MEMCLKs 4'b0110 PHY MASTER Request Interval = 16777216 MEMCLKs 4'b0111 PHY MASTER Request Interval = 33554432 MEMCLKs 4'b1000 PHY MASTER Request Interval = 67108864 MEMCLKs 4'b1001 PHY MASTER Request Interval = 134217728 MEMCLKs 4'b1010 PHY MASTER Request Interval = 268435456 MEMCLKs 4'b1011 - 4'b1111 PHY MASTER Request Interval = undefined

9.4.3.6.15 ATestMode control (ATestMode)

9.4.3.6.15.1 Offset

Register	Offset
ATestMode	24h

9.4.3.6.15.2 Diagram



9.4.3.6.15.3 Fields

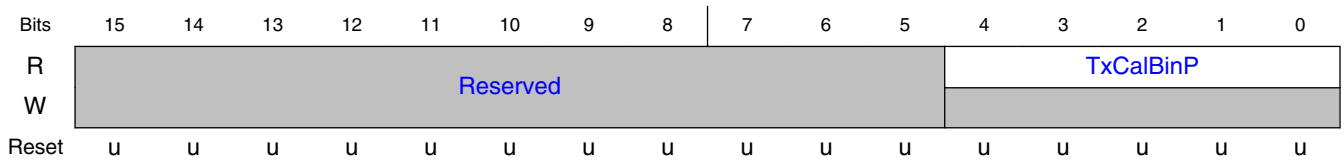
Field	Function
15-5 —	Reserved
4-2 ATestModeSel	Master Mode select for ATest (Loopback) 000 - Mission mode, all ATest disabled, loopback receivers powered down 001 - External Loopback mode [Single data rate pattern - dfi_cas sent to all lanes] 010 - Internal Loopback mode [Single data rate pattern] 011 - Internal Loopback mode [Double data rate pattern] 100 - External Loopback mode [Single data rate pattern - corresponding DFI signal sent to each lane]
1 ATestClkEn	Enables the clock for loopback PRBS7 testing for all BP_A* pins.
0 ATestPrbsEn	Enables loopback PRBS7 testing of all the DDR output pins in this chiplet. Enables loopback PRBS7 testing of all the DDR output pins in this chiplet. The asserting edge resets all the individual TestPrbsErrCnt & consequently TestPrbsErr; there is no other reset for the TestPrbsErrCnt. The ATestClkEn CSR below must be enabled first prior to enabling this bit.

9.4.3.6.16 TX P Impedance Calibration observation (TxCalBinP)

9.4.3.6.16.1 Offset

Register	Offset
TxCalBinP	28h

9.4.3.6.16.2 Diagram



9.4.3.6.16.3 Fields

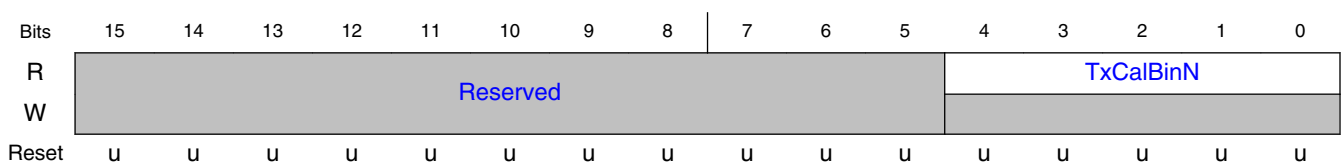
Field	Function
15-5 —	Reserved
4-0 TxCalBinP	This csr holds the binary result of the 31 bit thermometer pullup code.

9.4.3.6.17 TX N Impedance Calibration observation (TxCalBinN)

9.4.3.6.17.1 Offset

Register	Offset
TxCalBinN	2Ah

9.4.3.6.17.2 Diagram



9.4.3.6.17.3 Fields

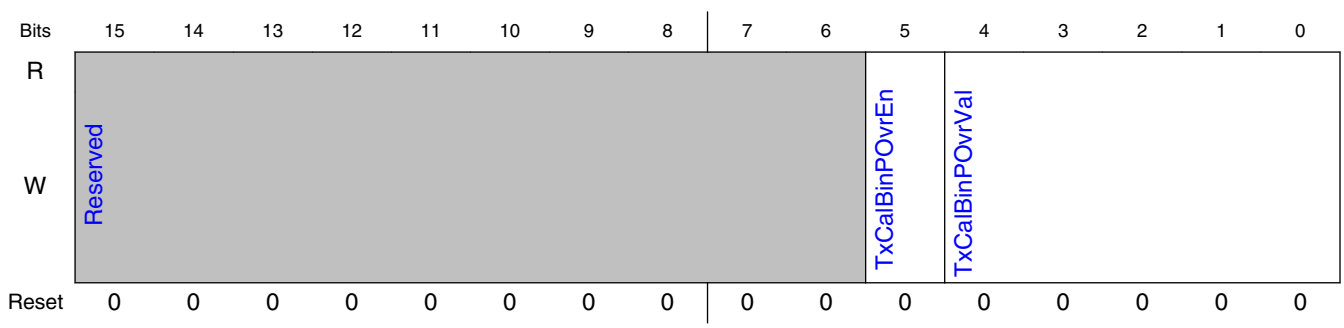
Field	Function
15-5 —	Reserved
4-0 TxCalBinN	This csr holds the binary result of the 31 bit thermometer pulldown code.

9.4.3.6.18 TX P Impedance Calibration override (TxCalPOvr)

9.4.3.6.18.1 Offset

Register	Offset
TxCalPOvr	2Ch

9.4.3.6.18.2 Diagram



9.4.3.6.18.3 Fields

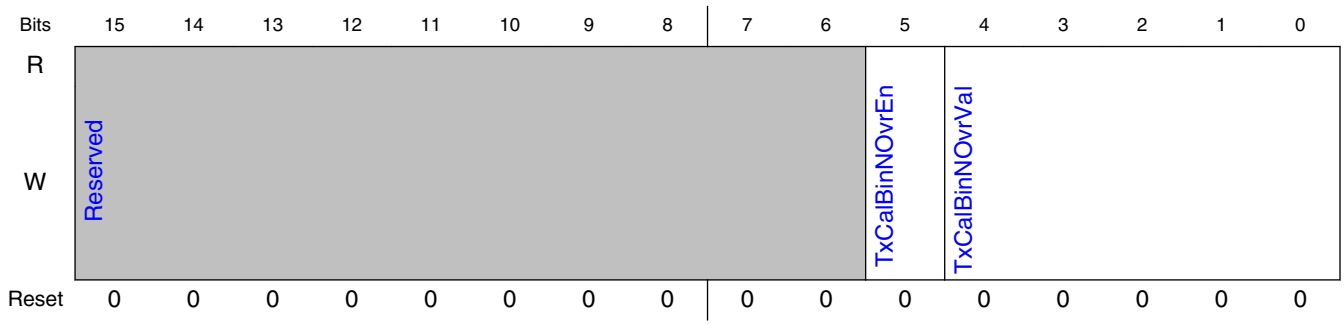
Field	Function
15-6 —	Reserved
5 TxCalBinPOvrEn	1 = use the override value present in Register TxCalBinPOvrVal 0 = don't.
4-0 TxCalBinPOvrVal	The binary value which can overide the Register TxCalBinP calibrator results if Register TxCalBinPOvrEn is set.

9.4.3.6.19 TX N Impedance Calibration override (TxCalNOvr)

9.4.3.6.19.1 Offset

Register	Offset
TxCalNOvr	2Eh

9.4.3.6.19.2 Diagram



9.4.3.6.19.3 Fields

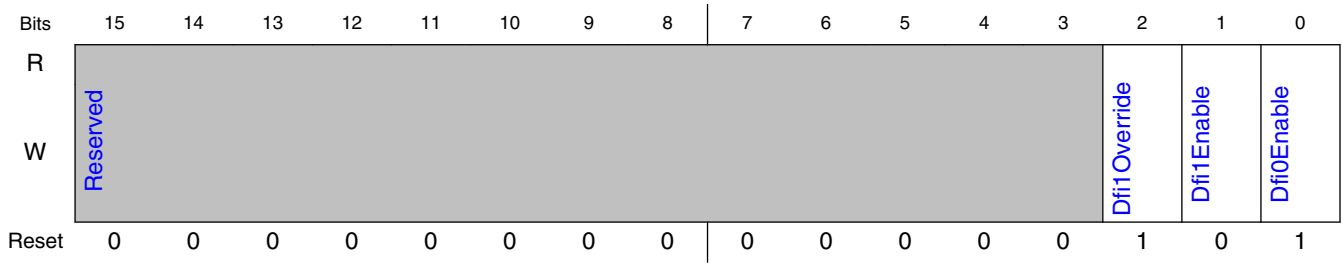
Field	Function
15-6 —	Reserved
5 TxCalBinNOvrEn	1 = use the override value present in Register TxCalBinNOvrVal 0 = don't.
4-0 TxCalBinNOvrVal	The binary value which can overide the Register TxCalBinN calibrator results if Register TxCalBinPOvrEn is set.

9.4.3.6.20 Enables for update and low-power interfaces for DFI0 and DFI1 (DfiMode)

9.4.3.6.20.1 Offset

Register	Offset
DfiMode	30h

9.4.3.6.20.2 Diagram



9.4.3.6.20.3 Fields

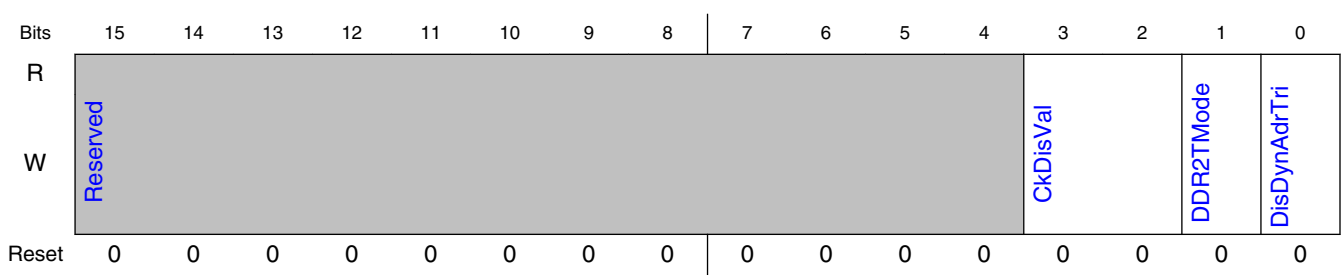
Field	Function
15-3 —	Reserved
2 Dfi1Override	DFI0 is used to control the PHY logic associated with both DFI0 and DFI1
1 Dfi1Enable	Enables operation for the PHY logic associated with DFI1
0 Dfi0Enable	Enables operation for the PHY logic associated with DFI0

9.4.3.6.21 Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p0)

9.4.3.6.21.1 Offset

Register	Offset
TristateModeCA_p0	32h

9.4.3.6.21.2 Diagram



9.4.3.6.21.3 Fields

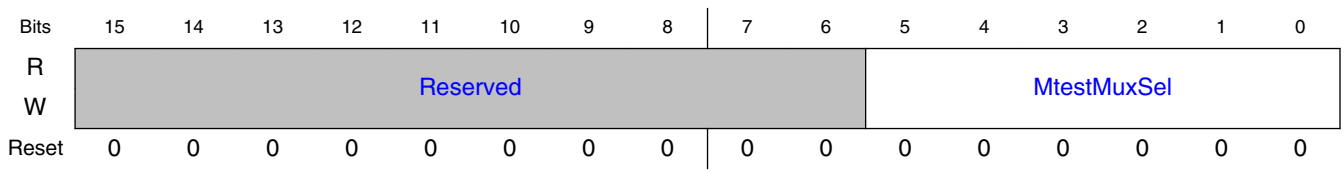
Field	Function
15-4 —	Reserved
3-2 CkDisVal	The PHY provides 4 memory clocks, n=0. The PHY provides 4 memory clocks, n=0..3. When the toggling of memory clock CK_t[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_t[n] is driven with Register CkDisVal[1]. When the toggling of memory clock CK_c[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_c[n] is driven with ~(Register CkDisVal[0]) (ie inverted). Note that the non-toggling differential memory clock CK_t[n],CK_c[n] may be driven with any combination, LL,LH,HL,HH; the default CK_t[n],CK_c[n]=LH.
1 DDR2TMode	Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. This CSR bit has no effect when DisDynAdrTri==1
0 DisDynAdrTri	When DisDynAdrTri=1, Dynamic Tristating is disabled. When DisDynAdrTri=1, Dynamic Tristating is disabled. Dynamic Tristating is on by default. . Dynamic Tristating should be disabled (DisDynAdrTri=0) for these modes: 1. DDR3/2T if the controller cannot follow the 2T PHY tristate protocol. 2. DDR4/2T/2N if the controller cannot follow the 2T PHY tristate protocol. 3. LPDDR4 . When DisDynAdrTri=0 (default), In DDR3 mode, The following SDRAM pins can be dynamically tristated: A,BA,RAS_n,CAS_n,WE_n In DDR4 mode, The following SDRAM pins can be dynamically tristated: A,BA,BG,ACT_n In LPDDR3 mode, The following SDRAM pins can be dynamically tristated: CA[*] . In 1T mode, the PHY will tristate the relevant pins when all ranks are DESelected . In 2T mode (or geardown), the PHY will tristate the relevant pins when: D3 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0 = 1,1,1,0 D4 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0,ACT_n = 1,1,1,0,1 When in this mode, the controller should avoid sending the above patterns with any rank selected. [e.g. NOPs sent by the controller should have BA0 set to a non-zero value]

9.4.3.6.22 Digital Observation Pin control (MtestMuxSel)

9.4.3.6.22.1 Offset

Register	Offset
MtestMuxSel	34h

9.4.3.6.22.2 Diagram



9.4.3.6.22.3 Fields

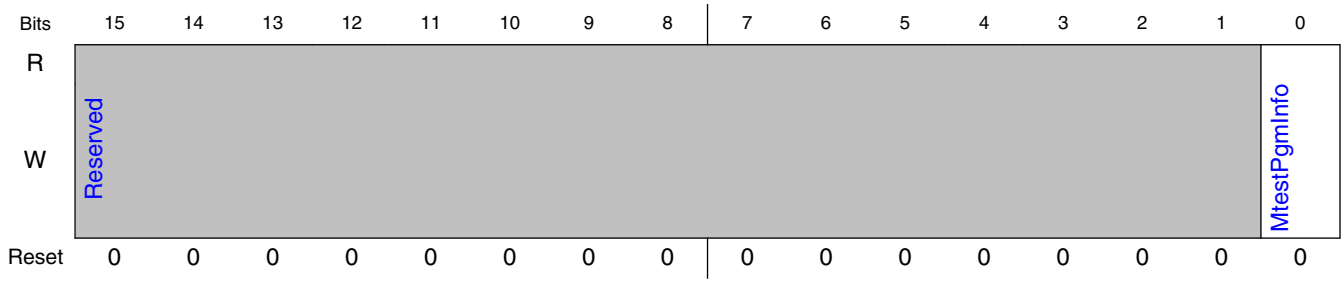
Field	Function
15-6 —	Reserved
5-0 MtestMuxSel	Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Controls for the 64-1 mux for asynchronous data to the Digital Observation Pin. Encoding 6'h0 --> Drive 0 from this chiplet (allows flat 'OR' of pass-through information) Encoding 6'h01:1f --> Select local data from AC/DBYTE/MASTER macro Encoding 6'h20 --> Reserved (Drive 0 from macro into mux; Drive 0 from PUB synth logic path into mux) Encoding 6'h21:3f --> Select local data from PUB AC/DBYTE/MASTER synthesized logic Note: See PUB documentation for how, or if, the Digital Observation Pin is mapped to a physical bump in this configuration.

9.4.3.6.23 Digital Observation Pin program info for debug (MtestPgmInfo)

9.4.3.6.23.1 Offset

Register	Offset
MtestPgmInfo	36h

9.4.3.6.23.2 Diagram



9.4.3.6.23.3 Fields

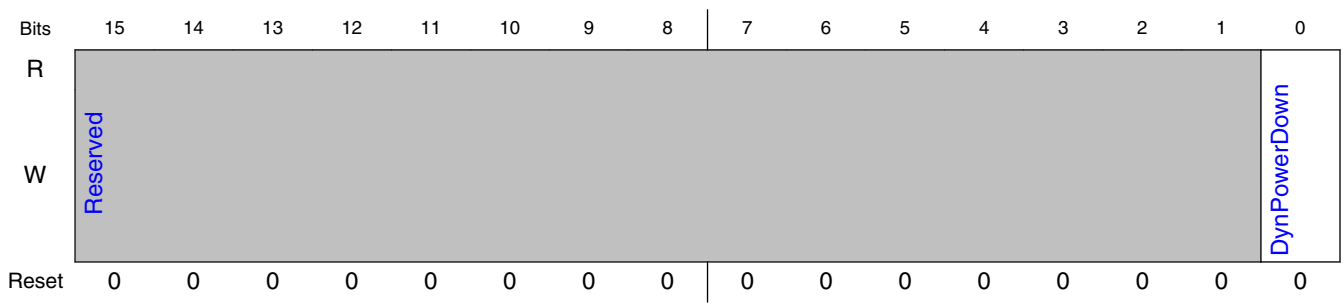
Field	Function
15-1 —	Reserved
0 MtestPgmInfo	The value of this csr may be driven onto the Digital Observation Pin. The value of this csr may be driven onto the Digital Observation Pin. It has no other hardware effect.

9.4.3.6.24 Dynaimc Power Up/Down control (DynPwrDnUp)

9.4.3.6.24.1 Offset

Register	Offset
DynPwrDnUp	38h

9.4.3.6.24.2 Diagram



9.4.3.6.24.3 Fields

Field	Function
15-1 —	Reserved
0 DynPowerDown	1 - analog circuitry (voltage dacs, bias gen) is turned off. 1 - analog circuitry (voltage dacs, bias gen) is turned off. 0 - normal operation.

9.4.3.6.25 PHY Technology ID Register (PhyTID)

9.4.3.6.25.1 Offset

Register	Offset
PhyTID	3Ch

9.4.3.6.25.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PhyTID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.25.3 Fields

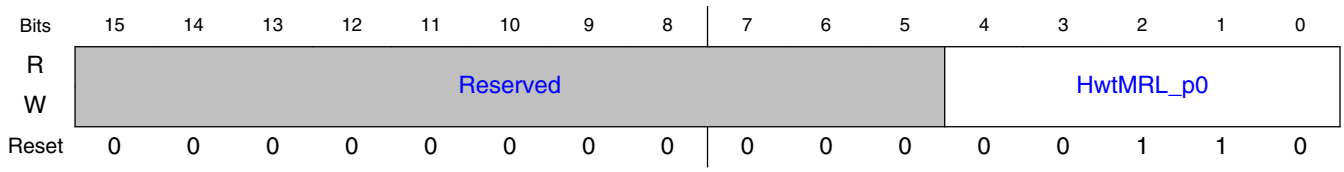
Field	Function
15-0 PhyTID	This register is a placeholder to store technology-specific information

9.4.3.6.26 HWT MaxReadLatency. (HwtMRL_p0)

9.4.3.6.26.1 Offset

Register	Offset
HwtMRL_p0	40h

9.4.3.6.26.2 Diagram



9.4.3.6.26.3 Fields

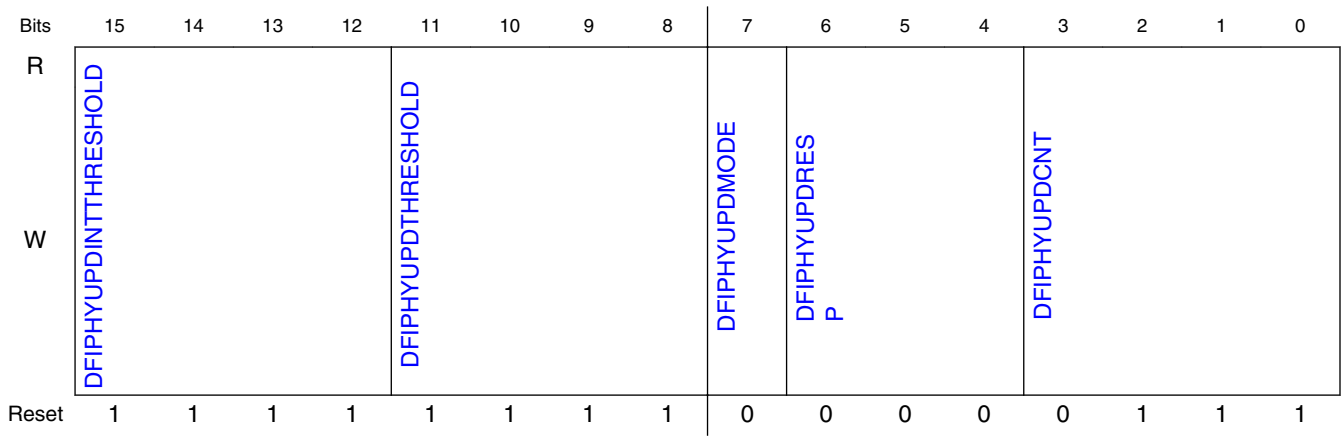
Field	Function
15-5 —	Reserved
4-0 HwtMRL_p0	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. The CSR is in units of two mem clocks; that is, a unit change in the LSB is a change in MRL of two mem clocks. This MASTER copy of MRL is used by the PHY training hardware only.

9.4.3.6.27 DFI PhyUpdate Request time counter (in MEMCLKs) (DFIPHYUPD)

9.4.3.6.27.1 Offset

Register	Offset
DFIPHYUPD	42h

9.4.3.6.27.2 Diagram



9.4.3.6.27.3 Fields

Field	Function
15-12 DFIPHYUPDINTTHRESHOLD	This subfield is similar to DFIPHYUPDTHRESHOLD except that rather than affecting the Phy Update request, it affects only the threshold used to generate the VT Drift Alarm Interrupt. This subfield is similar to DFIPHYUPDTHRESHOLD except that rather than affecting the Phy Update request, it affects only the threshold used to generate the VT Drift Alarm Interrupt. Further, the actual threshold is twice what is specified in the field.
11-8 DFIPHYUPDTHRESHOLD	4'h0 Disable Threshold-based Phy Update Requests when DFIPHYUPDMODE==1'b1 Nonzero codes are the threshold value for the change in the master LCDL 1UI phase code since the last Phy Update Request that will trigger a new Phy Update Request; If (current_1UI_phase - last_1UI_phase) > DFIPHYUPDTHRESHOLD, then a Phy Update will be requested. 4'h0 Disable Threshold-based Phy Update Requests when DFIPHYUPDMODE==1'b1 Nonzero codes are the threshold value for the change in the master LCDL 1UI phase code since the last Phy Update Request that will trigger a new Phy Update Request; If (current_1UI_phase - last_1UI_phase) > DFIPHYUPDTHRESHOLD, then a Phy Update will be requested. In units of native-control phase units, Minimum value is 1. With DFIPHYUPDTHRESHOLD=1, a master LCDL dithering between two values will not cause a Phy Update Request.
7 DFIPHYUPDMODE	1'b0 [Default] deterministic timer-based Phy Update Requests; enables multi-channel/multi-phy lockstep operation. 1'b0 [Default] deterministic timer-based Phy Update Requests; enables multi-channel/multi-phy lockstep operation. 1'b1 Non-deterministic threshold-based Phy Update Requests; multi-channel/multi-phy operation not permitted.
6-4 DFIPHYUPDRESP	Enforces the t_phyupd_resp time, the maximum time that is allowed to controller to respond to the request for a PHY update. Enforces the t_phyupd_resp time, the maximum time that is allowed to controller

Table continues on the next page...

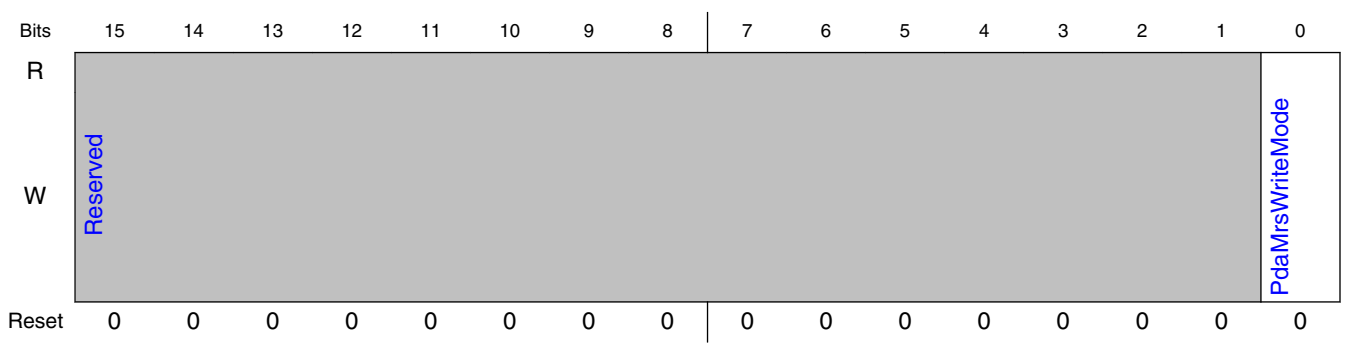
Field	Function
	to respond to the request for a PHY update. A dfi_error will be signaled if there is no acknowledgement of the update request within nMEMCLKs_phyupd_resp. 3'b000 nMEMCLKs_phyupd_resp= 1024 MEMCLKs, default value 3'b001 nMEMCLKs_phyupd_resp= 2048 MEMCLKs 3'b010 nMEMCLKs_phyupd_resp= 4096 MEMCLKs 3'b011 nMEMCLKs_phyupd_resp= 8192 MEMCLKs, [not allowed in LPDDR3, LPDDR4 modes] Any code not enumerated above is RSVD and should not be set.
3-0 DFIPHYUPDCNT	This controls the interval between the end of a phyupdate transaction and a subsequent request. This controls the interval between the end of a phyupdate transaction and a subsequent request. 4'b0000 - Disable timer-based Phy Update when DFIPHYUPDMODE==1'b0 4'b0001 - 16K MEMCLKs minus nMEMCLKs_phyupd_resp 4'b0011 - 32K MEMCLKs minus nMEMCLKs_phyupd_resp 4'b0111 - 64K MEMCLKs minus nMEMCLKs_phyupd_resp 4'b1111 - 128K MEMCLKs minus nMEMCLKs_phyupd_resp Any code not enumerated above is RSVD and should not be set.

9.4.3.6.28 Controls the write DQ generation for Per-Dram-Addressing of MRS (PdaMrsWriteMode)

9.4.3.6.28.1 Offset

Register	Offset
PdaMrsWriteMode	44h

9.4.3.6.28.2 Diagram



9.4.3.6.28.3 Fields

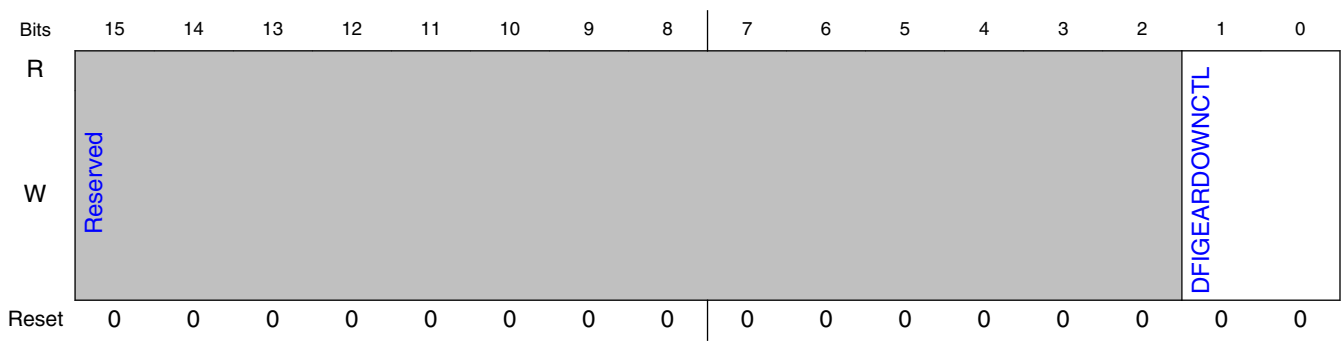
Field	Function
15-1 —	Reserved
0 PdaMrsWriteMode	Controls the write DQ generation per the timing requirements on the DQ signals used for Per-Dram-Addressing mode of MRS commands. Controls the write DQ generation per the timing requirements on the DQ signals used for Per-Dram-Addressing mode of MRS commands. When asserted, the DQ are driven for an additional UI before the logical burst and for an additonal UI after the logical burst. The early UI has the value of first logical UI, and the late UI has the value of the last logical UI. The timing of the logical burst is unchanged; no modification to csr TxDqDly is required or performed. Strongly recommended to be cleared after completing the MRS that clears MR3 A4 and for mission-mode writes, else required bubbles will be larger and the power consumed will be greater. See JEDEC MR3 A4, Per DRAM Addresssability.

9.4.3.6.29 Controls whether dfi_geardown_en will cause CS and CKE timing to change. (DFIGEARDOWNCTL)

9.4.3.6.29.1 Offset

Register	Offset
DFIGEARDOWNCTL	46h

9.4.3.6.29.2 Diagram



9.4.3.6.29.3 Fields

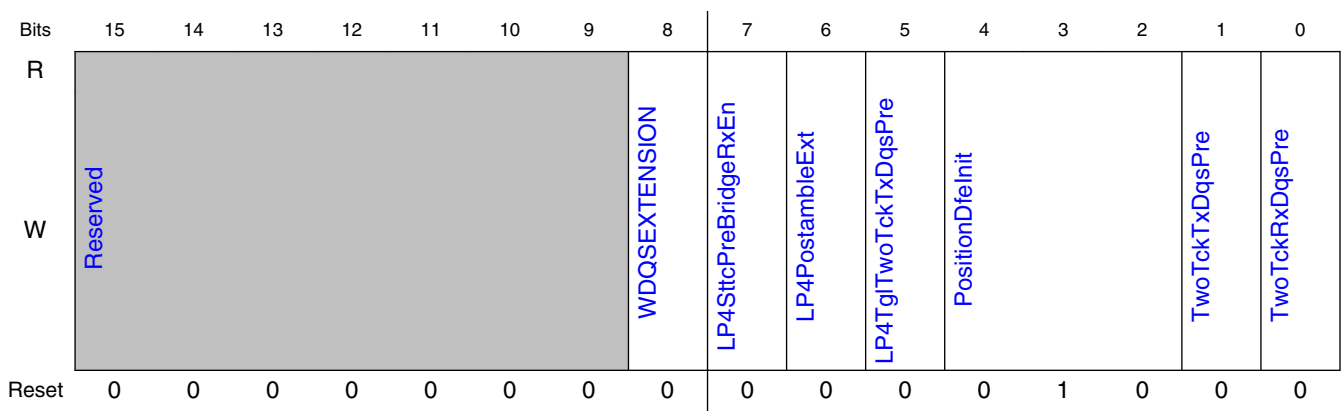
Field	Function
15-2 —	Reserved
1-0 DFIGEARDOWNCTL	DFIGEARDOWNCTL[0] controls whether dfi_geardown_en will cause chip-select (CS) timing to change. DFIGEARDOWNCTL[0] controls whether dfi_geardown_en will cause chip-select (CS) timing to change. 0 - CS timing will not be dynamically modified wtih dfi_geardown_en 1 - CS will be delayed by an additional 1UI dynamically when dfi_geardown_en==1 Note, this CSR must be set to zero if the PHY is not in DDR4 Mode DFIGEARDOWNCTL[1] controls whether dfi_geardown_en will cause clock-enable (CKE) timing to change. 0 - CKE timing will not be dynamically modified with dfi_geardown_en 1 - CKE will be delayed by an additional 1UI dynamically when dfi_geardown_en==1 Note, this CSR must be set to zero if the PHY is not in DDR4 Mode

9.4.3.6.30 Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p0)

9.4.3.6.30.1 Offset

Register	Offset
DqsPreambleControl_p0	48h

9.4.3.6.30.2 Diagram



9.4.3.6.30.3 Fields

Field	Function
15-9 —	Reserved
8 WDQSEXTENSION	When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. See DesignWare Cores LPDDR4 MultiPHY: WDQS Extension Application Note. Use of WDQSEXTENSION requires POdtTailWidth=3 and POdtStartDelay=00
7 LP4SttcPreBridgeRxEn	Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. LPDDR4 static-preamble mode is set in the DRAMs with MR1, OP[3]=0.
6 LP4PostambleExt	In LPDDR4 mode must be set to extend the write postamble. In LPDDR4 mode must be set to extend the write postamble. 0: half-memclk postamble, as in D4 1: one-and-one-half-memclk postamble; see LPDDR4 Spec MR3, OP[1] WR PST, vendor-specific function.
5 LP4TglTwoTckTxDqsPre	Used in LPDDR4 mode to modify the early preamble when Register TwoTckTxDqsPre=1 0: level first-memclk preamble 1: toggling first-memclk preamble
4-2 PositionDfeInit	For DDR4 phy only when receive DFE is enabled. For DDR4 phy only when receive DFE is enabled. PositionDfeInit[2]=1 causes DfeInit to be asserted late PositionDfeInit[1]=1 causes DfeInit to be asserted at the nominal time PositionDfeInit[0]=1 causes DfeInit to be asserted early PositionDfeInit=3'b010 is the nominal, recommended value for leading edge used by receiver.
1 TwoTckTxDqsPre	0: Standard 1tck TxDqs Preamble 1: Enable Optional D4 2tck TxDqs Preamble The DDR4 MR4 A12 is Write Preamble, 1=2nCK, 0=1nCK.
0 TwoTckRxDqsPre	Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. For LPDDR4, all read operations are 2nCK such that this control must be set to 1. Note the nominal trained-to-center point will be earlier relative to the first strobing edge of DQS than when not in this mode.

9.4.3.6.31 DBYTE module controls to select X4 Dram device mode (MasterX4Config)

9.4.3.6.31.1 Offset

Register	Offset
MasterX4Config	4Ah

9.4.3.6.31.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												X4TG			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.31.3 Fields

Field	Function
15-4 —	Reserved
3-0 X4TG	Set to 1 if this Timing Group/Rank is x4 (as opposed to x8) memory. Set to 1 if this Timing Group/Rank is x4 (as opposed to x8) memory. XXX[1|0] : TxReqDest/RxReqDest 0 X4/X8. XX[1|0]X : TxReqDest/RxReqDest 1 X4/X8. X[1|0]XX : TxReqDest/RxReqDest 2 X4/X8. [1|0]XXX : TxReqDest/RxReqDest 3 X4/X8. This CSR must be written before dfi_init_start is asserted.

9.4.3.6.32 Write level feedback DQ observability select. (WrLevBits)

9.4.3.6.32.1 Offset

Register	Offset
WrLevBits	4Ch

9.4.3.6.32.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								WrLevForDQSU				WrLevForDQSL			
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0

9.4.3.6.32.3 Fields

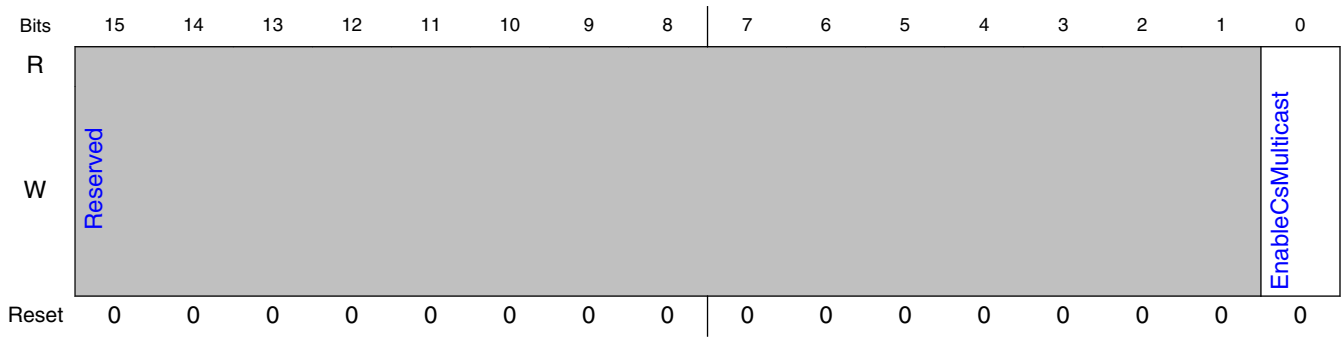
Field	Function
15-8 —	Reserved
7-4 WrLevForDQSU	Indicates which DQ bit is used for Write Levelization. Indicates which DQ bit is used for Write Levelization. 0000 - use DQ0 ... 0111 - use DQ7 1000 - use lDQ[3:0] 1001 - use lDQ[7:4] 1010 - use lDQ[7:0]
3-0 WrLevForDQSL	Indicates which DQ bit is used for Write Levelization. Indicates which DQ bit is used for Write Levelization. 0000 - use DQ0 ... 0111 - use DQ7 1000 - use lDQ[3:0] 1001 - use lDQ[7:4] 1010 - use lDQ[7:0]

9.4.3.6.33 In DDR4 Mode , this controls whether CS_N[3:2] should be multicast on CID[1:0] (EnableCsMulticast)

9.4.3.6.33.1 Offset

Register	Offset
EnableCsMulticast	4Eh

9.4.3.6.33.2 Diagram



9.4.3.6.33.3 Fields

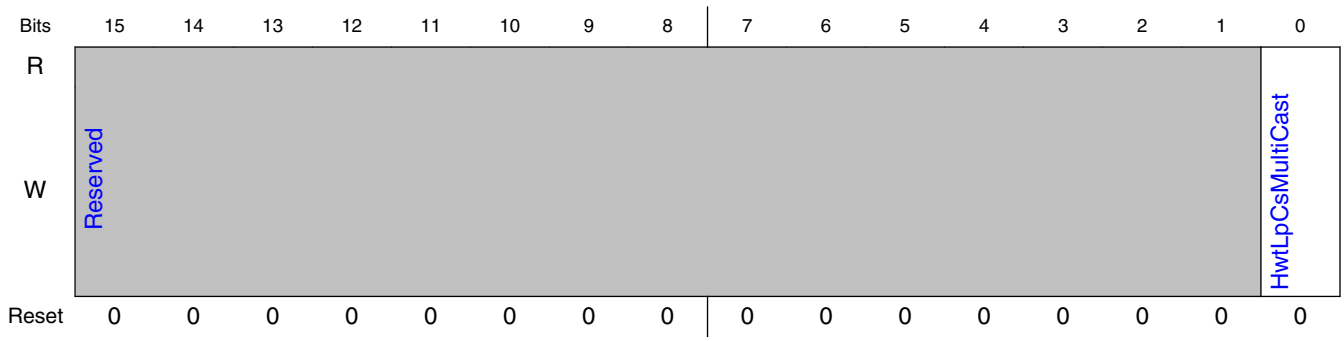
Field	Function
15-1 —	Reserved
0 EnableCsMulticast	In DDR4 Mode , this controls whether CS_N[3:2] should be multicast on CID[1:0] 0 - Do not override pins corresponding to cid[1:0] (dfi_cid[1:0] will connect to the pads) 1 - Overrirde pins corresponding to cid[1:0] with dfi_cs[3:2]. In DDR4 Mode , this controls whether CS_N[3:2] should be multicast on CID[1:0] 0 - Do not override pins corresponding to cid[1:0] (dfi_cid[1:0] will connect to the pads) 1 - Overrirde pins corresponding to cid[1:0] with dfi_cs[3:2]. (cs[3:2] will be multicast on cs[3:2] and cid[1:0])

9.4.3.6.34 Drives cs_n[0] onto cs_n[1] during training (HwtLpCsMultiCast)

9.4.3.6.34.1 Offset

Register	Offset
HwtLpCsMultiCast	50h

9.4.3.6.34.2 Diagram



9.4.3.6.34.3 Fields

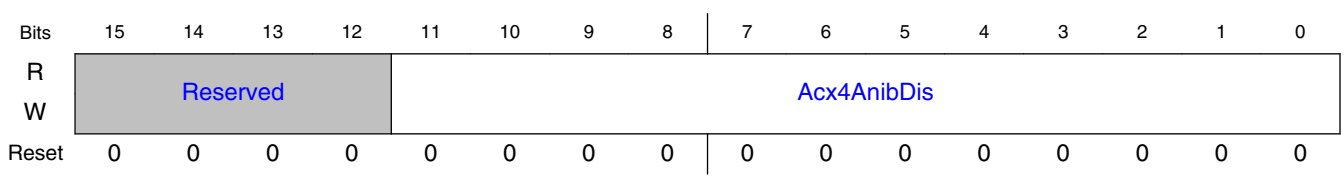
Field	Function
15-1 —	Reserved
0 HwtLpCsMultiCast	When set, drives cs_n[0] onto cs_n[1] during training

9.4.3.6.35 Disable for unused ACX Nibbles (Acx4AnibDis)

9.4.3.6.35.1 Offset

Register	Offset
Acx4AnibDis	58h

9.4.3.6.35.2 Diagram



9.4.3.6.35.3 Fields

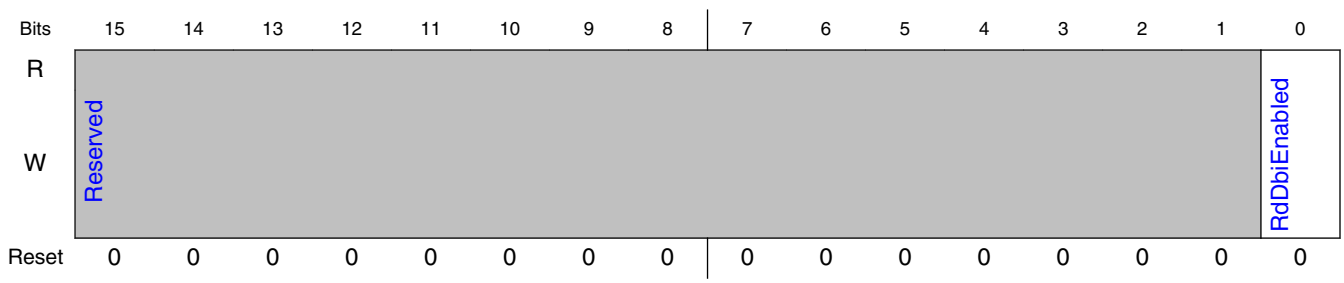
Field	Function
15-12 —	Reserved
11-0 Acx4AnibDis	When a bit is set, the corresponding ACX nibble is disabled (specifically, the I/O OE is disabled, as is the Dfi-side FIFO clock

9.4.3.6.36 This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p0)

9.4.3.6.36.1 Offset

Register	Offset
DMIPinPresent_p0	5Ah

9.4.3.6.36.2 Diagram



9.4.3.6.36.3 Fields

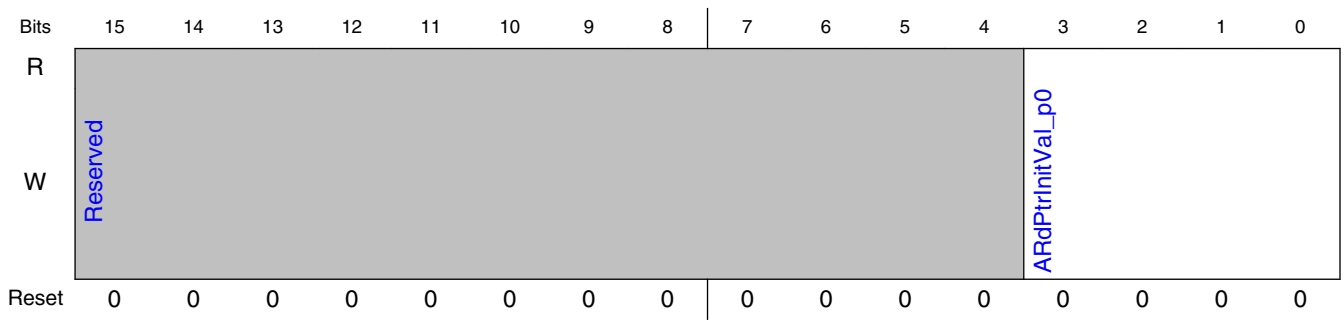
Field	Function
15-1 —	Reserved
0 RdDbiEnabled	This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. If set, the following DRAM MR should also be set [DDR4.MR5.A12=1 or LPDDR4.MR3.OP[7]=1]

9.4.3.6.37 Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p0)

9.4.3.6.37.1 Offset

Register	Offset
ARdPtrInitVal_p0	5Ch

9.4.3.6.37.2 Diagram



9.4.3.6.37.3 Fields

Field	Function
15-4 —	Reserved
3-0 ARdPtrInitVal_p0	This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. The units of this register are in UI. The pointer separation should be chosen to compensate for all sources of skew and drift of the PHY DFICLK and PCLK networks. Please see PUB databook section 8.1.1 This CSR must be programmed in Step C of the PHY Initialization sequence

9.4.3.6.38 DLL Mode control CSR for DBYTEs (DbyteDllModeCntrl)

9.4.3.6.38.1 Offset

Register	Offset
DbyteDllModeCntrl	74h

9.4.3.6.38.2 Diagram

9.4.3.6.38.3 Fields

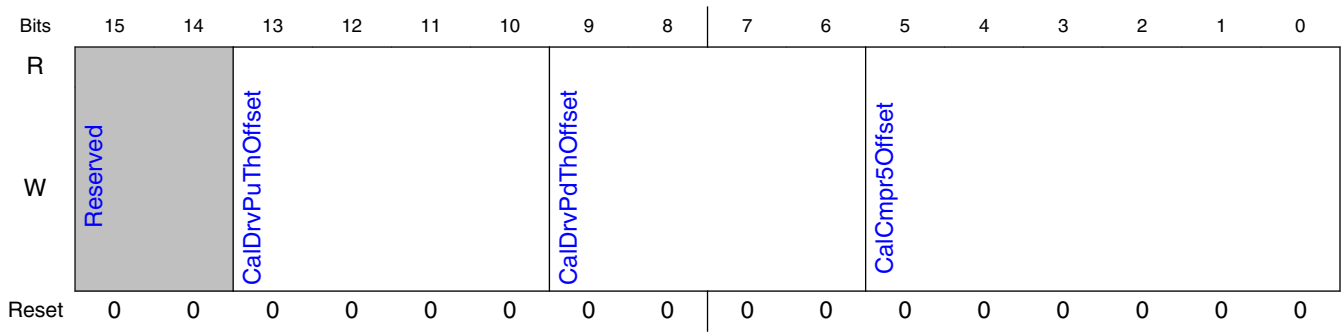
Field	Function
15-2 —	Reserved
1 DllRxPreambleMode	Must be set to 1 if read DQS preamble contains a toggle, for example DDR4 or LPDDR4 read toggling preambe mode
0 —	Reserved

9.4.3.6.39 Impedance Calibration offsets control (CalOffsets)

9.4.3.6.39.1 Offset

Register	Offset
CalOffsets	8Ah

9.4.3.6.39.2 Diagram



9.4.3.6.39.3 Fields

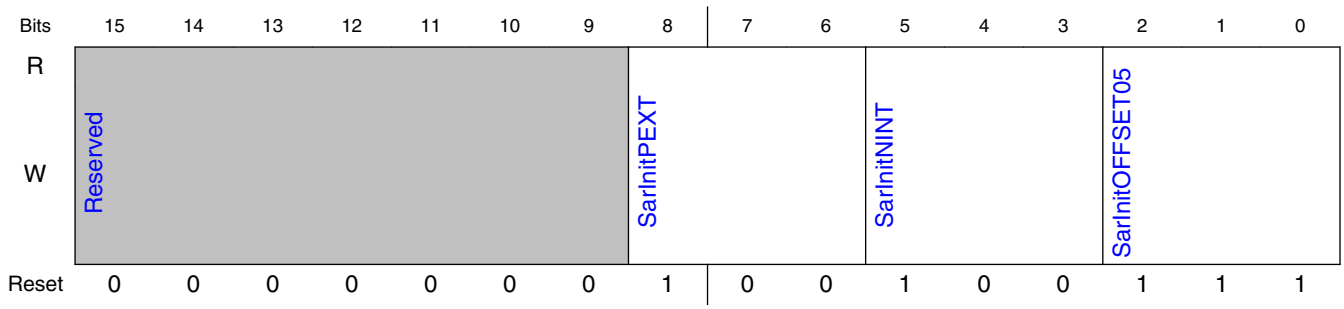
Field	Function
15-14 —	Reserved
13-10 CalDrvPuThOffset	This value adjusts the driver pullup calibration code
9-6 CalDrvPdThOffset	This value adjusts the driver pulldown calibration code
5-0 CalCmpr5Offset	This value adjusts the offset-compensated DAC code for the cmpana circuit at VRef == 0. This value adjusts the offset-compensated DAC code for the cmpana circuit at VRef == 0.5 * VDDQ.

9.4.3.6.40 Sar Init Vals (SarInitVals)

9.4.3.6.40.1 Offset

Register	Offset
SarInitVals	8Eh

9.4.3.6.40.2 Diagram



9.4.3.6.40.3 Fields

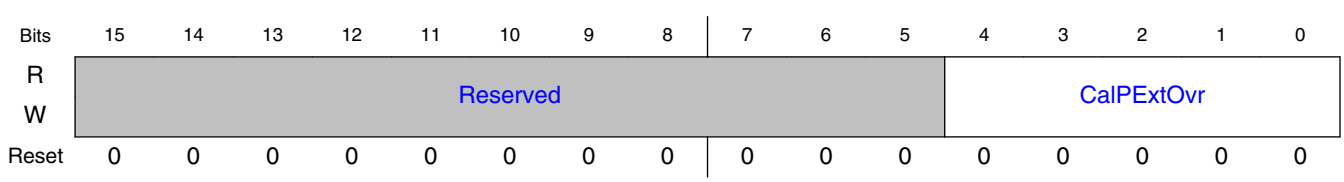
Field	Function
15-9 —	Reserved
8-6 SarInitPEXT	Specify the SAR starting value for PEXT calibration. Specify the SAR starting value for PEXT calibration. This is a 5 bit wide target.
5-3 SarInitNINT	Specify the SAR starting value for NINT calibration. Specify the SAR starting value for NINT calibration. This is a 5 bit wide target.
2-0 SarInitOFFSET05	Specify the SAR starting value for OFFSET05 calibration. Specify the SAR starting value for OFFSET05 calibration. This is an 8 bit wide target.

9.4.3.6.41 Impedance Calibration PExt Override control (CalPExtOvr)

9.4.3.6.41.1 Offset

Register	Offset
CalPExtOvr	92h

9.4.3.6.41.2 Diagram



9.4.3.6.41.3 Fields

Field	Function
15-5 —	Reserved
4-0 CalPExtOvr	If the CSR CalPExtDis is set then the value provided here by software will be used instead of the automatically generated value which is visible via CSR CalPExt This CSR may only be written when the calibrator is not running.

9.4.3.6.42 Impedance Calibration Cmpr 50 control (CalCmpr5Ovr)

9.4.3.6.42.1 Offset

Register	Offset
CalCmpr5Ovr	94h

9.4.3.6.42.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CalCmpr5Ovr							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.42.3 Fields

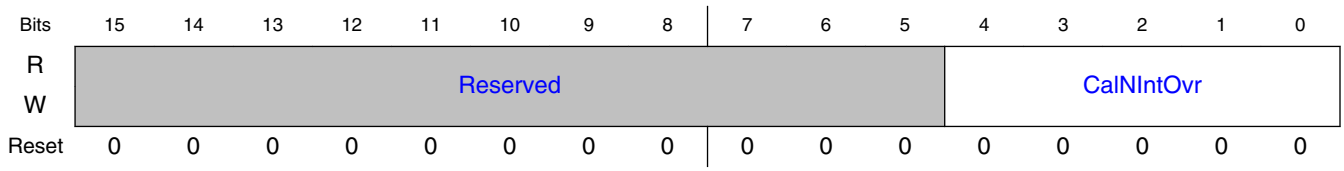
Field	Function
15-8 —	Reserved
7-0 CalCmpr5Ovr	If the CSR CalCmpr5Dis is set then the value provided here by software will be used instead of the automatically generated value which is visible via CSR CalCmpr5 This CSR may only be written when the calibrator is not running.

9.4.3.6.43 Impedance Calibration NInt Override control (CalNIntOvr)

9.4.3.6.43.1 Offset

Register	Offset
CalNIntOvr	96h

9.4.3.6.43.2 Diagram



9.4.3.6.43.3 Fields

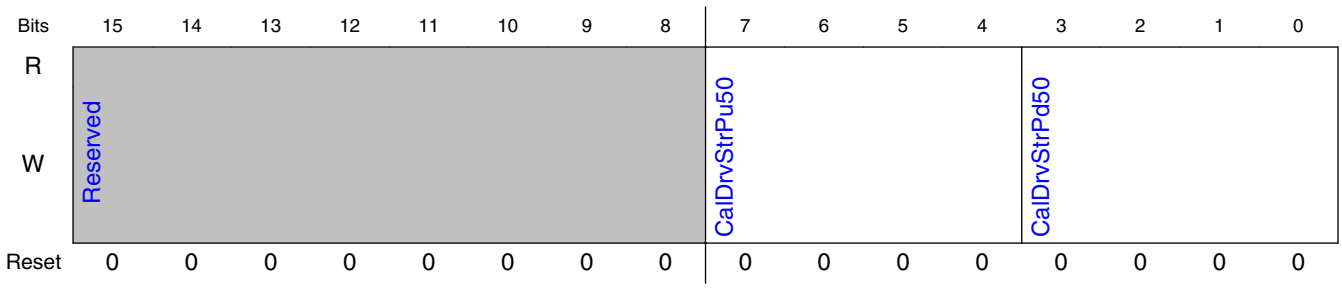
Field	Function
15-5 —	Reserved
4-0 CalNIntOvr	If the CSR CalNIntDis is set then the value provided here by software will be used instead of the automatically generated value which is visible via CSR CalNInt. If the CSR CalNIntDis is set then the value provided here by software will be used instead of the automatically generated value which is visible via CSR CalNInt. This CSR may only be written when the calibrator is not running.

9.4.3.6.44 Impedance Calibration driver strength control (CalDrvStr0)

9.4.3.6.44.1 Offset

Register	Offset
CalDrvStr0	A0h

9.4.3.6.44.2 Diagram



9.4.3.6.44.3 Fields

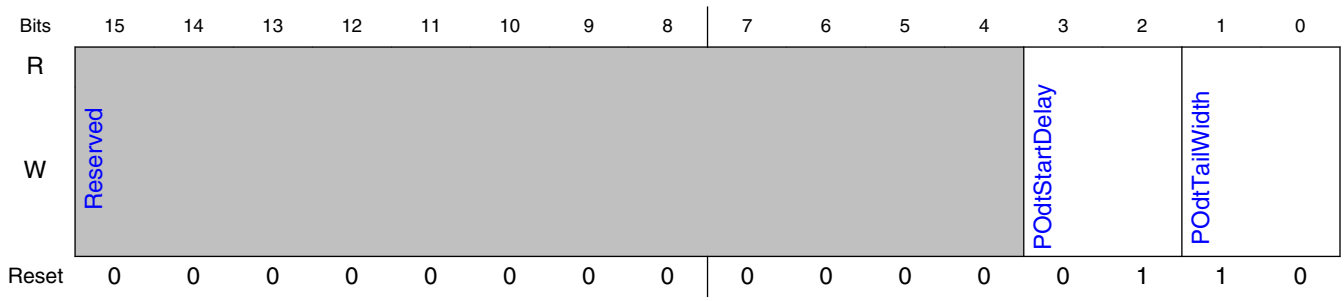
Field	Function
15-8 —	Reserved
7-4 CalDrvStrPu50	3. 3..15 = Reserved for future. 2 - 40.0R 1 - 120.0R 0 - 240.0R (Default)
3-0 CalDrvStrPd50	3. 3..15 = Reserved for future. 2 - 40.0R 1 - 120.0R 0 - 240.0R (Default)

9.4.3.6.45 READ DATA On-Die Termination Timing Control (by PHY) (Proc OdtTimeCtl_p0)

9.4.3.6.45.1 Offset

Register	Offset
ProcOdtTimeCtl_p0	ACh

9.4.3.6.45.2 Diagram



9.4.3.6.45.3 Fields

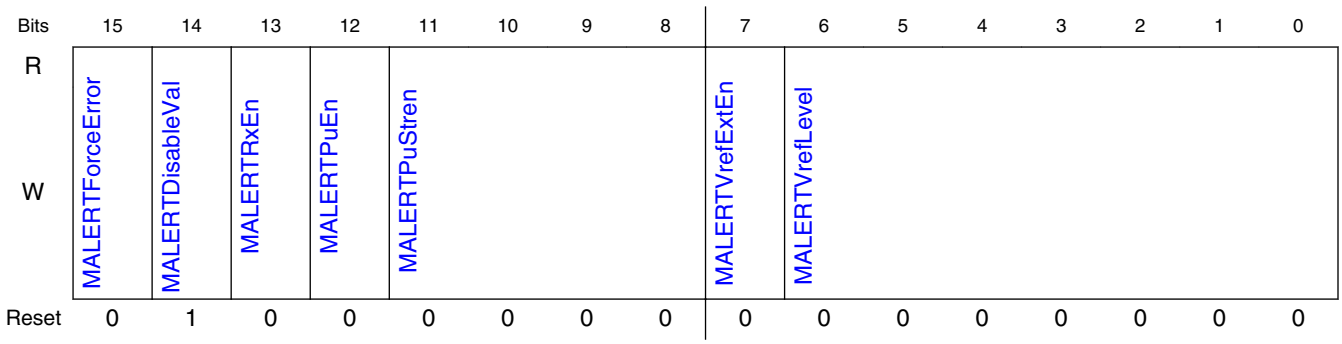
Field	Function
15-4 —	Reserved
3-2 POdtStartDelay	controls the start of ProcOdt, units of UI 3 delay start 2 UI, maximum delay of start of ProcOdt 2 delay start 1 UI, 1 delay start 0 UI, default 0 early by 1 UI, The time from ProcODT assertion to opening the window to receive DQS is (10 - POdtStartDelay) UI.
1-0 POdtTailWidth	controls the length of the tail of ProcOdt, units of UI 3 tail 3UI more than for Register POdtTailWidth=0, maximum 2 tail 2UI more than for Register POdtTailWidth=0, default 1 tail 1UI more than for Register POdtTailWidth=0 0 minimum length tail The time from ProcODT to closing the window to receive DQS to ProcODT POdtTailWidth is (2 + POdtTailWidth) UI

9.4.3.6.46 This Register is used to configure the MemAlert Receiver (MemAlertControl)

9.4.3.6.46.1 Offset

Register	Offset
MemAlertControl	B6h

9.4.3.6.46.2 Diagram



9.4.3.6.46.3 Fields

Field	Function
15 MALERTForceError	When MALERTForceError is set, this CSR state is used to force parity error to memory.
14 MALERTDisableVal	When MALERTRxEn is not set, this CSR state is used to drive dfi_alert_n.
13 MALERTRxEn	1 - Enables receiver and received data is forwared to dfi_alert_n. 1 - Enables receiver and received data is forwared to dfi_alert_n. 0 - Disable the receiver and send MALERTDisableVal to dfi_alert_n
12 MALERTPuEn	When set, enables the Pull-up termination on MALERT
11-8 MALERTPuStren	Controls the Pull-up termination on MALERT ========== bit[8] - controls a 240 Ohm Pull-up leg bit[9] - controls a 240 Ohm Pull-up leg bit[10] - controls a 120 Ohm Pull-up leg bit[11] - controls a 120 Ohm Pull-up leg ========== 0000 - No PullUp Strength 0001 - 240 Ohm PullUp Strength 0010 - 240 Ohm PullUp Strength 0011 - 120 Ohm PullUp Strength 0100 - 120 Ohm PullUp Strength 0101 - 80 Ohm PullUp Strength 0110 - 80 Ohm PullUp Strength 0111 - 60 Ohm PullUp Strength 1000 - 120 Ohm PullUp Strength 1001 - 80 Ohm PullUp Strength 1010 - 80 Ohm PullUp Strength 1011 - 60 Ohm PullUp Strength 1100 - 60 Ohm PullUp Strength 1101 - 48 Ohm PullUp Strength 1110 - 48 Ohm PullUp Strength 1111 - 40 Ohm PullUp Strength
7 MALERTVrefExtEn	When set for test/debug, selects external Vref source, This should not be set in mission mode.
6-0 MALERTVrefLevel	Sets the vref level of internal VREF DAC. Sets the vref level of internal VREF DAC. Assuming the following Mission mode settings: TestExtVrefRange,TestMajorMode = 0011 or 0000) The VREF level chosen by the receiver follows the equation:: VREF = VDDQ*(0.51 + MALERTVrefLevel[6:0]*0.00345)

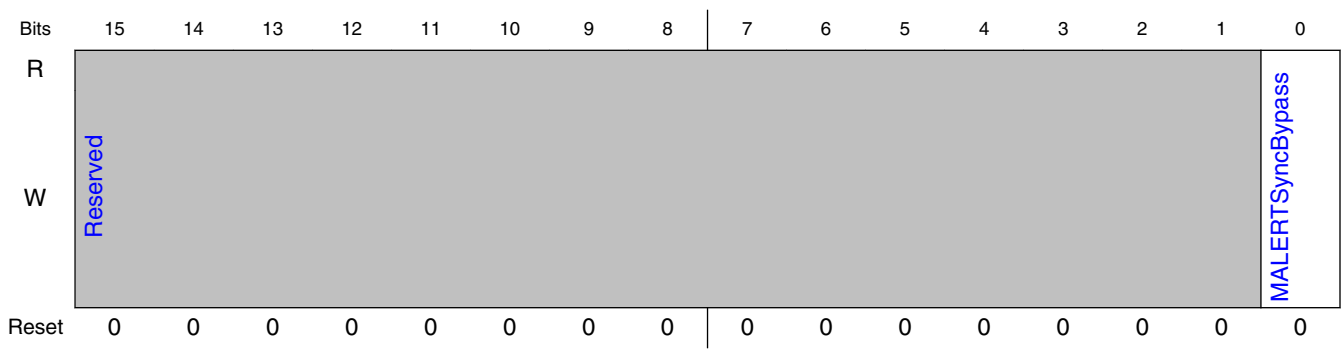
Field	Function
	The min/max values possible are: VREF_min = VDDQ*(0.51) ; VREF_max = VDDQ*(0.948)

9.4.3.6.47 This Register is used to configure the MemAlert Receiver (MemAlertControl2)

9.4.3.6.47.1 Offset

Register	Offset
MemAlertControl2	B8h

9.4.3.6.47.2 Diagram



9.4.3.6.47.3 Fields

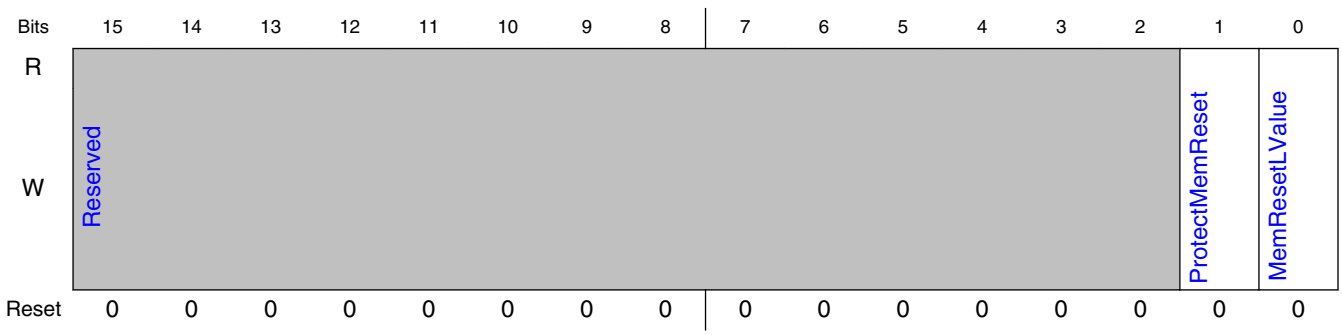
Field	Function
15-1 —	Reserved
0 MALERTSyncBypass	MALERTSyncBypass==[0], the phy will drive dfi_alert_n with a synchronized value of the ALERT_N receiver. MALERTSyncBypass==[0], the phy will drive dfi_alert_n with a synchronized value of the ALERT_N receiver. MALERTSyncBypass==[1], the phy will drive dfi_alert_n with the asynchronous value of the ALERT_N receiver.

9.4.3.6.48 Protection and control of BP_MemReset_L (MemResetL)

9.4.3.6.48.1 Offset

Register	Offset
MemResetL	C0h

9.4.3.6.48.2 Diagram



9.4.3.6.48.3 Fields

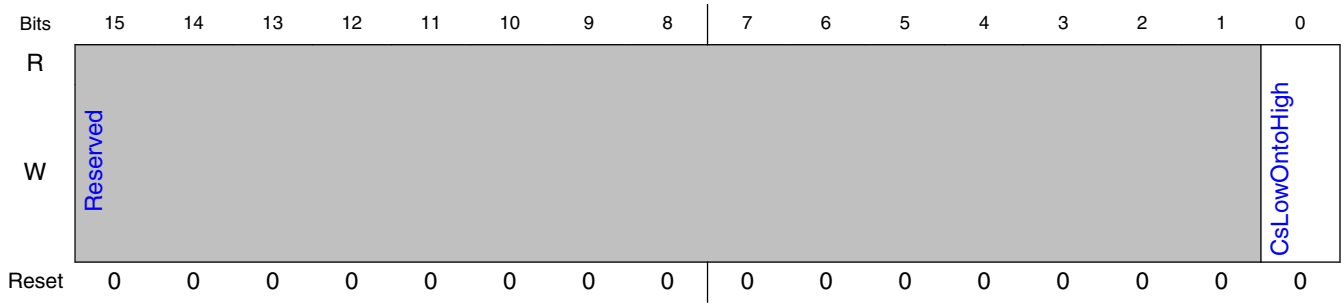
Field	Function
15-2 —	Reserved
1 ProtectMemReset	Control the MemResetL output of the PHY. Control the MemResetL output of the PHY. The state of the MemResetL field is transfered to the PHY BP_MemReset_L output when ProtectMemReset is set.
0 MemResetLValue	Control the MemResetL output of the PHY. Control the MemResetL output of the PHY. The state of this field is transfered to the PHY BP_MemReset_L output when ProtectMemReset is set. BP_MemReset_L cannot reset LPDDR3.

9.4.3.6.49 Drive CS_N 3:0 onto CS_N 7:4 (DriveCSLowOntoHigh)

9.4.3.6.49.1 Offset

Register	Offset
DriveCSLowOntoHigh	DAh

9.4.3.6.49.2 Diagram



9.4.3.6.49.3 Fields

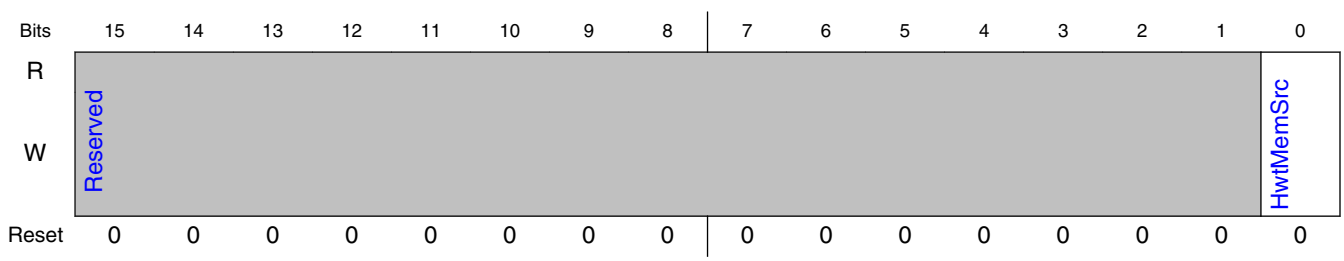
Field	Function
15-1 —	Reserved
0 CsLowOntoHigh	When this is set to a 1, CS[3:0] from the ACSM are driven to CS[7:4] pins and CS[3:0] are deasserted. When this is set to a 1, CS[3:0] from the ACSM are driven to CS[7:4] pins and CS[3:0] are deasserted. When this is set to a 0, CS[3:0] from the ACSM are driven to CS[3:0] pins and CS[7:4] are deasserted. This is used only in DDR4 DirectQuadCS ACX13 mode

9.4.3.6.50 PUBMODE - HWT Mux Select (PUBMODE)

9.4.3.6.50.1 Offset

Register	Offset
PUBMODE	DCh

9.4.3.6.50.2 Diagram



9.4.3.6.50.3 Fields

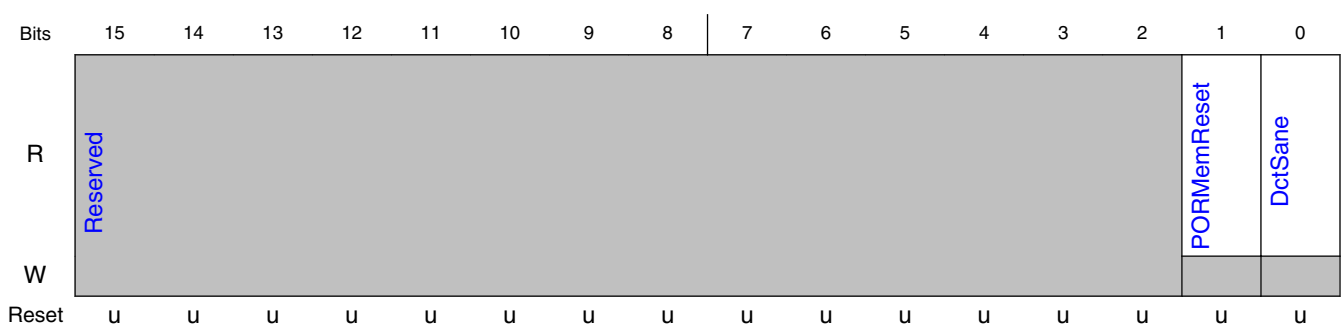
Field	Function
15-1 —	Reserved
0 HwtMemSrc	When this is set to a 1, the mux that switches between DCT and HWT for the source of memory transactions is switched to HWT. When this is set to a 1, the mux that switches between DCT and HWT for the source of memory transactions is switched to HWT. Setting this bit relinquishes control to the HWT. Clearing this bit relinquishes control to the DCT. The transition can only be made when the FIFOs are all in PtrInit (i.e. all PtrInit signals are high).

9.4.3.6.51 Misc PHY status bits (MiscPhyStatus)

9.4.3.6.51.1 Offset

Register	Offset
MiscPhyStatus	DEh

9.4.3.6.51.2 Diagram



9.4.3.6.51.3 Fields

Field	Function
15-2	Reserved

Table continues on the next page...

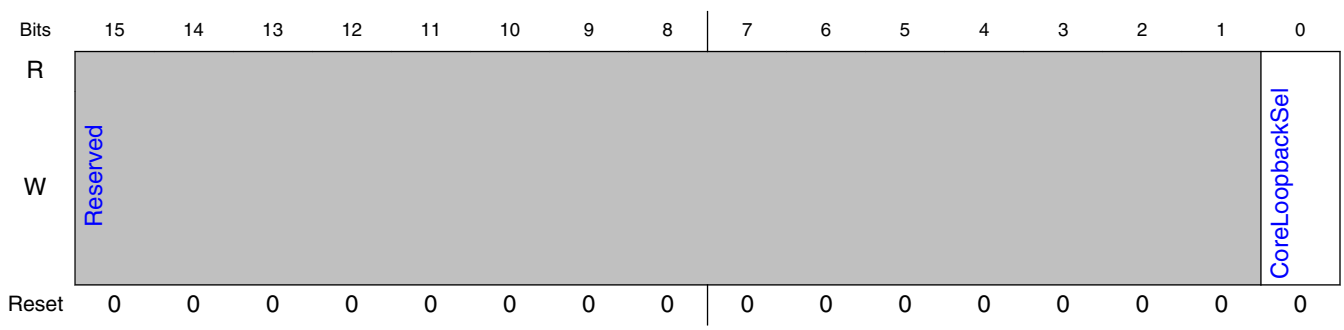
Field	Function
—	
1 PORMemReset	Returns the active-high value used by the custom circuit which drives the memory RESET signal. Returns the active-high value used by the custom circuit which drives the memory RESET signal. To comply with JEDEC specifications for memory reset, this signal powers on 1'b1. This signal is reset during the PHY Initialization sequence.
0 DctSane	Returns the status of the custom circuit which protects the MemResetL output of the PHY on initial power-on or reset. Returns the status of the custom circuit which protects the MemResetL output of the PHY on initial power-on or reset. This circuit resets to 0 meaning that hardware has control of the MemResetL output. If set to 1, memory RESET is being controlled by Registers MemResetLVaue and ProtectMemReset or dfi_reset_n.

9.4.3.6.52 Controls whether the loopback path bypasses the final PAD node. (CoreLoopbackSel)

9.4.3.6.52.1 Offset

Register	Offset
CoreLoopbackSel	E0h

9.4.3.6.52.2 Diagram



9.4.3.6.52.3 Fields

Field	Function
15-1	Reserved

Table continues on the next page...

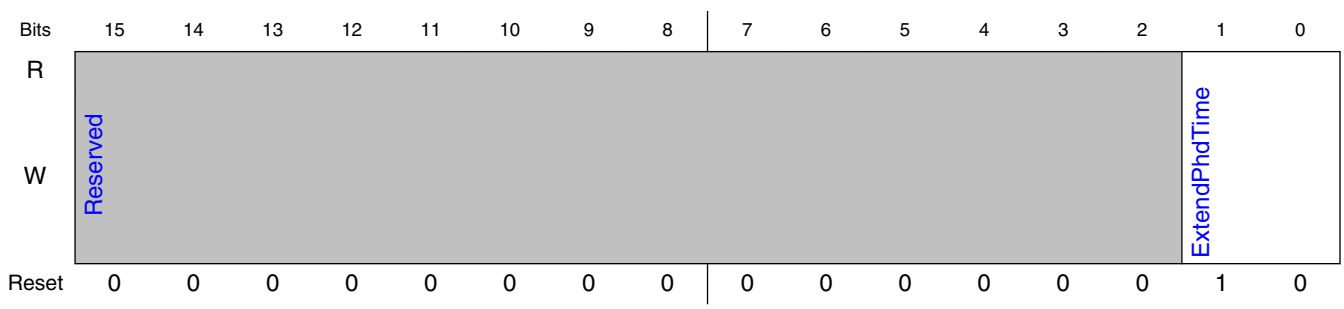
Field	Function
—	
0 CoreLoopbackSel	This register is controlled by the PHY test firmware This register enables Core-Side loopback operation of the PHY. This register is controlled by the PHY test firmware This register enables Core-Side loopback operation of the PHY. When set, the PHY PAD node will be bypassed, and pre-pad node will be selected for sampling by the PHY RX. Setting this register enables full-speed loopback operation of the PHY regardless of the loading of the pad. This register must be set to 1'b0 for mission mode operation (or pad-side loopback).

9.4.3.6.53 DLL Various Training Parameters (DllTrainParam)

9.4.3.6.53.1 Offset

Register	Offset
DllTrainParam	E2h

9.4.3.6.53.2 Diagram



9.4.3.6.53.3 Fields

Field	Function
15-2 —	Reserved
1-0 ExtendPhdTime	Used by the PHY firmware locking the LCDL delay cells. Used by the PHY firmware locking the LCDL delay cells. The number of extra DfiClk periods (= 4*UI) for the DLL phase-detector outputs to become not-metastable

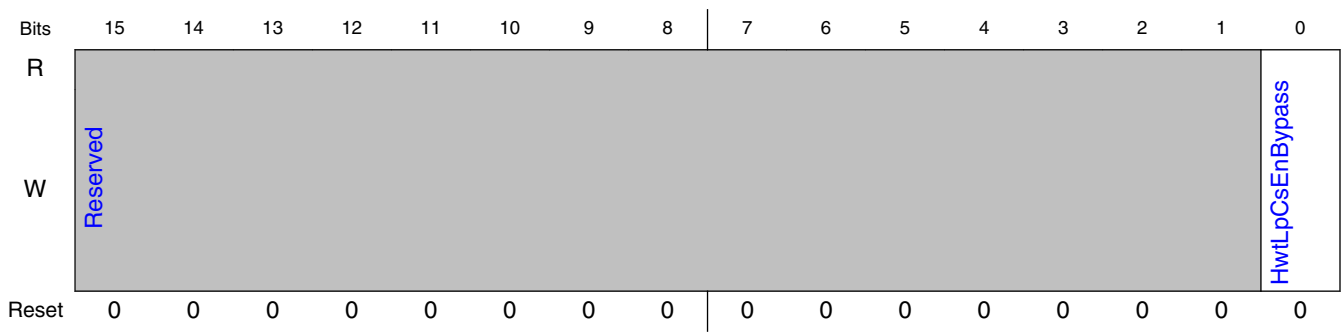
Field	Function
	Recommend to set to 3 if mem CK frequency ge 1800MHz Recommend to set to 2 if mem CK frequency lt 1800MHz and ge 1600MHz Recommend to set to 1 if mem CK frequency lt 1600MHz and ge 1000MHz Recommend to set to 0 if mem CK frequency lt 1000MHz

9.4.3.6.54 CSn Disable Bypass for LPDDR3/4 (HwtLpCsEnBypass)

9.4.3.6.54.1 Offset

Register	Offset
HwtLpCsEnBypass	E8h

9.4.3.6.54.2 Diagram



9.4.3.6.54.3 Fields

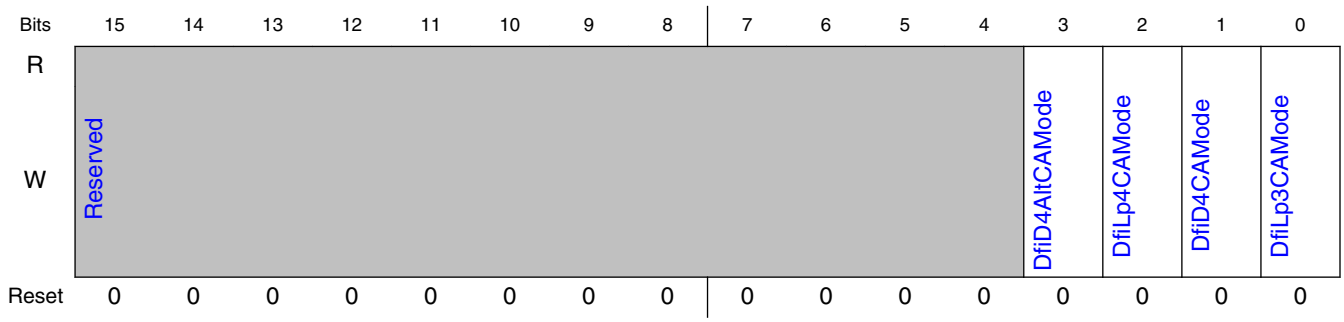
Field	Function
15-1 —	Reserved
0 HwtLpCsEnBypass	When set, these bits disable LpCsEn function for LPDDR3/4

9.4.3.6.55 Dfi Command/Address Mode (DfiCAMode)

9.4.3.6.55.1 Offset

Register	Offset
DfiCAMode	EAh

9.4.3.6.55.2 Diagram



9.4.3.6.55.3 Fields

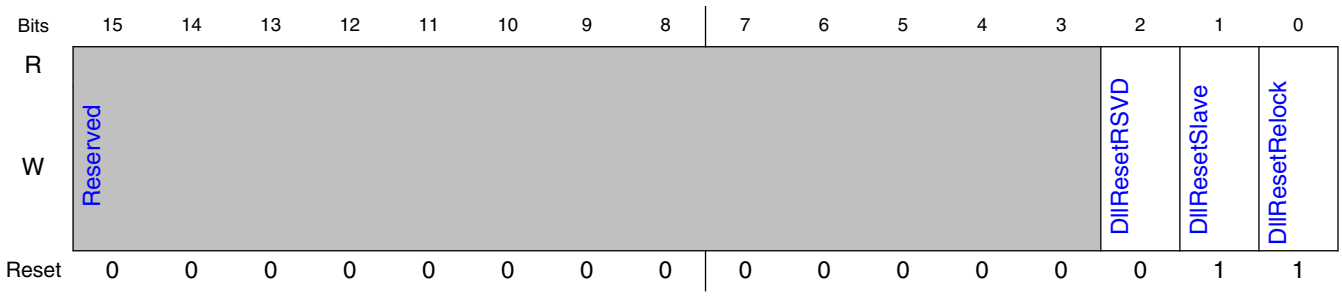
Field	Function
15-4 —	Reserved
3 DfiD4AltCAMode	Enable D4-Alt Mode 0: D4-Altmode disabled 1: D4-Altmode enabled
2 DfiLp4CAMode	Enable LP4 Mode 0: LP4 mode disabled 1: LP4 mode enabled
1 DfiD4CAMode	Enable D4 Mode 0: D4 mode disabled 1: D4 mode enabled
0 DfiLp3CAMode	Controls the output data-rate of the AC module Command/Address pins 0: LP3 DDR address mode disabled 1: LP3 DDR address mode enabled

9.4.3.6.56 DLL Lock State machine control register (DllControl)

9.4.3.6.56.1 Offset

Register	Offset
DllControl	F0h

9.4.3.6.56.2 Diagram



9.4.3.6.56.3 Fields

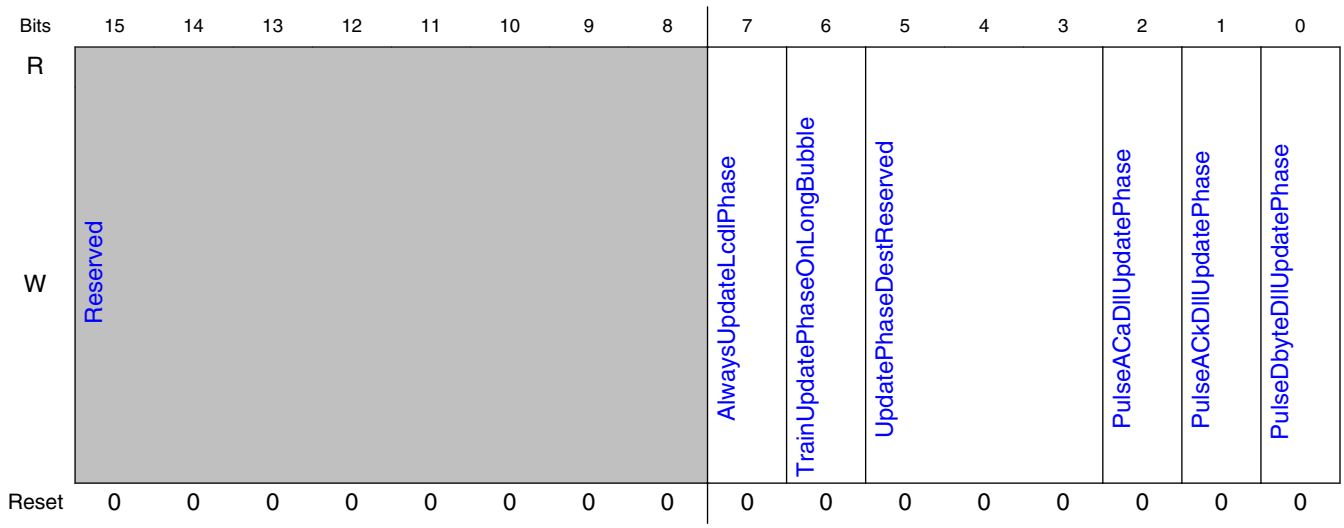
Field	Function
15-3 —	Reserved
2 DllResetRSVD	RSVD for future use
1 DllResetSlave	Reserved
0 DllResetRelock	Used to reset the DDL/LCDL lock state machine Deasserting starts locking sequence. Used to reset the DDL/LCDL lock state machine Deasserting starts locking sequence. This must be done anytime the DLLs are to be relocked (e.g. frequency change).

9.4.3.6.57 DLL update phase control (PulseDllUpdatePhase)

9.4.3.6.57.1 Offset

Register	Offset
PulseDllUpdatePhase	F2h

9.4.3.6.57.2 Diagram



9.4.3.6.57.3 Fields

Field	Function
15-8 —	Reserved
7 AlwaysUpdateLcdlPhase	Causes each new operation to reload the LcdlPhase; will increase bubbles.
6 TrainUpdatePhaseOnLongBubble	Causes LongBubble to update the dbyte & anib LDCL Phase. Causes LongBubble to update the dbyte & anib LDCL Phase. May be used for training, but not to be used for mission mode.
5-3 UpdatePhaseDestReserved	reserved, not used
2 PulseACaDllUpdatePhase	Causes an AC module CA (command/address/cke/odt) DLL phase update.
1 PulseACkDllUpdatePhase	Causes an AC module CK (memck) DLL phase update. Causes an AC module CK (memck) DLL phase update. This should be written with 1 only when the memory CKE=0 or memory RST=0, or when there is no memory, as in tester mode.
0 PulseDbyteDllUpdatePhase	Causes a LongBubble to the DBYTE modules, which causes a update of the DBYTE module DLLs (tx,rxen,rxclk). Causes a LongBubble to the DBYTE modules, which causes a update of the DBYTE module DLLs (tx,rxen,rxclk). The value of Register UpdatePhaseDest must be setup prior to this pulse.

9.4.3.6.58 DLL gain control (DllGainCtl_p0)

9.4.3.6.58.1 Offset

Register	Offset
DllGainCtl_p0	F8h

9.4.3.6.58.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DllSeedSel				DllGainTV				DllGainIV			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1

9.4.3.6.58.3 Fields

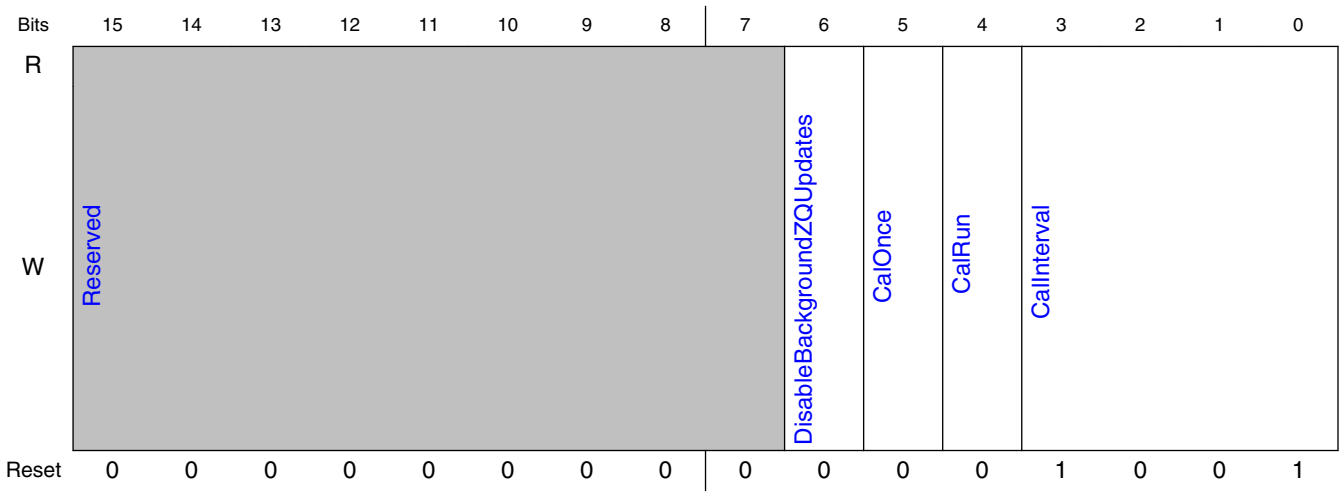
Field	Function
15-12 —	Reserved
11-8 DllSeedSel	Reserved, must be configured to be 0.
7-4 DllGainTV	Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. DllGainTV must be greater than or equal to DllGainIV. The maximum value is 10. The minimum value is 6.
3-0 DllGainIV	Initial value of DllGain.

9.4.3.6.59 Impedance Calibration Control (CalRate)

9.4.3.6.59.1 Offset

Register	Offset
CalRate	110h

9.4.3.6.59.2 Diagram



9.4.3.6.59.3 Fields

Field	Function
15-7 —	Reserved
6 DisableBackgroundZQUpdates	1: Instead of having the driver compensation codes go asynchronously out to all IO, hold until for any of PHYUPD ACK, CTRLUPD ACK, PHYMSTR ACK) 0: Calibrated ZQ Updates to IO aren't gated. 1: Instead of having the driver compensation codes go asynchronously out to all IO, hold until for any of PHYUPD ACK, CTRLUPD ACK, PHYMSTR ACK) 0: Calibrated ZQ Updates to IO aren't gated. The default value of this field is 0 because the minor Impedance corrections (Converted from binary to thermometer) is applied over multiple cycles and the impact caused is minor.
5 CalOnce	The setting of this CSR changes the behaviour of CSR CalRun. The setting of this CSR changes the behaviour of CSR CalRun. 1: The 0->1 transition of CSR CalRun causes a single iteration of the calibration sequence to occur. Once the calibration is complete, the calibration engine will remain IDLE until another CalRun Trigger. 0: Calibration will proceed at the rate determined by Register CalInterval.

Table continues on the next page...

Field	Function
	This field should only be changed while the calibrator is idle. ie before csr CalRun is set.
4 CalRun	1: A calibration sequence will be triggered by the 0->1 transition of this bit, as determined by CSR CalOnce. 1: A calibration sequence will be triggered by the 0->1 transition of this bit, as determined by CSR CalOnce. 0: Calibrator will not run. If already active then it will complete the current sequence before quiescing. The default value of this field is 0 because the calibrator needs to know whether it is in D3 mode before it starts up. This csr won't do anything while CalZap is asserted.
3-0 CalInterval	This CSR specifies the interval between successive calibrations, in mS. This CSR specifies the interval between successive calibrations, in mS. csrValue : Interval (mS) 0 : 0 (continuous) 1 : 0.013 2 : 0.10 3 : 1 4 : 2 5 : 3 6 : 4 7 : 8 8 : 10 9 : 20 10-15 : Reserved. This field should only be changed while the calibrator is idle. ie before csr CalRun is set.

9.4.3.6.60 Impedance Calibration Zap/Reset (CalZap)

9.4.3.6.60.1 Offset

Register	Offset
CalZap	112h

9.4.3.6.60.2 Diagram

9.4.3.6.60.3 Fields

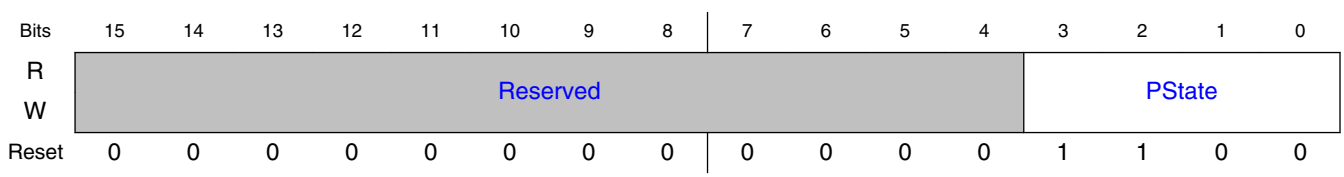
Field	Function
15-1 —	Reserved
0 CalZap	NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. Setting this csr resets the calibrator to its idle state. Must be cleared in order for the calibration engine to run again.

9.4.3.6.61 PSTATE Selection (PState)

9.4.3.6.61.1 Offset

Register	Offset
PState	116h

9.4.3.6.61.2 Diagram



9.4.3.6.61.3 Fields

Field	Function
15-4	Reserved

Table continues on the next page...

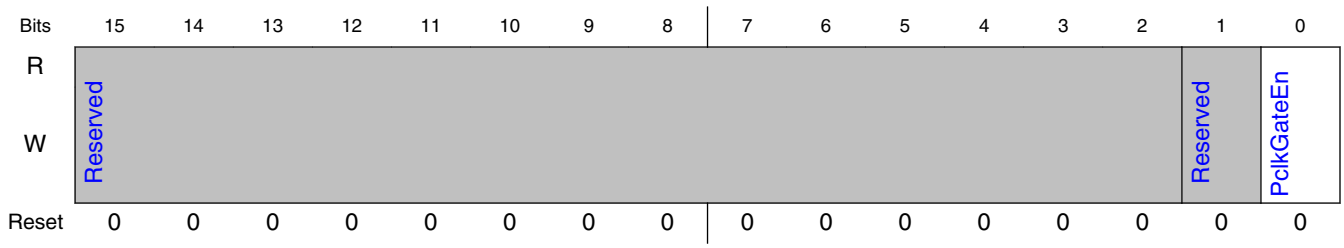
Field	Function
—	
3-0 PState	NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. The current PState inferred by the PIE from handshake requests on DFI status interface. 0000 - PState P0 0001 - PState P1 0010 - PState P2 0011 - PState P3 0100 - Reserved 0101 - Reserved 0110 - Reserved 0111 - Reserved 1000 - Reserved 1001 - Reserved 1010 - Reserved 1011 - Reserved 1100 - LP2 1101 - Reserved 1110 - Reserved 1111 - LP3 The lower two bits of this CSR are used to select the PHY's current Pstate for all Pstateable registers. Note: It is a don't care what value is selected in either LP2 or LP3 because no transactions will be taking place when in those states.

9.4.3.6.62 PLL Output Control (PllOutGateControl)

9.4.3.6.62.1 Offset

Register	Offset
PllOutGateControl	11Ah

9.4.3.6.62.2 Diagram



9.4.3.6.62.3 Fields

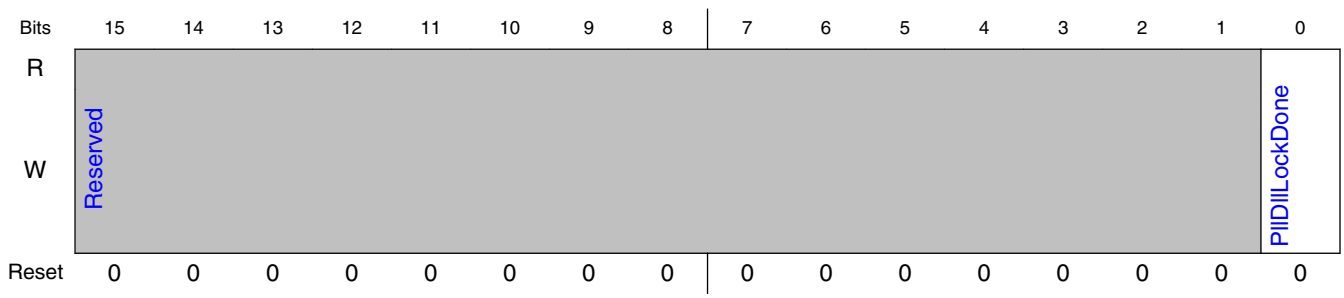
Field	Function
15-2 —	Reserved
1 —	Reserved for future.
0 PclkGateEn	Reserved

9.4.3.6.63 PMU Power-on Reset Control (PLL/DLL Lock Done) (PorControl)

9.4.3.6.63.1 Offset

Register	Offset
PorControl	120h

9.4.3.6.63.2 Diagram



9.4.3.6.63.3 Fields

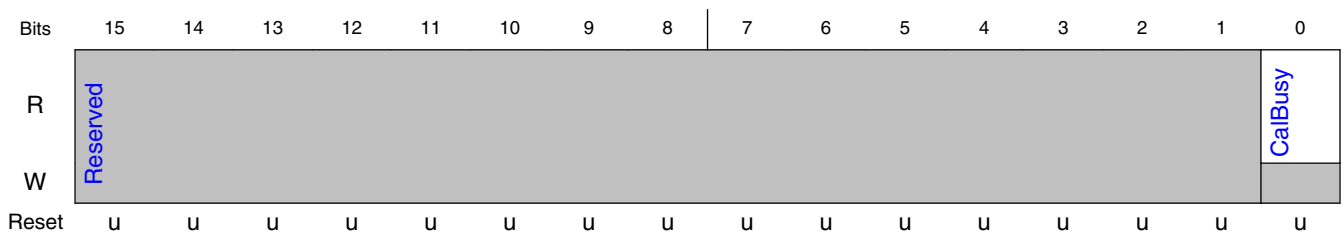
Field	Function
15-1 —	Reserved
0 PllDllLockDone	Set by the PIE to 1 after it has finished the PLL/DLL lock sequence. Set by the PIE to 1 after it has finished the PLL/DLL lock sequence. It is only cleared to 0 when PHY is Reset or exiting LP3 state.

9.4.3.6.64 Impedance Calibration Busy Status (CalBusy)

9.4.3.6.64.1 Offset

Register	Offset
CalBusy	12Eh

9.4.3.6.64.2 Diagram



9.4.3.6.64.3 Fields

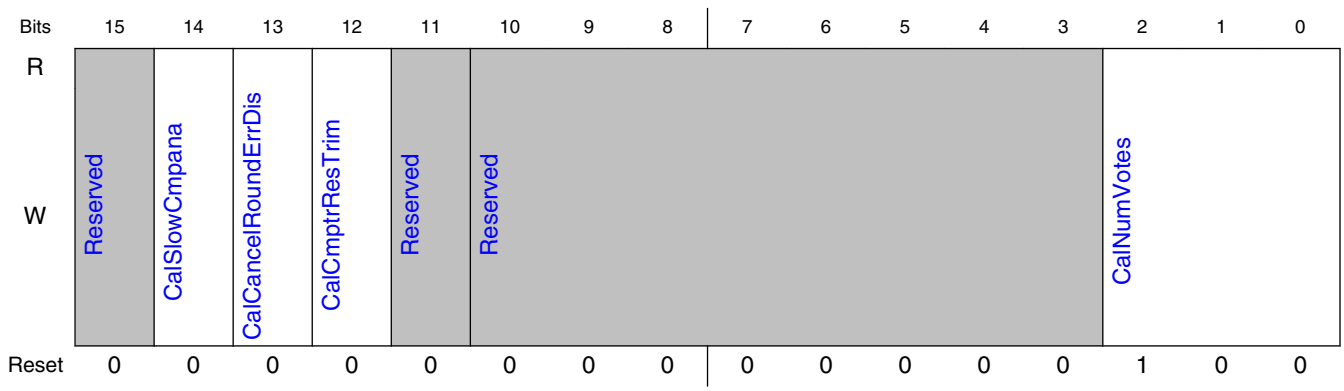
Field	Function
15-1 —	Reserved
0 CalBusy	Read 1 if the calibrator is actively calibrating. Read 1 if the calibrator is actively calibrating. Any changes to calibrator-related CSRs may only be made if the calibrator is disabled (via CSR CalRun) and this CSR reads 0.

9.4.3.6.65 Miscellaneous impedance calibration controls. (CalMisc2)

9.4.3.6.65.1 Offset

Register	Offset
CalMisc2	130h

9.4.3.6.65.2 Diagram



9.4.3.6.65.3 Fields

Field	Function
15 —	Miscellaneous impedance calibration controls.
14 CalSlowCmpana	When set, this CSR increases the time allowed for the cmpana cell to settle, by 50%. When set, this CSR increases the time allowed for the cmpana cell to settle, by 50%. It changes from 3 to 6 refClk ticks.
13 CalCancelRoundErrDis	The PEXT calibration result and NINT calibration results naturally include a rounding error which manifests as a change of impedance at the pad. The PEXT calibration result and NINT calibration results naturally include a rounding error which manifests as a change of impedance at the pad. Where both of these errors are in the same direction the PEXT calibration result will be adjusted by 1 to offset NINT error. Setting thie csr to 1 prevents this adjustment.
12 CalCmptrResTrim	Reserved for future use.

Table continues on the next page...

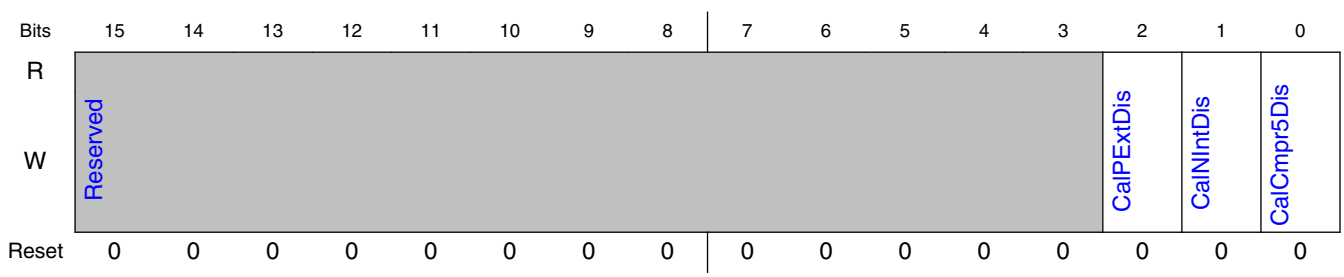
Field	Function
11 — 	Reserved for future use. Reserved for future use. Formerly used to Skip ODT calibration if required.
10-3 —	Reserved for future use.
2-0 CalNumVotes	This CSR controls the number of consecutive comparator output bits over which majority voting is done. This CSR controls the number of consecutive comparator output bits over which majority voting is done. The result of the voting is fed into the SAR. The voting is initiated after the time configured via csrs CalSlowCmpana has elapsed. CSR Value : number of Votes 0 : 1 1 : 9 2 : 17 3 : 33 4 : 65 (defaults) 5 : 129 6 : 193 7 : 255

9.4.3.6.66 Controls for disabling the impedance calibration of certain targets. (CalMisc)

9.4.3.6.66.1 Offset

Register	Offset
CalMisc	134h

9.4.3.6.66.2 Diagram



9.4.3.6.66.3 Fields

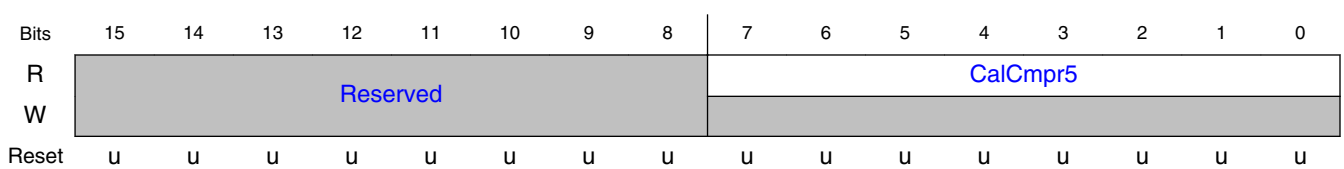
Field	Function
15-3 —	Reserved
2 CalPExtDis	Setting this CSR prevents the calibration engine from overwriting the CSRs TxCalBinP and TxCalThP with an automatically generated value, in which case a value must be supplied by software.
1 CalNIntDis	Setting this CSR prevents the calibration engine from overwriting the CSRs TxCalBinN and TxCalThN with an automatically generated value, in which case a value must be supplied by software.
0 CalCmpr5Dis	Setting this CSR prevents the calibration engine from using the result from the CalCmpr5 stage of calibration. Setting this CSR prevents the calibration engine from using the result from the CalCmpr5 stage of calibration. Software must supply a replacement value via CSR CalCmpr5Ovr.

9.4.3.6.67 Impedance Calibration Cmpr control (CalCmpr5)

9.4.3.6.67.1 Offset

Register	Offset
CalCmpr5	138h

9.4.3.6.67.2 Diagram



9.4.3.6.67.3 Fields

Field	Function
15-8 —	Reserved
7-0 CalCmpr5	Returns the offset-compensated DAC code for the cmpana circuit at VRef == 0. Returns the offset-compensated DAC code for the cmpana circuit at VRef == 0.5 * VDDQ.

9.4.3.6.68 Impedance Calibration NInt control (CalNInt)

9.4.3.6.68.1 Offset

Register	Offset
CalNInt	13Ah

9.4.3.6.68.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											CalNIntThB				
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.6.68.3 Fields

Field	Function
15-5 —	Reserved
4-0 CalNIntThB	The value here is the number of thermometer bits which are set.

9.4.3.6.69 Impedance Calibration PExt control (CalPExt)

9.4.3.6.69.1 Offset

Register	Offset
CalPExt	13Ch

9.4.3.6.69.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved											CalPExtThB				
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.6.69.3 Fields

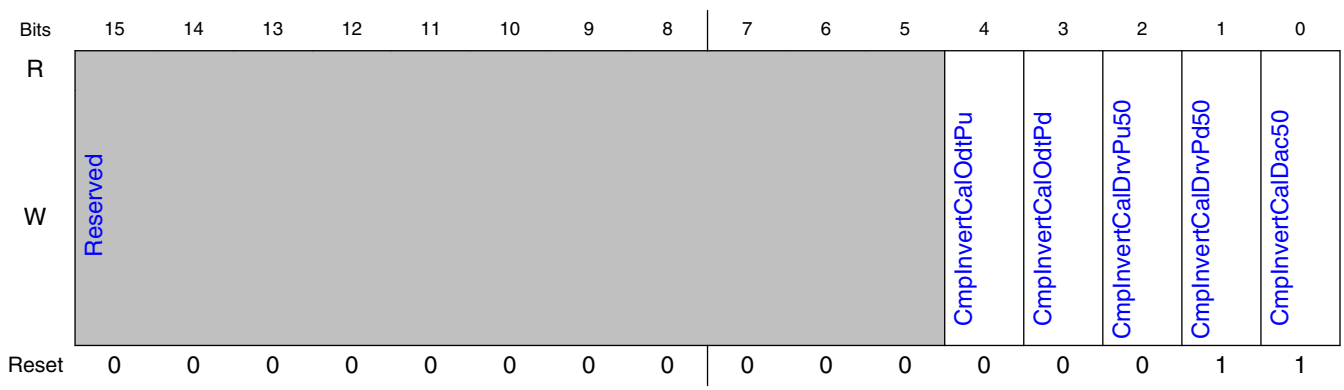
Field	Function
15-5 —	Reserved
4-0 CalPExtThB	The value here is the number of thermometer bits which are set.

9.4.3.6.70 Impedance Calibration Cmp Invert control (CalCmpInvert)

9.4.3.6.70.1 Offset

Register	Offset
CalCmpInvert	150h

9.4.3.6.70.2 Diagram



9.4.3.6.70.3 Fields

Field	Function
15-5 —	Reserved
4 CmpInvertCalOdtPu	Impedance Calibration Cmp Invert control
3	Impedance Calibration Cmp Invert control

Table continues on the next page...

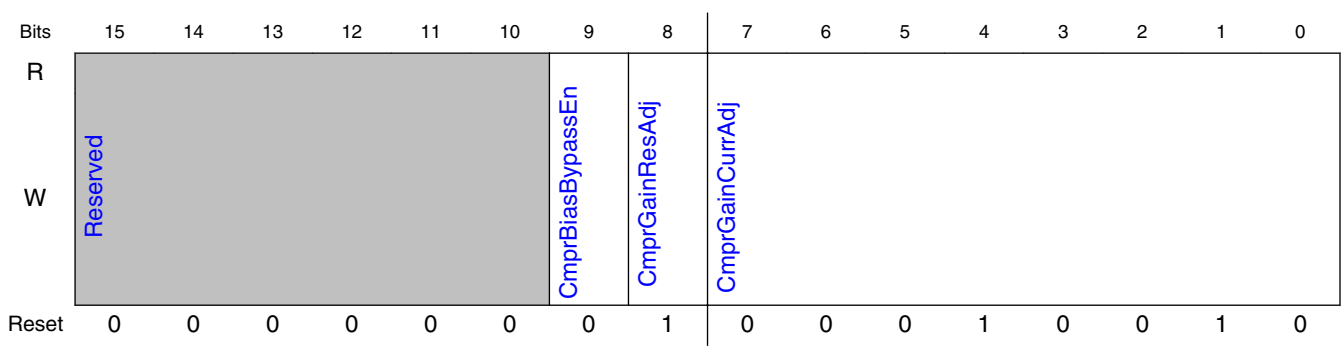
Field	Function
CmpInvertCalOdtPd	
2 CmpInvertCalDrvPu50	Impedance Calibration Cmp Invert control
1 CmpInvertCalDrvPd50	Impedance Calibration Cmp Invert control
0 CmpInvertCalDac50	Impedance Calibration Cmp Invert control

9.4.3.6.71 Impedance Calibration Cmpana control (CalCmpanaCntrl)

9.4.3.6.71.1 Offset

Register	Offset
CalCmpanaCntrl	15Ch

9.4.3.6.71.2 Diagram



9.4.3.6.71.3 Fields

Field	Function
15-10 —	Reserved
9	Impedance Calibration Cmpana control

Table continues on the next page...

Field	Function
CmprBiasBypassEn	
8 CmprGainResAdj	Impedance Calibration Cmpana control
7-0 CmprGainCurrAdj	Impedance Calibration Cmpana control

9.4.3.6.72 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p0)

9.4.3.6.72.1 Offset

Register	Offset
DfiRdDataCsDestMap_p0	160h

9.4.3.6.72.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								DfiRdDestm3		DfiRdDestm2		DfiRdDestm1		DfiRdDestm0	
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

9.4.3.6.72.3 Fields

Field	Function
15-8 —	Reserved
7-6 DfiRdDestm3	Maps dfi_rddata_cs_n_p0[3] to dest DfiRdDestm3 timing For example, if 3 dfi_rddata_cs_n_p0[3] will use Register RxEn,ClkDlyTg3 timing.
5-4 DfiRdDestm2	Maps dfi_rddata_cs_n_p0[2] to dest DfiRdDestm2 timing For example, if 2 dfi_rddata_cs_n_p0[2] will use Register RxEn,ClkDlyTg2 timing.

Table continues on the next page...

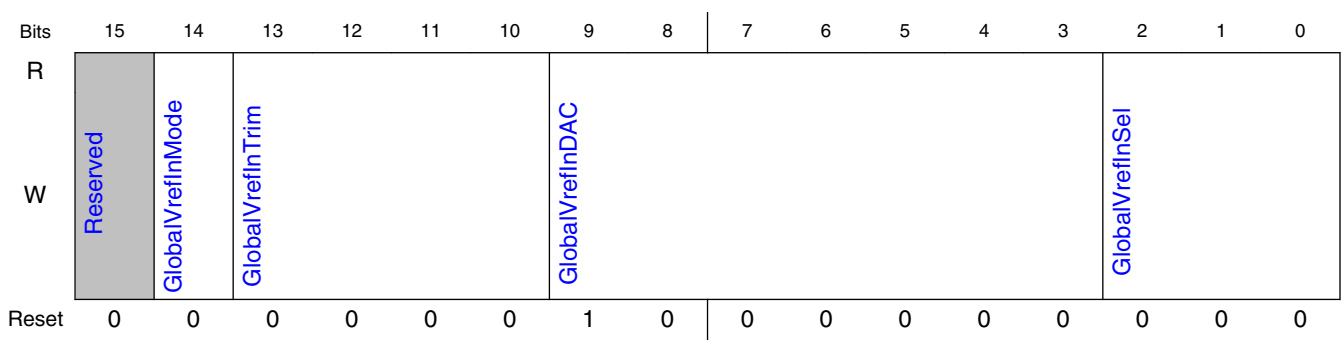
Field	Function
3-2 DfiRdDestm1	Maps dfi_rddata_cs_n_p0[1] to dest DfiRdDestm1 timing For example, if 1 dfi_rddata_cs_n[_p01] will use Register RxEn,ClkDlyTg1 timing.
1-0 DfiRdDestm0	Maps dfi_rddata_cs_n_p0[0] to dest DfiRdDestm0 timing For example, if 0 dfi_rddata_cs_n_p0[0] will use Register RxEn,ClkDlyTg0 timing.

9.4.3.6.73 PHY Global Vref Controls (VrefInGlobal_p0)

9.4.3.6.73.1 Offset

Register	Offset
VrefInGlobal_p0	164h

9.4.3.6.73.2 Diagram



9.4.3.6.73.3 Fields

Field	Function
15 —	Reserved
14 GlobalVrefInMode	RSVD
13-10 GlobalVrefInTrim	RSVD

Table continues on the next page...

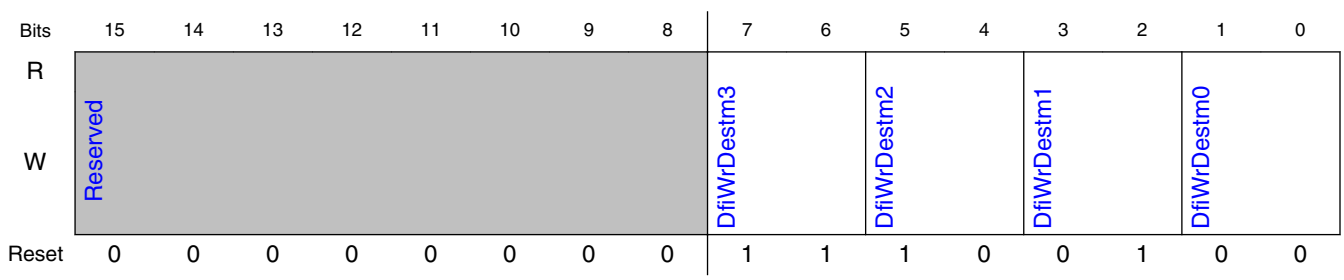
Field	Function
9-3 GlobalVrefInDAC	DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0. DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0.345*VDDQ + (0.005*GlobalVrefInDAC)*VDDQ ========== RANGE1 : LPDDR4 [GlobalVrefInSel[2] = 1] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : (0.005*(GlobalVrefInDAC-1))*VDDQ
2-0 GlobalVrefInSel	GlobalVrefInSel[1:0] controls the mode of the PHY VREF DAC and the BP_VREF pin ========== 2'b00 - PHY Vref DAC Range0 -- BP_VREF = Hi-Z 2'b01 - Reserved Encoding 2'b10 - PHY Vref DAC Range0 -- BP_VREF connected to PLL Analog Bus 2'b11 - PHY Vref DAC Range0 -- BP_VREF connected to PHY Vref DAC ========== GlobalVrefInSel[2] shall be set according to Dram Protocol: Protocol GlobalVrefInSel[2] ---------- DDR3 1'b0 DDR4 1'b0 LPDDR3 1'b0 LPDDR4 1'b1

9.4.3.6.74 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p0)

9.4.3.6.74.1 Offset

Register	Offset
DfiWrDataCsDestMap_p0	168h

9.4.3.6.74.2 Diagram



9.4.3.6.74.3 Fields

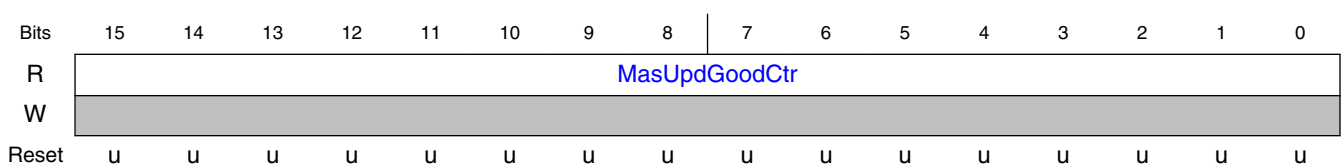
Field	Function
15-8 —	Reserved
7-6 DfiWrDestm3	Maps dfi_wrdata_cs_n_p0[3] to dest DfiWrDestm3 timing (use Register TxDq,DqsDlyTg3) For example, if 3 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg3 timing.
5-4 DfiWrDestm2	Maps dfi_wrdata_cs_n_p0[2] to dest DfiWrDestm2 timing For example, if 2 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg2 timing.
3-2 DfiWrDestm1	Maps dfi_wrdata_cs_n_p0[1] to dest DfiWrDestm1 timing For example, if 1 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg1 timing.
1-0 DfiWrDestm0	Maps dfi_wrdata_cs_n_p0[0] to dest DfiWrDestm0 timing For example, if 0 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg0 timing.

9.4.3.6.75 Counts successful PHY Master Interface Updates (PPTs) (MasUpdGoodCtr)

9.4.3.6.75.1 Offset

Register	Offset
MasUpdGoodCtr	16Ah

9.4.3.6.75.2 Diagram



9.4.3.6.75.3 Fields

Field	Function
15-0 MasUpdGoodCtr	This register increments whenever the Memory Controller acknowledges a PHY Master Interface request (i. This register increments whenever the Memory Controller acknowledges a PHY Master Interface request (i.e., a request for the PHY to take over the DFI for Periodic Phase Training (PPT)). (As per section 11 of the Preliminary DFI 4.0 Specification Rev. 2)

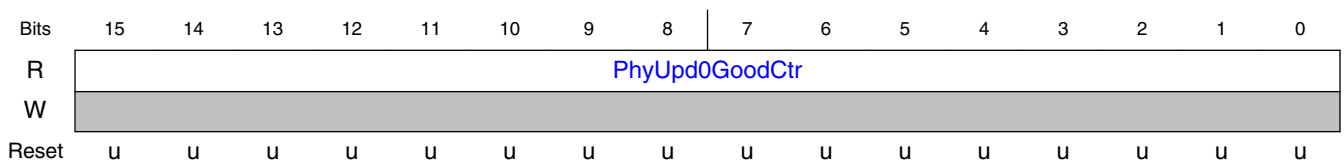
Field	Function
	In a 2-channel system, both MC channels must ack. The counter saturates at max value 65535.

9.4.3.6.76 Counts successful PHY-initiated DFI0 Interface Updates (PhyUpd0GoodCtr)

9.4.3.6.76.1 Offset

Register	Offset
PhyUpd0GoodCtr	16Ch

9.4.3.6.76.2 Diagram



9.4.3.6.76.3 Fields

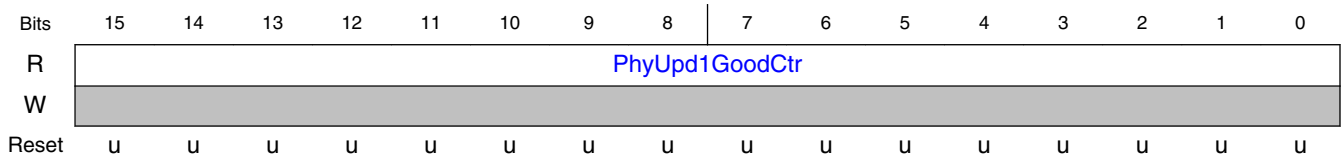
Field	Function
15-0 PhyUpd0GoodCtr	This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI0 interface update request. This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI0 interface update request. The counter saturates at max value 65535.

9.4.3.6.77 Counts successful PHY-initiated DFI1 Interface Updates (PhyUpd1GoodCtr)

9.4.3.6.77.1 Offset

Register	Offset
PhyUpd1GoodCtr	16Eh

9.4.3.6.77.2 Diagram



9.4.3.6.77.3 Fields

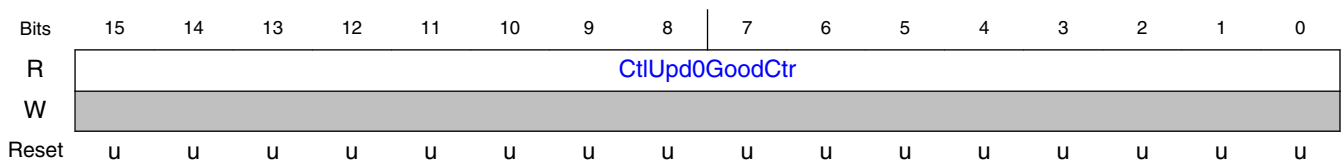
Field	Function
15-0 PhyUpd1GoodCtr	This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI1 interface update request. This register increments whenever the Memory Controller acknowledges a PHY-initiated DFI1 interface update request. The counter saturates at max value 65535.

9.4.3.6.78 Counts successful Memory Controller DFI0 Interface Updates (CtlUpd0GoodCtr)

9.4.3.6.78.1 Offset

Register	Offset
CtlUpd0GoodCtr	170h

9.4.3.6.78.2 Diagram



9.4.3.6.78.3 Fields

Field	Function
15-0 CtlUpd0GoodCtr	This register increments whenever the PHY acknowledges a Memory Controller-initiated DFI0 interface update request.

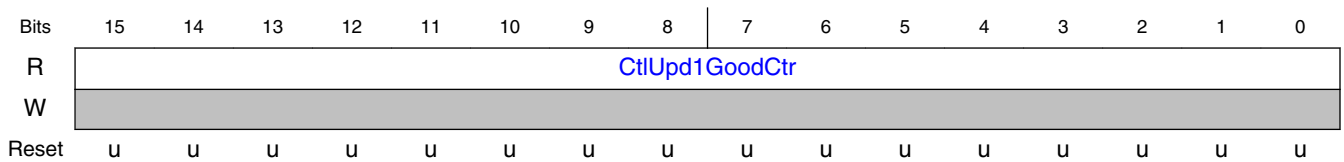
Field	Function
	This register increments whenever the PHY acknowledges a Memory Controller-initiated DFI0 interface update request. The counter saturates at max value 65535.

9.4.3.6.79 Counts successful Memory Controller DFI1 Interface Updates (CtlUpd1GoodCtr)

9.4.3.6.79.1 Offset

Register	Offset
CtlUpd1GoodCtr	172h

9.4.3.6.79.2 Diagram



9.4.3.6.79.3 Fields

Field	Function
15-0 CtlUpd1GoodCtr	This register increments whenever the PHY acknowledges a Memory Controller-initiated DFI1 interface update request. This register increments whenever the PHY acknowledges a Memory Controller-initiated DFI1 interface update request. The counter saturates at max value 65535.

9.4.3.6.80 Counts unsuccessful PHY Master Interface Updates (MasUpdFailCtr)

9.4.3.6.80.1 Offset

Register	Offset
MasUpdFailCtr	174h

9.4.3.6.80.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MasUpdFailCtr															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.6.80.3 Fields

Field	Function
15-0 MasUpdFailCtr	This register increments whenever the PHY asserts a PHY Master Interface request, but the Memory Controller doesn't acknowledge the request within the allowed interval. This register increments whenever the PHY asserts a PHY Master Interface request, but the Memory Controller doesn't acknowledge the request within the allowed interval. (As per section 11 of the Preliminary DFI 4.0 Specification Rev. 2) The counter saturates at max value 65535. In a 2-channel system, a failure occurs if either MC channels fails to ack.

9.4.3.6.81 Counts unsuccessful PHY-initiated DFI0 Interface Updates (PhyUpd0FailCtr)

9.4.3.6.81.1 Offset

Register	Offset
PhyUpd0FailCtr	176h

9.4.3.6.81.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PhyUpd0FailCtr															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

9.4.3.6.81.3 Fields

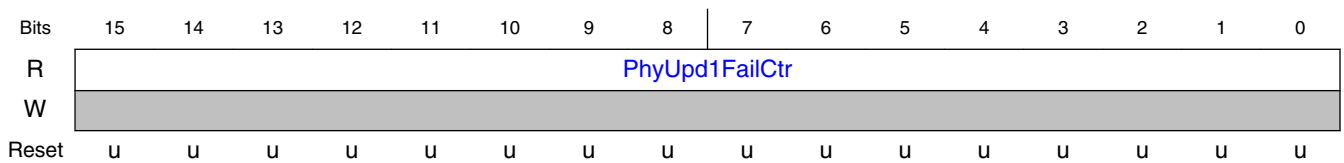
Field	Function
15-0 PhyUpd0FailCtr	This register increments whenever the PHY asserts a DFI0 Interface update request, but the Memory Controller doesn't acknowledge the request within the allowed interval. This register increments whenever the PHY asserts a DFI0 Interface update request, but the Memory Controller doesn't acknowledge the request within the allowed interval. (As per section 11 of the Preliminary DFI 4.0 Specification Rev. 2) The counter saturates at max value 65535.

9.4.3.6.82 Counts unsuccessful PHY-initiated DFI1 Interface Updates (PhyUpd1FailCtr)

9.4.3.6.82.1 Offset

Register	Offset
PhyUpd1FailCtr	178h

9.4.3.6.82.2 Diagram



9.4.3.6.82.3 Fields

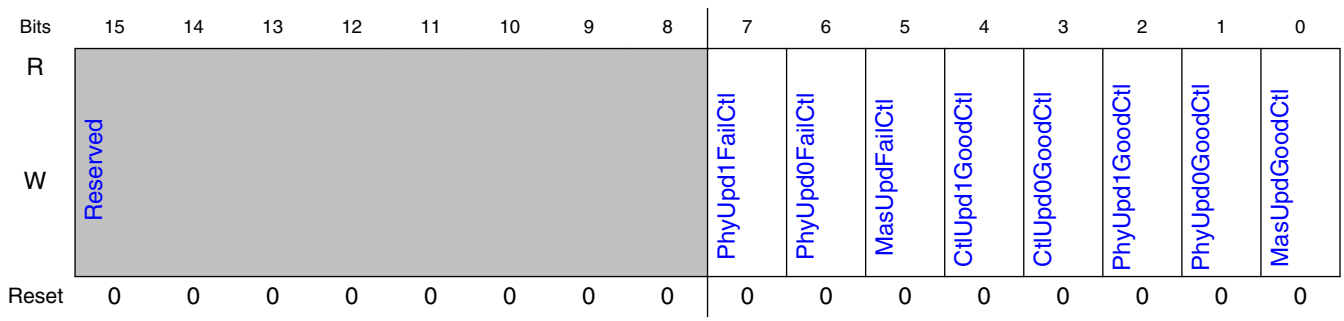
Field	Function
15-0 PhyUpd1FailCtr	This register increments whenever the PHY asserts a DFI1 Interface update request, but the Memory Controller doesn't acknowledge the request within the allowed interval. This register increments whenever the PHY asserts a DFI1 Interface update request, but the Memory Controller doesn't acknowledge the request within the allowed interval. (As per section 11 of the Preliminary DFI 4.0 Specification Rev. 2) The counter saturates at max value 65535.

9.4.3.6.83 Enables for Performance Counters (PhyPerfCtrEnable)

9.4.3.6.83.1 Offset

Register	Offset
PhyPerfCtrEnable	17Ah

9.4.3.6.83.2 Diagram



9.4.3.6.83.3 Fields

Field	Function
15-8 —	Reserved
7 PhyUpd1FailCtl	Enables PhyUpd1FailCtr
6 PhyUpd0FailCtl	Enables PhyUpd0FailCtr
5 MasUpdFailCtl	Enables MasUpdFailCtr
4 CtlUpd1GoodCtl	Enables CtlUpd1GoodCtr
3 CtlUpd0GoodCtl	Enables CtlUpd0GoodCtr
2 PhyUpd1GoodCtl	Enables PhyUpd1GoodCtr
1 PhyUpd0GoodCtl	Enables PhyUpd0GoodCtr

Table continues on the next page...

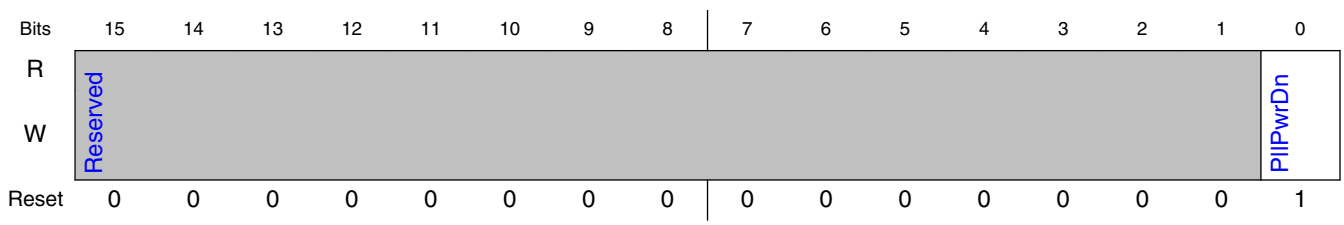
Field	Function
0 MasUpdGoodCtl	Enables MasUpdGoodCtr

9.4.3.6.84 PLL Power Down (PllPwrDn)

9.4.3.6.84.1 Offset

Register	Offset
PllPwrDn	186h

9.4.3.6.84.2 Diagram



9.4.3.6.84.3 Fields

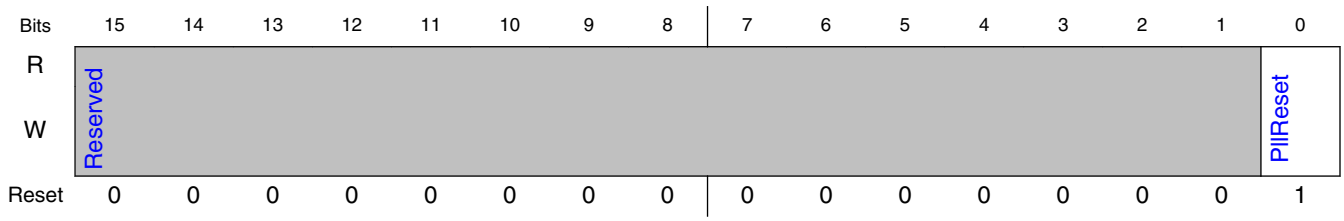
Field	Function
15-1 —	Reserved
0 PllPwrDn	NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. NOTE : This CSR is written by PHY Initialization Engine (PIE) and the data in here will be overwritten. Places the PLL in Power Down when asserted.

9.4.3.6.85 PLL Reset (PllReset)

9.4.3.6.85.1 Offset

Register	Offset
PllReset	188h

9.4.3.6.85.2 Diagram



9.4.3.6.85.3 Fields

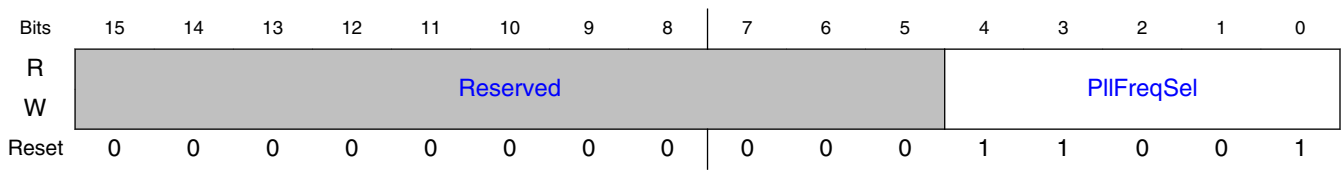
Field	Function
15-1 —	Reserved
0 PllReset	Reserved

9.4.3.6.86 PState dependent PLL Control Register 2 (PllCtrl2_p0)

9.4.3.6.86.1 Offset

Register	Offset
PllCtrl2_p0	18Ah

9.4.3.6.86.2 Diagram



9.4.3.6.86.3 Fields

Field	Function
15-5 —	Reserved

Table continues on the next page...

Field	Function
4-0 PllFreqSel	Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Program based on reference clock freqency (DfiClk) with this table.

9.4.3.6.87 PLL Control Register 0 (PllCtrl0)

9.4.3.6.87.1 Offset

Register	Offset
PllCtrl0	18Ch

9.4.3.6.87.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PllSpareCtrl0	PllLockPhSel		PllLockCntSel	PllGearShift	PllReserved10x9		PllSyncBusByp	PllSyncBusFlush	PllSelDfiFreqRatio	PllBypassMode	PllPreset	PllOutBypEn	PllX2Mode	PllBypSel	PllStandby
Reset	0	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0

9.4.3.6.87.3 Fields

Field	Function
15 PllSpareCtrl0	Spare bits for PLL control.
14-13 PllLockPhSel	Lock detect phase selection. Lock detect phase selection. 0: Test mode reduced lock window 1: Default operation phase lock 2: Test mode (mismatch check) 3: Test mode (+80ps window)
12	Lock detect counter selection.

Table continues on the next page...

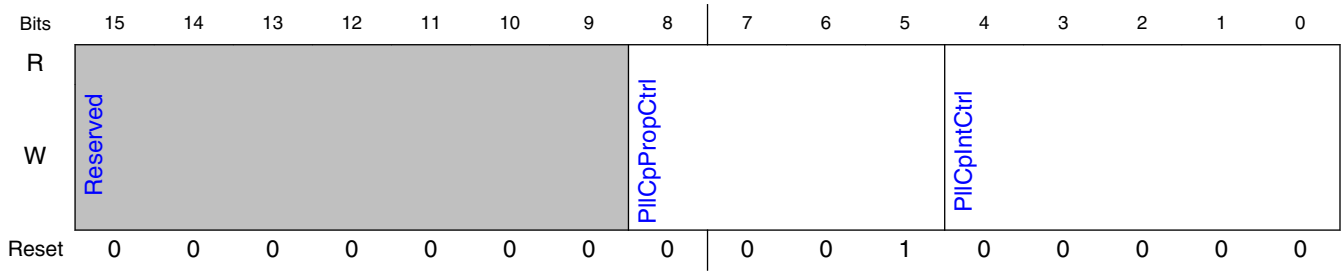
Field	Function
PllLockCntSel	Lock detect counter selection. 0: Normal lock detector behavior phase lock 1: Test mode (Shorten lock cycle count 16->8)
11 PllGearShift	Puts PLL in fast re-locking mode 0: default, normal mode 1: fast relock gear
10-9 PllReserved10x9	for future use.
8 PllSyncBusByp	When asserted bypasses the Pll SyncPulse and uses a synchronizer of the same latency.
7 PllSyncBusFlush	Used to flush the syncbus logic of the PLL during PHY initialization or LP3 Exit sequence. Used to flush the syncbus logic of the PLL during PHY initialization or LP3 Exit sequence. Should Only be toggled when synbus is not used and Register PllOutBypEn is asserted.
6 PllSelDfiFreqRatio	reserved.
5 PllBypassMode	PLL Bypass clock mux control. PLL Bypass clock mux control. If set, BypassPclk input will be used as byp_pll1x_in clock for PLL. Default 0.
4 PllPreset	Put PLL in preset mode.
3 PllOutBypEn	Controls the antiglitch mux on the pllout_x1x2x4 path 1: pllout_x1x2x4 = byp_pllin_x1 0: pllout_x1x2x4 = VCO (SCD) (selected by x2_mode)
2 PllX2Mode	conects to x2_mode pins of PLL. conects to x2_mode pins of PLL. pllout_x4x2 output frequency selection. 0: PLOUT_X4X2 output = 4*pllin_x1 frequency 1: PLOUT_X4X2 output = 2*pllin_x1 frequency
1 PllBypSel	Reserved
0 PllStandby	Connects directly to standby pin of PLL. Connects directly to standby pin of PLL. Puts PLL into standbymode.

9.4.3.6.88 PState dependent PLL Control Register 1 (PllCtrl1_p0)

9.4.3.6.88.1 Offset

Register	Offset
PllCtrl1_p0	18Eh

9.4.3.6.88.2 Diagram



9.4.3.6.88.3 Fields

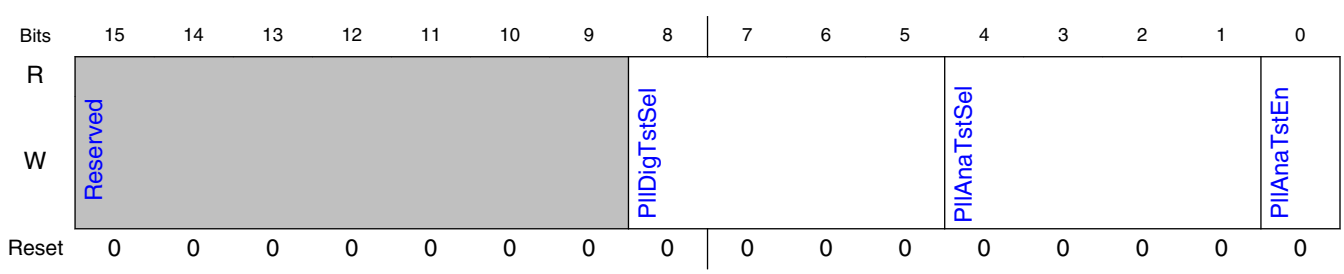
Field	Function
15-9 —	Reserved
8-5 PllCpPropCtrl	connects directly to cp_prop_cntrl<3:0> of PLL. connects directly to cp_prop_cntrl<3:0> of PLL. Charge pump proportional current control.
4-0 PllCpIntCtrl	connects directly to cp_int_cntrl<1:0> in PLL. connects directly to cp_int_cntrl<1:0> in PLL. Charge pump integrating current control.

9.4.3.6.89 PLL Testing Control Register (PllTst)

9.4.3.6.89.1 Offset

Register	Offset
PllTst	190h

9.4.3.6.89.2 Diagram



9.4.3.6.89.3 Fields

Field	Function
15-9 —	Reserved
8-5 PllDigTstSel	Connects directly to pll_dig_test_sel<2:0> of PLL. Connects directly to pll_dig_test_sel<2:0> of PLL. Digital test mux select. Control bits are decoded to enable various digital test signals to be brought out via digital test pin, see decoder table for detail. PllDigTstOut[1] is observable on BP_ALERT and Digital Observation Pin via MtestMuxSel logic. (if enabled in this configuration) PllDigTstOut[0] is observable on Digital Observation Pin via MtestMuxSel logic only. During mission mode PllDigTstSel should be set to 3'h0. Byp = PllBypassMode CSR value Sel = PllDigTstSel CSR value |---------- | | | PllDigTstOut | |Byp| Sel | [1] [0] | Description | ========== ===== | 0 | 000 | 0 | 0 | This code should be set during normal use | | | | | to minimize clock jitter | 0 | 001 | fbk_clk_ldet | pllin_x1 | Buffered version of fbk_clk at PFD input | 0 | 010 | ref_clk_ldet | pllin_x1 | Buffered version of ref_clk at PFD input | 0 | 011 | pllin_x1 | pllin_x1 | For calibrating delay skew between two | | | | | test outputs - input reference clock | 0 | 100 | clk_fbk | pllin_x1 | Buffered version of pllout_x1 | 0 | 101 | pllout_amux_x1x2x4 | pllin_x1 | Syncbus clock output (Fout<Fmax_dto only) | 0 | 110 | en_count | pll_lock | Lock detector state test | 0 | 111 | standby_eff | eoc | Calibration state test | 1 | 000 | 0 | 0 | This code should be set during normal use | | | | | to minimize clock jitter | 1 | 001 | fbk_clk_ldet | pllin_x1 | Buffered version of fbk_clk at PFD input | | | | | if PLL in mission/byp 1 modes | 1 | 010 | ref_clk_ldet | pllin_x1 | Buffered version of ref_clk at PFD input | | | | | if PLL in mission/byp 1 modes | 1 | 011 | pllin_x1 | pllin_x1 | For calibrating delay skew between two | | | | | test outputs - input reference clock

Table continues on the next page...

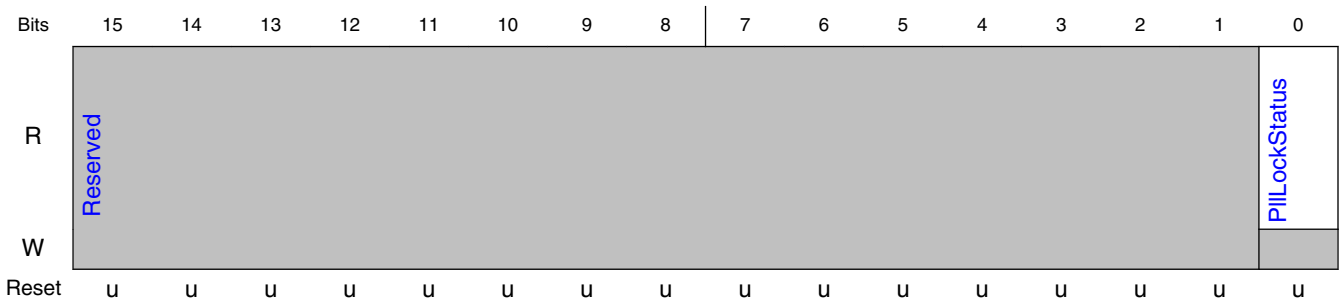
Field	Function
	| 1 | 100 | clk_fbk | pllin_x1 | Buffered version of pllout_x1 if PLL in | | | | | mission/byp 1 modes | 1 | 101 | pllout_amux_x1x2x4 | pllin_x1 | Syncbus clock output (Fout<Fmax_dto only) | 1 | 110 | en_count | pll_lock | Lock detector test if PLL in mission/byp | | | | | 1 modes | 1 | 111 | standby_eff | eoc | Calibration state test if PLL in | | | | | mission/byp 1 modes
4-1 PllAnaTstSel	Connects directly to pll_ana_test_sel<3:0> of PLL. Connects directly to pll_ana_test_sel<3:0> of PLL. Control bits are decoded to enable various analog test signals to be brought out via analog test pin (pll_analog_test). Table below provides decode detail. |PllAnaTstSel | Pll Analog | Description | [4:1] | Pin Function | |========== | 0000 | Hi-Z | This code should be set when the PLL | | | is in normal use or when the digital | | | test port is being used for accurate | | | timing measurements | 0001 | core | Override vmarg_ext with analog pin | 0010 | core | Override regulator input with analog pin | 0011 | core | Observe gd voltage on analog test pin | 0100 | core | Observe vp voltage on analog test pin | 0101 | core | Observe vph voltage on analog test pin | 0110 | core | Observe vreg_cp voltage on analog test pin | 0111 | core | Observe v2i current (~10uA) on analog test pin | 1000 | core | Observe vreg_v2i voltage on analog test pin | 1001 | core | Observe vbp voltage from current source on | | | analog pin | 1010 | core | Observe vbn voltage from current source on | | | analog tpin | 1011 | core | NC | 1100 | core | NC | 1101 | VSS | 0v probe ir drop test | 1110 | VP | Core supply probe SCD IR drop test | 1111 | VPH | Core Supply probe Refdiv IR drop test
0 PllAnaTstEn	Connects directly to pll_ana_test_en of PLL. Connects directly to pll_ana_test_en of PLL. Analog test port enable 0 = analog test port disabled (hi-Z) 1 = analog test port enabled

9.4.3.6.90 PLL's pll_lock pin output (PllLockStatus)

9.4.3.6.90.1 Offset

Register	Offset
PllLockStatus	192h

9.4.3.6.90.2 Diagram



9.4.3.6.90.3 Fields

Field	Function
15-1 —	Reserved
0 PllLockStatus	Directly connected to the pll_Lock output. Directly connected to the pll_Lock output. When asserted Pll macro has asserted the Lock status. To avoid metastability issues, this register should perform a majority-vote after reading this register several times.

9.4.3.6.91 Additional controls for PLL CP/VCO modes of operation (PllTestMode_p0)

9.4.3.6.91.1 Offset

Register	Offset
PllTestMode_p0	194h

9.4.3.6.91.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PllTestMode_p0															
Reset	0	0	0	0	0	0	0	1	0	0	1	0	0	1	0	0

9.4.3.6.91.3 Fields

Field	Function
15-0 PllTestMode_p0	It is required to use default values for this CSR unless directed otherwise by Synopsys. It is required to use default values for this CSR unless directed otherwise by Synopsys. Directly connected to testmode[15:0] pin of PLL

9.4.3.6.92 PLL Control Register 3 (PllCtrl3)

9.4.3.6.92.1 Offset

Register	Offset
PllCtrl3	196h

9.4.3.6.92.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PllEnCal	PllForceCal	PllDacValIn					PllMaxRange					PllSpare			
Reset	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0

9.4.3.6.92.3 Fields

Field	Function
15	Calibration will run at standby rising edge if en_cal=1 if en_cal=0 calibration will not run

Table continues on the next page...

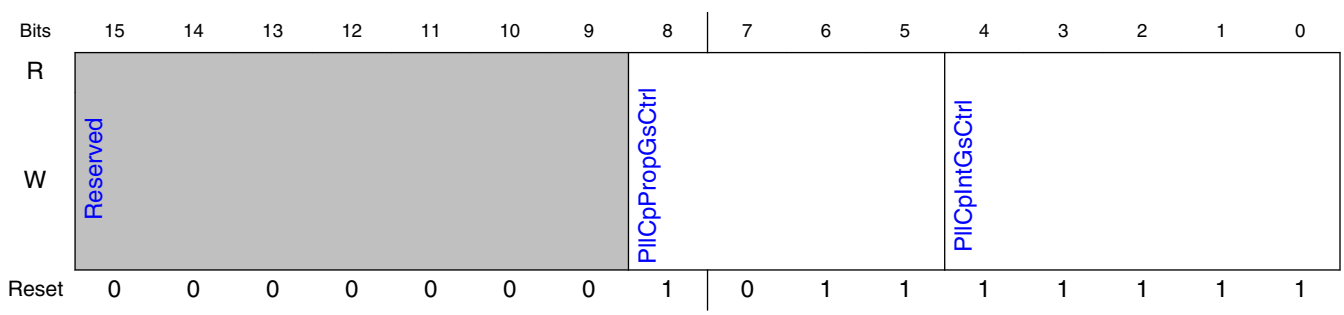
Field	Function
PllEnCal	
14 PllForceCal	connects directly to force_cal of PLL. connects directly to force_cal of PLL. Forces Calibration code dacval_in<4:0> to be used at preset mode if force_cal=1 if force_cal=0 previous stored value will be used
13-9 PllDacValIn	connects directly to dacval_in<4:0> of PLL. connects directly to dacval_in<4:0> of PLL. Calibration DAC input current setting, can be forced at rising edge of preset if force_cal =1
8-4 PllMaxRange	connects directly to maxrange of PLL. connects directly to maxrange of PLL. Setting for Saturation control of PLL.
3-0 PllSpare	Spare bits for future PLL control modes

9.4.3.6.93 PState dependent PLL Control Register 4 (PllCtrl4_p0)

9.4.3.6.93.1 Offset

Register	Offset
PllCtrl4_p0	198h

9.4.3.6.93.2 Diagram



9.4.3.6.93.3 Fields

Field	Function
15-9	Reserved

Table continues on the next page...

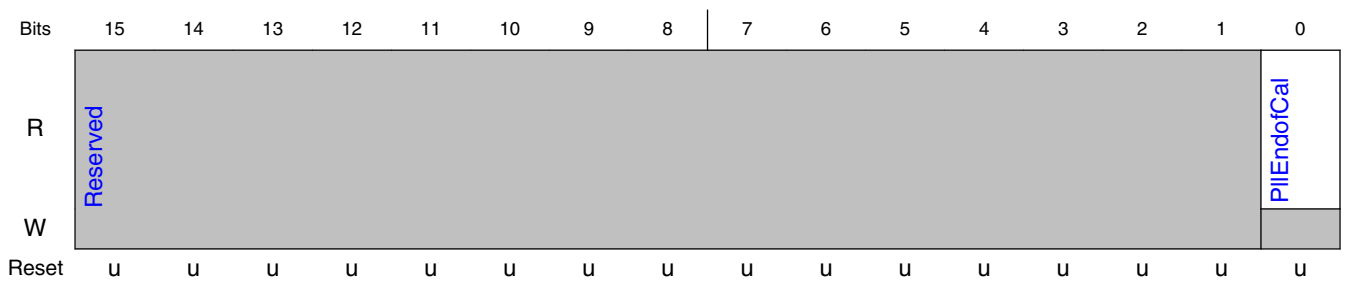
Field	Function
—	
8-5 PllCpPropGsCtrl	connects directly to cp_prop_gs_cntrl<3:0> of PLL. connects directly to cp_prop_gs_cntrl<3:0> of PLL. Charge pump proportional current control for fast relock and gearshift.
4-0 PllCpIntGsCtrl	connects directly to cp_int_gs_cntrl<4:0> in PLL. connects directly to cp_int_gs_cntrl<4:0> in PLL. Charge pump integrating current control for fast relock and gearshift.

9.4.3.6.94 PLL's eoc (end of calibration) output (PllEndofCal)

9.4.3.6.94.1 Offset

Register	Offset
PllEndofCal	19Ah

9.4.3.6.94.2 Diagram



9.4.3.6.94.3 Fields

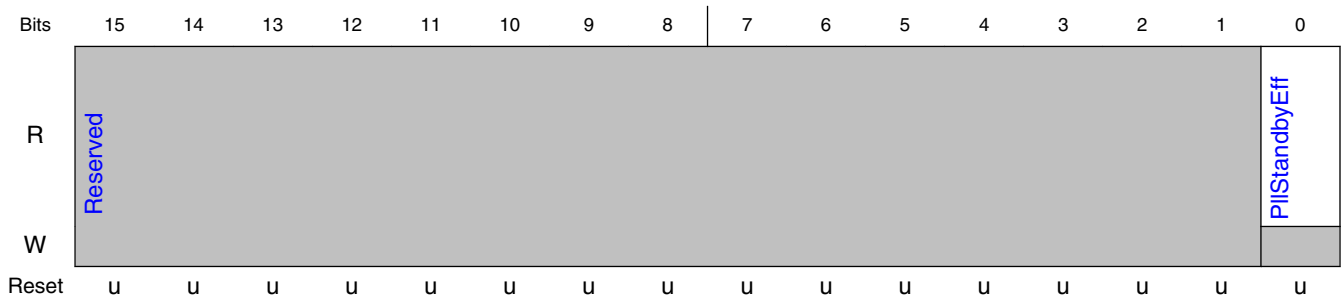
Field	Function
15-1 —	Reserved
0 PllEndofCal	Directly connected to the pll's eoc output.

9.4.3.6.95 PLL's standby_eff (effective standby) output (PllStandbyEff)

9.4.3.6.95.1 Offset

Register	Offset
PllStandbyEff	19Ch

9.4.3.6.95.2 Diagram



9.4.3.6.95.3 Fields

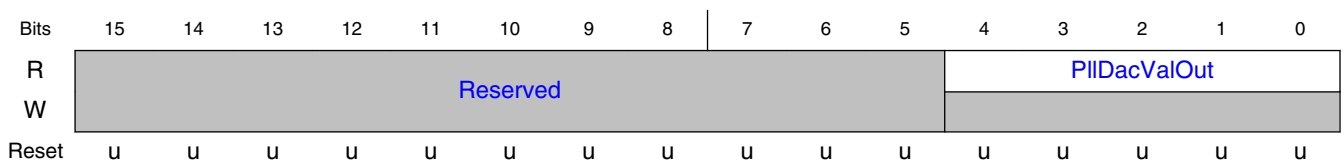
Field	Function
15-1 —	Reserved
0 PllStandbyEff	Returns state off PLL standby. Returns state off PLL standby. PLL will be in standby in PHY LP2 states.

9.4.3.6.96 PLL's Dacval_out output (PllDacValOut)

9.4.3.6.96.1 Offset

Register	Offset
PllDacValOut	19Eh

9.4.3.6.96.2 Diagram



9.4.3.6.96.3 Fields

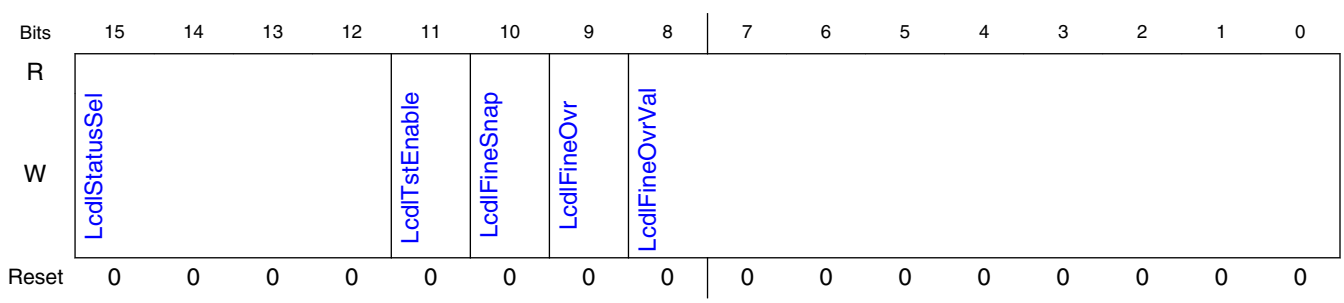
Field	Function
15-5 —	Reserved
4-0 PllDacValOut	Directly connected to the pll's dacval_out output. Directly connected to the pll's dacval_out output. Used for observation of PLL DAC code.

9.4.3.6.97 Controls for use in observing and testing the LCDLs. (LcdlDbgCntl)

9.4.3.6.97.1 Offset

Register	Offset
LcdlDbgCntl	1C6h

9.4.3.6.97.2 Diagram



9.4.3.6.97.3 Fields

Field	Function
15-12 LcdlStatusSel	Selects the LCDL status, from among the status for the 16 LCDLs in the DBYTE, for reading via Register DxLcdlStatus and an LCDL from among the LCDLs in the ANIB for reading via Register AcLcdlStatus LcdlStatusSel source for DxLcdlStatus source for AcLcdlStatus 15 lcdl_rxclk1t reserved 14 lcdl_rxclk0t reserved 13 lcdl_rxclk1c reserved 12 lcdl_rxclk0c reserved 11 lcdl_rxen1 anib11-tx 10 lcdl_rxen0 anib10-tx 9 lcdl_txln9 (dqs-lower) anib9-tx 8 lcdl_txln8 (dm/dqs-upper) anib8-tx 7 lcdl_txln7 (dq7) anib7-tx 6 lcdl_txln6 (dq6) anib6-tx 5 lcdl_txln5 (dq5) anib5-tx 4 lcdl_txln4 (dq4) anib4-tx 3 lcdl_txln3 (dq3) anib3-tx 2 lcdl_txln2 (dq2) anib2-tx 1 lcdl_txln1 (dq1) anib1-tx 0 lcdl_txln0 (dq0) anib0-tx
11 LcdlTstEnable	Enables the debug/test operations and status Ovr,Snap,StickyLock,StickyUnlock, and LiveLock.

Table continues on the next page...

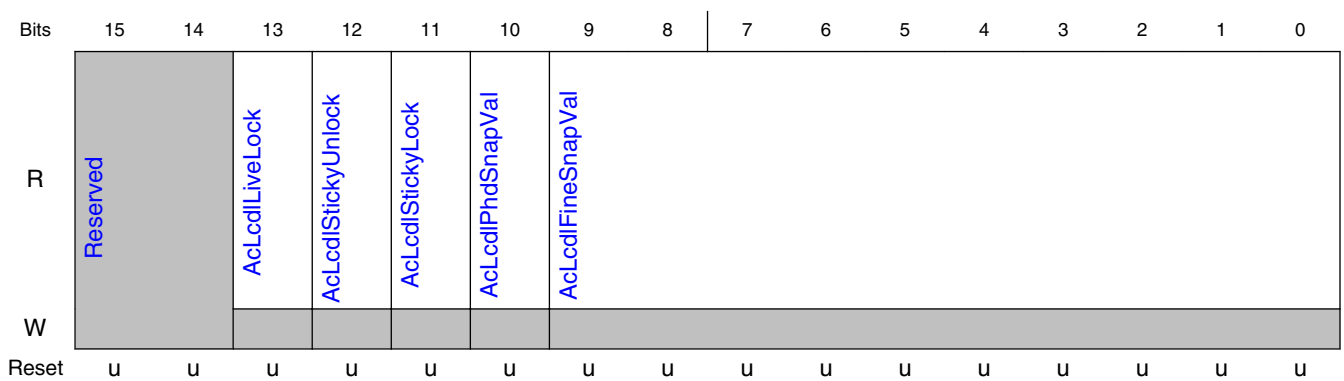
Field	Function
10 LcdlFineSnap	Latch enable for reading the present LCDL 1UI estimate code in LcdlFineSnapVal and the present phase-detector value in LcdlPhdSnapVal
9 LcdlFineOvr	Forces the value of the present LCDL 1UI estimate code to be LcdlFineOvrVal for all LCDLs.
8-0 LcdlFineOvrVal	Value forced as the initial value while LcdlTstEnable=1 and LcdlFineOvr. Value forced as the initial value while LcdlTstEnable=1 and LcdlFineOvr. After the deassertion of LcdlFineOvr, the locking state-machine will attempt locking.

9.4.3.6.98 Debug status of the DBYTE LCDL (AcLcdlStatus)

9.4.3.6.98.1 Offset

Register	Offset
AcLcdlStatus	1C8h

9.4.3.6.98.2 Diagram



9.4.3.6.98.3 Fields

Field	Function
15-14 —	Reserved
13 AcLcdlLiveLock	present value of whether the LCDL is locked, valid when LcdlTstEnable=1.
12	latched value of whether the LCDL ever lost lock after the assertion of LcdlTstEnable.

Table continues on the next page...

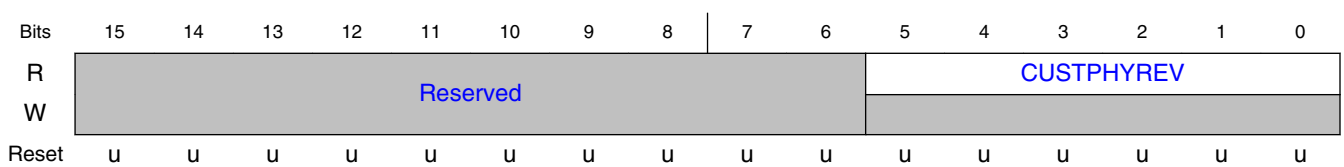
Field	Function
AcLcdlStickyUnlock	
11 AcLcdlStickyLock	latched value of whether the LCDL ever achieved lock after the assertion of LcdlTstEnable.
10 AcLcdlPhdSnapVal	Value of the LCDL phase-detector output, latched by pulse on LcdlFineSnap while csr LcdlTstEnable=1.
9-0 AcLcdlFineSnapVal	Value of the LCDL 1UI estimate code, latched by pulse on csrLcdlFineSnap while csr LcdlTstEnable=1. Value of the LCDL 1UI estimate code, latched by pulse on csrLcdlFineSnap while csr LcdlTstEnable=1. Index 9 is reserved for growth.

9.4.3.6.99 Customer settable by the customer (CUSTPHYREV)

9.4.3.6.99.1 Offset

Register	Offset
CUSTPHYREV	1DAh

9.4.3.6.99.2 Diagram



9.4.3.6.99.3 Fields

Field	Function
15-6 —	Reserved
5-0 CUSTPHYREV	The customer settable PHY version number. The customer settable PHY version number. The value is derived from the `define DWC_DDRPHY_CUST_PHYREV

9.4.3.6.100 The hardware version of this PHY, excluding the PUB (PHYREV)

9.4.3.6.100.1 Offset

Register	Offset
PHYREV	1DCh

9.4.3.6.100.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PHYMJR								PHYMDR				PHYMNR			
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

9.4.3.6.100.3 Fields

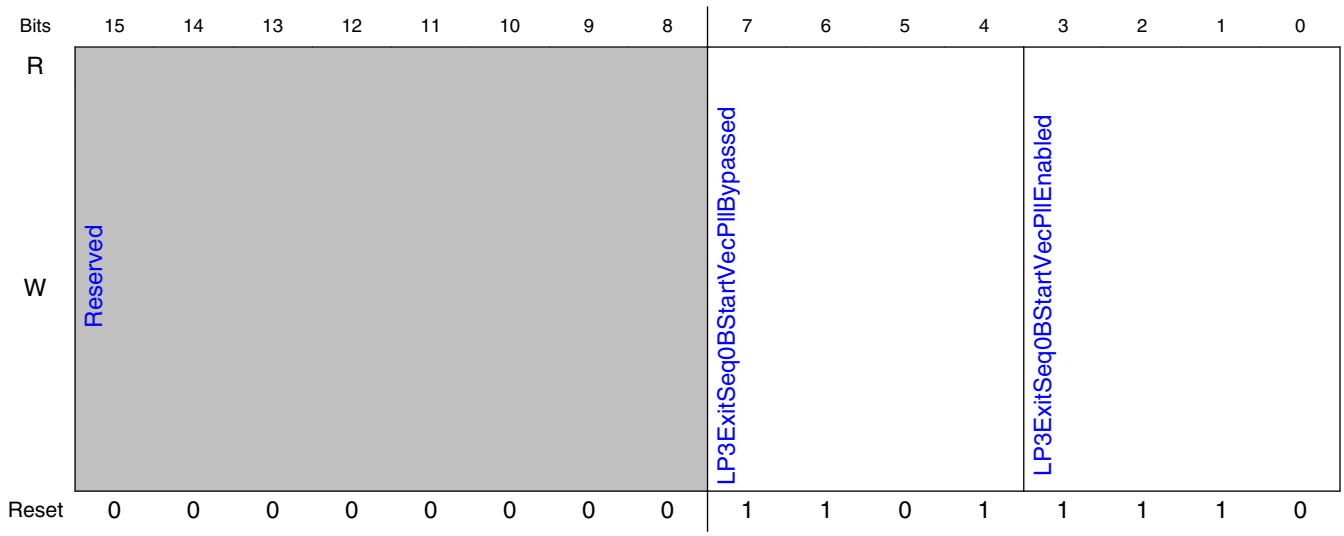
Field	Function
15-8 PHYMJR	Indicates major revision of the PHY.
7-4 PHYMDR	Indicates moderate revision of the PHY.
3-0 PHYMNR	Indicates minor update of the PHY.

9.4.3.6.101 Start vector value to be used for LP3-exit or Init PIE Sequence (LP3ExitSeq0BStartVector)

9.4.3.6.101.1 Offset

Register	Offset
LP3ExitSeq0BStartVector	1DEh

9.4.3.6.101.2 Diagram



9.4.3.6.101.3 Fields

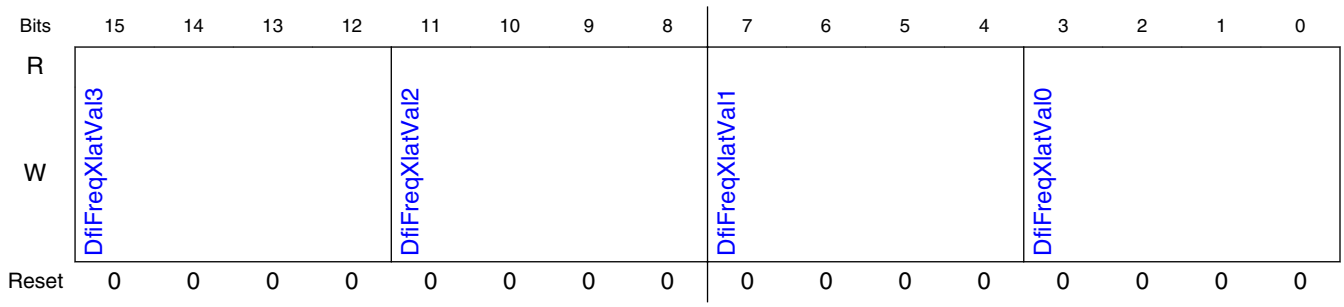
Field	Function
15-8 —	Reserved
7-4 LP3ExitSeq0BStartVecPllBypassed	PIE Start Vector value to be used for LP3-exit or Init and target P-state has PLL bypassed
3-0 LP3ExitSeq0BStartVecPllEnabled	PIE Start Vector value to be used for LP3-exit or Init and target P-state has PLL enabled

9.4.3.6.102 DFI Frequency Translation Register 0 (DfiFreqXlat0)

9.4.3.6.102.1 Offset

Register	Offset
DfiFreqXlat0	1E0h

9.4.3.6.102.2 Diagram



9.4.3.6.102.3 Fields

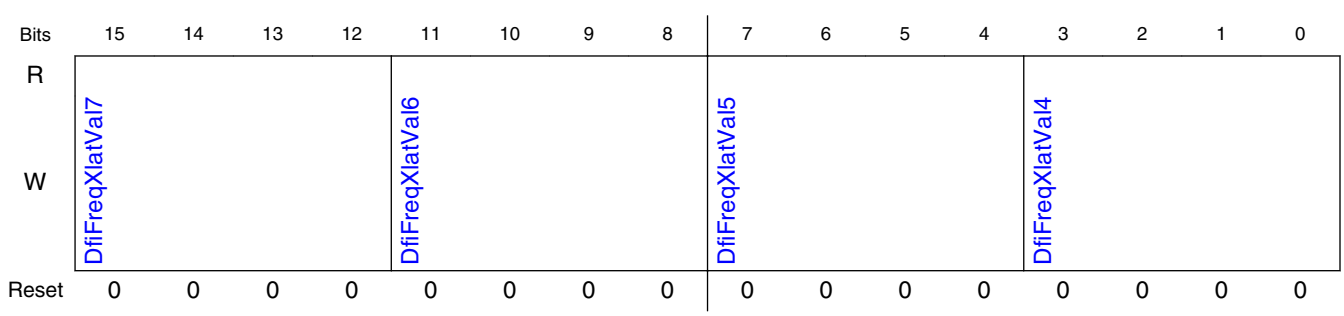
Field	Function
15-12 DfiFreqXlatVal3	The sequencer start vector used when dfi_freq value is 3.
11-8 DfiFreqXlatVal2	The sequencer start vector used when dfi_freq value is 2.
7-4 DfiFreqXlatVal1	The sequencer start vector used when dfi_freq value is 1.
3-0 DfiFreqXlatVal0	The sequencer start vector used when dfi_freq value is 0.

9.4.3.6.103 DFI Frequency Translation Register 1 (DfiFreqXlat1)

9.4.3.6.103.1 Offset

Register	Offset
DfiFreqXlat1	1E2h

9.4.3.6.103.2 Diagram



9.4.3.6.103.3 Fields

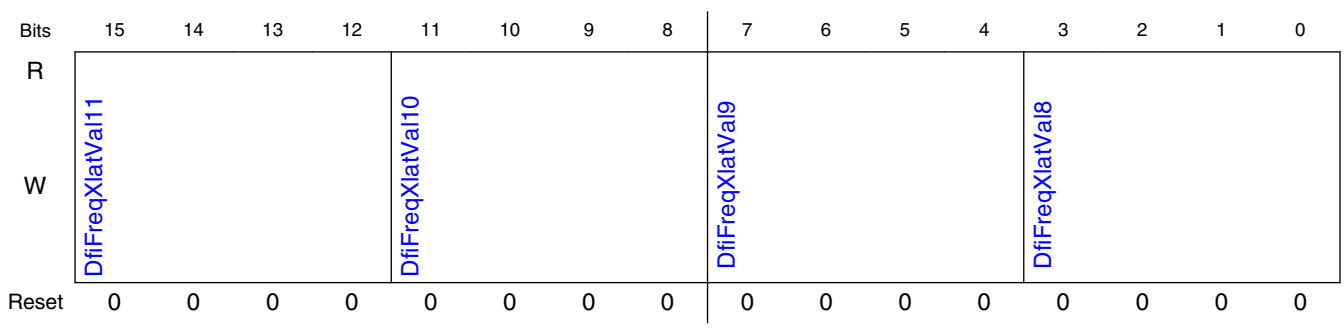
Field	Function
15-12 DfiFreqXlatVal7	The sequencer start vector used when dfi_freq value is 7.
11-8 DfiFreqXlatVal6	The sequencer start vector used when dfi_freq value is 6.
7-4 DfiFreqXlatVal5	The sequencer start vector used when dfi_freq value is 5.
3-0 DfiFreqXlatVal4	The sequencer start vector used when dfi_freq value is 4.

9.4.3.6.104 DFI Frequency Translation Register 2 (DfiFreqXlat2)

9.4.3.6.104.1 Offset

Register	Offset
DfiFreqXlat2	1E4h

9.4.3.6.104.2 Diagram



9.4.3.6.104.3 Fields

Field	Function
15-12 DfiFreqXlatVal11	The sequencer start vector used when dfi_freq value is 11.

Table continues on the next page...

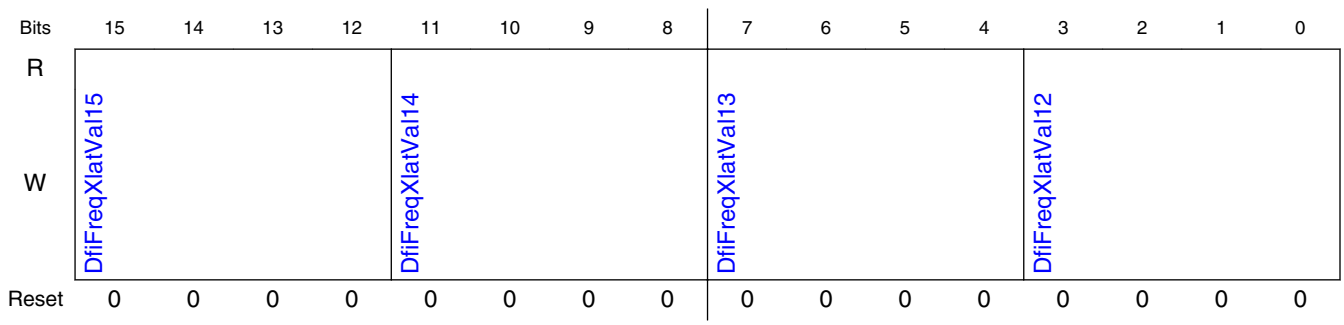
Field	Function
11-8 DfiFreqXlatVal10	The sequencer start vector used when dfi_freq value is 10.
7-4 DfiFreqXlatVal9	The sequencer start vector used when dfi_freq value is 9.
3-0 DfiFreqXlatVal8	The sequencer start vector used when dfi_freq value is 8.

9.4.3.6.105 DFI Frequency Translation Register 3 (DfiFreqXlat3)

9.4.3.6.105.1 Offset

Register	Offset
DfiFreqXlat3	1E6h

9.4.3.6.105.2 Diagram



9.4.3.6.105.3 Fields

Field	Function
15-12 DfiFreqXlatVal15	The sequencer start vector used when dfi_freq value is 15.
11-8 DfiFreqXlatVal14	The sequencer start vector used when dfi_freq value is 14.
7-4	The sequencer start vector used when dfi_freq value is 13.

Table continues on the next page...

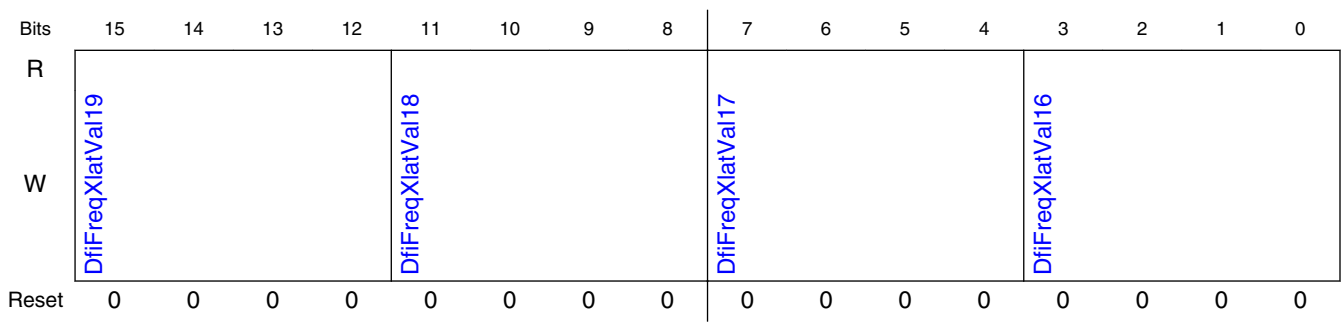
Field	Function
DfiFreqXlatVal13	
3-0 DfiFreqXlatVal12	The sequencer start vector used when dfi_freq value is 12.

9.4.3.6.106 DFI Frequency Translation Register 4 (DfiFreqXlat4)

9.4.3.6.106.1 Offset

Register	Offset
DfiFreqXlat4	1E8h

9.4.3.6.106.2 Diagram



9.4.3.6.106.3 Fields

Field	Function
15-12 DfiFreqXlatVal19	The sequencer start vector used when dfi_freq value is 19.
11-8 DfiFreqXlatVal18	The sequencer start vector used when dfi_freq value is 18.
7-4 DfiFreqXlatVal17	The sequencer start vector used when dfi_freq value is 17.
3-0 DfiFreqXlatVal16	The sequencer start vector used when dfi_freq value is 16.

9.4.3.6.107 DFI Frequency Translation Register 5 (DfiFreqXlat5)

9.4.3.6.107.1 Offset

Register	Offset
DfiFreqXlat5	1EAh

9.4.3.6.107.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DfiFreqXlatVal23				DfiFreqXlatVal22				DfiFreqXlatVal21				DfiFreqXlatVal20			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.107.3 Fields

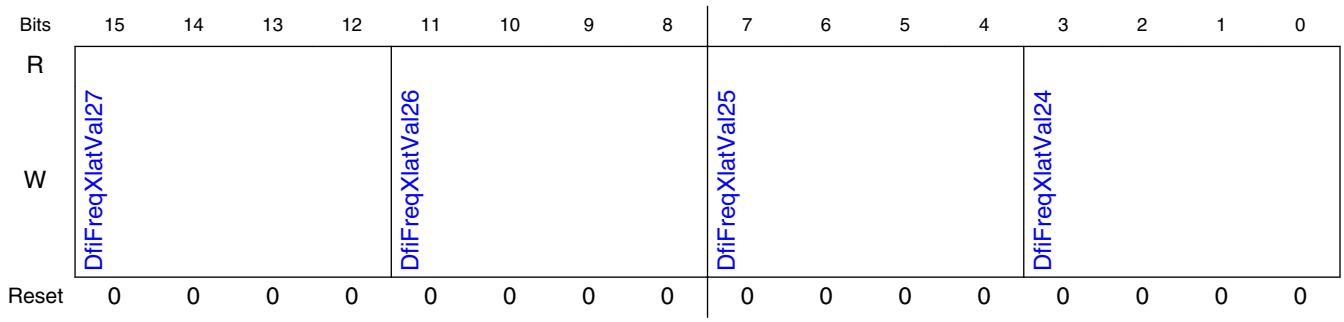
Field	Function
15-12 DfiFreqXlatVal23	The sequencer start vector used when dfi_freq value is 23.
11-8 DfiFreqXlatVal22	The sequencer start vector used when dfi_freq value is 22.
7-4 DfiFreqXlatVal21	The sequencer start vector used when dfi_freq value is 21.
3-0 DfiFreqXlatVal20	The sequencer start vector used when dfi_freq value is 20.

9.4.3.6.108 DFI Frequency Translation Register 6 (DfiFreqXlat6)

9.4.3.6.108.1 Offset

Register	Offset
DfiFreqXlat6	1ECh

9.4.3.6.108.2 Diagram



9.4.3.6.108.3 Fields

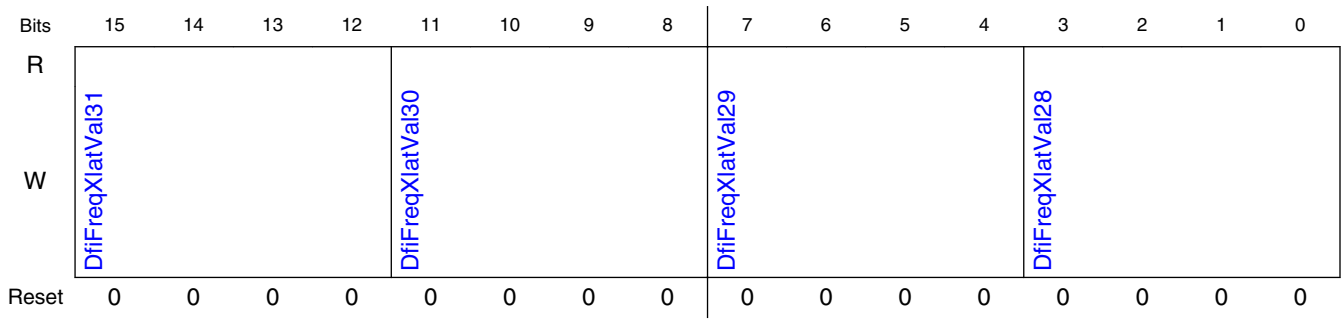
Field	Function
15-12 DfiFreqXlatVal27	The sequencer start vector used when dfi_freq value is 27.
11-8 DfiFreqXlatVal26	The sequencer start vector used when dfi_freq value is 26.
7-4 DfiFreqXlatVal25	The sequencer start vector used when dfi_freq value is 25.
3-0 DfiFreqXlatVal24	The sequencer start vector used when dfi_freq value is 24.

9.4.3.6.109 DFI Frequency Translation Register 7 (DfiFreqXlat7)

9.4.3.6.109.1 Offset

Register	Offset
DfiFreqXlat7	1EEh

9.4.3.6.109.2 Diagram



9.4.3.6.109.3 Fields

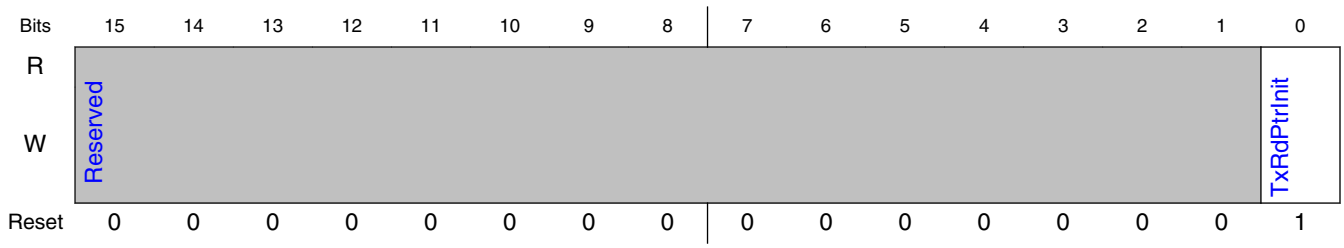
Field	Function
15-12 DfiFreqXlatVal31	The sequencer start vector used when dfi_freq value is 31.
11-8 DfiFreqXlatVal30	The sequencer start vector used when dfi_freq value is 30.
7-4 DfiFreqXlatVal29	The sequencer start vector used when dfi_freq value is 29.
3-0 DfiFreqXlatVal28	The sequencer start vector used when dfi_freq value is 28.

9.4.3.6.110 TxRdPtrInit control register (TxRdPtrInit)

9.4.3.6.110.1 Offset

Register	Offset
TxRdPtrInit	1F0h

9.4.3.6.110.2 Diagram



9.4.3.6.110.3 Fields

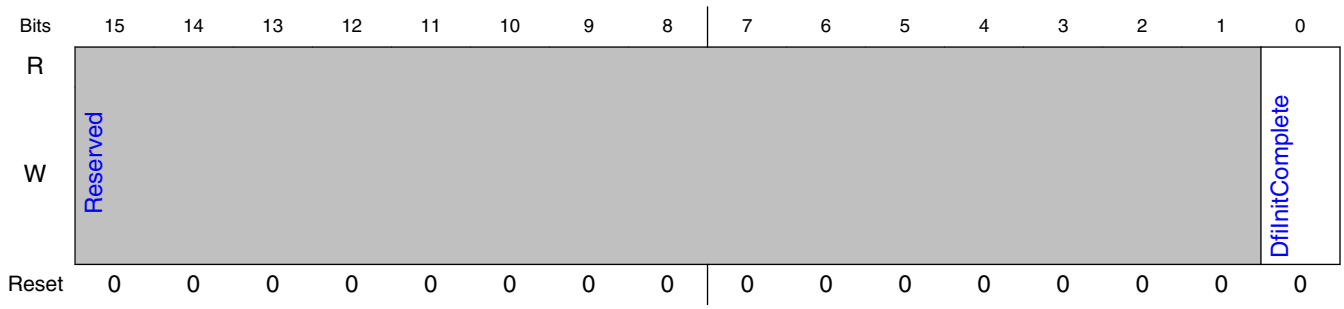
Field	Function
15-1 —	Reserved
0 TxRdPtrInit	This register directly controls TxRdPtrInit, and is meant to be written by the PState sequencer as part of the power state switching sequence. This register directly controls TxRdPtrInit, and is meant to be written by the PState sequencer as part of the power state switching sequence. It should be written to a 1 once the transition to LP2 is complete and 32 cycles before the de-asserting of init_complete. Likewise it should be written to 0 once the transition to an active PState is complete and 32 cycles before the assertion of init_complete. It should reset to a 1 on both warm and cold reset.

9.4.3.6.111 DFI Init Complete control (DfiInitComplete)

9.4.3.6.111.1 Offset

Register	Offset
DfiInitComplete	1F2h

9.4.3.6.111.2 Diagram



9.4.3.6.111.3 Fields

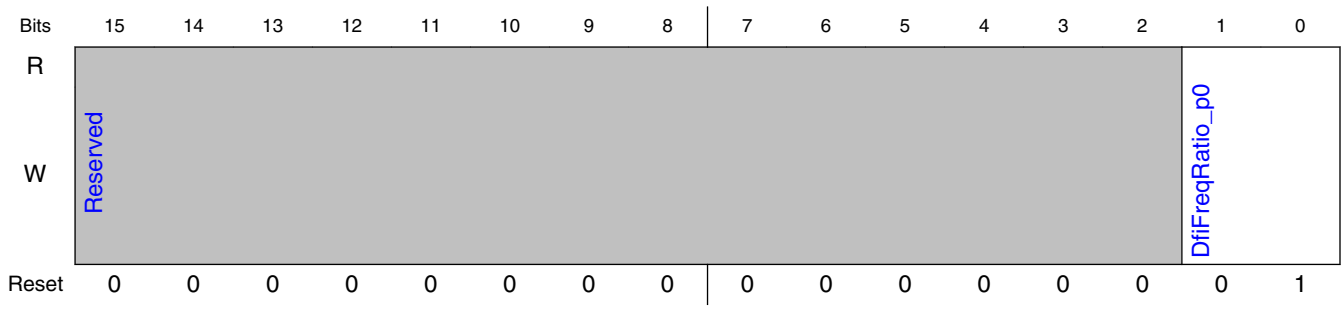
Field	Function
15-1 —	Reserved
0 DfiInitComplete	This register directly controls DfiInitComplete, and is meant to be written by the PState sequencer as part of the power state switching sequence. This register directly controls DfiInitComplete, and is meant to be written by the PState sequencer as part of the power state switching sequence. It should be written to a 1 once the transition to LP2 is complete. Likewise it should be written to 0 once the transition to an active and PState is complete. It should reset to a 0 on both warm and cold reset.

9.4.3.6.112 DFI Frequency Ratio (DfiFreqRatio_p0)

9.4.3.6.112.1 Offset

Register	Offset
DfiFreqRatio_p0	1F4h

9.4.3.6.112.2 Diagram



9.4.3.6.112.3 Fields

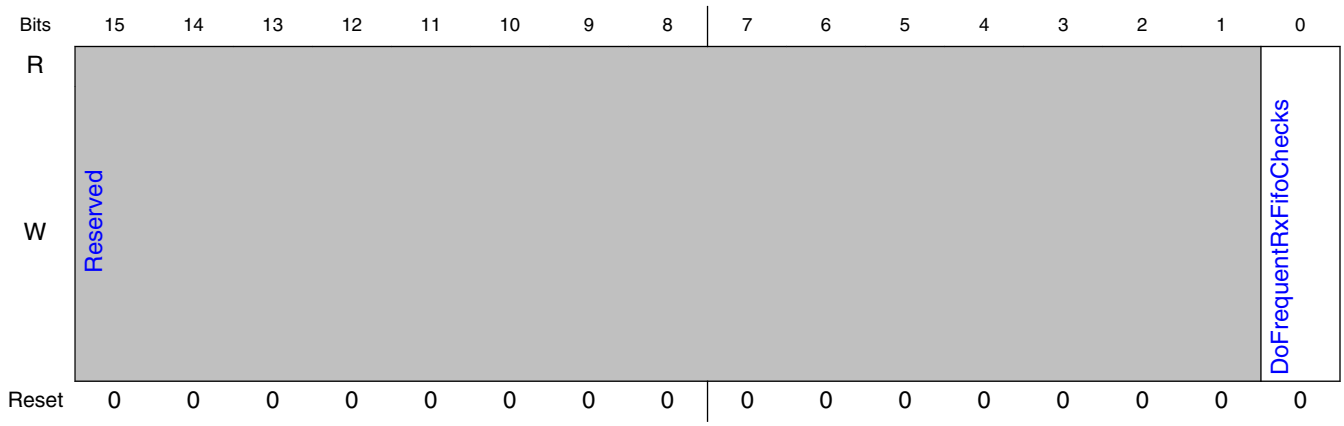
Field	Function
15-2 —	Reserved
1-0 DfiFreqRatio_p0	Used in dwc_ddrphy_pub_serdes to serialize or de-serialize DFI signals 00 = 1:1 mode 01 = 1:2 mode 1x = 1:4 mode* *Note: 1:4 is for future pub revision.

9.4.3.6.113 Enable more frequent consistency checks of the RX FIFOs (RxFifoChecks)

9.4.3.6.113.1 Offset

Register	Offset
RxFifoChecks	1F6h

9.4.3.6.113.2 Diagram



9.4.3.6.113.3 Fields

Field	Function
15-1 —	Reserved
0 DoFrequentRxFifoChecks	When 0, read data FIFO pointer consistency checks are performed only during sideband transactions (i. When 0, read data FIFO pointer consistency checks are performed only during sideband transactions (i.e., phyupd, ctrlupd, lp_data, phymstr) or when dfi_init_start is asserted. This option relies upon intentional quiescing of the DFI interface in order for the check to be performed. When 1, read data FIFO consistency checks are performed as when this bit is 0, and also at the beginning of each break in read data traffic from the DRAM. This options provides the lowest latency in reporting of a discrepancy in the read data FIFO pointer values.

9.4.3.6.114 Maps PHY CAA lane 0 from dfi0_address of the index of the register contents (MapCAA0toDfi)

9.4.3.6.114.1 Offset

Register	Offset
MapCAA0toDfi	200h

9.4.3.6.114.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA0toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.114.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA0toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 0. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 0.

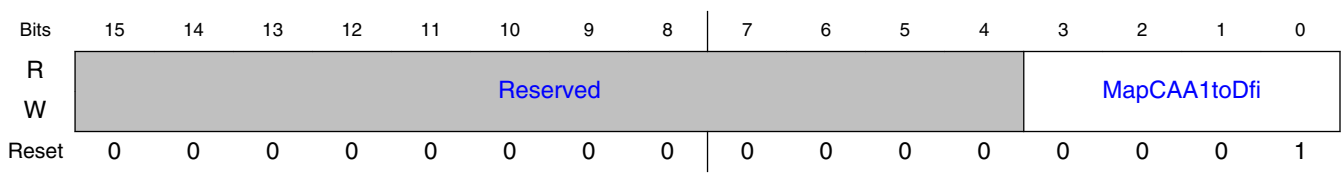
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA0 is mapped from dfi0_address[2], then Register MapCAA0toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.115 Maps PHY CAA lane 1 from dfi0_address of the index of the register contents (MapCAA1toDfi)

9.4.3.6.115.1 Offset

Register	Offset
MapCAA1toDfi	202h

9.4.3.6.115.2 Diagram



9.4.3.6.115.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA1toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 1. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 1. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA1 is mapped from dfi0_address[2], then Register MapCAA1toDfi should be 2.

Field	Function
	LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.116 Maps PHY CAA lane 2 from dfi0_address of the index of the register contents (MapCAA2toDfi)

9.4.3.6.116.1 Offset

Register	Offset
MapCAA2toDfi	204h

9.4.3.6.116.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA2toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

9.4.3.6.116.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA2toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 2. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 2. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA2 is mapped from dfi0_address[2], then Register MapCAA2toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved

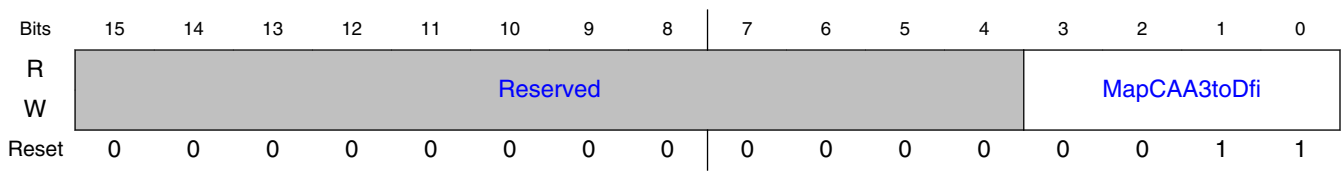
Field	Function
	LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.117 Maps PHY CAA lane 3 from dfi0_address of the index of the register contents (MapCAA3toDfi)

9.4.3.6.117.1 Offset

Register	Offset
MapCAA3toDfi	206h

9.4.3.6.117.2 Diagram



9.4.3.6.117.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA3toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 3. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 3. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA3 is mapped from dfi0_address[2], then Register MapCAA3toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.118 Maps PHY CAA lane 4 from dfi0_address of the index of the register contents (MapCAA4toDfi)

9.4.3.6.118.1 Offset

Register	Offset
MapCAA4toDfi	208h

9.4.3.6.118.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA4toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

9.4.3.6.118.3 Fields

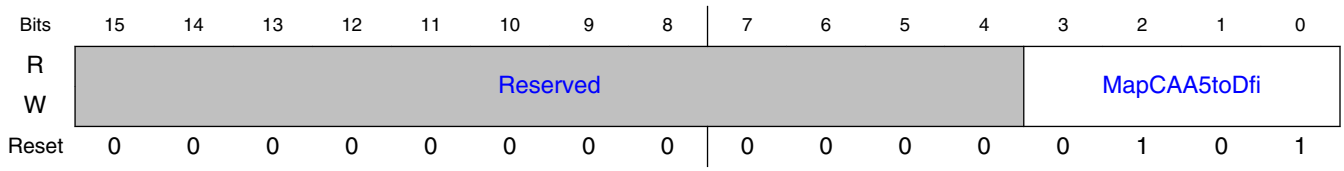
Field	Function
15-4 —	Reserved
3-0 MapCAA4toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 4. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 4. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA4 is mapped from dfi0_address[2], then Register MapCAA4toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.119 Maps PHY CAA lane 5 from dfi0_address of the index of the register contents (MapCAA5toDfi)

9.4.3.6.119.1 Offset

Register	Offset
MapCAA5toDfi	20Ah

9.4.3.6.119.2 Diagram



9.4.3.6.119.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA5toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 5. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 5. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA5 is mapped from dfi0_address[2], then Register MapCAA5toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.120 Maps PHY CAA lane 6 from dfi0_address of the index of the register contents (MapCAA6toDfi)

9.4.3.6.120.1 Offset

Register	Offset
MapCAA6toDfi	20Ch

9.4.3.6.120.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA6toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

9.4.3.6.120.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA6toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 6. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 6. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA6 is mapped from dfi0_address[2], then Register MapCAA6toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.121 Maps PHY CAA lane 7 from dfi0_address of the index of the register contents (MapCAA7toDfi)

9.4.3.6.121.1 Offset

Register	Offset
MapCAA7toDfi	20Eh

9.4.3.6.121.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA7toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

9.4.3.6.121.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAA7toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 7. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 7. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA7 is mapped from dfi0_address[2], then Register MapCAA7toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.122 Maps PHY CAA lane 8 from dfi0_address of the index of the register contents (MapCAA8toDfi)

9.4.3.6.122.1 Offset

Register	Offset
MapCAA8toDfi	210h

9.4.3.6.122.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAA8toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

9.4.3.6.122.3 Fields

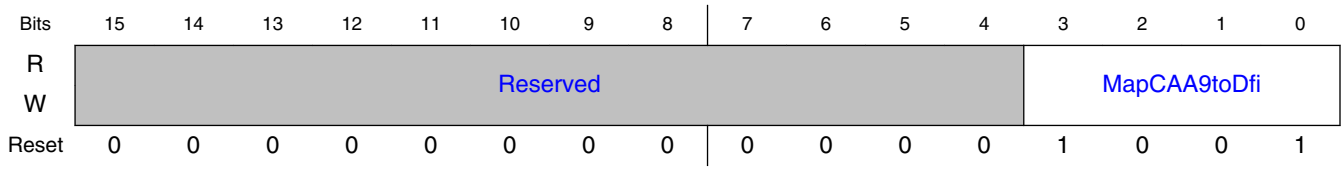
Field	Function
15-4 —	Reserved
3-0 MapCAA8toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 8. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 8. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA8 is mapped from dfi0_address[2], then Register MapCAA8toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.123 Maps PHY CAA lane 9 from dfi0_address of the index of the register contents (MapCAA9toDfi)

9.4.3.6.123.1 Offset

Register	Offset
MapCAA9toDfi	212h

9.4.3.6.123.2 Diagram



9.4.3.6.123.3 Fields

Field	Function
15-4	Reserved

Table continues on the next page...

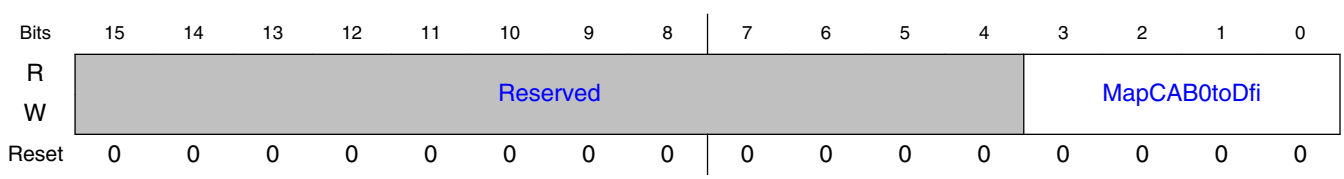
Field	Function
—	
3-0 MapCAA9toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 9. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi0_address to CAA 9. Unused for DDR3 and unused for DDR4 applications. Each register in the set of MapCAAtoDfi must have a unique value within the set. For example, if PHY CAA9 is mapped from dfi0_address[2], then Register MapCAA9toDfi should be 2. LP4 ACX3,ACX6,ACX10,ACX12 registers MapCAA0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAA6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6,ACX10,ACX12 registers MapCAA0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.124 Maps PHY CAB lane 0 from dfi1_address of the index of the register contents (MapCAB0toDfi)

9.4.3.6.124.1 Offset

Register	Offset
MapCAB0toDfi	220h

9.4.3.6.124.2 Diagram



9.4.3.6.124.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB0toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 0. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 0.

Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB0 is mapped from dfi1_address[2], then Register MapCAB0toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.125 Maps PHY CAB lane 1 from dfi1_address of the index of the register contents (MapCAB1toDfi)

9.4.3.6.125.1 Offset

Register	Offset
MapCAB1toDfi	222h

9.4.3.6.125.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAB1toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

9.4.3.6.125.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB1toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 1. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 1.

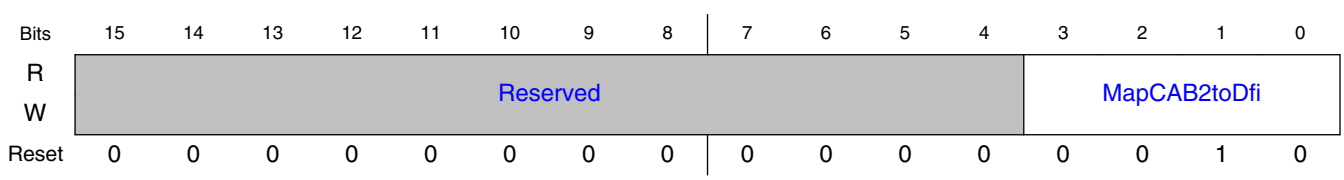
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB1 is mapped from dfi1_address[2], then Register MapCAB1toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.126 Maps PHY CAB lane 2 from dfi1_address of the index of the register contents (MapCAB2toDfi)

9.4.3.6.126.1 Offset

Register	Offset
MapCAB2toDfi	224h

9.4.3.6.126.2 Diagram



9.4.3.6.126.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB2toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 2. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 2.

Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB2 is mapped from dfi1_address[2], then Register MapCAB2toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.127 Maps PHY CAB lane 3 from dfi1_address of the index of the register contents (MapCAB3toDfi)

9.4.3.6.127.1 Offset

Register	Offset
MapCAB3toDfi	226h

9.4.3.6.127.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAB3toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

9.4.3.6.127.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB3toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 3. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 3.

Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB3 is mapped from dfi1_address[2], then Register MapCAB3toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.128 Maps PHY CAB lane 4 from dfi1_address of the index of the register contents (MapCAB4toDfi)

9.4.3.6.128.1 Offset

Register	Offset
MapCAB4toDfi	228h

9.4.3.6.128.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAB4toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

9.4.3.6.128.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB4toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 4. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 4.

Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB4 is mapped from dfi1_address[2], then Register MapCAB4toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.129 Maps PHY CAB lane 5 from dfi1_address of the index of the register contents (MapCAB5toDfi)

9.4.3.6.129.1 Offset

Register	Offset
MapCAB5toDfi	22Ah

9.4.3.6.129.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved												MapCAB5toDfi			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

9.4.3.6.129.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB5toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 5. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 5.

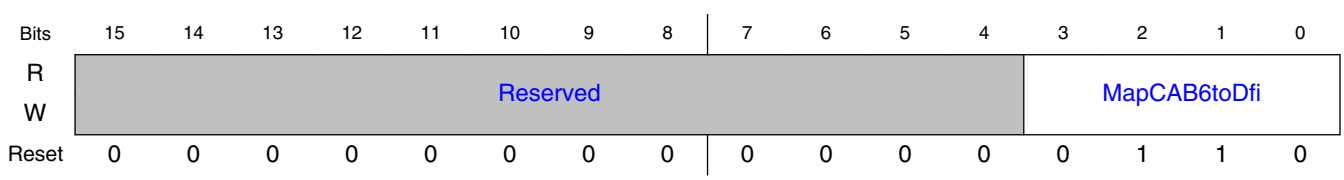
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB5 is mapped from dfi1_address[2], then Register MapCAB5toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.130 Maps PHY CAB lane 6 from dfi1_address of the index of the register contents (MapCAB6toDfi)

9.4.3.6.130.1 Offset

Register	Offset
MapCAB6toDfi	22Ch

9.4.3.6.130.2 Diagram



9.4.3.6.130.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB6toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 6. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 6.

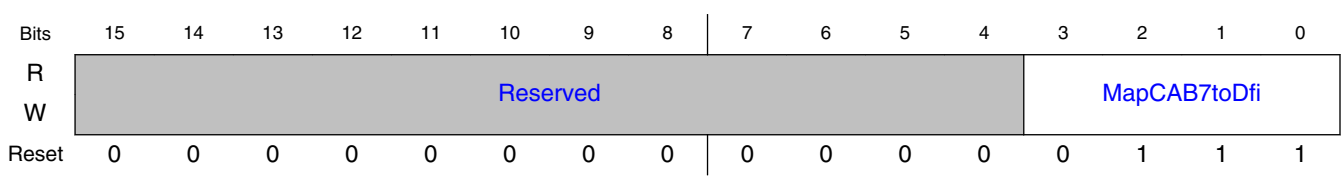
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB6 is mapped from dfi1_address[2], then Register MapCAB6toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.131 Maps PHY CAB lane 7 from dfi1_address of the index of the register contents (MapCAB7toDfi)

9.4.3.6.131.1 Offset

Register	Offset
MapCAB7toDfi	22Eh

9.4.3.6.131.2 Diagram



9.4.3.6.131.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB7toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 7. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 7.

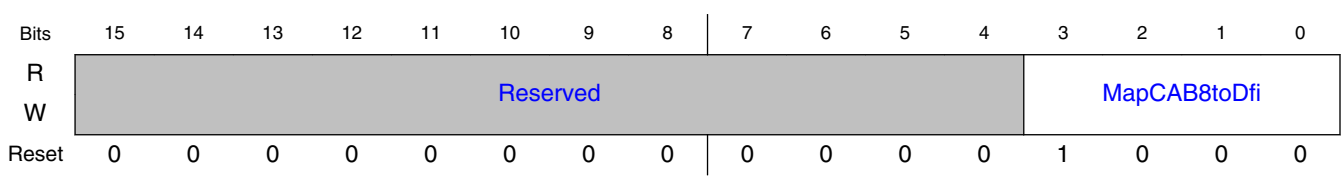
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB7 is mapped from dfi1_address[2], then Register MapCAB7toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.132 Maps PHY CAB lane 8 from dfi1_address of the index of the register contents (MapCAB8toDfi)

9.4.3.6.132.1 Offset

Register	Offset
MapCAB8toDfi	230h

9.4.3.6.132.2 Diagram



9.4.3.6.132.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB8toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 8. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 8.

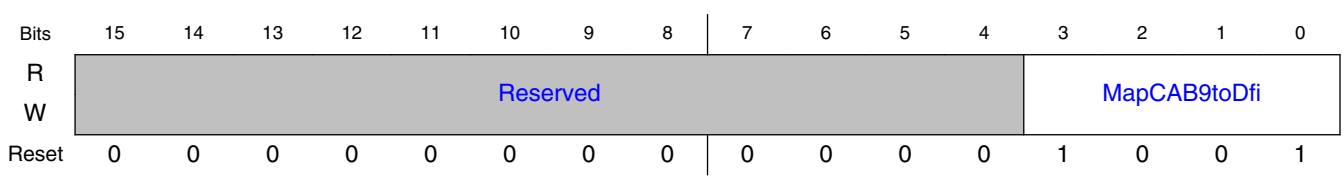
Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB8 is mapped from dfi1_address[2], then Register MapCAB8toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.133 Maps PHY CAB lane 9 from dfi1_address of the index of the register contents (MapCAB9toDfi)

9.4.3.6.133.1 Offset

Register	Offset
MapCAB9toDfi	232h

9.4.3.6.133.2 Diagram



9.4.3.6.133.3 Fields

Field	Function
15-4 —	Reserved
3-0 MapCAB9toDfi	For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 9. For LPDDR3 and LPDDR4 applications, these CSRs map a dfi1_address to CAB 9.

Field	Function
	Unused for DDR3 and unused for DDR4 applications. Unused for the 3-ANIB configuration. Each register in the set of MapCABtoDfi must have a unique value within the set. For example, if PHY CAB9 is mapped from dfi1_address[2], then Register MapCAB9toDfi should be 2. LP4 ACX3 registers MapCAB0..9toDfi are reserved, second channel not supportd LP4 ACX6,ACX10,ACX12 registers MapCAB0..5toDfi are used, values 0-5 are valid, values 6-15 are reserved registers MapCAB6..9toDfi are reserved LP3 ACX3 not supported LP3 ACX6 registers MapCAB0..9 are unused, second channel not supported LP3 ACX10,ACX12 registers MapCAB0..9toDfi are used, values 0-9 are valid, values 10-15 are reserved

9.4.3.6.134 Interrupt Enable Bits (PhyInterruptEnable)

9.4.3.6.134.1 Offset

Register	Offset
PhyInterruptEnable	236h

9.4.3.6.134.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PhyHWReservedEn					PhyRxFifoCheckEn	PhyVTDriftAlarmEn		PhyFWReservedEn					PhyTrngFailEn	PhyInitCmpltEn	PhyTrngCmpltEn
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.134.3 Fields

Field	Function
15-11	Reserved

Table continues on the next page...

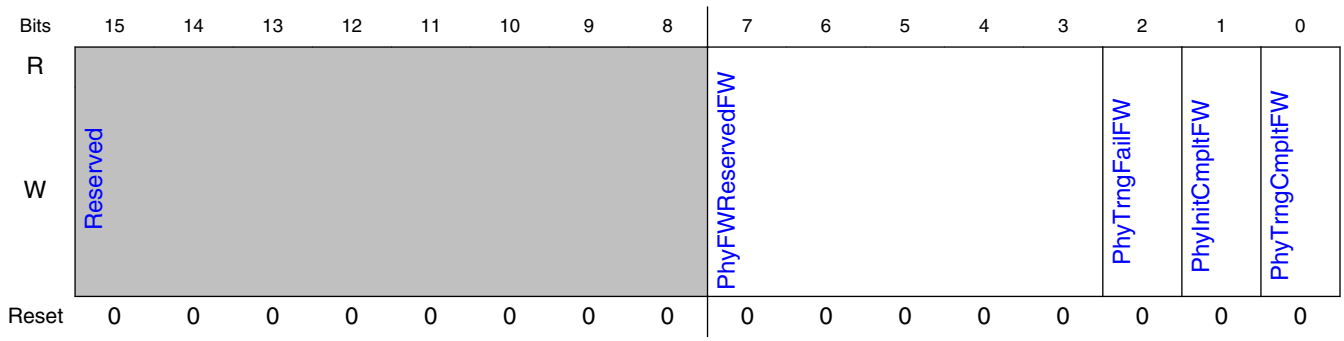
Field	Function
PhyHWReservedEn	
10 PhyRxFifoCheckEn	Enable for the RxFifo Pointers Check Interrupt 0 : Interrupt not enabled 1 : Interrupt enabled
9-8 PhyVTDriftAlarmEn	Enable for the PHY VT Drift Alarm interrupts. Enable for the PHY VT Drift Alarm interrupts. 0 : Interrupt not enabled 1 : Interrupt enabled
7-3 PhyFWReservedEn	Reserved
2 PhyTrngFailEn	Enable for the PHY Training Failure interrupt. Enable for the PHY Training Failure interrupt. 0 : Interrupt not enabled 1 : Interrupt enabled
1 PhyInitCmpltEn	Enable for the PHY Initialization Complete interrupt. Enable for the PHY Initialization Complete interrupt. 0 : Interrupt not enabled 1 : Interrupt enabled
0 PhyTrngCmpltEn	Enable for the PHY Training Complete interrupt. Enable for the PHY Training Complete interrupt. 0 : Interrupt not enabled 1 : Interrupt enabled

9.4.3.6.135 Interrupt Firmware Control Bits (PhyInterruptFWControl)

9.4.3.6.135.1 Offset

Register	Offset
PhyInterruptFWControl	238h

9.4.3.6.135.2 Diagram



9.4.3.6.135.3 Fields

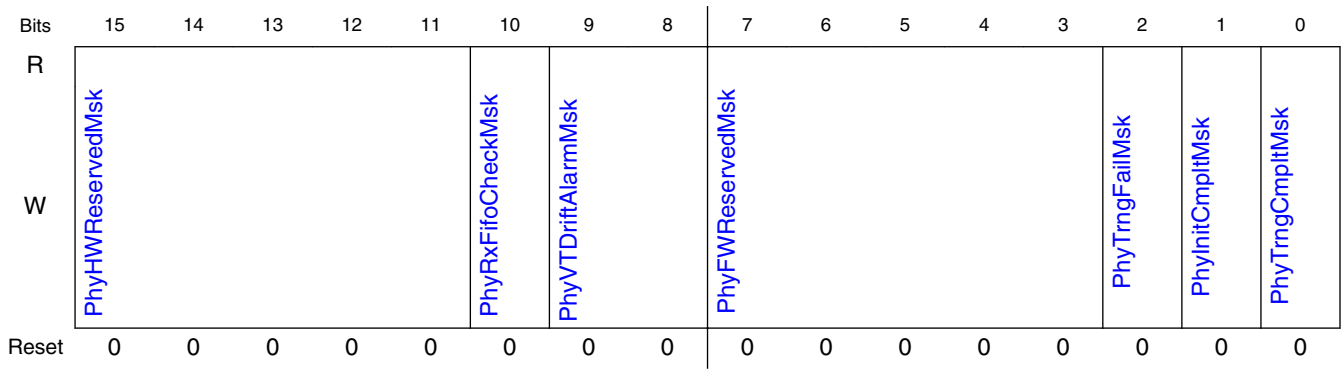
Field	Function
15-8 —	Reserved
7-3 PhyFWReservedFW	Reserved
2 PhyTrngFailFW	PHY Training Failure Firmware interrupt. PHY Training Failure Firmware interrupt. 0 : Interrupt not asserted 1 : Interrupt asserted
1 PhyInitCmpltFW	PHY Initialization Complete Firmware interrupt. PHY Initialization Complete Firmware interrupt. 0 : Interrupt not asserted 1 : Interrupt asserted
0 PhyTrngCmpltFW	PHY Training Complete Firmware interrupt. PHY Training Complete Firmware interrupt. 0 : Interrupt not asserted 1 : Interrupt asserted

9.4.3.6.136 Interrupt Mask Bits (PhyInterruptMask)

9.4.3.6.136.1 Offset

Register	Offset
PhyInterruptMask	23Ah

9.4.3.6.136.2 Diagram



9.4.3.6.136.3 Fields

Field	Function
15-11 PhyHWReservedMsk	Reserved
10 PhyRxFifoCheckMsk	Mask for the RxFifo Pointers Check Interrupt 0 : Interrupt not masked 1 : Interrupt masked
9-8 PhyVTDriftAlarmMsk	Mask for the PHY VT Drift Alarm interrupts. Mask for the PHY VT Drift Alarm interrupts. 0 : Interrupt not masked 1 : Interrupt masked
7-3 PhyFWReservedMsk	Reserved
2 PhyTrngFailMsk	Mask for the PHY Training Failure interrupt. Mask for the PHY Training Failure interrupt. 0 : Interrupt not masked 1 : Interrupt masked
1 PhyInitCmpltMsk	Mask for the PHY Initialization Complete interrupt. Mask for the PHY Initialization Complete interrupt. 0 : Interrupt not masked 1 : Interrupt masked
0 PhyTrngCmpltMsk	Mask for the PHY Training Complete interrupt. Mask for the PHY Training Complete interrupt. 0 : Interrupt not masked 1 : Interrupt masked

9.4.3.6.137 Interrupt Clear Bits (PhyInterruptClear)

9.4.3.6.137.1 Offset

Register	Offset
PhyInterruptClear	23Ch

9.4.3.6.137.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	PhyHWReservedClr					PhyRxFifoCheckClr	PhyVTDriftAlarmClr		PhyFWReservedClr					PhyTrngFailClr	PhyInitCmpltClr	PhyTrngCmpltClr
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.137.3 Fields

Field	Function
15-11 PhyHWReservedClr	Reserved
10 PhyRxFifoCheckClr	Clear for the RxFifo Pointers Check Interrupt 0 : Interrupt not affected 1 : Interrupt cleared
9-8 PhyVTDriftAlarmClr	Clear for the PHY VT Drift Alarm interrupt. Clear for the PHY VT Drift Alarm interrupt. 0 : Interrupt not affected 1 : Interrupt cleared
7-3 PhyFWReservedClr	Reserved
2 PhyTrngFailClr	Clear for the PHY Training Failure interrupt. Clear for the PHY Training Failure interrupt. 0 : Interrupt not affected 1 : Interrupt cleared

Table continues on the next page...

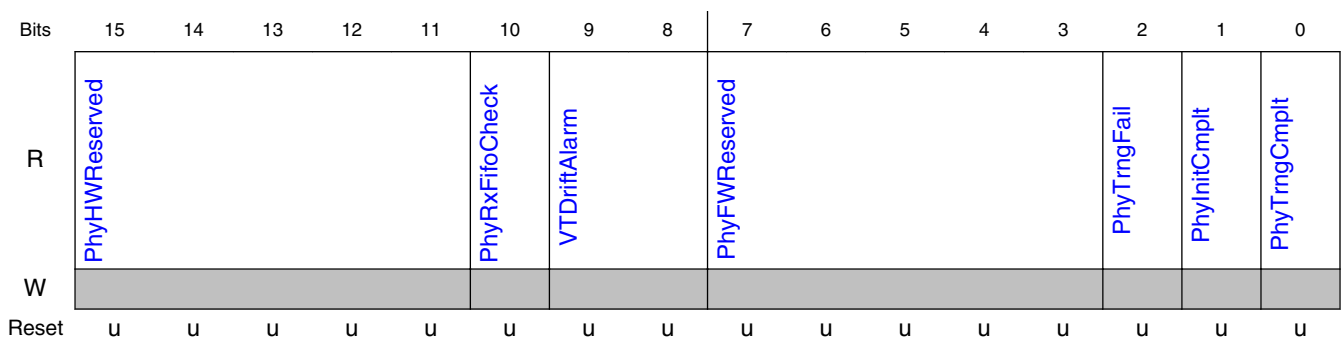
Field	Function
1 PhyInitCmpltClr	Clear for the PHY Initialization Complete interrupt. Clear for the PHY Initialization Complete interrupt. 0 : Interrupt not affected 1 : Interrupt cleared
0 PhyTrngCmpltClr	Clear for the PHY Training Complete interrupt. Clear for the PHY Training Complete interrupt. 0 : Interrupt not affected 1 : Interrupt cleared

9.4.3.6.138 Interrupt Status Bits (PhyInterruptStatus)

9.4.3.6.138.1 Offset

Register	Offset
PhyInterruptStatus	23Eh

9.4.3.6.138.2 Diagram



9.4.3.6.138.3 Fields

Field	Function
15-11 PhyHWReserved	Reserved
10 PhyRxFifoCheck	A mechanism in the PHY checks the Read Fifo pointers for consistency at times they are idle. A mechanism in the PHY checks the Read Fifo pointers for consistency at times they are idle. If a check fails, this interrupt is asserted, and detailed status is found in the RxFifoCheckStatus register.

Table continues on the next page...

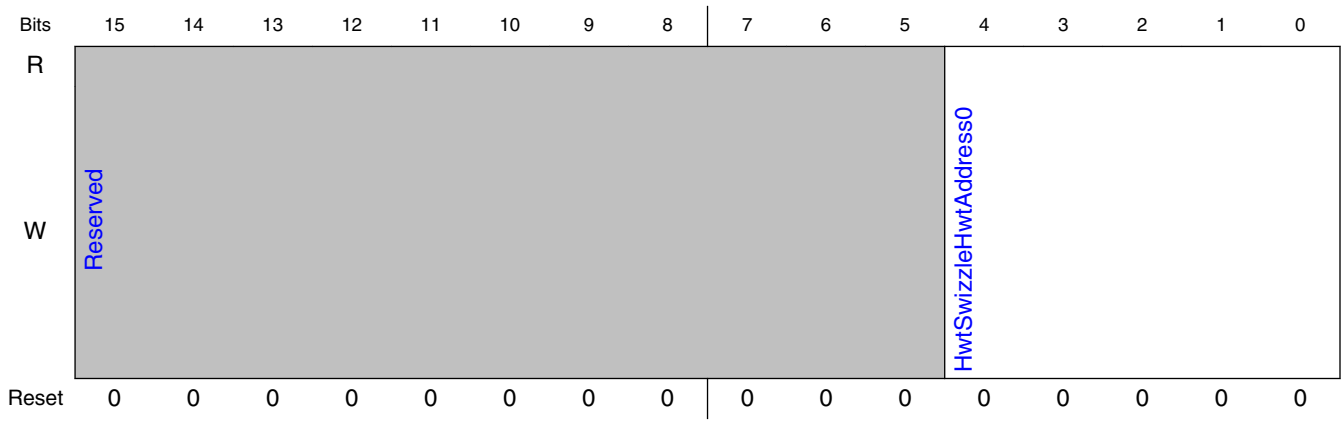
Field	Function
	0 : Interrupt not enabled and asserted 1 : Interrupt enabled and asserted
9-8 VTDriftAlarm	PHY VT Drift Alarm interrupt. PHY VT Drift Alarm interrupt. Assertion of this interrupt indicates the number of tap points in the master DLL has shifted by an amount greater than 2 * DfiPhyUpdIntThreshold. There is one bit for each master DLL and they are fully independent of one another. Please see the DfiPhyUpdIntThreshold CSR for additional information. 0 : Interrupt not enabled and asserted 1 : Interrupt enabled and asserted
7-3 PhyFWReserved	Reserved
2 PhyTrngFail	PHY Training Failure interrupt. PHY Training Failure interrupt. Assertion of this interrupt indicates there was an issue during training which did not allow the PHY to successfully complete training. 0 : Interrupt not enabled and asserted 1 : Interrupt enabled and asserted
1 PhyInitCmplt	PHY Initialization Complete interrupt. PHY Initialization Complete interrupt. Assertion of this interrupt indicates that the PHY Initialization Engine has been fully loaded and the PHY is ready and able to assert dfi_init_complete following the assertion of dfi_init_start. It is possible this interrupt may not assert until after dfi_init_complete has asserted, depending on when dfi_init_start asserts. 0 : Interrupt not enabled and asserted 1 : Interrupt enabled and asserted
0 PhyTrngCmplt	PHY Training Complete interrupt. PHY Training Complete interrupt. Assertion of this interrupt indicates that the PHY has successfully trained the memory interface at all indicated frequencies and is ready for the PHY Initialization Engine to be loaded. 0 : Interrupt not enabled and asserted 1 : Interrupt enabled and asserted

9.4.3.6.139 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress0)

9.4.3.6.139.1 Offset

Register	Offset
HwtSwizzleHwtAddress0	240h

9.4.3.6.139.2 Diagram



9.4.3.6.139.3 Fields

Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress0	This set of registers is used in DDR3/DDR4 mode where a user has re-mapped the DFI inputs to the PHY. This set of registers is used in DDR3/DDR4 mode where a user has re-mapped the DFI inputs to the PHY. This register does not affect the mapping of those signals, or remap the tristate properties of the bumps. However, when the user remaps the DFI signals, the HWT inputs must be mapped in a similar fashion using this register. Each remappable hwt bit has a register associated with it. When the register value is zero, the bit is not re-mapped. When the register value is non-zero, the signal associated with the number in the table below is mapped to the signal associated with the register. So, for example, if you set the value for the register associated with hwt_act_n (HwtSwizzleHwtActN) to 27, then hwt_parity_in will be mapped to hwt_act_n. These registers should always be 0 for LPDDR3 and LPDDR4. Some registers are used only for DDR3 or DDR4. These are noted in the descriptions. Note: If the system uses DDR3-RDIMM or

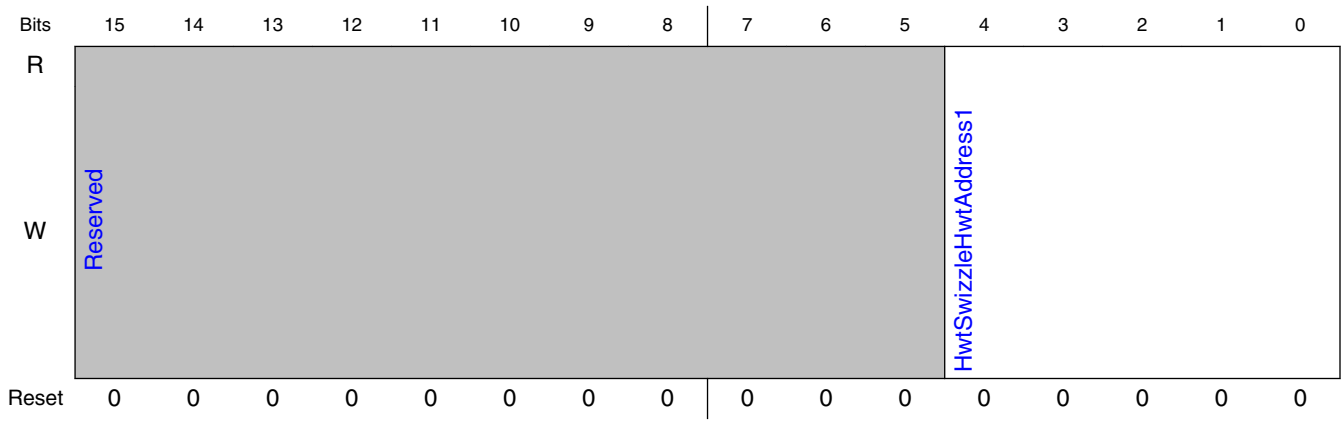
Field	Function
	DDR4-RDIMM/LRDIMM, it is strongly recommended NOT to swizzle the PAR pin (parity). If the PAR pin is swizzled, the Dynamic Address Tristate feature must be disabled. (see the DisDynAdrTri field of the TristateMode CA Register). ----- Index: Signal 0: No-remapping 1: hwt_address[0] 2: hwt_address[1] 3: hwt_address[2] 4: hwt_address[3] 5: hwt_address[4] 6: hwt_address[5] 7: hwt_address[6] 8: hwt_address[7] 9: hwt_address[8] 10: hwt_address[9] 11: hwt_address[10] 12: hwt_address[11] 13: hwt_address[12] 14: hwt_address[13] 15: hwt_address[14] (DDR3 Only) 16: hwt_address[15] (DDR3 Only) 17: hwt_address[17] (DDR4 Only) 18: hwt_act_n (DDR4 Only) 19: hwt_bank[0] 20: hwt_bank[1] 21: hwt_bank[2] (DDR3 Only) 22: hwt_bg[0] (DDR4 Only) 23: hwt_bg[1] (DDR4 Only) 24: hwt_cas_n 25: hwt_ras_n 26: hwt_we_n 27: hwt_parity_in

9.4.3.6.140 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress1)

9.4.3.6.140.1 Offset

Register	Offset
HwtSwizzleHwtAddress1	242h

9.4.3.6.140.2 Diagram



9.4.3.6.140.3 Fields

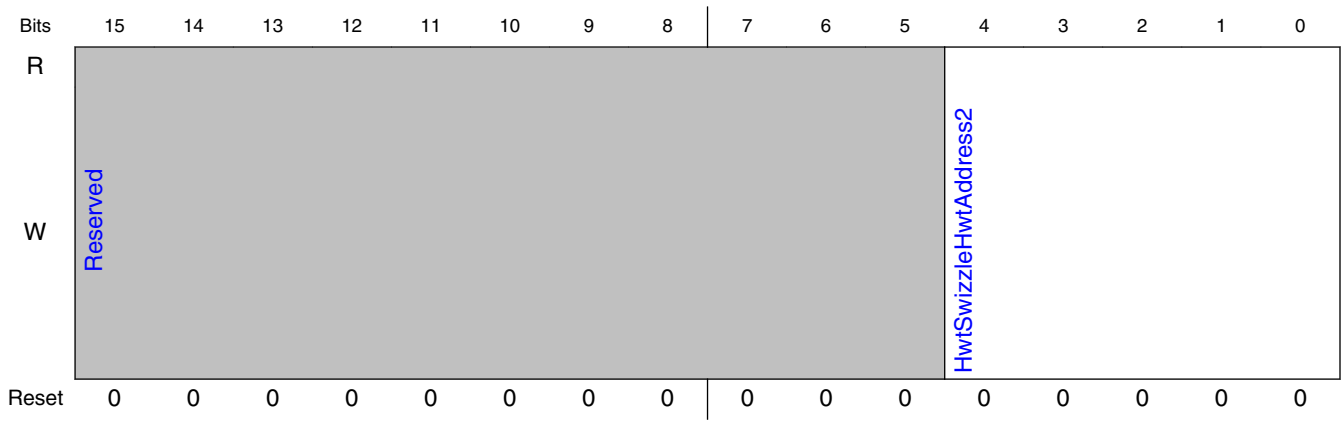
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress1	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.141 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress2)

9.4.3.6.141.1 Offset

Register	Offset
HwtSwizzleHwtAddress2	244h

9.4.3.6.141.2 Diagram



9.4.3.6.141.3 Fields

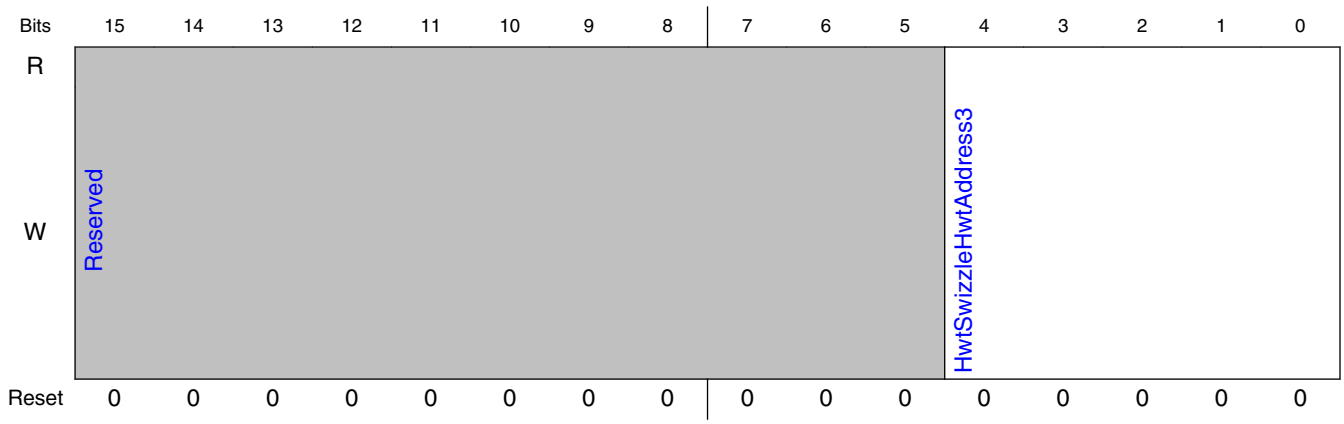
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress2	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.142 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress3)

9.4.3.6.142.1 Offset

Register	Offset
HwtSwizzleHwtAddress3	246h

9.4.3.6.142.2 Diagram



9.4.3.6.142.3 Fields

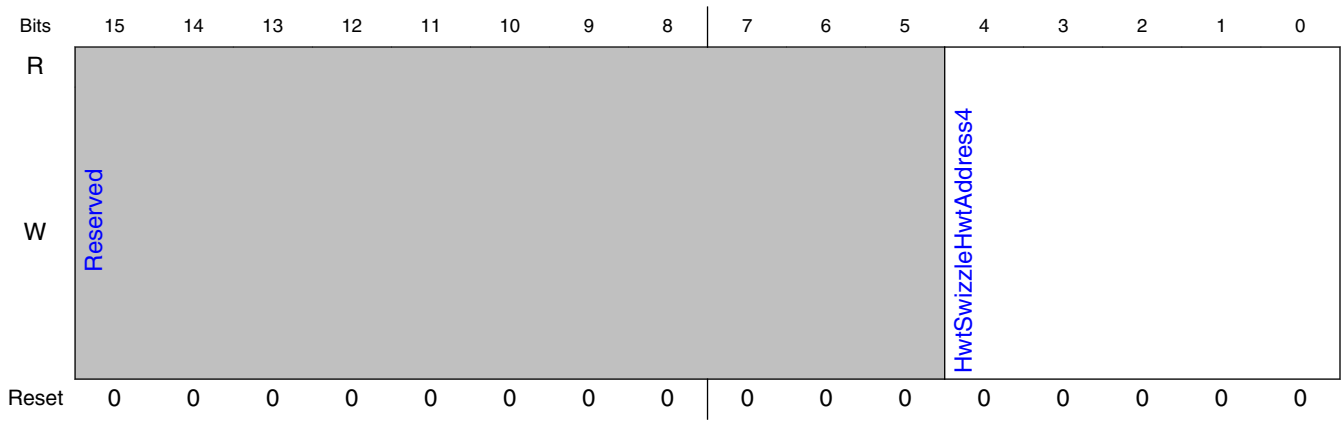
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress3	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.143 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress4)

9.4.3.6.143.1 Offset

Register	Offset
HwtSwizzleHwtAddress4	248h

9.4.3.6.143.2 Diagram



9.4.3.6.143.3 Fields

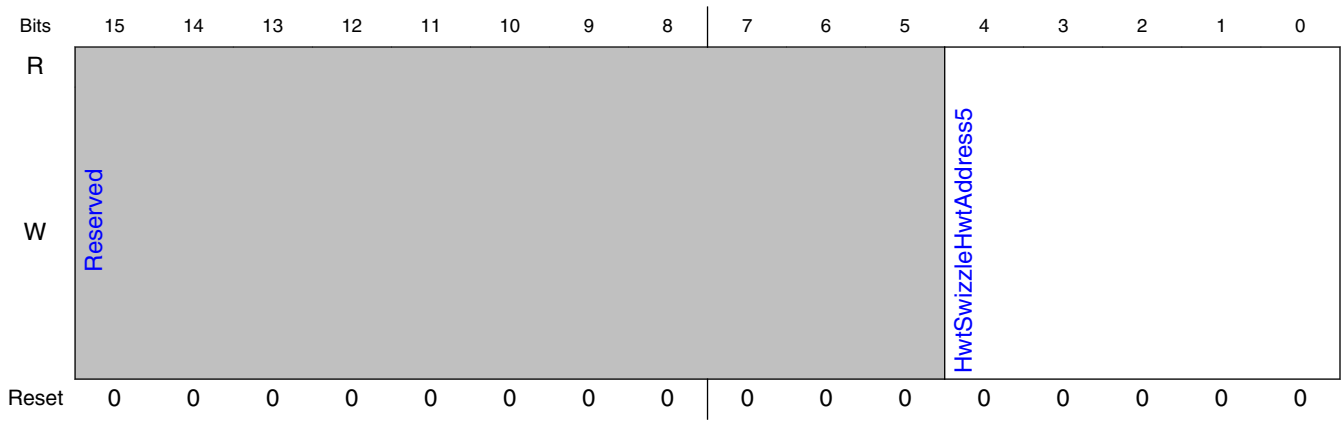
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress4	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.144 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress5)

9.4.3.6.144.1 Offset

Register	Offset
HwtSwizzleHwtAddress5	24Ah

9.4.3.6.144.2 Diagram



9.4.3.6.144.3 Fields

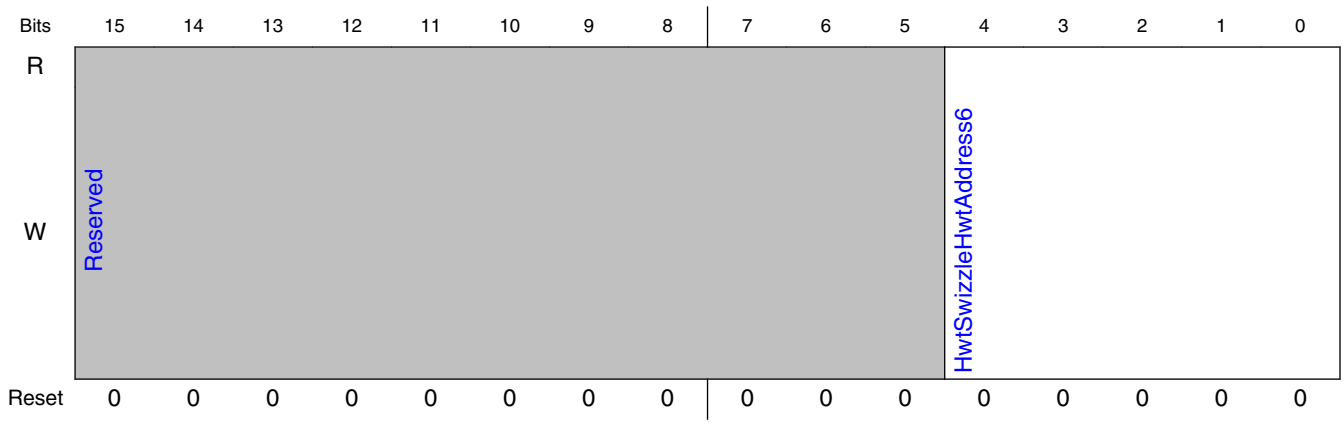
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress5	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.145 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress6)

9.4.3.6.145.1 Offset

Register	Offset
HwtSwizzleHwtAddress6	24Ch

9.4.3.6.145.2 Diagram



9.4.3.6.145.3 Fields

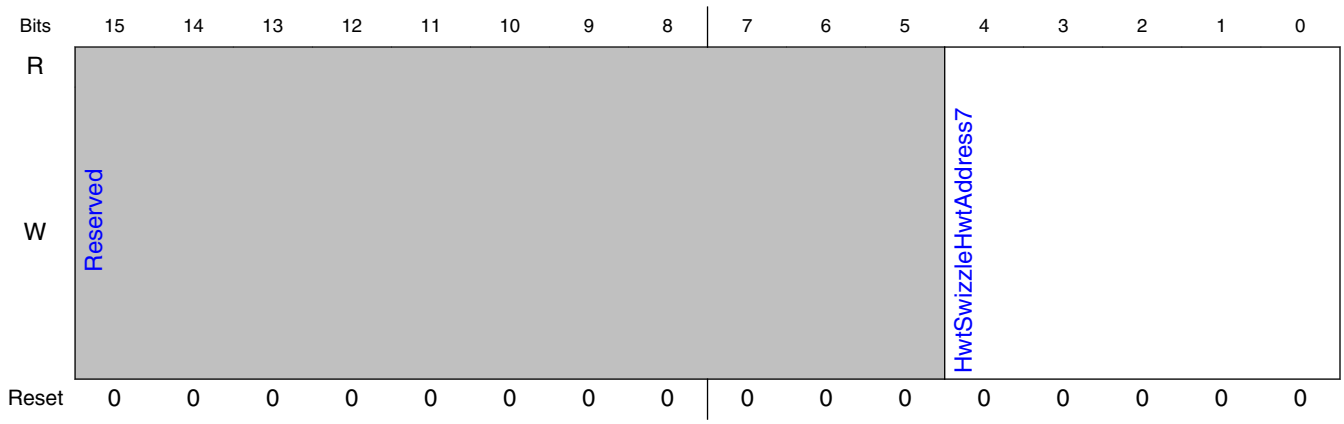
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress6	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.146 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress7)

9.4.3.6.146.1 Offset

Register	Offset
HwtSwizzleHwtAddress7	24Eh

9.4.3.6.146.2 Diagram



9.4.3.6.146.3 Fields

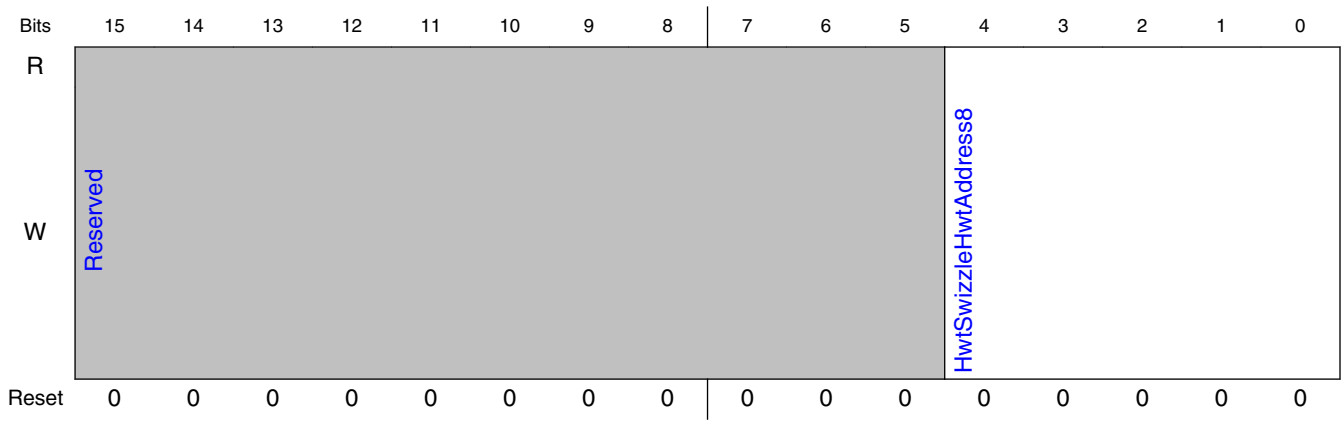
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress7	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.147 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress8)

9.4.3.6.147.1 Offset

Register	Offset
HwtSwizzleHwtAddress8	250h

9.4.3.6.147.2 Diagram



9.4.3.6.147.3 Fields

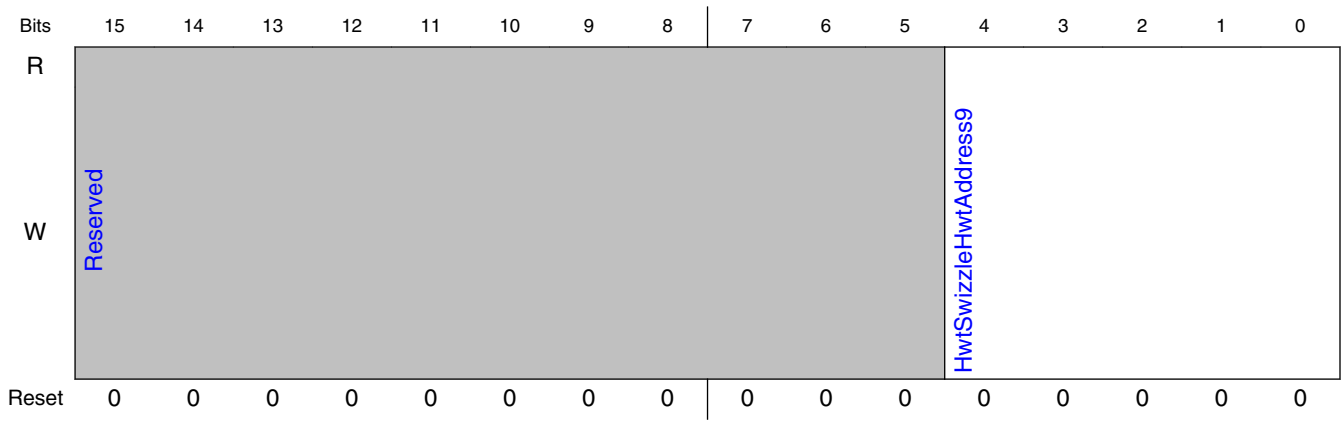
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress8	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.148 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress9)

9.4.3.6.148.1 Offset

Register	Offset
HwtSwizzleHwtAddress9	252h

9.4.3.6.148.2 Diagram



9.4.3.6.148.3 Fields

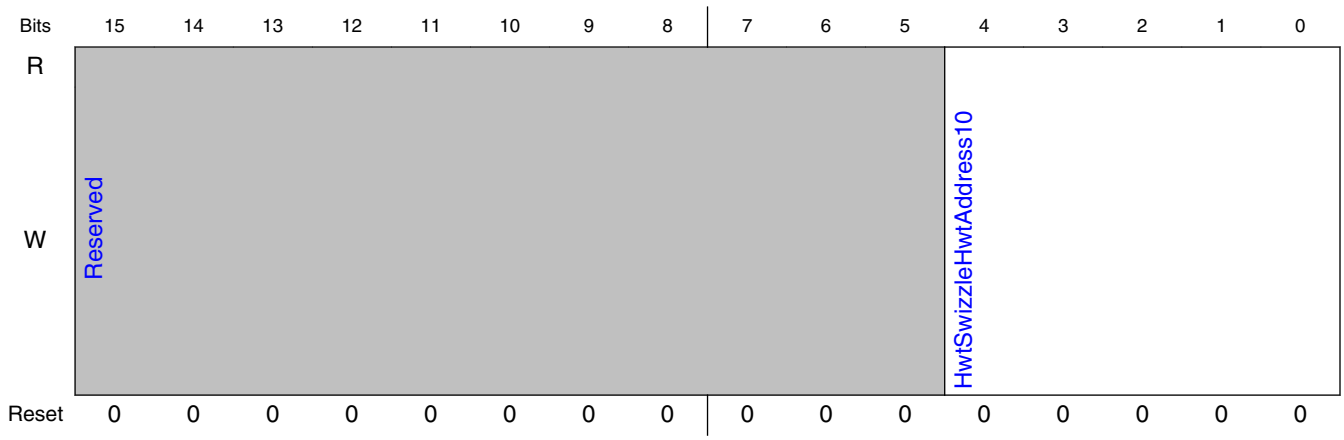
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress9	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.149 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress10)

9.4.3.6.149.1 Offset

Register	Offset
HwtSwizzleHwtAddress10	254h

9.4.3.6.149.2 Diagram



9.4.3.6.149.3 Fields

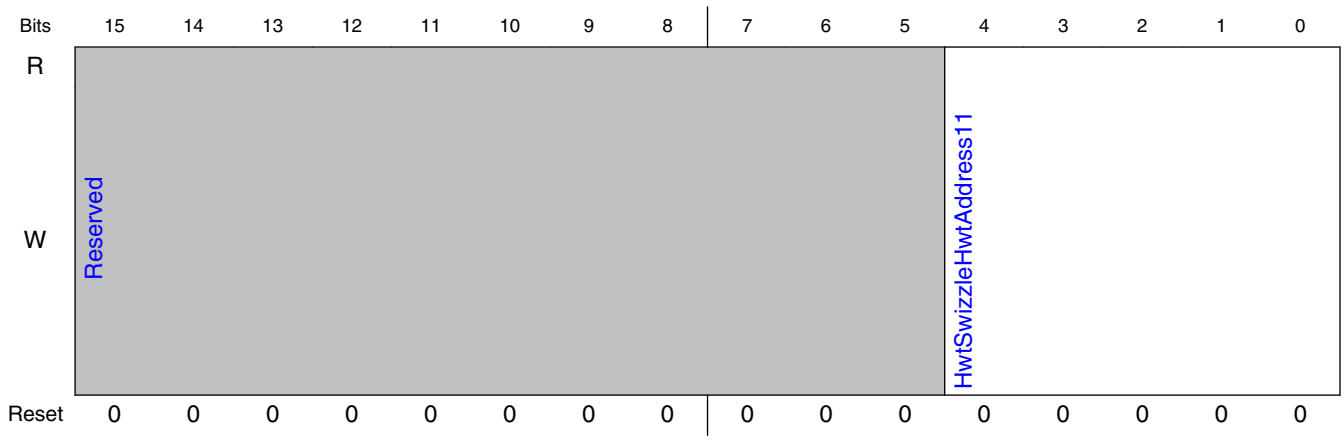
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress10	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.150 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress11)

9.4.3.6.150.1 Offset

Register	Offset
HwtSwizzleHwtAddress11	256h

9.4.3.6.150.2 Diagram



9.4.3.6.150.3 Fields

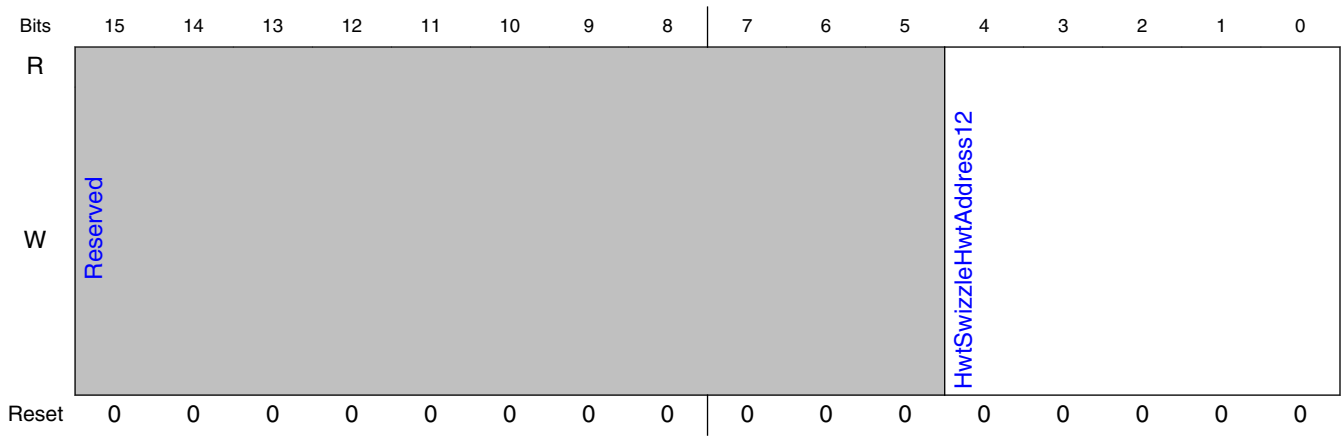
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress11	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.151 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress12)

9.4.3.6.151.1 Offset

Register	Offset
HwtSwizzleHwtAddress12	258h

9.4.3.6.151.2 Diagram



9.4.3.6.151.3 Fields

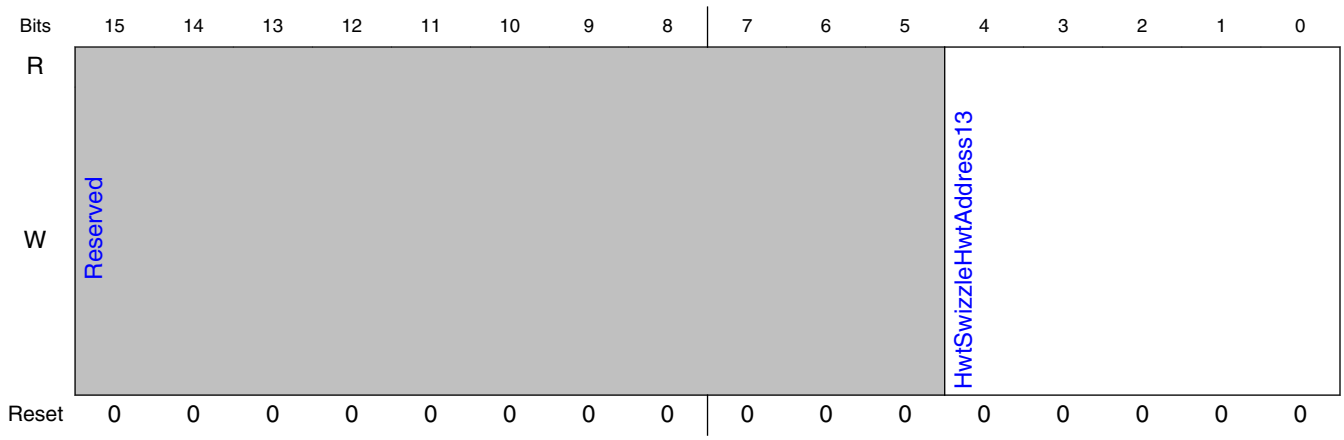
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress12	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.152 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress13)

9.4.3.6.152.1 Offset

Register	Offset
HwtSwizzleHwtAddress13	25Ah

9.4.3.6.152.2 Diagram



9.4.3.6.152.3 Fields

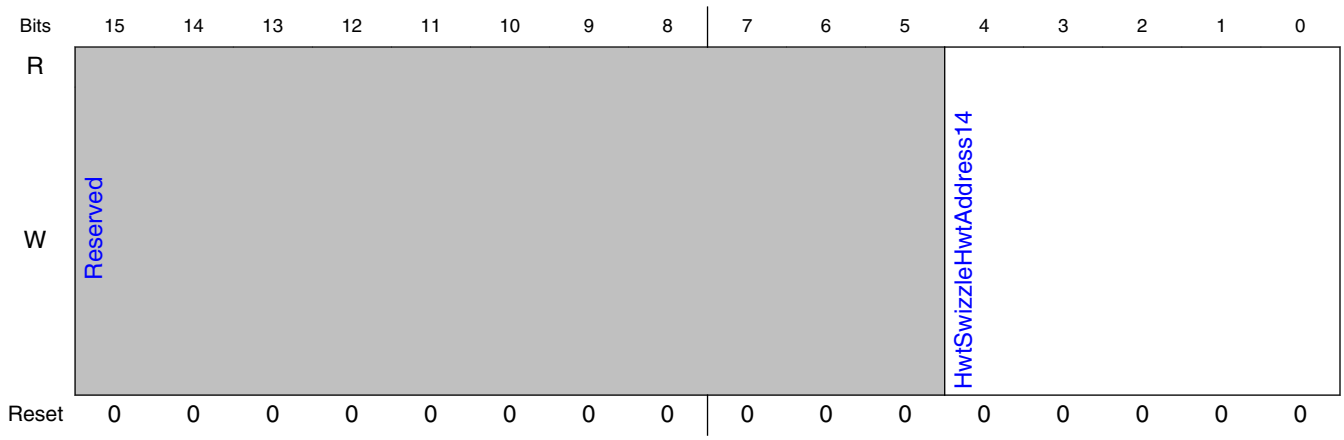
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress13	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.153 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress14)

9.4.3.6.153.1 Offset

Register	Offset
HwtSwizzleHwtAddress14	25Ch

9.4.3.6.153.2 Diagram



9.4.3.6.153.3 Fields

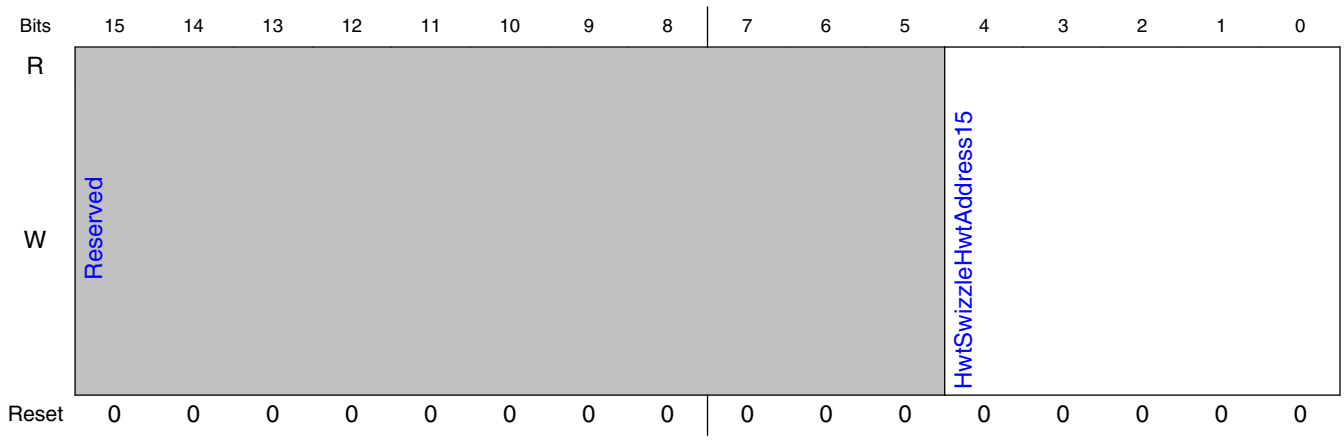
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress14	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR3.

9.4.3.6.154 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress15)

9.4.3.6.154.1 Offset

Register	Offset
HwtSwizzleHwtAddress15	25Eh

9.4.3.6.154.2 Diagram



9.4.3.6.154.3 Fields

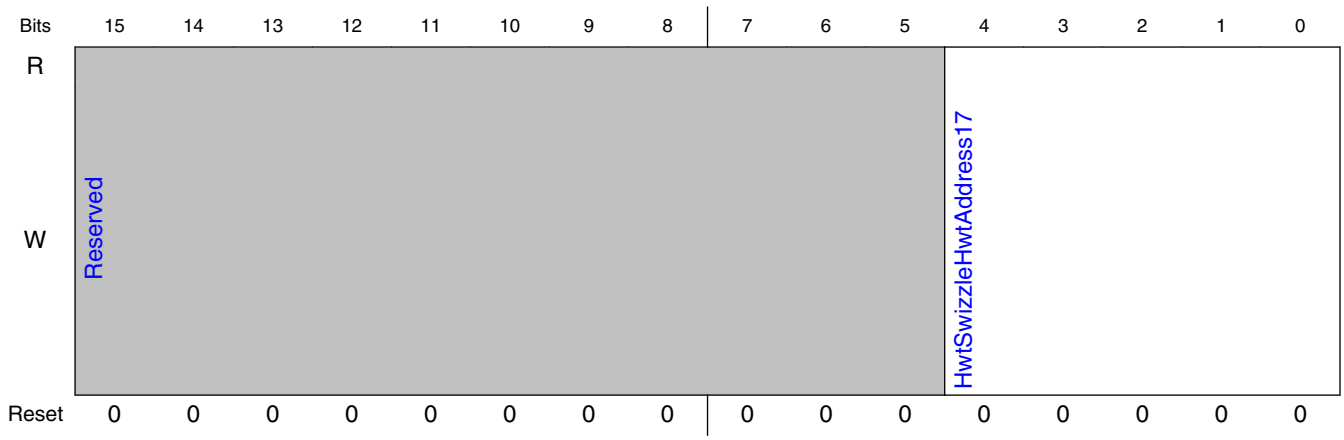
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress15	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR3.

9.4.3.6.155 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtAddress17)

9.4.3.6.155.1 Offset

Register	Offset
HwtSwizzleHwtAddress17	260h

9.4.3.6.155.2 Diagram



9.4.3.6.155.3 Fields

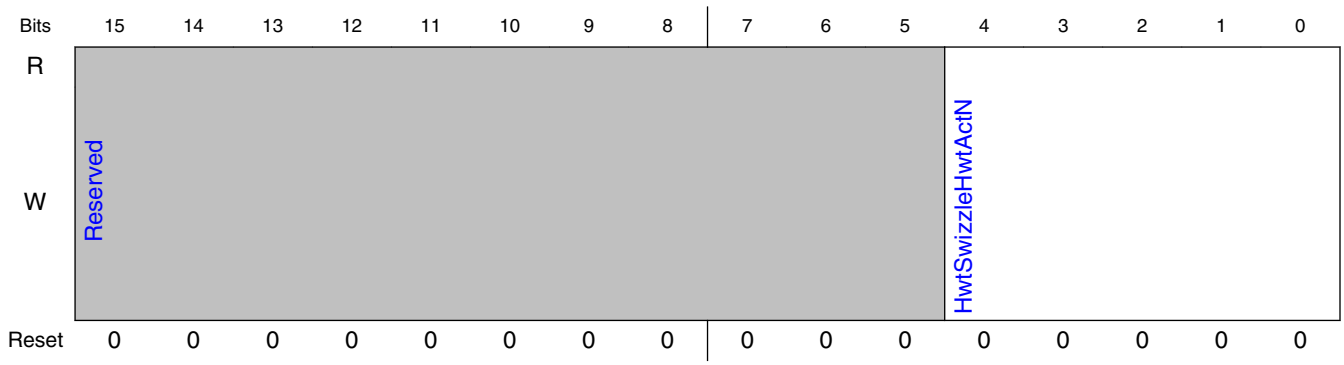
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtAddress17	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR4.

9.4.3.6.156 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtActN)

9.4.3.6.156.1 Offset

Register	Offset
HwtSwizzleHwtActN	262h

9.4.3.6.156.2 Diagram



9.4.3.6.156.3 Fields

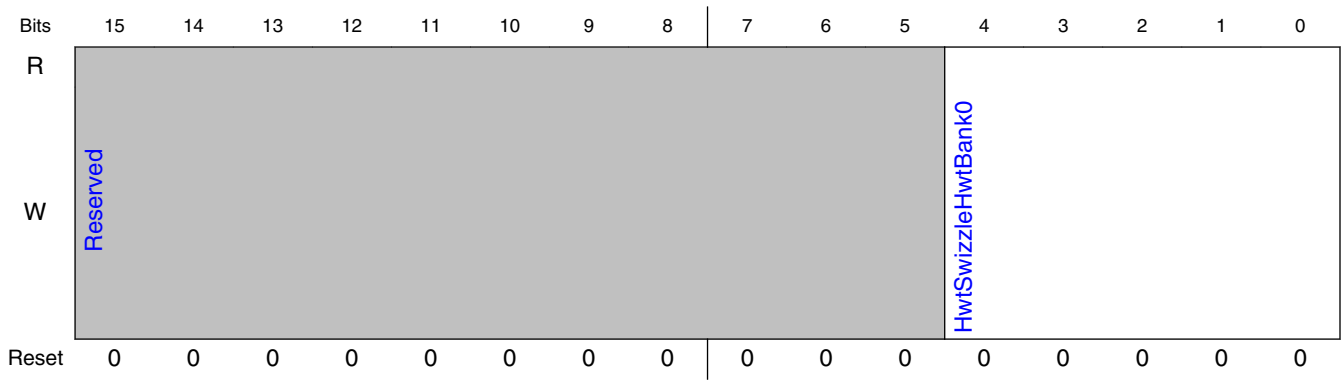
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtActN	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR4.

9.4.3.6.157 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank0)

9.4.3.6.157.1 Offset

Register	Offset
HwtSwizzleHwtBank0	264h

9.4.3.6.157.2 Diagram



9.4.3.6.157.3 Fields

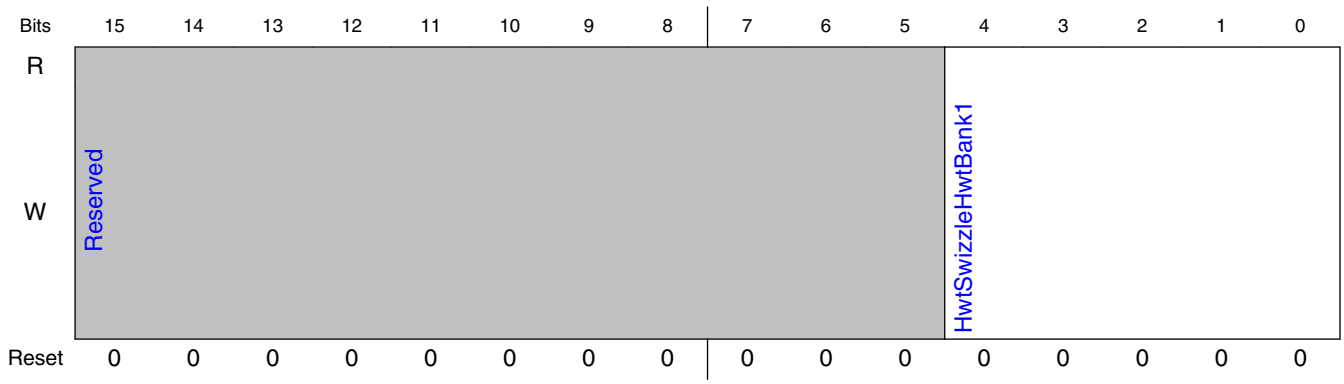
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtBank0	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.158 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank1)

9.4.3.6.158.1 Offset

Register	Offset
HwtSwizzleHwtBank1	266h

9.4.3.6.158.2 Diagram



9.4.3.6.158.3 Fields

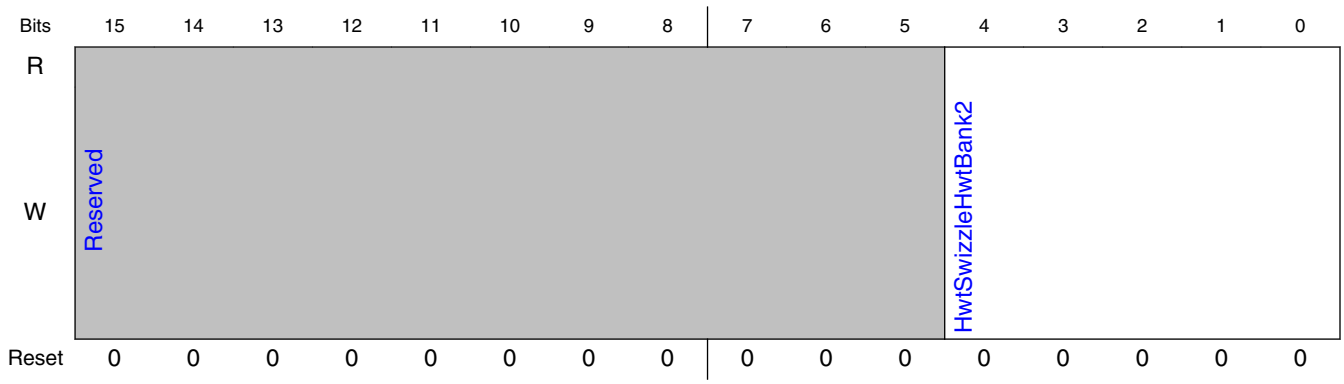
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtBank1	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.159 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBank2)

9.4.3.6.159.1 Offset

Register	Offset
HwtSwizzleHwtBank2	268h

9.4.3.6.159.2 Diagram



9.4.3.6.159.3 Fields

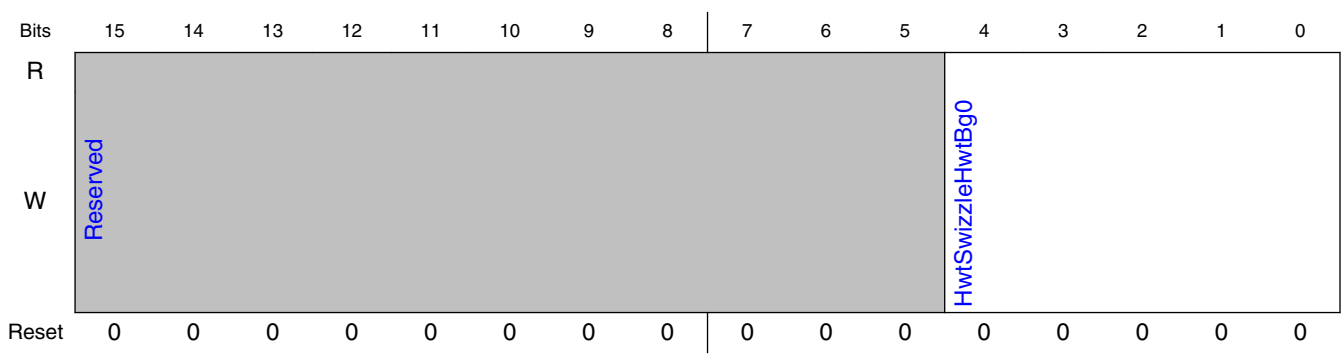
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtBank2	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR3.

9.4.3.6.160 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBg0)

9.4.3.6.160.1 Offset

Register	Offset
HwtSwizzleHwtBg0	26Ah

9.4.3.6.160.2 Diagram



9.4.3.6.160.3 Fields

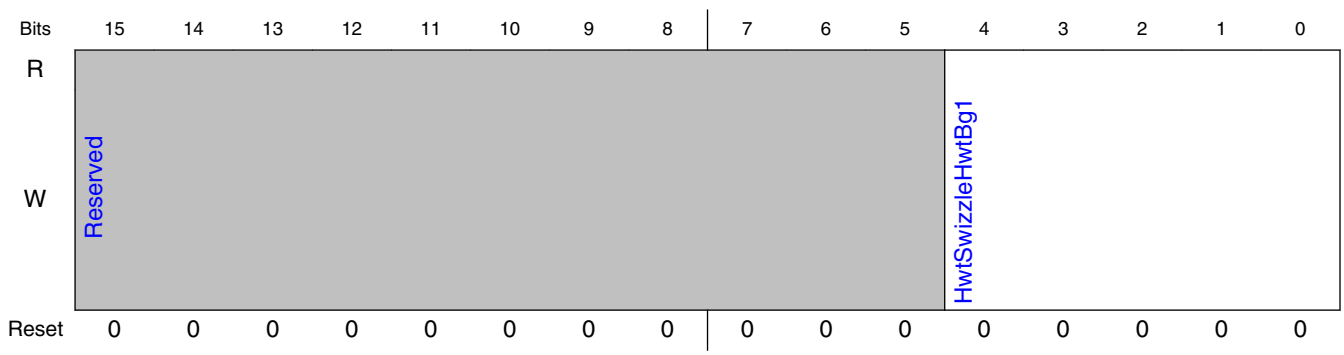
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtBg0	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR4.

9.4.3.6.161 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtBg1)

9.4.3.6.161.1 Offset

Register	Offset
HwtSwizzleHwtBg1	26Ch

9.4.3.6.161.2 Diagram



9.4.3.6.161.3 Fields

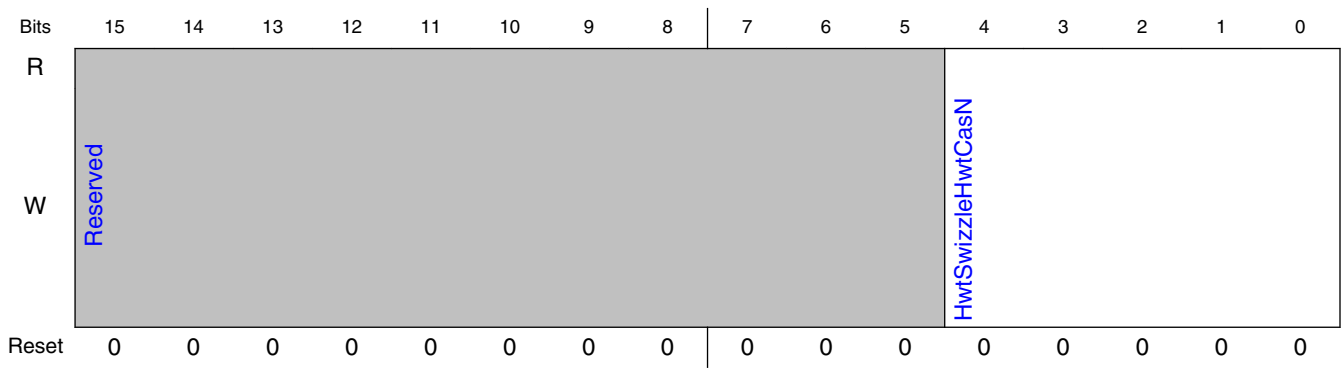
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtBg1	See Description of HwtSwizzleHwtAddress0 for details. See Description of HwtSwizzleHwtAddress0 for details. Used only for DDR4.

9.4.3.6.162 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtCasN)

9.4.3.6.162.1 Offset

Register	Offset
HwtSwizzleHwtCasN	26Eh

9.4.3.6.162.2 Diagram



9.4.3.6.162.3 Fields

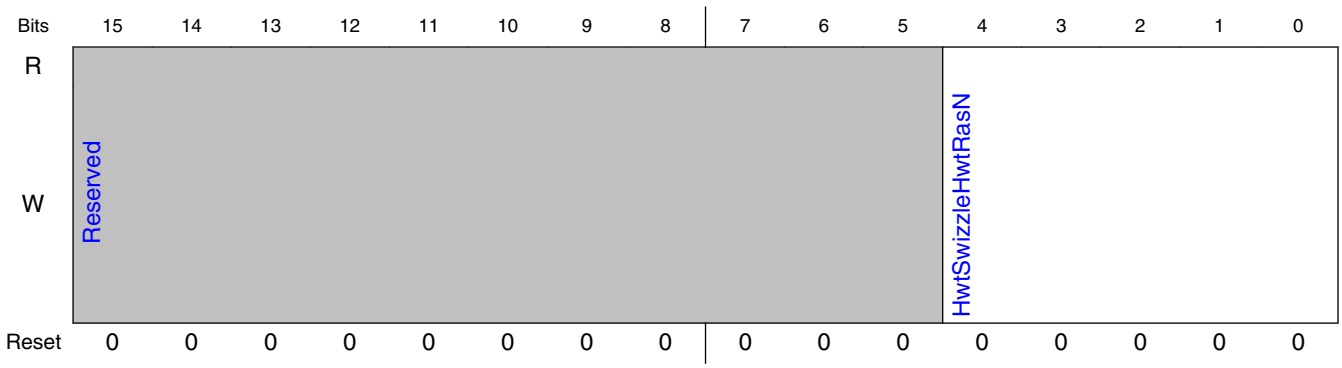
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtCasN	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.163 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtRasN)

9.4.3.6.163.1 Offset

Register	Offset
HwtSwizzleHwtRasN	270h

9.4.3.6.163.2 Diagram



9.4.3.6.163.3 Fields

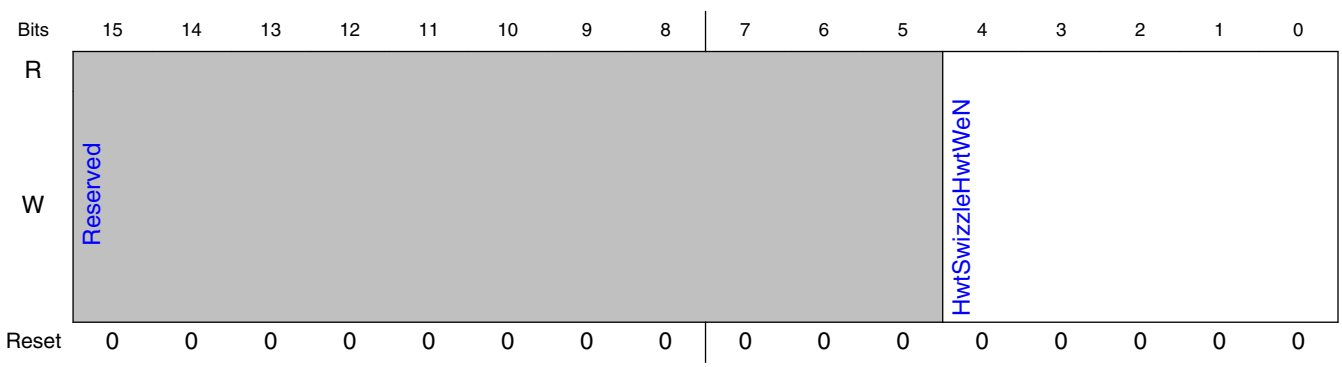
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtRasN	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.164 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtWeN)

9.4.3.6.164.1 Offset

Register	Offset
HwtSwizzleHwtWeN	272h

9.4.3.6.164.2 Diagram



9.4.3.6.164.3 Fields

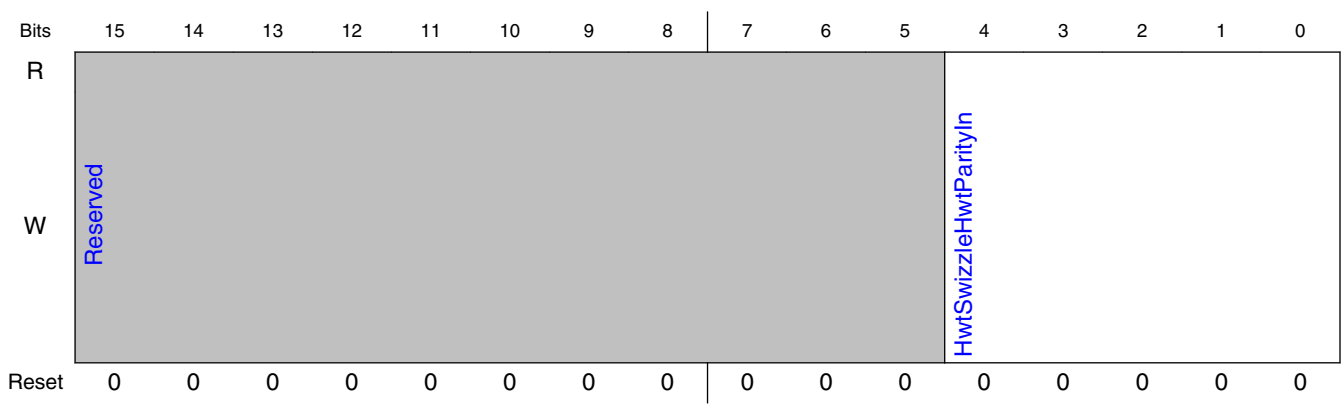
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtWeN	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.165 Signal swizzle selection for HWT swizzle (HwtSwizzleHwtParityIn)

9.4.3.6.165.1 Offset

Register	Offset
HwtSwizzleHwtParityIn	274h

9.4.3.6.165.2 Diagram



9.4.3.6.165.3 Fields

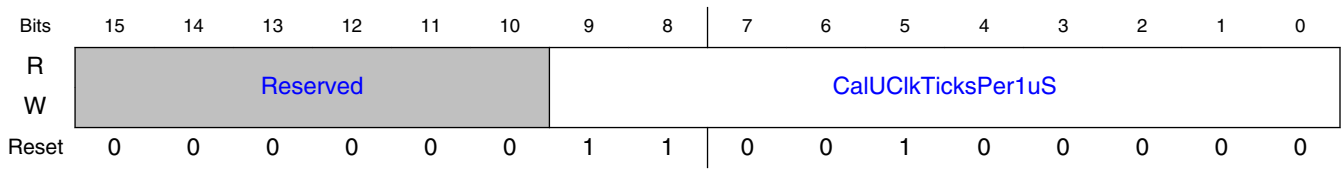
Field	Function
15-5 —	Reserved
4-0 HwtSwizzleHwtParityIn	See Description of HwtSwizzleHwtAddress0 for details.

9.4.3.6.166 Impedance Calibration Clock Ratio (CalUclkInfo_p1)

9.4.3.6.166.1 Offset

Register	Offset
CalUclkInfo_p1	20_0010h

9.4.3.6.166.2 Diagram



9.4.3.6.166.3 Fields

Field	Function
15-10 —	Reserved
9-0 CalUClkTicksPer1uS	Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. if (DfiClk < 24MHz) CalUclkInfo = 24 else CalUclkInfo = (number of DfiClks in 1us)

9.4.3.6.167 PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p1)

9.4.3.6.167.1 Offset

Register	Offset
Seq0BDLY0_p1	20_0016h

9.4.3.6.167.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY0_p1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.167.3 Fields

Field	Function
15-0 Seq0BDLY0_p1	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.168 PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p1)

9.4.3.6.168.1 Offset

Register	Offset
Seq0BDLY1_p1	20_0018h

9.4.3.6.168.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY1_p1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.168.3 Fields

Field	Function
15-0 Seq0BDLY1_p1	PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: -------- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --------- --------- ---------- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.169 PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p1)

9.4.3.6.169.1 Offset

Register	Offset
Seq0BDLY2_p1	20_001Ah

9.4.3.6.169.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY2_p1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.169.3 Fields

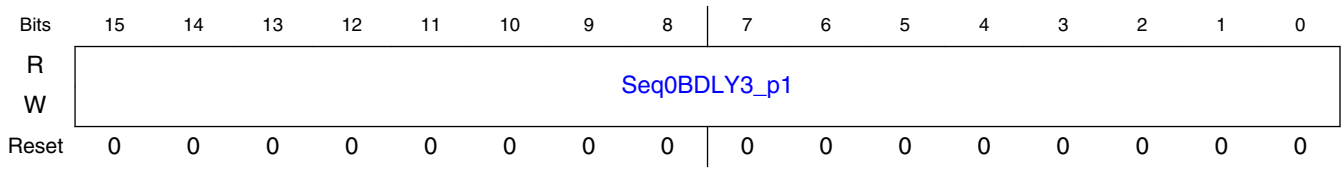
Field	Function
15-0 Seq0BDLY2_p1	PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.170 PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p1)

9.4.3.6.170.1 Offset

Register	Offset
Seq0BDLY3_p1	20_001Ch

9.4.3.6.170.2 Diagram



9.4.3.6.170.3 Fields

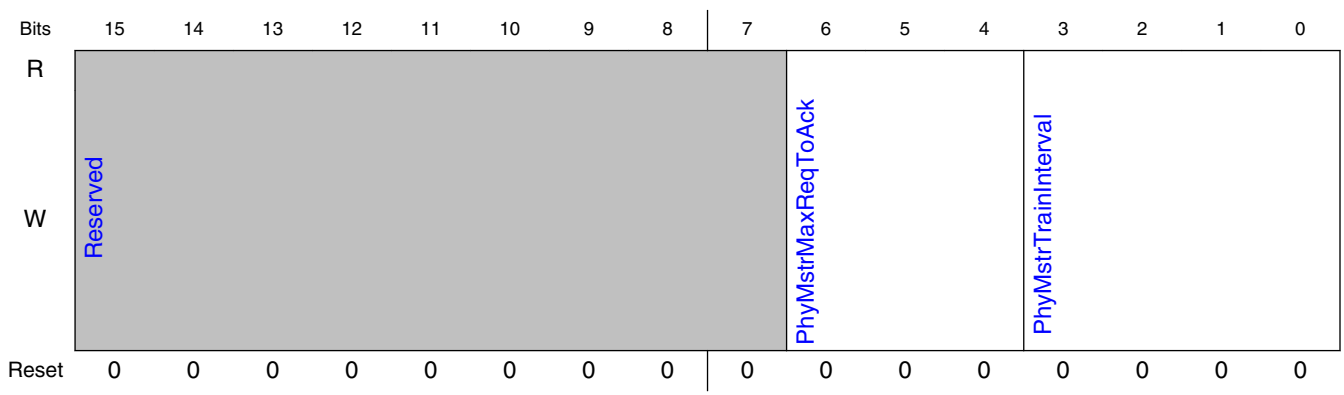
Field	Function
15-0 Seq0BDLY3_p1	PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: -------------------------------- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --------- --------------- ------------------------------ Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.171 Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p1)

9.4.3.6.171.1 Offset

Register	Offset
PPTTrainSetup_p1	20_0020h

9.4.3.6.171.2 Diagram



9.4.3.6.171.3 Fields

Field	Function
15-7 —	Reserved
6-4 PhyMstrMaxReqToAck	Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. 3'b000 Disable PHY Master Interface 3'b001 set tPHYMSTR_resp. = 512 MEMCLKs 3'b010 set tPHYMSTR_resp. = 1024 MEMCLKs 3'b011 set tPHYMSTR_resp. = 2048 MEMCLKs 3'b100 set tPHYMSTR_resp. = 4096 MEMCLKs 3'b101 set tPHYMSTR_resp. = 8192 MEMCLKs 3'b110 set tPHYMSTR_resp. = 32768 MEMCLKs

Table continues on the next page...

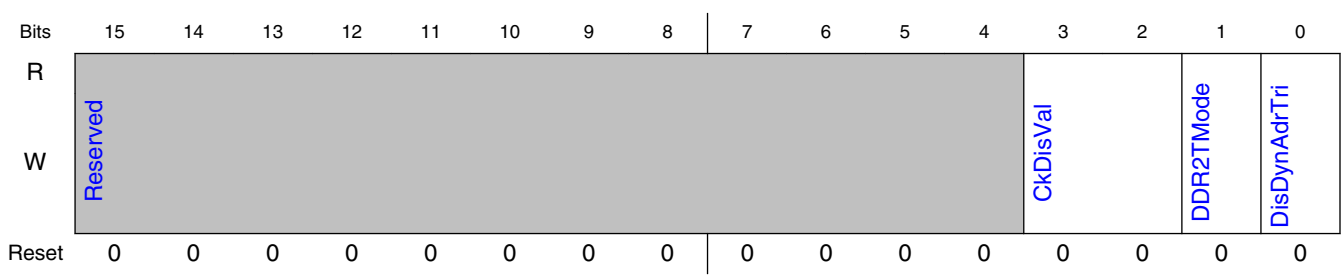
Field	Function
	3'b111 set tPHYMSTR_resp. = undefined
3-0 PhyMstrTrainInterval	Bits 3:0 of this register specifies the time between the end of one training and the start of the next. Bits 3:0 of this register specifies the time between the end of one training and the start of the next. (As per section 11 of the Preliminary DFI 4.0 Specification, V2, it is the max expected time from dfi_init_complete asserted to tdfi_phymstr_ack asserted). 4'b0000 Disable PHY Master Interface 4'b0001 PHY MASTER Request Interval = 524288 MEMCLKs 4'b0010 PHY MASTER Request Interval = 1048576 MEMCLKs 4'b0011 PHY MASTER Request Interval = 2097152 MEMCLKs 4'b0100 PHY MASTER Request Interval = 4194304 MEMCLKs 4'b0101 PHY MASTER Request Interval = 8388608 MEMCLKs 4'b0110 PHY MASTER Request Interval = 16777216 MEMCLKs 4'b0111 PHY MASTER Request Interval = 33554432 MEMCLKs 4'b1000 PHY MASTER Request Interval = 67108864 MEMCLKs 4'b1001 PHY MASTER Request Interval = 134217728 MEMCLKs 4'b1010 PHY MASTER Request Interval = 268435456 MEMCLKs 4'b1011 - 4'b1111 PHY MASTER Request Interval = undefined

9.4.3.6.172 Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p1)

9.4.3.6.172.1 Offset

Register	Offset
TristateModeCA_p1	20_0032h

9.4.3.6.172.2 Diagram



9.4.3.6.172.3 Fields

Field	Function
15-4 —	Reserved
3-2 CkDisVal	The PHY provides 4 memory clocks, n=0. The PHY provides 4 memory clocks, n=0..3. When the toggling of memory clock CK_t[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_t[n] is driven with Register CkDisVal[1]. When the toggling of memory clock CK_c[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_c[n] is driven with ~(Register CkDisVal[0]) (ie inverted). Note that the non-toggling differential memory clock CK_t[n],CK_c[n] may be driven with any combination, LL,LH,HL,HH; the default CK_t[n],CK_c[n]=LH.
1 DDR2TMode	Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. This CSR bit has no effect when DisDynAdrTri==1
0 DisDynAdrTri	When DisDynAdrTri=1, Dynamic Tristating is disabled. When DisDynAdrTri=1, Dynamic Tristating is disabled. Dynamic Tristating is on by default. . Dynamic Tristating should be disabled (DisDynAdrTri=0) for these modes: 1. DDR3/2T if the controller cannot follow the 2T PHY tristate protocol. 2. DDR4/2T/2N if the controller cannot follow the 2T PHY tristate protocol. 3. LPDDR4 . When DisDynAdrTri=0 (default), In DDR3 mode, The following SDRAM pins can be dynamically tristated: A,BA,RAS_n,CAS_n,WE_n In DDR4 mode, The following SDRAM pins can be dynamically tristated: A,BA,BG,ACT_n In LPDDR3 mode, The following SDRAM pins can be dynamically tristated: CA[*] . In 1T mode, the PHY will tristate the relevant pins when all ranks are DESelected . In 2T mode (or geardown), the PHY will tristate the relevant pins when: D3 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0 = 1,1,1,0 D4 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0,ACT_n = 1,1,1,0,1 When in this mode, the controller should avoid sending the above patterns with any rank selected. [e.g. NOPs sent by the controller should have BA0 set to a non-zero value]

9.4.3.6.173 HWT MaxReadLatency. (HwtMRL_p1)

9.4.3.6.173.1 Offset

Register	Offset
HwtMRL_p1	20_0040h

9.4.3.6.173.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											HwtMRL_p1				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

9.4.3.6.173.3 Fields

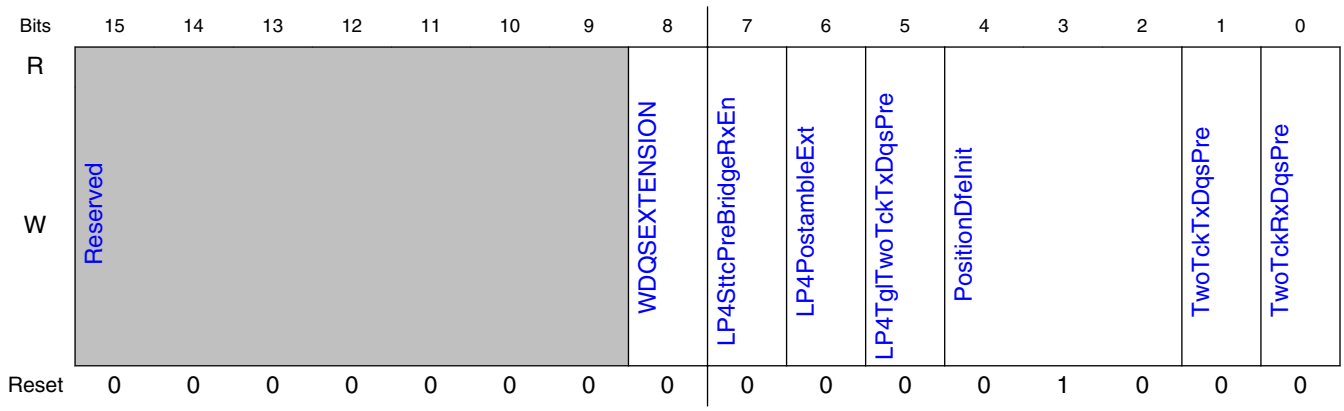
Field	Function
15-5 —	Reserved
4-0 HwtMRL_p1	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. The CSR is in units of two mem clocks; that is, a unit change in the LSB is a change in MRL of two mem clocks. This MASTER copy of MRL is used by the PHY training hardware only.

9.4.3.6.174 Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p1)

9.4.3.6.174.1 Offset

Register	Offset
DqsPreambleControl_p1	20_0048h

9.4.3.6.174.2 Diagram



9.4.3.6.174.3 Fields

Field	Function
15-9 —	Reserved
8 WDQSEXTENSION	When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. See DesignWare Cores LPDDR4 MultiPHY: WDQS Extension Application Note. Use of WDQSEXTENSION requires POdtTailWidth=3 and POdtStartDelay=00
7 LP4SttcPreBridgeRxEn	Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. LPDDR4 static-preamble mode is set in the DRAMs with MR1, OP[3]=0.
6 LP4PostambleExt	In LPDDR4 mode must be set to extend the write postamble. In LPDDR4 mode must be set to extend the write postamble. 0: half-memclk postamble, as in D4 1: one-and-one-half-memclk postamble; see LPDDR4 Spec MR3, OP[1] WR PST, vendor-specific function.
5 LP4TglTwoTckTxDqsPre	Used in LPDDR4 mode to modify the early preamble when Register TwoTckTxDqsPre=1 0: level first-memclk preamble 1: toggling first-memclk preamble
4-2 PositionDfeInit	For DDR4 phy only when receive DFE is enabled. For DDR4 phy only when receive DFE is enabled. PositionDfeInit[2]=1 causes DfeInit to be asserted late PositionDfeInit[1]=1 causes DfeInit to be asserted at the nominal time

Table continues on the next page...

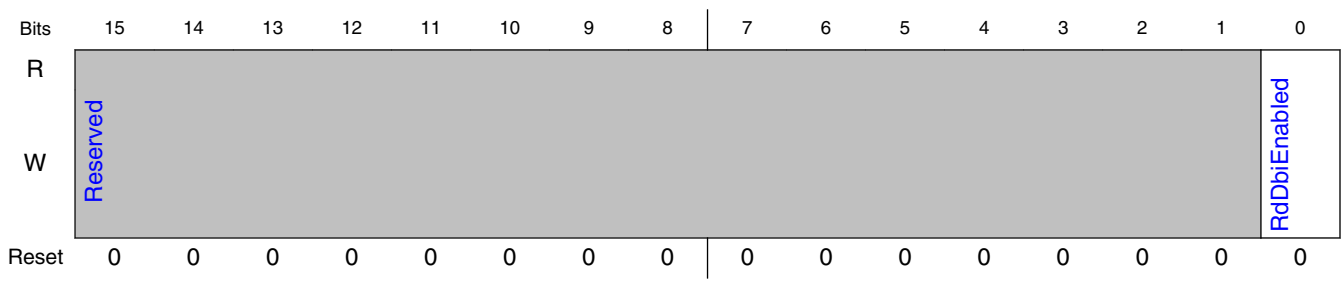
Field	Function
	PositionDfeInit[0]=1 causes DfeInit to be asserted early PositionDfeInit=3'b010 is the nominal, recommended value for leading edge used by receiver.
1 TwoTckTxDqsPre	0: Standard 1tck TxDqs Preamble 1: Enable Optional D4 2tck TxDqs Preamble The DDR4 MR4 A12 is Write Preamble, 1=2nCK, 0=1nCK.
0 TwoTckRxDqsPre	Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. For LPDDR4, all read operations are 2nCK such that this control must be set to 1. Note the nominal trained-to-center point will be earlier relative to the first strobing edge of DQS than when not in this mode.

9.4.3.6.175 This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p1)

9.4.3.6.175.1 Offset

Register	Offset
DMIPinPresent_p1	20_005Ah

9.4.3.6.175.2 Diagram



9.4.3.6.175.3 Fields

Field	Function
15-1	Reserved

Table continues on the next page...

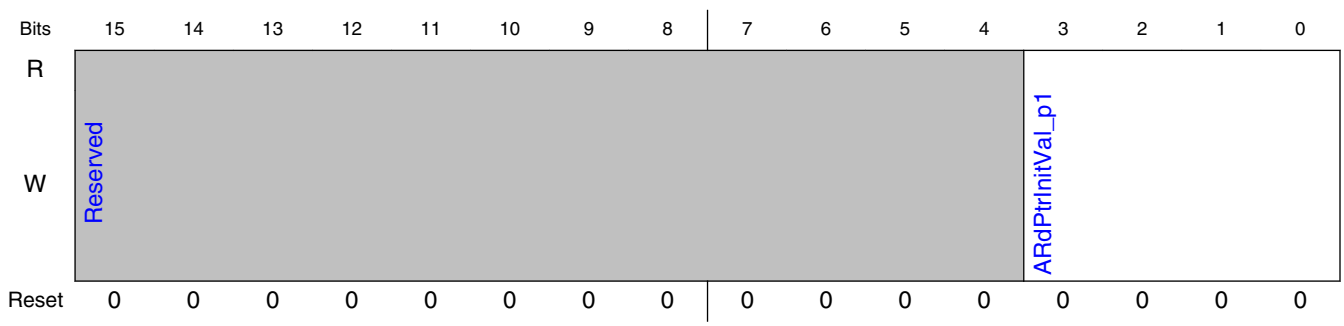
Field	Function
—	
0 RdDbiEnabled	This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. If set, the following DRAM MR should also be set [DDR4.MR5.A12=1 or LPDDR4.MR3.OP[7]=1]

9.4.3.6.176 Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p1)

9.4.3.6.176.1 Offset

Register	Offset
ARdPtrInitVal_p1	20_005Ch

9.4.3.6.176.2 Diagram



9.4.3.6.176.3 Fields

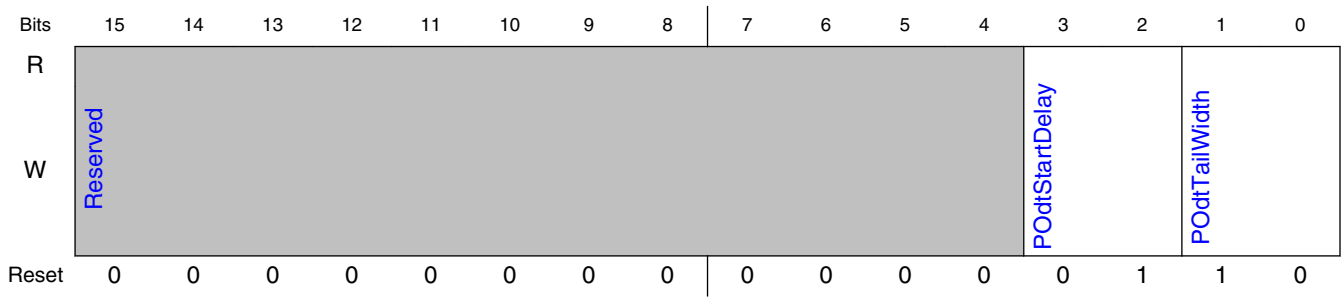
Field	Function
15-4 —	Reserved
3-0 ARdPtrInitVal_p1	This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. The units of this register are in UI. The pointer separation should be chosen to compensate for all sources of skew and drift of the PHY DFICLK and PCLK networks. Please see PUB databook section 8.1.1 This CSR must be programmed in Step C of the PHY Initialization sequence

9.4.3.6.177 READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p1)

9.4.3.6.177.1 Offset

Register	Offset
ProcOdtTimeCtl_p1	20_00ACh

9.4.3.6.177.2 Diagram



9.4.3.6.177.3 Fields

Field	Function
15-4 —	Reserved
3-2 POdtStartDelay	controls the start of ProcOdt, units of UI 3 delay start 2 UI, maximum delay of start of ProcOdt 2 delay start 1 UI, 1 delay start 0 UI, default 0 early by 1 UI, The time from ProcODT assertion to opening the window to receive DQS is (10 - POdtStartDelay) UI.
1-0 POdtTailWidth	controls the length of the tail of ProcOdt, units of UI 3 tail 3UI more than for Register POdtTailWidth=0, maximum 2 tail 2UI more than for Register POdtTailWidth=0, default 1 tail 1UI more than for Register POdtTailWidth=0 0 minimum length tail The time from ProcODT to closing the window to receive DQS to ProcODT POdtTailWidth is (2 + POdtTailWidth) UI

9.4.3.6.178 DLL gain control (DllGainCtl_p1)

9.4.3.6.178.1 Offset

Register	Offset
DllGainCtl_p1	20_00F8h

9.4.3.6.178.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DllSeedSel				DllGainTV				DllGainIV			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1

9.4.3.6.178.3 Fields

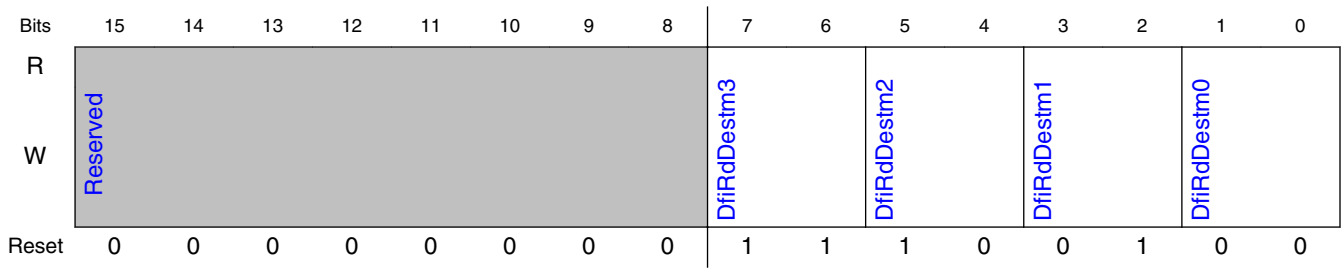
Field	Function
15-12 —	Reserved
11-8 DllSeedSel	Reserved, must be configured to be 0.
7-4 DllGainTV	Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. DllGainTV must be greater than or equal to DllGainIV. The maximum value is 10. The minimum value is 6.
3-0 DllGainIV	Initial value of DllGain.

9.4.3.6.179 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p1)

9.4.3.6.179.1 Offset

Register	Offset
DfiRdDataCsDestMap_p1	20_0160h

9.4.3.6.179.2 Diagram



9.4.3.6.179.3 Fields

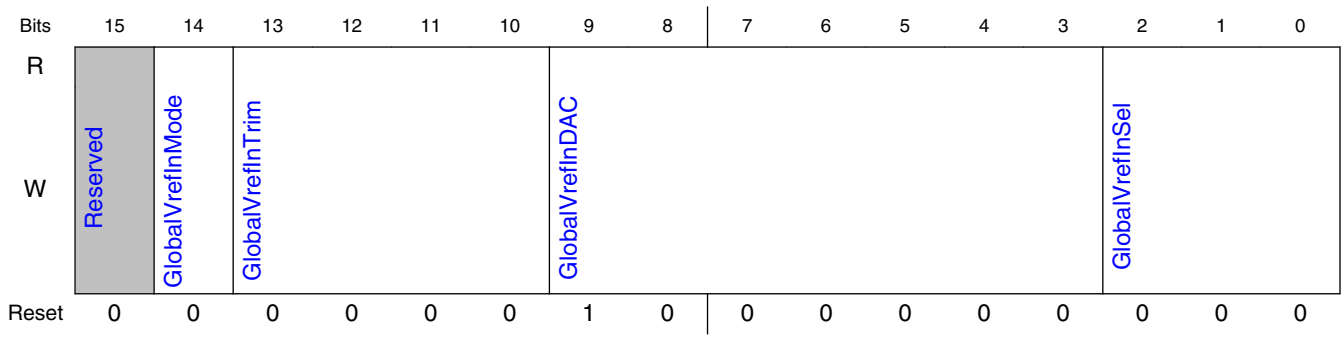
Field	Function
15-8 —	Reserved
7-6 DfiRdDestm3	Maps dfi_rddata_cs_n_p0[3] to dest DfiRdDestm3 timing For example, if 3 dfi_rddata_cs_n_p0[3] will use Register RxEn,ClkDlyTg3 timing.
5-4 DfiRdDestm2	Maps dfi_rddata_cs_n_p0[2] to dest DfiRdDestm2 timing For example, if 2 dfi_rddata_cs_n_p0[2] will use Register RxEn,ClkDlyTg2 timing.
3-2 DfiRdDestm1	Maps dfi_rddata_cs_n_p0[1] to dest DfiRdDestm1 timing For example, if 1 dfi_rddata_cs_n[_p01] will use Register RxEn,ClkDlyTg1 timing.
1-0 DfiRdDestm0	Maps dfi_rddata_cs_n_p0[0] to dest DfiRdDestm0 timing For example, if 0 dfi_rddata_cs_n_p0[0] will use Register RxEn,ClkDlyTg0 timing.

9.4.3.6.180 PHY Global Vref Controls (VrefInGlobal_p1)

9.4.3.6.180.1 Offset

Register	Offset
VrefInGlobal_p1	20_0164h

9.4.3.6.180.2 Diagram



9.4.3.6.180.3 Fields

Field	Function
15 —	Reserved
14 GlobalVrefInMode	RSVD
13-10 GlobalVrefInTrim	RSVD
9-3 GlobalVrefInDAC	DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0. DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0.345*VDDQ + (0.005*GlobalVrefInDAC)*VDDQ ========== RANGE1 : LPDDR4 [GlobalVrefInSel[2] = 1] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : (0.005*(GlobalVrefInDAC-1))*VDDQ
2-0 GlobalVrefInSel	GlobalVrefInSel[1:0] controls the mode of the PHY VREF DAC and the BP_VREF pin ========== 2'b00 - PHY Vref DAC Range0 -- BP_VREF = Hi-Z 2'b01 - Reserved Encoding 2'b10 - PHY Vref DAC Range0 -- BP_VREF connected to PLL Analog Bus 2'b11 - PHY Vref DAC Range0 -- BP_VREF connected to PHY Vref DAC ========== GlobalVrefInSel[2] shall be set according to Dram Protocol: Protocol GlobalVrefInSel[2] ---------- DDR3 1'b0 DDR4 1'b0 LPDDR3 1'b0 LPDDR4 1'b1

9.4.3.6.181 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p1)

9.4.3.6.181.1 Offset

Register	Offset
DfiWrDataCsDestMap_p1	20_0168h

9.4.3.6.181.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								DfiWrDestm3		DfiWrDestm2		DfiWrDestm1		DfiWrDestm0	
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

9.4.3.6.181.3 Fields

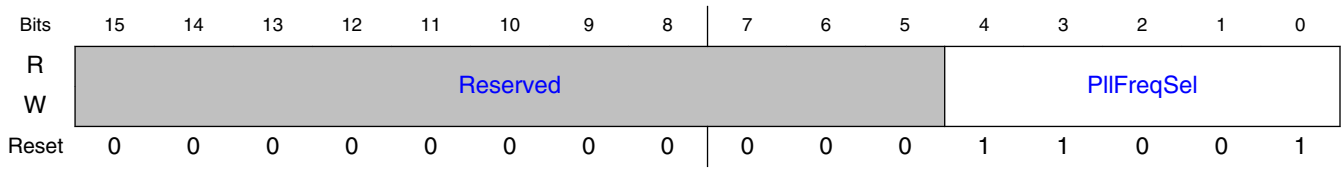
Field	Function
15-8 —	Reserved
7-6 DfiWrDestm3	Maps dfi_wrdata_cs_n_p0[3] to dest DfiWrDestm3 timing (use Register TxDq,DqsDlyTg3) For example, if 3 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg3 timing.
5-4 DfiWrDestm2	Maps dfi_wrdata_cs_n_p0[2] to dest DfiWrDestm2 timing For example, if 2 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg2 timing.
3-2 DfiWrDestm1	Maps dfi_wrdata_cs_n_p0[1] to dest DfiWrDestm1 timing For example, if 1 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg1 timing.
1-0 DfiWrDestm0	Maps dfi_wrdata_cs_n_p0[0] to dest DfiWrDestm0 timing For example, if 0 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg0 timing.

9.4.3.6.182 PState dependent PLL Control Register 2 (PllCtrl2_p1)

9.4.3.6.182.1 Offset

Register	Offset
PllCtrl2_p1	20_018Ah

9.4.3.6.182.2 Diagram



9.4.3.6.182.3 Fields

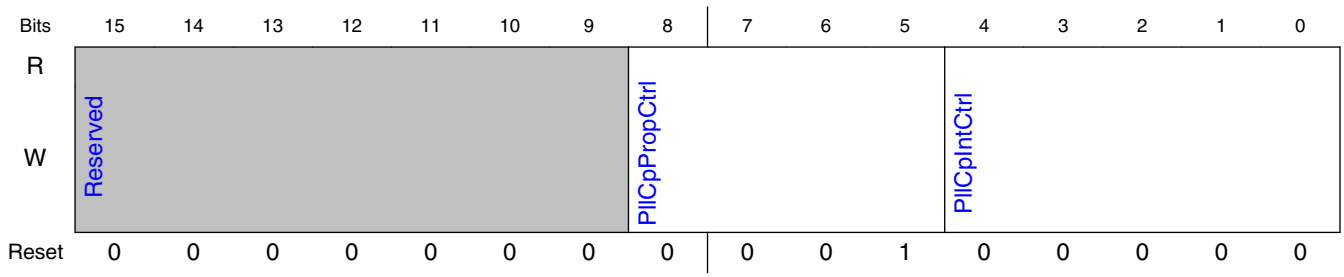
Field	Function
15-5 —	Reserved
4-0 PllFreqSel	Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Program based on reference clock freqency (DfiClk) with this table.

9.4.3.6.183 PState dependent PLL Control Register 1 (PllCtrl1_p1)

9.4.3.6.183.1 Offset

Register	Offset
PllCtrl1_p1	20_018Eh

9.4.3.6.183.2 Diagram



9.4.3.6.183.3 Fields

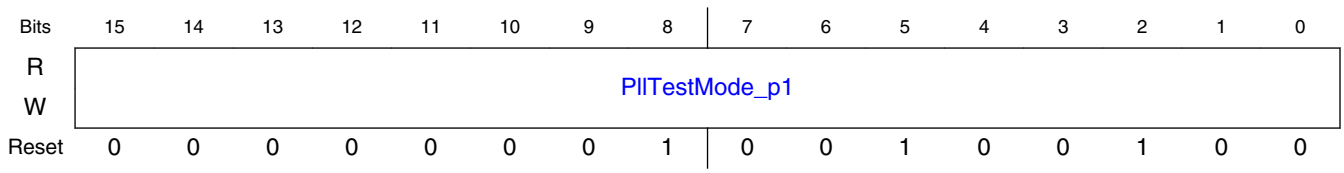
Field	Function
15-9 —	Reserved
8-5 PllCpPropCtrl	connects directly to cp_prop_cntrl<3:0> of PLL. connects directly to cp_prop_cntrl<3:0> of PLL. Charge pump proportional current control.
4-0 PllCpIntCtrl	connects directly to cp_int_cntrl<1:0> in PLL. connects directly to cp_int_cntrl<1:0> in PLL. Charge pump integrating current control.

9.4.3.6.184 Additional controls for PLL CP/VCO modes of operation (PllTestMode_p1)

9.4.3.6.184.1 Offset

Register	Offset
PllTestMode_p1	20_0194h

9.4.3.6.184.2 Diagram



9.4.3.6.184.3 Fields

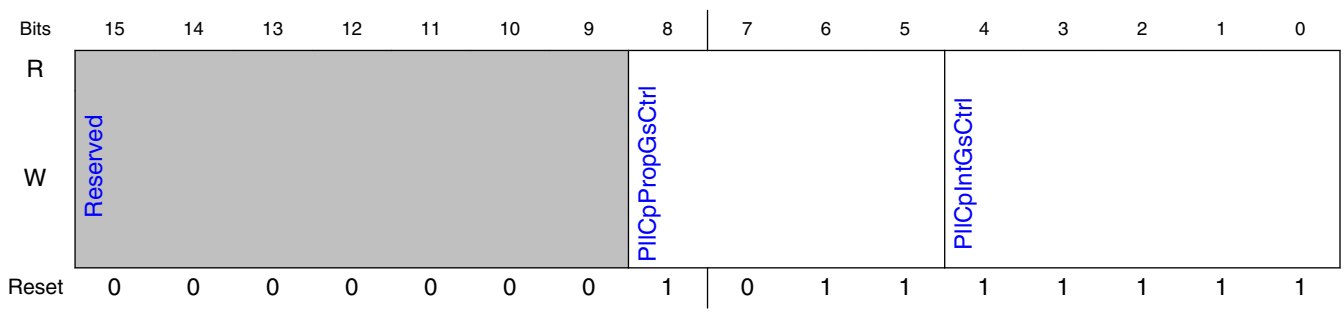
Field	Function
15-0 PllTestMode_p1	It is required to use default values for this CSR unless directed otherwise by Synopsys. It is required to use default values for this CSR unless directed otherwise by Synopsys. Directly connected to testmode[15:0] pin of PLL

9.4.3.6.185 PState dependent PLL Control Register 4 (PllCtrl4_p1)

9.4.3.6.185.1 Offset

Register	Offset
PllCtrl4_p1	20_0198h

9.4.3.6.185.2 Diagram



9.4.3.6.185.3 Fields

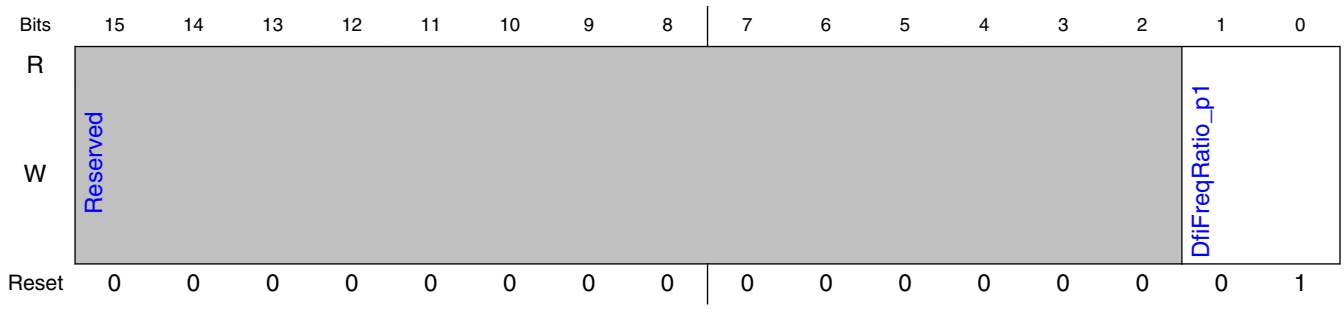
Field	Function
15-9 —	Reserved
8-5 PllCpPropGsCtrl	connects directly to cp_prop_gs_cntrl<3:0> of PLL. connects directly to cp_prop_gs_cntrl<3:0> of PLL. Charge pump proportional current control for fast relock and gearshift.
4-0 PllCpIntGsCtrl	connects directly to cp_int_gs_cntrl<4:0> in PLL. connects directly to cp_int_gs_cntrl<4:0> in PLL. Charge pump integrating current control for fast relock and gearshift.

9.4.3.6.186 DFI Frequency Ratio (DfiFreqRatio_p1)

9.4.3.6.186.1 Offset

Register	Offset
DfiFreqRatio_p1	20_01F4h

9.4.3.6.186.2 Diagram



9.4.3.6.186.3 Fields

Field	Function
15-2 —	Reserved
1-0 DfiFreqRatio_p1	Used in dwc_ddrphy_pub_serdes to serialize or de-serialize DFI signals 00 = 1:1 mode 01 = 1:2 mode 1x = 1:4 mode* *Note: 1:4 is for future pub revision.

9.4.3.6.187 Impedance Calibration Clock Ratio (CalUclkInfo_p2)

9.4.3.6.187.1 Offset

Register	Offset
CalUclkInfo_p2	40_0010h

9.4.3.6.187.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								CalUClkTicksPer1uS							
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0

9.4.3.6.187.3 Fields

Field	Function
15-10 —	Reserved
9-0 CalUClkTicksPer1uS	Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. if (DfiClk < 24MHz) CalUclkInfo = 24 else CalUclkInfo = (number of DfiClks in 1us)

9.4.3.6.188 PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p2)

9.4.3.6.188.1 Offset

Register	Offset
Seq0BDLY0_p2	40_0016h

9.4.3.6.188.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY0_p2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.188.3 Fields

Field	Function
15-0 Seq0BDLY0_p2	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value.

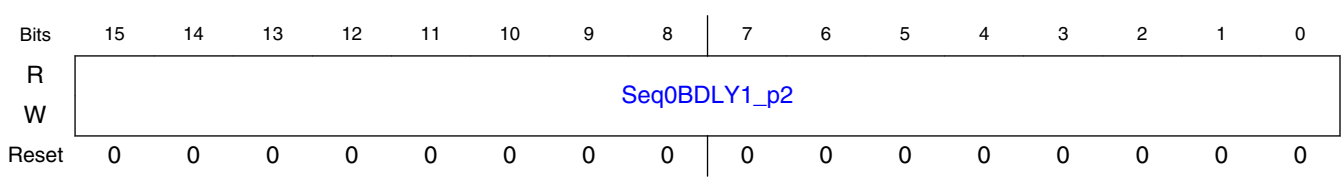
Field	Function
	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.189 PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p2)

9.4.3.6.189.1 Offset

Register	Offset
Seq0BDLY1_p2	40_0018h

9.4.3.6.189.2 Diagram



9.4.3.6.189.3 Fields

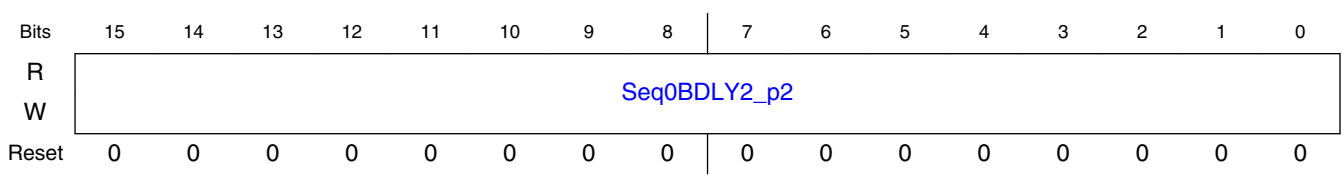
Field	Function
15-0 Seq0BDLY1_p2	PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.190 PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p2)

9.4.3.6.190.1 Offset

Register	Offset
Seq0BDLY2_p2	40_001Ah

9.4.3.6.190.2 Diagram



9.4.3.6.190.3 Fields

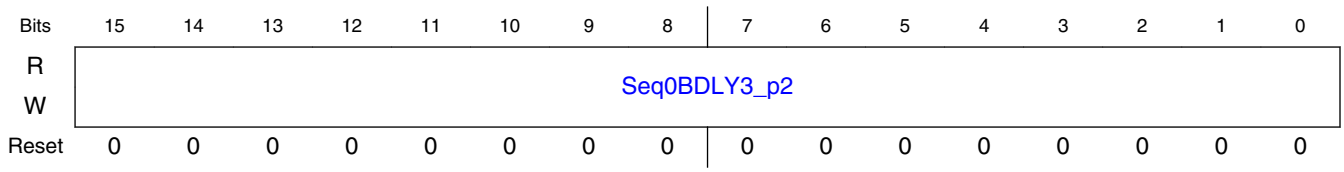
Field	Function
15-0 Seq0BDLY2_p2	PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.191 PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p2)

9.4.3.6.191.1 Offset

Register	Offset
Seq0BDLY3_p2	40_001Ch

9.4.3.6.191.2 Diagram



9.4.3.6.191.3 Fields

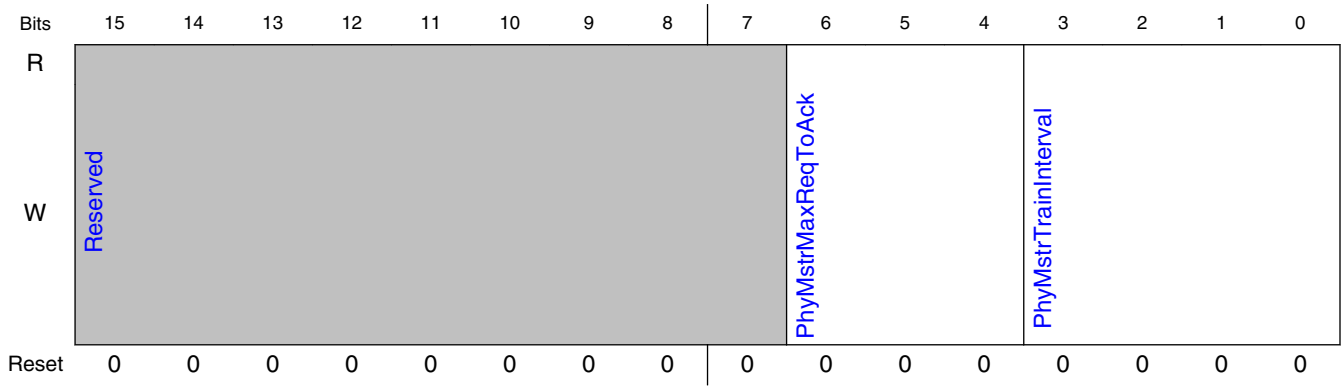
Field	Function
15-0 Seq0BDLY3_p2	PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.192 Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p2)

9.4.3.6.192.1 Offset

Register	Offset
PPTTrainSetup_p2	40_0020h

9.4.3.6.192.2 Diagram



9.4.3.6.192.3 Fields

Field	Function
15-7 —	Reserved
6-4 PhyMstrMaxReqToAck	Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. 3'b000 Disable PHY Master Interface 3'b001 set tPHYMSTR_resp. = 512 MEMCLKs 3'b010 set tPHYMSTR_resp. = 1024 MEMCLKs 3'b011 set tPHYMSTR_resp. = 2048 MEMCLKs 3'b100 set tPHYMSTR_resp. = 4096 MEMCLKs 3'b101 set tPHYMSTR_resp. = 8192 MEMCLKs 3'b110 set tPHYMSTR_resp. = 32768 MEMCLKs 3'b111 set tPHYMSTR_resp. = undefined
3-0 PhyMstrTrainInterval	Bits 3:0 of this register specifies the time between the end of one training and the start of the next. Bits 3:0 of this register specifies the time between the end of one training and the start of the next. (As per section 11 of the Preliminary DFI 4.0 Specification, V2, it is the max expected time from dfi_init_complete asserted to tdfi_phymstr_ack asserted). 4'b0000 Disable PHY Master Interface 4'b0001 PHY MASTER Request Interval = 524288 MEMCLKs 4'b0010 PHY MASTER Request Interval = 1048576 MEMCLKs

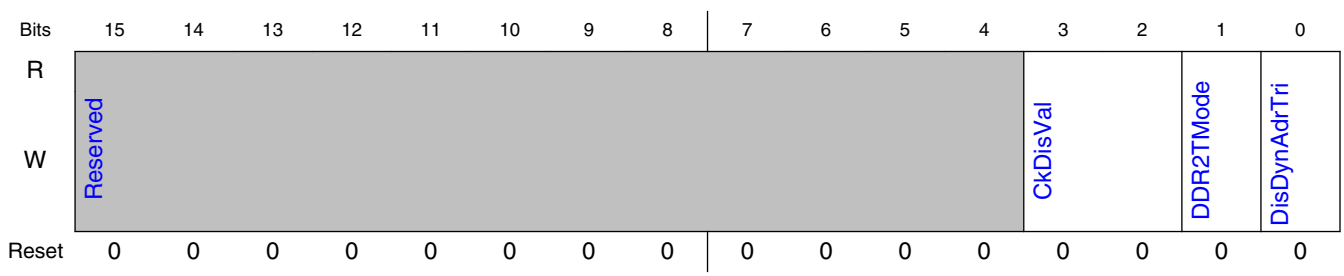
Field	Function
	4'b0011 PHY MASTER Request Interval = 2097152 MEMCLKs
	4'b0100 PHY MASTER Request Interval = 4194304 MEMCLKs
	4'b0101 PHY MASTER Request Interval = 8388608 MEMCLKs
	4'b0110 PHY MASTER Request Interval = 16777216 MEMCLKs
	4'b0111 PHY MASTER Request Interval = 33554432 MEMCLKs
	4'b1000 PHY MASTER Request Interval = 67108864 MEMCLKs
	4'b1001 PHY MASTER Request Interval = 134217728 MEMCLKs
	4'b1010 PHY MASTER Request Interval = 268435456 MEMCLKs
	4'b1011 - 4'b1111 PHY MASTER Request Interval = undefined

9.4.3.6.193 Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p2)

9.4.3.6.193.1 Offset

Register	Offset
TristateModeCA_p2	40_0032h

9.4.3.6.193.2 Diagram



9.4.3.6.193.3 Fields

Field	Function
15-4 —	Reserved
3-2 CkDisVal	The PHY provides 4 memory clocks, n=0. The PHY provides 4 memory clocks, n=0..3. When the toggling of memory clock CK_t[n] is disabled with dfi_clk_disable[n]=1,

Table continues on the next page...

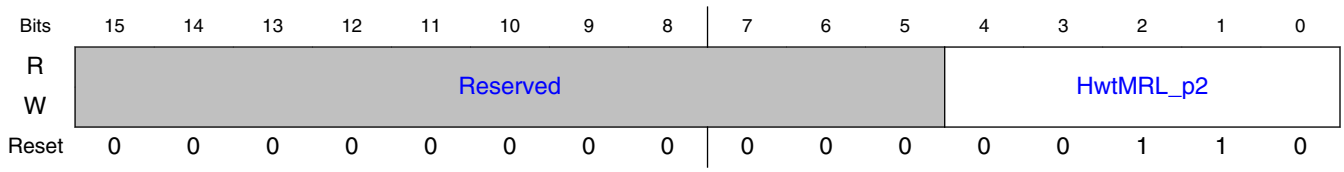
Field	Function
	the memory clock CK_t[n] is driven with Register CkDisVal[1]. When the toggling of memory clock CK_c[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_c[n] is driven with ~(Register CkDisVal[0]) (ie inverted). Note that the non-toggling differential memory clock CK_t[n],CK_c[n] may be driven with any combination, LL,LH,HL,HH; the default CK_t[n],CK_c[n]=LH.
1 DDR2TMode	Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. This CSR bit has no effect when DisDynAdrTri==1
0 DisDynAdrTri	When DisDynAdrTri=1, Dynamic Tristating is disabled. When DisDynAdrTri=1, Dynamic Tristating is disabled. Dynamic Tristating is on by default. . Dynamic Tristating should be disabled (DisDynAdrTri=0) for these modes: 1. DDR3/2T if the controller cannot follow the 2T PHY tristate protocol. 2. DDR4/2T/2N if the controller cannot follow the 2T PHY tristate protocol. 3. LPDDR4 . When DisDynAdrTri=0 (default), In DDR3 mode, The following SDRAM pins can be dynamically tristated: A,BA,RAS_n,CAS_n,WE_n In DDR4 mode, The following SDRAM pins can be dynamically tristated: A,BA,BG,ACT_n In LPDDR3 mode, The following SDRAM pins can be dynamically tristated: CA[*] . In 1T mode, the PHY will tristate the relevant pins when all ranks are DESelected . In 2T mode (or geardown), the PHY will tristate the relevant pins when: D3 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0 = 1,1,1,0 D4 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0,ACT_n = 1,1,1,0,1 When in this mode, the controller should avoid sending the above patterns with any rank selected. [e.g. NOPs sent by the controller should have BA0 set to a non-zero value]

9.4.3.6.194 HWT MaxReadLatency. (HwtMRL_p2)

9.4.3.6.194.1 Offset

Register	Offset
HwtMRL_p2	40_0040h

9.4.3.6.194.2 Diagram



9.4.3.6.194.3 Fields

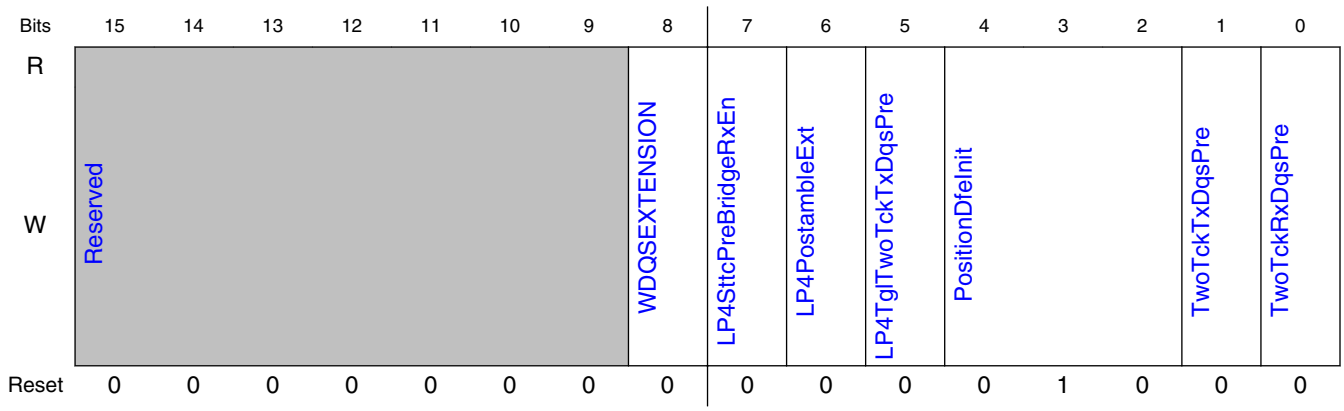
Field	Function
15-5 —	Reserved
4-0 HwtMRL_p2	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. The CSR is in units of two mem clocks; that is, a unit change in the LSB is a change in MRL of two mem clocks. This MASTER copy of MRL is used by the PHY training hardware only.

9.4.3.6.195 Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p2)

9.4.3.6.195.1 Offset

Register	Offset
DqsPreambleControl_p2	40_0048h

9.4.3.6.195.2 Diagram



9.4.3.6.195.3 Fields

Field	Function
15-9 —	Reserved
8 WDQSEXTENSION	When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. See DesignWare Cores LPDDR4 MultiPHY: WDQS Extension Application Note. Use of WDQSEXTENSION requires POdtTailWidth=3 and POdtStartDelay=00
7 LP4SttcPreBridgeRxEn	Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. LPDDR4 static-preamble mode is set in the DRAMs with MR1, OP[3]=0.
6 LP4PostambleExt	In LPDDR4 mode must be set to extend the write postamble. In LPDDR4 mode must be set to extend the write postamble. 0: half-memclk postamble, as in D4 1: one-and-one-half-memclk postamble; see LPDDR4 Spec MR3, OP[1] WR PST, vendor-specific function.
5 LP4TglTwoTckTxDqsPre	Used in LPDDR4 mode to modify the early preamble when Register TwoTckTxDqsPre=1 0: level first-memclk preamble 1: toggling first-memclk preamble
4-2 PositionDfeInit	For DDR4 phy only when receive DFE is enabled. For DDR4 phy only when receive DFE is enabled. PositionDfeInit[2]=1 causes DfeInit to be asserted late PositionDfeInit[1]=1 causes DfeInit to be asserted at the nominal time

Table continues on the next page...

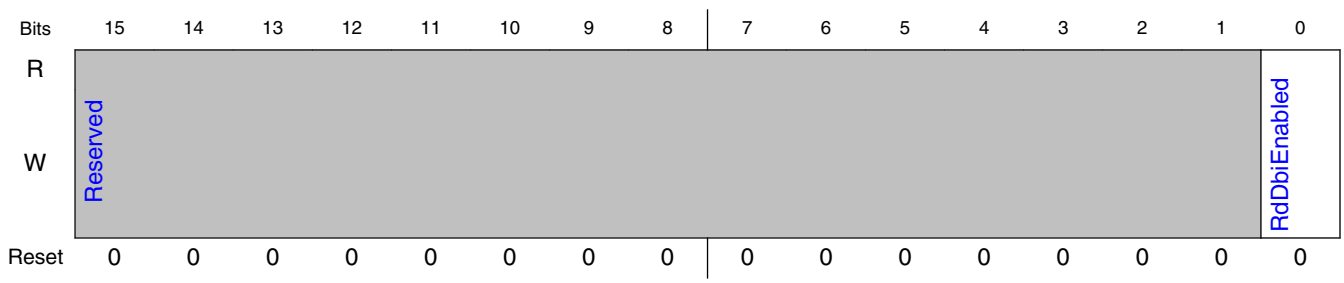
Field	Function
	PositionDfeInit[0]=1 causes DfeInit to be asserted early PositionDfeInit=3'b010 is the nominal, recommended value for leading edge used by receiver.
1 TwoTckTxDqsPre	0: Standard 1tck TxDqs Preamble 1: Enable Optional D4 2tck TxDqs Preamble The DDR4 MR4 A12 is Write Preamble, 1=2nCK, 0=1nCK.
0 TwoTckRxDqsPre	Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. For LPDDR4, all read operations are 2nCK such that this control must be set to 1. Note the nominal trained-to-center point will be earlier relative to the first strobing edge of DQS than when not in this mode.

9.4.3.6.196 This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p2)

9.4.3.6.196.1 Offset

Register	Offset
DMIPinPresent_p2	40_005Ah

9.4.3.6.196.2 Diagram



9.4.3.6.196.3 Fields

Field	Function
15-1	Reserved

Table continues on the next page...

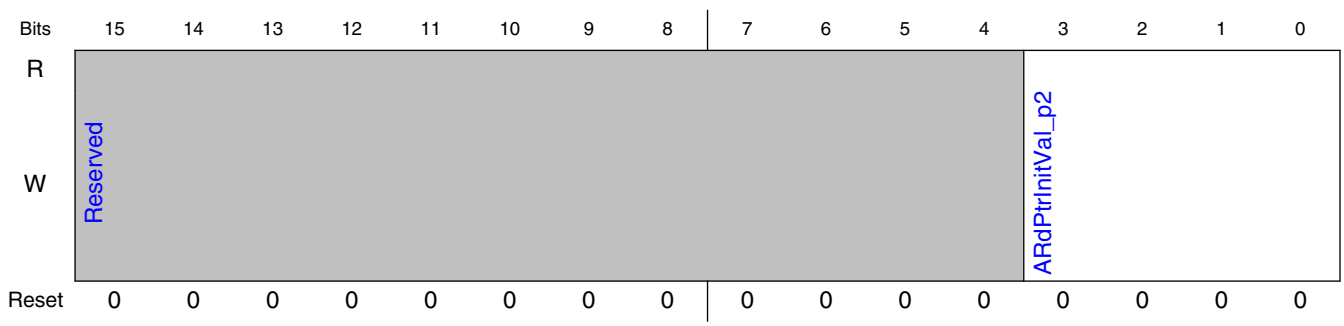
Field	Function
—	
0 RdDbiEnabled	This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. If set, the following DRAM MR should also be set [DDR4.MR5.A12=1 or LPDDR4.MR3.OP[7]=1]

9.4.3.6.197 Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p2)

9.4.3.6.197.1 Offset

Register	Offset
ARdPtrInitVal_p2	40_005Ch

9.4.3.6.197.2 Diagram



9.4.3.6.197.3 Fields

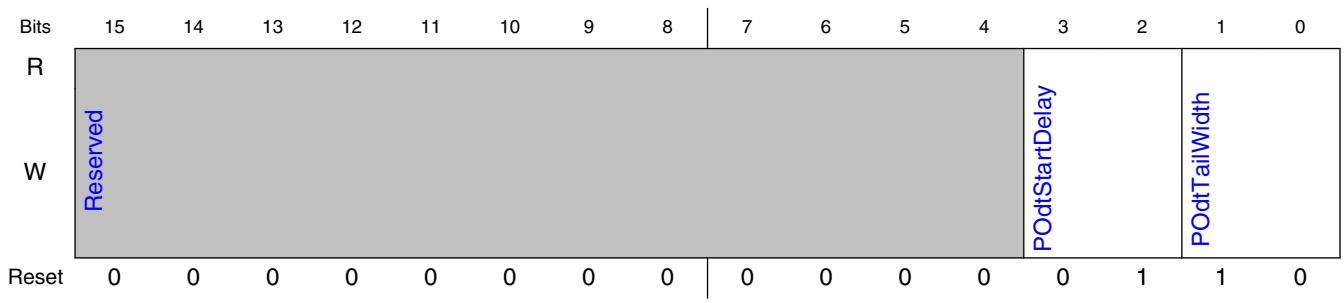
Field	Function
15-4 —	Reserved
3-0 ARdPtrInitVal_p2	This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. The units of this register are in UI. The pointer separation should be chosen to compensate for all sources of skew and drift of the PHY DFICLK and PCLK networks. Please see PUB databook section 8.1.1 This CSR must be programmed in Step C of the PHY Initialization sequence

9.4.3.6.198 READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p2)

9.4.3.6.198.1 Offset

Register	Offset
ProcOdtTimeCtl_p2	40_00ACh

9.4.3.6.198.2 Diagram



9.4.3.6.198.3 Fields

Field	Function
15-4 —	Reserved
3-2 POdtStartDelay	controls the start of ProcOdt, units of UI 3 delay start 2 UI, maximum delay of start of ProcOdt 2 delay start 1 UI, 1 delay start 0 UI, default 0 early by 1 UI, The time from ProcODT assertion to opening the window to receive DQS is (10 - POdtStartDelay) UI.
1-0 POdtTailWidth	controls the length of the tail of ProcOdt, units of UI 3 tail 3UI more than for Register POdtTailWidth=0, maximum 2 tail 2UI more than for Register POdtTailWidth=0, default 1 tail 1UI more than for Register POdtTailWidth=0 0 minimum length tail The time from ProcODT to closing the window to receive DQS to ProcODT POdtTailWidth is (2 + POdtTailWidth) UI

9.4.3.6.199 DLL gain control (DllGainCtl_p2)

9.4.3.6.199.1 Offset

Register	Offset
DllGainCtl_p2	40_00F8h

9.4.3.6.199.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DllSeedSel				DllGainTV				DllGainIV			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1

9.4.3.6.199.3 Fields

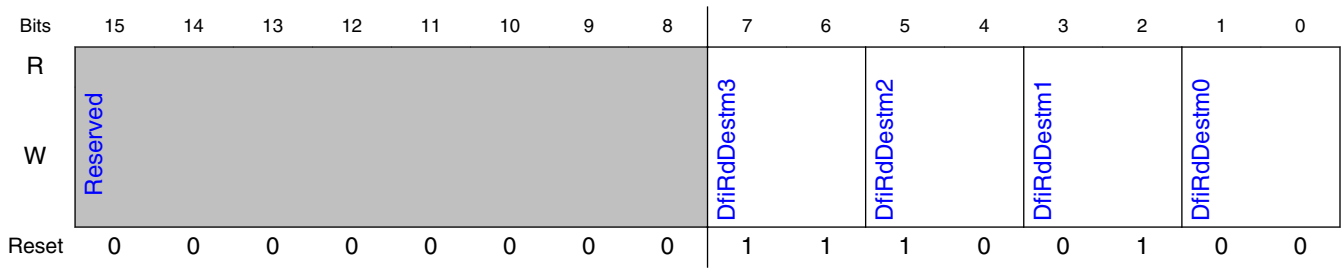
Field	Function
15-12 —	Reserved
11-8 DllSeedSel	Reserved, must be configured to be 0.
7-4 DllGainTV	Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. DllGainTV must be greater than or equal to DllGainIV. The maximum value is 10. The minimum value is 6.
3-0 DllGainIV	Initial value of DllGain.

9.4.3.6.200 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p2)

9.4.3.6.200.1 Offset

Register	Offset
DfiRdDataCsDestMap_p2	40_0160h

9.4.3.6.200.2 Diagram



9.4.3.6.200.3 Fields

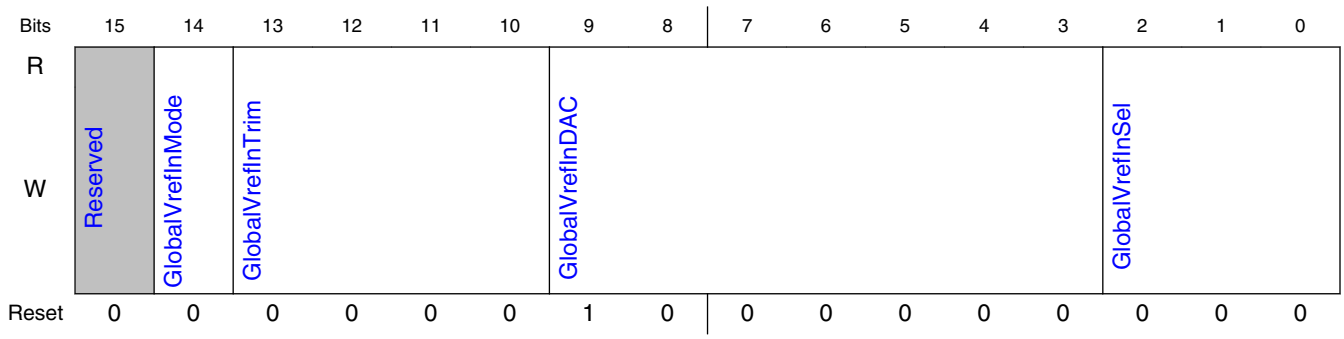
Field	Function
15-8 —	Reserved
7-6 DfiRdDestm3	Maps dfi_rddata_cs_n_p0[3] to dest DfiRdDestm3 timing For example, if 3 dfi_rddata_cs_n_p0[3] will use Register RxEn,ClkDlyTg3 timing.
5-4 DfiRdDestm2	Maps dfi_rddata_cs_n_p0[2] to dest DfiRdDestm2 timing For example, if 2 dfi_rddata_cs_n_p0[2] will use Register RxEn,ClkDlyTg2 timing.
3-2 DfiRdDestm1	Maps dfi_rddata_cs_n_p0[1] to dest DfiRdDestm1 timing For example, if 1 dfi_rddata_cs_n[_p01] will use Register RxEn,ClkDlyTg1 timing.
1-0 DfiRdDestm0	Maps dfi_rddata_cs_n_p0[0] to dest DfiRdDestm0 timing For example, if 0 dfi_rddata_cs_n_p0[0] will use Register RxEn,ClkDlyTg0 timing.

9.4.3.6.201 PHY Global Vref Controls (VrefInGlobal_p2)

9.4.3.6.201.1 Offset

Register	Offset
VrefInGlobal_p2	40_0164h

9.4.3.6.201.2 Diagram



9.4.3.6.201.3 Fields

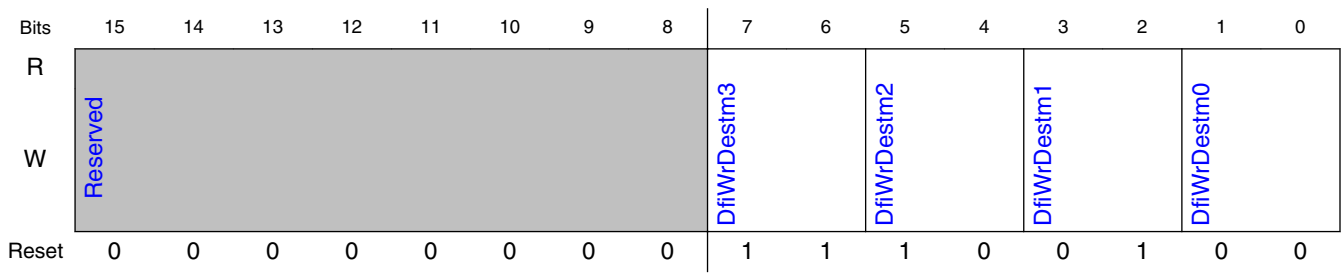
Field	Function
15 —	Reserved
14 GlobalVrefInMode	RSVD
13-10 GlobalVrefInTrim	RSVD
9-3 GlobalVrefInDAC	DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0. DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0.345*VDDQ + (0.005*GlobalVrefInDAC)*VDDQ ========== RANGE1 : LPDDR4 [GlobalVrefInSel[2] = 1] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : (0.005*(GlobalVrefInDAC-1))*VDDQ
2-0 GlobalVrefInSel	GlobalVrefInSel[1:0] controls the mode of the PHY VREF DAC and the BP_VREF pin ========== 2'b00 - PHY Vref DAC Range0 -- BP_VREF = Hi-Z 2'b01 - Reserved Encoding 2'b10 - PHY Vref DAC Range0 -- BP_VREF connected to PLL Analog Bus 2'b11 - PHY Vref DAC Range0 -- BP_VREF connected to PHY Vref DAC ========== GlobalVrefInSel[2] shall be set according to Dram Protocol: Protocol GlobalVrefInSel[2] ---------- DDR3 1'b0 DDR4 1'b0 LPDDR3 1'b0 LPDDR4 1'b1

9.4.3.6.202 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p2)

9.4.3.6.202.1 Offset

Register	Offset
DfiWrDataCsDestMap_p2	40_0168h

9.4.3.6.202.2 Diagram



9.4.3.6.202.3 Fields

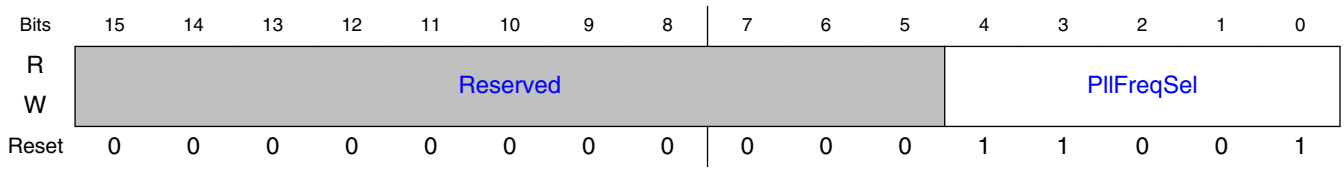
Field	Function
15-8 —	Reserved
7-6 DfiWrDestm3	Maps dfi_wrdata_cs_n_p0[3] to dest DfiWrDestm3 timing (use Register TxDq,DqsDlyTg3) For example, if 3 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg3 timing.
5-4 DfiWrDestm2	Maps dfi_wrdata_cs_n_p0[2] to dest DfiWrDestm2 timing For example, if 2 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg2 timing.
3-2 DfiWrDestm1	Maps dfi_wrdata_cs_n_p0[1] to dest DfiWrDestm1 timing For example, if 1 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg1 timing.
1-0 DfiWrDestm0	Maps dfi_wrdata_cs_n_p0[0] to dest DfiWrDestm0 timing For example, if 0 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg0 timing.

9.4.3.6.203 PState dependent PLL Control Register 2 (PllCtrl2_p2)

9.4.3.6.203.1 Offset

Register	Offset
PllCtrl2_p2	40_018Ah

9.4.3.6.203.2 Diagram



9.4.3.6.203.3 Fields

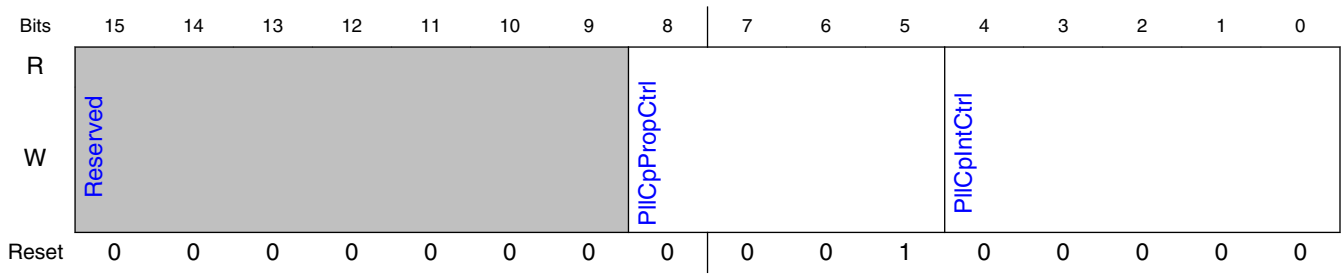
Field	Function
15-5 —	Reserved
4-0 PllFreqSel	Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Program based on reference clock freqency (DfiClk) with this table.

9.4.3.6.204 PState dependent PLL Control Register 1 (PllCtrl1_p2)

9.4.3.6.204.1 Offset

Register	Offset
PllCtrl1_p2	40_018Eh

9.4.3.6.204.2 Diagram



9.4.3.6.204.3 Fields

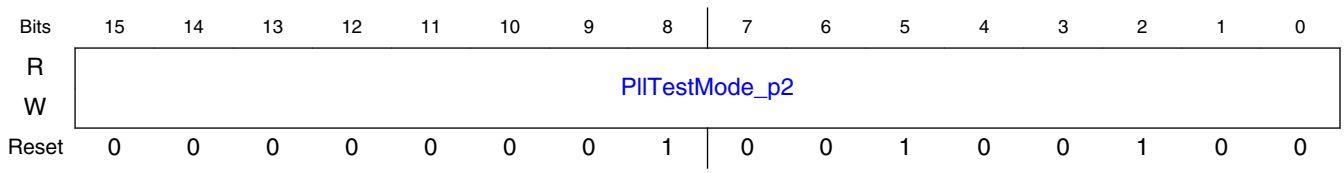
Field	Function
15-9 —	Reserved
8-5 PllCpPropCtrl	connects directly to cp_prop_cntrl<3:0> of PLL. connects directly to cp_prop_cntrl<3:0> of PLL. Charge pump proportional current control.
4-0 PllCpIntCtrl	connects directly to cp_int_cntrl<1:0> in PLL. connects directly to cp_int_cntrl<1:0> in PLL. Charge pump integrating current control.

9.4.3.6.205 Additional controls for PLL CP/VCO modes of operation (PllTestMode_p2)

9.4.3.6.205.1 Offset

Register	Offset
PllTestMode_p2	40_0194h

9.4.3.6.205.2 Diagram



9.4.3.6.205.3 Fields

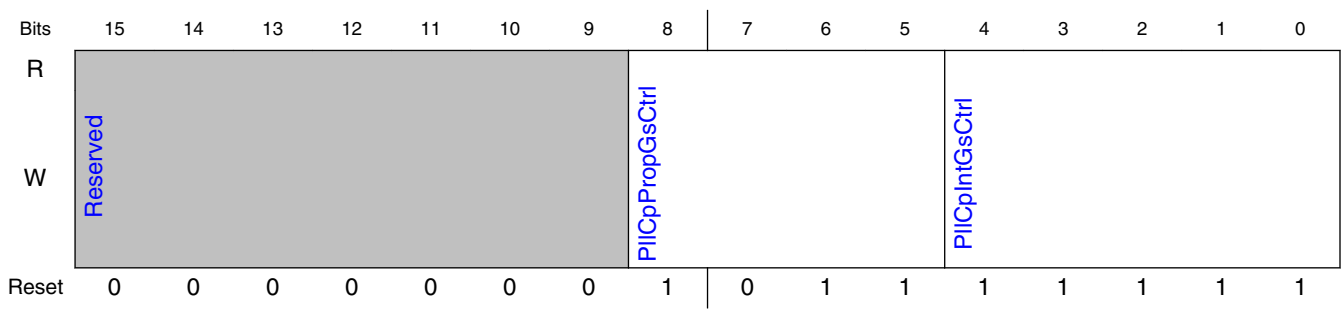
Field	Function
15-0 PllTestMode_p2	It is required to use default values for this CSR unless directed otherwise by Synopsys. It is required to use default values for this CSR unless directed otherwise by Synopsys. Directly connected to testmode[15:0] pin of PLL

9.4.3.6.206 PState dependent PLL Control Register 4 (PllCtrl4_p2)

9.4.3.6.206.1 Offset

Register	Offset
PllCtrl4_p2	40_0198h

9.4.3.6.206.2 Diagram



9.4.3.6.206.3 Fields

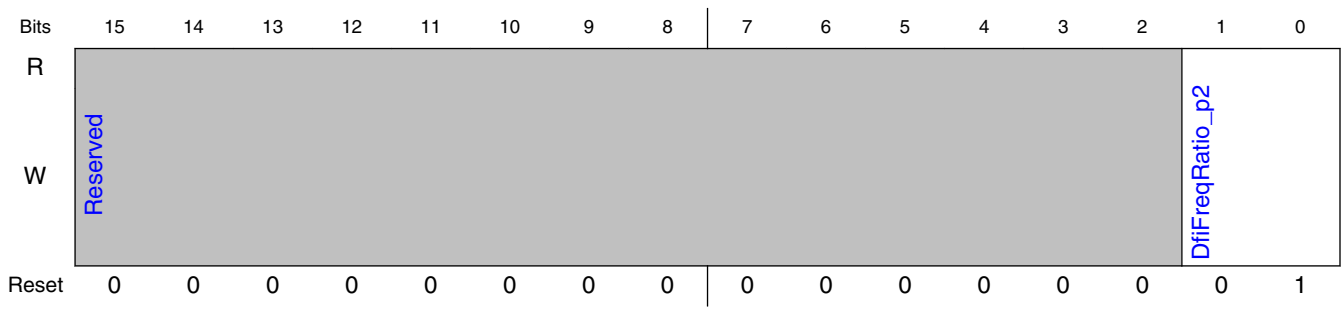
Field	Function
15-9 —	Reserved
8-5 PllCpPropGsCtrl	connects directly to cp_prop_gs_cntrl<3:0> of PLL. connects directly to cp_prop_gs_cntrl<3:0> of PLL. Charge pump proportional current control for fast relock and gearshift.
4-0 PllCpIntGsCtrl	connects directly to cp_int_gs_cntrl<4:0> in PLL. connects directly to cp_int_gs_cntrl<4:0> in PLL. Charge pump integrating current control for fast relock and gearshift.

9.4.3.6.207 DFI Frequency Ratio (DfiFreqRatio_p2)

9.4.3.6.207.1 Offset

Register	Offset
DfiFreqRatio_p2	40_01F4h

9.4.3.6.207.2 Diagram



9.4.3.6.207.3 Fields

Field	Function
15-2 —	Reserved
1-0 DfiFreqRatio_p2	Used in dwc_ddrphy_pub_serdes to serialize or de-serialize DFI signals 00 = 1:1 mode 01 = 1:2 mode 1x = 1:4 mode* *Note: 1:4 is for future pub revision.

9.4.3.6.208 Impedance Calibration Clock Ratio (CalUclkInfo_p3)

9.4.3.6.208.1 Offset

Register	Offset
CalUclkInfo_p3	60_0010h

9.4.3.6.208.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						CalUClkTicksPer1uS									
Reset	0	0	0	0	0	0	1	1	0	0	1	0	0	0	0	0

9.4.3.6.208.3 Fields

Field	Function
15-10 —	Reserved
9-0 CalUClkTicksPer1uS	Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. Must be programmed to the number of DfiClks in 1us (rounded up), with minimum value of 24. if (DfiClk < 24MHz) CalUclkInfo = 24 else CalUclkInfo = (number of DfiClks in 1us)

9.4.3.6.209 PHY Initialization Engine (PIE) Delay Register 0 (Seq0BDLY0_p3)

9.4.3.6.209.1 Offset

Register	Offset
Seq0BDLY0_p3	60_0016h

9.4.3.6.209.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY0_p3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.209.3 Fields

Field	Function
15-0 Seq0BDLY0_p3	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value.

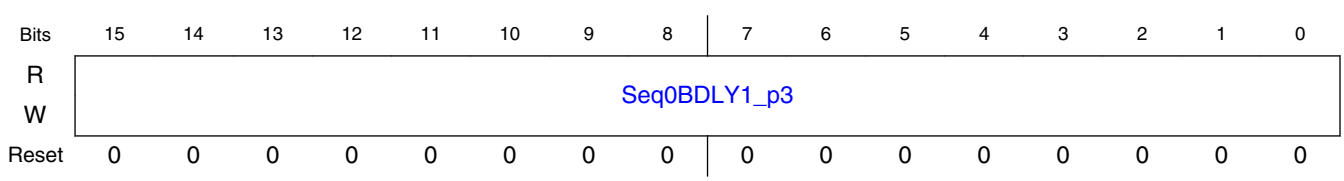
Field	Function
	PHY Initialization Engine (PIE) Delay Register 0 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.210 PHY Initialization Engine (PIE) Delay Register 1 (Seq0BDLY1_p3)

9.4.3.6.210.1 Offset

Register	Offset
Seq0BDLY1_p3	60_0018h

9.4.3.6.210.2 Diagram



9.4.3.6.210.3 Fields

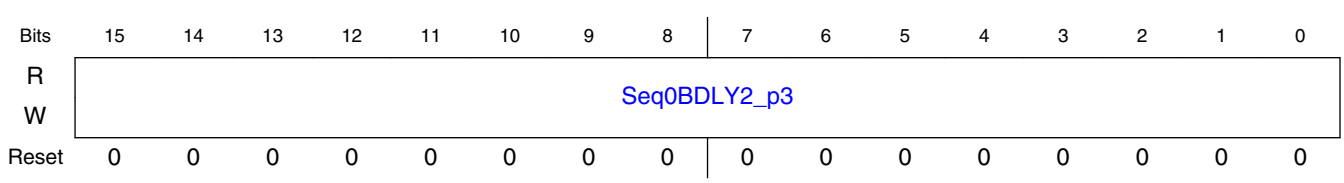
Field	Function
15-0 Seq0BDLY1_p3	PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 1 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.211 PHY Initialization Engine (PIE) Delay Register 2 (Seq0BDLY2_p3)

9.4.3.6.211.1 Offset

Register	Offset
Seq0BDLY2_p3	60_001Ah

9.4.3.6.211.2 Diagram



9.4.3.6.211.3 Fields

Field	Function
15-0 Seq0BDLY2_p3	PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 2 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.212 PHY Initialization Engine (PIE) Delay Register 3 (Seq0BDLY3_p3)

9.4.3.6.212.1 Offset

Register	Offset
Seq0BDLY3_p3	60_001Ch

9.4.3.6.212.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Seq0BDLY3_p3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

9.4.3.6.212.3 Fields

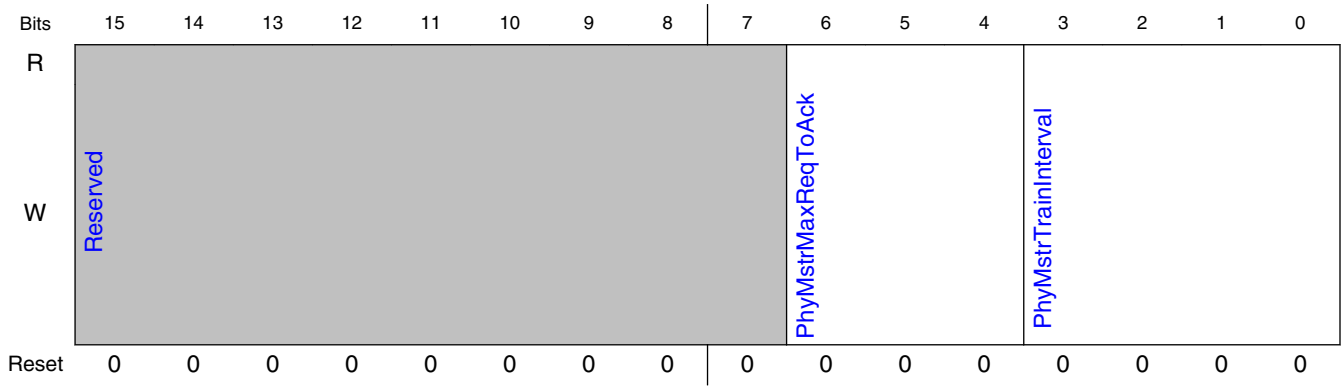
Field	Function
15-0 Seq0BDLY3_p3	PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. PHY Initialization Engine (PIE) Delay Register 3 This register is available for selection by the NOP and WAIT instructions in the PIE for the delay value. Programmed by dwc_ddrphy_phyinit_I_loadPIEImage() to support frequency changes. Programming Seq0BDLY[3,2,1,0] Registers: --- These registers are used to control various delays during PLL, DLL initialization. These regiseters should be programmed with the corresponding number of MEMCLK cycles representing a fixed amount of delay irrespective of DfiClk Frequency. When calculating the number of cycles to be programmed for each PState, it should be noted that the sequencer multiplies programmed values by a factor of 8. Register Required Delay Programmed Value --- --- --- Seq0BDLY0 0.5us (0.5us / MEMCLK period) / 8 Seq0BDLY1 1.0us (1.0us / MEMCLK period) / 8 Seq0BDLY2 10.0us (10.0us / MEMCLK period) / 8 64 if (MEMCLK Freq <= 400MHz) Seq0BDLY3 - 132 if (400MHz Freq < MEMCLK Freq <= 533MHz) 176 if (MEMCLK Freq > 533MHz)

9.4.3.6.213 Setup Intervals for DFI PHY Master operations (PPTTrainSetup_p3)

9.4.3.6.213.1 Offset

Register	Offset
PPTTrainSetup_p3	60_0020h

9.4.3.6.213.2 Diagram



9.4.3.6.213.3 Fields

Field	Function
15-7 —	Reserved
6-4 PhyMstrMaxReqToAck	Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. Bits 6:4 of this register specify the max time from tdfi_phymstr_req asserted to tdfi_phymstr_ack asserted. 3'b000 Disable PHY Master Interface 3'b001 set tPHYMSTR_resp. = 512 MEMCLKs 3'b010 set tPHYMSTR_resp. = 1024 MEMCLKs 3'b011 set tPHYMSTR_resp. = 2048 MEMCLKs 3'b100 set tPHYMSTR_resp. = 4096 MEMCLKs 3'b101 set tPHYMSTR_resp. = 8192 MEMCLKs 3'b110 set tPHYMSTR_resp. = 32768 MEMCLKs 3'b111 set tPHYMSTR_resp. = undefined
3-0 PhyMstrTrainInterval	Bits 3:0 of this register specifies the time between the end of one training and the start of the next. Bits 3:0 of this register specifies the time between the end of one training and the start of the next. (As per section 11 of the Preliminary DFI 4.0 Specification, V2, it is the max expected time from dfi_init_complete asserted to tdfi_phymstr_ack asserted). 4'b0000 Disable PHY Master Interface 4'b0001 PHY MASTER Request Interval = 524288 MEMCLKs 4'b0010 PHY MASTER Request Interval = 1048576 MEMCLKs

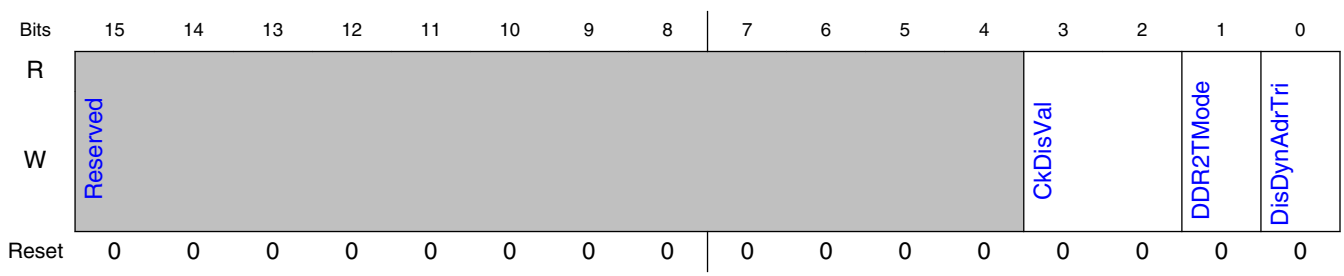
Field	Function
	4'b0011 PHY MASTER Request Interval = 2097152 MEMCLKs
	4'b0100 PHY MASTER Request Interval = 4194304 MEMCLKs
	4'b0101 PHY MASTER Request Interval = 8388608 MEMCLKs
	4'b0110 PHY MASTER Request Interval = 16777216 MEMCLKs
	4'b0111 PHY MASTER Request Interval = 33554432 MEMCLKs
	4'b1000 PHY MASTER Request Interval = 67108864 MEMCLKs
	4'b1001 PHY MASTER Request Interval = 134217728 MEMCLKs
	4'b1010 PHY MASTER Request Interval = 268435456 MEMCLKs
	4'b1011 - 4'b1111 PHY MASTER Request Interval = undefined

9.4.3.6.214 Mode select register for MEMCLK/Address/Command Tristates (TristateModeCA_p3)

9.4.3.6.214.1 Offset

Register	Offset
TristateModeCA_p3	60_0032h

9.4.3.6.214.2 Diagram



9.4.3.6.214.3 Fields

Field	Function
15-4 —	Reserved
3-2 CkDisVal	The PHY provides 4 memory clocks, n=0. The PHY provides 4 memory clocks, n=0..3. When the toggling of memory clock CK_t[n] is disabled with dfi_clk_disable[n]=1,

Table continues on the next page...

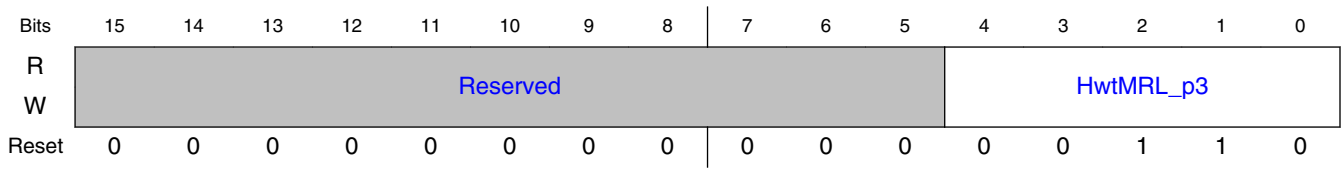
Field	Function
	the memory clock CK_t[n] is driven with Register CkDisVal[1]. When the toggling of memory clock CK_c[n] is disabled with dfi_clk_disable[n]=1, the memory clock CK_c[n] is driven with ~(Register CkDisVal[0]) (ie inverted). Note that the non-toggling differential memory clock CK_t[n],CK_c[n] may be driven with any combination, LL,LH,HL,HH; the default CK_t[n],CK_c[n]=LH.
1 DDR2TMode	Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. Must be set to 1 for Dynamic Tristate to work when CA bus is 2T or Geardown mode. This CSR bit has no effect when DisDynAdrTri==1
0 DisDynAdrTri	When DisDynAdrTri=1, Dynamic Tristating is disabled. When DisDynAdrTri=1, Dynamic Tristating is disabled. Dynamic Tristating is on by default. . Dynamic Tristating should be disabled (DisDynAdrTri=0) for these modes: 1. DDR3/2T if the controller cannot follow the 2T PHY tristate protocol. 2. DDR4/2T/2N if the controller cannot follow the 2T PHY tristate protocol. 3. LPDDR4 . When DisDynAdrTri=0 (default), In DDR3 mode, The following SDRAM pins can be dynamically tristated: A,BA,RAS_n,CAS_n,WE_n In DDR4 mode, The following SDRAM pins can be dynamically tristated: A,BA,BG,ACT_n In LPDDR3 mode, The following SDRAM pins can be dynamically tristated: CA[*] . In 1T mode, the PHY will tristate the relevant pins when all ranks are DESelected . In 2T mode (or geardown), the PHY will tristate the relevant pins when: D3 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0 = 1,1,1,0 D4 -- All ranks DESelected (all CS_L==1) AND RAS_n,CAS_n,WE_n,BA0,ACT_n = 1,1,1,0,1 When in this mode, the controller should avoid sending the above patterns with any rank selected. [e.g. NOPs sent by the controller should have BA0 set to a non-zero value]

9.4.3.6.215 HWT MaxReadLatency. (HwtMRL_p3)

9.4.3.6.215.1 Offset

Register	Offset
HwtMRL_p3	60_0040h

9.4.3.6.215.2 Diagram



9.4.3.6.215.3 Fields

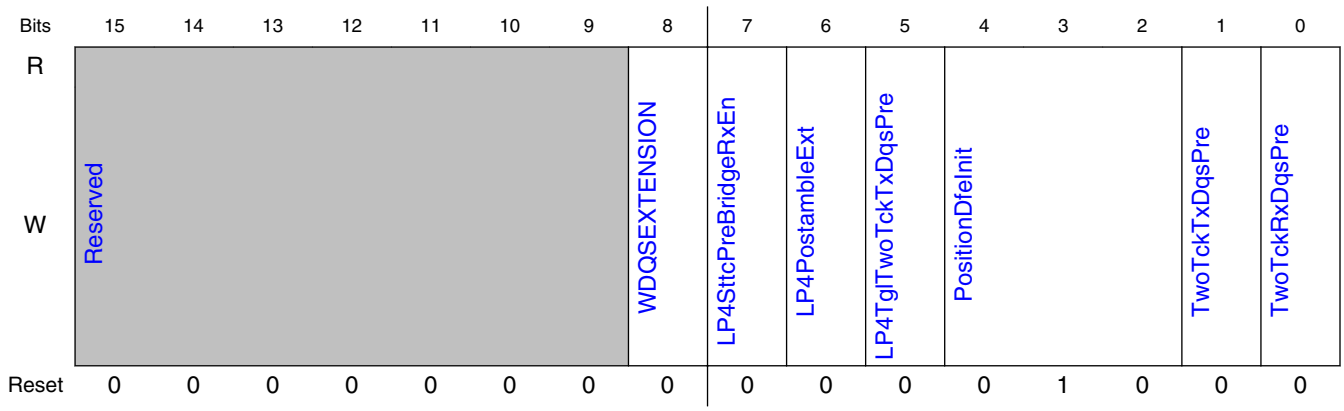
Field	Function
15-5 —	Reserved
4-0 HwtMRL_p3	This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. This Max Read Latency CSR is to be trained to ensure the rx-data fifo is not read until after all dbytes have their read data valid. The CSR is in units of two mem clocks; that is, a unit change in the LSB is a change in MRL of two mem clocks. This MASTER copy of MRL is used by the PHY training hardware only.

9.4.3.6.216 Control the PHY logic related to the read and write DQS preamble (DqsPreambleControl_p3)

9.4.3.6.216.1 Offset

Register	Offset
DqsPreambleControl_p3	60_0048h

9.4.3.6.216.2 Diagram



9.4.3.6.216.3 Fields

Field	Function
15-9 —	Reserved
8 WDQSEXTENSION	When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. When set, DQS_T and DQS_C will be driven differentially to 0 and 1, respectively, before and after a write burst, except during a memory read transaction. See DesignWare Cores LPDDR4 MultiPHY: WDQS Extension Application Note. Use of WDQSEXTENSION requires POdtTailWidth=3 and POdtStartDelay=00
7 LP4SttcPreBridgeRxEn	Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. Used in LPDDR4 static-preamble mode to bridge the RxEn between two reads to the same timing group when the bubble is 1 memclk. LPDDR4 static-preamble mode is set in the DRAMs with MR1, OP[3]=0.
6 LP4PostambleExt	In LPDDR4 mode must be set to extend the write postamble. In LPDDR4 mode must be set to extend the write postamble. 0: half-memclk postamble, as in D4 1: one-and-one-half-memclk postamble; see LPDDR4 Spec MR3, OP[1] WR PST, vendor-specific function.
5 LP4TglTwoTckTxDqsPre	Used in LPDDR4 mode to modify the early preamble when Register TwoTckTxDqsPre=1 0: level first-memclk preamble 1: toggling first-memclk preamble
4-2 PositionDfeInit	For DDR4 phy only when receive DFE is enabled. For DDR4 phy only when receive DFE is enabled. PositionDfeInit[2]=1 causes DfeInit to be asserted late PositionDfeInit[1]=1 causes DfeInit to be asserted at the nominal time

Table continues on the next page...

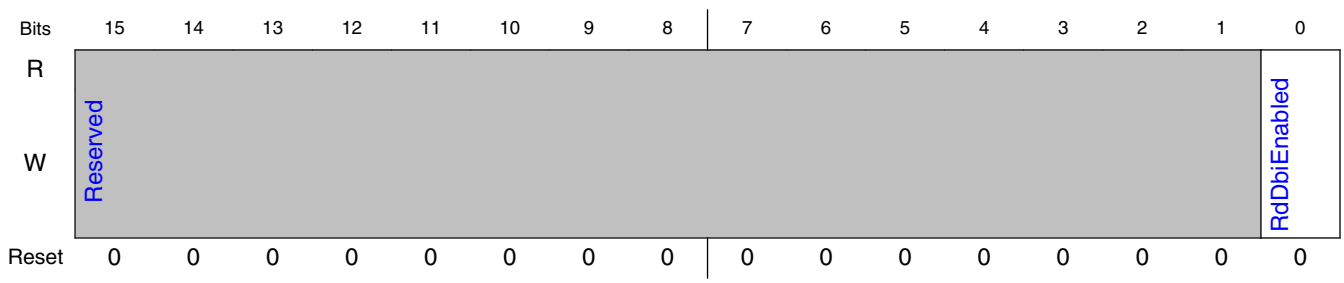
Field	Function
	PositionDfeInit[0]=1 causes DfeInit to be asserted early PositionDfeInit=3'b010 is the nominal, recommended value for leading edge used by receiver.
1 TwoTckTxDqsPre	0: Standard 1tck TxDqs Preamble 1: Enable Optional D4 2tck TxDqs Preamble The DDR4 MR4 A12 is Write Preamble, 1=2nCK, 0=1nCK.
0 TwoTckRxDqsPre	Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. Widens the RxDqsEn window to allow larger drift in the incoming read DQS to take advantage of the larger/wider preamble generated by the DRAMSs when the D4 DRAMS are configured with DDR4 MR4 A11 Read Preamble=1 for causing a 2nCK read preamble. For LPDDR4, all read operations are 2nCK such that this control must be set to 1. Note the nominal trained-to-center point will be earlier relative to the first strobing edge of DQS than when not in this mode.

9.4.3.6.217 This Register is used to enable the Read-DBI function in each DBYTE (DMIPinPresent_p3)

9.4.3.6.217.1 Offset

Register	Offset
DMIPinPresent_p3	60_005Ah

9.4.3.6.217.2 Diagram



9.4.3.6.217.3 Fields

Field	Function
15-1	Reserved

Table continues on the next page...

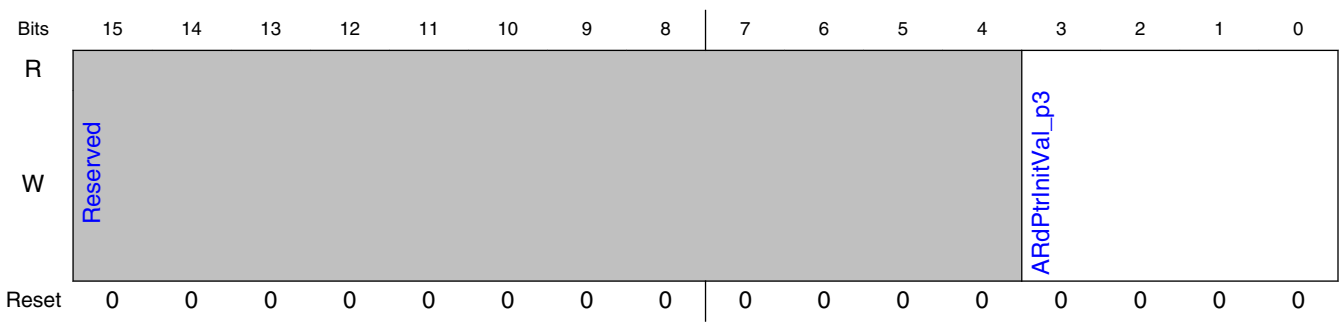
Field	Function
—	
0 RdDbiEnabled	This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. This bit must be set to 1'b1 if Read-DBI is enabled in a connected DDR4 or LPDDR4 device. If set, the following DRAM MR should also be set [DDR4.MR5.A12=1 or LPDDR4.MR3.OP[7]=1]

9.4.3.6.218 Address/Command FIFO ReadPointer Initial Value (ARdPtrInitVal_p3)

9.4.3.6.218.1 Offset

Register	Offset
ARdPtrInitVal_p3	60_005Ch

9.4.3.6.218.2 Diagram



9.4.3.6.218.3 Fields

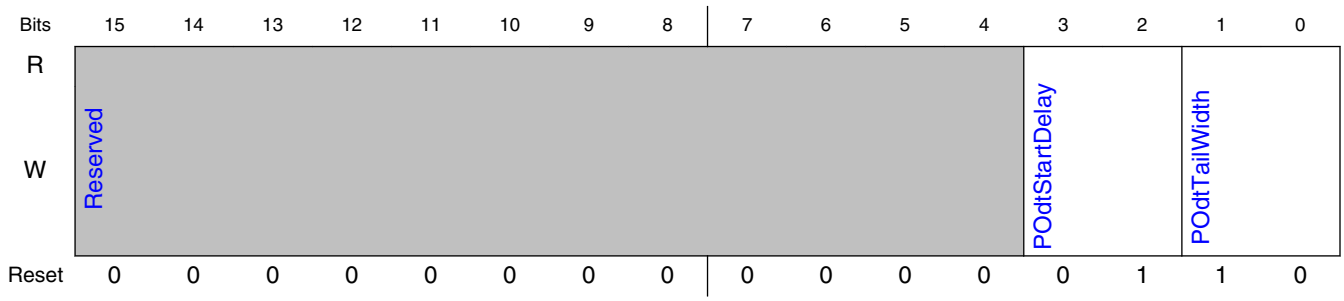
Field	Function
15-4 —	Reserved
3-0 ARdPtrInitVal_p3	This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. This is the initial Pointer Offset for the free-running FIFOs in the DBYTE and ACX4 hardips. The units of this register are in UI. The pointer separation should be chosen to compensate for all sources of skew and drift of the PHY DFICLK and PCLK networks. Please see PUB databook section 8.1.1 This CSR must be programmed in Step C of the PHY Initialization sequence

9.4.3.6.219 READ DATA On-Die Termination Timing Control (by PHY) (ProcOdtTimeCtl_p3)

9.4.3.6.219.1 Offset

Register	Offset
ProcOdtTimeCtl_p3	60_00ACh

9.4.3.6.219.2 Diagram



9.4.3.6.219.3 Fields

Field	Function
15-4 —	Reserved
3-2 POdtStartDelay	controls the start of ProcOdt, units of UI 3 delay start 2 UI, maximum delay of start of ProcOdt 2 delay start 1 UI, 1 delay start 0 UI, default 0 early by 1 UI, The time from ProcODT assertion to opening the window to receive DQS is (10 - POdtStartDelay) UI.
1-0 POdtTailWidth	controls the length of the tail of ProcOdt, units of UI 3 tail 3UI more than for Register POdtTailWidth=0, maximum 2 tail 2UI more than for Register POdtTailWidth=0, default 1 tail 1UI more than for Register POdtTailWidth=0 0 minimum length tail The time from ProcODT to closing the window to receive DQS to ProcODT POdtTailWidth is (2 + POdtTailWidth) UI

9.4.3.6.220 DLL gain control (DllGainCtl_p3)

9.4.3.6.220.1 Offset

Register	Offset
DllGainCtl_p3	60_00F8h

9.4.3.6.220.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				DllSeedSel				DllGainTV				DllGainIV			
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	1

9.4.3.6.220.3 Fields

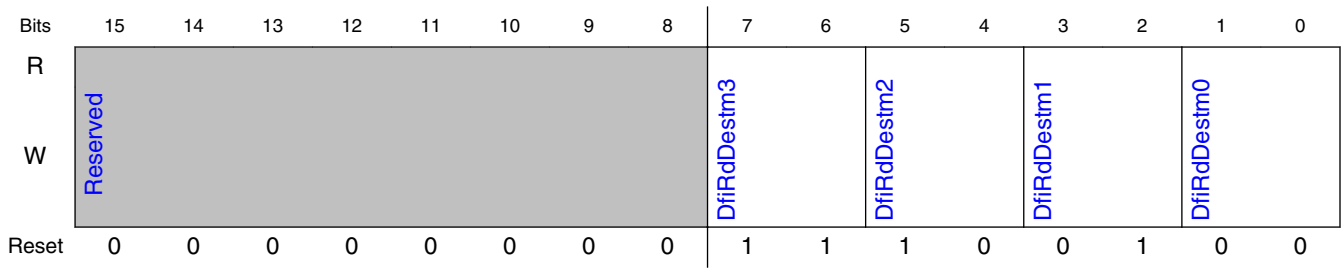
Field	Function
15-12 —	Reserved
11-8 DllSeedSel	Reserved, must be configured to be 0.
7-4 DllGainTV	Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. Terminal value of DllGain, ie the value in effect when locking is done and the value used for maintaining lock, ie tracking pclk variation. DllGainTV must be greater than or equal to DllGainIV. The maximum value is 10. The minimum value is 6.
3-0 DllGainIV	Initial value of DllGain.

9.4.3.6.221 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiRdDataCsDestMap_p3)

9.4.3.6.221.1 Offset

Register	Offset
DfiRdDataCsDestMap_p3	60_0160h

9.4.3.6.221.2 Diagram



9.4.3.6.221.3 Fields

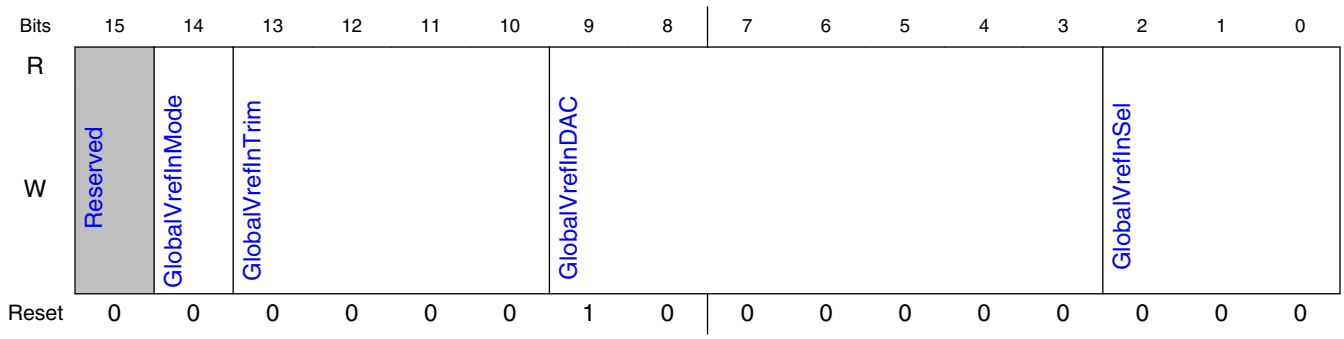
Field	Function
15-8 —	Reserved
7-6 DfiRdDestm3	Maps dfi_rddata_cs_n_p0[3] to dest DfiRdDestm3 timing For example, if 3 dfi_rddata_cs_n_p0[3] will use Register RxEn,ClkDlyTg3 timing.
5-4 DfiRdDestm2	Maps dfi_rddata_cs_n_p0[2] to dest DfiRdDestm2 timing For example, if 2 dfi_rddata_cs_n_p0[2] will use Register RxEn,ClkDlyTg2 timing.
3-2 DfiRdDestm1	Maps dfi_rddata_cs_n_p0[1] to dest DfiRdDestm1 timing For example, if 1 dfi_rddata_cs_n[_p01] will use Register RxEn,ClkDlyTg1 timing.
1-0 DfiRdDestm0	Maps dfi_rddata_cs_n_p0[0] to dest DfiRdDestm0 timing For example, if 0 dfi_rddata_cs_n_p0[0] will use Register RxEn,ClkDlyTg0 timing.

9.4.3.6.222 PHY Global Vref Controls (VrefInGlobal_p3)

9.4.3.6.222.1 Offset

Register	Offset
VrefInGlobal_p3	60_0164h

9.4.3.6.222.2 Diagram



9.4.3.6.222.3 Fields

Field	Function
15 —	Reserved
14 GlobalVrefInMode	RSVD
13-10 GlobalVrefInTrim	RSVD
9-3 GlobalVrefInDAC	DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0. DAC code for internal Vref generation The DAC has two ranges; the range is set by GlobalVrefInSel[2] ========== RANGE0 : DDR3,DDR4,LPDDR3 [GlobalVrefInSel[2] = 0] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : 0.345*VDDQ + (0.005*GlobalVrefInDAC)*VDDQ ========== RANGE1 : LPDDR4 [GlobalVrefInSel[2] = 1] DAC Output Voltage = GlobalVrefInDAC == 6'h00 ? Hi-Z : (0.005*(GlobalVrefInDAC-1))*VDDQ
2-0 GlobalVrefInSel	GlobalVrefInSel[1:0] controls the mode of the PHY VREF DAC and the BP_VREF pin ========== 2'b00 - PHY Vref DAC Range0 -- BP_VREF = Hi-Z 2'b01 - Reserved Encoding 2'b10 - PHY Vref DAC Range0 -- BP_VREF connected to PLL Analog Bus 2'b11 - PHY Vref DAC Range0 -- BP_VREF connected to PHY Vref DAC ========== GlobalVrefInSel[2] shall be set according to Dram Protocol: Protocol GlobalVrefInSel[2] ---------- DDR3 1'b0 DDR4 1'b0 LPDDR3 1'b0 LPDDR4 1'b1

9.4.3.6.223 Maps dfi_rddata_cs_n to destination dimm timing group for use with D4 LRDIMM (DfiWrDataCsDestMap_p3)

9.4.3.6.223.1 Offset

Register	Offset
DfiWrDataCsDestMap_p3	60_0168h

9.4.3.6.223.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								DfiWrDestm3		DfiWrDestm2		DfiWrDestm1		DfiWrDestm0	
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0

9.4.3.6.223.3 Fields

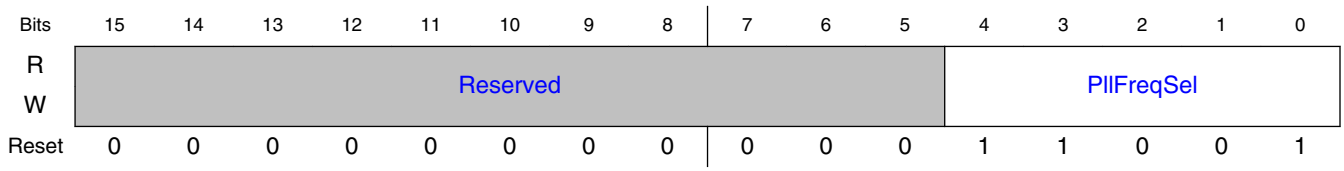
Field	Function
15-8 —	Reserved
7-6 DfiWrDestm3	Maps dfi_wrdata_cs_n_p0[3] to dest DfiWrDestm3 timing (use Register TxDq,DqsDlyTg3) For example, if 3 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg3 timing.
5-4 DfiWrDestm2	Maps dfi_wrdata_cs_n_p0[2] to dest DfiWrDestm2 timing For example, if 2 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg2 timing.
3-2 DfiWrDestm1	Maps dfi_wrdata_cs_n_p0[1] to dest DfiWrDestm1 timing For example, if 1 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg1 timing.
1-0 DfiWrDestm0	Maps dfi_wrdata_cs_n_p0[0] to dest DfiWrDestm0 timing For example, if 0 dfi_wrdata_cs_n_p0[0] will use Register TxDq,DqsDlyTg0 timing.

9.4.3.6.224 PState dependent PLL Control Register 2 (PllCtrl2_p3)

9.4.3.6.224.1 Offset

Register	Offset
PllCtrl2_p3	60_018Ah

9.4.3.6.224.2 Diagram



9.4.3.6.224.3 Fields

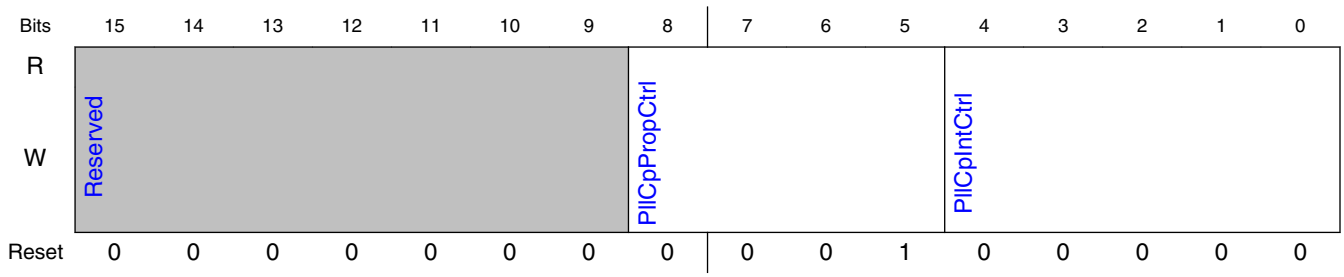
Field	Function
15-5 —	Reserved
4-0 PllFreqSel	Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Adjusts the loop parameters to compensate for different VCO bias points, and input/output clock division ratios. Program based on reference clock freqency (DfiClk) with this table.

9.4.3.6.225 PState dependent PLL Control Register 1 (PllCtrl1_p3)

9.4.3.6.225.1 Offset

Register	Offset
PllCtrl1_p3	60_018Eh

9.4.3.6.225.2 Diagram



9.4.3.6.225.3 Fields

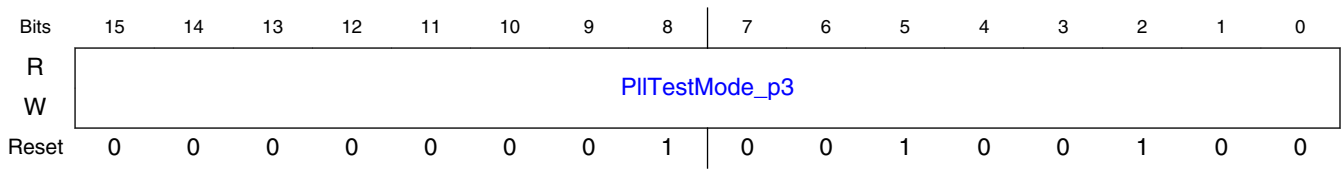
Field	Function
15-9 —	Reserved
8-5 PllCpPropCtrl	connects directly to cp_prop_cntrl<3:0> of PLL. connects directly to cp_prop_cntrl<3:0> of PLL. Charge pump proportional current control.
4-0 PllCpIntCtrl	connects directly to cp_int_cntrl<1:0> in PLL. connects directly to cp_int_cntrl<1:0> in PLL. Charge pump integrating current control.

9.4.3.6.226 Additional controls for PLL CP/VCO modes of operation (PllTestMode_p3)

9.4.3.6.226.1 Offset

Register	Offset
PllTestMode_p3	60_0194h

9.4.3.6.226.2 Diagram



9.4.3.6.226.3 Fields

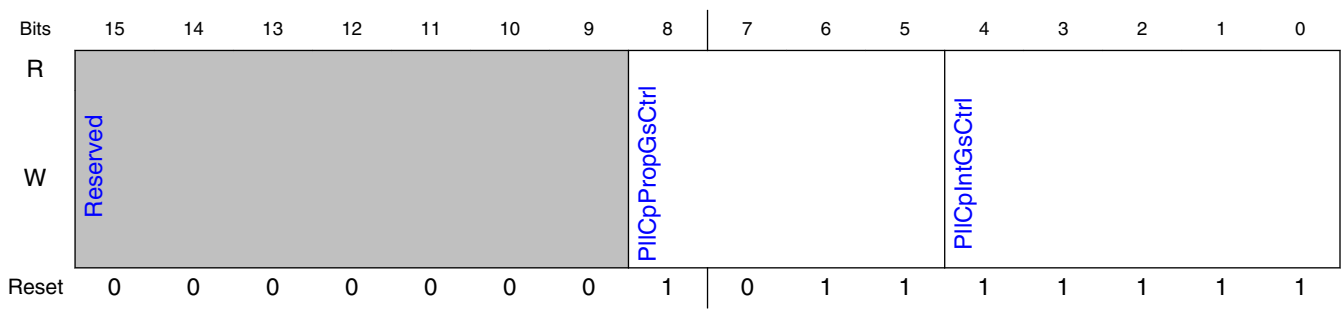
Field	Function
15-0 PllTestMode_p3	It is required to use default values for this CSR unless directed otherwise by Synopsys. It is required to use default values for this CSR unless directed otherwise by Synopsys. Directly connected to testmode[15:0] pin of PLL

9.4.3.6.227 PState dependent PLL Control Register 4 (PllCtrl4_p3)

9.4.3.6.227.1 Offset

Register	Offset
PllCtrl4_p3	60_0198h

9.4.3.6.227.2 Diagram



9.4.3.6.227.3 Fields

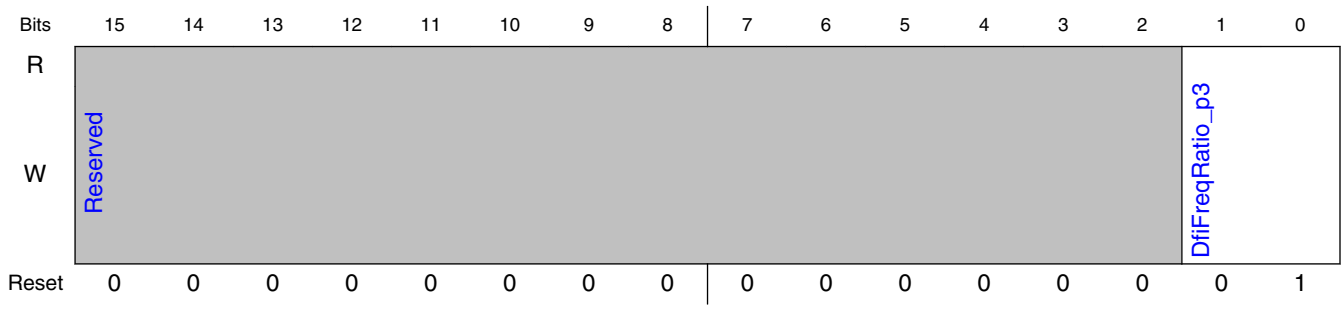
Field	Function
15-9 —	Reserved
8-5 PllCpPropGsCtrl	connects directly to cp_prop_gs_cntrl<3:0> of PLL. connects directly to cp_prop_gs_cntrl<3:0> of PLL. Charge pump proportional current control for fast relock and gearshift.
4-0 PllCpIntGsCtrl	connects directly to cp_int_gs_cntrl<4:0> in PLL. connects directly to cp_int_gs_cntrl<4:0> in PLL. Charge pump integrating current control for fast relock and gearshift.

9.4.3.6.228 DFI Frequency Ratio (DfiFreqRatio_p3)

9.4.3.6.228.1 Offset

Register	Offset
DfiFreqRatio_p3	60_01F4h

9.4.3.6.228.2 Diagram



9.4.3.6.228.3 Fields

Field	Function
15-2 —	Reserved
1-0 DfiFreqRatio_p3	Used in dwc_ddrphy_pub_serdes to serialize or de-serialize DFI signals 00 = 1:1 mode 01 = 1:2 mode 1x = 1:4 mode* *Note: 1:4 is for future pub revision.

9.5 AHB-to-APBH Bridge with DMA (APBH-Bridge-DMA)

9.5.1 Overview

The AHB-to-APBH bridge provides the chip with an inexpensive peripheral attachment bus running on the AHB's HCLK. (The H in APBH denotes that the APB*H* is synchronous to HCLK.)

As shown in the figure below, the AHB-to-APBH bridge includes the AHB-to-APB PIO bridge for a memory-mapped I/O to the APB devices, as well as a central DMA facility for devices on this bus and a vectored interrupt controller for the Arm core. Each one of the APB peripherals, including the vectored interrupt controller, is documented in their respective chapters.

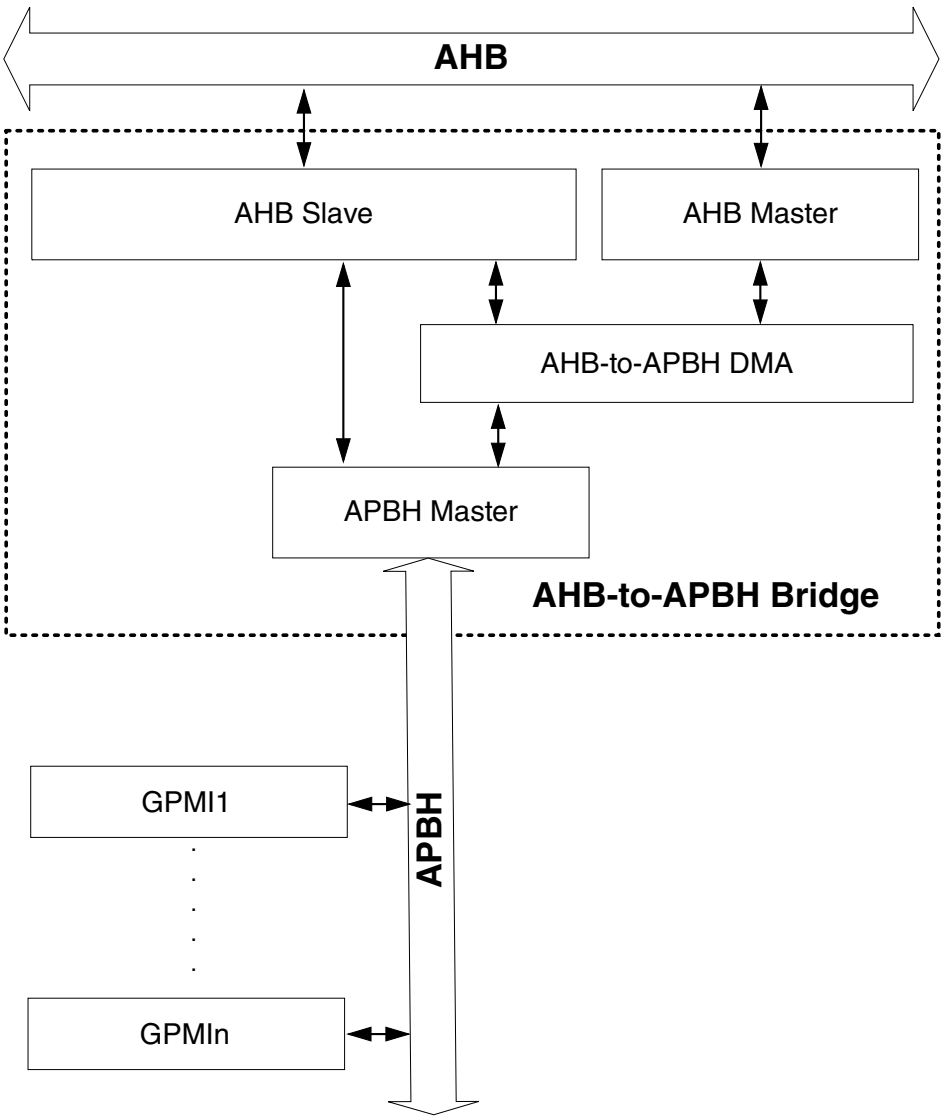


Figure 9-7. AHB-to-APBH Bridge DMA Block Diagram

The DMA controller uses the APBH bus to transfer read and write data to and from each peripheral. There is no separate DMA bus for these devices. Contention between the DMA's use of the APBH bus and the AHB-to-APB bridge functions' use of the APBH is mediated by an internal arbitration logic. For contention between these two units, the DMA is favored and the AHB slave will report "not ready" through its HREADY output until the bridge transfer can complete. The arbiter tracks repeated lockouts and inverts the priority, guaranteeing the Arm platform every fourth transfer on the APB.

9.5.2 Clocks

The table found here describes the clock sources for APBH. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 9-5. APBH Clocks

Clock name	Clock Root	Description
hclk		Module clock

9.5.3 APBH DMA

The DMA supports four channels of DMA services, as shown in the following table. The shared DMA resource allows each independent channel to follow a simple chained command list. Command chains are built up using the general structure, as shown in Figure 9-8.

Table 9-6. APBH DMA channel assignments

APBH DMA Channel #	Usage
0	GPMI0
1	GPMI1
2	GPMI2
3	GPMI3

A single command structure or channel command word specifies a number of operations to be performed by the DMA in support of a given device. Thus, the Arm platform can set up large units of work, chaining together many DMA channel command words, pass them off to the DMA, and have no further concern for the device until the DMA completion interrupt occurs. The goal is to have enough intelligence in the DMA and the devices to keep the interrupt frequency from any device below 1 KHz (arrival intervals longer than 1 ms).

A single command structure can issue 32-bit PIO write operations to key registers in the associated device using the same APB bus and controls that it uses to write DMA data bytes to the device. For example, this allows a chain of operations to be issued to the GPMI controller to send NAND command bytes, address bytes, and data transfers where the command and the address structure is completely under software control, but the administration of that transfer is handled autonomously by the DMA. Each DMA

structure can have 0–15 PIO words appended to it. The CMDPIOWORDS field, if non-zero, instructs the DMA engine to copy these words to the APB, beginning at the first register address offset for the peripheral and incrementing the register offset each cycle.

The DMA master generates only normal read/write transfers to the APBH. It does *not* generate set, clear, or toggle (SCT) transfers.

After any requested PIO words have been transferred to the peripheral, the DMA examines the two-bit command field in the channel command structure. Table 9-7 shows the four commands implemented by the DMA.

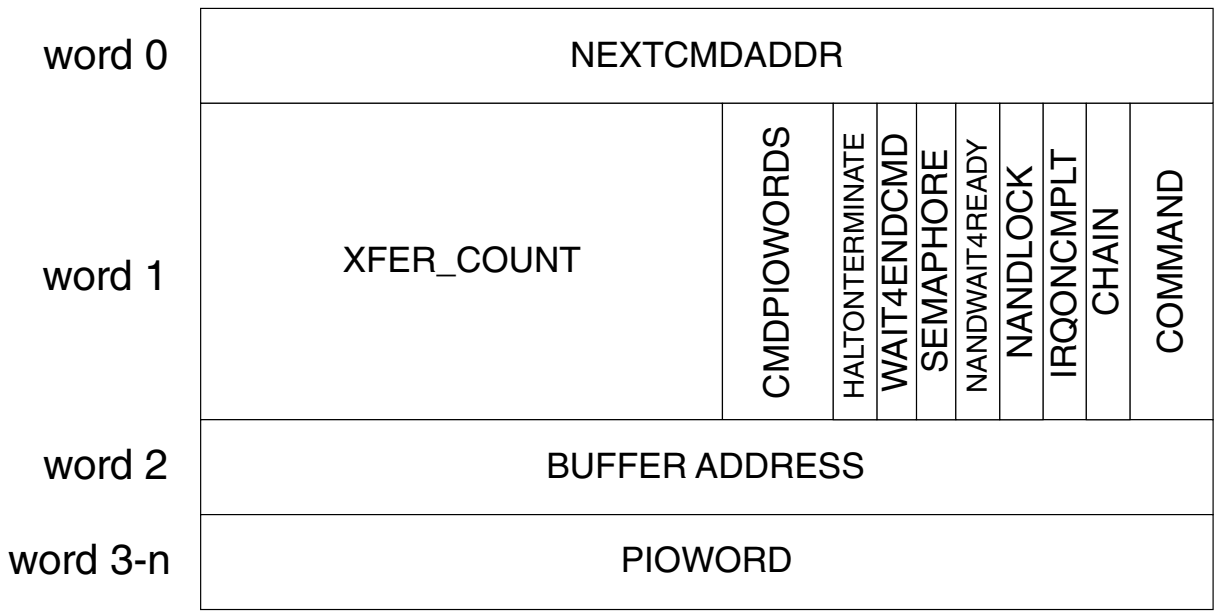


Figure 9-8. AHB-to-APBH Bridge DMA channel command structure

Table 9-7. APBH DMA commands

DMA Command	Usage
00	NO_DMA_XFER. Perform any requested PIO word transfers, but terminate the command before any DMA transfer.
01	DMA_WRITE. Perform any requested PIO word transfers, then perform a DMA transfer from the peripheral for the specified number of bytes.
10	DMA_READ. Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes.
11	DMA_SENSE. Perform any requested PIO word transfers, then perform a conditional branch to the next chained device. Follow the NEXTCMD_ADDR pointer if the peripheral sense is false. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is true. This command becomes a no-operation for any channel other than a GPMI channel.

DMA_WRITE operations copy data bytes to the system memory (on-chip RAM or SDRAM) from the associated peripheral.

DMA_READ operations copy data bytes to the APB peripheral from the system memory. The DMA engine contains a shared byte aligner that aligns bytes from system memory to or from the peripherals. Peripherals always assume little-endian-aligned data arrives or departs on their 32-bit APB. The DMA_READ transfer uses the BUFFER_ADDRESS word in the command structure to point to the DMA data buffer to be read by the DMA_READ command.

The NO_DMA_XFER command is used to write PIO words to a device without performing any DMA data byte transfers. This command is useful in such applications as activating the NAND devices CHECKSTATUS operation. The check status command reads a status byte from the NAND device, performs an XOR and MASK against an expected value supplied as part of the PIO transfer. Once the read check completes (see NAND Read Status Polling Example), the NO_DMA_XFER command completes. The result in the peripheral is that its sense line is driven by the results of the comparison. The sense flip-flop is only updated by CHECKSTATUS for the device that is executed. At some future point, the chain contains a DMA command structure with the fourth and final command value, that is, the DMA_SENSE command.

As each DMA command completes, it triggers the DMA to load the next DMA command structure in the chain. The normal flow list of DMA commands is found by following the NEXTCMD_ADDR pointer in the DMA command structure. The DMA_SENSE command uses the DMA buffer pointer word of the command structure to point to an alternate DMA command structure chain or list. The DMA_SENSE command examines the sense line of the associated peripheral. If the sense line is false, then the DMA follows the standard list found whose next command is found from the pointer in the NEXTCMD_ADDR word of the command structure. If the sense line is true, then the DMA follows the alternate list whose next command is found from the pointer in the DMA Buffer Pointer word of the DMA_SENSE command structure (see Figure 9-8). The sense command ignores the CHAIN bit, so that both pointers must be valid when the DMA comes to a sense command.

If the wait-for-end-command bit (WAIT4ENDCMD) is set in a command structure, the DMA channel waits for the device to signal completion of a command by toggling the endcmd signal before proceeding to load and execute the next command structure. Then, if DECREMENT_SEMAPHORE is set, the semaphore is decremented after the end command is seen.

A detailed bit-field view of the DMA command structure is shown in the following table, which shows a field that specifies the number of bytes to be transferred by this DMA command. The transfer-count mechanism is duplicated in the associated peripheral, either as an implied or as a specified count in the peripheral.

Table 9-8. DMA channel command word in system memory

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
NEXT_COMMAND_ADDRESS																															
Number DMA Bytes to Transfer																Number PIO Words to Write							HALTONTERMINATE	WAIT4ENDCMD	DECREMENT SEMAPHORE	NANDwAIT4READY	NANDLOCK	IRQ_COMPLETE	CHAIN	COMMAND	
DMA Buffer or Alternate CCW																															
Zero or More PIO Words to Write to the Associated Peripheral Starting at its Base Address on the APBH Bus																															

Figure 9-8 also shows the CHAIN bit in bit 2 of the second word of the command structure. This bit is set to 1, if the NEXT_COMMAND_ADDRESS contains a pointer to another DMA command structure. If a null pointer (0) is loaded into the NEXT_COMMAND_ADDRESS, it is not detected by the DMA hardware. Only the CHAIN bit indicates whether a valid list exists beyond the current structure.

If the IRQ_COMPLETE bit is set in the command structure, then the last act of the DMA before loading the next command is to set the interrupt-status bit corresponding to the current channel. The sticky interrupt request bit in the DMA CSR remains set until cleared by the software. It can be used to interrupt the Arm platform.

The NAND_LOCK bit is monitored by the DMA channel arbiter. Once a NAND channel (from channel 0 to channel 3) succeeds in the arbiter with its NAND_LOCK bit set, then the arbiter ignores the other NAND channels until a command is completed in which the NAND_LOCK is not set. Notice that the semantic here is that the NAND_LOCK state is to limit scheduling of a non-locked DMA. A DMA channel can go from unlocked to locked in the arbiter at the beginning of a command when the NAND_LOCK bit is set. When the last DMA command of an atomic sequence is completed, the lock should be removed. To accomplish this, the last command does not have the NAND_LOCK bit. It is still locked in the atomic state within the arbiter when the command starts, so that it is the only NAND command that can be executed. At the end, it drops from the atomic state within the arbiter.

The NAND_WAIT4READY bit also has a special use for GPMI channels (from channel 0 to channel 3), i.e., the NAND device channels. The GPMI peripheral supplies a sample of the ready line from the NAND device. This ready value is used to hold off of a command with this bit set until the ready line is asserted to 1. Once the arbiter sees a command with a wait-for-ready set, it holds off that channel until ready is asserted.

Receiving an IRQ for HALTONTERMINATE (HOT) is a feature in the APBH DMA descriptor that allows GPMI to signal to the DMA engine that an error has occurred. If a command is stalled due to an error, a HOT signal is sent from the peripheral to the DMA engine and causes an IRQ after terminating the DMA descriptor being executed.

Therefore, it is recommended that software use this signal as follows:

- Always set HALTONTERMINATE to 1 in a DMA descriptor. That way, if a peripheral signals HOT, the transfer will end, leaving the peripheral block and the DMA engine synchronized (but at the end of a command).
- When an IRQ from an APBH channel is received, and the IRQ is determined to be due to an error (as opposed to an IRQONCOMPLETE interrupt) the software should:
  - Reset the channel.
  - Determine the error from error reporting in the peripheral block, then manage the error in the peripheral that is attached to that channel in whatever appropriate way exists for that device (software recovery, device reset, block reset, etc).

Each channel has an eight-bit counting semaphore that controls whether it is in the idle state. When the semaphore is non-zero, the channel is ready to run, process commands and perform DMA transfers. Whenever a command finishes its DMA transfer, it checks the DECREMENT_SEMAPHORE bit. If set, it decrements the counting semaphore. If the semaphore goes to 0 as a result, then the channel enters the idle state and remains there until the semaphore is incremented by the software. When the semaphore goes to non-zero and the channel is in its idle state, then it uses the value in the APBH_CHn_NXTCMDAR register (next command address register) to fetch a pointer to the next command to process.

NOTE

This is a double indirect case. This method allows the software to append to a running command list under the protection of the counting semaphore.

To start processing the first time, software creates the command list to be processed. It writes the address of the first command into the APBH_CHn_NXTCMDAR register, and then writes 1 to the counting semaphore in APBH_CHn_SEMA. The DMA channel loads APBH_CHn_CURCMDAR register and then enters the normal state machine processing for the next command. When the software writes a value to the counting semaphore, it is added to the semaphore count by hardware, protecting the case where both hardware and software are trying to change the semaphore on the same clock edge.

Software can examine the value of APBH_CHn_CURCMDAR at any time to determine the location of the command structure currently being processed.

9.5.4 NAND Read Status Polling Example

The following figure shows a more complicated scenario.

This subset of a NAND device workload shows that the first two command structures are used during the data-write phase of an NAND device write operation (CLE and ALE transfers omitted for clarity).

- After writing the data, one must wait until the NAND device status register indicates that the write charge has been transferred. This is built into the workload using a check status command in the NAND in a loop created from the next two DMA command structures.
- The NO_DMA_TRANSFER command is shown here performing the read check, followed by a DMA_SENSE command to branch the DMA command structure list, based on the status of a bit in the external NAND device.

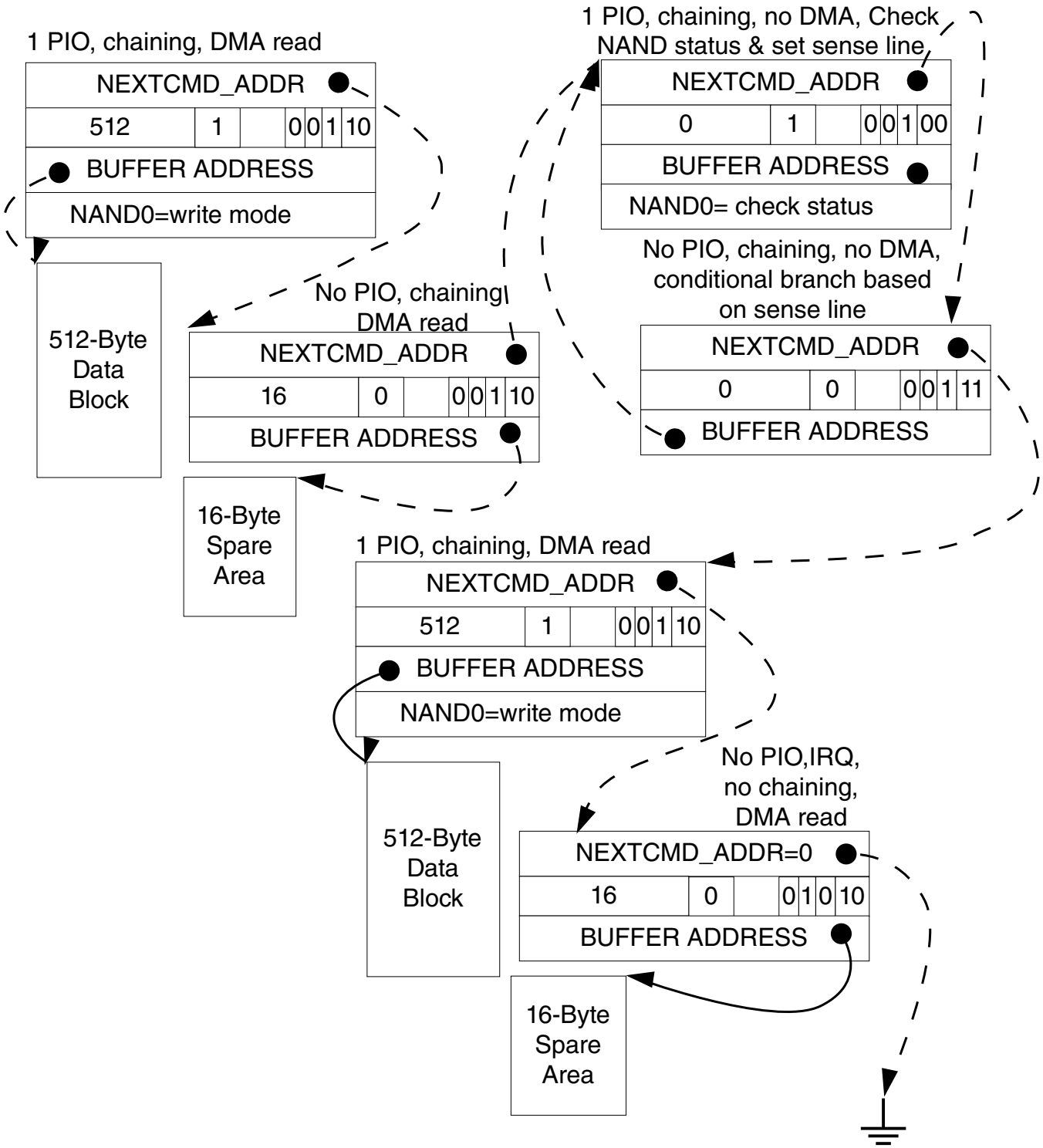


Figure 9-9. AHB-to-APBH Bridge DMA NAND Read Status Polling with DMA Sense Command

The example in the above figure shows the workload continuing immediately to the next NAND page transfer. However, one could perform a second sense operation to see if an error has occurred after the write. One could then point the sense command alternate branch at a NO_DMA_XFER command with the interrupt bit set. If the CHAIN bit is not

set on this failure branch, then the Arm platform is interrupted immediately, and the channel process is also immediately terminated in the presence of a workload-detected NAND error bit.

Note that each word of the three-word DMA command structure corresponds to a PIO register of the DMA that is accessible on the APBH bus. Normally, the DMA copies the next command structure onto these registers for processing at the start of each command by following the value of the pointer previously loaded into the NEXTCMD_ADDR register.

To start DMA processing for the first command, initialize the PIO registers of the desired channel, as follows:

- First, load the next command address register with a pointer to the first command to be loaded.
- Then, write 1 to the counting semaphore register. This causes the DMA to schedule the targeted channel for the DMA command structure load, just as if it had finished its previous command.

9.5.5 APBH Memory Map/Register Definition

APBH Hardware Register Format Summary

APBH memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_0000	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0)	32	R/W	E000_0000h	9.5.5.1/ 2357
3300_0004	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_SET)	32	R/W	E000_0000h	9.5.5.1/ 2357
3300_0008	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_CLR)	32	R/W	E000_0000h	9.5.5.1/ 2357
3300_000C	AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0_TOG)	32	R/W	E000_0000h	9.5.5.1/ 2357
3300_0010	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1)	32	R/W	0000_0000h	9.5.5.2/ 2359
3300_0014	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_SET)	32	R/W	0000_0000h	9.5.5.2/ 2359
3300_0018	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_CLR)	32	R/W	0000_0000h	9.5.5.2/ 2359

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_001C	AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1_TOG)	32	R/W	0000_0000h	9.5.5.2/ 2359
3300_0020	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2)	32	R/W	0000_0000h	9.5.5.3/ 2362
3300_0024	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_SET)	32	R/W	0000_0000h	9.5.5.3/ 2362
3300_0028	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_CLR)	32	R/W	0000_0000h	9.5.5.3/ 2362
3300_002C	AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2_TOG)	32	R/W	0000_0000h	9.5.5.3/ 2362
3300_0030	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL)	32	R/W	0000_0000h	9.5.5.4/ 2367
3300_0034	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_SET)	32	R/W	0000_0000h	9.5.5.4/ 2367
3300_0038	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_CLR)	32	R/W	0000_0000h	9.5.5.4/ 2367
3300_003C	AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL_TOG)	32	R/W	0000_0000h	9.5.5.4/ 2367
3300_0040	AHB to APBH DMA Device Assignment Register (APBH_DEVSEL)	32	R	0000_0000h	9.5.5.5/ 2368
3300_0050	AHB to APBH DMA burst size (APBH_DMA_BURST_SIZE)	32	R/W	0055_5555h	9.5.5.6/ 2369
3300_0060	AHB to APBH DMA Debug Register (APBH_DEBUG)	32	R/W	0000_0000h	9.5.5.7/ 2370
3300_0100	APBH DMA Channel n Current Command Address Register (APBH_CH0_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0110	APBH DMA Channel n Next Command Address Register (APBH_CH0_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0120	APBH DMA Channel n Command Register (APBH_CH0_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0130	APBH DMA Channel n Buffer Address Register (APBH_CH0_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0140	APBH DMA Channel n Semaphore Register (APBH_CH0_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0150	AHB to APBH DMA Channel n Debug Information (APBH_CH0_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0160	AHB to APBH DMA Channel n Debug Information (APBH_CH0_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0170	APBH DMA Channel n Current Command Address Register (APBH_CH1_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0180	APBH DMA Channel n Next Command Address Register (APBH_CH1_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0190	APBH DMA Channel n Command Register (APBH_CH1_CMD)	32	R	0000_0000h	9.5.5.10/ 2372

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_01A0	APBH DMA Channel n Buffer Address Register (APBH_CH1_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_01B0	APBH DMA Channel n Semaphore Register (APBH_CH1_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_01C0	AHB to APBH DMA Channel n Debug Information (APBH_CH1_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_01D0	AHB to APBH DMA Channel n Debug Information (APBH_CH1_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_01E0	APBH DMA Channel n Current Command Address Register (APBH_CH2_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_01F0	APBH DMA Channel n Next Command Address Register (APBH_CH2_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0200	APBH DMA Channel n Command Register (APBH_CH2_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0210	APBH DMA Channel n Buffer Address Register (APBH_CH2_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0220	APBH DMA Channel n Semaphore Register (APBH_CH2_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0230	AHB to APBH DMA Channel n Debug Information (APBH_CH2_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0240	AHB to APBH DMA Channel n Debug Information (APBH_CH2_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0250	APBH DMA Channel n Current Command Address Register (APBH_CH3_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0260	APBH DMA Channel n Next Command Address Register (APBH_CH3_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0270	APBH DMA Channel n Command Register (APBH_CH3_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0280	APBH DMA Channel n Buffer Address Register (APBH_CH3_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0290	APBH DMA Channel n Semaphore Register (APBH_CH3_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_02A0	AHB to APBH DMA Channel n Debug Information (APBH_CH3_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_02B0	AHB to APBH DMA Channel n Debug Information (APBH_CH3_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_02C0	APBH DMA Channel n Current Command Address Register (APBH_CH4_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_02D0	APBH DMA Channel n Next Command Address Register (APBH_CH4_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_02E0	APBH DMA Channel n Command Register (APBH_CH4_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_02F0	APBH DMA Channel n Buffer Address Register (APBH_CH4_BAR)	32	R	0000_0000h	9.5.5.11/ 2374

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_0300	APBH DMA Channel n Semaphore Register (APBH_CH4_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0310	AHB to APBH DMA Channel n Debug Information (APBH_CH4_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0320	AHB to APBH DMA Channel n Debug Information (APBH_CH4_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0330	APBH DMA Channel n Current Command Address Register (APBH_CH5_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0340	APBH DMA Channel n Next Command Address Register (APBH_CH5_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0350	APBH DMA Channel n Command Register (APBH_CH5_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0360	APBH DMA Channel n Buffer Address Register (APBH_CH5_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0370	APBH DMA Channel n Semaphore Register (APBH_CH5_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0380	AHB to APBH DMA Channel n Debug Information (APBH_CH5_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0390	AHB to APBH DMA Channel n Debug Information (APBH_CH5_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_03A0	APBH DMA Channel n Current Command Address Register (APBH_CH6_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_03B0	APBH DMA Channel n Next Command Address Register (APBH_CH6_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_03C0	APBH DMA Channel n Command Register (APBH_CH6_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_03D0	APBH DMA Channel n Buffer Address Register (APBH_CH6_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_03E0	APBH DMA Channel n Semaphore Register (APBH_CH6_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_03F0	AHB to APBH DMA Channel n Debug Information (APBH_CH6_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0400	AHB to APBH DMA Channel n Debug Information (APBH_CH6_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0410	APBH DMA Channel n Current Command Address Register (APBH_CH7_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0420	APBH DMA Channel n Next Command Address Register (APBH_CH7_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0430	APBH DMA Channel n Command Register (APBH_CH7_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0440	APBH DMA Channel n Buffer Address Register (APBH_CH7_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0450	APBH DMA Channel n Semaphore Register (APBH_CH7_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_0460	AHB to APBH DMA Channel n Debug Information (APBH_CH7_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0470	AHB to APBH DMA Channel n Debug Information (APBH_CH7_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0480	APBH DMA Channel n Current Command Address Register (APBH_CH8_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0490	APBH DMA Channel n Next Command Address Register (APBH_CH8_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_04A0	APBH DMA Channel n Command Register (APBH_CH8_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_04B0	APBH DMA Channel n Buffer Address Register (APBH_CH8_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_04C0	APBH DMA Channel n Semaphore Register (APBH_CH8_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_04D0	AHB to APBH DMA Channel n Debug Information (APBH_CH8_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_04E0	AHB to APBH DMA Channel n Debug Information (APBH_CH8_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_04F0	APBH DMA Channel n Current Command Address Register (APBH_CH9_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0500	APBH DMA Channel n Next Command Address Register (APBH_CH9_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0510	APBH DMA Channel n Command Register (APBH_CH9_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0520	APBH DMA Channel n Buffer Address Register (APBH_CH9_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0530	APBH DMA Channel n Semaphore Register (APBH_CH9_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0540	AHB to APBH DMA Channel n Debug Information (APBH_CH9_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0550	AHB to APBH DMA Channel n Debug Information (APBH_CH9_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0560	APBH DMA Channel n Current Command Address Register (APBH_CH10_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0570	APBH DMA Channel n Next Command Address Register (APBH_CH10_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0580	APBH DMA Channel n Command Register (APBH_CH10_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0590	APBH DMA Channel n Buffer Address Register (APBH_CH10_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_05A0	APBH DMA Channel n Semaphore Register (APBH_CH10_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_05B0	AHB to APBH DMA Channel n Debug Information (APBH_CH10_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_05C0	AHB to APBH DMA Channel n Debug Information (APBH_CH10_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_05D0	APBH DMA Channel n Current Command Address Register (APBH_CH11_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_05E0	APBH DMA Channel n Next Command Address Register (APBH_CH11_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_05F0	APBH DMA Channel n Command Register (APBH_CH11_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0600	APBH DMA Channel n Buffer Address Register (APBH_CH11_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0610	APBH DMA Channel n Semaphore Register (APBH_CH11_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0620	AHB to APBH DMA Channel n Debug Information (APBH_CH11_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0630	AHB to APBH DMA Channel n Debug Information (APBH_CH11_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0640	APBH DMA Channel n Current Command Address Register (APBH_CH12_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0650	APBH DMA Channel n Next Command Address Register (APBH_CH12_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0660	APBH DMA Channel n Command Register (APBH_CH12_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0670	APBH DMA Channel n Buffer Address Register (APBH_CH12_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0680	APBH DMA Channel n Semaphore Register (APBH_CH12_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0690	AHB to APBH DMA Channel n Debug Information (APBH_CH12_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_06A0	AHB to APBH DMA Channel n Debug Information (APBH_CH12_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_06B0	APBH DMA Channel n Current Command Address Register (APBH_CH13_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_06C0	APBH DMA Channel n Next Command Address Register (APBH_CH13_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_06D0	APBH DMA Channel n Command Register (APBH_CH13_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_06E0	APBH DMA Channel n Buffer Address Register (APBH_CH13_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_06F0	APBH DMA Channel n Semaphore Register (APBH_CH13_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0700	AHB to APBH DMA Channel n Debug Information (APBH_CH13_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0710	AHB to APBH DMA Channel n Debug Information (APBH_CH13_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379

Table continues on the next page...

APBH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_0720	APBH DMA Channel n Current Command Address Register (APBH_CH14_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_0730	APBH DMA Channel n Next Command Address Register (APBH_CH14_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_0740	APBH DMA Channel n Command Register (APBH_CH14_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_0750	APBH DMA Channel n Buffer Address Register (APBH_CH14_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_0760	APBH DMA Channel n Semaphore Register (APBH_CH14_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_0770	AHB to APBH DMA Channel n Debug Information (APBH_CH14_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_0780	AHB to APBH DMA Channel n Debug Information (APBH_CH14_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0790	APBH DMA Channel n Current Command Address Register (APBH_CH15_CURCMDAR)	32	R	0000_0000h	9.5.5.8/ 2371
3300_07A0	APBH DMA Channel n Next Command Address Register (APBH_CH15_NXTCMDAR)	32	R/W	0000_0000h	9.5.5.9/ 2372
3300_07B0	APBH DMA Channel n Command Register (APBH_CH15_CMD)	32	R	0000_0000h	9.5.5.10/ 2372
3300_07C0	APBH DMA Channel n Buffer Address Register (APBH_CH15_BAR)	32	R	0000_0000h	9.5.5.11/ 2374
3300_07D0	APBH DMA Channel n Semaphore Register (APBH_CH15_SEMA)	32	R/W	0000_0000h	9.5.5.12/ 2375
3300_07E0	AHB to APBH DMA Channel n Debug Information (APBH_CH15_DEBUG1)	32	R	00A0_0000h	9.5.5.13/ 2376
3300_07F0	AHB to APBH DMA Channel n Debug Information (APBH_CH15_DEBUG2)	32	R	0000_0000h	9.5.5.14/ 2379
3300_0800	APBH Bridge Version Register (APBH_VERSION)	32	R/W	0301_0000h	9.5.5.15/ 2379

9.5.5.1 AHB to APBH Bridge Control and Status Register 0 (APBH_CTRL0*n*)

The APBH CTRL 0 provides overall control of the AHB to APBH bridge and DMA.

This register contains module softreset, clock gating, channel clock gating/freeze bits.

Address: 3300_0000h base + 0h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SFTRST	CLKGATE	AHB_BURST8_EN	APB_BURST_EN	Reserved											
W																
Reset	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CLKGATE_CHANNEL															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CTRL0*n* field descriptions

Field	Description
31 SFTRST	Set this bit to zero to enable normal APBH DMA operation. Set this bit to one (default) to disable clocking with the APBH DMA and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the APBH DMA block to its default state.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 AHB_BURST8_EN	Set this bit to one (default) to enable AHB 8-beat burst. Set to zero to disable 8-beat burst on AHB interface.
28 APB_BURST_EN	Set this bit to one to enable apb master do a continous transfers when a device request a burst dma. Set to zero will treat a burst dma request as 4/8 individual requests.
27–16 RSVD0	This field is reserved. Reserved, always set to zero.
CLKGATE_CHANNEL	These bits must be set to zero for normal operation of each channel. When set to one they gate off the individual clocks to the channels. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** — 0x0080 **NAND7** — 0x0100 **SSP** —

9.5.5.2 AHB to APBH Bridge Control and Status Register 1 (APBH_CTRL1*n*)

The APBH CTRL one provides overall control of the interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
                    BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);  // use bitfield write macro
                    BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;      // or, assign to register
struct's bitfield
```

Address: 3300_0000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CH15_CMDCMPLT_IRQ_EN	CH14_CMDCMPLT_IRQ_EN	CH13_CMDCMPLT_IRQ_EN	CH12_CMDCMPLT_IRQ_EN	CH11_CMDCMPLT_IRQ_EN	CH10_CMDCMPLT_IRQ_EN	CH9_CMDCMPLT_IRQ_EN	CH8_CMDCMPLT_IRQ_EN	CH7_CMDCMPLT_IRQ_EN	CH6_CMDCMPLT_IRQ_EN	CH5_CMDCMPLT_IRQ_EN	CH4_CMDCMPLT_IRQ_EN	CH3_CMDCMPLT_IRQ_EN	CH2_CMDCMPLT_IRQ_EN	CH1_CMDCMPLT_IRQ_EN	CH0_CMDCMPLT_IRQ_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH15_CMDCMPLT_IRQ	CH14_CMDCMPLT_IRQ	CH13_CMDCMPLT_IRQ	CH12_CMDCMPLT_IRQ	CH11_CMDCMPLT_IRQ	CH10_CMDCMPLT_IRQ	CH9_CMDCMPLT_IRQ	CH8_CMDCMPLT_IRQ	CH7_CMDCMPLT_IRQ	CH6_CMDCMPLT_IRQ	CH5_CMDCMPLT_IRQ	CH4_CMDCMPLT_IRQ	CH3_CMDCMPLT_IRQ	CH2_CMDCMPLT_IRQ	CH1_CMDCMPLT_IRQ	CH0_CMDCMPLT_IRQ
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CTRL1*n* field descriptions

Field	Description
31 CH15_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 15.
30 CH14_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 14.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
29 CH13_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 13.
28 CH12_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 12.
27 CH11_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 11.
26 CH10_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 10.
25 CH9_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 9.
24 CH8_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 8.
23 CH7_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 7.
22 CH6_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 6.
21 CH5_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 5.
20 CH4_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 4.
19 CH3_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 3.
18 CH2_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 2.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
17 CH1_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 1.
16 CH0_ CMDCMPLT_ IRQ_EN	Setting this bit enables the generation of an interrupt request for APBH DMA channel 0.
15 CH15_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
14 CH14_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
13 CH13_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
12 CH12_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
11 CH11_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
10 CH10_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
9 CH9_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
8 CH8_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
7 CH7_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 7. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
6 CH6_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 6. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

Table continues on the next page...

APBH_CTRL1*n* field descriptions (continued)

Field	Description
5 CH5_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 5. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
4 CH4_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 4. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
3 CH3_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 3. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
2 CH2_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 2. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
1 CH1_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 1. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.
0 CH0_ CMDCMPLT_ IRQ	Interrupt request status bit for APBH DMA channel 0. This sticky bit is set by DMA hardware and reset by software. It is ANDed with its corresponding enable bit to generate an interrupt.

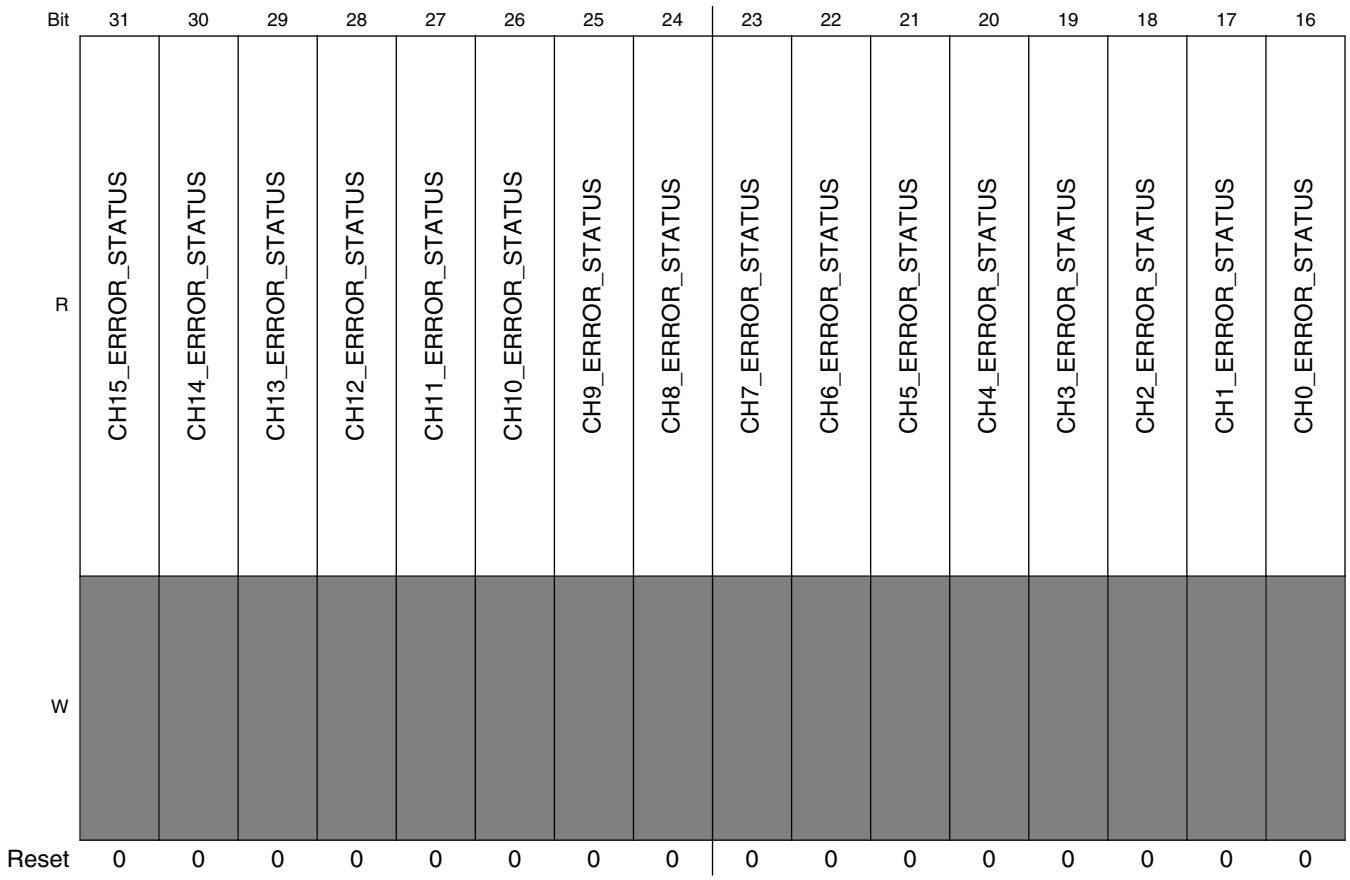
9.5.5.3 AHB to APBH Bridge Control and Status Register 2 (APBH_CTRL2*n*)

The APBH CTRL 2 provides channel error interrupts generated by the AHB to APBH DMA. This register contains the per channel interrupt status bits and the per channel interrupt enable bits. Each channel has a dedicated interrupt vector in the vectored interrupt controller.

EXAMPLE

```
                    BF_WR(APBH_CTRL1, CH5_CMDCMPLT_IRQ, 0);  // use bitfield write macro
                    BF_APBH_CTRL1.CH5_CMDCMPLT_IRQ = 0;      // or, assign to register
struct's bitfield
```

Address: 3300_0000h base + 20h offset + (4d × i), where i=0d to 3d



APBH_CTRL2*n* field descriptions

Field	Description
31 CH15_ERROR_STATUS	Error status bit for APBH DMA Channel 15. Valid when corresponding Error IRQ is set. 1 - AHB bus error

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
	0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
30 CH14_ERROR_ STATUS	Error status bit for APBH DMA Channel 14. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
29 CH13_ERROR_ STATUS	Error status bit for APBH DMA Channel 13. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
28 CH12_ERROR_ STATUS	Error status bit for APBH DMA Channel 12. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
27 CH11_ERROR_ STATUS	Error status bit for APBH DMA Channel 11. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
26 CH10_ERROR_ STATUS	Error status bit for APBH DMA Channel 10. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
25 CH9_ERROR_ STATUS	Error status bit for APBH DMA Channel 9. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
24 CH8_ERROR_ STATUS	Error status bit for APBH DMA Channel 8. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
	0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
23 CH7_ERROR_STATUS	Error status bit for APBX DMA Channel 7. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
22 CH6_ERROR_STATUS	Error status bit for APBX DMA Channel 6. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
21 CH5_ERROR_STATUS	Error status bit for APBX DMA Channel 5. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
20 CH4_ERROR_STATUS	Error status bit for APBX DMA Channel 4. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
19 CH3_ERROR_STATUS	Error status bit for APBX DMA Channel 3. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
18 CH2_ERROR_STATUS	Error status bit for APBX DMA Channel 2. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
17 CH1_ERROR_STATUS	Error status bit for APBX DMA Channel 1. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
16 CH0_ERROR_STATUS	Error status bit for APBX DMA Channel 0. Valid when corresponding Error IRQ is set. 1 - AHB bus error 0 - channel early termination. 0x0 **TERMINATION** — An early termination from the device causes error IRQ. 0x1 **BUS_ERROR** — An AHB bus error causes error IRQ.
15 CH15_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 15. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
14 CH14_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 14. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
13 CH13_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 13. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
12 CH12_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 12. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
11 CH11_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 11. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
10 CH10_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 10. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
9 CH9_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 9. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
8 CH8_ERROR_IRQ	Error interrupt status bit for APBH DMA Channel 8. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
7 CH7_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 7. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
6 CH6_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 6. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
5 CH5_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 5. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
4 CH4_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 4. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
3 CH3_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 3. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
2 CH2_ERROR_IRQ	Error interrupt status bit for APBX DMA Channel 2. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.

Table continues on the next page...

APBH_CTRL2*n* field descriptions (continued)

Field	Description
1 CH1_ERROR_ IRQ	Error interrupt status bit for APBX DMA Channel 1. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.
0 CH0_ERROR_ IRQ	Error interrupt status bit for APBX DMA Channel 0. This sticky bit is set by DMA hardware and reset by software. It is ORed with the corresponding cmdcmplt irq to generate an irq to Arm.

9.5.5.4 AHB to APBH Bridge Channel Register (APBH_CHANNEL_CTRL*n*)

The APBH CHANNEL CTRL provides reset/freeze control of each DMA channel. This register contains individual channel reset/freeze bits.

Address: 3300_0000h base + 30h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RESET_CHANNEL																FREEZE_CHANNEL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CHANNEL_CTRL*n* field descriptions

Field	Description
31–16 RESET_ CHANNEL	Setting a bit in this field causes the DMA controller to take the corresponding channel through its reset state. The bit is reset after the channel resources are cleared. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** — 0x0080 **NAND7** — 0x0100 **SSP** —
FREEZE_ CHANNEL	Setting a bit in this field will freeze the DMA channel associated with it. This field is a direct input to the DMA channel arbiter. When frozen, the channel is deined access to the central DMA resources. 0x0001 **NAND0** — 0x0002 **NAND1** — 0x0004 **NAND2** — 0x0008 **NAND3** — 0x0010 **NAND4** — 0x0020 **NAND5** — 0x0040 **NAND6** —

Table continues on the next page...

APBH_CHANNEL_CTRL*n* field descriptions (continued)

Field	Description
	0x0080 **NAND7** — 0x0100 **SSP** —

9.5.5.5 AHB to APBH DMA Device Assignment Register (APBH_DEVSEL)

This register allows reassignment of the APBH device connected to the DMA Channels.

In this chip, APBH DMA channel resource is enough for high speed peripherals, so this register is of no use and reserved.

Address: 3300_0000h base + 40h offset = 3300_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_DEVSEL field descriptions

Field	Description
31–30 CH15	This field is reserved. Reserved.
29–28 CH14	This field is reserved. Reserved.
27–26 CH13	This field is reserved. Reserved.
25–24 CH12	This field is reserved. Reserved.
23–22 CH11	This field is reserved. Reserved.
21–20 CH10	This field is reserved. Reserved.
19–18 CH9	This field is reserved. Reserved.
17–16 CH8	This field is reserved. Reserved.
15–14 CH7	This field is reserved. Reserved.
13–12 CH6	This field is reserved. Reserved.

Table continues on the next page...

APBH_DEVSEL field descriptions (continued)

Field	Description
11–10 CH5	This field is reserved. Reserved.
9–8 CH4	This field is reserved. Reserved.
7–6 CH3	This field is reserved. Reserved.
5–4 CH2	This field is reserved. Reserved.
3–2 CH1	This field is reserved. Reserved.
CH0	This field is reserved. Reserved.

9.5.5.6 AHB to APBH DMA burst size (APBH_DMA_BURST_SIZE)

This register programs the apbh burst size of the APBH DMA devices when a DMA burst request is issued.

This register provides a mechanism for assigning the device.

Address: 3300_0000h base + 50h offset = 3300_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		Reserved		CH8	
Reset	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CH7		CH6		CH5		CH4		CH3		CH2		CH1		CH0	
Reset	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1

APBH_DMA_BURST_SIZE field descriptions

Field	Description
31–30 CH15	This field is reserved. Reserved.
29–28 CH14	This field is reserved. Reserved.
27–26 CH13	This field is reserved. Reserved.
25–24 CH12	This field is reserved. Reserved.
23–22 CH11	This field is reserved. Reserved.

Table continues on the next page...

APBH_DMA_BURST_SIZE field descriptions (continued)

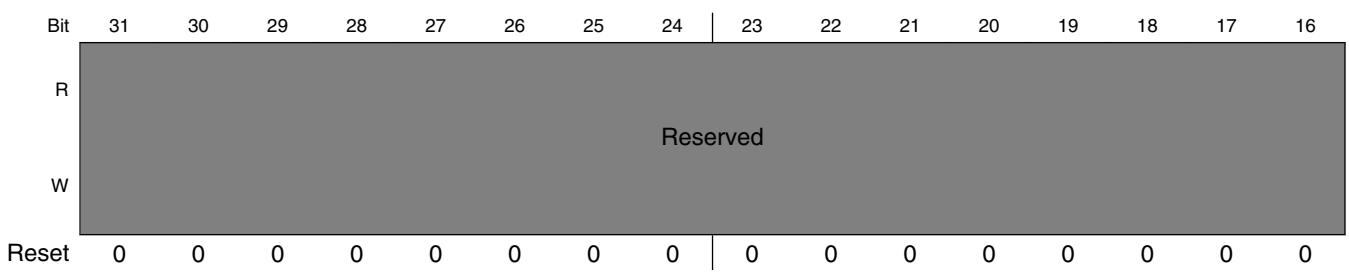
Field	Description
21–20 CH10	This field is reserved. Reserved.
19–18 CH9	This field is reserved. Reserved.
17–16 CH8	DMA burst size for SSP. 0x0 **BURST0** — 0x1 **BURST4** — 0x2 **BURST8** —
15–14 CH7	DMA burst size for GPMI channel 7. Do not change. GPMI only support burst size 4.
13–12 CH6	DMA burst size for GPMI channel 6. Do not change. GPMI only support burst size 4.
11–10 CH5	DMA burst size for GPMI channel 5. Do not change. GPMI only support burst size 4.
9–8 CH4	DMA burst size for GPMI channel 4. Do not change. GPMI only support burst size 4.
7–6 CH3	DMA burst size for GPMI channel 3. Do not change. GPMI only support burst size 4.
5–4 CH2	DMA burst size for GPMI channel 2. Do not change. GPMI only support burst size 4.
3–2 CH1	DMA burst size for GPMI channel 1. Do not change. GPMI only support burst size 4.
CH0	DMA burst size for GPMI channel 0. Do not change. GPMI only support burst size 4.

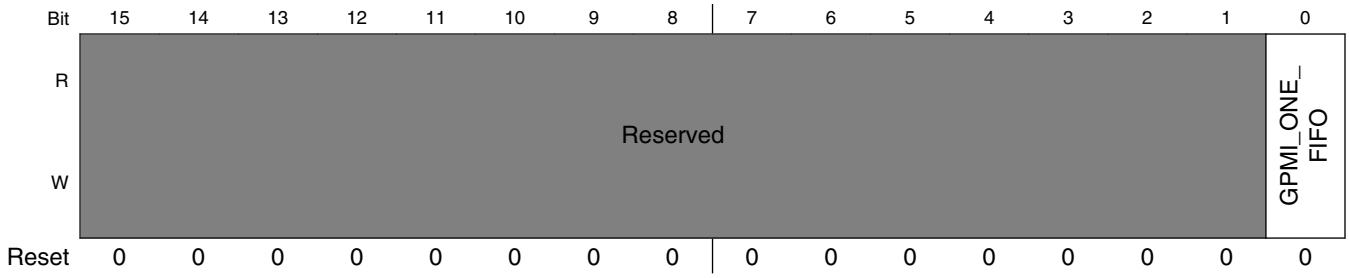
9.5.5.7 AHB to APBH DMA Debug Register (APBH_DEBUG)

This register is for debug purpose.

The debug register is for internal use only. Not recommend for customer useage.

Address: 3300_0000h base + 60h offset = 3300_0060h





APBH_DEBUG field descriptions

Field	Description
31–1 -	This field is reserved. Reserved, always set to zero.
0 GPMI_ONE_FIFO	Set to 0ne and the 8 GPMI channels will share the DMA FIFO, and when set to zero, the 8 GPMI channels will use its own DMA FIFO.

9.5.5.8 APBH DMA Channel n Current Command Address Register (APBH_CH*n*_CURCMDAR)

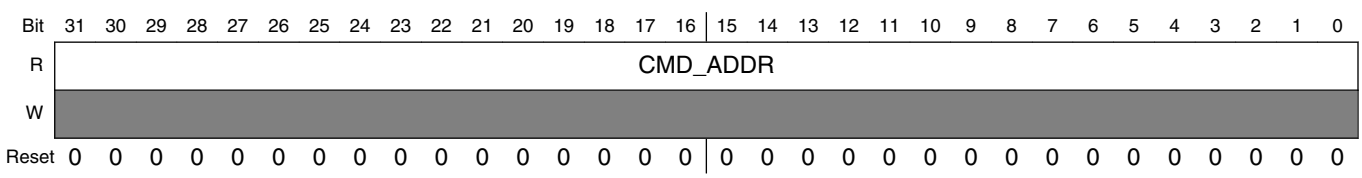
The APBH DMA channel n current command address register points to the multiword command that is currently being executed. Commands are threaded on the command address.

APBH DMA Channel n is controlled by a variable sized command structure. This register points to the command structure currently being executed.

EXAMPLE

```
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR_RD(0);              // read the whole
register, since there is only one field
pCurCmd = (apbh_chn_cmd_t *) BF_RDn(APBH_CHn_CURCMDAR, 0, CMD_ADDR);  // or, use multi-
register bitfield read macro
pCurCmd = (apbh_chn_cmd_t *) APBH_CHn_CURCMDAR(0).CMD_ADDR;        // or, assign from
bitfield of indexed register's struct
```

Address: 3300_0000h base + 100h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_CURCMDAR field descriptions

Field	Description
CMD_ADDR	Pointer to command structure currently being processed for channel n.

9.5.5.9 APBH DMA Channel n Next Command Address Register (APBH_CH*n*_NXTCMDAR)

The APBH DMA Channel n Next Command Address register contains the address of the next multiword command to be executed. Commands are threaded on the command address. Set CHAIN to 1 in the DMA command word to process command lists.

APBH DMA Channel n is controlled by a variable sized command structure. Software loads this register with the address of the first command structure to process and increments the Channel n semaphore to start processing. This register points to the next command structure to be executed when the current command is completed.

EXAMPLE

```
APBH_CHn_NXTCMDAR_WR(0, (reg32_t) pCommandTwoStructure);         // write the entire
register, since there is only one field
BF_WRn(APBH_CHn_NXTCMDAR, 0, (reg32_t) pCommandTwoStructure);       // or, use multi-
register bitfield write macro
APBH_CHn_NXTCMDAR(0).CMD_ADDR = (reg32_t) pCommandTwoStructure;  // or, assign to bitfield
of indexed register's struct
```

Address: 3300_0000h base + 110h offset + (112d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CH*n*_NXTCMDAR field descriptions

Field	Description
CMD_ADDR	Pointer to next command structure for channel n.

9.5.5.10 APBH DMA Channel n Command Register (APBH_CH*n*_CMD)

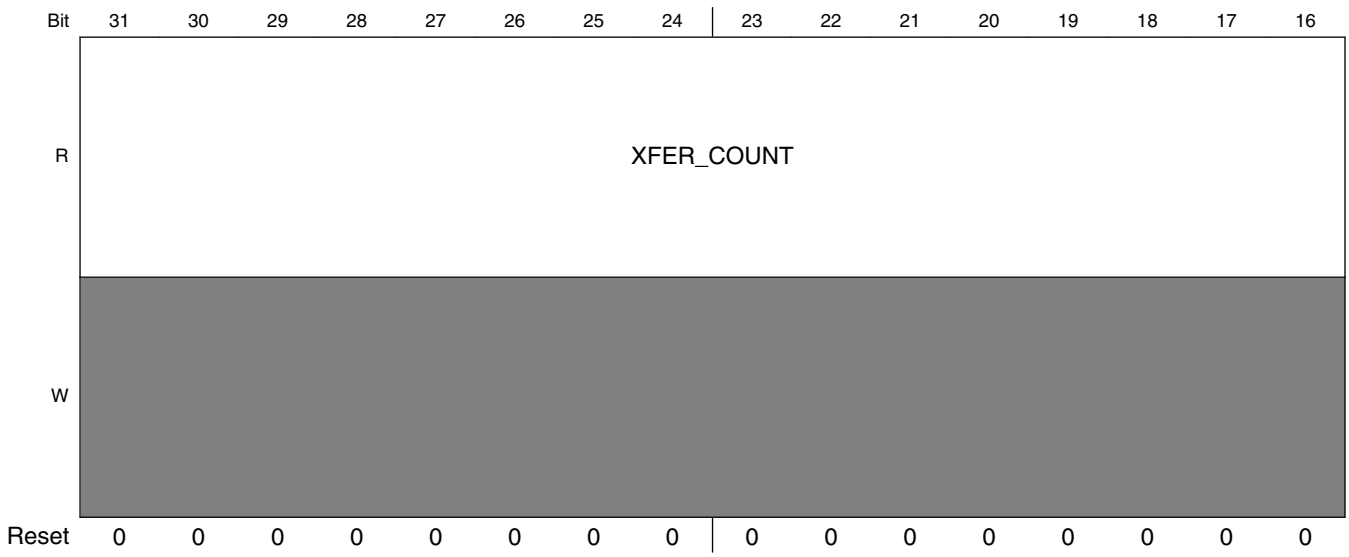
The APBH DMA Channel n command register specifies the DMA transaction to perform for the current command chain item.

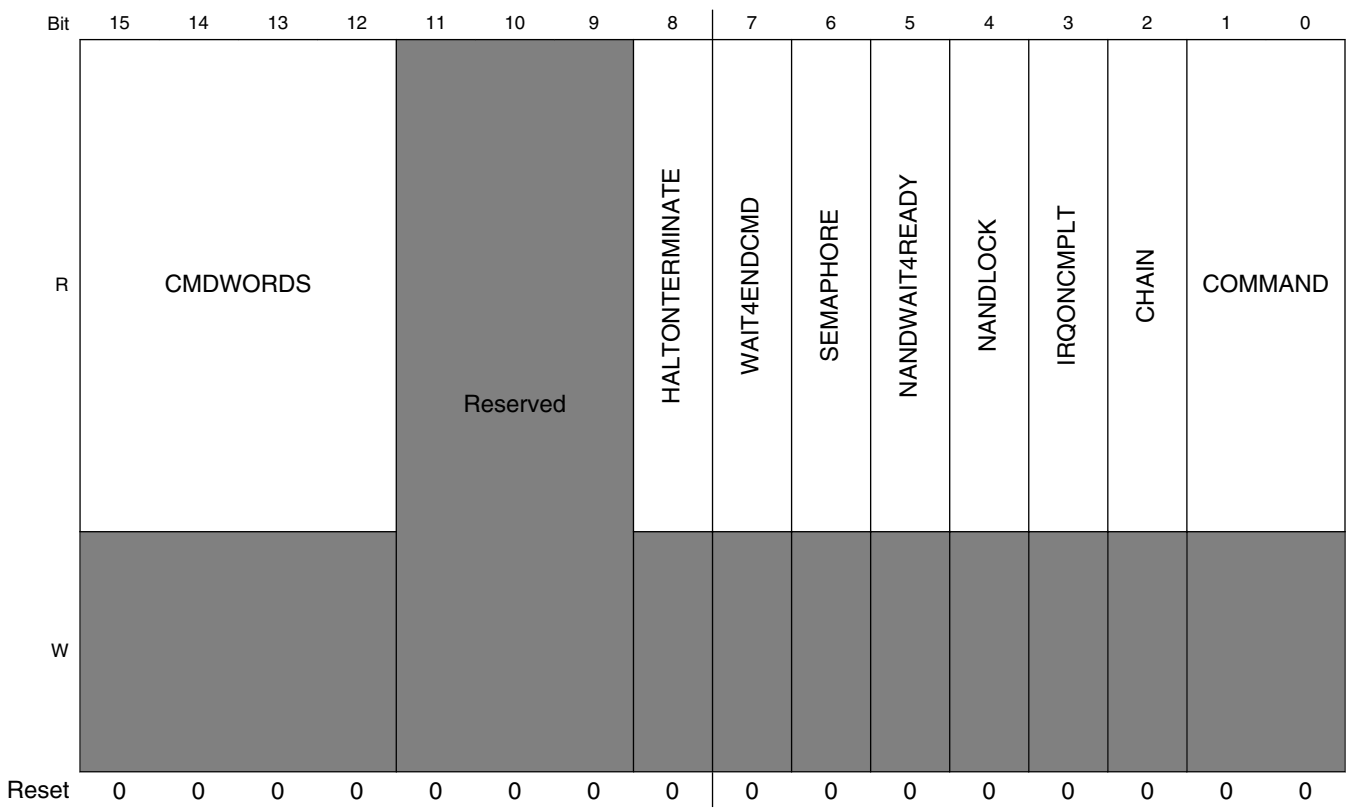
The command register controls the overall operation of each DMA command for this channel. It includes the number of bytes to transfer to or from the device, the number of APB PIO command words included with this command structure, whether to interrupt at command completion, whether to chain an additional command to the end of this one and whether this transfer is a read or write DMA transfer.

EXAMPLE

```
apbh_chn_cmd_t  dma_cmd;
dma_cmd.XFER_COUNT = 512;                                    // transfer 512 bytes
dma_cmd.COMMAND = BV_APBH_CHn_CMD_COMMAND__DMA_WRITE;  // transfer to system memory from
peripheral device
dma_cmd.CHAIN = 1;                                           // chain an additional command
structure on to the list
dma_cmd.IRQONCMPLT = 1;                                      // generate an interrupt on
completion of this command structure
```

Address: 3300_0000h base + 120h offset + (112d × i), where i=0d to 15d





APBH_CH*n*_CMD field descriptions

Field	Description
31–16 XFER_COUNT	This field indicates the number of bytes to transfer to or from the appropriate PIO register in the GPMI0 device. A value of 0 indicates a 64 KBytes transfer.
15–12 CMDWORDS	This field indicates the number of command words to send to the GPMI0, starting with the base PIO address of the GPMI0 control register and incrementing from there. Zero means transfer NO command words
11–9 -	This field is reserved. Reserved, always set to zero.
8 HALTONTERMINATE	A value of one indicates that the channel will immeditately terminate the current descriptor and halt the DMA channel if a terminate signal is set. A value of 0 will still cause an immediate terminate of the channel if the terminate signal is set, but the channel will continue as if the count had been exhausted, meaning it will honor IRQONCMPLT, CHAIN, SEMAPHORE, and WAIT4ENDCMD.
7 WAIT4ENDCMD	A value of one indicates that the channel will wait for the end of command signal to be sent from the APBH device to the DMA before starting the next DMA command.
6 SEMAPHORE	A value of one indicates that the channel will decrement its semaphore at the completion of the current command structure. If the semaphore decrements to zero, then this channel stalls until software increments it again.
5 NANDWAIT4READY	A value of one indicates that the NAND DMA channel will will wait until the NAND device reports "ready" before executing the command. It is ignored for non-NAND DMA channels.
4 NANDLOCK	A value of one indicates that the NAND DMA channel will remain "locked" in the arbiter at the expense of other NAND DMA channels. It is ignored for non-NAND DMA channels.
3 IRQONCMPLT	A value of one indicates that the channel will cause the interrupt status bit to be set upon completion of the current command, i.e. after the DMA transfer is complete.
2 CHAIN	A value of one indicates that another command is chained onto the end of the current command structure. At the completion of the current command, this channel will follow the pointer in APBH_CHn_CMDAR to find the next command.
COMMAND	This bitfield indicates the type of current command: 0x0 **NO_DMA_XFER** — Perform any requested PIO word transfers but terminate command before any DMA transfer. 0x1 **DMA_WRITE** — Perform any requested PIO word transfers and then perform a DMA transfer from the peripheral for the specified number of bytes. 0x2 **DMA_READ** — Perform any requested PIO word transfers and then perform a DMA transfer to the peripheral for the specified number of bytes. 0x3 **DMA_SENSE** — Perform any requested PIO word transfers and then perform a conditional branch to the next chained device. Follow the NEXCMD_ADDR pointer if the perpheral sense is true. Follow the BUFFER_ADDRESS as a chain pointer if the peripheral sense line is false.

9.5.5.11 APBH DMA Channel n Buffer Address Register (APBH_CH*n*_BAR)

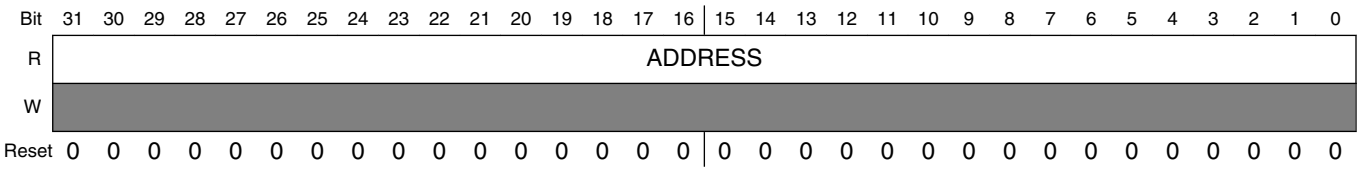
The APBH DMA Channel n buffer address register contains a pointer to the data buffer for the transfer. For immediate forms, the data is taken from this register. This is a byte address which means transfers can start on any byte boundary.

This register holds a pointer to the data buffer in system memory. After the command values have been read into the DMA controller and the device controlled by this channel, then the DMA transfer will begin, to or from the buffer pointed to by this register.

EXAMPLE

```
apbh_chn_bar_t  dma_data;
dma_data.ADDRESS = (reg32_t) pDataBuffer;
```

Address: 3300_0000h base + 130h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_BAR field descriptions

Field	Description
ADDRESS	Address of system memory buffer to be read or written over the AHB bus.

9.5.5.12 APBH DMA Channel n Semaphore Register (APBH_CH*n*_SEMA)

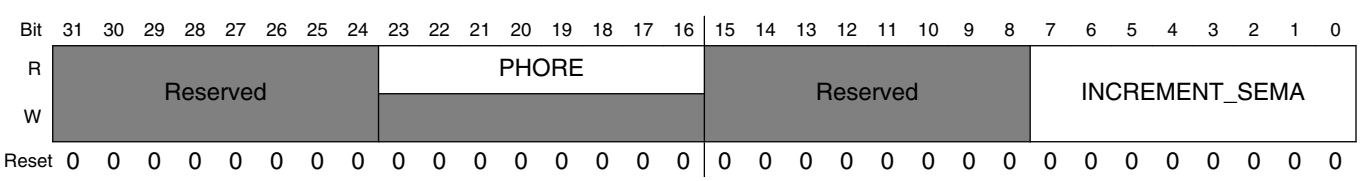
The APBH DMA Channel n semaphore register is used to synchronize the Arm platform instruction stream and the DMA chain processing state.

Each DMA channel has an 8 bit counting semaphore that is used to synchronize between the program stream and and the DMA chain processing. DMA processing continues until the DMA attempts to decrement a semaphore that has already reached a value of zero. When the attempt is made, the DMA channel is stalled until software increments the semaphore count.

EXAMPLE

```
BF_WR(APBH_CHn_SEMA, 0, INCREMENT_SEMA, 2);     // increment semaphore by two
current_sema = BF_RD(APBH_CHn_SEMA, 0, PHORE);  // get instantaneous value
```

Address: 3300_0000h base + 140h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_SEMA field descriptions

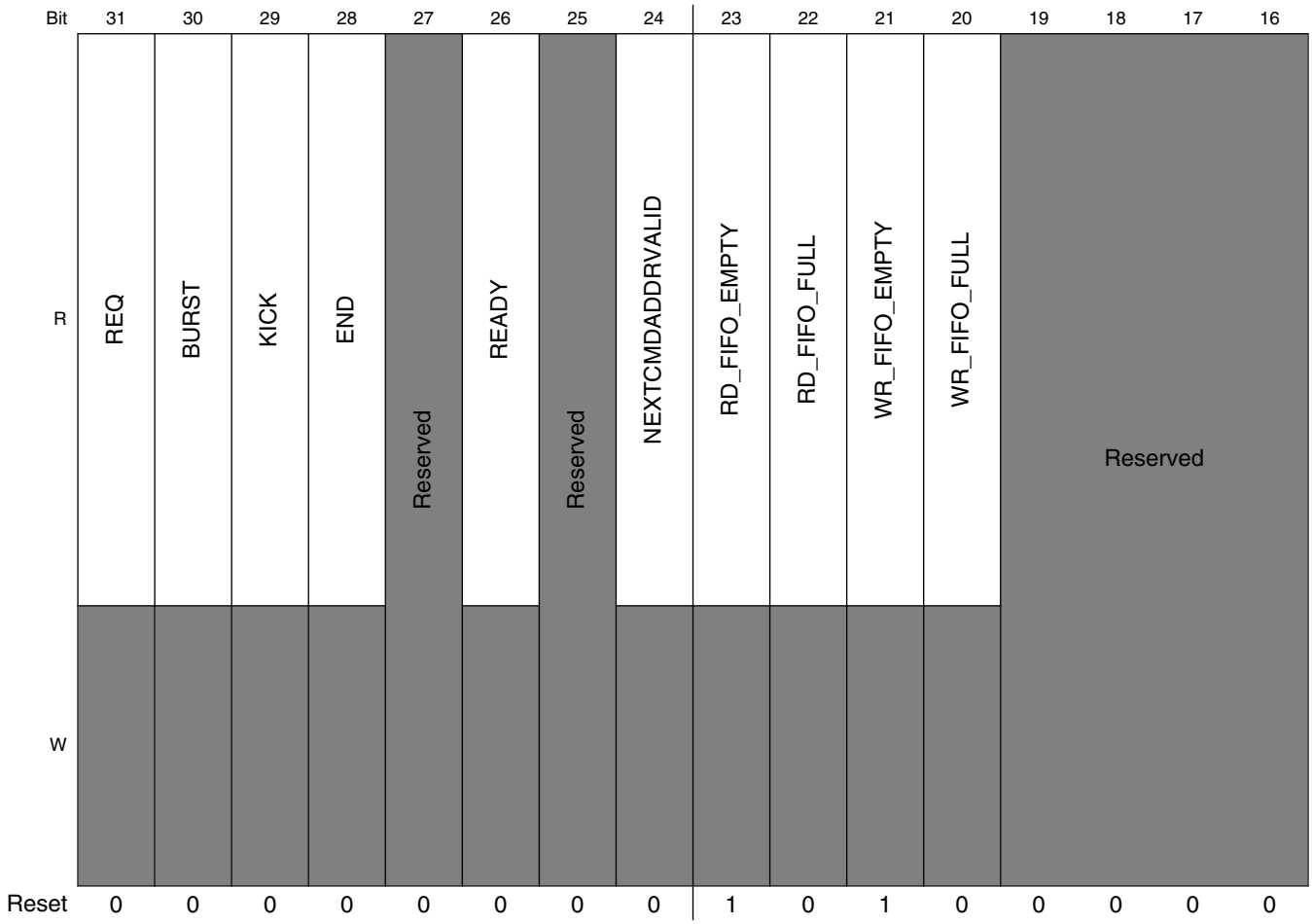
Field	Description
31–24 -	This field is reserved. Reserved, always set to zero.
23–16 PHORE	This read-only field shows the current (instantaneous) value of the semaphore counter.
15–8 -	This field is reserved. Reserved, always set to zero.
INCREMENT_ SEMA	The value written to this field is added to the semaphore count in an atomic way such that simultaneous software adds and DMA hardware substracts happening on the same clock are protected. This bit field reads back a value of 0x00. Writing a value of 0x02 increments the semaphore count by two, unless the DMA channel decrements the count on the same clock, then the count is incremented by a net one.

9.5.5.13 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG1)

This register gives debug visibility into the APBH DMA Channel n state machine and controls.

This register allows debug visibility of the APBH DMA Channel n.

Address: 3300_0000h base + 150h offset + (112d × i), where i=0d to 15d



APBH_CH*n*_DEBUG1 field descriptions

Field	Description
31 REQ	This bit reflects the current state of the DMA Request Signal from the APB device
30 BURST	This bit reflects the current state of the DMA Burst Signal from the APB device
29 KICK	This bit reflects the current state of the DMA Kick Signal sent to the APB Device
28 END	This bit reflects the current state of the DMA End Command Signal sent from the APB Device
27 SENSE	This field is reserved. This bit is reserved for this DMA Channel and always reads 0.
26 READY	This bit is reserved for this DMA Channel and always reads 0.
25 LOCK	This field is reserved. This bit is reserved for this Channel and always reads 0.
24 NEXTCMDADDRVALID	This bit reflects the internal bit which indicates whether the channel's next command address is valid.
23 RD_FIFO_EMPTY	This bit reflects the current state of the DMA Channel's Read FIFO Empty signal.
22 RD_FIFO_FULL	This bit reflects the current state of the DMA Channel's Read FIFO Full signal.
21 WR_FIFO_EMPTY	This bit reflects the current state of the DMA Channel's Write FIFO Empty signal.
20 WR_FIFO_FULL	This bit reflects the current state of the DMA Channel's Write FIFO Full signal.
19–5 RSVD1	This field is reserved. Reserved
STATEMACHINE	PIO Display of the DMA Channel n state machine state. 0x00 **IDLE** — This is the idle state of the DMA state machine. 0x01 **REQ_CMD1** — State in which the DMA is waiting to receive the first word of a command. 0x02 **REQ_CMD3** — State in which the DMA is waiting to receive the third word of a command. 0x03 **REQ_CMD2** — State in which the DMA is waiting to receive the second word of a command. 0x04 **XFER_DECODE** — The state machine processes the descriptor command field in this state and branches accordingly. 0x05 **REQ_WAIT** — The state machine waits in this state for the PIO APB cycles to complete. 0x06 **REQ_CMD4** — State in which the DMA is waiting to receive the fourth word of a command, or waiting to receive the PIO words when PIO count is greater than 1. 0x07 **PIO_REQ** — This state determines whether another PIO cycle needs to occur before starting DMA transfers. 0x08 **READ_FLUSH** — During a read transfers, the state machine enters this state waiting for the last bytes to be pushed out on the APB. 0x09 **READ_WAIT** — When an AHB read request occurs, the state machine waits in this state for the AHB transfer to complete. 0x0C **WRITE** — During DMA Write transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel. 0x0D **READ_REQ** — During DMA Read transfers, the state machine waits in this state until the AHB master arbiter accepts the request from this channel.

Table continues on the next page...

APBH_CH*n*_DEBUG1 field descriptions (continued)

Field	Description
	0x0E **CHECK_CHAIN** — Upon completion of the DMA transfers, this state checks the value of the Chain bit and branches accordingly. 0x0F **XFER_COMPLETE** — The state machine goes to this state after the DMA transfers are complete, and determines what step to take next. 0x14 **TERMINATE** — When a terminate signal is set, the state machine enters this state until the current AHB transfer is completed. 0x15 **WAIT_END** — When the Wait for Command End bit is set, the state machine enters this state until the DMA device indicates that the command is complete. 0x1C **WRITE_WAIT** — During DMA Write transfers, the state machine waits in this state until the AHB master completes the write to the AHB memory space. 0x1D **HALT_AFTER_TERM** — If HALTONTERMINATE is set and a terminate signal is set, the state machine enters this state and effectively halts. A channel reset is required to exit this state 0x1E **CHECK_WAIT** — If the Chain bit is a 0, the state machine enters this state and effectively halts. 0x1F **WAIT_READY** — When the NAND Wait for Ready bit is set, the state machine enters this state until the GPMI device indicates that the external device is ready.

9.5.5.14 AHB to APBH DMA Channel n Debug Information (APBH_CH*n*_DEBUG2)

This register gives debug visibility for the APB and AHB byte counts for DMA Channel n.

This register allows debug visibility of the APBH DMA Channel n.

Address: 3300_0000h base + 160h offset + (112d × i), where i=0d to 15d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	APB_BYTES																AHB_BYTES															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_CH*n*_DEBUG2 field descriptions

Field	Description
31–16 APB_BYTES	This value reflects the current number of APB bytes remaining to be transfered in the current transfer.
AHB_BYTES	This value reflects the current number of AHB bytes remaining to be transfered in the current transfer.

9.5.5.15 APBH Bridge Version Register (APBH_VERSION)

This register always returns a known read value for debug purposes it indicates the version of the block.

This register indicates the RTL version in use.

EXAMPLE

```
if (APBH_VERSION.B.MAJOR != 3)
   Error();
```

Address: 3300_0000h base + 800h offset = 3300_0800h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

APBH_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

9.6 62BIT Correcting ECC Accelerator (BCH)

9.6.1 Overview

The hardware ECC accelerator provides a forward error-correction function for improving the reliability of various storage media that may be attached to the device.

For example, NAND flash devices use a spare area to store ecc codes to correct some hard bit errors in data stored within the device, allowing higher device yields and, therefore, lower NAND device costs.

The Bose, Ray-Chaudhuri, Hocquenghem (BCH) Encoder and Decoder module is capable of correcting from 2 to 62 single bit errors within a block of data no larger than about 1900 bytes (512 bytes or 1024 bytes are typical) in applications such as protecting data and resources stored on modern NAND flash devices. The correction level in the BCH block is programmable to provide flexibility for varying applications and configurations of flash page size. The design can be programmed to encode protection of 2 to 62 bit errors when writing flash and to correct the corresponding number of errors on decode. The correction level when decoding MUST be programmed to the same correction level as was used during the encode phase.

BCH-codes are a type of block-code, which implies that all error-correction is performed over a block of N-symbols. The BCH operation will be performed over GF(2^{13} = 8192) or GF(2^{14} = 16384), which is the Galois Field consisting of 8191 or 16383 one-bit symbols. BCH-encoding (or encode for any block-code) can be performed by two algorithms: systematic encoding or multiplicative encoding. Systematic encoding is the process of reading all the symbols which constitute a block, dividing continuously these symbols by the generator polynomial for the GF(8192) or GF(16384) and appending the resulting ***t*** parity symbols to the block to create a BCH codeword (where ***t*** is the number of correctable bits).

The BCH encode process creates *t**13 (or *t**14)-bit parity symbols for each data block when the data is written to the flash device. The parity symbols are written to the flash device after the corresponding data block, and together these are collectively called the codeword. The codeword can be used during the decode process to correct errors that occur in either the data or parity blocks.

The BCH decoder processes code words in a 4-step fashion:

1. Syndrome Calculation (SC): This is the process of reading in all of the symbols of the codeword and continuously dividing by the generator polynomial for the field. 2**t* syndromes must be calculated for each codeword and inspection of the syndromes determines if there are errors: a non-zero set of syndromes indicates one or more errors. This process is implemented parallel hardware to minimize processing time since it must be done every time the decode is performed.
2. Key Equation Solver (KES): The syndromes represent 2t-linear equations with 2t-unknown variables. The process of solving these equations and selecting from the numerous solutions constitutes the KES module. When the KES block completes its operations, it generates an error locator polynomial (sigma) that is used in the proceeding block to determine the locations and values of the errors.
3. Chien Search (CS): This block takes input from the KES block and uses the Chien Algorithm for finding the locations of the errors based on the error locator polynomial. The method basically involves substituting all 8191 symbols from the GF(8192) or 16383 symbols from the GF(16383) into the locator polynomial. All evaluations that produce a zero solution indicate locations of the various errors. Since each located error corresponds to a single bit, the bit in the original data may be corrected by simply flipping the polarity of the incorrect location.
4. Correction: this block has to convert the symbol index and mask information to memory byte indexes and masks.

The BCH block, shown in the figure, was designed to operate in a pipelined fashion to maximize throughput. Aside from the initial latency to fill the pipeline stages, the BCH throughput is about 7/4 cycles/byte. Thus, the bottleneck in performing NAND reads and error corrections is the BCH rate. Current GPMI read rates are approximately 1/2 cycles/byte maximally for the current generation of NAND flash. Fortunately, BCH has a different master clock from GPMI, this gives some flexibility to match the throughput rate. The CPU is not directly involved in generating parity symbols, checking for errors, or correcting them.

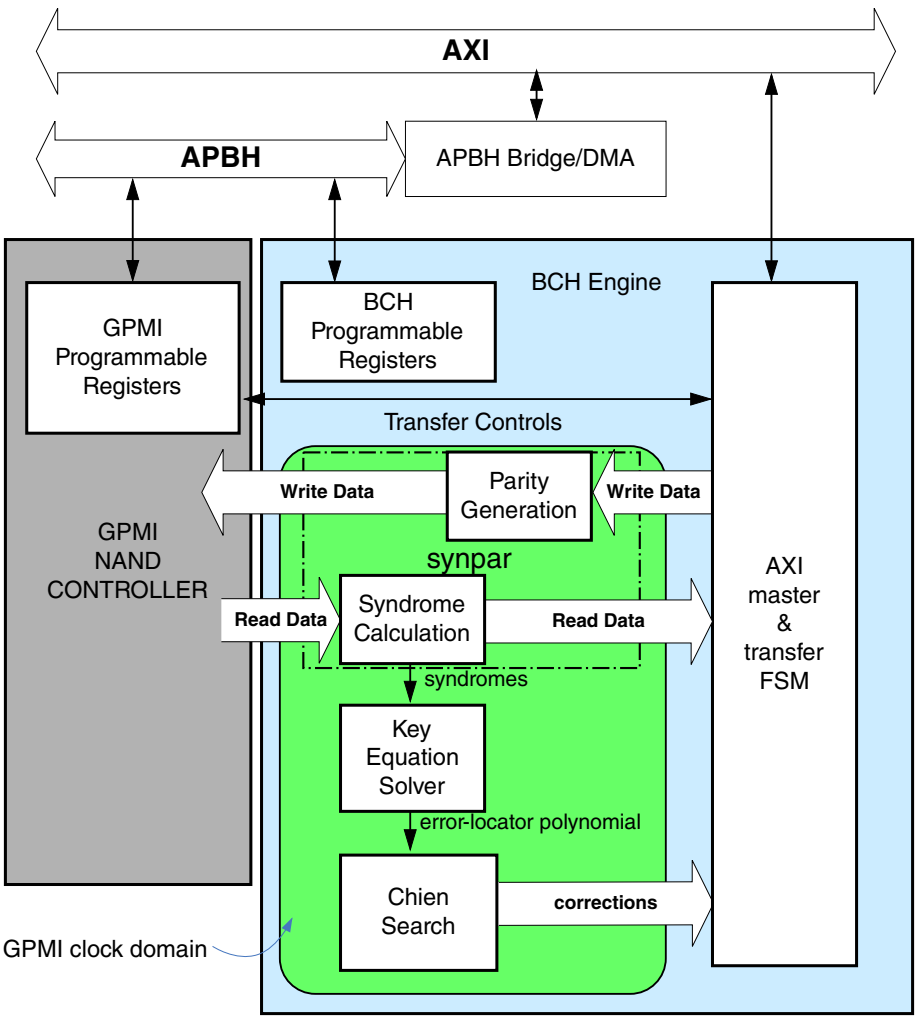


Figure 9-10. Hardware BCH Accelerator

9.6.2 Operation

Before performing any NAND flash read or write operations, software should first program the BCH's flash layout registers (see Flash Page Layout) to specify how data is to be formatted on the flash device. The BCH hardware allows full programmability over the flash page layout to enable users flexibility in balancing ECC correction levels and ever-changing flash page sizes.

To initiate a NAND Flash write, software will program a GPMI DMA operation. The DMA need only program the GPMI control registers (and handle the requisite flash addressing handshakes) since the BCH will handle all data operations using its AXI bus interface. The BCH will then send the data to the GPMI controller to be written to flash as it computes the parity symbols. At the end of each data block the BCH will insert the parity symbols into the data stream so that the GPMI sees only a continuous stream of data to be written.

NAND Flash read operations operate in a similar manner. As the GPMI controller reads the device, all data is sent to the BCH hardware for error detection/correction. The BCH controller writes all incoming read data to system memory and in parallel computes the syndromes used to detect bit errors. If errors are detected within a block, the BCH hardware activates the error correction logic to determine where bit errors have occurred and ultimately correct them in the data buffer in system memory. After an entire flash page has been read and corrected, the BCH will signal an interrupt to the CPU.

The figure below indicates how data read from the GPMI is operated on within the BCH hardware. As the BCH receives data from the GPMI (top row), it is written to memory by the BCH's Bus Interface Unit (BIU) (second row). For blocks requiring correction, the KES logic will be activated after the entire block has been received. Once the error locator polynomial has been computed, the corrections are determined by the Chien Search and fed back to the BIU, which performs a read, modify, write operation on the buffer in memory to correct the data.

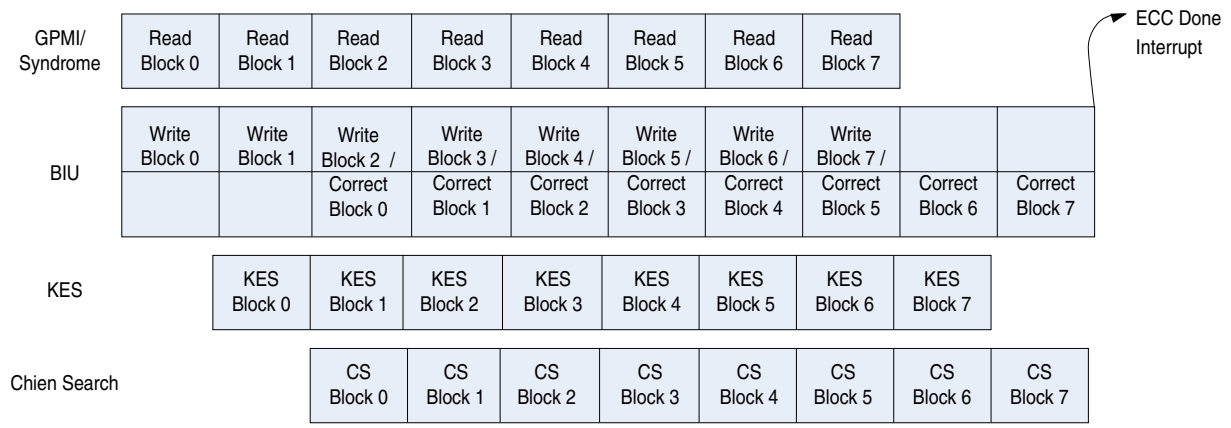


Figure 9-11. Block Pipeline while Reading Flash

9.6.2.1 BCH Limitations and Assumptions

- The BCH is programmable to support 2 to 62 bit error correction. ECC0 is supported as a pass-through, non-correcting mode.
- Data block sizes must be a multiple of 4 bytes and be aligned in system memory.
- The BCH supports a programmable number of metadata/auxiliary data bytes, from 0 to 255.
- Metadata will be written at the beginning of the flash page to facilitate fast access for filesystem operations.
- Metadata may be treated as an independent block for ECC purposes or combined with the first data block to conserve bits in the flash.
- The BCH does not support a partial page write (this can be accomplished by programming the BCH layout registers such that the BCH only sees a portion of the page).
- Flash read operations can read the entire page or the first block on the page.
- The BCH also supports a memory-to-memory mode of operation that does not require the use of DMA or the GPMI.

9.6.2.2 Flash Page Layout

The BCH supports a fully programmable flash page layout. The BCH maintains four independent layout registers that can describe four completely different NAND devices or layouts.

When the BCH initiates an operation, it selects one of the layouts by using the chip select as an index into the BCH_LAYOUTSELECT register that determines which layout should be used for the operation.

Three possible (generic) flash layout schemes are supported, as indicated in the figure below. (In each case, the metadata size may also be programmed to 0 bytes). Metadata may either be combined with the first block of data or the size of the first data block can be programmed to 0 to allow the metadata to be protected by its own ECC parity bits.

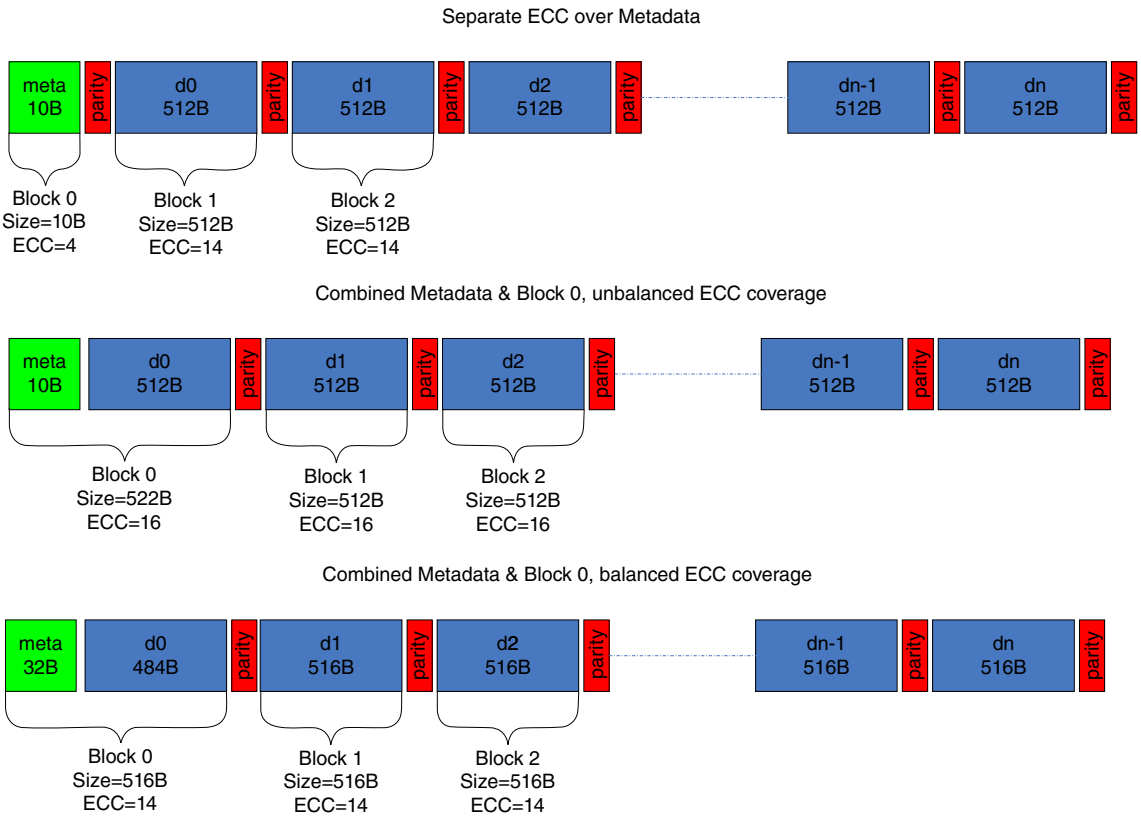


Figure 9-12. FLASH Page Layout Options

Each layout is determined by a pair of registers that define the following parameters:

- DATA0_SIZE: Indicates the number of data bytes in the first block on the page (this should not include parity or metadata bytes). This should be set to 0 when the metadata is to be covered separately with its own ECC. This must be a multiple of 4 bytes.
- ECC0: Indicates the ECC level to be used for the first block on the flash (data0+metadata).
- META_SIZE: Indicates the number of bytes (from 0-255) that are stored as metadata.

- NBLOCKS: Indicates the number of subsequent DATAN blocks on the flash, or the number of blocks following the DATA0 block.
- DATAN_SIZE: Indicates the number of data bytes in all subsequent data blocks. This MUST be a multiple of 4 bytes.
- ECCN: Indicates the ECC level to be used for the subsequent data blocks.
- GF0 or GFN: Indicates the Galois field the meta / data blocks are using
- PAGE_SIZE: Indicates the total number of bytes available per page on the physical flash device. This includes the spare area and is typically 4096+128, 4096+218, or 2048+64 bytes.

9.6.2.3 Determining the ECC layout for a device

Since the BCH is programmable, a system can trade off ECC levels for flash size and layout configurations.

The following examples indicate how to determine a valid layout based on the required storage space and flash size. For all cases, the size of the parity will be 13 (or 14 for GF(2^{14}))*ECC level ***bits***-- so for ECC8, 13 (or 14) bytes are required (per block).

9.6.2.3.1 4K+218 flash, 10 bytes metadata, 512 byte data blocks, separate metadata, Assuming GF(2^{13})

In this case, we have 8 data blocks each consisting of 512 bytes. Since the flash has 218 spare bytes (1744 bits), first estimate an ECC level for the data blocks by first subtracting the number of metadata bytes from the spare bytes (218 – 10 = 208 bytes = 1664 bits) then dividing the number of bits by 8 (number of blocks) and then by 13 (bits per ECC level).

(218 - 10) x 8 = 1664/13(8) = 16

Therefore all the data blocks could be covered by ECC16 if the metadata had no parity. This isn't acceptable, so assume ECC14 for all the data blocks. Now calculate the number of free bits for the metadata parity as

1664 - (14) x 13 x 8 = 208

Therefore, 208 bits remain for metadata parity. Dividing by 13 (bits/ECC) gives 16, so the metadata can be covered with ECC16. The settings for this device would then be

Table 9-9. Settings for 4K+218 FLASH

Setting	Value
PAGE_SIZE	4096+218=4314=0x10DA
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	16=0x10
GF0	GF(2^{13})
DATAN_SIZE	512=0x200 (in register interface, assigned as 0x80)
ECCN	14=0x0E
GFN	GF(2^{13})
NBLOCKS	8

9.6.2.3.2 4K+128 flash, 10 bytes metadata, 1024 byte data blocks, separate metadata, assuming GF(2^{13}) for data and GF(2^{14}) for metadata

This flash will have 118 bytes available for ECC (after subtracting the metadata size), therefore, 994 bits.

Dividing by 4*14 (number of blocks * ECC level) we get 17.75, therefore we can support ECC16 on the data blocks. The number of free spare bits becomes 944 - 16 * 4 * 14 = 944 - 896 = 48, divided by 13 = 3.69, therefore the metadata can be also covered by ECC2.

Table 9-10. Settings for 4K+128 FLASH

Setting	Value
PAGE_SIZE	4096+128=4224=0x1080
META_SIZE	10=0x0A
DATA0_SIZE	0
ECC0	2
GF0	GF(2^{13})
DATAN_SIZE	1024=0x400 (in register interface, assigned as 0x100)
ECCN	16
GFN	GF(2^{14})
NBLOCKS	4

In this case, there will be additional unused spare bits, with the BCH will pad out with zeros.

9.6.2.4 Data Buffers in System Memory

While the data on the flash is interleaved with parity symbols, the BCH assumes that the data buffers in memory are contiguous.

Metadata read from the flash will be stored to the location pointed to by the GPMI_AUXILIARY register and data will be written to the address specified in the GPMI_PAYLOAD register as is shown in the following figure where the block length is 512 bytes for example. Since the number of blocks on a flash page is programmable, the BCH also writes individual block correction status to the auxiliary pointer at the word-aligned address following the end of the metadata. Optionally, the computed syndromes may also be written to the auxiliary area if the DEBUGSYNDROME bit is set in the control register.

As blocks complete processing, the bus master will accumulate the status for each block and write it to the auxiliary data buffer following the metadata. The metadata area will be padded with 0's until the next word boundary and the status for blocks 0-3 will be written to the next word. The status for subsequent blocks will then be written to the buffer. The status for the first block (metadata block) is also stored in the STATUS_BLK0 register in the BCH_STATUS register. The completion codes for the blocks are indicated in the Table 9-11. Note that the definition of the bytes and their ordering in the auxiliary and payload storage areas are user defined. When this data is read back from the flash and put into memory, it will resemble the original buffer that was written out to the flash.

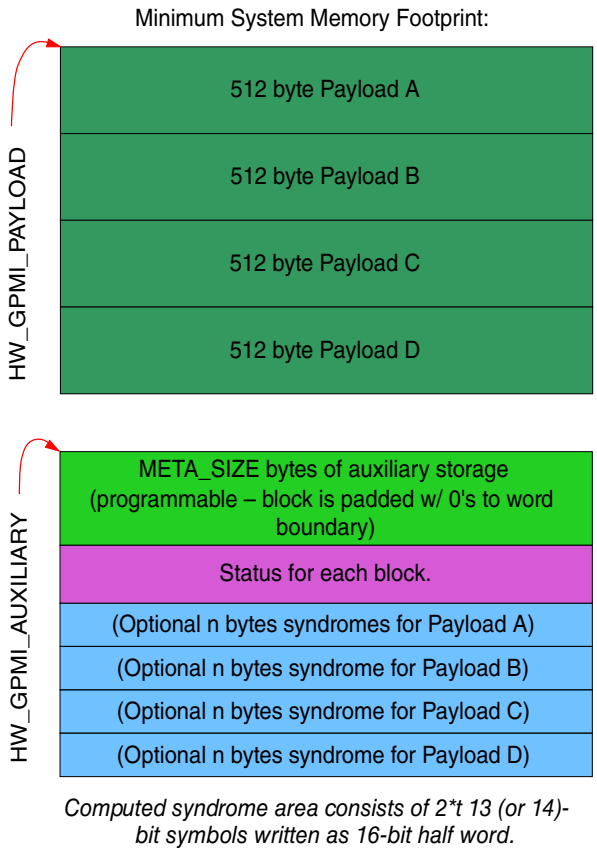
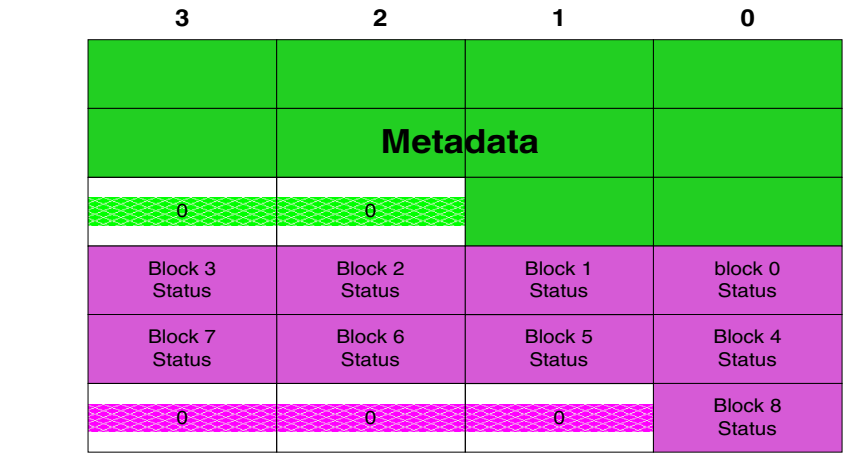


Figure 9-13. BCH Data Buffers in Memory

Table 9-11. Status Block Completion Codes

Code	Description
0xFF	Block is erased
0xFE	Block is uncorrectable
0x00	No errors found
0x01-0x3E	Number of errors corrected

The following figure shows the layout of the bytes within the status field.



Status bytes are allocated based on the NBLOCKS programmed into the flash format register. The number of status bytes are computed by NBLOCK+1. The status area will be padded with zeros to the next word boundary.

Syndrome data written for debug purposes follows the end of the status block.

Figure 9-14. Memory-to-Memory Operations

9.6.3 Memory to Memory (Loopback) Operation

The BCH supports a memory-to-memory mode of operation where both the encoded and decoded buffers reside in system memory.

This can be useful for applications where data must be protected by ECC, but the storage device does not reside on the GPMI bus.

The BCH operation in memory to memory mode is much simpler than in GPMI mode since DMAs are not required to manage the operation. Instead, software simply writes the BCH_DATAPTR and BCH_METAPTR with the addresses of the data and metadata (auxiliary) buffers and the BCH_ENCODEPTR with the address of the buffer for encoded data. To initiate the operation, software simply sets the M2M_ENCODE and M2M_ENABLE bits in the control register. The BCH can be programmed to either issue an interrupt at the end of the operation or software may poll the status bits for completion.

Memory to memory decode operations work in a similar manner. The encoded data address is written to the BCH_ENCODEPTR and the data and meta pointers are written to buffers that correspond to the desired decoded data addresses. To initiate a decode, software must set the M2M_ENCODE bit to 0 while writing the M2M_ENABLE bit. Note that the addresses written to the BCH_DATAPTR, BCH_METAPTR and BCH_ENCODEPTR registers should always be aligned on a 4 byte boundary. In other words, the 2 lower bits of the address should always be written with zeros.

9.6.4 Programming the BCH/GPMI Interfaces

Programming the BCH for NAND operations consists largely of disabling the soft reset and clock bits (SFTRST and CLKGATE) from the BCH_CTRL register and then programming the flash layout registers to correspond to the format of the attached NAND device(s).

The BCH_LAYOUTSELECT register should also be programmed to map the chip select of each attached device into one of the four layout registers.

The bulk of the programming is actually applied to the GPMI through PIO operations embedded in DMA command structures. The DMA will perform all the requisite handshaking with the GPMI interface to negotiate the address portion of the transfer, then the BCH will handle all the movement of data from memory to the GPMI (writes) or the GPMI to memory (reads). The BCH will direct all data blocks to the buffer pointed to by the PAYLOAD_BUFFER and the metadata will be written to the AUXILIARY_BUFFER. Both of these registers are located in the GPMI PIO data space and are communicated to the BCH hardware at the beginning of the transfer. Thus, the normal multi-NAND DMA based device interleaving is preserved, that is, four NANDs on four separate chip selects can be scheduled for read or write operations using the BCH. Whichever channel finishes its ready wait first and enters the DMA arbiter with its lock bit set owns the GPMI command interface and through it owns the BCH resources for the duration of its processing.

9.6.4.1 BCH Encoding for NAND Writes

The BCH encoder flowchart in Figure 9-15 shows the detailed steps involved in programming and using the BCH encoder. This flowchart shows how to use the BCH block with the GPMI.

To use the BCH encoder with the GPMI's DMA, create a DMA command chain containing ten descriptor structures, as shown in Figure 9-17 and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The ten descriptors perform the following tasks:

1. Disable the BCH block (in case it was enabled) and issue NAND write setup command byte (under CLE) and address bytes (under ALE).
2. Configure and enable the BCH and GPMI blocks to perform the NAND write.
3. Disable the BCH block and issue NAND write execute command byte (under CLE).

4. Wait for the NAND device to finish writing the data by watching the ready signal.
5. Check for NAND timeout through PSENSE.
6. Issue NAND status command byte (under CLE).
7. Read the status and compare against expected.
8. If status is incorrect or incomplete, branch to error handling descriptor chain.
9. Otherwise, write is complete and emit GPMI interrupt.

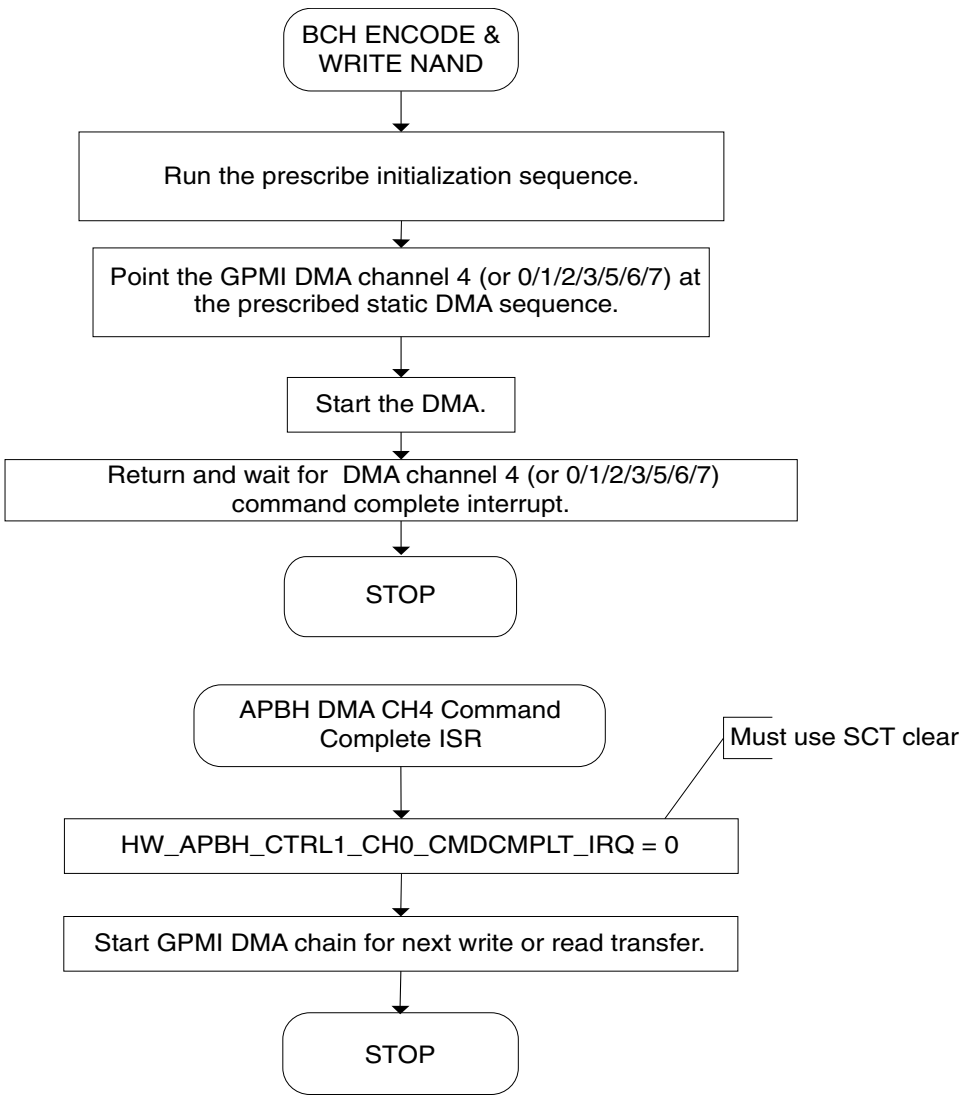


Figure 9-15. BCH Encode Flowchart

Descriptor Legend

NEXT CMD ADDR										
CMD	<=	xfer_count	cmdwords	wait4endcmd	semaphore	nandwait4ready	nandlock	irqoncmplt	chain	command
BUFFER ADDR										
HW_GPMI_CTRL0	<=	command_mode	word_length	lock_cs	CS	address	address_increment	xfer_count		
HW_GPMI_COMPARE	<=	mask				reference				
HW_GPMI_ECCCTRL	<=	ecc_cmd			enable_ecc				buffer_mask	
HW_GPMI_ECCCOUNT										
HW_GPMI_PAYLOAD										
HW_GPMI_AUXILIARY										

Figure 9-16. BCH DMA Descriptor Legend

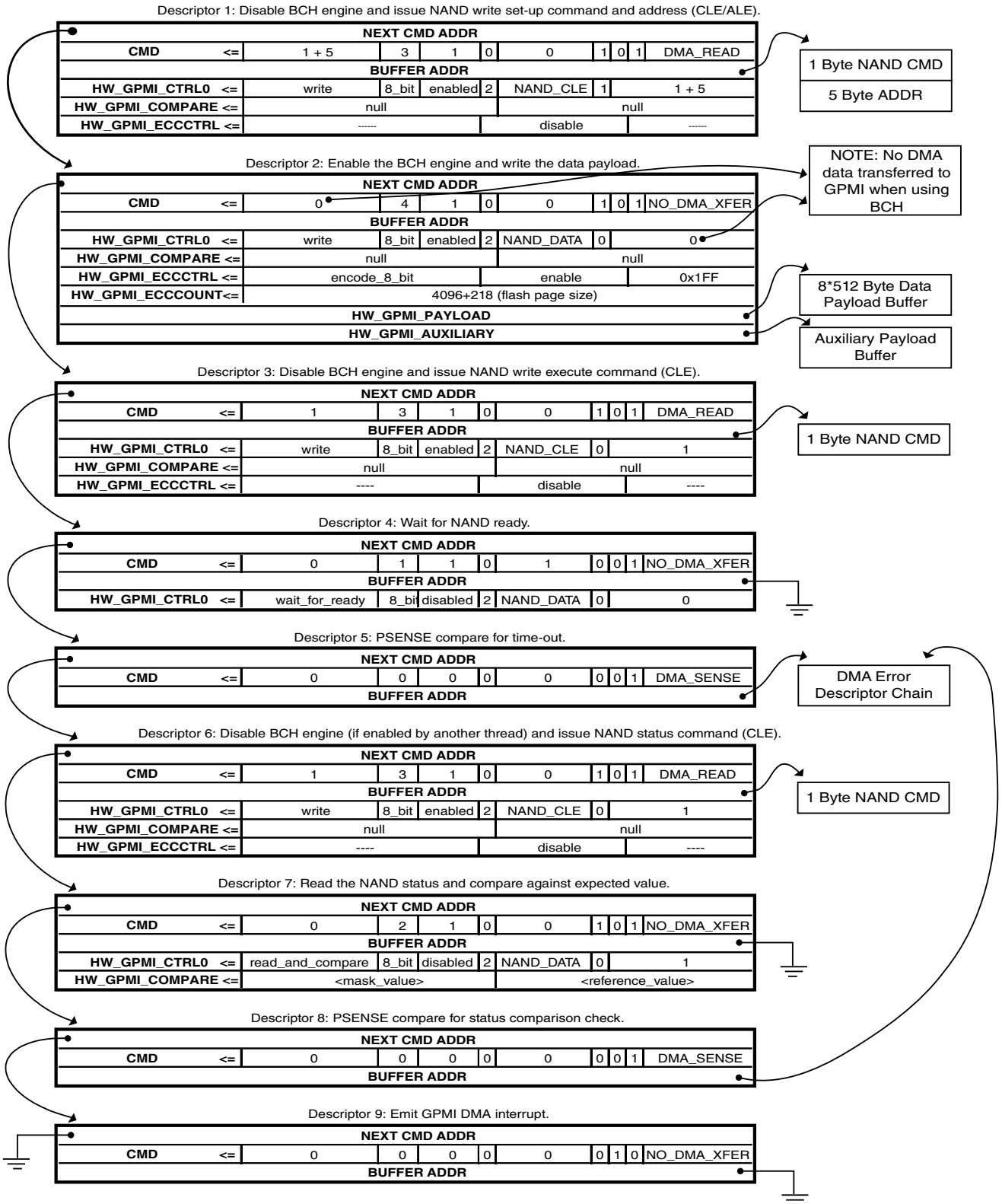


Figure 9-17. BCH Encode DMA Descriptor Chain

9.6.4.1.1 DMA Structure Code Example

The following code sample illustrates the coding for one write transaction involving 4096 bytes of data payload (eight 512-byte blocks) and 10 bytes of auxiliary payload (also referred to as metadata) to a 4K NAND page sitting on GPMI CS2.

```
//-----------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-----------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-----------------------------------------------------------------------------
// allocate 10 descriptors for doing a NAND ECC Write
//-----------------------------------------------------------------------------
GENERIC_DESCRIPTOR write[10];
//-----------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-----------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-----------------------------------------------------------------------------
// 8 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is write setup command
//   bytes 1-5 is the NAND address
//   byte 6 is write execute command
//   byte 7 is status command
//-----------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[8];
//-----------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int write_payload_buffer[(4096/4)];
//-----------------------------------------------------------------------------
// 65 byte meta-data to be written to NAND
// needs to be word aligned
//-----------------------------------------------------------------------------
unsigned int write_aux_buffer[65];
//-----------------------------------------------------------------------------
// Descriptor 1: issue NAND write setup command (CLE/ALE)
//-----------------------------------------------------------------------------
write[0].dma_nxtcmdar = &write[1];                          // point to the next descriptor
write[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5)|   // 1 byte command, 5 byte address
                   BF_APBH_CHn_CMD_CMDWORDS      (3)    |   // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |  // wait for command to finish
before
                                                           //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                   BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels
from
                                                           //    taking over
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                   BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
```

```
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ); // read data from DMA, write to
NAND
write[0].dma_bar = &nand_cmd_addr_buffer;           // byte 0 write setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
write[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS              (2)      | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)       | // send command and
address
                    BF_GPMI_CTRL0_XFER_COUNT       (1 + 5);    // 1 byte command, 5 byte
address
write[0].gpmi_compare = NULL;                            // field not used but necessary to
set eccctrl
write[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//--------------------------------------------------------------------------
// Descriptor 2: write the data payload (DATA)
//--------------------------------------------------------------------------
write[1].dma_nxtcmdar = &write[2];                       // point to the next descriptor
write[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT      (0)|  // NOTE: No DMA data transfer
                   BF_APBH_CHn_CMD_CMDWORDS        (4)|  // send 4 words to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD     (1)|  // Wait to end
                   BF_APBH_CHn_CMD_SEMAPHORE       (0)|
                   BF_APBH_CHn_CMD_NANDWAIT4READY  (0)|
                   BF_APBH_CHn_CMD_NANDLOCK        (1)|  // maintain resource lock
                   BF_APBH_CHn_CMD_IRQONCMPLT      (0)|
                   BF_APBH_CHn_CMD_CHAIN           (1)|  // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_NO_XFER);  // No data transferred
write[1].dma_bar = &write_payload_buffer;                  // pointer for the 4K byte
data area
// 4 words sent to the GPMI
write[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS                (2)       | // must correspond to NAND
CS used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)|
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);        // NOTE: this field
contains
                                                                 //  the total amount
                              // DMA transferred to GPMI via DMA (0)!
write[1].gpmi_compare = NULL;                              // field not used but necessary
to
set eccctrl
write[1].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    ENCODE_8_BIT) | // specify t = 8
mode
                        BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       | // enable ECC module
                     BF_GPMI_ECCCTRL_BUFFER_MASK   (0x1FF);          // write all 8 data
blocks
                                                                   //   and 1 aux block
write[1].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);      // specify number of bytes
                                                                //   written to NAND
write[1].gpmi_data_pointer = &write_payload_pointer;            // data buffer address
write[1].gpmi_aux_pointer  = &write_aux_pointer;                // metadata pointer
//--------------------------------------------------------------------------
// Descriptor 3: issue NAND write execute command (CLE)
//--------------------------------------------------------------------------
write[2].dma_nxtcmdar = &write[3];                          // point to the next descriptor
write[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1) |     // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3) |     // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |     // wait for command to finish
before
                                                          //    continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0) |
                   BF_APBH_CHn_CMD_NANDLOCK      (1) |     // maintain resource lock
```

```
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                  BF_APBH_CHn_CMD_CHAIN         (1) |     // follow chain to next command
                BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);   // read data from DMA, write to
NAND
write[2].dma_bar = &nand_cmd_addr_buffer[6];             // point to byte 6, write execute
command
// 3 words sent to the GPMI
write[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)      | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);       // 1 byte command
write[2].gpmi_compare = NULL;                           // field not used but necessary to set
eccctrl
write[2].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//--------------------------------------------------------------------------
// Descriptor 4: wait for ready (CLE)
//--------------------------------------------------------------------------
write[3].dma_nxtcmdar = &write[4];                       // point to the next descriptor

write[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0) |   // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (1) |   // send 1 word to the GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (1) |   // wait for command to finish before
                                                         //      continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0) |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(1) |   // wait for nand to be ready
                   BF_APBH_CHn_CMD_NANDLOCK      (0) |   // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0) |
                   BF_APBH_CHn_CMD_CHAIN         (1) |   // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);          // no dma transfer
write[3].dma_bar = NULL;                                                // field not used
// 1 word sent to the GPMI
write[3].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)         |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)      |
                  BF_GPMI_CTRL0_CS              (2)         | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)     |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)            |
                      BF_GPMI_CTRL0_XFER_COUNT         (0);
//--------------------------------------------------------------------------
// Descriptor 5: psense compare (time out check)
//--------------------------------------------------------------------------
write[4].dma_nxtcmdar = &write[5];                         // point to the next descriptor
write[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)       | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)       | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)       |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)       |
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                   BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[4].dma_bar = dma_error_handler;              // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 6: issue NAND status command (CLE)
//--------------------------------------------------------------------------
write[5].dma_nxtcmdar = &write[6];                         // point to the next descriptor
write[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)     |  // 1 byte command
                   BF_APBH_CHn_CMD_CMDWORDS      (3)     |  // send 3 words to the GPMI
                 BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     |  // wait for command to finish
before
continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
```

```
                    BF_APBH_CHn_CMD_NANDLOCK      (1)    |  // prevent other DMA channels from
taking over
                     BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                     BF_APBH_CHn_CMD_CHAIN         (1)    |  // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
NAND
write[5].dma_bar = &nand_cmd_addr_buffer[7];                 // point to byte 7, status
command
write[5].gpmi_compare = NULL;                              // field not used but necessary to set
eccctrl
write[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
// 3 words sent to the GPMI
write[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)    | // write to the NAND
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)    |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)  |
                    BF_GPMI_CTRL0_CS               (2)     | // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)       |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//--------------------------------------------------------------------------
// Descriptor 7: read status and compare (DATA)
//--------------------------------------------------------------------------
write[6].dma_nxtcmdar = &write[7];                           // point to the next descriptor
write[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)  |    // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (2)  |    // send 2 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)  |    // wait for command to finish
before
                                                           //      continuing
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)  |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)  |
                   BF_APBH_CHn_CMD_NANDLOCK      (1)  |    // maintain resource lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)  |
                   BF_APBH_CHn_CMD_CHAIN         (1)  |    // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
write[6].dma_bar = NULL;                                     // field not used
// 2 word sent to the GPMI
write[6].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ_AND_COMPARE) | // read from the
                                                                  //       NAND and
//                                                               // compare to expect
                      BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                      BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                    BF_GPMI_CTRL0_CS             (2)       |   // must correspond to NAND CS
used
                      BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA) |
                      BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                      BF_GPMI_CTRL0_XFER_COUNT         (1);
write[6].gpmi_compare = <MASK_AND_REFERENCE_VALUE>;  // NOTE: mask and reference values are
NAND
                                                          // SPECIFIC to evaluate the NAND
status
//--------------------------------------------------------------------------
// Descriptor 8: psense compare (time out check)
//--------------------------------------------------------------------------
write[7].dma_nxtcmdar = &write[8];                             // point to the next descriptor
write[7].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)         | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS      (0)         | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)         | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE     (0)         |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)         |
                   BF_APBH_CHn_CMD_NANDLOCK      (0)         | // relinquish nand lock
                   BF_APBH_CHn_CMD_IRQONCMPLT    (0)         |
                   BF_APBH_CHn_CMD_CHAIN         (1)         | // follow chain to next command
                   BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);  // perform a sense check
write[7].dma_bar = dma_error_handler;              // if sense check fails, branch to error
handler
//--------------------------------------------------------------------------
// Descriptor 9: emit GPMI interrupt
//--------------------------------------------------------------------------
write[8].dma_nxtcmdar = NULL;                                // not used since this is
```

```
last
descriptor
write[8].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT     (0)          | // no dma transfer
                   BF_APBH_CHn_CMD_CMDWORDS       (0)          | // no words sent to GPMI
                   BF_APBH_CHn_CMD_WAIT4ENDCMD    (0)          | // do not wait to continue
                   BF_APBH_CHn_CMD_SEMAPHORE      (0)          |
                   BF_APBH_CHn_CMD_NANDWAIT4READY(0)           |
                   BF_APBH_CHn_CMD_NANDLOCK       (0)          |
                   BF_APBH_CHn_CMD_IRQONCMPLT     (1)          | // emit GPMI interrupt
                   BF_APBH_CHn_CMD_CHAIN          (0)          | // terminate DMA chain
processing
                   BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);  // no dma transfer
```

9.6.4.1.2 Using the BCH Encoder

To use the BCH encoder, first turn off the module-wide soft reset bit in both the GPMI and BCH blocks before starting any DMA activity.

Turning off the soft reset must take place by itself, prior to programming the rest of the control registers. Turn off the BCH bus master soft reset bit. Turn off the clock gate bits.

Program the remainder of the GPMI, BCH and APBH DMA as follows:

```
// bring APBH out of reset
APBH_CTRL0_CLR(BM_APBH_CTRL0_SFRST);
APBH_CTRL0_CLR(BM_APBH_CTRL0_CLKGATE);

// bring BCH out of reset
BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);

// bring gpmi out of reset
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_SFTRST);
GPMI_CTRL0_CLR(BM_GPMI_CTRL0_CLKGATE);
GPMI_CTRL1_SET(BM_GPMI_CTRL1_DEV_RESET | // deassert reset
                   BM_GPMI_CTRL1_BCH_MODE  );    // enable BCH mode

// enable pinctrl
PINCTRL_CTRL_WR(0x00000000);

// enable gpmi pins
PINCTRL_MUXSEL0_CLR(0x0000ffff);    // data bits
PINCTRL_MUXSEL1_CLR(0x03ffffff);    // control bits
PINCTRL_MUXSEL8_CLR(0x0003f3ff);    // control bits
PINCTRL_MUXSEL8_SET(0x00015155);    // control bits
```

Note that for writing NANDs (ECC encoding), only GPMI DMA command complete interrupts are used. The BCH engine is used for writing to the NAND but may optionally produces an interrupt. From the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data write by using the GPMI to issue a write setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is disabled and not used for these commands.
- DMA descriptor 2 enables the BCH engine for encoding to begin the initial writing of the NAND data by specifying where the data and auxiliary payload are coming from in system memory.

- DMA descriptor 3 issues the write commit command byte under CLE to the NAND.
- DMA descriptor 4 waits for the NAND to complete the write commit/transfer by watching the NAND's ready line status. This descriptor relinquishes the NANDLOCK on the GPMI to enable the other DMA channels to initiate NAND transactions on different NAND CS lines.
- DMA descriptor 6 issues a NAND status command byte under "CLE" to check the status of the NAND device following the page write.
- DMA descriptor 7 reads back the NAND status and compares the status with an expected value. If there are differences, then the DMA processing engine follows an error-handling DMA descriptor path.
- DMA descriptor 8 disables the BCH engine and emits a GPMI interrupt to indicate that the NAND write has been completed.

9.6.4.2 BCH Decoding for NAND Reads

When a page is read from NAND flash, BCH syndromes will be computed and, if correctable errors are found, they will be corrected on a per block basis within the NAND page. This decoding process is fully overlapped with other NAND data reads and with CPU execution. The BCH decoder flowchart in the figure below shows the steps involved in programming the decoder. The hardware flow of reading and decoding a 4096-byte page is shown in Figure 9-19.

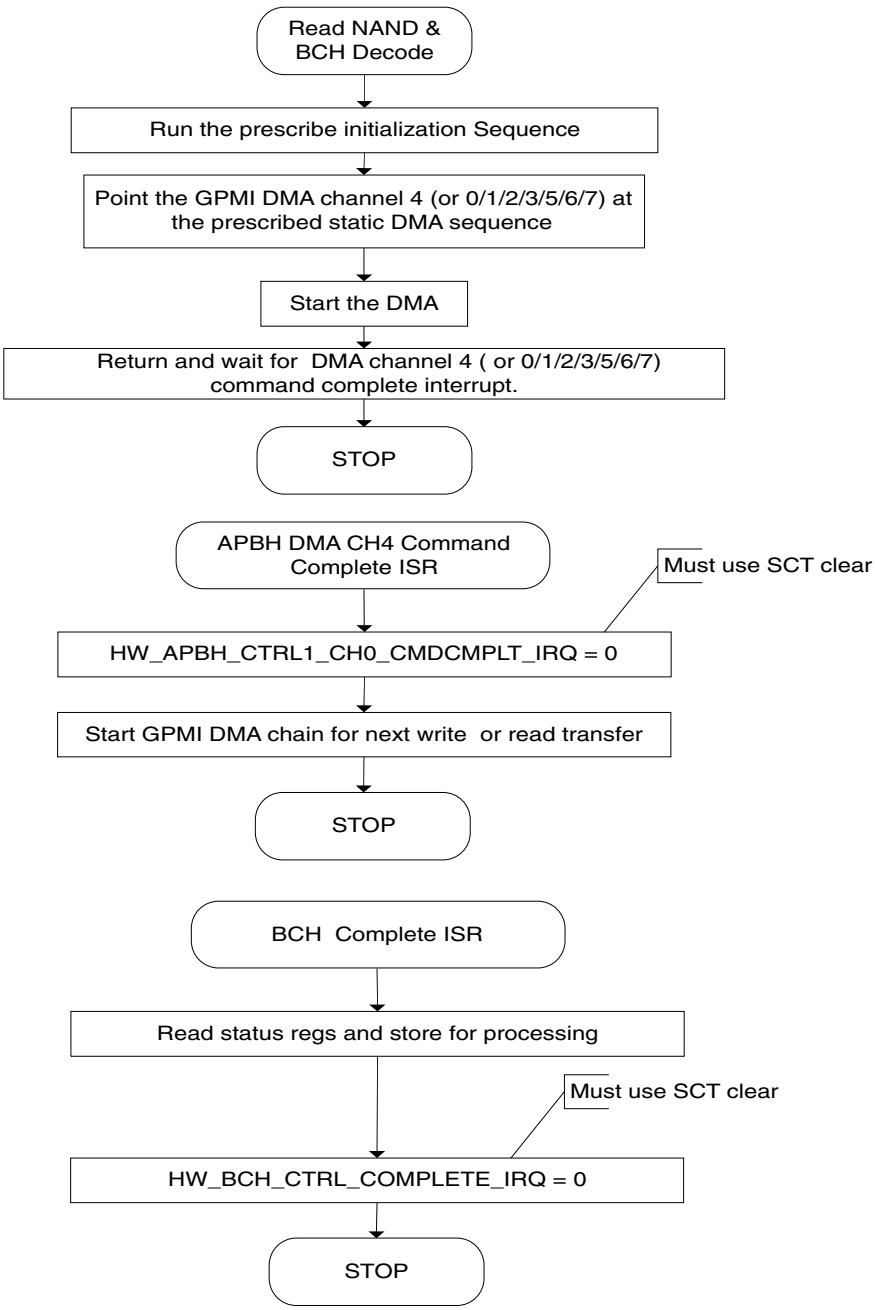


Figure 9-18. BCH Decode Flowchart

Conceptually, an APHB DMA Channel (0,1,2 or 3) command chain with seven command structures linked together is used to perform the BCH decode operation (as shown in Figure 9-19).

Note

The GPMI's DMA command structures controls the BCH decode operation.

To use the BCH decoder with the GPMI's DMA, create a DMA command chain containing seven descriptor structures, as shown in the figure below and detailed in the DMA structure code example that follows it in DMA Structure Code Example. The seven DMA descriptors perform the following tasks:

1. Issue NAND read setup command byte (under "CLE") and address bytes (under "ALE").
2. Issue NAND read execute command byte (under "CLE").
3. Wait for the NAND device to complete accessing the block data by watching the ready signal.
4. Check for NAND timeout through "PSENSE".
5. Configure and enable the BCH block and read the NAND block data.
6. Disable the BCH block.
7. Descriptor NOP to allow NANDLOCK in the previous descriptor to the thread-safe.

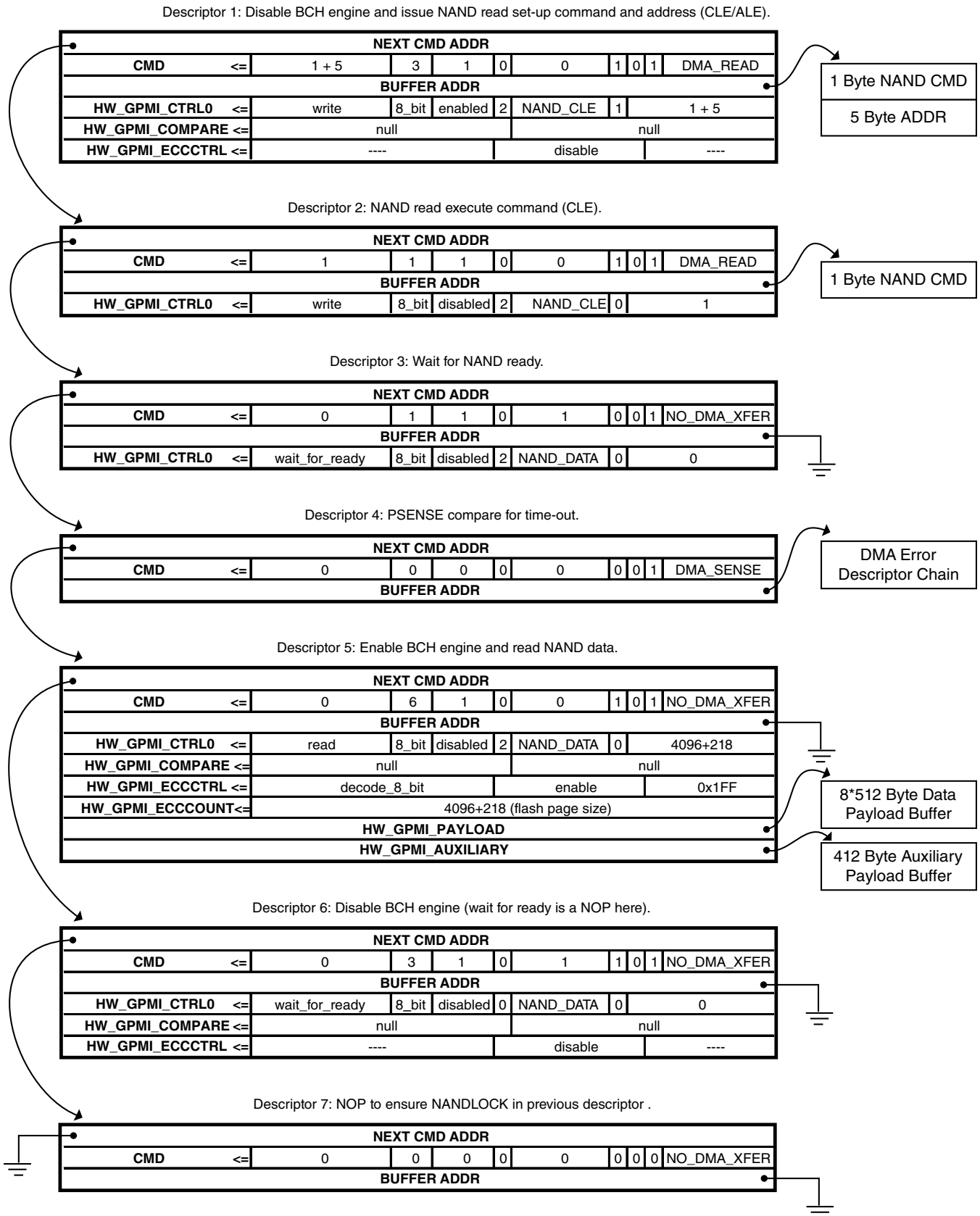


Figure 9-19. BCH Decode DMA Descriptor Chain

9.6.4.2.1 DMA Structure Code Example

The following sample code illustrates the coding for one read transaction, consisting of a seven DMA command structure chain for reading all 4096 bytes of payload data (eight 512-byte blocks) and 65 bytes of metadata with the associative parity bytes (8 * (18) + 9) from a 4K NAND page sitting on GPMI CS2.

```
//-------------------------------------------------------------------------
// generic DMA/GPMI/ECC descriptor struct, order sensitive!
//-------------------------------------------------------------------------
typedef struct {
  // DMA related fields
  unsigned int dma_nxtcmdar;
  unsigned int dma_cmd;
  unsigned int dma_bar;
  // GPMI related fields
  unsigned int gpmi_ctrl0;
  unsigned int gpmi_compare;
  unsigned int gpmi_eccctrl;
  unsigned int gpmi_ecccount;
  unsigned int gpmi_data_ptr;
  unsigned int gpmi_aux_ptr;
} GENERIC_DESCRIPTOR;
//-------------------------------------------------------------------------
// allocate 7 descriptors for doing a NAND ECC Read
//-------------------------------------------------------------------------
GENERIC_DESCRIPTOR read[7];
//-------------------------------------------------------------------------
// DMA descriptor pointer to handle error conditions from psense checks
//-------------------------------------------------------------------------
unsigned int * dma_error_handler;
//-------------------------------------------------------------------------
// 7 byte NAND command and address buffer
// any alignment is ok, it is read by the GPMI DMA
//   byte 0 is read setup command
//   bytes 1-5 is the NAND address
//   byte 6 is read execute command
//-------------------------------------------------------------------------
unsigned char nand_cmd_addr_buffer[7];
//-------------------------------------------------------------------------
// 4096 byte payload buffer used for reads or writes
// needs to be word aligned
//-------------------------------------------------------------------------
unsigned int read_payload_buffer[(4096/4)];
//-------------------------------------------------------------------------
// 412 byte auxiliary buffer used for reads
// needs to be word aligned
//-------------------------------------------------------------------------
unsigned int read_aux_buffer[(412/4)];
//-------------------------------------------------------------------------
// Descriptor 1: issue NAND read setup command (CLE/ALE)
//-------------------------------------------------------------------------
read[0].dma_nxtcmdar = &read[1];                              // point to the next descriptor
read[0].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1 + 5) |  // 1 byte command, 5 byte address
                  BF_APBH_CHn_CMD_CMDWORDS      (3)     |  // send 3 words to the GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     |  // wait for command to finish
                                                           //     before continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)     |  // prevent other DMA channels from
                                                           //    taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)     |
                  BF_APBH_CHn_CMD_CHAIN         (1)     |   // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);  // read data from DMA, write to
```

```
NAND
read[0].dma_bar = &nand_cmd_addr_buffer;            // byte 0 read setup, bytes 1 - 5 NAND
address
// 3 words sent to the GPMI
read[0].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      ENABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (1)        | // send command and address
                     BF_GPMI_CTRL0_XFER_COUNT         (1 + 5);     // 1 byte command, 5 byte
address
read[0].gpmi_compare = NULL;                                     // field not used but necessary to set
eccctrl
read[0].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE); // disable the ECC block
//-------------------------------------------------------------------------
// Descriptor 2: issue NAND read execute command (CLE)
//-------------------------------------------------------------------------
read[1].dma_nxtcmdar = &read[2];                                 // point to the next descriptor
read[1].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (1)    |   // 1 byte read command
                  BF_APBH_CHn_CMD_CMDWORDS      (1)    |   // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)    |   // wait for command to finish
before
                                                         //    continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)    |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)    |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)    |   // prevent other DMA channels from
                                                         //     taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)    |
                  BF_APBH_CHn_CMD_CHAIN         (1)    |   // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_READ);     // read data from DMA, write
to NAND
read[1].dma_bar = &nand_cmd_addr_buffer[6];              // point to byte 6, read execute
command
// 1 word sent to the GPMI
read[1].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WRITE)     | // write to the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)     |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)  |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to NAND
CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_CLE)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                     BF_GPMI_CTRL0_XFER_COUNT         (1);        // 1 byte command
//-------------------------------------------------------------------------
// Descriptor 3: wait for ready (DATA)
//-------------------------------------------------------------------------
read[2].dma_nxtcmdar = &read[3];                                 // point to the next descriptor
read[2].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)      | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (1)      | // send 1 word to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)      | // wait for command to finish
before
                                                         //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)      |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)      | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (0)      | // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)      |
                  BF_APBH_CHn_CMD_CHAIN         (1)      | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);   // no dma transfer
read[2].dma_bar = NULL;                                          // field not used
// 1 word sent to the GPMI
read[2].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, WAIT_FOR_READY) | // wait for NAND
ready
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)          |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)       |
                     BF_GPMI_CTRL0_CS                 (2)             | // must correspond
to NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)      |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)             |
                     BF_GPMI_CTRL0_XFER_COUNT         (0);
```

```
//-------------------------------------------------------------------------
// Descriptor 4: psense compare (time out check)
//-------------------------------------------------------------------------
read[3].dma_nxtcmdar = &read[4];                                    // point to the next
descriptor
read[3].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)         |       // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)         |       // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)         |       // do not wait to continue
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)         |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)         |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)         |
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)         |
                  BF_APBH_CHn_CMD_CHAIN         (1)         |       // follow chain to next
command
                  BV_FLD(APBH_CHn_CMD, COMMAND, DMA_SENSE);         // perform a sense check
read[3].dma_bar = dma_error_handler;                        // if sense check fails, branch to
error handler
//-------------------------------------------------------------------------
// Descriptor 5: read 4K page plus 65 byte meta-data Nand data
//               and send it to ECC block (DATA)
//-------------------------------------------------------------------------
read[4].dma_nxtcmdar = &read[5];                              // point to the next descriptor
read[4].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)     | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (6)     | // send 6 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)     | // wait for command to finish before
                                                          //       continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)     |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)     |
                  BF_APBH_CHn_CMD_NANDLOCK      (1)     | // prevent other DMA channels from
taking over
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)     | // ECC block generates BCH interrupt
                                                          //      on completion
                  BF_APBH_CHn_CMD_CHAIN         (1)     | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);     // no DMA transfer,
                                                                  // ECC block handles
transfer
read[4].dma_bar = NULL;                                           // field not used
// 6 words sent to the GPMI
read[4].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)       |  // read from the NAND
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)      |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)         |  // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA)  |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)         |
                     BF_GPMI_CTRL0_XFER_COUNT         (4096+218);    // eight 512 byte data
blocks
                                                                     // metadata, and parity

read[4].gpmi_compare = NULL;                              // field not used but necessary to set
eccctrl
// GPMI ECCCTRL PIO This launches the 4K byte transfer through BCH's
// bus master. Setting the ECC_ENABLE bit redirects the data flow
// within the GPMI so that read data flows to the BCH engine instead
// of flowing to the GPMI's DMA channel.
read[4].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ECC_CMD,    DECODE_8_BIT) |    // specify t = 8
mode
                       BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, ENABLE)       |    // enable ECC
module
                       BF_GPMI_ECCCTRL_BUFFER_MASK     (0X1FF);  // read all 8 data blocks
and 1 aux block
read[4].gpmi_ecccount = BF_GPMI_ECCCOUNT_COUNT(4096+218);         // specify number of bytes
                                                                  //        read from NAND
read[4].gpmi_data_ptr = &read_payload_buffer;                     // pointer for the 4K byte
                                                                  //    data area
read[4].gpmi_aux_ptr = &read_aux_buffer;                          // pointer for the 65 byte
aux area +
                                                                  // parity and syndrome
bytes for both
                                                                 // data and aux blocks.
```

```
//-------------------------------------------------------------------------
// Descriptor 6: disable ECC block
//-------------------------------------------------------------------------
read[5].dma_nxtcmdar = &read[6];                                  // point to the next descriptor
read[5].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)       | // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (3)       | // send 3 words to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (1)       | // wait for command to finish
before
                                                            //     continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)       |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(1)       | // wait for nand to be ready
                  BF_APBH_CHn_CMD_NANDLOCK      (1)       | // need nand lock to be
                                                            // thread safe while turn-off BCH
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)       |
                  BF_APBH_CHn_CMD_CHAIN         (1)       | // follow chain to next command
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[5].dma_bar = NULL;                                           // field not used
// 3 words sent to the GPMI
read[5].gpmi_ctrl0 = BV_FLD(GPMI_CTRL0, COMMAND_MODE, READ)       |
                     BV_FLD(GPMI_CTRL0, WORD_LENGTH,  8_BIT)      |
                     BV_FLD(GPMI_CTRL0, LOCK_CS,      DISABLED)   |
                     BF_GPMI_CTRL0_CS                 (2)        | // must correspond to
NAND CS used
                     BV_FLD(GPMI_CTRL0, ADDRESS,      NAND_DATA) |
                     BF_GPMI_CTRL0_ADDRESS_INCREMENT  (0)        |
                     BF_GPMI_CTRL0_XFER_COUNT         (0);
read[5].gpmi_compare = NULL;                                      // field not used but necessary to set
eccctrl
read[5].gpmi_eccctrl = BV_FLD(GPMI_ECCCTRL, ENABLE_ECC, DISABLE);  // disable the ECC block
//-------------------------------------------------------------------------
// Descriptor 7: deassert nand lock
//-------------------------------------------------------------------------
read[6].dma_nxtcmdar = NULL;                                      // not used since this is last
descriptor
read[6].dma_cmd = BF_APBH_CHn_CMD_XFER_COUNT    (0)   |   // no dma transfer
                  BF_APBH_CHn_CMD_CMDWORDS      (0)   |   // no words sent to GPMI
                  BF_APBH_CHn_CMD_WAIT4ENDCMD   (0)   |   // wait for command to finish
before
                                                            //        continuing
                  BF_APBH_CHn_CMD_SEMAPHORE     (0)   |
                  BF_APBH_CHn_CMD_NANDWAIT4READY(0)   |
                  BF_APBH_CHn_CMD_NANDLOCK      (0)   |   // relinquish nand lock
                  BF_APBH_CHn_CMD_IRQONCMPLT    (0)   |   // BCH engine generates interrupt
                  BF_APBH_CHn_CMD_CHAIN         (0)   |   // terminate DMA chain processing
                  BV_FLD(APBH_CHn_CMD, COMMAND, NO_DMA_XFER);      // no dma transfer
read[6].dma_bar = NULL;                                           // field not used
```

9.6.4.2.2 Using the Decoder

As illustrated in Figure 9-19 and the sample code in DMA Structure Code Example :

- DMA descriptor 1 prepares the NAND for data read by using the GPMI to issue a NAND read setup command byte under CLE, then sends a 5-byte address under ALE. The BCH engine is not used for these commands.
- DMA descriptor 2 issues a one-byte read execute command to the NAND device that triggers its read access. The NAND then goes not ready.
- DMA descriptor 3 performs a wait for ready operation allowing the DMA chain to remain dormant until the NAND device completes its read access time.

- DMA descriptor 5 handles the reading and error correction of the NAND data. This command's PIOs activate the BCH engine to write the read NAND data to system memory and to process it for any errors that need to be corrected. This DMA descriptor contains two PIO values that are system memory addresses pointing to the PAYLOAD data area and to the AUXILIARY data area. These addresses are used by the BCH engine's AHB master to move data into system memory and to correct it. While this example is reading an entire 4K page—payload plus metadata—it is equally possible to read just one 512-byte payload block or just the uniquely protected metadata block in a single 7 DMA structure transfer.
- DMA descriptor 6 disables the BCH engine with the NANDLOCK asserted. This is necessary to ensure that the GPMI resource is not arbitrated to another DMA channel when multiple DMA channels are active concurrently.
- DMA descriptor 7 de-asserts the NANDLOCK to free up the GPMI resource to another channel.

As the BCH block receives data from the GPMI:

- The decoder transforms the read NAND data block into a BCH code word and computes the codeword syndrome.
- If no errors are present, then the BCH block can immediately report back to firmware. This report is passed as the BCH_CTRL_COMPLETE_IRQ interrupt status bit and the associated status registers in BCH_STATUS0/1 registers.
- If an error is present, then the BCH block corrects the necessary data block or parity block bytes, if possible (not all errors are correctable).

As the BCH decoder reads the data and parity blocks, it records a special condition, i.e., that all of the bits of a payload data block or metadata block are one, including any associated parity bytes. The all-ones case for both parity and data indicates an erased block in the NAND device.

The BCH_STATUS0 register contains a 4-bit field that indicates the final status of the auxiliary block. A value of 0x0 indicates no errors found for a block.

- A value of 1 to 20 inclusive indicates that many correctable errors were found and fixed.
- A value of 0xFE indicates uncorrectable errors detected on the block.

- A value of 0xFF indicates that the block was in the special ALL ONES state and is therefore considered to be an ERASED block.
- All other values are disallowed by the hardware design.

Recall that up to eight NAND devices can have DMA chains in-flight at once, i.e. they can all be contending for access to the GPMI data bus. It is impossible to predict which NAND device will enter the BCH engine with a transfer first, because each chain includes a wait4ready command structure. As a result, firmware should look at the BCH_STATUS0_COMPLETED_CE bit field to determine which block is being reported in the status register. There is also a 16-bit HANDLE field in the GPMI_ECCCTRL register that is passed down the pipeline with each transaction. This handle field can be used to speed firmware's detection of which transaction is being reported.

These examples of reading and writing have focused on full page transfers of 4K page NAND devices. Other device configurations can be specified by changing the ECCOUNT field in the GPMI registers and reprogramming the BCH's FLASHnLAYOUTm registers.

The BCH and GPMI blocks are designed to be very efficient at reading single 512 (or 1024)-byte pages in one transaction. With no errors, the transaction takes less than 20 HCLKs longer than the time to read the raw data from the NAND.

To summarize, the APBH DMA command chain for a BCH decode operation is shown in Figure 9-19. Seven DMA command structures must be present for each NAND read transaction decoded by the BCH. The seven DMA command structures for multiple NAND read transaction blocks can be chained together to make larger units of work for the BCH, and each will produce an appropriate error report in the BCH PIO space. Multiple NAND devices can have such multiple chains scheduled. The results can come back out of order with respect to the multiple chains.

9.6.4.3 Interrupts

There are two interrupt sources used in processing BCH protected NAND read and write transfers.

Since all BCH operations are initiated by GPMI DMA command structures, the DMA completion interrupt for the GPMI is an important ISR. Both of the flow charts of Figure 9-15 and Figure 9-18 show the GPMI DMA complete ISR skeleton. In both reads and writes, the GPMI DMA completion interrupt is used to schedule work ***INTO*** the error correction pipeline. As the front end processing completes, the DMA interrupt is

generated and additional work, such as DMA chains, are passed to the GPMI DMA to keep it *fed*. For write operations, this is the only interrupt that is generated for processing the NAND write transfer.

For reads, however, two interrupts are needed. Every read is started by a GPMI DMA command chain and the front end queue is fed as described above. The back end of the read pipeline is drained by monitoring the BCH completion interrupt found int HW_BCH_CTRL_COMPLETE_IRQ.

An BCH transaction consists of reading or writing all of the blocks requested in the HW_GPMI_ECCCTRL_BUFFER_MASK bit field. As every read transaction completes, it posts the status of all of the blocks to the HW_BCH_STATUS0 and HW_BCH_STATUS1 registers and sets the completion interrupt. The five stages of the BCH read pipeline completes, one in the GPMI and four in the BCH, are independently stalled as they complete and try to deliver to the next stage in the data flow. Several of these stages can be skipped if no-errors are found or once an uncorrectable error is found in a block.

In any case, the final stage will stall if the status register is busy waiting for the CPU to take status register results. The hardware monitors the state of the HW_BCH_CTRL_COMPLETE_IRQ bit. If it is still set when the last pipeline stage is ready to post data, then the stage will stall. It follows that the next previous stage will stall when it is ready to hand off work to the final stage, and so on up the pipeline.

CAUTION

It is important that firmware read the STATUS0/1 results and save them before clearing the interrupt request bit. Otherwise, a transaction and its results could be completely lost.

9.6.4.4 Randomizer

BCH ECC has a Randomizer module that is interfaced through the GPMI APBHDMA chain. The Randomizer can generate random data based on BCH ECC encoded/decoded data. It can be employed to reduce the disturbances caused by a neighboring cell in the NAND chip, thus reducing bit errors.

To enable the Randomizer module, set GPMI_ECCCTRL[RANDOMIZER_ENABLE] to 1, then set GPMI_ECCCOUNT[RANDOMIZER_PAGE] to select randomizer page number needed to be randomized. All these registers can be programmed by the DMA chain. The randomized data should start from the zero column address and be the size of the whole NAND page. If the randomizer function is enabled,

GPMI_ECCCTRL[ENABLE_ECC] should also be enabled. To bypass BCH error correction function, set BCH_FLASHxLAYOUT0[ECC0] and BCH_FLASHxLAYOUT1[ECCN] to 0.

9.6.5 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST.

The reset process gates the clocks automatically. The example code is shown below.

```
// A soft reset can take multiple clocks to complete, so do NOT gate the
// clock when setting soft reset. The reset process will gate the clock
// automatically. Poll until this has happened before subsequently
// preparing soft-reset and clock gate
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
// asserting soft-reset
  BCH_CTRL_SET(BM_BCH_CTRL_SFTRST);
// waiting for confirmation of soft-reset
  while (!BCH_CTRL.B.CLKGATE)
  {
// busy wait
  }
// Done.
  BCH_CTRL_CLR(BM_BCH_CTRL_SFTRST);
  BCH_CTRL_CLR(BM_BCH_CTRL_CLKGATE);
```

9.6.6 BCH Memory Map/Register Definition

BCH Hardware Register Format Summary

BCH memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_4000	Hardware BCH ECC Accelerator Control Register (BCH_CTRL)	32	R/W	C000_0000h	9.6.6.1/ 2413
3300_4004	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_SET)	32	R/W	C000_0000h	9.6.6.1/ 2413
3300_4008	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_CLR)	32	R/W	C000_0000h	9.6.6.1/ 2413
3300_400C	Hardware BCH ECC Accelerator Control Register (BCH_CTRL_TOG)	32	R/W	C000_0000h	9.6.6.1/ 2413
3300_4010	Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)	32	R	0000_0010h	9.6.6.2/ 2415

Table continues on the next page...

BCH memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_4020	Hardware ECC Accelerator Mode Register (BCH_MODE)	32	R/W	0000_0000h	9.6.6.3/ 2417
3300_4030	Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR)	32	R/W	0000_0000h	9.6.6.4/ 2418
3300_4040	Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)	32	R/W	0000_0000h	9.6.6.5/ 2418
3300_4050	Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR)	32	R/W	0000_0000h	9.6.6.6/ 2419
3300_4070	Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)	32	R/W	E4E4_E4E4h	9.6.6.7/ 2419
3300_4080	Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0)	32	R/W	070A_4080h	9.6.6.8/ 2420
3300_4090	Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1)	32	R/W	10DA_4080h	9.6.6.9/ 2422
3300_40A0	Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0)	32	R/W	070A_4080h	9.6.6.10/ 2423
3300_40B0	Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1)	32	R/W	10DA_4080h	9.6.6.11/ 2425
3300_40C0	Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0)	32	R/W	070A_4080h	9.6.6.12/ 2426
3300_40D0	Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1)	32	R/W	10DA_4080h	9.6.6.13/ 2427
3300_40E0	Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0)	32	R/W	070A_4080h	9.6.6.14/ 2428
3300_40F0	Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1)	32	R/W	10DA_4080h	9.6.6.15/ 2430
3300_4100	Hardware BCH ECC Debug Register0 (BCH_DEBUG0)	32	R/W	0000_0000h	9.6.6.16/ 2431
3300_4104	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_SET)	32	R/W	0000_0000h	9.6.6.16/ 2431
3300_4108	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_CLR)	32	R/W	0000_0000h	9.6.6.16/ 2431
3300_410C	Hardware BCH ECC Debug Register0 (BCH_DEBUG0_TOG)	32	R/W	0000_0000h	9.6.6.16/ 2431
3300_4110	KES Debug Read Register (BCH_DBGKESREAD)	32	R	0000_0000h	9.6.6.17/ 2433
3300_4120	Chien Search Debug Read Register (BCH_DBGCSFEREAD)	32	R	0000_0000h	9.6.6.18/ 2433
3300_4130	Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD)	32	R	0000_0000h	9.6.6.19/ 2433
3300_4140	Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD)	32	R	0000_0000h	9.6.6.20/ 2434
3300_4150	Block Name Register (BCH_BLOCKNAME)	32	R	2048_4342h	9.6.6.21/ 2434

Table continues on the next page...

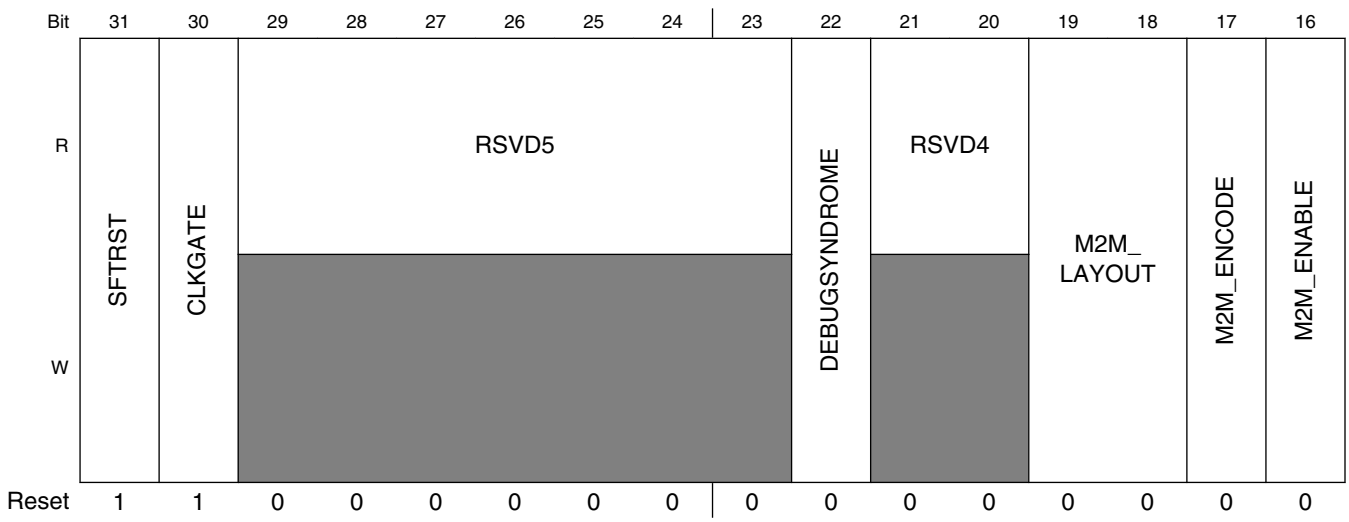
BCH memory map (continued)

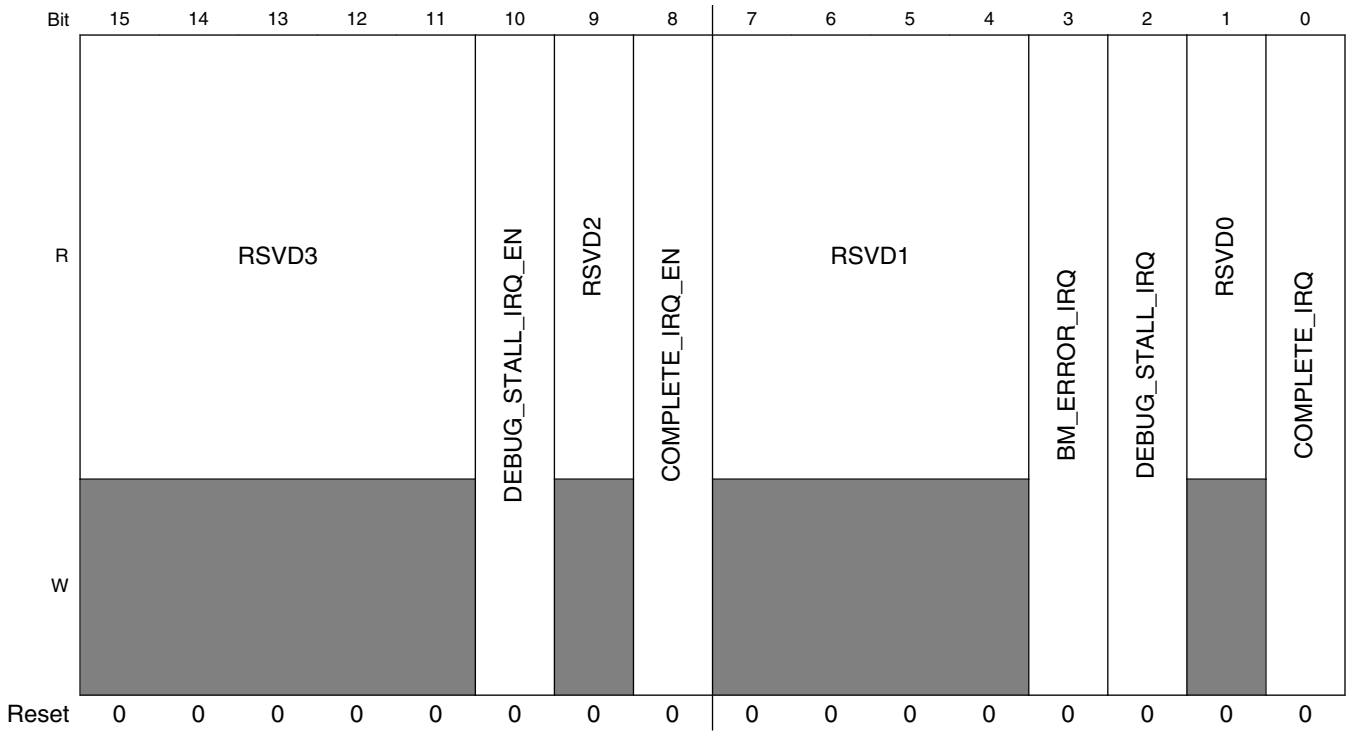
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_4160	BCH Version Register (BCH_VERSION)	32	R	0100_0000h	9.6.6.22/ 2435
3300_4170	Hardware BCH ECC Debug Register 1 (BCH_DEBUG1)	32	R/W	0000_0000h	9.6.6.23/ 2436

9.6.6.1 Hardware BCH ECC Accelerator Control Register (BCH_CTRL*n*)

The BCH CTRL provides overall control of the hardware ECC accelerator

Address: 3300_4000h base + 0h offset + (4d × i), where i=0d to 3d





BCH_CTRL*n* field descriptions

Field	Description
31 SFTRST	Set this bit to 0 to enable normal BCH operation. Set this bit to 1 (default) to disable clocking with the BCH and hold it in its reset (lowest power) state. This bit can be turned on and then off to reset the BCH block to its default state. This bit resets all state machines except for the AHB master state machine 0x0 **RUN** — Allow BCH to operate normally. 0x1 **RESET** — Hold BCH in reset.
30 CLKGATE	This bit must be set to 0 for normal operation. When set to 1 it gates off the clocks to the block. 0x0 **RUN** — Allow BCH to operate normally. 0x1 **NO_CLKS** — Do not clock BCH gates in order to minimize power consumption.
29–23 RSVD5	This field is reserved. This read-only field is reserved and always has the value 0.
22 DEBUGSYNDROME	(For debug purposes only). Enable write of computed syndromes to memory on BCH decode operations. Computed syndromes will be written to the auxiliary buffer after the status block. Syndromes will be written as padded 16-bit values.
21–20 RSVD4	This field is reserved. This read-only field is reserved and always has the value 0
19–18 M2M_LAYOUT	Selects the flash page format for memory-to-memory operations.
17 M2M_ENCODE	Selects encode (parity generation) or decode (correction) mode for memory-to-memory operations.
16 M2M_ENABLE	NOTE! WRITING THIS BIT INITIATES A MEMORY-TO-MEMORY OPERATION. The BCH module must be inactive (not processing data from the GPMI) when this bit is set. The M2M_ENCODE and

Table continues on the next page...

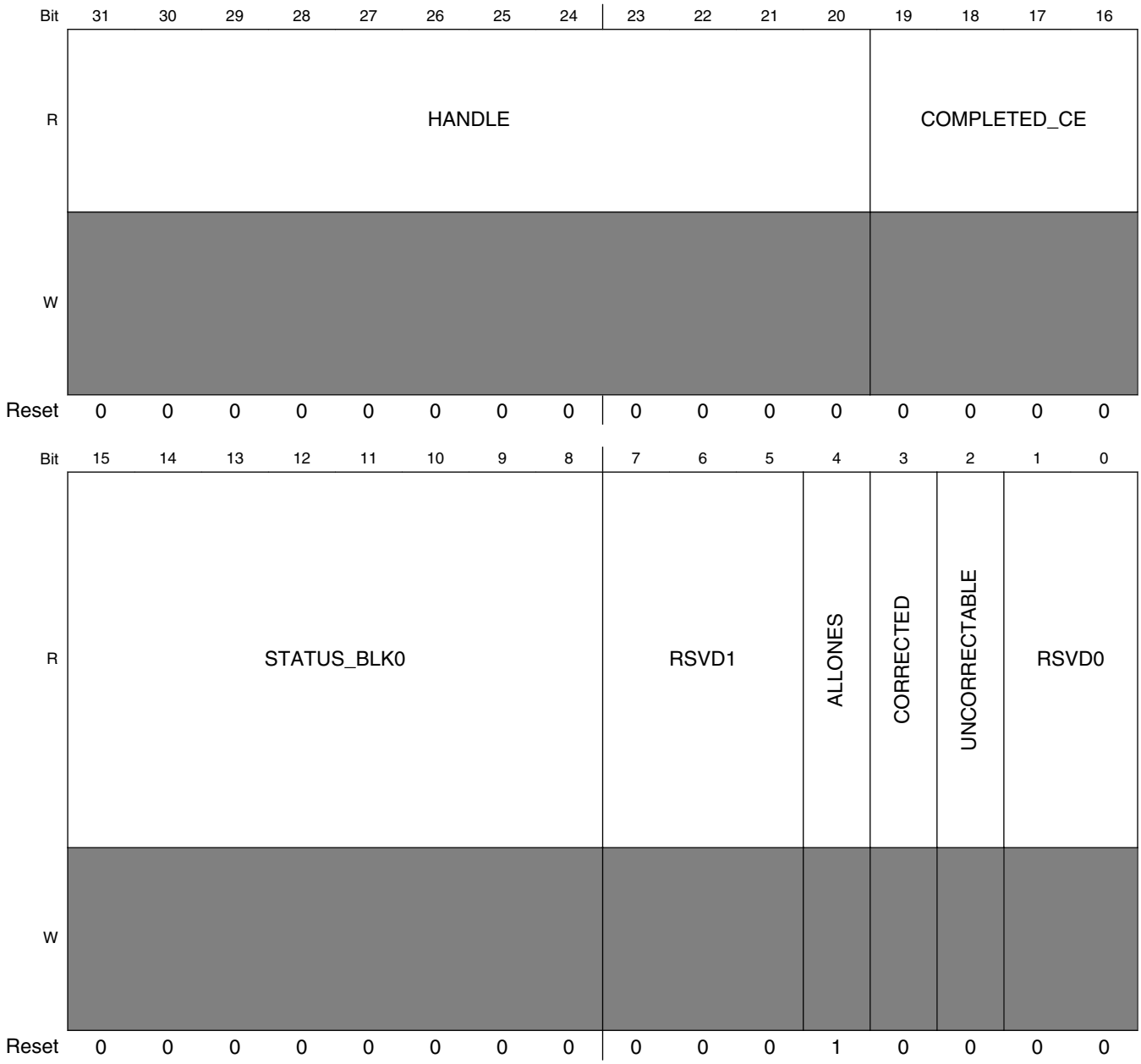
BCH_CTRL*n* field descriptions (continued)

Field	Description
	M2M_LAYOUT bits as well as the ENCODEPTR, DATAPTR, and METAPTR registers are used for memory-to-memory operations and must be correctly programmed before writing this bit.
15–11 RSVD3	This field is reserved. This read-only field is reserved and always has the value 0
10 DEBUG_STALL_ IRQ_EN	1 = interrupt on debug stall mode is enabled. The IRQ is raised on every block
9 RSVD2	This field is reserved. This read-only field is reserved and always has the value 0.
8 COMPLETE_IRQ_ EN	1 = interrupt on completion of correction is enabled.
7–4 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
3 BM_ERROR_IRQ	AHB Bus interface Error Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
2 DEBUG_STALL_IRQ	DEBUG STALL Interrupt Status. Write a 1 to the SCT clear address to clear the interrupt status bit.
1 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
0 COMPLETE_IRQ	This bit indicates the state of the external interrupt line. Write a 1 to the SCT clear address to clear the interrupt status bit. NOTE: subsequent ECC completions will be held off as long as this bit is set. Be sure to read the data from BCH_STATUS0, 1 before clearing this interrupt bit.

9.6.6.2 Hardware ECC Accelerator Status Register 0 (BCH_STATUS0)

The BCH STAT register provides visibility into the run-time status of the BCH and status information when processing is complete. It provides overall status of the hardware ECC accelerator.

Address: 3300_4000h base + 10h offset = 3300_4010h



BCH_STATUS0 field descriptions

Field	Description
31–20 HANDLE	Software supplies a 12 bit handle for this transfer as part of the GPMI DMA PIO operation that started the transaction. That handle passes down the pipeline and ends up here at the time the BCH interrupt is signaled.
19–16 COMPLETED_CE	This is the chip enable number corresponding to the NAND device from which this data came.
15–8 STATUS_BLK0	Count of symbols in error during processing of first block of flash (metadata block). The number of errors reported will be in the range of 0 to the ECC correction level for block 0. 0x00 **ZERO** — No errors found on block.

Table continues on the next page...

BCH_STATUS0 field descriptions (continued)

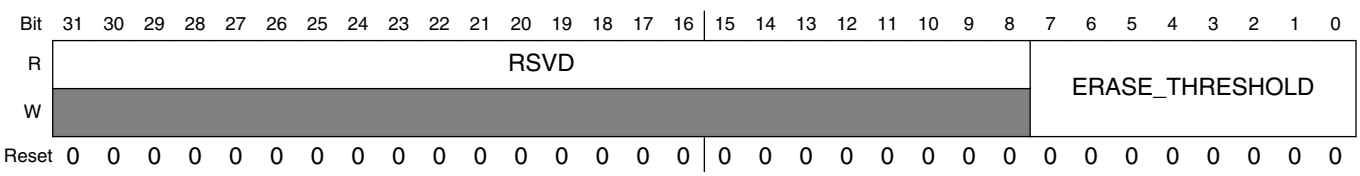
Field	Description
	0x01 **ERROR1** — One error found on block. 0x02 **ERROR2** — One errors found on block. 0x03 **ERROR3** — One errors found on block. 0x04 **ERROR4** — One errors found on block. 0xFE **UNCORRECTABLE** — Block exhibited uncorrectable errors. 0xFF **ERASED** — Page is erased.
7–5 RSVD1	This field is reserved. This read-only field is reserved and always has the value 0.
4 ALLONES	1 = All data bits of this transaction are ONE.
3 CORRECTED	1 = At least one correctable error encountered during last processing cycle.
2 UNCORRECTABLE	1 = Uncorrectable error encountered during last processing cycle.
RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.

9.6.6.3 Hardware ECC Accelerator Mode Register (BCH_MODE)

The BCH MODE register provides additional mode controls.

Contains additional global mode controls for the BCH engine.

Address: 3300_4000h base + 20h offset = 3300_4020h



BCH_MODE field descriptions

Field	Description
31–8 RSVD	This field is reserved. This read-only field is reserved and always has the value 0.
ERASE_ THRESHOLD	This value indicates the maximum number of zero bits on a flash subpage for it to be considered erased. For SLC NAND devices, this value should be programmed to 0 (meaning that the entire page should consist of bytes of 0xFF. For MLC NAND devices, bit errors may occur on reads (even on blank pages), so this threshold can be used to tune the erased page checking algorithm.

9.6.6.4 Hardware BCH ECC Loopback Encode Buffer Register (BCH_ENCODEPTR)

When performing memory to memory operations, indicates the address of the encode buffer. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register.

For memory to memory operations, this register is used as the pointer to the encoded data, which is an output when encoding and an input while decoding.

Address: 3300_4000h base + 30h offset = 3300_4030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_ENCODEPTR field descriptions

Field	Description
ADDR	Address pointer to encode buffer. This is the source for decode operations and the destination for encode operations. This value must be aligned on a 4 bytes boundary.

9.6.6.5 Hardware BCH ECC Loopback Data Buffer Register (BCH_DATAPTR)

When performing memory to memory operations, indicates the address of the data buffer.

For memory to memory operations, this register is used as the pointer to the data to encode or the destination buffer for decode operations.

Address: 3300_4000h base + 40h offset = 3300_4040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DATAPTR field descriptions

Field	Description
ADDR	Address pointer to data buffer. This is the source for encode operations and the destination for decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 byte boundary.

9.6.6.6 Hardware BCH ECC Loopback Metadata Buffer Register (BCH_METAPTR)

When performing memory to memory operations, indicates the address of the metadata buffer.

For memory to memory operations, this register is used as the pointer to the metadata to encode or the extracted metadata for decode operations.

Address: 3300_4000h base + 50h offset = 3300_4050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_METAPTR field descriptions

Field	Description
ADDR	Address pointer to metadata buffer. This is the source for encode metadata read operations and the destination for metadata decode operations. This register should be programmed before writing a 1 to the M2M_ENABLE bit in the CTRL register. This value must be aligned on a 4 bytes boundary.

9.6.6.7 Hardware ECC Accelerator Layout Select Register (BCH_LAYOUTSELECT)

The BCH LAYOUTSELECT register provides a mapping of chip selects to layout registers.

When the BCH engine receives a request to process a data block from the GPMI interface, it will use this register to map the incoming chip select to one of the four possible flash layout registers

Address: 3300_4000h base + 70h offset = 3300_4070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CS15_SELECT		CS14_SELECT		CS13_SELECT		CS12_SELECT		CS11_SELECT		CS10_SELECT		CS9_SELECT		CS8_SELECT	
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CS7_SELECT		CS6_SELECT		CS5_SELECT		CS4_SELECT		CS3_SELECT		CS2_SELECT		CS1_SELECT		CS0_SELECT	
Reset	1	1	1	0	0	1	0	0	1	1	1	0	0	1	0	0

BCH_LAYOUTSELECT field descriptions

Field	Description
31–30 CS15_SELECT	Selects which layout is used for chip select 15.
29–28 CS14_SELECT	Selects which layout is used for chip select 14.
27–26 CS13_SELECT	Selects which layout is used for chip select 13.
25–24 CS12_SELECT	Selects which layout is used for chip select 12.
23–22 CS11_SELECT	Selects which layout is used for chip select 11.
21–20 CS10_SELECT	Selects which layout is used for chip select 10.
19–18 CS9_SELECT	Selects which layout is used for chip select 9.
17–16 CS8_SELECT	Selects which layout is used for chip select 8.
15–14 CS7_SELECT	Selects which layout is used for chip select 7.
13–12 CS6_SELECT	Selects which layout is used for chip select 6.
11–10 CS5_SELECT	Selects which layout is used for chip select 5.
9–8 CS4_SELECT	Selects which layout is used for chip select 4.
7–6 CS3_SELECT	Selects which layout is used for chip select 3.
5–4 CS2_SELECT	Selects which layout is used for chip select 2.
3–2 CS1_SELECT	Selects which layout is used for chip select 1.
CS0_SELECT	Selects which layout is used for chip select 0.

9.6.6.8 Hardware BCH ECC Flash 0 Layout 0 Register (BCH_FLASH0LAYOUT0)

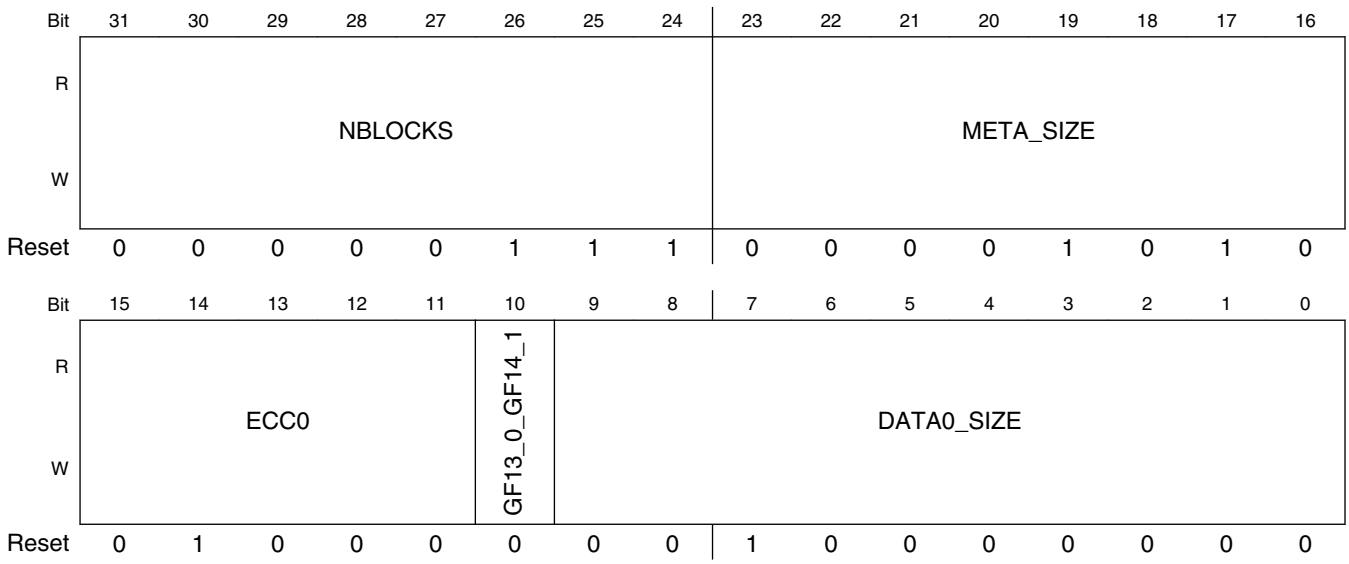
The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT1 register to control the format for the devices selecting layout 0 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data,

metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 3300_4000h base + 80h offset = 3300_4080h



BCH_FLASH0LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14

Table continues on the next page...

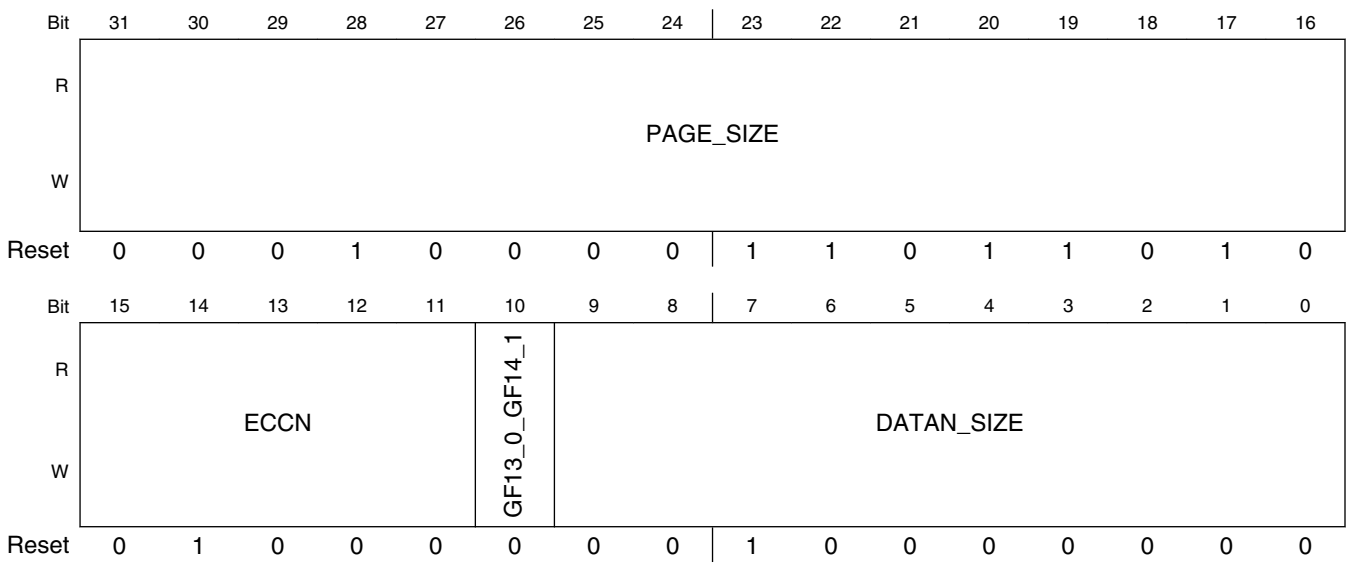
BCH_FLASH0LAYOUT0 field descriptions (continued)

Field	Description
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block only contains metadata.

9.6.6.9 Hardware BCH ECC Flash 0 Layout 1 Register (BCH_FLASH0LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH0LAYOUT0 register to control the format for the device selecting layout 0 in the LAYOUTSELECT register.

Address: 3300_4000h base + 90h offset = 3300_4090h



BCH_FLASH0LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : —

Table continues on the next page...

BCH_FLASH0LAYOUT1 field descriptions (continued)

Field	Description
	0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

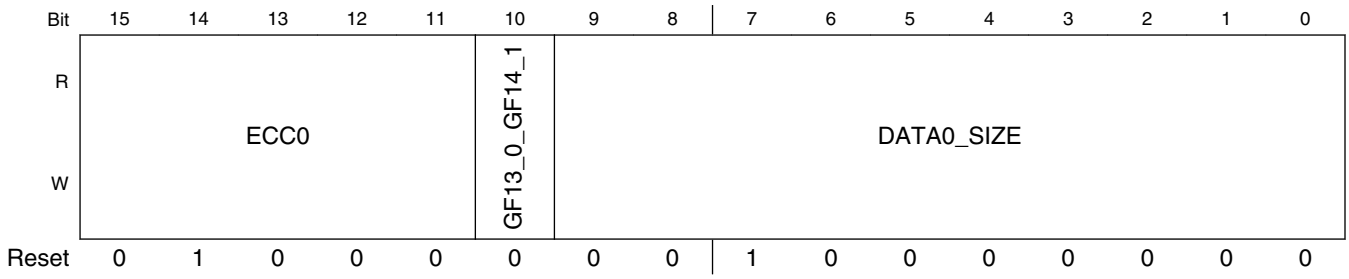
9.6.6.10 Hardware BCH ECC Flash 1 Layout 0 Register (BCH_FLASH1LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT1 register to control the format for the devices selecting layout 1 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 3300_4000h base + A0h offset = 3300_40A0h



BCH_FLASH1LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design supports from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data is in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will contains metadata.

9.6.6.11 Hardware BCH ECC Flash 1 Layout 1 Register (BCH_FLASH1LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH1LAYOUT0 register to control the format for the device selecting layout 1 in the LAYOUTSELECT register.

Address: 3300_4000h base + B0h offset = 3300_40B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PAGE_SIZE															
W																
Reset	0	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECCN					GF13_0_GF14_1	DATAN_SIZE									
W																
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

BCH_FLASH1LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

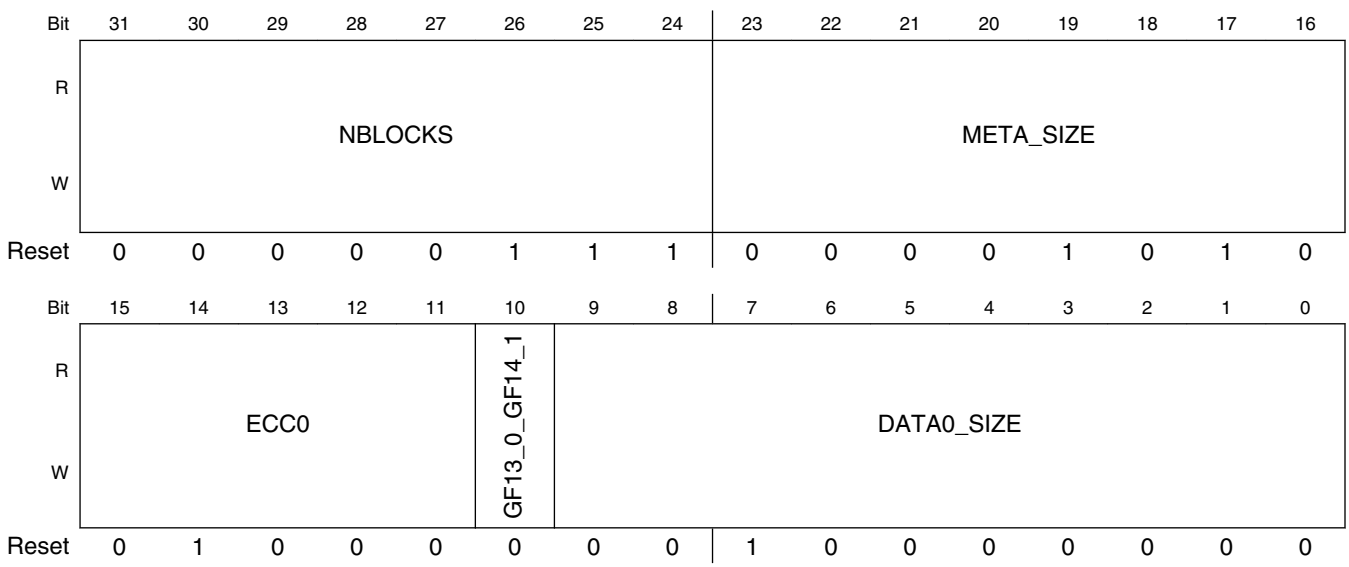
9.6.6.12 Hardware BCH ECC Flash 2 Layout 0 Register (BCH_FLASH2LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT1 register to control the format for the devices selecting layout 2 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 3300_4000h base + C0h offset = 3300_40C0h



BCH_FLASH2LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that eight subsequent blocks are present for a total of nine blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.

Table continues on the next page...

BCH_FLASH2LAYOUT0 field descriptions (continued)

Field	Description
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

9.6.6.13 Hardware BCH ECC Flash 2 Layout 1 Register (BCH_FLASH2LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH2LAYOUT0 register to control the format for the device selecting layout 2 in the LAYOUTSELECT register.

Address: 3300_4000h base + D0h offset = 3300_40D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PAGE_SIZE															
Reset	0	0	0	1	0	0	0	0	1	1	0	1	1	0	1	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ECCN					GF13_0_GF14_1	DATAN_SIZE									
Reset	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0

BCH_FLASH2LAYOUT1 field descriptions

Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

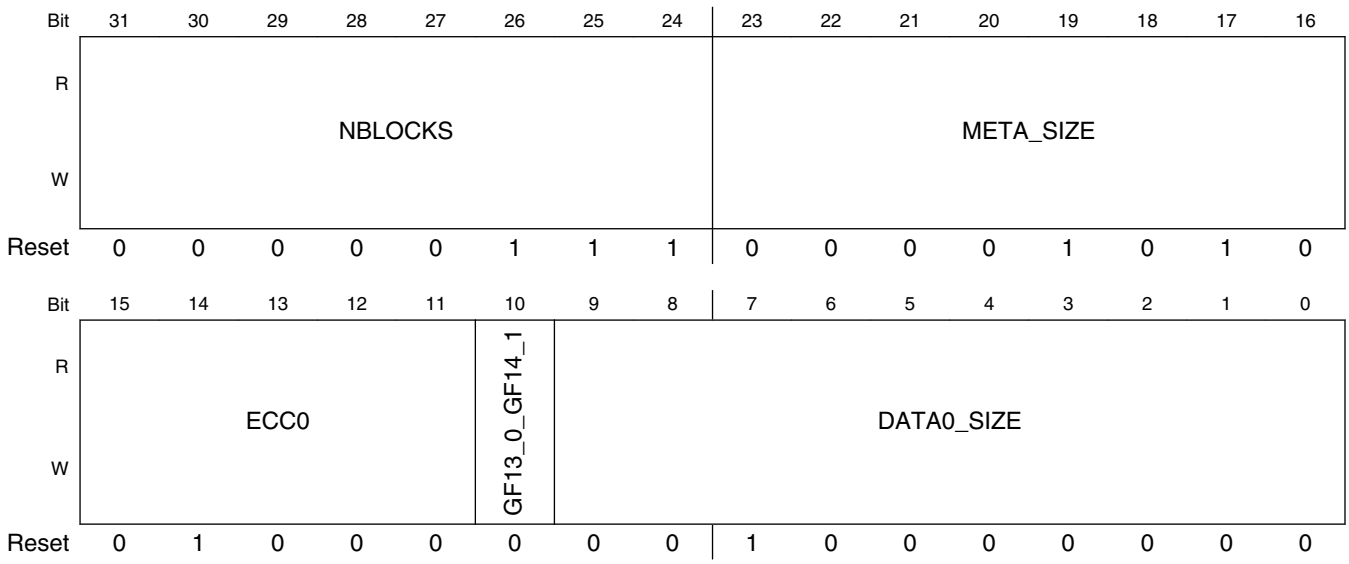
9.6.6.14 Hardware BCH ECC Flash 3 Layout 0 Register (BCH_FLASH3LAYOUT0)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT1 register to control the format for the devices selecting layout 3 in the LAYOUTSELECT register.

Each pair of layout registers describes one of four supported flash configurations. Software should program the LAYOUTSELECT register for each supported GPMI chip select to select from one of the four layout values. Each pair of registers contains settings that are used by the BCH block while reading / writing the flash page to control data, metadata, and flash page sizes as well as the ECC correction level. The first block written to flash can be programmed to have different ECC, metadata, and data sizes from subsequent data blocks on the device. In addition, the number of blocks stored on a page of flash is not fixed, but instead is determined by the number of bytes consumed by the initial (block 0) and subsequent data blocks.

See sections Flash Page Layout and Determining the ECC layout for a device for more detail information on setting up the flash layout registers.

Address: 3300_4000h base + E0h offset = 3300_40E0h



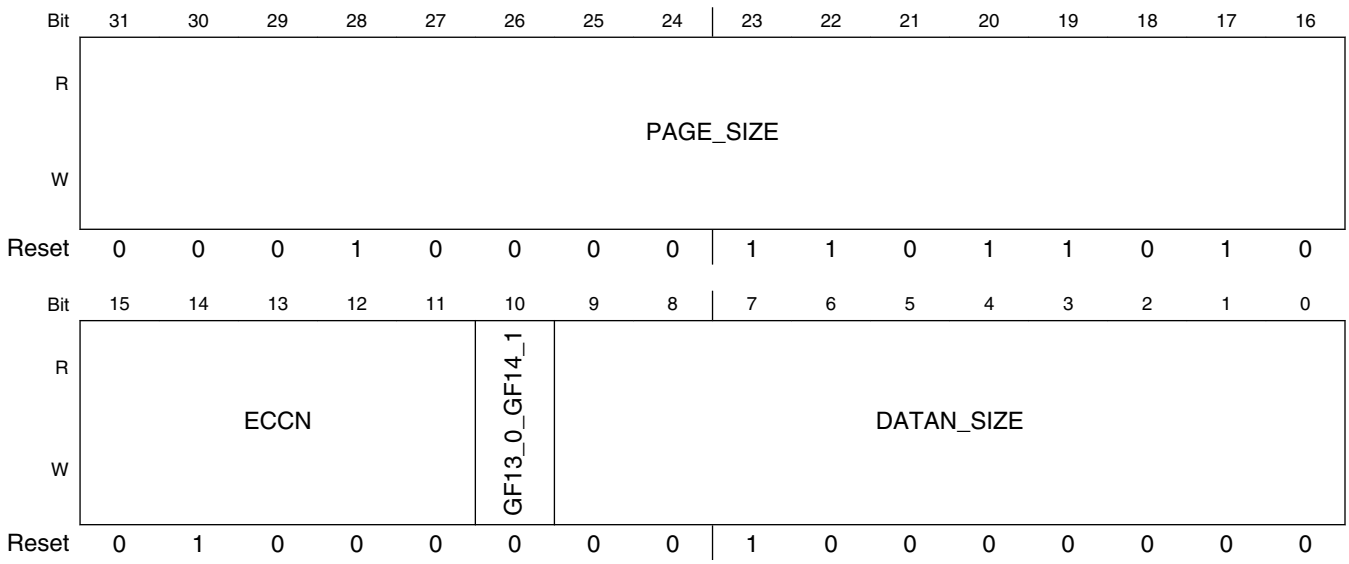
BCH_FLASH3LAYOUT0 field descriptions

Field	Description
31–24 NBLOCKS	Number of subsequent blocks on the flash page (excluding the data0 block). A value of 0 indicates that only the DATA0 block is present and a value of 8 indicates that 8 subsequent blocks are present for a total of 9 blocks on the flash (including the DATA0 block). Any values from 0 to 255 are supported by the hardware.
23–16 META_SIZE	Indicates the size of the metadata (in bytes) to be stored on a flash page. The BCH design support from 0 to 255 bytes for metadata—if set to 0, no metadata will be stored. Metadata is stored before the associated data in block 0. If the DATA0_SIZE field is programmed to a 0, then metadata effectively be stored with its own parity. When both the metadata and data0 fields are programmed with non-zero values, the first block will contain both portions of data and will be covered by a single parity block.
15–11 ECC0	Indicates the ECC level for the first block on the flash page. The first block covers metadata plus the associated data from the DATA0_SIZE field. 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATA0_SIZE	Indicates the size of the data 0 block (in DWORDS / four bytes) to be stored on the flash page. If set to 0, the first block will only contain metadata.

9.6.6.15 Hardware BCH ECC Flash 3 Layout 1 Register (BCH_FLASH3LAYOUT1)

The flash format register contains a description of the logical layout of data on the flash device. This register is used in conjunction with the FLASH3LAYOUT0 register to control the format for the device selecting layout 3 in the LAYOUTSELECT register.

Address: 3300_4000h base + F0h offset = 3300_40F0h



BCH_FLASH3LAYOUT1 field descriptions

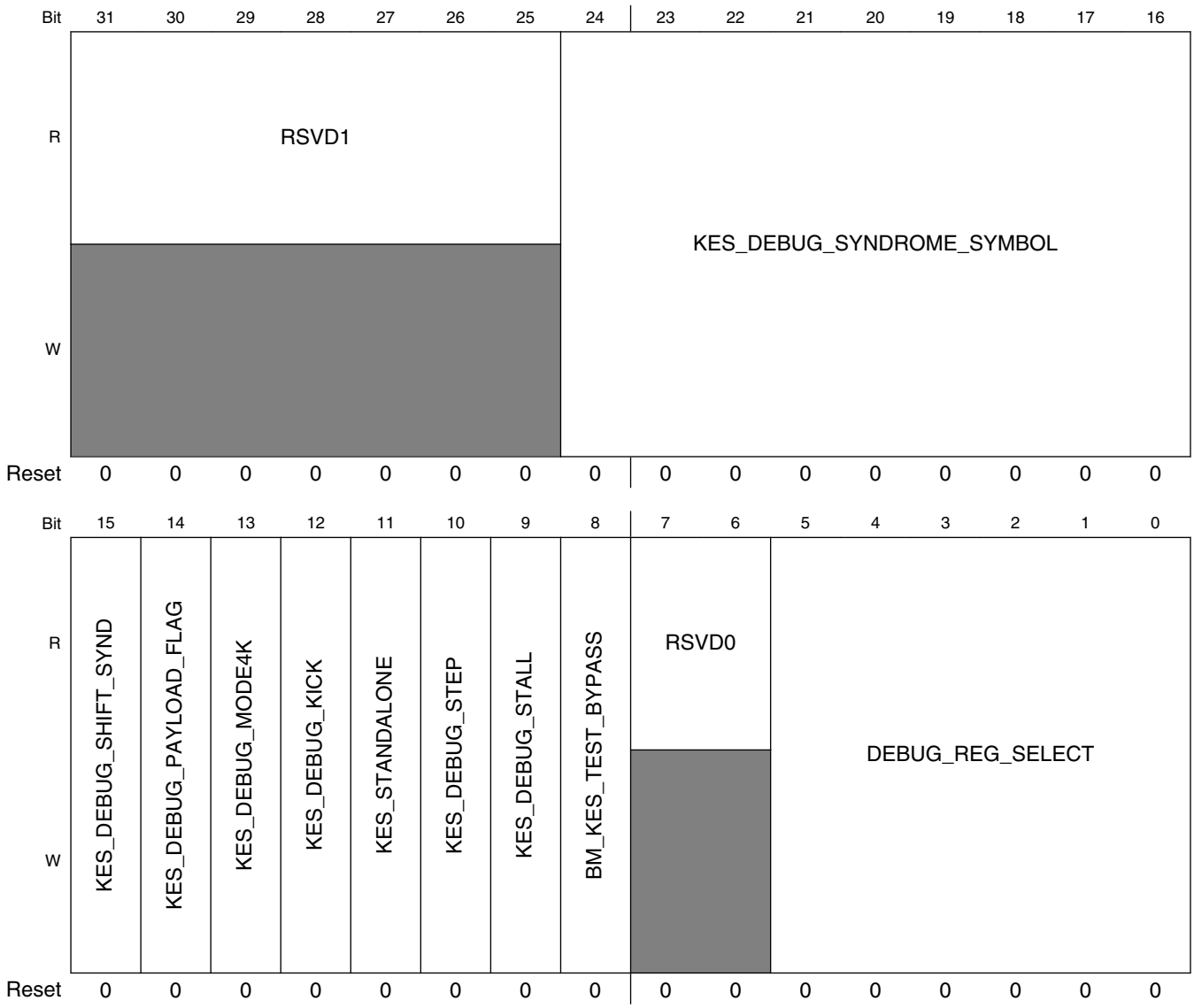
Field	Description
31–16 PAGE_SIZE	Indicates the total size of the flash page (in bytes). This should be set to the page size including spare area. The page size is programmable to accomodate different flash configurations that may be available in the future.
15–11 ECCN	Indicates the ECC level for the subsequent blocks on the flash page (blocks 1-n). Subsequent blocks only contain data (no metadata). 0x0 **NONE** — No ECC to be performed 0x1 **ECC2** — ECC 2 to be performed 0x2 **ECC4** — ECC 4 to be performed : — 0x1E **ECC60** — ECC 60 to be performed 0x1F **ECC62** — ECC 62 to be performed
10 GF13_0_GF14_1	Select GF13 or GF14: 0-GF13; 1-GF14
DATAN_SIZE	Indicates the size of the subsequent data blocks (in DWORDS / four bytes) to be stored on the flash page. The size of subsequent data blocks does not have to match the data size for block 0, which is important when metadata is stored separately or for balancing the amount of data stored in each block.

9.6.6.16 Hardware BCH ECC Debug Register0 (BCH_DEBUG0*n*)

The hardware BCH accelerator internal state machines and signals can be seen in the ECC debug register.

The BCH_DEBUG0 register provides access to various internal state information which might prove useful during hardware debug and validation.

Address: 3300_4000h base + 100h offset + (4d × i), where i=0d to 3d



BCH_DEBUG0*n* field descriptions

Field	Description
31–25 RSVD1	This field is reserved.

Table continues on the next page...

BCH_DEBUG0*n* field descriptions (continued)

Field	Description
	This read-only field is reserved and always has the value 0.
24–16 KES_DEBUG_ SYNDROME_ SYMBOL	The 9 bit value in this bit field shifts into the syndrome register array at the input of the KES engine whenever BCH_DEBUG0_KES_DEBUG_SHIFT_SYND is toggled. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
15 KES_DEBUG_ SHIFT_SYND	Toggling this bit causes the value in BCH_DEBUG0_KES_SYNDROME_SYMBOL to be shift into the syndrome register array at the input to the KES engine. After shifting in 16 symbols, one can kick off both KES and CF cycles by toggling BCH_DEBUG0_KES_DEBUG_KICK. Make sure that set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
14 KES_DEBUG_ PAYLOAD_FLAG	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input payload flag. 0x1 **DATA** — Payload is set for 512 bytes data block. 0x1 **AUX** — Payload is set for 65 or 19 bytes auxiliary block.
13 KES_DEBUG_ MODE4K	When running the stand alone debug mode on the error calculator, the state of this bit is presented to the KES engine as the input mode (4K or 2K pages). 0x1 **4k** — Mode is set for 4K NAND pages. 0x1 **2k** — Mode is set for 2K NAND pages.
12 KES_DEBUG_ KICK	Toggling causes KES engine FSM to start as if kick by the Bus Master. This allows stand alone testing of the KES and Chien Search engines. Be sure to set KES_BCH_DEBUG0_KES_STANDALONE mode to 1 before kicking.
11 KES_ STANDALONE	Set to one, cause the KES engine to suppress toggling the KES_BM_DONE signal to the bus master and suppress toggling the CF_BM_DONE signal by the CF engine. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
10 KES_DEBUG_ STEP	Toggling this bit causes the KES FSM to skip passed the stall state if it is in DEBUG_STALL mode and completed processing a block.
9 KES_DEBUG_ STALL	Set to one to cause KES FSM to stall after notifying Chien search engine to start processing its block but before notifying the bus master that the KES computation is complete. This allows a diagnostic to stall the FSM after each blocks key equations are solved. This also has the effect of stalling the CSFE search engine so it's state can be examined after it finishes processing the KES stalled block. 0x0 **NORMAL** — KES FSM proceeds to next block supplied by bus master. 0x1 **WAIT** — KES FSM waits after current equations are solved and the search engine is started.
8 BM_KES_TEST_ BYPASS	1 = Point all SYND_GEN writes to dummy area at the end of the AUXILLIARY block so that diagnostics can preload all payload, parity bytes and computed syndrome bytes for test the KES engine. 0x0 **NORMAL** — Bus master address generator for SYND_GEN writes operates normally. 0x1 **TEST_MODE** — Bus master address generator always addresses last four bytes in Auxiliary block.
7–6 RSVD0	This field is reserved. This read-only field is reserved and always has the value 0.
DEBUG_REG_ SELECT	The value loaded in this bit field is used to select the internal register state view of KES engine or the Chien search engine.

9.6.6.17 KES Debug Read Register (BCH_DBGKESREAD)

The hardware BCH ECC accelerator key equation solver internal state machines and signals can be seen in the ECC debug registers.

Address: 3300_4000h base + 110h offset = 3300_4110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DBGKESREAD field descriptions

Field	Description
VALUES	This register returns the ROM BIST CRC value after a BIST test.

9.6.6.18 Chien Search Debug Read Register (BCH_DBGCSFEREAD)

The hardware BCH ECC accelerator Chien Search internal state machines and signals can be seen in the ECC debug registers.

Address: 3300_4000h base + 120h offset = 3300_4120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DBGCSFEREAD field descriptions

Field	Description
VALUES	Reserved

9.6.6.19 Syndrome Generator Debug Read Register (BCH_DBGSYNDGENREAD)

The hardware BCH ECC accelerator syndrome generator internal state machines and signals can be seen in the ECC debug registers.

Address: 3300_4000h base + 130h offset = 3300_4130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DBGSYNDGENREAD field descriptions

Field	Description
VALUES	Reserved

9.6.6.20 Bus Master and ECC Controller Debug Read Register (BCH_DBGAHBMREAD)

The hardware BCH ECC accelerator bus master, ECC controller internal state machines, and signals can be seen in the ECC debug registers.

Address: 3300_4000h base + 140h offset = 3300_4140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VALUES																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

BCH_DBGAHBMREAD field descriptions

Field	Description
VALUES	Reserved

9.6.6.21 Block Name Register (BCH_BLOCKNAME)

Read only view of the block name string BCH.

Fixed pattern read only value is for test purposes. It can be read as an ASCII string with the zero termination coming from the first byte of the BLOCKVERSION register.

Address: 3300_4000h base + 150h offset = 3300_4150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME																															
W																																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	0	0	1	1	0	1	0	0	0	0	1	0

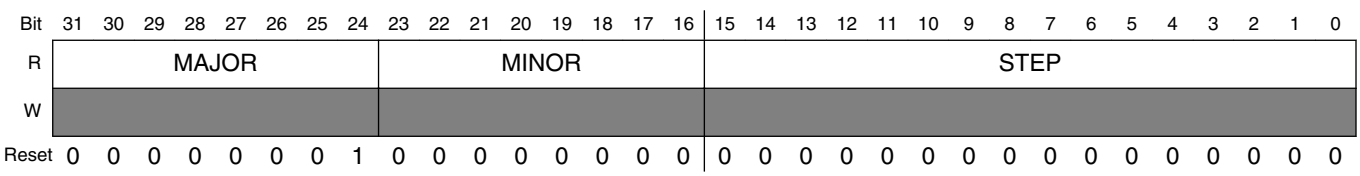
BCH_BLOCKNAME field descriptions

Field	Description
NAME	The name is in the ASCII characters BCH (0x20, H, C, B).

9.6.6.22 BCH Version Register (BCH_VERSION)

This register always returns a known read value for debug purposes and indicates the version of the block and RTL version in use.

Address: 3300_4000h base + 160h offset = 3300_4160h



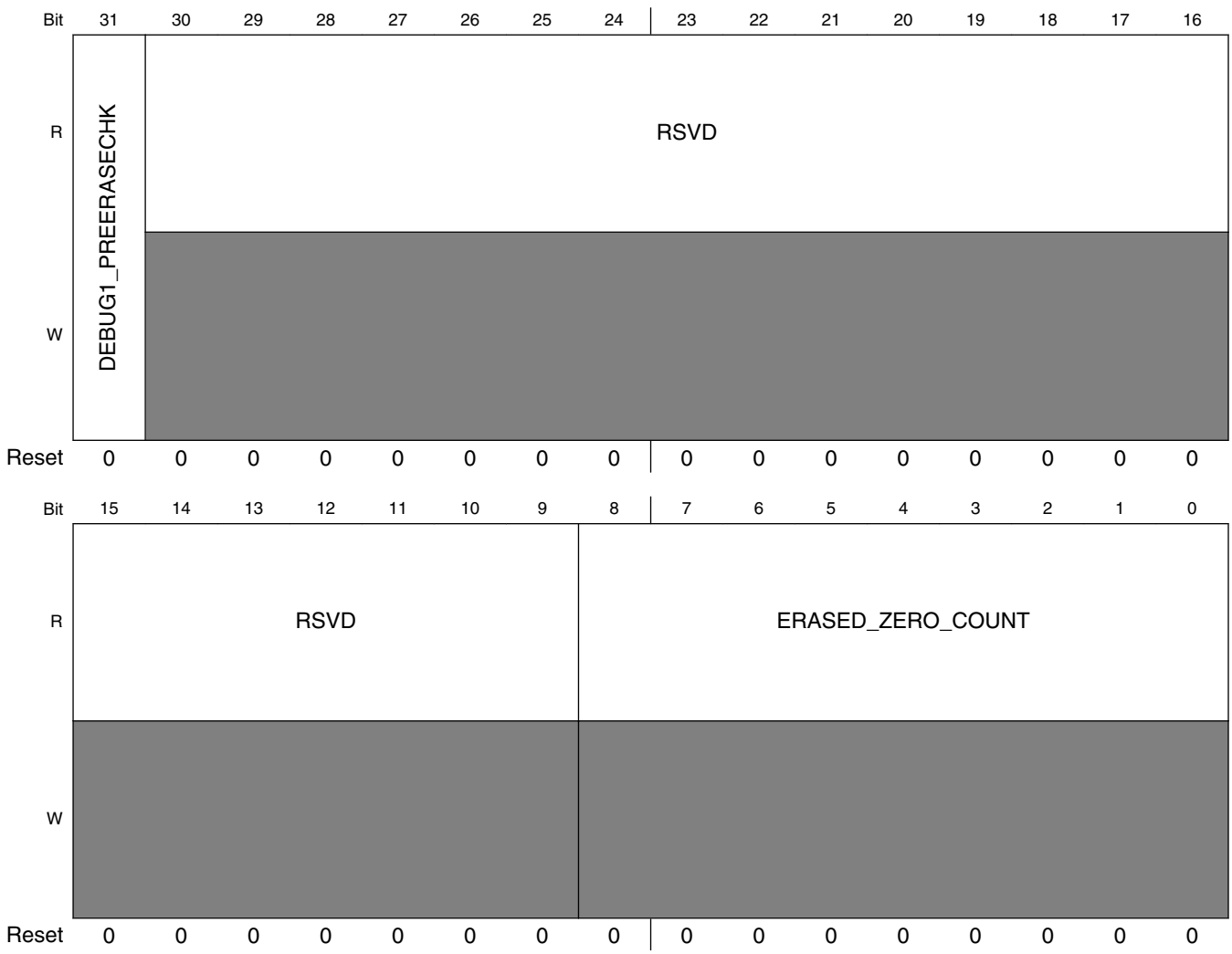
BCH_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value indicates the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value indicates the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

9.6.6.23 Hardware BCH ECC Debug Register 1 (BCH_DEBUG1)

The BCH_DEBUG1 register provides erased zero count information and pre-erase check.

Address: 3300_4000h base + 170h offset = 3300_4170h



BCH_DEBUG1 field descriptions

Field	Description
31 DEBUG1_ PREERASECHK	Blank page enables pre-erase check. 0x0 Turn off pre-erase check 0x1 Turn on pre-erase check
30–9 RSVD	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

BCH_DEBUG1 field descriptions (continued)

Field	Description
ERASED_ ZERO_COUNT	The zero counts on one page.

9.7 General Purpose Media Interface (GPMI)

9.7.1 Overview

The GPMI controller is a flexible interface to supporting up to four NAND flash chip selects.

- ONFI3.2, DDR Mode, Samsung / Toshiba Toggle NAND protocol compatible.
- Fully configurable address and command behavior, providing support for future devices not yet specified.

The GPMI resides on the APBH. The GPMI also provides an interface to the BCH module to allow direct parity processing.

Registers are clocked on the HCLK domain. The I/O and pin timing are clocked on a dedicated GPMICLK domain. GPMICLK can be set to maximize I/O performance.

The following figure shows a block diagram of the GPMI controller.

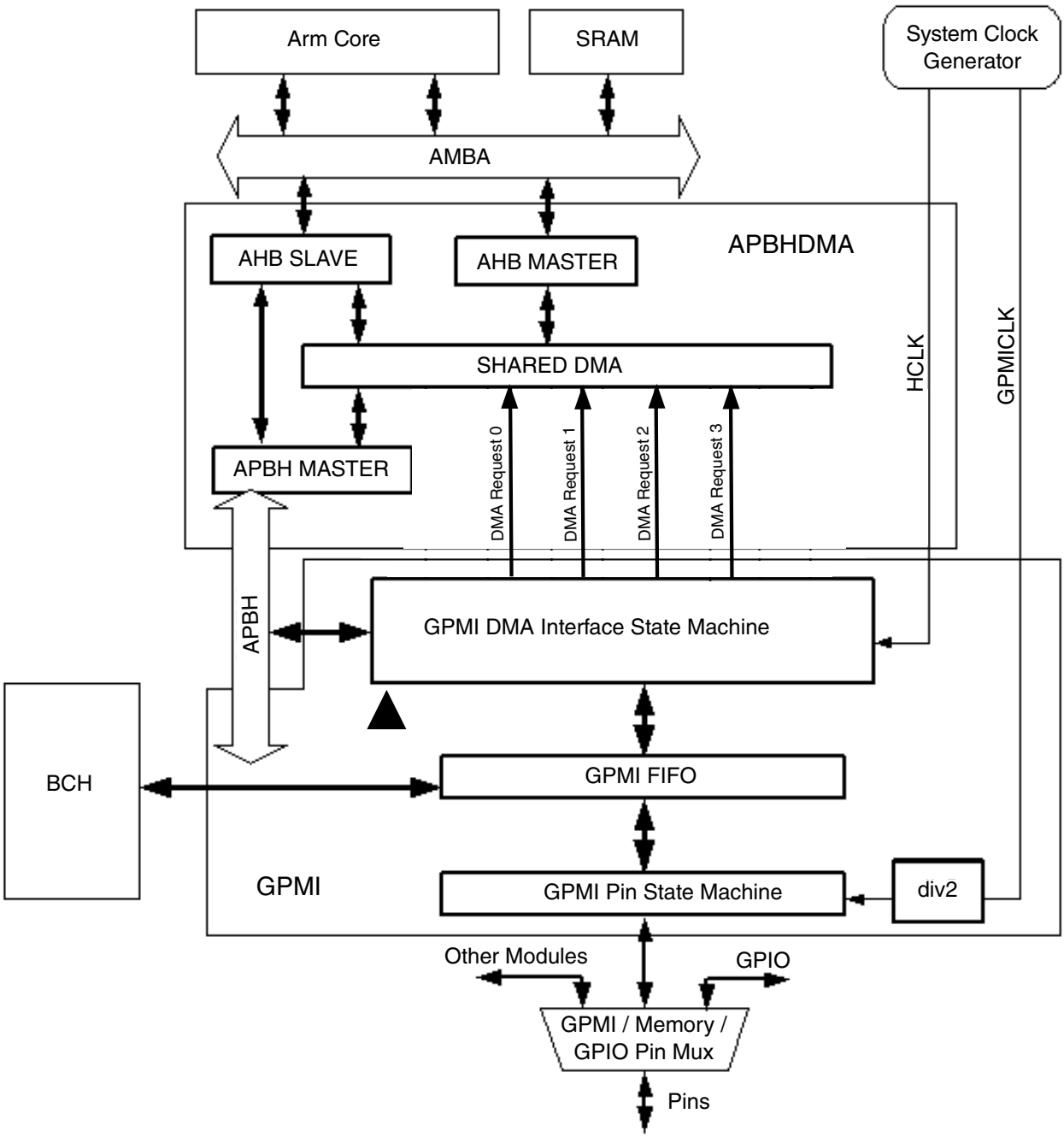


Figure 9-20. General-Purpose Media Interface Controller Block Diagram

9.7.2 Clocks

The table found here describes the clock sources for GPMI.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 9-12. GPMI Clocks

Clock name	Clock Root	Description
bch_input_apb_clk		BCH to APBH input clock
gpmi_bch_input_bch_clk		BCH input clock
gpmi_bch_input_gpmi_io_clk		GPMI IO input clock
gpmi_input_apb_clk		GPMI to APBH clock

9.7.3 GPMI NAND Mode

The general-purpose media interface has several features to efficiently support NAND:

- Individual chip select pins and ganged ready/busy pin for up to four NANDs.
- Individual state machine and DMA channel for each chip select.
- Special command modes work with DMA controller to perform all normal NAND functions without CPU intervention.
- Configurable timing based on a dedicated clock allows optimal balance of high NAND performance and low system power.

GPMI and DMA have been designed to handle complex multi-page operations without CPU intervention. The DMA uses a linked descriptor function with branching capability to automatically handle all of the operations needed to read/write multiple pages:

- **Data/Register Read/Write**-The GPMI can be programmed to read or write multiple cycles to the NAND address, command or data registers.
- **Wait for NAND Ready**-The GPMI's Wait-for-Ready mode can monitor the ready/busy signal of a single NAND flash and signal the DMA when the device has become ready. It also has a time-out counter and can indicate to the DMA that a time-out error has occurred. The DMAs can conditionally branch to a different descriptor in the case of an error.
- **Check Status**-The Read-and-Compare mode allows the GPMI to check NAND status against a reference. If an error is found, the GPMI can instruct the DMA to branch to an alternate descriptor, which attempts to fix the problem or asserts a CPU IRQ.

9.7.3.1 Multiple NAND Support

The GPMI supports up to four NAND chip selects, with ganged ready/busy pins. Since they share a data bus and control lines, the GPMI can only actively communicate with a single NAND at a time. However, all NANDs can concurrently perform internal read, write, or erase operations. With fast NAND flash and software support for concurrent NAND operations, this architecture allows the total throughput to approach the data bus speed, which can be as high as 50 MB/s (8-bit bus running at 50 MHz single clock edge) in asynchronous mode and 200MB/s (8-bit bus running at 100MHz both clock edges) in Source Synchronous mode.

There are two options for controlling the four NAND chip selects via the DMA interface. The first option is the one to one mapping, where the each DMA channel is tied to its own NAND chip select. For example DMA channel 'n' accesses only NAND attached to chip select 'n'. The second option is the decoupled mode where a DMA channel can access any or all NAND chip selects connected to the GPMI. A DMA channel will signify the NAND chip select it wants to access by writing its chip select value in the GPMI_CTRL0[CS] field and setting the GPMI_CTRL1[DECOUPLE_CS] to 1. This option is useful if software chooses to use only one DMA channel to access all the attached NAND devices.

9.7.3.2 GPMI NAND Timing and Clocking

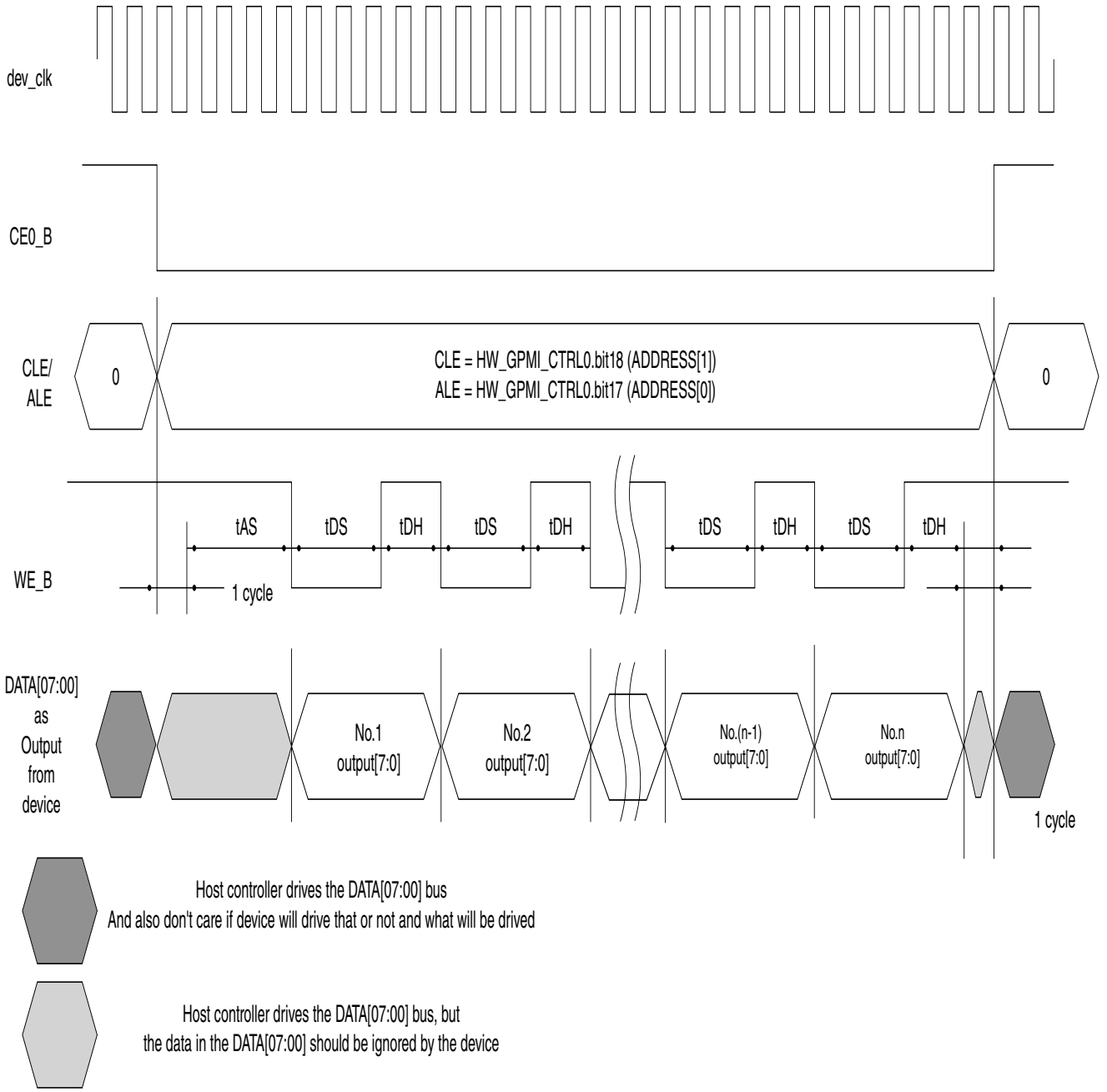
The dedicated clock, GPMICLK, is used as a timing reference for NAND flash I/O. Since various NANDs have different timing requirements, GPMICLK may need to be adjusted for each application.

While the actual pin timings are limited by the NAND chips used and the I/O pad configuration, the GPMI can support data bus speeds of up to 200 MHz x 8 bits. The actual read/write strobe timing parameters are adjusted as indicated in the register descriptions in Memory Map.

9.7.3.3 Basic NAND Timing

9.7.3.3.1 NAND Asynchronous Timing

Figure 9-22 and illustrates the operation of the output (from host to device) timing parameters in NAND ONFI asynchronous mode.



- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup: in this example, Address_Setup = 4, tAS is equal to 4 dev_clk cycles.
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup; in this example, Data_Setup = 3, tDS is equal to 3 dev_clk cycles
- tDH is configuarble by programming HW_GPMI_TIMING0 Data_Hold: in this example, Data_Hold = 2, tDH is equal to 2 dev_clk cycles
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 9-21. Asynchronous Mode Basic Write Timing Diagram (command write, address write, or data write

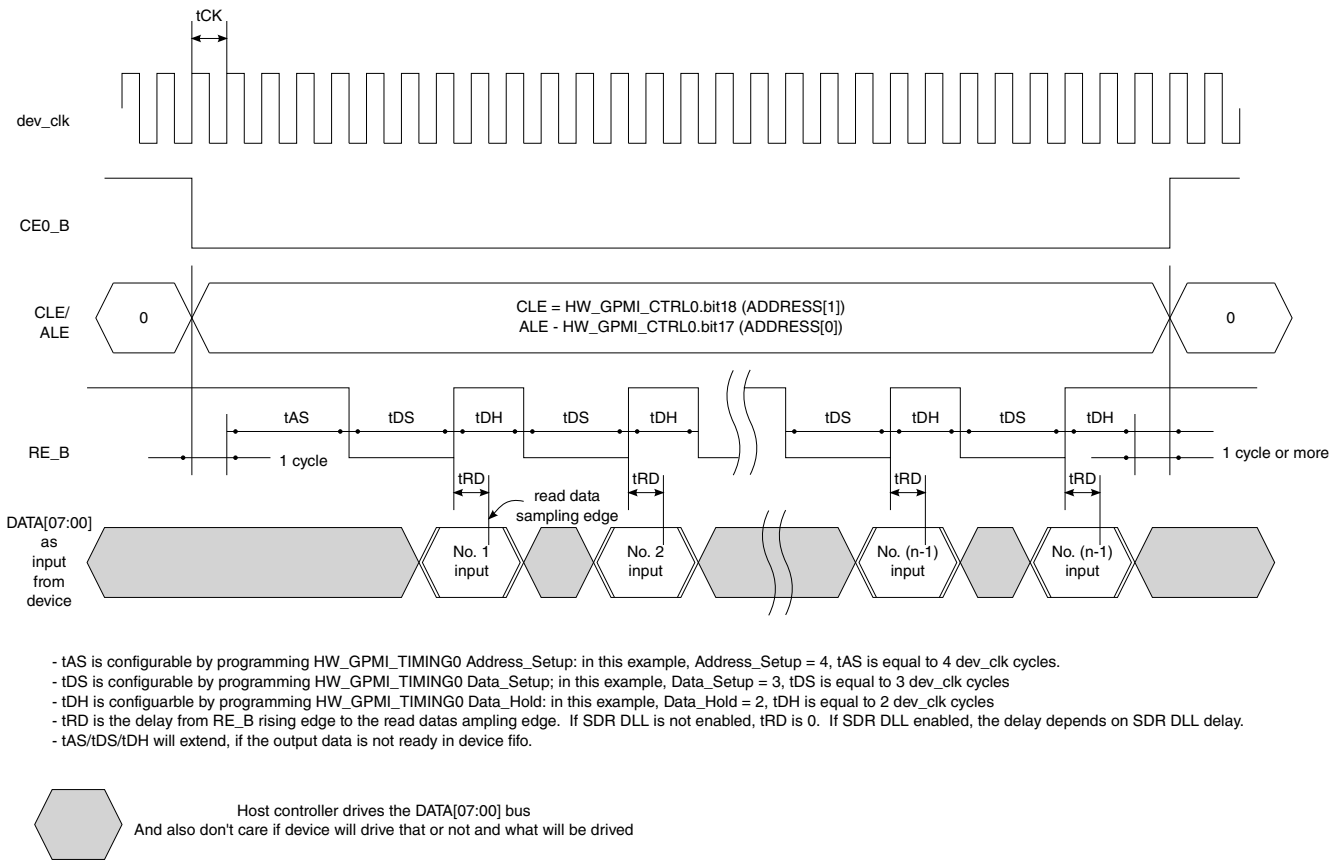


Figure 9-22. ONFI Asynchronous Mode Basic Read Timing Diagram (data read)

9.7.3.3.2 NAND Asynchronous EDO Mode Timing

In high-speed NANDS, the read data may not be valid until after the read strobe (NAND_RE_B) deasserts. This is the case when the minimum tDS is programmed to achieve higher bandwidth.

The GPMI implements a feedback read strobe to sample the read data. The feedback read strobe can be delayed to support fast nand EDO (Extended Data Out) timing where the read strobe may deassert before the read data is valid, and read data is valid for some time after read strobe. Nand EDO timings is applied typically for read cycle frequency above 33 MHz. See Figure 9-23.

The GPMI provides control over the amount of delay applied to the feedback read strobe. This delay depends on the maximum read access time (tREA) of the nand and the read pulse width (tRP) used to access the nand. tRP is specified by GPMI_TIMING0[DATA_SETUP] register. When (tREA + 4ns) is less than tRP, no

delay is required to sample to nand read data. (The 4ns provides adequate data setup time for the GPMI.) In this case set GPMI_CTRL1[HALF_PERIOD] = 0; GPMI_CTRL1[RDN_DELAY] = 0; GPMI_CTRL1[DLL_ENABLE] = 0.

When (tREA + 4ns) is greater than or equal to tRP, a delay of the feedback read strobe is required to sample to nand read data. This delay is equal to the difference between these two timings:

DELAY = tREA + 4ns - tRP.

Since the GPMI delay chain is limited to 6ns maximum, if DELAY > 6ns then increase tRP by increasing the value of GPMI_TIMING0[DATA_SETUP] until DELAY is less than or equal to 6ns.

The GPMI programming for this DELAY depends on the GPMICLK period. The GPMI DLL will not function properly if the GPMICLK period is greater than 12ns: disable the DLL if this is the case. If the GPMICLK period is greater than 6ns (and not greater than 12ns), set the GPMI_CTRL1[HALF_PERIOD]=1; This will cause the DLL reference period (RP) to be one-half of the GPMICLK period. If the GPMICLK period is 6ns or less then set the GPMI_CTRL1[HALF_PERIOD]=0; This will cause the DLL reference period (RP) to be equal to the GPMICLK period. DELAY is a multiple (0 to 1.875) of RP.

The GPMI_CTRL1[RDN_DELAY] is encoded as a 1-bit integer and 3-bit fraction delay factor. DELAY is a multiple of the delay factor and the reference period. See table below for details.

Table 9-13. RDN DELAY

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
0	0.000
1	0.125
2	0.250
3	0.375
4	0.500
5	0.625
6	0.750
7	0.875
8	1.000
9	1.125
10	1.250
11	1.375
12	1.500
13	1.625

Table continues on the next page...

Table 9-13. RDN DELAY (continued)

HW_GPMI_CTRL1[RDN_DELAY]	Delay Factor
14	1.750
15	1.875

DELAY = DelayFactor x RP or DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP.

Use this equation to calculate the value for GPMI_CTRL1[RDN_DELAY]. Then set GPMI_CTRL1[DLL_ENABLE]=1.

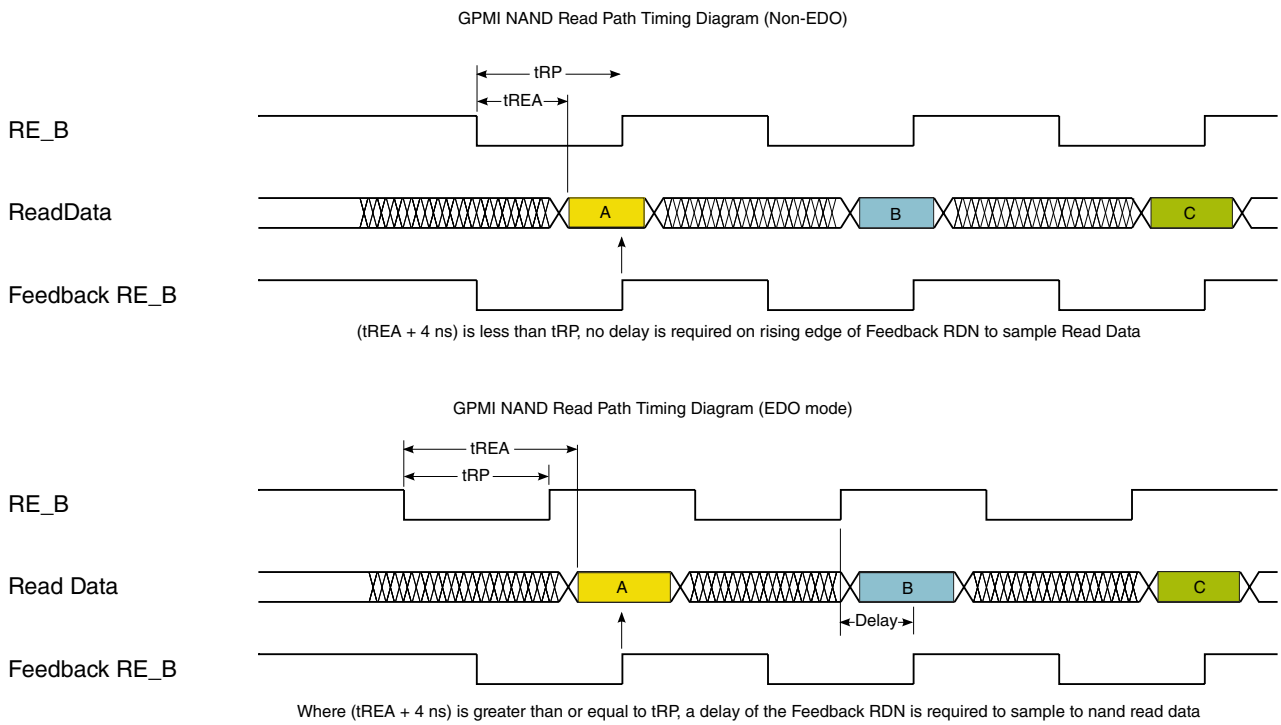


Figure 9-23. NAND Read Path Timing

For example, a NAND with tREAmax = 20 ns, tRPmin = 12 ns, and tRCmin = 25 ns (read cycle time) may be programmed as follows:

- GPMICLK clock frequency: Consider 480/6 = 80 MHz which is 12.5 ns clock period. This is too close to the minimum NAND spec if we program the data setup and hold to 1 GPMICLK cycle. Consider 480/7=68.57 MHz which is 14.58 ns clock period. With data setup and hold set to 1, we have a tRP of 14.58ns and a tRC of 29.16 ns (good margins).
- Since (tREA +4ns) is greater than tRP, required DELAY = tREA + 4ns - tRP = 20 + 4 - 14.58ns = 9.42 ns.

- GPMI_CTRL1[HALF_PERIOD] =, since GPMICLK period is ns. So RP=GPMICLK period = ns.
- DELAY = GPMI_CTRL1[RDN_DELAY] x 0.125 x RP. 9.42 ns = GPMI_CTRL1[RDN_DELAY] x 0.125 x ns. GPMI_CTRL1[RDN_DELAY] =

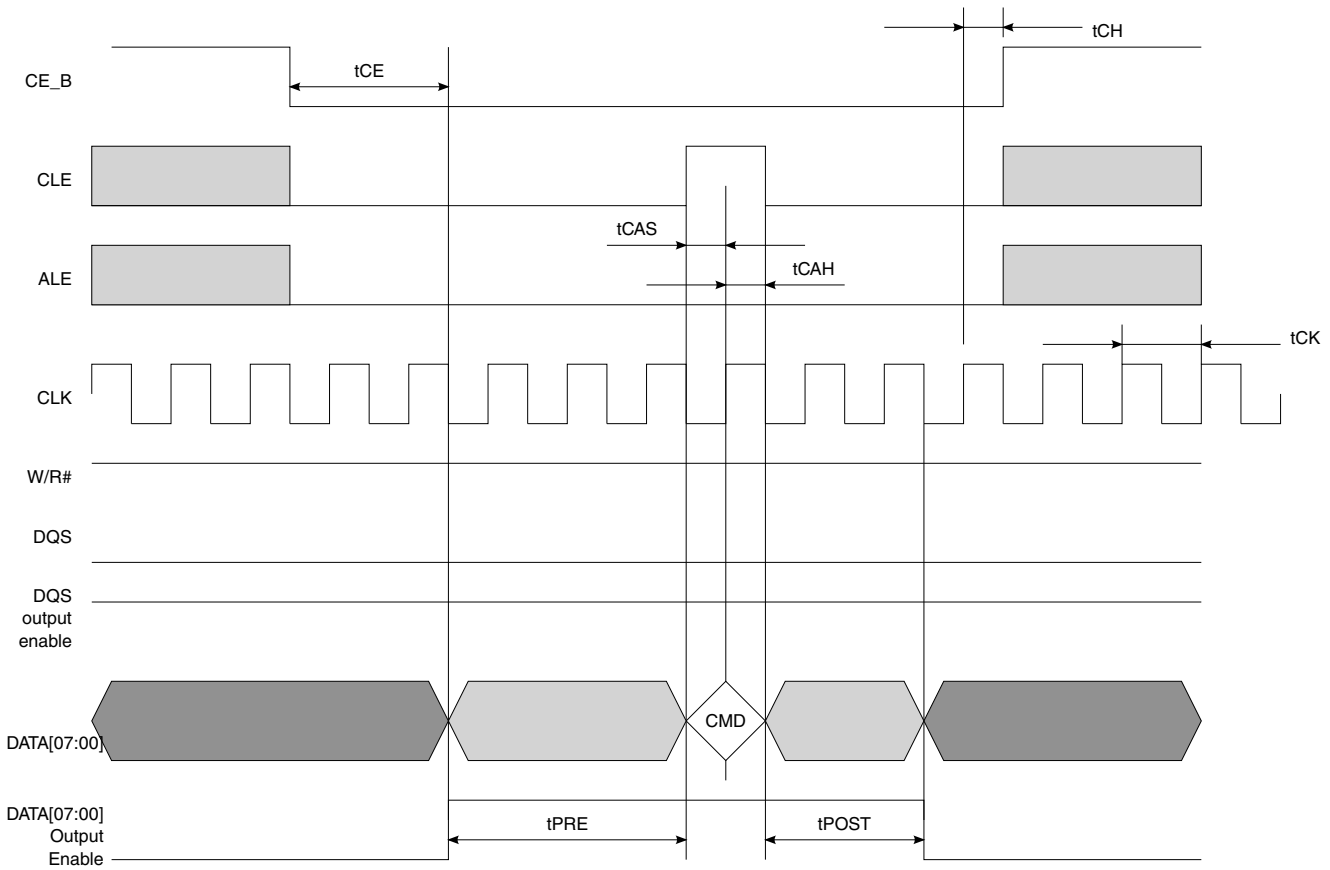
NOTE

It is recommended that the drive strength of NAND_RE_B and NAND_WE_B output pins be set to 8 mA. This will reduce the transition time under heavy loads. Low transition times will be important when NAND interface read and write cycle times are below 30 ns. The other GPMI pins may remain at 4 mA, since their frequency is only up to half that of NAND_RE_B and NAND_WE_B.

9.7.3.3.3 NAND ONFI Source Synchronous Mode Timing

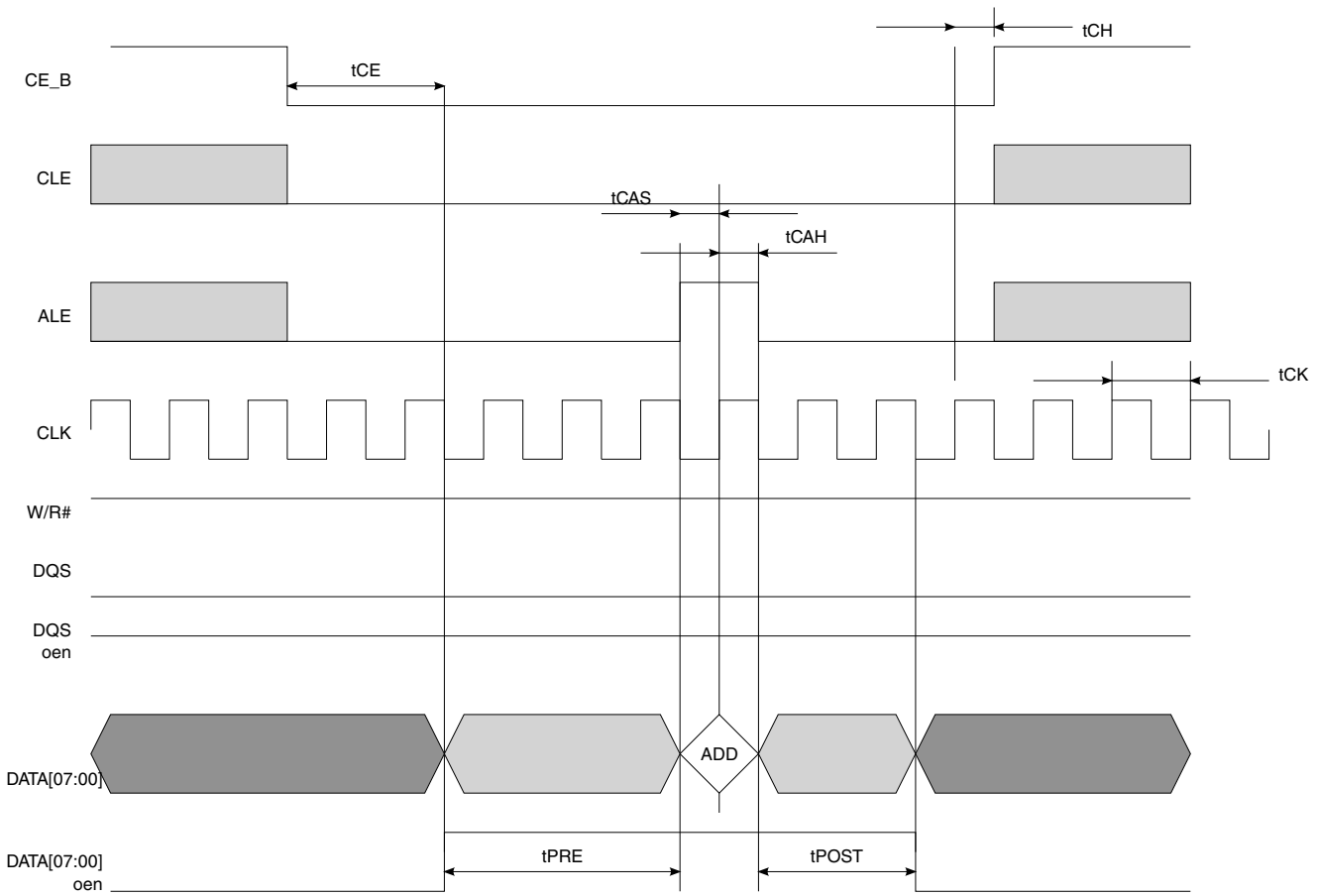
NOTE

In the following figures, CLK shares the same pin as WE_B in Async Mode. And W/R# shares the same pin as RE_B in Async Mode.



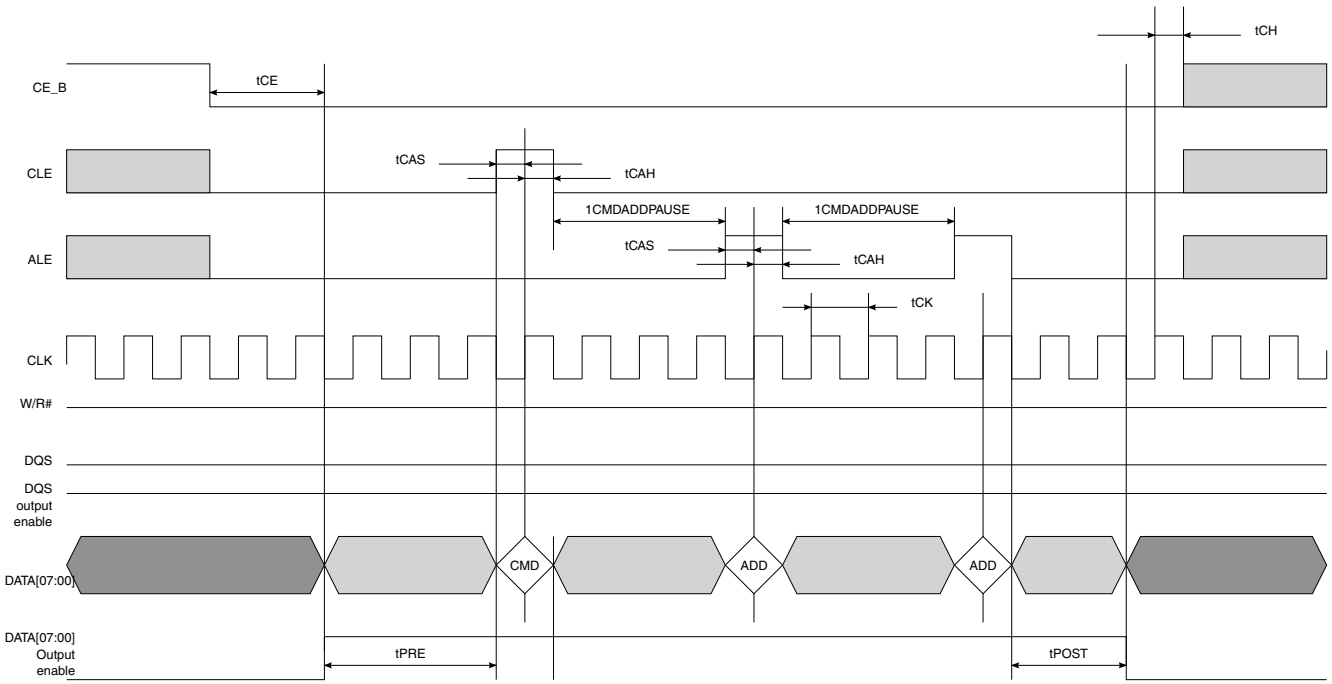
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 9-24. ONFI Source Synchronous Mode Basic Command Write Timing Diagram



tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 * tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK

Figure 9-25. ONFI Source Synchronous Mode Basic Address Write Timing Diagram

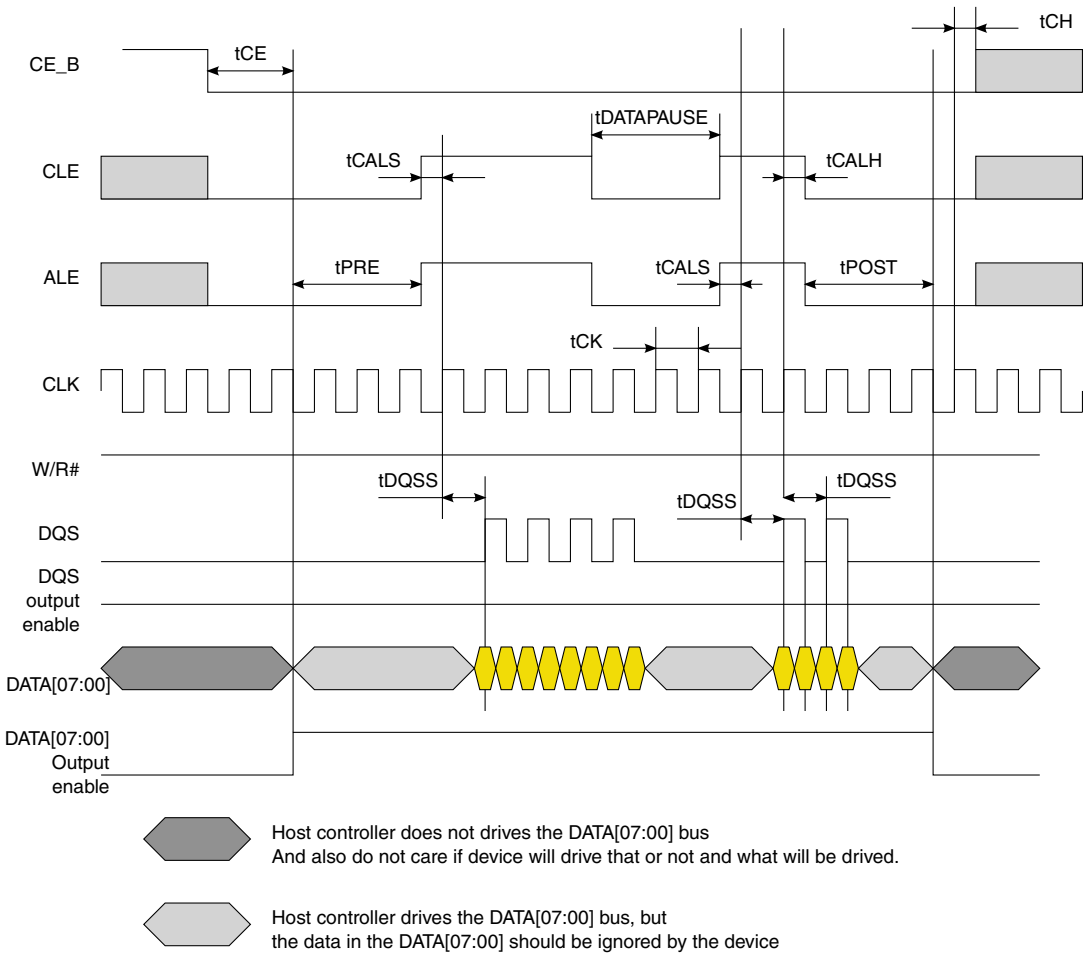


Host controller does not drives the DATA[07:00] bus
And also do not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

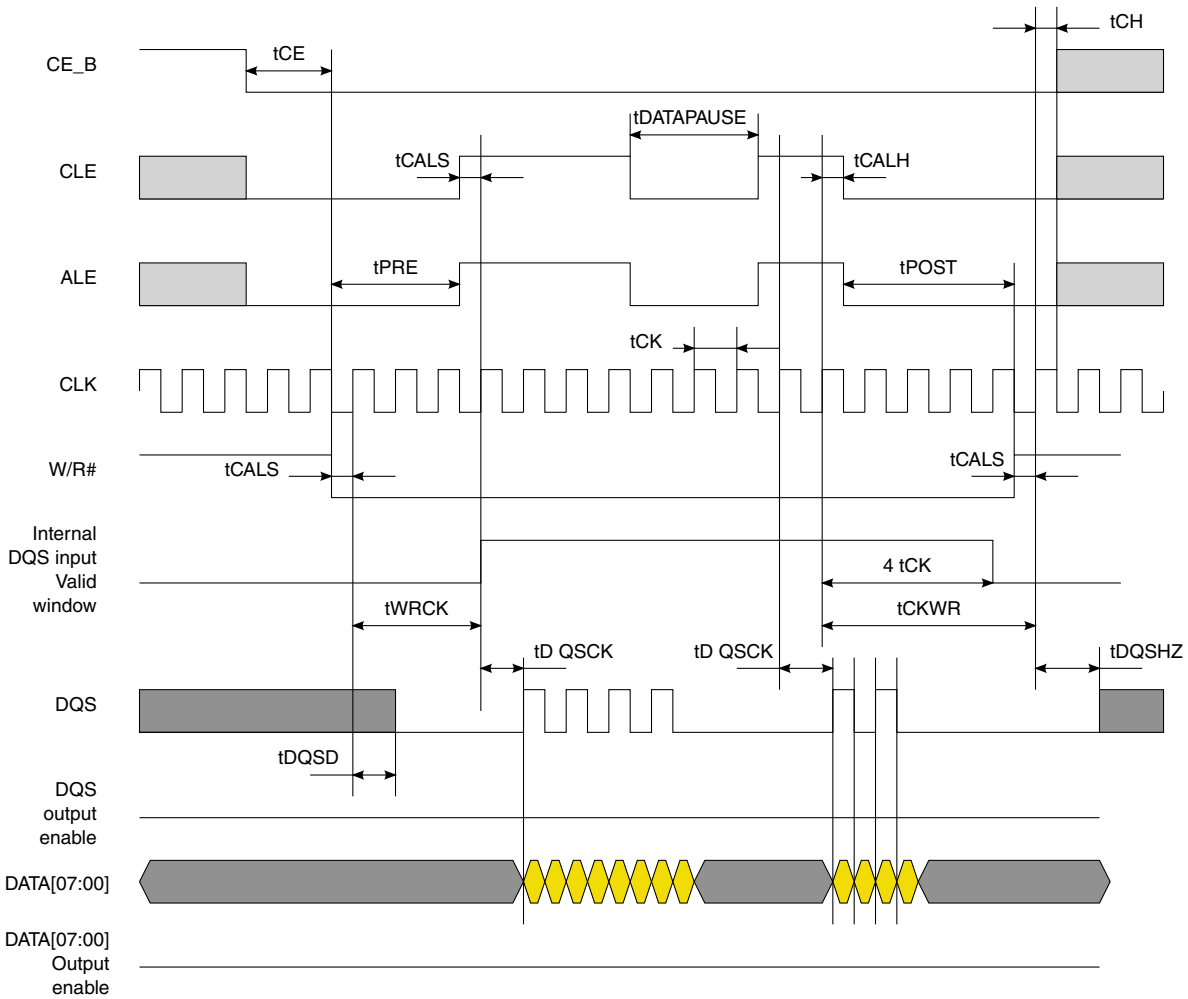
tCK is equal to device clock period
tCAS = 0.5 * tCK
tCAH = 0.5 *tCK
tCE = HW_GPMI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY * tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tCMDADDPAUSE = HW_GPMI_TIMING2.CMDADD_PAUSE * tCK

Figure 9-26. ONFI Source Synchronous Mode Command + Address Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tDQSS = tCK

Figure 9-27. ONFI Source Synchronous Mode Data Write Timing Diagram



tCK is equal to device clock period
tCE = HW_GPI_TIMING2.CE_DELAY * tCK
tCH = 0.5 * tCK
tCALS = 0.5 * tCK
tCALH = 0.5 * tCK
tPRE = HW_GPMI_TIMING2.PREAMBLE_DELAY *tCK
tPOST = HW_GPMI_TIMING2.POST_DELAY * tCK
tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE * tCK
tWRCK/tCKWR/tDQSD/tDASCK/tDASHZ are device parameters

Figure 9-28. ONFI Source Synchronous Mode Data Read Timing Diagram

9.7.3.3.4 NAND Toggle Mode Timing

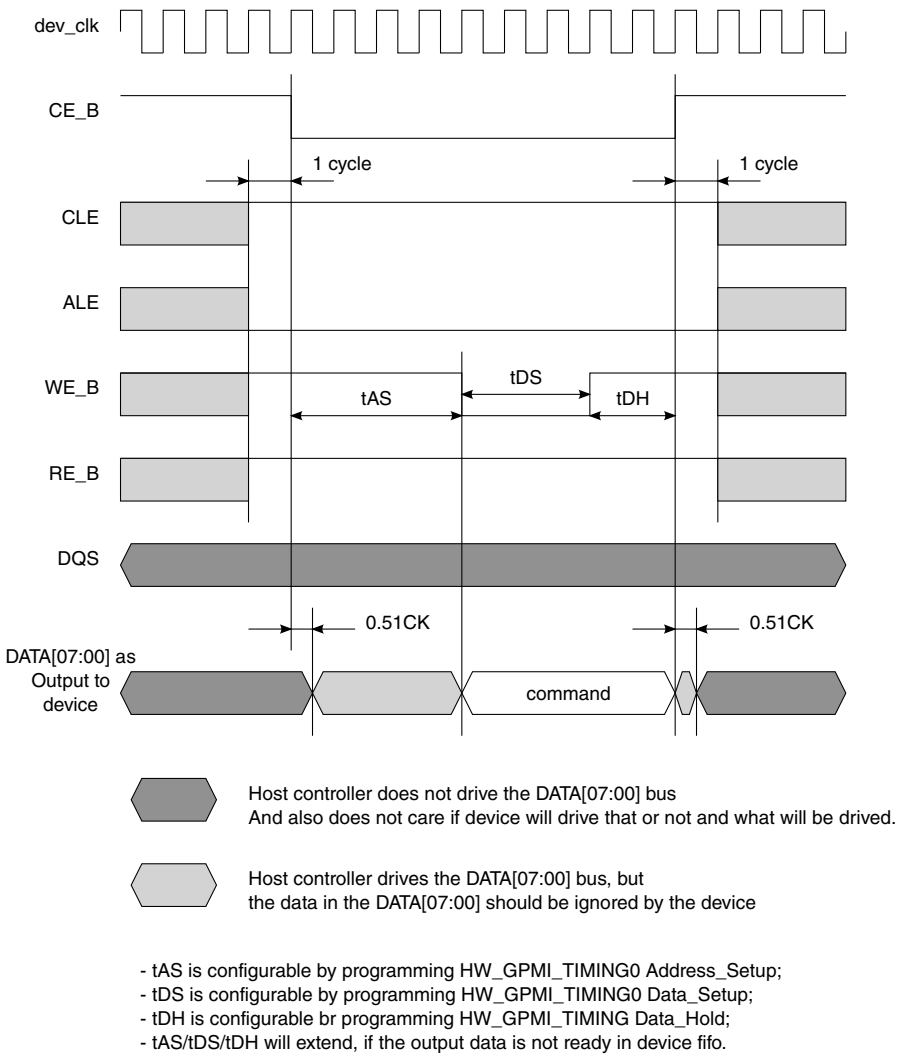
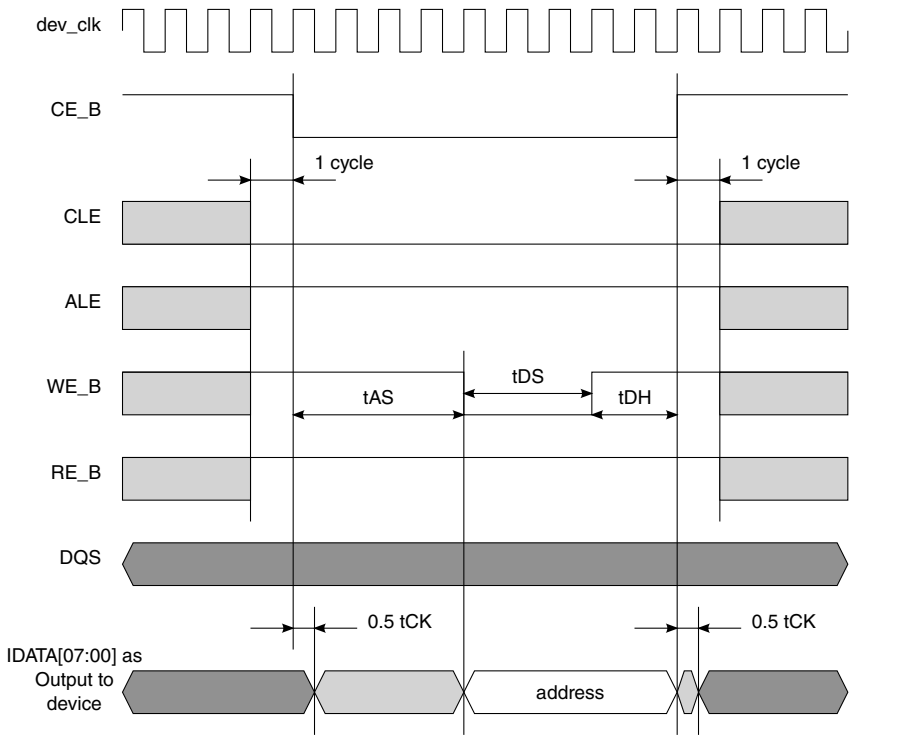


Figure 9-29. Samsung Toggle Mode Basic Command Write Timing Diagram



Host controller does not drive the DATA[07:00] bus
And also does not care if device will drive that or not and what will be drived.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by the device

- tAS is configurable by programming HW_GPMI_TIMING0 Address_Setup;
- tDS is configurable by programming HW_GPMI_TIMING0 Data_Setup;
- tDH is configurable br programming HW_GPMI_TIMING Data_Hold;
- tAS/tDS/tDH will extend, if the output data is not ready in device fifo.

Figure 9-30. Samsung Toggle Mode Basic Address Write Timing Diagram

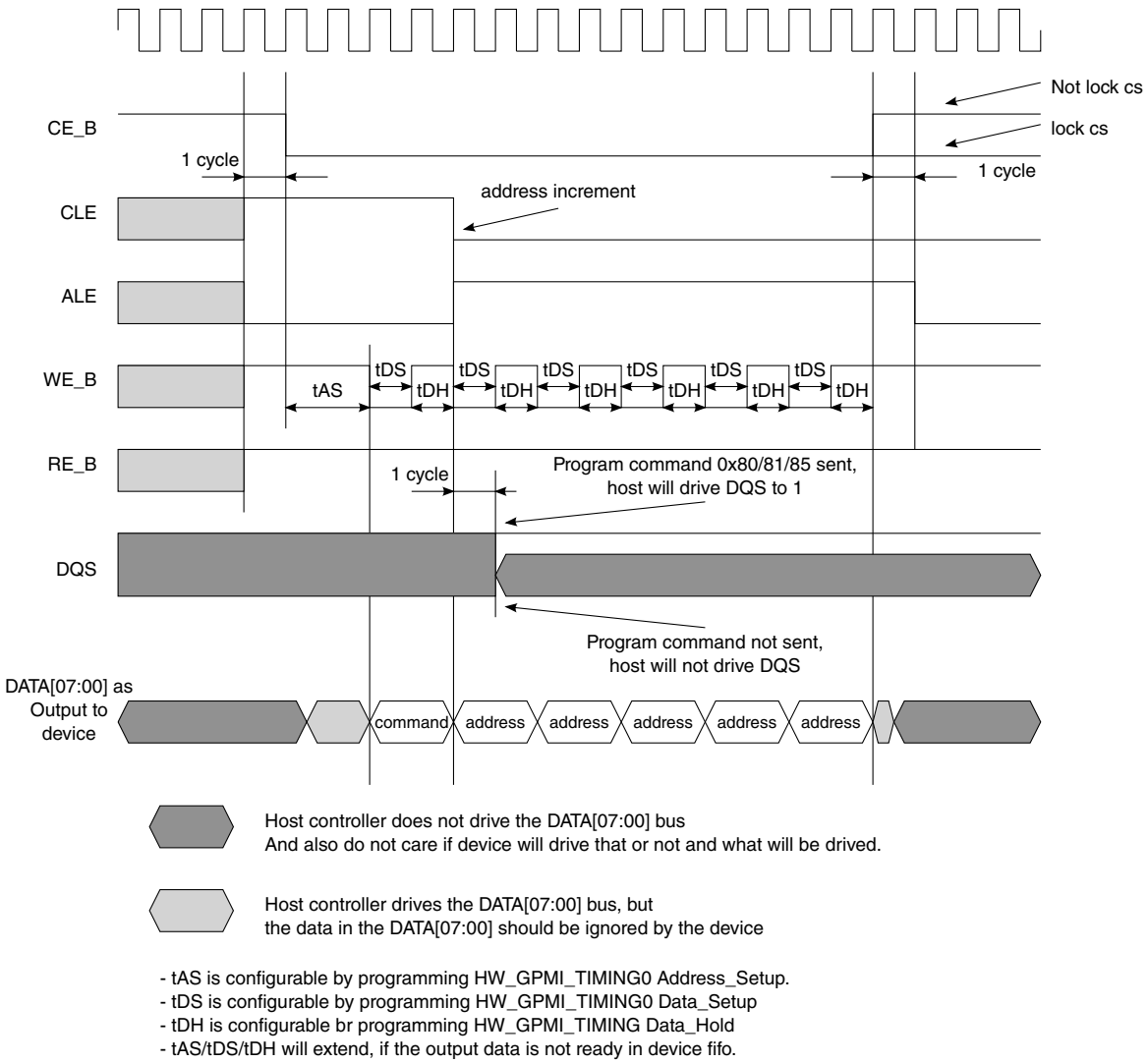


Figure 9-31. Samsung Toggle Mode Basic Command + Address Timing Diagram

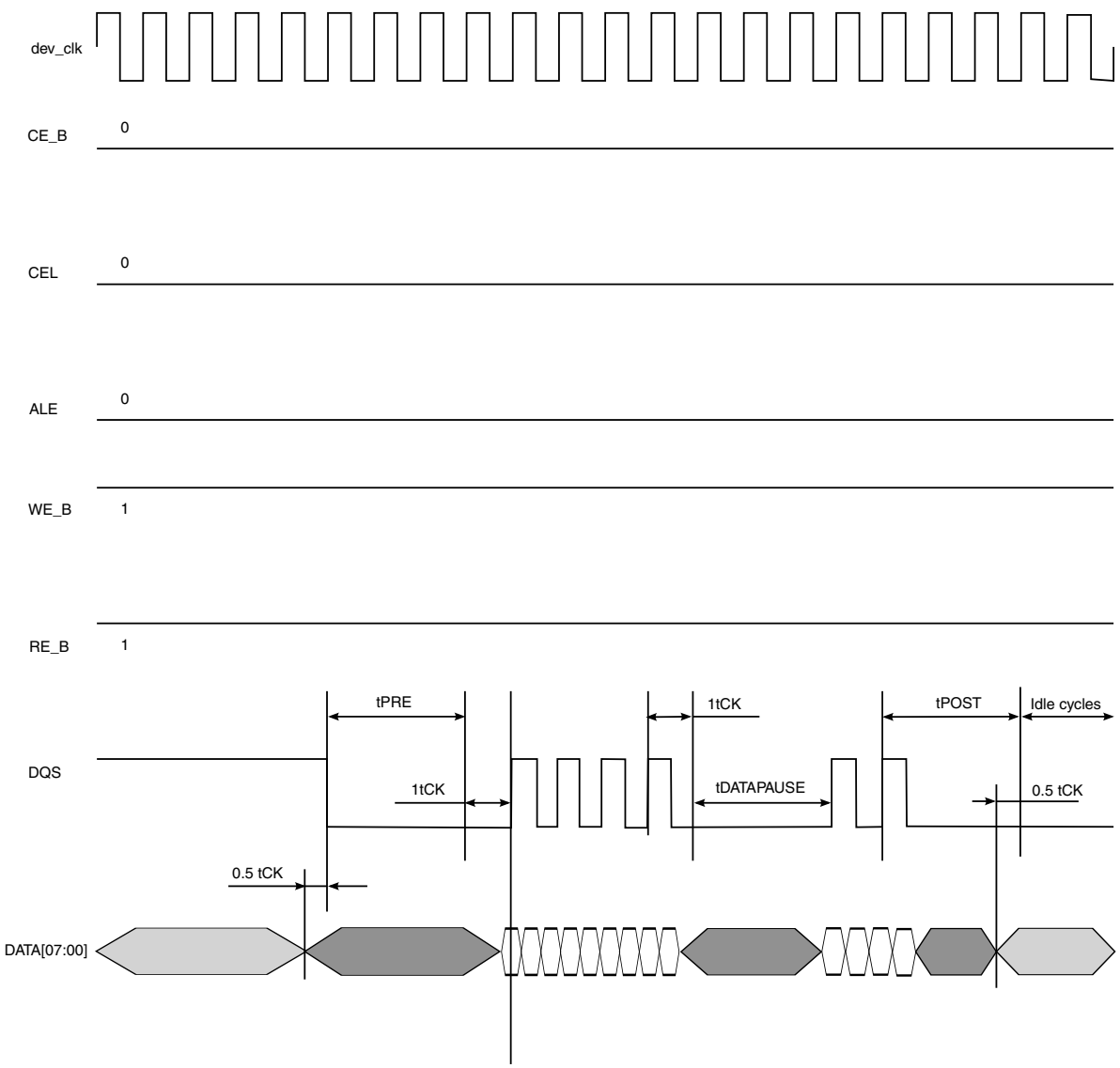
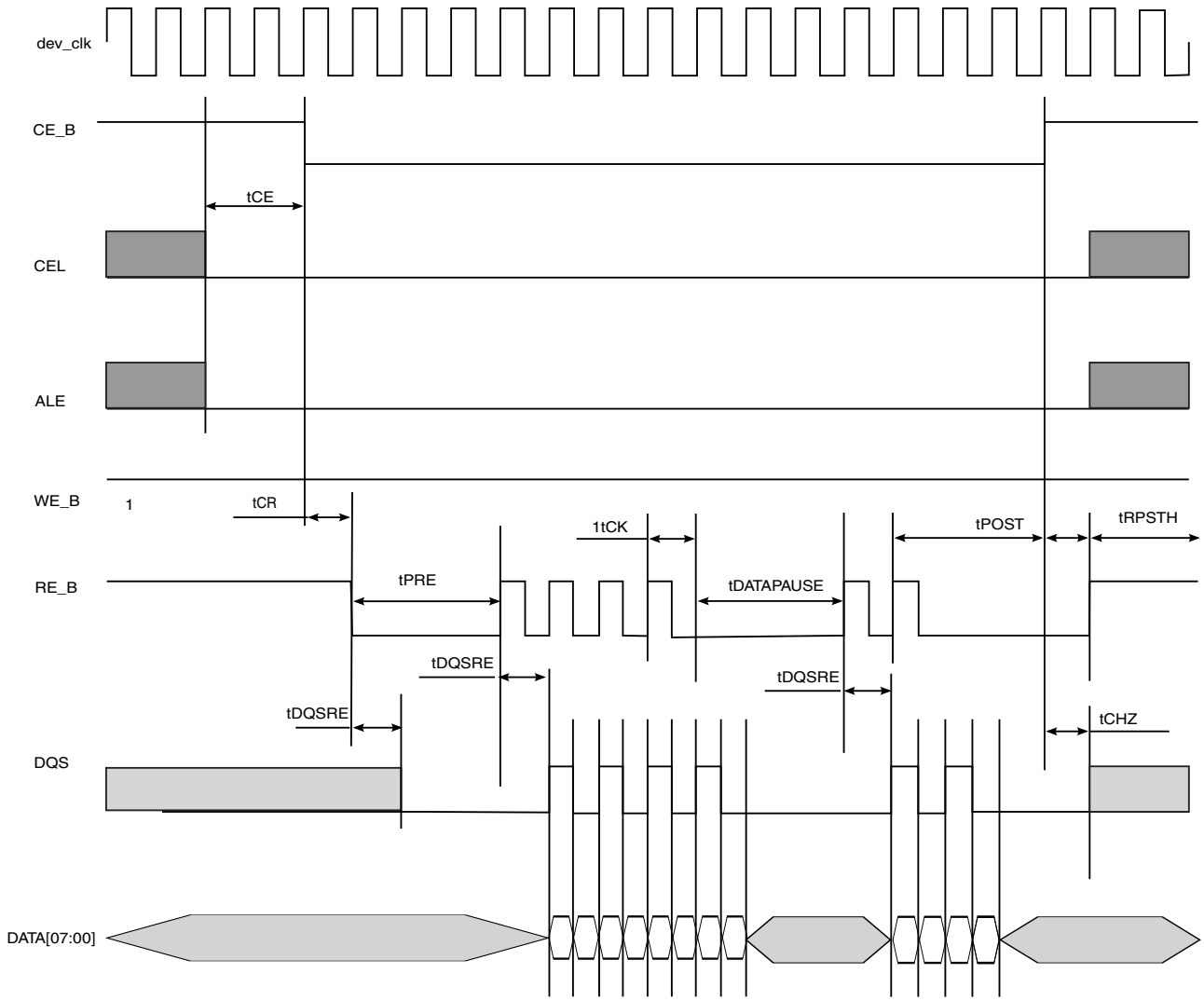


Figure 9-32. Toggle Mode Data Write Timing Diagram



Host controller does not drive the DATA[07:00] bus.
It also does not care if the device will drive or what will be driven.

Host controller drives the DATA[07:00] bus, but
the data in the DATA[07:00] should be ignored by device.

tCK is equal to device clock period

tCE = HW_GPMI_TIMING2.CE_DELAY* tCK

tPRE = (HW_GPMI_TIMING2.PREAMBLE_DELAY - HW_GPMI_TIMING2.TCR) * tCK

tPOST = HW_GPMI_TIMING2.POSTAMBLE_DELAY* tCK

tDATAPAUSE = HW_GPMI_TIMING2.DATA_PAUSE* tCK

tCR = (HW_GPMI_TIMING2.TCR + 1) * tCK

tRPSTH = HW_GPMI_TIMING2.TRPSTH * tCK

Figure 9-33. Toggle Mode Data Read Timing Diagram

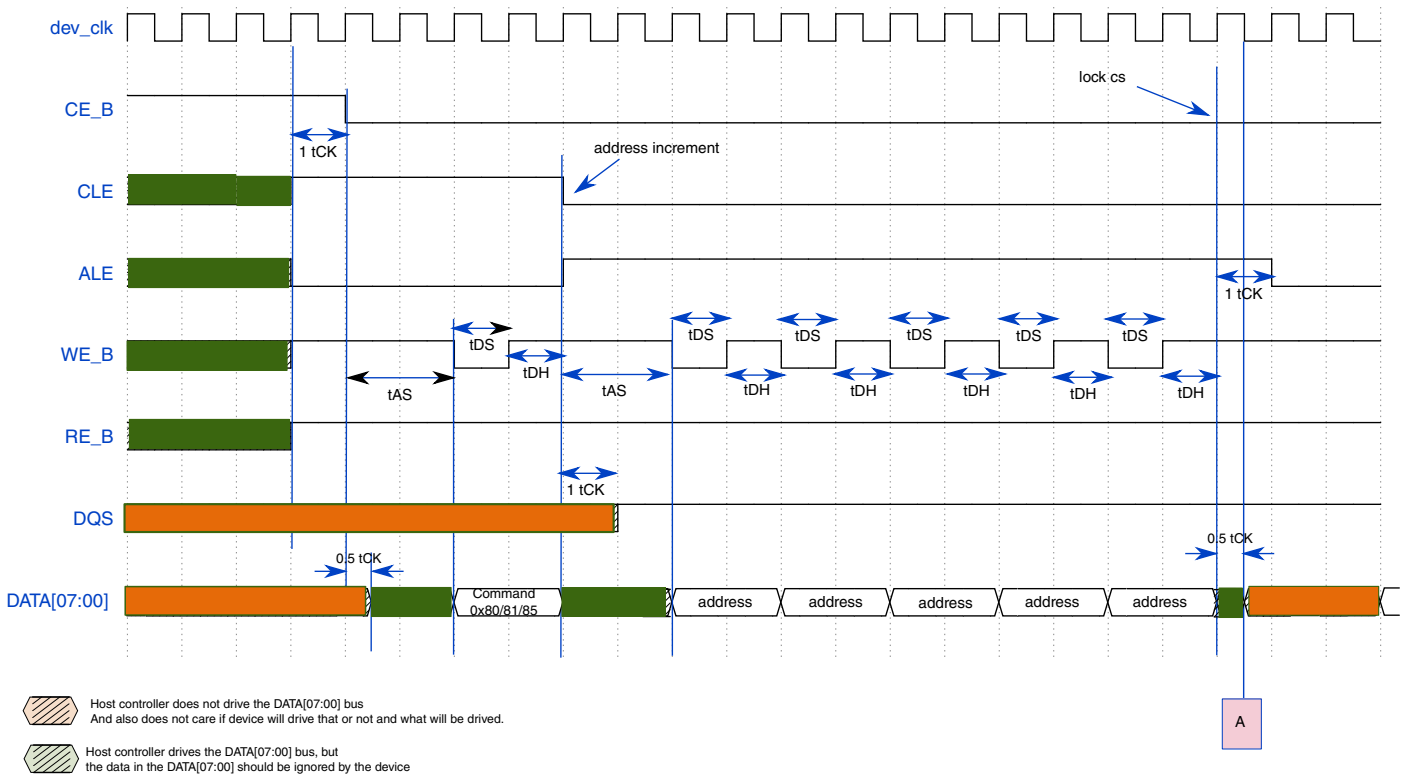


Figure 9-34. Toggle Mode Program Timing Diagram (A)

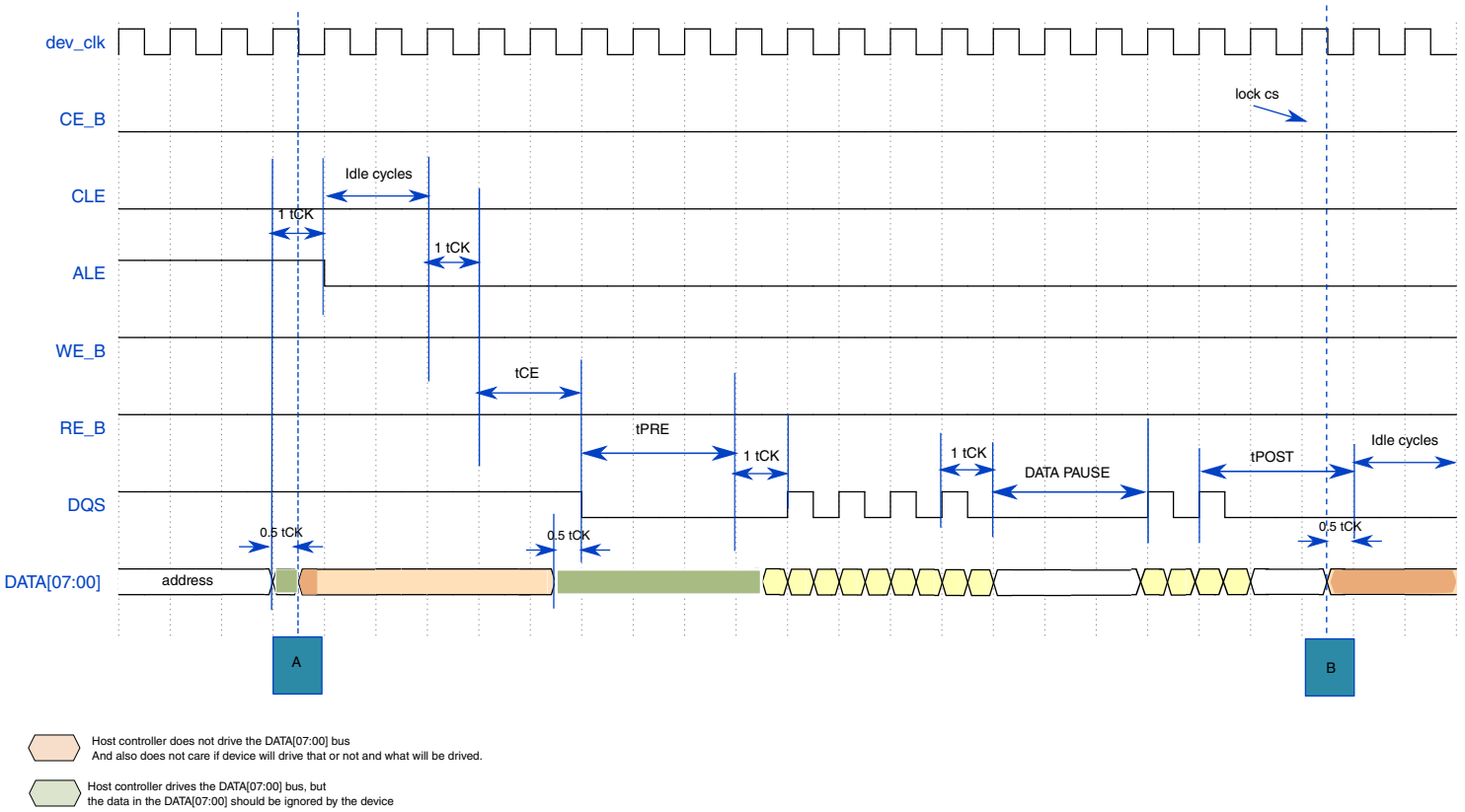


Figure 9-35. Toggle Mode Program Timing Diagram (B)

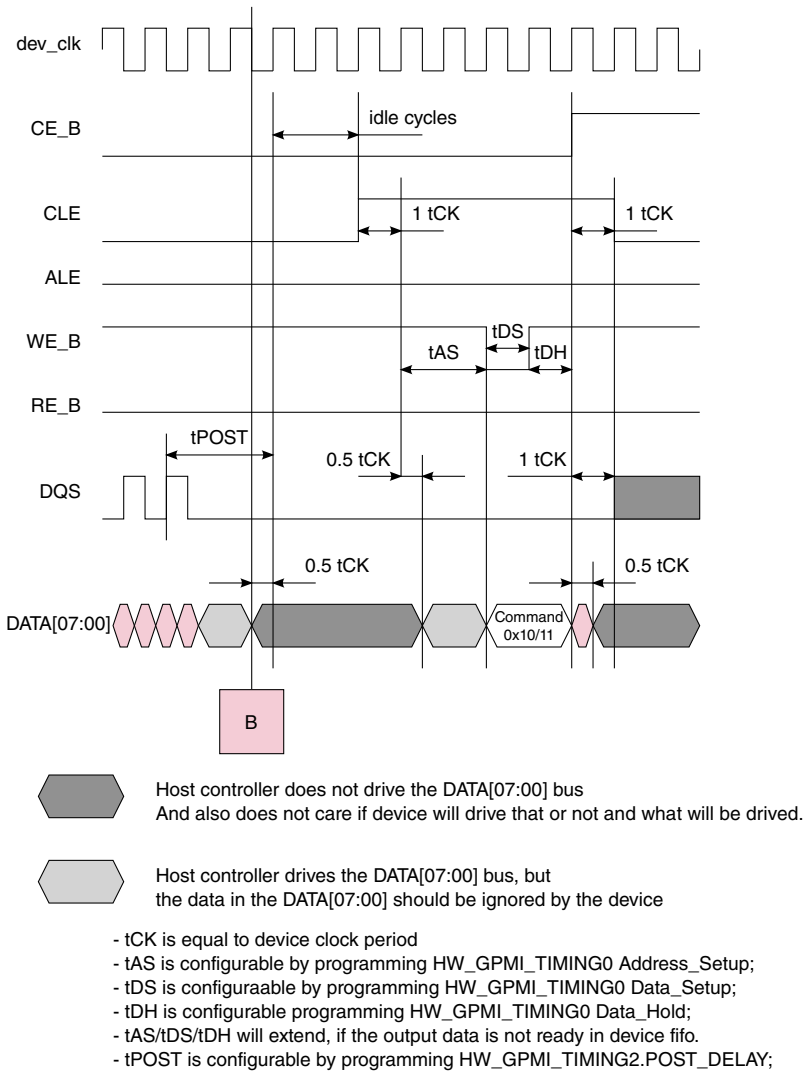


Figure 9-36. Toggle Mode Program Timing Diagram (C)

9.7.3.4 Hardware BCH Interface

The GPMI provides an interface to the BCH module. This reduces the SOC bus traffic and the software involvement.

The GPMI also provides an interface to the Randomizer module. The Randomizer can generate random data based on BCH ECC encoded / decoded data. It can be employed to reduce the disturbances caused by neighboring cells in the NAND chip, thus reducing bit errors. To enable the Randomizer module, set GPMI_ECCCTRL[RANDOMIZER_ENABLE] to 1, then set GPMI_ECCCOUNT[RANDOMIZER_PAGE] to select randomizer page number needed

to be randomized. All these registers can be programmed by the DMA chain. The randomized data should start from the zero column address and be the size of the whole NAND page. If the randomizer function is enabled, GPMI_ECCCTRL[ENABLE_ECC] should also be enabled. To bypass BCH error correction function, set BCH_FLASHxLAYOUT0[ECC0] and BCH_FLASHxLAYOUT1[ECCN] to 0.

When BCH ECC is enable, parity information is inserted on-the-fly during writes to 8-bit NAND devices. The BCH will supply payload and parity to the GPMI to write to the NAND. During NAND reads, parity is checked and ECC processing is performed after each read block. In this case the GPMI reads the NAND device and redirects the data and parity to the BCH module for ECC processing.

To program the BCH for NAND writes, remove the soft reset and clock gates from BCH_CTRL[SFTRST] and BCH_CTRL[CLKGATE]. The bulk of BCH programming is actually applied to the GPMI via PIO operations embedded in its DMA command structures. This has a subtle implication when writing to the GPMI ECC registers: access to the these registers must be written in progressive register order. Thus, to write to the GPMI_ECCCOUNT register, write first (in order) to registers GPMI_CTRL0, GPMI_COMPARE, and GPMI_ECCCTRL before writing to GPMI_ECCCOUNT. These additional register writes need to be accounted for in the CMDWORDS field of the respective DMA channel command register.

NOTE

Note that the GPMI_PAYLOAD and GPMI_AUXILIARY pointers need to be word-aligned for proper ECC operation. If those pointers are non-word-aligned, then the BCH engine will not operate properly and could possibly corrupt system memory in the adjoining memory regions.

9.7.4 Behavior During Reset

A soft reset (SFTRST) can take multiple clock periods to complete, so do NOT set CLKGATE when setting SFTRST. The reset process gates the clocks automatically.

9.7.5 GPMI Memory Map/Register Definition

The following registers provide control for programmable elements of the GPMI module.

NOTE

All ATA or UDMA features are not supported for the chip.

NOTE

GPMI does not support the Set feature command in Toggle mode. The NANDF_DQS output is only enabled in program operation for Toggle mode, but the Set feature command also needs to use the NANDF_DQS signal to write data to Toggle NAND flash. So the Set feature command in Toggle mode is not supported.

GPMI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_2000	GPMI Control Register 0 Description (GPMI_CTRL0)	32	R/W	C000_0000h	9.7.5.1/ 2462
3300_2004	GPMI Control Register 0 Description (GPMI_CTRL0_SET)	32	R/W	C000_0000h	9.7.5.1/ 2462
3300_2008	GPMI Control Register 0 Description (GPMI_CTRL0_CLR)	32	R/W	C000_0000h	9.7.5.1/ 2462
3300_200C	GPMI Control Register 0 Description (GPMI_CTRL0_TOG)	32	R/W	C000_0000h	9.7.5.1/ 2462
3300_2010	GPMI Compare Register Description (GPMI_COMPARE)	32	R/W	0000_0000h	9.7.5.2/ 2464
3300_2020	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL)	32	R/W	0000_0000h	9.7.5.3/ 2465
3300_2024	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_SET)	32	R/W	0000_0000h	9.7.5.3/ 2465
3300_2028	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_CLR)	32	R/W	0000_0000h	9.7.5.3/ 2465
3300_202C	GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL_TOG)	32	R/W	0000_0000h	9.7.5.3/ 2465
3300_2030	GPMI Integrated ECC Transfer Count Register Description (GPMI_ECCCOUNT)	32	R/W	0000_0000h	9.7.5.4/ 2466
3300_2040	GPMI Payload Address Register Description (GPMI_PAYLOAD)	32	R/W	0000_0000h	9.7.5.5/ 2467
3300_2050	GPMI Auxiliary Address Register Description (GPMI_AUXILIARY)	32	R/W	0000_0000h	9.7.5.6/ 2467

Table continues on the next page...

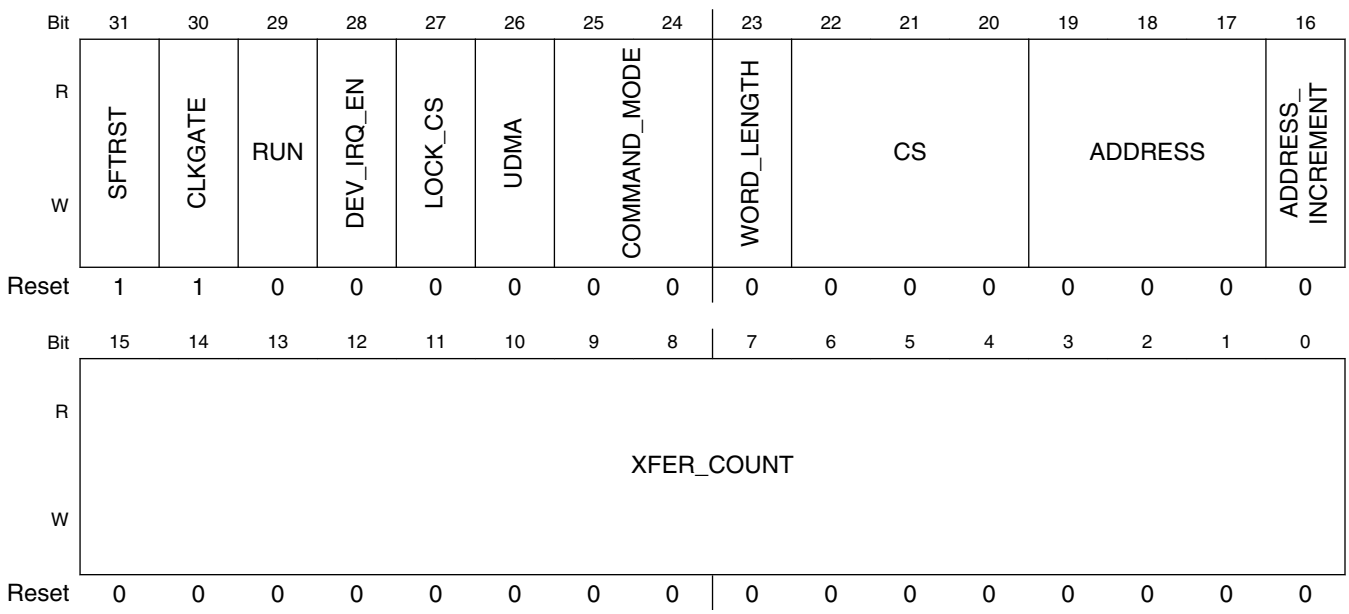
GPMI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3300_2060	GPMI Control Register 1 Description (GPMI_CTRL1)	32	R/W	0004_0004h	9.7.5.7/ 2468
3300_2064	GPMI Control Register 1 Description (GPMI_CTRL1_SET)	32	R/W	0004_0004h	9.7.5.7/ 2468
3300_2068	GPMI Control Register 1 Description (GPMI_CTRL1_CLR)	32	R/W	0004_0004h	9.7.5.7/ 2468
3300_206C	GPMI Control Register 1 Description (GPMI_CTRL1_TOG)	32	R/W	0004_0004h	9.7.5.7/ 2468
3300_2070	GPMI Timing Register 0 Description (GPMI_TIMING0)	32	R/W	0001_0203h	9.7.5.8/ 2471
3300_2080	GPMI Timing Register 1 Description (GPMI_TIMING1)	32	R/W	0000_0000h	9.7.5.9/ 2471
3300_2090	GPMI Timing Register 2 Description (GPMI_TIMING2)	32	R/W	0302_3336h	9.7.5.10/ 2472
3300_20A0	GPMI DMA Data Transfer Register Description (GPMI_DATA)	32	R/W	0000_0000h	9.7.5.11/ 2473
3300_20B0	GPMI Status Register Description (GPMI_STAT)	32	R	0000_0005h	9.7.5.12/ 2473
3300_20C0	GPMI Debug Information Register Description (GPMI_DEBUG)	32	R	0000_0000h	9.7.5.13/ 2476
3300_20D0	GPMI Version Register Description (GPMI_VERSION)	32	R	0502_0000h	9.7.5.14/ 2477
3300_20E0	GPMI Debug2 Information Register Description (GPMI_DEBUG2)	32	R/W	0000_F100h	9.7.5.15/ 2477
3300_20F0	GPMI Debug3 Information Register Description (GPMI_DEBUG3)	32	R	0000_0000h	9.7.5.16/ 2480
3300_2100	GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)	32	R/W	0000_0038h	9.7.5.17/ 2481
3300_2110	GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL)	32	R/W	0000_0038h	9.7.5.18/ 2482
3300_2120	GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)	32	R	0000_0000h	9.7.5.19/ 2484
3300_2130	GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)	32	R	0000_0000h	9.7.5.20/ 2485

9.7.5.1 GPMI Control Register 0 Description (GPMI_CTRL0*n*)

The GPMI control register 0 specifies the GPMI transaction to perform for the current command chain item.

Address: 3300_2000h base + 0h offset + (4d × i), where i=0d to 3d



GPMI_CTRL0*n* field descriptions

Field	Description
31 SFTRST	Set to zero for normal operation. When this bit is set to one (default), then the entire block is held in its reset state. This will not work if the CLKGATE bit is already set to '1'. CLKGATE must be cleared to '0' before issuing a soft reset. Also the GPMICLK must be running for this to work properly. RUN = 0x0 Allow GPMI to operate normally. RESET = 0x1 Hold GPMI in reset.
30 CLKGATE	Set this bit zero for normal operation. Setting this bit to one (default), gates all of the block level clocks off for miniminizing AC energy consumption. RUN = 0x0 Allow GPMI to operate normally. NO_CLKS = 0x1 Do not clock GPMI gates in order to minimize power consumption.
29 RUN	The GPMI is busy running a command whenever this bit is set to '1'. The GPMI is idle whenever this bit set to zero. This can be set to one by a CPU write. In addition, the DMA sets this bit each time a DMA command has finished its PIO transfer phase. IDLE = 0x0 The GPMI is idle. BUSY = 0x1 The GPMI is busy running a command.

Table continues on the next page...

GPMI_CTRL0*n* field descriptions (continued)

Field	Description
28 DEV_IRQ_EN	When set to '1' and ATA_IRQ pin is asserted, the GPMI_IRQ output will assert.
27 LOCK_CS	For ATA/NAND mode: 0= Deassert chip select (CS) after RUN is complete. 1= Continue to assert chip select (CS) after RUN is complete. For Camera Mode: 0= Dont wait for VSYNC rising edge before capturing data. 1= Wait for VSYNC rising edge before capturing data (Camera mode only). DISABLED = 0x0 Deassert chip select (CS) after RUN is complete. ENABLED = 0x1 Continue to assert chip select (CS) after RUN is complete.
26 UDMA	DISABLED = 0x0 Use ATA-PIO mode on the external bus. ENABLED = 0x1 Use ATA-Ultra DMA mode on the external bus. 0 Use ATA-PIO mode on the external bus. 1 Use ATA-Ultra DMA mode on the external bus.
25–24 COMMAND_ MODE	WRITE = 0x0 Write mode. READ = 0x1 Read mode. READ_AND_COMPARE = 0x2 Read and Compare mode (setting sense flop). WAIT_FOR_READY = 0x3 Wait for Ready mode. For ATA WAIT_FOR_READY command set CS=01. 00 Write mode. 01 Read Mode. 10 Read and Compare Mode (setting sense flop). 11 Wait for Ready.
23 WORD_LENGTH	This bit should only be changed when RUN==0. Reserve = 0x0 Reserved. 8_BIT = 0x1 8-bit Data Bus mode. 0 Reserved. 1 8-bit Data Bus mode.
22–20 CS	Selects which chip select is active for this command. For ATA WAIT_FOR_READY command, this must be set to b01.
19–17 ADDRESS	Specifies the three address lines for ATA mode. In NAND mode, use A0 for CLE and A1 for ALE. NAND_DATA = 0x0 In NAND mode, this address is used to read and write data bytes. NAND_CLE = 0x1 In NAND mode, this address is used to write command bytes. NAND_ALE = 0x2 In NAND mode, this address is used to write address bytes.
16 ADDRESS_ INCREMENT	In ATA mode, the address will increment with each cycle. In NAND mode, the address will increment once, after the first cycle (going from CLE to ALE). DISABLED = 0x0 Address does not increment. ENABLED = 0x1 Increment address. 0 Address does not increment. 1 Increment address.
XFER_COUNT	Number of bytes to transfer for this command. A value of zero will transfer 64K bytes.

9.7.5.2 GPMI Compare Register Description (GPMI_COMPARE)

The GPMI compare register specifies the expect data and the xor mask for comparing to the status values read from the device. This register is used by the Read and Compare command.

GPMI_COMPARE 0x010

Address: 3300_2000h base + 10h offset = 3300_2010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK																REFERENCE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

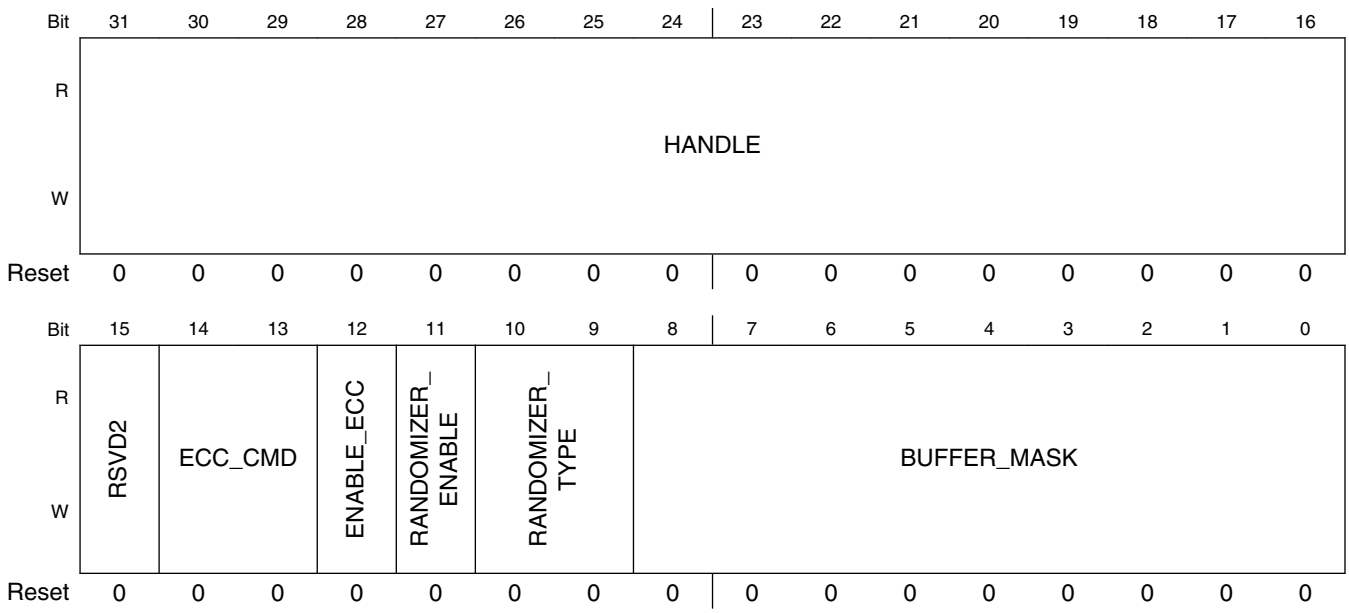
GPMI_COMPARE field descriptions

Field	Description
31–16 MASK	16-bit mask which is applied after the read data is XORed with the REFERENCE bit field.
REFERENCE	16-bit value which is XORed with data read from the NAND device.

9.7.5.3 GPMI Integrated ECC Control Register Description (GPMI_ECCCTRL*n*)

The GPMI ECC control register handles configuration of the integrated ECC / Randomizer accelerator.

Address: 3300_2000h base + 20h offset + (4d × i), where i=0d to 3d



GPMI_ECCCTRL*n* field descriptions

Field	Description
31–16 HANDLE	This is a register available to software to attach an identifier to a transaction in progress. This handle will be available from the ECC register space when the completion interrupt occurs.
15 RSVD2	Always write zeroes to this bit field.
14–13 ECC_CMD	ECC Command information. DECODE = 0x0 Decode. ENCODE = 0x1 Encode. RESERVE2 = 0x2 Reserved. RESERVE3 = 0x3 Reserved.
12 ENABLE_ECC	Enable ECC processing of GPMI transfers. ENABLE = 0x1 Use integrated ECC for read and write transfers. DISABLE = 0x0 Integrated ECC remains in idle.
11 RANDOMIZER_ ENABLE	Enable randomizer function. If this bit is set to enable, ENABLE_ECC should be also enable.

Table continues on the next page...

GPMI_ECCCTRL*n* field descriptions (continued)

Field	Description
	0x0 disable 0x1 enable
10–9 RANDOMIZER_TYPE	Set randomizer type 0x00 Type 0 0x01 Type 1 0x10 Type 2
BUFFER_MASK	ECC buffer information. The BCH error correction only allows two configurations of the buffer mask - software may either read just the first block on the flash page or the entire flash page. Write operations must be for the entire flash page. Invalid buffer mask values will cause the DMA descriptor command to be terminated. BCH_AUXONLY = 0x100 Set to request transfer from only the auxiliary buffer (block 0 on flash). BCH_PAGE = 0x1FF Set to request transfer to/from the entire page.

9.7.5.4 GPMI Integrated ECC Transfer Count Register Description (GPMI_ECCCOUNT)

The GPMI ECC Transfer Count Register contains the count of bytes that flow through the ECC / Randomizer subsystem.

GPMI_ECCCOUNT 0x030

Address: 3300_2000h base + 30h offset = 3300_2030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RANDOMIZER_PAGE								COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_ECCCOUNT field descriptions

Field	Description
31–24 -	Always write zeroes to this bit field.
23–16 RANDOMIZER_PAGE	Set NAND page number needed to be randomized. The value is between 0-255
COUNT	Number of bytes to pass through ECC. This is the GPMI transfer count plus the syndrome count that will be inserted into the stream by the ECC. In DMA2ECC_MODE this count must match the GPMI_CTRL0_XFER_COUNT. A value of zero will transfer 64K words.

9.7.5.5 GPMI Payload Address Register Description (GPMI_PAYLOAD)

The GPMI payload address register specifies the location of the data buffers in system memory. This value must be word aligned.

GPMI_PAYLOAD 0x040

Address: 3300_2000h base + 40h offset = 3300_2040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDRESS														RSVD0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_PAYLOAD field descriptions

Field	Description
31–2 ADDRESS	Pointer to an array of one or more 512 byte payload buffers.
RSVD0	Always write zeroes to this bit field.

9.7.5.6 GPMI Auxiliary Address Register Description (GPMI_AUXILIARY)

The GPMI auxiliary address register specifies the location of the auxiliary buffers in system memory. This value must be word aligned.

GPMI_AUXILIARY 0x050

Address: 3300_2000h base + 50h offset = 3300_2050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ADDRESS														RSVD0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_AUXILIARY field descriptions

Field	Description
31–2 ADDRESS	Pointer to ECC control structure and meta-data storage.
RSVD0	Always write zeroes to this bit field.

9.7.5.7 GPMI Control Register 1 Description (GPMI_CTRL1*n*)

The GPMI control register 1 specifies additional control fields that are not used on a per-transaction basis.

Address: 3300_2000h base + 60h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEV_CLK_STOP	SSYNC_CLK_STOP	WRITE_CLK_STOP	TOGGLE_MODE	GPMI_CLK_DIV2_EN	UPDATE_CS	SSYNCMODE	DECOUPLE_CS	WRN_DLY_SEL		TEST_TRIGGER	TIMEOUT_IRQ_EN	GANGED_RDYBUSY	BCH_MODE	DLL_ENABLE	HALF_PERIOD
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDN_DELAY				DMA2ECC_MODE	DEV_IRQ	TIMEOUT_IRQ	BURST_EN	ABORT_WAIT_REQUEST	ABORT_WAIT_FOR_READY_CHANNEL			DEV_RESET	ATA_IRQRDY_POLARITY	CAMERA_MODE	GPMI_MODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

GPMI_CTRL1*n* field descriptions

Field	Description
31 DEV_CLK_STOP	set this bit to 1 will stop gpmi io working clk.
30 SSYNC_CLK_STOP	set this bit to 1 will stop the source synchronous mode clk.

Table continues on the next page...

GPMI_CTRL1*n* field descriptions (continued)

Field	Description
29 WRITE_CLK_STOP	In onfi source synchronous mode, host may save power during the data write cycles by holding the CLK signal high (i.e. stopping the CLK). The host may only stop the CLK during data write, by setting this bit to 1, if the device supports this feature as indicated in the parameter page.
28 TOGGLE_MODE	enable samsung toggle mode.
27 GPMI_CLK_DIV2_EN	This bit should be reset to 0 in asynchronous mode. The frequence ratio of (device clock : ccm gpmi clock) will be (1 : 1). This bit should be set to 1, in source synchronous mode or toggle mode. The frequence raito of (device clock : ccm gpmi clock) will be (1 : 2). enable the gpmi clk divider. 0x0 internal factor-2 clock divider is disabled 0x1 internal factor-2 clock divider is enabled.
26 UPDATE_CS	force the CS value is be updated to external chip select pin, even GPMI is idle.
25 SSYNCMODE	source synchronouse mode 1 or asynchrous mode 0. ASYNC = 0x0 Asynchronous mode. SSYNC = 0x1 Source Synchronous mode.
24 DECOUPLE_CS	Decouple Chip Select from DMA Channel. Setting this bit to 1 will allow a DMA channel to specify any value in the CTRL0_CS register field. Software can use one DMA channel to access all 8 Nand devices.
23–22 WRN_DLY_SEL	Since the GPMI write strobe (WRN) is a fast clock pin, the delay on this signal can be programmed to match the load on this pin. 0 = ~2ns; 1 = ~4ns; 2 = ~6ns; 3 = no delay.
21 TEST_TRIGGER	Test Trigger Enable 0 Disable 1 Enable
20 TIMEOUT_IRQ_EN	Setting this bit to '1' will enable timeout IRQ for transfers in ATA mode only, and for WAIT_FOR_READY commands in both ATA and Nand mode. The Device_Busy_Timeout value is used for this timeout.
19 GANGED_RDYBUSY	Set this bit to 1 will force all Nand RDY_BUSY inputs to be sourced from (tied to) RDY_BUSY0. This will free up all, except one, RDY_BUSY input pins.
18 BCH_MODE	This bit selects which error correction unit will access GPMI. This bit must always be set to '1', since only the BCH unit is available in this design.
17 DLL_ENABLE	Set this bit to 1 to enable the GPMI DLL. This is required for fast NAND reads (above 30 MHz read strobe). After setting this bit, wait 64 GPMI clock cycles for the DLL to lock before performing a NAND read.
16 HALF_PERIOD	Set this bit to 1 if the GPMI clock period is greater than 16ns for proper DLL operation. DLL_ENABLE must be zero while changing this field.
15–12 RDN_DELAY	This variable is a factor in the calculated delay to apply to the internal read strobe for correct read data sampling. The applied delay (AD) is between 0 and 1.875 times the reference period (RP). RP is one half of the GPMI clock period if HALF_PERIOD=1 otherwise it is the full GPMI clock period. The equation is: AD = RDN_DELAY x 0.125 x RP. This value must not exceed 16ns.

Table continues on the next page...

GPMI_CTRL1*n* field descriptions (continued)

Field	Description
	This variable is used to achieve faster NAND access. For example if the Read Strobe is asserted from time 0 to 13ns but the read access time is 20ns, then choose AD=12ns will cause the data to be sampled at time 25ns (13+12) giving a 5ns data setup time. If RP=13ns then RDN_DELAY = 12/(0.125 x 13ns) = 7.38 (0111b). DLL_ENABLE must be zero while changing this field.
11 DMA2ECC_MODE	This is mainly for testing HWECC without involving the Nand device. Setting this bit will cause DMA write data to redirected to HWECC module (instead of Nand Device) for encoding or decoding.
10 DEV_IRQ	This bit is set when an Interrupt is received from the ATA device. Write 0 to clear.
9 TIMEOUT_IRQ	This bit is set when a timeout occurs using the Device_Busy_Timeout value. Write 0 to clear.
8 BURST_EN	When set to 1 each DMA request will generate a 4-transfer burst on the APB bus.
7 ABORT_WAIT_REQUEST	Request to abort "wait for ready" command on channel indicated by ABORT_WAIT_FOR_READY_CHANNEL. Hardware will clear this bit when abort is done.
6–4 ABORT_WAIT_FOR_READY_CHANNEL	Abort a wait for ready command on selected channel. Set the ABORT_WAIT_REQUEST to kick of operation.
3 DEV_RESET	ENABLED = 0x0 NANDF_WP_B(WPN) pin is held low (asserted). DISABLED = 0x1 NANDF_WP_B(WPN) pin is held high (de-asserted). 0 NANDF_WP_B pin is held low (asserted). 1 NANDF_WP_B pin is held high (de-asserted).
2 ATA_IRQRDY_POLARITY	For ATA MODE: Note NAND_RDY_BUSY[3:2] are not affected by this bit. ACTIVELOW = 0x0 ATA IORDY and IRQ are active low, or NAND_RDY_BUSY[1:0] are active low ready. ACTIVEHIGH = 0x1 ATA IORDY and IRQ are active high, or NAND_RDY_BUSY[1:0] are active high ready. 0 External ATA IORDY and IRQ are active low. 1 External ATA IORDY and IRQ are active high. For NAND MODE: 0 External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when low and busy when high. 1 External RDY_BUSY[1] and RDY_BUSY[0] pins are ready when high and busy when low.
1 CAMERA_MODE	When set to 1 and ATA UDMA is enabled the UDMA interface becomes a camera interface.
0 GPMI_MODE	ATA mode is only supported on channel zero. If ATA mode is selected, then only channel three is available for NAND use. NAND = 0x0 NAND mode. ATA = 0x1 ATA mode.

Table continues on the next page...

GPMI_CTRL1*n* field descriptions (continued)

Field	Description
	0 NAND mode. 1 ATA mode.

9.7.5.8 GPMI Timing Register 0 Description (GPMI_TIMING0)

The GPMI timing register 0 specifies the timing parameters that are used by the cycle state machine to guarantee the various setup, hold and cycle times for the external media type.

GPMI_TIMING0 0x070

Address: 3300_2000h base + 70h offset = 3300_2070h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									ADDRESS_SETUP								DATA_HOLD								DATA_SETUP							
W	0																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	1

GPMI_TIMING0 field descriptions

Field	Description
31–24 RSVD1	Always write zeroes to this bit field.
23–16 ADDRESS_ SETUP	Number of GPMICLK cycles that the CE/ADDR signals are active before a strobe is asserted. A value of zero is interpreted as 0. For ATA PIO modes this is known in the ATA7 specification as "Address valid to DIOR-/DIOW- setup"
15–8 DATA_HOLD	Data bus hold time in GPMICLK cycles. Also the time that the data strobe is de-asserted in a cycle. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as "DIOR-/DIOW- recovery time"
DATA_SETUP	Data bus setup time in GPMICLK cycles. Also the time that the data strobe is asserted in a cycle. This value must be greater than 2 for ATA devices that use IORDY to extend transfer cycles. A value of zero is interpreted as 256. For ATA PIO modes this is known in the ATA7 specification as ""DIOR-/DIOW-"

9.7.5.9 GPMI Timing Register 1 Description (GPMI_TIMING1)

The GPMI timing register 1 specifies the timeouts used when monitoring the NAND READY pin or the ATA IRQ and IOWAIT signals.

GPMI_TIMING1 0x080

Address: 3300_2000h base + 80h offset = 3300_2080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEVICE_BUSY_TIMEOUT																RSVD1																
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_TIMING1 field descriptions

Field	Description
31–16 DEVICE_BUSY_TIMEOUT	Timeout waiting for NAND Ready/Busy or ATA IRQ. Used in WAIT_FOR_READY mode. This value is the number of GPMI_CLK cycles multiplied by 4096.
RSVD1	Always write zeroes to this bit field.

9.7.5.10 GPMI Timing Register 2 Description (GPMI_TIMING2)

The GPMI timing register 2 specifies the double data rate timing parameters that are used by the cycle state machine to guarantee the various cs delay, pre-amble delay, post-amble delay, command/address delay, data delay, TCR, TRPSTH, and read latency cycle times for the external media type.

GPMI_TIMING2 0x090

Address: 3300_2000h base + 90h offset = 3300_2090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TRPSTH			TCR		READ_LATENCY			RSVD0			CE_DELAY					PREAMBLE_DELAY				POSTAMBLE_DELAY				CMDADD_PAUSE				DATA_PAUSE				
W																																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	1	0	0	1	1	0	1	1	0

GPMI_TIMING2 field descriptions

Field	Description
31–29 TRPSTH	Only for Toggle NAND timing control delay TRPSTH GPMICLK cycles for CEn_B high to RE_B high, A value of zero is interpreted as 8
28–27 TCR	Only for Toggle NAND timing control delay (TCR+1) GPMICLK cycles for CEn_B low to RE_B low, 0 is less than or equal to TCR, which is less than the PREAMBLE_DELAY
26–24 READ_LATENCY	This field is for double data rate read latency configuration. others READ LATENCY is 3 000 READ LATENCY is 0 001 READ LATENCY is 1 010 READ LATENCY is 2 011 READ LATENCY is 3 100 READ LATENCY is 4 101 READ LATENCY is 5

Table continues on the next page...

GPMI_TIMING2 field descriptions (continued)

Field	Description
23–21 RSVD0	Always write zeroes to this bit field.
20–16 CE_DELAY	GPMI dealy from CEn assert to W/Rn changing edge. value of zero is interpreted as 32.
15–12 PREAMBLE_ DELAY	GPMI pre-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
11–8 POSTAMBLE_ DELAY	GPMI post-amble delay in GPMICLK cycles. A value of zero is interpreted as 16.
7–4 CMDADD_ PAUSE	GPMI delay time from command or addres pause to command or address resume in GPMICLK cycles. A value of zero is interpreted as 16.
DATA_PAUSE	GPMI delay time from data pause to data resume in GPMICLK cycles. A value of zero is interpreted as 16.

9.7.5.11 GPMI DMA Data Transfer Register Description (GPMI_DATA)

The GPMI DMA data transfer register is used by the DMA to read or write data to or from the ATA/NAND control state machine.

GPMI_DATA 0x0A0

Address: 3300_2000h base + A0h offset = 3300_20A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DATA field descriptions

Field	Description
DATA	In 8-bit mode, one, two, three or four bytes can can be accessed to send the same number of bus cycles.

9.7.5.12 GPMI Status Register Description (GPMI_STAT)

The GPMI control and status register provides a read back path for various operational states of the GPMI controller.

GPMI_STAT 0x0B0

Address: 3300_2000h base + B0h offset = 3300_20B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	READY_BUSY								RDY_TIMEOUT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEV7_ERROR	DEV6_ERROR	DEV5_ERROR	DEV4_ERROR	DEV3_ERROR	DEV2_ERROR	DEV1_ERROR	DEV0_ERROR	RSVD1			ATA_IRQ	INVALID_BUFFER_MASK	FIFO_EMPTY	FIFO_FULL	PRESENT
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

GPMI_STAT field descriptions

Field	Description
31–24 READY_BUSY	Read-only view of NAND Ready_Busy Input pins.
23–16 RDY_TIMEOUT	State of the RDY/BUSY Timeout Flags. When any bit is set to '1' in this field, it indicates that a time out has occurred while waiting for the ready state of the requested NAND device. Multiple bits may be set simultaneously. When GPMI_CTRL1_DECOUPLE_CS = 0, RDY_TIMEOUT[n] is associated with the NAND device on chip_select[n]. When GPMI_CTRL1_DECOUPLE_CS = 1, these flags become associated to a DMA channel instead of a NAND device. For example if DMA channel 6 sends a WAIT_FOR_READY command for NAND Device 2, and a timeout occured on READY_BUSY2, then READY_TIMEOUT[6] will be set instead of READY_TIMEOUT[2].
15 DEV7_ERROR	DMA channel 7 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 7. 1 An Error has occurred on ATA/NAND Device accessed by
14 DEV6_ERROR	DMA channel 6 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 6. 1 An Error has occurred on ATA/NAND Device accessed by
13 DEV5_ERROR	DMA channel 5 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 5. 1 An Error has occurred on ATA/NAND Device accessed by
12 DEV4_ERROR	DMA channel 4 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 4. 1 An Error has occurred on ATA/NAND Device accessed by
11 DEV3_ERROR	DMA channel 3 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 3. 1 An Error has occurred on ATA/NAND Device accessed by
10 DEV2_ERROR	DMA channel 2 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 2. 1 An Error has occurred on ATA/NAND Device accessed by
9 DEV1_ERROR	DMA channel 1 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 1. 1 An Error has occurred on ATA/NAND Device accessed by
8 DEV0_ERROR	DMA channel 0 (Timeout or compare failure, depending on COMMAND_MODE). 0 No error condition present on ATA/NAND Device accessed by DMA channel 0. 1 An Error has occurred on ATA/NAND Device accessed by
7–5 RSVD1	Always write zeroes to this bit field.

Table continues on the next page...

GPMI_STAT field descriptions (continued)

Field	Description
4 ATA_IRQ	Status of the ATA_IRQ input pin.
3 INVALID_ BUFFER_MASK	Buffer Mask Validity bit. 0 ECC Buffer Mask is not invalid. 1 ECC Buffer Mask is invalid.
2 FIFO_EMPTY	NOT_EMPTY = 0x0 FIFO is not empty. EMPTY = 0x1 FIFO is empty. 0 FIFO is not empty. 1 FIFO is empty.
1 FIFO_FULL	NOT_FULL = 0x0 FIFO is not full. FULL = 0x1 FIFO is full. 0 FIFO is not full. 1 FIFO is full.
0 PRESENT	UNAVAILABLE = 0x0 GPMI is not present in this product. AVAILABLE = 0x1 GPMI is present in this product. 0 GPMI is not present in this product. 1 GPMI is present is in this product.

9.7.5.13 GPMI Debug Information Register Description (GPMI_DEBUG)

The GPMI debug information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG 0x0C0

Address: 3300_2000h base + C0h offset = 3300_20C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WAIT_FOR_READY_END								DMA_SENSE								DMAREQ								CMD_END							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DEBUG field descriptions

Field	Description
31–24 WAIT_FOR_ READY_END	Read Only view of the Wait_For_Ready End toggle signals to DMA. One per channel

Table continues on the next page...

GPMI_DEBUG field descriptions (continued)

Field	Description
23–16 DMA_SENSE	Read-only view of sense state of the 8 DMA channels. A value of "1" in any bit position indicates that a read and compare command failed or a timeout occured for the corresponding channel.
15–8 DMAREQ	Read-only view of DMA request line for 8 DMA channels. A toggle on any bit position indicates a DMA request for the corresponding channel.
CMD_END	Read Only view of the Command End toggle signals to DMA. One per channel

9.7.5.14 GPMI Version Register Description (GPMI_VERSION)

This register reflects the version number for the GPMI.

GPMI_VERSION 0x0D0

Address: 3300_2000h base + D0h offset = 3300_20D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR								MINOR								STEP															
W																																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_VERSION field descriptions

Field	Description
31–24 MAJOR	Fixed read-only value reflecting the MAJOR field of the RTL version.
23–16 MINOR	Fixed read-only value reflecting the MINOR field of the RTL version.
STEP	Fixed read-only value reflecting the stepping of the RTL version.

9.7.5.15 GPMI Debug2 Information Register Description (GPMI_DEBUG2)

The GPMI Debug2 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG2 0x0E0

Address: 3300_2000h base + E0h offset = 3300_20E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVD1				UDMA_STATE				BUSY	PIN_STATE			MAIN_STATE			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SYND2GPMI_BE				GPMI2SYND_VALID	GPMI2SYND_READY	SYND2GPMI_VALID	SYND2GPMI_READY	VIEW_DELAYED_RDN	UPDATE_WINDOW	RDN_TAP					
W																
Reset	1	1	1	1	0	0	0	1	0	0	0	0	0	0	0	0

GPMI_DEBUG2 field descriptions

Field	Description
31–28 RSVD1	Always write zeroes to this bit field.
27–24 UDMA_STATE	USM_IDLE = 4'h0, idle USM_DMARQ = 4'h1, DMA req USM_ACK = 4'h2, DMA ACK USM_FIFO_E = 4'h3, Fifo empty USM_WPAUSE = 4'h4, WR DMA Paused by device USM_TSTRB = 4'h5, Toggle HSTROBE USM_CAPTUR = 4'h6, Capture Stage, (data sampled with DSTROBE is valid) USM_DATOUT = 4'h7, Change Burst DATAOUT USM_CRC = 4'h8, Source CRC to Device USM_WAIT_R = 4'h9, Waiting for DDMARDY- USM_END = 4'ha; Negate DMAACK (end of DMA) USM_WAIT_S = 4'hb, Waiting for DSTROBE USM_RPAUSE = 4'hc, Rd DMA Paused by Host USM_RSTOP = 4'hd, Rd DMA Stopped by Host USM_WTERM = 4'he, Wr DMA Termination State USM_RTERM = 4'hf, Rd DMA Termination state
23 BUSY	When asserted the GPMI is busy. Undefined results may occur if any registers are written when BUSY is asserted. DISABLED = 0x0 The GPMI is not busy. ENABLED = 0x1 The GPMI is busy.
22–20 PIN_STATE	parameter PSM_IDLE = 3'h0, PSM_BYTCNT = 3'h1, PSM_ADDR = 3'h2, PSM_STALL = 3'h3, PSM_STROBE = 3'h4, PSM_ATARDY = 3'h5, PSM_DHOLD = 3'h6, PSM_DONE = 3'h7. PSM_IDLE = 0x0 PSM_BYTCNT = 0x1 PSM_ADDR = 0x2 PSM_STALL = 0x3 PSM_STROBE = 0x4 PSM_ATARDY = 0x5 PSM_DHOLD = 0x6 PSM_DONE = 0x7
19–16 MAIN_STATE	parameter MSM_IDLE = 4'h0, MSM_BYTCNT = 4'h1, MSM_WAITFE = 4'h2, MSM_WAITFR = 4'h3, MSM_DMAREQ = 4'h4, MSM_DMAACK = 4'h5, MSM_WAITFF = 4'h6, MSM_LDFIFO = 4'h7, MSM_LDDMAR = 4'h8, MSM_RDCMP = 4'h9, MSM_DONE = 4'hA. MSM_IDLE = 0x0 MSM_BYTCNT = 0x1 MSM_WAITFE = 0x2 MSM_WAITFR = 0x3

Table continues on the next page...

GPMI_DEBUG2 field descriptions (continued)

Field	Description
	MSM_DMAREQ = 0x4 MSM_DMAACK = 0x5 MSM_WAITFF = 0x6 MSM_LDFIFO = 0x7 MSM_LDDMAR = 0x8 MSM_RDCMP = 0x9 MSM_DONE = 0xA
15–12 SYND2GPMI_BE	Data byte enable Input from BCH.
11 GPMI2SYND_VALID	Data handshake output to BCH.
10 GPMI2SYND_READY	Data handshake output to BCH.
9 SYND2GPMI_VALID	Data handshake Input from BCH.
8 SYND2GPMI_READY	Data handshake Input from BCH.
7 VIEW_DELAYED_RDN	Set to a 1 to select the delayed feedback RE_B to drive the GPMI_ADDR[0] (Nand CLE) pin. For debug purposes, this will allow you see if DLL is functioning properly.
6 UPDATE_WINDOW	A 1 indicates that the DLL is busy generating the required delay.
RDN_TAP	This is the DLL tap calculated by the DLL controller. The selects the amount of delay form the DLL chain.

9.7.5.16 GPMI Debug3 Information Register Description (GPMI_DEBUG3)

The GPMI Debug3 information register provides a read back path for diagnostics to determine the current operating state of the GPMI controller.

GPMI_DEBUG3 0x0F0

Address: 3300_2000h base + F0h offset = 3300_20F0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	APB_WORD_CNTR																DEV_WORD_CNTR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPMI_DEBUG3 field descriptions

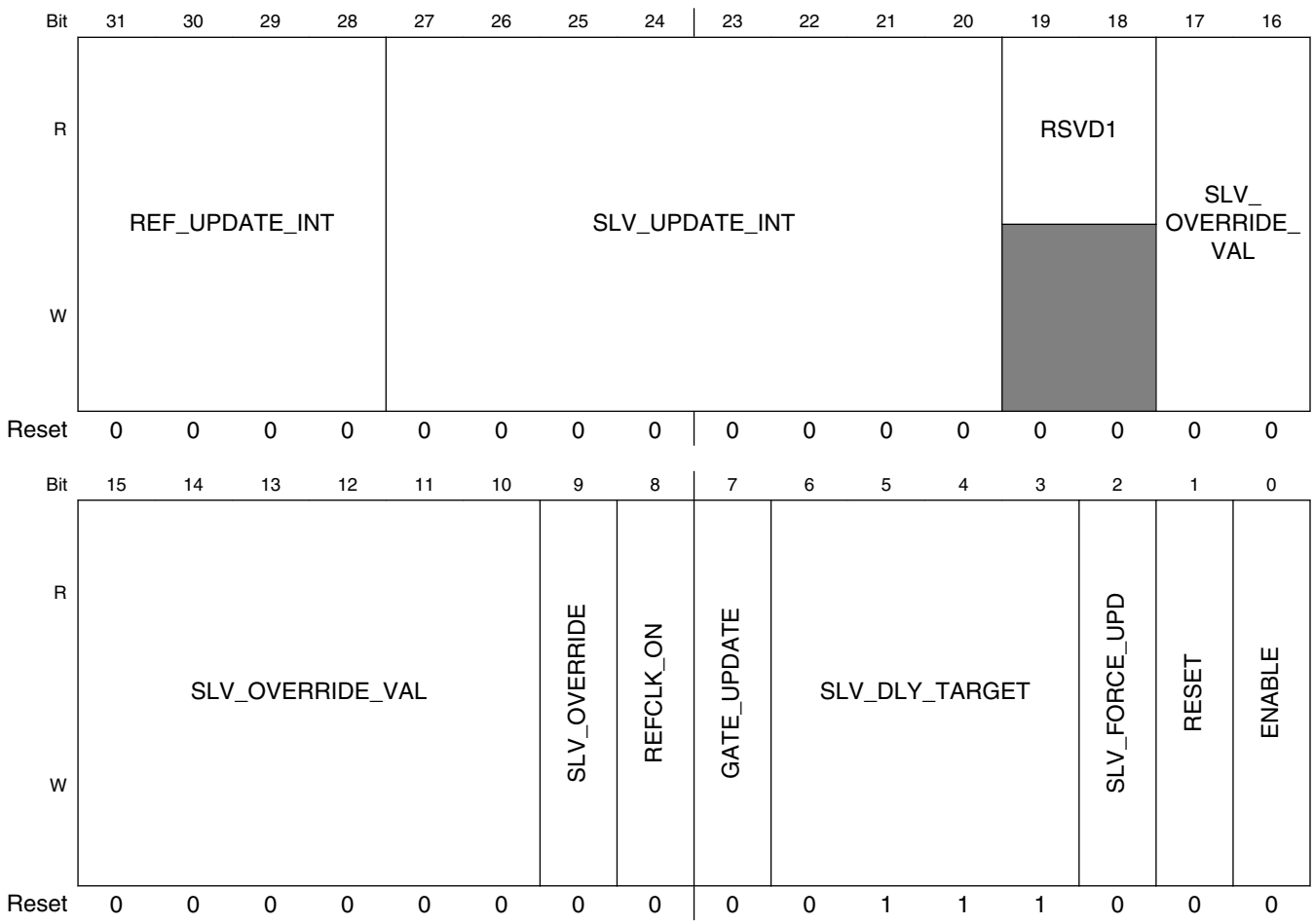
Field	Description
31–16 APB_WORD_ CNTR	Reflects the number of bytes remains to be transferred on the APB bus.
DEV_WORD_ CNTR	Reflects the number of bytes remains to be transferred on the ATA/Nand bus.

9.7.5.17 GPMI Double Rate Read DLL Control Register Description (GPMI_READ_DDR_DLL_CTRL)

GPMI DDR Read Delay Loop Lock Control Register. This register provides programmability in DDR mode for data input timing and data formats.

GPMI_READ_DDR_DLL_CTRL 0x100

Address: 3300_2000h base + 100h offset = 3300_2100h



GPMI_READ_DDR_DLL_CTRL field descriptions

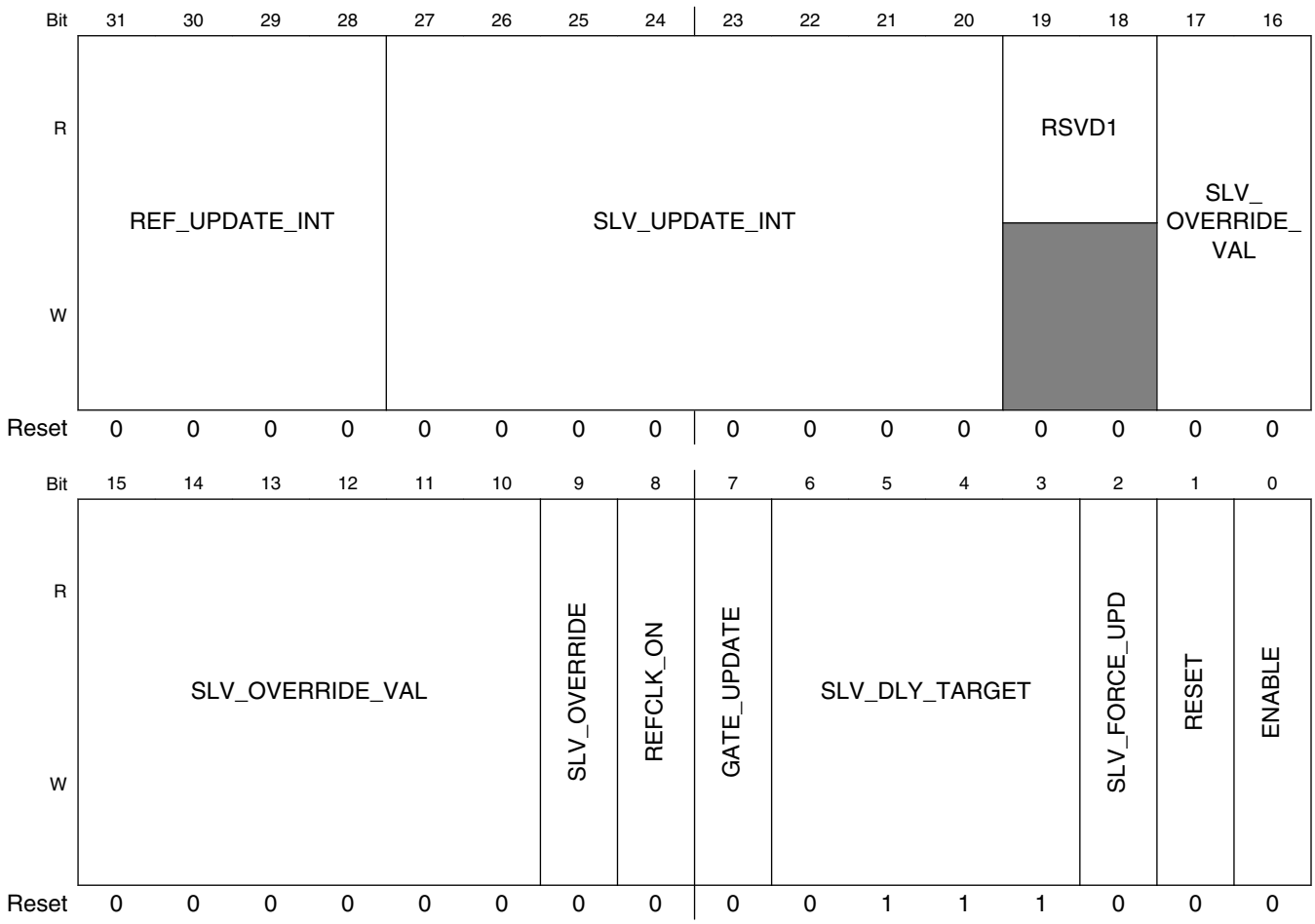
Field	Description
31–28 REF_UPDATE_INT	This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 SLV_UPDATE_INT	Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19–18 RSVD1	Reserved
17–10 SLV_OVERRIDE_VAL	When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
9 SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0
8 REFCLK_ON	set this bit to 1 will turn on the reference clock
7 GATE_UPDATE	Setting this bit to 1, forces the slave delay line not update
6–3 SLV_DLY_TARGET	The delay target for the read clock is can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
2 SLV_FORCE_UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
1 RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
0 ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

9.7.5.18 GPMI Double Rate Write DLL Control Register Description (GPMI_WRITE_DDR_DLL_CTRL)

GPMI DDR Write Delay Loop Lock Control Register. This register provides programmability in DDR mode for data output timing and data formats.

GPMI_WRITE_DDR_DLL_CTRL 0x110

Address: 3300_2000h base + 110h offset = 3300_2110h



GPMI_WRITE_DDR_DLL_CTRL field descriptions

Field	Description
31–28 REF_UPDATE_ INT	This field allows the user to add additional delay cycles to the DLL control loop (reference delay line control). By default, the DLL control loop shall update every two GPMICLK cycles. Programming this field results in a DLL control loop update interval of (2 + REF_UPDATE_INT) * GPMICLK. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that may effect the delay (such as voltage and temperature)
27–20 SLV_UPDATE_ INT	Setting a value greater than 0 in this field, shall over-ride the default slave delay-line update interval of 256 GPMICLK cycles. A value of 0 results in an update interval of 256 GPMICLK cycles (default setting). A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line will also update automatically when the reference DLL transitions to a locked state (from an un-locked state).
19–18 RSVD1	Reserved
17–10 SLV_ OVERRIDE_VAL	When SLV_OVERRIDE=1 This field is used to select 1 of 256 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 256.
9 SLV_OVERRIDE	Set this bit to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to tbe enabled using the ENABLE bit. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE=0

Table continues on the next page...

GPMI_WRITE_DDR_DLL_CTRL field descriptions (continued)

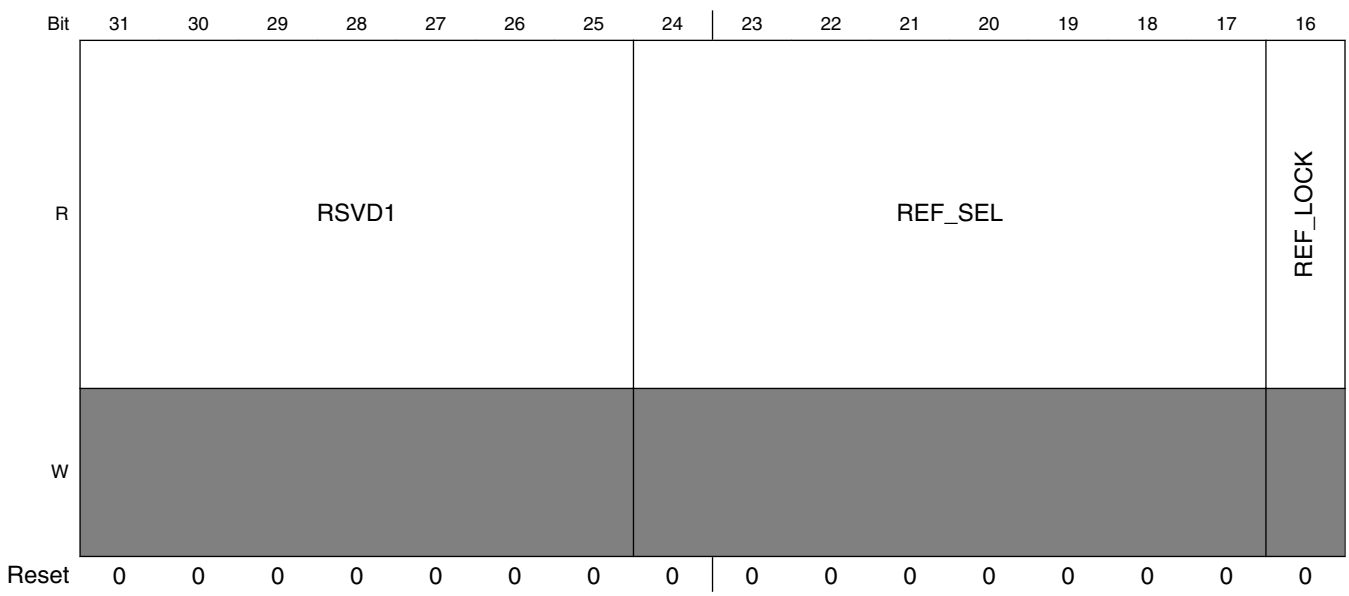
Field	Description
8 REFCLK_ON	set this bit to 1 will turn on the reference clock
7 GATE_UPDATE	Setting this bit to 1, forces the slave delay line not update
6–3 SLV_DLY_ TARGET	The delay target for the read clock can be programmed in 1/16th increments of an GPMICLK half-period. So the input read-clock can be delayed relative input data from (GPMICLK/2)/16 to GPMICLK/2.
2 SLV_FORCE_ UPD	Setting this bit to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line shall update automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered).
1 RESET	Setting this bit to 1 force a reset on DLL. This will cause the DLL to lose lock and re-calibrate to detect an GPMICLK half period phase shift. This signal is used by the DLL as edge-sensitive, so in order to create a subsequent reset, RESET must be taken low and then asserted again.
0 ENABLE	Set this bit to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

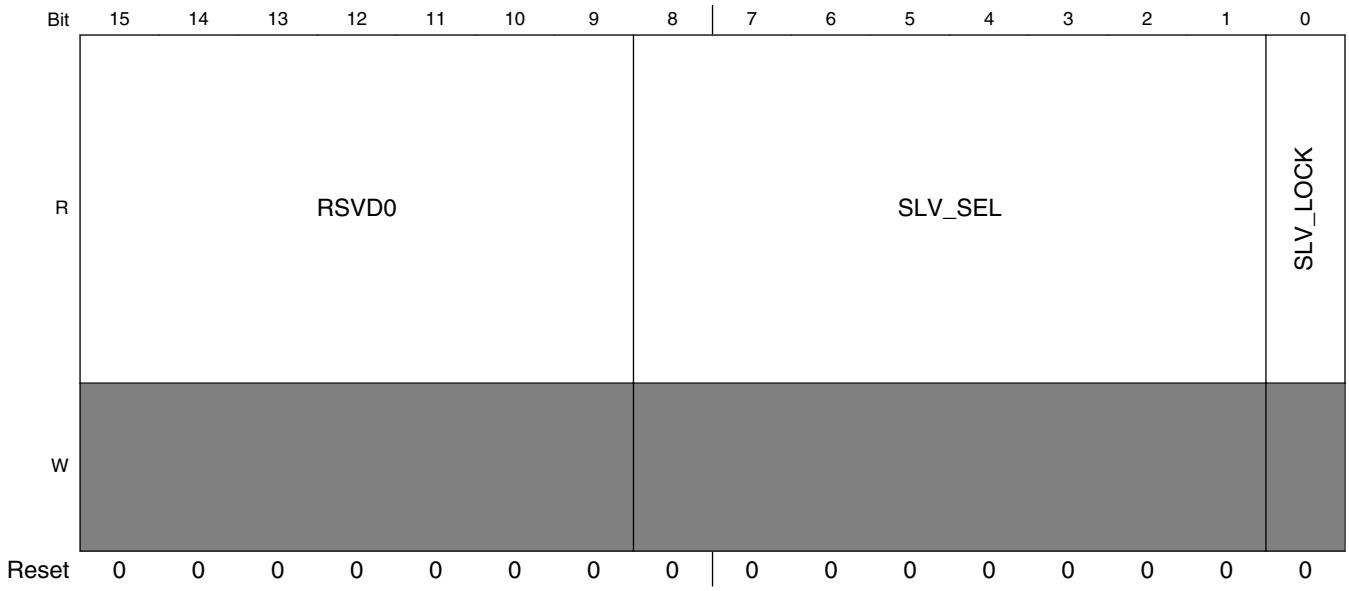
9.7.5.19 GPMI Double Rate Read DLL Status Register Description (GPMI_READ_DDR_DLL_STS)

GPMI Double Rate Read DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_READ_DDR_DLL_STS 0x120

Address: 3300_2000h base + 120h offset = 3300_2120h





GPMI_READ_DDR_DLL_STS field descriptions

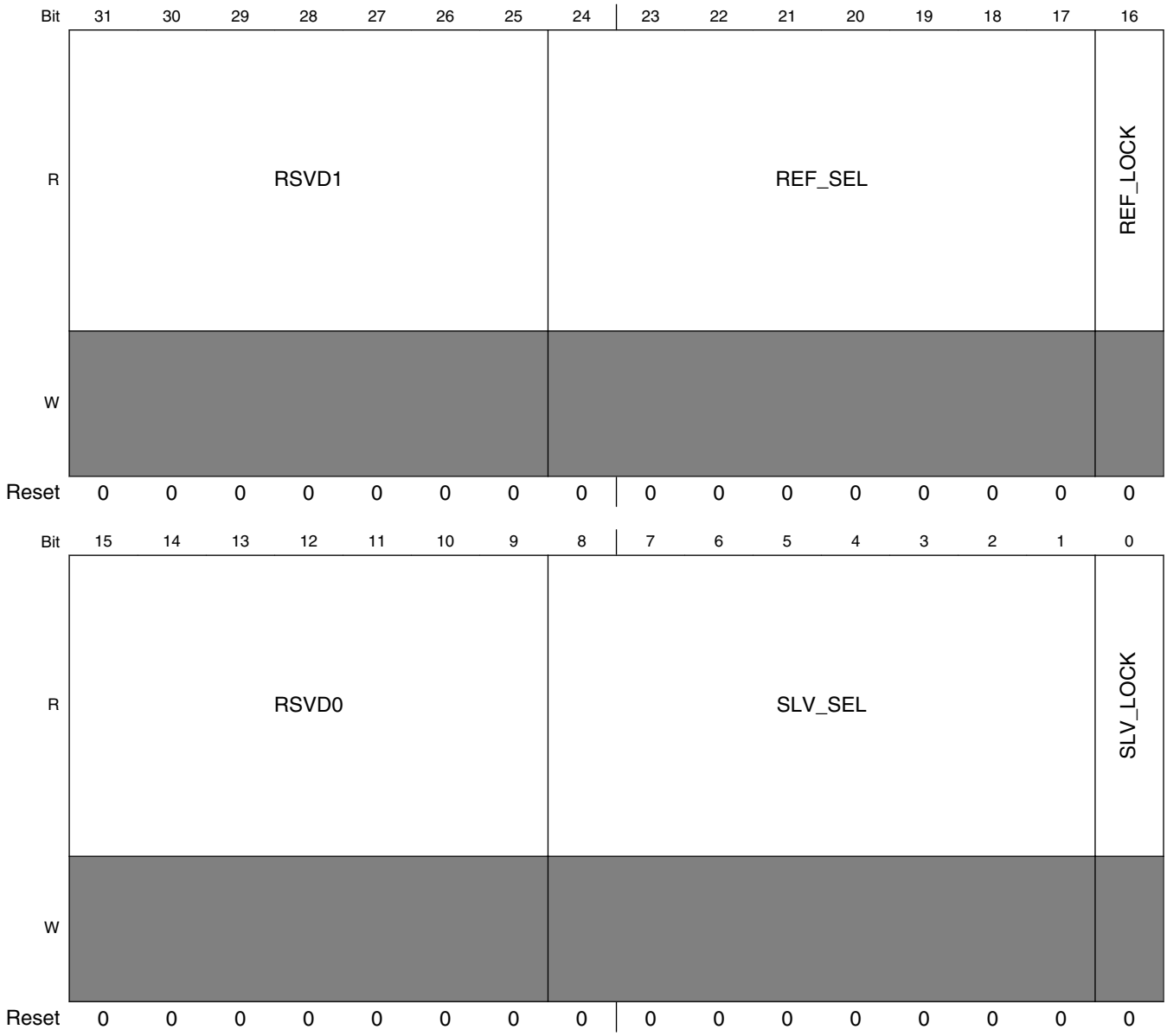
Field	Description
31–25 RSVD1	Reserved
24–17 REF_SEL	Reference delay line select status.
16 REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
15–9 RSVD0	Reserved
8–1 SLV_SEL	Slave delay line select status
0 SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.

9.7.5.20 GPMI Double Rate Write DLL Status Register Description (GPMI_WRITE_DDR_DLL_STS)

GPMI Double Rate Write DLL Status Register, Read Only. GPMI DLL status fields are provided in this register.

GPMI_WRITE_DDR_DLL_STS 0x130

Address: 3300_2000h base + 130h offset = 3300_2130h



GPMI_WRITE_DDR_DLL_STS field descriptions

Field	Description
31–25 RSVD1	Reserved
24–17 REF_SEL	Reference delay line select status.
16 REF_LOCK	Reference DLL lock status. This signifies that the DLL has detected and locked to a half-phase GPMICLK shift, allowing the slave delay-line to perform programmed clock delays.
15–9 RSVD0	Reserved
8–1 SLV_SEL	Slave delay line select status

Table continues on the next page...

GPMI_WRITE_DDR_DLL_STS field descriptions (continued)

Field	Description
0 SLV_LOCK	Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value.

Chapter 10
Mass Storage

10.1 Enhanced Configurable SPI (ECSPI)

10.1.1 Overview

The Enhanced Configurable Serial Peripheral Interface (ECSPI) is a full-duplex, synchronous, four-wire serial communication block.

The ECSPI contains a 64 x 32 receive buffer (RXFIFO) and a 64 x 32 transmit buffer (TXFIFO). With data FIFOs, the ECSPI allows rapid data communication with fewer software interrupts. The figure below shows a block diagram of the ECSPI.

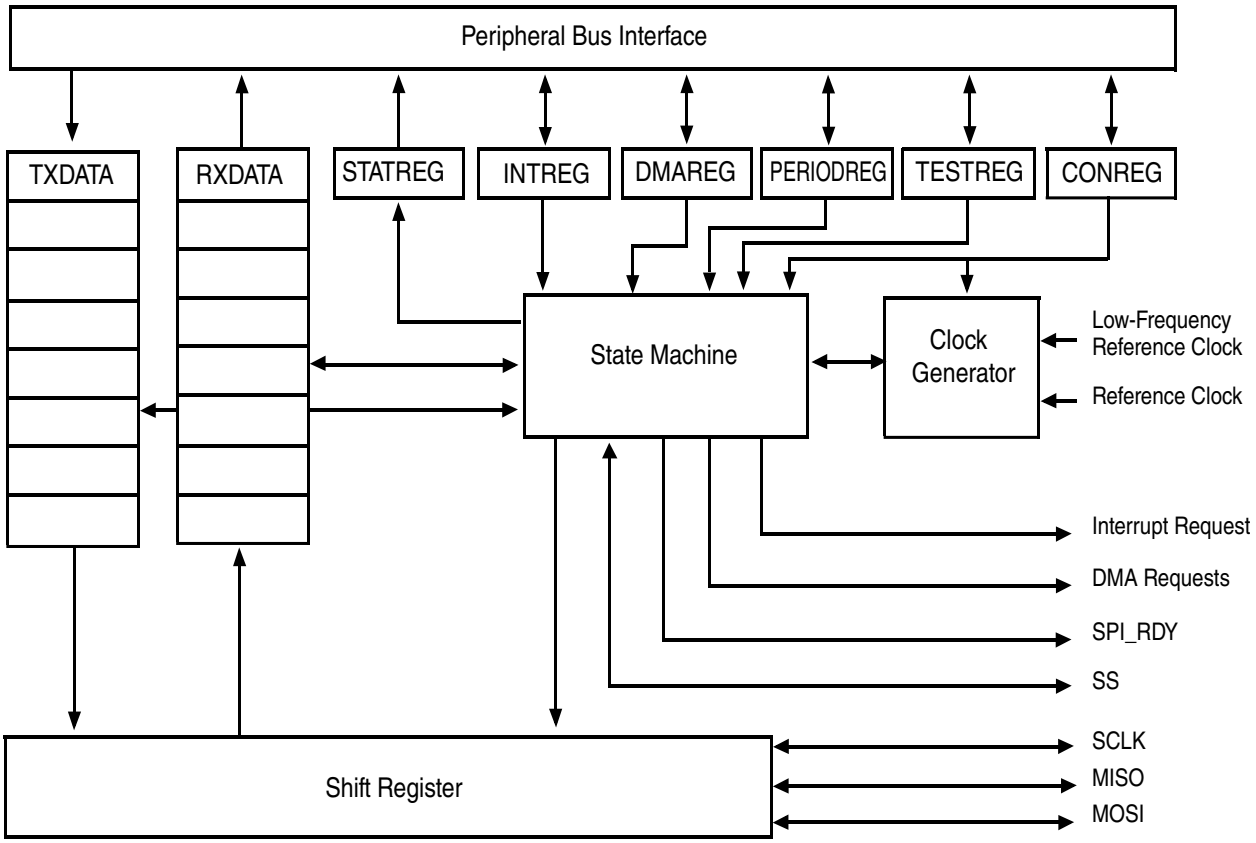


Figure 10-1. ECSPI Block Diagram

10.1.1.1 Features

Key features of the ECSPI include:

- Full-duplex synchronous serial interface
- Master/Slave configurable
- One Chip Select (SS) signal
- Transfer continuation function allows unlimited length data transfers
- 32-bit wide by 64-entry FIFO for both transmit and receive data
- Polarity and phase of the Chip Select (SS) and SPI Clock (SCLK) are configurable
- Direct Memory Access (DMA) support
- Refer to the product data sheet for the maximum operating frequency

10.1.1.2 Modes and Operations

The ECSPI supports the modes described in the indicated sections:

- Master Mode
- Slave Mode
- Low Power Modes

As described in Operations, the ECSPI supports the operations described in the indicated sections:

- Typical Master Mode
  - Master Mode with SPI_RDY
  - Master Mode with Wait States
  - Master Mode with SS_CTL[3:0] Control
  - Master Mode with Phase Control
- Typical Slave Mode

10.1.2 Clocks

The following table describes the clock sources for eCSPI. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 10-1. eCSPI Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_32k	ckil_sync_clk_root	Low-frequency reference clock (32kHz)
ipg_clk_per	ecspi_clk_root	eCSPI module clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

10.1.3 Functional Description

This section provides a complete functional description of the ECSPI. The figure found here shows the relationship of SCLK and data lines while ECSPI has been configured with different POL and PHA settings.

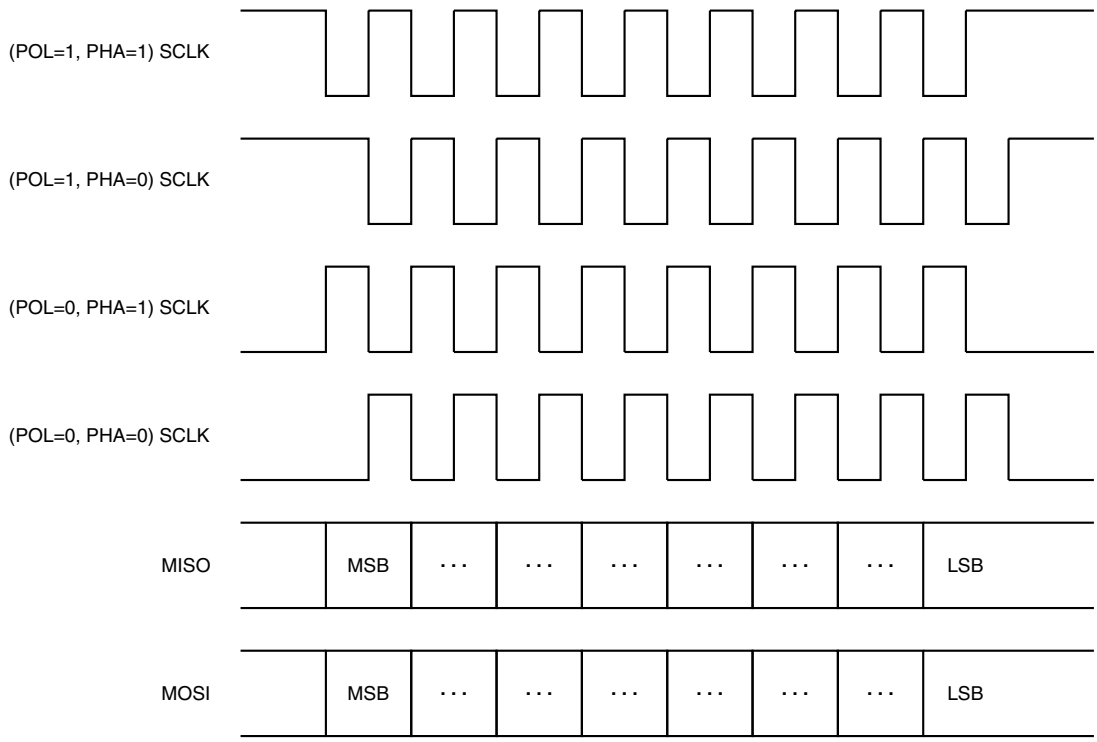


Figure 10-2. ECSPI SCLK, MISO, and MOSI Relationship

10.1.3.1 Master Mode

When the ECSPI is configured as a master, it uses a serial link to transfer data between the ECSPI and an external slave device.

The Chip Select (SS) signal and the clock signal (SCLK) are used to transfer data between two devices. If the external device is a transmit-only device, the ECSPI master's output port can be ignored and used for other purposes. In order to use the internal TXFIFO and RXFIFO, two auxiliary output signals, Chip Select (SS) and SPI_RDY, are used for data transfer rate control. Software can also configure the sample period control register to a fixed data transfer rate.

10.1.3.2 Slave Mode

When the ECSPI is configured as a slave, software can configure the ECSPI Control register to match the external SPI master's timing.

In this configuration, Chip Select (SS) becomes an input signal, and is used to control data transfers through the Shift register, as well as to load/store the data FIFO.

Slave mode only supports the case when ECSPIx_CONFIGREG[SS_CTL] is cleared. The accurate burst length should always be specified using the BURST_LENGTH parameter. ECSPIx_CONFIGREG[SS_CTL] set to 1 is not supported in slave mode.

10.1.3.3 Low Power Modes

The ECSPI does not operate under low power mode.

It holds its operation when its clock is gated off in master mode. In slave mode, the ECSPI does not respond when its clock is gated off.

10.1.3.4 Operations

The information found here describes the ECSPI's operations.

10.1.3.4.1 Typical Master Mode

The ECSPI master uses the Chip Select (SS) signal to enable an external SPI device, and uses the SCLK signal to transfer data in and out of the Shift register.

The SPI_RDY enables fast data communication with fewer software interrupts. By programming the ECSPI_PERIODREG register accordingly, the ECSPI can be used for a fixed data transfer rate.

When the ECSPI is in Master mode the SS, SCLK, and MOSI are output signals, and the MISO signal is an input.

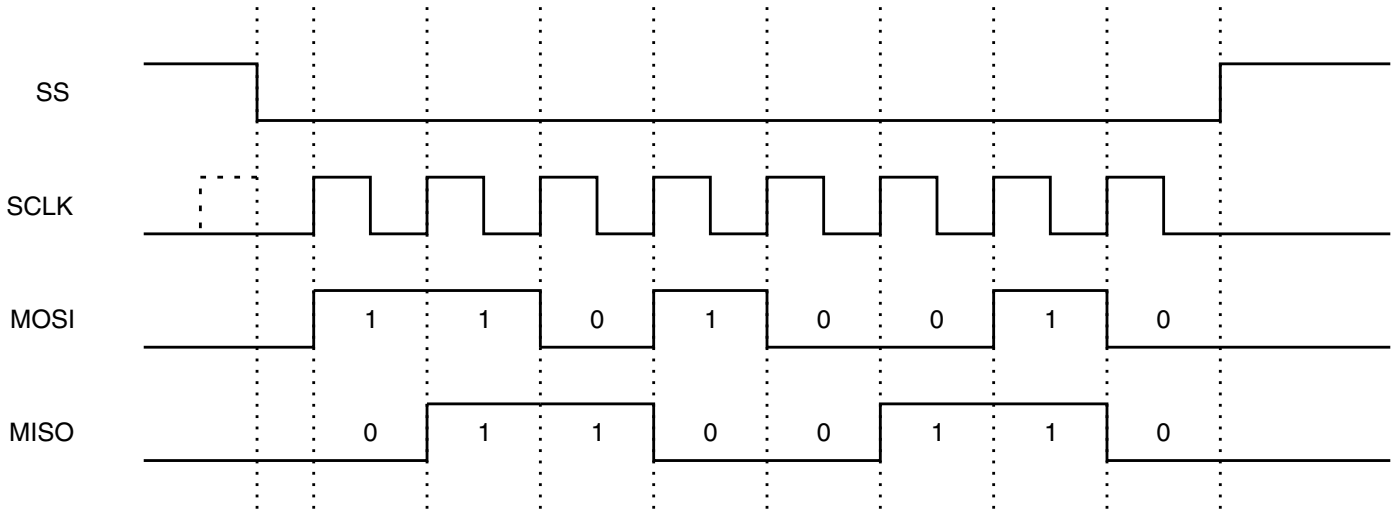


Figure 10-3. Typical SPI Burst (8-bit Transfer)

In the above figure, the Chip Select (SS) signal enables the selected external SPI device, and the SCLK synchronizes the data transfer. The MOSI and MISO signals change on rising edge of SCLK and the MISO signal is latched on the falling edge of the SCLK. The figure above shows a data of 0xD2 is shifted out, and a data of 0x66 is shifted in.

10.1.3.4.1.1 Master Mode with SPI_RDY

By default, the ECSPI does not use the SPI_RDY signal in master mode (MODE =1).

A SPI burst begins when the following events happen:

- The ECSPI is enabled, TXFIFO has data in it, and ECSPI_CONREG[XCH] bit or the ECSPI_CONREG[SMC] bit is set.
- When the SPI Data Ready Control (ECSPI_CONREG[DRCTL]) bits contains either 01 or 10, the SPI_RDY signal controls when a SPI burst starts.

A SPI burst is defined as a bus transaction that starts when the slave select is asserted and ends when the slave select is negated. The Chip Select (SS) signal will remain asserted until all the bits in a SPI burst are shifted out.

If ECSPI_CONREG[DRCTL] is set to 01, the SPI burst can be triggered only if a falling edge of the SPI_RDY signal has been detected.

The following figure shows the relationship between a SPI burst and the falling edge of SPI_RDY signal.

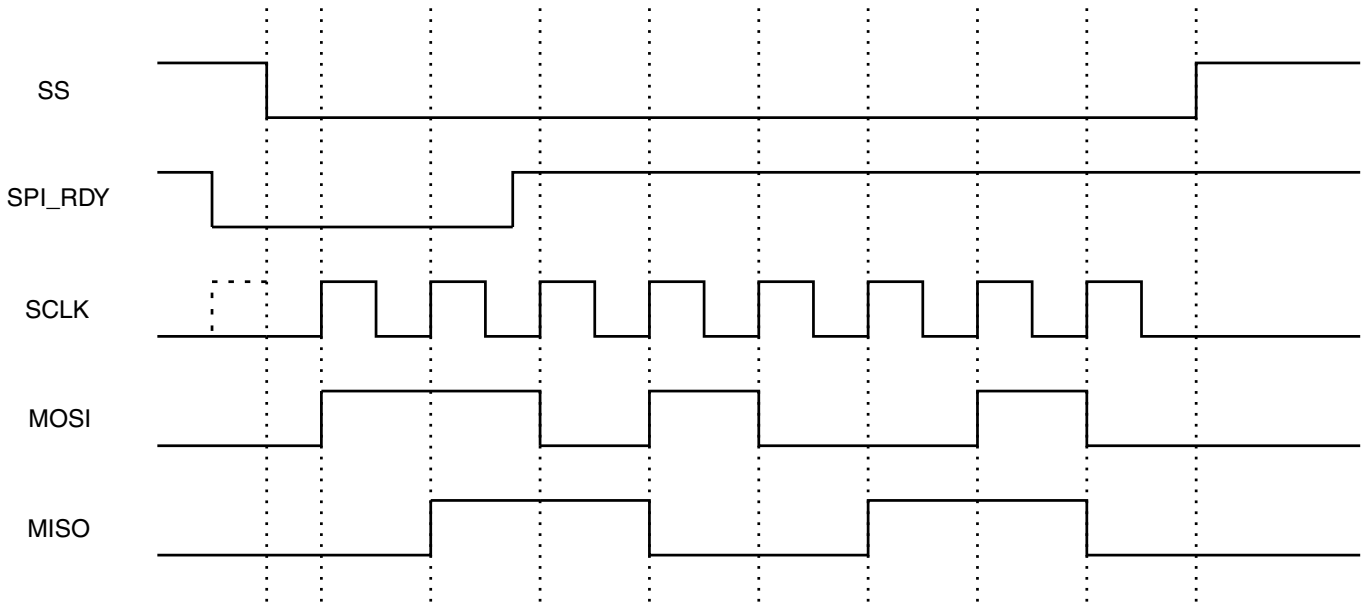


Figure 10-4. Relationship Between a SPI Burst and SPI_RDY: Falling-Edge Triggered

A SPI burst does not start until the falling edge of the SPI_RDY signal is detected. The next SPI burst starts when the next SPI_RDY falling edge is detected, after the last burst has finished.

If SPI Data Ready Control (ECSPI_CONREG[DRCTL]) is set to 10, the SPI burst can be triggered only if the SPI_RDY signal is low.

The following figure shows the relationship between a SPI burst and the SPI_RDY signal. The SPI burst does not begin until the SPI_RDY signal goes low. The ECSPI will keep transmitting SPI burst if the SPI_RDY signal remains low.

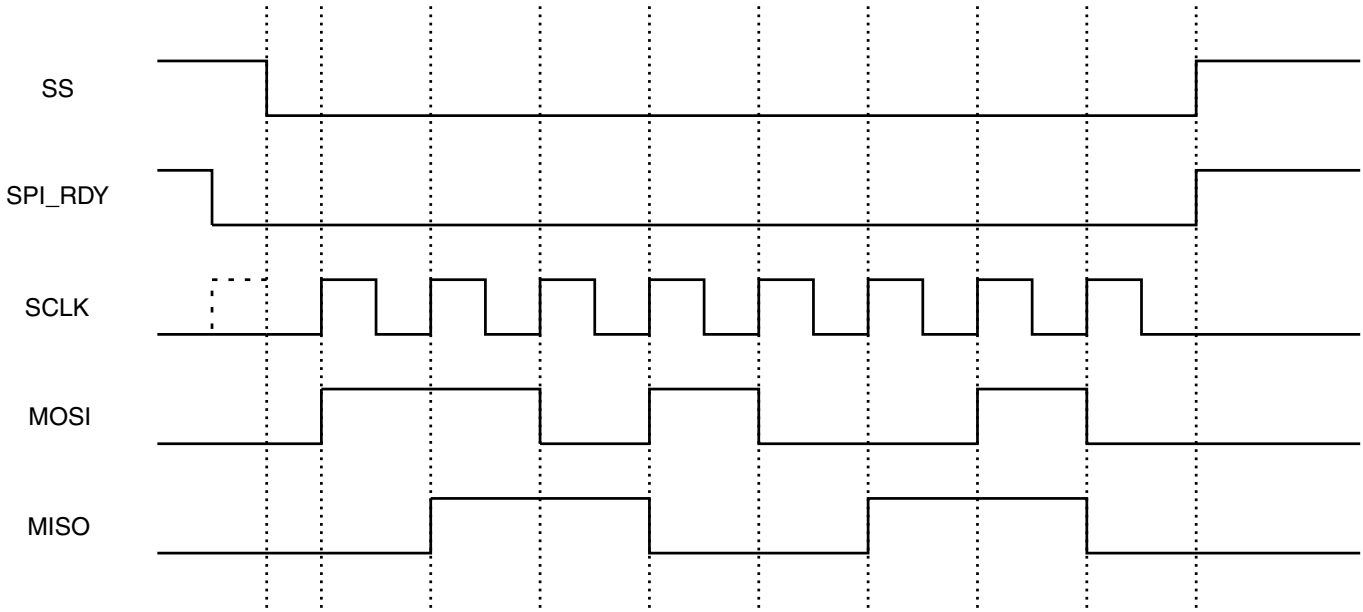


Figure 10-5. Relationship Between a SPI Burst and SPI_RDY: Low-Level Triggered

10.1.3.4.1.2 Master Mode with Wait States

Wait states can be inserted between SPI bursts. This provides a way for software to slow down the SPI burst to meet the timing requirements of a slower SPI device.

The following figure shows wait states inserted between SPI bursts.

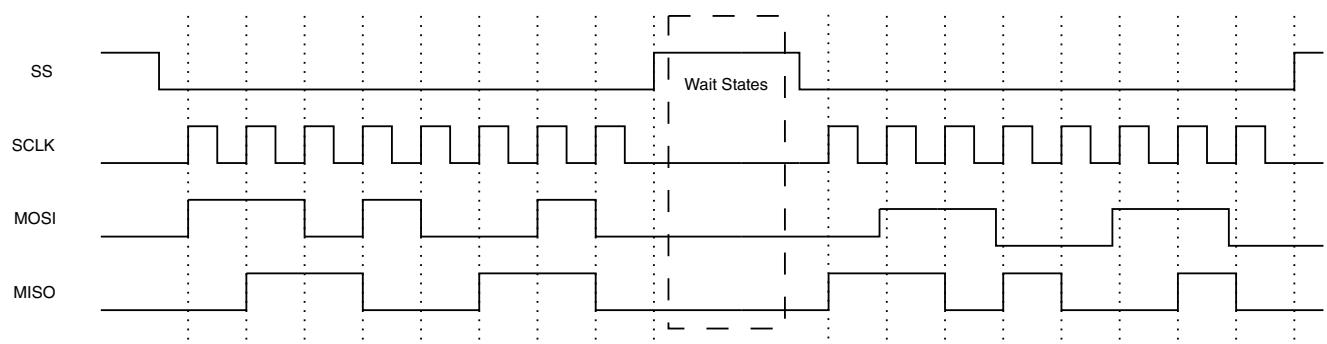


Figure 10-6. SPI Bursts with Wait States

In this case, the number of wait states is controlled by ECSPI_PERIODREG[SAMPLE PERIOD] and the wait states' clock source is selected by ECSPI_PERIODREG[CSRC].

10.1.3.4.1.3 Master Mode with SS_CTL[3:0] Control

The SPI SS Control (SS_CTL[3:0]) controls whether the current operation is single burst or multiple bursts.

When the SPI SS Wave Form Select (SS_CTL[3:0]) is set, the current operation is multiple bursts transfer. When the SPI SS Wave Form Select (SS_CTL[3:0]) bit is cleared, the current operation is single burst transfer. A SPI burst can contains multiple words as defined in the BURST LENGTH field of the ECSPI_CONREG register.

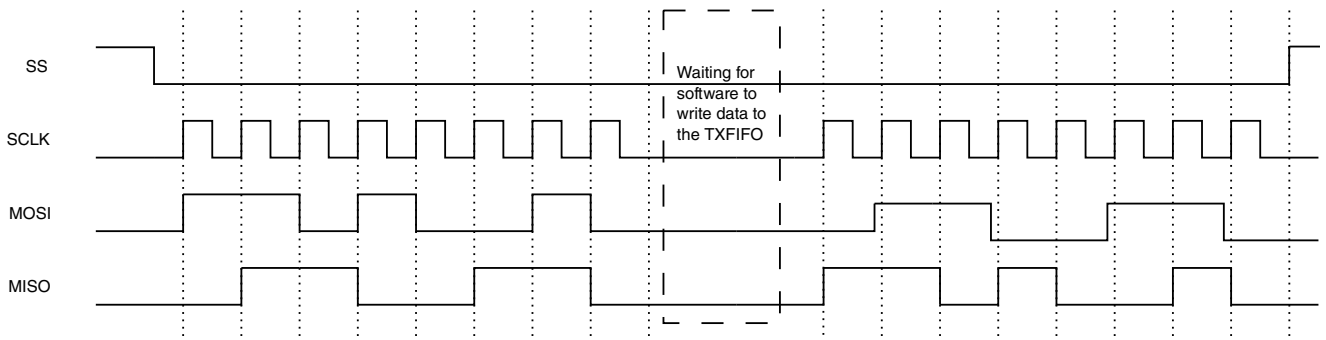


Figure 10-7. SPI Burst While SS_CTL[3:0] is Clear

In Figure 10-7, two 8-bit bursts in the TXFIFO have been combined and transmitted in one SPI burst. The maximum length of a single SPI burst is defined by the BURST LENGTH and limited by the FIFO size. (Figure 10-7 corresponds to a BURST LENGTH of 8.) This provides a way for transferring a longer SPI burst by writing data into TXFIFO while the ECSPI is transmitting.

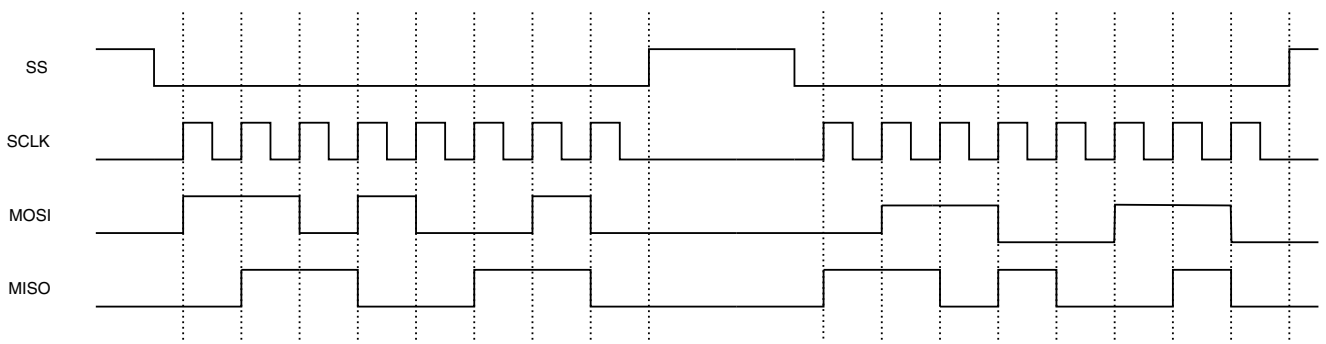


Figure 10-8. SPI Bursts While SS_CTL[3:0] is Set

In Figure 10-8, two FIFO entries are transmitted, one entry with each SPI burst. The ECSPI will continue to transmit SPI bursts until the TXFIFO is empty. When wait states can be inserted between SPI bursts, the SS will negate between SPI bursts until the wait states finish.

10.1.3.4.1.4 Master Mode with Phase Control

The Phase Control (ECSPI_CONREG[PHA]) bit controls how the transmit data shifts out and the receive data shifts in.

When the Phase control (ECSPI_CONREG[PHA]) bit is set, the transmit data will shift out on the rising edge of SCLK, and the receive data is latched on the falling edge of SCLK. The most-significant bit is output on the first rising SCLK edge.

When ECSPI_CONREG[PHA] is cleared, the transmit data is shifted out on the falling edge of SCLK and the receive data is latched on the rising edge of SCLK. The MSB is output when the host processor loads the transmitted data.

Inverting the SCLK polarity does not impact the edge-triggered operations because they are internal to the serial peripheral interface master. Figure 10-9 shows how SPI burst works with different POL and PHA configuration.

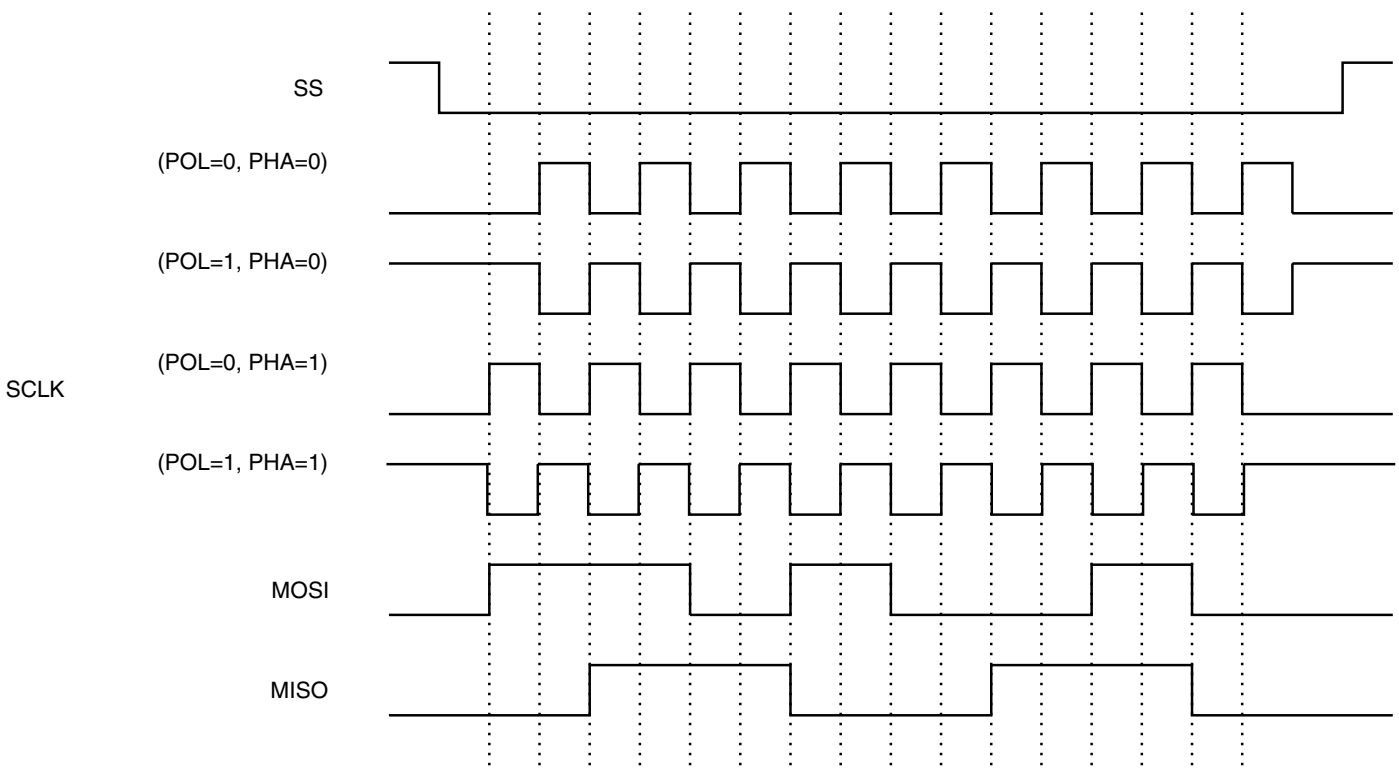


Figure 10-9. SPI Burst with Different POL and PHA Configurations

10.1.3.4.2 Typical Slave Mode

When the ECSPI is configured as a slave (Mode = 0), software can configure the ECSPI Control register to match the external SPI master's timing. In this configuration, SS becomes an input signal, and is used to latch data in and out of the internal data Shift registers, as well as to advance the data FIFO.

The SS, SCLK, and MOSI are inputs and MISO is output. Most of the timing diagrams are similar to the diagrams shown previously for the SPI in Master mode (Mode = 1), because the inputs come from a SPI master device.

However, the timing is different when SS is used to advance the data FIFO. When the SS_POL=0 is set while the ECSPI is configured in Slave mode, the data FIFO will advance on the rising edge of the SS signal. When the polarity is reversed (SS_POL = 1), the data FIFO will advance on the falling edge of the SS signal.

The figure below shows a SPI burst in which the data FIFO is advanced by the rising edge of the SS signal.

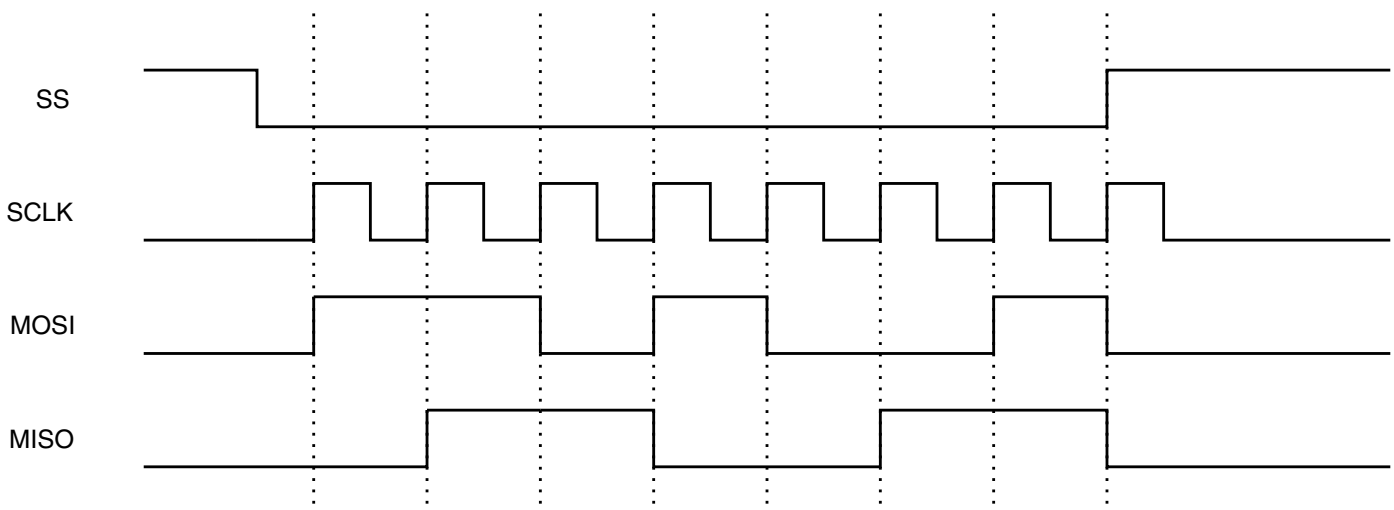


Figure 10-10. Advancing the Data FIFO on the Rising Edge of SS

In the above case, only the most significant 7 bits are loaded to the RXFIFO.

10.1.3.5 Reset

Whenever a device reset occurs, a reset is performed on the ECSPI, resetting all registers to their default values.

Software can reset the block using the CONREG[EN] bit; see ECSPI.

10.1.3.6 Interrupts

Interrupt control provides a way to manage the ECSPI FIFOs:

- For transmitting data, software can enable the TXFIFO empty, TXFIFO data request, and TXFIFO full interrupts to maintain the TXFIFO using an interrupt service routine.
- For receiving data, software can enable the RXFIFO ready, RXFIFO data request, and RXFIFO full interrupts to retrieve data from the RXFIFO using an interrupt service routine.

Other interrupt sources can be used to control or debug the SPI bursts:

- The transfer-completed interrupt means that there is no data left in the TXFIFO and that the data in the Shift register has been shifted out.
- The RXFIFO overflow interrupt means that the RXFIFO received more than 64 words and will not accept any other words.

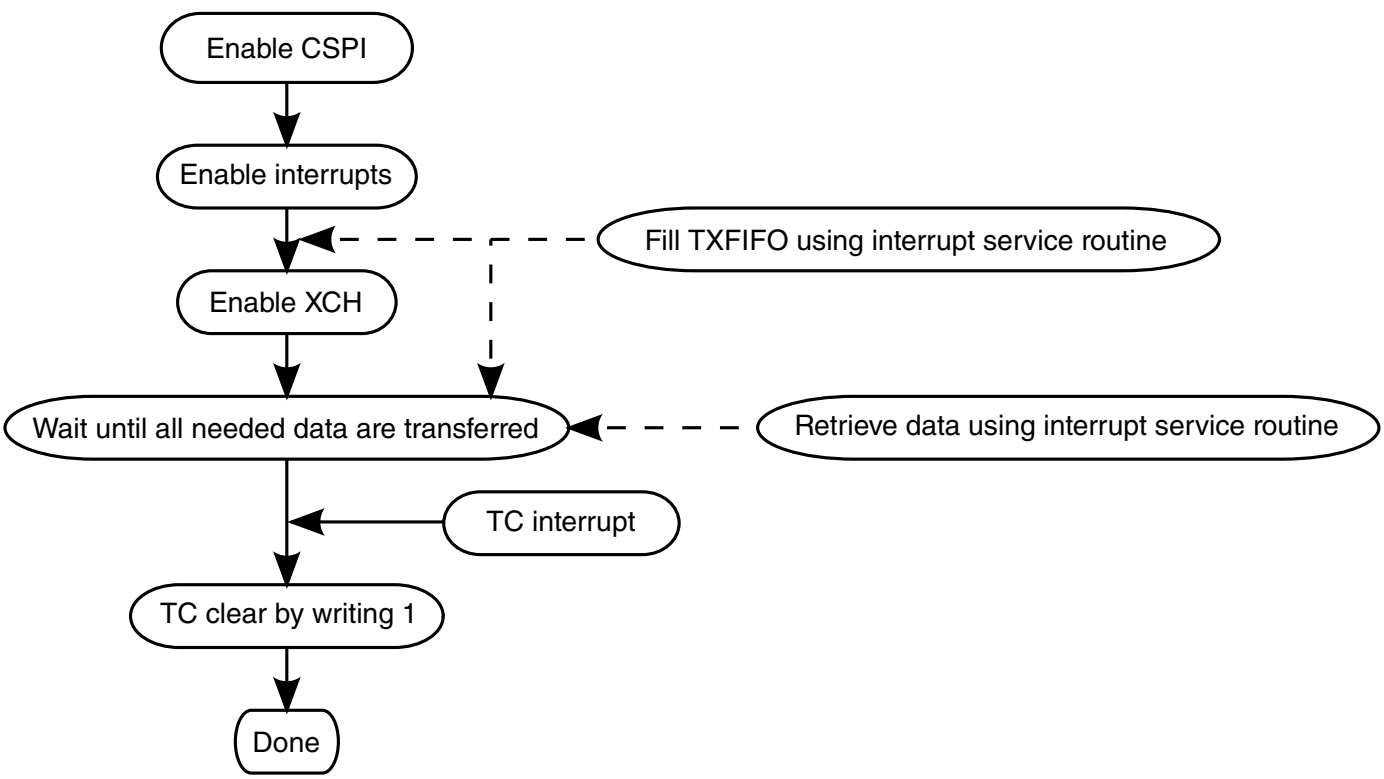


Figure 10-11. Program Sequence of SPI Burst Using Interrupt Control

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.3.7 DMA

DMA control provides another method to utilize the FIFOs in the ECSPI. By using DMA request and acknowledge signals, larger amounts of data can be transferred, and will reduce interrupts and host processor loading. When the appropriate conditions are matched, the block will send out a DMA request.

The DMA can deal with the following conditions:

- TXFIFO empty
- TXFIFO data request
- RXFIFO data request
- RXFIFO full

The figure below shows a program sequence of SPI bursts using DMA control.

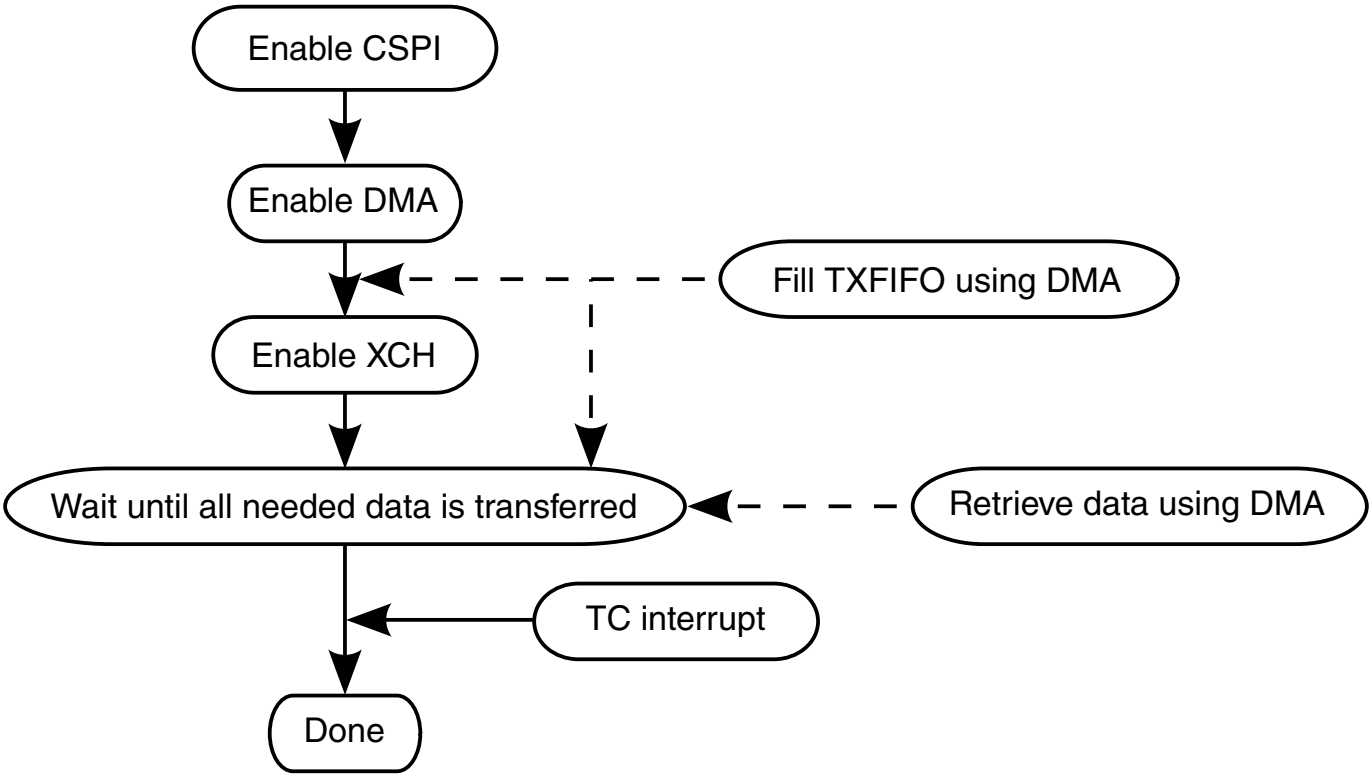


Figure 10-12. Program Sequence of SPI Burst Using DMA

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.3.8 Byte Order

The ECSPI does not support byte re-ordering in hardware.

10.1.4 Initialization

This section provides initialization information for ECSPI.

To initialize the block:

1. Clear the EN bit in ECSPI_CONREG to reset the block.

2. Enable the clocks for ECSPI within the CCM.
3. Configure the Control Register and then set the EN bit in the ECSPI_CONREG to put ECSPI out of reset.
4. Configure corresponding IOMUX for ECSPI external signals.
5. Configure registers of ECSPI properly according to the specifications of the external SPI device.

10.1.5 Applications

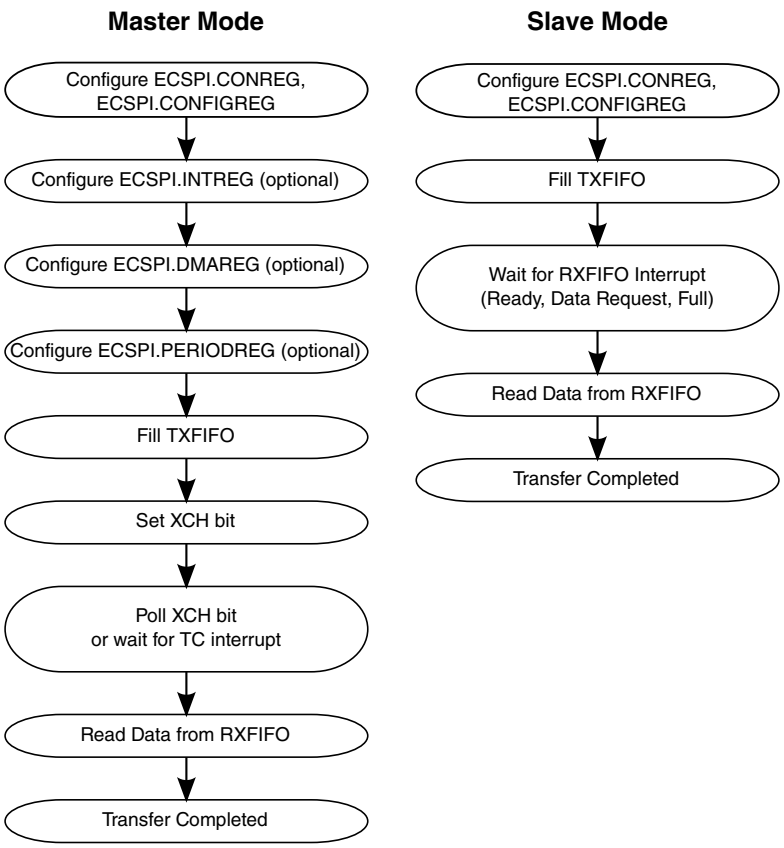


Figure 10-13. Flowchart of the ECSPI Operation

NOTE

The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete.

10.1.6 ECSPI Memory Map/Register Definition

This section includes the block memory map and detailed descriptions of all registers. For the base address of a particular block instantiation, see the system memory map.

ECSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3082_0000	Receive Data Register (ECSPI1_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2505
3082_0004	Transmit Data Register (ECSPI1_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2506
3082_0008	Control Register (ECSPI1_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2507
3082_000C	Config Register (ECSPI1_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2509
3082_0010	Interrupt Control Register (ECSPI1_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2511
3082_0014	DMA Control Register (ECSPI1_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2512
3082_0018	Status Register (ECSPI1_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2514
3082_001C	Sample Period Control Register (ECSPI1_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2515
3082_0020	Test Control Register (ECSPI1_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2517
3082_0040	Message Data Register (ECSPI1_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2518
3083_0000	Receive Data Register (ECSPI2_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2505
3083_0004	Transmit Data Register (ECSPI2_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2506
3083_0008	Control Register (ECSPI2_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2507
3083_000C	Config Register (ECSPI2_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2509
3083_0010	Interrupt Control Register (ECSPI2_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2511
3083_0014	DMA Control Register (ECSPI2_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2512
3083_0018	Status Register (ECSPI2_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2514

Table continues on the next page...

ECSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3083_001C	Sample Period Control Register (ECSPI2_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2515
3083_0020	Test Control Register (ECSPI2_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2517
3083_0040	Message Data Register (ECSPI2_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2518
3084_0000	Receive Data Register (ECSPI3_RXDATA)	32	R	0000_0000h	10.1.6.1/ 2505
3084_0004	Transmit Data Register (ECSPI3_TXDATA)	32	W	0000_0000h	10.1.6.2/ 2506
3084_0008	Control Register (ECSPI3_CONREG)	32	R/W	0000_0000h	10.1.6.3/ 2507
3084_000C	Config Register (ECSPI3_CONFIGREG)	32	R/W	0000_0000h	10.1.6.4/ 2509
3084_0010	Interrupt Control Register (ECSPI3_INTREG)	32	R/W	0000_0000h	10.1.6.5/ 2511
3084_0014	DMA Control Register (ECSPI3_DMAREG)	32	R/W	0000_0000h	10.1.6.6/ 2512
3084_0018	Status Register (ECSPI3_STATREG)	32	R/W	0000_0003h	10.1.6.7/ 2514
3084_001C	Sample Period Control Register (ECSPI3_PERIODREG)	32	R/W	0000_0000h	10.1.6.8/ 2515
3084_0020	Test Control Register (ECSPI3_TESTREG)	32	R/W	0000_0000h	10.1.6.9/ 2517
3084_0040	Message Data Register (ECSPI3_MSGDATA)	32	W	0000_0000h	10.1.6.10/ 2518

10.1.6.1 Receive Data Register (ECSPIx_RXDATA)

The Receive Data register (ECSPI_RXDATA) is a read-only register that forms the top word of the 64 x 32 receive FIFO. This register holds the data received from an external SPI device during a data transaction. Only word-sized read operations are allowed.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ECSPI_RXDATA																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_RXDATA field descriptions

Field	Description
ECSPI_RXDATA	Receive Data. This register holds the top word of the receive data FIFO. The FIFO is advanced for each read of this register. The data read is undefined when the Receive Data Ready (RR) bit in the Interrupt Control/Status register is cleared. Zeros are read when ECSPI is disabled.

10.1.6.2 Transmit Data Register (ECSPIx_TXDATA)

The Transmit Data (ECSPI_TXDATA) register is a write-only data register that forms the bottom word of the 64 x 32 TXFIFO. The TXFIFO can be written to as long as it is not full, even when the SPI Exchange bit (XCH) in ECSPI_CONREG is set. This allows software to write to the TXFIFO during a SPI data exchange process. Writes to this register are ignored when the ECSPI is disabled (ECSPI_CONREG[EN] bit is cleared).

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_TXDATA field descriptions

Field	Description
ECSPI_TXDATA	Transmit Data. This register holds the top word of data loaded into the FIFO. Data written to this register must be a word operation. The number of bits actually transmitted is determined by the BURST_LENGTH field of the corresponding SPI Control register. If this field contains more bits than the number specified by BURST_LENGTH, the extra bits are ignored. For example, to transfer 10 bits of data, a 32-bit word must be written to this register. Bits 9-0 are shifted out and bits 31-10 are ignored. When the ECSPI is operating in Slave mode, zeros are shifted out when the FIFO is empty. Zeros are read when ECSPI is disabled.

10.1.6.3 Control Register (ECSPIx_CONREG)

The Control Register (ECSPI_CONREG) allows software to enable the ECSPI , configure its operating modes, specify the divider value, and SPI_RDY control signal, and define the transfer length.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BURST_LENGTH												CHANNEL_ SELECT		DRCTL	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRE_DIVIDER				POST_DIVIDER				CHANNEL_MODE				SMC	XCH	HT	EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONREG field descriptions

Field	Description
31–20 BURST_ LENGTH	Burst Length. This field defines the length of a SPI burst to be transferred. The Chip Select (SS) will remain asserted until all bits in a SPI burst are shifted out. A maximum of 2^12 bits can be transferred in a single SPI burst. In master mode, it controls the number of bits per SPI burst. Since the shift register always loads 32-bit data from transmit FIFO, only the n least-significant (n = BURST LENGTH + 1) will be shifted out. The remaining bits will be ignored. Number of Valid Bits in a SPI burst. 0x000 A SPI burst contains the 1 LSB in a word. 0x001 A SPI burst contains the 2 LSB in a word. 0x002 A SPI burst contains the 3 LSB in a word. ... 0x01F A SPI burst contains all 32 bits in a word. 0x020 A SPI burst contains the 1 LSB in first word and all 32 bits in second word. 0x021 A SPI burst contains the 2 LSB in first word and all 32 bits in second word. ... 0xFFE A SPI burst contains the 31 LSB in first word and 2^7 -1 words. 0xFFF A SPI burst contains 2^7 words.
19–18 CHANNEL_ SELECT	SPI CHANNEL SELECT bits. Selects the external SPI Master/Slave Device. In master mode, these two bits select the external slave device by asserting the Chip Select (SS) output. 00 Channel 0 is selected. Chip Select (SS) will be asserted. 01 Reserved. 10 Reserved. 11 Reserved.
17–16 DRCTL	SPI Data Ready Control. This field selects the utilization of the SPI_RDY signal in master mode. ECSPI checks this field before it starts an SPI burst.

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
	00 The SPI_RDY signal is a don't care. 01 Burst will be triggered by the falling edge of the SPI_RDY signal (edge-triggered). 10 Burst will be triggered by a low level of the SPI_RDY signal (level-triggered). 11 Reserved.
15–12 PRE_DIVIDER	SPI Pre Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the pre-divider of the reference clock. 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 3. ... 1101 Divide by 14. 1110 Divide by 15. 1111 Divide by 16.
11–8 POST_DIVIDER	SPI Post Divider. ECSPI uses a two-stage divider to generate the SPI clock. This field defines the post-divider of the reference clock using the equation: 2^{n} . 0000 Divide by 1. 0001 Divide by 2. 0010 Divide by 4. ... 1110 Divide by 2^{14} . 1111 Divide by 2^{15} .
7–4 CHANNEL_ MODE	SPI CHANNEL MODE selects the mode for each SPI channel. CHANNEL MODE[3] is for SPI channel 3. CHANNEL MODE[2] is for SPI channel 2. CHANNEL MODE[1] is for SPI channel 1. CHANNEL MODE[0] is for SPI channel 0. 0 Slave mode. 1 Master mode.
3 SMC	Start Mode Control. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). It controls how the ECSPI starts a SPI burst, either through the SPI exchange bit, or immediately when the TXFIFO is written to. 0 SPI Exchange Bit (XCH) controls when a SPI burst can start. Setting the XCH bit will start a SPI burst or multiple bursts. This is controlled by the SPI SS Wave Form Select (SS_CTL). Refer to XCH and SS_CTL descriptions. 1 Immediately starts a SPI burst when data is written in TXFIFO.
2 XCH	SPI Exchange Bit. This bit applies only to channels configured in Master mode (CHANNEL MODE = 1). If the Start Mode Control (SMC) bit is cleared, writing a 1 to this bit starts one SPI burst or multiple SPI bursts according to the SPI SS Wave Form Select (SS_CTL). The XCH bit remains set while either the data exchange is in progress, or when the ECSPI is waiting for an active input if SPIRDY is enabled through DRCTL. This bit is cleared automatically when all data in the TXFIFO and the shift register has been shifted out. 0 Idle. 1 Initiates exchange (write) or busy (read).

Table continues on the next page...

ECSPIx_CONREG field descriptions (continued)

Field	Description
1 HT	Hardware Trigger Enable. This bit is used in master mode only. It enables hardware trigger (HT) mode. Note, HT mode is not supported by this product. 0 Disable HT mode. 1 Enable HT mode.
0 EN	SPI Block Enable Control. This bit enables the ECSPI. This bit must be set before writing to other registers or initiating an exchange. Writing zero to this bit disables the block and resets the internal logic with the exception of the ECSPI_CONREG. The block's internal clocks are gated off whenever the block is disabled. 0 Disable the block. 1 Enable the block.

10.1.6.4 Config Register (ECSPIx_CONFIGREG)

The Config Register (ECSPI_CONFIGREG) allows software to configure each SPI channel, configure its operating modes, specify the phase and polarity of the clock, configure the Chip Select (SS), and define the HT transfer length. Note, HT mode is not supported by this product.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			HT_LENGTH					SCLK_CTL				DATA_CTL				SS_POL				SS_CTL				SCLK_POL				SCLK_PHA			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_CONFIGREG field descriptions

Field	Description
31–29 -	This field is reserved. Reserved
28–24 HT_LENGTH	HT LENGTH. This field defines the message length in HT Mode. Note, HT mode is not supported by this product. The length in bits of one message is (HT LENGTH + 1).
23–20 SCLK_CTL	SCLK CTL. This field controls the inactive state of SCLK for each SPI channel. SCLK CTL[3] is for SPI channel 3. SCLK CTL[2] is for SPI channel 2. SCLK CTL[1] is for SPI channel 1. SCLK CTL[0] is for SPI channel 0. 0 Stay low. 1 Stay high.

Table continues on the next page...

ECSPIx_CONFIGREG field descriptions (continued)

Field	Description
19–16 DATA_CTL	DATA CTL. This field controls inactive state of the data line for each SPI channel. DATA CTL[3] is for SPI channel 3. DATA CTL[2] is for SPI channel 2. DATA CTL[1] is for SPI channel 1. DATA CTL[0] is for SPI channel 0. 0 Stay high. 1 Stay low.
15–12 SS_POL	SPI SS Polarity Select. In both Master and Slave modes, this field selects the polarity of the Chip Select (SS) signal. SS POL[3] is reserved. SS POL[2] is reserved. SS POL[1] is reserved. SS POL[0] is for SPI channel 0. 0 Active low. 1 Active high.
11–8 SS_CTL	SPI SS Wave Form Select. In master mode, this field controls the output wave form of the Chip Select (SS) signal when the SMC (Start Mode Control) bit is cleared. The SS_CTL bits are ignored if the SMC bit is set. SS CTL[3] is reserved. SS CTL[2] is reserved. SS CTL[1] is reserved. SS CTL[0] is for SPI channel 0. In slave mode, this bit controls when the SPI burst is completed. An SPI burst is completed by the Chip Select (SS) signal edges. (SSPOL = 0: rising edge; SSPOL = 1: falling edge) The RXFIFO is advanced whenever a Chip Select (SS) signal edge is detected or the shift register contains 32-bits of valid data. 0 In master mode - only one SPI burst will be transmitted. 1 In master mode - Negate Chip Select (SS) signal between SPI bursts. Multiple SPI bursts will be transmitted. The SPI transfer will automatically stop when the TXFIFO is empty. 0 In slave mode - an SPI burst is completed when the number of bits received in the shift register is equal to (BURST LENGTH + 1). Only the n least-significant bits (n = BURST LENGTH[4:0] + 1) of the first received word are valid. All bits subsequent to the first received word in RXFIFO are valid. 1 Reserved
7–4 SCLK_POL	SPI Clock Polarity Control. This field controls the polarity of the SCLK signal. See Figure 10-9 for more information. SCLK_POL[3] is for SPI channel 3. SCLK_POL[2] is for SPI channel 2. SCLK_POL[1] is for SPI channel 1. SCLK_POL[0] is for SPI channel 0.

Table continues on the next page...

ECSPIx_CONFIGREG field descriptions (continued)

Field	Description
	0 Active high polarity (0 = Idle). 1 Active low polarity (1 = Idle).
SCLK_PHA	SPI Clock/Data Phase Control. This field controls the clock/data phase relationship. See Figure 10-9 for more information. SCLK PHA[3] is for SPI channel 3. SCLK PHA[2] is for SPI channel 2. SCLK PHA[1] is for SPI channel 1. SCLK PHA[0] is for SPI channel 0. 0 Phase 0 operation. 1 Phase 1 operation.

10.1.6.5 Interrupt Control Register (ECSPIx_INTREG)

The Interrupt Control Register (ECSPI_INTREG) enables the generation of interrupts to the host processor. If the ECSPI is disabled, this register reads zero.

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TCEN	ROEN	RFEN	RDREN	RREN	TFEN	TDREN	TEEN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_INTREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TCEN	Transfer Completed Interrupt enable. This bit enables the Transfer Completed Interrupt. 0 Disable 1 Enable

Table continues on the next page...

ECSPIx_INTREG field descriptions (continued)

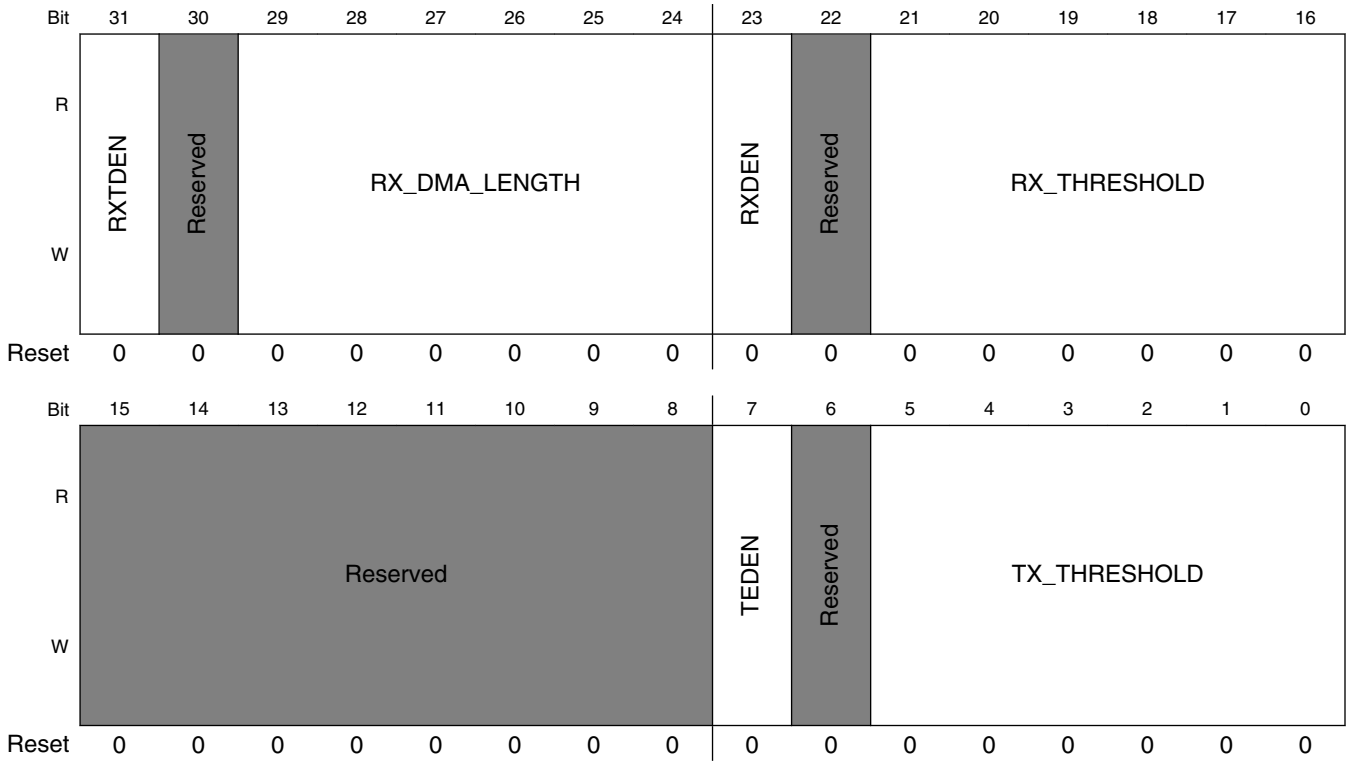
Field	Description
6 ROEN	RXFIFO Overflow Interrupt enable. This bit enables the RXFIFO Overflow Interrupt. 0 Disable 1 Enable
5 RFEN	RXFIFO Full Interrupt enable. This bit enables the RXFIFO Full Interrupt. 0 Disable 1 Enable
4 RDREN	RXFIFO Data Request Interrupt enable. This bit enables the RXFIFO Data Request Interrupt when the number of data entries in the RXFIFO is greater than RX_THRESHOLD. 0 Disable 1 Enable
3 RREN	RXFIFO Ready Interrupt enable. This bit enables the RXFIFO Ready Interrupt. 0 Disable 1 Enable
2 TFEN	TXFIFO Full Interrupt enable. This bit enables the TXFIFO Full Interrupt. 0 Disable 1 Enable
1 TDREN	TXFIFO Data Request Interrupt enable. This bit enables the TXFIFO Data Request Interrupt when the number of data entries in the TXFIFO is less than or equal to TX_THRESHOLD. 0 Disable 1 Enable
0 TEEN	TXFIFO Empty Interrupt enable. This bit enables the TXFIFO Empty Interrupt. 0 Disable 1 Enable

10.1.6.6 DMA Control Register (ECSPIx_DMAREG)

The Direct Memory Access Control Register (ECSPI_DMAREG) provides software a way to use an on-chip DMA controller for ECSPI data. Internal DMA request signals enable direct data transfers between the ECSPI FIFOs and system memory. The ECSPI sends out DMA requests when the appropriate FIFO conditions are matched.

If the ECSPI is disabled, this register is read as 0.

Address: Base address + 14h offset



ECSPIx_DMAREG field descriptions

Field	Description
31 RXTDEN	RXFIFO TAIL DMA Request/Interrupt Enable. This bit enables an internal counter that is increased at each read of the RXFIFO. This counter is cleared automatically when it reaches RX DMA LENGTH. If the number of words remaining in the RXFIFO is greater than or equal to RX DMA LENGTH, a DMA request/ interrupt is generated even if it is less than or equal to RX_THRESHOLD. 0 Disable 1 Enable
30 -	This field is reserved. Reserved
29–24 RX_DMA_ LENGTH	RX DMA LENGTH. This field defines the burst length of a DMA operation. Applies only when RXTDEN is set.
23 RXDEN	RXFIFO DMA Request Enable. This bit enables/disables the RXFIFO DMA Request. 0 Disable 1 Enable
22 -	This field is reserved. Reserved
21–16 RX_ THRESHOLD	RX THRESHOLD. This field defines the FIFO threshold that triggers a RX DMA/INT request. A RX DMA/INT request is issued when the number of data entries in the RXFIFO is greater than RX_THRESHOLD.
15–8 -	This field is reserved. Reserved

Table continues on the next page...

ECSPIx_DMAREG field descriptions (continued)

Field	Description
7 TEDEN	TXFIFO Empty DMA Request Enable. This bit enables/disables the TXFIFO Empty DMA Request. 0 Disable 1 Enable
6 -	This field is reserved. Reserved
TX_ THRESHOLD	TX THRESHOLD. This field defines the FIFO threshold that triggers a TX DMA/INT request. A TX DMA/INT request is issued when the number of data entries in the TXFIFO is not greater than TX_THRESHOLD.

10.1.6.7 Status Register (ECSPIx_STATREG)

The ECSPI Status Register (ECSPI_STATREG) reflects the status of the ECSPI's operating condition. If the ECSPI is disabled, this register reads 0x0000_0003.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								TC	RO	RF	RDR	RR	TF	TDR	TE
W									w1c	w1c						
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

ECSPIx_STATREG field descriptions

Field	Description
31–8 -	This field is reserved. Reserved
7 TC	Transfer Completed Status bit. Writing 1 to this bit clears it. **NOTE:** The TC bit does not provide a reliable indication that the transfer is complete. Under some conditions, the TC interrupt can occur before the transfer is completed. If the TC bit is used as an interrupt source, the XCH bit should be polled after the TC interrupt occurs to accurately confirm that the transfer is complete. 0 Transfer in progress. 1 Transfer completed.
6 RO	RXFIFO Overflow. When set, this bit indicates that RXFIFO has overflowed. Writing 1 to this bit clears it.

Table continues on the next page...

ECSPIx_STATREG field descriptions (continued)

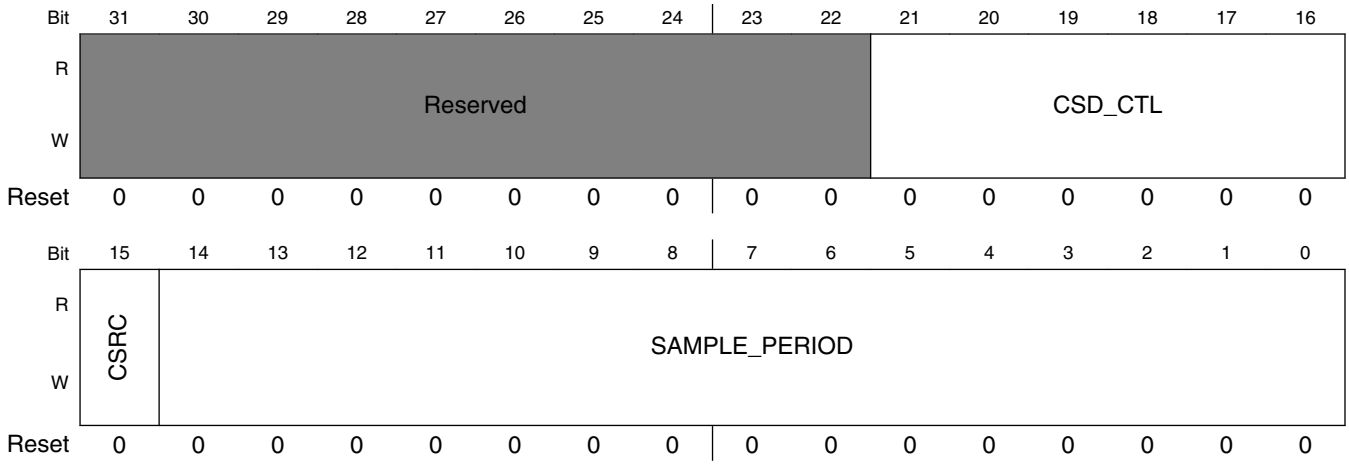
Field	Description
	0 RXFIFO has no overflow. 1 RXFIFO has overflowed.
5 RF	RXFIFO Full. This bit is set when the RXFIFO is full. 0 Not Full. 1 Full.
4 RDR	RXFIFO Data Request. 0 When RXTDE is set - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is set - Number of data entries in the RXFIFO is greater than RX_THRESHOLD or a DMA TAIL DMA condition exists. 0 When RXTDE is clear - Number of data entries in the RXFIFO is not greater than RX_THRESHOLD. 1 When RXTDE is clear - Number of data entries in the RXFIFO is greater than RX_THRESHOLD.
3 RR	RXFIFO Ready. This bit is set when one or more words are stored in the RXFIFO. 0 No valid data in RXFIFO. 1 More than 1 word in RXFIFO.
2 TF	TXFIFO Full. This bit is set when if the TXFIFO is full. 0 TXFIFO is not Full. 1 TXFIFO is Full.
1 TDR	TXFIFO Data Request. 0 Number of valid data slots in TXFIFO is greater than TX_THRESHOLD. 1 Number of valid data slots in TXFIFO is not greater than TX_THRESHOLD.
0 TE	TXFIFO Empty. This bit is set if the TXFIFO is empty. 0 TXFIFO contains one or more words. 1 TXFIFO is empty.

10.1.6.8 Sample Period Control Register (ECSPIx_PERIODREG)

The Sample Period Control Register (ECSPI_PERIODREG) provides software a way to insert delays (wait states) between consecutive SPI transfers. Control bits in this register select the clock source for the sample period counter and the delay count indicating the number of wait states to be inserted between data transfers.

The delay counts apply only when the current channel is operating in Master mode (ECSPI_CONREG[CHANNEL MODE] = 1).ECSPI_PERIODREG also contains the CSD CTRL field used to insert a delay between the Chip Select's active edge and the first SPI Clock edge.

Address: Base address + 1Ch offset



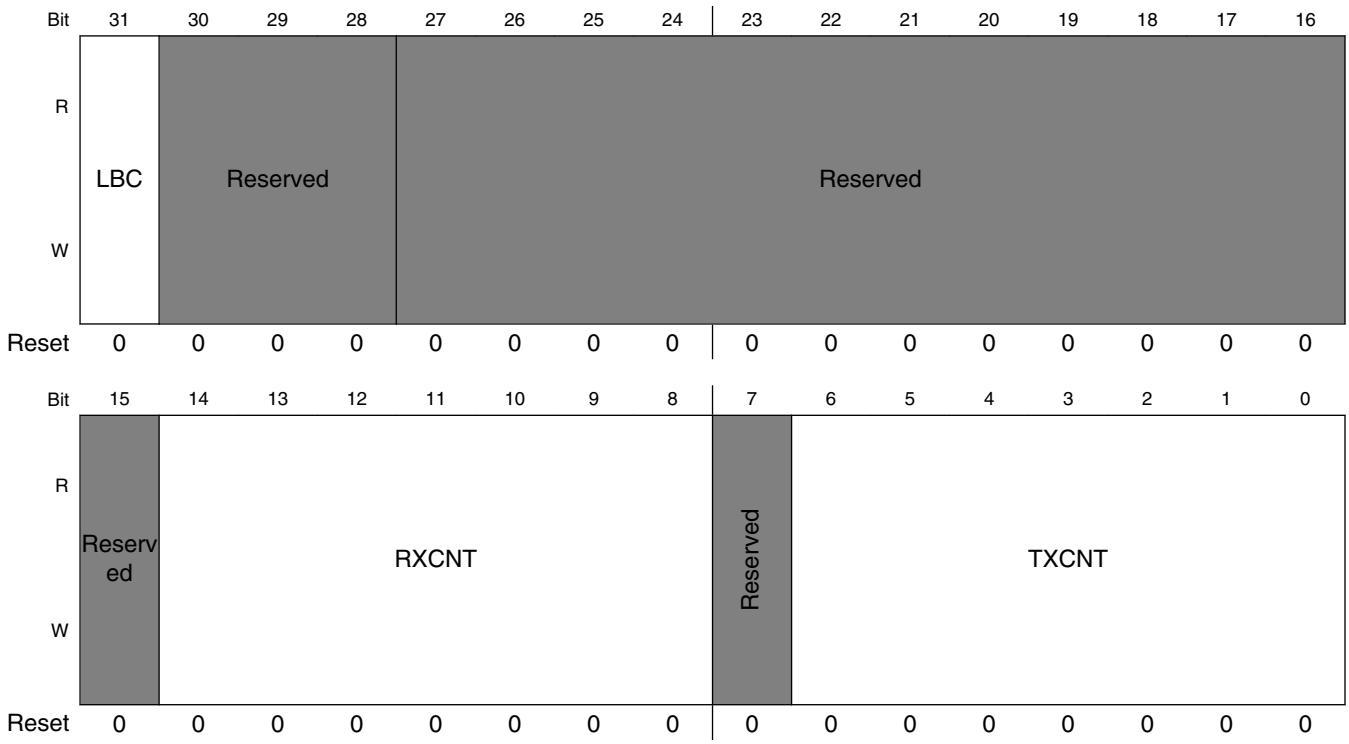
ECSPIx_PERIODREG field descriptions

Field	Description
31–22 -	This field is reserved. Reserved
21–16 CSD_CTL	Chip Select Delay Control bits. This field defines how many SPI clocks will be inserted between the chip select's active edge and the first SPI clock edge. The range is from 0 to 63.
15 CSRC	Clock Source Control. This bit selects the clock source for the sample period counter. 0 SPI Clock (SCLK) 1 Low-Frequency Reference Clock (32.768 KHz)
SAMPLE_ PERIOD	Sample Period Control. These bits control the number of wait states to be inserted in data transfers. During the idle clocks, the state of the SS output will operate according to the SS_CTL control field in the ECSPI_CONREG register. 0x0000 0 wait states inserted 0x0001 1 wait state inserted ... ... 0x7FFE 32766 wait states inserted 0x7FFF 32767 wait states inserted

10.1.6.9 Test Control Register (ECSPIx_TESTREG)

The Test Control Register (ECSPI_TESTREG) provides software a mechanism to internally connect the receive and transmit devices of the ECSPI, and monitor the contents of the receive and transmit FIFOs.

Address: Base address + 20h offset



ECSPIx_TESTREG field descriptions

Field	Description
31 LBC	Loop Back Control. This bit is used in Master mode only. When this bit is set, the ECSPI connects the transmitter and receiver sections internally, and the data shifted out from the most-significant bit of the shift register is looped back into the least-significant bit of the Shift register. In this way, a self-test of the complete transmit/receive path can be made. The output pins are not affected, and the input pins are ignored. 0 Not connected. 1 Transmitter and receiver sections internally connected for Loopback.
30–28 -	This field is reserved. Reserved, all bits should be ignored.
27–15 -	This field is reserved. Reserved
14–8 RXCNT	RXFIFO Counter. This field indicates the number of words in the RXFIFO.

Table continues on the next page...

ECSPIx_TESTREG field descriptions (continued)

Field	Description
7 -	This field is reserved. Reserved
TXCNT	TXFIFO Counter. This field indicates the number of words in the TXFIFO.

10.1.6.10 Message Data Register (ECSPIx_MSGDATA)

The Message Data Register (ECSPI_MSGDATA) forms the top word of the 16 x 32 MSG Data FIFO. Only word-size accesses are allowed for this register. Reading from this register returns zero, and writing to this register stores data in the MSG Data FIFO.

Address: Base address + 40h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																																
W	ECSPI_MSGDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ECSPIx_MSGDATA field descriptions

Field	Description
ECSPI_ MSGDATA	ECSPI_MSGDATA holds the top word of MSG Data FIFO. The MSG Data FIFO is advanced for each write of this register. The data read is zero. The data written to this register is stored in the MSG Data FIFO.

10.2 Quad Serial Peripheral Interface (QuadSPI)

10.2.1 Overview

The Quad Serial Peripheral Interface (QuadSPI) block acts as an interface to one or two external serial flash devices, each with up to four bidirectional data lines. The following figure is a block diagram of the QuadSPI module.

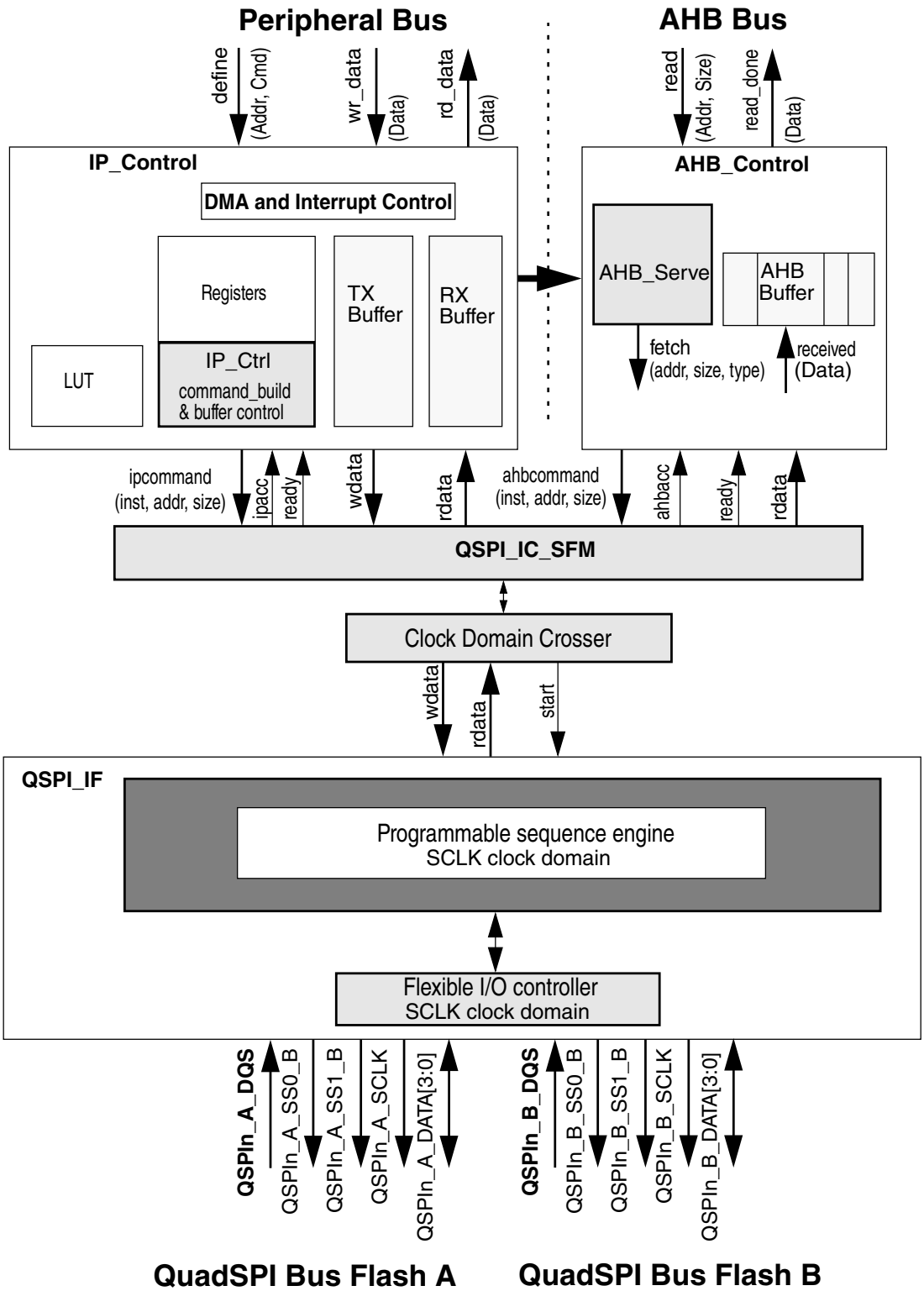


Figure 10-14. QuadSPI Block Diagram

The following figure describes the serial flash clock diagram in QuadSPI:

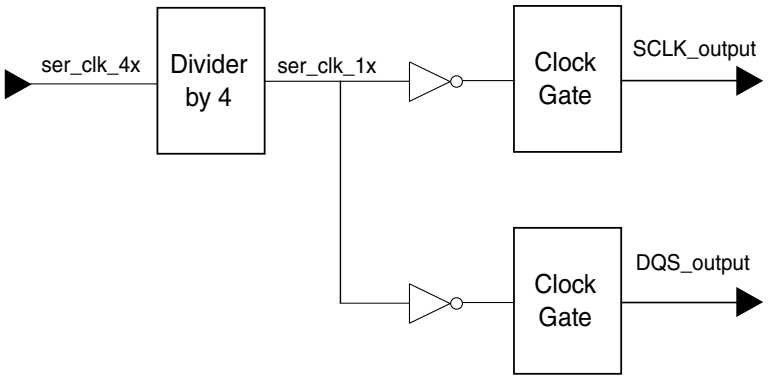


Figure 10-15. Flash Clock Diagram

ser_clk_4x is the serial clock root from CCM

ser_clk_1x' is generated clock from 'ser_clk_4x' divided by 4

SCLK_output is serial output clock generated from 'ser_clk_1x' with inverter and clock gate, it's used as Clock by external serial flash device.

DQS_output is serial flash data strobe signal generated from 'ser_clk_1x' with inverter and clock gate. This signal could be loopback from pad and used as sampling clock for serial flash data.

10.2.1.1 Features

The QuadSPI supports the following features:

- Flexible sequence engine to support various flash vendor devices.
- Single, dual, quad mode of operation.
- DDR/DTR mode wherein the data is generated on every edge of the serial flash clock.
- Support for flash data strobe signal for data sampling in DDR and SDR mode.
- Two identical serial flash devices can be connected and accessed in parallel for data read operations, forming one (virtual) flash memory with doubled readout bandwidth.
- DMA support to read RX Buffer data via AMBA AHB bus (64-bit width interface) or IP registers space (32-bit access).
    - Inner loop size of DMA access can be configured.
- Multi master accesses with priority

    - Flexible and configurable buffer for each master
- Thirteen interrupt conditions (see Table 10-10)
- Memory mapped read access to connected flash devices.
- Programmable sequence engine to cater to future command/protocol changes and able to support all existing vendor commands and operations.
    - Supports 3-byte and 4-byte addressing.
    - TXFIFO size is 512Byte
    - RXFIFO size is 512Byte
    - AHB BUF size is 1KByte

10.2.1.2 QuadSPI Modes of Operation

This section provides information about the modes in which the QuadSPI module can be used.

10.2.1.2.1 Normal Mode

In this mode, one or two external serial flash memory devices can be accessed. Further details about this mode of operation can be found in Modes of Operation (Normal Mode).

10.2.1.2.2 Module Disable Mode

This mode is used for power management of the device containing the QuadSPI module. It is controlled by signals external to the QuadSPI. The clock to the non-memory mapped logic in the QuadSPI can be stopped while in Module Disable Mode.

10.2.1.3 Acronyms and Abbreviations

The following table contains acronyms and abbreviations used in this document.

Table 10-2. Acronyms and Abbreviations

Terms	Description
AHB	Advanced High-performance Bus, version of AMBA
AMBA	Advanced Microcontroller Bus Architecture
BE	Big Endian Byte Ordering
CS	Chip Select
DMA	Direct Memory Access

Table continues on the next page...

Table 10-2. Acronyms and Abbreviations (continued)

Terms	Description
MB	Megabyte. Each MB is 1024 * 1024 bytes
IFM	Individual Flash Mode
PFM	Parallel Flash Mode
LSB	Least Significant Bit
MSB	Most Significant Bit
PCS	Peripheral Chip Select
QSPI, QuadSPI	Quad Serial Peripheral Interface
SCK	Serial Communications Clock
w1c	Write 1 to clear, writing a 1 to this field resets the flag

10.2.1.4 Glossary for QuadSPI module

Table 10-3. Glossary

Term	Definition
AHB Command	An AHB Command is a SFM Command triggered by a read access to the address range belonging to the memory mapped access defined in Table 10-7. Refer to AHB Commands for details.
Asserted	A signal that is asserted is in its active state. An active low signal changes from logic level one to logic level zero when asserted, and an active high signal changes from logic level zero to logic level one.
Clear	To clear a bit or bits means to establish logic level zero on the bit or bits.
Clock Phase	Determines when the data should be sampled relative to the active edge of SCK
Clock Polarity	Determines the idle state of the SCK signal.
Drain	To remove entries from a FIFO by software or hardware.
Endianness	Byte Ordering scheme.
Field	Two or more register bits grouped together.
Fill	To add entries to a FIFO by software or hardware.
Host	Refers to another functional block in the device containing the QuadSPI module
Instruction Code	8 bits defining the type of command to be executed.
IP Command	An IP Command is a SFM Command triggered by writing into the QSPI_IPCR[SEQID] field.
Logic level one	The voltage that corresponds to Boolean true (1) state.
Logic level zero	The voltage that corresponds to Boolean false (0) state.
Negated	A signal that is negated is in its inactive state. An active low signal changes from logic level 0 to logic level 1 when negated, and an active high signal changes from logic level 1 to logic level 0.
QSPI_AMBA_BASE	First address of QuadSPI address space on system memory map.
QSPI_ARDB_BASE	First address of QuadSPI Rx Buffer on system memory map.
Set	To set a bit or bits means to establish logic level one on the bit or bits.

Table continues on the next page...

Table 10-3. Glossary (continued)

Term	Definition
RX Buffer PUSH Event	Addition of valid entries into the RX Buffer. In the default case each Buffer PUSH Event adds 2 entries to the RX Buffer since the interface to the serial clock domain is 64 bits in width. Depending on the number of bytes read from the serial flash device it is possible for the very last Buffer PUSH Event that only one entry is added. The QSPI_RBSR[RDBFL] field is incremented by the number of entries added to the RX Buffer.
RX Buffer POP Event	Removal of valid entries from the RX Buffer. Each Buffer POP Event removes (QSPI_RBCT[WMRK] + 1) valid entries from the buffer. The QSPI_RBSR[RDBFL] field is decremented by the same number and the QSPI_RBSR[RDCTR] field is incremented accordingly.
Individual Flash Mode	Access to a single, individual serial flash device. Refer to Serial Flash Access Schemes for details.
Parallel Flash Mode	Read access to two serial flash devices attached to the QuadSPI module in parallel. Refer to Serial Flash Access Schemes for details.
SFM Command	Serial Flash Memory Command. A SFM command consists of an instruction code and all other parameters (for example, size or mode bytes) needed for that specific instruction code. Triggering a command either initiates a transaction on the external serial flash or results in an error. Refer to Table 10-20 for details on errors.
Single/Dual/Quad Instructions	Depending on the serial flash device connected to the QuadSPI module there will be instructions using a different number of data lines. • Single: Single line I/O with one data out and one data in line to/from the serial flash device. • Dual: Dual line I/O with two bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI module • Quad: Quad line I/O with 4 bidirectional I/O lines, driven alternatively by the serial flash device or the QuadSPI
Transaction	A transaction consists of all flags, data and signals in either direction to execute a command for an attached serial flash device. It is a combination of chip select, sclk, instruction code, address, mode- and/or dummy bytes, transmit and/or receive data.
LUT	Look-up table.

10.2.2 Memory Map and Register Definition

This section provides the memory map and register definitions of the QuadSPI module.

10.2.2.1 Register Write Access

This section describes the write access restriction terms that apply to all registers, which can be one of the following:

- **Register Write Access Restriction**

For each register bit and register field, the write access conditions are specified in the detailed register description. A description of the write access conditions is given in the following table. If, for a specific register bit or field, none of the given write access conditions is fulfilled, any write attempt to this register bit or field is ignored without any notification. The values of the bits or fields are not changed.

The condition term [A or B] indicates that the register or field can be written to if at least one of the conditions is fulfilled.

Table 10-4. Register Write Access Restrictions

Condition	Description
Anytime	No write access restriction.
Disabled Mode	Write access only if *QSPI_MCR[MDIS] = 1*.
Normal Mode	Write access only if the module is in *Normal Mode*.

- **Register Write Access Requirements**

  All registers can be accessed with 8-bit, 16-bit, and 32-bit wide operations. For some of the registers, at least a 16/32-bit wide write access is required to ensure correct operation. This write access requirement is stated in the detailed register description for each register affected.

10.2.2.2 Serial Flash Address Assignment

The serial flash address assignment may be modified by writing into Serial Flash A1 Top Address (QuadSPI_SFA1AD) and Serial Flash A2 Top Address (QuadSPI_SFA2AD) for device A and into Serial Flash B1Top Address (QuadSPI_SFB1AD) and Serial Flash B2Top Address (QuadSPI_SFB2AD) for device B. The following table shows how different access modes are related to the address specified for the next SFM Command. Note that this address assignment is valid for both IP and AHB commands.

Table 10-5. Serial Flash Address Assignment

Parameter	Function	Access Mode
QSPI_AMBA_BASE ((31:10) - 22 bits)	QuadSPI AHB base address	
TOP_ADDR_MEMA1(TPADA1)	Top address for the external flash A1 (first device of the dual die flash A, or the first of the two independent flashes sharing the IOFA)	Any access to the address space between TOP_ADDR_MEMA1 and QSPI_AMBA_BASE will be routed to **Serial Flash A1**

Table continues on the next page...

Table 10-5. Serial Flash Address Assignment (continued)

Parameter	Function	Access Mode
TOP_ADDR_MEMA2(TPADA2)	Top address for the external flash A2 (second device of the dual die flash A, or the second of the two independent flashes sharing the IOFA).	Any access to the address space between TOP_ADDR_MEMA2 and TOP_ADDR_MEMA1 will be routed to **Serial Flash A2**
TOP_ADDR_MEMB1(TPADB1)	Top address for the external flash B1 (first device of the dual die flash B, or the first of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB1 and TOP_ADDR_MEMA2 will be routed to **Serial Flash B1**
TOP_ADDR_MEMB2(TPADB2)	Top address for the external flash B2 (second device of the dual die flash B or the second of the two independent flashes sharing the IOFB)	Any access to the address space between TOP_ADDR_MEMB2 and TOP_ADDR_MEMB1 will be routed to **Serial Flash A2**

10.2.2.3 AMBA Bus Register Memory Map

QSPI_AMBA_BASE defines the address to be used as start address of the serial flash device as defined by the system memory map..

Table 10-6. QuadSPI AMBA Bus Memory Map

Address	Register Name
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A	
QSPI_AMBA_BASE to (TOP_ADDR_MEMA2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A for details and to Table 10-14 and Table 10-18 for information about the byte ordering.
Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B	
TOP_ADDR_MEMA2 to (TOP_ADDR_MEMB2 - 0x01)	Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B Refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B for details and to Table 10-14 and Table 10-18 for information about the byte ordering.
Parallel Flash Mode	
QSPI_AMBA_BASE to (TOP_ADDR_MEMB2 - 0x01)	Parallel Flash Mode Refer to Parallel Flash Mode for details and to Table 10-17 and Table 10-18 for information about the byte ordering.
AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)	
QSPI_ARDB_BASE to… (32 * 4 Byte) QSPI_ARDB_BASE + 0x0000_01FF	AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) Refer to Table 10-14 and Table 10-16 for information about the byte ordering.

Note

Any read access to non-implemented addresses will provide undefined results.

In case single die flash devices, TOP_ADDR_MEMA2 and TOP_ADDR_MEMB2 should be initialized/programmed to TOP_ADDR_MEMA1 and TOP_ADDR_MEMB1 respectively- in effect, setting the size of these devices to 0. This would ensure that the complete memory map is assigned to only one flash device.

Parallel Flash Mode is valid only for commands related to data read from the serial flash. The first device of flash A has to be paired with the first device of flash B and the second device of flash A has to be paired with the second device of flash B in parallel mode. Parallel mode is selected via the QSPI_BFGENCR[PAR_EN] bit for all masters in AHB driven mode and via the QSPI_IPCR[PAR_EN] in IP driven mode. In parallel mode, the incoming address (SFAR address in case of IP initiated transactions and the incoming AHB address in case of AHB initiated transactions) is divided by 2 and sent to the two flashes connected in parallel.

Any IP Command other than data read in Parallel Flash Mode will result in the assertion of the QSPI_FR[IUEF] flag and any AHB Command other than data read in Parallel Flash Mode will result in the assertion of the QSPI_FR[ABSEF] flag.

In the Individual Flash Modes, the 3/4 address bytes (as programmed in the instruction/operand in the sequence) available for the flash address is determined by SFADR [23:0] or SFADR [31:0]as given in the table above.

In Parallel Flash Mode, both flashes are read with the same starting address of 3/4 (as programmed in the instruction/ operand in the sequence) bytes in size. This address is derived from SFADR [24:1] or SFADR [31:1] as given in the table above. The LSB of the SFADR field is used to select the appropriate bits of both flash devices to combine the byte corresponding to the selected address.

10.2.2.4 AHB Bus Register Memory Map Descriptions

This chapter contains definitions of registers in the AMBA address space.

10.2.2.4.1 AHB Bus Access Considerations

It has to be noted that all logic in the QuadSPI module implementing the AHB Bus access is designed to read the content of an external serial flash device. Therefore the following restrictions apply to the QuadSPI module with respect to accesses to the AHB bus:

- Any write access is answered with the ERROR condition according to the AMBA AHB Specification. No write occurs.
- Any AHB Command resulting in the assertion of the QSPI_FR[ABSEF] flag is answered with the ERROR condition according to the AMBA_AHB specification. The resulting AHB Command is ignored.
- AHB Bus access types fully supported are NONSEQ and BUSY.
- AHB access type SEQ is treated in the same way like NONSEQ. Refer to the AMBA AHB Specification for further details.

10.2.2.4.2 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A

Starting with address QSPI_AMBA_BASE the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address QSPI_AMBA_BASE with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 10-14 and for 64 bit read access the byte ordering is given in Table 10-18.

Table 10-7. Memory Mapped Individual Flash Mode - Flash A Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
QSPI_AMBA_BASE + 0x00	QSPI_AMBA_BASE + 0x00	0x00_0000 to 0x00_0003	A1
QSPI_AMBA_BASE + 0x04		0x00_0004 to 0x00_0007	
…	…	…	
TOP_ADDR_MEMA1 - 0x08	TOP_ADDR_MEMA1 - 0x08	(TOP_ADDR_MEMA1- 0x08) to (TOP_ADDR_MEMA1 - 0x04 -0x01)	
TOP_ADDR_MEMA1 - 0x04		(TOP_ADDR_MEMA1 - 0x04) to (TOP_ADDR_MEMA1 - 0x01)	

Table continues on the next page...

Table 10-7. Memory Mapped Individual Flash Mode - Flash A Address Scheme (continued)

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
TOP_ADDR_MEMA1 + 0x00	TOP_ADDR_MEMA1 + 0x00_0000	0x00_0000 to 0x00_0003	A2
TOP_ADDR_MEMA1 + 0x04		0x00_0004 to 0x00_0007	
…..	…	…	
TOP_ADDR_MEMA2 - 0x08	TOP_ADDR_MEMA2 - 0x08	(TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMA2 - 0x04 - 0x01)	
TOP_ADDR_MEMA2 - 0x04		(TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMA2 - 0x01)	

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

10.2.2.4.3 Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B

Starting with address TOP_ADDR_MEMA2 the content of the first external serial flash devices is mapped into the address space of the device containing the QuadSPI module. Serial flash address byte address 0x0 corresponds to bus address TOP_ADDR_MEMA2 with increasing order. Assuming that a dual-die flash is connected on the first set of external pads, the address space is divided into two parts, one for each device of the dual die package. Refer the following table for the address mapping. The byte ordering for 32 bit access is given in Table 10-14 and for 64 bit read access the byte ordering is given in Table 10-18.

Table 10-8. Memory Mapped Individual Flash Mode - Flash B Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
TOP_ADDR_MEMA2 + 0x00	TOP_ADDR_MEMA2 + 0x00	0x00_0000 to 0x00_0003	B1
TOP_ADDR_MEMA2 + 0x04		0x00_0004 to 0x00_0007	
…	…	…	
TOP_ADDR_MEMB1 - 0x08	TOP_ADDR_MEMB1 - 0x08	(TOP_ADDR_MEMB1- TOP_ADDR_MEMA2 - 0x08) to (TOP_ADDR_MEMB1 - TOP_ADDR_MEMA2 - 0x04 -0x01)	
TOP_ADDR_MEMB1 - 0x04		(TOP_ADDR_MEMB1- TOP_ADDR_MEMA2 - 0x04) to (TOP_ADDR_MEMB1- TOP_ADDR_MEMA2- 0x01)	
TOP_ADDR_MEMB1 + 0x00	TOP_ADDR_MEMB1 + 0x00_0000	0x00_0000 to 0x00_0003	B2

Table continues on the next page...

Table 10-8. Memory Mapped Individual Flash Mode - Flash B Address Scheme (continued)

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash Byte Address	Flash Device
TOP_ADDR_MEMB1 + 0x04		0x00_0004 to 0x00_0007	
…..	…	…	
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMA2 - 0x08	(TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1- 0x08) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x04 - 0x01)	
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2- TOP_ADDR_MEMB1 - 0x04) to (TOP_ADDR_MEMB2 - TOP_ADDR_MEMB1 - 0x01)	

The available address range depends from the size of the external serial flash device. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

10.2.2.4.4 Parallel Flash Mode

Any of the AHB flexible-buffers can be configured to work in parallel flash mode by programming the QSPI_BFGENCR[PAR_EN] bit to '1'. When parallel mode is set, Flash A1 is paired with Flash B1 and Flash A2 is paired with Flash B2. In parallel mode, software should ensure that the size of Flash A1(A2) is equal to the size of Flash B1(B2).

Reads from any even AHB bus address provides bits [7:4] of both serial flash devices and reads from any odd AHB bus address provides bits [3:0] of both flash devices. Refer to the following table for the address mapping. The byte ordering for 32 bit access is given in Table 10-16 and for 64 bit read access the byte ordering is given in Table 10-18.

Table 10-9. Memory Mapped Parallel Flash Mode Address Scheme

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash A Byte Address	Serial Flash B Byte Address
QSPI_AMBA_BASE + 0x0000_0000	QSPI_AMBA_BASE + 0x00 For details, please refer to Parallel mode and Dual Die Flashes.	0x00_0000 - 0x00_0001	0x00_0000 - 0x00_0001
QSPI_AMBA_BASE + 0x0000_0004		0x00_002 - 0x00_0003	0x00_0002 - 0x00_0003
QSPI_AMBA_BASE + 0x0000_0008	QSPI_AMBA_BASE + 0x08	0x00_0004 - 0x00_0005	0x00_0004 - 0x00_0005

Table continues on the next page...

Table 10-9. Memory Mapped Parallel Flash Mode Address Scheme (continued)

Memory Mapped Address 32 Bit Access	Memory Mapped Address 64 Bit Access	Serial Flash A Byte Address	Serial Flash B Byte Address
QSPI_AMBA_BASE + 0x0000_000C		0x00_0006 - 0x00_0007	0x00_0006 - 0x00_0007
…	…	…	…
TOP_ADDR_MEMB2 - 0x08	TOP_ADDR_MEMB2 - 0x08	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x08)/2
TOP_ADDR_MEMB2 - 0x04		(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE - 0x04)/2 +0x01	(TOP_ADDR_MEMB2 - QSPI_AMBA_BASE- 0x04)/2 + 0x01

The available address range covers 27 address bits, corresponding to 128 MB per flash device. The usable space depends from the size of the external serial flash devices. Any access beyond the size of the external serial flash provides undefined results.

For details concerning the read process refer to Flash Read.

10.2.3 Interrupt Signals

The interrupt request lines of the QuadSPI module are mapped to the internal flags according to the following table.

Table 10-10. Assignment of Interrupt Request Lines

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_tfff	TBFF	TX Buffer Fill
ipi_int_tcf	TFF	Peripheral Command Transaction Finished
ipi_int_rfdf	RBDF	RX Buffer Drain
ipi_int_overrun		Buffer Overflow/Underrun Error Logical OR from:
	RBOF	RX Buffer Overrun
	TBUF	TX Buffer Underrun
	ABOF	AHB Buffer Overflow
ipi_int_cerr		Serial Flash Command Error Logical OR from:
	IPAEF	Peripheral access while AHB busy Error
	IPIEF	Peripheral Command could not be triggered Error
	IPGEF	Peripheral access while AHB Grant Error
	IUEF	Peripheral Command Usage Error

Table continues on the next page...

Table 10-10. Assignment of Interrupt Request Lines (continued)

IRQ/DMA line	QSPI_FR Flag	Interrupt Description
ipi_int_ored	DLPFF, TBFF, TFF, ILLINE, RBDF, RBOF, TBUF, ABSEF, ABOF, IPAEF, IPIEF, IPGEF, IUEF	Logical OR from all the QSPI_FR flags mentioned

10.2.4 Functional Description

This section provides the functional information of the QuadSPI module.

10.2.4.1 Serial Flash Access Schemes

The Quad Serial Peripheral Interface (QuadSPI) block acts as an interface to one single or two external serial flash devices, each with up to 4 bidirectional data lines. Depending from the serial flash devices attached to the QuadSPI module the following access schemes are possible:

Table 10-11. Access Schemes for Serial Flash Data Access

Access Scheme	One Flash Device on Port A	One Flash Device on Port B	Two identical Flash Devices connected on Port A and Port B
Individual Flash Mode: Access to Flash A	**Yes**	**N/a**	**Yes**
Individual Flash Mode: Access to Flash B	**N/a**	**Yes**	**Yes**
Parallel Flash Mode: Read from Flash A and Flash B	**N/a**	**N/a**	**Yes**

Note

If two flash devices are accessed in Parallel Flash Mode, they are accessed with identical control signals. Special alignment on per-flash basis is **not** possible. It is within the responsibility of the application to ensure that the identical signals are applicable to both flash devices.

In Parallel Flash Mode, both external serial flash devices appear logically as one single memory doubled in size with respect to one individual flash device.

If two different flash devices are attached, they can be operated only in Individual Flash Mode.

In the Parallel Flash Mode, only data read commands are supported. Any other IP Command will result in an error condition signaled by the assertion of the QSPI_FR[IUEF] flag and any other AHB Command will result in the assertion of the QSPI_FR[ABSEF] flag.

In the Individual Flash Mode, all supported commands are available.

Unless explicitly noted, all the following descriptions relate to the Individual Flash Mode.

10.2.4.2 Modes of Operation

Refer to QuadSPI Modes of Operation for an overview over the possible operational modes of the QuadSPI block.

- Normal Mode can be used for write or read accesses to an external serial flash device.
  - Serial Flash Write: Data can be programmed into the flash via the IP interface only. Refer to Flash Programming for further details.
  - Serial Flash Read: Read the contents of the serial flash device. Two separate read channels are available via RX Buffer and AHB Buffer, see Flash Read.
- Stop Mode: The mode is used for power management. When a request is made to enter Stop Mode, the QuadSPI block acknowledges the request and completes the SFM Command in progress, then the system clocks to the QuadSPI block may be shut off
- Module Disable Mode: The mode is used for power management. The clock to the non-memory mapped logic in the QuadSPI can be stopped while in Module Disable Mode. The module enters the mode by setting QSPI_MCR[MDIS].

10.2.4.3 Normal Mode

This mode is used to allow communication with an external serial flash device. Compared to the standard SPI protocol, this communication method uses up to 4 bidirectional data lines operating at high data rates. The communication to the external serial flash device consists of an instruction code and optional address, mode, dummy

and data transfers. The flexible programmable core engine described below is immune to a wide variety of command/protocol differences in the serial flash devices provided by various flash vendors.

10.2.4.3.1 Programmable Sequence Engine

The core of the QuadSPI module is a programmable sequence engine that works on "instruction-operand" pairs. The core controller executes each programmed instruction sequentially. The complete list of instructions and the corresponding operands is given in the following table.

Table 10-12. Instruction set

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
CMD	6'd1	N=2'd{0,1,2} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads	8 bit command value	Provide the serial flash with operand on the number of pads specified
ADDR	6'd2		Number of address bits to be sent (for example, 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads specified. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of SFAR in case of IPS initiated transactions .
DUMMY	6'd3		Number of dummy clock cycles (should be <= 64 cycles)	Provide the serial flash with dummy cycles as per the operand. The PAD information defines the number of pads in input mode. (for example, one pad implies that pad 1 is not driven, rest all are driven)
MODE	6'd4		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified
MODE2	6'd5	N=2'd{0,1}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads[1] specified
MODE4	6'd6	N=2'd{0,1,2}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads[2] specified
READ	6'd7	N=2'd{0,1,2} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads	Read data size in bytes (for AHB transactions, the user's application should ensure that data size is a multiple of 8 bytes)	Read data from flash on the number of pads specified. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions.
WRITE	6'd8		Write data size in bytes	Write data on number of pads sepcified. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register

Table continues on the next page...

Table 10-12. Instruction set (continued)

Instruction	Instruction encoding	Pins	Operand	Action on Serial Flash(es)
JMP_ON_CS	6'd9	NA	Instruction number	Every time the CS is deasserted, jump to the instruction pointed to by the operand. This instruction allows the programmer to specify the behavior of the controller when a new read transaction is initiated following a CS deassertion.
ADDR_DDR	6'd10	N=2'd{0,1,2} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads	Number of address bits to be sent (for example, 8'd24 => 24 address bits required)	Provide the serial flash with address cycles according to the operand on the number of pads specified at each clock edge of serial flash clock. The actual address to be provided will be derived from the incoming address in case of AHB initiated transactions and the value of QSPI_SFAR in case of IPS initiated transactions .
MODE_DDR	6'd11		8 bit mode value	Provide the serial flash with 8 bit operand on the number of pads specified at each clock edge of serial flash.
MODE2_DDR	6'd12	N=2'd{0}	2 bit mode value	Provide the serial flash with 2 bit operand on the number of pads specified at each clock edge of serial flash[3]
MODE4_DDR	6'd13	N=2'd{0,1}	4 bit mode value	Provide the serial flash with 4 bit operand on the number of pads specified at each clock edge of serial flash[4].
READ_DDR	6'd14	N=2'd{0,1,2} 2'd0 - One pad 2'd1 - Two pads 2'd2 - Four pads	Read data size in bytes (for AHB transactions, the user's application should ensure that data size is in multiple of 8 bytes)	Read data from flash on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the ADATSZ field of the QSPI_BUFxCR registers for AHB initiated transactions and IDATSZ field of IP Configuration Register (QuadSPI_IPCR) for IP initiated transactions
WRITE_DDR	6'd15		Write data size in bytes	Write data on the number of pads specified at each clock edge of serial flash. The data size may be overwritten by writing to the IDATSZ field of IP Configuration Register (QuadSPI_IPCR) register
DATA_LEARN[5]	6'd16		8 bit Data learning pattern	Find the correct sampling point with the data learning pattern. When this instruction is encountered, the QSPI_SMPR[DDRSMP] values are ignored and the controller finds the correct sampling point on its own by sampling the data learning pattern.[6]
STOP	8'd0	NA	NA	Stop execution; deassert CS

1. For a one pad instruction, MODE2 will take 2 serial flash clock cycles on the flash interface.
2. For a one pad instruction, MODE4 will take 4 serial flash clock cycles on the flash interface. For a 4 pad instruction, MODE4 will take 1 serial flash clock cycle on the flash interface.
3. For a one pad instruction, MODE2_DDR will take 1 serial flash clock cycle on the flash interface.
4. For a one pad instruction, MODE4_DDR will take 2 serial flash clock cycles on the flash interface. For a 4 pad instruction MODE4_DDR will take half a cycle on the serial flash interface.
5. Data learning is not implemented on this chip.
6. t is not recommended to have 0x00 or 0xFF as the data learning pattern.

A sequence of such instruction-operand pairs may be pre-populated in the LUT according to the device connected on board. Each instruction-operand pair is of 16 bits (2 bytes) each. Every sequence pre-programmed in the LUT is referred to by its index.

The programmable sequence engine allows the user to configure the QuadSPI module according to the serial flash connected on board. The flexible structure is easily adaptable to new command/protocol changes from different vendors.

10.2.4.3.2 Flexible AHB buffers

In order to reduce the latency of the reads for AHB masters, the data read from the serial flash is buffered in flexible AHB buffers. There are four such flexible buffers. The size of each of these buffers is configurable with the minimum size being 0 Bytes and maximum size being the size of the complete buffer instantiated.The size of buffer 0 is defined as being from 0 to QSPI_BUF0IND. The Size of buffer 1 is from QSPI_BUF0IND to QSPI_BUF1IND, buffer2 is from QSPI_BUF1IND to QSPI_BUF2IND and buffer 3 is from QSPI_BUF2IND to the size of the complete buffer, which is given in the chip-specific QuadSPI information.

Each flexible AHB buffer is associated with the following

1. An AHB master. Optionally, buffer3 may be configured as an "all master" buffer by setting the QSPI_BUF3CR[ALLMST] bit. When buffer3 is configured in such a way, any access from a master not associated with any other buffer is routed to buffer3.
2. A datasize field representing the amount of data to be fetched from the flash on every "missed" access.

The master port number of every incoming request is checked and the data is returned/ fetched into the corresponding associated buffer. Every "missed" access to the buffer causes the controller to clear the buffer and fetch QSPI_BUFxCR[ADATSZ] amount of data from the serial flash.As such, there is no benefit in configuring a buffer size of greater than ADATSZ, as the locations greater than ADATSZ will never be used. For any AHB access, the sequence pointed to by the QSPI_BFGENCR[SEQID] field is used for the flash transaction initiated. The data is returned to the master as soon as the requested amount is read from the serial flash. The controller however, continues to prefetch the rest of the data in anticipation of a next consecutive request. Figure 10-16 shows the flexible AHB buffers.

The QSPI_BFGENCR[SEQID] field points to an index of the LUT. Refer to Look-up Table for details.

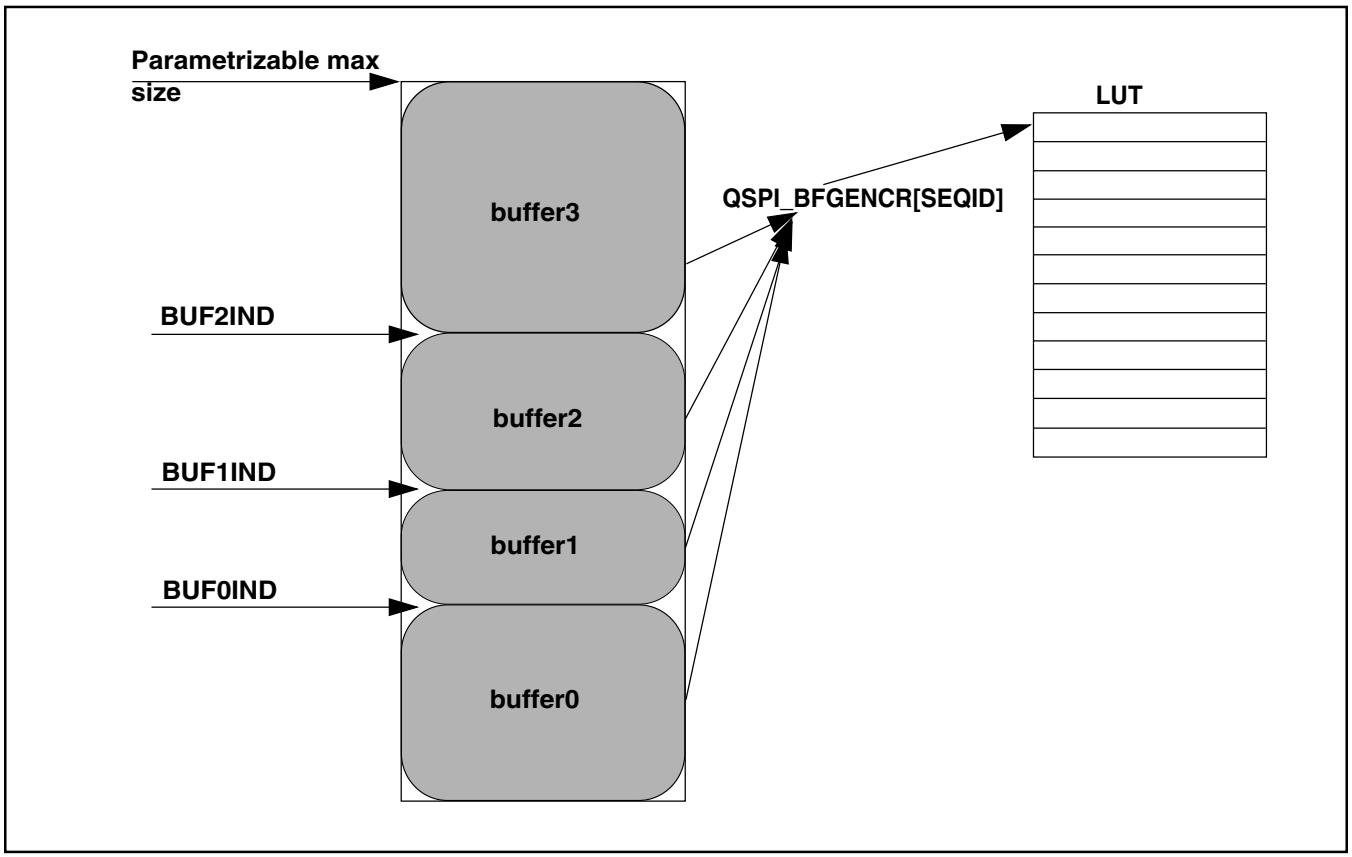


Figure 10-16. Flexible AHB Buffers

Buffer0 may optionally be configured to be associated with a high priority master by setting the QSPU_BUF0CR[HP_EN] bit. An access by a high priority master suspends any ongoing prefetch to any of the other buffers. The ongoing prefetch is suspended and the high priority master is serviced first. Once the high priority masters access completes, the suspended transaction is resumed (before any other AHB access is entertained). The status of the suspended buffer can be read from Sequence Suspend Status Register (QuadSPI_SPNDST).

10.2.4.3.3 Suspend-Abort Mechanism

Any low priority AHB access can be suspended by a high priority AHB master request. The ongoing transaction is suspended at 64 bit boundary. The suspended transaction is restarted after the high priority master is served and the high priority transaction including data prefetch is completed. While a transaction is in suspended state, it may be aborted if a transaction by the same suspended master is made to a location which is different from the location of the suspended transaction.

Any ongoing transaction is aborted if a request from the same master arrives for a location other than the location at which the transaction is going on. The abort can happen at any point of time.

10.2.4.3.4 Look-up Table

The Look-up-table or LUT consists of a number of pre-programmed sequences. Each sequence is basically a sequence of instruction-operand pairs which when executed sequentially generates a valid serial flash transaction. Each sequence can have a maximum of 8 instruction-operand pairs. The LUT can hold a maximum of 16 sequences. The figure below shows the basic structure of the sequence in the LUT.

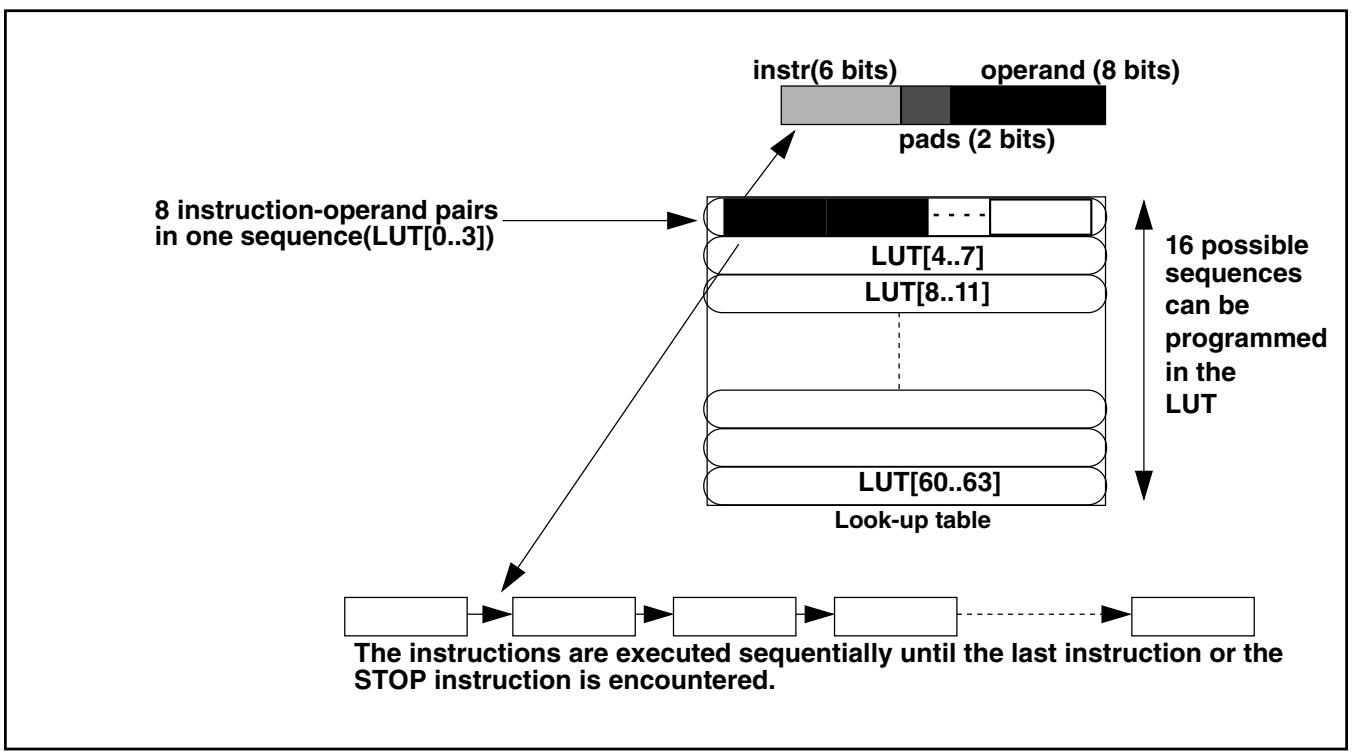


Figure 10-17. LUT and sequence structure

At reset, the index 0 of the look-up-table (LUT[0..3]) is programmed with a basic read sequence as given in Table 10-13. After reset the complete LUT may be reprogrammed according to the device connected on board.In order to protect its contents during a code runover the LUT may be locked, after which a write to the LUT will not be successful until it has been unlocked again.The key for locking or unlocking the LUT is **0x5AF05AF0**. The process for locking and un-locking the LUT is as follows:

Locking the LUT

1. Write the key **(0x5AF05AF0)** in to the LUT Key Register (QuadSPI_LUTKEY).
2. Write 0b01 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register locks the LUT.

Unlocking the LUT

1. Write the key **(0x5AF05AF0)** into the LUT Key Register (QuadSPI_LUTKEY)
2. Write 0b10 to the LUT Lock Configuration Register (QuadSPI_LCKCR). Note that this IPS transaction should immediately follow the above IPS transaction (no other IPS transaction can be issued in between). A successful write into this register unlocks the LUT.

The lock status of the LUT can be read from QSPI_LCKCR[UNLOCK] and QSPI_LCKCR[LOCK] bit.

Some example sequences are defined in Example Sequences. The reset sequence at LUT index 0 is given in the following table.

Table 10-13. Reset sequence

Instruction	Pad	Operand	Comment
CMD	0x00	0x03	Read Data byte command on one pad
ADDR	0x00	0x18	24 Addr bits to be sent on one pad
READ	0x00	0x08	Read 64 bits
JMP_ON_CS	0x00	0x00	Jump to instruction 0 (CMD)

10.2.4.3.5 Issuing SFM Commands

Each access to the external device follows the same sequence:

1. The user must pre-populate the LUT with the serial flash command sequences that are required for the flash device being used.
2. The QuadSPI module starts executing the instructions in the sequence one by one. The transaction starts and the status bit QSPI_SR[BUSY] is set.
3. Communication with the external serial flash device is started and the transaction is executed.
4. When the transaction is finished (all transmit- and receive operations with the external serial flash device are finished) the status bit QSPI_SR[BUSY] is reset. In case of an IP Command the QSPI_FR[TFF] flag is asserted.

Further details are given in below in Flash Programming and Flash Read.

You can trigger the processing of SFM commands in the QuadSPI module in one of the following ways:

- **Using IP commands**

For IP Commands the required components need to be written into the following registers:

- Write the serial flash address to be used by the instruction into QSPI_SFAR, refer to Serial Flash Address Register (QSPI_SFAR). For IP Commands not related to specific addresses, the base address of the related flash need to be programmed.For example, for an instruction which does not require an address (i.e. write enable instruction) the SFAR should be programmed with the base address of the memory the command is to be sent to.
- Write the sequence ID and data size details in the IP Configuration Register (QSPI_IPCR).
- Note that the write into the QSPI_IPCR[SEQID] field must be the last step of the sequence. It is possible to combine all fields of the QSPI_IPCR into one single write. Refer to IP Configuration Register (QSPI_IPCR) for details.

Note that there are some conditions where no IP Command is executed after writing the QSPI_IPCR[SEQID] field and the write operation itself is ignored. They are described in Command Arbitration.

- **Using AHB commands**

  Any AHB memory mapped access is routed to one of the buffers depending on the master port number of the request. If the access is a "miss", a new serial flash transaction is started. The transaction is based on the sequence pointed to by the BFGENCR[SEQID] field as described in Flexible AHB buffers.

  An AHB access is termed memory mapped when the access is to the memory mapped serial flashes, as described in Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A and Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B.

  Again the possible error conditions are described in Command Arbitration .

10.2.4.3.6 Flash Programming

In all cases the memory sector to be written needs to be erased first. The programming sequence itself is then initiated in the following way:

1. Check that the TX Buffer is empty. If the QSPI_SR[TXEDA] bit is set then the TX Buffer must be cleared by writing 1 into the QSPI_MCR[CLR_TXF] bit.
2. Program the address related to the command in the QSPI_SFAR register.

3. Provide initial data for the program command into the circular buffer via register TX Buffer Data Register (QSPI_TBDR) . At least four word of data must be written into the TX Buffer up to a maximum of 32.
4. Program the QSPI_IPCR register to trigger the command. The QSPI_IPCR[SEQID] should point to an index of the LUT which has the flash program sequence pre-programmed.The IDATSZ field should be set to denote the size of the write.
5. Depending on the amount of data required, step 3 must be repeated until all the required data have been written into the QSPI_TBDR register.The QSPI_SR[TXFULL] can be used to check if the buffer is ready to receive more data. At any time, the QSPI_TBSR[TRCTR] field can be read to check how many words have been written actually into the TX Buffer.

Upon writing the QSPI_IPCR[SEQID] field (refer to step 4) the QuadSPI module will start to execute the programmed sequence. It is the responsibility of the software to ensure that a correct sequence is programmed into the LUT in accordance with the flash memory connected to the module. The data is fetched from the TX Buffer. It consists of **32** entries of 32-bits and is organized as a circular FIFO, whose read pointer is incremented by four after each fetch. When all data are transmitted, the QuadSPI module will return from 'busy' to 'idle'. However, this is not true for the external device since the internal programming is still ongoing. It is up to the user to monitor the relevant status information available from the serial flash device and to ensure that the programming is finished properly.

10.2.4.3.7 Flash Read

Host access to the data stored in the external serial flash device is done in two steps. First, the data must be read into the internal buffers and in the second step these internal buffers can be read by the host.

1. **Reading Serial Flash Data into the QuadSPI Module Internal Buffers**

   A read access to the external serial flash device can be triggered in two different ways:

   - **IP Command Read**: For **reading flash data into the RX Buffer** the user must provide the correct sequence ID in the QuadSPI_IPCR[SEQID] register. The sequence ID points to a sequence in the LUT. It is the responsibility of the software to ensure that a correct read sequence is programmed in the LUT in accordance with the serial flash device connected on board. The user should program the Serial Flash Address Register (QSPI_SFAR) and the IP Configuration Register (QSPI_IPCR) registers. All available read commands supported by the external serial flash are possible.

Optionally it is possible to clear the RX Buffer pointer prior to triggering the IP Command by writing a 1 into the QuadSPI_MCR[CLR_RXF] field.

From these inputs, the complete transaction is built when the QSPI_IPCR[SEQID] field is written. The transaction related to the read access starts and the requested number of bytes is fetched from the external serial flash device into the RX Buffer. Since the read access is triggered by an IP command, the IP_ACC status bit and the BUSY bit are both set (both are located in the Status Register (QSPI_SR)). A count of the number of entries currently in the Rx Buffer can be obtained from QSPI_RBCT[RXBRD].

The communication with the external serial flash is stopped when the specified number of bytes has been read (successful completion of the transaction).

- **AHB Command Read**: For **reading flash data into the AHB Buffer** the user must set up a read access by a master to the address range in the system memory map which the external serial flash devices are mapped to. The user should also program the buffer registers corresponding to the AHB master initiating the request, this is depends on the configuration of the QSPI_RBCT[RXBRD]. The user should provide the correct sequence ID into the buffer generic configuration register (QSPI_BFGENCR). It is the responsibility of the software to ensure that a correct read sequence in programmed in the LUT in accordance with the serial flash device connected on board. Flash device selection and access mode are determined by the address accessed in the AHB address space associated to the QuadSPI module (refer to Memory Mapped Serial Flash Data - Individual Flash Mode on Flash A, Memory Mapped Serial Flash Data - Individual Flash Mode on Flash B and Parallel Flash Mode.)

  On each AHB read access to the memory mapped area the valid data in the AHB Buffer is checked against the address requested in the actual read. When the AHB read request can't be served from the content of the AHB Buffer, the buffer is flushed and the sequence pointed to by the sequence ID is executed by the controller. The requested number of buffer entries defined in the QSPI_BUFxCR[ADATSZ] field is then fetched from the external serial flash device into the internal AHB Buffer. Since the read access is triggered via the AHB bus, the QSPI_SR[AHB_ACC] status bit is set driving in turn the QSPI_SR[BUSY] bit until the transaction is finished. The communication with the external serial flash is stopped when the specified number of entries has been filled.

2. **Data Transfer from the QuadSPI Module Internal Buffers**

The data read out from the external serial flash device by the QuadSPI module is stored in the internal buffers.The means of accessing the data from the buffer differs depending on which buffer the data has been loaded to. Refer to Block Diagram for details about the two available buffers, the RX Buffer and the AHB Buffer, in the QuadSPI module:

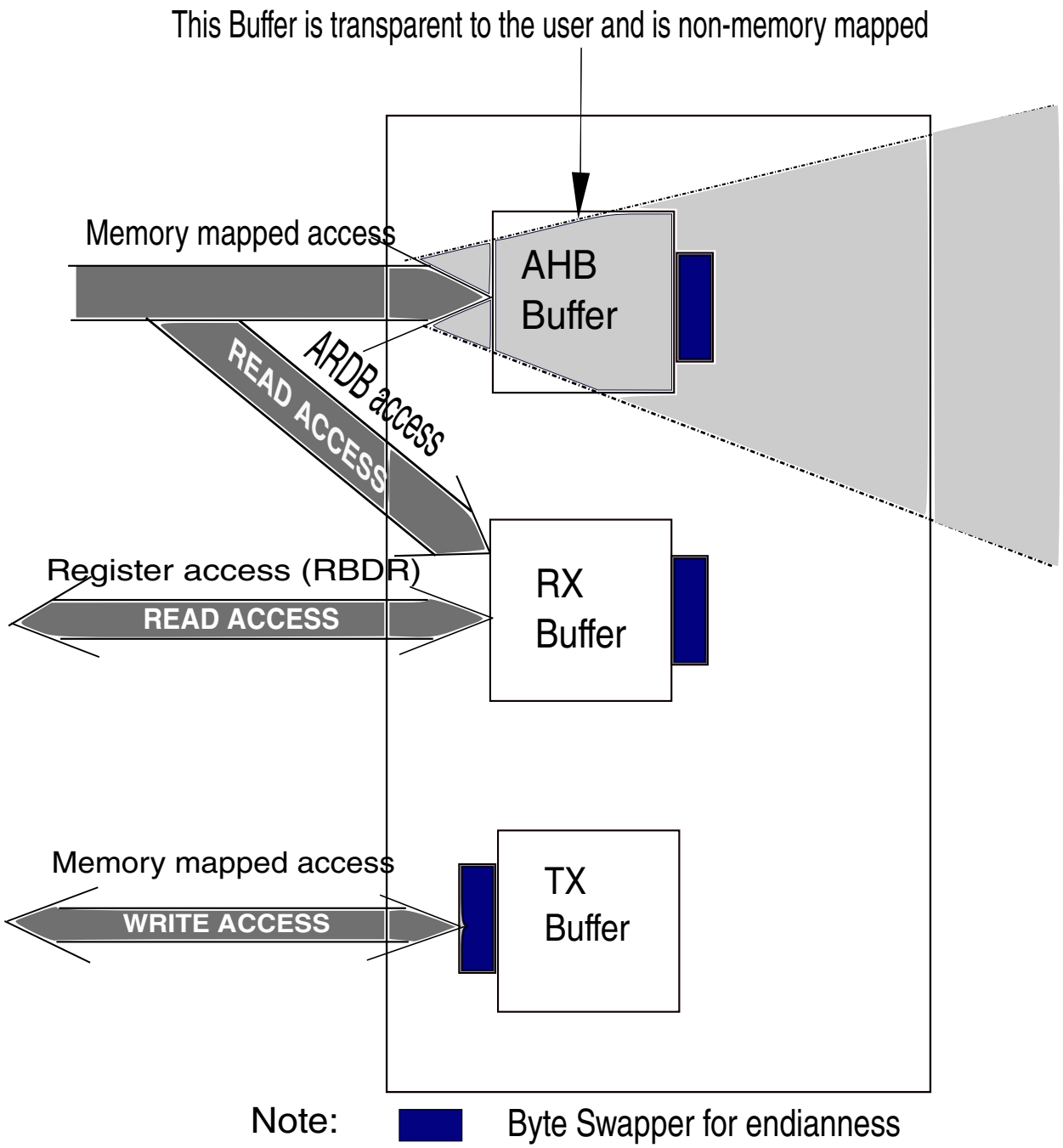


Figure 10-18. QuadSPI memory map

- The RX Buffer is implemented as FIFO of depth 32 entries of 4 bytes. Its content is accessible in two different address areas both referring to the identical data and the same physical memory.

  In the IPS address space in the area associated to QSPI_RBDR0 to QSPI_RBDR31

  In the AHB address space in the area associated to QSPI_ARDB0 to QSPI_ARDB31. Two successive entries are accessed with one single 64 bit AHB read operation.

  RX Buffer operation can be summarized as follows: The QSPI_RBCT[WMRK] field determines at which fill level the RXWE bit is asserted and how many entries are removed from the RX Buffer on each Buffer POP operation. So the QSPI_SR[RXWE] bit indicates that the configured number of data entries is available in the RX Buffer and the QSPI_RBSR[RDBFL] field indicates how many valid entries are available in total. Note that the first entry (QSPI_RBDR0 or QSPI_ARDB0) always corresponds to the first valid entry in the RX Buffer. The software needs to manage the number of valid data bytes itself.

  Further details can be found in RX Buffer Data Register (QuadSPI_RBDR*n*) and in AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31).

- **Flag-based Data Read of the RX Buffer** is done by polling the QSPI_SR[RXWE] bit. When it is asserted the valid entries can be read either via the IPS address space (QSPI_RBDRn) or the AHB address space (QSPI_ARDBn). A Buffer POP operation must be triggered by the application by writing a 1 into the QSPI_FR[RBDF] bit - this automatically updates the FIFO to point to the next entry as defined by RBCT[WMRK]. For example, if WMRK is set to 3, then the buffer will discard 16 bytes of data.

- **DMA controlled Data Read of the RX Buffer** is done by using the DMA module.The application must ensure that the DMA controller of the related device is programmed appropriately like it is described in DMA Usage.

  DMA controlled read out is triggered fully automatically by the assertion of the QSPI_SR[RXWE] bit. The related Buffer POP operation is also handled completely inside the QuadSPI module. Like in the case above, accessing the RX Buffer content either on QSPI_RBDRn or QSPI_ARDBn related addresses is equivalent.

- **AHB Buffer data read via memory mapped access:** This kind of access is done by reading one of the addresses assigned to the external serial flash device(s) within the range given in Table 10-6 table *under the condition that the data requested are already present in the AHB Buffer or it is currently being read from the serial flash device by the instruction in progress.* If this is not the

case a memory mapped AHB command read is triggered as described above. If the requested data is already available in the AHB Buffer they are provided directly to the host.

When AHB access are made to the flash memory mapped address, the data will be fetched and returned to the AHB interface. Till the data is being fetched the AHB interface would be stalled. As soon as the data from the requested address has been read by the QuadSPI module the AHB read access is served. So it is possible to run sequential reads from the AHB buffer at arbitrary speed without the need to monitor any information about the availability of the data. Nevertheless this access scheme stalls the AHB bus for the time required to read the data from the serial flash device. A better way is (when it is known the access is sequential) to have a prefetch enabled (by programming the ADATSZ field) such that before the next sequential AHB access come, the data is already fetched into the buffer.

As long as the host restricts its accesses to the data already in the buffer and the data currently fetched from the serial flash, it is possible to run the host read from the AHB Buffer in parallel to the serial flash read into the AHB Buffer.

10.2.4.3.8 Byte Ordering of Serial Flash Read Data

In this paragraph the byte ordering of the serial flash data is given. The basic scheme is that the **first** byte read out of the serial flash device - which is addressed by the QSPI_SFAR[SFADR] field - corresponds to bit position QSPI_RBDR0[31:24] register for IP Command read. In contrast to that for AHB Command read the bytes are always positioned according to the byte ordering of the AHB bus.

- **Byte Ordering in Individual Flash Mode**

  The following table gives the byte ordering scheme of how the byte oriented data space of the serial flash device is mapped into one single 32 bit entry of the RX Buffer or the AHB Buffer. The table is valid within the following context:

  - Flash A or Flash B in Individual Flash Mode
  - All AHB data read commands with access size of 32 bit

Table 10-14. Byte Ordering in Individual Flash Mode

Serial Flash Byte Numbering	3	2	1	0
Buffer Entry Bit Position [31:0] (32 Bit data width)	[31:24]	[23:16]	[15:8]	[7:0]

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4, then the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

For AHB Commands, reads, starting from an address not aligned to 32 bit boundaries, the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Byte Ordering in Parallel Flash Mode**

  In Parallel Flash Mode each byte is combined out of 2 half bytes which are read in parallel from the two serial flash devices. The following tables shows how the flash content is separated into the half bytes and how the half bytes are assembled to the content of the QSPI_RBDR0 register.

Table 10-15. Serial Flash Device Half Byte Ordering

Serial Flash Device Byte #	Flash A Bit Position		Flash B Bit Position	
	[7:4]	[3:0]	[7:4]	[3:0]
0	fah0	fal0	fbh0	fbl0
1	fah1	fal1	fbh1	fbl1
2	fah2	fal2	fbh2	fbl2
3	fah3	fal3	fbh3	fbl3
4	fah4	fal4	fbh4	fbl4
5	fah5	fal5	fbh5	fbl5
6	fah6	fal6	fbh6	fbl6
7	fah7	fal7	fbh7	fbl7
8	fah8	fal8	fbh8	fbl8

The table entry naming reflects the half byte positioning in the serial flash devices:

- **<fa>**h0 means **Flash A**, <fb>h0 means Flash B.

- fa**<h>**0 means half byte in **high position**, fa<l>o means half byte in low position.
- fah**<0>** means **physical byte address 0** in the serial flash device, fal<1> means physical byte address 1 in the serial flash device.

Table 10-16. Byte Ordering in Parallel Flash Mode - RX Buffer

QSPI_SFAR[SFADR] set to 0x000_0000								
QSPI_RBDR0 **QSPI_ARDB0**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
QSPI_RBDR1 **QSPI_ARDB1**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2
QSPI_SFAR[SFADR] set to 0x000_0001								
QSPI_RBDR0 **QSPI_ARDB0**	fal2	fbl2	fah2	fbh2	fal1	fbl1	fah1	fbh1
QSPI_RBDR1 **QSPI_ARDB1**	fal4	fbl4	fah4	fbh4	fal3	fbl3	fah3	fbh3

Note

For IP Commands the read size can be given in number of bytes. If this number is not a multiple of 4 the last buffer entry is not completely filled with the missing higher numbered bytes at undefined values.

* Appicable only for single io mode.

Table 10-17. Byte Ordering in Parallel Flash Mode - AHB Buffer

AHB Address **(32 Bit Access)**	fal1	fbl1	fah1	fbh1	fal0	fbl0	fah0	fbh0
AHB Address 0x800_0004 **(32 Bit Access)**	fal3	fbl3	fah3	fbh3	fal2	fbl2	fah2	fbh2

Note

For AHB Command read starting from an address not aligned to 32 bit boundaries or AHB access size smaller than 32 bit the requested bytes are given at the appropriate positions according to the AMBA AHB specification.

- **Buffer Entry Ordering for 64 Bit Read Access**

  For read access via the AHB interface 64 bit access is possible. Each 64 bit access reads 2 32 bit entries simultaneously. The ordering of these 32 bit entries within the 64 bit word is given in the following table.

Table 10-18. 64 Bit Read Access Buffer Entry Ordering

AHB Read Data Bit Position [63:0]	[63:32]	[31:0]
Buffer Entry #	Odd (1, 3, 5, ...)	Even (0, 2, 4, ...)

10.2.4.3.9 Normal Mode Interrupt and DMA Requests

The QuadSPI module has different flags that can only generate interrupt requests and one flag that can generate interrupt as well as DMA requests. The following table lists the eight conditions. Note that the flags mentioned in the table are related to the Flag Register (QSPI_FR).

Table 10-19. Interrupt and DMA Request Conditions

Condition	Flag(QSPI_FR)	DMA
Data Learn patttern Failure	DLPFF	-
TX Buffer Fill	TBFF	-
TX Buffer Underrun	TBUF	-
Illegal Instruction Error	ILLINE	-
RX Buffer Drain	RBDF	X
RX Buffer Overflow	RBOF	-
AHB Buffer Overflow	ABOF	-
AHB Sequence Error	ABSEF	-
IP Command Usage Error	IUEF	-
IP Command Trigger during AHB Access Error	IPAEF	-
IP Command Trigger could not be executed Error	IPIEF	-
IP Access during AHB Grant Error	IPGEF	-

Table continues on the next page...

Table 10-19. Interrupt and DMA Request Conditions (continued)

Condition	Flag(QSPI_FR)	DMA
IP Command related Transaction Finished	TFF	-

Each condition has a flag bit in the Flag Register (QSPI_FR) and a Request Enable bit in the DMA Request Select and Enable Register (QSPI_RSER) . The RX Buffer Drain Flag (RBDF) has separate enable bits for generating IRQ and DMA requests. Note that not all flags have an individual IRQ line. Check the devices Interrupt Vector Table for more details.

- Transmit Buffer Fill Interrupt Request:

  The Transmit Buffer Fill IRQ indicates that the TX Buffer can accept new data. It is asserted if the QSPI_FR[TBFF] flag is asserted and if the corresponding enable bit (QSIP_RSER[TBFIE]) is set. Refer to TX Buffer Operation, for details about the assertion of the QSPI_FR[TBFF] flag.

- Receive Buffer Drain Interrupt or DMA Request:

  The Receive Buffer Drain IRQ derived from the QSPI_FR[RBDF] flag indicates that the RX Buffer of the QuadSPI module has data available from the serial flash device to be read by the host. It remains set as long as the QSPI_RBSR[RXWE] bit is set. The QSPI_RSER[RBDIE] bit enables the related IRQ.

  Aside from the IRQ it is possible to handle RX Buffer drain by DMA. If the QSPI_RSER[RBDDE] bit is set, a DMA request will be triggered when the RX Buffer contains more than QSPI_RBCT[WMRK] valid entries.The application must set the environment appropriately (for example, the DMA controller) for the DMA transfers.

- Buffer Overflow/Underrun Interrupt Request:

  The Buffer Overflow/Underrun IRQ is a combination of the following flags (all located in the QSPI_FR register with the related enable bits in the QSPI_RSER register):

  - TBUF - TX Buffer Underrun, enabled by TBUIE
  - RBOF - RX Buffer Overflow, enabled by RBOIE
  - ABOF - AHB Buffer Overflow, enabled by ABOIE

    The Transmit Buffer Underrun indicates that an underrun condition in the TX Buffer has occurred. It is generated when a write instruction is triggered whilst the Tx Buffer is empty and the QSPI_RSER[TFUFIE] bit is set.

The Receive Buffer Overflow indicates that an overflow condition in the RX Buffer has occurred. It is generated when the RX Buffer is full, an additional read transfer attempts to write into the RX Buffer and the QSPI_RSER[RBOIE] bit is set.

The AHB Buffer Overflow indicates that an overflow condition in the AHB Buffer has occurred. It is generated when the AHB Buffer is full, an additional read transfer attempts to write into the AHB Buffer and the QSPI_RSER[ABOIE] bit is set.

The data from the transfers that generated the individual overflow conditions is ignored.

- Serial Flash Command Error Interrupt Request

  If the IPAEF, IPIEF, IPGEF or IUEF flags in the QSPI_FR are set, and the related interrupt enable bits in the QSPI_RSER are also set, then an interrupt is requested.

- Transaction Finished Interrupt Request

  The IP Command Transaction Finished IRQ indicates the completion of the current IP Command.It is triggered by the QSPI_FR[TFF] flag and is masked by the QSPI_RSER[TFIE] bit.

10.2.4.3.10 TX Buffer Operation

The TX Buffer provides the data used for page programming. For proper operation it is required to provide at least four entry in the TX Buffer prior to starting the execution of the page programming command. The application must ensure that the required number of data bytes is written into the TX Buffer fast enough as long as the command is executed without a TX Buffer overflow or underrun.

The QuadSPI module sets the QSPI_FR[TBFF] flag so long as the TX Buffer is not full and can accept more data.

When the QuadSPI module tries to pull data out of an empty TX Buffer the TX Buffer underrun is signaled by the QSPI_FR[TBUF] flag.The TX buffer underrun flag is also asserted when TX buffer contains less than 128 bits of data and QuadSPI module tries to pull out data from it. The current IP Command leading to the underrun condition is continued until the specified number of bytes has been sent to the serial flash device, in the underrun condition when QuadSPI module tries to pull out data of empty TX buffer, the data transferred is all F's i.e. once the underrun flag is set under this condition, it will return F's until the required number of bytes are not sent. This has been done to ensure that the software need not to erase whole sector after underrun, just reprogramming from

failure point will serve the purpose. When this Sequence Command is finished, the QSPI_FR[TBFF] flag is asserted indicating that the Tx Buffer is ready to be written again.

The TX Buffer overflow isn't signaled explicitly, but the TX Buffer fill level can be monitored by the QSPI_TBSR[TRBFL] field.

Refer to TX Buffer Status Register (QuadSPI_TBSR) and Flag Register (QuadSPI_FR) for details about the TX Buffer related registers.

10.2.4.3.11 Address scheme

Earlier serial flash memories supported only 24-bit address space hence restricting the maximum memory size of the serial flash as 16 MB. The new memory specification supports two types of 32-bit addressing mode in addition to legacy 24-bit address mode.

- **Extended Address Mode**

  In this mode, the legacy 24-bit commands are converted to accept 32-bit address commands. The flash memory needs to be configured for 32-bit address mode. Also, while programming the LUT sequence in QuadSPI for 32-bit mode, the ADDR and ADDR_DDR command should be programmed with 8’d32 as the operand value. By default, the QuadSPI is in 24-bit legacy address mode. Each of the memory vendors have a different way of enabling this mode (Refer to the memory specification from memory vendors). For example, the command B7h sent to Macronix flash will enable it for 32-bit address mode.

- **Extended Address register**

  In this mode, the upper 8-bit of the 32-bit address is provided by the Extended address register in the memory itself. The memory provides a specific register which is updated according to the address to be accessed. This effectively converts the legacy 24-bit address command into 32-bit address commands. The memories greater in size than 16 MB, consists of banks of 16 MB. The 8-bit written in the extended address register effectively enables a bank. For example in Spansion memory, when the extended address register is updated with a value of 0x01 with the help of the command 17h, it will open Bank1 of the memory. The consequent 24-bit address commands will lead to Bank1. The extended address register needs to be update with the respective value for access to other banks. This effectively converts the legacy 24-bit address command into 32-bit address commands.

10.2.5 Initialization/Application Information

This section provides the initialization and application information of the QuadSPI module.

10.2.5.1 Power Up and Reset

Note that the serial flash devices connected to the QuadSPI module may require special voltage characteristics of their inputs during power up or reset. It is the responsibility of the application to ensure this.

10.2.5.2 Available Status/Flag Information

This paragraph gives an overview of the different status and flag information available and their interdependencies for different use cases. Related registers are QSPI_SR and QSPI_FR. Refer to the related descriptions how to set up the QuadSPI module appropriately.

10.2.5.2.1 IP Commands

Refer to IP Configuration Register (QuadSPI_IPCR) for additional details not explicitly covered in this paragraph.

- **IP Commands - Normal Operation**

  Writing the QSPI_IPCR[SEQID] field triggers the execution of a new IP Command. Given that this is a legal command the QSPI_SR[IPACC] and the QSPI_SR[BUSY] bits are asserted simultaneously, immediately after the execution is started.

  When the instruction on the serial flash device has been finished these bits are de-asserted and the QSPI_FR[TFF] flag is set.

- **IP Commands - Error Situations**

  Refer to Table 10-20 below.

10.2.5.2.2 AHB Commands

Refer to Section 1, Reading Serial Flash Data into the QuadSPI Module, in Flash Read for additional details not explicitly covered in this paragraph.

- **AHB Commands - Normal Operation**

  Memory mapped read access to a serial flash address not contained in the AHB Buffer, triggers the execution of an AHB Command. Given that this is a legal command the QSPI_SR[AHBACC] and the QSPI_SR[BUSY] bits are asserted simultaneously immediately after the execution is started. When the instruction on the serial flash device has been finished these bits are de-asserted.

- **IP Commands - Error Situations**

  Refer to Table 10-20 below.

10.2.5.2.3 Overview of Error Flags

The following table gives an overview of the different error flags in the QSPI_FR register and additional error-related details.

Table 10-20. Overview of QSPI_FR Error Flags

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
AHB Error Flag	ABSEF	Flash transaction is aborted	AHB sequence contains • WRITE instruction • WRITE_DDR instruction
AHB Error Flag	ABOF	Flash transaction continues until it finishes	Set when the module tried to push data into the AHB buffer that exceeded the size of the AHB buffer. Only occurs due to wrong programming of the QSPI_BUFxCR[ADATSZ].
Miscellaneous Error Flag	DLPFF[1]	Flash transaction continues until it finishes	Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning pattern.
Miscellaneous Error Flag	ILLINE	Flash transaction aborted	Illegal instruction Error Flag – Set when an illegal instruction is encountered by the controller in any of the sequences.
Command Arbitration Error	IPIEF	TFF not asserted in conjunction with that command	IP Command Error - caused when IP access is currently in progress (IP_ACC set) and • write attempt to QSPI_IPCR register.

Table continues on the next page...

Table 10-20. Overview of QSPI_FR Error Flags (continued)

Error Category	Error Flag in QSPI_FR	Command Execution on Serial Flash Device TFF Behavior (in case of IP commands only)	Description
			• write attempt to QSPI_SFAR register. • write attempt to QSPI_RBCT register.
Command Arbitration Error	IPAEF		• AHB Command already running, another IP Command could not be executed. • AHB Command already running, write attempt to QSPI_IPCR[SEQID] field.
Command Arbitration Error	IPGEF		• Exclusive access to the serial flash granted for AHB Commands, write attempt to QSPI_IPCR[SEQID] field.
IP Command Error	IUEF	—	• IP Command Usage Error
Buffer Related Error	RBOF	TFF is asserted on completion	• RX Buffer Overrun
Buffer Related Error	TBUF		• TX Buffer Underrun

1. Data learning is not implemented on this chip.

Note that only the buffer related errors are related to a transaction on the external serial flash. All the other errors do not trigger an actual transaction.

10.2.5.2.4 IP Bus and AHB Access Command Collisions

There are two flags related to this topic, the QSPI_FR[IPAEF] and QSPI_FR[IPIEF]. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for a description of the flags and Command Arbitration , for details about possible command collisions.

10.2.5.3 Exclusive Access to Serial Flash for AHB Commands

It is possible that several masters need to access the serial flash device connected to the QuadSPI module separately, one master by triggering IP Commands and reading the RX Buffer (via RBDR*n* register) and the other masters by triggering AHB Commands (via ARDB*n* Registers). These two set of buffer (RBDR and ARDB Buffer) points to the same physical buffer.Refer to Figure 10-18 To avoid command collisions resulting in

excessive latencies the QuadSPI module implements a request-handshake mechanism between the master triggering AHB Commands and the QuadSPI module allowing this specific master to request exclusive access to the serial flash device for AHB Commands. If this exclusive access is granted the execution of IP Commands is blocked. This resolves command collisions and excessive times where the AHB interface may be blocked.

If this capability is used in the device there is additional status and flag information available related to this mechanism. The QSPI_SR[AHBGNT] bit reflects the module-internal state that the exclusive access mentioned above is granted, any attempt to trigger an IP Command is rejected and results in the assertion of the QSPI_FR[IPGEF] flag. Refer to the descriptions of the related bit and flag for details.

It is within the responsibility of the application to set up the master using this mechanism appropriately, if used incorrectly no IP Commands at all can be triggered.

Two different cases can be distinguished:

10.2.5.3.1 RX Buffer Read via QSPI_ARDB Registers

In this case all masters share the AHB bus for RX Buffer as well as for AHB Buffer read. In this case the access to the AHB interface by the master triggering AHB Commands must be deferred until any pending IP Command has been finished **and** the RX Buffer readout has been finished as well. The QSPI ARDB Buffers access the Rx buffer i.e the data from the Rx Buffer is returned and no data from AHB Buffer is touched. This is the conservative use case, corresponding to the reset value 0 of the QSPI_RBCT[RXBRD] bit.

In this case the QSPI_SR[AHBGNT] bit is asserted not earlier than any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0 and Rx Buffer readout is via AHB(QSPI_RBCT[RXBRD]) equal to 1.

10.2.5.3.2 RX Buffer Read via QSPI_RBDR Registers

This is the preferred use case as an access to the AHB buffer (memory mapped flash) does not interfere with any IPS access to read the RBDR buffer. It is not possible that a pending AHB bus access triggered by an AHB Command stalls the AHB bus and blocks the RX Buffer readout since the RX Buffer is read via the IP bus based registers QSPI_RBDR0 to QSPI_RBDR31.

For this case it is recommended to program the QSPI_RBCT[RXBRD] bit to 1. The QSPI_SR[AHBGNT] bit is asserted immediately after any running IP Command has been finished (QSPI_SR[IP_ACC] is 0), the RX Buffer has been read out completely (QSPI_RBSR[RDBFL] equal to 0) or no DMA read is pending (QSPI_SR[RXDMA] equal to 0, allowing the master triggering AHB Commands to trigger AHB Commands as soon as possible without the need to wait for the RX Buffer readout to be finished.

10.2.5.4 Command Arbitration

In case of overlapping commands, the arbitration scheme is described in the following paragraphs under the assumption that the priority mechanism described in Exclusive Access to Serial Flash for AHB Commands is **not** used:

- During the execution of an IP Command, the running IP Command can't be terminated by issuing another IP Command or AHB Command. The QSPI_FR[IPIEF] flag is asserted when the host tries to write into the QSPI_IPCR register. When the host triggers an AHB Command (refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for details), this command is stalled until the currently running IP Command is finished.
- During the execution of an AHB Command, the running AHB Command can't be terminated by issuing an IP Command. The command is ignored and the QSPI_FR[IPAEF] flag is asserted. Refer to Flag Register (QuadSPI_FR) for the description of these flags.

  When another AHB Command is triggered the address of the memory mapped access is considered. If the requested address is currently read from the serial flash device, the running command is continued. If this is not the case the currently running command is terminated and another AHB Command related to the requested address is executed. Refer to sub-section "Reading Serial Flash Data into the QuadSPI Module" of Flash Read section, for further details.

In case of coinciding commands the IP Command is triggered and the AHB Command is stalled until the IP Command has been finished (QSPI_SR[IP_ACC] has been deasserted).

The IP Commands ignored in case of command collision will not result in the assertion of the QSPI_FR[TFF] flag.

10.2.5.5 Flash Device Selection

Regardless of the SFM Command (IP or AHB) the access mode is selected by specifying the 32 bit address value for the following SFM Command.

For IP Commands the access mode is selected with the address programmed into the QSPI_SFAR register. Refer to Serial Flash Address Register (QuadSPI_SFAR) for details.

For AHB Commands the access mode is determined by the memory mapped address which is accessed Refer to AMBA Bus Register Memory Map for details.

10.2.5.6 DMA Usage

For the complete description of the DMA module refer to the related DMA Controller chapter. In this paragraph only the details specific to the DMA usage related to the QuadSPI module are given.

10.2.5.6.1 DMA Usage in Normal Mode

10.2.5.6.1.1 Bandwidth considerations

Careful consideration of the throughput rate of the entire chain (serial flash -> AHB bus / IP Bus -> DMA controller) involved in the read data process is essential for proper operation. Such analysis must take into account not only the data rate provided by the serial flash but also the data rate of the AHB bus and the performance of the DMA controller in reading data from the RX buffer.

Two figures must match for proper operation, that means that the data rate provided by the serial flash device must not exceed the average RX Buffer readout data rate. Otherwise, the longer this state persists, a RX Buffer overflow will result.

AHB Bus Side (data read):

The total number of bus cycles for each DMA Minor Loop completion is added from the following components:

- Overhead for each minor loop, given by DMA controller: Assume 10 cycles
- Overhead due to clock domain crossing: Assume 2 cycles
- Number of bus clock cycles required for 8 bytes (64 bit read size): Assume 2 cycles (read/write sequence of DMA controller)

Note that the size of the minor loop is determined by the size of the QSPI_RBCT[WMRK] field, therefore the overhead given above distributes among (QSPI_RBCT[WMRK]+1)/2 read accesses of 64 bit each.

The following table gives some examples for typical use cases:

Table 10-21. Access Duration Examples - Bus Clock Side

QSPI_RBCT[WMRK]	Number of Bytes per DMA Loop [1]	Number of Bus Clock cycles for DMA Minor Loop	Time Duration of DMA Minor Loop for 120Mhz Bus clock Frequency
0	4	12+2 = 14	~117ns
1	8	12+2 = 14	~117ns
3	16	12+4 = 16	~133ns
7	32	12+8 = 20	~167ns
11	48	12+12 = 24	~200ns

1. DMA Loop means one Minor Loop Completion which is equivalent to one.

NOTE

The table figure represents ideal scenario, actual performance will depend on how the system is integrated.

Serial Flash Device Side (data read):

The number of serial flash cycles can be determined in the following way:

- Number of serial flash clock cycles required to read 4 bytes, corresponding to one RX Buffer entry (setup of command and address not considered): 2 cycles for Quad DDR mode instructions in Parallel Flash Mode, 4 cycles for Quad (SDR) mode instruction in parallel flash mode or Dual IO DDR mode instruction in parallel flash mode, 8 cycles for Quad Mode (SDR) instructions in Individual Flash Mode etc.
- Overhead due to clock domain crossing : 1 cycle.

The following table lists the number of clock cycles required to read the data from the serial flash corresponding to the different settings of the QSPI_RBCT[WMRK] field:

Table 10-22. Access Duration Examples - Serial Flash side

QSPI_RBCT[WMRK] setting	Num Bytes per DMA Loop [1]	Num SCKFx for 60MHz SCKFx			Time duration of Flash data readout for 60MHz SCKFx (~16.6ns period)		
		IFM [2] Quad	IFM Quad DDR	PFM [3] Quad DDR	IFM Quad	IFM Quad DDR	PFM Quad DDR
0	4	9	5	3	~150ns	~83ns	~50ns
1	8	17	9	5	~282ns	~150ns	~83ns
3	16	33	17	9	~548ns	~282ns	~150ns
7	32	65	33	17	~1079ns	~548ns	~282ns

Table continues on the next page...

Table 10-22. Access Duration Examples - Serial Flash side (continued)

QSPI_RBCT[WMRK] setting	Num Bytes per DMA Loop [1]	Num SCKFx for 60MHz SCKFx			Time duration of Flash data readout for 60MHz SCKFx (~16.6ns period)		
		IFM [2] Quad	IFM Quad DDR	PFM [3] Quad DDR	IFM Quad	IFM Quad DDR	PFM Quad DDR
11	48	97	49	25	~1610ns	~813ns	~415ns

1. DMA Loop means one Minor loop completion which is equivalent to one Major Loop iteration.
2. Individual flash mode.
3. Parallel flash mode.

From the examples given in the two tables above, it can be seen that depending on the relationship between the Bus clock and Serial flash clock frequencies, there are settings possible where the serial flash provides the read data faster than the AHB bus can read out the RX buffer. In the above tables, it is the case of PFM Quad DDR mode with Watermark up to 3 and other cases. In these cases, the RX buffer data keeps accumulating over time and will eventually overflow. To avoid RX Buffer overflow, the data transaction size should be small enough.

A complementary example would be when the watermark is set to be too high. In such a case, the time taken by the DMA to read out the RX buffer entries should be smaller than the time taken by the controller to push in the remaining entries in the buffer.

NOTE

The tables mentioned above are only examples which must be correlated with the DMA in the system.

10.2.5.7 Parallel mode

QuadSPI can access two flashes in parallel. This increases the throughput of the QuadSPI by two times. Both write and read operations are supported in parallel mode. When dual die flashes are accessed in parallel mode, it is mandatory for flash A1 to be of the same size as B1 and A2 to be of the same size as B2. The following figure shows how QuadSPI maps the incoming addresses to the different flashes connected on board.

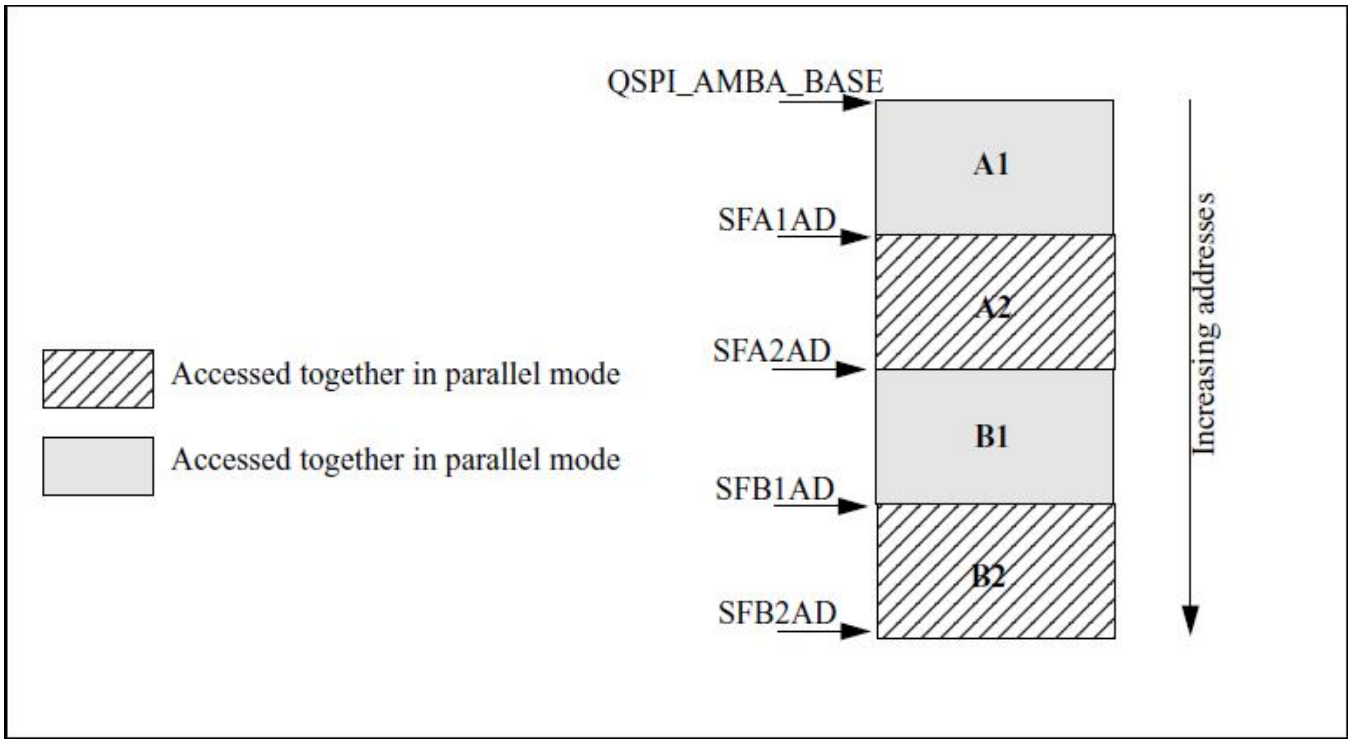


Figure 10-19. Flash addressing

An example programming for parallel mode access is given below (flash sizes are assumed to be 256MB):

- QSPI_AMBA_BASE - 0x10000000
- QSPI_SFA1AD[TPADA1] - 0x20000000
- QSPI_SFA2AD[TPADA2] - 0x30000000
- QSPI_SFB1AD[TPADB1] - 0x40000000
- QSPI_SFB2AD[TPADB2] - 0x50000000

In order to access the first location of A1/B1 pair, the incoming address should be 0x10000000. QSPI_AMBA_BASE is subtracted from this address and the result is divided by two. Therefore, address provided to flash A1 and B1

Flash Address = (Memory mapped address - QSPI_AMBA_BASE)/2

For Memory Mapped address:

- 0x10000000, flash address: 0x0 (Or, the first address of flash A1 and B1)
- 0x10000004, flash address: 0x2
- 0x10000008, flash address: 0x4 etc.

Similarly, in order to access the first location of A2/B2 pair, the incoming address should be 0x30000000.

Flash Address = (Memory mapped address - SFA2AD)/2

For Memory Mapped address:

- 0x30000000, flash address: 0x0 (Or, the first address of flash A2 and B2)
- 0x30000004, flash address: 0x2
- 0x30000008, flash address: 0x4 etc.

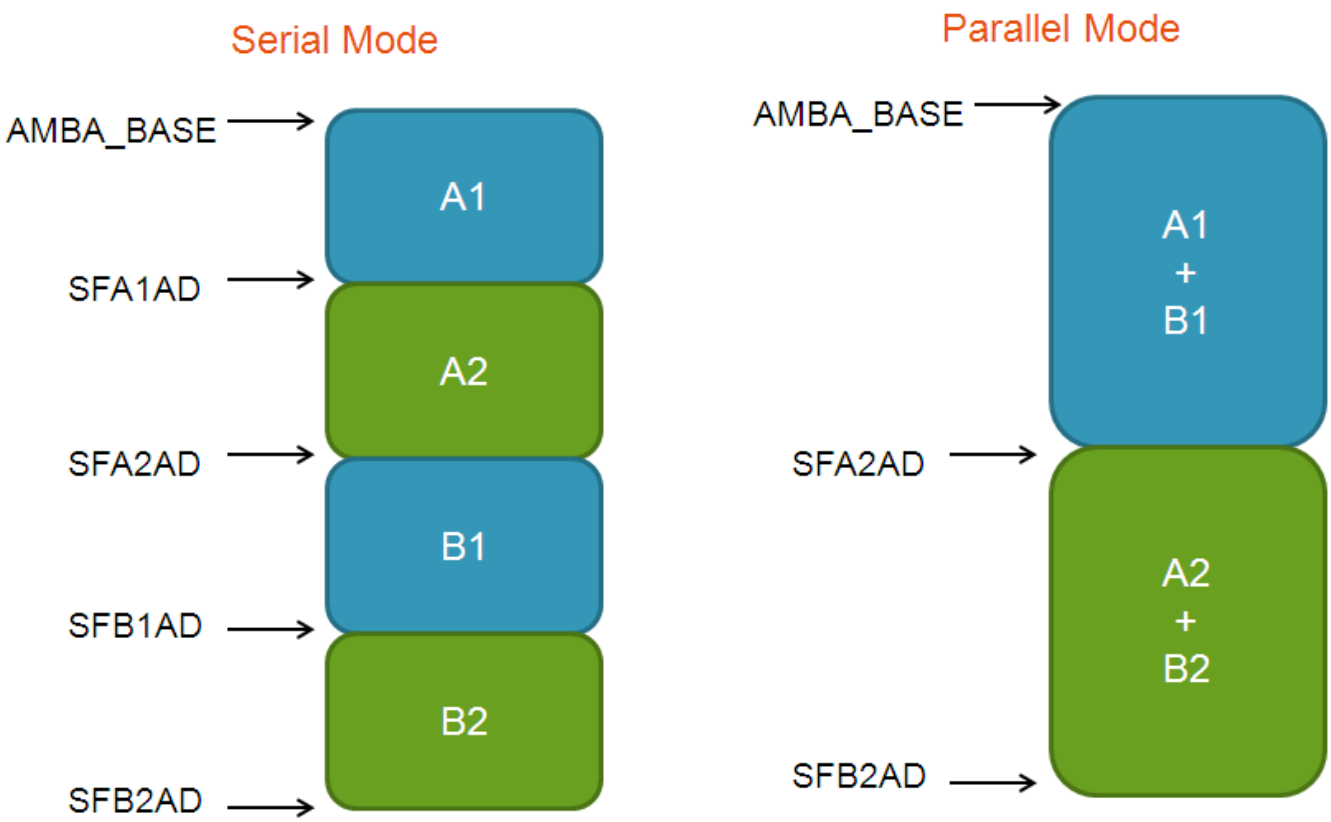


Figure 10-20. Memory map - Serial and Parallel

10.2.6 Byte Ordering - Endianness

QuadSPI provides support for swapping the flash read/write data based on the configuration of the QSPI_MCR[END_CFG]. By default the data is always returned in 64 bit LE format on the AHB bus and 32 bit LE format on the IPS interface when read via the RX buffer and written in 32 bit LE format when written via the TX buffer.

The table(QSPI_MCR[END_CFG]) below shows the complete bit ordering. BE signifies Big Endian which means the high order bits of the associated data vectors are associated with low order address positions. LE signifies Little Endian which means the lower order bits of the associated data vectors are associated with low order address positions.Refer to figure Figure 10-18

Table 10-23. QSPI_MCR[END_CFG]

00	64 bit BE
01	32 bit LE
10	32 bit BE
11	64 bit LE

The tables below (Byte ordering configuration in AHB) and (Byte ordering configuration in IPS) show how this configuration is implemented in QSPI AHB and IPS interfaces respectively. B in the table signifies Byte and the index 1-8 refers to the byte position i.e. 1 refer to bits[7:0], 8 refer to bits[63:56] and so on.

Table 10-24. Byte ordering configuration in AHB

64 bit BE	B1	B2	B3	B4	B5	B6	B7	B8
64 bit LE	B8	B7	B6	B5	B4	B3	B2	B1
32 bit BE	B5	B6	B7	B8	B1	B2	B3	B4
32 bit LE	B4	B3	B2	B1	B8	B7	B6	B5

Table 10-25. Byte ordering configuration in IPS

32BE	B1	B2	B3	B4
32LE	B4	B3	B2	B1

The examples below show the byte ordering in 64 bit BE configuration for AHB Buffer and 32 bit BE for TX/RX Buffer:

10.2.6.1 Programming Flash Data

CPU write instructions to the QSPI_TBDR register like

- Write QSPI_TBDR -> 0x01_02_03_04
- Write QSPI_TBDR -> 0x05_06_07_08

result in the following content of the TX Buffer:

Table 10-26. Example of QuadSPI TX Buffer

TX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

Programming the TX Buffer into the external serial flash device results in the following byte order to be sent to the serial flash:

- 01…02…03…04…05…06…07…08

10.2.6.2 Reading Flash Data into the RX Buffer

Reading the content from the same address provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the RX Buffer filled with:

Table 10-27. Resulting RX Buffer Content

RX Buffer Entry	Content
0	32'h01_02_03_04
1	32'h05_06_07_08

10.2.6.2.1 Readout of the RX Buffer via QSPI_RBDRn

The RX Buffer content appears at CPU read access via the Peripheral bus interface in the following order:

- Read QSPI_RBDR0 <- 0x01_02_03_04
- Read QSPI_RBDR1 <- 0x05_06_07_08

10.2.6.2.2 Readout of the RX Buffer via ARDBn

The RX Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04

- (2a): 32 Bit Access: Read QSPI_ARDB1 <- 0x05_06_07_08
- (1b/2b): 64 Bit Access: Read QSPI_ARDB0 <- 0x01_02_03_04_05_06_07_08

10.2.6.3 Reading Flash Data into the AHB Buffer

Reading the content from the same address as it was written to provides the following sequence of bytes, identical to the write case:

- 01…02…03…04…05…06…07…08

This results in the AHB Buffer filled with:

Table 10-28. Resulting AHB Buffer Content

AHB Buffer Entry	Content
0	64'h01_02_03_04_05_06_07_08

10.2.6.3.1 Readout of the AHB Buffer via Memory Mapped Read

The AHB Buffer content appears at read access on the AMBA AHB interface at the QuadSPI module boundary:

- (1a): 32 Bit Read Access: <- 0x01_02_03_04
- (2a): 32 Bit Read Access: <- 0x05_06_07_08
- (1/2): 64 Bit Read Access: <- 0x01_02_03_04_05_06_07_08

10.2.7 Serial Flash Devices

Several different vendors make flash devices with a QuadSPI interface. At present there is no set standard for the QuadSPI instruction set. Most common commands currently have the same instruction code for all vendors, however some commands are unique to specific vendors. Some example sequences are provided below.

10.2.7.1 Example Sequences

This section provides the example sequences of the QuadSPI module.

Table 10-29. Exit 4 x I/O Read Enhance Performance Mode (XIP) (Macronix) and Read Status

INSTR	PAD	OPERAND	COMMENT
CMD	0x0	0xEB	4xIO Read Command
ADDR	0x2	0x18	24 Bit address to be send on 4 pads
MODE	0x2	0x00	2 mode cycles (exit XIP)
DUMMY	0x0	0x04	4 dummy cycles
READ	0x2	0x08	Read 64 bits
CMD	0x0	0x05	Read Status register
READ	0x0	0x01	Status register data
STOP	0x0	0x00	STOP, Instruction over

10.2.7.1.1 Fast Read Sequence (Macronix/Numonyx/Spansion/Winbond)

The following table shows the fast read sequence for Macronix/Numonyx/Spansion/ Winbond flashes.

Table 10-30. Fast Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x0B	Fast Read command = 0x0B
ADDR	0x0	0x18	24 Addr bits to be sent on one pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x0	0x04	Read 32 Bits on one pad
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

10.2.7.1.2 Fast Dual I/O DT Read Sequence (Macronix)

The following table shows the Fast Dual I/O DT read sequence for Macronix flashes.

Table 10-31. Fast Dual I/O DT Read sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xBD	Fast Dual I/O DT read command = 0xBD
ADDR_DDR	0x1	0x18	24 Addr bits to be sent on 2 pads in DDR mode
MODE4_DDR	0x1	0x00	P2=P0 or P3=P1 is necessary. Refer to Macronix datasheet for details. One clock cycle for mode.
DUMMY	0x0	0x06	6 Dummy cycles

Table continues on the next page...

Table 10-31. Fast Dual I/O DT Read sequence (continued)

Instruction	Pad	Operand	Comment
READ_DDR	0x1	0x04	Read 32 Bits on 2 pads in DDR mode
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

10.2.7.1.3 Fast Read Quad Output (Winbond)

The following table shows the Fast read quad output sequence for Winbond memories

Table 10-32. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x6B	Fast read quad output command = 0x6B
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
DUMMY	0x0	0x08	8 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x00	Jump to instruction 0 (CMD)

10.2.7.1.4 4 x I/O Read Enhance Performance Mode (XIP) (Macronix)

The following table shows the 4 x I/O Read Enhance Performance Mode for Macronix flashes. The enhanced performance mode is also known as XIP mode.

Table 10-33. Fast Read Quad output sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xEB	4xI/O Read command = 0xEB
ADDR	0x2	0x18	24 Addr bits to be sent on 4 pads
MODE	0x2	0xA5	2 mode cycles
DUMMY	0x0	0x04	4 Dummy cycles
READ	0x2	0x04	Read 32 Bits on 4 pads
JMP_ON_CS	0x0	0x01	Jump to instruction 1 (ADDR)

When in XIP mode the software should ensure that all the flashes connected to the controller are in XIP mode. As a part of initializing the controller, all the flashes may be enabled with XIP by carrying out dummy reads.

10.2.7.1.5 Dual Command Page Program (Numonyx)

The following table shows the Dual command page program sequence for Numonyx flashes.

Table 10-34. Dual Command Page Program sequence

Instruction	Pad	Operand	Comment
CMD	0x1	0x02	Dual command page program = 0x02 on 2 pads
ADDR	0x1	0x18	24 Addr bits to be sent on 2 pads
WRITE	0x1	0x20	Write 32 Bytes on 2 pads
STOP	0x0	0x00	STOP, Instruction over

10.2.7.1.6 Sector Erase (Macronix/Spansion/Numonyx)

The following table shows the Sector erase sequence for Macronix/Spansion/Numonyx flashes

Table 10-35. Sector Erase sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0xD8	Sector erase command = 0xD8
ADDR	0x0	0x18	24 Addr bits to be sent on 1 pad
STOP	0x0	0x00	STOP, Instruction over

10.2.7.1.7 Read Status Register (Macronix/Spansion/Numonyx/Winbond)

The following table shows the Read status register sequence for Macronix/Spansion/ Numonyx/Winbond flashes.

Table 10-36. Read Status Register Sequence

Instruction	Pad	Operand	Comment
CMD	0x0	0x05	Read status register command = 0x05
READ	0x0	0x01	Read status register data
STOP	0x0	0x00	STOP, Instruction over

10.2.7.2 Dual Die Flashes

Certain serial flash vendors provide dual-die packages which are essentially two devices (dies) stacked within the same package to increase the memory capacity of a single package. These two devices within a package share the same data and clock pins, but have individual Chip Selects. QuadSPI controller provides support for two dual-die packages to be connected simultaneously. The figure below shows the two dual-die packages and the naming conventions used in this document. For simplicity, the data pins are shown to be unidirectional.

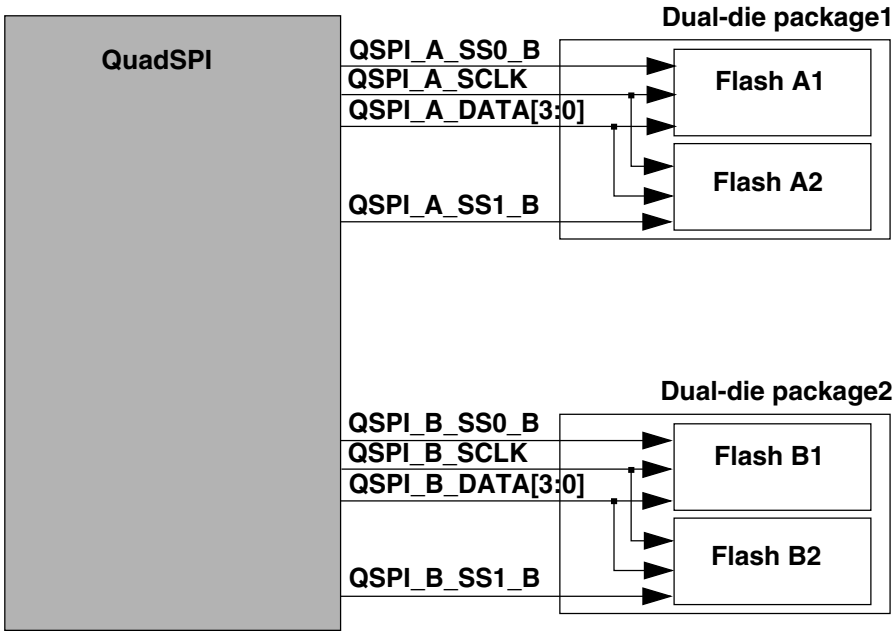


Figure 10-21. Dual-die support

Since the two devices within one package share the same i/o pads, they cannot function in parallel mode. Software should ensure that when QuadSPI is configured in parallel mode the two selected flash devices are from different dual-die packages.

10.2.7.3 Boot initialization sequence

The following are the recommended sequence of steps for booting from QuadSPI:

- System out of reset and flash available (300us)
- Clocks still at very low frequency. Clock tree configured, I/O pins configured. First request sent to QuadSPI for address 0x0 of flash.
- The reset command sequence in QuadSPI has 0x03 (basic read command) which is applicable to all flashes at < 50MHz serial flash clock
- The first few bytes of data is read from the flash which contains the following information:
  - The total sizes of all the flashes connected on board
  - Whether DDR mode supported
  - Frequency of DDR operation

- Continuous mode entry sequence
- 24bit or 32bit addressing (assuming 24bit for first accesses)

- All the serial flashes are configured
  - Quad Mode enabled
  - Dummy reads to enter into XIP
- QuadSPI is configured
  - Parallel enable set
  - LUT configured for highest performance reads
  - DDR mode enabled (if applicable)
  - Buffers configured
- Serial flash clock frequency increased.
- Boot reads happen in parallel, DDR enabled, quad output mode @66MHz.

10.2.8 Sampling of Serial Flash Input Data

10.2.8.1 Internal Sampling of Serial Flash Input Data

Depending from the actual implementation there is a delay between the internal clocking in the QuadSPI module and the external serial flash device.Refer to the following figure for an overview of this scheme.

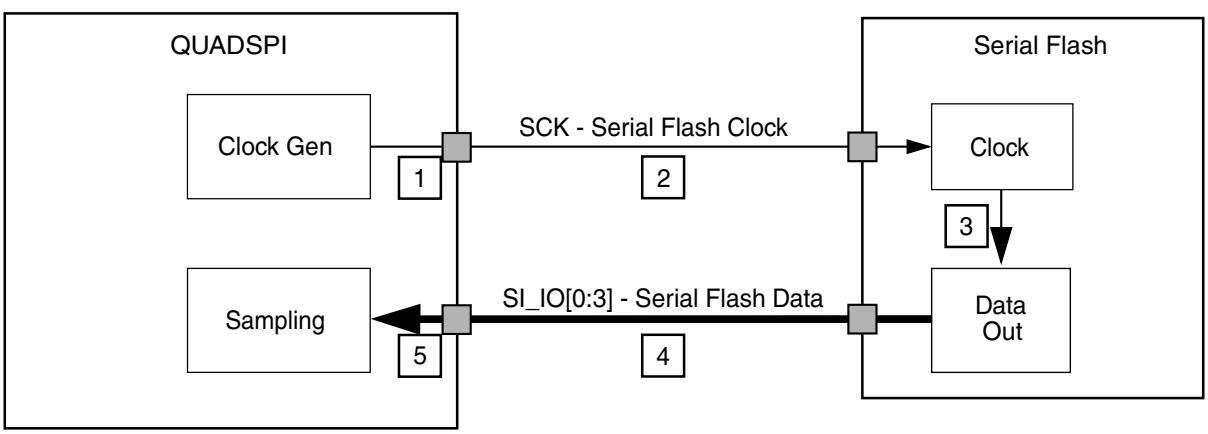


Figure 10-22. Serial Flash Sampling Clock Overview

Note

The arrival of the serial flash data in the sampling stage of the QuadSPI module are given in the following figure. Note that the amount of the total delay $t_{Del,total}$ is very specific to the characteristics of the actual implementation.

Note also that the serial flash device clock SCK is inverted with respect to the QuadSPI internal reference clock.

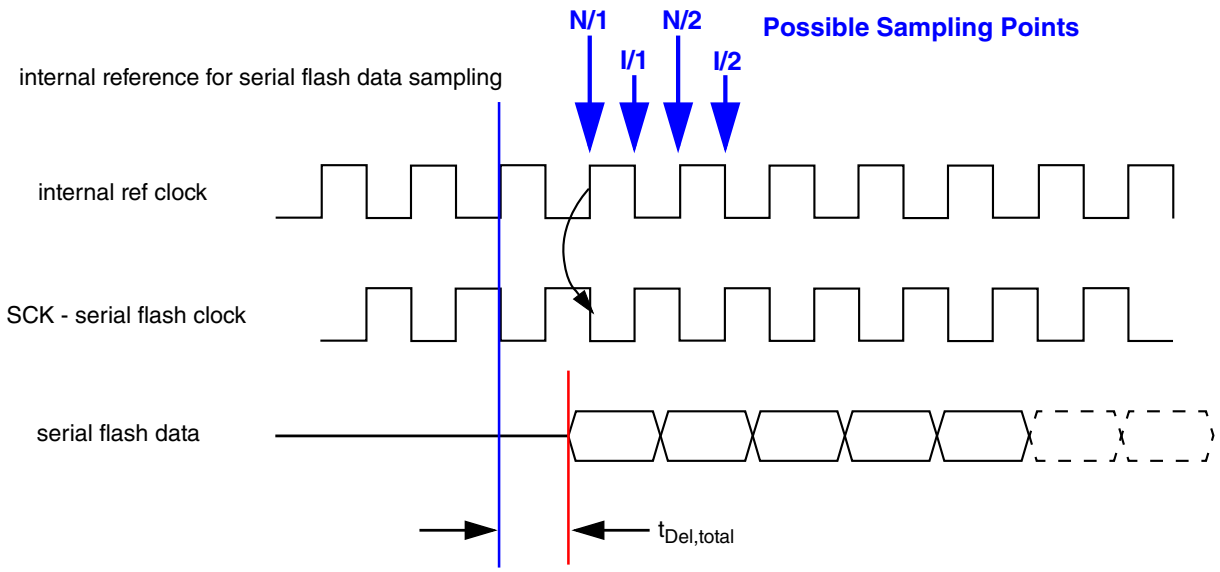


Figure 10-23. Serial Flash Sampling Clock Timing

The rising edge of the internal reference clock is taken as timing reference for the data output of the serial flash. After a time of $t_{Del,total}$ the data arrive at the internal sampling stage of the QuadSPI module.

According to the Serial Flash Sampling Clock Overview figure, the following parts of the delay chain contribute to $t_{Del,total}$:

1. Output delay of the serial flash clock output of the device containing the QuadSPI module
2. Wire delay of application/PCB from the device containing the QuadSPI module to the external serial flash device
3. Clock to data out delay of the external serial flash device, including input and output delays
4. Wire delay of application/PCB from the external serial flash device to the device containing the QuadSPI module
5. Input delay belonging to the data in input

The possible points in time for the sampling of the incoming data are denoted as N/1, I/1, N/2 and I/2 above. The sampling point relevant for the internal sampling is configured in the QSPI_SMPR register, refer to Sampling Register (QuadSPI_SMPR) for details. Note that the falling edges of the reference clock are not actually used, instead the inverted clock is used for sampling at these positions. The following table gives an overview of the available configurations for the commands running at regular (full) speed:

Table 10-37. Sampling Configuration

Sampling Point	Description	Delay [FSDLY] [HSDLY]	Phase [FSPHS] [HSPHS]	QSPI_SMPR for Full Speed Setting[1]
N/1	sampling with non-inverted clock, 1 sample delay	0	0	0x0000000x
I/1	sampling with inverted clock, 1 sample delay	0	1	0x0000002x
N/2	sampling with non-inverted clock, 2 samples delay	1	0	0x0000004x
I/2	sampling with inverted clock, 2 samples delay	1	1	0x0000006x

1. 'x' is not considered here

Depending from the actual delay and the serial flash clock frequency the appropriate sampling point can be chosen. The following remarks should be considered when selecting the appropriate setting:

- Theoretically there should be 2 settings possible to capture the correct data since the serial flash output is valid for 1 clock cycle, disregarding rise and fall times and timing uncertainties.
- Depending from the timing uncertainties it may turn out in actual applications that only one possible sample positions remains. This is subject to careful consideration depending from the actual implementation.
- The delay $t_{Del,total}$ is an absolute size to shift the point in time when the serial flash date get valid at the QuadSPI input.
- For decreasing frequency of the serial flash clock the distance between the edges increases. So for large differences in the frequency the required setting may change.
- For commands running at half of the regular serial flash clock (QSPI_SMPR[HSENA] bit set) the sampling point must be figured separately to allow for the compensation of the absolute shift in time with respect to the sample-relative setting in the QSPI_SPMPR register.

10.2.8.2 DDR Mode

The increasing requirement of improved throughput has introduced the double data rate (DDR) mode. In DDR mode, the data is transferred on both the rising and falling edges of the serial flash clock. The DDR serial flashes sample as well as drive the data on both rising and falling edges of serial flash clock.

10.2.9 Serial Flash Data Input Timing

There are sampling modes for input flash data:

- Internal sampling - Input serial flash data is captured by internal serial clock. In SDR mode, serial data is sampled by serial clock 1x (ser_clk_1x) rise edge. In DDR mode, serial data is sampled by serial clock 4x (ser_clk_4x) rise edge.
- Loopback DQS sampling - Soc will output serial data strobe with internal serial clock. This serial data strobe would be loopback from pad and used to sample input serial data. In SDR mode, serial data is sampled by loopback DQS rise edge. In DDR mode, serial data is sampled by loopback DQS both edge

NOTE

DQS pad need to be set to force input for loopback.

- Flash DQS sampling - Some serial Flash device provide the data strobe output together with serial data. This strobe signal is used to sample input serial data directly. In SDR mode, serial data is sampled by Flash DQS rise edge. In DDR mode, serial data is sampled by Flash DQS both edge

Following table shows selection for sampling mode:

Sampling mode	DQS_EN	DQS_LOOPBACK_EN
Internal sampling	0	doesn't matter
Loopback DQS sampling	1	1
Flash DQS sampling	1	0

Serial flash data and clock path for input timing is shown in the following figure.

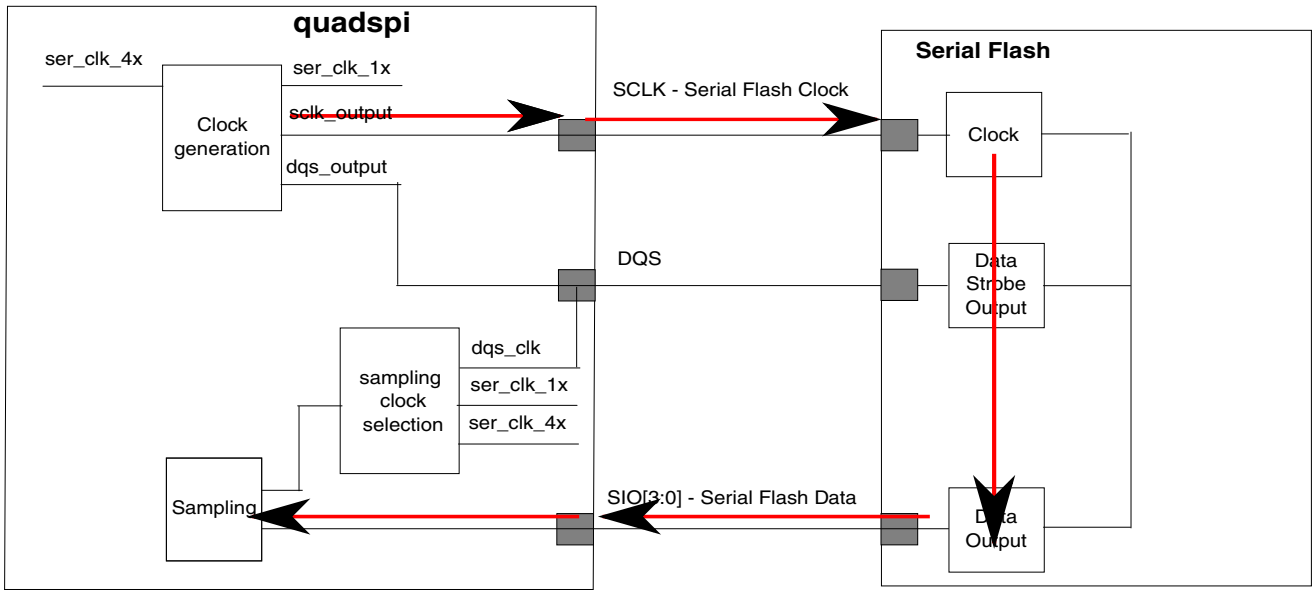


Figure 10-24. Serial flash data and clock path

NOTE

The red line is for data input path, the blue line is for DQS loopback path, and the purple line is for DQS input path from Serial Flash.

Total delay (Ttotal_delay_data) for serial flash data input is the sum of following delay:

1. Output delay of serial flash clock from internal serial clock to SCLK pad inside SOC
2. Wire delay of serial flash clock (SCLK) from SOC to external Serial Flash Device
3. Clock to Output Valid time of external Serial Flash Device
4. Wire delay of serial flash data (SIO) from external serial Flash Device to SOC
5. Input delay of serial flash data from SIO pad to internal sample register

Total delay (Ttotal_delay_loopback_dqs) for loopback DQS clock is the sum of following delay:

1. Output delay of DQS clock from internal serial clock to DQS pad inside SOC
2. Input delay of DQS clock from DQS pad to internal sample register

Total delay (Ttotal_delay_flash_dqs) for Flash DQS clock is the sum of following delay:

1. Output delay of serial flash clock from internal serial clock to SCLK pad inside SOC
2. Wire delay of serial flash clock (SCLK) from SOC to external Serial Flash Device
3. Clock to DQS Output time of external Serial Flash Device
4. Wire delay of serial flash data strobe (DQS) from external serial Flash Device to SOC
5. Input delay of DQS clock from DQS pad to internal sample register

10.2.9.1 Input timing in SDR mode with internal sampling

Input Timing diagram in SDR mode with internal sampling is show in the figure below.

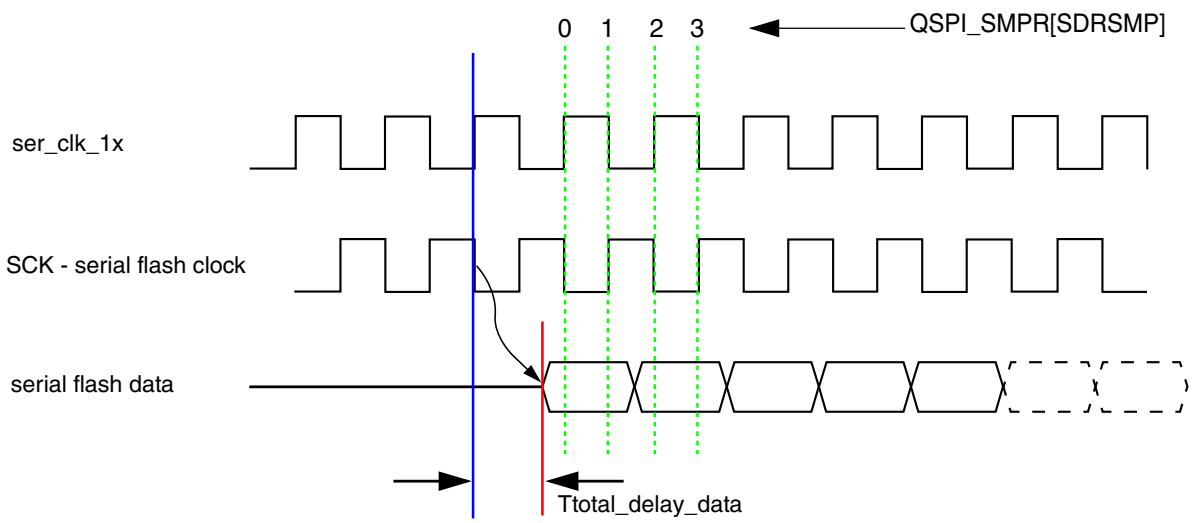


Figure 10-25. Internal sample SDR

There are four sample points for this sampling mode, which is determined by register field QSPI_SMPR.SDRSMP.

Sampling point need to be select correctly to meet both Setup and Hold timing for internal sample registers:

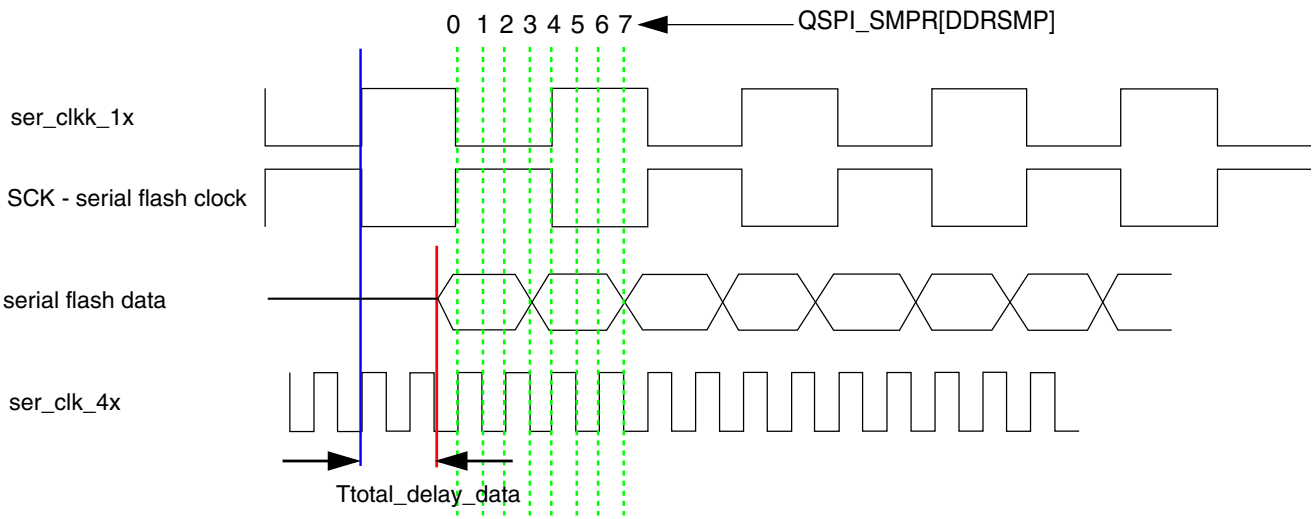
- For sample point N, Setup requirement is: Tcycle*(N+2)/2>Ttotal_delay_data,max
- For sample point N, Hold requirement is: Ttotal_delay_data,min>Tcycle*N/2

NOTE

Tcycle is the cycle of ser_clk_1x. Ttotal_delay_data,max is maximum delay of serial data input path. Ttotal_delay_data,min is the minimum delay of serial data input path. N=0,1,2,3

10.2.9.2 Input timing in DDR mode with internal sampling

Input Timing diagram in SDR mode with internal sampling is show in the following figure.



There are 8 sample points for this sampling mode, which is determined by register field DDRSMP.

Sampling point need to be select correctly to meet both Setup and Hold timing for internal sample registers:

- For sample point N, Setup requirement is: Tcycle * (N+2)/8 >Ttotal_delay_data,max
- For sample point N, Hold requirement is: Ttotal_delay_data,min> Tcycle * N/8

10.2.9.3 Input timing in SDR mode with loopback DQS sampling

Input Timing diagram in SDR mode with internal sampling is show in the following figure.

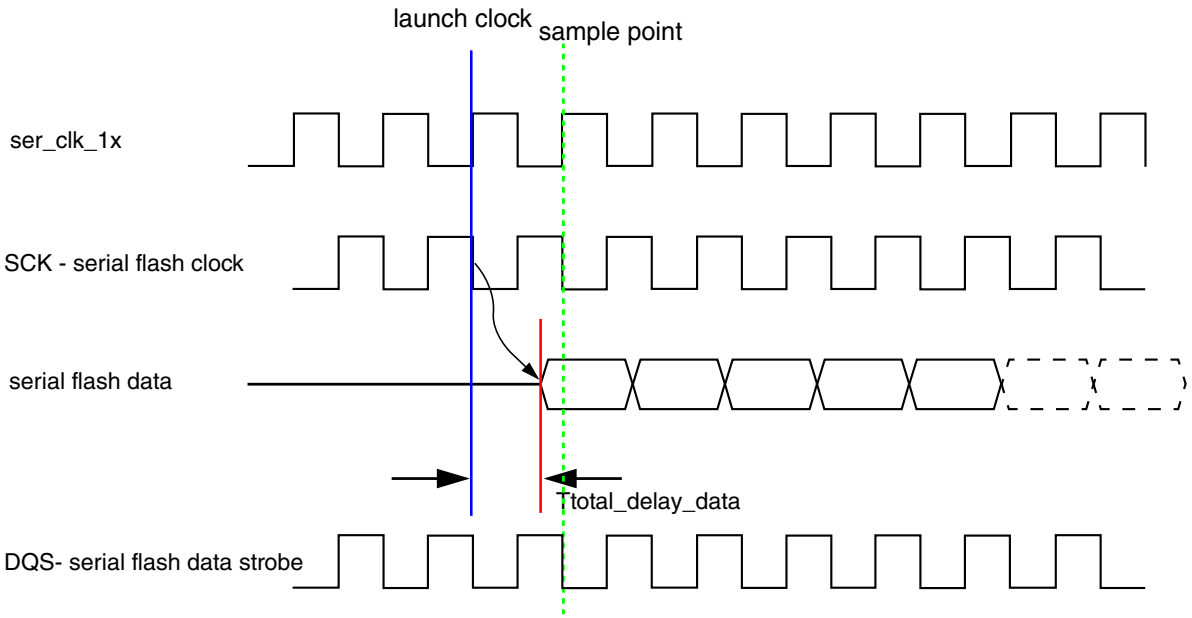


Figure 10-26. Input timing in SDR mode with loopback DQS sampling

In SDR mode and loopback sampling mode, DQS_PHASE_EN should be set 1.

For this sample point, the Setup requirement is: Tcycle >max(Ttotal_delay_data-Ttotal_delay_loopback_dqs). The Hold requirement is: min(Ttotal_delay_data-Ttotal_delay_loopback_dqs)> 0.

10.2.9.4 Input timing in DDR mode with loopback DQS sampling

Input Timing diagram in DDR mode with internal sampling is show in the following figure.

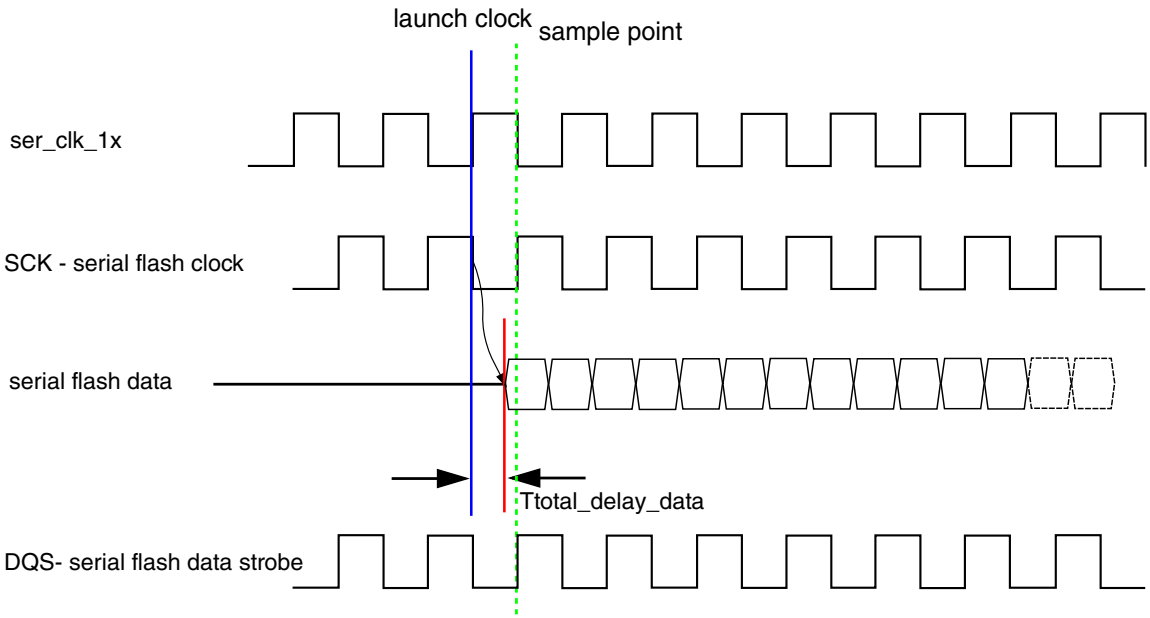


Figure 10-27. Input timing in DDR mode with loopback DQS sampling

In DDR mode and loopback sampling mode, DQS_PHASE_EN should be set 0.

For this sample point, the Setup requirement is: Tcycle >max(Ttotal_delay_data-Ttotal_delay_loopback_dqs). The Hold requirement is: min(Ttotal_delay_data-Ttotal_delay_loopback_dqs)> 0.

10.2.9.5 Input timing in SDR mode with flash DQS sampling

Input Timing diagram in SDR mode with internal sampling is show in the following figure.

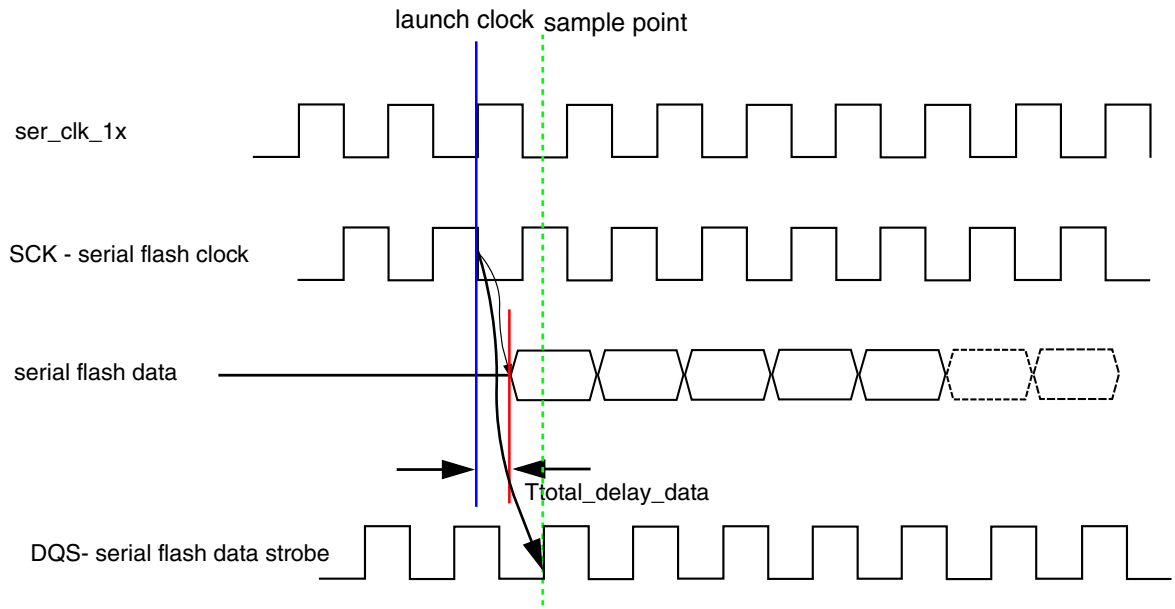


Figure 10-28. Input timing in SDR mode with flash DQS sampling

In this sampling mode, there will be a setup/hold requirement on Flash serial Data and Flash serial Data Strobe. This value will be specified in the data sheet.

10.2.9.6 Input timing in DDR mode with flash DQS sampling

Input Timing diagram in DDR mode with internal sampling is show in the following figure.

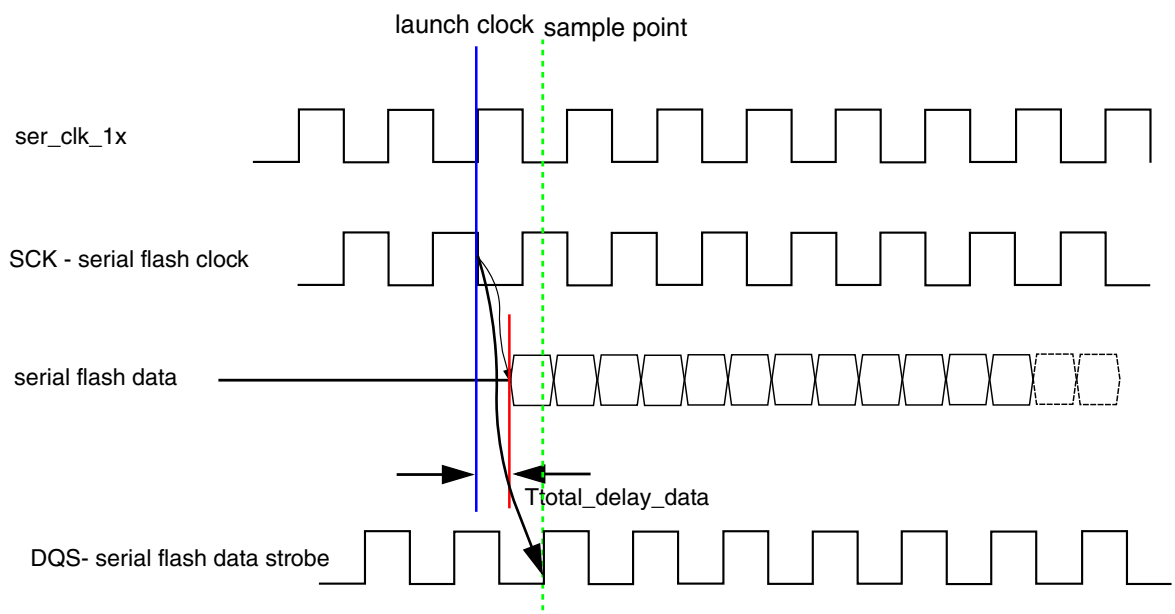


Figure 10-29. Input timing in DDR mode with flash DQS sampling

In this sampling mode, there will be a setup/hold requirement on Flash serial Data and Flash serial Data Strobe. This value will be specified in the data sheet.

10.2.9.7 Data Strobe Signal functionality

Some external serial flashes provide the data strobe (DQS/RDS) output which is fed directly to the QuadSPI module. The strobe (DQS/RDS) signal needs to be delayed to have the edges aligned to the data valid period. QuadSPI internally samples the incoming data at posedge of the strobe signal for SDR and on both the edges of the strobe signal for DDR. Refer to the figure for more detail.

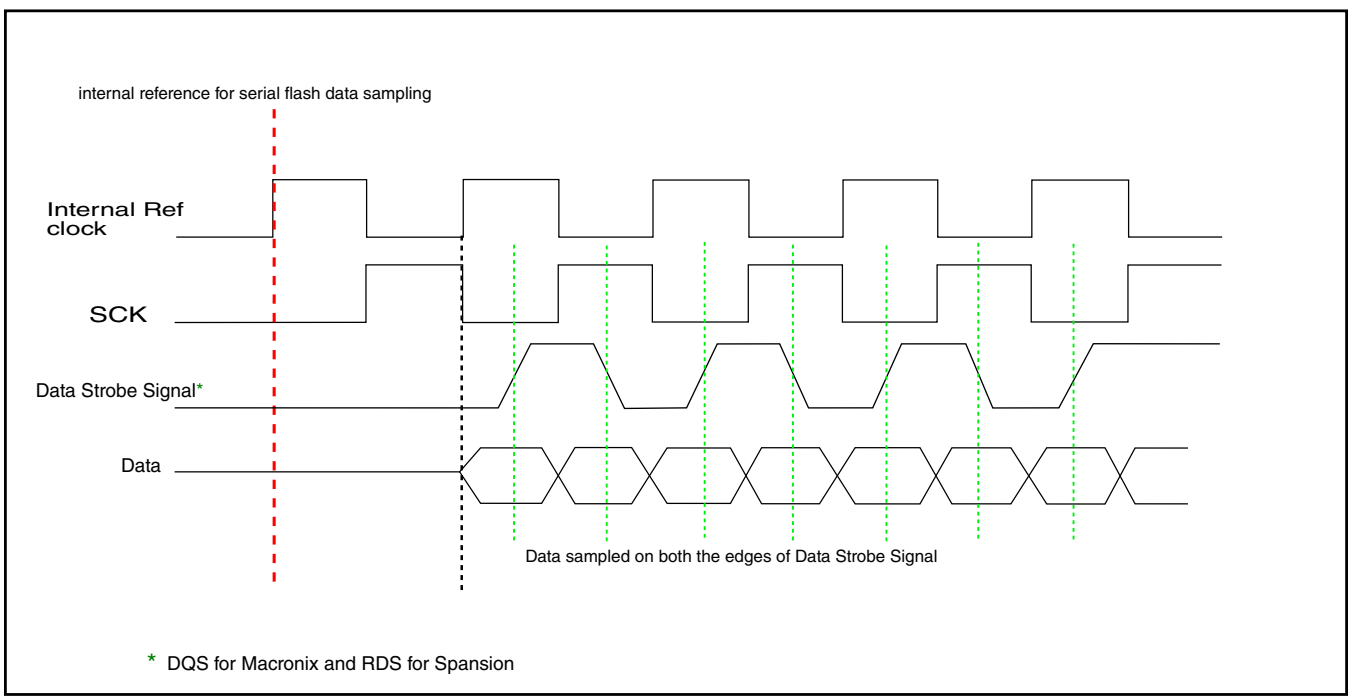


Figure 10-30. Data strobe signal functionality

10.2.10 Output timing in SDR mode

Output timing diagram in SDR mode is show in the following figure.

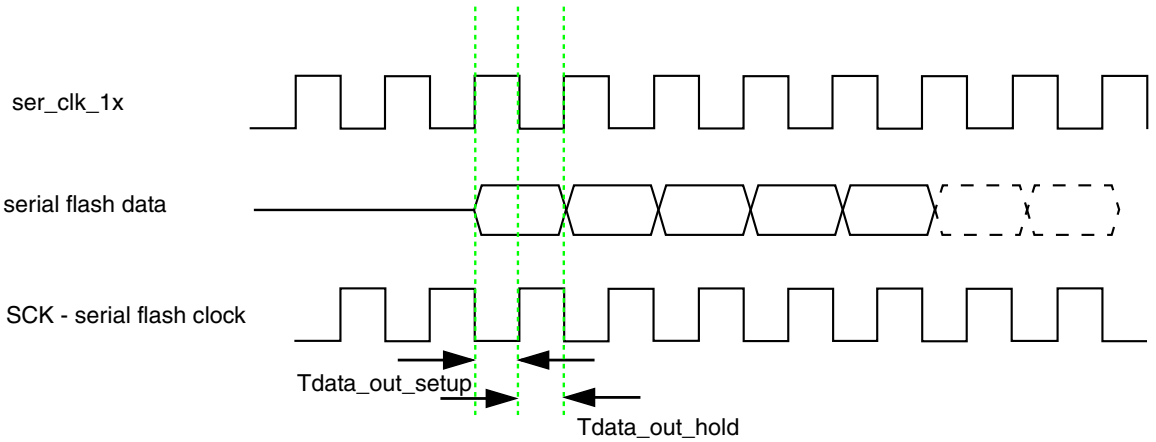


Figure 10-31. Output timing in SDR mode

In SDR mode, quadspi output serial data with internal serial clock rise edge 1x (ser_clk_1x). Flash Device has requirement on Data and Clock setup and hold timing.

10.2.11 Output timing in DDR mode

Output Timing diagram in DDR mode is show in the following figure.

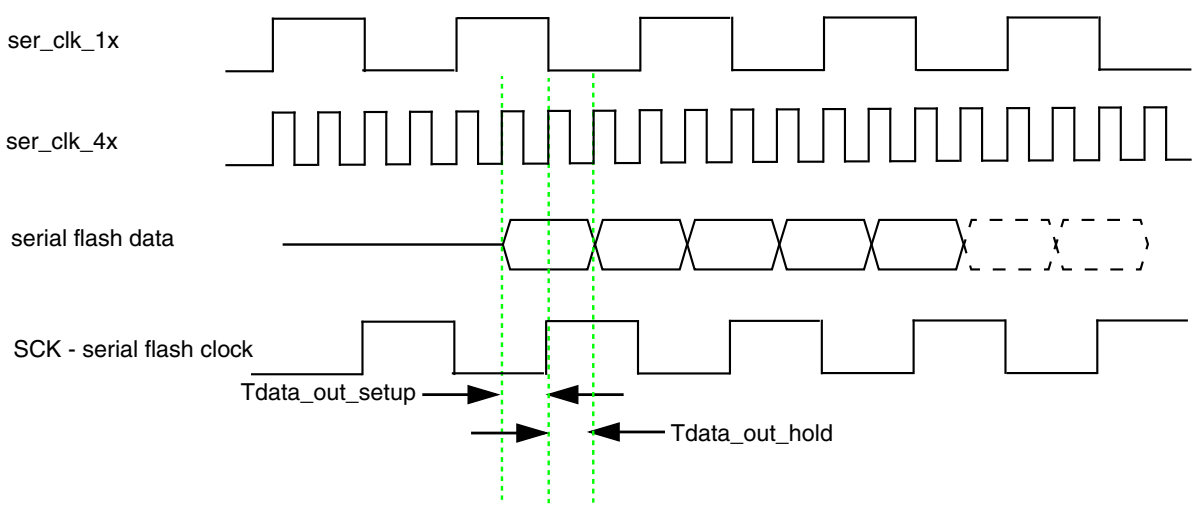


Figure 10-32. Output timing in DDR mode

In DDR mode, quadspi output serial data with internal serial clock both edge 1x (ser_clk_1x) and then delay one ser_clk_4x cycle for hold timing.

NOTE

TX_DDR_DELAY_EN should be set to 1 for DDR mode.

10.2.12 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)

10.2.12.1 AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31)

NOTE

See the System Memory map in this document for the base address of the QSPI AHB RX Data Buffer.

memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	AHB RX Data Buffer register (ARDB0)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
4	AHB RX Data Buffer register (ARDB1)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
8	AHB RX Data Buffer register (ARDB2)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
C	AHB RX Data Buffer register (ARDB3)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
10	AHB RX Data Buffer register (ARDB4)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
14	AHB RX Data Buffer register (ARDB5)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
18	AHB RX Data Buffer register (ARDB6)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
1C	AHB RX Data Buffer register (ARDB7)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
20	AHB RX Data Buffer register (ARDB8)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
24	AHB RX Data Buffer register (ARDB9)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
28	AHB RX Data Buffer register (ARDB10)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
2C	AHB RX Data Buffer register (ARDB11)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
30	AHB RX Data Buffer register (ARDB12)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
34	AHB RX Data Buffer register (ARDB13)	32	R/W	0000_0000h	10.2.12.1.1/ 2580

Table continues on the next page...

memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
38	AHB RX Data Buffer register (ARDB14)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
3C	AHB RX Data Buffer register (ARDB15)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
40	AHB RX Data Buffer register (ARDB16)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
44	AHB RX Data Buffer register (ARDB17)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
48	AHB RX Data Buffer register (ARDB18)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
4C	AHB RX Data Buffer register (ARDB19)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
50	AHB RX Data Buffer register (ARDB20)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
54	AHB RX Data Buffer register (ARDB21)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
58	AHB RX Data Buffer register (ARDB22)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
5C	AHB RX Data Buffer register (ARDB23)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
60	AHB RX Data Buffer register (ARDB24)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
64	AHB RX Data Buffer register (ARDB25)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
68	AHB RX Data Buffer register (ARDB26)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
6C	AHB RX Data Buffer register (ARDB27)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
70	AHB RX Data Buffer register (ARDB28)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
74	AHB RX Data Buffer register (ARDB29)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
78	AHB RX Data Buffer register (ARDB30)	32	R/W	0000_0000h	10.2.12.1.1/ 2580
7C	AHB RX Data Buffer register (ARDB31)	32	R/W	0000_0000h	10.2.12.1.1/ 2580

10.2.12.1.1 AHB RX Data Buffer register (ARDB*n*)

The AHB RX Data Buffer register 0 to 31 can be used to read the buffer content of the RX Buffer from successive addresses. QSPI_ARDB0 corresponds to the RX Buffer register entry corresponding to the current value of the read pointer with increasing order.

The increment of the read pointer depends from the access scheme (DMA or flag-driven). Refer to "Data Transfer from the QuadSPI Module Internal Buffers" section in Flash Read section, RX Buffer, data read via register interface and AHB read, for the description of successive accesses to the RX Buffer content. Refer also to Byte Ordering of Serial Flash Read Data for the byte ordering scheme.

Valid address range accessible in the QSPI_ARDBn range depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

- Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QSPI_ARDB0 to QSPI_ARDB31.
- Example 2, RX Buffer filled with 5 valid words, RX Buffer fill level QSPI_RBSR[RDBFL] is 5. In this case an access to QSPI_ARDB4 provides the last valid entry.

Address: 0h base + 0h offset + (4d × i), where i=0d to 31d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ARXD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ARDB*n* field descriptions

Field	Description
ARXD	ARDB provided RX Buffer Data. Byte order (endianness) is identical to the RX Buffer Data Registers.

10.2.13 Peripheral Bus Register Descriptions

This section provides the peripheral bus register information of the QuadSPI module.

This section provides the memory map and register definitions of the QuadSPI module.

QuadSPI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_0000	Module Configuration Register (QuadSPI_MCR)	32	R/W	000F_4000h	10.2.13.1/ 2588
30BB_0008	IP Configuration Register (QuadSPI_IPCR)	32	R/W	0000_0000h	10.2.13.2/ 2591

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_000C	Flash Configuration Register (QuadSPI_FLSHCR)	32	R/W	0000_0303h	10.2.13.3/ 2591
30BB_0010	Buffer0 Configuration Register (QuadSPI_BUF0CR)	32	R/W	0000_0000h	10.2.13.4/ 2592
30BB_0014	Buffer1 Configuration Register (QuadSPI_BUF1CR)	32	R/W	0000_0000h	10.2.13.5/ 2593
30BB_0018	Buffer2 Configuration Register (QuadSPI_BUF2CR)	32	R/W	0000_0000h	10.2.13.6/ 2594
30BB_001C	Buffer3 Configuration Register (QuadSPI_BUF3CR)	32	R/W	See section	10.2.13.7/ 2595
30BB_0020	Buffer Generic Configuration Register (QuadSPI_BFGENCR)	32	R/W	0000_0000h	10.2.13.8/ 2596
30BB_0030	Buffer0 Top Index Register (QuadSPI_BUF0IND)	32	R/W	0000_0000h	10.2.13.9/ 2597
30BB_0034	Buffer1 Top Index Register (QuadSPI_BUF1IND)	32	R/W	0000_0000h	10.2.13.10/ 2597
30BB_0038	Buffer2 Top Index Register (QuadSPI_BUF2IND)	32	R/W	0000_0000h	10.2.13.11/ 2598
30BB_0100	Serial Flash Address Register (QuadSPI_SFAR)	32	R/W	0000_0000h	10.2.13.12/ 2599
30BB_0108	Sampling Register (QuadSPI_SMPR)	32	R/W	0000_0000h	10.2.13.13/ 2599
30BB_010C	RX Buffer Status Register (QuadSPI_RBSR)	32	R	0000_0000h	10.2.13.14/ 2600
30BB_0110	RX Buffer Control Register (QuadSPI_RBCT)	32	R/W	0000_0000h	10.2.13.15/ 2601
30BB_0150	TX Buffer Status Register (QuadSPI_TBSR)	32	R	0000_0000h	10.2.13.16/ 2602
30BB_0154	TX Buffer Data Register (QuadSPI_TBDR)	32	R/W	0000_0000h	10.2.13.17/ 2602
30BB_015C	Status Register (QuadSPI_SR)	32	R	0000_3800h	10.2.13.18/ 2604
30BB_0160	Flag Register (QuadSPI_FR)	32	w1c	0800_0000h	10.2.13.19/ 2607
30BB_0164	Interrupt and DMA Request Select and Enable Register (QuadSPI_RSER)	32	R/W	0000_0000h	10.2.13.20/ 2610
30BB_0168	Sequence Suspend Status Register (QuadSPI_SPNDST)	32	R	0000_0000h	10.2.13.21/ 2613
30BB_016C	Sequence Pointer Clear Register (QuadSPI_SPTRCLR)	32	R/W	0000_0000h	10.2.13.22/ 2615
30BB_0180	Serial Flash A1 Top Address (QuadSPI_SFA1AD)	32	R/W	0000_0000h	10.2.13.23/ 2615
30BB_0184	Serial Flash A2 Top Address (QuadSPI_SFA2AD)	32	R/W	0000_0000h	10.2.13.24/ 2616

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_0188	Serial Flash B1Top Address (QuadSPI_SFB1AD)	32	R/W	0000_0000h	10.2.13.25/ 2616
30BB_018C	Serial Flash B2Top Address (QuadSPI_SFB2AD)	32	R/W	0000_0000h	10.2.13.26/ 2617
30BB_0200	RX Buffer Data Register (QuadSPI_RBDR0)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0204	RX Buffer Data Register (QuadSPI_RBDR1)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0208	RX Buffer Data Register (QuadSPI_RBDR2)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_020C	RX Buffer Data Register (QuadSPI_RBDR3)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0210	RX Buffer Data Register (QuadSPI_RBDR4)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0214	RX Buffer Data Register (QuadSPI_RBDR5)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0218	RX Buffer Data Register (QuadSPI_RBDR6)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_021C	RX Buffer Data Register (QuadSPI_RBDR7)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0220	RX Buffer Data Register (QuadSPI_RBDR8)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0224	RX Buffer Data Register (QuadSPI_RBDR9)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0228	RX Buffer Data Register (QuadSPI_RBDR10)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_022C	RX Buffer Data Register (QuadSPI_RBDR11)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0230	RX Buffer Data Register (QuadSPI_RBDR12)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0234	RX Buffer Data Register (QuadSPI_RBDR13)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0238	RX Buffer Data Register (QuadSPI_RBDR14)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_023C	RX Buffer Data Register (QuadSPI_RBDR15)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0240	RX Buffer Data Register (QuadSPI_RBDR16)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0244	RX Buffer Data Register (QuadSPI_RBDR17)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0248	RX Buffer Data Register (QuadSPI_RBDR18)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_024C	RX Buffer Data Register (QuadSPI_RBDR19)	32	R/W	0000_0000h	10.2.13.27/ 2617

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_0250	RX Buffer Data Register (QuadSPI_RBDR20)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0254	RX Buffer Data Register (QuadSPI_RBDR21)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0258	RX Buffer Data Register (QuadSPI_RBDR22)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_025C	RX Buffer Data Register (QuadSPI_RBDR23)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0260	RX Buffer Data Register (QuadSPI_RBDR24)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0264	RX Buffer Data Register (QuadSPI_RBDR25)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0268	RX Buffer Data Register (QuadSPI_RBDR26)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_026C	RX Buffer Data Register (QuadSPI_RBDR27)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0270	RX Buffer Data Register (QuadSPI_RBDR28)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0274	RX Buffer Data Register (QuadSPI_RBDR29)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0278	RX Buffer Data Register (QuadSPI_RBDR30)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_027C	RX Buffer Data Register (QuadSPI_RBDR31)	32	R/W	0000_0000h	10.2.13.27/ 2617
30BB_0300	LUT Key Register (QuadSPI_LUTKEY)	32	R/W	5AF0_5AF0h	10.2.13.28/ 2618
30BB_0304	LUT Lock Configuration Register (QuadSPI_LCKCR)	32	R/W	0000_0002h	10.2.13.29/ 2619
30BB_0310	Look-up Table register (QuadSPI_LUT0)	32	R/W	0818_0403h	10.2.13.30/ 2620
30BB_0314	Look-up Table register (QuadSPI_LUT1)	32	R/W	2400_1C08h	10.2.13.31/ 2621
30BB_0318	Look-up Table register (QuadSPI_LUT2)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_031C	Look-up Table register (QuadSPI_LUT3)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0320	Look-up Table register (QuadSPI_LUT4)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0324	Look-up Table register (QuadSPI_LUT5)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0328	Look-up Table register (QuadSPI_LUT6)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_032C	Look-up Table register (QuadSPI_LUT7)	32	R/W	0000_0000h	10.2.13.32/ 2622

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_0330	Look-up Table register (QuadSPI_LUT8)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0334	Look-up Table register (QuadSPI_LUT9)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0338	Look-up Table register (QuadSPI_LUT10)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_033C	Look-up Table register (QuadSPI_LUT11)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0340	Look-up Table register (QuadSPI_LUT12)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0344	Look-up Table register (QuadSPI_LUT13)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0348	Look-up Table register (QuadSPI_LUT14)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_034C	Look-up Table register (QuadSPI_LUT15)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0350	Look-up Table register (QuadSPI_LUT16)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0354	Look-up Table register (QuadSPI_LUT17)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0358	Look-up Table register (QuadSPI_LUT18)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_035C	Look-up Table register (QuadSPI_LUT19)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0360	Look-up Table register (QuadSPI_LUT20)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0364	Look-up Table register (QuadSPI_LUT21)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0368	Look-up Table register (QuadSPI_LUT22)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_036C	Look-up Table register (QuadSPI_LUT23)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0370	Look-up Table register (QuadSPI_LUT24)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0374	Look-up Table register (QuadSPI_LUT25)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0378	Look-up Table register (QuadSPI_LUT26)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_037C	Look-up Table register (QuadSPI_LUT27)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0380	Look-up Table register (QuadSPI_LUT28)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0384	Look-up Table register (QuadSPI_LUT29)	32	R/W	0000_0000h	10.2.13.32/ 2622

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_0388	Look-up Table register (QuadSPI_LUT30)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_038C	Look-up Table register (QuadSPI_LUT31)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0390	Look-up Table register (QuadSPI_LUT32)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0394	Look-up Table register (QuadSPI_LUT33)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0398	Look-up Table register (QuadSPI_LUT34)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_039C	Look-up Table register (QuadSPI_LUT35)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03A0	Look-up Table register (QuadSPI_LUT36)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03A4	Look-up Table register (QuadSPI_LUT37)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03A8	Look-up Table register (QuadSPI_LUT38)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03AC	Look-up Table register (QuadSPI_LUT39)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03B0	Look-up Table register (QuadSPI_LUT40)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03B4	Look-up Table register (QuadSPI_LUT41)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03B8	Look-up Table register (QuadSPI_LUT42)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03BC	Look-up Table register (QuadSPI_LUT43)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03C0	Look-up Table register (QuadSPI_LUT44)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03C4	Look-up Table register (QuadSPI_LUT45)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03C8	Look-up Table register (QuadSPI_LUT46)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03CC	Look-up Table register (QuadSPI_LUT47)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03D0	Look-up Table register (QuadSPI_LUT48)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03D4	Look-up Table register (QuadSPI_LUT49)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03D8	Look-up Table register (QuadSPI_LUT50)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03DC	Look-up Table register (QuadSPI_LUT51)	32	R/W	0000_0000h	10.2.13.32/ 2622

Table continues on the next page...

QuadSPI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BB_03E0	Look-up Table register (QuadSPI_LUT52)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03E4	Look-up Table register (QuadSPI_LUT53)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03E8	Look-up Table register (QuadSPI_LUT54)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03EC	Look-up Table register (QuadSPI_LUT55)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03F0	Look-up Table register (QuadSPI_LUT56)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03F4	Look-up Table register (QuadSPI_LUT57)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03F8	Look-up Table register (QuadSPI_LUT58)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_03FC	Look-up Table register (QuadSPI_LUT59)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0400	Look-up Table register (QuadSPI_LUT60)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0404	Look-up Table register (QuadSPI_LUT61)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_0408	Look-up Table register (QuadSPI_LUT62)	32	R/W	0000_0000h	10.2.13.32/ 2622
30BB_040C	Look-up Table register (QuadSPI_LUT63)	32	R/W	0000_0000h	10.2.13.32/ 2622

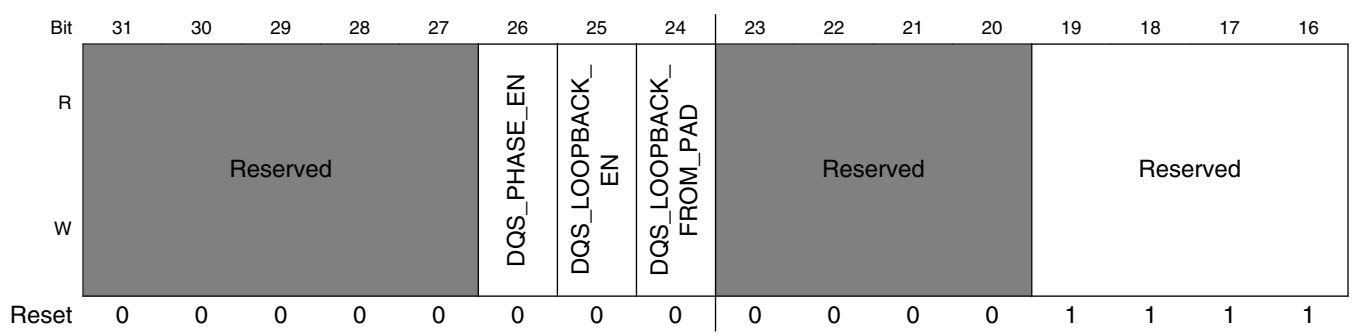
10.2.13.1 Module Configuration Register (QuadSPI_MCR)

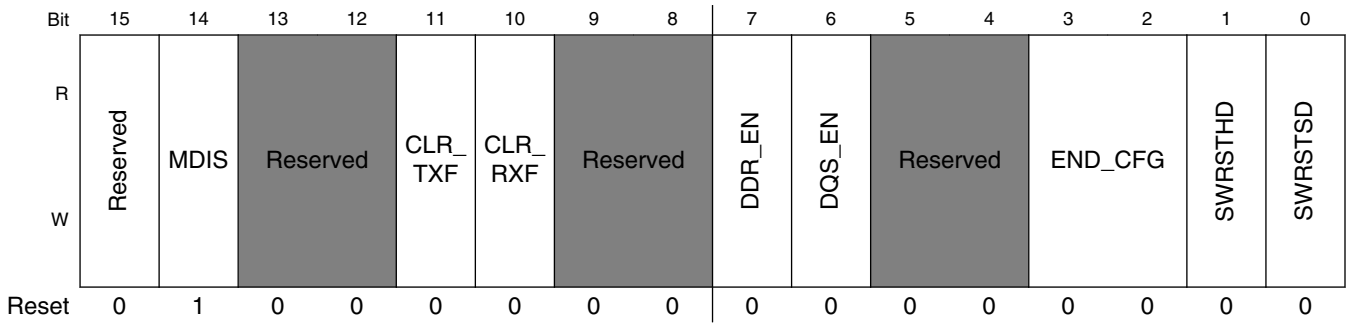
The QuadSPI_MCR holds configuration data associated with QuadSPI operation.

Write:

- *All other fields: Anytime*

Address: 30BB_0000h base + 0h offset = 30BB_0000h





QuadSPI_MCR field descriptions

Field	Description
31–27 Reserved	This field is reserved.
26 DQS_PHASE_ EN	This bit controls internal DQS output phase. If DQS_EN and DQS_LOOPBACK_EN are both set to 1, this bit should be set in SDR mode, and cleared in DDR mode. If either DQS_EN or DQS_LOOPBACK_EN is set to 0, this bit is ignored.
25 DQS_ LOOPBACK_EN	Quadspi will output serial data strobe signal which will be loopback from pad to sample input flash serial data. Please note pad should be force input for loopback.Quadspi will output serial data strobe signal which will be loopback from pad to sample input flash serial data. Please note pad should be force input for loopback. This bit is a don't care when DQS_EN is set to 0. 1'b1 DQS loopback sampling enabled 1'b0 DQS loopback sampling disabled
24 DQS_ LOOPBACK_ FROM_PAD	This bit should always be set to '1' when DQS_LOOPBACK_EN is set to '1'. When DQS_LOOPBACK_EN is set to '1', this bit is ignored.
23–20 Reserved	This field is reserved.
19–16 Reserved	This field is reserved. This field is reserved and should always be set to 0xF.
15 Reserved	This field is reserved. This field is reserved.
14 MDIS	Module Disable. The MDIS bit allows the clock to the non-memory mapped logic in the QuadSPI to be stopped, putting the QuadSPI in a software controlled power-saving state. Please refer to , for more information. 0 Enable QuadSPI clocks. 1 Allow external logic to disable QuadSPI clocks.
13–12 Reserved	This field is reserved.
11 CLR_TXF	Clear TX FIFO/Buffer. Invalidate the TX Buffer content. 0 No action. 1 Read and write pointers of the TX Buffer are reset to 0. QSPI_TBSR[TRCTR] is reset to 0.

Table continues on the next page...

QuadSPI_MCR field descriptions (continued)

Field	Description
10 CLR_RXF	Clear RX FIFO. Invalidate the RX Buffer. 0 No action. 1 Read and write pointers of the RX Buffer are reset to 0. QSPI_RBSR[RDBFL] is reset to 0.
9–8 Reserved	This field is reserved.
7 DDR_EN	DDR mode enable: 0 2x and 4x clocks are disabled for SDR instructions only 1 2x and 4x clocks are enabled supports both SDR and DDR instruction.
6 DQS_EN	DQS enable: This field is valid for both SDR and DDR mode. For more details Refer Data Strobe Signal Functionality 0 DQS disabled. 1 DQS enabled- When enabled, the incoming data is sampled on both the edges of DQS input when QSPI_MCR[DDR_EN] is set, else, on only one edge when QSPI_MCR[DDR_EN] is 0. The QSPI_SMPR[DDR_SMP] values are ignored.
5–4 Reserved	This field is reserved.
3–2 END_CFG	Defines the endianness of the QSPI module.For more details refer to Byte Ordering Endianess
1 SWRSTHD	Software reset for AHB domain **NOTE:** Please keep other fields value when write to SWRSTHD and SWRSTSD **NOTE:** These software reset don't reset register setting but only reset internal flip-flops in quadspi controller **NOTE:** To remove the reset, need to write 0 to SWRSTHD and SWRSTSD 0 No action 1 AHB domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects.
0 SWRSTSD	Software reset for Serial Flash domain **NOTE:** Please keep other fields value when write to SWRSTHD and SWRSTSD **NOTE:** These software reset don't reset register setting but only reset internal flip-flops in quadspi controller **NOTE:** To remove the reset, need to write 0 to SWRSTHD and SWRSTSD 0 No action 1 Serial Flash domain flops are reset. Does not reset configuration registers. It is advisable to reset both the serial flash domain and AHB domain at the same time. Resetting only one domain might lead to side effects.

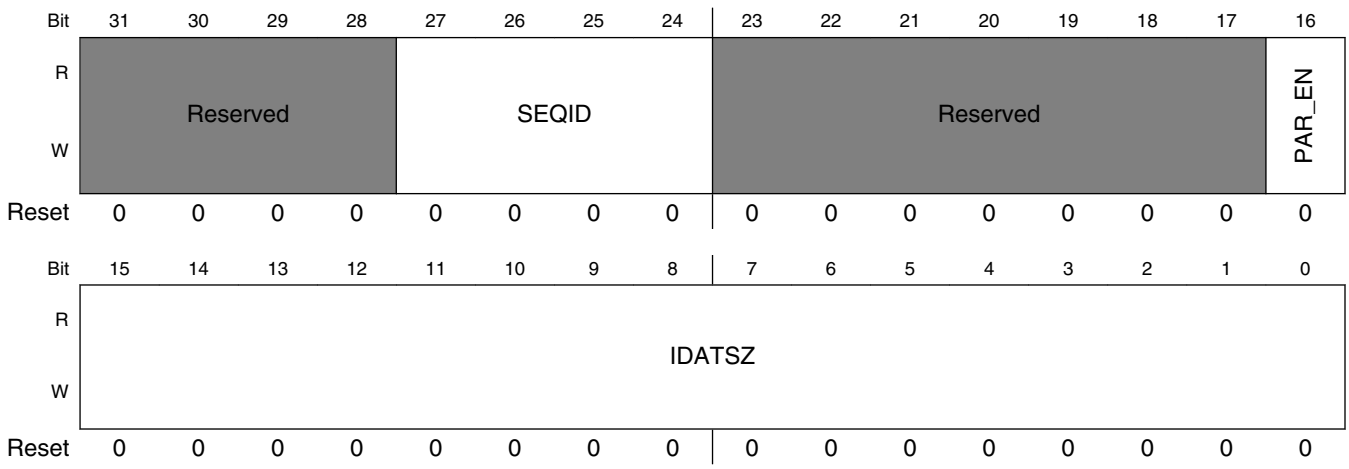
10.2.13.2 IP Configuration Register (QuadSPI_IPCR)

The IP configuration register provides all the configuration required for an IP initiated command. An IP command can be triggered by writing in the SEQID field of this register. If the SEQID field is written successfully, a new command to the external serial flash is started as per the sequence pointed to by the SEQID field. Refer to Normal Mode, for details about the command triggering and command execution.

Write:

- *QSPI_SR[IP_ACC]=0*

Address: 30BB_0000h base + 8h offset = 30BB_0008h



QuadSPI_IPCR field descriptions

Field	Description
31–28 Reserved	This field is reserved.
27–24 SEQID	Points to a sequence in the Look-up-table. The SEQID defines the bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table for more details. A write to this bit -field triggers a transaction on the serial flash interface.
23–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.
IDATSZ	IP data transfer size: Defines the data transfer size in bytes of the IP command.

10.2.13.3 Flash Configuration Register (QuadSPI_FLSHCR)

The Flash configuration register contains the flash device specific timings that must be met by the QuadSPI controller for the device to function correctly.

Write:

- *QSPI_SR[AHB_ACC] = 0*
- *QSPI_SR[IP_ACC] = 0*

Address: 30BB_0000h base + Ch offset = 30BB_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														TDH		Reserved				TCSH				Reserved				TCSS			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	1	1

QuadSPI_FLSHCR field descriptions

Field	Description
31–18 Reserved	This field is reserved.
17–16 TDH	Serial flash Data In hold time: This helps in meeting the Data In Hold time requirement of a Flash. This is valid only in DDR mode. Refer to Data input hold requirement of Flash for details. **NOTE:** This field should be set to 01 in DDR mode (DDR_EN=1) and set to 00 in SDR mode (DDR_EN=0). Other value are reserved. 00 Data aligned with the posedge of Internal reference clock of QuadSPI 01 Data aligned with 2x serial flash half clock 10 Reserved 11 Reserved
15–12 Reserved	This field is reserved.
11–8 TCSH	Serial flash CS hold time in terms of serial flash clock cycles. **NOTE:** The actual delay between chip select assertion and clock fall edge is defined as: • 1 SCK cycle if TCSH is set 0 or 1 • N SCK cycle if TCSH is set N (N>1)
7–4 Reserved	This field is reserved. Reserved.
TCSS	Serial flash CS setup time in terms of serial flash clock cycles. **NOTE:** 1. The actual delay between chip select assertion and clock fall edge is defined as: • 0.5 SCK cycle if TCSS is set 0 or 1 • N+0.5 SCK cycle if TCSS is set N (N>1) 2. Any update to TCSS register bits is visible on the flash interface only from the second transaction following the update.

10.2.13.4 Buffer0 Configuration Register (QuadSPI_BUF0CR)

This register provides the configuration for any access to buffer0. An access is routed to buffer0 when the master port number of the incoming AHB request matches the MSTRID field of the BUF0CR. Any buffer "miss" leads to a serial flash transaction being triggered as per the sequence pointed to the SEQID field. Buffer0 may also be configured as a high priority buffer by setting the HP_EN field of this register.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 10h offset = 30BB_0010h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	HP_EN	Reserved														
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF0CR field descriptions

Field	Description
31 HP_EN	High Priority Enable: When set, the master associated with this buffer is assigned a priority higher than the rest of the masters. An access by a high priority master will suspend any ongoing prefetch by another AHB master and will be serviced on high priority. Refer to Flexible AHB Buffers for details.
30–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash. For example,a value of 0x2 will set transfer size to 16bytes.When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved. Reserved.
MSTRID	Master ID: The ID of the AHB master associated with BUFFER0. Any AHB access with this master port number is routed to this buffer.

10.2.13.5 Buffer1 Configuration Register (QuadSPI_BUF1CR)

This register provides the configuration for any access to buffer1. An access is routed to buffer1 when the master port number of the incoming AHB request matches the MSTRID field of the BUF1CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 14h offset = 30BB_0014h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF1CR field descriptions

Field	Description
31–16 Reserved	This field is reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 bytes of an AHB triggered access to serial flash.For example, a value of 0x2 will set transfer size to 16bytes. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
MSTRID	Master ID: The ID of the AHB master associated with BUFFER1. Any AHB access with this master port number is routed to this buffer.

10.2.13.6 Buffer2 Configuration Register (QuadSPI_BUF2CR)

This register provides the configuration for any access to buffer2. An access is routed to buffer2 when the master port number of the incoming AHB request matches the MSTRID field of the BUF2CR. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 18h offset = 30BB_0018h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																ADATSZ								Reserved				MSTRID			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF2CR field descriptions

Field	Description
31–16 Reserved	This field is reserved. Reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 Bytes of an AHB triggered access to serial flash. For example, a value of 0x2 will set transfer size to 16bytes.When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved. Reserved.
MSTRID	Master ID: The ID of the AHB master associated with BUFFER2. Any AHB access with this master port number is routed to this buffer.

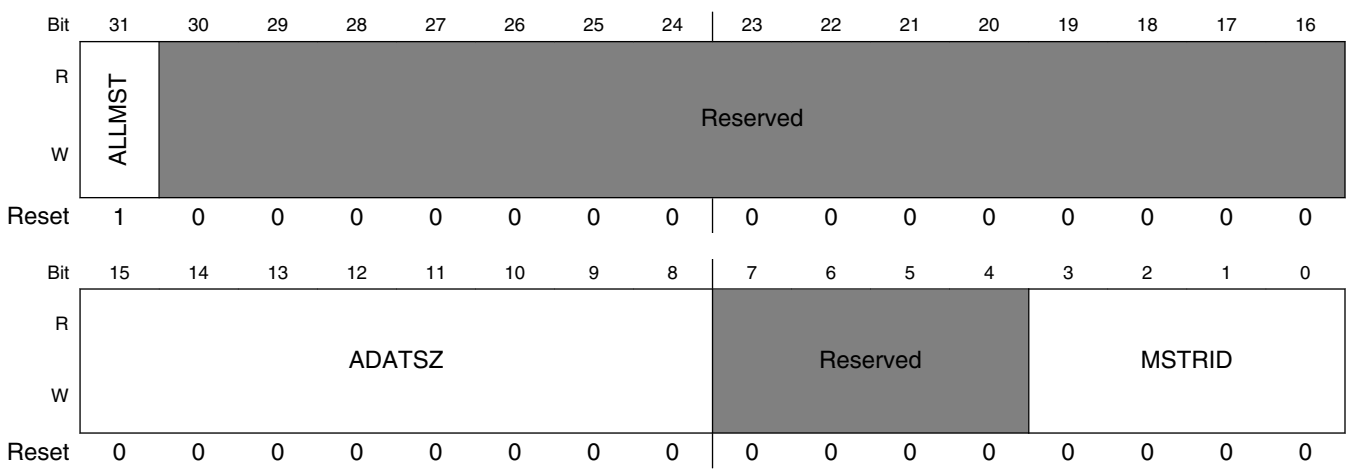
10.2.13.7 Buffer3 Configuration Register (QuadSPI_BUF3CR)

This register provides the configuration for any access to buffer3. An access is routed to buffer3 when the master port number of the incoming AHB request matches the MSTRID field of the BUF3CR. Any buffer "miss" leads to the buffer being flushed a serial flash transaction being triggered as per the sequence pointed to by the SEQID field. If the ALLMST field is set, any transaction where the master port number does not match any of the buffer MSTRID fields will be routed to this buffer. In the case that the ALLMST field is not set, any such transaction (where master port number does not match any of the MSTRID fields) will be returned an ERROR response.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 1Ch offset = 30BB_001Ch



QuadSPI_BUF3CR field descriptions

Field	Description
31 ALLMST	All master enable: When set, buffer3 acts as an all-master buffer. Any AHB access with a master port number not matching with the master ID of buffer0 or buffer1 or buffer2 is routed to buffer3. When set, the MSTRID field of this register is ignored.
30–16 Reserved	This field is reserved. Reserved.
15–8 ADATSZ	AHB data transfer size: Defines the data transfer size in 8 Bytes of an AHB triggered access to serial flash. When ADATSZ = 0, the data size mentioned the sequence pointed to by the SEQID field overrides this value. SW should ensure that this transfer size is not greater than the size of this buffer.
7–4 Reserved	This field is reserved.
MSTRID	Master ID: The ID of the AHB master associated with BUFFER3. Any AHB access with this master port number is routed to this buffer.

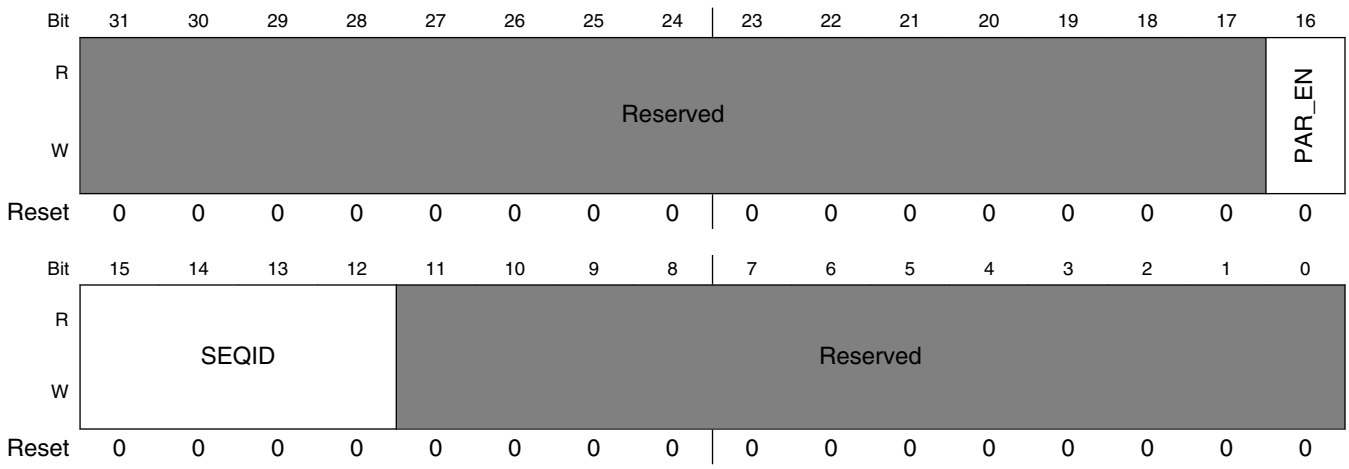
10.2.13.8 Buffer Generic Configuration Register (QuadSPI_BFGENCR)

This register provides the generic configuration to any of the buffer accesses. Any buffer "miss" leads to the buffer being flushed and a serial flash transaction being triggered as per the sequence pointed to by the SEQID field. If the PAR_EN field is set, all the buffer accesses result in parallel accesses to the flashes.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 20h offset = 30BB_0020h



QuadSPI_BFGENCR field descriptions

Field	Description
31–17 Reserved	This field is reserved.
16 PAR_EN	When set, a transaction to two serial flash devices is triggered in parallel mode. Refer to Parallel Flash Mode for more details.
15–12 SEQID	Points to a sequence in the Look-up-table. The SEQID defines the bits [6:2] of the LUT index. The bits [1:0] are always assumed to be 0. Refer to Look-up Table. **NOTE:** If the sequence pointer differs between the new and previous sequence then the user should reset this. See QSPI_SPTRCLR for more information.
Reserved	This field is reserved.

10.2.13.9 Buffer0 Top Index Register (QuadSPI_BUF0IND)

This register specifies the top index of buffer0, which defines its size. Note that that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned, as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less 8. For example, setting BUF0IND to 0 gives 8 bytes, 1 give 16bytes etc.

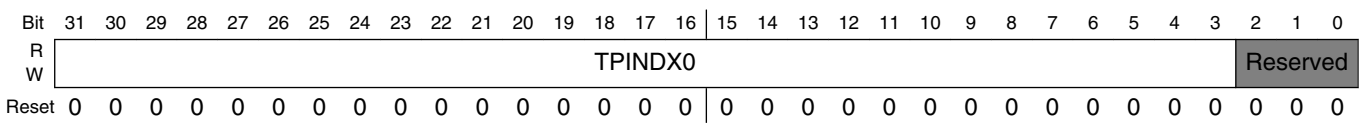
The size of buffer0 is the difference between the BUF0IND+8 and 0.

It is the responsibility of the software to ensure that BUF0IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 30h offset = 30BB_0030h



QuadSPI_BUF0IND field descriptions

Field	Description
31–3 TPINDX0	Top index of buffer 0.
Reserved	This field is reserved. Reserved.

10.2.13.10 Buffer1 Top Index Register (QuadSPI_BUF1IND)

This register specifies the top index of buffer1, which defines its size. Note that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less. The size of buffer1 is the difference between the BUF1IND and BUF0IND.

It is the responsibility of the software to ensure that BUF1IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 34h offset = 30BB_0034h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX1																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF1IND field descriptions

Field	Description
31–3 TPINDX1	Top index of buffer 1.
Reserved	This field is reserved.

10.2.13.11 Buffer2 Top Index Register (QuadSPI_BUF2IND)

This register specifies the top index of buffer2, which defines its size. Note that that the 3 LSBs of this register are set to zero - this ensures that the buffer is 64bit aligned as each buffer entry is 64bits long.

The register value should be set to the desired number of bytes less 8. The size of buffer2 is the difference between the BUF2IND and BUF1IND.

It is the responsibility of the software to ensure that BUF2IND value is not greater than the overall size of the buffer. The hardware does not provide any protection against illegal programming.

Write:

- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 38h offset = 30BB_0038h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPINDX2																													Reserved		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_BUF2IND field descriptions

Field	Description
31–3 TPINDX2	Top index of buffer 2.
Reserved	This field is reserved.

10.2.13.12 Serial Flash Address Register (QuadSPI_SFAR)

The module automatically translates this address on the memory map to the address on the flash itself. When operating in 24bit mode, only bits 23-0 are sent to the flash, in 32bit mode, bits 27-0 are used with bits 31-28 driven to 0 Refer to Table 10-5 for the mapping between the access mode and the QSPI_SFAR content and to Normal Mode for details about the command triggering and command execution. The software should ensure that the serial flash address provided in the QSPI_SFAR register lies in the valid flash address range as defined in Table 10-5.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: 30BB_0000h base + 100h offset = 30BB_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SFADR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SFAR field descriptions

Field	Description
SFADR	Serial Flash Address. The register content is used as byte address for all following IP Commands.

10.2.13.13 Sampling Register (QuadSPI_SMPR)

The Sampling Register allows configuration of how the incoming data from the external serial flash devices are sampled in the QuadSPI module.

Write:Disabled Mode

Address: 30BB_0000h base + 108h offset = 30BB_0108h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0													DDRSMP		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									SDRSMP		0		0		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

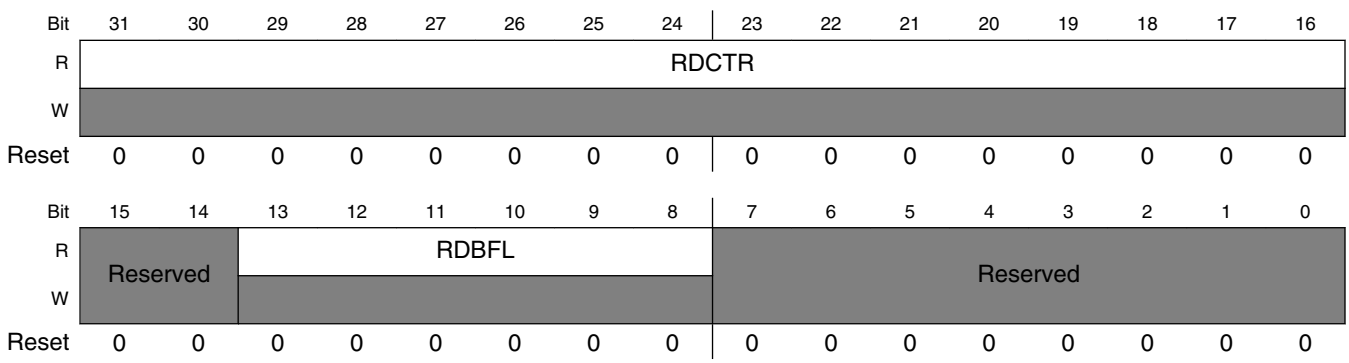
QuadSPI_SMPR field descriptions

Field	Description
31–19 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
18–16 DDRSMP	DDR Sampling point. Select the sampling point for incoming data when serial flash is executing a DDR instruction. Refer to Input timing in DDR mode with internal sampling , for details on the sampling points.
15–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–5 SDRSMP	SDR sampling point.
4–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

10.2.13.14 RX Buffer Status Register (QuadSPI_RBSR)

This register contains information related to the receive data buffer.

Address: 30BB_0000h base + 10Ch offset = 30BB_010Ch



QuadSPI_RBSR field descriptions

Field	Description
31–16 RDCTR	Read Counter, indicates how many entries of 4 bytes have been removed from the RX Buffer.For example a value of 0x2 would indicate 8bytes have been removed It is incremented by the number (QSPI_RBCT[WMRK] + 1) on RX Buffer POP event. The RX Buffer can be popped using DMA or pop flag QSPI_FR[RBDF]. The QSPI_RSER[RBDDE] defines which pop has to be done. For further details please refer to AHB RX Data Buffer (QSPI_ARDB0 to QSPI_ARDB31) and "Data Transfer from the QuadSPI Module Internal Buffers section in Flash Read section.
15–14 Reserved	This field is reserved.
13–8 RDBFL	RX Buffer Fill Level, indicates how many entries of 4 bytes are still available in the RX Buffer.For example a value of 0x2 would indicate 8bytes are available.
Reserved	This field is reserved.

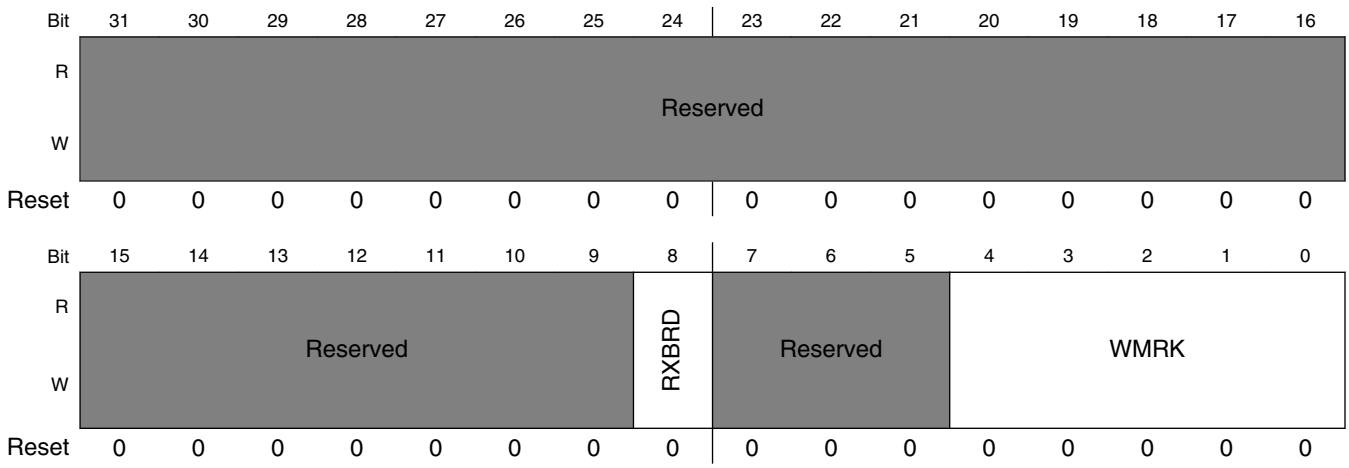
10.2.13.15 RX Buffer Control Register (QuadSPI_RBCT)

This register contains control data related to the receive data buffer.

Write:

- *QSPI_SR[IP_ACC] = 0*

Address: 30BB_0000h base + 110h offset = 30BB_0110h



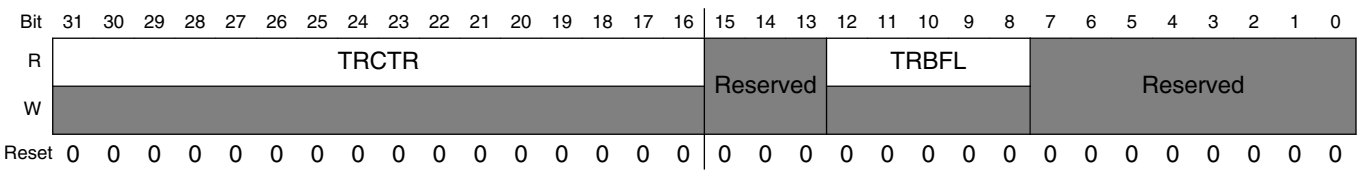
QuadSPI_RBCT field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 RXBRD	RX Buffer Readout: This bit specifies the access scheme for the RX Buffer readout. 0 RX Buffer content is read using the AHB Bus registers QSPI_ARDB0 to QSPI_ARDB31. For details, refer to Exclusive Access to Serial Flash for AHB Commands. 1 RX Buffer content is read using the IP Bus registers QSPI_RBDR0 to QSPI_RBDR31.
7–5 Reserved	This field is reserved.
WMRK	RX Buffer Watermark: This field determines when the readout action of the RX Buffer is triggered. When the number of valid entries in the RX Buffer is equal to or greater than the number given by (WMRK+1) the QSPI_SR[RXWE] flag is asserted.The value should be entered as the number of 4byte entries minus 1. For example a value of 0x0 would set the watermark to 4bytes, 1 to 8bytes, 2 to 12bytes etc. For details, refer to DMA Usage.

10.2.13.16 TX Buffer Status Register (QuadSPI_TBSR)

This register contains information related to the transmit data buffer.

Address: 30BB_0000h base + 150h offset = 30BB_0150h



QuadSPI_TBSR field descriptions

Field	Description
31–16 TRCTR	Transmit Counter. This field indicates how many entries of 4 bytes have been written into the TX Buffer by host accesses. It is reset to 0 when a 1 is written into the QSPI_MCR[CLR_TXF] bit. It is incremented on each write access to the QSPI_TBDR register when another word has been pushed onto the TX Buffer. When it is not cleared the TRCTR field wraps around to 0. Refer to TX Buffer Data Register (QuadSPI_TBDR) for details.
15–13 Reserved	This field is reserved.
12–8 TRBFL	TX Buffer Fill Level. The TRBFL field contains the number of entries of 4 bytes each available in the TX Buffer for the QuadSPI module to transmit to the serial flash device.
Reserved	This field is reserved.

10.2.13.17 TX Buffer Data Register (QuadSPI_TBDR)

The QSPI_TBDR register provides access to the circular TX Buffer of depth 128 bytes. This buffer provides the data written into it as write data for the page programming commands to the serial flash device. Refer to Table 10-14 for the byte ordering scheme. A write transaction on the flash with data size of less than 32 bits will lead to the removal of four data entry from the TX buffer. The valid bits will be used and the rest of the bits will be discarded.

Write:

- *QSPI_SR[TXFULL] = 0*

32-bit write access required

NOTE

There is no limitation on Flash programming size. But the data size written to TX Buffer should be 64 Byte aligned. If flash programming size is not 64 bytes aligned, please write redundant data to TX Buffers. The redundant data will be ignored by QuadSPI controller.

Address: 30BB_0000h base + 154h offset = 30BB_0154h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXDATA																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

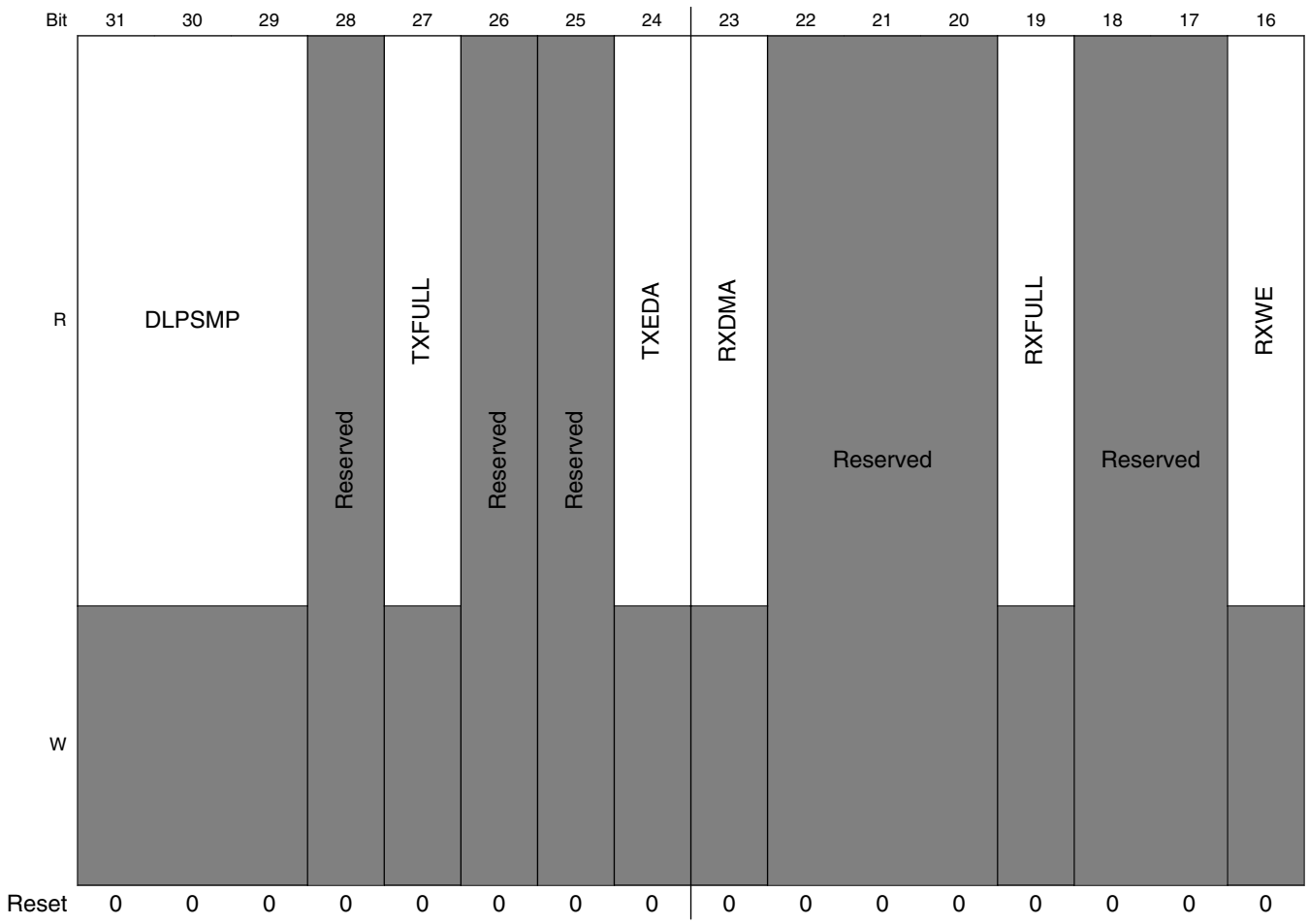
QuadSPI_TBDR field descriptions

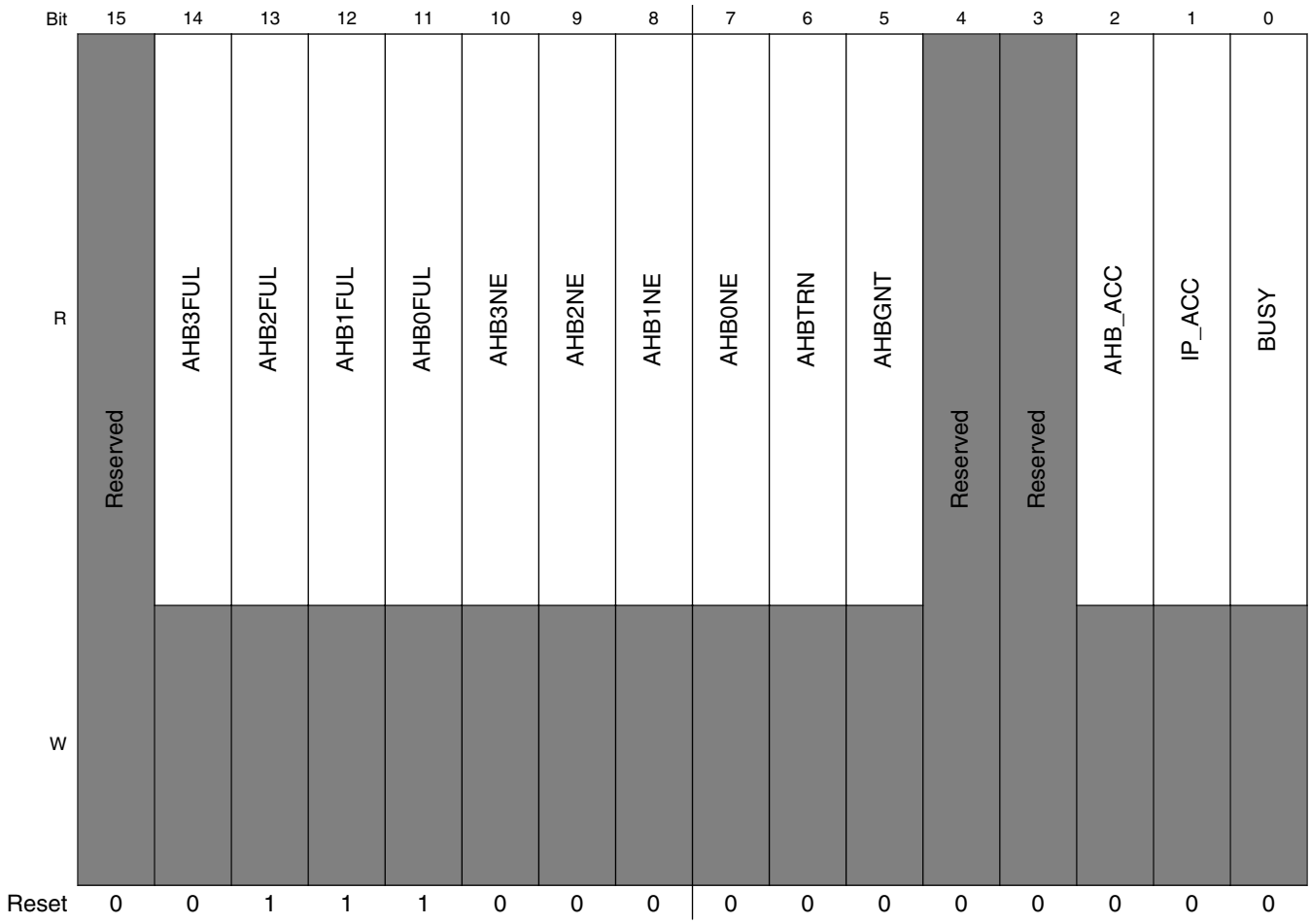
Field	Description
TXDATA	TX Data On write access the data is written into the next available entry of the TX Buffer and the QPSI_TBSR[TRBFL] field is updated accordingly. On a read access, the last data written to the register is returned.

10.2.13.18 Status Register (QuadSPI_SR)

The QSPI_SR register provides all available status information about SFM command execution and arbitration, the RX Buffer and TX Buffer and the AHB Buffer.

Address: 30BB_0000h base + 15Ch offset = 30BB_015Ch





QuadSPI_SR field descriptions

Field	Description
31–29 DLPSMP	**NOTE:** Data learning is not implemented on this chip. Data learning pattern sampling point: The sampling point found by the controller with the data learning pattern. • This is used for DDR only. • If the learning fails, this field will return garbage and DLPFF bit will be set.
28 Reserved	This field is reserved.
27 TXFULL	TX Buffer Full: Asserted when no more data can be stored.
26 Reserved	This field is reserved.
25 Reserved	This field is reserved.
24 TXEDA	Tx Buffer Enough Data Available Asserted when TX Buffer contains enough data for any pop operation to take place. There must be atleast 128bit data available in TX FIFO for any pop operation otherwise QSPI_FR[TBUF] will be set.

Table continues on the next page...

QuadSPI_SR field descriptions (continued)

Field	Description
23 RXDMA	RX Buffer DMA: Asserted when RX Buffer read out via DMA is active i.e DMA is requested or running.
22–20 Reserved	This field is reserved.
19 RXFULL	RX Buffer Full: Asserted when the RX Buffer is full, i.e. that QSPI_RBSR[RDBFL] field is equal to 32.
18–17 Reserved	This field is reserved.
16 RXWE	RX Buffer Watermark Exceeded: Asserted when the number of valid entries in the RX Buffer exceeds the number given in the QSPI_RBCT[WMRK] field.
15 Reserved	This field is reserved.
14 AHB3FUL	AHB 3 Buffer Full: Asserted when AHB 3 buffer is full.
13 AHB2FUL	AHB 2 Buffer Full: Asserted when AHB 2 buffer is full.
12 AHB1FUL	AHB 1 Buffer Full: Asserted when AHB 1 buffer is full.
11 AHB0FUL	AHB 0 Buffer Full: Asserted when AHB 0 buffer is full.
10 AHB3NE	AHB 3 Buffer Not Empty: Asserted when AHB 3 buffer contains data.
9 AHB2NE	AHB 2 Buffer Not Empty: Asserted when AHB 2 buffer contains data.
8 AHB1NE	AHB 1 Buffer Not Empty: Asserted when AHB 1 buffer contains data.
7 AHB0NE	AHB 0 Buffer Not Empty: Asserted when AHB 0 buffer contains data.
6 AHBTRN	AHB Access Transaction pending: Asserted when there is a pending request on the AHB interface. Refer to the AMBA specification for details.
5 AHBGNT	AHB Command priority Granted: Asserted when another module has been granted priority of AHB Commands against IP Commands. For details refer to Command Arbitration.
4 Reserved	This field is reserved.
3 RESERVED	This field is reserved.
2 AHB_ACC	AHB Access: Asserted when the transaction currently executed was initiated by AHB bus.
1 IP_ACC	IP Access: Asserted when transaction currently executed was initiated by IP bus.
0 BUSY	Module Busy: Asserted when module is currently busy handling a transaction to an external flash device.

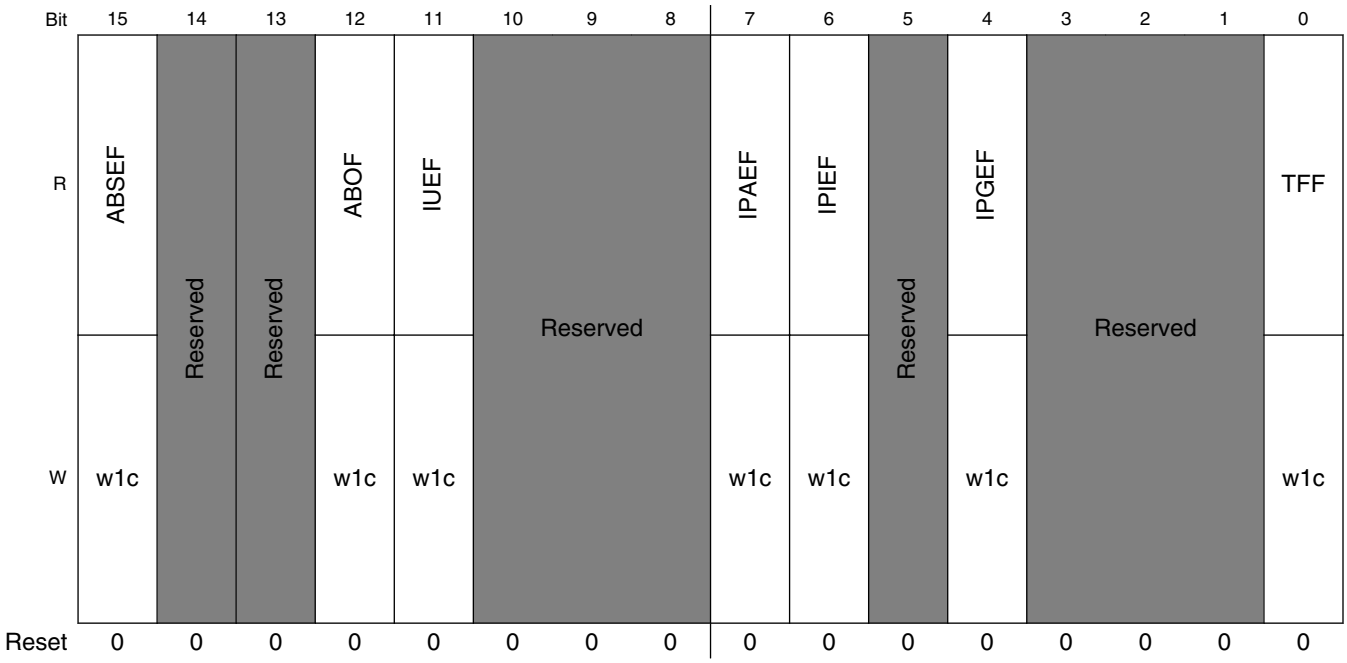
10.2.13.19 Flag Register (QuadSPI_FR)

The QSPI_FR register provides all available flags about SFM command execution and arbitration which may serve as source for the generation of interrupt service requests. Note that the error flags in this register do not relate directly to the execution of the transaction in the serial flash device itself but only to the behavior and conditions visible in the QuadSPI module.

Write:Enabled Mode

Address: 30BB_0000h base + 160h offset = 30BB_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DLPFF	Reserved	Reserved		TBFF	TBUF	Reserved		ILLINE	Reserved					RBOF	RBDF
W	w1c				w1c	w1c			w1c						w1c	w1c
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0



QuadSPI_FR field descriptions

Field	Description
31 DLPFF	**NOTE:** Data learning is not implemented on this chip. Data Learning Pattern Failure Flag: Set when DATA_LEARN instruction was encountered in a sequence but no sampling point was found for the data learning pattern The controller automatically starts sampling using the value in QSPI_SMPR[DDRSMP].
30 Reserved	This field is reserved.
29–28 Reserved	This field is reserved.
27 TBFF	TX Buffer Fill Flag: Before writing to the TX buffer, this bit should be cleared. Then this bit has to be read back. If the bit is set, the TX Buffer can take more data. If the bit remains cleared, the TX buffer is full. Refer to Tx Buffer Operation for details.
26 TBUF	TX Buffer Underrun Flag: Set when the module tried to pull data although TX Buffer was emptyor the buffer contains less than 128bits of data. The application must ensure that the buffer never goes empty during a transaction expect for the last data fetch. The IP Command leading to the TX Buffer underrun is continued (data sent to the serial flash device is all F in case of valid tx underrun. The application must clear the TX Buffer in response to this event by writing a 1 into the QSPI_MCR[CLR_TXF] bit.
25–24 Reserved	This field is reserved.
23 ILLINE	Illegal Instruction Error Flag: Set when an illegal instruction is encountered by the controller in any of the sequences. Refer to Table 10-12 for a list of legal instructions.
22–18 Reserved	This field is reserved.
17 RBOF	RX Buffer Overflow Flag: Set when not all the data read from the serial flash device could be pushed into the RX Buffer. The IP Command leading to this condition is continued until the number of bytes according to the QSPI_IPCR[IDATSZ] field has been read from the serial flash device.

Table continues on the next page...

QuadSPI_FR field descriptions (continued)

Field	Description
	The content of the RX Buffer is not changed.
16 RBDF	RX Buffer Drain Flag: Will be set if the QuadSPI_SR[RXWE] status bit is asserted. Writing 1 into this bit triggers one of the following actions: • If the RX Buffer has up to QuadSPI_RBCT[WMRK] valid entries then the flag is cleared. • If the RX Buffer has more than QuadSPI_RBCT[WMRK] valid entries and the QuadSPI_RSER[RBDDE] bit is not set (flag driven mode) a RX Buffer POP event is triggered. The flag remains set if the RX Buffer contains more than QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. The flag is cleared if the RX Buffer contains less than or equal to QuadSPI_RBCT[WMRK] valid entries after the RX Buffer POP event is finished. Refer to "Receive Buffer Drain Interrupt or DMA Request" section in Normal Mode Interrupt and DMA Requests, for details.
15 ABSEF	AHB Sequence Error Flag: Set when the execution of an AHB Command is started with an WRITE or WRITE_DDR Command in the sequence pointed to by the QSPI_BUFxCR register Communication with the serial flash device is terminated before the execution of WRITE/WRITE_DDR command by the QuadSPI module. The AHB bus request which triggered this command is answered with an ERROR response.
14 Reserved	Reserved This field is reserved.
13 Reserved	Reserved This field is reserved.
12 ABOF	AHB Buffer Overflow Flag: Set when the size of the AHB access exceeds the size of the AHB buffer. This condition can occur only if the QSPI_BUFxCR[ADATSZ] field is programmed incorrectly. The AHB Command leading to this condition is continued until the number of entries according to the QSPI_BUFxCR[ADATSZ] field has been read from the serial flash device. The content of the AHB Buffer is not changed.
11 IUEF	IP Command Usage Error Flag: Set when in parallel flash mode the execution of an IP Command is started and the sequence pointed to by the sequence ID contains a WRITE or a WRITE_DDR command. Refer to Table 10-12 table for the related commands. Communication with the serial flash device is terminated before the execution of WRITE/WRITE_DDR command by the QuadSPI module.
10–8 Reserved	This field is reserved.
7 IPAEF	IP Command Trigger during AHB Access Error Flag. Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHB_ACC] bit is set. Any command leading to the assertion of the IPAEF flag is ignored.
6 IPIEF	IP Command Trigger could not be executed Error Flag. Set when the QSPI_SR[IP_ACC] bit is set (i.e. an IP triggered command is currently executing) and any of the following conditions occurs: • Write access to the QSPI_IPCR register. Any command leading to the assertion of the IPIEF flag is ignored • Write access to the QSPI_SFAR register. • Write access to the QSPI_RBCT register.
5 Reserved	This field is reserved.

Table continues on the next page...

QuadSPI_FR field descriptions (continued)

Field	Description
4 IPGEF	IP Command Trigger during AHB Grant Error Flag: Set when the following condition occurs: • A write access occurs to the QSPI_IPCR[SEQID] field and the QSPI_SR[AHBGNT] bit is set. Any command leading to the assertion of the IPGEF flag is ignored.
3–1 Reserved	This field is reserved.
0 TFF	IP Command Transaction Finished Flag: Set when the QuadSPI module has finished a running IP Command. If an error occurred the related error flags are valid, at the latest, in the same clock cycle when the TFF flag is asserted.

1. QSPI_BUFxCR implies anyone of QSPI_BUF0CR/QSPI_BUF1CR/QSPI_BUF2CR/QSPI_BUF3CR

10.2.13.20 Interrupt and DMA Request Select and Enable Register (QuadSPI_RSER)

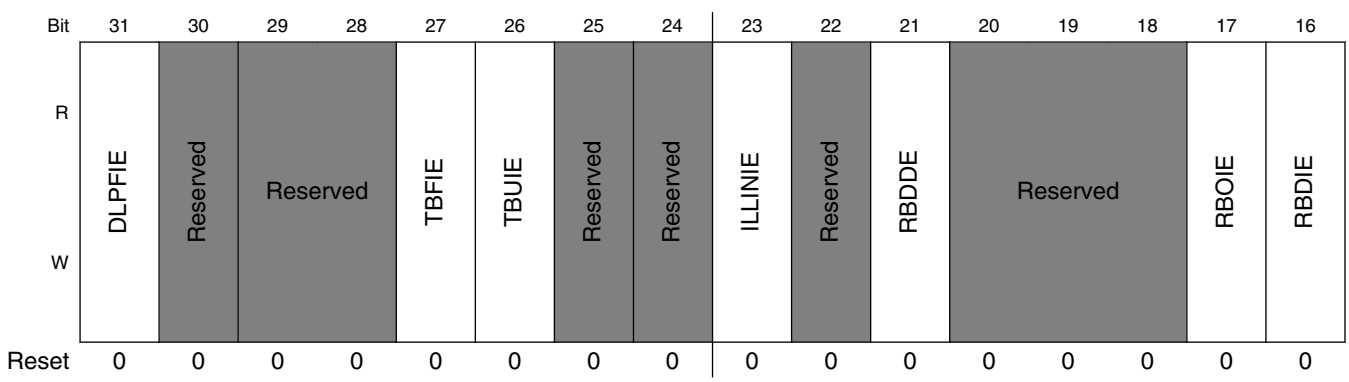
The QuadSPI_RSER register provides enables and selectors for the interrupts in the QuadSPI module.

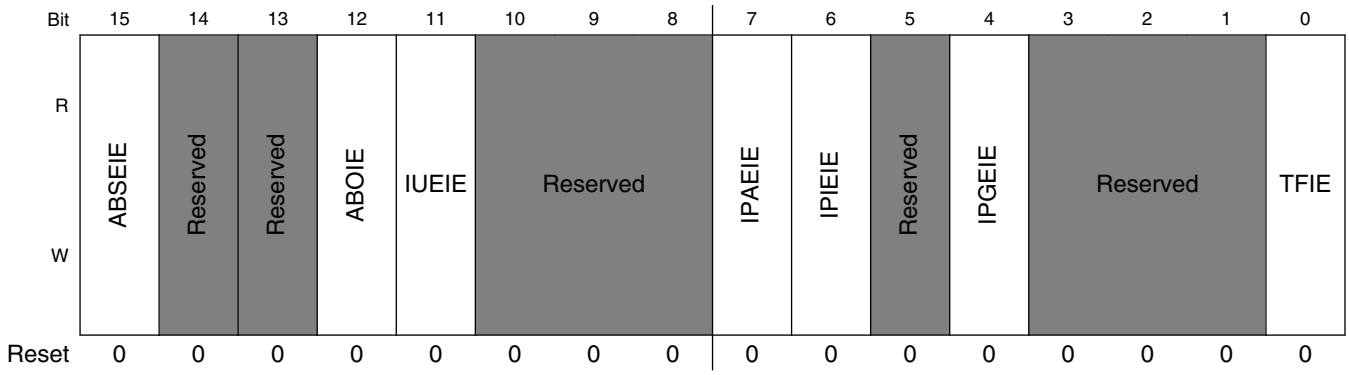
NOTE

Each flag of the QuadSPI_FR register enabled as source for an interrupt prevents the QuadSPI module from entering Stop Mode or Module Disable Mode when this flag is set.

Write:Anytime

Address: 30BB_0000h base + 164h offset = 30BB_0164h





QuadSPI_RSER field descriptions

Field	Description
31 DLPFIE	**NOTE:** Data learning is not implemented on this chip. Data Learning Pattern Failure Interrupt enable. Triggered by DLPFF flag in QSPI_FR register 0 No DLPFF interrupt will be generated 1 DLPFF interrupt will be generated
30 Reserved	This field is reserved.
29–28 RESERVED	This field is reserved.
27 TBFIE	TX Buffer Fill Interrupt Enable 0 No TBFF interrupt will be generated 1 TBFF interrupt will be generated
26 TBUIE	TX Buffer Underrun Interrupt Enable 0 No TBUF interrupt will be generated 1 TBUF interrupt will be generated
25 Reserved	This field is reserved.
24 Reserved	This field is reserved.
23 ILLINIE	Illegal Instruction Error Interrupt Enable. Triggered by ILLINE flag in QSPI_FR 0 No ILLINE interrupt will be generated 1 ILLINE interrupt will be generated
22 Reserved	This field is reserved.
21 RBDDE	RX Buffer Drain DMA Enable: Enables generation of DMA requests for RX Buffer Drain. When this bit is set DMA requests are generated as long as the QSPI_SR[RXWE] status bit is set. 0 No DMA request will be generated 1 DMA request will be generated
20–18 Reserved	This field is reserved.

Table continues on the next page...

QuadSPI_RSER field descriptions (continued)

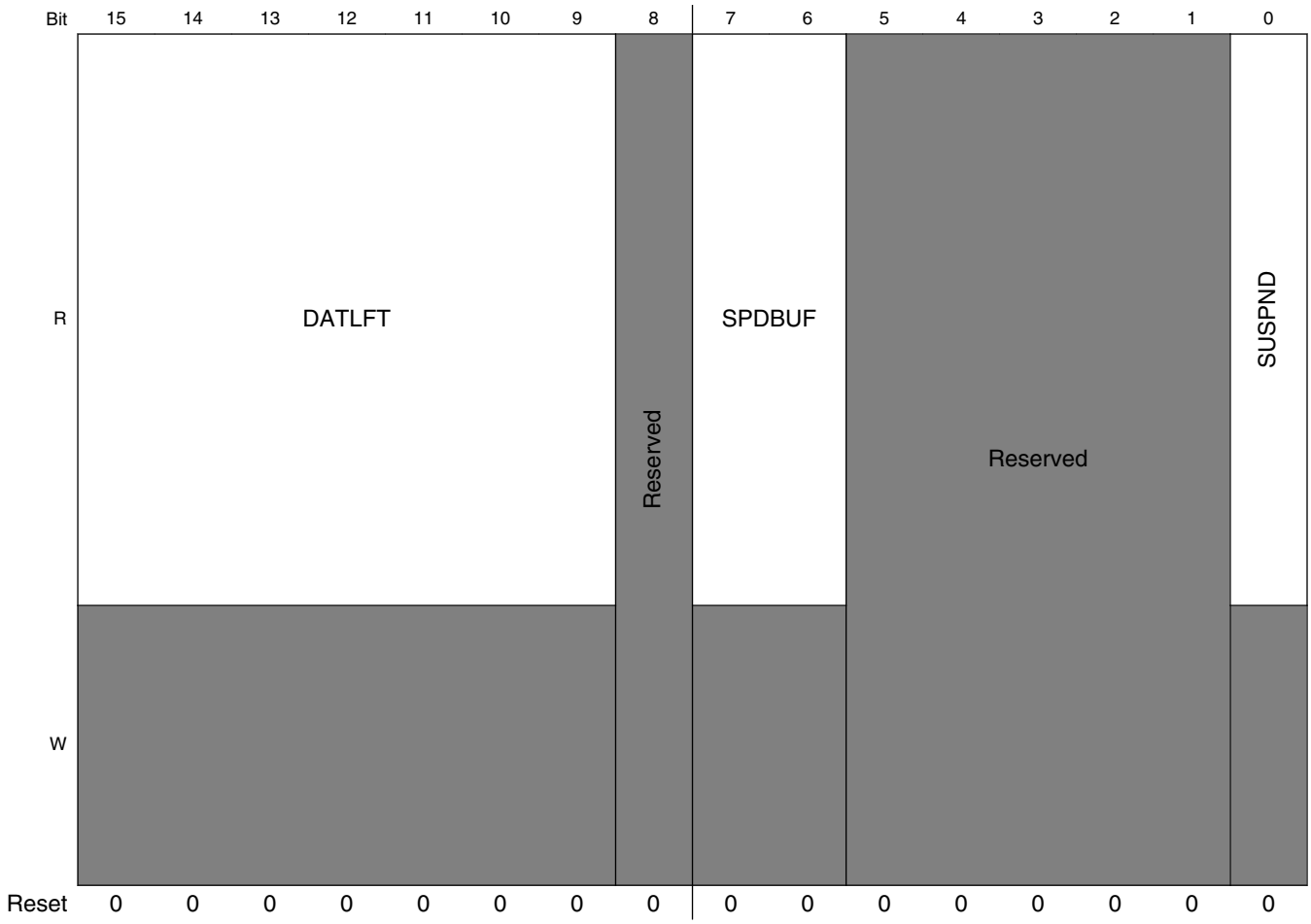
Field	Description
17 RBOIE	RX Buffer Overflow Interrupt Enable 0 No RBOF interrupt will be generated 1 RBOF interrupt will be generated
16 RBDIE	RX Buffer Drain Interrupt Enable: Enables generation of IRQ requests for RX Buffer Drain. When this bit is set the interrupt is asserted as long as the QuadSPI_SR[RBDF] flag is set. 0 No RBDF interrupt will be generated 1 RBDF Interrupt will be generated
15 ABSEIE	AHB Sequence Error Interrupt Enable: Triggered by ABSEF flags of QSPI_FR 0 No ABSEF interrupt will be generated 1 ABSEF interrupt will be generated
14 Reserved	Reserved This field is reserved.
13 Reserved	Reserved This field is reserved.
12 ABOIE	AHB Buffer Overflow Interrupt Enable 0 No ABOF interrupt will be generated 1 ABOF interrupt will be generated
11 IUEIE	IP Command Usage Error Interrupt Enable 0 No IUEF interrupt will be generated 1 IUEF interrupt will be generated
10–8 Reserved	This field is reserved.
7 IPAEIE	IP Command Trigger during AHB Access Error Interrupt Enable 0 No IPAEF interrupt will be generated 1 IPAEF interrupt will be generated
6 IPIEIE	IP Command Trigger during IP Access Error Interrupt Enable 0 No IPIEF interrupt will be generated 0 IPIEF interrupt will be generated
5 Reserved	This field is reserved.
4 IPGEIE	IP Command Trigger during AHB Grant Error Interrupt Enable 0 No IPGEF interrupt will be generated 1 IPGEF interrupt will be generated
3–1 Reserved	This field is reserved. Reserved.
0 TFIE	Transaction Finished Interrupt Enable 0 No TFF interrupt will be generated 1 TFF interrupt will be generated

10.2.13.21 Sequence Suspend Status Register (QuadSPI_SPNDST)

The sequence suspend status register provides information specific to any suspended sequence. An AHB sequence may be suspended when a high priority AHB master makes an access before the AHB sequence completes the data transfer requested.

Address: 30BB_0000h base + 168h offset = 30BB_0168h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0



QuadSPI_SPNDST field descriptions

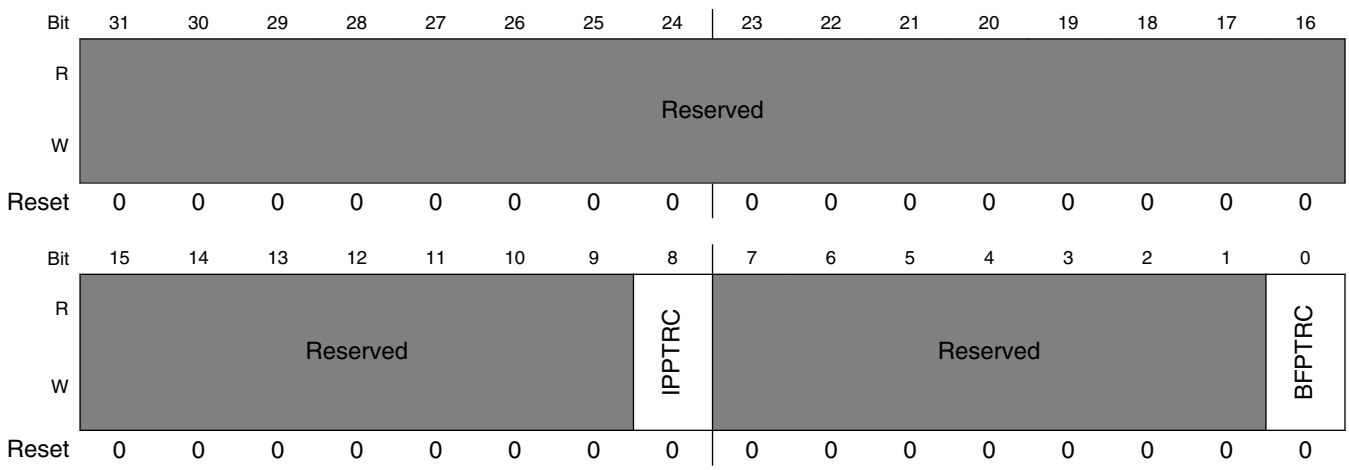
Field	Description
31–16 Reserved	This field is reserved.
15–9 DATLFT	Data left: Provides information about the amount of data left to be read in the suspended sequence. Valid only when SUSPND is set to 1'b1. Value in terms of 64 bits or 8 bytes
8 Reserved	This field is reserved.
7–6 SPDBUF	Suspended Buffer: Provides the suspended buffer number. Valid only when SUSPND is set to 1'b1
5–1 Reserved	This field is reserved.
0 SUSPND	When set, it signifies that a sequence is in suspended state

10.2.13.22 Sequence Pointer Clear Register (QuadSPI_SPTRCLR)

The sequence pointer clear register provides bits to reset the IP and Buffer sequence pointers. The sequence pointer contains the index of which instruction within the LUT entry is to be executed next. For example, if the LUT entry ends on a JMP_ON_CS value of 2, the index will be stored as 2.

The software should reset the sequence pointers whenever the sequence ID is changed by updating the SEQID field in QSPI_IPCR or QSPI_BFGENCR.

Address: 30BB_0000h base + 16Ch offset = 30BB_016Ch



QuadSPI_SPTRCLR field descriptions

Field	Description
31–9 Reserved	This field is reserved.
8 IPPTRC	IP Pointer Clear: 1: Clears the sequence pointer for IP accesses as defined in QuadSPI_IPCR
7–1 Reserved	This field is reserved. Reserved.
0 BFPTRC	Buffer Pointer Clear: 1: Clears the sequence pointer for AHB accesses as defined in QuadSPI_BFGENCR.

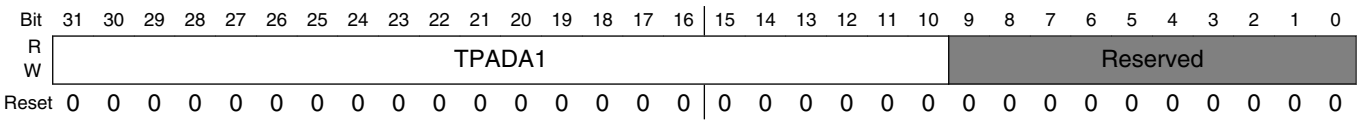
10.2.13.23 Serial Flash A1 Top Address (QuadSPI_SFA1AD)

The QSPI_SFA1AD register provides the address mapping for the serial flash A1.The difference between QSPI_SFA1AD[TPADA1] and QSPI_AMBA_BASE defines the size of the memory map for serial flash A1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 180h offset = 30BB_0180h



QuadSPI_SFA1AD field descriptions

Field	Description
31–10 TPADA1	Top address for Serial Flash A1. In effect, TPADxx is the first location of the next memory.
Reserved	This field is reserved.

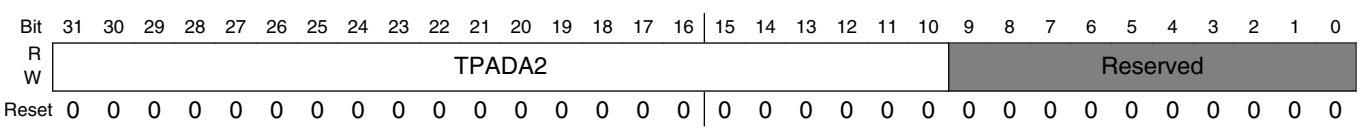
10.2.13.24 Serial Flash A2 Top Address (QuadSPI_SFA2AD)

The QSPI_SFA2AD register provides the address mapping for the serial flash A2.The difference between QSPI_SFA2AD[TPADA2] and QSPI_SFA1AD[TPADA1] defines the size of the memory map for serial flash A2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 184h offset = 30BB_0184h



QuadSPI_SFA2AD field descriptions

Field	Description
31–10 TPADA2	Top address for Serial Flash A2. In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

10.2.13.25 Serial Flash B1Top Address (QuadSPI_SFB1AD)

The QSPI_SFB1AD register provides the address mapping for the serial flash B1.The difference between QSPI_SFB1AD[TPADB1] and QSPI_SFA2AD[TPADA2] defines the size of the memory map for serial flash B1.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 188h offset = 30BB_0188h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB1																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SFB1AD field descriptions

Field	Description
31–10 TPADB1	Top address for Serial Flash B1.In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

10.2.13.26 Serial Flash B2Top Address (QuadSPI_SFB2AD)

The QSPI_SFB2AD register provides the address mapping for the serial flash B2.The difference between QSPI_SFB2AD[TPADB2] and QSPI_SFB1AD[TPADB1] defines the size of the memory map for serial flash B2.

Write:

- *QSPI_SR[IP_ACC] = 0*
- *QSPI_SR[AHB_ACC] = 0*

Address: 30BB_0000h base + 18Ch offset = 30BB_018Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TPADB2																						Reserved									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_SFB2AD field descriptions

Field	Description
31–10 TPADB2	Top address for Serial Flash B2. In effect, TPxxAD is the first location of the next memory.
Reserved	This field is reserved.

10.2.13.27 RX Buffer Data Register (QuadSPI_RBDR*n*)

The QuadSPI_RBDR registers provide access to the individual entries in the RX Buffer. Refer to Table 10-14 for the byte ordering scheme.

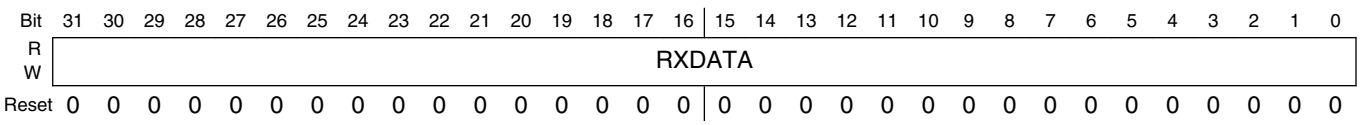
QuadSPI_RBDR0 corresponds to the actual position of the read pointer within the RX Buffer. The number of valid entries available depends from the number of RX Buffer entries implemented and from the number of valid buffer entries available in the RX Buffer.

Example 1, RX Buffer filled completely with 32 words: In this case the address range for valid read access extends from QuadSPI_RBDR0 to QuadSPI_RBDR31.

Example 2, RX Buffer filled with 5 valid words: RX Buffer fill level QuadSPI_RBSR[RDBFL] is 5. In this case an access to QuadSPI_RBDR4 provides the last valid entry.

Any access beyond the range of valid RX Buffer entries provides undefined results.

Address: 30BB_0000h base + 200h offset + (4d × i), where i=0d to 31d



QuadSPI_RBDR*n* field descriptions

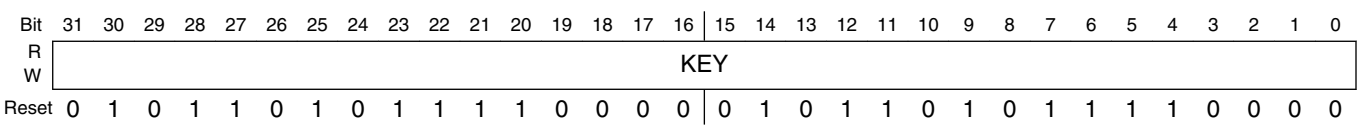
Field	Description
RXDATA	RX Data. The RXDATA field contains the data associated with the related RX Buffer entry. Data format and byte ordering is given in Byte Ordering of Serial Flash Read Data.

10.2.13.28 LUT Key Register (QuadSPI_LUTKEY)

The LUT Key register contains the key to lock and unlock the Look-up-table. Refer to Look-up Table for details.

Write:Anytime

Address: 30BB_0000h base + 300h offset = 30BB_0300h



QuadSPI_LUTKEY field descriptions

Field	Description
KEY	The key to lock or unlock the LUT. The KEY is 0x5AF05AF0. The read value is always 0x5AF05AF0

10.2.13.29 LUT Lock Configuration Register (QuadSPI_LCKCR)

The LUT lock configuration register is used along with QSPI_LUTKEY register to lock or unlock the LUT. This register has to be written immediately after QSPI_LUTKEY register for the lock or unlock operation to be successful. Refer to Look-up Table for details. Setting both the LOCK and UNLOCK bits as "00" or "11" is not allowed.

Write:Just after writing the LUT Key Register

(QSPI_LUTKEY)

Address: 30BB_0000h base + 304h offset = 30BB_0304h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														UNLOCK	LOCK
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

QuadSPI_LCKCR field descriptions

Field	Description
31–2 Reserved	This field is reserved.
1 UNLOCK	Unlocks the LUT when the following two conditions are met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key
0 LOCK	Locks the LUT when the following condition is met: 1. This register is written just after the LUT Key Register (QuadSPI_LUTKEY) 2. The LUT key register was written with 0x5AF05AF0 key

10.2.13.30 Look-up Table register (QuadSPI_LUT0)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: 30BB_0000h base + 310h offset = 30BB_0310h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0	0	0	0	1	0	0	0	0	0	0	1	1	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1

QuadSPI_LUT0 field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
OPRND0	Operand for INSTR0.

10.2.13.31 Look-up Table register (QuadSPI_LUT1)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: 30BB_0000h base + 314h offset = 30BB_0314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	1	1	1	0	0	0	0	0	0	1	0	0	0

QuadSPI_LUT1 field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
OPRND0	Operand for INSTR0.

10.2.13.32 Look-up Table register (QuadSPI_LUT*n*)

The LUT registers are a look-up-table for sequences of instructions. The programmable sequence engine executes the instructions in these sequences to generate a valid serial flash transaction. There are a total of 64 LUT registers. These 64 registers are divided into groups of 4 registers that make a valid sequence. Therefore, QSPI_LUT[0], QSPI_LUT[4], QSPI_LUT[8] ..... QSPI_LUT[60] are the starting registers of a valid sequence. Each of these sets of 4 registers can have a maximum of 8 instructions. A maximum of 16 sequences can be defined at one time. Look-up Table describes the LUT registers in detail.

Write:Once the LUT is unlocked

Address: 30BB_0000h base + 318h offset + (4d × i), where i=0d to 61d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INSTR1						PAD1		OPRND1							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INSTR0						PAD0		OPRND0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

QuadSPI_LUT*n* field descriptions

Field	Description
31–26 INSTR1	Instruction 1
25–24 PAD1	Pad information for INSTR1. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
23–16 OPRND1	Operand for INSTR1.
15–10 INSTR0	Instruction 0
9–8 PAD0	Pad information for INSTR0. 00 1 Pad 01 2 Pads 10 4 Pads 11 NA
OPRND0	Operand for INSTR0.

10.3 Ultra Secured Digital Host Controller (uSDHC)

10.3.1 Introduction

The uSDHC provides the interface between the host system and the SD/SDIO/MMC cards, as depicted in Figure 10-33. It acts as a bridge, passing host bus transactions to the SD/SDIO/MMC cards by sending commands and performing data accesses to/from the cards. It handles the SD/SDIO/MMC protocols at the transmission level. The following are brief descriptions of the cards supported by uSDHC:

The Multi Media Card (MMC) is a universal low-cost data storage and communication media designed to cover a wide array of applications including mobile video and gaming. Previous MMC cards were based on a 7-pin serial bus with a single data pin, while the new high speed MMC communication is based on an advanced 11-pin serial bus designed to operate in the low-voltage range.

The Secure Digital Card (SD) is an evolution of the old MMC technology. It is specifically designed to meet the security, capacity, performance, and environmental requirements inherent in newly emerging audio and video consumer electronic devices. The physical form factor, pin assignment, and data transfer protocol are forward compatible with the old MMC (with some additions).

Under the SD protocol, it can be categorized into memory card, I/O card, and combo card, which have both memory and I/O functions. The memory card invokes a copyright protection mechanism that complies with the security of the SDMI standard. The I/O card, which is also known as SDIO card, provides high-speed data I/O with low-power consumption for mobile electronic devices. For the sake of simplicity, the next figure does not show cards with reduced size or mini cards.

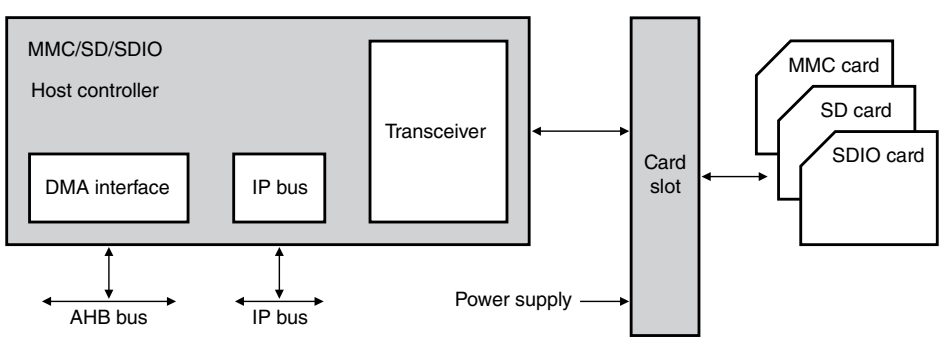


Figure 10-33. System connection of uSDHC

The following figure illustrates the block diagram of uSDHC.

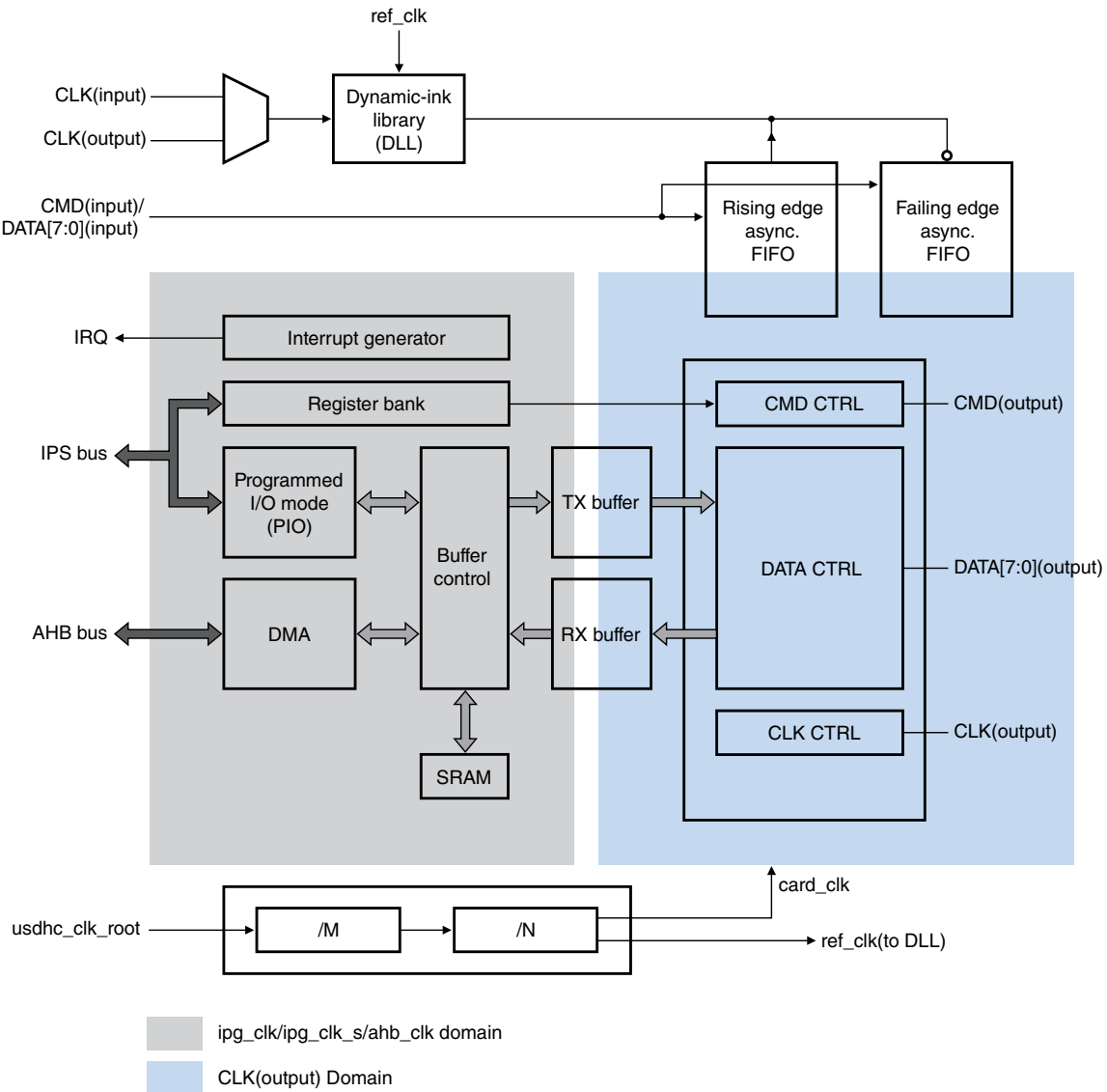


Figure 10-34. uSDHC block diagram

10.3.1.1 Features

The features of the uSDHC module include the following:

- Conforms to the SD Host Controller Standard Specification version 2.0/3.0
- Compatible with the MMC System Specification version 4.2/4.3/4.4/4.41/5.0/5.1
- Compatible with the SD Memory Card Specification version 3.0 and supports the Extended Capacity SD Memory Card
- Compatible with the SDIO Card Specification version 2.0/3.0
- Designed to work with SD Memory, miniSD Memory, SDIO, miniSDIO, SD Combo, MMC, MMC plus, and MMC RS cards
- Card bus clock frequency up to 208 MHz

- Supports 1-bit/4-bit SD and SDIO modes, and 1-bit/4-bit/8-bit MMC modes
  - Up to 832 Mbps of data transfer for SDIO cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDIO card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 832 Mbps of data transfer for SDXC cards using four parallel data lines in the Single Data Rate (SDR) mode
  - Up to 400 Mbps of data transfer for SDXC card using four parallel data lines in the Dual Data Rate (DDR) mode
  - Up to 416 Mbps of data transfer for MMC cards using eight parallel data lines in the Single Data Rate (SDR) mode
  - Up to 3200 Mbps of data transfer for MMC cards using eight parallel data lines in the Dual Data Rate (DDR) mode
- Supports a single block/multi-block read and write
- Supports block sizes of 1 ~ 4096 bytes
- Supports the write protection switch for write operations
- Supports both synchronous and asynchronous abort
- Supports pause during the data transfer at block gap
- Supports SDIO Read Wait and Suspend Resume operations
- Supports Auto CMD12 for multi-block transfer
- Host can initiate non-data transfer command while data transfer is in progress
- Allows cards to interrupt the host in 1-bit and 4-bit SDIO modes; also supports interrupt period
- Embodies two fully configurable 128x32-bit FIFO for read/write data
- Supports internaland external DMA capabilities
- Support voltage selection by configuring vendor-specific register bit
- Supports Advanced DMA to perform linked memory access
- support Command queue mechanism

10.3.1.2 Modes and operations

10.3.1.2.1 Data transfer modes

The uSDHC module can select the following modes for data transfer:

- SD 1-bit
- SD 4-bit
- MMC 1-bit
- MMC 4-bit
- MMC 8-bit
- Identification mode (up to 400 kHz)

- MMC full-speed mode (up to 26 MHz)
- MMC high-speed mode (up to 52 MHz)
- MMC HS400 mode (200 MHz both edges)
- MMC DDR mode (52 MHz both edges)
- SD/SDIO full-speed mode (up to 25 MHz)
- SD/SDIO high-speed mode (up to 50 MHz)
- SD/SDIO UHS-I mode(up to 208 MHz in SDR mode, up to 50 MHz in the DDR mode)

10.3.2 External signals

The following table describes the external signals of uSDHC:

Table 10-38. uSDHC external signals

Signal	Description	Direction
CLK	Clock for MMC/SD/SDIO card	O
CMD	CMD line connect to card	I/O
DATA7	DATA7 line in the 8-bit mode — Not used in other modes	I/O
DATA6	DATA6 line in the 8-bit mode — Not used in other modes	I/O
DATA5	DATA5 line in the 8-bit mode — Not used in other modes	I/O
DATA4	DATA4 line in the 8-bit mode — Not used in other modes	I/O
DATA3	DATA3 line in the 4/8-bit mode or configured as card detection pin. The bit may be configured as card detection pin in the 1-bit mode.	I/O
DATA2	DATA2 line or Read Wait in the 4-bit mode Read Wait in 1-bit mode	I/O
DATA1	DATA1 line in the 4/8-bit mode Also, used to detect interrupt in 1/4-bit mode	I/O
DATA0	DATA0 line in all the modes Also, used to detect busy state	I/O
CD_B	Card detection pin If not used (for the embedded memory), tie low to indicate that there is a card attached.	I
WP	Card write protect detect If not used (for the embedded memory), tie low to indicate that it is not write protected.	I

Table continues on the next page...

Table 10-38. uSDHC external signals (continued)

Signal	Description	Direction
LCTL	LED control used to drive an external LED active high, fully controlled by the driver Optional output	O
RESET_B	Card hardware reset signal, active low	O
VSELECT	IO power voltage selection signal	O
STROBE	Input clock for eMMC HS400 mode	I

10.3.2.1 Signals overview

uSDHC has 15 associated I/O signals.

- The CLK is an internally generated clock used to drive the MMC, SD, and SDIO cards.
- The CMD I/O is used to send commands and receive responses to and from the card. Eight data lines (DATA7~DATA0) are used to perform data transfers between the uSDHC module and the card.
- The CD_B and WP are card detection and write protection signals directly routed from the socket. These two signals are active low (0). A low on CD_B means that a card is inserted, and a high on WP means that the write protect switch is active.
- LCTL is an output signal used to drive an external LED to indicate that the SD interface is busy.
- RESET_B is an output signal used to reset the MMC card.
- VSELECT is an output signal used to change the voltage of the external power supplier.
- STROBE is input clock signal for eMMC HS400 mode.

CD_B, WP, LCTL, RESET_B, and VSELECT are all optional for system implementation. If uSDHC needs to support a 4-bit data transfer, DATA7~DATA4 can also be optional and tied to high. If the uSDHC does not support the HS400 mode, STROBE can also be optional and tied to low.

10.3.3 Functional description

The following sections provide a brief functional description of the major system blocks, including the data buffer, DMA AHB interface, register bank as well as IP Bus interface, dual-port memory wrapper, data/command controller, clock and reset manager, and clock generator.

10.3.3.1 Data buffer

The uSDHC module uses one configurable data buffer to transfer data between the system bus (IP bus or advanced high-performance bus (AHB) bus) and the SD card in an optimized manner, maximizing throughput between the two clock domains (IP peripheral clock and the master clock).

The buffer is used as a temporary storage for transferring data between the host system and the card. The watermark levels for read and write are both configurable and can range between 1 to 128 words. The burst lengths for read and write are also configurable and can range between 1 to 31 words. The next figure provides the uSDHC buffer scheme.

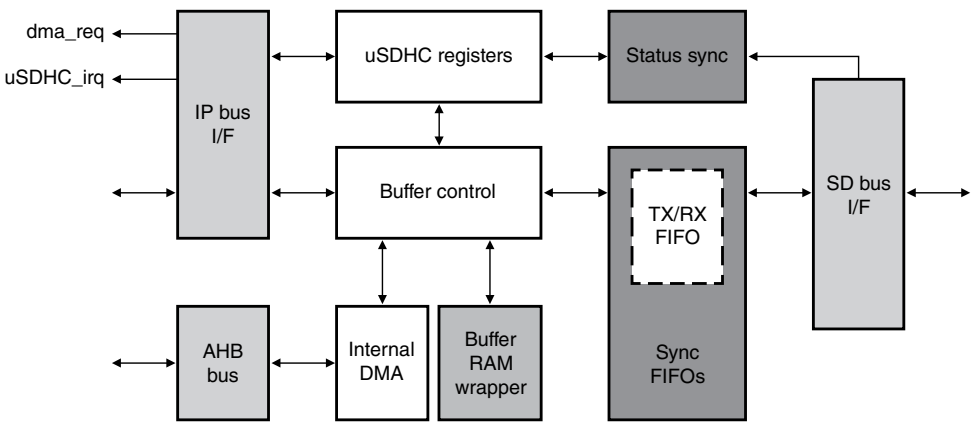


Figure 10-35. uSDHC buffer scheme

Here are 3 the two transfer modes to access the data buffer:

- CPU polling mode:
  - For a host-read operation, when the number of words received in the buffer meets or exceeds the RD_WML watermark value, by polling the BRR bit, the host driver can read the Buffer Data Port register to fetch the amount of words set in the RD_WML register from the buffer. The write operation is similar. For more information on the process of writing operation, see Write operation sequence.
- External DMA mode:
  - For a read operation, when there are more words received in the buffer than the amount set in the RD_WML register, a DMA request is sent out to inform the external DMA to fetch the data. The request will be immediately de-asserted when there is an access on the Buffer Data Port register. If the number of words in the buffer after the current burst meets or exceeds RD_WML value, the DMA request is asserted again. For instance, if there are twice as many words in the

buffer as there are in the RD_WML value, there are two successive DMA requests with only one cycle of de-assertion between. The write operation is similar. Note the accesses CPU polling mode and external DMA mode both use the IP bus, and if the external DMA is enabled, in both modes an external DMA request is sent when the buffer is ready.

- Internal DMA mode (includes simple and advanced DMA accesses):
  - The internal DMA access, either by simple or advanced DMA, is over the AHB bus. For internal DMA access mode, the external DMA request will never be sent out.

For a read operation, when there are more words in the buffer than the amount set in the RD_WML register, the internal DMA starts fetching data over the AHB bus. Except for INCR4 and INCR8, the burst type is always the INCR mode and the burst length depends on the shortest of the following factors:

- Burst length configured in the burst length field of the Watermark Level register
- Watermark level boundary
- Block size boundary
- Data boundary configured in the current descriptor (if the ADMA is active)
- 1 KB address boundary defined in the AHB protocol

The Write operation functions in a similar manner—sequential and contiguous access is necessary to ensure that the pointer address value is correct. Random or skipped access is not possible. The byte order, by reset, is little endian mode. The actual byte order is swapped inside the buffer, according to the endian mode configured by software (see the following figures). For a host write operation, the byte order is swapped after the data is fetched from the buffer and ready to send to the SD Bus. For a host read operation, the byte order is swapped before the data is stored in the buffer.

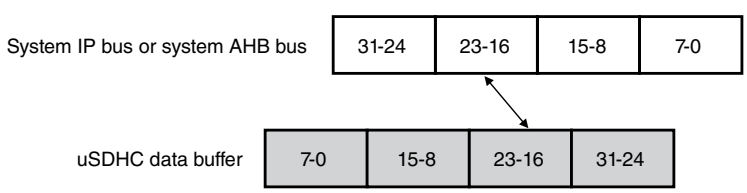


Figure 10-36. Data swap between system bus and uSDHC data buffer in the byte little endian mode

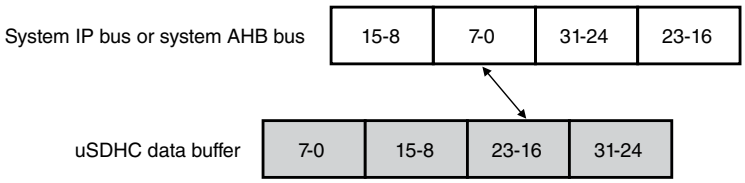


Figure 10-37. Data swap between system bus and uSDHC data buffer in half word big endian mode

10.3.3.1.1 Write operation sequence

There are 3 ways to write data into the buffer when the user transfers data to the card:

- External DMA through the uSDHC DMA request signal
- Processor core polling through the BWR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When the internal DMA is not used, the DMAEN bit in the Transfer Type register is not set when the command is sent, uSDHC asserts a DMA request when the amount of buffer space exceeds the value set in the WR_WML register and is ready for receiving new data. At the same time, uSDHC sets the BWR bit. The buffer write ready interrupt is generated if it is enabled by software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the contents of the DMA System Address register to obtain the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the write operation from that address. It is recommended that a software reset for Data be applied before the transfer is restarted.

The uSDHC module does not start data transmission until the number of words set in the WR_WML register can be held in the buffer. If the buffer is empty and the host system does not write data in time, uSDHC stops the CLK to avoid the data buffer underrun situation.

10.3.3.1.2 Read operation sequence

There are 3 ways to read data from the buffer when the user transfers data to the card:

- External DMA through uSDHC DMA request signal

- Processor core polling through the BRR bit in the Interrupt Status register (interrupt or polling)
- Internal DMA

When internal DMA is not used (DMAEN bit in Transfer Type register is not set when the command is sent), uSDHC asserts a DMA request when the amount of data exceeds the value set in the RD_WML register, which is available and ready for system fetching data. At the same time, uSDHC sets the BRR bit. The buffer read ready interrupt is generated if it is enabled by the software.

When internal DMA is used, uSDHC does not inform the system before all the required number of bytes are transferred (if no error is encountered). When an error occurs during the data transfer, uSDHC aborts the data transfer and abandons the current block. The host driver should read the content of the DMA System Address register to get the starting address of the abandoned data block. If the current data transfer is in multi-block mode, uSDHC does not automatically send CMD12, even though the AC12EN bit in the Transfer Type register is set. The host driver sends CMD12 in this scenario and restarts the read operation from that address. It is recommended that a software reset for data be applied before the transfer is restarted.

For any write transfer mode, uSDHC does not start data transmission until the number of words set in the RD_WML register are in buffer. If the buffer is full and the host system does not read data in time, uSDHC stops the CLK to avoid the data buffer overrun situation.

10.3.3.1.3 Data buffer and block size

The user needs to know the buffer size for the buffer operation during a data transfer to utilize it in the most optimized way. In the uSDHC module, the only data buffer can hold up to 128 words (32-bit) and the watermark levels for write and read can be configured accordingly.

For both read and write, the watermark level can range between 1 to 128 words. For both read and write, the burst length can range between from 1 to 31 words. The host driver may configure the value according to the system situation and requirement.

During a multi-block data transfer, the block length can be set to any value between 1 and 4096 bytes, satisfying the requirements of the external card. The only restriction is from the external card, which can be limited in size or support of a partial block access (which is not the integer times of 512 bytes).

As uSDHC treats each block individually, for block sizes which are not multiples of four (not word-aligned), stuffed bytes are required at the end of each block. For example, if the block size is 7 bytes and there are 12 blocks to write, the system side must write twice

for each block. For each block, the ending byte is abandoned by uSDHC because it only sends 7 bytes to the card and picks data from the following system write, resulting in 24 beats of write access in total.

10.3.3.1.4 Dividing large data transfer

This SDIO command CMD53 definition limits the maximum data size of data transfers according to the following formula:

Max data size = Block size x Block count

The length of a multiple block transfer needs to be in block size units. If the total data length cannot be divided evenly into a multiple of the block size, then there are two ways to transfer the data. These two ways depend on the function and the card design. Option 1 is for the host driver to split the transaction. The remainder of the block size data is then transferred by using a single block command at the end. Option 2 is to add dummy data in the last block to fill the block size. For option 2, the card must manage the removal of the dummy data.

See the figure below for an example showing the division of large data transfers, assuming a kind of WLAN SDIO card that only supports a block size of up to 64 bytes. Although uSDHC supports a block size of up to 4096 bytes, the SDIO can only accept a block size of less than 64 bytes, so the data must be divided (see the example below).

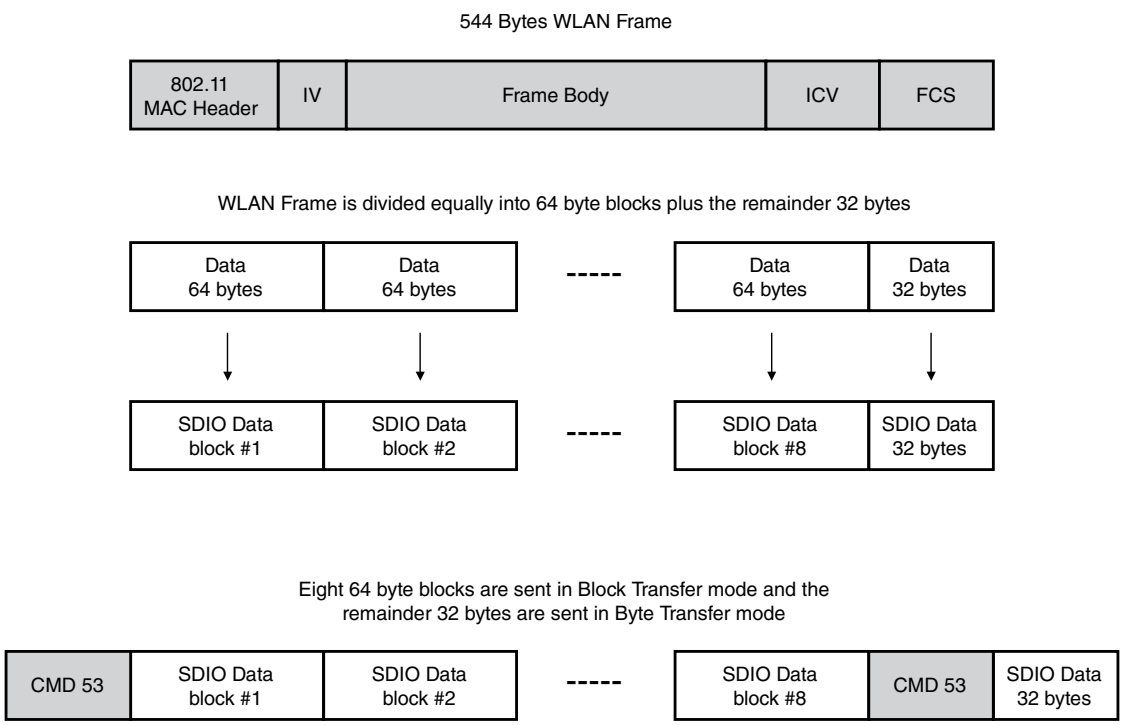


Figure 10-38. Example for dividing large data transfers

10.3.3.1.5 External DMA Request

When the internal DMA is not in use and external DMA is enabled, the Data Buffer will generate a DMA request to the system. During a write operation, when the number of WR_WML words can be held in the buffer free space, the signal uSDHC_dreq_b is asserted to 0, informing the Host System of a DMA write.

The BWR bit in the Interrupt Status register is also set, as long as the BWRSEN bit in the Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port register are made while the buffer's free space can't meet the watermark condition (free space> write watermark level).

On read operation, when the number of RD_WML words are already in the buffer, the signal uSDHC_dreq_b is asserted to 0, informing the Host System for a DMA read. The BRR bit in the Interrupt Status register is also set, as long as the BRRSEN bit in the Interrupt Status Enable register is set. The DMA request is de-asserted after several accesses to the Data Port register are made while the buffer's data can't meet the watermark condition (the number of data in buffer > read watermark level).

If the DMA burst length can't change during a data transfer for an external DMA transfer, the watermark level (read or write) must be a divisor of the block size. If it is not, transferring the block may cause buffer under-run (read operation) or over-run (write operation). For example, if the block size is 512 bytes, the watermark level of read (or write) must be a power of two between 1 and 128. For processor core polling access there is no such issue, as the last access in the block transfer can be controlled by software. The watermark level can be any value, even larger than the block size (but no greater than 128 words) because the actual number of bytes transferred by the software can be controlled and does not exceed the block size in each transfer.

The uSDHC also supports non-word aligned block size, as long as the card supports that block size. In this case, the watermark level should be set as the number of words. For example, if the block size is 31 bytes, the watermark level can be set to any number of words. For this case, the BLKSIZE bits of the Block Attribute register will be set as 1fh. For the CPU polling access, the burst length can be 1 to 128 words, without restriction. This is because the software will transfer 8 words, and the uSDHC will also set the BWR or BRR bits when the remaining data does not violate data buffer. See DMA burst length for more details about the dynamic watermark level of the data buffer.

For the above example, even though 8 words are transferred via the Data Port register, the uSDHC will transfer only 31 bytes over the SD Bus, as required by the BLKSIZE bits. In this data transfer, with non-word aligned block size, the endian mode should be set cautiously or invalid data will be transferred to and from the card.

10.3.3.2 DMA AHB interface

The internal DMA implements a DMA engine and the AHB master. When the internal DMA is enabled, the uSDHC_dreq_b is not asserted during the transfer, but the BWR and BRR bits are set if the BWRSEN and BRRSEN bits have been set in the Interrupt Status Enable register.

See the figure below for an illustration of the DMA AHB interface block.

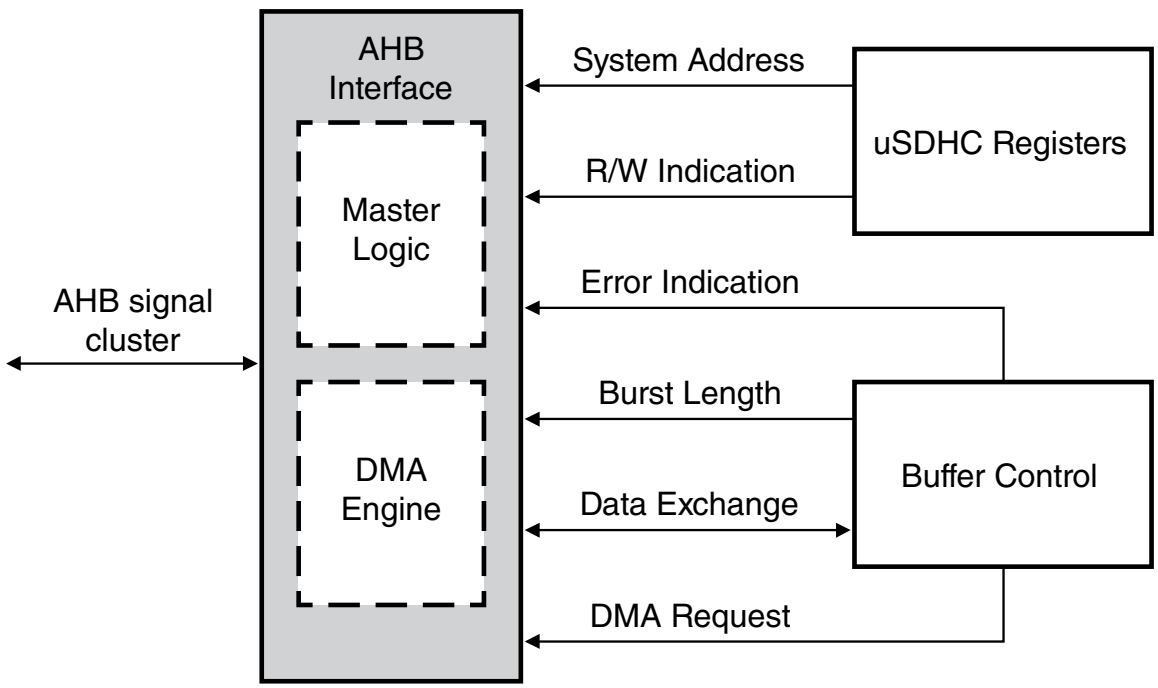


Figure 10-39. DMA AHB interface block

10.3.3.2.1 Internal DMA request

If the watermark level requirement is met in data transfer or if the last data of current block is ready in the data buffer, and the Internal DMA is enabled, the data buffer block sends a DMA request to AHB interface.Meanwhile, the external DMA request signal (uSDHC_dreq_b) is disabled.

The delay in response from the internal DMA engine depends on the system AHB bus loading and the priority assigned to uSDHC. The DMA engine does not respond to the request during its burst transfer, but is ready to serve as soon as the burst is over. The data buffer de-asserts the request if the data buffer space (for write) or bytes in data buffer is

smaller than the watermark level. Upon access to the buffer by internal DMA, the data buffer updates its internal buffer pointer, and when the watermark level is satisfied or the last data of the current block is ready in the data buffer, another DMA request is sent.

The data transfer is in the block unit, and the subsequent watermark level is always set as the remaining number of words. For instance, for a multi block data read with each block size of 31 bytes, and the burst length set to six words. After the first burst transfer, if there are more than two words in the buffer, which might contain some data of the next block), another DMA request is sent. This is because the remaining number of words to send for the current block is (31 - 6 * 4) / 4 = 2. The uSDHC module reads two words out of the buffer, with seven valid bytes and one stuffed byte.

10.3.3.2.2 DMA burst length

Just like a CPU polling access, the DMA burst length for the internal DMA engine can range between 1 to 16 words. The actual burst length for the DMA depends on the lesser of the configured burst length or the remaining words of the current block. See the example in Internal DMA request. After six words are read, the burst length is two words, then the next burst length is six words. This is because the next block starts, which is 31 bytes, more than six words. The host driver may take this variable burst length into account. It is also acceptable to configure the burst length as the divisor of the block size, so that each time the burst length is the same.

10.3.3.2.3 AHB master interface

It is possible that the internal AHB DMA engine could fail during the data transfer. Upon detection of an AHB bus error during DMA transfer, the DMA engine stops the transfer and goes to the idle state. At that point, the internal data buffer stops receiving incoming data and sending out data. The DMAE bit in the Interrupt Status register is generated to host CPU to report a bus error condition.

After the DMAE interrupt is received, the software sends a CMD12 to abort the current transfer and read the DS_ADDR bits of the DMA System Address register to get the starting address of the corrupted block. After the DMA error is fixed, the software should apply a data reset and restart the transfer from this address to recover the corrupted block. DMA operation resumes when the interrupt is serviced by the software.

10.3.3.2.4 ADMA engine

In the SD host controller standard, a new DMA transfer algorithm called the Advanced DMA (ADMA) is defined. For simple DMA, after the page boundary is reached, a DMA interrupt is generated and the new system address is programmed by the host driver.

The ADMA defines the programmable descriptor table in the system memory. The host driver can calculate the system address at the page boundary and program the descriptor table before executing ADMA. It reduces the frequency of interrupts to the host system. Therefore, higher speed DMA transfers could be realized because the host MCU intervention is not be needed during long DMA-based data transfers.

There are two types of ADMA in host controller: ADMA1 and ADMA2. ADMA1 can support data transfer of 4KB aligned data in system memory, and ADMA2 improves the restriction so that the data of any location and any size can be transferred in system memory. Their formats of Descriptor Table are different.

ADMA can recognize all kinds of descriptors defined in SD host controller Standard, and if the "End" flag is detected in the descriptor, ADMA stops after this descriptor is processed.

10.3.3.2.4.1 ADMA concept and descriptor format

ADMA1 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Set data length descriptor
- Set data address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA2 includes the following descriptors:

- Valid/invalid descriptor
- Nop descriptor
- Rsv descriptor
- Set data length and address descriptor
- Link descriptor
- Interrupt flag and end flag in descriptor

ADMA starts read/write operation after it reaches the tran state, using the data length and data address analyzed from most recent descriptor(s).

For ADMA1, the valid data length descriptor is the last set type descriptor before the tran type descriptor. Every tran type triggers a transfer, and the transfer data length is extracted from the most recent set type descriptor. If there is no set type descriptor after the previous Trans descriptor, the data length is the value for previous transfer, or 0 if no set descriptor is ever met.

For ADMA2, the tran type descriptor contains both data length and transfer data address, so only a tran type descriptor can start a data transfer.

See the figure below for the format of the descriptor table for ADMA1.

Address/ Page Field		Address/ Page Field		Attribute Field					
31	12	11	6	5	4	3	2	1	0
Address or Data Length		000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	31- 28	27- 12
0	0	Nop	No Operation	Don't Care	
0	1	Set	Set Data Length	0000	Data Length
1	0	Tran	Transfer Data	Data Address	
1	1	Link	Link Descriptor	Descriptor Address	

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 10-40. Format of the ADMA1 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA1.

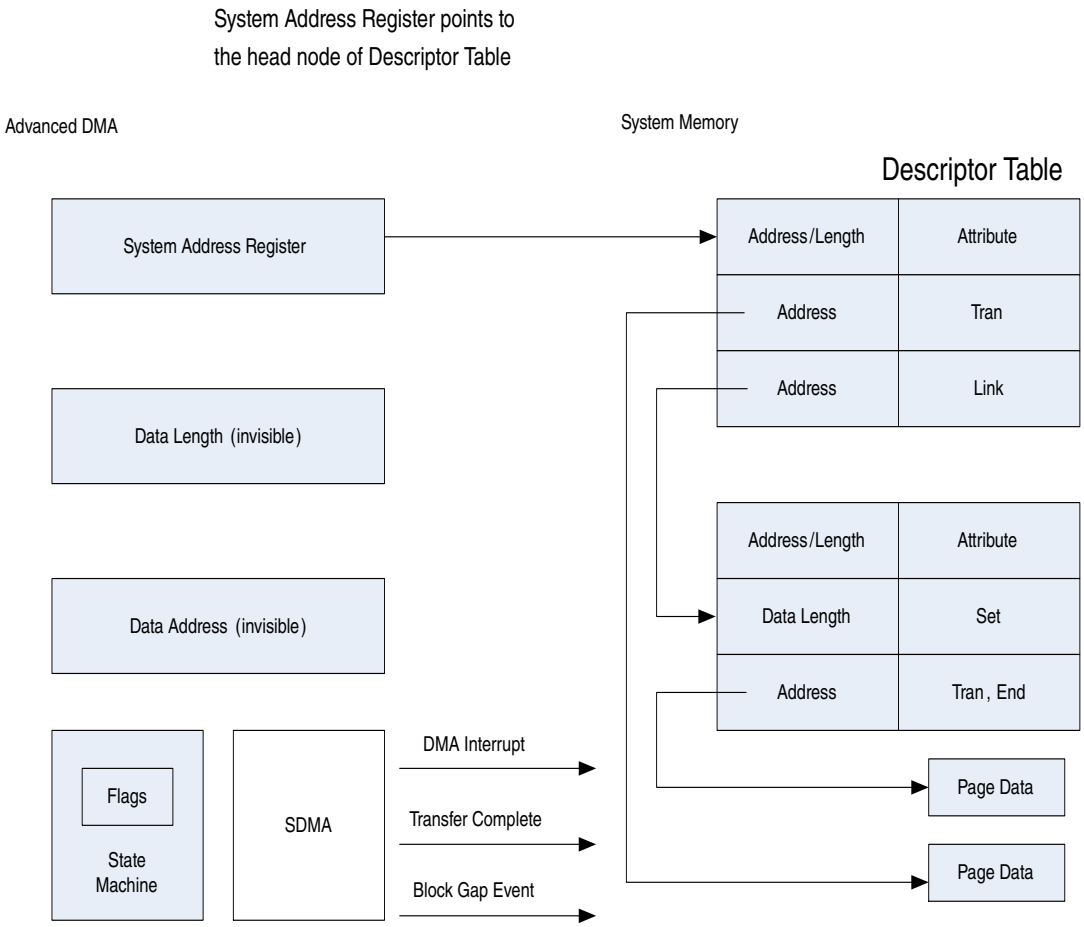


Figure 10-41. Concept and access method of the ADMA1 descriptor table

The figure below explains the ADMA2 format. ADMA2 deals with the lower 32-bit first, and then the higher 32-bit. If the 'Valid' flag of descriptor is 0, it ignores the higher 32-bit. The Address field should be set to word aligned (lower 2-bit is always set to 0). Data length is in byte unit.

Address Field		Length		Reserved		Attribute Field					
63	32	31	16	15	06	05	04	03	02	01	00
32-bit Address		16-bit length		0000000000		Act 2	Act 1	0	Int	End	Valid

Act 2	Act1	Symbol	Comment	Operation
0	0	Nop	No Operation	Don't Care
0	1	Rsv	Reserved	Same as Nop. Read this line and go to next one
1	0	Tran	Transfer Data	Transfer data with address and length set in this descriptor line
1	1	Link	Link Descriptor	Link to another descriptor

Valid	Valid = 1 indicates this line of descriptor is effective. If Valid = 0 generate ADMA Error Interrupt and stop ADMA.
End	End = 1 indicates current descriptor is the ending one.
Int	Int = 1 generates DMA Interrupt when this descriptor is processed.

Figure 10-42. Format of the ADMA2 descriptor table

See the figure below for the concept and access method of the descriptor table for ADMA2.

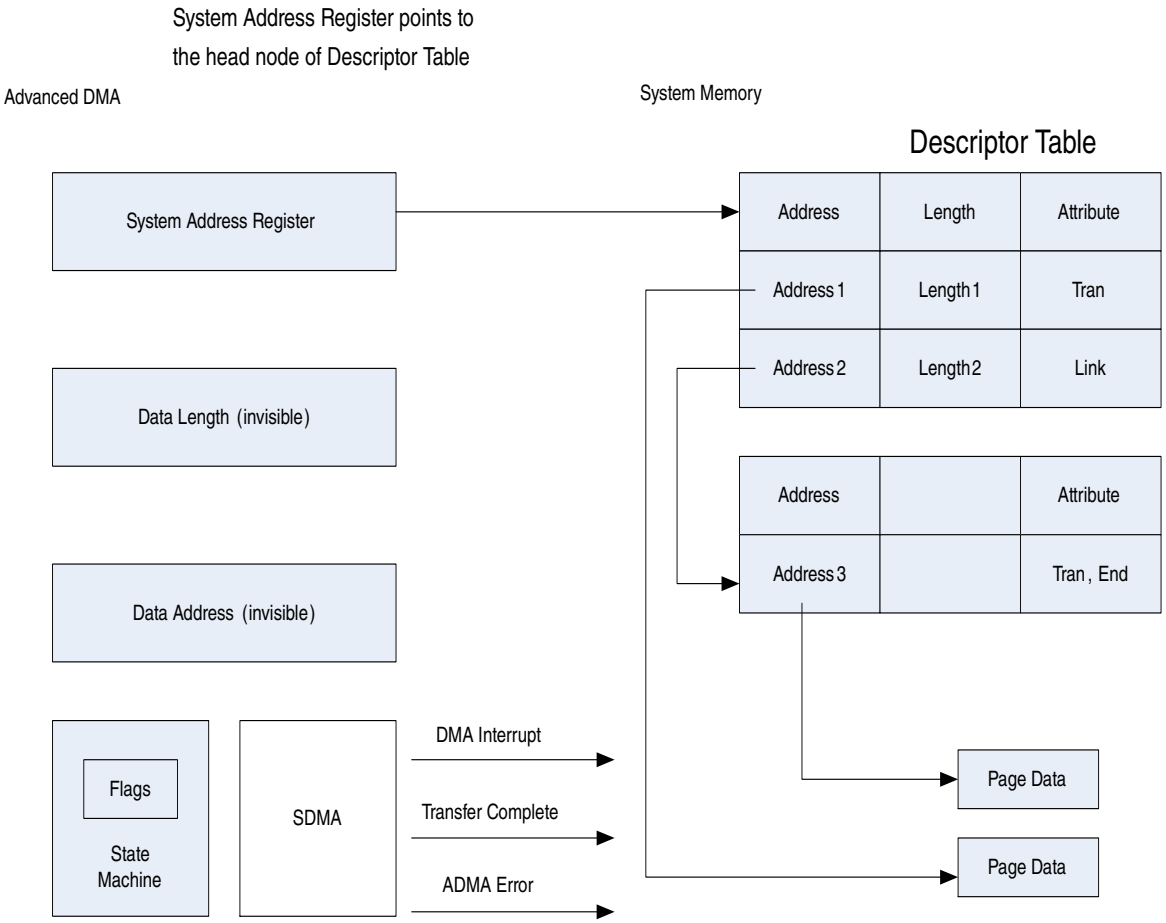


Figure 10-43. Concept and access method of the ADMA2 descriptor table

10.3.3.2.4.2 ADMA interrupt

If the interrupt flag descriptor is set, ADMA generates an interrupt according to various types of descriptors.

For ADMA1:

- Set type of descriptor: interrupt is generated when data length is set.
- Tran type descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop type of descriptor: interrupt is generated just after this descriptor is fetched.

For ADMA2:

- Tran type of descriptor: interrupt is generated when this transfer is complete.
- Link type of descriptor: interrupt is generated when new descriptor address is set.
- Nop/Rsv type of descriptor: interrupt is generated just after this descriptor is fetched.

10.3.3.2.4.3 ADMA error

The ADMA stops whenever an error is encountered. These errors include:

- Fetching descriptor error
- AHB response error
- Data length mismatch error

An ADMA descriptor error is generated when it fails to detect a "valid" flag in the descriptor. If an ADMA descriptor error occurs, the interrupt is not generated even if the "Interrupt" flag of this descriptor is set.

When BLKCNTEN bit is set, data length set in buffer must be equal to the whole data length set in descriptor nodes, otherwise data length mismatch error is generated.

When BLKCNTEN bit is not set, then the whole data length set in descriptor is a multiple of block lengths; otherwise, when data set in the descriptor nodes are not performed at block boundaries, then data mismatch errors occur.

10.3.3.3 Register bank with IP bus interface

Register accesses through the IP bus interface are on the register bank.

See the figure below for the block diagram.

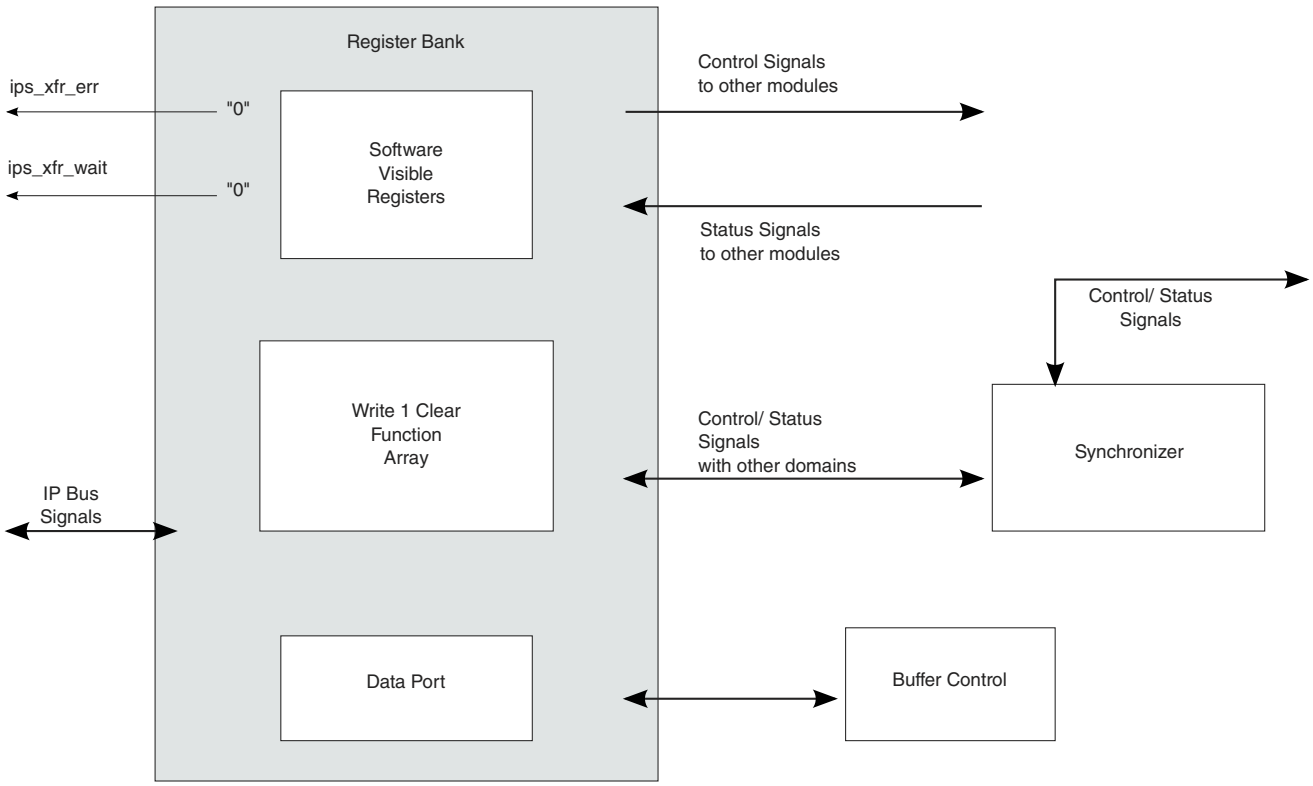


Figure 10-44. Register bank diagram

Only a 32-bit access is allowed, and no partial read/write is supported; therefore, all accesses are word aligned.

10.3.3.3.1 SD protocol unit

The SD protocol unit deals with all SD protocol affairs.

The SD protocol unit performs the following functions:

- Acts as the bridge between the internal buffer and the SD bus
- Sends the command data as well as its argument serially
- Stores the serial response bit stream into corresponding registers
- Detects the bus state on the CMD/DATA lines
- Monitors the interrupt from the SDIO card
- Asserts the read wait signal
- Gates off the SD clock when buffer is announcing danger status
- Detects the write protect state

The SD protocol unit consists of four submodules:

- SD control misc
- Command control

- Data control
- Clock control

10.3.3.3.2 SD control misc

In the SD control misc unit, the card detection (including the CD_B and DATA3 used for card detection), write protection, and card interrupt are implemented.

This module monitors the signal level on all the eight data lines and the command lines. It directly routes the level values into the register bank.

The module also detects the Write Protect (WP) line. If WP is active, writes to the register bank are ignored.

This module also drives the LCTL output signal when the LCTL bit is set by the driver.

10.3.3.3.3 SD clock control

If the internal data buffer is near full (for read) or near empty (for write), the SD clock must be gated off to avoid buffer over/under-run. This module asserts the gate of the output SD clock to shut the clock off. After the buffer has space (for read) or has data (for write), the clock gate of this module opens, and the SD clock is active again.

10.3.3.3.4 Command control

The Command Control module deals with the transactions on the CMD line.

See the figure below for an illustration of the structure for the Command CRC shift register.

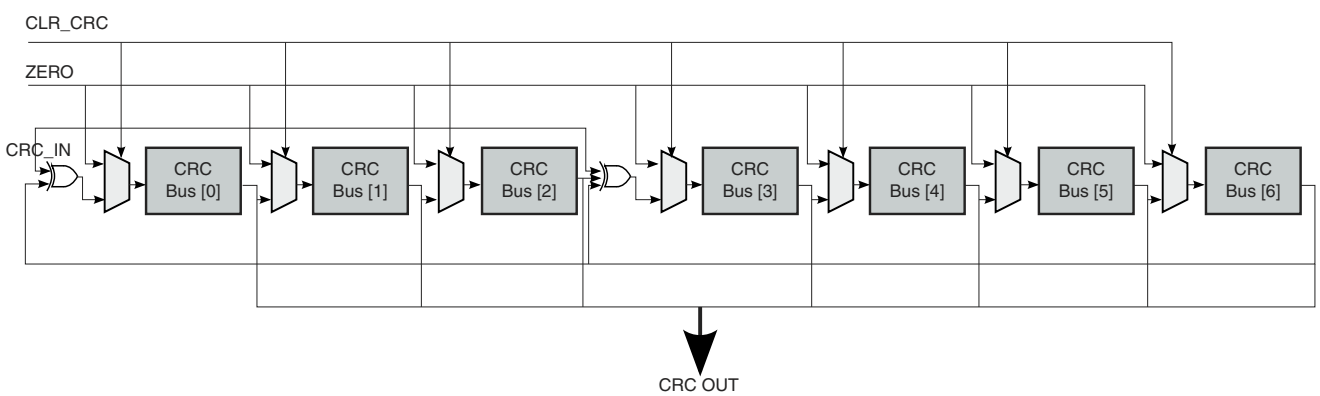


Figure 10-45. Command CRC shift register

The CRC polynomials for the CMD are as follows:

Generator polynomial: $G(x) = x^7 + x^3 + 1$
$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \ldots + (\text{last bit}) * x^0$
$CRC[6:0] = \text{Remainder } [(M(x) * x^7) / G(x)]$

10.3.3.3.5 Data control

The data agent deals with the transactions on the eight data lines. Moreover, this module also detects the busy state on the DATA0 line, and generates the read wait state by the request from the transceiver.

The CRC polynomials for the data are as follows:

Generator polynomial: $G(x) = x^{16} + x^{12} + x^5 + 1$

$M(x) = (\text{first bit}) * x^n + (\text{second bit}) * x^{n-1} + \ldots + (\text{last bit}) * x^0$

$CRC[15:0] = \text{Remainder } [(M(x) * x^{16}) / G(x)]$

10.3.3.4 Clock and reset manager

This module controls all four kinds reset signals within uSDHC:

- Hardware reset
- Software reset for all logic
- Software reset for the data logic
- Software reset for the command logic

All these signals are fed into this module and stable signals are generated inside the module to reset all other modules. The module also gates off all the inside signals.

10.3.3.5 Clock generator

The clock generator generates the card CLK by peripheral source clock in two stages. The clock divisor can be configured through register SYS_CTRL [SDCLKFS] is for prescaler configuration while the [DVS] is for divisor configuration. Details can be found in the register function description. See the following figure for the structure of the divider. The term "Base" represents the frequency of peripheral source clock.

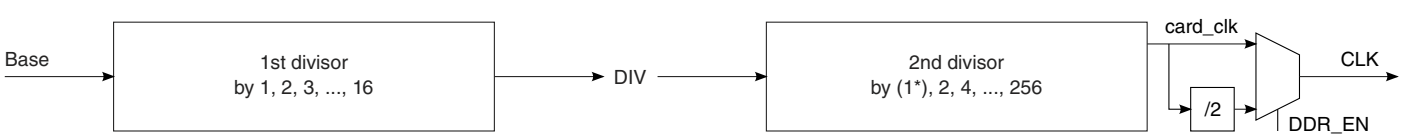


Figure 10-46. Two stages of the clock divider

The first stage outputs an intermediate clock (DIV) that can be Base, Base/2, Base/3, ..., or Base/16.

The second stage is a prescaler and outputs the actual internal working clock (card_clk). This clock is the driving clock for all the sub modules of the SD protocol unit, and helps in syncing FIFOs (see Figure 10-35) to synchronize with the data rate from the internal data buffer. The frequency of the clock output from this stage can be DIV, DIV/2, DIV/4,..., or DIV/256. Therefore, the highest frequency of the card_clk is base, and the next highest is Base/2, while the lowest frequency is Base/4096. If the duty cycle of the base clock is 50%, the duty cycle of card_clk is also 50%, even when the compound divisor is an odd value.

NOTE

CLK is different for the SDR and DDR modes.

- In the SDR mode, CLK is equal to the internal working clock (card_clk).

- In the DDR mode, CLK is equal to card_clk/2.

10.3.3.6 SDIO card interrupt

Information on interrupts in 1-bit mode, interrupts in 4-bit mode, and card interrupt handling are detailed in the sections below.

10.3.3.6.1 Interrupts in 1-bit mode

In this case, the DATA1 pin provides the interrupt function. An interrupt is asserted by pulling the DATA1 low from the SDIO card, until the interrupt service is finished to clear the interrupt.

10.3.3.6.2 Interrupt in 4-bit mode

As the interrupt and data line 1 share pin 8 in a 4-bit mode, an interrupt is only sent by the card and recognized by the host during a specific time. This is known as the interrupt period. The uSDHC module only provides sample the level on pin 8 during the interrupt period. At all other times, the host ignores the level on pin 8 and treats it as the data signal. The definition of the interrupt period is different for operations with single block and multiple block data transfers.

In the case of normal single data block transmissions, the interrupt period becomes active two clock cycles after the completion of a data packet. This interrupt period lasts until after the card receives the end bit of the next command that has a data block transfer associated with it.

For multiple block data transfers in a 4-bit mode, there is only a limited period of time that the interrupt period can be active because of the limited period of data line availability between the multiple blocks of data. This requires stricter definition of the interrupt period. For this case, the interrupt period is limited to two clock cycles. This begins two clocks after the end bit of the previous data block. During this 2-clock cycle interrupt period, if an interrupt is pending, the DATA1 line holds low for one clock cycle with the last clock cycle pulling DATA1 high. On completion of the interrupt period, the card releases the DATA1 line into the high Z state. The uSDHC module provides sample of the DATA1 during the interrupt period when the IABG bit in the Protocol Control register is set.

Refer to SDIO Card Specification v1.10 for further information about the SDIO card interrupt.

10.3.3.6.3 Card interrupt handling

When the CINTIEN bit in the Interrupt Signal Enable Register is set to 0, uSDHC clears the interrupt request to the host system. The host driver should clear this bit before servicing the SDIO interrupt and should set this bit again after all interrupt requests from the card are cleared to prevent inadvertent interrupts.

The SDIO Card Interrupt Status can be cleared by writing 1 to this bit. But as the interrupt source from the SDIO card does not clear, this bit is set again. To clear this bit, it is required to reset the interrupt source from the external card followed by writing 1 to this bit. In a 1-bit mode, uSDHC detects the SDIO interrupt with or without the SD clock (to support wakeup). In a 4-bit mode, the interrupt signal is sampled during the interrupt period, so there are some sample delays between the interrupt signal from the SDIO card and the interrupt to the Host System Interrupt Controller. When the SDIO status is set and the host driver needs to service this interrupt, the SDIO bit in the Interrupt Control Register of SDIO card is cleared. This is required to clear the SDIO interrupt status latched in uSDHC and to stop driving the interrupt signal to the System Interrupt Controller. The host driver must issue a CMD52 to clear the card interrupt. After completion of the card interrupt service, the SDIO Interrupt Status Enable bit is set to 1 and uSDHC starts sampling the interrupt signal again.

See the figure below for an illustration of the SDIO card interrupt scheme and for the sequences of software and hardware events that take place during a card interrupt handling procedure.

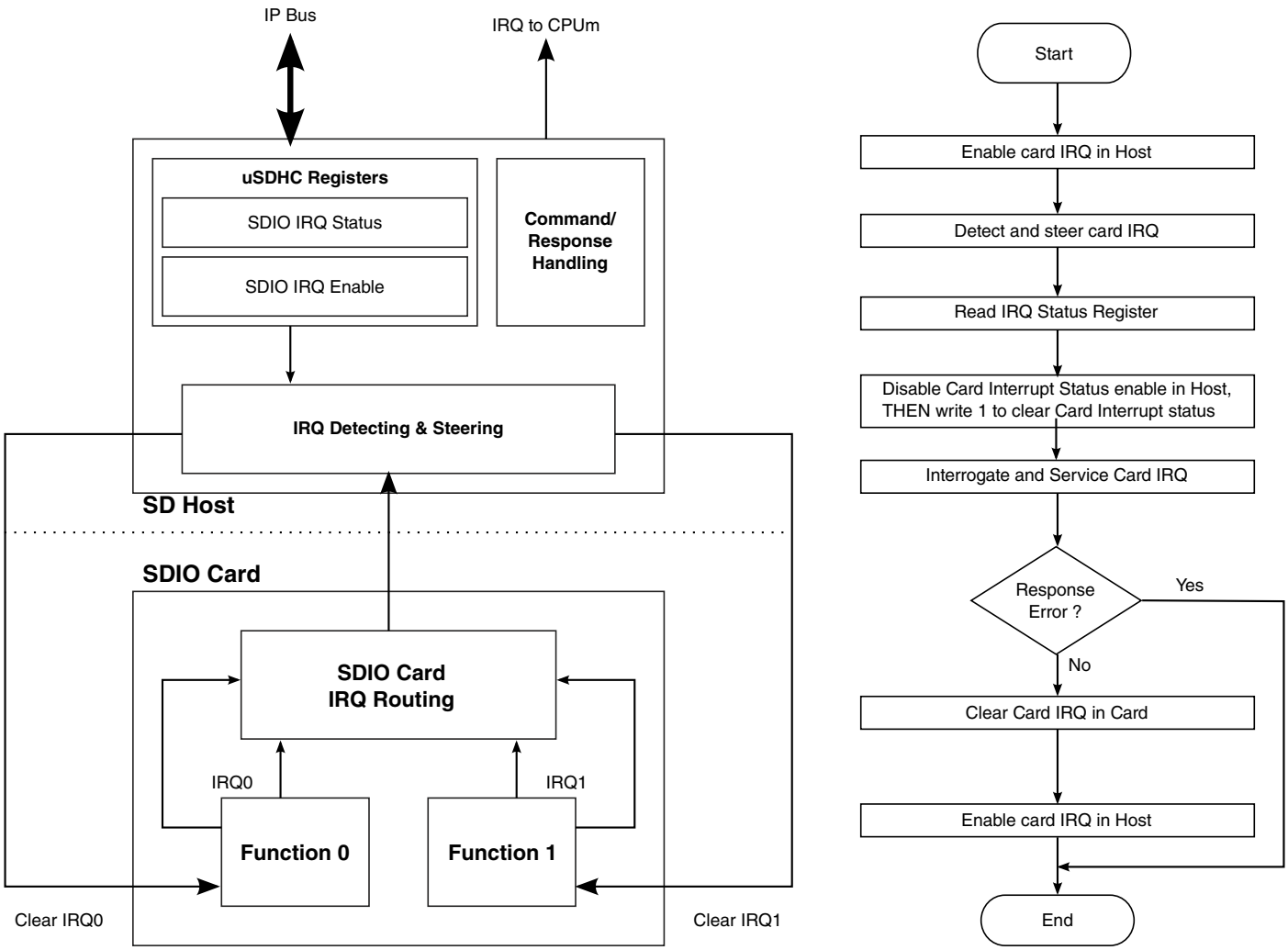


Figure 10-47. Card interrupt scheme, card interrupt detection, and handling procedure

10.3.3.7 Card insertion and removal detection

The uSDHC module uses either the DATA3 pin or the CD_B pin to detect card insertion or removal. When there is no card on the MMC/SD bus, the DATA3 is pulled to a low voltage level by default.

When any card is inserted to or removed from the socket, uSDHC detects the logic value changes on the DATA3 pin and generates an interrupt. When the DATA3 pin is not used for card detection (for example, it is implemented in GPIO), the CD_B pin must be connected for card detection. Whether DATA3 is configured for card detection or not, the CD_B pin is always a reference for card detection. Whether the DATA3 pin or the CD_B pin is used to detect card insertion, uSDHC sends an interrupt (if enabled) to inform the Host system that a card is inserted.

10.3.3.8 Power management and wakeup events

When there is no operation between uSDHC and the card through the SD bus, the user can completely disable the peripheral clock and base clock in the chip-level clock control module to save power. When the user needs to use uSDHC to communicate with the card, it can enable the clock and start the operation.

In some circumstances, when the clocks to uSDHC are disabled, for instance, when the system is in low-power mode, there are some events for which the user needs to enable the clock and handle the event. These events are called wakeup interrupts. The uSDHC module can generate these interrupts even when there are no clocks enabled. The three interrupts that can be used as wakeup events are these:

- Card removal interrupt
- Card insertion interrupt
- Interrupt from SDIO card

The uSDHC module offers a power management feature. By clearing the clock enabled bits in the System Control register, the clocks are gated in the low position to uSDHC. For maximum power saving, the user can disable all the clocks to uSDHC when there is no operation in progress.

These three wakeup events (or wakeup interrupts) can also be used to wakeup the system from low-power modes.

NOTE

To make the interrupt a wakeup event, when all the clocks to uSDHC are disabled or when the entire system is in low power mode, the corresponding wakeup enabled bit needs to be set. Refer to Protocol Control (PROT_CTRL) for more information on the uSDHC Protocol Control register.

10.3.3.8.1 Setting wakeup events

For uSDHC to respond to a wakeup event, the software must set the respective wakeup enable bit before the CPU enters the sleep mode.

Before the software disables the host clock, it should ensure that all the following conditions have been met:

- No read or write transfer is active
- Data and command lines are not active
- No interrupts are pending
- Internal data buffer is empty

10.3.3.9 MMC fast boot

The Embedded MultiMediaCard (eMMC4.3) specification adds a fast boot feature that requires hardware support. There are two types of fast boot modes in the eMMC4.3 specification: boot operation and alternative boot operation in the eMMC4.3 specification. Each type also has with-acknowledge and without-acknowledge modes.

In the boot operation mode, the master (MultiMediaCard host) can read boot data from the slave (MMC device) by keeping CMD line low after power-on, or sending CMD0 with argument + 0xFFFFFFFA (optional for slave), before issuing CMD1.

NOTE

For the eMMC4.3 card setting, please see the eMMC4.3 specification.

10.3.3.9.1 Boot operation

If the CMD line is held low for 74 clock cycles and more after power-up before the first command is issued, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after the CMD line goes low, the slave starts to send the first boot data to the master on the DATA line(s). The master must keep the CMD line low to read all of the boot data.

NOTE

For the purposes of this documentation, fast boot is called "normal fast boot mode".

If boot acknowledge is enabled, the slave has to send acknowledge pattern '010' to the master within 50ms after the CMD line goes low. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode with the CMD line high.

The boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.

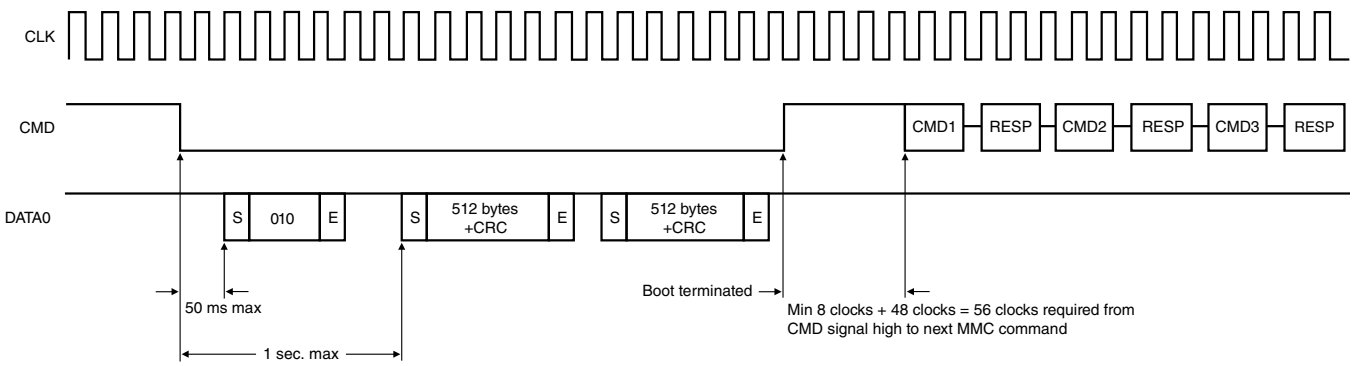


Figure 10-48. MultiMediaCard state diagram (normal boot mode)

10.3.3.9.2 Alternative boot operation

This boot function is optional for the device. If bit 0 in the extended CSD byte[228] is set to '1', the device supports the alternative boot operation.

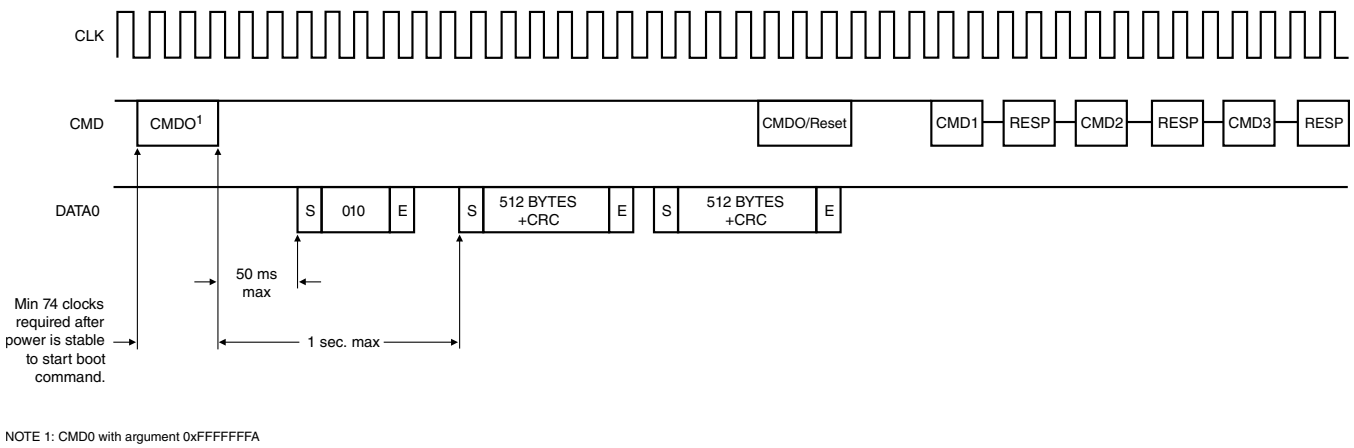
After power-up, if the host issues CMD0 with the argument of 0xFFFFFFFA after 74 clock cycles, before CMD1 is issued or the CMD line goes low, the slave recognizes that boot mode is being initiated and starts preparing boot data internally.

Within one second after CMD0 with the argument of 0xFFFFFFFA is issued, the slave starts to send the first boot data to the master on the DATA line(s).

If boot acknowledge is enabled, the slave has to send the acknowledge pattern '010' to the master within 50ms after the CMD0 with the argument of 0xFFFFFFFA is received. If boot acknowledge is disabled, the slave does not send out acknowledge pattern '010'.

The master can terminate the boot mode by issuing CMD0 (Reset).

Boot operation is terminated when all the contents of the enabled boot data are sent to the master. After the boot operation is executed, the slave gets ready for the CMD1 operation and the master needs to start a normal MMC initialization sequence by sending CMD1.



NOTE 1: CMD0 with argument 0xFFFFFFFA

Figure 10-49. MultiMediaCard state diagram (alternative boot mode)

10.3.4 Initialization/application of uSDHC

All communication between the system and cards are controlled by the host. The host sends commands of two types: broadcast and addressed (point-to-point).

Broadcast commands are intended for all cards, such as GO_IDLE_STATE, SEND_OP_COND, and ALL_SEND_CID. In the Broadcast mode, all cards are in the open-drain mode to avoid bus contention. Refer to Commands for MMC/SD/SDIO for the commands of bc and bcr categories.

After the broadcast command CMD3 is issued, the cards enter the standby mode. Addressed type commands are used from this point. In this mode, the CMD/DATA I/O pads turns to the push-pull mode to have the driving capability for maximum frequency operation. Refer to Commands for MMC/SD/SDIO for the commands of ac and adtc categories.

10.3.4.1 Command send & response receive basic operation

Assuming that the data type WORD is an unsigned 32-bit integer, the flow indicated below presents a guideline for sending a command to the card(s):

```
send_command(cmd_index, cmd_arg, other requirements)
{
WORD wCmd; // 32-bit integer to make up the data to write into Transfer Type register, it is
recommended to implement in a bit-field manner
wCmd = (<cmd_index> & 0x3f) >> 24; // set the first 8 bits as '00'+<cmd_index>
set CMDTYP, DPSEL, CICEN, CCCEN, RSTTYP, DTDSEL accorind to the command index;
if (internal DMA is used) wCmd |= 0x1;
if (multi-block transfer) {
          set MSBSEL bit;
          if (finite block number) {
                    set BCEN bit;
                    if (auto12 command is to use) set AC12EN bit;
          }
```

```
}
write_reg(CMDARG, <cmd_arg>); // configure the command argument
write_reg(XFERTYP, wCmd); // set Transfer Type register as wCmd value to issue the command
}
wait_for_response(cmd_index)
{
while (CC bit in IRQ Status register is not set); // wait until Command Complete bit is set
read IRQ Status register and check if any error bits about Command are set
if (any error bits are set) report error;
write 1 to clear CC bit and all Command Error bits;
}
```

For the sake of simplicity, the function wait_for_response is implemented here by means of polling. For an effective and formal way, the response is usually checked after the Command Complete Interrupt is received. When doing this, make sure that the corresponding interrupt status bits are enabled.

For some scenarios, the response time-out is expected. For instance, after all cards respond to CMD3 and move to the Standby state no response to the Host when CMD2 is sent. The host driver deals with "fake" errors like this with caution.

10.3.4.2 Card identification mode

When a card is inserted to the socket or the card is reset by the host, the host needs to validate the operation voltage range, identify the cards, request the cards to publish the Relative Card Address (RCA) or to set the RCA for MMC cards.

10.3.4.2.1 Card detect

See the figure below for a flow diagram showing the detection of MMC, SD, and SDIO cards using uSDHC.

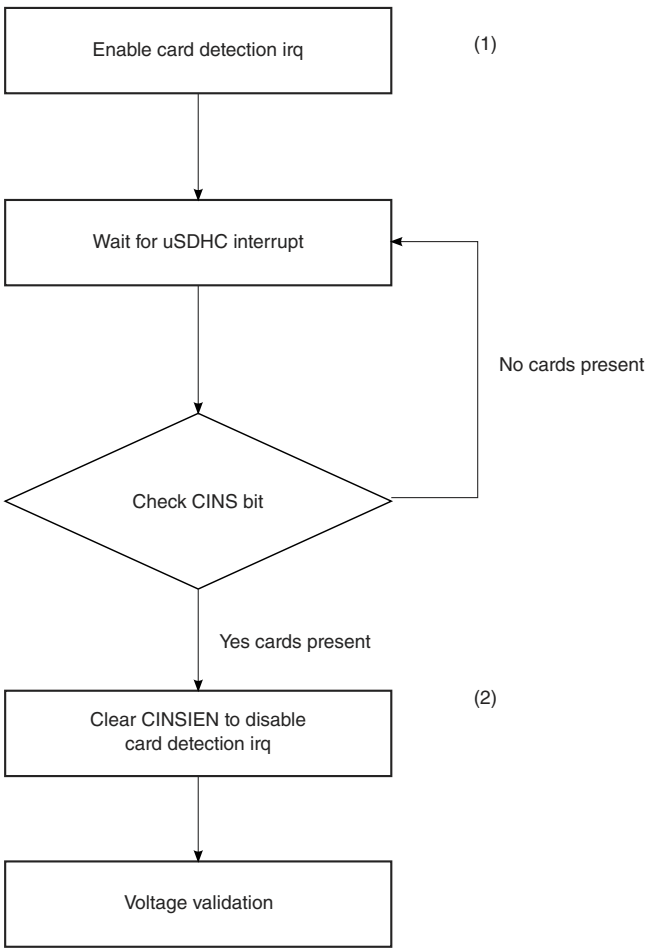


Figure 10-50. Flow diagram for card detection

Here is the card detect sequence:

- Set the CINSIEN bit to enable card detection interrupt.
- When an interrupt from uSDHC is received, check the CINS bit in the Interrupt Status register to see if it was caused by card insertion.
- Clear the CINSIEN bit to disable the card detection interrupt and ignore all card insertion interrupts afterwards.

10.3.4.2.2 Reset

The host consists of three types of resets:

- Hardware reset (Card and Host) that is driven by Power On Reset (POR).

- Software reset (Host only) is initiated by the write operation on the RSTD, RSTC, or RSTA bits of the System Control register to reset the data part, command part, or all parts of the host controller, respectively.
- Card reset (Card only): The command, "Go_Idle_State" (CMD0), is the software reset command for all types of MMC cards and SD memory cards. This command sets each card into the idle state regardless of the current card state. For an SD I/O card, CMD52 is used to write an I/O reset in CCCR. The cards are initialized with a default relative card address (RCA=0x0000) and with a default driver stage register setting (lowest speed, highest driving current capability).

After the card is reset, the host needs to validate the voltage range of the card. See the figure below for the software flow to reset both uSDHC and the card.

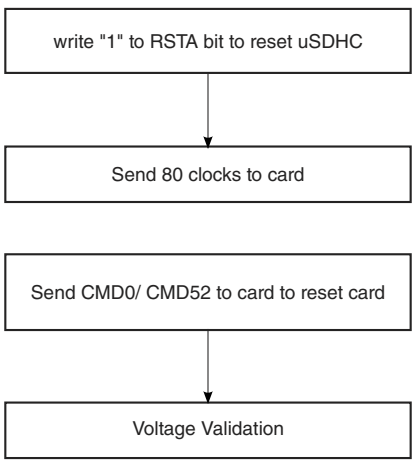


Figure 10-51. Flow chart for resetting uSDHC and SD I/O card

```
software_reset()
{
set_bit(SYSCTRL, RSTA); // software reset the Host
set DTOCV and SDCLKFS bit fields to get the CLK of frequency around 400kHz
configure IO pad to set the power voltage of external card to around 3.0V
poll bits CIHB and CDIHB bits of PRSSTAT to wait both bits are cleared
set_bit(SYSCTRL, INTIA); // send 80 clock ticks for card to power up
send_command(CMD_GO_IDLE_STATE, <other parameters>); // reset the card with CMD0
or send_command(CMD_IO_RW_DIRECT, <other parameters>);
}
```

10.3.4.2.3 Voltage validation

All cards should be able to establish communication with the host using any operation voltage in the maximum allowed voltage range specified in the card specification. However, the supported minimum and maximum values for VDD are defined in the Operation Conditions Register (OCR) and may not cover the whole range.

Cards that store the CID and CSD data in the preload memory are only able to communicate this information under data transfer VDD conditions. This means that if the host and card have non-common VDD ranges, the card is neither able to complete the identification cycle nor able to send CSD data.

Therefore, special commands Send_Op_Cont (CMD1 for MMC), SD_Send_Op_Cont (ACMD41 for SD Memory), and IO_Send_Op_Cont (CMD5 for SD I/O) are used. The voltage validation procedure is designed to provide a mechanism to identify and reject cards that do not match the VDD range(s) desired by the host. This is accomplished when the host sends the desired VDD voltage window as the operand of this command. Cards that cannot perform the data transfer in the specified range must discard themselves from further bus operations and go into the Inactive state. By omitting the voltage range in the command, the host can query each card and determine the common voltage range before sending out-of-range cards into the Inactive state. This query should be used if the host is able to select a common voltage range or if a notification is sent to the system when a non-usable card in the stack is detected.

The following steps show how to perform voltage validation when a card is inserted:

```
voltage_validation(voltage_range_arguement)
{
label the card as UNKNOWN;
send_command(IO_SEND_OP_COND, 0x0, <other parameters are omitted>); // CMD5, check SDIO
operation voltage, command argument is zero
if (RESP_TIMEOUT != wait_for_response(IO_SEND_OP_COND)) { // SDIO command is accepted
          if (0 < number of IO functions) {
                    label the card as SDIO;
                    IORDY = 0;
                    while (!(IORDY in IO OCR response)) { // set voltage range for each IO
function
                              send_command(IO_SEND_OP_COND, <voltage range>, <other
parameter>);
                              wait_for_response(IO_SEND_OP_COND);
                    } // end of while ...
          } // end of if (0 < ...
          if (memory part is present inside SDIO card) Label the card as SDCombo; // this is
an
SD-Combo card
} // end of if (RESP_TIMEOUT ...
if (the card is labelled as SDIO card) return; // card type is identified and voltage range
is
set, so exit the function;
send_command(APP_CMD, 0x0, <other parameters are omitted>); // CMD55, Application specific
CMD
prefix
if (no error calling wait_for_response(APP_CMD, <...>) { // CMD55 is accepted
          send_command(SD_APP_OP_COND, <voltage range>, <...>); // ACMD41, to set voltage
range
for memory part or SD card
          wait_for_response(SD_APP_OP_COND); // voltage range is set
          if (card type is UNKNOWN) label the card as SD;
          return; //
} // end of if (no error ...
else if (errors other than time-out occur) { // command/response pair is corrupted
          deal with it by program specific manner;
} // of else if (response time-out
else { // CMD55 is refuse, it must be MMC card  if (card is already labelled as SDCombo)
{ //
change label
```

```
                    re-label the card as SDIO;
                    ignore the error or report it;
                    return; // card is identified as SDIO card
          } // of if (card is ...
          send_command(SEND_OP_COND, <voltage range>, <...>);
          if (RESP_TIMEOUT == wait_for_response(SEND_OP_COND)) { // CMD1 is not accepted,
either
                    label the card as UNKNOWN;
                    return;
          } // of if (RESP_TIMEOUT ...
} // of else
}
```

10.3.4.2.4 Card registry

Card registry for the MMC and SD/SDIO/SD combo cards are different. For the SD card, the identification process starts at a clock rate lower than 400 kHz and the power voltage higher than 2.7 V (as defined by the card spec). Currently, the CMD line output drives are push-pull drivers instead of open-drain. After the bus is activated, the host requests the card to send their valid operation conditions.

The response to ACMD41 is the operation condition register of the card. The same command is sent to all the new cards in the system. Incompatible cards are put into the Inactive state. The host then issues the command, All_Send_CID (CMD2), to each card to get its unique card identification (CID) number. Cards that are currently unidentified (in the Ready state), send their CID number as the response. After the CID is sent by the card, the card goes into the Identification state.

The host then issues Send_Relative_Addr (CMD3), requesting the card to publish a new relative card address (RCA) that is shorter than the CID. This RCA is used to address the card for future data transfer operations. After the RCA is received, the card changes its state to the Standby state. At this point, if the host wants the card to have an alternative RCA number, it may ask the card to publish a new number by sending another Send_Relative_Addr command to the card. The last published RCA is the actual RCA of the card.

The host repeats the identification process with CMD2 and CMD3 for each card in the system until the last CMD2 gets no response from any of the cards in the system.

For MMC operation, the host starts the card identification process in the open-drain mode with the identification clock rate lower than 400 kHz and the power voltage higher than 2.7 V. The open drain driver stages on the CMD line to allow parallel card operation during card identification. After the bus is activated, the host requests the cards to send their valid operation conditions (CMD1). The response to CMD1 is the "wired OR" operation on the condition restrictions of all cards in the system. Incompatible cards are sent into the Inactive state. The host then issues the broadcast command All_Send_CID (CMD2), asking all cards for their unique card identification (CID) number. All unidentified cards (the cards in the Ready state) simultaneously start sending their CID

numbers serially, while bit-wise monitoring their outgoing bit stream. Those cards, whose outgoing CID bits do not match the corresponding bits on the command line in any one of the bit periods, stop sending their CID immediately and must wait for the next identification cycle. As the CID is unique for each card, only one card can be successfully sent its full CID to the host. This card then goes into the Identification state. Thereafter, the host issues Set_Relative_Addr (CMD3) to assign a relative card address (RCA) to the card. After the RCA is received, the state of the card changes to standby, and the card does not react in further identification cycles. Also, its output driver switches from open-drain to push-pull. The host repeats the process, mainly CMD2 and CMD3, until the host receives a time-out condition to recognize the completion of the identification process.

The following steps show how to perform an operation using MMC cards:

```
card_registry()
{
do { // decide RCA for each card until response time-out
          if(card is labelled as SDCombo or SDIO) { // for SDIO card like device
                    send_command(SET_RELATIVE_ADDR, 0x00, <...>); // ask SDIO card to
publish its
RCA
                    retrieve RCA from response;
          } // end if (card is labelled as SDCombo ...
          else if (card is labelled as SD) { // for SD card
                    send_command(ALL_SEND_CID, <...>);
                    if (RESP_TIMEOUT == wait_for_response(ALL_SEND_CID)) break;
                    send_command(SET_RELATIVE_ADDR, <...>);
                    retrieve RCA from response;
          } // else if (card is labelled as SD ...
          else if (card is labelled as MMC) {
                    send_command(ALL_SEND_CID, <...>);
                    rca = 0x1; // arbitrarily set RCA, 1 here for example
                    send_command(SET_RELATIVE_ADDR, 0x1 << 16, <...>); // send RCA at upper
16
bits
          } // end of else if (card is labelled as MMC ...
} while (response is not time-out);
}
```

10.3.4.3 Card access

Information about Block Write, Block Read, Suspense Resume, ADMA Usage, Transfer Error, and Card Interrupt are detailed in the sections below.

10.3.4.3.1 Block write

Information on Normal Write, DDR Write, and Write with Pause are detailed in the sections below.

10.3.4.3.1.1 Normal Write

During a block write (CMD24 - 27, CMD60, CMD61), one or more blocks of data are transferred from the host to the card with a CRC appended to the end of each block by the host. If the CRC fails, the card indicates that the failure on the DATA line. The transferred data is discarded and not written, and all further transmitted blocks (in multiple block write mode) are ignored.

If the host uses partial blocks whose accumulated length is not block aligned and block misalignment is not allowed (CSD parameter WRITE_BLK_MISALIGN is not set), the card detects the block misalignment error and aborts the programming before the beginning of the first misaligned block. The card sets the ADDRESS_ERROR error bit in the status register, and while ignoring all further data transfer, waits in the Receive-data-State for a stop command. The write operation is also aborted if the host tries to write over a write-protected area.

For MMC and SD cards, programming of the CID and CSD registers does not require a previous block length setting. The transferred data is also CRC protected. If a part of the CSD or CID register is stored in ROM, then this unchangeable part must match the corresponding part of the receive buffer. If this match fails, then the card reports an error and does not change any register contents.

For all types of cards, some may require long and unpredictable periods of time to write a block of data. After receiving a block of data and completing the CRC check, the card begins writing and holds the DATA line low if its write buffer is full and unable to accept new data from a new WRITE_BLOCK command. The host may poll the status of the card with a SEND_STATUS command (CMD13) or other means for SDIO cards at any time, and the card responds with its status. The responded status indicates whether the card can accept new data or whether the write process is still in progress. The host may deselect the card by issuing a CMD7 (to select a different card) to place the card into the Standby state and release the DATA line without interrupting the write operation. When re-selecting the card, it reactivates the busy indication by pulling data to low if the programming is still in progress and the write buffer is unavailable.

The software flow to write to a card incorporates the internal DMA and the write operation is a multi-block write with the Auto CMD12 enabled. For the other two methods (by means of external DMA orCPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to write to a card is described below:

1. Check the card status, wait until the card is ready for data.
2. Set the card block length/size:

- For SD/MMC cards, use SET_BLOCKLEN (CMD16).
- For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT (CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).

3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.1.2 DDR write

uSDHC supports the dual data rate mode.

The software flow to write to a card in the DDR mode is described as below:

1. Check the card status and wait until the card is ready for data.

NOTE

For eMMC4.4 card, block length only can be set to 512byte.

2. Set the uSDHC number block register (NOB), where nob is 5, for instance.
3. Set the eMMC4.4 card to high-speed mode and use SWITCH(CMD6).
4. Set the eMMC4.4 card bus with 4-bit/8-bit DDR mode and use SWITCH(CMD6).
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
6. Wait for the Transfer Complete interrupt.
7. Check the status bit to see if a write CRC error occurred or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.1.3 Write with Pause

The write operation can be paused during the transfer. Instead of stopping the CLK at any time to pause all the operations, which is also inaccessible to the host driver, the driver can set the Stop At Block Gap Request (SABGREQ) bit in the Protocol Control register to pause the transfer between the data blocks. As there is no time-out condition in a write

operation during the data blocks, a write to all types of cards can be paused in this way, and if the DATA0 line is not required to de-assert to release the busy state, no suspend command is needed.

Similar to the flow described in Normal Write, the write with pause is shown with the same kind of write operation:

1. Check the card status and wait until the card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer write ready interrupt, configure the DMA settings and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Set the SABGREQ bit.
7. Wait for the Transfer Complete interrupt.
8. Clear the SABGREQ bit.
9. Check the status bit to see if a write CRC error occurred.
10. Set the CREQ bit to continue the write operation.
11. Wait for the Transfer Complete interrupt.
12. Check the status bit to see if a write CRC error occurred or some another error that occurred during the auto12 command sending and response receiving.

The number of blocks left during the data transfer is accessible by reading the contents of the BLKCNT field in the Block Attribute register. As the data transfer and the setting of the SABGREQ bit are concurrent, and the delay of register read and the register setting, the actual number of blocks left may not be exactly the value read earlier. The host driver reads the value of BLKCNT after the transfer is paused and the Transfer Complete interrupt is received.

It is also possible that the last block has begun when the Stop At Block Gap Request is sent to the buffer. In this case, the next block gap is the end of the transfer. These types of requests are ignored, the driver should treat these as a non-pause transfer, and deal with it as a common write operation.

When the write operation is paused, the data transfer inside the host system is not stopped, and the transfer is active until the data buffer is full. Because of this, it is recommended to avoid using the Suspend command for the SDIO card. This is because when such a command is sent, uSDHC interprets the system that switches to another

function on the SDIO card and flush the data buffer. uSDHC takes the Resume command as a normal command with data transfer, and it is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set as well as the AC12EN bit. However, the uSDHC module automatically sends a CMD12 to mark the end of the multi-block transfer.

10.3.4.3.2 Block read

Information about Normal read, DDR read, Read with Pause, and Delay Line (DLL) in Read Path are detailed in the sections below.

10.3.4.3.2.1 Normal read

For block reads, the basic unit of data transfer is a block whose maximum size is stored in areas defined by the corresponding card specification. A CRC is appended to the end of each block, ensuring data transfer integrity. The CMD17, CMD18, CMD53, CMD60, CMD61, and so on, can initiate a block read. After completing the transfer, the card returns to the Transfer state. For multi blocks read, data blocks are continuously transferred until a stop command is issued.

The software flow to read from a card incorporates the internal DMA and the read operation is a multi-block read with the Auto CMD12 enabled. For the other two methods (by means of external DMA orCPU polling status) with different transfer methods, the internal DMA parts should be removed and the alternative steps should be straightforward.

The software flow to read from a card is described below:

1. Check the card status and wait until card is ready for data.
2. Set the card block length/size:
   - For SD/MMC, use SET_BLOCKLEN (CMD16).
   - For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).
3. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
4. Set the uSDHC number block register (NOB), where nob is 5, for instance.
5. Disable the buffer read ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
6. Wait for the Transfer Complete interrupt.

7. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

10.3.4.3.2.2 DDR read

The uSDHC module supports dual data rate mode.

The software flow to write to a card in the DDR mode is described below:

1. Check the card status ans wait until the card is ready for data.
2. For eMMC4.4 card, block length can be set to only 512 bytes.
3. Set the uSDHC number block register (NOB) where nob is 5, for instance.
4. Set the eMMC4.4 card to high-speed mode and use SWITCH (CMD6).
5. Set the eMMC4.4 card bus with 4-bit /8-bit DDR mode and use SWITCH(CMD6).
6. Disable the buffer write ready interrupt, configure the DMA settings, and enable the uSDHC DMA when sending the command with data transfer. The DDR_EN and AC12EN bits should be set.
7. Wait for the Transfer Complete interrupt.
8. Check the status bit to see if a write CRC error occurred, or another error that occurred during the auto12 command sending and response receiving.

10.3.4.3.2.3 Read with Pause

The read operation is not generally able to pause. Only the SDIO card (and SDCombo card working under I/O mode) supporting the Read Wait feature can pause during the read operation. If the SDIO card supports Read Wait (SRW bit in CCCR register is 1), the host driver can set the SABGREQ bit in the Protocol Control register to pause the transfer between the data blocks.

Before setting the SABGREQ bit, ensure that the RWCTL bit in the Protocol Control register is set, otherwise uSDHC does not assert the Read Wait signal during the block gap and data corruption occurs. It is recommended to set the RWCTL bit after the Read Wait capability of the SDIO card is recognized.

Similar to the flow described in Normal read, the read with pause is shown with the same kind of read operation:

1. Check the SRW bit in the CCR register on the SDIO card to confirm that the card supports the Read Wait mode.
2. Set the RWCTL bit.
3. Check the card status and wait until the card is ready for data.
4. Set the card block length/size:

- For SD/MMC, use SET_BLOCKLEN (CMD16).
- For SDIO cards or the I/O portion of SDCombo cards, use IO_RW_DIRECT(CMD52) to set the I/O Block Size bit field in the CCCR register (for function 0) or FBR register (for functions 1~7).

5. Set the uSDHC block length register to be the same as the block length set for the card in step 2.
6. Set the uSDHC number block register (NOB), where nob is 5, for instance.
7. Disable the buffer read ready interrupt, configure the DMA setting, and enable the uSDHC DMA when sending the command with data transfer. The AC12EN bit should also be set.
8. Set the SABGREQ bit.
9. Wait for the Transfer Complete interrupt.
10. Clear the SABGREQ bit.
11. Check the status bit to see if read CRC error occurred.
12. Set the CREQ bit to continue the read operation.
13. Wait for the Transfer Complete interrupt.
14. Check the status bit to see if a read CRC error occurred, or another error occurred during the auto12 command sending and response receiving.

Similar to the Write operation, it is possible to meet the ending block of the transfer when paused. In this case, uSDHC ignores the Stop At Block Gap Request and treats it as a command read operation.

Unlike the write operation, there is no remaining data inside the buffer when the transfer is paused. All data received before the pause is transferred to the host system. No matter if the Suspend command is sent or not, the internal data buffer is not flushed.

If the Suspend command is sent and the transfer is later resumed by means of a Resume command, uSDHC takes the command as a normal one accompanied with data transfer. It is left for the host driver to set all the relevant registers before the transfer is resumed. If there is only one block to send when the transfer is resumed, the MSBSEL and BCEN bits of the Transfer Type register are set, as well as the AC12EN bit. However, the uSDHC automatically sends CMD12 to mark the end of multi-block transfer.

10.3.4.3.2.4 Delay Line (DLL) in Read Path

The DLL is newly added to assist in sampling read data. The DLL provides the ability to programmatically select a quantized delay (in fractions of the clock period) regardless of on-chip variations such as process, voltage, and temperature (PVT).

The reasons why DLL is needed for uSDHC are these:

- The path of read data traveling from card to host varies.
- In the SD/MMC DDR mode, the minimum input setup and hold time are both at 2.5 ns.

The data sampling window is so small that the delay of loopback clock needs to be accurate and consistent regardless of PVT. The DLL takes the divided card_clk as the reference clock and loopback clock as the input clock. It then generates a delayed version of the input clock according to the programmed target delay.

The DLL can be disabled or bypassed, and it can also be manually set for a fixed delay in the override mode. The override value set is the number of delay cells. In the override mode, there is no need to set the DLL_enable. Another DLL mode is target value mode. In this mode, the DLL automatically adjusts the number of delay cells according to the target value set by the user and PVT changes. Be aware that the target value is in units of 1/32 of the clock reference period. If the card_clk is 100Mhz, then the reference clock period is 10ns; setting target value of 16 means 5ns =(16/32)*10ns. The software can disable automatic update by the setting dll_gate_update bit.

As the user may change the frequency of card_clk from time to time by changing SDCLKFS[7:0]/DVS[3:0], the software must adjust the delay value to ensure it works correctly when the reference clock (card_clk) is changed. The following is the correct flow, which should be ignored if DLL_CTRL_ENABLE is not set.

Step 1: Set the DLL_CTRL_RESET bit

Step 2: Configure the SDCLKFS[7:0] and DVS[3:0]

Step 3: Wait until SDSTB is asserted

Step 4: Clear the DLL_CTRL_RESET bit

Step 5: Wait until both the DLL_STS_SLV_LOCK and DLL_STS_REF_LOCK are asserted

Step 6: Set the DLL_CTRL_SLV_FORCE_UPD

Step 7: Clear the DLL_CTRL_SLV_FORCE_UPD

NOTE

The software should make sure that the DLL_CTRL_SLV_FORCE_UPD lasts for at least one card_clk. So, the software may need to add some delay between step 6 and step 7.

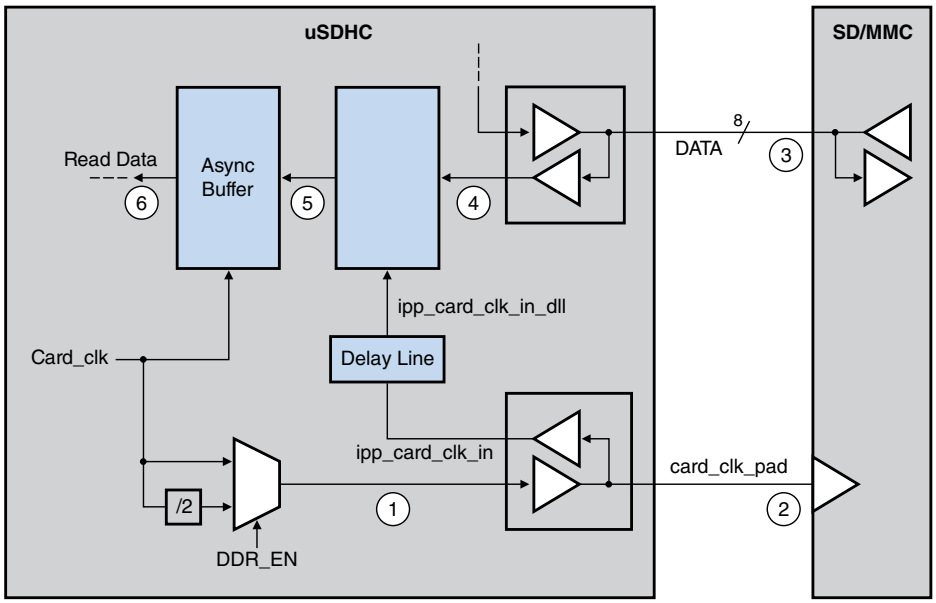


Figure 10-52. DLL in read path

10.3.4.3.3 Suspend Resume

The uSDHC module supports the Suspend Resume operations of SDIO cards, although slightly different than the suggested implementation of Suspend in the SDIO card specification.

10.3.4.3.3.1 Suspend

After setting the SABGREQ bit, the host driver may send a Suspend command to switch to another function of the SDIO card. The uSDHC module does not monitor the content of the response, so it does not know if the Suspend command succeeded or not. Accordingly, it does not de-assert Read Wait for read pause. To solve this problem, the host driver does not mark the Suspend command as "Suspend", (that is, setting the CMDTYP bits to 01). Instead, the driver sends this command as if it were a normal command, and only when the command succeeds, and the BS bit is set in the response, can the Driver send another command marked as "Suspend" to inform uSDHC that the current transfer is suspended. Here is the sequence for the Suspend operation:

1. Set the SABREQ bit to pause the current data transfer at block gap.
2. After the BGE bit is set, send the Suspend command to suspend the active function. The CMDTYP bit field must be 2'b00.
3. Check the BS bit of the CCCR in the response. If it is 1, repeat this step until the BS bit is cleared or abandon the suspend operation according to the Driver strategy.

4. Send another normal I/O command to the suspended function. The CMDTYP of this command must be 2'b01, so uSDHC can detect this special setting and be informed that the paused operation has successfully suspended. If the paused transfer is a read operation, uSDHC stops driving DATA2 and goes to the idle state.
5. Save the context registers in the system memory for later use, including the DMA System Address Register (for internal DMA operation), and the Block Attribute Register.
6. Begin operation for another function on the SDIO card.

10.3.4.3.3.2 Resume

To resume the data transfer, a Resume command is issued:

1. To resume the suspended function, restore the context register with the saved value in step #5 of the Suspend operation.
2. Send the Resume command: In the Transfer Type register, all the bit fields are set to the value as if this were another ordinary data transfer instead of a transfer resume (except the CMDTYP is set to 2'b10).
3. If the Resume command has responded, the data transfer is resumed.

10.3.4.3.4 ADMA usage

To use the ADMA in a data transfer, the host driver must prepare the correct descriptor chain prior to sending the read/write command.

The steps to prepare the correct descriptor chain are these:

1. Create a descriptor to set the data length that the current descriptor group is about to transfer. The data length should be even numbers of the block size.
2. Create another descriptor to transfer the data from the address setting in this descriptor. The data address must be at a page boundary (4KB address aligned).
3. If necessary, create a Link descriptor containing the address of the next descriptor. The descriptor group is created in steps 1 ~ 3.
4. Repeat steps 1 ~ 3 until all descriptors are created.
5. In the last descriptor, set the End flag to 1 and make sure the total length of all descriptors match the product of the block size and block number configured in the Block Attribute Register.
6. Set the ADMA System Address Register to the address of the first descriptor and set the DMAS field in the Protocol Control Register to 01 to select the ADMA.
7. Issue a write or read command with the DMAEN bit set to 1 in the Transfer Type Register.

Steps 1 ~ 5 are independent of step 6, so step 6 can finish before steps 1 ~ 5. Regarding the descriptor configuration, it is recommended not to use the Link descriptor as it requires extra system memory access.

10.3.4.3.5 Transfer error

Information about CRC, Internal DMA, Transfer ADMA, and Auto CMD12 errors are detailed in the sections below.

10.3.4.3.5.1 CRC error

It is possible at the end of a block transfer that a write CRC status error or read CRC error occurs. For this type of error, the latest block received is discarded. This is because the integrity of the data block is not guaranteed. It is recommended to discard the following data blocks and re-transfer the block from the corrupted one.

For a multi-block transfer, the host driver issues a CMD12 to abort the current process and start the transfer by a new data command. In this scenario, even when the AC12EN and BCEND bits are set, uSDHC does not automatically send a CMD12 because the last block is not transferred. On the other hand, if it is within the last block that the CRC error occurs, an Auto CMD12 is sent by uSDHC. In this case, the host driver re-sends or re-obtains the last block with a single block transfer.

10.3.4.3.5.2 Internal DMA error

During the data transfer with internal Simple DMA, if the DMA engine encounters an error on the AHB bus, the DMA operation is aborted, and the DMA error interrupt is sent to the host system. When acknowledged by such an interrupt, the host driver calculates the start address of data block in which the error occurs.

The start address can be calculated by either:

- Reading the DMA System Address register: The error occurs during the previous burst. Considering the block size, the previous burst length and the start address of the next burst transfer, it is straight forward to obtain the start address of the corrupted block.
- Reading the BLKCNT field of the Block Attribute register: Considering the number of blocks left, the total number to transfer, the start address of transfer, and the size of each block, the start address of corrupted block can be determined. When the BCEN bit is not set, the contents of the Block Attribute register do not change, so this method does not work.

When a DMA error occurs, it is recommended to abort the current transfer by means of a CMD12 (for multi block transfer), apply a reset for data, and restart the transfer from the corrupted block to recover from the error.

10.3.4.3.5.3 Transfer ADMA error

There are three kinds of possible ADMA errors: The AHB transfer, invalid descriptor, and data-length mismatch. Whenever these errors occur, the DMA transfer stops and the corresponding error status bit is set.

For acknowledging the status, the host driver should recover the error as shown below and re-transfer from the place of interruption.

- AHB transfer error: Such errors may occur during data transfer or descriptor fetch. For either scenario, it is recommended to retrieve the transfer context, reset for the data part and re-transfer the block that was corrupted, or to the next block if no block is corrupted.
- Invalid descriptor error: For such errors, it is recommended to retrieve the transfer context, reset for the data part and recreate the descriptor chain from the invalid descriptor and issue a new transfer. As the data to transfer now may be less than the previous setting, the data length configured in the new descriptor chain should match the new value.
- Data-length mismatch error: It is similar to recover from this error. The Host Driver polls relating registers to retrieve the transfer context, apply a reset for the data part, configure a new descriptor chain and make another transfer if there is data left. Like the previous scenario of the invalid descriptor error, the data length must match the new transfer.

10.3.4.3.5.4 Auto CMD12 error

After the last block of the multi-block transfer is sent or received, and the AC12EN bit is set when the data transfer is initiated by the data command, uSDHC automatically sends a CMD12 to the card to stop the transfer.

When errors with this command occur, it is recommended to the host driver to deal with the situations in the following manner:

- Auto CMD12 response time-out: It is not certain whether the command is accepted by the card or not. The host driver clears the auto CMD12 error status bits and re-send CMD12 until it is accepted by the card.

- Auto CMD12 response CRC error: As the card responds to CMD12, it aborts the transfer. The host driver may ignore the error and clear the error status bit.
- Auto CMD12 conflict error or not sent: The command is not sent; therefore, the host driver sends a CMD12 manually.

10.3.4.3.6 Card interrupt

The external cards can inform the host controller by means of some special signals. For the SDIO card, it can be the low-level on the DATA1 line during some special period. The uSDHC module only monitors the DATA1 line and supports the SDIO interrupt.

When the SDIO interrupt is captured by uSDHC, and the host system is informed by uSDHC asserting the uSDHC interrupt line, the interrupt service from the host driver is called.

As the interrupt source is controlled by the external card, the interrupt from the SDIO card must be serviced before the CINT bit is cleared by written. Refer to Card interrupt handling for the card interrupt handling flow.

10.3.4.4 Switch function

A switch command is issued by the host driver to enable new features added to the SD/MMC spec. The SD/MMC cards can transfer data at bus widths other than 1-bit. Different speed modes are also defined. To enable these features, a switch command is issued by the host driver.

For SDIO cards, the high-speed mode/DDR50/SDR50/SDR104 are enabled by writing the EHS bit in the CCCR register after the SHS bit is confirmed. For SD cards, the high-speed mode/DDR50/SDR50/SDR104 are queried and enabled by a CMD6 (with the mnemonic symbol as SWITCH_FUNC). For MMC cards, the high-speed mode/HS200/HS400 are queried by a CMD8 and enabled by a CMD6 (with the mnemonic symbol as SWITCH).

The SDR4-bit, SDR8-bit, DDR4-bit, and DDR8-bit width of MMC is also enabled by the SWITCH command, but with a different argument.

These new functions can also be disabled by a software reset. For SDIO cards, it can be done by setting the RES bit in the CCCR register. For other cards, it can be accomplished by issuing a CMD0. This method of restoring to the normal mode is not recommended because a complete identification process is needed before the card is ready for data transfer.

For the sake of simplicity, the following pseudocode examples do not show current capability check, which is recommended in the function switch process.

10.3.4.4.1 Query, enable, and disable SDIO high-speed mode

```
enable_sdio_high_speed_mode(void)
{
send CMD52 to query bit SHS at address 0x13;
if (SHS bit is '0') report the SDIO card does not support high speed mode and return;
send CMD52 to set bit EHS at address 0x13 and read after write to confirm EHS bit is set;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz;
(data transactions like normal peers)
}
disable_sdio_high_speed_mode(void)
{
send CMD52 to clear bit EHS at address 0x13 and read after write to confirm EHS bit is
cleared;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

10.3.4.4.2 Query, enable, and disable SD high-speed mode/DDR50/SDR50/ SDR104

```
enable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0xFFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104,x=3; DDR50,x=4;)
wait data transfer done bit is set;
check if the bit x of received 512 bits is set;
if (bit 401 is '0') report the SD card does not support high speed mode and return;
if (bit 402 is '0') report the SD card does not support SDR50 mode and return;
if (bit 403 is '0') report the SD card does not support SDR104 mode and return;
if (bit 404 is '0') report the SD card does not support DDR50 mode and return;
send CMD6, with argument 0x80FFFFFx and read 64 bytes of data accompanying the R1 response;
(high speed mode,x=1; SDR50,x=2; SDR104 x=3; DDR50 x=4;)
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 50MHz for high speed mode, 100MHz for SDR50, 200MHz for SDR104, 50MHz for
DDR50;
(data transactions like normal peers)
}
disable_sd_speed_mode(void)
{
set BLKCNT field to 1 (block), set BLKSIZE field to 64 (bytes);
send CMD6, with argument 0x80FFFFF0 and read 64 bytes of data accompanying the R1 response;
check if the bit field 379~376 is 0xF;
if (the bit field is 0xF) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 25MHz;
(data transactions like normal peers)
}
```

10.3.4.4.3 Query, enable, and disable MMC high-speed mode

```
enable_mmc_high_speed_mode(void)
{
send CMD9 to get CSD value of MMC;
```

```
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support high speed mode and
return;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
extract the value of CARD_TYPE field to check the 'high speed mode' in this MMC is 26MHz or
52MHz;
send CMD6 with argument 0x1B90100;
send CMD13 to wait card ready (busy line released);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 1;
if (HS_TIMING is not 1) report MMC switching to high speed mode failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of around 26MHz or 52MHz according to the CARD_TYPE;
(data transactions like normal peers)
}
disable_mmc_high_speed_mode(void)
{
send CMD6 with argument 0x2B90100;
set BLKCNT field to 1 (block), set BLKSIZE field to 512 (bytes);
send CMD8 to get EXT_CSD value of MMC;
check if HS_TIMING byte (byte number 185) is 0;
if (HS_TIMING is not 0) report the function switch failed and return;
change clock divisor value or configure the system clock feeding into uSDHC to generate the
card_clk of the desired value below 20MHz;
(data transactions like normal peers)
}
```

10.3.4.4.4 Set MMC bus width

```
change_mmc_bus_width(void)
{
send CMD9 to get CSD value of MMC;
check if the value of SPEC_VER field is 4 or above;
if (SPEC_VER value is less than 4) report the MMC does not support multiple bit width and
return;
send CMD6 with argument 0x3B70x00; (8-bit(dual data rate), x=6; 4-bit(dual data rate), x=5;8-
bit, x=2; 4-bit, x=1; 1-bit, x=0)
send CMD13 to wait card ready (busy line released);
(data transactions like normal peers)
}
```

10.3.4.5 ADMA operation

Here are the codes for the ADMA1 and ADMA2 operations.

10.3.4.5.1 ADMA1 operation

```
Set_adma1_descriptor
{
if (to start data transfer) {
// Make sure the address is 4KB align.
Set 'Set' type descriptor;
{
Set Act bits to 01;
Set [31:12] bits data length (byte unit);
}
Set 'Tran' type descriptor;
{
Set Act bits to 10;
Set [31:12] bits address (4KB align);
}
```

```
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to 11;
Set [31:12] bits the next descriptor address (4KB aligned);
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set End bit to 1;
}
if (to generate interrupt for this descriptor) {
Set Int bit to 1;
}
Set Valid bit to 1;
}
```

10.3.4.5.2 ADMA2 operation

```
Set_adma2_descriptor
{
if (to start data transfer) {
// Make sure the address is a 32-bit boundary (lower 2-bit are always '00').
Set higher 32-bit of descriptor for this data transfer initial address;
Set [31:16] bits data length (byte unit);
Set Act bits to '10';
}
else if (to fetch descriptor at non-continuous address) {
Set Act bits to '11';
// Make sure the address is 32-bit boundary (lower 2-bit are always set to '00').
Set higher 32-bit of descriptor for the next descriptor address;
}
else { // other types of descriptor
Set Act bits accordingly
}
if (this descriptor is the last one) {
Set 'End' bit '1';
}
if (to generate interrupt for this descriptor) {
Set 'Int' bit '1';
}
Set the 'Valid' bit to '1';
}
```

10.3.4.6 Fast boot operation

10.3.4.6.1 Normal fast boot flow

Here are the steps of normal fast boot flow:

1. Software must configure the init_active bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. Software must configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 0 (normal fast boot), and bit 4 to select the ack mode. If the data is sent through the DMA mode, the software should configure bit 7 to enable the automatic stop at block gap feature, and configure bit 3-bit 0 to select the ack timeout value according to the SD CLK frequency.

3. Software then needs to configure the Block Attributes Register to set the block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes.
4. Software must configure the Protocol control register to set Data Transfer Width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure the Command Argument Register to set argument if needed (no need in normal fast boot).
6. Software must configure the Transfer Type Register to start the boot process. In normal boot mode, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN and DMAEN retain the default value, where DPSEL bit is set to 1, DTDSEL is set to 1 and MSBSEL is set to 1.
7. DMAEN should be configured as 0 in the polling mode and if BCEN is configured as 1, it is recommended to configure the number of blocks in the Block Attributes Register to the maximum value. If in DDR fast boot mode, DDR_EN needs to be set to 1.
8. When step 6 is configured, the boot process begins. The software needs to poll the data buffer ready status to read the data from the buffer in time. If a boot timeout happens (ack times out or the first data read times out), an interrupt is triggered, and the software must configure MMC Boot Register to bit 6 to 0 to disable boot. This makes CMD high, then after at least 56 clocks, it is ready to begin a normal initialization process.
9. If there is no timeout, software needs to determine when the data read is finished and then configure MMC Boot Register bit 6 to 0 to disable boot. This render CMD line high and command completed asserted. After at least 56 clocks, it is ready to begin the normal initialization process.
10. Reset the host and then can begin the normal process.

10.3.4.6.2 Alternative fast boot flow

Here are the steps of alternative fast boot flow:

1. Software needs to configure init_active bit (system control register bit 27) to make sure 74 card clocks are finished.
2. Software needs to configure MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot), and bit 5 to 1 (alternative boot), and bit 4 to select the ack mode or not. If data needs to be sent through the DMA mode, then configure bit 7 to enable the automatic stop at block gap feature. Software should also configure bit 3-bit 0 to select the ack timeout value according to the SD clock frequency.
3. Software then needs to configure the Block Attributes Register to set the block size and count. If in the DDR fast boot mode, the block size only can be configured to 512 bytes.

4. Software needs to configure the Protocol control register to set the data transfer width (DTW). If in the DDR fast boot mode, DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software needs to configure Command Argument Register to set argument to 0xFFFFFFFA.
6. Software needs to configure the Transfer Type Register to start the boot process by CMD0 with the 0xFFFFFFFA argument. In alternative boot, CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. Note DMAEN should be configured as 0 in the polling mode, and if BCEN is configured as 1 in polling mode, it is recommended to configure the block count in the Block Attributes Register to the maximum value. If in the DDR fast boot mode, DDR_EN needs to be set to 1.
7. When step 6 is configured, the boot process begins. Software needs to poll the data buffer ready status to read the data from the buffer in time. If there is a boot timeout (ack data timeout in 50ms or data timeout in 1s), the host sends out the interrupt and the software needs to send CMD0 with reset and then configure the boot enable bit to 0 to stop this process.
8. If there is no time out, the software needs to decide when to stop the boot process, and send out the CMD0 with reset and then after the command is completed, configure the MMC Boot Register bit 6 to stop the process. After 8 clocks from the command completion, the slave (card) is ready for the identification step.
9. Reset the host and then begin the normal process.

10.3.4.6.3 Fast boot application case (in DMA mode)

In the boot application case, because the image destination and the image size are contained in the beginning of the image, it is necessary to switch DMA parameters on the fly during MMC fast boot.

In fast boot, the host can use Advanced DMA2 (ADMA2) with two destinations.

The detailed flow is described below:

1. The software needs to configure INIT_ACTIVE bit (system control register bit 27) to make sure that 74 card clocks are finished.
2. The software needs to configure the MMC Boot Register (offset 0xc4) bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot) or 1 (alternative boot); and bit 4 to select the ack mode. In DMA mode, configure bit 7 to 1 to enable the automatic stop at block gap feature. Also configure bits[31-16] to set the (BLK_CNT - VALUE1). Here VALUE1 is the value of the block count that needs to transfer the first time, so that the host stops at the block gap when the uSDHC controller gets VAULE1 blocks

from the device. Also, configure bits[3-0] to select the ack timeout value according to the SD clock frequency.

3. The software then needs to configure the Block Attributes Register to set block size and count. If in DDR fast boot mode, the block size only can be configured to 512 bytes. In DMA mode, it is recommended to set the block count (BLK_CNT) to the max value (16'hffff).
4. The software needs to configure Protocol Control Register to set DTW (data transfer width). If in DDR fast boot mode, the DTW only can be configured to 4-bit/8-bit dataline mode.
5. Software enable ADMA2 by configuring Protocol Control Register bits [9-8].
6. The software needs to set at least three pairs of ADMA2 descriptor in boot memory (that is, in IRAM, at least six words). The first pair descriptor defines the start address (that is, IRAM) and data length (that is,512byte*VALUE1) of the first part boot code. The software also needs to set the second pair descriptor, the second start address (any value that is writable), and data length is suggested to set 1~2word (record as VALUE2). Note that the second couple desc also transfers useful data even at lease 1 word, because our ADMA2 cannot support 0 data_length data transfer descriptor.
7. The software needs to configure Command Argument Register to set argument to 0xFFFFFFFA in alternative fast boot and do not need to be set in normal fast boot.
8. The software needs to configure Transfer Type Register to start the boot process. CMDINX, CMDTYP, RSPTYP, CICEN, CCCEN, AC12EN, BCEN, and DMAEN retain the default value. DPSEL bit is set to 1, DTDSEL is set to 1, and MSBSEL is set to 1. DMAEN is configured as 1 in the DMA mode. And, if BCEN is configured as 1, then configure blk no in Bock Attributes Register to the max value. And, if in the DDR fast boot mode, DDR_EN needs to be set to 1.
9. When step 8 is configured, boot process begins, the first VALUE1 block number data gets transferred. The software needs to poll the TC bit (bit1 in Interrupt Status Register) to determine first transfer is ended. Also, the software needs to polling the BGE bit (bit2 in Interrupt Status Register) to determine if the first transfer stops at the block gap.
10. When TC and BGE bits are set to 1, the software can analyze the first code of VALUE1 block, initializes the new memory device, if required, and sets the third pair of descriptors to define the start address and length of the remaining part of the boot code (VALUE3, the remain boot code block). Remember to set the last descriptor with END.
11. The software needs to configure the MMC Boot Register (offset 0xc4) again. Set bit 6 to 1 (enable boot); and bit 5 to 0 (normal fast boot), to 1 (alternative boot); and bit 4 to select the ack mode or not. In the DMA mode, configure bit 7 to 1 for enabling the automatically stop at block gap feature. Also, configure bit31-bit16 to set the (BLK_CNT - (VALUE1+1+VALUE3)), that host stops at block gap when the

uSDHC controller gets (VALUE1+1+VALUE3)) blocks from device totally include the blocks received in step 9. And need to configure bit 3-bit0 to select the ack timeout value according to the sd clk frequency. Note that the software does not need to configure the BLK_CNT again, because it is counted down automatically by the uSDHC controller.

12. The software needs to clear the TC and BGE bits and the software needs to clear SABGREQ (bit 16 in the Protocol control register) and set CREQ (bit17 in the Protocol control register) to 1 to resume the data transfer. Host transfers the VALUE2 and VALUE3 data to the destination that is set by descriptor.
13. The software needs to do poll BGE bit to determine if the fast boot is over.

Note:

1. When ADMA boot flow starts, for uSDHC, it is like a normal ADMA read operation. So, set ADMA2 descriptor as the normal ADMA2 transfer.
2. Need a few words length memory to keep descriptor.
3. For the 1~2-word data in second descriptor setting, it is a useful data, so the software needs to deal the data because of the application case.

10.3.5 Commands for MMC/SD/SDIO

A table containing the list of commands for the MMC/SD/SDIO cards is provided here.

Refer to the corresponding specifications for more details about the command information.

There are four kinds of commands defined to control the MultiMediaCard:

1. Broadcast commands (bc), no response
2. Broadcast commands with response (bcr), response from all cards simultaneously
3. Addressed (point-to-point) commands (ac), no data transfer on the DATA
4. Addressed (point-to-point) data transfer commands (adtc)

Response: A response is a token that is sent from the card to the host as an answer to a previously received command. A response is transferred serially on the CMD line.

Table 10-39. Commands for MMC/SD/SDIO cards

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD0	bc	[31:0] stuff bits	-	GO_IDLE_STATE	Resets all MMC and SD memory cards to idle state.

Table continues on the next page...

Table 10-39. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD1	bcr	[31:0] OCR without busy	R3	SEND_OP_COND	Asks all MMC and SD Memory cards in idle state to send their operation conditions register contents in the response on the CMD line.
CMD2	bcr	[31:0] stuff bits	R2	ALL_SEND_CID	Asks all cards to send their CID numbers on the CMD line.
CMD3[1]	ac	[31:6] RCA [15:0] stuff bits	R1 R6 (SDIO)	SET/ SEND_RELATIVE_ADDR	Assigns relative address to the card.
CMD4	bc	[31:0] DSR [15:0] stuff bits	-	SET_DSR	Programs the DSR of all cards.
CMD5	bc	[31:0] OCR without busy	R4	IO_SEND_OP_COND	Asks all SDIO cards in idle state to send them operation conditions register contents in the response on the CMD line.
CMD6[2]	adtc	[31] Mode 0: Check function 1: Switch function [30:8] Reserved for function groups 6 ~ 3 (All 0 or 0xFFFF) [7:4] Function group1 for command system [3:0] Function group2 for access mode	R1	SWITCH_FUNC	Checks switch ability (mode 0) and switch card function (mode 1). Refer to "SD Physical Specification V1.1" for more details.
CMD6[3]	ac	[31:26] Set to 0 [25:24] Access [23:16] Index [15:8] Value [7:3] Set to 0 [2:0] Cmd Set	R1b	SWITCH	Switches the mode of operation of the selected card or modifies the EXT_CSD registers. Refer to "The MultiMediaCard System Specification Version 4.0 Final draft 2" for more details.
CMD7	ac	[31:6] RCA [15:0] stuff bits	R1b	SELECT/ DESELECT_CARD	Toggles a card between the stand-by and transfer states or between the programming and disconnect states. In both cases, the card is selected by its own relative address and gets deselected by any other address. Address 0 deselects all.
CMD8	adtc	[31:0] stuff bits	R1	SEND_EXT_CSD	The card sends its EXT_CSD register as a block of data, with a block size of 512 bytes.
CMD9	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CSD	Addressed card sends its card-specific data (CSD) on the CMD line.

Table continues on the next page...

Table 10-39. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD10	ac	[31:6] RCA [15:0] stuff bits	R2	SEND_CID	Addressed card sends its card-identification (CID) on the CMD line.
CMD11	adtc	[31:0] data address	R1	READ_DAT_UNTIL_STOP	Reads data stream from the card, starting at the given address, until a STOP_TRANSMISSION follows.
CMD12	ac	[31:0] stuff bits	R1b	STOP_TRANSMISSION	Forces the card to stop transmission.
CMD13	ac	[31:6] RCA [15:0] stuff bits	R1	SEND_STATUS	Addressed card sends its status register.
CMD14	Reserved				
CMD15	ac	[31:6] RCA [15:0] stuff bits	-	GO_INACTIVE_STATE	Sets the card to inactive state in order to protect the card stack against communication breakdowns.
CMD16	ac	[31:0] block length	R1	SET_BLOCKLEN	Sets the block length (in bytes) for all following block commands (read and write). Default block length is specified in the CSD.
CMD17	adtc	[31:0] data address	R1	READ_SINGLE_BLOCK	Reads a block of the size selected by the SET_BLOCKLEN command.
CMD18	adtc	[31:0] data address	R1	READ_MULTIPLE_BLOCK	Continuously transfers data blocks from card to host until interrupted by a stop command.
CMD19	adtc	[31:0] reserved bits(all 0)	R1	SEND_TUNING_BLOCK	64 bytes tuning pattern is sent for SDR50 and SDR104.
CMD20	adtc	[31:0] data address	R1	WRITE_DAT_UNTIL_STOP	Writes data stream from the host, starting at the given address, until a STOP_TRANSMISION follows.
CMD21	Reserved				
CMD22-23	Reserved				
CMD24	adtc	[31:0] data address	R1	WRITE_BLOCK	Writes a block of the size selected by the SET_BLOCKLEN command.
CMD25	adtc	[31:0] data address	R1	WRITE_MULTIPLE_BLOCK	Continuously writes blocks of data until a STOP_TRANSMISSION follows.
CMD26	adtc	[31:0] stuff bits	R1	PROGRAM_CID	Programming of the card identification register. This command is issued only once per card. The card contains hardware to prevent this operation after the first programming. Normally this command is reserved for the manufacturer.
CMD27	adtc	[31:0] stuff bits	R1	PROGRAM_CSD	Programming of the programmable bits of the CSD.

Table continues on the next page...

Table 10-39. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CMD28	ac	[31:0] data address	R1b	SET_WRITE_PROT	If the card has write protection features, this command sets the write protection bit of the addressed group. The properties of write protection are coded in the card specific data (WP_GRP_SIZE).
CMD29	ac	[31:0] data address	R1b	CLR_WRITE_PROT	If the card provides write protection features, this command clears the write protection bit of the addressed group.
CMD30	adtc	[31:0] write protect data address	R1	SEND_WRITE_PROT	If the card provides write protection features, this command asks the card to send the status of the write protection bits.
CMD31	Reserved				
CMD32	ac	[31:0] data address	R1	TAG_SECTOR_START	Sets the address of the first sector of the erase group.
CMD33	ac	[31:0] data address	R1	TAG_SECTOR_END	Sets the address of the last sector in a continuous range within the selection of a single sector to be selected for erase.
CMD34	ac	[31:0] data address	R1	UNTAG_SECTOR	Removes one previously selected sector from the erase selection.
CMD35	ac	[31:0] data address	R1	TAG_ERASE_GROUP_START	Sets the address of the first erase group within a range to be selected for erase.
CMD36	ac	[31:0] data address	R1	TAG_ERASE_GROUP_END	Sets the address of the last erase group within a continuous range to be selected for erase.
CMD37	ac	[31:0] data address	R1	UNTAG_ERASE_GROUP	Removes one previously selected erase group from the erase selection.
CMD38	ac	[31:0] stuff bits	R1b	ERASE	Erase all previously selected sectors.
CMD39	ac	[31:0] RCA [15] register write flag [14:8] register address [7:0] register data	R4	FAST_IO	Used to write and read 8-bit (register) data fields. The command addresses a card, and a register, and provides the data for writing if the write flag is set. The R4 response contains data read from the address register. This command accesses application dependent registers which are not defined in the MMC standard.
CMD40	bcr	[31:0] stuff bits	R5	GO_IRQ_STATE	Sets the system into interrupt mode.
CMD41	Reserved				

Table continues on the next page...

Table 10-39. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
CDM42	adtc	[31:0] stuff bits	R1b	LOCK_UNLOCK	Used to set/reset the password or lock/unlock the card. The size of the data block is set by the SET_BLOCK_LEN command.
CMD43~51	Reserved				
CMD52	ac	[31:0] stuff bits	R5	IO_RW_DIRECT	Access a single register within the total 128k of register space in any I/O function.
CMD53	ac	[31:0] stuff bits	R5	IO_RW_EXTENDED	Accesses a multiple I/O register with a single command. Allows the reading or writing of a large number of I/O registers.
CMD54	Reserved				
CMD55	ac	[31:16] RCA [15:0] stuff bits	R1	APP_CMD	Indicates to the card that the next command is an application specific command rather that a standard command.
CMD56	adtc	[31:1] stuff bits [0]: RD/WR	R1b	GEN_CMD	Used either to transfer a data block to the card or to get a data block from the card for general purpose / application specific commands. The size of the data block is set by the SET_BLOCK_LEN command.
CMD57-59	Reserved				
CMD60	adtc	[31] WR [30:24] stuff bits [23:16] address [15:8] stuff bits [7:0] byte count	R1b	RW_MULTIPLE_REGISTER	These registers are used to control the behavior of the device and to retrieve status information regarding the operation of the device. All Status and Control registers are WORD (32-bit) in size and are WORD aligned. CMD60 is used to read and write these registers.
CMD61	adtc	[31] WR [30:16] stuff bits [15:0] data unit count	R1b	RW_MULTIPLE_BLOCK	The host issues a RW_MULTIPLE_BLOCK (CMD61) to begin the data transfer.
CMD62-63	Reserved				
ACMD6[4]	ac	[31:2] stuff bits [1:0] bus width	R1	SET_BUS_WIDTH	Defines the data bus width ('00'=1bit or '10'=4bit bus) to be used for data transfer. The allowed data bus widths are given in SCR register.
ACMD13[4]	adtc	[31:0] stuff bits	R1	SD_STATUS	Send the SD Memory Card status.
ACMD22[4]	adtc	[31:0] stuff bits	R1	SEND_NUM_WR_SECTORS	Send the number of the written sectors (without errors). Responds with 32-bit plus the CRC data block.

Table continues on the next page...

Table 10-39. Commands for MMC/SD/SDIO cards (continued)

CMD INDEX	Type	Argument	Response type	Abbreviation	Description
ACMD23[4]	ac	[31:23] stuff bits [22:0] Number of blocks	R1	SET_WR_BLK_ERASE_COUNT	Set the number of write blocks to be pre-erased before writing (to be used for fast Multiple Block WR command). "1"=default(one write block).
ACMD41[4]	bcr	[31:0] OCR	R3	SD_APP_OP_COND	Asks the accessed card to send its operating condition register (OCR) contents in the response on the CMD line.
ACMD42[4]	ac	[31:1] stuff bits [0] set_cd	R1	SET_CLR_CARD_DETECT	Connect(1)/Disconnect(0) the 50KOhm pull-up resistor on CD_B/DATA3 of the card.
ACMD51[4]	adtc	[31:0] stuff bits	R1	SEND_SCR	Reads the SD Configuration Register (SCR).

1. CMD3 differs for MMC and SD cards. For MMC cards, it is referred to as SET_RELATIVE_ADDR, with a response type of R1. For SD cards, it is referred to as SEND_RELATIVE_ADDR, with a response type of R6 (with RCA inside).
2. CMD6 differs completely between high speed MMC cards and high-speed SD cards. Command SWITCH_FUNC is for high-speed SD cards.
3. Command SWITCH is for high speed MMC cards. The Index field can contain any value from 0-255, but only values 0-191 are valid. If the Index value is in the 192-255 range the card does not perform any modification and the SWITCH_ERROR status bit in the EXT_CSD register is set. The Access Bits are shown in Table 2.
4. ACMDs is preceded with the APP_CMD command. Commands listed are used for SD only, other SD commands not listed are not supported on this module.

The access bits for the EXT_CSD access modes are shown below.

Table 10-40. EXT_CSD access modes

Bits	Access name	Operation
00	Command set	The command set is changed according to the Cmd Set field of the argument.
01	Set bits	The bits in the pointed byte are set, according to the bits set to 1 in the Value field.
10	Clear bits	The bits in the pointed byte are cleared, according to the bits set to 1 in the Value field.
11	Write byte	The Value field is written into the pointed byte.

10.3.6 Software restrictions

10.3.6.1 Initialization active

The driver cannot set INITA bit in System Control register when any of the command line or data lines are active, so the driver must ensure both CDIHB and CIHB bits are cleared.

10.3.6.2 Software polling procedure

For polling read or write, after the software begins a buffer read or write, it must access exactly the number of times as the values set in the Watermark Level Register; moreover, if the block size is not a multiple of the value in the Watermark Level Register (read and write respectively), the software must access exactly the remaining number of words at the end of each block.

For example, for a read operation, if the RD_WML is 4, indicating the watermark level is 16 bytes, block size is 40 bytes, and the block number is 2, then the access times for the burst sequence in the whole transfer process must be 4, 4, 2, 4, 4, 2.

10.3.6.3 Suspend operation

To suspend the data transfer, the software must inform uSDHC that the suspend command is successfully accepted. To achieve this, after the Suspend command is accepted by the SDIO card, software must send another normal command marked as suspend command (CMDTYP bits set as '01') to inform uSDHC that the transfer is suspended.

If software needs to resume the suspended transfer, it should read the value in BLKCNT register to save the remaining number of blocks before sending the normal command marked as suspend, otherwise on sending such a 'suspend' command, uSDHC treats the current transfer is aborted and change the BLKCNT register to its original value, instead of retaining the remaining number of blocks.

10.3.6.4 Data length setting

For either ADMA (ADMA1 or ADMA2) transfer, the data in the data buffer must be word aligned, so the data length set in the descriptor must be a multiple of 4.

10.3.6.5 (A)DMA address setting

To configure the ADMA1/ADMA2/DMA address register, when TC bit is set, the register always updates itself with the internal address value to support dynamic address synchronization, so the software must ensure that the TC bit is cleared prior to configuring the ADMA1/ADMA2/DMA address register.

10.3.6.6 Data port access

Data port does not support parallel access. For example, during an internal DMA access, it is not allowed to write any data to the data port by CPU; or during a CPU read operation, it is also prohibited to write any data to the data port, by either CPU or internal DMA. Otherwise the data is corrupted inside the uSDHC buffer.

10.3.6.7 Change clock frequency

The uSDHC module does not automatically gate off the card clock when the host driver changes the clock frequency. To prevent possible glitch on the card clock, clear the FRC_SDCLK_ON bit when changing the clock divisor value (SDCLKFS or DVS in System Control Register) or setting the RSTA bit.

Also, before changing the clock divisor value, the host driver should make sure that the SDSTB bit is high.

10.3.6.8 Multi-block read

For pre-defined multi-block read operation, that is, the number of blocks to read has been defined by previous CMD23 for MMC, or pre-defined number of blocks in CMD53 for SDIO/SDCombo, or whatever multi-block read without abort command at card side, an abort command, either automatic or manual CMD12/CMD52, is still required by uSDHC after the pre-defined number of blocks are done, to drive the internal state machine to idle mode.

In this case, the card may not respond to this extra abort command and uSDHC gets response timeout. It is recommended to manually send an abort command with RSPTYP[1:0] both bits cleared.

10.3.7 uSDHC memory map/register definition

10.3.7.1 uSDHC register descriptions

This section includes the module memory map and detailed descriptions of all registers.

See the table below for the register memory map of uSDHC. All these registers only support 32-bit accesses.

NOTE

The uSDHC registers are 32-bit wide and only support 32-bit access.

10.3.7.1.1 uSDHC memory map

uSDHC1 base address: 30B4_0000h

uSDHC2 base address: 30B5_0000h

uSDHC3 base address: 30B6_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	DMA System Address (DS_ADDR)	32	RW	0000_0000h
4h	Block Attributes (BLK_ATT)	32	RW	0001_0000h
8h	Command Argument (CMD_ARG)	32	RW	0000_0000h
Ch	Command Transfer Type (CMD_XFR_TYP)	32	RW	0000_0000h
10h	Command Response0 (CMD_RSP0)	32	RO	0000_0000h
14h	Command Response1 (CMD_RSP1)	32	RO	0000_0000h
18h	Command Response2 (CMD_RSP2)	32	RO	0000_0000h
1Ch	Command Response3 (CMD_RSP3)	32	RO	0000_0000h
20h	Data Buffer Access Port (DATA_BUFF_ACC_PORT)	32	RW	0000_0000h
24h	Present State (PRES_STATE)	32	RO	Table 10-248
28h	Protocol Control (PROT_CTRL)	32	RW	0880_0020h
2Ch	System Control (SYS_CTRL)	32	RW	0080_800Fh
30h	Interrupt Status (INT_STATUS)	32	W1C	0000_0000h
34h	Interrupt Status Enable (INT_STATUS_EN)	32	RW	0000_0000h
38h	Interrupt Signal Enable (INT_SIGNAL_EN)	32	RW	0000_0000h
3Ch	Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)	32	RW	0000_0000h
40h	Host Controller Capabilities (HOST_CTRL_CAP)	32	RW	07F3_B407h
44h	Watermark Level (WTMK_LVL)	32	RW	0810_0810h
48h	Mixer Control (MIX_CTRL)	32	RW	8000_0000h
50h	Force Event (FORCE_EVENT)	32	WORZ	0000_0000h
54h	ADMA Error Status (ADMA_ERR_STATUS)	32	RO	0000_0000h
58h	ADMA System Address (ADMA_SYS_ADDR)	32	RW	0000_0000h
60h	DLL (Delay Line) Control (DLL_CTRL)	32	RW	0000_0000h
64h	DLL Status (DLL_STATUS)	32	RO	0000_0200h
68h	CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)	32	RW	0000_0000h
70h	Strobe DLL control (STROBE_DLL_CTRL)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
74h	Strobe DLL status (STROBE_DLL_STATUS)	32	RO	0000_0200h
C0h	Vendor Specific Register (VEND_SPEC)	32	RW	3000_7809h
C4h	MMC Boot (MMC_BOOT)	32	RW	0000_0000h
C8h	Vendor Specific 2 Register (VEND_SPEC2)	32	RW	0001_9006h
CCh	Tuning Control (TUNING_CTRL)	32	RW	0021_2800h
100h	Command Queue (CQE)	32	ROZ	0000_0510h

10.3.7.1.2 DMA System Address (DS_ADDR)

10.3.7.1.2.1 Offset

Register	Offset
DS_ADDR	0h

10.3.7.1.2.2 Function

This register contains the physical system memory address used for DMA transfers.

10.3.7.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.2.4 Fields

Field	Function
31-0 DS_ADDR	System address **DMA system address / argument 2**

Field	Function
	When ACMD23_ARGU2_EN is set to 0, SDMA uses this register as system address and supports only 32-bit addressing mode. Auto CMD23 cannot be used with SDMA. When ACMD23_ARGU2_EN is set to 1, SDMA uses ADMA System Address register (05Fh – 058h) instead of this register to support both 32-bit and 64-bit addressing. This register is used only for Argument2 and SDMA may use Auto CMD23. 1. **SDMA system address** Because the address must be word (4 bytes) aligned, the least 2 bits are reserved, always 0. When uSDHC stops a DMA transfer, this register points out the system address of the next contiguous data position. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operation during transfers may return an invalid value. The host driver initializes this register before starting a DMA transaction. After DMA has stopped, the system address of the next contiguous data position can be read from this register. This register is protected during a data transfer. When data lines are active, write to this register is ignored. The host driver waits until the DLA bit in the Present State register is cleared, before writing to this register. The uSDHC internal DMA does not support a virtual memory system. It only supports continuous physical memory access. And due to AHB burst limitations, if the burst must cross the 1 KB boundary, uSDHC automatically changes SEQ burst type to NSEQ. Because this register supports dynamic address reflecting, when TC bit is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC bit is set. Such restriction is also listed in Software restrictions. 2. **Argument 2** This register is used with the Auto CMD23 to set a 32-bit block count value to the argument of the CMD23 while executing Auto CMD23. If Auto CMD23 is used with ADMA, the full 32-bit block count value can be used. If Auto CMD23 is used without ADMA, the available block count value is limited by the Block Count register. 65535 blocks is the maximum value in this case.

10.3.7.1.3 Block Attributes (BLK_ATT)

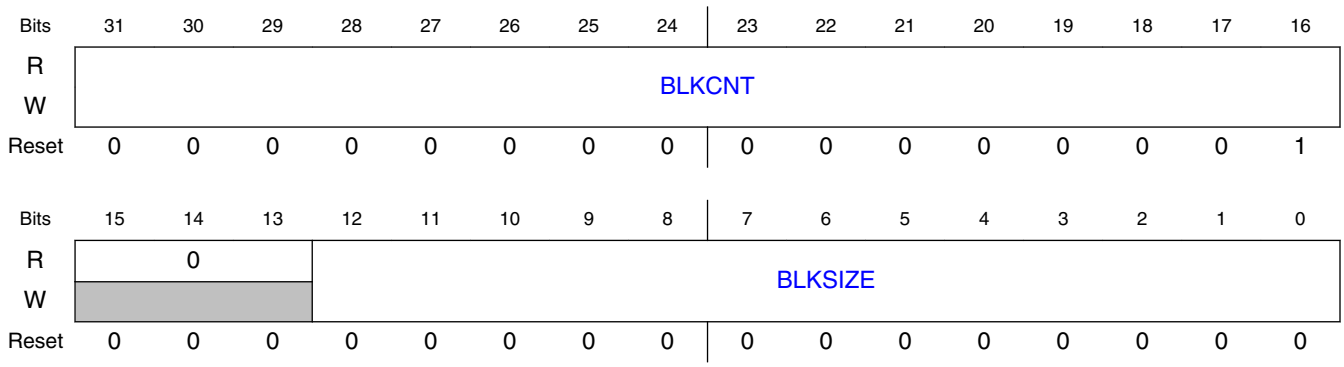
10.3.7.1.3.1 Offset

Register	Offset
BLK_ATT	4h

10.3.7.1.3.2 Function

This register is used to configure the number of data blocks and the number of bytes in each block.

10.3.7.1.3.3 Diagram



10.3.7.1.3.4 Fields

Field	Function
31-16 BLKCNT	Blocks count for current transfer This register is enabled when the Block Count Enable field in the Transfer Mode register is set to 1 and is valid only for multiple block transfers. For single block transfer, this register always reads as 1. The host driver sets this register to a value between 1 and the maximum block count. The uSDHC module decrements the block count after each block transfer and stops when the count reaches zero. Setting the block count to zero results in no data blocks being transferred. This register should be accessed only when no transaction is executing (that is, after transactions are stopped). During data transfer, read operations on this register may return an invalid value and write operations are ignored. When saving transfer content because of a Suspend command, the number of blocks yet to be transferred can be determined by reading this register. The reading of this register should be applied after transfer is paused by stop at block gap operation and before sending the command marked as suspend. This is because when the Suspend command is sent out, uSDHC treats the current transfer as aborted and change the BLKCNT register back to its original value instead of keeping the dynamical indicator of the remaining block count. When restoring transfer content prior to issuing a Resume command, the host driver restores the previously saved block count. **NOTE:** Although the BLKCNT field is 0 after reset, the read of reset value is 0x1. This is because when MSBSEL field is indicating a single block transfer, the read value of BLKCNT is always 1. 0000000000000000b - Stop count 0000000000000001b - 1 block 0000000000000010b - 2 blocks 1111111111111111b - 65535 blocks
15-13 —	Reserved
12-0 BLKSIZE	Transfer block size This register specifies the block size for block data transfers. Values ranging from 1 byte up to the maximum buffer size can be set. It can be accessed only when no transaction is executing (that is, after a transaction has stopped). Read operations during transfers may return an invalid value, and write operations are ignored. 0000000000000b - No data transfer 0000000000001b - 1 byte

Field	Function
	0000000000010b - 2 bytes 0000000000011b - 3 bytes 0000000000100b - 4 bytes 0000111111111b - 511 bytes 0001000000000b - 512 bytes 0100000000000b - 2048 bytes 1000000000000b - 4096 bytes

10.3.7.1.4 Command Argument (CMD_ARG)

10.3.7.1.4.1 Offset

Register	Offset
CMD_ARG	8h

10.3.7.1.4.2 Function

This register contains the SD/MMC command argument.

10.3.7.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMDARG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.4.4 Fields

Field	Function
31-0 CMDARG	Command argument The SD/MMC command argument is specified as bits 39-8 of the command format in the SD or MMC specification. This register is write protected when the Command Inhibit (CMD) field in the Present State register is set.

10.3.7.1.5 Command Transfer Type (CMD_XFR_TYP)

10.3.7.1.5.1 Offset

Register	Offset
CMD_XFR_TYP	Ch

10.3.7.1.5.2 Function

This register is used to control the operation of data transfers. The host driver sets this register before issuing a command followed by a data transfer or before issuing a Resume command. To prevent data loss, uSDHC prevents writing to the bits, which are involved in the data transfer of this register, when data transfer is active. These bits are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

The host driver checks the Command Inhibit DAT field (PRES_STATE[CDIHB]) and the Command Inhibit CMD field (PRES_STATE[CIHB]) in the Present State register before writing to this register. When the CDIHB field in the Present State register is set, any attempt to send a command with data by writing to this register is ignored; when the CIHB field is set, any write to this register is ignored.

On sending commands with data transfer involved, it is mandatory that the block size is non-zero. Block count must also be non-zero, or indicated as a single block transfer (bit 5 of this register is '0' when written), or block count is disabled (bit 1 of this register is '0' when written), otherwise uSDHC ignores the sending of this command and do nothing. For write command, with all above restrictions, it is also mandatory that the write protect switch is not active (PRES_STATE[WPSPL] field of Present State register is '1'); otherwise, uSDHC also ignores the command.

If the commands with data transfer do not receive the response in 64 clock cycles, that is, if response time-out happens, uSDHC treats the external device, does not accept the command, and aborts the data transfer. In this scenario, the driver should issue the command again to retry the transfer. It is also possible that for some reason the card responds to the command but uSDHC does not receive the response, and if it is internal DMA (either simple DMA or ADMA) read operation, the external system memory is over-written by the internal DMA with data sent back from the card.

The table below shows the summary of how register settings determine the type of data transfer.

Table 10-41. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

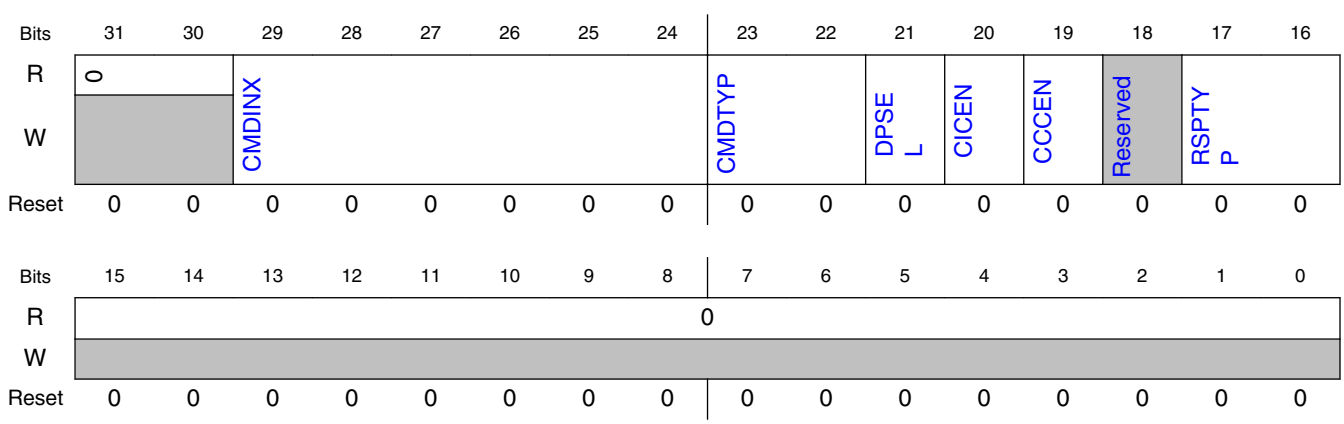
The table below shows the relationship between the Command Index Check Enable and the Command CRC Check Enable, regarding the Response Type bits as well as the name of the response type.

Table 10-42. Relationship between parameters and the name of the response type

Response type	Index check enable	CRC check enable	Name of response type
00	0	0	No response
01	0	1	R2
10	0	0	R3, R4
10	1	1	R1, R5, R6
11	1	1	R1b,R5b

- In the SDIO specification, response type notation for R5b is not defined. R5 includes R5b in the SDIO specification, but R5b is defined in this specification to specify that uSDHC checks the busy status after receiving a response. For example, usually CMD52 is used with R5, but the I/O abort command is used with R5b.
- The CRC fields for R3 and R4 are expected to be all 1 bits. The CRC check is disabled for these response types.

10.3.7.1.5.3 Diagram



10.3.7.1.5.4 Fields

Field	Function
31-30 —	Reserved
29-24 CMDINX	Command index These bits are set to the command number that is specified in bits 45-40 of the command-format in the SD Memory Card Physical Layer Specification and SDIO Card Specification.
23-22 CMDTYP	Command type There are three types of special commands: Suspend, Resume, and Abort. These bits are set to 00b for all other commands. • Suspend command: If the Suspend command succeeds, uSDHC assumes that the card bus has been released and that it is possible to issue the next command that uses the DATA line. Because uSDHC does not monitor the content of command response, it does not know if the Suspend command succeeded or not. It is the host driver's responsibility to check the status of the Suspend command and send another command marked as Suspend to inform uSDHC that a Suspend command was successfully issued. See Suspend Resume for more details. After the end bit of command is sent, uSDHC deasserts Read Wait for read transactions and stops checking busy for write transactions. In a 4-bit mode, the interrupt cycle starts. If the Suspend command fails, uSDHC maintains its current state, and the host driver restarts the transfer by setting the Continue Request field in the Protocol Control register. • Resume command: The host driver re-starts the data transfer by restoring the registers saved before sending the Suspend command and then sends the Resume command. The uSDHC module checks for a pending busy state before starting write transfers. • Abort command: If this command is set when executing a read transfer, uSDHC stops reads to the buffer. If this command is set when executing a write transfer, uSDHC stops driving the DATA line. After issuing the Abort command, the host driver should issue a software reset (Abort Transaction). 00b - Normal other commands 01b - Suspend CMD52 for writing bus suspend in CCCR 10b - Resume CMD52 for writing function select in CCCR 11b - Abort CMD12, CMD52 for writing I/O Abort in CCCR
21 DPSEL	Data present select This field is set to 1 to indicate that data is present and is transferred using the DATA line. It is set to 0 for the following: • Commands using only the CMD line (for example, CMD52) • Commands with no data transfer, but using the busy signal on DATA0 line (R1b or R5b (for example, CMD38)) **NOTE:** In resume command, this field is set, and other bits in this register is set the same as when the transfer was initially launched. When the write protect switch is on, (that is, the WPSPL field is active as '0'), any command with a write operation ignored. When this field is set, while the DTDSEL field is 0, writes to the register Transfer Type are ignored. 0b - No data present 1b - Data present
20 CICEN	Command index check enable If this field is set to 1, uSDHC checks the Index field in the response to see if it has the same value as the command index. If it is not, it is reported as a Command Index Error. If this field is set to 0, the Index field is not checked. 0b - Disable command index check 1b - Enables command index check

Table continues on the next page...

Field	Function
19 CCCEN	Command CRC check enable If this field is set to 1, uSDHC checks the CRC field in the response. If an error is detected, it is reported as a Command CRC Error. If this field is set to 0, the CRC field is not checked. The number of bits checked by the CRC field value changes according to the length of the response. See RSPTYP[1:0] and Command Transfer Type (CMD_XFR_TYP). 0b - Disables command CRC check 1b - Enables command CRC check
18 —	Reserved
17-16 RSPTYP	Response type select 00b - No response 01b - Response length 136 10b - Response length 48 11b - Response length 48, check busy after response
15-0 —	Reserved

10.3.7.1.6 Command Response0 (CMD_RSP0)

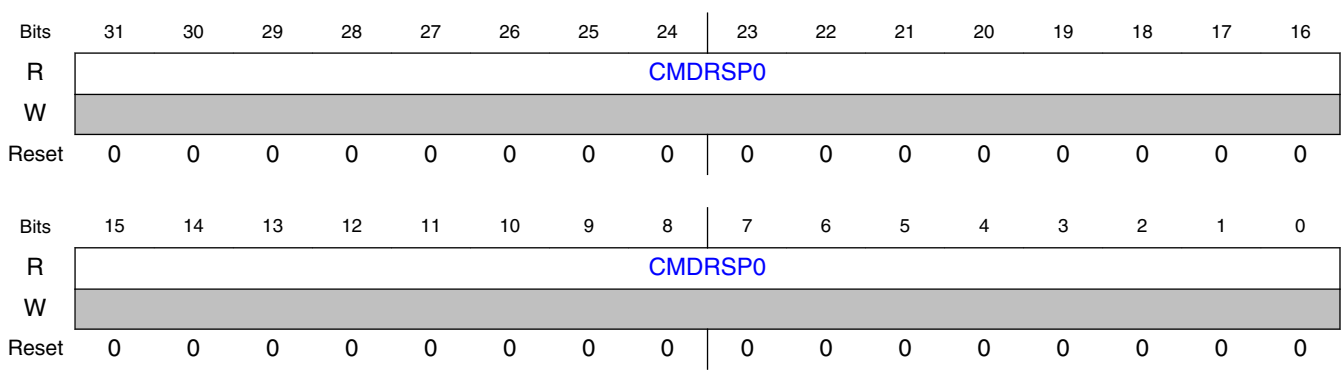
10.3.7.1.6.1 Offset

Register	Offset
CMD_RSP0	10h

10.3.7.1.6.2 Function

This register is used to store part 0 of the response bits from the card.

10.3.7.1.6.3 Diagram



10.3.7.1.6.4 Fields

Field	Function
31-0 CMDRSP0	Command response 0 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this register for each response type.

10.3.7.1.7 Command Response1 (CMD_RSP1)

10.3.7.1.7.1 Offset

Register	Offset
CMD_RSP1	14h

10.3.7.1.7.2 Function

This register is used to store part 1 of the response bits from the card.

10.3.7.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.7.4 Fields

Field	Function
31-0 CMDRSP1	Command response 1 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this register for each response type.

10.3.7.1.8 Command Response2 (CMD_RSP2)

10.3.7.1.8.1 Offset

Register	Offset
CMD_RSP2	18h

10.3.7.1.8.2 Function

This register is used to store part 2 of the response bits from the card.

10.3.7.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.8.4 Fields

Field	Function
31-0 CMDRSP2	Command response 2 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this register for each response type.

10.3.7.1.9 Command Response3 (CMD_RSP3)

10.3.7.1.9.1 Offset

Register	Offset
CMD_RSP3	1Ch

10.3.7.1.9.2 Function

This register is used to store part 3 of the response bits from the card.

The table below describes the mapping of command responses from the SD bus to Command Response registers for each response type. In this table, R[ ] refers to a bit range within the response data as transmitted on the SD bus.

Table 10-43. Response bit definition for each response type

Response type	Meaning of response	Response field	Response register
R1,R1b (normal response)	Card status	R[39:8]	CMDRSP0
R1b (auto CMD12 response)	Card status for auto CMD12	R[39:8]	CMDRSP3
R2 (CID, CSD register)	CID/CSD register [127:8]	R[127:8]	{CMDRSP3[23:0], CMDRSP2, CMDRSP1, CMDRSP0}
R3 (OCR register)	OCR register for memory	R[39:8]	CMDRSP0
R4 (OCR register)	OCR register for I/O etc.	R[39:8]	CMDRSP0
R5, R5b	SDIO response	R[39:8]	CMDRSP0
R6 (publish RCA)	New published RCA[31:16] and card status[15:0]	R[39:9]	CMDRSP0

This table shows that most responses with a length of 48 (R[47:0]) have 32-bits of the response data (R[39:8]) stored in the CMDRSP0 register. Responses of type R1b (Auto CMD12 responses) have response data bits (R[39:8]) stored in the CMDRSP3 register. Responses with length 136 (R[135:0]) have 120-bits of the response data (R[127:8]) stored in the CMDRSP0, 1, 2, and 3 registers.

To be able to read the response status efficiently, uSDHC only stores part of the response data in the Command Response registers. This enables the host driver to efficiently read 32-bits of response data in one read cycle on a 32-bit bus system. Parts of the response, the Index field and the CRC, are checked by uSDHC (as specified by the Command Index Check Enable and the Command CRC Check Enable bits in the Transfer Type register) and generate an error interrupt if any error is detected. The bit range for the CRC check depends on the response length. If the response length is 48, uSDHC checks R[47:1], and if the response length is 136 the uSDHC checks R[119:1].

Because uSDHC may have a multiple block data transfer executing concurrently with a CMD_wo_DAT command, uSDHC stores the Auto CMD12 response in the CMDRSP3 register. The CMD_wo_DAT response is stored in CMDRSP0. This allows uSDHC to

avoid overwriting the Auto CMD12 response with the CMD_wo_DAT and vice versa. When uSDHC modifies part of the Command Response registers, as shown in the table above, it preserves the unmodified bits.

10.3.7.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CMDRSP3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.9.4 Fields

Field	Function
31-0 CMDRSP3	Command response 3 See Command Response3 (CMD_RSP3) for the mapping of command responses from the SD bus to this register for each response type.

10.3.7.1.10 Data Buffer Access Port (DATA_BUFF_ACC_PORT)

10.3.7.1.10.1 Offset

Register	Offset
DATA_BUFF_ACC_PORT	20h

10.3.7.1.10.2 Function

This is a 32-bit data port register used to access the internal buffer.

10.3.7.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATCONT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.10.4 Fields

Field	Function
31-0 DATCONT	Data content The Buffer Data Port register is for 32-bit data access by the Arm platform or the external DMA. When the internal DMA is enabled, any write to this register is ignored, and any read from this register always yields 0s.

10.3.7.1.11 Present State (PRES_STATE)

10.3.7.1.11.1 Offset

Register	Offset
PRES_STATE	24h

10.3.7.1.11.2 Function

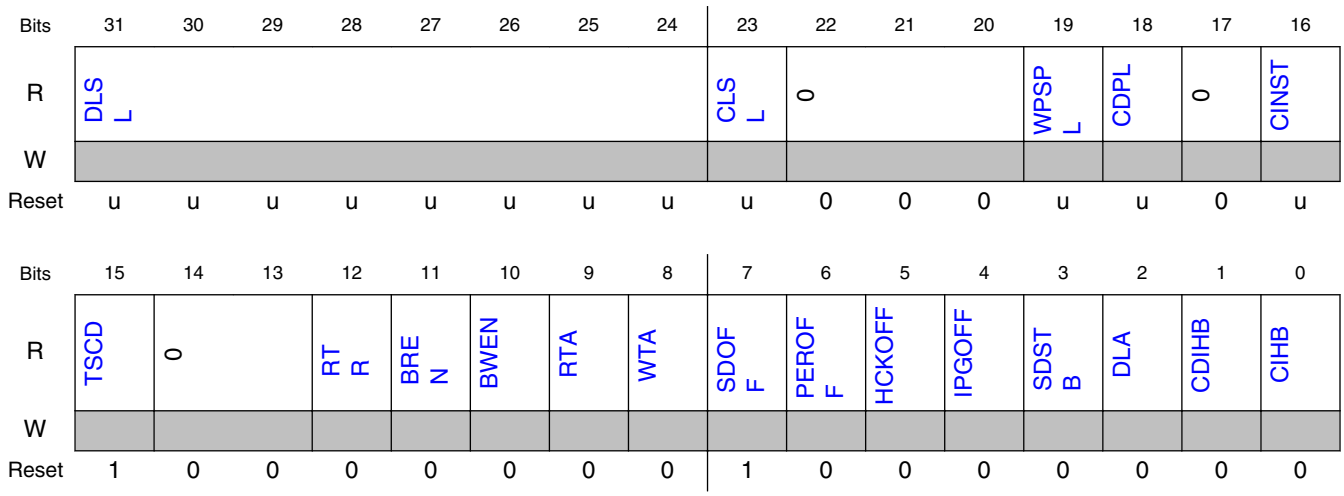
The host driver can get status of uSDHC from this 32-bit read only register.

The host driver can issue CMD0, CMD12, CMD13 (for memory) and CMD52 (for SDIO) when the DATA lines are busy during a data transfer. These commands can be issued when Command Inhibit (CMD) is set to zero. Other commands are issued when Command Inhibit (DATA) is set to zero. Possible changes to the SD Physical Specification may add other commands to this list in the future.

NOTE

The reset value of Present State register depends on board connectivity.

10.3.7.1.11.3 Diagram



10.3.7.1.11.4 Fields

Field	Function
31-24 DLSL	DATA[7:0] line signal level This status is used to check the DATA line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DATA0. The reset value is affected by the external pull-up / pull-down resistors. By default, the read value of this field field after reset is 8'b11110111, when DATA3 is pulled down and the other lines are pulled up. 00000000b - Data 0 line signal level 00000001b - Data 1 line signal level 00000010b - Data 2 line signal level 00000011b - Data 3 line signal level 00000100b - Data 4 line signal level 00000101b - Data 5 line signal level 00000110b - Data 6 line signal level 00000111b - Data 7 line signal level
23 CLSL	CMD line signal level This status is used to check the CMD line level to recover from errors, and for debugging. The reset value is affected by the external pull-up / pull-down resistor, by default, the read value of this field after reset is 1'b1, when the command line is pulled up.
22-20 —	Reserved
19 WPSPL	Write protect switch pin level The Write Protect switch is supported for memory and combo cards. This field reflects the inverted value of the WP pin of the card socket. A software reset does not affect this field. The reset value is affected by the external write protect switch. If the WP pin is not used, it should be tied low, so that the reset value of this field is high and write is enabled. 0b - Write protected (WP = 1) 1b - Write enabled (WP = 0)

Table continues on the next page...

Field	Function
18 CDPL	Card detect pin level This field reflects the inverse value of the CD_B pin for the card socket. Debouncing is not performed on this field. This field may be valid, but is not guaranteed, because of propagation delay. Use of this field is limited to testing because it must be debounced by software. A software reset does not affect this field. A write to the Force Event Register does not affect this field. The reset value is affected by the external card detection pin. This field shows the value on the CD_B pin (that is, when a card is inserted in the socket, it is 0 on the CD_B input, and consequently, the CDPL reads 1. 0b - No card present (CD_B = 1) 1b - Card present (CD_B = 0)
17 —	Reserved
16 CINST	Card inserted This field indicates whether a card has been inserted. The uSDHC module debounces this signal so that the host driver does not need to wait for it to stabilize. Changing from a 0 to 1 generates a Card Insertion interrupt in the Interrupt Status register. Changing from a 1 to 0 generates a Card Removal interrupt in the Interrupt Status register. A write to the Force Event Register does not affect this field. The Software Reset For All in the System Control register does not affect this field. A software reset does not affect this field. 0b - Power on reset or no card 1b - Card inserted
15 TSCD	Tape select change done This field indicates the delay setting is effective after write CLK_TUNE_CTRL_STATUS register. 0b - Delay cell select change is not finished. 1b - Delay cell select change is finished.
14-13 —	Reserved
12 RTR	Re-Tuning Request (only for SD3.0 SDR104 mode and EMMC HS200 mode) Host controller may request host driver to execute re-tuning sequence by setting this field when the data window is shifted by temperature drift and a tuned sampling point does not have a good margin to receive correct data. This field is cleared when a command is issued with setting Execute Tuning field in MIXER_CTRL register. Changing of this field from 0 to 1 generates Re-Tuning Event. See Interrupt status registers for more detail. This field isn't set to 1 if Sampling Clock Select in the MIXER_CTRL register is set to 0 (using fixed sampling clock). 0b - Fixed or well tuned sampling clock 1b - Sampling clock needs re-tuning
11 BREN	Buffer read enable This status field is used for non-DMA read transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates that valid data exists in the host side buffer. If this field is high, valid data greater than the watermark level exist in the buffer. A change of this field from 1 to 0 occurs when some reads from the buffer (read DATPORT (Base + 0x20)) are made and the buffer hasn't valid data greater than the watermark level. A change of this field from 0 to1 occurs when there is enough valid data ready in the buffer and the Buffer Read Ready interrupt has been generated and enabled. 0b - Read disable

Table continues on the next page...

Field	Function
	1b - Read enable
10 BWEN	Buffer write enable This status field is used for non-DMA write transfers. The uSDHC module implements an internal buffer to transfer data efficiently. This read only flag indicates if space is available for write data. If this field is 1, valid data greater than the watermark level can be written to the buffer. A change of this field from 1 to 0 occurs when some writes to the buffer (write DATPORT(Base + 0x20)) are made and the buffer hasn't valid space greater than the watermark level. A change of this field from 0 to 1 occurs when the buffer can hold valid data greater than the write watermark level and the Buffer Write Ready interrupt is generated and enabled. 0b - Write disable 1b - Write enable
9 RTA	Read transfer active This status field is used for detecting completion of a read transfer. This field is set for either of the following conditions: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer A transfer complete interrupt is generated when this field changes to 0. This field is cleared for either of the following conditions: • When the last data block as specified by block length is transferred to the System, that is, all data are read away from uSDHC internal buffer. • When all valid data blocks have been transferred from uSDHC internal buffer to the System and no current block transfers are being sent because of the Stop At Block Gap Request being set to 1. 0b - No valid data 1b - Transferring data
8 WTA	Write transfer active This status field indicates a write transfer is active. If this field is 0, it means no valid write data exists in uSDHC. This field is set in either of the following cases: • After the end field of the write command • When writing 1 to the Continue Request field in the Protocol Control register to restart a write transfer This field is cleared in either of the following cases: • After getting the CRC status of the last data block as specified by the transfer count (Single and Multiple) • After getting the CRC status of any block where data transmission is about to be stopped by a Stop At Block Gap Request During a write transaction, a Block Gap Event interrupt is generated when this field is changed to 0, as result of the Stop At Block Gap Request being set. This status is useful for the host driver in determining when to issue commands during Write Busy state. 0b - No valid data 1b - Transferring data
7 SDOFF	SD clock gated off internally This status field indicates that the SD clock is internally gated off, because of buffer over / under-run or read pause without read wait assertion, or the driver set FRC_SDCLK_ON field is 0 to stop the SD clock in idle status. This field is for the host driver to debug data transaction on the SD bus.

Table continues on the next page...

Field	Function
	0b - SD clock is active. 1b - SD clock is gated off.
6 PEROFF	IPG_PERCLK gated off internally This status field indicates that the IPG_PERCLK is internally gated off. This field is for the host driver to debug transaction on the SD bus. When IPG_CLK_SOFT_EN is cleared, IPG_PERCLK is gated off, otherwise IPG_PERCLK is always active. 0b - IPG_PERCLK is active. 1b - IPG_PERCLK is gated off.
5 HCKOFF	HCLK gated off internally This status field indicates that the HCLK is internally gated off. This field is for the host driver to debug during a data transfer. 0b - HCLK is active. 1b - HCLK is gated off.
4 IPGOFF	Peripheral clock gated off internally This status field indicates that the peripheral clock is internally gated off. This field is for the host driver to debug. 0b - Peripheral clock is active. 1b - Peripheral clock is gated off.
3 SDSTB	SD clock stable This status field indicates that the internal card clock is stable. This field is for the host driver to poll clock status when changing the clock frequency. It is recommended to clear FRC_SDCLK_ON field in System Control register to remove glitches on the card clock when the frequency is changing. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. 0b - Clock is changing frequency and not stable. 1b - Clock is stable.
2 DLA	Data line active This status field indicates whether one of the DATA lines on the SD bus is in use. In the case of read transactions: This status indicates if a read transfer is executing on the SD bus. Changes in this value from 1 to 0, between data blocks, generates a Block Gap Event interrupt in the Interrupt Status register. This field is set in either of the following cases: • After the end field of the read command • When writing a 1 to the Continue Request field in the Protocol Control register to restart a read transfer This field is cleared in either of the following cases: • When the end field of the last data block is sent from the SD bus to uSDHC. • When the Read Wait state is stopped by a Suspend command and the DATA2 line is released. The uSDHC module waits at the next block gap by driving Read Wait at the start of the interrupt cycle. If the Read Wait signal is already driven (data buffer cannot receive data), uSDHC can wait for a current block gap by continuing to drive the Read Wait signal. It is necessary to support Read Wait to use the suspend / resume function. This field remains 1 during Read Wait. In the case of write transactions: This status indicates that a write transfer is executing on the SD bus. Changes in this value from 1 to 0 generate a Transfer Complete interrupt in the Interrupt Status register.

Table continues on the next page...

Field	Function
	This field is set in either of the following cases: • After the end field of the write command • When writing to 1 to the Continue Request field in the Protocol Control register to continue a write transfer This field is cleared in either of the following cases: • When the SD card releases Write Busy of the last data block, uSDHC also detects if the output is not busy. If the SD card does not drive the busy signal after the CRC status is received, uSDHC assumes the card drive "Not Busy". • When the SD card releases write busy, prior to waiting for write transfer, and because of a Stop At Block Gap Request. In the case of command with busy pending: This status indicates that a busy state follows the command and the data line is in use. This field is cleared when the DATA0 line is released. 0b - DATA line inactive 1b - DATA line active
1 CDIHB	Command inhibit (DATA) This status field is generated if either the DAT Line Active or the Read Transfer Active is set to 1. If this field is 0, it indicates that uSDHC can issue the next SD / MMC Command. Commands with a busy signal belong to Command Inhibit (DATA) (for example. R1b, R5b type). Changing from 1 to 0 generates a Transfer Complete interrupt in the Interrupt Status register. **NOTE:** The SD host driver can save registers for a suspend transaction after this field has changed from 1 to 0. 0b - Can issue command that uses the DATA line 1b - Cannot issue command that uses the DATA line
0 CIHB	Command inhibit (CMD) If this status bit is 0, it indicates that the CMD line is not in use and uSDHC can issue a SD / MMC command using the CMD line. This field is set also immediately after the Transfer Type register is written. This field is cleared when the command response is received. Even if the Command Inhibit (DATA) is set to 1, commands using only the CMD line can be issued if this field is 0. Changing from 1 to 0 generates a Command Complete interrupt in the Interrupt Status register. If uSDHC cannot issue the command because of a command conflict error (see Command CRC Error) or because of a Command Not Issued By Auto CMD12 Error, this field remains 1 and the Command Complete is not set. The Status of issuing an auto CMD12 does not show on this field. 0b - Can issue command using only CMD line 1b - Cannot issue command

10.3.7.1.12 Protocol Control (PROT_CTRL)

10.3.7.1.12.1 Offset

Register	Offset
PROT_CTRL	28h

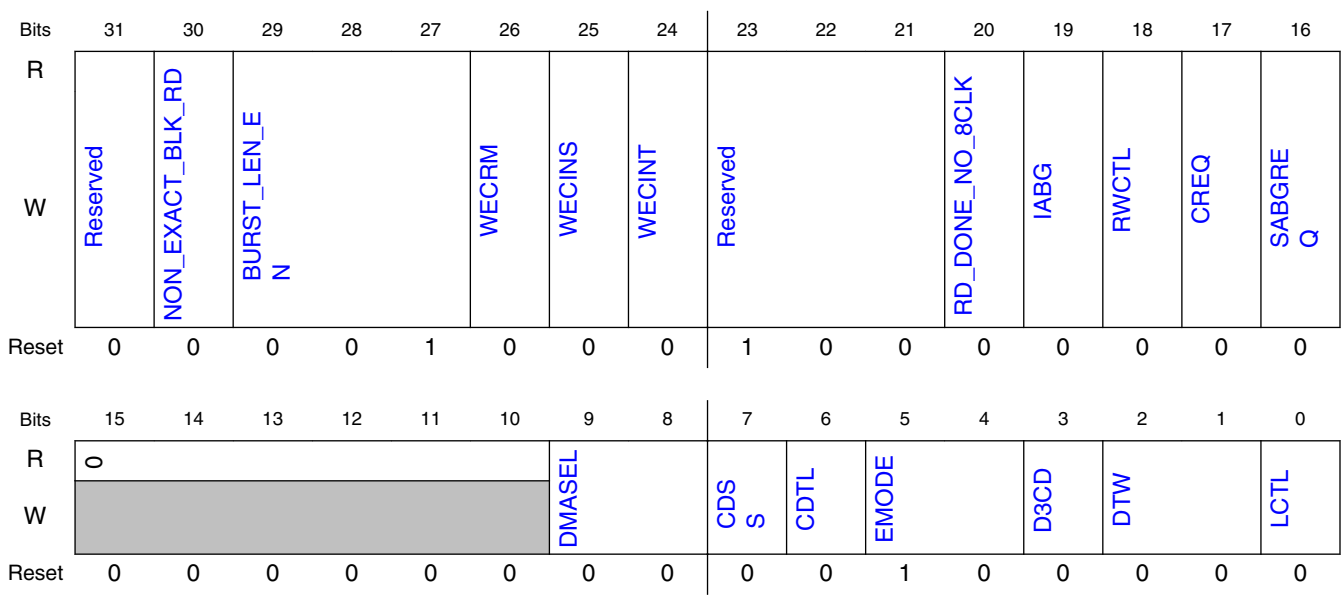
10.3.7.1.12.2 Function

There are three cases to restart the transfer after stop at the block gap. Which case is appropriate depends on whether uSDHC issues a Suspend command or the SD card accepts the Suspend command.

- If the host driver does not issue a Suspend command, the Continue request is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card accepts it, a Resume command is used to restart the transfer.
- If the host driver issues a Suspend command and the SD card does not accept it, the Continue request is used to restart the transfer.

Any time stop at block gap request stops the data transfer, the host driver waits for a Transfer Complete (in the Interrupt Status register), before attempting to restart the transfer. When restarting the data transfer by Continue Request, the host driver clears the Stop At Block Gap Request before or simultaneously.

10.3.7.1.12.3 Diagram



10.3.7.1.12.4 Fields

Field	Function
31	Reserved
—	Always write as 0

Table continues on the next page...

Field	Function
30 NON_EXACT_BLK_RD	Non-exact block read Current block read is non-exact block read. It is only used for SDIO. 0b - The block read is exact block read. Host driver does not need to issue abort command to terminate this multi-block read. 1b - The block read is non-exact block read. Host driver needs to issue abort command to terminate this multi-block read.
29-27 BURST_LEN_EN	BURST length enable for INCR, INCR4 / INCR8 / INCR16, INCR4-WRAP / INCR8-WRAP / INCR16-WRAP This is used to enable / disable the burst length for the external AHB2AXI bridge. It is useful especially for INCR transfer because without burst length indicator, the AHB2AXI bridge does not know the burst length in advance. Without burst length indicator, AHB INCR transfers can only be converted to SINGLEs on the AXI side. 1xxb - Burst length is enabled for INCR4-WRAP / INCR8-WRAP / INCR16-WRAP. x1xb - Burst length is enabled for INCR4 / INCR8 / INCR16. xx1b - Burst length is enabled for INCR.
26 WECRM	Wakeup event enable on SD card removal This field enables a wakeup event, via a card removal, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this field. When this field is set, the Card Removal Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Removal Status and uSDHC interrupt. 0b - Disables wakeup event enable on SD card removal 1b - Enables wakeup event enable on SD card removal
25 WECINS	Wakeup event enable on SD card insertion This field enables a wakeup event, via a card insertion, in the Interrupt Status register. FN_WUS (Wake Up Support) in CIS does not affect this field. When this field is set, the Card Insertion Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Insertion Status and uSDHC interrupt. 0b - Disable wakeup event enable on SD card insertion 1b - Enable wakeup event enable on SD card insertion
24 WECINT	Wakeup event enable on card interrupt This field enables a wakeup event, via a card interrupt, in the Interrupt Status register. This field can be set to 1 if FN_WUS (Wake Up Support) in CIS is set to 1. When this field is set, the Card Interrupt Status and uSDHC interrupt can be asserted without CLK toggling. When the wakeup feature is not enabled, the CLK must be active to assert the Card Interrupt Status and uSDHC interrupt. 0b - Disables wakeup event enable on card interrupt 1b - Enables wakeup event enable on card interrupt
23-21 —	Reserved Always write as 3'b100
20 RD_DONE_NO_8CLK	Read performed number 8 clock According to the SD/MMC spec, for read data transaction, 8 clocks are needed after the end field of the last data block. So, by default(RD_DONE_NO_8CLK=0), 8 clocks are active after the end field of the last read data transaction. However, these 8 clocks should not be active if user wants to use stop at block gap (include the auto stop at block gap in boot mode) feature for read and the RWCTL field (bit18) is not enabled. In this case, software should set RD_DONE_NO_8CLK to avoid these 8 clocks. Otherwise, the device might send extra data to uSDHC while uSDHC ignores these data. In a summary, this field should be set only if the use case needs to use stop at block gap feature while the device can't support the read wait feature.

Table continues on the next page...

Field	Function
19 IABG	Interrupt at block gap This field is valid only in 4-bit mode, of the SDIO card, and selects a sample point in the interrupt cycle. Setting to 1 enables interrupt detection at the block gap for a multiple block transfer. Setting to 0 disables interrupt detection during a multiple block transfer. If the SDIO card cannot signal an interrupt during a multiple block transfer, this field should be set to 0 to avoid an inadvertent interrupt. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card. 0b - Disables interrupt at block gap 1b - Enables interrupt at block gap
18 RWCTL	Read wait control The read wait function is optional for SDIO cards. If the card supports read wait, set this field to enable use of the read wait protocol to stop read data using the DATA2 line. Otherwise, uSDHC has to stop the SD clock to hold read data, which restricts commands generation. When the host driver detects an SDIO card insertion, it sets this field according to the CCCR of the card. If the card does not support read wait, this field should never be set to 1; otherwise, DATA line conflicts might occur. If this field is set to 0, stop at block gap during read operation is also supported, but uSDHC stops the SD clock to pause reading operation. 0b - Disables read wait control and stop SD clock at block gap when SABGREQ field is set 1b - Enables read wait control and assert read wait without stopping SD clock at block gap when SABGREQ field is set
17 CREQ	Continue request This field is used to restart a transaction which was stopped using the stop at block gap request. When a suspend operation is not accepted by the card, it is also by setting this field to restart the paused transfer. To cancel stop at the block gap, set stop at block gap request to 0 and set this field to 1 to restart the transfer. The uSDHC module automatically clears this field, therefore it is not necessary for the host driver to set this field to 0. If both stop at block gap request and this field are 1, the continue request is ignored. 0b - No effect 1b - Restart
16 SABGREQ	Stop at block gap request This field is used to stop executing a transaction at the next block gap for both DMA and non-DMA transfers. Until the transfer complete is set to 1, indicating a transfer completion, the host driver leaves this field set to 1. Clearing both the stop at block gap request and continue request does not cause the transaction to restart. Read Wait is used to stop the read transaction at the block gap. The uSDHC module supports the stop at block gap request for write transfers, but for read transfers it requires that the SDIO card support read wait. Therefore, the host driver does not set this field during read transfers unless the SDIO card supports Read Wait and has set the read wait control to 1; otherwise, uSDHC stops the SD bus clock to pause the read operation during block gap. In the case of write transfers in which the host driver writes data to the Data Port register, the host driver sets this field after all block data is written. If this field is set to 1, the host driver does not write data to the Data Port register after a block is sent. Once this field is set, the host driver does not clear this field before the Transfer Complete field in Interrupt Status register is set, otherwise uSDHC's behavior is undefined. This field effects read transfer active, write transfer active, DATA Line Active and Command Inhibit (DATA) in the Present State register. 0b - Transfer 1b - Stop
15-10 —	Reserved
9-8 DMASEL	DMA select This field is valid while DMA (SDMA or ADMA) is enabled and selects the DMA operation.

Table continues on the next page...

Field	Function
	00b - No DMA or simple DMA is selected. 01b - ADMA1 is selected. 10b - ADMA2 is selected. 11b - Reserved
7 CDSS	Card detect signal selection This field selects the source for the card detection. 0b - Card detection level is selected (for normal purpose). 1b - Card detection test level is selected (for test purpose).
6 CDTL	Card detect test level This bit is enabled while the card detection signal selection is set to 1 and it indicates card insertion. 0b - Card detect test level is 0, no card inserted 1b - Card detect test level is 1, card inserted
5-4 EMODE	Endian mode The uSDHC module supports all three endian modes in data transfer. See Data buffer for more details. 00b - Big endian mode 01b - Half word big endian mode 10b - Little endian mode 11b - Reserved
3 D3CD	DATA3 as card detection pin If this field is set, DATA3 should be pulled down to act as a card detection pin. Be cautious when using this feature, because DATA3 is also a chip-select for the SPI mode. A pull-down on this pin and CMD0 might set the card into the SPI mode, which uSDHC does not support. 0b - DATA3 does not monitor card insertion 1b - DATA3 as card detection pin
2-1 DTW	Data transfer width This field selects the data width of the SD bus for a data transfer. The host driver sets it to match the data width of the card. Possible data transfer width is 1-bit, 4-bits or 8-bits. 00b - 1-bit mode 01b - 4-bit mode 10b - 8-bit mode 11b - Reserved
0 LCTL	LED control This field, fully controlled by the host driver, is used to caution the user not to remove the card while the card is being accessed. If the software is going to issue multiple SD commands, this field can be set during all these transactions. It is not necessary to change for each transaction. When the software issues multiple SD commands, setting the field once before the first command is sufficient: it is not necessary to reset the bit between commands. 0b - LED off 1b - LED on

10.3.7.1.13 System Control (SYS_CTRL)

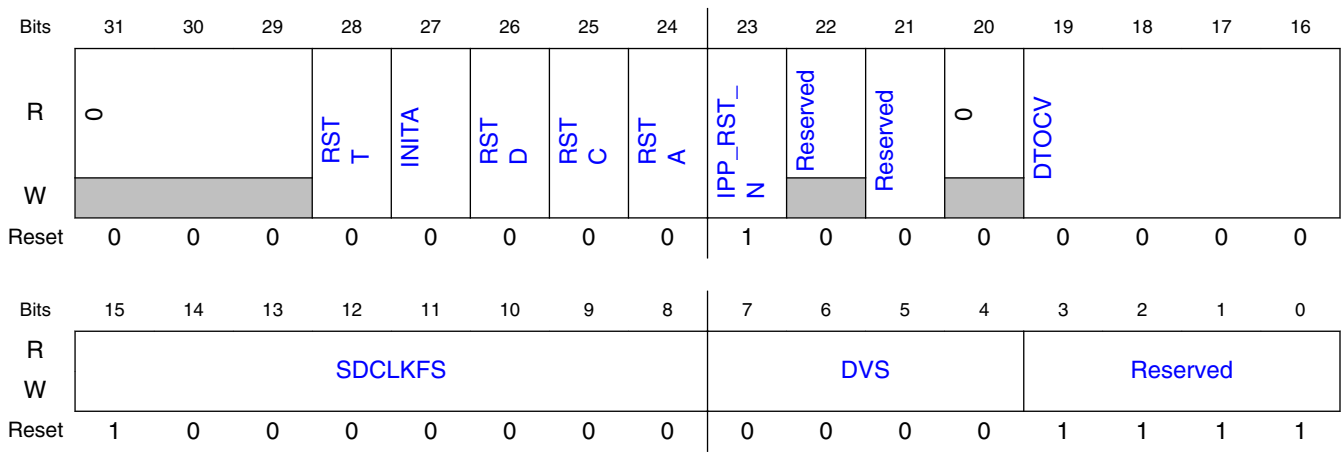
10.3.7.1.13.1 Offset

Register	Offset
SYS_CTRL	2Ch

10.3.7.1.13.2 Function

This register provides control of the system. See detail in the field description.

10.3.7.1.13.3 Diagram



10.3.7.1.13.4 Fields

Field	Function
31-29 —	Reserved
28 RSTT	Reset tuning When set this field to 1, it resets tuning circuit. After tuning circuits are reset, bit value is 0. Clearing execute_tuning field in AUTOCMD12_ERR_STATUS also sets this field to 1 to reset tuning circuit
27 INITA	Initialization active When this field is set, 80 SD-clocks are sent to the card. After the 80 clocks are sent, this field is self cleared. This field is very useful during the card power-up period when 74 SD-clocks are needed and the clock auto gating feature is enabled. Writing 1 to this bit when this field is already 1 has no effect. Writing 0 to this field at any time has no effect. When either of the CIHB and CDIHB fields in the Present State register are set, writing 1 to this field is ignored (that is, when command line or data lines are active, write to this field is not allowed). On the other-hand, when this field is set, that is, during initialization active period, it is allowed to issue command, and the command bit stream appears on the CMD pad after all 80 clock cycles are done. So, when this command ends, the driver can make sure the 80 clock cycles are sent out. This is very useful when the driver needs to send 80 cycles to the card and does not want to wait till this field is self cleared.
26	Software reset for data line

Table continues on the next page...

Field	Function
RSTD	Only part of the data circuit is reset. DMA circuit is also reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Data Port register • Buffer is cleared and initialized • Present State register • Buffer read enable • Buffer write enable • Read transfer active • Write transfer active • DATA line active • Command Inhibit (DATA) Protocol Control register • Continue request • Stop At Block Gap Request Interrupt Status register • Buffer read ready • Buffer write ready • DMA interrupt • Block gap event • Transfer complete **NOTE:** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0b - No reset 1b - Reset
25 RSTC	Software reset for CMD line Only part of the command circuit is reset. After this field is set, the software waits for self-clear. The following registers and bits are cleared by this field: • Present State Register Command Inhibit (CMD) • Interrupt Status register Command Complete 0b - No reset 1b - Reset
24 RSTA	Software reset for all This reset effects the entire host controller except for the card detection circuit. Register bits of type ROC, RW, RW1C, RWAC are cleared. During its initialization, the host driver is set this field to 1 to reset uSDHC. The uSDHC module resets this field to 0 when the capabilities registers are valid and the host driver can read them. Additional use of Software Reset For All does not affect the value of the capabilities registers. After this field is set, it is recommended that the host driver reset the external card and re-initialize it. After this field is set, the software should wait for self-clear. In tuning process, after every CMD19 is finished, this field is set to retest uSDHC. **NOTE:** When reset, the software must make sure there is no incomplete data transferring. If there is data transfer going on, the software needs to wait TC or DC INT_STATUS register is set. 0b - No reset 1b - Reset
23 IPP_RST_N	Hardware reset This register's value is output to card through pad directly to hardware reset pin of the card if the card supports this feature.
22 —	Reserved
21	Reserved

Table continues on the next page...

Field	Function
—	
20 —	Reserved
19-16 DTOCV	Data timeout counter value This value determines the interval by which DAT line timeouts are detected. See the Data Timeout Error field in the Interrupt Status register for information on factors that dictate time-out generation. Time-out clock frequency is generated by dividing the base clock SDCLK value by this value. The host driver can clear the Data Timeout Error Status Enable (in the Interrupt Status Enable register) to prevent inadvertent time-out events. 0000b - SDCLK x 2^{14} 0001b - SDCLK x 2^{15} 0010b - SDCLK x 2^{16} 0011b - SDCLK x 2^{17} 0100b - SDCLK x 2^{18} 0101b - SDCLK x 2^{19} 0110b - SDCLK x 2^{20} 0111b - SDCLK x 2^{21} 1000b - SDCLK x 2^{22} 1001b - SDCLK x 2^{23} 1010b - SDCLK x 2^{24} 1011b - SDCLK x 2^{25} 1100b - SDCLK x 2^{26} 1101b - SDCLK x 2^{27} 1110b - SDCLK x 2^{28} 1111b - SDCLK x 2^{29}
15-8 SDCLKFS	SDCLK frequency select This register is used to select the frequency of the SDCLK pin. The frequency is not programmed directly, rather this register holds the prescaler (this register) and divisor (next register) of the Base Clock Frequency register. In Single Data Rate mode (DDR_EN field of MIXERCTRL is '0') Only the following settings are allowed: 80h) Base clock divided by 256 40h) Base clock divided by 128 20h) Base clock divided by 64 10h) Base clock divided by 32 08h) Base clock divided by 16 04h) Base clock divided by 8 02h) Base clock divided by 4 01h) Base clock divided by 2 00h) Base clock divided by 1 While in Dual Data Rate mode (DDR_EN field of MIXERCTRL is '1') Only the following settings are allowed: 80h) Base clock divided by 512 40h) Base clock divided by 256 20h) Base clock divided by 128

Table continues on the next page...

Field	Function
	10h) Base clock divided by 64 08h) Base clock divided by 32 04h) Base clock divided by 16 02h) Base clock divided by 8 01h) Base clock divided by 4 00h) Base clock divided by 2 When the software changes the DDR_EN field, SDCLKFS might need to be changed also. In Single Data Rate mode, setting 00h bypasses the frequency prescaler of the SD clock. Multiple bits must not be set, or the behavior of this prescaler is undefined. The two default divider values can be calculated by the frequency of ipg_perclk and the following Divisor bits. The frequency of SDCLK is set by the following formula: Clock Frequency = (Base Clock) / (prescaler x divisor) For example, in Single Data Rate mode, if the Base Clock Frequency is 96 MHz, and the target frequency is 25 MHz, then choosing the prescaler value of 01h and divisor value of 1h yields 24 MHz, which is the nearest frequency less than or equal to the target. Similarly, to approach a clock value of 400 kHz, the prescaler value of 08h and divisor value of eh yields the exact clock value of 400 kHz. The reset value of this field is 80h, so if the input Base Clock (ipg_perclk) is about 96 MHz, the default SD clock after reset is 375 kHz. According to the SD Physical Specification Version 1.1 and the SDIO Card Specification Version 1.2, the maximum SD clock frequency is 50 MHz and is never exceed this limit. Before changing clock divisor value (SDCLKFS or DVS), host driver should make sure the SDSTB field is high. If setting SDCLKFS and DVS can generate the same clock frequency,(for example, in SDR mode, SDCLKFS = 01h is same as DVS = 01h.) SDCLKFS is highly recommended.
7-4 DVS	Divisor This register is used to provide a more exact divisor to generate the desired SD clock frequency. Note the divider can even support odd divisors without deterioration of duty cycle. Before changing clock divisor value (SDCLKFS or DVS), Host driver should make sure the SDSTB field is high. The settings are as follows: 0000b - Divide-by-1 0001b - Divide-by-2 1110b - Divide-by-15 1111b - Divide-by-16
3-0 —	Reserved Always write as 1.

10.3.7.1.14 Interrupt Status (INT_STATUS)

10.3.7.1.14.1 Offset

Register	Offset
INT_STATUS	30h

10.3.7.1.14.2 Function

An interrupt is generated when the normal interrupt signal enable is enabled and at least one of the status fields is set to 1. For all fields, writing 1 to a bit clears it; writing to 0 keeps the bit unchanged. More than one status can be cleared with a single register write. For card interrupt, before writing 1 to clear, it is required that the card stops asserting the interrupt, meaning that when the card driver services the interrupt condition; otherwise, the CINT field is asserted again.

The table below shows the relationship between the command timeout error and the command complete.

Table 10-44. uSDHC status for command timeout error/command complete bit combinations

Command CRC Error	Command timeout error	Meaning of the status
0	0	X
X	1	Response not received within 64 SDCLK cycles
1	0	Response received

The table below shows the relationship between the transfer complete and the data timeout error.

Table 10-45. uSDHC status for data timeout error/transfer complete bit combinations

Transfer complete	Data timeout error	Meaning of the status
0	0	X
0	1	Timeout occurred during transfer
1	X	Data transfer complete

The table below shows the relationship between the command CRC error and command timeout error.

Table 10-46. uSDHC status for command CRC error/command timeout error bit combinations

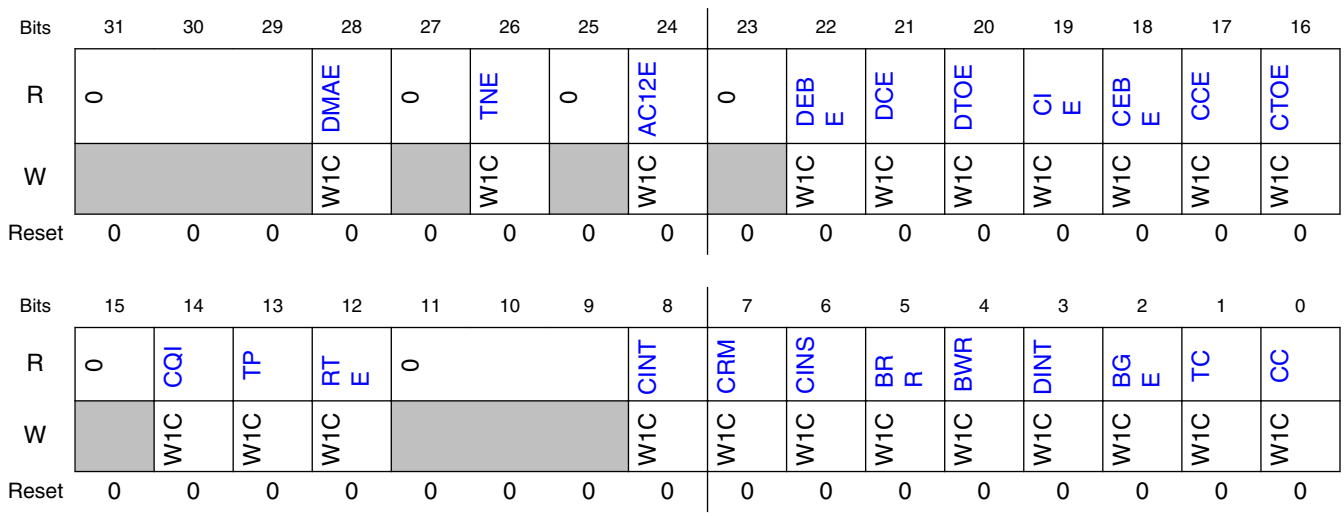
Command complete	Command timeout error	Meaning of the status
0	0	No error

Table continues on the next page...

Table 10-46. uSDHC status for command CRC error/command timeout error bit combinations (continued)

Command complete	Command timeout error	Meaning of the status
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

10.3.7.1.14.3 Diagram



10.3.7.1.14.4 Fields

Field	Function
31-29 —	Reserved
28 DMAE	DMA error Occurs when an Internal DMA transfer has failed. This field is set to 1, when some error occurs in the data transfer. This error can be caused by either simple DMA or ADMA, depending on which DMA is in use. The value in DMA System Address register is the next fetch address where the error occurs. Because any error corrupts the whole data block, the host driver restarts the transfer from the corrupted block boundary. The address of the block boundary can be calculated either from the current DS_ADDR value or from the remaining number of blocks and the block size. 0b - No error 1b - Error
27 —	Reserved
26 TNE	Tuning error: (only for SD3.0 SDR104 mode and EMMC HS200 mode)

Table continues on the next page...

Field	Function
	This field is set when an unrecoverable error is detected in a tuning circuit. By detecting Tuning Error, host driver needs to abort a command executing and perform tuning.
25 —	Reserved
24 AC12E	Auto CMD12 error Occurs when detecting that one of the fields in the Auto CMD12 Error Status register has changed from 0 to 1. This field is set to 1, not only when the errors in Auto CMD12 occur, but also, when the Auto CMD12 is not executed due to the previous command error. 0b - No error 1b - Error
23 —	Reserved
22 DEBE	Data end bit error Occurs either when detecting 0 at the end field position of read data that uses the DATA line, or at the end field position of the CRC. This field is not asserted in tuning process. 0b - No error 1b - Error
21 DCE	Data CRC error Occurs when detecting a CRC error when transferring read data that uses the DATA line, or when detecting the Write CRC status having a value other than 010. This field is not asserted in tuning process. 0b - No error 1b - Error
20 DTOE	Data timeout error Occurs when detecting one of following time-out conditions. • Busy time-out for R1b, R5b type • Busy time-out after Write CRC status • Read Data time-out. This field is not asserted in tuning process. 0b - No error 1b - Time out
19 CIE	Command index error Occurs if a command index error occurs in the command response. This field is not asserted in tuning process. 0b - No error 1b - Error
18 CEBE	Command end bit error Occurs when detecting that the end field of a command response is 0. This field is not asserted in tuning process. 0b - No error 1b - End bit error generated
17	Command CRC error

Table continues on the next page...

Field	Function
CCE	Command CRC Error is generated in two cases. • If a response is returned and the Command Timeout Error is set to 0 (indicating no time-out), this field is set when detecting a CRC error in the command response. • The uSDHC module detects a CMD line conflict by monitoring the CMD line when a command is issued. If uSDHC drives the CMD line to 1, but detects 0 on the CMD line at the next SDCLK edge, then uSDHC aborts the command (Stop driving CMD line) and set this bit to 1. The Command Timeout Error should also be set to 1 to distinguish CMD line conflict. This field is not asserted in tuning process. 0b - No error 1b - CRC error generated
16 CTOE	Command timeout error Occurs only if no response is returned within 64 SDCLK cycles from the end field of the command. If uSDHC detects a CMD line conflict, in which case a Command CRC Error is also be set (as shown in Interrupt Status (INT_STATUS)), this field is set without waiting for 64 SDCLK cycles. This is because the command is aborted by uSDHC. This field is not asserted in tuning process. 0b - No error 1b - Time out
15 —	Reserved
14 CQI	Command queuing interrupt This interrupt is asserted when at least one of the bits in CQIS register is set. This interrupt is cleared only by clearing the source interrupt in CQIS register.
13 TP	Tuning pass:(only for SD3.0 SDR104 mode and EMMC HS200 mode) Current CMD19 transfer is done successfully. That is, current sampling point is correct.
12 RTE	Re-tuning event: (only for SD3.0 SDR104 mode and EMMC HS200 mode) This status is set if Re-Tuning Request in the Present State register changes from 0 to 1. host controller requests host driver to perform re-tuning for next data transfer. Current data transfer (not large block count) can be completed without re-tuning. 0b - Re-tuning is not required. 1b - Re-tuning should be performed.
11-9 —	Reserved
8 CINT	Card interrupt This status field is set when an interrupt signal is detected from the external card. In 1-bit mode, uSDHC detects the Card Interrupt without the SD clock to support wakeup. In 4-bit mode, the card interrupt signal is sampled during the interrupt cycle, so the interrupt from card can only be sampled during interrupt cycle, introducing some delay between the interrupt signal from the SDIO card and the interrupt to the host system. Writing this field to 1 can clear this field, but as the interrupt source from the SDIO card does not clear, this field is set again. to clear this field, it is required to reset the interrupt source from the external card followed by a writing 1 to this field. When this status has been set, and the host driver needs to service this interrupt, the Card Interrupt Signal Enable in the Interrupt Signal Enable register should be 0 to stop driving the interrupt signal to the host system. After completion of the card interrupt service (It should reset the interrupt sources in the SDIO card and the interrupt signal might not be asserted), write 1 to clear this field, set the Card Interrupt Signal Enable to 1, and start sampling the interrupt signal again. 0b - No card interrupt

Table continues on the next page...

Field	Function
	1b - Generate card interrupt
7 CRM	Card removal This status field is set if the Card Inserted field in the Present State register changes from 1 to 0. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if no card is inserted. to leave it cleared, clear the Card Removal Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or inserted 1b - Card removed
6 CINS	Card insertion This status field is set if the Card Inserted field in the Present State register changes from 0 to 1. When the host driver writes this field to 1 to clear this status, the status of the Card Inserted in the Present State register should be confirmed. Because the card state might possibly be changed when the host driver clears this field and the interrupt event might not be generated. When this field is cleared, it is set again if a card is inserted. to leave it cleared, clear the Card Inserted Status Enable field in Interrupt Status Enable register. 0b - Card state unstable or removed 1b - Card inserted
5 BRR	Buffer read ready This status field is set if the Buffer Read Enable field, in the Present State register, changes from 0 to 1. See the Buffer Read Enable field in the Present State register for additional information. This field indicates that cmd19 is finished in tuning process. 0b - Not ready to read buffer 1b - Ready to read buffer
4 BWR	Buffer write ready This status field is set if the Buffer Write Enable field, in the Present State register, changes from 0 to 1. See the Buffer Write Enable field in the Present State register for additional information. 0b - Not ready to write buffer 1b - Ready to write buffer
3 DINT	DMA interrupt Occurs only when the internal DMA finishes the data transfer successfully. Whenever errors occur during data transfer, this field does not be set. Instead, the DMAE field is set. Either Simple DMA or ADMA finishes data transferring, this field is set. 0b - No DMA interrupt 1b - DMA interrupt is generated.
2 BGE	Block gap event If the Stop At Block Gap Request field in the Protocol Control register is set, this field is set when a read or write transaction is stopped at a block gap. If Stop At Block Gap Request is not set to 1, this field is not set to 1. In the case of a Read Transaction: This field is set at the falling edge of the DATA Line Active Status (When the transaction is stopped at SD bus timing). The Read Wait must be supported to use this function. In the case of Write Transaction: This field is set at the falling edge of Write Transfer Active Status (After getting CRC status at SD Bus timing). 0b - No block gap event 1b - Transaction stopped at block gap

Table continues on the next page...

Field	Function
1 TC	Transfer complete This field is set when a read or write transfer is completed. In the case of a Read Transaction: This field is set at the falling edge of the Read Transfer Active Status. There are two cases in which this interrupt is generated. The first is when a data transfer is completed as specified by the data length (after the last data has been read to the host system). The second is when data has stopped at the block gap and completed the data transfer by setting the Stop At Block Gap Request field in the Protocol Control register (after valid data has been read to the host system). In the case of a Write Transaction: This field is set at the falling edge of the DATA Line Active Status. There are two cases in which this interrupt is generated. The first is when the last data is written to the SD card as specified by the data length and the busy signal is released. The second is when data transfers are stopped at the block gap, by setting the Stop At Block Gap Request field in the Protocol Control register, and the data transfers are completed. (after valid data is written to the SD card and the busy signal released). In the case of a command with busy, this field is set when busy is deasserted. This field is not asserted in tuning process. 0b - Transfer does not complete 1b - Transfer complete
0 CC	Command complete This field is set when you receive the end field of the command response (except auto CMD12). See the Command Inhibit (CMD) in the Present State register. This field is not asserted in tuning process. 0b - Command not complete 1b - Command complete

10.3.7.1.15 Interrupt Status Enable (INT_STATUS_EN)

10.3.7.1.15.1 Offset

Register	Offset
INT_STATUS_EN	34h

10.3.7.1.15.2 Function

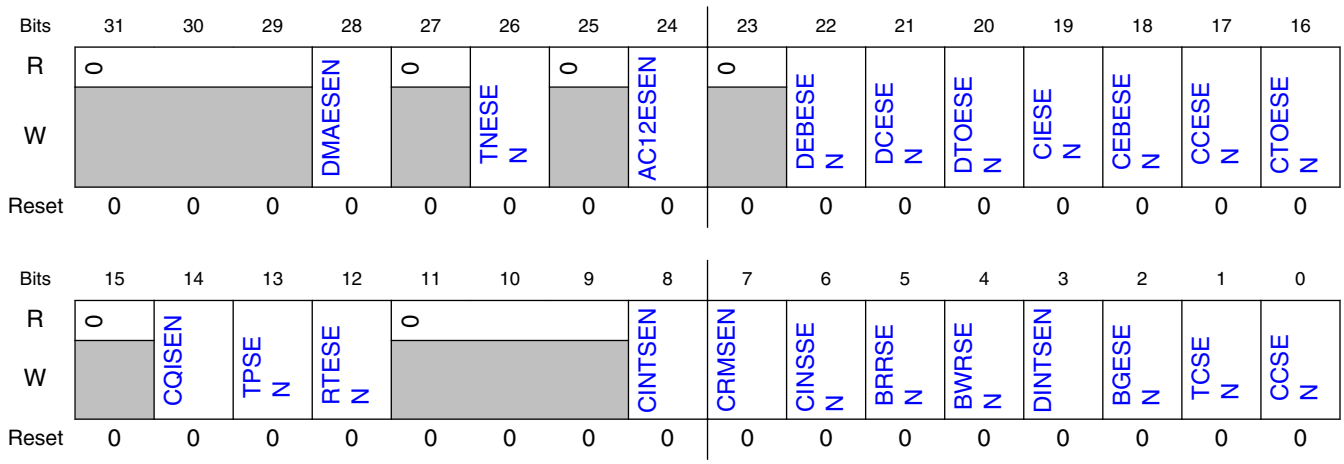
Setting the fields in this register to 1 enables the corresponding Interrupt Status to be set by the specified event. If any field is cleared, the corresponding Interrupt Status field is also cleared (that is, when the field in this register is cleared, the corresponding field in Interrupt Status register is always 0).

- Depending on IABG field setting, uSDHC might be programmed to sample the card interrupt signal during the interrupt period and hold its value in the flip-flop. There

are some delays on the Card Interrupt, asserted from the card, to the time the host system is informed.

- To detect a CMD line conflict, the host driver must set both Command Timeout Error Status Enable and Command CRC Error Status Enable to 1.

10.3.7.1.15.3 Diagram



10.3.7.1.15.4 Fields

Field	Function
31-29 —	Reserved
28 DMAESEN	DMA error status enable 0b - Masked 1b - Enabled
27 —	Reserved
26 TNESEN	Tuning error status enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12ESEN	Auto CMD12 error status enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBESEN	Data end bit error status enable 0b - Masked 1b - Enabled

Table continues on the next page...

Field	Function
21 DCESEN	Data CRC error status enable 0b - Masked 1b - Enabled
20 DTOESEN	Data timeout error status enable 0b - Masked 1b - Enabled
19 CIESEN	Command index error status enable 0b - Masked 1b - Enabled
18 CEBESEN	Command end bit error status enable 0b - Masked 1b - Enabled
17 CCESEN	Command CRC error status enable 0b - Masked 1b - Enabled
16 CTOESEN	Command timeout error status enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQISEN	Command queuing status enable 0b - Masked 1b - Enabled
13 TPSEN	Tuning pass status enable 0b - Masked 1b - Enabled
12 RTESEN	Re-tuning event status enable 0b - Masked 1b - Enabled
11-9 —	Reserved
8 CINTSEN	Card interrupt status enable If this field is set to 0, uSDHC clears the interrupt request to the system. The Card Interrupt detection is stopped when this field is cleared and restarted when this field is set to 1. The host driver should clear the Card Interrupt Status Enable before servicing the Card Interrupt and should set this field again after all interrupt requests from the card are cleared to prevent inadvertent interrupts. 0b - Masked 1b - Enabled
7 CRMSEN	Card removal status enable 0b - Masked 1b - Enabled
6 CINSSEN	Card insertion status enable 0b - Masked 1b - Enabled
5 BRRSEN	Buffer read ready status enable 0b - Masked 1b - Enabled
4	Buffer write ready status enable 0b - Masked

Table continues on the next page...

Field	Function
BWRSEN	1b - Enabled
3 DINTSEN	DMA interrupt status enable 0b - Masked 1b - Enabled
2 BGESEN	Block gap event status enable 0b - Masked 1b - Enabled
1 TCSEN	Transfer complete status enable 0b - Masked 1b - Enabled
0 CCSEN	Command complete status enable 0b - Masked 1b - Enabled

10.3.7.1.16 Interrupt Signal Enable (INT_SIGNAL_EN)

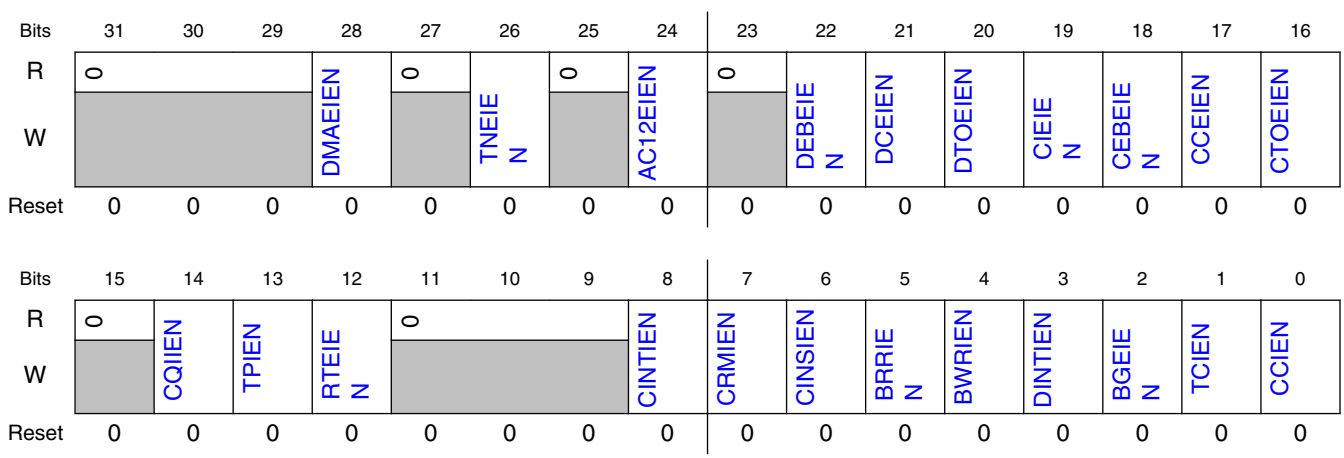
10.3.7.1.16.1 Offset

Register	Offset
INT_SIGNAL_EN	38h

10.3.7.1.16.2 Function

This register is used to select which interrupt status is indicated to the host system as the interrupt. These status fields all share the same interrupt line. Setting any of these fields to 1 enables interrupt generation. The corresponding Status register field generates an interrupt when the corresponding interrupt signal enable field is set.

10.3.7.1.16.3 Diagram



10.3.7.1.16.4 Fields

Field	Function
31-29 —	Reserved
28 DMAEIEN	DMA error interrupt enable 0b - Masked 1b - Enable
27 —	Reserved
26 TNEIEN	Tuning error interrupt enable 0b - Masked 1b - Enabled
25 —	Reserved
24 AC12EIEN	Auto CMD12 error interrupt enable 0b - Masked 1b - Enabled
23 —	Reserved
22 DEBEIEN	Data end bit error interrupt enable 0b - Masked 1b - Enabled
21 DCEIEN	Data CRC error interrupt enable 0b - Masked 1b - Enabled
20 DTOEIEN	Data timeout error interrupt enable 0b - Masked 1b - Enabled
19 CIEIEN	Command index error interrupt enable 0b - Masked 1b - Enabled
18 CEBEIEN	Command end bit error interrupt enable 0b - Masked 1b - Enabled
17 CCEIEN	Command CRC error interrupt enable 0b - Masked 1b - Enabled
16 CTOEIEN	Command timeout error interrupt enable 0b - Masked 1b - Enabled
15 —	Reserved
14 CQIIEN	Command queuing signal enable 0b - Masked 1b - Enabled

Table continues on the next page...

Field	Function
13 TPIEN	Tuning pass interrupt enable 0b - Masked 1b - Enabled
12 RTEIEN	Re-tuning event interrupt enable 0b - Masked 1b - Enabled
11-9 —	Reserved
8 CINTIEN	Card interrupt enable 0b - Masked 1b - Enabled
7 CRMIEN	Card removal interrupt enable 0b - Masked 1b - Enabled
6 CINSIEN	Card insertion interrupt enable 0b - Masked 1b - Enabled
5 BRRIEN	Buffer read ready interrupt enable 0b - Masked 1b - Enabled
4 BWRIEN	Buffer write ready interrupt enable 0b - Masked 1b - Enabled
3 DINTIEN	DMA interrupt enable 0b - Masked 1b - Enabled
2 BGEIEN	Block gap event interrupt enable 0b - Masked 1b - Enabled
1 TCIEN	Transfer complete interrupt enable 0b - Masked 1b - Enabled
0 CCIEN	Command complete interrupt enable 0b - Masked 1b - Enabled

10.3.7.1.17 Auto CMD12 Error Status (AUTOCMD12_ERR_STATUS)

10.3.7.1.17.1 Offset

Register	Offset
AUTOCMD12_ERR_STATUS	3Ch

10.3.7.1.17.2 Function

When the Auto CMD12 Error Status field in the Status register is set, the host driver checks this register to identify what kind of error the Auto CMD12 / CMD 23 indicated. Auto CMD23 errors are indicated in field 04-01. This register is valid only when the Auto CMD12 Error status field is set.

The table below shows the relationship between the Auto CMGD12 CRC Error and the Auto CMD12 Command Timeout Error.

Table 10-47. Relationship between command CRC error and command timeout error for auto CMD12

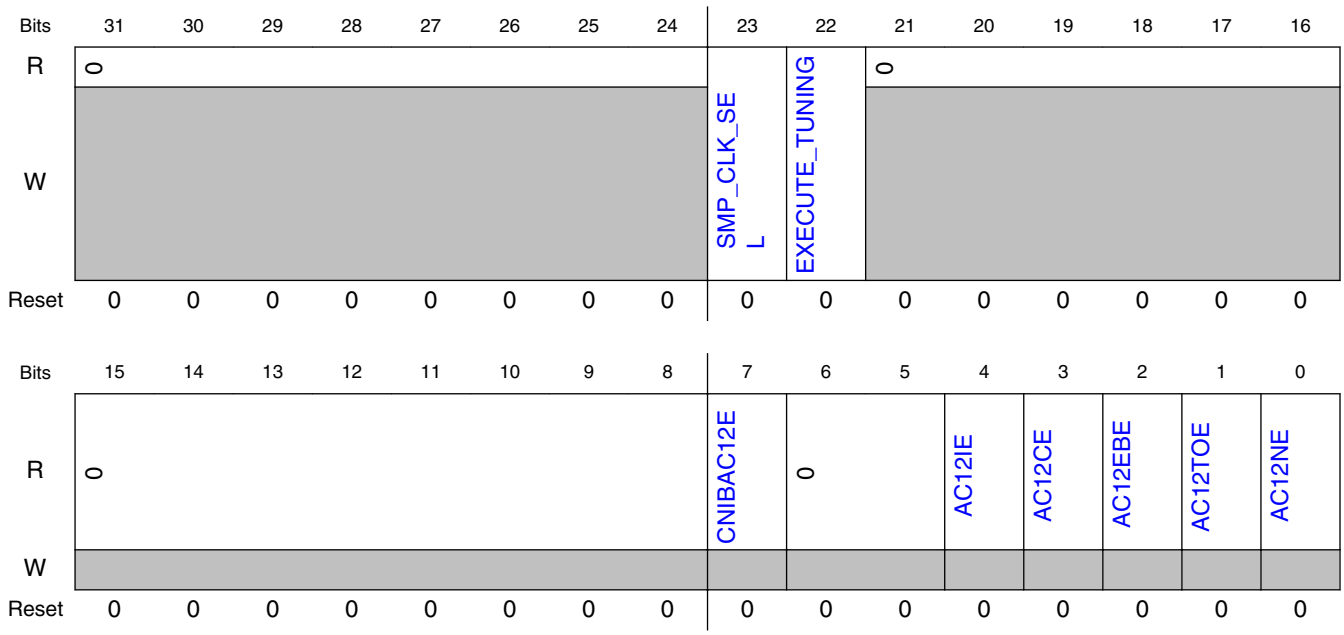
Auto CMD12 CRC error	Auto CMD12 timeout error	Type of error
0	0	No error
0	1	Response timeout error
1	0	Response CRC error
1	1	CMD line conflict

Changes in Auto CMD12 Error Status register can be classified in three scenarios:

- When uSDHC is going to issue an Auto CMD12
  - Set field 0 to 1 if the Auto CMD12 cannot be issued due to an error in the previous command
  - Set field 0 to 0 if the Auto CMD12 is issued
- At the end field of an Auto CMD12 response
  - Check errors correspond to fields 1-4
  - Set fields 1-4 corresponding to detected errors
  - Clear fields 1-4 corresponding to detected errors
- Before reading the Auto CMD12 Error Status field 7
  - Set field 7 to 1 if there is a command that can't be issued
  - Clear field 7 if there is no command to issue

The timing for generating the Auto CMD12 Error and writing to the Command register are asynchronous. After that, field 7 is sampled when the driver is not writing to the Command register. So, it is suggested to read this register only when the AC12E field in Interrupt Status register is set. An Auto CMD12 Error Interrupt is generated when one of the error fields (0-4) is set to 1. The Command Not Issued By Auto CMD12 Error does not generate an interrupt.

10.3.7.1.17.3 Diagram



10.3.7.1.17.4 Fields

Field	Function
31-24 —	Reserved
23 SMP_CLK_SEL	Sample clock select When std_tuning_en field is set, this field is used to select sampling clock to receive CMD and DATA. Otherwise, this field is reserved. This field is set by ty tuning procedure and valid after the completion of tuning(When Execute Tuning is cleared). Setting 1 means that tuning is completed successfully and setting 0 means that tuning is failed. Writing 1 to this field is meaningless and ignored. A tuning circuit is reset by writing to 0. This field can be cleared with setting Execute Tuning. Once the tuning circuit is reset, it takes time to complete tuning sequence. Therefore, host driver should keep this field to 1 to perform re-tuning sequence to complete re-tuning sequence in a short time. Change of this field is not allowed while the Host controller us receiving response or a read data block. 0b - Fixed clock is used to sample data 1b - Tuned clock is used to sample data
22 EXECUTE_TUNING	Execute tuning When std_tuning_en field is set, this field is used to start tuning procedure. Otherwise, this field is reserved. This field is set to start tuning procedure and automatically cleared when runing procedure is completed. The result of tuning is indicated to sam_clk_sel field. Tuning procedure is aborted by writing 0.
21-8 —	Reserved
7 CNIBAC12E	Command not issued by Auto CMD12 error Setting this field to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 error (D04-D01) in this register.

Table continues on the next page...

Field	Function
	0b - No error 1b - Not issued
6-5 —	Reserved
4 AC12IE	Auto CMD12 / 23 index error Occurs if the command index error occurs in response to a command. 0b - No error 1b - Error, the CMD index in response is not CMD12/23
3 AC12CE	Auto CMD12 / 23 CRC error Occurs when detecting a CRC error in the command response. 0b - No CRC error 1b - CRC error met in Auto CMD12/23 response
2 AC12EBE	Auto CMD12 / 23 end bit error Occurs when detecting that the end field of command response is 0 which should be 1. 0b - No error 1b - End bit error generated
1 AC12TOE	Auto CMD12 / 23 timeout error Occurs if no response is returned within 64 SDCLK cycles from the end field of the command. If this field is set to1, the other error status fields (2-4) have no meaning. 0b - No error 1b - Time out
0 AC12NE	Auto CMD12 not executed If memory multiple block data transfer is not started, due to a command error, this field is not set because it is not necessary to issue an Auto CMD12. Setting this field to 1 means uSDHC cannot issue the Auto CMD12 to stop a memory multiple block data transfer due to some error. If this field is set to 1, other error status fields (1-4) have no meaning. 0b - Executed 1b - Not executed

10.3.7.1.18 Host Controller Capabilities (HOST_CTRL_CAP)

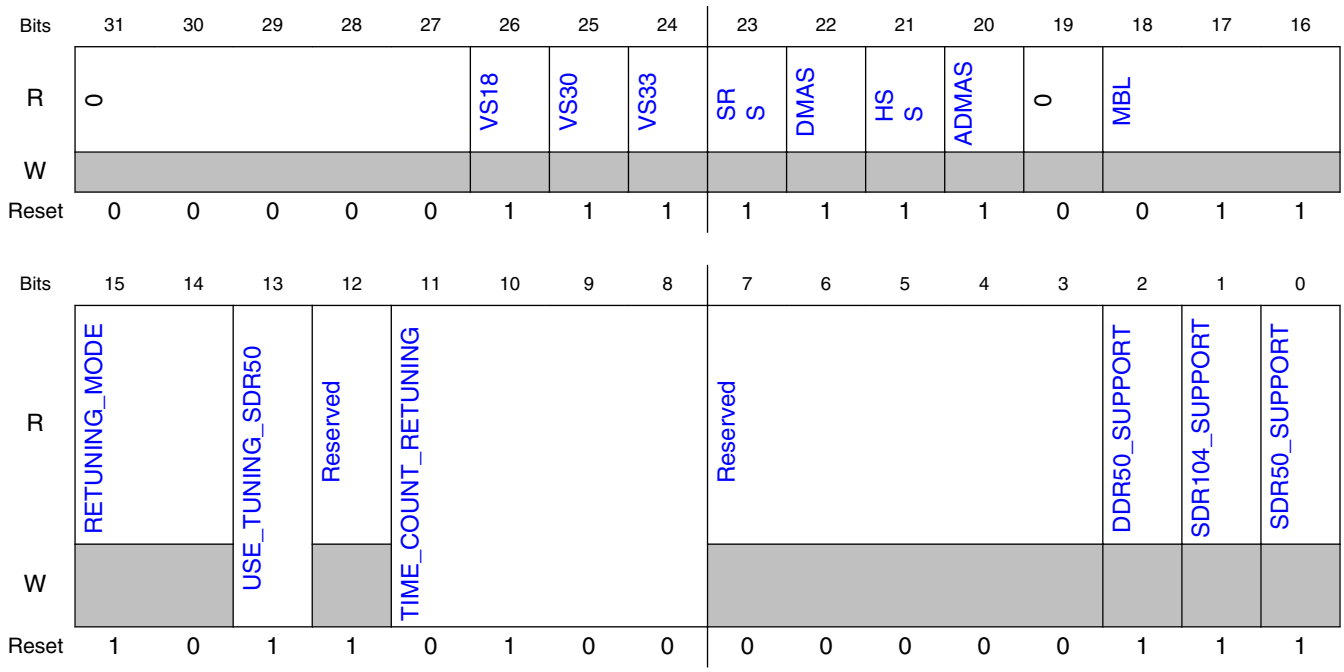
10.3.7.1.18.1 Offset

Register	Offset
HOST_CTRL_CAP	40h

10.3.7.1.18.2 Function

This register provides the host driver with information specific to uSDHC implementation. The value in this register is the power-on-reset value and does not change with a software reset.

10.3.7.1.18.3 Diagram



10.3.7.1.18.4 Fields

Field	Function
31-27 —	Reserved
26 VS18	Voltage support 1.8 V This field depends on the host system ability. 0b - 1.8 V not supported 1b - 1.8 V supported
25 VS30	Voltage support 3.0 V This field depends on the host system ability. 0b - 3.0 V not supported 1b - 3.0 V supported
24 VS33	Voltage support 3.3 V This field depends on the host system ability. 0b - 3.3 V not supported 1b - 3.3 V supported
23 SRS	Suspend / resume support This field indicates whether uSDHC supports Suspend / Resume functionality. If this field is 0, the Suspend and Resume mechanism, as well as the read wait, are not supported, and the host driver does not issue either Suspend or Resume commands. 0b - Not supported

Table continues on the next page...

Field	Function
	1b - Supported
22 DMAS	DMA support This field indicates whether uSDHC is capable of using the internal DMA to transfer data between system memory and the data buffer directly. 0b - DMA not supported 1b - DMA supported
21 HSS	High speed support This field indicates whether uSDHC supports High Speed mode and the host system can supply a SD clock frequency from 25 MHz to 50 MHz. 0b - High speed not supported 1b - High speed supported
20 ADMAS	ADMA support This field indicates whether uSDHC supports the ADMA feature. 0b - Advanced DMA not supported 1b - Advanced DMA supported
19 —	Reserved
18-16 MBL	Max block length This value indicates the maximum block size that the host driver can read and write to the buffer in uSDHC. The buffer transfers block size without wait cycles. 000b - 512 bytes 001b - 1024 bytes 010b - 2048 bytes 011b - 4096 bytes
15-14 RETUNING_MODE	Retuning Mode This field selects retuning method. 00b - Mode 1 01b - Mode 2 10b - Mode 3 11b - Reserved
13 USE_TUNING_SDR50	Use Tuning for SDR50 This field is set to 1. Host controller requires tuning to operate SDR50. 0b - SDR does not require tuning. 1b - SDR50 requires tuning.
12 —	Reserved
11-8 TIME_COUNT_RETUNING	Time counter for retuning This field indicates an initial value of the Retuning Timer for Re-Tuning Mode1 and 3.Setting to 0 disables Retuning Timer.
7-3 —	Reserved
2 DDR50_SUPPORT	DDR50 support This field indicates support of DDR50 mode.

Table continues on the next page...

Field	Function
1 SDR104_SUPPORT	SDR104 support This field indicates support of SDR104 mode.
0 SDR50_SUPPORT	SDR50 support This field indicates support of SDR50 mode.

10.3.7.1.19 Watermark Level (WTMK_LVL)

10.3.7.1.19.1 Offset

Register	Offset
WTMK_LVL	44h

10.3.7.1.19.2 Function

Both write and read watermark levels (FIFO threshold) are configurable. Their value can range from 1 to 128 words. Both write and read burst lengths are also Configurable. There value can range from 1 to 31 words.

10.3.7.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WR_BRST_LEN					WR_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			RD_BRST_LEN					RD_WML							
W																
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0

10.3.7.1.19.4 Fields

Field	Function
31-29 —	Reserved

Table continues on the next page...

Field	Function
28-24 WR_BRST_LEN	Write burst length due to system restriction, the actual burst length might not exceed 16 The number of words uSDHC writes in a single burst. The write burst length must be less than or equal to the write watermark level, and all bursts within a watermark level transfer is in back-to-back mode. On reset, this field is 8. Writing 0 to this field results in '01000' (that is, it is not able to clear this field).
23-16 WR_WML	Write watermark level The number of words used as the watermark level (FIFO threshold) in a DMA write operation. Also, the number of words as a sequence of write bursts in back-to-back mode. The maximum legal value for the write watermark level is 128.
15-13 —	Reserved
12-8 RD_BRST_LEN	Read burst length due to system restriction, the actual burst length might not exceed 16 The number of words uSDHC reads in a single burst. The read burst length must be less than or equal to the read watermark level, and all bursts within a watermark level transfer is in back-to-back mode. On reset, this field is 8. Writing 0 to this field results in '01000' (that is, it is not able to clear this field).
7-0 RD_WML	Read watermark level The number of words used as the watermark level (FIFO threshold) in a DMA read operation. Also, the number of words as a sequence of read bursts in back-to-back mode. The maximum legal value for the read water mark level is 128.

10.3.7.1.20 Mixer Control (MIX_CTRL)

10.3.7.1.20.1 Offset

Register	Offset
MIX_CTRL	48h

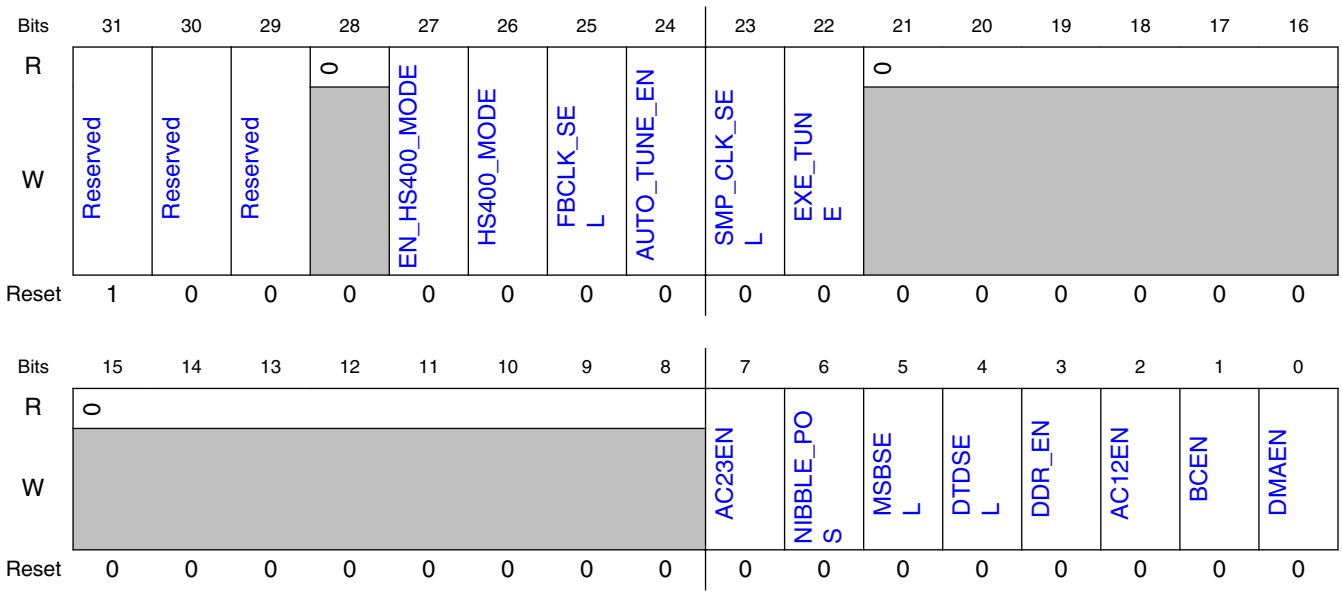
10.3.7.1.20.2 Function

This register is used to DMA and data transfer. To prevent data loss, The software should check if data transfer is active before writing this register. These fields are DPSEL, MBSEL, DTDSEL, AC12EN, BCEN, and DMAEN.

Table 10-48. Transfer type register setting for various transfer types

Multi/single block select	Block count enable	Block count	Function
0	Do not care	Do not care	Single transfer
1	0	Do not care	Infinite transfer
1	1	Positive number	Multiple transfer
1	1	Zero	No data transfer

10.3.7.1.20.3 Diagram



10.3.7.1.20.4 Fields

Field	Function
31 —	Reserved Always write as 1.
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 0.
28 —	Reserved
27 EN_HS400_MODE	Enable enhance HS400 mode Enhance HS400 enable
26 HS400_MODE	Enable HS400 mode HS400 Enable
25 FBCLK_SEL	Feedback clock source selection (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) 0b - Feedback clock comes from the loopback CLK 1b - Feedback clock comes from the ipp_card_clk_out
24 AUTO_TUNE_EN	Auto tuning enable (Only used for SD3.0, SDR104 mode and and EMMC HS200 mode) 0b - Disable auto tuning 1b - Enable auto tuning
23 SMP_CLK_SEL	Clock selection When STD_TUNING_EN is 0, this field is used to select Tuned clock or Fixed clock to sample data / cmd (Only used for SD3.0, SDR104 mode and EMMC HS200 mode)

Table continues on the next page...

Field	Function
	0b - Fixed clock is used to sample data / cmd 1b - Tuned clock is used to sample data / cmd
22 EXE_TUNE	Execute tuning: (Only used for SD3.0, SDR104 mode and EMMC HS200 mode) When STD_TUNING_EN is 0, this field is set to 1 to indicate the host driver is starting tuning procedure. Tuning procedure is aborted by writing 0. 0b - Not tuned or tuning completed 1b - Execute tuning
21-8 —	Reserved
7 AC23EN	Auto CMD23 enable When this field is set to 1, the host controller issues a CMD23 automatically before issuing a command specified in the Command Register.
6 NIBBLE_POS	Nibble position indication In DDR 4-bit mode nibble position indication. 0- the sequence is 'odd high nibble -> even high nibble -> odd low nibble -> even low nibble'; 1- the sequence is 'odd high nibble -> odd low nibble -> even high nibble -> even low nibble'.
5 MSBSEL	Multi / Single block select This field enables multiple block DATA line data transfers. For any other commands, this field can be set to 0. If this field is 0, it is not necessary to set the Block Count register. See Command Transfer Type (CMD_XFR_TYP). 0b - Single block 1b - Multiple blocks
4 DTDSEL	Data transfer direction select This field defines the direction of DATA line data transfers. The field is set to 1 by the host driver to transfer data from the SD card to uSDHC and is set to 0 for all other commands. 0b - Write (Host to card) 1b - Read (Card to host)
3 DDR_EN	Dual data rate mode selection
2 AC12EN	Auto CMD12 enable Multiple block transfers for memory require a CMD12 to stop the transaction. When this field is set to 1, uSDHC issues a CMD12 automatically when the last block transfer has completed. The host driver is not set this field to issue commands that do not require CMD12 to stop a multiple block data transfer. In particular, secure commands defined in File Security Specification (see reference list) do not require CMD12. In single block transfer, uSDHC ignores this field no matter it is set or not. 0b - Disable 1b - Enable
1 BCEN	Block count enable This field is used to enable the Block Count register, which is only relevant for multiple block transfers. When this field is 0, the internal counter for block is disabled, which is useful in executing an infinite transfer. 0b - Disable 1b - Enable
0 DMAEN	DMA enable

Field	Function
	This field enables DMA functionality. If this field is set to 1, a DMA operation begins when the host driver sets the DPSEL field of this register. Whether the simple DMA or the advanced DMA is active depends on the DMA Select field of the Protocol Control register. 0b - Disable 1b - Enable

10.3.7.1.21 Force Event (FORCE_EVENT)

10.3.7.1.21.1 Offset

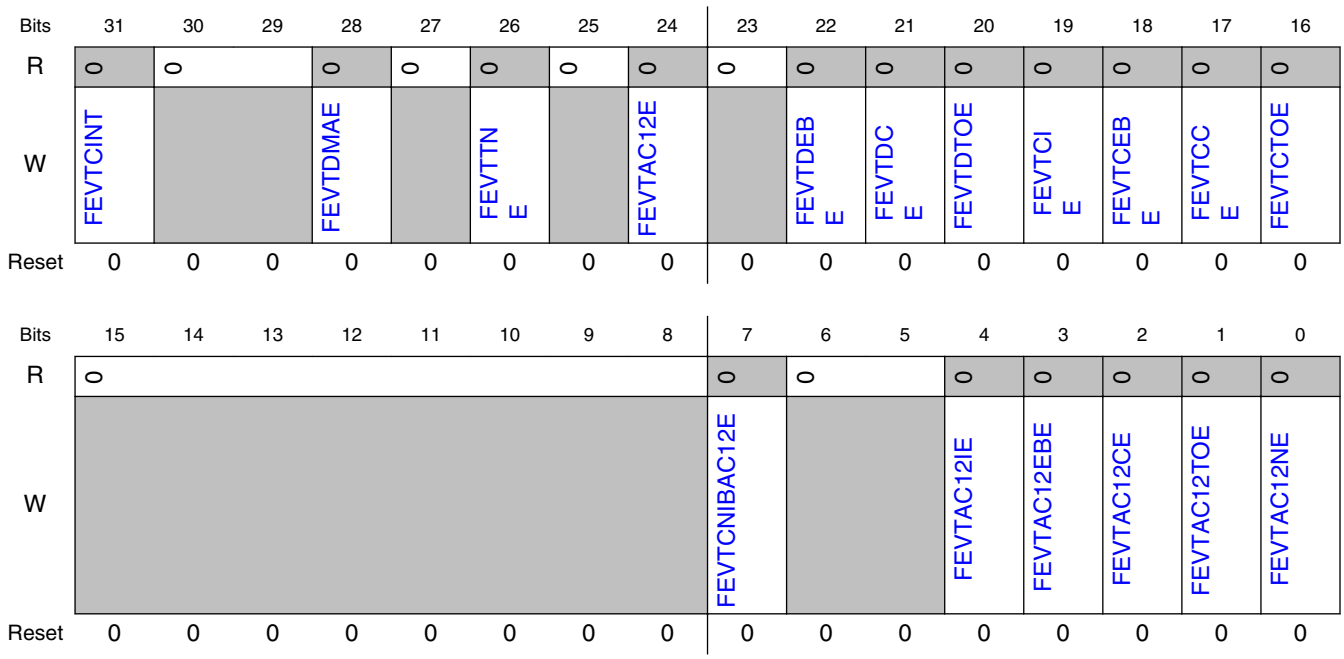
Register	Offset
FORCE_EVENT	50h

10.3.7.1.21.2 Function

The Force Event Register is not a physically implemented register. Rather, it is an address at which the Interrupt Status register can be written if the corresponding field of the Interrupt Status Enable Register is set. This register is a write only register and writing 0 to it has no effect. Writing 1 to this register actually sets the corresponding field of Interrupt Status register. A read from this register always results in 0's. to change the corresponding status fields in the Interrupt Status register, make sure to set IPGEN field in System Control Register so that peripheral clock is always active.

Forcing a card interrupt generates a short pulse on the DATA1 line, and the driver might treat this interrupt as a normal interrupt. The interrupt service routine might skip polling the card interrupt factor as the interrupt is self cleared.

10.3.7.1.21.3 Diagram



10.3.7.1.21.4 Fields

Field	Function
31 FEVTCINT	Force event card interrupt Writing 1 to this field generates a short low-level pulse on the internal DATA1 line, as if a self clearing interrupt was received from the external card. If enabled, the CINT field is set and the interrupt service routine might treat this interrupt as a normal interrupt from the external card.
30-29 —	Reserved
28 FEVTDMAE	Force event DMA error Forces the DMAE field of Interrupt Status register to be set.
27 —	Reserved
26 FEVTTNE	Force tuning error Forces the TNE field of Interrupt Status register to be set.
25 —	Reserved
24 FEVTAC12E	Force event Auto Command 12 error Forces the AC12E field of Interrupt Status register to be set.
23 —	Reserved
22	Force event data end bit error

Table continues on the next page...

Field	Function
FEVTDEBE	Forces the DEBE field of Interrupt Status register to be set.
21 FEVTDCE	Force event data CRC error Forces the DCE field of Interrupt Status register to be set.
20 FEVTDTOE	Force event data time out error Force the DTOE field of Interrupt Status register to be set.
19 FEVTCIE	Force event command index error Forces the CCE field of Interrupt Status register to be set.
18 FEVTCEBE	Force event command end bit error Forces the CEBE field of Interrupt Status register to be set.
17 FEVTCCE	Force event command CRC error Forces the CCE field of Interrupt Status register to be set.
16 FEVTCTOE	Force event command time out error Forces the CTOE field of Interrupt Status register to be set.
15-8 —	Reserved
7 FEVTCNIBAC12E	Force event command not executed by Auto Command 12 error Forces the CNIBAC12E field in the Auto Command12 Error Status register to be set.
6-5 —	Reserved
4 FEVTAC12IE	Force event Auto Command 12 index error Forces the AC12IE field in the Auto Command12 Error Status register to be set.
3 FEVTAC12EBE	Force event Auto Command 12 end bit error Forces the AC12EBE field in the Auto Command12 Error Status register to be set.
2 FEVTAC12CE	Force event auto command 12 CRC error Forces the AC12CE field in the Auto Command12 Error Status register to be set.
1 FEVTAC12TOE	Force event auto command 12 time out error Forces the AC12TOE field in the Auto Command12 Error Status register to be set.
0 FEVTAC12NE	Force event auto command 12 not executed Forces the AC12NE field in the Auto Command12 Error Status register to be set.

10.3.7.1.22 ADMA Error Status (ADMA_ERR_STATUS)

10.3.7.1.22.1 Offset

Register	Offset
ADMA_ERR_STATUS	54h

10.3.7.1.22.2 Function

When an ADMA Error Interrupt has occurred, the ADMA Error States field in this register holds the ADMA state and the ADMA System Address register holds the address around the error descriptor.

For recovering from this error, the host driver requires the ADMA state to identify the error descriptor address as follows:

- ST_STOP: Previous location set in the ADMA System Address register is the error descriptor address.
- ST_FDS: Current location set in the ADMA System Address register is the error descriptor address.
- ST_CADR: This state is never set because it only increments the descriptor pointer and does not generate an ADMA error.
- ST_TFR: Previous location set in the ADMA System Address register is the error descriptor address.

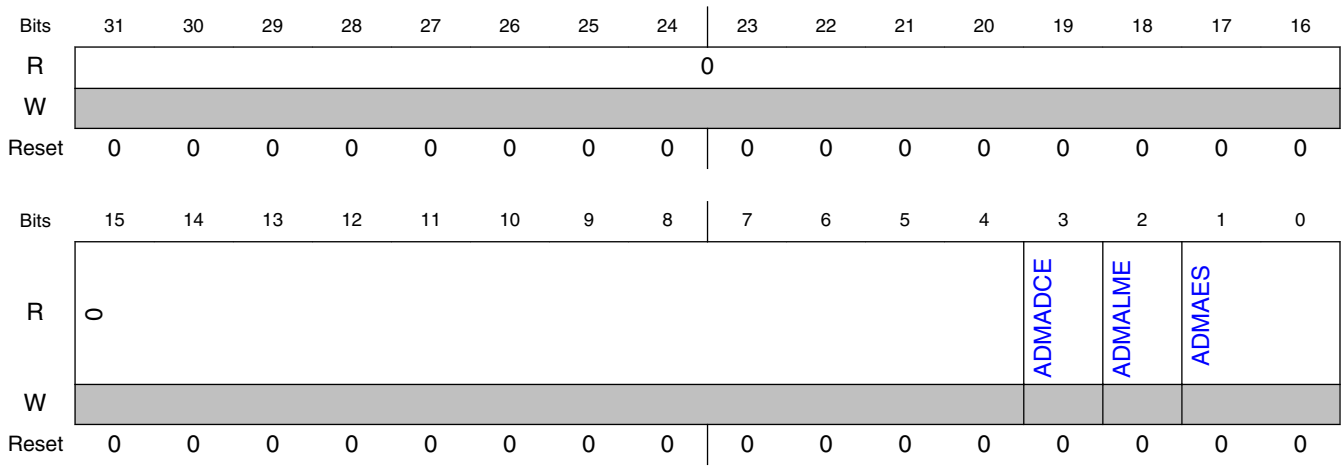
In case of a write operation, the host driver should use the ACMD22 to get the number of the written block, rather than using this information, because unwritten data might exist in the host controller.

The host controller generates the ADMA Error Interrupt when it detects invalid descriptor data (Valid=0) in the ST_FDS state. The host driver can distinguish this error by reading the Valid field of the error descriptor.

Table 10-49. ADMA error state coding

D01-D00	ADMA error state (when error has occurred)	Contents of ADMA System Address register
00	ST_STOP (Stop DMA)	Holds the address of the next executable descriptor command
01	ST_FDS (Fetch descriptor)	Holds the valid descriptor address
10	ST_CADR (Change address)	No ADMA error is generated
11	ST_TFR (Transfer data)	Holds the address of the next executable descriptor command

10.3.7.1.22.3 Diagram



10.3.7.1.22.4 Fields

Field	Function
31-4 —	Reserved
3 ADMADCE	ADMA descriptor error This error occurs when invalid descriptor fetched by ADMA. 0b - No error 1b - Error
2 ADMALME	ADMA length mismatch error This error occurs in the following two cases: • While the block count enable is being set, the total data length specified by the descriptor table is different from that specified by the block count and block length. • Total data length cannot be divided by the block length. 0b - No error 1b - Error
1-0 ADMAES	ADMA error state (when ADMA error is occurred) This field indicates the state of the ADMA when an error has occurred during an ADMA data transfer. See ADMA Error Status (ADMA_ERR_STATUS) for more details.

10.3.7.1.23 ADMA System Address (ADMA_SYS_ADDR)

10.3.7.1.23.1 Offset

Register	Offset
ADMA_SYS_ADDR	58h

10.3.7.1.23.2 Function

This register contains the physical system memory address used for ADMA transfers.

10.3.7.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ADS_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADS_ADDR														0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

10.3.7.1.23.4 Fields

Field	Function
31-2 ADS_ADDR	ADMA system address This register holds the word address of the executing command in the Descriptor table. At the start of ADMA, the host driver sets the start address of the Descriptor table. The ADMA engine increments this register address whenever fetching a Descriptor command. When the ADMA is stopped at the Block Gap, this register indicates the address of the next executable Descriptor command. When the ADMA Error Interrupt is generated, this register holds the valid Descriptor address depending on the ADMA state. The lower 2 bits of this register is tied to '0' so the ADMA address is always word aligned. Because this register supports dynamic address reflecting, when TC field is set, it automatically alters the value of internal address counter, so the software cannot change this register when TC field is set. Such restriction is also listed in Software restrictions.
1-0 —	Reserved

10.3.7.1.24 DLL (Delay Line) Control (DLL_CTRL)

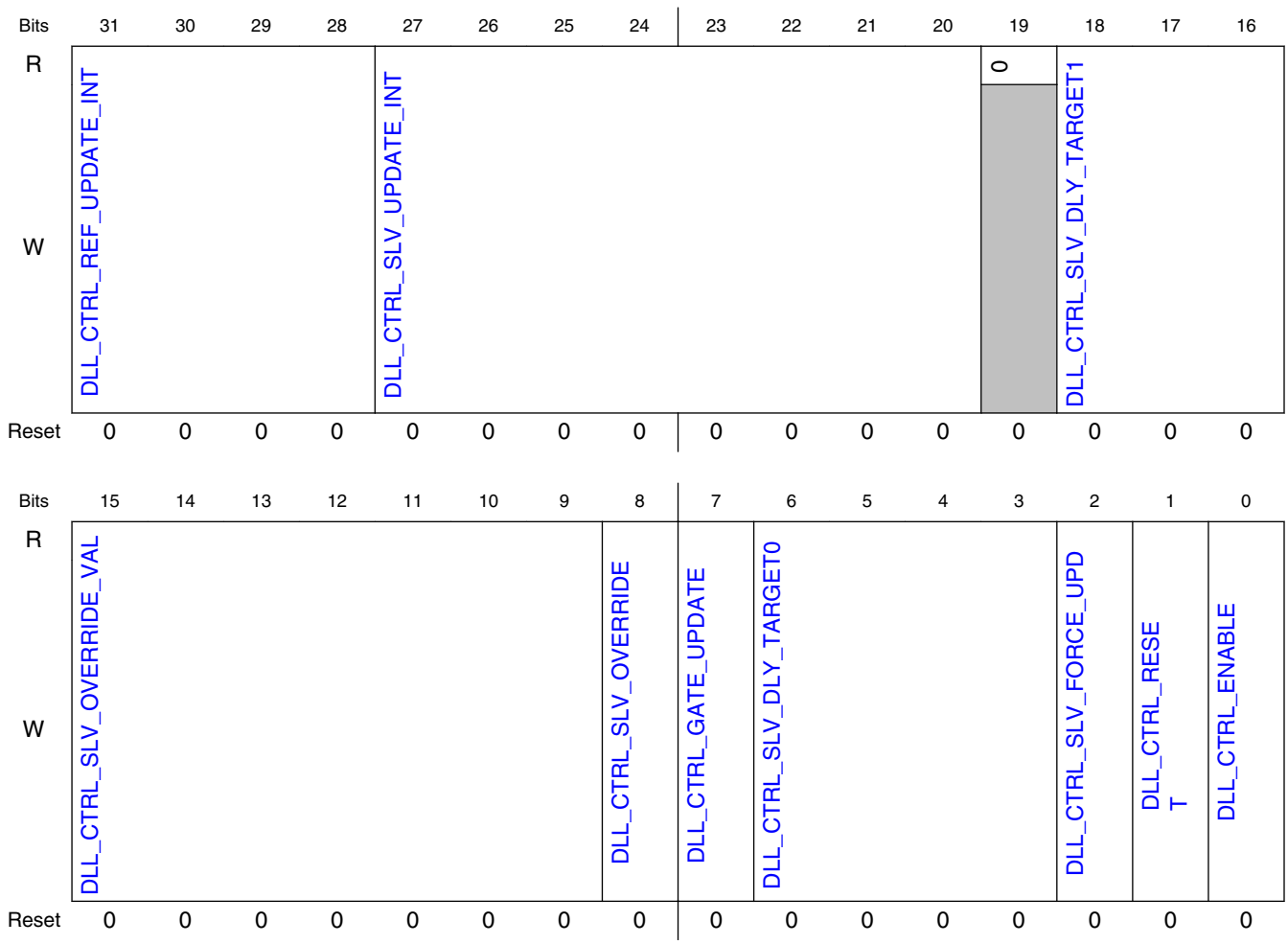
10.3.7.1.24.1 Offset

Register	Offset
DLL_CTRL	60h

10.3.7.1.24.2 Function

This register contains control fields for DLL.

10.3.7.1.24.3 Diagram



10.3.7.1.24.4 Fields

Field	Function
31-28 DLL_CTRL_REF_UPDATE_INT	DLL control loop update interval The interval cycle is (2 + REF_UPDATE_INT) * REF_CLOCK. By default, the DLL control loop updates every two REF_CLOCK cycles. It should be noted that increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20	Slave delay line update interval

Table continues on the next page...

Field	Function
DLL_CTRL_SLV_UPDATE_INT	If default 0 is used, it means 256 cycles of REF_CLOCK. A value of 0x0f results in 15 cycles and so on. Note that software can always cause an update of the slave-delay line using the SLV_FORCE_UPDATE register. Note that the slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19 —	Reserved
18-16 DLL_CTRL_SLV_DLY_TARGET1	DLL slave delay target1 See DLL_CTRL_SLV_DLY_TARGET0 below.
15-9 DLL_CTRL_SLV_OVERRIDE_VAL	DLL slave override val When SLV_OVERRIDE = 1, this field is used to select 1 of 128 physical taps manually. A value of 0 selects tap 1, and a value of 0x7f selects tap 128.
8 DLL_CTRL_SLV_OVERRIDE	DLL slave override Set this field to 1 to Enable manual override for slave delay chain using SLV_OVERRIDE_VAL; to set 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0
7 DLL_CTRL_GATE_UPDATE	DLL gate update Set this field to 1 to prevent the DLL from updating (because when clock_in exists, glitches might appear during DLL updates). This field might be used by software if such a condition occurs. Clear the bit to 0 to allow the DLL to update automatically.
6-3 DLL_CTRL_SLV_DLY_TARGET0	DLL slave delay target0 The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an ref_clock half-period. The delay is (({DLL_CTRL_SLV_DLY_TARGET1,DLL_CTRL_SLV_DLY_TARGET0} +1)* REF_CLOCK / 2) / 16 So the input read-clock can be delayed relative input data from (REF_CLOCK / 2) / 16 to REF_CLOCK * 4.
2 DLL_CTRL_SLV_FORCE_UPD	DLL slave delay line Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line updates automatically based on the SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 DLL_CTRL_RESET	DLL reset Setting this field to 1 force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, so to create a subsequent reset, RESET must be taken low and then asserted again.
0 DLL_CTRL_ENABLE	DLL and delay chain Set this field to 1 to enable the DLL and delay chain; otherwise; set to 0 to bypasses DLL. Note that using the slave delay line override feature with SLV_OVERRIDE and SLV_OVERRIDE VAL, the DLL does not need to be enabled.

10.3.7.1.25 DLL Status (DLL_STATUS)

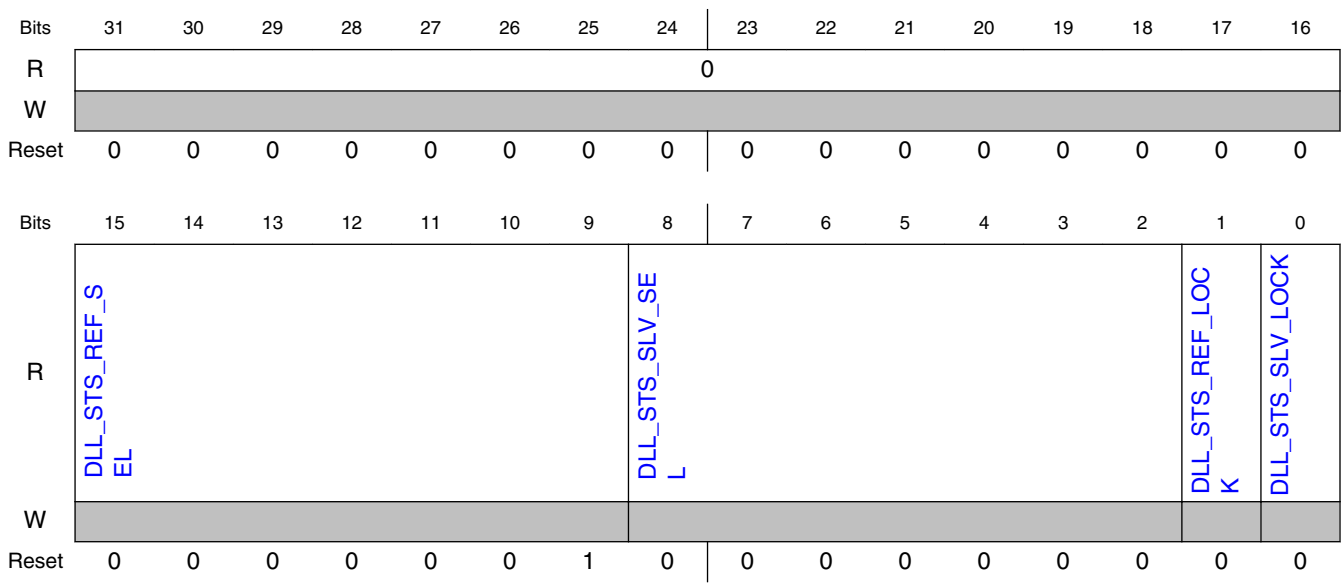
10.3.7.1.25.1 Offset

Register	Offset
DLL_STATUS	64h

10.3.7.1.25.2 Function

This register contains the DLL status information. All fields are read only and reads the same as the power-reset value.

10.3.7.1.25.3 Diagram



10.3.7.1.25.4 Fields

Field	Function
31-16 —	Reserved
15-9 DLL_STS_REF_SEL	Reference delay line select taps This is encoded by 7 fields for 127 taps.
8-2 DLL_STS_SLV_SEL	Slave delay line select status This is the instant value generated from reference chain. Because the reference chain can only be updated when REF_CLOCK is detected, this value should be the right value to be updated when the reference is locked.
1	Reference DLL lock status

Table continues on the next page...

Field	Function
DLL_STS_REF_LOCK	This signifies that the DLL has detected and locked to a half-phase ref_clock shift, allowing the slave delay-line to perform programmed clock delays
0	Slave delay-line lock status
DLL_STS_SLV_LOCK	This signifies that a valid calibration has been set to the slave-delay line and that the slave-delay line is implementing the programmed delay value

10.3.7.1.26 CLK Tuning Control and Status (CLK_TUNE_CTRL_STATUS)

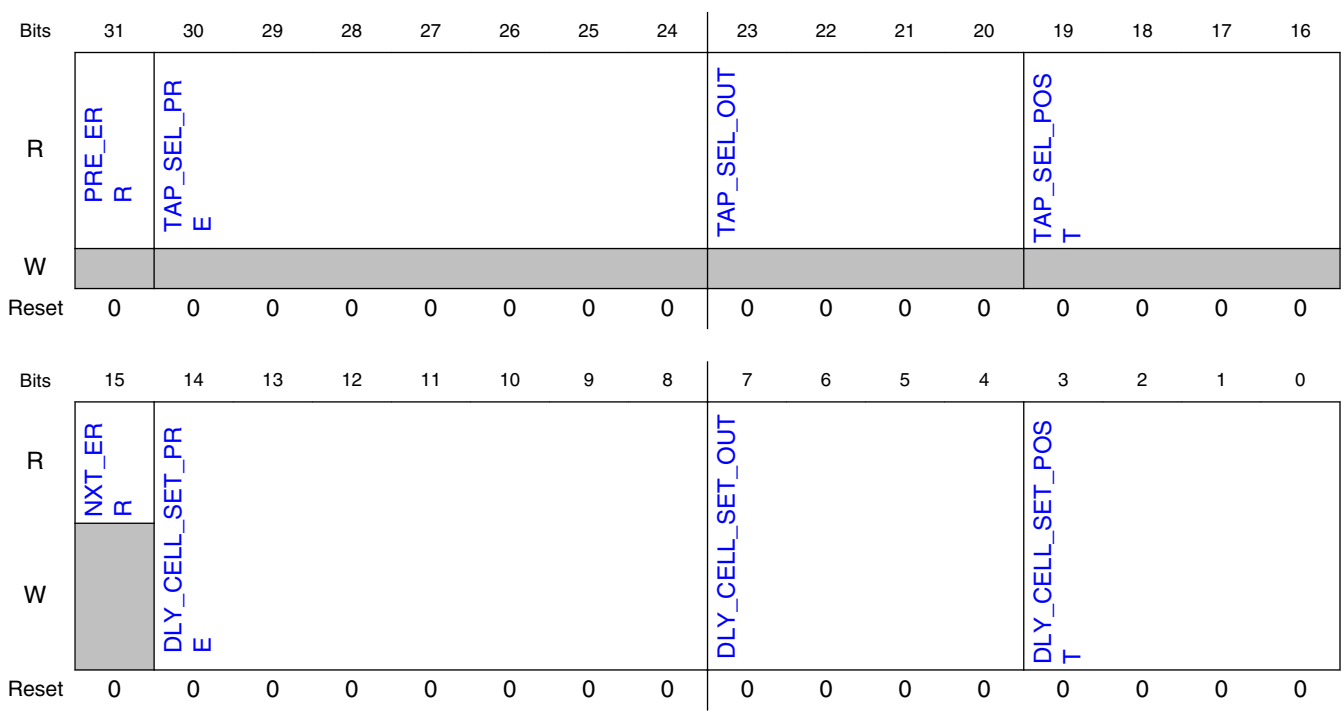
10.3.7.1.26.1 Offset

Register	Offset
CLK_TUNE_CTRL_STATUS	68h

10.3.7.1.26.2 Function

This register contains the Clock Tuning Control status information. All fields are read only and reads the same as the power-reset value. This register is added to support SD3.0 UHS-I SDR104 mode and EMMC HS200 mode.

10.3.7.1.26.3 Diagram



10.3.7.1.26.4 Fields

Field	Function
31 PRE_ERR	PRE error PRE error which means the number of delay cells added on the feedback clock is too small. It is valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
30-24 TAP_SEL_PRE	TAP_SEL_PRE Reflects the number of delay cells added on the feedback clock between the feedback clock and CLK_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is disabled, TAP_SEL_PRE is always equal to DLY_CELL_SET_PRE. When AUTO_TUNE_EN (bit24 of 0x48) is enabled, TAP_SEL_PRE is updated automatically according to the status of the auto tuning circuit to adjust the sample clock phase.
23-20 TAP_SEL_OUT	Delay cells added on the feedback clock between CLK_PRE and CLK_OUT Reflects the number of delay cells added on the feedback clock between CLK_PRE and CLK_OUT.
19-16 TAP_SEL_POST	Delay cells added on the feedback clock between CLK_OUT and CLK_POST Reflects the number of delay cells added on the feedback clock between CLK_OUT and CLK_POST.
15 NXT_ERR	NXT error NXT error which means the number of delay cells added on the feedback clock is too large. It's valid only when SMP_CLK_SEL of Mix control register (bit23 of 0x48) is enabled.
14-8 DLY_CELL_SET_PRE	delay cells on the feedback clock between the feedback clock and CLK_PRE Set the number of delay cells on the feedback clock between the feedback clock and CLK_PRE.
7-4 DLY_CELL_SET_OUT	Delay cells on the feedback clock between CLK_PRE and CLK_OUT Set the number of delay cells on the feedback clock between CLK_PRE and CLK_OUT.
3-0 DLY_CELL_SET_POST	Delay cells on the feedback clock between CLK_OUT and CLK_POST Set the number of delay cells on the feedback clock between CLK_OUT and CLK_POST.

10.3.7.1.27 Strobe DLL control (STROBE_DLL_CTRL)

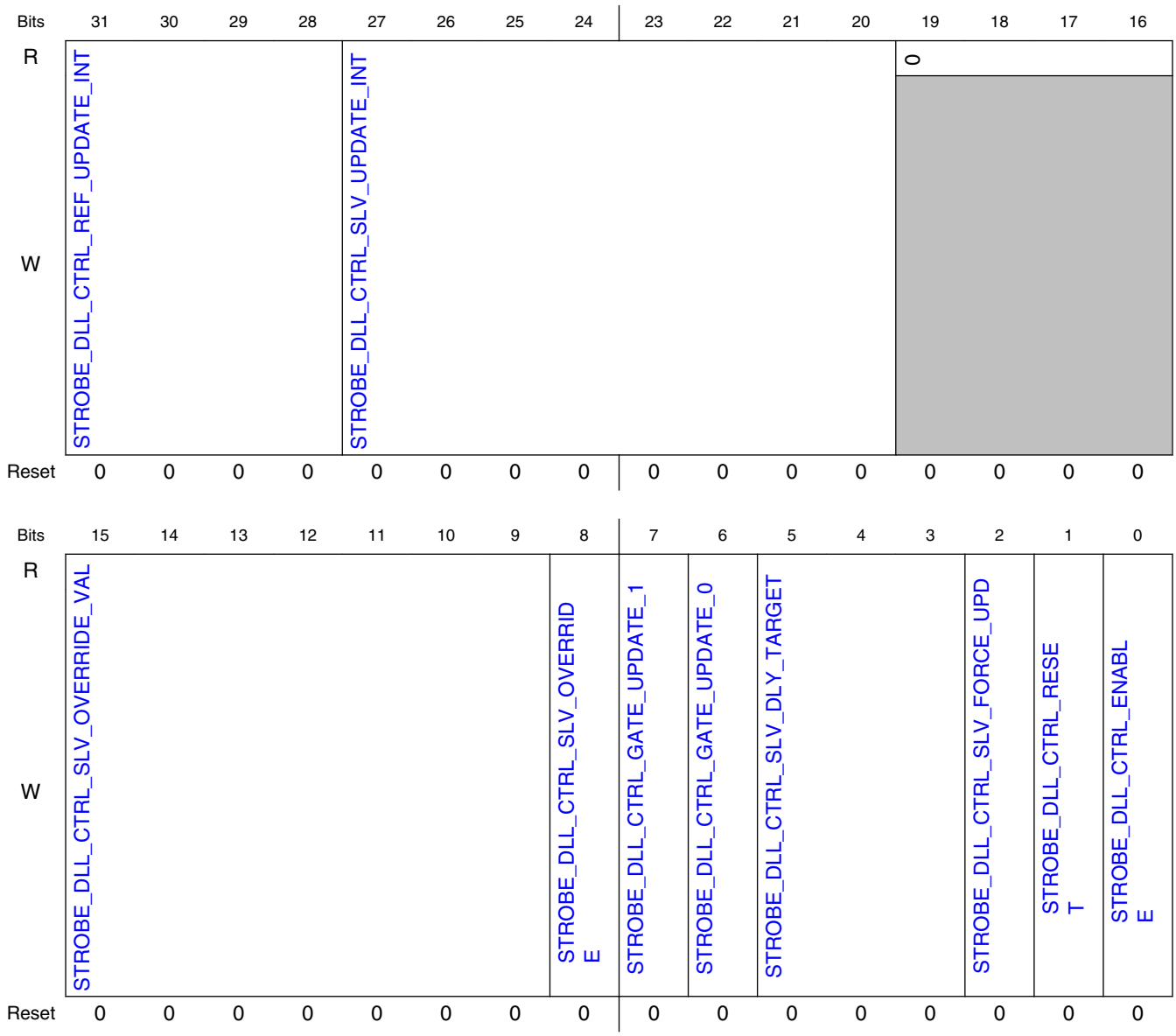
10.3.7.1.27.1 Offset

Register	Offset
STROBE_DLL_CTRL	70h

10.3.7.1.27.2 Function

This register contains the strobe DLL control information.

10.3.7.1.27.3 Diagram



10.3.7.1.27.4 Fields

Field	Function
31-28 STROBE_DLL_CTRL_REF_UPDATE_INT	Strobe DLL control reference update interval The interval cycle is (2 + STROBE_REF_UPDATE_INT) * STROBE_REF_CLOCK. By default, the DLL control loop updates every two STROBE_REF_CLOCK cycles.

Table continues on the next page...

Field	Function
	NOTE: Increasing the reference delay-line update interval reduces the ability of the DLL to adjust to fast changes in conditions that might effect the delay (such as voltage and temperature)
27-20 STROBE_DLL_CTRL_SLV_UPDATE_INT	Strobe DLL control slave update interval Slave delay line update interval. If default 0 is used, it means 256 cycles of STROBE_REF_CLOCK. A value of 0x0F results in 15 cycles and so on. **NOTE:** software can always cause an update of the slave-delay line using the STROBE_SLV_FORCE_UPDATE register. The slave delay line also updates automatically when the reference DLL transitions to a locked state (from an un-locked state).
19-16 —	This field is reserved. This read-only field is reserved and always has the value 0.
15-9 STROBE_DLL_CTRL_SLV_OVERRIDE_VAL	Strobe DLL control slave Override value When STROBE_SLV_OVERRIDE = 1, this field is used to manually select one of 128 physical taps. A value of 0 selects tap 1, and a value of 0x7F selects tap 128.
8 STROBE_DLL_CTRL_SLV_OVERRIDE	Strobe DLL control slave override Set this field to 1 to Enable manual override for slave delay chain using STROBE_SLV_OVERRIDE_VAL; set this field to 0 to disable manual override. This feature does not require the DLL to be enabled using the ENABLE field. In fact to reduce power, if STROBE_SLV_OVERRIDE is used, it is recommended to disable the DLL with ENABLE = 0.
7 STROBE_DLL_CTRL_GATE_UPDATE_1	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
6 STROBE_DLL_CTRL_GATE_UPDATE_0	Strobe DLL control gate update Set this field to 1 to prevent the DLL from updating (because when STROBE_CLOCK_IN exists, glitches might appear during DLL updates). This field can be used by software if such a condition occurs. Clear the field to 0 to allow the DLL to update automatically.
5-3 STROBE_DLL_CTRL_SLV_DLY_TARGET	Strobe DLL Control Slave Delay Target The delay target for uSDHC loopback read clock can be programmed in 1/16th increments of an STROBE_REF_CLOCK half-period. The delay is ((STROBE_DLL_CTRL_SLV_DLY_TARGET + 1) * STROBE_REF_CLOCK / 2) / 16, So the input read-clock can be delayed relative input data from (STROBE_REF_CLOCK / 2) / 16 to (STROBE_REF_CLOCK *4)/16.
2 STROBE_DLL_CTRL_SLV_FORCE_UPD	Strobe DLL control slave force updated Setting this field to 1, forces the slave delay line to update to the DLL calibrated value immediately. The slave delay line should automatically update the STROBE_SLV_UPDATE_INT interval or when a DLL lock condition is sensed. Subsequent forcing of the slave-line update can only occur if STROBE_SLV_FORCE_UP is set back to 0 and then asserted again (edge triggered). Be sure to use it when uSDHC is idle. This function might not work when uSDHC is working on data / cmd / response.
1 STROBE_DLL_CTRL_RESET	Strobe DLL control reset Setting this field to 1 to force a reset on DLL. This causes the DLL to lose lock and re-calibrate to detect an REF_CLOCK half period phase shift. This signal is used by the DLL as edge-sensitive, to create a subsequent reset, RESET must be taken low and then asserted again.
0 STROBE_DLL_CTRL_ENABLE	Strobe DLL control enable Set this field to 1 to enable the DLL and delay chain; otherwise, set to 0 to bypasses DLL. **NOTE:** Using the slave delay line override feature with STROBE_SLV_OVERRIDE and STROBE_SLV_OVERRIDE VAL, the DLL does not need to be enabled.

10.3.7.1.28 Strobe DLL status (STROBE_DLL_STATUS)

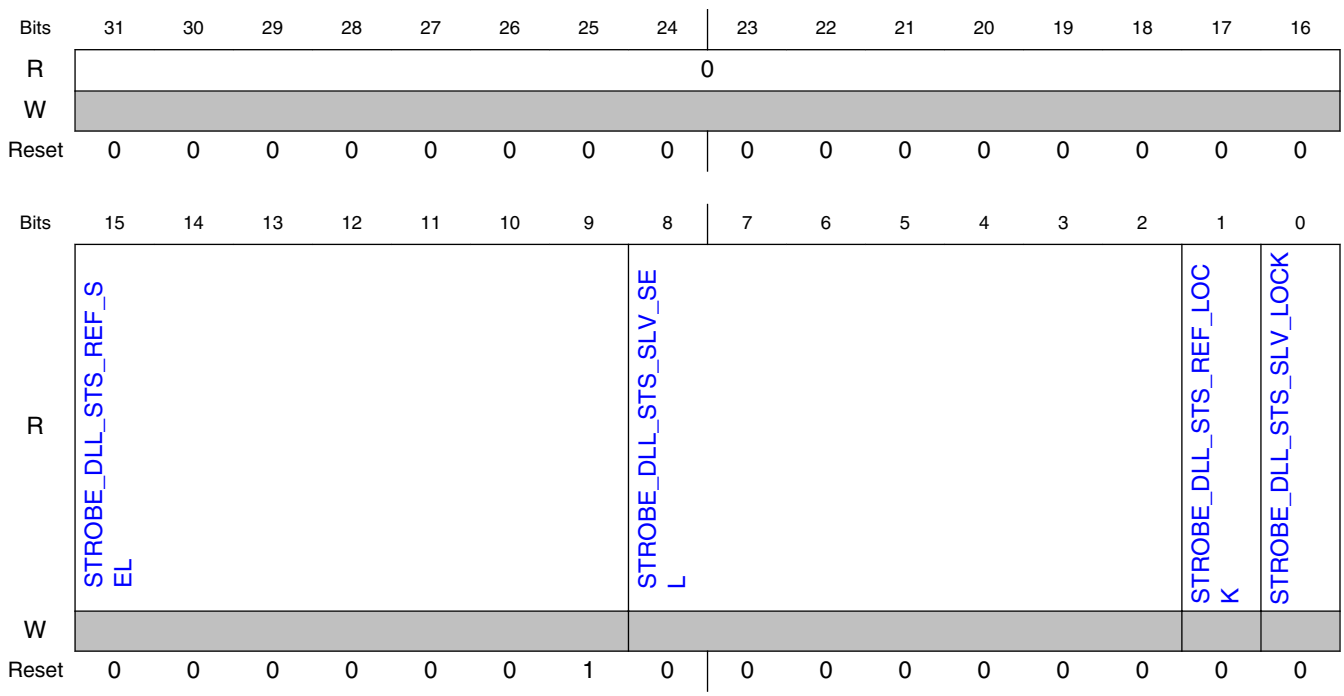
10.3.7.1.28.1 Offset

Register	Offset
STROBE_DLL_STATUS	74h

10.3.7.1.28.2 Function

This register contains the strobe DLL status information. All fields are read only and read the same as the power-reset value.

10.3.7.1.28.3 Diagram



10.3.7.1.28.4 Fields

Field	Function
31-16	This field is reserved.
—	This read-only field is reserved and always has the value 0.
15-9	Strobe DLL status reference select

Table continues on the next page...

Field	Function
STROBE_DLL_STS_REF_SEL	Reference delay line select taps. This is encoded by 7 fields for 127 taps.
8-2 STROBE_DLL_STS_SLV_SEL	Strobe DLL status slave select Slave delay line select status. This is the instant value generated from reference chain. Because the reference chain can only be updated when STROBE_REF_CLOCK is detected, this value can be updated with the right value when the reference is locked.
1 STROBE_DLL_STS_REF_LOCK	Strobe DLL status reference lock This signifies that the DLL has detected and locked to a half-phase REF_CLOCK shift, it allows the slave delay-line to perform programmed clock delays.
0 STROBE_DLL_STS_SLV_LOCK	Strobe DLL status slave lock Slave delay-line lock status. This signifies that a valid calibration has been set to the slave-delay line, and the slave-delay line is implementing the programmed delay value.

10.3.7.1.29 Vendor Specific Register (VEND_SPEC)

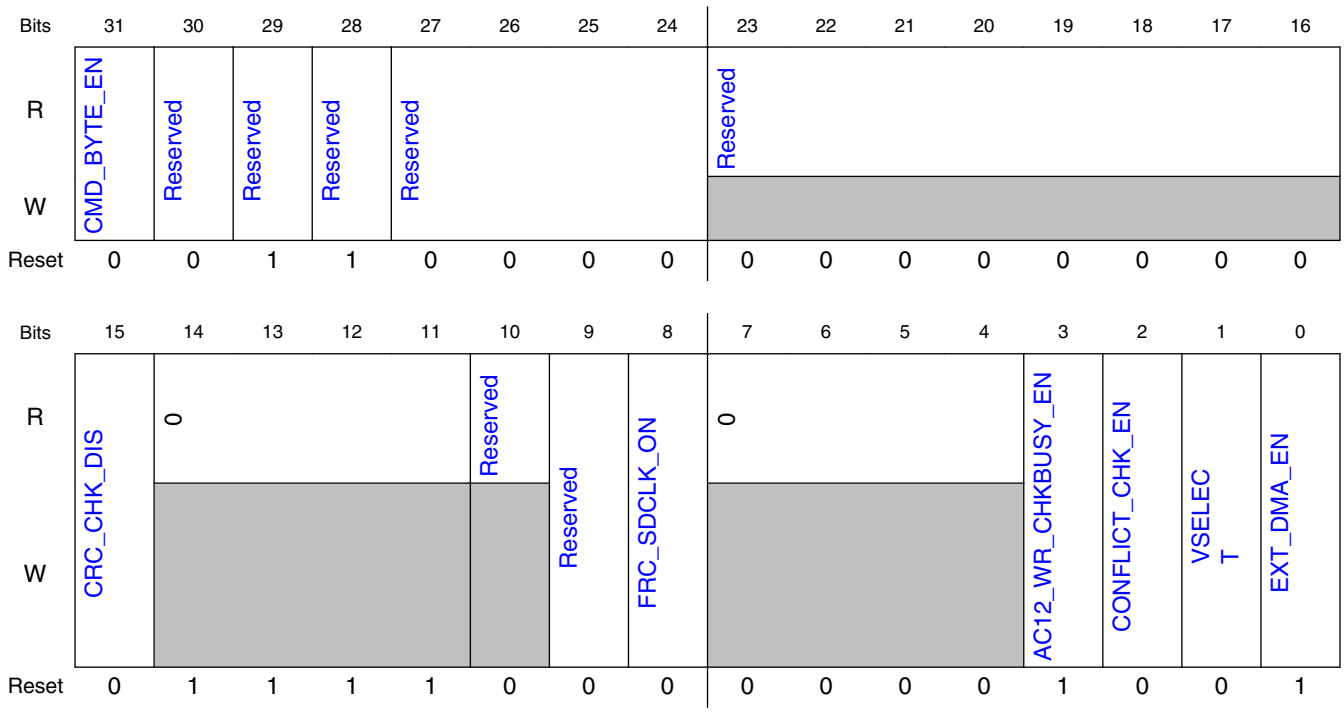
10.3.7.1.29.1 Offset

Register	Offset
VEND_SPEC	C0h

10.3.7.1.29.2 Function

This register contains the vendor specific control / status register.

10.3.7.1.29.3 Diagram



10.3.7.1.29.4 Fields

Field	Function
31 CMD_BYTE_EN	Byte access This bit controls the byte access. 0b - Disable 1b - Enable
30 —	Reserved Always write as 0.
29 —	Reserved Always write as 1
28 —	Reserved Always write as 0.
27-24 —	Reserved Always write as 4'b0000.
23-16 —	Reserved Always write as 8'h00.
15 CRC_CHK_DIS	CRC Check Disable 0b - Check CRC16 for every read data packet and check CRC fields for every write data packet 1b - Ignore CRC16 check for every read data packet and ignore CRC fields check for every write data packet
14-11	Reserved

Table continues on the next page...

Field	Function
—	Reserved
10 —	Reserved Always write as 0.
9 —	Reserved Always write as 0.
8 FRC_SDCLK_ON	Force CLK Force CLK output active 0b - CLK active or inactive is fully controlled by the hardware. 1b - Force CLK active
7-4 —	Reserved
3 AC12_WR_CHKBUSY_EN	Check busy enable Check busy enable after auto CMD12 for write data packet 0b - Do not check busy after auto CMD12 for write data packet 1b - Check busy after auto CMD12 for write data packet
2 CONFLICT_CHK_EN	Conflict check enable It is not implemented in uSDHC IP. 0b - Conflict check disable 1b - Conflict check enable
1 VSELECT	Voltage selection Change the value of output signal VSELECT, to control the voltage on pads for external card. There must be a control circuit out of uSDHC to change the voltage on pads. 0b - Change the voltage to high voltage range, around 3.0 V 1b - Change the voltage to low voltage range, around 1.8 V
0 EXT_DMA_EN	External DMA request enable Enable the request to external DMA. When the internal DMA (either Simple DMA or Advanced DMA) is not in use, and this field is set, uSDHC sends out DMA request when the internal buffer is ready. This field is particularly useful when transferring data by Arm platform polling mode, and it is not allowed to send out the external DMA request. By default, this field is set. 0b - In any scenario, uSDHC does not send out external DMA request. 1b - When internal DMA is not active, the external DMA request is sent out.

10.3.7.1.30 MMC Boot (MMC_BOOT)

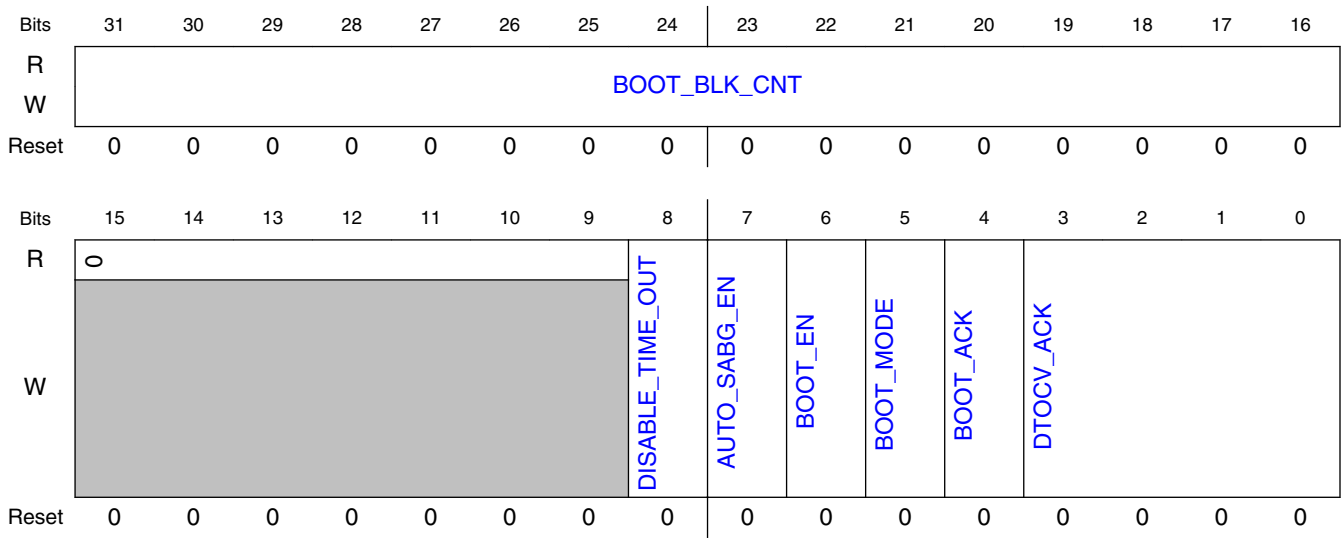
10.3.7.1.30.1 Offset

Register	Offset
MMC_BOOT	C4h

10.3.7.1.30.2 Function

This register contains the MMC Fast Boot control register.

10.3.7.1.30.3 Diagram



10.3.7.1.30.4 Fields

Field	Function
31-16 BOOT_BLK_CNT	Stop At Block Gap value of automatic mode The value defines the Stop At Block Gap value of automatic mode. When received, card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT) and AUTO_SABG_EN is 1, then Stop At Block Gap. Here, BLK_CNT is defined in the Block Attributes Register, field 31 - 16 of 0x04.
15-9 —	Reserved
8 DISABLE_TIME_OUT	Time out **NOTE:** When this field is set, there is no timeout check no matter whether BOOT_EN is set or not. 0b - Enable time out 1b - Disable time out
7 AUTO_SABG_EN	Auto stop at block gap During boot, enable auto stop at block gap function. This function is triggered, and host stops at block gap when received card block cnt is equal to (BLK_CNT - BOOT_BLK_CNT).
6 BOOT_EN	Boot enable Boot mode enable 0b - Fast boot disable 1b - Fast boot enable
5 BOOT_MODE	Boot mode Boot mode select

Table continues on the next page...

Field	Function
	0b - Normal boot 1b - Alternative boot
4 BOOT_ACK	BOOT ACK Boot ACK mode select 0b - No ack 1b - Ack
3-0 DTOCV_ACK	DTOCV_ACK Boot ACK time out counter value 0000b - SDCLK x 2^32 0001b - SDCLK x 2^33 0010b - SDCLK x 2^18 0011b - SDCLK x 2^19 0100b - SDCLK x 2^20 0101b - SDCLK x 2^21 0110b - SDCLK x 2^22 0111b - SDCLK x 2^23 1110b - SDCLK x 2^30 1111b - SDCLK x 2^31

10.3.7.1.31 Vendor Specific 2 Register (VEND_SPEC2)

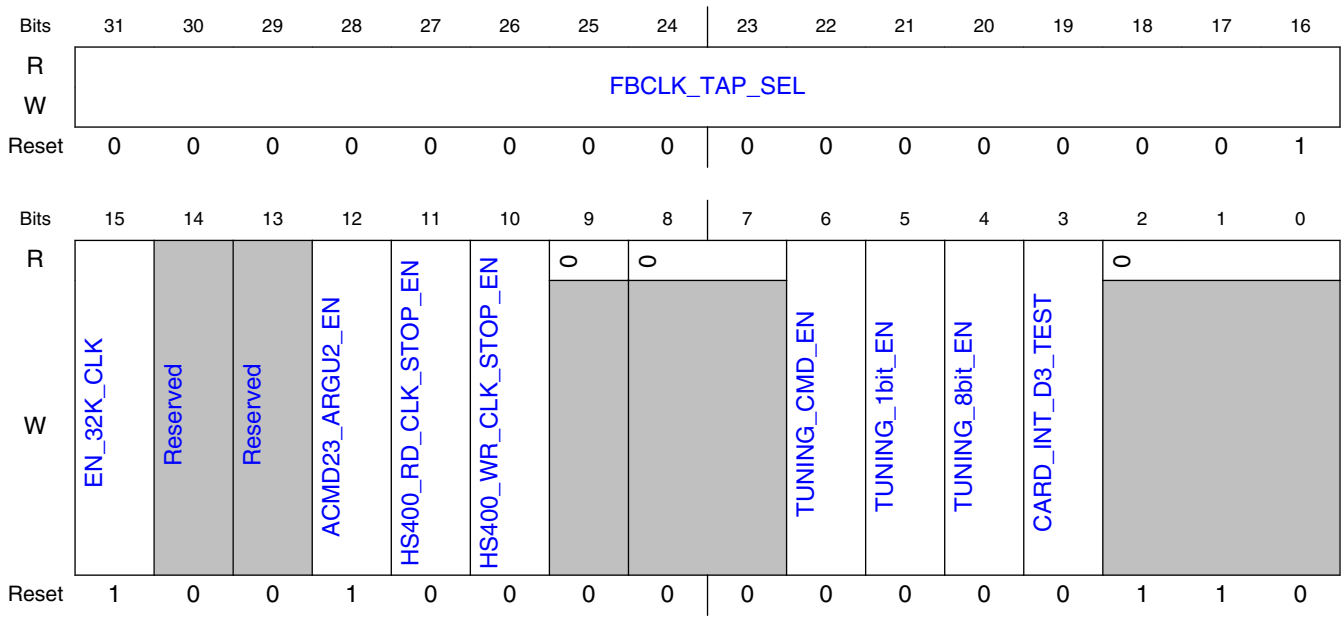
10.3.7.1.31.1 Offset

Register	Offset
VEND_SPEC2	C8h

10.3.7.1.31.2 Function

This register contains the vendor specific control 2 register.

10.3.7.1.31.3 Diagram



10.3.7.1.31.4 Fields

Field	Function
31-16 FBCLK_TAP_SEL	Enable extra delay on internal feedback clock
15 EN_32K_CLK	Enable 32khz clock for card detection Use 32khz clock for card detection
14 —	Reserved
13 —	Reserved
12 ACMD23_ARGU2_EN	Argument2 register enable for ACMD23 0b - Disable 1b - Argument2 register enable for ACMD23 sharing with SDMA system address register. Default is enabled.
11 HS400_RD_CLK_STOP_EN	HS400 read clock stop enable Only stop clock at read block gap.
10 HS400_WR_CLK_STOP_EN	HS400 write clock stop enable Only stop clock at write block gap.
9 —	Reserved

Table continues on the next page...

Field	Function
8-7 —	Reserved
6 TUNING_CMD_EN	Tuning command enable Enable the auto tuning circuit to check the CMD line. 0b - Auto tuning circuit does not check the CMD line. 1b - Auto tuning circuit checks the CMD line.
5 TUNING_1bit_EN	Tuning 1bit enable Enable the auto tuning circuit to check the DATA0 only. It is used with the TUNING_8bit_EN together.
4 TUNING_8bit_EN	Tuning 8bit enable Enable the auto tuning circuit to check the DATA[7:0]. It is used with the TUNING_1bit_EN together. 0b00 - Tuning circuit only checks the DATA[3:0] 0b01 - Tuning circuit only checks the DATA0 0b10 - Tuning circuit checks the whole DATA[7:0] 0b11 - Invalid **NOTE:** The format of these two fields are [TUNNING_8bit_EN:TUNNING_1bit_EN].
3 CARD_INT_D3_TEST	Card interrupt detection test This field only uses for debugging. 0b - Check the card interrupt only when DATA3 is high. 1b - Check the card interrupt by ignoring the status of DATA3.
2-0 —	Reserved

10.3.7.1.32 Tuning Control (TUNING_CTRL)

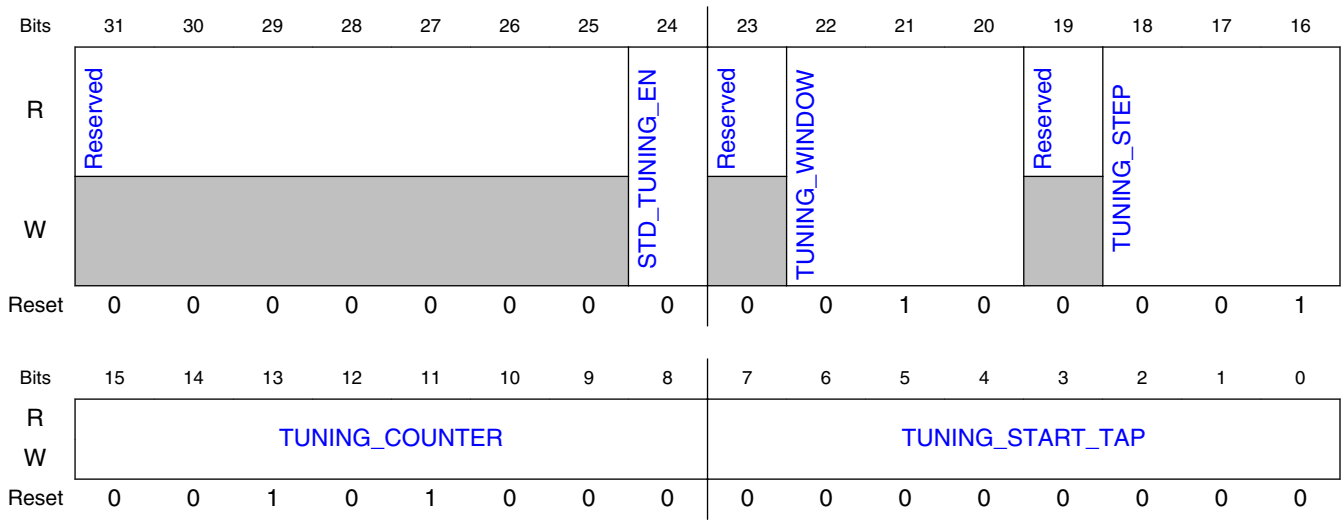
10.3.7.1.32.1 Offset

Register	Offset
TUNING_CTRL	CCh

10.3.7.1.32.2 Function

The register contains configuration of tuning circuit.

10.3.7.1.32.3 Diagram



10.3.7.1.32.4 Fields

Field	Function
31-25 —	Reserved
24 STD_TUNING_EN	Standard tuning circuit and procedure enable This field is used to enable standard tuning circuit and procedure.
23 —	Reserved
22-20 TUNING_WINDOW	Data window Select data window value for auto tuning
19 —	Reserved
18-16 TUNING_STEP	TUNING_STEP The increasing delay cell steps in tuning procedure.
15-8 TUNING_COUNTER	Tuning counter The MAX repeat CMD19 times in tuning procedure.
7-0 TUNING_START_TAP	Tuning start The start delay cell point when send first CMD19 in tuning procedure.

10.3.7.1.33 Command Queue (CQE)

10.3.7.1.33.1 Offset

Register	Offset
CQE	100h

10.3.7.1.33.2 Function

All the CQE registers are defined in JESD84-B51.pdf. See the B.4 section of JESD84-B51.pdf

10.3.7.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	1	0	0	0	0

10.3.7.1.33.4 Fields

Field	Function
31-0 —	Reserved

Chapter 11
Connectivity

11.1 Universal Serial Bus Controller (USB)

11.1.1 Overview

The USB module is a USB 3.0-compliant serial interface engine for implementing a USB interface. This module may be connected to an external port. Collectively the module and external port are called the USB 3.0 interface. USB 3.0 supports super-speed (SS), high-speed (HS), full-speed (FS), and low-speed (LS) operations.

NOTE

Synopsys Proprietary. Used with permission.

- The upper layer is common for USB 2.0 and USB 3.0 operation. This has the bus interface, buffer management block, list processor for scheduling, and control and status register (CSR) functions
- USB 2.0 PHY and MAC layers
- USB 3.0 PHY, LINK, and MAC layers

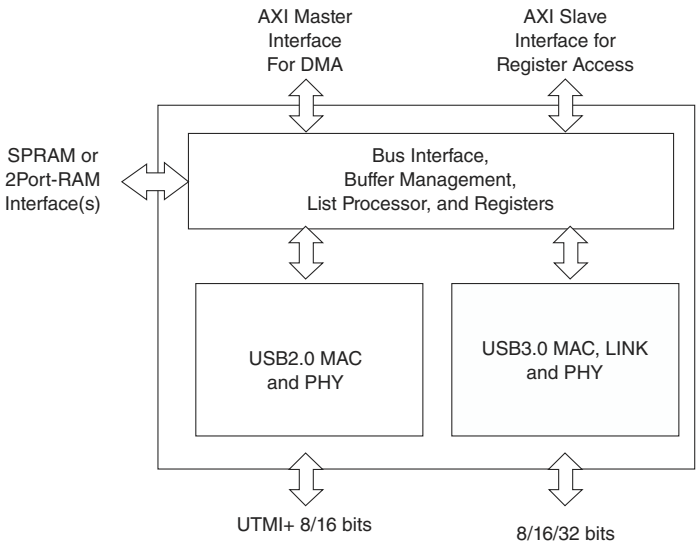


Figure 11-1. USB interface block diagram

11.1.1.1 Features

The USB 3.0 module includes the following features:

- Complies with USB specification rev 3.0 (xHCI compatible)
- Supports operation as a standalone USB host controller
- USB dual-role operation and can be configured as host or device
- Super-speed (5 Gbit/s), high-speed (480 Mbit/s), full-speed (12 Mbit/s), and low-speed (1.5 Mbit/s) operations.
- Supports operation as a standalone single port USB
- Supports four programmable, bidirectional USB endpoints
- OTG (on-the-go) 2.0 compliant, which includes both device and host capability. Super-speed operation is not supported when OTG is enabled.
- Supports system memory interface with -bit addressing capability

11.1.1.2 Modes of Operation

The USB 3.0 module operates in following modes.

- Host Mode: SS/HS/FS/LS
- Device Mode: SS/HS/FS
- OTG: HS/FS/LS

11.1.1.3 External Signals

This section contains detailed descriptions of all the USB 3.0 controller signals. Many of the signals for the PHY interfaces are muxed onto the same pins in order to reduce pin count. The following table shows the USB signals, indicating which interface supports each signal.

Table 11-1. USB 3.0 External Signals

Signal	I/O	Description
USB1_D_P	IO	USB PHY Data Plus
USB1_D_M	IO	USB PHY Data Minus
USB1_VBUS	IO	USB PHY VBUS
USB1_ID	IO	USB PHY ID Detect
USB1_TX_P	O	USB PHY 3.0 Transmit Data (positive)
USB1_TX_M	O	USB PHY 3.0 Transmit Data (negative)
USB1_RX_P	I	USB PHY 3.0 Receive Data (positive)
USB1_RX_M	I	USB PHY 3.0 Receive Data (negative)
USB1_RESREF	IO	USB PHY Impedance Calibration
USB1_DRVVBUS	O	VBus power enable.
USB1_PWRFAULT	I	Indicates that a Vbus fault has occurred.

11.1.2 Functional Description

11.1.2.1 System memory descriptor and data buffers

The software creates transfer request blocks (TRBs), and four DWORDs each, that point to the data buffers. Normally, the TRBs are allocated consecutively in system memory; only the data buffers can be scattered. In the case of a circular buffer, the link-TRB points to the next TRB. Once the TRBs and data buffers are set up in the system memory, the software driver issues a start transfer command that points to the location of the first TRB in the system memory to start the DMA operation. TRBs, though small (only four DWORDs), provide a rich set of features for the software to schedule transfers, isochronous, control, interrupt moderation, and so on.

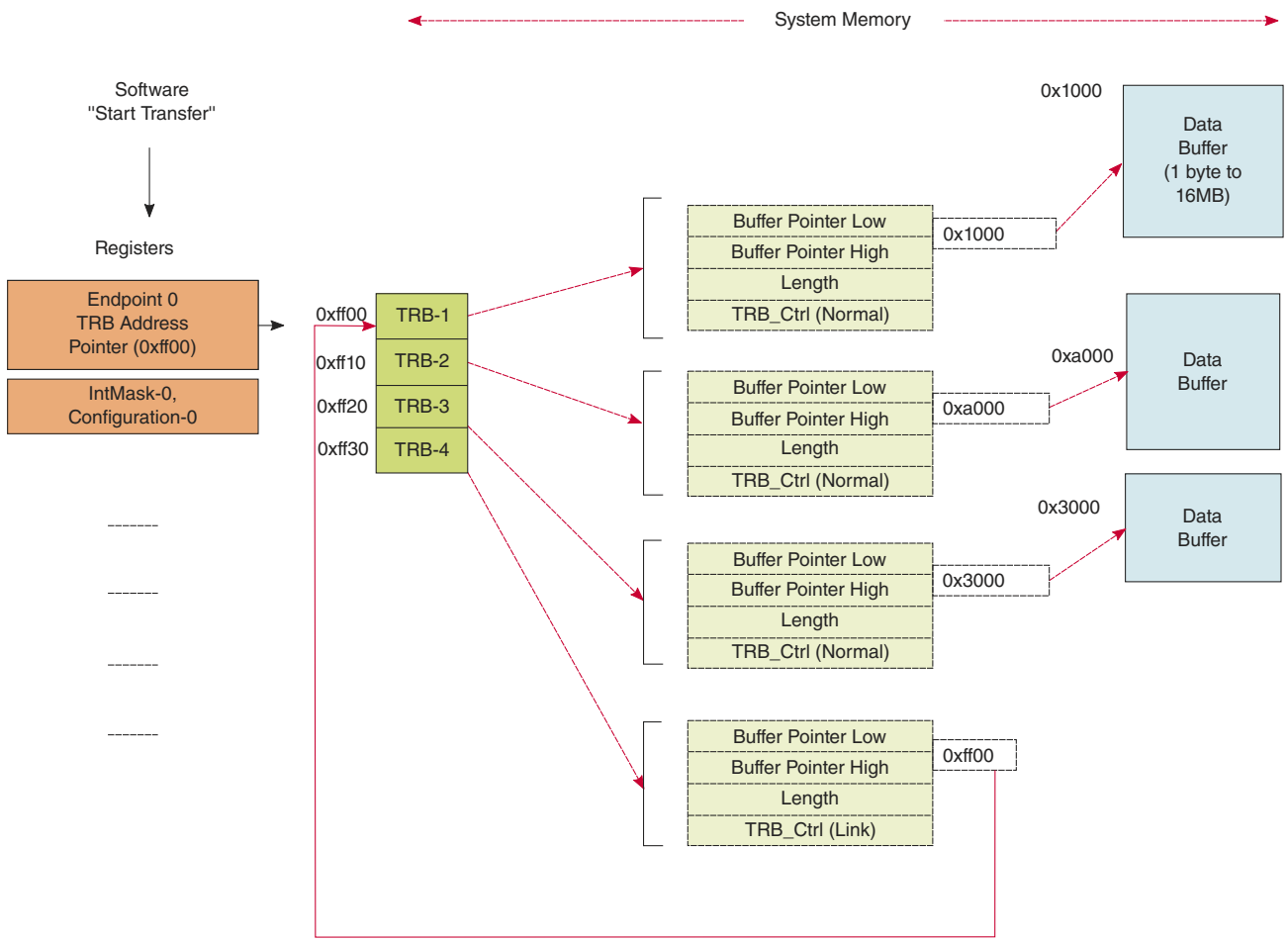


Figure 11-2. System Memory Descriptor and Data Buffers

11.1.2.2 Device descriptor structures

Device mode transfer request blocks (TRBs) are small 4 DWORDs and at the same time provide a rich set of features so that the software can efficiently manage the USB core, memory buffers, and MIPS requirements.

The following is a list of device mode TRB characteristics:

- TRBs provide support for scattered data structures. The scattered data buffers can be zero bytes to 16 MB in length.
- TRBs must be placed in system memory aligned to a 16-byte boundary.
- TRBs are kept in linear memory to enhance descriptor caching performance. If the data buffers are small (as in an Ethernet application) the core can efficiently collect the scattered data to build USB packets without wasting bus efficiency that collecting scattered descriptors requires.

- Supports TRB linking. This allows software to set up a circular ring of TRBs with the last TRB linking to the first one.
- Supports system memory interface with -bit addressing capability
- Supports byte-aligned buffers for each TRB of a transfer. This feature prevents the need for buffer copying in cases where the USB device driver receives unaligned data buffers from other applications (for example, Ethernet). Whenever the application controls buffer allocation, it must allocate SoC bus width and burst-aligned buffers to facilitate efficient bus and memory utilization. For transmitting, the core supports byte-aligned buffers on all TRBs.

  For example: To use 16 bursts in a 64-bit system, you must try to allocate buffers that are 16 * 8 = 128 bytes aligned. This is the normal buffer structure because Linux-like OSs allocate 4 KB buffers. SDR/DDR memory controllers also provide better performance when requests are burst aligned.

- Supports software queueing of multiple USB transfers (LST bit for IN/OUT transfers and CSP bit for OUT transfers control this function).
- Provides interrupt moderation capability, allowing software to selectively enable events on TRB completion, as controlled by the IOC (Interrupt on completion) bit. The IOC event is also used by software to reallocate the released buffers, allowing software to re-use just a few buffers in a circular fashion. This helps when the memory is limited but have enough MIPS to process interrupts. Larger buffers can be allocated to reduce the number of interrupts.
  - In USB 3.0 larger transfers are recommended because the raw transfer rate is almost 10 times faster, unlike USB 2.0, where drivers set up only 64 KB or 128 KB transfers. For example, a 64 KB transfer that takes 1.3 ms in USB 2.0 requires only 164 s in USB 3.0. If you set up buffers of 64 KB and enable the transfer completion event, then you receive an interrupt every 164 sec.
- Supports streaming (Stream ID field used for this purpose).

11.1.2.2.1 Structures

This figure shows the control and status field of a transfer request block.

Figure 11-3. TRB control and status fields

Table 11-2. Device descriptor structure field definitions

Field	Description	Hardware access
DW 03-00		
31:0	Buffer pointer low (BPTRL)	R/W

Table continues on the next page...

Table 11-2. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
	Data buffer pointer to low 32 bit address (BPTR[31:0]). Hardware may also update this field (implementation specific).	
DW 07-04		
31:0	Buffer pointer high (BPTRH) Data buffer pointer to high 32-bit address (BPTR[63:32]).	R/W
DW 0B-08		
31:28	TRB status (TRBSTS) Hardware updates this field with transfer status information before releasing the TRB. • 4'h0: OK • 4'h1: MissedIsoc: Isochronous interval missed or incomplete • 4'h2: SetupPending - During the current control transfer data/status phase, another SETUP was received. • 4'h4: TransferInProgress - During the current transfer, an end transfer command was received.	R/W
25:24	Packet count M1 (PCM1) For high-speed, High bandwidth isochronous IN endpoints, this field in an Isoc-First TRB represents the total number of packets in the Buffer Descriptor minus 1.	R/W
23:0	Buffer size (BUFSIZ) If CHN=0 and HWO=0 for the TRB, this field represents the total remaining Buffer Descriptor buffer size in bytes. Valid Range: 0 bytes to (16 MB - 1 byte). The hardware decrements this field to represent the remaining buffer size after data is transferred. For a Link TRB, the buffer size should be "0".	R/W
DW 0F-0C		
31:30	Reserved.	R/W
29:14	Stream ID / SOF Number For stream-based bulk endpoints: The Stream ID of the transfer this TRB is associated with. Stream ID must be the same in all TRBs (R/W). For isochronous endpoints: The (micro)frame number in which the last packet of this TRB's buffer was transmitted or received (debug purposes only) (RO).	R/W
13:12	Reserved.	R/W
11	Interrupt on Complete (IOC) When IOC is set in a TRB, and once the transfer for this buffer is completed, the core will issue XferInProgress event with IOC bit set in the event's status. This indicates the buffer is available for software to reuse or release.	R
10	Interrupt on Short Packet / Interrupt on Missed ISOC (ISP/IMI) Applicable to OUT endpoints when a short packet is received, and CSP=1 and LST=0. If this bit is 1, the core generates an XferInProgress event. For Isochronous endpoints: If this bit is 1, the core generates an XferInProgress event when the interval represented by the Buffer Descriptor completes with a "Missed Isoc" status.	R
9:4	TRB Control (TRBCTL) Indicates the type of TRB:	R

Table continues on the next page...

Table 11-2. Device descriptor structure field definitions (continued)

Field	Description	Hardware access
	1: Normal (Control-Data-2+ / Bulk / Interrupt) - Set TRBCTL to 1 for all TRBs used in data stage except the first TRB 2: Control-Setup 3: Control-Status-2 - Set TRBCTL to 3 for a SETUP request without data stage 4: Control-Status-3 - Set TRBCTL to 4 for a SETUP request with data stage 5: Control-Data - Set TRBCTL to 5 for the first TRB of a data stage 6: Isochronous-First - Set TRBCTL to 6 for the first TRB of a Service Interval 7: Isochronous 8: Link TRB Others: Reserved	
3	Continue on Short Packet (CSP) Applicable to OUT endpoints only when a short packet is received. If this bit is 1, the core will continue to the next Buffer Descriptor. This setting is required for isochronous endpoints. If this bit is 0, the core will generate an XferComplete event and remove the stream.	R
2	Chain Buffers (CHN) Applicable to IN and OUT endpoints. Set to 1 by software to associate this TRB with the next TRB. A Buffer Descriptor is defined as one or more TRBs. The CHN bit is used to identify the TRBs that comprise a Buffer Descriptor. The CHN bit is always 0 in the last TRB of a Buffer Descriptor and when the LST field is set to 1.	R
1	Last TRB (LST) Indicates this is the last TRB in a list. After completing the transfer for the associated buffer the core will stop the transfer for the endpoint / bulk-stream and issues an XferComplete event. The stream is automatically removed by the hardware.	R
0	Hardware Owner of Descriptor (HWO) Indicates that hardware owns the TRB. Software sets this bit to 1 when it creates the TRB, and cannot modify it until hardware resets this bit to 0. However, there are exceptions for short packets on OUT endpoints and Link TRBs. Because the hardware autonomously checks this bit to determine if the entire TRB is valid, software must set this field to '1' after preparing the other three DWORDs of the TRB with valid information.	R/W

11.1.2.2.1.1 Normal (Control-Data/Bulk/Interrupt), Isochronous, and Status Transfer Request Block Structure

The Normal TRB is used for Bulk/Control-Data/Interrupt endpoint transfers. The data buffers can be scattered anywhere and each may have different sizes. The TRB Buffer Pointer and the Buffer Size fields point to buffer address and size respectively. The Stream ID for bulk endpoints will be programmed by the application.

For isochronous endpoints, the first TRB in a service interval must have the Isoc-First type, the last TRB in a service interval must have CHN=0, and any other TRBs have CHN=1. The starting (micro)frame time is communicated via the Start Transfer command, and the core tracks the (micro)frame times of the subsequent Buffer Descriptors.

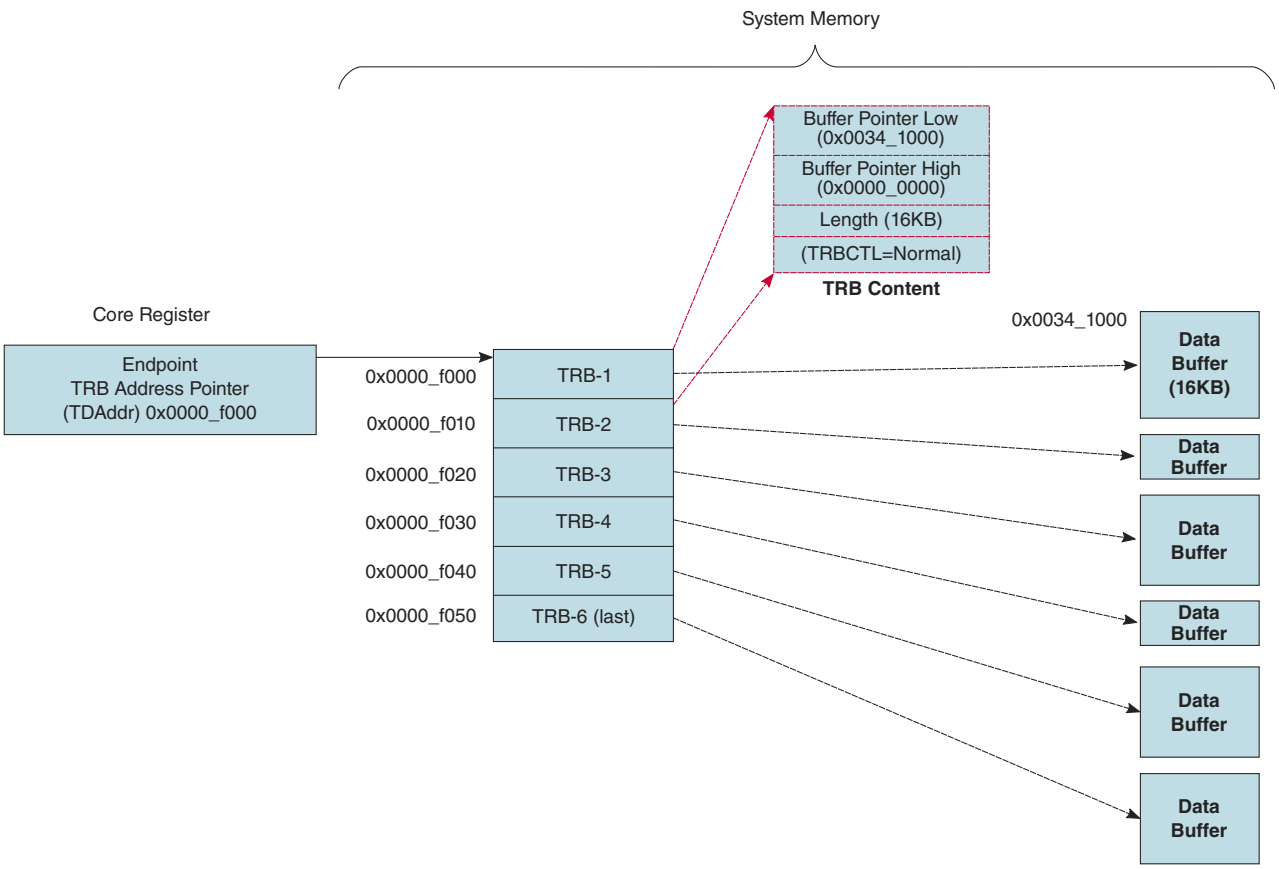


Figure 11-4. Normal (Control-Data/Bulk/Interrupt) Descriptor Structure

11.1.2.2.1.2 Setup and Status TRB Structure

To receive a SETUP packet, the driver queues up a single Setup TRB, whose buffer pointer value may be set to any address, including the address of the TRB. The buffer size must be set to 8. The core will write the 8 bytes of the received SETUP to the address requested. If the address of the TRB is used, there is no need for a separate data buffer to receive a SETUP packet. After completing Setup stage, driver will schedule Data stage and Status stage transfers. For more information, see Control transfer programming model

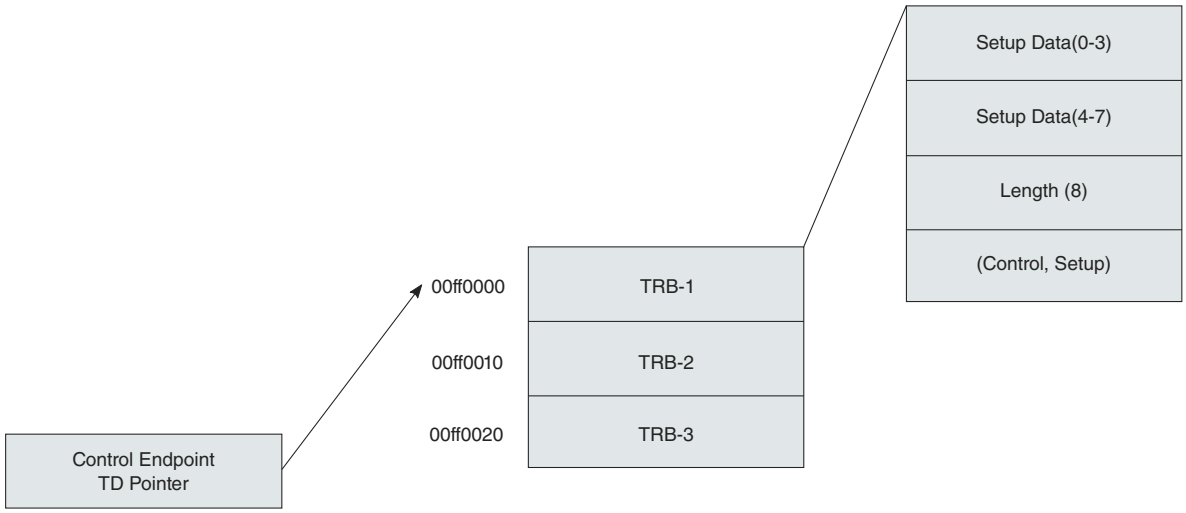


Figure 11-5. SETUP Descriptor Structure with Buffer Pointing to Setup TRB

After interpreting the SETUP bytes, software will determine if the next stage of the control transfer is a data stage or status stage.

If the SETUP bytes require a 3-stage control transfer, the TRB type used in the Data stage must be Control- Data for the first TRB of the Buffer Descriptor. When the host moves on to the Status stage, the TRB Type must be Control-Status-3.

If the SETUP bytes require a 2-stage control transfer, the TRB Type must be Control-Status-2.

The Status TRB is a zero-length TRB, with an unspecified Buffer Pointer value and a Buffer Size of zero. There is no data buffer associated with a Status TRB.

11.1.2.2.1.3 Link TRB Structure

The Link TRB is used to link back to the starting TRB for reusing TRBs in a circular fashion.

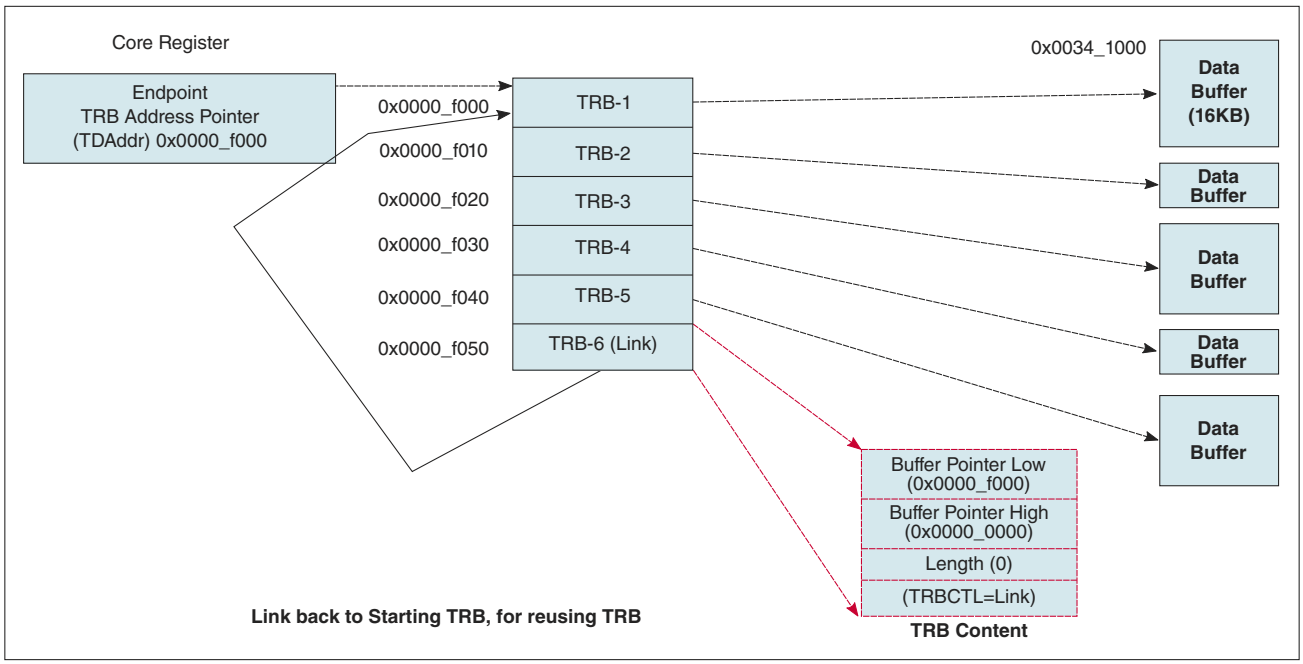


Figure 11-6. Link TRB Structure

If software prepares a circular TRB list which is shorter than the number of cached TRBs (four), the core automatically detects the loop and will not re-fetch a TRB that has already been fetched. For example, for four cached TRBs the software setup three TRBs in the following way:

- TRB-1: Normal, HWO=1
- TRB-2: Normal, HWO=1
- TRB-3: Link to TRB 1

The core will fetch TRBs 1, 2, and 3. Then the core will follow the link to TRB-1, note that its address is the same as the TRB-1 that has already been fetched into the cache, and will temporarily stop fetching TRBs. When TRB-1 completes due to traffic on the USB, the core will write TRB-1 back to memory with the HWO field set to '0', generate a XferInProgress event if necessary, and will automatically attempt to fetch TRB-1 again. In most cases, the core will see the HWO field set to '0' and will stop fetching until software updates the TRB, sets the HWO field back to '1', and issues an Update Transfer command.

11.1.2.2.1.4 Chaining buffers (CHN) and interrupt on completion (IOC) usage

This figure shows a chaining buffer example for an isochronous IN and a bulk IN transaction.

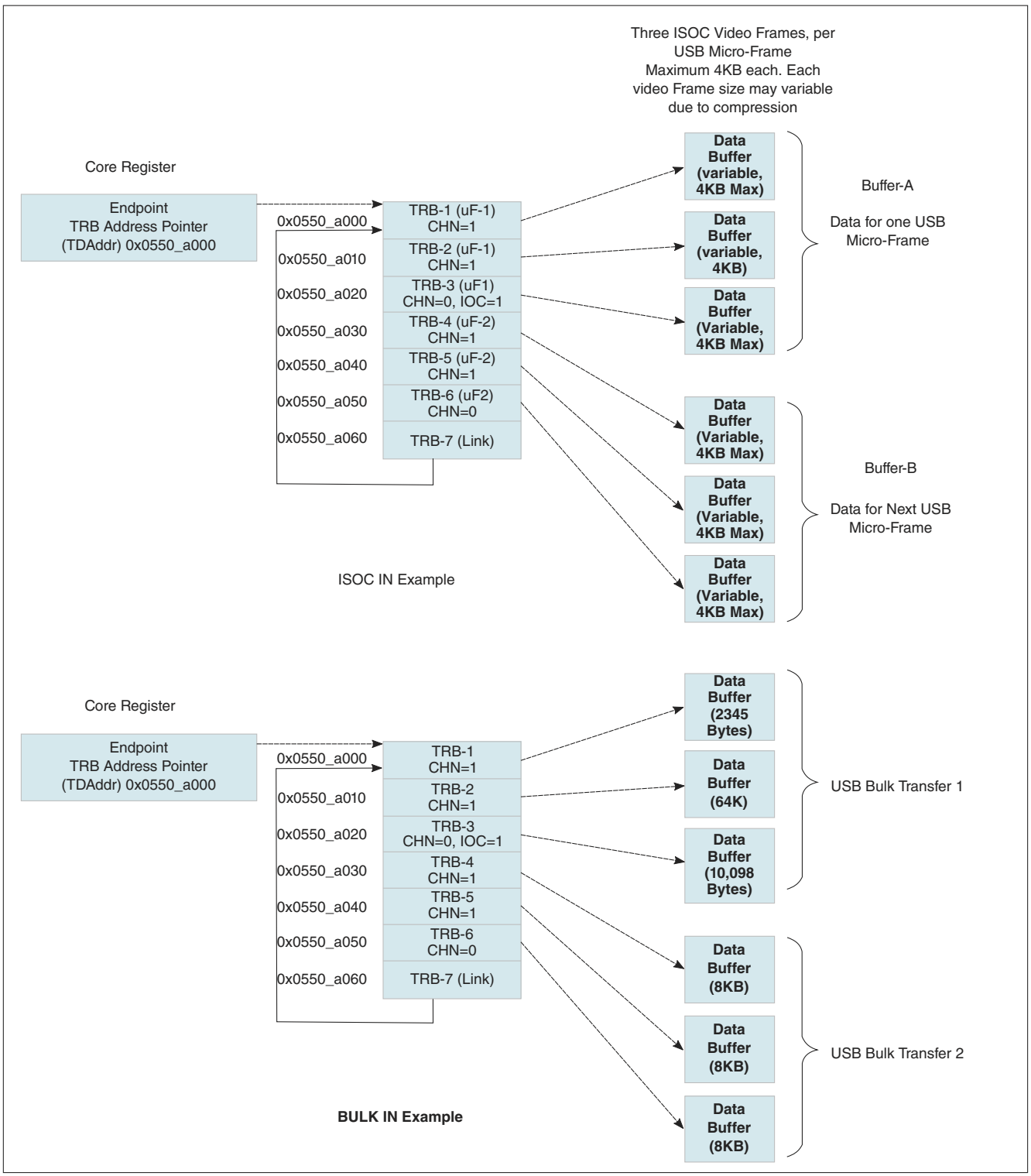


Figure 11-7. Chaining buffers for isochronous and bulk IN

In the isochronous IN application shown in the above figure, there are three video frames to be sent to the host in each microframe. Each video buffer size is maximum of 4 KB and the size may vary for each microframe depending on the compression. The CHN bit

in the TRB-3 is 0, which indicates this is the last buffer of a transfer for the given microframe. The device schedules TRB-1, 2, and 3 in the first bInterval. Similarly, TRB-6 indicates (CHN=0) microframe boundary, and TRB-4, 5, and 6 will be sent during the next bInterval.

For the last packet of last transfer in a microframe, the device internally sets last packet flag when responding at SuperSpeed for isochronous IN transfers.

The application can also use the IOC bit to receive an interrupt when the buffer-A transfer is completed, so it can use buffer-A to send data on the bInterval following the next bInterval. As long as the driver can service the interrupt and set up data before another bInterval, the USB transfers can continue without interruption. Depending upon the system interrupt latency, the buffer size can be adjusted.

If interrupt latency is high occasionally, the software may not have time to set up the linked TRB before the next bInterval. In this case, the core, on seeing HWO bit set to 0 in the TRB, stops processing further TRBs of this endpoint. When hardware receives the Update Transfer command from software, it will re-fetch the TRB. In the case of bulk application, the core will issue the NRDY signal. In isochronous transfers, zero- length packets are sent to the host until software enables the transfer again.

In the bulk IN example, the CHN bit indicates USB Transfer boundary. During a TRB transfer, CHN indicates whether to send the bytes from next TRB buffer as part of the current transfer. For example, even though TRB-1 has 2,345 bytes, the last 297 (2345 - 2 * 1024 = 297) bytes will be combined with the data in TRB-2 and sent as a 1,024 byte packet on the USB since CHN=1. On the other hand, when there is a short packet left in TRB-3, the short packet will be sent separately, since CHN=0.

The device writes data into TRB-1 and TRB-2 for the first transfer. The CHN bit (CHN=0) in TRB-2 indicates that this is the last buffer of the transfer, then TRB-3 and TRB-4 are written for the second transfer. Similarly, the CHN bit (CHN=0) in TRB-4 indicates the second transfer boundary.

If an interrupt latency is high, software may not have time to set up the TRB. In this case, the core, on seeing the HWO bit set to 0 in the TRB, will stop processing further TRBs for the endpoint. Hardware will re-fetch the TRB when software issues the update transfer command. In the case of bulk, control, or interrupt endpoints, the core will issue NRDY. In the case of isochronous endpoints, packets will be dropped until software enables the transfer again.

In the bulk OUT example below, when a short packet is received for TRB-1, TRB-3 will be used for the next OUT transfer and TRB-2 will be closed.

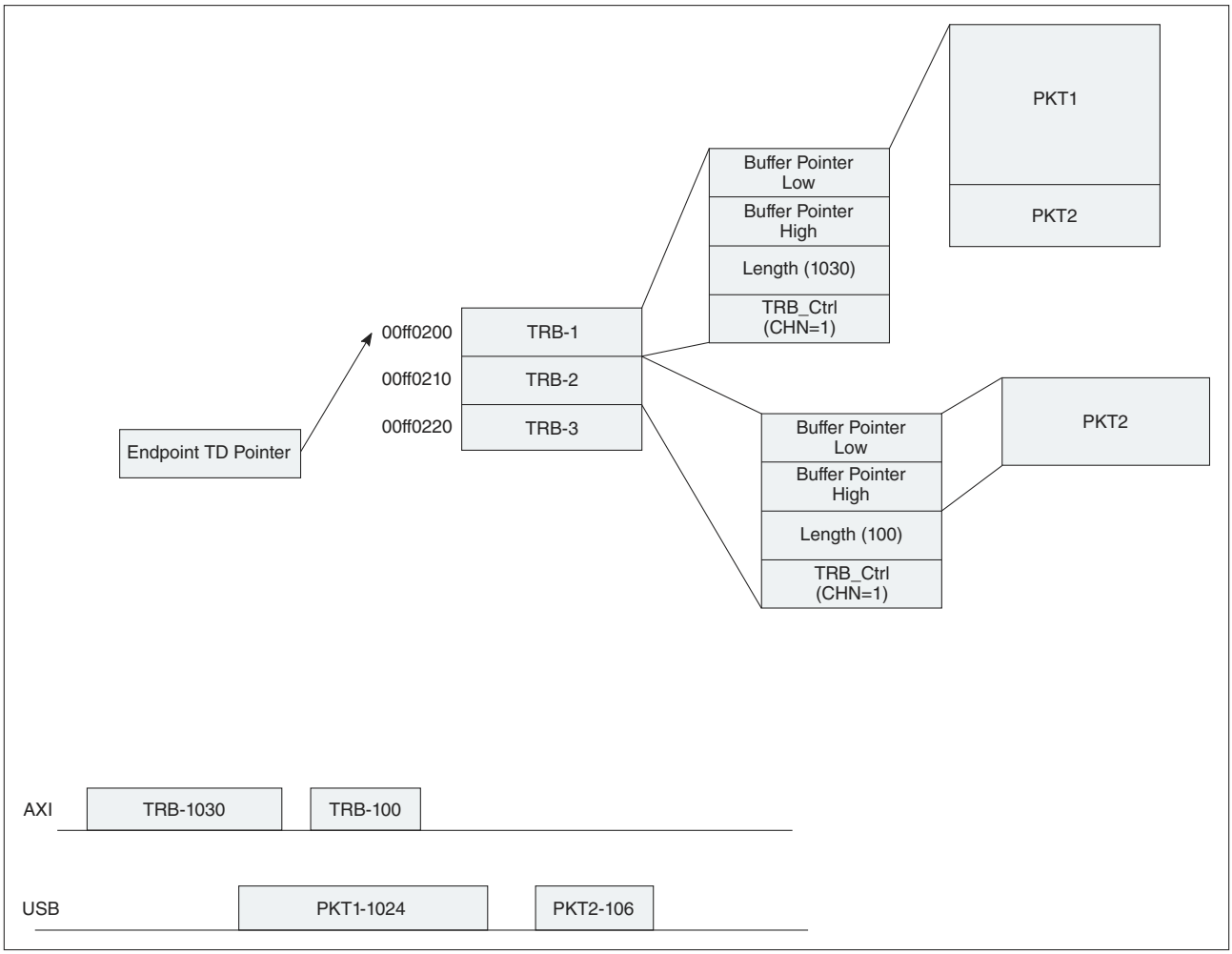


Figure 11-8. Chaining buffers, example 1

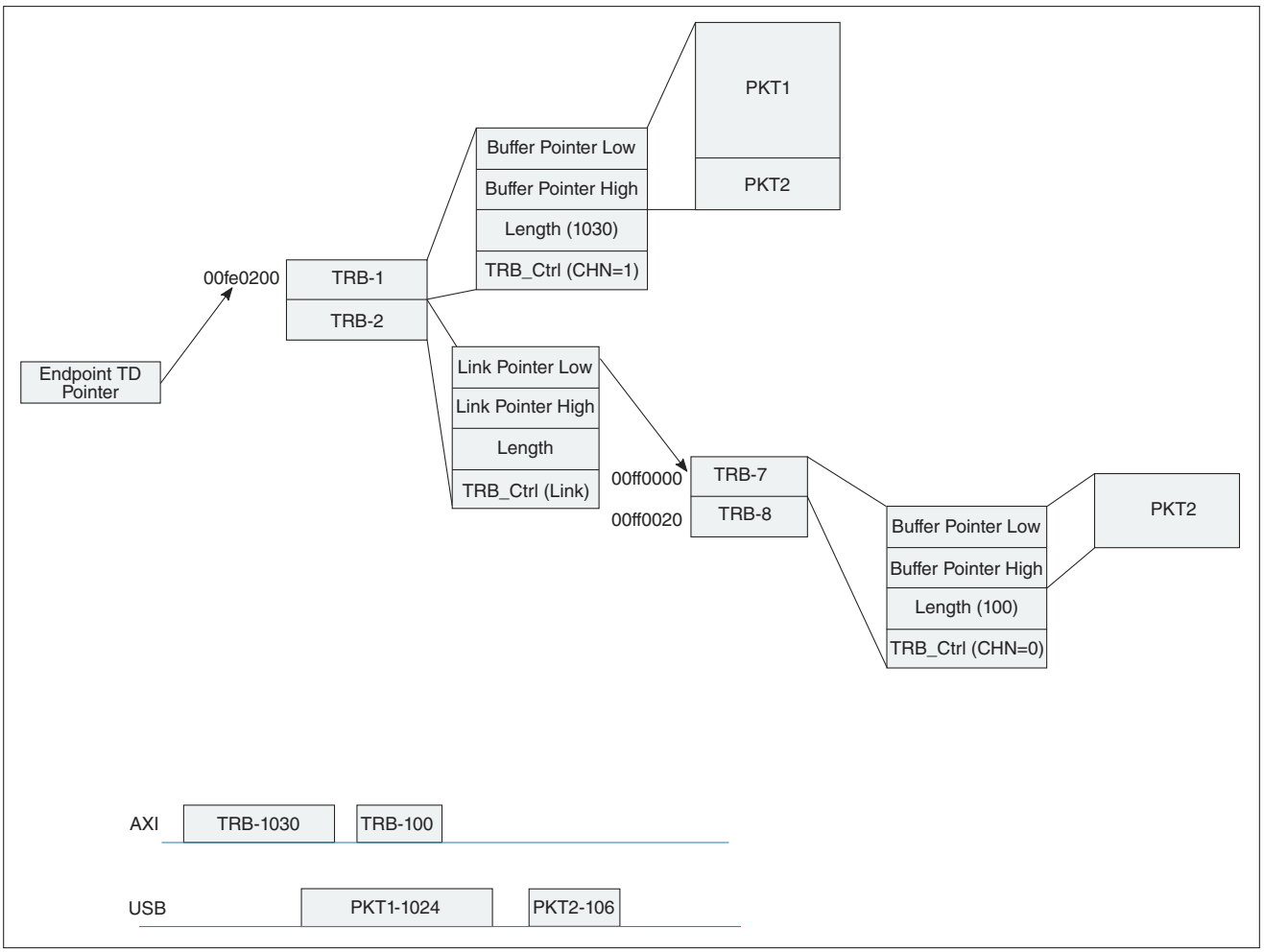


Figure 11-9. Chaining buffers, example 2

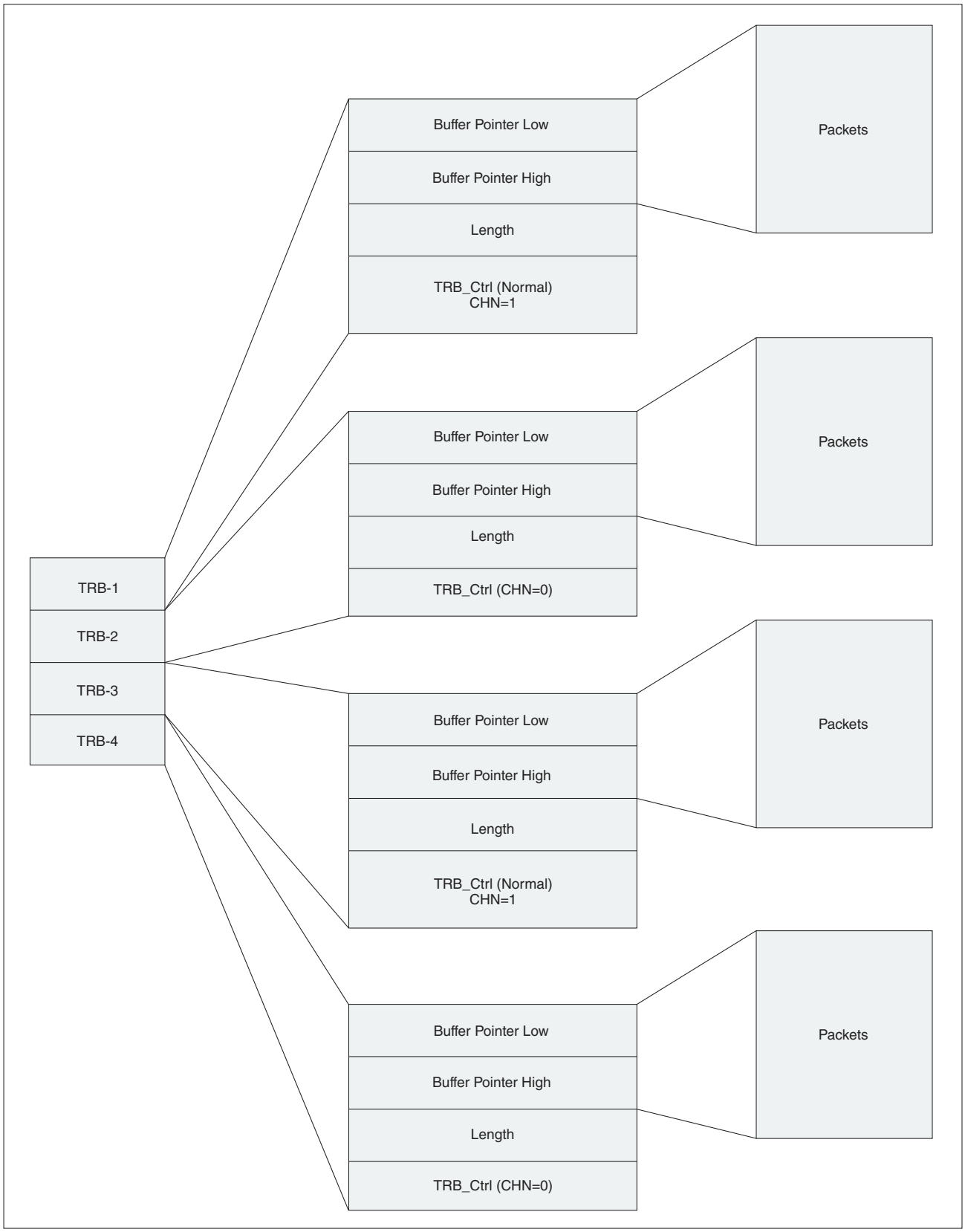


Figure 11-10. Bulk OUT with two transfers

11.1.2.2.1.5 Interrupt on Short Packet (ISP) and Continue on Short Packet (CSP) Usage

These two bits are used to schedule single or multiple OUT transfer. In most applications where only one OUT transfer per endpoint is scheduled, the software always sets ISP=1 and CSP=0. If the device receives a short packet, then it indicates a USB transfer completion through XferComplete events.

In an Ethernet-over-USB application, where software knows it is going to receive short packets, it can set up multiple transfer size buffers in one step by setting CSP=1 in all the TRBs. On a short packet, the device will update the byte count and move to next TRB. Software can also set ISP once in n number of TRBs to receive an XferInProgress event so it can process the previous short packets.

The following example shows software setting multiple 1500-byte Ethernet transfers and enabling the XferInProgress event once for four Ethernet packets to reduce the number of interrupts.

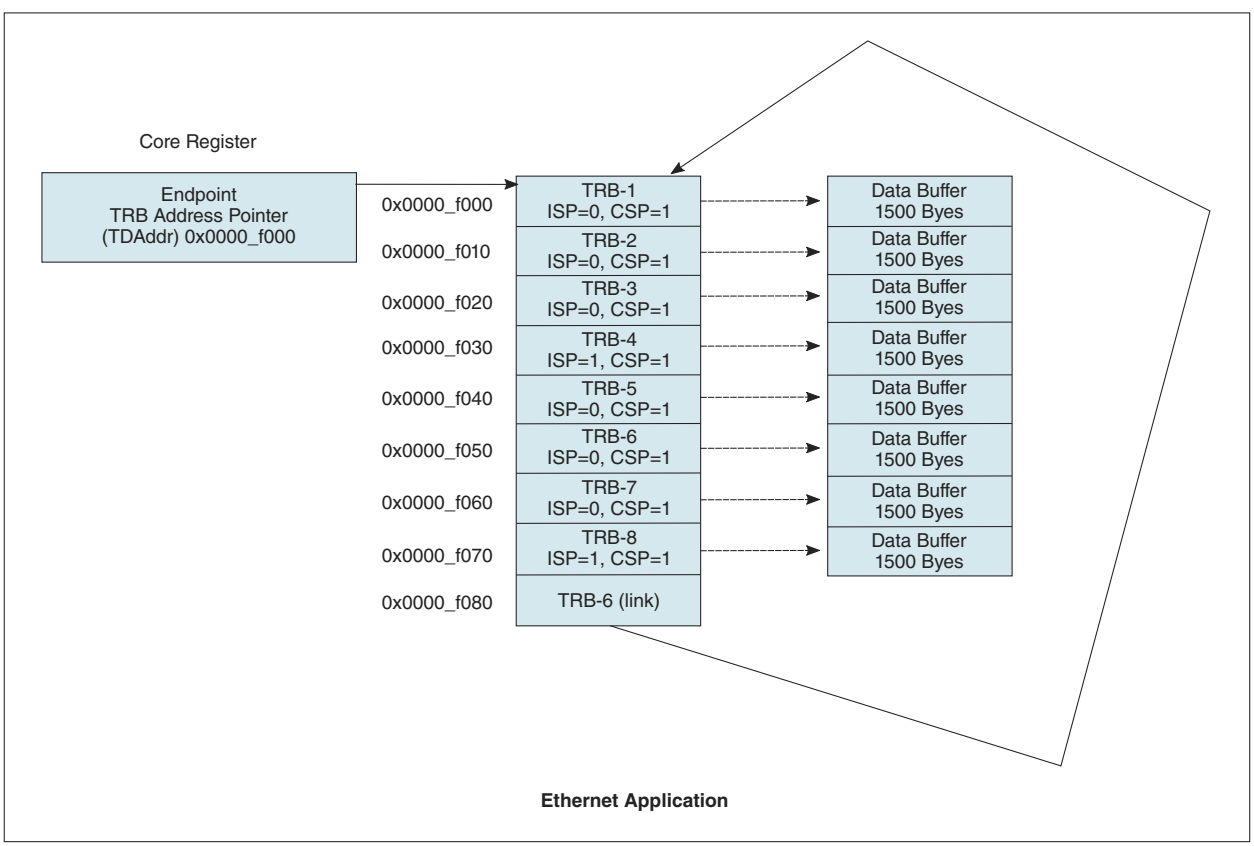


Figure 11-11. ISP and CSP Usage

11.1.2.2.1.6 Example of Setting Up TRBs

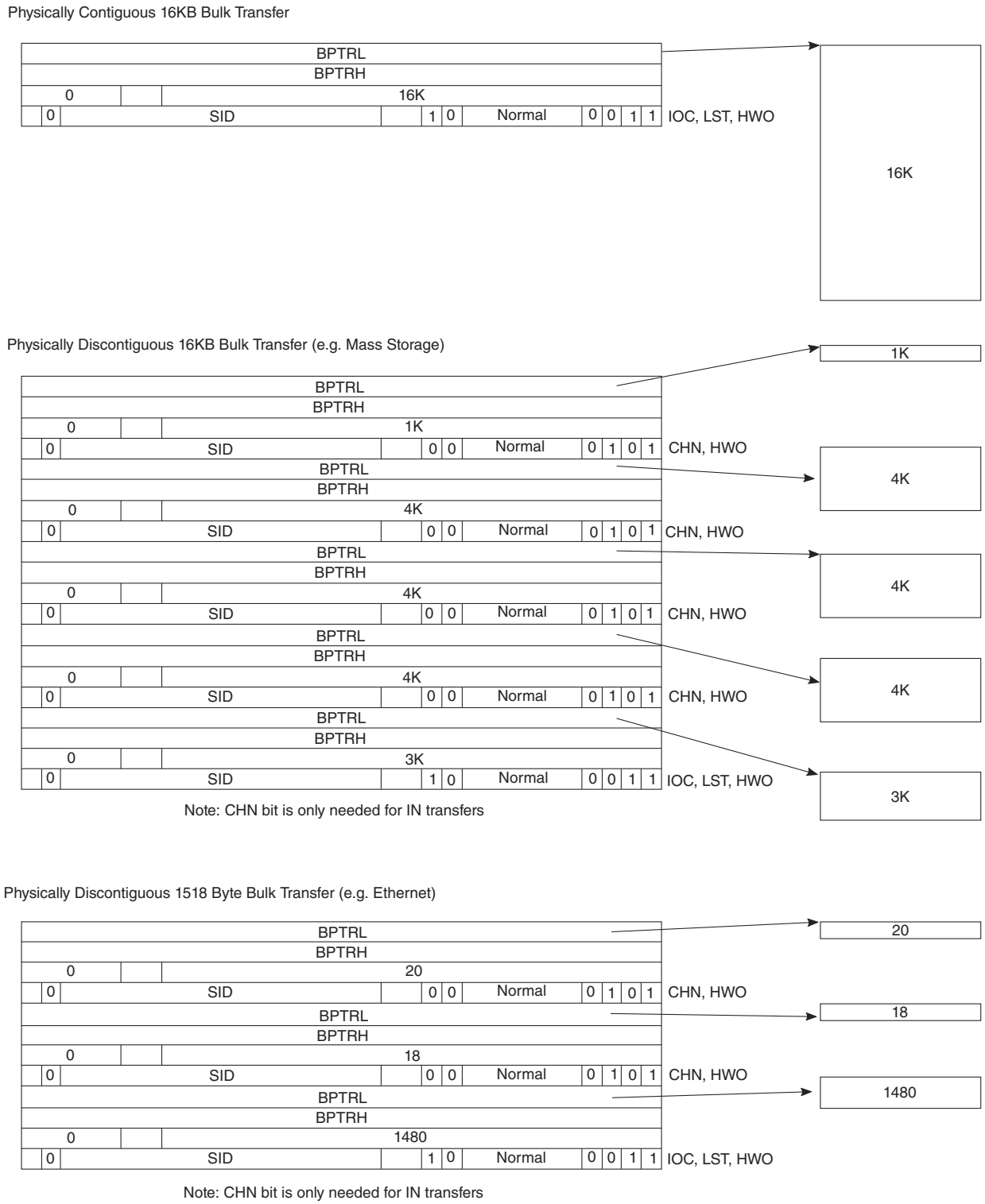


Figure 11-12. Bulk IN TRB Examples

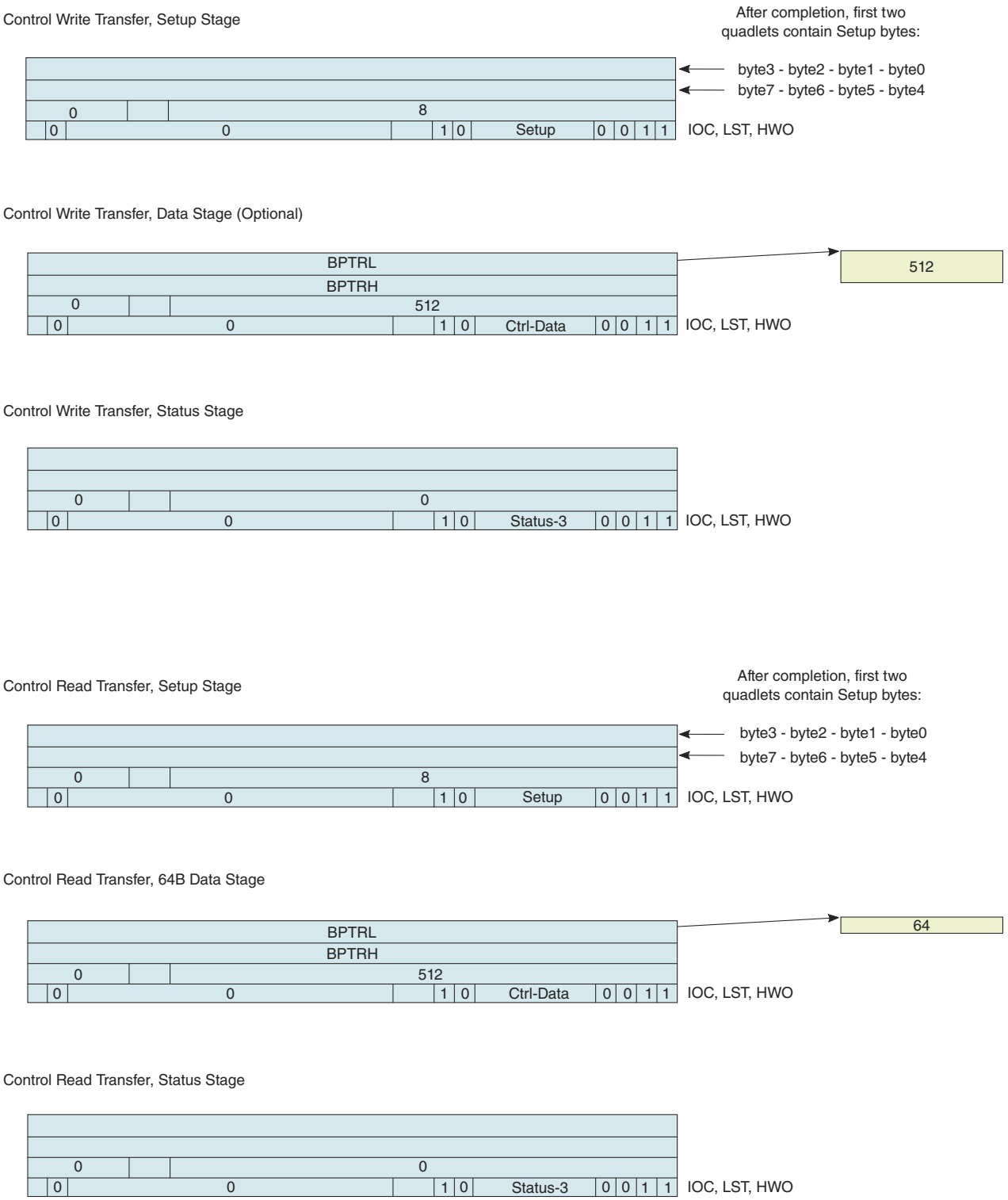


Figure 11-13. Setup/Control/Status TRB Examples

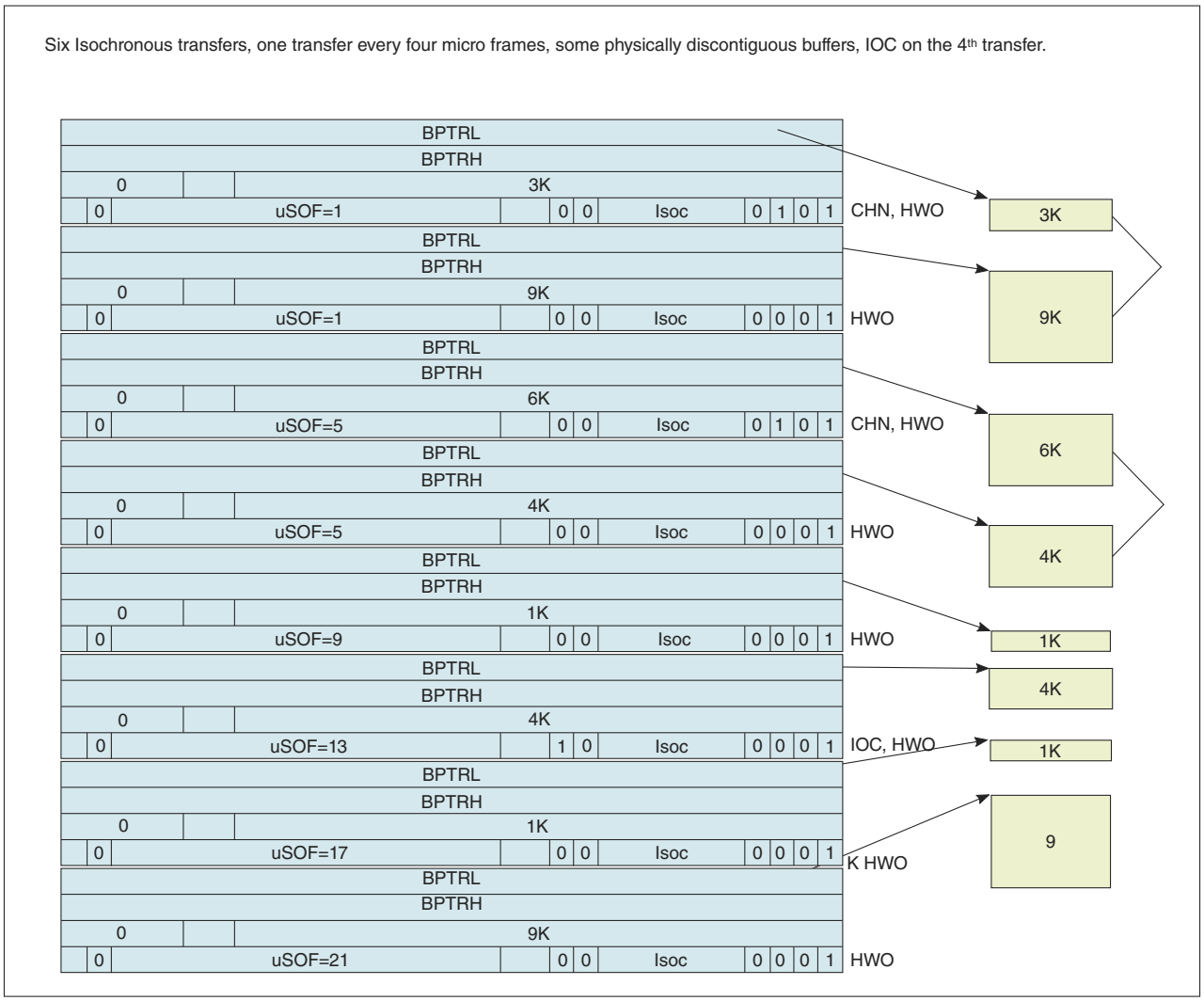


Figure 11-14. Isochronous IN TRB, Example 2

Eight Isochronous Transfers, Physically Contiguous Buffers, IOC on the 4th and 8th Transfers, Circular Linked

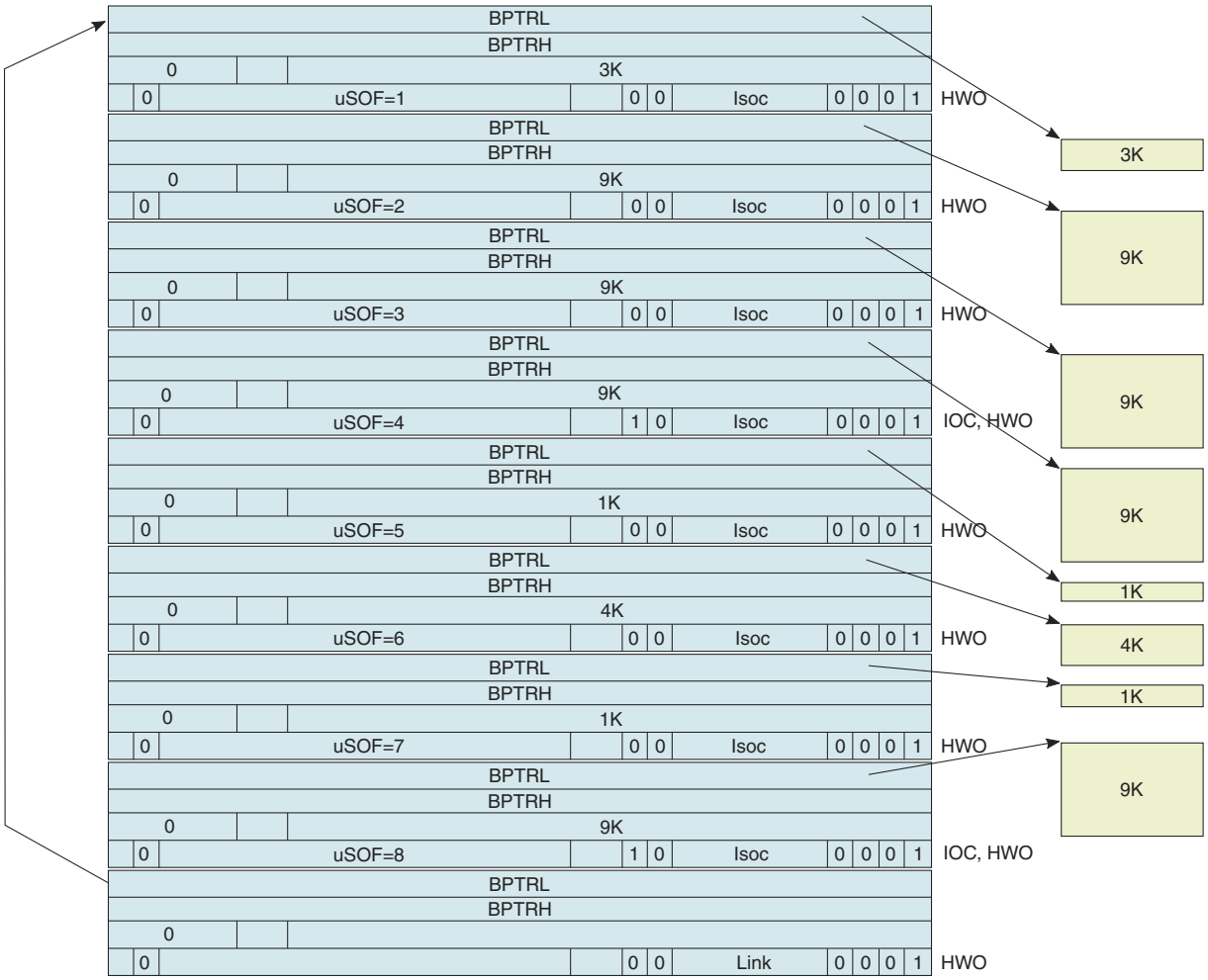


Figure 11-15. Isochronous IN TRB, Example 1

11.1.2.3 Device Programming Model

11.1.2.3.1 Register Initialization

The USB3 core contains global registers (prefixed by G*) and device registers (prefixed by D*) that are programmed to start operation and handle certain events. This section describes which registers must be accessed depending on the event that software is attempting to handle:

- Power-On or Soft Reset
- USB Reset Event
- Connect Done Event

- SetAddress Device Request
- SetConfiguration Device Request
- Disconnect Event
- Device-Initiated Disconnect and Reconnect

11.1.2.3.1.1 Device Power-On or Soft Reset

This section explains device core initialization after power-on or soft reset. The application must follow this initialization sequence for device mode operation.

When the core is first powered on, software initializes the following registers. The order of operations is not important, except for the first and last steps (DCTL[CSFTRST]=1 and DCTL[RUN_STOP]=1).

Table 11-3. Power-On or Soft Reset Register Initialization

Register	Description
DCTL	Set the CSftRst field to '1' and wait for a read to return '0'. This resets the device core.
GSBUSCFG0/1	Leave the default values, refer Global SoC Bus Configuration Register 0 (GSBUSCFG0) and Global SoC Bus Configuration Register 1 (GSBUSCFG1)
GTXTHRCFG/ GRXTHRCFG	This is required only if threshold is enabled.
GUSB2PHYCFG	Program PHY as per Global USB2 PHY Configuration Register (GUSB2PHYCFG) or leave the default values if the correct power-on values were selected. Note: The PHY must not be enabled for auto-resume in device mode.
GUSB3PIPECTL	Program the following PHY [DatWidth] or leave the default values if the correct power-on values were selected.
GTXFIFOSIZn	Write these registers to allocate prefetch buffers for each Tx endpoint. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GRXFIFOSIZ0	Write this register to allocate the receive buffer for all endpoints. Unless the packet sizes of the endpoints are application-specific, it is recommended to use the default value.
GEVNTADR/ GEVNTSIZ/ GEVNTCOUNT	Depending on the number of interrupts allocated, program the Event Buffer Address and Size registers to point to the Event Buffer locations in system memory, the sizes of the buffers, and unmask the interrupt. Note: USB operation stops if the Event Buffer memory is insufficient, because the core stops receiving/ transmitting packets.
GCTL	Program this register to override scaledown, RAM clock select, and clock gating parameters.
DCFG	Program device speed and periodic frame interval
DEVTEN	At a minimum, enable USB Reset, Connection Done, and USB/Link State Change events
DEPCMD0	Issue a DEPSTARTCFG command with DEPEVT.XferRscIdx set to 0 and CmdIOC set to 0 to initialize the transfer resource allocation. Poll CmdAct for completion.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command for physical endpoints 0 & 1 with the following characteristics, and poll CmdAct for completions: USB Endpoint Number = 0 or 1 (for physical endpoint 0 or 1) FIFONum= 0 XferNRdyEn and XferCmplEn = 1

Table continues on the next page...

Table 11-3. Power-On or Soft Reset Register Initialization (continued)

Register	Description
	Maximum Packet Size = 512 Burst Size = 0 EPType = 2'b00 (Control)
DEPCMD0/ DEPCMD1	Issue a DEPXFERCFG command for physical endpoints 0 & 1 with DEPCMDPAR0_0/1 set to 1, and poll CmdAct for completions
DEPCMD0	Prepare a buffer for a setup packet, initialize a setup TRB, and issue a DEPSTRTXFER command for physical endpoint 0, pointing to the setup TRB. Poll CmdAct for completion. Note: The core attempts to fetch the setup TRB via the master interface after this command completes.
DALEPENA	Enable physical endpoints 0 & 1 by writing 0x3 to this register
DCTL	Set DCTL[RunStop] to '1' to allow the device to attach to the host. At this point, the device is ready to receive SOF packets, respond to control transfers on control endpoint 0, and generate events.

After the controller has been started, wait for the following events:

- Wait for a DEVT.USBReset event. This indicates that a reset is detected on the USB.
- Wait for a DEVT.ConnectionDone event. This event indicates the end of the reset on the USB. On this event, read the DSTS register to get the connection speed.

11.1.2.3.1.2 Initialization on USB reset

To initialize the core as a device, during USB Reset, the application must perform the following steps:

Table 11-4. Initialization on USB reset

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DEPCMDn	Issue a DEPCSTALL (ClearStall) command for any endpoint in STALL mode prior to the USB Reset (excluding control endpoints)
DCFG	Set DevAddr to '0'

NOTE

- Special Reset Considerations for Default Control Endpoint 0: The default control endpoint is not affected by a USB Reset, so software must continue following the software flow for control transfers explained in "Control Transfer Programming Model" even across a USB Reset event. Resources can only be assigned to the default control endpoint once after a power on or soft reset.

11.1.2.3.1.3 Initialization on connect done

When this event is received, software must perform the following steps:

Table 11-5. Initialization on connect done

Register	Description
DSTS	Read this register to obtain the connection speed
GCTL	Program the RAMClkSel field to select the correct clock for the RAM clock domain. This field is reset to 0 after USB reset, so it must be reprogrammed each time on Connect Done.
DEPCMD0/ DEPCMD1	Issue a DEPCFG command (with Config Action set to "Modify") for physical endpoints 0 & 1 using the same endpoint characteristics from Power-On Reset, but set MaxPacketSize to 512 (SuperSpeed), 64 (High-Speed), 8/16/32/64 (Full-Speed), or 8 (Low-Speed).
GUSB2CFG/ GUSB3PIPECTL	Depending on the connected speed, write to the other PHY's control register to suspend it
GTXFIFOSIZn	(optional) Based on the new MaxPacketSize of IN endpoint 0, software may choose to re-allocate the TXFIFO sizes by writing to these registers.

11.1.2.3.1.4 Initialization on SetAddress request

When the application receives a SetAddress request in a SETUP packet, it performs the following steps:

Table 11-6. Initialization on SetAddress request

Register	Description
DCFG	Program the DCFG register with the device address received as part of the SetAddress request when SETUP packet is decoded.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the DCFG register is programmed with the new device address.

At this point, the device is ready to receive micro-SOF/ITP and is configured to receive control transfers on control endpoint 0 with a new address assigned.

11.1.2.3.1.5 Initialization on SetConfiguration or SetInterface Request

When the application receives a SetConfiguration or SetInterface request in a SETUP packet, it performs the following steps:

Table 11-7. Initialization on SetConfiguration or SetInterface Request

Register	Description
DALEPENA	Set this register to 0x3 to disable all endpoints other than the default control endpoint 0.
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0).
DEPCMD1	Issue a DEPCFG command (with Config Action field set to "Modify") for physical endpoint 1 using the current endpoint characteristics to re-initialize the TXFIFO allocation.
DEPCMD0	Issue a DEPSTARTCFG command with DEPCMD0.XferRscIdx set to 2 to re-initialize the transfer resource allocation.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is present in the new configuration (except for the default control endpoint). Note: Control endpoints are bi-directional and must use consecutive even/odd physical endpoint numbers for the OUT/IN direction (such as 2/3, or 4/5), and the FIFONum must be configured to the same value in both directions.
DEPCMDn	Issue a DEPXFERCFG command for each endpoint that is present in the new configuration (except for the default control endpoint 0).
GTXFIFOSIZn	(optional) Based on the new configuration of IN endpoints, software may choose to re-allocate the TXFIFO sizes by writing to these registers.
DALEPENA	Enable the logical endpoints that are active in the new configuration.
DEPCMD1	After receiving the XferNotReady(Status) event, acknowledge the status stage by issuing a DEPSTRTXFER command pointing to a Status TRB. This step must be done after the previous steps to ensure the host does not access any other endpoints that are being set up.

At this point, the core is prepared to accept Start Transfer commands for the newly-configured endpoints. For information on how to set up transfers, see the Operational Model.

11.1.2.3.1.6 Alternate initialization on SetInterface request

When handling a SetInterface device request, another possibility is that software may reconfigure existing endpoints instead of starting the configuration from the beginning. This is only possible if no TXFIFOs need to be reassigned. This flow also assumes that the endpoints that are being removed or reconfigured in the new interface have halted their traffic prior to the host issuing the SetInterface request.

Table 11-8. Alternate initialization on SetInterface request

Register	Description
DEPCMDn	Make sure there are no transfers still active on the endpoints that are changing in the new interface. This is normally ensured by the host prior to issuing the SetInterface, but if not, issue an End Transfer command for the transfer on each endpoint that is changing.
DEPCMDn	Issue a DEPCFG command (with the Config Action field set to "Initialize") for each endpoint that is changing in the new configuration. The side effect of the DEPCFG command is that the endpoint's sequence number is automatically set to 0.
DALEPENA	Write this register with all the endpoints that are enabled (including the ones that are not changing).
DEPCMDn	Using the StartXfer command, acknowledge the status stage response for the SetInterface request.

11.1.2.3.1.7 Initialization on Disconnect Event

When the application receives a Disconnect event, it must set DCTL[ULSTCHNGREQ] to 5. Other than this, the core does not require any initialization. Because the DCTL[RUN_STOP] bit is still '1', the device attempts to reconnect to the host, at which time a USB Reset and Connect Done event occurs. However, if the application does not want to attempt to reconnect to the host, it should perform the steps in the next section.

NOTE

When DCFG[DEVSPD] is programmed for 2.0 only mode (such as, High-Speed or Full-Speed), and if the application wants to issue any commands to clear any pending transfers during a Disconnect interrupt, then it has to disable USB_GUSB2PHYCFG[SUSPENDUSB20] before issuing any commands and re-enable it after the commands have completed.

11.1.2.3.1.8 Device-initiated disconnect

If the application wants to disconnect from the host, it should perform the following actions:

Table 11-9. Initialization on device-initiated disconnect

Register	Description
DEPCMD0	If a control transfer is still in progress, complete it and get the core into the "Setup a Control-Setup TRB / Start Transfer" state
DEPCMDn	Issue a DEPENDXFER command for any active transfers (except for the default control endpoint 0)
DCTL	Set DCTL[RUN_STOP] to '0' to disconnect from the host
DSTS	Poll [DEVCTRLHLT] until it is '1'

At this point, the device is disconnected from the host and will not attempt to reconnect.

11.1.2.3.1.9 Reconnect after Device-Initiated Disconnect

If the application decides to reconnect to the host, it must follow the steps in Device Power-On or Soft Reset.

11.1.2.3.2 Operational Model

The following sections describes the processes and data structures that the core uses to implement the USB 3.0 specification.

11.1.2.3.2.1 USB and Physical Endpoints

Endpoints are referred to in two ways:

- As a USB endpoint number: USB endpoints are defined in the USB specification.
- As a physical endpoint resource number: The hardware has a fixed number of physical endpoint resources. Each resource is unidirectional and can be configured to refer to either direction of any USB endpoint.

The software always works on physical endpoints and it knows how physical endpoint corresponds to USB endpoints. DALEPENA is the only register that has one enable bit per USB endpoint.

During SetConfiguration, software maps physical endpoint resources to the required USB endpoints. When a USB request comes in, the USB endpoint number gets converted to a physical endpoint number in the core. Similarly, when a USB packet is sent out, the physical endpoint number is converted to the USB endpoint and sent out.

A USB control endpoint requires two physical endpoints. One physical endpoint is mapped to the OUT direction of the Control endpoint, and the other one is mapped to the IN direction of the Control endpoint. Specifically the two physical endpoints are used as the following:

- The Setup stage of any control transfer uses the OUT direction physical endpoint.
- For a control write or 2-stage transfer, the Data stage (if present) uses the OUT direction physical endpoint and the Status stage uses the IN direction physical endpoint.
- For a control read transfer, the Data stage uses the IN direction physical endpoint and the Status stage uses the OUT direction physical endpoint.

11.1.2.3.2.2 Event Buffers

Hardware passes command completion, transfer progress, and asynchronous events to software through one or more Event Buffers.

To configure an Event Buffer, the software performs the following steps:

1. Sets up an empty buffer in system memory.

2. Writes the address of the beginning of the buffer into GEVNTADR. This address must be aligned to the Event Buffer size.
3. Writes the size of the buffer and interrupt mask into GEVNTSIZ. Depending on your system interrupt latency, enough Event Buffer space must be allocated to avoid lost interrupts or reduced performance.
4. Write a 0 into the GEVNTCOUNT register. This must be the last step, as it enables the Event Buffer. After the Event Buffer has been configured, software must not change the size or address.

There is one interrupt line per Event Buffer that indicates there are one or more events present. Software reads one or more events out of the buffer and indicates to hardware how many events it processed by writing the byte count to the GEVNTCOUNT register.

Clock crossing delays may result in the interrupt's continual assertion after software acknowledges the last event. Therefore, when the interrupt line is asserted, software must read the GEVNTCOUNT register and only process events if the GEVNTCOUNT is greater than 0.

The first event produced by the core after the Event Buffer is configured will be written to the address specified in GEVNTADR. Most events are 32 bits, and subsequent events will be written to the address (PreviousEventAddress + 4). When that address exceeds the sum of GEVNTADR and GEVNTSIZ, the core wraps around to the first GEVNTADR value. In this way, the Event Buffer operates like a circular buffer with hardware writing to the "tail" of the buffer and software reading from the "head."

Most events are exactly 4 bytes in size, but there is one exception: The Vendor Device Test LMP Received Event (VndrDevTstRcved) is a 12 byte event that includes a header in the first 4 bytes and the contents of the LMP in the following 8 bytes.

When an event occurs within the core, hardware checks the enable bit that corresponds to the event to decide whether the event will be written to the Event Buffer or not. The Event Buffer contains one of the following types of information, depending on the value of the lower bits of the event:

- Device Endpoint-Specific Event (DEPEVT) (Event[0] = 0x0)
  - The DEPCFG endpoint-specific command specifies the bits that are enabled and which Event Buffer to use for these events.
- Device-Specific Event (DEVT) (Event[0] = 0x1, Event[7:1] = 0x00)
  - The Generic Command Complete event is enabled through the DGCMD[CmdIOC] field when the command is issued.
  - The Event Buffer Overflow Event cannot be disabled and is written to the Event Buffer that encounters the overflow.

- The rest of the Device-Specific events are enabled through the DEVTEN register.
- Except for the Event Buffer Overflow Event, these events are written to the Event Buffer specified in the DCFG[IntrNum] field.

The core always leaves one entry free in each Event Buffer. When the Event Buffer is almost full, hardware writes the Event Buffer Overflow event and the USB will eventually get stalled when endpoints start responding NRDY or the link layer will stop returning credits (in SuperSpeed). This event is an indication to software that it is not processing events quickly enough. During this time, events will be queued up internally. When software frees up Event Buffer space, the queued up events will be written out and the USB will return to normal operation.

Event Buffer Content for Device Endpoint-Specific Events (DEPEVT)

Table 11-10. Device Endpoint-n Events: DEPEVT

Field	Description
31–16	Event Parameters (EventParam) For XferNotReady, XferComplete, and Stream events on Bulk Endpoints: [31-16]StreamID. Applies only to bulk endpoints that support streams. This indicates the Stream ID [1] of the transfer for which the event is generated For XferInProgress: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number of the beginning of the interval that generated the XferInProgress event (debug purposes only) For XferNotReady events on Isochronous Endpoints: [31-16] Isochronous Microframe Number (IsocMicroFrameNum). Indicates the microframe number during which the endpoint was not ready **NOTE:** USB 3.0 core represents USB bus time as a 14-bit value on the bus and also in the DSTS register (DSTS[SOFFN]), but as a 16-bit value in the XferNotReady event. Use the 16-bit value to interact with Isochronous endpoints via the StartXfer command. The extra two bits that the USB 3.0 core produces will be necessary for handling wrap-around conditions in the interaction between software and hardware. EPCmdCmplt events For all EPCmdCmplt events [27-24]- Command Type. The command type that completed (Valid only in a DEPEVT event. Undefined when read from the DEPCMD.EventParam field). For EPCmdCmplt event in response to Start Transfer command. [22-16] Transfer Resource Index (XferRscIdx). The internal hardware transfer resource index assigned to this transfer. This index must be used in all Update Transfer and End Transfer commands.
15–12	Event Status (EventStatus) Within an XferNotReady event- [15]: Indicates the reason why the XferNotReady event is generated: 0- XferNotActive- Host initiated a transfer, but the requested transfer is not present in the hardware 1- XferActive- Host initiated a transfer, the transfer is present, but no valid TRBs are available

Table continues on the next page...

Table 11-10. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	[14]: Not Used [13:12]: For control endpoints, indicates what stage was requested when the transfer was not ready: 2'b01- Control Data Request 2'b10- Control Status Request Within an XferComplete or XferInProgress event- [15]: LST bit of the completed TRB (XferComplete only) [15]: MissedIsoc: Indicates the interval did not complete successfully (XferInProgress only) [14]: IOC bit of the TRB that completed [13]: Indicates the TRB completed with a short packet reception or the last packet of an isochronous interval [12]: Reserved If the host aborts the data stage of a control transfer, software may receive a XferComplete event with the EventStatus field equal to '0'. This is a valid event that must be processed as a part of the Control transfer programming model. Within a Stream Event- [15:12]: - 4'h2- StreamNotFound- This stream event is issued when the stream-capable endpoint performed a search in its transfer resource cache, but could not find an active and ready stream. - 4'h1- StreamFound- This stream event is issued when the stream-capable endpoint found an active and ready stream in its transfer resource cache, and initiated traffic for that stream to the host. The ID of the selected Stream is in the EventParam field. In response to a Start Transfer command- 4'h2- Indicates expiry of the bus time reflected in the Start Transfer command. 4'h1- Indicates there is no transfer resource available on the endpoint. In response to a Set Transfer Resource (DEPXFERCFG) command- 4'h1- Indicates an error has occurred because software is requesting more transfer resources to be assigned than have been configured in the hardware. In response to a End Transfer command- 4'h1- Indicates an invalid transfer resource was specified.
11–10	This field is reserved.
9–6	4'h7- Endpoint Command Complete (EPCmdCmplt) Indicates software may issue another Device Endpoint command to the endpoint. When issued in response to an End Transfer command, indicates that DMA stopped for the endpoint. For all other commands, this event does not imply that all effects of the commandtook place. The DEPCMD register contains the same status information present in the Event Status Bits field. 4'h6- Stream Event (StreamEvt) Indicates that a stream-capable endpoint initiated a search within its transfer resource cache. The result of the search is in the EventStatus field (Found or NotFound). 4'h5- Reserved 4'h4 - Reserved

Table continues on the next page...

Table 11-10. Device Endpoint-n Events: DEPEVT (continued)

Field	Description
	4'h3- XferNotReady Event (XferNotReady) Indicates receipt of a transaction when no TRBs are available for the endpoint. For isochronous IN endpoints, a zero-length packet is automatically sent by hardware and NRDY for non-isochronous endpoints. The application must enable this event if it plans to issue Start Transfer on demand. This event can happen when software issues a Start Transfer or Update Transfer. In this case, software must ignore this event because it has already issued the Start Transfer or Update Transfer. XferNotReady is generated when the core responds NRDY to the host on the USB. It is useful in the beginning of a transfer when software wants to wait for the host to start polling the endpoint before setting up TRBs, but it is not efficient to use this event to determine when the core has run out of TRBs. For determining when the core has processed TRBs and needs software to setup more, use the IOC field in a TRB along with the XferInProgress event that is generated when the core completes a TRB with IOC=1. For non-stream-capable endpoints, the hardware filters multiple events if the host continues transactions, even after the core responds with NRDY. This internal filter is reset after software issues any endpoint command to this endpoint. For stream-capable endpoints, this event is generated each time the host attempts a transaction, even if the core responds with NRDY. For isochronous endpoints, this event is generated only once prior to the Start Transfer command to communicate the current bus time. For additional information, see the Event Status field. 4'h2- XferInProgress Event (XferInProgress) Applies to IN and OUT endpoints. Indicates an EP/Stream specific event happened and it is continuing the transfer. For additional information, see the Device Programming Model. 4'h1- XferComplete Event (XferComplete) Applies to IN and OUT endpoints. Indicates that a EP/Stream transfer completed and the core stopped the transfer. For additional information, see the Device Programming Model. 4'h0- Reserved
5–1	Physical Endpoint Number (0-31)
0	1'b0- Indicates that this is an endpoint-specific event.

1. Note that term Stream ID used in USB chapter is different from Arm SMMU terminology.

Event Buffer Content for Device-Specific Events (DEVT)

Table 11-11. Device-specific events: DEVT

Field	Description
31-25	Reserved
24-16	Event Information Bits (EvtInfo) For a USB/Link State change Event, this field indicates the state of the link- EvtInfo[8:5] - HIRD value received from the LPM token (valid for a Hibernation Request Event) EvtInfo[4] - SuperSpeed event. Set to 1 for SS; Set to 0 for non-SS.

Table continues on the next page...

Table 11-11. Device-specific events: DEVT (continued)

Field	Description
	EvtInfo[3:0] - Link State. Indicates link state at the time of the event. Follows same encoding as DSTS link state bits.
15-12	Reserved
11-8	11- Event Buffer Overflow Event (EvntOverflow) The core writes this event to indicate there is no space in the event buffer, and one or more Device-specific events may have been dropped after this event. Endpoint-Specific events will not be dropped, but they will be delayed which can cause a drop in performance on the USB. Software uses this event as a warning that the Event Buffer is too small and the core should be reconfigured with a larger Event Buffer or software needs to process events more quickly. In order to avoid repeated Event Buffer Overflow Events, software must free up space in the Event Buffer by acknowledging more than 1 event (writing a value greater than 4 to the GEVNTCOUNT register). This event cannot be disabled, and is always written to the Event Buffer that encounters the overflow. 10- Generic Command Complete Event (CmdCmplt) The core writes this event to indicate the generic command is complete. 9- Erratic Error Event (ErrticErr) The core writes this event to report erratic errors (phy_rxvalid_i/phy_rxvldh_i or phy_rxactive_i is asserted for at least 2 ms, due to PHY error) seen on the UTMI+. Due to erratic errors, the USB 3.0 core goes into Suspended state and a USB/Link state change event (ULStChng) is generated to the application. If the early suspend is asserted due to an erratic error, the application can only perform a soft disconnect recover. For SuperSpeed operation, the PHY erratic error event indicates that the PIPE is not responding to the PHY command (for example, pipe3_PowerDown change or receiver detection). After the core requested any PHY command, if it did not receive pipe3_PhyStatus within 100ms, the core will timeout and generate the erratic error event. After generating the erratic error, the core assumes that the PHY command is completed successfully and proceeds with the next pending PHY command. Software must reset the controller on receiving the erratic error. 8- Reserved 7- Start of (micro)Frame (Sof) 6- USB Suspend Entry Event This event provides a notification that the link has gone to a suspend state (L2, U3, or L1). This event is generated when hibernation mode is disabled. The existing Link State Change event (3) provides the same information, but is generated for every link state change. 5- Hibernation Request Event This event provides a notification that the link has gone to a suspend state where hibernation is supported (L2, U3, or L1) and requires software intervention - When DCTL.KeepConnect=1 and L1HibernationEn=0, this event is generated only when the link goes to L2 or U3 - When DCTL.KeepConnect=1, L1HibernationEn=1, and the core is configured to support LPM (DCTL.LPMCap=1 and DCTL.HIRD_Thres[4]=1), this event is generated when the link goes to L2, U3, or L1 with an HIRD value greater than the DCTL.HIRD_Thres[3-0] value The existing Link State Change event (3) provides the same information, but is generated for every link state change. If both events are enabled, only the Hibernation Request event will be generated if the conditions are met. When this event occurs, software is required to start the hibernation process

Table continues on the next page...

Table 11-11. Device-specific events: DEVT (continued)

Field	Description
	EvtInfo[4:0] indicates the Link State and Speed which caused the Hibernation Event (using the same encoding as the Link State Change event. Suspend is 3, and L1 is 2)EvtInfo[8:5] provides the HIRD value from the host's LPM tokenFor L2 and U3, this event is only generated when DCTL.KeepConnect=1For L1, this event is only generated when DCTL.L1HibernationEn=1 and DCTL.KeepConnect=1In addition to the suspend states that trigger this event, hibernation is also supported when the link is disconnected. See 'Device-Initiated Disconnect and Low Power While Disconnected' for details. 4- Resume/Remote Wakeup Detected Event (WkUpEvt) This event is generated when the host initiates a resume condition on the USB bus. This event is not generated by the device initiating a remote wakeup to the host. 3- USB/Link State Change (ULStChng) The core writes this event to indicate a change of USB or Link state. Bits 23-16 (EvtInfo) of this event indicate the Link Status. The event is generated in SuperSpeed when the link LTSSM state changes, except when LTSSM exits from HOT_RESET, POLL or LTSSM enters into RECOVERY. 2- Connection Done (ConnectDone) 1- USB Reset (USBRst) 0- Disconnect Detected Event (DisconnEvt) Note: This event is generated only if VBUS is off.
7-1	7'h00 - Device Specific Event
0	1'b1 - Non-Endpoint-Specific Event

11.1.2.3.2.3 Transfer and Buffer Rules

Buffers that are used to transfer data to and from an endpoint are defined using a Buffer Descriptor, which consists of one or more Transfer Request Blocks (TRBs). A CHN flag in the TRB is used to identify the TRBs that comprise a Buffer Descriptor. Therefore, a Buffer Descriptor refers to a consecutive set of TRB data structures where the CHN flag is set in all TRBs, except the last. Note that a Buffer Descriptor may consist of a single TRB, whose CHN flag will not be set.

Software communicates the location of the first TRB by using the Start Transfer command on an endpoint. Even endpoints that are not stream-capable use this command.The core fetches TRBs from external memory starting at the address provided in the Start Transfer command parameters, continuing in a linear fashion and following the Link TRB until a TRB is encountered that has its LST bit set to '1' or HWO bit set to '0'. Therefore, software must ensure that it has valid TRBs prepared before issuing the Start Transfer command to prevent the core from reading uninitialized memory.

Descriptor fetch requests are buffered within the hardware and handled separately from the progress of the current transfer. Therefore, it is possible that the core completes a transfer with XferComplete but still continues reading TRBs that it has not cached yet. If

software is immediately de-allocating the memory for TRBs based on the XferComplete event, it is recommended that software issue an End Transfer command for the endpoint/transfer resource prior to de-allocating the memory. The completion of the End Transfer command flushes out any pipelined descriptor fetches and avoids a potential bus error.

While processing TRBs, two conditions may cause the core to write out an event and raise an interrupt line:

- TRB Complete:
  - For OUT endpoints, a packet is received which reduces the remaining byte count in the TRB buffer to zero.
  - For IN endpoints, an acknowledgement is received for a transmitted packet which reduces the remaining byte count in the TRB buffer to zero.
- Short Packet Received:
  - For OUT endpoints only. While writing to a TRB buffer, the endpoint receives a packet that is smaller than the endpoint's MaxPacketSize.

This table shows the USB 3.0 Core Actions Based on TRB Control Bits.

Table 11-12. USB 3.0 Core Actions Based on TRB Control Bits

Direction	TRB Complete Packet	ISP	IOC	CHN	LST	CSP	Action
IN	Yes	-	X	0	1	-	XferComplete event
IN	Yes	-	0	X	0	-	No event
IN	Yes	-	1	X	0	-	XferInProgress event
OUT	Yes No	-	X	0	1	-	XferComplete event
OUT	Yes No	-	0	X	0	-	No event
OUT	Yes No	-	1	X	0	-	XferInProgress event
OUT	X Yes	X	X	X	X	0	XferComplete event [1]
OUT	X Yes	X	X	0	1	1	XferComplete event
OUT	X Yes	X	X	1	0	1	Search for CHN=0, accumulate IOC and ISP, then follow CHN=0 rules [2]
OUT	X Yes	X	1	0	0	1	XferInProgress event
OUT	X Yes	0	0	0	0	1	No event
OUT	X Yes	1	0	0	0	1	XferInProgress event

1. When a TRB receives whose CSP bit is 0 and CHN bit is 1 receives a short packet, the chained TRBs that follow it are not written back (for example, the BUFSIZ and HWO fields remain the same as the software-prepared value)
2. In the case of an OUT endpoint, if the CHN bit is set (and CSP is also set), and a short packet is received, the core retires the TRB in progress and skip past the TRB where CHN=0, accumulating the ISP and IOC bits from each TRB. If ISP or IOC is set in any TRB, the core generates an XferInProgress event. Hardware does not set the HWO bit to 0 in skipped TRBs. If the endpoint type is isochronous, the CHN=0 TRB will also be retired and its buffer size field updated with the total number of bytes remaining in the BD.

On XferComplete and XferInProgress events, status bits in the event indicate LST, IOC, Short Packet Received, or Bus Error status. In the "fast-forward" case, the IOC status is accumulated from the skipped TRBs, not just the TRB that received the short packet.

When the hardware writes back the TRBs, it updates the BUFSIZ field to represent the remaining unused buffer.

11.1.2.3.2.3.1 Number of TRBs Rule

- Software must set up only one TRB for a control setup or status stage.
- If software is preparing multiple transfers for an IN endpoint, it may be necessary to place a 0-length TRB between transfers that are MaxPacketSize aligned to indicate transfer boundaries to the host. This is not necessary, if the class driver and the host can handle bursting between transfers.
- If software is preparing multiple transfers for an OUT endpoint, it needs to place a MaxPacketSize TRB between transfers, if it expects the host to transmit a 0-length packet between transfers.

11.1.2.3.2.3.2 TRB Control Bit Rules

Transfer control bits must conform to the following restrictions:

For OUT endpoints, the CSP bit must be the same in every TRB within a Buffer Descriptor (either set or clear).

- The core autonomously checks the HWO field of a TRB to determine if the entire TRB is valid.
- Therefore, software must ensure that the rest of the TRB is valid before setting the HWO field to '1'. In most systems, this means that software must update the fourth DWORD of a TRB last. Every time software validates a TRB by setting HWO=1, it must also issue an Update Transfer command to the core.
- Software sets the HWO bit to 1 when it creates the TRB, and cannot modify it until hardware resets it to 0. However,
  - Software must detect when a "fast-forward" occurs on an OUT endpoint that receives a short packet, since some TRBs in a chain may still have their HWO bit set to 1 while belonging to software.
  - Hardware will not clear the HWO bit of a Link TRB. Therefore, software can only modify a Link TRB if the TRB prior to the Link TRB has its HWO bit set to 0.
- The LST bit must not be set to 1 for isochronous endpoints.
- For a Setup or Status TRB, set CHN=0, LST=1, and CSP=0.
- For the data stage of a control transfer, set CSP=0 in all TRBs and LST=1 in the last TRB of the data stage.

- For Link TRBs, the LST, CHN, IOC, ISP, CSP, and Stream ID fields are ignored and must be set to 0.
- The Link TRB's chain bit is implicitly equal to the chain bit of the TRB before it. If the TRB before it has CHN=1, then the Link acts as if it's CHN=1. If the TRB before it has CHN=0, then the Link acts as if it's CHN=0. A Link TRB cannot be the last TRB in a Buffer Descriptor.
- When CSP=1 and the core receives a short packet, it searches for the end of the Buffer Descriptor by finding the Normal TRB that has CHN=0. Therefore, it is illegal to setup a circular buffer with all Normal TRBs with CHN=1.

11.1.2.3.2.3.3 Buffer size rules and zero-length packets

The hardware contains a cache that holds four TRBs per transfer.

For IN endpoints, the following rules apply:

- The number of chained TRBs necessary to construct a single packet must never exceed 3 . A maximum of one link TRB can be present in the chain.
- If software wants to indicate a transfer completion to the host by sending a zero-length packet after a multiple of MAXPACKETSIZE, it must set up a zero-length TRB following the last TRB in the transfer.

For OUT endpoints, the following rules apply:

- If the first TRB has CHN=1, its buffer size must be ≥ 1 byte
- The total size of a buffer descriptor must be a multiple of MAXPACKETSIZE
- A received zero-length packet still requires a MAXPACKETSIZE buffer. Therefore, if the expected amount of data to be received is a multiple of MAXPACKETSIZE, software should add MAXPACKETSIZE bytes to the buffer to sink a possible zero-length packet at the end of the transfer.

For IN and OUT endpoints, the following rule applies:

- The BUFSIZ field in a link TRB must be set to 0.

11.1.2.3.2.3.4 Transfer setup recommendations

Software can either set up transfers before the host attempts to move data on an endpoint ("preset" transfers) or can set up transfers on demand ("on-demand" transfers). When using preset transfers, software can safely disable the XferNotReady event in the endpoint configuration. However, when using on-demand transfers, the XferNotReady event must be enabled and software may not use the "No Response" variant of the Update Transfer command. The XferNotReady event is issued when the host attempts to move data on an endpoint when one of the following conditions is present:

- No previous transfer was started with Start Transfer.
- Not enough hardware-owned (HWO=1) TRBs are available to handle the requested data movement. The XferNotReady event must not be disabled for control endpoints because the event is an integral part ofcontrol transfer handling.

Although there are many valid ways to set up transfers, it is recommended that you choose one of three general mechanisms:

- When software wants to set up one transfer at a time and has the entire buffer available for transfer, it must set up TRBs that point to the data buffers and in the last TRB it must set the LST bit and issue Start Transfer, which points to the first TRB location.
  - When the USB transfer completes, the core will notify the software through the XferComplete event. The LST bit will be set in the status field of the event. The XferComplete will also release the Transfer Resource.
  - A premature XferComplete event can happen before all the data buffers are exhausted, if there is a Bus Error or, in an OUT transfer a short packet has been received and CSP=0. During these conditions the TRB will have an updated BUFSIZ field which represents the amount of buffer remaining after the successful part of data transfer.
  - Software can also set up an IOC bit TRB, so that it gets notified when data buffers are used up and can free them up sooner than waiting for the entire transfer to complete. On completing a TRB with IOC set, and not the last one, the core will issue an XferInProgress event with IOC set in the status field of the event.
- When software wants to set up one transfer at a time, but it has fewer data buffers available than the full transfer size, it must set up circular TRBs (using a Link TRB), and also set up IOC bits in the TRBs. Depending on buffer allocation and interrupt frequency it can set IOC for once in "x" number of TRBs.
  - As soon as the USB transfer for a TRB is completed, and if IOC is set, the core will generate an XferInProgress event with IOC set in the status field.
  - On seeing the event, software reuses the TRB and updates it with the next data buffer. It also issues an Update Transfer command, indicating it has updated a TRB.
  - If software is slow and hardware finds a TRB with HWO reset to 0, it waits for an Update Transfer command. Upon seeing the Update Transfer, it prefetches the TRB and continues the transfer.
  - When software reaches the end of the transfer, it sets the TRB LST bit. When hardware completes the TRB, it issues an XferComplete event and releases the Transfer Resource.
- When the Device software has multiple transfers to set up, it must set up circular TRBs and also set up CHN bits in all TRBs, except the last of each transfer. For

OUT endpoints, multiple transfers can be supported when the CSP field is set to '0' if each transfer has a multiple of MaxPacketSize bytes, since a short packet will end the transfer. If multiple transfers may contain short packets, the CSP field must be set to '1' to enable the next transfer to continue even if a short packet is received. Depending on buffer allocation and interrupt frequency, it can either set the IOC for once in "x" number of TRBs or in the last TRB of each transfer.

- For OUT endpoints, the transfer can finish prematurely due to a short packet from the host. In this case, the core processes the remaining TRBs, skipping the updates to these TRBs until it reaches TRB of the next transfer. While skipping these TRBs, if any of the TRB it encounters the interrupt setting of ISP or IOC, it will generate the corresponding event once and ignore the interrupt settings of the remaining TRBs until a TRB of next transfer is reached.
- Device software can reclaim these skipped TRBs even though the HWO still indicates 1 (hardware-owned).
- For IN endpoints, software cannot stop providing transfers while it is ending transfers with CHN=0 and LST=0, otherwise it is possible that the endpoint is left in a flow-controlled state on the USB.

11.1.2.3.2.4 Transfer Resource Usage and Transfer State

Software allocates one Transfer Resource for an endpoint during initialization. When software issues a Start Transfer command, this Transfer Resource is used. When an XferComplete event happens, the Transfer Resource is released back to software. Similarly, when End Transfer completes, the Transfer Resource is released.

Following is a typical transfer usage summary (not complete usage model):

- Software sets up data buffer(s) and TRB(s) in external memory with the TRB(s) pointing to the buffer(s).
- Software issues Start Transfer with the pointer to the first TRB in the command parameters.
- If software sets up all the buffers and TRBs at the same time (Host and Device Software negotiate the transfer size), then just one Start Transfer is enough, and software waits for a XferComplete event.
- If software did not set up all the TRBs, then every time software adds a new TRB (by setting HWO=1) it must issue an Update Transfer command. The hardware will fetch the TRB again and continue the transfer.
- The End Transfer command is used only during error conditions and not used during normal transfers. For example, if software has set up multiple transfers and if a USB Reset event happens, it must remove all the transfers from the queue using End Transfer with the ForceRM bit set 1.

- A transfer is completed when all the data has been transferred or a short packet has been received. If software is queuing in only single transfers at a time (normal method for mass storage - for ethernet over USB, software can set up multiple transfers using the Chain bit and Continue on Short Packet bit), once XferComplete interrupt has happened, the transfer is completed. When starting a new transfer, software now needs to issue the Start Transfer command.
- An OUT transfer's transfer size (Total TRB buffer allocation) must be a multiple of MaxPacketSize even if software is expecting a fixed non-multiple of MaxPacketSize transfer from the Host.

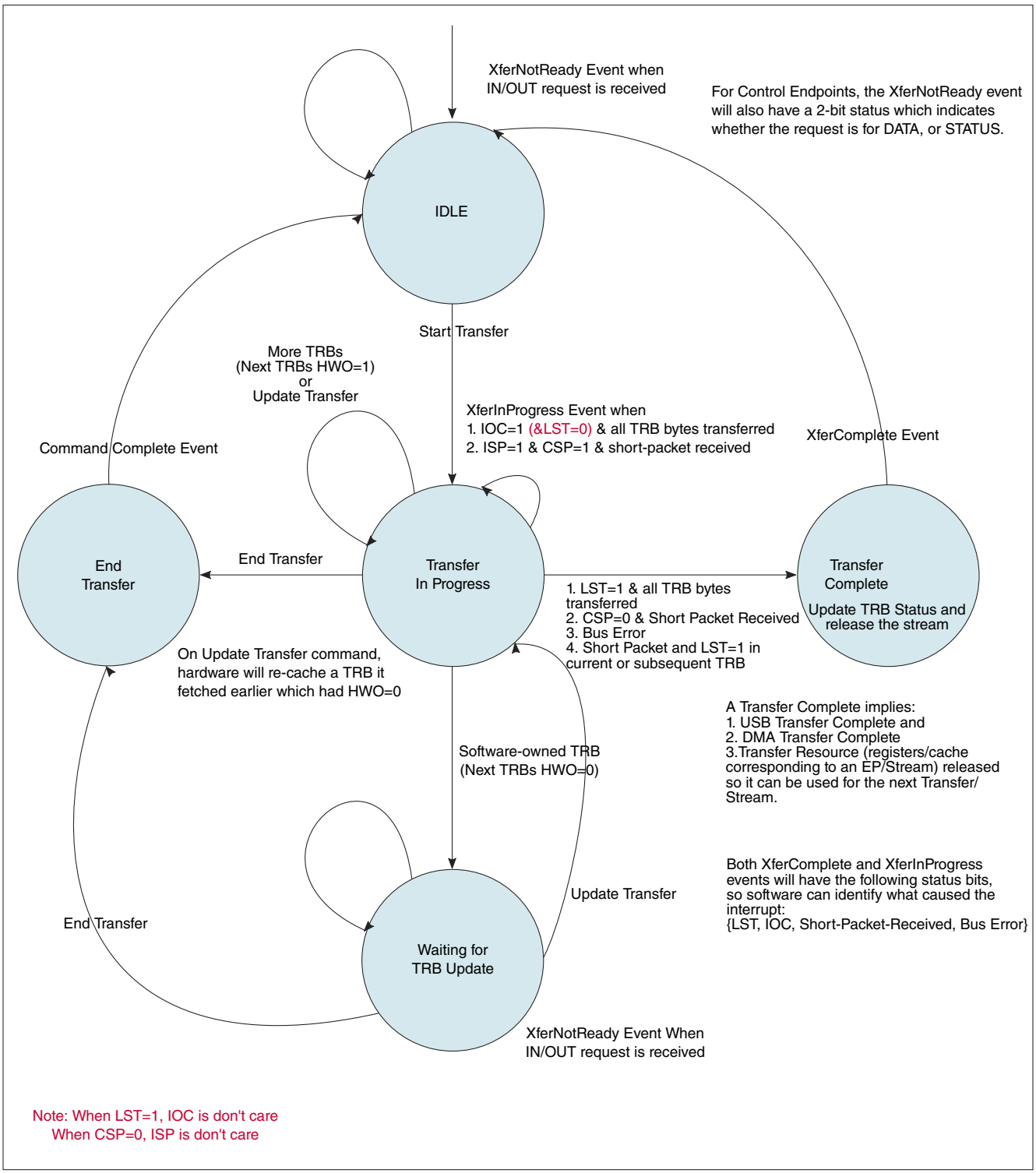


Figure 11-16. Transfer Usage State Machine.

11.1.2.3.2.5 Transfer Descriptions

The following sections describe typical transfer processes.

11.1.2.3.2.5.1 Non-Isochronous OUT Transfers

The sequence below describes a regular, non-isochronous OUT transfer (control-data/ bulk/interrupt).

Table 11-13. Non-Isochronous OUT Transfer Sequence

Steps	Host	Device Core	Device Software
Software sets up data buffers before a host request			
Step 1	-	-	Software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB and caches it.	-
Step 3	Host sends data packets	-	-
Step 4	-	Core receives the packets in RXFIFO and stores them in to TRB data buffers through DMAs. While the RXFIFO has space and the BUS DMA keeps up with the host data transfers, the device receives all the packets. Once all the data is received, the core updates the TRB status and then generates a XferComplete event. In SS mode, if GRXTHRCFG.USBRxPktCntSel is set to, • 0: the NumP sent depends on the DCFG.NumP and bMaxBurstSize • 1: the NumP sent depends on the RXFIFO space available when the packet is received When DMA is slow or CRC fails: • If the RXFIFO overruns or CRC error is detected, device times out in Non-SS mode or asks for the same data packet (through ACK/SeqNumber/Retry Bit) in SS mode. • In HS mode, on NAK the host will issue PING and device will ACK it (if there is enough space in the RXFIFO) and the host resumes with data again. In SS mode, the host will send the requested data. If the device receives any packet with other than the requested data sequence, it will ignore it.	-
Step 5	-	-	Software processes the XferComplete event.
Software sets up data buffers after the host request.			
Step 6	Host sends data packets.	-	-
Step 7	-	Since DMA is not set up, the device will send NAK in Non-SS mode and will send NRDY in SS mode to the host. It also generates an XferNotReady event.	-
Step 8	-	-	On seeing the XferNotReady event, Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-

Table continues on the next page...

Table 11-13. Non-Isochronous OUT Transfer Sequence (continued)

Steps	Host	Device Core	Device Software
Step 10	-	If the core encounters a short packet before the transfer size data programmed in the TRB is received, it skips the remaining TRBs of the transfer. If the core encounters control bits LST, IOC, or ISP it will generate events and, in the case of an LST TRB, the stream will be completed. Software has to reclaim any TRBs skipped with HWO=1.	On detecting a skipped TRB condition, software reclaims the TRBs with HWO=1.

11.1.2.3.2.5.2 Isochronous OUT Transfers

This section describes the difference between isochronous and non-isochronous OUT transfers referring to the sequence in Non-Isochronous OUT Transfers

- In Step 2, if the TRB is not fetched in time, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 4, if DMA is enabled and if RXFIFO overrun happens or CRC fails, the core discards the packet. In addition subsequent packets in the service interval also will be dropped.
- In Step 7, if DMA is not enabled, the core drops the data and generates XferNotReady event.

In general, the TRB chain of a service interval will be released if the core receives DATA0 PID in HS USB 2.0 mode or receives a packet with LPF set in SuperSpeed mode.

Special scenarios are:

- If the core misses this last packet indication, it releases TRB at the service interval boundary.
- If the core encounters a short packet before the transfer size data programmed in TRB is received, it skips the remaining TRBs of the transfer.
- If the core encounters control bits 'LST' or 'IOC' or "ISP" it will generate event and, in the case of LST TRB, the stream will be completed, but software has to reclaim TRBs if any TRBs skipped with HWO=1.

11.1.2.3.2.5.3 Non-isochronous IN transfers

The sequence below describes a regular, non-isochronous IN data transfer (control-data/ bulk/interrupt).

Table 11-14. Non-isochrnous IN transfers

Steps	Host	Device Core	Device Software

Table continues on the next page...

Table 11-14. Non-isochrnous IN transfers (continued)

	Software sets up data buffers before a host request		
Step 1	-	-	Software sets up TRBs and enables DMA by issuing Start Transfer.
Step 2	-	Device fetches the TRB, caches it, and prefetches the data into TXFIFO. Only in SS mode, the core sends ERDY (Async ERDY, even before host requesting data provided endpoint is in flow controlled state. LSP flow is same regardless of if the host asked for it.)	-
Step 3	Non-SS host issues a token request, SS host issues ACK TP to request data.	-	-
Step 4	-	If DMA fetches keep up with host requests, the device sends all the data, updates the TRB status, and then generates an XferComplete event when ACKs for all the packets are received. When DMA is slow: • If the TXFIFO becomes empty on a host data request, the device sends a NAK in Non-SS mode. In SS mode, the device sends NRDY, and when data is available in TXFIFO it sends ERDY and waits for the host request again. • If the TXFIFO becomes empty in the middle of a packet transfer (happens only in Threshold mode), in Non-SS mode the device will invert the CRC. In SS mode, it terminates with EDB after appending the corrupted CRC. The core waits for the host to retry the packet before fetching the data again. If the host retries a packet: • In Non-SS mode, a transmitted packet is kept in the TXFIFO until the ACK arrives (this is immediate in Non-SS mode). If the host retries a packet, it is immediately sent out from the TXFIFO. • In SS mode, before ACK can be received, the core could have sent additional packets in a burst. When a retry happens, NRDY is sent to the host and the TXFIFO is flushed. A DMA command is issued on the master bus to prefetch data from where the host is asking. Once data is fetched into the TXFIFO, the device sends ERDY and waits for host request again. • In Threshold mode, when the TX threshold amount of data is fetched, the device sends ERDY to the host.	-
Step 5	-	-	Software processes XferComplete event.
	Software sets up data buffers after a host request.		

Table continues on the next page...

Table 11-14. Non-isochrnous IN transfers (continued)

Step 6	Non-SS host issues a token request. SS host issues ACK TP to request data.	-	-
Step 7	-	Since DMA is not set up, the device sends s NAK in Non- SS mode, or NRDY in SS mode, to the host. The core also generates an XferNotReady event	-
Step 8	-	-	On seeing the XferNotReady event, software sets up Normal TRBs and enables DMA by issuing Start Transfer.
Step 9	-	Steps 2-5 are followed.	-
Step 10	-	When software knows it is talking to a host that always ends transfers with short packets, and therefore always requires a zero-byte packet to end a transfer, it includes a zero-byte TRB (CHN=0) after the last TRB which is an even multiple of MaxPacketSize. When software does not know if the host is going to do another IN token after receiving the exact number of bytes it expected (and it was a multiple of MPS), it does not include a zero-byte TRB, and sets LST in the last TRB. If the host returns with another IN for a zero-byte packet, it uses the XferNotReady/Start Transfer mechanism to add one zero-byte TRB. This is called "On- Demand."	-

11.1.2.3.2.5.4 Isochronous IN Transfers

This section describes the difference between isochronous and non-isochronous IN transfers referring to the sequence in Non-isochronous IN transfers.

- In Step 4, if the TXFIFO is empty when host IN request arrives, the core returns a zero length packet in both Non-SS and SS modes.
- In Step 7, the device returns zero length packet and generates an XferNotReady event.

Special scenarios:

- When data is fetched and there is no request from the host (possibly due to a missing request or corrupted request) the data will be flushed at the end of the service interval (corresponding Micro- Frame) boundary and the core moves onto fetching next service interval data. If there are some TRBs that are not serviced in the service interval, the core will skip these TRBs.
- If the core encounters control bits LST or IOC, it will generate an event, and in the case of LST TRB, the stream will be completed, but has to reclaim TRBs if any TRBs skipped with HWO=1.

11.1.2.3.2.6 Handling ENDPOINT_HALT

On receiving a SetFeature (ENDPOINT_HALT) control transfer, software issues a Set Stall command on the endpoint. Because of the delay from when software issues the command and when it is recognized by the core, other transfer events (XferInProgress, XferComplete) may be received prior to the endpoint being halted. For a description of the Set Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

On ClearFeature (ENDPOINT_HALT), software must first remove all pending transfers for the endpoint through the End Transfer command. It may then issue a Clear Stall command on the endpoint followed by Start Transfer to start transfers again. For a description of the Clear Stall endpoint command and its effects, see Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL).

11.1.2.3.2.7 Handling L1 Event During a Transfer

This section discusses the scenario in which the software already issued the Start Transfer command but the LPM occurred before or during the data transfer.

When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

NOTE

- The GDBGFIFOSPACE register is a debug register which gives the 'Space Available" for the endpoint selected with a write prior to the read. If this space available is less than the particular TXFIFO's maximum size, then the TXFIFO is not empty.
- Software can enable the Link State change event for USB 2.0 as there are not a lot of events compared to SS.

11.1.2.3.3 Isochronous Transfer Programming Model

Isochronous endpoints get guaranteed service by the host during a "Service Interval". The service interval for an endpoint is communicated to the host via the endpoint descriptor, and it is configured in the hardware through the DEPCFG command. However, unexpected behavior on the bus may prevent the host from servicing the endpoint as frequently as expected.

Although the endpoint receives "guaranteed" service during the interval, there are no bus-level acknowledgments, which means that there is no guarantee that the host (IN endpoints) or the device (OUT endpoints) receives correct data.

Therefore, the isochronous programming model differs from the bulk programming model to accommodate these two factors:

- No bus-level acknowledgement of packet transmission or reception
- No guarantee that all the data setup for an interval has been transmitted or received

For isochronous endpoints, software has the following responsibilities:

- Set up enough TRBs to keep up with the rate of data transmission or reception
- For FIFO-based IN endpoints, guarantee the validity of the TX data at least 1 micro Frame before the beginning of the interval that the data will be transmitted

Hardware has the following responsibilities:

- Transmit or receive data at the appropriate bus time
- For FIFO-based IN endpoints, delay fetching until 1 micro Frame before the beginning of the interval that the data will be transmitted
- Maintain a 1-to-1 correspondence between Buffer Descriptors and the intervals on the bus
- Report missed isoc intervals and update the buffer sizes in TRBs to reflect the amount of data moved

11.1.2.3.3.1 Definitions

- bInterval: The field in an endpoint descriptor used to communicate the service interval of the device. Its value ranges from 1 to 16, however, the USB 3.0 core supports a range of 1 to 14.
- Service interval: An integral number of microframes (2^[bInterval-1]) during which the host will poll the device to move data.
- Beginning of interval: The interval begins when the least significant (bInterval-1) bits of the bus time are all 0's.
- End of interval: The interval ends when the least significant (bInterval-1) bits of the bus time are all 1's.
- Data payload: The number of bytes to be moved during the service interval. It will be transmitted using MaxPacketSize packets.
- Buffer descriptor: A set of one or more TRBs with CHN=1 and the last one having CHN=0. The group of TRBs represents the data buffers for one service interval.
- Last TRB of a Buffer Descriptor: The TRB within a Buffer Descriptor that has CHN=0. When an interval has completed, this TRB will contain the total remaining buffer size of the Buffer Descriptor andus also the completion status of the interval.

- Microframe (μF): A unit of time on the USB that lasts 125 μs.
- Start of Frame (SOF): The beginning of a microframe.
- FIFO-Based Isoc IN: A sub-type of Isoc IN endpoints that has the characteristic of always assuming the HWO bit in a TRB is '1' and delays fetching of transmit data.
- Prefetch delta: For FIFO-based IN endpoints, TX data may be prefetched as early as 1 μF before the beginning of a Service Interval, but never earlier.
- Retire a TRB: When HW sets the HWO bit to 0 in a TRB and writes it back.
- Retire a Buffer Descriptor: When HW retires at least the first and last TRB of a Buffer Descriptor.
  - Other TRBs of the Buffer Descriptor may not be retired if an interval was missed (IN or OUT) or if a short packet/last packet is received (OUT). In this case, software may reclaim those TRBs even though HWO=1.
- Missed interval: When the device does not move all the data in an interval. This may occur when the host does not poll for all the data, or because of application-side delays that prevent all the data from being moved.
  - For IN endpoints, this occurs when the host does not request all the data prepared in the Buffer Descriptor, or not all the data is fetched from the system bus in time.
  - For OUT endpoints, this occurs when the host does not send a packet, a packet is dropped due to CRC error, or the system bus is too busy to accept the data.

11.1.2.3.3.2 Endpoint configuration

An isochronous endpoint is setup in much the same way as a bulk endpoint, with the following exceptions:

- The bInterval_m1 value in DEPCFG Parameter 1 must be set to the value reported in the endpoint descriptor minus 1. When the core is operating in Full Speed, bInterval_m1 must be set to 0.
- The MaxPacketSize value in DEPCFG Parameter 0 must be set to the same value reported in the endpoint descriptor.
- The "FIFO-based" bit in DEPCFG Parameter 1, bit 31, must be set for FIFO-based isochronous endpoints.
- For the best performance, set the TXFIFOs corresponding to Isochronous IN endpoints to a high priority.

After the endpoint is configured, it can be enabled by setting the appropriate bit in the DALEPENA register.

11.1.2.3.3.3 Transfer configuration

Software describes isochronous transfers through a series of Buffer Descriptors. Each Buffer Descriptor corresponds directly to one service interval of data. The fields within an isochronous TRB are the same as bulk TRBs with the following exceptions:

- The SOF field in an IN endpoint TRB is a don't care. For an OUT endpoint, the core will write this field with the timestamp of the last packet received into the TRB's buffer. This information is not needed for normal operation, only for debug purposes.
- For High-Speed, High-Bandwidth IN endpoints, a maximum of 3 packets can be sent during an interval. The PktCntM1 field ([25:24] of the 3rd DWORD) must be set to the (number of packets in the Buffer Descriptor - 1). For example, if 3 packets are to be transmitted during the interval, this field must be set to 2. For Super-Speed and OUT endpoints, this field is a don't care.
- The first TRB in a Buffer Descriptor must have the TRBCTL field set to the "Isochronous-First" type while all others have this field set to "Isochronous".
- The ISP bit is renamed IMI (Interrupt on Missed Interval) and should be set if software wants to receive an XferInProgress event when at least 1 packet is missed within an interval.
- The IMI bit should be set to the same value in all TRBs of a Buffer Descriptor.
- The CSP bit must be set to 1 (short packets cause the hardware to move to the next Buffer Descriptor, they do not end an isochronous transfer).
- The LST bit should be set to 0 (isochronous transfers normally continue until the endpoint is removed entirely, at which time an End Transfer command is used to stop the transfer).

All other bits and fields (IOC, HWO, CHN, BPTR, BUFSIZ) retain the same behavior as they have for bulk endpoints. If software needs to receive an interrupt after every service interval, it should set the IOC bit to '1' in each TRB. However, if software wants to reduce the interrupt frequency and the application can tolerate some latency, the IOC bit can be set to '0' and the core will not generate a XferInProgress event when the TRB is completed.

For OUT endpoints, the Buffer Descriptor must describe a total buffer size of:

```
(Mult+1)*MaxBurst*MaxPacketSize
```

11.1.2.3.3.4 Starting a Transfer

The hardware will report the bus time that the host starts polling the endpoint inside the XferNotReady event. Software will use this value as a reference when issuing the Start Transfer command to serve as time synchronization with the host.

1. Set up TRBs and data buffer for the endpoint.

2. After configuring and enabling the endpoint, wait for an XferNotReady event.

   The XferNotReady event will contain the time that the host started polling the endpoint in the upper 16 bits.

3. Issue the Start Transfer command to the endpoint with a future microframe time written into the upper 16 bits of the DEPCMD register. The future microframe time must be a value that is an integral multiple of intervals after the time reported in the XferNotReady event and aligned to the beginning of an interval. For example, if bInterval is 3 (4 microframes), and the XferNotReady time is 2, the value can be 4, 8, 12, and so on. The future microframe time must also be no greater than 4 seconds past the time reported in the XferNotReady event.
4. If the future microframe time has already passed when the command is received, the core will respond with an error (bit 13 in the Command Complete event).

In this case, software must issue End Transfer, then wait for another XferNotReady event and attempt the command again with a time that is further in the future.Otherwise, the first Buffer Descriptor will be used for the interval starting with the microframe time specified in the Start Transfer command.

11.1.2.3.3.5 Core Behavior During an Interval

After a transfer has been started, the hardware will perform the following functions for IN endpoints:

1. Fetch TX data as early as one interval prior to the beginning of the interval (call it A) if the HWO bit is set to one in the TRB.
2. Decrement the buffer size of each TRB as packets are transmitted.
3. Retire TRBs when their buffer size has reached 0, issuing an XferInProgress event if the IOC bit is set.
4. If the next interval (B) starts before all the packets have been transmitted for interval A:
   a. Flush the TXFIFO.
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   c. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.
   d. See Checking interval status for a description of how software can determine that an interval ended unexpectedly.
   e. Go to step 1 to prepare for interval C
5. Otherwise, if all the TRBs of interval A are completed, the hardware will prepare for interval B.
6. If the host completes an interval by polling for all the data, but then it polls the endpoint again during the same interval, the hardware will respond with a zero-length packet and no interrupt will be made to software.

7. If the host polls the endpoint prior to the time specified in the Start Transfer command, the hardware will respond with a zero-length packet and no interrupt will be made to software.
8. If the host polls the endpoint during the expected interval and the hardware has no data prepared, the core will respond with a zero-length packet, but no XferNotReady event will be generated because the transfer is active. When the next interval starts, the XferInProgress event is generated (based on ISP and IMI) with the "Missed Isochronous" status, which is the way software finds out that this occurred.

For OUT endpoints, the hardware performs the following functions:

1. If a packet is received for the correct interval represented by the Buffer Descriptor (call it A).
   a. Write the packet into the buffer.
   b. Decrement the buffer size of the TRB.
   c. Write the timestamp of the received packet into the TRB.
   d. Retire a TRB when its buffer size reaches 0, issuing an XferInProgress event if the IOC bit is set.
2. If a short packet or packet with the last packet flag (lpf) is received for the correct interval.
   a. Write the packet into the buffer.
   b. Retire the Buffer Descriptor of interval A with the TRBSTS set to 0.
   c. If the IOC bit is set in the last TRB of the Buffer Descriptor, issue an XferInProgress event with bit [13] set.
   d. Go to step 1 to prepare for interval B.
3. If a packet is received for the correct interval but it has an error (CRC error, RXFIFO overflow), the core will not increment its expected sequence number value which causes future packets within the same interval to be dropped.
4. If a packet is received at a bus time prior to the time specified in the Start Transfer command, it will be dropped.
5. If a packet is received for the correct interval (A), but not enough buffer space is available for the core to write the packet (due to the HWO bit still '0' in one or more of the TRBs of the Buffer Descriptor), no XferNotReady event will be generated. The core will behave as follows:
   a. Wait until software sets the HWO bit to '1' and issues an Update Transfer command
   b. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status
   c. Go to step 1 to prepare for interval B.
6. If a packet is received for the next interval (B) before all the packets have been received for interval A:
   a. Retire the Buffer Descriptor of interval A with a "Missed Isochronous" status.
   b. Retire the Buffer Descriptor of interval B with a "Missed Isochronous" status.

c. See section Checking interval status for a description of how software can determine that an interval ended unexpectedly.
d. Go to step 1 to prepare for interval C.

Special Considerations for Isochronous OUT Endpoints

For OUT isochronous endpoints, the hardware detects a missed interval when the host sends a data packet in a future interval. It does not detect the missed interval at the exact interval boundary. Therefore, if the host abruptly stops sending isochronous OUT packets, there will be no interrupt or event indicating this to software. Software should use one of the following mechanisms to detect the interruption of isochronous OUT traffic:

1. The ultimate consumer of the isochronous OUT data will detect an underflow.
2. The device driver can use a timer detect that no XferInProgress events have been received for multiple intervals.

If this occurs, software should issue an End Transfer command to the endpoint and wait for a XferNotReady event which signals that the host is ready to resume isochronous traffic.

11.1.2.3.3.6 Checking interval status

As packets are transmitted or received during an interval, the buffer size of each TRB will be decremented. When the host is operating normally and polling for all the interval data, the buffer size of all the TRBs of an IN endpoint Buffer Descriptor will be zero after the interval has completed. For OUT endpoints, if the host sends less data than the Buffer Descriptor was setup for, the remaining buffer size may be greater than 0.

When the interval completes, the final remaining buffer size and missed isoc status is written to the last TRB of the Buffer Descriptor. If MissedIsoc is set, then it means the BUFSIZ is not accurate and may indicate that more data was transmitted or received than in reality. If the MissedIsoc is not set, it means the BUFSIZ field is correct.

When hardware detects the end of an interval (including normal and abnormal ends), it performs the following:

- If it has not already been retired, the first TRB of the Buffer Descriptor will be retired.
- Any non-first TRBs with CHN=1 that had not already been retired will not be written back. HWO will still be 1, and software can reclaim them for another transfer.
- The last TRB of the Buffer Descriptor will be retired with:
  - HWO = 0.

    - BUFSIZ = The total remaining buffer size of the Buffer Descriptor.
    - TRBSTS = "Missed Isoc" if any packets were missed, zero otherwise.
- If the IOC bit is set in the last TRB, or the IMI bit is set and packets were missed, hardware will issue an XferInProgress event

Software can get notification of an interval completing by setting the IOC bit in the last TRB of the Buffer Descriptor. The IOC bit may also be set in any other TRB of the Buffer Descriptor if software wants earlier notification that a TRB has completed. Software can also get only a notification of an interval completing unexpectedly by setting the IMI (Interrupt on Missed Isoc) bit in the last TRB of the Buffer Descriptor. When an interval completes unexpectedly and either the IOC or IMI bit is set in the TRB, the MissedIsoc bit (15) of the XferInProgress event will be set. The following table shows which events are generated in each scenario:

Table 11-15. TRB event generation events

Scenario	IOC	IMI	Action
TRB completed with a MaxPacketSize packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (IOC=1)
	1	1	XrefInProgress (IOC=1)
TRB completed with a short packet	0	0	No interrupt
	0	1	No interrupt
	1	0	XferInProgress (Short=1, IOC=1)
	1	1	XrefInProgress (Short=1, IOC=1)
Interval completed due to missed packet (missed isoc)	0	0	No interrupt
	0	1	XferInProgress (MissedIsoc=1, IOC=1)
	1	0	XferInProgress (MissedIsoc=1, IOC=1)
	1	1	XrefInProgress (MissedIsoc=1, IOC=1)

It is normal to lose two intervals at a time when an error occurs during one interval. One interval is lost because of the host (which may not be polling enough) and the second interval is dropped because the device needs time to synchronize up to the next (third) interval.

11.1.2.3.3.7 Adding intervals to a transfer

Because isochronous endpoints represent a stream of data, the TRBs of an isochronous endpoint will normally be setup in a circular list, such as:

- TRB 1 (CHN=0, IOC=1)

- TRB 2 (CHN=0, IOC=1)
- Link (to TRB 1)

Example: Assume TRB 1 represents the data for the first interval and TRB 2 represents the data for the second interval. Because IOC=1, when the core completes the first interval, it will issue an XferInProgress event which indicates to software that TRB 1 can be analyzed to retrieve the results from interval 1. TRB 1 can the be re-used to represent the data for the third interval by setting HWO=1 and issuing an Update Transfer command.

Each time software sets up TRBs for a new interval, it must follow the guidelines in the Transfer Configuration section. Software will set up TRBs for future intervals when it obtains those buffers from another software layer or when TRBs are retired by the core.

11.1.2.3.3.8 Moderating Events

During a transfer, software may enable or disable any endpoint-specific event by re-issuing the DEPCFG command with the "Config Action" set to "Modify" and a modified DEPEVTEN field. All other field. All other fields in all other parameters must remain the same as the initial endpoint configuration.

11.1.2.3.3.9 Other Types of Isochronous Endpoints

The above description of the isochronous endpoint programming model assumes that software is setting up buffers in external memory that apply to certain intervals of data. This is called a "buffer-based" isochronous endpoint.

However, there is another mode that the hardware supports to accommodate other mechanisms of sourcing or sinking isochronous data, called "FIFO-based" and the types are defined by bits [31:30] of DEPCFG Parameter 1. This table illustrates the differences between the models:

Table 11-16. Isochronous Endpoints Models

Type	FIFO-Based [31]	Bulk-based [30]	TRB Fetch	TRB Write back	TX DMA	Retire Buffer Descriptor at passed interval
Buffer-based	0	0	When HWO=1	Yes	When HWO=1, no earlier than one interval ahead of time.	Yes
FIFO-based	1	0	When internal cache space available	No	No earlier than 1 µF before the interval	Yes
Invalid	1	1				

11.1.2.3.3.9.1 FIFO-based isochronous IN Endpoints

The software of some applications is unable to keep up with the short latency and high bandwidth of isochronous traffic and require the data to be sourced and done a sync through external FIFOs. One example is that there are two external TXFIFOs, one for each interval, and an address-to-FIFO-pop logic is implemented so that when the core attempts to read from interval 1's address, the translation logic converts this into a pop signal for interval 1's TXFIFO. When the core attempts to read from interval 2's address, the translation logic converts this into a pop signal for interval 2's TXFIFO.

To describe a FIFO-based implementation, software sets up an endpoint with the "FIFO-based" configuration bit set. This type of endpoint ignores the HWO bit in all TRBs, assuming that the TRB is always valid, and that the core should never write it back. Software chooses the buffer pointers within the TRBs to correspond to the address needed by the translation logic to specify which FIFO should be popped. By also using the CHN bit, headers and payload can be concatenated from different FIFOs. For example, if there are 4 external FIFOs: HA, PA, HB, PB, where HA/HB contain the 4 byte header for 2 intervals and PA/PB contain the 512 byte payload for 2 intervals, this can be described by using the following 5 TRBs:

1. BUFPTR=HA, IOC=0, CHN=1, BUFSIZ=4
2. BUFPTR=PA, IOC=1, CHN=0, BUFSIZ=512
3. BUFPTR=HB, IOC=0, CHN=1, BUFSIZ=4
4. BUFPTR=PB, IOC=1, CHN=0, BUFSIZ=512
5. Link to (1)

Every interval, the core will be creating a 516 byte packet that consists of 4 bytes from the header FIFO and 512 bytes from the payload FIFO. However, if the host does not poll for the packet, the core will skip 1 or 2 intervals of popping, depending on whether it has already started reading the next interval. External logic (or software) is responsible for flushing and refilling alternate FIFOs so that the core is always reading the correct data for the next interval. In normal intervals, the FIFOs will be empty, but when an interval is missed, there may still be data present in the FIFOs.

Software Requirements:

- The "FIFO-based" bit must be set in the DEPCFG when configuring the endpoint.
- No field within any TRB may be changed after the Start Transfer command is issued.
- Data must be valid in the external FIFO at least 1 uF before the beginning of the interval for which the data is intended.

Core Behavior:

- The earliest the core will read from the FIFO will be 1 µF before the beginning of the interval for which the data is intended.
- The core will not write back the TRB after it has completed it.
- The core will re-read the TRB from external memory even though it is not allowed to change.
- The only indication of missed intervals is the XferInProgress event if IOC or IMI is set in the TRBs. No indication will be made of how much data was actually transmitted.

11.1.2.3.3.9.2 FIFO-based isochronous OUT Endpoints

The FIFO-based model is also supported for OUT endpoints. In this case, core writes to specific addresses will be translated into external FIFO pushes. The same example from above (4 byte headers and 512 byte payloads) can be applied to OUT endpoints where the headers are split from the payload as the core receives 516 byte packets. The same software requirements and core behavior apply (for example, no writebacks and no indication of how much data was received). External logic assumes that if less than the expected amount of data is pushed into an interval's FIFO, the interval is invalid.

11.1.2.3.3.10 End a Transfer

Since isochronous endpoints have no bus-level acknowledgements, it is not necessary to set LST=1 in a TRB to gracefully end an isochronous transfer. Software can issue an End Transfer command to end an isochronous transfer. The core will wait until it can complete operations for the endpoint before returning the Command Complete event in response.

11.1.2.3.4 Control transfer programming model

Control transfers follow a flow of setup/data/status. On the USB bus, there are many error scenarios that can disrupt this flow. The hardware has a special mechanism that automatically recovers from these scenarios as long as software follows this single control transfer programming model.

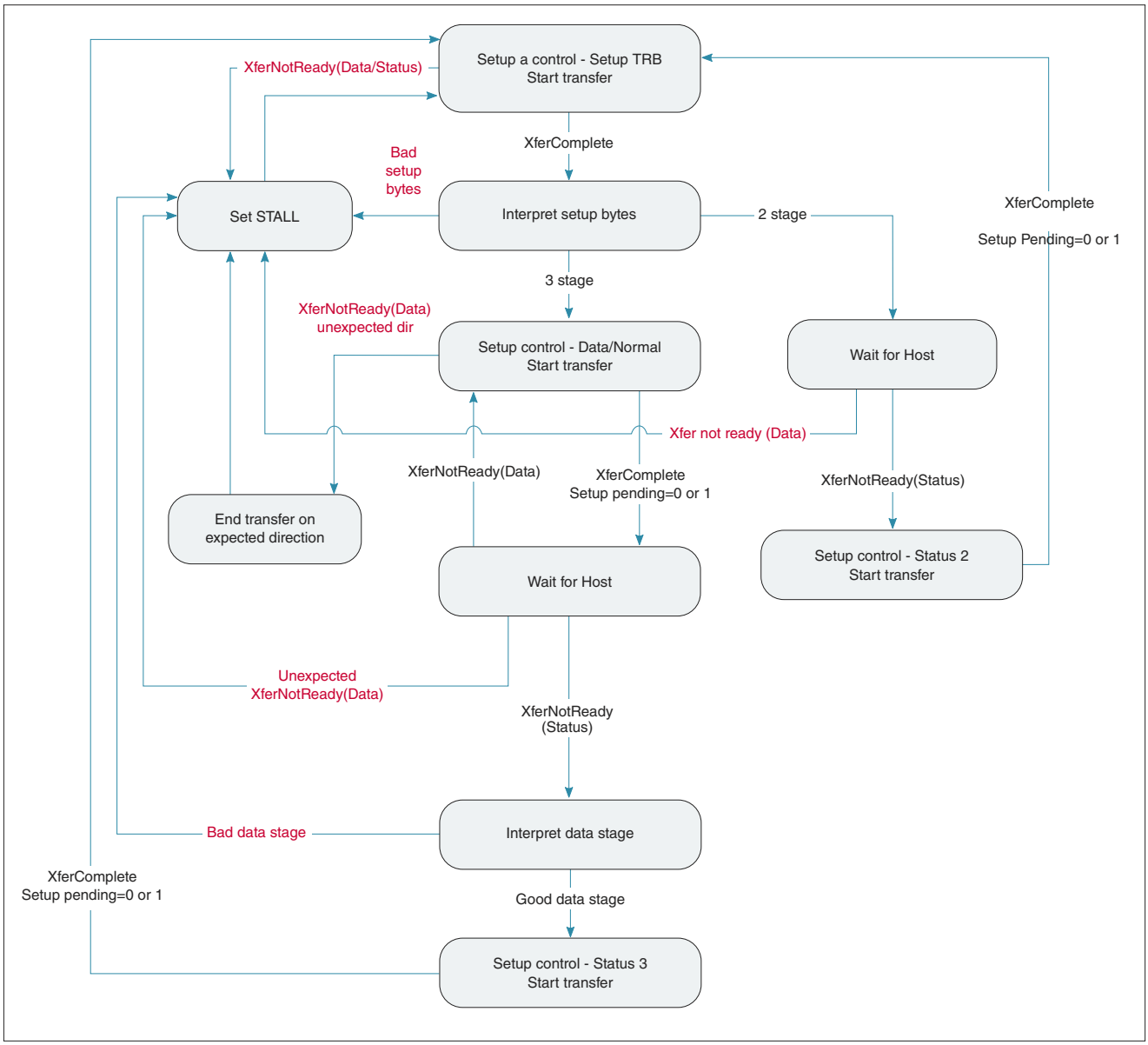


Figure 11-17. Software flow for control transfers

If hardware does not receive a setup packet, it discards the unexpected data and status stages. It does not issue XferNotReady to the application. The hardware will also not generate a XferNotReady event if it receives a setup packet before a TRB is prepared for it. As control endpoints are bi-directional, the IN and OUT directions of the endpoint are used for the different stages as follows:

Control read:

- EP0 - Setup OUT
- EP1 - Data IN
- EP0 - Status OUT/Status TP

Control write or 2-stage transfer:

- EP0 - Setup OUT
- EP0 - Data OUT (if present)
- EP1 - Status IN/Status TP

Set STALL is always issued on EP0, and the XferComplete/XferNotReady events are generated on the correct direction. For other control endpoints, substitute EP0 with the OUT direction (such as, 2, 4, 6) and EP1 with the IN direction (such as, 3, 5, 7). The same programming model is used for USB 2.0 and USB 3.0 control transfers.

11.1.2.3.4.1 Two-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Wait for an XferNotReady event for the Status stage which will occur on EP1. If an XferNotReady event for the Data stage is received (either direction), issue Set Stall on EP0 and go back to the step 1. This is an error case where the host is attempting to start the data stage when the setup bytes did not indicate a data stage was present.
4. After XferNotReady (Status) is received, start the status stage by issuing Start Transfer on EP1 pointing to a valid Status-2 TRB.
5. After XferComplete is received on EP1, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not receive the Status stage ACK.

11.1.2.3.4.2 Three-stage control transfer programming model

1. Software sets up a Setup TRB and issues Start Transfer on EP0 pointing to the Setup TRB.
2. After XferComplete is received, software interprets the setup bytes. If they are bad, issue Set Stall on EP0 and go back to the step 1. If a XferNotReady (Data/Status) event is received before the XferComplete event for the Setup stage, issue Set Stall. This is an error case where the host is attempting to move data or start the status stage for a previous control transfer that has already completed.
3. Start the data stage by issuing Start Transfer on EP0 (control write) or EP1 (control read) pointing to a valid Control-Data TRB.

   a. If an XferNotReady (Data) event is received for the incorrect direction, software must issue an End Transfer for the data stage it has already started, then issue Set Stall. This is an error case where the host is attempting to move data in the wrong direction.
   b. If a XferNotReady (Data) event is received for the correct direction, ignore the event and continue to wait for a XferComplete event.
4. After the XferComplete event is received, software waits for an XferNotReady event. Although not required, software may look at the completed Data TRB(s) and examine the Setup Pending bit in the TRB status field and the BUFSIZ. If the Setup Pending bit is set or the BUFSIZ is non-zero, it means that this control transfer was aborted on the USB bus and that the host did not complete the data stage.
5. If a XferNotReady (Data) event is received after the XferComplete for the Data stage, it could mean one of two things:
   a. This host is trying to complete the data stage by moving a 0-length packet. This can occur if the data stage was an exact multiple of max packet size. If this is the case, software sets up an extra TRB (with a BUFSIZ of max packet size for control writes, or 0 for control reads), issues Start Transfer, and goes back to the step 4.
   b. This host is trying to move more data than specified in the wLength field of the setup bytes. In this case, software issues Set Stall on EP0 and goes back to the step 1.
6. When the XferNotReady (Status) event is received, software interprets the data stage (for example, number of bytes transferred, content of the control write buffer) and decides if the data stage was successful. If not, it issues Set Stall on EP0 and goes back to the step 1.
7. Start the status stage by issuing Start Transfer on EP1 (control write) or EP0 (control read), pointing to a valid Status-3 TRB.
8. After XferComplete is received, go back to the step 1.

Although not required, software may look at the Status TRB and examine the Setup Pending bit in the TRB status field. If it is set, it means that this control transfer was aborted on the USB bus and that the host did not actually receive the Status stage ACK. When device software sets STALL with the Set Stall command, it is possible that this command can be delayed on the chip bus due to latency or the chip bus servicing other agents. In this case, software can receive an XferNotReady event if the core detects no TRB active before it detected the Set Stall. Software must service these events knowing that it stalled the control command.

11.1.2.3.4.3 Handling fewer requests than wLength

In a control IN transfer, if the host asks for less data than wLength, the core skips the TRBs of the data stage and generates an XferComplete event after encountering the TRB that has LST=1. Hardware sets HWO to 0 in those TRBs that it skipped and in the last TRB. In a control OUT transfer, if the host sends less data than wLength, the core treats this like a short packet received into a TRB with CSP=0. It immediately generates a XferComplete event, not setting the HWO bit in the remaining TRBs to 0. Software needs to reclaim those TRBs that still have HWO=1.

11.1.2.3.4.4 Control OUT transfer examples

The sequence in the following table describes two and three stage control OUT transfers.

Table 11-17. Control OUT transfer

Steps	Host	Device core	Device software
	Setup stage		
1	-	-	Software sets up the Setup TRB and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction).
2	-	Device fetches the TRB and caches it.	-
3	Host sends SETUP packet.	-	-
4	-	SETUP packet is received in the RXFIFO. There are 24 bytes allocated for each control endpoint, which guarantees reception of the three back-to-back SETUP packets. An ACK handshake is returned in non-SS mode. An ACK TP with NumP=1 is returned in SS mode. After SETUP, bytes are transferred through the DMA, and the core issues an XferComplete event.	-
5	-	If the host starts a Data/Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode. If the SETUP request is invalid, core returns STALL (when software sets STALL) in non-SS mode. In SS mode the core returns ERDY, and when the Data/Status stage is received it returns STALL.	Software processes the XferComplete event and decodes the SETUP bytes. If a Data stage is required, it will set up one Control-Data TRB and possibly subsequent Normal TRBs, then issue Start Transfer to EP0 (endpoint 0, OUT direction). If a Data stage is not needed, it will skip to the Status stage. If the SETUP request is invalid, software instead sets the STALL bit by issuing Set Stall command for EP0 (OUT direction).
	Data stage		
6	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY.	-

Table continues on the next page...

Table 11-17. Control OUT transfer (continued)

Steps	Host	Device core	Device software
7	Host sends data packets.	-	-
8	-	The NumP for control transfer will be 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer that software sets up without waiting for it to happen on the USB bus, and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs.	-
9	-	If the host starts a Status stage, the core returns NAK in Non-SS mode or NRDY TP in SS mode.	On seeing the event: • Software decodes the control write data. If the Data stage is OK, then it will wait for XferNotReady with "Control Status Request" set, set up a Control-Status-3 TRB (0-bytes), and enable DMA by issuing Start Transfer to EP1 (endpoint 0, IN direction). • If the Data stage is not OK, then software sets STALL by issuing Set Stall command to EP0.
	Status stage		
10	-	Device fetches the TRB and caches it. In SS mode only, the core will send ERDY	Software sets up a Status stage TRB only after the Data stage of a three-stage control transfer is completed and an XferNotReady event occurred.
11	Host requests status packets.	-	-
12	-	In Non-SS mode, the device will send a zero-length packet if STALL is not set; otherwise, it will send STALL. In SS mode, if STALL is not set, then it sends ACK TP with NUMP=1 else it sends STALL. If another SETUP is received before the status transfer on USB, the core will send XferComplete with the SetupPndg status bit set. The STALL bit will be cleared by the core whenever it receives a SETUP packet. SETUPs are always accepted.	-
13	-	-	On seeing the event: • If there are no other errors, then the transfer has completed successfully.

11.1.2.3.4.5 Control IN transfer examples

The sequence in the following table describes two and three stage control IN transfers.

Table 11-18. Control IN transfer

Steps	Host	Device Core	Device Software
	Setup stage		
1-5	Steps 1-5 are same as Control OUT transfer examples, except that software sets up Tx buffers and Start Transfer is issued to EP1 (endpoint 0, IN direction).		
	Data stage		
6	-	Device fetches the TRB and caches it. It prefetches the data and puts it in the TXFIFO. The core sends ERDY only in SS mode.	-
7	Host requests data packets	-	-
8	-	This step is similar to the step 4 of Non-isochronous IN transfers. The NumP for control transfer is 1. In addition, if another SETUP is received before the data transfer happens on USB, the core will skip the data transfer software has set up without waiting for it to happen on the USB bus and send XferComplete (when LST=1 TRB encountered) with the SetupPndg status bit set. Software has to reclaim the TRBs with HWO=1 in the skipped TRBs and flush the TXFIFO.	-
9	-	If the host starts the Status stage, the core returns NAK in non-SS mode or NRDY TP in SS mode.	On seeing the event: • If the Data stage is OK, then it sets up Control-Status-3 TRB (zero-bytes) and enables DMA by issuing Start Transfer to EP0 (endpoint 0, OUT direction). • If the Data stage is not OK, then software will set a STALL by issuing Set Stall command to EP0 (OUT direction).
	Status stage		
	Same as in Control OUT transfer examples.		

11.1.2.3.5 Stream handling in SuperSpeed

The SuperSpeed stream protocol consists of a negotiation between the host and device concerning the selection of a stream, followed by data movement on the selected stream. The core automatically handles the stream protocol by initiating stream selection and also taking into account streams initiated by the host.

NOTE

This section is not applicable for USB 2.0 mode.

11.1.2.3.5.1 Stream IDs and transfer resources

A stream ID is a 16-bit value which represents an individual flow of data between host and device. For stream-capable endpoints, the stream ID must be non-zero and is used in:

- The Command Parameter field in the DEPCMD register used when issuing a Start Transfer command.
- The Stream ID field in a TRB.
- The EventParam field in an endpoint-specific event (DEPEVT): XferComplete, XferInProgress, XferNotReady, and StreamEvt.

It is necessary to allocate 1 Transfer Resource per endpoint. The cost of each transfer resource is (32 + N_TRBS_PER_XFER x 16) bytes of memory in the descriptor cache. Core can cache maximum four TRBs per transfer.

After the Start Transfer command completes, the hardware returns the 7-bit Transfer Resource Index in the command complete event (and Device Physical Endpoint-n Command Register (DEPCMD)). This index is used only for the Update Transfer and End Transfer commands. In all other places, the Stream ID is used.

11.1.2.3.5.2 Stream selection and stream programming model

Referring to the USB 3.0 Device Stream Protocol State Machine, the Idle state is the state where a device does not have a current stream to work on. When the hardware detects that it is in the Idle state, it will attempt to initiate and move data on a stream by transmitting ERDY to the host.

In order for this to occur, the endpoint must be "primed" and the stream must be both active and ready.

- **Primed**: After a stream-capable endpoint is configured, it is in the disabled state, which prevents the device from initiating any stream selection. After the host performs a Prime transaction, the endpoint is moved into the "Primed" state where it is eligible to initiate stream selection.
- **Active**: Software makes a stream active by issuing a Start Transfer command for the stream. A stream becomes inactive after an XferComplete event or after software issues an End Transfer command.
- **Ready**: After a stream is added via the Start Transfer command, its default state is "ready." The hardware will label the stream as "not ready" if the host gives a NoStream rejection to the hardware's attempt to initiate the stream. If the host

performs a Prime, all currently active streams are automatically set to the "ready" state by hardware.

In the Idle state, the hardware selects a stream from its cache to initiate. Once the core has chosen an active stream, it attempts to initiate and move data on that stream until the endpoint enters the Idle state, which occurs when the host rejects the stream selection (NoStream) or the stream is terminated (PP=0) by the host or device. Therefore, software is required to prepare enough HWO=1 TRBs for at least one packet prior to issuing the StartXfer command.

In device-oriented stream selection class drivers (such as UASP), the host will not reject the device's stream selection. However, other class drivers may be developed in which the host initiates stream selection. In this situation, software will not start any transfers and will wait for a XferNotReady event. When that event is received, software will look at the StreamID in the event and start the transfer associated with the given Stream ID. If for some reason the class driver allows the host to reject a stream that has already been initiated by the host or device, software will receive a Stream (NotFound) event. If this occurs, software may replace the existing stream by issuing End Transfer for it and then issuing Start Transfer for another stream.

Some host implementation scenarios exist where the host and device become out of sync in the stream protocol, creating a deadlock condition where the device waits for the host to issue a Prime transaction while the host waits for the device to issue an ERDY. To resolve potential deadlock conditions, software should perform the following:

1. Implement a time out for stalled transfers on stream-capable endpoints.
2. When the timeout elapses with no data transfer for any stream after a StreamEvent (NotFound) event, restart one of the streams by issuing an End Transfer command followed by a Start Transfer command.

This places the stream in the Ready state and causes the core to transmit an ERDY to the host.

11.1.2.3.5.3 Data movement within a stream

Data movement within a stream is the same as data movement using the Bulk endpoint type (Start Transfer, Update Transfer, and updating TRBs). The XferInProgress, XferComplete, and XferNotReady events are issued as data is transferred between host and device. The Stream ID field in the events indicate which stream within the endpoint is generating the events. If software receives an XferNotReady event with the EventStatus field indicating XferNotActive, it means that the host is attempting to initiate a stream that has not been added to the hardware's cache through Start Transfer. In response to this event, software should add the requested stream if it is available.

11.1.2.3.6 Low power operation

11.1.2.3.6.1 Low power operation of USB

The USB 3.0 protocol allows a SuperSpeed host and device to negotiate with each other and decide when to bring their link into a lower power state (U1 or U2). This is accomplished by each side making individual decisions about whether they have any traffic currently pending or if they expect any traffic soon. The device requests low power entry by transmitting an LGO_U1 (or LGO_U2) Link Command and exits low power by using an LFPS handshake. The USB 3.0 Device Controller uses information from the host (bus side) and from software (application side) to decide when the best time to request low power entry is and when to exit from low power. The same calculation is used when the controller decides whether to accept a low power request from the host. The controller maintains state information about each endpoint and whether it is "active" or "paused". If all enabled endpoints are paused from either the application side or the bus side, low power entry is allowed. If at least one endpoint is not paused, low power entry is not allowed. To indicate that it does not have any pending traffic and allow the link to go into a low power state, the software may pause all enabled endpoints using one of the following mechanisms:

- Allow transfers to complete and not start any new ones
- Not setting the HWO bit in the TRBs to '1'

Exception: Software may prepare a Setup TRB for control endpoints without affecting the ability of the link going into low power. Otherwise, an endpoint is considered active on the application side. For these endpoints, the host indicates that it is paused in one of the following ways:

- Setting the Packet Pending (PP) flag to '0' in the DPH
- Setting the PP flag to '0' and NumP to '0' in the ACK TP
- For interrupt endpoints, by not polling the endpoint for two service intervals
- Note: After an endpoint is configured using the DEPCFG command and before the host sends any packets to it, the endpoint is considered to be paused on the bus side

Exception: If the data packet payload has a CRC error, the core responds with an ACK (Retry=1), and the endpoint is not paused. Otherwise, if PP=1 or Deferred=1, the endpoint is considered to be active on the bus side. Control endpoints have an additional condition as follows:

The core does not allow low power entry during a control transfer (starting from the reception of setup packet and ending at the completion of the status stage).If the core is configured for isochronous support, the following additional rules apply:

- The controller enters U0 at SystemExitLatency microseconds prior to the beginning of every microframe to receive the Isochronous Timestamp Packet (ITP).
- A PING causes every isochronous endpoint to wake up and be considered active, preventing low power entry.
- An isochronous endpoint is paused automatically when it transmits or receives the last packet of its interval (lpf=1) or if two intervals pass and the endpoint has not transmitted or received its last packet. In this way, after the PING, the core does not enter low power until all isochronous endpoints have moved their last packet or two intervals pass without another PING.

If all enabled endpoints are paused by software or the host, the core issues an LGO_U1 (or LGO_U2) link command or it responds with an LAU link command to the low power request of the host or hub. Otherwise, the core rejects the low power requests with an LXU link command.

Additionally,

- For bulk endpoints, the USB 3.0 specification says that the device may go to U1/U2 500ms (tERDYTimeout) after transmitting ERDY. It is the responsibility of the software to detect no activity on a bulk endpoint for 500ms or more and issue End Transfer if there is no activity. When the transfer is removed, the endpoint allows U1/U2 entry again.
- Because U1/U2 entry is based on the presence of packets in the TX FIFO and not based on the TRBs that are available to the core, it is possible that an IN endpoint responds with a NRDY, goes to U1, then U0, then transmits an ERDY if the timing and TXFIFO size works out such that the core cannot keep at least one packet in the TXFIFO at all times.
- In addition, before the final decision made to either initiate or accept low power state, device checks the state of DCTL[INITU2ENA][ACCEPTU2ENA][INITU1ENA] [ACCEPTU1ENA]. If the device receives Set Link Function LMP with Force_LinkPM_Accept bit asserted, the device accepts low power request from its link partner independent of endpoints state and DCTL Power control settings.

11.1.2.3.6.2 Low power operation of core

When the link is in the U1, U2, U3, SS.Inactive, SS.Rxdetect, or SS.Disabled state (for USB 3.0) or the USB is in Sleep, Suspend, or V_{BUS}-off mode (for USB 2.0), the core can be put into a low power mode to save power. The core supports the following low power mode:

- **Clock-gating mode**: In this mode, the clocks connected to most of the core modules are gated off. Modules that still get clocks in the low power mode detect wakeup conditions and remove clock gating to other modules when a wakeup condition is

detected. Clock-Gating mode can be used when the USB is in any low power state. Clock-Gating mode is enabled by setting the Disable Clock Gating bit in GCTL to 0.

11.1.2.3.6.2.1 Clock-Gating Mode

Clock-Gating mode is enabled if the Disable Clock Gating bit in the GCTL is set to 0. When the USB/link is in a low power state, the clocks to most of the core modules are gated off automatically by the core. When a wakeup condition is detected by the core, or when the software schedules new work to the core (in U1 or U2 states) or changes the USB/Link state, the core turns on clocks and the entire core goes into active mode.

11.1.2.4 Host programming model

Please refer xHCI specification for detailed information.

11.1.2.4.1 Initializing host registers

In order to initialize the core as a host, the application should perform the steps described in the xHCI specification. Global registers need to be re-initialized as described in Initializing global registers.

11.1.2.4.2 Host controller capability registers

For register definition, refer to Capability registers length and Host Controller Operational Registers (CAPLENGTH) to Runtime Register Space Offset Register (RTSOFF).

11.1.2.4.3 xHCI implementation details

This section discusses xHCI implementation details specific to the USB 3.0 controller.

11.1.2.4.3.1 LHCRST behavior

The xHCI Specification does not describe the programming model of light reset. It only specifies that the Operational and Runtime Registers that are not contained in the Aux Power well are at their default values.

Light reset must only be applied when USB_DCTL[Run_Stop] is equal to '0' to ensure that there is no discrepancy between hardware and software states. If light reset is applied during a transfer, the behavior on the USB is undefined, and may cause a packet to be terminated abruptly.

On light reset, the hardware resets all non-Aux registers to their default values which means that the port state (which is contained in PORTSC) does not change, but USB_DCTL[Run_Stop] is immediately set to '0' and all the context information about connected devices is lost. It is the responsibility of software to re-initialize the controller.

11.1.2.4.3.2 ENT requirements

In the xHCI Specification, the rules about when the ENT (Evaluate Next TRB) flag in a TRB must be set or cleared are not very strict. However, it states that the ENT flag must be set to '1' in the last Normal TRB when a TD ends with an Event Data TRB.

The controller requires software to follow these rules:

1. Regardless of whether the endpoint is stream-capable or not, if a TD ends with an Event Data TRB, the Normal TRB that precedes it must have ENT set to '1'. This Normal TRB may be separated from the Event Data TRB by a Link TRB
2. In all other cases, the ENT flag must be set to '0' Failure to follow these rules results in undefined behavior.

11.1.2.4.3.3 Behavior on babble error

If a babble error is detected and the received data passes all integrity checks, the host controller may write the received data (up to the expected data length) to the data buffer, and the value of the TRB transfer length field in the babble detected error transfer event shall be consistent with the number of data bytes written to the buffer.

The controller does not write the received data to the data buffer. The entire packet is discarded and the residual byte count is written to the TRB transfer length field.

11.1.2.4.3.4 Max_exit_latency_too_large message

If periodic transfers are on every microframe (Binterval 1), the controller reports max_exit_latency_too_large if PING scheduling is enabled. The xHCI scheduler must prefetch data from the system memory ahead of next microframe, manage non-periodic transfer, and schedule PING before ISOC. Because the system memory access and USB responses are not predictable Quality of Service (QoS) is not guaranteed if the xHCI host needs to perform all of the above for every microframe when U1/U2 is enabled. Therefore, the expectation is to disable U1/U2 if periodic data needs to be scheduled every microframe. In addition, because the U1 exit latency is in the order of 10usec, only very low system-level power saving is achieved even if the scheduler allows U1 transition by reducing QoS.

U1, U2, and U3 link states and exit time observed during lab testing:

- U1: Lowest power saving
  - In this state the chip is active, USB controller is inactive, only the Tx transmitter of the PHY is inactive.
  - U1 exit time per link partner: ~10 - 19 usec
- U2: Higher power saving than U1
  - In this state the chip is active, USB controller is inactive, both the Rx and Tx transceivers of the PHY are inactive.
  - U2 exit time per link partner: ~85 - 115 usec
- U3: Highest power saving
  - In this state, the chip and USB controller/PHY are inactive.
  - U3 exit time per link partner: 200 - 450 usec

Following are the recommendations for driver to handle periodic endpoints and "max_exit_latency_too_large" message:

- If Binterval of any endpoint is '1', then disable U1/U2 and do not schedule PING ("max_exit_latency" set to 0).
- If periodic scheduling is expected on every microframe, then disable U1/U2 and do not schedule PING.
- If periodic scheduling is expected only on every other microframe, then disable U2 and use U1 exit latency for "max_exit_latency" calculation.
- If ((number of hubs +1) * U2 exit latency) >= (2**(Binterval-1) * 125 usec), then disable U2.
- For all other periodic scheduling use U2 exit latency for "max_exit_latency" calculation.
- If the "max_exit_latency_too_large" error is sent to software, then disable U2 and re-issue ep- config with U1 exit latency.
- If "max_exit_latency_too_large" still happens, then disable U1 and re-issue ep_config with max_exit_latency set to 0 (no PING).

11.1.2.5 Device physical endpoint-specific commands

11.1.2.5.1 Command 1: Set endpoint configuration: DEPCFG

This command sets the physical endpoint configuration information.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-19. Command 1: Set endpoint configuration parameters: DEPCFG

Field	Description
Parameter 2	
31-0	Set to endpoint state when the Config Action field of Parameter 0 is 1. Otherwise, this is Reserved.
Parameter 1	
31	FIFO-based Set to '1' if this isochronous endpoint represents a FIFO-based data stream where TRBs have fixed values and are never written back by the core.
30	Bulk-based Set to '1' if this isochronous endpoint represents a bulk data stream that ignores the relationship of bus time to the intervals programmed in TRBs.
29-25	USB Endpoint Number bit[29:26]: Endpoint number bit[25]: Endpoint direction: 0 OUT 1 IN Physical endpoint 0 (EP0) must be allocated for control endpoint 0 OUT. Physical endpoint 1 (EP1) must be allocated for control endpoint 0 IN.
24	Stream Capable (StrmCap) Indicates this endpoint is stream-capable (MaxStreams != 0).
23-16	bInterval_m1 Set to the bInterval value minus 1. The valid values for this field are 0 through 13. The bInterval value is reported in the endpoint descriptor. When the core is operating in Full-Speed mode, this field must be set to 0.
15	Set to '1' if this bulk endpoint utilizes the External Buffer Control (EBC) mode. This limits the number of outstanding DMA transfers to one read and one write. This is a requirement for the EBC to operate properly. Using EBC without setting this bit to '1' results in undefined behavior.
14	Reserved
13-8	Bits 13-8 are used for Device Endpoint Specific Event Enable (DEPEVTEN). If an enable bit is set to 0, the corresponding event is not generated. Bit 13 Stream Event Enable (StreamEvtEn) Bit 12 Reserved Bit 11 Reserved Bit 10 XferNotReady Enable (XferNRdyEn) Bit 9 XferInProgress Enable (XferInProgEn) Bit 8 XferComplete Enable (XferCmplEn)
7-5	Reserved
4:0	Interrupt number (IntrNum) Applies to IN and OUT endpoints.

Table continues on the next page...

Table 11-19. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
	Indicates interrupt/Event Buffer number on which endpoint related interrupts for this endpoint are generated. This must be set to 0.
	Parameter 0
31-30	Config Action 0 Initialize endpoint state: This action is used when an endpoint is configured the first time. It will cause the data sequence number and flow control state to be reset. DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 0. 1 Restore endpoint state: This action is used when reconfiguring an endpoint with saved state after hibernation. It will cause the data sequence number and flow control state to be restored from DEPCMDPAR2. 2 Modify endpoint state: This action is used when modifying an existing endpoint configuration, such as changing the DEPEVTEN event enable bits, interrupt number, or MaxPacketSize. The data sequence number and flow control state will be unchanged, and DEPCMDPAR2 will be ignored. The encoding of this action is backward compatible with software that previously set "Ignore Sequence Number" to 1. When issuing a Endpoint Configuration command with Config Action=Restore, Parameter 2 must be filled in with the same value returned by the Get Endpoint State command prior to hibernation.
29-26	Reserved
25-22	Burst Size (BrstSiz) When field is set to- 0 Burst length = 1 1 Burst length = 2, and so on, up to, 15 Burst length = 16 For IN transfers, this value represents the maximum length of the burst that the device attempts when the Host initiates an IN transfer with a TP_ACK. If BrstSiz < the NumP value in the initiating TP_ACK, then the device controller limits the burst length to BrstSiz. If BrstSiz >= the NumP value in the initiating TP_ACK, then the device controller attempts a burst length of NumP. For OUT transfers, this value represents the NumP value utilized in the response TP_ACK from the device controller. The NumP value indicates (to the host) the burst length the device desires. Note: In the special case of BrstSiz=0 (burst length = 1), the TP_ACK from the device controller has NumP=0; which is a flow-control condition. If this value is utilized, then the endpoint enters flow control for each DP received and a subsequent ERDY is transmitted (according to the USB Specification). However, this may have an undesirable impact on the system throughput. To avoid this impact, it is recommended to set BrstSize=1 to prevent the flow-control condition. The trade-off is that the host could potentially burst two OUT data packets to the device, resulting in an ACK for the first packet, and an NRDY for the second packet.
21-17	FIFO Number (FIFONum) Indicates which transmit FIFO is assigned to this endpoint. For control endpoints, the FIFONum value in the OUT direction must be programmed to the same value as the IN direction. This field should be set to 0 for all other OUT endpoints. Even though there may be more than 16 TXFIFOs in DRD mode, the device mode must use lower 16 TXFIFOs.
16-14	Reserved

Table continues on the next page...

Table 11-19. Command 1: Set endpoint configuration parameters: DEPCFG (continued)

Field	Description
13-3	Maximum Packet Size (MPS) Applies to IN and OUT endpoints. The application must program this field with the maximum packet size (in bytes) for the current USB endpoint. USB 3.0 supports up to 1024 bytes.
2-1	Endpoint Type (EPType) This is the transfer type supported by this USB endpoint. 00 Control 01 Isochronous 10 Bulk 11 Interrupt
0	Reserved

11.1.2.5.2 Command 2: Set endpoint transfer resource configuration (DEPXFERCFG)

There must be only one transfer resource allocated per endpoint. Start transfer causes the use of the transfer resource. End Transfer or an XferComplete event releases the transfer resource. If software attempts to allocate more transfer resources than have been configured in the hardware, this command will return an error in the CmdStatus/ EventStatus field.

NOTE

If GUSB2PHYCFG[SUSPENDUSB20] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-20. Command 2: Set endpoint transfer resource configuration parameters: DEPXFERCFG

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved
Parameter 0	
31-16	Reserved
15-0	Number of Transfer Resources (NumXferRes) Defines the number of Transfer Resources allocated to this endpoint. This field must be set to 1.

11.1.2.5.3 Command 3: Get endpoint state (DEPGETSTATE)

After Get Endpoint State (DEPGETSTATE) command completes, the 32 bits in DEPCMDPAR2 represent the endpoint state to be saved. This includes the current data sequence number, flow control state, and control transfer state (for control endpoints).

This command is only used when the controller enters hibernation. The controller requires the 32 bits of state information when exiting hibernation as part of the DEPCFG command with the Config Action field set to "Restore endpoint state".

NOTE

If GUSB2PHYCFG[6] is set to ‘1’, it must be set to '0' prior to issuing this command and may be set to ‘1’ after the command completes.

11.1.2.5.4 Commands 4 and 5: Set Stall and Clear Stall (DEPSSTALL, DEPCSTALL)

These commands apply to non-isochronous IN and OUT endpoints only.

The application issues a Set Stall command to stall all tokens from the USB host to this endpoint. If the endpoint is in the NRDY state, the STALL state takes priority.

If a transaction is currently in progress when the software issues Set Stall, the behavior depends on the direction of the endpoint:

- For OUT endpoints, the current packet will complete. The core will respond with STALL to the next OUT DP or token.
- For IN endpoints in SuperSpeed, the current burst transaction will complete, and the core will respond with STALL to the next ACK TP. For other speeds, the core will complete the current packet and respond STALL to the next IN token.

For non-control endpoints, the application is responsible for both setting and clearing STALL via the Set Stall/Clear Stall commands. When the application clears the STALL, the endpoint's data sequence number is reset to zero.

For control endpoints, the application issues only the Set Stall command, and only on the OUT direction of the control endpoint. The controller automatically clears the STALL when it receives a SETUP token for the endpoint.

The application must not issue the Clear Stall command on a control endpoint. If the endpoint is in flow control (NRDY) and also in the STALL state, it responds with a STALL for any packet. The only exception is that the controller always responds to SETUP data packets with an ACK handshake, independent of the STALL state.

NOTE

- If GUSB2PHYCFG[6] is '1', it must be set to '0' prior to issuing this command and may be set to ‘1’ after the command completes.
- When issuing Clear Stall command for IN endpoints in SuperSpeed mode, the software must set the "ClearPendIN" bit to '1' to clear any pending IN transcations, so that the device does not expect any ACK TP from the host for the data sent earlier.

11.1.2.5.5 Command 6: Start transfer (DEPSTRTXFER)

This command applies to IN and OUT endpoints.

A Transfer Descriptor (TD) in Device mode is a list of TRBs that comprises one or more transfers. The pointer to a TD is added to the core's cache by the Start Transfer command and removed by an XferComplete event or by the End Transfer command.

Software issues this command indicating that a descriptor is ready to be processed and DMA can start for this endpoint/stream combination. For IN endpoints, this causes the descriptor to be processed, and data is moved from the corresponding memory buffer to corresponding transmit FIFO when the transfer is started. For OUT endpoints, this causes the descriptor to be processed and data is moved from the receive FIFO to corresponding memory buffer.

In response to the Start Transfer command, the hardware assigns this transfer a resource index number (XferRscIdx) and returns the index in the DEPCMDn register and in the Command Complete event. This index must be used in subsequent Update and End Transfer commands.

It is illegal to issue a Start Transfer command for the same TD if it is already present in the core's cache.

Non-stream capable endpoints rules:

- The Stream ID field must be set to 0.
- The HighPri field is reserved.

Stream-capable endpoint rules:

- The Stream ID field must be set to non-zero and match the Stream ID passed into the Start Transfer endpoint command associated with this transfer. In all the TRBs of a transfer, the Stream ID fields must be the same.
- The HighPri field is reserved.

NOTE

- If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.
- Before issuing a start transfer command, software needs to verify whether the link is in U0. If the link is not in U0, software needs to bring the link to U0 by performing a remote wakeup. This is applicable to both SS and USB 2.0 speeds.
- When an IN transfer is in progress, (that is, started and not completed yet), and the software sees a link state change event to L1, then the software can use the GDBGFIFOSPACE register (write followed by a read) to determine if the TXFIFO is empty or not. If the TXFIFO is not empty, it can initiate a remote wakeup.

Table 11-21. Command 6: Start transfer parameters: DEPSTRTXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Transfer Descriptor Address (TDAddr Low) Indicates the lower 32 bits of the external memory's start address for the transfer descriptor. Because TRBs must be aligned to a 16-byte boundary, the lower 4 bits of this address must be 0.
Parameter 0	
31-0	Transfer Descriptor Address (TDAddr High) Indicates the higher 32 bits of the external memory's start address for the transfer descriptor.

11.1.2.5.6 Command 7: Update transfer (DEPUPDXFER)

If software uses circular TRB buffers and updates a TRB, whose Hardware Owner (HWO) bit was 0, by setting HWO=1, it must execute the Update Transfer command, specifying the transfer resource index of the TRB in the DEPCMD register. The hardware uses this information to re-cache the TRB.

Software may issue a special “No Response Update Transfer” command by setting CmdAct=0 and CmdIOC=0. In this case, the hardware does not generate a Command Complete event, does not set the CmdAct bit to '0' (because it will be '0'), and software may immediately issue another command to the same endpoint following this one. This

special type of Update Transfer may not be used when software depends on the XferNotReady event to setup TRBs (see “On Demand” transfers in Transfer setup recommendations).

Software may issue an Update Transfer command for a transfer resource that has already completed (either due to XferComplete event or an End Transfer command), and the core will detect that the Update Transfer is unnecessary. However, software must not issue an Update Transfer command for a transfer resource index that has never been started.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

11.1.2.5.7 Command 8: End transfer (DEPENDXFER)

This command applies to IN and OUT endpoints. Software issues this command requesting DMA to stop for the endpoint/stream specifying the transfer resource index of the TRB and the ForceRM parameter to be set to 1 in the DEPCMD register.

When issuing an End Transfer command, software must set the CmdIOC bit (field 8) so that an Endpoint Command Complete event is generated after the transfer ends. This is necessary to synchronize the conclusion of system bus traffic before the End Transfer command is completed.

For IN endpoints, this command causes descriptor processing to stop and the core stops fetching new data for the endpoint and stops transfers on the USB. The core may truncate a packet being transmitted with the DPPABORT ordered set, does not wait for any pending ACKs from the USB, and does not update the TRB status. For OUT endpoints, this command causes descriptor processing to stop. If there is currently data, it is moved from the receive FIFO to corresponding memory buffer and completed at packet boundary. The core does not update the TRB status.

Use this command under the following conditions:

- When handling USBReset or SetConfiguration, endpoints are closed using this command.
- After receiving a ClearFeature (STALL) control transfer, software issues End Transfer followed by Clear Stall, followed by Start Transfer.
- After an XferInProgress event when the TRB after the one that caused the XferInProgress event has its HWO bit set to '0', this command can be used.
- For isochronous endpoints, if the host stops moving data for many intervals, software may force the end of the transfer and wait for the host to restart.

The hardware does not issue a XferComplete event on End Transfer, but only issues a CommandComplete event.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

Table 11-22. Command 8: End Transfer Parameters: DEPENDXFER

Field	Description
Parameter 2	
31-0	Reserved
Parameter 1	
31-0	Reserved
Parameter 0	
31-0	Reserved

End Transfer Command With Hibernation Enabled The following applies only when hibernation is enabled. Prior to hibernation, software will set ForceRM to '0' when it issues End Transfer commands for active transfers. This command may cause a TRB to be written back with intermediate state.When software issues an End Transfer command with the ForceRM bit set to 0, and the core has at least one valid current TRB, it will perform the following:

- Write back the TRB updating the Buffer Pointer and Buffer Size fields to reflect the current state of the TRB.
- Write the value '4' (TRBInProgress) into the TRBSTS field to indicate that this TRB may have been partially used and is still active. This is an indication to software that the original buffer pointer field may have been changed by hardware.
- Set the HWO bit to '0' unless it is a Link TRB. The HWO bit in a Link TRB will be unchanged.

After the state is restored, software may restart a transfer from a partially completed TRB by setting TRBSTS to '0', setting the HWO bit back to '1', and issuing the Start Transfer command.Software is not required to perform any special manipulation on partially completed TRBs. When the core completes a restarted partially complete TRB, it will set TRBSTS back to '0'.

11.1.2.5.8 Command 9: Start new configuration (DEPSTARTCFG)

Software issues this command under the following conditions:

- After power-on-reset with XferRscIdx=0 before starting to configure Endpoints 0 and 1. CmdIOC must be set to '0' and software must poll the CmdAct bit to determine when the command is complete because Endpoint 0 is not yet configured with a valid interrupt number.
- With XferRscIdx=2 when it receives SetConfiguration before starting to configure Endpoints > 1. CmdIOC may be set to '0' or '1'.

This command should always be issued to Endpoint 0 (DEPCMD0).

Hardware resets the transfer resource allocation to the value in the XferRscIdx parameter (must be 0 or 2) upon receiving this command.

NOTE

If GUSB2PHYCFG[6] is set to '1', it must be set to '0' prior to issuing this command and may be set to '1' after the command completes.

11.1.2.6 OTG

This section describes OTG for USB 3.0, and the programming requirements for the USB 3.0 core in OTG mode.

11.1.2.6.1 OTG 2.0 for USB 3.0 Functionality

The following sections describe the OTG 2.0 functions for USB 3.0.

11.1.2.6.1.1 Core OTG functions

The OTG Interface block within the U2MAC handles the core OTG functions: Session Request Protocol (SRP) and Host Negotiation Protocol (HNP). These OTG protocols are implemented through the regular UTMI+ OTG interface.

The MAC handles HNP for host and peripheral role swapping.

The MAC also handles SRP, which allows an A-Device to turn off VBUS to save power when the USB is not used, and provides a means for a B-Device to monitor VBUS and request the A-Device to activate VBUS.

It includes the following logic necessary to achieve SRP and HNP:

- Control of VBUS through Vbus drive enable (utmiotg_drvvbus) as A-Device
- Control of IDDIG line sampling enable control output (utmiotg_idpullup)

- Control of D+/D- pull-down resistor enables (utmiotg_dppulldown/ utmiotg_dmpulldown)
- Generates SRP (data line pulsing) as B-Device when the session is off.

NOTE

- When the application programs the USB_GCTL[PRTCAPDIR] as 2'b11 to enable OTG mode, then in the A- Device mode, the core will only enumerate as a HS device and hence will not support any SS-capable device to be connected. This restriction is not valid when the USB_GCTL[PRTCAPDIR] is programmed to 2'b01 or 2'b10.
- The utmisrp_chrgvbus and utmisrp_dischrgvbus outputs are provided for legacy PHY connections but they are both set to 1'b0 from the core since VBUS charging is not supported in OTG 2.0 specifications.
- The application should not program GCTL[0] as 1'b0 when it wants to do a ADP or HNP. In such a case, the application should disable the Clock Gating feature by programming GCTL[0] as 1'b1. The application should re-enable the Clock Gating feature only when the core returns to its original role of operation.

11.1.2.6.1.1.1 HNP Polling and Enable

This HNP Polling activity involves an OTG device acting as the current host to periodically poll the remote device to check if the remote device wishes to take the host role. It will then grant the role swap opportunity at the earliest opportunity. Being a software activity, HNP Polling is expected to be implemented through software timers and periodic exchange of SetFeature.SetHNPEnable packets.

The core is then informed of successful exchange of these packets to Enable HNP activity within the core.

11.1.2.6.1.2 ADP functions

The ADP functions involve the following two processes:

- ADP sensing
- ADP probing

ADP probing capability is required in both A and B devices, while sensing is a must only for B-Device. The main functional unit is the ADP controller. You can choose to have the ADP controller logic supplied along with the USB 3.0 core or can choose to implement proprietary ADP controller logic.

11.1.2.6.1.2.1 Internal ADP controller

In this option, ADP controller logic is supplied alongside the USB 3.0 controller by enabling DWC_USB3_EN_ADP coreConsultant parameter. ADP controller logic is inferred along with outside USB 3.0 core functions within the controller.

- All ADP timers are maintained in pwrm (Power) module and generate the enable and disable signaling to PHY for ADP probing and sensing.
- PHY contains the following circuitry related to ADP functionality:
  - Comparators for PRB and SNS
  - I_ADP_SRC and I_ADP_SNK
  - Vbus circuitry
- Application software programs the ADP timing registers residing in the pwrm module.
- pwrm module provides a mechanism to application through the interrupts to log and report events pertaining to ADP probing and sensing.
- Only pwrm module needs to be always powered on in this option. USB 3.0 controller may or may not be powered on.
- As a product, both the ADP controller logic and OTG controller are packaged into USB3 core. In the USB3 core, ADP sensing and probing is loosely coupled with the rest of the logic and is directly under software control. This achieves maximum flexibility to interact with OTG core functions.

The ADP controller in the block diagram is a fully digital controller. This has mainly timers inside it to help ADP operation. All ADP related timers are part of pwrm module.

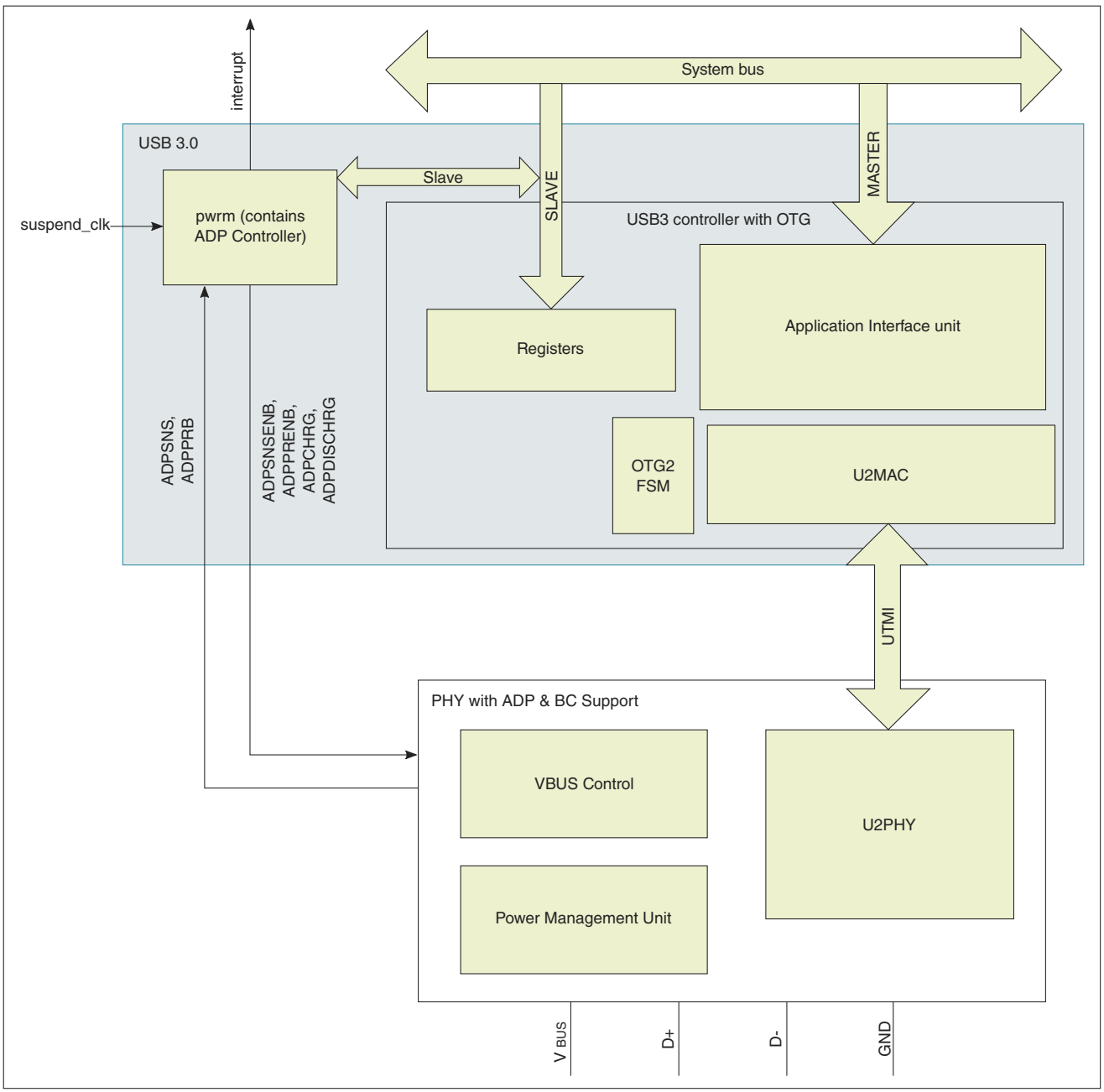


Figure 11-18. Internal ADP controller

For the PHY to generate ADPPRB and ADPSNS, the PHY needs to know the value to compare with VADP_PRB and VADP_SNS respectively. These values can be hardcoded in the PHY or the PHY can have a limited programmable option by which these reference voltage values can be changed. It is assumed that the PHY takes care of this. The pwrm module also implements SRP detection in A-Device mode. This is required if pwrdwn module of the USB 3.0 controller is powered down completely while ADP is in progress and B-Device initiates SRP.

NOTE

- The ADP block works with the suspend clock.
- The ADP timers run with the suspend clock along with the programmed USB_GCTL[PWRDNSCALE] value. The USB_GCTL[PWRDNSCALE] value is used internally to generate 32 KHz pulse for the timers.

11.1.2.6.1.2.2 External ADP Controller Option

In this option, ADP controller logic supplied with USB 3.0 controller is not required. The following points apply:

- ADP logic is controlled directly by software.
- Application software controls ADP probe/sense enable and disable directly. The USB 3.0 controller can be completely powered off during this period.
- All ADP timers programmed are programmed by application through a dedicated interface to the ADP controller.
- Any ADP_CHANGE status from the external ADP controller is handled by the application directly.

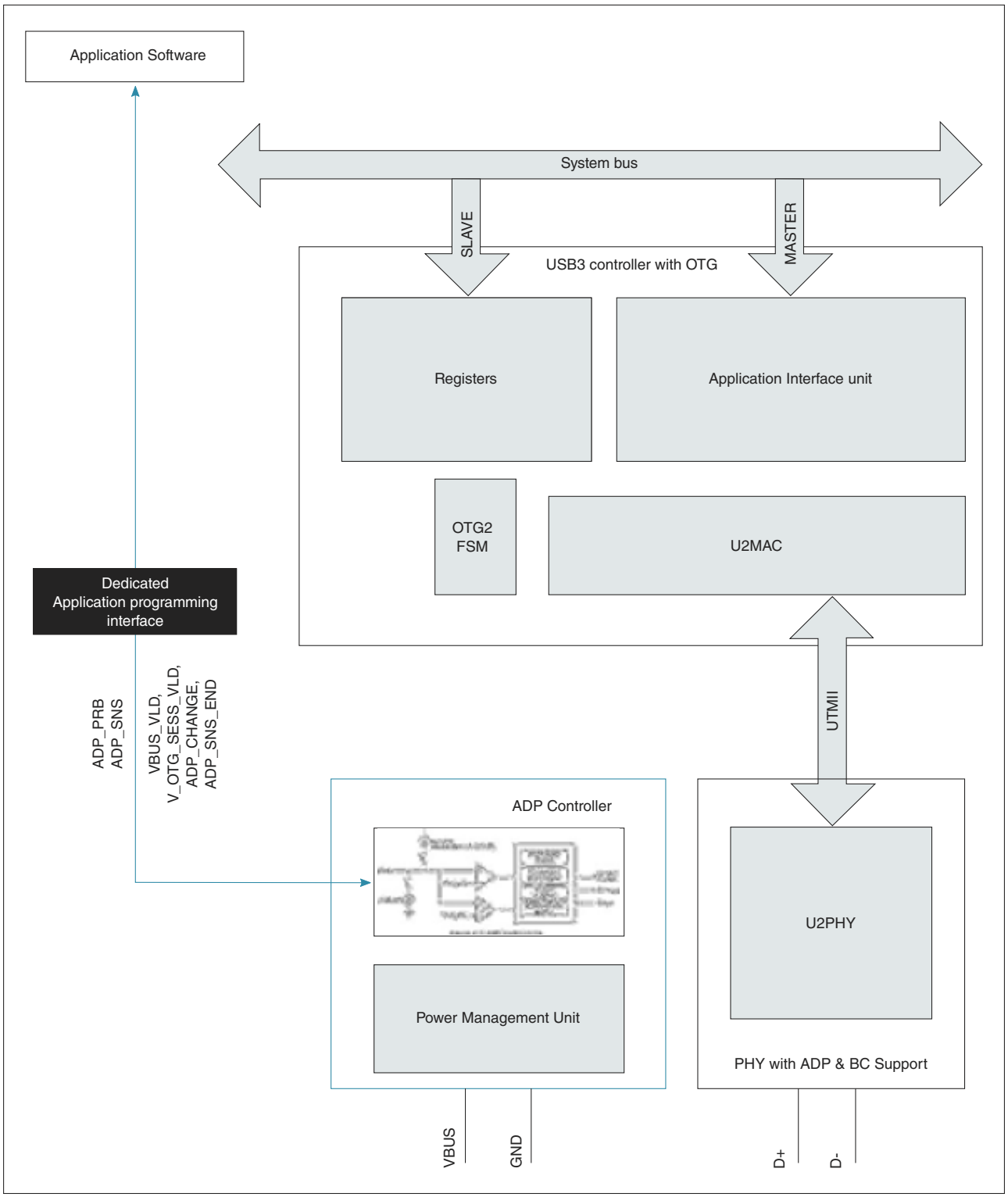


Figure 11-19. External ADP Controller Option

11.1.2.6.1.3 Software flow

The software flow diagram is illustrated in the following figure.

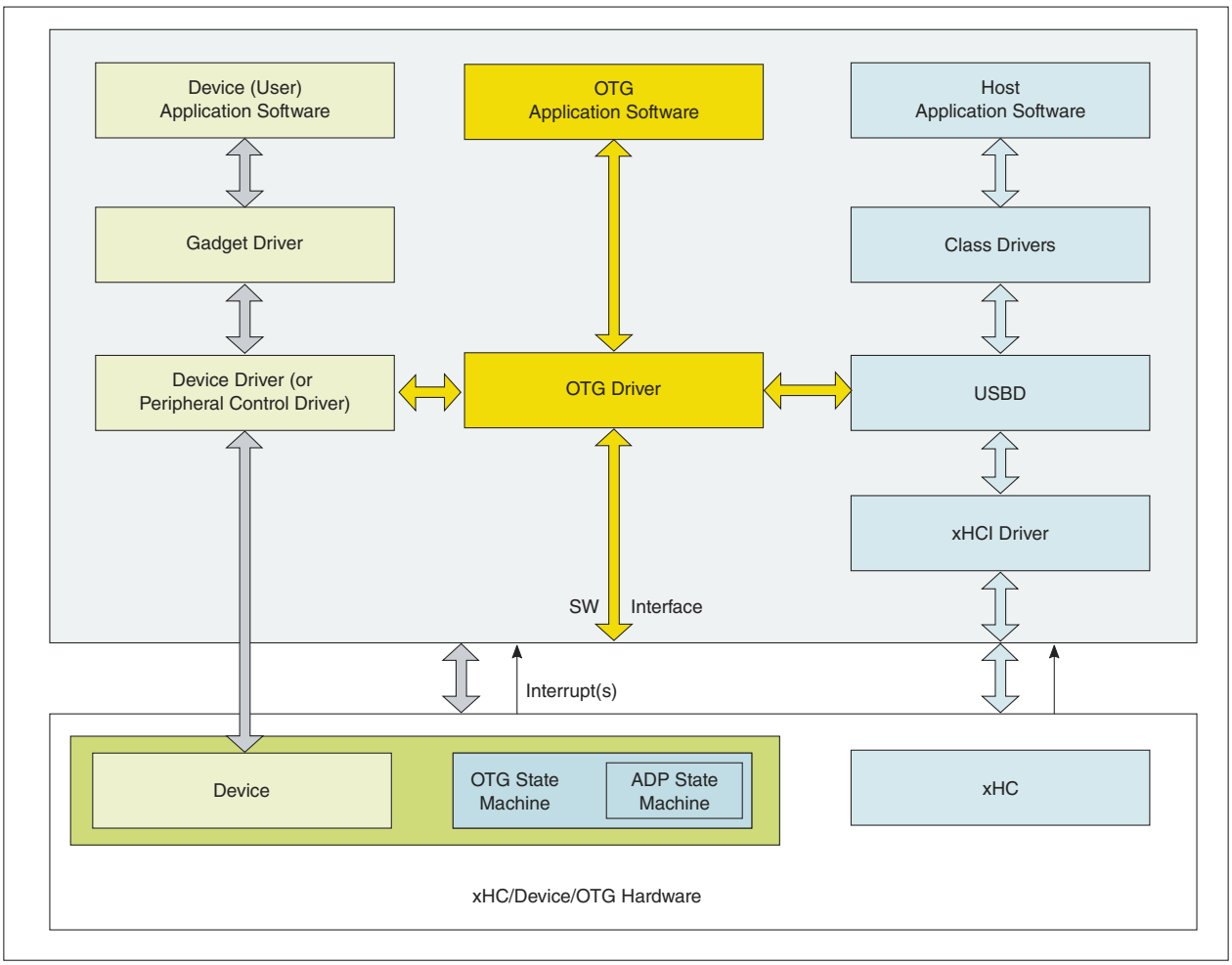


Figure 11-20. Software flow diagram

The following two sections describe the A-device and B-device flows briefly with respect to the above flow diagram.

NOTE

The following concise flows are not very detailed, and are provided here to help understand at an abstract level.

11.1.2.6.1.3.1 A-device activity concise flow

- Power-on-reset
- Only the OTG driver is active after power-on-reset. The host and device drivers are inactive after power-on-reset since at this point of time it is unsure what role the device assumes.
- The OTG driver initializes ADP and waits until a successful ADP event is received.

- The OTG driver senses ID pin = 0 and initializes the OTG registers.
- The OTG driver waits until a successful SRP event is received and turns on the VBUS.
- The OTG driver requests the USBD (USB driver) to bring up the xHC.
- After successful xHC bring up, device enumeration (and data transfers) occurs during which the device capabilities are exchanged for determining a possible role switch.
- In the event where a role switch is possible, the OTG application software or the B-device may initiate a role switch (The OTG application software may request via the Host application software to create favorable conditions for a role switch).
- The xHC is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the PCD (Peripheral driver) to bring up the Peripheral.
- After successful Peripheral bring-up, device enumeration (and data transfers) occurs.
- In an event where a role switch is possible, OTG application software may initiate a role switch (The OTG application software may request via the Device application software for a role switch).
- The PCD is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the USBD to bring-up the xHC, thus reverting to the original roles.

11.1.2.6.1.3.2 B-device activity concise flow

- Power-on-reset
- Only the OTG driver is active after power-on-reset. The host and device drivers are inactive after power-on-reset since at this point of time it is unsure what role the device assumes.
- The OTG driver initializes ADP and waits until a successful ADP event is received.
- The OTG driver senses ID pin = 1 and initializes the OTG registers.
- The OTG driver initiates a SRP and waits for the VBUS to be turned on by the A-device.
- The OTG driver requests the PCD to bring up the peripheral.
- After successful peripheral bring up, device enumeration (and data transfers) occurs during which the device capabilities are exchanged for determining a possible role switch.
- In an event where a role switch is possible, the OTG application software or the A-device may initiate a role switch (The OTG application software may request through the device application software for a role switch).
- The PCD is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the USBD to bring up the xHC.
- After successful xHC bring-up, device enumeration (and data transfers) occurs.
- In an event where a role switch is possible, the OTG application software or A-device may initiate a role switch (The OTG application software may request the host application software for a role switch).

- The xHC is shut down and the control is passed on to the OTG driver.
- The OTG driver requests the PCD to bring-up the Peripheral, thus reverting to the original roles.

11.1.2.6.1.4 Programming model

This section describes the programming requirements for the USB 3.0 core in OTG mode. The core can be configured either as a DRD-device or an OTG, based on bit [13:12] of the Global Control Register's (GCTL) Power-On Initialization Value. If power-on configuration option is chosen as DRD-device, then PrtCapDir has to be explicitly programmed to 2'b11 for operating as an OTG device. The premise for the programming model is as follows.

1. The OTG driver is expected to run concurrently and independently with the PCD and xHCI drivers.
2. The OTG driver doesn't get involved in the actual data transfers but is responsible for communicating messages between the xHCI driver and PCD during role changes.
3. The OTG driver is responsible for handling the HNP, SRP and ADP (internal) events from the core. After power on, the OTG driver is responsible for loading, starting and switching between the xHCI driver and PCD. When started, the xHCI driver initializes the xHCI register set before enumerating the connected device. Similarly, when started, the PCD initializes the device register set and waits for the USB reset event to continue. Once the connection is established between the xHCI driver or the PCD by the OTG driver, the corresponding driver takes over for data transfers.

After every successful HNP switching, it is necessary that the active driver is changed from the xHCI driver to PCD or vice-versa. In such cases, the application may also unload the active driver from the memory and re-use the same memory area for loading the next active driver. The subsequent flow described in this section assumes that the core can get enumerated in any of the speeds (SS or HS/FS) and bases its discussion accordingly. For example, if the core enumerates in HS/FS, follow the OTG 2.0 flow.

11.1.2.6.1.4.1 Initializing global registers

The global registers of the USB 3.0 core are shared between the device and host modes. It is the responsibility of the individual driver to initialize the core's global register specific to its functions. This is not explicitly mentioned in the flows described in this chapter. This specifically includes initializing the GTXTHRCFG/GRXTHRCFG, GEVNTEN, and GPRTBIMAP registers, and the Global FIFO Size and Global Event Buffer Registers. For more details, refer to Global registers in the register content. However, these global registers that are programmed once at power on can be initialized in the global initialization routine in the OTG driver. Examples of one-time programmable

registers are GSBUSCFG, GCTL, GSNPSID, and GUCTL. Examples of fields that are one-time programmable are the PHY interface (UTMI), AXI parameters like burst size, and so on. The standard xHCI host driver does not get involved in the core-specific global register programming, and therefore, needs to be handled by the Board Support Package software.

11.1.2.6.1.4.2 Initializing host registers

In order to initialize the core as a host, the application should perform the steps described in the xHCI specification. Global registers need to be re-initialized as described in Initializing global registers.

11.1.2.6.1.4.3 Initializing device registers

In order to initialize the core as a device, the application should perform the steps described in the section Device Programming Model. Global registers need to be re-initialized as described in Initializing global registers. For specific registers, refer to Device Programming Model.

11.1.2.6.1.4.4 Initializing OTG registers

The application should initialize the OTG registers based on the initial value of the OSTS.ConIDSts after power on. The following sections depict the programming flow for the OTG application in detail.

11.1.2.6.1.4.5 Programming flow for OTG in USB 3.0

The following figure shows the programming flow after power-on reset.

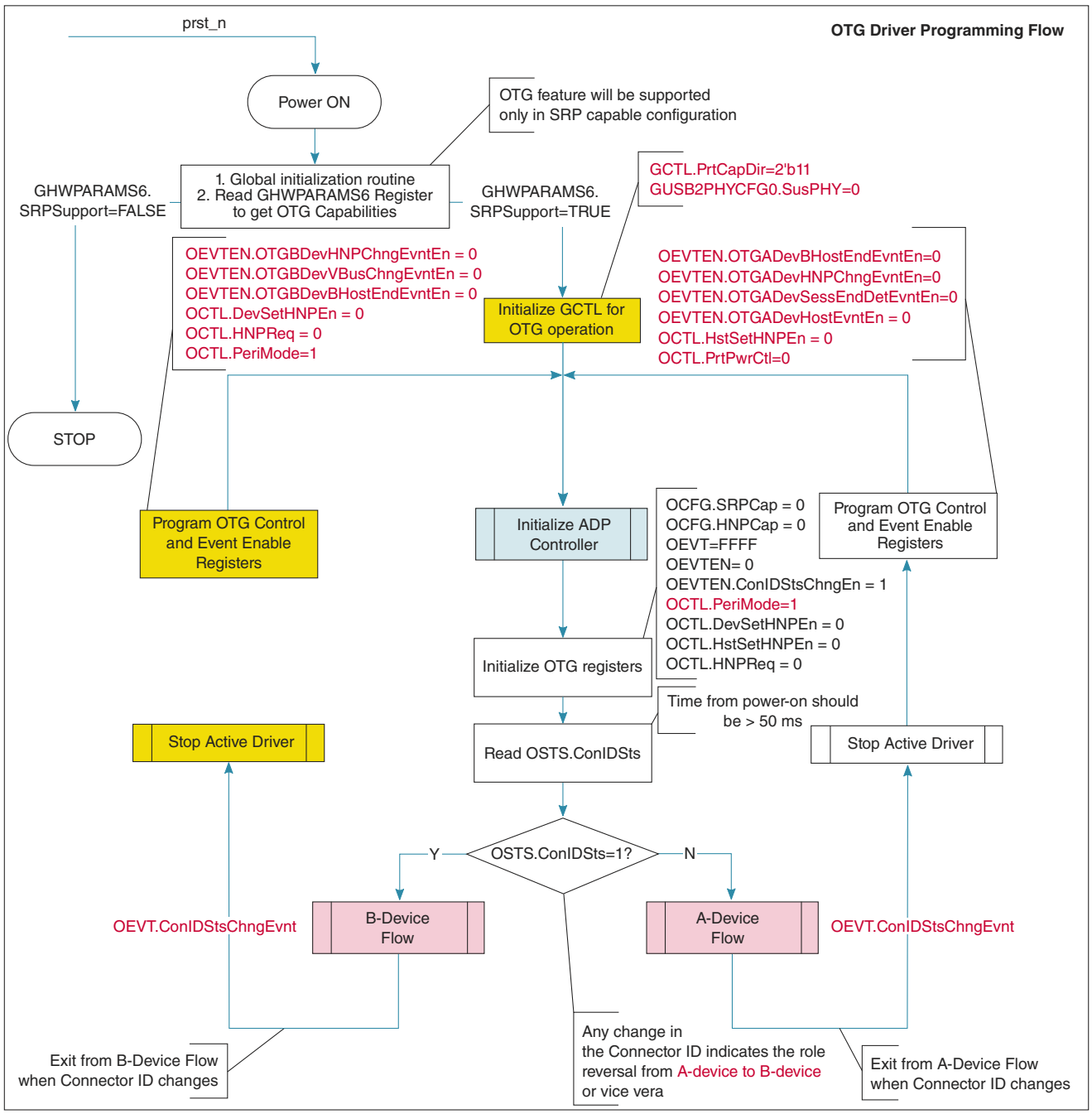


Figure 11-21. OTG driver overall programming flow

Following steps describe overall programming flow:

1. During global initialization, read the GHWPARAMS6 register to see if SRP support is enabled in the configured core. If SRP support is enabled, then proceed with the OTG programming flow by programming USB_GCTL[PRTCAPDIR] as 2'b11. If SRP support is not enabled, do not proceed further in the OTG programming flow. The configured device must at least support SRP for cable connection-based (Connector ID) role of operation.

2. Initialize the ADP controller as explained in ADP programming flow.
3. The OTG 2.0 state machine is initially in B-IDLE if IDDIG is 1, or A-IDLE if IDDIG is 0. Enable USB_OEVTEN[ConIDStsChngEvntEn] for any change in the Connector ID status.
4. Initialize the OTG control and configuration register to default values.
5. Read the OSTS register to find out the current status of the Connector ID (ConIDSts). After power-on or soft reset, the ConIDSts will be valid only after the PHY delay from IDPULL=1 to IDDIG active and any filter delay for IDDIG inside or outside the USB 3.0 core.
6. If USB_OSTS[ConIDSts] is 1, the OTG 2.0 state machine is in B-IDLE, follow the B-Device flow. Otherwise, if OTG 2.0 state machine is in A-IDLE, follow the A-Device flow.
7. When there is any connector ID (IDDIG) change resulting in USB_OEVT[ConIDStsChngEvnt], then exit the A-Device/B-Device flow. Stop the active driver and re-initialize the OTG control and status registers.

11.1.2.6.1.5 Common driver tasks

The common tasks for both A-Device and B-Device flow are as follows:

- Start xHCI driver
- Start peripheral control driver (PCD)
- Switch peripheral
- Switch host
- Stop active driver
- Enable HNP change

The Stop Active Driver is already introduced in the common flow diagram. Other tasks are useful for the A-Device/B-Device flows.

Start xHCI driver

The OTG driver, being responsible for starting the xHCI driver, executes this task. The OTG driver loads the xHCI driver after initializing the global registers, and the xHCI driver is responsible for all the functions thereafter.

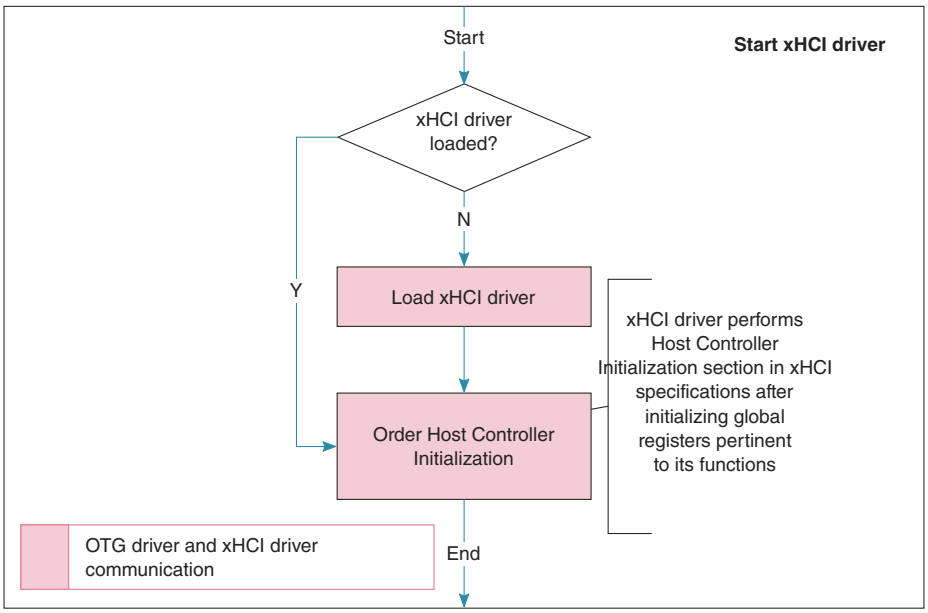


Figure 11-22. Start xHCI driver task

Start peripheral control driver

The OTG driver, being responsible for starting the Peripheral Control Driver (PCD), executes this task. The OTG driver only loads the PCD, which is responsible for all the functions thereafter, including the global registers initialization specific to the USB 3.0 core. Note that except the GTXFIFOSIZ and GEVNTADR, none of the other global registers need to be potentially re-programmed with different values.

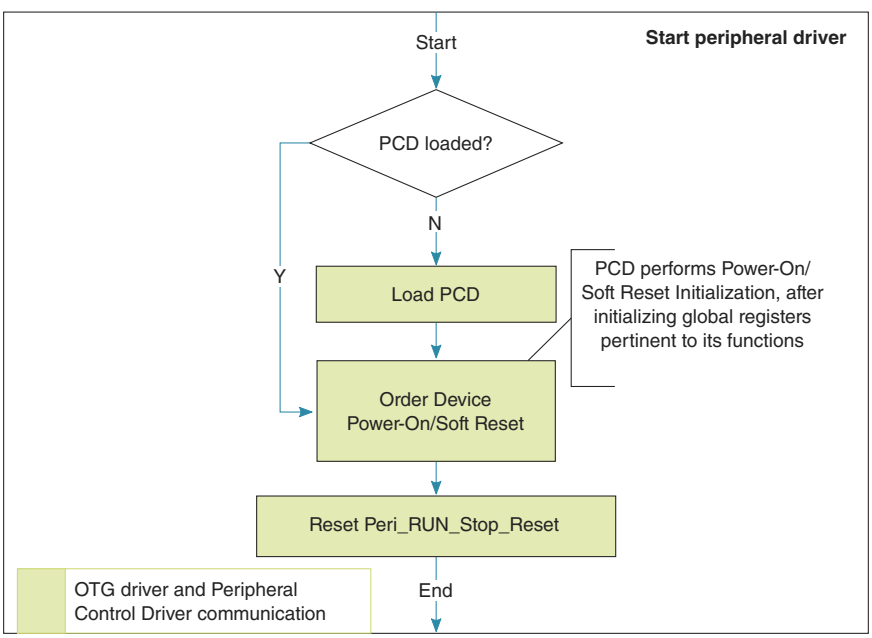


Figure 11-23. Start peripheral driver task

Stop active driver

This task checks which driver is currently active and stops the active driver. Optionally, it may unload the driver if the OS supports dynamic loading of drivers.

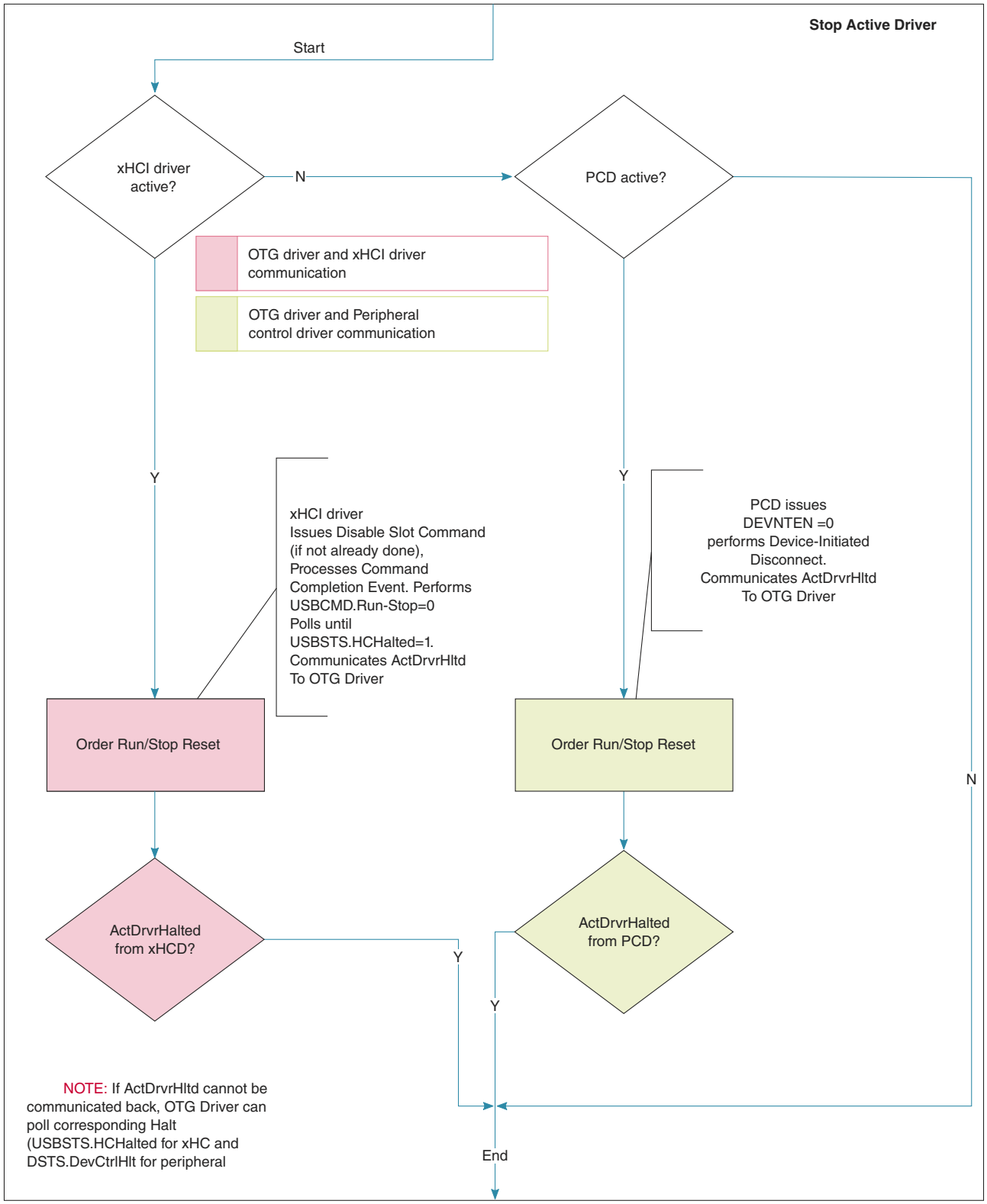


Figure 11-24. Stop active driver task

Switch peripheral and switch host

These tasks are used during HNP and each consists of two individual tasks.

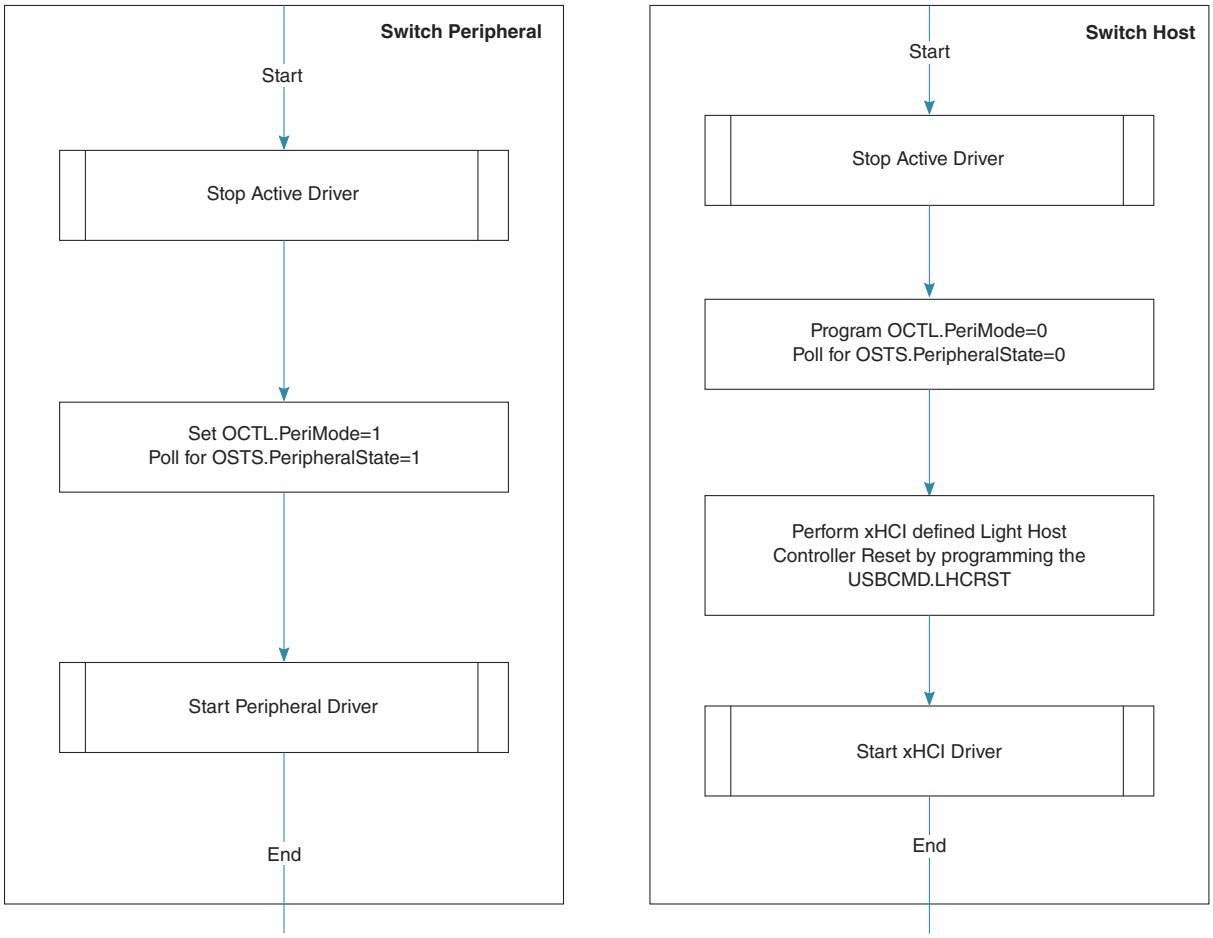


Figure 11-25. Switch peripheral task and switch host task

Programming flow for switch peripheral in SS mode:

This programming flow indicates the steps to be followed by the software when Host to Peripheral role change is intended. Note that the handover of control from the xHCI driver to PCD is accomplished using the OTG driver. The OTG driver plays the role of a mediator here. The steps indicated below is the sequence in which the xHCI driver stops its operation.

1. The xHCI driver decides to end all data transfers.
2. The xHCI driver services all pending interrupts.
3. The xHCI driver issues disable slot command.
4. The xHCI driver waits for the successful completion of disable slot command.
5. The xHCI driver clears the connect status change and port enabled status change bits of the PORTSC0 register.
6. The xHCI driver programs run/stop as 1'b0.
7. The xHCI driver waits for the HCH bit to be set in the USBSTS register.

8. When the HCH bit is set, the xHCI driver hands over the control to the OTG driver.

Programming flow for switch host in SS mode:

This programming flow indicates the steps to be followed by the software when Peripheral to Host role change is intended. Note that the handover of control from the PCD to xHCI driver is accomplished using the OTG driver. The OTG driver plays the role of a mediator here. The steps indicated below is the sequence in which the PCD stops its operation.

1. The PCD sends a Device Notification packet for role swap.
2. The PCD services all pending interrupts.
3. The PCD programs DCTL.Run bit to 1'b0.
4. The PCD waits for DevCtrlHlt = 1'b1.
5. When the DevCtrlHlt bit is set, the PCD hands over the control to the OTG driver.

Enable HNP change

For HS/FS, this task is called when SetFeature.SetHNP is successfully exchanged between the host and the device. The set Peri_Run_Stop_Reset indicates to the device driver to automatically disable all events and perform a peripheral-initiated disconnect when the host puts the device in Suspend. Program the OCTL.DevSetHNPEn as 1, and OCTL.DevSetHNPReq as 1 immediately after the successful completion of SetFeature.SetHNP on the USB. This ensures the HNP attempt from the OTG core in the following Suspend state.

NOTE

By default, at power-on, program GUSB2PHYCFG.SusPHY as 1 to ensure that the PHY enters low power mode during normal suspend/resume without role switch.

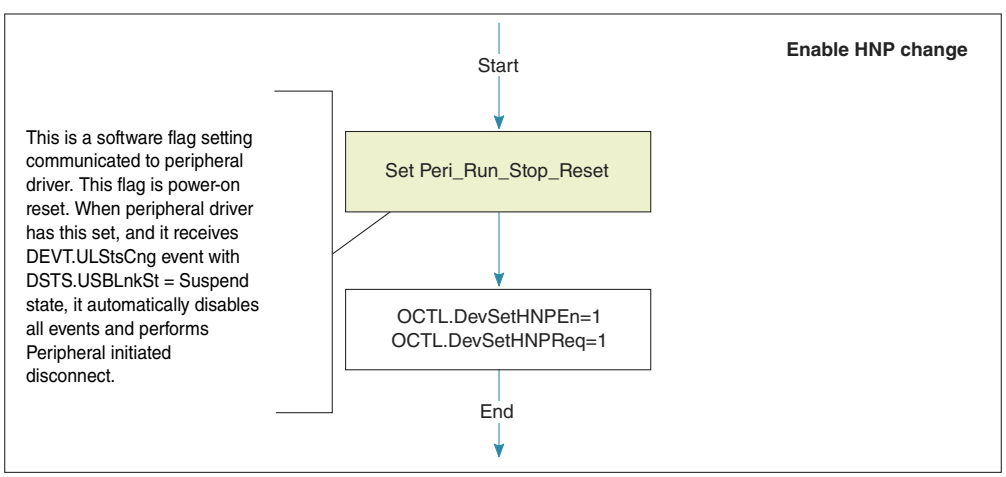


Figure 11-26. Enable HNP change

11.1.2.6.1.6 A-device flow

The following figure shows the A-device flow.

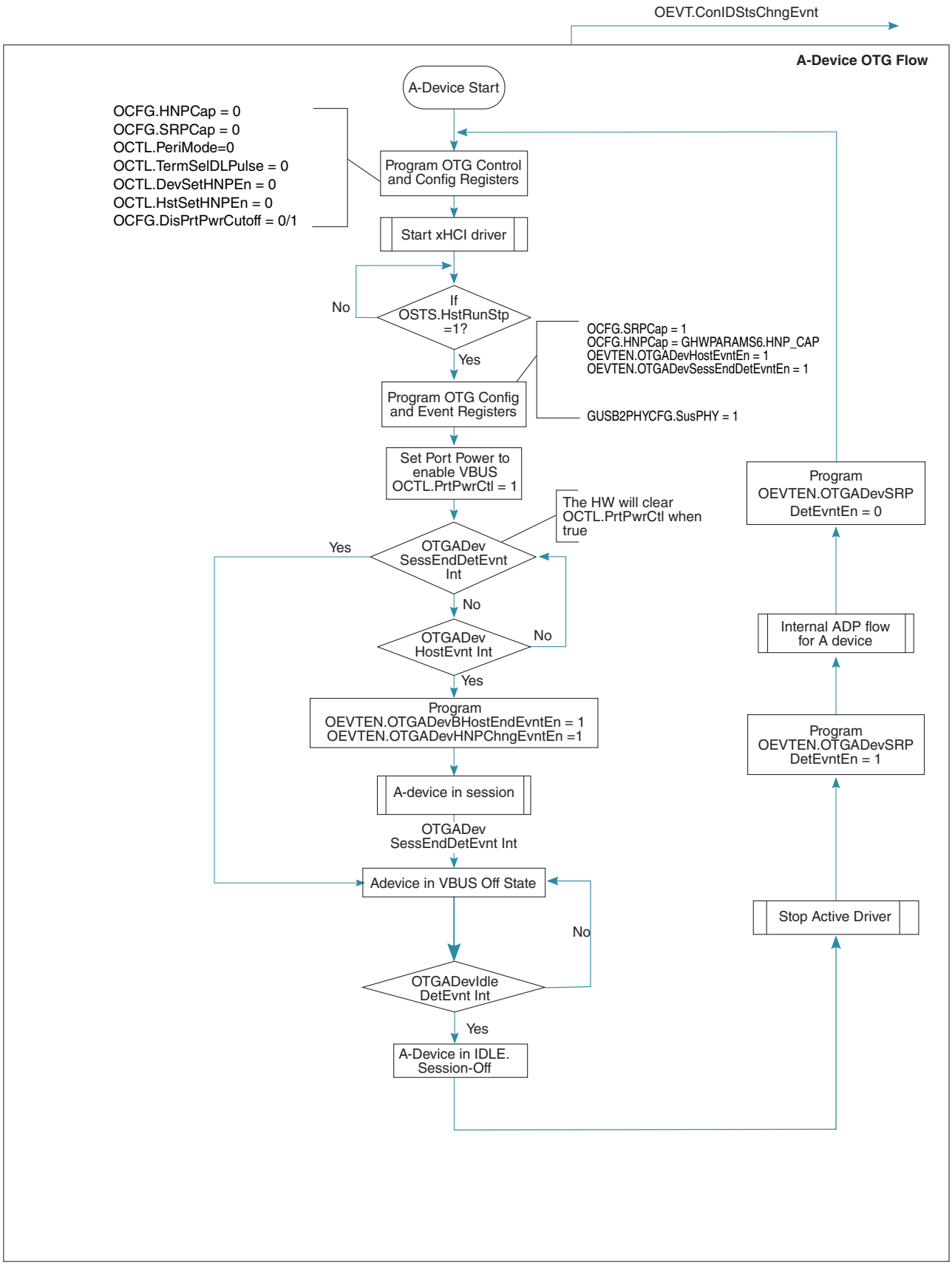


Figure 11-27. A-Device flow diagram

A-device flow:

1. At Start, the A-device core will be in A_IDLE VBUS off state.
2. The OTG software should program the OTG control and configuration registers for A-device operation:
   - OCFG[HNPCAP] = 0
   - OCFG[SRPCAP] = 0
   - OCTL[PERIMODE] = 0
   - OCTL[TERMSELDLPULSE] = 0
   - OCTL[DEVHNPEN] = 0
   - OCTL[HSTSETHNPEN] = 0
   - OCFG[DISPRTPWRCUTOFF] = 0 or 1, based on whether the application requires the core disable the feature to automatically switch off VBUS after 1 second if port is disconnected.
3. Start the xHCI driver.
4. The OTG software should wait for OSTS[HSTRUNSTP] to become 1 to continue. If OSTS[HSTRUNSTP] is 0, the OTG Software should wait for a software event that indicates the change in the xHCI RUN_STOP bit to continue. Alternatively, the OTG Software can poll for OSTS[HSTRUNSTP] bit to be set.
5. OTG Software should program the OTG configuration and event registers to enable A-device events to detect peripheral connect and session end. Refer to A-device flow diagram.
6. Program GUSB2PHYCFG0[SUSPHY] = 1 to enable the core to switch off the PHY clock in the possible non-HNP suspend scenarios.
7. Set OCTL[PRTPWRCTL] when OSTS[XHCIPRTPOWER] is 1. This enables driving of VBUS to the B-device.
8. The A-device may enumerate in HS/FS based on the device signaling.
9. The OTG 2.0 state machine enters a3_ds_host/A-host for HS (FS) if there is a successful B-device connect (ADEVHOSTEVNT).
10. The OTG software should enable the ADEVHNPCHNG and ADEVBHOSTEND events to detect the transition from A-host to A-peripheral and the end of A-peripheral events.
11. If there is an ADEVSESSENDDETEVNT interrupt indicating that the core has now stopped driving the VBUS, the software should wait for ADEVIDLEDETEVNT interrupt and then disable all the OTG events.
12. The Software can enable the ADP flow.
13. After an ADEVHOSTEVNT interrupt, the core is in session. The flow for the A-device in session is shown in earlier in this chapter.

NOTE

- At any step, if CONIDSTSCHNGEVNT occurs, A-device flow is terminated as per the figure "OTG Driver Overall Programming Flow".
- Over current is not detected by the OTG state machine. The xHCI controller detects it, and the OTG driver in turn sees an ADEVSESSENDDETEVNT (from PORTSC[PP] = 0) due to an overcurrent.

OTG state machine mapping to the software flow

- Steps 1-5 are A-IDLE.
- Step 7 is A_WAIT_VRISE in which A-device has detected SRP and therefore waits for VBUS to stabilize.
- Step 8 and step 10 are primarily A_HOST through A_WAIT_VRISE and A_WAIT_BCON. If there is an error in either A_WAIT_VRISE or A_WAIT_BCON, ADEVSESSENDDETEVNT occurs making the state-machine fall back to A-IDLE.

Handling over-current in OTG mode of operation

- When the core is operating in USB 2.0 mode, an over-current condition causes PORTSC[OCA] = 1 and PORTSC[OCC] = 1 on the corresponding USB 2.0 port number. The core also updates a port status change event TRB for the corresponding USB 2.0 port number.
- When the core is operating in USB 3.0 mode, an over-current condition causes PORTSC[OCA] = 1 and PORTSC[OCC] = 1 on the corresponding USB 2.0 and USB 3.0 port numbers. The core also updates a port status change event TRB, one each for the corresponding USB 2.0 port number and the corresponding USB 3.0 port number.
- Note that PORTSC[OCA] may transition quickly from 0 to 1 and 1 to 0 before xHCI driver can see two distinct interrupts. This may occur due to over-current condition being removed quickly. In such a case, the xHCI driver will receive a single interrupt (PORTSC[OCC]) for over-current.

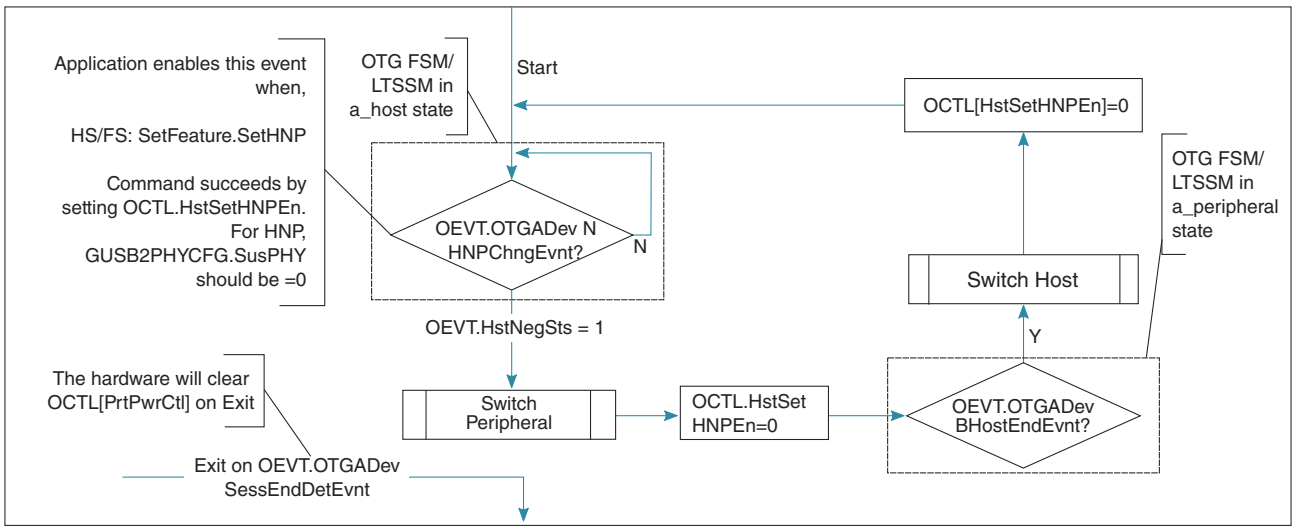


Figure 11-28. A-host -> A-peripheral -> A-host flow diagram

For HS/FS mode:

1. When the core starts the session with A-device as host, the application may periodically exchange SetFeature.SetHNPEnable packets. If B-device wants to become the host, this packet exchange results in the host application knowing about B-Device's intent to become the host, and it directs the OTG driver to set OCTL[HSTSETHNPEN]. This enables the core to cater for HNP from B-device when suspend is initiated. During suspend, ADEVHNPCHNGEVNT is set.
2. If ADEVHNPCHNGEVNT is set, the OTG driver reads OEVT[HSTNEGSTS]. If OEVT[HSTNEGSTS] is set, then the core transitions to A-Peripheral state by executing Switch Peripheral task. Otherwise, wait for ADEVSESSENDDETEVNT and go to ADP probing.
3. When in A-peripheral state, if there is ADEVBHOSTENDEVNT, it signifies that the B-device no longer wants to be the host. Therefore, the driver goes back to Step 1 by executing the switch host task and assumes an A-host role.

   Alternatively, if there is an xHCI[PRTPOWER] de-assertion, the state machine goes to start ADP probing due to the occurrence of ADEVSESSENDDETEVNT.

4. If ADEVBHOSTENDEVNT is set, then the core would be in A-host state by executing switch host task. Otherwise, perform either of the following:
   a. Drop VBUS and go to ADP probing.

      If the role change is successful, the OTG driver hands over the control to the xHCI driver.

Internal ADP flow for A-device

The internal ADP flow of A-device is as follows:

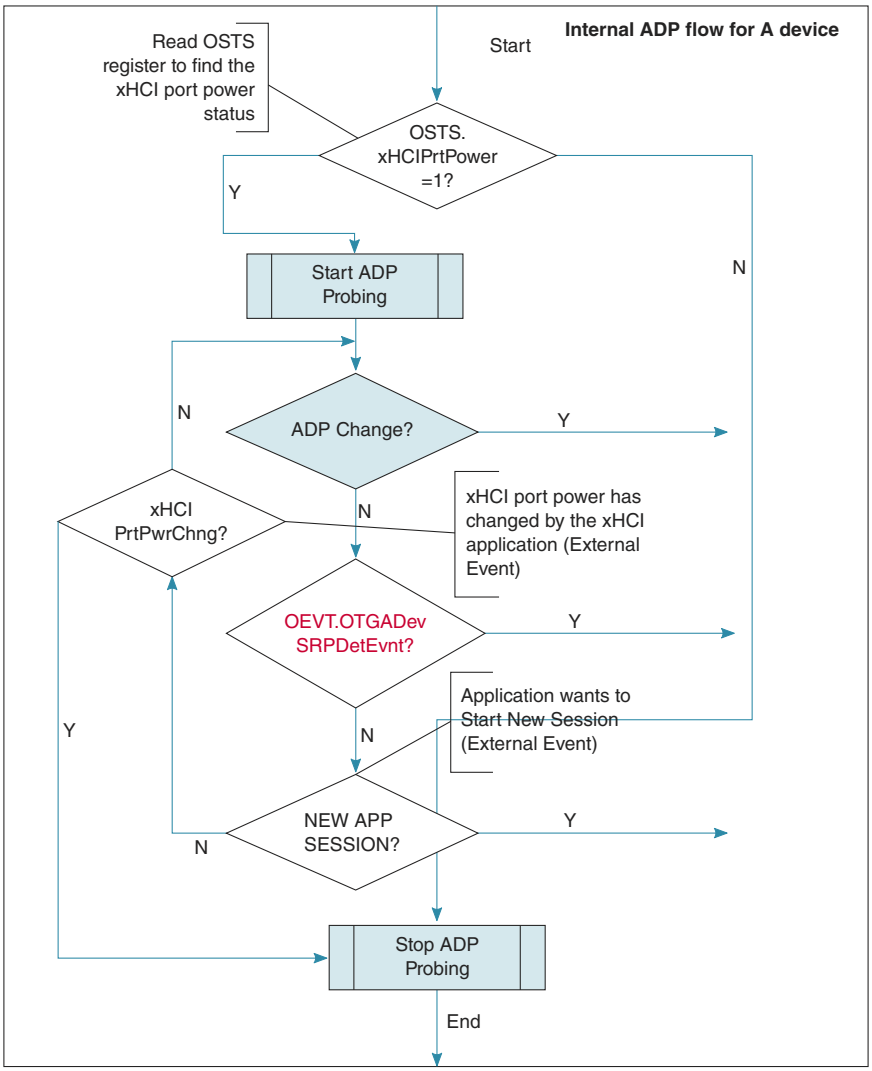


Figure 11-29. Internal ADP flow for A-device

- ADEVSESSENDDETEVNT occurs due to:
  - PORTSC[PRTPOWER] = 1'b0
  - OTG driver stopping VBUS due to:
    - BCON_TOUT when in A_WAIT_BCON
    - A_AIDL_BDIS_TOUT when in A_SUSPEND
- At any step, if ADEVSESSENDDETEVNT occurs, the software starts with ADP probing.
- Once the ADP probing (both external and internal) is started, the software should wait for one of the following events to occur (to stop ADP probing and proceed to Step 2 in A-device flow) to continue.
  - ADP change event
  - SRP detect event
  - New session event from A-device application
  - xHCI port power change event from A-device application

11.1.2.6.1.7 SRP detection by the core (Timeline for ADevSRPDetEvnt)

ADevSRPDetEvnt is triggered by a valid DLINE pulsing on the USB D+ line by B-Device. The A-Device looks at both the positive and negative edge transition occurrence for a valid data-line pulse time before detecting SRP.

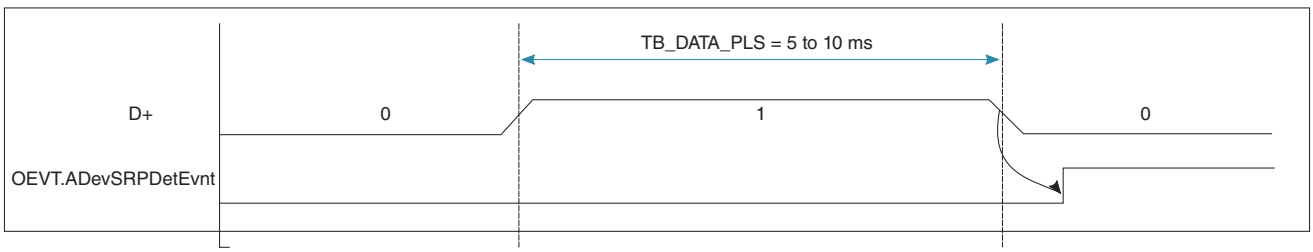


Figure 11-30. Timeline for ADevSRPDetEvnt

11.1.2.6.1.8 VBUS turned ON by the core (Timeline for ADevBSessEndEvnt)

When the A-device turns on the VBUS, it is unsure at this point of time at what speed the USB enumerates to. So, both the LTSSM and the OTG 2.0 FSMs start concurrently. The LTSSM proceeds with the training sequence after Rx_detect and the OTG 2.0 FSM waits for A_WAIT_BCON.

SS Mode: Before entering a3_ds_host, the training sequence is started. If the training sequence fails, the core automatically falls back to operate in HS/FS mode.

HS/FS Mode: Before entering A-Host, there is a check done in the OTG state machine to ensure B-Device is connected. If B-Device is not connected within TA_WAIT_BCON, ADevBSessEndDetEvnt is set. The core resets OCTL.PrtPwrCtl, and the OTG state machine enters A_WAIT_VFALL. Then VBUS to B-Device is removed and this results in ADevSessEndEvnt. Note that the A_WAIT_VFALL transition can occur due to over current, which is not depicted in the following figure.

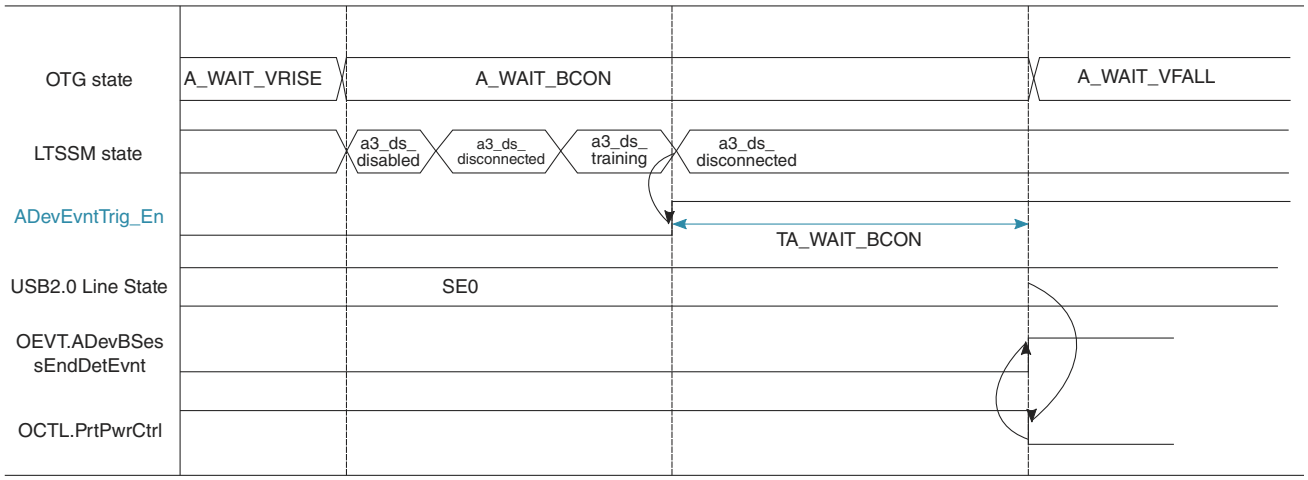


Figure 11-31. Timeline for ADevBSessEndEvnt

11.1.2.6.1.9 Core entering a3_ds_host in SS mode or A-Host in HS/FS mode (Timeline for ADevBHostEvnt)

The core gives the first chance to connect in SS mode. If SS connection times out (2 ms of polling timeout in LTSSM), the core enables the OTG 2.0 FSM to detect any connect on D+ line (TA_WAIT_BCON).

SS Mode: ADevHostEvnt is not set while operating in SS mode.

HS/FS Mode: If the B-Device is connected within TA_WAIT_BCON, ADevBHostEvnt is set.

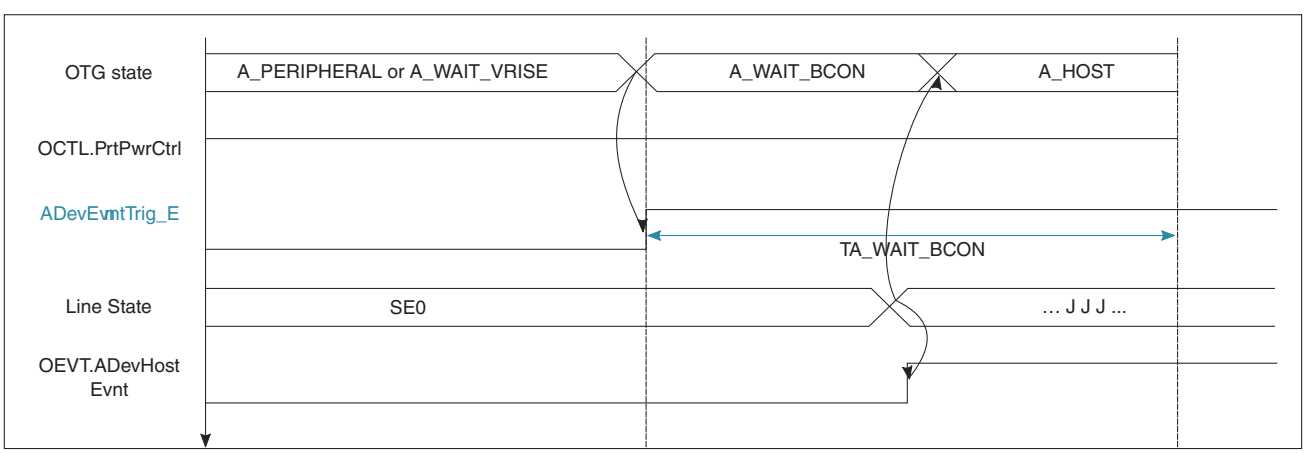


Figure 11-32. Timeline for ADevBHostEvnt

11.1.2.6.1.10 B-device flow

The following figure shows the B-device flow.

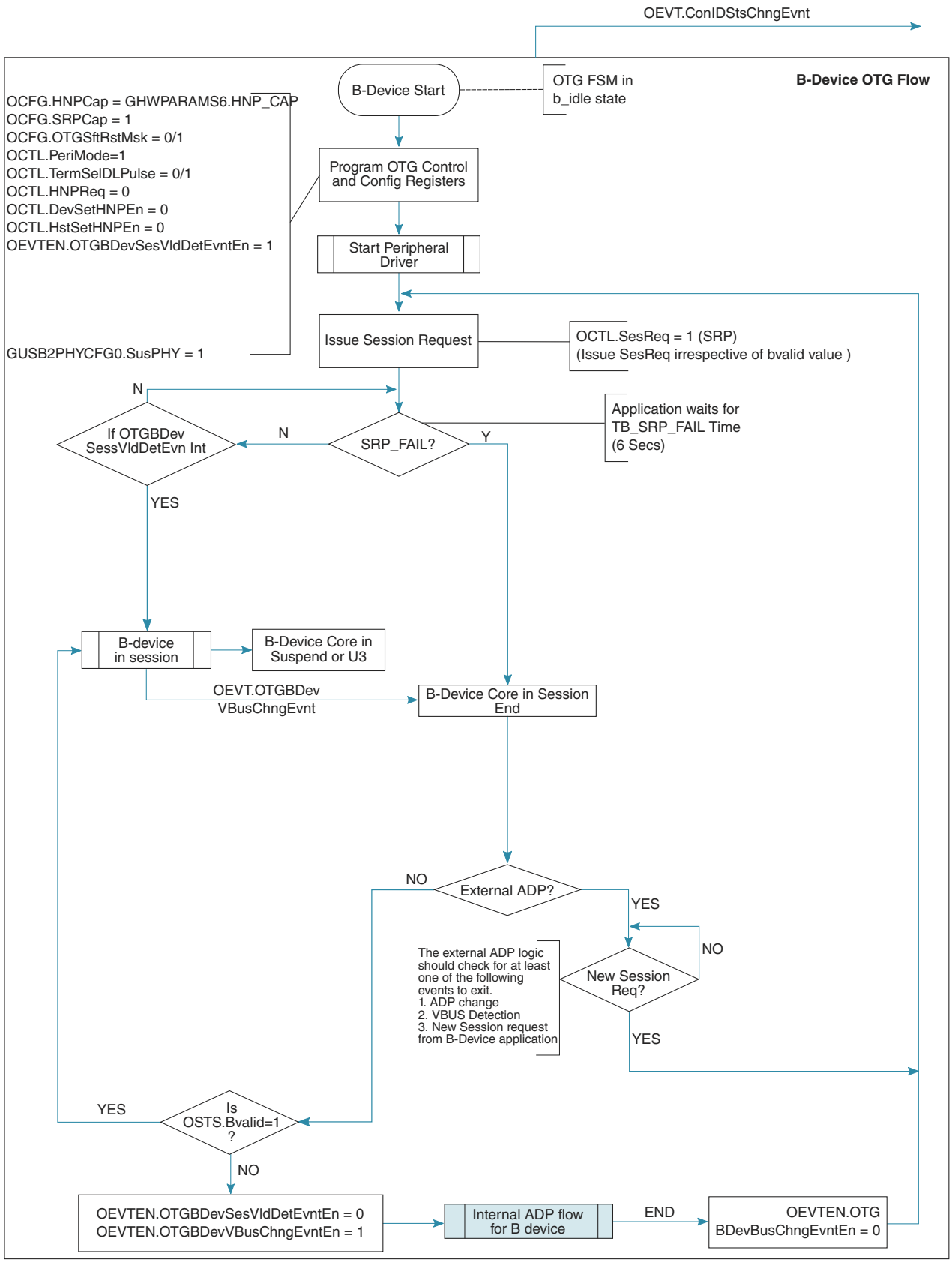


Figure 11-33. B-device flow diagram

1. At start, the core is in B-IDLE state. Note that for OTG functionality in peripheral mode, DCTL[RUN_STOP] should be set, otherwise the core as B-peripheral does not initiate SRP pulsing. Also, it is important for the device driver to program the speed of operation as B-peripheral (DCFG[SPD]).
2. The OTG software should program the following OTG control registers for B-device operation:
   - OCFG[HNPCAP] = GHWPARAMS6[HNPSUPPORT]
   - OCFG[SRPCAP] = 1
   - OCTL[PERIMODE] = 1
   - OCTL[TERMSELDLPULSE] = 0
   - OCTL[HNPREQ] = 0
   - OCTL[DEVHNPEN] = 0
   - OCTL[HSTSETHNPEN] = 0
   - OEVTEN[OTGBDEVSESVLDEVNTEN] = 1
   - GUSB2PHYCFG0[SUSPHY] = 1
   - GCTL[GBLHIBERNATIONEN] = 0 to prevent hibernation entry so that the OTG software can perform SRP.
3. Start peripheral driver.
4. Program OCTL[SESREQ] = 1 to initiate SRP by D-line pulsing. The core initiates SRP and waits for a valid VBUS (BVALID).
5. The B-device core continues to wait for BVALID if the A-device does not respond. The application should timeout if SESSVLDDETEVNT does not come within the SRP fail time (TB_SRP_FAIL is 6 seconds).

   If hibernation feature is enabled, the OTG Software can program the GCTL[GBLHIBERNATIONEN] = 1 to enable hibernation entry.

6. When SRP fails, the application should enable ADP probing.

   If the core had entered Hibernation after SRP Failure, the OTG Software should direct the power management software to exit hibernation if the OTG software decides to perform internal ADP. On hibernation exit, the OTG software should program the GCTL[GBLHIBERNATIONEN] = 0 before starting the internal ADP Flow.

7. If the A-device asserts a valid VBUS in response to SRP, the core reports an BDEVSESSVLDDETEVNT to indicate the start of a session.
8. If there is a BDEVSESSVLDDETEVNT, the core enters a session on state. For HNP flows in B-peripheral -> B-host -> B-peripheral flow diagram.
9. If there is a BDEVVBUSCHNGEVNT anytime, indicating VBUS is no longer valid, the core enters session-end state.

If hibernation feature is enabled, the OTG Software can program the GCTL[GBLHIBERNATIONEN] = 1 to enable hibernation entry.

NOTE

- For OTG 2.0, during HNP process where the B-device is going to assume the role of a host, the B-device application needs to ensure that a USB reset process is programmed within 150 ms (TB_ACON_BSE0) of getting a device connect interrupt.
- At any step, if CONIDSTSCHNGEVNT occurs, the A-device flow is terminated as per "A-device Flow Diagram".

The OTG state machine mapping to the software flow is as follows:

- Steps 1-3 and 6 are B-IDLE
- Step 4 is B-SRP-INIT
- Step 7 and 8 are B-PERIPHERAL

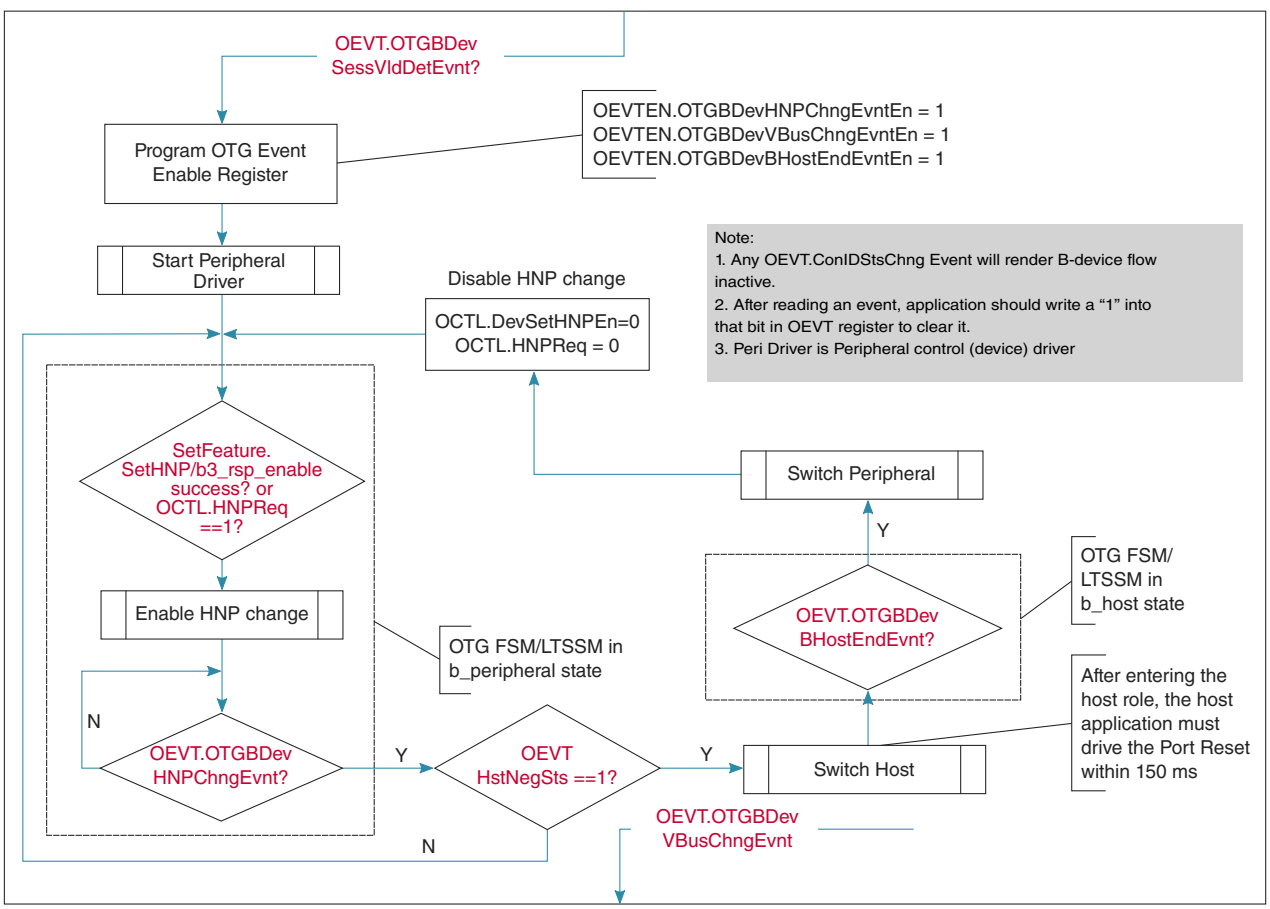


Figure 11-34. B-peripheral -> B-host -> B-peripheral flow diagram

For HS/FS mode:

1. The OTG state machine starts operation as a B-peripheral. The following events are enabled.
   - OEVTEN[OTGBDEVHNPCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVVBUSCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVBHOSTENDEVNTEN] = 1

   In this state, the application periodically exchanges SetFeature.SetHNPEnable packets. If the B-device wants to become host, this packet exchange results in a device application sending an ACK to this packet, and it directs the OTG driver to execute Enable HNP change. Ensure that GUSB2PHYCFG0[SUSPHY] is 1'b0 when SetFeature.SetHNP command succeeds. This enables the core to initiate HNP from B-device when a suspend is initiated. During a suspend, BDEVHNPCHNGEVNT is set. OTGBDEVHNPCHNGEVNT starts the transition from B-peripheral ->B-WAIT-ACON to either B-host or B-peripheral.
2. If BDEVHNPCHNGEVNT is set, the OTG driver reads OEVT[HSTNEGSTS]. If OEVT[HSTNEGSTS] is set, it indicates successful connection of A-device as peripheral, the core therefore transitions to B-host state. After entering the host role, the host application must drive the port reset within 150 ms. If OEVT[HSTNEGSTS] is not set, it indicates failure of HNP and it stays as B-peripheral.
3. When in B-host state, if there is BDEVBHOSTENDEVNT, it signifies that the B-device no longer wants to be the host, therefore suspended the bus and A-device has stopped signaling connect. Therefore, the driver goes back to step 1 and assumes B-peripheral role.
4. The core generates BDEVBHOSTENDEVNT.

NOTE

- Step 3 above is B-host and B-host to B-peripheral.
- OTGBDEVVBUSCHNGEVNT indicates transition back to B-IDLE.

For SS mode:

1. The OTG state machine starts operation as b3_us_peripheral. The following events are enabled.
   - OEVTEN[OTGBDEVHNPCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVVBUSCHNGEVNTEN] = 1
   - OEVTEN[OTGBDEVBHOSTENDEVNTEN] = 1

   a. The A-device (host) should send a SET_FEATURE with NTF_HOST_REL feature selector. This SET_FEATURE needs to be sent by the A-device within TRSP_ACK. The core receives this information and is passed onto PCD. If this SET_FEATURE is not received within TRSP_ACK_ERR the OTG driver needs to program OCTL[OTG3_GOERR] bit for instructing the core's LTSSM to go to

b3_us_error state. The TRSP_ACK_ERR timer needs to be maintained in software.

b. As a response, the PCD programs the device generic command register (command value = 07h). For programming the device generic command register, refer to "Device Programming Model". The PCD should program the core to send this device notification packet within TRSP_CNF.
c. The PCD programs OCTL[DEVSETHNPEN] = 1 as a result of the above step.
d. The core is responsible for internally detecting warm reset and generating OTGBDEVHNPCHNGEVNT. If the warm reset is not received within TRSP_WRST, the core doesn't generate OTGBDEVHNPCHNGEVNT and hence the PCD should program OCTL[OTG3_GOERR] which will direct the core to b3_us_error state.
e. As a result of above step and if OTGBDEVHNPEVNT is generated, the OTG driver informs the PCD and in turn, the PCD completes its activities, programs the core to stop the activities as a peripheral and hands over the control to OTG driver. The OTG driver programs OCTL[PERIMODE] as 0. Ensure that GCTL[DSBLCLKGTNG] is 1 before role change.

2. If BDEVHNPCHNGEVNT is set, it indicates that the core has transitioned to b3_ds_disconnected state. The OTG driver now hands over the control to xHCI driver. Note that after xHCI driver has taken control, if there is a transition to b3_ds_error state by the A-device, this error handling is performed by the xHCI driver.
3. When in b3_ds_host_state
4. The core generates BDEVBHOSTENDEVNT.

NOTE

- Step 3 above is B-host and B-host to B-peripheral.
- OTGBDEVVBUSCHNGEVNT indicates transition back to B-IDLE.
- If there is any unexpected error scenario like VBUS drop after the OCTL[DEVSETHNPEN] bit is set, the OTG driver must clear the OCTL[DEVSETHNPEN] bit.

Internal ADP flow for B-device

The internal ADP flow of B-device is as follows:

Figure 11-35. Internal ADP flow for B-device

While probing, the application should wait for one of the following events to exit probing and start SRP request again.

- ADP change
- VBus detection
- New session request from B-device application

While sensing, the application should wait for one of the following events to exit sensing.

- ADP sensed failed
- VBUS detection
- New session request from B-device application

The application should continue with ADP probing if ADP sense failed. Else, the application should exit sensing and start SRP request again

11.1.2.6.1.11 Core entering b3_us_peripheral in SS mode or B-Peripheral in HS/FS mode (Timeline for BDevSessVldDetEvnt)

The PCD decides the speed in which the USB3 core needs to operate by programming DCFG.DevSpd bits. Accordingly, the core decides to operate either in SS or FS to start with.

SS Mode: OEVT.BDevSessVldDetEvnt is triggered when the LTSSM enters b3_us_disconnected state after detecting VBUS turned on by the A-device

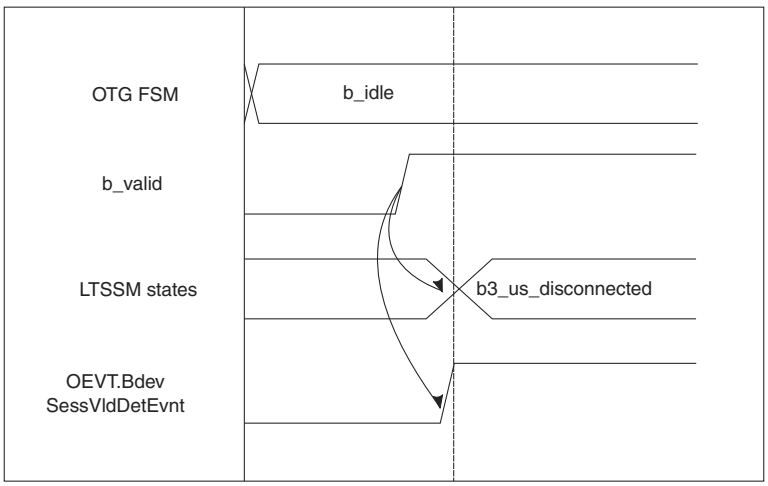


Figure 11-36. Timeline for BDevSessVldDetEvnt in SS mode

HS/FS Mode: BDevSessVldDetEvnt is triggered by a valid VBUS detection by B-Device in B-IDLE state. The B-Device will transition to B-PERIPHERAL state before triggering this event. Note that this is common to both SS and HS(FS).

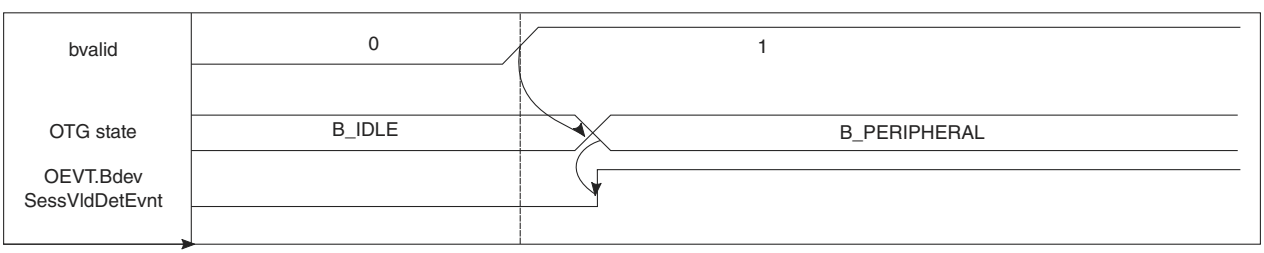


Figure 11-37. Timeline for BDevSessVldDetEvnt in HS/FS mode

11.1.2.6.1.12 VBUS change detected on USB (Timeline for BDevVBusChngEvnt)

BDevVBusChngEvnt is triggered when a change in bvalid is detected irrespective of OTG state machine.

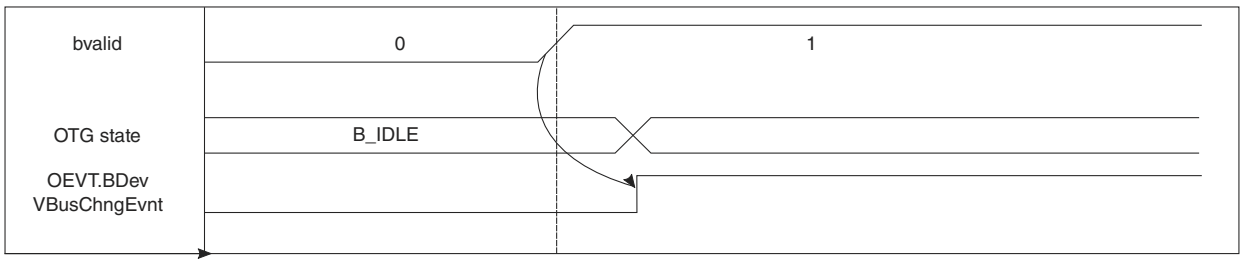


Figure 11-38. Timeline for BDevVBusChngEvnt

11.1.2.6.1.13 Internal ADP controller logic

The ADP controller has timers that assist in ADP operation. The ADP-related registers are a part of the pwrm_otgif module. For the PHY to generate ADPPRB and ADPSNS, the PHY needs to know the value to compare with VADP_PRB and VADP_SNS, respectively. These values can be hardcoded in the PHY or the PHY can have limited programmable option by which these reference voltage values can be changed. The "Wakeup Logic" module implements SRP detection in A-Device mode. This is required if the USB 3.0 controller is powered down completely while ADP is still in progress and B-Device initiates SRP.

NOTE

ADP controller logic is inferred along with, but outside USB 3.0 controller.

- As a product, both ADP controller logic and OTG controller are packaged into USB 3.0 core.
- All ADP timers are maintained in the ADP controller module, which generates the enable and disable signaling to the PHY for ADP probing and sensing.
- PHY contains the following circuitry related to ADP functionality:
  - Comparators for PRB and SNS
  - I_ADP_SRC and I_ADP_SNK
  - V_{BUS} circuitry
- Application software programs the ADP timing registers that resides in the pwrm_otgif module.
- The pwrm_otgif module provides a mechanism to the application via interrupts to log and report events pertaining to ADP probing and sensing.
- In this operation, only the pwrm module needs to be always powered on. The pwrdwn module can be either powered on or off.

11.1.2.6.1.14 Initializing ADP Controller

Initializing the ADP controller comprises of the following steps:

1. Set ADPEVTEN.ADPRstCmpltEvntEn to 1'b1.
2. Set ADPCTL.ADPRes to 1'b1. This initiates reset of the ADP controller.
3. Wait for ADPEVT.ADPRstCmpltEvnt to ensure that the reset process is complete.
4. Program the appropriate ADPCFG.PrbDschg (Discharge time), ADPCFG.PrbDelta (Resolution unit), and ADPCFG.PrbPer (Probe period) values.
5. Read the ADPCTL register to see if the ADPCFG values have been written into the ADP controller (Suspend clock domain) by looking at the ADPCTL.WB bit. Wait for the ADPCTL.WB bit to be cleared by the hardware to confirm that the write operation is complete.
6. Clear ADPEVTEN.ADPRstCmpltEvntEn to 1'b0 in order to complete the ADP initialization sequence.

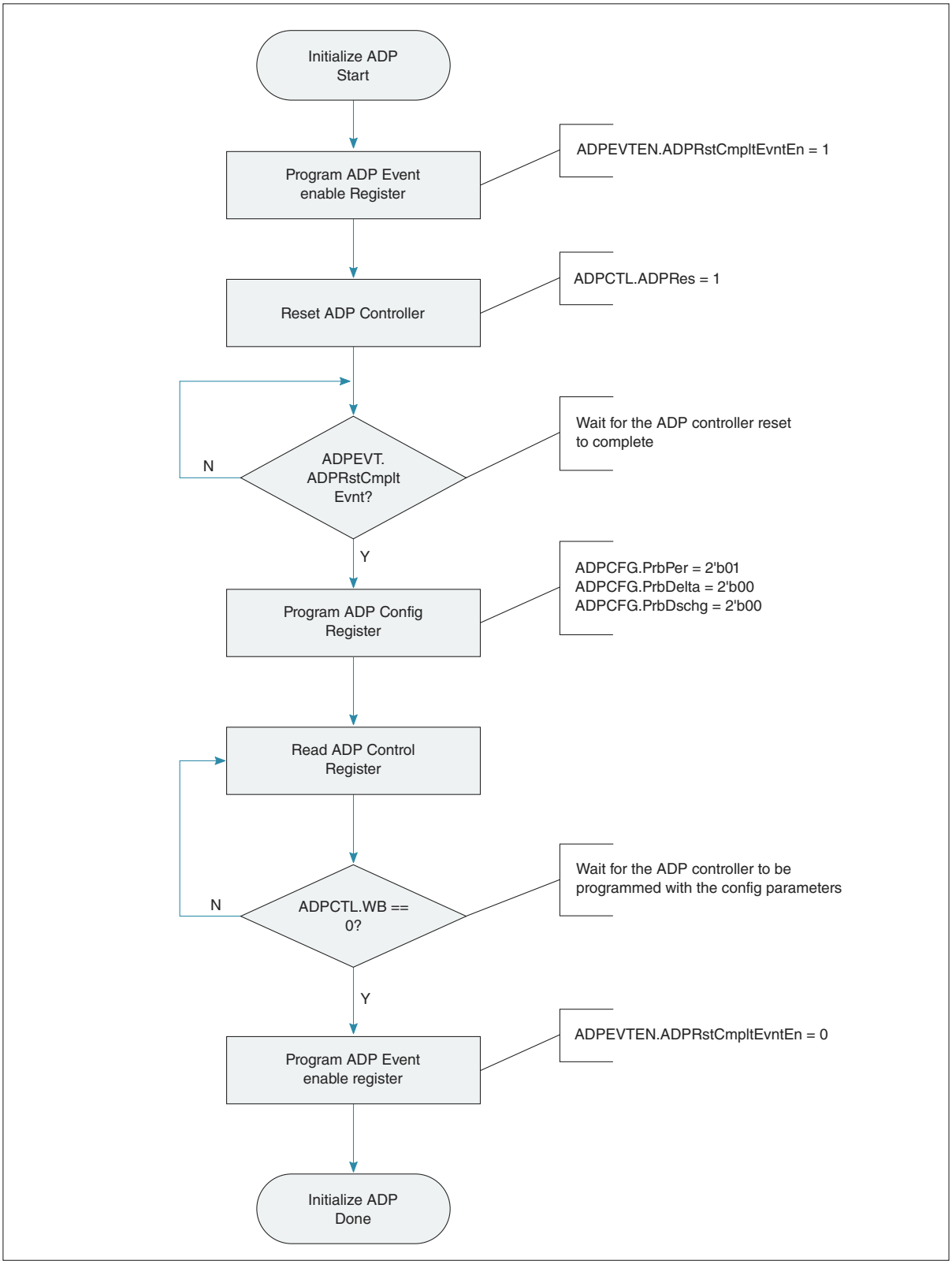


Figure 11-39. Flow Diagram for Initializing ADP

11.1.2.6.1.15 ADP sensing

ADP sensing is allowed only when operating as a B-device and not allowed when operating as an A-device. ADP sensing is done with the following three steps:

1. Start ADP sense
2. Detect ADP sense
3. Stop ADP sense

Start ADP sense

To start ADP sensing, perform the following steps:

1. Set ADPEVTEN[ADPSNSEVNTEN] to 1'b1.
2. Set ADPCTL[ADPEN] and ADPCTL[ENASNS] to 1'b1.
3. Read the ADPCTL register to see if the ADPCTL values have been written into the ADP controller (Suspend clock domain) by looking at the ADPCTL[WB] bit. Wait for the ADPCTL[WB] bit to be cleared by the hardware to confirm that the write operation is complete. Now the core starts an ADP sense to the PHY.

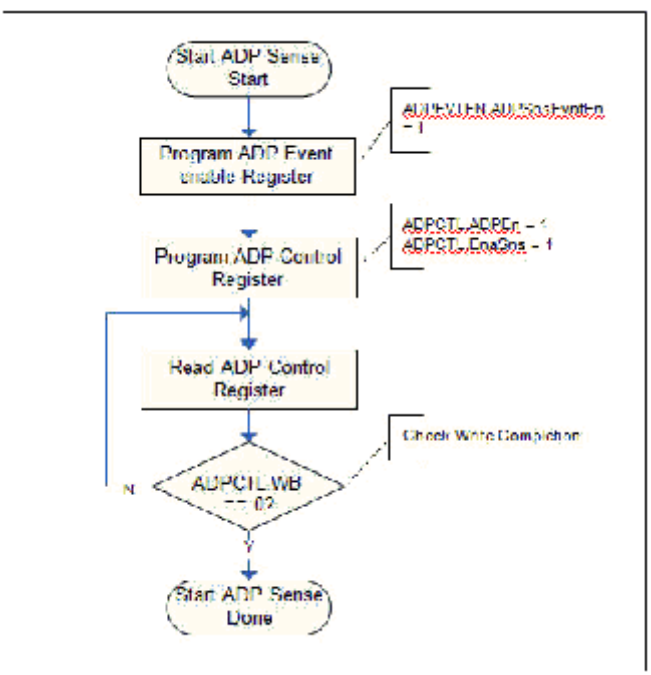


Figure 11-40. Start ADP sense

Detect ADP sense

To detect ADP sensing, perform the following steps:

1. Wait for ADPEVTEN[ADPSNSEVNT] to conclude that A-host is performing ADP probing. After starting the ADP sense, ADPEVTEN[ADPSNSEVNT] should occur at every probe period of A-host.

2. If ADPEVTEN[ADPSNSEVNT] is not detected within TB_ADP_DETACH time (3 seconds), conclude that the ADP sense has failed.

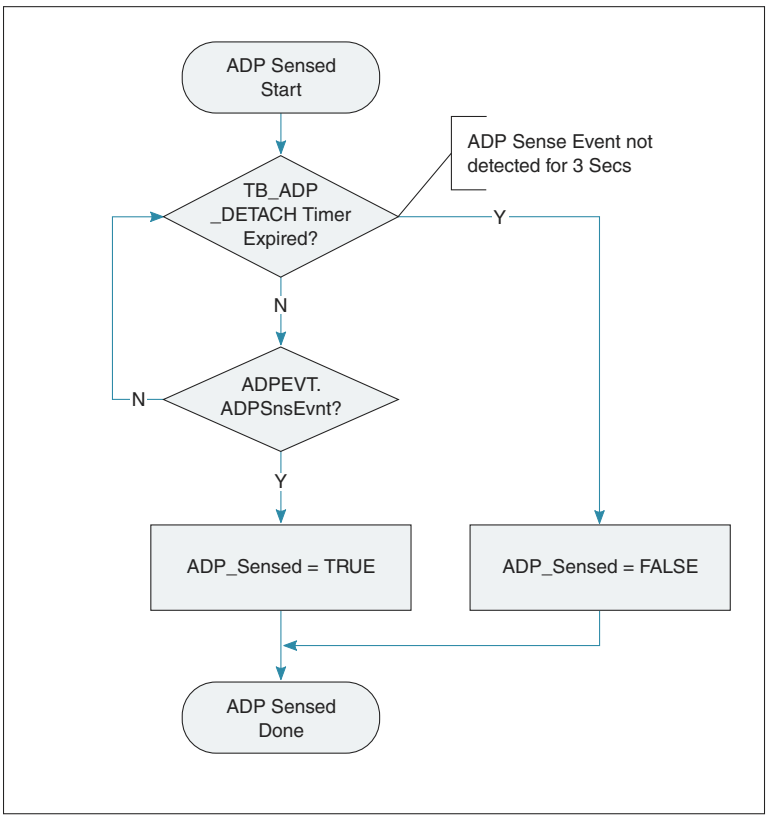


Figure 11-41. Flow diagram to detect ADP sense

Stop ADP sense

To stop ADP sensing, perform the following steps:

1. Clear ADPEVTEN[ADPSNSEVNTEN] to 1'b0.
2. CLEARADPCTL[ADPEN] and ADPCTL[ENASNS] to 1'b0.
3. Read the ADPCTL register to see if the ADPCTL values have been written into the ADP controller (Suspend clock domain) by looking at the ADPCTL[WB] bit. Wait for the hardware to clear the ADPCTL[WB] bit to confirm that the write operation is complete. Now the core will stop ADP sense to PHY.

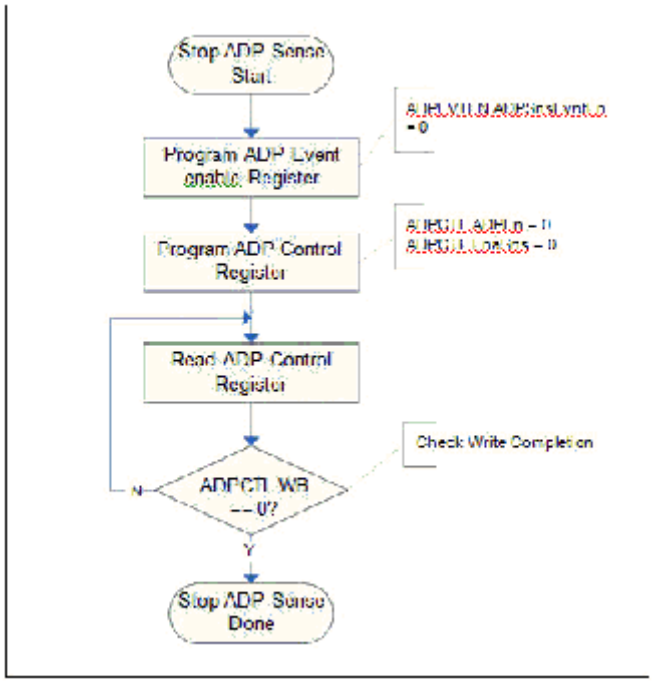


Figure 11-42. Stop ADP sense

11.1.2.6.1.16 ADP probing

ADP probing is allowed in both A-device and B-device. ADP probing is handled with the following three steps:

1. Start ADP probe
2. Detect ADP change
3. Stop ADP probe

Start ADP probe

To start ADP probing, perform the following steps:

1. Set ADPEVTEN[ADPPRBEVNTEN] and ADPEVTEN[ADPTMOUTEVNTEN] to 1'b1.
2. Set ADPCTL[ADPEN] and ADPCTL[ENAPRB] to 1'b1.
3. Read the ADPCTL register to see if the ADPCTL values have been written into the ADP controller (suspend clock domain) by looking at the ADPCTL[WB] bit. Wait for the hardware to clear the ADPCTL[WB] bit to confirm that the write operation is complete. Now the core will start ADP probing to PHY.

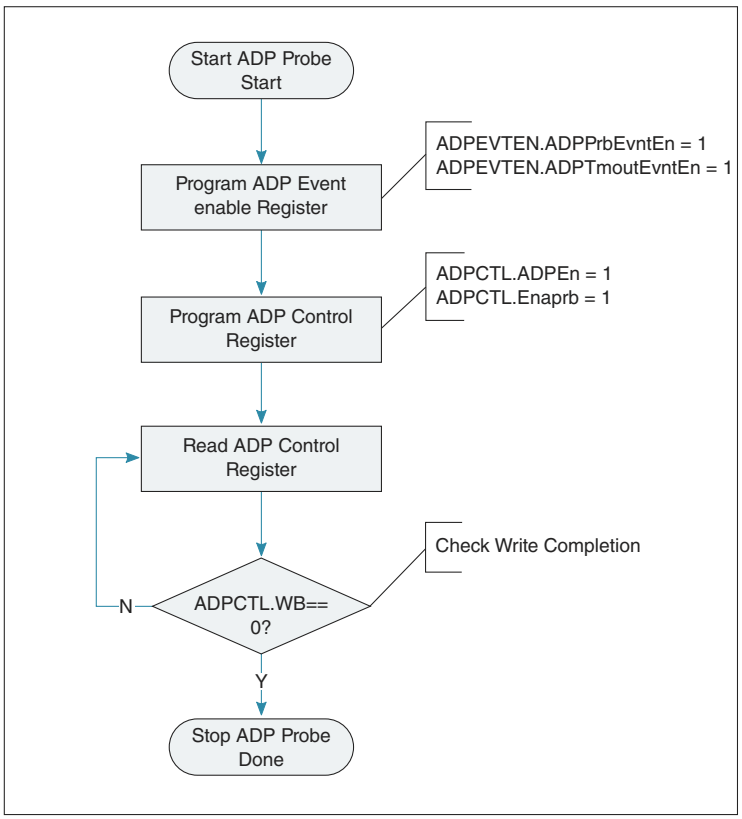


Figure 11-43. Flow diagram for start ADP probe

Detect ADP change

To detect ADP change, perform the following steps:

1. Wait for ADPEVT[ADPPRBEVNT].
2. Read the Ramp Time (RTIM) value and compare it with the previous read value. If the difference is greater than the minimum ramp value of attach/detach (including the noise delta), then report ADP change = TRUE, else report ADP change = FALSE.

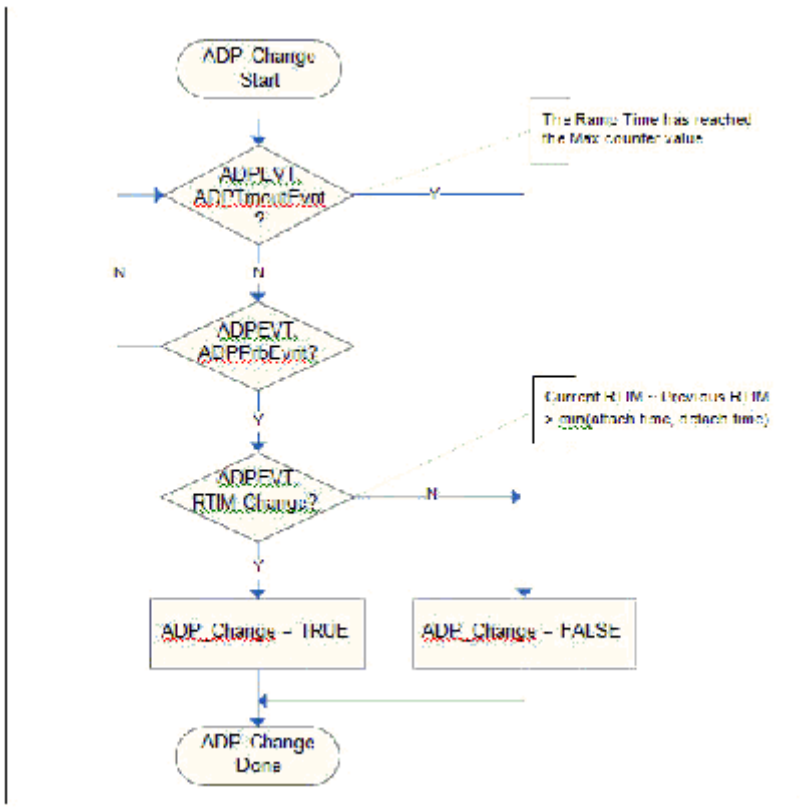


Figure 11-44. Detect ADP change

Table 11-23. Cause of interrupt

Status bit values	Explanation
ADP_PRB_INT=0, ADP_TMOUT_INT=0	Not applicable
ADP_PRB_INT=0, ADP_TMOUT_INT=1	During an ADP probe, the ADPEVT[RTIM] has reached the terminal value (while trying to charge VBUS) and the VADP PRB value has not been reached. The application software must note down the ADPEVT[RTIM] value for this charging cycle.
ADP_PRB_INT=1, ADP_TMOUT_INT=0	VADP PRB voltage has been reached on VBUS. The application software can note down the intermediate ADPEVT[RTIM] value. The timer value indicates the time taken for the ramp to reach VADP PRB voltage.
ADP_PRB_INT=1, ADP_TIMOUT_INT=1	The ramp time has reached the terminal value (0x7FF) and VADP PRB voltage is also reached. The application must note down the RTIM value.

Stop ADP probe

To stop ADP probing, perform the following steps:

1. Clear ADPEVTEN[ADPPRBEVNTEN] and ADPEVTEN[ADPTMOUTEVNTEN] to 1'b0.
2. Clear ADPCTL[ADPEN] and ADPCTL[ENAPRB] to 1'b0.

3. Read the ADPCTL register to see if the ADPCTL values have been written into the ADP controller (Suspend clock domain) by looking at the ADPCTL[WB] bit. Wait for the hardware to clear the ADPCTL[WB] bit to confirm that the write operation is complete. Now the core stops ADP probing to PHY.

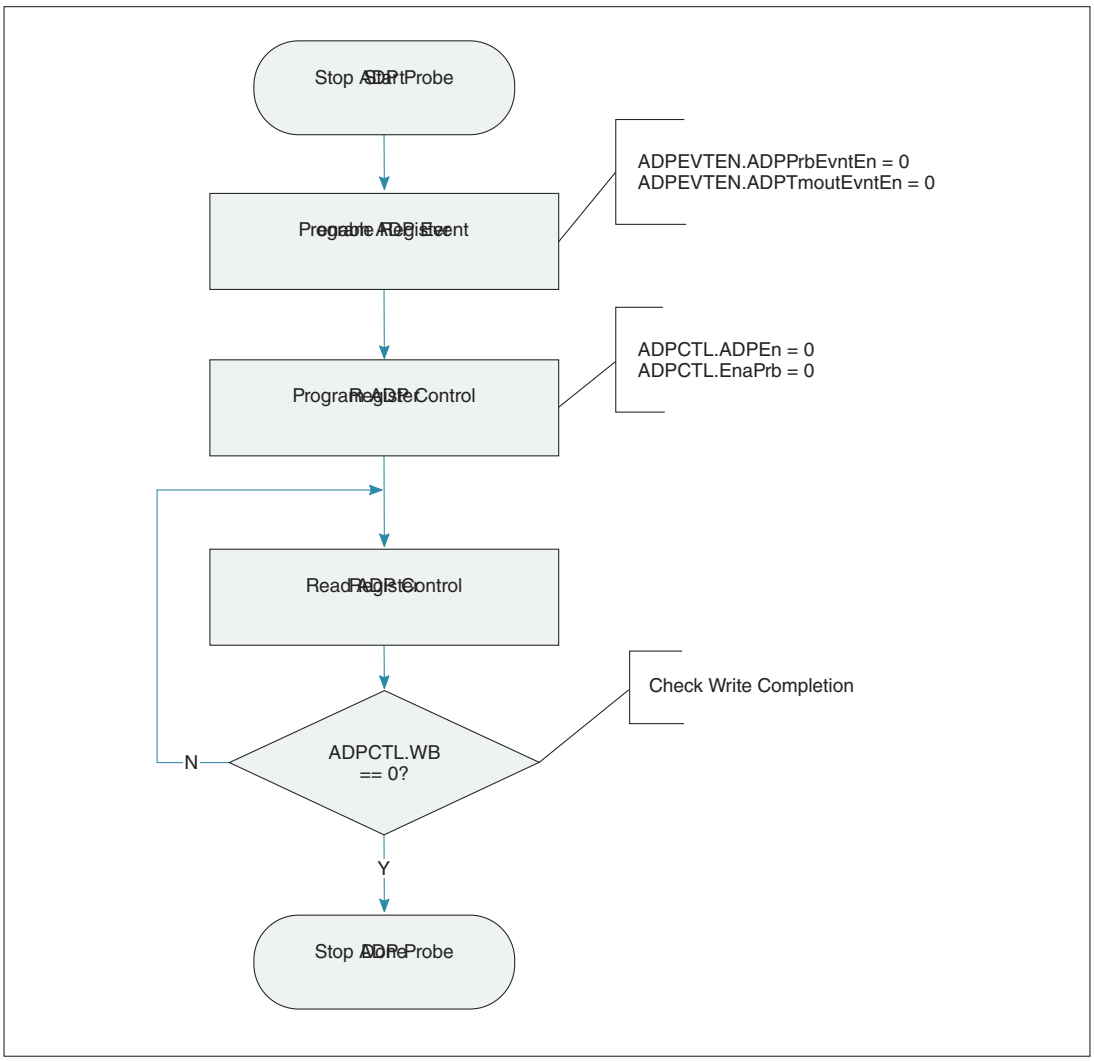


Figure 11-45. Flow diagram to stop ADP probe

11.1.2.7 Initialization/application information

The PHY can be configured for fixed equalization by programming relevant control registers in the USB 3.0 PHY.

In order to initialize the USB 3.0 PHY, software should perform the following steps:

1. Write 1'b0 to RX_OVRD_IN_HI.RX_EQ_EN [address 16'h1006: bit 6].

2. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_EN_OVRD [address 16'h1006: bit 7].
3. Write a fixed value to RX_OVRD_IN_HI.RX_EQ [address 16'h1006: bits 10–8] [equalization setting] (generally 2–4, based on testing in customer environment).
4. Write 1'b1 to RX_OVRD_IN_HI.RX_EQ_OVRD [address 16'h1006: bit 11].

11.1.2.8 Power management overview

The following power management features are available in the USB 3.0 core:

- Hibernation (save/restore) in device (L1, L2, U3, Disconnected) and host (U3, Disconnected)
- Hardware-controlled LPM in USB 2.0 host
- Clock gating in device (U1/U2/U3) and host (U1/U2/U3)

When hibernation is supported, the core saves some of its internal state to an external location and allows the USB3 core to be powered off. Small modules, namely the PMUs, detect wakeup conditions and request power be reapplied to the core, at which time the external state is brought back into the core.

Since the PMU modules are not inside the USB 3.0 core, the PMUs can easily be placed anywhere in the chip during floor-planning, adding isolation cell/voltage shifter cells. The PMU modules can be placed in the always ON power domain which could be operating at a different voltage than the USB 3.0 power domain.

Hibernation is described in the xHCI Specification as an optional feature that can be supported.

In addition, this feature is also added to the device controller.

NOTE

- The PMU does not support the following features in Hibernation mode:
  - ADP
  - BC
  - SRP signalling
- It is required that the SuperSpeed PHY must support receiver detection in P3 PowerDown mode.
- Hibernation is not supported in OTG mode when the roles are reversed.

- In the host mode, the USB 3.0 core does not support clock gating and hibernation when the Debug Capability is enabled during runtime (DCCTRL.DCE=1).
- During the normal host mode of operation, the USB 3.0 core supports clock gating and hibernation features when the Debug Capability is not enabled by the software (DCCTRL.DCE=0).

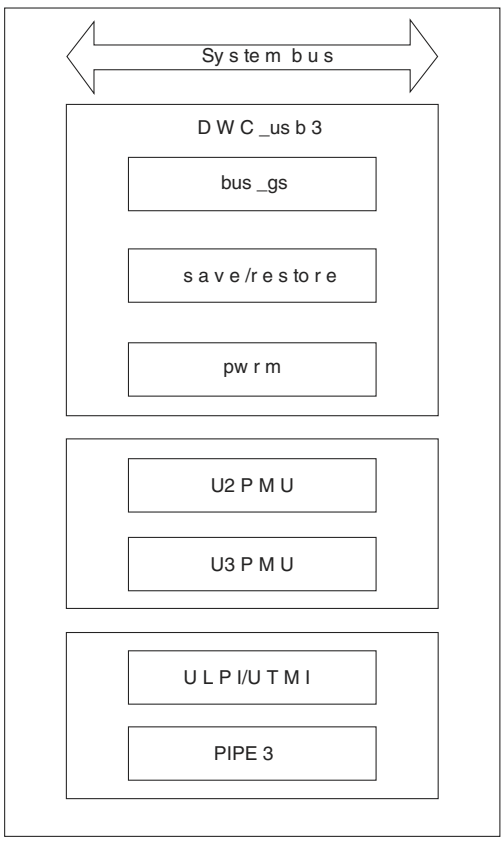


Figure 11-46. Power management hibernation architecture

The second power saving mechanism is hardware-controlled LPM. This is a USB 2.0 host-only feature where the host controller automatically detects the downstream port is idle for a certain amount of time and autonomously initiates an LPM token to the connected device. If the device accepts, the link is put into the standard USB 2.0 Suspend state.

Finally, clock gating is also supported in the host controller as well as in the device controller. The following table gives the list of power saving mechanisms available in USB 3.0 core.

Table 11-24. Power savings support

USB 3.0 State	USB 2.0 State	Device Power Savings	Host Power Savings
U1 (Hardware Initiated)	None	Core: Clock Gating PHY: P1	Core: Clock Gating PHY: P1
U2 (Hardware Initiated)	LPM-L1 (Hardware Initiated)	Core: Clock Gating, LPM-L1 Hibernation PHY: P2/Sleep	Core: Clock Gating, PHY: P2/ Sleep
U3 (Software Initiated)	Suspend (Software Initiated)	Core: Clock Gating, Hibernation PHY: P3/Suspend	Core: Clock Gating, Hibernation PHY: P3/Suspend
Rx.Detect (Software Initiated)	None	None	Core: Clock Gating, Hibernation PHY: P3/Suspend
SS.Disabled (Software Initiated)	None	Core: Clock Gating, Hibernation PHY: P3/Suspend	None

11.1.2.8.1 Clock gating

The core implements clock gating in the following situations:

- In USB 2.0 device mode
  - When the UTMI suspend or l1_suspend signal is asserted and the core is idle
- In USB 3.0 device mode, when the link is in U1, U2 or U3 state and the core is idle.

In host mode, clock gating is enabled in the following situations:

- When all USB 2.0 ports are in the suspend or L1 Suspend state, and all USB 3.0 ports are in the U1/U2/U3 state, and the core is idle.

Internal clock gating will apply to:

- Modules that use the ram_clk
- Some modules that use bus_clk

Internal clock gating will not apply to:

- Modules that use mac3_clk: When the USB 3.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- Modules that use mac2_clk: When the USB 2.0 PHY is suspended, these modules switch to using suspend_clk, so their power consumption is reduced but the clock is not completely stopped.
- The bus_gs module, which uses bus_clk. This module detects wakeup on the slave interface and thus needs a non-gated clock.

Clock gating can be enabled or disabled using the GCTL register. The GCTL register is "sticky," it retains its value across soft resets and hibernation.

11.1.2.8.2 Hardware-Controlled LPM

In USB 2.0, the main link power management mechanism was for the host to suspend the link. However, suspending the link requires a long period of idle and was wasteful of power, especially because the host knows ahead of time whether it is going to schedule any traffic. It also required a long period of resume signaling.

The USB 2.0 Link Power Management Addendum [LPM-ECN] added a new token to the USB 2.0 specification, the LPM token. The host has the option of suspending the link quickly instead of waiting for a long period of idle, and communicates its desire to the device by sending an LPM token. The ECN also introduced a terminology for different link states. This new link state was called L1. The traditional USB 2.0 suspend was called L2. The L1 state requires a much shorter resume signaling period too.

An xHCI driver can initiate LPM by writing '2' to the PORTSC.PLS field. However, there is a lot of overhead for an xHCI driver to be constantly trying to keep the link in a low power state. Because of this, the xHCI specification also allows a compatible xHC to be enabled with "Hardware-Controlled LPM" by setting PORTPMSC.HLE as 1. When this happens, the xHC is expected to automatically send LPM tokens when it has no pending transfers and thinks that the link is idle enough to enter L1. This feature is supported by the host core.

11.1.2.8.2.1 Special consideration for OTG

When the core is configured as an OTG device, it will not initiate LPM when acting in a reverse role as a B- Host. This is because suspend on the link is treated as a request to switch back to its original role.

11.1.2.8.3 USB PHY power gating

The USB PHY power gating is used to save the leakage power of PHY if it is not required. Follow the steps given below to take PHY into power gating state:

1. Assert PHY_RESET by writing 1'b0 to the USB_PHY_CTRL[RST] bit.
2. Set the USB_PHY_CTRL[PDN_SSP] and USB_PHY_CTRL[PDN_HSP] to 2'b11.

NOTE

Both the above steps must have separate writes.

11.1.3 Memory Map and Register Definition

This section includes the memory map and detailed descriptions of all registers.

NOTE

Synopsys Proprietary. Used with permission.

11.1.3.1 USB3 register descriptions

11.1.3.1.1 DWC_usb3 Memory map

Base Address USB1-38100000 USB2-38200000

USB3 base address: 3810_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Capability registers length and Host Controller Operational Registers (CAPLENGTH)	32	RO	Table 11-24
4h	Structural Parameters 1 Register (HCSPARAMS1)	32	RO	Table 11-24
8h	Structural Parameters 2 Register (HCSPARAMS2)	32	RO	Table 11-24
Ch	Structural Parameters 3 Register (HCSPARAMS3)	32	RO	Table 11-24
10h	Capability Parameters 1 Register (HCCPARAMS1)	32	RO	0220_FE6Ch
14h	Doorbell Offset Register (DBOFF)	32	RO	Table 11-24
18h	Runtime Register Space Offset Register (RTSOFF)	32	RO	Table 11-24
1Ch	Host Controller Capability Parameters 2 (HCCPARAMS2)	32	RO	0000_002Fh
20h	USB Command Register (USBCMD)	32	RW	Table 11-24
24h	USB Status Register (USBSTS)	32	RW	Table 11-24
28h	Page Size Register (PAGESIZE)	32	RO	Table 11-24
34h	Device Notification Register (DNCTRL)	32	RW	0000_0000h
38h	CRCR_LO (CRCR_LO)	32	RW	Table 11-24
3Ch	(CRCR_HI)	32	RW	0000_0000h
50h	DCBAAP_LO (DCBAAP_LO)	32	RW	0000_0000h
54h	DCBAAP_HI (DCBAAP_HI)	32	RW	0000_0000h
58h	Configuration Register (CONFIG)	32	RW	Table 11-24
420h	Port Status and Control Register (PORTSC_20)	32	RW	Table 11-24
424h	USB3 Port Power Management Status and Control Register (PORT PMSC_20)	32	RW	Table 11-24
428h	(PORTLI_20)	32	RO	0000_0000h
42Ch	(PORTHLPMC_20)	32	RW	Table 11-24

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
430h	(PORTSC_30)	32	RW	Table 11-24
434h	USB3 Port Power Management Status and Control Register (PORTPMSC_30)	32	RW	Table 11-24
438h	Port Link Info Register (PORTLI_30)	32	RO	Table 11-24
43Ch	USB2 Port Hardware LPM Control Register (PORTHLPMC_30)	32	RO	0000_0000h
440h	Microframe Index Register (MFINDEX)	32	RO	Table 11-24
460h	Interrupter Management Register (IMAN)	32	RW	Table 11-24
464h	Interrupter Moderation Register (IMOD)	32	RW	0000_0FA0h
468h	ERSTSZ (ERSTSZ)	32	RW	Table 11-24
470h	ERSTBA_LO (ERSTBA_LO)	32	RW	0000_0000h
474h	ERSTBA_HI (ERSTBA_HI)	32	RW	0000_0000h
478h	ERDP_LO (ERDP_LO)	32	RW	0000_0000h
47Ch	ERDP_HI (ERDP_HI)	32	RW	0000_0000h
480h	Doorbell Register (DB)	32	RW	Table 11-24
880h	USBLEGSUP (USBLEGSUP)	32	RW	Table 11-24
884h	USBLEGCTLSTS (USBLEGCTLSTS)	32	RW	Table 11-24
890h	SUPTPRT2_DW0 (SUPTPRT2_DW0)	32	RO	0200_0402h
894h	SUPTPRT2_DW1 Register (SUPTPRT2_DW1)	32	RO	2042_5355h
898h	xHCI Supported Protocol Capability_ Data Word 2 (SUPTPRT2_DW2)	32	RO	Table 11-24
89Ch	SUPTPRT2_DW3 Register (SUPTPRT2_DW3)	32	RO	Table 11-24
8A0h	(SUPTPRT3_DW0)	32	RO	0300_0402h
8A4h	SUPTPRT3_DW1 Register (SUPTPRT3_DW1)	32	RO	2042_5355h
8A8h	SUPTPRT3_DW2 (SUPTPRT3_DW2)	32	RO	0000_0102h
8ACh	SUPTPRT3_DW3 (SUPTPRT3_DW3)	32	RO	Table 11-24
C100h	Global SoC Bus Configuration Register 0 (GSBUSCFG0)	32	RW	Table 11-24
C104h	Global SoC Bus Configuration Register 1 (GSBUSCFG1)	32	RW	Table 11-24
C108h	Global Tx Threshold Control Register (GTXTHRCFG)	32	RW	Table 11-24
C10Ch	Global Rx Threshold Control Register (GRXTHRCFG)	32	RW	Table 11-24
C110h	Global Core Control Register (GCTL)	32	RW	30C1_3004h
C118h	Global Status Register (GSTS)	32	RW	Table 11-24
C11Ch	(GUCTL1)	32	RW	Table 11-24
C128h	Global User ID Register (GUID)	32	RW	A000_0000h
C12Ch	Global User Control Register (GUCTL)	32	RW	Table 11-24
C130h	Gobal SoC Bus Error Address Register - Low (GBUSERRADDRLO)	32	RO	0000_0000h
C134h	Gobal SoC Bus Error Address Register - High (GBUSERRADDRHI)	32	RO	0000_0000h
C138h	Global SS Port to Bus Instance Mapping Register - Low (GPRTBIMAPLO)	32	RW	0000_0000h
C13Ch	Global SS Port to Bus Instance Mapping Register - High (GPRTBIMAPHI)	32	RW	Table 11-24
C140h	Global Hardware Parameters Register 0 (GHWPARAMS0)	32	RO	2020_400Ah

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C144h	Global Hardware Parameters Register 1 (GHWPARAMS1)	32	RO	8160_C93Bh
C148h	Global Hardware Parameters Register 2 (GHWPARAMS2)	32	RO	A000_0000h
C14Ch	Global Hardware Parameters Register 3 (GHWPARAMS3)	32	RO	0821_0085h
C150h	Global Hardware Parameters Register 4 (GHWPARAMS4)	32	RO	4782_2004h
C154h	Global Hardware Parameters Register 5 (GHWPARAMS5)	32	RO	0420_4108h
C158h	Global Hardware Parameters Register 6 (GHWPARAMS6)	32	RO	0A58_DC20h
C15Ch	Global Hardware Parameters Register 7 (GHWPARAMS7)	32	RO	0608_11ECh
C180h	Global High-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_HSLO)	32	RW	0000_0000h
C184h	Global High-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_HSHI)	32	RW	Table 11-24
C188h	Global Full-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_FSLO)	32	RW	0000_0000h
C18Ch	Global Full-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_FSHI)	32	RW	Table 11-24
C19Ch	Global User Control Register 2 (GUCTL2)	32	RW	Table 11-24
C200h	Global USB2 PHY Configuration Register (GUSB2PHYCFG)	32	RW	Table 11-24
C280h	Global USB 2.0 UTMI PHY vendor control register (GUSB2PHY ACC_ULPI)	32	RO	Table 11-24
C2C0h	Global USB 3.0 PIPE control register (GUSB3PIPECTL)	32	RW	010C_0002h
C300h - C31Ch	Global transmit FIFO size register (GTXFIFOSIZ0 - GTXFIFOSIZ7)	32	RW	0000_0042h
C380h - C388h	Global receive FIFO size register (GRXFIFOSIZ0 - GRXFIFOSIZ2)	32	RW	0000_0285h
C400h	Global Event Buffer Address (Low) Register (GEVNTADRLO)	32	RW	0000_0000h
C404h	Global Event Buffer Address (High) Register (GEVNTADRHI)	32	RW	0000_0000h
C408h	Global event buffer size register (GEVNTSIZ)	32	RW	Table 11-24
C40Ch	Global event buffer count register (GEVNTCOUNT)	32	RW	0000_0000h
C600h	Global Hardware Parameters Register 8 (GHWPARAMS8)	32	RO	0000_0A58h
C610h	Global Device TX FIFO DMA Priority Register (GTXFIFOPRIDEV)	32	RW	Table 11-24
C618h	Global Host TX FIFO DMA Priority Register (GTXFIFOPRIHST)	32	RW	Table 11-24
C61Ch	Global Host RX FIFO DMA Priority Register (GRXFIFOPRIHST)	32	RW	Table 11-24
C620h	Global Host Debug Capability DMA Priority Register (GFIFOPRIDBC)	32	RW	Table 11-24
C624h	(GDMAHLRATIO)	32	RW	Table 11-24
C630h	Global Frame Length Adjustment Register (GFLADJ)	32	RW	Table 11-24
C700h	Device Configuration Register (DCFG)	32	RW	Table 11-24
C704h	Device control register (DCTL)	32	RW	Table 11-24
C708h	Device Event Enable Register (DEVTEN)	32	RW	Table 11-24
C70Ch	Device Status Register (DSTS)	32	RW	Table 11-24
C710h	Device Generic Command Parameter Register (DGCMDPAR)	32	RW	0000_0000h
C714h	(DGCMD)	32	RW	Table 11-24
C720h	Device Active USB Endpoint Enable Register (DALEPENA)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C800h	Device physical endpoint-n command parameter 2 register (DEPCMDPAR2)	32	RW	0000_0000h
C804h	Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)	32	RW	0000_0000h
C808h	Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)	32	RW	0000_0000h
C80Ch	Device Physical Endpoint-n Command Register (DEPCMD)	32	RW	0000_0000h
CA00h	Device Interrupt Moderation Register (DEV_IMOD)	32	RW	0000_0000h
CC00h	OTG Configuration Register (OCFG)	32	RW	Table 11-24
CC04h	OTG Control Register (OCTL)	32	RW	Table 11-24
CC08h	OTG Events Register (OEVT)	32	RW	Table 11-24
CC0Ch	OTG Events Enable Register (OEVTEN)	32	RW	Table 11-24
CC10h	OTG Status Register (OSTS)	32	RO	Table 11-24
CC20h	ADP Configuration Register (ADPCFG)	32	RW	Table 11-24
CC24h	ADP Control Register (ADPCTL)	32	RW	Table 11-24
CC28h	ADP Event Register (ADPEVT)	32	RW	Table 11-24
CC2Ch	(ADPEVTEN)	32	RW	Table 11-24
CC30h	BCFG (BCFG)	32	RW	Table 11-24
CC38h	BCEVT (BCEVT)	32	RW	0000_0000h
CC3Ch	BCEVTEN (BCEVTEN)	32	RW	Table 11-24

11.1.3.1.2 Capability registers length and Host Controller Operational Registers (CAPLENGTH)

11.1.3.1.2.1 Offset

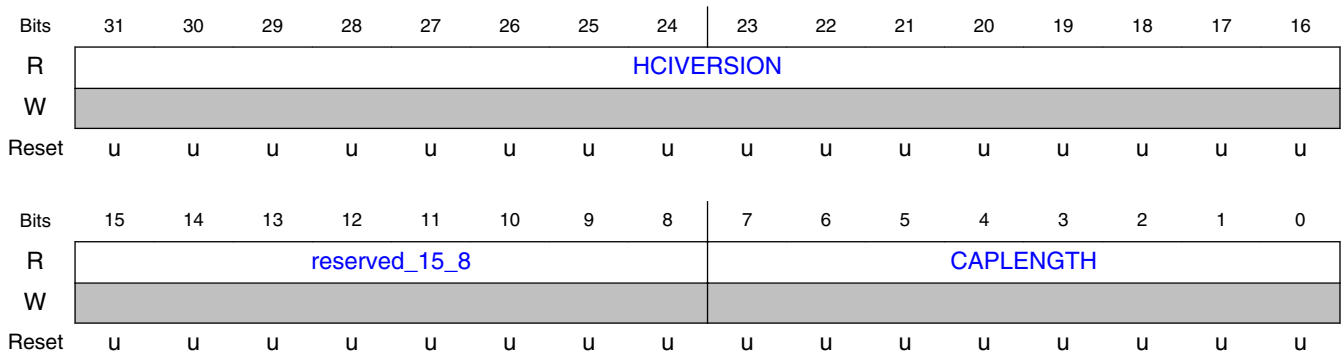
Register	Offset
CAPLENGTH	0h

11.1.3.1.2.2 Function

Capability Registers Length = Base address + CAPLENGTH where CAPLENGTH is `DWC_USB3_HOST_CAP_REG_LEN whose default value is 20h.

Reset Mask:0xFF00

11.1.3.1.2.3 Diagram



11.1.3.1.2.4 Fields

Field	Function
31-16 HCIVERSION	HC Interface Version Number (HCIVERSION)
15-8 reserved_15_8	Reserved
7-0 CAPLENGTH	Capability Registers Length (CAPLENGTH)

11.1.3.1.3 Structural Parameters 1 Register (HCSPARAMS1)

11.1.3.1.3.1 Offset

Register	Offset
HCSPARAMS1	4h

11.1.3.1.3.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0xF80000

11.1.3.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAXPORTS								reserved_23_19					MAXINTRS		
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAXINTRS								MAXSLOTS							
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.1.3.1.3.4 Fields

Field	Function
31-24 MAXPORTS	Number of Ports (MaxPorts) - Number of ports implemented is defined by the parameter (`DWC_USB3_HOST_NUM_U2_ROOT_PORTS + `DWC_USB3_HOST_NUM_U3_ROOT_PORTS) - Number of ports enabled is controlled by the core input signals host_num_u2_port[3:0]+host_num_u3_port[3:0] Note: In USB 2.0-only mode, the host_num_u3_port signal is zero.
23-19 reserved_23_19	Reserved
18-8 MAXINTRS	Number of Interrupters (MaxIntrs) Defined by the configurable parameter `DWC_USB3_HOST_NUM_INTERRUPTER_SUPT
7-0 MAXSLOTS	Number of device slots (MaxSlots) Defined by configurable parameter `DWC_USB3_NUM_DEVICE_SUPT

11.1.3.1.4 Structural Parameters 2 Register (HCSPARAMS2)

11.1.3.1.4.1 Offset

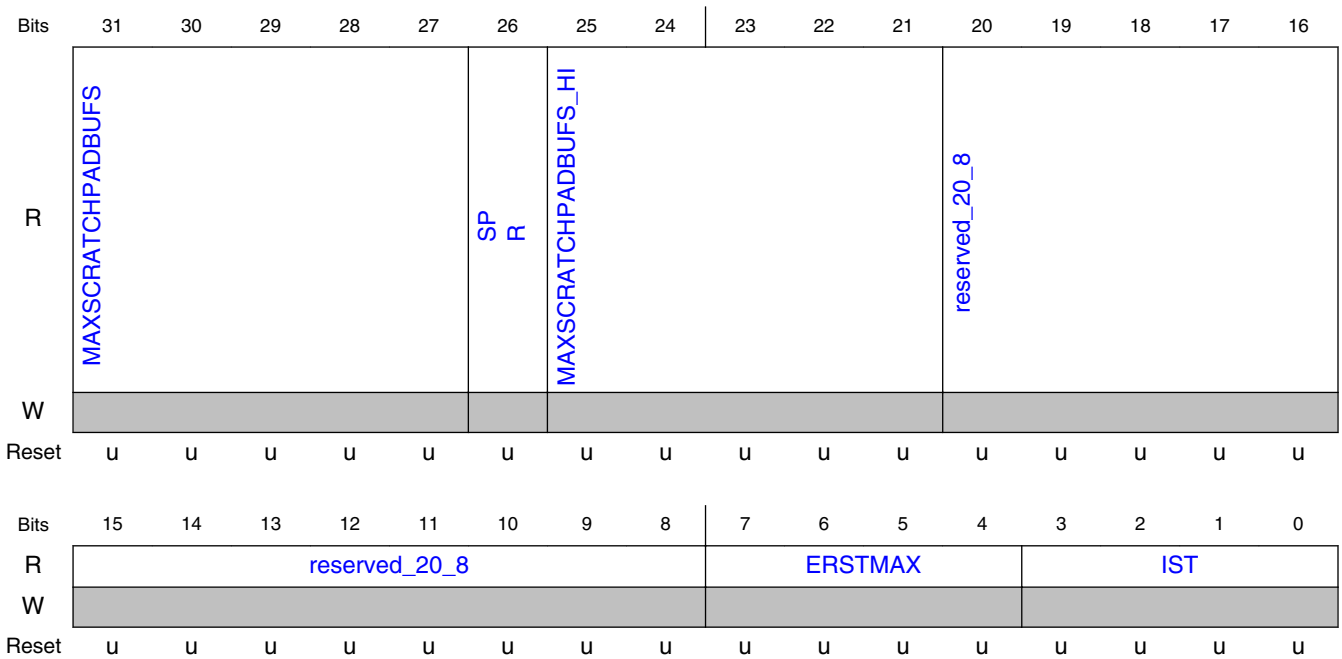
Register	Offset
HCSPARAMS2	8h

11.1.3.1.4.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0x1FFF00

11.1.3.1.4.3 Diagram



11.1.3.1.4.4 Fields

Field	Function
31-27 MAXSCRATCHPADBUFS	Max Scratchpad Bufs Lo The value is calculated based on chosen configuration parameter values. Possible values are 1-4.
26 SPR	Scratchpad Restore (SPR)
25-21 MAXSCRATCHPADBUFS_HI	Max Scratchpad Bufs HI The core automatically updates this field.
20-8 reserved_20_8	Reserved
7-4 ERSTMAX	Event Ring Segment Table Max (ERST Max)
3-0 IST	Isochronous Scheduling Threshold (IST)

11.1.3.1.5 Structural Parameters 3 Register (HCSPARAMS3)

11.1.3.1.5.1 Offset

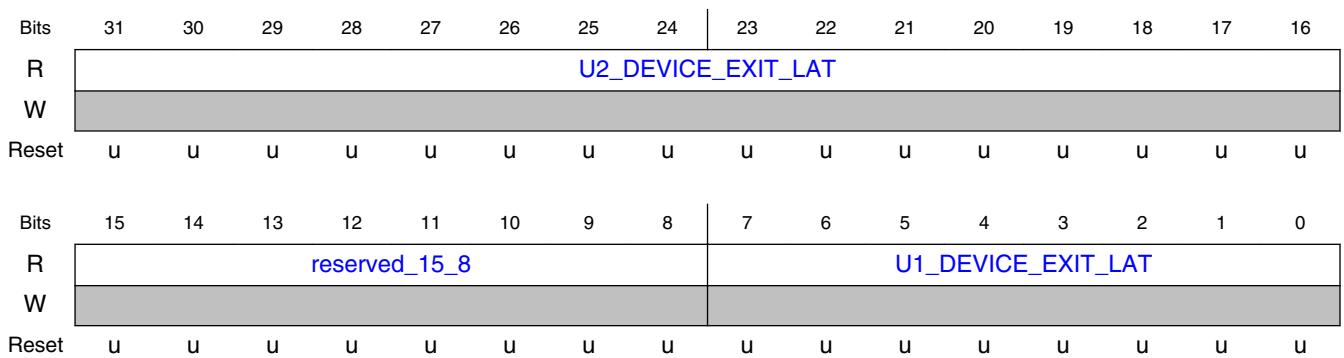
Register	Offset
HCSPARAMS3	Ch

11.1.3.1.5.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0xFF00

11.1.3.1.5.3 Diagram



11.1.3.1.5.4 Fields

Field	Function
31-16 U2_DEVICE_EXIT_LAT	U2 Device Exit Latency
15-8 reserved_15_8	Reserved
7-0 U1_DEVICE_EXIT_LAT	U1 Device Exit Latency

11.1.3.1.6 Capability Parameters 1 Register (HCCPARAMS1)

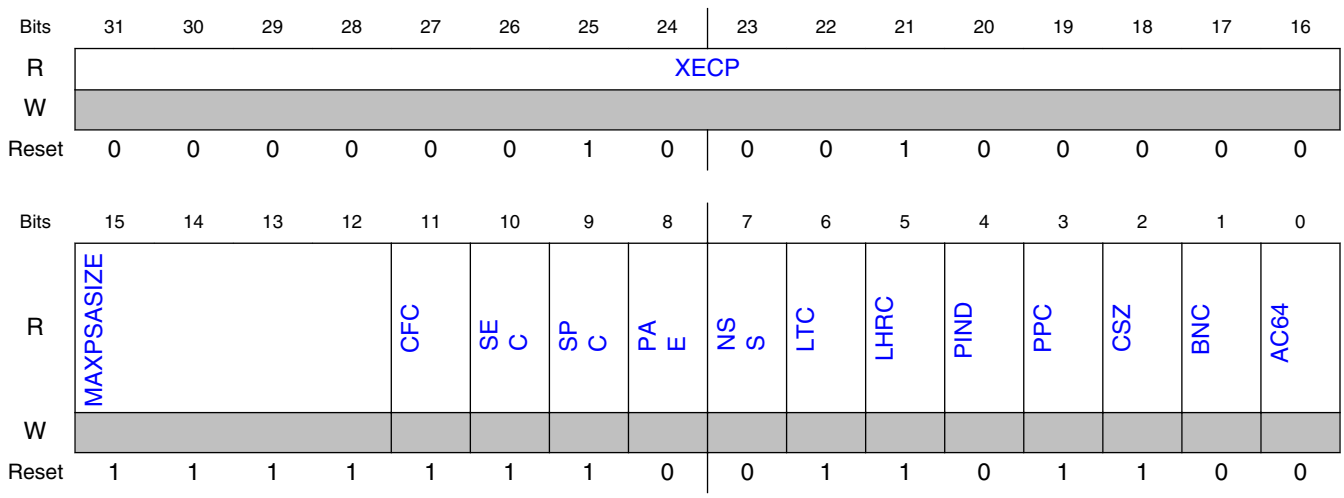
11.1.3.1.6.1 Offset

Register	Offset
HCCPARAMS1	10h

11.1.3.1.6.2 Function

For register definitions, refer to the xHCI specification.

11.1.3.1.6.3 Diagram



11.1.3.1.6.4 Fields

Field	Function
31-16 XECP	xHCI Extended Capabilities Pointer (xECP) Based on configuration, core automatically updates it. Refer to <workspace>/src/DWC_usb3_params.v for details on DWC_USB3_HC_XECP.
15-12 MAXPSASIZE	Maximum Primary Stream Array Size (MaxPSASize) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
11 CFC	Contiguous Frame ID Capability (CFC)
10 SEC	Stopped EDLTA Capability (SEC) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 SPC	Short Packet Capability (SPC) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8 PAE	Parse All Event Data (PAE) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Table continues on the next page...

Field	Function
7 NSS	No Secondary SID Support (NSS) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
6 LTC	Latency Tolerance Messaging Capability (LTC) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5 LHRC	Light HC Reset Capability For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4 PIND	Port Indicators (PIND) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3 PPC	Port Power Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 CSZ	Context Size (CSZ) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 BNC	BW Negotiation Capability (BNC) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 AC64	64-bit Addressing Capability (AC64) For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.7 Doorbell Offset Register (DBOFF)

11.1.3.1.7.1 Offset

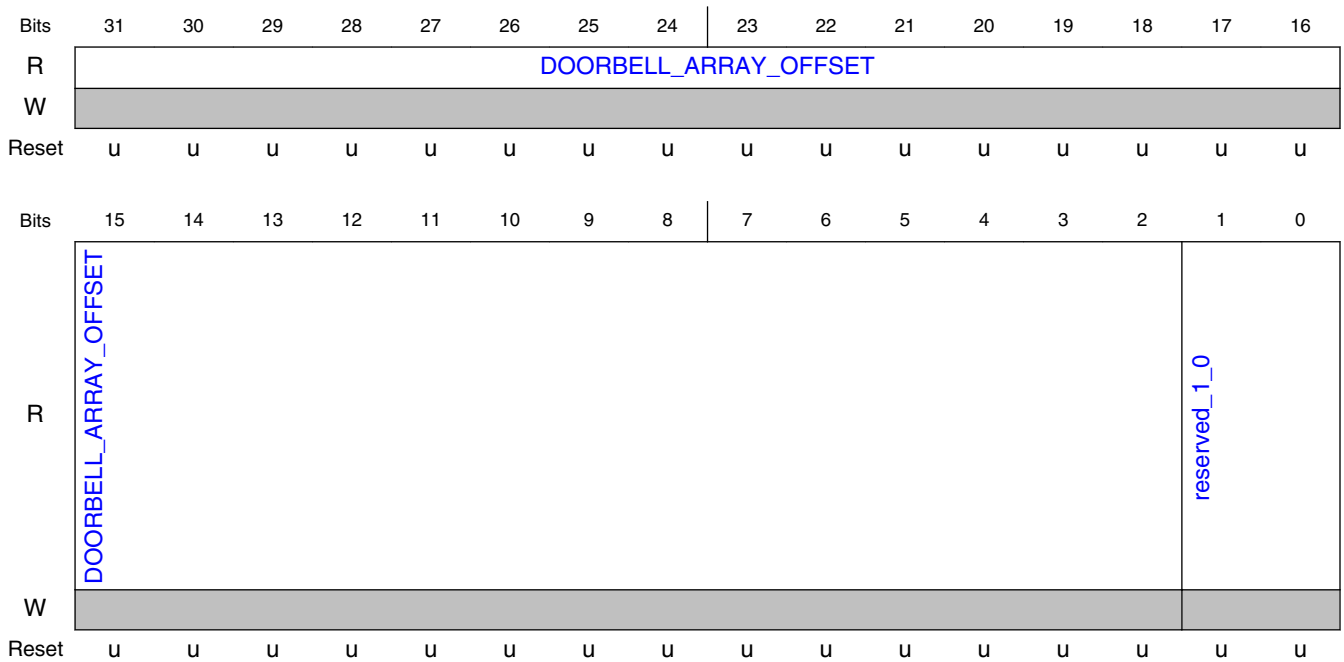
Register	Offset
DBOFF	14h

11.1.3.1.7.2 Function

For register definitions, refer to the xHCI specification.

Reset Mask:0x3

11.1.3.1.7.3 Diagram



11.1.3.1.7.4 Fields

Field	Function
31-2 DOORBELL_ARRAY_OFFSET	Doorbell Array Offset - RO Based on configuration, core automatically updates it. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1-0 reserved_1_0	Reserved

11.1.3.1.8 Runtime Register Space Offset Register (RTSOFF)

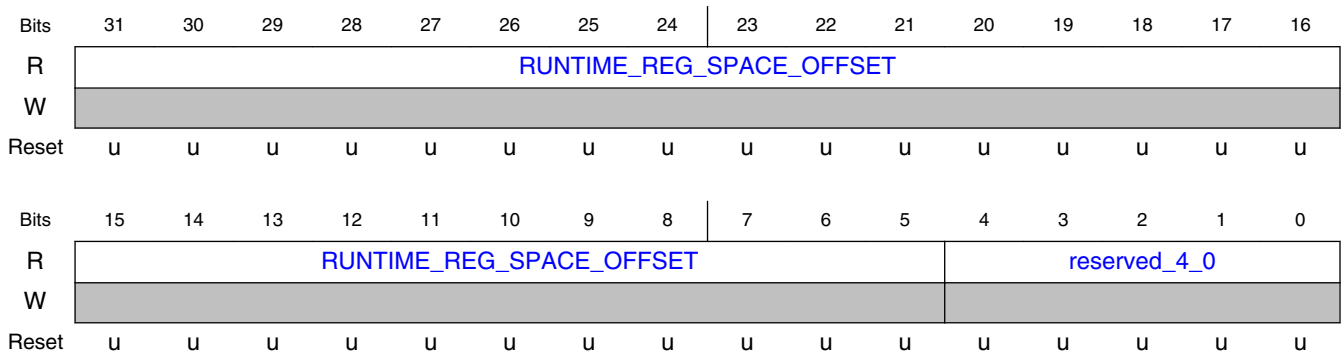
11.1.3.1.8.1 Offset

Register	Offset
RTSOFF	18h

11.1.3.1.8.2 Function

Reset Mask:0x1F

11.1.3.1.8.3 Diagram



11.1.3.1.8.4 Fields

Field	Function
31-5 RUNTIME_REG_SPACE_OFFSET	Runtime Register Space Offset Based on configuration, core automatically updates it. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
4-0 reserved_4_0	Reserved

11.1.3.1.9 Host Controller Capability Parameters 2 (HCCPARAMS2)

11.1.3.1.9.1 Offset

Register	Offset
HCCPARAMS2	1Ch

11.1.3.1.9.2 Function

For register definitions, refer to the xHCI specification.

11.1.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_31_6															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_31_6										CIC	LEC	CTC	FSC	CMC	U3C
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1	1

11.1.3.1.9.4 Fields

Field	Function
31-6 reserved_31_6	Reserved
5 CIC	Configuration Information Capability (CIC) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.
4 LEC	Large ESIT Payload Capability (LEC) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.
3 CTC	Compliance Transition Capability (CTC) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.
2 FSC	Force Save Context Capability (FSC) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.
1 CMC	Configure Endpoint Command Max Exit Latency Too Large Capability (CMC) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.
0 U3C	U3 Entry Capability (U3C) For a description of this standard USB register field, see the eXtensible Host Controller I nterface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.10 USB Command Register (USBCMD)

11.1.3.1.10.1 Offset

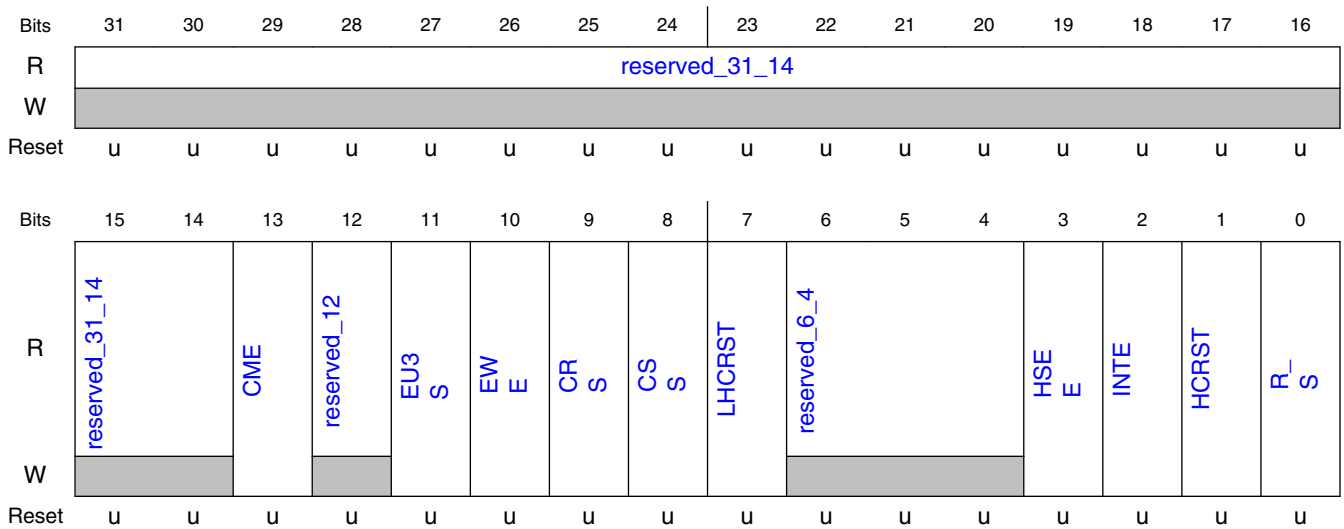
Register	Offset
USBCMD	20h

11.1.3.1.10.2 Function

For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0xFFFFC070

11.1.3.1.10.3 Diagram



11.1.3.1.10.4 Fields

Field	Function
31-14 reserved_31_14	Reserved
13 CME	CEM Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
12 reserved_12	Reserved
11 EU3S	EU3S For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
10 EWE	EWE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
9 CRS	Controller Restore State This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'. Note: When read, this field always returns '0'.
8 CSS	Controller Save State This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When

Table continues on the next page...

Field	Function
	the controller has finished the save process, it sets DSTS.SSS to '0'. Note: When read, this field always returns '0'.
7 LHCRST	Light Host Controller Reset For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0. The following bits reset the internal logic of the host controller. Under soft reset, some CSR accesses may fail (Timeout). - HCRST - LHCRST Bit Bash register testing is not recommended.
6-4 reserved_6_4	Reserved
3 HSEE	HSEE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 INTE	INTE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 HCRST	HCRST The following bits reset the internal logic of the host controller. Under soft reset, some CSR accesses may fail (Timeout). - HCRST - LHCRST Bit Bash register testing is not recommended.
0 R_S	R_S For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0. Due to side-effects this reguster field is not recommended for Bit-Bash testing.

11.1.3.1.11 USB Status Register (USBSTS)

11.1.3.1.11.1 Offset

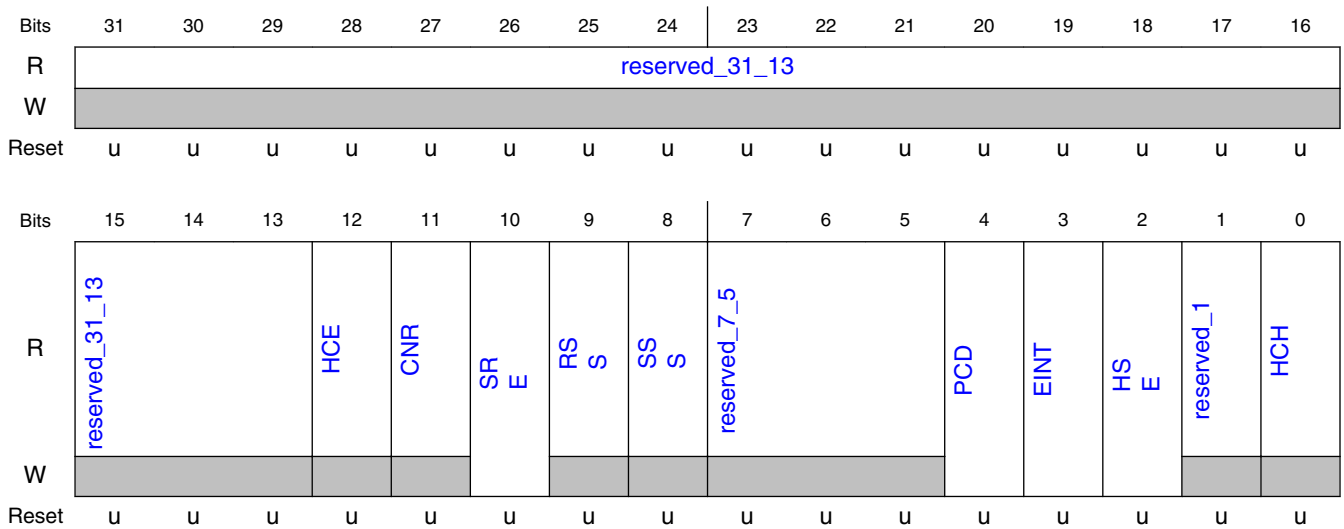
Register	Offset
USBSTS	24h

11.1.3.1.11.2 Function

Bit Definitions For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0xFFFFE0E2

11.1.3.1.11.3 Diagram



11.1.3.1.11.4 Fields

Field	Function
31-13 reserved_31_13	Reserved
12 HCE	Host Controller Error (HCE) - RO Default = 0. '0' = No internal xHC error conditions exist and '1' = Internal xHC error condition. This flag must be set to indicate that an internal error condition has been detected which requires software to reset and reinitialize the xHC. Refer to section 4.24.1 of the DesignWare Cores SuperSpeed USB 3.0 Controller Databookfor more information.
11 CNR	Controller Not Ready (CNR) - RO Default = '1'. '0' = Ready and '1' = Not Ready. Software must not write to thes Doorbell or Operational register of the xHC, other than the USBSTS register, until CNR = '0'. This flag is set by the xHC after a Chip Hardware Reset and cleared when the xHC is ready to begin accepting register writes. This flag remains cleared ('0') until the next Chip Hardware Reset.
10 SRE	Save/Restore Error This bit is currently not supported.
9 RSS	Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller has finished the restore process, it completes the command by setting DSTS.RSS to '0'.
8 SSS	Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to '0'.
7-5 reserved_7_5	Reserved
4 PCD	Reset Value for PCD
3 EINT	Reset Value for EINT
2	HSE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0..

Table continues on the next page...

Field	Function
HSE	
1 reserved_1	Reserved
0 HCH	HCH For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.12 Page Size Register (PAGESIZE)

11.1.3.1.12.1 Offset

Register	Offset
PAGESIZE	28h

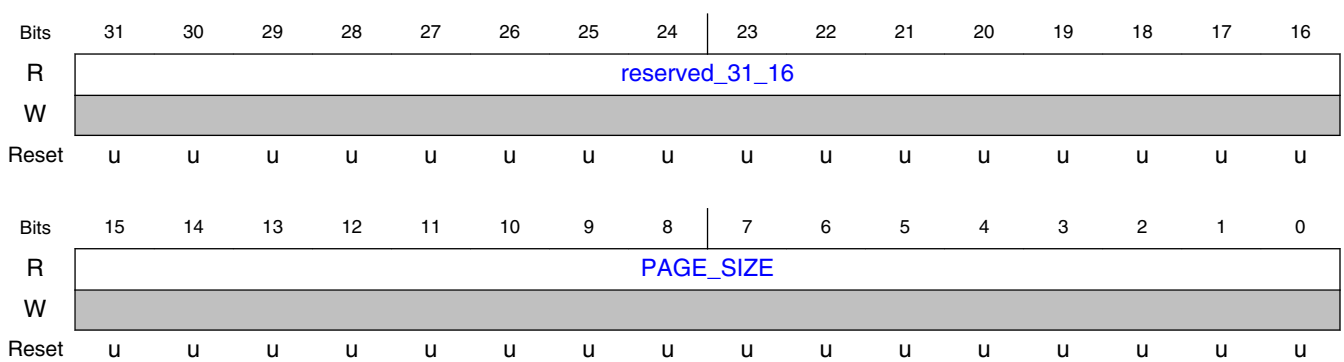
11.1.3.1.12.2 Function

Bit Definitions Use this register to enable or disable the reporting of specific USB Device Notification Transaction Packets being received.

Reset Mask:0xFFFF0000

A Notification Enable (Nx, where x = 0 to 15) flag is defined for each of the 16 possible device notification types. If a flag is set for a specific notification type, a Device Notification Event is generated when the respective notification packet is received. After reset, all notifications are disabled. Refer to section 6.4.2.7 of the DesignWare Cores SuperSpeed USB 3.0 Controller Databookfor more information. This register is written as a Dword. Byte writes produce undefined results.

11.1.3.1.12.3 Diagram



11.1.3.1.12.4 Fields

Field	Function
31-16 reserved_31_16	Reserved
15-0 PAGE_SIZE	PAGE_SIZE

11.1.3.1.13 Device Notification Register (DNCTRL)

11.1.3.1.13.1 Offset

Register	Offset
DNCTRL	34h

11.1.3.1.13.2 Function

Bit Definitions For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_31_16															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	N0_N15															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.13.4 Fields

Field	Function
31-16 reserved_31_16	Reserved

Table continues on the next page...

Field	Function
15-0 N0_N15	N0_N15 For a description of this standard USB register field see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.14 CRCR_LO (CRCR_LO)

11.1.3.1.14.1 Offset

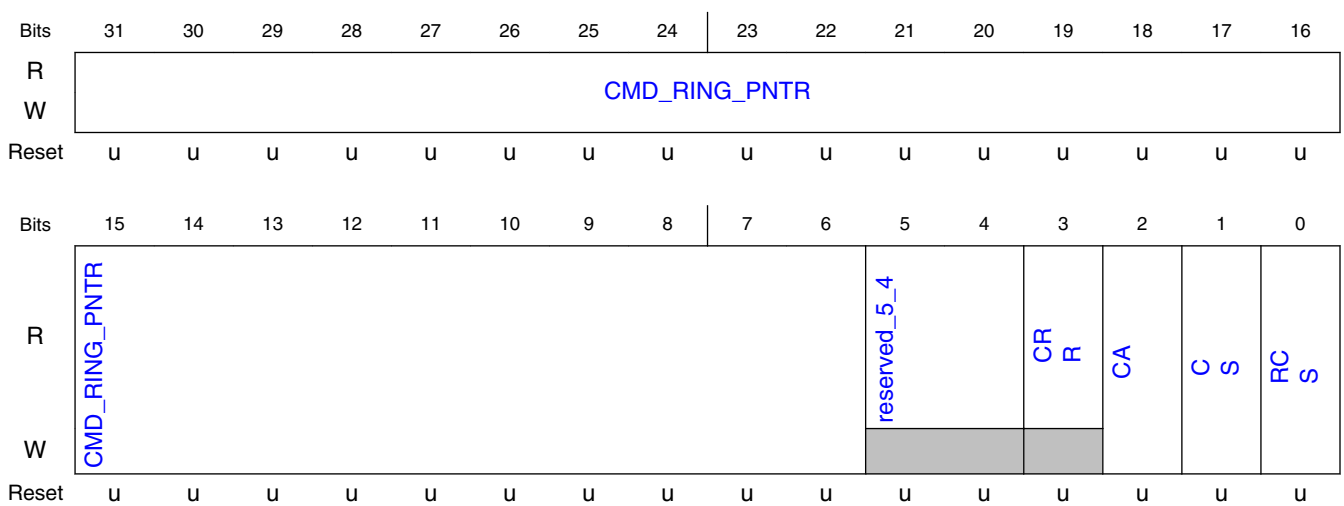
Register	Offset
CRCR_LO	38h

11.1.3.1.14.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

Reset Mask:0x30

11.1.3.1.14.3 Diagram



11.1.3.1.14.4 Fields

Field	Function
31-6	CMD_RING_PNTR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Table continues on the next page...

Field	Function
CMD_RING_PNTR	
5-4 reserved_5_4	Reserved
3 CRR	CRR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
2 CA	CA For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
1 CS	CS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
0 RCS	RCS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.15 (CRCR_HI)

11.1.3.1.15.1 Offset

Register	Offset
CRCR_HI	3Ch

11.1.3.1.15.2 Function

CRCR_HI

11.1.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CMD_RING_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_RING_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.15.4 Fields

Field	Function
31-0 CMD_RING_PNTR	COMMAND_RING_POINTER Reading this field always returns '0'. For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.16 DCBAAP_LO (DCBAAP_LO)

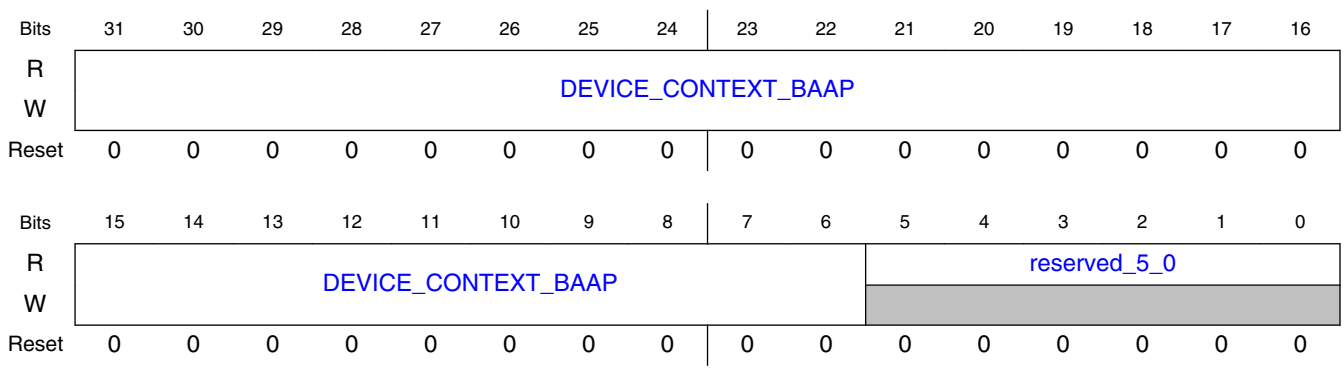
11.1.3.1.16.1 Offset

Register	Offset
DCBAAP_LO	50h

11.1.3.1.16.2 Function

DCBAAP_LO

11.1.3.1.16.3 Diagram



11.1.3.1.16.4 Fields

Field	Function
31-6 DEVICE_CONTEXT_BAAP	DEVICE_CONTEXT_BAAP For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
5-0 reserved_5_0	Reserved

11.1.3.1.17 DCBAAP_HI (DCBAAP_HI)

11.1.3.1.17.1 Offset

Register	Offset
DCBAAP_HI	54h

11.1.3.1.17.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEVICE_CONTEXT_BAAP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEVICE_CONTEXT_BAAP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.17.4 Fields

Field	Function
31-0 DEVICE_CONTEXT_BAAP	DEVICE_CONTEXT_BAAP

11.1.3.1.18 Configuration Register (CONFIG)

11.1.3.1.18.1 Offset

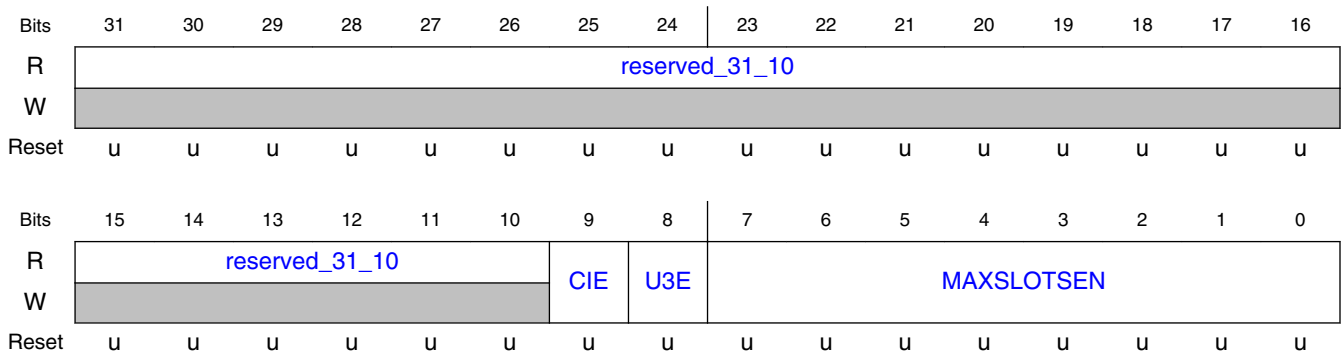
Register	Offset
CONFIG	58h

11.1.3.1.18.2 Function

This register is in the Aux Power well.

Reset Mask:0xFFFFFC00

11.1.3.1.18.3 Diagram



11.1.3.1.18.4 Fields

Field	Function
31-10 reserved_31_10	Reserved
9 CIE	Configuration Information Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
8 U3E	U3 Entry Enable For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 MAXSLOTSEN	MAXSLOTSEN For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.19 Port Status and Control Register (PORTSC_20)

11.1.3.1.19.1 Offset

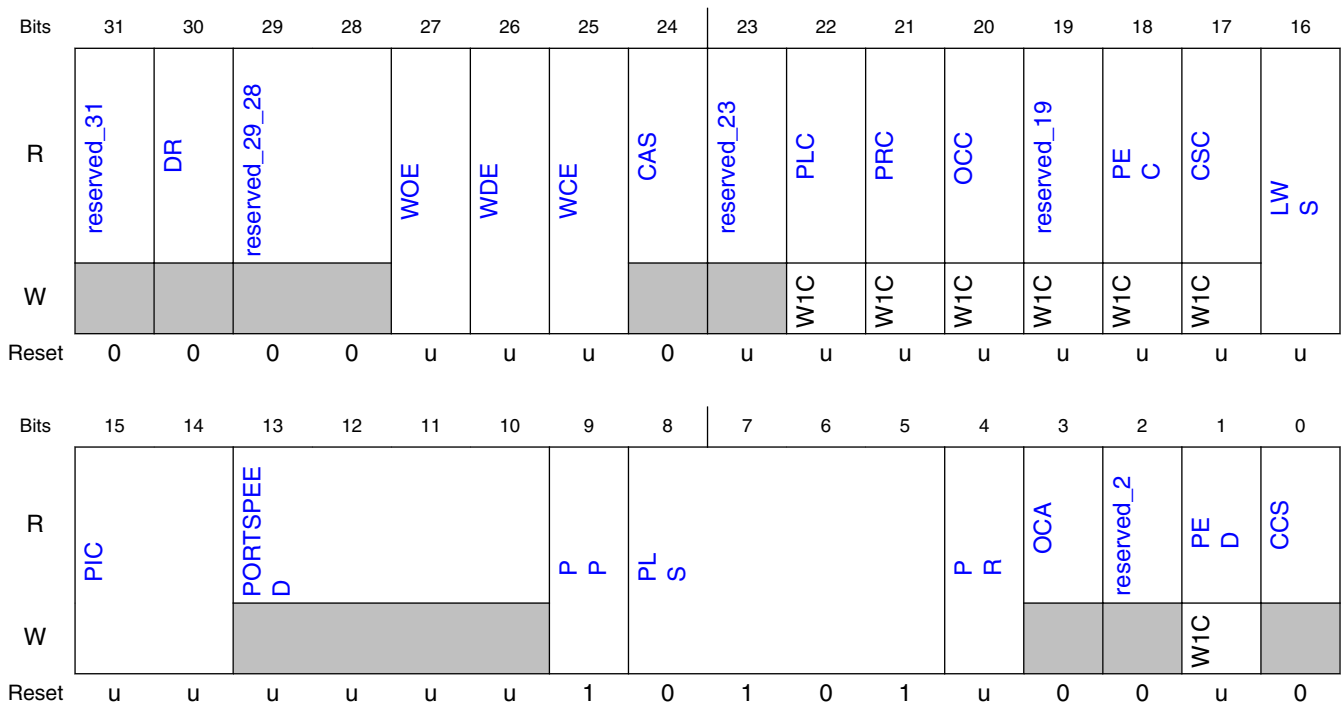
Register	Offset
PORTSC_20	420h

11.1.3.1.19.2 Function

Access fails (Timeout) if the UTMI/ULPI clock is not running or one of the following bits is asserted.

Reset Mask:0xF10003ED

11.1.3.1.19.3 Diagram



11.1.3.1.19.4 Fields

Field	Function
31 reserved_31	
30 DR	Reset value For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
29-28 reserved_29_28	Reserved
27 WOE	WOE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
26	WDE

Table continues on the next page...

Field	Function
WDE	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
25 WCE	WCE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
24 CAS	CAS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
23 reserved_23	
22 PLC	PLC PLC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
21 PRC	PRC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
20 OCC	OCC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
19 reserved_19	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
18 PEC	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
17 CSC	CSC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
16 LWS	LWS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
15-14 PIC	PIC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
13-10 PORTSPEED	PORTSPEED For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
9 PP	PP For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
8-5 PLS	PLS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
4	PR

Table continues on the next page...

Field	Function
PR	DWC_usb3_map/DWC_usb3_block_Host_Cntrl_Port_Reg_Set/PORTSC_20_REGS/PORTSC_20/PR VolatileMemory 1 Programming this field with random data will cause side effect.
3 OCA	OCA For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
2 reserved_2	Reserved
1 PED	PED For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
0 CCS	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3

11.1.3.1.20 USB3 Port Power Management Status and Control Register (PORTPMSC_20)

11.1.3.1.20.1 Offset

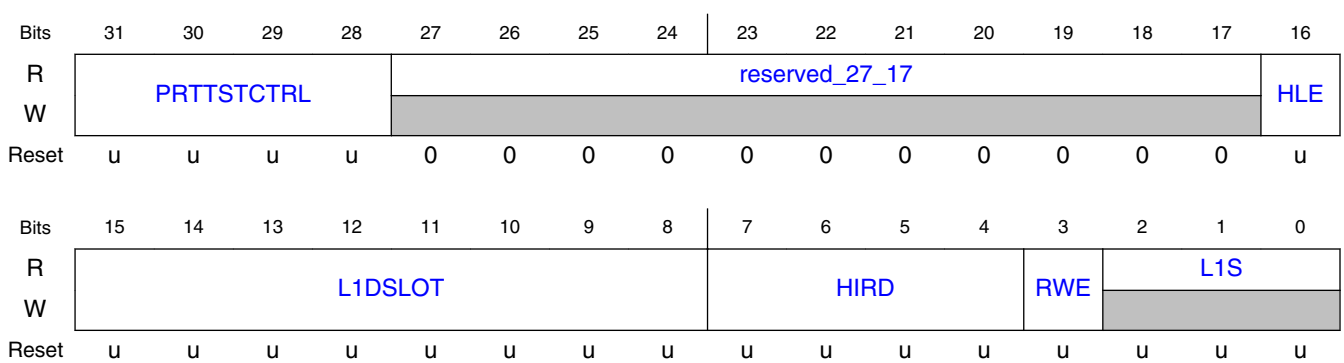
Register	Offset
PORTPMSC_20	424h

11.1.3.1.20.2 Function

This register is in the Aux Power well.

Reset Mask:0xFFE0000

11.1.3.1.20.3 Diagram



11.1.3.1.20.4 Fields

Field	Function
31-28 PRTTSTCTRL	Port Test Control . When this field is '0', the port is NOT operating in a test mode. A non-zero value indicates that it is operating in test mode and the specific test mode is indicated by the specific value. A non-zero Port Test Control value is only valid to a port that is in the Powered-Off state (PLS = Disabled). If the port is not in this state, the xHC shall respond with the Port Test Control field set to Port Test Control Error. Refer to 'Operational Model' topic for using these test modes. The encoding of the Test Mode bits for a USB2 protocol port are: 0: Test mode not enabled 1: Test J_STATE 2: Test K_STATE 3: Test SE0_NAK 4: Test Packet 5: Test FORCE_ENABLE 6-14: Reserved. 15: Port Test Control Error.
27-17 reserved_27_17	Reserved
16 HLE	Port Test Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
15-8 L1DSLOT	L1DSLOT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
7-4 HIRD	Host Initiated Resume Duration (HIRD)
3 RWE	RWE Port Test Control For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
2-0 L1S	L1 Status (L1S)

11.1.3.1.21 (PORTLI_20)

11.1.3.1.21.1 Offset

Register	Offset
PORTLI_20	428h

11.1.3.1.21.2 Function

Port Link Info Register Programming this field with random data will cause side effect i.

11.1.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.21.4 Fields

Field	Function
31-0 reserved_31_0	Reserved

11.1.3.1.22 (PORTHLPMC_20)

11.1.3.1.22.1 Offset

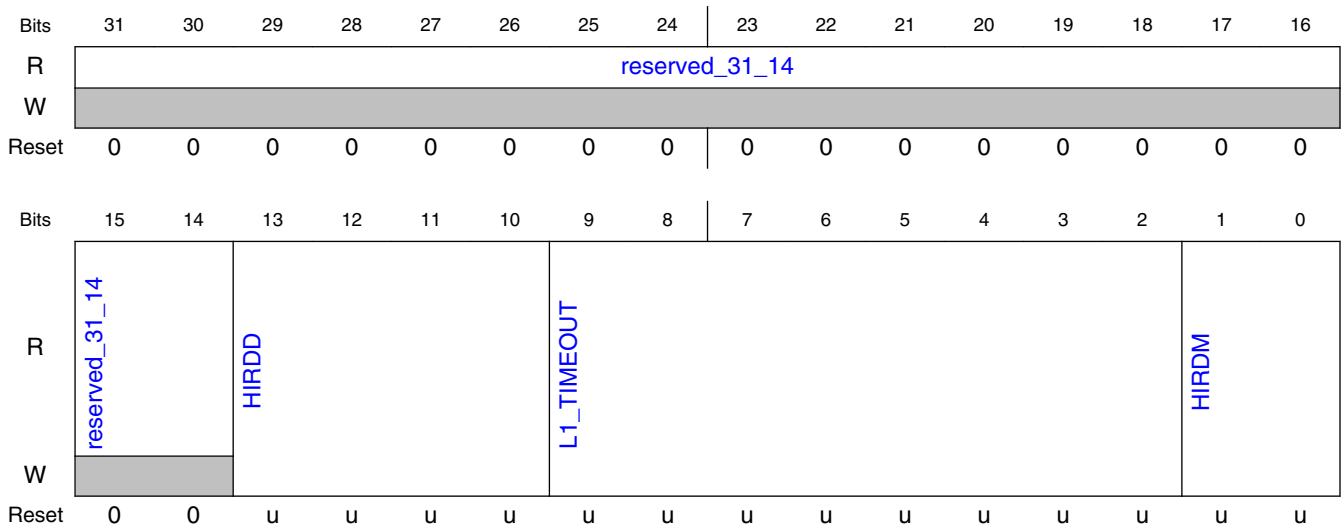
Register	Offset
PORTHLPMC_20	42Ch

11.1.3.1.22.2 Function

USB2 Port Hardware LPM Control Register Bit Definitions For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

Reset Mask:0xFFFFC000

11.1.3.1.22.3 Diagram



11.1.3.1.22.4 Fields

Field	Function
31-14 reserved_31_14	Reserved For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
13-10 HIRDD	PORTHLPMC_20 HIRDD
9-2 L1_TIMEOUT	PORTHLPMC_20 L1_TIMEOUT.
1-0 HIRDM	Host Initiated Resume Duration Mode (HIRDM) - RWS.

11.1.3.1.23 (PORTSC_30)

11.1.3.1.23.1 Offset

Register	Offset
PORTSC_30	430h

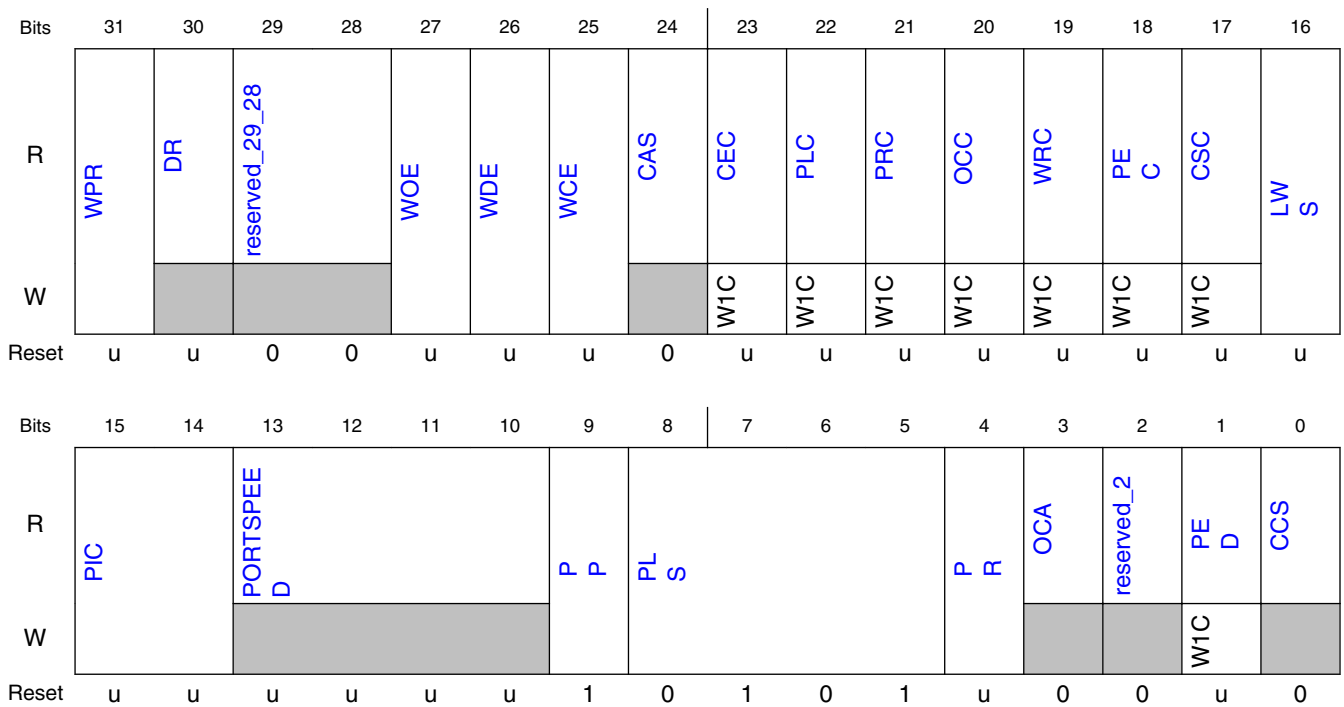
11.1.3.1.23.2 Function

Port Status and Control Register Bit Definitions The PORTSC Register

Access fails (Timeout) if the UTMI/ULPI clock is not running or one of the following bits is asserted.

Reset Mask:0x310003ED

11.1.3.1.23.3 Diagram



11.1.3.1.23.4 Fields

Field	Function
31 WPR	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
30 DR	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
29-28 reserved_29_28	Reserved
27 WOE	WOE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

Table continues on the next page...

Field	Function
26 WDE	WDE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
25 WCE	WCE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
24 CAS	Cold Attach Status For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
23 CEC	CEC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
22 PLC	PLC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
21 PRC	PRC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
20 OCC	OCC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
19 WRC	WRC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
18 PEC	PEC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
17 CSC	CSC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
16 LWS	LWS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
15-14 PIC	PIC For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
13-10 PORTSPEED	PORTSPEED For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
9 PP	PP For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification

Table continues on the next page...

Field	Function
	3.
8-5 PLS	PLS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
4 PR	PR set_register_field_attribute DWC_usb3_map/DWC_usb3_block_Host_Cntrl_Port_Reg_Set/PORTSC_30_REGS/PORTSC_30/PR VolatileMemory 1 Programming this field with random data will cause side effect.
3 OCA	OCA For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
2 reserved_2	Reserved
1 PED	PED For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
0 CCS	CCS For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.24 USB3 Port Power Management Status and Control Register (PORTPMSC_30)

11.1.3.1.24.1 Offset

Register	Offset
PORTPMSC_30	434h

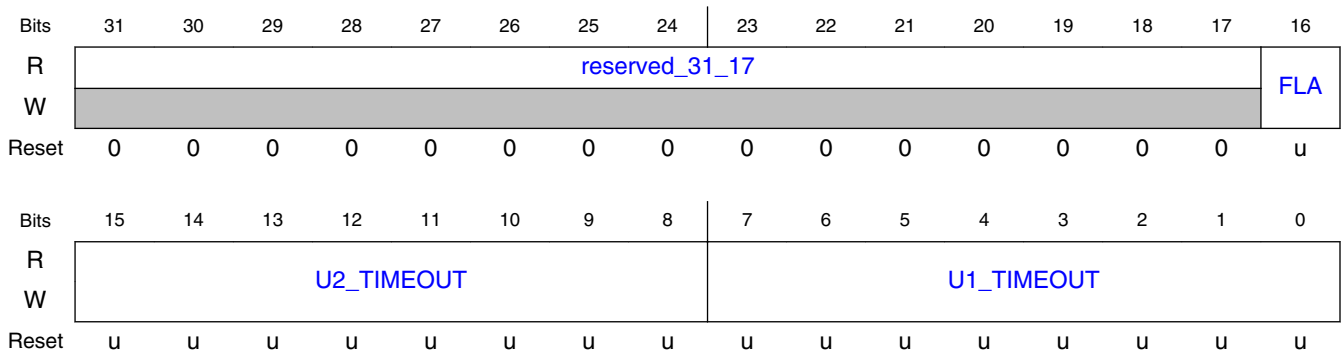
11.1.3.1.24.2 Function

USB3 Port Power Management Status and Control Register Bit Definitions

This register is in the Aux Power well.

Reset Mask:0xFFFE0000

11.1.3.1.24.3 Diagram



11.1.3.1.24.4 Fields

Field	Function
31-17 reserved_31_17	Reserved
16 FLA	FLA For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
15-8 U2_TIMEOUT	U2_TIMEOUT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
7-0 U1_TIMEOUT	U1_TIMEOUT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.25 Port Link Info Register (PORTLI_30)

11.1.3.1.25.1 Offset

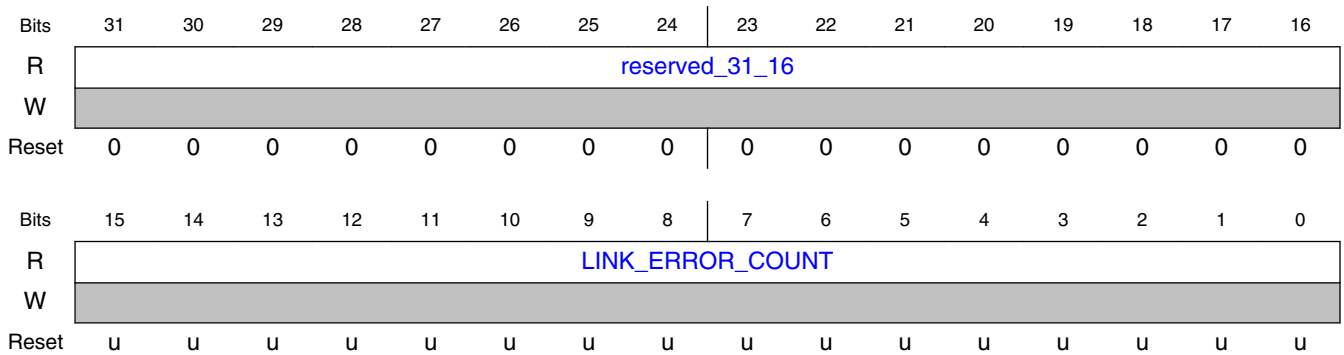
Register	Offset
PORTLI_30	438h

11.1.3.1.25.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

Reset Mask:0xFFFF0000

11.1.3.1.25.3 Diagram



11.1.3.1.25.4 Fields

Field	Function
31-16 reserved_31_16	Reserved
15-0 LINK_ERROR_COUNT	LINK_ERROR_COUNT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.26 USB2 Port Hardware LPM Control Register (PORTHLPMC_30)

11.1.3.1.26.1 Offset

Register	Offset
PORTHLPMC_30	43Ch

11.1.3.1.26.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_31_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.26.4 Fields

Field	Function
31-0 reserved_31_0	Reserved For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.27 Microframe Index Register (MFINDEX)

11.1.3.1.27.1 Offset

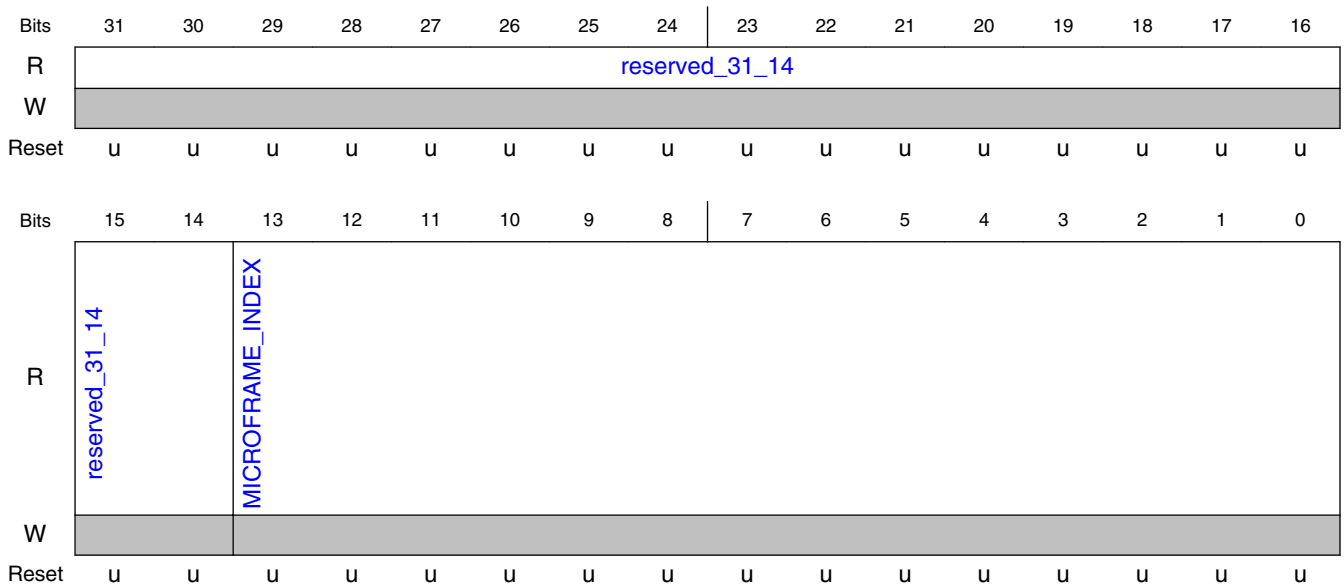
Register	Offset
MFINDEX	440h

11.1.3.1.27.2 Function

See the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

Reset Mask:0xFFFFC000

11.1.3.1.27.3 Diagram



11.1.3.1.27.4 Fields

Field	Function
31-14 reserved_31_14	Reserved
13-0 MICROFRAME_INDEX	MICROFRAME_INDEX For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.28 Interrupter Management Register (IMAN)

11.1.3.1.28.1 Offset

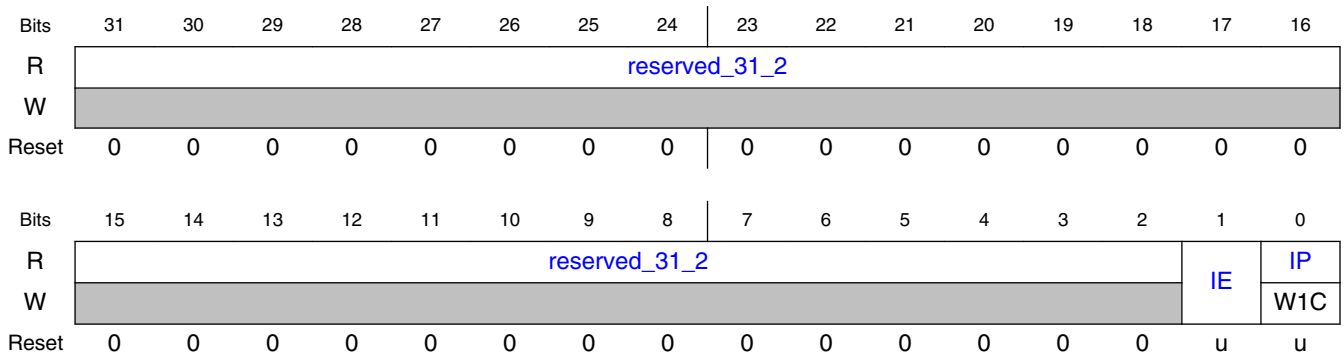
Register	Offset
IMAN	460h

11.1.3.1.28.2 Function

Reset Mask:0xFFFFFFFC

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.28.3 Diagram



11.1.3.1.28.4 Fields

Field	Function
31-2 reserved_31_2	Reserved
1 IE	For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
0 IP	IP Interrupt Pending For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.29 Interrupter Moderation Register (IMOD)

11.1.3.1.29.1 Offset

Register	Offset
IMOD	464h

11.1.3.1.29.2 Function

The software may use this register to

pace (or even out) the delivery of interrupts to the host CPU.

11.1.3.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	IMODC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMODI															
Reset	0	0	0	0	1	1	1	1	1	0	1	0	0	0	0	0

11.1.3.1.29.4 Fields

Field	Function
31-16 IMODC	Interrupt Moderation Counter (IMODC) - RW.
15-0 IMODI	Interrupt Moderation Interval (IMODI) - RW.

11.1.3.1.30 ERSTSZ (ERSTSZ)

11.1.3.1.30.1 Offset

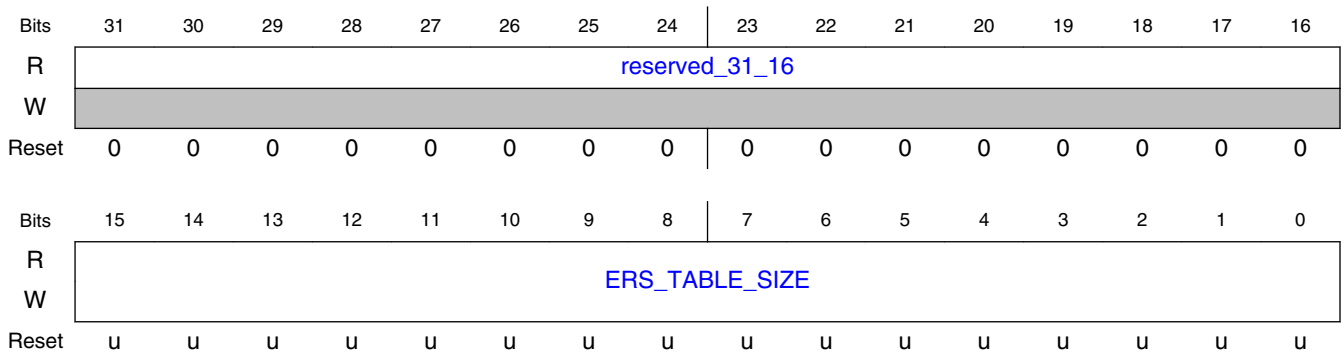
Register	Offset
ERSTSZ	468h

11.1.3.1.30.2 Function

Event Ring Segment Table Size Register Bit Definitions For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

Reset Mask:0xFFFF0000

11.1.3.1.30.3 Diagram



11.1.3.1.30.4 Fields

Field	Function
31-16 reserved_31_16	Reserved
15-0 ERS_TABLE_SIZE	ERS_TABLE_SIZE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.31 ERSTBA_LO (ERSTBA_LO)

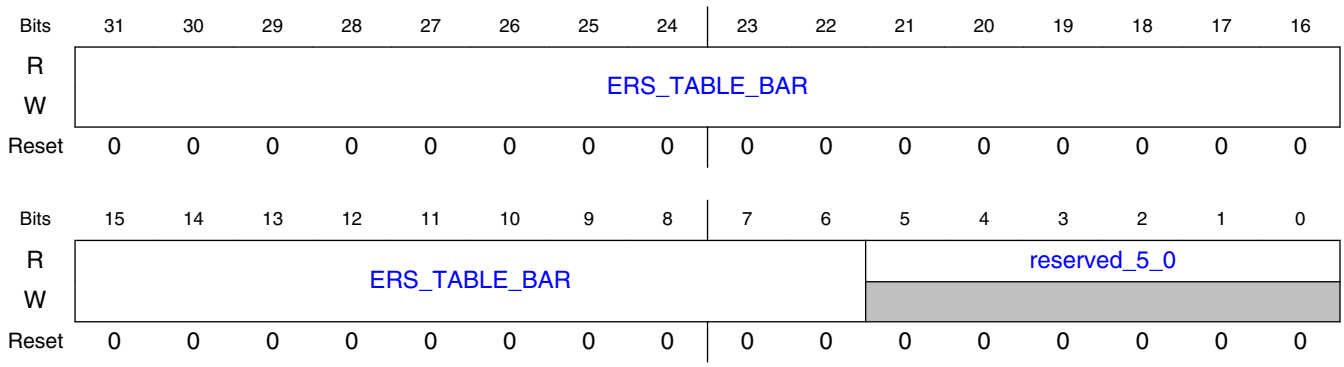
11.1.3.1.31.1 Offset

Register	Offset
ERSTBA_LO	470h

11.1.3.1.31.2 Function

ERSTBA_LO

11.1.3.1.31.3 Diagram



11.1.3.1.31.4 Fields

Field	Function
31-6 ERS_TABLE_BAR	ERS_TABLE_BAR
5-0 reserved_5_0	Reserved

11.1.3.1.32 ERSTBA_HI (ERSTBA_HI)

11.1.3.1.32.1 Offset

Register	Offset
ERSTBA_HI	474h

11.1.3.1.32.2 Function

ERSTBA_HI

11.1.3.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERS_TABLE_BAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERS_TABLE_BAR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.32.4 Fields

Field	Function
31-0 ERS_TABLE_BAR	ERS_TABLE_BAR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0

11.1.3.1.33 ERDP_LO (ERDP_LO)

11.1.3.1.33.1 Offset

Register	Offset
ERDP_LO	478h

11.1.3.1.33.2 Function

ERDP_LO For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERD_PNTR												EHB W1C	DESI		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.33.4 Fields

Field	Function
31-4 ERD_PNTR	ERD_PNTR ERD_PNTR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.
3 EHB	EHB HC OS Owned SemaphoreERS_TABLE_SIZEHC BIOS Owned Semaphore USB SMI Enable SMI on Host System Error Enable - SMI on OS Ownership Enable - SMI on PCI Command Enable - SMI on BAR Enable - SMI on Event Interrupt - SMI on Host System Error - SMI on OS Ownership Change - SMI on PCI Command - SMI on BAR - Compatible Port Count - HC OS Owned Semaphore - HC BIOS Owned Semaphore - USB SMI Enable - SMI on Host System Error Enable - SMI on OS Ownership Enable - SMI on PCI Command Enable - SMI on BAR Enable - SMI on Event Interrupt - SMI on Host System Error - SMI on OS Ownership Change - SMI on PCI Command - SMI on BAR - Compatible Port Count
2-0 DESI	DESI - For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.34 ERDP_HI (ERDP_HI)

11.1.3.1.34.1 Offset

Register	Offset
ERDP_HI	47Ch

11.1.3.1.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ERD_PNTR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.34.3 Fields

Field	Function
31-0 ERD_PNTR	ERD_PNTR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.35 Doorbell Register (DB)

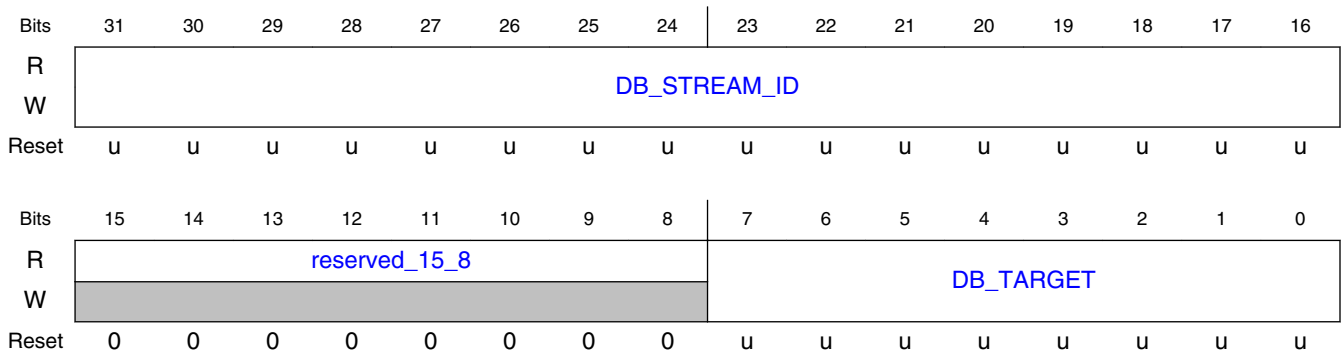
11.1.3.1.35.1 Offset

Register	Offset
DB	480h

11.1.3.1.35.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3. Reset Mask:0xFF00

11.1.3.1.35.3 Diagram



11.1.3.1.35.4 Fields

Field	Function
31-16 DB_STREAM_ID	DB_STREAM_ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0
15-8 reserved_15_8	Reserved
7-0 DB_TARGET	DB_TARGET For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.36 USBLEGSUP (USBLEGSUP)

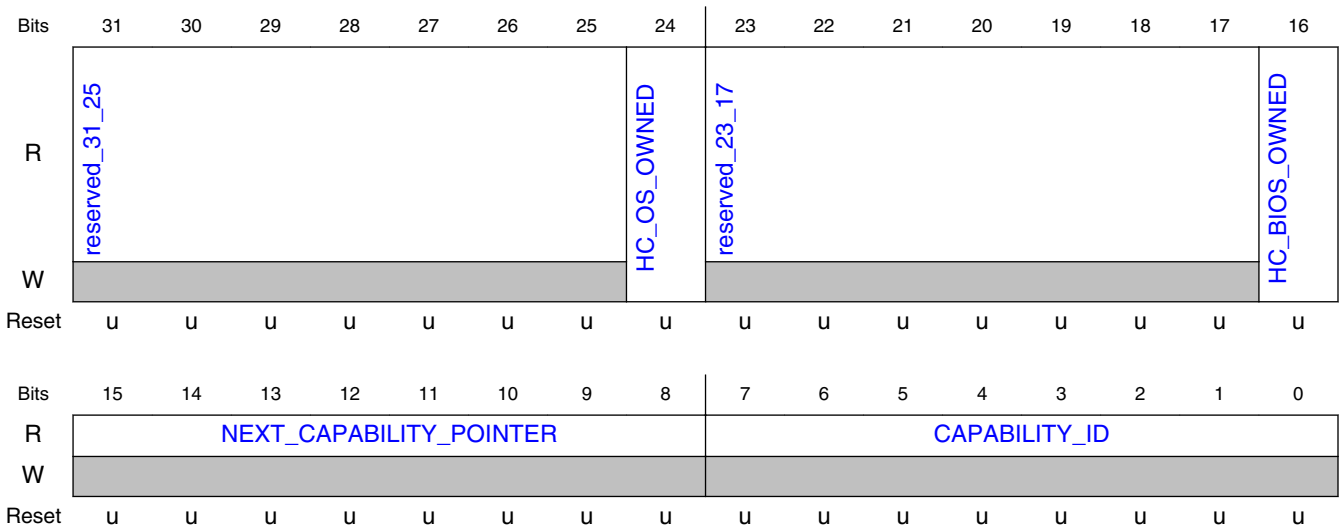
11.1.3.1.36.1 Offset

Register	Offset
USBLEGSUP	880h

11.1.3.1.36.2 Function

Reset Mask:0x7E7E0000

11.1.3.1.36.3 Diagram



11.1.3.1.36.4 Fields

Field	Function
31-25 reserved_31_25	Reserved
24 HC_OS_OWNED	HC_OS_OWNED SEMAPHORE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23-17 reserved_23_17	Reserved
16 HC_BIOS_OWNED	HC_BIOS_OWNED SEMAPHORE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 NEXT_CAPABILITY_POINTER	NEXT_CAPABILITY_POINTER For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 CAPABILITY_ID	CAPABILITY_ID set_register_field_attribute DWC_usb3_map/ DWC_usb3_block_HC_Extended_Capability_Register/USBLEGSUP/CAPABILITY_ID RegisterResetValue 0x1 For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.37 USBLEGCTLSTS (USBLEGCTLSTS)

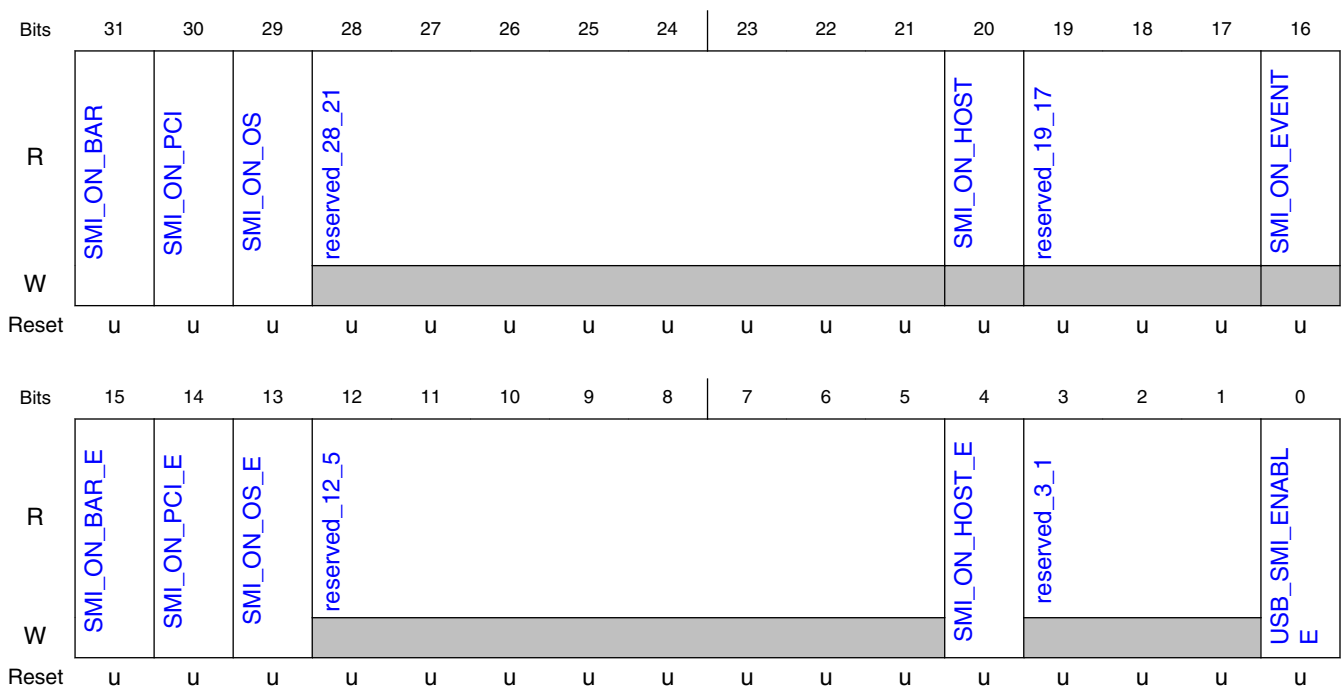
11.1.3.1.37.1 Offset

Register	Offset
USBLEGCTLSTS	884h

11.1.3.1.37.2 Function

Reset Mask:0x1FEE1FEE

11.1.3.1.37.3 Diagram



11.1.3.1.37.4 Fields

Field	Function
31 SMI_ON_BAR	SMI_ON_BAR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
30 SMI_ON_PCI	SMI_ON_PCI COMMAND For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
29 SMI_ON_OS	SMI_ON_OS OWNERSHIP CHANGE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
28-21	Reserved

Table continues on the next page...

Field	Function
reserved_28_21	
20 SMI_ON_HOST	SMI_ON_HOST SYSTEM ERROR For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
19-17 reserved_19_17	Reserved
16 SMI_ON_EVENT	SMI_ON_EVENT INTERRUPT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15 SMI_ON_BAR_E	SMI_ON_BAR ENABLE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
14 SMI_ON_PCI_E	SMI_ON_PCI COMMAND ENABLE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
13 SMI_ON_OS_E	SMI_ON_OS OWNERSHIP ENABLE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
12-5 reserved_12_5	Reserved
4 SMI_ON_HOST_E	SMI_ON_HOST SYSTEM ERROR ENABLE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
3-1 reserved_3_1	Reserved
0 USB_SMI_ENABLE	USB_SMI_ENABLE For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.38 SUPTPRT2_DW0 (SUPTPRT2_DW0)

11.1.3.1.38.1 Offset

Register	Offset
SUPTPRT2_DW0	890h

11.1.3.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR_REVISION								MINOR_REVISION							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NEXT_CAPABILITY_POINTER								CAPABILITY_ID							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

11.1.3.1.38.3 Fields

Field	Function
31-24 MAJOR_REVISION	MAJOR_REVISION
23-16 MINOR_REVISION	MINOR_REVISION
15-8 NEXT_CAPABILITY_POINTER	NEXT_CAPABILITY_POINTER
7-0 CAPABILITY_ID	CAPABILITY_ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.39 SUPTPRT2_DW1 Register (SUPTPRT2_DW1)

11.1.3.1.39.1 Offset

Register	Offset
SUPTPRT2_DW1	894h

11.1.3.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NAME_STRING															
W																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME_STRING															
W																
Reset	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1

11.1.3.1.39.3 Fields

Field	Function
31-0 NAME_STRING	NAME_STRING For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.40 xHCI Supported Protocol Capability_ Data Word 2 (SUPTPRT2_DW2)

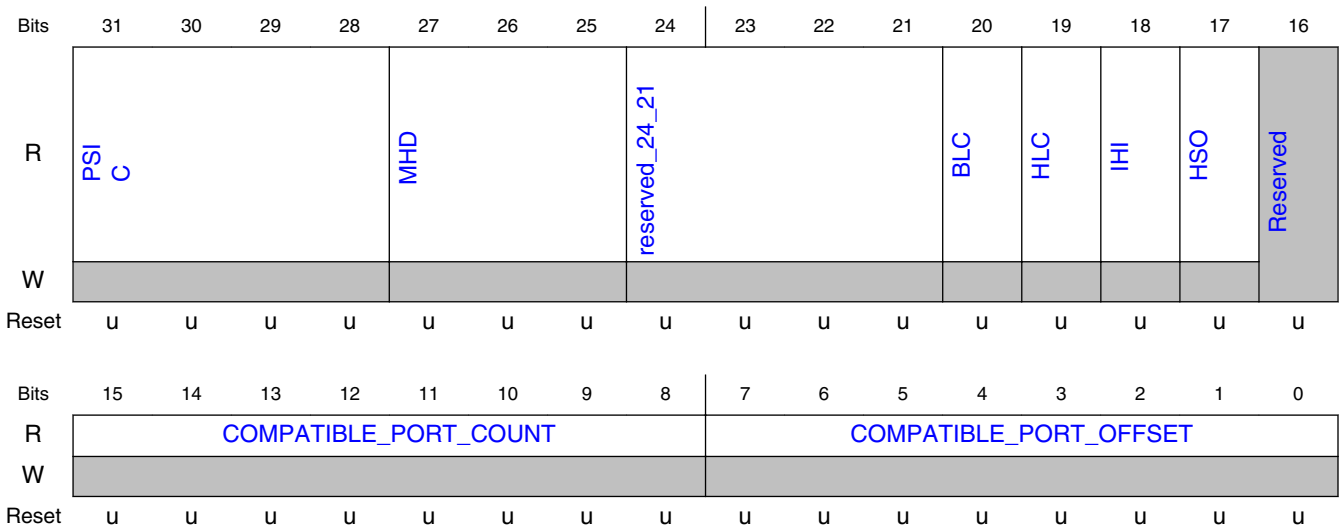
11.1.3.1.40.1 Offset

Register	Offset
SUPTPRT2_DW2	898h

11.1.3.1.40.2 Function

Reset Mask:0x1E00000

11.1.3.1.40.3 Diagram



11.1.3.1.40.4 Fields

Field	Function
31-28 PSIC	PSIC
27-25 MHD	Hub Depth For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24-21 reserved_24_21	Reserved
20 BLC	BESL LPM Capability When this bit is set to: - 1: The ports described by this xHCI Supported Protocol Capability applies BESL timing to the BESL and BESLD fields of the PORTPMSC and PORTHLPMC registers. - 0: The ports described by this xHCI Supported Protocol Capability applies HIRD timing to the BESL and BESLD fields of the PORTPMSC and PORTHLPMC registers.
19 HLC	Compatible Port Offset. Compatible Port Count Refer to Table 7-3 in the DWC Cores SuperSpeed USB 3.0 Controller Databook.
18 IHI	IHI
17 HSO	HSO
16 —	Reserved
15-8 COMPATIBLE_PORT_COUNT	COMPATIBLE_PORT_COUNT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

Table continues on the next page...

Field	Function
7-0 COMPATIBLE_PORT_OFFSET	COMPATIBLE_PORT_OFFSET For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.41 SUPTPRT2_DW3 Register (SUPTPRT2_DW3)

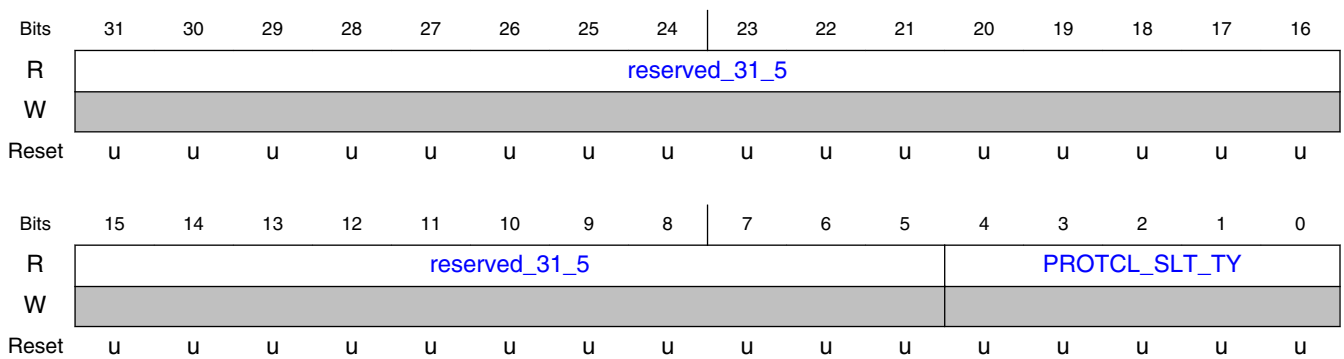
11.1.3.1.41.1 Offset

Register	Offset
SUPTPRT2_DW3	89Ch

11.1.3.1.41.2 Function

Reset Mask:0xFFFFFFE0

11.1.3.1.41.3 Diagram



11.1.3.1.41.4 Fields

Field	Function
31-5 reserved_31_5	Reserved
4-0 PROTCL_SLT_TY	Protocol Slot Type

11.1.3.1.42 (SUPTPRT3_DW0)

11.1.3.1.42.1 Offset

Register	Offset
SUPTPRT3_DW0	8A0h

11.1.3.1.42.2 Function

Register SUPTPRT3_DW0

11.1.3.1.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR_REVISION								MINOR_REVISION							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NEXT_CAPABILITY_POINTER								CAPABILITY_ID							
W																
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0

11.1.3.1.42.4 Fields

Field	Function
31-24 MAJOR_REVISION	MAJOR_REVISION For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
23-16 MINOR_REVISION	MINOR_REVISION For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
15-8 NEXT_CAPABILITY_POINTER	NEXT_CAPABILITY_POINTER For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 CAPABILITY_ID	CAPABILITY_ID For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.43 SUPTPRT3_DW1 Register (SUPTPRT3_DW1)

11.1.3.1.43.1 Offset

Register	Offset
SUPTPRT3_DW1	8A4h

11.1.3.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	NAME_STRING															
W																
Reset	0	0	1	0	0	0	0	0	0	1	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	NAME_STRING															
W																
Reset	0	1	0	1	0	0	1	1	0	1	0	1	0	1	0	1

11.1.3.1.43.3 Fields

Field	Function
31-0 NAME_STRING	NAME_STRING For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.44 SUPTPRT3_DW2 (SUPTPRT3_DW2)

11.1.3.1.44.1 Offset

Register	Offset
SUPTPRT3_DW2	8A8h

11.1.3.1.44.2 Function

For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.

11.1.3.1.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PSIC				MHD				Reserved							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COMPATIBLE_PORT_COUNT								COMPATIBLE_PORT_OFFSET							
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	1	0

11.1.3.1.44.4 Fields

Field	Function
31-28 PSIC	PSIC
27-25 MHD	Hub Depth For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
24-16 —	Reserved
15-8 COMPATIBLE_PORT_COUNT	COMPATIBLE_PORT_COUNT For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.
7-0 COMPATIBLE_PORT_OFFSET	COMPATIBLE_PORT_OFFSET For a description of this standard USB register field, see the eXtensible Host Controller Interface for Universal Serial Bus (USB) Specification 3.0.

11.1.3.1.45 SUPTPRT3_DW3 (SUPTPRT3_DW3)

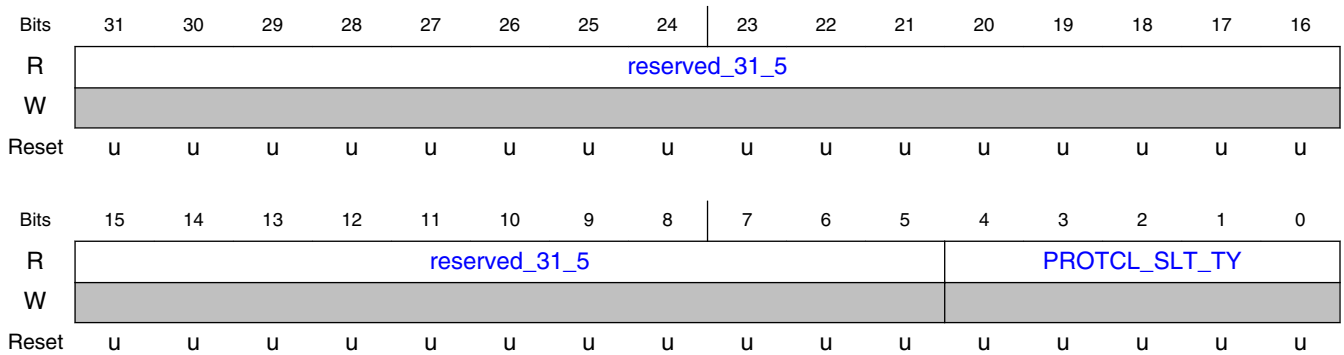
11.1.3.1.45.1 Offset

Register	Offset
SUPTPRT3_DW3	8ACh

11.1.3.1.45.2 Function

Reset Mask:0xFFFFFFE0

11.1.3.1.45.3 Diagram



11.1.3.1.45.4 Fields

Field	Function
31-5 reserved_31_5	Reserved
4-0 PROTCL_SLT_TY	Protocol Slot Type Protocol Slot Type

11.1.3.1.46 Global SoC Bus Configuration Register 0 (GSBUSCFG0)

11.1.3.1.46.1 Offset

Register	Offset
GSBUSCFG0	C100h

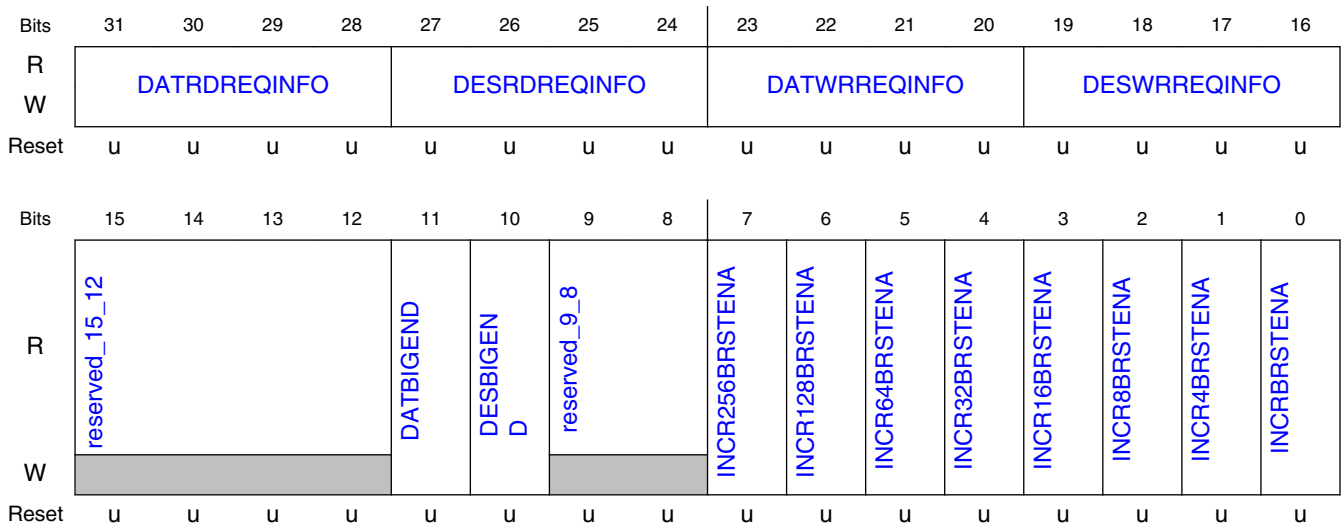
11.1.3.1.46.2 Function

This register configures system bus DMA options for the master bus, which may be configured as AHB, AXI, or Native.

Reset Mask:0xF300

Reset Default Value:0x1

11.1.3.1.46.3 Diagram



11.1.3.1.46.4 Fields

Field	Function
31-28 DATRDREQINFO	DATRDREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Data Read (DatRdReqInfo) Input to BUS-GM.
27-24 DESRDREQINFO	DESRDREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Descriptor Read (DesRdReqInfo). Input to BUS-GM.
23-20 DATWRREQINFO	DATWRREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Data Write (DatWrReqInfo). Input to BUS-GM.
19-16 DESWRREQINFO	DESWRREQINFO AHB-prot/AXI-cache/OCP-ReqInfo for Descriptor Write (DesWrReqInfo) Input to BUS-GM.
15-12 reserved_15_12	Reserved for future use
11 DATBIGEND	Data Access is Big Endian This bit controls the endian mode for data accesses. - Little-endian (default); - Big-endian; In big-endian mode, DMA access (both read and write) for packet data a Byte Invariant Big-Endian mode (see "Little-Endian and Big-Endian" section in the DWC SuperSpeed USB 3.0 Controller User Guide). Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the AMBA AXI Specification. Hence for an AXI master (DWC_USB3_MBUS_TYPE=1), this bit must be set to zero.
10 DESBIGEND	Descriptor Access is Big Endian This bit controls the endian mode for descriptor accesses. - Little-endian (default) - Big-endian In big-endian mode, DMA access (both read and write) for descriptors uses a Byte Invariant Big-Endian mode (see "Little-Endian and Big-Endian" section in the DWC SuperSpeed USB 3.0

Table continues on the next page...

Field	Function
	Controller User Guide. Data is considered as 'embedded data' in the descriptors in the following cases: - Device mode: The buffer pointer of a Setup TRB points to the Setup TRB itself. - Host mode: The Immediate Data (IDT) bit in a Transfer TRB is set to 1. In device mode, if the system uses different endian modes for descriptor and data, software must not use 'embedded' data. In host mode, if the system uses different endian modes for data and descriptors, the controller treats 'embedded data' as descriptor (not as data) in terms of endian mode handling. If this is not the expectation of the system, the software must manipulate the 'embedded data' accordingly. Note: Since AXI requires byte invariant endianness, setting DescBigend and DatBigEnd to one causes an address invariant transform to be applied, which is not appropriate. See section 9.3 and 9.4 of the AMBA AXI Specification. Hence for an AXI master (DWC_USB3_MBUS_TYPE=1), this bit must be set to zero.
9-8 reserved_9_8	Reserved
7 INCR256BRSTENA	INCR256 Burst Type Enable Input to BUS-GM. For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 256-beat burst.
6 INCR128BRSTENA	INCR128 Burst Type Enable Input to BUS-GM; For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 128-beat burst.
5 INCR64BRSTENA	INCR64 Burst Type Enable - Input to BUS-GM; For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 64-beat burst.
4 INCR32BRSTENA	INCR32BRSTENA INCR32 Burst Type Enable Input to BUS-GM; For the AHB/AXI configuration, if software set this bit to 1, the AHB/AXI master uses INCR to do the 32-beat burst.
3 INCR16BRSTENA	INCR16BRSTENA INCR16 Burst Type Enable Input to BUS-GM. For the AHB/AXI configuration, if software set this bit to '1', the AHB/AXI master uses INCR to do the 16-beat burst.
2 INCR8BRSTENA	INCR8BRSTENA INCR8 Burst Type Enable Input to BUS-GM; For the AHB/AXI configuration, if software set this bit to "1", the AHB/AXI master uses INCR to do the 8-beat burst.
1 INCR4BRSTENA	INCR4BRSTENA INCR4 Burst Type Enable Input to BUS-GM; For the AXI configuration, when this bit is enabled the controller is allowed to do bursts of beat length 1, 2, 3, and 4. It is highly recommended that this bit is enabled to prevent descriptor reads and writes from being broken up into separate transfers.
0 INCRBRSTENA	INCRBRSTENA Undefined Length INCR Burst Type Enable (INCRBrstEna) Input to BUS-GM; This bit determines the set of burst lengths the master interface uses. It works in conjunction with the GSBUSCFG0[7:1] enables (INCR256/128/64/32/16/8/4). 0: INCRX burst mode HBURST (for AHB configurations) and ARLEN/ AWLEN (for AXI configurations) do not use INCR except in case of non-aligned burst transfers. In the case of address-aligned transfers, they use only the following burst lengths: - 1 - 4 (if GSBUSCFG0.INCR4BrstEna = 1) - 8 (if GSBUSCFG0.INCR8BrstEna = 1) - 16 (if GSBUSCFG0.INCR16BrstEna = 1) - 32 (if GSBUSCFG0.INCR32BrstEna = 1) - 64 (if GSBUSCFG0.INCR64BrstEna = 1) - 128 (if GSBUSCFG0.INCR128BrstEna = 1) - 256 (if GSBUSCFG0.INCR256BrstEna = 1) Note: - In case of non-address-aligned transfers, INCR may get generated at the beginning and end of the transfers to align the address boundaries, even though INCR is disabled. - In AHB mode, if INCRX burst mode is enabled, but none of the supported INCRx bursts bits are enabled, then the controller will perform (undefined length) INCR bursts. 1: INCR (undefined length) burst mode - AHB configurations: HBURST uses SINGLE or INCR of any length with handling 1KB boundary breakup. - AXI configurations: ARLEN/AWLEN uses any length less than or equal to the

Field	Function
	largest-enabled burst length of INCR32/64/128/256. For cache line-aligned applications, this bit is typically set to 0 to ensure that the master interface uses only power-of-2 burst lengths (as enabled via GSBUSCFG0[7:0]).

11.1.3.1.47 Global SoC Bus Configuration Register 1 (GSBUSCFG1)

11.1.3.1.47.1 Offset

Register	Offset
GSBUSCFG1	C104h

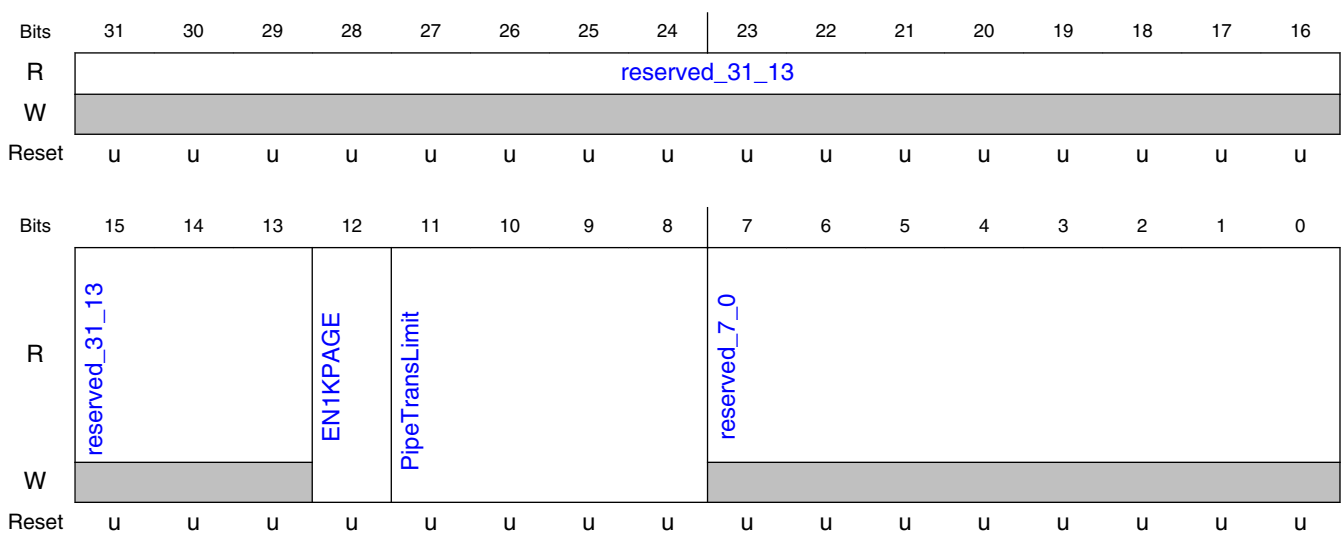
11.1.3.1.47.2 Function

xHCI Register Power-On Value: If you are using a standard xHCI host driver, make sure to set the register's power-on value during coreConsultant configuration (DWC_USB3_GSBUSCFG1_INIT parameter) because the standard xHCI driver does not access this register.

Reset Mask:0xFFFFE0FF

Reset Default Value:0x300

11.1.3.1.47.3 Diagram



11.1.3.1.47.4 Fields

Field	Function
31-13 reserved_31_13	Reserved
12 EN1KPAGE	1k Page Boundary Enable By default (this bit is disabled) the AXI breaks transfers at the 4k page boundary. When this bit is enabled, the AXI master (DMA data) breaks transfers at the 1k page boundary.
11-8 PipeTransLimit	AXI Pipelined Transfers Burst Request Limit The field controls the number of outstanding pipelined transfer requests the AXI master pushes to the AXI slave. When the AXI master reaches this limit, it does not make any more requests on the AXI ARADDR and AWADDR buses until the associated data phases complete. This field is encoded as follows: - 'h0: 1 request - 'h1: 2 requests - 'h2: 3 requests - 'h3: 4 requests - ... - 'hF: 16 requests
7-0 reserved_7_0	Reserved

11.1.3.1.48 Global Tx Threshold Control Register (GTXTHRCFG)

11.1.3.1.48.1 Offset

Register	Offset
GTXTHRCFG	C108h

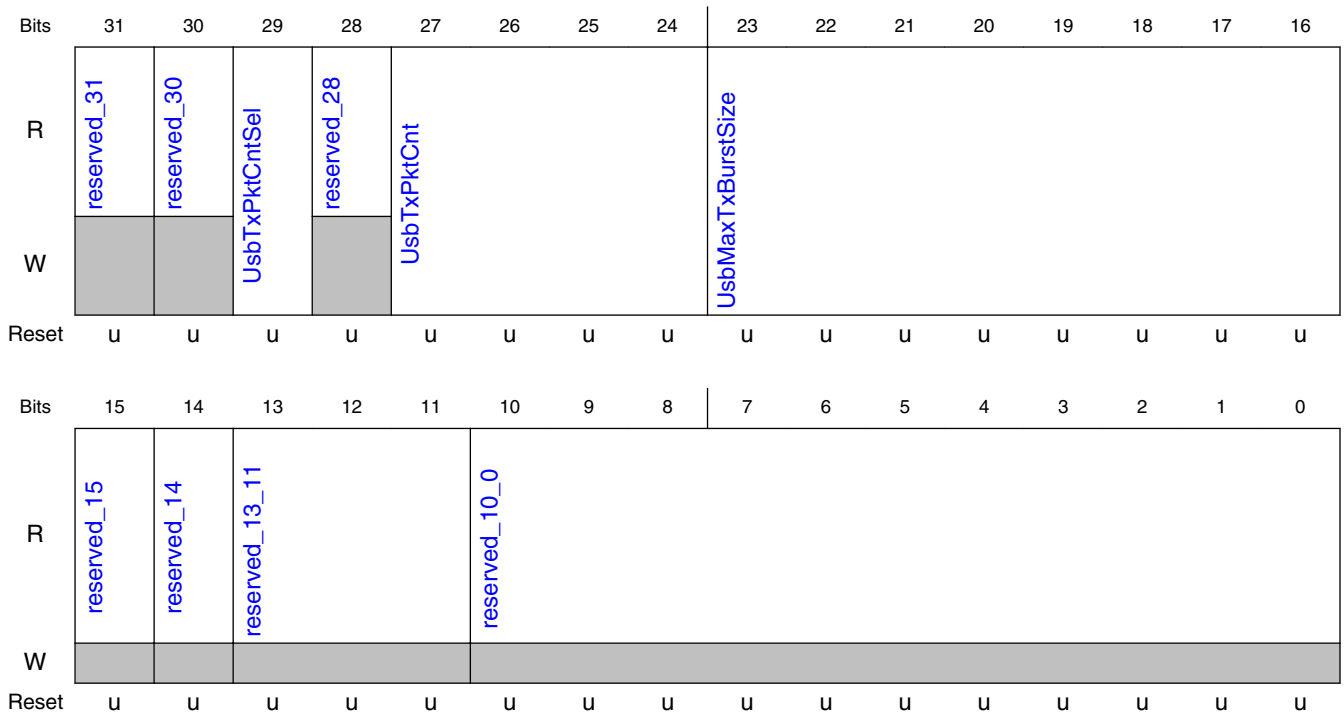
11.1.3.1.48.2 Function

All bits in the GTXTHRCFG register are valid only in host mode. This register is not applicable for debug target and USB 2.0-only mode.

Reset Mask:0xD000FFFF

Reset Default Value:0x0

11.1.3.1.48.3 Diagram



11.1.3.1.48.4 Fields

Field	Function
31 reserved_31	Reserved
30 reserved_30	Reserved
29 UsbTxPktCntSel	USB Transmit Packet Count Enable This field enables/disables the USB transmission multi-packet thresholding: - 0: USB transmission multi-packet thresholding is disabled; the core can start transmission on the USB after the entire (one full) packet has been fetched into the corresponding TXFIFO. - 1: USB transmission multi-packet thresholding is enabled. The core can only start transmission on the USB after USB Transmit Packet Count amount of packets for the USB transaction (burst) are already in the corresponding TXFIFO. This mode is valid in both host and device modes. It is only used for SuperSpeed operation.
28 reserved_28	Reserved
27-24 UsbTxPktCnt	USB Transmit Packet Count This field specifies the number of packets that must be in the TXFIFO before the core can start transmission for the corresponding USB transaction (burst). This field is only valid when the USB Transmit Packet Count Enable field is set to one. Valid values are from 1 to 15. Note: - In device mode, if device controller does not have the TRBs for the number of packets or if it cannot fetch the TRBs because of high latency or switching between other endpoints, then it does not wait for the threshold number of packets. The threshold number of packets will be honored only when the TRBs are available in the controller for the number of packets before it starts the data fetch. - This field must be less than or equal to the USB Maximum TX Burst Size field.

Table continues on the next page...

Field	Function
23-16 UsbMaxTxBurstSize	USB Maximum TX Burst Size When UsbTxPktCntSel is one, this field specifies the Maximum Bulk OUT burst the core can do. When the system bus is slower than the USB, TX FIFO can underrun during a long burst. User can program a smaller value to this field to limit the TX burst size that the core can do. Host mode: It only applies to SS Bulk, Isochronous, and Interrupt OUT endpoints. Device mode: This value is not used in device mode, but users need to program a value when using the TX threshold feature to make sure that the value programmed in UsbTxPktCnt is less than this value. Valid values are from 1 to 16.
15 reserved_15	Reserved_15
14 reserved_14	Reserved1(Rsvd/Rs) Register field must write only 0 by the application. The read value must be treated as X (unknown).
13-11 reserved_13_11	Reserved (Rsvd/Rs) The register field must write only 0 by the application. The read value must be treated as X (unknown).
10-0 reserved_10_0	Reserved for future use

11.1.3.1.49 Global Rx Threshold Control Register (GRXTHRCFG)

11.1.3.1.49.1 Offset

Register	Offset
GRXTHRCFG	C10Ch

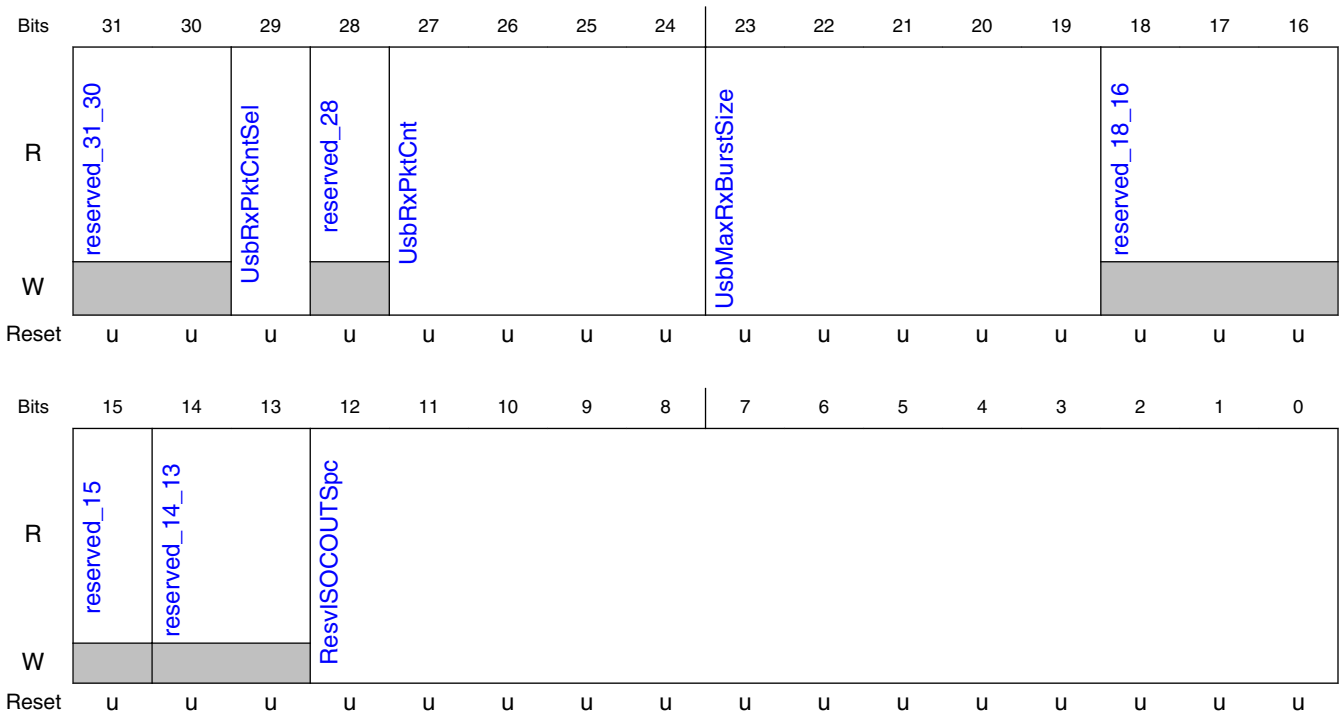
11.1.3.1.49.2 Function

This register is not applicable for debug target and USB 2.0-only mode. In a normal case, a Tx burst starts as soon as one packet is prefetched; an Rx burst starts as soon as 1-packet space is available.

Reset Mask:0xD007E000

Reset Default Value:0x0

11.1.3.1.49.3 Diagram



11.1.3.1.49.4 Fields

Field	Function
31-30 reserved_31_30	Reserved
29 UsbRxPktCntSel	USB Receive Packet Count Enable This field enables/disables the USB reception multi-packet thresholding: - 0: The core can only start reception on the USB when the RX FIFO has space for at least one packet. - 1: The core can only start reception on the USB when the RX FIFO has space for at least UsbRxPktCnt amount of packets. This mode is valid in both host and device mode. It is only used for SuperSpeed. In device mode, - Setting this bit to 1 also enables the functionality of reporting NUMP in the ACK TP based on the RX FIFO space instead of reporting a fixed NUMP derived from DCFG.NUMP for non-control endpoints. - If you are using external buffer control (EBC) feature, disable this mode by setting UsbRxPktCntSel to 0.
28 reserved_28	Reserved
27-24 UsbRxPktCnt	USB Receive Packet Count In host mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can start the corresponding USB RX transaction (burst). In device mode, this field specifies the space (in terms of the number of packets) that must be available in the RX FIFO before the core can send ERDY for a flow-controlled endpoint. This field is valid only when the USB Receive Packet Count Enable field is set to 1. The valid values for this field are from 1 to 15. Note: This field must be less than or equal to the USB Maximum Receive Burst Size field.
23-19	USB Maximum Receive Burst Size In host mode, this field specifies the Maximum Bulk IN burst the DWC_usb3 core can perform. When the system bus is slower than the USB, RX FIFO can overrun during a long burst. You can program a smaller value to this field to limit the RX burst size that the core can

Table continues on the next page...

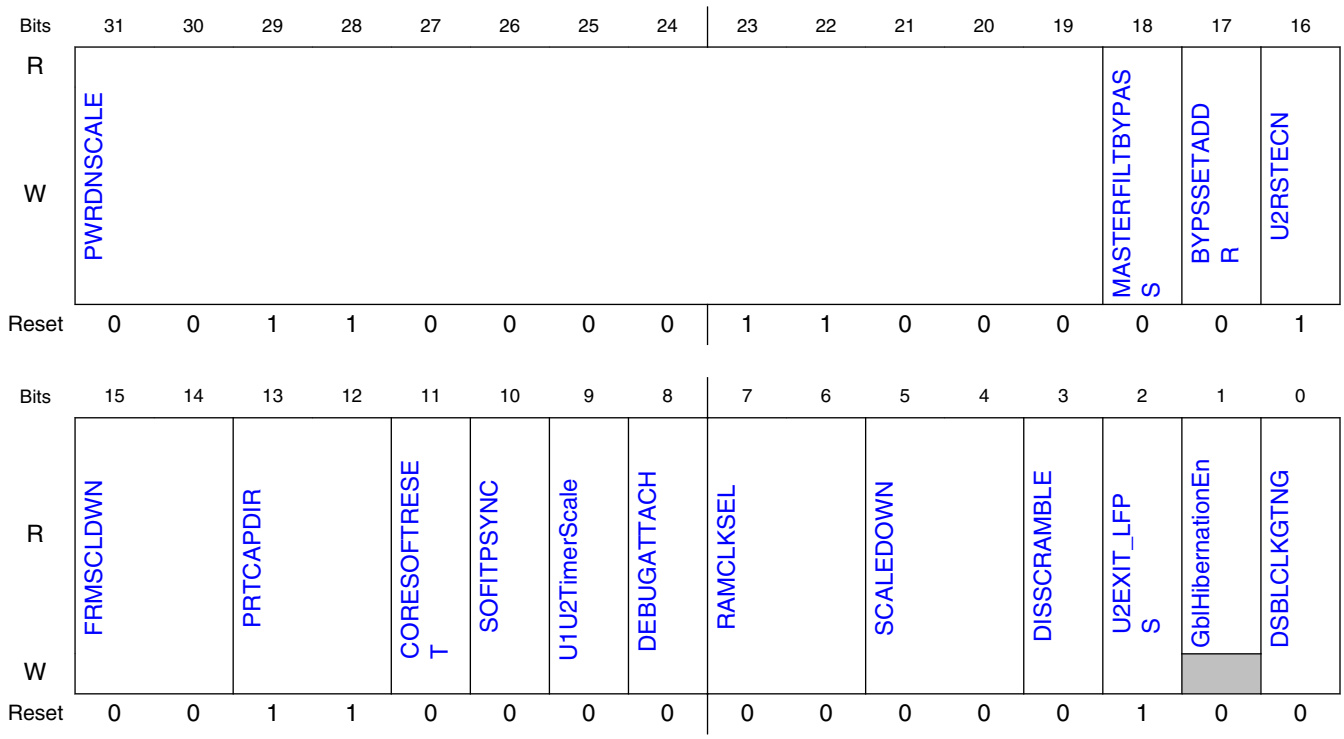
Field	Function
UsbMaxRxBurstSize	perform. It only applies to SS Bulk, Isochronous, and Interrupt IN endpoints in the host mode. In device mode, this field specifies the NUMP value that is sent in ERDY for an OUT endpoint. The programmed value should not exceed the RXFIFO size. This field is valid only when UsbRxPktCntSel is one. The valid values for this field are from 1 to 16.
18-16 reserved_18_16	Reserved
15 reserved_15	Reserved
14-13 reserved_14_13	Reserved
12-0 ResvISOCOUTSpc	Space reserved in Rx FIFO for ISOC OUT In host mode, this field is not applicable and must be programmed to 0. In device mode, this value represents the amount of space to be reserved for ISOC OUT packets. The value to be programmed should be chosen so as to ensure that non ISOC packets are not completely dropped. If no space needs to be reserved for ISOC OUT packets, program this field to 0. This field is valid only in device mode. The maximum configurable depth of RX FIFO is 8192. Therefore, this field is 13 bits wide. The value of space reserved is in terms of DWC_USB3_MDWIDTH. For SS, the space reservation is always rounded off to the nearest packet boundary. Therefore, it is always recommended to program a value corresponding to MPS or its multiples. For HS/FS, the space reservation is the actual value. Note: For SS, reserve space for ISOC when the Rx FIFO space can accommodate two MPS or more. Otherwise, this may result in degraded performance for non-ISOC packets. If the space is entirely allocated for ISOC, the non-ISOC packets will be completely dropped. To help you decide during the time of configuring the core, refer to the section "Device-Mode Receive Path" in Chapter 3.

11.1.3.1.50 Global Core Control Register (GCTL)

11.1.3.1.50.1 Offset

Register	Offset
GCTL	C110h

11.1.3.1.50.2 Diagram



11.1.3.1.50.3 Fields

Field	Function
31-19 PWRDNSCALE	Power Down Scale (PwrDnScale) The USB3 suspend_clk input replaces pipe3_rx_pclk as a clock source to a small part of the USB3 core that operates when the SS PHY is in its lowest power (P3) state, and therefore does not provide a clock. The Power Down Scale field specifies how many suspend_clk periods fit into a 16 kHz clock period. When performing the division, round up the remainder. For example, when using an 8-bit/16-bit/32-bit PHY and 25-MHz Suspend clock, Power Down Scale = 25000 kHz/16 kHz = 13'd1563 (rounder up) Note: - Minimum Suspend clock frequency is 32 kHz - Maximum Suspend clock frequency is 125 MHz The LTSSM uses Suspend clock for 12-ms and 100-ms timers during suspend mode. According to the USB 3.0 specification, the accuracy on these timers is 0% to +50%. - 12 ms + 0~+50% accuracy = 18 ms (Range is 12 ms - 18 ms) - 100 ms + 0~+50% accuracy = 150 ms (Range is 100 ms - 150 ms). The suspend clock accuracy requirement is: - (12,000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 12,000 and 18,000 - (100,0000/62.5) * (GCTL[31:19]) * actual suspend_clk_period must be between 100,000 and 150,000 For example, if your suspend_clk frequency varies from 7.5 MHz to 10.5MHz, then the value needs to programmed is: Power Down Scale = 10500/16 = 657 (rounded up; and fastest frequency used).
18 MASTERFILTBYPASS	Master Filter Bypass When this bit is set to 1'b1, irrespective of the parameter `DWC_USB3_EN_BUS_FILTERS chosen, all the filters in the DWC_usb3_filter module are bypassed. The double synchronizers to mac_clk preceding the filters are also bypassed. For enabling the filters, this bit must be 1'b0.
17 BYPSSETADDR	Bypass SetAddress in Device Mode. When BYPSSETADDR bit is set, the device core uses the value in the DCFG[DevAddr] bits directly for comparing the device address in the tokens. For simulation, you can use this feature to avoid sending an actual SET ADDRESS control transfer on the USB, and make the device core respond to a new address. When the xHCI Debug capability is enabled and this bit is set, the Debug Target immediately enters the configured state without requiring the Debug Host to send a

Table continues on the next page...

Field	Function
	SetAddress or SetConfig request. Note: You can set this bit for simulation purposes only. In the actual hardware, this bit must be set to 1'b0.
16 U2RSTECN	U2RSTECN If the SuperSpeed connection fails during POLL or LMP exchange, the device connects at non-SS mode. If this bit is set, then device attempts three more times to connect at SS, even if it previously failed to operate in SS mode. For each attempt, the device checks receiver termination eight times. From 2.60a release, this bit controls whether to check for Rx.Detect eight times or one time for every attempt. Device controller on USB 2.0 reset checks for receiver termination eight times per attempt if this bit is set to zero, or only once per attempt if the bit is set to one. Note: This bit is applicable only in device mode.
15-14 FRMSCLDWN	FRMSCLDWN This field scales down device view of a SOF/USOF/ITP duration. For SS/HS mode: - Value of 2'h3 implements interval to be 15.625 us - Value of 2'h2 implements interval to be 31.25 us - Value of 2'h1 implements interval to be 62.5 us - Value of 2'h0 implements interval to be 125us For FS mode, the scale-down value is multiplied by 8. When xHCI Debug Capability is enabled, this field also scales down the MaxPacketSize of the IN and OUT bulk endpoint to allow more traffic during simulation. It can only be changed from a non-zero value during simulation. - 2'h0: 1024 bytes - 2'h1: 512 bytes - 2'h2: 256 bytes - 2'h3: 128 bytes
13-12 PRTCAPDIR	PRTCAPDIR: Port Capability Direction (PrtCapDir) - 2'b01: for Host configurations - 2'b10: for Device configurations - 2'b11: for OTG configurations When DWC_USB3_EN_OTG is 0, then the core acts as a DRD. When DWC_USB3_EN_OTG is 1, - If PrtCapDir is 2'b11, it acts as an OTG 2.0 device with A-device or B-device determined by the IDDIG input, and Host or Peripheral role based on HNP. - If PrtCapDir is 2'b01, it acts as a DRD in host mode. - If PrtCapDir is 2'b10, it acts as a DRD in device mode. The OTG device can be programmed to enable/disable SRP and HNP by using the fields present in OCFG register. When DWC_USB3_EN_OTG is 2, - If PrtCapDir is 2'b11, it acts as an OTG 3.0 device with A-device or B-device determined by the IDDIG input, and Host or Peripheral role based on HNP/ RSP. - If PrtCapDir is 2'b01, it acts as a DRD in host mode. In this mode, the VBUS is driven immediately after power-on reset. When the DWC_usb3 controller is plugged into a PC and the application configures the device later on, VBUS is driven on the USB from both sides of the cable. This will potentially damage the PC host. - If PrtCapDir is 2'b10, it acts as a DRD in device mode. The OTG device can be programmed to enable/disable SRP and HNP/RSP using the fields present in the OCFG register. Note: For static Host-only/Device-only applications, use DRD Host or DRD Device mode. The combination of GCTL.PrtCapDir=2'b11 with SRP and HNP/RSP disabled is not recommended for these applications. The sequence for switching modes in DRD configuration is as follows: Switching from Device to Host: 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b01 (Host mode). 3. Reset the host using USBCMD.HCRESET. 4. Follow the steps in "Initializing Host Registers" section of the databook. Switching from Host to Device: 1. Reset the controller using GCTL[11] (CoreSoftReset). 2. Set GCTL[13:12] (PrtCapDir) to 2'b10 (Device mode). 3. Reset the device by setting DCTL[30] (CSftRst). 4. Follow the steps in "Register Initialization" section of the databook. Programming this field with random data causes the core to keep toggling between the host mode and the device mode . Bit Bash register testing is not recommended.
11 CORESOFTRESET	Core Soft Reset (CoreSoftReset) - 1'b0 - No soft reset - 1'b1 - Soft reset to core Clears the interrupts and all the CSRs except the following registers: - GCTL - GUCTL - GSTS - GSNPSID - GGPIO - GUID - GUSB2PHYCFGn registers - GUSB3PIPECTLn registers - DCFG - DCTL - DEVTEN - DSTS When you reset PHYs (using GUBS3PHYCFG or GUSB3PIPECTL registers), you must keep the core in reset state until PHY clocks are stable. This controls the bus, ram, and mac domain resets. Refer to the "Reset Generation" section in the Architecture Details chapter in the DesignWare Cores SuperSpeed USB 3.0 Controller Databook. Note: This bit is for debug purposes only. Use USBCMD.HCRESET in xHCI Mode and DCTL.SoftReset in device mode for soft reset. Programming this field with random data will reset the internal logic of the host controller. Due to this side effect Bit Bash register testing is not recommended.
10 SOFITPSYNC	SOFITPSYNC If this bit is set to '0' operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever there is a SuperSpeed port that is not in Rx.Detect, SS.Disable and U3. If this bit is set to '1' operating in host mode, the core keeps the UTMI/ULPI PHY on the first port in a non-suspended state whenever the other non-SuperSpeed ports are not in a suspended state. This feature is useful because it saves power by suspending UTMI/ULPI when SuperSpeed only is active, and it helps resolve when the PHY does not transmit a host resume unless it is placed in suspend state. This bit must be programmed as a part of initialization at power-on reset, and must not be

Table continues on the next page...

Field	Function
	dynamically changed afterwards. Note: - USB2PHYCFGn[6].PhySusp eventually decides to put the UTMI/ULPI PHY in to suspend state. In addition, when this bit is set to '1', the core generates ITP from the ref_clk based counter. Otherwise, ITP and SOF are generated from utmi/ulpi_clk[0] based counter. To program the reference clock period inside the core, refer to GUCTL[31:22].REFCLKPER. - This feature is valid in Host and DRD/OTG configurations and used only in Host mode operation. - If you never use this feature or the GFLADJ.GFLADJ_REFCLK_LPM_SEL, the minimum frequency for the ref_clk can be as low as 32KHz. You can connect the suspend_clk (as low as 32 KHz) to the ref_clk. - If you plan to enable hardware-based LPM or software-based LPM (PORTPMSC. HLE=1), then you cannot use this feature. Turn off this feature by setting this bit to '0' and use the GFLADJ.GFLADJ_REFCLK_LPM_SEL feature. - If you set this bit to '1', the GUSB2PHYCFG.U2_FREECLK_EXISTS bit and the DWC_USB3_FREECLK_USB2_EXIST parameter must be set to '0'. Program this bit to 0 if the core is intended to be operated in USB 3.0 mode.
9 U1U2TimerScale	Disable U1/U2 timer Scaledown (U1U2TimerScale). If set to '1' along with GCTL[5:4] (ScaleDown) = 2'bX1, disables the scale down of U1/U2 inactive timer values. This is for simulation mode only.
8 DEBUGATTACH	Debug Attach When this bit is set, - SS Link proceeds directly to the Polling link state (after RUN/STOP in the DCTL register is asserted) without checking remote termination; - Link LFPS polling timeout is infinite; - Polling timeout during TS1 is infinite (in case link is waiting for TXEQ to finish).
7-6 RAMCLKSEL	RAM Clock Select (RAMClkSel) - 2'b00: bus clock - 2'b01: pipe clock (Only used in device mode) - 2'b10: In device mode , pipe/2 clock.In Host mode, controller switches ram_clk between pipe/2 clock, mac2_clk and bus_clk based on the status of the U2/U3 ports - 2'b11: In device mode, selects mac2_clk as ram_clk (when 8-bit UTMI or ULPI used. Not supported in 16-bit UTMI mode) In Host mode, controller switches ram_clk between pipe_clk, mac2_clk and bus_clk based on the status of the U2/U3 ports. In device mode, upon a USB reset and USB disconnect, the hardware clears these bits to 2'b00. For more information on how to select the RAM clock, see "Clock Generation and Clock Tree Synthesis (CTS) Requirements" in the Architectural Details chapter in the DesignWare Cores SuperSpeed USB 3.0 Controller Databook. Note: - In device mode, if you set RAMClkSel to 2'b11 (mac2_clk), the controller internally switches the ram_clk to bus_clk when the link state changes to Suspend (L2 or L3), and switches the ram_clk back to mac2_clk when the link state changes to resume or U2. - In host mode, if a value of 2/3 is chosen, then controller switches ram_clk between bus_clk, mac2_clk and pipe_clk, pipe_clk/2, based on the state of the U2/U3 ports. For example if only the U2 port is active and the U3 ports are suspended, then the ram_clk is swicthed to mac2_clk. When only the U3 ports are active and the U2 ports are suspended, then the core internally swicth the ram_clk to pipe3 clock and when all U2 and U3 ports are suspended, it switch the ram_clk to bus_clk. This allows de coupling the ram_clk from the bus_clk and depending on the bandwidth requiredmnet allows the bus_clk to be run at a lower frequency than the ram_clk requirements. bus_clk frequency still cannot go below 60Mhz in host mode, and this is not verified. A value of 2 can be chosen only if the pipe data width is 8 or 16 bits. In this case the when the ram_clk is switched to pipe_clk, it uses pipe_clk/2 instead of pipe_clk. If a value of 3 is chosen for RAMClkSel, then when ram_clk is switched to pipe_clk, then pipe_clk is used without any divider. - In device mode, when RAMClkSel != 2'b00, the bus_clk_early frequency can be a minimum of 1 MHz. This is tested in simulation and also in hardware with Linux, Microsoft Windows 8, and MCCI Windows7 host drivers. Only control and non periodic transfers are supported when bus_clk is 1 MHz. For periodic applications, the bus_clk_early minimum frequency is higher depending on your application and SoC bus. Even though 1 MHz has been tested with standard host drivers, Synopsys recommends 5 MHz minimum for ASIC designs to provide a margin or at least have a backup option to increase the bus_clk frequency to 5 MHz if needed. Programming this field with random data will cause side effect. Bit Bash register testing is not recommended.
5-4 SCALEDOWN	Scale-Down Mode (ScaleDown) When Scale-Down mode is enabled for simulation, the core uses scaled-down timing values, resulting in faster simulations. When Scale-Down mode is disabled, actual timing values are used. This is required for hardware operation. HS/FS/LS Modes - 2'b00: Disables all scale-downs. Actual timing values are used. - 2'b01: Enables scale-down of all timing values except Device mode suspend and resume. These include Speed enumeration, HNP/SRP, and Host mode suspend and resume - 2'b10: Enables scale-down of Device mode suspend and resume timing values only. - 2'b11: Enables bit 0 and bit 1 scale-down timing values. SS Mode - 2'b00: Disables all scale-downs. Actual timing values are used. - 2'b01: Enables scaled down SS timing and repeat values including: (1) Number

Table continues on the next page...

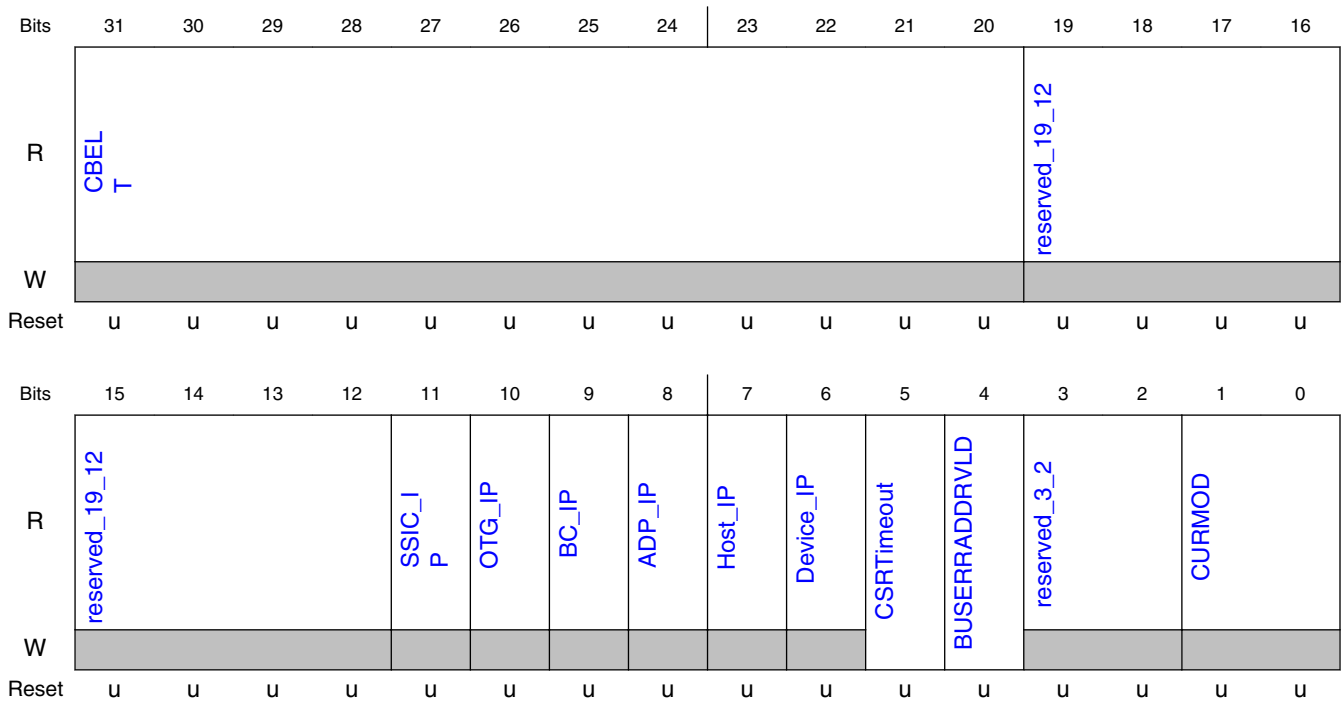
Field	Function
	of TxEq training sequences reduce to 8; (2) LFPS polling burst time reduce to 256 nS; (3) LFPS warm reset receive reduce to 30 uS. Refer to the rtl_vip_scaledown_mapping.xls file under <workspace>/sim/SoC_sim directory for the complete list. - 2'b10: No TxEq training sequences are sent. Overrides Bit 4. - 2'b11: Enables bit 0 and bit 1 scale-down timing values.
3 DISSCRAMBLE	Disable Scrambling (DisScramble) Transmit request to Link Partner on next transition to Recovery or Polling.
2 U2EXIT_LFPS	U2EXIT_LFPS If this bit is, - 0: the link treats 248ns LFPS as a valid U2 exit. - 1: the link waits for 8us of LFPS before it detects a valid U2 exit. This bit is added to improve interoperability with a third-party host/device controller. This host/device controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the host/device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the host/device can stay in U2 while ignoring this glitch from the host/device controller. This bit is applicable for both host and device controller. This bit is added to improve interoperability with a third party host controller. This host controller in U2 state while performing receiver detection generates an LFPS glitch of about 4ms duration. This causes the device to exit from U2 state because the LFPS filter value is 248ns. With the new functionality enabled, the device can stay in U2 while ignoring this glitch from the host controller.
1 GblHibernationEn	GblHibernationEn This bit enables hibernation at the global level. If hibernation is not enabled through this bit, the PMU immediately accepts the D0->D3 and D3->D0 power state change requests, but does not save or restore any core state. In addition, the PMUs never drive the PHY interfaces and let the core continue to drive the PHY interfaces.
0 DSBLCLKGTNG	Disable Clock Gating (DsblClkGtng) This bit is set to 1 and the core is in Low Power mode, internal clock gating is disabled. You can set this bit to 1'b1 after Power On Reset.

11.1.3.1.51 Global Status Register (GSTS)

11.1.3.1.51.1 Offset

Register	Offset
GSTS	C118h

11.1.3.1.51.2 Diagram



11.1.3.1.51.3 Fields

Field	Function
31-20 CBELT	Current BELT Value In Host mode, this field indicates the minimum value of all received device BELT values and the BELT value that is set by the Set Latency Tolerance Value command.
19-12 reserved_19_12	Reserved
11 SSIC_IP	SSIC interrupt pending (SSIC_IP) This field indicates that there is a pending interrupt related to SSIC in the SEVT register. Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is reserved.
10 OTG_IP	OTG Interrupt Pending This field indicates that there is a pending interrupt pertaining to OTG in OEVT register.
9 BC_IP	Battery Charger Interrupt Pending This field indicates that there is a pending interrupt pertaining to BC in BCEVT register.
8 ADP_IP	ADP Interrupt Pending This field indicates that there is a pending interrupt pertaining to ADP in ADPEVT register.
7 Host_IP	Host Interrupt Pending: This field indicates that there is a pending interrupt pertaining to xHC in the Host event queue.
6 Device_IP	Device Interrupt Pending This field indicates that there is a pending interrupt pertaining to peripheral (device) operation in the Device event queue.

Table continues on the next page...

Field	Function
5 CSRTimeout	CSR Timeout When this bit is 1'b1, it indicates that the software performed a write or read to a core register that could not be completed within `DWC_USB3_CSR_ACCESS_TIMEOUT bus clock cycles (default: h1FFFF).
4 BUSERRADDRVLD	Bus Error Address Valid (BusErrAddrVld) Indicates that the GBUSERRADDR register is valid and reports the first bus address that encounters a bus error. Note: Only supported in AHB and AXI configurations.
3-2 reserved_3_2	Reserved
1-0 CURMOD	Current Mode of Operation (CurMod) Indicates the current mode of operation: - 2'b00: Device mode - 2'b01: Host mode

11.1.3.1.52 (GUCTL1)

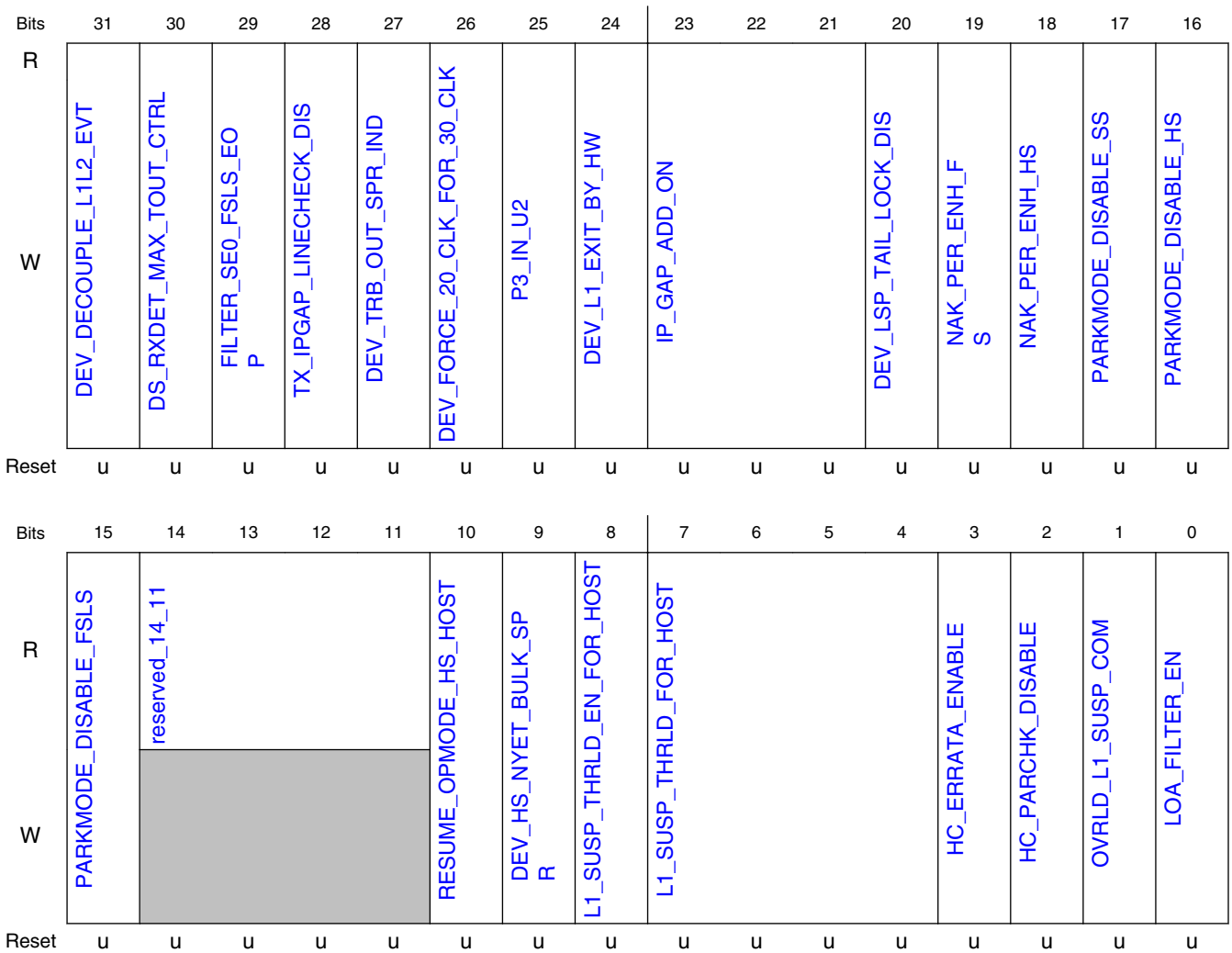
11.1.3.1.52.1 Offset

Register	Offset
GUCTL1	C11Ch

11.1.3.1.52.2 Function

Global User Control Register 1

11.1.3.1.52.3 Diagram



11.1.3.1.52.4 Fields

Field	Function
31 DEV_DECOUPLE_L1L2_EVT	DEV_DECOUPLE_L1L2_EVT - 0: Default behavior, no change in device events L1/L2U3 events are not decoupled (old behavior of v2.90a and before) - 1: Feature enabled, L1 and L2 events are separated when operating in 2.0 mode. Separate event enable bits for L1 suspend and wake events. This bit is applicable for device mode only. If this feature is enabled, L1 suspend and wake events have individual controls to enable/mask them. Enable this feature if you want to get L1 (LPM) events separtely and not combined with L2 events when operating in 2.0 speeds.
30 DS_RXDET_MAX_TOUT_CTRL	DS_RXDET_MAX_TOUT_CTRL This bit is used to control the tRxDetectTimeoutDFP timer for the SuperSpeed link. - 0: Default behavior; 12ms is used as tRxDetectTimeoutDFP. - 1: 120ms is used as the tRxDetectTimeoutDFP. This bit is used only in host mode. For more details, refer to ECN020 for USB 3.0 Specification.
29	FILTER_SE0_FSLS_EOP - 0: Default behavior, no change in Linestate check for SE0 detection in FS/LS - 1: Feature enabled, FS/LS SE0 is filtered for 2 clocks for detecting EOP This bit is applicable for FS/LS operation. If this feature is enabled, then SE0 on the linestate is validated for 2 consecutive utmi/ulpi

Table continues on the next page...

Field	Function
FILTER_SE0_FSLS_EOP	clock edges for EOP detection. This feature is applicable only in FS in device mode and FS/LS mode of operation in host mode. Device mode: FS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then for device LPM handshake, the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS. Host mode: FS/LS - If GUCTL1.FILTER_SE0_FSLS_EOP is set, then the core will ignore single SE0 glitch on the linestate during transmit. Only 2 or more SE0 is considered as a valid EOP on FS/LS port. Enable this feature if the LineState has SE0 glitches during transmission. This bit is quasi-static, that is, it must not be changed during device operation.
28 TX_IPGAP_LINECHECK_DIS	TX_IPGAP_LINECHECK_DIS - 0: Default behavior, no change in Linestate check - 1: Feature enabled, 2.0 MAC disables Linestate check during HS transmit This bit is applicable for HS operation of u2mac. If this feature is enabled, then the 2.0 mac operating in HS ignores the UTMI/ULPI Linestate during the transmit of a token (during token-to-token and token-to-data IPGAP). When enabled, the controller implements a fixed 40-bit TxEndDelay after the packet is given on UTMI and ignores the Linestate during this time. This feature is applicable only in HS mode of operation. Device mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then for device LPM handshake, the core will ignore the linestate after TX and wait for fixed clocks (40 bit times equivalent) after transmitting ACK on utmi. Host mode: If GUCTL1.TX_IPGAP_LINECHECK_DIS is set, then the ipgap between (tkn to tkn/data) is added by 40 bit times of TXENDDELAY, and linestate is ignored during this 40 bit times delay. Enable this bit if the LineState will not reflect the expected line state (J) during transmission. This bit is quasi-static, that is, it must not be changed during device operation.
27 DEV_TRB_OUT_SPR_IND	DEV_TRB_OUT_SPR_IND - 0: Default behavior, no change in TRB status dword - 1: Feature enabled, OUT TRB status indicates Short Packet This bit is applicable for device mode only (and ignored in host mode). If the device application (software/hardware) wants to know if a short packet was received for an OUT in the TRB status itself, then this feature can be enabled, so that a bit is set in the TRB writeback in the buf_size dword. Bit[26] - SPR of the {trbstatus, RSVD, SPR, PCM1, bufsize} dword will be set during an OUT transfer TRB write back if this is the last TRB used for that transfer descriptor. This bit is quasi-static, that is, it must not be changed during device operation.
26 DEV_FORCE_20_CLK_FOR_30_CLK	DEV_FORCE_20_CLK_FOR_30_CLK - 0: Default behavior, Uses 3.0 clock when operating in 2.0 mode - 1: Feature enabled This bit is applicable (and to be set) for device mode (DCFG.Speed != SS) only. In the 3.0 device core, if the core is programmed to operate in 2.0 only (that is, Device Speed is programmed to 2.0 speeds in DCFG[Speed]), then setting this bit makes the internal 2.0 (utmi/ulpi) clock to be routed as the 3.0 (pipe) clock. Enabling this feature allows the pipe3 clock to be not-running when forcibly operating in 2.0 device mode. Note: - When using this feature, all pipe3 inputs must be in inactive mode. In particular, the pipe3 clocks must not be running and the pipe3_phystatus_async must be tied to 0. This bit should not be set if the core is programmed to operate in SuperSpeed mode (even when it falls back to 2.0). - This bit is quasi-static, that is, it must not be changed during operation. - If the parameter "DWC_USB3_REMOVE_PIPE_CLK_MUX_FOR_20_MODE" is enabled, then muxing 2.0 clock/signals to pipe_clk/signals have to be done outside the controller for this feature to work.
25 P3_IN_U2	P3_IN_U2 - 0: Default behavior, When SuperSpeed link is in U2, PowerState P2 is attempted on the PIPE Interface. - 1: When SuperSpeed link is in U2, PowerState P3 is attempted if GUSB3PIPECTL[17] is set. Setting this bit enables P3 Power State when the SuperSpeed link is in U2. Another Power Saving option. Check with your PHY vendor before enabling this option.When setting this bit to 1 to enable P3 in P2, GUSB3PIPECTL[27] should be set to 0 to make sure that the U2 exit is attempted in P0. This bit should be set only when GCTL.SOFITPSYNC=1 or GFLADJ.GFLADJ_REFCLK_LPM_SEL=1.
24 DEV_L1_EXIT_BY_HW	DEV_L1_EXIT_BY_HW - 0: Default behavior, disables device L1 hardware exit logic - 1: feature enabled This bit is applicable for device mode (2.0) only. This field enables device controller sending remote wakeup for L1 if the device becomes ready for sending/accepting data when in L1 state. If the host expects the device to send remote wakeup signaling to resume after going into L1 in flow controlled state, then this bit can be set to send the remote wake signal automatically when the device controller becomes ready. This hardware remote wake feature is applicable only to bulk and interrupt transfers, and not for Isoch/Control - When control transfers are in progress, the LPM will be rejected (NYET response). Only after control transfers are completed (either with ACK/STALL), LPM will be accepted - For Isoch transfers, the host needs to do the wake-up and start the transfer. Device controller will not do remote-wakeup when Isoch endpoints get ready. The device SW needs to keep the GUSB2PHYCFG[EnblSlpM] reset in order to keep the PHY clock to be running for keeping track of SOF intervals. - When L1

Table continues on the next page...

Field	Function
	hibernation is enabled, the controller will not do automatic exit for hibernation requests thru L1. This bit is quasi-static, that is, it must not be changed during device operation.
23-21 IP_GAP_ADD_ON	This register field is used to add on to the default inter packet gap setting in the USB 2.0 MAC. This should be programmed to a non zero value only in case where you need to increase the default inter packet delay calculations in the USB 2.0 MAC module DWC_usb3_u2mac.v
20 DEV_LSP_TAIL_LOCK_DIS	DEV_LSP_TAIL_LOCK_DIS - 0: Default behavior, enables device lsp lock logic for tail TRB update - 1: Fix disabled This is a bug fix for STAR 9000716195 that affects the CSP mode for OUT endpoints in device mode. The issue is that tail TRB index is not synchronized with the cache Scratchpad bytecount update. If the fast-forward request comes in-between the bytecount update on a newly fetched TRB and the tail-index write update in TPF, the RDP works on an incorrect tail index and misses the byte count decrement for the newly fetched TRB in the fast-forwarding process. This fix needs to be present all the times.
19 NAK_PER_ENH_FS	NAK_PER_ENH_FS - 1: Enables performance enhancement for FS async endpoints in the presence of NAKs - 0: Enhancement not applied If a periodic endpoint is present , and if a bulk endpoint which is also active is being NAKed by the device, then this could result in a decrease in performance of other Full Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your FullSpeed application. Setting this bit will only control, and is only required for Full Speed transfers.
18 NAK_PER_ENH_HS	NAK_PER_ENH_HS - 1: Enables performance enhancement for HS async endpoints in the presence of NAKs - 0: Enhancement not applied If a periodic endpoint is present, and if a bulk endpoint which is also active is being NAKed by the device, then this could result in decrease in performance of other High Speed bulk endpoint which is ACked by the device. Setting this bit to 1, will enable the host controller to schedule more transactions to the async endpoints (bulk/ control) and hence will improve the performance of the bulk endpoint. This control bit should be enabled only if the existing performance with the default setting is not sufficient for your HighSpeed application. Setting this bit will only control, and is only required for High Speed transfers.
17 PARKMODE_DISABLE_SS	PARKMODE_DISABLE_SS This bit is used only in host mode, and is for debug purpose only. When this bit is set to '1' all SS bus instances in park mode are disabled.
16 PARKMODE_DISABLE_HS	PARKMODE_DISABLE_HS This bit is used only in host mode. When this bit is set to '1' all HS bus instances park mode are disabled. To improve performance in park mode, the xHCI scheduler queues in three requests of 4 packets each for High Speed asynchronous endpoints in a micro-frame. But if a device is slow and if it NAKs more than 3 times, then it is rescheduled only in the next micro-frame. This could decrease the performance of a slow device even further. In a few high speed devices (such as Sandisk Cruzer Blade 4GB VID:1921, PID:21863 and Flex Drive VID:3744, PID:8552) when an IN request is sent within 900ns of the ACK of the previous packet, these devices send a NAK. When connected to these devices, if required, the software can disable the park mode if you see performance drop in your system. When park mode is disabled, pipelining of multiple packet is disabled and instead one packet at a time is requested by the scheduler. This allows up to 12 NAKs in a micro-frame and improves performance of these slow devices.
15 PARKMODE_DISABLE_FSLS	PARKMODE_DISABLE_FSLS This bit is used only in host mode, and is for debug purpose only. When this bit is set to '1' all FS/LS bus instances in park mode disabled.
14-11 reserved_14_11	Reserved
10 RESUME_OPMODE_HS_HOST	RESUME_OPMODE_HS_HOST This bit is used only in host mode, and is for USB 2.0 opmode behavior in HS Resume. - When this bit is set to '1', the UTMI/ULPI opmode will be changed to "normal" along with HS terminations after EOR. This option is to support certain legacy UTMI/ULPI PHYs. - When this bit is set to '0', the UTMI/ULPI opmode will be changed to "normal" 2us after HS terminations change after EOR. This is the default behavior.

Table continues on the next page...

Field	Function
9 DEV_HS_NYET_BULK_SPR	DEV_HS_NYET_BULK_SPR - 0: Default behavior, no change in device response - 1: Feature enabled, HS bulk OUT short packet gets NYET response This bit is applicable for device mode only (and ignored in host mode) to be used in 2.0 operation. If this bit is set, the device core sends NYET response instead of ACK response for a successfully received bulk OUT short packet. If NYET is sent after receiving short packet, then the host would PING before sending the next OUT; this improves the performance as well as clears up the buffer/cache on the host side. Internal to the device core, short packet (SPR=1) processing takes some time, and during this time, the USB is flow controlled. With NYET response instead of ACK on short packet, the host does not send another OUT-DATA without pinging in HS mode. This bit is quasi-static, that is, it must not be changed during device operation.
8 L1_SUSP_THRLD_EN_FOR_HOST	L1_SUSP_THRLD_EN_FOR_HOST This bit is used only in host mode. The host controller asserts the utmi_l1_suspend_n and utmi_sleep_n output signals (see "LPM Interface Signals" table in the Databook) as follows: The controller asserts the utmi_l1_suspend_n signal to put the PHY into deep low-power mode in L1 when both of the following are true: - The HIRD/BESL value used is greater than or equal to the value in L1_SUSP_THRLD_FOR_HOST field. - The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b1. The controller asserts utmi_sleep_n on L1 when one of the following is true: - The HIRD/BESL value used is less than the value in L1_SUSP_THRLD_FOR_HOST field. - The L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1'b0.
7-4 L1_SUSP_THRLD_FOR_HOST	L1_SUSP_THRLD_FOR_HOST This field is effective only when the L1_SUSP_THRLD_EN_FOR_HOST bit is set to 1. For more details, refer to the description of the L1_SUSP_THRLD_EN_FOR_HOST bit.
3 HC_ERRATA_ENABLE	Host ELD Enable (HELDEn) When this bit is set to 1, it enables the Exit Latency Delta (ELD) support defined in the xHCI 1.0 Errata. This bit is used only in the host mode. This bit has to be set to 1 in Host mode.
2 HC_PARCHK_DISABLE	Host Parameter Check Disable (HParChkDisable) When this bit is set to '0' (by default), the xHC checks that the input slot/EP context fields comply to the xHCI Specification. Upon detection of a parameter error during command execution, the xHC generates an event TRB with completion code indicating 'PARAMETER ERROR'. When the bit is set to '1', the xHC does not perform parameter checks and does not generate 'PARAMETER ERROR' completion code.
1 OVRLD_L1_SUSP_COM	OVRLD_L1_SUSP_COM If this bit is set, the utmi_l1_suspend_com_n is overloaded with the utmi_sleep_n signal. This bit is usually set if the PHY stops the port clock during L1 sleep condition. Note: The recommended connection for the SUSPENDM/SLEEPM signals to the PHY with respect to this bit is as follows. For non-zero ports: Connect: - utmi_sleep_n[n] to SLEEPM[n] - (utmi_suspend_n[n] & utmi_l1_suspend_n[n]) to SUSPENDM[n] - USB2 PHYCLK[n] to utmi_clk[n] GUCTL1.OVRLD_L1_SUSP_COM impacts only Port0. For Port0: For Synopsys PHY, GUSB2PHYCFGn.U2_FREECLK_EXISTS=1; With this connection, the PHY keeps PLL active so that FREECLK is always available irrespective of suspend/sleep. - Connect USB2 PHY COMMONONN to 0. - Connect utmi_sleep_n[0] to SLEEPM[0]. - Connect (utmi_suspend_n[0] & utmi_l1_suspend_n[0]) to SUSPENDM[0]. - Connect USB2 PHY FREECLK to utmi_clk[0]. - Leave utmi_suspend_com_n, utmi_l1_suspend_com_n unconnected. - GUCTL1.OVRLD_L1_SUSP_COM can be set to any value. For Third Party PHY, GUSB2PHYCFGn.U2_FREECLK_EXISTS=0; With this connection the PHY can shut off all the clocks when the required conditions are met (like, GUSB2PHYCFGn[8,6], GUCTL1[1], GFLADJ[23], GCTL[10], Suspend condition, HW LPM enable etc). - Connect -utmi_suspend_com_n to SUSPENDM[0] (or equivalent). - Connect -utmi_l1_suspend_com_n to SLEEPM[0] (or equivalent). - Connect PHYCLK0 (first port clock) to utmi_clk[0]. - Leave utmi_suspend_n[0], utmi_l1_suspend_ n[0], utmi_sleep_n[0] unconnected. - Set GUCTL1.OVRLD_L1_SUSP_COM to 1'b1.
0 LOA_FILTER_EN	LOA_FILTER_EN If this bit is set, the USB 2.0 port babble is checked at least three consecutive times before the port is disabled. This prevents false triggering of the babble condition when using low quality cables. Note: This bit is valid only in host mode.

11.1.3.1.53 Global User ID Register (GUID)

11.1.3.1.53.1 Offset

Register	Offset
GUID	C128h

11.1.3.1.53.2 Function

This is a read/write register containing the User ID.

11.1.3.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USERID															
Reset	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USERID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.53.4 Fields

Field	Function
31-0 USERID	USERID Application-programmable ID field.

11.1.3.1.54 Global User Control Register (GUCTL)

11.1.3.1.54.1 Offset

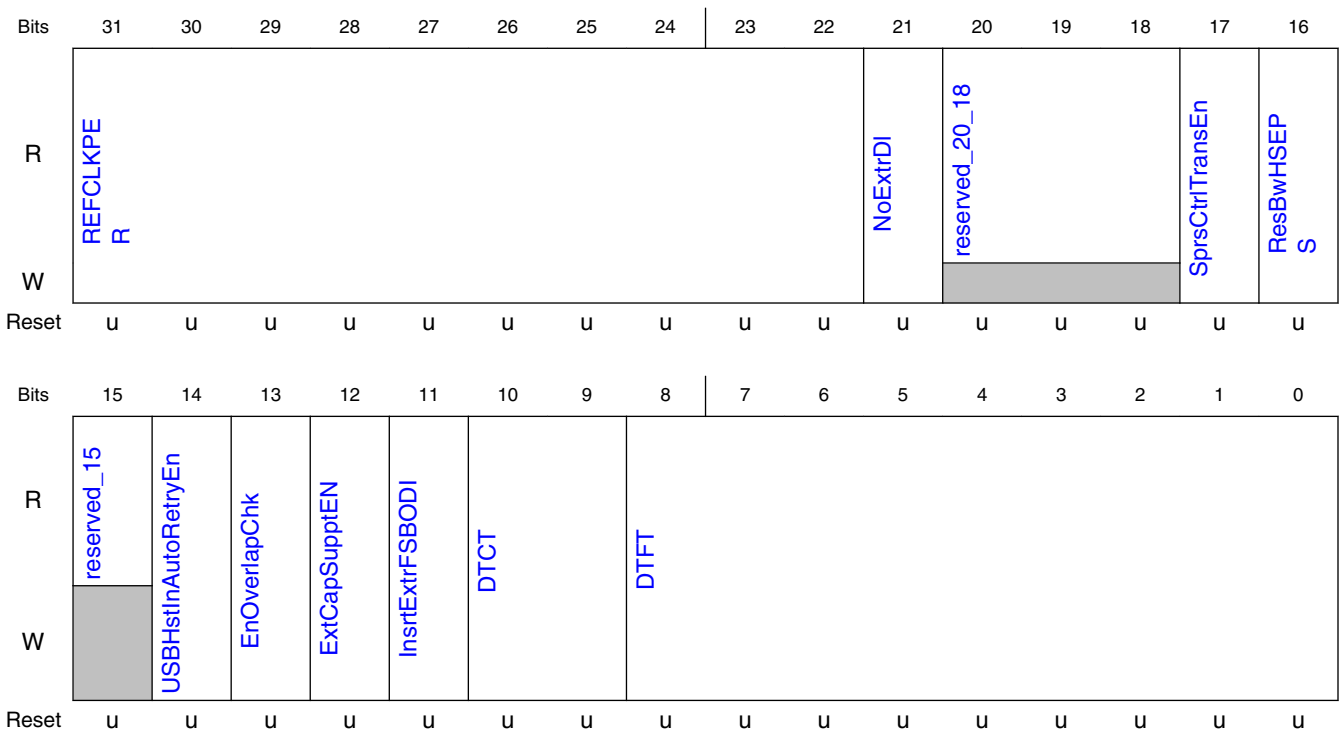
Register	Offset
GUCTL	C12Ch

11.1.3.1.54.2 Function

This register provides a few options for the software to control the core behavior in the Host mode.

Reset Mask: 0x1C8000

11.1.3.1.54.3 Diagram



11.1.3.1.54.4 Fields

Field	Function
31-22 REFCLKPER	REFCLKPER This field indicates in terms of nano seconds the period of ref_clk. The default value of this register is set to 'h8 (8ns/125 MHz). This field needs to be updated during power-on initialization, if GCTL.SOFITPSYNC or GFLADJ.GFLADJ_REFCLK_LPM_SEL is set to '1'. The programmable maximum value is 62ns, and the minimum value is 8ns. You must use a reference clock with a period that is an integer multiple, so that ITP can meet the jitter margin of 32ns. The allowable ref_clk frequencies whose period is not integer multiples are 16/17/19.2/24/39.7MHz. This field must not be set to '0' at any time. If you never plan to use this feature, then set this field to 'h8, the default value.
21 NoExtrDl	No Extra Delay Between SOF and the First Packet(NoExtrDl) Some HS devices misbehave when the host sends a packet immediately after a SOF. However, adding an extra delay between a SOF and the first packet can reduce the USB data rate and performance. This bit is used to control whether the host must wait for 2 microseconds before it sends the first packet after a SOF, or not. User can set this bit to one to improve the performance if those problematic devices are not a concern in the user's host environment. - 1'b0: Host waits for 2 microseconds after a SOF before it sends the first USB packet. - 1'b1: Host doesn't wait after a SOF before it sends the first USB packet.

Table continues on the next page...

Field	Function
20-18 reserved_20_18	Reserved
17 SprsCtrlTransEn	Sparse Control Transaction Enable Some devices are slow in responding to Control transfers. Scheduling multiple transactions in one microframe/frame can cause these devices to misbehave. If this bit is set to 1'b1, the host controller schedules transactions for a Control transfer in different microframes/frames.
16 ResBwHSEPS	Reserving 85% Bandwidth for HS Periodic EPs (ResBwHSEPS) By default, HC reserves 80% of the bandwidth for periodic EPs. If this bit is set, the bandwidth is relaxed to 85% to accommodate two high speed, high bandwidth ISOC EPs. USB 2.0 required 80% bandwidth allocated for ISOC traffic. If two High-bandwidth ISOC devices (HD Webcams) are connected, and if each requires 1024-bytes X 3 packets per Micro-Frame, then the bandwidth required is around 82%. If this bit is set, then it is possible to connect two Webcams of 1024bytes X 3 paylod per Micro-Frame each. Otherwise, you may have to reduce the resolution of the Webcams. This bit is valid in Host and DRD configuration and is used in host mode operation only. Ignore this bit in device mode.
15 reserved_15	Reserved
14 USBHstInAutoRetryEn	Host IN Auto Retry (USBHstInAutoRetryEn) When set, this field enables the Auto Retry feature. For IN transfers (non-isochronous) that encounter data packets with CRC errors or internal overrun scenarios, the auto retry feature causes the Host core to reply to the device with a non-terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP != 0). If the Auto Retry feature is disabled (default), the core will respond with a terminating retry ACK (that is, an ACK transaction packet with Retry = 1 and NumP = 0). - 1'b0: Auto Retry Disabled - 1'b1: Auto Retry Enabled Note: When enabling Auto Retry feature, if the system latency is large enough to cause the internal PSQ full (PSQ can be full as the result of messages not being processed because of pending fetches before flushing the TxQ due to NRDY/ERDY conditions), then the host controller can generate a transaction error.
13 EnOverlapChk	Enable Check for LFPS Overlap During Remote Ux Exit: If this bit is set to, - 1'b1: The SuperSpeed link when exiting U1/U2/U3 waits for either the remote link LFPS or TS1/TS2 training symbols before it confirms that the LFPS handshake is complete. This is done to handle the case where the LFPS glitch causes the link to start exiting from the low power state. Looking for the LFPS overlap makes sure that the link partner also sees the LFPS. - 1'b0: When the link exists U1/U2/U3 because of a remote exit, it does not look for an LFPS overlap.
12 ExtCapSupptEN	External Extended Capability Support Enable (ExtCapSuptEN) When set, this field enables extended capabilities to be implemented outside the core. When the ExtCapSupEN is set and the Debug Capability is enabled, the Next Capability pointer in "Debug Capability" returns 16. A read to the first DWORD of the last internal extended capability (the "xHCI Supported Protocol Capability for USB 3.0" when the Debug Capability is not enabled) returns a value of 4 in the Next Capability Pointer field. This indicates to software that there is another capability four DWORDs after this capability (for example, at address N+16 where N is the address of this DWORD). If enabled, an external address decoder that snoops the xHC slave interface must be implemented. If it sees an access to N+16 or greater, the slave access is re-routed to a piece of hardware which returns the external capability pointer register of the new capability and also handles reads/writes to this new capability and the side effects. If disabled, a read to the first DWORD of the last internal extended capability returns 0 in the 'Next Capability Pointer field. This indicates there are no more capabilities.
11 InsrtExtrFSBODI	Insert Extra Delay Between FS Bulk OUT Transactions (InsrtExtrFSBODl). Some FS devices are slow to receive Bulk OUT data and can get stuck when there are consecutive Bulk OUT transactions with short inter-transaction delays. This bit is used to control whether the host inserts extra delay between consecutive Bulk OUT transactions to a FS Endpoint. - 1'b0: Host doesn't insert extra delay between consecutive Bulk OUT transactions to a FS Endpoint. - 1'b1: Host inserts about 12us extra delay between consecutive Bulk OUT transactions to a FS Endpoint to work around the device issue. Note: Setting this bit to one will reduce the Bulk OUT transfer performance for most of the FS devices.
10-9 DTCT	Device Timeout Coarse Tuning (DTCT) This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. The core first checks the DTCT value. If it is 0, then the timeout value is defined by the DTFT. If it is non-zero, then it uses the following timeout values: - 2'b00: 0 usec -> use DTFT value instead - 2'b01: 500 usec - 2'b10: 1.5 msec - 2'b11:

Table continues on the next page...

Field	Function
	6.5 msec Note: When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.
8-0 DTFT	Device Timeout Fine Tuning (DTFT) This field is a Host mode parameter which determines how long the host waits for a response from device before considering a timeout. For the DTFT field to take effect, DTCT must be set to 2'b00. The DTFT value is the number of 125 MHz clocks * 256 to count before considering a device timeout. The minimum value of DTFT is 2. For example, if the mac3_clk is 125 MHz clk (8 ns period), this is calculated as follows: (DTFT value) * 256 * (8 ns) Quick Reference: - if DTFT = 0x2, 2*256*8 = 4usec timeout - if DTFT = 0x5, 5*256*8 = 10usec timeout - if DTFT = 0xA, 10*256*8 = 20usec timeout - if DTFT = 0x10, 16*256*8 = 32usec timeout - if DTFT = 0x19, 25*256*8 = 51usec timeout - if DTFT = 0x31, 49*256*8 = 100usec timeout - if DTFT = 0x62, 98*256*8 = 200usec timeout Note: - When SSIC is enabled, in HS_G1_G2_G3 mode when ssic_soc_pa_clk_freq = 2'b01, then DTFT value must be calculated for 156.25 MHz clock, for example, for equivalent delay of 32usec DTFT value must be 9'h14. - When the system latency is larger than the programmed DTCT/DTFT value, if the host controller is not able to accept certain transactions on the bus (because of system bus delays), the controller may not release header credits which in turn can cause the host to report a transaction error. Therefore, program this value to be larger than your system delay.

11.1.3.1.55 Gobal SoC Bus Error Address Register - Low (GBUSERRADDRLO)

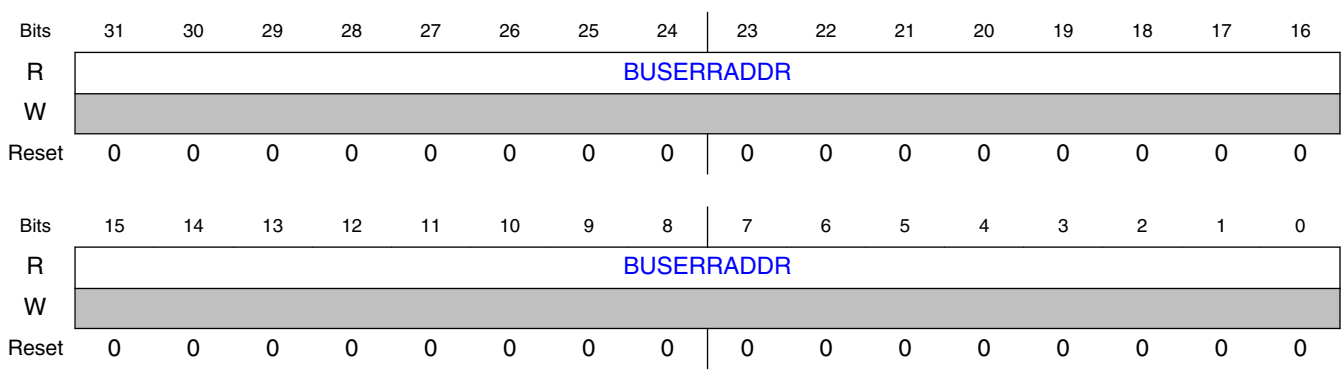
11.1.3.1.55.1 Offset

Register	Offset
GBUSERRADDRLO	C130h

11.1.3.1.55.2 Function

This is an alternate register for the GBUSERRADDR register.

11.1.3.1.55.3 Diagram



11.1.3.1.55.4 Fields

Field	Function
31-0 BUSERRADDR	Bus Address - Low (BusAddrLo) This register contains the lower 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

11.1.3.1.56 Gobal SoC Bus Error Address Register - High (GBUSERRADDRHI)

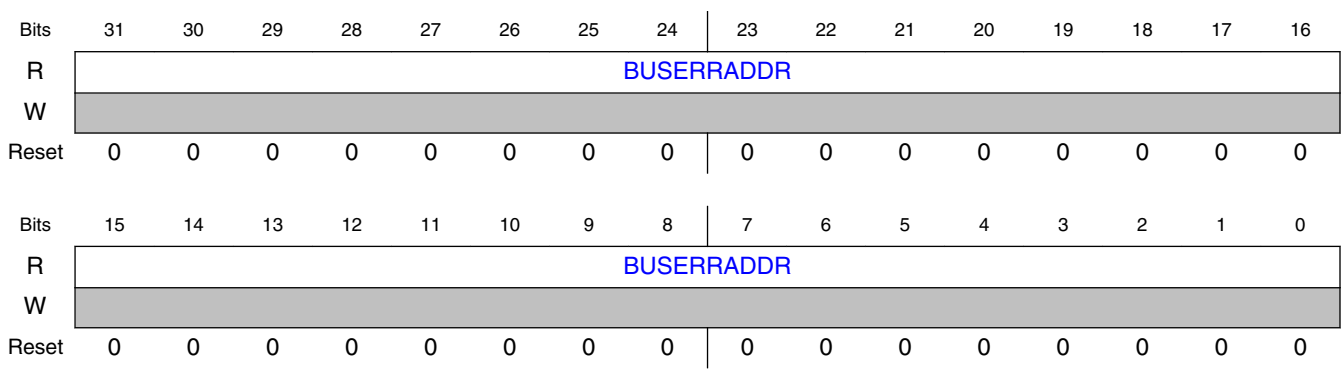
11.1.3.1.56.1 Offset

Register	Offset
GBUSERRADDRHI	C134h

11.1.3.1.56.2 Function

This is an alternate register for the GBUSERRADDR register.

11.1.3.1.56.3 Diagram



11.1.3.1.56.4 Fields

Field	Function
31-0 BUSERRADDR	Bus Address - High (BusAddrHi) This register contains the higher 32 bits of the first bus address that encountered a SoC bus error. It is valid when the GSTS.BusErrAddrVld field is 1. It can only be cleared by resetting the core. Note: Only supported in AHB and AXI configurations.

11.1.3.1.57 Global SS Port to Bus Instance Mapping Register - Low (GPRTBIMAPLO)

11.1.3.1.57.1 Offset

Register	Offset
GPRTBIMAPLO	C138h

11.1.3.1.57.2 Function

This is an alternate register for the GPRTBIMAP register.

11.1.3.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.57.4 Fields

Field	Function
31-28 BINUM8	BINUM8: SS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	BINUM7: SS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	BINUM6: SS USB Instance Number for Port 6. Application-programmable ID field.
19-16 BINUM5	BINUM5: SS USB Instance Number for Port 5. Application-programmable ID field.
15-12 BINUM4	BINUM4: SS USB Instance Number for Port 4. Application-programmable ID field.
11-8 BINUM3	BINUM3: SS USB Instance Number for Port 3. Application-programmable ID field.

Table continues on the next page...

Field	Function
7-4 BINUM2	BINUM2: SS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	BINUM1: SS USB Instance Number for Port 1. Application-programmable ID field.

11.1.3.1.58 Global SS Port to Bus Instance Mapping Register - High (GPRTBIMAPHI)

11.1.3.1.58.1 Offset

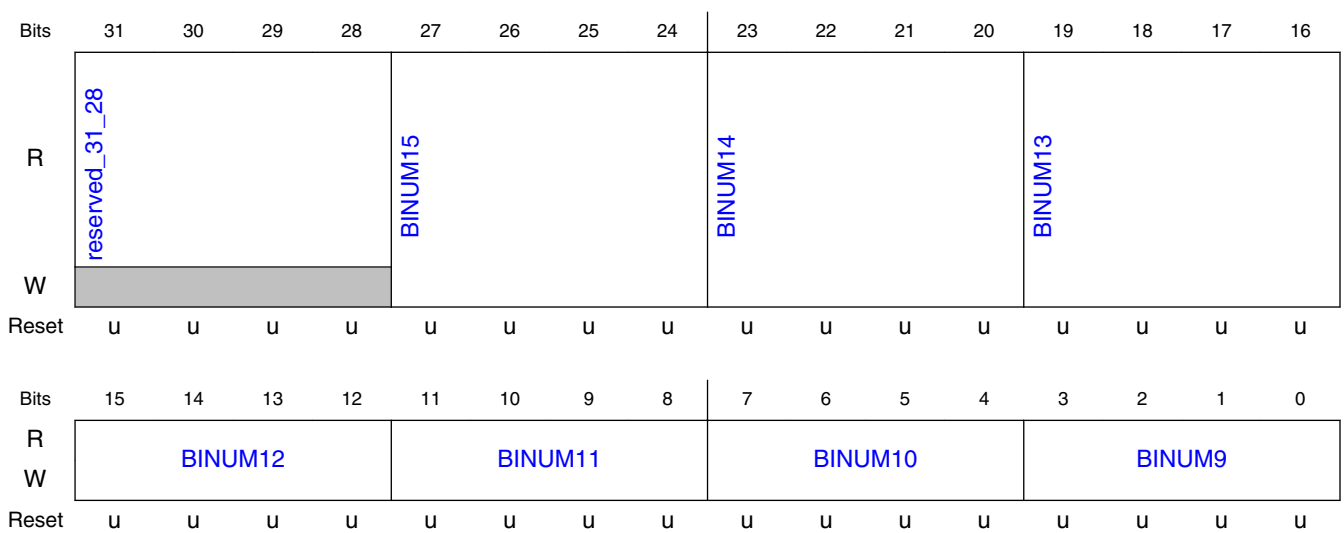
Register	Offset
GPRTBIMAPHI	C13Ch

11.1.3.1.58.2 Function

This is an alternate register for the GPRTBIMAP register.

Reset Mask: 0xF0000000

11.1.3.1.58.3 Diagram



11.1.3.1.58.4 Fields

Field	Function
31-28 reserved_31_28	Reserved
27-24 BINUM15	BINUM15: SS USB Instance Number for Port 15. Application-programmable ID field.
23-20 BINUM14	BINUM14: SS USB Instance Number for Port 14. Application-programmable ID field.
19-16 BINUM13	BINUM13: SS USB Instance Number for Port 13. Application-programmable ID field.
15-12 BINUM12	BINUM12: SS USB Instance Number for Port 12. Application-programmable ID field.
11-8 BINUM11	BINUM11: SS USB Instance Number for Port 11. Application-programmable ID field.
7-4 BINUM10	BINUM10: SS USB Instance Number for Port 10. Application-programmable ID field.
3-0 BINUM9	BINUM9: SS USB Instance Number for Port 9. Application-programmable ID field.

11.1.3.1.59 Global Hardware Parameters Register 0 (GHWPARAMS0)

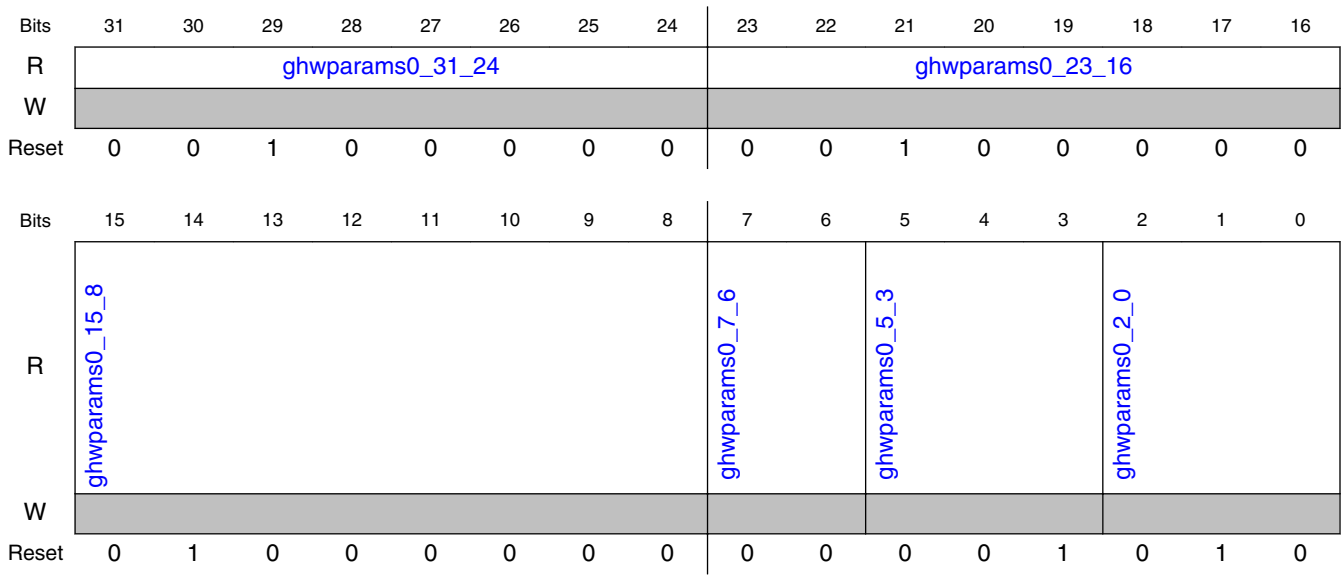
11.1.3.1.59.1 Offset

Register	Offset
GHWPARAMS0	C140h

11.1.3.1.59.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.59.3 Diagram



11.1.3.1.59.4 Fields

Field	Function
31-24 ghwparams0_31_24	`DWC_USB3_AWIDTH
23-16 ghwparams0_23_16	`DWC_USB3_SDWIDTH
15-8 ghwparams0_15_8	`DWC_USB3_MDWIDTH
7-6 ghwparams0_7_6	`DWC_USB3_SBUS_TYPE
5-3 ghwparams0_5_3	`DWC_USB3_MBUS_TYPE
2-0 ghwparams0_2_0	`DWC_USB3_MODE

11.1.3.1.60 Global Hardware Parameters Register 1 (GHWPARAMS1)

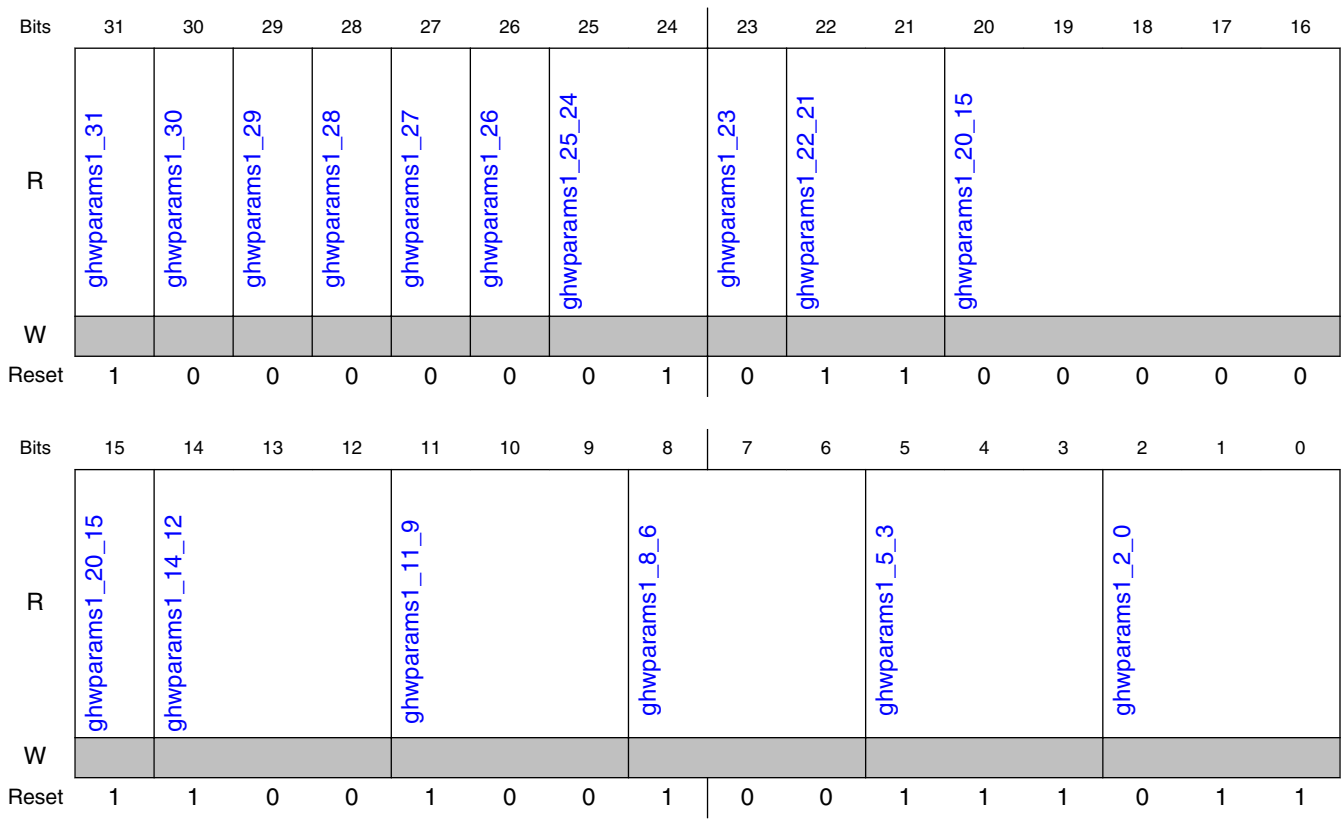
11.1.3.1.60.1 Offset

Register	Offset
GHWPARAMS1	C144h

11.1.3.1.60.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.60.3 Diagram



11.1.3.1.60.4 Fields

Field	Function
31 ghwparams1_31	`DWC_USB3_EN_DBC
30 ghwparams1_30	`DWC_USB3_RM_OPT_FEATURES

Table continues on the next page...

Field	Function
29 ghwparams1_29	Reserved1
28 ghwparams1_28	`DWC_USB3_RAM_BUS_CLKS_SYNC
27 ghwparams1_27	`DWC_USB3_MAC_RAM_CLKS_SYNC
26 ghwparams1_26	`DWC_USB3_MAC_PHY_CLKS_SYNC
25-24 ghwparams1_25_24	`DWC_USB3_EN_PWROPT
23 ghwparams1_23	`DWC_USB3_SPRAM_TYP
22-21 ghwparams1_22_21	`DWC_USB3_NUM_RAMS
20-15 ghwparams1_20_15	`DWC_USB3_DEVICE_NUM_INT For details on `DWC_USB3_DEVICE_NUM_INT, refer to <workspace>/src/DWC_usb3_params.v file.
14-12 ghwparams1_14_12	`DWC_USB3_ASPACEWIDTH
11-9 ghwparams1_11_9	`DWC_USB3_REQINFOWIDTH
8-6 ghwparams1_8_6	`DWC_USB3_DATAINFOWIDTH
5-3 ghwparams1_5_3	`DWC_USB3_BURSTWIDTH-1
2-0 ghwparams1_2_0	`DWC_USB3_IDWIDTH-1

11.1.3.1.61 Global Hardware Parameters Register 2 (GHWPARAMS2)

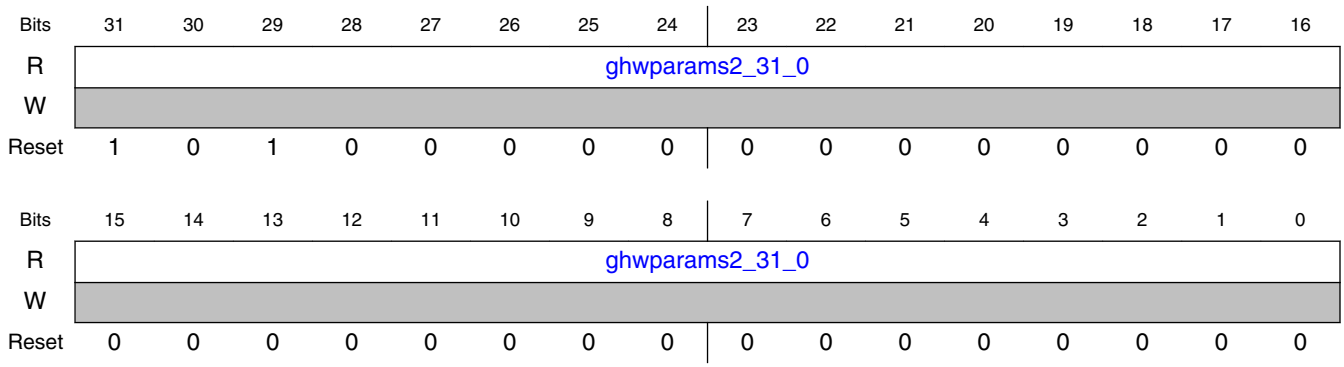
11.1.3.1.61.1 Offset

Register	Offset
GHWPARAMS2	C148h

11.1.3.1.61.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.61.3 Diagram



11.1.3.1.61.4 Fields

Field	Function
31-0 ghwparams2_31_0	`DWC_USB3_USERID

11.1.3.1.62 Global Hardware Parameters Register 3 (GHWPARAMS3)

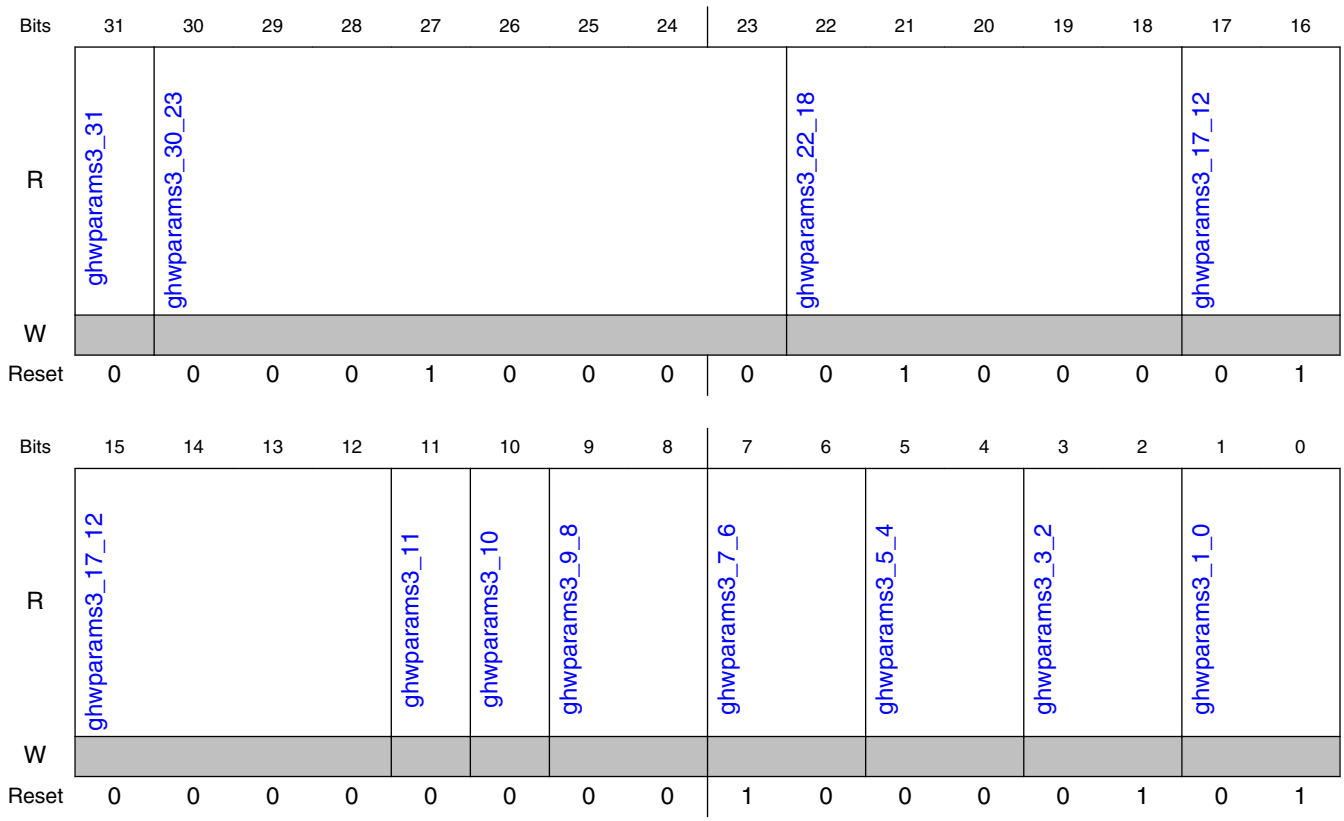
11.1.3.1.62.1 Offset

Register	Offset
GHWPARAMS3	C14Ch

11.1.3.1.62.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.62.3 Diagram



11.1.3.1.62.4 Fields

Field	Function
31 ghwparams3_31	Reserved
30-23 ghwparams3_30_23	`DWC_USB3_CACHE_TOTAL_XFER_RESOURCES
22-18 ghwparams3_22_18	`DWC_USB3_NUM_IN_EPS
17-12 ghwparams3_17_12	`DWC_USB3_NUM_EPS
11 ghwparams3_11	`DWC_USB3_ULPI_CARKIT
10 ghwparams3_10	`DWC_USB3_VENDOR_CTL_INTERFACE
9-8	Reserved

Table continues on the next page...

Field	Function
ghwparams3_9_8	
7-6 ghwparams3_7_6	`DWC_USB3_HSPHY_DWIDTH
5-4 ghwparams3_5_4	`DWC_USB3_FSPHY_INTERFACE
3-2 ghwparams3_3_2	`DWC_USB3_HSPHY_INTERFACE
1-0 ghwparams3_1_0	`DWC_USB3_SSPHY_INTERFACE

11.1.3.1.63 Global Hardware Parameters Register 4 (GHWPARAMS4)

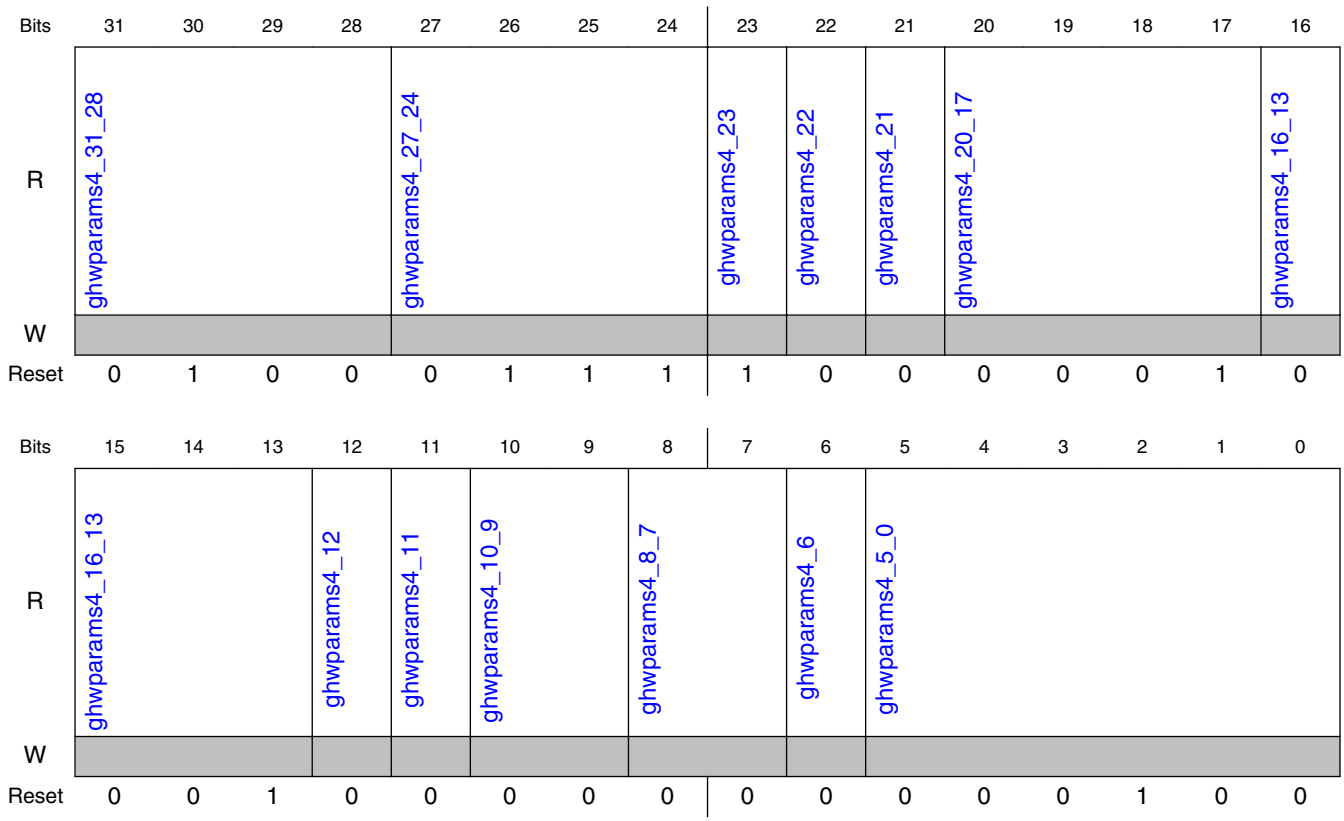
11.1.3.1.63.1 Offset

Register	Offset
GHWPARAMS4	C150h

11.1.3.1.63.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.63.3 Diagram



11.1.3.1.63.4 Fields

Field	Function
31-28 ghwparams4_31_28	`DWC_USB3_BMU_LSP_DEPTH
27-24 ghwparams4_27_24	`DWC_USB3_BMU_PTL_DEPTH-1
23 ghwparams4_23	`DWC_USB3_EN_ISOC_SUPT
22 ghwparams4_22	Reserved
21 ghwparams4_21	`DWC_USB3_EXT_BUFF_CONTROL
20-17 ghwparams4_20_17	`DWC_USB3_NUM_SS_USB_INSTANCES

Table continues on the next page...

Field	Function
16-13 ghwparams4_16_13	`DWC_USB3_HIBER_SCRATCHBUFS Number of external scratchpad buffers the core requires to save its internal state in the device mode. Each buffer is assumed to be 4KB. The scratchpad buffer array must have this many buffer pointers.
12 ghwparams4_12	`DWC_USB3_EN_SSIC - 1'b0: if DWC_USB3_EN_SSIC == 0 - 1'b1: if DWC_USB3_EN_SSIC != 0 Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is Reserved.
11 ghwparams4_11	`DWC_USB3_SSIC_NON_SNPS_MPHY This field indicates whether Synopsys M-PHY or a third-party M-PHY is used with SSIC ports. - 1'b0: Synopsys M-PHY - 1'b1: Third-party M-PHY Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this bit is Reserved.
10-9 ghwparams4_10_9	`DWC_USB3_SSIC_GEAR This field indicates DWC_USB3_SSIC_GEAR parameter value chosen by the user - 2'b00: Reserved - 2'b01: HS-G1 - 2'b10: HS-G2 - 2'b11: HS-G3 Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this field is Reserved.
8-7 ghwparams4_8_7	`DWC_USB3_NUM_SSIC_NUM_LANE This bit indicates `DWC_USB3_SSIC_NUM_LANE parameter value chosen by the user - 2'b00: 4 lane - 2'b01: 1 lane - 2'b10: 2 lane - 2'b11: Reserved Note: When the DWC_USB3_NUM_SSIC_PORTS parameter is set to zero, this field is Reserved.
6 ghwparams4_6	Reserved
5-0 ghwparams4_5_0	`DWC_USB3_CACHE_TRBS_PER_TRANSFER

11.1.3.1.64 Global Hardware Parameters Register 5 (GHWPARAMS5)

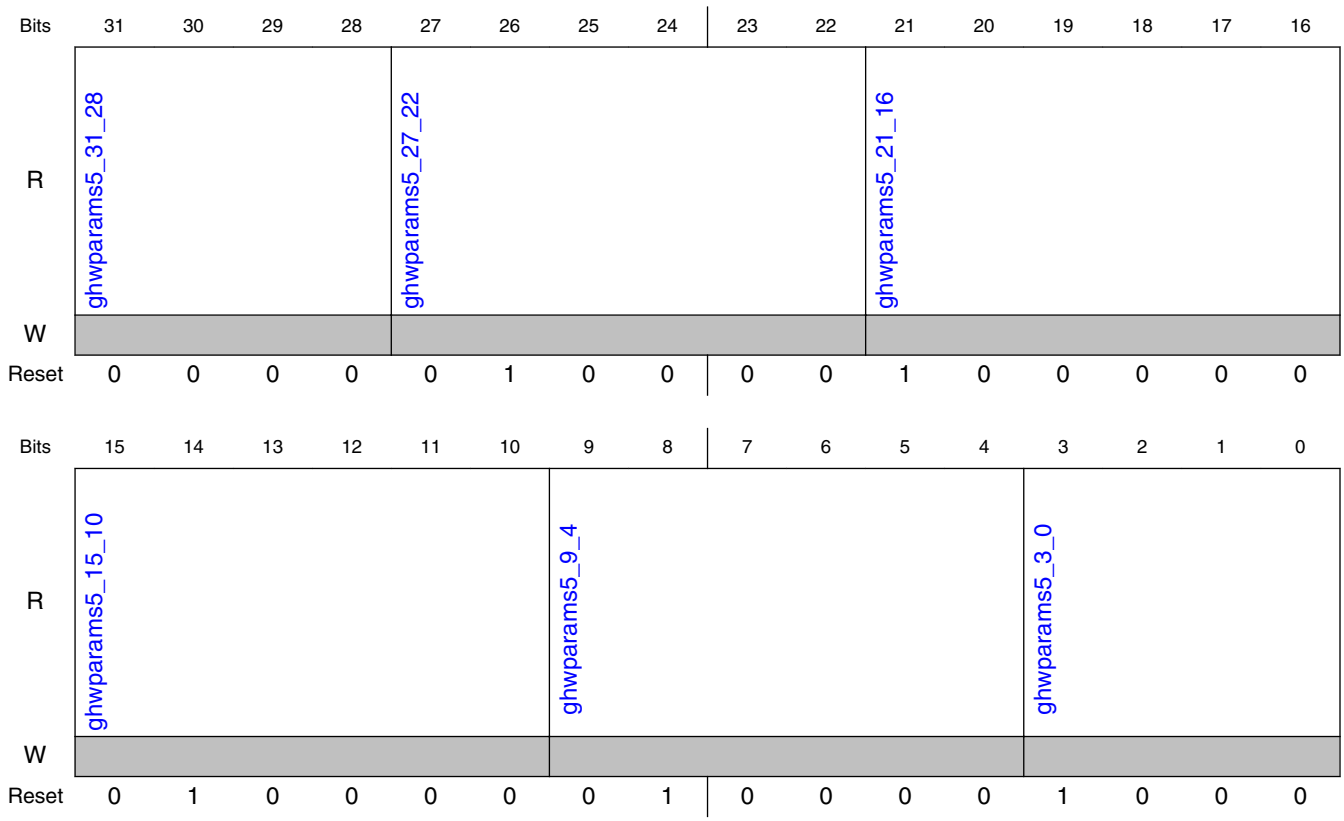
11.1.3.1.64.1 Offset

Register	Offset
GHWPARAMS5	C154h

11.1.3.1.64.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.64.3 Diagram



11.1.3.1.64.4 Fields

Field	Function
31-28 ghwparams5_31_28	Reserved
27-22 ghwparams5_27_22	`DWC_USB3_DFQ_FIFO_DEPTH
21-16 ghwparams5_21_16	`DWC_USB3_DWQ_FIFO_DEPTH
15-10 ghwparams5_15_10	`DWC_USB3_TXQ_FIFO_DEPTH
9-4 ghwparams5_9_4	`DWC_USB3_RXQ_FIFO_DEPTH
3-0	`DWC_USB3_BMU_BUSGM_DEPTH

Field	Function
ghwparams5_3_0	

11.1.3.1.65 Global Hardware Parameters Register 6 (GHWPARAMS6)

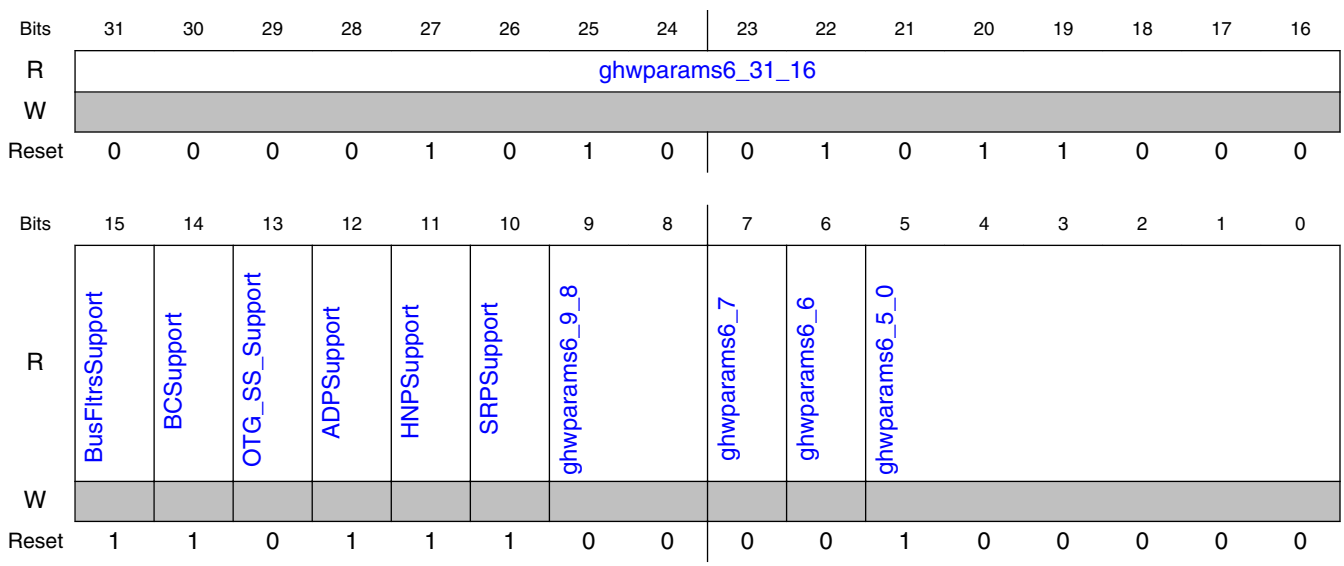
11.1.3.1.65.1 Offset

Register	Offset
GHWPARAMS6	C158h

11.1.3.1.65.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.65.3 Diagram



11.1.3.1.65.4 Fields

Field	Function
31-16 ghwparams6_31_16	`DWC_USB3_RAM0_DEPTH

Table continues on the next page...

Field	Function
15 BusFltrsSupport	`DWC_USB3_EN_BUS_FILTERS
14 BCSupport	`DWC_USB3_EN_BC
13 OTG_SS_Support	OTG 3.0 Support Enabled This bit indicates whether the parameter `DWC_USB3_EN_OTG is set to 2. In other words, it indicates that whether the DWC_usb3 core supports OTG 3.0. - 1'b0: No OTG 3.0 support - 1'b1: Supports OTG 3.0
12 ADPSupport	`DWC_USB3_EN_ADP
11 HNPSupport	RSP/HNP Support Enabled The application uses this bit to determine the DWC_usb3 core's RSP/HNP support. If DWC_USB3_EN_OTG=2, - 1'b0: RSP and HNP support is not enabled. The only exception for this rule is for SSPC-OTG devices where RSP support is not enabled, but HNP support is enabled. (Refer to the OCFG.SSPC-OTG bit.) - 1'b1: RSP and HNP support is enabled If DWC_USB3_EN_OTG=1, - 1'b0: HNP support is not enabled; - 1'b1: HNP support is enabled; This bit is enabled only if HNP mode was specified for HNP Mode of Operation in coreConsultant (parameter DWC_USB3_EN_OTG is not 0, and DWC_USB3_MODE is DRD). Otherwise, it reads 0.
10 SRPSupport	SRP Support Enabled The application uses this bit to determine the DWC_usb3 core's SRP support. - 1'b0: SRP support is not enabled; - 1'b1: SRP support is enabled; This bit is 1'b1 when the parameter DWC_USB3_EN_OTG is not 0.
9-8 ghwparams6_9_8	Reserved
7 ghwparams6_7	`DWC_USB3_EN_FPGA
6 ghwparams6_6	`DWC_USB3_EN_DBG_PORTS
5-0 ghwparams6_5_0	`DWC_USB3_PSQ_FIFO_DEPTH

11.1.3.1.66 Global Hardware Parameters Register 7 (GHWPARAMS7)

11.1.3.1.66.1 Offset

Register	Offset
GHWPARAMS7	C15Ch

11.1.3.1.66.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.66.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ghwparams7_31_16															
W																
Reset	0	0	0	0	0	1	1	0	0	0	0	0	1	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ghwparams7_15_0															
W																
Reset	0	0	0	1	0	0	0	1	1	1	1	0	1	1	0	0

11.1.3.1.66.4 Fields

Field	Function
31-16 ghwparams7_31_16	`DWC_USB3_RAM2_DEPTH
15-0 ghwparams7_15_0	`DWC_USB3_RAM1_DEPTH

11.1.3.1.67 Global High-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_HSLO)

11.1.3.1.67.1 Offset

Register	Offset
GPRTBIMAP_HSLO	C180h

11.1.3.1.67.2 Function

This is an alternate register for the GPRTBIMAP_HS register.

11.1.3.1.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.67.4 Fields

Field	Function
31-28 BINUM8	BINUM8: HS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	BINUM7: HS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	BINUM6 USB Instance Number for Port 6. Application-programmable ID field.
19-16 BINUM5	BINUM5: HS USB Instance Number for Port 5. Application-programmable ID field.
15-12 BINUM4	BINUM4: HS USB Instance Number for Port 4. Application-programmable ID field.
11-8 BINUM3	BINUM3: HS USB Instance Number for Port 3. Application-programmable ID field.
7-4 BINUM2	BINUM2: HS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	BINUM1: HS USB Instance Number for Port 1. Application-programmable ID field.

11.1.3.1.68 Global High-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_HSHI)

11.1.3.1.68.1 Offset

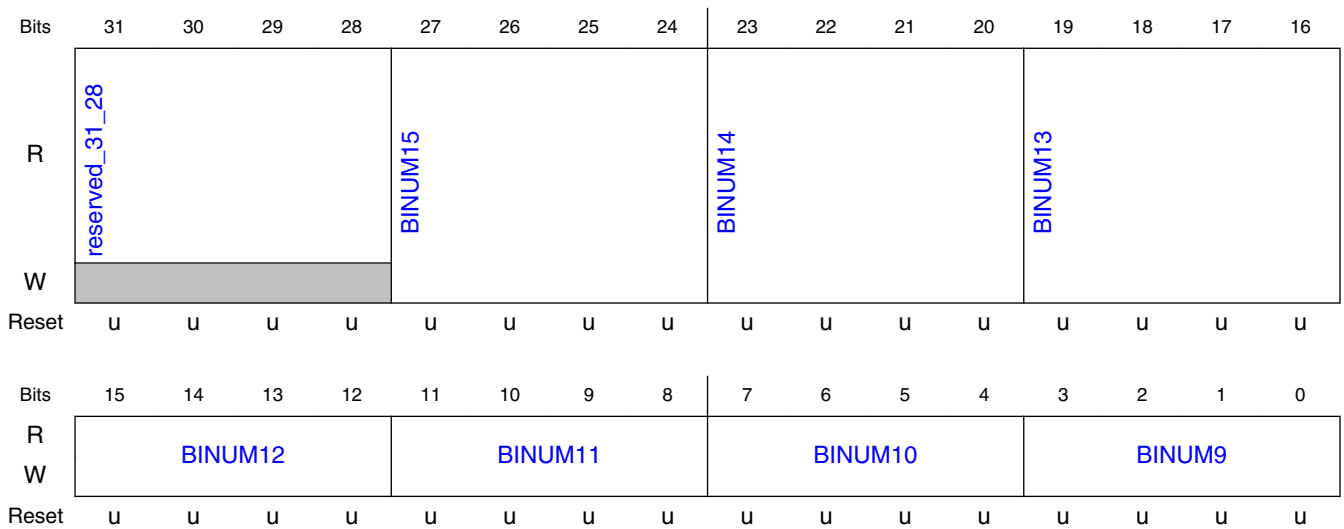
Register	Offset
GPRTBIMAP_HSHI	C184h

11.1.3.1.68.2 Function

This is an alternate register for the GPRTBIMAP_HS register.

Reset Mask: 0xF0000000

11.1.3.1.68.3 Diagram



11.1.3.1.68.4 Fields

Field	Function
31-28 reserved_31_28	Reserved
27-24 BINUM15	BINUM15: HS USB Instance Number for Port 15. Application-programmable ID field.
23-20 BINUM14	BINUM14: HS USB Instance Number for Port 14. Application-programmable ID field.
19-16 BINUM13	BINUM13: HS USB Instance Number for Port 13. Application-programmable ID field.
15-12 BINUM12	BINUM12: HS USB Instance Number for Port 12. SApplication-programmable ID field.
11-8 BINUM11	BINUM11: HS USB Instance Number for 11. Application-programmable ID field.
7-4 BINUM10	BINUM10: HS USB Instance Number for Port 10. Application-programmable ID field.
3-0 BINUM9	BINUM9: HS USB Instance Number for Port 9. Application-programmable ID field.

11.1.3.1.69 Global Full-Speed Port to Bus Instance Mapping Register - Low (GPRTBIMAP_FSLO)

11.1.3.1.69.1 Offset

Register	Offset
GPRTBIMAP_FSLO	C188h

11.1.3.1.69.2 Function

This is an alternate register for the GPRTBIMAP_FS register.

11.1.3.1.69.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BINUM8				BINUM7				BINUM6				BINUM5			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BINUM4				BINUM3				BINUM2				BINUM1			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.69.4 Fields

Field	Function
31-28 BINUM8	BINUM8: FS USB Instance Number for Port 8. Application-programmable ID field.
27-24 BINUM7	BINUM7: FS USB Instance Number for Port 7. Application-programmable ID field.
23-20 BINUM6	BINUM6: FS USB Instance Number for Port 6. Application-programmable ID field.
19-16 BINUM5	BINUM5: FS USB Instance Number for Port 5. Application-programmable ID field.
15-12 BINUM4	BINUM4: FS USB Instance Number for Port 4. Application-programmable ID field.
11-8	BINUM3: FS USB Instance Number for Port 3. Application-programmable ID field.

Table continues on the next page...

Field	Function
BINUM3	
7-4 BINUM2	BINUM2: FS USB Instance Number for Port 2. Application-programmable ID field.
3-0 BINUM1	BINUM1: FS USB Instance Number for Port 1. Application-programmable ID field.

11.1.3.1.70 Global Full-Speed Port to Bus Instance Mapping Register - High (GPRTBIMAP_FSHI)

11.1.3.1.70.1 Offset

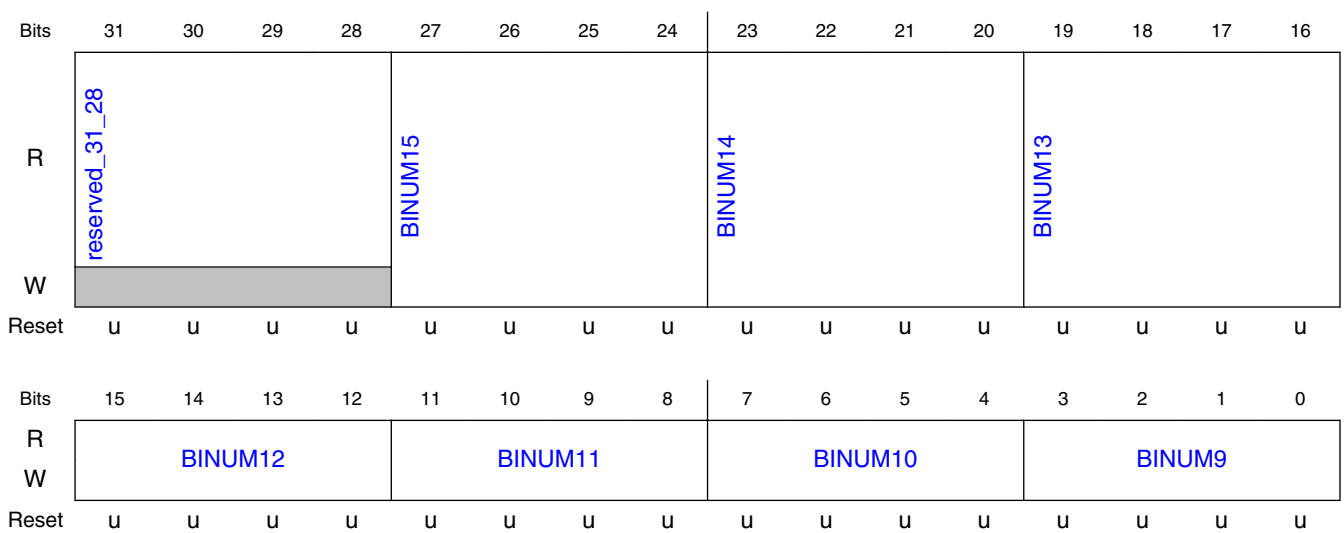
Register	Offset
GPRTBIMAP_FSHI	C18Ch

11.1.3.1.70.2 Function

This is an alternate register for the GPRTBIMAP_FS register.

Reset Mask: 0xF0000000

11.1.3.1.70.3 Diagram



11.1.3.1.70.4 Fields

Field	Function
31-28 reserved_31_28	Reserved
27-24 BINUM15	BINUM15: FS USB Instance Number for Port 15. Application-programmable ID field
23-20 BINUM14	BINUM14: FS USB Instance Number for Port 14. Application-programmable ID field
19-16 BINUM13	BINUM13: FS USB Instance Number for Port 13. Application-programmable ID field
15-12 BINUM12	BINUM12: FS USB Instance Number for Port 12. Application-programmable ID field
11-8 BINUM11	BINUM11: FS USB Instance Number for Port 11. Application-programmable ID field
7-4 BINUM10	BINUM10: FS USB Instance Number for Port 10. Application-programmable ID field
3-0 BINUM9	BINUM9: FS USB Instance Number for Port 9. Application-programmable ID field.

11.1.3.1.71 Global User Control Register 2 (GUCTL2)

11.1.3.1.71.1 Offset

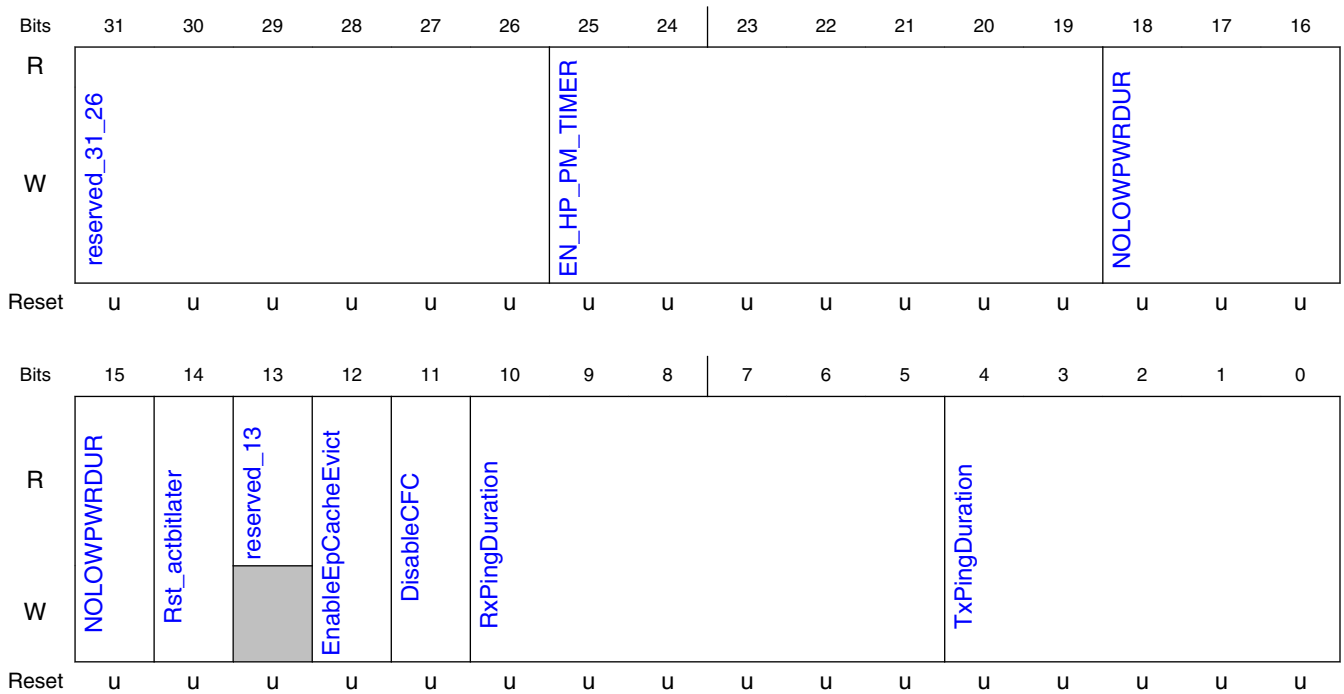
Register	Offset
GUCTL2	C19Ch

11.1.3.1.71.2 Function

This register provides a few options for the software to control the core behavior in the Host and device mode.

Reset Mask: 0x3FC008000

11.1.3.1.71.3 Diagram



11.1.3.1.71.4 Fields

Field	Function
31-26 reserved_31_26	Reserved
25-19 EN_HP_PM_TIMER	This register field is used to set new HP and PM timers. - To enable PM timer, set GUCTL2[19] bit as 1. - To enable HP timer, set GUCTL2[20] bit as 1. Default value is 4us, when new HP timer is enabled default value is 11us. Use GUCTL2[25:21] to specify HP timer value in microseconds.
18-15 NOLOWPWRDUR	No Low Power Duration (NOLOWPWRDUR) This bit is applicable for device mode only and is ignored in host mode. After starting a transfer on an ESS ISOC endpoint, the application must program these bits. Each count represents the duration in terms of milli seconds. For example, a value of 3 represents 3 ms.
14 Rst_actbitlater	Enable clearing of the command active bit for the ENDXFER command after the command execution is completed. This bit is valid in device mode only.
13 reserved_13	Reserved for future use
12 EnableEpCacheEvict	Enable Evicting Endpoint cache after Flow Control for bulk endpoints. In 3.00a release, a performance enhancement was done to keep the non-stream capable bulk IN endpoint in cache after flow control. Setting this bit will disable this enhancement. This should be set only for debug purpose.
11 DisableCFC	Disable xHCI Errata Feature Contiguous Frame ID Capability This field controls the xHCI Errata feature Contiguous FrameID capability. When set, the xHCI HCCPARAMS1 bit 11 will be set to 0 indicating that CFC is not supported. Disable this feature only if your application cannot tolerate Misssed Service Error events for Isoc transfers, and your system latencies are large to cause Missed Service errors even if the software is following the Isochronous Thresholding rules.

Table continues on the next page...

Field	Function
10-5 RxPingDuration	Recieve Ping Maximum Duration This field is relevant to Host mode and controls the maximum duration of received LFPS to be treated as a Ping LFPS. The Max duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 32 indicates 256 ns.
4-0 TxPingDuration	Transmit Ping Maximum Duration This field is relevant to Device mode and controls the maximum duration for which the controller should instruct the PHY to transmit a Ping LFPS. The duration of the Ping LFPS is controlled by programming this value and is in terms of 8 ns granularity. Eg: A value of 13 indicates 104 ns.

11.1.3.1.72 Global USB2 PHY Configuration Register (GUSB2PHYCFG)

11.1.3.1.72.1 Offset

Register	Offset
GUSB2PHYCFG	C200h

11.1.3.1.72.2 Function

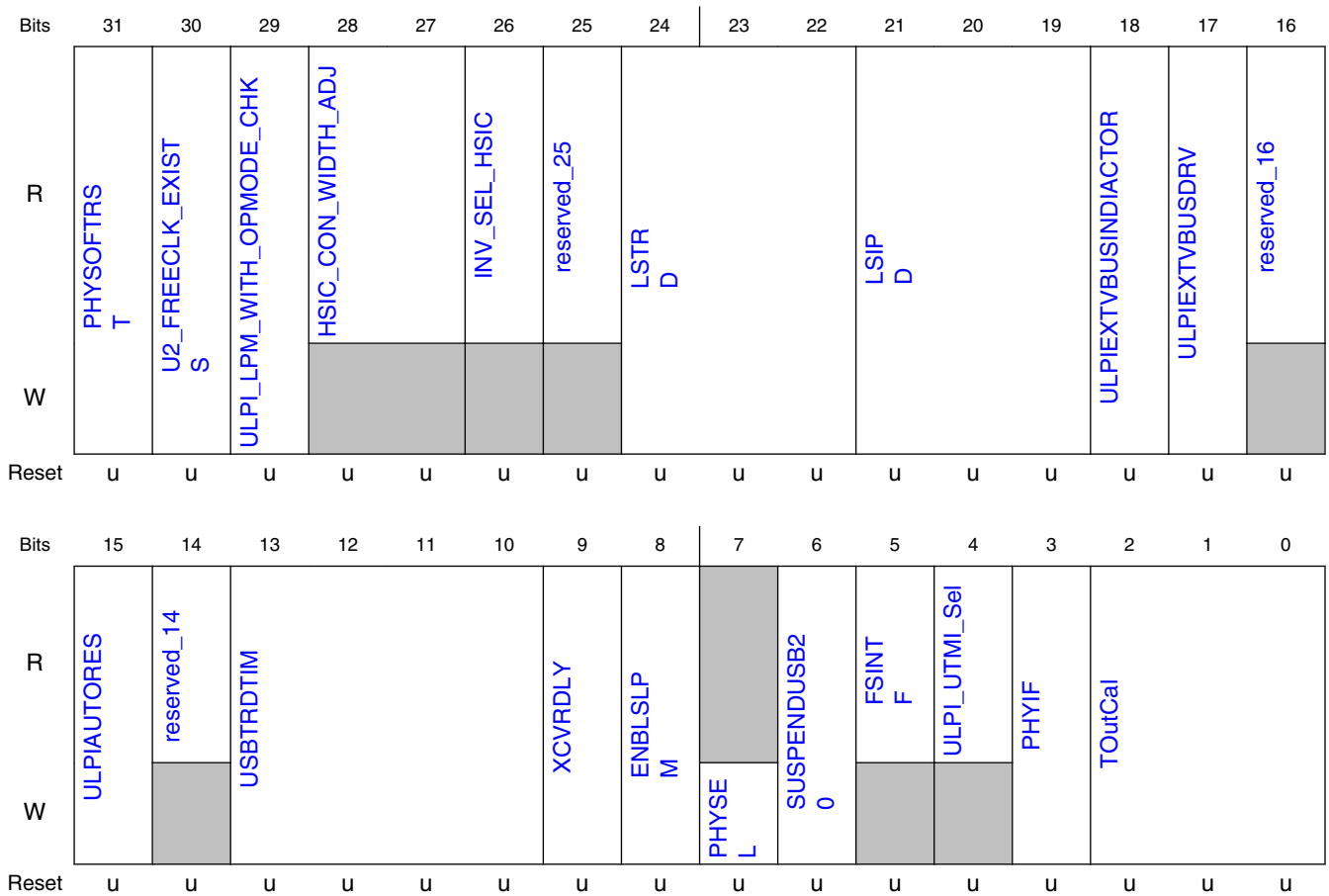
The application must program this register before starting any transactions on either the SoC bus or the USB.

Reset Mask:0x2014000

this register before starting any transactions on either the SoC bus or the USB. In

registers are implemented.

11.1.3.1.72.3 Diagram



11.1.3.1.72.4 Fields

Field	Function
31 PHYSOFTRST	UTMI PHY Soft Reset (PHYSoftRst) Causes the usb2phy_reset signal to be asserted to reset a UTMI PHY.
30 U2_FREECLK_EXISTS	U2_FREECLK_EXISTS Specifies whether your USB 2. free-running PHY clock, which is active when the clock control input is active. If your USB 2.0 PHY provides a free-running PHY clock, it must be connected to the utmi_clk[0] input. The remaining utmi_clk[n] must be connected to the respective port clocks. The core uses the Port-0 clock for generating the internal mac2 clock. - 1'b0: USB 2.0 free clock does not exist - 1'b1: USB 2.0 free clock exists Note: When the core is configured as device-only (DWC_USB3_MODE = 0), do not set this bit to 1.
29 ULPI_LPM_WITH_OPMODE_CHK	ULPI_LPM_WITH_OPMODE_CHK Support the LPM over ULPI without NOPID token to the ULPI PHY. token to the ULPI PHY. If this bit is set, the ULPI PHY is expected to qualify the EXT PID with OPMODE=2'b00 for LPM and not treat it as a NOPID. Check with your PHY vendor about your PHY behavior. This bit is valid only when the DWC_USB3_HSPHY_INTERFACE parameter is 2 or 3. - 1'b0: A

Table continues on the next page...

Field	Function
	NOPID is sent before sending an EXTPID for LPM; - 1'b1: An EXTPID is sent without previously sending a NOPID; Note: This bit is valid only in host mode. This bit should be '0' for Synopsys PHY.
28-27 HSIC_CON_WIDTH_ADJ	HSIC_CON_WIDTH_ADJ This bit is used in the HSIC device mode of operation.
26 INV_SEL_HSIC	INV_SEL_HSIC The application driver uses this bit to control the HSIC enable/disable function.
25 reserved_25	Reserved
24-22 LSTRD	LS Turnaround Time (LSTRDTIM) This field indicates the value of the Rx-to-Tx packet gap for LS devices.
21-19 LSIPD	LS Inter-Packet Time (LSIPD) This field indicates the value of Tx-to-Tx packet gap for LS devices.
18 ULPIEXTVBUSINDIACTOR	ULPI External VBUS Indicator (ULPIExtVbusIndicator) Indicates the ULPI PHY VBUS over-current indicator.
17 ULPIEXTVBUSDRV	ULPI External VBUS Drive (ULPIExtVbusDrv) Selects supply source to drive 5V on VBUS, in the ULPI PHY.
16 reserved_16	Reserved
15 ULPIAUTORES	ULPI Auto Resume (ULPIAutoRes) Sets the AutoResume bit in Interface Control register on the ULPI PHY.
14 reserved_14	Reserved
13-10 USBTRDTIM	USB 2.
9 XCVRDLY	Transceiver Delay: Enables a delay between the assertion of the UTMI/ULPI Transceiver Select signal (for HS) and the assertion of the TxValid signal during a HS Chirp.
8 ENBLSLPM	Enable utmi_sleep_n and utmi_l1_suspend_n (EnblSlpM) The application uses this bit to control utmi_sleep_n and utmi_l1_suspend_n assertion to the PHY in the L1 state.
7 PHYSEL	USB 2.
6 SUSPENDUSB20	Suspend USB2.
5 FSINTF	Full-Speed Serial Interface Select (FSIntf) The application uses this bit to select a unidirectional or bidirectional USB 1.
4	ULPI or UTMI+ Select (ULPI_UTMI_Sel) The application uses this bit

Table continues on the next page...

Field	Function
ULPI_UTMI_Sel	to select a UTMI+ or ULPI Interface.
3 PHYIF	PHY Interface (PHYIf) If UTMI+ is selected, the application uses this bit to configure the core to support a UTMI+ PHY with an 8- or 16-bit interface.
2-0 TOutCal	HS/FS Timeout Calibration (TOutCal) The number of PHY clocks, as indicated by the application in this field, is multiplied by a bit-time factor; this factor is added to the high-speed/full-speed interpacket timeout duration in the core to account for additional delays introduced by the PHY.

11.1.3.1.73 Global USB 2.0 UTMI PHY vendor control register (GUSB2PHYACC_ULPI)

11.1.3.1.73.1 Offset

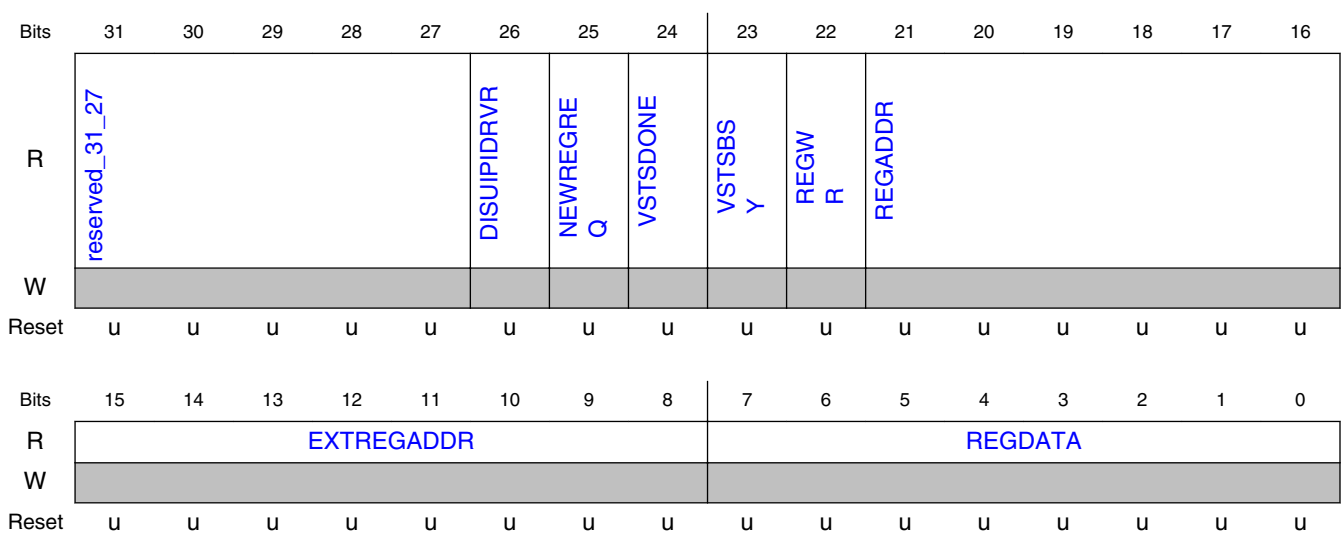
Register	Offset
GUSB2PHYACC_ULPI	C280h

11.1.3.1.73.2 Function

Global USB 2.

Reset Mask:0xF8000000

11.1.3.1.73.3 Diagram



11.1.3.1.73.4 Fields

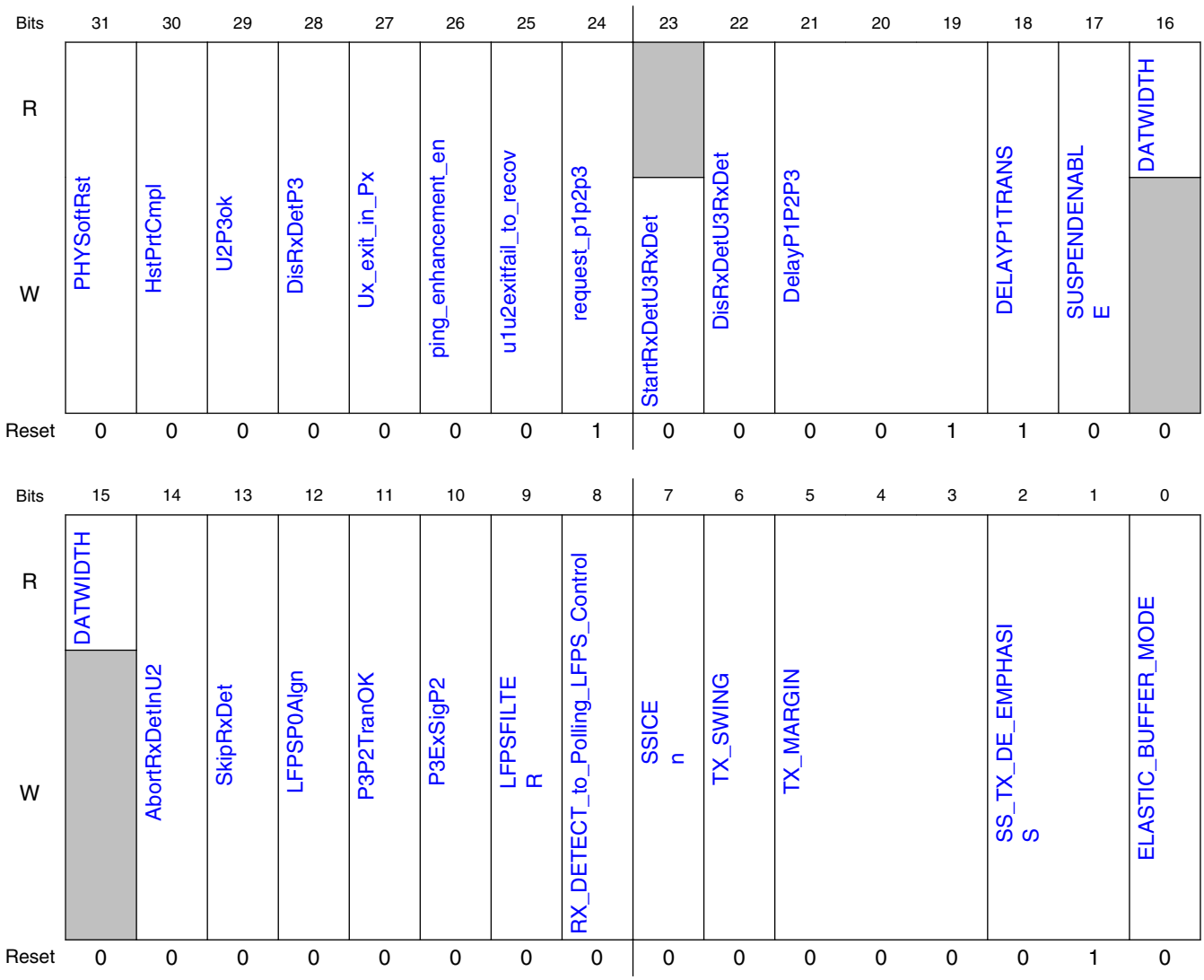
Field	Function
31-27 reserved_31_27	Reserved
26 DISUIPIDRVR	DISUIPIDRVR
25 NEWREGREQ	New Register Request The application sets this bit for a new vendor control access.
24 VSTSDONE	VSTSDONE
23 VSTSBSY	VSTSBSY
22 REGWR	Register Write The application sets this bit for register writes and clears it for register reads. clears it for register reads. Note: This bit is applicable for ULPI register read/write access only.
21-16 REGADDR	Register Address The 6-bit PHY register address for immediate PHY Register Set access. Register Set access. Set to 6'h2F for Extended PHY Register Set access. Note: These bits are applicable for ULPI only.
15-8 EXTREGADDR	EXTREGADDR
7-0 REGDATA	REGDATA

11.1.3.1.74 Global USB 3.0 PIPE control register (GUSB3PIPECTL)

11.1.3.1.74.1 Offset

Register	Offset
GUSB3PIPECTL	C2C0h

11.1.3.1.74.2 Diagram



11.1.3.1.74.3 Fields

Field	Function
31 PHYSoftRst	USB3 PHY Soft Reset After setting this bit to '1', the software needs to clear this bit.
30 HstPrtCmpl	HstPrtCmpl This feature tests the PIPE PHY compliance patterns without having to have a test fixture on the USB 3.
29 U2P3ok	P3 OK for U2 (u2P3ok) - 0: During link state U2/SS.
28 DisRxDetP3	Disabled receiver detection in P3 (DisRxDetP3) - 0: If PHY is in P3 and Core needs to perform receiver detection, The core performs receiver detection in

Table continues on the next page...

Field	Function
	P3.
27 Ux_exit_in_Px	Ux Exit in Px (Ux_exit_in_Px) - 0: The core does U1/U2/U3 exit in PHY power state P0 (default behavior).
26 ping_enhancement_en	Ping Enhancement Enable (ping_enhancement_en) When set, the Downstream port U1 ping receive timeout becomes 500 ms instead of 300 ms.
25 u1u2exitfail_to_recov	U1U2exitfail to Recovery (u1u2exitfail_to_recov) When set, and U1/U2 LFPS handshake fails, the LTSSM transitions from U1/U2 to Recovery instead of SS Inactive.
24 request_p1p2p3	Always Request P1/P2/P3 for U1/U2/U3 (request_p1p2p3) When set, the core always requests PHY power change from P0 to P1/P2/P3 during U0 to U1/U2/U3 transition.
23 StartRxDetU3RxDet	Start Receiver Detection in U3/Rx.
22 DisRxDetU3RxDet	Disable Receiver Detection in U3/Rx.
21-19 DelayP1P2P3	Delay P1P2P3 Delay P0 to P1/P2/P3 request when entering U1/U2/U3 until (DWC_USB3_GUSB3PIPECTL_INIT[21:19]*8) 8B10B error occurs, or Pipe3_RxValid drops to 0.
18 DELAYP1TRANS	Delay PHY power change from P0 to P1/P2/P3 when link state changing from U0 to U1/U2/U3 respectively.
17 SUSPENDENABLE	Suspend USB3.
16-15 DATWIDTH	PIPE Data Width (DatWidth) - 2'b00: 32 bits - 2'b01: 16 bits - 2'b10: 8 bits One clock after reset, these bits receive the value seen on the pipe3_DataBusWidth.
14 AbortRxDetInU2	Abort Rx Detect in U2 (AbortRxDetInU2) When set, and the link state is U2, then the core will abort receiver detection if it receives U2 exit LFPS from the remote link partner.
13 SkipRxDet	Skip Rx Detect: When set, the core skips Rx Detection if pipe3_RxElecIdle is low.
12 LFPSP0Algn	LFPS P0 Align: When set, - The core deasserts LFPS transmission on the clock edge that it requests Phy power state 0 when exiting U1, U2, or U3 low power states.
11 P3P2TranOK	P3 P2 Transitions OK (P3P2TranOK) When set, the core transitions directly from Phy power state P2 to P3 or from state P3 to P2.
10	P3 Exit Signal in P2 (P3ExSigP2) When this bit is set, the core

Table continues on the next page...

Field	Function
P3ExSigP2	always changes the PHY power state to P2, before attempting a U3 exit handshake.
9 LFPSFILTER	LFPS Filter (LFPSFilt) When set, filter LFPS reception with pipe3_RxValid in PHY power state P0, that is, ignore LFPS reception from the PHY unless both pipe3_Rxelecidle and pipe3_RxValid are deasserted.
8 RX_DETECT_to_Polling_LFPS_Control	RX_DETECT to Polling.
7 SSICEn	USB3 SSIC Enable (SSICEn) This bit is valid only when coreConsultant parameter DWC_USB3_EN_SSIC=1 else this bit needs to be set to 1'b0.
6 TX_SWING	Tx Swing (TxSwing) Refer to the PIPE3 specification.
5-3 TX_MARGIN	Tx Margin[2:0] (TxMargin) Refer to Table 5-3 of the PIPE3 Specification.
2-1 SS_TX_DE_EMPHASIS	Tx Deemphasis (TxDeemphasis) The value driven to the PHY is controlled by the LTSSM during USB3 Compliance mode.
0 ELASTIC_BUFFER_MODE	Elastic Buffer Mode (ElasticBufferMode) (Refer to Table 5-3 of the PIPE3 specification.

11.1.3.1.75 Global transmit FIFO size register (GTXFIFOSIZ0 - GTXFIFOSIZ7)

11.1.3.1.75.1 Offset

For a = 0 to 7:

Register	Offset
GTXFIFOSIZa	C300h + (a × 4h)

11.1.3.1.75.2 Function

This register specifies the RAM start address and depth (both in MDWIDTH-bit words) for each implemented TxFIFO.

11.1.3.1.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TXFDEP_N															
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

11.1.3.1.75.4 Fields

Field	Function
31-16 TXFSTADDR_N	Transmit FIFOn RAM Start Address This field contains the memory start address for TxFIFOn in MDWIDTH-bit words.
15-0 TXFDEP_N	TXFIFO depth This field contains the depth of TxFIFOn in MDWIDTH-bit words.

11.1.3.1.76 Global receive FIFO size register (GRXFIFOSIZ0 - GRXFIFOSIZ2)

11.1.3.1.76.1 Offset

Register	Offset
GRXFIFOSIZ0	C380h
GRXFIFOSIZ1	C384h
GRXFIFOSIZ2	C388h

11.1.3.1.76.2 Function

This register specifies the RAM start address and depth (both in MDWIDTH-bit words) for each implemented RxFIFO.

11.1.3.1.76.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RXFSTADDR_N															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RXFDEP_N															
Reset	0	0	0	0	0	0	1	0	1	0	0	0	0	1	0	1

11.1.3.1.76.4 Fields

Field	Function
31-16 RXFSTADDR_N	RxFIFOn RAM Start Address (RxFStAddr_n) This field contains the memory start address for RxFIFOn in MDWIDTH-bit words.
15-0 RXFDEP_N	RxFIFO Depth (RxFDep_n) This field contains the depth of RxFIFOn in MDWIDTH-bit words.

11.1.3.1.77 Global Event Buffer Address (Low) Register (GEVNTADRLO)

11.1.3.1.77.1 Offset

Register	Offset
GEVNTADRLO	C400h

11.1.3.1.77.2 Function

This is an alternate register for the GEVNTADRn register.

11.1.3.1.77.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRLO															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.77.4 Fields

Field	Function
31-0 EVNTADRLO	Event Buffer Address (EvntAdrLo) Holds the lower 32 bits of start address of the external memory for the Event Buffer.

11.1.3.1.78 Global Event Buffer Address (High) Register (GEVNTADRHI)

11.1.3.1.78.1 Offset

Register	Offset
GEVNTADRHI	C404h

11.1.3.1.78.2 Function

This is an alternate register for the GEVNTADRn register.

11.1.3.1.78.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EVNTADRHI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.78.4 Fields

Field	Function
31-0 EVNTADRHI	Event Buffer Address (EvntAdrHi) Holds the higher 32 bits of start address of the external memory for the Event Buffer.

11.1.3.1.79 Global event buffer size register (GEVNTSIZ)

11.1.3.1.79.1 Offset

Register	Offset
GEVNTSIZ	C408h

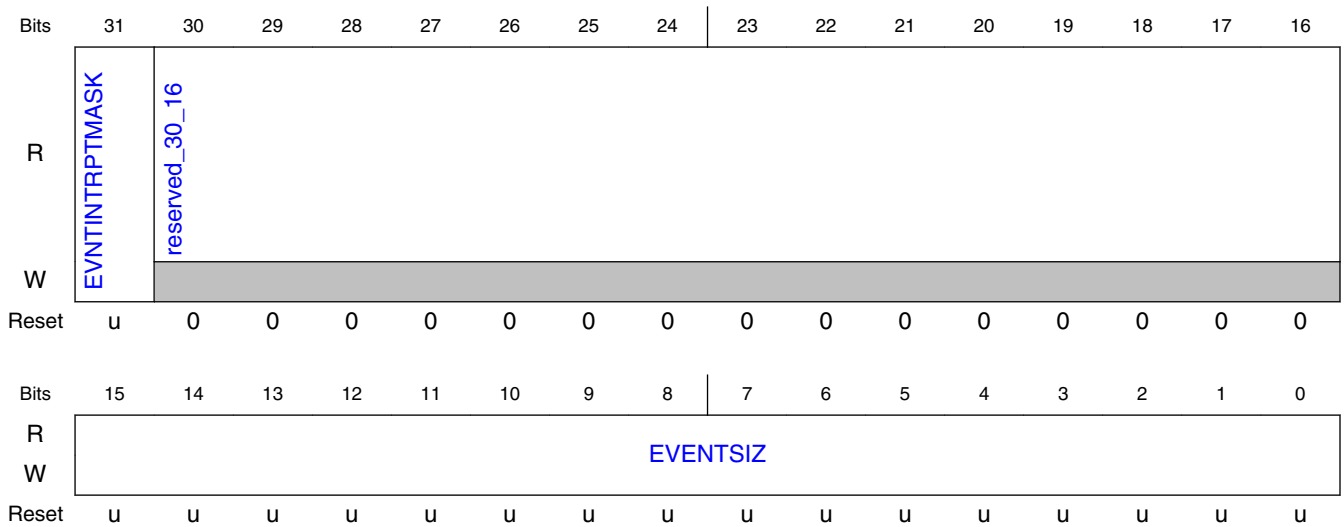
11.1.3.1.79.2 Function

This register holds the Event Buffer Size and the Event Interrupt Mask bit.

Reset Mask:0x7FFF0000

Global Event Buffer Size Register This register holds the Event Buffer Size and the Event Interrupt Mask bit. During power-on initialization, software must initialize the size with the number of bytes allocated for the Event Buffer. The Event Interrupt Mask will mask the interrupt, but events are still queued. After configuration, software must preserve the Event Buffer Size value when changing the Event Interrupt Mask.

11.1.3.1.79.3 Diagram



11.1.3.1.79.4 Fields

Field	Function
31 EVNTINTRPTMASK	Event Interrupt Mask (EvntIntMask).
30-16 reserved_30_16	Reserved
15-0 EVENTSIZ	Event Buffer Size in bytes (EVNTSiz) Holds the size of the Event Buffer in bytes; must be a multiple of four.

11.1.3.1.80 Global event buffer count register (GEVNTCOUNT)

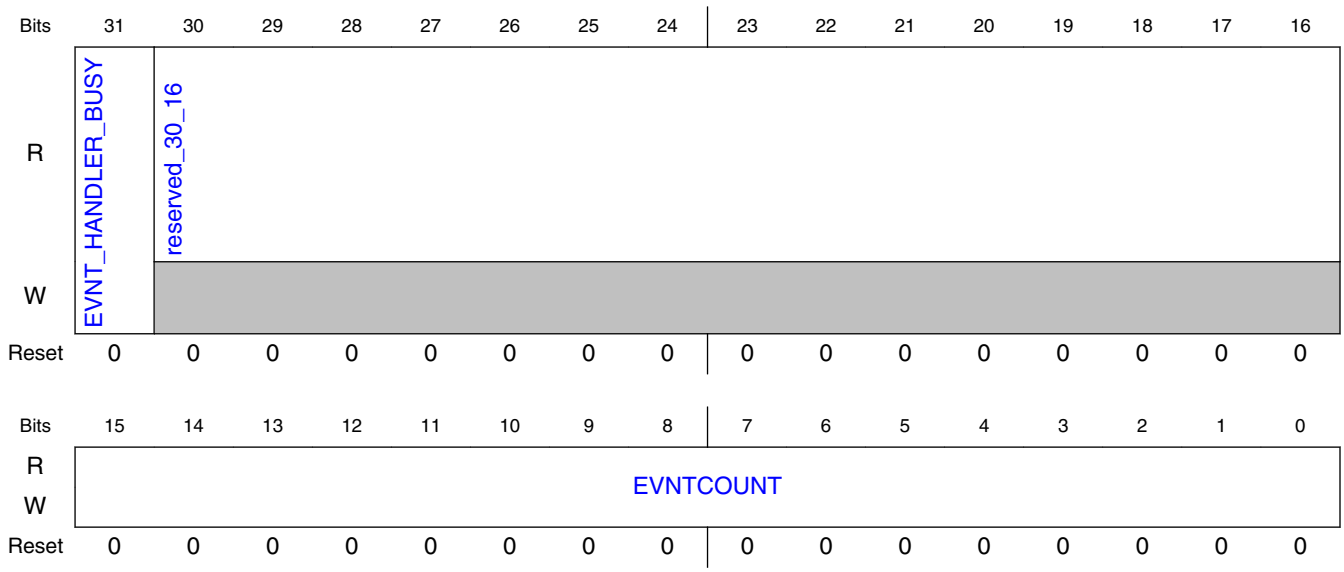
11.1.3.1.80.1 Offset

Register	Offset
GEVNTCOUNT	C40Ch

11.1.3.1.80.2 Function

This register holds the number of valid bytes in the Event Buffer.

11.1.3.1.80.3 Diagram



11.1.3.1.80.4 Fields

Field	Function
31 EVNT_HANDLER_BUSY	Event Handler Busy Device software event handler busy indication.
30-16 reserved_30_16	Reserved
15-0 EVNTCOUNT	Event Count (EVNTCount) When read, returns the number of valid events in the Event Buffer (in bytes).

11.1.3.1.81 Global Hardware Parameters Register 8 (GHWPARAMS8)

11.1.3.1.81.1 Offset

Register	Offset
GHWPARAMS8	C600h

11.1.3.1.81.2 Function

This register contains the hardware configuration options that you can select in the coreConsultant GUI.

11.1.3.1.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ghwparams8_32_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ghwparams8_32_0															
W																
Reset	0	0	0	0	1	0	1	0	0	1	0	1	1	0	0	0

11.1.3.1.81.4 Fields

Field	Function
31-0 ghwparams8_32_0	`DWC_USB3_DCACHE_DEPTH_INFO

11.1.3.1.82 Global Device TX FIFO DMA Priority Register (GTXFIFOPRIDEV)

11.1.3.1.82.1 Offset

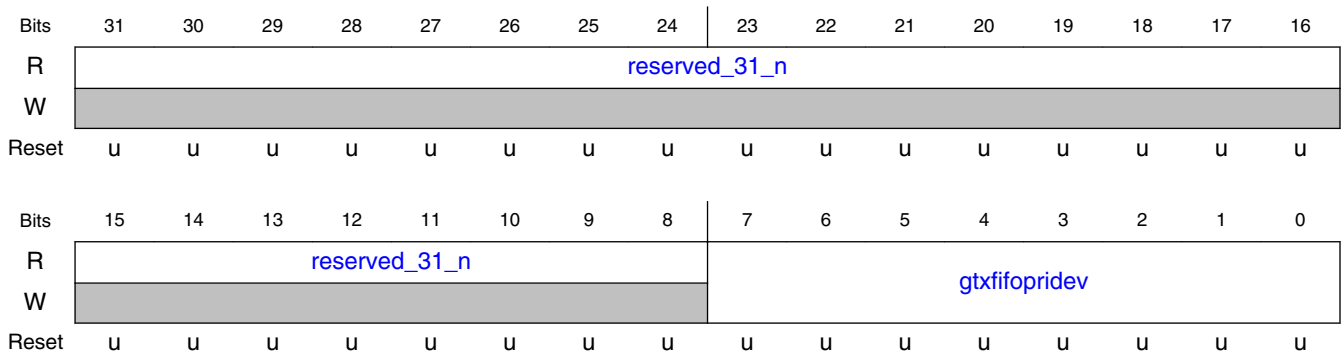
Register	Offset
GTXFIFOPRIDEV	C610h

11.1.3.1.82.2 Function

This register specifies the relative DMA priority level among the Device TXFIFOs (one per IN endpoint).

Reset Mask: 0xFFFF00

11.1.3.1.82.3 Diagram



11.1.3.1.82.4 Fields

Field	Function
31-8 reserved_31_n	Reserved
7-0 gtxfifopridev	Device TxFIFO priority

11.1.3.1.83 Global Host TX FIFO DMA Priority Register (GTXFIFOPRIHST)

11.1.3.1.83.1 Offset

Register	Offset
GTXFIFOPRIHST	C618h

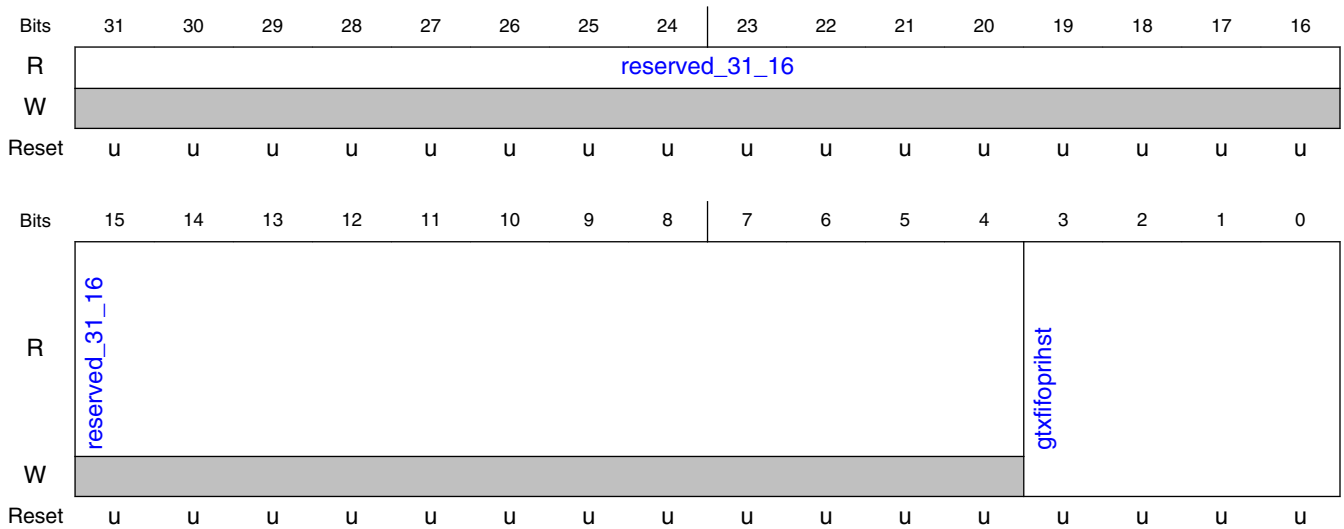
11.1.3.1.83.2 Function

This register specifies the relative DMA priority level among the Host TXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS).

Reset Mask: 0xFFFF0

Reset Default Value: 0x0

11.1.3.1.83.3 Diagram



11.1.3.1.83.4 Fields

Field	Function
31-4 reserved_31_16	Reserved
3-0 gtxfifoprihst	Host TxFIFO priority

11.1.3.1.84 Global Host RX FIFO DMA Priority Register (GRXFIFOPRIHST)

11.1.3.1.84.1 Offset

Register	Offset
GRXFIFOPRIHST	C61Ch

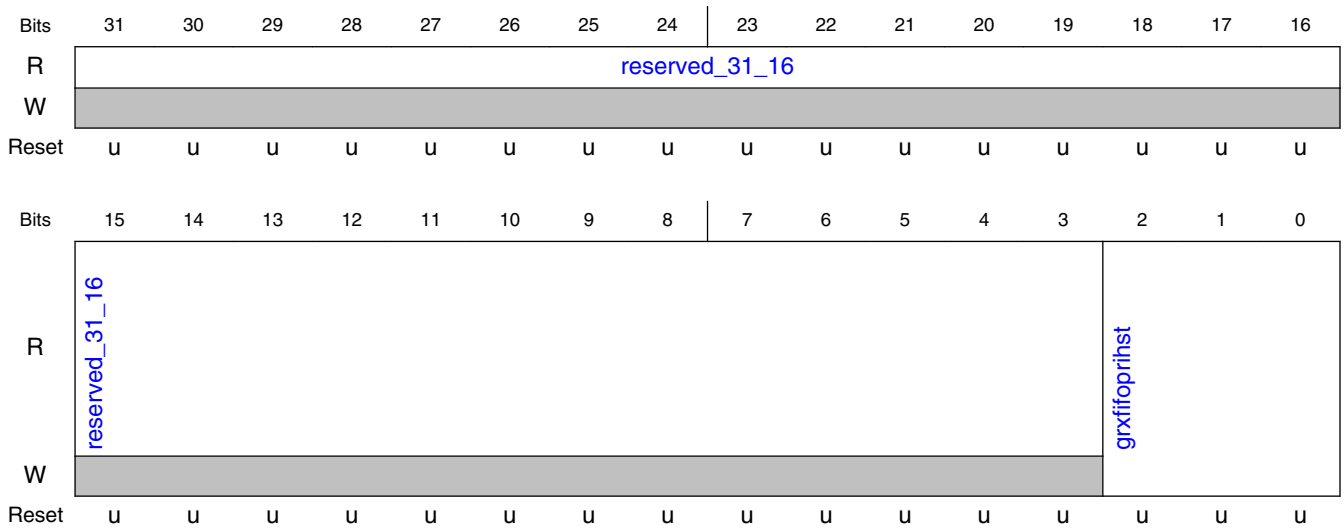
11.1.3.1.84.2 Function

This register specifies the relative DMA priority level among the Host RXFIFOs (one per USB bus instance) within the associated speed group (SS or HS/FSLS).

Reset Mask: 0x7FFF8

Reset Default Value: 0x0

11.1.3.1.84.3 Diagram



11.1.3.1.84.4 Fields

Field	Function
31-3 reserved_31_16	Reserved
2-0 grxfifoprihst	Host RxFIFO priority

11.1.3.1.85 Global Host Debug Capability DMA Priority Register (GFIFOPRIDBC)

11.1.3.1.85.1 Offset

Register	Offset
GFIFOPRIDBC	C620h

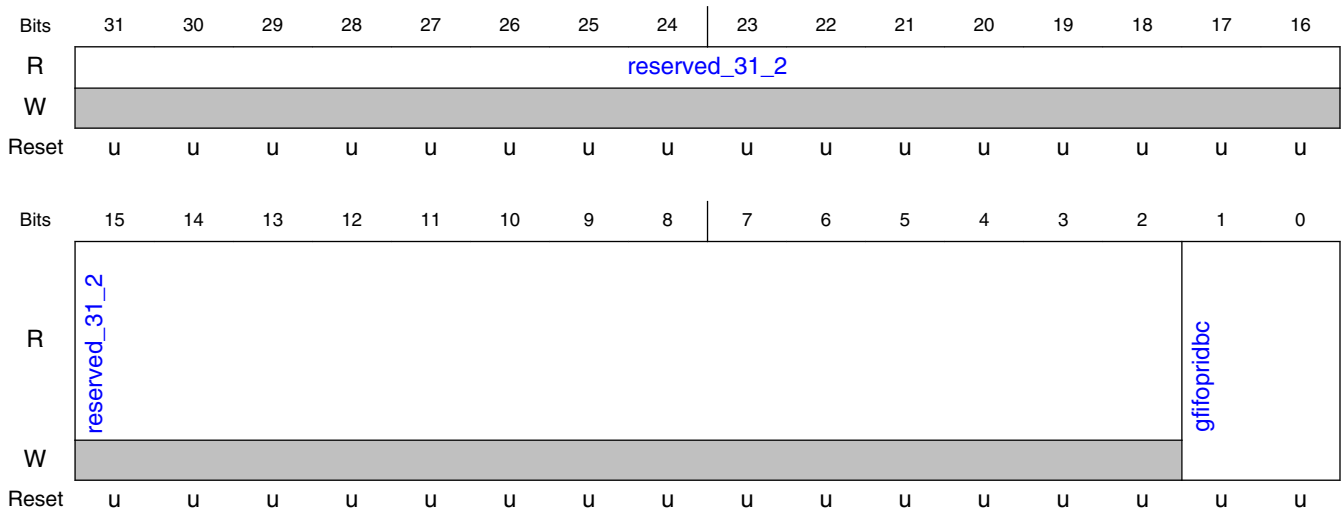
11.1.3.1.85.2 Function

This register specifies the relative priority of the RXFIFOs and TXFIFOs associated with the DbC mode.

Reset Mask: 0xFFFFFFFC

Reset Default Value: 0x0

11.1.3.1.85.3 Diagram



11.1.3.1.85.4 Fields

Field	Function
31-2 reserved_31_2	Reserved
1-0 gfifopridbc	Host DbC DMA priority

11.1.3.1.86 (GDMAHLRATIO)

11.1.3.1.86.1 Offset

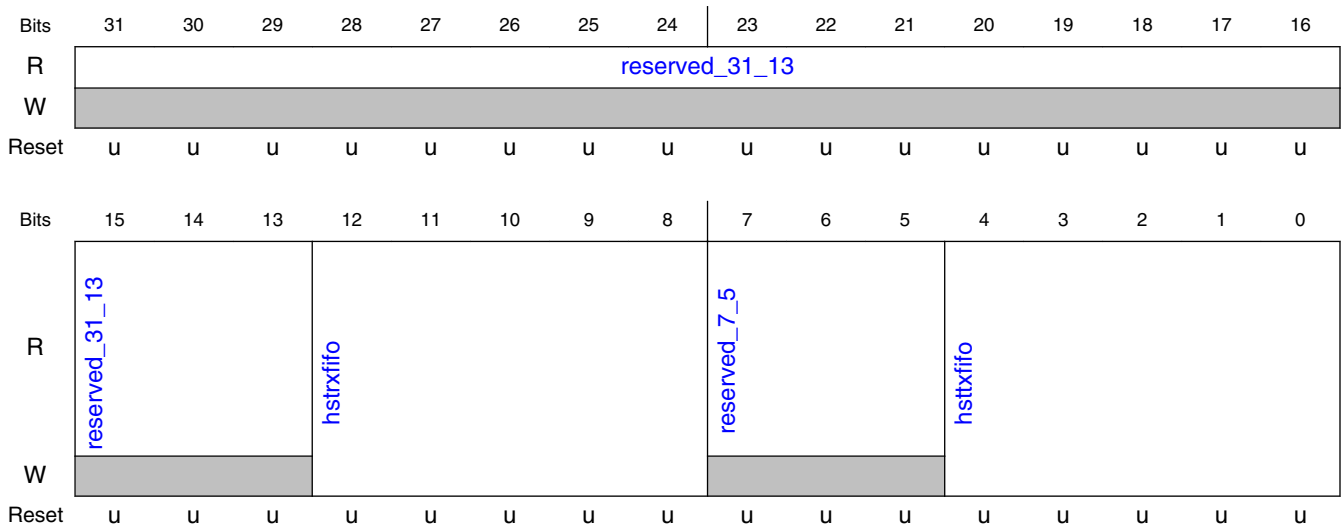
Register	Offset
GDMAHLRATIO	C624h

11.1.3.1.86.2 Function

Global Host FIFO DMA High-Low Priority Ratio Register This register specifies the relative priority of the SS FIFOs with respect to the HS/FSLS FIFOs.

Reset Mask:0xFFFFE0E0

11.1.3.1.86.3 Diagram



11.1.3.1.86.4 Fields

Field	Function
31-13 reserved_31_13	Reserved
12-8 hstrxfifo	Host RXFIFO DMA High-Low Priority
7-5 reserved_7_5	Reserved
4-0 hsttxfifo	Host TXFIFO DMA High-Low Priority

11.1.3.1.87 Global Frame Length Adjustment Register (GFLADJ)

11.1.3.1.87.1 Offset

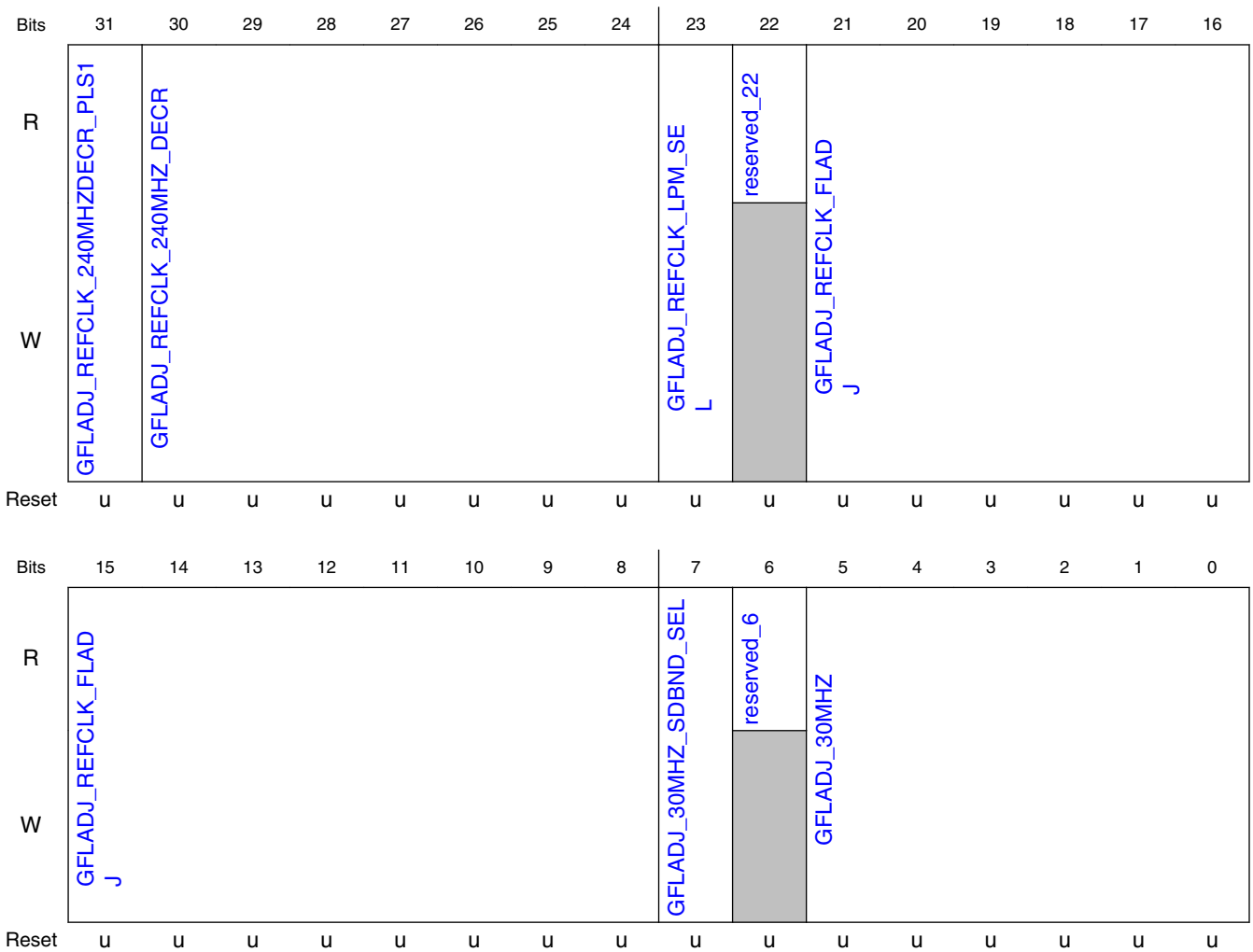
Register	Offset
GFLADJ	C630h

11.1.3.1.87.2 Function

This register provides options for the software to control the core behavior with respect to SOF (Start of Frame) and ITP (Isochronous Timestamp Packet) timers and frame timer functionality.

Reset Mask:0x400040

11.1.3.1.87.3 Diagram



11.1.3.1.87.4 Fields

Field	Function
31 GFLADJ_REFCLK_240MHZDECR_PLS1	GFLADJ_REFCLK_240MHZDECR_PLS1 This field indicates that the decrement value that the controller applies for each ref_clk must be GFLADJ_REFCLK_240MHZ_DECR and GFLADJ_REFCLK_240MHZ_DECR +1 alternatively on each ref_clk. Set this bit to a '1' only if GFLADJ_REFCLK_LPM_SEL is set to '1' and the fractional component of 240/ref_frequency is greater

Table continues on the next page...

Field	Function
	than or equal to 0.5. Examples: If the ref_clk is 19.2 MHz then - GUCTL.REF_CLK_PERIOD = 52 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/19.2) = 12.5 - GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 1 If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = (240/24) = 10 - GFLADJ.GFLADJ_REFCLK_240MHZDECR_PLS1 = 0
30-24 GFLADJ_REFCLK_240MHZ_DECR	This field indicates the decrement value that the controller applies for each ref_clk in order to derive a frame timer in terms of a 240-MHz clock. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1'. The value is derived as follows: GFLADJ_REFCLK_240MHZ_DECR = 240/ref_clk_frequency Examples: If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/24 = 10 If the ref_clk is 48 MHz then - GUCTL.REF_CLK_PERIOD = 20 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/48 = 5 If the ref_clk is 17 MHz then - GUCTL.REF_CLK_PERIOD = 58 - GFLADJ.GFLADJ_REFCLK_240MHZ_DECR = 240/17 = 14
23 GFLADJ_REFCLK_LPM_SEL	This bit enables the functionality of running SOF/ITP counters on the ref_clk. This bit must not be set to '1' if GCTL.SOFITPSYNC bit is set to '1'. Similarly, if GFLADJ_REFCLK_LPM_SEL set to '1', GCTL.SOFITPSYNC must not be set to '1'. When GFLADJ_REFCLK_LPM_SEL is set to '1' the overloading of the suspend control of the USB 2.0 first port PHY (UTMI/ULPI) with USB 3.0 port states is removed. For example, for Synopsys PHY, the COMMONONN signal can be tied to '1'. Note that the ref_clk frequencies supported in this mode are 16/17/19.2/20/24/39.7/40 MHz. The utmi_clk[0] signal of the core must be connected to the FREECLK of the PHY. Note: If you set this bit to '1', the GUSB2PHYCFG.U2_FREECLK_EXISTS bit must be set to '0'.
22 reserved_22	Reserved for future use
21-8 GFLADJ_REFCLK_FLADJ	This field indicates the frame length adjustment to be applied when SOF/ITP counter is running on the ref_clk. This register value is used to adjust the ITP interval when GCTL[SOFITPSYNC] is set to '1'; SOF and ITP interval when GLADJ.GFLADJ_REFCLK_LPM_SEL is set to '1'. This field must be programmed to a non-zero value only if GFLADJ_REFCLK_LPM_SEL is set to '1' or GCTL.SOFITPSYNC is set to '1'. The value is derived as follows: FLADJ_REF_CLK_FLADJ=((125000/ref_clk_period_integer)-(125000/ref_clk_period)) * ref_clk_period where - the ref_clk_period_integer is the integer value of the ref_clk period got by truncating the decimal (fractional) value that is programmed in the GUCTL.REF_CLK_PERIOD field. - the ref_clk_period is the ref_clk period including the fractional value. Examples: If the ref_clk is 24 MHz then - GUCTL.REF_CLK_PERIOD = 41 - GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/41)-(125000/41.6666))*41.6666 = 2032 (ignoring the fractional value) If the ref_clk is 48 MHz then - GUCTL.REF_CLK_PERIOD = 20 - GFLADJ.GLADJ_REFCLK_FLADJ = ((125000/20)-(125000/20.8333))*20.8333 = 5208 (ignoring the fractional value)
7 GFLADJ_30MHZ_SDBND_SEL	GFLADJ_30MHZ_SDBND_SEL This field selects whether to use the input signal fladj_30mhz_reg or the GFLADJ.GFLADJ_30MHZ to adjust the frame length for the SOF/ITP. When this bit is set to, - 1, the controller uses the register field GFLADJ.GFLADJ_30MHZ value - 0, the controller uses the input signal fladj_30mhz_reg value
6 reserved_6	Reserved for future use
5-0 GFLADJ_30MHZ	GFLADJ_30MHZ This field indicates the value that is used for frame length adjustment instead of considering from the sideband input signal fladj_30mhz_reg. This enables post-silicon frame length adjustment in case the input signal fladj_30mhz_reg is connected to a wrong value or is not valid. For details on how to set this value, refer to section 5.2.4, "Frame Length Adjustment Register (FLADJ)," of the xHCI Specification.

11.1.3.1.88 Device Configuration Register (DCFG)

11.1.3.1.88.1 Offset

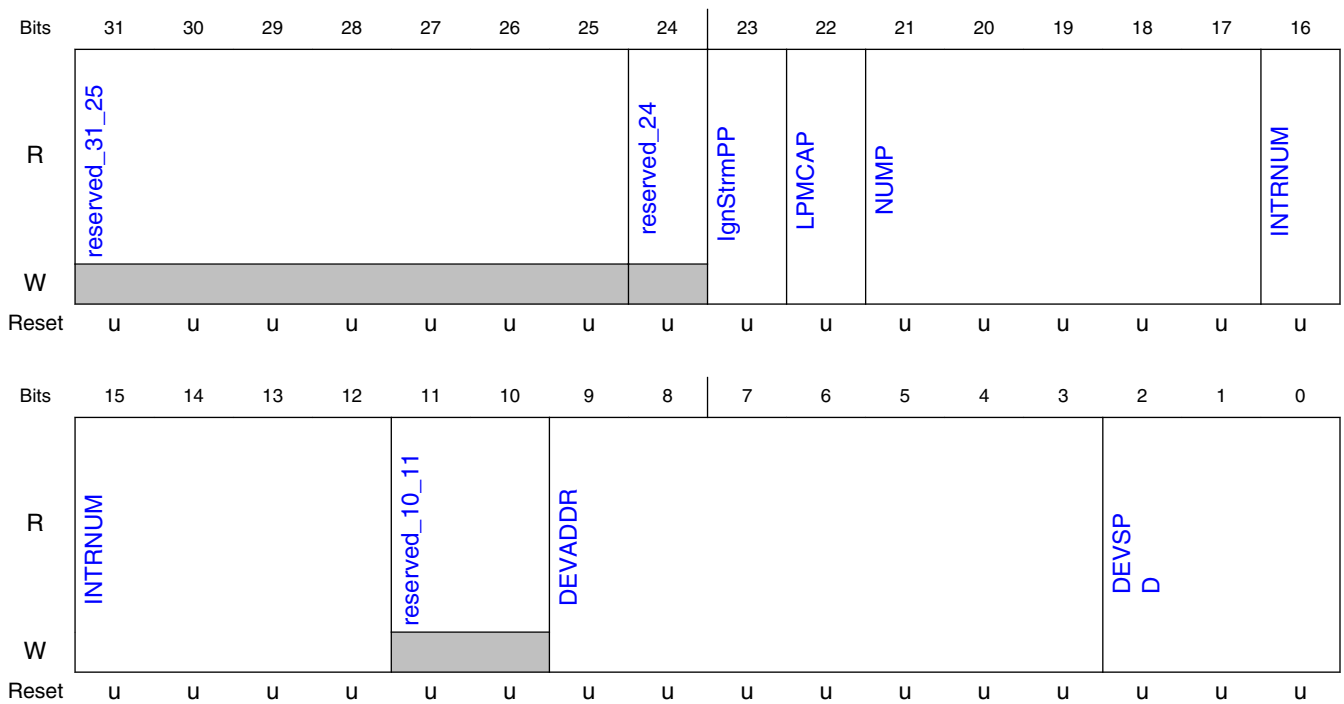
Register	Offset
DCFG	C700h

11.1.3.1.88.2 Function

Reset Mask:0x7F000C00

This register configures the core in device mode after power-on or after certain control commands or enumeration. Do not make changes to this register after initial programming.

11.1.3.1.88.3 Diagram



11.1.3.1.88.4 Fields

Field	Function
31-25 reserved_31_25	Reserved
24	reserved

Table continues on the next page...

Field	Function
reserved_24	
23 IgnStrmPP	IgnoreStreamPP This bit only affects stream-capable bulk endpoints. When this bit is set to '0' and the controller receives a Data Packet with the Packet Pending (PP) bit set to 0 for OUT endpoints, or it receives an ACK with the NumP field set to 0 and PP set to 0 for IN endpoints, the core attempts to search for another stream (CStream) to initiate to the host. However, there are two situations where this behavior is not optimal: - When the host is setting PP=0 even though it has not finished the stream, or - When the endpoint on the device is configured with one transfer resource and therefore does not have any other streams to initiate to the host. When this bit is set to '1', the core ignores the Packet Pending bit for the purposes of stream selection and does not search for another stream when it receives DP(PP=0) or ACK(NumP=0, PP=0). This can enhance the performance when the device system bus bandwidth is low or the host responds to the core's ERDY transmission very quickly.
22 LPMCAP	LPM Capable The application uses this bit to control the DWC_usb3 core LPM capabilities. If the core operates as a non-LPM-capable device, it cannot respond to LPM transactions. - 1'b0: LPM capability is not enabled. - 1'b1: LPM capability is enabled.
21-17 NUMP	Number of Receive Buffers. This bit indicates the number of receive buffers to be reported in the ACK TP. The DWC_usb3 controller uses this field for non-control endpoints if GRXTHRCFG.UsbRxPktCntSel is set to '0'. The application can program this value based on RxFIFO size, buffer sizes programmed in descriptors, and system latency. For an OUT endpoint, this field controls the number of receive buffers reported in the NumP field of the ACK TP transmitted by the core. Note: This bit is used in host mode when Debug Capability is enabled.
16-12 INTRNUM	Interrupt number Indicates interrupt/EventQ number on which non-endpoint-specific device-related interrupts (see DEVT) are generated.
11-10 reserved_10_11	Reserved
9-3 DEVADDR	Device Address. The application must perform the following: - Program this field after every SetAddress request. - Reset this field to zero after USB reset.
2-0 DEVSPD	Device Speed. Indicates the speed at which the application requires the core to connect, or the maximum speed the application can support. However, the actual bus speed is determined only after the chirp sequence is completed, and is based on the speed of the USB host to which the core is connected. - 3'b100: SuperSpeed (USB 3.0 PHY clock is 125 MHz or 250 MHz) - 3'b000: High-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) - 3'b001: Full-speed (USB 2.0 PHY clock is 30 MHz or 60 MHz) 000b - 001b - 100b -

11.1.3.1.89 Device control register (DCTL)

11.1.3.1.89.1 Offset

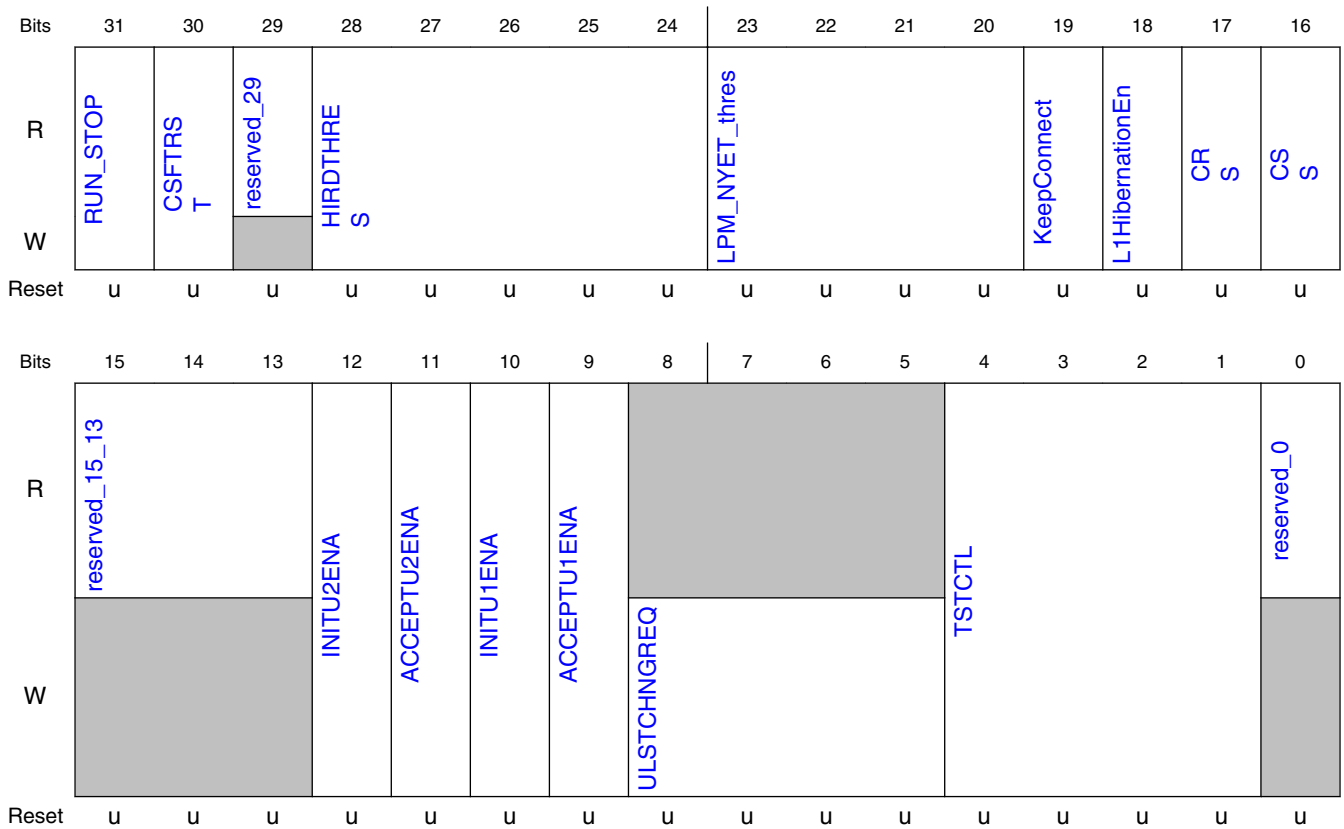
Register	Offset
DCTL	C704h

11.1.3.1.89.2 Function

When hibernation is not enabled using GCTL[GBLHIBERNATIONEN] bit,

- Any value can be written to CSS, CRS, L1HIBERNATIONEN, and KEEPCONNECT bits
- L1HIBERNATIONEN and KEEPCONNET bits always return 0 when read in this hibernation-disabled state

11.1.3.1.89.3 Diagram



11.1.3.1.89.4 Fields

Field	Function
31 RUN_STOP	Run/Stop The software writes 1 to this bit to start the device controller operation. To stop the device controller operation, the software must remove any active transfers and write 0 to this bit. When the controller is stopped, it sets the DSTS.DevCtrlHlt bit when the core is idle and the lower layer finishes the disconnect process. The Run/Stop bit must be used in following cases as specified: - After power-on reset and CSR initialization, the software must write 1 to this bit to start the device controller. The controller does not signal connect to the host until this bit is set. - The software uses this bit to control the device controller to perform a soft disconnect. When the software writes 0 to this bit, the host does not see that the device is connected. The device controller stays in the disconnected state until the software writes 1 to this bit. The minimum duration of keeping this bit cleared is specified in the Note below. If the software attempts a connect after the soft disconnect or detects a disconnect event, it must set DCTL[8:5] to 5 before reasserting the Run/Stop bit. - When the USB or Link is in a lower power state and the Two Power Rails configuration is selected, software writes 0 to this bit to indicate that it is going to turn off the Core Power Rail. After the software turns on the Core Power Rail again and re-initializes the device

Table continues on the next page...

Field	Function
	controller, it must set this bit to start the device controller. For more details, see "Low Power Operation" section of the Databook. Note: The following is the minimum duration under various conditions for which the soft disconnect (SftDiscon) bit must be set for the USB host to detect a device disconnect: 30ms: - For SuperSpeed, when the device state is Suspended, Idle, Transmit, or Receive. 10ms: - For high-speed, when the device state is Suspended, Idle, or not Idle/Suspended (performing transactions) - For full-speed/low-speed, when the device state is Suspended, Idle, or not Idle/Supended (performing transactions) To accommodate clock jitter, it is recommended that the application add extra delay to the specified minimum duration.
30 CSFTRST	Core Soft Reset Resets the all clock domains as follows: - This bit clears the interrupts and all the CSRs except GSTS, GSNPSID, GGPIO, GUID, GUSB2PHYCFGn registers, GUSB3PIPECTLn registers, DCFG, DCTL, DEVTEN, and DSTS registers. - All module state machines (except the SoC Bus Slave Unit) are reset to the IDLE state, and all the TxFIFOs and the RxFIFO are flushed. - Any transactions on the SoC bus Master are terminated as soon as possible, after gracefully completing the last data phase of a SoC bus transfer. Any transactions on the USB are terminated immediately. The application can write this bit at any time to reset the core. This is a self-clearing bit; the core clears this bit after all necessary logic is reset in the core, which may take several clocks depending on the core's current state. Once this bit is cleared, the software must wait at least 3 PHY clocks before accessing the PHY domain (synchronization delay). Typically, software reset is used during software development and also when you dynamically change the PHY selection bits in the USB configuration registers listed above. When you change the PHY, the corresponding clock for the PHY is selected and used in the PHY domain. Once a new clock is selected, the PHY domain must be reset for proper operation. Note: Programming this field with random data causes side effect . Bit Bash register testing is not recommended.
29 reserved_29	Reserved1
28-24 HIRDTHRES	HIRD Threshold (HIRD_Thres) The core asserts output signals utmi_l1_suspend_n and utmi_sleep_n (see "LPM Interface Signals" table in the Databook) on the basis of this signal: The core asserts utmi_l1_suspend_n to put the PHY into Deep Low-Power mode in L1 when both of the following are true: - HIRD value is greater than or equal to the value in DCTL.HIRD_Thres[3:0] - HIRD_Thres[4] is set to 1'b1. The core asserts utmi_sleep_n on L1 when one of the following is true: - If the HIRD value is less than HIRD_Thres[3:0] or - HIRD_Thres[4] is set to 1'b0. Note: This field must be set to '0' during SuperSpeed mode of operation.
23-20 LPM_NYET_thres	LPM NYET Threshold When LPM Errata is enabled: Bits [23:20]: LPM NYET Response Threshold (LPM_NYET_thres) Handshake response to LPM token specified by device application. Response depends on DCFG.LPMCap. - DCFG.LPMCap is 1'b0 - The core always responds with Timeout (that is, no response). - DCFG.LPMCap is 1'b1 - The core responds with an ACK on successful LPM transaction, which requires that all of the following are satisfied: - There are no PID or CRC5 errors in both the EXT token and the LPM token (if not true, inactivity results in a timeout ERROR). - No data is pending in the TxFIFO and RxFIFO is empty (else NYET). - The BESL value in the LPM token is less than or equal to LPM_NYET_thres[3:0]
19 KeepConnect	Keep Connect When '1', this bit enables the save and restore programming model by preventing the core from disconnecting from the host when DCTL.RunStop is set to '0'. It also enables the Hibernation Request Event to be generated when the link goes to U3 or L2. The device core disconnects from the host when DCTL.RunStop is set to '0'. This bit indicates whether to preserve this behavior ('0'), or if the core must not disconnect when RunStop is set to 0 ('1'). This bit also prevents the LTSSM from automatically going to U0/L0 when the host requests resume from U3/L2. Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.
18 L1HibernationEn	L1HibernationEn When this bit is set along with KeepConnect, the device core generates a Hibernation Request Event if L1 is enabled and the HIRD value in the LPM token is larger than the threshold programmed in DCTL.HIRD_Thres. The core does not exit the LPM L1 state until software writes Recovery into the DCTL.ULStChngReq field. This prevents corner cases where the device is entering hibernation at the same time the host is attempting to exit L1. Note: If Hibernation is disabled, that is, GCTL[1].GblHibernationEn = 0, this bit is tied to zero.
17	Controller Restore State (CRS) This command is similar to the USBCMD.CRS bit in host mode and initiates the restore process. When software sets this bit to '1', the controller immediately sets DSTS.RSS

Table continues on the next page...

Field	Function
CRS	to '1'. When the controller has finished the restore process, it sets DSTS.RSS to '0'. Note: When read, this field always returns '0'.
16 CSS	Controller Save State (CSS) This command is similar to the USBCMD.CSS bit in host mode and initiates the save process. When software sets this bit to '1', the controller immediately sets DSTS.SSS to '1'. When the controller has finished the save process, it sets DSTS.SSS to '0'. Note: When read, this field always returns '0'.
15-13 reserved_15_13	Reserved
12 INITU2ENA	Initiate U2 Enable - 1'b0: May not initiate U2 (default) - 1'b1: May initiate U2 On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U2_ENABLE), and clears this bit when ClearFeature(U2_ENABLE) is received. If DCTL[11] (AcceptU2Ena) is 0, the link immediately exits U2 state.
11 ACCEPTU2ENA	Accept U2 Enable - 1'b0: Reject U2 except when Force_LinkPM_Accept bit is set (default) - 1'b1: Core accepts transition to U2 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.
10 INITU1ENA	Initiate U1 Enable - 1'b0: May not initiate U1 (default); - 1'b1: May initiate U1. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving SetFeature(U1_ENABLE), and clears this bit when ClearFeature(U1_ENABLE) is received. If DCTL[9] (AcceptU1Ena) is 0, the link immediately exits U1 state.
9 ACCEPTU1ENA	Accept U1 Enable - 1'b0: Core rejects U1 except when Force_LinkPM_Accept bit is set (default) - 1'b1: Core accepts transition to U1 state if nothing is pending on the application side. On USB reset, hardware clears this bit to 0. Software sets this bit after receiving a SetConfiguration command.
8-5 ULSTCHNGREQ	ULSTCHNGREQ Software writes this field to issue a USB/Link state change request. A change in this field indicates a new request to the core. If software wants to issue the same request back-to-back, it must write a 0 to this field between the two requests. The result of the state change request is reflected in the USB/Link State in DSTS. These bits are self-cleared on the MAC Layer exiting suspended state. If software is updating other fields of the DCTL register and not intending to force any link state change, then it must write a 0 to this field. SS Compliance mode is normally entered and controlled by the remote link partner. Refer to the USB 3.0 specification. Alternatively, you can force the local link directly into compliance mode, by resetting the SS link with the RUN/STOP bit set to zero. If you then write '10' to the USB/Link State Change field and '1' to RUN/STOP, the link goes to compliance mode. Once you are in compliance, you may alternately write zero and '10' to this field to advance the compliance pattern. In SS mode: - Value Requested Link State Transition/Action - 0 No Action - 4 SS.Disabled - 5 Rx.Detect - 6 SS.Inactive - 8 Recovery - 10 Compliance - Others: Reserved In HS/FS/LS mode: - ValueRequested USB state transition - 8 Remote wakeup request - Others: Reserved The Remote wakeup request must be issued 2us after the device goes into suspend state (DSTS[21:18] is 3 - refer to Table "Fields for Register: DSTS" in the Databook). Note: After coming out of hibernation, software must write 8 (Recovery) into this field to confirm exit from the suspended state.
4-1 TSTCTL	Test Control - 4'b000: Test mode disabled - 4'b001: Test_J mode - 4'b010: Test_K mode - 4'b011: Test_SE0_NAK mode - 4'b100: Test_Packet mode - 4'b101: Test_Force_Enable - Others: Reserved
0 reserved_0	Reserved

11.1.3.1.90 Device Event Enable Register (DEVTEN)

11.1.3.1.90.1 Offset

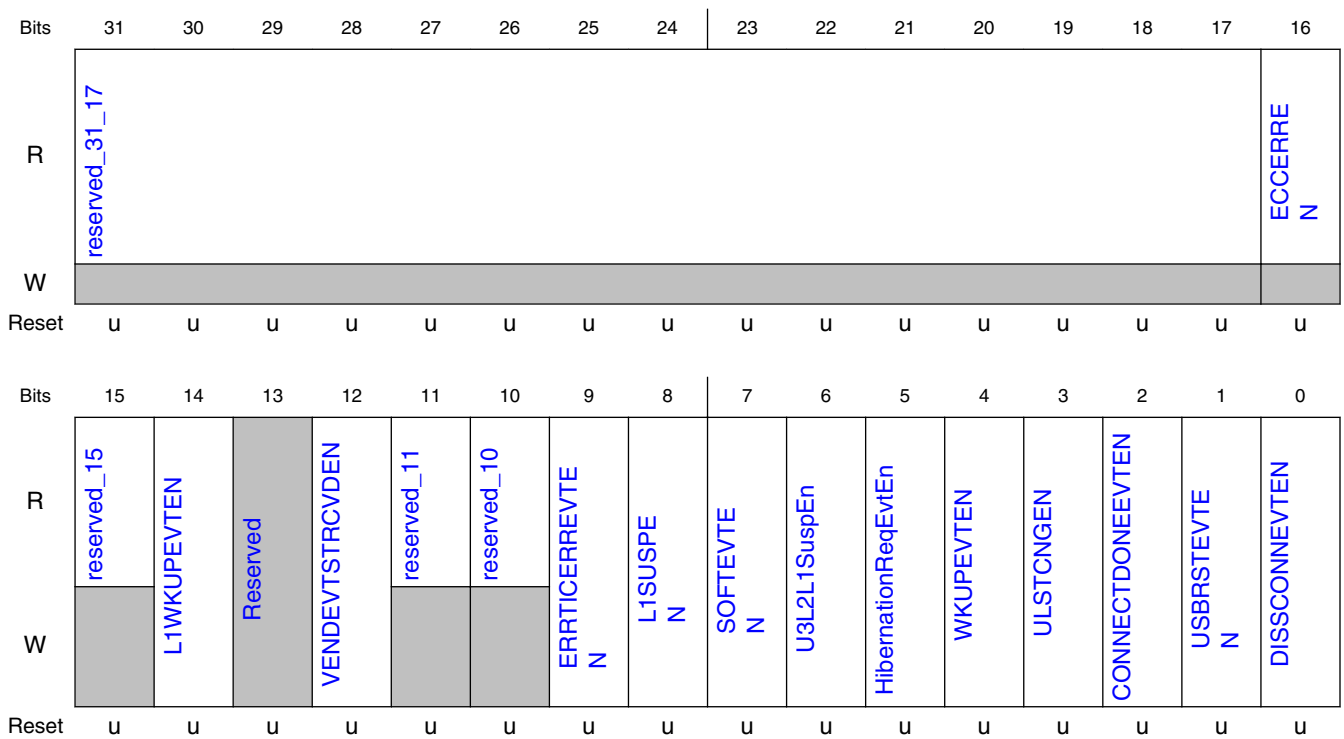
Register	Offset
DEVTEN	C708h

11.1.3.1.90.2 Function

This register controls the generation of device-specific events (see "Event Buffer Content for Device-Specific Events (DEVT)")

Reset Mask:0xFFFE0C00

11.1.3.1.90.3 Diagram



11.1.3.1.90.4 Fields

Field	Function
31-17 reserved_31_17	Reserved
16 ECCERREN	ECC Error Enable. If this bit is set to 1, the controller reports an ECC error to the software when an uncorrectable ECC occurs internally.

Table continues on the next page...

Field	Function
15 reserved_15	Reserved
14 L1WKUPEVTEN	L1 Resume Detected Event Enable. Note: If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for L1 Resume Detected Event Enable.
13 —	Reserved
12 VENDEVTSTRCVDEN	Vendor Device Test LMP Received Event (VndrDevTstRcvedEn)
11 reserved_11	Reserved
10 reserved_10	Reserved
9 ERRTICERREVTEN	Erratic Error Event Enable
8 L1SUSPEN	L1 Suspend Event Enable Note: Only if GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, this bit is for L1 Suspend Event Enable.
7 SOFTEVTEN	Start of (u)frame
6 U3L2L1SuspEn	U3/L2 or U3/L2L1 Suspend Event Enable. Note: - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Suspend Event Enable. - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Suspend Event Enable.
5 HibernationReqEvtEn	This bit enables/disables the generation of the Hibernation Request Event.
4 WKUPEVTEN	U3/L2 or U3/L2L1 Resume Detected Event Enable. Note: - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is enabled, then this bit is for U3/L2 Resume Detected Event Enable. - If GUCTL1[DEV_DECOUPLE_L1L2_EVT] is not enabled, then this bit is for U3/L2L1 Resume Detected Event Enable.
3 ULSTCNGEN	USB/Link State Change Event Enable
2 CONNECTDONEEVTEN	Connection Done Enable
1 USBRSTEVTEN	USB Reset Enable
0 DISSCONNEVTEN	Disconnect Detected Event Enable

11.1.3.1.91 Device Status Register (DSTS)

11.1.3.1.91.1 Offset

Register	Offset
DSTS	C70Ch

11.1.3.1.91.2 Function

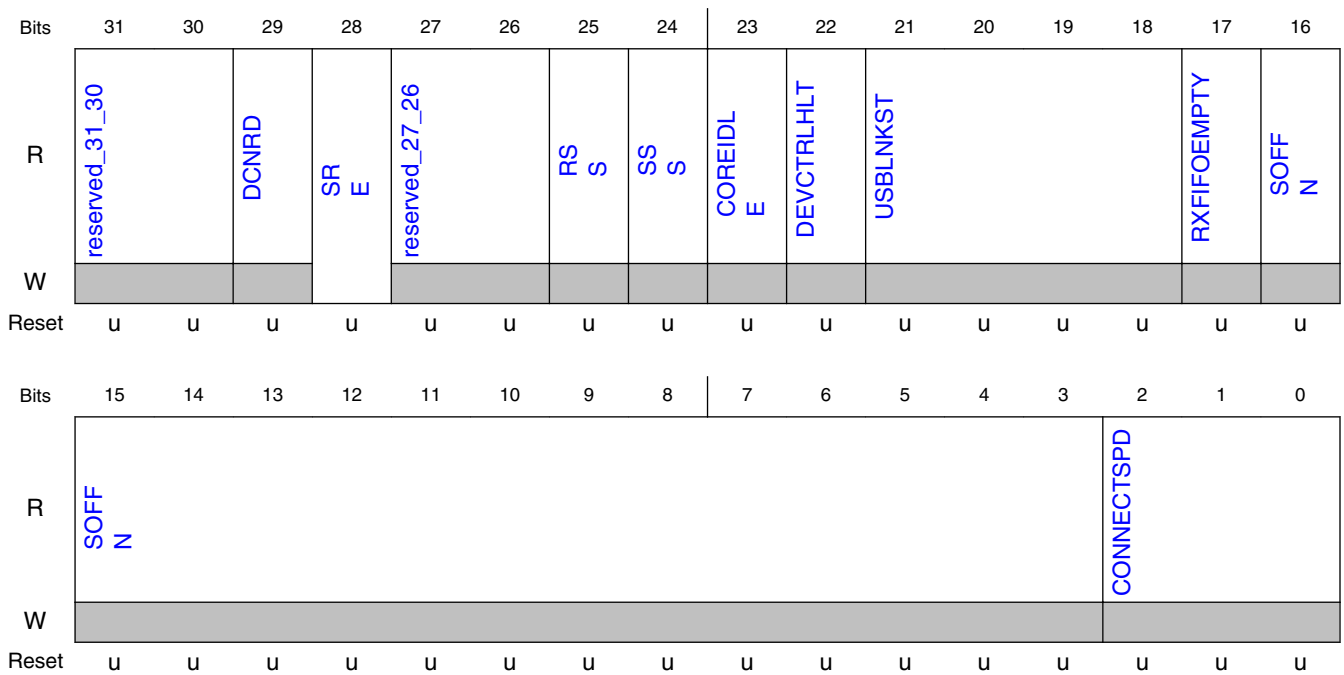
Device Status Register This register indicates the status of the device controller with respect to USB-related events.

Reset Mask:0xDC000000

NOTE

When hibernation is not enabled, RSS and SSS bits always return 0 when read.

11.1.3.1.91.3 Diagram



11.1.3.1.91.4 Fields

Field	Function
31-30 reserved_31_30	Reserved
29 DCNRD	Device Controller Not Ready The bit indicates that the core is in the process of completing the state transitions after exiting from hibernation. To complete the state transitions, it takes 256 bus clock cycles from the time DCTL[31].Run/Stop is set. During hibernation, if the UTMI/ULPI PHY is in suspended state, then the 256-bus clock cycle delay starts after the PHY exited suspended state. Software must set DCTL[31].Run/Stop to '1' and wait for this bit to be de-asserted to zero before processing DSTS.USBLnkSt. This bit is valid only when DWC_USB3_EN_PWROPT is set to 2 and GCTL[1].GblHibernationEn =1.
28 SRE	Save Restore Error. Currently not supported.
27-26 reserved_27_26	Reserved
25 RSS	RSS Restore State Status This bit is similar to the USBSTS.RSS in host mode. When the controller finishes the restore process, it completes the command by setting DSTS.RSS to '0'.
24 SSS	SSS Save State Status This bit is similar to the USBSTS.SSS in host mode. When the controller has finished the save process, it completes the command by setting DSTS.SSS to '0'.
23 COREIDLE	Core Idle The bit indicates that the core finished transferring all RxFIFO data to system memory, writing out all completed descriptors, and all Event Counts are zero. Note: While testing for Reset values, mask out the read value. This bit represents the changing state of the core and does not hold a static value.
22 DEVCTRLHLT	Device Controller Halted This bit is set to 0 when the Run/Stop bit in the DCTL register is set to 1. The core sets this bit to 1 when, after SW sets Run/Stop to 0, the core is idle and the lower layer finishes the disconnect process. When Halted=1, the core does not generate Device events. Note: - The core does not set this bit to 1 if GEVNTCOUNTn has some valid value. Software needs to acknowledge the events that are generated (by writing to GEVNTCOUNTn) while it is waiting for this bit to be set to 1. - When Interrupt Moderation is enabled, there could be delay in raising the interrupt line when the event count is non-zero. Software should read the GEVNTCOUNT register directly and acknowledge them.
21-18 USBLNKST	USBLNKST. USB/Link State In SS mode: LTSSM State - 4'h0: U0 - 4'h1: U1 - 4'h2: U2 - 4'h3: U3 - 4'h4: SS_DIS - 4'h5: RX_DET - 4'h6: SS_INACT - 4'h7: POLL - 4'h8: RECOV - 4'h9: HRESET - 4'ha: CMPLY - 4'hb: LPBK - 4'hf: Resume/Reset In HS/FS/LS mode: - 4'h0: On state - 4'h2: Sleep (L1) state - 4'h3: Suspend (L2) state - 4'h4: Disconnected state (Default state) - 4'h5: Early Suspend state (valid only when Hibernation is disabled, GCTL[1].GblHibernationEn = 0) - 4'he: Reset (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) - 4'hf: Resume (valid only when Hibernation is enabled, GCTL[1].GblHibernationEn = 1) The link state Resume/Reset indicates that the core received a resume or USB reset request from the host while the link was in hibernation. Software must write '8' (Recovery) to the DCTL.ULStChngReq field to acknowledge the resume/reset request. When Hibernation is enabled, GCTL[1].GblHibernationEn = 1, this field USBLnkSt is valid only when DCTL[31].Run/Stop set to '1' and DSTS[29].DCNRD = 0. Note: If SSIC is enabled, while exiting a low power state, the USBLnkSt field indicates Resume/Reset even for a disconnect condition because a resume precedes the disconnect.
17 RXFIFOEMPTY	RxFIFO Empty.
16-3 SOFFN	Frame/Microframe Number of the Received SOF. When the core is operating at SuperSpeed, - [16:3] indicates the uframe/ITP number When the core is operating at high-speed, - [16:6] indicates the frame number - [5:3] indicates the microframe number When the core is operating at full-speed, - [16:14] is not used. Software can ignore these 3 bits - [13:3] indicates the frame number
2-0 CONNECTSPD	Connected Speed (ConnectSpd) Indicates the speed at which the DWC_usb3 core has come up after speed detection through a chirp sequence. - 3'b100: SuperSpeed (PHY clock is running at 125 or 250

Field	Function
	MHz) - 3'b000: High-speed (PHY clock is running at 30 or 60 MHz) - 3'b001: Full-speed (PHY clock is running at 30 or 60 MHz) Low-speed is not supported for devices using a UTMI+ PHY. 000b - 001b - 100b -

11.1.3.1.92 Device Generic Command Parameter Register (DGCMDPAR)

11.1.3.1.92.1 Offset

Register	Offset
DGCMDPAR	C710h

11.1.3.1.92.2 Function

This register indicates the device command parameter.

11.1.3.1.92.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.92.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.1.3.1.93 (DGCMD)

11.1.3.1.93.1 Offset

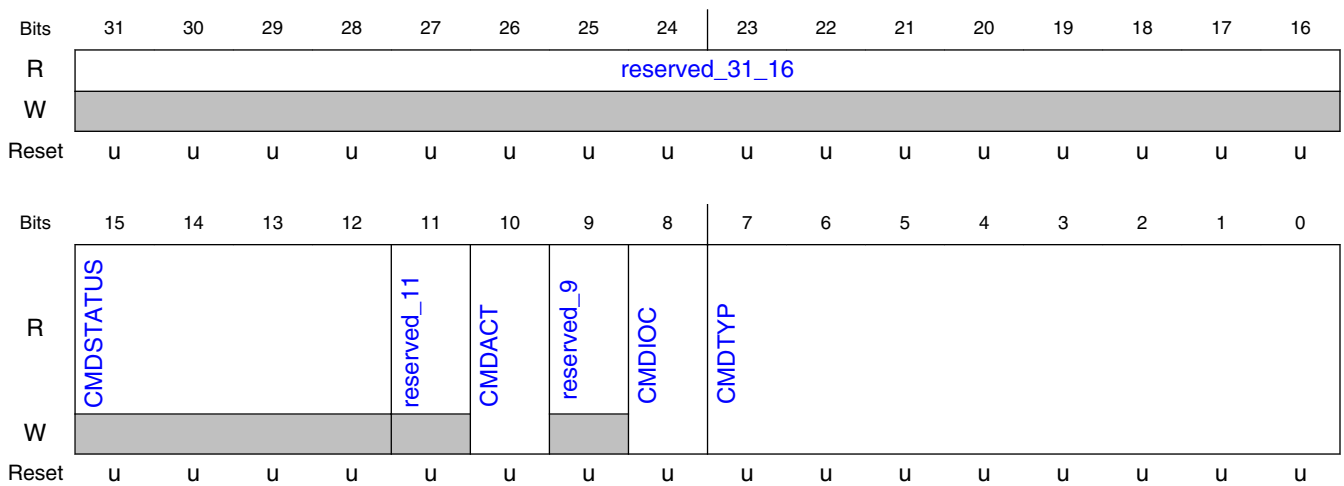
Register	Offset
DGCMD	C714h

11.1.3.1.93.2 Function

Device Generic Command Register This register enables software to program the core using a single generic command interface to send link management packets and notifications.

Reset Mask:0xFFFF0A00

11.1.3.1.93.3 Diagram



11.1.3.1.93.4 Fields

Field	Function
31-16 reserved_31_16	Reserved
15-12 CMDSTATUS	Command Status - 1: CmdErr: Indicates that the device controller encountered an error while processing the command. - 0: Indicates command success
11 reserved_11	Reserved
10 CMDACT	Command Active The software sets this bit to 1 to enable the device controller to execute the generic command. The device controller sets this bit to 0 after executing the command.
9	Reserved

Table continues on the next page...

Field	Function
reserved_9	
8 CMDIOC	Command Interrupt on Complete When this bit is set, the device controller issues a Generic Command Completion event after executing the command. Note that this interrupt is mapped to DCFG.IntrNum. Note: This field must not set to '1' if the DCTL.RunStop field is '0'.
7-0 CMDTYP	CMDTYP Generic Command Type Specifies the type of generic command the software driver is requesting the controller to perform. - 02h: Set Periodic Parameters - 04h: Set Scratchpad Buffer Array Address Lo - 05h: Set Scratchpad Buffer Array Address Hi - 07h: Transmit Device Notification - 09h: Selected FIFO Flush - 0Ah: All FIFO Flush - 0Ch: Set Endpoint NRDY - 10h: Run SoC Bus LoopBack Test - 11h: Restart After Disconnect All other values are reserved.

11.1.3.1.94 Device Active USB Endpoint Enable Register (DALEPENA)

11.1.3.1.94.1 Offset

Register	Offset
DALEPENA	C720h

11.1.3.1.94.2 Function

This register indicates whether a USB endpoint is active in a given configuration or interface.

11.1.3.1.94.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	USBACTEP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	USBACTEP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.94.4 Fields

Field	Function
31-0 USBACTEP	USB active endpoints This bit indicates if a USB endpoint is active in the current configuration and interface.

Field	Function
	Bit[0]: USB EP0-OUT Bit[1]: USB EP0-IN Bit[2]: USB EP1-OUT Bit[3]: USB EP1-IN Bit[4]: USB EP2-OUT Bit[5]: USB EP2-IN Bit[6]: USB EP3-OUT Bit[7]: USB EP3-IN The entity programming this register must set bits 0 and 1 because they enable control endpoints that map to physical endpoints (resources) after USB reset. Hardware clears these bits for all endpoints (other than EP0-OUT and EP0-IN) after detecting a USB reset event. After receiving SetConfiguration and SetInterface requests, the application must program endpoint registers accordingly and set these bits.

11.1.3.1.95 Device physical endpoint-n command parameter 2 register (DEPCMDPAR2)

11.1.3.1.95.1 Offset

Register	Offset
DEPCMDPAR2	C800h

11.1.3.1.95.2 Function

This register indicates the physical endpoint command Parameter 2.

11.1.3.1.95.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.95.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.1.3.1.96 Device Physical Endpoint-n Command Parameter 1 Register (DEPCMDPAR1)

11.1.3.1.96.1 Offset

Register	Offset
DEPCMDPAR1	C804h

11.1.3.1.96.2 Function

This register indicates the physical endpoint command parameter 1. It must be programmed before issuing the command.

11.1.3.1.96.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.96.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.1.3.1.97 Device Physical Endpoint-n Command Parameter 0 Register (DEPCMDPAR0)

11.1.3.1.97.1 Offset

Register	Offset
DEPCMDPAR0	C808h

11.1.3.1.97.2 Function

This register indicates the physical endpoint command parameter 0. This must be programmed before or along with the command. For commands needing only one 32-bit parameter, this register must be programmed with the command register.

11.1.3.1.97.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PARAMETER															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.97.4 Fields

Field	Function
31-0 PARAMETER	PARAMETER

11.1.3.1.98 Device Physical Endpoint-n Command Register (DEPCMD)

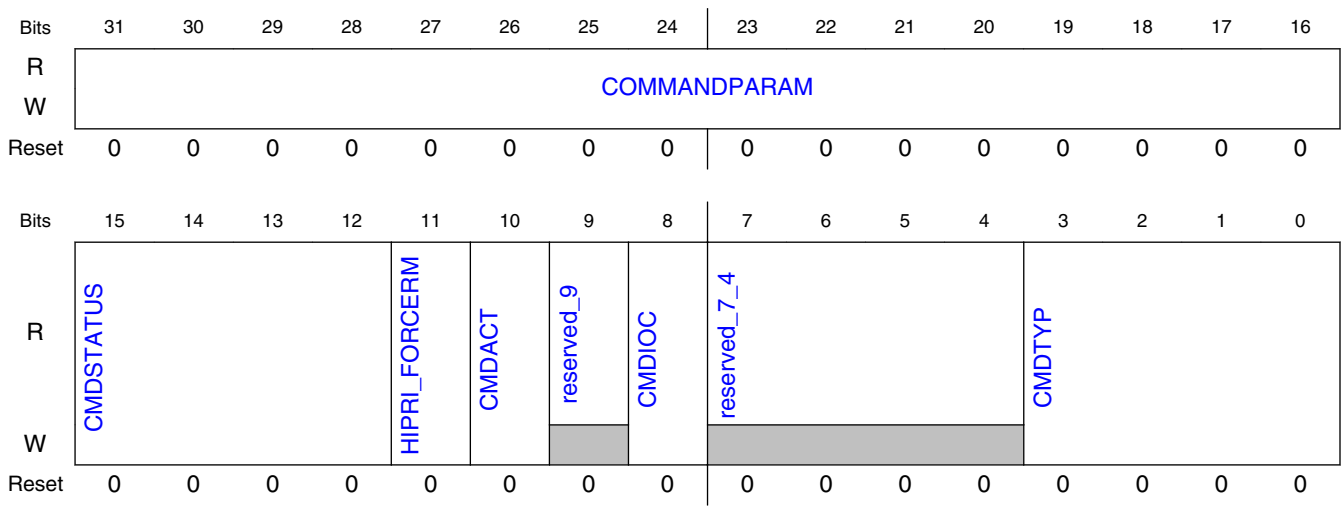
11.1.3.1.98.1 Offset

Register	Offset
DEPCMD	C80Ch

11.1.3.1.98.2 Function

This register enables software to issue physical endpoint-specific commands.

11.1.3.1.98.3 Diagram



11.1.3.1.98.4 Fields

Field	Function
31-16 COMMANDPARAM	Command Parameters or Event Parameters Command Parameters (CommandParam), when this register is written: For Start Transfer command: - [31:16]: StreamID.
15-12 CMDSTATUS	Command Completion Status (CmdStatus) Additional information about the completion of this command is available in this field.
11 HIPRI_FORCERM	HighPriority/ForceRM (HiPri_ForceRM) - HighPriority: Only valid for Start Transfer command - ForceRM: Only valid for End Transfer command - ClearPendIN: Only valid for Clear Stall command .
10 CMDACT	Command Active (CmdAct) Software sets this bit to 1 to enable the device endpoint controller to execute the generic command.
9 reserved_9	Reserved
8 CMDIOC	CMDIOC Command Interrupt on Complete (CmdIOC) When this bit is set, the device controller issues a generic Endpoint Command Complete event after executing the command.
7-4 reserved_7_4	Reserved

Table continues on the next page...

Field	Function
3-0 CMDTYP	Command Type Specifies the type of command the software driver is requesting the core to perform. 01h: Set Endpoint Configuration 64 or 96-bit Parameter

11.1.3.1.99 Device Interrupt Moderation Register (DEV_IMOD)

11.1.3.1.99.1 Offset

Register	Offset
DEV_IMOD	CA00h

11.1.3.1.99.2 Function

the Interrupt Moderation feature that allows the device software to throttle the interrupt rate.

11.1.3.1.99.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEVICE_IMODC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEVICE_IMODI															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.1.3.1.99.4 Fields

Field	Function
31-16 DEVICE_IMODC	Interrupt Moderation Down Counter Loaded with the DEVICE_IMODI value, whenever the hardware interrupt(n) line is de-asserted from the asserted state, counts down to 0, and stops.
15-0 DEVICE_IMODI	Moderation Interval (DEVICE_IMODI) This field holds the minimum inter-interrupt interval between events.

11.1.3.1.100 OTG Configuration Register (OCFG)

11.1.3.1.100.1 Offset

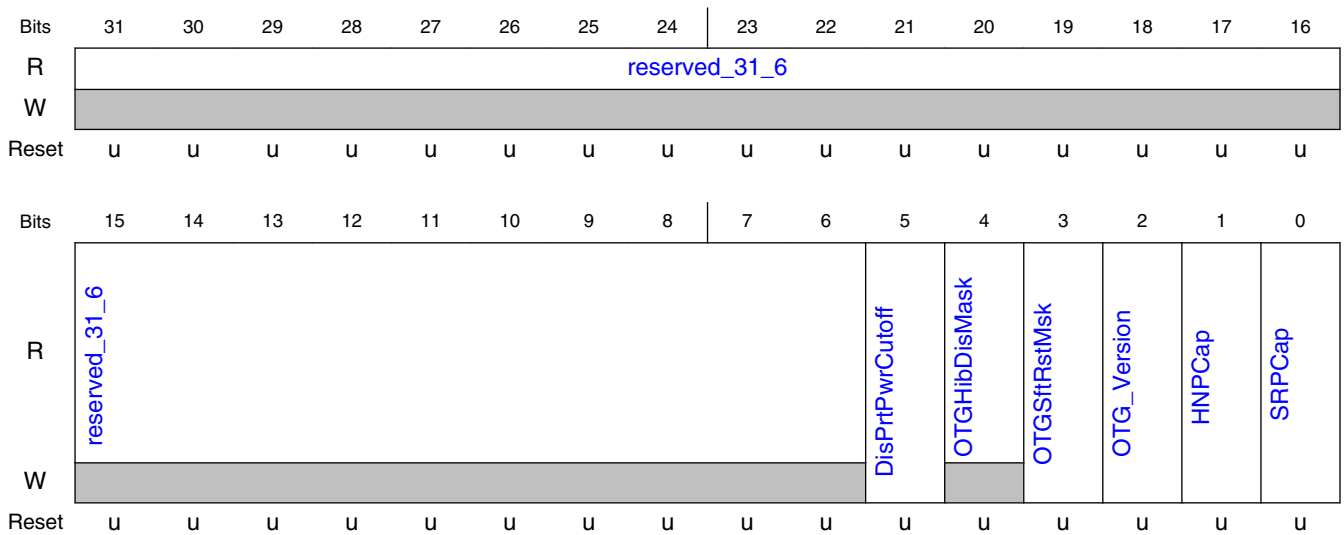
Register	Offset
OCFG	CC00h

11.1.3.1.100.2 Function

This register specifies the HNP and SRP capability of the DWC_usb3 core.

Reset Mask:0xFFFFFFC0

11.1.3.1.100.3 Diagram



11.1.3.1.100.4 Fields

Field	Function
31-6 reserved_31_6	Reserved
5 DisPrtPwrCutoff	OTG Disable Port Power Cut Off (DisPrtPwrCutoff) - 0: The core automatically turns off the VBUS by clearing the OCTL.PrtPwrCtl after A_WAIT_BCON Timeout whenever the port is disconnected in disconnected state. If Hibernation is enabled and when a hibernation request is received in disconnected state, the core switches off VBUS instantly by clearing the OCTL.PrtPwrCtl. - 1: The core maintains VBUS ON even after A_WAIT_BCON Timeout when port is in disconnected state. The core is in a A_WAIT_BCON state continuously waiting for a Connect. If hibernation is enabled and when hibernation

Table continues on the next page...

Field	Function
	request is received in disconnected state, the core maintains VBUS ON and continues to drive VBUS even in hibernation.
4 OTGHibDisMask	OTG Hibernation Disable Mask. - 0 - Any change in PHY input signals relevant to OTG (ID, Vbus-valid, B-Valid) are masked from generating a corresponding event when the core receives Hibernation Save request from Host or Device Driver. The change in signals is masked until the Host or Device Run/Stop is programmed to 1. - 1 - The PHY input signals are not masked even after the core receives the Hibernation Save request from Host or Device Driver. For normal functionality, this bit must be 1'b0.
3 OTGSftRstMsk	OTG Soft Reset Mask. This bit is used to mask specific soft resets from affecting the OTG functionality of the core. When set, the xHCI-based USBCMD.HCRST in host mode and DCTL.CSftRst in device mode is masked from affecting reset signal outputs sent to the PHY, the OTG FSM logic of the core and also the resets to the VBUS filters inside the core. - 1'b0: The xHCI-based USBCMD.HCRST and DCTL.CSftRst resets the OTG logic of the core. - 1'b1: The xHCI-based USBCMD.HCRST and DCTL.CSftRst is masked from the OTG logic of the core. This bit can be programmed to allow existing xHCI flows (with USBCMD.HCRST programming) to function in OTG scenarios without any software changes. This bit must be programmed only when GCTL.PrtCapDir = 2'b11. Otherwise it must be set at 1'b0. Note: When using the core for OTG2 or OTG3 applications, it is not recommended to program USBCMD.HCRST during role switch.
2 OTG_Version	This is a debug bit and it must always be set to 1'b0.
1 HNPCap	RSP/HNP Capability. The terminology RSP is used when the core is operating in SS mode, and HNP is used when the core is operating in non-SS mode. The application uses this bit to control the DWC_usb3 core's RSP/HNP capabilities. - 1'b0: RSP/HNP capability is not enabled. - 1'b1: RSP/HNP capability is enabled. Note: This bit is writable only if RSP/HNP mode is specified for Mode of Operation in coreConsultant, that is when DWC_USB3_EN_OTG != 0 and DWC_USB3_MODE = DRD. If RSP/HNP mode is not specified, this bit is Read Only, and is set to 1'b0.
0 SRPCap	SRP capability The application uses this bit to control the USB 3.0 core's SRP capabilities. If this bit is not set for B-device, it cannot request the connected A-device (host) to activate VBUS and start a session. If this bit is not set for A-device, it cannot detect the SRP from B-device (device) to activate VBUS and start a session. **NOTE:** This bit is writable only if OTG is enabled. If OTG is not enabled, then this bit is Read Only and is set to 1'b0.

11.1.3.1.101 OTG Control Register (OCTL)

11.1.3.1.101.1 Offset

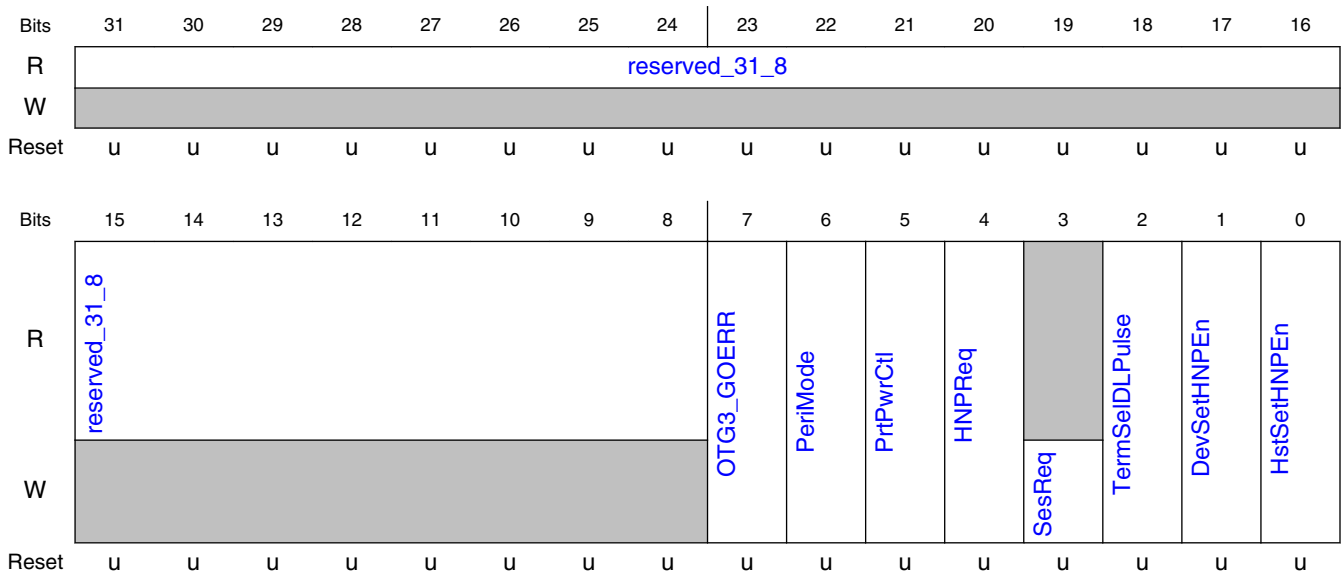
Register	Offset
OCTL	CC04h

11.1.3.1.101.2 Function

The OTG Control register controls the behavior of the OTG function of the core.

Reset Mask:0xFFFFFFA0

11.1.3.1.101.3 Diagram



11.1.3.1.101.4 Fields

Field	Function
31-8 reserved_31_8	Reserved
7 OTG3_GOERR	OTG3_GOERR When set, this instructs the core's LTSSM to go to the error state during OTG 3.0 RSP. The software must set this bit when TRSP_ACK_ERR, TRSP_CNF_ERR, or TRSP_WRST_ERR timeout occurs. This bit is self-cleared by the core when the LTSSM moves to the error state.
6 PeriMode	Peripheral Mode The application uses this bit to program the core to work as a peripheral or as a host. - 1'b0: The OTG device acts as a host - 1'b1: The OTG device acts as a peripheral
5 PrtPwrCtl	Port Power Control Application sets this bit to initiate Vbus drive when it is an A-device. The application must clear this bit only if it wants to switch off the Vbus to B-device. The core clears this bit in the following conditions: Transition from any state to A-IDLE state defined in OTG2.0 state machine: - When AIDL_BDIS_TOUT occurs in A_SUSPEND; - When A_WAIT_BCON_TOUT occurs in A_WAIT_BCON; - Transition to any B- state defined in OTG2.0 state machine;
4 HNPReq	HNP Request - 1'b0: No HNP request - 1'b1: HNP request The application sets this bit to initiate a HNP request to the connected USB host. The application clears this bit by writing a 1'b0 when either of the following is detected: - OEVT.OTGBDevBHostEndEvnt - OEVT.OTGBDevVBusChngEvnt
3 SesReq	Session Request - 1'b0: No session request - 1'b1: Session request The application sets this bit to initiate a session request on the USB. Writing 1'b1 to this field triggers the core to send SRP (data line pulsing) on the PHY interface. In the absence of OEVT.OTGBDevSessVldDetEvnt after a session request, the application must wait for atleast TB_SRP_FAIL time (6 secs) before initiating another session request. This field returns 1'b0 when read.
2 TermSelDLPulse	TermSel DLine Pulsing Selection This bit selects utmi_termselect to drive data line pulse during SRP. - 1'b0: Data line pulsing using utmi_txvalid (default). - 1'b1: Data line pulsing using utmi_termsel.

Table continues on the next page...

Field	Function
1 DevSetHNPEn	Device Set RSP/HNP Enable - 1'b0: RSP/HNP is not enabled in the application. - 1'b1: RSP/HNP is enabled in the application. The application sets this bit in the following scenario: - In HS/FS mode, when it successfully receives a SetFeature.SetHNPEnable command from the connected USB host. - In SS mode, when it has sent a b3_ntf_hst_rel to the A-device, or the A-device has sent an a3_ntf_host_req. Note: The terminology RSP is used when the core is operating in SS mode, and HNP is used when the core is operating in non-SS mode.
0 HstSetHNPEn	Host Set RSP/HNP Enable. - 1'b0: Host Set RSP/HNP is not enabled; - 1'b1: Host Set RSP/HNP is enabled; The application sets this bit in the following scenario: - In HS/FS mode, when it has successfully enabled HNP (using the SetFeature.SetHNPEnable command) from the connected device. - In SS mode, when it has successfully enabled the b3_rsp_enable feature in RSP-capable Device using SetFeature command while operating as an A-Host, or when it has received a b3_ntf_hst_rel through the SetFeature command while operating as A-peripheral. Note: The terminology RSP is used when the core is operating in SS mode, and HNP is used when the core is operating in non-SS mode.

11.1.3.1.102 OTG Events Register (OEVT)

11.1.3.1.102.1 Offset

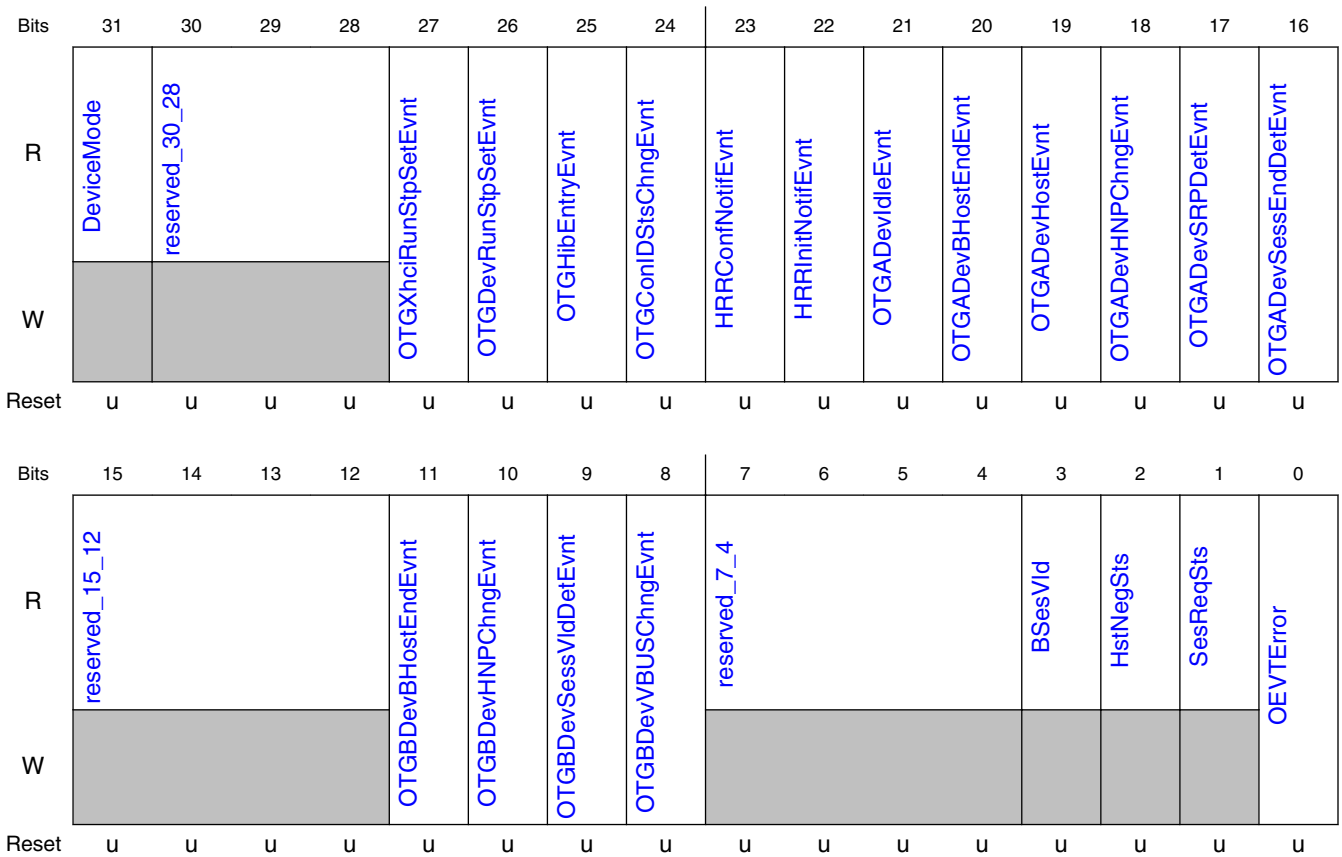
Register	Offset
OEVT	CC08h

11.1.3.1.102.2 Function

Any event set in this register will cause otg_interrupt signal to go high.

Reset Mask:0x7FDFFFF1

11.1.3.1.102.3 Diagram



11.1.3.1.102.4 Fields

Field	Function
31 DeviceMode	Device Mode Indicates whether the device is in A-device or B-device mode based on utmiotg_iddig - 1'b0: A-Device mode - 1'b1: B-Device mode The rest of the OTG Event Information bits (OTGxxxxEvtInfo) in OEVT register is based on the contents of this field.
30-28 reserved_30_28	Reserved
27 OTGXhciRunStpSetEvnt	OTG Host Run Stop Set Event This event is set when the Host Driver programs the USBCMD.Run/Stop to 1'b1. Note: - During Hibernation Exit, upon receiving this event, the OTG Driver can start register accesses to the core. - This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation.
26 OTGDevRunStpSetEvnt	OTG Device Run Stop Set Event This event is set when the Device Driver programs the DCTL.Run/Stop to 1'b1. Note: During Hibernation Exit, upon receiving this event, the OTG Driver can start register accesses to the core.
25 OTGHibEntryEvnt	OTG Hibernation Entry Event: - A-Device mode: This event is set when there is hibernation save request from the Host Driver by programming USBCMD.CSS bit to 1 when USBCMD.RunStp is 0. - B-Device mode: This event is set when there the Device Driver by programming DCTL.RunStp bit to 0. When this event is generated, the OTG Driver must disable all register accesses to the core.

Table continues on the next page...

Field	Function
24 OTGConIDStsChngEvnt	Connector ID Status Change Event Set in both A-Dev/B-Dev Mode. This event is generated when there is a change in connector ID status. When this bit is set, OEVT.OTGConIDStsChngEvnt is enabled. If not, the event is disabled.
23 HRRConfNotifEvnt	Host Role Request Confirm Notifier Event The core sets this bit after receiving a Host Role Request Device Notification TP with Confirm field set while operating as either an A-host or a B-host. Note: - This bit is applicable only when operating in SS mode. - This bit is applicable only for OTG 3.0 mode of operation.
22 HRRInitNotifEvnt	Host Role Request Initiate Notifier Event The core sets this bit after receiving a Host Role Request Device Notification TP with Initiate field set while operating as either an A-host or a B-host.
21 OTGADevIdleEvnt	A-device A-IDLE Event Set in A-device Mode Only. The event is generated when A-device enters A-IDLE state. This event is set when the OTG 2.0 FSM of the core enters A-IDLE state from any other OTG state. A-device A-IDLE Event When this bit is set, OEVT.OTGADevIdleEvnt is enabled. If not, the event is disabled.
20 OTGADevBHostEndEvnt	A-device B-Host End Event Set in A-device Mode Only. The event is generated when B-device has completed its host role. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. When this bit is set, OEVT.OTGADevBHostEndEvnt is enabled. If not, the event is disabled.
19 OTGADevHostEvnt	A-device host event Set in A-device Mode Only. This event is generated when A-device enters host role. In HS/FS mode, it occurs after the initial connect to a B-device as A-host as well as when there is a role change from A-peripheral to A-host. Note: This bit is applicable only for OTG 2.0 mode of operation.
18 OTGADevHNPChngEvnt	A-Dev HNP Change Event Set in A-device Mode Only. The event is generated when there is an HNP attempt. Note: This bit is applicable only for OTG 2.0 mode of operation. A-Dev HNP Change EventEn (OTGADevHNPChngEvntEn) When this bit is set, OEVT.OTGADevHNPChngEvnt is enabled. If not, the event is disabled
17 OTGADevSRPDetEvnt	SRP Detect Event Set in A-device Mode Only. This event is asserted when a session request from the B-device is detected via SRP. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. A-Dev HNP Change EventEn (OTGADevHNPChngEvntEn) When this bit is set, OEVT.OTGADevHNPChngEvnt is enabled. If not, the event is disabled.
16 OTGADevSessEndDetEvnt	Session End Detected Event Set in A-device Mode Only. This event is asserted when the utmiotg_vbusvalid signal goes low indicating the end of a session. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. When this bit is set, OEVT.OTGADevSessEndEvnt is enabled. If not, the event is disabled.
15-12 reserved_15_12	Reserved
11 OTGBDevBHostEndEvnt	B-Device B-Host End Event Set in B-device Mode Only. This event is generated when B-device has completed its host role. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. When this bit is set, OEVT.OTGBDevHostEndEvnt is enabled. If not, the event is disabled.
10 OTGBDevHNPChngEvnt	B-Dev HNP Change Event: Set in B-Device Mode only. This event is generated when there is a Success or Failure of an HNP attempt. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. When this bit is set, OEVT.OTGBDevHNPChngEvnt is enabled. If not, the event is disabled.
9 OTGBDevSessVldDetEvnt	Session Valid Detected Event Set in B-device Mode Only. This event is asserted when there is a valid Vbus from A-device and B-device succeeds in starting a session. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation.
8 OTGBDevVBUSChngEvnt	Vbus Change Event Set in B-device Mode Only. This event is asserted when the utmisrp_bvalid signal goes low (indicating the end of a session), or goes high. Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation. When this bit is set, OEVT.OTGBDevVBUSChngEvnt is enabled. If not, the event is disabled.

Table continues on the next page...

Field	Function
7-4 reserved_7_4	Reserved
3 BSesVld	Indicates the Device mode transceiver status. Indicates the Device mode transceiver status. The core updates this bit when: OEVTEN.OTGBDevVBUSChngEvnt is set. - 1'b0: B-session is not valid - 1'b1: B-session is valid Note: This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation.
2 HstNegSts	Host Negotiation Status The core updates this bit when any of the following bits is set: - OEVTEN.OTGADevHNPChngEvnt - OEVTEN.OTGBDevHNPChngEvnt This bit indicates Host Negotiation Success or Failure. 1'b0: Host negotiation failure. - In A-device, for HS/FS, this indicates an imminent end of session indication from the core. - In B-device, for HS/LS, it indicates that the timer used to wait for an A-device to signal a connection (b_ase0_brst_tmout in OTG 2.0) timed out resulting in B-device staying as B-peripheral. 1'b1: Host negotiation success. - This indicates that the host negotiation was successful. Note: This bit is applicable only for OTG 2.0 mode of operation.
1 SesReqSts	Session Request Status Ignore this field. It is used only for Synopsys internal testing.
0 OEVTError	OTG Event Error There are no errors currently defined.

11.1.3.1.103 OTG Events Enable Register (OEVTEN)

11.1.3.1.103.1 Offset

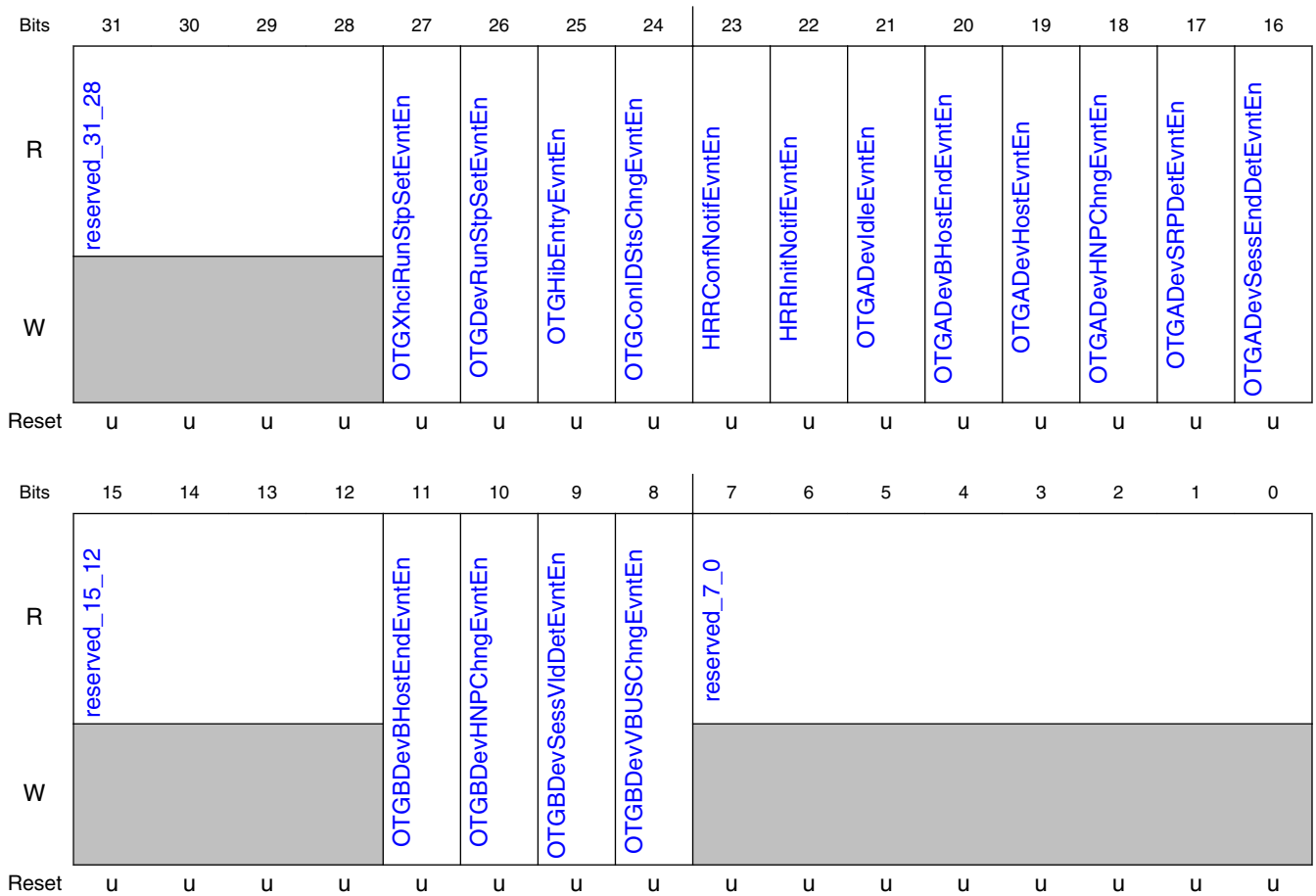
Register	Offset
OEVTEN	CC0Ch

11.1.3.1.103.2 Function

Setting a bit in this register enables the generation of corresponding events in OEVT and assertion of otg_interrupt due to this event.

Reset Mask:0xF000F0FF

11.1.3.1.103.3 Diagram



11.1.3.1.103.4 Fields

Field	Function
31-28 reserved_31_28	Reserved
27 OTGXhciRunStpSetEvntEn	OTG Host Run Stop Set Event Enable (OTGXhciRunStpSetEvntEn) When this bit is set, OEVT.OTGXhciRunStpSetEvnt is enabled. If not, the event is disabled.
26 OTGDevRunStpSetEvntEn	OTG Device Run Stop Set Event Enable (OTGDevRunStpSetEvntEn) When this bit is set, OEVT.DevRunStpSet event is enabled. If not, that event is disabled.
25 OTGHibEntryEvntEn	OTG Hibernation Entry Event Enable (OTGHibEntryEn) When this bit is set, OEVT.HibEntryEvnt is enabled. If not, the event is disabled.
24	OTGCommonEvtInfoEn[0] Connector ID Status Change Event (OTGConIDStsChngEvnt) Set in both A-Dev/B-Dev Mode: This event is generated when there is a change in connector ID status

Table continues on the next page...

Field	Function
OTGConIDStsChngEvntEn	
23 HRRConfNotifEvntEn	OTGCommonEvtInfoEn[2] HRRConfNotif Event Enable (HRRConfNotifEvntEn) When this bit is set, OEVT.HRRConfNotifEvnt is enabled. If not, the event is disabled.
22 HRRInitNotifEvntEn	OTGCommonEvtInfoEn[1] HRRInitNotif Event Enable (HRRInitNotifEvntEn) When this bit is set, OEVT.HRRInitNotifEvnt is enabled. If not, the event is disabled.
21 OTGADevIdleEvntEn	OTGADevEvtInfoEn[5] A-device A-IDLE Event (OTGADevIdleEvntEn) When this bit is set, OEVT.OTGADevIdleEvnt is enabled. If not, the event is disabled.
20 OTGADevBHostEndEvntEn	OTGADevEvtInfoEn[4] A-device B-Host End Event Enable (OTGADevBHostEndEvntEn) When this bit is set, OEVT.OTGADevBHostEndEvnt is enabled. If not, the event is disabled.
19 OTGADevHostEvntEn	OTGADevEvtInfoEn[3] A-device host event (OTGADevHostEvntEn) When this bit is set, OEVT.OTGADevHostEvnt is enabled. If not, the event is disabled.
18 OTGADevHNPChngEvntEn	OTGADevEvtInfoEn[2] A-Dev HNP Change EventEn (OTGADevHNPChngEvntEn) When this bit is set, OEVT.OTGADevHNPChngEvnt is enabled. If not, the event is disabled.
17 OTGADevSRPDetEvntEn	OTGADevEvtInfoEn[1] SRP Detect Event Enable (OTGADevSRPDetEvntEn) When this bit is set, OEVT.OTGADevSRPDetEvnt is enabled. If not, the event is disabled.
16 OTGADevSessEndDetEvntEn	OTGADevEvtInfoEn[0] Session End Detected Event Enable (OTGADevSessEndDetEvntEn) When this bit is set, OEVT.OTGADevSessEndEvnt is enabled. Otherwise, the event is disabled.
15-12 reserved_15_12	Reserved
11 OTGBDevBHostEndEvntEn	OTGBDevEvtInfoEn[3] B-device B-Host End Event Enable (OTGBDevBHostEndEvntEn) When this bit is set, OEVT.OTGBDevHostEndEvnt is enabled. If not, the event is disabled
10 OTGBDevHNPChngEvntEn	OTGBDevEvtInfoEn[2] B-Dev HNP Change Event Enable (OTGBDevHNPChngEvntEn) When this bit is set, OEVT.OTGBDevHNPChngEvnt is enabled. If not, the event is disabled
9 OTGBDevSessVldDetEvntEn	OTGBDevEvtInfoEn[1] Session Valid Detected Event Enable (OTGBDevSessVldDetEvntEn) Set in B-device mode only. This event is asserted when there is a valid VBUS from A-device and B-device succeeds in starting a session.
8 OTGBDevVBUSChngEvntEn	OTGBDevEvtInfoEn[0] VBUS Change Event Enable (OTGBDevVBUSChngEvntEn) When this bit is set, OEVT.OTGBDevVBUSChngEvnt is enabled. If not, the event is disabled.
7-0 reserved_7_0	Reserved

11.1.3.1.104 OTG Status Register (OSTS)

11.1.3.1.104.1 Offset

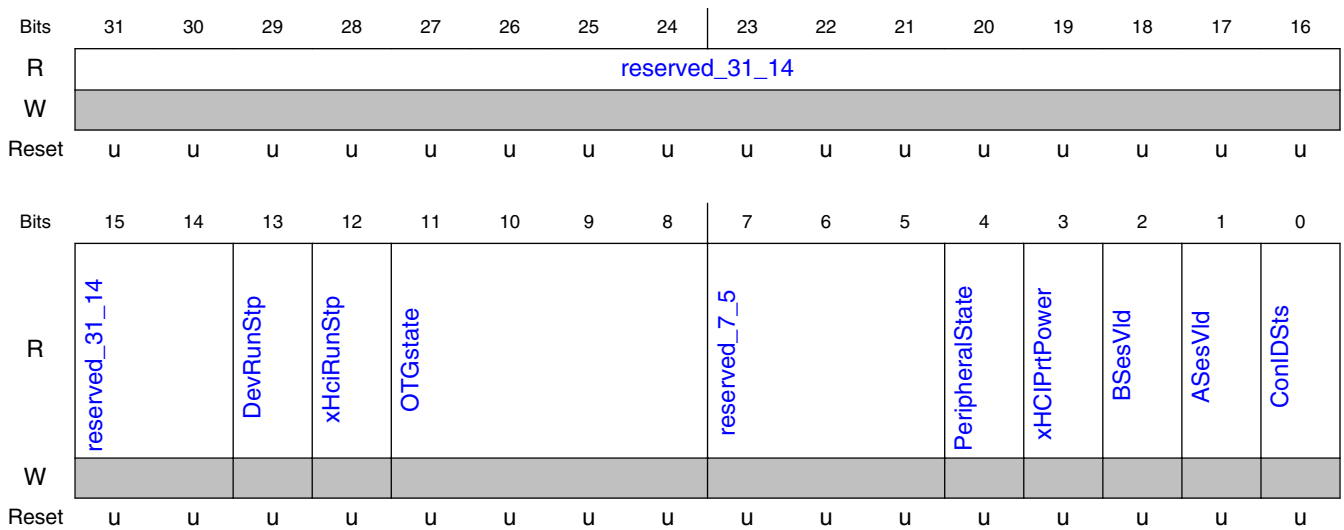
Register	Offset
OSTS	CC10h

11.1.3.1.104.2 Function

The OTG Status Register reflects the status of the OTG function of the core.

Reset Mask:0xFFFFC0E0

11.1.3.1.104.3 Diagram



11.1.3.1.104.4 Fields

Field	Function
31-14 reserved_31_14	Reserved
13 DevRunStp	This bit reflects the status of the Run/Stop bit in the DCTL Device register. - 0 DCTL.Run/Stop is set to 0. - 1 DCTL.Run/Stop is set to 1.
12 xHciRunStp	OTG Host Run Stop Set Event This event is set when the Host Driver programs the USBCMD.Run/Stop to 1'b1. Note: - During Hibernation Exit, upon receiving this event, the OTG Driver can start register accesses to the core. - This bit is applicable for OTG 2.0 and OTG 3.0 modes of operation.
11-8	This is a debug field indicating the current state of the OTG state machine. The encoding is as follows: - 4'b0000: A_IDLE - 4'b0010: A_WAIT_BCON - 4'b0011: A_WAIT_VFALL - 4'b0100: A_VBUS_ERR -

Table continues on the next page...

Field	Function
OTGstate	4'b0101: A_HOST - 4'b0110: A_SUSPEND - 4'b1111: A_PERIPHERAL - 4'b0111: A_WAIT_PPWR - 4'b1000: B_IDLE - 4'b1001: B_SRP_INIT - 4'b1010: B_PERIPHERAL - 4'b1011: B_WAIT_ACON - 4'b1100: B_HOST - 4'b1101: A_WAIT_SWITCH - 4'b1110: B_WAIT_SWITCH
7-5 reserved_7_5	Reserved
4 PeripheralState	OTG state Indicates whether the core is acting as a peripheral or host. - 1'b0: Host - 1'b1: Peripheral
3 xHCIPrtPower	This bit reflects the PORTSC.PP bit in the xHCI register.
2 BSesVld	B-Session Valid Indicates the Device mode transceiver status. In OTG mode, applications use this bit to determine if the device is connected. - 1'b0: B-session is not valid. - 1'b1: B-session is valid.
1 ASesVld	Indicates the Host mode transceiver status. - 1'b0: A-session is not valid - 1'b1: A-session is valid
0 ConIDSts	Connector ID Status: - Indicates the connector ID status - 1'b0: The DWC_usb3 core is in A-device mode - 1'b1: The DWC_usb3 core is in B-device mode Note: The reset value of this field depends on the power-on value of the IDDIG signal from the PHY.

11.1.3.1.105 ADP Configuration Register (ADPCFG)

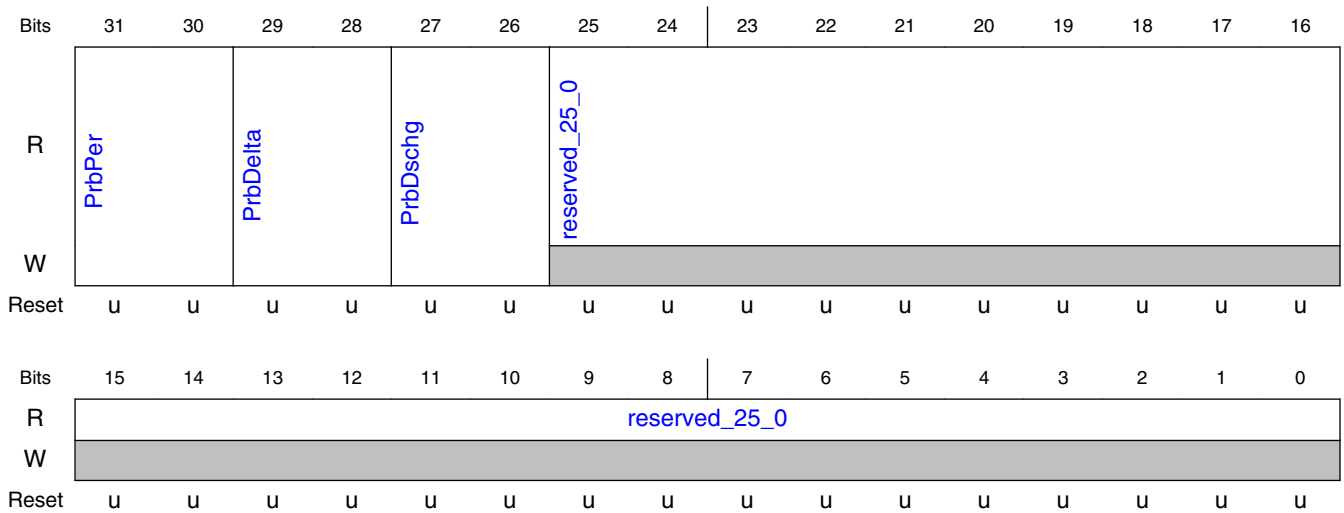
11.1.3.1.105.1 Offset

Register	Offset
ADPCFG	CC20h

11.1.3.1.105.2 Function

Reset Mask:0x3FFFFF

11.1.3.1.105.3 Diagram



11.1.3.1.105.4 Fields

Field	Function
31-30 PrbPer	Probe Period These bits set the T_ADP_PRB as follows: - 2'b00: 0.775 sec - 2'b01: 1.55 sec - 2'b10: 2.275 sec - 2'b11: Reserved The scaledown values for PrbPer are: - 2'b00: 12.5ms - 2'b10: 25 ms - 2'b11: 31.25 ms
29-28 PrbDelta	Probe Delta These bits set the resolution for RTIM value. They are defined in units of 32 kHz clock cycles as follows: - 2'b00: 1 cycles - 2'b01: 2 cycles - 2'b10: 3 cycles - 2'b11: 4 cycles For example, if this value is chosen to be 2'b01, it means that RTIM increments for every two 32 kHz clock cycles.
27-26 PrbDschg	Probe Discharge These bits set the time for TADP_DSCHG. They are defined as follows: - 2'b00: 4 msec - 2'b01: 8 msec - 2'b10: 16 msec - 2'b11: 32 msec The scaledown values for the PrbDschg are as follows: - 2'b00: 62.5 us - 2'b01: 125 us - 2'b10: 250 us - 2'b11: 500 us
25-0 reserved_25_0	Reserved

11.1.3.1.106 ADP Control Register (ADPCTL)

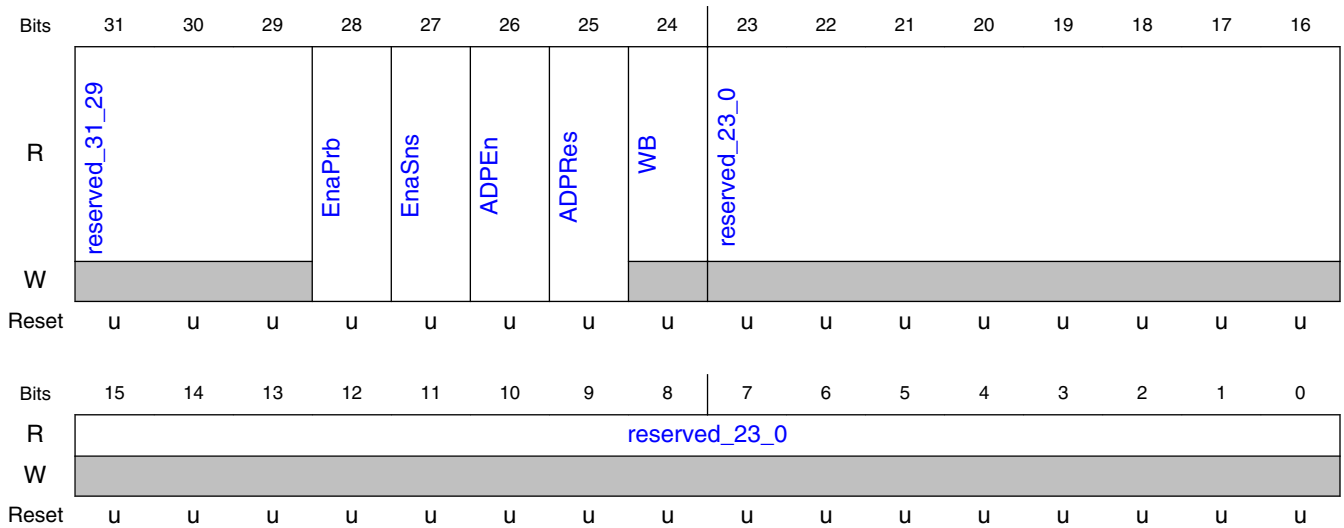
11.1.3.1.106.1 Offset

Register	Offset
ADPCTL	CC24h

11.1.3.1.106.2 Function

Reset Mask:0xE2FFFFFF

11.1.3.1.106.3 Diagram



11.1.3.1.106.4 Fields

Field	Function
31-29 reserved_31_29	Reserved
28 EnaPrb	Enable Probe: When set to 1'b1 along with ADPEn, the core performs a probe operation.
27 EnaSns	Enable Sense When set to 1'b1 along with ADPEn, the core performs a sense operation.
26 ADPEn	ADP Enable When set to 1'b1, the core performs either ADP probing or sensing based on EnaPrb and EnaSns. - ADPEn = 1'b0 gates the suspend clock for major portion of ADP related logic.
25 ADPRes	ADP Reset When set to 1'b1, the ADP controller is reset. This bit is auto-cleared after the reset procedure is complete in the ADP controller.
24 WB	Write Busy - 1'b0: Write Completed - 1'b1: Write in Progress The application can read or write ADPCFG and ADPCTL registers only if this field is cleared. The hardware sets this bit when the write is in progress in the Suspend clock domain.
23-0 reserved_23_0	Reserved

11.1.3.1.107 ADP Event Register (ADPEVT)

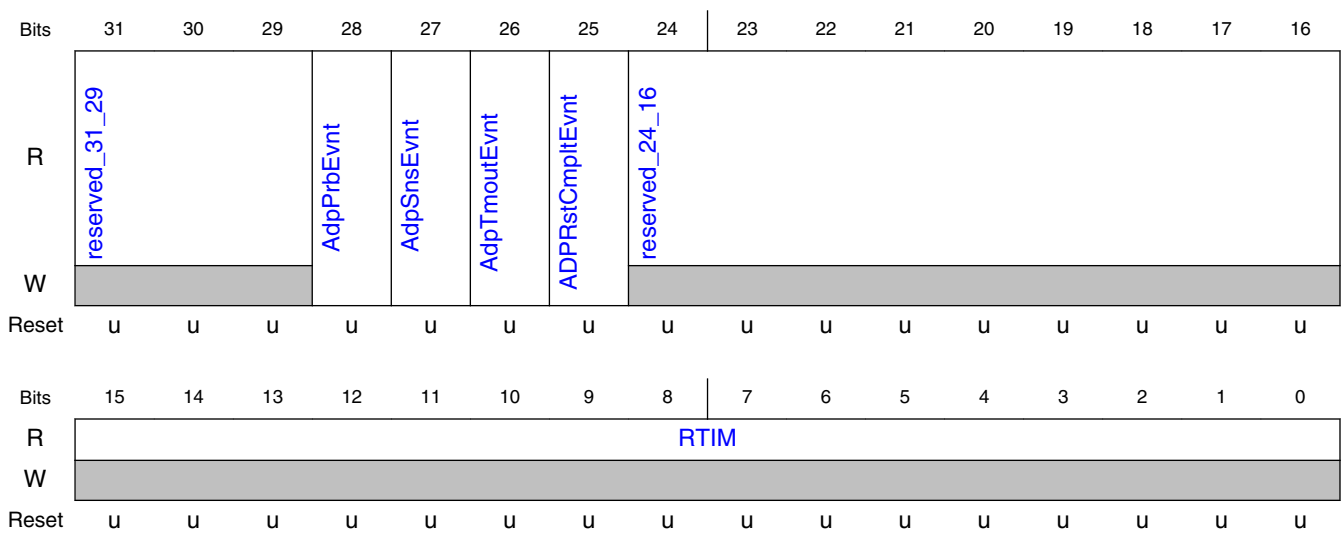
11.1.3.1.107.1 Offset

Register	Offset
ADPEVT	CC28h

11.1.3.1.107.2 Function

Reset Mask:0xFFFF0000

11.1.3.1.107.3 Diagram



11.1.3.1.107.4 Fields

Field	Function
31-29 reserved_31_29	Reserved
28 AdpPrbEvnt	ADP Probe Event When this event is set, it means that the Vbus voltage is greater than VadpPrb or VadpPrb is reached.
27 AdpSnsEvnt	ADP Sense Event When this event is set, it means that the Vbus voltage is greater than VadpSns or VadpSns is reached.
26 AdpTmoutEvnt	ADP Timeout Event This event is relevant when ADP probe command is executed. When this event is set, it means that the ramp time is completed (GADPCTL.RTIM has reached its terminal value of 0x7FF). This is a debug feature that allows software to read the ramp time after each cycle.
25 ADPRstCmpltEvnt	This event, when set, indicates that the ADP Reset command is successful.

Table continues on the next page...

Field	Function
24-16 reserved_24_16	Reserved
15-0 RTIM	RAMP TIME: These bits capture the latest time it took for Vbus to ramp from VADP_SINK to VADP_PRB. The bits are defined in units of 32 kHz clock cycles as follows: - 0x000: 1 cycles - 0x001: 2 cycles - 0x002: 3 cycles and so on till, - 0xFFFF: 65536 cycles The maximum time of 65536 cycles corresponds to a time of 2.04 seconds. Note: For scaledown ramp_timeout: - PrbDelta = 2'b11 => 781.25 us - PrbDelta = 2'b10 => 1562.5 us - PrbDelta = 2'b01 => 3125 us - PrbDelta = 2'b00 => 6250 us

11.1.3.1.108 (ADPEVTEN)

11.1.3.1.108.1 Offset

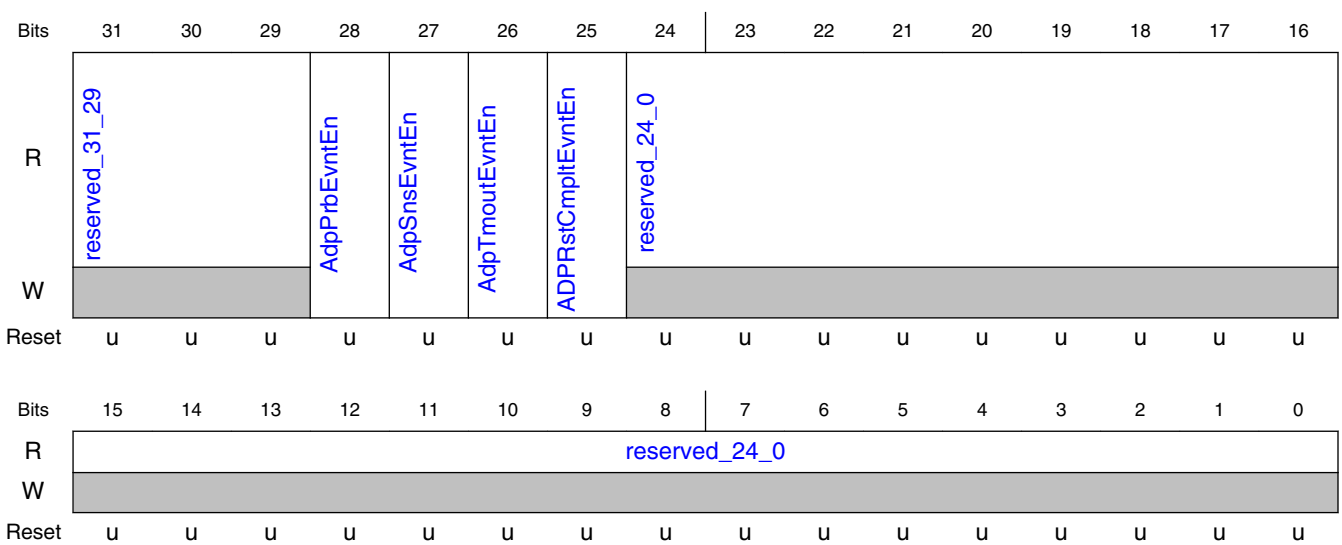
Register	Offset
ADPEVTEN	CC2Ch

11.1.3.1.108.2 Function

ADP Event Enable Register

Reset Mask:0xE1FFFFFF

11.1.3.1.108.3 Diagram



11.1.3.1.108.4 Fields

Field	Function
31-29 reserved_31_29	Reserved
28 AdpPrbEvntEn	ADP Probe Event Enable When this bit is set, ADPPrbEvnt in ADPEVT register is enabled.
27 AdpSnsEvntEn	ADP Sense Event Enable When this bit is set, AdpSnsEvntEn in ADPEVT register is enabled.
26 AdpTmoutEvntEn	ADP Timeout Event Enable When this bit is set, AdpTmoutEvntEn in ADPEVT register is enabled.
25 ADPRstCmpltEvntEn	ADP Reset complete Event Enable When this bit is set, ADPRstCmpltEvnt in ADPEVT register is enabled.
24-0 reserved_24_0	Reserved

11.1.3.1.109 BCFG (BCFG)

11.1.3.1.109.1 Offset

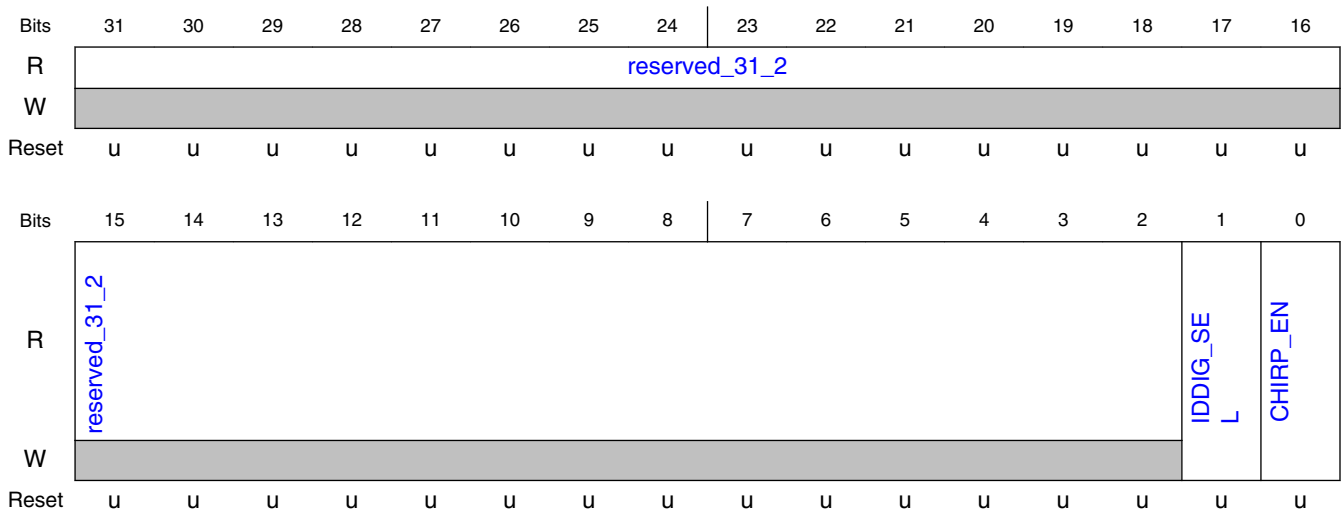
Register	Offset
BCFG	CC30h

11.1.3.1.109.2 Function

BC Configuration Register

Reset Mask:0xFFFFFFFC

11.1.3.1.109.3 Diagram



11.1.3.1.109.4 Fields

Field	Function
31-2 reserved_31_2	Reserved
1 IDDIG_SEL	IDDIG Select: When this bit is programmed as: - 1'b1: The core determines the ID pin value from RID_A and RID_GND inputs of the Battery Charger. - 1'b0: The core determines ID pin value from utmiotg_iddig input. When ULPI PHY is used, IDDIG_SEL needs to be programmed to 1'b0 always, so that the core uses the ID value obtained from the RX_CMD.
0 CHIRP_EN	Chirp Enable When this bit is 1'b1, the core asserts the bc_chirp_on output signal to indicate an imminent chirp.

11.1.3.1.110 BCEVT (BCEVT)

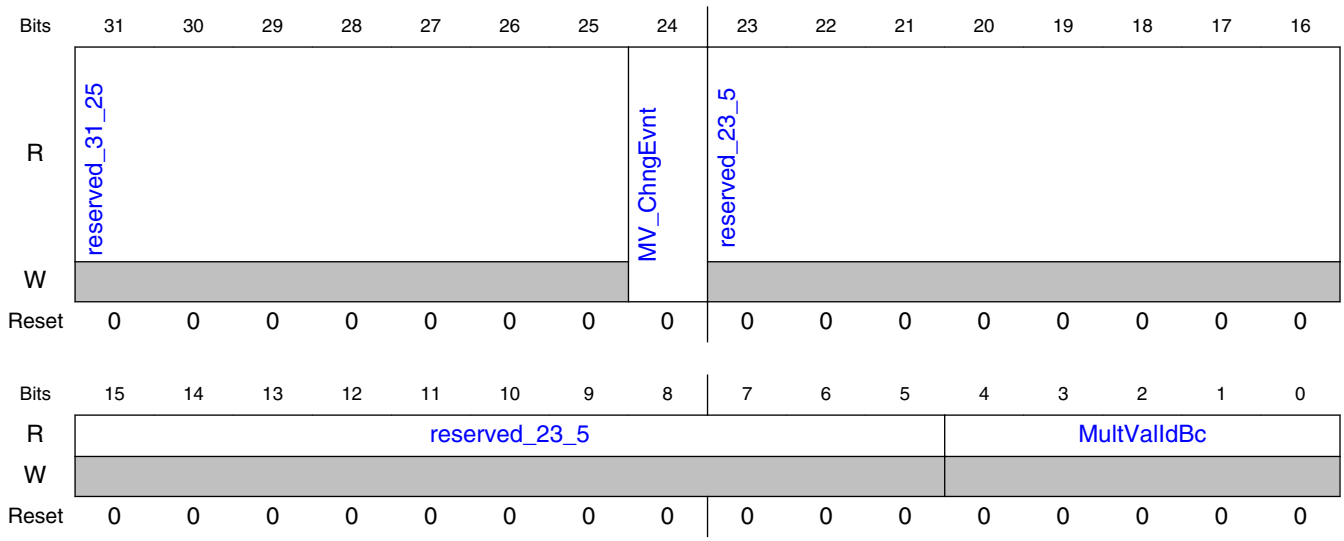
11.1.3.1.110.1 Offset

Register	Offset
BCEVT	CC38h

11.1.3.1.110.2 Function

BC Event Register Writing 1 to the info bit in this register clears the register bit and the associated interrupt.

11.1.3.1.110.3 Diagram



11.1.3.1.110.4 Fields

Field	Function
31-25 reserved_31_25	Reserved
24 MV_ChngEvnt	Multi-Valued input changed Event: This event indicates that there is a change in the value of at least one ACA pin value. This bit is present only if DWC_USB3_EN_BC = 1, otherwise it is Reserved.
23-5 reserved_23_5	Reserved
4-0 MultValIdBc	Multi Valued ID pin Indicates the Battery Charger ACA inputs. - MultValIdBc[4]: bc_rid_float - MultValIdBc[3]: bc_rid_gnd - MultValIdBc[2]: bc_rid_a - MultValIdBc[1]: bc_rid_b - MultValIdBc[0]: bc_rid_c

11.1.3.1.111 BCEVTEN (BCEVTEN)

11.1.3.1.111.1 Offset

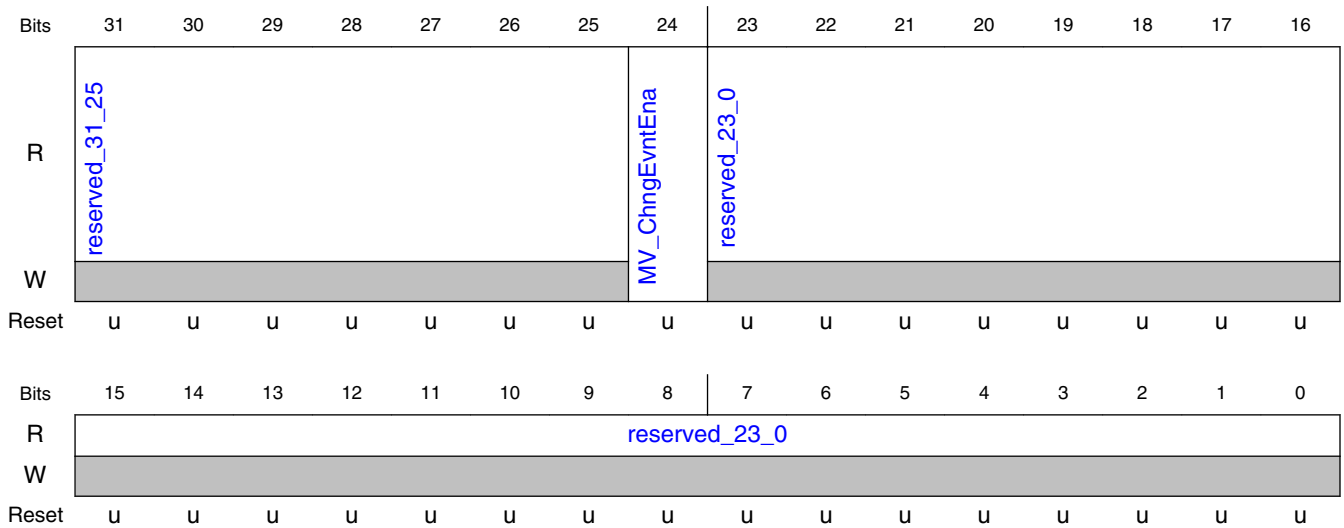
Register	Offset
BCEVTEN	CC3Ch

11.1.3.1.111.2 Function

BC Event Enable Register

Reset Mask:0xFEFFFFFF

11.1.3.1.111.3 Diagram



11.1.3.1.111.4 Fields

Field	Function
31-25 reserved_31_25	Reserved
24 MV_ChngEvntEna	BCEvtInfoEna[0] Multi-Valued input changed Event Enable (MV_ChngEvntEn) When this bit is set, MV_ChngEvnt in BCEVT register is enabled.
23-0 reserved_23_0	Reserved

11.1.3.2 USB3_GLUE register descriptions

Between USB3.0 Core and USB PHY, there is a GLUE architecture to make the connection more configurable.

11.1.3.2.1 GLUE_usb3 memory map

Base Address for GLUE registers - USB1: 381F_0000, and USB2: 382F_0000

USB GLUE architecture makes the connection between Core and PHY more configurable. USB GLUE contains some configurable registers and status signal monitoring to Core/PHY. There are two GLUE blocks and each has its own configurable registers.

Offset	Register	Width (In bits)	Access	Reset value
0h	USB_CTL0_ADDR (USB_CTL0_ADDR)	32	RW	0020_F003h
20h	USB_STS0_ADDR (USB_STS0_ADDR)	32	RO	0000_07E8h
40h	PHY_CTL0_ADDR (PHY_CTL0_ADDR)	32	RW	3200_0CF0h
44h	PHY_CTL1_ADDR (PHY_CTL1_ADDR)	32	RW	0018_0006h
48h	PHY_CTL2_ADDR (PHY_CTL2_ADDR)	32	RW	0000_0100h
4Ch	PHY_CTL3_ADDR (PHY_CTL3_ADDR)	32	RW	94D4_E464h
50h	PHY_CTL4_ADDR (PHY_CTL4_ADDR)	32	RW	046C_7D09h
54h	PHY_CTL5_ADDR (PHY_CTL5_ADDR)	32	RW	0000_007Fh
80h	PHY_STS0_ADDR (PHY_STS0_ADDR)	32	W1C	0000_0000h

11.1.3.2.2 USB_CTL0_ADDR (USB_CTL0_ADDR)

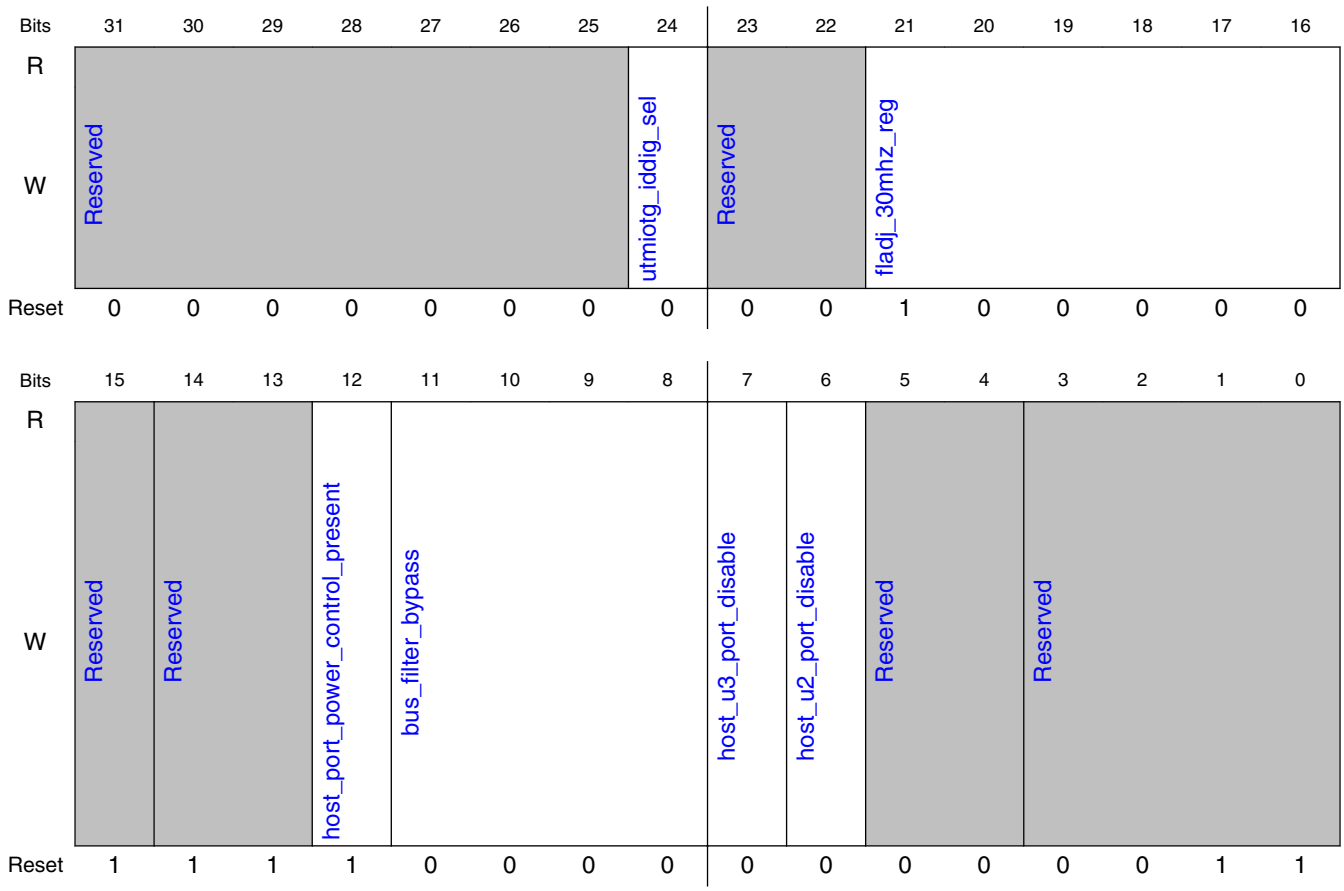
11.1.3.2.2.1 Offset

Register	Offset
USB_CTL0_ADDR	0h

11.1.3.2.2.2 Function

USB Core Configurable bits Register

11.1.3.2.2.3 Diagram



11.1.3.2.2.4 Fields

Field	Function
31-25 —	Reserved
24 utmiotg_iddig_sel	USB3.0 Core signal 0b - USB PHY ID0 1b - GPIO PAD
23-22 —	Reserved
21-16 fladj_30mhz_reg	USB3.0 Core signal
15 —	Reserved
14-13 —	Reserved

Table continues on the next page...

Field	Function
12 host_port_power_control_present	USB3.0 Core signal
11-8 bus_filter_bypass	USB3.0 Core signal
7 host_u3_port_disable	USB3.0 Core signal
6 host_u2_port_disable	USB3.0 Core signal
5-4 —	Reserved
3-0 —	Reserved

11.1.3.2.3 USB_STS0_ADDR (USB_STS0_ADDR)

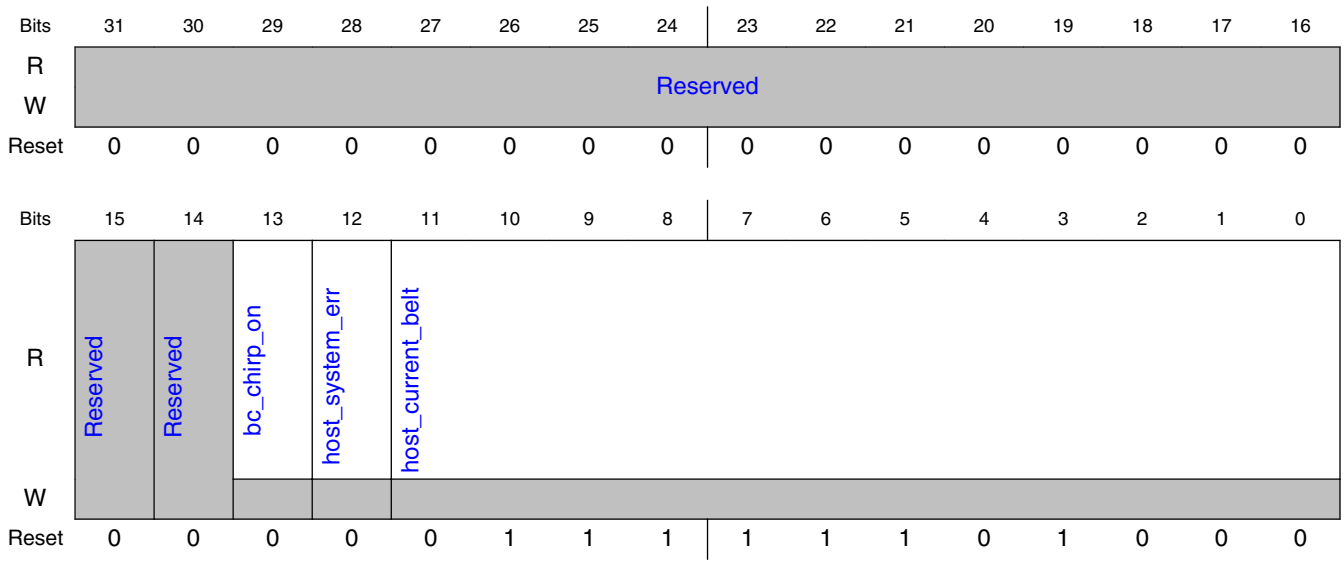
11.1.3.2.3.1 Offset

Register	Offset
USB_STS0_ADDR	20h

11.1.3.2.3.2 Function

USB3.0 Core Status bits Register

11.1.3.2.3.3 Diagram



11.1.3.2.3.4 Fields

Field	Function
31-15 —	Reserved
14 —	Reserved
13 bc_chirp_on	USB3.0 Core Signal
12 host_system_err	USB3.0 Core Signal
11-0 host_current_belt	USB3.0 Core Signal

11.1.3.2.4 PHY_CTL0_ADDR (PHY_CTL0_ADDR)

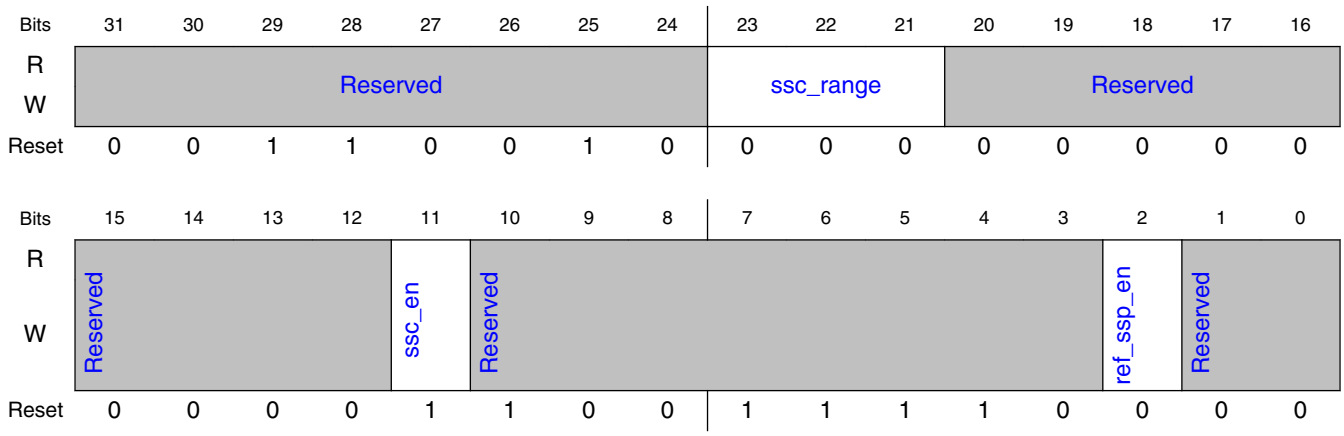
11.1.3.2.4.1 Offset

Register	Offset
PHY_CTL0_ADDR	40h

11.1.3.2.4.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.4.3 Diagram



11.1.3.2.4.4 Fields

Field	Function
31-24 —	Reserved
23-21 ssc_range	USB3.0 PHY Signal
20-12 —	Reserved
11 ssc_en	USB3.0 PHY Signal
10-3 —	Reserved
2 ref_ssp_en	USB3.0 PHY Signal
1-0 —	Reserved

11.1.3.2.5 PHY_CTL1_ADDR (PHY_CTL1_ADDR)

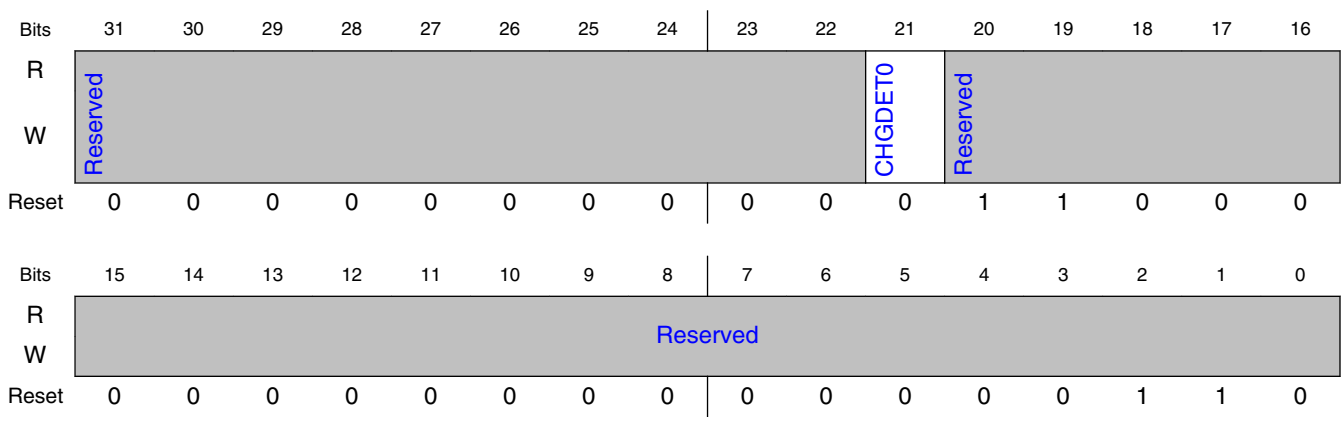
11.1.3.2.5.1 Offset

Register	Offset
PHY_CTL1_ADDR	44h

11.1.3.2.5.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.5.3 Diagram



11.1.3.2.5.4 Fields

Field	Function
31-22 —	Reserved
21 CHGDET0	USB3.0 PHY Signal 0b - CHGDET0 will not generate USB interrupt 1b - CHGDET0 will generate USB interrupt when CHGDET0=1
20-0 —	Reserved

11.1.3.2.6 PHY_CTL2_ADDR (PHY_CTL2_ADDR)

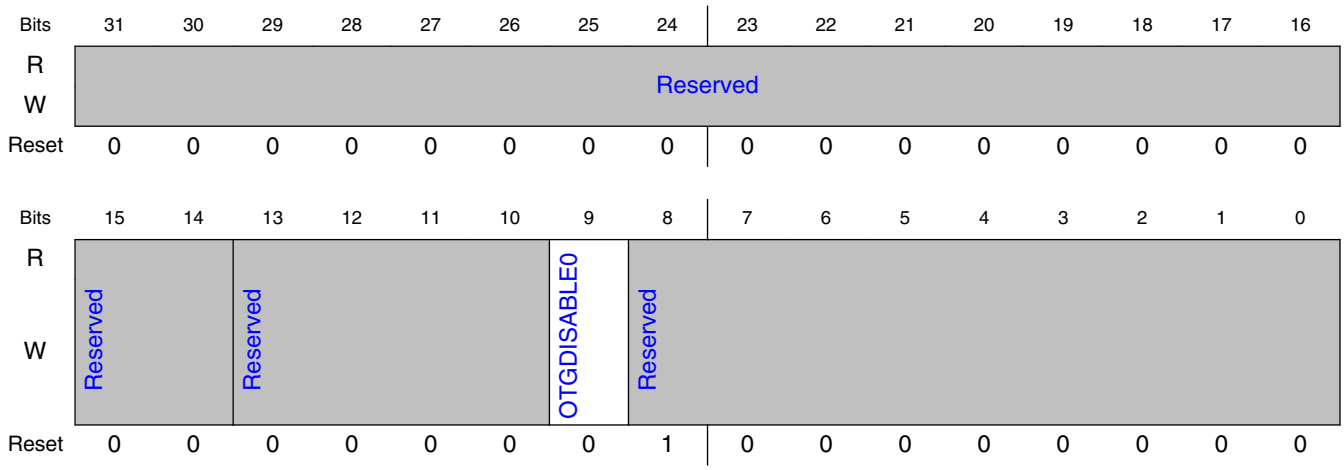
11.1.3.2.6.1 Offset

Register	Offset
PHY_CTL2_ADDR	48h

11.1.3.2.6.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.6.3 Diagram



11.1.3.2.6.4 Fields

Field	Function
31-14 —	Reserved
13-10 —	Reserved
9 OTGDISABLE0	USB3.0 PHY Signal
8-0 —	Reserved

11.1.3.2.7 PHY_CTL3_ADDR (PHY_CTL3_ADDR)

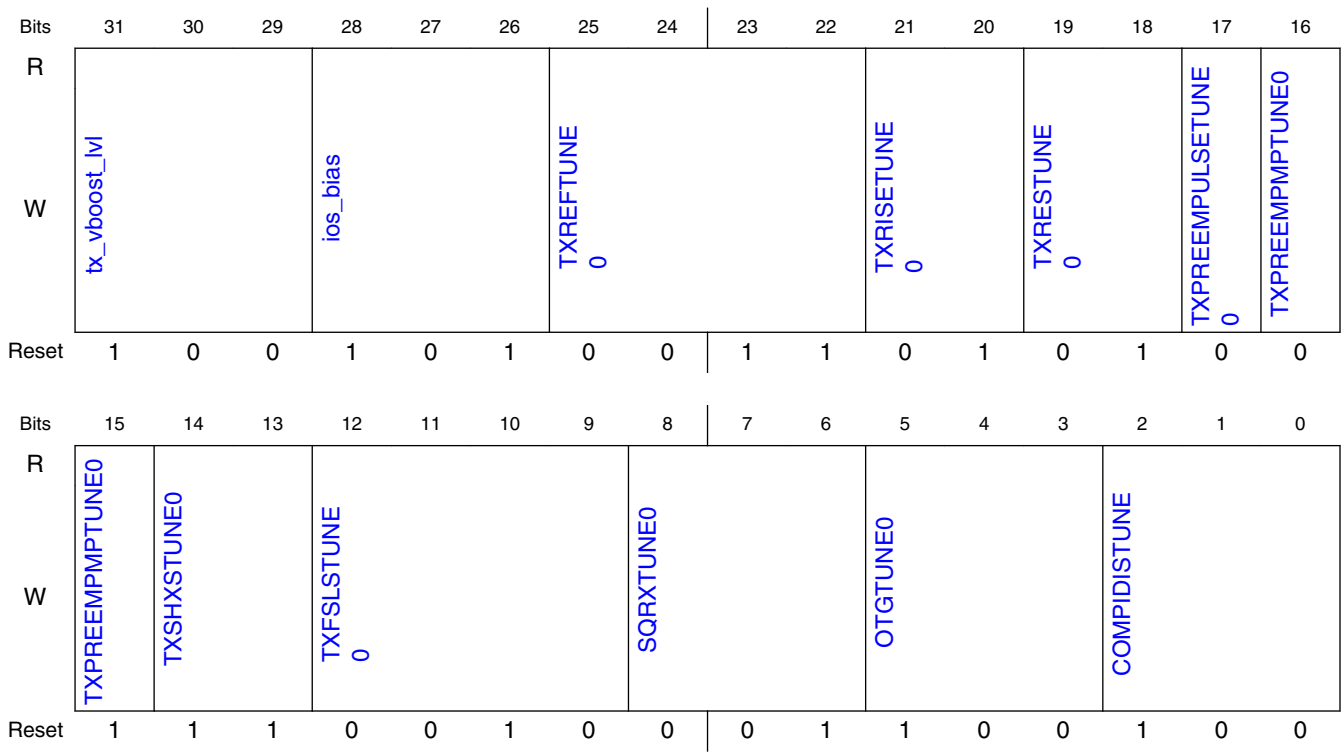
11.1.3.2.7.1 Offset

Register	Offset
PHY_CTL3_ADDR	4Ch

11.1.3.2.7.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.7.3 Diagram



11.1.3.2.7.4 Fields

Field	Function
31-29 tx_vboost_lvl	USB3.0 PHY Signal
28-26 ios_bias	USB3.0 PHY Signal
25-22 TXREFTUNE0	USB3.0 PHY Signal
21-20 TXRISETUNE0	HS Transmitter Rise/Fall Time Adjustment Adjusts the rise/fall times of the high-speed waveform. To enable tuning at the board level, connect this bit to a register. **NOTE:** A positive binary bit setting change results in a –4% incremental change in the HS rise/fall time. A negative binary bit setting change results in a +4% incremental change in the HS rise/fall time.
19-18	USB Source Impedance Adjustment

Table continues on the next page...

Field	Function
TXRESTUNE0	Some applications require additional devices to be added on the USB, such as a series switch, which can add significant series resistance. This bus adjusts the driver source impedance to compensate for added series resistance on the USB. **NOTE:** Any setting other than the default can result in source impedance variation across process, voltage, and temperature conditions that does not meet USB 2.0 specification limits. If this bus is not used, leave it at the default setting 00b - Source impedance is increased by approximately 1.5 Ω 01b - Design default 10b - Source impedance is decreased by approximately 2 Ω 11b - Source impedance is decreased by approximately 4 Ω
17 TXPREEMPULSETUNE0	HS Transmitter Pre-Emphasis Duration Control This signal controls the duration for which the HS pre-emphasis current is sourced onto DP or DM. The HS Transmitter pre-emphasis duration is defined in terms of unit amounts. One unit of pre-emphasis duration is approximately 580 ps and is defined as 1X pre-emphasis duration. This signal is valid only if either TXPREEMPMPTUNE[1] or TXPREEMPMPTUNE[0] is set to 1'b1. 0b - 2X (design default), long pre-emphasis current duration If TXPREEMPULSETUNE is not used, set it to 1'b0 1b - 1X, short pre-emphasis current duration
16-15 TXPREEMPMPTUNE0	USB3.0 PHY Signal
14-13 TXSHXSTUNE0	USB3.0 PHY Signal
12-9 TXFSLSTUNE0	USB3.0 PHY Signal
8-6 SQRXTUNE0	USB3.0 PHY Signal
5-3 OTGTUNE0	USB3.0 PHY Signal
2-0 COMPIDISTUNE	USB3.0 PHY Signal

11.1.3.2.8 PHY_CTL4_ADDR (PHY_CTL4_ADDR)

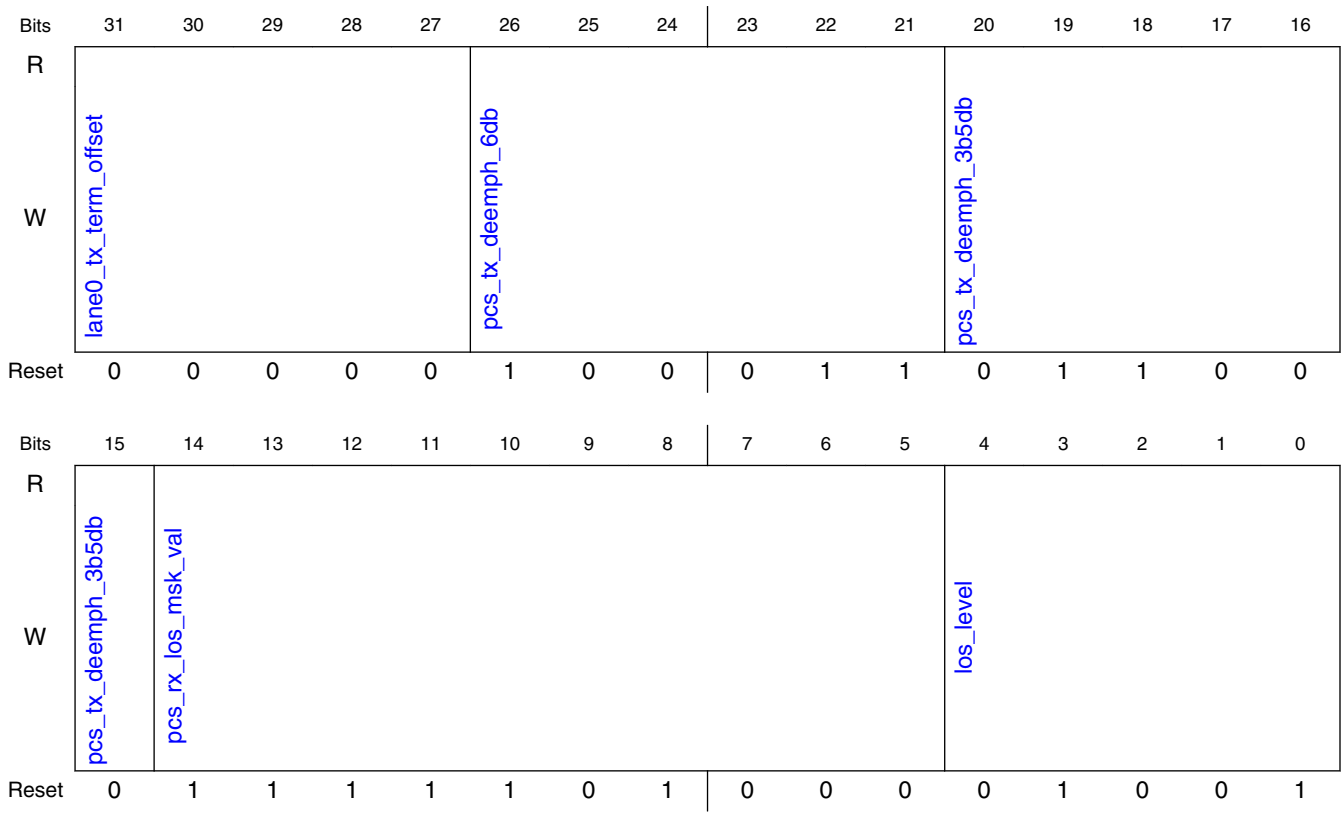
11.1.3.2.8.1 Offset

Register	Offset
PHY_CTL4_ADDR	50h

11.1.3.2.8.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.8.3 Diagram



11.1.3.2.8.4 Fields

Field	Function
31-27 lane0_tx_term_offset	USB3.0 PHY Signal
26-21 pcs_tx_deemph_6db	USB3.0 PHY Signal
20-15 pcs_tx_deemph_3b5db	USB3.0 PHY Signal
14-5 pcs_rx_los_msk_val	USB3.0 PHY Signal
4-0 los_level	USB3.0 PHY Signal

11.1.3.2.9 PHY_CTL5_ADDR (PHY_CTL5_ADDR)

11.1.3.2.9.1 Offset

Register	Offset
PHY_CTL5_ADDR	54h

11.1.3.2.9.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								pcs_tx_swing_full							
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

11.1.3.2.9.4 Fields

Field	Function
31-7 —	Reserved
6-0 pcs_tx_swing_full	USB3.0 PHY Signal

11.1.3.2.10 PHY_STS0_ADDR (PHY_STS0_ADDR)

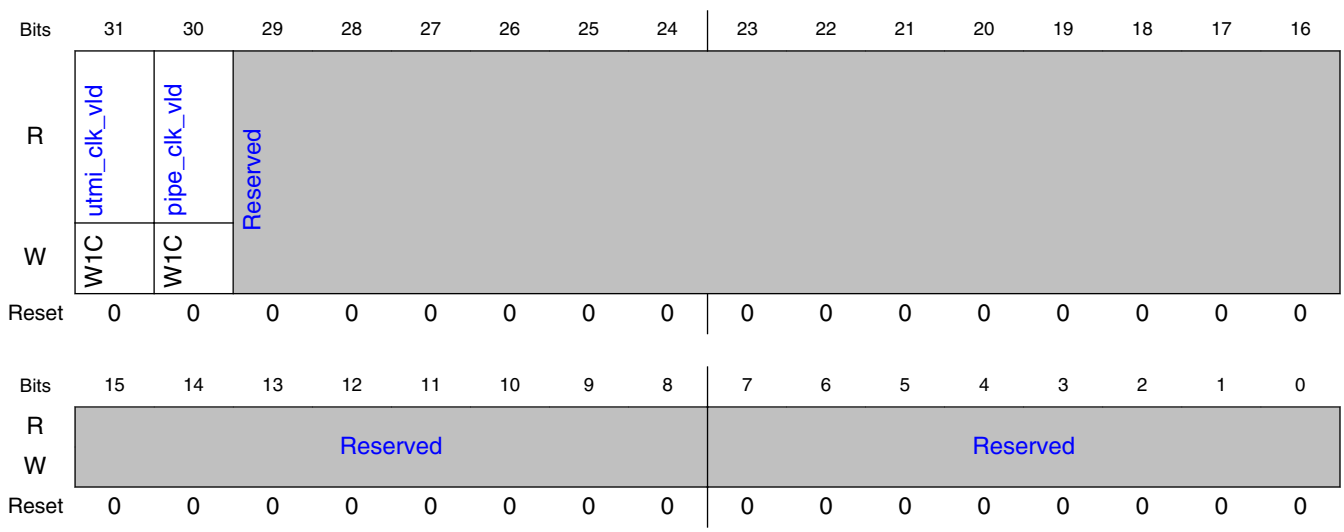
11.1.3.2.10.1 Offset

Register	Offset
PHY_STS0_ADDR	80h

11.1.3.2.10.2 Function

USB3.0 PHY Status bits Register

11.1.3.2.10.3 Diagram



11.1.3.2.10.4 Fields

Field	Function
31 utmi_clk_vld	USB3.0 PHY Signal synchronised by USB bus clock.After PHY and core reset pipe clock is stable if this bit is set.
30 pipe_clk_vld	USB3.0 PHY Signal synchronised by USB bus clock.After PHY and core reset pipe clock is stable if this bit is set.
29-8 —	Reserved
7-0 —	Reserved

11.2 Universal Serial Bus PHY (USB_PHY)

11.2.1 Overview

The USB 3.0 PHY is designed for USB 3.0 SuperSpeed applications connecting to a USB OTG controller. The USB 3.0 PHY supports the USB 3.0 SuperSpeed (5 Gbps) protocol and data rate and is backward compatible with USB 2.0 high-speed (480 Mbps), full-speed (12 Mbps), and low-speed (1.5 Mbps) protocols and data rates. This block guide provides an introduction to the USB 3.0 PHY and its features.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

11.2.1.1 Block Diagram

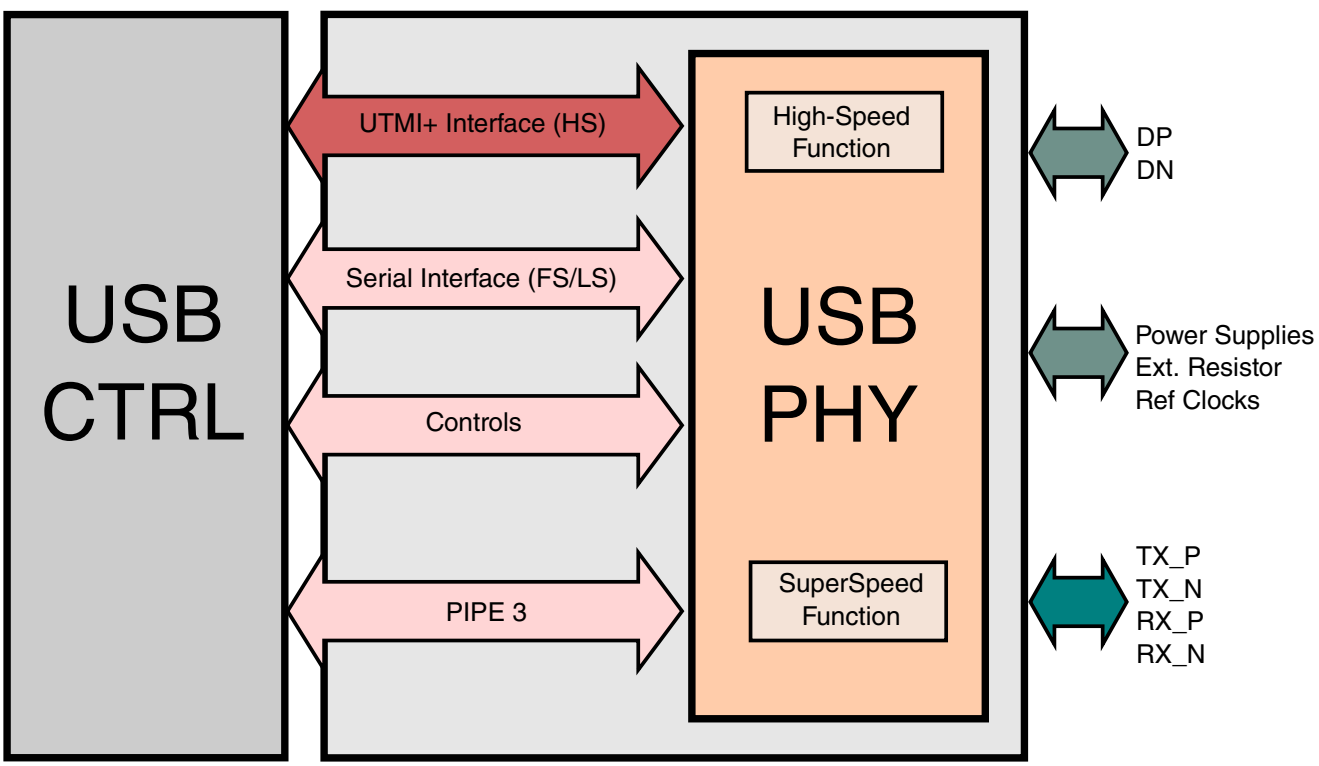


Figure 11-47. USB PHY Block Diagram

11.2.1.2 Features

The USB 3.0 PHY provides the following general features:

- Built-in VBUS threshold comparators;
- Single reference clock input for both USB 3.0 and USB 2.0 operating modes
- Integrated 3.3-V-to-1.8-V regulator in the PHY

The USB 3.0 PHY provides the following USB 3.0 features:

- 5-Gbps SuperSpeed data transmission rate through 3-m USB 3.0 cable
- PIPE 3-compliant SuperSpeed USB 3.0 Transceiver interface
- Integrated PHY includes transmitter, receiver, SSC generation, PLL, digital core, and ESD
- Adaptive Rx equalization
- Low-jitter PLL technology with excellent supply isolation

The USB 3.0 PHY provides the following USB 2.0 features:

- Built-in VBUS threshold comparators;
- 480-Mbps high-speed, 12-Mbps full-speed, and 1.5-Mbps low-speed serial (Host mode only) data transmission rates
- Supports high-speed power modes: suspend, resume, and remote wakeup
- 45-Ω termination, 1.5-kΩ pull-up and 15-kΩ pull-down resistors with support for independence control of the pull-down resistors
- 8/16-bit unidirectional parallel interfaces for HS, FS, and LS (Host mode only) modes of operation in accordance with the UTMI+ specification
- Dual (HS/FS) mode host/device support
- Data recovery from serial data on the USB connector
- SYNC/End-of-Packet (EOP) generation and checking
- Bit stuffing and unstuffing, and bit-stuffing error detection
- Non Return to Zero Invert (NRZI) encoding and decoding
- Bit serialization and deserialization
- VBUS threshold comparators

11.2.1.3 Standards Compliance

The USB 3.0 PHY is fully compliant with all the required features of the following standards:

- Universal Serial Bus 3.0 Specification, Revision 1.0
  - Supports 5.0-Gbps SuperSpeed (SS) through 3-m cable, 480-Mbps high-speed (HS), 12-Mbps fullspeed (FS), and 1.5-Mbps low-speed (LS) serial (Host mode only) data transmission rates, with a high-speed (HS) transmit-to-receive interpacket delay requirement of 38 bit-times as implied by the UTMI specification, version 1.05.

    - Supports beaconing, receiver detection, and electrical idle
    - Four-wire SuperSpeed connection (two Tx and two Rx unidirectional) with built-in Tx and Rx equalization
- PIPE 3-compliant USB 3.0 SuperSpeed interface
- On-The-Go Supplement to the USB 2.0 Specification, Revision 2.0
- UTMI+ Specification, Revision 1.0 (Level 3)
- Battery Charging Specification, Revision 1.2

11.2.2 Functional Description

11.2.2.1 PIPE Interface

The datapath of the USB 3.0 PHY is through the standard PIPE 3 interface. This interface can connect directly to any USB 3.0 controller. For more information about the PIPE 3 interface signals, refer to PHY Interface For the PCI Express and USB 3.0 Architectures.

11.2.2.2 Termination Tuning

The PHY uses an external resistor to calibrate the termination impedances of the PHY's high-speed inputs and outputs. The resistor is shared between the USB 2.0 high-speed outputs and the SuperSpeed I/O. Each time the PHY is taken out of a reset, a termination calibration is performed. The termination calibration is continuous for USB 2.0 operation except when the USB 2.0 port is suspended. The termination calibration is performed only when coming out of a reset in SuperSpeed operation.

A resistor calibration on the SuperSpeed link cannot be performed while the link is operational. At no point is a request for calibrating the SuperSpeed link using rtune_req required; this feature is provided only for completeness.

11.2.2.3 Battery Charging

The USB 3.0 PHY supports battery charger detection to enable devices to draw increased current from VBUS for charging and/or powering up from dedicated charging ports or charging downstream ports.

11.2.2.4 Dead Battery Provision

The dead battery provision enables a device to draw increased current (up to 100 mA) from VBUS in certain conditions. This function can be used only if all power supplies are stable and at their respective, normal, valid operating levels.

11.2.2.5 Data Contact Detect

Prior to a device performing battery charger detection, DCD is performed to determine if the D+ and D– pins in the device have made contact with those in the host or charger.

11.2.2.6 UART / Autoresume

The USB 3.0 PHY's single-ended receivers are used for UART and Autoresume data reception, while the full-speed drivers are used for data transmission. UART and Autoresume signaling levels are the same as signaling levels for USB 2.0 FS mode.

In this mode of operation, the driver source impedance is not calibrated; however, the driver retains the settings of the last calibration. In addition, the rise/fall time control is disabled, so changes in the state of DP and DN driven by this mode are not expected to meet the rise and fall times as defined in the USB 2.0 specification.

11.2.2.7 Serial Interface

The serial interface supports full-speed and low-speed data transmission rates to and from a USB 1.1 controller.

11.2.3 Clocks

The USB 3.0 PHY supports a wide range of input reference clocks from both external and on-chip clock sources. To support both SuperSpeed and high-speed operations, one of the following must be provided:

- A compatible, shared reference clock frequency
- Separate clock sources to support SuperSpeed operation and high-speed operation

11.2.4 External Signals

This section describes the USB 3.0 PHY signals that go to pins:

- External reference resistor: resref

- Receive pins: RX_N, RX_P
- Transmit pins: TX_N, TX_P
- USB 2.0 D+/D– lines: DN, DP
- USB mini-receptacle identifier: ID
- USB 5-V power supply pin: VBUS

11.3 PCI Express (PCIe)

11.3.1 Overview

This PCI Express dual mode (DM) controller provides a solution to implement a PCI Express port for a PCI Express root complex or endpoint application. A complete PCI Express port solution includes the controller, an analog PHY macro, and application logic to source and sink data.

NOTE

The information within this block guide is Synopsys Proprietary. Used with permission.

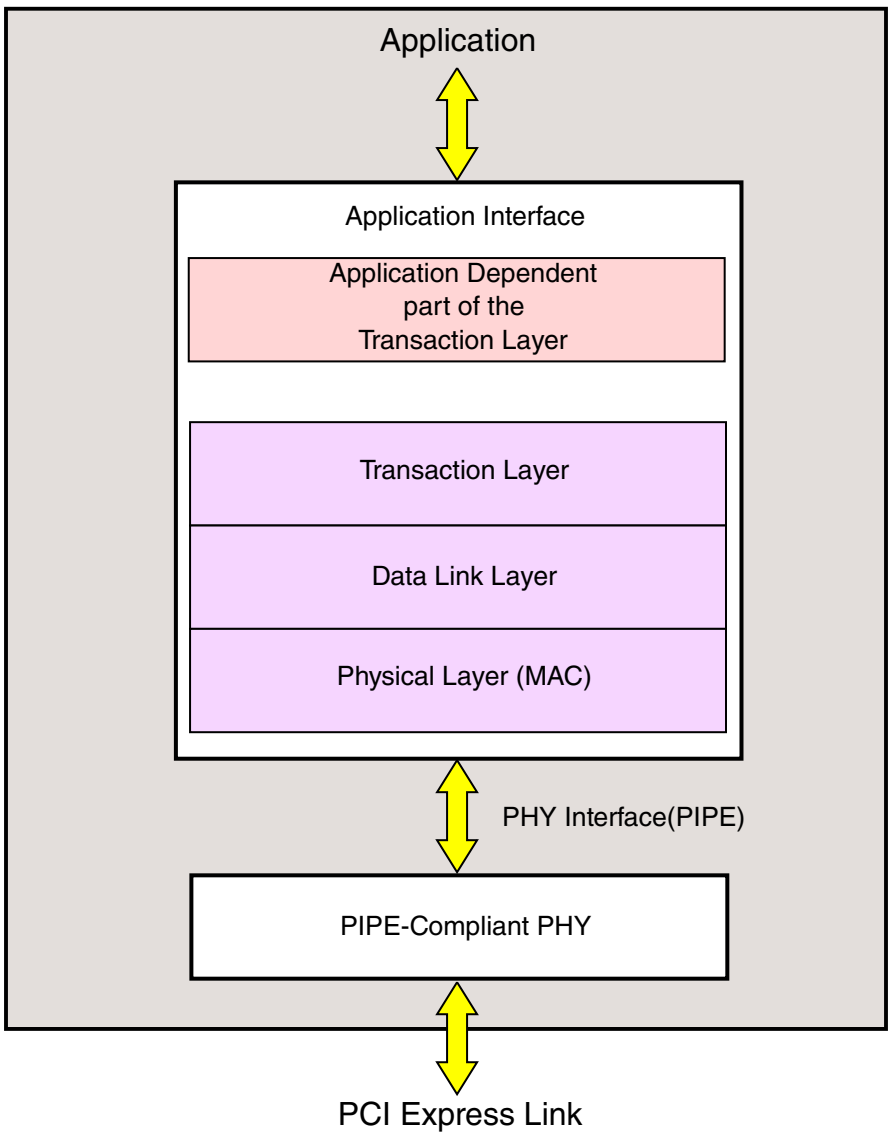


Figure 11-48. PCIe Application

11.3.1.1 Features

The PCI Express DM controller supports:

- Latency Tolerance Reporting (LTR)
- Address Translation Services (ATS) by your application
- Page Request Services (PRS) by your application
- Completion Timeout Ranges
- ID-Based Ordering (IDO) by your application
- M-PCIe
- PASID

- L1 Substates (L1SS)
- Extended Tag Support
- Resizable BAR (RBAR) with Expanded RBAR support,
- Separate Refclk with Independent Spread Spectrum Clocking (SRIS)
- Lightweight Notifications (LN)
- Readiness Notifications (RN)
- PCI Express Active State Power Management (ASPM)
- PCI Express Advanced Error Reporting (AER) with Multiple Header Logging
- PCI Express Advanced Error Reporting (AER)
- Vital Product Data (VPD)
- x1 lane
- Power Gating (UPF) Support
- Advanced Power and Clock Management
- Internal Address Translation Unit
- Internal MSI-X Generation Module
- AXI3
- Embedded DMA
- Upconfigure Support
- ECRC Generation and Checking
- Supports two Virtual Channels (VC)
- Max_Payload_Size (128 bytes)
- Configurable Filtering Rules for Posted, Non-posted, and Completion Traffic
- Configurable BAR Filtering, I/O Filtering, Configuration Filtering and Completion Lookup/Timeout
- MSI and MSI-X with Per-Vector Masking (PVM), Extended message data for MSI
- Type 1 Configuration Space
- Application-initiated Manual Lane Reversal/flip for situations where controller does not detect Lane 0
- Type 0 Configuration space

11.3.1.2 Standards

The PCIe controller implements the following standards:

- *PCI Express Base Specification, Revision 4.0, Version 0.7*
- Access to this specification requires membership in PCI-SIG.
- PCI Local Bus Specification, Revision 3.0
- Access to this specification requires membership in PCI-SIG.
- PCI Bus Power Management Specification, Revision 1.2 Access to this specification requires membership in PCI-SIG.
- PCI Express Card Electromechanical Specification, Revision 1.1 Access to this specification requires membership in PCI-SIG.

11.3.2 Functional Description

11.3.2.1 Link Establishment

The controller implements the LTSSM function according to the *PCI Express Base Specification, Revision 4.0, Version 0.7. .*

11.3.2.1.1 Transmit TLP Processing

The following section describe the flow of transmit TLPs through the controller.

NOTE

The controller does not check the TLP for errors.

The PCIe controller ensures transmitted TLP Payload sizes are less than or equal to the maximum payload size limit.

11.3.2.1.1.1 Transmit TLP Arbitration

TLPs and DLLPs have equal priorities during transmit arbitration as shown in the figure below. The priority of TLP types and DLLP types are shown in tables below.

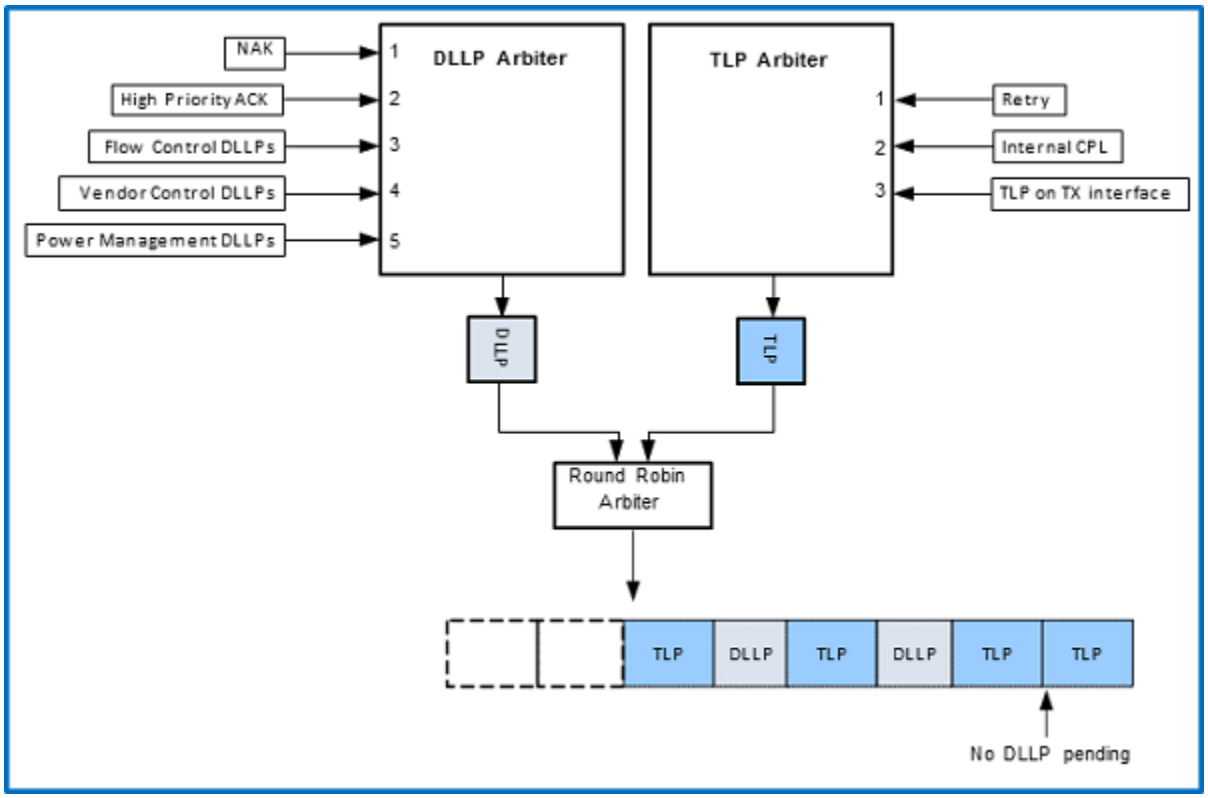


Figure 11-49. Transmit TLP/DLLP Arbitration

Table 11-25. TLP Arbitration Priority (1= Highest, 4= Lowest)

Priority	TLP Type
1	Completion of any TLP currently in progress
2	Retry buffer retransmissions
3	TLPs from the transaction layer (in the following order of priority): • Messages generated by the controller (including by your application through the MSI interface) • Upstream ports: Completions generated by the controller (including memory or I/O mapped application register space) for Type 0 configuration read and write requests, or responses to error conditions (unsupported requests) • Downstream ports: Completions generated by the controller for unsupported requests or completer aborts
4	TLPs on the AXI Master/Slave interface

Table 11-26. DLLP Arbitration Priority (1= Highest, 6 = Lowest)

Priority	DLLP Type
1	Completion of any DLLP currently in progress
2	NAK DLLP

Table continues on the next page...

Table 11-26. DLLP Arbitration Priority (1= Highest, 6 = Lowest) (continued)

3	High Priority ACK DLLP
4	Flow Control DLLP
5	Vendor Specific Message DLLP
6	Power management or any other Low Priority DLLP

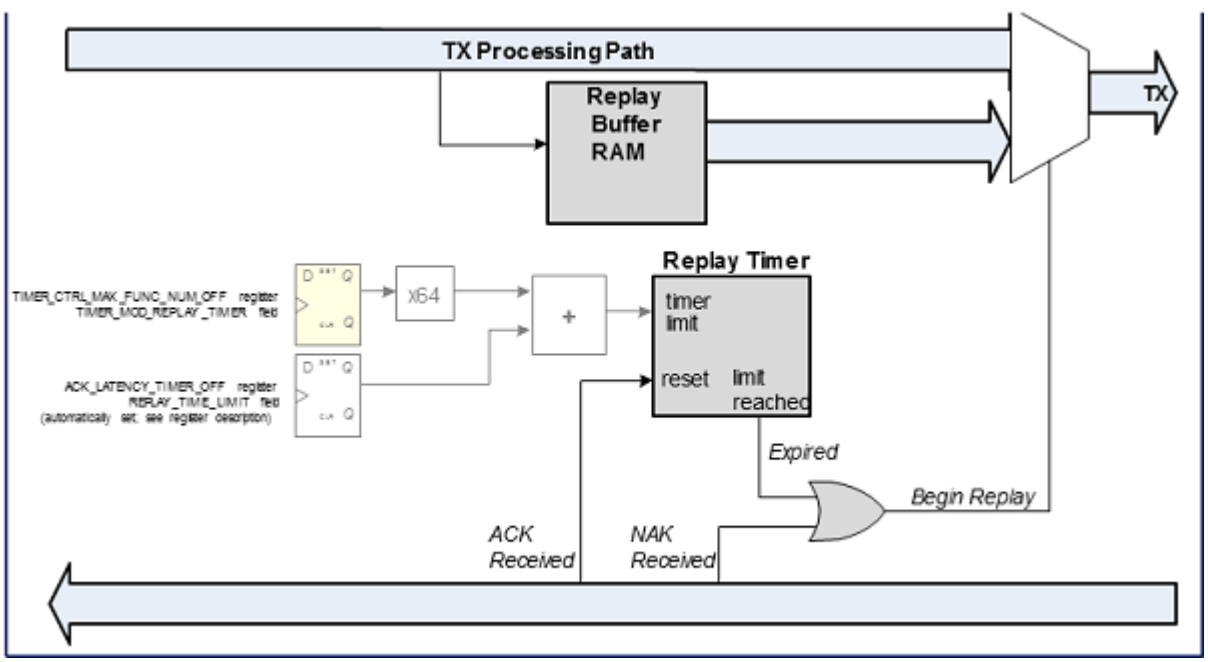
11.3.2.1.1.1.1 Effects of Flow Control Credits On Transmit Client Arbitration

The controller checks that enough flow control (FC) credits are available in the remote device for the specific type of transaction (posted, non-posted, completion) before allowing a transmission of a TLP. TLPs that passed the credit check are arbitrated according to the supported arbitration method. Internally generated completions and messages are also gated by the arbitration logic, though at highest priority, and must also pass the FC credit test before they are accepted for transmission.

11.3.2.1.1.2 Transmit Replay

You can modify the time-out value of the replay buffer as shown in the figure below. For a typical system, you do not have to modify the default settings, unless the remote device is executing unexpected replays.

Figure 11-50. Replay Buffer And Time (Representative)



11.3.2.1.2 Receive TLP Processing

This section describes the flow of received TLPs through the controller.

11.3.2.1.2.1 Receive Filtering

The controller contains a filter module that is responsible for the following tasks:

- Determine the status of a received TLP using filtering rules.
- Determine the destination of a received TLP based on the filtering status.
- Indicate the status of the received TLP using outputs.
- Report Errors to AER registers based on filter results. When more than one type of error is detected, Error Pollution, of the *PCI Express Base Specification, Revision 4.0, Version 0.7* is followed.
- The controller filters and routes received TLPs according to a set of rules determined by the TLP type based on the *PCI Express Base Specification, Revision 4.0, Version 0.7* and user-configurable filtering options.

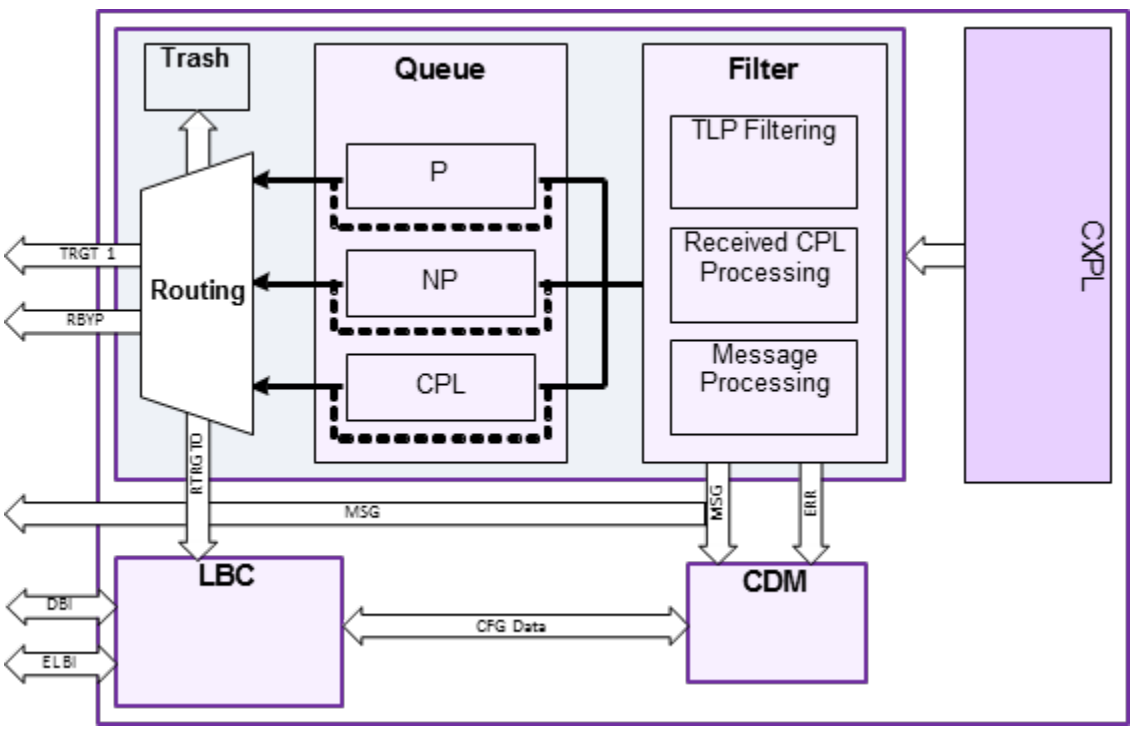


Figure 11-51. RADM Block Diagram

The following general rules apply to all incoming TLPs that are not malformed. For details on what happens to malformed TLPs, see "Error Detection for Received TLPs" By default:

- For a function in device power states D1, D2, and D3hot, the controller only accepts CFG and MSG requests TLPs for that function. All other incoming request types for that function are treated as unsupported requests (UR).
- When the controller detects an error in a received TLP, it normally performs the following:
  - Discards the TLP
  - Generates a completion (for non-posted requests) with the completion status set to CA or UR
  - Sets the status in the PCI-compatible Status register
  - Sets the status in the AER registers
  - Generates an error message (upstream port only)

All error-free MSG requests are decoded internally, signaled on the SII interface and then dropped. When you want to have the decoded message also sent to the application interface, then see “Routing of Received Messages”.

11.3.2.1.2.2 Receive Routing

This section discusses how the RADM routes different TLP types to different receive interfaces depending on TLP type and filter result. TRGT1 refers to the AXI interface. TRGT0 refers to the internal interface used to access the CDM registers or the ELBI in an upstream port.

11.3.2.1.2.2.1 EP Mode Routing Overview

The possible destinations of a posted or non-posted request TLP are TRGT1 interface, TRGT0 interface and Discard. By default:

- CFG requests are routed to TRGT0 and then to CDM through the LBC.
- BAR-matched MEM and I/O requests are routed to TRGT1.
- MSG requests are decoded internally, signaled on the SII interface, and then terminated.

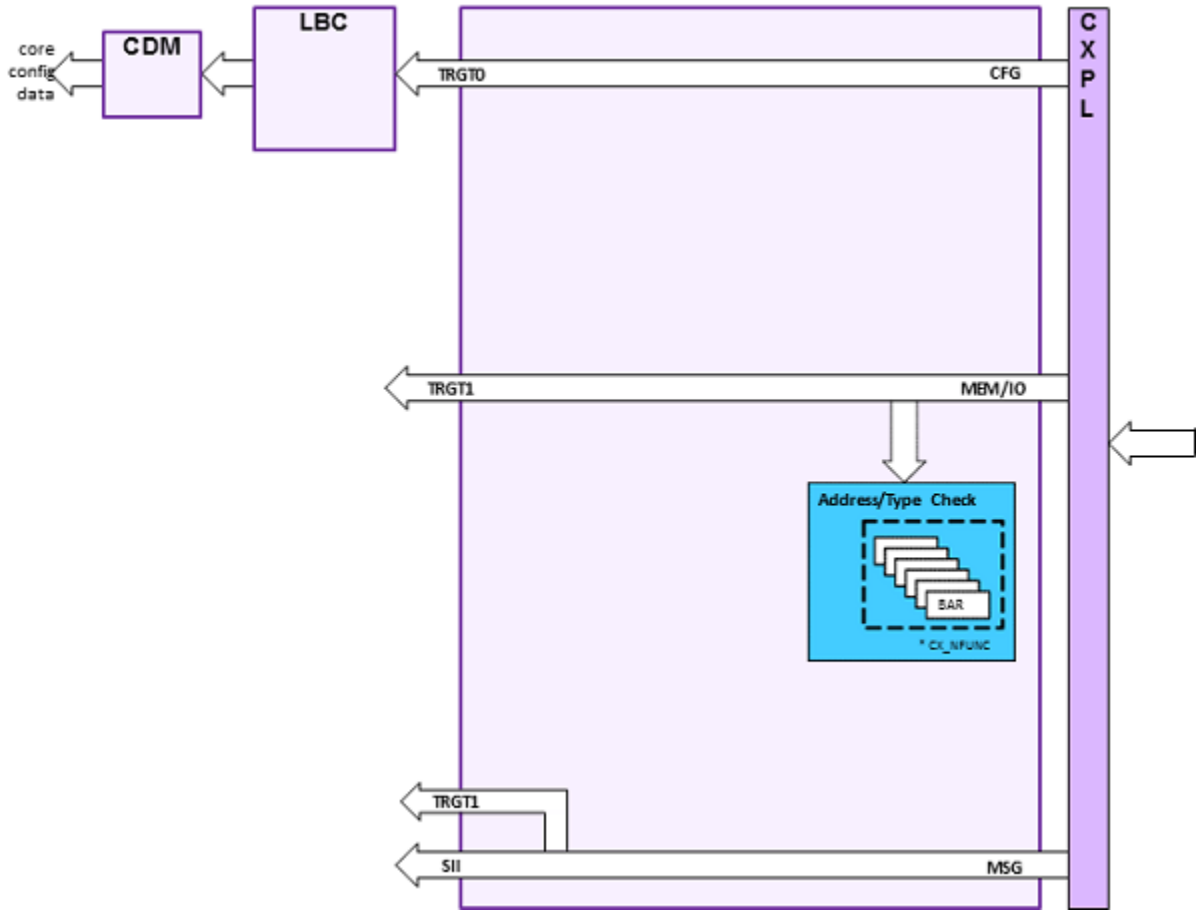


Figure 11-52. EP Mode Routing

11.3.2.1.2.2.2 RC Mode Routing Overview

The possible destinations of a posted or non-posted request are TRGT1 and Discard. By default:

- MEM requests outside of the memory range and prefetchable memory range as determined by the corresponding Base and Limit fields in the Type-1 header are routed to TRGT1.
- MSG requests are decoded internally, signaled on the SII interface, and then terminated.
- An RC does not expect to receive CFG or I/O requests.
- BARs should be disabled and not used.

The possible destinations of a completion TLP are TRGT1 and Discard.

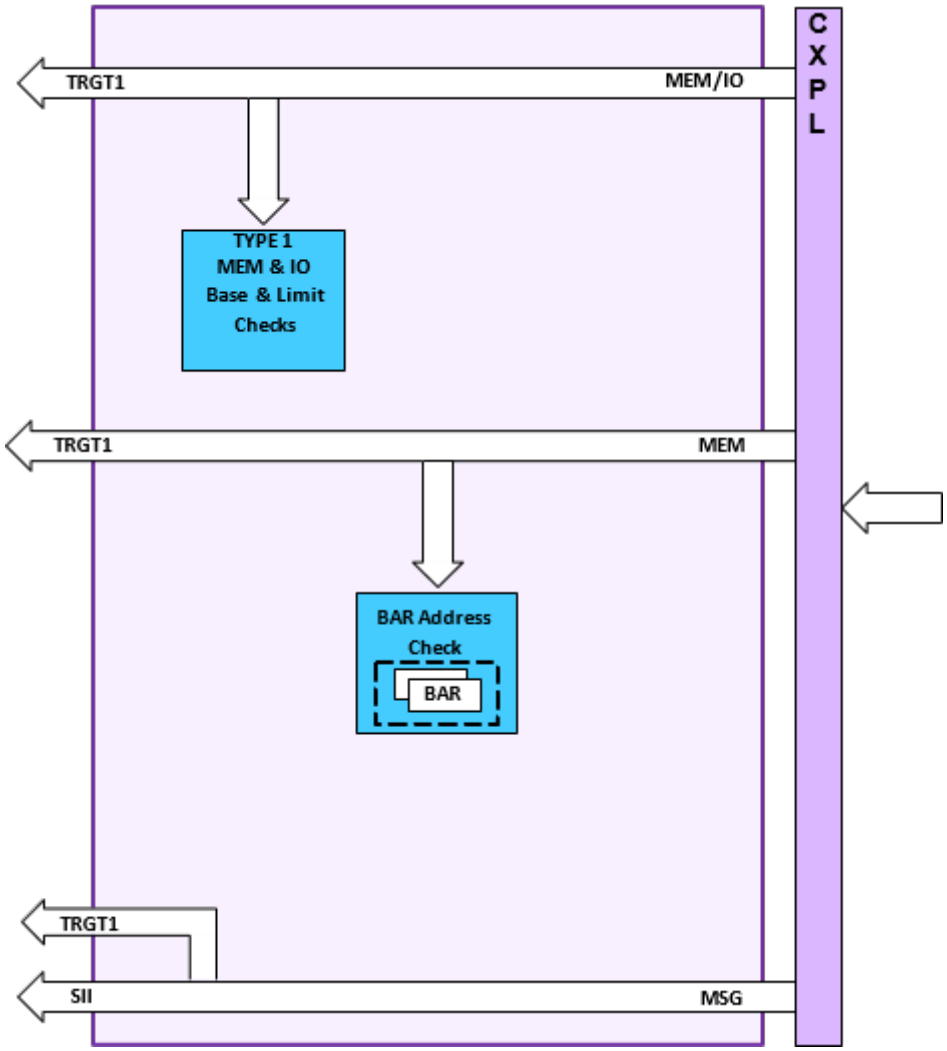


Figure 11-53. Default Request TLP Routing (Assuming no TLPs with CA/CRS/UR Error Status)

11.3.2.1.2.3 Error Handling

The controller supports the Baseline Capabilities, AER capabilities, and Advisory Non-Fatal Error Messaging as specified in section "Error Signaling and Logging" of the *PCI Express Base Specification, Revision 4.0, Version 0.7*. These include Correctable and Uncorrectable (Fatal and Non-Fatal) errors. For an RC port, the reporting of most errors is internal to the root port. No external error notifications are generated. One exception to this (for example) is unsupported request (UR) completion status.

11.3.2.1.2.3.1 Error Detection for Received TLPs

The controller performs all mandatory error detections, the error reporting mechanism based on the PCI Express Base Specification, Revision 4.0, Version 0.7. When the controller detects an error in a received TLP, it normally performs the following:

- Discards the TLP
- Generates a completion (for non-posted requests) with the completion status set to CA or UR
- Sets the status in the PCI-compatible status register
- Sets the status in the AER registers
- Generates an error MSG (upstream port only)

11.3.2.1.2.3.2 AER Multiple Header Logging

The controller supports the AER multiple header logging as specified in Section, "Multiple Error Handling (Advanced Error Reporting Capability)" of the *PCI Express Base Specification, Revision 4.0, Version 0.7*.

11.3.2.1.2.3.3 Physical Functions

By default, each physical function only logs the first header that caused an error. Per physical function, the controller implements a FIFO queue that interfaces with the AER Header Log registers (HDR_LOG_i_OFF). A valid queue entry is never overwritten. When the queue is full and your software does not read the header log registers fast enough, new header log information will be missed.

11.3.2.1.2.3.4 Virtual Functions

By default, each virtual function only logs the first header that caused an error. Each VF in each physical function has the exact same header log queue depth. Per VF, the controller implements a queuethat interfaces with the AER Header Log registers (VF_HDR_LOG_i_OFF). A valid queue entry is never overwritten. When the queue is full and your software does not read the header log registers fast enough, new header log information will be missed.

The depth of each shared queue is identical for all PFs . The queue is not a FIFO, it is just a resource-saving mechanism, and each VF can randomly access it’s own header logs independently of other VFs.

Table 11-27. Shared VF Header Log Full Behavior

VF First Error Pointer Status	VF Shared Header Log Status	Core Actions

Table continues on the next page...

Table 11-27. Shared VF Header Log Full Behavior (continued)

Invalid	Full	• A read to the Header Log returns all 1's. • A read to the "TLP Prefix Log Present" bit indicates ‘0’. • The controller sets the Uncorrectable Error Status Register according to the First Error Pointer.
Valid	Full	• The controller sets the Uncorrectable Error Status Register according to the First Error Pointer that is associated with the rejected TLP. • The Header log can be cleared or updated to the next stored header log, if one exists, by writing to the bit of the Uncorrectable Status Error Register pointed to by the First Error Pointer. • This applies to both when a normally displayed TLP Header is present in the Header Log Register or when an overflow condition is shown on the Uncorrectable Status Error Register.

11.3.2.1.3 Register Module and Data Bus Interface (DBI)

11.3.2.1.3.1 Overview

The Local Bus Controller (LBC) module provides a mechanism for a link partner (in upstream mode) or a local CPU (through the DBI) to access:

- Internal registers (in the CDM)
- External application registers connected externally to the ELBI
- In RC mode:
- There is no PCIe wire access to the CDM registers and ELBI.
- There is full DBI access to the CDM registers and ELBI.
- The controller has 4096 bytes of register PCIe configuration and port logic space per function. The controller also has a common iATU and DMA configuration register space for all functions. The controller register space is fully accessible from the DBI without any restrictions.

NOTE

Please refer to the Memory Map for the assigned space for the PCIe controller.

Table 11-28. EP Mode Port Region Access

Register Location		CDM					ELBI
Register Type		PCI-SIG		Port Logic			User
		Normal	Shadow	Misc	IATU	DMA	User
Access Details	CDM/ELBI Selector Bit[1] Value	0	0	0	1	1	1
	CS2 Selector Bit Value	0	1	0	1	1	0
DBI/Wire Access Allowed							
DBI		Y	Y	Y	Y	Y	Y
Wire	**CFG Request**	Y	-	Y	-	-	Y
	BAR Matched MEM Request	-	-	Y[2]	Y	Y	Y[-1]

1. Selector bit locations are configuration-dependent.
2. IO request also supported.

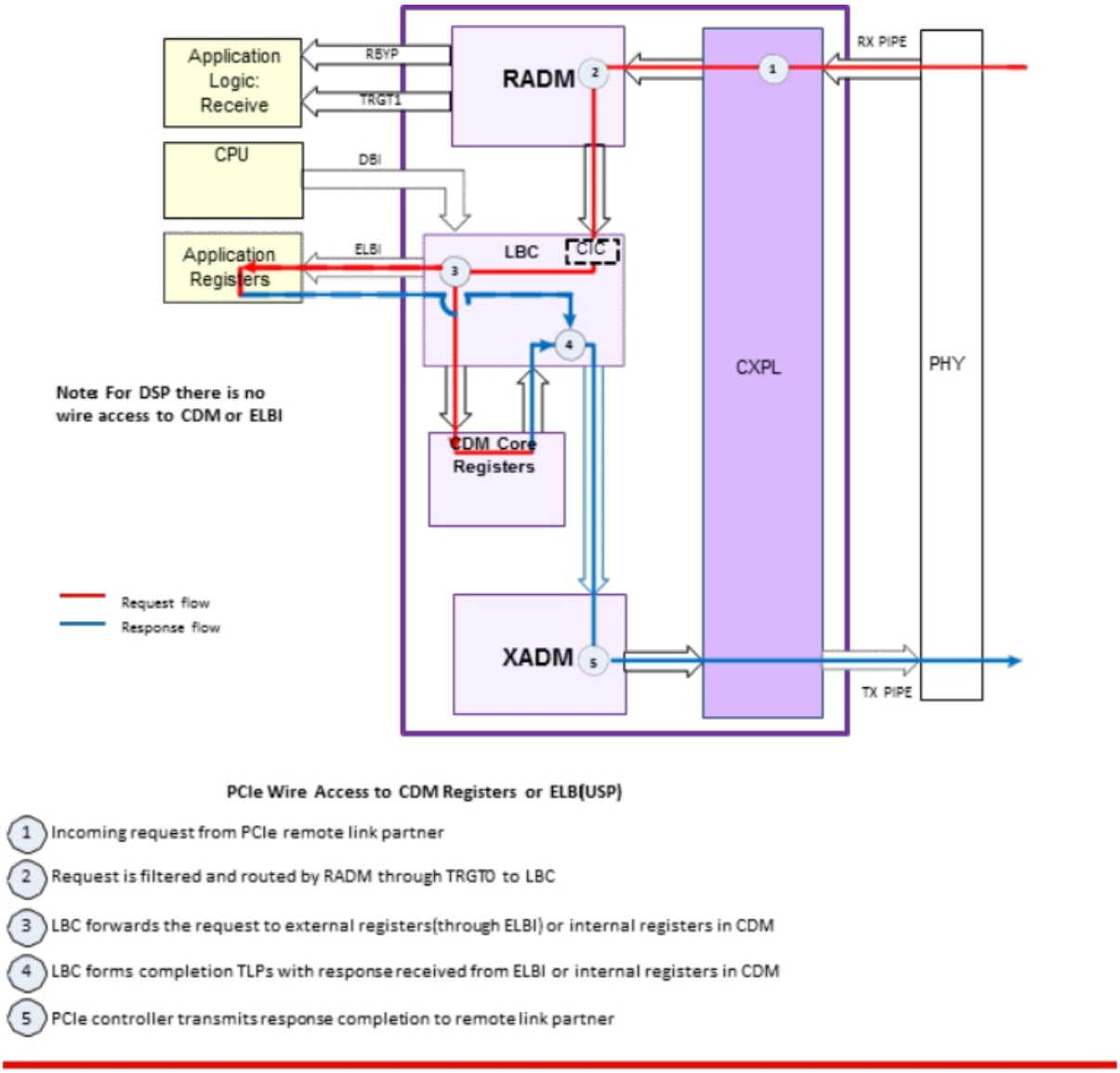


Figure 11-54. PCIe Wire Access to CDM Registers or ELB

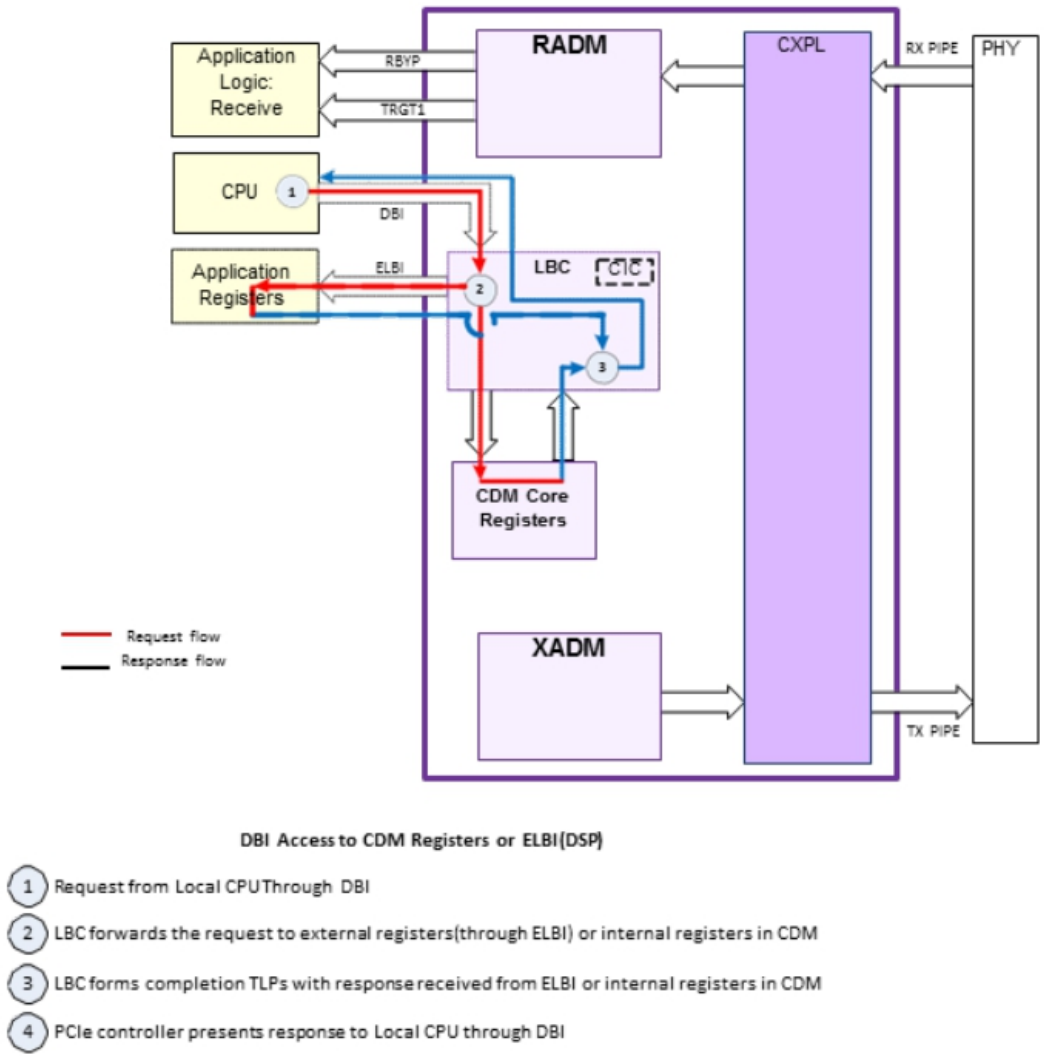


Figure 11-55. DBI Access to CDM Registers or ELBI

11.3.2.1.3.2 CDM Register Space Layout

11.3.2.1.3.2.1 Overview

The controller has 4096(0x1000) bytes of register PCIe configuration and port logic space per function. The controller also has a common iATU and DMA configuration register space for all functions. The controller register space is fully accessible from the DBI without any restrictions. Please see the Memory Map for the register base address assignment.

Table 11-29. Port Region Access

Register Location	CDM	ELBI

Table continues on the next page...

Table 11-29. Port Region Access (continued)

Register Type		PCI-SIG		Port Logic			User
		Normal	Shadow	Misc	IATU	DMA	User
Access Details	CDM/ELBI Selector Bit[1] Value	0	0	0	1	1	1
	CS2 Selector Bit Value	0	1	0	1	1	0
DBI/Wire Access Allowed							
DBI		Y	Y	Y	Y	Y	Y
Wire	**CFG Request**	Y	-	Y	-	-	Y
	BAR Matched MEM Request	-	-	Y[2]	Y	Y	Y[-1]

1. Selector bit locations are configuration-dependent.
2. IO request also supported.

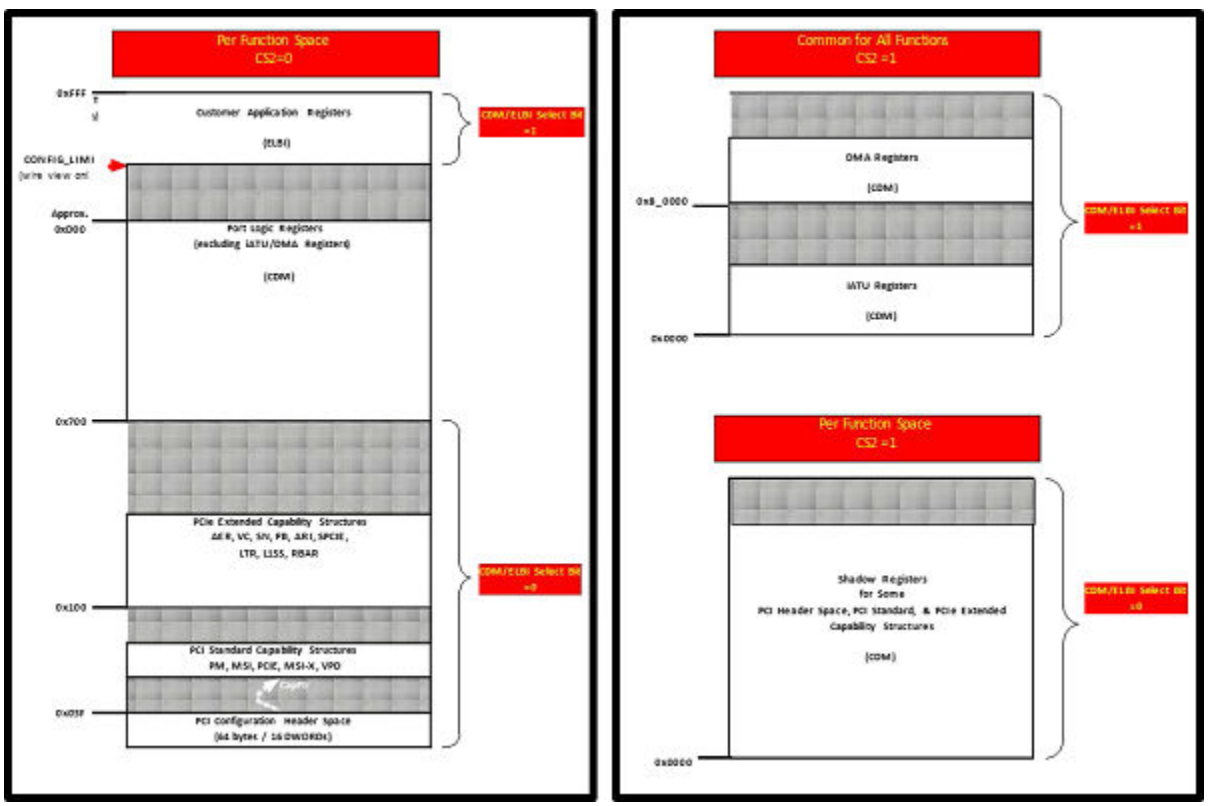


Figure 11-56. Controller Configuration Space Layout (EP Mode)

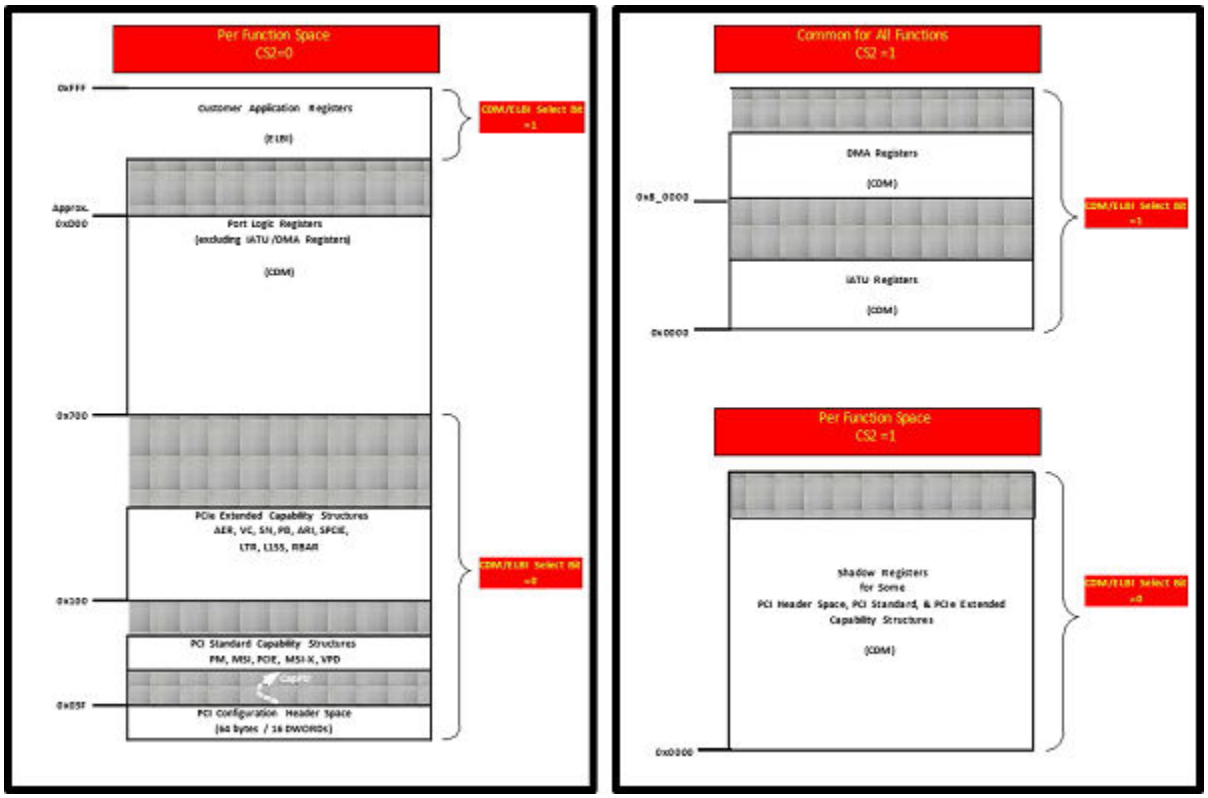


Figure 11-57. Controller Configuration Space Layout: (RC Mode)

Wire Access (Upstream Ports Only):

- Any CFG > CONFIG_LIMIT goes to TRGT1 or ELBI
- Any MEM/IO captured by a BAR whose interface target is TRGT0 goes to ELBI
- Any MEM captured by BAR*n* (*n* =UNROLL_BAR_NUM) when
- ENABLE_MEM_MAP_UNROLL_{DMA|ATU}_REG=1, whose interface target is TRGT0, and whose address falls into the range defined in "Memory Mapping iATU and DMA Port Logic Registers" goes to the IATU/DMA registers.

DBI Access (Upstream and Downstream Ports):

- No concept of MEM/IO/CFG
- Uses the specified address bit to route the access to ELBI or CDM
- No restrictions regarding what spaces can be accessed or not

11.3.2.1.3.2.2 Core Configuration Space Registers

- CDM Registers
  - PCI Configuration Header and Capability Registers:
    - The *PCI configuration header and capability registers* in the previous diagrams are PCIe controller configuration registers specified by the *PCI Express Base Specification, Revision 4.0, Version 0.7*. Access from the PCIe wire is possible with CFG requests (upstream ports). These registers are fully accessible from the DBI without any restrictions.

- Port Logic Registers:
  - The port logic (PL) registers are configuration registers which are not specified by the *PCI Express Base Specification, Revision 4.0, Version 0.7*. Access from the PCIe wire is normally with CFG requests. These registers are fully accessible from the DBI without any restrictions.
- iATU and DMA Port Logic Registers:
  - This memory space is dedicated for iATU and DMA configuration registers. Access from the PCIe wire is with specific BAR matched MEM requests. These registers are fully accessible from the DBI without any restrictions.

11.3.2.1.3.3 Data Bus Interface (DBI)

11.3.2.1.3.3.1 Overview

The DBI can access all 4096 bytes (1024 DWORDs) of the PCI Express configuration space per function. DBI can also access iATU and DMA configuration space. This address space is fully accessible from the DBI without any restrictions.

11.3.2.1.3.4 Features and Limitations

The following limitations exist for the LBC.

- CDM Access Restrictions:
  - Maximum payload Length =1 DWORD.
  - PCIe MEM requests with Length > 1 are processed as completer abort.
  - PCIe MEM requests with Length =0 are processed as completer abort.
  - PCIe IO or CFG requests with Length !=1 are processed as malformed TLPs.
- ELBI Access Restrictions:
  - CS2 bit must be 1’b0.
  - PCIe MEM requests with Length =0 are processed as completer abort.
  - PCIe IO or CFG requests with Length !=1 are processed as malformed TLPs.
  - Does not support "Atomic Operations (AtomicOps)"
  - Does not support TPH. Core does not forward the TPH bits to the ELBI.
  - Data (Read)
- DBI Data/Address Port Widths (with no AXI bridge):
  - Data: 32 bit
  - Address: 32 bit

11.3.2.1.3.4.1 Additional AXI DBI Limitations

The following limitations exist when you configure the controller to use the AXI bridge module.

- The DBI address must be a DWORD address.
- AXI transfer widths of 8, 16, and 32 bits are supported. The AXI dedicated/shared DBI interface returns an error response to requests with width greater than 32 bits. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register1.
- For AHB you get an error response whenever the size is not equal to 32 bits.
- The CDM only supports 32-bit data access. The AXI dedicated/shared DBI interface returns an error response to multi beat requests. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register.
- Byte access within a DWORD for DBI requests is not possible for AHB.
- Byte access within a DWORD for DBI WRITE requests is possible for AXI (not for AHB)
  - Dedicated DBI: The write strobes dbi_wstrb[3:0] are sent to the LBC.
  - Shared DBI: The relevant four strobes from slv_wstrb are sent to the LBC

A zero-byte write over DBI is processed as a read and should never be issued. The AXI dedicated/shared DBI slave drops a zero-byte write before it reaches the internal (native controller) DBI and returns an error response. The type of error response is programmable through AMBA_ERROR_RESPONSE_DEFAULT_OFF register.

The type of error response returned by the AXI dedicated/shared DBI slave is the same as that programmed for UR completions in the AMBA_ERROR_RESPONSE_DEFAULT_OFF register.

11.3.2.1.4 Flow Control

The flow control mechanism is divided into two phases: initialization and update. The controller automatically performs both of these phases with minimal support required from your application. By default the RADM is responsible for returning flow control credits when it reads data out of the receive queues.

11.3.2.1.5 Completion Timeout Ranges

Timeout ranges are supported as defined in the *PCI Express Base Specification, Revision 4.0, Version 0.7*. The controller supports all ranges. The Device Control 2 Register has a default equal to the default in the specification: 0000b Default range: 50 us to 50 ms. When the default is used, then the timeout is in Range B: 0101b: 16 ms to 55 ms. This range was chosen for the default because the *PCI Express Base Specification, Revision*

4.0, Version 0.7 states, "It is strongly recommended that the completion Timeout mechanism not expire in less than 10 ms". The table below illustrates the specification values versus the PCI Express controller values for the ranges.

NOTE

As the PCIe specification states, This mechanism is intended to be activated only when there is no reasonable expectation that the completion is returned, and should never occur under normal operating conditions.

Table 11-30. Comparison of PCIe Specification and PCIe Core Completion Timeout Ranges

Range	Encoding	Spec Minimum	Spec Maximum	PCIe controller Minimum	PCIe controller Maximum
Default	0000b	50 us	50 ms	28 ms	45 ms
A	0001b	50 us	100 us	65 us	99 ms
A	0010b	1 ms	10 ms	4.1 ms	6.2 ms
B	0101b	16 ms	55 ms	28 ms	44 ms
B	0110b	65 ms	210 ms	86 ms	131 ms
C	1001b	260 ms	900 ms	260 ms	390 ms
C	1010b	1s	3.5 s	1.8 s	2.8 s
D	1101b	4s	13 s	5.4 s	8.2 s
D	1110b	17s	64 s	38 s	58 s

11.3.2.1.6 Crosslink

Crosslink allows a downstream port to be connected to another downstream port or an upstream port to be connected to another upstream port. When a crosslink capable port negotiates a crosslink connection, the port changes its behavior accordingly. For this reason crosslink is supported in the DM controller because it contains the functionality for both upstream and downstream support.

Crosslink provides more than simply the ability to connect two like ports together. For instance, a DM operating in EP mode that negotiates a crosslink connection switches to RC mode and follows all of the *PCI Express Base Specification, Revision 4.0, Version 0.7* rules for an RC.

11.3.2.2 DMA Controller

This section describes the embedded multichannel DMA controller.

11.3.2.2.1 DMA Overview

The RC system CPU, or the EP application CPU, can offload the transferring of large blocks of data to the embedded DMA controller, leaving the CPU free to perform other tasks. It can simultaneously perform the following types of memory transactions (as shown in the figure 'System Level View of DMA'):

DMA write: Transfer (copy) of a block of data from local (application) memory to remote (link partner) memory.

DMA read: Transfer (copy) of a block of data from remote (link partner) memory to local (application) memory.

Therefore the DMA supports full duplex operation, processing read and write transfers at the same time, and in parallel with normal (non-DMA) traffic. Upon completion of a DMA transfer or an error, the DMA optionally interrupts the local CPU or sends an interrupt MWr (IMWr) to the remote CPU.

In linked list mode, the DMA fetches the transfer control information (called channel context) for each transfer (block), from a list of DMA elements that you have constructed in local memory.

11.3.2.2.2 DMA Architecture

This section describes the architecture of the DMA.

11.3.2.2.2.1 Architecture Overview

A DMA write and read channel operate independently to maximize the performance of the DMA read and write data transfers over the PCIe link. DMA with multiple read channels uses a weighted round robin (WRR) arbitration scheme to select the next read channel to be serviced. The same applies for multiple write channels.

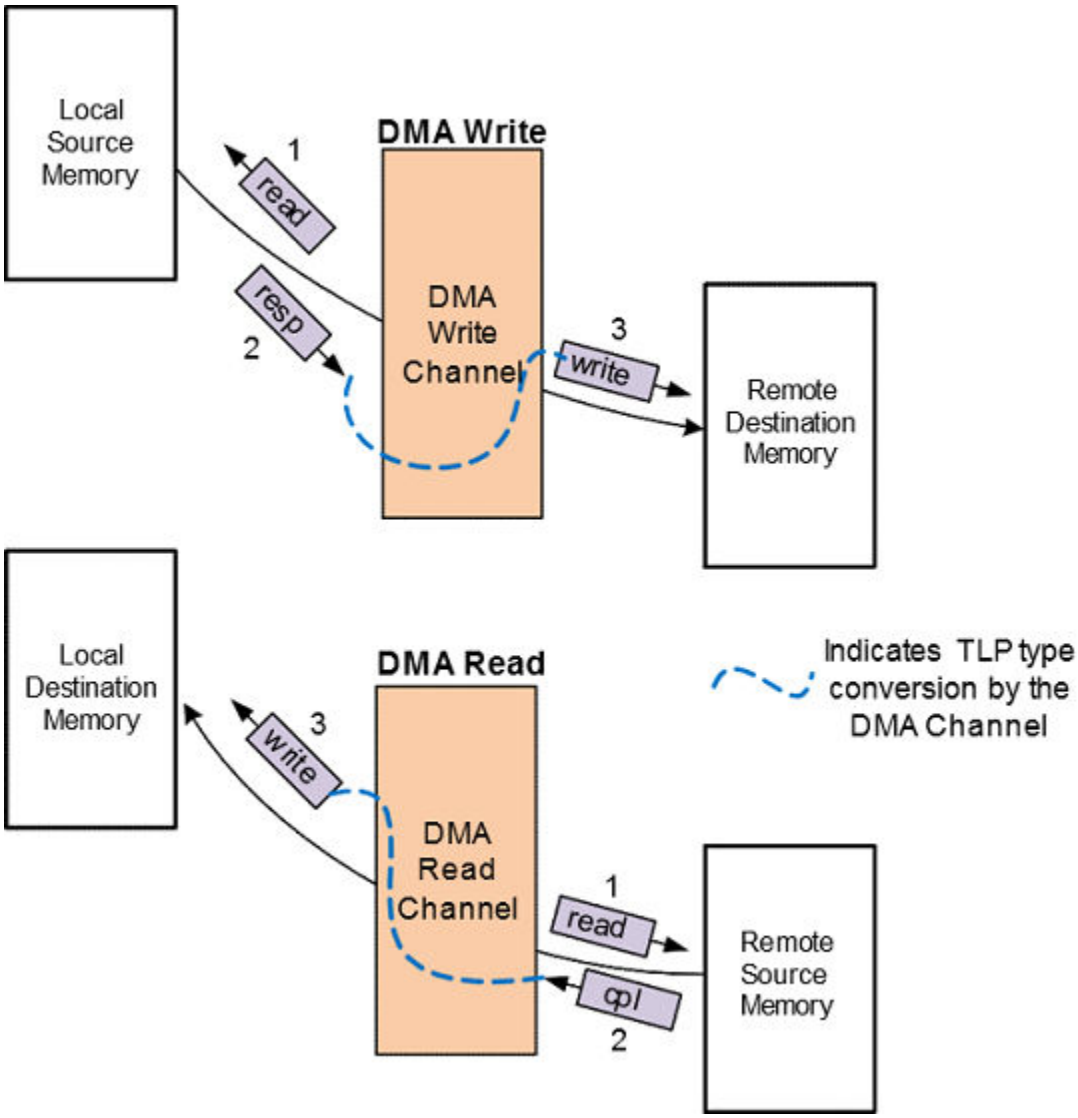


Figure 11-58. System Level View of DMA

11.3.2.2.2.1.1 DMA Read Transfer

The DMA injects multiple MRd requests of size less than or equal to the minimum of {Max Read Request Size, Max Payload Size} into the outbound request path, directed toward the remote link partner. The DMA converts the read responses into write requests, that it then transmits to the local application. When the DMA data transfer is complete, the CPU is notified. For a read transfer, the SAR is the address of the remote memory, and the DAR is the address of the local memory.

11.3.2.2.2.1.2 DMA Write Transfer

The DMA injects multiple MRd requests of size less than or equal to Max_Payload_Size into the inbound request path, directed toward the local application. The DMA converts the read responses into MWr TLPs, that it then transmits to the remote link partner. When the DMA data transfer is complete, the CPU is notified. For a write transfer, the SAR is the address of the local memory, and the DAR is the address of the remote memory.

11.3.2.2.2.2 Interrupts and Error Handling

11.3.2.2.2.2.1 Interrupt Handling

The DMA generates two interrupts per channel:

Done: The DMA successfully completes the transfer.

Abort: The DMA fails to complete the transfer, or an error occurs during the transfer.

The interrupt handling mechanism is different for linked list (LL) mode (than non LL mode), and there are also some differences between the read and write channels.

11.3.2.2.2.2.2 Non Linked List Mode

You enable the local and remote interrupts through the local and remote interrupt enable (LIE and RIE) bits:

DMA_CH_CONTROL1_OFF_WRCH_0.lie and
DMA_CH_CONTROL1_OFF_WRCH_0.rie.

In the write channel, there is only one error condition that results in an abort interrupt. For more details, see "Linked List Mode". You mask, clear, and read the status of each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in figure 'Write Interrupt Generation - Non Linked List Mode - Shown For Write Channel #0'.

In the read channel, there are five error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in figure 'Read Interrupt Generation - Non Linked List Mode - Shown For Read Channel #0'. However, you can read the status of each of the five abort errors (that contribute to the abort interrupt) through DMA_READ_ERR_STATUS_LOW_OFF and DMA_READ_ERR_STATUS_HIGH_OFF.

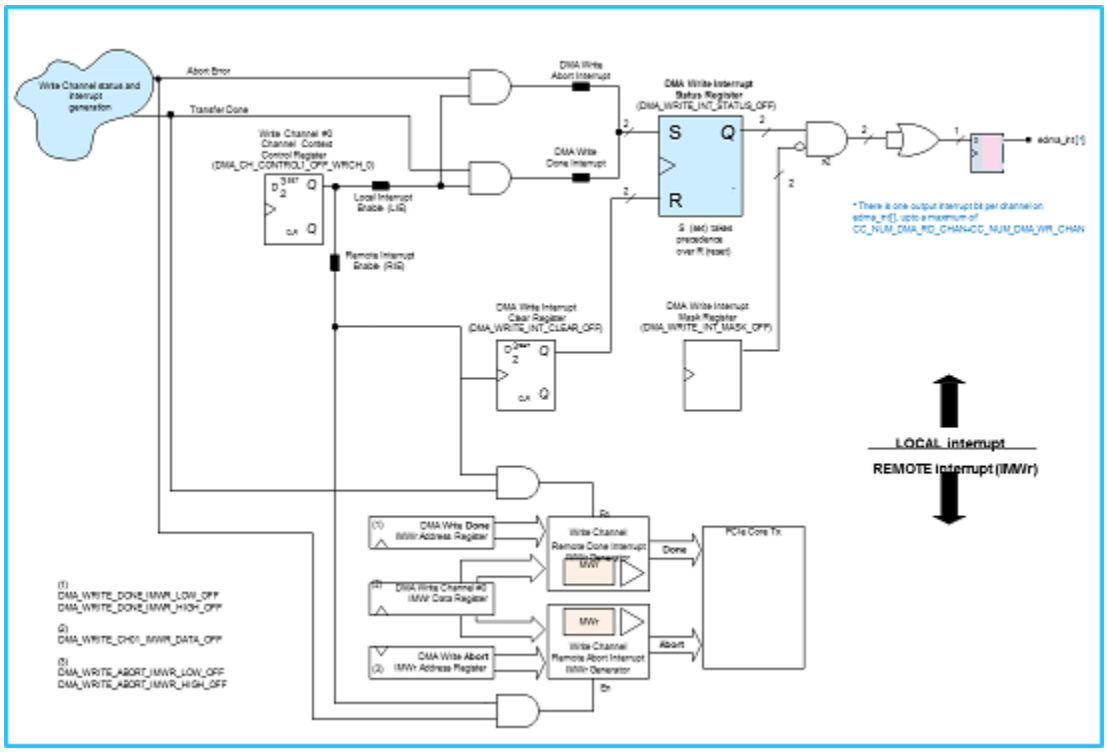


Figure 11-59. Write Interrupt Generation - Non Linked List Mode - Shown For Write Channel #0

If you want a remote interrupt and not a local interrupt then:

- Set LIE and RIE
- Mask the local interrupt using the mask register

This will allow you to poll the status register to distinguish a DONE from an ABORT

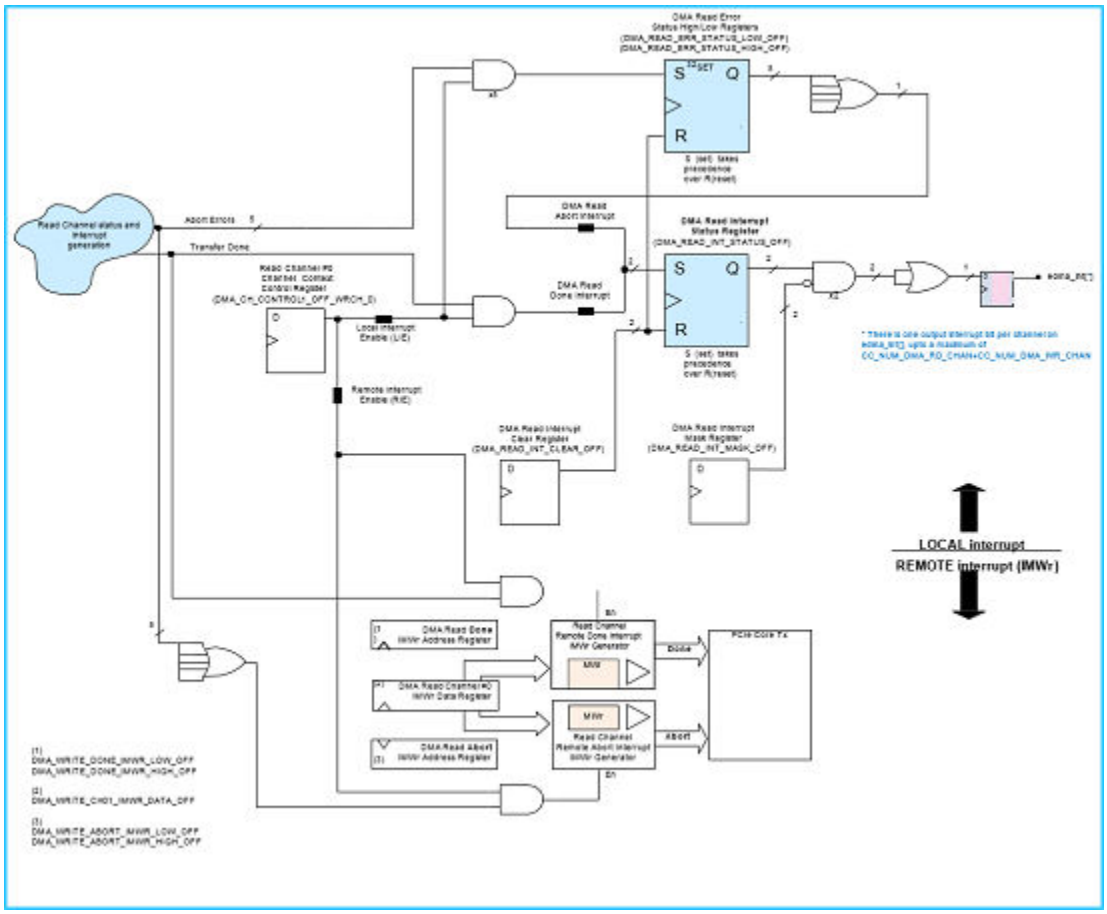


Figure 11-60. Read Interrupt Generation - Non Linked List Mode - Shown For Read Channel #0

11.3.2.2.2.2.3 Linked List Mode

The LIE and RIE bits in the LL element enable the channel done interrupts (local and remote). The LLLAIE and LLRAIE bits of the DMA_WRITE_LINKED_LIST_ERR_EN_OFF/ DMA_READ_LINKED_LIST_ERR_EN_OFF registers enable the channel abort interrupts (local and remote). In the write channel, there are two error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in the figure below. You can read the status of each of the two abort errors (that contribute to the abort interrupt) through the DMA_WRITE_ERR_STATUS_OFF register.

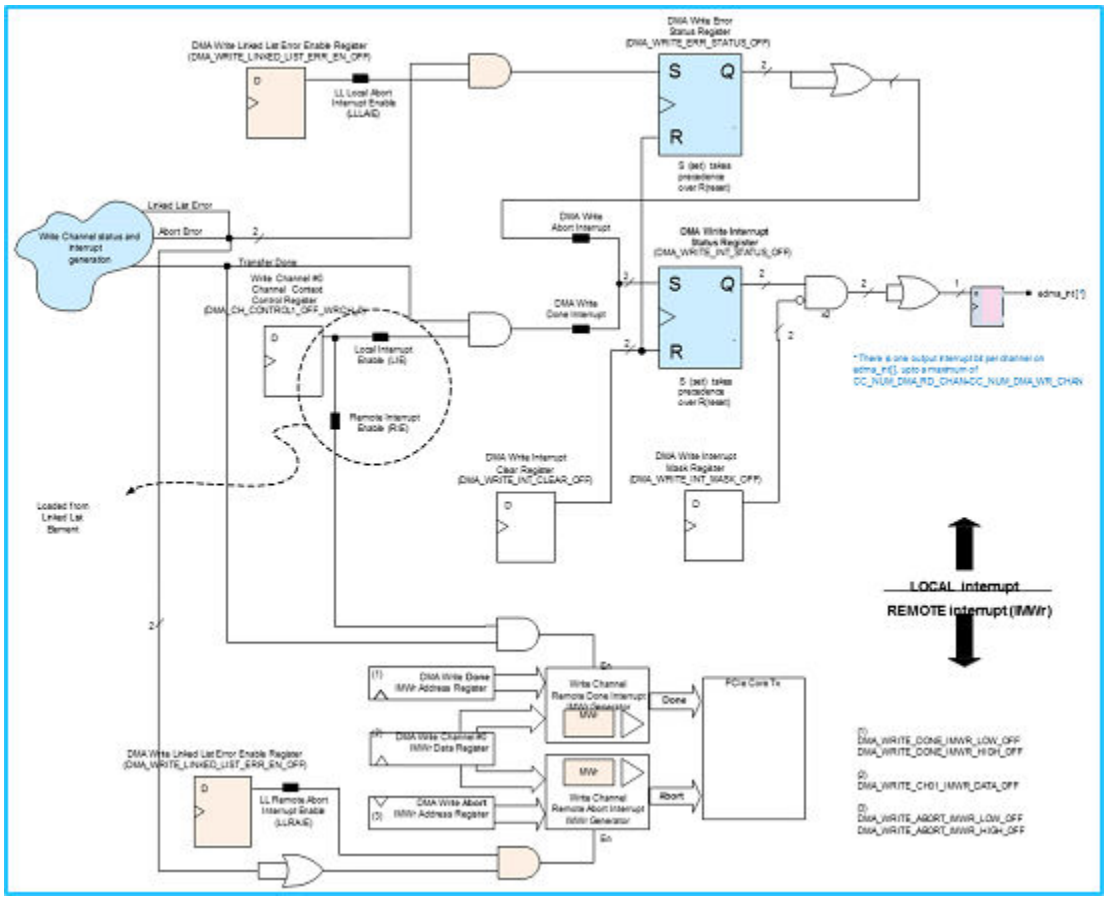


Figure 11-61. Write Interrupt Generation - Linked List Mode - Shown For Write Channel #0

In the read channel, there are six error conditions that results in an abort interrupt. For more details, see "Linked List Mode". You mask and clear each of the two interrupts (done and abort) through the DMA interrupt registers as indicated in the figure below. You can read the status of each of the six abort errors (that contribute to the abort interrupt) through the DMA_READ_ERR_STATUS_LOW_OFF and DMA_READ_ERR_STATUS_HIGH_OFF registers.

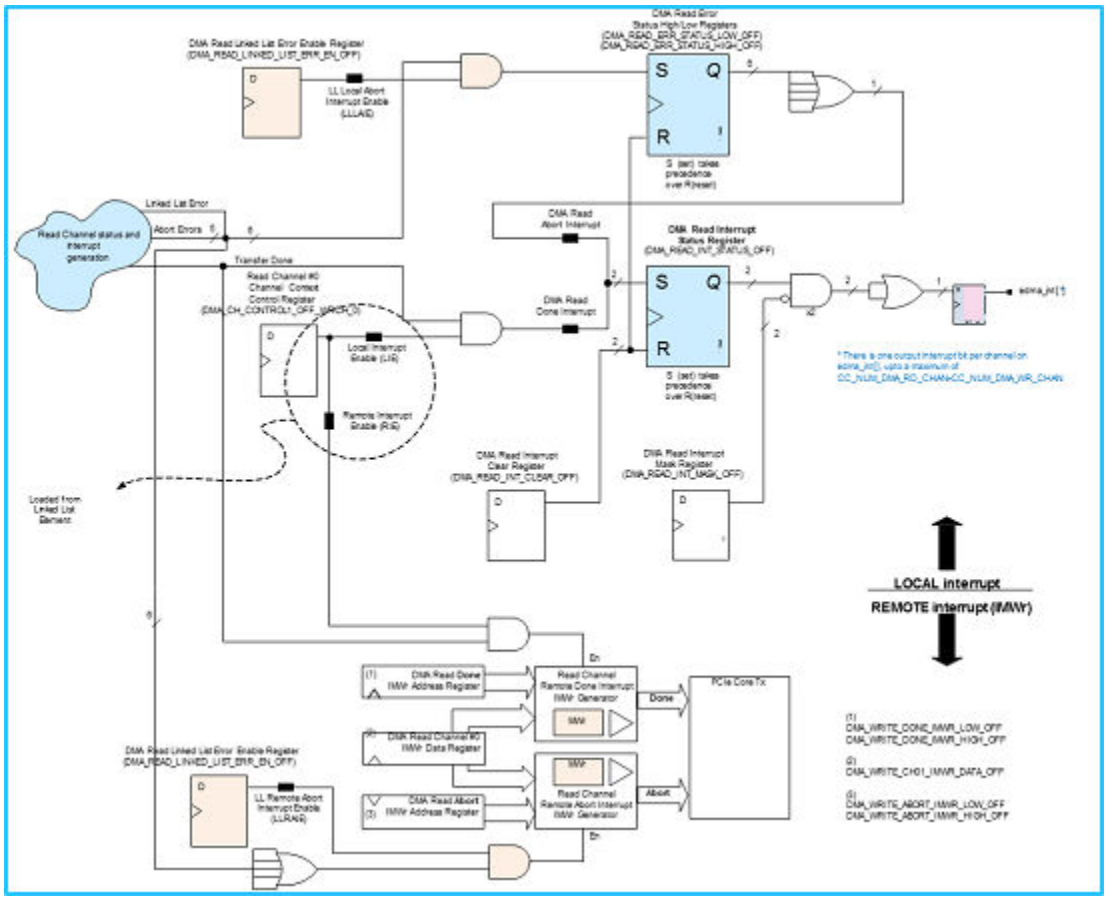


Figure 11-62. Read Interrupt Generation - Linked List Mode - Shown For Read Channel #0

NOTE

In non-linked list mode, LIE acts as a global switch. However when in linked list mode, LIE is just local to the current linked list element and the global switch is LLLAIE.

NOTE

If the DMA driver is running on the host and the interrupt service routine is reading local interrupts to determine if the transfer is successful, then you must set LIE and RIE in the same element and you should mask or ignore the local interrupt pin. Setting RIE and LIE in element A followed by RIE (only) in element B is not a verified usage scenario.

11.3.2.2.2.3 Linked List Mode

The DMA provides a linked list (LL) mode to efficiently move data from source to destination with minimal intervention from the local CPU. This mode provides an alternative to programming the DMA multiple times to transfer multiple blocks of data. The programming information (address, size, and so on) for each block of memory is pre-programmed by your software into a LL element (also known as a descriptor) in local memory. Each element (called a data element) in the LL structure (called a transfer list) can transfer up to 4 GB of data. You enable LL operation for a channel, by setting the LLE field of the DMA_CH_CONTROL1_OFF_WRCH_0 (Replace WRCH with RDCH for a read channel) register to 1. You can enable LL mode independently for each channel. When you enable LL for more than one channel, then you must have a separate LL structure in local memory for each channel. Your application must produce the LL element structure in local memory as shown in the figure below. Normally, all of the elements are contiguous (one after the other) in memory, and each element has six DWORDs containing the information about the block of data to be transferred. You program the channel context registers (DMA_LLP_LOW_REG_WRCH_0 and DMA_LLP_HIGH_OFF_WRCH_0) with the location of where you have placed the LL element structure in local memory.

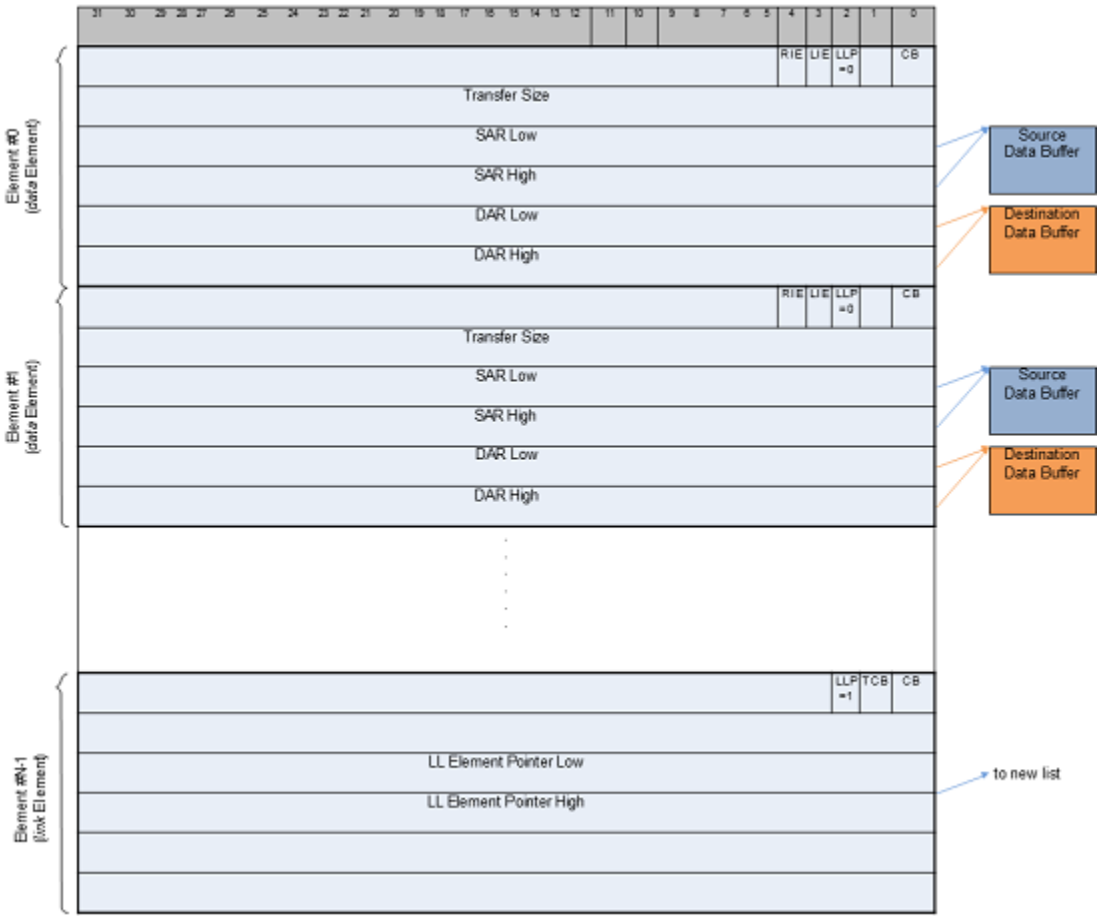


Figure 11-63. Linked List Element/Descriptor Structure in Local Memory with N Elements

When you start the DMA transfer (by writing to the DMA Write Doorbell Register DMA_WRITE_DOORBELL_OFF or DMA Read Doorbell Register DMA_READ_DOORBELL_OFF), the DMA reads (consumes) each element from local memory, and loads the information (SAR, DAR, size, and so on) from that element into the channel context registers in the DMA. These channel context registers determine the operation of the channel that the DMA controller is currently servicing. The DMA then proceeds to transfer the block of data (as defined by the element), and when it is finished, reads the next element from local memory. Normally, all of the elements are contiguous (one after the other) in memory, with the starting address defined in the channel context DMA Linked List Pointer Low Register DMA_LLP_LOW_OFF_WRCH_0. When you want to jump in local memory to another element list (or recycle the consumed elements), then you set the LLP bit in the element (for example, link element #N-1 in figure shown)), specify the location of the next element structure using the LL Element Pointer DWORDs (as indicated in the figure), and, set TCB to 1 (for recycling) or to 0 (to jump to another list).

11.3.2.2.2.3.1 Relationship Between Element DWORDs and Channel Context Registers

Notice the similarity between a data element and the DMA Channel Context registers for each channel. Each element has six DWORDs as in figure 'Linked List Element/ Descriptor Structure in Local Memory with N Elements'. There are eight channel context registers (DWORDs) for a channel. The DMA loads the six element DWORDs into the following channel context:

- CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
- DMA Transfer Size
- DMA SAR Low and DMA SAR High
- DMA DAR Low and DMA DAR High

The definitions of the element DWORD bit fields are the same as the DMA Channel Context registers described in the "DMA Software Register Map", with the exception of the LIE and RIE bits. The LIE and RIE bits in a LL element, only enable the done interrupt. In non-LL mode, the RIE and LIE bits (in the channel context registers) enable the done and abort interrupts.

When the LLP field of the first DWORD in the element is set to 1, the element is a link element. The DMA only loads the following information into the channel context registers:

- CB, TCB, and LLP bits of the first DWORD. The LIE and RIE bits are not defined in a link element.
- LL Element Pointer (3rd and 4th DWORDs) into the DMA Linked List Pointer registers (DMA_LL- P_LOW_REG_WRCH_0 and DMA_LLP_HIGH_OFF_WRCH_0).

Linked List Operation

- When you enable linked list operation (DMA_CH_CONTROL1_OFF_WRCH_0.lle =1b1), then the DMA overwrites the following DMA channel context registers with the following information from linked list data elements:
  - The CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
  - DMA Transfer Size
  - DMA SAR Low & High
  - DMA DAR Low & High
- The structure of a link element is different to that of the data element. A data element has no TCB field. A link element has no LIE or RIE fields. It has no SAR, DAR, or Transfer Size DWORDs, but has LL Element Pointer DWORDs instead of the SAR DWORDs.
- The LIE and RIE bits in a LL element, only enable the done interrupt. In non-LL mode, the RIE and LIE bits enable the done and abort interrupts. For more information, see "Interrupts and Error Handling".

11.3.2.2.3 Using the DMA

This section describes how to use the DMA. It can simultaneously perform the following types of memory transactions:

- DMA write : Transfer (copy) of a block of data from local memory to remote memory
- DMA read :Transfer (copy) of a block of data from remote memory to local memory

Therefore the DMA supports full duplex operation, processing read and write transfers at the same time, and in parallel with normal (non-DMA) traffic. Upon completion of a DMA transfer or an error, the DMA sends an interrupt MWr (IMWr).

The channels are named read channel 7..0 and write channel 7..0. Each channel is programmed using a number of registers.

11.3.2.2.3.1 Source and Destination Address Registers (SAR, DAR)

The DMA channel context SAR and DAR registers (DMA_SAR_LOW_OFF_WRCH_01 / DMA_SAR_HIGH_OFF_WRCH_0 / DMA_DAR_LOW_OFF_WRCH_0 / DMA_DAR_HIGH_OFF_WRCH_0) provide support for remote-to-local, and local-to-remote PCIe address mapping. You program the start of the local and remote data buffers using these registers, and the DMA increments the SAR and DAR as the DMA transfer progresses. For a write transfer, the SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in Figure 'Write Transfer: SAR and DAR for Write Channel 0'.

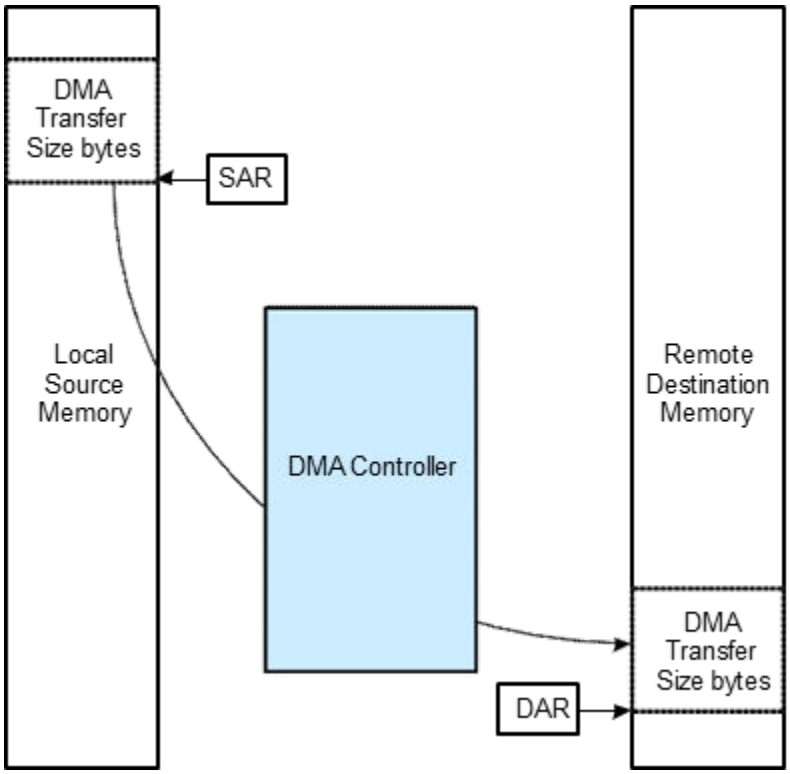


Figure 11-64. Write Transfer: SAR and DAR for Write Channel 0

For a read transfer, the SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in the figure below.

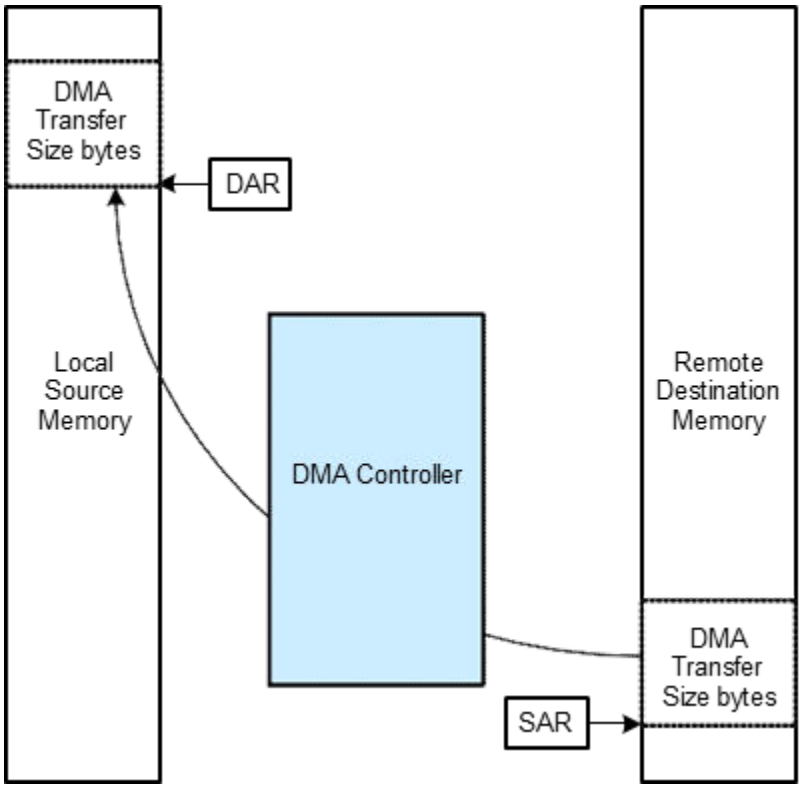


Figure 11-65. Read Transfer: SAR and DAR for Read Channel 0

The presence of the DMA controller does not affect:

- Normal filtering rules for inbound TLPs (for example, BAR checking in an Endpoint, as described in as described in “Receive Filtering”
- The operation of any internal or external address translation as described in “Internal Address Translation Unit (iATU)”).

NOTE

When you do not want the iATU to translate outbound requests that are generated by the internal DMA module, then you must implement one of the following approaches:

- Ensure that the combination of DMA channel programming and iATU control register programming, causes no translation of DMA traffic to be done in the iATU.
- Activate the DMA bypass mode to allow request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. You can activate DMA bypass mode by setting the DMA Bypass field of the iATU Control 2 Register (IATU_REGION_C-TRL_OFF_2_OUTBOUND_0).

11.3.2.2.3.2 DMA Transfer Size Register

You program the DMA transfer size using the DMA Transfer Size Register (DMA_TRANSFER_SIZE_OFF_WRCH_0). The DMA decrements the value in this register as the DMA transfer progresses. You can read this register at any time to determine the number of bytes remaining to be transferred. When all bytes are successfully transferred, the result is zero. The DMA transfer size can be of any value from one byte, up to a maximum of 4 GB.

11.3.2.2.3.3 Starting The DMA Transfer

After you program the DMA controller registers (including writing to the DMA Read Engine Enable or DMA Write Engine Enable register), you start a DMA transfer by writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF). You can program and start both a read and a write transfer at the same time. The DMA supports full duplex operation, processing read and write transfers at the same time and in parallel with normal (non-DMA) traffic. You can program and start any number of channels in the DMA controller sequentially or simultaneously.

Sequentially: configure and start a channel, and then configure and start another channel

Simultaneously: configure multiple channels, and then start multiple channels

NOTE

You must not write to any of the context registers for a particular channel after you start the channel by writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF).

11.3.2.2.3.4 Detecting the End of The DMA Transfer

If a DMA transfer proceeds without any errors, it stops automatically when finished.

11.3.2.2.3.4.1 Detecting End of Transfer without Errors

The normal end of a DMA transfer is detected by any of the following methods:

- Done local interrupt (pin) asserted.
- Done remote interrupt (IMWr) received.

- Channel status field of the Channel Control 1 register is Stopped, and the DMA Transfer Size register is 0x0.
- Polling of the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF).

11.3.2.2.3.4.2 Detecting End of Transfer with Errors

The abnormal end of a DMA transfer is detected by any of the following methods:

- Abort local interrupt (pin) asserted.
- Abort remote interrupt (IMWr) received.
- Polling of the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF).
- Channel Status field of the Channel Control 1 register is halted. When the DMA controller detects an error, it forces the DMA to stop issuing requests for the channel. It also sets the channel status field of the Channel Control 1 register to halted, generates an abort interrupt (if enabled), and sends an abort IMWr (if enabled). The DMA Transfer Size register indicates the remaining number bytes to be transferred, except when there is an AXI write error during a DMA read transfer. For more details, see table 'Possible Sources of an Error during a DMA Read Transfer'.
- Channel Status field of the Channel Control 1 register is Stopped, and the DMA Transfer Size register is not 0x0. You have prematurely stopped this channel as described in “Stopping the DMA Transfer (Software Stop)”.

11.3.2.2.3.5 Stopping the DMA Transfer (Software Stop)

You can manually abort (stop) the DMA transfer by writing the channel number to the Doorbell Number field and writing 1 to the Stop field in DMA_WRITE_DOORBELL_OFF. This causes the DMA to:

- Place the channel in a Stopped state.
- The channel Status field in DMA_CH_CONTROL1_OFF_WRCH_0 is Stopped and the value in DMA_TRANS- FER_SIZE_OFF_WRCH_0 will not be 0x0.
- Wait for all outstanding pending transactions.
- Assert the abort interrupt (if it is enabled) in DMA_WRITE_INT_STATUS_OFF.

You might do this as part of error handling which is described in “Error Handling Assistance by Remote Software”, which is only necessary during software development if you incorrectly program the DMA write channel DAR. You might also do this as part of a function level reset (FLR). FLR does not directly affect the DMA transfer so you must manually stop the DMA transfer before initiating an FLR.

Use This Feature With Caution in Linked List Mode

You should not use the STOP feature in operational mode. It is only a debug feature and has associated hazards.

- Before setting the Stop bit, you must read the channel status field (CS) of the DMA Channel Control 1 register to ensure that the corresponding channel is Running (transferring data). To eliminate the possibility of a race condition between these two actions (read and write), you should confirm the presence of the abort bit in the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_STATUS_OFF) and check the status of any fatal error if some other event has occurred.
- After a Stop event, you cannot seamlessly resume the transfer again because the DMA will not carry on exactly from the point that it was stopped at. Therefore you must setup and start the complete channel context again, including the
  - linked list pointer
  - linked list structure in memory
  - related PCS and CCS values

11.3.2.2.3.6 TLP Generator Header Control

You must program bits [31:12] of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_REG_ WRCH_0/DMA_CH_CONTROL1_OFF_RDCH_0), so that the DMA TLP generator correctly sets the TC, RO, NS, AT, and FN TLP header fields for any TLPs that it generates. When SRIOV is enabled, then you must also program DMA_CH_CONTROL2_OFF_WRCH_0/ DMA_CH_CONTROL2_OFF_RDCH_0.

11.3.2.2.3.7 DMA Software Register Map

This section discusses the grouping and access methods of the DMA registers. The following registers are present when the DMA is enabled.

The DMA Global Registers control the global settings

Table 11-31. DMA Global Registers

Name	Description
0x8_0000 +0x000	DMA Arbitration Scheme for TRGT1 Interface
+0x008	Number of DMA Channels
+0x00C	DMA Write Engine Enable
+0x010	DMA Write Doorbell
+0x018	DMA Write Engine Channel Arbitration Weight Low
+0x01C	DMA Write Engine Channel Arbitration Weight High
+0x02C	DMA Read Engine Enable
+0x030	DMA Read Doorbell
+0x034	DMA Global Read Control
+0x038	DMA Read Engine Channel Arbitration Weight Low
+0x03C	DMA Read Engine Channel Arbitration Weight High

The DMA Interrupt Registers control the generation of local interrupts and remote IMWr. The IMWr addresses and data are stored in RAM.

Table 11-32. DMA Interrupt Registers

Byte Offset	Description
0x8_0000 +0x4C	DMA Write Interrupt Status
+0x50	Not used (RsvdP).
+0x54	DMA Write Interrupt Mask
+0x58	DMA Write Interrupt Clear
+0x5C	DMA Write Error Status
+0x60	DMA Write Done IMWr Address Low
+0x64	DMA Write Done IMWr Address High
+0x68	DMA Write Abort IMWr Address Low
+0x6C	DMA Write Abort IMWr Address High
+0x70	DMA Write Channels 1 and 0 IMWr Data
+0x84	DMA Write Channels 3 and 2 IMWr Data
+0x88	DMA Write Channels 5 and 4 IMWr Data
+0x8C	DMA Write Channels 7 and 6 IMWr Data
+0x90	DMA Write Linked List Error Enable
up to +0x9C	Not used (RsvdP).
+0xA0	DMA Read Interrupt Status
+0xA4	Not used (RsvdP).
+0xA8	DMA Read Interrupt Mask
+0xAC	DMA Read Interrupt Clear
+0xB0	Not used (RsvdP).
+0xB4	DMA Read Error Status Low
+0xB8	DMA Read Error Status High

Table continues on the next page...

Table 11-32. DMA Interrupt Registers (continued)

up to +0xC0	Not used (RsvdP).
+0xC4	DMA Read Linked List Error Enable
+0xC8	Not used (RsvdP).
+0xCC	DMA Read Done IMWr Address Low
+0xD0	DMA Read Done IMWr Address High
+0xD4	DMA Read Abort IMWr Address Low
+0xD8	DMA Read Abort IMWr Address High
+0xDC	DMA Read Channels 1 and 0 IMWr Data
+0xE0	DMA Read Channels 3 and 2 IMWr Data
+0xE4	DMA Read Channels 5 and 4 IMWr Data
+0xE8	DMA Read Channels 7 and 6 IMWr Data

DMA Channel Context Registers: these registers (which are stored in RAM) are for the control and status information that is specific to each channel. There is a set of registers for each channel.

Table 11-33. DMA Channel Context Registers for Each Channel

Byte Offset	Description
DMA Channel Context	
0x8_0000 + 0x200*ia + 0x100*(WR|RD)b	DMA Channel Control 1 Register
+0x004*i + 0x200*i + 0x100 *(WR|RD)	DMA Channel Control 2 Register
+0x008*i + 0x200*i + 0x100 *(WR|RD)	DMA Transfer Size
+0x00C*i +0x200*i + 0x100 *(WR|RD)	DMA SAR Low
+0x010*i + 0x200*i + 0x100 *(WR|RD)	DMA SAR High
+0x014*i + 0x200*i + 0x100 *(WR|RD)	DMA DAR Low
+0x018*i + 0x200*i + 0x100 *(WR|RD)	DMA DAR High
+0x01C*i + 0x200*i + 0x100 *(WR|RD)	DMA Linked List Pointer Low
+0x020*i + 0x200*i + 0x100 *(WR|RD)	DMA Linked List Pointer High

a. i is the number of channels

b. WR=0 | RD=1

11.3.2.2.3.7.1 Implementation of Channel Context Registers

- The channel context registers are stored in RAM whose contents are automatically initialized after power-on.
  - Therefore, you must program every “register” value as the default is undefined

  - You must program all fields marked Reserved to 1’b0
  - These registers are not affected by any of the reset signals
- You must not write to any of the context registers for a particular channel after you start the channel by writing the channel number to the Doorbell Number field of the DMA Write Door- bell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_OFF).

Linked List Operation

When you enable linked list operation (Linked List Enable field of DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) t o 1b1), the DMA overwrites the following DMA channel context registers with the following information from linked list data elements:

- The CB, LLP, LIE, and RIE fields of the DMA Channel Control 1 register.
- DMA Transfer Size
- DMA SAR Low
- DMA SAR High
- DMA DAR Low
- DMA DAR High

NOTE

- The structure of a link element is different to that of the data element. A data element has no TCB field. A link element has no LIE or RIE fields. It has no SAR, DAR, or Transfer Size DWORDs, but has LL Element Pointer DWORDs instead of the SAR DWORDs. For more details, see Relationship Between Element DWORDs and Channel Context Registers.
- The LIE and RIE bits in a LL element, only enable the Done interrupt. In non-LL mode, the RIE and LIE bits enable the Done and Abort interrupts. For more information, see Interrupts and Error Handling.

11.3.2.2.3.8 Programming Examples

This section provides a programming and operation flow for each of the following scenarios:

- 1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)
- 1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)
- 1 MB Write Transfer On Write Channel #5 Initiated by Remote CPU (Non LL mode)

- 1 MB Read Transfer On Read Channel #5 Initiated by Remote CPU (Non LL mode)
- Linked List Mode Programming Example (Write Channel #0)

NOTE

The order in which the registers in the following examples are programmed is not important, except for the Doorbell register, which initiates the DMA transfer. Therefore, to start both a "1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)" and a "1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)" by executing the commands on the table. If, as mentioned in "Starting The DMA Transfer" , you want both channels to start at (approximately) the same time, then do not execute the last command (at 0x280) in the table 'DMA Register Setup' until after you have executed all the commands in table.

NOTE

You must enable the DMA before you start a DMA channel. Therefore you must write ‘1’ to DMA Read Engine Enable or DMA Write Engine Enable register before you write ‘1’ to the DMA Read Doorbell or DMA Write Doorbell register.

11.3.2.2.3.8.1 1 MB Write Transfer On Write Channel #0 Initiated by Local CPU (Non LL mode)

In this example, the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus. The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below. The table below provides the programming details for this example transfer.

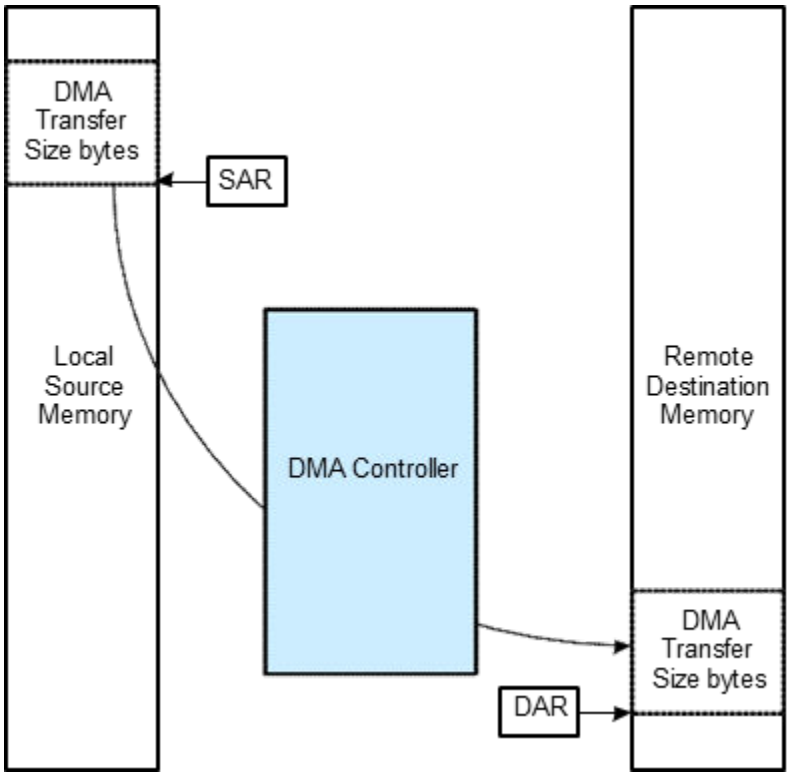


Figure 11-66. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-34. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Address (0x8_0000+)	Name	Value
0x00C	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x054	DMA Write Interrupt Mask	0x0
0x200	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =1 • Remote Interrupt Enable (RIE) =0 • AT, • RO, NS, TC, Function Number =0	0x04000008
0x208	DMA Transfer Size	0x00100000
0x20C	DMA SAR Low	0xBEEF_BEE0
0x210	DMA SAR High	0x0000_0000
0x214	DMA DAR Low	0xCAFE_CAF0
0x218	DMA DAR High	0x0000_0000
0x010	DMA Write Doorbell	0x0

11.3.2.2.3.8.2 1 MB Read Transfer On Read Channel #0 Initiated by Local CPU (Non LL mode)

In this example, the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus. The SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in the figure below. The table below provides the programming details for this example transfer.

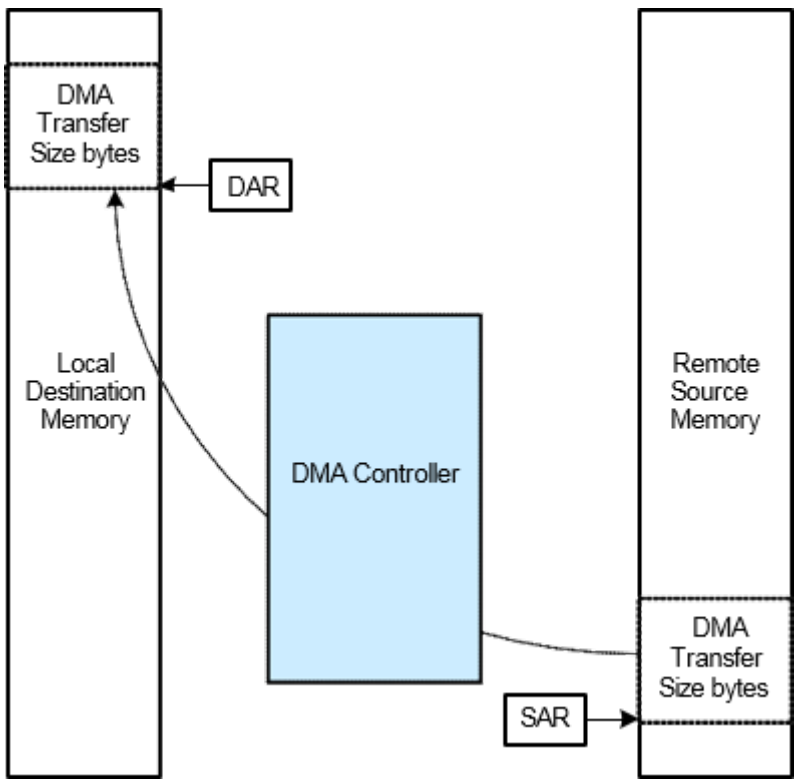


Figure 11-67. 1 MB Read DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-35. DMA Register Setup

Address (0x8_0000+)	Name	Value
0x02C	DMA Read Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Read Engine Enable Register (DMA_READ_ENGINE_EN_OFF).	0x1
0x0A8	DMA Read Interrupt Mask	0x0
0x300	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =1	0x04000008

Table continues on the next page...

Table 11-35. DMA Register Setup (continued)

	• Remote Interrupt Enable (RIE) =0 • AT, • RO, NS, TC, Function Number =0	
0x308	DMA Transfer Size	0x00100000
0x30C	DMA SAR Low	0xBEEF_BEE0
0x310	DMA SAR High	0x0000_0000
0x314	DMA DAR Low	0xCAFE_CAF0
0x318	DMA DAR High	0x0000_0000
0x030	DMA Read Doorbell	0x0

11.3.2.2.3.8.3 1 MB Write Transfer On Write Channel #5 Initiated by Remote CPU (Non LL mode)

In this example, the local interrupt generation is disabled, as the remote CPU initiates the DMA transfer. The remote CPU is interrupted using an IMWr. The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below. The table below provides the programming details for this example transfer.

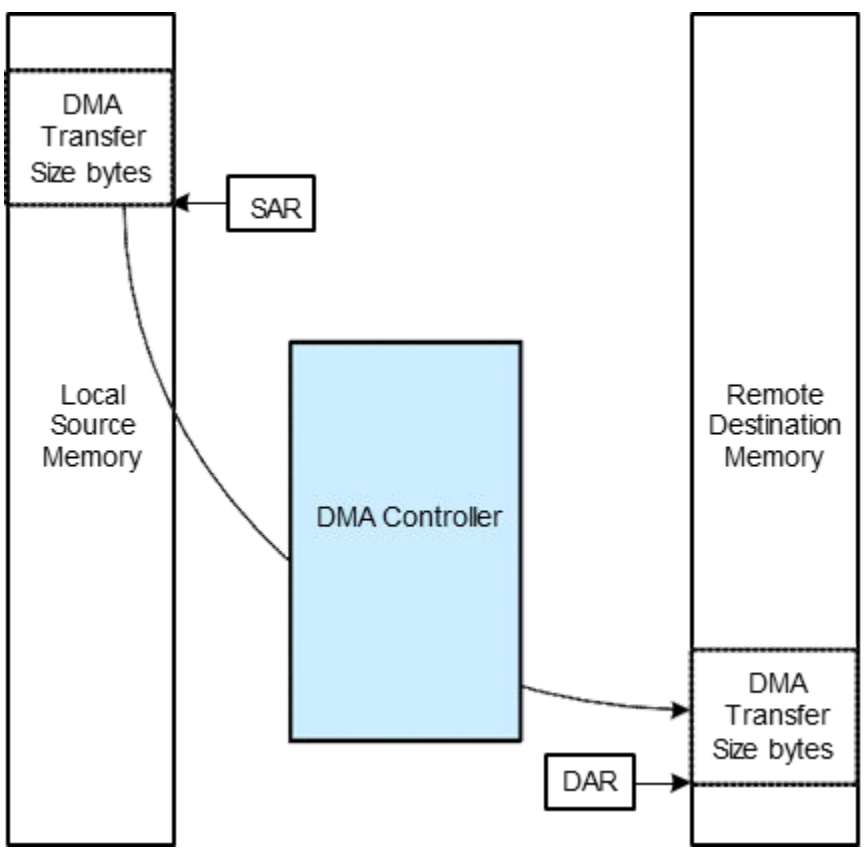


Figure 11-68. 1 MB Write DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-36. DMA Register Setup

Address (0x8_0000+)	Name	Value
0x00C	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x060 / 0x064	DMA Write Done IMWr Address Low and High	your IMWr Address #1
0x068 / 0x06C	DMA Write Abort IMWr Address Low and High	your IMWr Address #2
0x06C	DMA Write Channel 0 IMWr Data	your IMWr Data
0xA00	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =0 • Remote Interrupt Enable (RIE) =1 • AT, • RO, NS, TC, Function Number =0	0x04000010
0xA08	DMA Transfer Size	0x00100000
0xA0C	DMA SAR Low	0xBEEF_BEE0
0xA10	DMA SAR High	0x0000_0000
0xA14	DMA DAR Low	0xCAFE_CAF0
0xA18	DMA DAR High	0x0000_0000

11.3.2.2.3.8.4 1 MB Read Transfer On Read Channel #5 Initiated by Remote CPU (Non LL mode)

In this example, the local interrupt generation is disabled, as the remote CPU initiates the DMA transfer. The remote CPU is interrupted using an IMWr. The SAR is the address of the remote memory, and the DAR is the address of the local memory, as shown in Figure below. The table below provides the programming details for this example transfer.

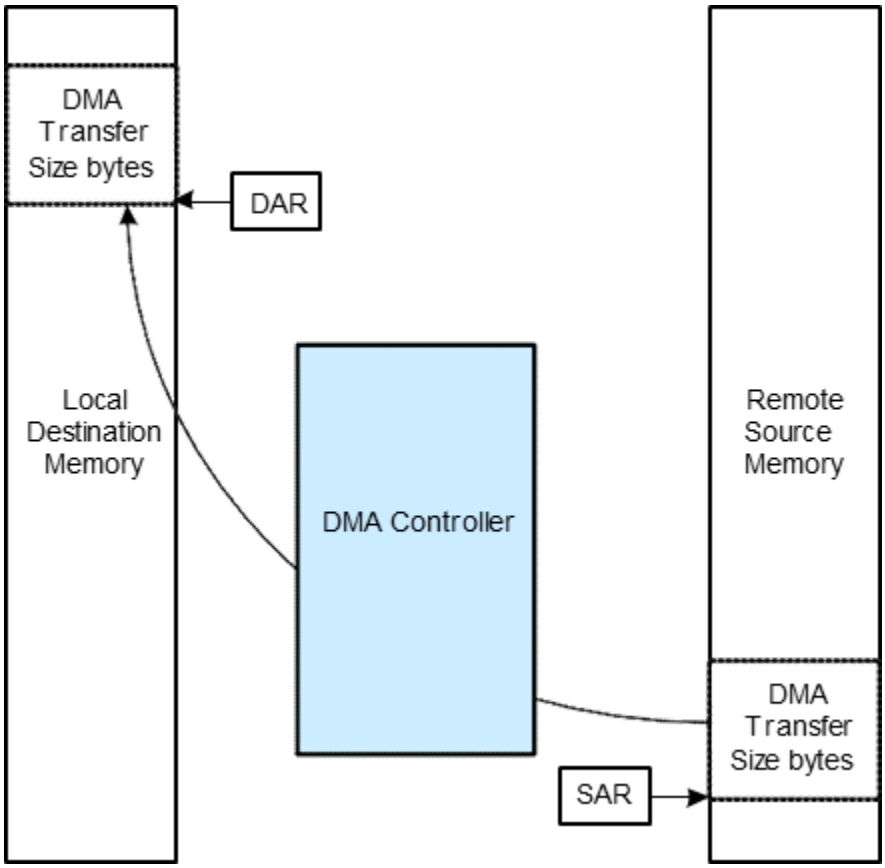


Figure 11-69. 1 MB Read DMA Transfer from 0xBEEF_BEE0 to 0xCAFE_CAF0

Table 11-37. DMA Register Setup

Address (0x700+)	Name	Value
0x29C	DMA Read Engine Enable You must not write “0” to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see “DMA Read Engine Enable Register” (DMA_READ_ENGINE_EN).	0x1
0x33C / 0x340	DMA Read Done IMWr Address Low and high	your IMWr Address #1
0x344 / 0x348	DMA Read Abort IMWr Address Low and High	your IMWr Address #2
0x34C	DMA Read Channel 0 IMWr Data	your IMWr Data
0x36C	DMA Channel Context Index	0x8000_0005
0x370	DMA Channel Control 1 register • Local Interrupt Enable (LIE) =0 • Remote Interrupt Enable (RIE) =1 • AT, RO, NS, TC, Function Number =0 • TD =1	0x04000010
0x378	DMA Transfer Size	0x00100000
0x37C	DMA SAR Low	0xBEEF_BEE0
0x380	DMA SAR High	0x0000_0000
0x384	DMA DAR Low	0xCAFE_CAF0
0x388	DMA DAR High	0x0000_0000

Table continues on the next page...

Table 11-37. DMA Register Setup (continued)

0x2A0	DMA Read Doorbell	0x5

11.3.2.2.3.8.5 Linked List Mode Programming Example (Write Channel #0)

This write channel example corresponds to figure 'Example Producer (Software) - Consumer (DMA) Synchronization Flow', with the following features:

- Recycled linked list (LL) of nine data elements1and one link element.
- Transfer is initiated by local CPU.
- Each data element corresponds to a 1 MB (0x0010_0000) write transfer.
- The SARs for each element are separated by 2 MB (0x0020_0000). The DAR for each element are sepa- rated by 16 MB (0x0100_0000). The SAR is the address of the local memory, and the DAR is the address of the remote memory, as shown in the figure below.
- The LL transfer list is located at address 0x0000_0200 in local memory.
- No remote interrupt is generated so the IMWr generation is disabled, as the local CPU initiates the DMA transfer. The local CPU is interrupted using the edma_int bus.
- There is a Watermark interrupt at element #5, and an Empty interrupt at element #8. Both of these interrupts are done interrupts (as opposed to abort interrupts).

You should familiarize yourself with the Linked List Flow for Producer and Consumer as shown in Figure 'Linked List Flow for Producer and Consumer'. These are the main steps for this process in this example:

Step 1: Create TL in local memory:

- Create the 10 elements as per Table 'Linked List Element Initial Setup'.
  - For elements 0, 1, 2, 3, 4, 6, 7: RIE, LIE, LLP, CB =0,0,0,1
  - For element 5: RIE, LIE, LLP, CB =0,1,0,1 ❑For element 8: RIE, LIE, LLP, CB =0,1,0,1 ❑For element 9: LLP, TCB, CB =1,1,0
- For more details, see “PCS-CCS-CB-TCB Producer-Consumer Synchronization” and “Linked List Mode”

Step 2: Program and Start DMA:

- Configure and start the DMA registers as per Table 'DMA Register Setup'.

Step 4: Wait for done interrupt and recycle TL

- For more details, see “Using Interrupts for Linked List Producer-Consumer Synchronization”
- When recycling elements, consider “Element Recycling”

NOTE

If the DMA driver is running on the host and the interrupt service routine is reading local interrupts to determine if the transfer is successful, then you must set LIE and RIE in the same element and you should mask or ignore the local interrupt pin. Setting RIE and LIE in element A followed by RIE (only) in element B is not a verified usage scenario.

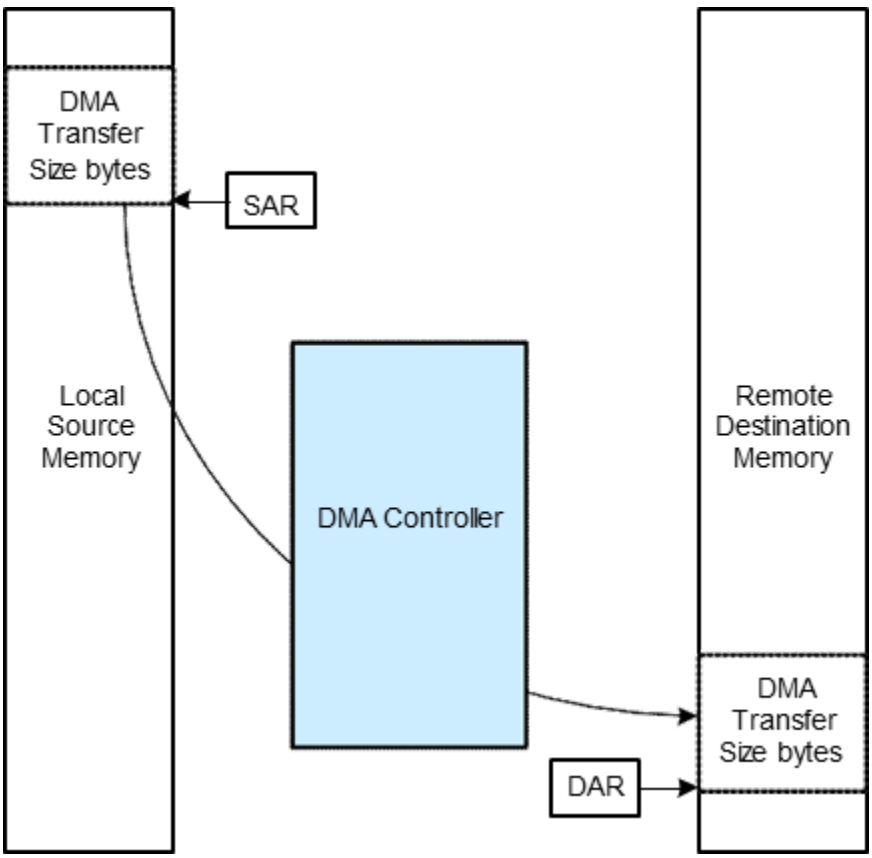


Figure 11-70. 1 MB Write DMA Transfer from SAR to DAR

Table 11-38. Linked List Element Initial Setup

Element #	Address	Data	Label
0	0x0000_0200	0x00000001	Channel Control
0	0x0000_0204	0x00100000	Transfer Size
0	0x0000_0208	0xBEEF_BEE0	SAR Low
0	0x0000_020C	0x0	SAR High
0	0x0000_0210	0xCAFE_CAF0	DAR Low
0	0x0000_0214	0x0	DAR High
1	0x0000_0218	0x00000001	Channel Control
1	0x0000_021C	0x00100000	Transfer Size
1	0x0000_0220	0xBF0F_BEE0	SAR Low

Table continues on the next page...

Table 11-38. Linked List Element Initial Setup (continued)

1	0x0000_0224	0x0	SAR High
1	0x0000_0228	0xCBFE_CAF0	DAR Low
1	0x0000_022C	0x0	DAR High
2	0x0000_0230	0x00000001	Channel Control
2	0x0000_0234	0x00100000	Transfer Size
2	0x0000_0238	0xBF2F_BEE0	SAR Low
2	0x0000_023C	0x0	SAR High
2	0x0000_0240	0xCCFE_CAF0	DAR Low
2	0x0000_0244	0x0	DAR High
3	0x0000_0248	0x00000001	Channel Control
3	0x0000_024C	0x00100000	Transfer Size
3	0x0000_0250	0xBF4F_BEE0	SAR Low
3	0x0000_0254	0x0	SAR High
3	0x0000_0258	0xCDFE_CAF0	DAR Low
3	0x0000_025C	0x0	DAR High
4	0x0000_0260	0x00000001	Channel Control
4	0x0000_0264	0x00100000	Transfer Size
4	0x0000_0268	0xBF6F_BEE0	SAR Low
4	0x0000_026C	0x0	SAR High
4	0x0000_0270	0xCEFE_CAF0	DAR Low
4	0x0000_0274	0x0	DAR High
5	0x0000_0278	0x00000009	Channel Control
5	0x0000_027C	0x00100000	Transfer Size
5	0x0000_0280	0xBF8F_BEE0	SAR Low
5	0x0000_0284	0x0	SAR High
5	0x0000_0288	0xCFFE_CAF0	DAR Low
5	0x0000_028C	0x0	DAR High
6	0x0000_0290	0x00000001	Channel Control
6	0x0000_0294	0x00100000	Transfer Size
6	0x0000_0298	0xBFAF_BEE0	SAR Low
6	0x0000_029C	0x0	SAR High
6	0x0000_0290	0xD0FE_CAF0	DAR Low
6	0x0000_0294	0x0	DAR High
7	0x0000_0298	0x00000001	Channel Control
7	0x0000_029C	0x00100000	Transfer Size
7	0x0000_02A0	0xBFCF_BEE0	SAR Low
7	0x0000_02A4	0x0	SAR High
7	0x0000_02A8	0xD1FE_CAF0	DAR Low
7	0x0000_02AC	0x0	DAR High
8	0x0000_02B0	0x00000009	Channel Control

Table continues on the next page...

Table 11-38. Linked List Element Initial Setup (continued)

8	0x0000_02B4	0x00100000	Transfer Size
8	0x0000_02B8	0xBFEF_BEE0	SAR Low
8	0x0000_02BC	0x0	SAR High
8	0x0000_02C0	0xD2FE_CAF0	DAR Low
8	0x0000_02C4	0x0	DAR High
9	0x0000_02C8	0x00000006	Channel Control
9	0x0000_02CC	0x0	Reserved
9	0x0000_02D0	0x0000_0200	Linked List Element Pointer Low
9	0x0000_02D4	0x0	Linked List Element Pointer High

Table 11-39. DMA Register Setup

Address **(0x8_0000+)**	**Name**	**Value**
0x030	DMA Write Engine Enable You must not write 0 to this register. Even temporarily writing 0 to this register resets the DMA logic. For more details, see DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF).	0x1
0x054	DMA Write Interrupt Mask	0x0
0x090	DMA Write Linked List Error Enable • LLLAIE =1	0x1
0x200	DMA Channel Control 1 register • Linked List Enable (LLE) =1 • Consumer Cycle Status (CCS) =PCS =1 • AT, • RO, NS, TC, Function Number =0	0x04000300
0x21C	DMA Linked List Pointer Low	0x0000_0200
0x220	DMA Linked List Pointer High	0x0000_0000
0x010	DMA Write Doorbell	0x0

11.3.2.2.4 Advanced DMA Information and Operation

This section discusses advanced features and operation of the DMA.

11.3.2.2.4.1 Overview

The DMA write and read channels operate independently to maximize the performance of the DMA read and write data transfers over the PCIe link.

The DMA can simultaneously perform the following types of memory transactions:

- DMA write: transfer (copy) of a block of data from local memory to remote memory
- DMA read: transfer (copy) of a block of data from remote memory to local memory

After you have programmed and started a DMA transfer (see "Using the DMA"), the DMA transfers the data as described in "**DMA Write Transfer"** and "**DMA Read Transfer"**

11.3.2.2.4.2 DMA Ordering

The DMA assumes that all DMA read channel data must be written to the local application memory in strict address order. For a DMA read transfer, each original outgoing request has a unique PCIe tag. The remote link partner or intermediate switch does not have to preserve the order of CPLs corresponding to such individual requests. When the DMA receives the CPLs out of order, it reorders that data in the "Read Buffer" as described in "**DMA Read Transfer"** . DMA write transfer data is sourced from the local application memory and sent to the remote memory in address order.

11.3.2.2.4.3 Error Handling

The DMA supports full AXI and PCIe error handling in conjunction with the PCIe controller. The tables below list the possible sources of an error for DMA read and write transfers.

Table 11-40. Possible Sources of an Error during a DMA Read Transfer

Source	Type	Description
Application Write Error Detected	Fatal	The DMA read channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while writing posted data to it. This error is fatal. • You must restart the transfer from the beginning because the channel context is corrupted.

Table continues on the next page...

Table 11-40. Possible Sources of an Error during a DMA Read Transfer (continued)

		• The MWr is not rolled back. • The AXI bridge master does not report this error to software through error logging or MSG. **NOTE:** During linked list mode, the Application Write Error only corrupts the current element in the linked list. This is because the DMA does not fetch the next element from local memory until it has completely finished transferring the current block of data (as defined by the current linked list element). Finished implies that your application has successfully acknowledged (without ERROR) all of the MWr requests.
Unsupported Request (UR)	Non-fatal	The DMA read channel has received a PCIe UR CPL status from the remote device in response to the MRd request.
Completer Abort (CA)	Non-fatal	The DMA read channel has received a PCIe CA CPL status from the remote device in response to the MRd request.
CPL Time Out	Non-fatal	The DMA read channel has timed-out while waiting for the remote device to respond to the MRd request, or a malformed CplD has been received.
Data Poisoning	Fatal	The DMA read channel has detected data poisoning in the CPL from the remote device in response to the MRd request. The DMA read channel will drop the completion and then be halted.
Linked List Element Fetch Error Detected	Fatal	The DMA read channel has received an error response from the AXI bus (or TRGT1 interface when the AXI bridge is not used) while reading a linked list element from local memory.

Table 11-41. Possible Sources of an Error during a DMA Write Transfer

Source	Type	Description
Application Read Error Detected	Fatal	The DMA write channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while reading data from it. • The response TLP is discarded by the DMA and is not converted to an outbound MWr TLP.

Table continues on the next page...

Table 11-41. Possible Sources of an Error during a DMA Write Transfer (continued)

		• The UR or CA CPL status TLP generated by the bridge is discarded by the DMA. • The AXI bridge master does not report this error to software through error logging or MSG.
Linked List Element Fetch Error Detected	Fatal	The DMA write channel has received an error response from the AXI bus at the bridge master interface (or TRGT1 interface when the AXI bridge is not used) while reading a linked list element from local memory. • The UR or CA CPL status TLP generated by the bridge is discarded by the DMA. • The AXI bridge master1 does not report this (or for non-DMA traffic) error to software through error logging or MSG.

When the DMA detects an error, it performs the following actions (for the channel with the error):

- Stops issuing new requests to the remote link partner.
- Stops issuing new requests.
  - All data in the “Read Buffer”, that was received before the error was detected, is valid. In the case of a non-fatal error, this is reflected in the DMA Transfer Size Register DMA_TRANSFER_SIZE_OFF_WRCH_0.
  - All data in the read buffer (for that channel), that was received after the error was detected, is discarded.
- Places the DMA channel in a halted state (channel status field of the DMA Channel Control 1 Register is halted).
- Waits for all outstanding requests and CPLs to complete.
- Generates the abort interrupt or IMWr, if enabled, as outlined in “Interrupt Handling”.

For a DMA read transfer, you can read the DMA Read Error Status High Register (DMA_READ_ERR_ STATUS_HIGH_OFF) and DMA Read Error Status Low Register (DMA_READ_ERR_STATUS_LOW_OFF) to identify the source of the error. For a DMA write transfer (in linked list mode only), you can read the DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF) to identify the source of the error. After your software has finished error and interrupt routine handling for Non-Fatal errors, you can request the DMA to continue processing (for the channel with the error) by:

1. Reprogramming the SAR based on the number of bytes remaining in the DMA Transfer Size register
2. Writing the channel number to the Doorbell Number field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF) or DMA Read Doorbell Register (DMA_READ_DOORBELL_REG).

NOTE

You cannot continue after a Fatal Error is detected, as the channel context is corrupted. Any actions that were in progress are not rolled back. To exit from this condition, you need to Soft Reset the complete DMA controller, as described in DMA Write Engine Enable Register (DMA_WRITE_ENGINE_EN_OFF) and DMA Read Engine Enable Register (DMA_READ_ENGINE_EN_OFF).

When an error occurs in linked list mode, the next LL element is not fetched from local memory. DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) is not incremented by the DMA, but remains pointing to the element that caused the error.

11.3.2.2.4.3.1 Error Handling Assistance by Remote Software

During a DMA write transfer, the DMA is not aware of MWr errors that are detected in the remote device, because an MWr is a posted request and has no corresponding completion TLP. In these cases, your remote CPU must manually abort the DMA transfer by writing the channel number to the Doorbell Number field, and 1 to the Stop field of the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF). This causes the DMA to:

- Place the channel in a stopped state. The channel Status field of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) is stopped and the DMA Transfer Size register is not 0x0.
- Wait for all outstanding pending transactions.
- Assert the abort interrupt if it is enabled

11.3.2.2.4.4 Linked List Operation

This section describes the detailed operation of the DMA in linked list (LL) mode. It also discusses how the software should produce LL elements1, and how the DMA and software maintain synchronization.

11.3.2.2.4.4.1 LL Operation Overview

You should first read "Linked List Mode" for an overview of LL mode. In this section, a normal LL operation called recycling2 is described in detail. The process is described for a write channel, but it is possible to have duplicate processes running in parallel for other write or read channels. In this process interrupts are used to trigger the producer (software) to recycle elements. Typically a Watermark interrupt (positioned near the middle of the transfer list (TL)) and an Empty interrupt (in the last data element) are used by setting LIE in these two elements to '1'.

The steps in this process are:

1. Software creates a LL element structure called a TL in local memory consisting of N-1 data elements and one link element.

   The link elements "LL Element Pointer Low/High DWORDs"(as shown in Linked List Element figure in Linked List Mode topic) are programmed to point back to the beginning of the first data element.

   One of the data elements (near the middle of the TL) is programmed to generate a "Watermark" interrupt.

   The last data element is programmed to generate an "Empty" interrupt.

2. Software programs the DMA with the location of the TL. This is done by writing to the DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LLP_HIGH_OFF_WRCH_0).
3. Software starts the DMA process by ringing the write channel Doorbell. This is done by writing to the DMA Write Doorbell Register (DMA_WRITE_DOORBELL_OFF).

   The solid purple loop L1 in Figure 11-71 corresponds to steps 1-3.

4. The DMA reads (**consumes**) each data element from local memory, and if the CB and CCS bits match, loads the information (SAR, DAR, size, and so on) from that data element into the channel context registers. These context registers determine the operation of the channel that the DMA is currently servicing. The DMA executes the check for the CB/CCS match after it has evaluated TCB and toggled CCS. The DMA always loads link elements into the channel context.

NOTE

If the DMA stops on a link element, then your software must set the DMA Linked List Pointer registers, and restart the DMA process by ringing the doorbell.

With recycling, the same LL structure is used over and over again. Another mode of operation would be to jump to a new LL element structure when the current one is consumed.

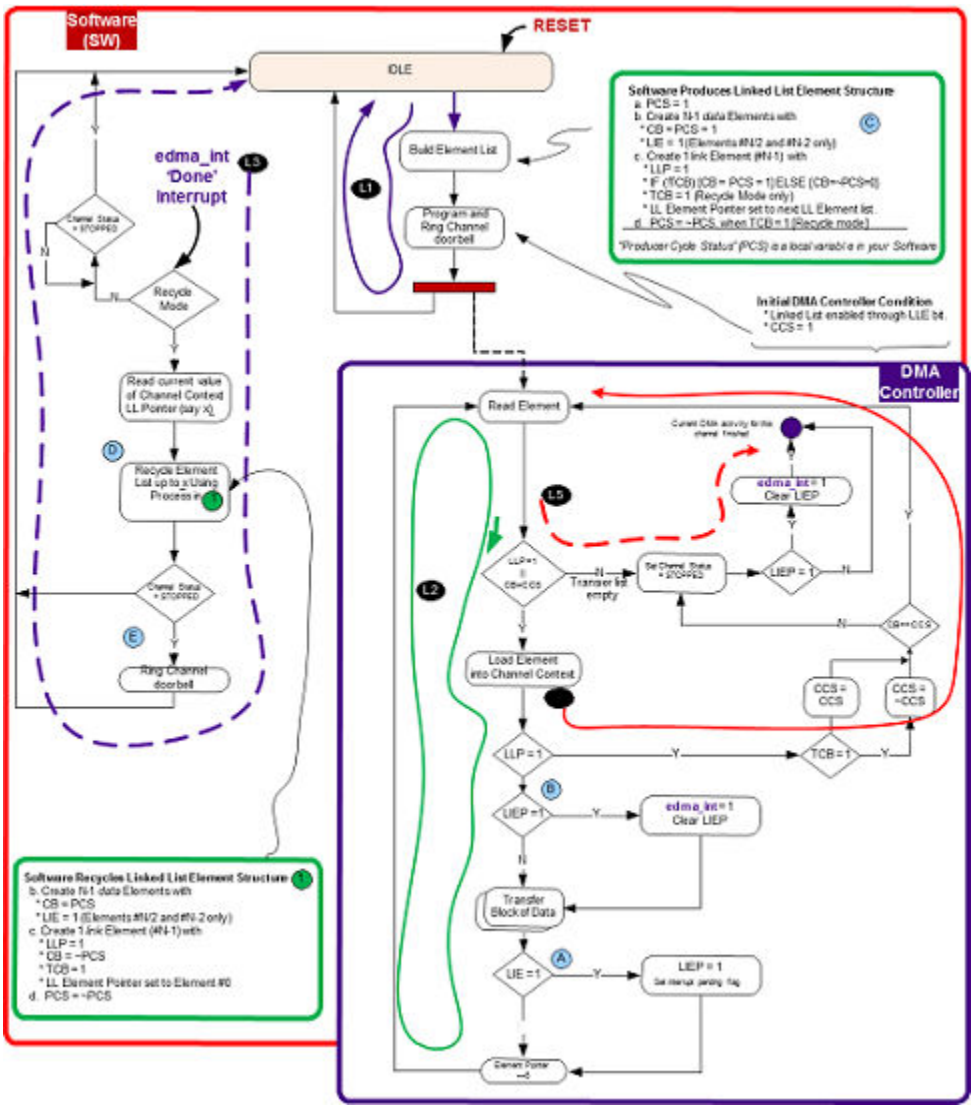


Figure 11-71. Linked List Flow for Producer and Consumer

5. The DMA then proceeds to transfer the block of data (as defined by the element), and when it is finished, reads the next element from local memory.

   The solid green loop L2 in Figure 11-71 corresponds to steps 4-5.

6. The last element in the list (called a link element, and indicated by having its LLP field set to 1) is not used to transfer a block of data.

   It causes the DMA to effectively repeat the task in step 4.

Its TCB is typically set to 1. This causes the DMA to toggle its CCS bit, and the software to toggle its PCS bit. The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mechanism to ensure that:

Software does not recycle elements that have not yet being consumed by the DMA

DMA correctly recognizes and consumes recycled elements. The solid red loop L4 in Figure 11-71 corresponds to step 6.

7. Upon reception of the interrupt (mentioned in step 1), the software starts to recycle the TL.

   Software reprograms each data element with new DMA transfer information.

   The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mecha- nism described later in this section.

   The dashed purple loop L3 in Figure 11-71 corresponds to step 7.

8. As some point, the software wants to terminate the complete DMA process, by not recycling any more elements. The DMA recognizes this condition when CB !=CCS, and sets the channel status to stopped

   The dashed red loop L5 in Figure 11-71 corresponds to step 8.

11.3.2.2.4.4.2 Using Interrupts for Linked List Producer-Consumer Synchronization

When the DMA is finished with an element, the DMA checks the LIE bit, before reading the next LL element. For more details, see circled step A in Figure 11-71. When the LIE is set, then the DMA does not assert the "done" interrupt immediately, but sets an internal interrupt pending flag (LIEP). It asserts the actual done interrupt after the next data element has been read from system memory. For more details, see circled step B in Figure 11-71. This automatic internal process of delaying the interrupt avoids a race condition between the DMA and software when the LIE bit is set in an element.

Your software should set the "Watermark" interrupt early to schedule the recycling of the consumed elements. Placing the Watermark interrupt too far down in the list, combined with a slow element recycling process in your application; ensures that the DMA returns to the start of the LL before your software has recycled it. The DMA channel STOPs, and you have to restart it by writing to its Doorbell. Typically a "Watermark interrupt" (positioned near the middle of the TL) and an Empty interrupt (in the last data element) are used by setting LIE in these two elements to 1b1. Upon reception of the interrupt, the software starts to recycle the TL.

Operational interrupts are always "done" interrupts. For more details, see "Interrupts and Error Handling"

NOTE

The DMA processes the remote interrupt enable (RIE) bit in the LL element in the same way as the LIE. Therefore, everywhere that LIE and edma_int are mentioned in this section can be interpreted to mean LIE and edma_int, or RIE and IMWr.

11.3.2.2.4.4.3 PCS-CCS-CB-TCB Producer-Consumer Synchronization

The software and DMA use the "PCS-CCS-CB-TCB" producer-consumer synchronization mechanism to ensure that software does not recycle elements that have not yet being consumed by the DMA, and that the DMA correctly recognizes and consumes recycled elements. This process, which is shared between the DMA and the software, is illustrated through an example producer-consumer flow in the figure below.

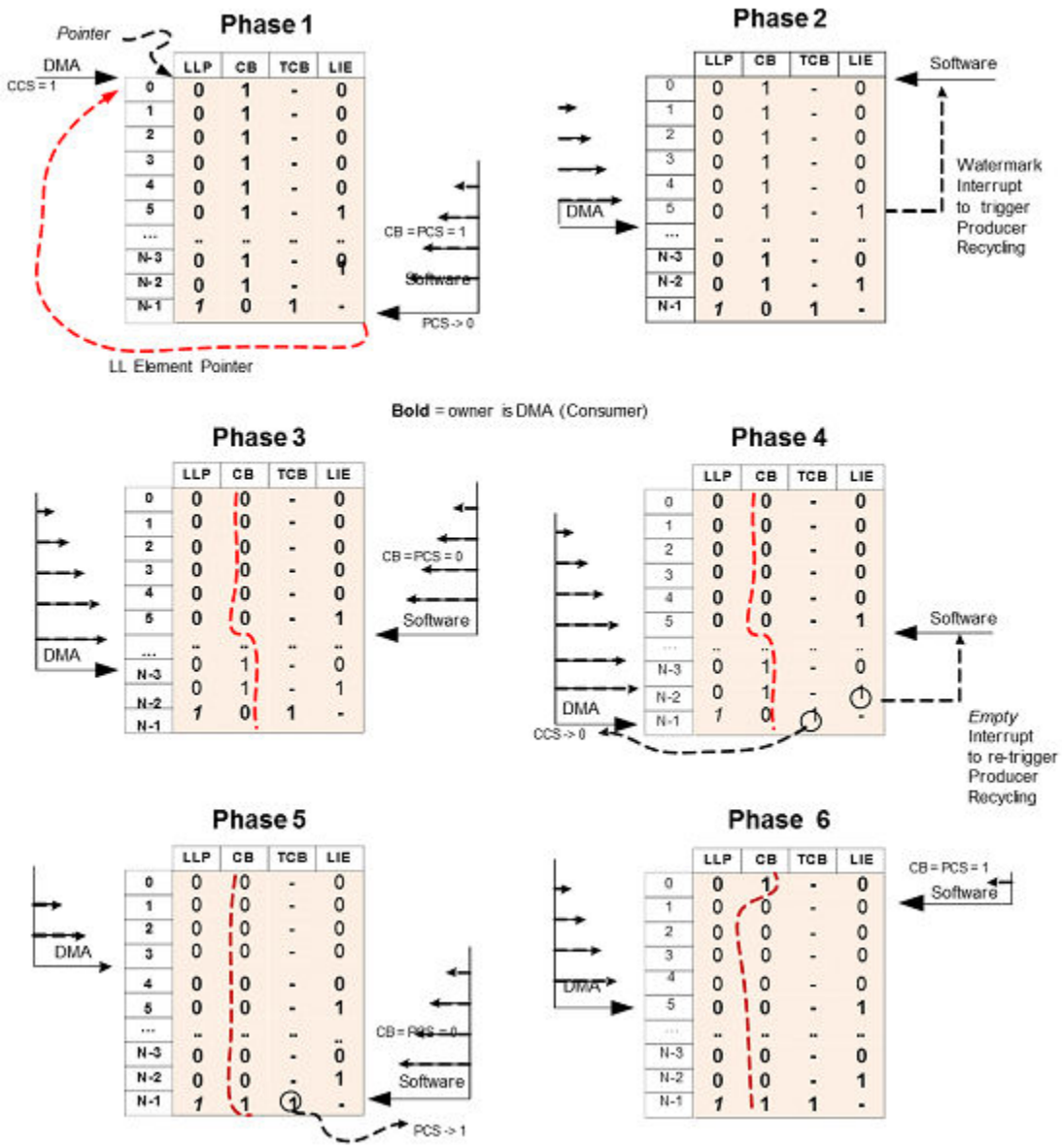


Figure 11-72. Example Producer (Software) - Consumer (DMA) Synchronization Flow

Before looking at the example in detail, it is useful to note that the DMA performs the following two tests as part of this process:

- Consumer-Owned Element, or Transfer List (TL) Empty Test

  The solid red loop L4 in the Figure 11-71 corresponds to this test. The Cycle Bit (CB) of DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) (which was loaded from the LL element) is tested against the DMA Consumer Cycle State (CCS) bit of the same register. When CB =CCS, the element is owned by the consumer (DMA) and the data transfer

is executed. When CB !=CCS, the TL is empty and the DMA sets channel Status (CS) in DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) to "Stopped".

- Toggle Cycle Bit (TCB) Test

  The Dashed Red Loop Marked L5 in Figure 11-71 corresponds to this test. When the Toggle Cycle Bit (TCB) of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) (which was loaded from the LL element) is set to 1, the DMA toggles the CCS bit of the DMA Channel Control 1 Register (DMA_CH_CONTROL1_OFF_WRCH_0) to be ~CB. The producer (software) also performs a similar action on its local variable "Producer Cycle Status" (PCS), as it sets PCS =~PCS. The TCB field is only valid in the last LL element (the link element). Toggling the PCS and CCS flags in this manner synchronizes the consumer to the producer.

The example in Figure 11-72 has the following six phases:

1. The software builds the transfer list for the write (or read) channel, according to the steps outlined in the circled step C in Figure 11-71. For each element, CB =PCS =CCS =1. The exception to this is the link element where CB is set to the next iteration value because the link element is always loaded into the channel context.
   - The link element points to the start of the TL and its TCB bit is 1. Software toggles its PCS variable.
   - Software writes the location of the TL into the channel context LL Pointer register.
   - Software rings the write channel Doorbell.
2. The DMA starts processing elements. After processing element 5 (and fetching the next element), the DMA asserts a Watermark “done” interrupt to signal the software to start recycling elements.
3. The software is recycling elements. It reads the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LL- P_HIGH_OFF_WRCH_0), and recycles elements up to this location1in the TL. It sets CB for each element to be PCS. The Dashed Purple Loop Marked L3 in Figure 11-71 corresponds to this activity. The DMA continues processing (consuming) elements as before.
4. The DMA then asserts an Empty “done” interrupt (from the LIE bit in the last data element at the end of TL that was set in the previous phase) to indicate to the software, to start recycling elements again. The DMA reaches the end of the TL, and as the TCB is set to 1, it toggles its CCS bit to 0 (~CB). For more details, see circled step D in Figure 11-71.
5. The software reads the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_REG_WRCH_0) and DMA Linked List Pointer High Register

(DMA_LLP_HIGH_OFF_WRCH_0), and recycles elements up to this location in the TL. It sets CB for each element to be PCS. As it recycles the link element, it detects that TCB is set to 1. It toggles its internal variable PCS to 0, and the initial TL has been fully recycled. The DMA starts processing the recycled elements.

6. The DMA continues processing (consuming) elements as before. Software did not have to ring the Doorbell, as the first element was already recycled before the DMA started processing it. The software might continue recycling from the top of the list, if the DMA had already returned to the top of the list when the software read the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LL-P_HIGH_OFF_WRCH_0), in step 5. If not, it starts recycling again when the DMA processes element 4 and the DMA asserts a Watermark “done” interrupt.

11.3.2.2.4.4.4 Element Recycling

When your software is recycling elements, it should program the first DWORD (control bits) in the Figure 11-63 only after the other DWORDs have been programmed. In addition, it should set the CB bit (to ~PCS) within this DWORD only after the other bits in this DWORD have been set. This programming sequence avoids the DMA incorrectly fetching an element that software is currently recycling.

When the software is finished recycling it must check the channel context status (CS). When CS is "Stopped", the software must ring the channel Doorbell again. For more details, see circled step E in Figure 11-59. If your element recycling process was fast enough, then the DMA will seamlessly move from processing the last old element to the first new element, without the need for the software to ring the channel Doorbell again. To eliminate the possibility of a race condition between these two actions (read and write), you should first confirm the presence of the "done" interrupt bit in the DMA Write Interrupt Status Register (DMA_WRITE_INT_STATUS_OFF) or DMA Read Interrupt Status Register (DMA_READ_INT_ STATUS_OFF) and also check the DMA Linked List Pointer Low Register (DMA_LLP_LOW_REG_WRCH_0) to confirm that it is pointing to the correct location.

11.3.2.2.4.4.5 Link Down Recovery

During normal operation when the link goes down, the controller generates a "link reset" request. A

high-to-low transition on link_req_rst_not indicates that the PCIe controller is requesting external logic to reset the PCIe controller because the PHY link is down. For more details, see "Hot Reset". To determine the progress of the linked list operation when the

link went down, you can read the contents of the channel context DMA Linked List Pointer Low Register (DMA_LLP_LOW_OFF_WRCH_0) and DMA Linked List Pointer High Register (DMA_LLP_HIGH_OFF_WRCH_0).

11.3.2.3 M-PCIe

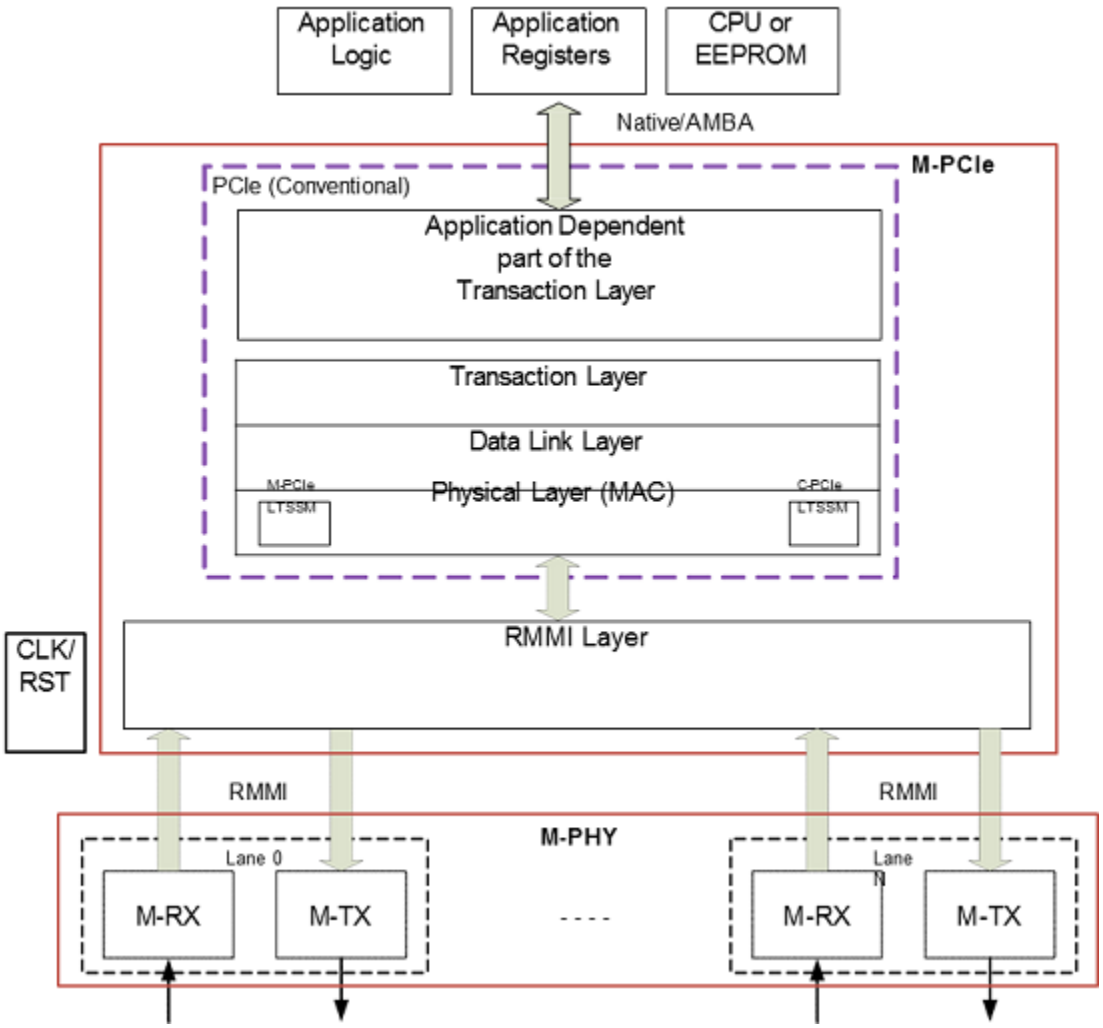
This section describes the optional M-PCIe feature.

11.3.2.3.1 Feature Description

The new M-PCIe specification enables the PCIe controller to operate over the low-power MIPI M-PHY physical layer technology, extending the benefits of PCIe to mobile devices including thin laptops, tablets and smartphones. The M-PCIe controller is compliant with the M-PCIe ECN1 specification. The interface between the controller and the PHY is compliant with the RMMI (Reference M-PHY MODULE Interface) specified in Annex A of the M-PHY Specification.

A complete M-PCIe port solution includes the controller, an analog PHY macro, and application logic to source and sink data. The physical layer is split across the RMMI such that the MAC functionality (LTSSM, lane-to-lane deskew) is in the controller and the PHY functionality is implemented in the RMMI-compliant PHY. The PHY module resides outside of the controller, interfacing through the standard RMMI. For more details, see Integrating with the PHY in the User Guide.

Figure 11-1 Complete M-PCIe Port Solution



a. HS=high-speed. LS=low-speed.

11.3.2.3.1.1 Features

This release supports the following features: **Table 11-1M- PCIe Features**

M-PCIe Feature	Supported
HSa-Gear Link Speeds	HS-Gear1 HS-Gear2 HS-Gear3
HS-Gear1 Base RATE-A Frequencies	31.2 MHz 62.4 MHz 124.8 MHz
Elastic Buffer Maximum Frequency Difference Compensation Between Link Partners • HS-Mode: • LS-Mode:	+/- 2000 ppm +/- 300 %

Table continues on the next page...

Dynamic Frequency Gear Speed Changing Mode	✓
Dynamic Width Gear Speed Changing Mode (RMMI Standard does not Support this)	✓
Controller and PHY can have Different Gear Speed Changing Modes	✓
x1, x2, x4, x8, x16 Link Width Support	✓
10-bit RMMI Width	✓
20-bit RMMI Width	✓
40-bit RMMI Width	✓
Controller and PHY can have Different RMMI Widths	✓
Runtime Selectable Mode of Operation Between M-PCIe and Conventional PCIe (Selectable PHY)	✓
Automatic RRAP Function (Off-loads Attribute Discovery/ Configuration From Your CPU In Downstream Port)	✓
LS-Gear Link Speeds Note: All bytes on the RMMI are valid in LS mode.	LS-Gear1
RATE-A / RATE-B Series Support	✓
Dynamic Asymmetric Link Width Adjustment	✓
Static (Compile Time) Asymmetric Link Width Adjustment	✗
Dynamic Link Reconfiguration Mechanism (Up Configure)	✓
Digital Remote Loopback	✓
Digital Local (RMMI) Loopback	✓

M-PCIe Feature	**Supported**
M-PHY Interoperability Support	✓
Third Party M-PHY Interoperability Support Through Custom PHY Flow	✓
Hardware Compliance Pattern Generator	✓
Internal Databus Width	32, 64, 128
Optional M-PCIe Features	
Manual Lane Flip	✓
Manual Polarity Change	✓
WAKE# Sideband Signal Support	✓
Optional System Features	
Advanced Power Management, Power Gating, and UPF Support	✓

Table continues on the next page...

Multifunction Support	✓
DMA Support	✓
ARI Support	✓
SR-IOV Support	✓
Latency Tolerance Reporting (LTR) Support	✓
OBFF Support	✓
Address Translation Services (ATS) Support	✓
Atomic Ops Support	✓
TLP Processing Hints (TPH) Support	✓
TLP Prefix Support (Including PASID)	✓
ID Based Ordering (IDO) Support	✓
Conventional PCIe Features Not Applicable in M-PCIe Mode	
CLKREQ# Sideband Signal and L1 Clock Power Management (CPM) Support	✗
Separate Refclk with Independent Spread Spectrum Clocking (SRIS)	✗
L1 Substates Support	✗
Dynamic Lane Reversal	✗
Dynamic Polarity Change	✗

a. HS=high-speed. LS=low-speed.

11.3.2.3.1.2 Limitations

Table 11-2 identifies the temporary limitations when using the M-PCIe module.

Table 11-2 M-PCIe Limitations

Limitation	Note
Limited Analog Loopback Support	Analog loopback requires the M-PHY to autonomously loop and re-transmit the Rx data. However, the RMMI specification does not define the entry mechanism for analog loopback. • If you are using the M-PHY (which has no input to accept the cores pa_rx_analog_loopback output), the analog loopback path must

Table continues on the next page...

	pass through the controller involving the elastic buffer. The Tx lanes in the controller use the Tx (and not the Rx) clock. The elastic buffer cannot operate indefinitely under this condition. If there is a frequency difference between the Rx and Tx clocks, there will be some data overflow or underflow. • If you are using a third party M-PHY, the M-PHY enters analog loop- back mode when the controller asserts pa_rx_analog_loopback. If the M-PHY does not support such an input, you can instruct the M- PCIe controller to do loopback from Rx to Tx through the elastic buffer.
Verilog Testbench (VTB) VIP not Included	M-PCIe configurations currently have no M-PCIe VIP and connect instead to an RTL DUT of the opposite port type.
No Midstream De-assertion of TX_PhyDIRDY	The controller does not support midstream de-assertion of phy_mac_tx_phydirdy during burst mode. This is not an M-PCIe or RMMI requirement. However, the tx_symbolclk is derived from the same clock that the M-PHY TX path uses, and there is no reason for the M-PHY to insert wait cycles.
Midstream De-assertion of RX_PhyDORDY	The rx_symbolclk is the recovered clock and there is no elastic buffer in the M-PHY. Therefore, there is no reason for the M-PHY to insert de-assertion of mac_phy_rx_phydordy during burst mode. If your M-PHY de-asserts mac_phy_rx_phydordy during burst mode, there is a risk of underflowing the elastic buffer in the controller.
TLP in 2K+ PPM System with Payload Greater than 2 KB not Supported.	The controller can handle a clock difference of 2K PPM only. When clock compensation greater than 2K PPM is required, the elastic buffer overflows or underflows on receiving a TLP with payload greater than 2 KB.
M-PHY RX Registers Cannot be set when LTSSM Exits L1 or L2.	The controller cannot set M-PHY RX registers when: • LTSSM exits L1 in a downstream or upstream port • LTSSM exits L2 in a downstream port.

11.3.2.3.1.3 Frequency, Speed, and Width Support

To set the M-PCIe frequency, datapath width, and speed modes of the controller, use the following configuration parameters in the coreConsultant GUI:

■ Controller Base Frequency (CM_FREQ)

■ PHY Base Frequency (CM_PHY_FREQ)

■ Controller Gear2 and Gear3 Modes (CM_GEAR2_MODE and CM_GEAR3_MODE)

■ PHY Gear2 and Gear3 Modes (CM_PHY_GEAR2_MODE and CM_PHY_GEAR3_MODE)

■ Maximum Tx and Rx Link Width (CM_TXNL_GUI and CM_RXNL_GUI)

❑CX_NL = max {CM_TXNL_GUI, CM_RXNL_GUI}

❑ CX_NB = (CM_FREQ == 124.8) ? 1 : (CM_FREQ == 62.4) ? 2 : 4

■CM_FREQ_STEP_DOWN_EN. This is the enable for the “Frequency Step Module”

11.3.2.3.1.4 Gear2/Gear3 Modes

For Dynamic Frequency (DF) configurations, the number of active symbols on the RMMI is constant; the frequency of the controller doubles each time as the controller transitions from Gear1 -> Gear2 -> Gear3 rates. For Dynamic Width (DW) configurations, the frequency of the controller is constant; the number of active symbols on the RMMI doubles each time as the controller transitions from Gear1 -> Gear2 -> Gear3 rates. The controller cannot have different Gear2 and Gear3 modes. For example, you cannot configure the controller for Dynamic Width in Gear2 mode, and at the same time configure it for Dynamic Frequency in Gear3 mode. For more details, see Figure 11-3.

Figure 11-3 Gear2/Gear3 Speed Changing Paths

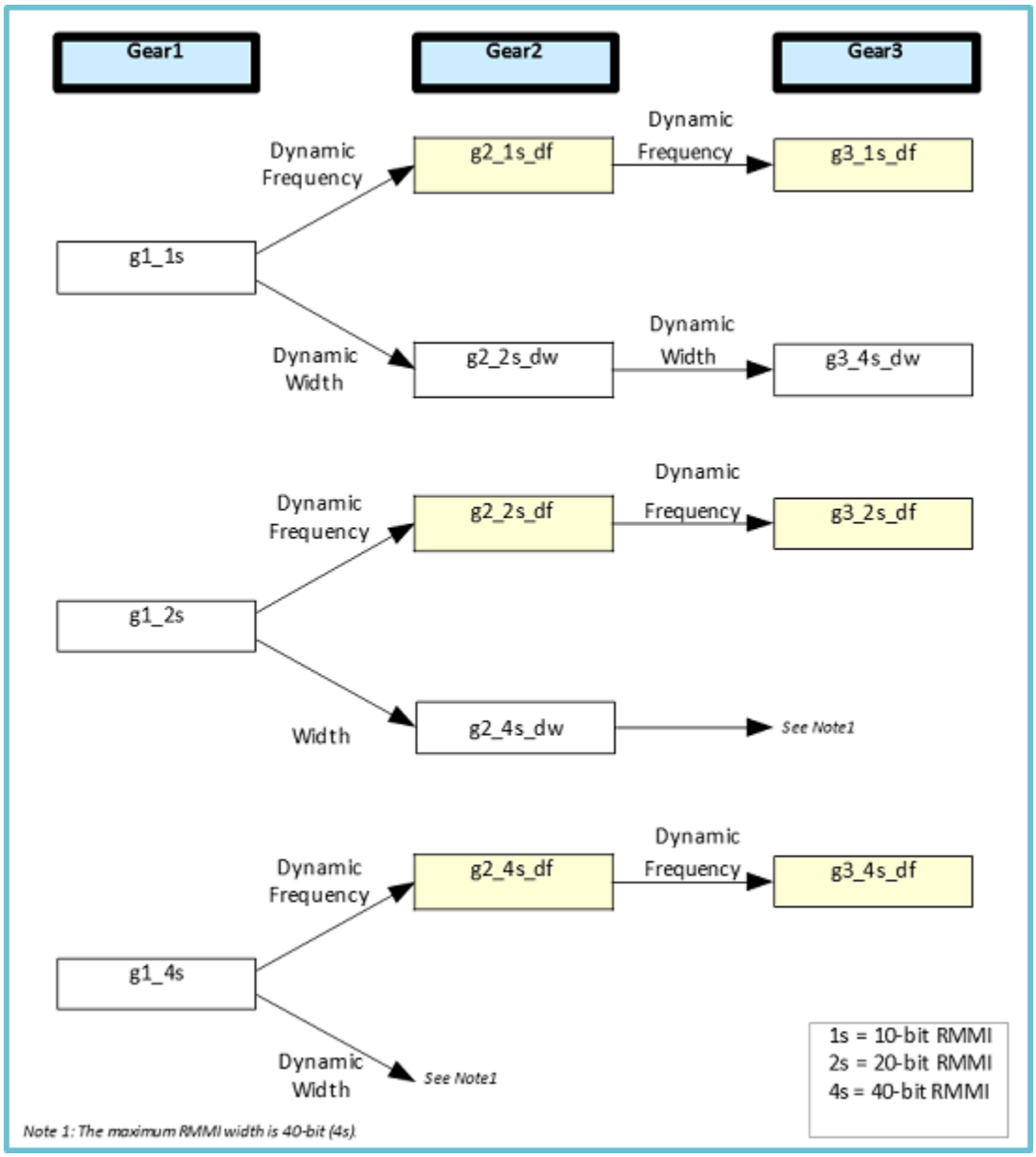


Table 11-3 Supported Controller Configurations in Standard M-PCIe Mode

				Maxa Link Width							
Gear1 RATE A Frequencyb (CM_FREQ)	**Gear2 Mode**	**Gear3 Mode**	**RMMI Widthc**	**For 32-bit datapath**	**For 64-bit datapathd**	**For 128-bit datapath**	**For 256-bit datapath**	**Maximum core_clk Frequency (x CM_FREQ)**	**Gear1 Speed Modee**	**Gear2 Speed Mode**	**Gear3 Speed Mode**
124.8	Disable	Disable	10-bit	x4	x8	x16	-	x1	g1_1s		
62.4			20-bit	x2	x4	x8	x16	x1	g1_2s		

Table continues on the next page...

31.2			40-bit	x1	x2	x4	x8	x1	g1_4s		
124.8	DW	Disable	20-bit	x2	x4	x8	x16	x1	g1_1s	g2_2s_dw	
62.4	DW		40-bit	x1	x2	x4	x8	x1	g1_2s	g2_4s_dw	
124.8	DF		10-bit	x4	x8	x16	-	x2	g1_1s	g2_1s_df	
62.4	DF		20-bit	x2	x4	x8	x16	x2	g1_2s	g2_2s_df	
31.2	DF		40-bit	x1	x2	x4	x8	x2	g1_4s	g2_4s_df	
124.8	DW	DW	40-bit	x1	x2	x4	x8	x1	g1_1s	g2_2s_dw	g3_4s_dw
124.8	DF	DF	10-bit	x4	x8	x16	-	x4	g1_1s	g2_1s_df	g3_1s_df
62.4	DF	DF	20-bit	x2	x4	x8	x16	x4	g1_2s	g2_2s_df	g3_2s_df
31.2	DF	DF	40-bit	x1	x2	x4	x8	x4	g1_4s	g2_4s_df	g3_4s_df

a. The value in each cell indicates the maximum link width supported. You can configure the controller with a link width up to this value, and coreConsul- tant automatically calculates the datapath width.

b. Corresponding RATE B frequencies are 145.76, 72.88, and 36.44 MHz.

c. The actual physical RMMI interface I/O width is 10*CM_NB bits per lane. The coreConsultant tool automatically calculates this width from CM_GEAR2_MODE, CM_GEAR3_MODE, and CM_FREQ. In a DW configuration, the number of active bits in the RMMI interface increases as the controller transitions to a higher Gear speed.

d. A 64-bit datapath requires a 32-bit/64-bit license. You cannot create a 32-bit or 64-bit configuration with a 128-bit/256-bit license

e. The ‘g’ value indicates the speed mode. The ‘s’ value indicates the number of 10-bit symbols processed per clock cycle per lane by the controller RMMI I/F, in the indicated speed mode. The ‘dw/df’ value indicates how the speed change to that mode is achieved

11.3.2.3.1.5 Controller PHY Compatibility

The following tables indicate which combinations of controller configurations from Table 11-3 and PHYs are supported. In Table 11-4, the triplet of values for each row indicates how the controller moves from Gear1 to Gear2 speed mode and from Gear2 to Gear3 speed mode. The triplet of values also identifies the particular controller configuration in Table 11-3. The controller always uses the same mechanism (DF or DW) to go from Gear1 to Gear2 and from Gear2 to Gear3.

Table 11-4 Gear3 Controller-PHY Combinations Supported

		PHY			
		g1_1s g2_2s_dw g3_4s_dw	**g1_1s g2_1s_df g3_1s_df**	**g1_2s g2_2s_df g3_2s_df**	**g1_4s g2_4s_df g3_4s_df**
CONTROLLER	g1_1s g2_2s_dw g3_4s_dw	✓	✗	✗	✗
	g1_1s g2_1s_df g3_1s_df	✗	✓	✗	✗
	g1_2s g2_2s_df g3_2s_df	✗	✓	✓	✗
	g1_4s g2_4s_df g3_4s_df	✗	✓	✓	✓

Table 11-5 Gear2 Controller-PHY Combinations Supported

		PHY				
		g1_1s g2_2s_dw	**g1_2s g2_4s_dw**	**g1_1s g2_1s_df**	**g1_2s g2_2s_df**	**g1_4s g2_4s_df**
CONTROLL ER	g1_1s g2_2s_dw	✓	✗	✓	✗	✗
	g1_2s g2_4s_dw	✓	✓	✓	✓	✗
	g1_1s g2_1s_df	✗	✗	✓	✗	✗
	g1_2s g2_2s_df	✗	✗	✓	✓	✗
	g1_4s g2_4s_df	✗	✗	✓	✓	✓

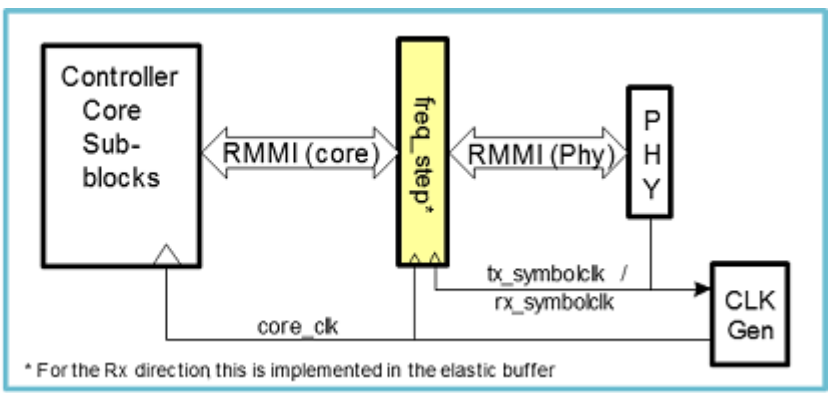
Table 11-6 Gear1 Controller-PHY Combinations Supported

		PHY		
		g1_1s	**g1_2s**	**g1_4s**
CONTROLLER	g1_1s	✓	✗	✗
	g1_2s	✓	✓	✗
	g1_4s	✓	✓	✓

11.3.2.3.1.6 Frequency Step Module

When the controller RMMI lane width (CM_NB) and M-PHY RMMI lane width (CM_PHY_NB) are different, or when the controller and the PHY have different Gear2 speed changing modes, the automatically-derived CM_FREQ_STEP_DOWN_EN parameter is defined, and a RMMI adapter module1 is added between the controller RMMI and the PHY RMMI.

Figure 11-4 Location of freq_step RMMI Adapter Module



This module steps up/down the signals to/from the RMMI interface and adapts (x2,x4, x0.5, x0.25) the controller RMMI width to the PHY RMMI width when the controller clock frequency is slower or faster than the RMMI clock frequency.

11.3.3 Memory Map and Register Definition

This section includes the PCIe module memory map and detailed descriptions of all registers.

NOTE

Synopsys Proprietary. Used with permission.

11.3.3.1 Register Descriptions

11.3.3.1.1 PCIe Memory Map

pcie base address: 3380_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Device ID and Vendor ID Register. (TYPE1_DEV_ID_VEND_ID_REG)	32	RW	ABCD_16C3h
4h	Status and Command Register. (TYPE1_STATUS_COMMAND_REG)	32	RW	0010_0000h
8h	Class Code and Revision ID Register. (TYPE1_CLASS_CODE_REV_ID_REG)	32	RW	0000_0001h
Ch	Header Type, Latency Timer, and Cache Line Size Register. (TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG)	32	RW	0001_0000h
18h	Secondary Latency Timer, Subordinate Bus Number, Secondary Bus Number, and Primary Bus Number Register. (SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG)	32	RW	0000_0000h
1Ch	Secondary Status, and I/O Limit and Base Register. (SEC_STAT_IO_LIMIT_IO_BASE_REG)	32	RW	0000_0000h
20h	Memory Limit and Base Register. (MEM_LIMIT_MEM_BASE_REG)	32	RW	0000_0000h
24h	Prefetchable Memory Limit and Base Register. (PREF_MEM_LIMIT_PREF_MEM_BASE_REG)	32	RW	0000_0000h
28h	Prefetchable Base Upper 32 Bits Register. (PREF_BASE_UPPER_REG)	32	RO	0000_0000h
2Ch	Prefetchable Limit Upper 32 Bits Register. (PREF_LIMIT_UPPER_REG)	32	RO	0000_0000h
30h	I/O Limit and Base Upper 16 Bits Register. (IO_LIMIT_UPPER_IO_BASE_UPPER_REG)	32	RO	0000_0000h
34h	Capabilities Pointer Register. (TYPE1_CAP_PTR_REG)	32	RW	0000_0040h
38h	Expansion ROM Base Address Register. (TYPE1_EXP_ROM_BASE_REG)	32	RW	0000_0000h
3Ch	Bridge Control, Interrupt Pin, and Interrupt Line Register. (BRIDGE_CTRL_INT_PIN_INT_LINE_REG)	32	RW	0000_01FFh
40h	Power Management Capabilities Register. (CAP_ID_NXT_PTR_REG)	32	RW	DBC3_5001h
44h	Power Management Control and Status Register. (CON_STATUS_REG)	32	RW	0000_0000h
50h	MSI Capability ID, Next Pointer, Capability/Control Registers. (PCI_MSI_CAP_ID_NEXT_CTRL_REG)	32	RW	0180_7005h
54h	MSI Message Lower Address Register. (MSI_CAP_OFF_04H_REG)	32	RW	0000_0000h
58h	For a 32 bit MSI Message, this register contains Data. (MSI_CAP_OFF_08H_REG)	32	RW	0000_0000h
5Ch	For a 64 bit MSI Message, this register contains Data. (MSI_CAP_OFF_0CH_REG)	32	RW	0000_0000h
60h	Used for MSI when Vector Masking Capable. (MSI_CAP_OFF_10H_REG)	32	RW	0000_0000h
64h	Used for MSI 64 bit messaging when Vector Masking Capable. (MSI_CAP_OFF_14H_REG)	32	RO	0000_0000h
70h	PCI Express Capabilities, ID, Next Pointer Register. (PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG)	32	RW	0042_0010h
74h	Device Capabilities Register. (DEVICE_CAPABILITIES_REG)	32	RW	0000_8000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
78h	Device Control and Status Register. (DEVICE_CONTROL_DEVICE_STATUS)	32	RW	0000_2010h
7Ch	Link Capabilities Register. (LINK_CAPABILITIES_REG)	32	RW	0073_CC12h
80h	Link Control and Status Register. (LINK_CONTROL_LINK_STATUS_REG)	32	RW	1011_0000h
84h	Slot Capabilities Register. (SLOT_CAPABILITIES_REG)	32	RW	0000_0000h
88h	Slot Control and Status Register. (SLOT_CONTROL_SLOT_STATUS)	32	RW	0040_03C0h
8Ch	Root Control and Capabilities Register. (ROOT_CONTROL_ROOT_CAPABILITIES_REG)	32	RW	0001_0000h
90h	Root Status Register. (ROOT_STATUS_REG)	32	W1C	0000_0000h
94h	Device Capabilities 2 Register. (DEVICE_CAPABILITIES2_REG)	32	RO	0000_041Fh
98h	Device Control 2 and Status 2 Register. (DEVICE_CONTROL2_DEVICE_STATUS2_REG)	32	RW	0000_0000h
9Ch	Link Capabilities 2 Register. (LINK_CAPABILITIES2_REG)	32	RO	0000_0006h
A0h	Link Control 2 and Status 2 Register. (LINK_CONTROL2_LINK_STATUS2_REG)	32	RW	0001_0002h
100h	Advanced Error Reporting Extended Capability Header. (AER_EXT_CAP_HDR_OFF)	32	RW	1482_0001h
104h	Uncorrectable Error Status Register. (UNCORR_ERR_STATUS_OFF)	32	W1C	0000_0000h
108h	Uncorrectable Error Mask Register. (UNCORR_ERR_MASK_OFF)	32	RW	0040_0000h
10Ch	Uncorrectable Error Severity Register. (UNCORR_ERR_SEV_OFF)	32	RW	0046_2030h
110h	Correctable Error Status Register. (CORR_ERR_STATUS_OFF)	32	W1C	0000_0000h
114h	Correctable Error Mask Register. (CORR_ERR_MASK_OFF)	32	RW	0000_E000h
118h	Advanced Error Capabilities and Control Register. (ADV_ERR_CAP_CTRL_OFF)	32	RW	0000_00A0h
11Ch	Header Log Register 0. (HDR_LOG_0_OFF)	32	RO	0000_0000h
120h	Header Log Register 1. (HDR_LOG_1_OFF)	32	RO	0000_0000h
124h	Header Log Register 2. (HDR_LOG_2_OFF)	32	RO	0000_0000h
128h	Header Log Register 3. (HDR_LOG_3_OFF)	32	RO	0000_0000h
12Ch	Root Error Command Register. (ROOT_ERR_CMD_OFF)	32	RW	0000_0000h
130h	Root Error Status Register. (ROOT_ERR_STATUS_OFF)	32	RW	0000_0000h
134h	Error Source Identification Register. (ERR_SRC_ID_OFF)	32	RO	0000_0000h
138h	TLP Prefix Log Register 1. (TLP_PREFIX_LOG_1_OFF)	32	RO	0000_0000h
13Ch	TLP Prefix Log Register 2. (TLP_PREFIX_LOG_2_OFF)	32	RO	0000_0000h
140h	TLP Prefix Log Register 3. (TLP_PREFIX_LOG_3_OFF)	32	RO	0000_0000h
144h	TLP Prefix Log Register 4. (TLP_PREFIX_LOG_4_OFF)	32	RO	0000_0000h
148h	L1 Substates Extended Capability Header. (L1SUB_CAP_HEADER_REG)	32	RW	0001_001Eh
14Ch	L1 Substates Capability Register. (L1SUB_CAPABILITY_REG)	32	RW	0028_0A1Bh
150h	L1 Substates Control 1 Register. (L1SUB_CONTROL1_REG)	32	RW	0000_0A00h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
154h	L1 Substates Control 2 Register. (L1SUB_CONTROL2_REG)	32	RW	0000_0028h
700h	Ack Latency Timer and Replay Timer Register. (ACK_LATENCY_TIMER_OFF)	32	RW	1846_0817h
704h	Vendor Specific DLLP Register. (VENDOR_SPEC_DLLP_OFF)	32	RW	FFFF_FFFFh
708h	Port Force Link Register. (PORT_FORCE_OFF)	32	RW	0000_0004h
70Ch	Ack Frequency and L0-L1 ASPM Control Register. (ACK_F_ASPM_CTRL_OFF)	32	RW	1B2C_2C00h
710h	Port Link Control Register. (PORT_LINK_CTRL_OFF)	32	RW	0001_0120h
714h	Lane Skew Register. (LANE_SKEW_OFF)	32	RW	0000_0000h
718h	Timer Control and Max Function Number Register. (TIMER_CTRL_MAX_FUNC_NUM_OFF)	32	RW	0000_C000h
71Ch	Symbol Timer Register and Filter Mask 1 Register. (SYMBOL_TIMER_FILTER_1_OFF)	32	RW	0000_0280h
720h	Filter Mask 2 Register. (FILTER_MASK_2_OFF)	32	RW	0000_0000h
724h	AMBA Multiple Outbound Decomposed NP SubRequests Control Register. (AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF)	32	RW	0000_0001h
728h	Debug Register 0 (PL_DEBUG0_OFF)	32	RO	See description.
72Ch	Debug Register 1 (PL_DEBUG1_OFF)	32	RO	See description.
730h	Transmit Posted FC Credit Status (TX_P_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h
734h	Transmit Non-Posted FC Credit Status (TX_NP_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h
738h	Transmit Completion FC Credit Status (TX_CPL_FC_CREDIT_STATUS_OFF)	32	RO	0000_0000h
73Ch	Queue Status (QUEUE_STATUS_OFF)	32	RW	0000_0000h
740h	VC Transmit Arbitration Register 1 (VC_TX_ARBI_1_OFF)	32	RO	0000_000Fh
744h	VC Transmit Arbitration Register 2 (VC_TX_ARBI_2_OFF)	32	RO	0000_0000h
748h	Segmented-Buffer VC0 Posted Receive Queue Control. (VC0_P_RX_Q_CTRL_OFF)	32	RW	4520_C019h
74Ch	Segmented-Buffer VC0 Non-Posted Receive Queue Control. (VC0_NP_RX_Q_CTRL_OFF)	32	RW	0520_C003h
750h	Segmented-Buffer VC0 Completion Receive Queue Control. (VC0_CPL_RX_Q_CTRL_OFF)	32	RW	0580_0000h
80Ch	Link Width and Speed Change Control Register. (GEN2_CTRL_OFF)	32	RW	0002_012Ch
810h	PHY Status Register. (PHY_STATUS_OFF)	32	RO	See description.
814h	PHY Control Register. (PHY_CONTROL_OFF)	32	RW	0000_0000h
81Ch	Programmable Target Map Control Register. (TRGT_MAP_CTRL_OFF)	32	RW	0000_007Fh
820h	Integrated MSI Reception Module (iMRM) Address Register. (MSI_CTRL_ADDR_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
824h	Integrated MSI Reception Module Upper Address Register. (MSI_CTRL_UPPER_ADDR_OFF)	32	RW	0000_0000h
828h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_0_EN_OFF)	32	RW	0000_0000h
82Ch	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_0_MASK_OFF)	32	RW	0000_0000h
830h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_0_STATUS_OFF)	32	W1C	0000_0000h
834h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_1_EN_OFF)	32	RW	0000_0000h
838h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_1_MASK_OFF)	32	RW	0000_0000h
83Ch	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_1_STATUS_OFF)	32	W1C	0000_0000h
840h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_2_EN_OFF)	32	RW	0000_0000h
844h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_2_MASK_OFF)	32	RW	0000_0000h
848h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_2_STATUS_OFF)	32	W1C	0000_0000h
84Ch	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_3_EN_OFF)	32	RW	0000_0000h
850h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_3_MASK_OFF)	32	RW	0000_0000h
854h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_3_STATUS_OFF)	32	W1C	0000_0000h
858h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_4_EN_OFF)	32	RW	0000_0000h
85Ch	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_4_MASK_OFF)	32	RW	0000_0000h
860h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_4_STATUS_OFF)	32	W1C	0000_0000h
864h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_5_EN_OFF)	32	RW	0000_0000h
868h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_5_MASK_OFF)	32	RW	0000_0000h
86Ch	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_5_STATUS_OFF)	32	W1C	0000_0000h
870h	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_6_EN_OFF)	32	RW	0000_0000h
874h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_6_MASK_OFF)	32	RW	0000_0000h
878h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_6_STATUS_OFF)	32	W1C	0000_0000h
87Ch	Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_7_EN_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
880h	Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_7_MASK_OFF)	32	RW	0000_0000h
884h	Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_7_STATUS_OFF)	32	W1C	0000_0000h
888h	Integrated MSI Reception Module General Purpose IO Register. (MSI_GPIO_IO_OFF)	32	RW	0000_0000h
88Ch	RADM clock gating enable control register. (CLOCK_GATING_CTRL_OFF)	32	RW	0000_0001h
8B4h	Order Rule Control Register. (ORDER_RULE_CTRL_OFF)	32	RW	0000_0000h
8B8h	PIPE Loopback Control Register. (PIPE_LOOPBACK_CONTROL_OFF)	32	RW	0000_0001h
8BCh	DBI Read-Only Write Enable Register. (MISC_CONTROL_1_OFF)	32	RW	0000_0001h
8C0h	UpConfigure Multi-lane Control Register. (MULTI_LANE_CONTROL_OFF)	32	RW	0000_0000h
8C4h	PHY Interoperability Control Register. (PHY_INTEROP_CTRL_OFF)	32	RW	0000_0237h
8C8h	TRGT_CPL_LUT Delete Entry Control register. (TRGT_CPL_LUT_DELETE_ENTRY_OFF)	32	RW	0000_0000h
8CCh	Link Reset Request Flush Control Register. (LINK_FLUSH_CONTROL_OFF)	32	RW	FF00_0001h
8D0h	AXI Bridge Slave Error Response Register. (AMBA_ERROR_RESPONSE_DEFAULT_OFF)	32	RW	0000_9C00h
8D4h	Link Down AXI Bridge Slave Timeout Register. (AMBA_LINK_TIMEOUT_OFF)	32	RW	0000_0032h
8D8h	AMBA Ordering Control. (AMBA_ORDERING_CTRL_OFF)	32	RW	0000_0000h
8E0h	ACE Cache Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)	32	RW	0000_0000h
8E4h	ACE Cache Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)	32	RW	0000_0000h
8E8h	ACE Cache Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)	32	RW	0000_0000h
8F0h	Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_LOW_OFF)	32	RW	0000_0000h
8F4h	Upper 32 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_HIGH_OFF)	32	RW	0000_0000h
8F8h	PCIe Controller IIP Release Version Number. (PCIE_VERSION_NUMBER_OFF)	32	RO	3530_302Ah
8FCh	PCIe Controller IIP Release Version Type. (PCIE_VERSION_TYPE_OFF)	32	RO	6761_2A2Ah
B40h	Auxiliary Clock Frequency Control Register. (AUX_CLK_FREQ_OFF)	32	RW	0000_000Ah
B44h	L1 Substates Timing Register. (L1_SUBSTATES_OFF)	32	RW	0000_00D2h
8000_0000h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)	32	RW	0000_0000h
8000_0004h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8000_0008h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
8000_000Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
8000_0010h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)	32	RW	0000_FFFFh
8000_0014h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
8000_0018h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)	32	RW	0000_0000h
8000_0100h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_0)	32	RW	0000_0000h
8000_0104h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_0)	32	RW	0000_0000h
8000_0108h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
8000_010Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
8000_0110h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_0)	32	RW	0000_FFFFh
8000_0114h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)	32	RW	0000_0000h
8000_0200h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_0204h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_0208h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_020Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_0210h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_1)	32	RW	0000_FFFFh
8000_0214h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_0218h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1)	32	RW	0000_0000h
8000_0300h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_1)	32	RW	0000_0000h
8000_0304h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_1)	32	RW	0000_0000h
8000_0308h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
8000_030Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
8000_0310h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_1)	32	RW	0000_FFFFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8000_0314h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_1)	32	RW	0000_0000h
8000_0400h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_0404h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_0408h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_040Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_0410h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_2)	32	RW	0000_FFFFh
8000_0414h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_0418h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2)	32	RW	0000_0000h
8000_0500h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_2)	32	RW	0000_0000h
8000_0504h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_2)	32	RW	0000_0000h
8000_0508h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
8000_050Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
8000_0510h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_2)	32	RW	0000_FFFFh
8000_0514h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_2)	32	RW	0000_0000h
8000_0600h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_0604h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_0608h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_060Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_0610h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_3)	32	RW	0000_FFFFh
8000_0614h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_0618h	iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3)	32	RW	0000_0000h
8000_0700h	iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_3)	32	RW	0000_0000h
8000_0704h	iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_3)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8000_0708h	iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
8000_070Ch	iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
8000_0710h	iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_3)	32	RW	0000_FFFFh
8000_0714h	iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_3)	32	RW	0000_0000h
8008_0000h	DMA Arbitration Scheme for TRGT1 Interface. (DMA_CTRL_DATA_ARB_PRIOR_OFF)	32	RW	0000_0688h
8008_0008h	DMA Number of Channels Register. (DMA_CTRL_OFF)	32	RW	0001_0001h
8008_000Ch	DMA Write Engine Enable Register. (DMA_WRITE_ENGINE_EN_OFF)	32	RW	0000_0000h
8008_0010h	DMA Write Doorbell Register. (DMA_WRITE_DOORBELL_OFF)	32	RW	0000_0000h
8008_0018h	DMA Write Engine Channel Arbitration Weight Low Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF)	32	RW	0000_8421h
8008_001Ch	DMA Write Engine Channel Arbitration Weight High Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF)	32	RW	0000_8421h
8008_002Ch	DMA Read Engine Enable Register. (DMA_READ_ENGINE_EN_OFF)	32	RW	0000_0000h
8008_0030h	DMA Read Doorbell Register. (DMA_READ_DOORBELL_OFF)	32	RW	0000_0000h
8008_0038h	DMA Read Engine Channel Arbitration Weight Low Register. (DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF)	32	RW	0000_8421h
8008_003Ch	DMA Read Engine Channel Arbitration Weight High Register. (DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF)	32	RW	0000_8421h
8008_004Ch	DMA Write Interrupt Status Register. (DMA_WRITE_INT_STATUS_OFF)	32	RW	0000_0000h
8008_0054h	DMA Write Interrupt Mask Register. (DMA_WRITE_INT_MASK_OFF)	32	RW	0001_0001h
8008_0058h	DMA Write Interrupt Clear Register. (DMA_WRITE_INT_CLEAR_OFF)	32	W1C	0000_0000h
8008_005Ch	DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF)	32	RO	0000_0000h
8008_0060h	DMA Write Done IMWr Address Low Register. (DMA_WRITE_DONE_IMWR_LOW_OFF)	32	RW	0000_0000h
8008_0064h	DMA Write Done IMWr Interrupt Address High Register. (DMA_WRITE_DONE_IMWR_HIGH_OFF)	32	RW	0000_0000h
8008_0068h	DMA Write Abort IMWr Address Low Register. (DMA_WRITE_ABORT_IMWR_LOW_OFF)	32	RW	0000_0000h
8008_006Ch	DMA Write Abort IMWr Address High Register. (DMA_WRITE_ABORT_IMWR_HIGH_OFF)	32	RW	0000_0000h
8008_0070h	DMA Write Channel 1 and 0 IMWr Data Register. (DMA_WRITE_CH01_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_0074h	DMA Write Channel 3 and 2 IMWr Data Register. (DMA_WRITE_CH23_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_0078h	DMA Write Channel 5 and 4 IMWr Data Register. (DMA_WRITE_CH45_IMWR_DATA_OFF)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8008_007Ch	DMA Write Channel 7 and 6 IMWr Data Register. (DMA_WRITE_CH67_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_0090h	DMA Write Linked List Error Enable Register. (DMA_WRITE_LINKED_LIST_ERR_EN_OFF)	32	RW	0000_0000h
8008_00A0h	DMA Read Interrupt Status Register. (DMA_READ_INT_STATUS_OFF)	32	RW	0000_0000h
8008_00A8h	DMA Read Interrupt Mask Register. (DMA_READ_INT_MASK_OFF)	32	RW	0001_0001h
8008_00ACh	DMA Read Interrupt Clear Register. (DMA_READ_INT_CLEAR_OFF)	32	WO	0000_0000h
8008_00B4h	DMA Read Error Status Low Register. (DMA_READ_ERR_STATUS_LOW_OFF)	32	RO	0000_0000h
8008_00B8h	DMA Read Error Status High Register. (DMA_READ_ERR_STATUS_HIGH_OFF)	32	RO	0000_0000h
8008_00C4h	DMA Read Linked List Error Enable Register. (DMA_READ_LINKED_LIST_ERR_EN_OFF)	32	RW	0000_0000h
8008_00CCh	DMA Read Done IMWr Address Low Register. (DMA_READ_DONE_IMWR_LOW_OFF)	32	RW	0000_0000h
8008_00D0h	DMA Read Done IMWr Address High Register. (DMA_READ_DONE_IMWR_HIGH_OFF)	32	RW	0000_0000h
8008_00D4h	DMA Read Abort IMWr Address Low Register. (DMA_READ_ABORT_IMWR_LOW_OFF)	32	RW	0000_0000h
8008_00D8h	DMA Read Abort IMWr Address High Register. (DMA_READ_ABORT_IMWR_HIGH_OFF)	32	RW	0000_0000h
8008_00DCh	DMA Read Channel 1 and 0 IMWr Data Register. (DMA_READ_CH01_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_00E0h	DMA Read Channel 3 and 2 IMWr Data Register. (DMA_READ_CH23_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_00E4h	DMA Read Channel 5 and 4 IMWr Data Register. (DMA_READ_CH45_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_00E8h	DMA Read Channel 7 and 6 IMWr Data Register. (DMA_READ_CH67_IMWR_DATA_OFF)	32	RW	0000_0000h
8008_0200h	DMA Write Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_WRCH_0)	32	RW	0000_0000h
8008_0208h	DMA Write Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_WRCH_0)	32	RW	0000_0000h
8008_020Ch	DMA Write SAR Low Register. (DMA_SAR_LOW_OFF_WRCH_0)	32	RW	0000_0000h
8008_0210h	DMA Write SAR High Register. (DMA_SAR_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
8008_0214h	DMA Write DAR Low Register. (DMA_DAR_LOW_OFF_WRCH_0)	32	RW	0000_0000h
8008_0218h	DMA Write DAR High Register. (DMA_DAR_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
8008_021Ch	DMA Write Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_WRCH_0)	32	RW	0000_0000h
8008_0220h	DMA Write Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_WRCH_0)	32	RW	0000_0000h
8008_0300h	DMA Read Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_RDCH_0)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8008_0308h	DMA Read Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_RDCH_0)	32	RW	0000_0000h
8008_030Ch	DMA Read SAR Low Register. (DMA_SAR_LOW_OFF_RDCH_0)	32	RW	0000_0000h
8008_0310h	DMA Read SAR High Register. (DMA_SAR_HIGH_OFF_RDCH_0)	32	RW	0000_0000h
8008_0314h	DMA Read DAR Low Register. (DMA_DAR_LOW_OFF_RDCH_0)	32	RW	0000_0000h
8008_0318h	DMA Read DAR High Register. (DMA_DAR_HIGH_OFF_RDCH_0)	32	RW	0000_0000h
8008_031Ch	DMA Read Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_RDCH_0)	32	RW	0000_0000h
8008_0320h	DMA Read Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_RDCH_0)	32	RW	0000_0000h

11.3.3.1.2 Device ID and Vendor ID Register. (TYPE1_DEV_ID_VEND_ID_REG)

11.3.3.1.2.1 Offset

Register	Offset
TYPE1_DEV_ID_VEND_ID_REG	0h

11.3.3.1.2.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DEVICE_ID															
Reset	1	0	1	0	1	0	1	1	1	1	0	0	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VENDOR_ID															
Reset	0	0	0	1	0	1	1	0	1	1	0	0	0	0	1	1

11.3.3.1.2.3 Fields

Field	Function
31-16 DEVICE_ID	Device ID. The Device ID register identifies the particular Function. This identifier is allocated by the vendor. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 VENDOR_ID	Vendor ID. The Vendor ID register identifies the manufacturer of the Function. Valid vendor identifiers are allocated by the PCI-SIG to ensure uniqueness. It is not permitted to populate this register with a value of FFFFh, which is an invalid value for Vendor ID. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.3.3.1.3 Status and Command Register. (TYPE1_STATUS_COMMAND_REG)

11.3.3.1.3.1 Offset

Register	Offset
TYPE1_STATUS_COMMAND_REG	4h

11.3.3.1.3.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DETECTED_PARITY_ERROR	SIGNALED_SYS_ERROR	RCVD_MASTER_ABORT	RCVD_TARGET_ABORT	SIGNALED_TARGET_ABORT	DEV_SEL_TIMING		MASTER_DPE	FAST_B2B_CAP	RSVDP_22	FAST_66MHZ_CAP	CAP_LIST	INT_STATUS	RSVDP_17		Reserved
W	W1C	W1C	W1C	W1C	W1C			W1C								
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RESERV					INT_EN	RSVDP_9	SERRE_N	IDSEL	PERRE_N	VGAPS	MWI_EN	SCO	BME	MSE	IO_EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.3.3 Fields

Field	Function
31 DETECTED_PARITY_ERROR	Detected Parity Error. This bit is set by a Function whenever it receives a Poisoned TLP, regardless of the state the Parity Error Response bit in the Command register. The bit is set when the Poisoned TLP is received by a Function's primary side.
30 SIGNALED_SYS_ERROR	Signaled System Error. This bit is set when a Function sends an ERR_FATAL or ERR_NONFATAL Message, and the SERR# Enable bit in the Command register is 1b.
29 RCVD_MASTER_ABORT	Received Master Abort. This bit is set when a Requester receives a Completion with Unsupported Request Completion status. The bit is set when the Unsupported Request is received by a Function's primary side.
28 RCVD_TARGET_ABORT	Received Target Abort. This bit is set when a Requester receives a Completion with Completer Abort Completion status. The bit is set when the Completer Abort is received by a Function's primary side.
27 SIGNALED_TARGET_ABORT	Signaled Target Abort. This bit is set when a Function completes a Posted or Non-Posted Request as a Completer Abort error. This applies to a Function when the Completer Abort was generated by its primary side.
26-25 DEV_SEL_TIMING	DEVSEL Timing. This field was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires it to 00b.
24 MASTER_DPE	Master Data Parity Error. This bit is set by a Function if the Parity Error Response bit in the Command register is 1b and either of the following two conditions occurs: - Port receives a Poisoned Completion going downstream - Port transmits a Poisoned Request upstream If the Parity Error Response bit is 0b, this bit is never set.
23 FAST_B2B_CAP	Fast Back-to-Back Transactions Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
22 RSVDP_22	Reserved for future use.
21 FAST_66MHZ_CAP	66 MHz Capable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
20 CAP_LIST	Capabilities List. Indicates the presence of an Extended Capability list item. Since all PCI Express device Functions are required to implement the PCI Express Capability structure, the controller hardwires this bit to 1b.
19 INT_STATUS	Interrupt Status. When set, indicates that an INTx emulation interrupt is pending internally in the Function. INTx emulation interrupts forwarded by Functions from the secondary side are not reflected in this bit. Setting the Interrupt Disable bit has no effect on the state of this bit. For Functions that do not generate INTx interrupts, the controller hardwires this bit to 0b.
18-17 RSVDP_17	Reserved for future use.
16 —	Reserved.
15-11	Reserved.

Table continues on the next page...

Field	Function
RESERV	
10 INT_EN	Interrupt Disable. Controls the ability of a Function to generate INTx emulation interrupts. When set, Functions are prevented from asserting INTx interrupts. Note: - Any INTx emulation interrupts already asserted by the Function must be deasserted when this bit is set. INTx interrupts use virtual wires that must, if asserted, be deasserted using the appropriate Deassert_INTx message(s) when this bit is set. - Only the INTx virtual wire interrupt(s) associated with the Function(s) for which this bit is set are affected. - For Functions that generate INTx interrupts on their own behalf, this bit is required. This bit has no effect on interrupts forwarded from the secondary side. For Functions that do not generate INTx interrupts on their own behalf this bit is optional. If this bit is not implemented, the controller hardwires it to 0b.
9 RSVDP_9	Reserved for future use.
8 SERREN	SERR# Enable. When set, this bit enables reporting upstream of Non-fatal and Fatal errors detected by the Function. Note: The errors are reported if enabled either through this bit or through the PCI Express specific bits in the Device Control register. For more details see the "Error Registers" section of the PCI Express Specification. In addition, this bit controls transmission by the primary interface of ERR_NONFATAL and ERR_FATAL error Messages forwarded from the secondary interface. This bit does not affect the transmission of forwarded ERR_COR messages.
7 IDSEL	IDSEL Stepping/Wait Cycle Control. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
6 PERREN	Parity Error Response. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Status register. For more details see the "Error Registers" section of the PCI Express Specification.
5 VGAPS	VGA Palette Snoop. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
4 MWI_EN	Memory Write and Invalidate. This bit was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. For PCI Express to PCI/PCI-X Bridges, refer to the PCI Express to PCI/PCI-X Bridge Specification for requirements for this register.
3 SCO	Special Cycle Enable. This bit was originally described in the PCI Local Bus Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b.
2 BME	Bus Master Enable. This bit controls forwarding of Memory or I/O requests by a port in the Upstream direction. When this bit is 0b, Memory and I/O Requests received at a Root Port must be handled as Unsupported Requests (UR) For Non-Posted Requests a Completion with UR completion status must be returned. This bit does not affect forwarding of Completions in either the Upstream or Downstream direction. The forwarding of Requests other than Memory or I/O Requests is not controlled by this bit.
1 MSE	Memory Space Enable. This bit controls a Function's response to Memory Space accesses received on its primary side. - When set, the Function is enabled to decode the address and further process Memory Space accesses. - When clear, all received Memory Space accesses are caused to be handled as Unsupported Requests. You cannot write to this register if your configuration has no MEM bars; that is, the internal signal has_mem_bar =0. Note: The access attributes of this field are as follows: - Dbi: ! has_mem_bar ? RO : RW
0 IO_EN	IO Space Enable. This bit controls a Function's response to I/O Space accesses received on its primary side. - When set, the Function is enabled to decode the address and further process I/O Space accesses. - When clear, all received I/O accesses are caused to be handled as Unsupported Requests. You cannot write to this register if your configuration has no IO bars; that is, the internal signal has_io_bar =0. Note: The access attributes of this field are as follows: - Dbi: !has_io_bar ? RO : RW

11.3.3.1.4 Class Code and Revision ID Register. (TYPE1_CLASS_CODE_REV_ID_REG)

11.3.3.1.4.1 Offset

Register	Offset
TYPE1_CLASS_CODE_REV_ID_REG	8h

11.3.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_CLASS_CODE								SUBCLASS_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PROGRAM_INTERFACE								REVISION_ID							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

11.3.3.1.4.3 Fields

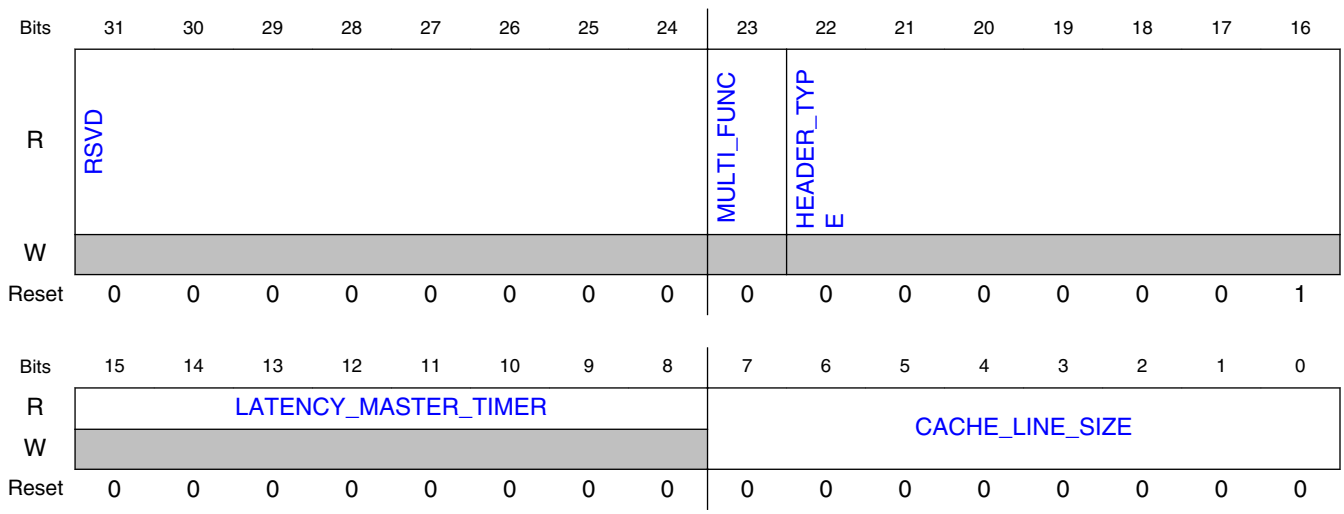
Field	Function
31-24 BASE_CLASS_CODE	Base Class Code. A code that broadly classifies the type of operation the Function performs. Encodings for base class, are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
23-16 SUBCLASS_CODE	Sub-Class Code. Specifies a base class sub-class, which identifies more specifically the operation of the Function. Encodings for sub-class are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-8 PROGRAM_INTERFACE	Programming Interface. This field identifies a specific register level programming interface (if any) so that device independent software can interact with the Function. Encodings for interface are provided in the PCI Code and ID Assignment Specification. All unspecified encodings are reserved. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
7-0 REVISION_ID	Revision ID. The value of this field specifies a Function specific revision identifier. The value is chosen by the vendor. Zero is an acceptable value. The Revision ID should be viewed as a vendor defined extension to the Device ID. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.3.3.1.5 Header Type, Latency Timer, and Cache Line Size Register. (TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG)

11.3.3.1.5.1 Offset

Register	Offset
TYPE1_HDR_TYPE_LAT_CACHE_LINE_SIZE_REG	Ch

11.3.3.1.5.2 Diagram



11.3.3.1.5.3 Fields

Field	Function
31-24 —	Reserved.
23 MULTI_FUNC	Multi-Function Device. - When set, indicates that the device may contain multiple Functions, but not necessarily. Software is permitted to probe for Functions other than Function 0. - When clear, software must not probe for Functions other than Function 0 unless explicitly indicated by another mechanism, such as an ARI or SR-IOV Capability structure. Except where stated otherwise, it is recommended that this bit be set if there are multiple Functions, and clear if there is only one Function. Note: This register field is sticky.
22-16 HEADER_TYPE	Header Layout. This field identifies the layout of the second part of the predefined header. The controller uses 000 0001b encoding. The encoding 000 0010b is reserved. This encoding was originally described in the PC Card Standard Electrical Specification and is used in previous versions of the programming model. Careful consideration should be given to any attempt to repurpose it.

Table continues on the next page...

Field	Function
15-8 LATENCY_MASTER_TIMER	Latency Timer. This register is also referred to as Primary Latency Timer. The Latency Timer was originally described in the PCI Local Bus Specification and the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this register to 00h.
7-0 CACHE_LINE_SIZE	Cache Line Size. The Cache Line Size register is programmed by the system firmware or the operating system to system cache line size. However, legacy conventional PCI software may not always be able to program this register correctly especially in the case of Hot-Plug devices. This read-write register is implemented for legacy compatibility purposes but has no effect on any PCI Express device behavior.

11.3.3.1.6 Secondary Latency Timer, Subordinate Bus Number, Secondary Bus Number, and Primary Bus Number Register. (SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG)

11.3.3.1.6.1 Offset

Register	Offset
SEC_LAT_TIMER_SUB_BUS_SEC_BUS_PRI_BUS_REG	18h

11.3.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SEC_LAT_TIMER								SUB_BUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SEC_BUS								PRIM_BUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.6.3 Fields

Field	Function
31-24 SEC_LAT_TIMER	Secondary Latency Timer. This register does not apply to PCI Express. The controller hardwires it to 00h.
23-16	Subordinate Bus Number. The Subordinate Bus Number register is used to record the bus number of the highest numbered PCI bus segment which is behind (or subordinate to) the bridge. Configuration

Table continues on the next page...

Field	Function
SUB_BUS	software programs the value in this register. The bridge uses this register in conjunction with the Secondary Bus Number register to determine when to respond to and pass on a Type 1 configuration transaction on the primary interface to the secondary interface.
15-8 SEC_BUS	Secondary Bus Number. The Secondary Bus Number register is used to record the bus number of the PCI bus segment to which the secondary interface of the bridge is connected. Configuration software programs the value in this register. The bridge uses this register to determine when to respond to and convert a Type 1 configuration transaction on the primary interface into a Type 0 transaction on the secondary interface.
7-0 PRIM_BUS	Primary Bus Number. This register is not used by PCI Express Functions. It is implemented for compatibility with legacy software.

11.3.3.1.7 Secondary Status, and I/O Limit and Base Register. (SEC_STAT _IO_LIMIT_IO_BASE_REG)

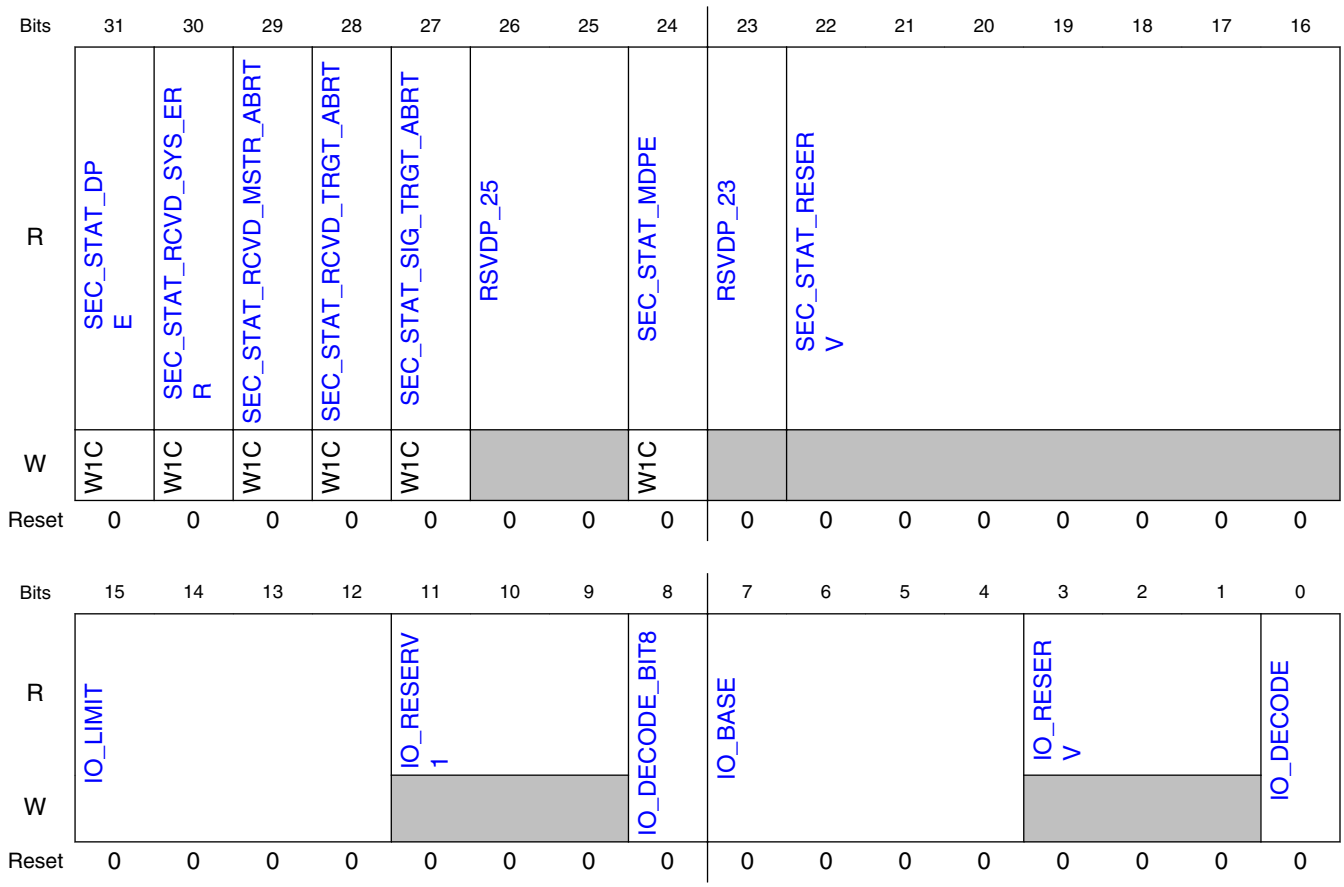
11.3.3.1.7.1 Offset

Register	Offset
SEC_STAT_IO_LIMIT_ IO_BASE_REG	1Ch

11.3.3.1.7.2 Function

Secondary Status, and I/O Limit and Base Register. The I/O Limit and I/O Base registers are optional and define an address range that is used by the bridge to determine when to forward I/O transactions from one interface to the other. If a bridge does not implement an I/O address range, then both the I/O Limit and I/O Base registers must be implemented as read-only registers that return zero when read. If a bridge supports an I/O address range, then these registers must be initialized by configuration software so default states are not specified.

11.3.3.1.7.3 Diagram



11.3.3.1.7.4 Fields

Field	Function
31 SEC_STAT_DPE	Detected Parity Error. This bit is set by a Function when a Poisoned TLP is received by its secondary side, regardless of the state the Parity Error Response Enable bit in the Bridge Control register.
30 SEC_STAT_RCVD_SYS_ERR	Received System Error. This bit is set when the secondary side of a Function receives an ERR_FATAL or ERR_NONFATAL message.
29 SEC_STAT_RCVD_MSTR_ABRT	Received Master Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Unsupported Request Completion status.
28 SEC_STAT_RCVD_TRGT_ABRT	Received Target Abort. This bit is set when the secondary side of a Function (for requests initiated by the Type 1 header Function itself) receives a Completion with Completer Abort Completion status.

Table continues on the next page...

Field	Function
27 SEC_STAT_SIG_TRGT_ABRT	Signaled Target Abort. This bit is set when the secondary side of the Function (for Requests completed by the Type 1 header Function itself) completes a Posted or Non-Posted request as a Completer Abort error.
26-25 RSVDP_25	Reserved for future use.
24 SEC_STAT_MDPE	Master Data Parity Error. This bit is set by a Function if the Parity Error Response Enable bit in the Bridge Control register is set, and either of the following two conditions occurs: - Port receives a Poisoned Completion coming Upstream - Port transmits a Poisoned Request Downstream If the Parity Error Response Enable bit is clear, this bit is never set.
23 RSVDP_23	Reserved for future use.
22-16 SEC_STAT_RESERV	Reserved.
15-12 IO_LIMIT	I/O Limit Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O limit address (not implemented in the I/O Limit register) are FFFh. The I/O Limit register can be programmed to a smaller value than the I/O Base register, if there are no I/O addresses on the secondary side of the bridge. In this case, the bridge will not forward any I/O transactions from the primary bus to the secondary and will forward all I/O transactions from the secondary bus to the primary bus.
11-9 IO_RESERV1	Reserved.
8 IO_DECODE_BIT8	I/O Addressing Encode (IO Limit Address). This bit encodes the IO addressing capability of the bridge. IO_DECODE_BIT8 indicates the following: - 0h: The bridge supports only 16-bit I/O addressing (for ISA compatibility). For the purpose of address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O limit address (not implemented in I/O Limit register) are zero. Note: The bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - 01h: The bridge supports 32-bit I/O address decoding, and the I/O Limit Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Limit address. In this case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. Note: The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Dbi: R
7-4 IO_BASE	I/O Base Address. These bits correspond to the address[15:12] of IO address range. For the purpose of address decoding, the bridge assumes that the lower 12 address bits, address[11:0], of the I/O base address (not implemented in the I/O Base register) are zero.
3-1 IO_RESERV	Reserved.
0 IO_DECODE	I/O Addressing Encode (IO Base Address) This bit encodes the IO addressing capability of the bridge. IO_DECODE indicates the following: - 0h: The bridge supports only 16-bit I/O addressing (for ISA compatibility). For the purpose of address decoding, the bridge assumes that the upper 16 address bits, Address[31:16], of the I/O base address (not implemented in I/O base register) are zero. Note: The bridge must still perform a full 32-bit decode of the I/O address (that is, check that Address[31:16] are 0000h). In this case, the I/O address range supported by the bridge will be restricted to the first 64 KB of I/O Space (0000 0000h to 0000 FFFFh). - 01h: The bridge supports 32-bit I/O address decoding, and the I/O Base Upper 16 Bits hold the upper 16 bits, corresponding to Address[31:16], of the 32-bit Base address. In this case, system configuration software is permitted to locate the I/O address range supported by the bridge anywhere in the 4-GB I/O Space. Note: The 4-KB alignment and granularity restrictions still apply when the bridge supports 32-bit I/O addressing. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.3.3.1.8 Memory Limit and Base Register. (MEM_LIMIT_MEM_BASE_REG)

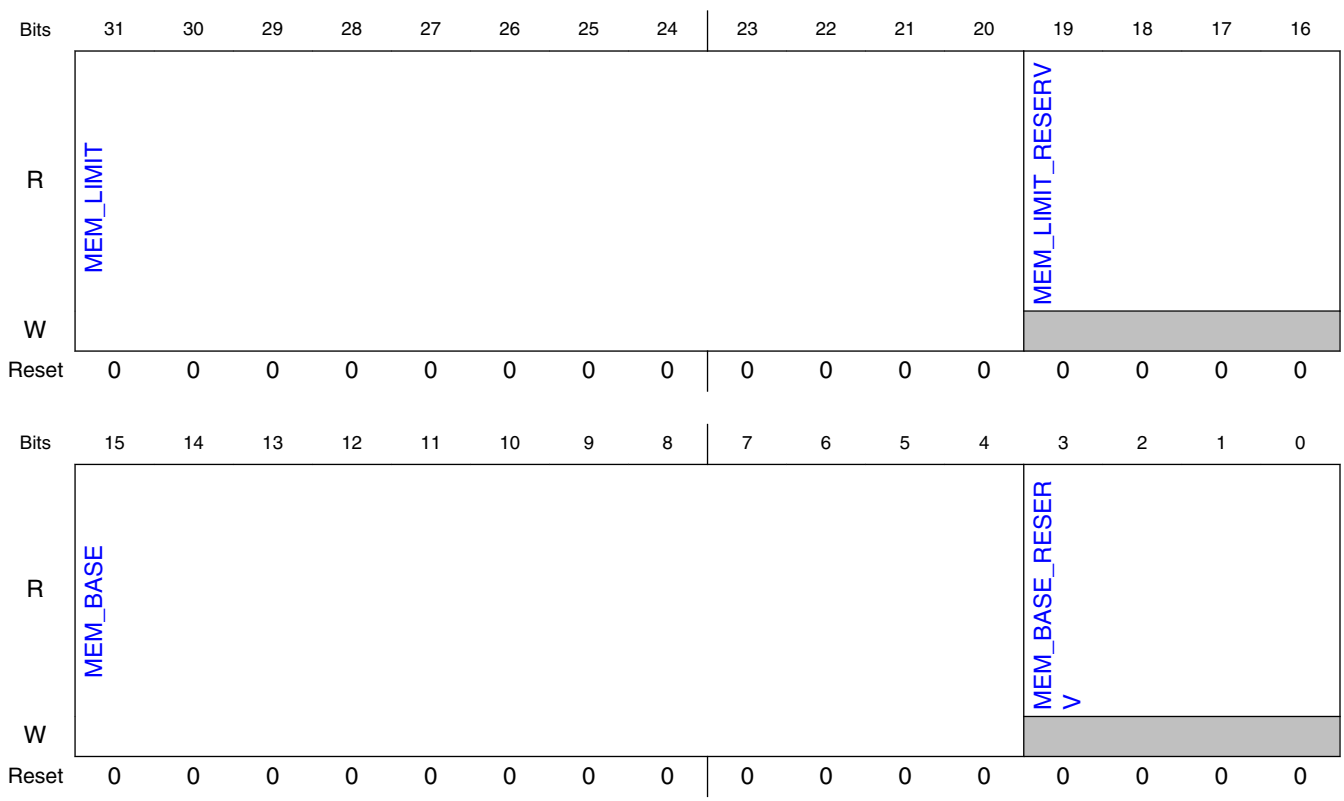
11.3.3.1.8.1 Offset

Register	Offset
MEM_LIMIT_MEM_BASE_REG	20h

11.3.3.1.8.2 Function

Memory Limit and Base Register. The Memory Limit and Memory Base registers define a memory mapped address range which is used by the bridge to determine when to forward memory transactions from one interface to the other. If there is no prefetchable memory space, and there is no memory-mapped space on the secondary side of the bridge, then the bridge will not forward any memory transactions from the primary bus to the secondary bus and will forward all memory transactions from the secondary bus to the primary bus.

11.3.3.1.8.3 Diagram



11.3.3.1.8.4 Fields

Field	Function
31-20 MEM_LIMIT	Memory Limit Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory limit address (not implemented in the Memory Limit register) are F FFFFh. The Memory Limit register must be programmed to a smaller value than the Memory Base register if there is no memory-mapped address space on the secondary side of the bridge.
19-16 MEM_LIMIT_RESERV	Reserved.
15-4 MEM_BASE	Memory Base Address. These bits correspond to the upper 12 address bits, Address[31:20], of 32-bit addresses. For the purpose of address decoding, the bridge assumes that the lower 20 address bits, Address[19:0], of the memory base address (not implemented in the Memory Base register) are zero.
3-0 MEM_BASE_RESERV	Reserved.

11.3.3.1.9 Prefetchable Memory Limit and Base Register. (PREF_MEM_LIMIT_PREF_MEM_BASE_REG)

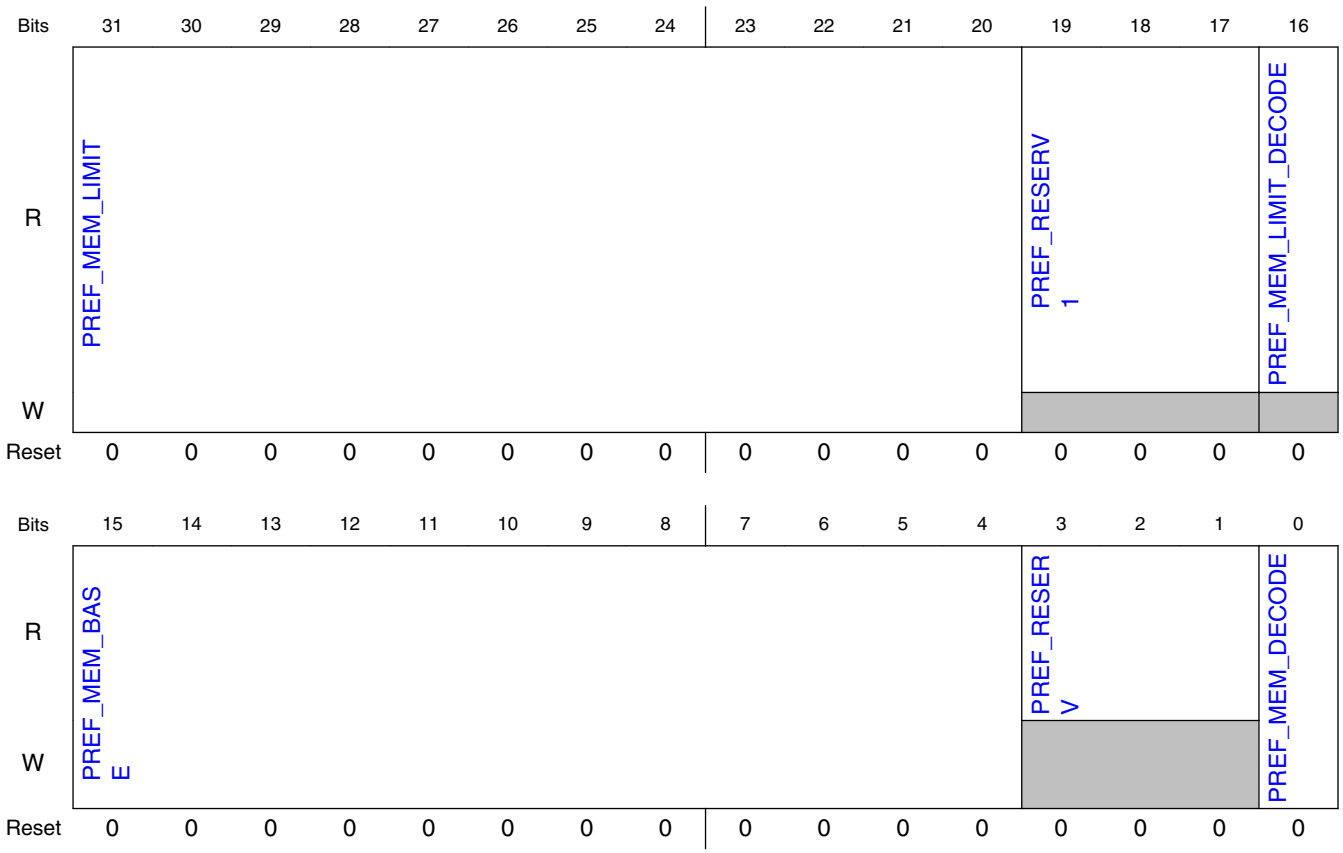
11.3.3.1.9.1 Offset

Register	Offset
PREF_MEM_LIMIT_PREF_MEM_BASE_REG	24h

11.3.3.1.9.2 Function

Prefetchable Memory Limit and Base Register. The Prefetchable Memory Limit and Prefetchable Memory Base registers must indicate that 64-bit addresses are supported, as defined in PCI-to-PCI Bridge Architecture Specification. The Prefetchable Memory Limit and Prefetchable Memory Base registers are optional. They define a prefetchable memory address range which is used by the bridge to determine when to forward memory transactions from one interface to the other (see the PCI-to-PCI Bridge Architecture Specification for additional details).

11.3.3.1.9.3 Diagram



11.3.3.1.9.4 Fields

Field	Function
31-20 PREF_MEM_LIMIT	Prefetchable Memory Limit Address. If the Prefetchable Memory Limit register indicates support for 32-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read-only register that returns zero when read. If the Prefetchable Memory Limit registers indicate support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register which must be initialized by configuration software. If a 64-bit prefetchable memory address range is supported, the Prefetchable Limit Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit limit addresses which specify the prefetchable memory address range.
19-17 PREF_RESERV 1	Reserved.
16 PREF_MEM_LIMIT_DECODE	Prefetchable Memory Limit Decode. This bit encodes whether or not the bridge supports 64-bit addresses. The value of PREF_MEM_LIMIT_DECODE indicates the following: - 0b: Indicates that the bridge supports only 32 bit addresses - 1b: Indicates that the bridge supports 64 bit addresses. Prefetchable Limit Upper 32 Bits registers holds the rest of the 64-bit prefetchable limit address.
15-4 PREF_MEM_BASE	Prefetchable Memory Base Address. If the Prefetchable Memory Base register indicates support for 32-bit addressing, then the Prefetchable Base Upper 32 Bits register is implemented as a read-only register that returns zero when read. If the Prefetchable Memory Base register indicates support for 64-bit addressing, then the Prefetchable Limit Upper 32 Bits register is implemented as a read/write register

Table continues on the next page...

Field	Function
	which must be initialized by configuration software. If a 64-bit prefetchable memory address range is supported, the Prefetchable Base Upper 32 Bits register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range.
3-1 PREF_RESERV	Reserved.
0 PREF_MEM_DECODE	Prefetchable Memory Base Decode. This bit encodes whether or not the bridge supports 64-bit addresses. The value of PREF_MEM_DECODE indicates the following: - 0b: Indicates that the bridge supports only 32 bit addresses. - 1b: Indicates that the bridge supports 64 bit addresses. Prefetchable Base Upper 32 Bits registers holds the rest of the 64-bit prefetchable base address. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.3.3.1.10 Prefetchable Base Upper 32 Bits Register. (PREF_BASE_UPPER_REG)

11.3.3.1.10.1 Offset

Register	Offset
PREF_BASE_UPPER_REG	28h

11.3.3.1.10.2 Function

Prefetchable Base Upper 32 Bits Register. The Prefetchable Base Upper 32 Bits register is an optional extension to the Prefetchable Memory Base register.

11.3.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PREF_MEM_BASE_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PREF_MEM_BASE_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.10.4 Fields

Field	Function
31-0 PREF_MEM_BASE_UPPER	Prefetchable Base Upper 32 Bit. If the Prefetchable Memory Base register indicates support for 32-bit addressing, then this register is implemented as read-only register that returns zero when read. If the Prefetchable Memory Base register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.3.3.1.11 Prefetchable Limit Upper 32 Bits Register. (PREF_LIMIT_UPPER_REG)

11.3.3.1.11.1 Offset

Register	Offset
PREF_LIMIT_UPPER_REG	2Ch

11.3.3.1.11.2 Function

Prefetchable Limit Upper 32 Bits Register. The Prefetchable Limit Upper 32 Bits register is an optional extension to the Prefetchable Memory Limit register.

11.3.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PREF_MEM_LIMIT_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PREF_MEM_LIMIT_UPPER															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.11.4 Fields

Field	Function
31-0 PREF_MEM_LIMIT_UPPER	Prefetchable Limit Upper 32 Bit. If the Prefetchable Memory Limit register indicate support for 64-bit addressing, then this register is implemented as read/write register which must be initialized by configuration software. This register specifies the upper 32 bits, corresponding to Address[63:32], of the 64-bit base addresses which specify the prefetchable memory address range. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.3.3.1.12 I/O Limit and Base Upper 16 Bits Register. (IO_LIMIT_UPPER_IO_BASE_UPPER_REG)

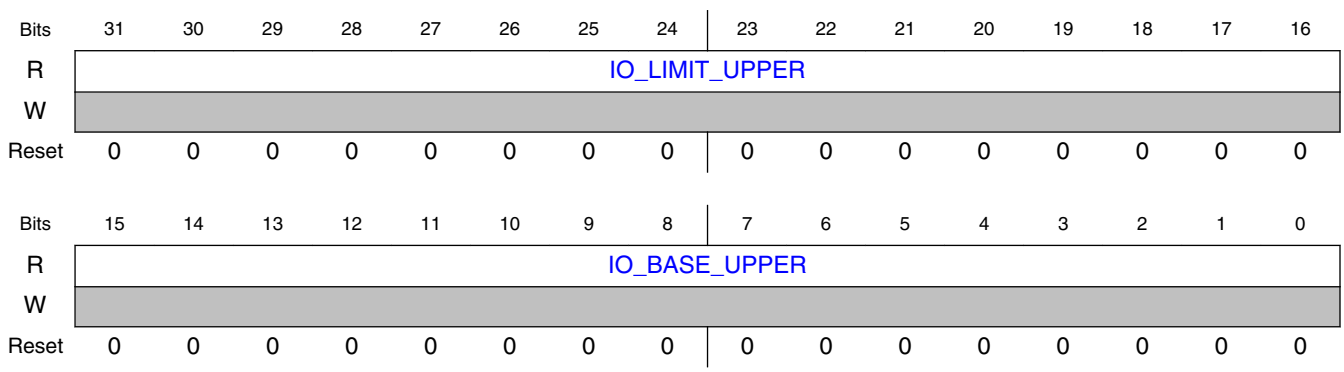
11.3.3.1.12.1 Offset

Register	Offset
IO_LIMIT_UPPER_IO_BASE_UPPER_REG	30h

11.3.3.1.12.2 Function

I/O Limit and Base Upper 16 Bits Register. The I/O Limit Upper 16 Bits and I/O Base Upper 16 Bits registers are optional extensions to the I/O Limit and I/O Base registers.

11.3.3.1.12.3 Diagram



11.3.3.1.12.4 Fields

Field	Function
31-16 IO_LIMIT_UPPER	I/O Limit Upper 16 Bits. If the I/O Limit register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. If the I/O Limit register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software.

Table continues on the next page...

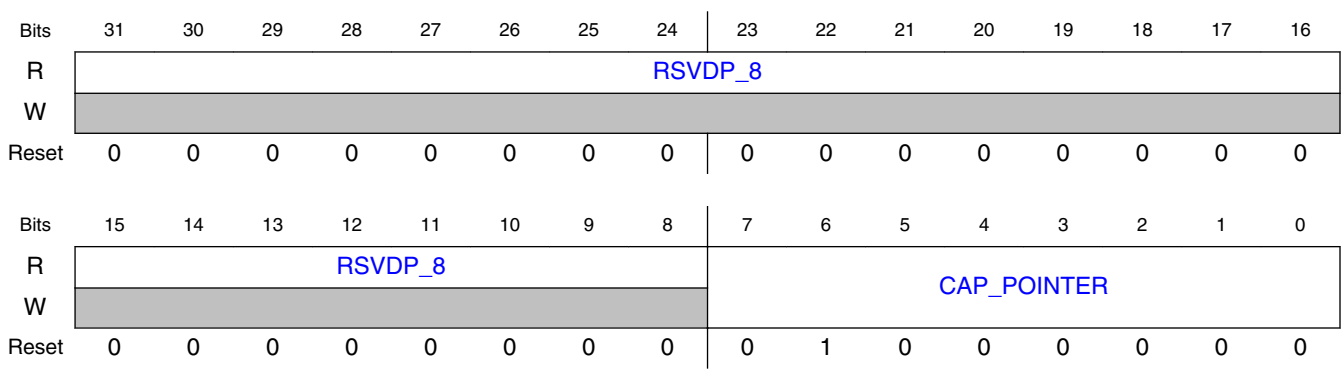
Field	Function
	If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit limit address, that specify the I/O address range. See the PCI-to-PCI Bridge Architecture Specification for additional details). Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
15-0 IO_BASE_UPPER	I/O Base Upper 16 Bits. If the I/O Base register indicates support for 16-bit I/O address decoding, then this register is implemented as a read-only register which return zero when read. If the I/O base register indicates support for 32-bit I/O addressing, then this register must be initialized by configuration software. If 32-bit I/O address decoding is supported, this register specifies the upper 16 bits, corresponding to Address[31:16], of the 32-bit base address, that specify the I/O address range. See the PCI-to-PCI Bridge Architecture Specification for additional details. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.3.3.1.13 Capabilities Pointer Register. (TYPE1_CAP_PTR_REG)

11.3.3.1.13.1 Offset

Register	Offset
TYPE1_CAP_PTR_REG	34h

11.3.3.1.13.2 Diagram



11.3.3.1.13.3 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7-0 CAP_POINTER	Capabilities Pointer. This register is used to point to a linked list of capabilities implemented by this Function. Since all PCI Express Functions are required to implement the PCI Express Capability structure, this register must point to a valid capability structure and either this structure is the PCI Express Capability structure, or a subsequent list item points to the PCI Express Capability structure. The bottom

Field	Function
	two bits are Reserved and must be set to 00b. Software must mask these bits off before using this register as a pointer in Configuration Space to the first entry of a linked list of new capabilities. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.3.3.1.14 Expansion ROM Base Address Register. (TYPE1_EXP_ROM_BASE_REG)

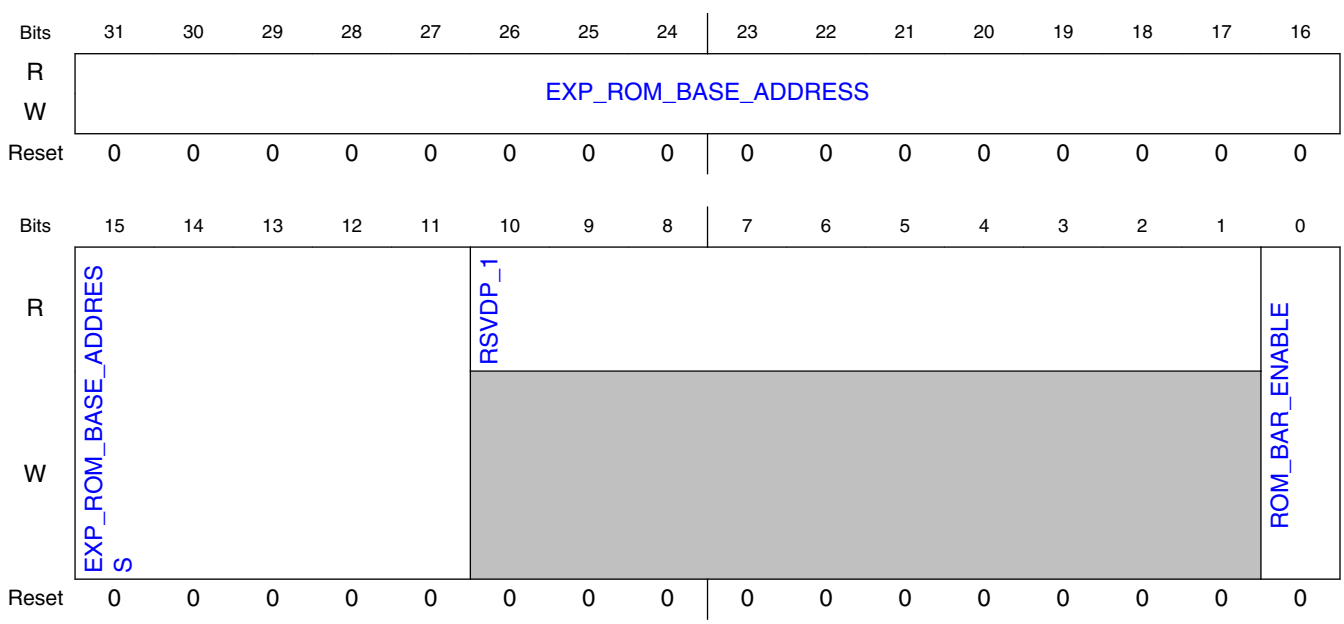
11.3.3.1.14.1 Offset

Register	Offset
TYPE1_EXP_ROM_BASE_REG	38h

11.3.3.1.14.2 Function

Expansion ROM Base Address Register. This register is defined to handle the base address and size information for this expansion ROM. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address.

11.3.3.1.14.3 Diagram



11.3.3.1.14.4 Fields

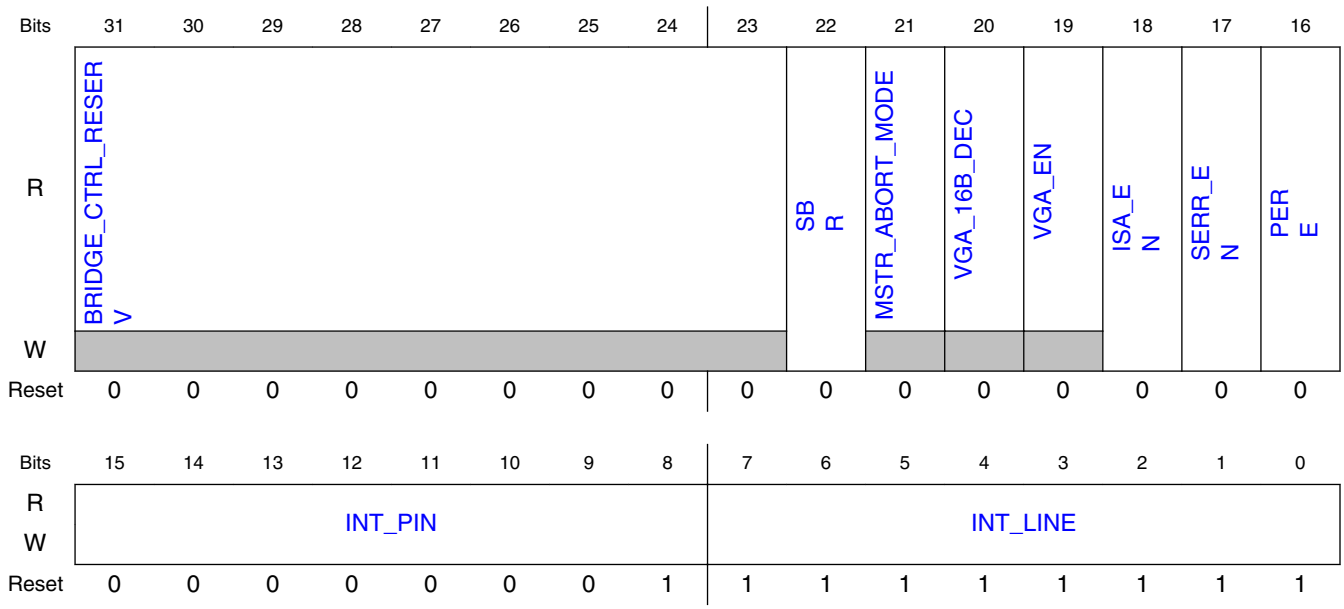
Field	Function
31-11 EXP_ROM_BASE_ADDRESS	Expansion ROM Base Address. Upper 21 bits of the Expansion ROM base address. The number of bits (out of these 21) that a Function actually implements depends on how much address space the Function requires. The mask for this ROM BAR exists (if implemented) as a shadow register at this address. The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the second register at this address. Note: The access attributes of this field are as follows: - Dbi: R/W
10-1 RSVDP_1	Reserved for future use.
0 ROM_BAR_ENABLE	Expansion ROM Enable. This bit controls whether or not the Function accepts accesses to its expansion ROM. When this bit is 0b, the Function's expansion ROM address space is disabled. When the bit is 1b, address decoding is enabled using the parameters in the other part of the Expansion ROM Base Address register. The Memory Space Enable bit in the Command register has precedence over the Expansion ROM Enable bit. A Function must claim accesses to its expansion ROM only if both the Memory Space Enable bit and the Expansion ROM Enable bit are set. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.15 Bridge Control, Interrupt Pin, and Interrupt Line Register. (BRIDGE_CTRL_INT_PIN_INT_LINE_REG)

11.3.3.1.15.1 Offset

Register	Offset
BRIDGE_CTRL_INT_PIN_INT_LINE_REG	3Ch

11.3.3.1.15.2 Diagram



11.3.3.1.15.3 Fields

Field	Function
31-23 BRIDGE_CTRL_RESERV	Reserved.
22 SBR	Secondary Bus Reset. Setting this bit triggers a hot reset on the corresponding PCI Express Port. Software must ensure a minimum reset duration (Trst) as defined in the PCI Local Bus Specification. Software and systems must honor first-access-following-reset timing requirements, unless the Readiness Notifications mechanism is used or if the Immediate Readiness bit in the relevant Function's Status Register register is set. Port configuration registers must not be changed, except as required to update Port status.
21 MSTR_ABORT_MODE	Master Abort Mode. This bit was originally described in the PCI-to-PCI Bridge Architecture Specification. Its functionality does not apply to PCI Express. The controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R/W
20 VGA_16B_DEC	VGA 16 bit decode. This bit only has meaning if VGA Enable bit is set. This bit enables system configuration software to select between 10-bit and 16-bit I/O address decoding for all VGA I/O register accesses that are forwarded from primary to secondary. The following actions are taken based on the value of the VGA_16B_DEC bit: - 0b: Execute 10-bit address decodes on VGA I/O accesses - 1b: Execute 16-bit address decodes on VGA I/O accesses For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R
19 VGA_EN	VGA Enable. Modifies the response by the bridge to VGA compatible addresses. If the VGA Enable bit is set, the bridge will positively decode and forward the following accesses on the primary interface to the secondary interface (and, conversely, block the forwarding of these addresses from the secondary to primary interface): - Memory accesses in the range 000A 0000h to 000B FFFFh - I/O addresses in the first 64 KB of the I/O address space (Address[31:16] are 0000h) where Address[9:0] are in the ranges 3B0h to 3BBh and 3C0h to 3DFh (inclusive of ISA address aliases determined by the setting of VGA 16-bit Decode) If the VGA Enable bit is set, forwarding of these accesses is independent of the I/O address range and memory address ranges defined by the I/O Base and Limit registers, the Memory Base and

Table continues on the next page...

Field	Function
	Limit registers, and the Prefetchable Memory Base and Limit registers of the bridge. (Forwarding of these accesses is also independent of the setting of the ISA Enable bit (in the Bridge Control register) when the VGA Enable bit is set. Forwarding of these accesses is qualified by the I/O Space Enable and Memory Space Enable bits in the Command register.) The following actions are taken based on the value of the VGA_EN bit: - 0b: Do not forward VGA compatible memory and I/O addresses from the primary to the secondary interface (addresses defined above) unless they are enabled for forwarding by the defined I/O and memory address ranges - 1b: Forward VGA compatible memory and I/O addresses (addresses defined above) from the primary interface to the secondary interface (if the I/O Space Enable and Memory Space Enable bits are set) independent of the I/O and memory address ranges and independent of the ISA Enable bit For Functions that do not support VGA, the controller hardwires this bit to 0b. Note: The access attributes of this field are as follows: - Dbi: R
18 ISA_EN	ISA Enable. Modifies the response by the bridge to ISA I/O addresses. This applies only to I/O addresses that are enabled by the I/O Base and I/O Limit registers and are in the first 64 KB of I/O address space (0000 0000h to 0000 FFFFh). If this bit is set, the bridge will block any forwarding from primary to secondary of I/O transactions addressing the last 768 bytes in each 1-KB block. In the opposite direction (secondary to primary), I/O transactions will be forwarded if they address the last 768 bytes in each 1-KB block. The following actions are taken based on the value of the ISA_EN bit: - 0b: Forward downstream all I/O addresses in the address range defined by the I/O Base and I/O Limit registers - 1b: Forward upstream ISA I/O addresses in the address range defined by the I/O Base and I/O Limit registers that are in the first 64 KB of PCI I/O address space (top 768 bytes of each 1-KB block.
17 SERR_EN	SERR# Enable. This bit controls forwarding of ERR_COR, ERR_NONFATAL and ERR_FATAL from secondary to primary.
16 PERE	Parity Error Response Enable. This bit controls the logging of poisoned TLPs in the Master Data Parity Error bit in the Secondary Status register.
15-8 INT_PIN	Interrupt PIN. The Interrupt Pin register register that identifies the legacy interrupt Message(s) the Function uses. Valid values are: - 01h, 02h, 03h, and 04h: map to legacy interrupt Messages for INTA, INTB, INTC, and INTD respectively. - 00h: indicates that the Function uses no legacy interrupt Message(s). - 05h through FFh: Reserved. PCI Express defines one legacy interrupt Message for a single Function device and up to four legacy interrupt Messages for a multi-Function device. For a single Function device, only INTA may be used. Any Function on a multi-Function device can use any of the INTx Messages. If a device implements a single legacy interrupt Message, it must be INTA; if it implements two legacy interrupt Messages, they must be INTA and INTB; and so forth. For a multi-Function device, all Functions may use the same INTx Message or each may have its own (up to a maximum of four Functions) or any combination thereof. A single Function can never generate an interrupt request on more than one INTx Message. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 INT_LINE	Interrupt Line. The Interrupt Line register communicates interrupt line routing information. The register must be implemented by any Function that uses an interrupt pin. Values in this register are programmed by system software and are system architecture specific. The Function itself does not use this value; rather the value in this register is used by device drivers and operating systems.

11.3.3.1.16 Power Management Capabilities Register. (CAP_ID_NXT_PTR_REG)

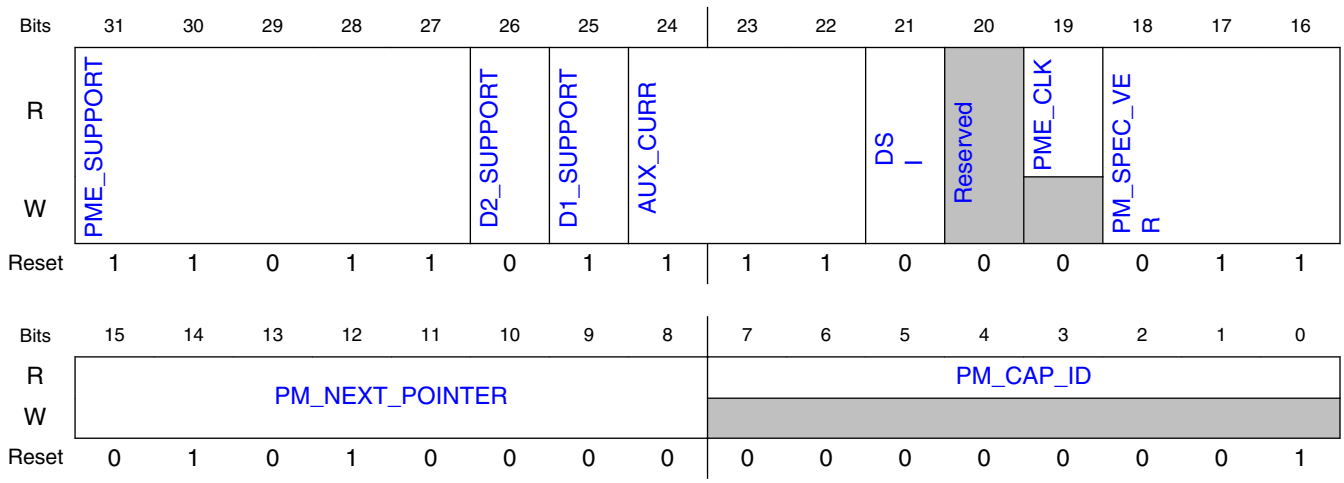
11.3.3.1.16.1 Offset

Register	Offset
CAP_ID_NXT_PTR_REG	40h

11.3.3.1.16.2 Function

Power Management Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.16.3 Diagram



11.3.3.1.16.4 Fields

Field	Function
31-27 PME_SUPPORT	Power Management Event Support. For a description of this standard PCIe register field, see the PCI Express Specification. The read value from this field is the write value && (sys_aux_pwr_det, 1'b1, D2_SUPPORT, D1_SUPPORT, 1'b1), where D1_SUPPORT and D2_SUPPORT are fields in this register. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
26 D2_SUPPORT	D2 State Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
25 D1_SUPPORT	D1 State Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24-22 AUX_CURR	Auxiliary Current Requirements. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
21 DSI	Device Specific Initialization Bit. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20 —	Reserved.
19 PME_CLK	PCI Clock Requirement. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
18-16 PM_SPEC_VER	Power Management Spec Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
15-8 PM_NEXT_POINTER	Next Capability Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PM_CAP_ID	Power Management Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.17 Power Management Control and Status Register. (CON_STATUS_REG)

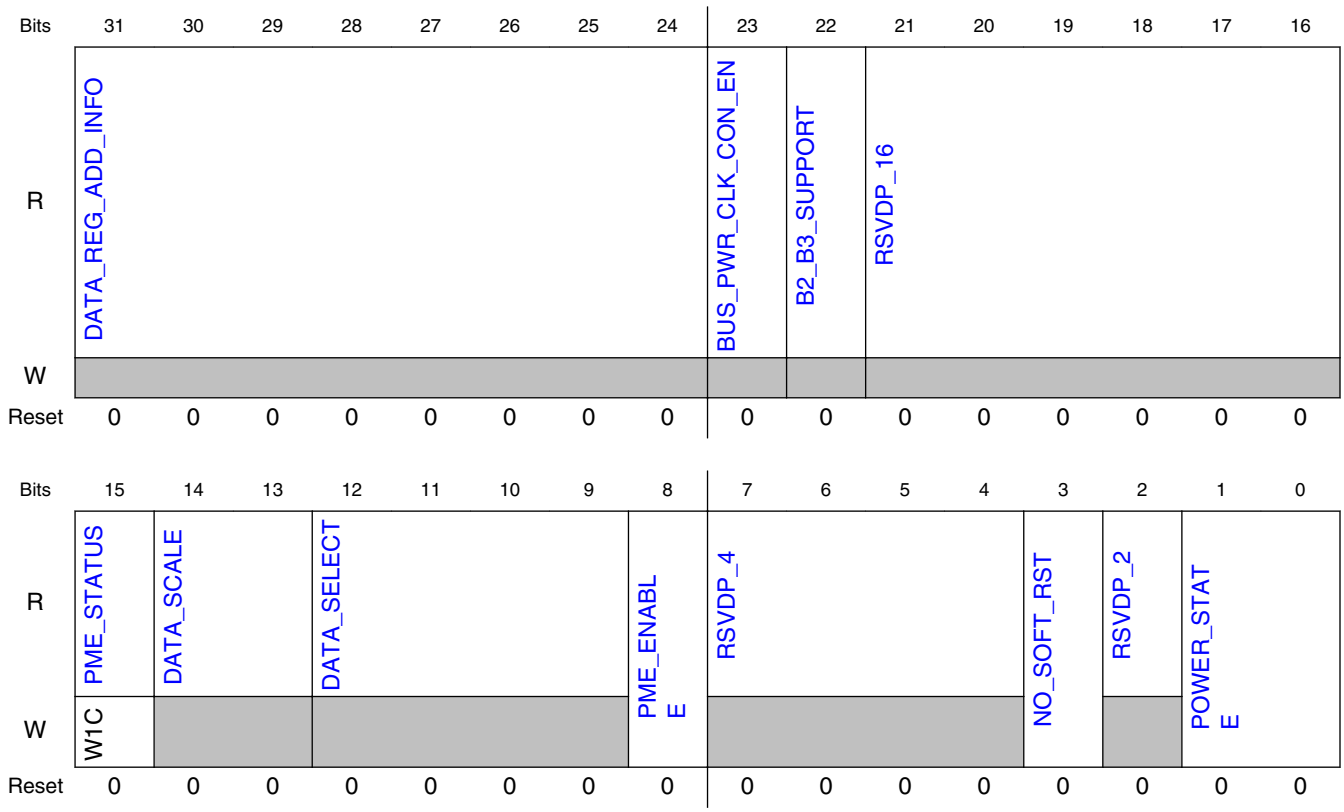
11.3.3.1.17.1 Offset

Register	Offset
CON_STATUS_REG	44h

11.3.3.1.17.2 Function

Power Management Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.17.3 Diagram



11.3.3.1.17.4 Fields

Field	Function
31-24 DATA_REG_ADD_INFO	Power Data Information Register. For a description of this standard PCIe register field, see the PCI Express Specification.
23 BUS_PWR_CLK_CON_EN	Bus Power/Clock Control Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
22 B2_B3_SUPPORT	B2B3 Support for D3hot. For a description of this standard PCIe register field, see the PCI Express Specification.
21-16 RSVDP_16	Reserved for future use.
15 PME_STATUS	PME Status. For a description of this standard PCIe register field, see the PCI Express Specification.
14-13 DATA_SCALE	Data Scaling Factor. For a description of this standard PCIe register field, see the PCI Express Specification.
12-9	Data Select. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
DATA_SELECT	
8 PME_ENABLE	PME Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The PMC registers this value under aux power. Sometimes it might remember the old value, even if you try to clear it by writing '0'. Note: This register field is sticky.
7-4 RSVDP_4	Reserved for future use.
3 NO_SOFT_RST	No soft Reset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2 RSVDP_2	Reserved for future use.
1-0 POWER_STATE	Power State. For a description of this standard PCIe register field, see the PCI Express Specification. You can write to this register. However, the read-back value is the actual power state, not the write value. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.18 MSI Capability ID, Next Pointer, Capability/Control Registers. (PCI_MSI_CAP_ID_NEXT_CTRL_REG)

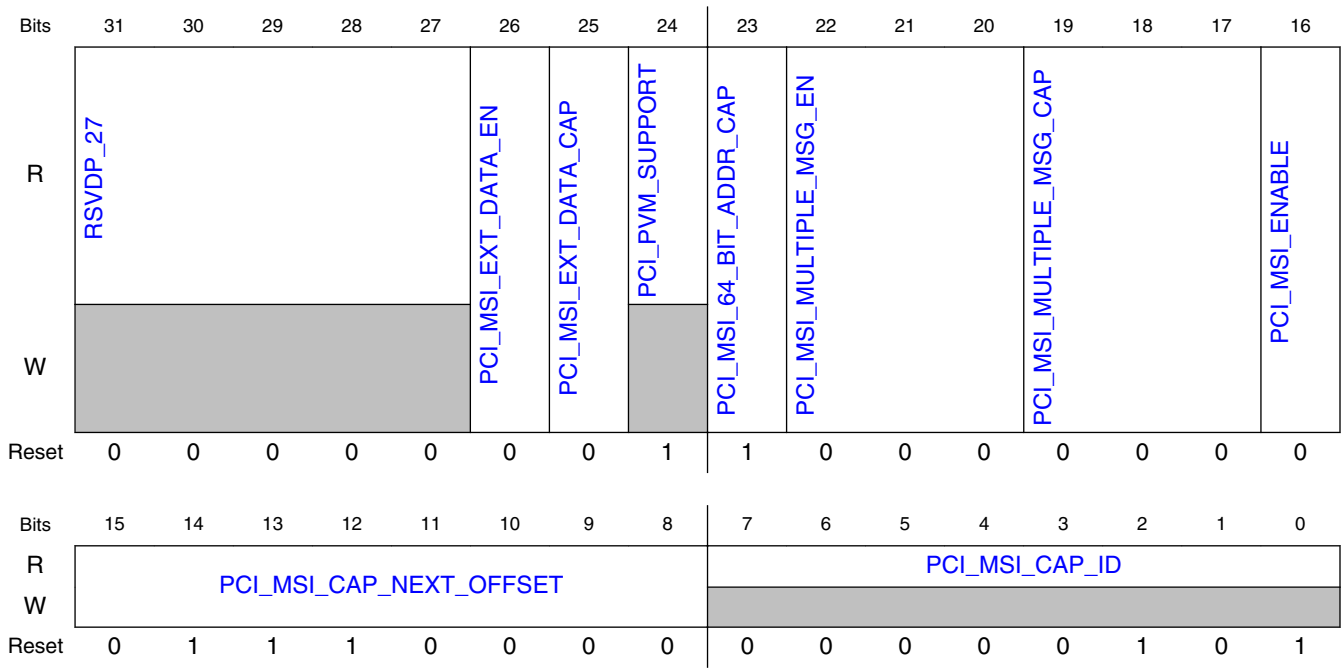
11.3.3.1.18.1 Offset

Register	Offset
PCI_MSI_CAP_ID_NEXT_CTRL_REG	50h

11.3.3.1.18.2 Function

MSI Capability ID, Next Pointer, Capability/Control Registers. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.18.3 Diagram



11.3.3.1.18.4 Fields

Field	Function
31-27 RSVDP_27	Reserved for future use.
26 PCI_MSI_EXT_DATA_EN	Extended Message Data Enable. For a description of this standard PCIe register, see the PCI-SIG ECN for Extended MSI Data, Feb 24, 2016, affecting PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_CAP_ID_NEXT_CTRL_REG.PCI_MSI_EXT_DATA_CAP ? RW : RO
25 PCI_MSI_EXT_DATA_CAP	Extended Message Data Capable. For a description of this standard PCIe register, see the PCI-SIG ECN for Extended MSI Data, Feb 24, 2016, affecting PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24 PCI_PVM_SUPPORT	MSI Per Vector Masking Capable. For a description of this standard PCIe register field, see the PCI Express Specification.
23 PCI_MSI_64_BIT_ADDR_CAP	MSI 64-bit Address Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
22-20 PCI_MSI_MULTIPLE_MSG_EN	MSI Multiple Message Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
19-17	MSI Multiple Message Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

Table continues on the next page...

Field	Function
PCI_MSI_MULTIPLE_MSG_CAP	
16 PCI_MSI_ENABLE	MSI Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
15-8 PCI_MSI_CAP_NEXT_OFFSET	MSI Capability Next Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PCI_MSI_CAP_ID	MSI Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.19 MSI Message Lower Address Register. (MSI_CAP_OFF_04H_REG)

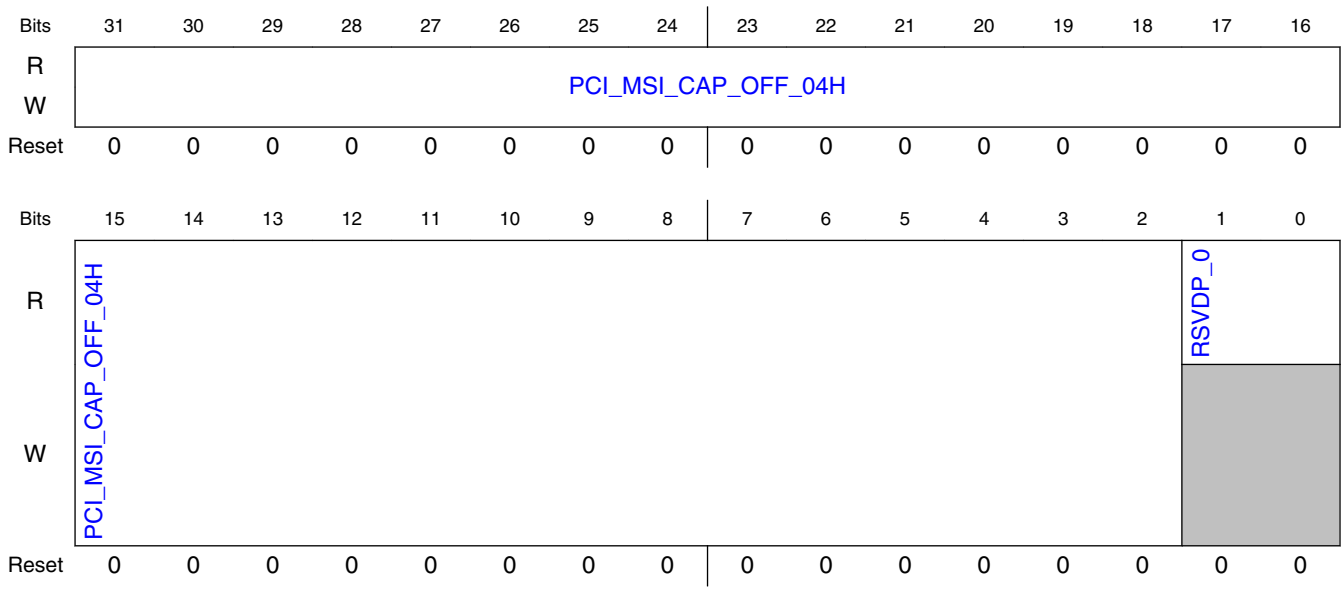
11.3.3.1.19.1 Offset

Register	Offset
MSI_CAP_OFF_04H_REG	54h

11.3.3.1.19.2 Function

MSI Message Lower Address Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.19.3 Diagram



11.3.3.1.19.4 Fields

Field	Function
31-2 PCI_MSI_CAP_OFF_04H	MSI Message Lower Address Field. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
1-0 RSVDP_0	Reserved for future use.

11.3.3.1.20 For a 32 bit MSI Message, this register contains Data. (MSI_CAP_OFF_08H_REG)

11.3.3.1.20.1 Offset

Register	Offset
MSI_CAP_OFF_08H_REG	58h

11.3.3.1.20.2 Function

For a 32 bit MSI Message, this register contains Data. For 64 bit it contains the Upper Address. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCI_MSI_CAP_OFF_0AH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCI_MSI_CAP_OFF_08H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.20.4 Fields

Field	Function
31-16 PCI_MSI_CAP_OFF_0AH	For a 32 bit MSI Message, this field contains Ext MSI Data. For 64-bit it contains upper 16 bits of the Upper Address. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP || `DEFAULT_EXT_MSI_DATA_CAPABLE ? R/W : R
15-0 PCI_MSI_CAP_OFF_08H	For a 32-bit MSI Message, this field contains Data. For 64-bit it contains lower 16 bits of the Upper Address. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP ? R/W : R

11.3.3.1.21 For a 64 bit MSI Message, this register contains Data. (MSI_CAP_OFF_0CH_REG)

11.3.3.1.21.1 Offset

Register	Offset
MSI_CAP_OFF_0CH_REG	5Ch

11.3.3.1.21.2 Function

For a 64 bit MSI Message, this register contains Data. For 32 bit, it contains Mask Bits if PVM enabled. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCI_MSI_CAP_OFF_0EH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCI_MSI_CAP_OFF_0CH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.21.4 Fields

Field	Function
31-16 PCI_MSI_CAP_OFF_0EH	For a 64-bit MSI Message, this field contains Data. For 32-bit, it contains the upper Mask Bits if PVM is enabled. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: (!MSI_64_EN && MSI_PVM_EN_VALUE) ? RW: MSI_64_EN && DEFAULT_EXT_MSI_DATA_CAPABLE ? RW : RO
15-0 PCI_MSI_CAP_OFF_0CH	For a 64-bit MSI Message, this field contains Data. For 32-bit, it contains the lower Mask Bits if PVM is enabled. For a description of this standard PCIe register field, see the PCI Express Specification Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP || MSI_PVM_EN ? R/W : R

11.3.3.1.22 Used for MSI when Vector Masking Capable. (MSI_CAP_OFF_10H_REG)

11.3.3.1.22.1 Offset

Register	Offset
MSI_CAP_OFF_10H_REG	60h

11.3.3.1.22.2 Function

Used for MSI when Vector Masking Capable. For 32 bit contains Pending Bits. For 64 bit, contains Mask Bits. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PCI_MSI_CAP_OFF_10H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PCI_MSI_CAP_OFF_10H															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.22.4 Fields

Field	Function
31-0 PCI_MSI_CAP_OFF_10H	Used for MSI when Vector Masking Capable. For 32-bit contains Pending Bits. For 64-bit, contains Mask Bits. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: PCI_MSI_64_BIT_ADDR_CAP && MSI_PVM_EN ? R/W : R

11.3.3.1.23 Used for MSI 64 bit messaging when Vector Masking Capable. (MSI_CAP_OFF_14H_REG)

11.3.3.1.23.1 Offset

Register	Offset
MSI_CAP_OFF_14H_REG	64h

11.3.3.1.23.2 Function

Used for MSI 64 bit messaging when Vector Masking Capable. Contains Pending Bits. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PCI_MSI_CAP_OFF_14H															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PCI_MSI_CAP_OFF_14H															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.23.4 Fields

Field	Function
31-0 PCI_MSI_CAP_OFF_14H	Used for MSI 64-bit messaging when Vector Masking Capable. Contains Pending Bits. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.24 PCI Express Capabilities, ID, Next Pointer Register. (PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG)

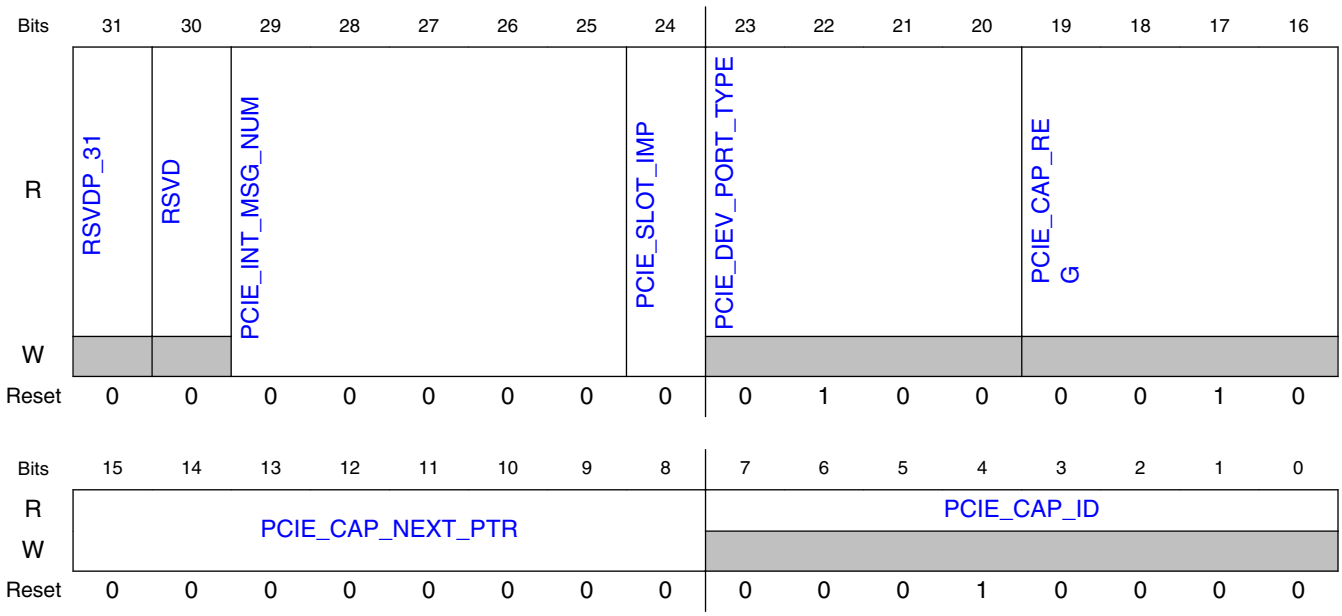
11.3.3.1.24.1 Offset

Register	Offset
PCIE_CAP_ID_PCIE_NEXT_CAP_PTR_PCIE_CAP_REG	70h

11.3.3.1.24.2 Function

PCI Express Capabilities, ID, Next Pointer Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.24.3 Diagram



11.3.3.1.24.4 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30 RSVD	Reserved. For a description of this standard PCIe register field, see the PCI Express Specification.
29-25 PCIE_INT_MSG_NUM	PCIE Interrupt Message Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
24 PCIE_SLOT_IMP	PCIe Slot Implemented Valid. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
23-20 PCIE_DEV_PORT_TYPE	PCIE Device/PortType. For a description of this standard PCIe register field, see the PCI Express Specification.
19-16 PCIE_CAP_REG	PCIE Capability Version Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 PCIE_CAP_NEXT_PTR	PCIE Next Capability Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
7-0 PCIE_CAP_ID	PCIE Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.25 Device Capabilities Register. (DEVICE_CAPABILITIES_REG)

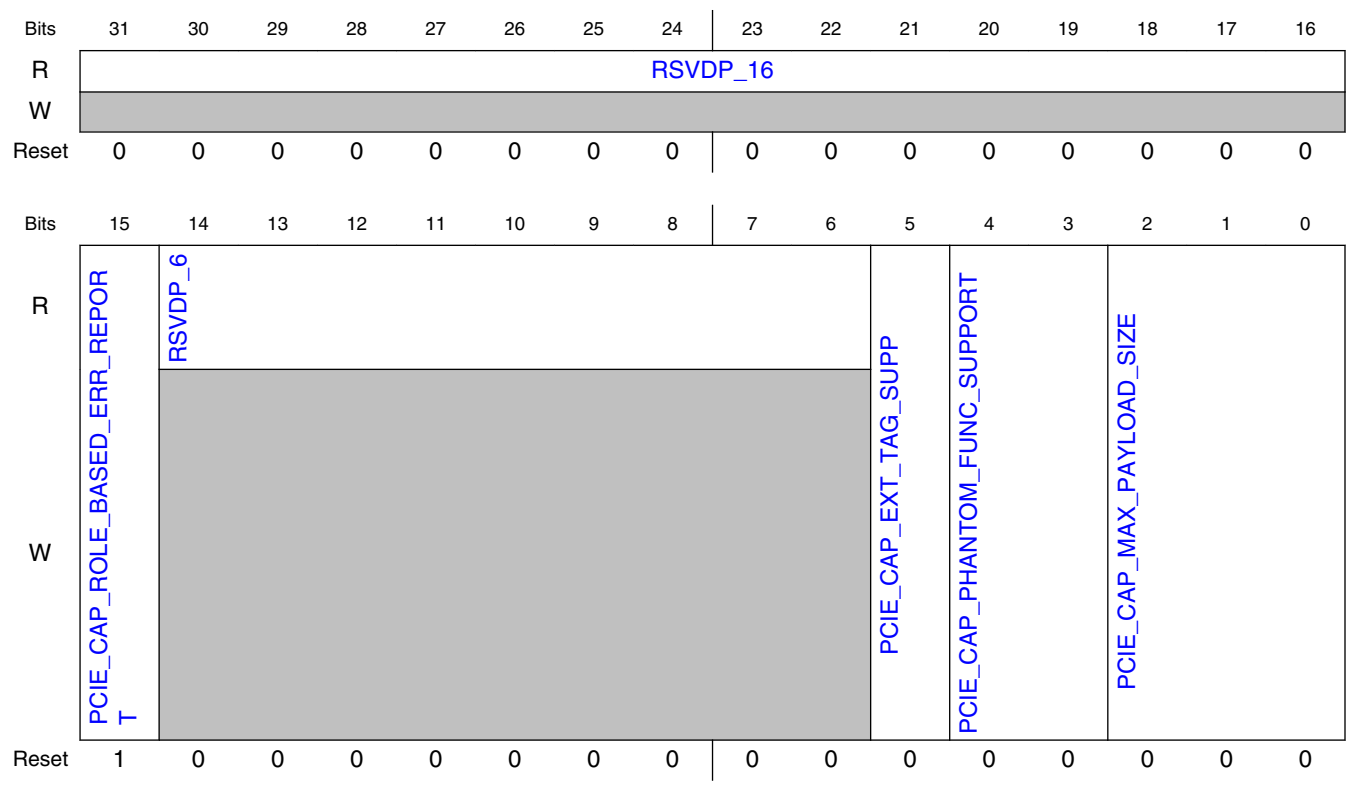
11.3.3.1.25.1 Offset

Register	Offset
DEVICE_CAPABILITIES_REG	74h

11.3.3.1.25.2 Function

Device Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.25.3 Diagram



11.3.3.1.25.4 Fields

Field	Function
31-16	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_16	
15 PCIE_CAP_ROLE_BASED_ERR_REPORT	Role-based Error Reporting Implemented. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14-6 RSVDP_6	Reserved for future use.
5 PCIE_CAP_EXT_TAG_SUPP	Extended Tag Field Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
4-3 PCIE_CAP_PHANTOM_FUNC_SUPPORT	Phantom Functions Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
2-0 PCIE_CAP_MAX_PAYLOAD_SIZE	Max Payload Size Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.3.3.1.26 Device Control and Status Register. (DEVICE_CONTROL_DEVICE_STATUS)

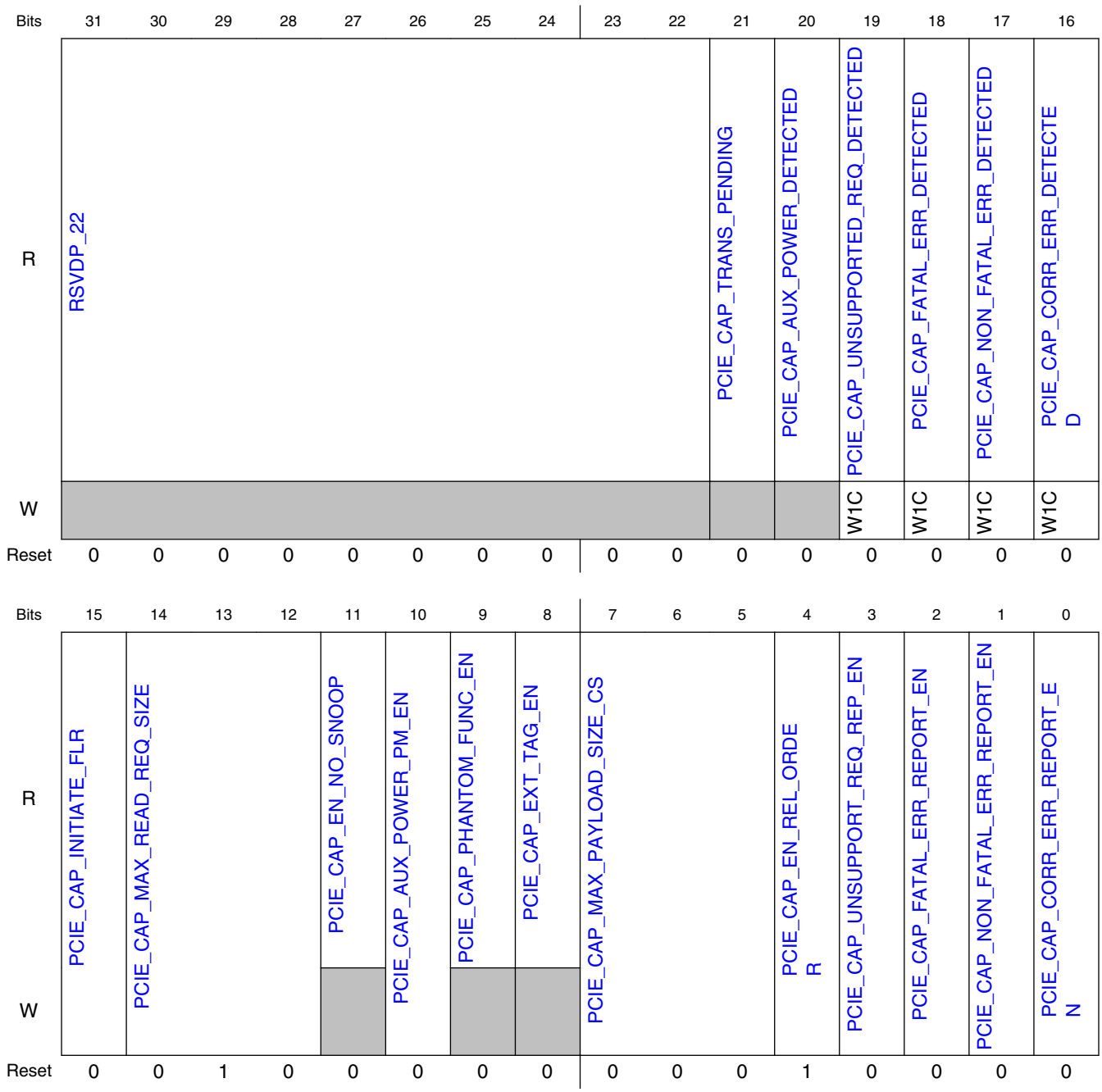
11.3.3.1.26.1 Offset

Register	Offset
DEVICE_CONTROL_DEVICE_STATUS	78h

11.3.3.1.26.2 Function

Device Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.26.3 Diagram



11.3.3.1.26.4 Fields

Field	Function
31-22 RSVDP_22	Reserved for future use.

Table continues on the next page...

Field	Function
21 PCIE_CAP_TRANS_PENDING	Transactions Pending Status. For a description of this standard PCIe register field, see the PCI Express Specification.
20 PCIE_CAP_AUX_POWER_DETECTED	Aux Power Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification. This bit is derived by sampling the sys_aux_pwr_det input.
19 PCIE_CAP_UNSUPPORTED_REQ_DETECTED	Unsupported Request Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
18 PCIE_CAP_FATAL_ERR_DETECTED	Fatal Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP_NON_FATAL_ERR_DETECTED	Non-Fatal Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_CORR_ERR_DETECTED	Correctable Error Detected Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15 PCIE_CAP_INITIATE_FLR	Initiate Function Level Reset (for endpoints). For a description of this standard PCIe register field, see the PCI Express Specification.
14-12 PCIE_CAP_MAX_READ_REQ_SIZE	Max Read Request Size. For a description of this standard PCIe register field, see the PCI Express Specification.
11 PCIE_CAP_EN_NO_SNOOP	Enable No Snoop. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R
10 PCIE_CAP_AUX_POWER_PM_EN	Aux Power PM Enable. For a description of this standard PCIe register field, see the PCI Express Specification. This bit is derived by sampling the sys_aux_pwr_det input. Note: This register field is sticky.
9 PCIE_CAP_PHANTOM_FUNC_EN	Phantom Functions Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_PHANTOM_FUNC_SUPPORT field of DEVICE_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_PHANTOM_FUNC_SUPPORT ? RW : RO
8 PCIE_CAP_EXT_TAG_EN	Extended Tag Field Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_EXT_TAG_SUPP field of DEVICE_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: DEVICE_CAPABILITIES_REG.PCIE_CAP_EXT_TAG_SUPP ? RW : RO

Table continues on the next page...

Field	Function
7-5 PCIE_CAP_MAX_PAYLOAD_SIZE_CS	Max Payload Size. Max_Payload_Size . This field sets maximum TLP payload size for the Function. Permissible values that can be programmed are indicated by the Max_Payload_Size Supported field (PCIE_CAP_MAX_PAYLOAD_SIZE) in the Device Capabilities register (DEVICE_CAPABILITIES_REG).
4 PCIE_CAP_EN_REL_ORDER	Enable Relaxed Ordering. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
3 PCIE_CAP_UNSUPPORT_REQ_REP_EN	Unsupported Request Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_FATAL_ERR_REPORT_EN	Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_NON_FATAL_ERR_REPORT_EN	Non-fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_CORR_ERR_REPORT_EN	Correctable Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.27 Link Capabilities Register. (LINK_CAPABILITIES_REG)

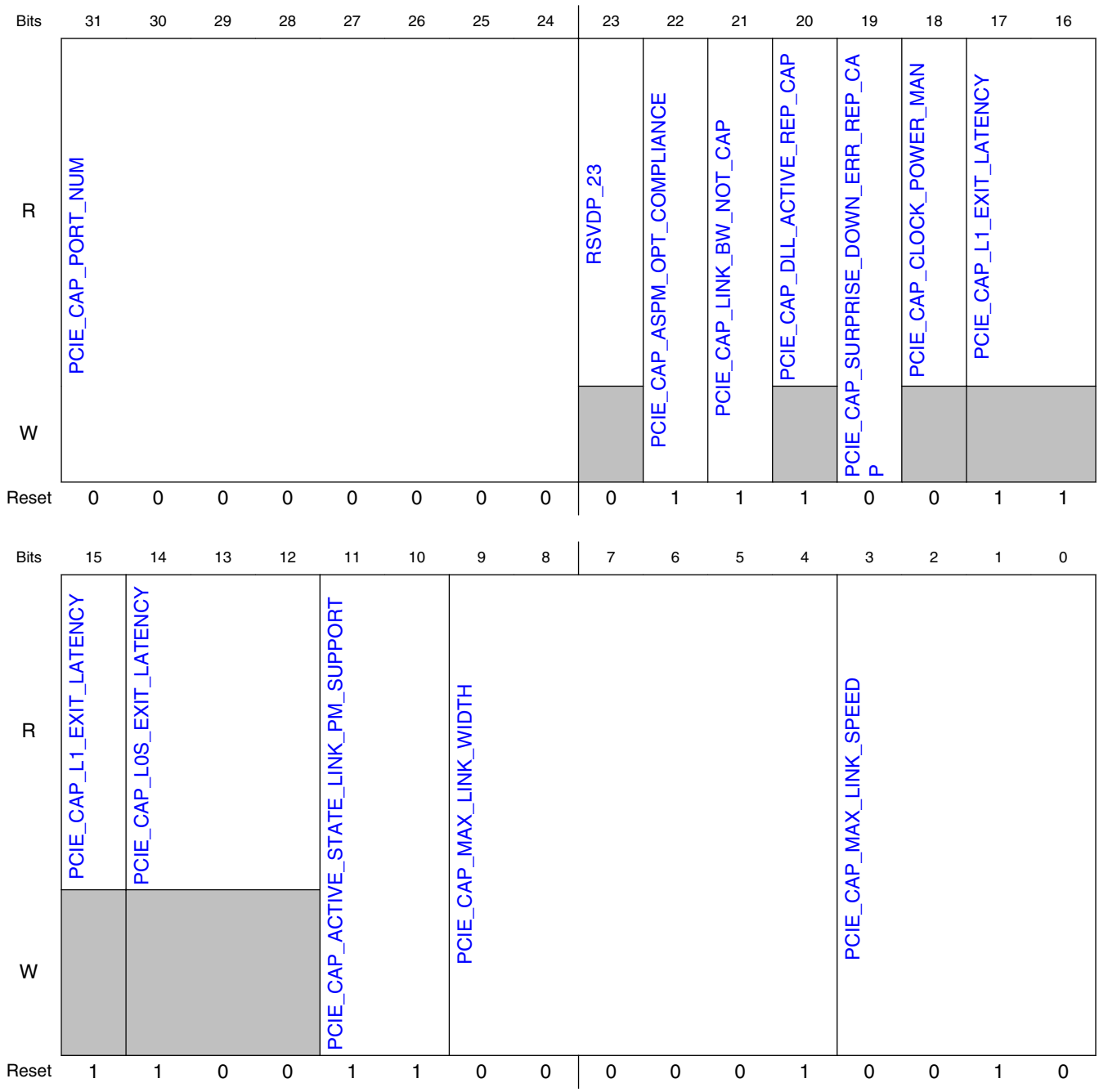
11.3.3.1.27.1 Offset

Register	Offset
LINK_CAPABILITIES_REG	7Ch

11.3.3.1.27.2 Function

Link Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.27.3 Diagram



11.3.3.1.27.4 Fields

Field	Function
31-24 PCIE_CAP_PORT_NUM	Port Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Table continues on the next page...

Field	Function
23 RSVDP_23	Reserved for future use.
22 PCIE_CAP_ASPM_OPT_COMPLIANCE	ASPM Optionality Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
21 PCIE_CAP_LINK_BW_NOT_CAP	Link Bandwidth Notification Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
20 PCIE_CAP_DLL_ACTIVE_REP_CAP	Data Link Layer Link Active Reporting Capable. For a description of this standard PCIe register field, see the PCI Express Specification.
19 PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP	Surprise Down Error Reporting Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
18 PCIE_CAP_CLOCK_POWER_MAN	Clock Power Management. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17-15 PCIE_CAP_L1_EXIT_LATENCY	L1 Exit Latency. For a description of this standard PCIe register field, see the PCI Express Specification. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
14-12 PCIE_CAP_L0S_EXIT_LATENCY	LOs Exit Latency. For a description of this standard PCIe register field, see the PCI Express Specification. There are two each of these register fields, this one and a shadow one at the same address. The Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG) determines which one is used by the controller and which one is accessed by a read request. Common Clock operation is enabled in the controller when you set the Common Clock bit (PCIE_CAP_COMMON_CLK_CONFIG) of the Link Control Register (LINK_CONTROL_LINK_STATUS_REG). The assertion of CS2 (that is, assert the dbi_cs2 input, or the CS2 address bit for the AXI bridge) is required to write to the shadow field at this location. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
11-10 PCIE_CAP_ACTIVE_STATE_LINK_PM_SUPPORT	Level of ASPM (Active State Power Management) Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
9-4	Maximum Link Width. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller.

Table continues on the next page...

Field	Function
PCIE_CAP_MAX_LINK_WIDTH	Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.
3-0 PCIE_CAP_MAX_LINK_SPEED	Maximum Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the reset and dynamic values of this field are calculated by the controller. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R Note: This register field is sticky.

11.3.3.1.28 Link Control and Status Register. (LINK_CONTROL_LINK_STATUS_REG)

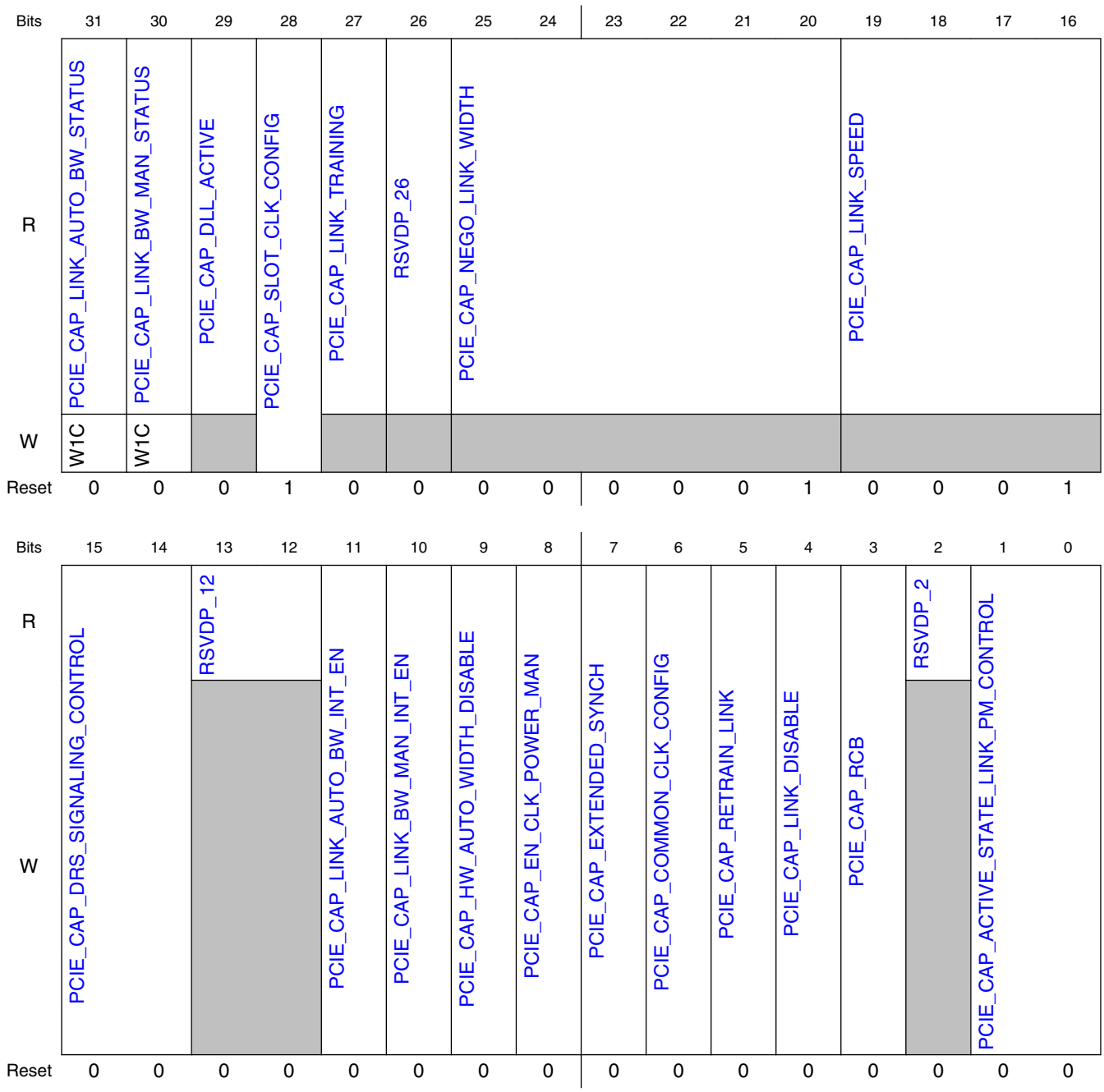
11.3.3.1.28.1 Offset

Register	Offset
LINK_CONTROL_LINK_STATUS_REG	80h

11.3.3.1.28.2 Function

Link Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.28.3 Diagram



11.3.3.1.28.4 Fields

Field	Function
31	Link Autonomous Bandwidth Status. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG.

Table continues on the next page...

Field	Function
PCIE_CAP_LINK_AUTO_BW_STATUS	
30 PCIE_CAP_LINK_BW_MAN_STATUS	Link Bandwidth Management Status. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG.
29 PCIE_CAP_DLL_ACTIVE	Data Link Layer Active. For a description of this standard PCIe register field, see the PCI Express Specification.
28 PCIE_CAP_SLOT_CLK_CONFIG	Slot Clock Configuration. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
27 PCIE_CAP_LINK_TRAINING	LTSSM is in Configuration or Recovery State. For a description of this standard PCIe register field, see the PCI Express Specification.
26 RSVDP_26	Reserved for future use.
25-20 PCIE_CAP_NEGO_LINK_WIDTH	Negotiated Link Width. For a description of this standard PCIe register field, see the PCI Express Specification.
19-16 PCIE_CAP_LINK_SPEED	Current Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification.
15-14 PCIE_CAP_DRS_SIGNALING_CONTROL	DRS Signaling Control. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES2_REG.DRS_SUPPORTED ? RW : RO
13-12 RSVDP_12	Reserved for future use.
11 PCIE_CAP_LINK_AUTO_BW_INT_EN	Link Autonomous Bandwidth Management Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
10 PCIE_CAP_LINK_BW_MAN_INT_EN	Link Bandwidth Management Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_LINK_BW_NOT_CAP field in LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_LINK_BW_NOT_CAP ? RW : RO
9 PCIE_CAP_HW_AUTO_WIDTH_DISABLE	Hardware Autonomous Width Disable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
8	Enable Clock Power Management. For a description of this standard PCIe register field, see the PCI Express Specification. The write value is gated with the PCIE_CAP_CLOCK_POWER_MAN field in

Table continues on the next page...

Field	Function
PCIE_CAP_EN_CLK_POWER_MAN	LINK_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_CLOCK_POWER_MAN ? RWS : ROS Note: This register field is sticky.
7 PCIE_CAP_EXTENDED_SYNCH	Extended Synch. For a description of this standard PCIe register field, see the PCI Express Specification.
6 PCIE_CAP_COMMON_CLK_CONFIG	Common Clock Configuration. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_RETRAIN_LINK	Initiate Link Retrain. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: see description
4 PCIE_CAP_LINK_DISABLE	Initiate Link Disable. For a description of this standard PCIe register field, see the PCI Express Specification. In a DSP that supports crosslink, the controller gates the write value with the CROSS_LINK_EN field in PORT_LINK_CTRL_OFF.
3 PCIE_CAP_RCB	Read Completion Boundary (RCB). Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2 RSVDP_2	Reserved for future use.
1-0 PCIE_CAP_ACTIVE_STATE_LINK_PM_CONTROL	Active State Power Management (ASPM) Control. Software must not enable L0s in either direction on a given Link unless components on both sides of the Link each support L0s; otherwise, the result is undefined. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.29 Slot Capabilities Register. (SLOT_CAPABILITIES_REG)

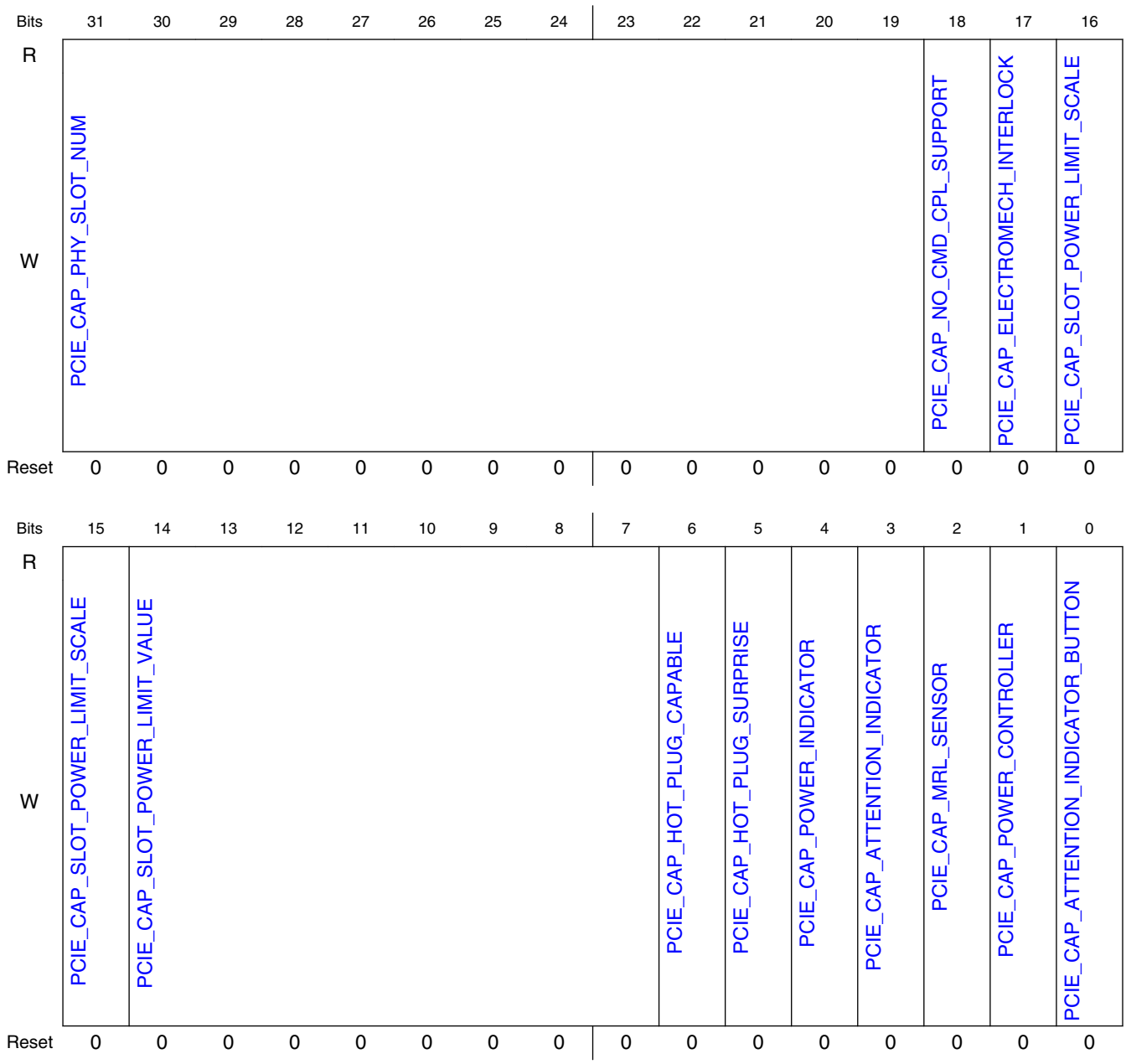
11.3.3.1.29.1 Offset

Register	Offset
SLOT_CAPABILITIES_REG	84h

11.3.3.1.29.2 Function

Slot Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.29.3 Diagram



11.3.3.1.29.4 Fields

Field	Function
31-19 PCIE_CAP_PHY_SLOT_NUM	Physical Slot Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

Table continues on the next page...

Field	Function
18 PCIE_CAP_NO_CMD_CPL_SUPPORT	No Command Completed Support. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
17 PCIE_CAP_ELECTROMECH_INTERLOCK	Electromechanical Interlock Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
16-15 PCIE_CAP_SLOT_POWER_LIMIT_SCALE	Slot Power Limit Scale. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
14-7 PCIE_CAP_SLOT_POWER_LIMIT_VALUE	Slot Power Limit Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
6 PCIE_CAP_HOT_PLUG_CAPABLE	Hot Plug Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
5 PCIE_CAP_HOT_PLUG_SURPRISE	Hot Plug Surprise possible. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
4 PCIE_CAP_POWER_INDICATOR	Power Indicator Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
3 PCIE_CAP_ATTENTION_INDICATOR	Attention Indicator Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
2 PCIE_CAP_MRL_SENSOR	MRL Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
1 PCIE_CAP_POWER_CONTROLLER	Power Controller Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R
0 PCIE_CAP_ATTENTION_INDICATOR_BUTTON	Attention Button Present. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W else R

11.3.3.1.30 Slot Control and Status Register. (SLOT_CONTROL_SLOT_STATUS)

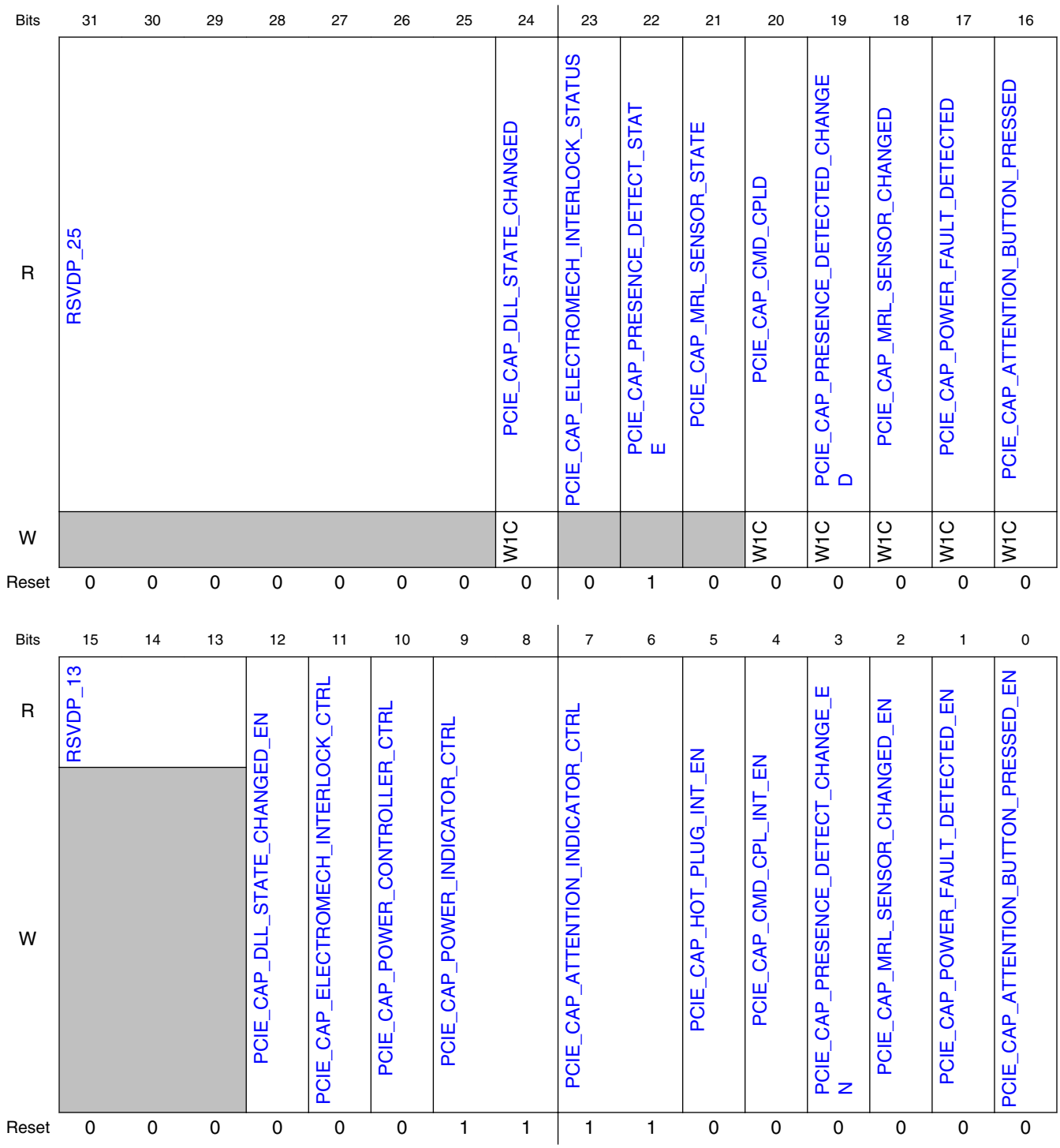
11.3.3.1.30.1 Offset

Register	Offset
SLOT_CONTROL_SLOT_STATUS	88h

11.3.3.1.30.2 Function

Slot Control and Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.30.3 Diagram



11.3.3.1.30.4 Fields

Field	Function
31-25	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_25	
24 PCIE_CAP_DLL_STATE_CHANGED	Data Link Layer State Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
23 PCIE_CAP_ELECTROMECH_INTERLOCK_STATUS	Electromechanical Interlock Status. For a description of this standard PCIe register field, see the PCI Express Specification.
22 PCIE_CAP_PRESENCE_DETECT_STATE	Presence Detect State. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R
21 PCIE_CAP_MRL_SENSOR_STATE	MRL Sensor State. For a description of this standard PCIe register field, see the PCI Express Specification.
20 PCIE_CAP_CMD_CPLD	Command Completed. For a description of this standard PCIe register field, see the PCI Express Specification.
19 PCIE_CAP_PRESENCE_DETECTED_CHANGED	Presence Detect Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
18 PCIE_CAP_MRL_SENSOR_CHANGED	MRL Sensor Changed. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP_POWER_FAULT_DETECTED	Power Fault Detected. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_ATTENTION_BUTTON_PRESSED	Attention Button Pressed. For a description of this standard PCIe register field, see the PCI Express Specification.
15-13 RSVDP_13	Reserved for future use.
12 PCIE_CAP_DLL_STATE_CHANGED_EN	Data Link Layer State Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
11	Electromechanical Interlock Control. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
PCIE_CAP_ELECTROMECH_INTERLOCK_CTRL	
10 PCIE_CAP_POWER_CONTROLLER_CTRL	Power Controller Control. For a description of this standard PCIe register field, see the PCI Express Specification.
9-8 PCIE_CAP_POWER_INDICATOR_CTRL	Power Indicator Control. For a description of this standard PCIe register field, see the PCI Express Specification.
7-6 PCIE_CAP_ATTENTION_INDICATOR_CTRL	Attention Indicator Control. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_HOT_PLUG_INT_EN	Hot Plug Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
4 PCIE_CAP_CMD_CPL_INT_EN	Command Completed Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Write value is gated with PCIE_CAP_NO_CMD_CPL_SUPPORT field in SLOT_CAPABILITIES_REG. Note: The access attributes of this field are as follows: - Dbi: SLOT_CAPABILITIES_REG.PCIE_CAP_NO_CMD_CPL_SUPPORT ? RO : RW
3 PCIE_CAP_PRESENCE_DETECT_CHANGE_EN	Presence Detect Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_MRL_SENSOR_CHANGED_EN	MRL Sensor Changed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_POWER_FAULT_DETECTED_EN	Power Fault Detected Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_ATTENTION_BUTTON_PRESSED_EN	Attention Button Pressed Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.31 Root Control and Capabilities Register. (ROOT_CONTROL_ROOT_CAPABILITIES_REG)

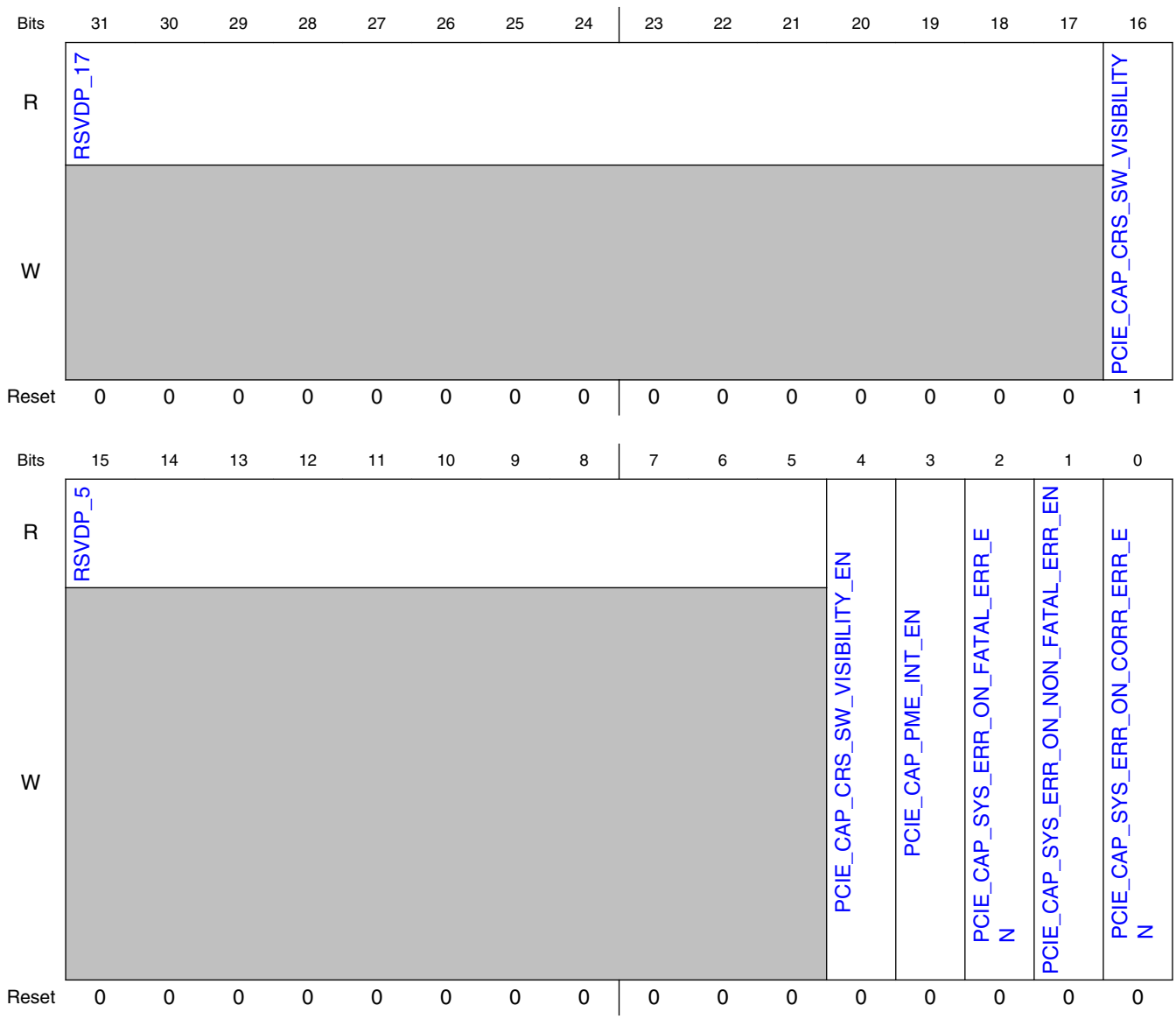
11.3.3.1.31.1 Offset

Register	Offset
ROOT_CONTROL_ROOT_CAPABILITIES_REG	8Ch

11.3.3.1.31.2 Function

Root Control and Capabilities Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.31.3 Diagram



11.3.3.1.31.4 Fields

Field	Function
31-17 RSVDP_17	Reserved for future use.
16 PCIE_CAP_CRS_SW_VISIBILITY	CRS Software Visibility Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W (Sticky) else R (Sticky) Note: This register field is sticky.
15-5 RSVDP_5	Reserved for future use.
4 PCIE_CAP_CRS_SW_VISIBILITY_EN	Configuration Request Retry Status (CRS) Software Visibility Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: ROOT_CONTROL_ROOT_CAPABILITIES_REG.PCIE_CAP_CRS_SW_VISIBILITY ? RW : RO
3 PCIE_CAP_PME_INT_EN	PME Interrupt Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 PCIE_CAP_SYS_ERR_ON_FATAL_ERR_EN	System Error on Fatal Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 PCIE_CAP_SYS_ERR_ON_NON_FATAL_ERR_EN	System Error on Non-fatal Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 PCIE_CAP_SYS_ERR_ON_CORR_ERR_EN	System Error on Correctable Error Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.32 Root Status Register. (ROOT_STATUS_REG)

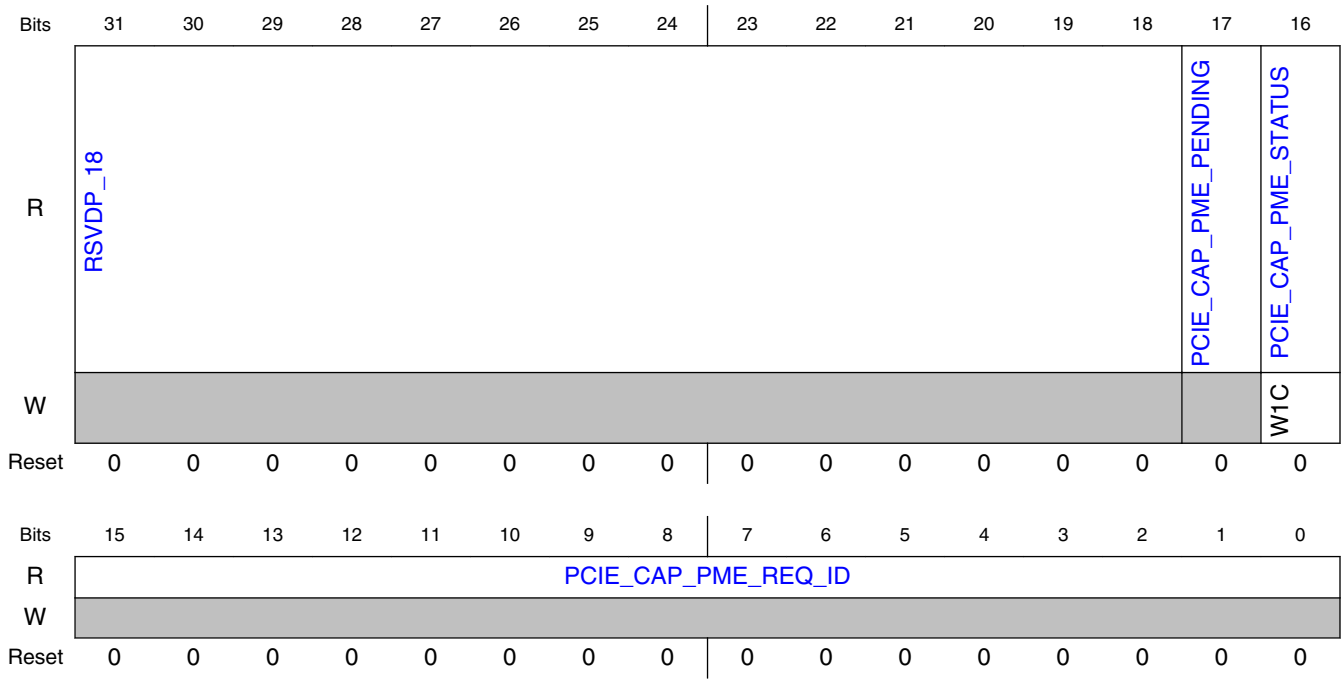
11.3.3.1.32.1 Offset

Register	Offset
ROOT_STATUS_REG	90h

11.3.3.1.32.2 Function

Root Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.32.3 Diagram



11.3.3.1.32.4 Fields

Field	Function
31-18 RSVDP_18	Reserved for future use.
17 PCIE_CAP_PME_PENDING	PME Pending. For a description of this standard PCIe register field, see the PCI Express Specification.
16 PCIE_CAP_PME_STATUS	PME Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15-0 PCIE_CAP_PME_REQ_ID	PME Requester ID. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.33 Device Capabilities 2 Register. (DEVICE_CAPABILITIES2_REG)

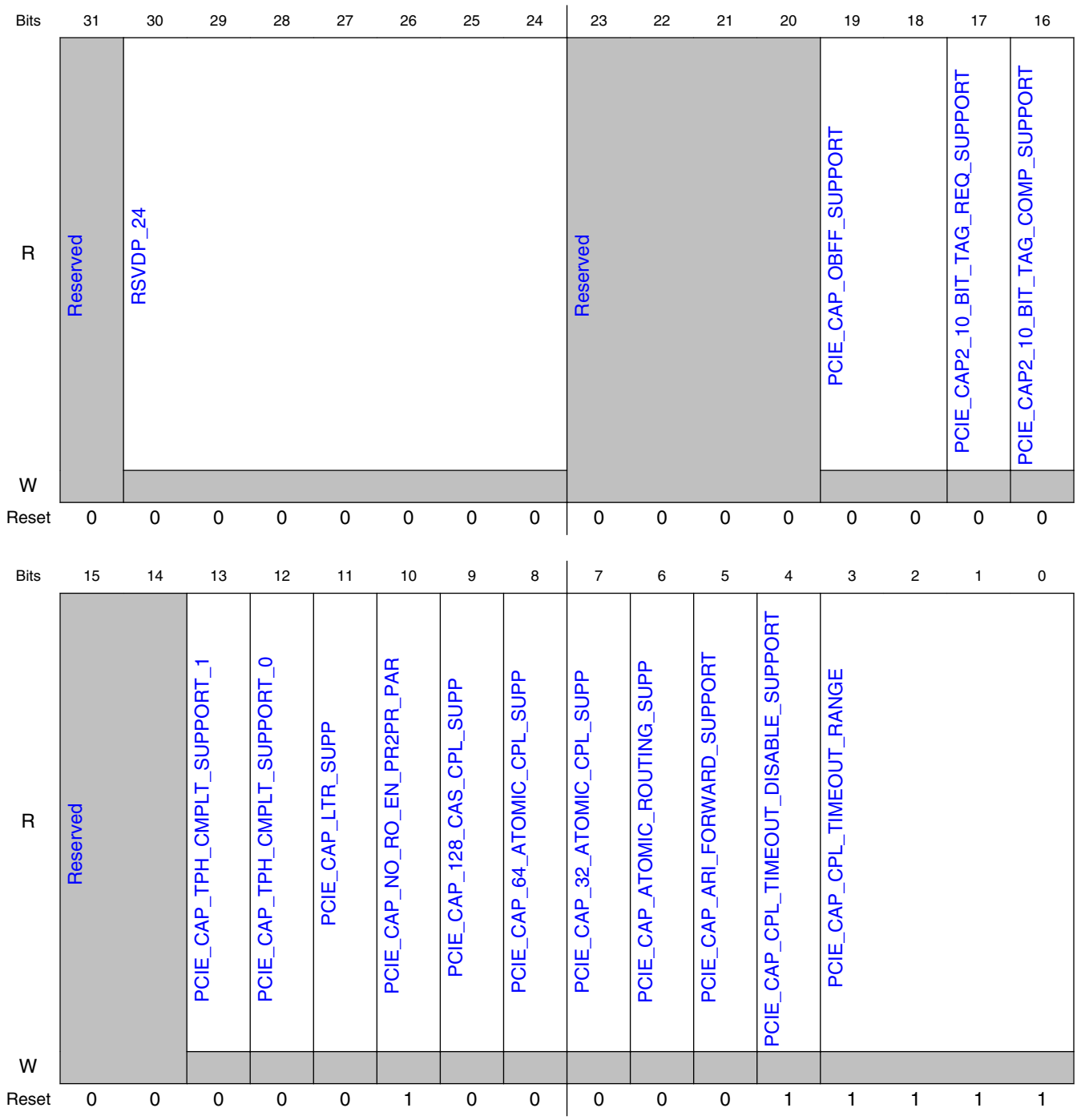
11.3.3.1.33.1 Offset

Register	Offset
DEVICE_CAPABILITIES2_REG	94h

11.3.3.1.33.2 Function

Device Capabilities 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.33.3 Diagram



11.3.3.1.33.4 Fields

Field	Function
31	Reserved.

Table continues on the next page...

Field	Function
—	
30-24 RSVDP_24	Reserved for future use.
23-20 —	Reserved.
19-18 PCIE_CAP_OBFF_SUPPORT	(OBFF) Optimized Buffer Flush/fill Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
17 PCIE_CAP2_10_BIT_TAG_REQ_SUPPORT	10-Bit Tag Requester Supported. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.
16 PCIE_CAP2_10_BIT_TAG_COMP_SUPPORT	10-Bit Tag Completer Supported. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.
15-14 —	Reserved.
13 PCIE_CAP_TPH_CMPLT_SUPPORT_1	TPH Completer Supported Bit 1. For a description of this standard PCIe register field, see the PCI Express Specification.
12 PCIE_CAP_TPH_CMPLT_SUPPORT_0	TPH Completer Supported Bit 0. For a description of this standard PCIe register field, see the PCI Express Specification.
11 PCIE_CAP_LTR_SUPP	LTR Mechanism Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
10 PCIE_CAP_NO_RO_EN_PR2PR_PAR	No Relaxed Ordering Enabled PR-PR Passing. For a description of this standard PCIe register field, see the PCI Express Specification.
9 PCIE_CAP_128_CAS_CPL_SUPP	128 Bit CAS Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
8 PCIE_CAP_64_ATOMIC_CPL_SUPP	64 Bit AtomicOp Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
7 PCIE_CAP_32_ATOMIC_CPL_SUPP	32 Bit AtomicOp Completer Supported. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
6 PCIE_CAP_ATOMIC_ROUTING_SUPP	Atomic Operation Routing Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
5 PCIE_CAP_ARI_FORWARD_SUPPORT	ARI Forwarding Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
4 PCIE_CAP_CPL_TIMEOUT_DISABLE_SUPPORT	Completion Timeout Disable Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 PCIE_CAP_CPL_TIMEOUT_RANGE	Completion Timeout Ranges Supported. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.34 Device Control 2 and Status 2 Register. (DEVICE_CONTROL2_DEVICE_STATUS2_REG)

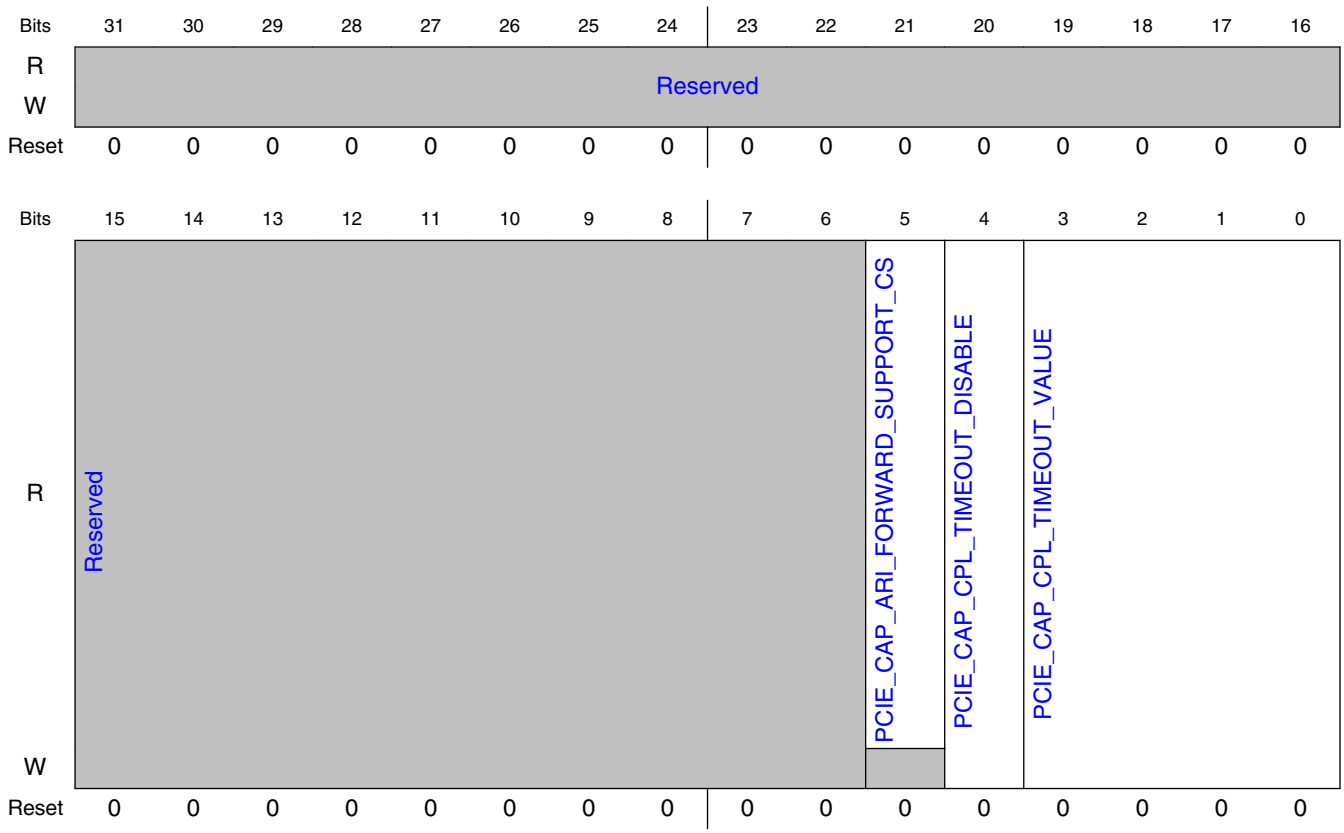
11.3.3.1.34.1 Offset

Register	Offset
DEVICE_CONTROL2_DEVICE_STATUS2_REG	98h

11.3.3.1.34.2 Function

Device Control 2 and Status 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.34.3 Diagram



11.3.3.1.34.4 Fields

Field	Function
31-6 —	Reserved.
5 PCIE_CAP_ARI_FORWARD_SUPPORT_CS	ARI Forwarding Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
4 PCIE_CAP_CPL_TIMEOUT_DISABLE	Completion Timeout Disable. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 PCIE_CAP_CPL_TIMEOUT_VALUE	Completion Timeout Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.35 Link Capabilities 2 Register. (LINK_CAPABILITIES2_REG)

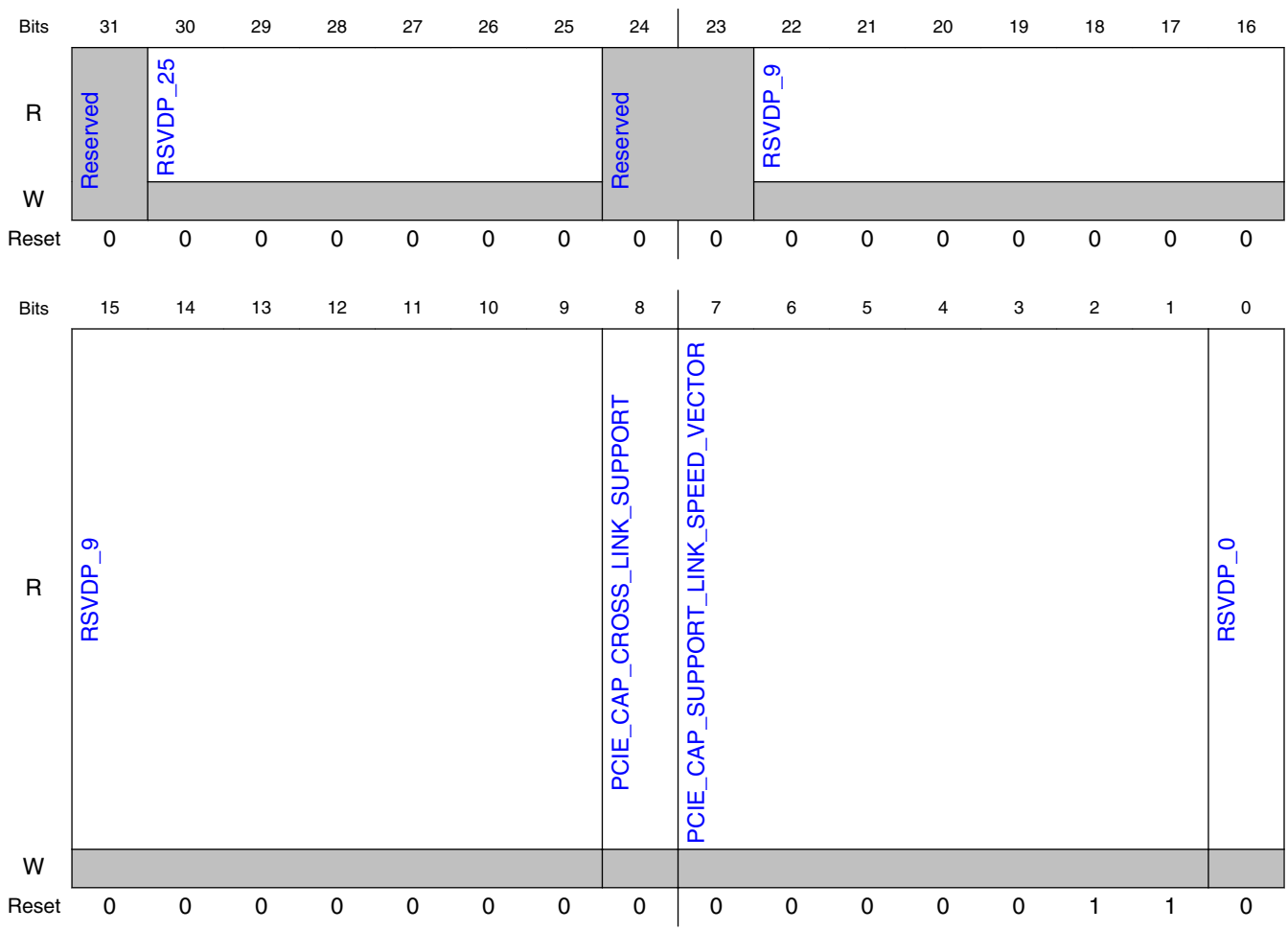
11.3.3.1.35.1 Offset

Register	Offset
LINK_CAPABILITIES2_REG	9Ch

11.3.3.1.35.2 Function

Link Capabilities 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.35.3 Diagram



11.3.3.1.35.4 Fields

Field	Function
31 —	Reserved.
30-25 RSVDP_25	Reserved for future use.
24-23 —	Reserved.
22-9 RSVDP_9	Reserved for future use.
8 PCIE_CAP_CROSS_LINK_SUPPORT	Cross Link Supported. For a description of this standard PCIe register field, see the PCI Express Specification.
7-1 PCIE_CAP_SUPPORT_LINK_SPEED_VECTOR	Supported Link Speeds Vector. For a description of this standard PCIe register field, see the PCI Express Specification. This field has a default of (PCIE_CAP_MAX_LINK_SPEED == 0100) ? 0001111 : (PCIE_CAP_MAX_LINK_SPEED == 0011) ? 0000111 : (PCIE_CAP_MAX_LINK_SPEED == 0010) ? 0000011 : 0000001 where PCIE_CAP_MAX_LINK_SPEED is a field in the LINK_CAPABILITIES_REG register.
0 RSVDP_0	Reserved for future use.

11.3.3.1.36 Link Control 2 and Status 2 Register. (LINK_CONTROL2_LINK_STATUS2_REG)

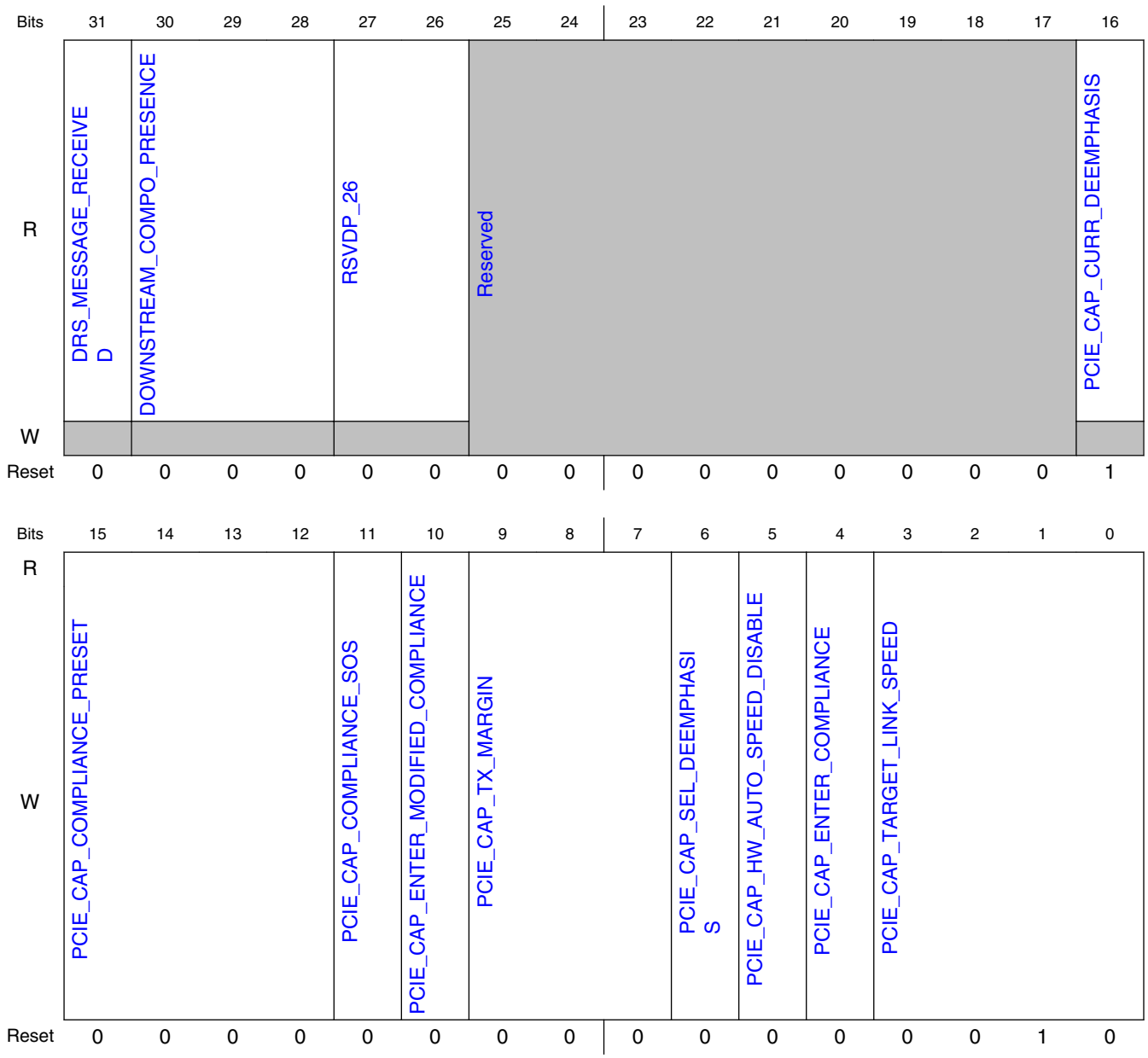
11.3.3.1.36.1 Offset

Register	Offset
LINK_CONTROL2_LINK_STATUS2_REG	A0h

11.3.3.1.36.2 Function

Link Control 2 and Status 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.36.3 Diagram



11.3.3.1.36.4 Fields

Field	Function
31 DRS_MESSAGE_RECEIVED	DRS Message Received. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0. Note: The access attributes of this field are as follows: - Dbi: RW1C
30-28	Downstream Component Presence. For a description of this standard PCIe register field, see the PCI Express Base Specification 4.0.

Table continues on the next page...

Field	Function
DOWNSTREAM_COMPO_PRESENCE	
27-26 RSVDP_26	Reserved for future use.
25-17 —	Reserved.
16 PCIE_CAP_CURR_DEEMPHASIS	Current De-emphasis Level. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode this register is always 0x0. In C-PCIe mode, its contents are derived by sampling the PIPE
15-12 PCIE_CAP_COMPLIANCE_PRESET	Sets Compliance Preset/De-emphasis for 5 GT/s and 8 GT/s. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
11 PCIE_CAP_COMPLIANCE_SOS	Sets Compliance Skip Ordered Sets transmission. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
10 PCIE_CAP_ENTER_MODIFIED_COMPLIANCE	Enter Modified Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
9-7 PCIE_CAP_TX_MARGIN	Controls Transmit Margin for Debug or Compliance. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 PCIE_CAP_SEL_DEEMPHASIS	Controls Selectable De-emphasis for 5 GT/s. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
5 PCIE_CAP_HW_AUTO_SPEED_DISABLE	Hardware Autonomous Speed Disable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
4 PCIE_CAP_ENTER_COMPLIANCE	Enter Compliance Mode. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 PCIE_CAP_TARGET_LINK_SPEED	Target Link Speed. For a description of this standard PCIe register field, see the PCI Express Specification. In M-PCIe mode, the contents of this field are derived from other registers. Note: This register field is sticky.

11.3.3.1.37 Advanced Error Reporting Extended Capability Header. (AER_EXT_CAP_HDR_OFF)

11.3.3.1.37.1 Offset

Register	Offset
AER_EXT_CAP_HDR_OFF	100h

11.3.3.1.37.2 Function

Advanced Error Reporting Extended Capability Header. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NEXT_OFFSET												CAP_VERSION			
Reset	0	0	0	1	0	1	0	0	1	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CAP_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

11.3.3.1.37.4 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19-16 CAP_VERSION	Capability Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 CAP_ID	AER Extended Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.3.3.1.38 Uncorrectable Error Status Register. (UNCORR_ERR_STATUS_OFF)

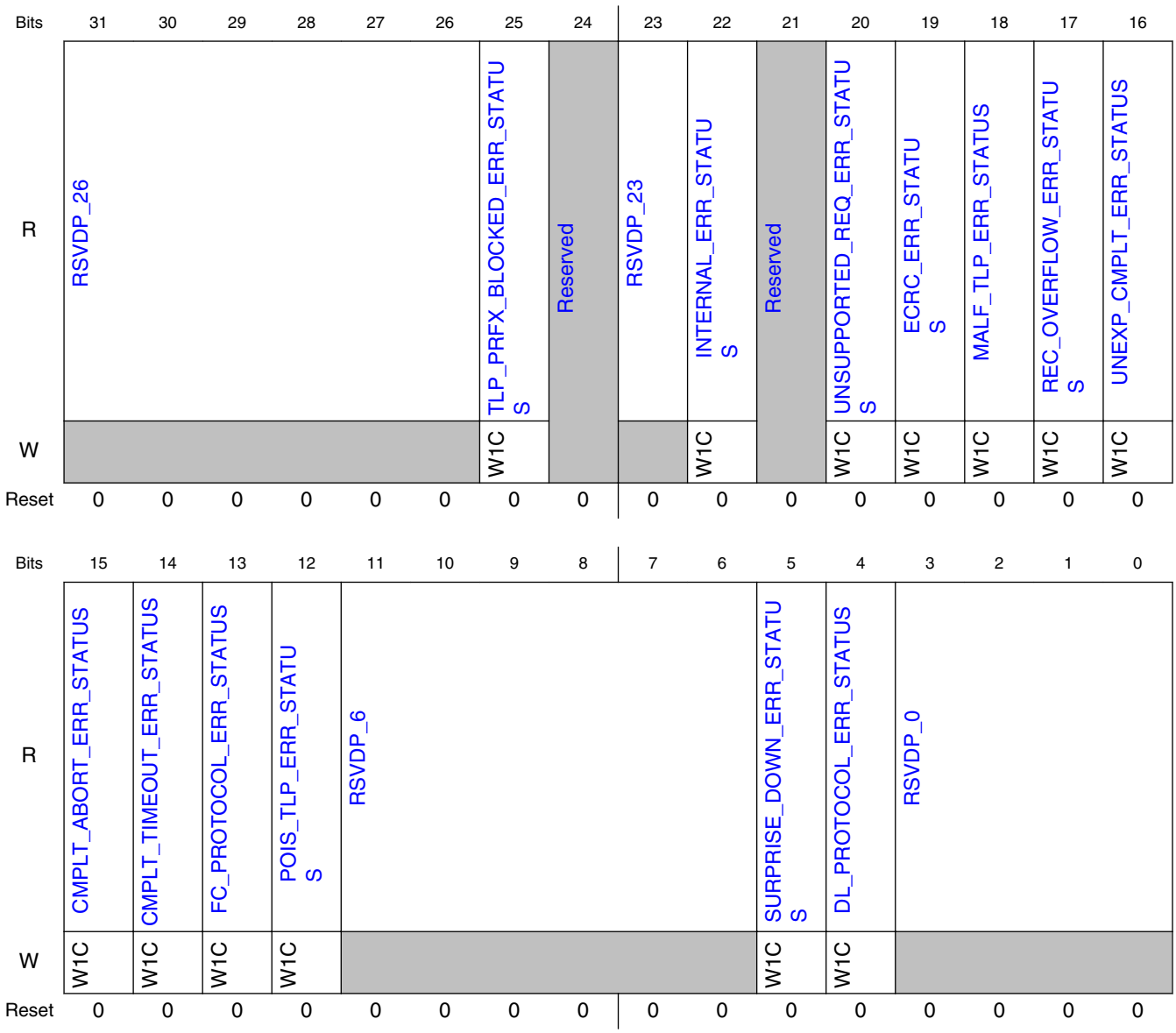
11.3.3.1.38.1 Offset

Register	Offset
UNCORR_ERR_STATUS_OFF	104h

11.3.3.1.38.2 Function

Uncorrectable Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.38.3 Diagram



11.3.3.1.38.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25 TLP_PRFX_BLOCKED_ERR_STATUS	TLP Prefix Blocked Error Status. For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported.
24	Reserved.

Table continues on the next page...

Field	Function
—	
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_STATUS	Uncorrectable Internal Error Status. For a description of this standard PCIe register field, see the PCI Express Specification. The controller sets this bit when your application asserts app_err_bus[9]. It does not set this bit when it detects internal uncorrectable internal errors such as parity and ECC failures. You should use the outputs from these errors to drive the app_err_bus[9] input. For more details, see the "Data Integrity (Wire, Datapath, and RAM Protection)" section in the Databook.
21 —	Reserved.
20 UNSUPPORTED_REQ_ERR_STATUS	Unsupported Request Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
19 ECRC_ERR_STATUS	ECRC Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
18 MALF_TLP_ERR_STATUS	Malformed TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
17 REC_OVERFLOW_ERR_STATUS	Receiver Overflow Status. For a description of this standard PCIe register field, see the PCI Express Specification.
16 UNEXP_CMPLT_ERR_STATUS	Unexpected Completion Status. For a description of this standard PCIe register field, see the PCI Express Specification.
15 CMPLT_ABORT_ERR_STATUS	Completer Abort Status. For a description of this standard PCIe register field, see the PCI Express Specification.
14 CMPLT_TIMEOUT_ERR_STATUS	Completion Timeout Status. For a description of this standard PCIe register field, see the PCI Express Specification.
13 FC_PROTOCOL_ERR_STATUS	Flow Control Protocol Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
12 POIS_TLP_ERR_STATUS	Poisoned TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
11-6 RSVDP_6	Reserved for future use.
5	Surprise Down Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
SURPRISE_DOWN_ERR_STATUS	
4 DL_PROTOCOL_ERR_STATUS	Data Link Protocol Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
3-0 RSVDP_0	Reserved for future use.

11.3.3.1.39 Uncorrectable Error Mask Register. (UNCORR_ERR_MASK_OFF)

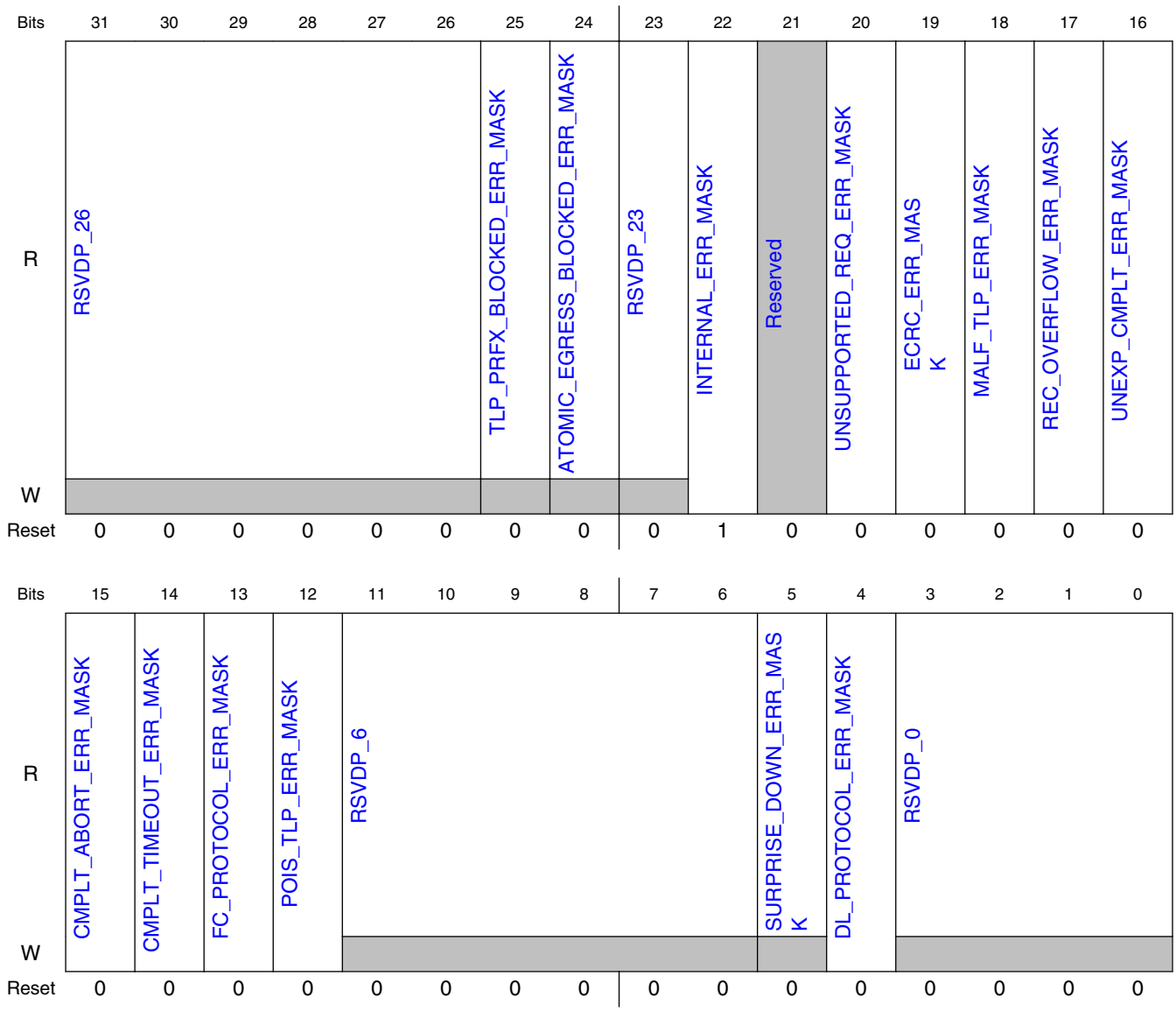
11.3.3.1.39.1 Offset

Register	Offset
UNCORR_ERR_MASK_OFF	108h

11.3.3.1.39.2 Function

Uncorrectable Error Mask Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.39.3 Diagram



11.3.3.1.39.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25 TLP_PRFX_BLOCKED_ERR_MASK	TLP Prefix Blocked Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
24 ATOMIC_EGRESS_BLOCKED_ERR_MASK	AtomicOp Egress Block Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_MASK	Uncorrectable Internal Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
21 —	Reserved.
20 UNSUPPORTED_REQ_ERR_MASK	Unsupported Request Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
19 ECRC_ERR_MASK	ECRC Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18 MALF_TLP_ERR_MASK	Malformed TLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17 REC_OVERFLOW_ERR_MASK	Receiver Overflow Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
16 UNEXP_CMPLT_ERR_MASK	Unexpected Completion Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15 CMPLT_ABORT_ERR_MASK	Completer Abort Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CMPLT_TIMEOUT_ERR_MASK	Completion Timeout Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 FC_PROTOCOL_ERR_MASK	Flow Control Protocol Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
12 POIS_TLP_ERR_MASK	Poisoned TLP Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
11-6 RSVDP_6	Reserved for future use.
5	Surprise Down Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi:

Table continues on the next page...

Field	Function
SURPRISE_DOWN_ERR_MASK	LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4 DL_PROTOCOL_ERR_MASK	Data Link Protocol Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 RSVDP_0	Reserved for future use.

11.3.3.1.40 Uncorrectable Error Severity Register. (UNCORR_ERR_SEV_OFF)

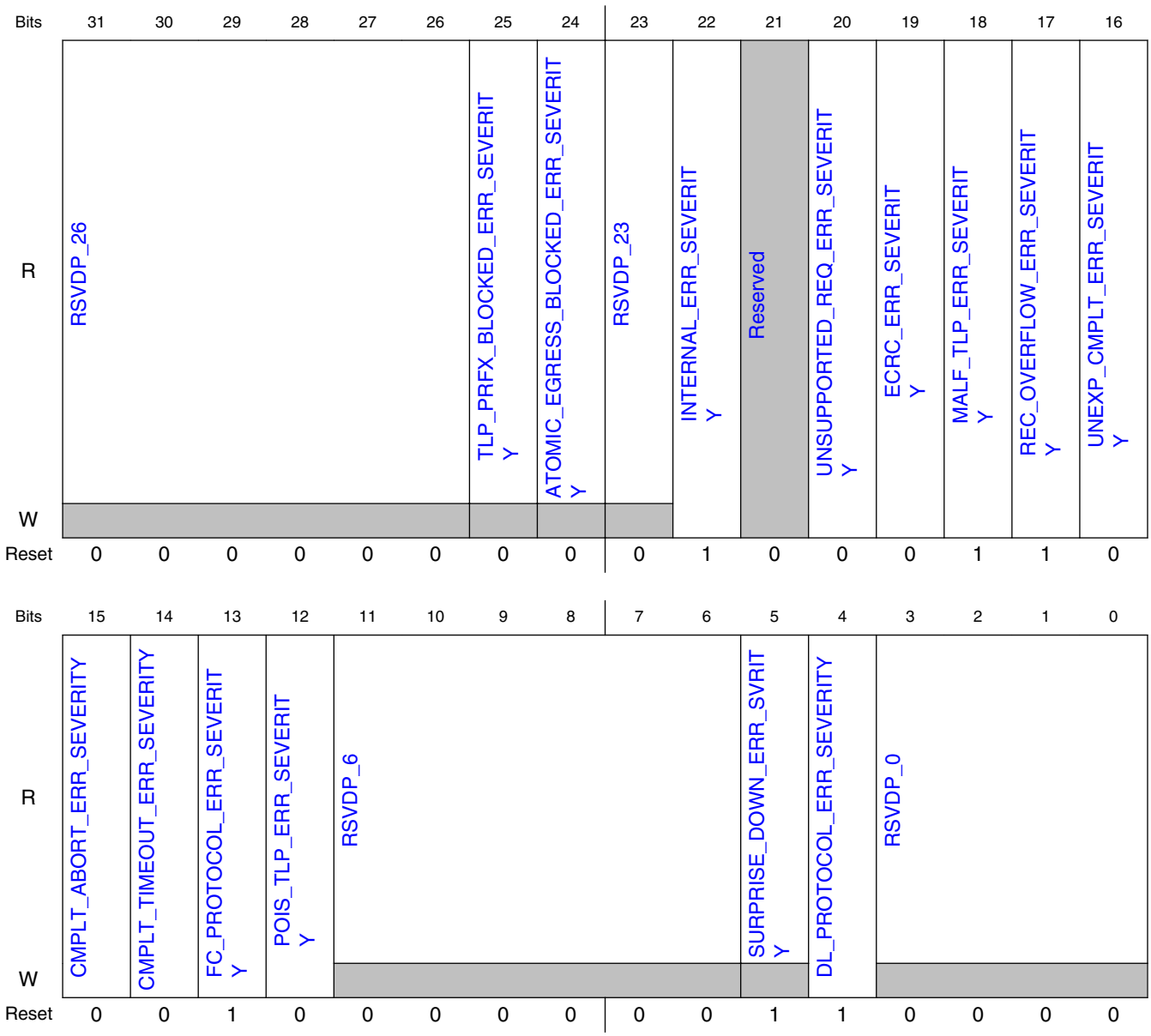
11.3.3.1.40.1 Offset

Register	Offset
UNCORR_ERR_SEV_OFF	10Ch

11.3.3.1.40.2 Function

Uncorrectable Error Severity Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.40.3 Diagram



11.3.3.1.40.4 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25 TLP_PRFX_BLOCKED_ERR_SEVERITY	TLP Prefix Blocked Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: Not supported. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
24 ATOMIC_EGRESS_BLOCKED_ERR_SEVERITY	AtomicOp Egress Blocked Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
23 RSVDP_23	Reserved for future use.
22 INTERNAL_ERR_SEVERITY	Uncorrectable Internal Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
21 —	Reserved.
20 UNSUPPORTED_REQ_ERR_SEVERITY	Unsupported Request Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
19 ECRC_ERR_SEVERITY	ECRC Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18 MALF_TLP_ERR_SEVERITY	Malformed TLP Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
17 REC_OVERFLOW_ERR_SEVERITY	Receiver Overflow Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
16 UNEXP_CMPLT_ERR_SEVERITY	Unexpected Completion Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15 CMPLT_ABORT_ERR_SEVERITY	Completer Abort Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CMPLT_TIMEOUT_ERR_SEVERITY	Completion Timeout Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 FC_PROTOCOL_ERR_SEVERITY	Flow Control Protocol Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
12 POIS_TLP_ERR_SEVERITY	Poisoned TLP Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
11-6 RSVDP_6	Reserved for future use.
5 SURPRISE_DOWN_ERR_SVRITY	Surprise Down Error Severity (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: LINK_CAPABILITIES_REG.PCIE_CAP_SURPRISE_DOWN_ERR_REP_CAP ? RW : RO Note: This register field is sticky.
4 DL_PROTOCOL_ERR_SEVERITY	Data Link Protocol Error Severity. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
3-0 RSVDP_0	Reserved for future use.

11.3.3.1.41 Correctable Error Status Register. (CORR_ERR_STATUS_OFF)

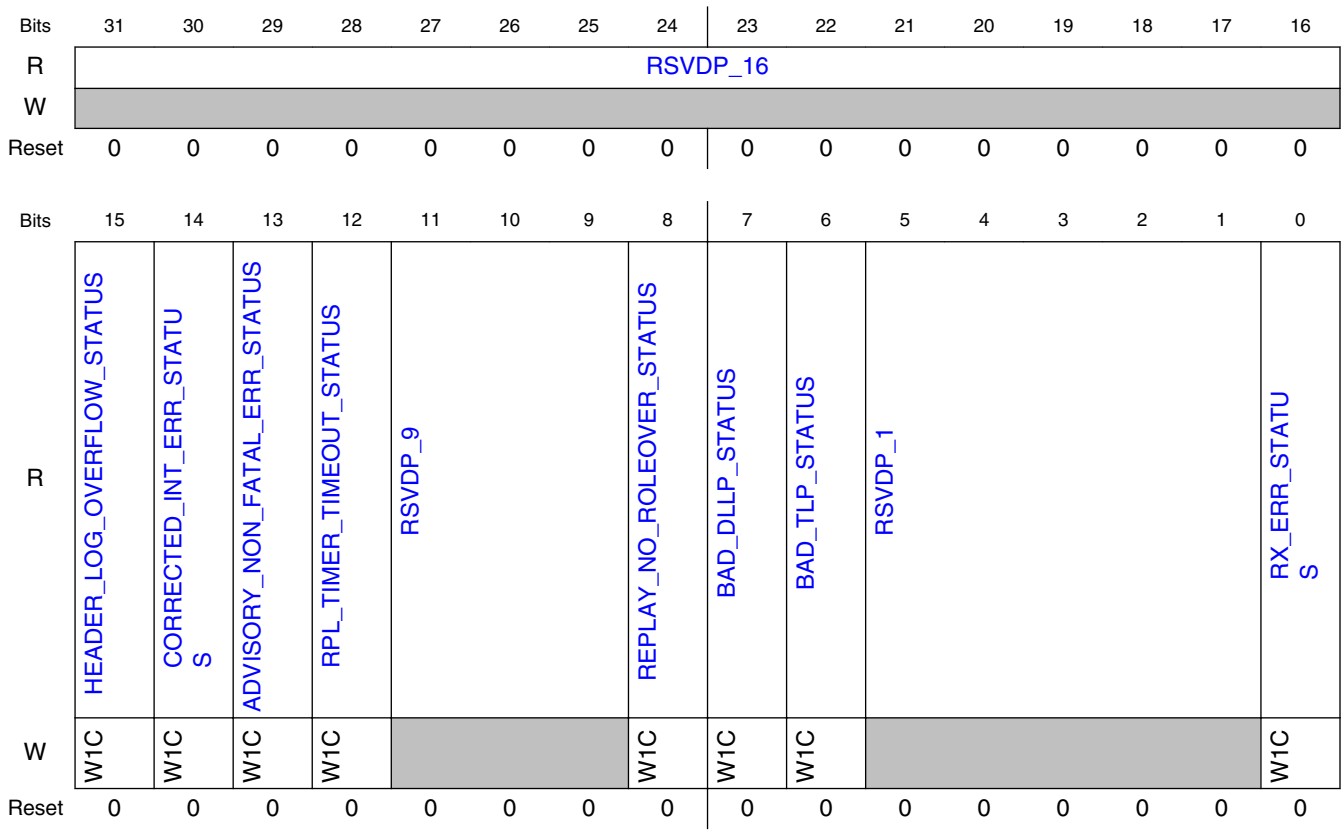
11.3.3.1.41.1 Offset

Register	Offset
CORR_ERR_STATUS_OFF	110h

11.3.3.1.41.2 Function

Correctable Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.41.3 Diagram



11.3.3.1.41.4 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15 HEADER_LOG_OVERFLOW_STATUS	Header Log Overflow Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.
14 CORRECTED_INT_ERR_STATUS	Corrected Internal Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.
13 ADVISORY_NON_FATAL_ERR_STATUS	Advisory Non-Fatal Error Status. For a description of this standard PCIe register field, see the PCI Express Specification.
12	Replay Timer Timeout Status. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
RPL_TIMER_TIMEOUT_STATUS	
11-9 RSVDP_9	Reserved for future use.
8 REPLAY_NO_ROLEOVER_STATUS	REPLAY_NUM Rollover Status. For a description of this standard PCIe register field, see the PCI Express Specification.
7 BAD_DLLP_STATUS	Bad DLLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
6 BAD_TLP_STATUS	Bad TLP Status. For a description of this standard PCIe register field, see the PCI Express Specification.
5-1 RSVDP_1	Reserved for future use.
0 RX_ERR_STATUS	Receiver Error Status (Optional). For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.42 Correctable Error Mask Register. (CORR_ERR_MASK_OFF)

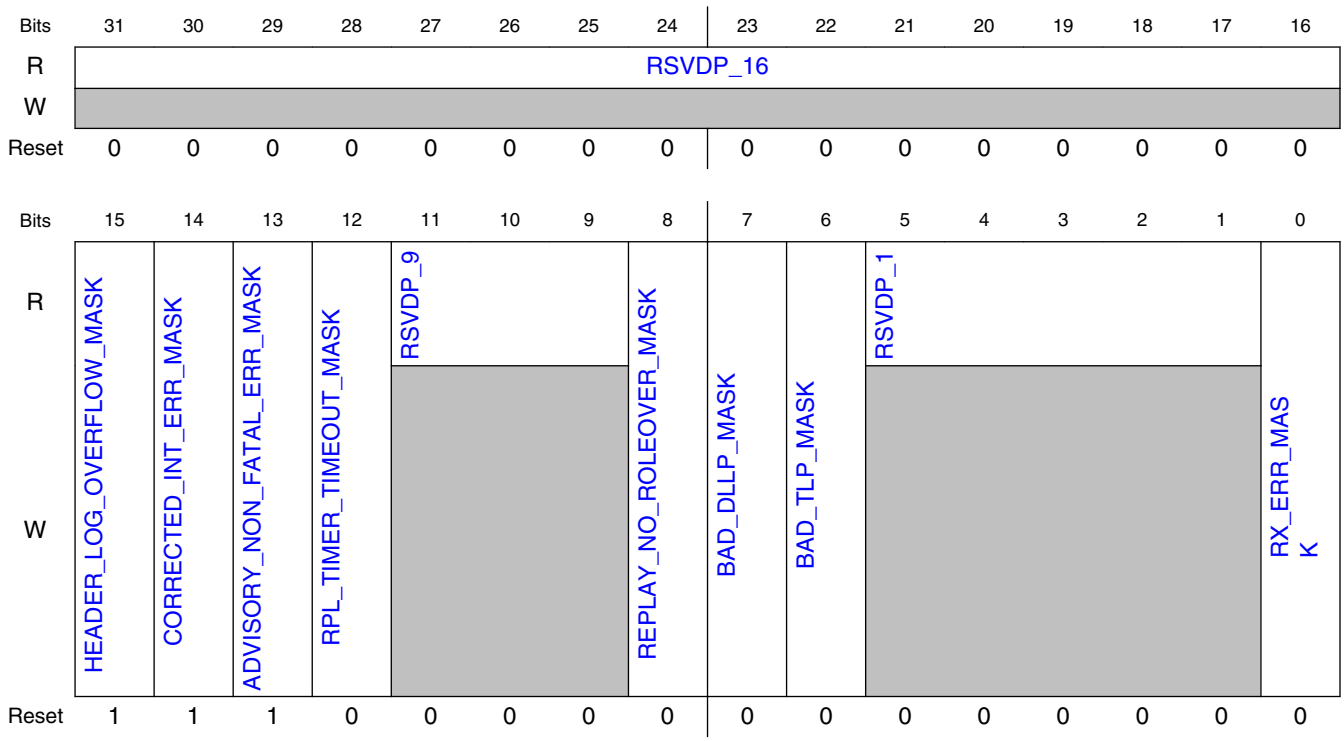
11.3.3.1.42.1 Offset

Register	Offset
CORR_ERR_MASK_OFF	114h

11.3.3.1.42.2 Function

Correctable Error Mask Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.42.3 Diagram



11.3.3.1.42.4 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15 HEADER_LOG_OVERFLOW_MASK	Header Log Overflow Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
14 CORRECTED_INT_ERR_MASK	Corrected Internal Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
13 ADVISORY_NON_FATAL_ERR_MASK	Advisory Non-Fatal Error Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
12 RPL_TIMER_TIMEOUT_MASK	Replay Timer Timeout Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
11-9 RSVDP_9	Reserved for future use.

Table continues on the next page...

Field	Function
8 REPLAY_NO_ROLEOVER_MASK	REPLAY_NUM Rollover Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7 BAD_DLLP_MASK	Bad DLLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 BAD_TLP_MASK	Bad TLP Mask. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
5-1 RSVDP_1	Reserved for future use.
0 RX_ERR_MASK	Receiver Error Mask (Optional). For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.43 Advanced Error Capabilities and Control Register. (ADV_ERR_CAP_CTRL_OFF)

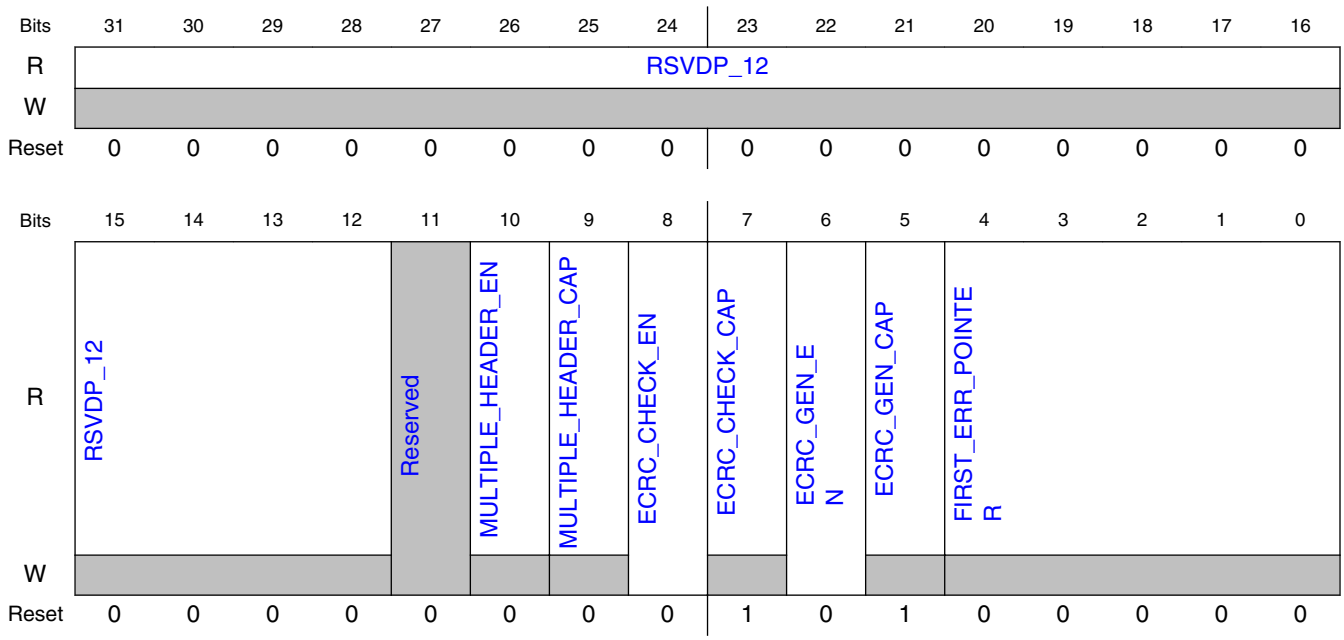
11.3.3.1.43.1 Offset

Register	Offset
ADV_ERR_CAP_CTRL_OFF	118h

11.3.3.1.43.2 Function

Advanced Error Capabilities and Control Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.43.3 Diagram



11.3.3.1.43.4 Fields

Field	Function
31-12 RSVDP_12	Reserved for future use.
11 —	Reserved.
10 MULTIPLE_HEADER_EN	Multiple Header Recording Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
9 MULTIPLE_HEADER_CAP	Multiple Header Recording Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
8 ECRC_CHECK_EN	ECRC Check Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7 ECRC_CHECK_CAP	ECRC Check Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
6 ECRC_GEN_EN	ECRC Generation Enable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
5	ECRC Generation Capable. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
ECRC_GEN_CAP	
4-0 FIRST_ERR_POINTER	First Error Pointer. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.44 Header Log Register 0. (HDR_LOG_0_OFF)

11.3.3.1.44.1 Offset

Register	Offset
HDR_LOG_0_OFF	11Ch

11.3.3.1.44.2 Function

Header Log Register 0. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FIRST_DWORD_FOURTH_BYTE								FIRST_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FIRST_DWORD_SECOND_BYTE								FIRST_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.44.4 Fields

Field	Function
31-24 FIRST_DWORD_FOURTH_BYTE	Byte 3 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
23-16 FIRST_DWORD_THIRD_BYTE	Byte 2 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 FIRST_DWORD_SECOND_BYTE	Byte 1 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 FIRST_DWORD_FIRST_BYTE	Byte 0 of Header log register of First 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.45 Header Log Register 1. (HDR_LOG_1_OFF)

11.3.3.1.45.1 Offset

Register	Offset
HDR_LOG_1_OFF	120h

11.3.3.1.45.2 Function

Header Log Register 1. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SECOND_DWORD_FOURTH_BYTE								SECOND_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SECOND_DWORD_SECOND_BYTE								SECOND_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.45.4 Fields

Field	Function
31-24 SECOND_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 SECOND_DWORD_THIRD_BYTE	Byte 2 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 SECOND_DWORD_SECOND_BYTE	Byte 1 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 SECOND_DWORD_FIRST_BYTE	Byte 0 of Header log register of Second 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.46 Header Log Register 2. (HDR_LOG_2_OFF)

11.3.3.1.46.1 Offset

Register	Offset
HDR_LOG_2_OFF	124h

11.3.3.1.46.2 Function

Header Log Register 2. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.46.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	THIRD_DWORD_FOURTH_BYTE								THIRD_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	THIRD_DWORD_SECOND_BYTE								THIRD_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.46.4 Fields

Field	Function
31-24 THIRD_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 THIRD_DWORD_THIRD_BYTE	Byte 2 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 THIRD_DWORD_SECOND_BYTE	Byte 1 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 THIRD_DWORD_FIRST_BYTE	Byte 0 of Header log register of Third 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.47 Header Log Register 3. (HDR_LOG_3_OFF)

11.3.3.1.47.1 Offset

Register	Offset
HDR_LOG_3_OFF	128h

11.3.3.1.47.2 Function

Header Log Register 3. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	FOURTH_DWORD_FOURTH_BYTE								FOURTH_DWORD_THIRD_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FOURTH_DWORD_SECOND_BYTE								FOURTH_DWORD_FIRST_BYTE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.47.4 Fields

Field	Function
31-24 FOURTH_DWORD_FOURTH_BYTE	Byte 3 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16 FOURTH_DWORD_THIRD_BYTE	Byte 2 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-8 FOURTH_DWORD_SECOND_BYTE	Byte 1 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 FOURTH_DWORD_FIRST_BYTE	Byte 0 of Header log register of Fourth 32 bit Data Word. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.48 Root Error Command Register. (ROOT_ERR_CMD_OFF)

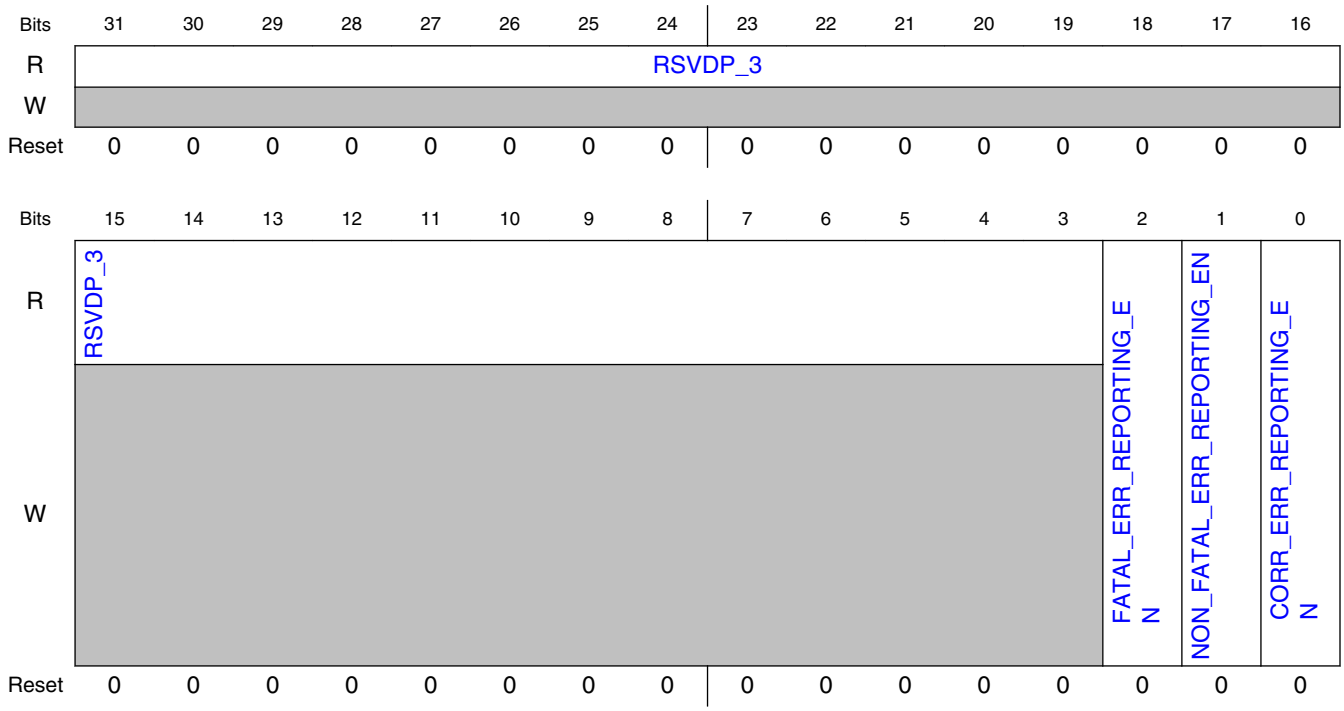
11.3.3.1.48.1 Offset

Register	Offset
ROOT_ERR_CMD_OFF	12Ch

11.3.3.1.48.2 Function

Root Error Command Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.48.3 Diagram



11.3.3.1.48.4 Fields

Field	Function
31-3 RSVDP_3	Reserved for future use.
2 FATAL_ERR_REPORTING_EN	Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 NON_FATAL_ERR_REPORTING_EN	Non-Fatal Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
0 CORR_ERR_REPORTING_EN	Correctable Error Reporting Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.49 Root Error Status Register. (ROOT_ERR_STATUS_OFF)

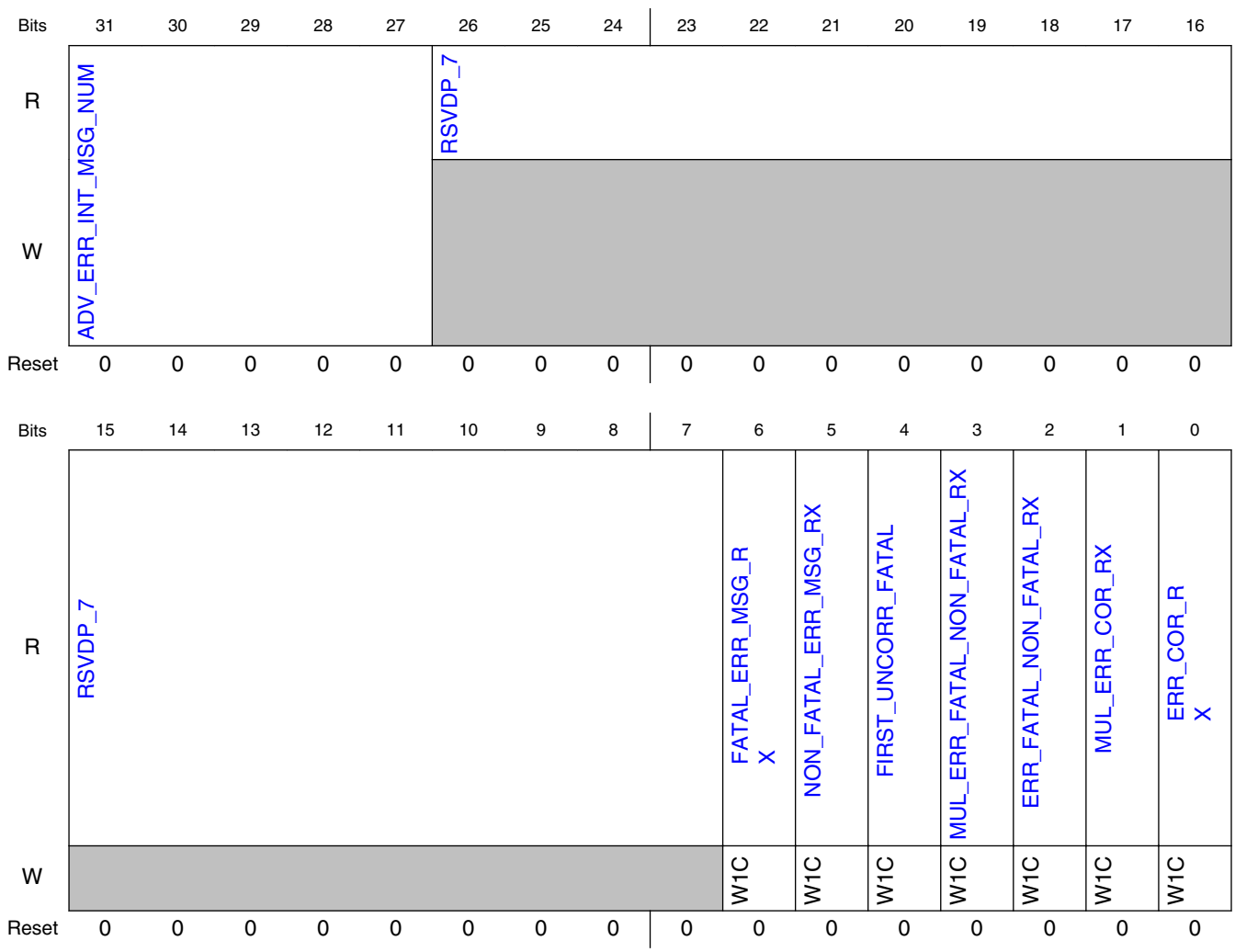
11.3.3.1.49.1 Offset

Register	Offset
ROOT_ERR_STATUS_OFF	130h

11.3.3.1.49.2 Function

Root Error Status Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.49.3 Diagram



11.3.3.1.49.4 Fields

Field	Function
31-27 ADV_ERR_INT_MSG_NUM	Advanced Error Interrupt Message Number. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
26-7 RSVDP_7	Reserved for future use.
6 FATAL_ERR_MSG_RX	One or more Fatal Error Messages Received. For a description of this standard PCIe register field, see the PCI Express Specification.
5 NON_FATAL_ERR_MSG_RX	One or more Non-Fatal Error Messages Received. For a description of this standard PCIe register field, see the PCI Express Specification.
4 FIRST_UNCOR_FATAL	First Uncorrectable Error is Fatal. For a description of this standard PCIe register field, see the PCI Express Specification.
3 MUL_ERR_FATAL_NON_FATAL_RX	Multiple Fatal or Non-Fatal Errors Received. For a description of this standard PCIe register field, see the PCI Express Specification.
2 ERR_FATAL_NON_FATAL_RX	Fatal or Non-Fatal Error Received. For a description of this standard PCIe register field, see the PCI Express Specification.
1 MUL_ERR_COR_RX	Multiple Correctable Errors Received. For a description of this standard PCIe register field, see the PCI Express Specification.
0 ERR_COR_RX	Correctable Error Received. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.50 Error Source Identification Register. (ERR_SRC_ID_OFF)

11.3.3.1.50.1 Offset

Register	Offset
ERR_SRC_ID_OFF	134h

11.3.3.1.50.2 Function

Error Source Identification Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	ERR_FATAL_NON_FATAL_SOURCE_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ERR_COR_SOURCE_ID															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.50.4 Fields

Field	Function
31-16 ERR_FATAL_NON_FATAL_SOURCE_ID	Source of Fatal/Non-Fatal Error. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
15-0 ERR_COR_SOURCE_ID	Source of Correctable Error. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.51 TLP Prefix Log Register 1. (TLP_PREFIX_LOG_1_OFF)

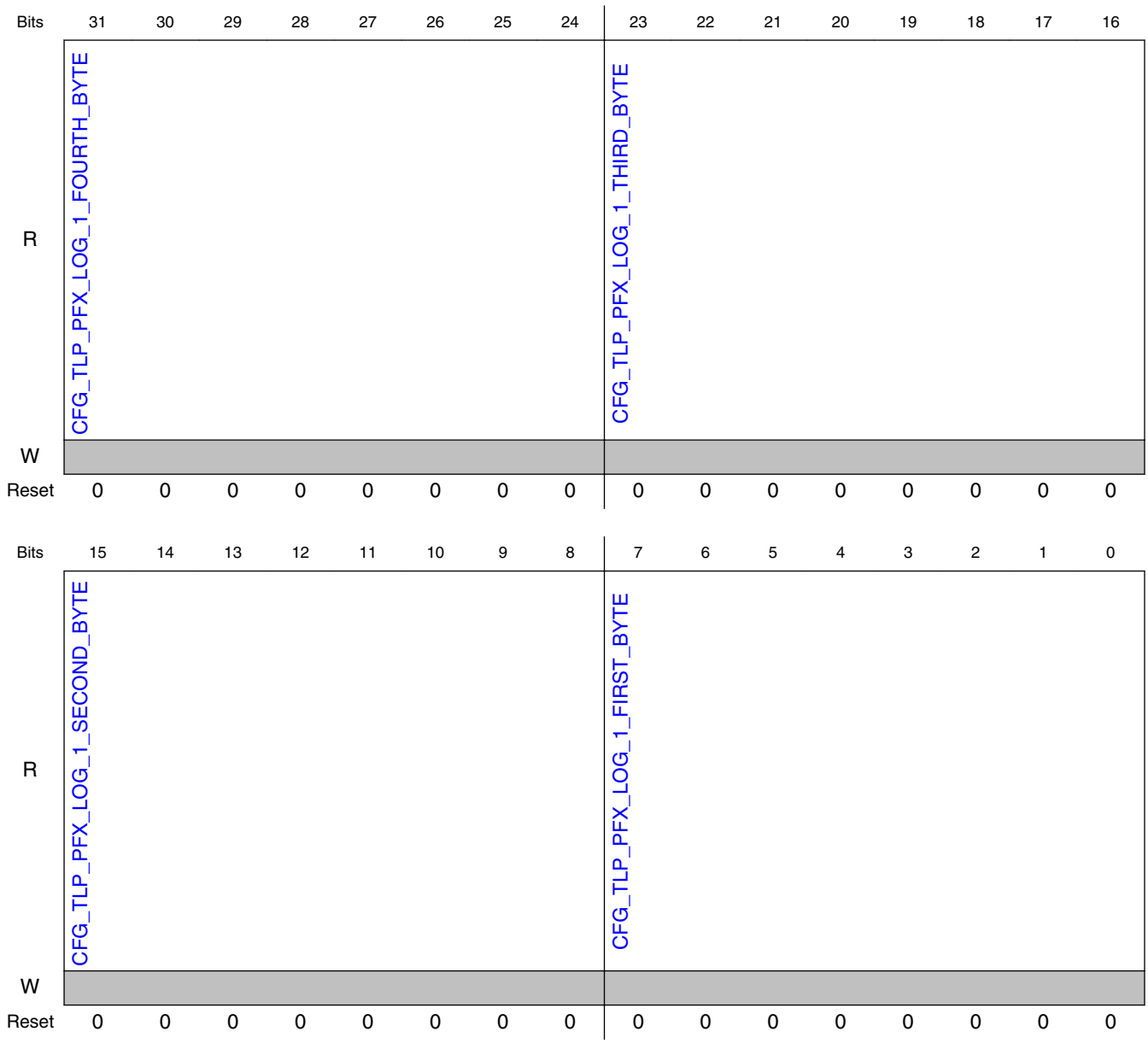
11.3.3.1.51.1 Offset

Register	Offset
TLP_PREFIX_LOG_1_OFF	138h

11.3.3.1.51.2 Function

TLP Prefix Log Register 1. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.51.3 Diagram



11.3.3.1.51.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_1_FOURTH_BYTE	Byte 3 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16	Byte 2 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
CFG_TLP_PFX_LOG_1_THIRD_BYTE	
15-8 CFG_TLP_PFX_LOG_1_SECOND_BYTE	Byte 1 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_1_FIRST_BYTE	Byte 0 of Error TLP Prefix Log 1. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.52 TLP Prefix Log Register 2. (TLP_PREFIX_LOG_2_OFF)

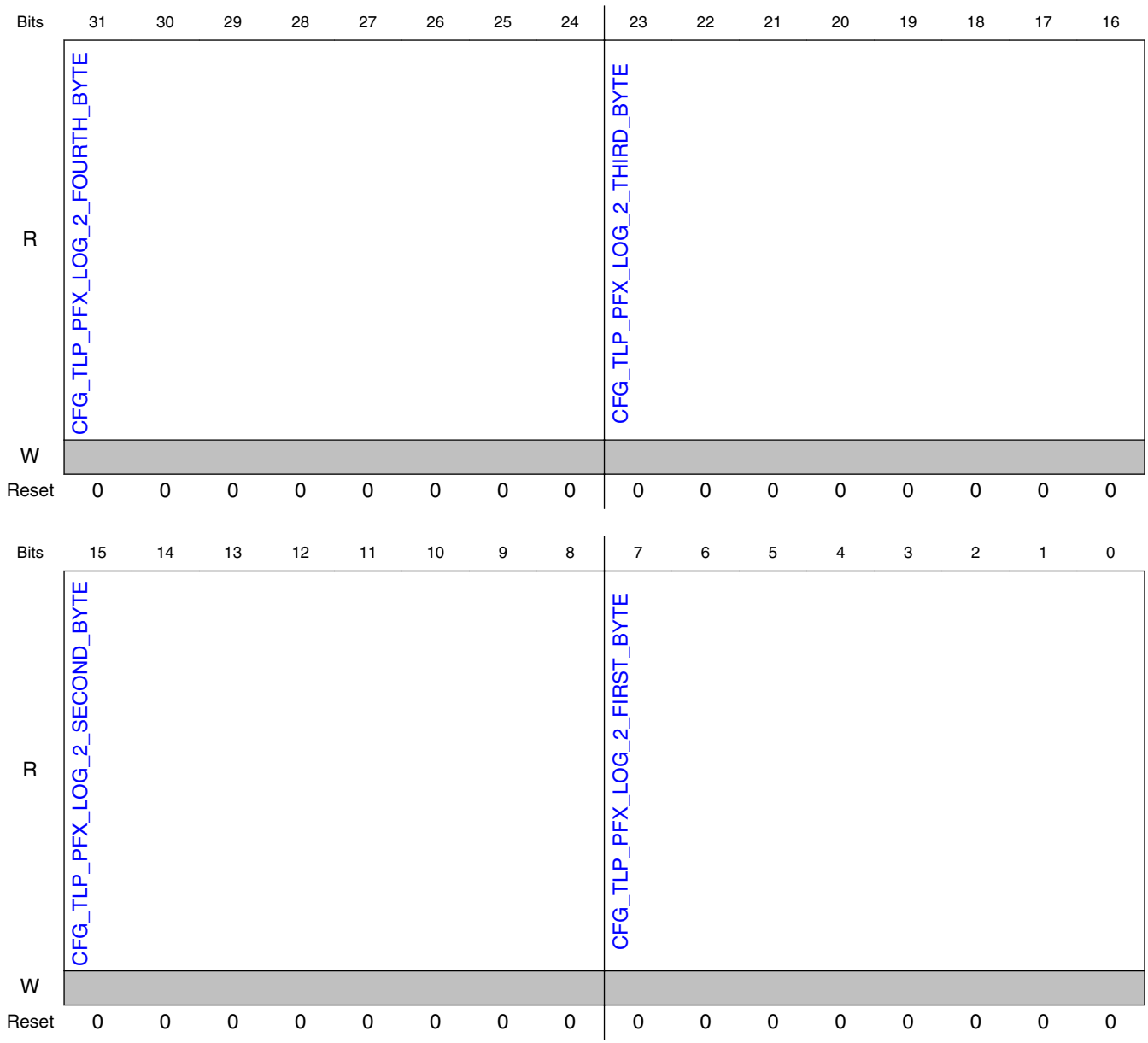
11.3.3.1.52.1 Offset

Register	Offset
TLP_PREFIX_LOG_2_OFF	13Ch

11.3.3.1.52.2 Function

TLP Prefix Log Register 2. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.52.3 Diagram



11.3.3.1.52.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_2_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16	Byte 2 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
CFG_TLP_PFX_LOG_2_THIRD_BYTE	
15-8 CFG_TLP_PFX_LOG_2_SECOND_BYTE	Byte 1 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_2_FIRST_BYTE	Byte 0 Error TLP Prefix Log 2. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.53 TLP Prefix Log Register 3. (TLP_PREFIX_LOG_3_OFF)

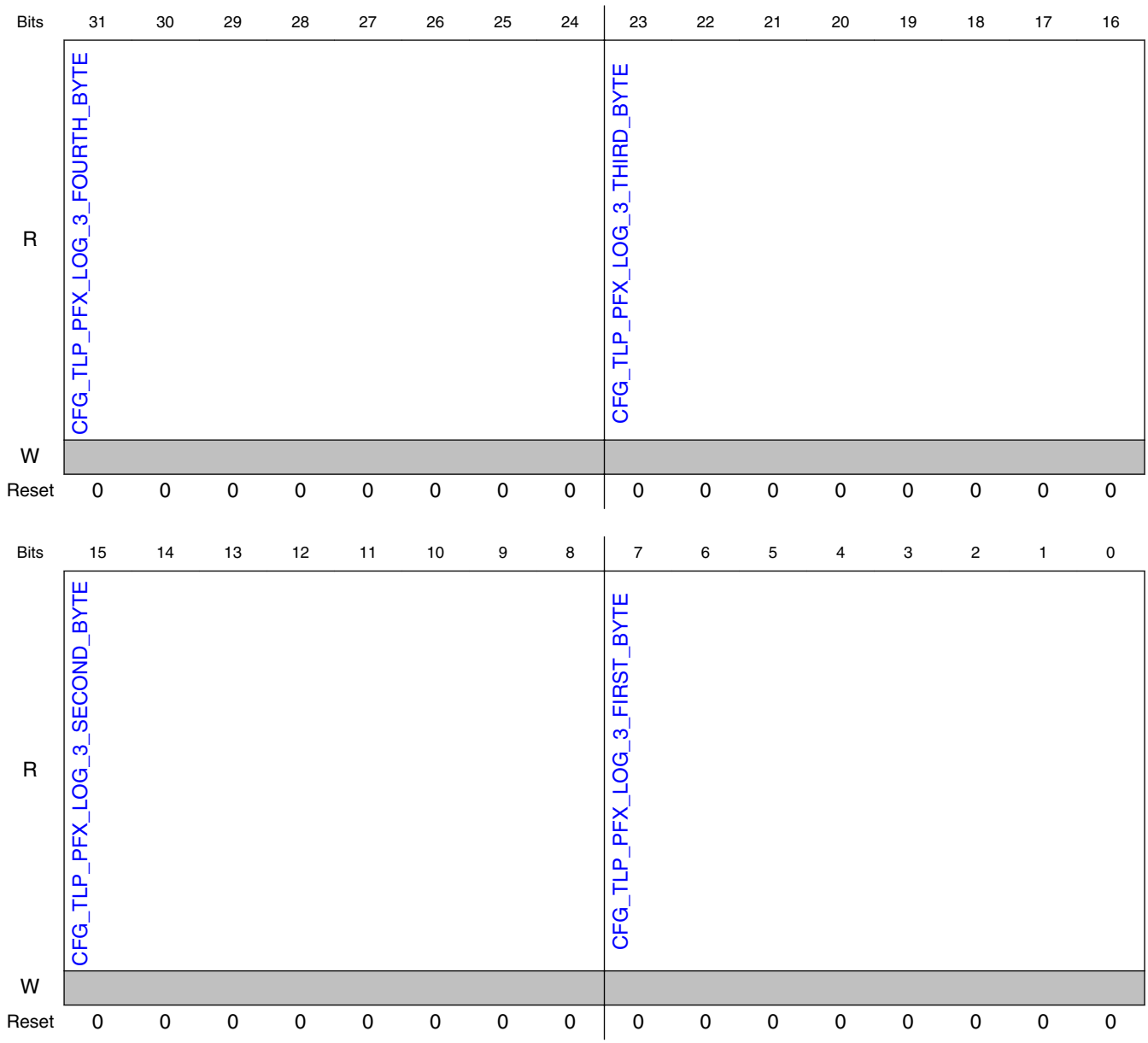
11.3.3.1.53.1 Offset

Register	Offset
TLP_PREFIX_LOG_3_OFF	140h

11.3.3.1.53.2 Function

TLP Prefix Log Register 3. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.53.3 Diagram



11.3.3.1.53.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_3_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16	Byte 2 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
CFG_TLP_PFX_LOG_3_THIRD_BYTE	
15-8 CFG_TLP_PFX_LOG_3_SECOND_BYTE	Byte 1 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_3_FIRST_BYTE	Byte 0 Error TLP Prefix Log 3. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.54 TLP Prefix Log Register 4. (TLP_PREFIX_LOG_4_OFF)

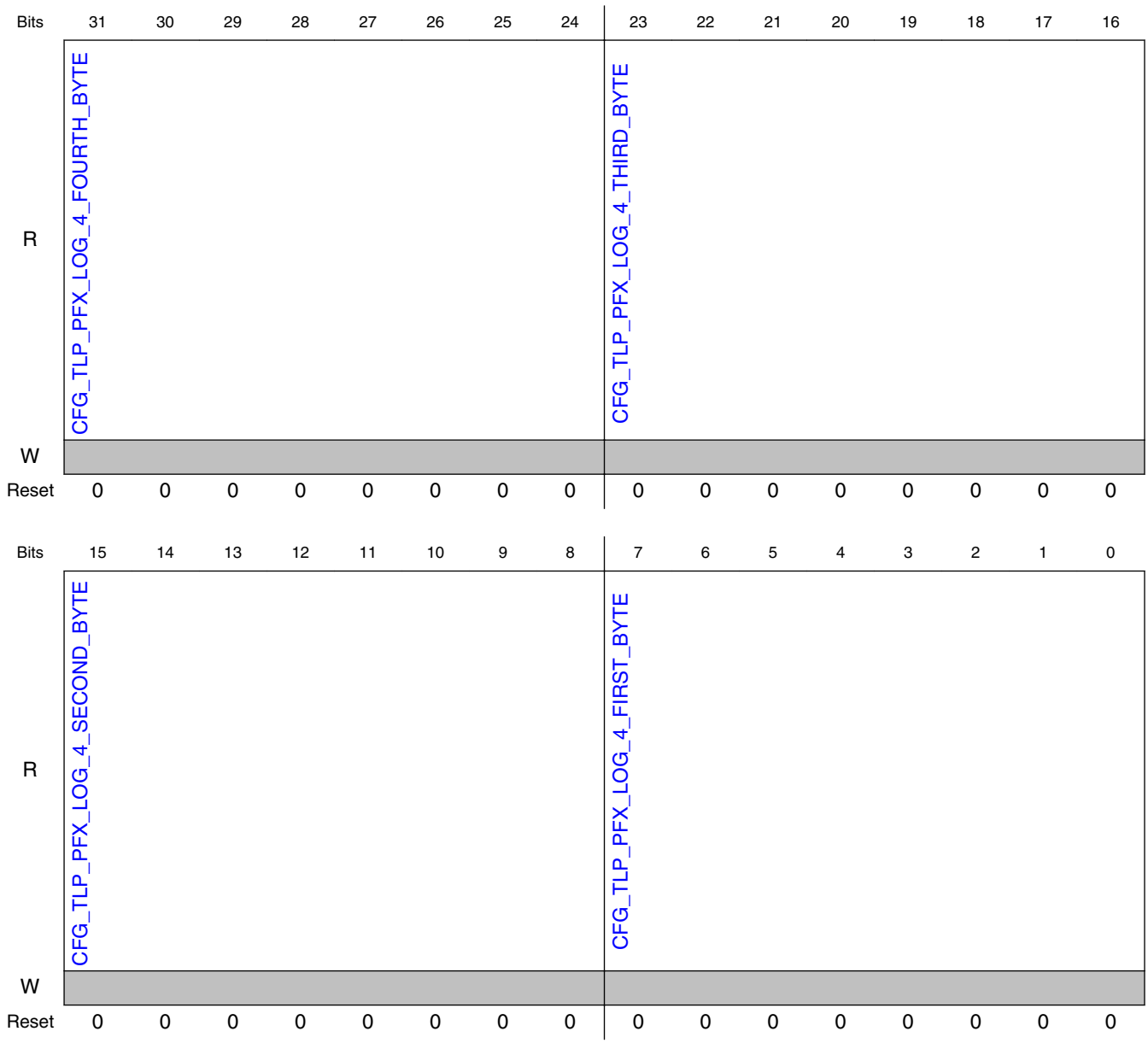
11.3.3.1.54.1 Offset

Register	Offset
TLP_PREFIX_LOG_4_OFF	144h

11.3.3.1.54.2 Function

TLP Prefix Log Register 4. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.54.3 Diagram



11.3.3.1.54.4 Fields

Field	Function
31-24 CFG_TLP_PFX_LOG_4_FOURTH_BYTE	Byte 3 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
23-16	Byte 2 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

Table continues on the next page...

Field	Function
CFG_TLP_PFX_LOG_4_THIRD_BYTE	
15-8 CFG_TLP_PFX_LOG_4_SECOND_BYTE	Byte 1 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.
7-0 CFG_TLP_PFX_LOG_4_FIRST_BYTE	Byte 0 Error TLP Prefix Log 4. For a description of this standard PCIe register field, see the PCI Express Specification. Note: This register field is sticky.

11.3.3.1.55 L1 Substates Extended Capability Header. (L1SUB_CAP_HEADER_REG)

11.3.3.1.55.1 Offset

Register	Offset
L1SUB_CAP_HEADER_REG	148h

11.3.3.1.55.2 Function

L1 Substates Extended Capability Header. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	NEXT_OFFSET												CAP_VERSION			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	EXTENDED_CAP_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0

11.3.3.1.55.4 Fields

Field	Function
31-20 NEXT_OFFSET	Next Capability Offset. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
19-16 CAP_VERSION	Capability Version. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.
15-0 EXTENDED_CAP_ID	L1SUB Extended Capability ID. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: if (DBI_RO_WR_EN == 1) then R/W(sticky) else R(sticky) Note: This register field is sticky.

11.3.3.1.56 L1 Substates Capability Register. (L1SUB_CAPABILITY_REG)

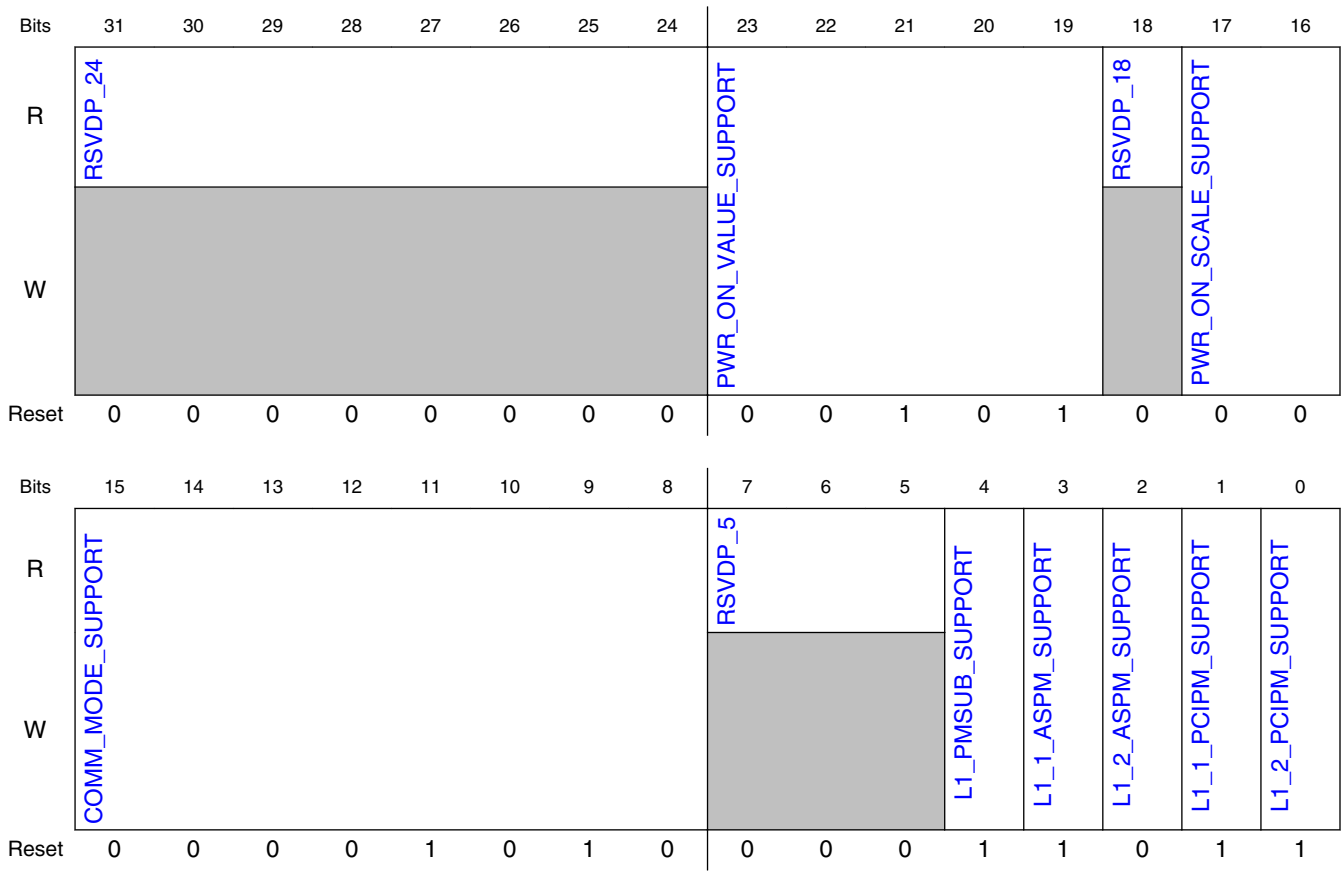
11.3.3.1.56.1 Offset

Register	Offset
L1SUB_CAPABILITY_REG	14Ch

11.3.3.1.56.2 Function

L1 Substates Capability Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.56.3 Diagram



11.3.3.1.56.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-19 PWR_ON_VALUE_SUPPORT	Port T Power On Value. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
18 RSVDP_18	Reserved for future use.
17-16 PWR_ON_SCALE_SUPPORT	Port T Power On Scale. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 COMM_MODE_SUPPORT	Port Common Mode Restore Time. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
7-5	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_5	
4 L1_PMSUB_SUPPORT	L1 PM Substates ECN Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
3 L1_1_ASPM_SUPPORT	ASPM L11 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
2 L1_2_ASPM_SUPPORT	ASPM L12 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
1 L1_1_PCIPM_SUPPORT	PCI-PM L11 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
0 L1_2_PCIPM_SUPPORT	PCI-PM L12 Supported. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.57 L1 Substates Control 1 Register. (L1SUB_CONTROL1_REG)

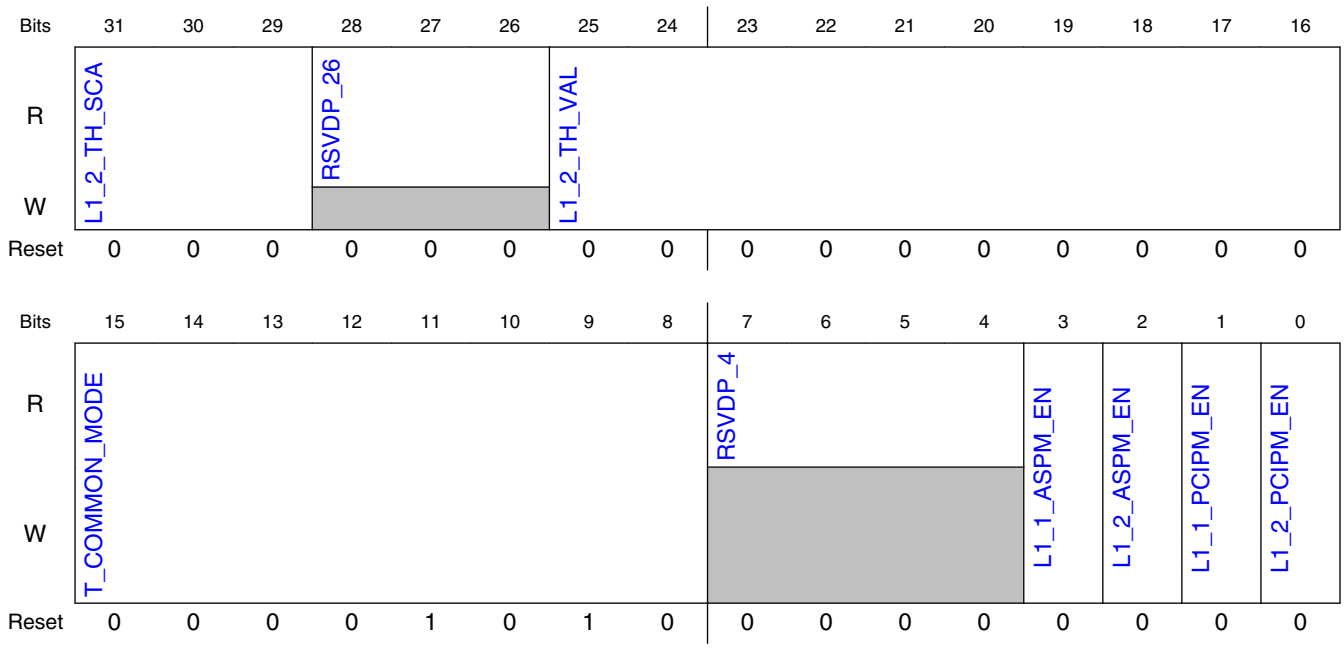
11.3.3.1.57.1 Offset

Register	Offset
L1SUB_CONTROL1_REG	150h

11.3.3.1.57.2 Function

L1 Substates Control 1 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.57.3 Diagram



11.3.3.1.57.4 Fields

Field	Function
31-29 L1_2_TH_SCA	LTR L12 Threshold Scale. For a description of this standard PCIe register field, see the PCI Express Specification.
28-26 RSVDP_26	Reserved for future use.
25-16 L1_2_TH_VAL	LTR L12 Threshold Value. For a description of this standard PCIe register field, see the PCI Express Specification.
15-8 T_COMMON_MODE	Common Mode Restore Time. For a description of this standard PCIe register field, see the PCI Express Specification. Note: The access attributes of this field are as follows: - Dbi: R/W
7-4 RSVDP_4	Reserved for future use.
3 L1_1_ASPM_EN	ASPM L11 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
2 L1_2_ASPM_EN	ASPM L12 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.
1 L1_1_PCIPM_EN	PCI-PM L11 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

Table continues on the next page...

Field	Function
0 L1_2_PCIPM_EN	PCI-PM L12 Enable. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.58 L1 Substates Control 2 Register. (L1SUB_CONTROL2_REG)

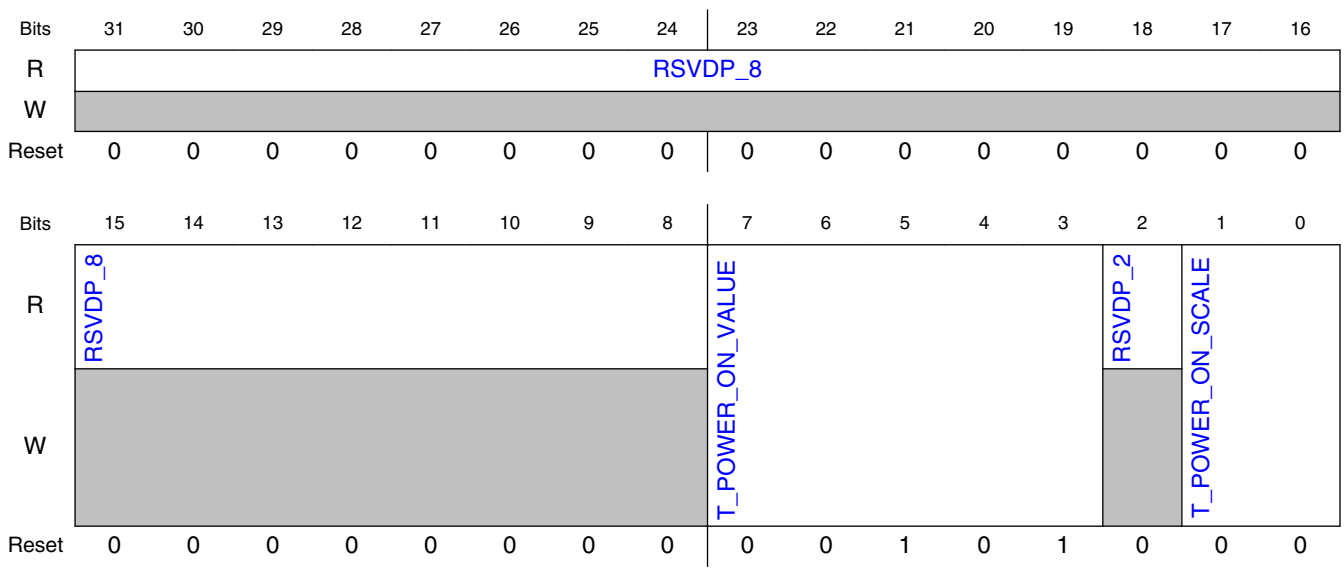
11.3.3.1.58.1 Offset

Register	Offset
L1SUB_CONTROL2_REG	154h

11.3.3.1.58.2 Function

L1 Substates Control 2 Register. For a description of this standard PCIe register, see the PCI Express Specification.

11.3.3.1.58.3 Diagram



11.3.3.1.58.4 Fields

Field	Function
31-8	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_8	
7-3 T_POWER_ON_VALUE	T Power On Value. For a description of this standard PCIe register field, see the PCI Express Specification.
2 RSVDP_2	Reserved for future use.
1-0 T_POWER_ON_SCALE	T Power On Scale. For a description of this standard PCIe register field, see the PCI Express Specification.

11.3.3.1.59 Ack Latency Timer and Replay Timer Register. (ACK_LATENCY_TIMER_OFF)

11.3.3.1.59.1 Offset

Register	Offset
ACK_LATENCY_TIMER_OFF	700h

11.3.3.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REPLAY_TIME_LIMIT															
Reset	0	0	0	1	1	0	0	0	0	1	0	0	0	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ROUND_TRIP_LATENCY_TIME_LIMIT															
Reset	0	0	0	0	1	0	0	0	0	0	0	1	0	1	1	1

11.3.3.1.59.3 Fields

Field	Function
31-16 REPLAY_TIME_LIMIT	Replay Timer Limit. The replay timer expires when it reaches this limit. The controller initiates a replay upon reception of a NAK or when the replay timer expires. For more details, see "Transmit Replay". You can modify the effective timer limit with the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-4,

Table continues on the next page...

Field	Function
	3-5, and 3-6 of the PCIe 3.0 specification. If there is a change in the payload size or link speed, the controller will override any value that you have written to this register field, and reset the field back to the specification-defined value. It will not change the value in the TIMER_MOD_REPLAY_TIMER field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.
15-0 ROUND_TRIP_LATENCY_TIME_LIMIT	Ack Latency Timer Limit. The Ack latency timer expires when it reaches this limit. For more details, see "Ack Scheduling". You can modify the effective timer limit with the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register. After reset, the controller updates the default according to the Negotiated Link Width, Max_Payload_Size, and speed. The value is determined from Tables 3-7, 3-8, and 3-9 of the PCIe 3.0 specification. The limit must reflect the round trip latency from requester to completer. If there is a change in the payload size or link width, the controller will override any value that you have written to this register field, and reset the field back to the specification-defined value. It will not change the value in the TIMER_MOD_ACK_NAK field of the TIMER_CTRL_MAX_FUNC_NUM_OFF register.

11.3.3.1.60 Vendor Specific DLLP Register. (VENDOR_SPEC_DLLP_OFF)

11.3.3.1.60.1 Offset

Register	Offset
VENDOR_SPEC_DLLP_OFF	704h

11.3.3.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	VENDOR_SPEC_DLLP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VENDOR_SPEC_DLLP															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.60.3 Fields

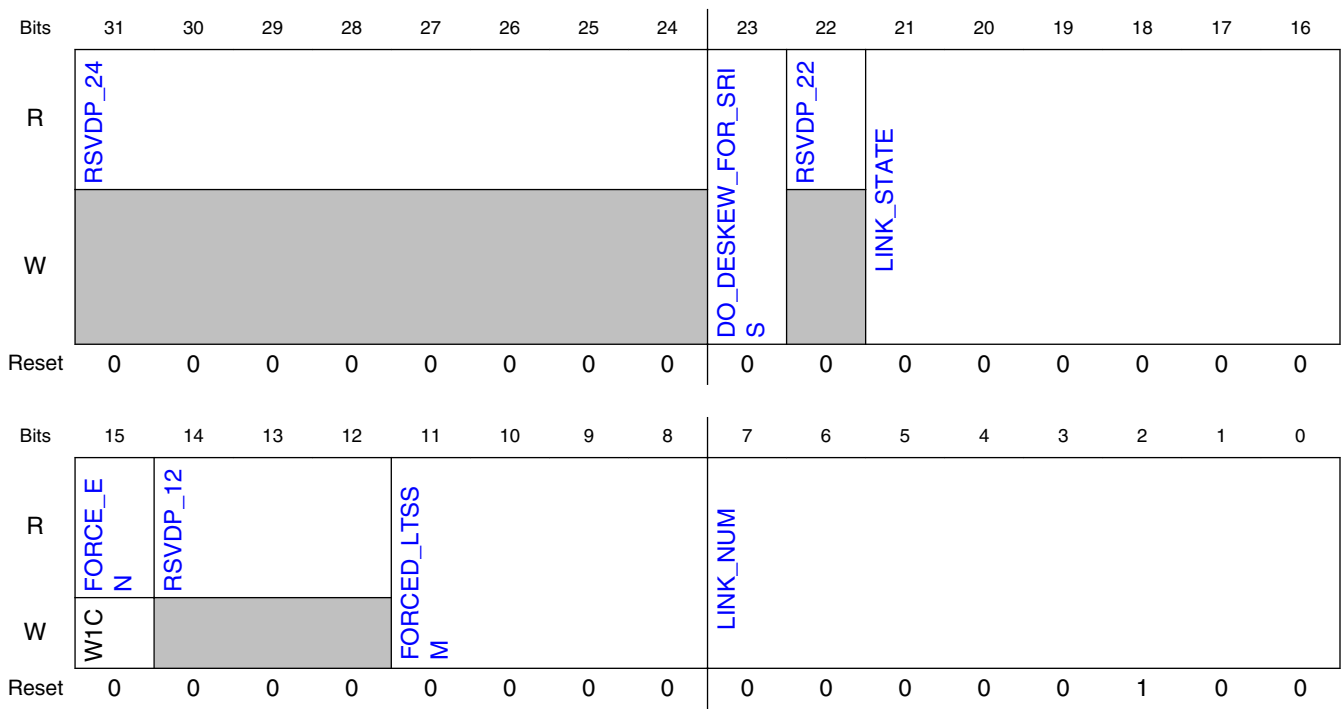
Field	Function
31-0 VENDOR_SPEC_DLLP	Vendor Specific DLLP Register. Used to send a specific PCI Express DLLP. Your application writes the 8-bit DLLP Type and 24-bits of Payload data into this register, then sets the field VENDOR_SPECIFIC_DLLP_REQ of PORT_LINK_CTRL_OFF to send the DLLP. - [7:0] = Type - [31:8] = Payload (24 bits) The dllp type is in bits [7:0] while the remainder is the vendor defined payload. Note: This register field is sticky.

11.3.3.1.61 Port Force Link Register. (PORT_FORCE_OFF)

11.3.3.1.61.1 Offset

Register	Offset
PORT_FORCE_OFF	708h

11.3.3.1.61.2 Diagram



11.3.3.1.61.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23 DO_DESKEW_FOR_SRIS	Use the transitions from TS2 to Logical Idle Symbol, SKP OS to Logical Idle Symbol, and FTS Sequence to SKP OS to do deskew for SRIS instead of using received SKP OS if DO_DESKEW_FOR_SRIS is set to 1. Note: This register field is sticky.
22 RSVDP_22	Reserved for future use.

Table continues on the next page...

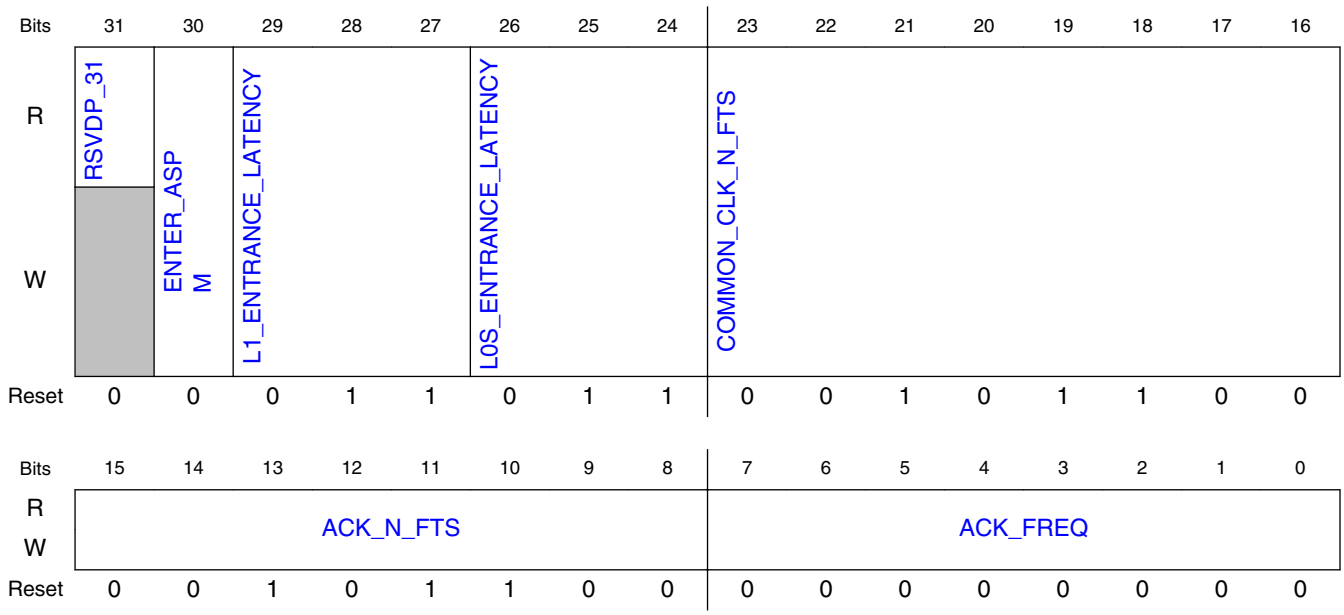
Field	Function
21-16 LINK_STATE	Forced LTSSM State. The LTSSM state that the controller is forced to when you set the FORCE_EN bit (Force Link). LTSSM state encoding is defined by the lts_state variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
15 FORCE_EN	Force Link. The controller supports a testing and debug capability to allow your software to force the LTSSM state machine into a specific state, and to force the controller to transmit a specific Link Command. Asserting this bit triggers the following actions: - Forces the LTSSM to the state specified by the Forced LTSSM State field. - Forces the controller to transmit the command specified by the Forced Link Command field. This is a self-clearing register field. Reading from this register field always returns a "0".
14-12 RSVDP_12	Reserved for future use.
11-8 FORCED_LTSSM	Forced Link Command. The link command that the controller is forced to transmit when you set FORCE_EN bit (Force Link). Link command encoding is defined by the ltssm_cmd variable in workspace/src/Layer1/smlh_ltssm.v. Note: This register field is sticky.
7-0 LINK_NUM	Link Number. Not used for endpoint. Not used for M-PCIe. Note: This register field is sticky.

11.3.3.1.62 Ack Frequency and L0-L1 ASPM Control Register. (ACK_F_ASPM_CTRL_OFF)

11.3.3.1.62.1 Offset

Register	Offset
ACK_F_ASPM_CTRL_OFF	70Ch

11.3.3.1.62.2 Diagram



11.3.3.1.62.3 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30 ENTER_ASPM	ASPM L1 Entry Control. - 1: Core enters ASPM L1 after a period in which it has been idle. - 0: Core enters ASPM L1 only after idle period during which both receive and transmit are in L0s. Note: This register field is sticky.
29-27 L1_ENTRANCE_LATENCY	L1 Entrance Latency. Value range is: - 000: 1 us - 001: 2 us - 010: 4 us - 011: 8 us - 100: 16 us - 101: 32 us - 110 or 111: 64 us Note: Programming this timer with a value greater that 32us has no effect unless extended sync is used, or all of the credits are infinite. Note: This register field is sticky.
26-24 L0S_ENTRANCE_LATENCY	L0s Entrance Latency. Values correspond to: - 000: 1 us - 001: 2 us - 010: 3 us - 011: 4 us - 100: 5 us - 101: 6 us - 110 or 111: 7 us This field is applicable to STALL while in L0 for M-PCIe. Note: This register field is sticky.
23-16 COMMON_CLK_N_FTS	Common Clock N_FTS. This is the N_FTS when common clock is used. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. The controller does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R
15-8 ACK_N_FTS	N_FTS. The number of Fast Training Sequence ordered sets to be transmitted when transitioning from L0s to L0. The maximum number of FTS ordered-sets that a component can request is 255. The controller does not support a value of zero; a value of zero can cause the LTSSM to go into the recovery state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
7-0 ACK_FREQ	Ack Frequency. The controller accumulates the number of pending ACKs specified here (up to 255) before sending an ACK DLLP. - 0: Indicates that this Ack frequency control feature is turned off. The controller schedules a low-priority ACK DLLP for every TLP that it receives. - 1-255: Indicates that the

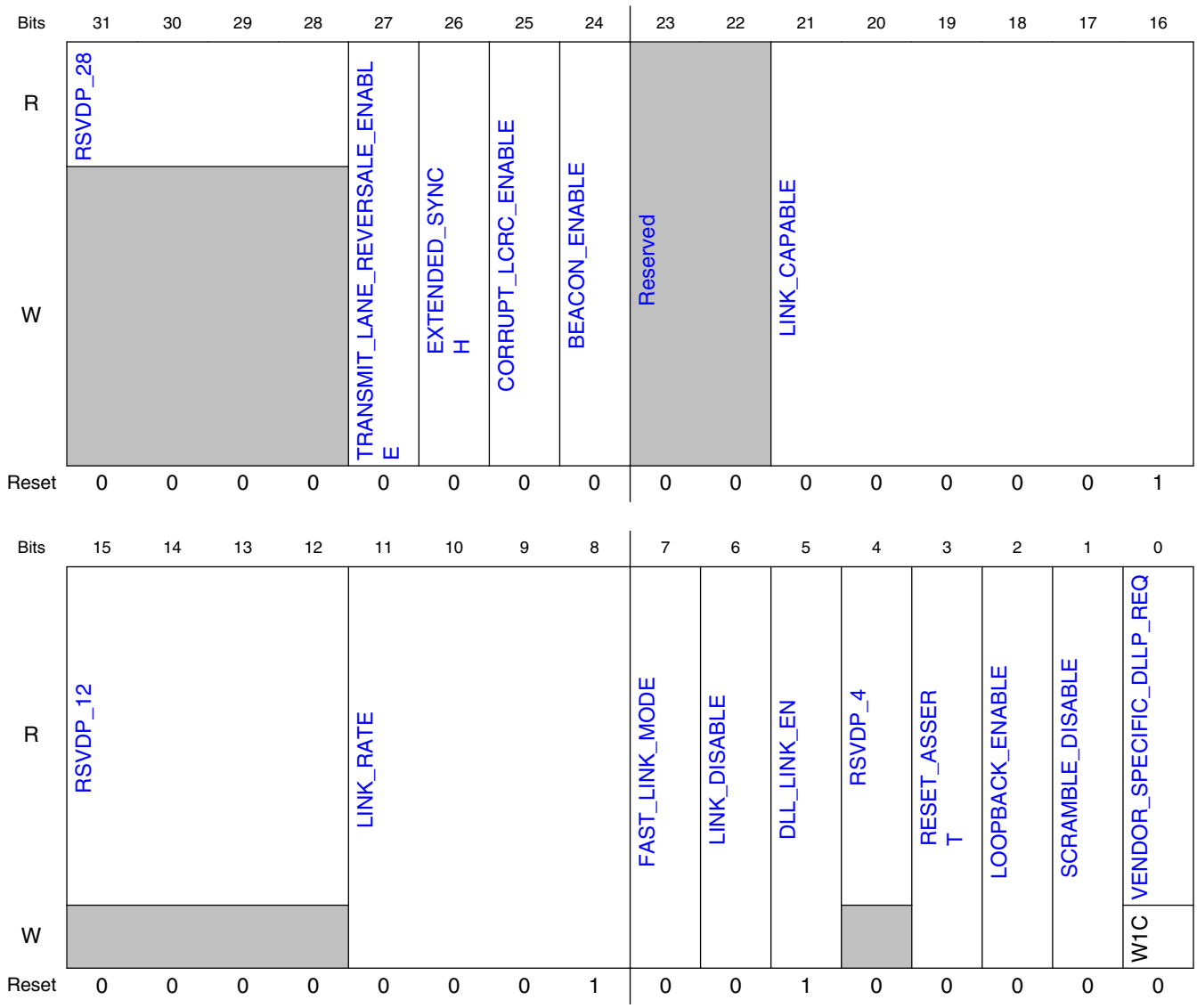
Field	Function
	controller will schedule a high-priority ACK after receiving this number of TLPs. It might schedule the ACK before receiving this number of TLPs, but never later. For a typical system, you do not have to modify the default setting. For more details, see "ACK/NAK Scheduling". Note: This register field is sticky.

11.3.3.1.63 Port Link Control Register. (PORT_LINK_CTRL_OFF)

11.3.3.1.63.1 Offset

Register	Offset
PORT_LINK_CTRL_OFF	710h

11.3.3.1.63.2 Diagram



11.3.3.1.63.3 Fields

Field	Function
31-28 RSVDP_28	Reserved for future use.
27 TRANSMIT_LANE_REVERSALE_ENABLE	TRANSMIT_LANE_REVERSALE_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
26	EXTENDED_SYNCH is an internally reserved field. Do not use. Note: This register field is sticky.

Table continues on the next page...

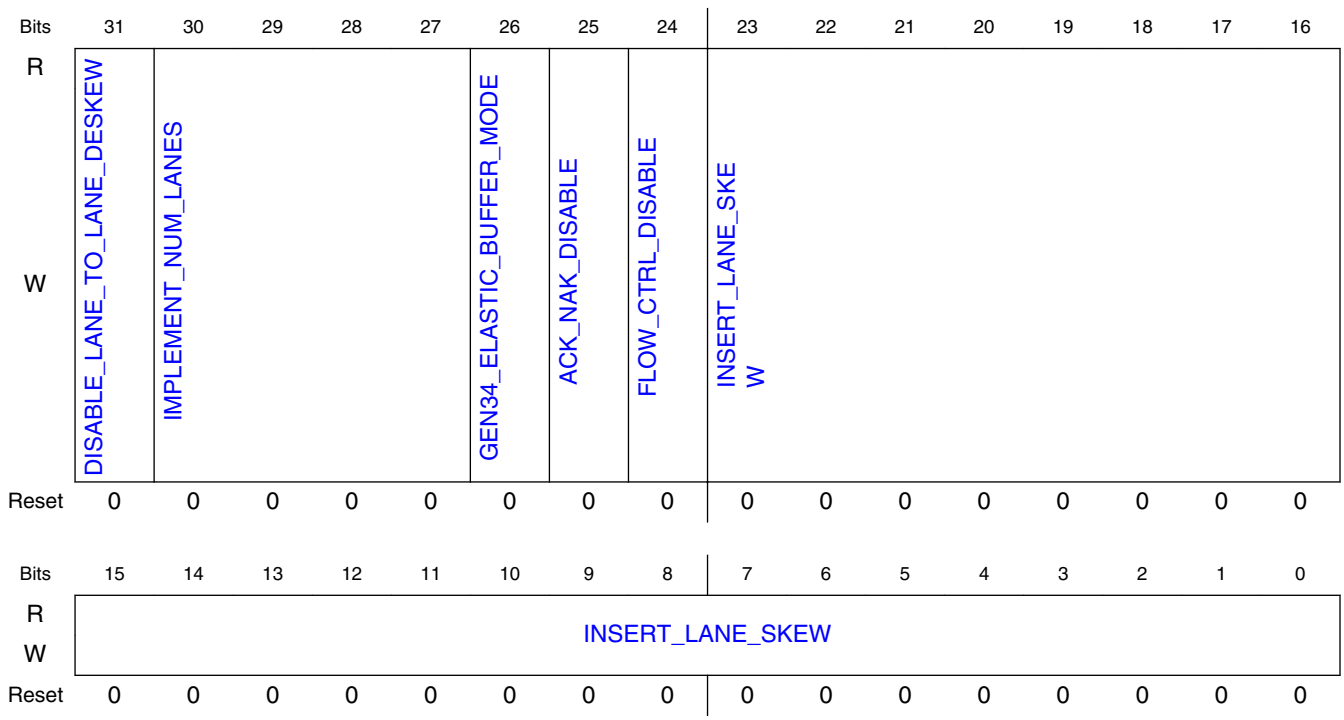
Field	Function
EXTENDED_SYNCH	
25 CORRUPT_LCRC_ENABLE	CORRUPT_LCRC_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
24 BEACON_ENABLE	BEACON_ENABLE is an internally reserved field. Do not use. Note: This register field is sticky.
23-22 —	Reserved.
21-16 LINK_CAPABLE	Link Mode Enable. Sets the number of lanes in the link that you want to connect to the link partner. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Predetermined Number of Lanes" field of the "Link Width and Speed Change Control Register". For more information, see "How to Tie Off Unused Lanes". For information on upsizing and downsizing the link width, see "Link Establishment". - 000001: x1 - 000011: x2 - 000111: x4 - 001111: x8 - 011111: x16 - 111111: x32 (not supported) This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
15-12 RSVDP_12	Reserved for future use.
11-8 LINK_RATE	LINK_RATE is an internally reserved field. Do not use. Note: This register field is sticky.
7 FAST_LINK_MODE	Fast Link Mode. Sets all internal LTSSM millisecond timers to Fast Mode for speeding up simulation. Forces the LTSSM training (link initialization) to use shorter time-outs and to link up faster. The default scaling factor can be changed through the FAST_LINK_SCALING_FACTOR field in the TIMER_CTRL_MAX_FUNC_NUM_OFF register. If this bit is set to '1', tRRAPInitiatorResponse is set to 1.88 ms(60 ms/32). Note: This register field is sticky.
6 LINK_DISABLE	LINK_DISABLE is an internally reserved field. Do not use. Note: This register field is sticky.
5 DLL_LINK_EN	DLL Link Enable. Enables link initialization. When DLL Link Enable =0, the controller does not transmit InitFC DLLPs and does not establish a link. Note: This register field is sticky.
4 RSVDP_4	Reserved for future use.
3 RESET_ASSERT	Reset Assert. Triggers a recovery and forces the LTSSM to the hot reset state (downstream port only). Note: This register field is sticky.
2 LOOPBACK_ENABLE	Loopback Enable. Turns on loopback. For more details, see "Loopback". For M-PCIe, to force the master to enter Digital Loopback mode, you must set this field to "1" during Configuration.start state(initial discovery/configuration). M-PCIe doesn't support loopback mode from L0 state - only from Configuration.start. Note: This register field is sticky.
1 SCRAMBLE_DISABLE	Scramble Disable. Turns off data scrambling. Note: This register field is sticky.
0 VENDOR_SPECIFIC_DLLP_REQ	Vendor Specific DLLP Request. When software writes a '1' to this bit, the controller transmits the DLLP contained in the VENDOR_SPEC_DLLP field of VENDOR_SPEC_DLLP_OFF. Reading from this self-clearing register field always returns a '0'.

11.3.3.1.64 Lane Skew Register. (LANE_SKEW_OFF)

11.3.3.1.64.1 Offset

Register	Offset
LANE_SKEW_OFF	714h

11.3.3.1.64.2 Diagram



11.3.3.1.64.3 Fields

Field	Function
31 DISABLE_LANE_TO_LANE_DESKEW	Disable Lane-to-Lane Deskew. Causes the controller to disable the internal Lane-to-Lane deskew logic. Note: This register field is sticky.
30-27 IMPLEMENT_NUM_LANES	Implementation-specific Number of Lanes. Set the implementation-specific number of lanes. Allowed values are: - 4'b0000: 1 lane - 4'b0001: 2 lanes - 4'b0011: 4 lanes - 4'b0111: 8 lanes - 4'b1111: 16 lanes The number of lanes to be used when in Loopback Master. The number of lanes programmed must be equal to or less than the valid number of lanes set in LINK_CAPABLE field. You must configure this field before initiating Loopback by writing in the LOOPBACK_ENABLE field. The controller will transition from Loopback.Entry to Loopback.Active after receiving two consecutive TS1 Ordered Sets with the Loopback

Table continues on the next page...

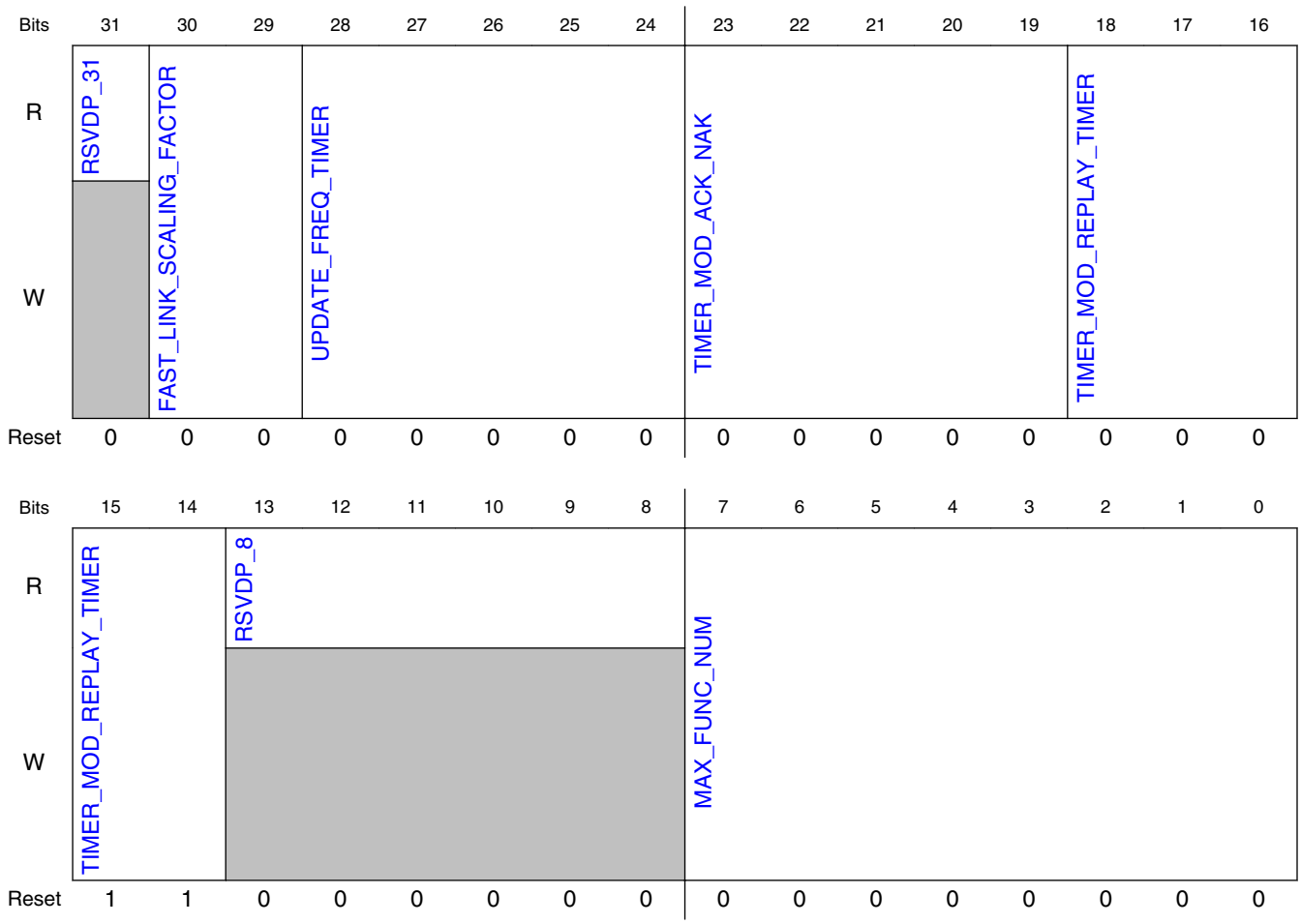
Field	Function
	bit asserted on the implementation specific number of lanes configured in this field. Note: This register field is sticky.
26 GEN34_ELASTIC_BUFFER_MODE	Selects Elasticity Buffer operating mode in Gen3 or Gen4 rate: 0: Nominal Half Full Buffer mode 1: Nominal Empty Buffer Mode This register bit only affects Gen3 or Gen4 operating rate. For Gen1 or Gen2 operating rate the Elasticity Buffer operating mode is always the Nominal Half Full Buffer mode. Note: This register field is sticky.
25 ACK_NAK_DISABLE	Ack/Nak Disable. Prevents the controller from sending ACK and NAK DLLPs. Note: This register field is sticky.
24 FLOW_CTRL_DISABLE	Flow Control Disable. Prevents the controller from sending FC DLLPs. Note: This register field is sticky.
23-0 INSERT_LANE_SKEW	INSERT_LANE_SKEW is an internally reserved field. Do not use. Note: This register field is sticky.

11.3.3.1.65 Timer Control and Max Function Number Register. (TIMER_CTRL_MAX_FUNC_NUM_OFF)

11.3.3.1.65.1 Offset

Register	Offset
TIMER_CTRL_MAX_FUNC_NUM_OFF	718h

11.3.3.1.65.2 Diagram



11.3.3.1.65.3 Fields

Field	Function
31 RSVDP_31	Reserved for future use.
30-29 FAST_LINK_SCALING_FACTOR	Fast Link Timer Scaling Factor. Sets the scaling factor of LTSSM timer when FAST_LINK_MODE field in PORT_LINK_CTRL_OFF is set to '1'. - 0: Scaling Factor is 1024 (1ms is 1us) - 1: Scaling Factor is 256 (1ms is 4us) - 2: Scaling Factor is 64 (1ms is 16us) - 3: Scaling Factor is 16 (1ms is 64us). Not used for M-PCIe. Note: This register field is sticky.
28-24 UPDATE_FREQ_TIMER	UPDATE_FREQ_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
23-19 TIMER_MOD_ACK_NAK	Ack Latency Timer Modifier. Increases the timer value for the Ack latency timer in increments of 64 clock cycles. A value of "0" represents no modification to the timer value. For more details, see the ROUND_TRIP_LATENCY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. Note: This register field is sticky.

Table continues on the next page...

Field	Function
18-14 TIMER_MOD_REPLAY_TIMER	Replay Timer Limit Modifier. Increases the time-out value for the replay timer in increments of 64 clock cycles at Gen1 or Gen2 speed, and in increments of 256 clock cycles at Gen3 speed. A value of "0" represents no modification to the timer limit. For more details, see the REPLAY_TIME_LIMIT field of the ACK_LATENCY_TIMER_OFF register. At Gen3 speed, the controller automatically changes the value of this field to DEFAULT_GEN3_REPLAY_ADJ. For M-PCIe, this field increases the time-out value for the replay timer in increments of 64 clock cycles at HS-Gear1, HS-Gear2, or HS-Gear3 speed. Note: This register field is sticky.
13-8 RSVDP_8	Reserved for future use.
7-0 MAX_FUNC_NUM	Maximum function number that can be used in a request. Configuration requests targeted at function numbers above this value are returned with UR (unsupported request). Note: This register field is sticky.

11.3.3.1.66 Symbol Timer Register and Filter Mask 1 Register. (SYMBOL_TIMER_FILTER_1_OFF)

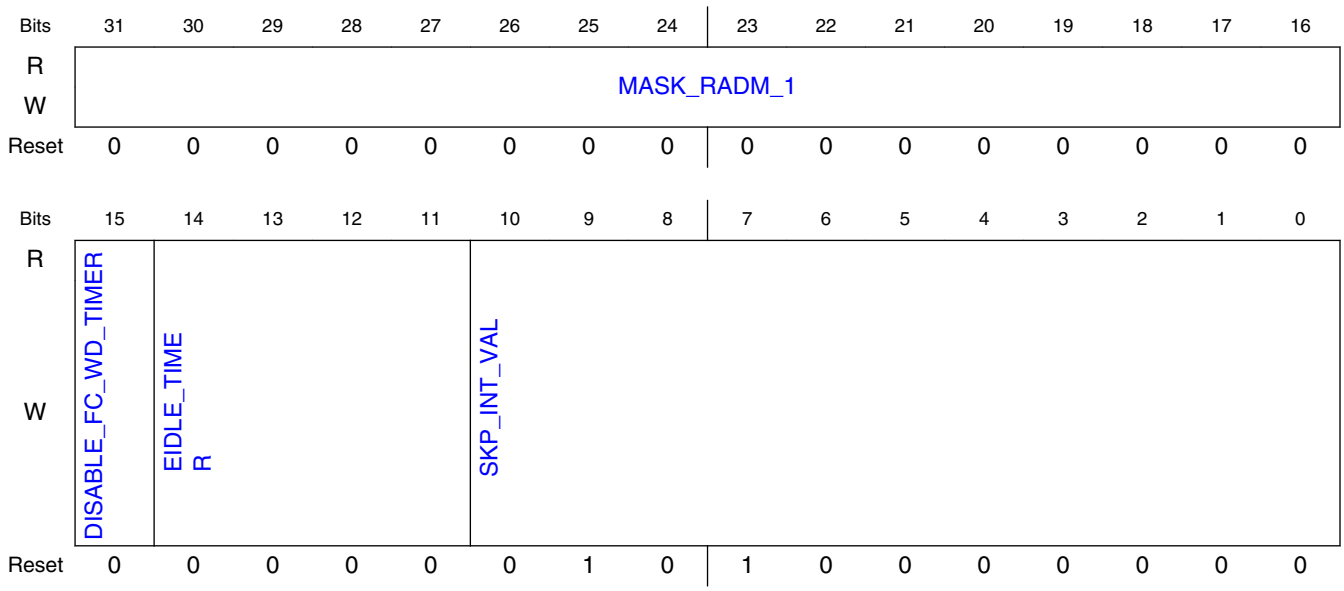
11.3.3.1.66.1 Offset

Register	Offset
SYMBOL_TIMER_FILTER_1_OFF	71Ch

11.3.3.1.66.2 Function

Symbol Timer Register and Filter Mask 1 Register. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

11.3.3.1.66.3 Diagram



11.3.3.1.66.4 Fields

Field	Function
31-16 MASK_RADM_1	Filter Mask 1. The Filter Mask 1 Register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. Note: This register field is sticky.
15 DISABLE_FC_WD_TIMER	Disable FC Watchdog Timer. Note: This register field is sticky.
14-11 EIDLE_TIMER	EIDLE_TIMER is an internally reserved field. Do not use. Note: This register field is sticky.
10-0 SKP_INT_VAL	SKP Interval Value. The number of symbol times to wait between transmitting SKP ordered sets. Note that the controller actually waits the number of symbol times in this register plus 1 between transmitting SKP ordered sets. Your application must program this register accordingly. For example, if 1536 were programmed into this register (in a 250 MHz controller), then the controller actually transmits SKP ordered sets once every 1537 symbol times. The value programmed to this register is actually clock ticks and not symbol times. In a 125 MHz controller, programming the value programmed to this register should be scaled down by a factor of 2 (because one clock tick = two symbol times in this case). Note: This value is not used at Gen3 speed; the skip interval is hardcoded to 370 blocks. Note: This register field is sticky.

11.3.3.1.67 Filter Mask 2 Register. (FILTER_MASK_2_OFF)

11.3.3.1.67.1 Offset

Register	Offset
FILTER_MASK_2_OFF	720h

11.3.3.1.67.2 Function

Filter Mask 2 Register. This register modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule.

11.3.3.1.67.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASK_RADM_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASK_RADM_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.67.4 Fields

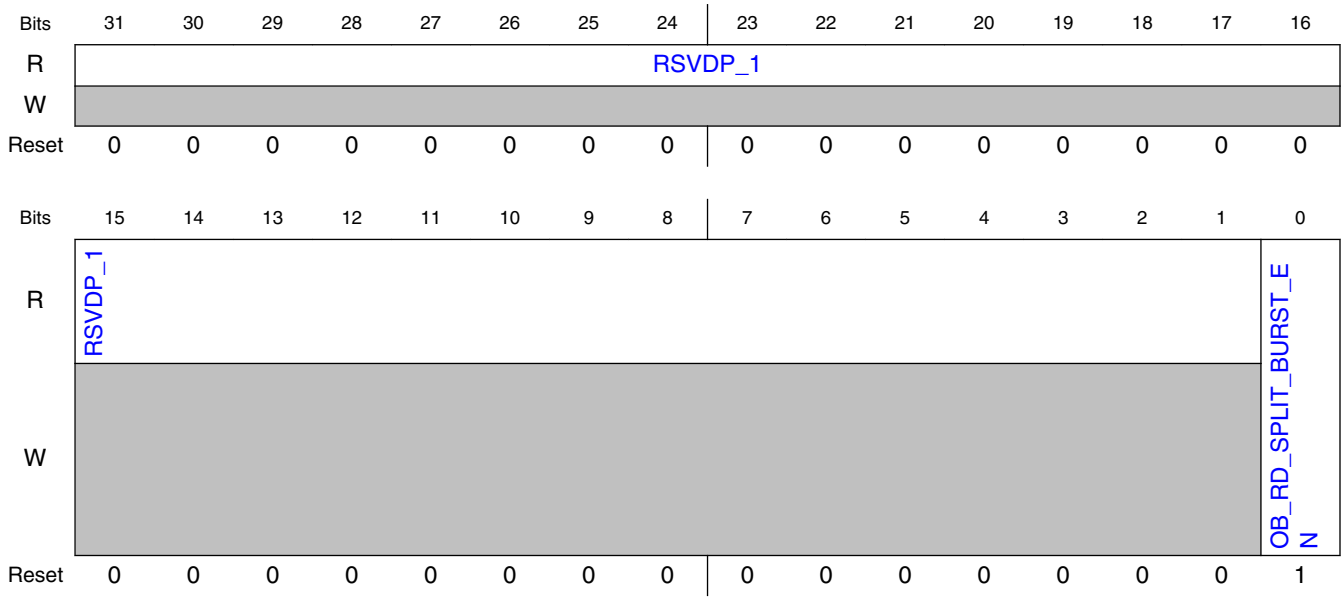
Field	Function
31-0 MASK_RADM_2	Filter Mask 2. This field modifies the RADM filtering and error handling rules. For more details, see the "Receive Filtering" section. In each case, '0' applies the associated filtering rule and '1' masks the associated filtering rule. Note: This register field is sticky.

11.3.3.1.68 AMBA Multiple Outbound Decomposed NP SubRequests Control Register. (AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF)

11.3.3.1.68.1 Offset

Register	Offset
AMBA_MUL_OB_DECOMP_NP_SUB_REQ_CTRL_OFF	724h

11.3.3.1.68.2 Diagram



11.3.3.1.68.3 Fields

Field	Function
31-1 RSVDP_1	Reserved for future use.
0 OB_RD_SPLIT_BURST_EN	Enable AMBA Multiple Outbound Decomposed NP SubRequests. This bit when set to "0" disables the possibility of having multiple outstanding non-posted requests that were derived from decomposition of an outbound AMBA request. For more details, see "AXI Bridge Ordering" in the AXI chapter of the Databook. You should not clear this register unless your application master is requesting an amount of read data greater than Max_Read_Request_Size, and the remote device (or switch) is reordering completions that have different tags. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

11.3.3.1.69 Debug Register 0 (PL_DEBUG0_OFF)

11.3.3.1.69.1 Offset

Register	Offset
PL_DEBUG0_OFF	728h

11.3.3.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEB_REG_0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEB_REG_0															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.3.3.1.69.3 Fields

Field	Function
31-0 DEB_REG_0	The value on cxpl_debug_info[31:0].

11.3.3.1.70 Debug Register 1 (PL_DEBUG1_OFF)

11.3.3.1.70.1 Offset

Register	Offset
PL_DEBUG1_OFF	72Ch

11.3.3.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEB_REG_1															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEB_REG_1															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

11.3.3.1.70.3 Fields

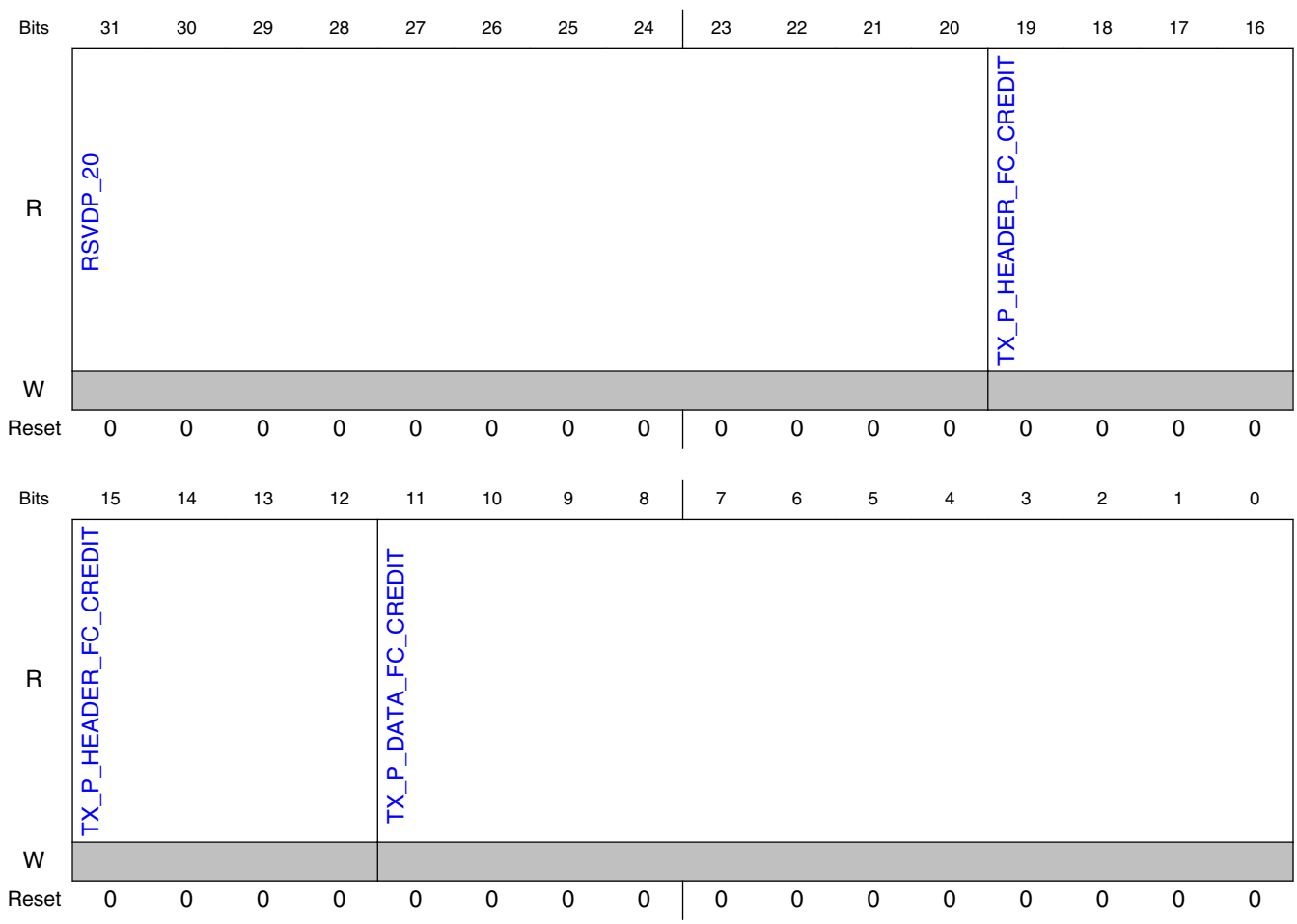
Field	Function
31-0 DEB_REG_1	The value on cxpl_debug_info[63:32].

11.3.3.1.71 Transmit Posted FC Credit Status (TX_P_FC_CREDIT_STATUS_OFF)

11.3.3.1.71.1 Offset

Register	Offset
TX_P_FC_CREDIT_STATUS_OFF	730h

11.3.3.1.71.2 Diagram



11.3.3.1.71.3 Fields

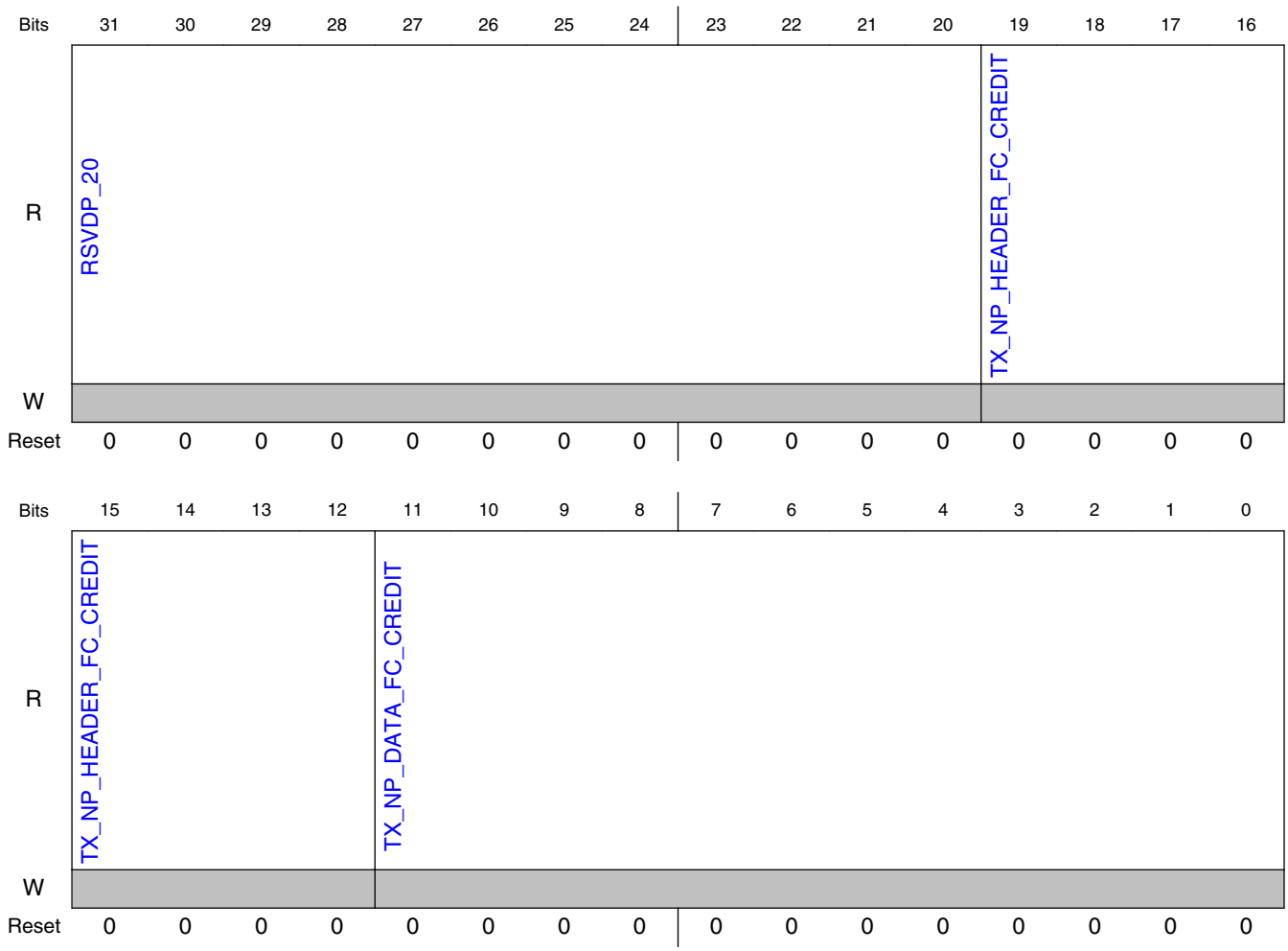
Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_P_HEADER_FC_CREDIT	Transmit Posted Header FC Credits. The posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_P_DATA_FC_CREDIT	Transmit Posted Data FC Credits. The posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_ph_cdts, xtlh_xadm_pd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.3.3.1.72 Transmit Non-Posted FC Credit Status (TX_NP_FC_CREDIT_STATUS_OFF)

11.3.3.1.72.1 Offset

Register	Offset
TX_NP_FC_CREDIT_STATUS_OFF	734h

11.3.3.1.72.2 Diagram



11.3.3.1.72.3 Fields

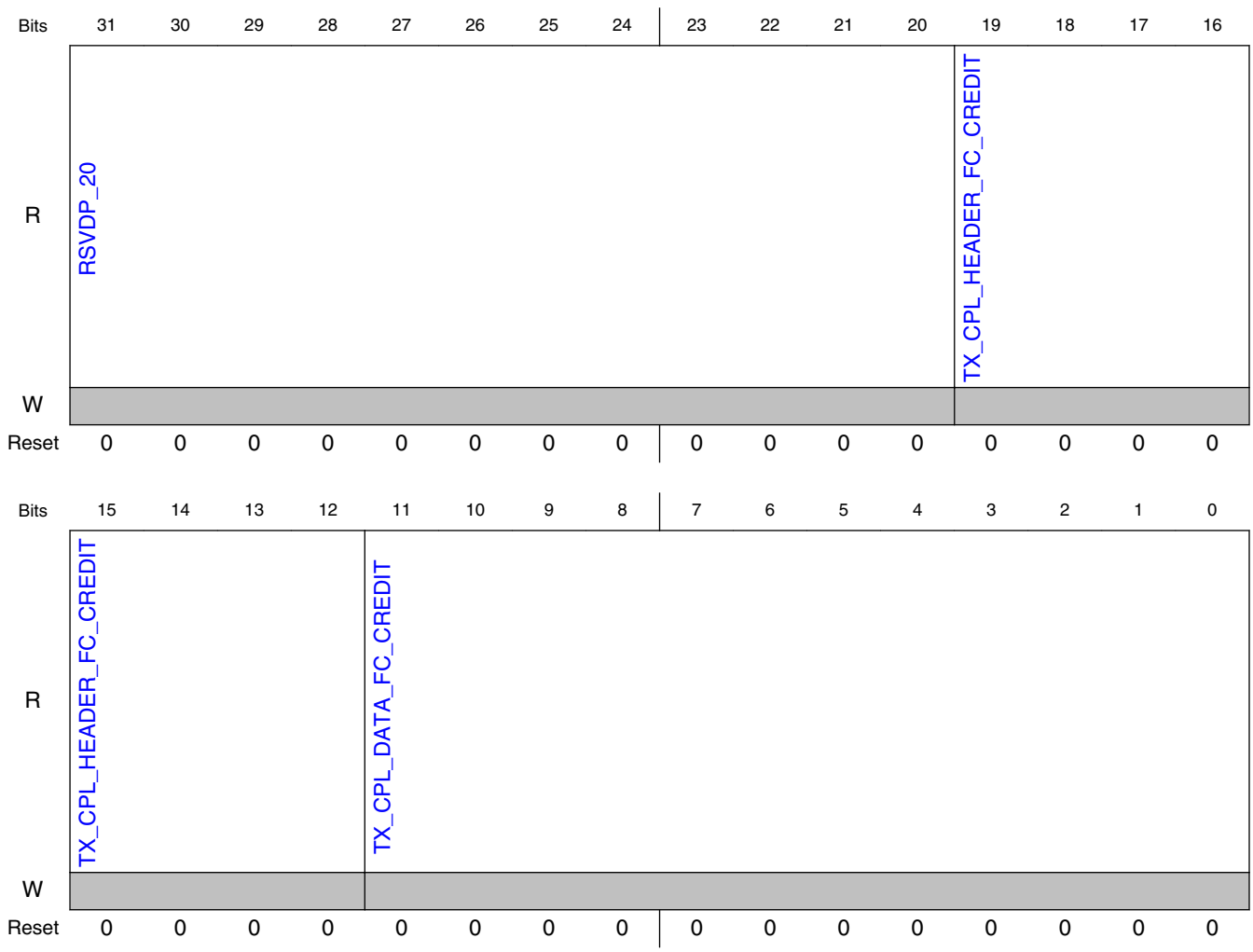
Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_NP_HEADER_FC_CREDIT	Transmit Non-Posted Header FC Credits. The non-posted Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_NP_DATA_FC_CREDIT	Transmit Non-Posted Data FC Credits. The non-posted Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_nph_cdts, xtlh_xadm_npd_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.3.3.1.73 Transmit Completion FC Credit Status (TX_CPL_FC_CREDIT_STATUS_OFF)

11.3.3.1.73.1 Offset

Register	Offset
TX_CPL_FC_CREDIT_STATUS_OFF	738h

11.3.3.1.73.2 Diagram



11.3.3.1.73.3 Fields

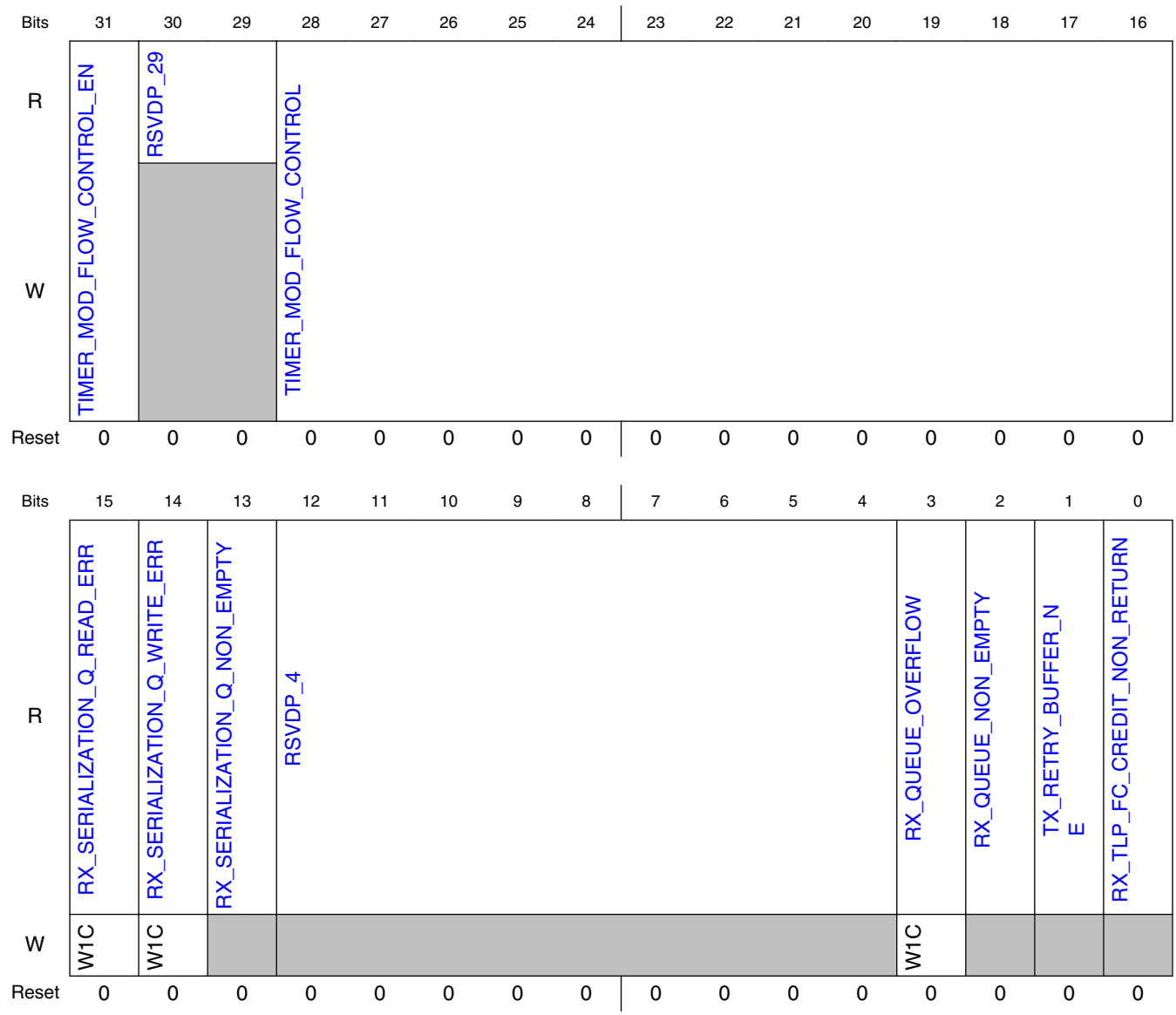
Field	Function
31-20 RSVDP_20	Reserved for future use.
19-12 TX_CPL_HEADER_FC_CREDIT	Transmit Completion Header FC Credits. The Completion Header credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].
11-0 TX_CPL_DATA_FC_CREDIT	Transmit Completion Data FC Credits. The Completion Data credits advertised by the receiver at the other end of the link, updated with each UpdateFC DLLP. Default value depends on the number of advertised credits for header and data [12'b0, xtlh_xadm_cplh_cdts, xtlh_xadm_cpld_cdts]; When the number of advertised completion credits (both header and data) are infinite, then the default would be [12'b0, 8'hFF, 12'hFFF].

11.3.3.1.74 Queue Status (QUEUE_STATUS_OFF)

11.3.3.1.74.1 Offset

Register	Offset
QUEUE_STATUS_OFF	73Ch

11.3.3.1.74.2 Diagram



11.3.3.1.74.3 Fields

Field	Function
31 TIMER_MOD_FLOW_CONTROL_EN	FC Latency Timer Override Enable. When this bit is set, the value from the "FC Latency Timer Override Value" field in this register will override the FC latency timer value that the controller calculates according to the PCIe specification. Note: This register field is sticky.
30-29 RSVDP_29	Reserved for future use.

Table continues on the next page...

Field	Function
28-16 TIMER_MOD_FLOW_CONTROL	FC Latency Timer Override Value. When you set the "FC Latency Timer Override Enable" in this register, the value in this field will override the FC latency timer value that the controller calculates according to the PCIe specification. For more details, see "Flow Control". Note: This register field is sticky.
15 RX_SERIALIZATION_Q_READ_ERR	Receive Serialization Read Error. Indicates the serialization queue has attempted to read an incorrectly formatted TLP.
14 RX_SERIALIZATION_Q_WRITE_ERR	Receive Serialization Queue Write Error. Indicates insufficient buffer space available to write to the serialization queue.
13 RX_SERIALIZATION_Q_NON_EMPTY	Receive Serialization Queue Not Empty. Indicates there is data in the serialization queue.
12-4 RSVDP_4	Reserved for future use.
3 RX_QUEUE_OVERFLOW	Receive Credit Queue Overflow. Indicates insufficient buffer space available to write to the P/NP/CPL credit queue.
2 RX_QUEUE_NON_EMPTY	Receive Credit Queue Not Empty. Indicates there is data in one or more of the receive buffers.
1 TX_RETRY_BUFFER_NE	Transmit Retry Buffer Not Empty. Indicates that there is data in the transmit retry buffer.
0 RX_TLP_FC_CREDIT_NON_RETURN	Received TLP FC Credits Not Returned. Indicates that the controller has received a TLP but has not yet sent an UpdateFC DLLP indicating that the credits for that TLP have been restored by the receiver at the other end of the link.

11.3.3.1.75 VC Transmit Arbitration Register 1 (VC_TX_ARBI_1_OFF)

11.3.3.1.75.1 Offset

Register	Offset
VC_TX_ARBI_1_OFF	740h

11.3.3.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRR_WEIGHT_VC_3								WRR_WEIGHT_VC_2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRR_WEIGHT_VC_1								WRR_WEIGHT_VC_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

11.3.3.1.75.3 Fields

Field	Function
31-24 WRR_WEIGHT_VC_3	WRR Weight for VC3. Note: The access attributes of this field are as follows: - Dbi: R
23-16 WRR_WEIGHT_VC_2	WRR Weight for VC2. Note: The access attributes of this field are as follows: - Dbi: R
15-8 WRR_WEIGHT_VC_1	WRR Weight for VC1. Note: The access attributes of this field are as follows: - Dbi: R
7-0 WRR_WEIGHT_VC_0	WRR Weight for VC0. Note: The access attributes of this field are as follows: - Dbi: R

11.3.3.1.76 VC Transmit Arbitration Register 2 (VC_TX_ARBI_2_OFF)

11.3.3.1.76.1 Offset

Register	Offset
VC_TX_ARBI_2_OFF	744h

11.3.3.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WRR_WEIGHT_VC_7								WRR_WEIGHT_VC_6							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	WRR_WEIGHT_VC_5								WRR_WEIGHT_VC_4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.76.3 Fields

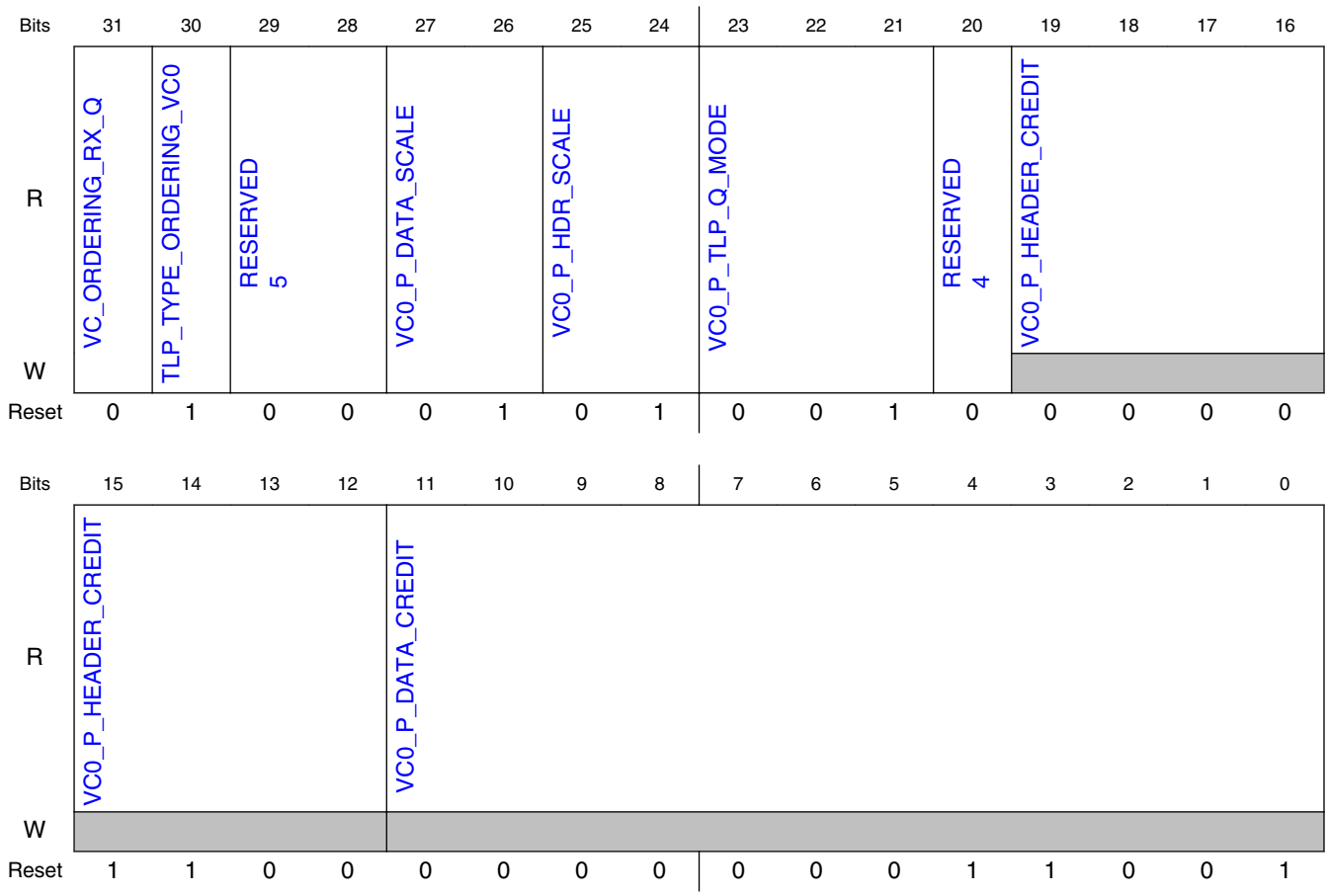
Field	Function
31-24 WRR_WEIGHT_VC_7	WRR Weight for VC7. Note: The access attributes of this field are as follows: - Dbi: R
23-16 WRR_WEIGHT_VC_6	WRR Weight for VC6. Note: The access attributes of this field are as follows: - Dbi: R
15-8 WRR_WEIGHT_VC_5	WRR Weight for VC5. Note: The access attributes of this field are as follows: - Dbi: R
7-0 WRR_WEIGHT_VC_4	WRR Weight for VC4. Note: The access attributes of this field are as follows: - Dbi: R

11.3.3.1.77 Segmented-Buffer VC0 Posted Receive Queue Control. (VC0_P_RX_Q_CTRL_OFF)

11.3.3.1.77.1 Offset

Register	Offset
VC0_P_RX_Q_CTRL_OFF	748h

11.3.3.1.77.2 Diagram



11.3.3.1.77.3 Fields

Field	Function
31 VC_ORDERING_RX_Q	VC Ordering for Receive Queues. Determines the VC ordering rule for the receive queues, used only in the segmented-buffer configuration: - 1: Strict ordering, higher numbered VCs have higher priority - 0: Round robin Note: This register field is sticky.
30 TLP_TYPE_ORDERING_VC0	TLP Type Ordering for VC0. Determines the TLP type ordering rule for VC0 receive queues, used only in the segmented-buffer configuration: - 1: PCIe ordering rules (recommended) - 0: Strict ordering: posted, completion, then non-posted Note: This register field is sticky.
29-28 RESERVED5	Reserved. Note: This register field is sticky.
27-26 VC0_P_DATA_SCALE	VC0 Scale Posted Data Credites. Note: This register field is sticky.
25-24 VC0_P_HDR_SCALE	VC0 Scale Posted Header Credites. Note: This register field is sticky.

Table continues on the next page...

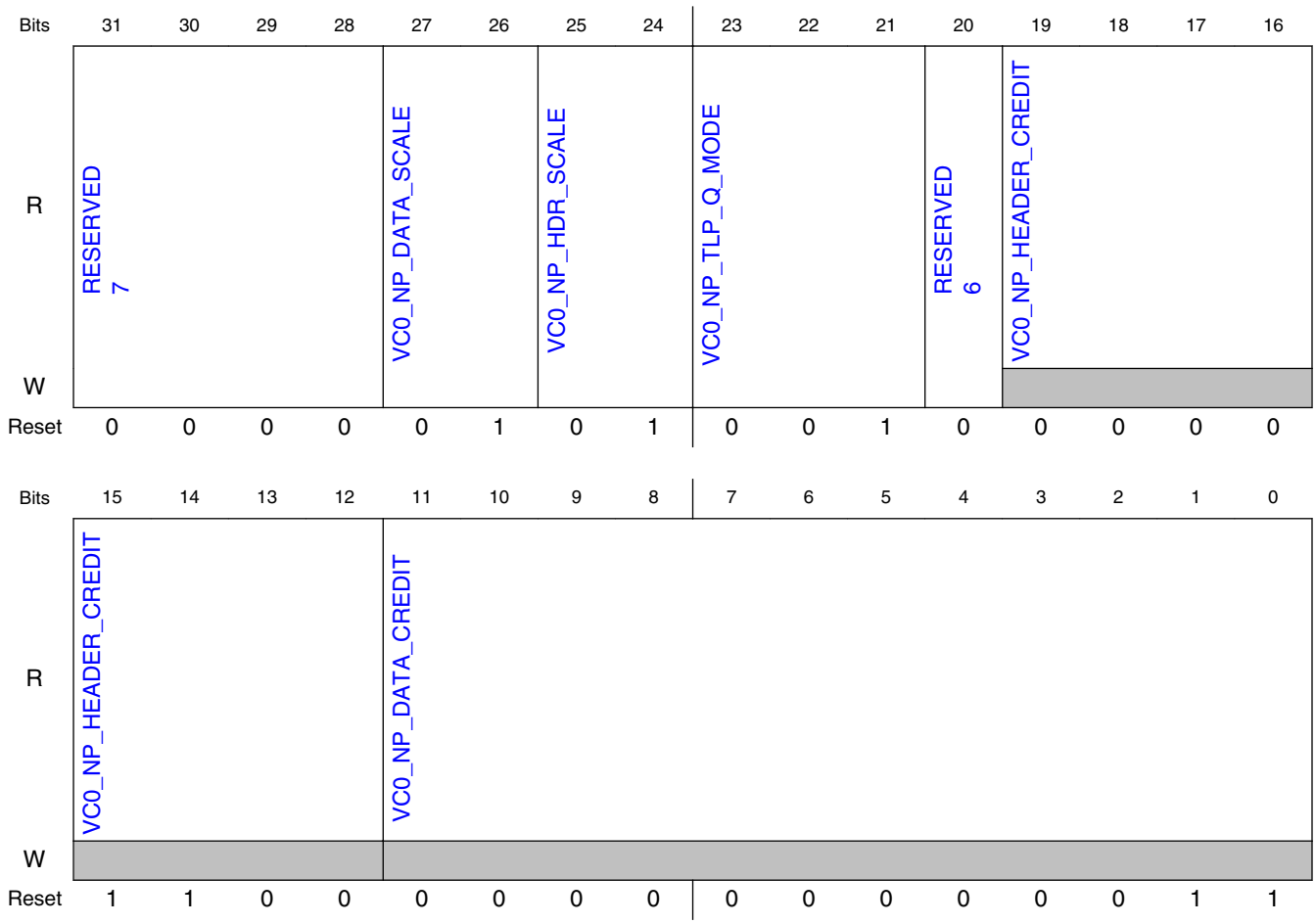
Field	Function
23-21 VC0_P_TLP_Q_MODE	Reserved. Note: This register field is sticky.
20 RESERVED4	Reserved. Note: This register field is sticky.
19-12 VC0_P_HEADER_CREDIT	VC0 Posted Header Credits. The number of initial posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_P_DATA_CREDIT	VC0 Posted Data Credits. The number of initial posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.3.3.1.78 Segmented-Buffer VC0 Non-Posted Receive Queue Control. (VC0_NP_RX_Q_CTRL_OFF)

11.3.3.1.78.1 Offset

Register	Offset
VC0_NP_RX_Q_CTRL_OFF	74Ch

11.3.3.1.78.2 Diagram



11.3.3.1.78.3 Fields

Field	Function
31-28 RESERVED7	Reserved. Note: This register field is sticky.
27-26 VC0_NP_DATA_SCALE	VC0 Scale Non-Posted Data Credites. Note: This register field is sticky.
25-24 VC0_NP_HDR_SCALE	VC0 Scale Non-Posted Header Credites. Note: This register field is sticky.
23-21 VC0_NP_TLP_Q_MODE	Reserved. Note: This register field is sticky.
20 RESERVED6	Reserved. Note: This register field is sticky.

Table continues on the next page...

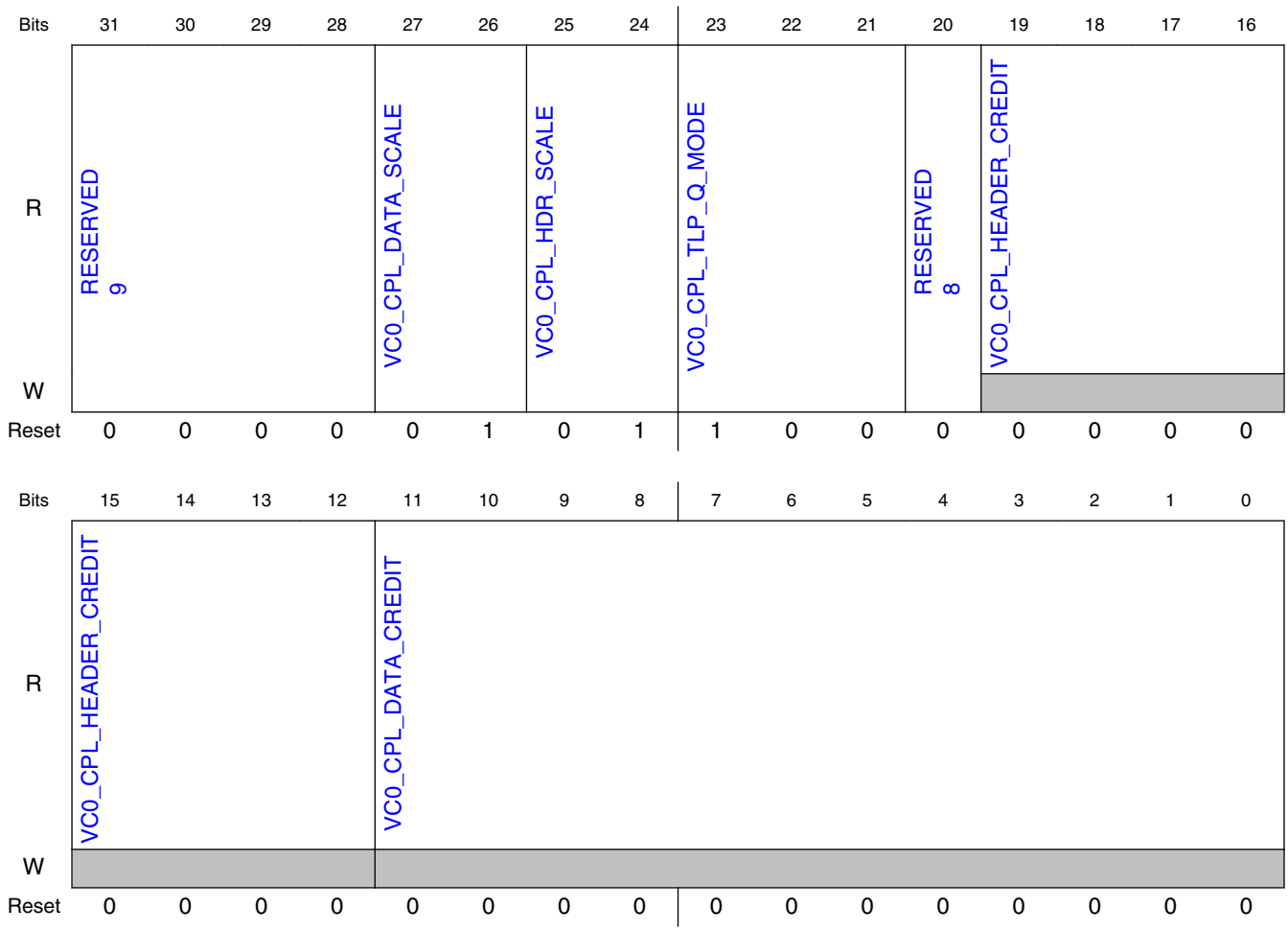
Field	Function
19-12 VC0_NP_HEADER_CREDIT	VC0 Non-Posted Header Credits. The number of initial non-posted header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_NP_DATA_CREDIT	VC0 Non-Posted Data Credits. The number of initial non-posted data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.3.3.1.79 Segmented-Buffer VC0 Completion Receive Queue Control. (VC0_CPL_RX_Q_CTRL_OFF)

11.3.3.1.79.1 Offset

Register	Offset
VC0_CPL_RX_Q_CTRL_OFF	750h

11.3.3.1.79.2 Diagram



11.3.3.1.79.3 Fields

Field	Function
31-28 RESERVED9	Reserved. Note: This register field is sticky.
27-26 VC0_CPL_DATA_SCALE	VC0 Scale CPL Data Credites. Note: This register field is sticky.
25-24 VC0_CPL_HDR_SCALE	VC0 Scale CPL Header Credites. Note: This register field is sticky.
23-21 VC0_CPL_TLP_Q_MODE	Reserved. Note: This register field is sticky.
20	Reserved. Note: This register field is sticky.

Table continues on the next page...

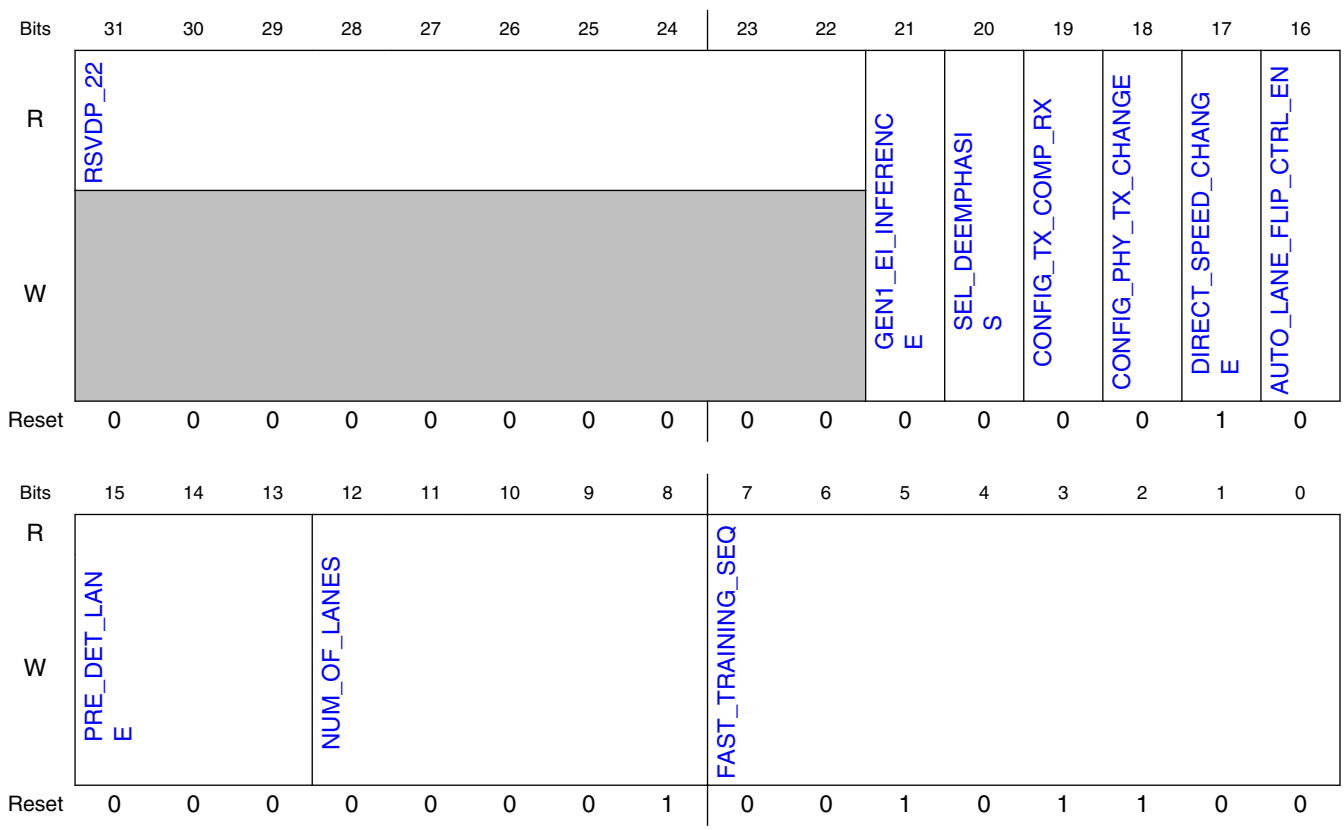
Field	Function
RESERVED8	
19-12 VC0_CPL_HEADER_CREDIT	VC0 Completion Header Credits. The number of initial Completion header credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.
11-0 VC0_CPL_DATA_CREDIT	VC0 Completion Data Credits. The number of initial Completion data credits for VC0, used only in the segmented-buffer configuration. Note: The access attributes of this field are as follows: - Dbi: R (sticky) Note: This register field is sticky.

11.3.3.1.80 Link Width and Speed Change Control Register. (GEN2_CTRL_OFF)

11.3.3.1.80.1 Offset

Register	Offset
GEN2_CTRL_OFF	80Ch

11.3.3.1.80.2 Diagram



11.3.3.1.80.3 Fields

Field	Function
31-22 RSVDP_22	Reserved for future use.
21 GEN1_EI_INFERENCE	Electrical Idle Inference Mode at Gen1 Rate. Programmable mode to determine inferred electrical idle (EI) in Recovery.Speed or Loopback.Active (as slave) state at Gen1 speed by looking for a "1" value on RxElecIdle instead of looking for a "0" on RxValid. If the PHY fails to deassert the RxValid signal in Recovery.Speed or Loopback.Active (because of corrupted EIOS for example), then EI cannot be inferred successfully in the controller by just detecting the condition RxValid=0. - 0: Use RxElecIdle signal to infer Electrical Idle - 1: Use RxValid signal to infer Electrical Idle Note: This register field is sticky.
20 SEL_DEEMPHASIS	Used to set the de-emphasis level for upstream ports. This bit selects the level of de-emphasis the link operates at. - 0: -6 dB - 1: -3.5 dB This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
19 CONFIG_TX_COMP_RX	Config Tx Compliance Receive Bit. When set to 1, signals LTSSM to transmit TS ordered sets with the compliance receive bit assert (equal to "1"). This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
18 CONFIG_PHY_TX_CHANGE	Config PHY Tx Swing. Controls the PHY transmitter voltage swing level. The controller drives the mac_phy_txswing output from this register bit field. - 0: Full Swing - 1: Low Swing This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
17 DIRECT_SPEED_CHANGE	Directed Speed Change. Writing "1" to this field instructs the LTSSM to initiate a speed change to Gen2 or Gen3 after the link is initialized at Gen1 speed. When the speed change occurs, the controller will clear the contents of this field; and a read to this field by your software will return a "0". To manually initiate the speed change: - Write to LINK_CONTROL2_LINK_STATUS2_REG[PCIE_CAP_TARGET_LINK_SPEED] in the local device, deassert this field, and then assert this field. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W
16 AUTO_LANE_FLIP_CTRL_EN	Enable Auto flipping of the lanes. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
15-13 PRE_DET_LANE	Predetermined Lane for Auto Flip. This field defines which physical lane is connected to logical Lane0 by the flip operation performed in Detect. Allowed values are: - 3'b000: Connect logical Lane0 to physical lane 0 - 3'b001: Connect logical Lane0 to physical lane 1 - 3'b010: Connect logical Lane0 to physical lane 3 - 3'b011: Connect logical Lane0 to physical lane 7 - 3'b100: Connect logical Lane0 to physical lane 15 This field is used to restrict the receiver detect procedure to a particular lane when the default detect and polling procedure performed on all lanes cannot be successful. A notable example of when it is useful to program this field to a value different from the default, is when a lane is asymmetrically broken, that is, it is detected in Detect LTSSM state but it cannot exit Electrical Idle in Polling LTSSM state. Note: This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
12-8 NUM_OF_LANES	Predetermined Number of Lanes. Defines the number of lanes which are connected and not bad. Used to limit the effective link width to ignore 'broken" or "unused" lanes that detect a receiver. Indicates the number of lanes to check for exit from Electrical Idle in Polling.Active and L2.Idle. It is possible that the LTSSM might detect a receiver on a bad or broken lane during the Detect Substate. However, it is also possible that such a lane might also fail to exit Electrical Idle and therefore prevent a valid link from being configured. This value is referred to as the "Predetermined Number of Lanes" in section 4.2.6.2.1 of the PCI Express Base 3.0 Specification, revision 1.0. Encoding is as follows: - 0x01: 1 lane - 0x02: 2 lanes - 0x03: 3 lanes - .. When you have unused lanes in your system, then you must change the value in this register to reflect the number of lanes. You must also change the value in the "Link Mode Enable" field of PORT_LINK_CTRL_OFF. The value in this register is normally the same as the encoded value in

Table continues on the next page...

Field	Function
	PORT_LINK_CTRL_OFF. If you find that one of your used lanes is bad then you must reduce the value in this register. For more information, see "How to Tie Off Unused Lanes." For information on upsizing and downsizing the link width, see "Link Establishment." This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.
7-0 FAST_TRAINING_SEQ	Sets the Number of Fast Training Sequences (N_FTS) that the controller advertises as its N_FTS during Gen2 or Gen3 link training. This value is used to inform the link partner about the PHY's ability to recover synchronization after a low power state. The number should be provided by the PHY vendor. Do not set N_FTS to zero; doing so can cause the LTSSM to go into the recovery state when exiting from L0s. This field is reserved (fixed to '0') for M-PCIe. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

11.3.3.1.81 PHY Status Register. (PHY_STATUS_OFF)

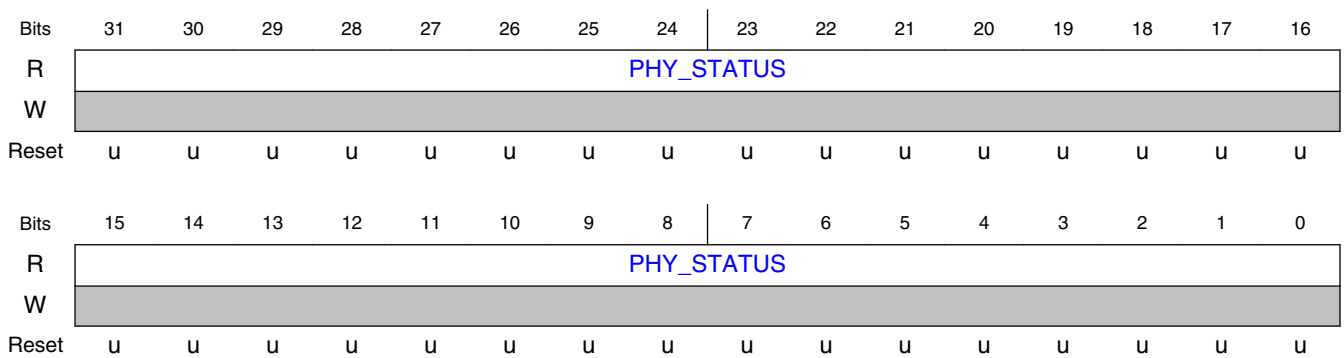
11.3.3.1.81.1 Offset

Register	Offset
PHY_STATUS_OFF	810h

11.3.3.1.81.2 Function

PHY Status Register. Memory mapped register from phy_cfg_status GPIO input pins.

11.3.3.1.81.3 Diagram



11.3.3.1.81.4 Fields

Field	Function
31-0 PHY_STATUS	PHY Status. Data received directly from the phy_cfg_status bus. These is a GPIO register reflecting the values on the static phy_cfg_status input signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband status signalling

Field	Function
	requirements that you have for your PHY. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.3.3.1.82 PHY Control Register. (PHY_CONTROL_OFF)

11.3.3.1.82.1 Offset

Register	Offset
PHY_CONTROL_OFF	814h

11.3.3.1.82.2 Function

PHY Control Register. Memory mapped register to cfg_phy_control GPIO output pins.

11.3.3.1.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PHY_CONTROL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PHY_CONTROL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.82.4 Fields

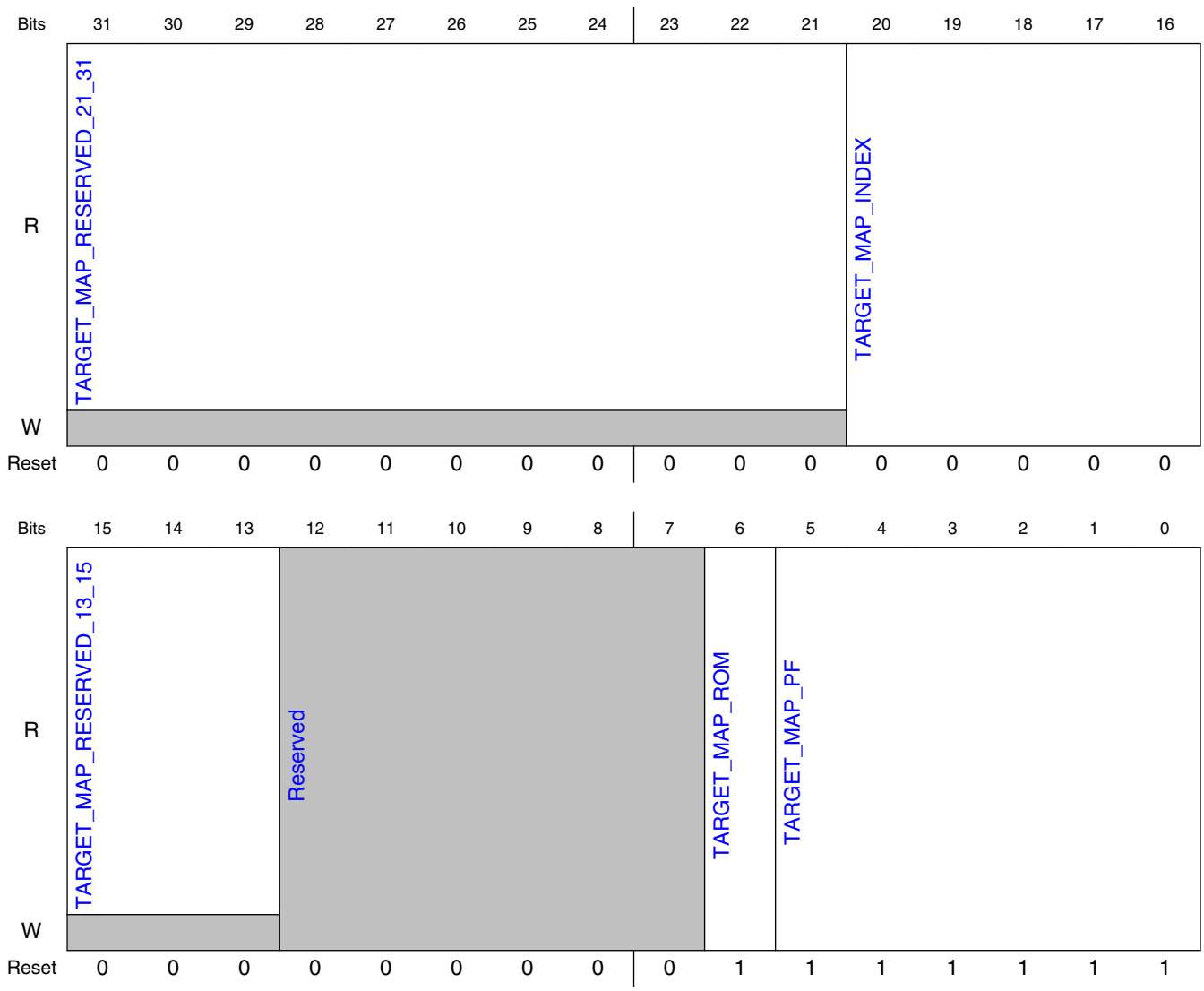
Field	Function
31-0 PHY_CONTROL	PHY Control. Data sent directly to the cfg_phy_control bus. These is a GPIO register driving the values on the static cfg_phy_control output signals. The usage is left completely to the user and does not in any way influence controller functionality. You can use it for any static sideband control signalling requirements that you have for your PHY. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.3.3.1.83 Programmable Target Map Control Register. (TRGT_MAP_CTRL_OFF)

11.3.3.1.83.1 Offset

Register	Offset
TRGT_MAP_CTRL_OFF	81Ch

11.3.3.1.83.2 Diagram



11.3.3.1.83.3 Fields

Field	Function
31-21 TARGET_MAP_RESERVED_21_31	Reserved. Note: The access attributes of this field are as follows: - Dbi: R (sticky)
20-16 TARGET_MAP_INDEX	The number of the PF Function on which the Target Values are set. This register does not respect the Byte Enable setting. any write will affect all register bits.
15-13 TARGET_MAP_RESERVED_13_15	Reserved. Note: The access attributes of this field are as follows: - Dbi: R (sticky)
12-7 —	Reserved.
6 TARGET_MAP_ROM	Target Value for the ROM page of the PF Function selected by the index number. This register does not respect the Byte Enable setting. any write will affect all register bits.
5-0 TARGET_MAP_PF	Target Values for each BAR on the PF Function selected by the index number. This register does not respect the Byte Enable setting. any write will affect all register bits.

11.3.3.1.84 Integrated MSI Reception Module (iMRM) Address Register. (MSI_CTRL_ADDR_OFF)

11.3.3.1.84.1 Offset

Register	Offset
MSI_CTRL_ADDR_OFF	820h

11.3.3.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.84.3 Fields

Field	Function
31-0 MSI_CTRL_ADDR	Integrated MSI Reception Module Address. System specified address for MSI memory write transaction termination. Within the AXI Bridge, every received Memory Write request is examined to see if it targets the MSI Address that has been specified in this register; and also to see if it satisfies the definition of an MSI interrupt request. When these conditions are satisfied the Memory Write request is marked as an MSI request. Note: This register field is sticky.

11.3.3.1.85 Integrated MSI Reception Module Upper Address Register. (MSI_CTRL_UPPER_ADDR_OFF)

11.3.3.1.85.1 Offset

Register	Offset
MSI_CTRL_UPPER_ADDR_OFF	824h

11.3.3.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_UPPER_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_UPPER_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.85.3 Fields

Field	Function
31-0 MSI_CTRL_UPPER_ADDR	Integrated MSI Reception Module Upper Address. System specified upper address for MSI memory write transaction termination. Allows functions to support a 64-bit MSI address. Note: This register field is sticky.

11.3.3.1.86 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_0_EN_OFF)

11.3.3.1.86.1 Offset

Register	Offset
MSI_CTRL_INT_0_EN_OFF	828h

11.3.3.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_0_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_0_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.86.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.87 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_0_MASK_OFF)

11.3.3.1.87.1 Offset

Register	Offset
MSI_CTRL_INT_0_MASK_OFF	82Ch

11.3.3.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_0_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_0_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.87.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.88 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_0_STATUS_OFF)

11.3.3.1.88.1 Offset

Register	Offset
MSI_CTRL_INT_0_STATUS_OFF	830h

11.3.3.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_0_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_0_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.88.3 Fields

Field	Function
31-0 MSI_CTRL_INT_0_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.89 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_1_EN_OFF)

11.3.3.1.89.1 Offset

Register	Offset
MSI_CTRL_INT_1_EN_OFF	834h

11.3.3.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_1_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_1_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.89.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.90 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_1_MASK_OFF)

11.3.3.1.90.1 Offset

Register	Offset
MSI_CTRL_INT_1_MASK_OFF	838h

11.3.3.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_1_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_1_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.90.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.91 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_1_STATUS_OFF)

11.3.3.1.91.1 Offset

Register	Offset
MSI_CTRL_INT_1_STATUS_OFF	83Ch

11.3.3.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_1_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_1_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.91.3 Fields

Field	Function
31-0 MSI_CTRL_INT_1_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.92 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_2_EN_OFF)

11.3.3.1.92.1 Offset

Register	Offset
MSI_CTRL_INT_2_EN_OFF	840h

11.3.3.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_2_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_2_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.92.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.93 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_2_MASK_OFF)

11.3.3.1.93.1 Offset

Register	Offset
MSI_CTRL_INT_2_MASK_OFF	844h

11.3.3.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_2_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_2_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.93.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.94 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_2_STATUS_OFF)

11.3.3.1.94.1 Offset

Register	Offset
MSI_CTRL_INT_2_STATUS_OFF	848h

11.3.3.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_2_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_2_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.94.3 Fields

Field	Function
31-0 MSI_CTRL_INT_2_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.95 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_3_EN_OFF)

11.3.3.1.95.1 Offset

Register	Offset
MSI_CTRL_INT_3_EN_OFF	84Ch

11.3.3.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_3_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_3_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.95.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.96 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_3_MASK_OFF)

11.3.3.1.96.1 Offset

Register	Offset
MSI_CTRL_INT_3_MASK_OFF	850h

11.3.3.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_3_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_3_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.96.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.97 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_3_STATUS_OFF)

11.3.3.1.97.1 Offset

Register	Offset
MSI_CTRL_INT_3_STATUS_OFF	854h

11.3.3.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_3_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_3_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.97.3 Fields

Field	Function
31-0 MSI_CTRL_INT_3_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.98 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_4_EN_OFF)

11.3.3.1.98.1 Offset

Register	Offset
MSI_CTRL_INT_4_EN_OFF	858h

11.3.3.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_4_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_4_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.98.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.99 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_4_MASK_OFF)

11.3.3.1.99.1 Offset

Register	Offset
MSI_CTRL_INT_4_MASK_OFF	85Ch

11.3.3.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_4_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_4_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.99.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.100 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_4_STATUS_OFF)

11.3.3.1.100.1 Offset

Register	Offset
MSI_CTRL_INT_4_STATUS_OFF	860h

11.3.3.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_4_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_4_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.100.3 Fields

Field	Function
31-0 MSI_CTRL_INT_4_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.101 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_5_EN_OFF)

11.3.3.1.101.1 Offset

Register	Offset
MSI_CTRL_INT_5_EN_OFF	864h

11.3.3.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_5_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_5_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.101.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.102 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_5_MASK_OFF)

11.3.3.1.102.1 Offset

Register	Offset
MSI_CTRL_INT_5_MASK_OFF	868h

11.3.3.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_5_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_5_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.102.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.103 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_5_STATUS_OFF)

11.3.3.1.103.1 Offset

Register	Offset
MSI_CTRL_INT_5_STATUS_OFF	86Ch

11.3.3.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_5_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_5_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.103.3 Fields

Field	Function
31-0 MSI_CTRL_INT_5_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.104 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_6_EN_OFF)

11.3.3.1.104.1 Offset

Register	Offset
MSI_CTRL_INT_6_EN_OFF	870h

11.3.3.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_6_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_6_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.104.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.105 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_6_MASK_OFF)

11.3.3.1.105.1 Offset

Register	Offset
MSI_CTRL_INT_6_MASK_OFF	874h

11.3.3.1.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_6_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_6_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.105.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.106 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_6_STATUS_OFF)

11.3.3.1.106.1 Offset

Register	Offset
MSI_CTRL_INT_6_STATUS_OFF	878h

11.3.3.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_6_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_6_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.106.3 Fields

Field	Function
31-0 MSI_CTRL_INT_6_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.107 Integrated MSI Reception Module Interrupt#i Enable Register. (MSI_CTRL_INT_7_EN_OFF)

11.3.3.1.107.1 Offset

Register	Offset
MSI_CTRL_INT_7_EN_OFF	87Ch

11.3.3.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_7_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_7_EN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.107.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_EN	MSI Interrupt#i Enable. Specifies which interrupts are enabled. When an MSI is received from a disabled interrupt, no status bit gets set in MSI controller interrupt status register. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.108 Integrated MSI Reception Module Interrupt#i Mask Register. (MSI_CTRL_INT_7_MASK_OFF)

11.3.3.1.108.1 Offset

Register	Offset
MSI_CTRL_INT_7_MASK_OFF	880h

11.3.3.1.108.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_CTRL_INT_7_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_CTRL_INT_7_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.108.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_MASK	MSI Interrupt#i Mask. Allows enabled interrupts to be masked. When an MSI is received for a masked interrupt, the corresponding status bit gets set in the interrupt status register but the msi_ctrl_int output is not set HIGH. Each bit corresponds to a single MSI Interrupt Vector. Note: This register field is sticky.

11.3.3.1.109 Integrated MSI Reception Module Interrupt#i Status Register. (MSI_CTRL_INT_7_STATUS_OFF)

11.3.3.1.109.1 Offset

Register	Offset
MSI_CTRL_INT_7_STATUS_OFF	884h

11.3.3.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MSI_CTRL_INT_7_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MSI_CTRL_INT_7_STATUS															
W	W1C															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.109.3 Fields

Field	Function
31-0 MSI_CTRL_INT_7_STATUS	MSI Interrupt#i Status. When an MSI is detected for EP#i, one bit in this register is set. The decoding of the data payload of the MSI Memory Write request determines which bit gets set. A status is bit is cleared by writing a 1 to the bit. Each bit corresponds to a single MSI Interrupt Vector.

11.3.3.1.110 Integrated MSI Reception Module General Purpose IO Register. (MSI_GPIO_IO_OFF)

11.3.3.1.110.1 Offset

Register	Offset
MSI_GPIO_IO_OFF	888h

11.3.3.1.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MSI_GPIO_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MSI_GPIO_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.110.3 Fields

Field	Function
31-0 MSI_GPIO_REG	MSI GPIO Register. The contents of this register drives the top-level GPIO msi_ctrl_io[31:0] Note: This register field is sticky.

11.3.3.1.111 RADM clock gating enable control register. (CLOCK_GATING_CTRL_OFF)

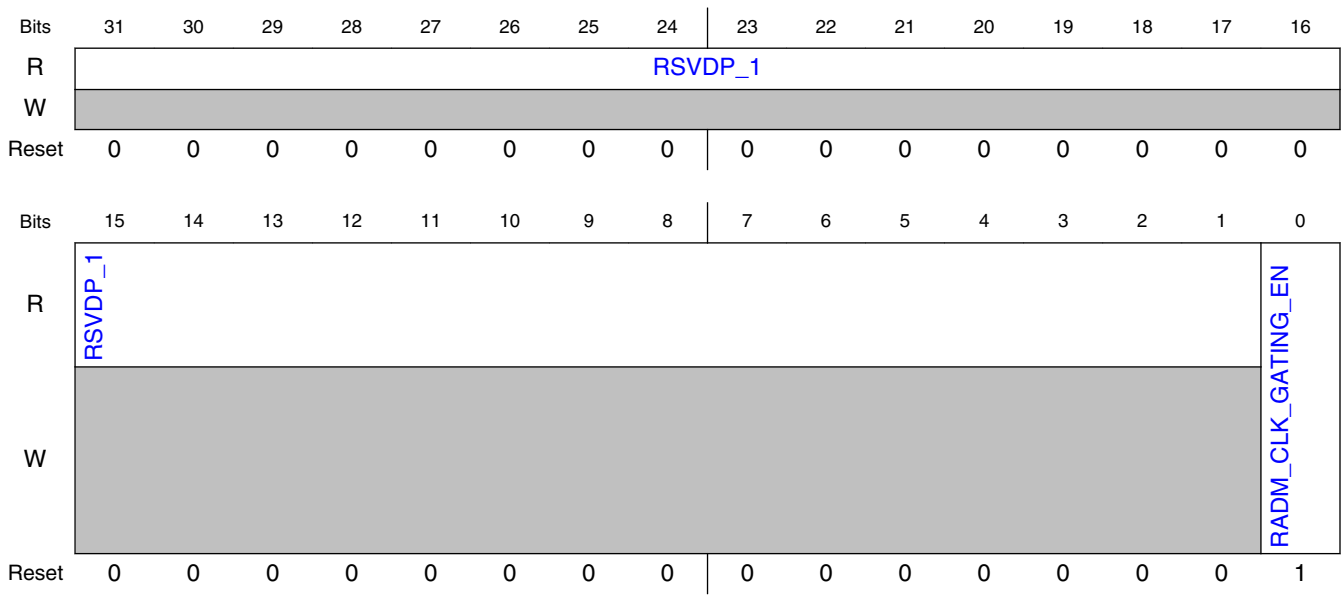
11.3.3.1.111.1 Offset

Register	Offset
CLOCK_GATING_CTRL_OFF	88Ch

11.3.3.1.111.2 Function

RADM clock gating enable control register. Using this register you can disable the RADM clock gating feature. The DWC_pcie_clk_rst.v modules uses the en_radm_clk_g output to gate core_clk and create the radm_clk_g clock input. The controller de-asserts the en_radm_clk_g output when there is no Rx traffic, Rx queues and pre/post-queue pipelines are empty, RADM completion LUT is empty, and there are no FLR actions pending. You must set the RADM_CLK_GATING_EN field to enable this functionality; otherwise the en_radm_clk_g output will always be set to '1'.

11.3.3.1.111.3 Diagram



11.3.3.1.111.4 Fields

Field	Function
31-1 RSVDP_1	Reserved for future use.
0 RADM_CLK_GATING_EN	Enable Radm clock gating feature. - 0: Disable - 1: Enable(default)

11.3.3.1.112 Order Rule Control Register. (ORDER_RULE_CTRL_OFF)

11.3.3.1.112.1 Offset

Register	Offset
ORDER_RULE_CTRL_OFF	8B4h

11.3.3.1.112.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_16															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CPL_PASS_P								NP_PASS_P							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.112.3 Fields

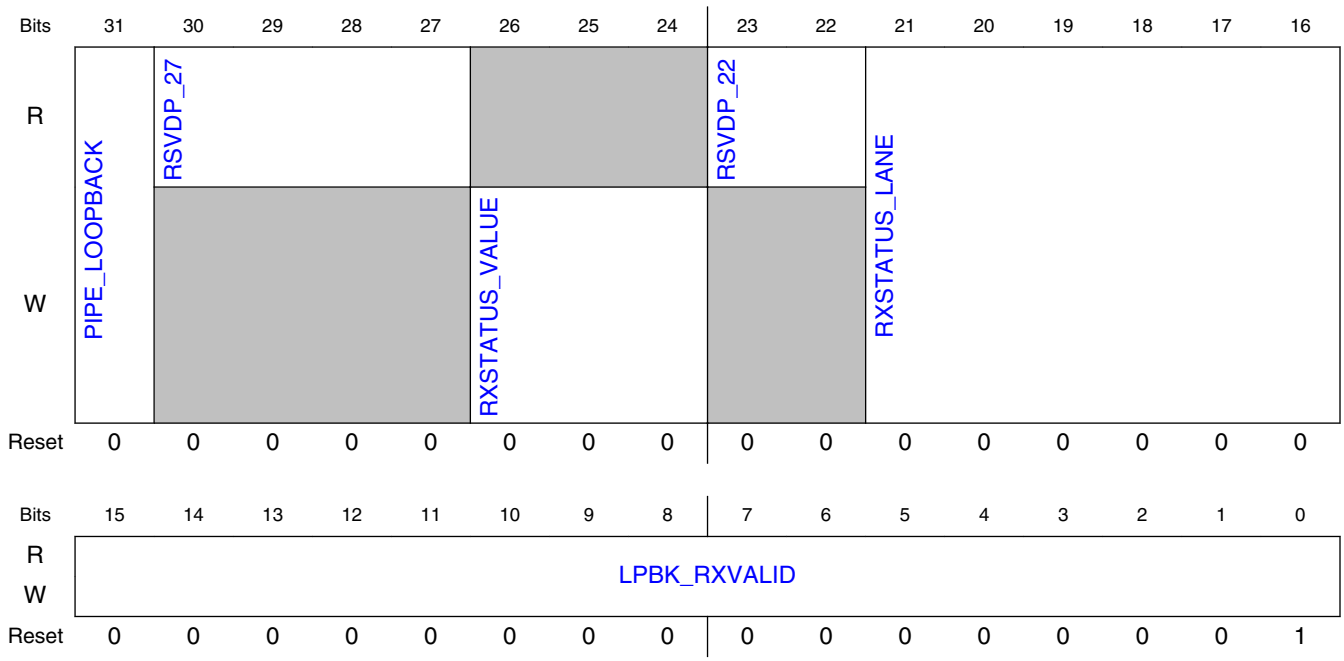
Field	Function
31-16 RSVDP_16	Reserved for future use.
15-8 CPL_PASS_P	Completion Passing Posted Ordering Rule Control. Determines if CPL can pass halted P queue. - 0: CPL can not pass P (recommended) - 1: CPL can pass P
7-0 NP_PASS_P	Non-Posted Passing Posted Ordering Rule Control. Determines if NP can pass halted P queue. - 0 : NP can not pass P (recommended). - 1 : NP can pass P

11.3.3.1.113 PIPE Loopback Control Register. (PIPE_LOOPBACK_CONTROL_OFF)

11.3.3.1.113.1 Offset

Register	Offset
PIPE_LOOPBACK_CONTROL_OFF	8B8h

11.3.3.1.113.2 Diagram



11.3.3.1.113.3 Fields

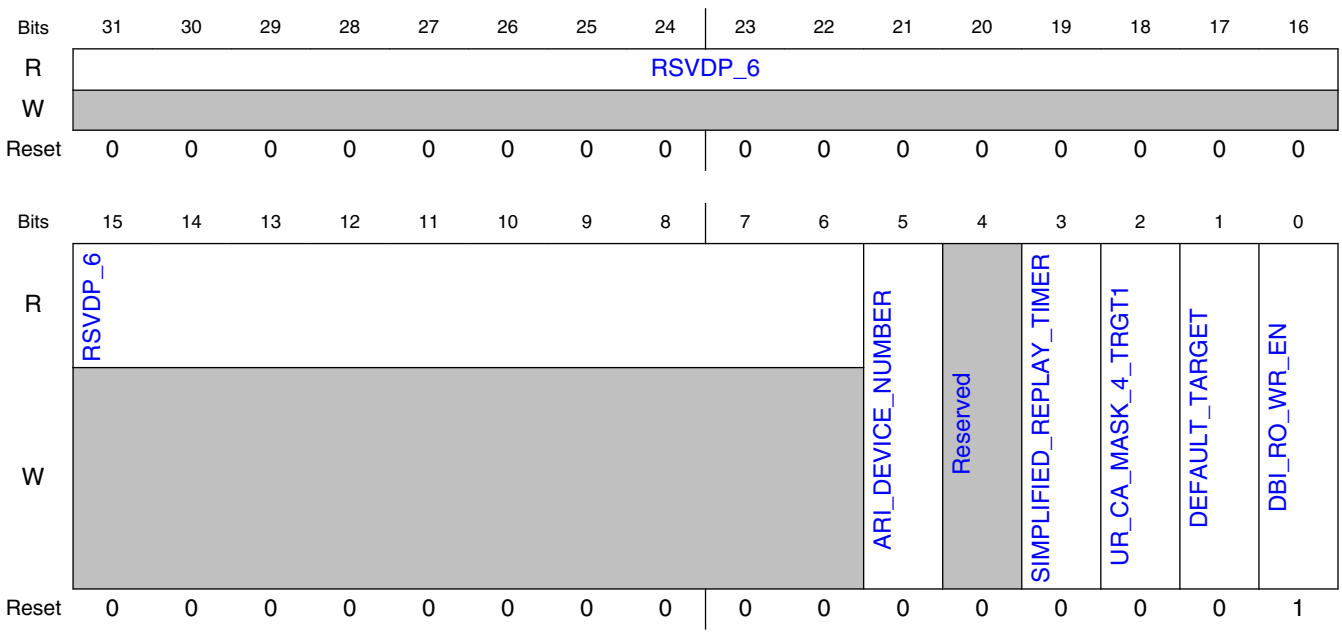
Field	Function
31 PIPE_LOOPBACK	PIPE Loopback Enable. Indicates RMMI Loopback if M-PCIe. Note: This register field is sticky.
30-27 RSVDP_27	Reserved for future use.
26-24 RXSTATUS_VALUE	RXSTATUS_VALUE is an internally reserved field. Do not use.
23-22 RSVDP_22	Reserved for future use.
21-16 RXSTATUS_LANE	RXSTATUS_LANE is an internally reserved field. Do not use. Note: This register field is sticky.
15-0 LPBK_RXVALID	LPBK_RXVALID is an internally reserved field. Do not use. Note: This register field is sticky.

11.3.3.1.114 DBI Read-Only Write Enable Register. (MISC_CONTROL_1_OFF)

11.3.3.1.114.1 Offset

Register	Offset
MISC_CONTROL_1_OFF	8BCh

11.3.3.1.114.2 Diagram



11.3.3.1.114.3 Fields

Field	Function
31-6 RSVDP_6	Reserved for future use.
5 ARI_DEVICE_NUMBER	When ARI is enabled, this field enables use of the device ID. Note: This register field is sticky.
4 —	Reserved.
3	Enables Simplified Replay Timer (Gen4). For more details, see "Transmit Replay" in the Controller Operations chapter of the Databook. Simplified Replay Timer Values are: - A value from 24,000 to 31,000

Table continues on the next page...

Field	Function
SIMPLIFIED_REPLAY_TIMER	Symbol Times when Extended Synch is 0b. - A value from 80,000 to 100,000 Symbol Times when Extended Synch is 1b. Must not be changed while link is in use. Note: This register field is sticky.
2 UR_CA_MASK_4_TRGT1	This field only applies to request TLPs (with UR filtering status) that you have chosen to forward to the application (when you set DEFAULT_TARGET in this register). - When you set this field to '1', the core suppresses error logging, Error Message generation, and CPL generation (for non-posted requests). Note: This register field is sticky.
1 DEFAULT_TARGET	Default target a received IO or MEM request with UR/CA/CRS is sent to by the controller. - 0: The controller drops all incoming I/O or MEM requests (after corresponding error reporting). A completion with UR status will be generated for non-posted requests. - 1: The controller forwards all incoming I/O or MEM requests with UR/CA/CRS status to your application. Note: This register field is sticky.
0 DBI_RO_WR_EN	Write to RO Registers Using DBI. When you set this field to "1", then some RO and HwInit bits are writable from the local application through the DBI. Note: This register field is sticky.

11.3.3.1.115 UpConfigure Multi-lane Control Register. (MULTI_LANE_CONTROL_OFF)

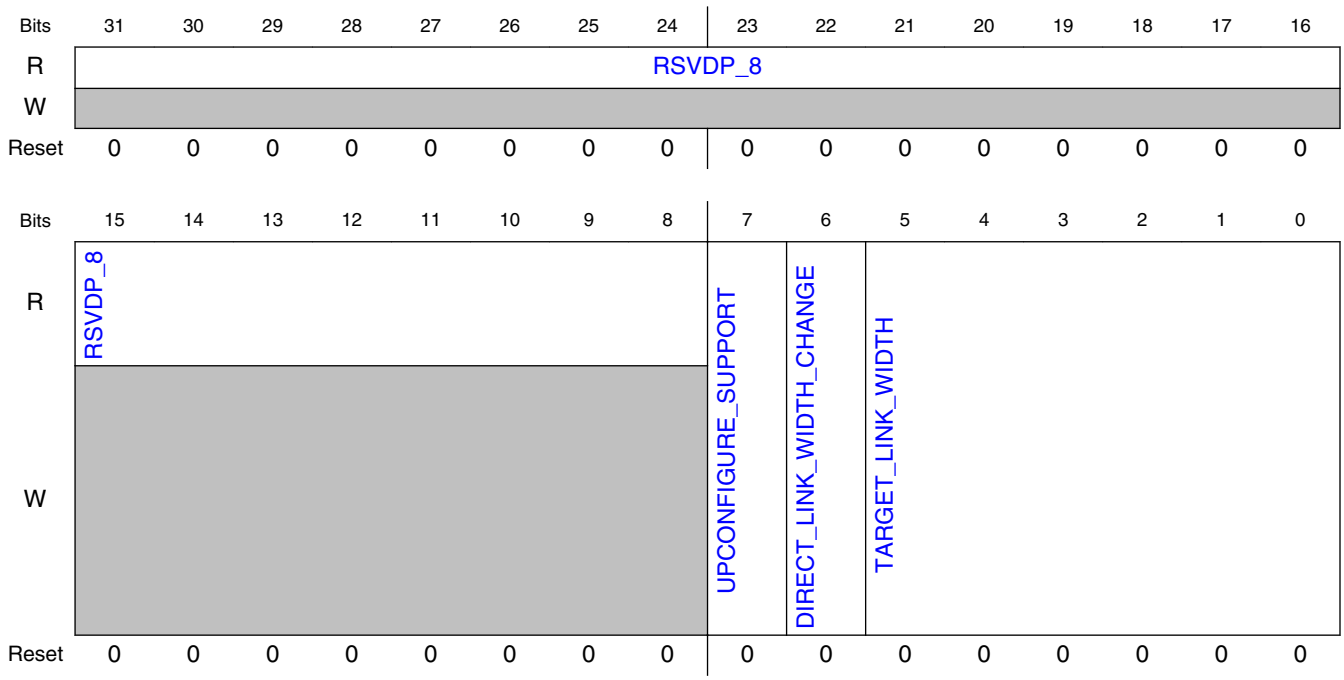
11.3.3.1.115.1 Offset

Register	Offset
MULTI_LANE_CONTROL_OFF	8C0h

11.3.3.1.115.2 Function

UpConfigure Multi-lane Control Register. Used when upsizing or downsizing the link width through Configuration state without bringing the link down. For more details, see the "Link Establishment" section in the "Controller Operations" chapter of the Databook.

11.3.3.1.115.3 Diagram



11.3.3.1.115.4 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7 UPCONFIGURE_SUPPORT	Upconfigure Support. The controller sends this value as the Link Upconfigure Capability in TS2 Ordered Sets in Configuration.Complete state. This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.
6 DIRECT_LINK_WIDTH_CHANGE	Directed Link Width Change. The controller always moves to Configuration state through Recovery state when this bit is set to '1'. - If the upconfigure_capable variable is '1' and the PCIE_CAP_HW_AUTO_WIDTH_DISABLE bit in LINK_CONTROL_LINK_STATUS_REG is '0', the controller starts upconfigure or autonomous width downsizing (to the TARGET_LINK_WIDTH value) in the Configuration state. - If TARGET_LINK_WIDTH value is 0x0, the controller does not start upconfigure or autonomous width downsizing in the Configuration state. The controller self-clears this field when the controller accepts this request. This field is reserved (fixed to '0') for M-PCIe.
5-0 TARGET_LINK_WIDTH	Target Link Width. Values correspond to: - 6'b000000: Core does not start upconfigure or autonomous width downsizing in the Configuration state. - 6'b000001: x1 - 6'b000010: x2 - 6'b000100: x4 - 6'b001000: x8 - 6'b010000: x16 - 6'b100000: x32 This field is reserved (fixed to '0') for M-PCIe.

11.3.3.1.116 PHY Interoperability Control Register. (PHY_INTEROP_CTRL_OFF)

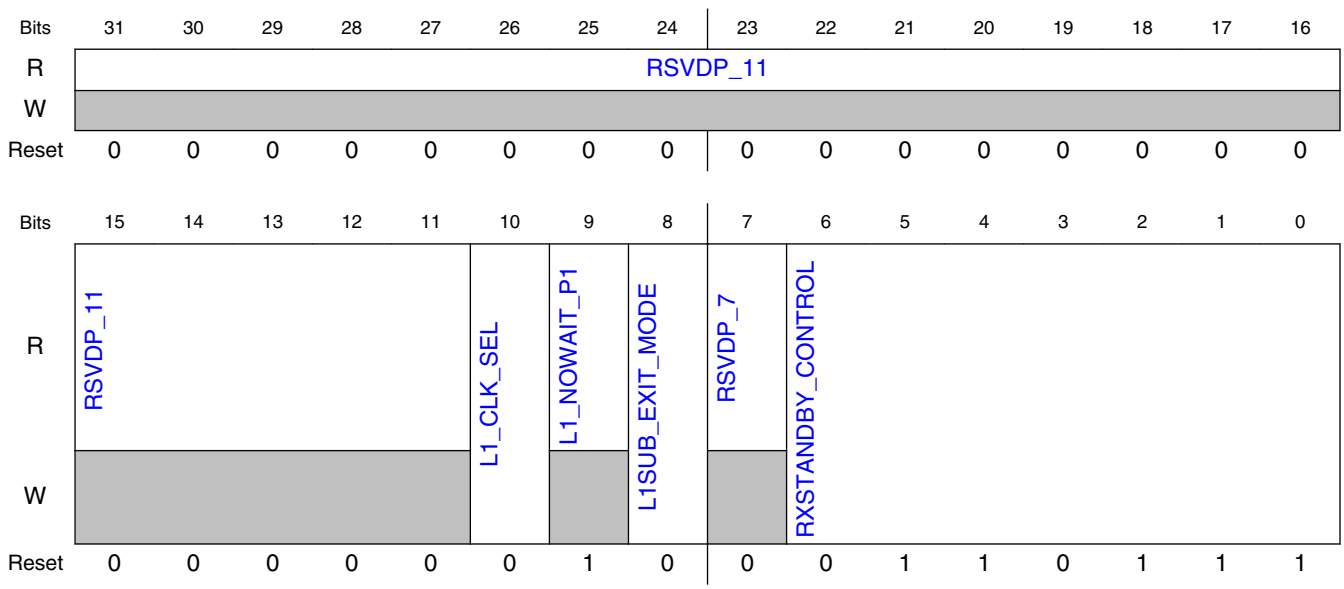
11.3.3.1.116.1 Offset

Register	Offset
PHY_INTEROP_CTRL_OFF	8C4h

11.3.3.1.116.2 Function

PHY Interoperability Control Register. This register is reserved for internal use. You should not write to this register and change the default unless specifically instructed by Synopsys support.

11.3.3.1.116.3 Diagram



11.3.3.1.116.4 Fields

Field	Function
31-11 RSVDP_11	Reserved for future use.
10 L1_CLK_SEL	L1 Clock control bit. - 1: Controller does not request aux_clk switch and core_clk gating in L1. - 0: Controller requests aux_clk switch and core_clk gating in L1. Note: This register field is sticky.
9 L1_NOWAIT_P1	L1 entry control bit. - 1: Core does not wait for PHY to acknowledge transition to P1 before entering L1. - 0: Core waits for the PHY to acknowledge transition to P1 before entering L1. Note: The access attributes of this field are as follows: - Dbi: R/W (sticky) Note: This register field is sticky.

Table continues on the next page...

Field	Function
8 L1SUB_EXIT_MODE	L1 Exit Control Using phy_mac_pclkack_n. - 1: Core exits L1 without waiting for the PHY to assert phy_mac_pclkack_n. - 0: Core waits for the PHY to assert phy_mac_pclkack_n before exiting L1. Note: This register field is sticky.
7 RSVDP_7	Reserved for future use.
6-0 RXSTANDBY_CONTROL	Rxstandby Control. Bits 0..5 determine if the controller asserts the RxStandby signal (mac_phy_rxstandby) in the indicated condition. Bit 6 enables the controller to perform the RxStandby/ RxStandbyStatus handshake. - [0]: Rx EIOS and subsequent T TX-IDLE-MIN - [1]: Rate Change - [2]: Inactive lane for upconfigure/downconfigure - [3]: PowerDown=P1orP2 - [4]: RxL0s.Idle - [5]: EI Infer in L0 - [6]: Execute RxStandby/RxStandbyStatus Handshake This field is reserved (fixed to '0') for M-PCIe. Note: This register field is sticky.

11.3.3.1.117 TRGT_CPL_LUT Delete Entry Control register. (TRGT_CPL_LUT_DELETE_ENTRY_OFF)

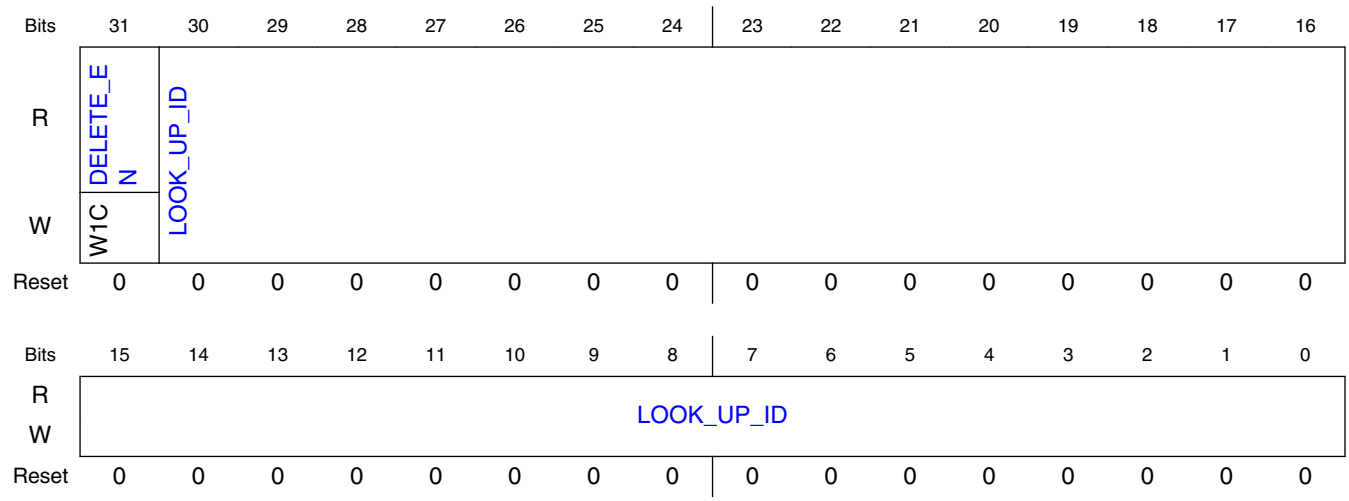
11.3.3.1.117.1 Offset

Register	Offset
TRGT_CPL_LUT_DELETE_ENTRY_OFF	8C8h

11.3.3.1.117.2 Function

TRGT_CPL_LUT Delete Entry Control register. Using this register you can delete one entry in the target completion LUT. You should only use this register when you know that your application will never send the completion because of an FLR or any other reason. Note:: The target completion LUT (and associated target completion timeout event) is watching for completions (from your application on XALI0/1/2 or AXI master read channel) corresponding to previously received non-posted requests from the PCIe wire.

11.3.3.1.117.3 Diagram



11.3.3.1.117.4 Fields

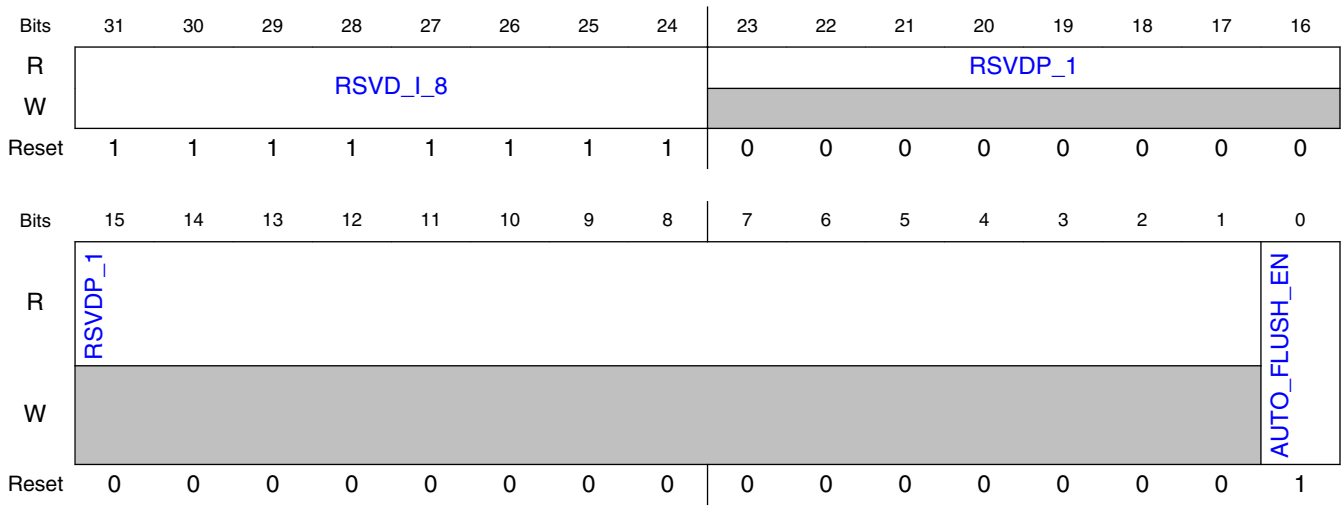
Field	Function
31 DELETE_EN	This is a one-shot bit. A '1' write to this bit triggers the deletion of the target completion LUT entry that is specified in the LOOK_UP_ID field. This is a self-clearing register field. Reading from this register field always returns a '0'.
30-0 LOOK_UP_ID	This number selects one entry to delete of the TRGT_CPL_LUT.

11.3.3.1.118 Link Reset Request Flush Control Register. (LINK_FLUSH_CONTROL_OFF)

11.3.3.1.118.1 Offset

Register	Offset
LINK_FLUSH_CONTROL_OFF	8CCh

11.3.3.1.118.2 Diagram



11.3.3.1.118.3 Fields

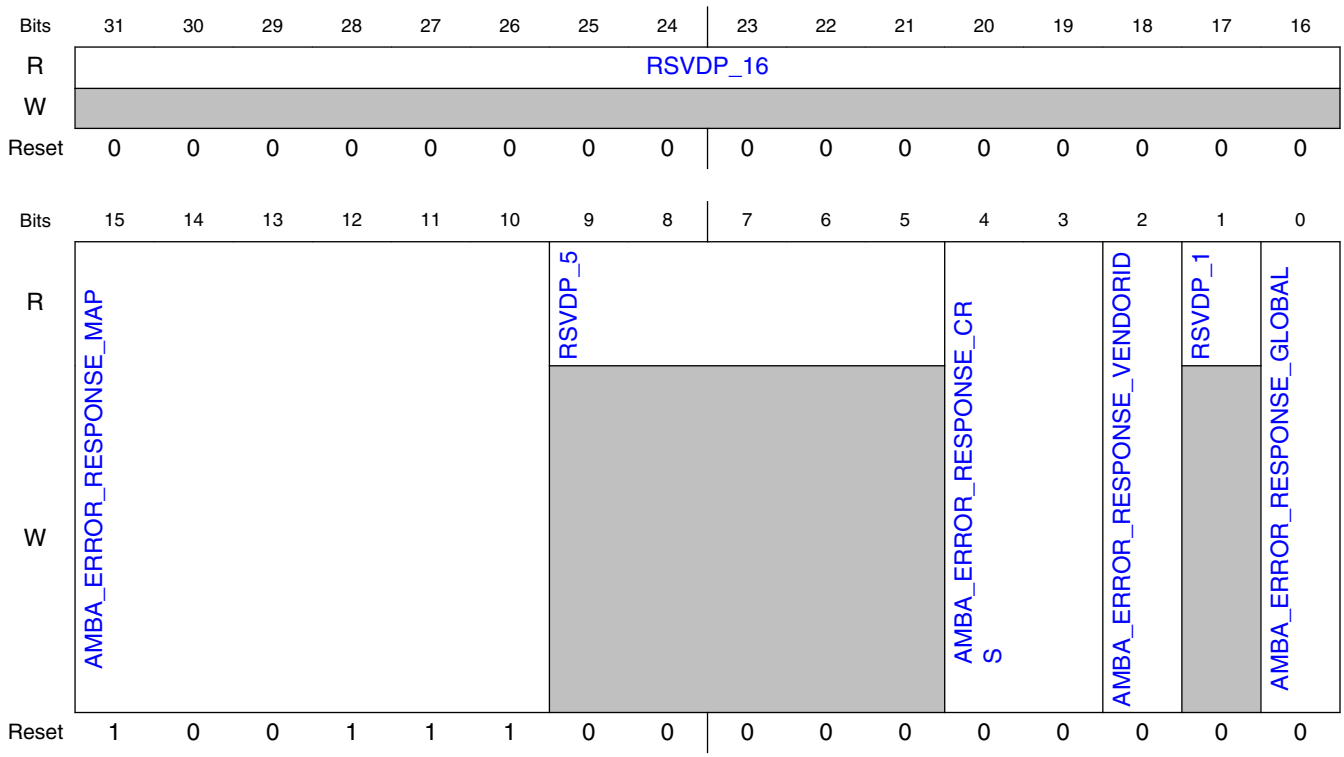
Field	Function
31-24 RSVD_I_8	This is an internally reserved field. Do not use. Note: This register field is sticky.
23-1 RSVDP_1	Reserved for future use.
0 AUTO_FLUSH_EN	Enables automatic flushing of pending requests before sending the reset request to the application logic to reset the PCIe controller and the AXI Bridge. The flushing process is initiated if any of the following events occur: - Hot reset request. A downstream port (DSP) can "hot reset" an upstream port (USP) by sending two consecutive TS1 ordered sets with the hot reset bit asserted. - Warm (Soft) reset request. Generated when exiting from D3 to D0 and cfg_pm_no_soft_rst=0. - Link down reset request. A high to low transition on smlh_req_rst_not indicates the link has gone down and the controller is requesting a reset. If you disable automatic flushing, your application is responsible for resetting the PCIe controller and the AXI Bridge. For more details see "Warm and Hot Resets" section in the Architecture chapter of the Databook. Note: This register field is sticky.

11.3.3.1.119 AXI Bridge Slave Error Response Register. (AMBA_ERROR_RESPONSE_DEFAULT_OFF)

11.3.3.1.119.1 Offset

Register	Offset
AMBA_ERROR_RESPONSE_DEFAULT_OFF	8D0h

11.3.3.1.119.2 Diagram



11.3.3.1.119.3 Fields

Field	Function
31-16 RSVDP_16	Reserved for future use.
15-10 AMBA_ERROR_RESPONSE_MAP	AXI Slave Response Error Map. Allows you to selectively map the errors received from the PCIe completion (for non-posted requests) to the AXI slave responses, slv_rresp or slv_bresp. The recommended setting is SLVERR. CRS is always mapped to OKAY. - [0] -- 0: UR (unsupported request) -> DECERR -- 1: UR (unsupported request) -> SLVERR - [1] -- 0: CRS (configuration retry status) -> DECERR -- 1: CRS (configuration retry status) -> SLVERR - [2] -- 0: CA (completer abort) -> DECERR -- 1: CA (completer abort) -> SLVERR - [3]: Reserved - [4]: Reserved - [5]: -- 0: Completion Timeout -> DECERR -- 1: Completion Timeout -> SLVERR The AXI bridge internally drops (processes internally but not passed to your application) a completion that has been marked by the Rx filter as UC or MLF, and does not pass its status directly down to the slave interface. It waits for a timeout and then signals "Completion Timeout" to the slave interface. The controller sets the AXI slave read databus to 0xFFFF for all error responses. Note: This register field is sticky.
9-5 RSVDP_5	Reserved for future use.
4-3 AMBA_ERROR_RESPONSE_CRS	CRS Slave Error Response Mapping. Determines the AXI slave response for CRS completions. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - always returns OKAY AXI: - 00: OKAY - 01: OKAY with all FFFF_FFFF data for all CRS completions - 10: OKAY with FFFF_0001 data for CRS completions to vendor ID read requests, OKAY with FFFF_FFFF data for all other CRS completions - 11: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) Note: This register field is sticky.

Table continues on the next page...

Field	Function
2 AMBA_ERROR_RESPONSE_VENDORID	Vendor ID Non-existent Slave Error Response Mapping. Determines the AXI slave response for errors on reads to non-existent Vendor ID register. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - 0: OKAY (with FFFF data). The controller ignores the setting in the bit when bit 0 of this register is '0'. - 1: ERROR AXI: - 0: OKAY (with FFFF data). - 1: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) Note: This register field is sticky.
1 RSVDP_1	Reserved for future use.
0 AMBA_ERROR_RESPONSE_GLOBAL	Global Slave Error Response Mapping. Determines the AXI slave response for all error scenarios on non-posted requests. For more details see "Error Handling" in the AXI chapter of the Databook. AHB: - 0: OKAY (with FFFF data for non-posted requests) and ignore the setting in bit [2] of this register. - 1: ERROR for normal link (data) accesses and look at bit [2] for other scenarios. AXI: - 0: OKAY (with FFFF data for non-posted requests) - 1: SLVERR/DECERR (the AXI_ERROR_RESPONSE_MAP field determines the PCIe-to-AXI Slave error response mapping) The error response mapping is not applicable to Non-existent Vendor ID register reads. Note: This register field is sticky.

11.3.3.1.120 Link Down AXI Bridge Slave Timeout Register. (AMBA_LINK_TIMEOUT_OFF)

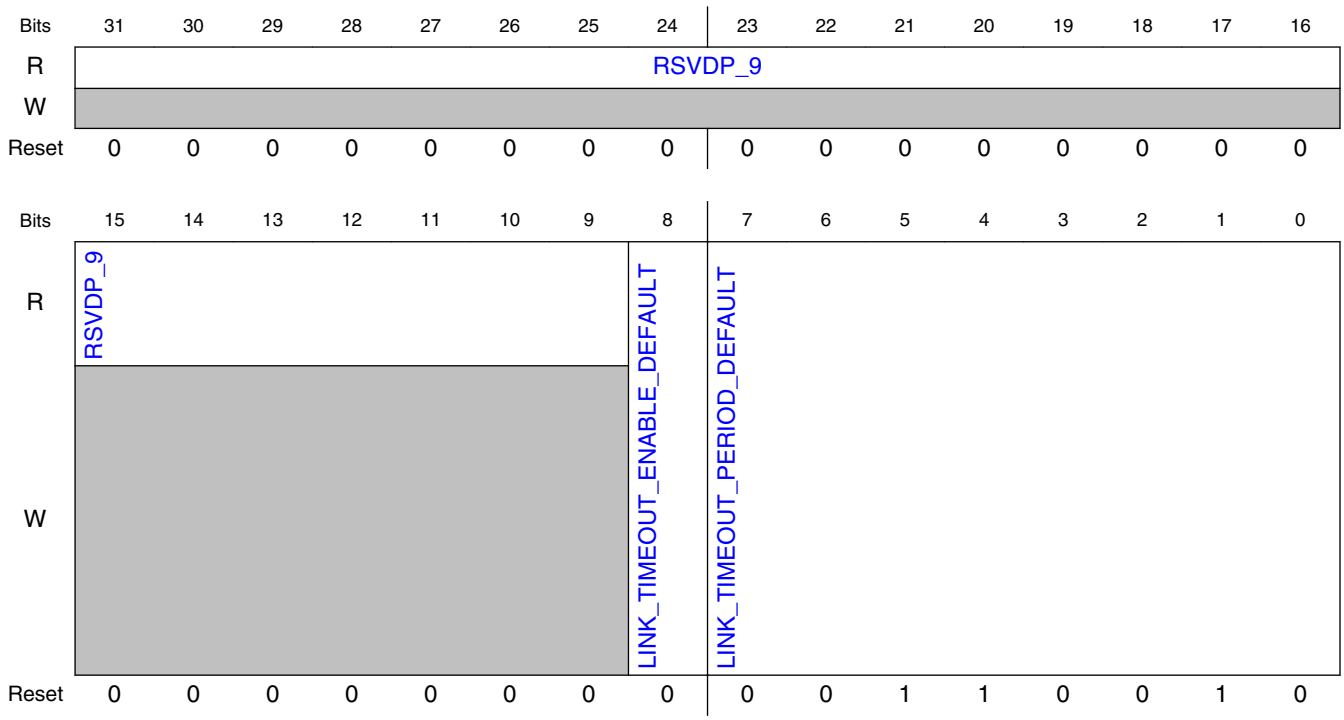
11.3.3.1.120.1 Offset

Register	Offset
AMBA_LINK_TIMEOUT_OFF	8D4h

11.3.3.1.120.2 Function

Link Down AXI Bridge Slave Timeout Register. If your application AXI master issues outbound requests to the AXI bridge slave interface before the PCIe link is operational, the controller starts a "flush" timer. The timeout value of the timer is set by this register. The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface (or DMA) requests and the PCIe TX link is not transmitting any of these requests. For more details, see the "AXI Bridge Initialization, Clocking and Reset" section in the AXI chapter of the Databook.

11.3.3.1.120.3 Diagram



11.3.3.1.120.4 Fields

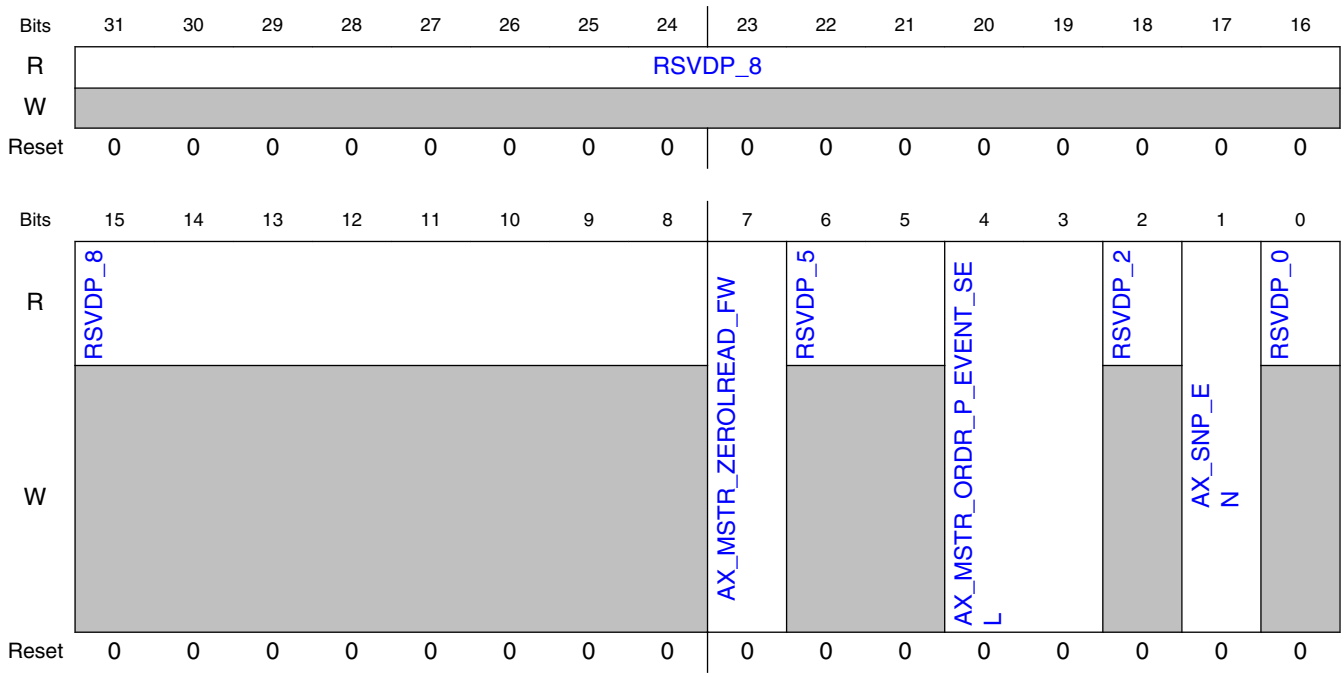
Field	Function
31-9 RSVDP_9	Reserved for future use.
8 LINK_TIMEOUT_ENABLE_DEFAULT	Disable Flush. You can disable the flush feature by setting this field to "1". Note: This register field is sticky.
7-0 LINK_TIMEOUT_PERIOD_DEFAULT	Timeout Value (ms). The timer will timeout and then flush the bridge TX request queues after this amount of time. The timer counts when there are pending outbound AXI slave interface requests and the PCIe TX link is not transmitting any of these requests. The timer is clocked by core_clk. For an M-PCIe configuration: - Time unit of this field is 4 ms. - Margin of error for RateA clock is < 1%. - Margin of error for RateB clock is between 16% and 17%. Note: This register field is sticky.

11.3.3.1.121 AMBA Ordering Control. (AMBA_ORDERING_CTRL_OFF)

11.3.3.1.121.1 Offset

Register	Offset
AMBA_ORDERING_CTRL_OFF	8D8h

11.3.3.1.121.2 Diagram



11.3.3.1.121.3 Fields

Field	Function
31-8 RSVDP_8	Reserved for future use.
7 AX_MSTR_ZEROLREAD_FW	AXI Master Zero Length Read Forward to the application. The DW PCIe controller AXI bridge is able to terminate in order with the Posted transactions the zero length read, implementing the PCIe express flush semantics of the Posted transactions. - 0x0: The zero length Read is terminated at the DW PCIe AXI bridge master - 0x1: The zero length Read is forward to the application.
6-5 RSVDP_5	Reserved for future use.
4-3 AX_MSTR_ORDR_P_EVENT_SEL	AXI Master Posted Ordering Event Selector. This field selects how the master interface determines when a P write is completed when enforcing the PCIe ordering rule, "NP must not pass P" at the AXI Master Interface. The AXI protocol does not support ordering between channels. Therefore, NP reads can pass P on your AXI bus fabric. This can result in an ordering violation when the read overtakes a P that is going to the same address. Therefore, the bridge master does not issue any NP requests until all outstanding P writes reach their destination. It does this by waiting for the all of the write responses on

Table continues on the next page...

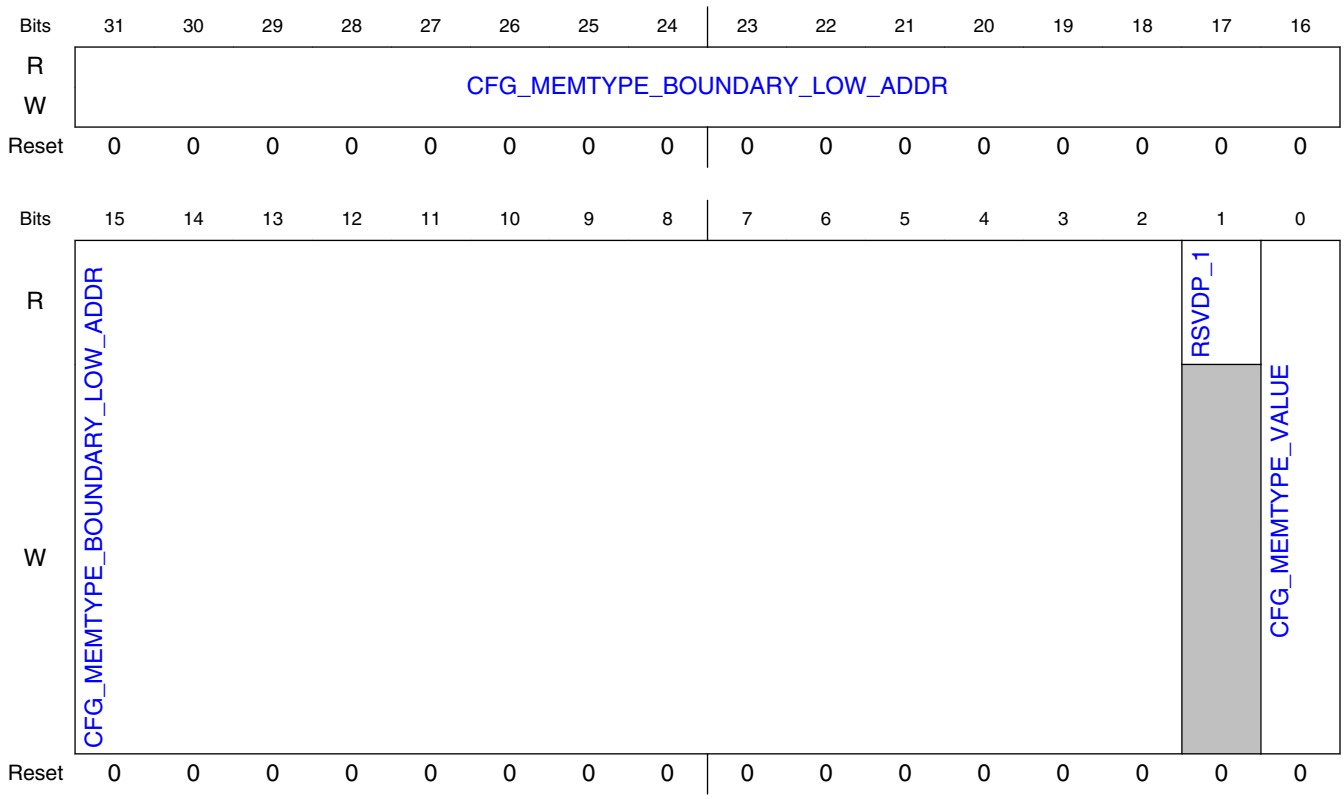
Field	Function
	the B channel. This can affect the performance of the master read channel. For scenarios where the interconnect serializes the AXI master "AW", "W" and "AR" channels,you can increase the performance by reducing the need to wait until the complete Posted transaction has effectively reached the application slave. - 00: B'last event: wait for the all of the write responses on the B channel thereby ensuring that the complete Posted transaction has effectively reached the application slave (default). - 01: AW'last event: wait until the complete Posted transaction has left the AXI address channel at the bridge master. - 10: W'last event: wait until the complete Posted transaction has left the AXI data channel at the bridge master. - 11: Reserved Note 2: This setting will not affect: - MSI interrupt catcher and P data ordering. This is always driven by the B'last event. - DMA read engine TLP ordering. This is always driven by the B'last event. - NP write transactions which are always serialized with P write transactions.
2 RSVDP_2	Reserved for future use.
1 AX_SNP_EN	AXI Serialize Non-Posted Requests Enable. This field enables the AXI Bridge to serialize same ID Non-Posted Read/Write Requests on the wire. Serialization implies one outstanding same ID NP Read or Write on the wire and used to avoid AXI RAR and WAW hazards at the remote link partner. For more details, see the "Optional Serialization of AXI Slave Non-posted Requests" section in the AXI chapter of the Databook.
0 RSVDP_0	Reserved for future use.

11.3.3.1.122 ACE Cache Coherency Control Register 1 (COHERENCY_CONTROL_1_OFF)

11.3.3.1.122.1 Offset

Register	Offset
COHERENCY_CONTROL_1_OFF	8E0h

11.3.3.1.122.2 Diagram



11.3.3.1.122.3 Fields

Field	Function
31-2 CFG_MEMTYPE_BOUNDARY_LOW_ADDR	Boundary Lower Address For Memory Type. Bits [31:0] of dword-aligned address of the boundary for Memory type. The two lower address LSBs are "00". Addresses up to but not including this value are in the lower address space region; addresses equal or greater than this value are in the upper address space region. Note: This register field is sticky.
1 RSVDP_1	Reserved for future use.
0 CFG_MEMTYPE_VALUE	Sets the memory type for the lower and upper parts of the address space: - 0: lower = Peripheral; upper = Memory - 1: lower = Memory type; upper = Peripheral Note: This register field is sticky.

11.3.3.1.123 ACE Cache Coherency Control Register 2 (COHERENCY_CONTROL_2_OFF)

11.3.3.1.123.1 Offset

Register	Offset
COHERENCY_CONTROL_2_OFF	8E4h

11.3.3.1.123.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_MEMTYPE_BOUNDARY_HIGH_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.123.3 Fields

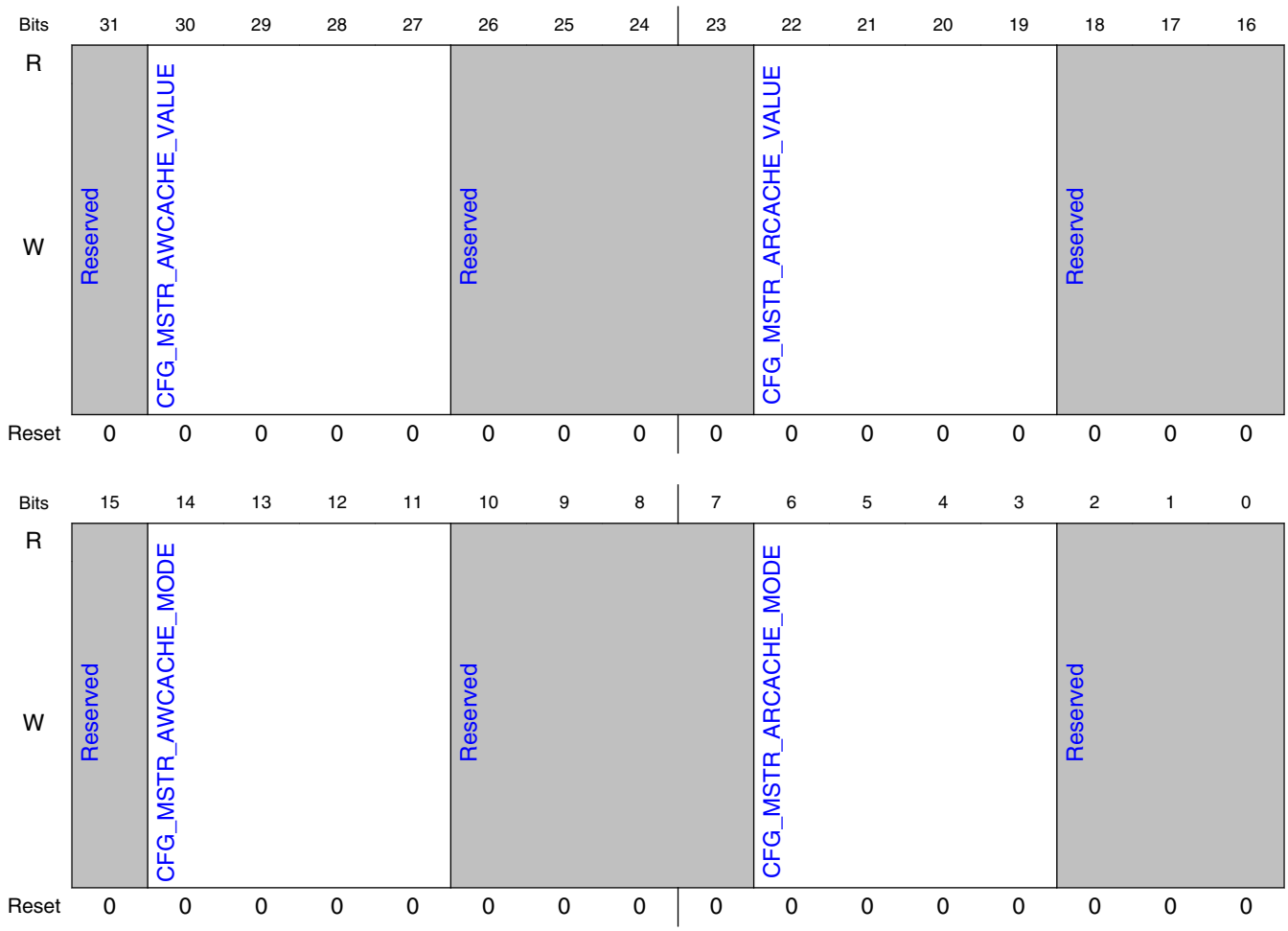
Field	Function
31-0 CFG_MEMTYPE_BOUNDARY_HIGH_ADDR	Boundary Upper Address For Memory Type. Bits [63:32] of the 64-bit dword-aligned address of the boundary for Memory type. Note: This register field is sticky.

11.3.3.1.124 ACE Cache Coherency Control Register 3 (COHERENCY_CONTROL_3_OFF)

11.3.3.1.124.1 Offset

Register	Offset
COHERENCY_CONTROL_3_OFF	8E8h

11.3.3.1.124.2 Diagram



11.3.3.1.124.3 Fields

Field	Function
31 —	Reserved.
30-27 CFG_MSTR_AWCACHE_VALUE	Master Write CACHE Signal Value. Value of the individual bits in mstr_awcache when CFG_MSTR_AWCACHE_MODE is '1'. Note: not applicable to message requests; for message requests the value of mstr_awcache is always "0000" Note: This register field is sticky.
26-23 —	Reserved.
22-19 CFG_MSTR_ARCACHE_VALUE	Master Read CACHE Signal Value. Value of the individual bits in mstr_arcache when CFG_MSTR_ARCACHE_MODE is '1'. Note: This register field is sticky.
18-15	Reserved.

Table continues on the next page...

Field	Function
—	
14-11 CFG_MSTR_AWCACHE_MODE	Master Write CACHE Signal Behavior. Defines how the individual bits in mstr_awcache are controlled: - 0: set automatically by the AXI master - 1: set by the value of the corresponding bit of the CFG_MSTR_AWCACHE_VALUE field Note: for message requests the value of mstr_awcache is always "0000" regardless of the value of this bit Note: This register field is sticky.
10-7 —	Reserved.
6-3 CFG_MSTR_ARCACHE_MODE	Master Read CACHE Signal Behavior. Defines how the individual bits in mstr_arcache are controlled: - 0: set automatically by the AXI master - 1: set by the value of the corresponding bit of the CFG_MSTR_ARCACHE_VALUE field Note: This register field is sticky.
2-0 —	Reserved.

11.3.3.1.125 Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_LOW_OFF)

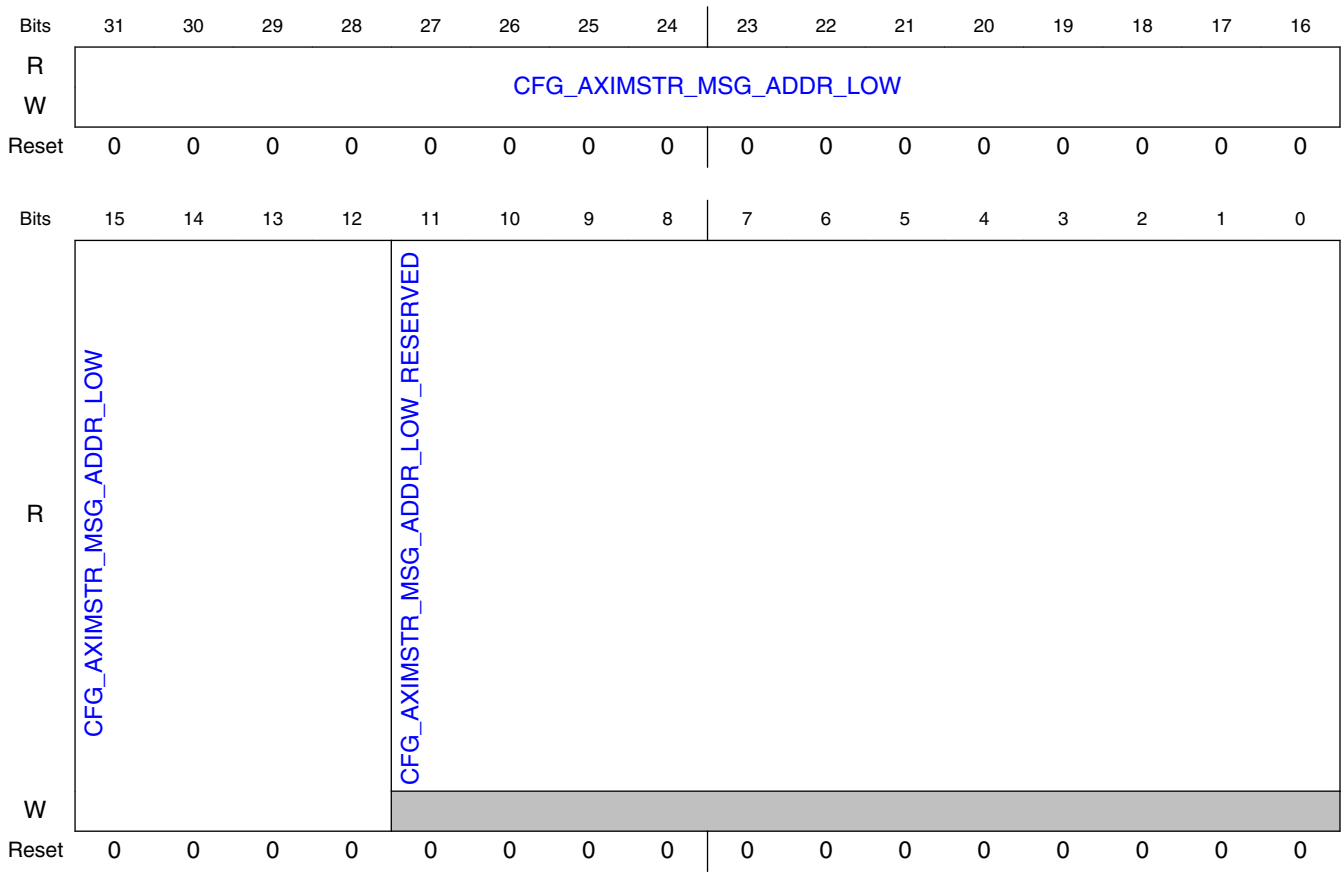
11.3.3.1.125.1 Offset

Register	Offset
AXI_MSTR_MSG_ADDR_LOW_OFF	8F0h

11.3.3.1.125.2 Function

Lower 20 bits of the programmable AXI address where Messages coming from wire are mapped to. Bits [11:0] of the register are tied to zero for the address to be 4k-aligned. In previous releases, the third and fourth DWORDs of a message (Msg/MsgD) TLP header were delivered though the AXI master address bus (mstr_awaddr). These DWORDS are now supplied through the mstr_awmisc_info_hdr_34dw[63:0] output; and the value on mstr_awaddr is driven to the value you have programmed into the AXI_MSTR_MSG_ADDR_LOW_OFF and AXI_MSTR_MSG_ADDR_HIGH_OFF registers.

11.3.3.1.125.3 Diagram



11.3.3.1.125.4 Fields

Field	Function
31-12 CFG_AXIMSTR_MSG_ADDR_LOW	Lower 20 bits of the programmable AXI address for Messages. Note: This register field is sticky.
11-0 CFG_AXIMSTR_MSG_ADDR_LOW_RESERVED	Reserved for future use. Note: This register field is sticky.

11.3.3.1.126 Upper 32 bits of the programmable AXI address where Messages coming from wire are mapped to. (AXI_MSTR_MSG_ADDR_HIGH_OFF)

11.3.3.1.126.1 Offset

Register	Offset
AXI_MSTR_MSG_ADDR_HIGH_OFF	8F4h

11.3.3.1.126.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CFG_AXIMSTR_MSG_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CFG_AXIMSTR_MSG_ADDR_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.126.3 Fields

Field	Function
31-0 CFG_AXIMSTR_MSG_ADDR_HIGH	Upper 32 bits of the programmable AXI address for Messages. Note: This register field is sticky.

11.3.3.1.127 PCIe Controller IIP Release Version Number. (PCIE_VERSION_NUMBER_OFF)

11.3.3.1.127.1 Offset

Register	Offset
PCIE_VERSION_NUMBER_OFF	8F8h

11.3.3.1.127.2 Function

PCIe Controller IIP Release Version Number. The version number is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA)

as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release

11.3.3.1.127.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	VERSION_NUMBER															
W																
Reset	0	0	1	1	0	1	0	1	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	VERSION_NUMBER															
W																
Reset	0	0	1	1	0	0	0	0	0	0	1	0	1	0	1	0

11.3.3.1.127.4 Fields

Field	Function
31-0 VERSION_NUMBER	Version Number.

11.3.3.1.128 PCIe Controller IIP Release Version Type. (PCIE_VERSION_TYPE_OFF)

11.3.3.1.128.1 Offset

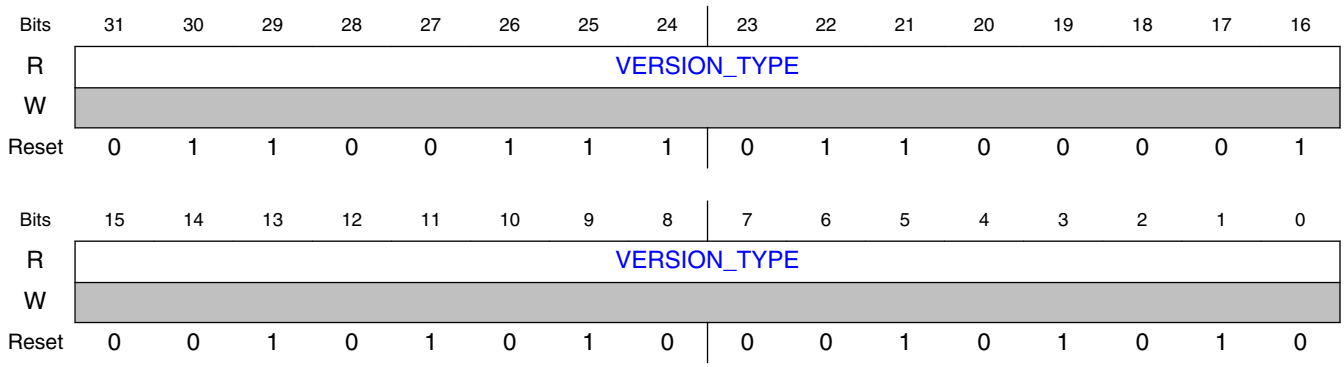
Register	Offset
PCIE_VERSION_TYPE_OFF	8FCh

11.3.3.1.128.2 Function

PCIe Controller IIP Release Version Type. The type is given in hex format. You should convert each pair of hex characters to ASCII to interpret. Using 4.70a (GA) as an example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x67612a2a which translates to ga** Using 4.70a-ea01 as an

example: - VERSION_NUMBER = 0x3437302a which translates to 470* - VERSION_TYPE = 0x65613031 which translates to ea01 GA is a general release.

11.3.3.1.128.3 Diagram



11.3.3.1.128.4 Fields

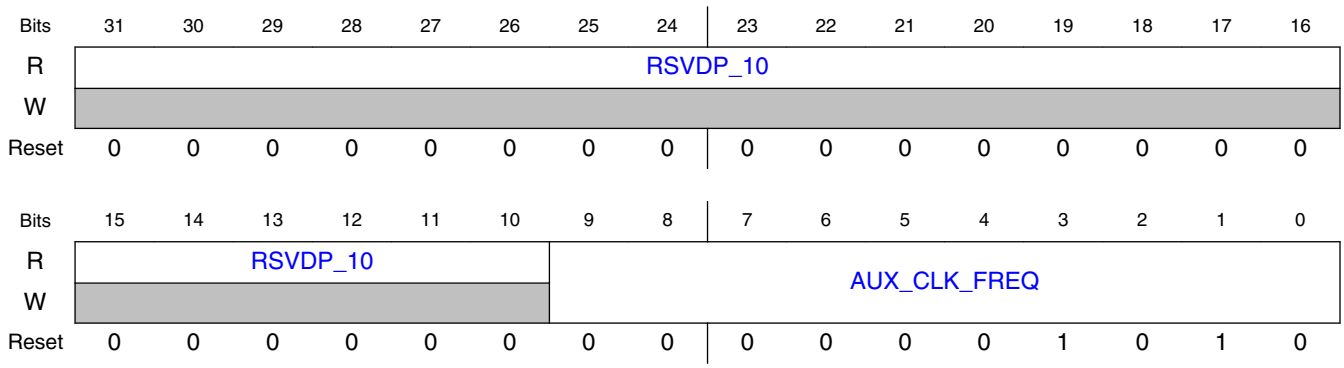
Field	Function
31-0 VERSION_TYPE	Version Type.

11.3.3.1.129 Auxiliary Clock Frequency Control Register. (AUX_CLK_FREQ_OFF)

11.3.3.1.129.1 Offset

Register	Offset
AUX_CLK_FREQ_OFF	B40h

11.3.3.1.129.2 Diagram



11.3.3.1.129.3 Fields

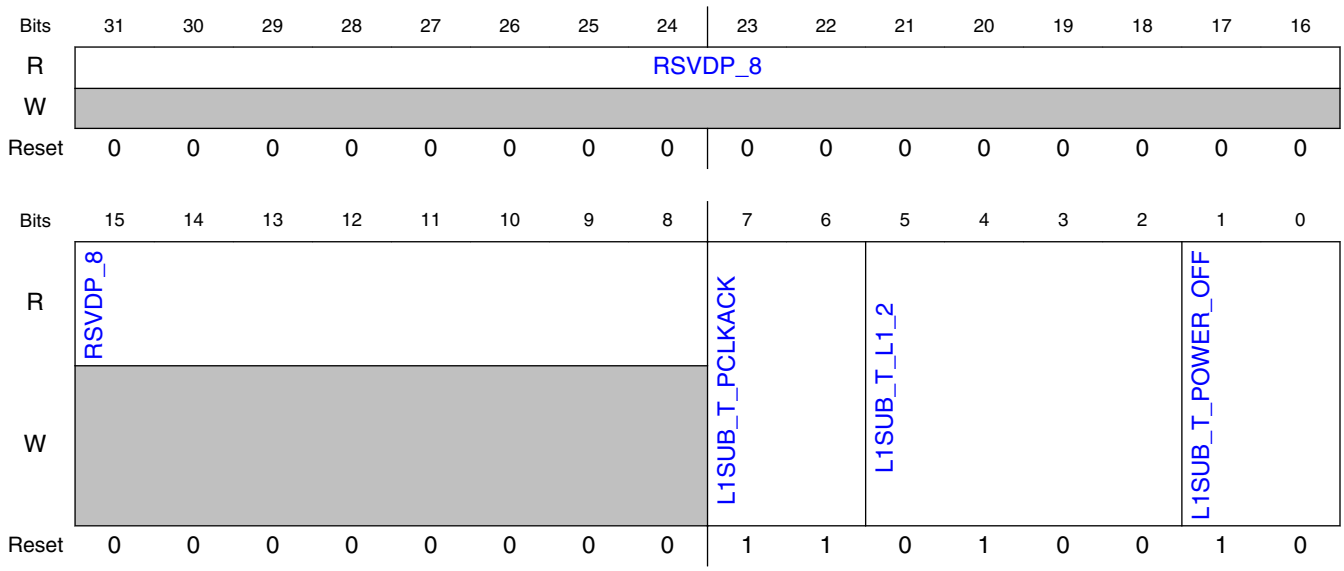
Field	Function
31-10 RSVDP_10	Reserved for future use.
9-0 AUX_CLK_FREQ	The aux_clk frequency in MHz. This value is used to provide a 1 us reference for counting time during low-power states with aux_clk when the PHY has removed the pipe_clk. Frequencies lower than 1 MHz are possible but with a loss of accuracy in the time counted. If the actual frequency (f) of aux_clk does not exactly match the programmed frequency (f_prog), then there is an error in the time counted by the controller that can be expressed in percentage as: err% = (f_prog/f-1)*100. For example if f=2.5 MHz and f_prog=3 MHz, then err% =(3/2.5-1)*100 =20%, meaning that the time counted by the controller on aux_clk will be 20% greater than the time in us programmed in the corresponding time register (for example T_POWER_ON). Note: This register field is sticky.

11.3.3.1.130 L1 Substates Timing Register. (L1_SUBSTATES_OFF)

11.3.3.1.130.1 Offset

Register	Offset
L1_SUBSTATES_OFF	B44h

11.3.3.1.130.2 Diagram



11.3.3.1.130.3 Fields

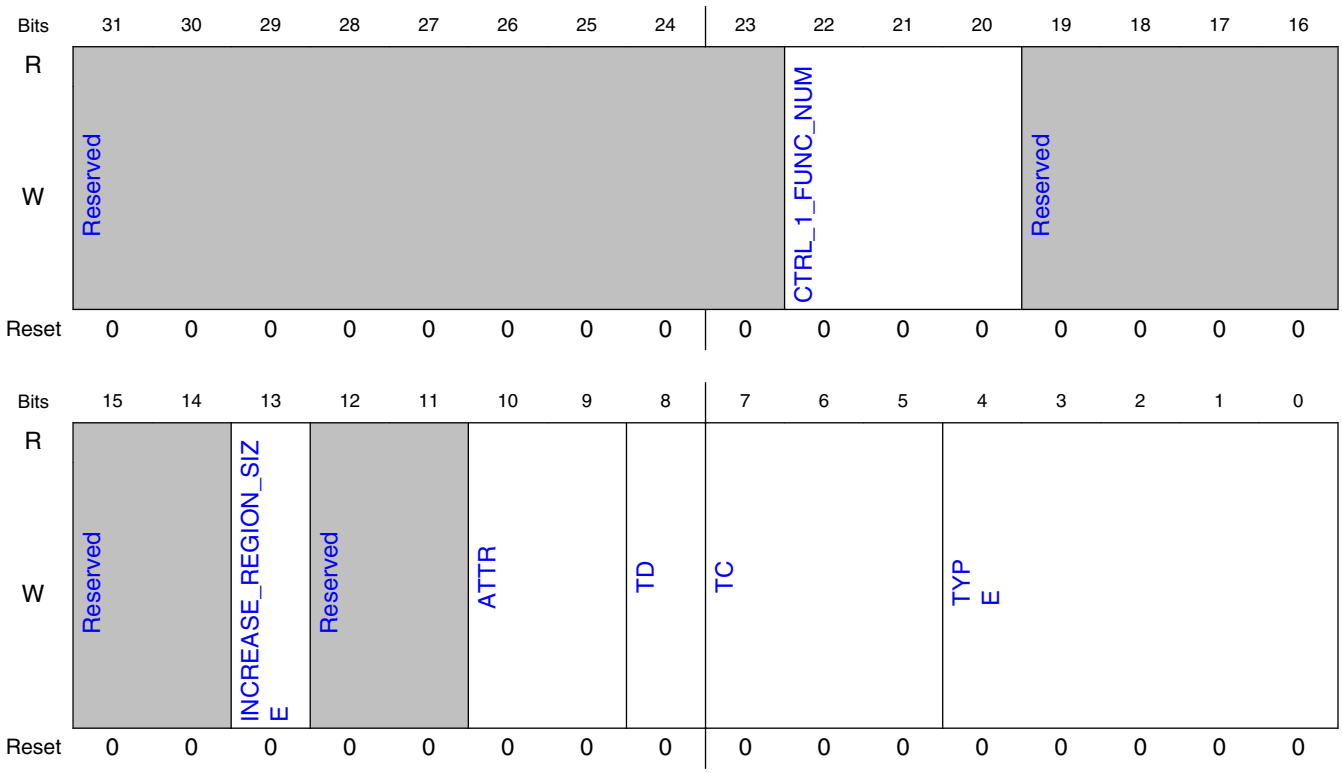
Field	Function
31-8 RSVDP_8	Reserved for future use.
7-6 L1SUB_T_PCLKACK	Max delay (in 1us units) between a MAC request to remove the clock on mac_phy_pclkreq_n and a PHY response on phy_mac_pclkack_n. If the PHY does not respond within this time the request is aborted. Range is 0..3 Note: This register field is sticky.
5-2 L1SUB_T_L1_2	Duration (in 1us units) of L1.2. Range is 0.15. Note: The timeout value can vary by 50%. Note: This register field is sticky.
1-0 L1SUB_T_POWER_OFF	Duration (in 1us units) of L1.2.Entry. Range is 0.3. Note: The timeout value can vary by 50%. Note: This register field is sticky.

11.3.3.1.131 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_0)

11.3.3.1.131.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_0	8000_0000h

11.3.3.1.131.2 Diagram



11.3.3.1.131.3 Fields

Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.

Table continues on the next page...

Field	Function
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.132 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_0)

11.3.3.1.132.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_0	8000_0004h

11.3.3.1.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	REGION_EN	Reserved	INVERT_MODE	CFG_SHIFT_MODE	DMA_BYPASS	Reserved			HEADER_SUBSTITUTE_EN	INHIBIT_PAYLOAD	Reserved	SNP	FUNC_BYPASS	Reserved		TAG_SUBSTITUTE_EN
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TAG								MSG_CODE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.132.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved.
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved.
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved.
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.

Table continues on the next page...

Field	Function
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.133 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0)

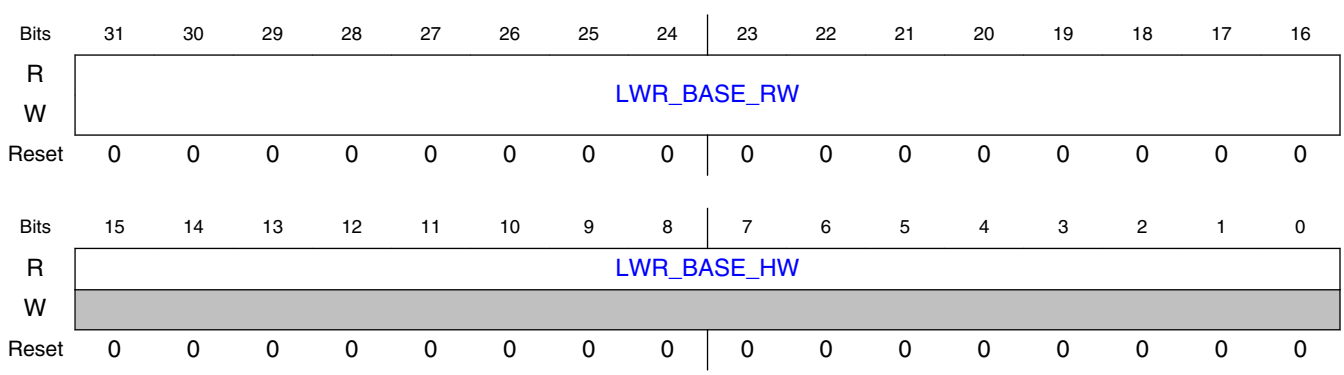
11.3.3.1.133.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_0	8000_0008h

11.3.3.1.133.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.133.3 Diagram



11.3.3.1.133.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.134 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0)

11.3.3.1.134.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_0	8000_000Ch

11.3.3.1.134.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.134.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.3.3.1.135 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_0)

11.3.3.1.135.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_0	8000_0010h

11.3.3.1.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.135.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.136 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0)

11.3.3.1.136.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_0	8000_0014h

11.3.3.1.136.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.136.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to the Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.3.3.1.137 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0)

11.3.3.1.137.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_0	8000_0018h

11.3.3.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.137.3 Fields

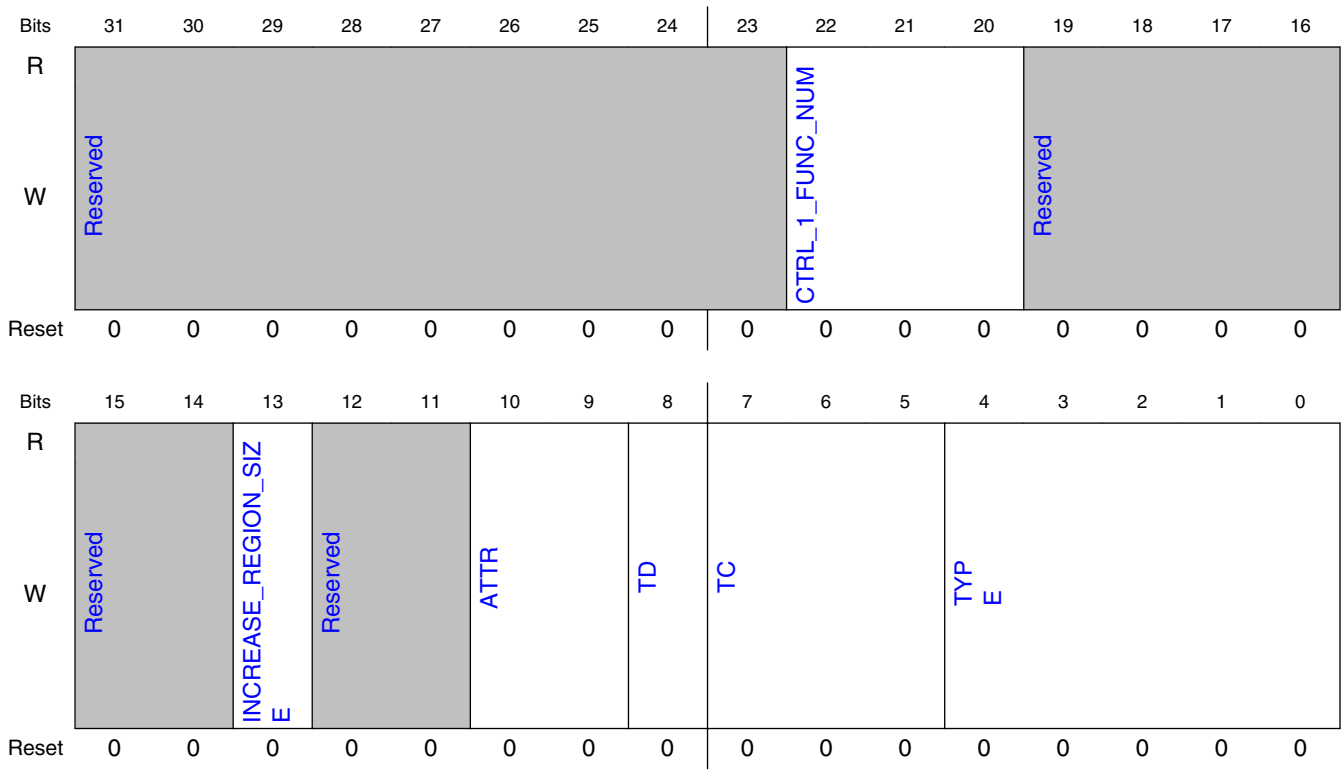
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.3.3.1.138 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_0)

11.3.3.1.138.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_0	8000_0100h

11.3.3.1.138.2 Diagram



11.3.3.1.138.3 Fields

Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.

Table continues on the next page...

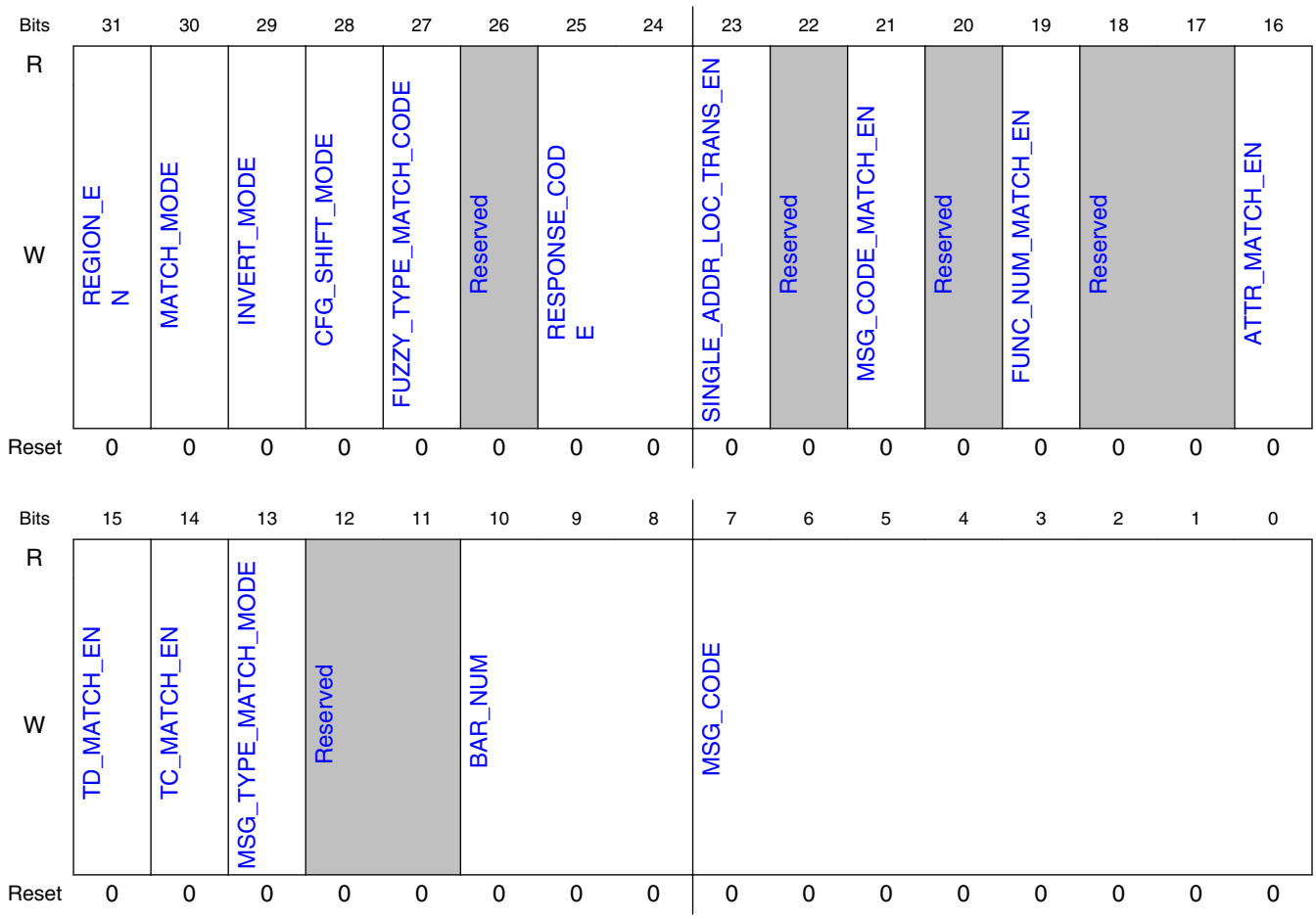
Field	Function
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.3.3.1.139 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_0)

11.3.3.1.139.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_0	8000_0104h

11.3.3.1.139.2 Diagram



11.3.3.1.139.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches

Table continues on the next page...

Field	Function
	against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved.
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved.
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Note: This register field is sticky.
20 —	Reserved.
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved.
16 ATTR_MATCH_EN	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved.
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Note: This register field is sticky.

11.3.3.1.140 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_0)

11.3.3.1.140.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_0	8000_0108h

11.3.3.1.140.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.140.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.140.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.141 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_0)

11.3.3.1.141.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_0	8000_010Ch

11.3.3.1.141.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.141.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.3.3.1.142 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_0)

11.3.3.1.142.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_0	8000_0110h

11.3.3.1.142.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.142.3 Fields

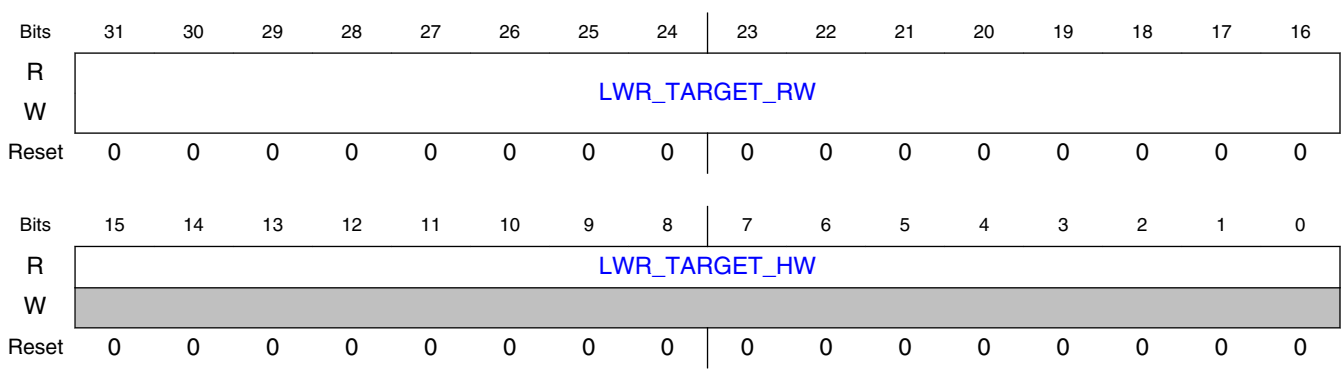
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.143 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_0)

11.3.3.1.143.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_0	8000_0114h

11.3.3.1.143.2 Diagram



11.3.3.1.143.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

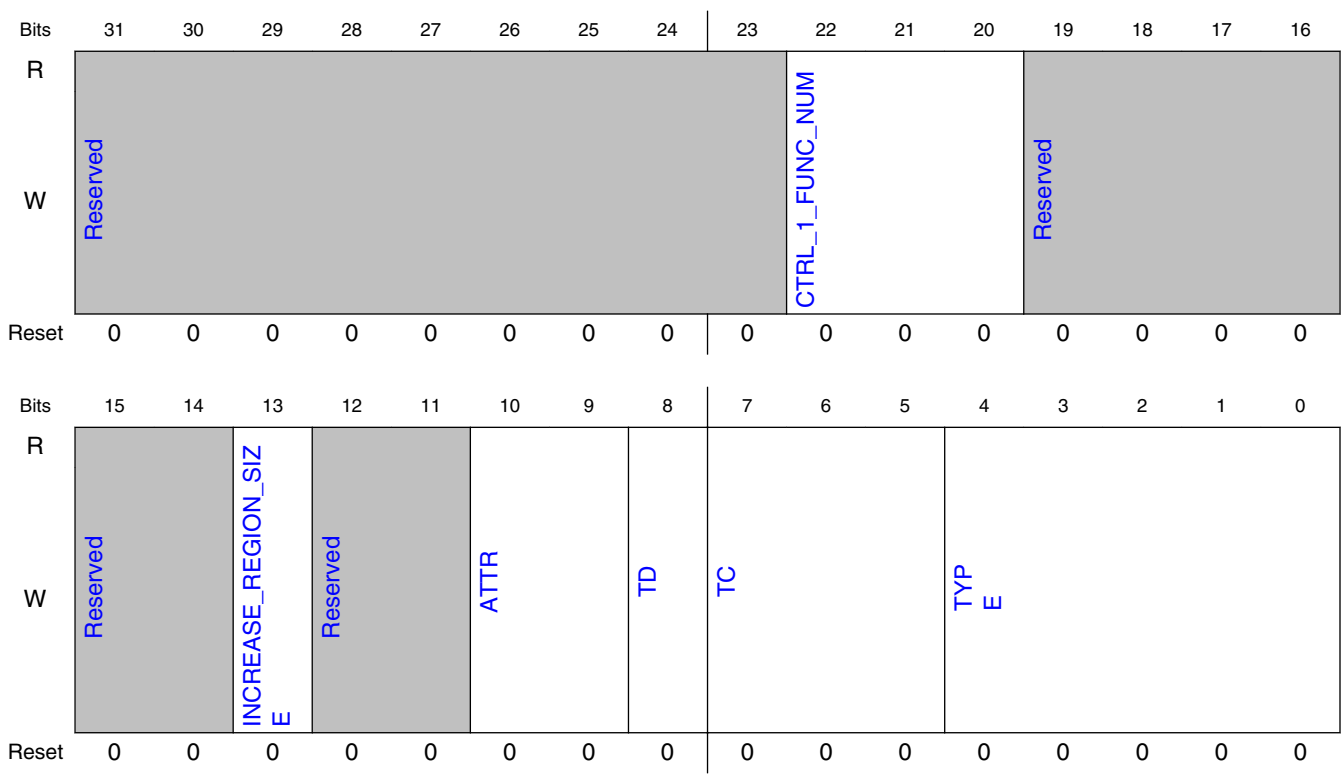
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to the Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK +1) in BAR match mode.

11.3.3.1.144 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_1)

11.3.3.1.144.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_1	8000_0200h

11.3.3.1.144.2 Diagram



11.3.3.1.144.3 Fields

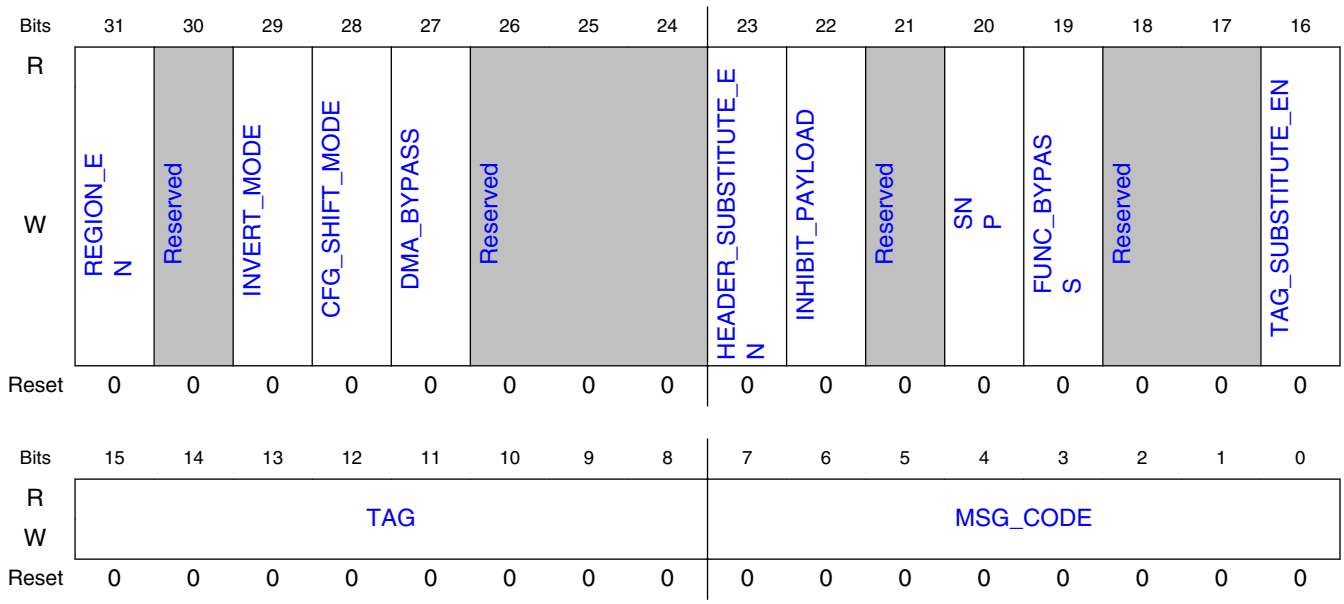
Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.145 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_1)

11.3.3.1.145.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_1	8000_0204h

11.3.3.1.145.2 Diagram



11.3.3.1.145.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved.
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.

Table continues on the next page...

Field	Function
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved.
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved.
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.146 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1)

11.3.3.1.146.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_1	8000_0208h

11.3.3.1.146.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.146.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.146.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.147 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1)

11.3.3.1.147.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_1	8000_020Ch

11.3.3.1.147.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.147.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.3.3.1.148 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_1)

11.3.3.1.148.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_1	8000_0210h

11.3.3.1.148.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.148.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.149 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1)

11.3.3.1.149.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_1	8000_0214h

11.3.3.1.149.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.149.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to the Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.3.3.1.150 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1)

11.3.3.1.150.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_1	8000_0218h

11.3.3.1.150.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.150.3 Fields

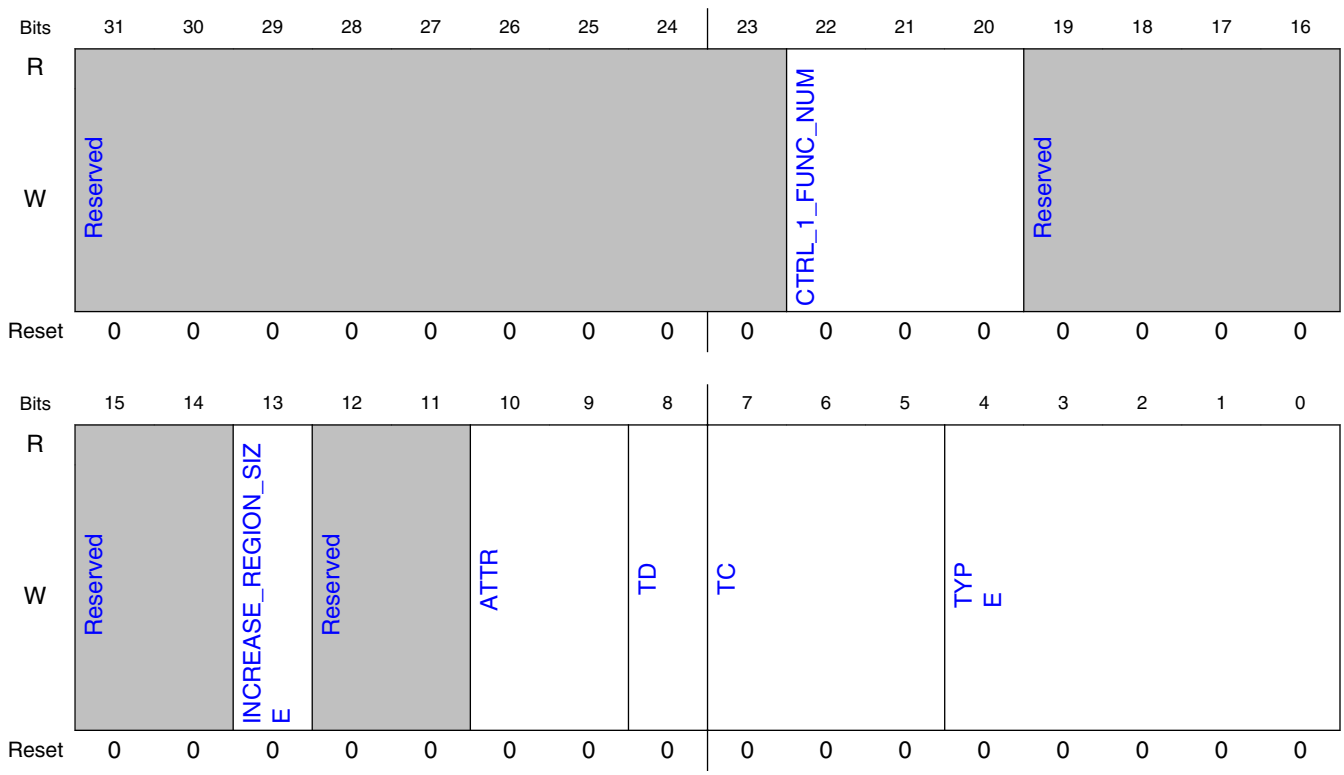
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.3.3.1.151 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_1)

11.3.3.1.151.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_1	8000_0300h

11.3.3.1.151.2 Diagram



11.3.3.1.151.3 Fields

Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the

Table continues on the next page...

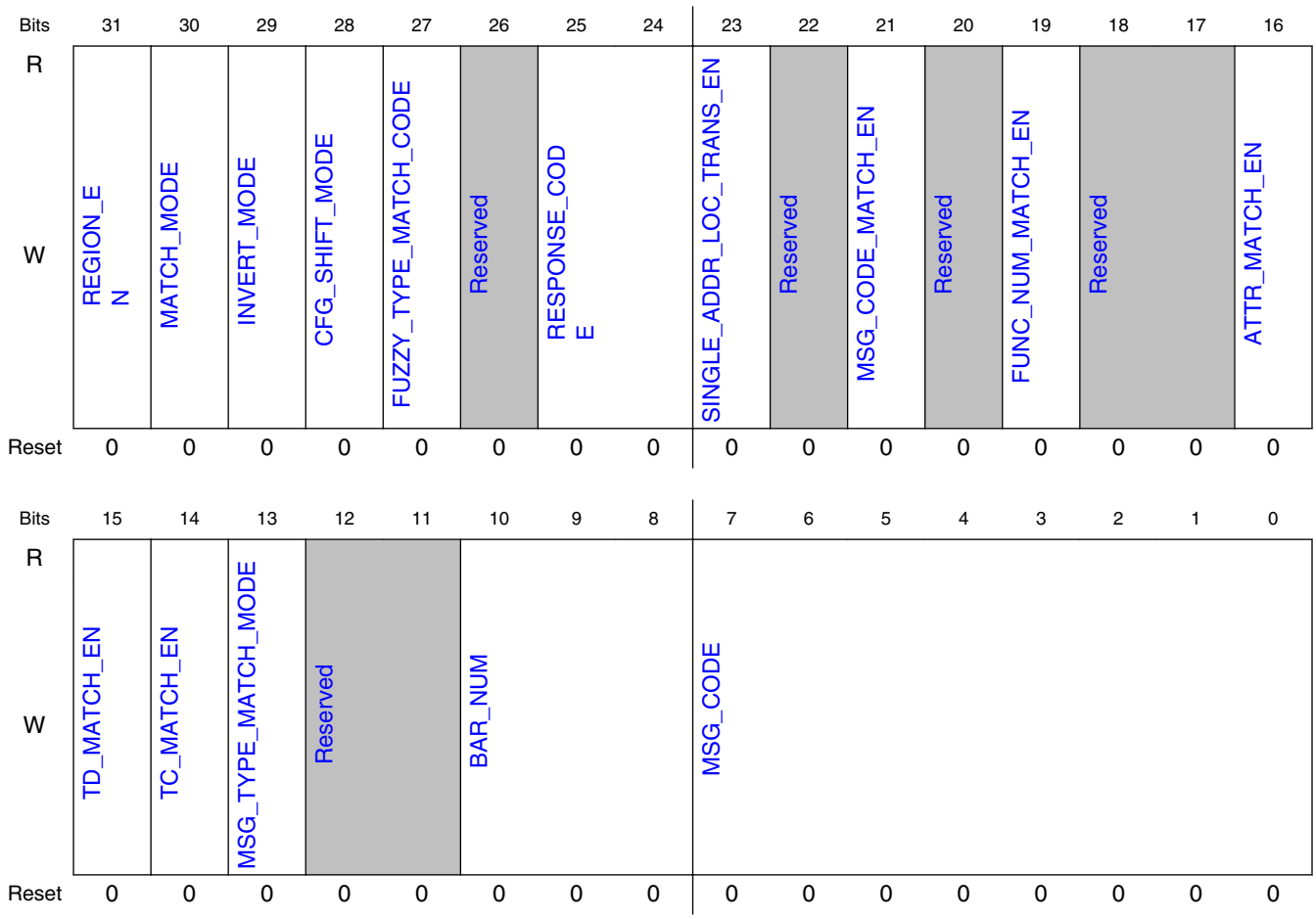
Field	Function
	"Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.3.3.1.152 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_1)

11.3.3.1.152.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_1	8000_0304h

11.3.3.1.152.2 Diagram



11.3.3.1.152.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches

Table continues on the next page...

Field	Function
	against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved.
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved.
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Note: This register field is sticky.
20 —	Reserved.
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved.
16 ATTR_MATCH_EN	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved.
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Note: This register field is sticky.

11.3.3.1.153 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_1)

11.3.3.1.153.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_1	8000_0308h

11.3.3.1.153.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.153.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.153.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.154 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_1)

11.3.3.1.154.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_1	8000_030Ch

11.3.3.1.154.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.154.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.3.3.1.155 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_1)

11.3.3.1.155.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_1	8000_0310h

11.3.3.1.155.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.155.3 Fields

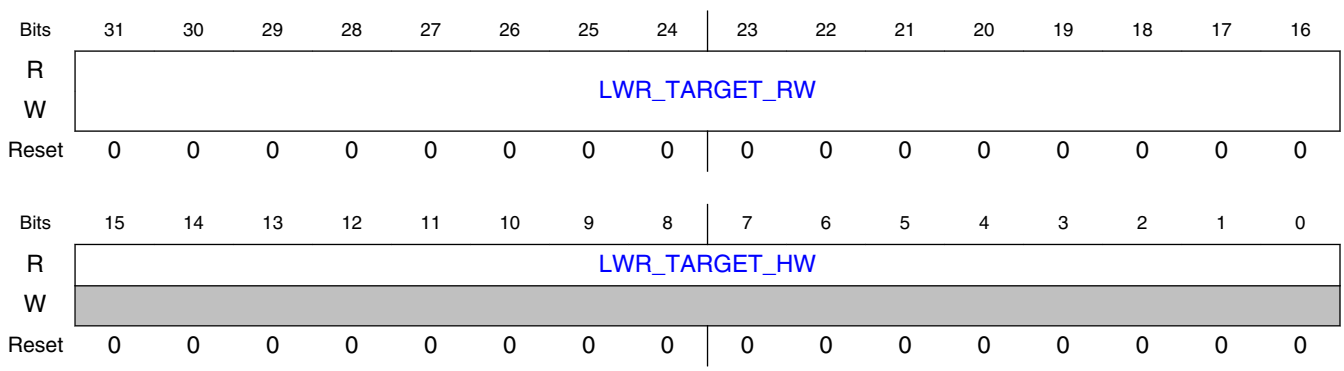
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.156 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_1)

11.3.3.1.156.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_1	8000_0314h

11.3.3.1.156.2 Diagram



11.3.3.1.156.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

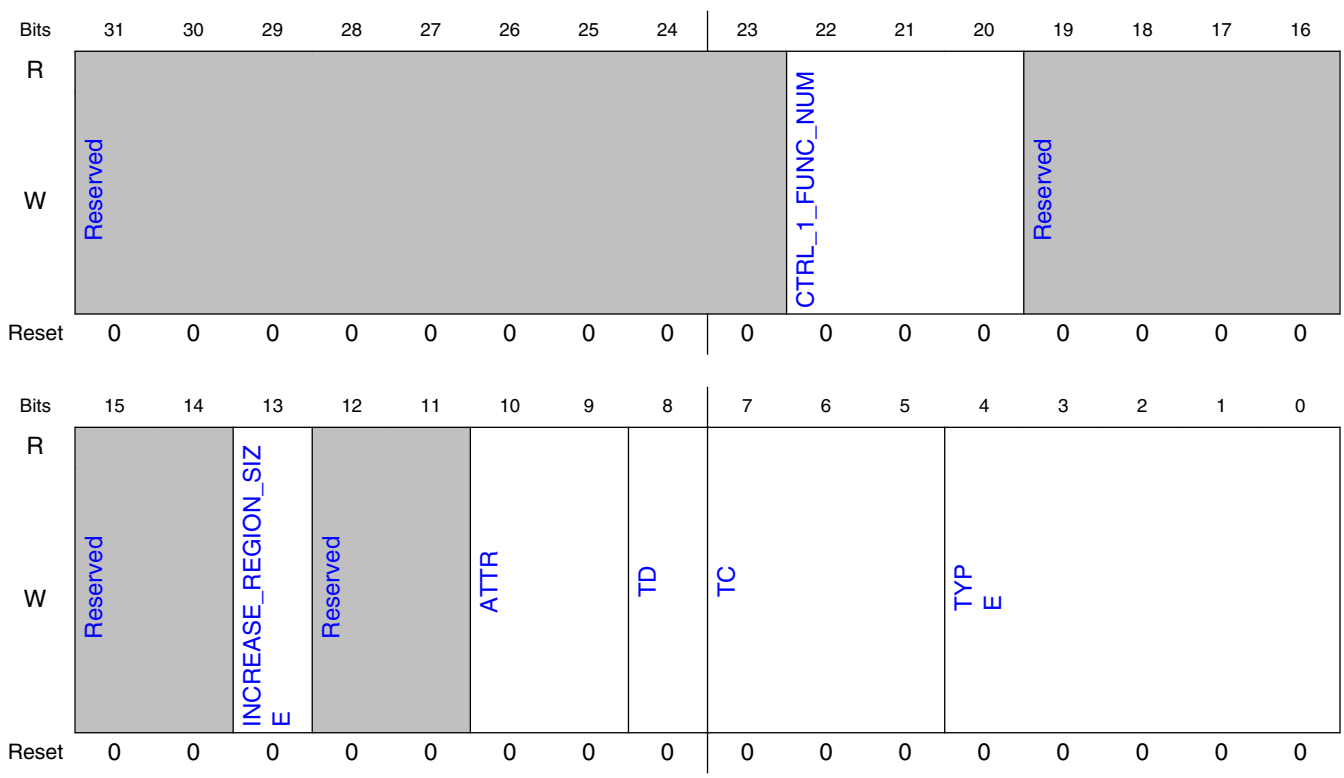
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to the Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK +1) in BAR match mode.

11.3.3.1.157 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_2)

11.3.3.1.157.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_2	8000_0400h

11.3.3.1.157.2 Diagram



11.3.3.1.157.3 Fields

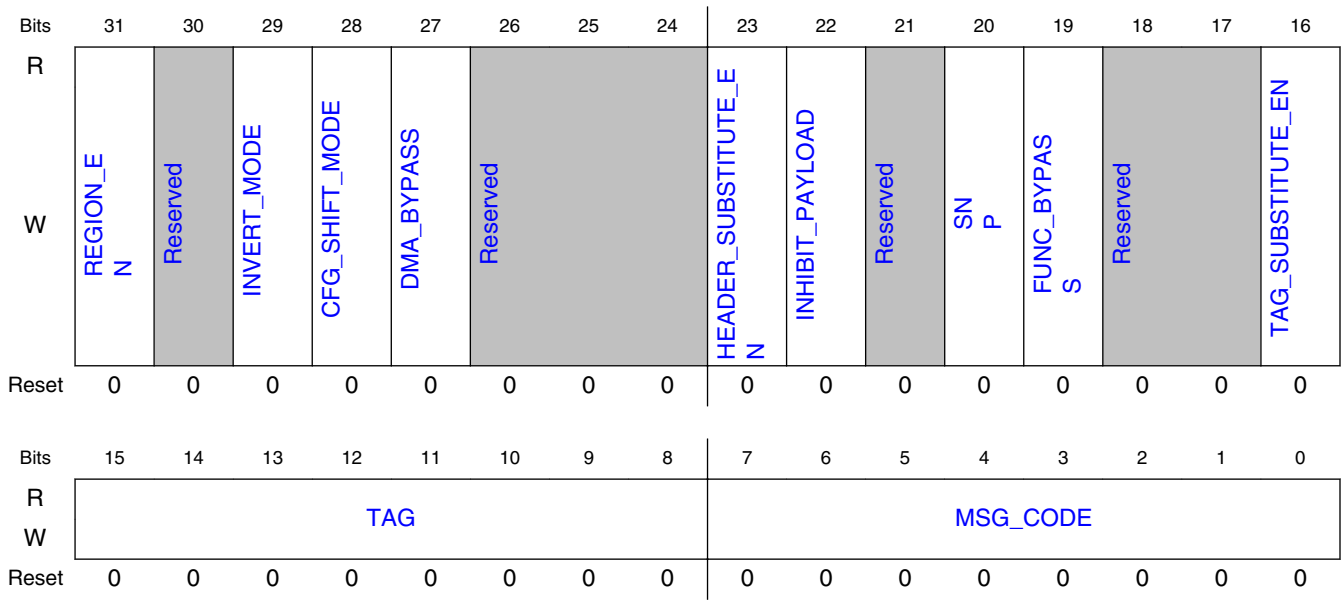
Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.158 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_2)

11.3.3.1.158.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_2	8000_0404h

11.3.3.1.158.2 Diagram



11.3.3.1.158.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved.
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.

Table continues on the next page...

Field	Function
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved.
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved.
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.159 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2)

11.3.3.1.159.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_2	8000_0408h

11.3.3.1.159.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.159.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	LWR_BASE_RW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.159.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.160 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2)

11.3.3.1.160.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_2	8000_040Ch

11.3.3.1.160.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.160.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.3.3.1.161 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_2)

11.3.3.1.161.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_2	8000_0410h

11.3.3.1.161.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.161.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.162 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2)

11.3.3.1.162.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_2	8000_0414h

11.3.3.1.162.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.162.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to the Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.3.3.1.163 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2)

11.3.3.1.163.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_2	8000_0418h

11.3.3.1.163.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.163.3 Fields

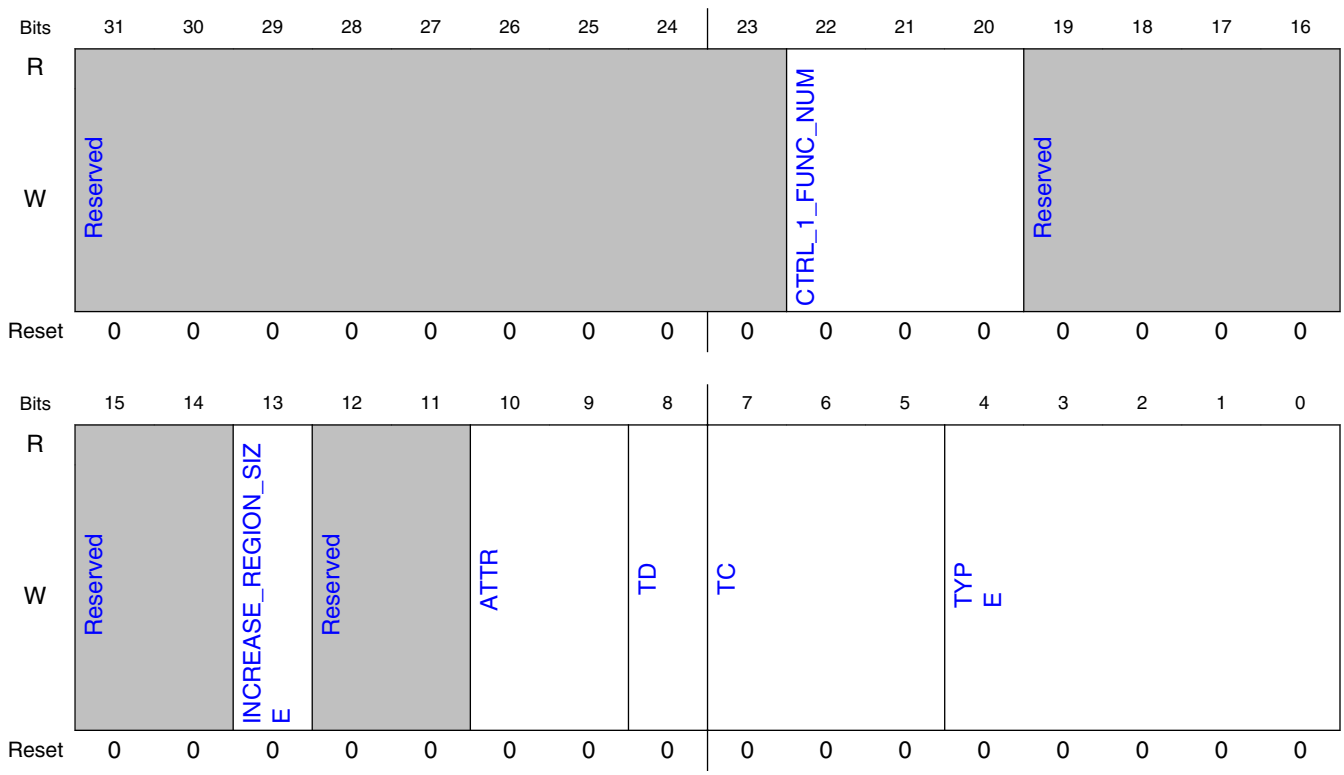
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.3.3.1.164 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_2)

11.3.3.1.164.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_2	8000_0500h

11.3.3.1.164.2 Diagram



11.3.3.1.164.3 Fields

Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the

Table continues on the next page...

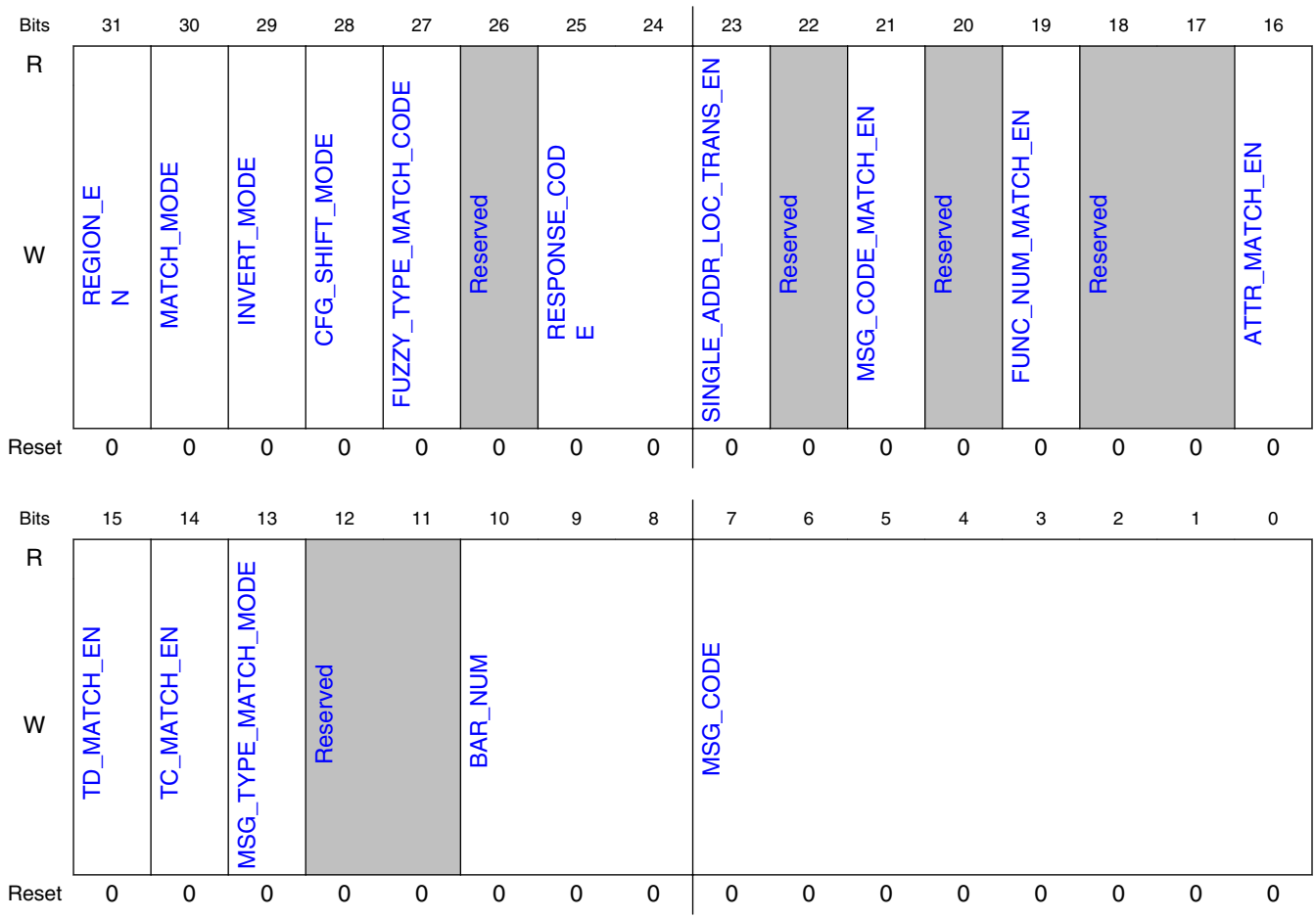
Field	Function
	"Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.3.3.1.165 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_2)

11.3.3.1.165.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_2	8000_0504h

11.3.3.1.165.2 Diagram



11.3.3.1.165.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches

Table continues on the next page...

Field	Function
	against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved.
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved.
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Note: This register field is sticky.
20 —	Reserved.
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved.
16 ATTR_MATCH_EN	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved.
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Note: This register field is sticky.

11.3.3.1.166 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_2)

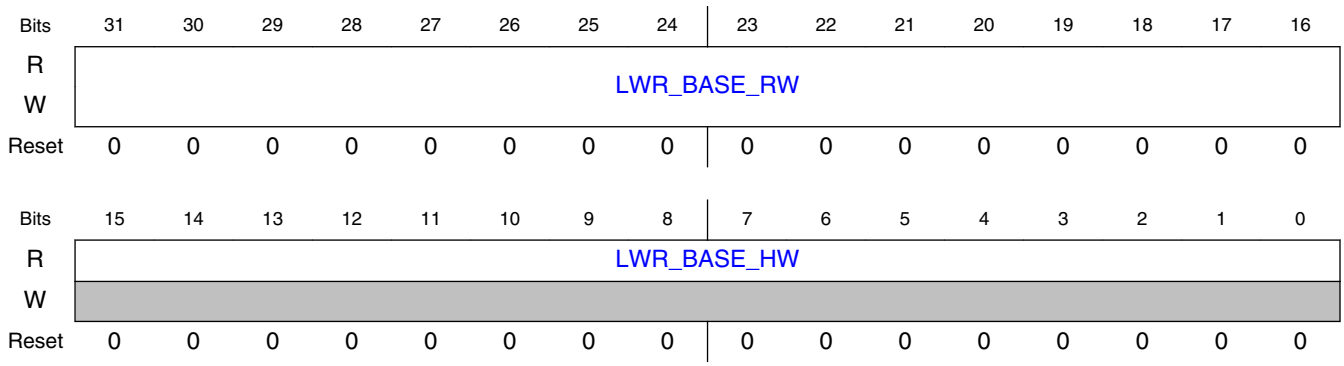
11.3.3.1.166.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_2	8000_0508h

11.3.3.1.166.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.166.3 Diagram



11.3.3.1.166.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.167 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_2)

11.3.3.1.167.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_2	8000_050Ch

11.3.3.1.167.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.167.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.3.3.1.168 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_2)

11.3.3.1.168.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_2	8000_0510h

11.3.3.1.168.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.168.3 Fields

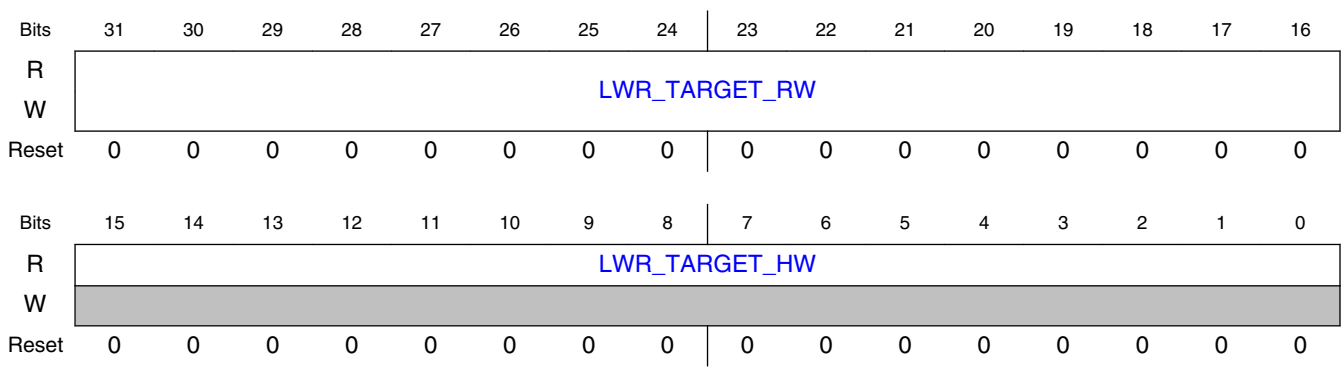
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.169 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_2)

11.3.3.1.169.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_2	8000_0514h

11.3.3.1.169.2 Diagram



11.3.3.1.169.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

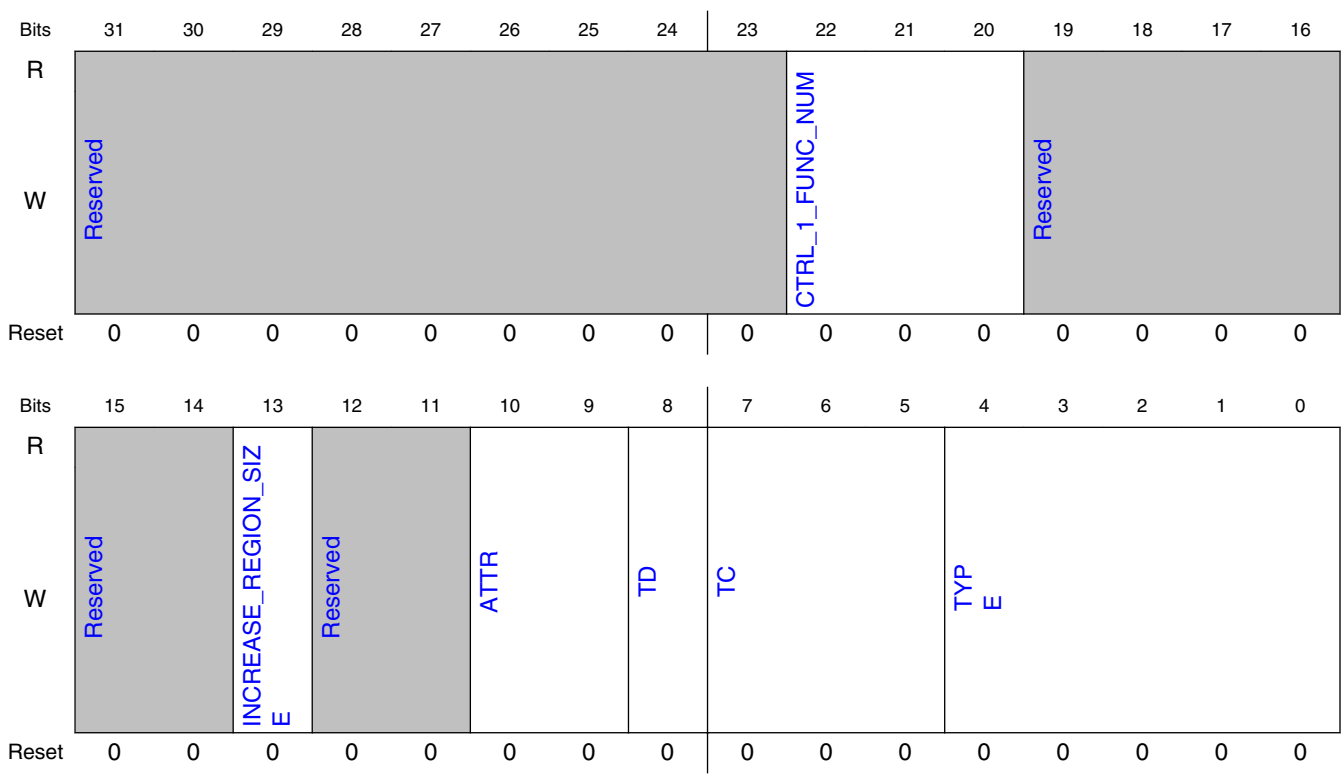
Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to the Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK +1) in BAR match mode.

11.3.3.1.170 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_OUTBOUND_3)

11.3.3.1.170.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_OUTBOUND_3	8000_0600h

11.3.3.1.170.2 Diagram



11.3.3.1.170.3 Fields

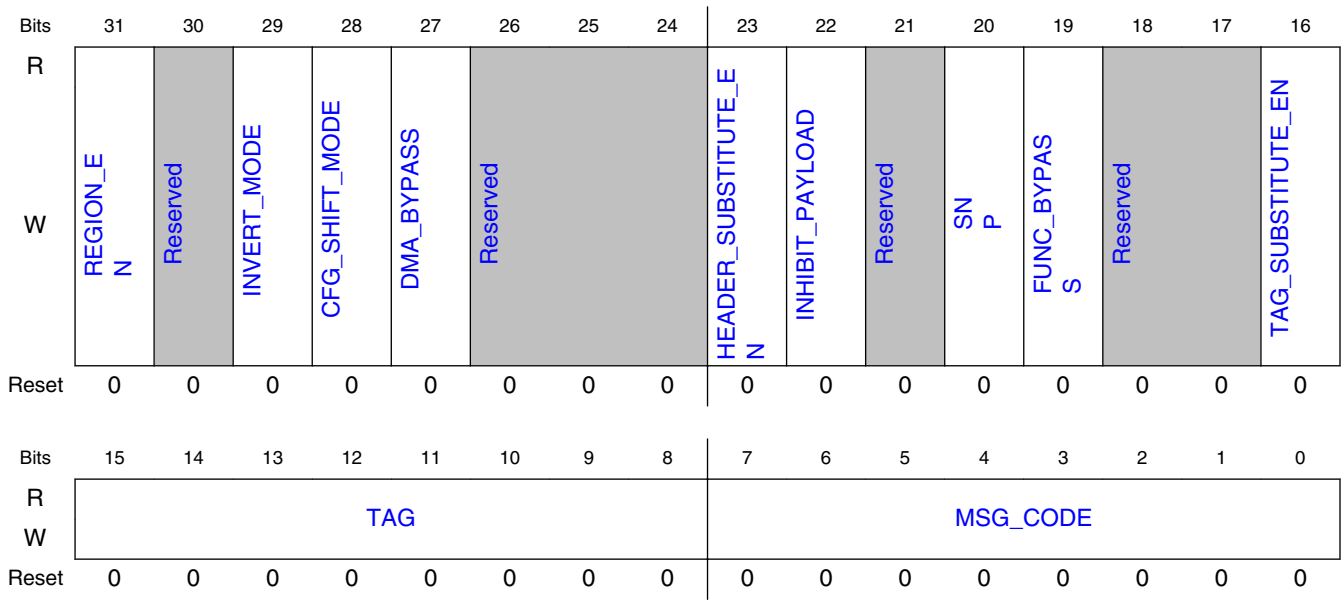
Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - When the address of an outbound TLP is matched to this region and the FUNC_BYPASS field in the "iATU Control 2 Register" is '0', then the function number used in generating the function part of the requester ID (RID) field of the TLP is taken from this 5-bit register. When you are using the AXI Bridge, then this field is swapped before AXI decomposition occurs so that the correct "Max_Read_Request_Size" and "Max_Payload_Size" values are used. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the address of an outbound TLP is matched to this region, then the ATTR field of the TLP is changed to the value in this register. Note: This register field is sticky.
8 TD	When the address of an outbound TLP is matched to this region, then the TD field of the TLP is changed to the value in this register. Note: This register field is sticky.
7-5 TC	When the address of an outbound TLP is matched to this region, then the TC field of the TLP is changed to the value in this register. Note: This register field is sticky.
4-0 TYPE	When the address of an outbound TLP is matched to this region, then the TYPE field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.171 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_OUTBOUND_3)

11.3.3.1.171.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_OUTBOUND_3	8000_0604h

11.3.3.1.171.2 Diagram



11.3.3.1.171.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 —	Reserved.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. The iATU uses bits [27:12] of the untranslated address (on the XALI0/1/2 interface or AXI slave interface address) to form the BDF number of the outgoing CFG TLP. This supports the Enhanced Configuration Address Mapping (ECAM) mechanism (Section 7.2.2 of the PCI Express Base 3.1 Specification, revision 1.0) by allowing all outgoing I/O and MEM TLPs (that have been translated to CFG) to be mapped from memory space into any 256 MB region of the PCIe configuration space. Note: This register field is sticky.
27 DMA_BYPASS	DMA Bypass Mode. Allows request TLPs which are initiated by the DMA controller to pass through the iATU untranslated. Note: This field is reserved for the SW product. You must set it to '0'. Note: This register field is sticky.
26-24 —	Reserved.
23 HEADER_SUBSTITUTE_EN	Header Substitute Enable. When enabled and region address is matched, the iATU fully substitutes bytes 8-11 (for 3 DWORD header) or bytes 12-15 (for 4 DWORD header) of the outbound TLP header with the contents of the LWR_TARGET_RW field in IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i. - 1: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register is used to fill bytes 8-to-11 (for 3 DWORD header) or bytes 12-to-15 (for 4 DWORD header) of the translated TLP header. - 0: LWR_TARGET_RW in the iATU_LWR_TARGET_ADDR_OFF_OUTBOUND_i register forms the new address of the translated region. Note: This register field is sticky.

Table continues on the next page...

Field	Function
22 INHIBIT_PAYLOAD	Inhibit TLP Payload Data for TLP's in Matched Region; assign iATU region to be TLP without data. When enabled and region address is matched, the iATU marks all TLPs as having no payload data by forcing the TLP header Fmt[1] bit =0, regardless of the application inputs such as slv_wstrb. - 1: Fmt[1] =0 so that only TLP type without data is sent. For example, a Msg instead of MsgD will be sent. - 0: Fmt[1] =0/1 so that TLPs with or without data can be sent. Note: This register field is sticky.
21 —	Reserved.
20 SNP	Serialize Non-Posted Requests. In this mode, when the AXI Bridge is populated, same AXI ID Non-Posted Read/Write Requests are transmitted on the wire if there are no other same ID Non-Posted Requests outstanding. Note: This register field is sticky.
19 FUNC_BYPASS	Function Number Translation Bypass. In this mode, the function number of the translated TLP is taken from your application transmit interface and not from the CTRL_1_FUNC_NUM field of the "iATU Control 1 Register" or the VF_NUMBER field of the "iATU Control 3 Register." Note: This register field is sticky.
18-17 —	Reserved.
16 TAG_SUBSTITUTE_EN	TAG Substitute Enable. When enabled and region address is matched, the iATU substitutes the TAG field of the outbound TLP header with the contents of the TAG field in this register. The expected usage scenario is translation from AXI MWr to Vendor Defined Msg/MsgD. Note: This register field is sticky.
15-8 TAG	TAG. The substituted TAG field (byte 6) in the outgoing TLP header when TAG_SUBSTITUTE_EN is set. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs (Message Code). When the address of an outbound TLP is matched to this region, and the translated TLP TYPE field is Msg or MsgD; then the message field of the TLP is changed to the value in this register. Memory TLPs: (ST;Steering Tag). When the ST field of an outbound TLP is matched to this region, and the translated TLP TYPE field targets memory space; then the ST field of the TLP is changed to the value in this register. Note: This register field is sticky.

11.3.3.1.172 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3)

11.3.3.1.172.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_OUTBOUND_3	8000_0608h

11.3.3.1.172.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.172.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LWR_BASE_HW															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.172.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.173 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3)

11.3.3.1.173.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_OUTBOUND_3	8000_060Ch

11.3.3.1.173.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.173.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. In systems with a 32-bit address space, this register is not used and therefore writing to this register has no effect. Note: This register field is sticky.

11.3.3.1.174 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_OUTBOUND_3)

11.3.3.1.174.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_OUTBOUND_3	8000_0610h

11.3.3.1.174.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.174.3 Fields

Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.175 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3)

11.3.3.1.175.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_OUTBOUND_3	8000_0614h

11.3.3.1.175.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LWR_TARGET_RW_OUTBOUND															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.175.3 Fields

Field	Function
31-0 LWR_TARGET_RW_OUTBOUND	When HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_ is '0' (normal operation): - LWR_TARGET_RW[31:n] forms MSB's of the Lower Target part of the new address of the translated region; - LWR_TARGET_RW[n-1:0] are not used. (The start address must be aligned to the Minimum Size of iATU Region kB boundary, so the lower bits of the start address of the new address of the translated region (bits n-1:0) are always '0'). - n is log2(Minimum Size of iATU Region). When

Field	Function
	HEADER_SUBSTITUTE_EN in IATU_REGION_CTRL_2_OFF_OUTBOUND_i is '1': - LWR_TARGET_RW[31:0] forms bytes 8-11 (for 3 dword header) or bytes 12-15 (for 4 dword header) of the outbound TLP header. Usage scenarios include the transmission of Vendor Defined Messages where the controller determines the content of bytes 12 to 15 of the TLP header. Note: This register field is sticky.

11.3.3.1.176 iATU Upper Target Address Register. (IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3)

11.3.3.1.176.1 Offset

Register	Offset
IATU_UPPER_TARGET_ADDR_OFF_OUTBOUND_3	8000_0618h

11.3.3.1.176.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_TARGET_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.176.3 Fields

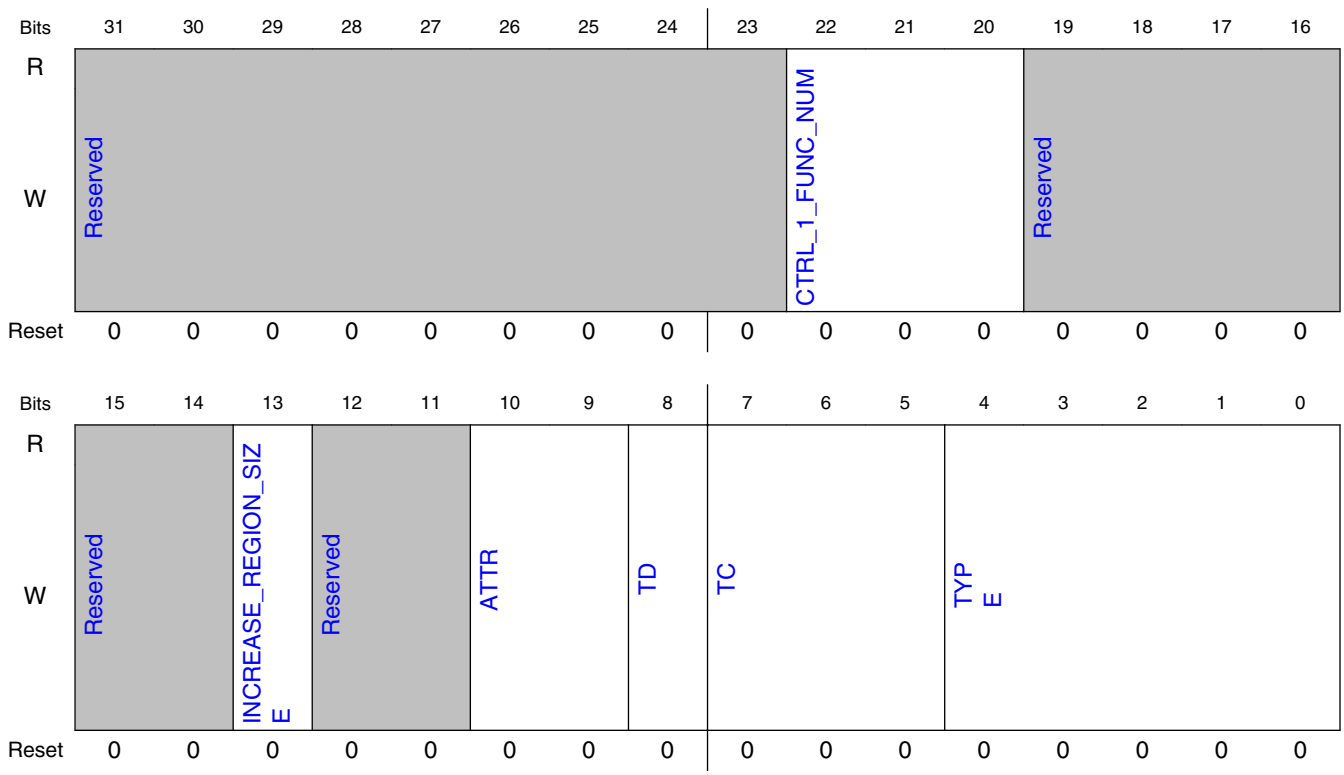
Field	Function
31-0 UPPER_TARGET_RW	Forms bits [63:32] of the start address (Upper Target part) of the new address of the translated region. Note: This register field is sticky.

11.3.3.1.177 iATU Region Control 1 Register. (IATU_REGION_CTRL_1_OFF_INBOUND_3)

11.3.3.1.177.1 Offset

Register	Offset
IATU_REGION_CTRL_1_OFF_INBOUND_3	8000_0700h

11.3.3.1.177.2 Diagram



11.3.3.1.177.3 Fields

Field	Function
31-23 —	Reserved.
22-20 CTRL_1_FUNC_NUM	Function Number. - MEM-I/O: When the Address and BAR matching logic in the controller indicate that a MEM-I/O transaction matches a BAR in the function corresponding to this value, then address translation proceeds. This check is only performed if the "Function Number Match Enable" bit of the "iATU Control 2 Register" is set. - CFG0/CFG1: When the destination function number as specified in the routing ID of the TLP header matches the function, then address translation proceeds. This check is only performed if the

Table continues on the next page...

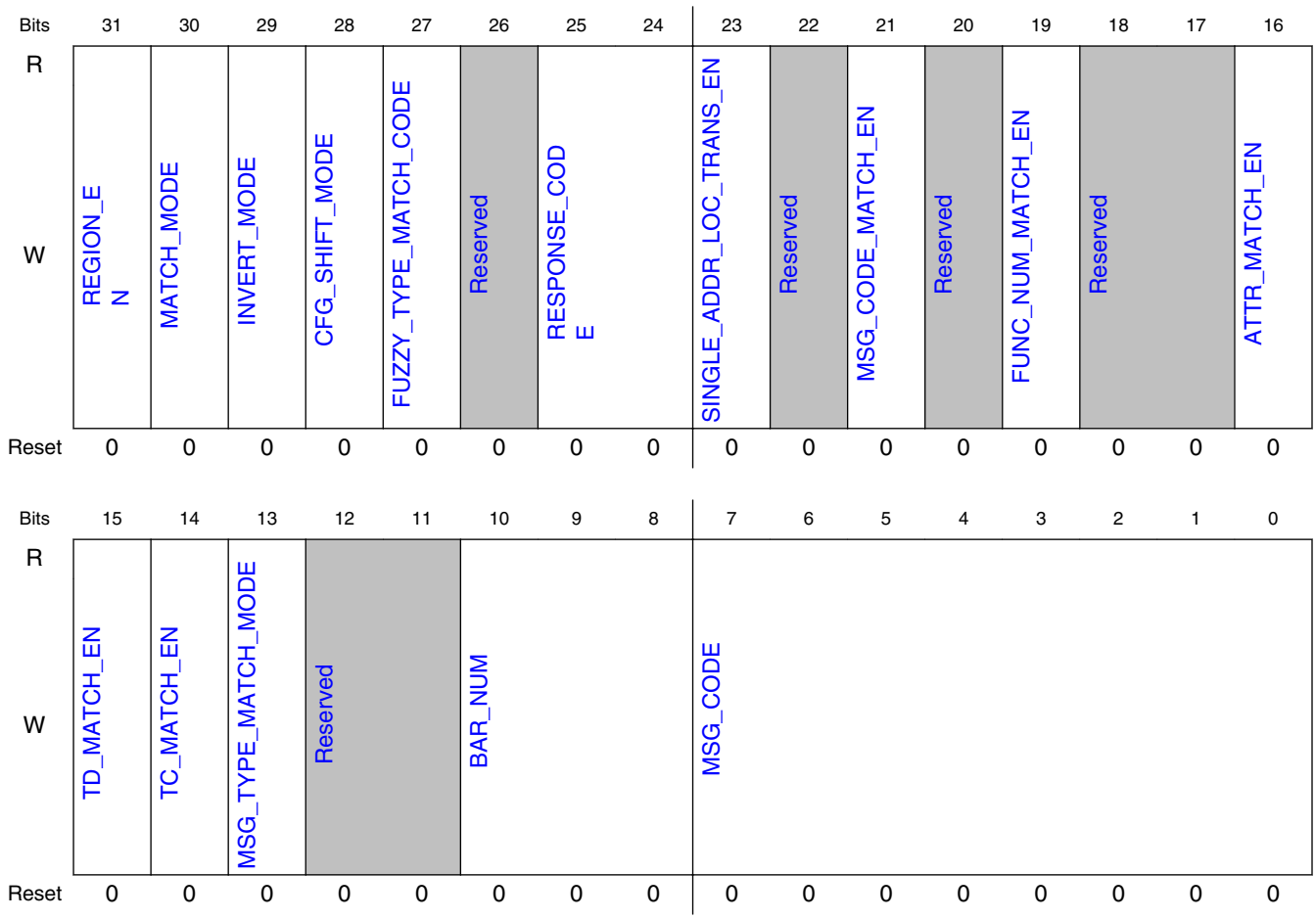
Field	Function
	"Function Number Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
19-14 —	Reserved.
13 INCREASE_REGION_SIZE	Increase the maximum ATU Region size. When set, the size is determined by the Maximum Size of iATU Region. When clear, the maximum ATU Region size is 4 GB (default). Note: This register field is sticky.
12-11 —	Reserved.
10-9 ATTR	When the ATTR field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "ATTR Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
8 TD	When the TD field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TD Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
7-5 TC	When the TC field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "TC Match Enable" bit of the "iATU Control 2 Register" is set. Note: This register field is sticky.
4-0 TYPE	When the TYPE field of an inbound TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). Note: This register field is sticky.

11.3.3.1.178 iATU Region Control 2 Register. (IATU_REGION_CTRL_2_OFF_INBOUND_3)

11.3.3.1.178.1 Offset

Register	Offset
IATU_REGION_CTRL_2_OFF_INBOUND_3	8000_0704h

11.3.3.1.178.2 Diagram



11.3.3.1.178.3 Fields

Field	Function
31 REGION_EN	Region Enable. This bit must be set to '1' for address translation to take place. Note: This register field is sticky.
30 MATCH_MODE	Match Mode. Determines Inbound matching mode for TLPs. The mode depends on the type of TLP that is received as follows: For MEM-I/O TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU operates using addresses as in the outbound direction. The Region Base and Limit Registers must be setup. - 1:BAR Match Mode. BAR matching is used. The "BAR Number" field is relevant. Not used for RC. For CFG0 TLPs, this field is interpreted as follows: - 0: Routing ID Match Mode. The iATU interprets the Routing ID (Bytes 8 to 11 of TLP header) as an address. This corresponds to the upper 16 bits of the address in MEM-I/O transactions. The Routing ID of the TLP must be within the base and limit of the iATU region for matching to proceed. - 1: Accept Mode. The iATU accepts all CFG0 transactions as address matches. The routing ID in the CFG0 TLP is ignored. This is useful as all received CFG0 TLPs should be processed regardless of the Bus number. For MSG/MSGD TLPs, this field is interpreted as follows: - 0: Address Match Mode. The iATU treats the third dword and fourth dword of the inbound MSG/MSGD TLP as an address and it is matched against the Region Base and Limit Registers. - 1: Vendor ID Match Mode. This mode is relevant for ID-routed Vendor Defined Messages. The iATU ignores the Routing ID (Bus, Device, Function) in bits [31:16] of the third dword of the TLP header, but matches

Table continues on the next page...

Field	Function
	against the Vendor ID in bits [15:0] of the third dword of the TLP header. Bits [15:0] of the Region Upper Base register should be programmed with the required Vendor ID. The lower Base and Limit Register should be programmed to translate TLPs based on vendor specific information in the fourth dword of the TLP header. - If SINGLE_ADDRESS_LOCATION_TRANSLATE_EN = 1 AND MSG_TYPE_MATCH_MODE = 1, then Match Mode is ignored. Note: This register field is sticky.
29 INVERT_MODE	Invert Mode. When set the address matching region is inverted. Therefore, an address match occurs when the untranslated address is in the region outside the defined range (Base Address to Limit Address). Note: This register field is sticky.
28 CFG_SHIFT_MODE	CFG Shift Mode. This is useful for CFG transactions where the PCIe configuration mechanism maps bits [27:12] of the address to the bus/device and function number. This allows a CFG configuration space to be located in any 256MB window of your application memory space using a 28-bit effective address. Shifts bits [31:16] of the untranslated address to form bits [27:12] of the translated address. Note: This register field is sticky.
27 FUZZY_TYPE_MATCH_CODE	Fuzzy Type Match Mode. When enabled, the iATU relaxes the matching of the TLP TYPE field against the expected TYPE field so that - CfgRd0 and CfgRd1 TLPs are seen as identical. Similarly with CfgWr0 and CfgWr1. - MWr, MRd and MRdLk TLPs are seen as identical - The Routing field of Msg/MsgD TLPs is ignored - FetchAdd, Swap and CAS are seen as identical. For example, CFG0 in the TYPE field in the "iATU Control 1 Register" matches against an inbound CfgRd0, CfgRd1, CfgWr0 or CfgWr1 TLP. Note: This register field is sticky.
26 —	Reserved.
25-24 RESPONSE_CODE	Response Code. Defines the type of response to give for accesses matching this region. This overrides the normal RADM filter response. Note that this feature is not available for any region where Single Address Location Translate is enabled. - 00 - Normal RADM filter response is used. - 01 - Unsupported request (UR) - 10 - Completer abort (CA) - 11 - Not used / undefined / reserved. Note: This register field is sticky.
23 SINGLE_ADDR_LOC_TRANS_EN	Single Address Location Translate Enable. When enabled, Rx TLPs can be translated to a single address location as determined by the target address register of the iATU region. The main usage scenario is translation of Messages (such as Vendor Defined or ATS Messages) to MWr TLPs when the AXI bridge is enabled. Note: This register field is sticky.
22 —	Reserved.
21 MSG_CODE_MATCH_EN	Message Code Match Enable (Msg TLPS). Ensures that a successful message Code TLP field comparison match (see Message Code field of the "iATU Control 2 Register") occurs (in MSG transactions) for address translation to proceed. ST Match Enable (Mem TLPs). Ensures that a successful ST TLP field comparison match (see ST field of the "iATU Control 2 Register") occurs (in MEM transactions) for address translation to proceed. Note: This register field is sticky.
20 —	Reserved.
19 FUNC_NUM_MATCH_EN	Function Number Match Enable. Ensures that a successful Function Number TLP field comparison match (see Function Number field of the "iATU Control 1 Register") occurs (in MEM-I/O and CFG0/CFG1 transactions) for address translation to proceed. Note: This register field is sticky.
18-17 —	Reserved.
16 ATTR_MATCH_EN	ATTR Match Enable. Ensures that a successful ATTR TLP field comparison match (see ATTR field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15 TD_MATCH_EN	TD Match Enable. Ensures that a successful TD TLP field comparison match (see TD field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
14 TC_MATCH_EN	TC Match Enable. Ensures that a successful TC TLP field comparison match (see TC field of the "iATU Control 1 Register") occurs for address translation to proceed. Note: This register field is sticky.
13 MSG_TYPE_MATCH_MODE	Message Type Match Mode. When enabled, and if single address location translate enable is set, then inbound TLPs of type MSG/MSGd which match the type field of the iatu_region_ctrl_1_OFF_inbound register (=>TYPE[4:3]=2'b10) will be translated. Message type match mode overrides any value of MATCH_MODE field in this register. Usage scenarios for this are translation of VDM or ATS messages when AXI bridge is configured on client interface. Note: This register field is sticky.
12-11 —	Reserved.
10-8 BAR_NUM	BAR Number. When the BAR number of an inbound MEM or IO TLP " that is matched by the normal internal BAR address matching mechanism " is the same as this field, address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Match Mode" bit of the "iATU Control 2 Register" is set. - 000b - BAR0 - 001b - BAR1 - 010b - BAR2 - 011b - BAR3 - 100b - BAR4 - 101b - BAR5 - 110b - ROM - 111b - reserved - IO translation would require either 00100b or 00101b in the inbound TLP TYPE; the BAR Number set in the range 000b - 101b and that BAR configured as an IO BAR. Note: This register field is sticky.
7-0 MSG_CODE	MSG TLPs: (Message Code). When the TYPE field of an inbound Msg/MsgD TLP is matched to this value, then address translation proceeds (when all other enabled field-matches are successful). This check is only performed if the "Message Code Match Enable" bit of the "iATU Control 2 Register" is set. Memory TLPs: (ST;Steering Tag). When the ST field of an inbound TLP is matched to this value, then address translation proceeds. This check is only performed if the "ST Match Enable" bit of the "iATU Control2 Register" is set. The setting is independent of the setting of the TH field. Note: This register field is sticky.

11.3.3.1.179 iATU Lower Base Address Register. (IATU_LWR_BASE_ADDR_OFF_INBOUND_3)

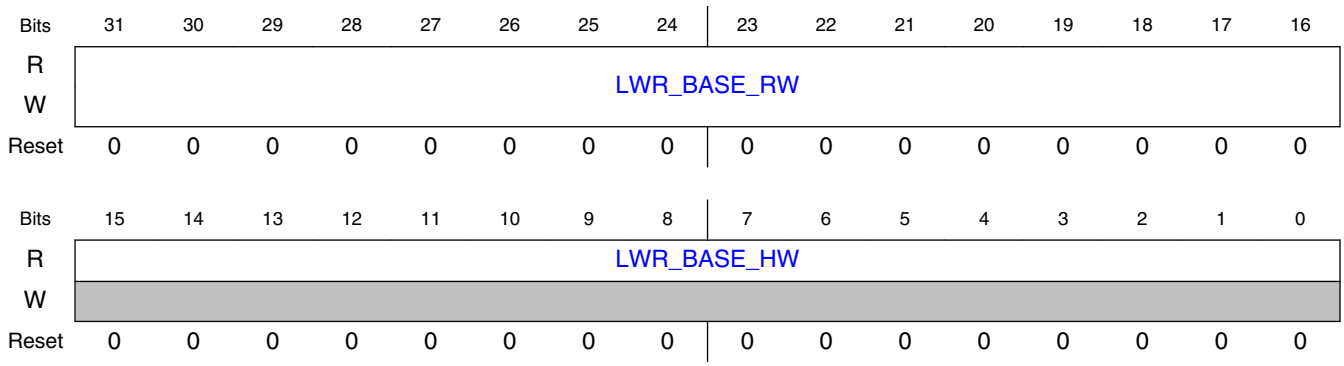
11.3.3.1.179.1 Offset

Register	Offset
IATU_LWR_BASE_ADDR_OFF_INBOUND_3	8000_0708h

11.3.3.1.179.2 Function

iATU Lower Base Address Register. The Minimum Size of iATU Region (Value Range: 4 kB, 8 kB, 16 kB, 32 kB, 64 kB defaults to 64 kB) specifies the minimum size of an address translation region. For example, if set to 64 kB; the lower 16 bits of the Base, Limit and Target registers are zero and all address regions are aligned on 64 kB boundaries. More precisely, the lower log2(Minimum Size of iATU Region) bits are zero.

11.3.3.1.179.3 Diagram



11.3.3.1.179.4 Fields

Field	Function
31-16 LWR_BASE_RW	Forms bits [31:n] of the start address of the address region to be translated. n is log2(Minimum Size of iATU Region) Note: This register field is sticky.
15-0 LWR_BASE_HW	Forms bits [n-1:0] of the start address of the address region to be translated. The start address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. n is log2(Minimum Size of iATU Region)

11.3.3.1.180 iATU Upper Base Address Register. (IATU_UPPER_BASE_ADDR_OFF_INBOUND_3)

11.3.3.1.180.1 Offset

Register	Offset
IATU_UPPER_BASE_ADDR_OFF_INBOUND_3	8000_070Ch

11.3.3.1.180.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	UPPER_BASE_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.180.3 Fields

Field	Function
31-0 UPPER_BASE_RW	Forms bits [63:32] of the start (and end) address of the address region to be translated. Note: This register field is sticky.

11.3.3.1.181 iATU Limit Address Register. (IATU_LIMIT_ADDR_OFF_INBOUND_3)

11.3.3.1.181.1 Offset

Register	Offset
IATU_LIMIT_ADDR_OFF_INBOUND_3	8000_0710h

11.3.3.1.181.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LIMIT_ADDR_RW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	LIMIT_ADDR_HW															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

11.3.3.1.181.3 Fields

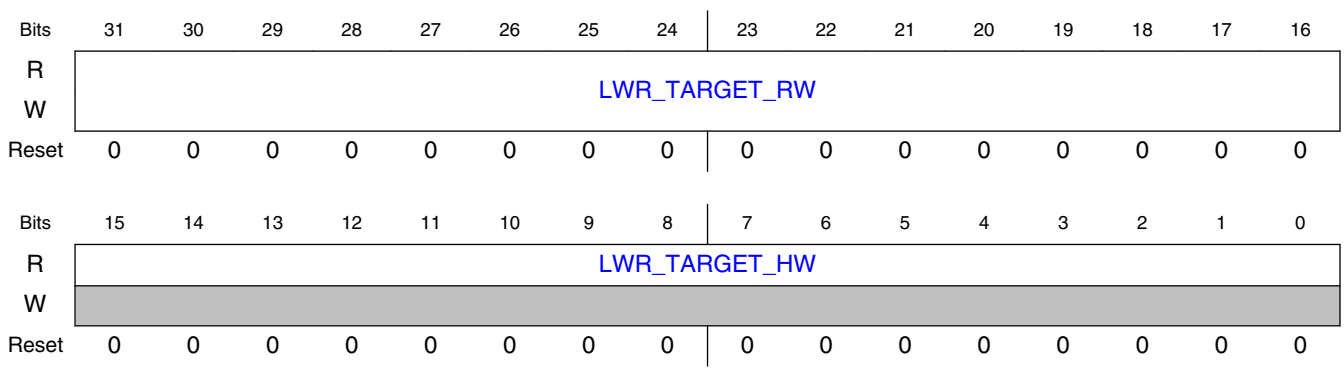
Field	Function
31-16 LIMIT_ADDR_RW	Forms upper bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller. Note: This register field is sticky.
15-0 LIMIT_ADDR_HW	Forms lower bits of the end address of the address region to be translated. The end address must be aligned to the Minimum Size of iATU Region kB boundary, so these bits are always 0. A write to this location is ignored by the PCIe controller.

11.3.3.1.182 iATU Lower Target Address Register. (IATU_LWR_TARGET_ADDR_OFF_INBOUND_3)

11.3.3.1.182.1 Offset

Register	Offset
IATU_LWR_TARGET_ADDR_OFF_INBOUND_3	8000_0714h

11.3.3.1.182.2 Diagram



11.3.3.1.182.3 Fields

Field	Function
31-16 LWR_TARGET_RW	Forms MSB's of the Lower Target part of the new address of the translated region. These bits are always '0'. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK+1) in BAR match mode. Note: This register field is sticky.

Table continues on the next page...

Field	Function
15-0 LWR_TARGET_HW	Forms the LSB's of the Lower Target part of the new address of the translated region. The start address must be aligned to the Minimum Size of iATU Region kB boundary (in address match mode); and to the Bar size boundary (in BAR match mode) so that these bits are always '0'. If the BAR is smaller than the iATU region size, then the iATU target address must align to the iATU region size; otherwise it must align to the BAR size. A write to this location is ignored by the PCIe controller. - Field size depends on log2(Minimum Size of iATU Region) in address match mode. - Field size depends on log2(BAR_MASK +1) in BAR match mode.

11.3.3.1.183 DMA Arbitration Scheme for TRGT1 Interface. (DMA_CTRL_DATA_ARB_PRIOR_OFF)

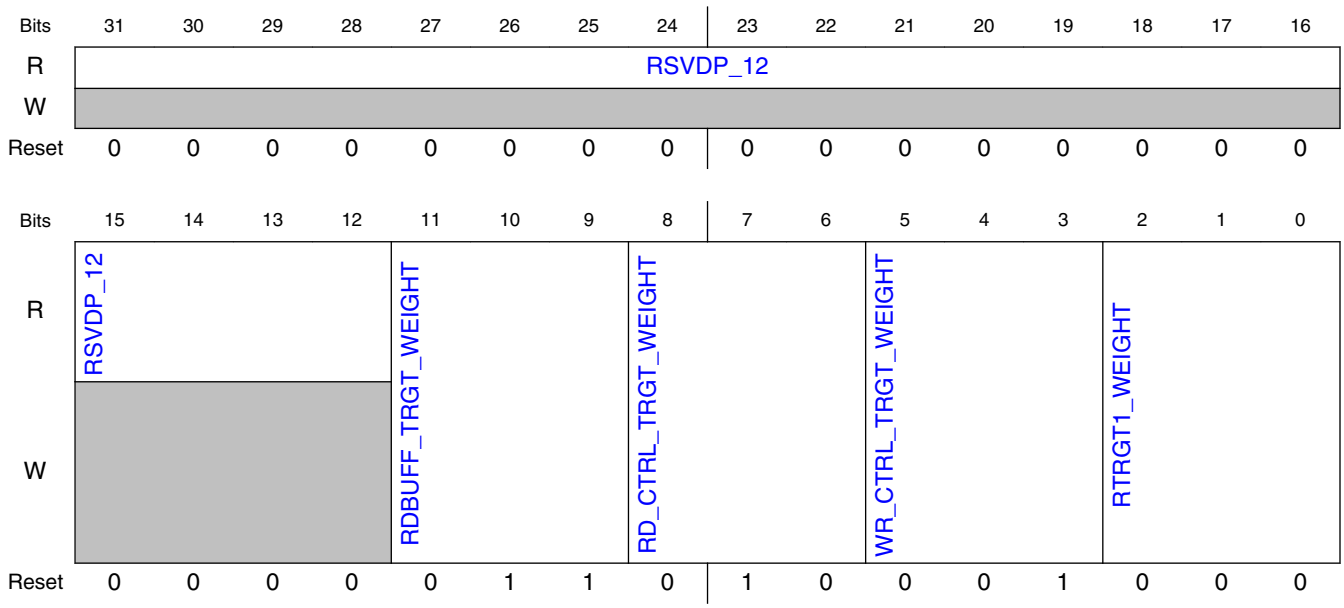
11.3.3.1.183.1 Offset

Register	Offset
DMA_CTRL_DATA_ARB_PRIOR_OFF	8008_0000h

11.3.3.1.183.2 Function

DMA Arbitration Scheme for TRGT1 Interface. This register is used to control traffic priorities among various sources that are delivered to your application through TRGT1 where 0x0 represents the highest priority. - Non-DMA Rx Requests - DMA Write Channel MRd Requests (DMA data requests and LL element/descriptor access) - DMA Read Channel MRd Requests (LL element/descriptor access) - DMA Read Channel MWr Requests Concurrent traffic from channels with same priority are sorted according to Round-Robin arbitration rules. The arbitration priority defaults to Non-DMA requests (highest), Write Channel MRd, Read Channel MRd, Read Channel MWr.

11.3.3.1.183.3 Diagram



11.3.3.1.183.4 Fields

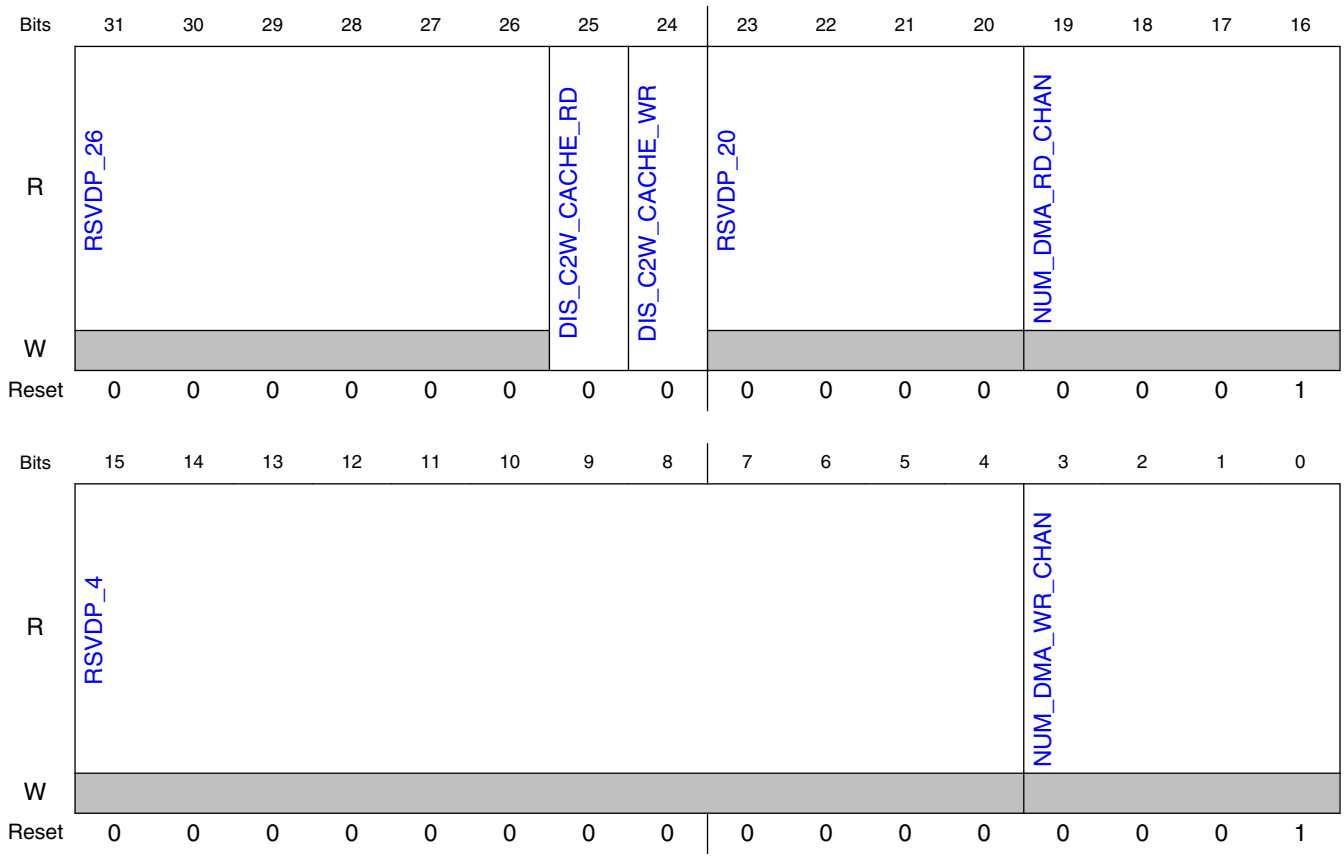
Field	Function
31-12 RSVDP_12	Reserved for future use.
11-9 RDBUFF_TRGT_WEIGHT	DMA Read Channel MWr Requests. Note: The access attributes of this field are as follows: - Dbi: R/W
8-6 RD_CTRL_TRGT_WEIGHT	DMA Read Channel MRd Requests. For LL element/descriptor access. Note: The access attributes of this field are as follows: - Dbi: R/W
5-3 WR_CTRL_TRGT_WEIGHT	DMA Write Channel MRd Requests. For DMA data requests and LL element/descriptor access. Note: The access attributes of this field are as follows: - Dbi: R/W
2-0 RTRGT1_WEIGHT	Non-DMA Rx Requests. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.184 DMA Number of Channels Register. (DMA_CTRL_OFF)

11.3.3.1.184.1 Offset

Register	Offset
DMA_CTRL_OFF	8008_0008h

11.3.3.1.184.2 Diagram



11.3.3.1.184.3 Fields

Field	Function
31-26 RSVDP_26	Reserved for future use.
25 DIS_C2W_CACHE_RD	Disable DMA Read Channels "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Dbi: R/W
24 DIS_C2W_CACHE_WR	Disable DMA Write Channels "completion to memory write" context cache pre-fetch function. Note: For internal debugging only. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

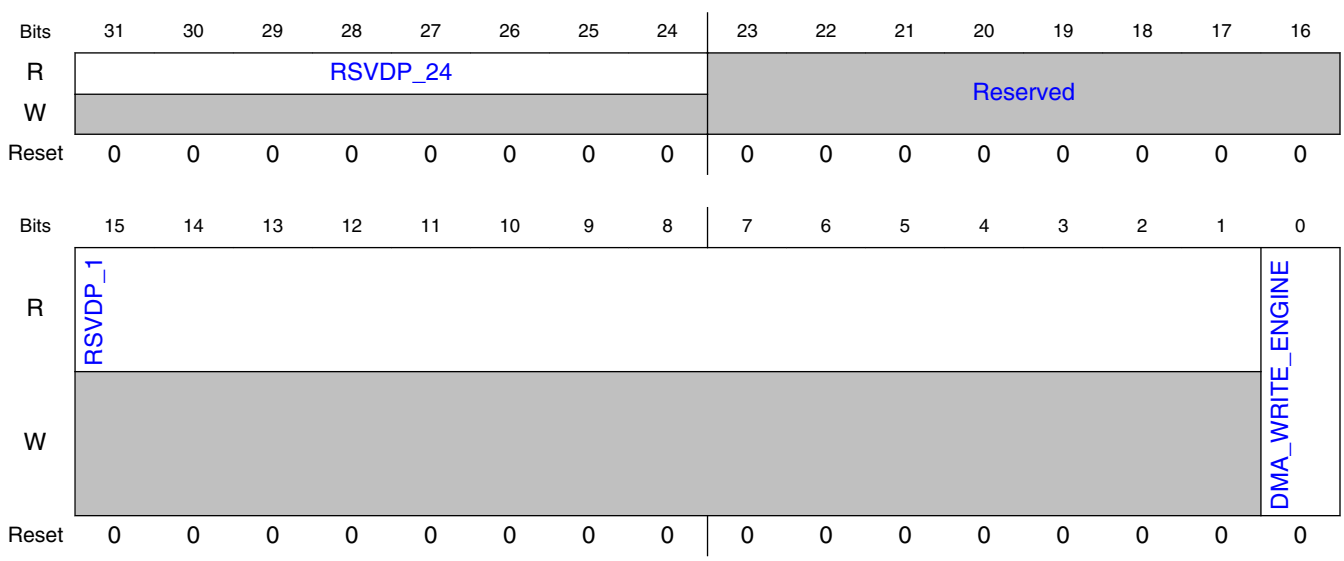
Field	Function
23-20 RSVDP_20	Reserved for future use.
19-16 NUM_DMA_RD_CHAN	Number of Read Channels. You can read this register to determine the number of read channels the DMA controller has been configured to support.
15-4 RSVDP_4	Reserved for future use.
3-0 NUM_DMA_WR_CHAN	Number of Write Channels. You can read this register to determine the number of write channels the DMA controller has been configured to support.

11.3.3.1.185 DMA Write Engine Enable Register. (DMA_WRITE_ENGINE_EN_OFF)

11.3.3.1.185.1 Offset

Register	Offset
DMA_WRITE_ENGINE_EN_OFF	8008_000Ch

11.3.3.1.185.2 Diagram



11.3.3.1.185.3 Fields

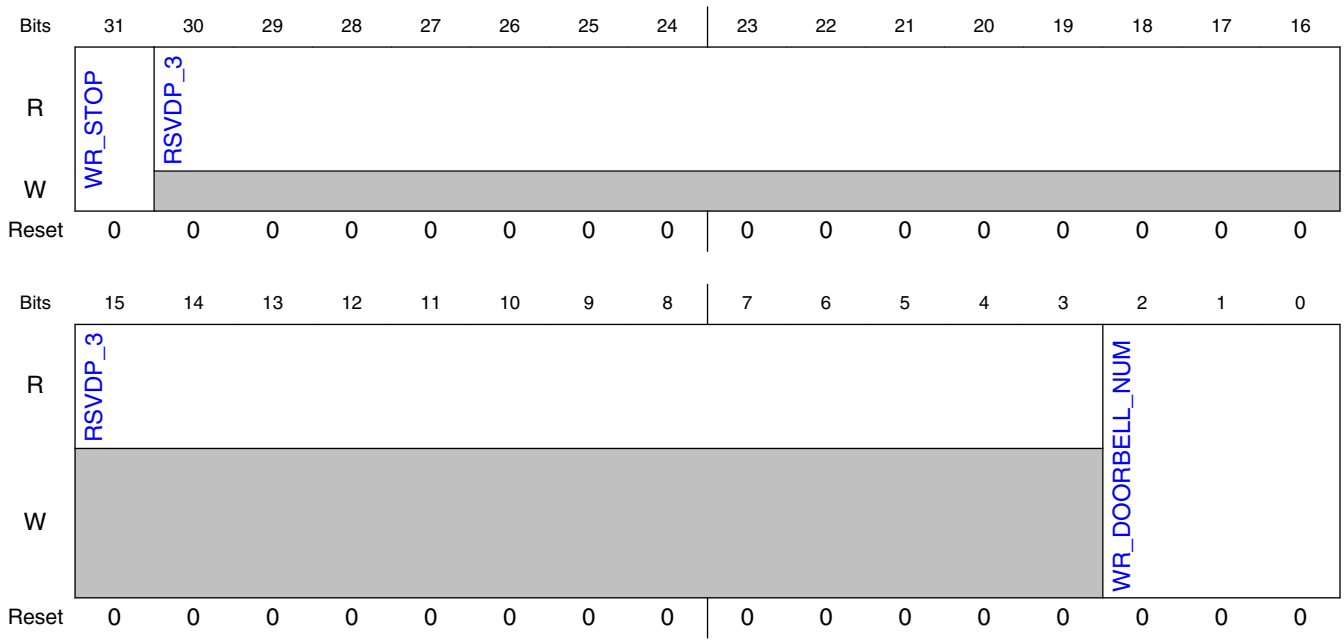
Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 —	Reserved.
15-1 RSVDP_1	Reserved for future use.
0 DMA_WRITE_ENGINE	DMA Write Engine Enable. - 1: Enable - 0: Disable (Soft Reset) For normal operation, you must initially set this bit to "1", before any other software setup actions. You do not need to toggle or rewrite to this bit during normal operation. You should set this bit to "0" when you want to "Soft Reset" the DMA controller write logic. There are three possible reasons for resetting the DMA controller write logic: - The "Abort Interrupt Status" bit is set (in the "DMA Write Interrupt Status Register" DMA_WRITE_INT_STATUS_OFF), and any of the bits is in the "DMA Write Error Status Register" (DMA_WRITE_ERR_STATUS_OFF) are set. Resetting the DMA controller write logic re-initializes the control logic, ensuring that the next DMA write transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit" , after which, the "Abort Interrupt Status" bit is set and the Channel Status field (CS) of the DMA write "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) is set to "Stopped." Resetting the DMA controller write logic re-initializes the control logic ensuring that the next DMA write transfer is executed successfully. - During software development, when you incorrectly program the DMA write engine. To "Soft Reset" the DMA controller write logic, you must: - De-assert the DMA write engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA write engine enable bit returns a "0". - Assert the DMA write engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA write transfer does not start until you write to the "DMA Write Doorbell Register" (DMA_WRITE_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.186 DMA Write Doorbell Register. (DMA_WRITE_DOORBELL_OFF)

11.3.3.1.186.1 Offset

Register	Offset
DMA_WRITE_DOORBELL_OFF	8008_0010h

11.3.3.1.186.2 Diagram



11.3.3.1.186.3 Fields

Field	Function
31 WR_STOP	Stop. Set in conjunction with the Doorbell Number field. The DMA write channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) to ensure that the write channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)." Note: The access attributes of this field are as follows: - Dbi: R/W
30-3 RSVDP_3	Reserved for future use.
2-0 WR_DOORBELL_NUM	Doorbell Number. You must write the channel number to this register to start the DMA write transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. You do not need to toggle or write any other value to this register to start a new transfer. The range of this field is 0x0 to 0x7, and 0x0 corresponds to channel 0. Also note that a write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.187 DMA Write Engine Channel Arbitration Weight Low Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF)

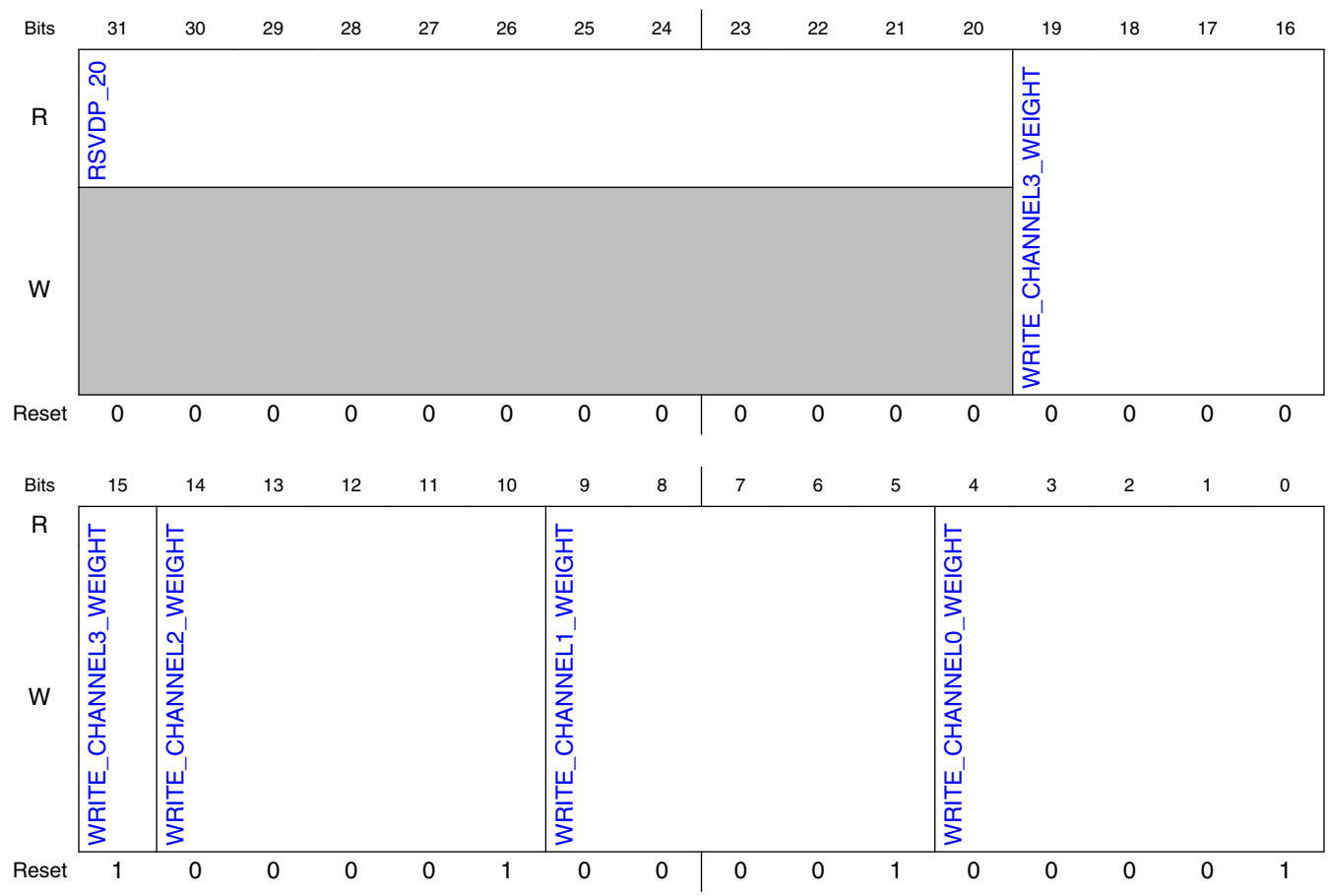
11.3.3.1.187.1 Offset

Register	Offset
DMA_WRITE_CHANNEL_ARB_WEIGHT_LOW_OFF	8008_0018h

11.3.3.1.187.2 Function

DMA Write Engine Channel Arbitration Weight Low Register. The 5-bit channel weight (for write channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.3.3.1.187.3 Diagram



11.3.3.1.187.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 WRITE_CHANNEL3_WEIGHT	Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 WRITE_CHANNEL2_WEIGHT	Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 WRITE_CHANNEL1_WEIGHT	Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0 WRITE_CHANNEL0_WEIGHT	Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.188 DMA Write Engine Channel Arbitration Weight High Register. (DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF)

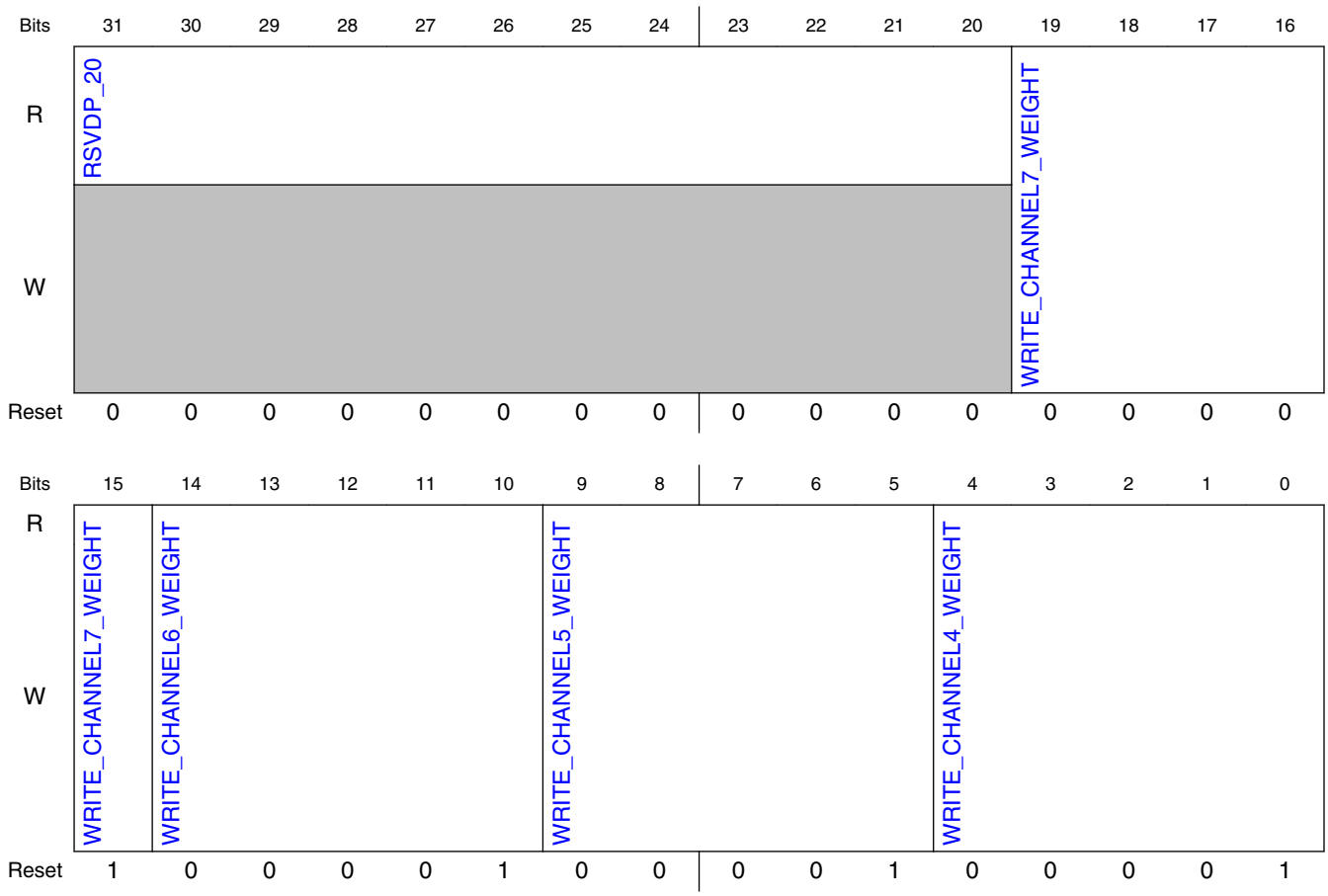
11.3.3.1.188.1 Offset

Register	Offset
DMA_WRITE_CHANNEL_ARB_WEIGHT_HIGH_OFF	8008_001Ch

11.3.3.1.188.2 Function

DMA Write Engine Channel Arbitration Weight High Register. The 5-bit channel weight (for write channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.3.3.1.188.3 Diagram



11.3.3.1.188.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 WRITE_CHANNEL7_WEIGHT	Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 WRITE_CHANNEL6_WEIGHT	Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 WRITE_CHANNEL5_WEIGHT	Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0	Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. A value of '0' means

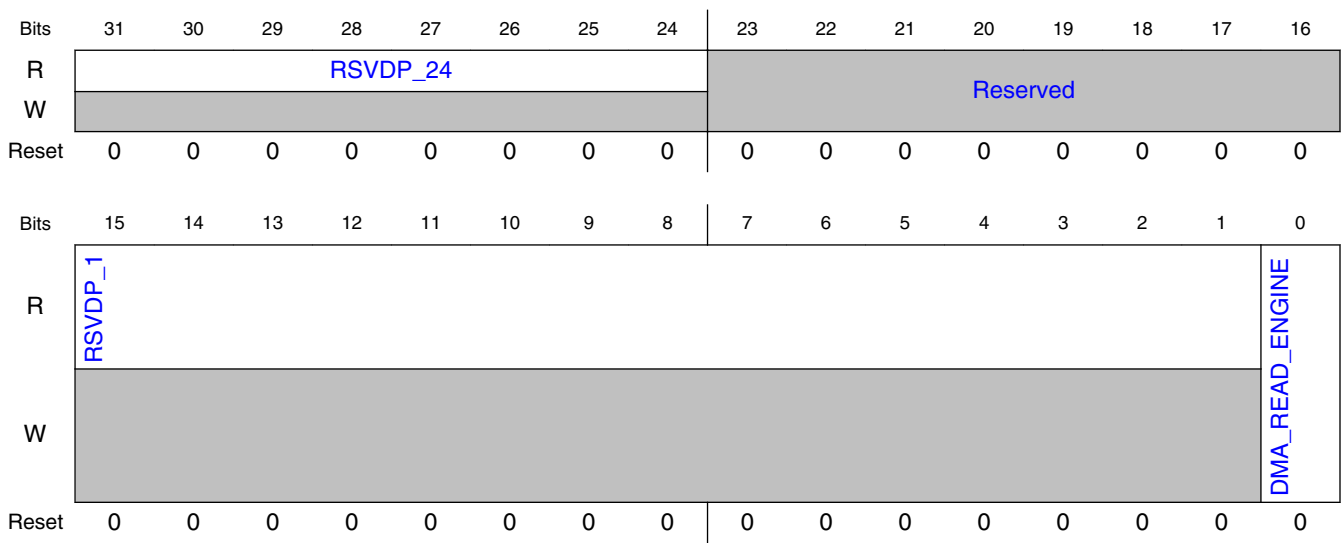
Field	Function
WRITE_CHANNEL4_WEIGHT	that one TLP is issued before moving to the next channel. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.189 DMA Read Engine Enable Register. (DMA_READ_ENGINE_EN_OFF)

11.3.3.1.189.1 Offset

Register	Offset
DMA_READ_ENGINE_EN_OFF	8008_002Ch

11.3.3.1.189.2 Diagram



11.3.3.1.189.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 —	Reserved.
15-1 RSVDP_1	Reserved for future use.

Table continues on the next page...

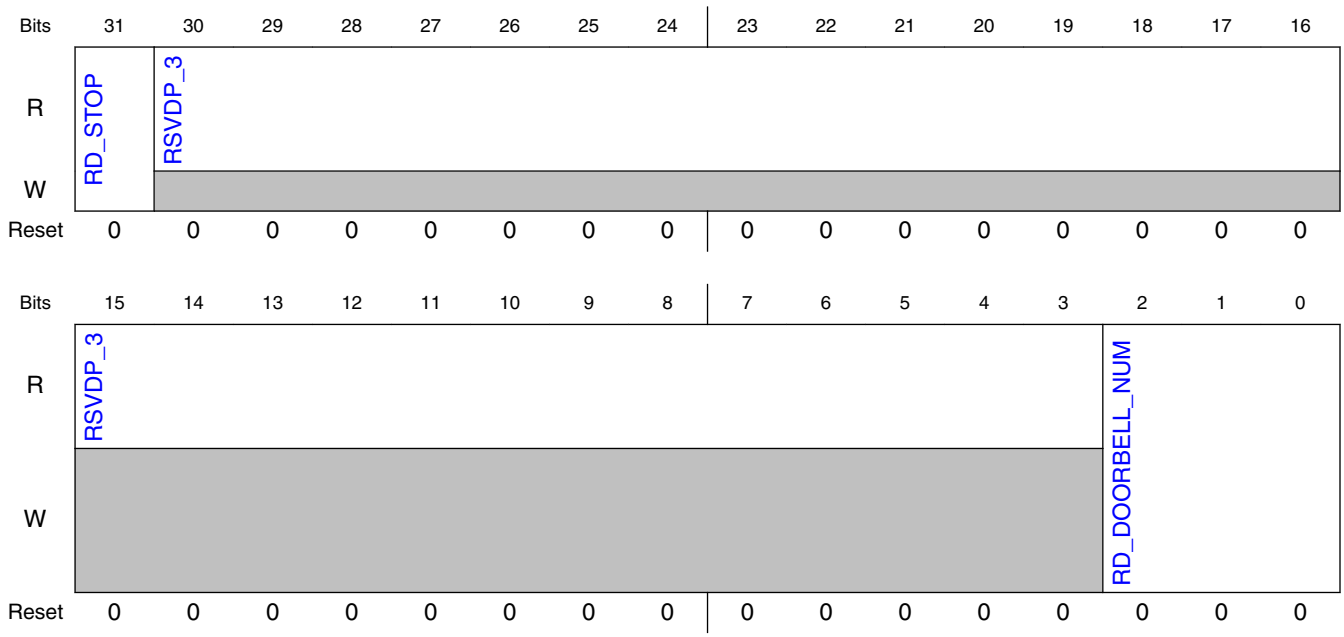
Field	Function
0 DMA_READ_ENGINE	DMA Read Engine Enable. - 1: Enable - 0: Disable (Soft Reset) For normal operation, you must initially set this bit to "1", before any other software setup actions. You do not need to toggle or rewrite to this bit during normal operation. You should set this field to "0" when you want to "Soft Reset" the DMA controller read logic. There are three possible reasons for resetting the DMA controller read logic: - The "Abort Interrupt Status" bit is set (in the "DMA Read Interrupt Status Register" (DMA_READ_INT_STATUS_OFF), and any of the bits in the "DMA Read Error Status Low Register" (DMA_READ_ERR_STATUS_LOW_OFF) is set. Resetting the DMA controller read logic re-initializes the control logic, ensuring that the next DMA read transfer is executed successfully. - You have executed the procedure outlined in "Stop Bit", after which, the "Abort Interrupt Status" bit is set and the channel Status field (CS) of the DMA read "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_WRCH_0) is set to "Stopped". Resetting the DMA controller read logic re-initializes the control logic ensuring that the next DMA read transfer is executed successfully. - During software development, when you incorrectly program the DMA read engine. To "Soft Reset" the DMA controller read logic, you must: - De-assert the DMA read engine enable bit. - Wait for the DMA to complete any in-progress TLP transfer, by waiting until a read on the DMA read engine enable bit returns a "0". - Assert the DMA read engine enable bit. This "Soft Reset" does not clear the DMA configuration registers. The DMA read transfer does not start until you write to the "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF). Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.190 DMA Read Doorbell Register. (DMA_READ_DOORBELL_OFF)

11.3.3.1.190.1 Offset

Register	Offset
DMA_READ_DOORBELL_OFF	8008_0030h

11.3.3.1.190.2 Diagram



11.3.3.1.190.3 Fields

Field	Function
31 RD_STOP	Stop. Set in conjunction with the Doorbell Number field. The DMA read channel stops issuing requests, sets the channel status to "Stopped", and asserts the "Abort" interrupt if it is enabled. Before setting the Stop bit, you must read the channel Status field (CS) of the "DMA Channel Control 1 Register " (DMA_CH_CONTROL1_OFF_RDCH_0) to ensure that the read channel is "Running" (transferring data). For more information, see "Stopping the DMA Transfer (Software Stop)". Note: The access attributes of this field are as follows: - Dbi: R/W
30-3 RSVDP_3	Reserved for future use.
2-0 RD_DOORBELL_NUM	Doorbell Number. You must write 0x0 to this register to start the DMA read transfer for that channel. The DMA detects a write to this register field even if the value of this field does not change. The range of this field is 0x0 to 0x7, and 0x0 corresponds to channel 0. Also note that a write to this field triggers the controller to exit L1 substates. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.191 DMA Read Engine Channel Arbitration Weight Low Register. (DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF)

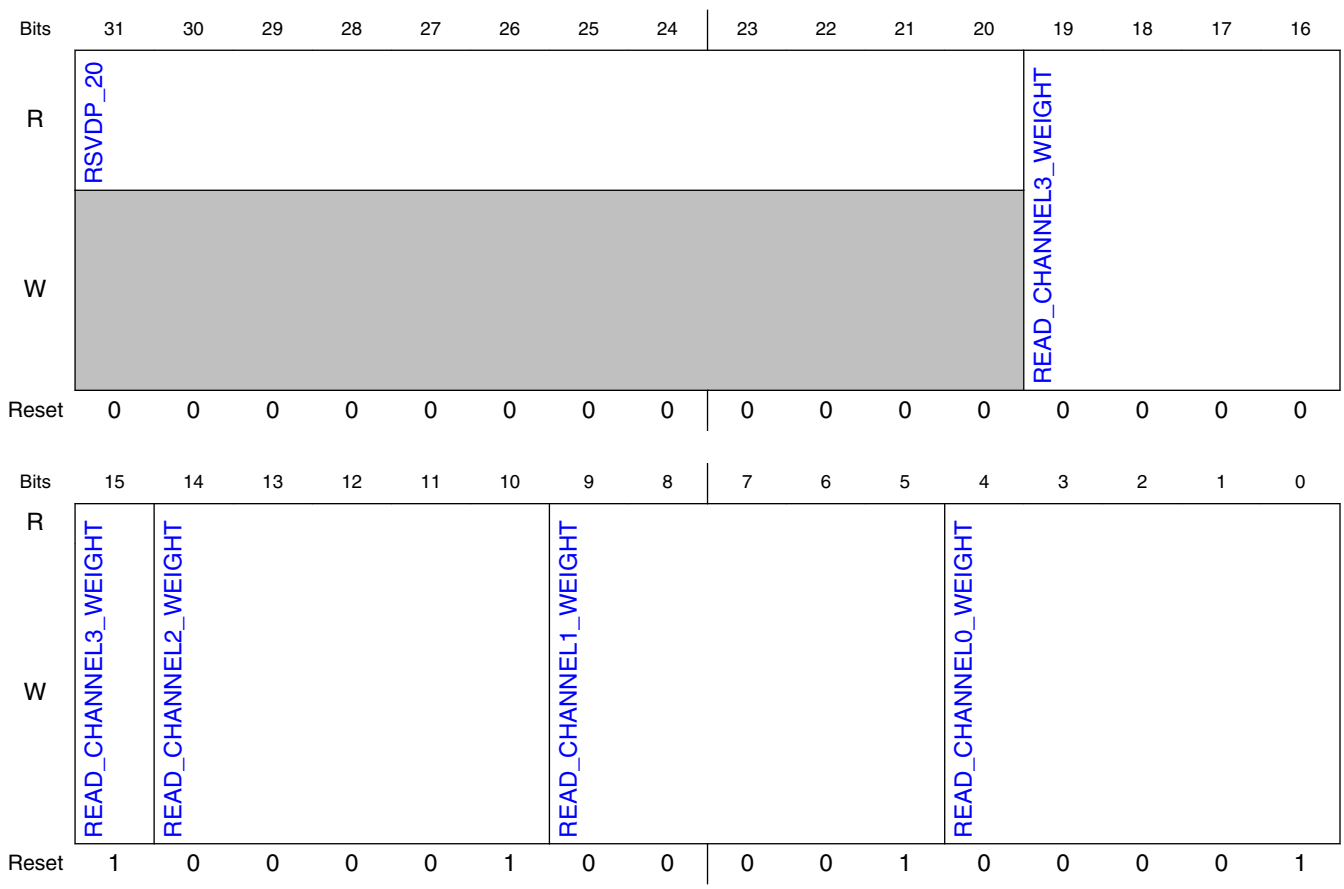
11.3.3.1.191.1 Offset

Register	Offset
DMA_READ_CHANNEL_ARB_WEIGHT_LOW_OFF	8008_0038h

11.3.3.1.191.2 Function

DMA Read Engine Channel Arbitration Weight Low Register. The 5-bit channel weight (for read channels 0-3) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.3.3.1.191.3 Diagram



11.3.3.1.191.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 READ_CHANNEL3_WEIGHT	Channel 3 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 READ_CHANNEL2_WEIGHT	Channel 2 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 READ_CHANNEL1_WEIGHT	Channel 1 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0 READ_CHANNEL0_WEIGHT	Channel 0 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.192 DMA Read Engine Channel Arbitration Weight High Register. (DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF)

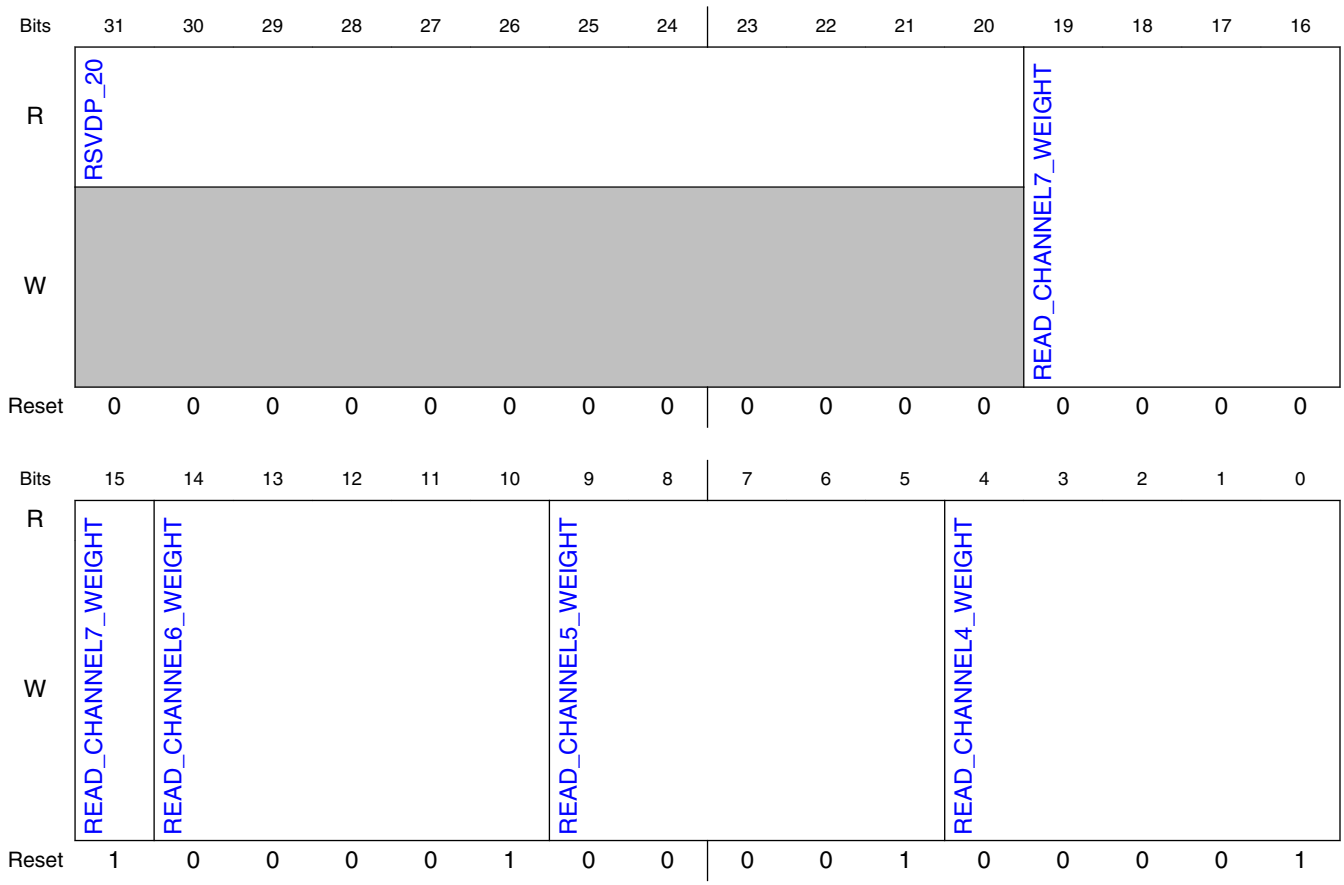
11.3.3.1.192.1 Offset

Register	Offset
DMA_READ_CHANNEL_ARB_WEIGHT_HIGH_OFF	8008_003Ch

11.3.3.1.192.2 Function

DMA Read Engine Channel Arbitration Weight High Register. The 5-bit channel weight (for read channels 4-7) specifies the maximum number of TLP requests that the DMA can issue for that channel before it must return to the arbitration routine. When the channel weight count is reached or DMA channel request transfer size reaches zero, the WWR arbiter selects the next channel to be processed. Your software must initialize this register before ringing the doorbell. For more details, see "Multichannel Arbitration". Value range is (0-0x1F) corresponding to (1-32) transaction requests.

11.3.3.1.192.3 Diagram



11.3.3.1.192.4 Fields

Field	Function
31-20 RSVDP_20	Reserved for future use.
19-15 READ_CHANNEL7_WEIGHT	Channel 7 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
14-10 READ_CHANNEL6_WEIGHT	Channel 6 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
9-5 READ_CHANNEL5_WEIGHT	Channel 5 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W
4-0 READ_CHANNEL4_WEIGHT	Channel 4 Weight. The weight is initialized by software before ringing the doorbell. The value is used by the channel weighted round robin arbiter to select the next channel read request. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.193 DMA Write Interrupt Status Register. (DMA_WRITE_INT_STATUS_OFF)

11.3.3.1.193.1 Offset

Register	Offset
DMA_WRITE_INT_STATUS_OFF	8008_004Ch

11.3.3.1.193.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24								WR_ABORT_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8								WR_DONE_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.193.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 WR_ABORT_INT_STATUS	Abort Interrupt Status. The DMA write channel has detected an error, or you manually stopped the transfer as described in "Error Handling Assistance by Remote Software". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-0 WR_DONE_INT_STATUS	Done Interrupt Status. The DMA write channel has successfully completed the DMA transfer. For more details, see "Interrupts and Error Handling". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA write interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to

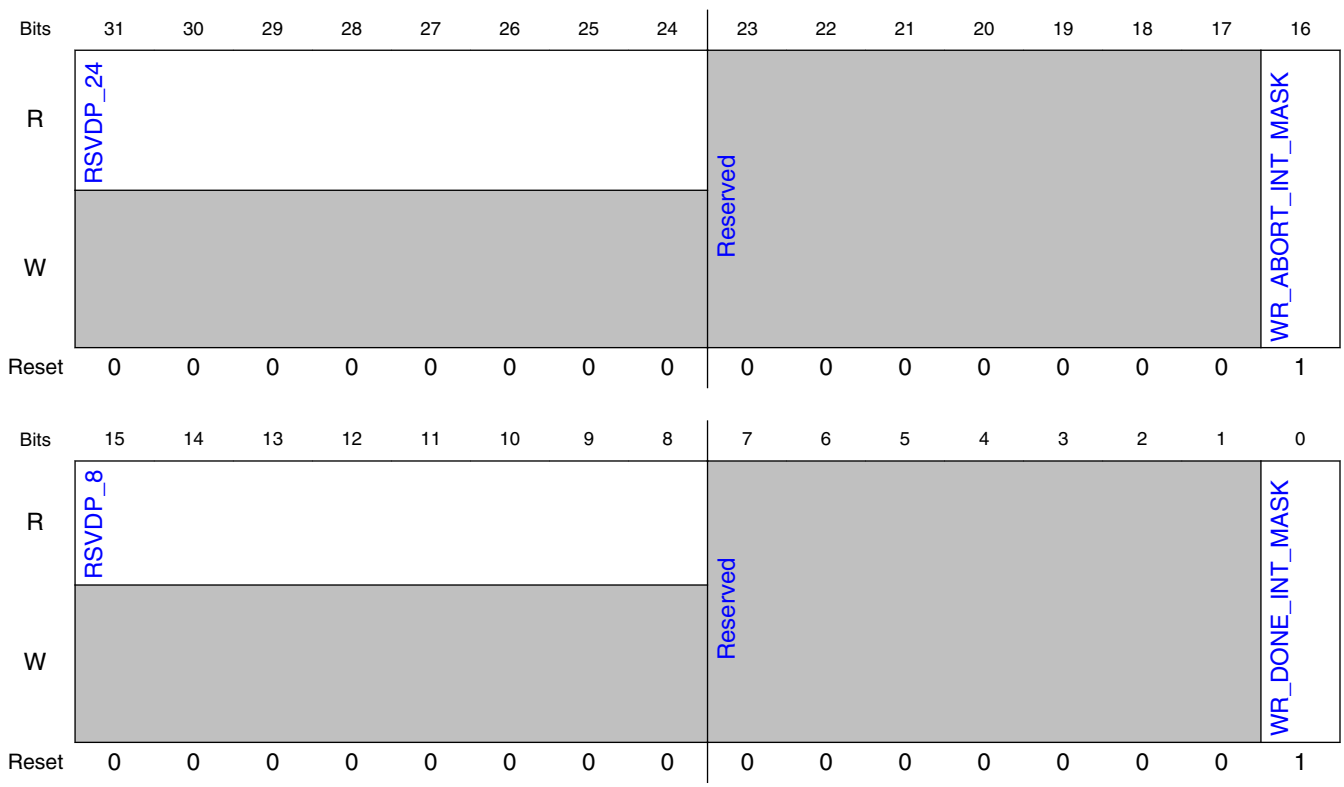
Field	Function
	the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.194 DMA Write Interrupt Mask Register. (DMA_WRITE_INT_MASK_OFF)

11.3.3.1.194.1 Offset

Register	Offset
DMA_WRITE_INT_MASK_OFF	8008_0054h

11.3.3.1.194.2 Diagram



11.3.3.1.194.3 Fields

Field	Function
31-24	Reserved for future use.

Table continues on the next page...

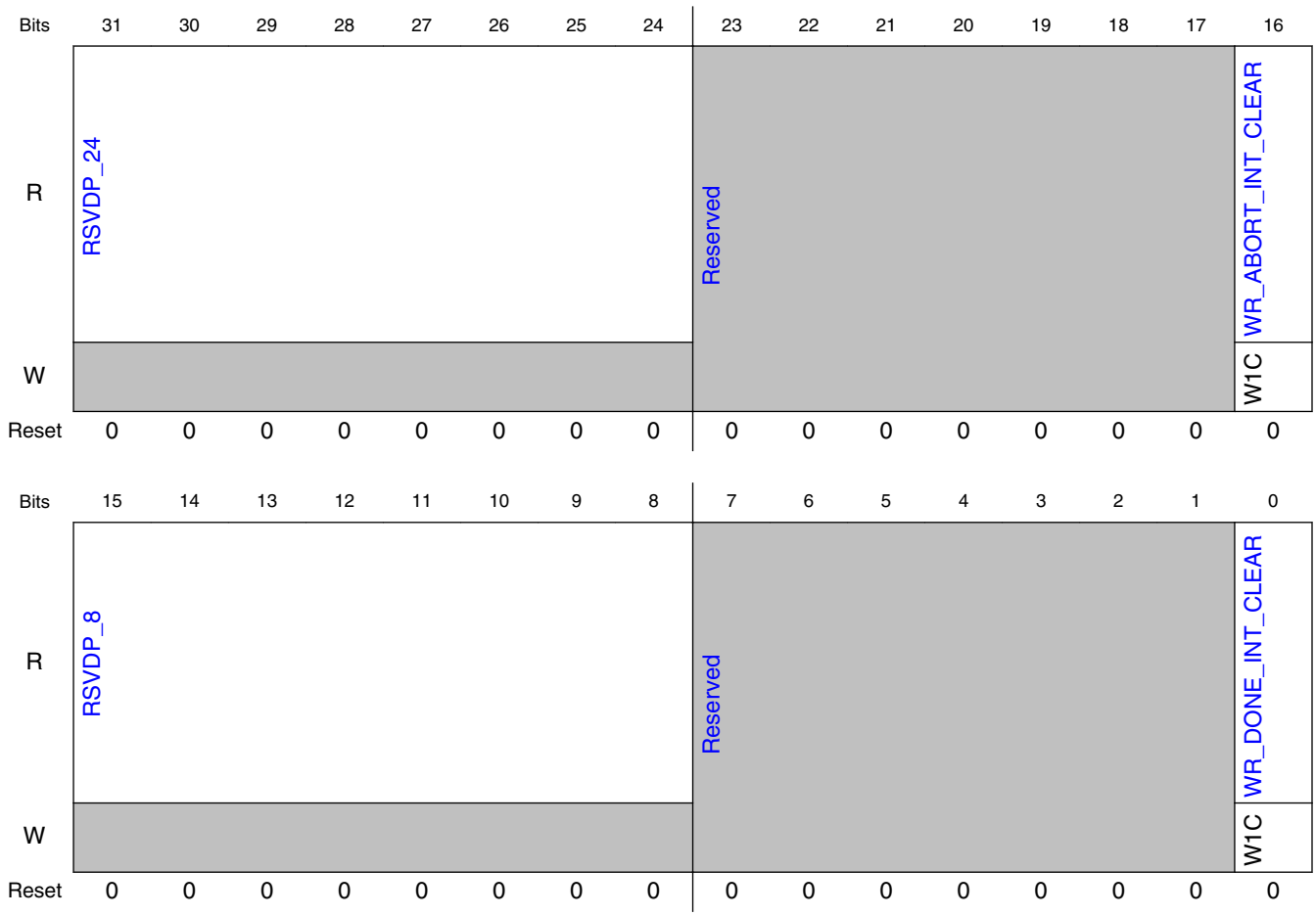
Field	Function
RSVDP_24	
23-17 —	Reserved.
16 WR_ABORT_INT_MASK	Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved.
0 WR_DONE_INT_MASK	Done Interrupt Mask. Prevents the Done interrupt status field in the DMA write interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.195 DMA Write Interrupt Clear Register. (DMA_WRITE_INT_CLEAR_OFF)

11.3.3.1.195.1 Offset

Register	Offset
DMA_WRITE_INT_CLEAR_OFF	8008_0058h

11.3.3.1.195.2 Diagram



11.3.3.1.195.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved.
16 WR_ABORT_INT_CLEAR	Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved.

Table continues on the next page...

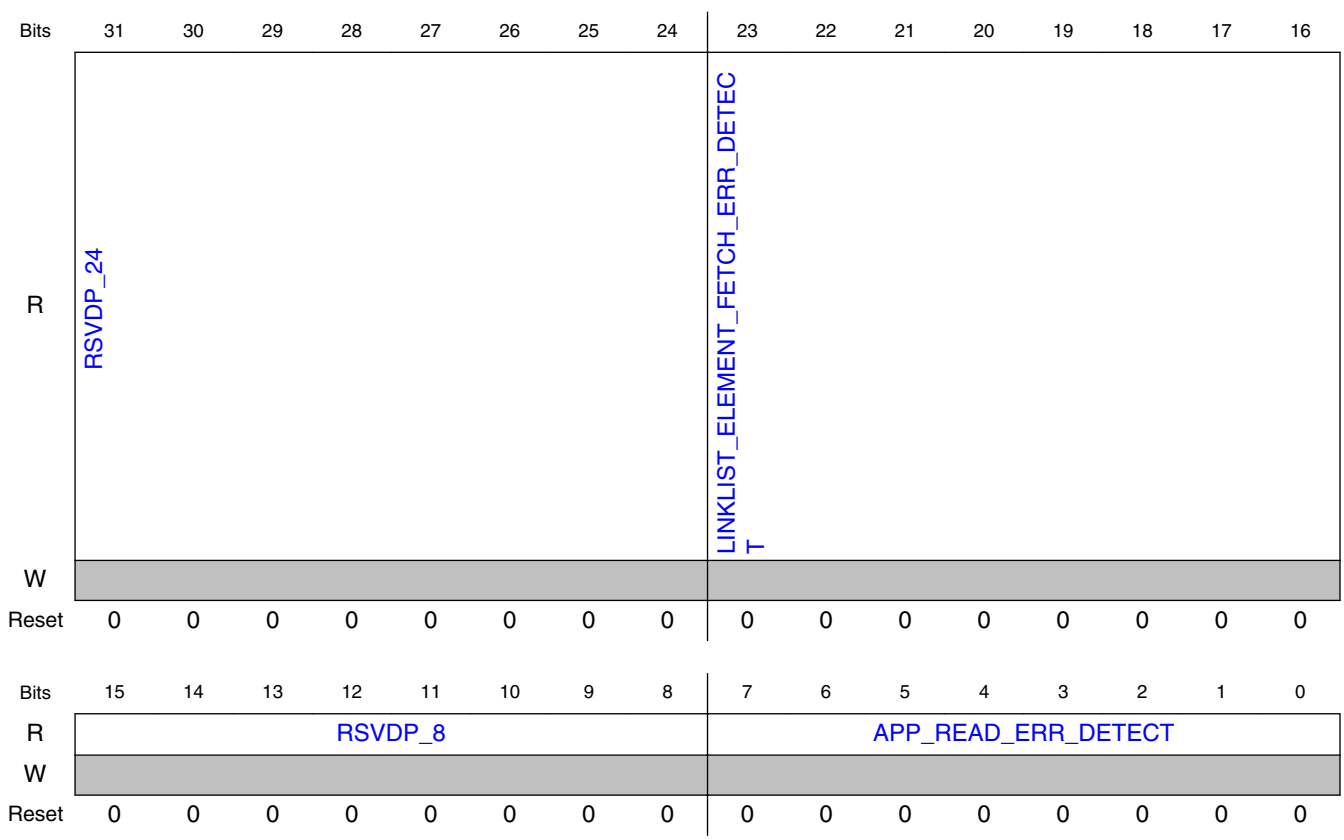
Field	Function
0 WR_DONE_INT_CLEAR	Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA write interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".

11.3.3.1.196 DMA Write Error Status Register (DMA_WRITE_ERR_STATUS_OFF)

11.3.3.1.196.1 Offset

Register	Offset
DMA_WRITE_ERR_STATUS_OFF	8008_005Ch

11.3.3.1.196.2 Diagram



11.3.3.1.196.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 LINKLIST_ELEMENT_FETCH_ERR_DETECT	Linked List Element Fetch Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Write Interrupt Clear Register" (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.
15-8 RSVDP_8	Reserved for future use.
7-0 APP_READ_ERR_DETECT	Application Read Error Detected. The DMA write channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while reading data from it. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA write interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Write Interrupt Clear Register" (DMA_WRITE_INT_CLEAR_OFF) to clear this error bit.

11.3.3.1.197 DMA Write Done IMWr Address Low Register. (DMA_WRITE_DONE_IMWR_LOW_OFF)

11.3.3.1.197.1 Offset

Register	Offset
DMA_WRITE_DONE_IMWR_LOW_OFF	8008_0060h

11.3.3.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.197.3 Fields

Field	Function
31-0 DMA_WRITE_DONE_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.198 DMA Write Done IMWr Interrupt Address High Register. (DMA_WRITE_DONE_IMWR_HIGH_OFF)

11.3.3.1.198.1 Offset

Register	Offset
DMA_WRITE_DONE_IMWR_HIGH_OFF	8008_0064h

11.3.3.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.198.3 Fields

Field	Function
31-0 DMA_WRITE_DONE_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.199 DMA Write Abort IMWr Address Low Register. (DMA_WRITE_ABORT_IMWR_LOW_OFF)

11.3.3.1.199.1 Offset

Register	Offset
DMA_WRITE_ABORT_IMWR_LOW_OFF	8008_0068h

11.3.3.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.199.3 Fields

Field	Function
31-0 DMA_WRITE_ABORT_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP it generates. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.200 DMA Write Abort IMWr Address High Register. (DMA_WRITE_ABORT_IMWR_HIGH_OFF)

11.3.3.1.200.1 Offset

Register	Offset
DMA_WRITE_ABORT_IMWR_HIGH_OFF	8008_006Ch

11.3.3.1.200.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_WRITE_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_WRITE_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.200.3 Fields

Field	Function
31-0 DMA_WRITE_ABORT_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.201 DMA Write Channel 1 and 0 IMWr Data Register. (DMA_WRITE_CH01_IMWR_DATA_OFF)

11.3.3.1.201.1 Offset

Register	Offset
DMA_WRITE_CH01_IMWR_DATA_OFF	8008_0070h

11.3.3.1.201.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_1_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_0_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.201.3 Fields

Field	Function
31-16 WR_CHANNEL_1_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 1. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_0_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.202 DMA Write Channel 3 and 2 IMWr Data Register. (DMA_WRITE_CH23_IMWR_DATA_OFF)

11.3.3.1.202.1 Offset

Register	Offset
DMA_WRITE_CH23_IMWR_DATA_OFF	8008_0074h

11.3.3.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_3_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_2_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.202.3 Fields

Field	Function
31-16 WR_CHANNEL_3_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 3. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
15-0 WR_CHANNEL_2_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 2. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.203 DMA Write Channel 5 and 4 IMWr Data Register. (DMA_WRITE_CH45_IMWR_DATA_OFF)

11.3.3.1.203.1 Offset

Register	Offset
DMA_WRITE_CH45_IMWR_DATA_OFF	8008_0078h

11.3.3.1.203.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_5_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_4_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.203.3 Fields

Field	Function
31-16 WR_CHANNEL_5_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 5. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_4_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 4. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.204 DMA Write Channel 7 and 6 IMWr Data Register. (DMA_WRITE_CH67_IMWR_DATA_OFF)

11.3.3.1.204.1 Offset

Register	Offset
DMA_WRITE_CH67_IMWR_DATA_OFF	8008_007Ch

11.3.3.1.204.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	WR_CHANNEL_7_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WR_CHANNEL_6_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.204.3 Fields

Field	Function
31-16 WR_CHANNEL_7_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 7. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 WR_CHANNEL_6_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for write channel 6. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.205 DMA Write Linked List Error Enable Register. (DMA_WRITE_LINKED_LIST_ERR_EN_OFF)

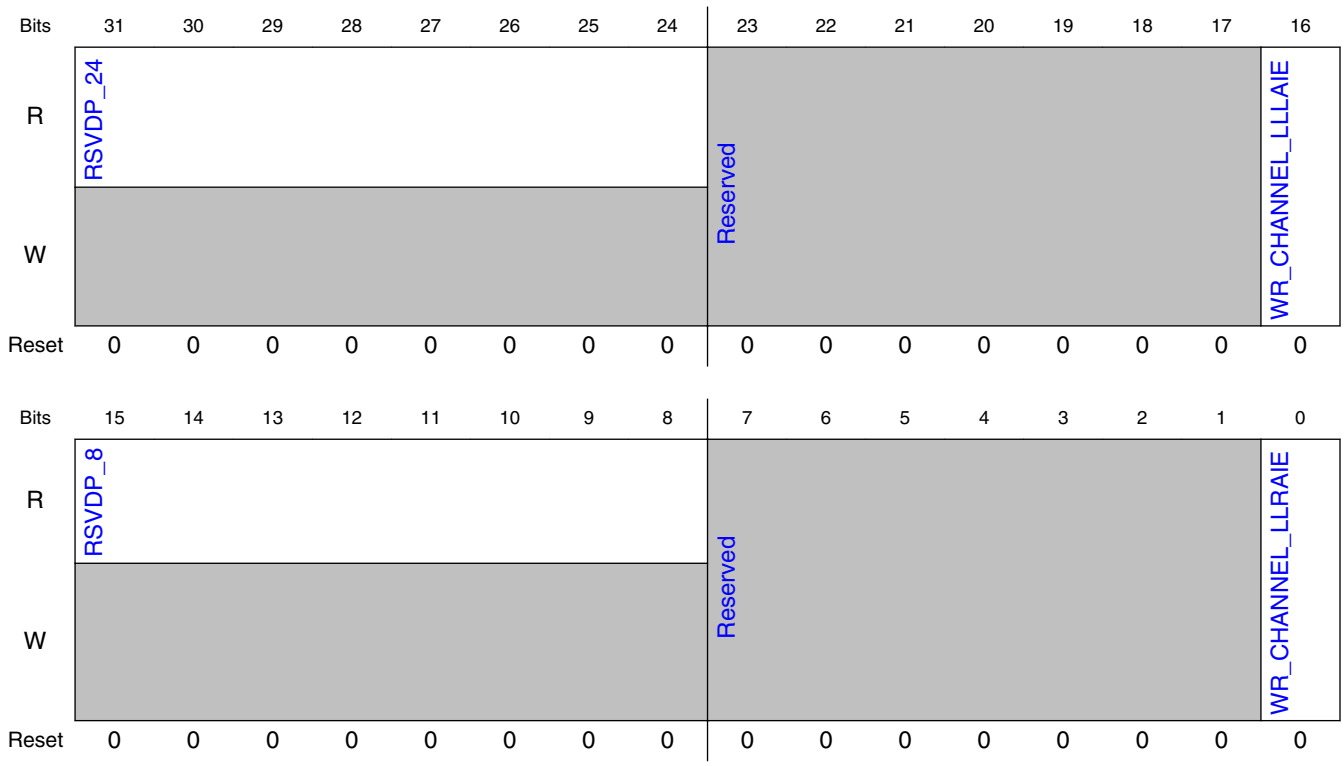
11.3.3.1.205.1 Offset

Register	Offset
DMA_WRITE_LINKED_LIST_ERR_EN_OFF	8008_0090h

11.3.3.1.205.2 Function

DMA Write Linked List Error Enable Register. The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

11.3.3.1.205.3 Diagram



11.3.3.1.205.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved.
16 WR_CHANNEL_LLLAIE	Write Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the write channel local abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W
15-8	Reserved for future use.

Table continues on the next page...

Field	Function
RSVDP_8	
7-1 —	Reserved.
0 WR_CHANNEL_LLRAIE	Write Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the write channel remote abort interrupt through this bit. The LIE and RIE bits in the LL element enable the write channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.206 DMA Read Interrupt Status Register. (DMA_READ_INT_STATUS_OFF)

11.3.3.1.206.1 Offset

Register	Offset
DMA_READ_INT_STATUS_OFF	8008_00A0h

11.3.3.1.206.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24								RD_ABORT_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8								RD_DONE_INT_STATUS							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.206.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 RD_ABORT_INT_STATUS	Abort Interrupt Status. The DMA read channel has detected an error, or you manually stopped the transfer as described in "Stopping the DMA Transfer (Software Stop)". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. You can read the "DMA Read Error Status Low Register" (DMA_READ_ERR_STATUS_LOW_OFF) and "DMA Read Error Status High Register"

Table continues on the next page...

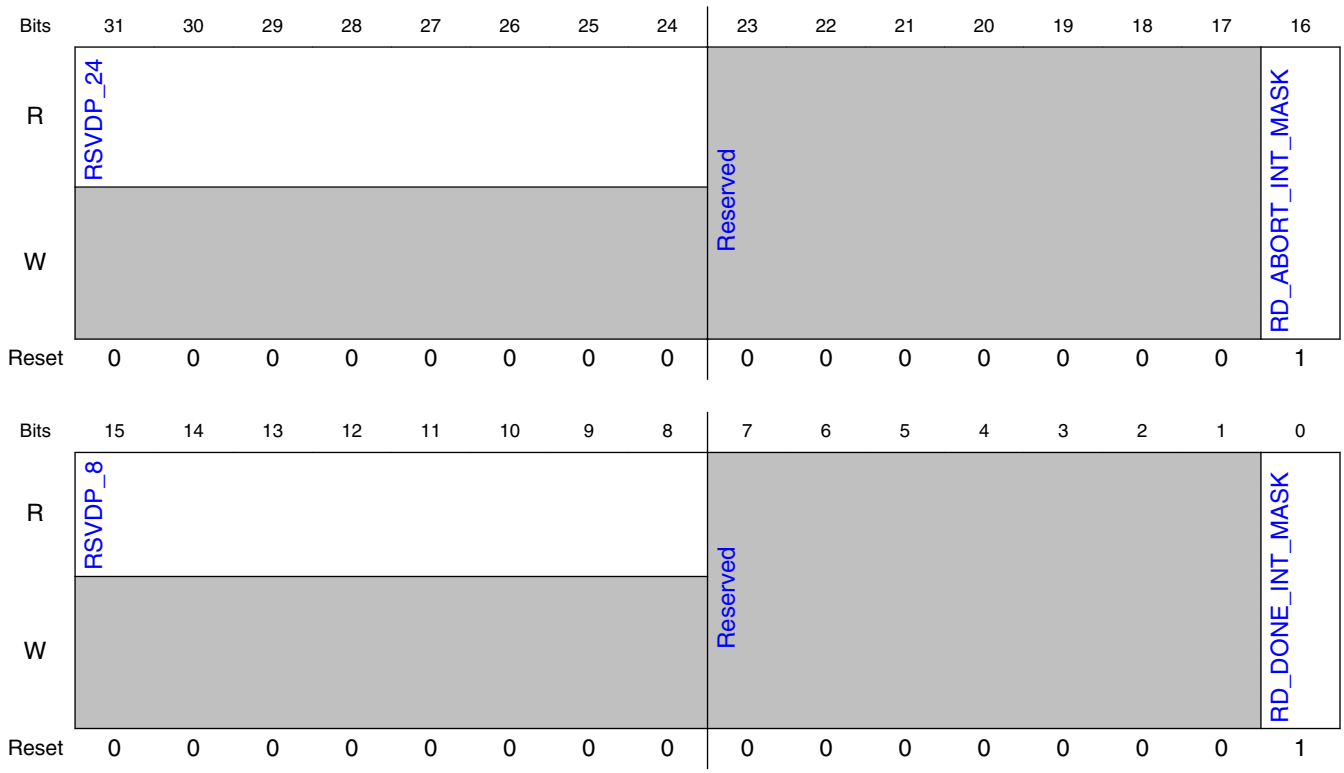
Field	Function
	(DMA_READ_ERR_STATUS_HIGH_OFF) to determine the source of the error. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-0 RD_DONE_INT_STATUS	Done Interrupt Status. The DMA read channel has successfully completed the DMA read transfer. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the DMA read interrupt Clear register to clear this interrupt bit. Note: You can write to this register to emulate interrupt generation, during software or hardware testing. A write to the address triggers an interrupt, but the DMA does not set the Done or Abort bits in this register. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.207 DMA Read Interrupt Mask Register. (DMA_READ_INT_MASK_OFF)

11.3.3.1.207.1 Offset

Register	Offset
DMA_READ_INT_MASK_OFF	8008_00A8h

11.3.3.1.207.2 Diagram



11.3.3.1.207.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved.
16 RD_ABORT_INT_MASK	Abort Interrupt Mask. Prevents the Abort interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved.
0 RD_DONE_INT_MASK	Done Interrupt Mask. Prevents the Done interrupt status field in the DMA read interrupt status register from asserting the edma_int output. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.208 DMA Read Interrupt Clear Register. (DMA_READ_INT_CLEAR_OFF)

11.3.3.1.208.1 Offset

Register	Offset
DMA_READ_INT_CLEAR_OFF	8008_00ACh

11.3.3.1.208.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RSVDP_24															
W									RD_ABORT_INT_CLEAR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RSVDP_8															
W									RD_DONE_INT_CLEAR							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.208.3 Fields

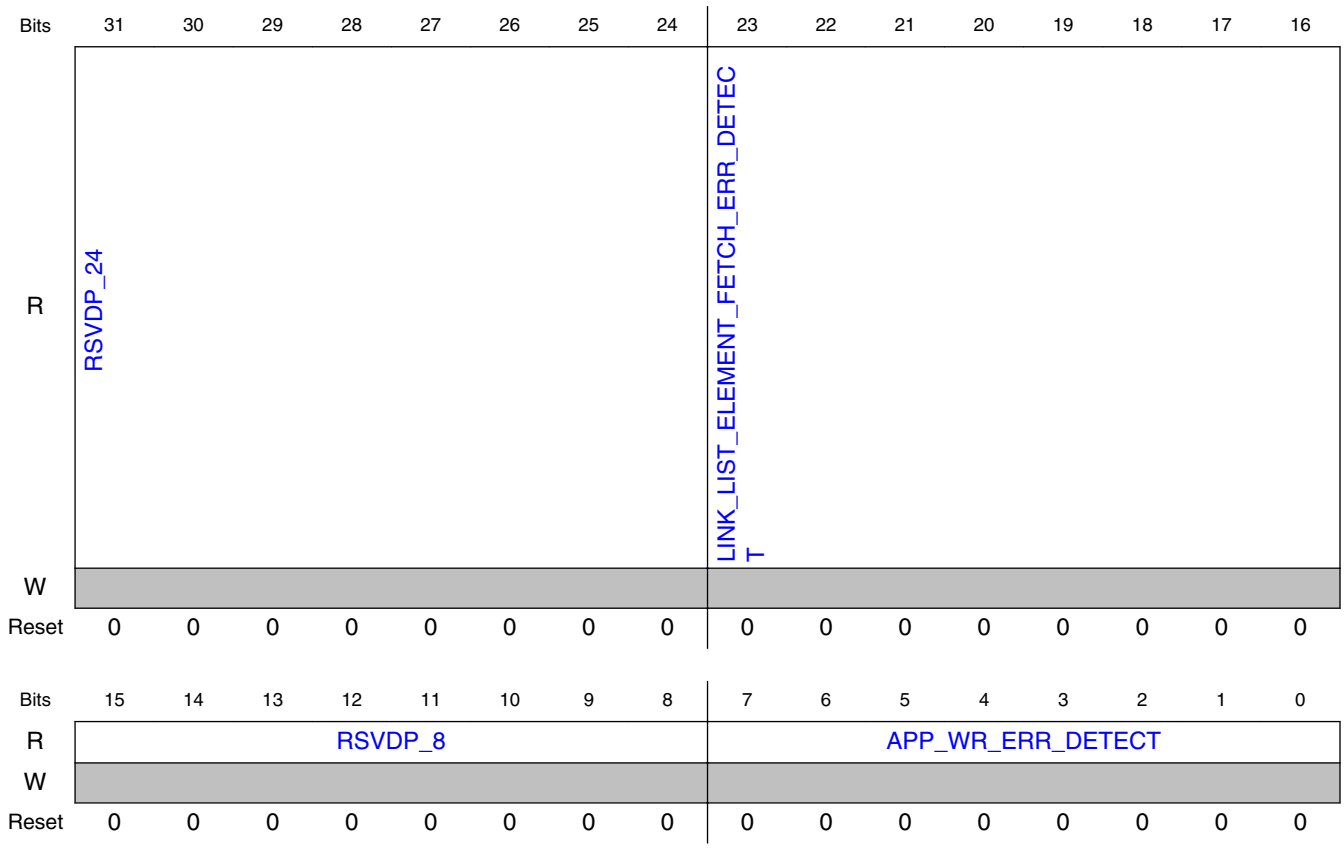
Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 RD_ABORT_INT_CLEAR	Abort Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Abort interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".
15-8 RSVDP_8	Reserved for future use.
7-0 RD_DONE_INT_CLEAR	Done Interrupt Clear. You must write a 1'b1 to clear the corresponding bit in the Done interrupt status field of the DMA read interrupt status register. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Note: Reading from this self-clearing register field always returns a "0".

11.3.3.1.209 DMA Read Error Status Low Register. (DMA_READ_ERR_STATUS_LOW_OFF)

11.3.3.1.209.1 Offset

Register	Offset
DMA_READ_ERR_STATUS_LOW_OFF	8008_00B4h

11.3.3.1.209.2 Diagram



11.3.3.1.209.3 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-16 LINK_LIST_ELEMENT_FETCH_ERR_DETECT	Linked List Element Fetch Error Detected. - The DMA read channel has received an error response from the AXI bus while reading a linked list element from local memory. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).

Table continues on the next page...

Field	Function
15-8 RSVDP_8	Reserved for future use.
7-0 APP_WR_ERR_DETECT	Application Write Error Detected. The DMA read channel has received an error response from the AXI bus (or TRGT1 interface when the AXI Bridge is not used) while writing data to it. This error is fatal. You must restart the transfer from the beginning, as the channel context is corrupted, and the transfer is not rolled back. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this clears all bits in this register, and also the DMA Read Error Status High register (DMA_READ_ERR_STATUS_HIGH_OFF).

11.3.3.1.210 DMA Read Error Status High Register. (DMA_READ_ERR_STATUS_HIGH_OFF)

11.3.3.1.210.1 Offset

Register	Offset
DMA_READ_ERR_STATUS_HIGH_OFF	8008_00B8h

11.3.3.1.210.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DATA_POISIONING								CPL_TIMEOUT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CPL_ABORT								UNSUPPORTED_REQ							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.210.3 Fields

Field	Function
31-24 DATA_POISIONING	Data Poisoning. The DMA read channel has detected data poisoning in the completion from the remote device (in response to the MRd request). The DMA read channel will drop the completion and then be halted. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field

Table continues on the next page...

Field	Function
	of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
23-16 CPL_TIMEOUT	Completion Time Out. The DMA read channel has timed-out while waiting for the remote device to respond to the MRd request, or a malformed CplD has been received. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling" . - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
15-8 CPL_ABORT	Completer Abort. The DMA read channel has received a PCIe completer abort completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.
7-0 UNSUPPORTED_REQ	Unsupported Request. The DMA read channel has received a PCIe unsupported request completion status from the remote device in response to the MRd request. For more details, see "Linked List Mode". Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. - Enabling: For details, see "Interrupts and Error Handling". - Masking: The DMA read interrupt Mask register has no effect on this register. - Clearing: You must write a 1'b1 to the corresponding channel bit in the Abort interrupt field of the "DMA Read Interrupt Clear Register" (DMA_READ_INT_CLEAR_OFF) to clear this error bit. Note, this also clears the other error bits for the same channel in this register and in the DMA Read Error Status Low register.

11.3.3.1.211 DMA Read Linked List Error Enable Register. (DMA_READ_LINKED_LIST_ERR_EN_OFF)

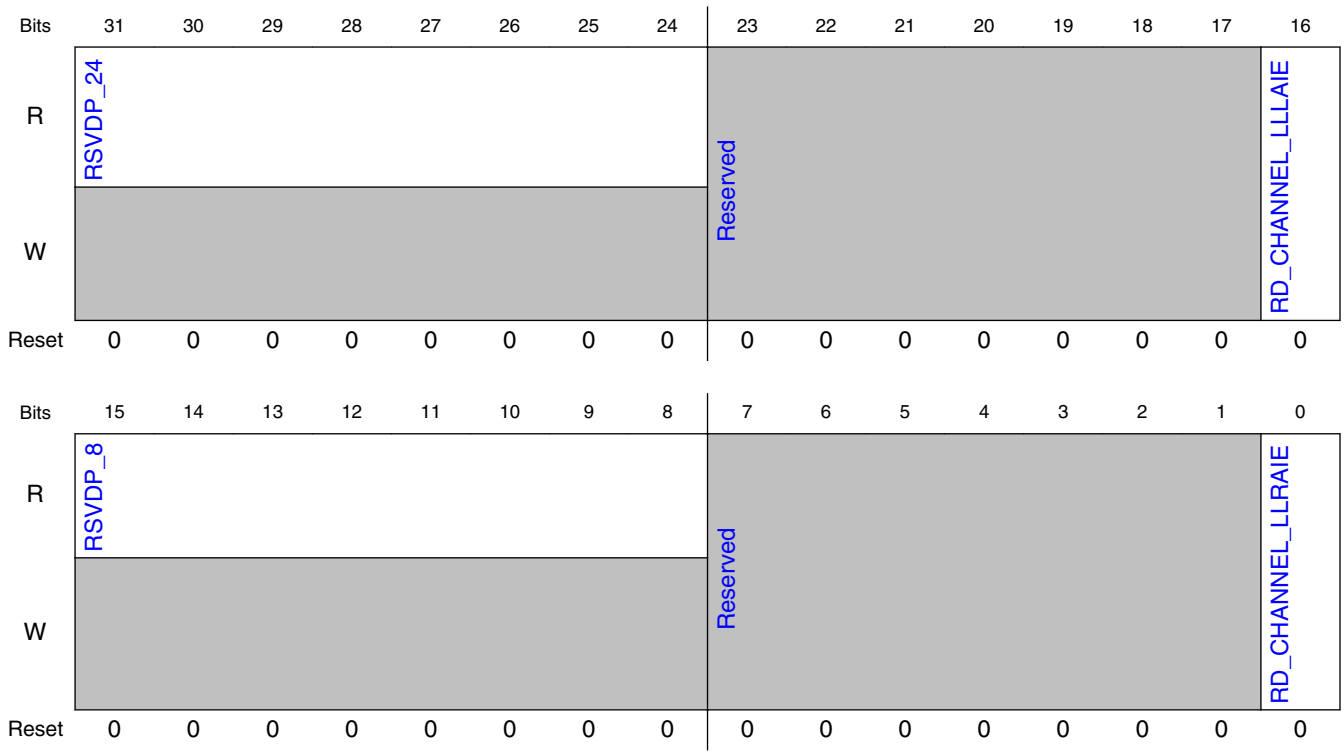
11.3.3.1.211.1 Offset

Register	Offset
DMA_READ_LINKED_LIST_ERR_EN_OFF	8008_00C4h

11.3.3.1.211.2 Function

DMA Read Linked List Error Enable Register. The LIE and RIE bits in the LL element enable the channel "done" interrupts (local and remote). The LLLAIE and LLRAIE bits enable the channel "abort" interrupts (local and remote).

11.3.3.1.211.3 Diagram



11.3.3.1.211.4 Fields

Field	Function
31-24 RSVDP_24	Reserved for future use.
23-17 —	Reserved.
16 RD_CHANNEL_LLLAIE	Read Channel LL Local Abort Interrupt Enable (LLLAIE). You enable the read channel Local Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W
15-8 RSVDP_8	Reserved for future use.
7-1 —	Reserved.
0 RD_CHANNEL_LLRAIE	Read Channel LL Remote Abort Interrupt Enable (LLRAIE). You enable the read channel Remote Abort interrupt through this bit. The LIE and RIE bits in the LL element enable the read channel done interrupts. Each bit corresponds to a DMA channel. Bit [0] corresponds to channel 0. Used in linked list mode only. For more details, see "Interrupt Handling". Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.212 DMA Read Done IMWr Address Low Register. (DMA_READ_DONE_IMWR_LOW_OFF)

11.3.3.1.212.1 Offset

Register	Offset
DMA_READ_DONE_IMWR_LOW_OFF	8008_00CCh

11.3.3.1.212.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_DONE_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.212.3 Fields

Field	Function
31-0 DMA_READ_DONE_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Done IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.213 DMA Read Done IMWr Address High Register. (DMA_READ_DONE_IMWR_HIGH_OFF)

11.3.3.1.213.1 Offset

Register	Offset
DMA_READ_DONE_IMWR_HIGH_OFF	8008_00D0h

11.3.3.1.213.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_DONE_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.213.3 Fields

Field	Function
31-0 DMA_READ_DONE_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Done IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.214 DMA Read Abort IMWr Address Low Register. (DMA_READ_ABORT_IMWR_LOW_OFF)

11.3.3.1.214.1 Offset

Register	Offset
DMA_READ_ABORT_IMWR_LOW_OFF	8008_00D4h

11.3.3.1.214.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_ABORT_LOW_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.214.3 Fields

Field	Function
31-0 DMA_READ_ABORT_LOW_REG	The DMA uses this field to generate bits [31:0] of the address field for the Abort IMWr TLP. Bits [1:0] must be "00" as this address must be dword aligned. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.215 DMA Read Abort IMWr Address High Register. (DMA_READ_ABORT_IMWR_HIGH_OFF)

11.3.3.1.215.1 Offset

Register	Offset
DMA_READ_ABORT_IMWR_HIGH_OFF	8008_00D8h

11.3.3.1.215.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_READ_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_READ_ABORT_HIGH_REG															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.215.3 Fields

Field	Function
31-0 DMA_READ_ABORT_HIGH_REG	The DMA uses this field to generate bits [63:32] of the address field for the Abort IMWr TLP. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.216 DMA Read Channel 1 and 0 IMWr Data Register. (DMA_READ_CH01_IMWR_DATA_OFF)

11.3.3.1.216.1 Offset

Register	Offset
DMA_READ_CH01_IMWR_DATA_OFF	8008_00DCh

11.3.3.1.216.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_1_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_0_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.216.3 Fields

Field	Function
31-16 RD_CHANNEL_1_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 1. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_0_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 0. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.217 DMA Read Channel 3 and 2 IMWr Data Register. (DMA_READ_CH23_IMWR_DATA_OFF)

11.3.3.1.217.1 Offset

Register	Offset
DMA_READ_CH23_IMWR_DATA_OFF	8008_00E0h

11.3.3.1.217.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_3_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_2_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.217.3 Fields

Field	Function
31-16 RD_CHANNEL_3_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 3. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_2_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 2. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.218 DMA Read Channel 5 and 4 IMWr Data Register. (DMA_READ_CH45_IMWR_DATA_OFF)

11.3.3.1.218.1 Offset

Register	Offset
DMA_READ_CH45_IMWR_DATA_OFF	8008_00E4h

11.3.3.1.218.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_5_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_4_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.218.3 Fields

Field	Function
31-16 RD_CHANNEL_5_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 5. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_4_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 4. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.219 DMA Read Channel 7 and 6 IMWr Data Register. (DMA_READ_CH67_IMWR_DATA_OFF)

11.3.3.1.219.1 Offset

Register	Offset
DMA_READ_CH67_IMWR_DATA_OFF	8008_00E8h

11.3.3.1.219.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_CHANNEL_7_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_CHANNEL_6_DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.219.3 Fields

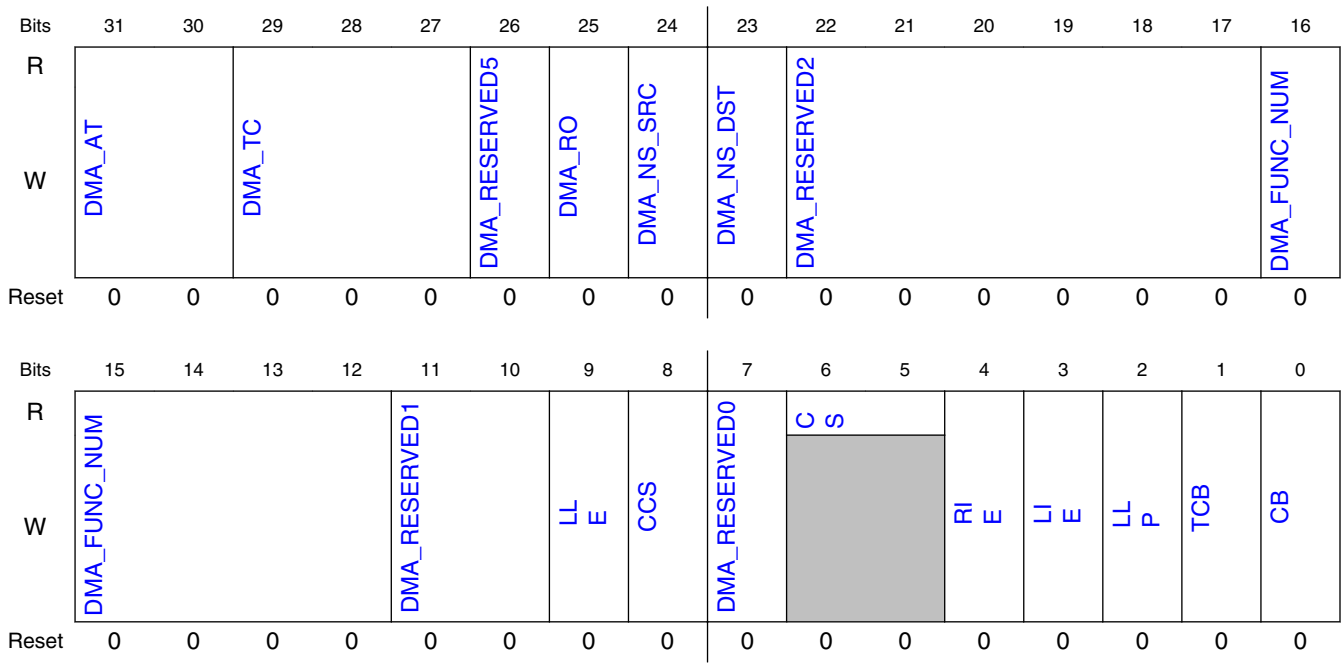
Field	Function
31-16 RD_CHANNEL_7_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 7. Note: The access attributes of this field are as follows: - Dbi: R/W
15-0 RD_CHANNEL_6_DATA	The DMA uses this field to generate the data field for the Done or Abort IMWr TLPs it generates for read channel 6. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.220 DMA Write Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_WRCH_0)

11.3.3.1.220.1 Offset

Register	Offset
DMA_CH_CONTROL1_OFF_WRCH_0	8008_0200h

11.3.3.1.220.2 Diagram



11.3.3.1.220.3 Fields

Field	Function
31-30 DMA_AT	Address Translation TLP Header Bit (AT) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
29-27 DMA_TC	Traffic Class TLP Header Bit (TC) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
26 DMA_RESERVED5	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
25 DMA_RO	Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
24 DMA_NS_SRC	Source No Snoop TLP Header Bit (DMA_NS_SRC). The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
23 DMA_NS_DST	Destination No Snoop TLP Header Bit (DMA_NS_DST). The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
22-17 DMA_RESERVED2	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
16-12	Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the

Table continues on the next page...

Field	Function
DMA_FUNC_NUM	"DMA Write Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_WRCH_0). Note: The access attributes of this field are as follows: - Dbi: R/W
11-10 DMA_RESERVED1	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
9 LLE	Linked List Enable (LLE). - 0: Disable linked list operation - 1: Enable linked list operation Note: The access attributes of this field are as follows: - Dbi: R/W
8 CCS	Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Dbi: R/W
7 DMA_RESERVED0	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
6-5 CS	Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows: - 00: Reserved - 01: Running. This channel is active and transferring data. - 10: Halted. An error condition has been detected, and the DMA has stopped this channel. - 11: Stopped. The DMA has transferred all data for this channel, or you have prematurely stopped this channel by writing to the Stop field of the "DMA Write Doorbell Register" (DMA_WRITE_DOORBELL_OFF) or "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF).
4 RIE	Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
3 LIE	Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
2 LLP	Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W
1 TCB	Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
0 CB	Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.221 DMA Write Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_WRCH_0)

11.3.3.1.221.1 Offset

Register	Offset
DMA_TRANSFER_SIZE_OFF_WRCH_0	8008_0208h

11.3.3.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.221.3 Fields

Field	Function
31-0 DMA_TRANSFER_SIZE	DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA write channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.222 DMA Write SAR Low Register. (DMA_SAR_LOW_OFF_WRCH_0)

11.3.3.1.222.1 Offset

Register	Offset
DMA_SAR_LOW_OFF_WRCH_0	8008_020Ch

11.3.3.1.222.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.222.3 Fields

Field	Function
31-0 SRC_ADDR_REG_LOW	Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Write: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.223 DMA Write SAR High Register. (DMA_SAR_HIGH_OFF_WRCH_0)

11.3.3.1.223.1 Offset

Register	Offset
DMA_SAR_HIGH_OFF_WRCH_0	8008_0210h

11.3.3.1.223.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.223.3 Fields

Field	Function
31-0 SRC_ADDR_REG_HIGH	Source Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.224 DMA Write DAR Low Register. (DMA_DAR_LOW_OFF_WRCH_0)

11.3.3.1.224.1 Offset

Register	Offset
DMA_DAR_LOW_OFF_WRCH_0	8008_0214h

11.3.3.1.224.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.224.3 Fields

Field	Function
31-0 DST_ADDR_REG_LOW	Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Write: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.225 DMA Write DAR High Register. (DMA_DAR_HIGH_OFF_WRCH_0)

11.3.3.1.225.1 Offset

Register	Offset
DMA_DAR_HIGH_OFF_WRCH_0	8008_0218h

11.3.3.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.225.3 Fields

Field	Function
31-0 DST_ADDR_REG_HIGH	Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.226 DMA Write Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_WRCH_0)

11.3.3.1.226.1 Offset

Register	Offset
DMA_LLP_LOW_OFF_WRCH_0	8008_021Ch

11.3.3.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.226.3 Fields

Field	Function
31-0 LLP_LOW	Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.227 DMA Write Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_WRCH_0)

11.3.3.1.227.1 Offset

Register	Offset
DMA_LLP_HIGH_OFF_WRCH_0	8008_0220h

11.3.3.1.227.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.227.3 Fields

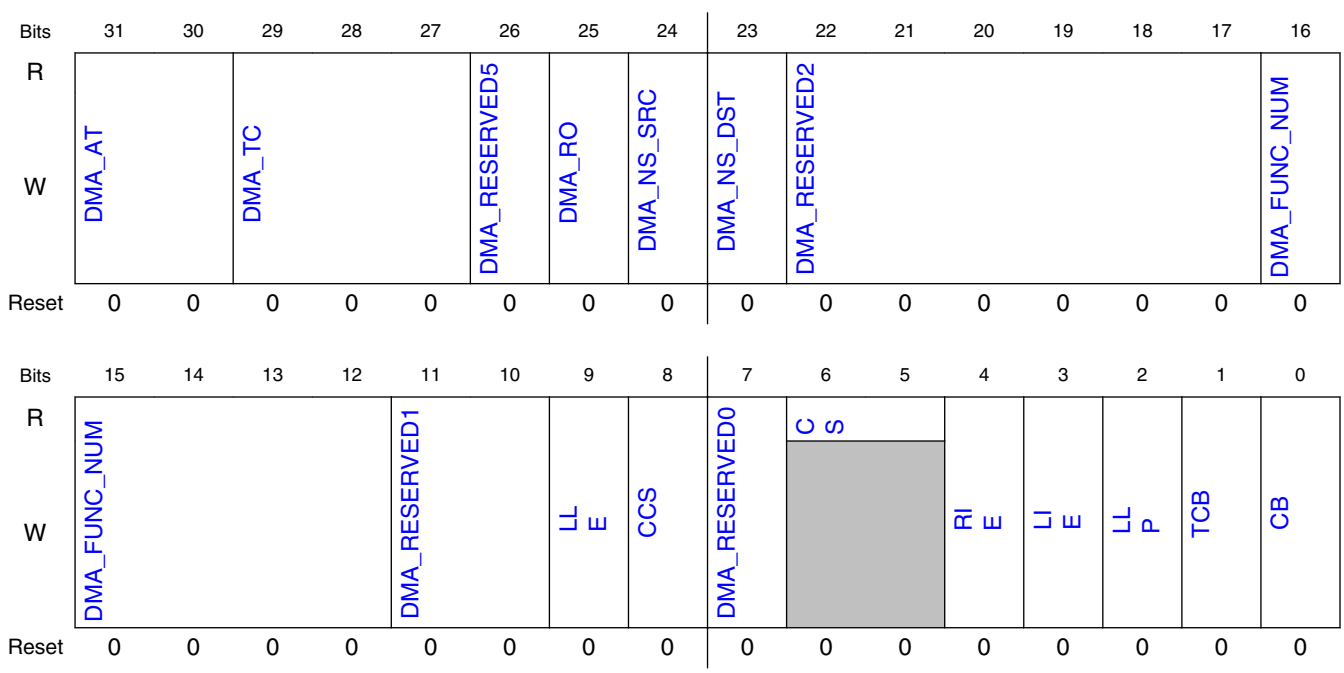
Field	Function
31-0 LLP_HIGH	Upper 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.228 DMA Read Channel Control 1 Register. (DMA_CH_CONTROL1_OFF_RDCH_0)

11.3.3.1.228.1 Offset

Register	Offset
DMA_CH_CONTROL1_OFF_RDCH_0	8008_0300h

11.3.3.1.228.2 Diagram



11.3.3.1.228.3 Fields

Field	Function
31-30 DMA_AT	Address Translation TLP Header Bit (AT) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
29-27 DMA_TC	Traffic Class TLP Header Bit (TC) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
26 DMA_RESERVED5	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
25 DMA_RO	Relaxed Ordering TLP Header Bit (RO) The DMA uses this TLP header field when generating MRd/MWr (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
24 DMA_NS_SRC	Source No Snoop TLP Header Bit (DMA_NS_SRC). The DMA uses this TLP header field when generating MRd (SAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
23 DMA_NS_DST	Destination No Snoop TLP Header Bit (DMA_NS_DST). The DMA uses this TLP header field when generating MWr (DAR addressing space) (not IMWr) TLPs. Note: The access attributes of this field are as follows: - Dbi: R/W
22-17 DMA_RESERVED2	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
16-12 DMA_FUNC_NUM	Function Number (FN). The controller uses this field when generating the requester ID for the MRd/MWr DMA TLP. When you have enabled SR-IOV, then this field is ignored if you have set the VFE field in the "DMA Read Channel Control 2 Register" (DMA_CH_CONTROL2_OFF_RDCH_0). Note: The access attributes of this field are as follows: - Dbi: R/W
11-10 DMA_RESERVED1	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
9 LLE	Linked List Enable (LLE). - 0: Disable linked list operation - 1: Enable linked list operation Note: The access attributes of this field are as follows: - Dbi: R/W
8 CCS	Consumer Cycle State (CCS). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". You must initialize this bit. The DMA updates this bit during linked list operation. Note: The access attributes of this field are as follows: - Dbi: R/W
7 DMA_RESERVED0	Reserved. Note: The access attributes of this field are as follows: - Dbi: R/W
6-5 CS	Channel Status (CS). The channel status bits identify the current operational state of the DMA channel. The operation state encoding for each DMA channel is a s follows: - 00: Reserved - 01: Running. This channel is active and transferring data. - 10: Halted. An error condition has been detected, and the DMA has stopped this channel. - 11: Stopped. The DMA has transferred all data for this channel, or you have prematurely stopped this channel by writing to the Stop field of the "DMA Read Doorbell Register" (DMA_WRITE_DOORBELL_OFF) or "DMA Read Doorbell Register" (DMA_READ_DOORBELL_OFF).
4 RIE	Remote Interrupt Enable (RIE). You must set this bit to enable the generation of the Done or Abort Remote interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the RIE of the LL element. The RIE bit in a LL element only enables the Done interrupt. In non-LL mode, the RIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

Table continues on the next page...

Field	Function
3 LIE	Local Interrupt Enable (LIE). You must set this bit to enable the generation of the Done or Abort Local interrupts. For more details, see "Interrupts and Error Handling". In LL mode, the DMA overwrites this with the LIE of the LL element. The LIE bit in a LL element only enables the Done interrupt. In non-LL mode, the LIE bit enables the Done and Abort interrupts. This field is not defined in a link LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
2 LLP	Load Link Pointer (LLP). Used in linked list mode only. Indicates that this linked list element is a link element, and its LL element pointer dwords are pointing to the next (non-contiguous) element. The DMA loads this field with the LLP of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W
1 TCB	Toggle Cycle Bit (TCB). Indicates to the DMA to toggle its interpretation of the CB. Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the TCB of the linked list element. this field is not defined in a data LL element. Note: The access attributes of this field are as follows: - Dbi: R/W
0 CB	Cycle Bit (CB). Used in linked list mode only. It is used to synchronize the producer (software) and the consumer (DMA). For more details, see "PCS-CCS-CB-TCB Producer-Consumer Synchronization". The DMA loads this field with the CB of the linked list element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.229 DMA Read Transfer Size Register. (DMA_TRANSFER_SIZE_OFF_RDCH_0)

11.3.3.1.229.1 Offset

Register	Offset
DMA_TRANSFER_SIZE_OFF_RDCH_0	8008_0308h

11.3.3.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_TRANSFER_SIZE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.229.3 Fields

Field	Function
31-0 DMA_TRANSFER_SIZE	DMA Transfer Size. You program this register with the size of the DMA transfer. The maximum DMA transfer size is 4Gbytes. The minimum transfer size is one byte (0x1). This field is automatically decremented by the DMA as the DMA read channel transfer progresses. This field indicates the number bytes remaining to be transferred. When all bytes are successfully transferred the current transfer size is zero. In LL mode, the DMA overwrites this register with the corresponding dword of the LL element. You can read this register to monitor the transfer progress, however in some scenarios this register is updated after a delay. For example, when less than 3 channels are doorbelled, this register is updated only after a descriptor finishes(linked list mode), or the transfer ends (non-linked list mode). Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.230 DMA Read SAR Low Register. (DMA_SAR_LOW_OFF_RDCH_0)

11.3.3.1.230.1 Offset

Register	Offset
DMA_SAR_LOW_OFF_RDCH_0	8008_030Ch

11.3.3.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.230.3 Fields

Field	Function
31-0 SRC_ADDR_REG_LOW	Source Address Register (Lower 32 bits). Indicates the next address to be read from. The DMA increments the SAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The SAR is the address of the remote memory. - DMA Read: The SAR is the address of the local memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.231 DMA Read SAR High Register. (DMA_SAR_HIGH_OFF_RDCH_0)

11.3.3.1.231.1 Offset

Register	Offset
DMA_SAR_HIGH_OFF_RDCH_0	8008_0310h

11.3.3.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SRC_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.231.3 Fields

Field	Function
31-0 SRC_ADDR_REG_HIGH	Source Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.232 DMA Read DAR Low Register. (DMA_DAR_LOW_OFF_RDCH_0)

11.3.3.1.232.1 Offset

Register	Offset
DMA_DAR_LOW_OFF_RDCH_0	8008_0314h

11.3.3.1.232.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.232.3 Fields

Field	Function
31-0 DST_ADDR_REG_LOW	Destination Address Register (Lower 32 bits). Indicates the next address to be written to. The DMA increments the DAR as the DMA transfer progresses. In LL mode, the DMA overwrites this with the corresponding dword of the LL element. - DMA Read: The DAR is the address of the local memory. - DMA Read: The DAR is the address of the remote memory. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.233 DMA Read DAR High Register. (DMA_DAR_HIGH_OFF_RDCH_0)

11.3.3.1.233.1 Offset

Register	Offset
DMA_DAR_HIGH_OFF_RDCH_0	8008_0318h

11.3.3.1.233.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DST_ADDR_REG_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.233.3 Fields

Field	Function
31-0 DST_ADDR_REG_HIGH	Destination Address Register (Higher 32 bits). In LL mode, the DMA overwrites this with the corresponding dword of the LL element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.234 DMA Read Linked List Pointer Low Register. (DMA_LLP_LOW_OFF_RDCH_0)

11.3.3.1.234.1 Offset

Register	Offset
DMA_LLP_LOW_OFF_RDCH_0	8008_031Ch

11.3.3.1.234.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_LOW															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.234.3 Fields

Field	Function
31-0 LLP_LOW	Lower bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list after the previous element is consumed. - When the current element is a data element; this field is incremented by 6. - When the current element is a link element; this field is overwritten by the LL Element Pointer of the element. Note: The access attributes of this field are as follows: - Dbi: R/W

11.3.3.1.235 DMA Read Linked List Pointer High Register. (DMA_LLP_HIGH_OFF_RDCH_0)

11.3.3.1.235.1 Offset

Register	Offset
DMA_LLP_HIGH_OFF_RDCH_0	8008_0320h

11.3.3.1.235.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	LLP_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

11.3.3.1.235.3 Fields

Field	Function
31-0 LLP_HIGH	Upper 32 bits of the address of the linked list transfer list in local memory. Used in linked list mode only. Updated by the DMA to point to the next element in the transfer list as elements are consumed. Note: The access attributes of this field are as follows: - Dbi: R/W

11.4 PCI Express PHY (PCIe_PHY)

11.4.1 Overview

This block provides information regarding PCIe PHY and its features. PCIe PHY supports 6.0 Gbps data rate and complies to PCI Express base specification 2.1. The functions that are performed by the transceiver include serializing the 8B/10B encoded data for transmission, de-serializing received code groups, and word alignment.

When transmitting, the transceiver accepts two or four 10-bit 8B/10B encoded transmit characters, latches them and serializes the data onto the PCIE_TX_P/PCIE_TX_N differential outputs at 1.5 / 2.5 / 3.0 / 5.0 / 6.0 Gbps. It also performs 8B/10B encoding for 8-bit data from the PIPE interface.

When receiving, the transceiver also samples received serial data on the PCIE_RX_P / PCIE_RX_N differential inputs, deserializes it into two or four 10-bit received characters and detects the K28.5 character (0011111010 or 1100000101) for word alignment. It also applies 8B/10B decoding for 8-bit data to the PIPE interface. PCIe PHY core contains on-chip PLL circuitry for synthesis of the baud-rate transmitting clocks, and extraction of the retimed clocks from the received serial stream.

11.4.1.1 Block Diagram

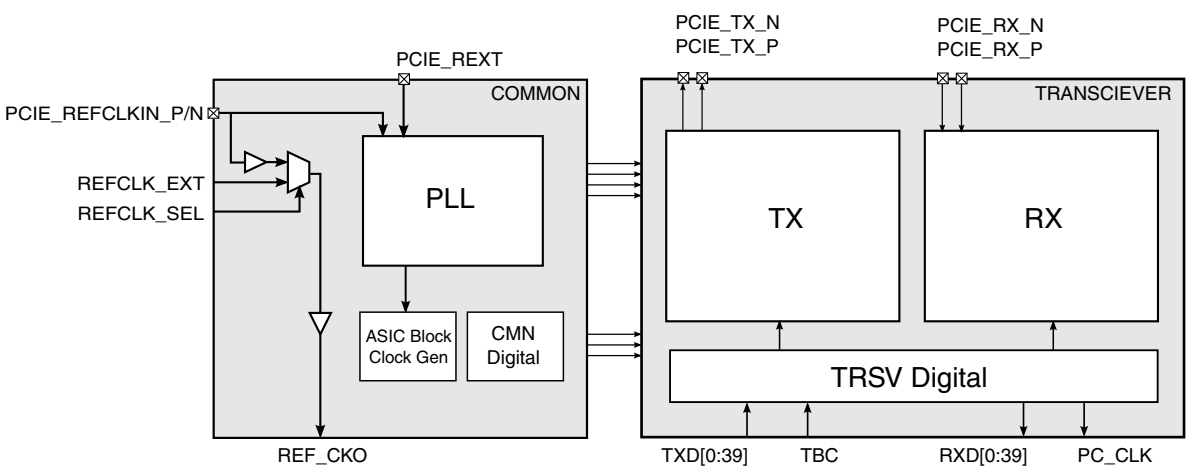


Figure 11-73. Block Diagram

11.4.1.2 Features

The following list the key features of the PCIe PHY:

- 1.5 / 2.5 / 3.0 / 5.0 / 6.0 Gbps Serializer / Deserializer
- Compliant with PCI Express Base Specification 2.1
- Compliant with PIPE Specification 2.0
- 8 / 16 / 20 / 40-bit CMOS Interface for Transmitter and Receiver
- 25 / 100 MHz Reference Clock
- K28.5 Detection for Word Alignment
- 8B/10B Encoding / Decoding
- Receiver Detection
- 28 nm CMOS Process (LN28LPP)
- Supports Spread Spectrum Clocking in Transmitter and Receiver

11.4.2 PCIe PHY Signals

11.4.2.1 Interface Signals

This section describes the interface signals to PCIe PHY.

Table 11-42. Interface Signals Description

Signal	Description
REF_CKO	Buffered Reference Clock Output. REF_CKO is 25 or 100 MHz according to the reference clock input frequency.
REFCLK_SEL	Internal Reference Clock Enable. When driven high, REFCLK_EXT is the reference clock of PHY.
REFCLK_EXT	Internal Reference Clock Input. REFCLK_EXT is the reference clock from logic. The nominal frequency is 25 / 100 MHz.
HIGH_SPEED	Maximum Speed Selection. When driven high, the maximum data rate is Gen2 PCIe. When driven low, the maximum data rate Gen1 PCIe.
TXD	*Transmit Data*
RXD	*Receive Data*

Table continues on the next page...

Table 11-42. Interface Signals Description (continued)

Signal	Description
TBC	*PIPE Interface Input Data Clock* TBC is the transmit byte clock used to latch all input data on the PIPE interface. Transitions of TXD and other inputs occur at rising edge of TBC. TBC is 250 MHz both 2.5 Gbps and 5.0 Gbps, when HIGH_SPEED is high. TBC is required to be frequency locked to reference clock from PCIE_REFCLKIN_P/N or REFCLK_EXT. The exact timing relationship between TBC and reference clock is not tightly specified, however the TBC should be derived from the PC_CLK (PCLK) clock.
RBC	*Receive Byte Clock.* RBC is locked to the received data. RBC is 75 MHz at 1.5 Gbps, 150 MHz at 3.0 Gbps, and 300 MHz at 6.0 Gbps. The parallel output data are valid on the rising edge of RBC.
PC_CLK	*PIPE Interface Output Data Clock.* All output data movement across the parallel interface is synchronized to this clock. This clock operates at 250 MHz regardless of the Rate control input. The rising edge of the clock is the reference for all output signals. Spread spectrum modulation on this clock is allowed.

11.4.2.2 Reference Clock

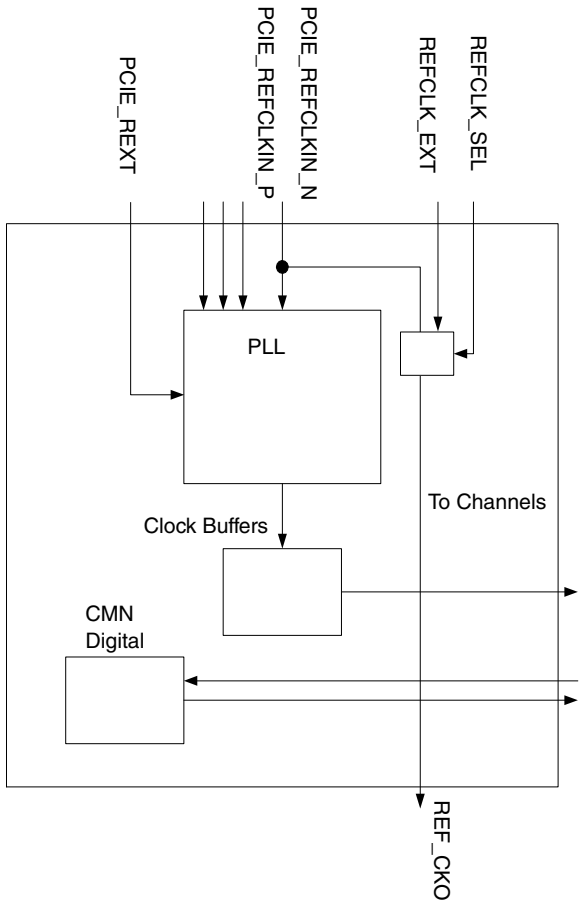
Reference Clock input jitter should be tightly controlled to ensure very high-speed clock operation and its jitter characteristics. Jitter performance of the PCIe PHY is dependent on a low-jitter reference clock source. It can use Crystal Oscillator input, PCIE_REFCLKIN_P/N, or internal clock input, REFCLK_EXT as a Reference Clock input. For more detail, please refer to the datasheet. When the REFCLK_EXT is used, the external IO pin of PCIE_REFCLKIN_P/N cannot be floating and proper powerdown control should be considered.

For the reference clock differential output for the PCIe PHY (PCIE_REFCLKOUT_P and PCIE_REFCLKOUT_N), the external resistors should be connected to ground in the PCB board. It is open drain structure.

11.4.3 Functional Description

11.4.3.1 Common Block

Figure 11-74. Common Block Diagram

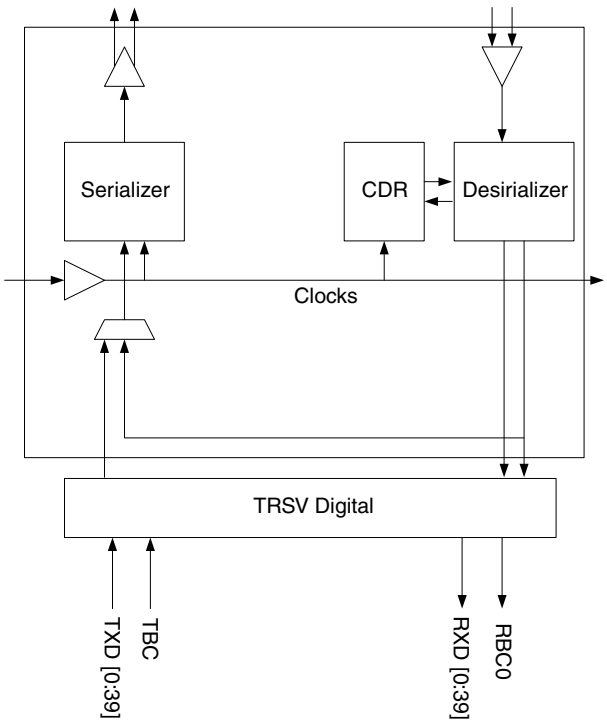


11.4.3.1.1 Phase Locked Loop (PLL)

In the PCIe PHY core, the PLL is locked at the multiple frequency of reference clock (PCIE_REFCLKIN_P/N) and generates all internal multi-phase clocks for transceiver. These clocks are based on the supplied PCIE_REFCLKIN_P/N.

11.4.3.2 Transceiver block

Figure 11-75. Transceiver Block Diagram



11.4.3.2.1 Serializer

The PCIe PHY core accepts two or four 10-bit parallel characters from the link layer on the TXD[0:19] or TXD[0:39] bus which are latched into the input latch. This data will be serialized and transmitted on the PCIE_TX_P / PCIE_TX_N differential outputs at 1.5 / 3.0 / 6.0 Gbps, with bit TXD [0] transmitted first.

11.4.3.2.2 Clock and Data Recovery (CDR)

The PCIe PHY core accepts high-speed differential serial inputs on the PCIE_RX_P / PCIE_RX_N pins, extracts the retimed clocks and recovers the data. The serial bit stream should be encoded so as to provide DC balance and limited run length using 8B/10B encoding. The PCIe PHY core clock recovery circuitry is completely monolithic and requires no external components. For proper operation, the baud rate of the data stream to be recovered should meet the PCIe specification.

The retimed serial bit stream is converted into two or four 10-bit parallel output characters on the RXD[0:19] or RXD[0:39] of which the RXD[0] is received first. The PCIe PHY core provides retimed clock, RBC. The RBC is generated from one of the high-speed clocks which is phase locked to the reference clock. The serial data is retimed by the internal high-speed clocks, and deserialized.

Even if serial input data is not present, or does not meet the required baud rate, the PCIe PHY core will continue to produce retimed clock, RBC, which locks to the REF_CK, reference clock.

11.4.3.2.3 Word Alignment

The PCIe PHY core provides K28.5 character recognition and data word alignment. When synchronization is enabled, the PCIe PHY core constantly examines the serial data for the presence of the K28.5 character (0011111010 / 1100000101).

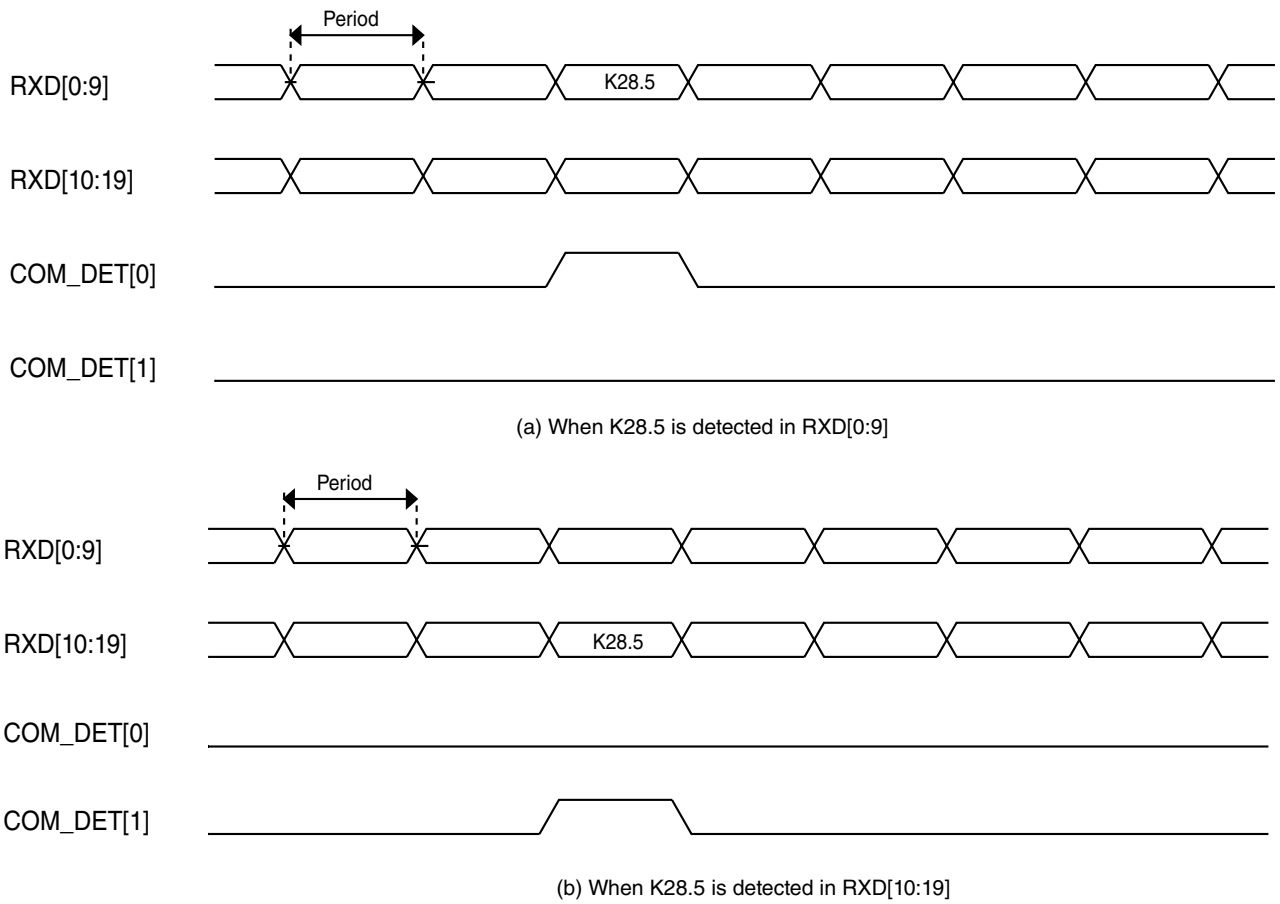


Figure 11-76. Detection of a K28.5 Character

11.4.3.2.4 Driver

Driver amplifies the level of received signal from Serializer to the level that is enough to drive the transmission line. The launch amplitude can be controlled by DRV_LVL [2:0]. Driver features the pre-emphasis that is effective to reduce ISI jitter caused by the loss of transmission line. The pre-emphasis level can be controlled by EMP_LVL [3:0].

11.4.3.3 Start-up sequence

To start PCIe PHY operation, RESET (G_RST) should be invoked at the power up. G_RST should be pulsed after the power-up following by REFCLK input and setting the asynchronous control inputs. The PHY powerdown and up sequences could be added after G_RST sequence for the initialization. The timing restrictions for RESET signal and the typical timing information for the related output signals are shown in the following figure.

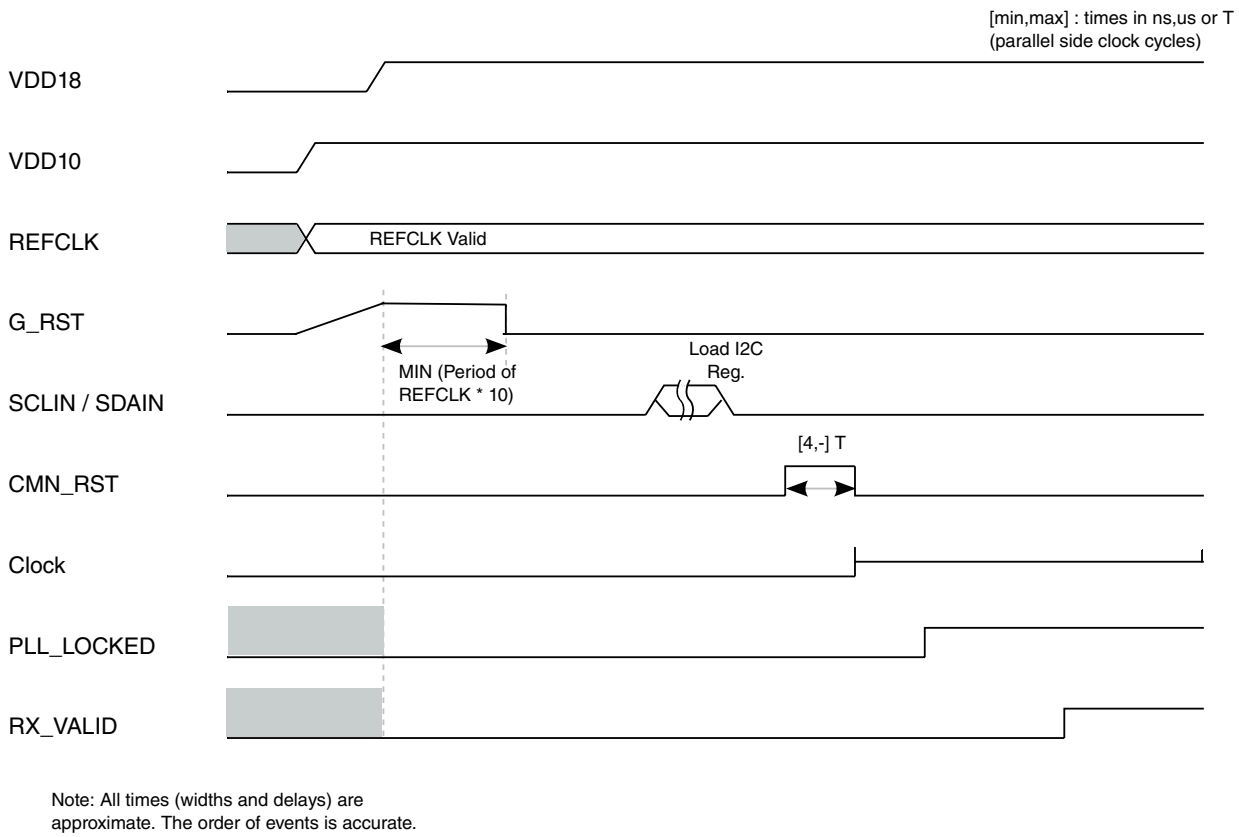


Figure 11-77. Start-up Sequence

NOTE

Before the PHY initialization sequences, all the input signals including REFCLK and power / ground for the PHY should be active and stable. After the PHY initialization sequences (for example, after RX_VALID = 1), the Link controller initialization including reset could be started (for example, PHY signals to the LINK controller must be neglected before the PHY initialization sequences is fully completed.

11.4.3.4 Interface power states

The interface power states are defined as described in the following table. For the power states such as Partial or Slumber, greater power savings can be achieved at a cost of increased exit latency.

Table 11-43. Interface Power States

State	Description
PHYRDY (or L0)	The Phy logic and main PLL are both on and active. The interface is synchronized and capable of receiving and sending data.
Partial (or L1)	The Phy logic is powered, but is in a reduced power state. This state allows an additional power savings over normal mode at the cost of additional resume latency. The exit latency from this state shall be no longer than 10 µs (Usually less than 2 µs in Samsung PCIe PHY). However, when Automatic Partial to Slumber Transitions are enabled the exit latency from this state shall be no longer than the maximum Slumber exit latency.
Slumber (or L2)	The Phy logic is powered but is in a reduced power state. Power can be aggressively conserved in this mode. Most of the Transmitter and Receiver may be shut off. Main power and clocks are not guaranteed, but REF_CLK is available. The exit latency from this state shall be no longer than 10 ms (Usually less than 500 µs in PCIe PHY).

NOTE

For the PCIe interface, recommended L0s Exit Latency and L1 Exit Latency is less than 64 ns and between 1 µs and 2 µs respectively. L0s state in the PCIe application is intended as a power savings state between L0 and L1.

11.4.3.5 Multi-channel support

- When the PCIe PHY is implemented to support four channels. Register map demonstrates how this 4-channel PCIe PHY is configured. One CMU block includes the main PLL, bias generator, termination calibration, and the reference clock I/O buffer. 4-channel TRSV block contains the full TX and RX parts including serializer, deserializer, clock and data recovery and high-speed I/Os. This SoC only support one channel phy.
- For the logic sides, interface signal ports are provided for each channel and named with channel number such as "*_CH[n]" (For example, *_CH0, *_CH1, *_CH2, and *_CH3).

- VDD10 power has to turn on first and then VDD18 power turns on. It is needed to protect damage of thin transistor from unknown situation when 1.8 V power supply only applied to the PHY.
- VDD18 turn on time has to wait at least 0.1 μs after VDD10 turns on. But if ramp up time of VDD10 is much slower than 0.1 μs, waiting time also has to be increased to meet following figure.

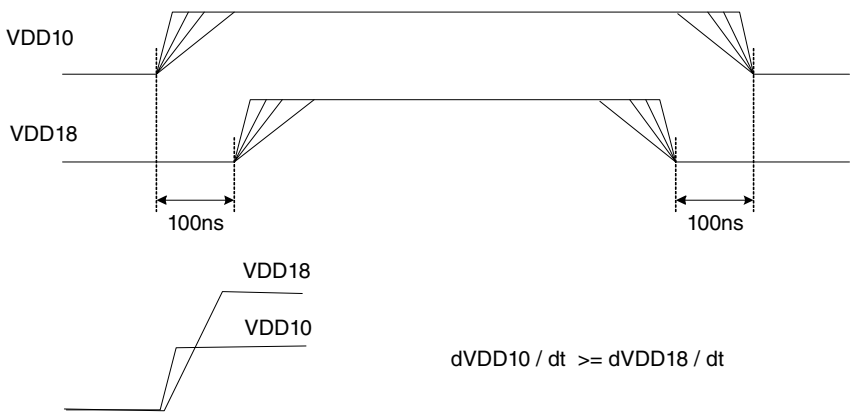


Figure 11-78. Power Up Sequence

11.4.4 PCIe PHY Register

The PCIe PHY has programmable test features that are controlled by registers. These internal registers can be accessible through the APB or I2C interface. The PHY comprises CMU part and TRSV part. Only one CMU block exists for the PHY, and one or more TRSV blocks can be implemented according to the system requirements.

This phy only has channel 0 (CH0). Address is word address, SOC Address = Base Address + 4 * (word address). For each word only byte0 is useful, byte3 ~ 1 are all read zero.

11.4.4.1 PHY Register for CMN Block

The PCIe PHY share a common (CMN) block for the clock and bias generation. The following tables list the capabilities implemented by the PHY design team for internal use.

PCIE_PHY_CMN memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
4	PCIE_PHY_CMN_REG01	32	R/W	0000_0000h	11.4.4.1.1/ 3441
8	PCIE_PHY_CMN_REG02	32	R/W	0000_0014h	11.4.4.1.2/ 3442
12	PCIE_PHY_CMN_REG03	32	R/W	0000_00E1h	11.4.4.1.3/ 3443
16	PCIE_PHY_CMN_REG04	32	R/W	0000_0035h	11.4.4.1.4/ 3444
20	PCIE_PHY_CMN_REG05	32	R/W	0000_0029h	11.4.4.1.5/ 3445
24	PCIE_PHY_CMN_REG06	32	R/W	0000_0083h	11.4.4.1.6/ 3446
28	PCIE_PHY_CMN_REG07	32	R/W	0000_003Ch	11.4.4.1.7/ 3447
2C	PCIE_PHY_CMN_REG0B	32	R/W	0000_003Fh	11.4.4.1.8/ 3448
32	PCIE_PHY_CMN_REG08	32	R/W	0000_005Eh	11.4.4.1.9/ 3449
36	PCIE_PHY_CMN_REG09	32	R/W	0000_0013h	11.4.4.1.10/ 3450
40	PCIE_PHY_CMN_REG11	32	R/W	0000_00FCh	11.4.4.1.11/ 3451
60	PCIE_PHY_CMN_REG15	32	R/W	0000_0004h	11.4.4.1.12/ 3452
64	PCIE_PHY_CMN_REG16	32	R/W	0000_0001h	11.4.4.1.13/ 3453
68	PCIE_PHY_CMN_REG17	32	R/W	0000_0001h	11.4.4.1.14/ 3454
72	PCIE_PHY_CMN_REG18	32	R/W	0000_0000h	11.4.4.1.15/ 3455
76	PCIE_PHY_CMN_REG19	32	R/W	0000_0043h	11.4.4.1.16/ 3456
80	PCIE_PHY_CMN_REG1A	32	R/W	0000_0000h	11.4.4.1.17/ 3457

11.4.4.1.1 Impedance Calibration Register (PCIE_PHY_CMN_REG01)

NOTE

When FORCE pin is set to "1", TX and RX impedance can be controlled by manual code setting of TCODE[3:0] and RCODE[3:0] respectively. It is not recommend using this feature in normal case and used for debug only.

Address: 0h base + 4h offset = 4h

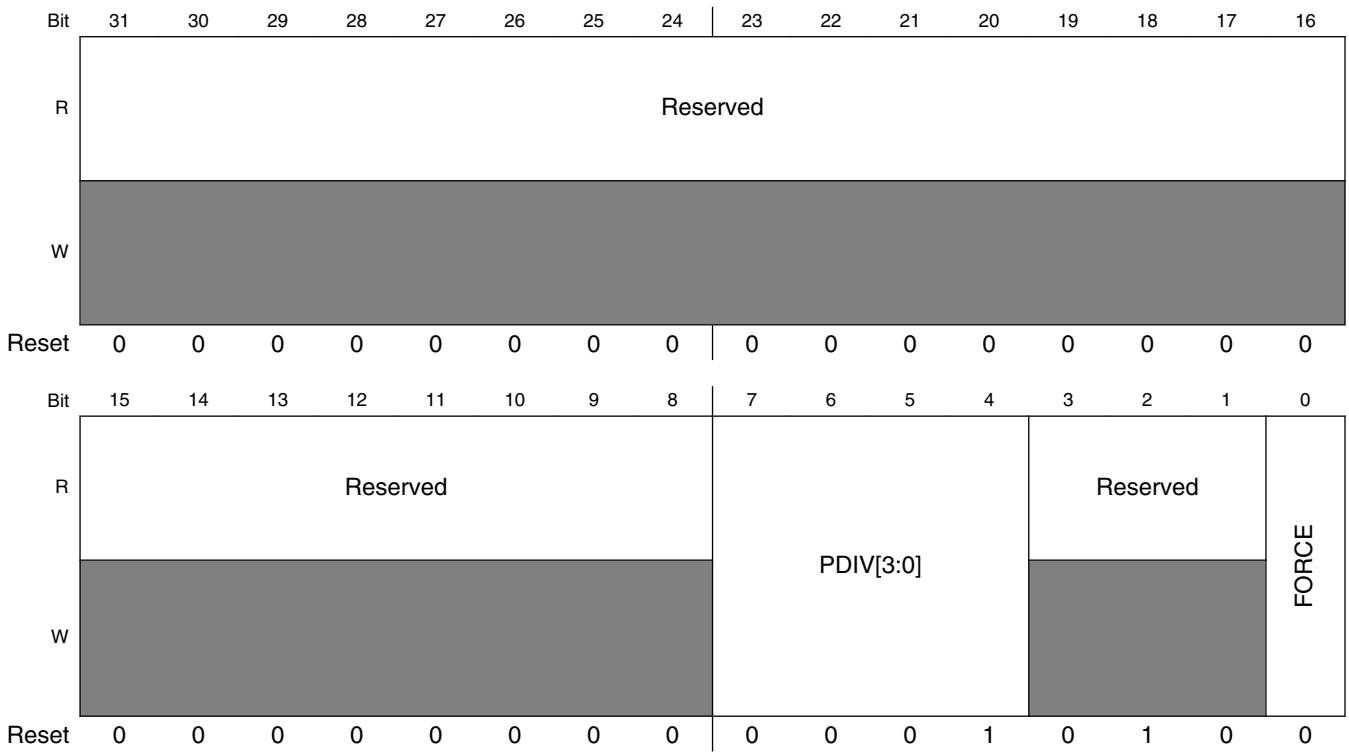
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																								RCODE[3:0]				TCODE[3:0]			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_CMN_REG01 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 RCODE[3:0]	Manual RX Impedance Control 0000 Large 1111 Small
TCODE[3:0]	Manual TX Impedance Control 0000 Large 1111 Small

11.4.4.1.2 PCIE_PHY_CMN_REG02

Address: 0h base + 8h offset = 8h



PCIE_PHY_CMN_REG02 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 PDIV[3:0]	PLL Pre-divider Ratio Control 0000 GND 0001 1/1 (Default) 0010 1/2 0011 1/3 0100 1/4 0101 1/5 0110 1/6 0111 1/7 1000 1/8 1001 1/9 1010 1/10 1011 1/11 ~ 1/11 1111 1/11
3–1 Reserved	This field is reserved.

Table continues on the next page...

PCIE_PHY_CMN_REG02 field descriptions (continued)

Field	Description
0 FORCE	Manual Impedance Control Enable When FORCE pin is set to '1', TX and RX impedance can be controlled by manual code setting of TCODE[3:0] and RCODE[3:0] respectively. It is recommend using this feature in normal case and used for debug only. 0 Auto Calibration (Default Value) 1 Manual Impedance Control Enable

11.4.4.1.3 PCIE_PHY_CMN_REG03

This register is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission.

Address: 0h base + 12h offset = 12h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																												CTRL_CP[3:0]			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	1

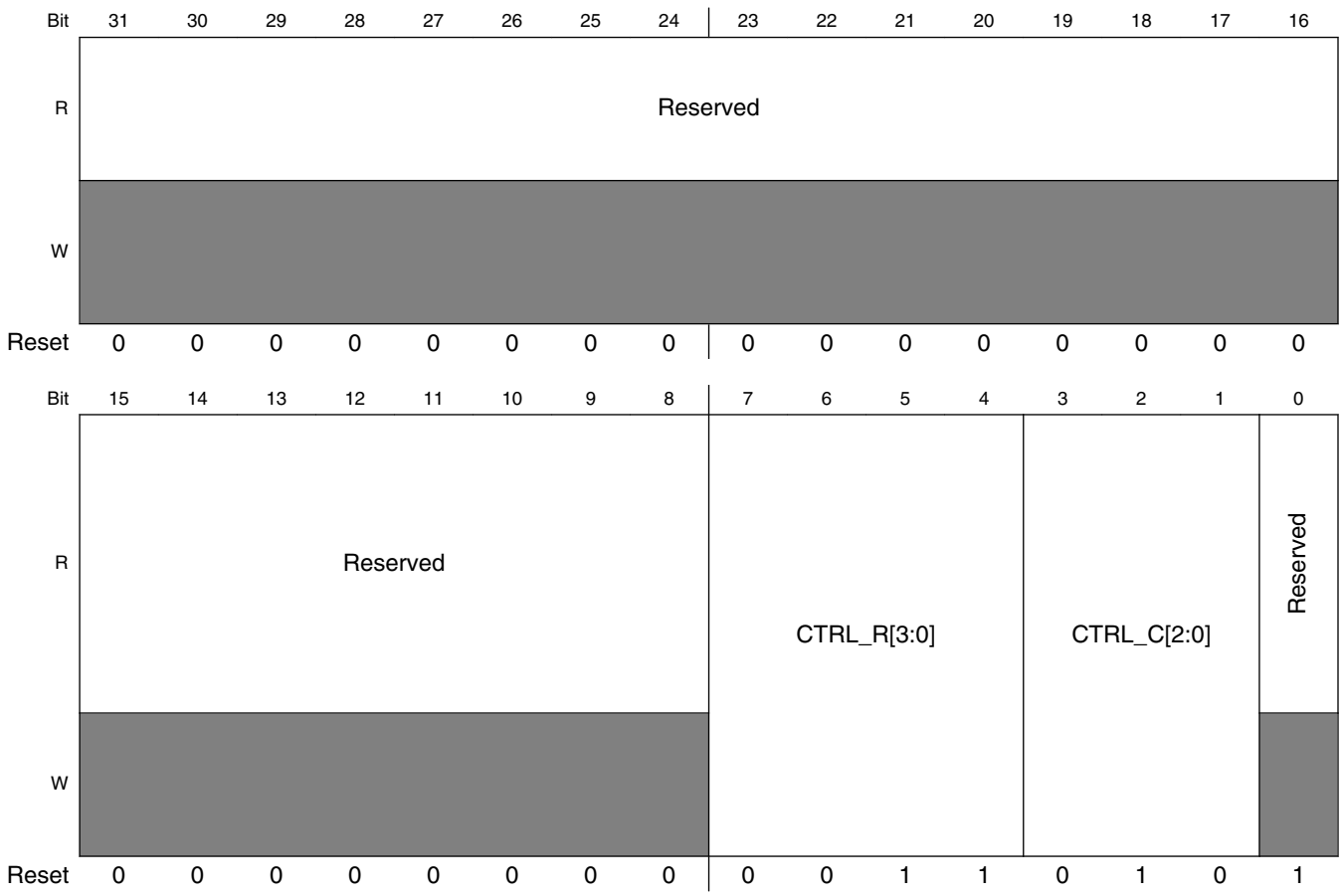
PCIE_PHY_CMN_REG03 field descriptions

Field	Description
31–4 Reserved	This field is reserved.
CTRL_CP[3:0]	PLL Bias Control ~ 0001 Default for SATA 0011 Default for PCIe ~

11.4.4.1.4 PCIE_PHY_CMN_REG04

This register is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission.

Address: 0h base + 16h offset = 16h



PCIE_PHY_CMN_REG04 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 CTRL_R[3:0]	PLL Loop Filter Res control ~ 0011 Default for SATA 1011 Default for PCIe ~
3–1 CTRL_C[2:0]	PLL Loop Filter Cap control ~

Table continues on the next page...

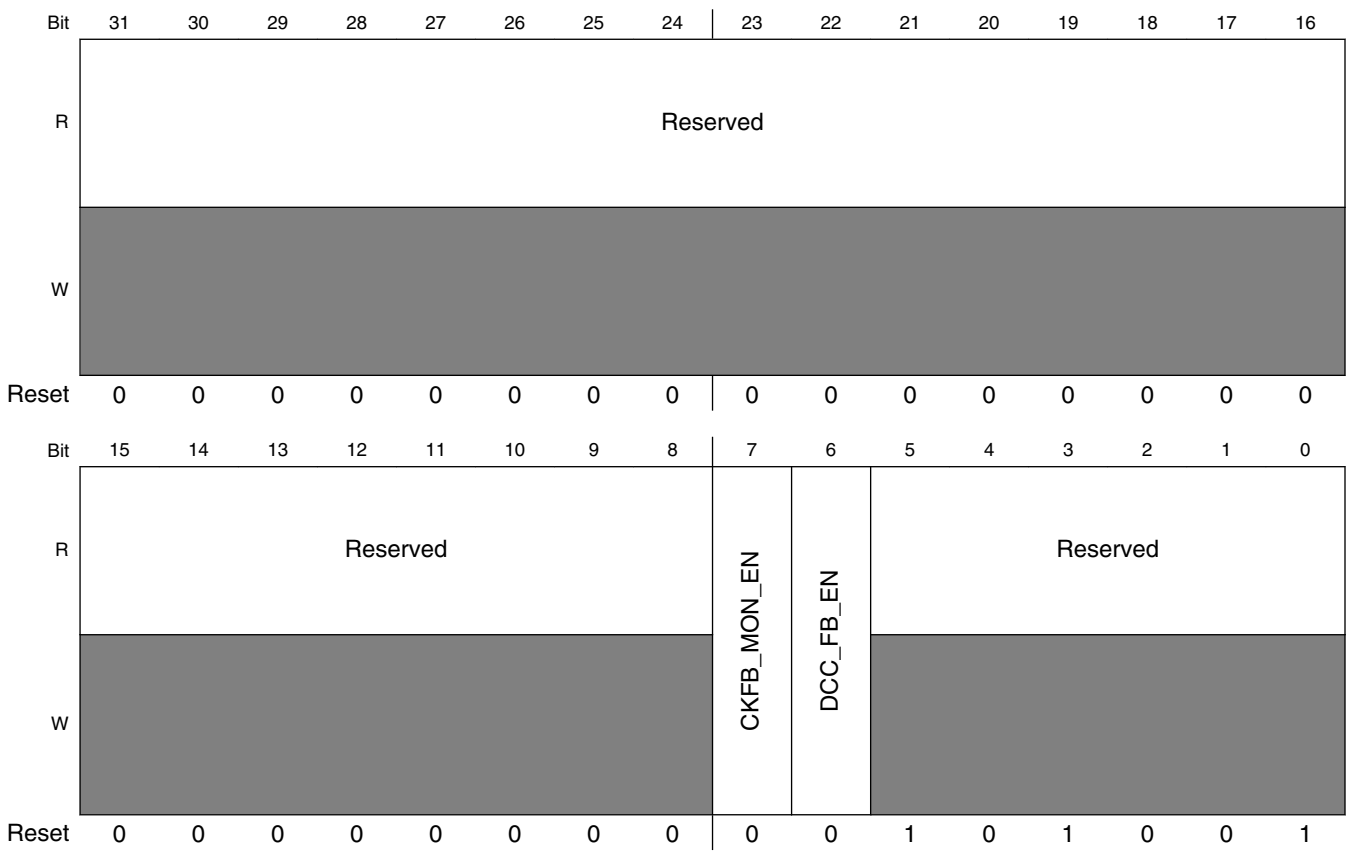
PCIE_PHY_CMN_REG04 field descriptions (continued)

Field	Description
	010 Default for SATA 100 Default for PCIe ~
0 Reserved	This field is reserved.

11.4.4.1.5 PCIE_PHY_CMN_REG05

This register is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission.

Address: 0h base + 20h offset = 20h



PCIE_PHY_CMN_REG05 field descriptions

Field	Description
31–8 Reserved	This field is reserved.

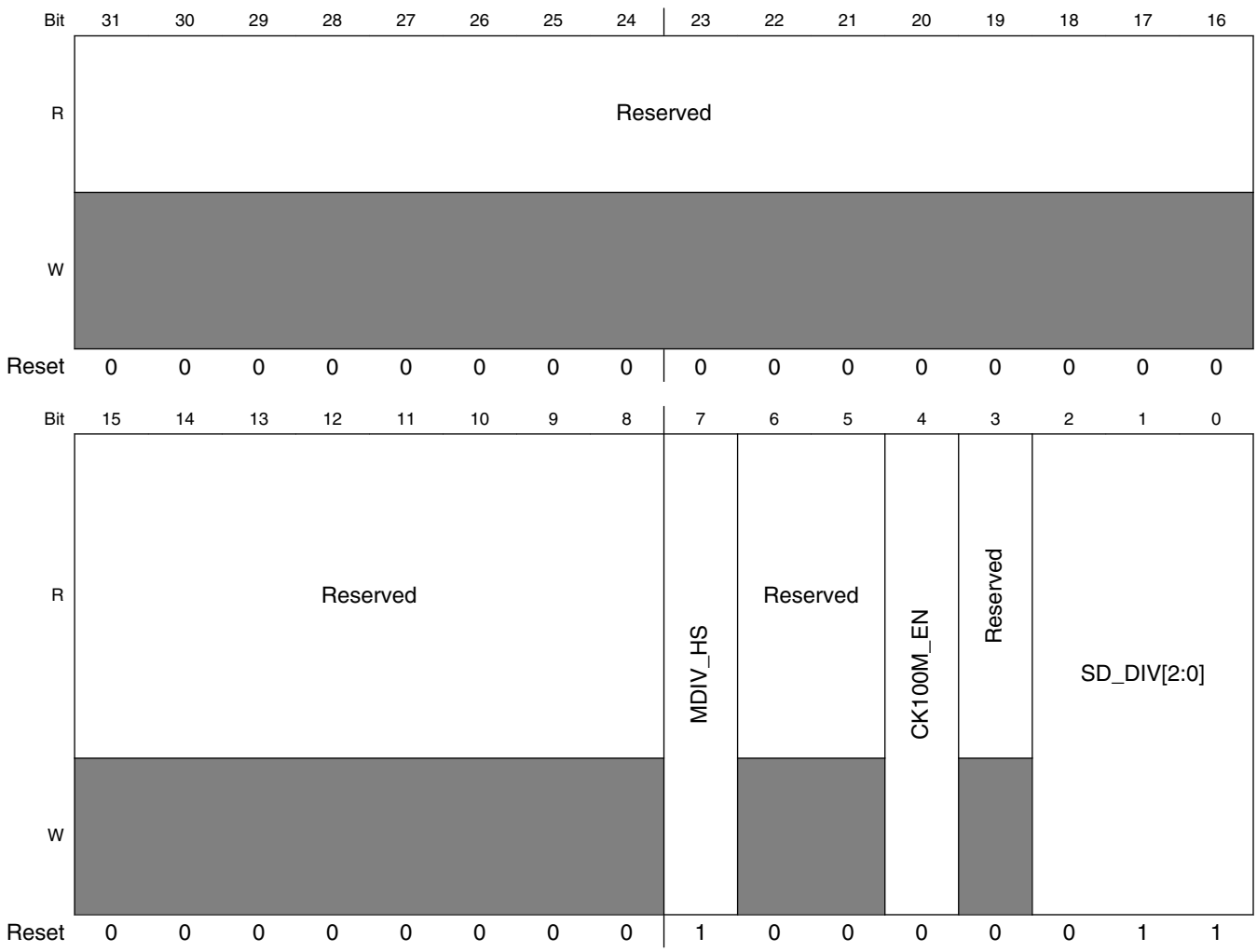
Table continues on the next page...

PCIE_PHY_CMN_REG05 field descriptions (continued)

Field	Description
7 CKFB_MON_EN	PLL Clock Monitoring Enable 0 (Default Value) 1
6 DCC_FB_EN	PLL DCC Feadback Enable 0 (Default Value) 1
Reserved	This field is reserved.

11.4.4.1.6 PCIE_PHY_CMN_REG06

Address: 0h base + 24h offset = 24h

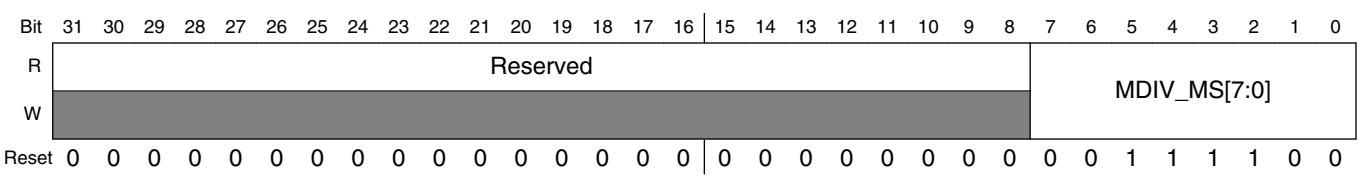


PCIE_PHY_CMN_REG06 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 MDIV_HS	PLL Main-divider Ratio Control 0 1/1 1 1/2 (Divide by 2, default)
6–5 Reserved	This field is reserved.
4 CK100M_EN	CK100M Enable **NOTE:** • It is also useful to monitor the REF_CKO clock in the ATE test because input reference clock connection can be checked. This CK100M_EN is used to enable 100MHz differential clock output for the PCIe PHY. 0 Static State 1 100 MHz REFCLK Output Enable
3 Reserved	This field is reserved.
SD_DIV[2:0]	Sigma Delta CLK control **NOTE:** It is prepared for the PLL target setting and should not be changed at all. The operation cannot be guaranteed for any change without written permission. ~ ~ 100 Default for PCIe ~

11.4.4.1.7 PCIE_PHY_CMN_REG07

Address: 0h base + 28h offset = 28h



PCIE_PHY_CMN_REG07 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
MDIV_MS[7:0]	PLL Main-divider Ratio Control 00000000 GND 00001011 GND

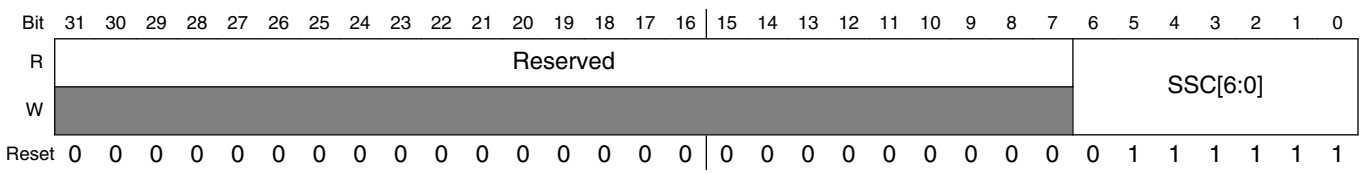
Table continues on the next page...

PCIE_PHY_CMN_REG07 field descriptions (continued)

Field	Description
	00001100 1/12 00001101 1/13 ~ 1/14 ~ 1/24 (Not shown in detail) 00011001 1/25 ~ 1/26 ~ 1/59 (Not shown in detail) 00111100 1/60 ~ 1/61 ~ 1/253 (Not shown in detail) 11111110 1/254 11111111 1/255

11.4.4.1.8 PCIE_PHY_CMN_REG0B

Address: 0h base + 2Ch offset = 2Ch

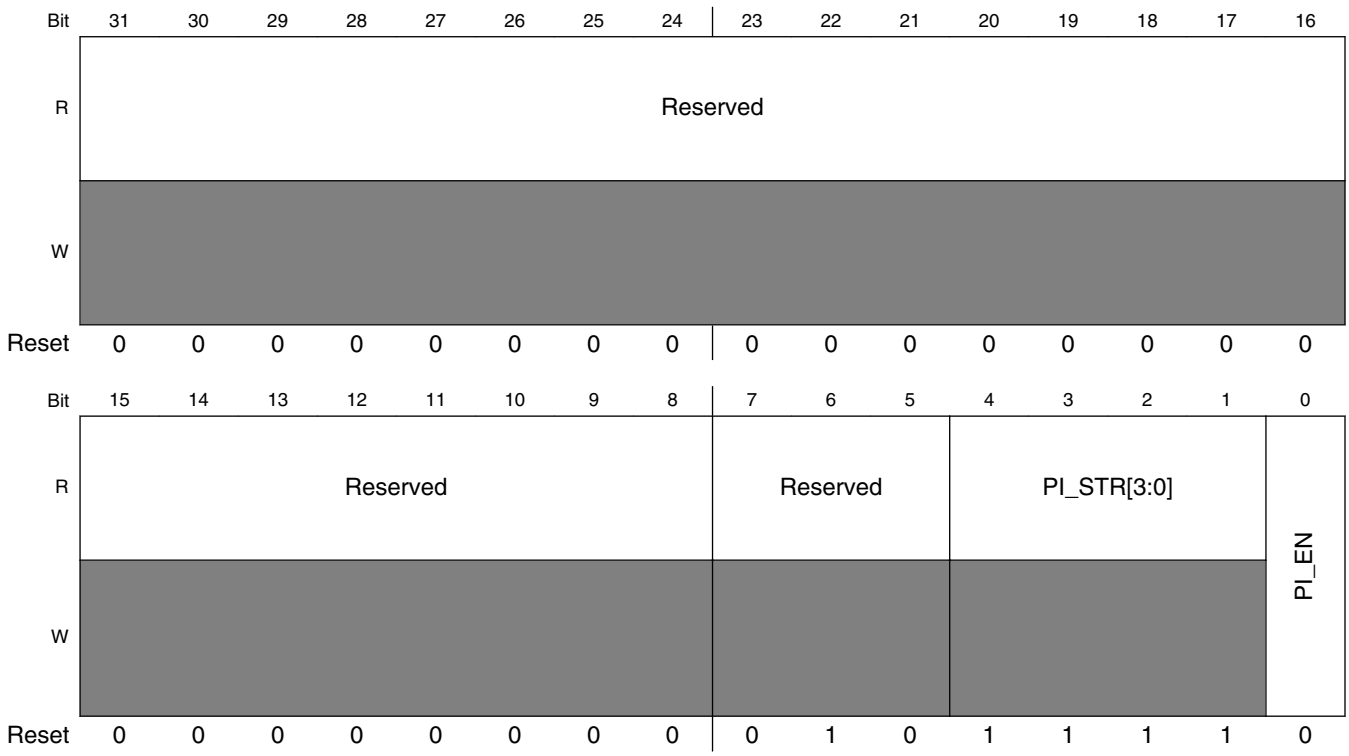


PCIE_PHY_CMN_REG0B field descriptions

Field	Description
31–7 Reserved	This field is reserved.
SSC[6:0]	SSC Amount Control **NOTE:** The amount of SSC modulation can be controlled by using SSC[6:0] and SSC_CNTL[1:0]. For example, SSC modulation amount = 20833 * SSC_CNTL[1:0] / SSC[6:0] ppm. In case of SSC[6:0] = 0011111 (62x) and SSC_CNTL[1:0] = 00 (1x), then SSC modulation = 20833 * 1 / 31 = 672 ppm. ~ Not allowed 0010101 21x (Minimum allowed value) ~ 0011111 31x (PCIe Default) 0111111 63x (SATA Default) 1111111 127x ~ Not allowed

11.4.4.1.9 PCIE_PHY_CMN_REG08

Address: 0h base + 32h offset = 32h

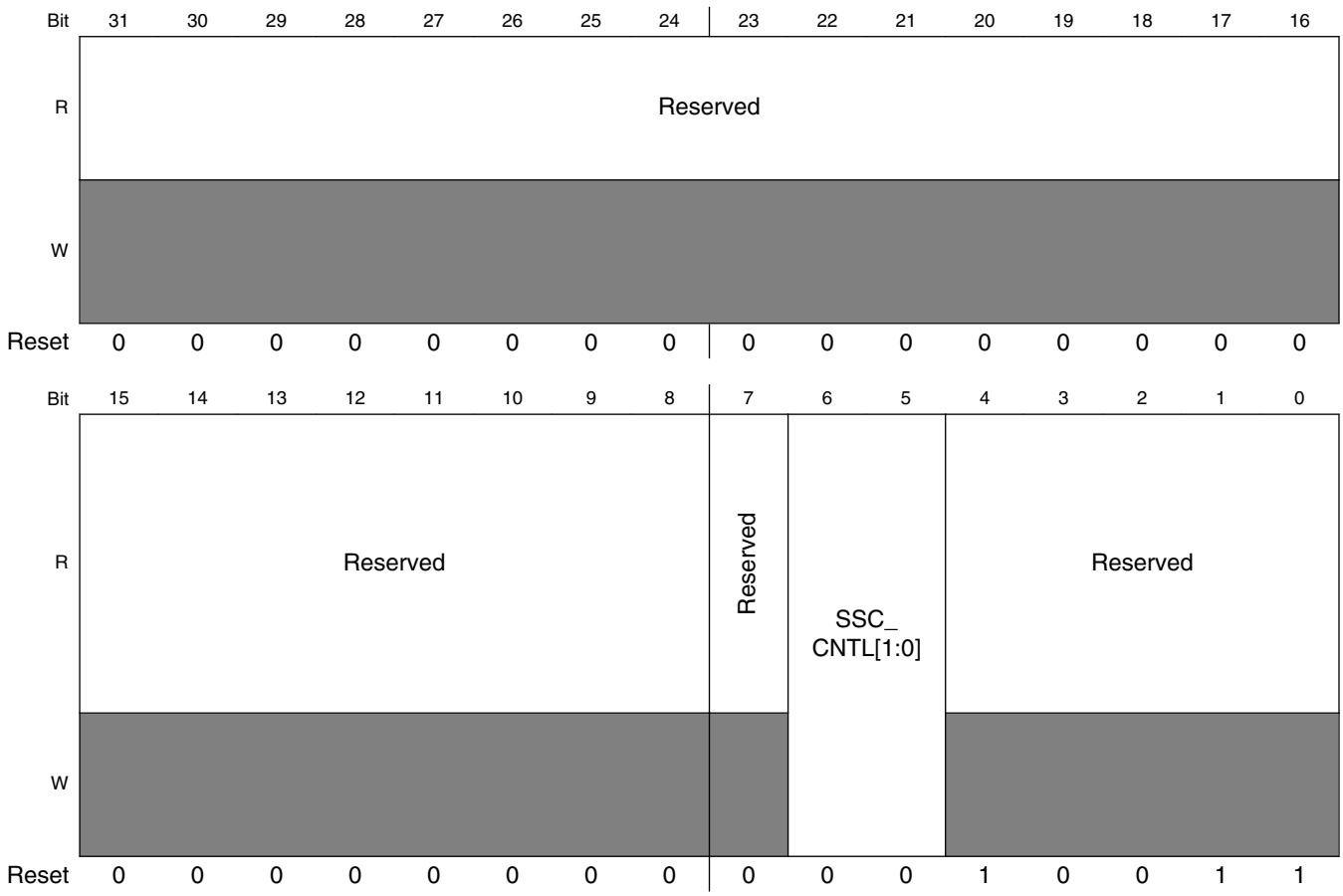


PCIE_PHY_CMN_REG08 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–5 Reserved	This field is reserved.
4–1 PI_STR[3:0]	PI Buffer strength control It is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission. ~ 1111 Default for SATA 0011 Default for PCIe ~
0 PI_EN	PI Enable 0 PI Disable (Default) 1 PI Enable

11.4.4.1.10 PCIE_PHY_CMN_REG09

Address: 0h base + 36h offset = 36h



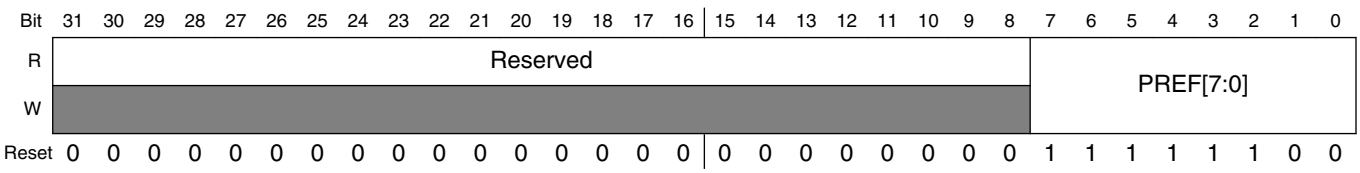
PCIE_PHY_CMN_REG09 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 Reserved	This field is reserved.
6–5 SSC_CNTL[1:0]	SSC Control **NOTE:** The amount of SSC modulation can be controlled by using SSC[6:0] and SSC_CNTL[1:0]. For example, SSC modulation amount = 20833 * SSC_CNTL[1:0] / SSC[6:0] ppm. In case of SSC[6:0] = 0011111 (62x) and SSC_CNTL[1:0] = 00 (1x), then SSC modulation = 20833 * 1 / 31 = 672 ppm. 00 1x (Default) 01 2x 10 4x 11 8x
Reserved	This field is reserved.

11.4.4.1.11 PCIE_PHY_CMN_REG11

This register is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission.

Address: 0h base + 40h offset = 40h



PCIE_PHY_CMN_REG11 field descriptions

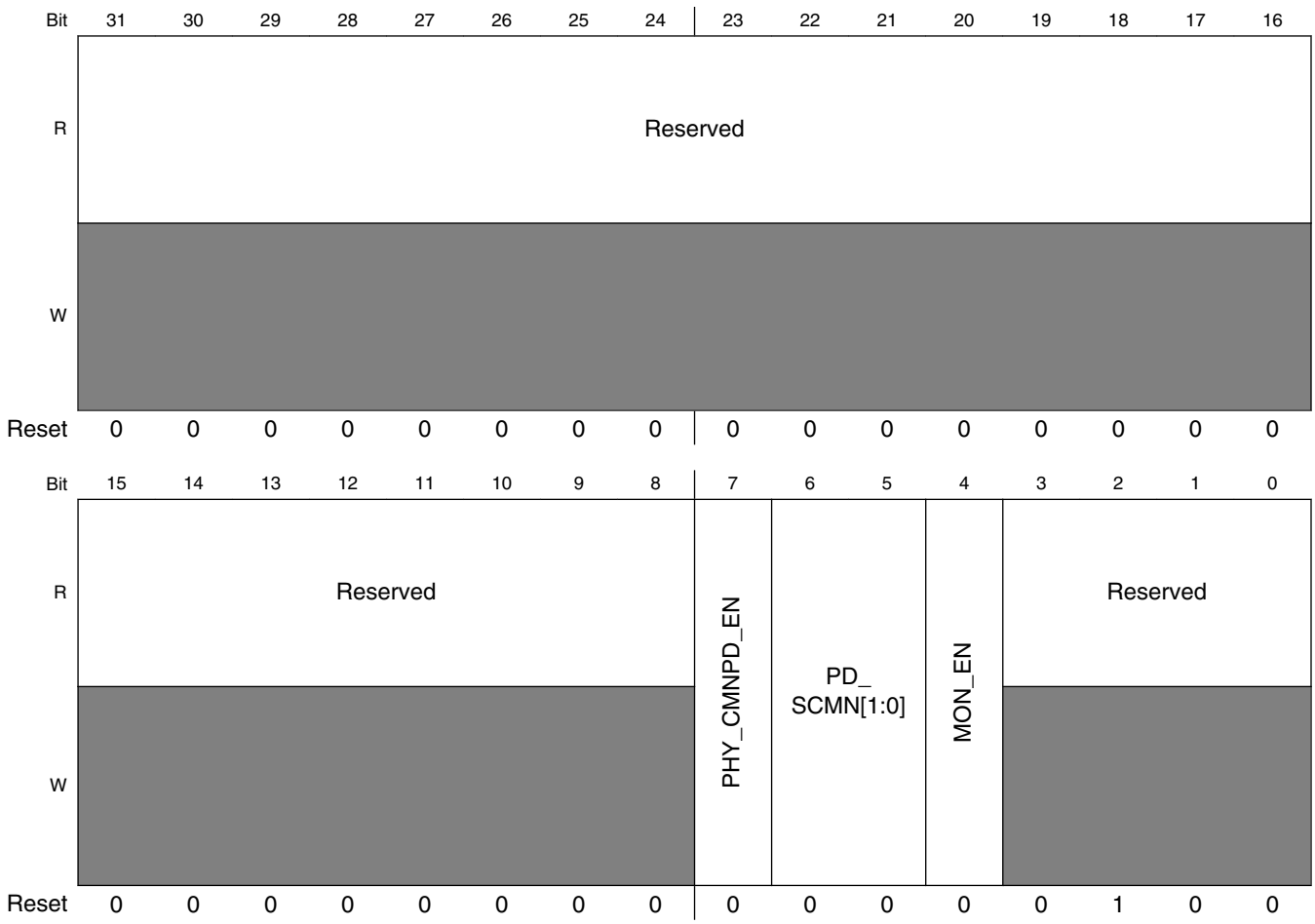
Field	Description
31–8 Reserved	This field is reserved.
PREF[7:0]	PLL Reference Control ~ 11111100 (Default Value) ~

11.4.4.1.12 PCIE_PHY_CMN_REG15

NOTE

- When you would like to disable all blocks of CMN, then PD_CMN is needed to be asserted (PD_CMN = 1). This PC_CMN powerdown mode can be used to save the power consumption for the unused CMN block.
- In the PHY, RESET pins for CMN block can be controlled by register for the debug purpose.

Address: 0h base + 60h offset = 60h



PCIE_PHY_CMN_REG15 field descriptions

Field	Description
31–8 Reserved	This field is reserved.

Table continues on the next page...

PCIE_PHY_CMN_REG15 field descriptions (continued)

Field	Description
7 PHY_CMNPD_EN	CMN Sub-block PD Control 0 Disabled for common block PD control 1 Enabled for common block PD control
6–5 PD_SCMN[1:0]	CMN Sub-block Powerdown 00 Common block enabled (Default) 10 Fine control option for PD 01 Fine control option for PD 11 Fine control option for PD
4 MON_EN	Clock Monitoring Enable **NOTE:** When MON_EN is "high", clock from PLL could be monitored via TESTCLKOUT port. The frequency of clock is 75 MHz. During the ATE test or bench test, this PLL clock monitoring feature can be used for debug. 0 Static State (Default Value) 1 Clock Monitoring Enable
Reserved	This field is reserved.

11.4.4.1.13 PCIE_PHY_CMN_REG16

Address: 0h base + 64h offset = 64h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved												PHY_SSC_EN[1:0]		Reserved	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

PCIE_PHY_CMN_REG16 field descriptions

Field	Description
31–4 Reserved	This field is reserved.
3–2 PHY_SSC_EN[1:0]	SSC Enable **NOTE:** This SSC feature can be enabled with SSC_EN pin from SoC and only the amount of SSC can be controlled with these registers for debug.When the SSC is enabled, PCIE_TX_P/N signal pins and the PCIE_REFCLKOUT_P/N signal pins would be SSC modulated. In addition, SSC mode can be overrided by (I2C or APB) register setting with I2C_SSC_EN[1:0] for debug purpose.

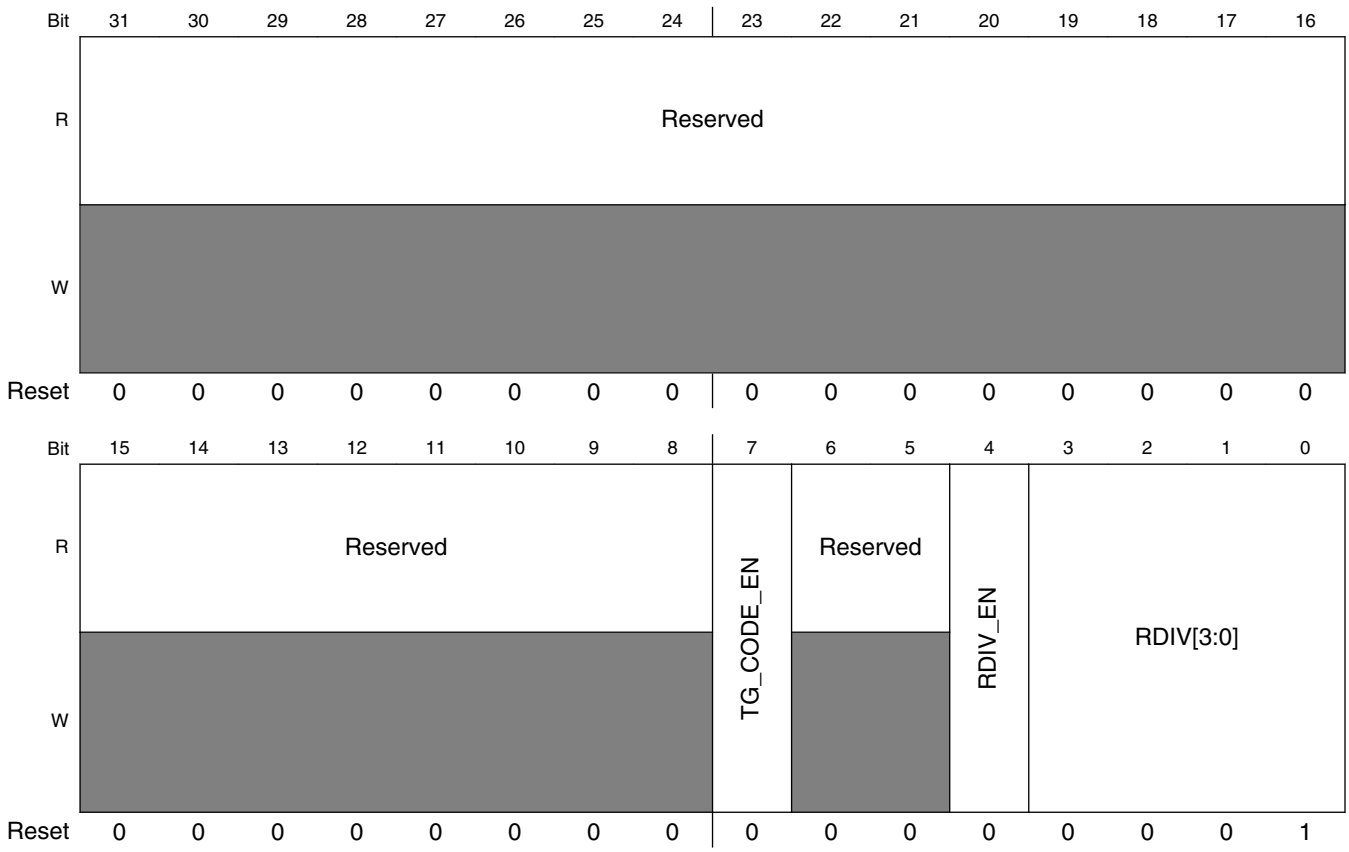
Table continues on the next page...

PCIE_PHY_CMN_REG16 field descriptions (continued)

Field	Description
	0x SSC control by primary port (SSC_EN) 10 SSC mode disabled by register setting 11 SSC mode enabled by register setting
Reserved	This field is reserved.

11.4.4.1.14 PCIE_PHY_CMN_REG17

Address: 0h base + 68h offset = 68h



PCIE_PHY_CMN_REG17 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 TG_CODE_EN	PLL Frequency Control Enable It is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission. 0 Disable TG_CODE_EN 1 Enable TG_CODE_EN

Table continues on the next page...

PCIE_PHY_CMN_REG17 field descriptions (continued)

Field	Description
6–5 Reserved	This field is reserved.
4 RDIV_EN	PLL Ref-divider Enable **NOTE:** In the PCIe PHY, the reference clock frequency should be 100 MHz. However only 50 MHz reference clock is available, then try to use this seeting for test purpose (For example, Reg02 = 14h, Reg17 = 12h. PDIV[3:0] = 0001, RDIV[3:0] = 0010, and RDIV_EN = 1). 0 Disabled 1 Enabled
RDIV[3:0]	PLL Ref-divider Ratio Control ~ < 1/2 (Not shown in detail) 0010 1/2 (Divide by 2, default) ~ > 1/2 (Not shown in detail)

11.4.4.1.15 PCIE_PHY_CMN_REG18

NOTE

This register is prepared for the PLL target setting and should not be changed at all. The operation cannot be guaranteed for any change without written permission.

Address: 0h base + 72h offset = 72h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																								TG_CODE[7:0]							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

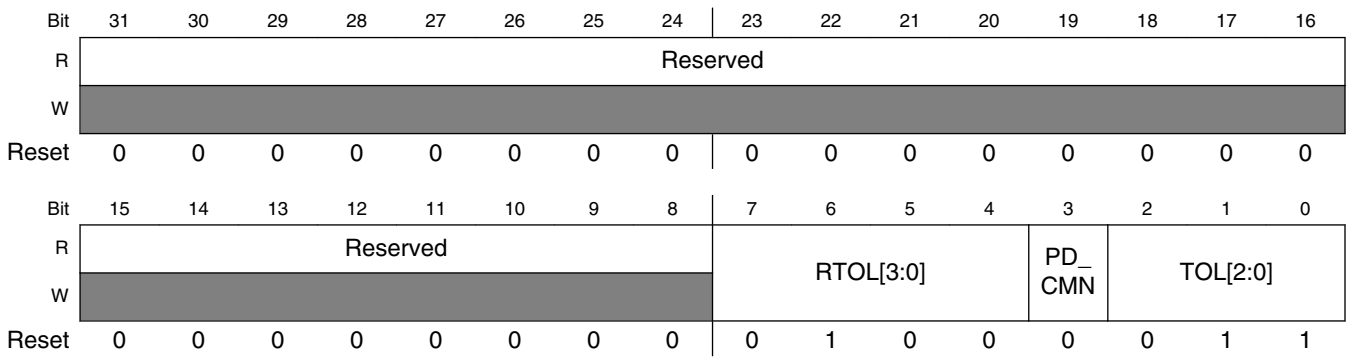
PCIE_PHY_CMN_REG18 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
TG_CODE[7:0]	PLL Frequency Control 00000000 (Default Value) ~ 11111111

11.4.4.1.16 PCIE_PHY_CMN_REG19

This register is prepared for the PLL target setting and cannot be changed at all. The operation cannot be guaranteed for any changes without written permission.

Address: 0h base + 76h offset = 76h



PCIE_PHY_CMN_REG19 field descriptions

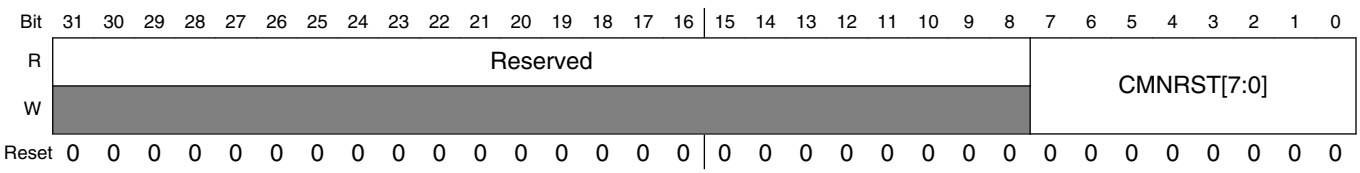
Field	Description
31–8 Reserved	This field is reserved.
7–4 RTOL[3:0]	PLL Tolerance Control ~ 0100 (Default Value) ~
3 PD_CMN	CMN All Block Powerdown **NOTE:** When disable all blocks of CMN, then PD_CMN is needed to be asserted (PD_CMN = 1). This PC_CMN powerdown mode can be used to save the power consumption for the unused CMN block. 0 Common block enabled (Default) 1 Common block all powerdown
TOL[2:0]	PLL Tolerance Control ~ 011 (Default Value) ~

11.4.4.1.17 PCIE_PHY_CMN_REG1A

NOTE

This register is prepared for the PLL target setting and should not be changed at all. The operation can not be guaranteed for any change without written permission.

Address: 0h base + 80h offset = 80h



PCIE_PHY_CMN_REG1A field descriptions

Field	Description
31–8 Reserved	This field is reserved.
CMNRST[7:0]	CMN Reset Control **NOTE:** In the PHY, RESET pins for CMN block can be controlled by register for the debug purpose. 0xxxxxxx Normal operation (Default) 10000000 Fine control option for RESET ~ Fine control option for RESET 11111111 Fine control option for RESET

11.4.4.2 PHY Register for Transceiver Block

The PCIe PHY has a common (CMN) block and multi-lanes of transceiver (TRSV) can be supported for the application for the high-speed serial data transmission and clock-data recovery. The following descriptions list the capabilities implemented by the PHY design team for internal use. The addresses of register in this chapter are for the transceiver ch0 and addition offset is required for another channel such as ch1 ~ 3.

PCIE_PHY_TRSV memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
84	PCIE_PHY_TRSV_REG21	32	R/W	0000_0016h	11.4.4.2.1/ 3459

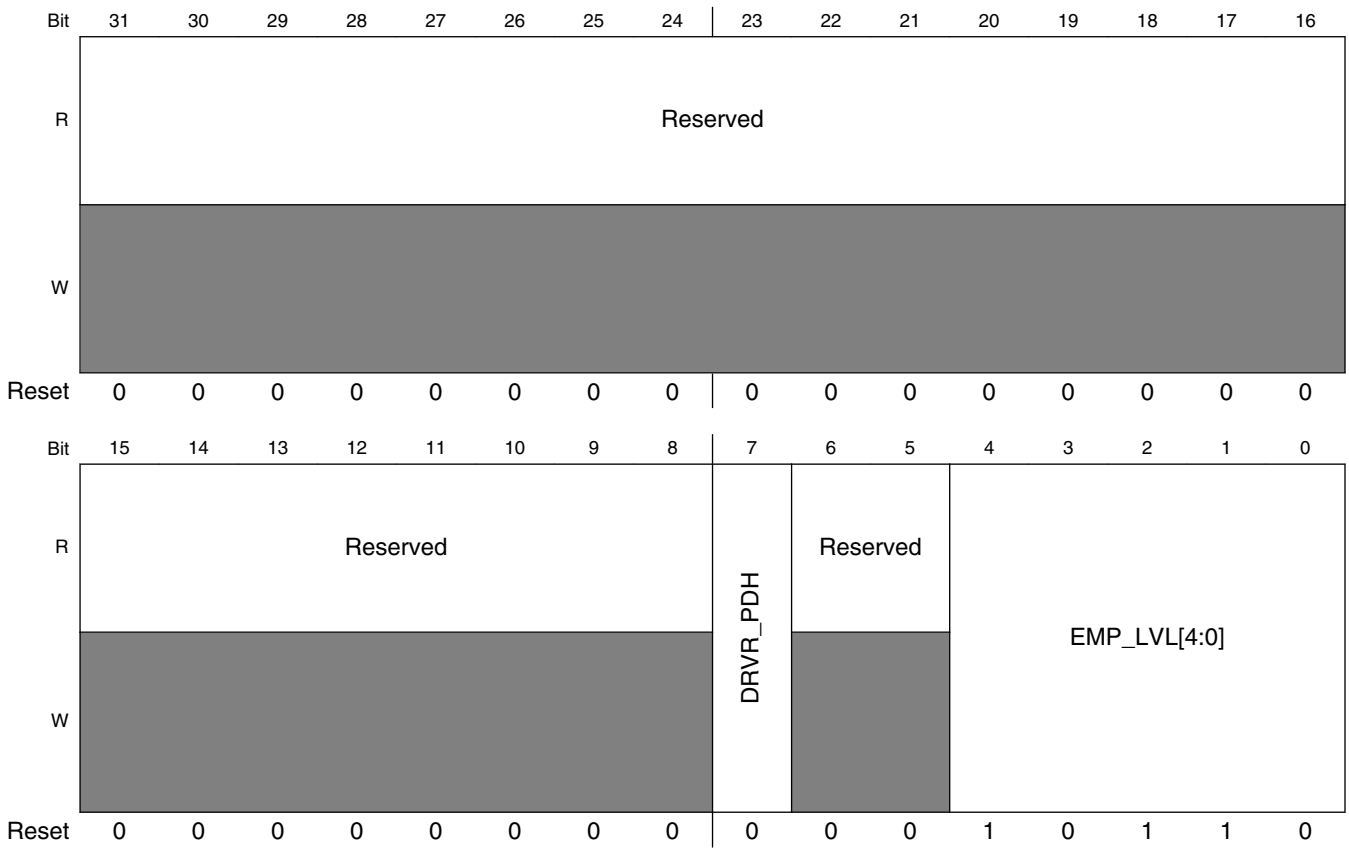
Table continues on the next page...

PCIE_PHY_TRSV memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
88	PCIE_PHY_TRSV_REG22	32	R/W	0000_002Ah	11.4.4.2.2/ 3460
96	PCIE_PHY_TRSV_REG24	32	R/W	0000_0048h	11.4.4.2.3/ 3461
AC	PCIE_PHY_TRSV_REG2B	32	R/W	0000_0081h	11.4.4.2.4/ 3462
E8	PCIE_PHY_TRSV_REG3A	32	R/W	0000_0008h	11.4.4.2.5/ 3463
F8	PCIE_PHY_TRSV_REG3E	32	R/W	0000_0004h	11.4.4.2.6/ 3464
100	PCIE_PHY_TRSV_REG25	32	R/W	0000_0042h	11.4.4.2.7/ 3466
104	PCIE_PHY_TRSV_REG26	32	R/W	0000_00BCh	11.4.4.2.8/ 3467
116	PCIE_PHY_TRSV_REG29	32	R/W	0000_000Ch	11.4.4.2.9/ 3468
124	PCIE_PHY_TRSV_REG31	32	R/W	0000_0035h	11.4.4.2.10/ 3468
132	PCIE_PHY_TRSV_REG33	32	R/W	0000_0020h	11.4.4.2.11/ 3469
144	PCIE_PHY_TRSV_REG36	32	R/W	0000_00B0h	11.4.4.2.12/ 3469
148	PCIE_PHY_TRSV_REG37	32	R/W	0000_00A0h	11.4.4.2.13/ 3471
152	PCIE_PHY_TRSV_REG38	32	R/W	0000_0000h	11.4.4.2.14/ 3472
156	PCIE_PHY_TRSV_REG39	32	R/W	0000_0000h	11.4.4.2.15/ 3473
160	PCIE_PHY_TRSV_REG40	32	R/W	0000_0000h	11.4.4.2.16/ 3474
168	PCIE_PHY_TRSV_REG42	32	R/W	0000_0000h	11.4.4.2.17/ 3475

11.4.4.2.1 PCIE_PHY_TRSV_REG21

Address: 0h base + 84h offset = 84h

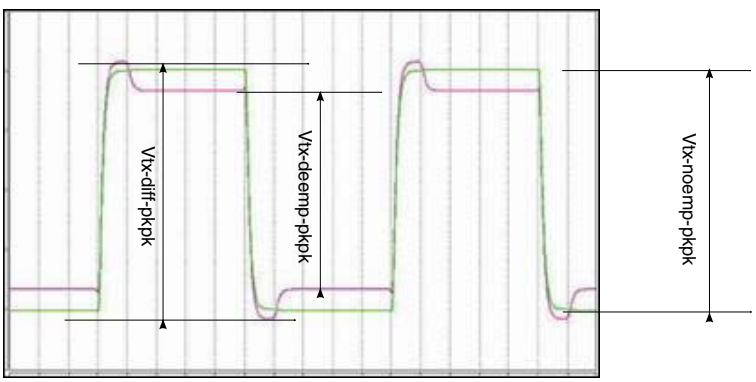


PCIE_PHY_TRSV_REG21 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 DRVR_PDH	TX Driver Option 0 Default for PCIe 1 Reduce Power consumption
6–5 Reserved	This field is reserved.
EMP_LVL[4:0]	TX De-emphasis Level Control **NOTE:** • For the TX properties such as amplitude and de-emphasis, please use default setting for the first time evaluation. If the PHY is connected through a long cable, (or media), generally high pre-emphasis level setting would be better. But there would be optimization setting points according to the cable length. • TX amplitude and de-emphasis ratio can be calculated as follows. Vtx-noemp-pkpk means differential peak to peak amplitude when there is no de-emphasis. Vtx-diff-pkpk means differential peak to peak amplitude which is included de-emphasis. And Vtx-deemp-pkpk means differential inner flatted peak to peak amplitude which is reduced by de-emphasis.

Table continues on the next page...

PCIE_PHY_TRSV_REG21 field descriptions (continued)

Field	Description
	• Vtx-noemp-pkpk = (7.5 * bit(DRV_LVL[5]) + 3.75 * bit (DRV_LVL[4]) + 2 * bit(DRV_LVL[3]) + 1 * bit (DRV_LVL[2]) + 0.5 * bit (DRV_LVL[1]) + 0.25 * bit (DRV_LVL[0]) + 7.5) * 5.32 • Vtx-diff-pkpk = ((7.5 * bit (DRV_LVL[5]) + 3.75 * bit (DRV_LVL[4]) + 2 * bit (DRV_LVL[3]) + 1 * bit (DRV_LVL[2]) + 0.5 * bit (DRV_LVL[1]) + 0.25 * bit (DRV_LVL[0]) + 7.5) + (3.75 * bit (EMP_LVL[4]) + 2 * bit (EMP_LVL[3]) + 1 * bit (EMP_LVL[2]) + 0.5 * bit (EMP_LVL[1]) + 0.25 * bit (EMP_LVL[0])) * 0.7) * 5.32 • Vtx-deemp-pkpk = ((7.5 * bit (DRV_LVL[5]) + 3.75 * bit (DRV_LVL[4]) + 2 * bit (DRV_LVL[3]) + 1 * bit (DRV_LVL[2]) + 0.5 * bit (DRV_LVL[1]) + 0.25 * bit (DRV_LVL[0]) + 7.5) - (3.75 * bit (EMP_LVL[4]) + 2 * bit (EMP_LVL[3]) + 1 * bit (EMP_LVL[2]) + 0.5 * bit (EMP_LVL[1]) + 0.25 * bit (EMP_LVL[0]))) * 5.32 • TX De-emphasis can be calculated as follows:  De – emphasis = –20 log ((Vtx - deemp - pkpk) / (Vtx - diff - pkpk)) (dB) • For example DRV_LVL[5:0] = 100000 and EMP_LVL[4:0] = 01000.Vtx – diff – pkpk = 7.5 + 0 + 0 + 0 + 0 + 0 + 7.5 + 0.7 * 0 + 2 + 0 + 0 + 0 * 5.32 = 872 mVpkpk De – emphasis = –20 log (7.5 + 0 + 0 + 0 + 0 + 0 + 7.5 – (0 + 2 + 0 + 0 + 0)) / (7.5 + 0 + 0 + 0 + 0 + 0 + 7.5 + 0.7 * (0 + 2 + 0 + 0 + 0)) = 2 dB 00000 No de-emphasis 00100 Default for SATA 10100 3.5 dB de-emphasis ~ 11110 -6 dB dB de-emphasis ~ 11111 Max de-emphasis

11.4.4.2.2 PCIE_PHY_TRSV_REG22

Address: 0h base + 88h offset = 88h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																										DRV_LVL[5:0]					
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0

PCIE_PHY_TRSV_REG22 field descriptions

Field	Description
31–6 Reserved	This field is reserved.
DRV_LVL[5:0]	TX Differential Output (PCIE_TX_P/PCIE_TX_N) Amplitude Control 000000 Minimum Swing 100000 Default Swing for SATA 101010 Default Swing for PCIe ~ 111111 Maximum Swing

11.4.4.2.3 PCIE_PHY_TRSV_REG24

NOTE

- For the RX squelch detector threshold votlage, this threshold voltage can be changed intentionally for debug purpose. In the noisy application, the threshold voltage of squelch detector might be increased for better detection although it could cause the violation in the specification.
- RX equalizer setting might be changed to have better signal quality. For example, RX input jitter from ISI (Inter Symbol Interference) can be improved with RX equalization feature.

Address: 0h base + 96h offset = 96h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RX_SS_PD	RX_EQS	RX_SS[1:0]		RX_EQ_SEL	Reserved		
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0

PCIE_PHY_TRSV_REG24 field descriptions

Field	Description
31–8 Reserved	This field is reserved.

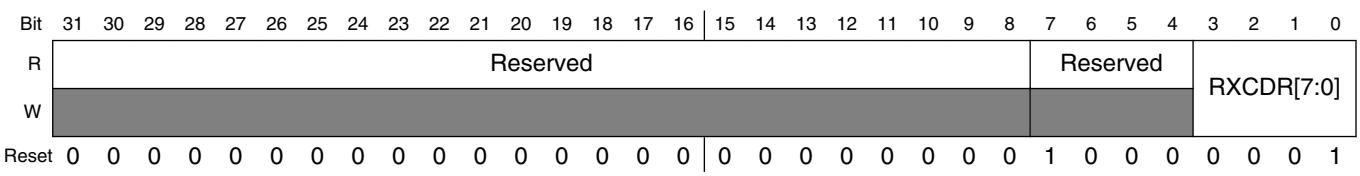
Table continues on the next page...

PCIE_PHY_TRSV_REG24 field descriptions (continued)

Field	Description
7 RX_SS_PD	RX Sense Powerdown 0 Default 1 Powerdown for Sense
6 RX_EQS	RX Equalizer Select 0 1 Default
5–4 RX_SS[1:0]	RX Sense Control 00 Default ~
3 RX_EQ_SEL	RX Equalizer Select Enable 0 1 Manual: Default
Reserved	This field is reserved.

11.4.4.2.4 PCIE_PHY_TRSV_REG2B

Address: 0h base + ACh offset = ACh



PCIE_PHY_TRSV_REG2B field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 Reserved	This field is reserved.
RXCDR[7:0]	RX CDR BW Control **NOTE:** RX CDR(Clock and data recovery) BW(bandwidth) can be adjusted using RXCDR[7:0], however do not change it. It may cause the performance degradation and can not guarantee the operation. ~ 10000010 Default ~

11.4.4.2.5 PCIE_PHY_TRSV_REG3A

NOTE

- This register can be used for the serial data (PCIE_TX_P / PCIE_TX_N or PCIE_RX_P / PCIE_RX_N) polarity inversion. Some cases, if high-speed signal pins are twisted or connected through vias, using this polarity.
- TX and RX parallel data also can have bit and / or byte order change to be flexible with connection errors between the PHY and the LINK controller.

Address: 0h base + E8h offset = E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								Reserved	Reserved	Reserved		COMDET_EN	Reserved	RDIMODE	TDIMODE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

PCIE_PHY_TRSV_REG3A field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 Reserved	This field is reserved.
6 Reserved	This field is reserved.
5–4 Reserved	This field is reserved.
3 COMDET_EN	Comma Detection Enable **NOTE:** When COMDET_EN is "low", RXD[0:19] is not aligned for debug purpose.When ADD_ALIGN is asserted (ADD_ALIGN = 1), ALIGNp is added during the pattern generation in BIST mode. 0 Comma Detection Disable 1 Comma Detection Enable (Default)
2 Reserved	This field is reserved.
1 RDIMODE	RX BitWidth Select 0 20-bit Mode (Default) 1 40-bit Mode
0 TDIMODE	TX BitWidth Select 0 20-bit Mode (Default) 1 40-bit Mode

11.4.4.2.6 PCIE_PHY_TRSV_REG3E

Address: 0h base + F8h offset = F8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																								Reserved				DET_CNT[3:0]			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

PCIE_PHY_TRSV_REG3E field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 Reserved	This field is reserved.
DET_CNT[3:0]	RX Detection Control **NOTE:** For the DET_CNT[3:0], the number of K28.5 detection for RX_VALID can be changed.

Table continues on the next page...

PCIE_PHY_TRSV_REG3E field descriptions (continued)

Field	Description
	0000 No detection ~ 0100 Four K28.5 detection (Default) ~

11.4.4.2.7 PCIE_PHY_TRSV_REG25

NOTE

- For the RX squelch detector threshold votlage, this threshold voltage can be changed intentionally for debug purpose. In the noisy application, the threshold voltage of squelch detector might be increased for better detection although it could cause the violation in the specification.
- RX equalizer setting might be changed to have better signal quality. For example, RX input jitter from ISI (Inter Symbol Interference) can be improved with RX equalization feature.

Address: 0h base + 100h offset = 100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

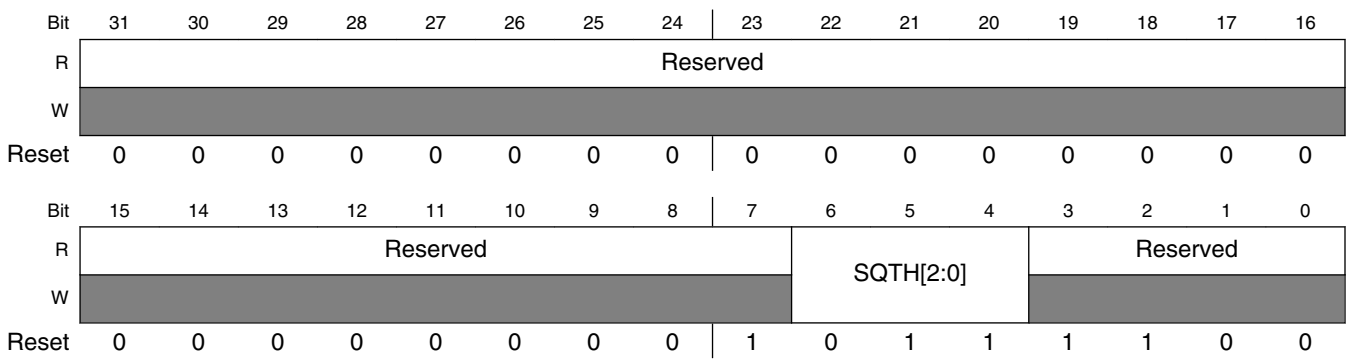
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RXEQ[3:0]				Reserved	RXEQS[2:0]		
W																
Reset	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0

PCIE_PHY_TRSV_REG25 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–4 RXEQ[3:0]	RX Equalizer Control ~ 0100 Default ~
3 Reserved	This field is reserved.
RXEQS[2:0]	RX Equalizer Setting ~ 0100 Default ~

11.4.4.2.8 PCIE_PHY_TRSV_REG26

Address: 0h base + 104h offset = 104h

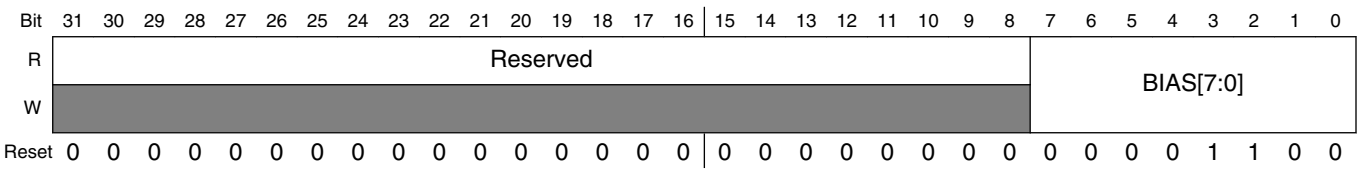


PCIE_PHY_TRSV_REG26 field descriptions

Field	Description
31–7 Reserved	This field is reserved.
6–4 SQTH[2:0]	RX Squelch Detect Threshold Control 000 Min threshold ~ 011 Default threshold voltage of squelch detect ~ 111 **Max threshold**
Reserved	This field is reserved.

11.4.4.2.9 PCIE_PHY_TRSV_REG29

Address: 0h base + 116h offset = 116h

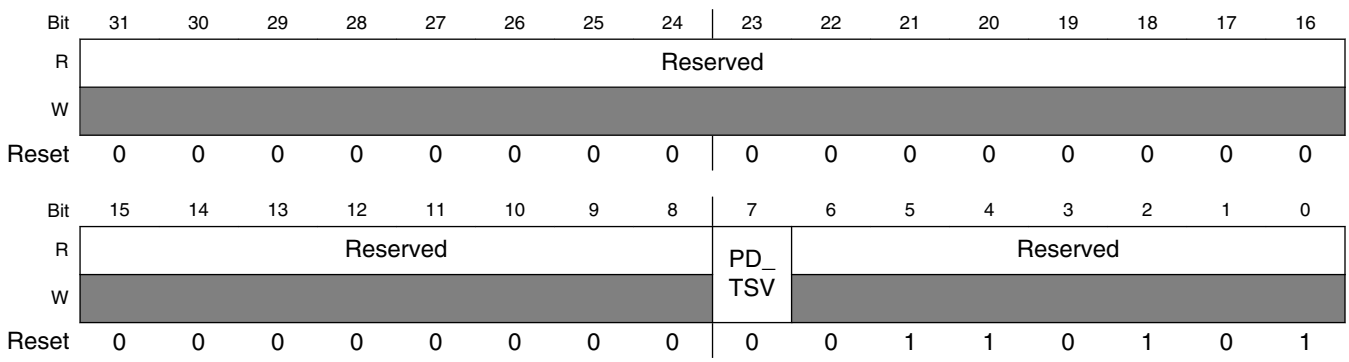


PCIE_PHY_TRSV_REG29 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
BIAS[7:0]	TRSV Bias Current Control 00000100 Default for SATA 00001100 Default for PCIe

11.4.4.2.10 PCIE_PHY_TRSV_REG31

Address: 0h base + 124h offset = 124h

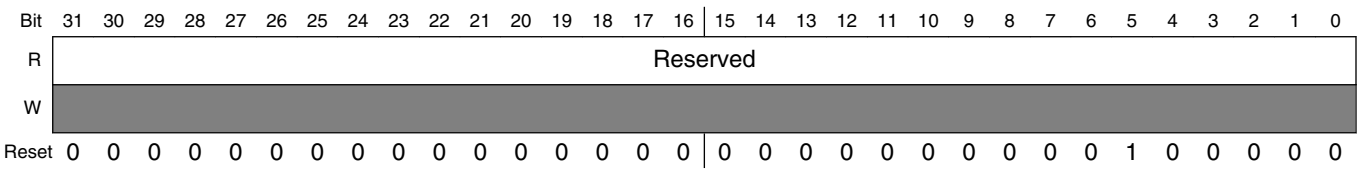


PCIE_PHY_TRSV_REG31 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 PD_TSV	Transceiver Block All Powerdown Enable 0 Transceiver Block powerdown disable 1 Transceiver Block powerdown enable
Reserved	This field is reserved.

11.4.4.2.11 PCIE_PHY_TRSV_REG33

Address: 0h base + 132h offset = 132h

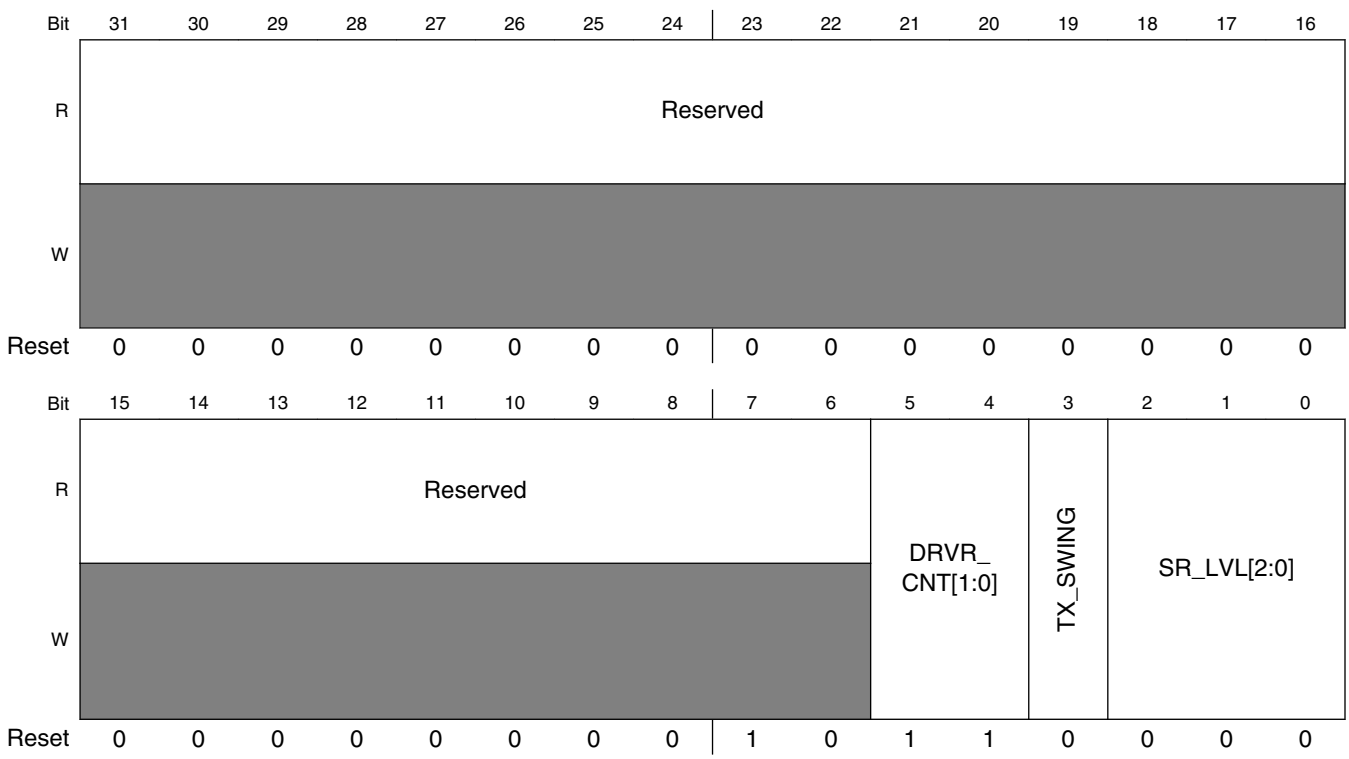


PCIE_PHY_TRSV_REG33 field descriptions

Field	Description
Reserved	This field is reserved.

11.4.4.2.12 PCIE_PHY_TRSV_REG36

Address: 0h base + 144h offset = 144h



PCIE_PHY_TRSV_REG36 field descriptions

Field	Description
31–6 Reserved	This field is reserved.
5–4 DRVR_CNT[1:0]	TX Driver Common Mode Control

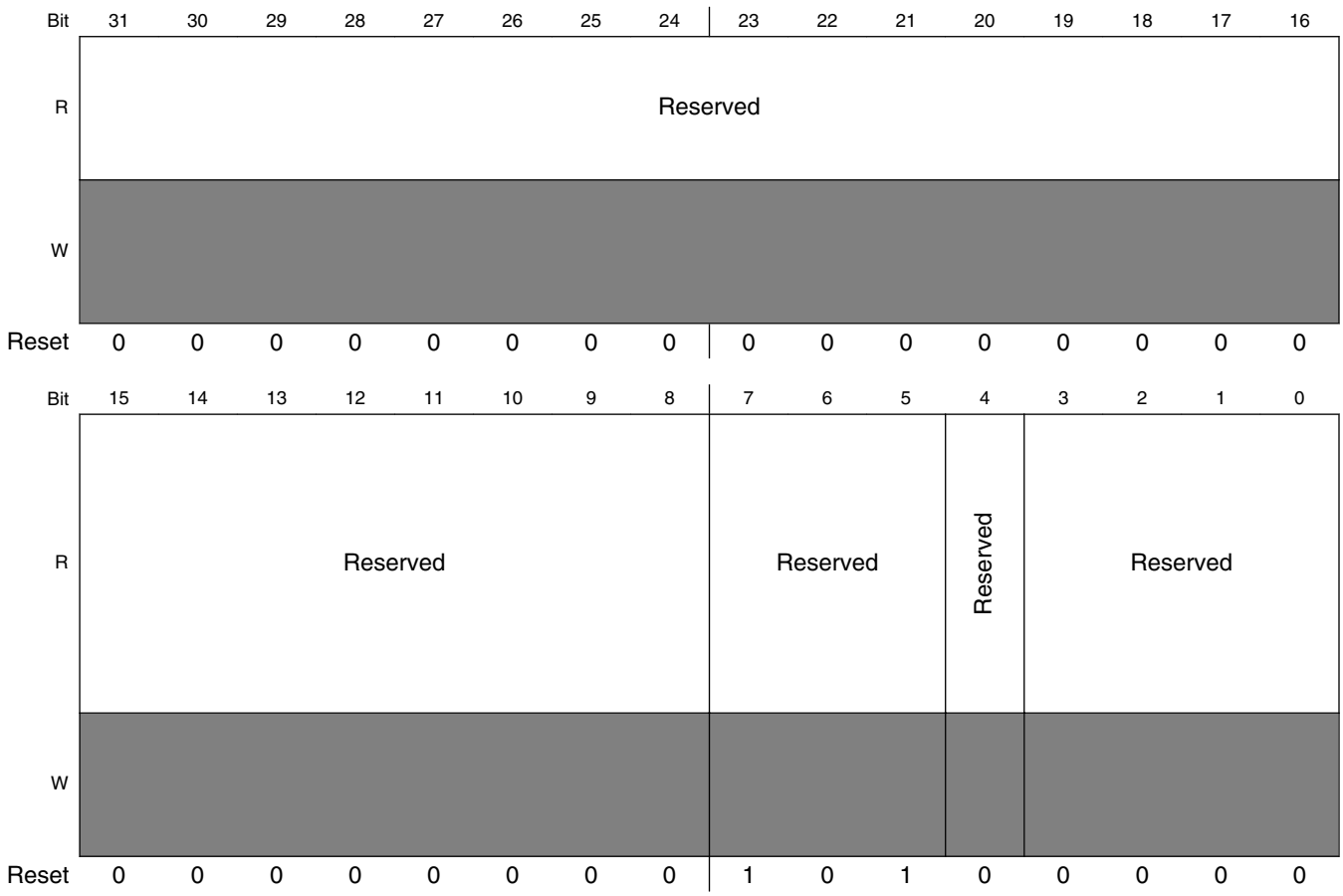
Table continues on the next page...

PCIE_PHY_TRSV_REG36 field descriptions (continued)

Field	Description
	00 Default for SATA 11 Default for PCIe
3 TX_SWING	TX Driver Control **NOTE:** • TX swing level can be changed according to the TX de-emphasis level control setting. • TX voltage swing unit step is different. TX swing specification may be changed because of PVT variation. If you reduce TX swing, TX IO power consumption would be also reduced proportionally. If the PHY is connected through a long cable, you had better large swing setting because of the loss in the channel. For the Compliance test, the test point should checked first and sometimes the de- embedding of cable and/or PCB trace might be needed. 0 Full-swing (Default) 1 Half-swing
SR_LVL[2:0]	TX Slew-rate Control 000 Default (~ 60 pS) 001 ~ 90 pS 010 ~ 120 pS 011 ~ 150 pS 1XX Max

11.4.4.2.13 PCIE_PHY_TRSV_REG37

Address: 0h base + 148h offset = 148h



PCIE_PHY_TRSV_REG37 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–5 Reserved	This field is reserved.
4 Reserved	This field is reserved.
Reserved	This field is reserved.

11.4.4.2.14 PCIE_PHY_TRSV_REG38

NOTE

- TX and RX parallel data also can have bit and / or byte order change to be flexible with connection errors between the PHY and the LINK controller.
- This BIST mode would be used for ATE test and addition features such as loopback mode can be controlled with the proper register settings. Make sure the powerdown mode control pins are not enabled. (i.e. these powerdown mode pins should be controllable by register in the SoC). These loopback mode are not Compliance mode which is usually implemented in the LINK controller side.When LB_EN is "high", Loopback mode works. This LB_EN is used with LB_MODE[1:0]. If you would like to use TX to RX serial loopback, you should set LB_EN = "1" and LB_MODE = "00'.

Address: 0h base + 152h offset = 152h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved										TX_INV	RX_INV	ADD_ALIGN	Reserved		
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_TRSV_REG38 field descriptions

Field	Description
31–6 Reserved	This field is reserved.
5 TX_INV	TX Inversion Control **NOTE:** This register is used for the serial data (PCIE_TX_P / PCIE_TX_N or PCIE_RX_P / PCIE_RX_N) polarity inversion. Some cases, if high-speed signal pins are twisted or connected through vias, using this polarity. 0 Normal Mode (Default Value) 1 PCIE_TX_P/PCIE_TX_N Output Data
4 RX_INV	0 Normal Mode (Default Value) 1 PCIE_RX_P/PCIE_RX_N Output Data
3 ADD_ALIGN	Align Primitive Control 0 No 1 ALIGNp is added.
Reserved	This field is reserved.

11.4.4.2.15 PCIE_PHY_TRSV_REG39

NOTE

- This register can be used for the serial data (PCIE_TX_P / PCIE_TX_N or PCIE_RX_P / PCIE_RX_N) polarity inversion. Some cases, if high-speed signal pins are twisted or connected through vias, you can use this polarity.
- TX and RX parallel data also can have bit and / or byte order change to be flexible with connection errors between the PHY and the LINK controller.

Address: 0h base + 156h offset = 156h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

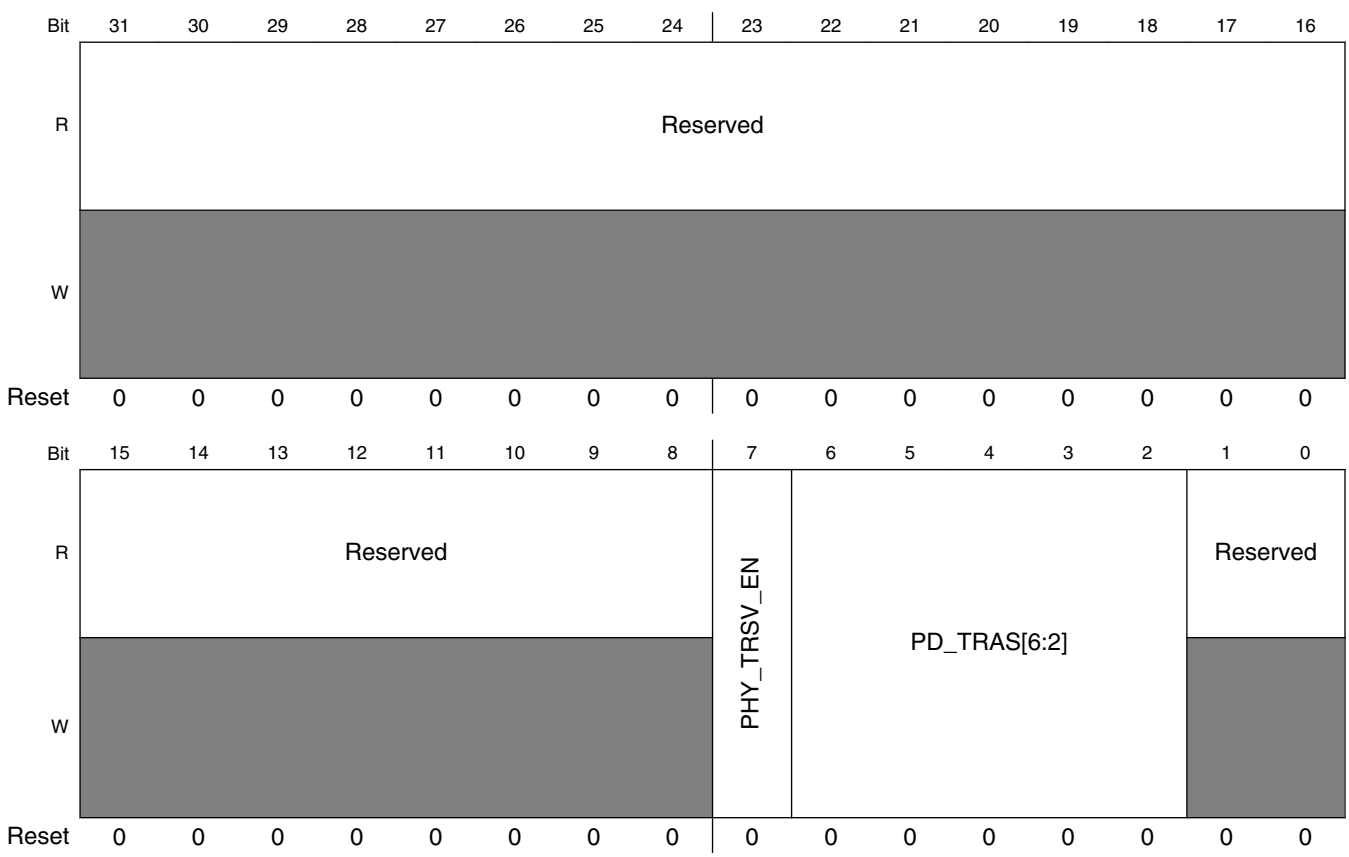
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RD_ ORD[1:0]		TD_ORD[1:0]		Reserved			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PCIE_PHY_TRSV_REG39 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7–6 RD_ORD[1:0]	RX Bit Order Control 00 No Order Reverse. RXD[0] to RXD[9] is received and then RXD[10] to RXD[19] is received for RXD[0:19]. 01 Bit Order Reverse. RXD[9] to RXD[0] is received and then RXD[19] to RXD[10] is received. 10 Byte Order Reverse. RXD[10] to RXD[19] is received and then RXD[0] to RXD[9] is received. 11 Bit and Byte Order Reverse. RXD[19] to RXD[10] is received and then RXD[9] to RXD[0] is received.
5–4 TD_ORD[1:0]	TX Bit Order Control 00 No Order Reverse. TXD[0] to TXD[9] is transmitted and then TXD[10] to TXD[19] is transmitted for TXD[0:19]. 01 Bit Order Reverse. TXD[9] to TXD[0] is transmitted and then TXD[19] to TXD[10] is transmitted. 10 Byte Order Reverse. TXD[10] to TXD[19] is transmitted and then TXD[0] to TXD[9] is transmitted. 11 Bit and Byte Order Reverse. RXD[19] to RXD[10] is received and then RXD[9] to RXD[0] is received.
Reserved	This field is reserved.

11.4.4.2.16 PCIE_PHY_TRSV_REG40

Address: 0h base + 160h offset = 160h

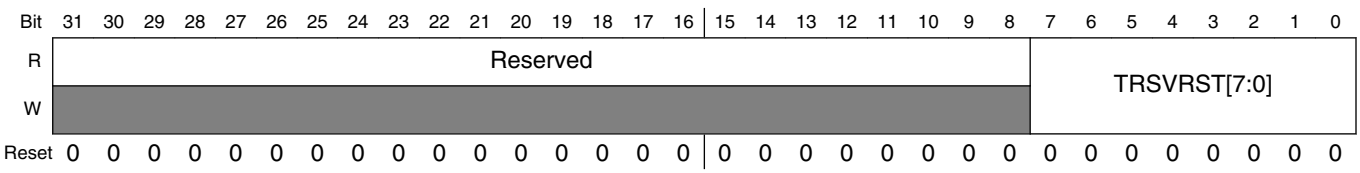


PCIE_PHY_TRSV_REG40 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
7 PHY_TRSV_EN	TRSV Sub-block PD Control 0 Disabled for TRSV block PD control 1 Enabled for TRSV block PD control
6–2 PD_TRAS[6:2]	TRSV Sub-block Powerdown • When Reg40 bit<6> = 1, RX Deserializer power down. • When Reg40 bit<5> = 1, TX Serializer power down. • When Reg40 bit<4> = 1, RXAFE power down. • When Reg40 bit<3> = 1, TX DRVR power down. • When Reg40 bit<2> = 1, RX TERM power down. 0000 Enable
Reserved	This field is reserved.

11.4.4.2.17 PCIE_PHY_TRSV_REG42

Address: 0h base + 168h offset = 168h



PCIE_PHY_TRSV_REG42 field descriptions

Field	Description
31–8 Reserved	This field is reserved.
TRSVRST[7:0]	TRSV Reset Control 0xxxxxxx Normal operation (Default) 10000000 Fine control option for RESET ~ Fine control option for RESET 11111111 Fine control option for RESET

11.5 Ethernet MAC (ENET)

11.5.1 Introduction

The MAC-NET core, in conjunction with a 10/100/1000-Mbit/s MAC, implements layer 3 network acceleration functions. These functions are designed to accelerate the processing of various common networking protocols, such as IP, TCP, UDP, and ICMP, providing wire speed services to client applications.

11.5.2 Overview

The core implements a triple-speed 10/100/1000-Mbit/s Ethernet MAC compliant with the IEEE802.3-2002 standard. The MAC layer provides compatibility with half- or full-duplex 10/100-Mbit/s and full-duplex gigabit Ethernet LANs.

The MAC operation is fully programmable and can be used in Network Interface Card (NIC), bridging, or switching applications. The core implements the remote network monitoring (RMON) counters according to IETF RFC 2819.

The core also implements a hardware acceleration block to optimize the performance of network controllers providing TCP/IP, UDP, and ICMP protocol services. The acceleration block performs critical functions in hardware, which are typically implemented with large software overhead.

The core implements programmable embedded FIFOs that can provide buffering on the receive path for lossless flow control.

Advanced power management features are available with magic packet detection and programmable power-down modes.

A unified DMA (uDMA), internal to the ENET module, optimizes data transfer between the ENET core and the SoC, and supports an enhanced buffer descriptor programming model to support IEEE 1588 functionality.

The programmable Ethernet MAC with IEEE 1588 integrates a standard IEEE 802.3 Ethernet MAC with a time-stamping module. The IEEE 1588 standard provides accurate clock synchronization for distributed control nodes for industrial automation applications.

11.5.2.1 Features

The MAC-NET core includes the following features.

11.5.2.1.1 Ethernet MAC features

- Implements the full 802.3 specification with preamble/SFD generation, frame padding generation, CRC generation and checking
- Supports zero-length preamble
- Dynamically configurable to support 10/100-Mbit/s and gigabit operation
- Supports 10/100 Mbit/s full-duplex and configurable half-duplex operation
- Supports gigabit full-duplex operation
- Compliant with the AMD magic packet detection with interrupt for node remote power management
- Seamless interface to commercial ethernet PHY devices via one of the following:
  - a 4-bit Media Independent Interface (MII) operating at 2.5/25 MHz.
  - a 4-bit non-standard MII-Lite (MII without the CRS and COL signals) operating at 2.5/25 MHz.
  - a 2-bit Reduced MII (RMII) operating at 50 MHz.
  - a (double data rate) 4-bit Reduced GMII (RGMII) operating at 125 MHz.
- Simple 64-Bit FIFO user-application interface
- CRC-32 checking at full speed with optional forwarding of the frame check sequence (FCS) field to the client
- CRC-32 generation and append on transmit or forwarding of user application provided FCS selectable on a per-frame basis
- In full-duplex mode:
  - Implements automated pause frame (802.3 x31A) generation and termination, providing flow control without user application intervention
  - Pause quanta used to form pause frames — dynamically programmable
  - Pause frame generation additionally controllable by user application offering flexible traffic flow control
  - Optional forwarding of received pause frames to the user application
  - Implements standard flow-control mechanism
- In half-duplex mode: provides full collision support, including jamming, backoff, and automatic retransmission
- Supports VLAN-tagged frames according to IEEE 802.1Q
- Programmable MAC address: Insertion on transmit; discards frames with mismatching destination address on receive (except broadcast and pause frames)
- Programmable promiscuous mode support to omit MAC destination address checking on receive
- Multicast and unicast address filtering on receive based on 64-entry hash table, reducing higher layer processing load

- Programmable frame maximum length providing support for any standard or proprietary frame length
- Statistics indicators for frame traffic and errors (alignment, CRC, length) and pause frames providing for IEEE 802.3 basic and mandatory management information database (MIB) package and remote network monitoring (RFC 2819)
- Simple handshake user application FIFO interface with fully programmable depth and threshold levels
- Provides separate status word for each received frame on the user interface providing information such as frame length, frame type, VLAN tag, and error information
- Multiple internal loopback options
- MDIO master interface for PHY device configuration and management supports two programmable MDIO base addresses, and standard (IEEE 802.3 Clause 22) and extended (Clause 45) MDIO frame formats
- Supports legacy FEC buffer descriptors
- Interrupt coalescing reduces the number of interrupts generated by the MAC, reducing CPU loading
- Traffic-shaping bandwidth distribution supports credit-based and round-robin-based policies. Either policy can be combined with time-based shaping.
- AVB (Audio Video Bridging, IEEE 802.1Qav) features:
  - Credit-based bandwidth distribution policy can be combined with time-based shaping
  - AVB endpoint talker and listener support
  - Support for arbitration between different priority traffic (for example, AVB class A, AVB class B, and non-AVB)

11.5.2.1.2 IP protocol performance optimization features

- Operates on TCP/IP and UDP/IP and ICMP/IP protocol data or IP header only
- Enables wire-speed processing
- Supports IPv4 and IPv6
- Transparent passing of frames of other types and protocols
- Supports VLAN tagged frames according to IEEE 802.1q with transparent forwarding of VLAN tag and control field
- Automatic IP-header and payload (protocol specific) checksum calculation and verification on receive
- Automatic IP-header and payload (protocol specific) checksum generation and automatic insertion on transmit configurable on a per-frame basis
- Supports IP and TCP, UDP, ICMP data for checksum generation and checking

- Supports full header options for IPv4 and TCP protocol headers
- Provides IPv6 support to datagrams with base header only — datagrams with extension headers are passed transparently unmodifed/unchecked
- Provides statistics information for received IP and protocol errors
- Configurable automatic discard of erroneous frames
- Configurable automatic host-to-network (RX) and network-to-host (TX) byte order conversion for IP and TCP/UDP/ICMP headers within the frame
- Configurable padding remove for short IP datagrams on receive
- Configurable Ethernet payload alignment to allow for 32-bit word-aligned header and payload processing
- Programmable store-and-forward operation with clock and rate decoupling FIFOs

11.5.2.1.3 IEEE 1588 features

- Supports all IEEE 1588 frames.
- Allows reference clock to be chosen independently of network speed.
- Software-programmable precise time-stamping of ingress and egress frames
- Timer monitoring capabilities for system calibration and timing accuracy management
- Precise time-stamping of external events with programmable interrupt generation
- Programmable event and interrupt generation for external system control
- Supports hardware- and software-controllable timer synchronization.
- Provides a 4-channel IEEE 1588 timer. Each channel supports input capture and output compare using the 1588 counter.

11.5.2.2 Block diagram

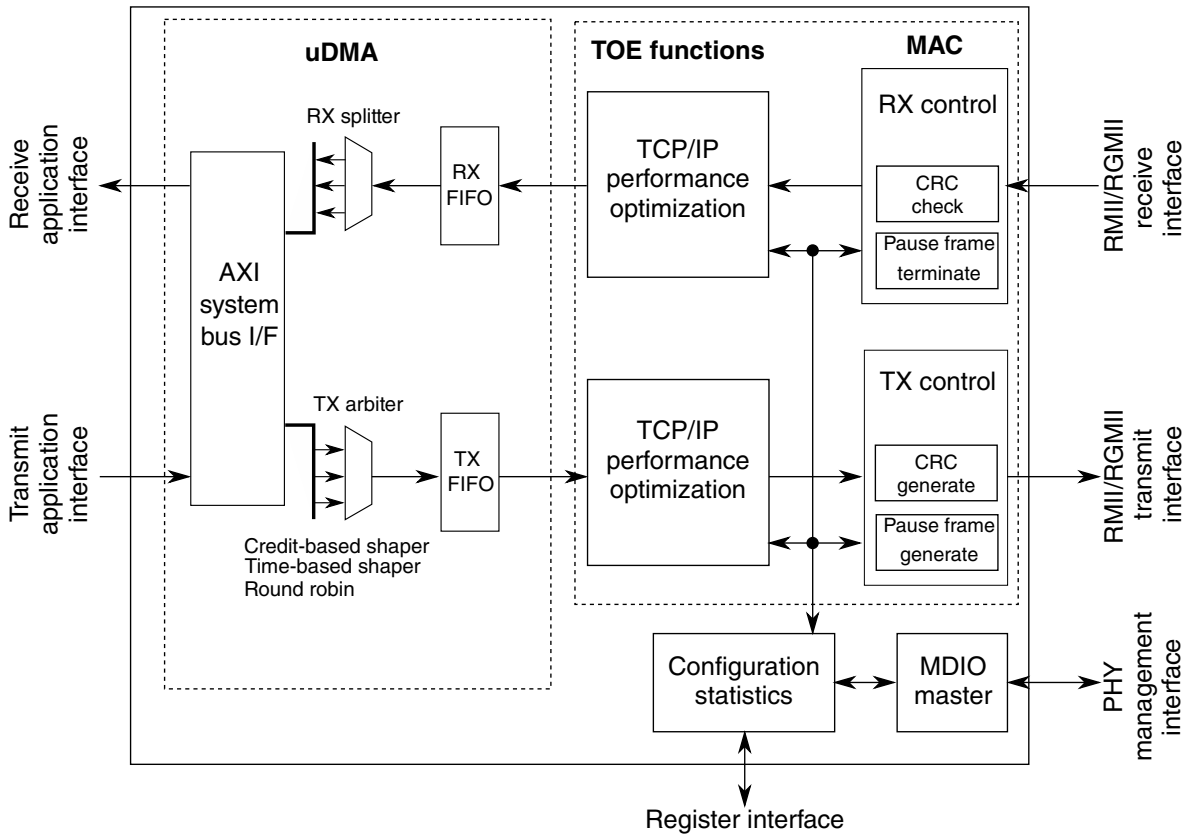


Figure 11-79. Ethernet MAC-NET core block diagram

11.5.3 External Signals and Clocks

The table found here describes the external signals of ENET.

Table 11-44. ENET External Signals

Signal	Description	Mode	Pad Name	Alt Mode	Direction
ENET1_MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	RMII/RGMII	ENET_MDC	ALT0	O
			GPIO1_IO06	ALT1	
			I2C1_SCL	ALT1	
ENET1_MDIO	Transfers control information between the external PHY and the media-access controller. Data is synchronous to MDC. This signal is an input after reset.	RMII/RGMII	ENET_MDIO	ALT0	I/O
			GPIO1_IO07	ALT1	
			I2C1_SDA	ALT1	
RGMII_TD3	Serial output Ethernet data. Only valid during TX_EN assertion. Only used for RGMII.	RGMII	ENET_TD3	ALT0	O
RGMII_TD2	Used as RGMII data.	RGMII	ENET_TD2	ALT0	O

Table continues on the next page...

Table 11-44. ENET External Signals (continued)

Signal	Description	Mode	Pad Name	Alt Mode	Direction
ENET1_TX_CLK	Used as RMII clock. There are two RGMII clock schemes. 1. MAC generates output 50M reference clock for PHY, and MAC also uses this 50M clock. 2. MAC uses external 50M clock.	RMII	ENET_TD2	ALT1	I/O
RGMII_TD1	Serial output Ethernet data. Only valid during TX_EN assertion.	RMII/RGMII	ENET_TD1	ALT0	O
RGMII_TD0	Serial output Ethernet data. Only valid during TX_EN assertion.	RMII/RGMII	ENET_TD0	ALT0	O
RGMII_TX_CTL	In RGMII mode, contains TX_EN on the rising edge of RGMII_TXC, and a logical derivative of TX_EN and TX_ER (TX_EN XOR TX_ER) on the falling edge of RGMII_TXC. (RMII_TX_EN).	RMII/RGMII	ENET_TX_CTL	ALT0	O
RGMII_TXC	For RGMII, ENET_TXC works as RGMII_TXC in ALT0 mode.	RGMII	ENET_TXC	ALT0	O
ENET1_TX_ER	For RMII, ENET_TXC works as ENET1_TX_ER in ALT1 mode.	RMII	ENET_TXC	ALT1	O
RGMII_RX_CTL	In RGMII mode, contains RX_EN on the rising edge of RGMII_RXC, and a logical derivative of RX_EN and RX_ER (RX_EN XOR RX_ER) on the falling edge of RGMII_RXC. (RMII_RX_EN (CRS_DV)).	RMII/RGMII	ENET_RX_CTL	ALT0	I
RGMII_RXC	For RGMII, ENET_RXC works as RGMII_RXC in ALT0 mode.	RGMII	ENET_RXC	ALT0	I
ENET1_RX_ER	For RMII, ENET_RXC works as ENET1_RX_ER in ALT1 mode.	RMII	ENET_RXC	ALT1	I
RGMII_RD0	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RMII/RGMII	ENET_RD0	ALT0	I
RGMII_RD1	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RMII/RGMII	ENET_RD1	ALT0	I
RGMII_RD2	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	ENET_RD2	ALT0	I
RGMII_RD3	Contains the Ethernet input data transferred from the PHY to the media-access controller when RX_EN is asserted.	RGMII	ENET_RD3	ALT0	I
ENET1_1588_EVENT0_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is	RMII/RGMII	GPIO1_IO08	ALT1	I

Table continues on the next page...

Table 11-44. ENET External Signals (continued)

Signal	Description	Mode	Pad Name	Alt Mode	Direction
	latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.				
ENET1_1588_EVENT0_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set. Note: ENET_1588_EVENT0_OUT has a programmable output width, see IOMUXC_GPR0[CLK_STRETCH], delayed one clock cycle in relation to all other EVENTx_OUT signals.	RMII/RGMII	GPIO1_IO09	ALT1	O
ENET1_1588_EVENT1_IN	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.	RMII/RGMII	I2C2_SCL	ALT1	I
ENET1_1588_EVENT1_OUT	Capture/compare block input/output event bus signal. When configured for capture and a rising edge is	RMII/RGMII	I2C2_SDA	ALT1	O

Table continues on the next page...

Table 11-44. ENET External Signals (continued)

Signal	Description	Mode	Pad Name	Alt Mode	Direction
	detected, the current timer value is latched and transferred into the corresponding ENET_TCCRn register for inspection by software. When configured for compare, the corresponding signal 1588_EVENT is asserted for one cycle when the timer reaches the compare value programmed in register ENET_TCCRn. An interrupt or DMA request can be triggered if the corresponding bit in ENET_TCSRn[TIE] or ENET_TCSRn[TDRE] is set.				
ENET_PHY_REF_CLK_ROOT	Reference clock for PHY.	RGMII	GPIO1_IO00	ALT1	O

The table below describes the clock sources for ENET. See Clock Controller Module (CCM) for clock setting, configuration, and gating information.

Table 11-45. ENET Clocks

Clock name	Clock Root	Description
ipp_ind_mac0_txclk	ENET_REF_CLK_ROOT	rmii/rgmii interface clock
ipg_clk_time	ENET_TIMER_CLK_ROOT	Time stamp clock

11.5.4 External signal description

NOTE

The MII column pertains only to devices that support MII.

MII	RMII	RGMII	Description	I/O
MII_COL	—	—	Asserted upon detection of a collision and remains asserted while the collision persists. This signal is not defined for full-duplex mode.	I
MII_CRS	—	—	Carrier sense. When asserted, indicates transmit or receive medium is not idle.	I

Table continues on the next page...

MII	RMII	RGMII	Description	I/O
			In RMII mode, this signal is present on the RMII_CRS_DV pin.	
MII_MDC	RMII_MDC	RGMII_MDC	Output clock provides a timing reference to the PHY for data transfers on the MDIO signal.	O
MII_MDIO	RMII_MDIO	RGMII_MDIO	Transfers control information between the external PHY and the media-access controller. Data is synchronous to MDC. This signal is an input after reset.	I/O
MII_RXCLK	—	RGMII_RXC	In MII mode, provides a timing reference for RXDV, RXD[3:0], and RXER. In RGMII mode, provides a timing reference for RGMII_RXD[3:0] and RGMII_RX_CTL.	I
MII_RXDV	RMII_CRS_DV	RGMII_RX_CTL	Asserting this input indicates the PHY has valid nibbles present on the MII. RXDV must remain asserted from the first recovered nibble of the frame through to the last nibble. Asserting RXDV must start no later than the SFD and exclude any EOF. In RMII mode, this pin also generates the CRS signal. In RGMII mode, contains RXDV on the rising edge of RGMII_RXC, and a logical derivative of RXDV and RXERR (RXDV XOR RXERR) on the falling edge of RGMII_RXC.	I
MII_RXD[3:0]	RMII_RXD[1:0]	RGMII_RXD[3:0]	Contains the Ethernet input data transferred from the PHY to the media-access controller when RXDV is asserted.	I

Table continues on the next page...

MII	RMII	RGMII	Description	I/O
MII_RXER	RMII_RXER	—	When asserted with RXDV, indicates the PHY detects an error in the current frame.	I
MII_TXCLK	—	—	Input clock, which provides a timing reference for TXEN, TXD[3:0], and TXER.	I
—	—	RGMII_TXC	Provides a timing reference for RGMII_TXD[3:0] and RGMII_TX_CTL. **NOTE:** This is an output signal.	O
MII_TXD[3:0]	RMII_TXD[1:0]	RGMII_TXD[3:0]	Serial output Ethernet data. Only valid during TXEN assertion.	O
MII_TXEN	RMII_TXEN	RGMII_TX_CTL	Indicates when valid nibbles are present on the MII. This signal is asserted with the first nibble of a preamble and is deasserted before the first TXCLK following the final nibble of the frame. In RGMII mode, contains TXEN on the rising edge of RGMII_TXC, and a logical derivative of TXEN and TXERR (TXEN XOR TXERR) on the falling edge of RGMII_TXC.	O
MII_TXER	—	—	When asserted for one or more clock cycles while TXEN is also asserted, PHY sends one or more illegal symbols.	O
—	RMII_REF_CLK	—	In RMII mode, this signal is the reference clock for receive, transmit, and the control interface.	I
1588_TMR*n*	1588_TMR*n*	1588_TMR*n*	Capture/Compare block input/output event bus. When configured for capture and a rising edge is detected, the current timer value is	I/O

Table continues on the next page...

MII	RMII	RGMII	Description	I/O
			latched and transferred into the corresponding ENET_TCCR*n* register for inspection by software. When configured for compare, the corresponding signal 1588_TMR*n* is asserted for one cycle when the timer reaches the compare value programmed in ENET_TCCR*n*. An interrupt can be triggered if ENET_TCSR*n*[TIE] is set. A DMA request can be triggered if ENET_TCSR*n*[TDRE] is set.	
ENET_1588_CLKIN	ENET_1588_CLKIN	—	Alternate IEEE 1588 Ethernet clock input; Clock period should be an integer number of nanoseconds	I

11.5.4.1 RMII Interface Mapping

The RMII interface maps as follows:

Table 11-46. RMII interface Mapping

Ball Name	RMII mode
ENETx_MDC	MDC
ENETx_MDIO	MDIO
ENETx_REFCLK_125M_25M	-
ENETx_RGMII_TXC	REF_CLK
ENETx_RGMII_TX_CTL	TX_EN
ENETx_RGMII_TXD0	TXD0
ENETx_RGMII_TXD1	TXD1
ENETx_RGMII_TXD2	-
ENETx_RGMII_TXD3	-
ENETx_RGMII_RXC	-
ENETx_RGMII_RX_CTL	CRS_DV
ENETx_RGMII_RXD0	RXD0

Table continues on the next page...

Table 11-46. RMII interface Mapping (continued)

Ball Name	RMII mode
ENETx_RGMII_RXD1	RXD1
ENETx_RGMII_RXD2	RX_ER
ENETx_RGMII_RXD3	-

11.5.5 Functional description

This section provides a complete functional description of the MAC-NET core.

11.5.5.1 Ethernet MAC frame formats

The IEEE 802.3 standard defines the Ethernet frame format as follows:

- Minimum length of 64 bytes
- Maximum length of 1518 bytes excluding the preamble and the start frame delimiter (SFD) bytes

An Ethernet frame consists of the following fields:

- Seven bytes preamble
- Start frame delimiter (SFD)
- Two address fields
- Length or type field
- Data field
- Frame check sequence (CRC value)
- Extension field is defined only for Gigabit Ethernet half-duplex implementations and is not supported by the MAC core

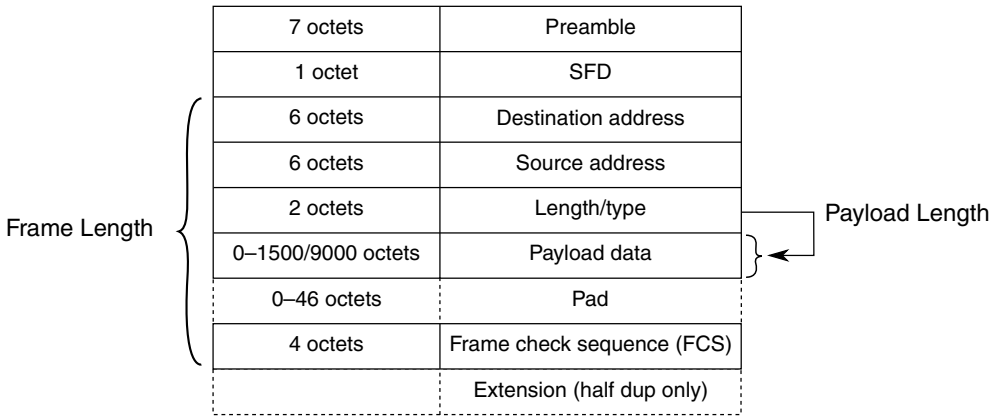


Figure 11-80. MAC frame format overview

Optionally, MAC frames can be VLAN-tagged with an additional four-byte field inserted between the MAC source address and the type/length field. VLAN tagging is defined by the IEEE P802.1q specification. VLAN-tagged frames have a maximum length of 1522 bytes, excluding the preamble and the SFD bytes.

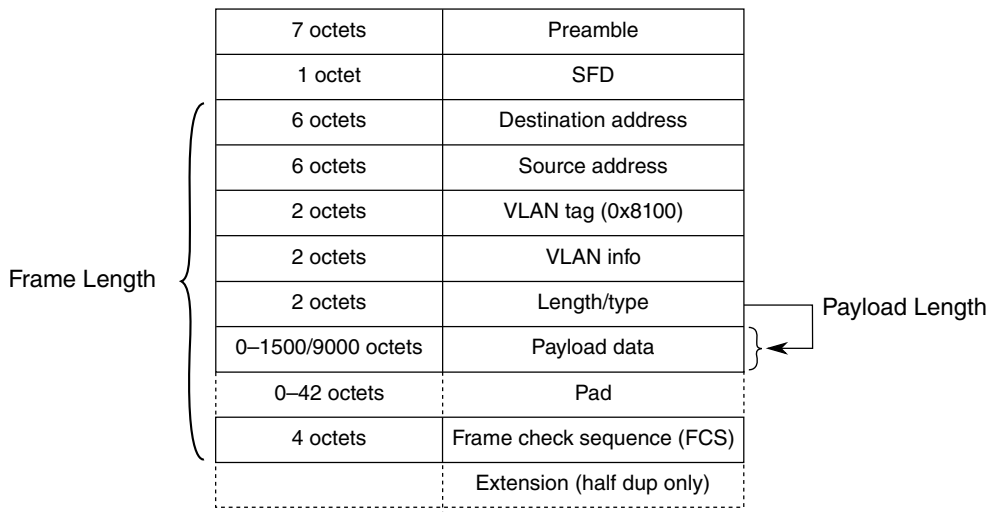


Figure 11-81. VLAN-tagged MAC frame format overview

Table 11-47. MAC frame definition

Term	Description
Frame length	Defines the length, in octets, of the complete frame without preamble and SFD. A frame has a valid length if it contains at least 64 octets and does not exceed the programmed maximum length.
Payload length	The length/type field indicates the length of the frame's payload section. The most significant byte is sent/received first. • If the length/type field is set to a value less than 46, the payload is padded so that the minimum frame length requirement (64 bytes) is met. For VLAN-tagged frames, a value less than 42 indicates a padded frame. • If the length/type field is set to a value larger than the programmed frame maximum length (e.g. 1518) it is interpreted as a type field.

Table continues on the next page...

Table 11-47. MAC frame definition (continued)

Term	Description
Destination and source address	48-bit MAC addresses. The least significant byte is sent/received first and the first two least significant bits of the MAC address distinguish MAC frames, as detailed in MAC address check.

Note

Although the IEEE specification defines a maximum frame length, the MAC core provides the flexibility to program any value for the frame maximum length.

11.5.5.1.1 Pause Frames

The receiving device generates a pause frame to indicate a congestion to the emitting device, which should stop sending data.

Pause frames are indicated by the length/type set to 0x8808. The two first bytes of a pause frame following the type, defines a 16-bit opcode field set to 0x0001 always. A 16-bit pause quanta is defined in the frame payload bytes 2 (P1) and 3 (P2) as defined in the following table. The P1 pause quanta byte is the most significant.

Table 11-48. Pause Frame Format (Values in Hex)

1	2	3	4	5	6	7	8	9	10	11	12	13	14
55	55	55	55	55	55	55	D5	01	80	C2	00	00	01
Preamble							SFD	Multicast Destination Address					
15	16	17	18	19	20	21	22	23	24	25	26	27 –68	
00	00	00	00	00	00	88	08	00	01	hi	lo	00	
Source Address						Type		Opcode		P1	P2	pad (42)	
69	70	71	72										
26	6B	AE	0A										
CRC-32													

There is no payload length field found within a pause frame and a pause frame is always padded with 42 bytes (0x00).

If a pause frame with a pause value greater than zero (XOFF condition) is received, the MAC stops transmitting data as soon the current frame transfer is completed. The MAC stops transmitting data for the value defined in pause quanta. One pause quanta fraction refers to 512 bit times.

If a pause frame with a pause value of zero (XON condition) is received, the transmitter is allowed to send data immediately (see Full-duplex flow control operation for details).

11.5.5.1.2 Magic packets

A magic packet is a unicast, multicast, or broadcast packet, which carries a defined sequence in the payload section.

Magic packets are received and inspected only under specific conditions as described in Magic packet detection.

The defined sequence to decode a magic packet is formed with a synchronization stream which consists of six consecutive 0xFF bytes, and is followed by sequence of sixteen consecutive unicast MAC addresses of the node to be awakened.

This sequence can be located anywhere in the magic packet payload. The magic packet is formed with a standard Ethernet header, optional padding, and CRC.

11.5.5.2 IP and higher layers frame format

The following sections use the term datagram to describe the protocol specific data unit that is found within the payload section of its container entity.

For example, an IP datagram specifies the payload section of an Ethernet frame. A TCP datagram specifies the payload section within an IP datagram.

11.5.5.2.1 Ethernet types

IP datagrams are carried in the payload section of an Ethernet frame. The Ethernet frame type/length field discriminates several datagram types.

The following table lists the types of interest:

Table 11-49. Ethernet type value examples

Type	Description
0x8100	VLAN-tagged frame. The actual type is found 4 octets later in the frame.
0x0800	IPv4
0x0806	ARP
0x86DD	IPv6

11.5.5.2.2 IPv4 datagram format

The following figure shows the IP Version 4 (IPv4) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words. The first byte sent/received is the leftmost byte of the first word (in other words, version/IHL field).

The IP header can contain further options, which are always padded if necessary to guarantee the payload following the header is aligned to a 32-bit boundary.

The IP header is immediately followed by the payload, which can contain further protocol headers (for example, TCP or UDP, as indicated by the protocol field value). The complete IP datagram is transported in the payload section of an Ethernet frame.

Table 11-50. IPv4 header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Version	IHL	TOS		Length			
Fragment ID				Flags	Fragment offset		
TTL		Protocol		Header checksum			
Source address							
Destination address							
Options							

Table 11-51. IPv4 header fields

Field name	Description
Version	4-bit IP version information. 0x4 for IPv4 frames.
IHL	4-bit Internet header length information. Determines number of 32-bit words found within the IP header. If no options are present, the default value is 0x5.
TOS	Type of service/DiffServ field.
Length	Total length of the datagram in bytes, including all octets of header and payload.
Fragment ID, flags, fragment offset	Fields used for IP fragmentation.
TTL	Time-to-live. In effect, is decremented at each router arrival. If zero, datagram must be discarded.
Protocol	Identifier of protocol that follows in the datagram.
Header checksum	Checksum of IP header. For computational purposes, this field's value is zero.
Source address	Source IP address.
Destination address	Destination IP address.

11.5.5.2.3 IPv6 datagram format

The following figure shows the IP version 6 (IPv6) header, which is located at the beginning of an IP datagram. It is organized in 32-bit words and has a fixed length of ten words (40 bytes). The next header field identifies the type of the header that follows the IPv6 header. It is defined similar to the protocol identifier within IPv4, with new definitions for identifying extension headers. These headers can be inserted between the IPv6 header and the protocol header, which will shift the protocol header accordingly.The accelerator currently only supports IPv6 without extension headers (in other words, the next header specifies TCP, UDP, or IMCP).

The first byte sent/received is the leftmost byte of the first word (in other words, version/ traffic class fields).

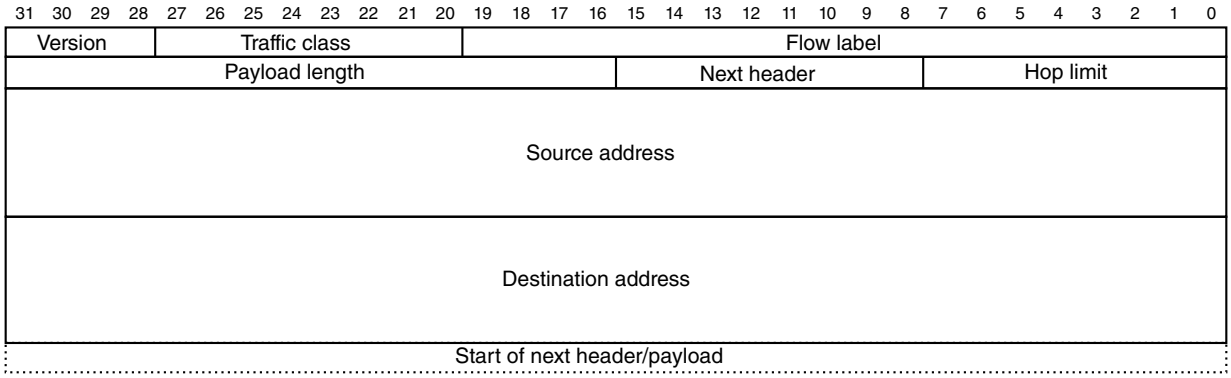


Figure 11-82. IPv6 header format

Table 11-52. IPv6 header fields

Field name	Description
Version	4-bit IP version information. 0x6 for all IPv6 frames.
Traffic class	8-bit field defining the traffic class.
Flow label	20-bit flow label identifying frames of the same flow.
Payload length	16-bit length of the datagram payload in bytes. It includes all octets following the IPv6 header.
Next header	Identifies the header that follows the IPv6 header. This can be the protocol header or any IPv6 defined extension header.
Hop limit	Hop counter, decremented by one by each station that forwards the frame. If hop limit is 0 the frame must be discarded.
Source address	128-bit IPv6 source address.
Destination address	128-bit IPv6 destination address.

11.5.5.2.4 Internet Control Message Protocol (ICMP) datagram format

An internet control message protocol (ICMP) is found following the IP header, if the protocol identifier is 1. The ICMP datagram has a four-octet header followed by additional message data.

Table 11-53. ICMP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Type		Code		Checksum			
ICMP message data							

Table 11-54. IP header fields

Field name	Description
Type	8-bit type information
Code	8-bit code that is related to the message type
Checksum	16-bit one's complement checksum over the complete ICMP datagram

11.5.5.2.5 User Datagram Protocol (UDP) datagram format

A user datagram protocol header is found after the IP header, when the protocol identifier is 17.

The payload of the datagram is after the UDP header. The header byte order follows the conventions given for the IP header above.

Table 11-55. UDP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Length				Checksum			

Table 11-56. UDP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Length	Length of user data which immediately follows the header, including the UDP header (that is, minimum value is 8)
Checksum	Checksum over the complete datagram and some IP header information

11.5.5.2.6 TCP datagram format

A TCP header is found following the IP header, when the protocol identifier has a value of 6.

The TCP payload immediately follows the TCP header.

Table 11-57. TCP header format

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source port				Destination port			
Sequence number							
Acknowledgement number							

Table continues on the next page...

Table 11-57. TCP header format (continued)

Offset	Reserved	Flags	Window	
Checksum			Urgent pointer	
Options				

Table 11-58. TCP header fields

Field name	Description
Source port	Source application port
Destination port	Destination application port
Sequence number	Transmit sequence number
Ack. number	Receive sequence number
Offset	Data offset, which is number of 32-bit words within TCP header — if no options selected, defaults to value of 5
Flags	URG, ACK, PSH, RST, SYN, FIN flags
Window	TCP receive window size information
Checksum	Checksum over the complete datagram (TCP header and data) and IP header information
Options	Additional 32-bit words for protocol options

11.5.5.3 IEEE 1588 message formats

The following sections describe the IEEE 1588 message formats.

11.5.5.3.1 Transport encapsulation

The precision time protocol (PTP) datagrams are encapsulated in Ethernet frames using the UDP/IP transport mechanism, or optionally, with the newer 1588v2 directly in Ethernet frames (layer 2).

Typically, multicast addresses are used to allow efficient distribution of the synchronization messages.

11.5.5.3.1.1 UDP/IP

The 1588 messages (v1 and v2) can be transported using UDP/IP multicast messages.

Table 11-59 shows IP multicast groups defined for PTP. The table also shows their respective MAC layer multicast address mapping according to RFC 1112 (last three octets of IP follow the fixed value of 01-00-5E).

Table 11-59. UDP/IP multicast domains

Name	IP Address	MAC Address mapping
DefaultPTPdomain	224.0.1.129	01-00-5E-00-01-81
AlternatePTPdomain1	224.0.1.130	01-00-5E-00-01-82
AlternatePTPdomain2	224.0.1.131	01-00-5E-00-01-83
AlternatePTPdomain3	224.0.1.132	01-00-5E-00-01-84

Table 11-60. UDP port numbers

Message type	UDP port	Note
Event	319	Used for SYNC and DELAY_REQUEST messages
General	320	All other messages (for example, follow-up, delay-response)

11.5.5.3.1.2 Native Ethernet (PTPv2)

In addition to using UDP/IP frames, IEEE 1588v2 defines a native Ethernet frame format that uses ethertype = 0x88F7. The payload of the Ethernet frame immediately contains the PTP datagram, starting with the PTPv2 header.

Besides others, version 2 adds a peer delay mechanism to allow delay measurements between individual point-to-point links along a path over multiple nodes. The following multicast domains are also defined in PTPv2.

Table 11-61. PTPv2 multicast domains

Name	MAC address
Normal messages	01-1B-19-00-00-00
Peer delay messages	01-80-C2-00-00-0E

11.5.5.3.2 PTP header

All PTP frames contain a common header that determines the protocol version and the type of message, which defines the remaining content of the message.

All multi-octet fields are transmitted in big-endian order (the most significant byte is transmitted/received first).

The last four bits of versionPTP are at the same position (second byte) for PTPv1 and PTPv2 headers. This allows accurate identification by inspecting the first two bytes of the message.

11.5.5.3.2.1 PTPv1 header

Table 11-62. Common PTPv1 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	2	versionPTP = 0x0001							
2	2	versionNetwork							
4	16	subdomain							
20	1	messageType							
21	1	sourceCommunicationTechnology							
22	6	sourceUuid							
28	2	sourcePortId							
30	2	sequenceId							
32	1	control							
33	1	0x00							
34	2	flags							
36	4	reserved							

The type of message is encoded in the messageType and control fields as shown in Table 11-63 :

Table 11-63. PTPv1 message type identification

messageType	control	Message Name	Message
0x01	0x0	SYNC	Event message
0x01	0x1	DELAY_REQ	Event message
0x02	0x2	FOLLOW_UP	General message
0x02	0x3	DELAY_RESP	General message
0x02	0x4	MANAGEMENT	General message
other	other	—	Reserved

The field sequenceId is used to non-ambiguously identify a message.

11.5.5.3.2.2 PTPv2 header

Table 11-64. Common PTPv2 message header

Offset	Octets	Bits							
		7	6	5	4	3	2	1	0
0	1	transportSpecific				messageId			
1	1	reserved				versionPTP = 0x2			
2	2	messageLength							
4	1	domainNumber							
5	1	reserved							
6	2	flags							
8	8	correctionField							
16	4	reserved							
20	10	sourcePortIdentity							
30	2	sequenceId							
32	1	control							
33	1	logMeanMessageInterval							

The type of message is encoded in the field messageId as follows:

Table 11-65. PTPv2 message type identification

messageId	Message name	Message
0x0	SYNC	Event message
0x1	DELAY_REQ	Event message
0x2	PATH_DELAY_REQ	Event message
0x3	PATH_DELAY_RESP	Event message
0x4–0x7	—	Reserved
0x8	FOLLOW_UP	General message
0x9	DELAY_RESP	General message
0xa	PATH_DELAY_FOLLOW_UP	General message
0xb	ANNOUNCE	General message
0xc	SIGNALING	General message
0xd	MANAGEMENT	General message

The PTPv2 flags field contains further details on the type of message, especially if one-step or two-step implementations are used. The one- or two-step implementation is controlled by the TWO_STEP bit in the first octet of the flags field as shown below. Reserved bits are cleared.

Table 11-66. PTPv2 message flags field definitions

Bit	Name	Description
0	ALTERNATE_MASTER	See IEEE 1588 Clause 17.4
1	TWO_STEP	1 Two-step clock 0 One-step clock
2	UNICAST	1 Transport layer address uses a unicast destination address 0 Multicast is used
3	—	Reserved
4	—	Reserved
5	Profile specific	
6	Profile specific	
7	—	Reserved

11.5.5.4 MAC receive

The MAC receive engine performs the following tasks:

- Check frame framing
- Remove frame preamble and frame SFD field
- Discard frame based on frame destination address field
- Terminate pause frames
- Check frame length
- Remove payload padding if it exists
- Calculate and verify CRC-32
- Write received frames in the core receive FIFO

If the MAC is programmed to operate in half-duplex mode, it will also check if the frame is received with a collision.

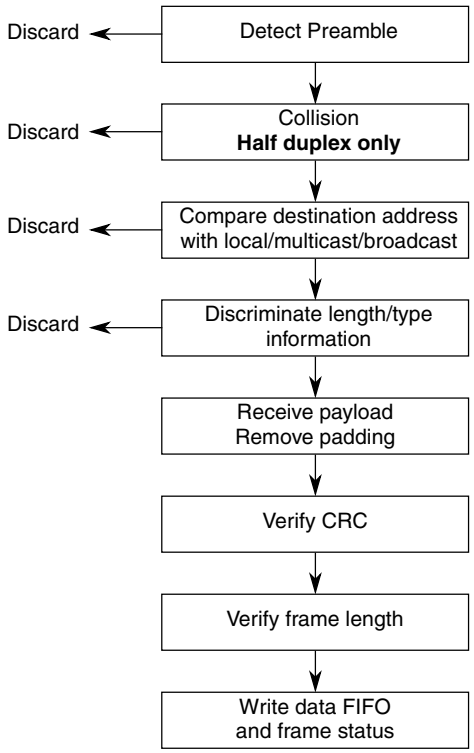


Figure 11-83. MAC receive flow

11.5.5.4.1 Collision detection in half-duplex mode

If the packet is received with a collision detected during reception of the first 64 bytes, the packet is discarded (if frame size was less than ~14 octets) or transmitted to the user application with an error and RxBD[CE] set.

11.5.5.4.2 Preamble processing

The IEEE 802.3 standard allows a maximum size of 56 bits (seven bytes) for the preamble, while the MAC core allows any preamble length, including zero length preamble.

The MAC core checks for the start frame delimiter (SFD) byte. If the next byte of the preamble, which is different from 0x55, is not 0xD5, the frame is discarded.

Although the IEEE specification dictates that the inner-packet gap should be at least 96 bits, the MAC core is designed to accept frames separated by only 64 10/100-Mbit/s operation (MII) bits.

The MAC core removes the preamble and SFD bytes.

11.5.5.4.3 MAC address check

The destination address bit 0 differentiates between multicast and unicast addresses.

- If bit 0 is 0, the MAC address is an individual (unicast) address.
- If bit 0 is 1, the MAC address defines a group (multicast) address.
- If all 48 bits of the MAC address are set, it indicates a broadcast address.

11.5.5.4.3.1 Unicast address check

If a unicast address is received, the destination MAC address is compared to the node MAC address programmed by the host in the PADDR1/2 registers.

If the destination address matches any of the programmed MAC addresses, the frame is accepted.

If Promiscuous mode is enabled (RCR[PROM] = 1) no address checking is performed and all unicast frames are accepted.

11.5.5.4.3.2 Multicast and unicast address resolution

The hash table algorithm used in the group and individual hash filtering operates as follows.

- The 48-bit destination address is mapped into one of 64 bits, represented by 64 bits in ENET*n*_GAUR/GALR (group address hash match) or ENET*n*_IAUR/IALR (individual address hash match).
- This mapping is performed by passing the 48-bit address through the on-chip 32-bit CRC generator and selecting the six most significant bits of the CRC-encoded result to generate a number between 0 and 63.
- The msb of the CRC result selects ENET*n*_GAUR (msb = 1) or ENET*n*_GALR (msb = 0).
- The five lsbs of the hash result select the bit within the selected register.
- If the CRC generator selects a bit set in the hash table, the frame is accepted; else, it is rejected.

For example, if eight group addresses are stored in the hash table and random group addresses are received, the hash table prevents roughly 56/64 (or 87.5%) of the group address frames from reaching memory. Those that do reach memory must be further filtered by the processor to determine if they truly contain one of the eight desired addresses.

The effectiveness of the hash table declines as the number of addresses increases.

The user must initialize the hash table registers. Use this CRC32 polynomial to compute the hash:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

If Promiscuous mode is enabled (ENET*n*_RCR[PROM] = 1) all unicast and multicast frames are accepted regardless of ENET*n*_GAUR/GALR and ENET*n*_IAUR/IALR settings.

11.5.5.4.3.3 Broadcast address reject

All broadcast frames are accepted if BC_REJ is cleared or ENET*n*_RCR[PROM] is set. If PROM is cleared when ENET*n*_RCR[BC_REJ] is set, all broadcast frames are rejected.

Table 11-67. Broadcast address reject programming

PROM	BC_REJ	Broadcast frames
0	0	Accepted
0	1	Rejected
1	0	Accepted
1	1	Accepted

11.5.5.4.3.4 Miss-bit implementation

For higher layer filtering purposes, RxBD[M] indicates an address miss when the MAC operates in promiscuous mode and accepts a frame that would otherwise be rejected.

If a group/individual hash or exact match does not occur and Promiscuous mode is enabled (RCR[PROM] = 1), the frame is accepted and the M bit is set in the buffer descriptor; otherwise, the frame is rejected.

This means the status bit is set in any of the following conditions during Promiscuous mode:

- A broadcast frame is received when BC_REJ is set
- A unicast is received that does not match either:

- Node address (PALR[PADDR1] and PAUR[PADDR2])
- Hash table for unicast (IAUR[IADDR1] and IALR[IADDR2])

- A multicast is received that does not match the GAUR[GADDR1] and GALR[GADDR2] hash table entries

11.5.5.4.4 Frame length/type verification: payload length check

If the length/type is less than 0x600 and NLC is set, the MAC checks the payload length and reports any error in the frame status word and interrupt bit PLR.

If the length/type is greater than or equal to 0x600, the MAC interprets the field as a type and no payload length check is performed.

The length check is performed on VLAN and stacked VLAN frames. If a padded frame is received, no length check can be performed due to the extended frame payload because padded frames can never have a payload length error.

11.5.5.4.5 Frame length/type verification: frame length check

When the receive frame length exceeds MAX_FL bytes, the BABR interrupt is generated and the RxBD[LG] bit is set.

The frame is not truncated unless the frame length exceeds the value programmed in ENET*n*_FTRL[TRUNC_FL]. If the frame is truncated, RxBD[TR] is set. In addition, a truncated frame always has the CRC error indication set (RxBD[CR]).

11.5.5.4.6 VLAN frames processing

VLAN frames have a length/type field set to 0x8100 immediately followed by a 16-Bit VLAN control information field.

VLAN-tagged frames are received as normal frames because the VLAN tag is not interpreted by the MAC function, and are pushed complete with the VLAN tag to the user application. If the length/type field of the VLAN-tagged frame, which is found four octets later in the frame, is less than 42, the padding is removed. In addition, the frame status word (RxBD[NO]) indicates that the current frame is VLAN tagged.

11.5.5.4.7 Pause frame termination

The receive engine terminates pause frames and does not transfer them to the receive FIFO. The quanta is extracted and sent to the MAC transmit path via a small internal clock rate decoupling asynchronous FIFO.

The quanta is written only if a correct CRC and frame length are detected by the control state machine. If not, the quanta is discarded and the MAC transmit path is not paused.

Good pause frames are ignored if ENET*n*_RCR[FCE] is cleared and are forwarded to the client interface when ENET*n*_RCR[PAUFWD] is set.

11.5.5.4.8 CRC check

The CRC-32 field is checked and forwarded to the core FIFO interface if ENET*n*_RCR[CRCFWD] is cleared and ENET*n*_RCR[PADEN] is cleared. When CRCFWD is set (regardless of PADEN), the CRC-32 field is checked and terminated (not transmitted to the FIFO).

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the frame check sequence (FCS) field with the x^{31} term as right-most bit of the first octet. The CRC bits are thus received in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

If a CRC error is detected, the frame is marked invalid and RxBD[CR] is set.

11.5.5.4.9 Frame padding removal

When a frame is received with a payload length field set to less than 46 (42 for VLAN-tagged frames and 38 for frames with stacked VLANs), the zero padding can be removed before the frame is written into the data FIFO depending on the setting of ENET*n*_RCR[PADEN].

Note

If a frame is received with excess padding (in other words, the length field is set as mentioned above, but the frame has more than 64 octets) and padding removal is enabled, then the padding is removed as normal and no error is reported if the frame is otherwise correct (for example: good CRC, less than maximum length, and no other error).

11.5.5.4.10 Frame classification (AVB)

To support protocols such as Audio Video Bridging (AVB, IEEE 802.1Qav), normal traffic is separated from time-sensitive traffic immediately at the application interface to allow storing them in different queues for further processing. For every frame received, a

classification based on VLAN priority can be performed. When a frame is received, and it contains a VLAN tag, the priority is compared against the values set in the classification match registers (see Receive Classification Match Register for Class n (ENET_RCMR*n*)) and the following figure.

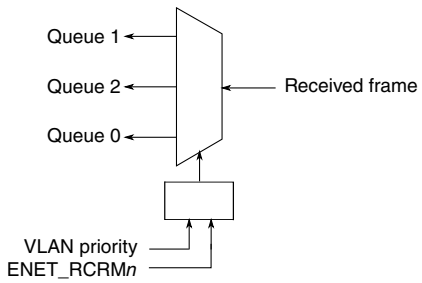


Figure 11-84. AVB frame classification

11.5.5.4.11 Receive flushing

RX flushing prevents frames in the RX FIFO from being blocked. Blocking can occur if the frame at the head of the RX FIFO cannot be forwarded because the ring it is associated with cannot accept it. This situation occurs when either the ring's RxBD[EMPTY] is not set or ENET_RDAR*n* is not set. When RX flushing is enabled, via QOS Scheme (ENET_QOS), the blocking frame will be flushed (discarded).

11.5.5.5 MAC transmit

Frame transmission starts when the transmit FIFO holds enough data.

After a transfer starts, the MAC transmit function performs the following tasks:

- Generates preamble and SFD field before frame transmission
- Generates XOFF pause frames if the receive FIFO reports a congestion or if ENET*n*_TCR[TFC_PAUSE] is set with ENET*n*_OPD[PAUSE_DUR] set to a non-zero value
- Generates XON pause frames if the receive FIFO congestion condition is cleared or if TFC_PAUSE is set with PAUSE_DUR cleared
- Suspends Ethernet frame transfer (XOFF) if a non-zero pause quanta is received from the MAC receive path
- Adds padding to the frame if required

- Calculates and appends CRC-32 to the transmitted frame
- Sends the frame with correct inter-packet gap (IPG) (deferring)

When the MAC is configured to operate in half-duplex mode, the following additional tasks are performed:

- Collision detection
- Frame retransmit after back-off timer expires

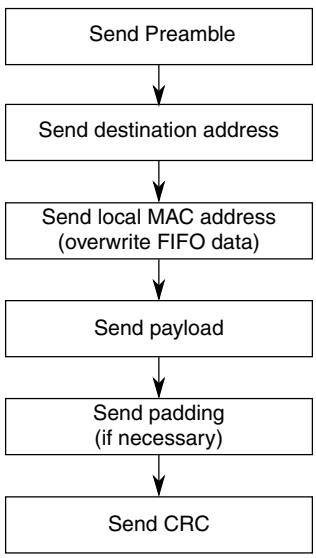


Figure 11-85. Frame transmit overview

11.5.5.5.1 Frame payload padding

The IEEE specification defines a minimum frame length of 64 bytes.

If the frame sent to the MAC from the user application has a size smaller than 60 bytes, the MAC automatically adds padding bytes (0x00) to comply with the Ethernet minimum frame length specification. Transmit padding is always performed and cannot be disabled.

If the MAC is not allowed to append a CRC (TxBD[TC] = 1), the user application is responsible for providing frames with a minimum length of 64 octets.

11.5.5.5.2 MAC address insertion

On each frame received from the core transmit FIFO interface, the source MAC address is either:

- Replaced by the address programmed in the PADDR1/2 fields (ENET*n*_TCR[ADDINS] = 1)
- Transparently forwarded to the Ethernet line (ENET*n*_TCR[ADDINS] = 0)

11.5.5.5.3 CRC-32 generation

The CRC-32 field is optionally generated and appended at the end of a frame.

The CRC polynomial, as specified in the 802.3 standard, is:

- $FCS(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^{8} + x^{7} + x^{5} + x^{4} + x^{2} + x^{1} + 1$

The 32 bits of the CRC value are placed in the FCS field so that the x^{31} term is the right-most bit of the first octet. The CRC bits are thus transmitted in the following order: x^{31}, x^{30},..., x^{1}, x^{0}.

11.5.5.5.4 Inter-packet gap (IPG)

In full-duplex mode, after frame transmission and before transmission of a new frame, an IPG (programmed in ENET*n*_TIPG) is maintained. The minimum IPG can be programmed between 8 and 26 byte-times (64 and 208 bit-times).

In half-duplex mode, the core constantly monitors the line. Actual transmission of the data onto the network occurs only if it has been idle for a 96-bit time period, and any back-off time requirements have been satisfied. In accordance with the standard, the core begins to measure the IPG from CRS/GMII_CRS de-assertion.

11.5.5.5.5 Collision detection and handling — half-duplex operation only

A collision occurs on a half-duplex network when concurrent transmissions from two or more nodes take place. During transmission, the core monitors the line condition and detects a collision when the PHY device asserts COL.

When the core detects a collision while transmitting, it stops transmission of the data and transmits a 32-bit jam pattern. If the collision is detected during the preamble or the SFD transmission, the jam pattern is transmitted after completing the SFD, which results in a minimum 96-bit fragment. The jam pattern is a fixed pattern that is not compared to the actual frame CRC, and has a very low probability (0.532) of having a jam pattern identical to the CRC.

If a collision occurs before transmission of 64 bytes (including preamble and SFD), the MAC core waits for the backoff period and retransmits the packet data (stored in a 64-byte re-transmit buffer) that has already been sent on the line. The backoff period is generated from a pseudo-random process (truncated binary exponential backoff).

If a collision occurs after transmission of 64 bytes (including preamble and SFD), the MAC discards the remainder of the frame, optionally sets the LC interrupt bit, and sets TxBD[LCE].

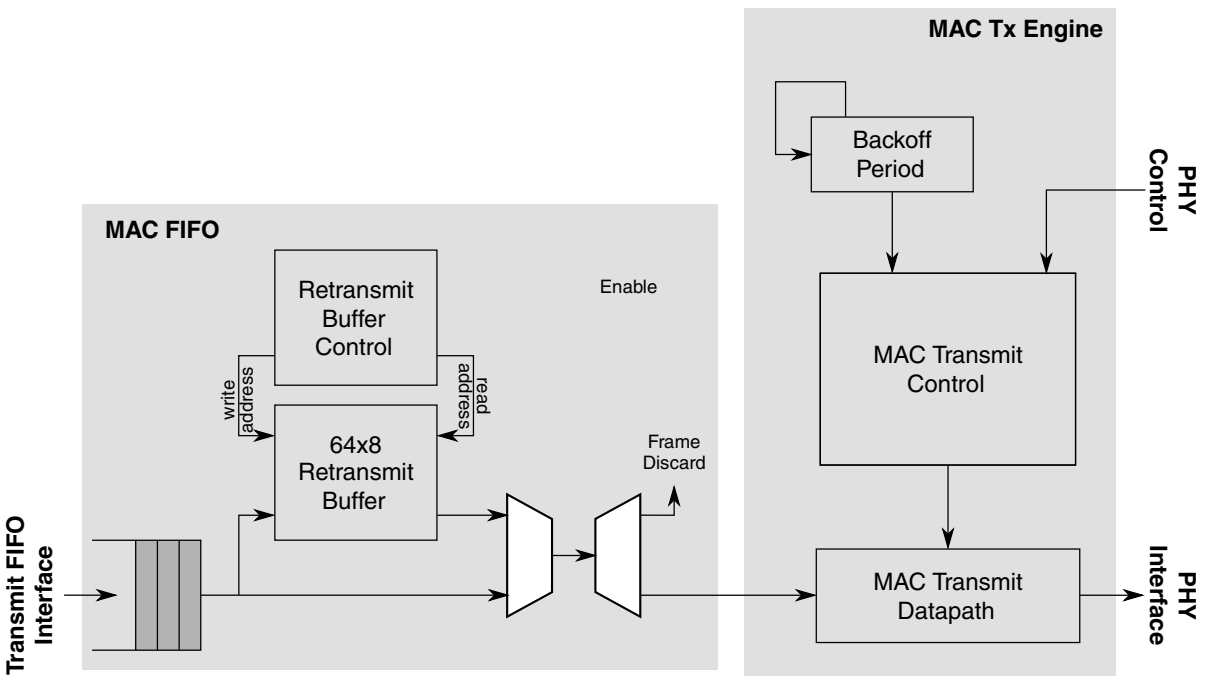


Figure 11-86. Packet re-transmit overview

The backoff time is represented by an integer multiple of slot times. One slot is equal to a 512-bit time period. The number of the delay slot times, before the n^{th} re-transmission attempt, is chosen as a uniformly-distributed random integer in the range:

- $0 < r < 2^k$
- $k = \min(n, N)$; where n is the number of retransmissions and $N = 10$

For example, after the first collision, the backoff period is 0 or 1 slot time. If a collision occurs on the first retransmission, the backoff period is 0, 1, 2, or 3, and so on.

The maximum backoff time (in 512-bit time slots) is limited by N = 10 as specified in the IEEE 802.3 standard.

If a collision occurs after 16 consecutive retransmissions, the core reports an excessive collision condition (ENET*n*_EIR[RL] interrupt field and TxBD[EE]) and discards the current packet from the FIFO.

In networks violating the standard requirements, a collision may occur after transmission of the first 64 bytes. In this case, the core stops the current packet transmission and discards the rest of the packet from the transmit FIFO. The core resumes transmission with the next packet available in the core transmit FIFO.

warning

Ethernet PHYs that support the SQE Test, or "heartbeat," feature must disable this feature. When this feature is enabled, the PHY asserts the collision signal after a frame is transmitted to indicate to the ENET that the PHY's collision logic is working. This may cause data corruption in the next frame from the ENET. This corrupted frame contains up to 21 zero bytes which start somewhere within the MAC destination address field. The ENET, however, will still generate a good FCS (CRC-32) but with corrupted data.

11.5.5.5.6 Rate limiting / traffic shaping support

The MAC-NET supports two methods to optimize frame traffic for either time-sensitive AVB frames (Class A and Class B) or best-effort non-AVB frames:

- Round-robin scheme
- Credit-based traffic shaping

To ensure that sufficient bandwidth is allocated for the AVB frames, the credit-based shaper must be used. Either method can be combined with a time-based shaper to ensure that a frame is always transmitted in its correct time slot.

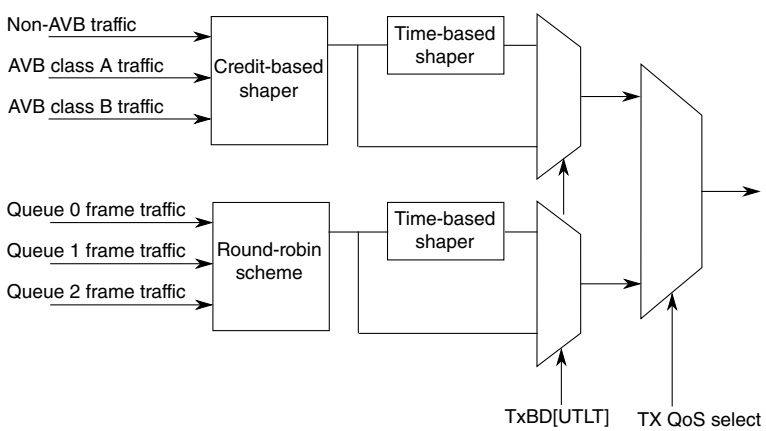


Figure 11-87. Transmit arbitration schemes

11.5.5.5.6.1 Round-robin policy

In the round-robin QoS scheme, each queue is given an equal opportunity to transmit one frame. For example, if queue *n* has a frame to transmit, the queue transmits its frame. After queue *n* has transmitted its frame, or if queue *n* does not have a frame to transmit, queue *n*+1 is then allowed to transmit its frame, and so on.

11.5.5.5.6.2 Credit-based shaper

The AVB credit based shaper acts independently, per class, to control the bandwidth distribution between normal traffic and time-sensitive traffic with respect to the total link bandwidth available. As per the IEEE 802.1Q, the maximum bandwidth distribution that can be allocated for the time-sensitive frames is 75%. The following example uses 70% for AVB frames and 30% for non-AVB frames:

Table 11-68. Bandwidth allocation example

AVB Class A	50%
AVB Class B	20%
non-AVB frames	30%
Total	100%

See DMA Class Based Configuration (ENET_DMA*n*CFG) for information on how to allocate bandwidth. The following figure shows how frame traffic is controlled within the MAC-NET core.

11.5.5.5.6.3 Time-based shaper

The time-based shaper enables the user to specify when a frame can be transmitted. It is always used in combination with either the round-robin scheme or the credit-based shaper. Use of the time-based shaper can ensure that frames are always transmitted in the correct time slot.

11.5.5.5.6.3.1 Time-based shaper example

Time-based shaping involves the following procedure:

1. Read the current value of the timer. See Timer Value Register (ENET_ATVR) and Adjustable Timer Control Register (ENET_ATCR).
2. Calculate the frame launch time and write this value to the TLT field of the frame's TxBD. See Enhanced transmit buffer descriptor.
3. Instruct the ENET to use the TLT by setting TxBD[UTLT].

The frame will be fetched and transmitted only if TxBD[TLT] is less than the current value of the timer. In other words, the timer must be past, that is, be greater than, the transmit launch time before the frame will be fetched and transmitted.

The transmit launch time must be not be greater than the current value of ENET_ATVR + (0.5 × ENET_ATPER). This means the application can not prepare frames with launch times beyond ENET_ATVR + (0.5 × ENET_ATPER). Because ENET_ATVR wraps, calculate TLT using the equation

(ENET_ATVR + (0.5 × ENET_ATPER)) mod ENET_ATPER

For example, if ENET_ATPER is set to the recommended value of 1,000,000,000 ns (one second), then the user must not prepare future frames with launch times greater than (500,000,000 + ENET_ATVR).

As a specific example, if the current value of ENET_ATVR is 100,000, then you can prepare the following frames:

1. Frame 1 TxBD[TLT] = 100,000ns + 125,000ns
2. Frame 2 TxBD[TLT] = 100,000ns + 250,000ns
3. And so on

11.5.5.6 Full-duplex flow control operation

Three conditions are handled by the core's flow control engine:

- Remote device congestion — The remote device connected to the same Ethernet segment as the core reports congestion and requests that the core stop sending data.
- Core FIFO congestion — When the core's receive FIFO reaches a user-programmable threshold (RX section empty), the core sends a pause frame back to the remote device requesting the data transfer to stop.
- Local device congestion — Any device connected to the core can request (typically, via the host processor) the remote device to stop transmitting data.

11.5.5.6.1 Remote device congestion

When the MAC transmit control gets a valid pause quanta from the receive path and if ENET*n*_RCR[FCE] is set, the MAC transmit logic:

- Completes the transfer of the current frame.

- Stops sending data for the amount of time specified by the pause quanta in 512 bit time increments.
- Sets ENET*n*_TCR[RFC_PAUSE].

Frame transfer resumes when the time specified by the quanta expires and if no new quanta value is received, or if a new pause frame with a quanta value set to 0x0000 is received. The MAC also resets RFC_PAUSE to zero.

If ENET*n*_RCR[FCE] cleared, the MAC ignores received pause frames.

Optionally and independent of ENET*n*_RCR[FCE], pause frames are forwarded to the client interface if PAUFWD is set.

11.5.5.6.2 Local device/FIFO congestion

The MAC transmit engine generates pause frames when the local receive FIFO is not able to receive more than a pre-defined number of words (FIFO programmable threshold) or when pause frame generation is requested by the local host processor.

- To generate a pause frame, the host processor sets ENET*n*_TCR[TFC_PAUSE]. A single pause frame is generated when the current frame transfer is completed and TFC_PAUSE is automatically cleared. Optionally, an interrupt is generated.
- An XOFF pause frame is generated when the receive FIFO asserts its section empty flag (internal). An XOFF pause frame is generated automatically, when the current frame transfer completes.
- An XON pause frame is generated when the receive FIFO deasserts its section empty flag (internal). An XON pause frame is generated automatically, when the current frame transfer completes.

When an XOFF pause frame is generated, the pause quanta (payload byte P1 and P2) is filled with the value programmed in ENET*n*_OPD[PAUSE_DUR].

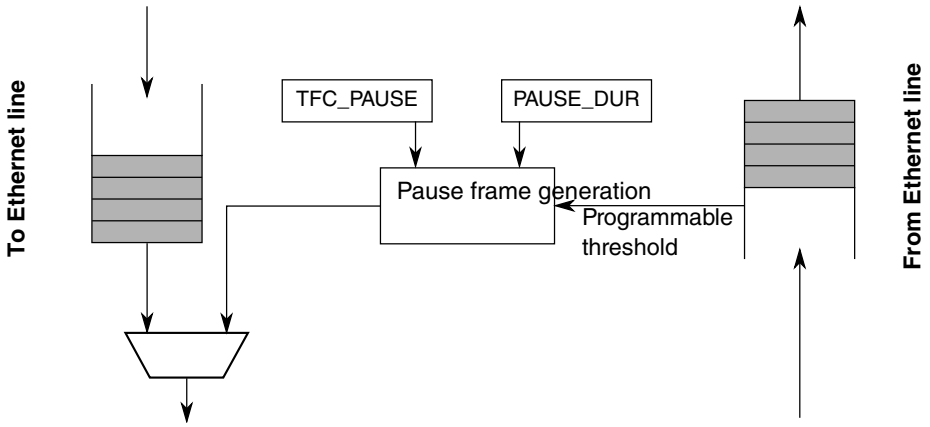


Figure 11-88. Pause frame generation overview

Note

Although the flow control mechanism should prevent any FIFO overflow on the MAC core receive path, the core receive FIFO is protected. When an overflow is detected on the receive FIFO, the current frame is truncated with an error indication set in the frame status word. The frame should subsequently be discarded by the user application.

11.5.5.7 Magic packet detection

Magic packet detection wakes a node that is put in power-down mode by the node management agent. Magic packet detection is supported only if the MAC is configured in sleep mode.

11.5.5.7.1 Sleep mode

To put the MAC in Sleep mode, set ENET*n*_ECR[SLEEP]. At the same time ENET*n*_ECR[MAGICEN] should also be set to enable magic packet detection.

In addition, if ENET is enabled, write 1 to ENET*n*_ECR[SLEEP] before entering into low power mode.

When the MAC is in Sleep mode:

- The transmit logic is disabled.
- The FIFO receive/transmit functions are disabled.
- The receive logic is kept in Normal mode, but it ignores all traffic from the line except magic packets. They are detected so that a remote agent can wake the node.

11.5.5.7.2 Magic packet detection

The core is designed to detect magic packets (see Magic packets) with the destination address set to:

- Any multicast address
- The broadcast address
- The unicast address programmed in PADDR1/2

When a magic packet is detected, EIR[WAKEUP] is set and none of the statistic registers are incremented.

11.5.5.7.3 Wakeup

When a magic packet is detected, indicated by ENET*n*_EIR[WAKEUP], ENET*n*_ECR[SLEEP] should be cleared to resume normal operation of the MAC. Clearing the SLEEP bit automatically masks ENET*n*_ECR[MAGICEN], disabling magic packet detection.

11.5.5.8 IP accelerator functions

The following sections describe the IP accelerator functions.

11.5.5.8.1 Checksum calculation

The IP and ICMP, TCP, UDP checksums are calculated with one's complement arithmetic summing up 16-bit values.

- For ICMP, the checksum is calculated over the complete ICMP datagram, in other words without IP header.
- For TCP and UDP, the checksums contain the header and data sections and values from the IP header, which can be seen as a pseudo-header that is not actually present in the data stream.

Table 11-69. IPv4 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
Zero		Protocol		TCP/UDP length			

Table 11-70. IPv6 pseudo-header for checksum calculation

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
Source address							
Destination address							
TCP/UDP length							
Zero						Next header	

The TCP/UDP length value is the length of the TCP or UDP datagram, which is equal to the payload of an IP datagram. It is derived by subtracting the IP header length from the complete IP datagram length that is given in the IP header (IPv4), or directly taken from the IP header (IPv6). The protocol field is the corresponding value from the IP header. The Zero fields are all zeroes.

For IPv6, the complete 128-bit addresses are considered. The next header value identifies the upper layer protocol as either TCP or UDP. It may differ from the next header value of the IPv6 header if extension headers are inserted before the protocol header.

The checksum calculation uses 16-bit words in network byte order: The first byte sent/received is the MSB, and the second byte sent/received is the LSB of the 16-bit value to add to the checksum. If the frame ends on an odd number of bytes, a zero byte is appended for checksum calculation only, and is not actually transmitted.

11.5.5.8.2 Additional padding processing

According to IEEE 802.3, any Ethernet frame must have a minimum length of 64 octets.

The MAC usually removes padding on receive when a frame with length information is received. Because IP frames have a type value instead of length, the MAC does not remove padding for short IP frames, as it is not aware of the frame contents.

The IP accelerator function can be configured to remove the Ethernet padding bytes that might follow the IP datagram.

On transmit, the MAC automatically adds padding as necessary to fill any frame to a 64-byte length.

11.5.5.8.3 32-bit Ethernet payload alignment

The data FIFOs allow inserting two additional arbitrary bytes in front of a frame. This extends the 14-byte Ethernet header to a 16-byte header, which leads to alignment of the Ethernet payload, following the Ethernet header, on a 32-bit boundary.

This function can be enabled for transmit and receive independently with the corresponding SHIFT16 bits in the ENET*n*_TACC and ENET*n*_RACC registers.

When enabled, the valid frame data is arranged as shown in Table 11-71.

Table 11-71. 64-bit interface data structure with SHIFT16 enabled

63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0	Any value	Any value
Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8	Byte 7	Byte 6
...							

11.5.5.8.3.1 Receive processing

When ENET*n*_RACC[SHIFT16] is set, each frame is received with two additional bytes in front of the frame.

The user application must ignore these first two bytes and find the first byte of the frame in bits 23–16 of the first word from the RX FIFO.

Note

SHIFT16 must be set during initialization and kept set during the complete operation, because it influences the FIFO write behavior.

11.5.5.8.3.2 Transmit processing

When ENET*n*_TACC[SHIFT16] is set, the first two bytes of the first word written (bits 15–0) are discarded immediately by the FIFO write logic.

The SHIFT16 bit can be enabled/disabled for each frame individually if required, but can be changed only between frames.

11.5.5.8.4 Received frame discard

Because the receive FIFO must be operated in store and forward mode (ENET*n*_RSFL cleared), received frames can be discarded based on the following errors:

- The MAC function receives the frame with an error:
  - The frame has an invalid payload length
  - Frame length is greater than MAX_FL

- Frame received with a CRC-32 error
- Frame truncated due to receive FIFO overflow
- Frame is corrupted as PHY signaled an error (RX_ERR asserted during reception)

- An IP frame is detected and the IP header checksum is wrong
- An IP frame with a valid IP header and a valid IP header checksum is detected, the protocol is known but the protocol-specific checksum is wrong

If one of the errors occurs and the IP accelerator function is configured to discard frames (ENET*n*_RACC), the frame is automatically discarded. Statistics are maintained normally and are not affected by this discard function.

11.5.5.8.5 IPv4 fragments

When an IPv4 IP fragment frame is received, only the IP header is inspected and its checksum verified. 32-bit alignment operates the same way on fragments as it does on normal IP frames, as specified above.

The IP fragment frame payload is not inspected for any protocol headers. As such, a protocol header would only exist in the very first fragment. To assist in protocol-specific checksum verification, the one's-complement sum is calculated on the IP payload (all bytes following the IP header) and provided with the frame status word.

The frame fragment status field (RxBD[FRAG]) is set to indicate a fragment reception, and the one's-complement sum of the IP payload is available in RxBD[Payload checksum].

Note

After all fragments have been received and reassembled, the application software can take advantage of the payload checksum delivered with the frame's status word to calculate the protocol-specific checksum of the datagram.

For example, if a TCP payload is delivered by multiple IP fragments, the application software can calculate the pseudo-header checksum value from the first fragment, and add the payload checksums delivered with the status for all fragments to verify the TCP datagram checksum.

11.5.5.8.6 IPv6 support

The following sections describe the IPv6 support.

11.5.5.8.6.1 Receive processing

An Ethernet frame of type 0x86DD identifies an IP Version 6 frame (IPv6) frame. If an IPv6 frame is received, the first IP header is inspected (first ten words), which is available in every IPv6 frame.

If the receive SHIFT16 function is enabled, the IP header is aligned on a 32-bit boundary allowing more efficient processing (see 32-bit Ethernet payload alignment).

For TCP and UDP datagrams, the pseudo-header checksum calculation is performed and verified.

To assist in protocol-specific checksum verification, the one's-complement sum is always calculated on the IP payload (all bytes following the IP header) and provided with the frame status word. For example, if extension headers were present, their sums can be subtracted in software from the checksum to isolate the TCP/UDP datagram checksum, if required.

11.5.5.8.6.2 Transmit processing

For IPv6 transmission, the SHIFT16 function is supported to process 32-bit aligned datagrams.

IPv6 has no IP header checksum; therefore, the IP checksum insertion configuration is ignored.

The protocol checksum is inserted only if the next header of the IP header is a known protocol (TCP, UDP, or ICMP). If a known protocol is detected, the checksum over all bytes following the IP header is calculated and inserted in the correct position.

The pseudo-header checksum calculation is performed for TCP and UDP datagrams accordingly.

11.5.5.9 Resets and stop controls

The following sections describe the resets and stop controls.

11.5.5.9.1 Hardware reset

To reset the Ethernet module, set ENET*n*_ECR[RESET].

11.5.5.9.2 Soft reset

When ENET*n*_ECR[ETHER_EN] is cleared during operation, the following occurs:

- uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers.
- A currently ongoing transmit is terminated by asserting GMII/TXER to the PHY.
- A currently ongoing transmit FIFO write from the application is terminated by stopping the write to the FIFO, and all further data from the application is ignored. All subsequent writes are ignored until re-enabled.
- A currently ongoing receive FIFO read is terminated. The RxBD has arbitrary values in this case.

11.5.5.9.3 Hardware freeze

When the processor enters debug mode and ECR[DBGEN] is set, the MAC enters a freeze state where it stops all transmit and receive activities gracefully.

The following happens when the MAC enters hardware freeze:

- A currently ongoing receive transaction on the receive application interface is completed as normal. No further frames are read from the FIFO.
- A currently ongoing transmit transaction on the transmit application interface is completed as normal (in other words, until writing end-of-packet (EOP)).
- A currently ongoing frame receive is completed normally, after which no further frames are accepted from the MII/GMII.
- A currently ongoing frame transmit is completed normally, after which no further frames are transmitted.

11.5.5.9.4 Graceful stop

During a graceful stop, any currently ongoing transactions are completed normally and no further frames are accepted. The MAC can resume from a graceful stop without the need for a reset (for example, clearing ETHER_EN is not required).

The following conditions lead to a graceful stop of the MAC transmit or receive datapaths.

11.5.5.9.4.1 Graceful transmit stop (GTS)

When gracefully stopped, the MAC is no longer reading frame data from the transmit FIFO and has completed any ongoing transmission.

In any of the following conditions, the transmit datapath stops after an ongoing frame transmission has been completed normally.

- ENET*n*_TCR[GTS] is set by software.
- ENET*n*_TCR[TFC_PAUSE] is set by software requesting a pause frame transmission. The status (and register bit) is cleared after the pause frame has been sent.
- A pause frame was received stopping the transmitter. The stopped situation is terminated when the pause timer expires or a pause frame with zero quanta is received.
- MAC is placed in Sleep mode by software or the processor entering Stop mode (see Sleep mode).
- The MAC is in Hardware Freeze mode.

When the transmitter has reached its stopped state, the following events occur:

- The GRA interrupt is asserted, when transitioned into stopped.
- In Hardware Freeze mode, the GRA interrupt does not wait for the application write completion and asserts when the transmit state machine (in other words, line side of TX FIFO) reaches its stopped state.

11.5.5.9.4.2 Graceful receive stop (GRS)

When gracefully stopped, the MAC is no longer writing frames into the receive FIFO.

The receive datapath stops after any ongoing frame reception has been completed normally, if any of the following conditions occur:

- MAC is placed in Sleep mode either by the software or the processor is in Stop mode). The MAC continues to receive frames and search for magic packets if enabled (see Magic packet detection). However, no frames are written into the receive FIFO, and therefore are not forwarded to the application.
- The MAC is in Hardware Freeze mode. The MAC does not accept any frames from the MII/GMII.

When the receive datapath is stopped, the following events occur:

- If the RX is in the stopped state, RCR[GRS] is set
- The GRA interrupt is asserted when the transmitter and receiver are stopped
- Any ongoing receive transaction to the application (RX FIFO read) continues normally until the frame end of package (EOP) is reached. After this, the following occurs:
  - When Sleep mode is active, all further frames are discarded, flushing the RX FIFO
  - In Hardware Freeze mode, no further frames are delivered to the application and they stay in the receive FIFO.

Note

The assertion of GRS does not wait for an ongoing FIFO read transaction on the application side of the FIFO (FIFO read).

11.5.5.9.4.3 Graceful stop interrupt (GRA)

The graceful stopped interrupt (GRA) is asserted for the following conditions:

- In Sleep mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- In Hardware Freeze mode, the interrupt asserts only after both TX and RX datapaths are stopped.
- The MAC transmit datapath is stopped for any other condition (GTS, TFC_PAUSE, pause received).

The GRA interrupt is triggered only once when the stopped state is entered. If the interrupt is cleared while the stop condition persists, no further interrupt is triggered.

11.5.5.10 IEEE 1588 functions

To allow for IEEE 1588 or similar time synchronization protocol implementations, the MAC is combined with a time-stamping module to support precise time-stamping of incoming and outgoing frames. Set ENET*n*_ECR[EN1588] to enable 1588 support.

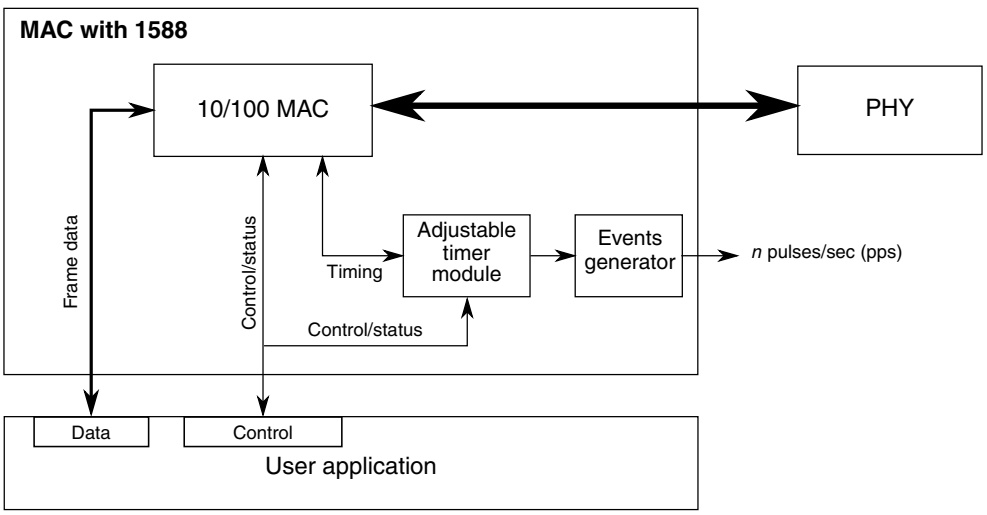


Figure 11-89. IEEE 1588 functions overview

11.5.5.10.1 Adjustable timer module

The adjustable timer module (TSM) implements the free-running counter (FRC), which generates the timestamps. The FRC operates with the time-stamping clock, which can be set to any value depending on your system requirements.

Through dedicated correction logic, the timer can be adjusted to allow synchronization to a remote master and provide a synchronized timing reference to the local system. The timer can be configured to cause an interrupt after a fixed time period, to allow synchronization of software timers or perform other synchronized system functions.

The timer is typically used to implement a period of one second; hence, its value ranges from 0 to (1×10^9)-1. The period event can trigger an interrupt, and software can maintain the seconds and hours time values as necessary.

11.5.5.10.1.1 Adjustable timer implementation

The adjustable timer consists of a programmable counter/accumulator and a correction counter. The periods of both counters and their increment rates are freely configurable, allowing very fine tuning of the timer.

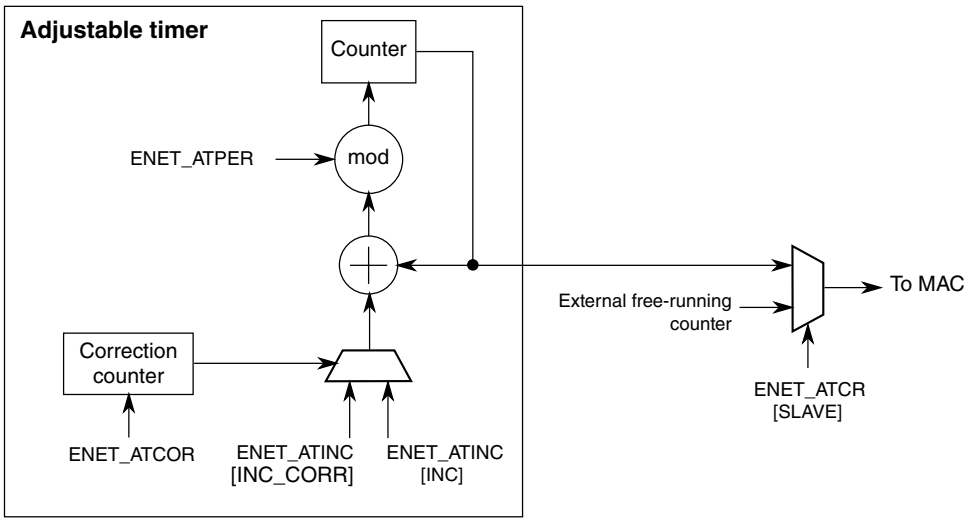


Figure 11-90. Adjustable timer implementation detail

The counter produces the current time. During each time-stamping clock cycle, a constant value is added to the current time as programmed in ENET*n*_ATINC. The value depends on the chosen time-stamping clock frequency. For example, if it operates at 125 MHz, setting the increment to eight represents 8 ns.

The period, configured in ENET*n*_ATPER, defines the modulo when the counter wraps. In a typical implementation, the period is set to 1×10^9 so that the counter wraps every second, and hence all timestamps represent the absolute nanoseconds within the one second period. When the period is reached, the counter wraps to start again respecting the period modulo. This means it does not necessarily start from zero, but instead the counter is loaded with the value (Current + Inc $-(1 \times 10^9)$), assuming the period is set to 1×10^9.

The correction counter operates fully independently, and increments by one with each time-stamping clock cycle. When it reaches the value configured in ENET*n*_ATCOR, it restarts and instructs the timer once to increment by the correction value, instead of the normal value.

The normal and correction increments are configured in ENET*n*_ATINC. To speed up the timer, set the correction increment more than the normal increment value. To slow down the timer, set the correction increment less than the normal increment value.

The correction counter only defines the distance of the corrective actions, not the amount. This allows very fine corrections and low jitter (in the range of 1 ns) independent of the chosen clock frequency.

By enabling slave mode (ENET*n*_ATCR[SLAVE] = 1), the timer is ignored and the current time is externally provided from one of the external modules. See the Chip Configuration details for which clock source is used. This is useful if multiple modules

within the system must operate from a single timer. When slave mode is enabled, you still must set ENET*n*_ATINC[INC] to the value of the master, since it is used for internal comparisons.

11.5.5.10.2 Transmit timestamping

Only 1588 event frames need to be time-stamped on transmit. The client application (for example, the MAC driver) should detect 1588 event frames and set TxBD[TS] together with the frame.

If TxBD[TS] is set, the MAC records the timestamp for the frame in ENET*n*_ATSTMP. ENET*n*_EIR[TS_AVAIL] is set to indicate that a new timestamp is available.

Software implements a handshaking procedure by setting TxBD[TS] when it transmits the frame for which a timestamp is needed, and then waits for ENET*n*_EIR[TS_AVAIL] to determine when the timestamp is available. The timestamp is then read from ENET*n*_ATSTMP. This is done for all event frames. Other frames do not use TxBD[TS] and, therefore, do not interfere with the timestamp capture.

11.5.5.10.3 Receive timestamping

When a frame is received, the MAC latches the value of the timer when the frame's start of frame delimiter (SFD) field is detected, and provides the captured timestamp on RxBD[1588 timestamp]. This is done for all received frames.

11.5.5.10.4 Time synchronization

The adjustable timer module is available to synchronize the local clock of a node to a remote master. It implements a free running 32-bit counter, and also contains an additional correction counter.

The correction counter increases or decreases the rate of the free running counter, enabling very fine granular changes of the timer for synchronization, yet adding only very low jitter when performing corrections.

The application software implements, in a slave scenario, the required control algorithm, setting the correction to compensate for local oscillator drifts and locking the timer to the remote master clock on the network.

The timer and all timestamp-related information should be configured to show the true nanoseconds value of a second (in other words, the timer is configured to have a period of one second). Hence, the values range from 0 to (1×10^9)–1. In this application, the seconds counter is implemented in software using an interrupt function that is executed when the nanoseconds counter wraps at 1×10^9.

11.5.5.10.5 Input Capture and Output Compare

The Input Capture Output Compare block can be used to provide precise hardware timing for input and output events.

11.5.5.10.5.1 Input capture

The TCCR*n* capture registers latch the time value when the corresponding external event occurs. An event can be a rising-, falling-, or either-edge of one of the 1588_TMR*n* signals. An event will cause the corresponding TCSR*n*[TF] timer flag to be set, indicating that an input capture has occurred. If the corresponding interrupt is enabled with the TCSR*n*[TIE] field, an interrupt can be generated.

11.5.5.10.5.2 Output compare

The TCCR*n* compare registers are loaded with the time at which the corresponding event should occur. When the ENET free-running counter value matches the output compare reference value in the TCCR*n* register, the corresponding flag, TCSR*n*[TF], is set, indicating that an output compare has occurred. The corresponding interrupt, if enabled by TCSR*n*[TIE], will be generated.The corresponding 1588_TMR*n* output signal will be asserted according to TCSR*n*[TMODE].

NOTE

If TSCR*n*[TMODE] is set to 10X1b or 1010b then the timer output pin toggle-on-overflow will occur only when PINPER, PEREN, and EN bits of the ATCR register are one.

11.5.5.10.5.3 DMA requests

A DMA request can be enabled by setting TCSR*n*[TDRE]. The corresponding DMA request is generated when the TCSR*n*[TF] timer flag is set. When the DMA has completed, the corresponding TCSRn[TF] flag is cleared.

11.5.5.11 FIFO thresholds

The core FIFO thresholds are fully programmable to dynamically change the FIFO operation.

For example, store and forward transfer can be enabled by a simple change in the FIFO threshold registers.

The thresholds are defined in 64-bit words.

The receive and transmit FIFOs both have a depth of 1024 words.

11.5.5.11.1 Receive FIFO

Four programmable thresholds are available, which can be set to any value to control the core operation as follows.

Table 11-72. Receive FIFO thresholds definition

Register	Description
ENET*n*_RSFL [RX_SECTION_FULL]	When the FIFO level reaches the ENET*n*_RSFL value, the MAC status signal is asserted to indicate that data is available in the receive FIFO (cut-through operation). Once asserted, if the FIFO empties below the threshold set with ENET*n*_RAEM and if the end-of-frame is not yet stored in the FIFO, the status signal is deasserted again. If a frame has a size smaller than the threshold (in other words, an end-of-frame is available for the frame), the status is also asserted. To enable store and forward on the receive path, clear ENET*n*_RSFL. The MAC status signal is asserted only when a complete frame is stored in the receive FIFO. When programming a non-zero value to ENET*n*_RSFL (cut-through operation) it should be greater than ENET*n*_RAEM.
ENET*n*_RAEM [RX_ALMOST_EMPTY]	When the FIFO level reaches the ENET*n*_RAEM value, and the end-of-frame has not been received, the core receive read control stops the FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. Set ENET*n*_RAEM to a minimum of six.
ENET*n*_RAFL [RX_ALMOST_FULL]	When the FIFO level approaches the maximum and there is no more space remaining for at least ENETn_RAFL number of words, the MAC control logic stops writing data in the FIFO and truncates the receive frame to avoid FIFO overflow. The corresponding error status is set when the frame is delivered to the application. Set ENET*n*_RAFL to a minimum of 4.
ENET*n*_RSEM [RX_SECTION_EMPTY]	When the FIFO level reaches the ENET*n*_RSEM value, an indication is sent to the MAC transmit logic, which generates an XOFF pause frame. This indicates FIFO congestion to the remote Ethernet client. When the FIFO level goes below the value programmed in ENET*n*_RSEM, an indication is sent to the MAC transmit logic, which generates an XON pause frame. This indicates the FIFO congestion is cleared to the remote Ethernet client. Clearing ENET*n*_RSEM disables any pause frame generation.

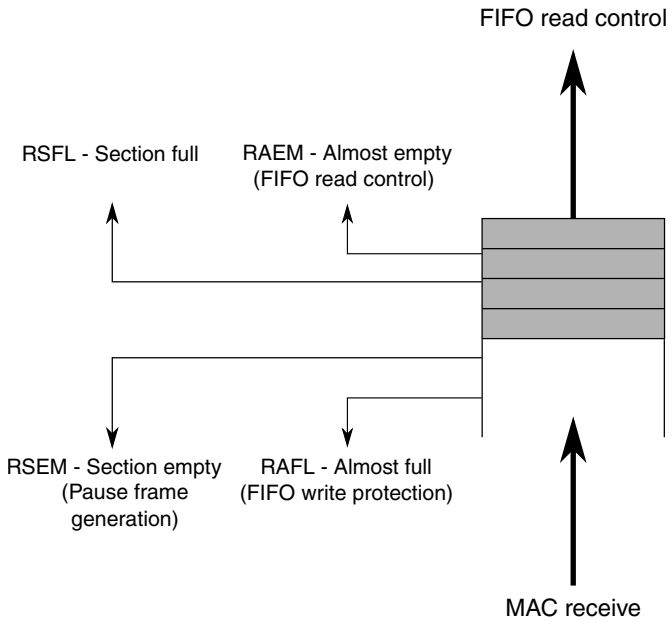


Figure 11-91. Receive FIFO overview

11.5.5.11.2 Transmit FIFO

Four programmable thresholds are available which control the core operation as described below.

Table 11-73. Transmit FIFO thresholds definition

Register	Description
ENET*n*_TAEM [TX_ALMOST _EMPTY]	When the FIFO level reaches the ENET*n*_TAEM value and no end-of-frame is available for the frame, the MAC transmit logic avoids a FIFO underflow by stopping FIFO reads and transmitting the Ethernet frame with an MII error indication. Set ENET*n*_TAEM to a minimum of 4.
ENET*n*_TAFL [TX_ALMOST _FULL]	When the FIFO level approaches the maximum, so that there is no more space for at least ENET*n*_TAFL number of words, the MAC deasserts its control signal to the application. If the application does not react on this signal, the FIFO write control logic avoids FIFO overflow by truncating the current frame and setting the error status. As a result, the frame is transmitted with an GMII/MII error indication. Set ENET*n*_TAFL to a minimum of 4. Larger values allow more latency for the application to react on the MAC control signal deassertion, before the frame is truncated. A typical setting is 8, which offers 3–4 clock cycles of latency to the application to react on the MAC control signal deassertion.
ENET*n*_TSEM [TX_SECTION _EMPTY]	When the FIFO level reaches the ENET*n*_TSEM value, a MAC status signal is deasserted to indicate that the transmit FIFO is getting full. This gives the ENET module an indication to slow or stop its write transaction to avoid a buffer overflow. This is a pure indication function to the application. It has no effect within the MAC. When ENET*n*_TSEM is 0, the signal is never deasserted.
ENET*n*_TFWR	When the FIFO level reaches the ENET*n*_TFWR value and when STRFWD is cleared, the MAC transmit control logic starts frame transmission before the end-of-frame is available in the FIFO (cut-through operation).

Table 11-73. Transmit FIFO thresholds definition

Register	Description
	If a complete frame has a size smaller than the ENET*n*_TFWR threshold, the MAC also transmits the frame to the line. To enable store and forward on the transmit path, set STRFWD. In this case, the MAC starts to transmit data only when a complete frame is stored in the transmit FIFO.

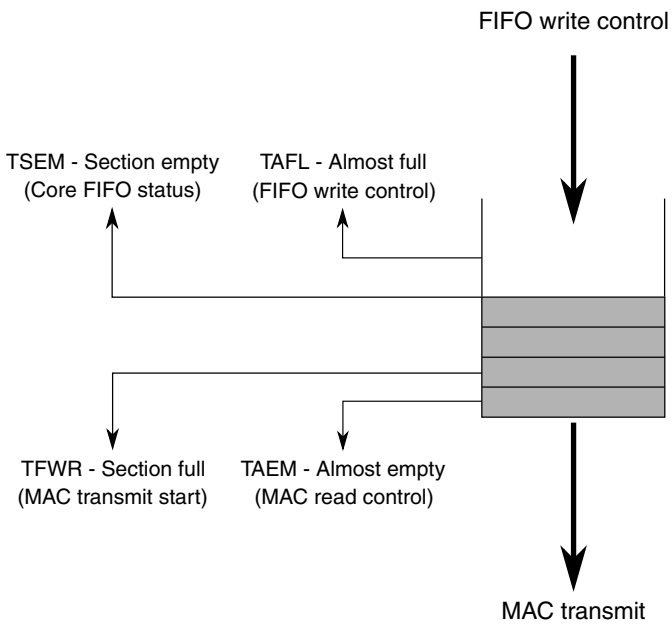


Figure 11-92. Transmit FIFO overview

11.5.5.12 Loopback options

The core implements external and internal loopback options, which are controlled by the ENET*n*_RCR register fields found here.

The core implements external and internal loopback options, which are controlled by the following ENET*n*_RCR register fields:

Table 11-74. Loopback options

Register field	Description
LOOP	Internal MII loopback. The MAC transmit is returned to the MAC receive. No data is transmitted to the external interfaces. In MII internal loopback, MII_TXCLK and MII_RXCLK must be provided with a clock signal (2.5 MHz for 10 Mbit/s, and 25 MHz for 100 Mbit/s))

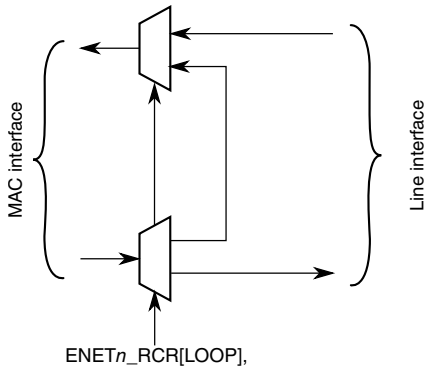


Figure 11-93. Loopback options

11.5.5.13 Legacy buffer descriptors

To support the Ethernet controller on previous chips, legacy FEC buffer descriptors are available. To enable legacy support, write 0 to ENET*n*_ECR[1588EN].

NOTE

- Legacy buffer descriptors are used in single-ring mode, that is, when DMA*n*CFG[DMA_CLASS_EN] are zero. In multi-ring mode, legacy TxBDs are used only with the round-robin scheme.
- The legacy buffer descriptor tables show the byte order for big-endian chips. DBSWP must be set to 0 after reset to enable big-endian mode.

11.5.5.13.1 Legacy receive buffer descriptor

The following table shows the legacy FEC receive buffer descriptor. Table 11-78 contains the descriptions for each field.

Table 11-75. Legacy FEC receive buffer descriptor (RxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer — high halfword															
Offset + 6	Rx data buffer pointer — low halfword															

11.5.5.13.2 Legacy transmit buffer descriptor

The following table shows the legacy FEC transmit buffer descriptor. Table 11-80 contains the descriptions for each field.

Table 11-76. Legacy FEC transmit buffer descriptor (TxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	ABC[1]	—	—	—	—	—	—	—	—	—
Offset + 2	Data Length															
Offset + 4	Tx Data Buffer Pointer — high halfword															
Offset + 6	Tx Data Buffer Pointer — low halfword															

1. This field is not supported by the uDMA.

11.5.5.14 Enhanced buffer descriptors

This section provides a description of the enhanced operation of the driver/uDMA via the buffer descriptors.

It is followed by a detailed description of the receive and transmit descriptor fields. To enable the enhanced features, set ENET*n*_ECR[1588EN].

NOTE

The enhanced buffer descriptor tables show the byte order for big-endian chips. DBSWP must be set to 0 after reset to enable big-endian mode.

11.5.5.14.1 Enhanced receive buffer descriptor

The following table shows the enhanced uDMA receive buffer descriptor. Table 11-78 contains the descriptions for each field.

Table 11-77. Enhanced uDMA receive buffer descriptor (RxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	E	RO1	W	RO2	L	—	—	M	BC	MC	LG	NO	—	CR	OV	TR
Offset + 2	Data length															
Offset + 4	Rx data buffer pointer – high halfword															
Offset + 6	Rx data buffer pointer – low halfword															
Offset + 8	ME	—	—	—	—	PE	CE	UC	INT	—	—	—	—	—	—	—
Offset + A	VPCP			—	—	—	—	—	—	—	ICE	PCR	—	VLAN	IPV6	FRAG

Table continues on the next page...

Table 11-77. Enhanced uDMA receive buffer descriptor (RxBD) (continued)

Offset + C	Header length					—	—	—	Protocol type							
Offset + E	Payload checksum															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – high halfword															
Offset + 16	1588 timestamp – low halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 11-78. Receive buffer descriptor field definitions

Word	Field	Description
Offset + 0	15 E	Empty. Written by the MAC (= 0) and user (= 1). 0 The data buffer associated with this BD is filled with received data, or data reception has aborted due to an error condition. The status and length fields have been updated as required. 1 The data buffer associated with this BD is empty, or reception is currently in progress.
Offset + 0	14 RO1	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 0	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location. 1 The next buffer descriptor is found at the location defined in ENET*n*_RDSR.
Offset + 0	12 RO2	Receive software ownership. This field is reserved for use by software. This read/write field is not modified by hardware, nor does its value affect hardware.
Offset + 0	11 L	Last in frame. Written by the uDMA. 0 The buffer is not the last in a frame. 1 The buffer is the last in a frame.
Offset + 0	10–9	Reserved, must be cleared.
Offset + 0	8 M	Miss. Written by the MAC. This field is set by the MAC for frames accepted in promiscuous mode, but flagged as a miss by the internal address recognition. Therefore, while in promiscuous mode, you can use this field to quickly determine whether the frame was destined to this station. This field is valid only if the L and PROM fields are set to 1. 0 The frame was received because of an address recognition hit. 1 The frame was received because of promiscuous mode. The information needed for this field comes from the promiscuous_miss(ff_rx_err_stat[26]) sideband signal.
Offset + 0	7 BC	Set if the DA is broadcast (FFFF_FFFF_FFFF).
Offset + 0	6 MC	Set if the DA is multicast and not BC.

Table continues on the next page...

Table 11-78. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 0	5 LG	Receive frame length violation. Written by the MAC. A frame length greater than RCR[MAX_FL] was recognized. This field is valid only if the L field is set. The receive data is not altered in any way unless the length exceeds TRUNC_FL bytes.
Offset + 0	4 NO	Receive non-octet aligned frame. Written by the MAC. A frame that contained a number of bits not divisible by 8 was received, and the CRC check that occurred at the preceding byte boundary generated an error or a PHY error occurred. This field is valid only if the L field is set. If this field is set, the CR field is not set.
Offset + 0	3	Reserved, must be cleared.
Offset + 0	2 CR	Receive CRC or frame error. Written by the MAC. This frame contains a PHY or CRC error and is an integral number of octets in length. This field is valid only if the L field is set.
Offset + 0	1 OV	Overrun. Written by the MAC. A receive FIFO overrun occurred during frame reception. If this field is set, the other status fields, M, LG, NO, and CR, lose their normal meaning and are zero. This field is valid only if the L field is set.
Offset + 0	0 TR	Set if the receive frame is truncated (frame length >TRUNC_FL). If the TR field is set, the frame must be discarded and the other error fields must be ignored because they may be incorrect.
Offset + 2	15–0 Data Length	Data length. Written by the MAC. Data length is the number of octets written by the MAC into this BD's data buffer if L is cleared (the value is equal to EMRBR), or the length of the frame including CRC if L is set. It is written by the MAC once as the BD is closed.
0ffset + 4	15–0 Data buffer pointer high	Receive data buffer pointer, high halfword[1]
Offset + 6	15–0 Data buffer pointer low	Receive data buffer pointer, low halfword
Offset + 8	15 ME	MAC error. This field is written by the uDMA. This field means that the frame stored in the system memory was received with an error (typically, a receive FIFO overflow). This field is only valid when the L field is set.
Offset + 8	14–11	Reserved, must be cleared.
Offset + 8	10 PE	PHY Error. This field is written by the uDMA. Set to "1"when the frame was received with an Error character on the PHY interface. The frame is invalid. This field is valid only when the L field is set.
Offset + 8	9 CE	Collision. This field is written by the uDMA. Set when the frame was received with a collision detected during reception. The frame is invalid and sent to the user application. This field is valid only when the L field is set.
Offset + 8	8 UC	Unicast. This field is written by the uDMA, and means that the frame is unicast. This field is valid regardless of whether the L field is set.
Offset + 8	7 INT	Generate RXB/RXF interrupt. This field is set by the user to indicate that the uDMA is to generate an interrupt on the *dma_int_rxb* / *dma_int_rxfevent*.
Offset + 8	6–0	Reserved, must be cleared.
Offset + A	15–13 VPCP	VLAN priority code point. This field is written by the uDMA to indicate the frame priority level. Valid values are from 0 (best effort) to 7 (highest). This value can be used to prioritize different classes of traffic (e.g., voice, video, data). This field is only valid if the L field is set.
Offset + A	12–6	Reserved, must be cleared.

Table continues on the next page...

Table 11-78. Receive buffer descriptor field definitions (continued)

Word	Field	Description
Offset + A	5 ICE	IP header checksum error. This is an accelerator option. This field is written by the uDMA. Set when either a non-IP frame is received or the IP header checksum was invalid. An IP frame with less than 3 bytes of payload is considered to be an invalid IP frame. This field is only valid if the L field is set.
Offset + A	4 PCR	Protocol checksum error. This is an accelerator option. This field is written by the uDMA. Set when the checksum of the protocol is invalid or an unknown protocol is found and checksumming could not be performed. This field is only valid if the L field is set.
Offset + A	3	Reserved, must be cleared.
Offset + A	2 VLAN	VLAN. This is an accelerator option. This field is written by the uDMA. It means that the frame has a VLAN tag. This field is valid only if the L field is set.
Offset + A	1 IPV6	IPV6 Frame. This field is written by the uDMA. This field indicates that the frame has an IPv6 frame type. If this field is not set it means that an IPv4 or other protocol frame was received. This field is valid only if the L field is set.
Offset + A	0 FRAG	IPv4 Fragment.This is an accelerator option.This field is written by the uDMA. It indicates that the frame is an IPv4 fragment frame. This field is only valid when the L field is set.
Offset + C	15–11 Header length	Header length. This is an accelerator option. This field is written by the uDMA. This field is the sum of 32-bit words found within the IP and its following protocol headers. If an IP datagram with an unknown protocol is found, then the value is the length of the IP header. If no IP frame or an erroneous IP header is found, the value is 0. The following values are minimum values if no header options exist in the respective headers: • ICMP/IP: 6 (5 IP header, 1 ICMP header) • UDP/IP: 7 (5 IP header, 2 UDP header) • TCP/IP: 10 (5 IP header, 5 TCP header) This field is only valid if the L field is set.
Offset + C	10–8	Reserved, must be cleared.
Offset + C	7–0 Protocol type	Protocol type. This is an accelerator option. The 8-bit protocol field found within the IP header of the frame. It is valid only when ICE is cleared. This field is valid only when the L field is set.
Offset + E	15–0 Payload checksum	Internet payload checksum. This is an accelerator option. It is the one's complement sum of the payload section of the IP frame. The sum is calculated over all data following the IP header until the end of the IP payload. This field is valid only when the L field is set.
Offset + 10	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 10	14–0	Reserved, must be cleared.
Offset + 12	15–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA. It is only valid if the L field is set.
Offset + 16		
Offset + 18 – Offset + 1E	15–0	Reserved, must be cleared.

1. The receive buffer pointer, containing the address of the associated data buffer, must always be evenly divisible by 64. The buffer must reside in memory external to the MAC. The Ethernet controller never modifies this value.

11.5.5.14.2 Enhanced transmit buffer descriptor

Table 11-79. Enhanced transmit buffer descriptor (TxBD)

	Byte 0								Byte 1							
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Offset + 0	R	TO1	W	TO2	L	TC	—	—	—	—	—	—	—	—	—	—
Offset + 2	Data length															
Offset + 4	Tx Data Buffer Pointer – high halfword															
Offset + 6	Tx Data Buffer Pointer – low halfword															
Offset + 8	—	INT	TS	PINS	IINS	—	—	UTLT	FTYPE				—	—	—	—
Offset + A	TXE	—	UE	EE	FE	LCE	OE	TSE	—	—	—	—	—	—	—	—
Offset + C	TLT – high halfword															
Offset + E	TLT – low halfword															
Offset + 10	BDU	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 12	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 14	1588 timestamp – high halfword															
Offset + 16	1588 timestamp – low halfword															
Offset + 18	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1A	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1C	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—
Offset + 1E	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—	—

Table 11-80. Enhanced transmit buffer descriptor field definitions

Word	Field	Description
Offset + 0	15 R	Ready. Written by the MAC and user. 0 The data buffer associated with this BD is not ready for transmission. You are free to manipulate this BD or its associated data buffer. The MAC clears this field after the buffer has been transmitted or after an error condition is encountered. 1 The data buffer, prepared for transmission by you, has not been transmitted or currently transmits. You may write no fields of this BD after this field is set.
Offset + 0	14 TO1	Transmit software ownership. This field is reserved for software use. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 0	13 W	Wrap. Written by user. 0 The next buffer descriptor is found in the consecutive location 1 The next buffer descriptor is found at the location defined in ETDSR.
Offset + 0	12 TO2	Transmit software ownership. This field is reserved for use by software. This read/write field is not modified by hardware and its value does not affect hardware.
Offset + 0	11 L	Last in frame. Written by user. 0 The buffer is not the last in the transmit frame 1 The buffer is the last in the transmit frame

Table continues on the next page...

Table 11-80. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 0	10 TC	Transmit CRC. Written by user, and valid only when L is set. 0 End transmission immediately after the last data byte 1 Transmit the CRC sequence after the last data byte This field is valid only when the L field is set.
Offset + 0	9 ABC	Append bad CRC. **Note:** This field is not supported by the uDMA and is ignored.
Offset + 0	8–0	Reserved, must be cleared.
Offset + 2	15–0 Data Length	Data length, written by user. Data length is the number of octets the MAC should transmit from this BD's data buffer. It is never modified by the MAC.
Offset + 4	15–0 Data buffer pointer high	Tx data buffer pointer, high halfword. The buffer must reside in memory external to the MAC. This value is never modified by the Ethernet controller. **NOTE:** For optimal performance, make the transmit buffer pointer evenly divisible by 64.
Offset + 6	15–0 Data buffer pointer low	Tx data buffer pointer, low halfword
Offset + 8	15	Reserved, must be cleared.
Offset + 8	14 INT	Generate interrupt flags. This field is written by the user. This field is valid regardless of the L field and must be the same for all EBD for a given frame. The uDMA does not update this value.
Offset + 8	13 TS	Timestamp. This field is written by the user. This indicates that the uDMA is to generate a timestamp frame to the MAC. This field is valid regardless of the L field and must be the same for all EBD for the given frame. The uDMA does not update this value.
Offset + 8	12 PINS	Insert protocol specific checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the protocol checksum and overwrites the corresponding checksum field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + 8	11 IINS	Insert IP header checksum. This field is written by the user. If set, the MAC's IP accelerator calculates the IP header checksum and overwrites the corresponding header field with the calculated value. The checksum field must be cleared by the application generating the frame. The uDMA does not update this value. This field is valid regardless of the L field and must be the same for all EBD for a given frame.
Offset + 8	10-9	Reserved, must be cleared.
Offset + 8	8 UTLT	Use transmit launch time. If set to 1, TLT high and low values are used to determine if the frame can be transmitted. This field must only be set in the *first* BD of a frame. TLT is always used in combination with either the round-robin scheme or the credit-based shaper. TLT can be used with single a BD queue. However, at least one DMA class, DMA*n*CFG[DMA_CLASS_ EN] must be enabled. For example, to use TLT with only queue 0, enable one of the DMA*n*CFG[DMA_CLASS_EN] fields and clear both TDAR1 and TDAR2. Although this enables multi-queue, only the single queue 0 is used.

Table continues on the next page...

Table 11-80. Enhanced transmit buffer descriptor field definitions (continued)

Word	Field	Description
Offset + 8	7–4 FTYPE	Type of frame to be transmitted. If a credit-based scheme is used, this field must match the BD ring queue. 0x0 – Non-AVB. Corresponds to ENET_TDSR. 0x1 – AVB Class A. Corresponds to ENET_TDSR1. 0x2 – AVB Class B. Corresponds to ENET_TDSR2. All other values are reserved.
Offset + 8	3–0	Reserved, must be cleared.
Offset + A	15 TXE	Transmit error occurred. This field is written by the uDMA. This field indicates that there was a transmit error of some sort reported with the frame. Effectively this field is an OR of the other error fields including UE, EE, FE, LCE, OE, and TSE. This field is valid only when the L field is set.
Offset + A	14	Reserved, must be cleared.
Offset + A	13 UE	Underflow error. This field is written by the uDMA. This field indicates that the MAC reported an underflow error on transmit. This field is valid only when the L field is set.
Offset + A	12 EE	Excess Collision error. This field is written by the uDMA. This field indicates that the MAC reported an excess collision error on transmit. This field is valid only when the L field is set.
Offset + A	11 FE	Frame with error. This field is written by the uDMA. This field indicates that the MAC reported that the uDMA reported an error when providing the packet. This field is valid only when the L field is set.
Offset + A	10 LCE	Late collision error. This field is written by the uDMA. This field indicates that the MAC reported that there was a Late Collision on transmit. This field is valid only when the L field is set.
Offset + A	9 OE	Overflow error. This field is written by the uDMA. This field indicates that the MAC reported that there was a FIFO overflow condition on transmit. This field is only valid when the L field is set.
Offset + A	8 TSE	Timestamp error. This field is written by the uDMA. This field indicates that the MAC reported a different frame type then a timestamp frame. This field is valid only when the L field is set.
Offset + A	7–0	Reserved, must be cleared.
Offset + C	15–0 TLT high	Transmit launch time high. Specifies when frame can by transmitted.
Offset + E	15–0 TLT low	Transmit launch time low. Specifies when frame can by transmitted.
Offset + 10	15 BDU	Last buffer descriptor update done. Indicates that the last BD data has been updated by uDMA. This field is written by the user (=0) and uDMA (=1).
Offset + 10	14–0	Reserved, must be cleared.
Offset + 12	15–0	Reserved, must be cleared.
Offset + 14	15–0 1588 timestamp	This value is written by the uDMA . It is valid only when the L field is set.
Offset + 16		
Offset + 18–Offset + 1E	15–0	Reserved, must be cleared.

11.5.5.15 Client FIFO application interface

The FIFO interface is completely asynchronous from the Ethernet line, and the transmit and receive interface can operate at a different clock rate.

All transfers to/from the user application are handled independently of the core operation, and the core provides a simple interface to user applications based on a two-signal handshake.

11.5.5.15.1 Data structure description

The data structure defined in the following tables for the FIFO interface must be respected to ensure proper data transmission on the Ethernet line. Byte 0 is sent to and received from the line first.

Table 11-81. FIFO interface data structure

	63 56	55 48	47 40	39 32	31 24	23 16	15 8	7 0
Word 0	Byte 7	Byte 6	Byte 5	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Word 1	Byte 15	Byte 14	Byte 13	Byte 12	Byte 11	Byte 10	Byte 9	Byte 8
...	...							

The size of a frame on the FIFO interface may not be a modulo of 64-bit.

The user application may not care about the Ethernet frame formats in full detail. It needs to provide and receive an Ethernet frame with the following structure:

- Ethernet MAC destination address
- Ethernet MAC source address
- Optional 802.1q VLAN tag (VLAN type and info field)
- Ethernet length/type field
- Payload

Frames on the FIFO interface do not contain preamble and SFD fields, which are inserted and discarded by the MAC on transmit and receive, respectively.

- On receive, CRC and frame padding can be stripped or passed through transparently.
- On transmit, padding and CRC can be provided by the user application, or appended automatically by the MAC independently for each frame. No size restrictions apply.

Note

On transmit, if ENET*n*_TCR[ADDINS] is set, bytes 6–11 of each frame can be set to any value, since the MAC overwrites the bytes with the MAC address programmed in the ENET*n*_PAUR and ENET*n*_PALR registers.

Table 11-82. FIFO interface frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–13	Length/type field
14–N	Payload data

VLAN-tagged frames are supported on both transmit and receive, and implement additional information (VLAN type and info).

Table 11-83. FIFO interface VLAN frame format

Byte number	Field
0–5	Destination MAC address
6–11	Source MAC address
12–15	VLAN tag and info
16–17	Length/type field
18–N	Payload data

Note

The standard defines that the LSB of the MAC address is sent/received first, while for all the other header fields — in other words, length/type, VLAN tag, VLAN info, and pause quanta — the MSB is sent/received first.

11.5.5.15.2 Data structure examples

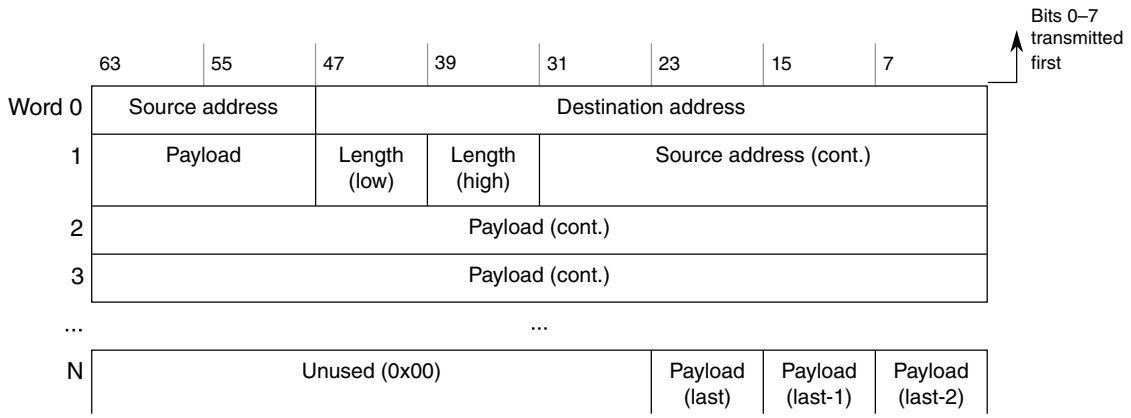


Figure 11-94. Normal Ethernet frame 64-bit mapping example

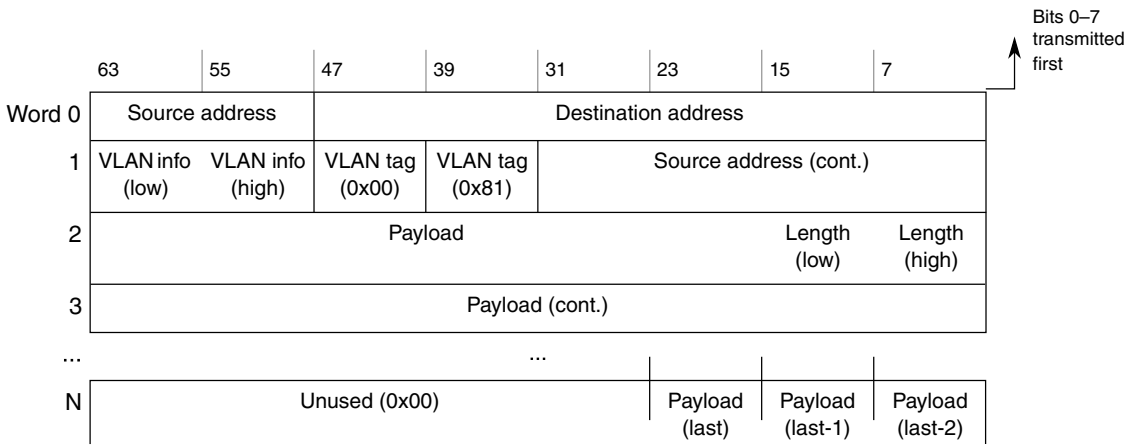


Figure 11-95. VLAN-tagged frame 64-bit mapping example

If CRC forwarding is enabled (CRCFWD = 0), the last four valid octets of the frame contain the FCS field. The non-significant bytes of the last word can have any value.

11.5.5.15.3 Frame status

A MAC layer status word and an accelerator status word is available in the receive buffer descriptor.

See Enhanced buffer descriptors for details.

The status is available with each frame with the last data of the frame.

If the frame status contains a MAC layer error (for example, CRC or length error), RxBD[ME] is also set with the last data of the frame.

11.5.5.16 FIFO protection

The following sections describe the FIFO protection mechanisms.

11.5.5.16.1 Transmit FIFO underflow

During a frame transfer, when the transmit FIFO reaches the almost empty threshold with no end-of-frame indication stored in the FIFO, the MAC logic:

- Stops reading data from the FIFO
- Asserts the MII error signal (MII_TXER) (1 in Figure 11-96) to indicate that the fragment already transferred is not valid
- Deasserts the MII transmit enable signal (MII_TXEN) to terminate the frame transfer (2)

After an underflow, when the application completes the frame transfer (3), the MAC transmit logic discards any new data available in the FIFO until the end of packet is reached (4) and sets the enhanced TxBD[UE] field.

The MAC starts to transfer data on the MII interface when the application sends a new frame with a start of frame indication (5).

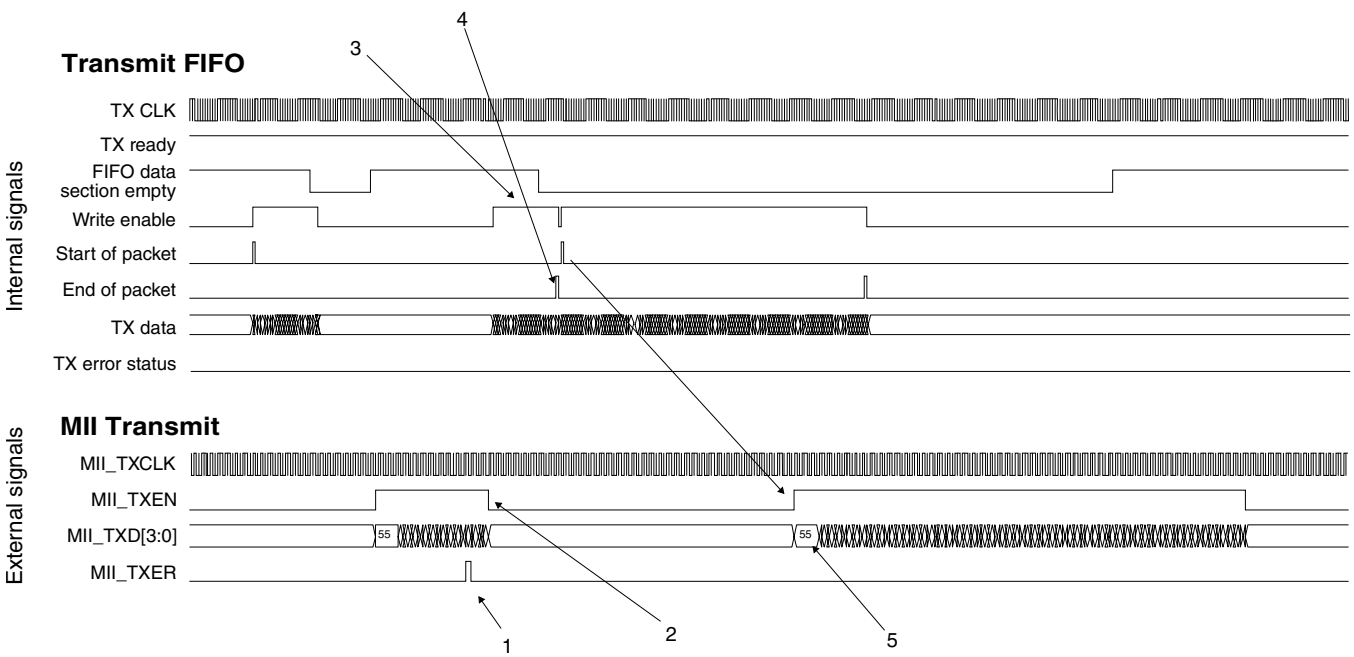


Figure 11-96. Transmit FIFO underflow protection

11.5.5.16.2 Transmit FIFO overflow

On the transmit path, when the FIFO reaches the programmable almost full threshold, the internal MAC ready signal is deasserted. The application should stop sending new data.

However, if the application keeps sending data, the transmit FIFO overflows, corrupting contents that were previously stored. The core logic sets the enhanced TxBD[OE] field for the next frame transmitted to indicate this overflow occurence.

Note

Overflow is a fatal error and must be addressed by resetting the core or clearing ENET*n*_ECR[ETHER_EN], to clear the FIFOs and prepare for normal operation again.

11.5.5.16.3 Receive FIFO overflow

During a frame reception, if the client application is not able to receive data (1), the MAC receive control truncates the incoming frame when the FIFO reaches the programmable almost-full threshold to avoid an overflow.

The frame is subsequently received on the FIFO interface with an error indication (enhanced RxBD[ME] field set together with receive end-of-packet) (2) with the truncation error status field set (3).

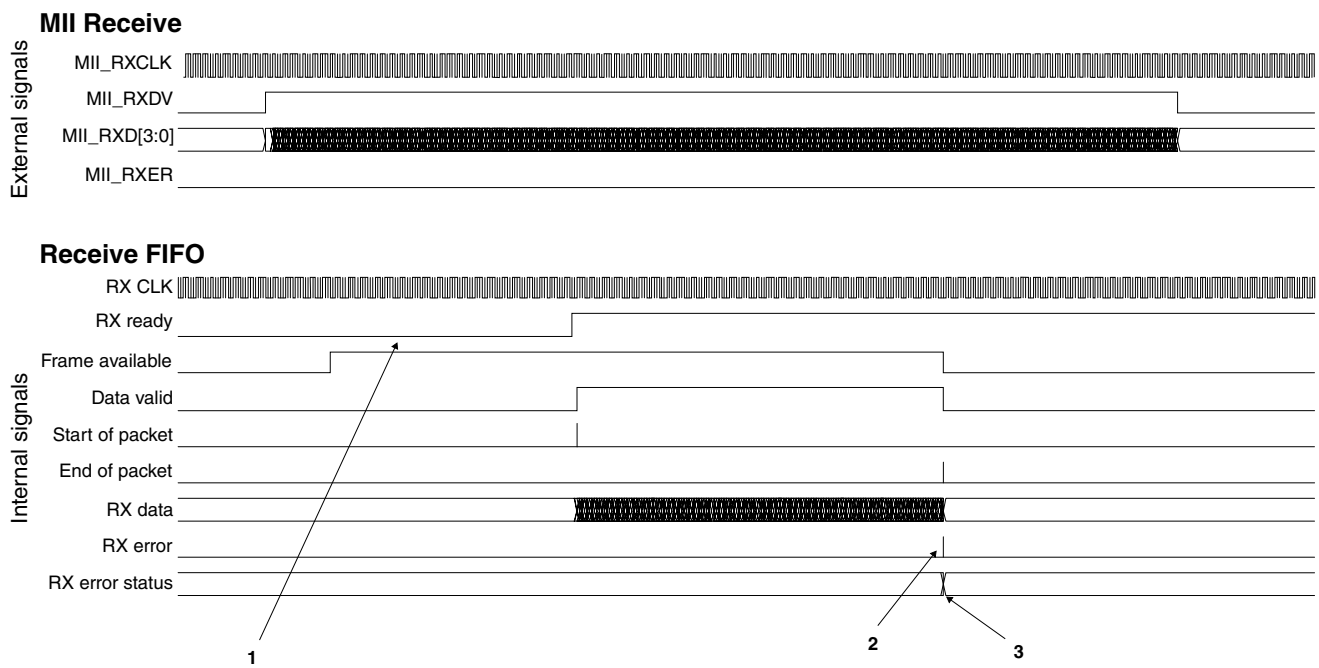


Figure 11-97. Receive FIFO overflow protection

11.5.5.17 PHY management interface

The MDIO interface is a two-wire management interface. The MDIO management interface implements a standardized method to access the PHY device management registers.

The core implements a master MDIO interface, which can be connected to up to 32 PHY devices.

11.5.5.17.1 MDIO clause 22 frame format

The core MDIO master controller communicates with the slave (PHY device) using frames that are defined in the following table.

A complete frame has a length of 64 bits made up of an optional 32-bit preamble, 14-bit command, 2-bit bus direction change, and 16-bit data. Each bit is transferred on the rising edge of the MDIO clock (MDC signal). The MDIO data signal is tri-stated between frames.

The core PHY management interface supports the standard MDIO specification (IEEE 802.3 Clause 22).

Table 11-84. MDIO clause 22 frame structure

ST	OP	PHYADR	REGADR	TA	DATA

Table 11-85. MDIO frame field descriptions

Field	Description
ST (2 bits)	Start indication field, programmed with ENET*n*_MMFR[ST] and equal to 01 for Standard MDIO (Clause 22).
OP (2 bits)	Opcode defines type of operation. Programmed with ENET*n*_MMFR[OP]. 01 Write operation 10 Read operation
PHYADR (5 bits)	Five-bit PHY device address, programmed with ENET*n*_MMFR[PA]. Up to 32 devices can be addressed.
REGADR (5 bits)	Five-bit register address, programmed with ENET*n*_MMFR[RA]. Each PHY can implement up to 32 registers.
TA (2 bits)	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
Data (16 bits)	Data, set by ENET*n*_MMFR[DATA]. Written to or read from the PHY

11.5.5.17.2 MDIO clause 45 frame format

The extended MDIO frame structure defined in IEEE 802.3 Clause 45 introduces indirect addressing. First, a write transaction to an address register is done, followed by a write or read transaction which will put the 16-bit data in the register or retrieve the register contents respectively. A preamble of 32 bits of logical ones is sent prior to every transaction. The MDIO data signal is tri-stated between frames.

The extended MDIO defines four transactions, which are determined by the two-bit opcode field.

Table 11-86. MDIO clause 45 frame structure

ST	OP	PRTAD	DEVAD	TA	ADDR/DATA

All bits are transmitted from left to right (Preamble bits first) and all fields have their Most-Significant bit sent first (leftmost in above table). The complete frame has a length of 64 bits (32-bit preamble, 14-bit command, 2-bit bus direction change, 16-bit data). Each bit is transferred with the rising edge of the MDIO clock (MDC).

The fields and transactions are summarized in the following tables.

Table 11-87. MDIO clause 45 frame field descriptions

Field	Description
ST	Start indication. Indicates the end of the preamble and start of the frame. This value is 00 for extended MDIO (Clause 45) frames.
OP	Opcode defines if a read or write operation is performed and is programmed with ENET*n*_MMFR[OP]. See Table 11-88 for more information. 00 Address write 01 Write operation 10 Read inc. operation 11 Read operation
PRTAD	The port address specifies a MDIO port. Each Port can have up to 32 devices which each can have a separate set of registers.
DEVAD	Device address. Up to 32 devices can be addressed (within a port).
TA	Turnaround time, programmed with ENET*n*_MMFR[TA]. Two bit-times are reserved for read operations to switch the data bus from write to read. The PHY device presents its register contents in the data phase and drives the bus from the second bit of the turnaround phase.
ADDR/DATA	16-bit address (for address write) or data, set by ENET*n*_MMFR[DATA], written to or read from the PHY.

Table 11-88. MDIO Clause 45 Transactions

Transaction Type	Description
Address	A write transaction to the internal address register of the device/port. The data section of the frame contains the value to be stored in the device's internal address "pointer" register for further transactions.
Write	Data write to a register. The 16 bit data will be written to the register identified by the device-internal address.
Read	Data is read from the register identified by the device-internal address.
Read inc.	Read with address postincrement. The register identified by the device-internal address is read. After this, the device-internal address is incremented. If the address register is all '1' (0xFFFF) no increment is done (i.e. increment does not wrap around).

11.5.5.17.3 MDIO clock generation

The MDC clock is generated from the internal bus clock (i.e., IPS bus clock) divided by the value programmed in ENET*n*_MSCR[MII_SPEED].

11.5.5.17.4 MDIO operation

To perform an MDIO access, set the MDIO command register (ENET*n*_MMFR) according to the description provided in MII Management Frame Register (ENETn_MMFR).

To check when the programmed access completes, read the ENET*n*_EIR[MII] field.

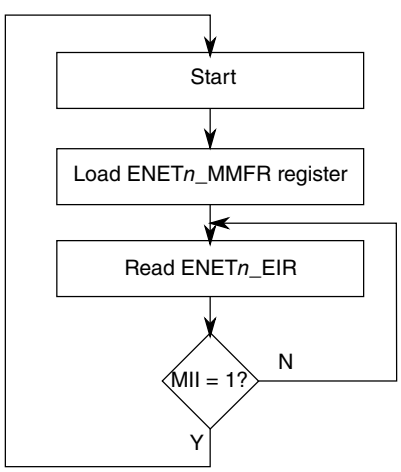


Figure 11-98. MDIO access overview

11.5.5.18 Ethernet interfaces

The following Ethernet interfaces are implemented:

- Fast Ethernet MII (Media Independent Interface)
- RMII 10/100 using interface converters/gaskets
- RGMII 10/100/1000 by way of interface converters/gaskets

The following table shows how to configure ENET registers to select each interface.

Mode	ECR[SPEED]	RCR[RMII_10T]	RCR[RMII_MODE]	RCR[RGMII_EN]
MII - 10 Mbit/s	0	—	0	0
MII - 100 Mbit/s	0	—	0	0
RMII - 10 Mbit/s	0	1	1	0
RMII - 100 Mbit/s	0	0	1	0
RGMII - 10 Mbit/s	0	1	0	1
RGMII - 100 Mbit/s	0	0	0	1
RGMII - 1000 Mbit/s	1	—	0	1

11.5.5.18.1 RMII interface

In RMII receive mode, for normal reception following assertion of CRS_DV, RXD[1:0] is 00 until the receiver determines that the receive event has a proper start-of-stream delimiter (SSD).

The preamble appears (RXD[1:0]=01) and the MACs begin capturing data following detection of SFD.

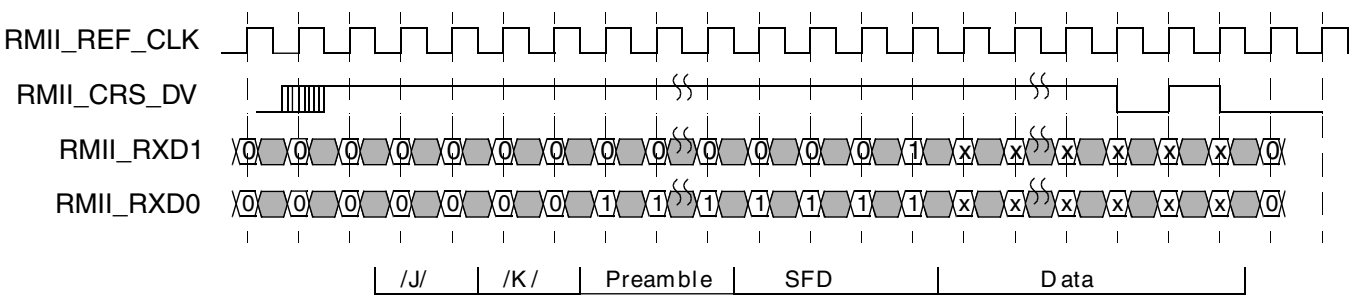


Figure 11-99. RMII receive operation

If a false carrier is detected (bad SSD), then RXD[1:0] is 10 until the end of the receive event. This is a unique pattern since a false carrier can only occur at the beginning of a packet where the preamble is decoded (RXD[1:0] = 01).

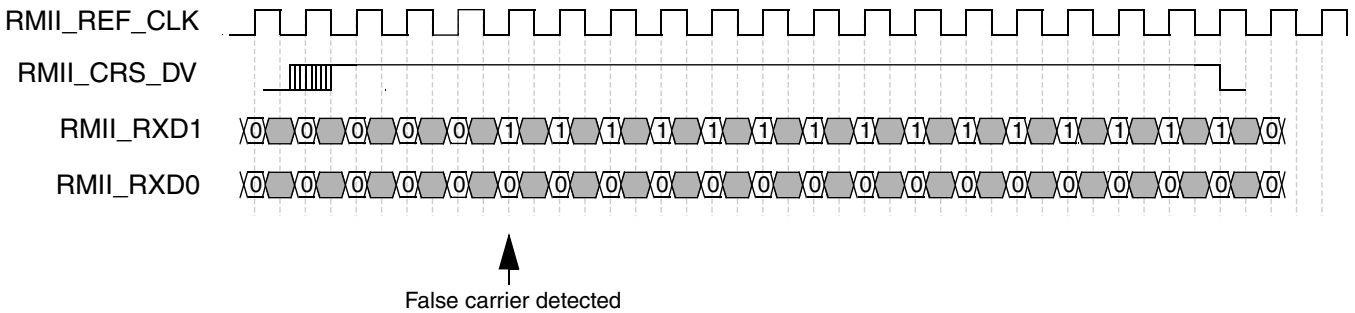


Figure 11-100. RMII receive operation with false carrier

In RMII transmit mode, TXD[1:0] provides valid data for each REF_CLK period while TXEN is asserted.

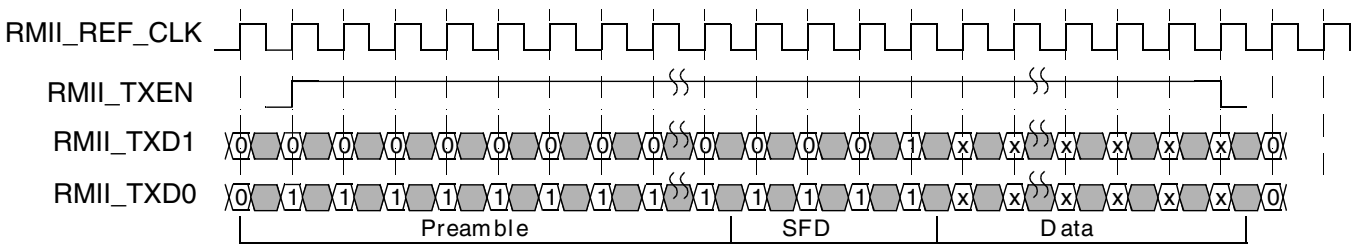


Figure 11-101. RMII transmit operation

11.5.5.18.2 RGMII interface

In RGMII modes, the data and control information is multiplexed by taking advantage of both edges of the reference clocks.

The data signals contain the lower four data bits on the rising edge and the upper four bits on the falling edge. The control signals are multiplexed into a single clock cycle using the same technique.

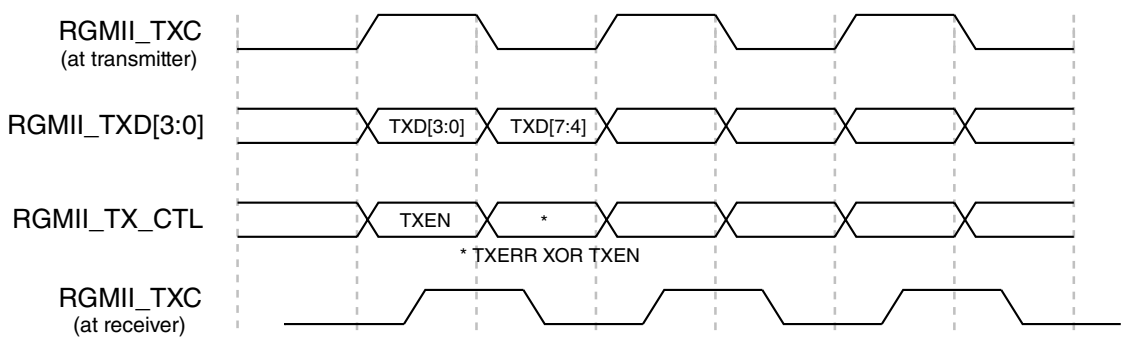


Figure 11-102. RGMII transmit operation

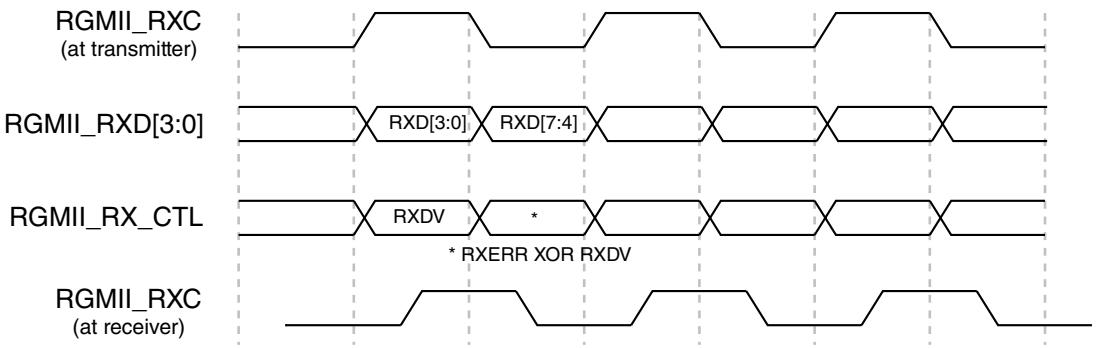


Figure 11-103. RGMII receive operation

11.5.5.18.3 MII Interface — transmit

On transmit, all data transfers are synchronous to MII_TXCLK rising edge. The MII data enable signal MII_TXEN is asserted to indicate the start of a new frame, and remains asserted until the last byte of the frame is present on the MII_TXD[3:0] bus.

Between frames, MII_TXEN remains deasserted.

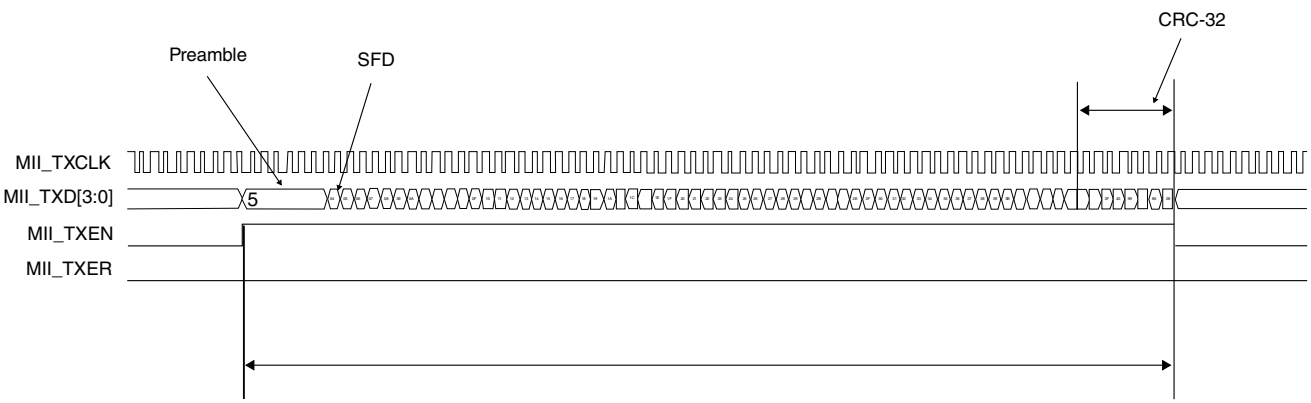


Figure 11-104. MII transmit operation

If a frame is received on the FIFO interface with an error (for example, RxBD[ME] set) the frame is subsequently transmitted with the MII_TXER error signal for one clock cycle at any time during the packet transfer.

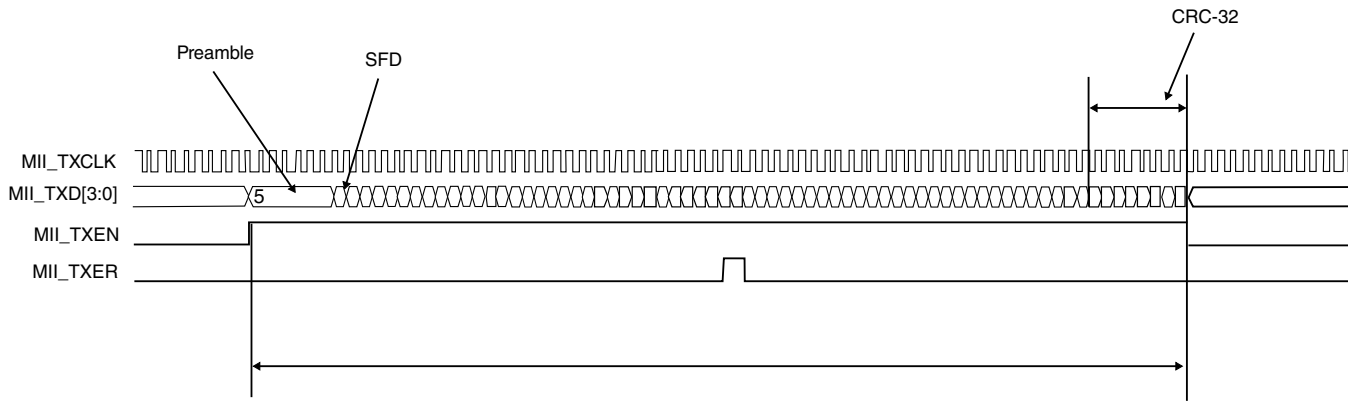


Figure 11-105. MII transmit operation — errored frame

11.5.5.18.3.1 Transmit with collision — half-duplex

When a collision is detected during a frame transmission (MII_COL asserted), the MAC stops the current transmission, sends a 32-bit jam pattern, and re-transmits the current frame.

(See Collision detection in half-duplex mode for details)

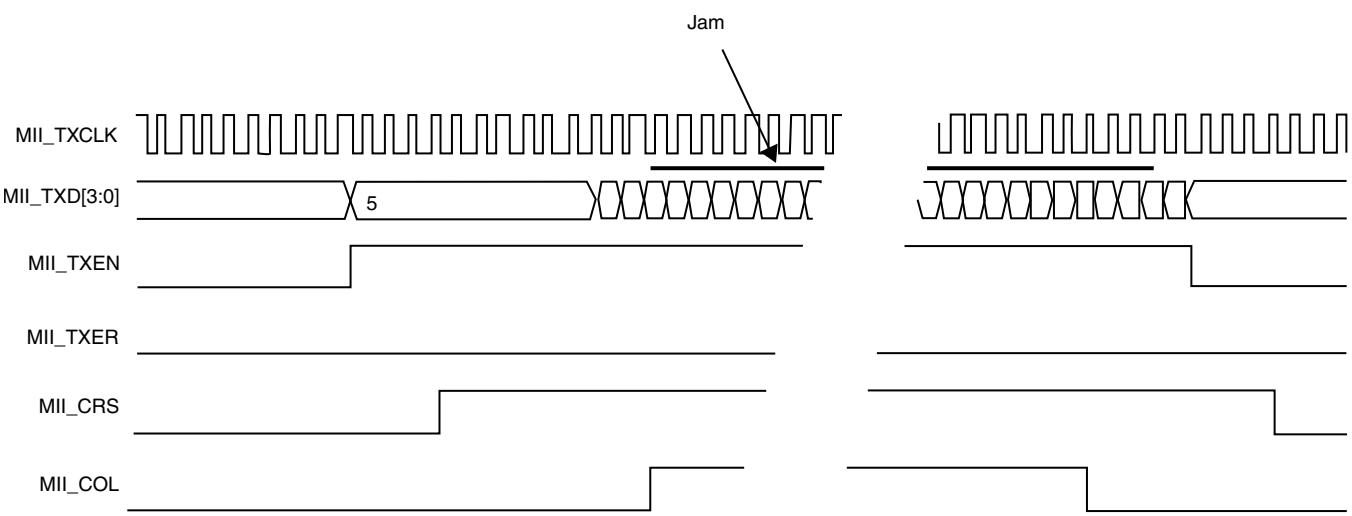


Figure 11-106. MII transmit operation — transmission with collision

11.5.5.18.4 MII interface — receive

On receive, all signals are sampled on the MII_RXCLK rising edge. The MII data enable signal, MII_RXDV, is asserted by the PHY to indicate the start of a new frame and remains asserted until the last byte of the frame is present on MII_RXD[3:0] bus.

Between frames, MII_RXDV remains deasserted.

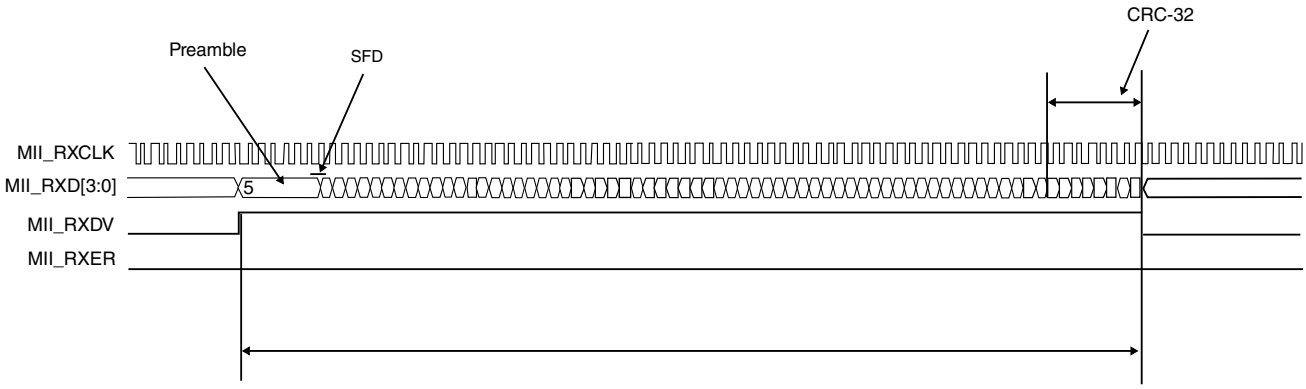


Figure 11-107. MII receive operation

If the PHY detects an error on the frame received from the line, the PHY asserts the MII error signal, MII_RXER, for at least one clock cycle at any time during the packet transfer.

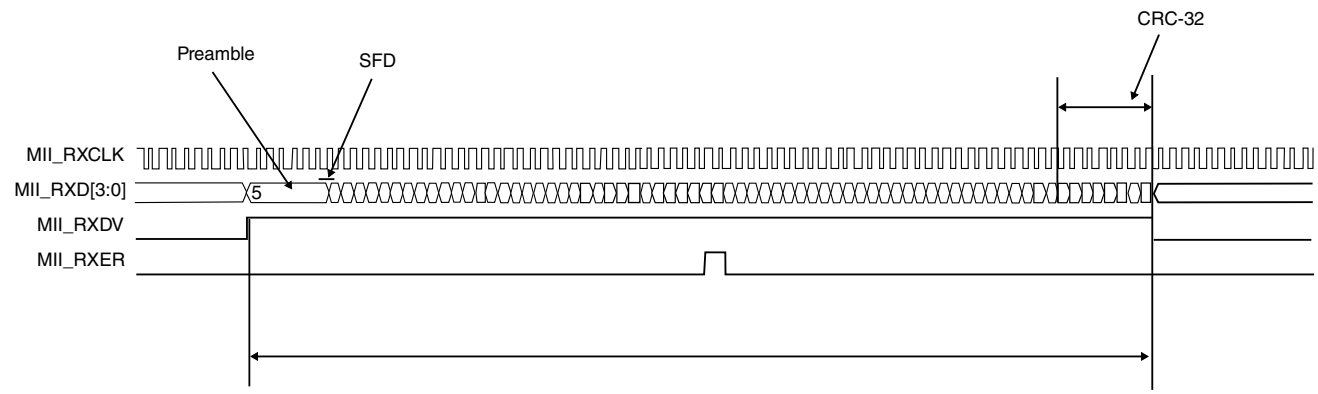


Figure 11-108. MII receive operation — errored frame

A frame received on the MII interface with a PHY error indication is subsequently transferred on the FIFO interface with RxBD[ME] set.

11.5.5.19 AVB configuration

The following steps give an example of how to initialize the ENET module for AVB.

1. Set up ENET_QOS:
   - Set TX_SCHEME to 000b, credit-based scheme.
   - Set RX_FLUSH0 to 1, enable RX flush for ring 0.
2. Set up TX BD ring and RX BD ring for each queue, 0-2.
   - Program ENET_MRBR, ENET_TDSR, and ENET_RDSR.
   - Program ENET_MRBR1, ENET_TDSR1, and ENET_RDSR1.

- Program ENET_MRBR2, ENET_TDSR2, and ENET_RDSR2.
- Program each TX and RX BD ring queue for all classes used in memory.

NOTE

If using credit-based scheme, ensure that enhanced transmit buffer descriptor FTYPE field matches BD ring queue, for example:
- FTYPE = 0h corresponds to ENET_TDSR
- FTYPE = 1h corresponds to ENET_TDSR1
- FTYPE = 2h corresponds to ENET_TDSR2

3. Program ENET_DMA1CFG and ENET_DMA2CFG for class 1 and class 2 corresponding to BD ring queue 1 and 2. DMA_CLASS_EN must be set to one for that class to be enabled. See DMA Class Based Configuration (ENET_DMA*n*CFG) for information on how to program IDLE_SLOPE.
4. Program ENET_RCMR1 and ENET_RCMR2 for class 1 and class 2 matching for receive.

NOTE

Even if a match occurs, if ENET_DMA*n*CFG[DMA_CLASS_EN] is zero for the corresponding class, the RX frame will be automatically forwarded to BD ring queue 0.

5. Program the other ENET registers according to application requirements.
6. Program ENET_RDAR, ENET_RDAR1, ENET_RDAR2, ENET_TDAR, ENET_TDAR1, and ENET_TDAR2 to 0100_0000h according to the classes used.
7. Set ENET_ECR[ETHEREN] to one.

11.5.5.20 Interrupt coalescence

The purpose of the interrupt coalescing is to reduce the number of interrupts generated by the MAC so as to reduce the CPU loading.

To facilitate this interrupt coalescing for each queue, these registers are available with the same control and configuration fields.

- Transmit Interrupt Coalescing Register (ENET_TXIC*n*) where n=0,1,2 for queue/class 0,1,2.
- Receive Interrupt Coalescing Register (ENET_RXIC*n*) where n=0,1,2 for queue/class 0,1,2.

When coalescing is enabled by asserting the corresponding ICEN field and such interrupt is also enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt when the threshold number of frames is reached (defined by ICFT) or when the threshold timer expires (defined by ICTT).

When coalescing is disabled by de-asserting ICEN, but interrupt is enabled by the corresponding interrupt mask of the EIMR register, the MAC generates an interrupt as they are received without using coalescing. Interrupt coalescing is done for each transmit and receive queue/class independently.

11.5.5.20.1 Interrupt coalescence setup

Interrupt coalescence supports both legacy and enhanced BDs. The following guidelines are recommended when setting up interrupt coalescence.

- When the MAC is configured for enhanced (IEEE 1588) mode, that is, enhanced BDs:
  - Set the INT bit in the enhanced received buffer descriptor to one.
  - Set the INT bit in the enhanced transmit buffer descriptor(s) to one.
- Clear the RXB, RXB1, and RXB2 fields in the EIMR register.
- Clear the TXB, TXB1, and TXB2 fields in the EIMR register.

11.5.5.20.2 Updating the frame count threshold on-the-fly

To update the ICFT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new threshold value to the ICFT field.
3. Set ICEN to one.

NOTE

The ICFT field can be updated on-the-fly without disabling the ICEN field. The hardware interrupt will continue and there is a possibility that an interrupt will occur depending on the state of the hardware counter and the previous ICFT value.

11.5.5.20.3 Updating the timer threshold on-the-fly

To update the ICTT field in the RXIC*n* and TXIC*n* registers:

1. Disable interrupt coalescence by clearing the appropriate ICEN field. This will allow the internal interrupt coalescence counter to reset to zero.

NOTE

When disabling interrupt coalescence, if an interrupt event is pending, that is, the interrupt counter is not zero, then an interrupt will occur.

2. Write the new timer value to the ICTT field.
3. Set ICEN to one.

11.5.6 Memory map/register definition

ENET registers must be read or written with 32-bit accesses. Non-32 bit accesses will terminate with an error.

Reserved bits should be written with 0 and ignored on read. Unused registers read zero and a write has no effect.

This table shows Ethernet registers organization.

Table 11-89. Register map summary

Offset Address	Section	Description
0x0000 – 0x01FF	Configuration	Core control and status registers
0x0200 – 0x03FF	Statistics counters	MIB and Remote Network Monitoring (RFC 2819) registers
0x0400 – 0x0430	1588 control	1588 adjustable timer (TSM) and 1588 frame control
0x0600 – 0x07FC	Capture/Compare block	Registers for the Capture/Compare block

ENET memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_0004	Interrupt Event Register (ENET_EIR)	32	w1c	0000_0000h	11.5.6.1/ 3557
30BE_0008	Interrupt Mask Register (ENET_EIMR)	32	R/W	0000_0000h	11.5.6.2/ 3560
30BE_0010	Receive Descriptor Active Register - Ring 0 (ENET_RDAR)	32	R/W	0000_0000h	11.5.6.3/ 3564
30BE_0014	Transmit Descriptor Active Register - Ring 0 (ENET_TDAR)	32	R/W	0000_0000h	11.5.6.4/ 3565
30BE_0024	Ethernet Control Register (ENET_ECR)	32	R/W	See section	11.5.6.5/ 3566

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_0040	MII Management Frame Register (ENET_MMFR)	32	R/W	0000_0000h	11.5.6.6/ 3568
30BE_0044	MII Speed Control Register (ENET_MSCR)	32	R/W	0000_0000h	11.5.6.7/ 3569
30BE_0064	MIB Control Register (ENET_MIBC)	32	R/W	C000_0000h	11.5.6.8/ 3571
30BE_0084	Receive Control Register (ENET_RCR)	32	R/W	05EE_0001h	11.5.6.9/ 3572
30BE_00C4	Transmit Control Register (ENET_TCR)	32	R/W	0000_0000h	11.5.6.10/ 3575
30BE_00E4	Physical Address Lower Register (ENET_PALR)	32	R/W	0000_0000h	11.5.6.11/ 3577
30BE_00E8	Physical Address Upper Register (ENET_PAUR)	32	R/W	0000_8808h	11.5.6.12/ 3577
30BE_00EC	Opcode/Pause Duration Register (ENET_OPD)	32	R/W	0001_0000h	11.5.6.13/ 3578
30BE_00F0	Transmit Interrupt Coalescing Register (ENET_TXIC0)	32	R/W	0000_0000h	11.5.6.14/ 3578
30BE_00F4	Transmit Interrupt Coalescing Register (ENET_TXIC1)	32	R/W	0000_0000h	11.5.6.14/ 3578
30BE_00F8	Transmit Interrupt Coalescing Register (ENET_TXIC2)	32	R/W	0000_0000h	11.5.6.14/ 3578
30BE_0100	Receive Interrupt Coalescing Register (ENET_RXIC0)	32	R/W	0000_0000h	11.5.6.15/ 3579
30BE_0104	Receive Interrupt Coalescing Register (ENET_RXIC1)	32	R/W	0000_0000h	11.5.6.15/ 3579
30BE_0108	Receive Interrupt Coalescing Register (ENET_RXIC2)	32	R/W	0000_0000h	11.5.6.15/ 3579
30BE_0118	Descriptor Individual Upper Address Register (ENET_IAUR)	32	R/W	0000_0000h	11.5.6.16/ 3580
30BE_011C	Descriptor Individual Lower Address Register (ENET_IALR)	32	R/W	0000_0000h	11.5.6.17/ 3581
30BE_0120	Descriptor Group Upper Address Register (ENET_GAUR)	32	R/W	0000_0000h	11.5.6.18/ 3581
30BE_0124	Descriptor Group Lower Address Register (ENET_GALR)	32	R/W	0000_0000h	11.5.6.19/ 3582
30BE_0144	Transmit FIFO Watermark Register (ENET_TFWR)	32	R/W	0000_0000h	11.5.6.20/ 3582
30BE_0160	Receive Descriptor Ring 1 Start Register (ENET_RDSR1)	32	R/W	0000_0000h	11.5.6.21/ 3583
30BE_0164	Transmit Buffer Descriptor Ring 1 Start Register (ENET_TDSR1)	32	R/W	0000_0000h	11.5.6.22/ 3584
30BE_0168	Maximum Receive Buffer Size Register - Ring 1 (ENET_MRBR1)	32	R/W	0000_0000h	11.5.6.23/ 3585

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_016C	Receive Descriptor Ring 2 Start Register (ENET_RDSR2)	32	R/W	0000_0000h	11.5.6.24/ 3586
30BE_0170	Transmit Buffer Descriptor Ring 2 Start Register (ENET_TDSR2)	32	R/W	0000_0000h	11.5.6.25/ 3586
30BE_0174	Maximum Receive Buffer Size Register - Ring 2 (ENET_MRBR2)	32	R/W	0000_0000h	11.5.6.26/ 3587
30BE_0180	Receive Descriptor Ring 0 Start Register (ENET_RDSR)	32	R/W	0000_0000h	11.5.6.27/ 3588
30BE_0184	Transmit Buffer Descriptor Ring 0 Start Register (ENET_TDSR)	32	R/W	0000_0000h	11.5.6.28/ 3589
30BE_0188	Maximum Receive Buffer Size Register - Ring 0 (ENET_MRBR)	32	R/W	0000_0000h	11.5.6.29/ 3589
30BE_0190	Receive FIFO Section Full Threshold (ENET_RSFL)	32	R/W	0000_0000h	11.5.6.30/ 3590
30BE_0194	Receive FIFO Section Empty Threshold (ENET_RSEM)	32	R/W	0000_0000h	11.5.6.31/ 3591
30BE_0198	Receive FIFO Almost Empty Threshold (ENET_RAEM)	32	R/W	0000_0004h	11.5.6.32/ 3591
30BE_019C	Receive FIFO Almost Full Threshold (ENET_RAFL)	32	R/W	0000_0004h	11.5.6.33/ 3592
30BE_01A0	Transmit FIFO Section Empty Threshold (ENET_TSEM)	32	R/W	0000_0000h	11.5.6.34/ 3592
30BE_01A4	Transmit FIFO Almost Empty Threshold (ENET_TAEM)	32	R/W	0000_0004h	11.5.6.35/ 3593
30BE_01A8	Transmit FIFO Almost Full Threshold (ENET_TAFL)	32	R/W	0000_0008h	11.5.6.36/ 3593
30BE_01AC	Transmit Inter-Packet Gap (ENET_TIPG)	32	R/W	0000_000Ch	11.5.6.37/ 3594
30BE_01B0	Frame Truncation Length (ENET_FTRL)	32	R/W	0000_07FFh	11.5.6.38/ 3594
30BE_01C0	Transmit Accelerator Function Configuration (ENET_TACC)	32	R/W	0000_0000h	11.5.6.39/ 3595
30BE_01C4	Receive Accelerator Function Configuration (ENET_RACC)	32	R/W	0000_0000h	11.5.6.40/ 3596
30BE_01C8	Receive Classification Match Register for Class n (ENET_RCMR1)	32	R/W	0000_0000h	11.5.6.41/ 3597
30BE_01CC	Receive Classification Match Register for Class n (ENET_RCMR2)	32	R/W	0000_0000h	11.5.6.41/ 3597
30BE_01D8	DMA Class Based Configuration (ENET_DMA1CFG)	32	R/W	0000_0000h	11.5.6.42/ 3598
30BE_01DC	DMA Class Based Configuration (ENET_DMA2CFG)	32	R/W	0000_0000h	11.5.6.42/ 3598
30BE_01E0	Receive Descriptor Active Register - Ring 1 (ENET_RDAR1)	32	R/W	0000_0000h	11.5.6.43/ 3600

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_01E4	Transmit Descriptor Active Register - Ring 1 (ENET_TDAR1)	32	R/W	0000_0000h	11.5.6.44/ 3601
30BE_01E8	Receive Descriptor Active Register - Ring 2 (ENET_RDAR2)	32	R/W	0000_0000h	11.5.6.45/ 3602
30BE_01EC	Transmit Descriptor Active Register - Ring 2 (ENET_TDAR2)	32	R/W	0000_0000h	11.5.6.46/ 3603
30BE_01F0	QOS Scheme (ENET_QOS)	32	R/W	0000_0000h	11.5.6.47/ 3603
30BE_0200	Reserved Statistic Register (ENET_RMON_T_DROP)	32	R	0000_0000h	11.5.6.48/ 3605
30BE_0204	Tx Packet Count Statistic Register (ENET_RMON_T_PACKETS)	32	R	0000_0000h	11.5.6.49/ 3605
30BE_0208	Tx Broadcast Packets Statistic Register (ENET_RMON_T_BC_PKT)	32	R	0000_0000h	11.5.6.50/ 3606
30BE_020C	Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)	32	R	0000_0000h	11.5.6.51/ 3606
30BE_0210	Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)	32	R	0000_0000h	11.5.6.52/ 3607
30BE_0214	Tx Packets Less Than Bytes and Good CRC Statistic Register (ENET_RMON_T_UNDERSIZE)	32	R	0000_0000h	11.5.6.53/ 3607
30BE_0218	Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (ENET_RMON_T_OVERSIZE)	32	R	0000_0000h	11.5.6.54/ 3607
30BE_021C	Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_T_FRAG)	32	R	0000_0000h	11.5.6.55/ 3608
30BE_0220	Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (ENET_RMON_T_JAB)	32	R	0000_0000h	11.5.6.56/ 3608
30BE_0224	Tx Collision Count Statistic Register (ENET_RMON_T_COL)	32	R	0000_0000h	11.5.6.57/ 3609
30BE_0228	Tx 64-Byte Packets Statistic Register (ENET_RMON_T_P64)	32	R	0000_0000h	11.5.6.58/ 3609
30BE_022C	Tx 65- to 127-byte Packets Statistic Register (ENET_RMON_T_P65TO127)	32	R	0000_0000h	11.5.6.59/ 3609
30BE_0230	Tx 128- to 255-byte Packets Statistic Register (ENET_RMON_T_P128TO255)	32	R	0000_0000h	11.5.6.60/ 3610
30BE_0234	Tx 256- to 511-byte Packets Statistic Register (ENET_RMON_T_P256TO511)	32	R	0000_0000h	11.5.6.61/ 3610
30BE_0238	Tx 512- to 1023-byte Packets Statistic Register (ENET_RMON_T_P512TO1023)	32	R	0000_0000h	11.5.6.62/ 3611
30BE_023C	Tx 1024- to 2047-byte Packets Statistic Register (ENET_RMON_T_P1024TO2047)	32	R	0000_0000h	11.5.6.63/ 3611
30BE_0240	Tx Packets Greater Than 2048 Bytes Statistic Register (ENET_RMON_T_P_GTE2048)	32	R	0000_0000h	11.5.6.64/ 3612
30BE_0244	Tx Octets Statistic Register (ENET_RMON_T_OCTETS)	32	R	0000_0000h	11.5.6.65/ 3612

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_0248	Reserved Statistic Register (ENET_IEEE_T_DROP)	32	R	0000_0000h	11.5.6.66/ 3612
30BE_024C	Frames Transmitted OK Statistic Register (ENET_IEEE_T_FRAME_OK)	32	R	0000_0000h	11.5.6.67/ 3613
30BE_0250	Frames Transmitted with Single Collision Statistic Register (ENET_IEEE_T_1COL)	32	R	0000_0000h	11.5.6.68/ 3613
30BE_0254	Frames Transmitted with Multiple Collisions Statistic Register (ENET_IEEE_T_MCOL)	32	R	0000_0000h	11.5.6.69/ 3614
30BE_0258	Frames Transmitted after Deferral Delay Statistic Register (ENET_IEEE_T_DEF)	32	R	0000_0000h	11.5.6.70/ 3614
30BE_025C	Frames Transmitted with Late Collision Statistic Register (ENET_IEEE_T_LCOL)	32	R	0000_0000h	11.5.6.71/ 3614
30BE_0260	Frames Transmitted with Excessive Collisions Statistic Register (ENET_IEEE_T_EXCOL)	32	R	0000_0000h	11.5.6.72/ 3615
30BE_0264	Frames Transmitted with Tx FIFO Underrun Statistic Register (ENET_IEEE_T_MACERR)	32	R	0000_0000h	11.5.6.73/ 3615
30BE_0268	Frames Transmitted with Carrier Sense Error Statistic Register (ENET_IEEE_T_CSERR)	32	R	0000_0000h	11.5.6.74/ 3616
30BE_026C	Reserved Statistic Register (ENET_IEEE_T_SQE)	32	R (reads 0)	0000_0000h	11.5.6.75/ 3616
30BE_0270	Flow Control Pause Frames Transmitted Statistic Register (ENET_IEEE_T_FDXFC)	32	R	0000_0000h	11.5.6.76/ 3616
30BE_0274	Octet Count for Frames Transmitted w/o Error Statistic Register (ENET_IEEE_T_OCTETS_OK)	32	R	0000_0000h	11.5.6.77/ 3617
30BE_0284	Rx Packet Count Statistic Register (ENET_RMON_R_PACKETS)	32	R	0000_0000h	11.5.6.78/ 3617
30BE_0288	Rx Broadcast Packets Statistic Register (ENET_RMON_R_BC_PKT)	32	R	0000_0000h	11.5.6.79/ 3618
30BE_028C	Rx Multicast Packets Statistic Register (ENET_RMON_R_MC_PKT)	32	R	0000_0000h	11.5.6.80/ 3618
30BE_0290	Rx Packets with CRC/Align Error Statistic Register (ENET_RMON_R_CRC_ALIGN)	32	R	0000_0000h	11.5.6.81/ 3618
30BE_0294	Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (ENET_RMON_R_UNDERSIZE)	32	R	0000_0000h	11.5.6.82/ 3619
30BE_0298	Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (ENET_RMON_R_OVERSIZE)	32	R	0000_0000h	11.5.6.83/ 3619
30BE_029C	Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_R_FRAG)	32	R	0000_0000h	11.5.6.84/ 3620
30BE_02A0	Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (ENET_RMON_R_JAB)	32	R	0000_0000h	11.5.6.85/ 3620
30BE_02A4	Reserved Statistic Register (ENET_RMON_R_RESVD_0)	32	R (reads 0)	0000_0000h	11.5.6.86/ 3620
30BE_02A8	Rx 64-Byte Packets Statistic Register (ENET_RMON_R_P64)	32	R	0000_0000h	11.5.6.87/ 3621

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_02AC	Rx 65- to 127-Byte Packets Statistic Register (ENET_RMON_R_P65TO127)	32	R	0000_0000h	11.5.6.88/ 3621
30BE_02B0	Rx 128- to 255-Byte Packets Statistic Register (ENET_RMON_R_P128TO255)	32	R	0000_0000h	11.5.6.89/ 3622
30BE_02B4	Rx 256- to 511-Byte Packets Statistic Register (ENET_RMON_R_P256TO511)	32	R	0000_0000h	11.5.6.90/ 3622
30BE_02B8	Rx 512- to 1023-Byte Packets Statistic Register (ENET_RMON_R_P512TO1023)	32	R	0000_0000h	11.5.6.91/ 3622
30BE_02BC	Rx 1024- to 2047-Byte Packets Statistic Register (ENET_RMON_R_P1024TO2047)	32	R	0000_0000h	11.5.6.92/ 3623
30BE_02C0	Rx Packets Greater than 2048 Bytes Statistic Register (ENET_RMON_R_P_GTE2048)	32	R	0000_0000h	11.5.6.93/ 3623
30BE_02C4	Rx Octets Statistic Register (ENET_RMON_R_OCTETS)	32	R	0000_0000h	11.5.6.94/ 3624
30BE_02C8	Frames not Counted Correctly Statistic Register (ENET_IEEE_R_DROP)	32	R	0000_0000h	11.5.6.95/ 3624
30BE_02CC	Frames Received OK Statistic Register (ENET_IEEE_R_FRAME_OK)	32	R	0000_0000h	11.5.6.96/ 3624
30BE_02D0	Frames Received with CRC Error Statistic Register (ENET_IEEE_R_CRC)	32	R	0000_0000h	11.5.6.97/ 3625
30BE_02D4	Frames Received with Alignment Error Statistic Register (ENET_IEEE_R_ALIGN)	32	R	0000_0000h	11.5.6.98/ 3625
30BE_02D8	Receive FIFO Overflow Count Statistic Register (ENET_IEEE_R_MACERR)	32	R	0000_0000h	11.5.6.99/ 3626
30BE_02DC	Flow Control Pause Frames Received Statistic Register (ENET_IEEE_R_FDXFC)	32	R	0000_0000h	11.5.6.100/ 3626
30BE_02E0	Octet Count for Frames Received without Error Statistic Register (ENET_IEEE_R_OCTETS_OK)	32	R	0000_0000h	11.5.6.101/ 3626
30BE_0400	Adjustable Timer Control Register (ENET_ATCR)	32	R/W	0000_0000h	11.5.6.102/ 3627
30BE_0404	Timer Value Register (ENET_ATVR)	32	R/W	0000_0000h	11.5.6.103/ 3629
30BE_0408	Timer Offset Register (ENET_ATOFF)	32	R/W	0000_0000h	11.5.6.104/ 3629
30BE_040C	Timer Period Register (ENET_ATPER)	32	R/W	3B9A_CA00h	11.5.6.105/ 3630
30BE_0410	Timer Correction Register (ENET_ATCOR)	32	R/W	0000_0000h	11.5.6.106/ 3630
30BE_0414	Time-Stamping Clock Period Register (ENET_ATINC)	32	R/W	0000_0000h	11.5.6.107/ 3631
30BE_0418	Timestamp of Last Transmitted Frame (ENET_ATSTMP)	32	R	0000_0000h	11.5.6.108/ 3631
30BE_0604	Timer Global Status Register (ENET_TGSR)	32	R/W	0000_0000h	11.5.6.109/ 3632

Table continues on the next page...

ENET memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30BE_0608	Timer Control Status Register (ENET_TCSR0)	32	R/W	0000_0000h	11.5.6.110/ 3633
30BE_060C	Timer Compare Capture Register (ENET_TCCR0)	32	R/W	0000_0000h	11.5.6.111/ 3634
30BE_0610	Timer Control Status Register (ENET_TCSR1)	32	R/W	0000_0000h	11.5.6.110/ 3633
30BE_0614	Timer Compare Capture Register (ENET_TCCR1)	32	R/W	0000_0000h	11.5.6.111/ 3634
30BE_0618	Timer Control Status Register (ENET_TCSR2)	32	R/W	0000_0000h	11.5.6.110/ 3633
30BE_061C	Timer Compare Capture Register (ENET_TCCR2)	32	R/W	0000_0000h	11.5.6.111/ 3634
30BE_0620	Timer Control Status Register (ENET_TCSR3)	32	R/W	0000_0000h	11.5.6.110/ 3633
30BE_0624	Timer Compare Capture Register (ENET_TCCR3)	32	R/W	0000_0000h	11.5.6.111/ 3634

11.5.6.1 Interrupt Event Register (ENET_EIR)

When an event occurs that sets a bit in EIR, an interrupt occurs if the corresponding bit in the interrupt mask register (EIMR) is also set. Writing a 1 to an EIR bit clears it; writing 0 has no effect. This register is cleared upon hardware reset.

NOTE

TxBD[INT] and RxBD[INT] must be set to 1 to allow setting the corresponding EIR register flags in enhanced mode, ENET_ECR[EN1588] = 1. Legacy mode does not require these flags to be enabled.

Address: 30BE_0000h base + 4h offset = 30BE_0004h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W		w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	RXFLUSH_2	RXFLUSH_1	RXFLUSH_0					TXF2	TXB2	RXF2	RXB2	TXF1	TXB1	RXF1	RXB1
W	w1c	w1c	w1c	w1c	0			0	w1c	w1c	w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_EIR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
30 BABR	Babbling Receive Error Indicates a frame was received with length in excess of RCR[MAX_FL] bytes.
29 BABT	Babbling Transmit Error Indicates the transmitted frame length exceeds RCR[MAX_FL] bytes. Usually this condition is caused when a frame that is too long is placed into the transmit data buffer(s). Truncation does not occur.
28 GRA	Graceful Stop Complete This interrupt is asserted after the transmitter is put into a pause state after completion of the frame currently being transmitted. See Graceful Transmit Stop (GTS) for conditions that lead to graceful stop. **NOTE:** The GRA interrupt is asserted only when the TX transitions into the stopped state. If this bit is cleared by writing 1 and the TX is still stopped, the bit is not set again.
27 TXF	Transmit Frame Interrupt Indicates a frame has been transmitted and the last corresponding buffer descriptor has been updated.
26 TXB	Transmit Buffer Interrupt Indicates a transmit buffer descriptor has been updated.
25 RXF	Receive Frame Interrupt Indicates a frame has been received and the last corresponding buffer descriptor has been updated.
24 RXB	Receive Buffer Interrupt Indicates a receive buffer descriptor is not the last in the frame has been updated.
23 MII	MII Interrupt. Indicates that the MII has completed the data transfer requested.
22 EBERR	Ethernet Bus Error Indicates a system bus error occurred when a uDMA transaction is underway. When this bit is set, ECR[ETHEREN] is cleared, halting frame processing by the MAC. When this occurs, software must ensure proper actions, possibly resetting the system, to resume normal operation.

Table continues on the next page...

ENET_EIR field descriptions (continued)

Field	Description
21 LC	Late Collision Indicates a collision occurred beyond the collision window (slot time) in half-duplex mode. The frame truncates with a bad CRC and the remainder of the frame is discarded.
20 RL	Collision Retry Limit Indicates a collision occurred on each of 16 successive attempts to transmit the frame. The frame is discarded without being transmitted and transmission of the next frame commences. This error can only occur in half-duplex mode.
19 UN	Transmit FIFO Underrun Indicates the transmit FIFO became empty before the complete frame was transmitted. **NOTE:** In situations where the device has various masters generating high traffic, a FIFO underrun can occur on the transmit FIFO. To avoid transmit FIFO underrun, store and forward can be enabled in ENET_TFWR[STRFWD]. See STRFWD. Also, a higher priority can be set for ENET traffic using available means on the central bus fabric connecting the ENET module.
18 PLR	Payload Receive Error Indicates a frame was received with a payload length error. See Frame Length/Type Verification: Payload Length Check for more information.
17 WAKEUP	Node Wakeup Request Indication Read-only status bit to indicate that a magic packet has been detected. Will act only if ECR[MAGICEN] is set.
16 TS_AVAIL	Transmit Timestamp Available Indicates that the timestamp of the last transmitted timing frame is available in the ATSTMP register.
15 TS_TIMER	Timestamp Timer The adjustable timer reached the period event. A period event interrupt can be generated if ATCR[PEREN] is set and the timer wraps according to the periodic setting in the ATPER register. Set the timer period value before setting ATCR[PEREN].
14 RXFLUSH_2	RX DMA Ring 2 flush indication. This ring's RX frame has been flushed due to either RDAR2[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_2] is enabled.
13 RXFLUSH_1	RX DMA Ring 1 flush indication. This ring's RX frame has been flushed due to either RDAR1[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_1] is enabled.
12 RXFLUSH_0	RX DMA Ring 0 flush indication. . This ring's RX frame has been flushed due to either RDAR[RDAR] or RxBD[E] being clear and only if QOS[RXFLUSH_0] is enabled.
11–9 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
7 TXF2	Transmit frame interrupt, class 2 This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 2).
6 TXB2	Transmit buffer interrupt, class 2 This field indicates that a transmit buffer descriptor has been updated (ring/class 2).

Table continues on the next page...

ENET_EIR field descriptions (continued)

Field	Description
5 RXF2	Receive frame interrupt, class 2 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 2).
4 RXB2	Receive buffer interrupt, class 2 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 2).
3 TXF1	Transmit frame interrupt, class 1 This bit indicates that a frame has been transmitted and the last corresponding buffer descriptor has been updated (ring/class 1).
2 TXB1	Transmit buffer interrupt, class 1 This field indicates that a transmit buffer descriptor has been updated (ring/class 1).
1 RXF1	Receive frame interrupt, class 1 This field indicates that a frame has been received and the last corresponding buffer descriptor has been updated (ring/class 1).
0 RXB1	Receive buffer interrupt, class 1 This field indicates that a receive buffer descriptor, that not the last in the frame, has been updated (ring/class 1).

11.5.6.2 Interrupt Mask Register (ENET_EIMR)

EIMR controls which interrupt events are allowed to generate actual interrupts. A hardware reset clears this register. If the corresponding bits in the EIR and EIMR registers are set, an interrupt is generated. The interrupt signal remains asserted until a 1 is written to the EIR field (write 1 to clear) or a 0 is written to the EIMR field.

Address: 30BE_0000h base + 8h offset = 30BE_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R		BABR	BABT	GRA	TXF	TXB	RXF	RXB	MII	EBERR	LC	RL	UN	PLR	WAKEUP	TS_AVAIL
W	0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TS_TIMER	RXFLUSH_2	RXFLUSH_1	RXFLUSH_0					TXF2	TXB2	RXF2	RXB2	TXF1	TXB1	RXF1	RXB1
W					0			0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_EIMR field descriptions

Field	Description
31 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
30 BABR	BABR Interrupt Mask Corresponds to interrupt source EIR[BABR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
29 BABT	BABT Interrupt Mask Corresponds to interrupt source EIR[BABT] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR BABT field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
28 GRA	GRA Interrupt Mask Corresponds to interrupt source EIR[GRA] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR GRA field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
27 TXF	TXF Interrupt Mask Corresponds to interrupt source EIR[TXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
26 TXB	TXB Interrupt Mask Corresponds to interrupt source EIR[TXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared. 0 The corresponding interrupt source is masked. 1 The corresponding interrupt source is not masked.
25 RXF	RXF Interrupt Mask Corresponds to interrupt source EIR[RXF] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXF field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

Table continues on the next page...

ENET_EIMR field descriptions (continued)

Field	Description
24 RXB	RXB Interrupt Mask Corresponds to interrupt source EIR[RXB] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RXB field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
23 MII	MII Interrupt Mask Corresponds to interrupt source EIR[MII] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR MII field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
22 EBERR	EBERR Interrupt Mask Corresponds to interrupt source EIR[EBERR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR EBERR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
21 LC	LC Interrupt Mask Corresponds to interrupt source EIR[LC] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR LC field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
20 RL	RL Interrupt Mask Corresponds to interrupt source EIR[RL] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR RL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
19 UN	UN Interrupt Mask Corresponds to interrupt source EIR[UN] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR UN field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
18 PLR	PLR Interrupt Mask Corresponds to interrupt source EIR[PLR] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR PLR field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
17 WAKEUP	WAKEUP Interrupt Mask Corresponds to interrupt source EIR[WAKEUP] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR WAKEUP field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
16 TS_AVAIL	TS_AVAIL Interrupt Mask Corresponds to interrupt source EIR[TS_AVAIL] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting

Table continues on the next page...

ENET_EIMR field descriptions (continued)

Field	Description
	source. The corresponding EIR TS_AVAIL field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
15 TS_TIMER	TS_TIMER Interrupt Mask Corresponds to interrupt source EIR[TS_TIMER] register and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. The corresponding EIR TS_TIMER field reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
14 RXFLUSH_2	Corresponds to interrupt source EIR[RXFLUSH_2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
13 RXFLUSH_1	Corresponds to interrupt source EIR[RXFLUSH_1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
12 RXFLUSH_0	Corresponds to interrupt source EIR[RXFLUSH_0] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXFLUSH_0] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
11–9 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
7 TXF2	Transmit frame interrupt, class 2 Corresponds to interrupt source EIR[TXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
6 TXB2	Transmit buffer interrupt, class 2 Corresponds to interrupt source EIR[TXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
5 RXF2	Receive frame interrupt, class 2 Corresponds to interrupt source EIR[RXF2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
4 RXB2	Receive buffer interrupt, class 2 Corresponds to interrupt source EIR[RXB2] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB2] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
3 TXF1	Transmit frame interrupt, class 1 Corresponds to interrupt source EIR[TXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
2 TXB1	Transmit buffer interrupt, class 1

Table continues on the next page...

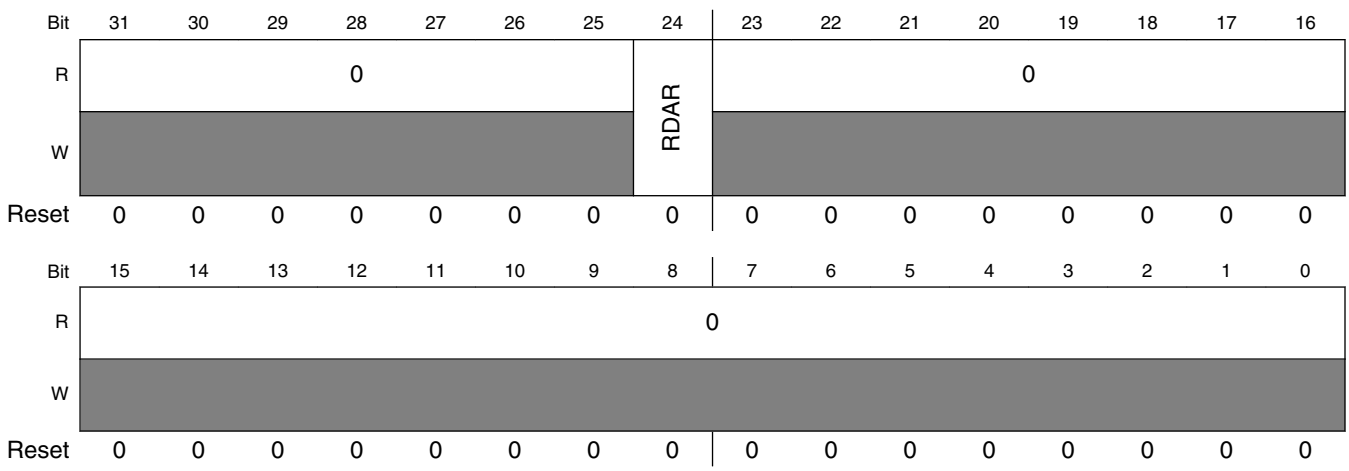
ENET_EIMR field descriptions (continued)

Field	Description
	Corresponds to interrupt source EIR[TXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[TXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
1 RXF1	Receive frame interrupt, class 1 Corresponds to interrupt source EIR[RXF1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXF1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.
0 RXB1	Receive buffer interrupt, class 1 Corresponds to interrupt source EIR[RXB1] and determines whether an interrupt condition can generate an interrupt. At every module clock, the EIR samples the signal generated by the interrupting source. EIR[RXB1] reflects the state of the interrupt signal even if the corresponding EIMR field is cleared.

11.5.6.3 Receive Descriptor Active Register - Ring 0 (ENET_RDAR)

RDAR is a command register, written by the user, to indicate that the receive descriptor ring 0 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Address: 30BE_0000h base + 10h offset = 30BE_0010h



ENET_RDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDAR	Receive Descriptor Active

Table continues on the next page...

ENET_RDAR field descriptions (continued)

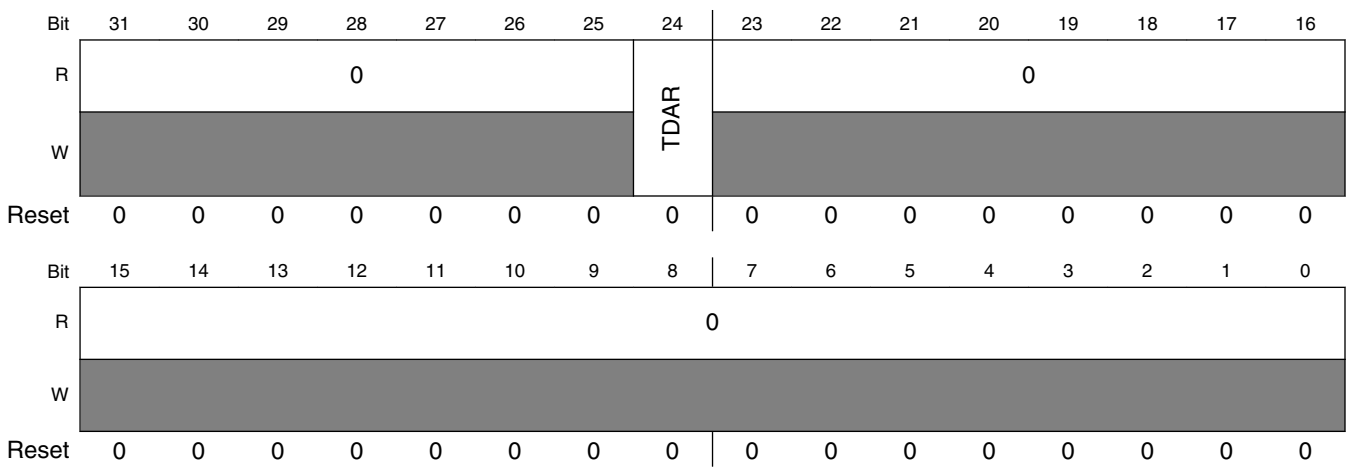
Field	Description
	Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.4 Transmit Descriptor Active Register - Ring 0 (ENET_TDAR)

The TDAR is a command register that the user writes to indicate that the transmit descriptor ring 0 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Address: 30BE_0000h base + 14h offset = 30BE_0014h



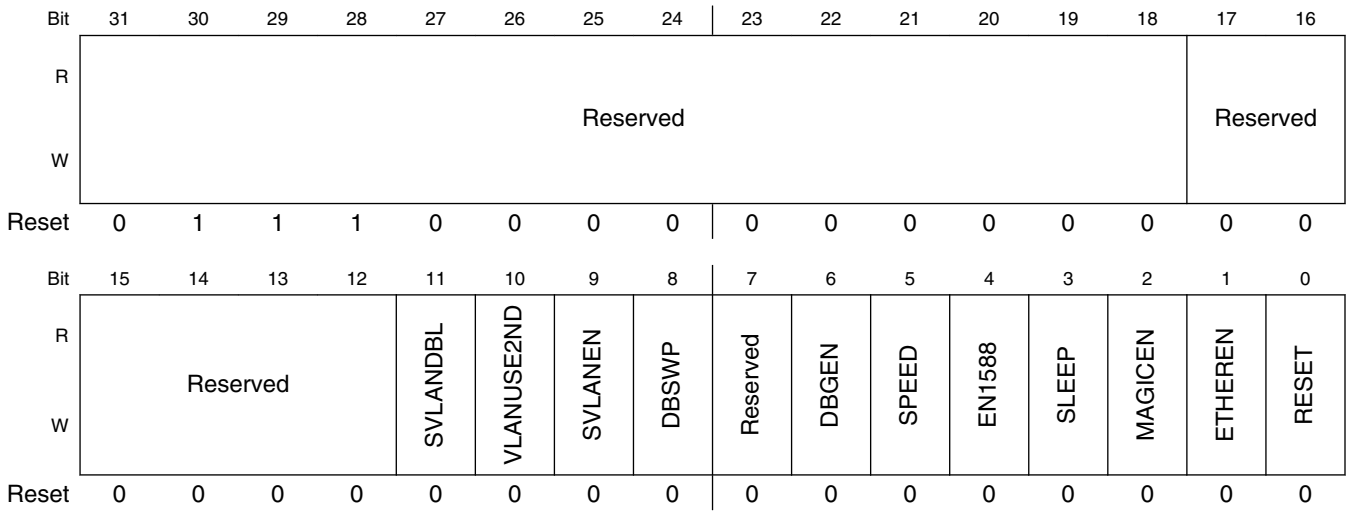
ENET_TDAR field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.5 Ethernet Control Register (ENET_ECR)

ECR is a read/write user register, though hardware may also alter fields in this register. It controls many of the high level features of the Ethernet MAC, including legacy FEC support through the EN1588 field.

Address: 30BE_0000h base + 24h offset = 30BE_0024h



ENET_ECR field descriptions

Field	Description
31–18 Reserved	This field is reserved. Always write 01110000000000b to this field.
17–12 Reserved	This field is reserved. Always write 0 to this field.
11 SVLANDBL	S-VLAN double tag If enabled, S-VLAN detection requires a double-tagged frame to define a frame as being a VLAN frame. The following rules apply: • If the first tag is the S-VLAN type, it must be followed by a second tag with the C-VLAN type to declare the frame as VLAN frame. • If the first tag is the S-VLAN type but no 2nd tag follows, it is not considered a VLAN frame but instead treated as an untagged frame. • If the first tag is the C-VLAN type, it is considered a VLAN frame as normal. **NOTE:** VLANUSE2ND can be used to determine from which tag the data should be extracted. This applies only if SVLAN_EN = 1, ignored otherwise.
10 VLANUSE2ND	VLAN use second tag 0 Always extract data from the first VLAN tag if it exists. 1 When a double-tagged frame is detected, the data of the second tag is extracted for further processing. A double-tagged frame is defined as: • The first tag can be a C-VLAN or a S-VLAN (if SVLAN_ENA = 1) • The second tag must be a C-VLAN

Table continues on the next page...

ENET_ECR field descriptions (continued)

Field	Description
9 SVLANEN	S-VLAN enable Enable additional detection of S-VLAN tag according to IEEE802.1Q. 0 Only the EtherType 0x8100 will be considered for VLAN detection. 1 The EtherType 0x88a8 will be considered in addition to 0x8100 (C-VLAN) to identify a VLAN frame in receive. When a VLAN frame is identified, the two bytes following the VLAN type are extracted and used by the classification match comparators, RCMRn.
8 DBSWP	Descriptor Byte Swapping Enable Swaps the byte locations of the buffer descriptors. **NOTE:** This field resets to 0 and must not be changed. 0 The buffer descriptor bytes are not swapped to support big-endian devices. 1 The buffer descriptor bytes are swapped to support little-endian devices.
7 Reserved	This field is reserved. Always write 0 to this field.
6 DBGEN	Debug Enable Enables the MAC to enter hardware freeze mode when the device enters debug mode. 0 MAC continues operation in debug mode. 1 MAC enters hardware freeze mode when the processor is in debug mode.
5 SPEED	Selects between 10/100-Mbit/s and 1000-Mbit/s modes of operation. 0 10/100-Mbit/s mode 1 1000-Mbit/s mode
4 EN1588	EN1588 Enable Enables enhanced functionality of the MAC. 0 Legacy FEC buffer descriptors and functions enabled. 1 Enhanced frame time-stamping functions enabled.
3 SLEEP	Sleep Mode Enable 0 Normal operating mode. 1 Sleep mode.
2 MAGICEN	Magic Packet Detection Enable Enables/disables magic packet detection. **NOTE:** MAGICEN is relevant only if the SLEEP field is set. If MAGICEN is set, changing the SLEEP field enables/disables sleep mode and magic packet detection. **NOTE:** EIMR[WAKEUP] must be written to one if Magic packet wakeup is programed to wake up the chip from low power mode. 0 Magic detection logic disabled. 1 The MAC core detects magic packets and asserts EIR[WAKEUP] when a frame is detected.
1 ETHEREN	Ethernet Enable

Table continues on the next page...

ENET_ECR field descriptions (continued)

Field	Description
	Enables/disables the Ethernet MAC. When the MAC is disabled, the buffer descriptors for an aborted transmit frame are not updated. The uDMA, buffer descriptor, and FIFO control logic are reset, including the buffer descriptor and FIFO pointers. Hardware clears this field under the following conditions: • RESET is set by software • An error condition causes the EBERR field to set. **NOTE:** • ETHEREN must be set at the very last step during ENET configuration/setup/initialization, only *after* all other ENET-related registers have been configured. • If ETHEREN is cleared to 0 by software then next time ETHEREN is set, the EIR interrupts must cleared to 0 due to previous pending interrupts. 0 Reception immediately stops and transmission stops after a bad CRC is appended to any currently transmitted frame. 1 MAC is enabled, and reception and transmission are possible.
0 RESET	Ethernet MAC Reset When this field is set, it clears the ETHEREN field.

11.5.6.6 MII Management Frame Register (ENET_MMFR)

Writing to MMFR triggers a management frame transaction to the PHY device unless MSCR is programmed to zero.

If MSCR is changed from zero to non-zero during a write to MMFR, an MII frame is generated with the data previously written to the MMFR. This allows MMFR and MSCR to be programmed in either order if MSCR is currently zero.

If the MMFR register is written while frame generation is in progress, the frame contents are altered. Software must use the EIR[MII] interrupt indication to avoid writing to the MMFR register while frame generation is in progress.

Address: 30BE_0000h base + 40h offset = 30BE_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ST		OP		PA					RA					TA		DATA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_MMFR field descriptions

Field	Description
31–30 ST	Start Of Frame Delimiter See Table 11-84 (Clause 22) or Table 11-86 (Clause 45) for correct value.
29–28 OP	Operation Code

Table continues on the next page...

ENET_MMFR field descriptions (continued)

Field	Description
	See Table 11-84 (Clause 22) or Table 11-86 (Clause 45) for correct value.
27–23 PA	PHY Address See Table 11-84 (Clause 22) or Table 11-86 (Clause 45) for correct value.
22–18 RA	Register Address See Table 11-84 (Clause 22) or Table 11-86 (Clause 45) for correct value.
17–16 TA	Turn Around This field must be programmed to 10 to generate a valid MII management frame.
DATA	Management Frame Data This is the field for data to be written to or read from the PHY register.

11.5.6.7 MII Speed Control Register (ENET_MSCR)

MSCR provides control of the MII clock (MDC pin) frequency and allows a preamble drop on the MII management frame.

The MII_SPEED field must be programmed with a value to provide an MDC frequency of less than or equal to 2.5 MHz to be compliant with the IEEE 802.3 MII specification. The MII_SPEED must be set to a non-zero value to source a read or write management frame. After the management frame is complete, the MSCR register may optionally be cleared to turn off MDC. The MDC signal generated has a 50% duty cycle except when MII_SPEED changes during operation. This change takes effect following a rising or falling edge of MDC.

For example, if the internal module clock (that is, peripheral bus clock) is 25 MHz, programming MII_SPEED to 0x4 results in an MDC as given in the following equation:

MII clock frequency = 25 MHz / ((4 + 1) x 2) = 2.5 MHz

The following table shows the optimum values for MII_SPEED as a function of IPS bus clock frequency.

Table 11-90. Programming Examples for MSCR

Internal module clock frequency	MSCR [MII_SPEED]	MDC frequency
25 MHz	0x4	2.50 MHz
33 MHz	0x6	2.36 MHz
40 MHz	0x7	2.50 MHz
50 MHz	0x9	2.50 MHz
66 MHz	0xD	2.36 MHz

Address: 30BE_0000h base + 44h offset = 30BE_0044h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0					HOLDTIME			DIS_ PRE	MII_SPEED						0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

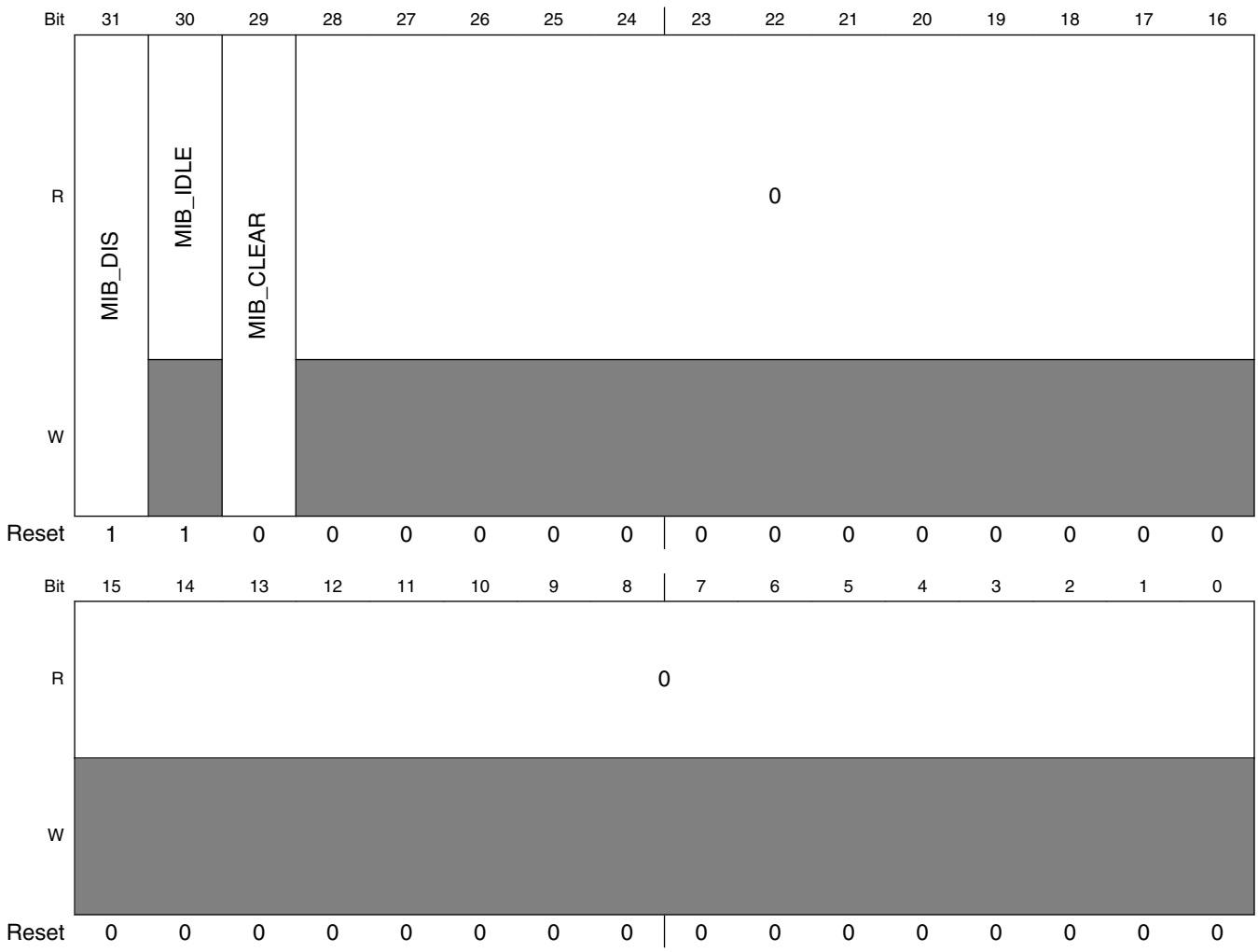
ENET_MSCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 HOLDTIME	Hold time On MDIO Output IEEE802.3 clause 22 defines a minimum of 10 ns for the hold time on the MDIO output. Depending on the host bus frequency, the setting may need to be increased. 000 1 internal module clock cycle 001 2 internal module clock cycles 010 3 internal module clock cycles 111 8 internal module clock cycles
7 DIS_PRE	Disable Preamble Enables/disables prepending a preamble to the MII management frame. The MII standard allows the preamble to be dropped if the attached PHY devices do not require it. 0 Preamble enabled. 1 Preamble (32 ones) is not prepended to the MII management frame.
6–1 MII_SPEED	MII Speed Controls the frequency of the MII management interface clock (MDC) relative to the internal module clock. A value of 0 in this field turns off MDC and leaves it in low voltage state. Any non-zero value results in the MDC frequency of: 1/((MII_SPEED + 1) x 2) of the internal module clock frequency
0 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.8 MIB Control Register (ENET_MIBC)

MIBC is a read/write register controlling and observing the state of the MIB block. Access this register to disable the MIB block operation or clear the MIB counters. The MIB_DIS field resets to 1.

Address: 30BE_0000h base + 64h offset = 30BE_0064h



ENET_MIBC field descriptions

Field	Description
31 MIB_DIS	Disable MIB Logic If this control field is set, 0 MIB logic is enabled. 1 MIB logic is disabled. The MIB logic halts and does not update any MIB counters.

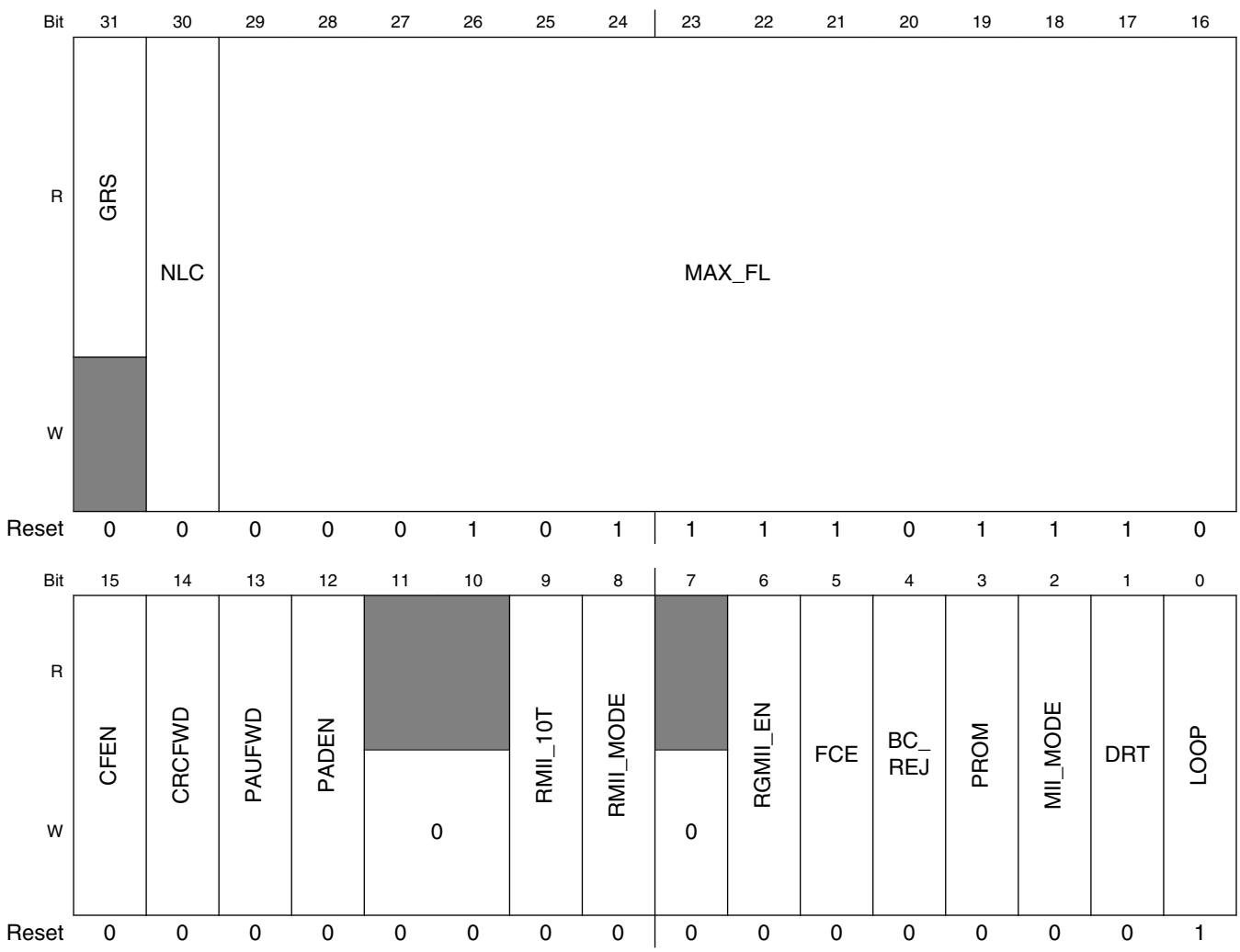
Table continues on the next page...

ENET_MIBC field descriptions (continued)

Field	Description
30 MIB_IDLE	MIB Idle 0 The MIB block is updating MIB counters. 1 The MIB block is not currently updating any MIB counters.
29 MIB_CLEAR	MIB Clear **NOTE:** This field is not self-clearing. To clear the MIB counters set and then clear this field. 0 See note above. 1 All statistics counters are reset to 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.9 Receive Control Register (ENET_RCR)

Address: 30BE_0000h base + 84h offset = 30BE_0084h



ENET_RCR field descriptions

Field	Description
31 GRS	Graceful Receive Stopped Read-only status indicating that the MAC receive datapath is stopped.
30 NLC	Payload Length Check Disable Enables/disables a payload length check. 0 The payload length check is disabled. 1 The core checks the frame's payload length with the frame length/type field. Errors are indicated in the EIR[PLR] field.
29–16 MAX_FL	Maximum Frame Length Resets to decimal 1518. Length is measured starting at DA and includes the CRC at the end of the frame. Transmit frames longer than MAX_FL cause the BABT interrupt to occur. Receive frames longer than MAX_FL cause the BABR interrupt to occur and set the LG field in the end of frame receive buffer descriptor. The recommended default value to be programmed is 1518 or 1522 if VLAN tags are supported.
15 CFEN	MAC Control Frame Enable Enables/disables the MAC control frame. 0 MAC control frames with any opcode other than 0x0001 (pause frame) are accepted and forwarded to the client interface. 1 MAC control frames with any opcode other than 0x0001 (pause frame) are silently discarded.
14 CRCFWD	Terminate/Forward Received CRC Specifies whether the CRC field of received frames is transmitted or stripped. **NOTE:** If padding function is enabled (PADEN = 1), CRCFWD is ignored and the CRC field is checked and always terminated and removed. 0 The CRC field of received frames is transmitted to the user application. 1 The CRC field is stripped from the frame.
13 PAUFWD	Terminate/Forward Pause Frames Specifies whether pause frames are terminated or forwarded. 0 Pause frames are terminated and discarded in the MAC. 1 Pause frames are forwarded to the user application.
12 PADEN	Enable Frame Padding Remove On Receive Specifies whether the MAC removes padding from received frames. 0 No padding is removed on receive by the MAC. 1 Padding is removed from received frames.
11–10 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
9 RMII_10T	Enables 10-Mbit/s mode of the RMII or RGMII . 0 100-Mbit/s operation. 1 10-Mbit/s operation.

Table continues on the next page...

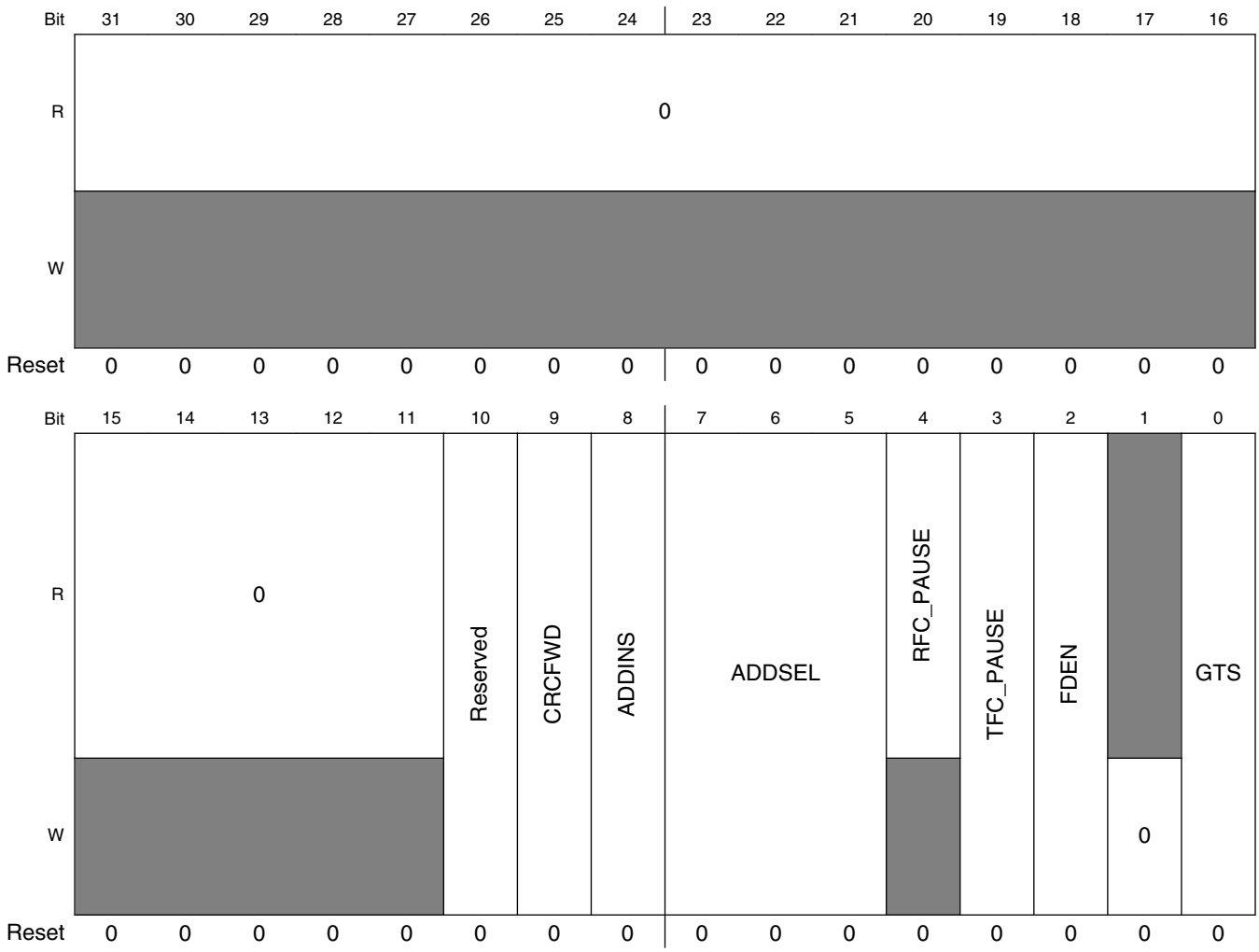
ENET_RCR field descriptions (continued)

Field	Description
8 RMII_MODE	RMII Mode Enable Specifies whether the MAC is configured for MII mode or RMII operation , when ECR[SPEED] is cleared . **NOTE:** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE]. 0 MAC configured for MII mode. 1 MAC configured for RMII operation.
7 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
6 RGMII_EN	RGMII Mode Enable **NOTE:** Do not set both RCR[RGMII_EN] and RCR[RMII_MODE]. 0 MAC configured for non-RGMII operation 1 MAC configured for RGMII operation. If ECR[SPEED] is set, the MAC is in RGMII 1000-Mbit/s mode. If ECR[SPEED] is cleared, the MAC is in RGMII 10/100-Mbit/s mode.
5 FCE	Flow Control Enable If set, the receiver detects PAUSE frames. Upon PAUSE frame detection, the transmitter stops transmitting data frames for a given duration.
4 BC_REJ	Broadcast Frame Reject If set, frames with destination address (DA) equal to 0xFFFF_FFFF_FFFF are rejected unless the PROM field is set. If BC_REJ and PROM are set, frames with broadcast DA are accepted and the MISS (M) is set in the receive buffer descriptor.
3 PROM	Promiscuous Mode All frames are accepted regardless of address matching. 0 Disabled. 1 Enabled.
2 MII_MODE	Media Independent Interface Mode This field must always be set. 0 Reserved. 1 MII or RMII mode, as indicated by the RMII_MODE field.
1 DRT	Disable Receive On Transmit 0 Receive path operates independently of transmit (i.e., full-duplex mode). Can also be used to monitor transmit activity in half-duplex mode. 1 Disable reception of frames while transmitting. (Normally used for half-duplex mode.)
0 LOOP	Internal Loopback This is an MII internal loopback, therefore MII_MODE must be written to 1 and RMII_MODE must be written to 0. 0 Loopback disabled. 1 Transmitted frames are looped back internal to the device and transmit MII output signals are not asserted. DRT must be cleared.

11.5.6.10 Transmit Control Register (ENET_TCR)

TCR is read/write and configures the transmit block. This register is cleared at system reset. FDEN can only be modified when ECR[ETHEREN] is cleared.

Address: 30BE_0000h base + C4h offset = 30BE_00C4h



ENET_TCR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10 Reserved	This field is reserved. This field is read/write and must be set to 0.
9 CRCFWD	Forward Frame From Application With CRC

Table continues on the next page...

ENET_TCR field descriptions (continued)

Field	Description
	0 TxBD[TC] controls whether the frame has a CRC from the application. 1 The transmitter does not append any CRC to transmitted frames, as it is expecting a frame with CRC from the application.
8 ADDINS	Set MAC Address On Transmit 0 The source MAC address is not modified by the MAC. 1 The MAC overwrites the source MAC address with the programmed MAC address according to ADDSEL.
7–5 ADDSEL	Source MAC Address Select On Transmit If ADDINS is set, indicates the MAC address that overwrites the source MAC address. 000 Node MAC address programmed on PADDR1/2 registers. 100 Reserved. 101 Reserved. 110 Reserved.
4 RFC_PAUSE	Receive Frame Control Pause This status field is set when a full-duplex flow control pause frame is received and the transmitter pauses for the duration defined in this pause frame. This field automatically clears when the pause duration is complete.
3 TFC_PAUSE	Transmit Frame Control Pause Pauses frame transmission. When this field is set, EIR[GRA] is set. With transmission of data frames stopped, the MAC transmits a MAC control PAUSE frame. Next, the MAC clears TFC_PAUSE and resumes transmitting data frames. If the transmitter pauses due to user assertion of GTS or reception of a PAUSE frame, the MAC may continue transmitting a MAC control PAUSE frame. 0 No PAUSE frame transmitted. 1 The MAC stops transmission of data frames after the current transmission is complete.
2 FDEN	Full-Duplex Enable If this field is set, frames transmit independent of carrier sense and collision inputs. Only modify this bit when ECR[ETHEREN] is cleared.
1 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
0 GTS	Graceful Transmit Stop When this field is set, MAC stops transmission after any frame currently transmitted is complete and EIR[GRA] is set. If frame transmission is not currently underway, the GRA interrupt is asserted immediately. After transmission finishes, clear GTS to restart. The next frame in the transmit FIFO is then transmitted. If an early collision occurs during transmission when GTS is set, transmission stops after the collision. The frame is transmitted again after GTS is cleared. There may be old frames in the transmit FIFO that transmit when GTS is reasserted. To avoid this, clear ECR[ETHEREN] following the GRA interrupt.

11.5.6.11 Physical Address Lower Register (ENET_PALR)

PALR contains the lower 32 bits (bytes 0, 1, 2, 3) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 0 through 3 of the six-byte source address field when transmitting PAUSE frames.

Address: 30BE_0000h base + E4h offset = 30BE_00E4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_PALR field descriptions

Field	Description
PADDR1	Pause Address Bytes 0 (bits 31:24), 1 (bits 23:16), 2 (bits 15:8), and 3 (bits 7:0) of the 6-byte individual address are used for exact match and the source address field in PAUSE frames.

11.5.6.12 Physical Address Upper Register (ENET_PAUR)

PAUR contains the upper 16 bits (bytes 4 and 5) of the 48-bit address used in the address recognition process to compare with the destination address (DA) field of receive frames with an individual DA. In addition, this register is used in bytes 4 and 5 of the six-byte source address field when transmitting PAUSE frames. Bits 15:0 of PAUR contain a constant type field (0x8808) for transmission of PAUSE frames.

Address: 30BE_0000h base + E8h offset = 30BE_00E8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PADDR2																TYPE															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0

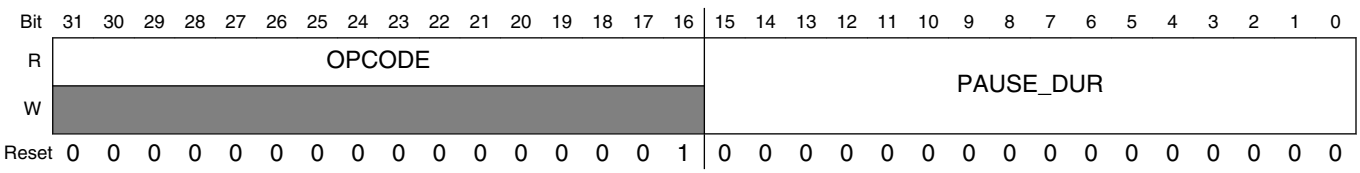
ENET_PAUR field descriptions

Field	Description
31–16 PADDR2	Bytes 4 (bits 31:24) and 5 (bits 23:16) of the 6-byte individual address used for exact match, and the source address field in PAUSE frames.
TYPE	Type Field In PAUSE Frames These fields have a constant value of 0x8808.

11.5.6.13 Opcode/Pause Duration Register (ENET_OPD)

OPD is read/write accessible. This register contains the 16-bit opcode and 16-bit pause duration fields used in transmission of a PAUSE frame. The opcode field is a constant value, 0x0001. When another node detects a PAUSE frame, that node pauses transmission for the duration specified in the pause duration field. The lower 16 bits of this register are not reset and you must initialize it.

Address: 30BE_0000h base + ECh offset = 30BE_00ECh



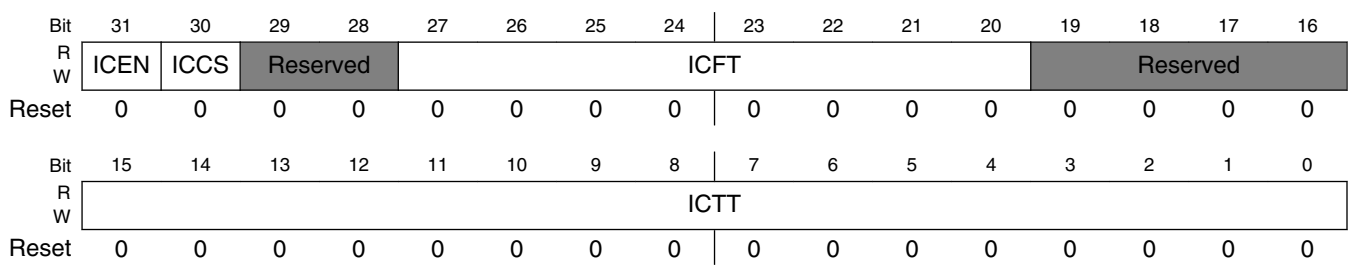
ENET_OPD field descriptions

Field	Description
31–16 OPCODE	Opcode Field In PAUSE Frames These fields have a constant value of 0x0001.
PAUSE_DUR	Pause Duration Pause duration field used in PAUSE frames.

11.5.6.14 Transmit Interrupt Coalescing Register (ENET_TXIC*n*)

See Interrupt coalescence for more information.

Address: 30BE_0000h base + F0h offset + (4d × i), where i=0d to 2d



ENET_TXIC*n* field descriptions

Field	Description
31 ICEN	Interrupt Coalescing Enable

Table continues on the next page...

ENET_TXIC*n* field descriptions (continued)

Field	Description
	0 Disable Interrupt coalescing. 1 Enable Interrupt coalescing.
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 Use MII/GMII TX clocks. 1 Use ENET system clock.
29–28 Reserved	This field must be set to 0. This field is reserved.
27–20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be transmitted for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19–16 Reserved	This field must be set to 0. This field is reserved.
ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after transmitting a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame transmission defined by ICFT and starts again upon transmission of the next first frame. Must be greater than zero to avoid unpredictable behavior.

11.5.6.15 Receive Interrupt Coalescing Register (ENET_RXIC*n*)

See Interrupt coalescence for more information.

Address: 30BE_0000h base + 100h offset + (4d × i), where i=0d to 2d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ICEN	ICCS	Reserved						ICFT				Reserved			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ICTT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RXIC*n* field descriptions

Field	Description
31 ICEN	Interrupt Coalescing Enable 0 Disable Interrupt coalescing. 1 Enable Interrupt coalescing.

Table continues on the next page...

ENET_RXIC*n* field descriptions (continued)

Field	Description
30 ICCS	Interrupt Coalescing Timer Clock Source Select 0 Use MII/GMII TX clocks. 1 Use ENET system clock.
29–28 Reserved	This field must be set to 0. This field is reserved.
27–20 ICFT	Interrupt coalescing frame count threshold This value determines the number of frames needed to be received for raising an interrupt. Frame counter restarts after reaching this threshold value or after the expiring of the coalescing timer. Must be greater than zero to avoid unpredictable behavior.
19–16 Reserved	This field must be set to 0. This field is reserved.
ICTT	Interrupt coalescing timer threshold Interrupt coalescing timer threshold in units of 64 clock periods. This value determines the maximum amount of time after receiving a frame before raising an interrupt. The threshold timer is disabled after expiring or number of frame reception defined by ICFT and starts again upon reception of the next first frame. Must be greater than zero to avoid unpredictable behavior.

11.5.6.16 Descriptor Individual Upper Address Register (ENET_IAUR)

IAUR contains the upper 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the destination address (DA) field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: 30BE_0000h base + 118h offset = 30BE_0118h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IAUR field descriptions

Field	Description
IADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR1 contains hash index bit 63. Bit 0 of IADDR1 contains hash index bit 32.

11.5.6.17 Descriptor Individual Lower Address Register (ENET_IALR)

IALR contains the lower 32 bits of the 64-bit individual address hash table. The address recognition process uses this table to check for a possible match with the DA field of receive frames with an individual DA. This register is not reset and you must initialize it.

Address: 30BE_0000h base + 11Ch offset = 30BE_011Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IALR field descriptions

Field	Description
IADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a unicast address. Bit 31 of IADDR2 contains hash index bit 31. Bit 0 of IADDR2 contains hash index bit 0.

11.5.6.18 Descriptor Group Upper Address Register (ENET_GAUR)

GAUR contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: 30BE_0000h base + 120h offset = 30BE_0120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR1																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_GAUR field descriptions

Field	Description
GADDR1	Contains the upper 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR1 contains hash index bit 63. Bit 0 of GADDR1 contains hash index bit 32.

11.5.6.19 Descriptor Group Lower Address Register (ENET_GALR)

GALR contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. You must initialize this register.

Address: 30BE_0000h base + 124h offset = 30BE_0124h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GADDR2																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_GALR field descriptions

Field	Description
GADDR2	Contains the lower 32 bits of the 64-bit hash table used in the address recognition process for receive frames with a multicast address. Bit 31 of GADDR2 contains hash index bit 31. Bit 0 of GADDR2 contains hash index bit 0.

11.5.6.20 Transmit FIFO Watermark Register (ENET_TFWR)

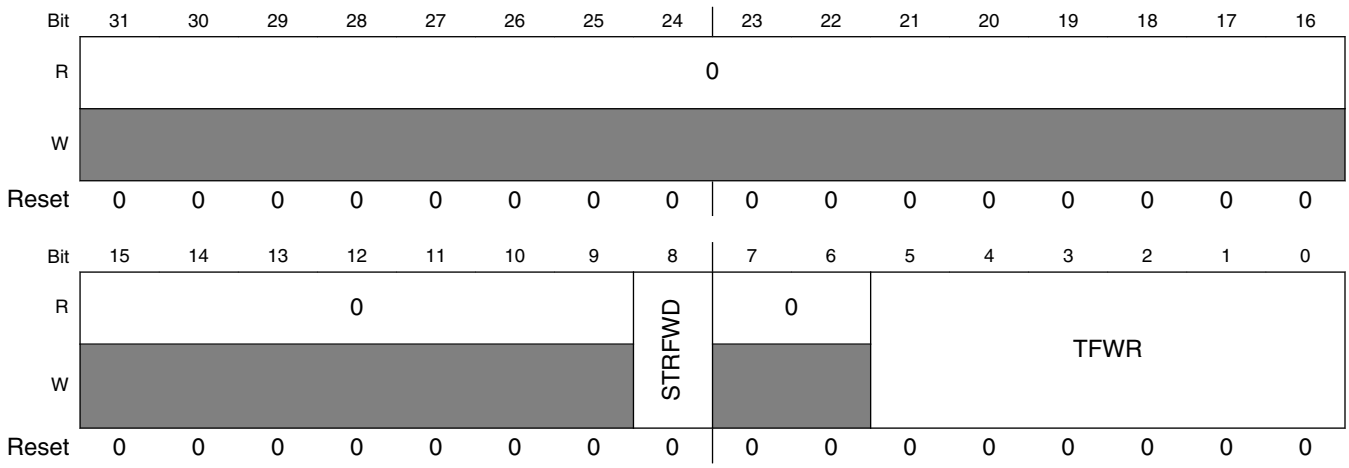
If TFWR[STRFWD] is cleared, TFWR[TFWR] controls the amount of data required in the transmit FIFO before transmission of a frame can begin. This allows you to minimize transmit latency (TFWR = 00 or 01) or allow for larger bus access latency (TFWR = 11) due to contention for the system bus. Setting the watermark to a high value minimizes the risk of transmit FIFO underrun due to contention for the system bus. The byte counts associated with the TFWR field may need to be modified to match a given system requirement, for example, worst-case bus access latency by the transmit data uDMA channel.

When the FIFO level reaches the value the TFWR field and when the STR_FWD is set to ‘0’, the MAC transmit control logic starts frame transmission even before the end-of-frame is available in the FIFO (cut-through operation).

If a complete frame has a size smaller than the threshold programmed with TFWR, the MAC also transmits the Frame to the line.

To enable store and forward on the Transmit path, set STR_FWD to ‘1’. In this case, the MAC starts to transmit data only when a complete frame is stored in the Transmit FIFO.

Address: 30BE_0000h base + 144h offset = 30BE_0144h



ENET_TFWR field descriptions

Field	Description
31–9 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
8 STRFWD	Store And Forward Enable 0 Reset. The transmission start threshold is programmed in TFWR[TFWR]. 1 Enabled.
7–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TFWR	Transmit FIFO Write If TFWR[STRFWD] is cleared, this field indicates the number of bytes, in steps of 64 bytes, written to the transmit FIFO before transmission of a frame begins. **NOTE:** If a frame with less than the threshold is written, it is still sent independently of this threshold setting. The threshold is relevant only if the frame is larger than the threshold given. 000000 64 bytes written. 000001 64 bytes written. 000010 128 bytes written. 000011 192 bytes written. ... ... 111111 4032 bytes written.

11.5.6.21 Receive Descriptor Ring 1 Start Register (ENET_RDSR1)

RDSR1 points to the beginning of circular receive buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

Address: 30BE_0000h base + 160h offset = 30BE_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RDSR1 field descriptions

Field	Description
31–3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue 1.
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.22 Transmit Buffer Descriptor Ring 1 Start Register (ENET_TDSR1)

TDSR1 provides a pointer to the beginning of the circular transmit buffer descriptor queue 1 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

Address: 30BE_0000h base + 164h offset = 30BE_0164h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TDSR1 field descriptions

Field	Description
31–3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 1.

Table continues on the next page...

ENET_TDSR1 field descriptions (continued)

Field	Description
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.23 Maximum Receive Buffer Size Register - Ring 1 (ENET_MRBR1)

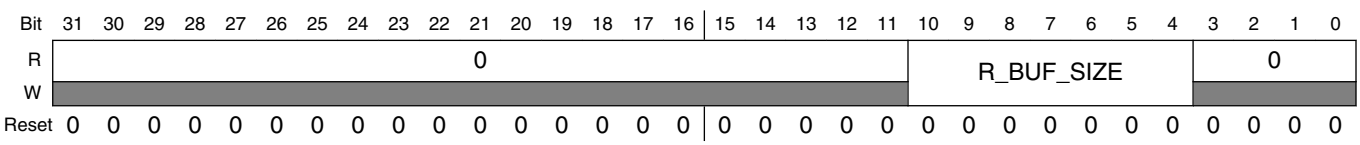
MRBR1 is a user-programmable register that dictates the maximum size of all ring-1 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR1 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR1 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_ZIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR1 greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

Address: 30BE_0000h base + 168h offset = 30BE_0168h



ENET_MRBR1 field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
Reserved	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size. This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.24 Receive Descriptor Ring 2 Start Register (ENET_RDSR2)

RDSR points to the beginning of circular receive buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

Address: 30BE_0000h base + 16Ch offset = 30BE_016Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	R_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	R_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RDSR2 field descriptions

Field	Description
31–3 R_DES_START	Pointer to the beginning of receive buffer descriptor queue 2.
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.25 Transmit Buffer Descriptor Ring 2 Start Register (ENET_TDSR2)

TDSR2 provides a pointer to the beginning of circular transmit buffer descriptor queue 2 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

Address: 30BE_0000h base + 170h offset = 30BE_0170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TDSR2 field descriptions

Field	Description
31–3 X_DES_START	Pointer to the beginning of transmit buffer descriptor queue 2.
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.26 Maximum Receive Buffer Size Register - Ring 2 (ENET_MRBR2)

MRBR2 is a user-programmable register that dictates the maximum size of all ring-2 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

- To allow one maximum size frame per buffer, MRBR2 must be set to RCR[MAX_FL] or larger.
- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To properly align the buffer, MRBR2 must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR2 greater than or equal to 256 bytes.

NOTE

This register must be initialized prior to operation

Address: 30BE_0000h base + 174h offset = 30BE_0174h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																R_BUF_SIZE												0			
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_MRBR2 field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
Reserved	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size. This field is reserved. This read-only field is reserved and always has the value 0.

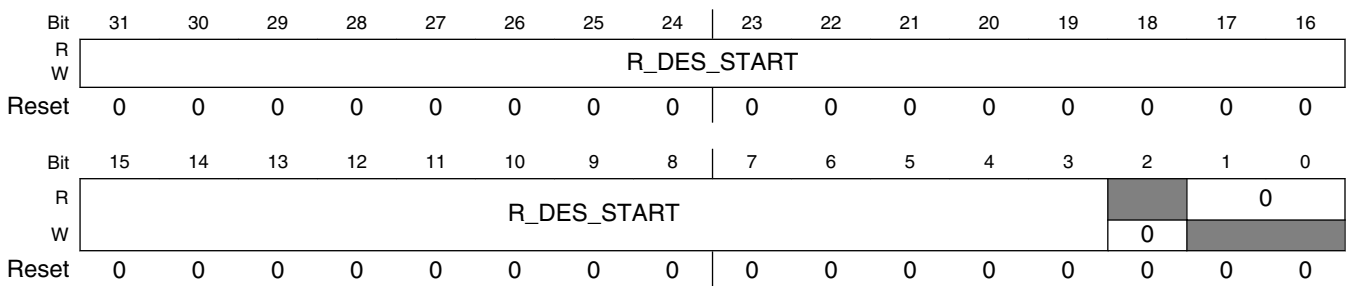
11.5.6.27 Receive Descriptor Ring 0 Start Register (ENET_RDSR)

RDSR points to the beginning of the circular receive buffer descriptor queue in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation

Address: 30BE_0000h base + 180h offset = 30BE_0180h



ENET_RDSR field descriptions

Field	Description
31–3 R_DES_START	Pointer to the beginning of the receive buffer descriptor queue. 0
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.28 Transmit Buffer Descriptor Ring 0 Start Register (ENET_TDSR)

TDSR provides a pointer to the beginning of the circular transmit buffer descriptor queue 0 in external memory. This pointer must be 64-bit aligned (bits 2–0 must be zero); however, for optimal performance the pointer should be 512-bit aligned, that is, evenly divisible by 64.

NOTE

This register must be initialized prior to operation.

Address: 30BE_0000h base + 184h offset = 30BE_0184h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	X_DES_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	X_DES_START														0	
W														0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TDSR field descriptions

Field	Description
31–3 X_DES_START	Pointer to the beginning of the transmit buffer descriptor queue.
2 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.29 Maximum Receive Buffer Size Register - Ring 0 (ENET_MRBR)

The MRBR is a user-programmable register that dictates the maximum size of all ring-0 receive buffers. This value should take into consideration that the receive CRC is always written into the last receive buffer.

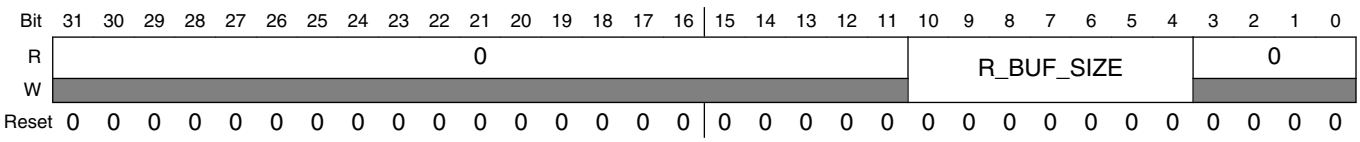
- R_BUF_SIZE is concatentated with the four least-significant bits of this register and are used as the maximum receive buffer size.
- To allow one maximum size frame per buffer, MRBR must be set to RCR[MAX_FL] or larger.

- To properly align the buffer, MRBR must be evenly divisible by 64. To ensure this, set the lower two bits of R_BUF_SIZE to zero. The lower four bits of this register are already set to zero by the device.
- To minimize bus usage (descriptor fetches), set MRBR greater than or equal to 256 bytes.

NOTE

This register must be initialized before operation.

Address: 30BE_0000h base + 188h offset = 30BE_0188h

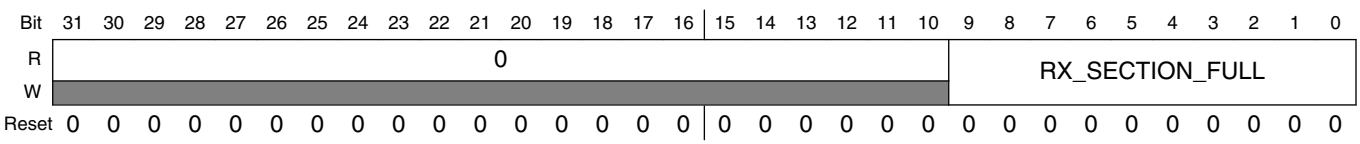


ENET_MRBR field descriptions

Field	Description
31–11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–4 R_BUF_SIZE	Receive buffer size in bytes. This value, concatenated with the four least-significant bits of this register (which are always zero), is the effective maximum receive buffer size.
Reserved	This field, which is always zero, is the four least-significant bits of the maximum receive buffer size. This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.30 Receive FIFO Section Full Threshold (ENET_RSFL)

Address: 30BE_0000h base + 190h offset = 30BE_0190h

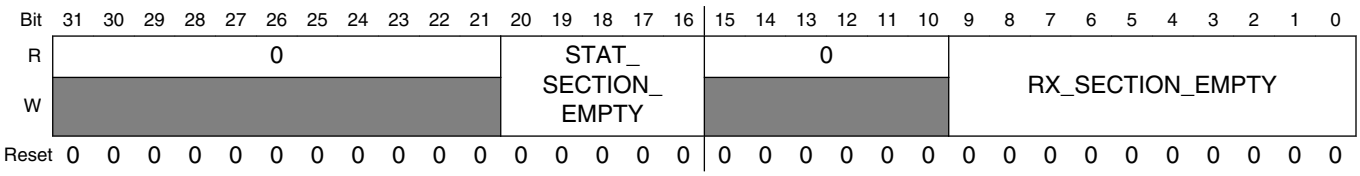


ENET_RSFL field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_SECTION_ FULL	Value Of Receive FIFO Section Full Threshold Value, in 64-bit words, of the receive FIFO section full threshold. Clear this field to enable store and forward on the RX FIFO. When programming a value greater than 0 (cut-through operation), it must be greater than RAEM[RX_ALMOST_EMPTY]. When the FIFO level reaches the value in this field, data is available in the Receive FIFO (cut-through operation).

11.5.6.31 Receive FIFO Section Empty Threshold (ENET_RSEM)

Address: 30BE_0000h base + 194h offset = 30BE_0194h

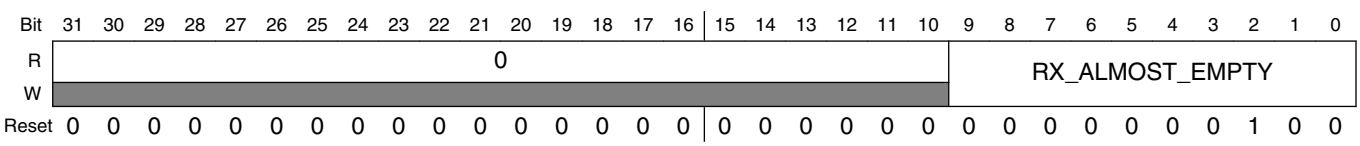


ENET_RSEM field descriptions

Field	Description
31–21 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
20–16 STAT_ SECTION_ EMPTY	RX Status FIFO Section Empty Threshold Defines number of frames in the receive FIFO, independent of its size, that can be accepted. If the limit is reached, reception will continue normally, however a pause frame will be triggered to indicate a possible congestion to the remote device to avoid FIFO overflow. A value of 0 disables automatic pause frame generation
15–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_SECTION_ EMPTY	Value Of The Receive FIFO Section Empty Threshold Value, in 64-bit words, of the receive FIFO section empty threshold. When the FIFO has reached this level, a pause frame will be issued. A value of 0 disables automatic pause frame generation. When the FIFO level goes below the value programmed in this field, an XON pause frame is issued to indicate the FIFO congestion is cleared to the remote Ethernet client. **NOTE:** The section-empty threshold indications from both FIFOs are OR'ed to cause XOFF pause frame generation.

11.5.6.32 Receive FIFO Almost Empty Threshold (ENET_RAEM)

Address: 30BE_0000h base + 198h offset = 30BE_0198h



ENET_RAEM field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

Table continues on the next page...

ENET_RAEM field descriptions (continued)

Field	Description
RX_ALMOST_EMPTY	Value Of The Receive FIFO Almost Empty Threshold Value, in 64-bit words, of the receive FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field and the end-of-frame has not been received for the frame yet, the core receive read control stops FIFO read (and subsequently stops transferring data to the MAC client application). It continues to deliver the frame, if again more data than the threshold or the end-of-frame is available in the FIFO. A minimum value of 4 should be set.

11.5.6.33 Receive FIFO Almost Full Threshold (ENET_RAFL)

Address: 30BE_0000h base + 19Ch offset = 30BE_019Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						RX_ALMOST_FULL									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

ENET_RAFL field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
RX_ALMOST_FULL	Value Of The Receive FIFO Almost Full Threshold Value, in 64-bit words, of the receive FIFO almost full threshold. When the FIFO level comes close to the maximum, so that there is no more space for at least RX_ALMOST_FULL number of words, the MAC stops writing data in the FIFO and truncates the received frame to avoid FIFO overflow. The corresponding error status will be set when the frame is delivered to the application. A minimum value of 4 should be set.

11.5.6.34 Transmit FIFO Section Empty Threshold (ENET_TSEM)

Address: 30BE_0000h base + 1A0h offset = 30BE_01A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																						TX_SECTION_EMPTY									
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TSEM field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_SECTION_EMPTY	Value Of The Transmit FIFO Section Empty Threshold

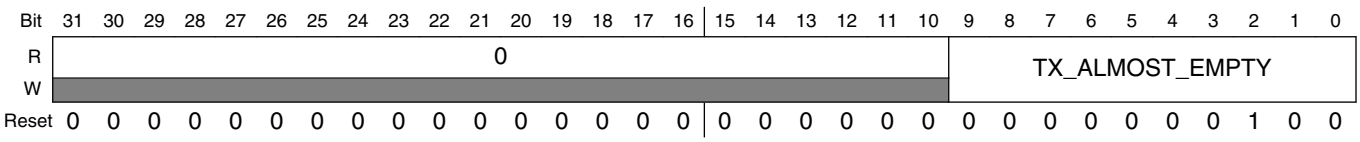
Table continues on the next page...

ENET_TSEM field descriptions (continued)

Field	Description
	Value, in 64-bit words, of the transmit FIFO section empty threshold. See Transmit FIFO for more information.

11.5.6.35 Transmit FIFO Almost Empty Threshold (ENET_TAEM)

Address: 30BE_0000h base + 1A4h offset = 30BE_01A4h

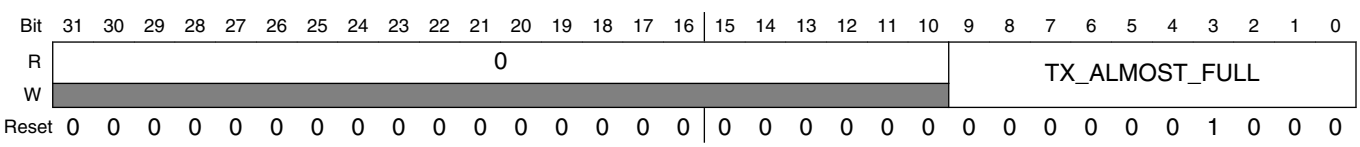


ENET_TAEM field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_ALMOST_EMPTY	Value of Transmit FIFO Almost Empty Threshold Value, in 64-bit words, of the transmit FIFO almost empty threshold. When the FIFO level reaches the value programmed in this field, and no end-of-frame is available for the frame, the MAC transmit logic, to avoid FIFO underflow, stops reading the FIFO and transmits a frame with an MII error indication. See Transmit FIFO for more information. A minimum value of 4 should be set.

11.5.6.36 Transmit FIFO Almost Full Threshold (ENET_TAFL)

Address: 30BE_0000h base + 1A8h offset = 30BE_01A8h



ENET_TAFL field descriptions

Field	Description
31–10 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TX_ALMOST_FULL	Value Of The Transmit FIFO Almost Full Threshold Value, in 64-bit words, of the transmit FIFO almost full threshold. A minimum value of six is required . A recommended value of at least 8 should be set allowing a latency of two clock cycles to the application. If more latency is required the value can be increased as necessary (latency = TAFL - 5).

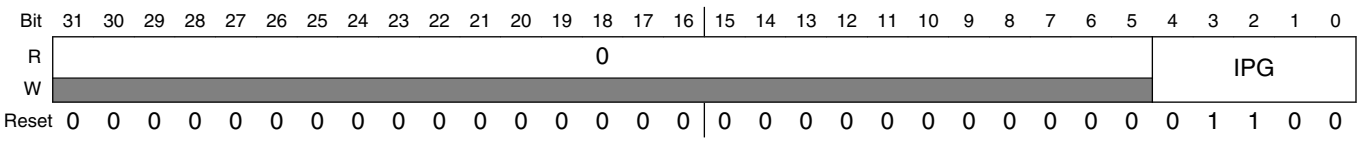
Table continues on the next page...

ENET_TAFL field descriptions (continued)

Field	Description
	When the FIFO level comes close to the maximum, so that there is no more space for at least TX_ALMOST_FULL number of words, the pin ff_tx_rdy is deasserted. If the application does not react on this signal, the FIFO write control logic, to avoid FIFO overflow, truncates the current frame and sets the error status. As a result, the frame will be transmitted with an GMII/MII error indication. See Transmit FIFO for more information. **NOTE:** A FIFO overflow is a fatal error and requires a global reset on the transmit datapath or at least deassertion of ETHEREN.

11.5.6.37 Transmit Inter-Packet Gap (ENET_TIPG)

Address: 30BE_0000h base + 1ACh offset = 30BE_01ACh

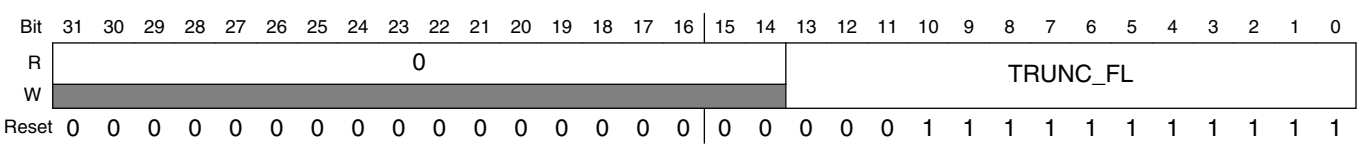


ENET_TIPG field descriptions

Field	Description
31–5 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
IPG	Transmit Inter-Packet Gap Indicates the IPG, in bytes, between transmitted frames. Valid values range from 8 to 26. If the written value is less than 8 or greater than 26, the internal (effective) IPG is 12. **NOTE:** The IPG value read will be the value that was written, even if it is out of range.

11.5.6.38 Frame Truncation Length (ENET_FTRL)

Address: 30BE_0000h base + 1B0h offset = 30BE_01B0h



ENET_FTRL field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TRUNC_FL	Frame Truncation Length

Table continues on the next page...

ENET_FTRL field descriptions (continued)

Field	Description
	Indicates the value a receive frame is truncated, if it is greater than this value. Must be greater than or equal to RCR[MAX_FL]. **NOTE:** Truncation happens at TRUNC_FL. However, when truncation occurs, the application (FIFO) may receive less data, guaranteeing that it never receives more than the set limit.

11.5.6.39 Transmit Accelerator Function Configuration (ENET_TACC)

TACC controls accelerator actions when sending frames. The register can be changed before or after each frame, but it must remain unmodified during frame writes into the transmit FIFO.

The TFWR[STRFWD] field must be set to use the checksum feature.

Address: 30BE_0000h base + 1C0h offset = 30BE_01C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W								0								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R												PROCHK	IPCHK			SHIFT16
W						0								0		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_TACC field descriptions

Field	Description
31–5 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
4 PROCHK	Enables insertion of protocol checksum. 0 Checksum not inserted. 1 If an IP frame with a known protocol is transmitted, the checksum is inserted automatically into the frame. The checksum field must be cleared. The other frames are not modified.
3 IPCHK	Enables insertion of IP header checksum. 0 Checksum is not inserted. 1 If an IP frame is transmitted, the checksum is inserted automatically. The IP header checksum field must be cleared. If a non-IP frame is transmitted the frame is not modified.

Table continues on the next page...

ENET_TACC field descriptions (continued)

Field	Description
2–1 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
0 SHIFT16	TX FIFO Shift-16 0 Disabled. 1 Indicates to the transmit data FIFO that the written frames contain two additional octets before the frame data. This means the actual frame begins at bit 16 of the first word written into the FIFO. This function allows putting the frame payload on a 32-bit boundary in memory, as the 14-byte Ethernet header is extended to a 16-byte header.

11.5.6.40 Receive Accelerator Function Configuration (ENET_RACC)

Address: 30BE_0000h base + 1C4h offset = 30BE_01C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R																
W									0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R									SHIFT16	LINEDIS				PRODIS	IPDIS	PADREM
W				0							0					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RACC field descriptions

Field	Description
31–8 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
7 SHIFT16	RX FIFO Shift-16 When this field is set, the actual frame data starts at bit 16 of the first word read from the RX FIFO aligning the Ethernet payload on a 32-bit boundary. **NOTE:** This function only affects the FIFO storage and has no influence on the statistics, which use the actual length of the frame received. 0 Disabled. 1 Instructs the MAC to write two additional bytes in front of each frame received into the RX FIFO.
6 LINEDIS	Enable Discard Of Frames With MAC Layer Errors 0 Frames with errors are not discarded. 1 Any frame received with a CRC, length, or PHY error is automatically discarded and not forwarded to the user application interface.

Table continues on the next page...

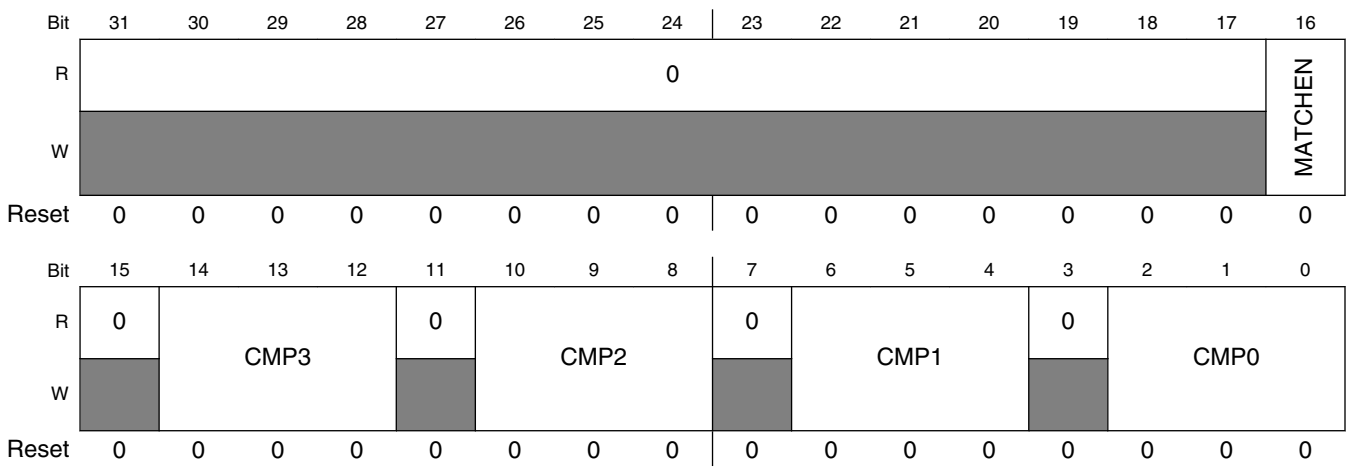
ENET_RACC field descriptions (continued)

Field	Description
5–3 Reserved	This field is reserved. This write-only field is reserved. It must always be written with the value 0.
2 PRODIS	Enable Discard Of Frames With Wrong Protocol Checksum 0 Frames with wrong checksum are not discarded. 1 If a TCP/IP, UDP/IP, or ICMP/IP frame is received that has a wrong TCP, UDP, or ICMP checksum, the frame is discarded. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
1 IPDIS	Enable Discard Of Frames With Wrong IPv4 Header Checksum 0 Frames with wrong IPv4 header checksum are not discarded. 1 If an IPv4 frame is received with a mismatching header checksum, the frame is discarded. IPv6 has no header checksum and is not affected by this setting. Discarding is only available when the RX FIFO operates in store and forward mode (RSFL cleared).
0 PADREM	Enable Padding Removal For Short IP Frames 0 Padding not removed. 1 Any bytes following the IP payload section of the frame are removed from the frame.

11.5.6.41 Receive Classification Match Register for Class n (ENET_RCMR*n*)

This match register allows specifying up to four priorities, which are tested (OR'ed) simultaneously. The match detection uses the extracted VLAN field according to the rules for VLAN detection configured through the ECR register. If both match registers, RCMR1 and RCMR2, report a match at the same time, only the class 1 match is indicated as the final result.

Address: 30BE_0000h base + 1C8h offset + (4d × i), where i=0d to 1d



ENET_RCMR*n* field descriptions

Field	Description
31–17 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
16 MATCHEN	Match Enable **NOTE:** A comparison is done only on incoming VLAN frames. If no VLAN frame is received no match will occur. If both match registers have overlapping compare values and hence can match both on the same frame, only class 1 will be indicated and the class 2 match is ignored. 0 Disabled (default): no compares will occur and the classification indicator for this class will never assert. 1 The register contents are valid and a comparison with all compare values is done when a VLAN frame is received.
15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–12 CMP3	Compare 3 Fourth value to compare against. If unused it must be set to the same value as CMP0.
11 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
10–8 CMP2	Compare 2 Third value to compare against. If unused it must be set to the same value as CMP0.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6–4 CMP1	Compare 1 Second value to compare against. If unused it must be set to the same value as CMP0.
3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
CMP0	Compare 0 A three-bit value that will be compared with the frame's VLAN priority field (if a VLAN frame is received). All four compare values, CMP0..3, will be used in parallel. If any of the values match, a match for the class is reported (if MATCHEN is 1). **NOTE:** To implement a single priority match, all four compare values must be set to the same value.

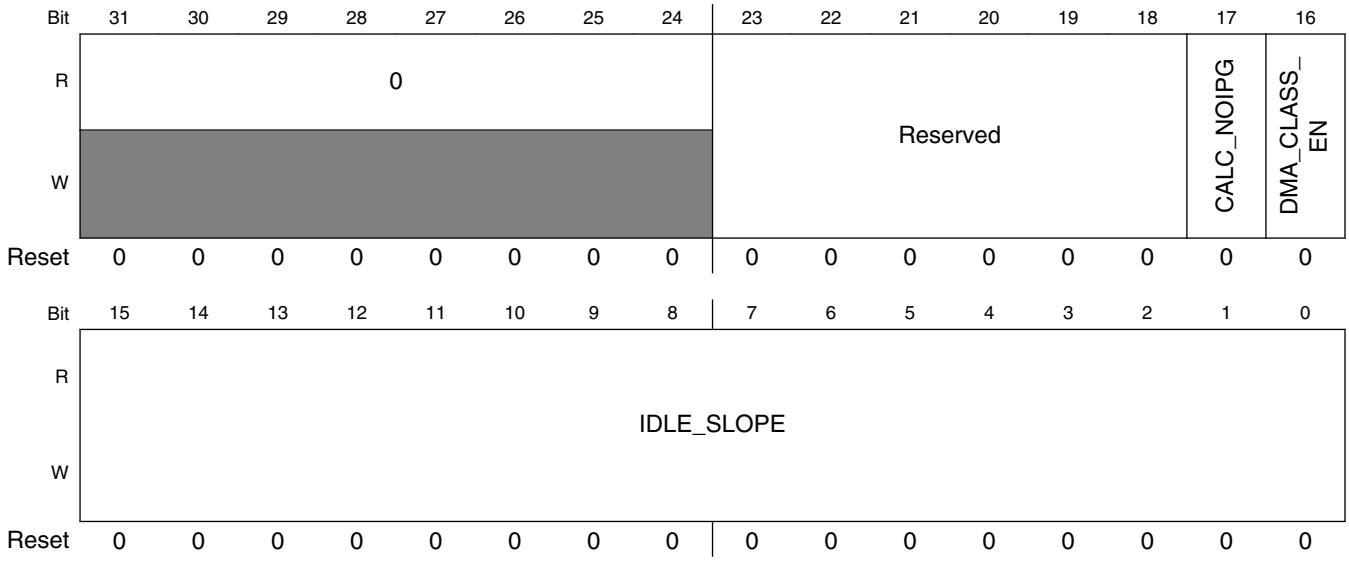
11.5.6.42 DMA Class Based Configuration (ENET_DMA*n*CFG)

The DMA class based configuration registers are used to configure the DMA controller interface to support the additional class 1 (buffer descriptor ring 1) and class 2 (buffer descriptor ring 2) traffic and define configuration options such as bandwidth allocation as needed.

NOTE

The registers are cleared when ECR[ETHEREN] becomes 0.

Address: 30BE_0000h base + 1D8h offset + (4d × i), where i=0d to 1d



ENET_DMA*n*CFG field descriptions

Field	Description
31–24 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
23–18 Reserved	This field is reserved. **NOTE:** Write only zeroes to this field.
17 CALC_NOIPG	Calculate no IPG Disable inclusion of IPG bytes for bandwidth calculations. 0 The traffic shaper function should consider 12 octets of IPG in addition to the frame data transferred for a frame when doing bandwidth calculations. This is the default. 1 Addition of 12 bytes for the IPG should be omitted when calculating the bandwidth (for traffic shaping, when writing a frame into the transmit FIFO, the shaper will usually consider 12 bytes of IPG for every frame as part of the bandwidth allocated by the frame. This addition can be suppressed, meaning short frames will become more bandwidth than large frames due to the relation of data to IPG overhead).
16 DMA_CLASS_EN	DMA class enable 0 The DMA controller's channel for the class is not used. **NOTE:** Disabling the DMA controller of a class also requires disabling the class match comparator for the class (see registers RCMRn). When class 1 and class 2 queues are disabled then their frames will be placed in queue 0. 1 Enable the DMA controller to support the corresponding descriptor ring for this class of traffic.
IDLE_SLOPE	Idle slope 16-bit value to define the per class idle slope setting used by the credit based shaper defining allocated bandwidth for the class. This value is used to calculate the BW (bandwidth) fraction, given by the equation, BW fraction = 1/(1+512/IDLE_SLOPE). Idle slope is restricted to certain values. For values less than 128, idle slope = 2^n, where n = 0, 1, 2, ...6. For values equal to or greater than 128, idle slope = 128×m, where m = 1, 2, 3, ...12.

Table continues on the next page...

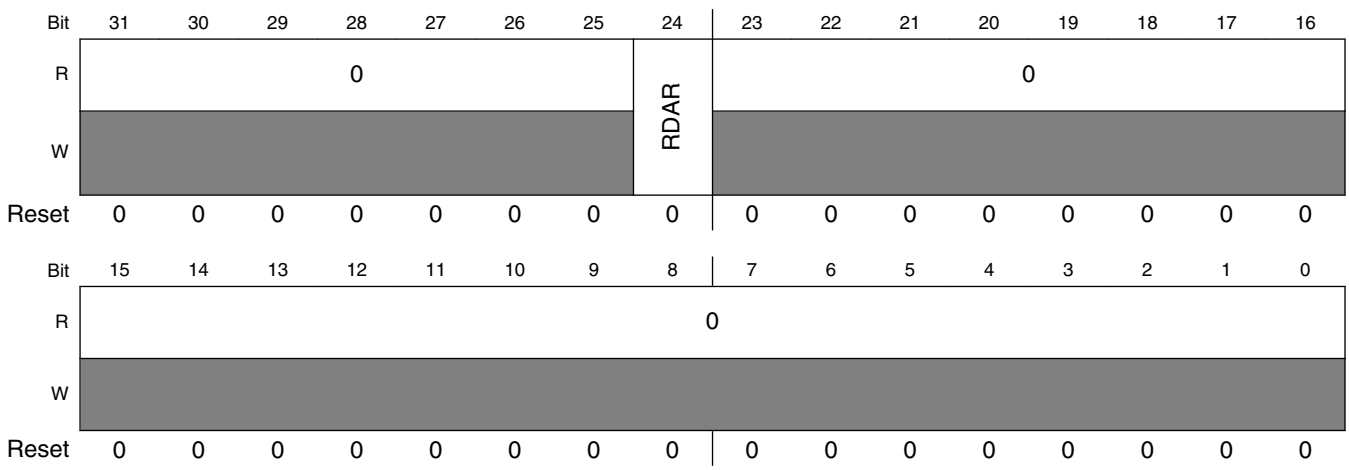
ENET_DMA*n*CFG field descriptions (continued)

Field	Description
	Example 1. BW fraction = 0.20 = 1/(1+(512/128)); therefore idleslope = 128. Example 2. BW fraction = 0.33 = 1/(1+(512/256)); therefore idleslope = 256. Example 3. BW fraction = 0.75 = 1/(1+(512/1536)); therefore idleslope = 1536. **NOTE:** For AVB applications, the BW fraction of class 1 and class 2 combined must not exceed .75.

11.5.6.43 Receive Descriptor Active Register - Ring 1 (ENET_RDAR1)

RDAR1 is a command register, written by the user, to indicate that the receive descriptor ring 1 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Address: 30BE_0000h base + 1E0h offset = 30BE_01E0h



ENET_RDAR1 field descriptions

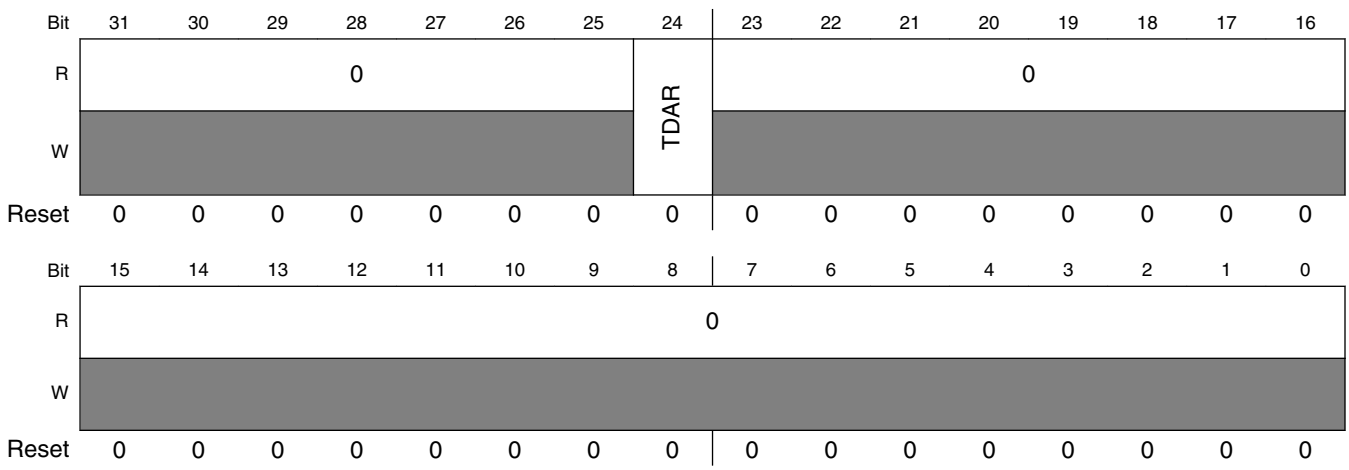
Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.44 Transmit Descriptor Active Register - Ring 1 (ENET_TDAR1)

TDAR1 is a command register that the user writes to indicate that transmit descriptor ring 1 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

The TDAR register is cleared at reset, when ECR[ETHEREN] transitions from set to cleared, or when ECR[RESET] is set.

Address: 30BE_0000h base + 1E4h offset = 30BE_01E4h



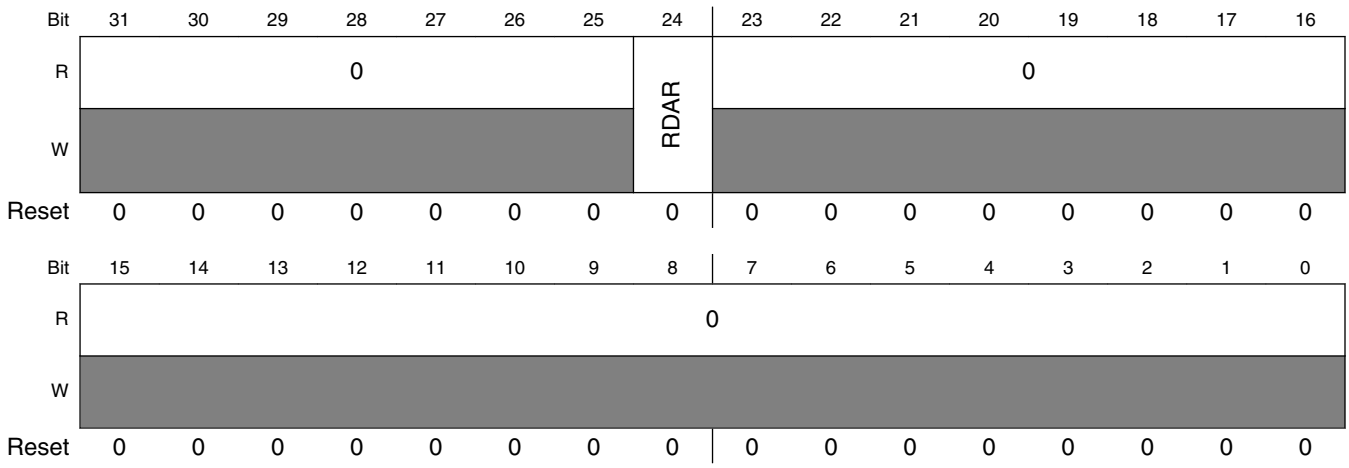
ENET_TDAR1 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.45 Receive Descriptor Active Register - Ring 2 (ENET_RDAR2)

RDAR2 is a command register, written by the user, to indicate that the receive descriptor ring 2 has been updated, that is, that the driver produced empty receive buffers with the empty bit set.

Address: 30BE_0000h base + 1E8h offset = 30BE_01E8h



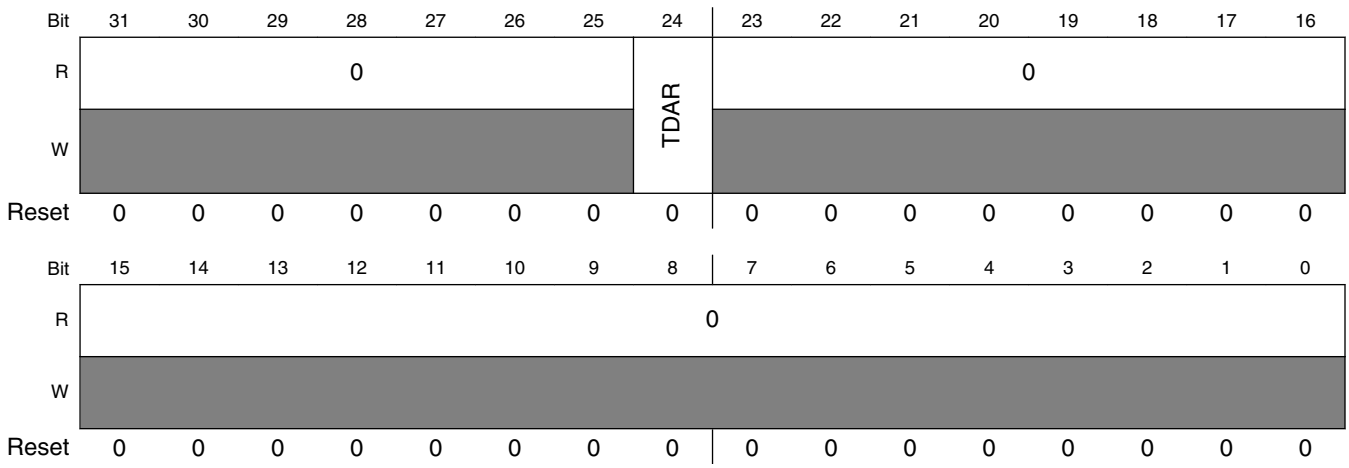
ENET_RDAR2 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 RDAR	Receive Descriptor Active Always set to 1 when this register is written, regardless of the value written. This field is cleared by the MAC device when no additional empty descriptors remain in the receive ring. It is also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.46 Transmit Descriptor Active Register - Ring 2 (ENET_TDAR2)

TDAR2 is a command register that the user writes to indicate that transmit descriptor ring 2 has been updated, that is, that transmit buffers have been produced by the driver with the ready bit set in the buffer descriptor.

Address: 30BE_0000h base + 1ECh offset = 30BE_01ECh



ENET_TDAR2 field descriptions

Field	Description
31–25 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
24 TDAR	Transmit Descriptor Active Always set to 1 when this register is written, regardless of the value written. This bit is cleared by the MAC device when no additional ready descriptors remain in the transmit ring. Also cleared when ECR[ETHEREN] transitions from set to cleared or when ECR[RESET] is set.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

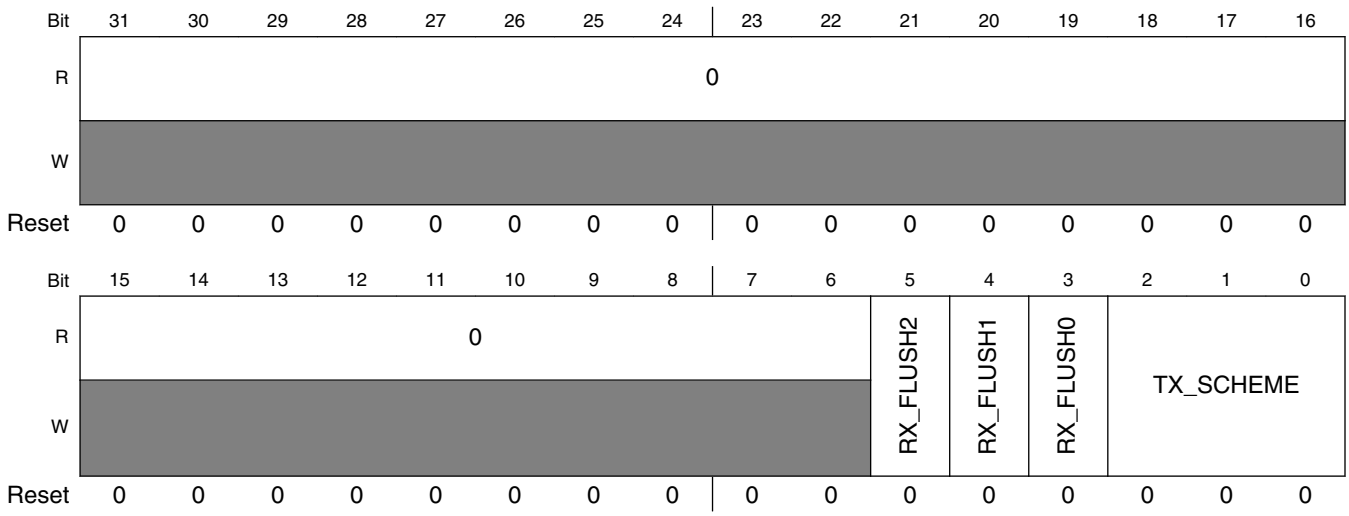
11.5.6.47 QOS Scheme (ENET_QOS)

This register sets the QOS scheme.

NOTE

When both class 1 and class 2 are disabled, RX flushing for these rings must also be disabled.

Address: 30BE_0000h base + 1F0h offset = 30BE_01F0h



ENET_QOS field descriptions

Field	Description
31–6 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
5 RX_FLUSH2	RX Flush Ring 2 Enable or disable RX Flush for ring 2. See Receive flushing. 0 Disable 1 Enable
4 RX_FLUSH1	RX Flush Ring 1 Enable or disable RX Flush for ring 1. See Receive flushing. 0 Disable 1 Enable
3 RX_FLUSH0	RX Flush Ring 0 Enable or disable RX Flush for ring 0. See Receive flushing. 0 Disable 1 Enable
TX_SCHEME	TX scheme configuration Configuration information for DMA to select transmitter queue selection/arbitration scheme. 000 Credit-based scheme 001 Round-robin scheme 010-111 Reserved

11.5.6.48 Reserved Statistic Register (ENET_RMON_T_DROP)

Address: 30BE_0000h base + 200h offset = 30BE_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_DROP field descriptions

Field	Description
Reserved	This read-only field always has the value 0. This field is reserved.

11.5.6.49 Tx Packet Count Statistic Register (ENET_RMON_T_PACKETS)

Address: 30BE_0000h base + 204h offset = 30BE_0204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_PACKETS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Packet count Transmit packet count

11.5.6.50 Tx Broadcast Packets Statistic Register (ENET_RMON_T_BC_PKT)

RMON Tx Broadcast Packets

Address: 30BE_0000h base + 208h offset = 30BE_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_BC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Broadcast packets

11.5.6.51 Tx Multicast Packets Statistic Register (ENET_RMON_T_MC_PKT)

Address: 30BE_0000h base + 20Ch offset = 30BE_020Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_MC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Multicast packets

11.5.6.52 Tx Packets with CRC/Align Error Statistic Register (ENET_RMON_T_CRC_ALIGN)

Address: 30BE_0000h base + 210h offset = 30BE_0210h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_CRC_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Packets with CRC/align error

11.5.6.53 Tx Packets Less Than Bytes and Good CRC Statistic Register (ENET_RMON_T_UNDERSIZE)

Address: 30BE_0000h base + 214h offset = 30BE_0214h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_UNDERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets less than 64 bytes with good CRC

11.5.6.54 Tx Packets GT MAX_FL bytes and Good CRC Statistic Register (ENET_RMON_T_OVERSIZE)

Address: 30BE_0000h base + 218h offset = 30BE_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_OVERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than MAX_FL bytes with good CRC

11.5.6.55 Tx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_T_FRAG)

.

Address: 30BE_0000h base + 21Ch offset = 30BE_021Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_FRAG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of packets less than 64 bytes with bad CRC

11.5.6.56 Tx Packets Greater Than MAX_FL bytes and Bad CRC Statistic Register (ENET_RMON_T_JAB)

Address: 30BE_0000h base + 220h offset = 30BE_0220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_JAB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than MAX_FL bytes and bad CRC

11.5.6.57 Tx Collision Count Statistic Register (ENET_RMON_T_COL)

Address: 30BE_0000h base + 224h offset = 30BE_0224h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_COL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit collisions

11.5.6.58 Tx 64-Byte Packets Statistic Register (ENET_RMON_T_P64)

.

Address: 30BE_0000h base + 228h offset = 30BE_0228h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P64 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 64-byte transmit packets

11.5.6.59 Tx 65- to 127-byte Packets Statistic Register (ENET_RMON_T_P65TO127)

Address: 30BE_0000h base + 22Ch offset = 30BE_022Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P65TO127 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 65- to 127-byte transmit packets

11.5.6.60 Tx 128- to 255-byte Packets Statistic Register (ENET_RMON_T_P128TO255)

Address: 30BE_0000h base + 230h offset = 30BE_0230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P128TO255 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 128- to 255-byte transmit packets

11.5.6.61 Tx 256- to 511-byte Packets Statistic Register (ENET_RMON_T_P256TO511)

Address: 30BE_0000h base + 234h offset = 30BE_0234h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P256TO511 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 256- to 511-byte transmit packets

11.5.6.62 Tx 512- to 1023-byte Packets Statistic Register (ENET_RMON_T_P512TO1023)

.

Address: 30BE_0000h base + 238h offset = 30BE_0238h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P512TO1023 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 512- to 1023-byte transmit packets

11.5.6.63 Tx 1024- to 2047-byte Packets Statistic Register (ENET_RMON_T_P1024TO2047)

Address: 30BE_0000h base + 23Ch offset = 30BE_023Ch

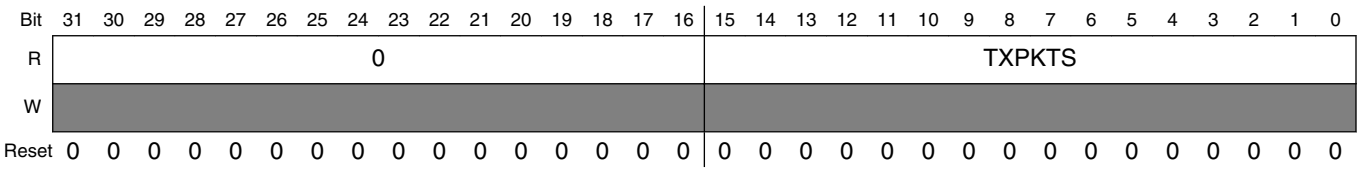
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																TXPKTS															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_T_P1024TO2047 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of 1024- to 2047-byte transmit packets

11.5.6.64 Tx Packets Greater Than 2048 Bytes Statistic Register (ENET_RMON_T_P_GTE2048)

Address: 30BE_0000h base + 240h offset = 30BE_0240h

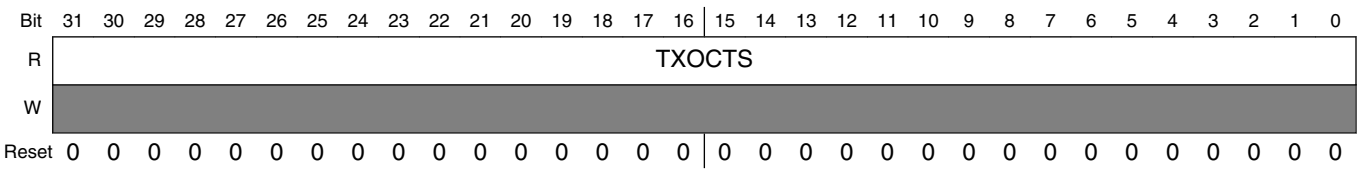


ENET_RMON_T_P_GTE2048 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
TXPKTS	Number of transmit packets greater than 2048 bytes

11.5.6.65 Tx Octets Statistic Register (ENET_RMON_T_OCTETS)

Address: 30BE_0000h base + 244h offset = 30BE_0244h

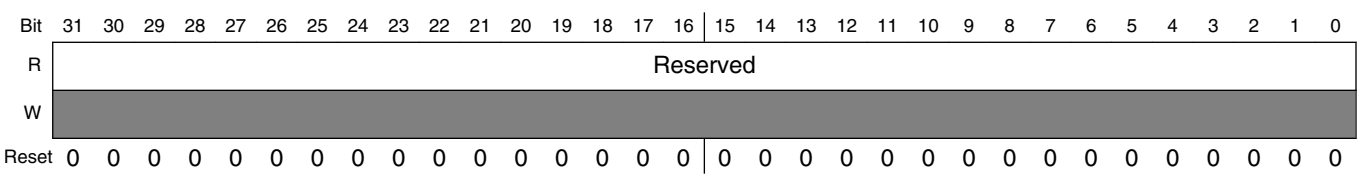


ENET_RMON_T_OCTETS field descriptions

Field	Description
TXOCTS	Number of transmit octets

11.5.6.66 Reserved Statistic Register (ENET_IEEE_T_DROP)

Address: 30BE_0000h base + 248h offset = 30BE_0248h

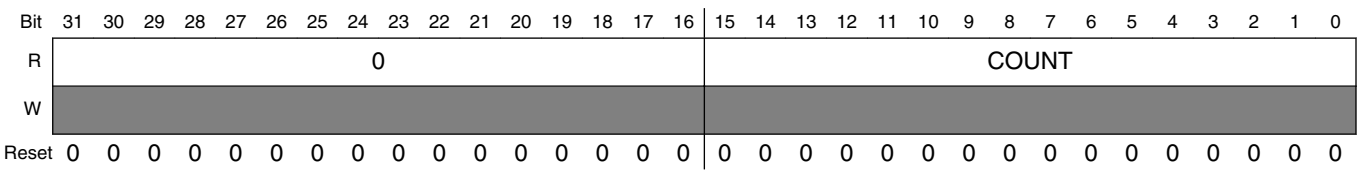


ENET_IEEE_T_DROP field descriptions

Field	Description
Reserved	This read-only field always has the value 0. This field is reserved.

11.5.6.67 Frames Transmitted OK Statistic Register (ENET_IEEE_T_FRAME_OK)

Address: 30BE_0000h base + 24Ch offset = 30BE_024Ch

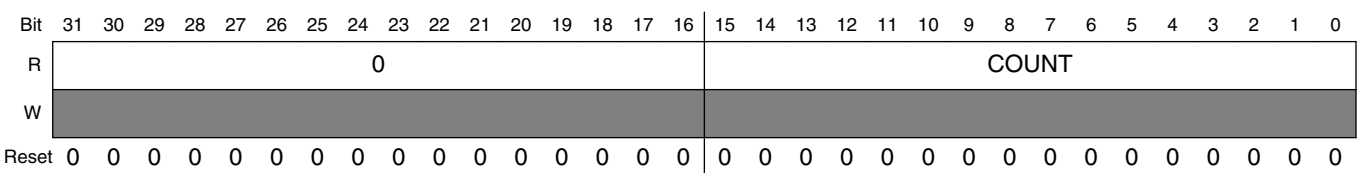


ENET_IEEE_T_FRAME_OK field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted OK **NOTE:** Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

11.5.6.68 Frames Transmitted with Single Collision Statistic Register (ENET_IEEE_T_1COL)

Address: 30BE_0000h base + 250h offset = 30BE_0250h

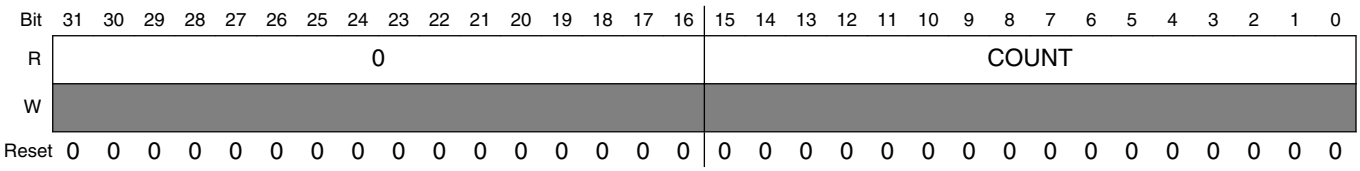


ENET_IEEE_T_1COL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with one collision

11.5.6.69 Frames Transmitted with Multiple Collisions Statistic Register (ENET_IEEE_T_MCOL)

Address: 30BE_0000h base + 254h offset = 30BE_0254h

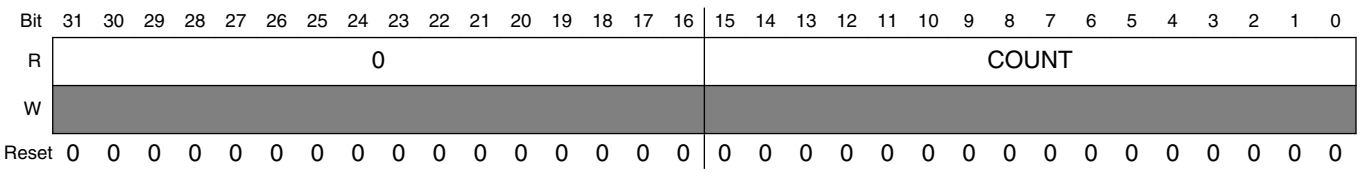


ENET_IEEE_T_MCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with multiple collisions

11.5.6.70 Frames Transmitted after Deferral Delay Statistic Register (ENET_IEEE_T_DEF)

Address: 30BE_0000h base + 258h offset = 30BE_0258h

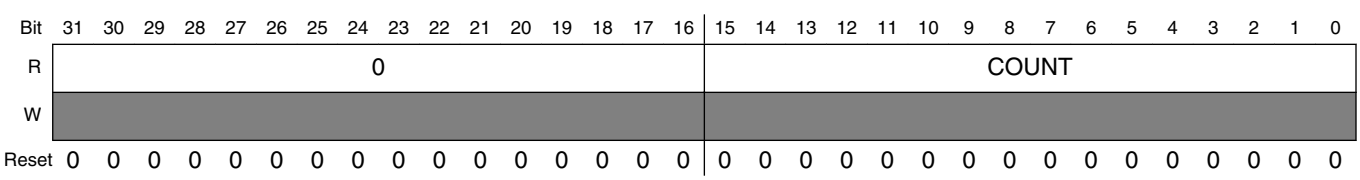


ENET_IEEE_T_DEF field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with deferral delay

11.5.6.71 Frames Transmitted with Late Collision Statistic Register (ENET_IEEE_T_LCOL)

Address: 30BE_0000h base + 25Ch offset = 30BE_025Ch



ENET_IEEE_T_LCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with late collision

11.5.6.72 Frames Transmitted with Excessive Collisions Statistic Register (ENET_IEEE_T_EXCOL)

Address: 30BE_0000h base + 260h offset = 30BE_0260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_EXCOL field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with excessive collisions

11.5.6.73 Frames Transmitted with Tx FIFO Underrun Statistic Register (ENET_IEEE_T_MACERR)

Address: 30BE_0000h base + 264h offset = 30BE_0264h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_MACERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with transmit FIFO underrun

11.5.6.74 Frames Transmitted with Carrier Sense Error Statistic Register (ENET_IEEE_T_CSERR)

Address: 30BE_0000h base + 268h offset = 30BE_0268h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_CSERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames transmitted with carrier sense error

11.5.6.75 Reserved Statistic Register (ENET_IEEE_T_SQE)

Address: 30BE_0000h base + 26Ch offset = 30BE_026Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_SQE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	This read-only field is reserved and always has the value 0. **NOTE:** Counter not implemented as no SQE information is available.

11.5.6.76 Flow Control Pause Frames Transmitted Statistic Register (ENET_IEEE_T_FDXFC)

Address: 30BE_0000h base + 270h offset = 30BE_0270h

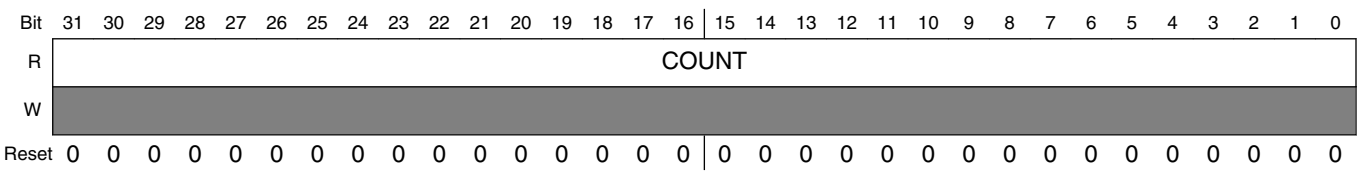
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_T_FDXFC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of flow-control pause frames transmitted

11.5.6.77 Octet Count for Frames Transmitted w/o Error Statistic Register (ENET_IEEE_T_OCTETS_OK)

Address: 30BE_0000h base + 274h offset = 30BE_0274h

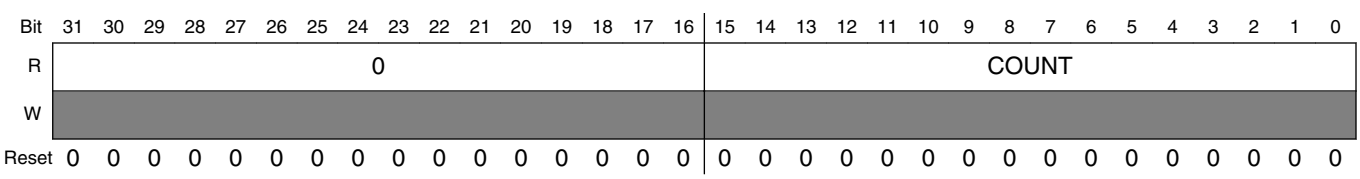


ENET_IEEE_T_OCTETS_OK field descriptions

Field	Description
COUNT	Octet count for frames transmitted without error **NOTE** Counts total octets (includes header and FCS fields).

11.5.6.78 Rx Packet Count Statistic Register (ENET_RMON_R_PACKETS)

Address: 30BE_0000h base + 284h offset = 30BE_0284h



ENET_RMON_R_PACKETS field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of packets received

11.5.6.79 Rx Broadcast Packets Statistic Register (ENET_RMON_R_BC_PKT)

Address: 30BE_0000h base + 288h offset = 30BE_0288h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_BC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive broadcast packets

11.5.6.80 Rx Multicast Packets Statistic Register (ENET_RMON_R_MC_PKT)

Address: 30BE_0000h base + 28Ch offset = 30BE_028Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_MC_PKT field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive multicast packets

11.5.6.81 Rx Packets with CRC/Align Error Statistic Register (ENET_RMON_R_CRC_ALIGN)

Address: 30BE_0000h base + 290h offset = 30BE_0290h

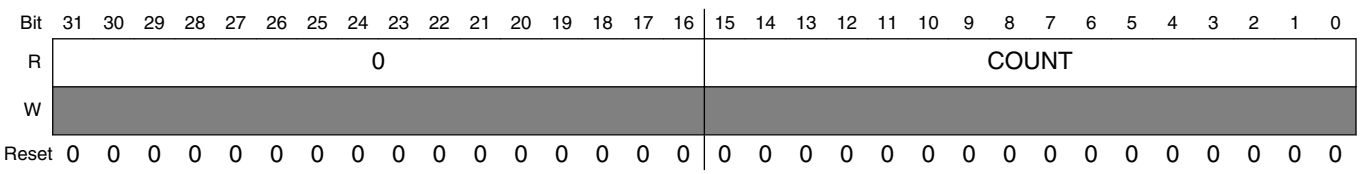
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_CRC_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with CRC or align error

11.5.6.82 Rx Packets with Less Than 64 Bytes and Good CRC Statistic Register (ENET_RMON_R_UNDERSIZE)

Address: 30BE_0000h base + 294h offset = 30BE_0294h

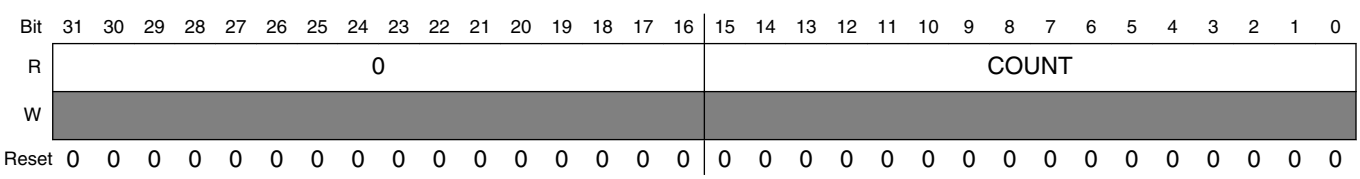


ENET_RMON_R_UNDERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with less than 64 bytes and good CRC

11.5.6.83 Rx Packets Greater Than MAX_FL and Good CRC Statistic Register (ENET_RMON_R_OVERSIZE)

Address: 30BE_0000h base + 298h offset = 30BE_0298h



ENET_RMON_R_OVERSIZE field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets greater than MAX_FL and good CRC

11.5.6.84 Rx Packets Less Than 64 Bytes and Bad CRC Statistic Register (ENET_RMON_R_FRAG)

Address: 30BE_0000h base + 29Ch offset = 30BE_029Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_FRAG field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets with less than 64 bytes and bad CRC

11.5.6.85 Rx Packets Greater Than MAX_FL Bytes and Bad CRC Statistic Register (ENET_RMON_R_JAB)

Address: 30BE_0000h base + 2A0h offset = 30BE_02A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_JAB field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of receive packets greater than MAX_FL and bad CRC

11.5.6.86 Reserved Statistic Register (ENET_RMON_R_RESVD_0)

Address: 30BE_0000h base + 2A4h offset = 30BE_02A4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_RESVD_0 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
Reserved	This field is reserved. This read-only field is reserved and always has the value 0.

11.5.6.87 Rx 64-Byte Packets Statistic Register (ENET_RMON_R_P64)

Address: 30BE_0000h base + 2A8h offset = 30BE_02A8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P64 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 64-byte receive packets

11.5.6.88 Rx 65- to 127-Byte Packets Statistic Register (ENET_RMON_R_P65TO127)

Address: 30BE_0000h base + 2ACh offset = 30BE_02ACh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P65TO127 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 65- to 127-byte recieve packets

11.5.6.89 Rx 128- to 255-Byte Packets Statistic Register (ENET_RMON_R_P128TO255)

Address: 30BE_0000h base + 2B0h offset = 30BE_02B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P128TO255 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 128- to 255-byte recieve packets

11.5.6.90 Rx 256- to 511-Byte Packets Statistic Register (ENET_RMON_R_P256TO511)

Address: 30BE_0000h base + 2B4h offset = 30BE_02B4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P256TO511 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 256- to 511-byte recieve packets

11.5.6.91 Rx 512- to 1023-Byte Packets Statistic Register (ENET_RMON_R_P512TO1023)

Address: 30BE_0000h base + 2B8h offset = 30BE_02B8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P512TO1023 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 512- to 1023-byte recieve packets

11.5.6.92 Rx 1024- to 2047-Byte Packets Statistic Register (ENET_RMON_R_P1024TO2047)

Address: 30BE_0000h base + 2BCh offset = 30BE_02BCh

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P1024TO2047 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of 1024- to 2047-byte recieve packets

11.5.6.93 Rx Packets Greater than 2048 Bytes Statistic Register (ENET_RMON_R_P_GTE2048)

Address: 30BE_0000h base + 2C0h offset = 30BE_02C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_P_GTE2048 field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of greater-than-2048-byte recieve packets

11.5.6.94 Rx Octets Statistic Register (ENET_RMON_R_OCTETS)

Address: 30BE_0000h base + 2C4h offset = 30BE_02C4h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_RMON_R_OCTETS field descriptions

Field	Description
COUNT	Number of receive octets

11.5.6.95 Frames not Counted Correctly Statistic Register (ENET_IEEE_R_DROP)

Counter increments if a frame with invalid or missing SFD character is detected and has been dropped. None of the other counters increments if this counter increments.

Address: 30BE_0000h base + 2C8h offset = 30BE_02C8h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_DROP field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Frame count

11.5.6.96 Frames Received OK Statistic Register (ENET_IEEE_R_FRAME_OK)

Address: 30BE_0000h base + 2CCh offset = 30BE_02CCh

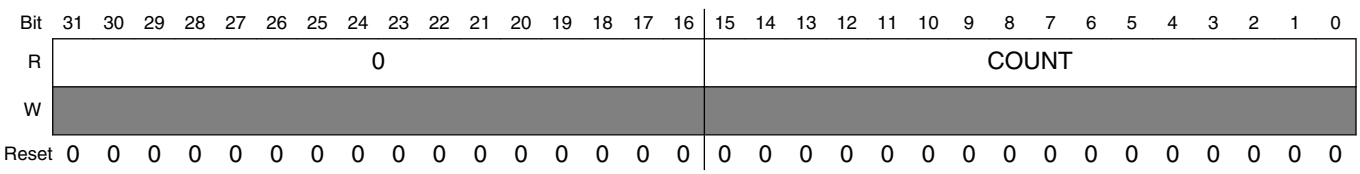
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_IEEE_R_FRAME_OK field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received OK

11.5.6.97 Frames Received with CRC Error Statistic Register (ENET_IEEE_R_CRC)

Address: 30BE_0000h base + 2D0h offset = 30BE_02D0h

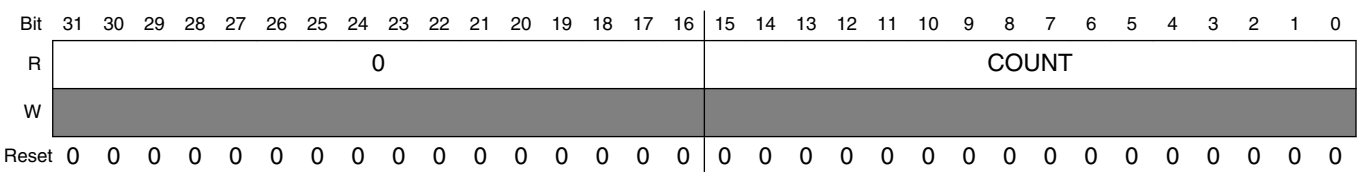


ENET_IEEE_R_CRC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received with CRC error

11.5.6.98 Frames Received with Alignment Error Statistic Register (ENET_IEEE_R_ALIGN)

Address: 30BE_0000h base + 2D4h offset = 30BE_02D4h

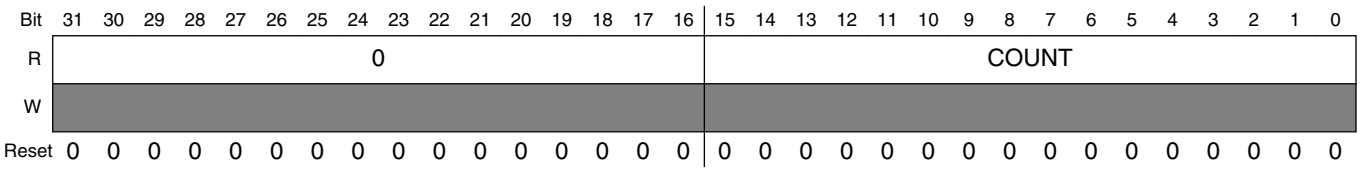


ENET_IEEE_R_ALIGN field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of frames received with alignment error

11.5.6.99 Receive FIFO Overflow Count Statistic Register (ENET_IEEE_R_MACERR)

Address: 30BE_0000h base + 2D8h offset = 30BE_02D8h

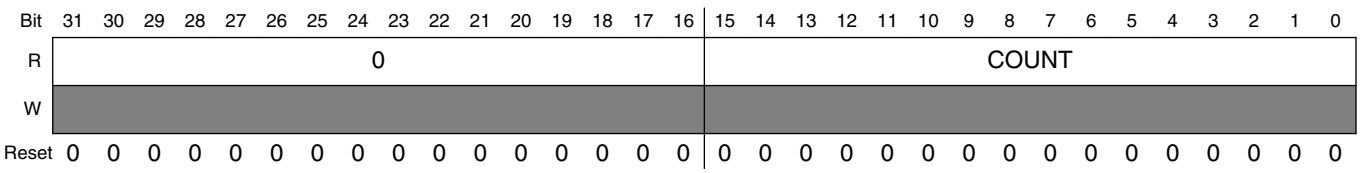


ENET_IEEE_R_MACERR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Receive FIFO overflow count

11.5.6.100 Flow Control Pause Frames Received Statistic Register (ENET_IEEE_R_FDXFC)

Address: 30BE_0000h base + 2DCh offset = 30BE_02DCh

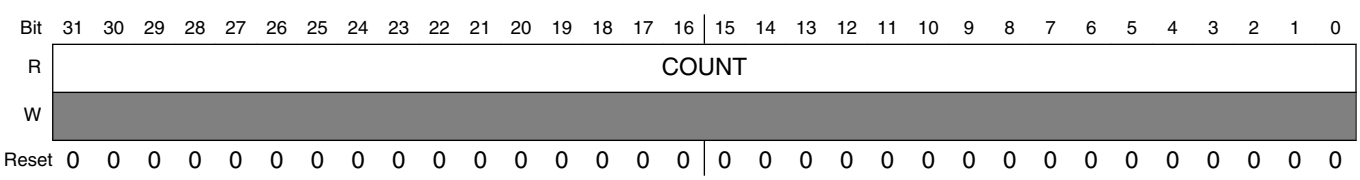


ENET_IEEE_R_FDXFC field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Number of flow-control pause frames received

11.5.6.101 Octet Count for Frames Received without Error Statistic Register (ENET_IEEE_R_OCTETS_OK)

Address: 30BE_0000h base + 2E0h offset = 30BE_02E0h



ENET_IEEE_R_OCTETS_OK field descriptions

Field	Description
COUNT	Number of octets for frames received without error **NOTE:** Counts total octets (includes header and FCS fields). Does not increment for the broadcast frames when broadcast reject is enabled and promiscuous mode is disabled within the receive control register (RCR).

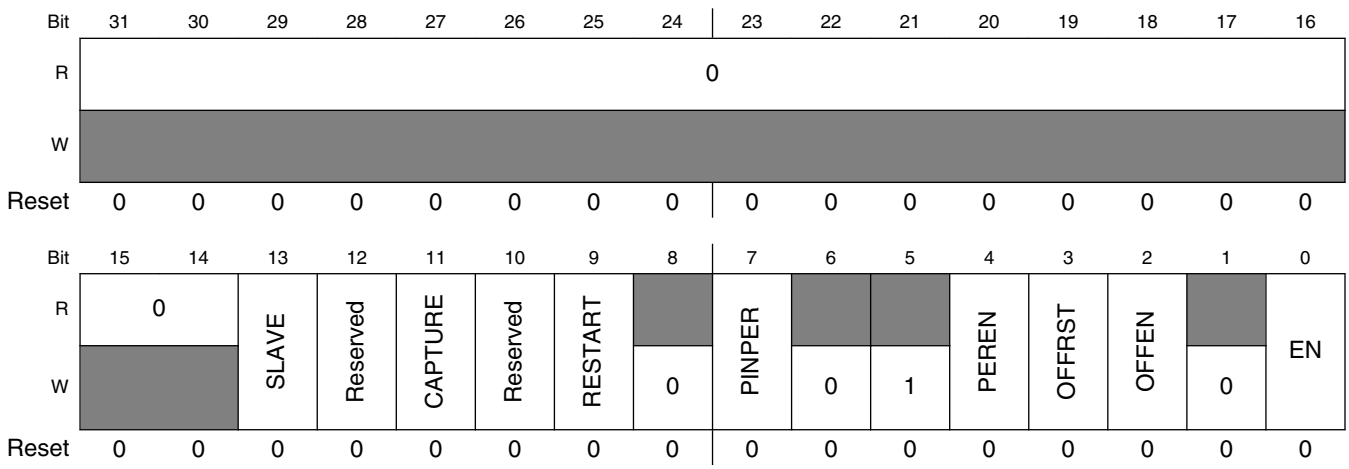
11.5.6.102 Adjustable Timer Control Register (ENET_ATCR)

ATCR command fields can trigger the corresponding events directly. It is not necessary to preserve any of the configuration fields when a command field is set in the register, that is, no read-modify-write is required.

NOTE

The CAPTURE and RESTART fields and bits 12 and 10 must be 0 in order to write to the other fields in this register.

Address: 30BE_0000h base + 400h offset = 30BE_0400h



ENET_ATCR field descriptions

Field	Description
31–14 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
13 SLAVE	Enable Timer Slave Mode 0 The timer is active and all configuration fields in this register are relevant. 1 The internal timer is disabled and the externally provided timer value is used. All other fields, except CAPTURE, in this register have no effect. CAPTURE can still be used to capture the current timer value.

Table continues on the next page...

ENET_ATCR field descriptions (continued)

Field	Description
12 Reserved	This field is reserved. Always write 0 to this field.
11 CAPTURE	Capture Timer Value When this field is set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. **NOTE:** To ensure that the correct time value is read from the ATVR register, a minimum amount of time must elapse from issuing this command to reading the ATVR register. This minimum time is defined by the greater of either six register clock cycles or six 1588/timestamp clock cycles. 0 No effect. 1 The current time is captured and can be read from the ATVR register.
10 Reserved	This field is reserved. Always write 0 to this field.
9 RESTART	Reset Timer Resets the timer to zero. This has no effect on the counter enable. If the counter is enabled when this field is set, the timer is reset to zero and starts counting from there. When set, all other fields are ignored during a write. This field automatically clears to 0 after the command completes. RESTART should be used when the timer is enabled. **NOTE:** The Reset Timer command requires at least 6 clock cycles of either the register clock or the 1588/timestamp clock, whichever is greater, to complete.
8 Reserved	This field is reserved.
7 PINPER	Enables event signal output assertion on period event. **NOTE:** Not all devices contain the event signal output. See the chip configuration details. 0 Disable. 1 Enable.
6 Reserved	This field is reserved.
5 Reserved	This field is reserved. **NOTE:** This field must be written always with one.
4 PEREN	Enable Periodical Event 0 Disable. 1 A period event interrupt can be generated (EIR[TS_TIMER]) and the event signal output is asserted when the timer wraps around according to the periodic setting ATPER. The timer period value must be set before setting this bit. **NOTE:** Not all devices contain the event signal output. See the chip configuration details.
3 OFFRST	Reset Timer On Offset Event 0 The timer is not affected and no action occurs, besides clearing OFFEN, when the offset is reached. 1 If OFFEN is set, the timer resets to zero when the offset setting is reached. The offset event does not cause a timer interrupt.
2 OFFEN	Enable One-Shot Offset Event

Table continues on the next page...

ENET_ATCR field descriptions (continued)

Field	Description
	0 Disable. 1 The timer can be reset to zero when the given offset time is reached (offset event). The field is cleared when the offset event is reached, so no further event occurs until the field is set again. The timer offset value must be set before setting this field.
1 Reserved	This field is reserved.
0 EN	Enable Timer 0 The timer stops at the current value. 1 The timer starts incrementing.

11.5.6.103 Timer Value Register (ENET_ATVR)

Address: 30BE_0000h base + 404h offset = 30BE_0404h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ATIME																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ATVR field descriptions

Field	Description
ATIME	A write sets the timer. A read returns the last captured value. To read the current value, issue a capture command (i.e., set ATCR[CAPTURE]) prior to reading this register.

11.5.6.104 Timer Offset Register (ENET_ATOFF)

Address: 30BE_0000h base + 408h offset = 30BE_0408h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OFFSET																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ATOFF field descriptions

Field	Description
OFFSET	Offset value for one-shot event generation. When the timer reaches the value, an event can be generated to reset the counter. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds.

11.5.6.105 Timer Period Register (ENET_ATPER)

Address: 30BE_0000h base + 40Ch offset = 30BE_040Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PERIOD																															
Reset	0	0	1	1	1	0	1	1	1	0	0	1	1	0	1	0	1	1	0	0	1	0	1	0	0	0	0	0	0	0	0	0

ENET_ATPER field descriptions

Field	Description
PERIOD	Value for generating periodic events. Each instance the timer reaches this value, the period event occurs and the timer restarts. If the increment value in ATINC is given in true nanoseconds, this value is also given in true nanoseconds. The value should be initialized to 1,000,000,000 (1×10^9) to represent a timer wrap around of one second. The increment value set in ATINC should be set to the true nanoseconds of the period of clock ts_clk, hence implementing a true 1 second counter. **NOTE:** The value of PERIOD has the following constraint: 2^{32} – ENET_ATINC[INC_COR] – 3×ENET_ATINC[INC] ≥ PERIOD > 0.

11.5.6.106 Timer Correction Register (ENET_ATCOR)

Address: 30BE_0000h base + 410h offset = 30BE_0410h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0	COR														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

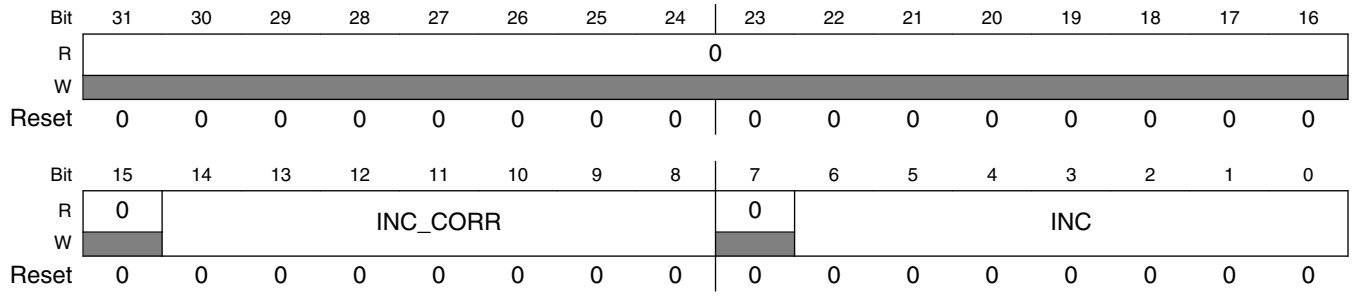
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

ENET_ATCOR field descriptions

Field	Description
31 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COR	Correction Counter Wrap-Around Value Defines after how many timer clock cycles (ts_clk) the correction counter should be reset and trigger a correction increment on the timer. The amount of correction is defined in ATINC[INC_CORR]. A value of 0 disables the correction counter and no corrections occur. **NOTE:** This value is given in clock cycles, not in nanoseconds as all other values.

11.5.6.107 Time-Stamping Clock Period Register (ENET_ATINC)

Address: 30BE_0000h base + 414h offset = 30BE_0414h

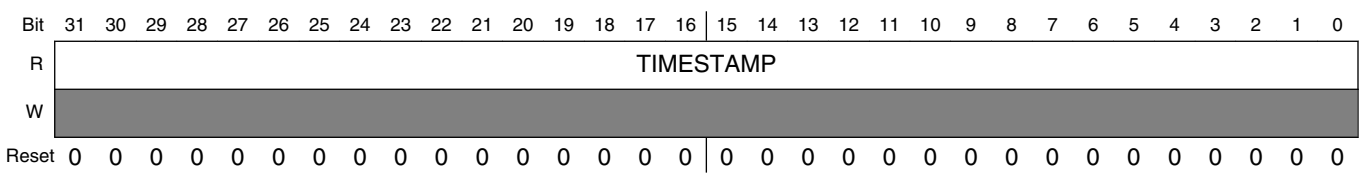


ENET_ATINC field descriptions

Field	Description
31–15 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
14–8 INC_CORR	Correction Increment Value This value is added every time the correction timer expires (every clock cycle given in ATCOR). A value less than INC slows down the timer. A value greater than INC speeds up the timer.
7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
INC	Clock Period Of The Timestamping Clock (ts_clk) In Nanoseconds The timer increments by this amount each clock cycle. For example, set to 10 for 100 MHz, 8 for 125 MHz, 5 for 200 MHz. **NOTE:** For highest precision, use a value that is an integer fraction of the period set in ATPER.

11.5.6.108 Timestamp of Last Transmitted Frame (ENET_ATSTMP)

Address: 30BE_0000h base + 418h offset = 30BE_0418h

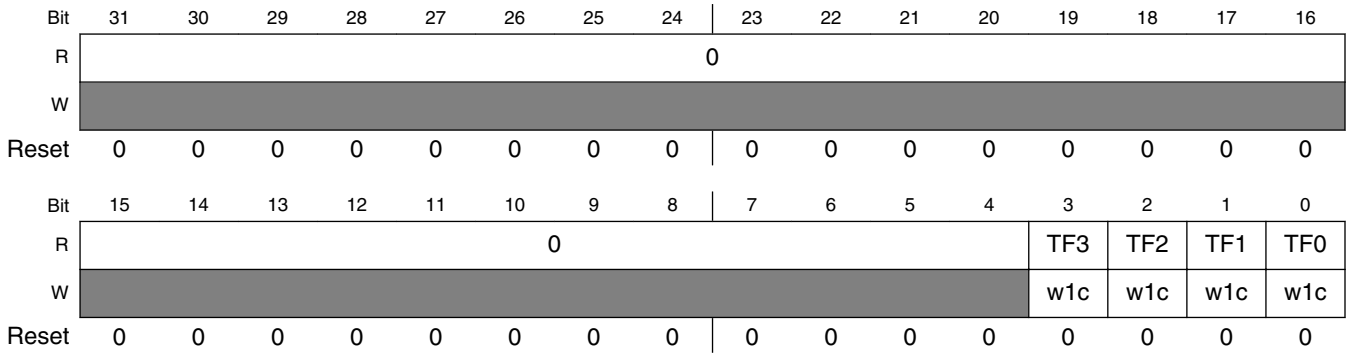


ENET_ATSTMP field descriptions

Field	Description
TIMESTAMP	Timestamp of the last frame transmitted by the core that had TxBD[TS] set . This register is only valid when EIR[TS_AVAIL] is set.

11.5.6.109 Timer Global Status Register (ENET_TGSR)

Address: 30BE_0000h base + 604h offset = 30BE_0604h

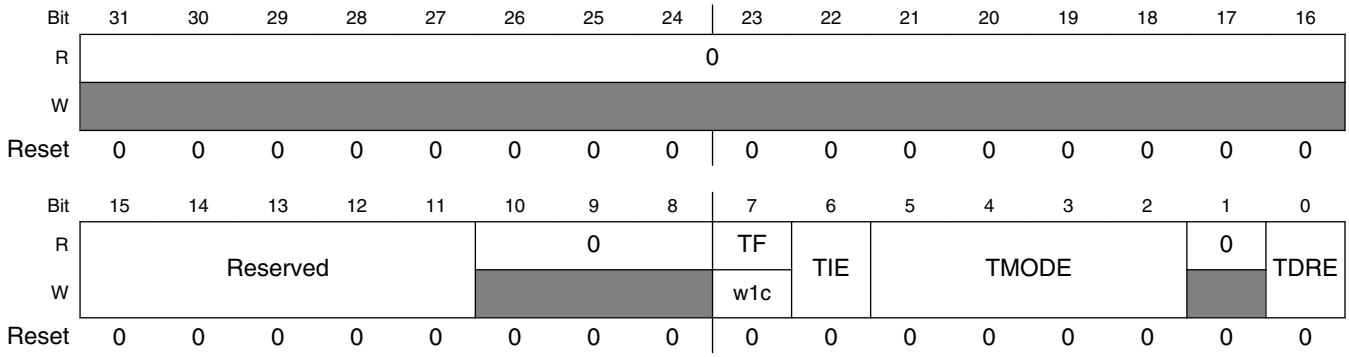


ENET_TGSR field descriptions

Field	Description
31–4 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
3 TF3	Copy Of Timer Flag For Channel 3 0 Timer Flag for Channel 3 is clear 1 Timer Flag for Channel 3 is set
2 TF2	Copy Of Timer Flag For Channel 2 0 Timer Flag for Channel 2 is clear 1 Timer Flag for Channel 2 is set
1 TF1	Copy Of Timer Flag For Channel 1 0 Timer Flag for Channel 1 is clear 1 Timer Flag for Channel 1 is set
0 TF0	Copy Of Timer Flag For Channel 0 0 Timer Flag for Channel 0 is clear 1 Timer Flag for Channel 0 is set

11.5.6.110 Timer Control Status Register (ENET_TCSR*n*)

Address: 30BE_0000h base + 608h offset + (8d × i), where i=0d to 3d



ENET_TCSR*n* field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
15–11 Reserved	This field is reserved. This field must be written with 0.
10–8 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
7 TF	Timer Flag Sets when input capture or output compare occurs. This flag is double buffered between the module clock and 1588 clock domains. When this field is 1, it can be cleared to 0 by writing 1 to it. 0 Input Capture or Output Compare has not occurred. 1 Input Capture or Output Compare has occurred.
6 TIE	Timer Interrupt Enable 0 Interrupt is disabled 1 Interrupt is enabled
5–2 TMODE	Timer Mode Updating the Timer Mode field takes a few cycles to register because it is synchronized to the 1588 clock. The version of Timer Mode returned on a read is from the 1588 clock domain. When changing Timer Mode, always disable the channel and read this register to verify the channel is disabled first. 0000 Timer Channel is disabled. 0001 Timer Channel is configured for Input Capture on rising edge. 0010 Timer Channel is configured for Input Capture on falling edge. 0011 Timer Channel is configured for Input Capture on both edges. 0100 Timer Channel is configured for Output Compare - software only. 0101 Timer Channel is configured for Output Compare - toggle output on compare. 0110 Timer Channel is configured for Output Compare - clear output on compare. 0111 Timer Channel is configured for Output Compare - set output on compare. 1000 Reserved

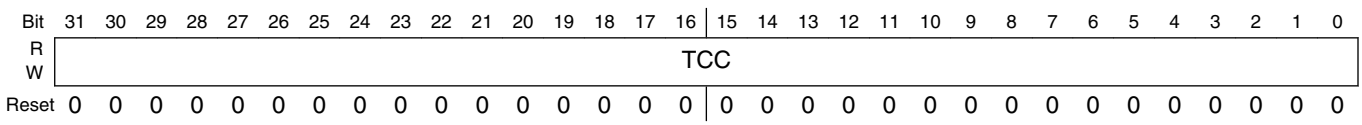
Table continues on the next page...

ENET_TCSR*n* field descriptions (continued)

Field	Description
	1010 Timer Channel is configured for Output Compare - clear output on compare, set output on overflow. 10X1 Timer Channel is configured for Output Compare - set output on compare, clear output on overflow. 110X Reserved 1110 Timer Channel is configured for Output Compare - pulse output low on compare for one 1588-clock cycle. 1111 Timer Channel is configured for Output Compare - pulse output high on compare for one 1588-clock cycle.
1 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
0 TDRE	Timer DMA Request Enable 0 DMA request is disabled 1 DMA request is enabled

11.5.6.111 Timer Compare Capture Register (ENET_TCCR*n*)

Address: 30BE_0000h base + 60Ch offset + (8d × i), where i=0d to 3d



ENET_TCCR*n* field descriptions

Field	Description
TCC	Timer Capture Compare This register is double buffered between the module clock and 1588 clock domains. When configured for compare, the 1588 clock domain updates with the value in the module clock domain whenever the Timer Channel is first enabled and on each subsequent compare. Write to this register with the first compare value before enabling the Timer Channel. When the Timer Channel is enabled, write the second compare value either immediately, or at least before the first compare occurs. After each compare, write the next compare value before the previous compare occurs and before clearing the Timer Flag. The compare occurs one 1588 clock cycle after the IEEE 1588 Counter increments past the compare value in the 1588 clock domain. If the compare value is less than the value of the 1588 Counter when the Timer Channel is first enabled, then the compare does not occur until following the next overflow of the 1588 Counter. If the compare value is greater than the IEEE 1588 Counter when the 1588 Counter overflows, or the compare value is less than the value of the IEEE 1588 Counter after the overflow, then the compare occurs one 1588 clock cycle following the overflow. When configured for capture, the value of the IEEE 1588 Counter is captured into the 1588 clock domain and then updated into the module clock domain, provided the Timer Flag is clear. Always read the capture value before clearing the Timer Flag.

Chapter 12
Timers

12.1 General Purpose Timer (GPT)

12.1.1 Overview

This chapter describes the General Purpose Timer (GPT) module interface. It is also a reference for software driver programming. The GPT has a 32-bit up-counter. The timer counter value can be captured in a register using an event on an external pin. The capture trigger can be programmed to be a rising or/and falling edge. The GPT can also generate an event on the output compare pins and an interrupt when the timer reaches a programmed value. The GPT has a 12-bit prescaler, which provides a programmable clock frequency derived from multiple clock sources.

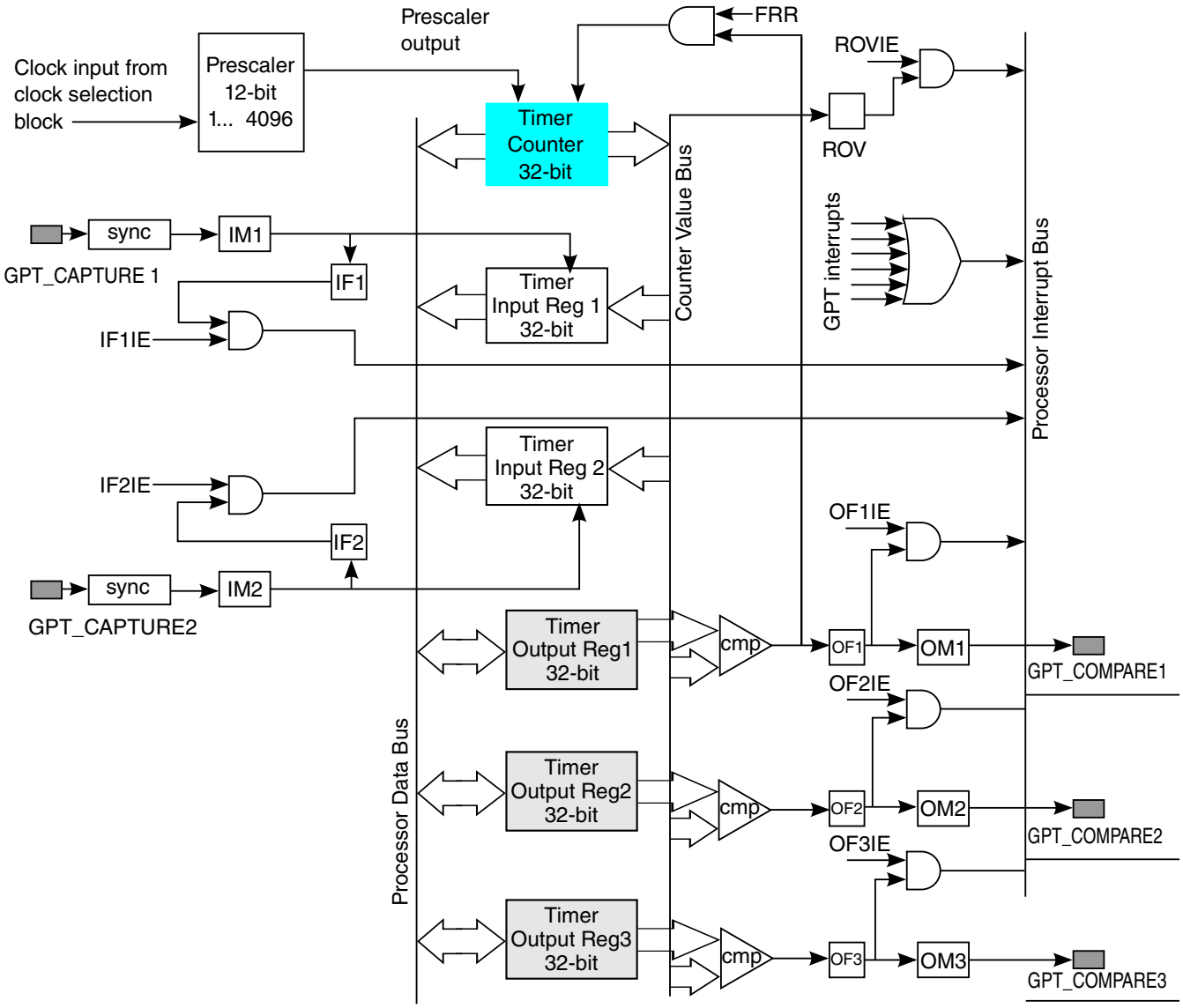


Figure 12-1. GPT Block Diagram

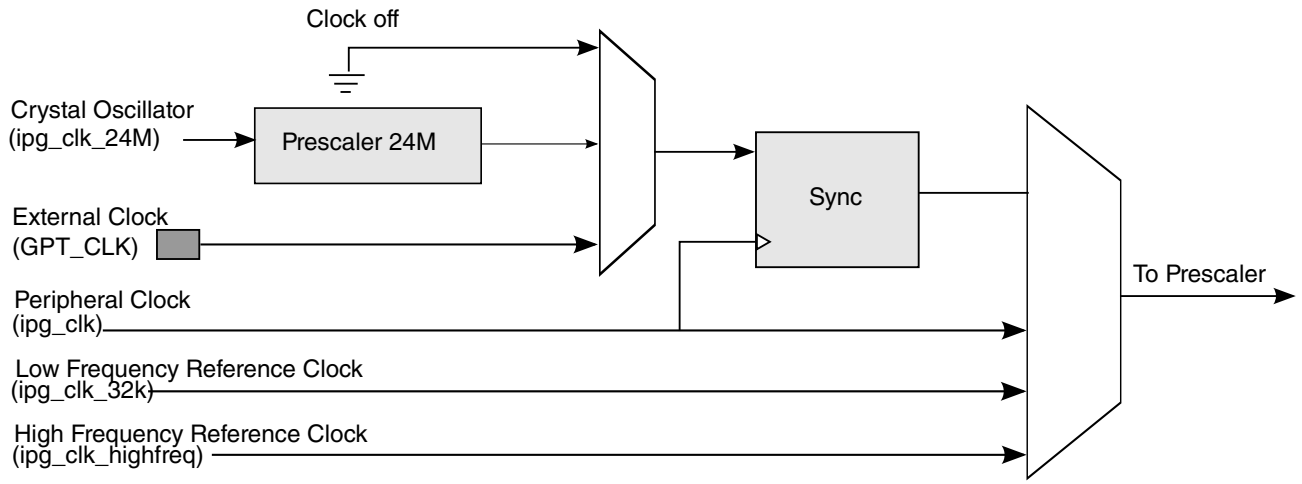


Figure 12-2. GPT Counter Clocks Diagram

12.1.1.1 Features

- One 32-bit up-counter with clock source selection, including external clock.
- Two input capture channels with a programmable trigger edge.
- Three output compare channels with a programmable output mode. A "forced compare" feature is also available.
- Can be programmed to be *active* in low power and debug modes.
- Interrupt generation at capture, compare, and rollover events.
- Restart or free-run modes for counter operations.

12.1.1.2 Modes and Operation

The GPT supports the modes described in the indicated sections:

- Operating Modes
  - Restart Mode
  - Free-Run Mode

12.1.2 External Signals

The GPT follows the IP Bus protocol for interfacing with the processor core. The GPT does not have *any interface signals with any other module inside the chip,* except for the clock and reset inputs (from the clock and reset controller module) and for the interrupt signals *to* the processor interrupt handler. There are functional and clock inputs, and functional output signals going outside the chip boundary.

There are six signals (three input, three output) in the GPT module that *can be* connected to the chip pads.

12.1.2.1 External Clock Input

The GPT counter can be operated using an external clock from outside the device, and this is the input pin used for that purpose. The external clock input (GPT_CLK) is treated as asynchronous to the peripheral clock (ipg_clk). To ensure proper operations of GPT, the external clock input frequency should be less than 1/4 of frequency of the peripheral clock (ipg_clk). Hysteresis characteristics on this pad will be required because this is a clock input.

12.1.2.2 Input Capture Trigger Signals

The GPT counter value can be stored in a register, triggered by an event from *outside the device*. A positive or/and negative edge on these signals GPT_CAPTURE1 , GPT_CAPTURE2 can trigger this capture event. These signals are treated as asynchronous to the peripheral clock (ipg_clk). Only those transitions which occur *at least a single clock cycle* (the clock selected to run the counter) *after the previous recorded transition* are guaranteed to trigger a capture event.

12.1.2.3 Output Compare Signals

The output compare signals: GPT_COMPARE1, GPT_COMPARE2, GPT_COMPARE3, indicate that output compare events have gone through a specified transition.

12.1.3 Clocks

The clock that is input to the prescaler can be selected from 4 clock sources. The following table describes the clock sources for GPT. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 12-1. GPT Clocks

Clock name	Clock Root	Description
ipg_clk		Peripheral clock
ipg_clk_32k		Low-frequency reference clock (32 kHz)

Table continues on the next page...

Table 12-1. GPT Clocks (continued)

Clock name	Clock Root	Description
ipg_clk_highfreq		High-frequency reference clock
ipg_clk_s		Peripheral access clock

- High-Frequency Clock (ipg_clk_highfreq)

  Provided by the Clock Controller Module (CCM), the High Frequency Clock is intended to be ON in Normal Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the High Frequency Clock *in Normal Power mode*. The CCM is expected to provide this clock *after* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the High Frequency Clock in a Low Power mode.

- Low-Reference Clock (ipg_clk_32k)

  This 32 kHz Low Reference Clock (provided by the CCM) is intended to be ON in Low Power mode when the Peripheral Clock (ipg_clk) is turned OFF, thereby enabling the GPT to be operated using the Low Reference Clock in Low Power mode. The CCM is expected to provide the Low Reference Clock *after* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode; the CCM is also expected to switch to the *unsynchronized* version of the Low Reference Clock in a Low Power mode.

- Peripheral Clock (ipg_clk)

  If the Peripheral Clock (ipg_clk) or the External Clock (GPT_CLK) is selected (CLKSRC=001 or 011) as Clock Source, then the Peripheral Clock will be ON in normal GPT operations. In Low Power modes, if the GPT is programmed to be disabled (STOPEN or WAITEN or DOZEN=0), then the Peripheral Clock (ipg_clk) can be switched OFF.

- External Clock (GPT_CLK)

  The External Clock comes from *outside the device* and can be selected to run the GPT counter. The External Clock is treated as *asynchronous to the Peripheral Clock,* (ipg_clk) and is synchronized to the Peripheral Clock (ipg_clk), *inside* the module. Therefore, the External Clock frequency is limited to < 1/4 frequency of the Peripheral Clock (ipg_clk), for proper GPT operations. Note that in Low Power modes, *if* the Peripheral Clock (ipg_clk) is not available, then the External Clock *cannot be used* to run the counter.

- Crystal Oscillator Clock (ipg_clk_24M)

This 24 MHz Crystal Oscillator Clock (provided by the CCM) is intended to be used against frequency change of Peripheral Clock (ipg_clk) changes to provide a more accurate timer clock for operation system. The CCM is expected to provide the 24 MHz Crystal Oscillator Clock *without* synchronizing it to the System Bus Clock (ahb_clk) in Normal functional mode. Synchronization is done in GPT module. Before synchronization, the 24 MHz Crystal Oscillator Clock is divided by a 24 MHz clock prescaler, to make sure the clock frequency less than half of System Bus Clock (ahb_clk).

The clock input source is configured using the clock source field (CLKSRC, in the GPT_CR control register). The clock input to the prescaler can be disabled by programming the CLKSRC bits (of the GPT_CR control register) to 000. **The CLKSRC field value should be changed** *only after disabling the GPT* (by setting the EN bit in the GPT_CR to 0).

The PRESCALER field selects the divide ratio of the input clock that drives the main counter. The prescaler can divide the input clock by a value (from 1 to 4096) and can be changed *at any time*. A change in the value of the PRESCALER field *immediately affects* the output clock frequency.

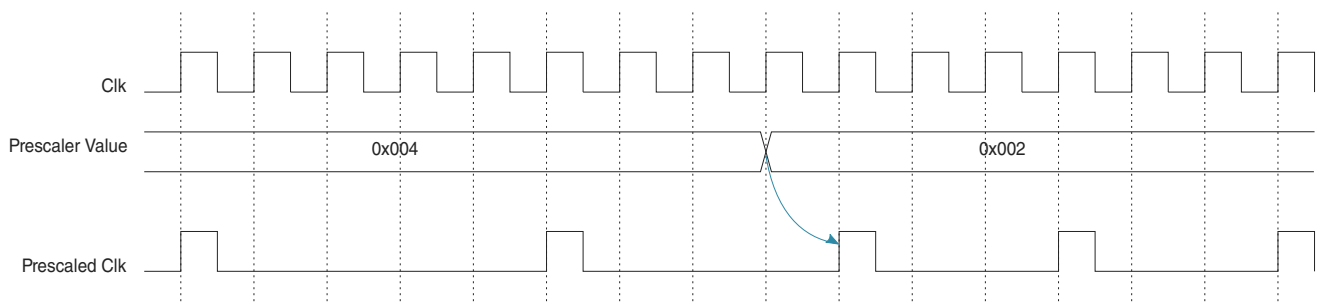


Figure 12-3. Prescaler Value Change Timing Diagram

12.1.4 Functional Description

This section provides a complete functional description of the GPT.

12.1.4.1 Operating Modes

The GPT counter can be programmed to work in either of two modes: Restart mode or Free-Run mode.

12.1.4.1.1 Restart Mode

In Restart mode (selectable through the GPT Control Register GPT_CR), when the counter reaches the compared value, the counter resets and starts again from 0x00000000. The Restart feature is associated only with Compare Channel 1.

Any write access to the Compare register of Channel 1 will reset the GPT counter. This is done to avoid possibly missing a compare event when compare value is changed from a higher value to lower value while counting is proceeding.

For the other two compare channels, when the compare event occurs the counter is *not reset*.

12.1.4.1.2 Free-Run Mode

In Free-Run mode, when compare events occur for all 3 channels, the counter is *not reset*; instead the counter continues to count until 0xffffffff, and then rolls over (to 0x00000000).

12.1.4.2 Operation

The General Purpose Timer (GPT) has a single counter (GPT_CNT) that is a 32-bit free-running *up-counter,* which starts counting *after it is enabled by software* (EN=1). The counter's clock source is the output of the prescaler labelled "Prescaler output" in Figure 12-1.

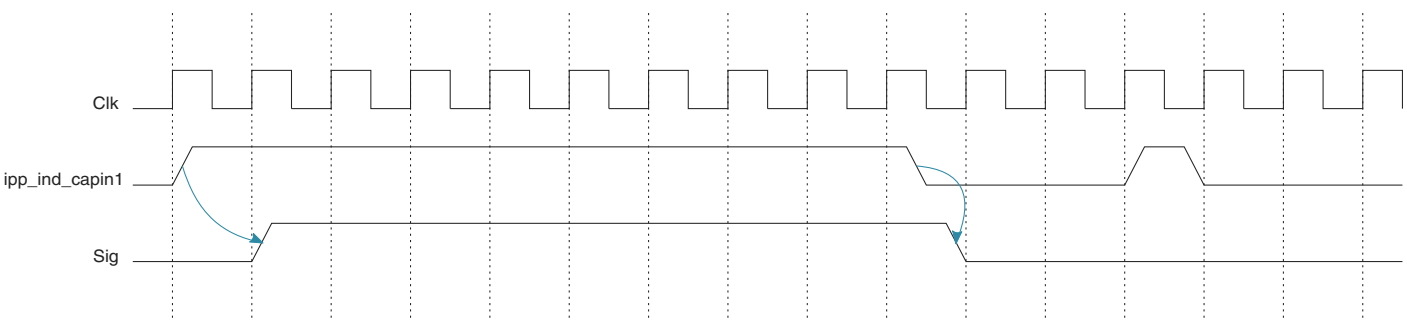
- If the GPT timer is disabled (EN=0), then the Main Counter *and* Prescaler Counter freeze their current count values. The ENMOD bit determines the value of the GPT counter when the EN bit is set and the Counter is enabled again.
  - If the ENMOD bit is set (=1), then the Main Counter and Prescaler Counter values are reset to 0, when GPT is enabled (EN=1).
  - If ENMOD bit is programmed to 0, then the Main Counter and Prescaler Counter restart counting from their frozen values, when GPT is enabled again (EN=1).
- If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter freeze at their current count values *when* GPT enters low power mode. When GPT exits a low power mode, the Main Counter and Prescaler Counter start counting from their frozen values *regardless* of the ENMOD bit value. Note that the GPT_CNT can be read *at any time* by the processor, and that *both* Input Capture Channels use the *same* counter (GPT_CNT).

- A hardware reset resets all the GPT registers to their respective reset values. All registers except the Output Compare Registers (OCR1, OCR2, OCR3) obtain a value of 0x0. The Compare registers are reset to 0xFFFF_FFFF.
- The software reset (SWR bit in the GPT_CR control register) resets *all* of the register bits *except* the EN, ENMOD, STOPEN, WAITEN, and DBGEN bits. The state of these bits is not affected by a software reset. Note that a software reset can be given *while the GPT is disabled*.

12.1.4.2.1 Input Capture

There are two Input Capture Channels, and each Input Capture Channel has a dedicated capture pin, capture register and input edge detection/selection logic. Each input capture function has an associated status flag, and can cause the processor to make an interrupt service request.

When a selected edge transition occurs on an Input Capture pin, the contents of the GPT_CNT is captured on the corresponding capture register and the appropriate interrupt status flag is set. An interrupt request can be generated when the transition is detected *if* its corresponding enable bit is set (in the Interrupt Register). The capture can be programmed to occur on the input pin's rising edge, falling edge, on both rising and falling edges, or the capture can be disabled. The events are synchronized with the clock that was selected to run the counter. Only those transitions that occur at least one clock cycle (clock selected to run the counter) *after* the previous recorded transition will be guaranteed to trigger a capture event. There can be up to one clock cycle of uncertainty in the latching of the input transition. The Input Capture registers can be read *at any time* without affecting their values.



Clk = The clock selected from CLKSRC bit field setting

ipp_ind_capin1 = Pad signal for capture channel 1

Sig = Capture signal as sensed by the module

Figure 12-4. Input Capture Event Timing

12.1.4.2.2 Output Compare

The three Output Compare Channels *use the same counter* (GPT_CNT) as the Input Capture Channels. When the programmed content of an Output Compare register matches the value in GPT_CNT, an output compare status flag is set and an interrupt is generated (if the corresponding bit is set in the interrupt register). Consequently, the Output Compare timer pin will be set, cleared, toggled, not affected at all or provide an active-low pulse for one input clock period (subject to the restriction on the maximum frequency allowed on the pad) according to the mode bits (that were programmed).

There is also a "forced-compare" feature that allows the software to generate a compare event when required, *without the condition of the counter value that is equal to the compare value*. The action taken as a result of a forced compare is the same as when an output compare match occurs, *except that the status flags are not set and no interrupt can be generated*. Forced channels take programmed action immediately after the write to the force-compare bits. These bits are self-negating and always read as zeros.

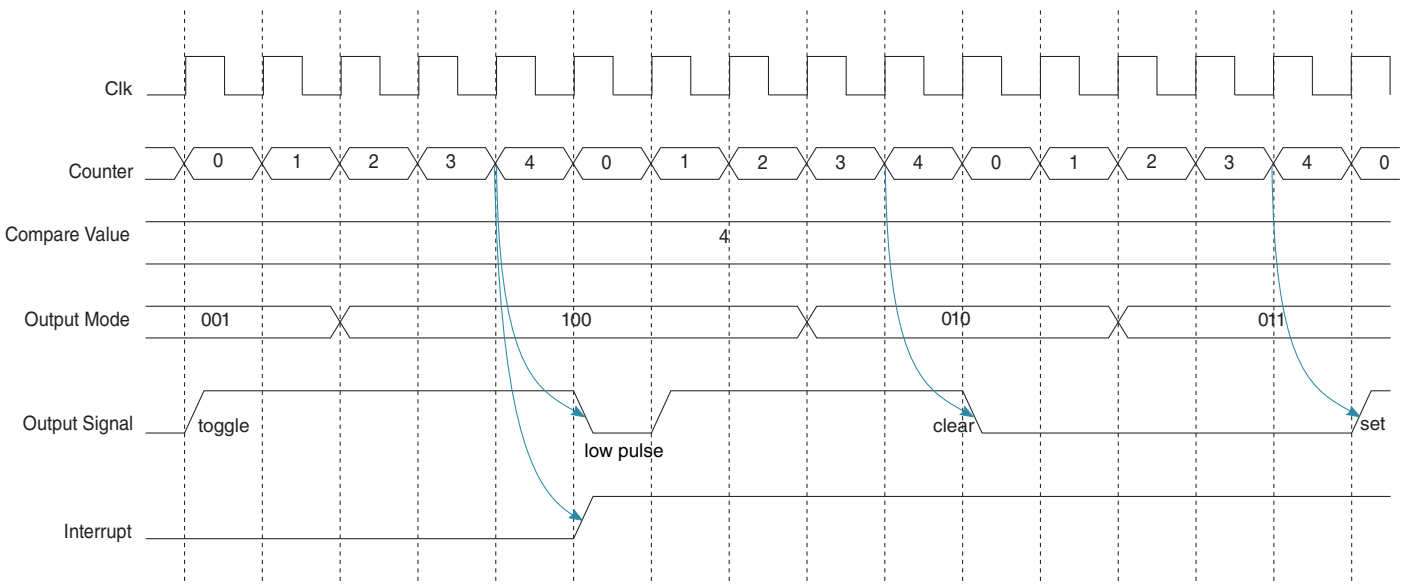


Figure 12-5. Output Compare and Interrupt Timing

12.1.4.2.3 Interrupts

There are 6 different interrupts that are generated by the GPT. If the selected clock for running the counter is available, then *all interrupts can be generated in Low Power and Debug modes*.

- Rollover Interrupt

The Rollover Interrupt is generated when the GPT counter reaches 0xffffffff, then resets to 0x00000000 and continues counting. The Rollover Interrupt is enabled by the ROVIE bit in the GPT_IR register; the associated status bit is the ROV bit in the GPT_SR register.

- Input Capture Interrupt 1, 2

  After a capture event occurs, the associated Input Capture Channel generates an interrupt. The "capture event" interrupts are enabled by the IF2IE and IF1IE bits (in the GPT_IR register); the associated status bits are IF2 and IF1 (in the GPT_SR register). The capture of the counter value because of a capture event is *not affected by a pending capture interrupt*. The Capture register is updated with a new counter value when a capture event occurs, regardless of whether that Capture Channels' interrupt has been serviced or not.

- Output Compare Interrupt 1, 2, 3

  After a compare event occurs, the associated Output Compare Channel generates an interrupt. The "compare event" interrupts are enabled by the OF3IE, OF2IE, and OF1IE bits (in the GPT_IR register); the associated status bits are OF3, OF2, and OF1 (in the GPT_SR register). A "forced compare" does not generate an interrupt.

A *cumulative* interrupt line is also present, which is asserted whenever any of the above interrupts are posted. The cumulative interrupt line has *no* associated enables or status bits.

12.1.4.2.4 Low Power Mode Behavior

In Low Power modes, if the clock from the selected clock source is available (except for the External Clock (GPT_CLK), which can be used *only if* the Peripheral Clock (ipg_clk) is available), the counter will continue to run depending on whether the control bit for that mode is set. If the clock is not present or if the corresponding low power bit in the GPT_CR control register is 0, the Main Counter and the Prescaler Counter freeze at their current values and resume counting (from their frozen values) when the Low Power mode is exited.

12.1.4.2.5 Debug Mode Behavior

In Debug mode, the modules in the device have the option of continuing to run or be halted.

- If the DBGEN bit is set, then the GPT timer will continue to run in Debug mode.
- If the DBGEN bit is not set (in the GPT_CR control register), then the GPT timer is halted.

12.1.5 Initialization/ Application Information

12.1.5.1 Selecting the Clock Source

The CLKSRC field in the GPT_CR register selects the clock source. The CLKSRC field value should be changed only after disabling the GPT (EN=0).

The software sequence to be followed while changing clock source is:

1. Disable GPT by setting EN=0 in GPT_CR register.
2. Disable GPT interrupt register (GPT_IR).
3. Configure Output Mode to unconnected/ disconnected—Write zeros in OM3, OM2, and OM1 in GPT_CR
4. Disable Input Capture Modes—Write zeros in IM1 and IM2 in GPT_CR
5. Change clock source CLKSRC to the desired value in GPT_CR register.
6. Assert the SWR bit in GPT_CR register.
7. Clear GPT status register (GPT_SR) (i.e., w1c).
8. Set ENMOD=1 in GPT_CR register, to bring GPT counter to 0x00000000.
9. Enable GPT (EN=1) in GPT_CR register.
10. Enable GPT interrupt register (GPT_IR).

12.1.6 GPT Memory Map/Register Definition

The GPT has 10 user-accessible 32-bit registers, which are used to configure, operate, and monitor the state of the GPT.

An IP bus write access to the GPT Control Register (GPT_CR) and the GPT Output Compare Register1 (GPT_OCR1) results in *one cycle of wait state*, while other valid IP bus accesses incur 0 wait states.

Irrespective of the Response Select signal value, a Write access to the GPT Status Registers (Read-only registers GPT_ICR1, GPT_ICR2, GPT_CNT) will generate a bus exception.

- If the Response Select signal is driven Low, then the Read/Write access to the *unimplemented* address space of GPT (*ips_addr* is greater than or equal to $BASE + $028) will generate a bus exception.
- If the Response Select is driven High, then the Read/Write access to the unimplemented address space of GPT will *not* generate any error response (like a bus exception).

GPT memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302D_0000	GPT Control Register (GPT1_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
302D_0004	GPT Prescaler Register (GPT1_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
302D_0008	GPT Status Register (GPT1_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
302D_000C	GPT Interrupt Register (GPT1_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
302D_0010	GPT Output Compare Register 1 (GPT1_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
302D_0014	GPT Output Compare Register 2 (GPT1_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
302D_0018	GPT Output Compare Register 3 (GPT1_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
302D_001C	GPT Input Capture Register 1 (GPT1_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658
302D_0020	GPT Input Capture Register 2 (GPT1_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
302D_0024	GPT Counter Register (GPT1_CNT)	32	R	0000_0000h	12.1.6.10/ 3659
302E_0000	GPT Control Register (GPT2_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
302E_0004	GPT Prescaler Register (GPT2_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
302E_0008	GPT Status Register (GPT2_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
302E_000C	GPT Interrupt Register (GPT2_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
302E_0010	GPT Output Compare Register 1 (GPT2_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
302E_0014	GPT Output Compare Register 2 (GPT2_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
302E_0018	GPT Output Compare Register 3 (GPT2_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
302E_001C	GPT Input Capture Register 1 (GPT2_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658

Table continues on the next page...

GPT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
302E_0020	GPT Input Capture Register 2 (GPT2_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
302E_0024	GPT Counter Register (GPT2_CNT)	32	R	0000_0000h	12.1.6.10/ 3659
302F_0000	GPT Control Register (GPT3_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
302F_0004	GPT Prescaler Register (GPT3_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
302F_0008	GPT Status Register (GPT3_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
302F_000C	GPT Interrupt Register (GPT3_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
302F_0010	GPT Output Compare Register 1 (GPT3_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
302F_0014	GPT Output Compare Register 2 (GPT3_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
302F_0018	GPT Output Compare Register 3 (GPT3_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
302F_001C	GPT Input Capture Register 1 (GPT3_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658
302F_0020	GPT Input Capture Register 2 (GPT3_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
302F_0024	GPT Counter Register (GPT3_CNT)	32	R	0000_0000h	12.1.6.10/ 3659
306E_0000	GPT Control Register (GPT6_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
306E_0004	GPT Prescaler Register (GPT6_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
306E_0008	GPT Status Register (GPT6_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
306E_000C	GPT Interrupt Register (GPT6_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
306E_0010	GPT Output Compare Register 1 (GPT6_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
306E_0014	GPT Output Compare Register 2 (GPT6_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
306E_0018	GPT Output Compare Register 3 (GPT6_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
306E_001C	GPT Input Capture Register 1 (GPT6_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658
306E_0020	GPT Input Capture Register 2 (GPT6_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
306E_0024	GPT Counter Register (GPT6_CNT)	32	R	0000_0000h	12.1.6.10/ 3659

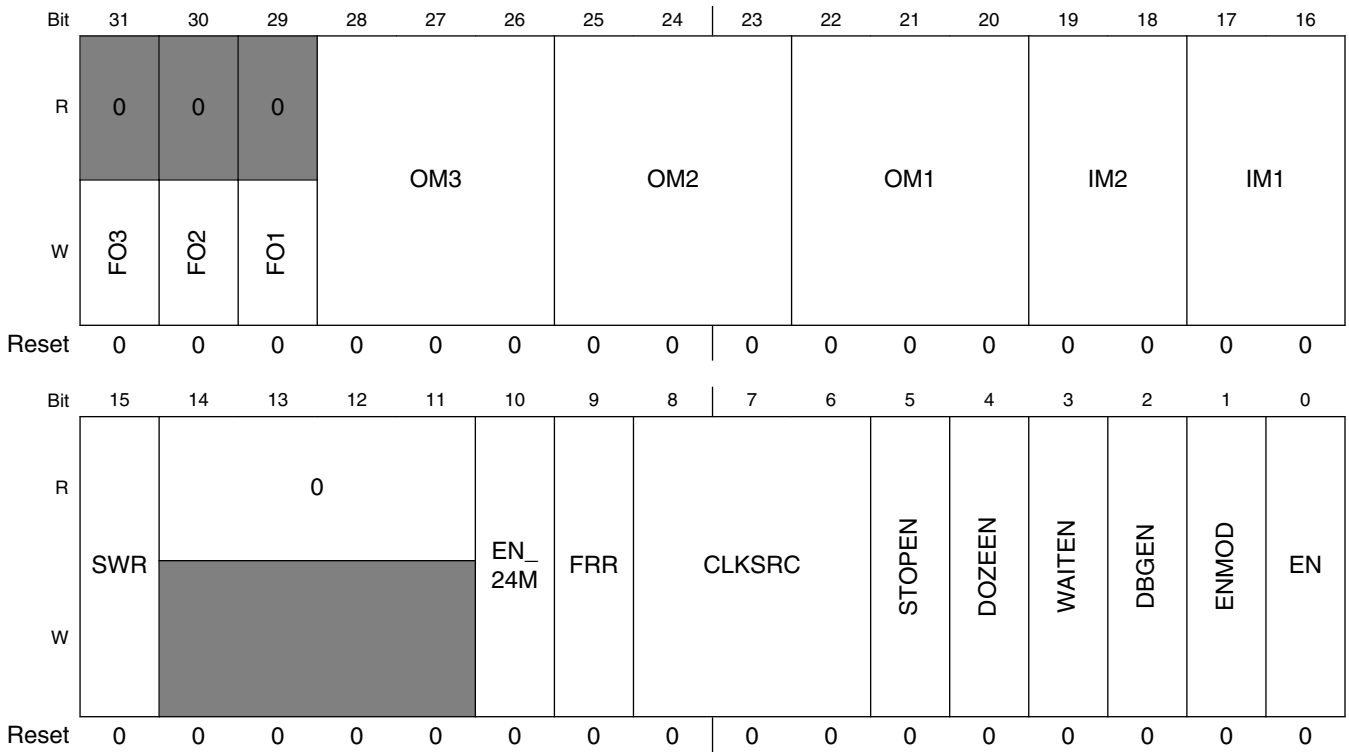
GPT memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
306F_0000	GPT Control Register (GPT5_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
306F_0004	GPT Prescaler Register (GPT5_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
306F_0008	GPT Status Register (GPT5_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
306F_000C	GPT Interrupt Register (GPT5_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
306F_0010	GPT Output Compare Register 1 (GPT5_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
306F_0014	GPT Output Compare Register 2 (GPT5_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
306F_0018	GPT Output Compare Register 3 (GPT5_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
306F_001C	GPT Input Capture Register 1 (GPT5_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658
306F_0020	GPT Input Capture Register 2 (GPT5_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
306F_0024	GPT Counter Register (GPT5_CNT)	32	R	0000_0000h	12.1.6.10/ 3659
3070_0000	GPT Control Register (GPT4_CR)	32	R/W	0000_0000h	12.1.6.1/ 3649
3070_0004	GPT Prescaler Register (GPT4_PR)	32	R/W	0000_0000h	12.1.6.2/ 3653
3070_0008	GPT Status Register (GPT4_SR)	32	R/W	0000_0000h	12.1.6.3/ 3654
3070_000C	GPT Interrupt Register (GPT4_IR)	32	R/W	0000_0000h	12.1.6.4/ 3655
3070_0010	GPT Output Compare Register 1 (GPT4_OCR1)	32	R/W	FFFF_FFFFh	12.1.6.5/ 3656
3070_0014	GPT Output Compare Register 2 (GPT4_OCR2)	32	R/W	FFFF_FFFFh	12.1.6.6/ 3657
3070_0018	GPT Output Compare Register 3 (GPT4_OCR3)	32	R/W	FFFF_FFFFh	12.1.6.7/ 3657
3070_001C	GPT Input Capture Register 1 (GPT4_ICR1)	32	R	0000_0000h	12.1.6.8/ 3658
3070_0020	GPT Input Capture Register 2 (GPT4_ICR2)	32	R	0000_0000h	12.1.6.9/ 3658
3070_0024	GPT Counter Register (GPT4_CNT)	32	R	0000_0000h	12.1.6.10/ 3659

12.1.6.1 GPT Control Register (GPTx_CR)

The GPT Control Register (GPT_CR) is used to program and configure GPT operations. An IP Bus Write to the GPT Control Register occurs after one cycle of wait state, while an IP Bus Read occurs after 0 wait states.

Address: Base address + 0h offset



GPTx_CR field descriptions

Field	Description
31 FO3	FO3 Force Output Compare Channel 3 FO2 Force Output Compare Channel 2 FO1 Force Output Compare Channel 1 The FO*n* bit causes the pin action *programmed* for the timer Output Compare *n* pin (according to the OM*n* bits in this register). • The OF*n* flag (OF3, OF2, OF1) in the status register is **not affected**. • This bit is self-negating and always read as zero. 0 Writing a 0 has no effect. 1 Causes the programmed pin action on the timer Output Compare *n* pin; the OF*n* flag is not set.
30 FO2	See F03

Table continues on the next page...

GPTx_CR field descriptions (continued)

Field	Description
29 FO1	See F03
28–26 OM3	OM3 (bits 28-26) controls the Output Compare Channel 3 operating mode. OM2 (bits 25-23) controls the Output Compare Channel 2 operating mode. OM1 (bits 22-20) controls the Output Compare Channel 1 operating mode. The OM*n* bits specify the response that a compare event will generate on the output pin of Output Compare Channel *n*. • The toggle, clear, and set options cause a change on the output pin *only* if a compare event occurs. • When OM*n* is programmed as 1xx (active low pulse), the output pin is set to one immediately on the next input clock; a low pulse (that is an input clock in width) occurs when there is a compare event. Note that here, "input clock" refers to the clock selected by the CLKSRC bits of the GPT Control Register. 000 Output disconnected. No response on pin. 001 Toggle output pin 010 Clear output pin 011 Set output pin 1xx Generate an active low pulse (that is one input clock wide) on the output pin.
25–23 OM2	See OM3
22–20 OM1	See OM3
19–18 IM2	IM2 (bits 19-18, Input Capture Channel 2 operating mode) IM1 (bits 17-16, Input Capture Channel 1 operating mode) The IM*n* bit field determines the transition on the input pin (for Input capture channel *n*), which will trigger a capture event. 00 capture disabled 01 capture on rising edge only 10 capture on falling edge only 11 capture on both edges
17–16 IM1	See IM2
15 SWR	Software reset. This is the software reset of the GPT module. It is a self-clearing bit. • The SWR bit is set when the module is in reset state. • The SWR bit is cleared when the reset procedure finishes. • Setting the SWR bit resets **all of the registers** to their default reset values, except for the EN, ENMOD, STOPEN, DOZEEN, WAITEN, and DBGEN bits in the GPT Control Register (this control register). 0 GPT is not in reset state 1 GPT is in reset state
14–11 Reserved	This read-only field is reserved and always has the value 0.
10 EN_24M	Enable 24 MHz clock input from crystal.

Table continues on the next page...

GPTx_CR field descriptions (continued)

Field	Description
	• A hardware reset resets the EN_24M bit. • A software reset *does not affect* the EN_24M bit. 0 24M clock disabled 1 24M clock enabled
9 FRR	Free-Run or Restart mode. The FFR bit determines the behavior of the GPT when a compare event in channel 1 occurs. • In Restart mode, after a compare event, the counter resets to 0x00000000 and resumes counting (after the occurrence of a compare event). • In Free-Run mode, after a compare event, the counter continues counting until 0xFFFFFFFF and then rolls over to 0. 0 Restart mode 1 Free-Run mode
8–6 CLKSRC	Clock Source select. The CLKSRC bits select which clock will go to the prescaler (and subsequently be used to run the GPT counter). • The CLKSRC bit field value should only be changed after disabling the GPT by clearing the EN bit in this register (GPT_CR). • A software reset does not affect the CLKSRC bit. 000 No clock 001 Peripheral Clock (ipg_clk) 010 High Frequency Reference Clock (ipg_clk_highfreq) 011 External Clock 100 Low Frequency Reference Clock (ipg_clk_32k) 101 Crystal oscillator as Reference Clock (ipg_clk_24M) others Reserved
5 STOPEN	GPT Stop Mode enable. The STOPEN read/write control bit enables GPT operation *during Stop mode*. • A hardware reset resets the STOPEN bit. • A software reset *does not affect* the STOPEN bit. 0 GPT is disabled in Stop mode. 1 GPT is enabled in Stop mode.
4 DOZEEN	GPT Doze Mode Enable. • A hardware reset resets the DOZEEN bit. • A software reset *does not affect* the DOZEEN bit. 0 GPT is disabled in doze mode. 1 GPT is enabled in doze mode.
3 WAITEN	GPT Wait Mode enable. The WAITEN read/write control bit enables GPT operation *during Wait mode*. • A hardware reset resets the WAITEN bit. • A software reset *does not affect* the WAITEN bit. 0 GPT is disabled in wait mode. 1 GPT is enabled in wait mode.

Table continues on the next page...

GPTx_CR field descriptions (continued)

Field	Description
2 DBGEN	GPT debug mode enable. The DBGEN read/write control bit enables GPT operation *during Debug mode*. • A hardware reset resets the DBGEN bit. • A software reset *does not affect* the DBGEN bit. 0 GPT is disabled in debug mode. 1 GPT is enabled in debug mode.
1 ENMOD	GPT Enable mode. When the GPT is disabled (EN=0), then both the Main Counter and Prescaler Counter *freeze their current count values*. The ENMOD bit determines the value of the GPT counter when Counter is enabled again (if the EN bit is set). • If the ENMOD bit is 1, then the Main Counter and Prescaler Counter values are reset to 0 after GPT is enabled (EN=1). • If the ENMOD bit is 0, then the Main Counter and Prescaler Counter restart counting *from their frozen values* after GPT is enabled (EN=1). • If GPT is programmed to be disabled in a low power mode (STOP/WAIT), then the Main Counter and Prescaler Counter *freeze at their current count values* when the GPT enters low power mode. • When GPT exits low power mode, the Main Counter and Prescaler Counter start counting from their frozen values, regardless of the ENMOD bit value. • Setting the SWR bit will clear the Main Counter and Prescaler Counter values, regardless of the value of EN or ENMOD bits. • A hardware reset resets the ENMOD bit. • A software reset *does not affect* the ENMOD bit. 0 GPT counter will retain its value when it is disabled. 1 GPT counter value is reset to 0 when it is disabled.
0 EN	GPT Enable. The EN bit is the GPT module enable bit. **Before setting the EN bit**, we recommend that *all registers be properly programmed*. • A hardware reset resets the EN bit. • A software reset *does not affect* the EN bit. 0 GPT is disabled. 1 GPT is enabled.

12.1.6.2 GPT Prescaler Register (GPTx_PR)

The GPT Prescaler Register (GPT_PR) contains bits that determine the *divide value* of the clock that runs the counter.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PRESCALER24M				PRESCALER											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTx_PR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–12 PRESCALER24M	Prescaler bits. 24M crystal clock is divided by [PRESCALER24M + 1] before selected by the CLKSRC field. If 24M crystal clock is not selected, this feild takes no effect. 0x0 Divide by 1 0x1 Divide by 2 ... ... 0xF Divide by 16
PRESCALER	Prescaler bits. The clock selected by the CLKSRC field is divided by [PRESCALER + 1], and then used to run the counter. • A change in the value of the PRESCALER bits cause the Prescaler counter to reset and a new count period to start immediately. • See Figure 12-3 for the timing diagram. 0x000 Divide by 1 0x001 Divide by 2 ... ... 0xFFF Divide by 4096

12.1.6.3 GPT Status Register (GPTx_SR)

The GPT Status Register (GPT_SR) contains bits that indicate that a counter has rolled over, and if any event has occurred on the Input Capture and Output Compare channels. The bits are cleared by writing a 1 to them.

Address: Base address + 8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										ROV	IF2	IF1	OF3	OF2	OF1
W											w1c	w1c	w1c	w1c	w1c	w1c
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTx_SR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 ROV	Rollover Flag. The ROV bit indicates that the counter has reached its *maximum possible value* and *rolled over* to 0 (from which the counter continues counting). The ROV bit is only set if the counter has reached 0xFFFFFFFF in both Restart and Free-Run modes. 0 Rollover has not occurred. 1 Rollover has occurred.
4 IF2	IF2 Input capture 2 Flag IF1 Input capture 1 Flag The IF*n* bit indicates that a capture event has occurred on Input Capture channel *n*. 0 Capture event has not occurred. 1 Capture event has occurred.
3 IF1	See IF2
2 OF3	OF3 Output Compare 3 Flag OF2 Output Compare 2 Flag OF1 Output Compare 1 Flag The OF*n* bit indicates that a compare event has occurred on Output Compare channel *n*. 0 Compare event has not occurred. 1 Compare event has occurred.

Table continues on the next page...

GPTx_SR field descriptions (continued)

Field	Description
1 OF2	See OF3
0 OF1	See OF3

12.1.6.4 GPT Interrupt Register (GPTx_IR)

The GPT Interrupt Register (GPT_IR) contains bits that control whether interrupts are generated after rollover, input capture and output compare events.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0										ROVIE	IF2IE	IF1IE	OF3IE	OF2IE	OF1IE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTx_IR field descriptions

Field	Description
31–6 Reserved	This read-only field is reserved and always has the value 0.
5 ROVIE	Rollover Interrupt Enable. The ROVIE bit controls the Rollover interrupt. 0 Rollover interrupt is disabled. 1 Rollover interrupt enabled.
4 IF2IE	IF2IE Input capture 2 Interrupt Enable IF1IE Input capture 1 Interrupt Enable The IF*n*IE bit controls the IF*n*IE Input Capture *n* Interrupt Enable. 0 IF2IE Input Capture *n* Interrupt Enable is disabled. 1 IF2IE Input Capture *n* Interrupt Enable is enabled.

Table continues on the next page...

GPTx_IR field descriptions (continued)

Field	Description
3 IF1IE	See IF2IE
2 OF3IE	OF3IE Output Compare 3 Interrupt Enable OF2IE Output Compare 2 Interrupt Enable OF1IE Output Compare 1 Interrupt Enable The OF*n*IE bit controls the Output Compare Channel *n* interrupt. 0 Output Compare Channel *n* interrupt is disabled. 1 Output Compare Channel *n* interrupt is enabled.
1 OF2IE	See OF3IE
0 OF1IE	See OF3IE

12.1.6.5 GPT Output Compare Register 1 (GPTx_OCR1)

The GPT Compare Register 1 (GPT_OCR1) holds the value that determines when a compare event will be generated on Output Compare Channel 1. Any write access to the Compare register of Channel 1 while in Restart mode (FRR=0) will reset the GPT counter.

An IP Bus Write access to the GPT Output Compare Register1 (GPT_OCR1) occurs *after* one cycle of wait state; an IP Bus Read access occurs *immediately* (0 wait states).

Address: Base address + 10h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GPTx_OCR1 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 1.

12.1.6.6 GPT Output Compare Register 2 (GPTx_OCR2)

The GPT Compare Register 2 (GPT_OCR2) holds the value that determines when a compare event will be generated on Output Compare Channel 2.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

GPTx_OCR2 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 2.

12.1.6.7 GPT Output Compare Register 3 (GPTx_OCR3)

The GPT Compare Register 3 (GPT_OCR3) holds the value that determines when a compare event will be generated on Output Compare Channel 3.

Address: Base address + 18h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	COMP																															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

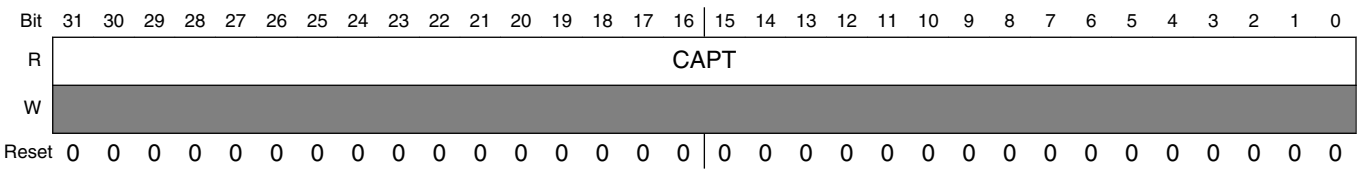
GPTx_OCR3 field descriptions

Field	Description
COMP	Compare Value. When the counter value equals the COMP bit field value, a compare event is generated on Output Compare Channel 3.

12.1.6.8 GPT Input Capture Register 1 (GPTx_ICR1)

The GPT Input Capture Register 1 (GPT_ICR1) is a read-only register that holds the value *that was in the counter during the last capture event* on Input Capture Channel 1.

Address: Base address + 1Ch offset



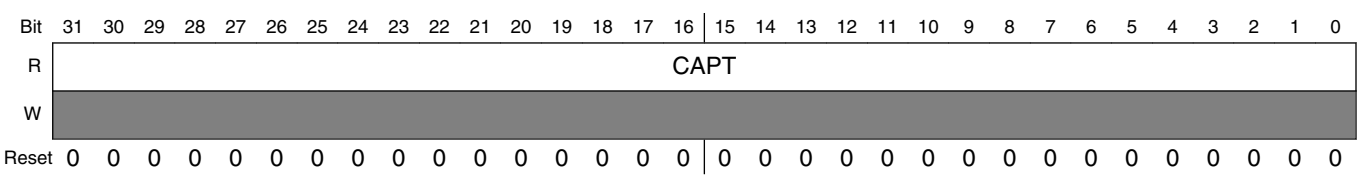
GPTx_ICR1 field descriptions

Field	Description
CAPT	Capture Value. After a capture event on Input Capture Channel 1 occurs, the current value of the counter is loaded into GPT Input Capture Register 1.

12.1.6.9 GPT Input Capture Register 2 (GPTx_ICR2)

The GPT Input capture Register 2 (GPT_ICR2) is a read-only register which holds the value that was in the counter during the last capture event on input capture channel 2.

Address: Base address + 20h offset



GPTx_ICR2 field descriptions

Field	Description
CAPT	Capture Value. After a capture event on Input Capture Channel 2 occurs, the current value of the counter is loaded into GPT Input Capture Register 2.

12.1.6.10 GPT Counter Register (GPTx_CNT)

The GPT Counter Register (GPT_CNT) is the main counter's register. GPT_CNT is a read-only register and can be read *without affecting the counting process* of the GPT.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	COUNT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

GPTx_CNT field descriptions

Field	Description
COUNT	Counter Value. The COUNT bits show the current count value of the GPT counter.

12.2 Pulse Width Modulation (PWM)

12.2.1 Overview

The Pulse Width Modulation (PWM) has a 16-bit counter, and is optimized to generate sound from stored sample audio images and it can also generate tones. It uses 16-bit resolution and a 4 x 16 data FIFO.

This section presents an overview of the PWM. A block diagram of the PWM module is shown in the figure below.

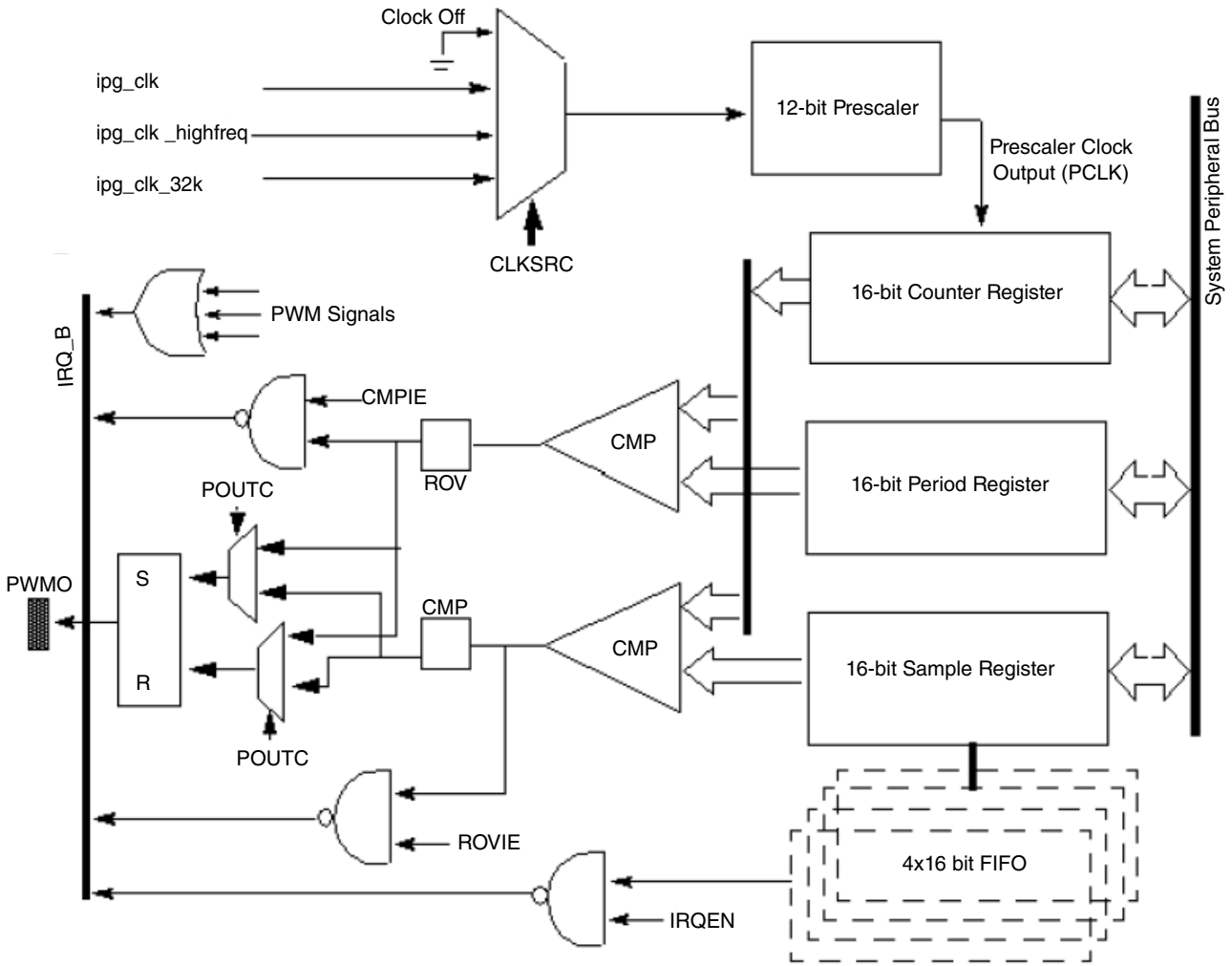


Figure 12-6. Pulse-Width Modulator Block Diagram

The following features characterize the PWM:

- 16-bit up-counter with clock source selection
- 4 x 16 FIFO to minimize interrupt overhead
- 12-bit prescaler for division of clock
- Sound and melody generation
- Active high or active low configured output
- Can be programmed to be active in low-power mode
- Can be programmed to be active in debug mode
- Interrupts at compare and rollover

12.2.2 External Signals

The PWM follows IP Bus protocol when interfacing with the processor core. PWM does not have any interface signals with any other block inside the chip except for clock and reset inputs from the Clock Control Module (CCM), System Reset Controller (SRC), and interrupt signals to the processor interrupt handler. There is a single output signal.

12.2.3 Clocks

The table found here describes the clock sources for PWM. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 12-2. PWM Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_32k	ckil_sync_clk_root	low-frequency reference clock (32 kHz)
ipg_clk_highfreq	perclk_clk_root	high-frequency reference clock
ipg_clk_s	ipg_clk_root	Peripheral access clock

The clock that feeds the prescaler can be selected from:

- High-frequency reference clock (ipg_clk_highfreq) pat_ref or CKIH

  This is a high frequency clock, provided by the Clock Control Module (CCM). This clock should be on in the low power mode when the ipg_clk is turned off. Thus, the PWM can be run on this clock in the low power mode. The CCM is expected to provide this clock after synchronizing it to ahb_clk in the normal functional mode and then switch to the unsynchronized version in the low power mode.

- Low-frequency reference clock (ipg_clk_32k, CKIL)

  This is the 32 KHz low reference clock which is provided by the CCM. This clock should be on in the low power mode when ipg_clk is turned off. Thus, PWM can be run on this clock in the low power mode. The CCM is expected to provide this clock after synchronizing it to ahb_clk in the normal functional mode and then switch to the unsynchronized version in the low power mode.

- Peripheral clock (ipg_clk)

  This clock should be on in normal operations. In low power mode, it can be switched off.

- Peripheral access clock (ipg_clk_s)

  This clock is used for register read/write.

The clock input source is determined by the PWM control register field PWM_CR[CLKSRC]. The CLKSRC value should only be changed when the PWM is disabled.

A change in the value of the PRESCALER field of the control register is immediately reflected on its output clock frequency.

12.2.4 Functional Description

The following sections detail the PWM operation and function.

12.2.4.1 Operation

The output of the PWM is a toggling signal whose frequency and duty cycle can be modulated by programming the appropriate registers. It has a 16-bit up counter which counts from 0x0000 until the counter value equals the PWM_PR + 1. After this match occurs the counter is reset to 0x0000.

At the beginning of a count period cycle, the PWMO pin is set to one (default) and the counter begins counting up from 0x0000. The sample value in the sample FIFO is compared on each count of prescaler clock. When the sample and count values match, the PWMO signal is cleared to zero (default). The counter continues counting until the period match occurs and subsequently another period cycle begins.

When the PWM is enabled, the counter starts running and generates an output with the reset values in the period and sample registers. It is recommended that the programming of these registers be done before PWM is enabled.

A hardware reset results in all the PWM count and sample registers being cleared and the FIFO being flushed. The control register shows that FIFO is empty and it can be written into, and the PWM is disabled. A software reset has the same results, however the state of the DBGEN, STOPEN, DOZEN, and WAITEN bits in the control register are not affected. Software reset can be asserted even when the PWM is in disabled state.

12.2.4.1.1 FIFO

Digital sample values can be loaded into the pulse-width modulator as 16-bit words. The endianess can be changed using the BCTR and HCTR bits of the control register. A 4-word (16-bit) FIFO minimizes interrupt overhead. A maskable interrupt is generated when the number of data words fall below the water level set by the FWM field in the control register.

A write to the PWM_SAR sample register results in the value being stored into the FIFO if it is not full. A write when the FIFO is full sets FWE (FIFO write error) bit in the status register and the FIFO contents remain unchanged. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM_SR[FIFOAV] field shows how many data words are currently contained in the FIFO and whether or not it can be written into.

A read on the sample register yields the current FIFO value that is being used, or will be used, by the PWM for generation on the output signal. Therefore, a write and a subsequent read on the sample register may result in different values being obtained.

12.2.4.1.2 Rollover and Compare Event

The counter is reset to 0x0000 after its value equals the PWM_PR[PERIOD] + 1 and resumes counting thereafter. This event is referred to as a rollover. For example, if PWM_PR[PERIOD] = 0x0000, the counter is reset when it equals 0x0001. When PWM_PR[PERIOD] = 0xFFFF or 0xFFFE, the counter is reset when it equals 0xFFFF. For more information, see the PWM Period Register (PWM_PR) description.

During a rollover event the output is either set (default), reset or has no effect according to the programming of the POUTC field in the control register. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

When the counter value reaches the sample value, the output of the PWM is reset (default), set or has no effect according to the programming of the POUTC field of control register. This event is referred to as a compare event. This event can also generate an interrupt if the respective interrupt enable bit is set in the control register.

If the rollover event sets the PWM output signal, the compare event will reset it and vice versa for a particular programming configuration of POUTC field.

12.2.4.1.3 Low Power Mode Behavior

In low power mode, if the clock from the selected clock source is available, the PWM counter continues to run and an output is produced, depending on whether the control bit for that mode is set or not. In the absence of the clock itself, or if the corresponding low power bit in the control register is 0, the counter is reset and resumes counting when it exits the low power mode.

12.2.4.1.4 Debug Mode Behavior

In debug mode, PWM has the option of continuing to run or be halted. If the DBGEN bit is not set in the PWM_PWMCR, the PWM is halted. If the DBGEN bit is set, then the PWM will continue to run in the debug mode.

12.2.5 Enable Sequence for the PWM

The sequence found here should be used to enable the PWM.

1. Configure the desired settings for the PWM Control Register (PWMx_PWMCR) while keeping the PWM disabled (PWMx_PWMCR[0]=0).
2. Enable the desired interrupts in the PWM Interrupt Register (PWMx_PWMIR).
3. One to three initial samples may be written to the PWM Sample Register (PWMx_PWMSAR). The initial sample values will be loaded into the PWM FIFO even if the PWM is not yet enabled. Do not write a 4th sample because the FIFO will become full and trigger a FIFO Write Error (FWE). This error will prevent the PWM from starting once it is enabled.
4. Check the FIFO Write Error status bit (FWE), the Compare status bit (CMP) and the Roll-over status bit (ROV) in the PWM Status Register (PWMx_PWMSR) to make sure they are all zero. Any non-zero status bits should be cleared by writing a 1 to them.
5. Write the desired period to the PWM Period Register (PWMx_PWMPR).
6. Enable the PWM by writing a 1 to the PWM Enable bit, PWMx_PWMCR[0], while maintaining the other register bits in their previously configured state.

12.2.6 Disable Sequence for the PWM

The PWM can be disabled at any time by clearing the PWM enable bit, PWMx_PWMCR[0] to 0.

Any data remaining in the FIFO will not be produced at the PWM output after the PWM has been disabled and will remain in the FIFO until the PWM is enabled again. A software reset (setting PWMx_PWMCR[3] to 1) or a hardware reset will clear the FIFO and any remaining data will be lost.

12.2.7 PWM Memory Map/Register Definition

The PWM includes six user-accessible 32-bit registers.

PWM memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3066_0000	PWM Control Register (PWM1_PWMCR)	32	R/W	0000_0000h	12.2.7.1/ 3666
3066_0004	PWM Status Register (PWM1_PWMSR)	32	w1c	0000_0008h	12.2.7.2/ 3668
3066_0008	PWM Interrupt Register (PWM1_PWMIR)	32	R/W	0000_0000h	12.2.7.3/ 3670
3066_000C	PWM Sample Register (PWM1_PWMSAR)	32	R/W	0000_0000h	12.2.7.4/ 3670
3066_0010	PWM Period Register (PWM1_PWMPR)	32	R/W	0000_FFFEh	12.2.7.5/ 3671
3066_0014	PWM Counter Register (PWM1_PWMCNR)	32	R	0000_0000h	12.2.7.6/ 3672
3067_0000	PWM Control Register (PWM2_PWMCR)	32	R/W	0000_0000h	12.2.7.1/ 3666
3067_0004	PWM Status Register (PWM2_PWMSR)	32	w1c	0000_0008h	12.2.7.2/ 3668
3067_0008	PWM Interrupt Register (PWM2_PWMIR)	32	R/W	0000_0000h	12.2.7.3/ 3670
3067_000C	PWM Sample Register (PWM2_PWMSAR)	32	R/W	0000_0000h	12.2.7.4/ 3670
3067_0010	PWM Period Register (PWM2_PWMPR)	32	R/W	0000_FFFEh	12.2.7.5/ 3671
3067_0014	PWM Counter Register (PWM2_PWMCNR)	32	R	0000_0000h	12.2.7.6/ 3672
3068_0000	PWM Control Register (PWM3_PWMCR)	32	R/W	0000_0000h	12.2.7.1/ 3666
3068_0004	PWM Status Register (PWM3_PWMSR)	32	w1c	0000_0008h	12.2.7.2/ 3668
3068_0008	PWM Interrupt Register (PWM3_PWMIR)	32	R/W	0000_0000h	12.2.7.3/ 3670
3068_000C	PWM Sample Register (PWM3_PWMSAR)	32	R/W	0000_0000h	12.2.7.4/ 3670
3068_0010	PWM Period Register (PWM3_PWMPR)	32	R/W	0000_FFFEh	12.2.7.5/ 3671
3068_0014	PWM Counter Register (PWM3_PWMCNR)	32	R	0000_0000h	12.2.7.6/ 3672
3069_0000	PWM Control Register (PWM4_PWMCR)	32	R/W	0000_0000h	12.2.7.1/ 3666
3069_0004	PWM Status Register (PWM4_PWMSR)	32	w1c	0000_0008h	12.2.7.2/ 3668
3069_0008	PWM Interrupt Register (PWM4_PWMIR)	32	R/W	0000_0000h	12.2.7.3/ 3670
3069_000C	PWM Sample Register (PWM4_PWMSAR)	32	R/W	0000_0000h	12.2.7.4/ 3670

Table continues on the next page...

PWM memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3069_0010	PWM Period Register (PWM4_PWMPR)	32	R/W	0000_FFFEh	12.2.7.5/ 3671
3069_0014	PWM Counter Register (PWM4_PWMCNR)	32	R	0000_0000h	12.2.7.6/ 3672

12.2.7.1 PWM Control Register (PWMx_PWMCR)

The PWM control register (PWM_PWMCR) is used to configure the operating settings of the PWM. It contains the prescaler for the clock division.

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0				FWM		STOPEN	DOZEN	WAITEN	DBGEN	BCTR	HCTR	POUTC		CLKSRC	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	PRESCALER												SWR	REPEAT		EN
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCR field descriptions

Field	Description
31–28 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
27–26 FWM	FIFO Water Mark. These bits are used to set the data level at which the FIFO empty flag will be set and the corresponding interrupt generated 00 FIFO empty flag is set when there are more than or equal to 1 empty slots in FIFO 01 FIFO empty flag is set when there are more than or equal to 2 empty slots in FIFO 10 FIFO empty flag is set when there are more than or equal to 3 empty slots in FIFO 11 FIFO empty flag is set when there are more than or equal to 4 empty slots in FIFO
25 STOPEN	Stop Mode Enable. This bit keeps the PWM functional while in stop mode. When this bit is cleared, the input clock is gated off in stop mode. This bit is not affected by software reset. It is cleared by hardware reset.

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
	0 Inactive in stop mode 1 Active in stop mode
24 DOZEN	Doze Mode Enable. This bit keeps the PWM functional in doze mode. When this bit is cleared, the input clock is gated off in doze mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in doze mode 1 Active in doze mode
23 WAITEN	Wait Mode Enable. This bit keeps the PWM functional in wait mode. When this bit is cleared, the input clock is gated off in wait mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in wait mode 1 Active in wait mode
22 DBGEN	Debug Mode Enable. This bit keeps the PWM functional in debug mode. When this bit is cleared, the input clock is gated off in debug mode. This bit is not affected by software reset. It is cleared by hardware reset. 0 Inactive in debug mode 1 Active in debug mode
21 BCTR	Byte Data Swap Control. This bit determines the byte ordering of the 16-bit data when it goes into the FIFO from the sample register. 0 byte ordering remains the same 1 byte ordering is reversed
20 HCTR	Half-word Data Swap Control. This bit determines which half word data from the 32-bit IP Bus interface is written into the lower 16 bits of the sample register. 0 Half word swapping does not take place 1 Half words from write data bus are swapped
19–18 POUTC	PWM Output Configuration. This bit field determines the mode of PWM output on the output pin. 00 Output pin is set at rollover and cleared at comparison 01 Output pin is cleared at rollover and set at comparison 10 PWM output is disconnected 11 PWM output is disconnected
17–16 CLKSRC	Select Clock Source. These bits determine which clock input will be selected for running the counter. After reset the system functional clock is selected. The input clock can also be turned off if these bits are set to 00. This field value should only be changed when the PWM is disabled 00 Clock is off 01 ipg_clk 10 ipg_clk_highfreq 11 ipg_clk_32k
15–4 PRESCALER	Counter Clock Prescaler Value. This bit field determines the value by which the clock will be divided before it goes to the counter. 0x000 Divide by 1 0x001 Divide by 2 0xfff Divide by 4096
3 SWR	Software Reset. PWM is reset when this bit is set to 1. It is a self clearing bit. A write 1 to this bit is a single wait state write cycle. When the block is in reset state this bit is set and is cleared when the reset

Table continues on the next page...

PWMx_PWMCR field descriptions (continued)

Field	Description
	procedure is over. Setting this bit resets all the registers to their reset values except for the DBGEN, STOPEN, DOZEN, and WAITEN bits in this control register. 0 PWM is out of reset 1 PWM is undergoing reset
2–1 REPEAT	Sample Repeat. This bit field determines the number of times each sample from the FIFO is to be used. 00 Use each sample once 01 Use each sample twice 10 Use each sample four times 11 Use each sample eight times
0 EN	PWM Enable. This bit enables the PWM. If this bit is not enabled, the clock prescaler and the counter is reset. When the PWM is enabled, it begins a new period, the output pin is set to start a new period while the prescaler and counter are released and counting begins. To make the PWM work with softreset and disable/enable, users can do software reset by seting the SWR bit, wait software reset done, configure the registers, and then enable the PWM by setting this bit to "1" Users can also disable/enable the PWM if PWM would like to be stopped and resumed with same registers configurations . 0 PWM disabled 1 PWM enabled

12.2.7.2 PWM Status Register (PWMx_PWMSR)

The PWM status register (PWM_PWMSR) contains seven bits which display the state of the FIFO and the occurrence of rollover and compare events. The FIFOAV bit is read-only but the other four bits can be cleared by writing 1 to them. The FE, ROV, and CMP bits are associated with FIFO-Empty, Roll-over, and Compare interrupts, respectively.

Address: Base address + 4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0									FWE	CMP	ROV	FE	FIFOAV		
W										w1c	w1c	w1c	w1c			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0

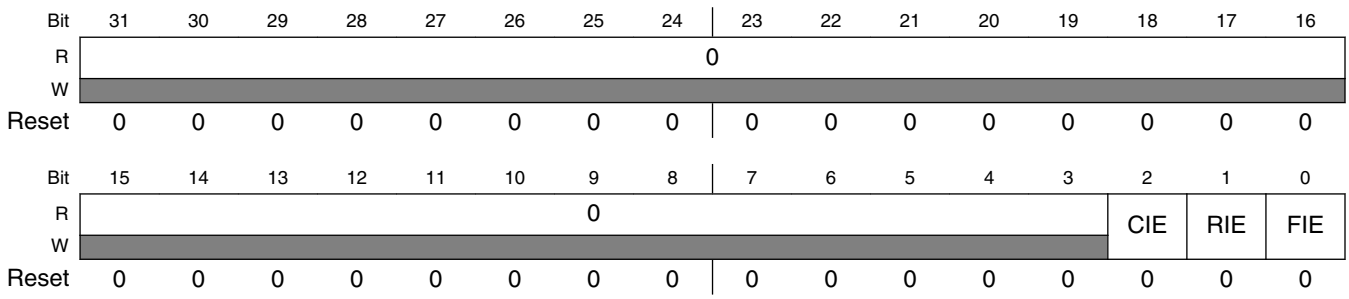
PWMx_PWMSR field descriptions

Field	Description
31–7 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
6 FWE	FIFO Write Error Status. This bit shows that an attempt has been made to write FIFO when it is full. 0 FIFO write error not occurred 1 FIFO write error occurred
5 CMP	Compare Status. This bit shows that a compare event has occurred. 0 Compare event not occurred 1 Compare event occurred
4 ROV	Roll-over Status. This bit shows that a roll-over event has occurred. 0 Roll-over event not occurred 1 Roll-over event occurred
3 FE	FIFO Empty Status Bit. This bit indicates the FIFO data level in comparison to the water level set by FWM field in the control register. 0 Data level is above water mark 1 When the data level falls below the mark set by FWM field
FIFOAV	FIFO Available. These read-only bits indicate the data level remaining in the FIFO. An attempted write to these bits will not affect their value and no transfer error is generated. 000 No data available 001 1 word of data in FIFO 010 2 words of data in FIFO 011 3 words of data in FIFO 100 4 words of data in FIFO 101 unused 110 unused 111 unused

12.2.7.3 PWM Interrupt Register (PWMx_PWMIR)

The PWM Interrupt register (PWM_PWMIR) contains three bits which control the generation of the compare, rollover and FIFO empty interrupts.

Address: Base address + 8h offset



PWMx_PWMIR field descriptions

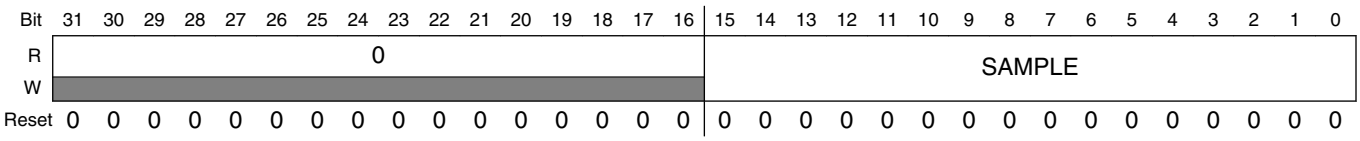
Field	Description
31–3 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
2 CIE	Compare Interrupt Enable. This bit controls the generation of the Compare interrupt. 0 Compare Interrupt not enabled 1 Compare Interrupt enabled
1 RIE	Roll-over Interrupt Enable. This bit controls the generation of the Rollover interrupt. 0 Roll-over interrupt not enabled 1 Roll-over Interrupt enabled
0 FIE	FIFO Empty Interrupt Enable. This bit controls the generation of the FIFO Empty interrupt. 0 FIFO Empty interrupt disabled 1 FIFO Empty interrupt enabled

12.2.7.4 PWM Sample Register (PWMx_PWMSAR)

The PWM sample register (PWM_PWMSAR) is the input to the FIFO. 16-bit words are loaded into the FIFO. The FIFO can be written at any time, but can be read only when the PWM is enabled. The PWM will run at the last set duty-cycle setting if all the values of the FIFO has been utilized, until the FIFO is reloaded or the PWM is disabled. When a new value is written, the duty cycle changes after the current period is over.

A value of zero in the sample register will result in the PWMO output signal always being low/high (POUTC = 00 it will be low and POUTC = 01 it will be high), and no output waveform will be produced. If the value in this register is higher than the PERIOD + 1, the output will never be set/reset depending on POUTC value.

Address: Base address + Ch offset



PWM*x*_PWMSAR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
SAMPLE	Sample Value. This is the input to the 4x16 FIFO. The value in this register denotes the value of the sample being currently used.

12.2.7.5 PWM Period Register (PWM*x*_PWMPR)

The PWM period register (PWM_PWMPR) determines the period of the PWM output signal. After the counter value matches PERIOD + 1, the counter is reset to start another period.

PWMO (Hz) = PCLK(Hz) / (period +2)

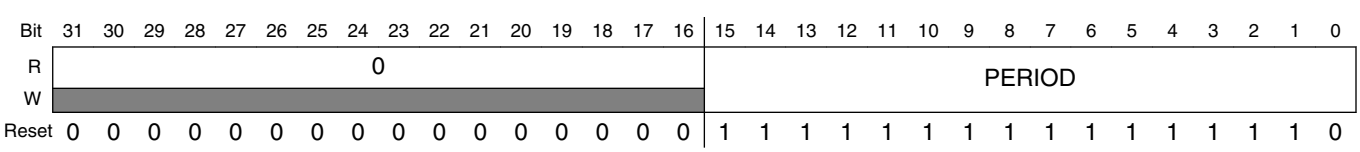
A value of zero in the PWM_PWMPR will result in a period of two clock cycles for the output signal. Writing 0xFFFF to this register will achieve the same result as writing 0xFFFE.

A change in the period value due to a write in PWM_PWMPR results in the counter being reset to zero and the start of a new count period.

NOTE

Settings PWM_PWMPR to 0xFFFF when PWMx_PWMCR REPEAT bits are set to non-zero values is not allowed.

Address: Base address + 10h offset



PWMx_PWMPR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
PERIOD	Period Value. These bits determine the Period of the count cycle. The counter counts up to [Period Value] +1 and is then reset to 0x0000.

12.2.7.6 PWM Counter Register (PWMx_PWMCNR)

The read-only pulse-width modulator counter register (PWM_PWMCNR) contains the current count value and can be read at any time without disturbing the counter.

Address: Base address + 14h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																COUNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

PWMx_PWMCNR field descriptions

Field	Description
31–16 Reserved	This field is reserved. This read-only field is reserved and always has the value 0.
COUNT	Counter Value. These bits are the counter register value and denotes the current count state the counter register is in.

Chapter 13
Multimedia

13.1 Multimedia Overview

13.1.1 Multimedia Overview

13.1.1.1 Display Interface

Display Interface consists of the following:

- Enhanced LCD Interface (eLCDIF)
  - Support 8-bit / 16-bit / 18-bit / 24-bit / 32-bit pixel depth
  - Support DOTCLK mode for MIPI-DPI interface
  - Support MPU mode for MIPI-DBI interface
  - Support resolution up to 1920x1080p60 and 1800x1200p60
  - Support one base layer and one graphics overlay with alpha blending
- Display Controller (DCSS) - provides a mechanism to display frame buffers in memory out to UltraHD or HDTVs. It has the capability to combine up to 3 layers of graphics or video overlay to the HDMI output. The key features of the display controller includes:
  - Supports up to 3 layers of graphics or video
    - Arbitrary offset
    - One plane can be graphics with 8 bit alpha support
    - Upscale 1920x1080p60 video or graphics to 3840x2160p60
    - Downscale 3840x2160p30 video to 1920x1080p30 or 1280x720p30
  - HDR support:
    - HDR10 with 2084 and 2020
    - Dolby Vision single and dual layer formats
    - HLG
  - HDMI 2.0a supporting one display:
    - Resolutions of: 640x480p60, 720x480p60, 1280x720p60, 1920x1080p60, 3840x2160p60, 4096x2160p60

        - HDCP 2.2 and HDCP 1.4
        - Pixel clock up to 596 MHz
    - Audio support:
        - 32 channel audio output support
        - 1 S/PDIF audio ARC input support
    - Output can also go to MIPI DSI output
    - Frame Buffer Compression – Lossless compression of buffers
- HD Display Transmitter Controller (HDMI TX) - The HDMI firmware and HDCP keys are loaded by the boot core running ROM code. Once they are loaded, they need to be retained in the HDMI logic until next chip power-on reset. More information about the HDCP key handling can be found in the system boot chapter.
    - HDMI 2.0a complaint
    - HDCP 2.2 and HDCP 1.4 encryption
    - Support a maximum display resolution of 4096x2160@60Hz with a pixel clock frequency of 596MHz
- MIPI DSI Host Controlller (MIPI_DSI) - one 4-lane MIPI DSI display with pixels from either the DCSS HDR Display Controller or the eLCDIF. The key features of the MIPI DSI (controller and PHY) include:
    - Complaint to MIPI-DSI standard v1.1
    - Support up to 4 data lanes
    - Support 80Mbps - 1.5Gbps data rate in high speed operation
    - Support 10Mbps data rate in low power operation
- MIPI CSI Host Controller (MIPI_CSI) - two 4-lane MIPI CSI2 camera input. The key features of the MIPI CSI2 (controller and PHY) includes:
    - Complaint to MIPI-CSI2 standard
    - Support up to 4 data lanes
    - Support 80Mbps - 1.5Gbps data rate in high speed operation
    - Support 10Mbps data rate in low power operation
    - Support 5M pixel at 15 fps, 1080p30, 720p60, VGA at 60 fp

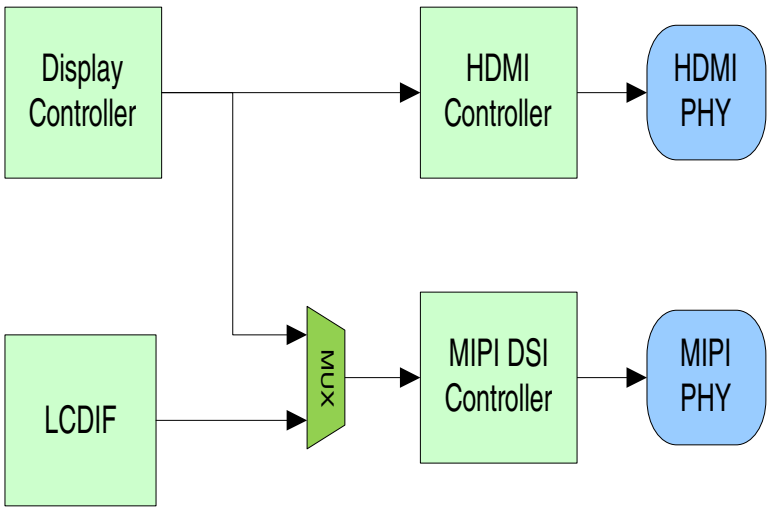


Figure 13-1. Display Interface

The following use cases are supported:

- Dual display: HDMI driven by Display Controller, with video playback support on HDMI; MIPI driven by LCDIF, without video playback support on MIPI
- Single display: HDMI driven by Display Controller, with video playback support on HDMI
- Single display: MIPI driven by Display Controller, with video playback support on MIPI
- Single display: MIPI driven by LCDIF, without video playback support on MIPI.

The i.MX 8M Dual/8M QuadLite/8M Quad do not support parallel display interface.

13.1.1.2 Graphics Processing Unit (GPU)

The chip utilizes a GPU to provide 2D/3D acceleration. Key features include:

- Run up to 800 MHz at 0.9V nominal voltage
- Support 267 million triangles/sec
- Support 1.6 Giga pixel/sec fill rate
- Support 25.6 GFLOPs 32-bit, 51.2 GFLOPs 16-bit
- Support OpenGL ES 1.1, 2.0, 3.0, 3.1
- Support OpenCL 1.1/1.2 FP
- TrustZone support using a local MMU to manage secure regions

For the best power and performance optimization, the GPU has following physical implementation:

- Independently powered with a dedicate power supply (VDD_GPU) to allow DVFS

- Support power down with internal power switch or its supply (VDD_GPU) shut off externally
- Dedicated PLL to allow most flexible clock frequency setting

13.1.1.3 Video Processing Unit (VPU)

The chip supports video decoding for various video format using the Hantro G2 + G1. Features include:

- 4Kp60 HEVC/H.265 Main, Main10 (Hantro G2) with best effort for 4Kp75
  - 4Kp75 is needed to support PiP (Picture in Picture) for live TV and dual layer Dolby Vision decoding one 4Kp60 stream and one 1080p60 stream
- 4Kp60 VP9 Profile 0, 2 (10 bit) (Hantro G2)
- 4Kp30 AVC/H.264 Baseline, Main, High decoder (Hantro G1)
- 1080p60 MPEG-2, MPEG-4p2, VC-1, VP8, RV9, AVS, MJPEG, H.263 decoder (Hantro G1)
- Lossless Frame Buffer Compression
- TrustZone support

For the best power and performance optimization, the VPU has following physical implementation:

- Independently powered with a dedicate power supply (VDD_VPU) to allow DVFS
- Support power down with internal power switch or its supply (VDD_VPU) shut off externally
- Dedicated PLL to allow most flexible clock frequency setting

13.1.1.4 Audio Interface

The device supports multiple audio interfaces as listed below:

Table 13-1. Audio Interface Summary

Interface	Function	RX Data Line	TX Data Line	Note
SAI-1	External audio	8	8	
SAI-2	External audio	1	1	
SAI-3	External audio	1	1	
SAI-4	HDMI-TX	0	4	Internal connection
SAI-5	External audio	4	4	
SAI-6	External audio	1	1	
SPDIF-1	External audio	1	1	
SPDIF-2	HDMI ARC	1	0	Internal connection

In addition to the general audio input/output functions, the audio interfaces support the following features:

- SAI-1 supports up to 16-channels TX (8 lanes) and 16-channels RX (8 lanes) at 384KHz/32-bit
- SAI-5 supports up to 8-channels TX (4 lanes) and 8-channels RX (4 lanes) at 384KHz/32-bit
- SAI-2/3/6 supports up to 2-channels TX (1 lanes) and 2-channels RX (1 lanes) at 384KHz/32-bit
- SAI-2/3/6 support up to 2-channels TX (1 lane) and 2-channels RX (1 lane) at 384KHz/32-bit
- SAI-1 supports glue-less switching between PCM and DSD operation for popular audio DACs
- SPDIF-1/2 supports raw capture mode that can save all incoming bits into an audio buffer

The SAI-1/2/3/5/6 and SPDIF-1 share GPIO pads on the chip through IOMUX. Common use cases supported by the audio interfaces are listed in the table below (many other configurations are possible). The number is the data lanes supported

Table 13-2. Audio Interface Use Cases

		UC 1	UC 2	UC 3	UC 4	UC 5	UC 6	UC 7	UC 8
SAI-1	TX	8 + DSD	8 + DSD	8 + DSD	8	8	8	4	4
	RX	8	8	8	8	8	8	4	4
SAI-2	TX	1		1		1	1		1
	RX	1		1		1	1		1
SAI-3	TX	1			1	1	1	1	1
	RX	1			1	1	1	1	1
SAI-5	TX		4		4	1		4	1
	RX		4	4	4	1	4	4	1
SAI-6	TX							1	1
	RX							1	1
SPDIF-1	TX	1	1	1	1	1	1	1	1
	RX	1	1	1	1	1	1	1	1

13.1.1.4.1 SAI Master Clock Inputs/Outputs

The MCLK pin on each SAI module can be configured as either an input or an output. When configured as an output, the SAIn_CLK_ROOT from the CCM is routed to the pad output. When configured as an input, the external input to the pad is routed to SAIn_MCLK, which can be used as master clock for SAI. Below is the diagram showing the both input/output options, by using SAI1 as the example.

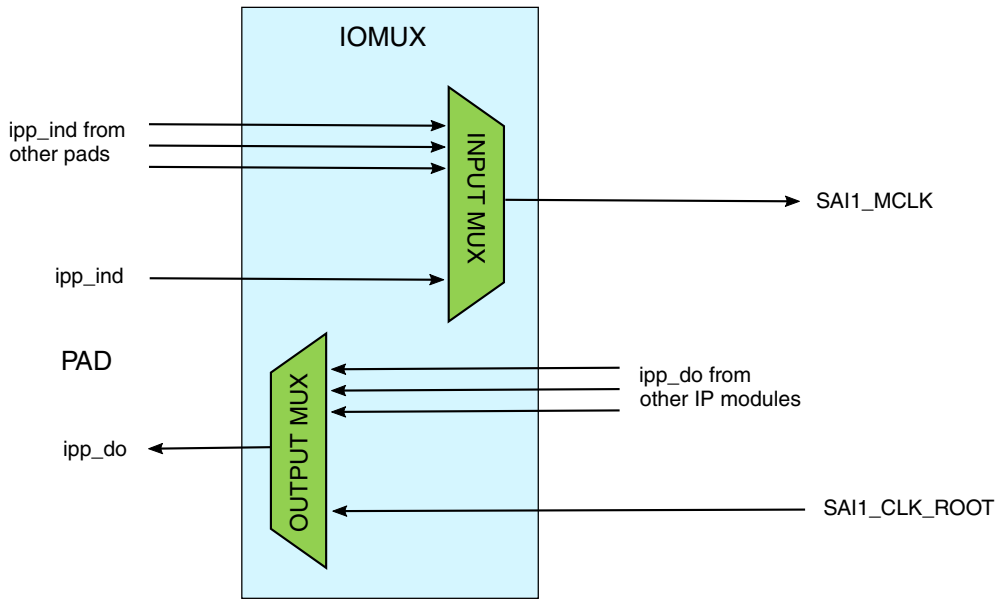


Figure 13-2. SAI MCLK Input and Output

Each SAI module supports up to 3 master clock inputs. The TX and RX sub-module inside each SAI can independently select one of the clock inputs as its master clock. This allows TX and RX of one SAI to run from different clock sources. The master clock inputs have the following options:

- SAIn_MCLK[1] can be selected from SAIn_CLK_ROOT from CCM or SAIn_MCLK from IOMUX. This is the most straight-forward clock routing in which SAIn only uses its own clock source from CCM or IO pad
- SAIn_MCLK[2] can be selected from the following clock sources:
  - Any of the SAIn_CLK_ROOT from CCM
  - Any of the SAIn_MCLK from IOMUX
  - Other clock sources from SPIDF
- SAIn_MCLK[3] has the same clock source options as SAIn_MCLK[2]. This allows both TX and RX to have access to all the options without any dependency between each other

The clock options for master clock on SAI are shown in the diagram blow, by using SAI-1 as an example.

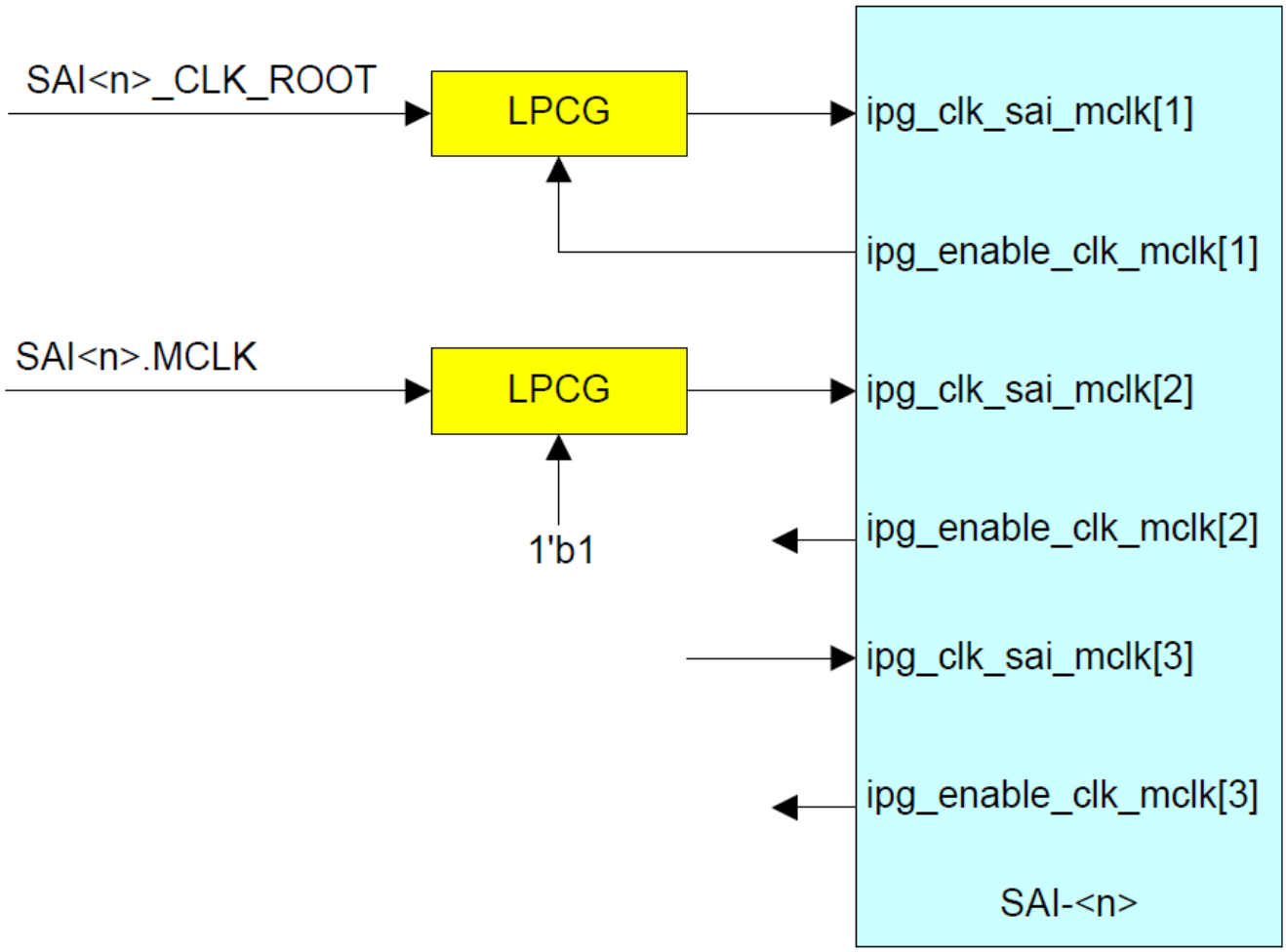


Figure 13-3. SAI Master Clock Options

The options on MCLK[1] are also available on MCLK[2] and MCLK[3]. This provides a similar SAI clock structure as i.MX6/i.MX7 processors.

The configuration of the MUX for master clock is controlled by IOMUXC_GPR registers. They should be configured before the SAI clock is enabled to avoid glitches on the clock.

13.1.1.4.2 SDMA Support

SDMA support is provided for SAI and SPDIF. In order to meet the audio data rate, two SDMA modules are used.

Because the SAI-2/3 and SPDIF-1/2 do not require high data throughput, they are assigned to SDMA-1, shared with other peripherals such as UART/SPI. SAI-1/4/5/6 need to support high sample rate and multi-channel audio, they are assigned to SDMA-2, which is a dedicated SDMA engine for audio. The SDMA-2 frequency is increased to 500/250 instead of 133/66 to make sure it has enough throughput.

Table 13-3. SDMA Assignment for Audio Interface

SDMA	Audio Interface	Note
SDMA-1 133MHz AHB, 66MHz IPG	SAI-2	1 TX + 1 RX
	SAI-3	1 TX + 1 RX
	SPDIF-1	TX + RX
	SPDIF-2	RX only, for HDMI ARC
SDMA-2 500MHz AHB, 250MHz IPG	SAI-1	8 TX + 8 RX
	SAI-4	4 TX only, for HDMI
	SAI-5	4 TX + 4 RX
	SAI-6	1 TX + 1 RX

13.2 Enhanced LCD Interface (eLCDIF)

13.2.1 Overview

The enhanced Liquid Crystal Display Interface (LCDIF) is a general purpose display controller used to drive a wide range of display devices varying in size and capability.

The LCDIF block supports the following:

- Displays with an asynchronous parallel MPU interface for command and data transfer to an integrated frame buffer.
- Displays that support moving pictures and require the RGB interface mode (DOTCLK interface).
- VSYNC mode for high-speed data transfers.
- Digital video encoders that accept ITU-R BT.656 format 4:2:2 YCbCr digital component video and convert it to analog TV signals.

The LCDIF provides fully programmable functionality to supported interfaces:

- Bus master interface to source frame buffer data for display refresh. This interface can also be used to drive data for "Smart" displays.
- PIO interface to manage data transfers between "Smart" displays and SoC.
- 8/16/18/24/32 bit LCD data bus support available depending on I/O mux options.
- Programmable timing and parameters for MPU, VSYNC, and DOTCLK LCD interfaces to support a wide variety of displays.
- ITU-R BT.656 mode (called Digital Video Interface or DVI mode here) including progressive-to-interlace feature and RGB to YCbCr 4:2:2 color space conversion to support 525/60 and 625/50 operation.

13.2.2 Clocks

The following table describes the clock sources for LCDIF. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 13-4. LCDIF Clocks

Clock name	Clock Root	Description
apb_clk	MAIN_AXI_CLK_ROOT	AXI clock
pix_clk	LCDIF_PIXEL_CLK_ROOT	Pixel clock

13.2.3 Functional Description

Bus Interface Mechanisms through Initializing the LCDIF, describe the internal pipeline for the LCDIF interfaces. Differences for each mode are then described in separate sections, as follows:

- MPU Interface
- VSYNC Interface
- DOTCLK Interface
- ITU-R BT.656 Digital Video Interface (DVI)

LCDIF pin usage by interface mode is described in LCDIF Pin Usage by Interface Mode.

Figure 13-4. Top-Level Block Diagram of LCDIF subsystem

13.2.3.1 Bus Interface Mechanisms

The LCDIF module has memory-mapped control, data and status registers. It provides several interfaces to transfer data between the display and SoC.

The bus master interface is used to initiate the requests to transfer data from external memory to the display. It is completely autonomous, or no CPU intervention is required, to manage the cyclical nature of refreshing standard display types. Bus mastering can also be used for MPU mode data writes.

The PIO interface is used to interface to "Smart" displays to transfer frame buffer data and control information to/from the external display. The host CPU executes display drivers to manage the display solution. The following sections describe the system bus interface mechanisms.

13.2.3.1.1 Bus Master Operation in Write/Display Modes

The LCDIF block has a bus master interface that initiates requests for data to drive the display. The LCDIF_MASTER bit must be set to 1 to enable the bus master interface. Software should program all control registers required to transfer the frame sequence.

In the MPU and VSYNC mode, single frames are transferred. When a complete frame is transferred, LCDIF enters idle and clears the RUN bit in the CTRL register. For subsequent frame transmission, the LCDIF setup sequence should be repeated.

The DOTCLK and DVI mode is used to refresh the display at the desired refresh rate and resolution, and drive displays that don't integrate a display buffer memory. When the display is refreshed, the LCDIF will automatically update the LCDIF_CUR_BUF_ADDR register with the value in LCDIF_NEXT_BUF_ADDR at the end of current frame and start fetching the next frame from the new address. If the LCDIF_NEXT_BUF_ADDR register was not updated within a frame refresh cycle, LCDIF will keep transmitting the last frame until a new value is programmed into that register.

LCDIF also provides the capability of interlacing a progressive frame by fetching odd lines in the first field and then fetching even lines in the second field. This feature can be used in the DVI mode and can be turned on by setting the INTERLACE_FIELDS bit in the LCDIF_CTRL1 register.

13.2.3.1.2 System Bus Master Performance

The performance of the LCDIF block can be controlled by changing the burst length and the outstanding cycle issuing capability depending on the memory bandwidth requirements. Two fields in the LCDIF_CTRL2 register will throttle system memory requests. The LCDIF_CTRL2_OUTSTANDING_REQS field will control how many requests the LCDIF can have in flight on any given clock cycle. This should be programmed based on the expected system bus latency for returned read data. Also, the LCDIF_CTRL2_BURST_LEN_8 bit will set the number of 64 bit words requested for each LCDIF system bus request to either 8 or 16 QWORDS. Generally, 4 outstanding requests of length 16 will provide enough performance to drive any standard display resolution. These configuration bits are intended to change the access pattern of the LCDIF to optimize system bus throughput when other system masters will contend for system memory resources.

The LCDIF_THRES register can also be used to optimize bus throughput and power consumption.

The LCDIF_THRES_PANIC value can be used to raise the priority of requests initiated by the LCDIF to alter how the LCDIF requests are arbitrated by the system bus infrastructure. The panic output control signal is raised when the number of 32bpp pixel equivalents in the LFIFO is less than this programmed value. Since the LFIFO is arranged as a 256x64bit quadword FIFO, it contains two 32bpp pixels per quadword, or 512 32bpp pixels total. To set the panic output when 3/4s of the LFIFO is empty, set the LCDIF_THRES_PANIC value to 3/4 * 512, or 128. The panic signal output is used to assess higher priority to LCDIF system requests to avoid LCDIF under run errors during periods of high system bandwidth utilization.

The features available with the LCDIF_THRES register require support from system clocking and dynamic priority control. Refer to the appropriate block documentation to assess the system support for these features.

13.2.3.2 Write Data Path

LCDIF supports raster based frame buffers and there is no support for tiled buffers.

There are several options to accommodate endianness of display buffers in memory before the data is processed for the external display. The LCDIF_CTRL[INPUT_DATA_SWIZZLE] field provides the following options for data word multiplexing:

```
00 (0): No swizzle (little-endian)
01 (1): Swap bytes 0 and 3, swap bytes 1 and 2 (big-endian)
10 (2): Swap half-words
11 (3): Swap bytes within each half-word
```

The LCDIF_CTRL[WORD_LENGTH] field indicates the input data/pixel format. LCDIF_TRANSFER_COUNT register denotes how much data is contained in each frame. The LCDIF_TRANSFER_COUNT[H_COUNT] field indicates the number of pixels per line and LCDIF_TRANSFER_COUNT[V_COUNT] indicates the total number of lines per frame. The LCDIF_CTRL1[BYTE_PACKING_FORMAT] field can be used to specify which bytes within the 32-bit word are going to be valid. For example, if the entire 32-bit word is valid, LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be set to 0xF, if only lower 3 bytes of each word in the frame buffer are valid, then LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be set to 0x7.

The LCDIF_CTRL[LCD_DATABUS_WIDTH] field suggests the width of the bus going to the display controller. There is an option to source all 32 bits of the input word and transfer it to the output I/O display interface. If the LCDIF_CTRL[LCD_DATABUS_WIDTH] is not the same as LCDIF_CTRL[WORD_LENGTH], LCDIF will perform RGB to RGB color space conversion. For example, if the input frame has fewer bits per pixel than the display, as in

a 16 bpp input frame going to 24 bpp LCD, LCDIF will pad the MSBs of each color to the LSBs of the same color for each pixel. If the input frame has more bits per pixel than the display, for example, 24 bpp input frame going to 16 bpp LCD, LCDIF will drop the LSBs of each color channel to convert to the lower color depth. LCDIF also has the capability to support delta pixel displays by swizzling the R, G and B colors of each pixel in the odd and even lines of the frame separately by programming the LCDIF_CTRL2[ODD_LINE_PATTERN] and the LCDIF_CTRL2[EVEN_LINE_PATTERN] bit fields. This operation occurs after the RGB-to-RGB color space conversion operation.

LCDIF also supports RGB to YCbCr 4:2:2 color space conversion. This is useful in the DVI mode since the TV encoder requires input in YCbCr 4:2:2 format. The LCDIF_CSC* registers have complete programmability over the CSC coefficients and offsets. The values must be written into these registers in the signed two's complement format.

The following list shows how the different input/output combinations can be obtained:

- LCDIF_CTRL[WORD_LENGTH]=1 indicates that the input is 8-bit data. This is most likely going to be used for sending commands in MPU interface, or maybe a gray scale image. Any combination of LCDIF_CTRL1[BYTE_PACKING_FORMAT] is permissible.

  Limitation: LCDIF_TRANSFER_COUNT[H_COUNT] must be a multiple of the sum of BYTE_PACKING_FORMAT [3], BYTE_PACKING_FORMAT [2], BYTE_PACKING_FORMAT [1] and BYTE_PACKING_FORMAT [0]. LCDIF_CTRL[LCD_DATABUS_WIDTH] must be 1, indicating an 8-bit data bus.

- LCDIF_CTRL[WORD_LENGTH]=0 implies the input frame buffer is RGB 16 bits per pixel. LCDIF_CTRL[DATA_FORMAT_16_BIT] field determines the pixels are RGB 555 or RGB 565.

  Limitation: LCDIF_CTRL1[BYTE_PACKING_FORMAT] should be 0x3 or 0xC if there is only one pixel per word. If there are two pixels per word, it should be 0xF and LCDIF_TRANSFER_COUNT[H_COUNT] will be restricted to be a multiple of 2 pixels.

- LCDIF_CTRL[WORD_LENGTH]=2 indicates that input frame buffer is RGB 18 bits per pixel, that is, RGB 666. The valid RGB values can be left-aligned or right-aligned within a 32-bit word. The alignment of the valid 18 bits within a word is indicated by the LCDIF_CTRL[DATA_FORMAT_18_BIT] bit.

  Limitation: LCDIF_CTRL1[BYTE_PACKING_FORMAT] can be 0x7, 0xE or 0xF. Packed pixels are not supported in this case. LCDIF_TRANSFER_COUNT[H_COUNT] can be any number.

- LCDIF_CTRL[WORD_LENGTH]=3 indicates that the input frame-buffer is RGB 24 bits per pixel (RGB 888). If LCDIF_CTRL1[BYTE_PACKING_FORMAT] is 0x7, it indicates that there is only one pixel per 32-bit word and there is no restriction on LCDIF_TRANSFER_COUNT[H_COUNT]. This is also the option that provides 32 bit output depending on the I/O muxing options available. The fourth byte, or bits [31:24], and connected to the I/Os if this muxing is available in the chip package.

  Limitation: If LCDIF_CTRL1[BYTE_PACKING_FORMAT] is 0xF, it indicates that the pixels are packed, that is, there are 4 pixels in 3 words or 12 bytes and LCDIF_TRANSFER_COUNT[H_COUNT] must be a multiple of 4 pixels.

- LCDIF_CTRL1[YCBCR422_INPUT]=1 implies that the input frame is in YCbCr 4:2:2 format. LCDIF_CTRL1[BYTE_PACKING_FORMAT] must be 0xF.

  Limitation: LCDIF_CTRL[LCD_DATABUS_WIDTH] must be 8-bit and LCDIF_TRANSFER_COUNT[H_COUNT] must be a multiple of 2 pixels.

  LCDIF_CTRL2[ODD_LINE_PATTERN] and LCDIF_CTRL2[EVEN_LINE_PATTERN] must be 0 when any of LCDIF_CTRL[RGB_TO_YCBCR422_CSC] or LCDIF_CTRL1[INTERLACE_FIELDS] or LCDIF_CTRL[YCBCR422_INPUT] bits is 1.

After the RGB to RGB or RGB to YCbCr 4:2:2 color space conversions, there is one more opportunity to swizzle the data before sending it out to the display or the encoder. This can be done with the LCDIF_CTRL[CSC_DATA_SWIZZLE] field, and it provides the same options as the LCDIF_CTRL[INPUT_DATA_SWIZZLE] register.

Finally, there is an option to shift the output data before sending it out to the display. This is done based on the LCDIF_CTRL[SHIFT_DIR] and LCDIF_CTRL[SHIFT_NUM_BITS] fields.

Figure 13-5. General Operations in Write Data Path

The examples in the following figures illustrate some different combinations of register programming for write mode. Assume that the data transferred over the system bus within a 32 bit word is organized as {A7-A0, B7-B0, C7-C0, D7-D0} in 8-bit mode and {A15-A0, B15-B0} in 16-bit mode.

In this example, all 32 bits of the input word are transferred out over an 8 bit display bus. Each byte within the 32 bit word is shifted to the right with zeros appended to bits D[7:6]. The input data bits [7:2] are shifted to the right by 2 bits and presented on the D[5:0].

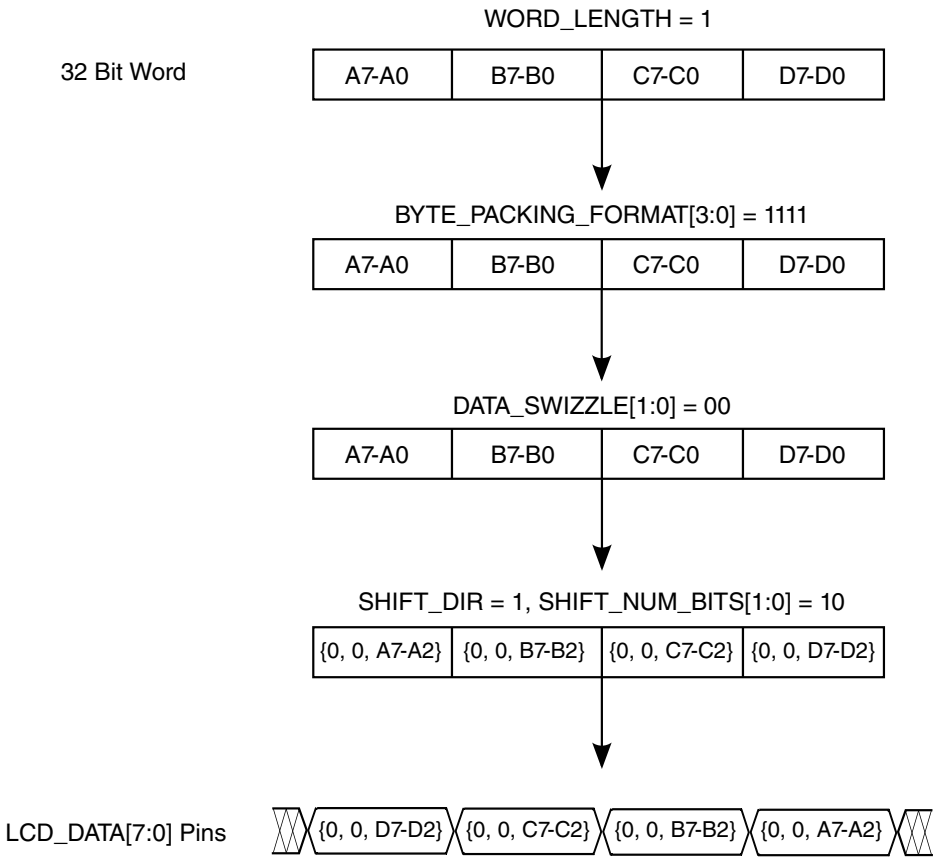


Figure 13-6. Register programming for write mode

In this 8 bit display interface example, one byte of the input word is deleted and not transferred over the external 8 bit display interface. This mode could be used to transfer 24bpp pixels over the 8 bit interface. In this case, the 4th unused byte is not transferred.

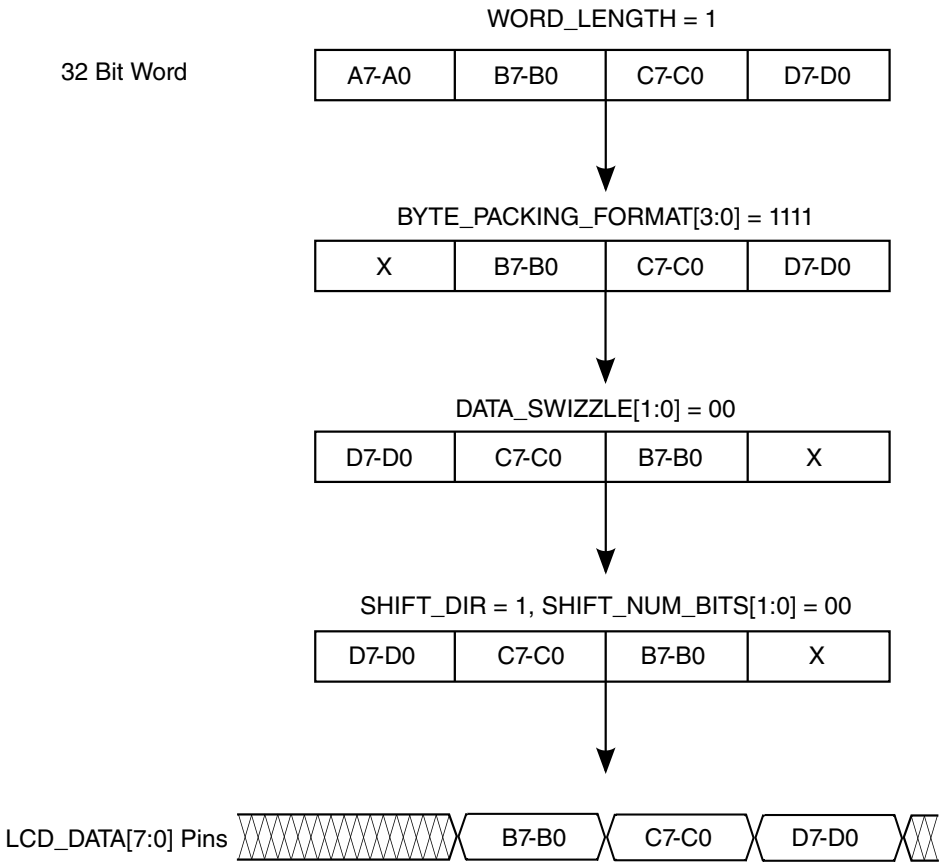


Figure 13-7. Register programming for write mode

The following example uses a 16 bit display interface. Each 16 bit half word is shifted to the right by two bits with zeros appended to the most significant two bits.

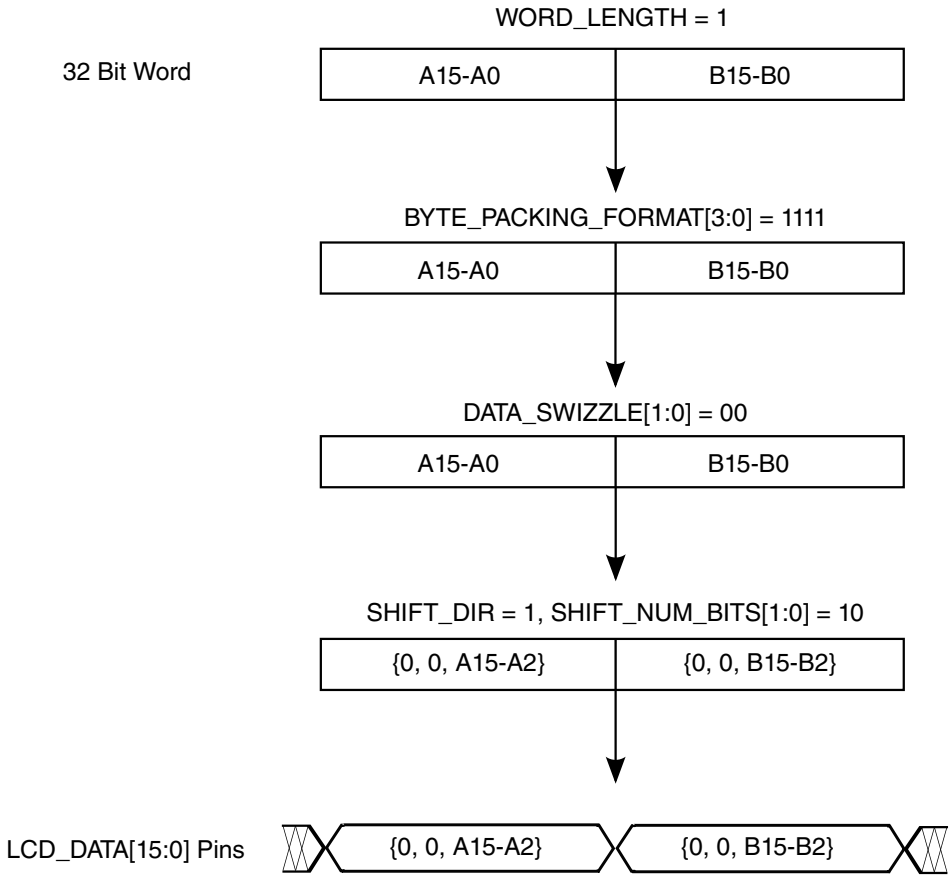


Figure 13-8. Register programming for write mode

This example indicates how an unpacked frame buffer can sourced for display. Only a single 16 bit half word within the 32 bit word is tranferred out via the 16 display bus.

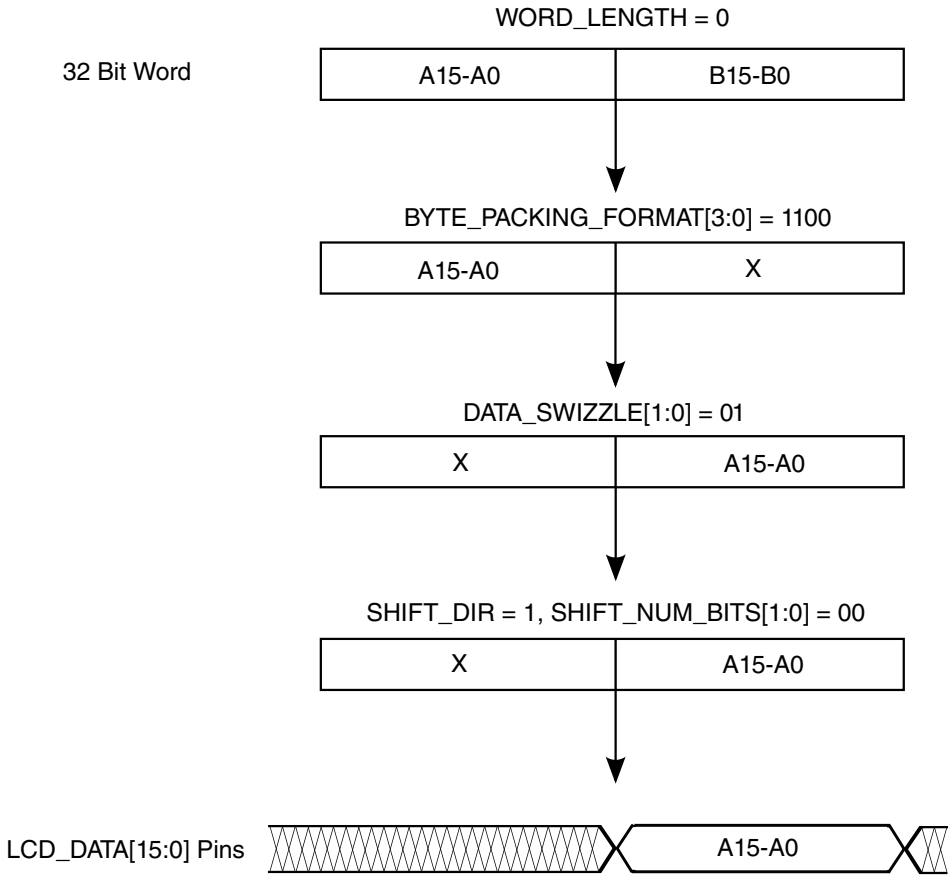


Figure 13-9. Register programming for write mode

13.2.3.3 Read Data Path

Figure 13-10 shows the MPU read data path in detail.

LCDIF can read from an external display that follows the 6800/8080 MPU protocol.

The display bus width is determined by the LCD_DATABUS_WIDTH bit field. The data sampled at every read strobe is called a subword and the number of subwords that can be packed in a 32-bit word is given by the READ_MODE_NUM_PACKED_SUBWORDS bit field. The INITIAL_DUMMY_READ bit field directs the LCDIF to skip the number of programmed subwords before starting to process read data. This feature is useful in the case of an LCD controller that returns the last written data the first time a read is issued, and then sends the correct data thereafter. SHIFT_DIR and SHIFT_NUM_BITS bit fields indicate whether the data needs to be shifted before getting stored in the internal registers. For example, a value of 2 in READ_MODE_NUM_PACKED_SUBWORDS if lcd databus width is 8 bits indicates two bytes should be packed in a 32-bit word, while if the lcd databus width is 16 bits, it indicates that two half words (or 4 bytes) should be packed.

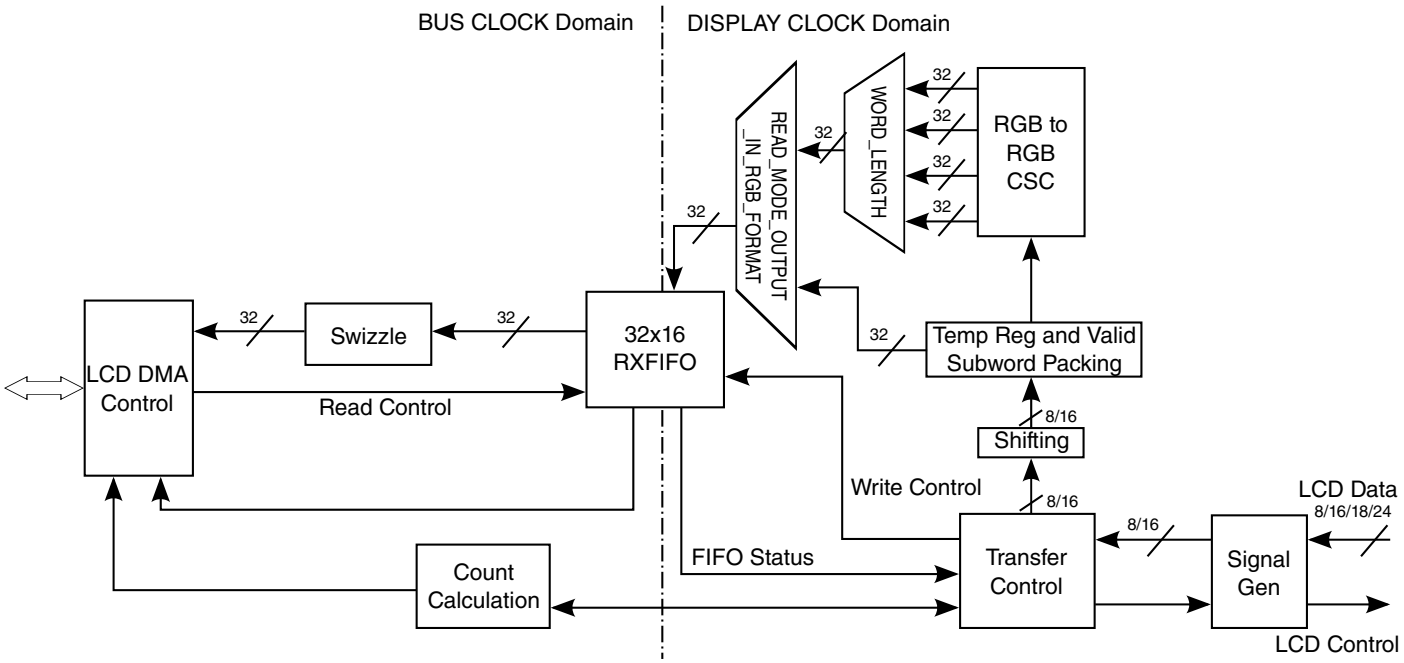


Figure 13-10. MPU Read Data Path

After the last subword within a word is reached, the block looks at the READ_PACK_DIR in the HW_LCDIF_CTRL2 register. If this bit is set, the block will swizzle the data, but only within the valid bytes, unlike in the write mode, where swizzle occurs across all 4 bytes. If the READ_MODE_OUTPUT_IN_RGB_FORMAT bit is set, LCDIF will convert the data obtained from the READ_PACK_DIR operation into 24-bit unpacked RGB and then re-convert it into 16/18/24 bpp RGB depending on the WORD_LENGTH field. The DATA_FORMAT_16/18/24_BIT bit fields are also considered while converting to 24-bit unpacked RGB format. For example, if DATA_FORMAT_18_BIT is 1, the RGB666 data will be packed in the upper bits [31:4]of a 32-bit word, and that bit is 0, the data will be packed in the lower bits [17:0]. After all these operations, the data gets written into the RXFIFO.

The following figures show some examples of how data is handled in different MPU read modes.

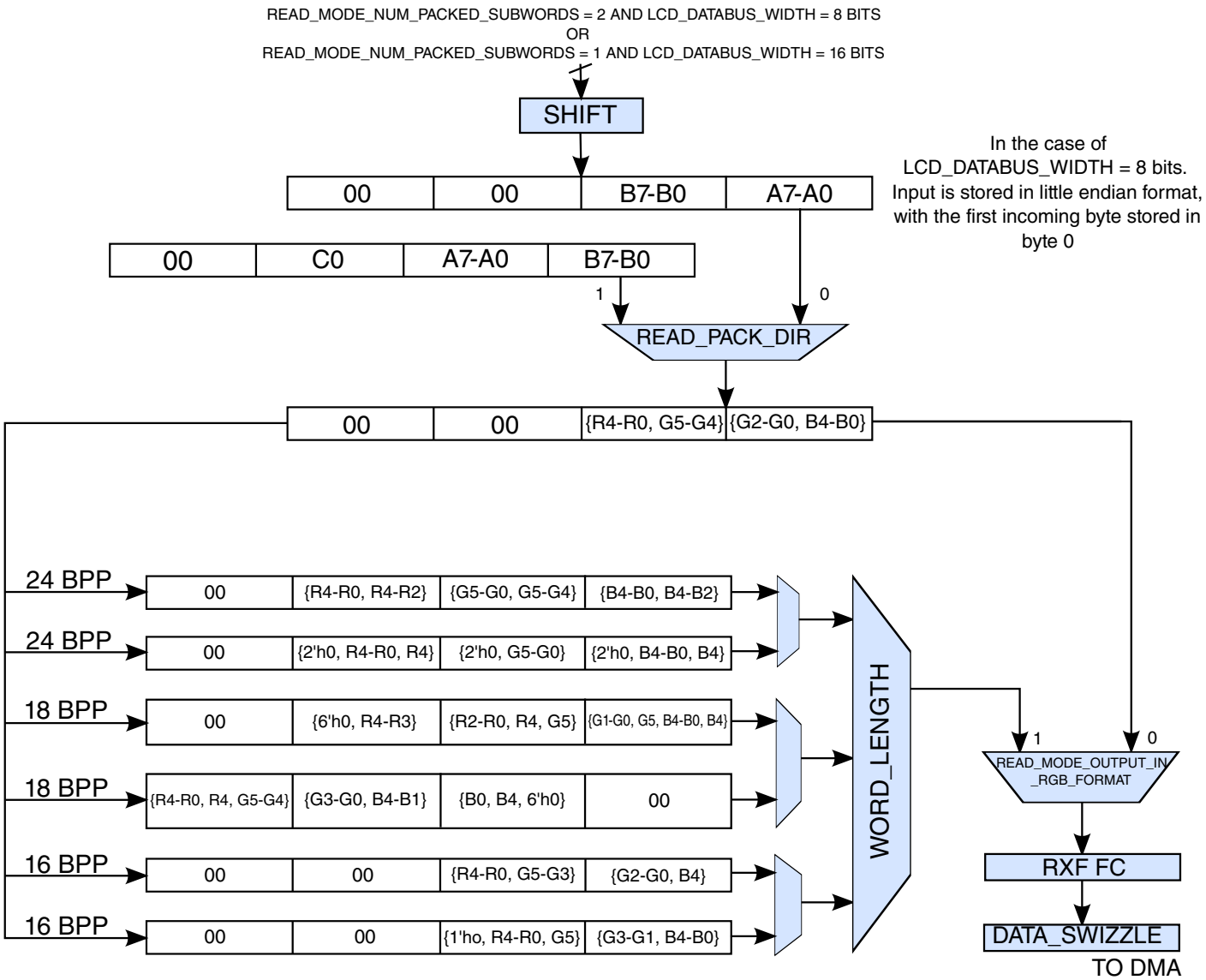


Figure 13-11. Data in MPU read mode

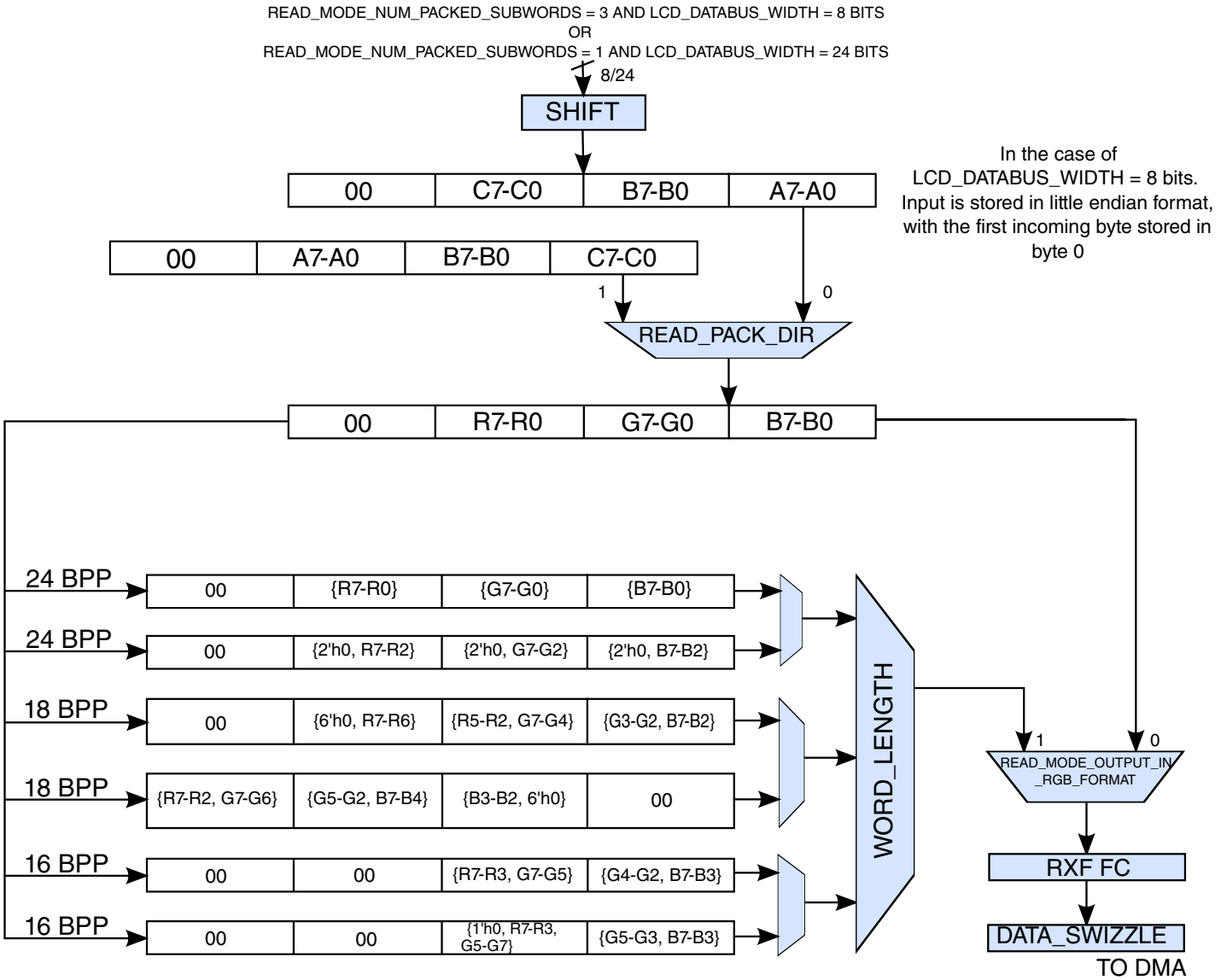


Figure 13-12. Data in MPU read mode

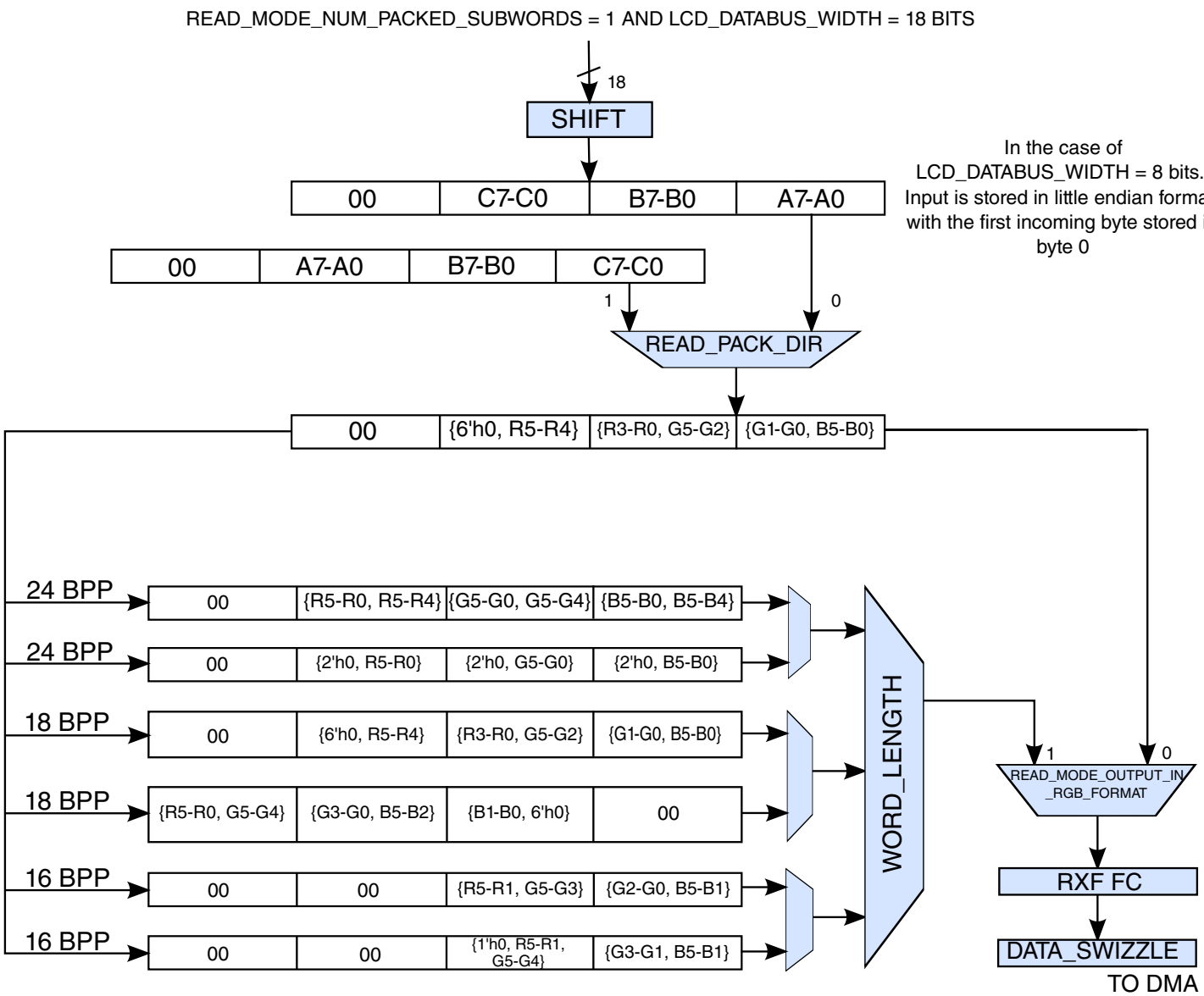


Figure 13-13. Data in MPU read mode

Restrictions:

READ_PACK_DIR should only be used if it is required to swizzle the subwords before doing RGB to RGB CSC, otherwise the DATA_SWIZZLE field should be used to swizzle across bytes.

READ_PACK_DIR must be 0 if LCD_DATABUS_WIDTH is 8 bits and READ_MODE_NUM_PACKED_SUBWORDS =1

If READ_MODE_OUTPUT_IN_RGB_FORMAT bit is set, the following restrictions should be followed:

- If LCD_DATABUS_WIDTH = 8 bits, then READ_MODE_NUM_PACKED_SUBWORDS <= 3.
- If LCD_DATABUS_WIDTH = 16/18/24 bits, then READ_MODE_NUM_PACKED_SUBWORDS = 1.

13.2.3.4 LCDIF Interrupts

LCDIF supports a number of interrupts to aid controlling and status reporting of the block.

All the interrupts have individual mask bits for enabling or disabling each of them. They all get funneled through a single interrupt line connected to the interrupt collector (ICOLL).

The following list describes the different interrupts supported by LCDIF:

- Underflow interrupt is asserted when the clock domain crossing FIFO (TXFIFO) becomes empty but the block is in active display portion during that time. Software should take corrective action to make sure that this does not happen.
- In the bus master mode, the overflow interrupt will be asserted if the block has requested more data than it's FIFOs could hold. In the read mode, it will be asserted if the RxFIFO becomes full and the block reads more data.
- VSYNC edge interrupt will be asserted every time a leading VSYNC edge occurs.
- Cur_frame_done interrupt occurs at the end of every frame in all modes except DVI. In DVI mode, if IRQ_ON_ALTERNATE_FIELDS bit is set, it will occur at the end of every frame, otherwise it will occur at the end of every field.

13.2.3.5 Initializing the LCDIF

This section describes write modes and MPU read mode.

13.2.3.5.1 Write Modes

The following initialization steps are common to all LCDIF write modes of operation before entering any particular mode.

Initialization steps:

1. Configure the external I/Os to correctly interface the external display, when required.
2. Start the DISPLAY CLOCK (pix_clk) clock and set the appropriate frequency by programming the registers in CCM.

3. Start the BUS CLOCK (apb_clk) and set the appropriate frequency by programming the registers in CCM.
4. Bring the LCDIF out of soft reset and disable the clock gate bit.
5. Reset the LCD controller by setting LCDIF_CTRL1[RESET] bit appropriately, being careful to observe the reset requirements of the controller. See Behavior During Reset for more information on Reset requirements.
6. Make sure LCDIF_CTRL[READ_WRITEB] bit is 0.
7. Set the transfer mode of operation to bus master. The LCDIF_CTRL[MASTER] bit determines the transfer mode selected. Bus master (LCDIF_CTRL[MASTER] =1), or PIO (LCDIF_CTRL[MASTER] =0) mode is the transfer mode to select.
8. Set the LCDIF_CTRL[INPUT_DATA_SWIZZLE] according to the endianness of the LCD controller. Also, set the LCDIF_CTRL[DATA_SHIFT_DIR] and LCDIF_CTRL[SHIFT_NUM_BITS] if it is required to shift the data left or right before it is output.
9. Set the LCDIF_CTRL[WORD_LENGTH] field appropriately: 0 = 16-bit input, 1 = 8-bit input, 2 = 18-bit input, 3 = 24/32-bit input. Also, select the correct 16/18/24 bit data format with the corresponding fields in LCDIF_CTRL register.
10. Set the LCDIF_CTRL1[BYTE_PACKING_FORMAT] field according to the input frame.
11. Set the LCDIF_CTRL[LCD_DATABUS_WIDTH] appropriately: 0 = 16-bit output, 1 = 8-bit output, 2 = 18-bit output, 3 = 24/32-bit output.
12. Enable the necessary IRQs.

13.2.3.5.2 MPU Read Mode

The following initialization steps should be done to enter the MPU read mode of operation:

Initialization steps:

1. Configure the external I/Os to correctly interface the external display.
2. Start the DISPLAY CLOCK (pix_clk) and set the appropriate frequency by programming the registers in CCM.
3. Start the BUS CLOCK (apb_clk) and set the appropriate frequency by programming the registers in CCM.
4. Bring the LCDIF out of soft reset and clock gate.
5. Reset the LCD controller by setting LCDIF_CTRL1_RESET bit appropriately, being careful to observe the reset requirements of the controller.
6. Set the READ_WRITEB bit in LCDIF_CTRL register to 1.
7. Set the LCDIF_MASTER bit in LCDIF_CTRL register to 0. Bus master mode is not supported for reading data from the display.

8. Also, set the DATA_SHIFT_DIR and SHIFT_NUM_BITS if it is required to shift the data left or right before it is output.
9. Indicate if the read data needs to color-space-converted and stored in a different RGB format by setting the READ_MODE_OUTPUT_IN_RGB_FORMAT field accordingly.
10. Set the WORD_LENGTH field appropriately: 0 = 16-bit input, 1 = 8-bit input, 2 = 18-bit input, 3 = 24-bit input if READ_MODE_OUTPUT_IN_RGB_FORMAT is required. Also, select the correct 16/18/24 bit data format with the corresponding fields in LCDIF_CTRL register.
11. Set the READ_MODE_NUM_PACKED_SUBWORDS field in LCDIF_CTRL2 according to the number of subwords per word required to be packed.
12. Set the READ_PACK_DIR to 1 if it is required to store the data in big-endian format.
13. Set the LCD_DATABUS_WIDTH appropriately: 0 = 16-bit output, 1 = 8-bit output, 2 = 18-bit output, 3 = 24-bit output.
14. Enable the necessary IRQs.

13.2.3.6 MPU Interface

The MPU interface is used to transfer data and commands between the SoC via the LCDIF and the external display at modest data rates.

Bus master or PIO transactions using the LCDIF_DATA register can be used for MPU mode write operations. For MPU mode read operations, only PIO can be used. LCDIF can support the 6800 as well as the 8080 MPU protocol. If DOTCLK_MODE, DVI_MODE and VSYNC_MODE bits in LCDIF_CTRL registers are 0, it implies that the block is in MPU interface mode of operation. The LCDIF MPU mode has four basic timing parameters: Setup and Hold for the Command/Data register selection (TCS, TCH) and Setup and Hold for the Data bus (TDS, TDH). These parameters are expressed in DISPLAY CLOCK (pix_clk) cycles. The LCD_WR signal is used as the write strobe while LCD_RS signal is typically used to switch between command and data modes.

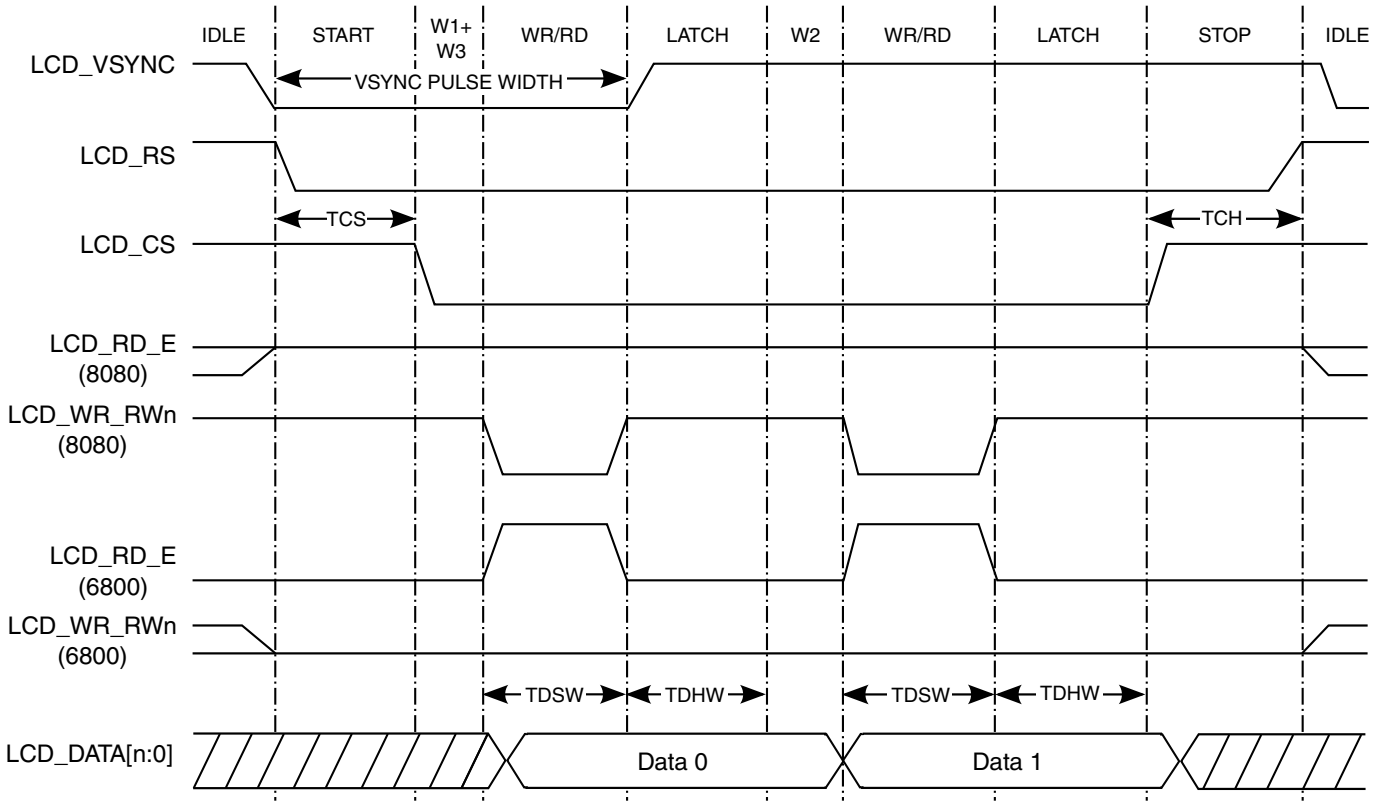


Figure 13-14. Timing in write mode of 6800 and 8080 protocols

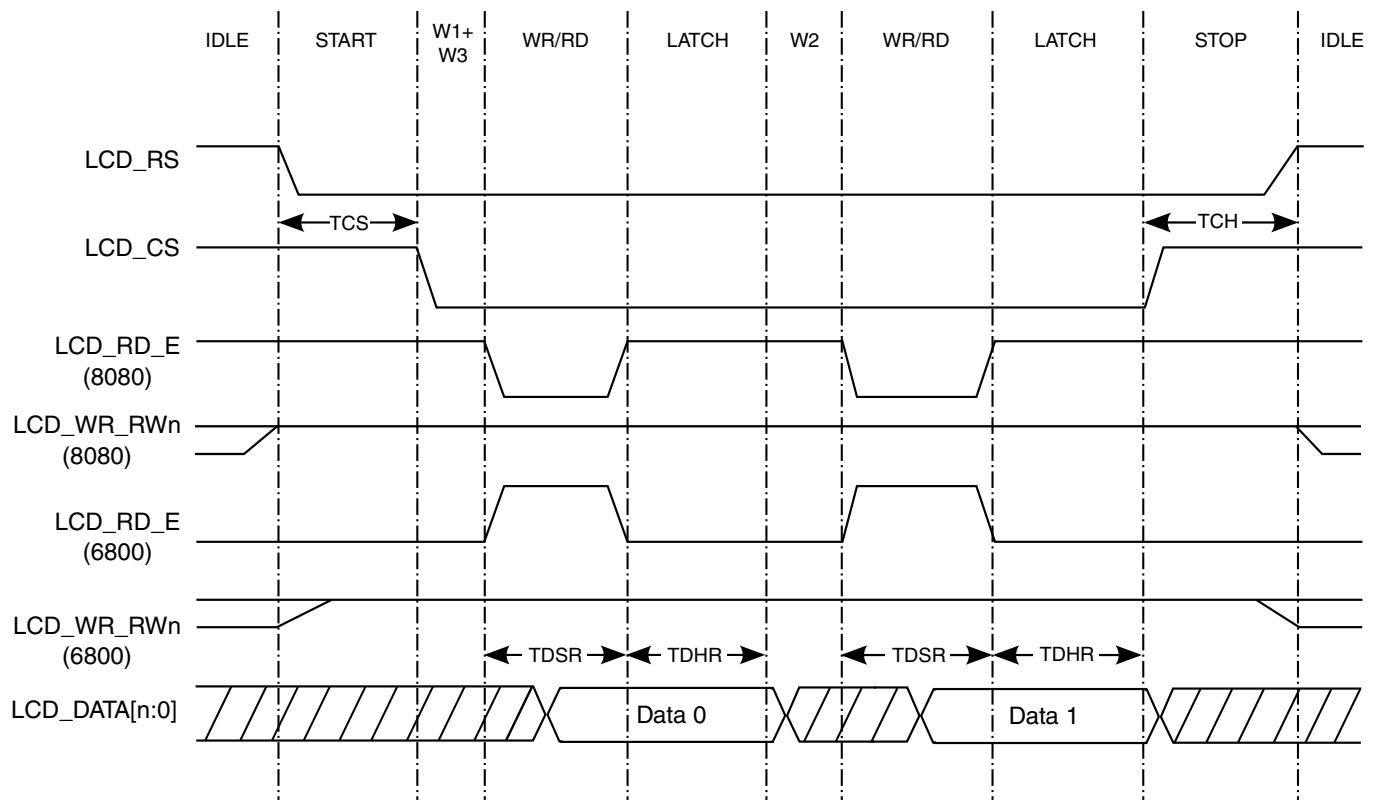


Figure 13-15. Read timing interface in 6800 and 8080 protocols

The LCDIF has flexible pin and strobe timings which enable it to optimally support a wide range of LCDs. The minimum cycle time is two DISPLAY CLOCK (pix_clk) cycles (TDS=TDH=1). For example, this results in a maximum LCD data rate of 12 MB/s when DISPLAY CLOCK (pix_clk) is 24 MHz. TDS and TDH are 8-bit values, so the minimum LCDIF period is 510 DISPLAY CLOCK (pix_clk) cycles (47 KHz with a 24 MHz DISPLAY CLOCK (pix_clk)). The timings are not automatically adjusted if the DISPLAY CLOCK (pix_clk) frequency changes, so it may be necessary to adjust the timings if DISPLAY CLOCK (pix_clk) changes.

In the MPU interface mode, the LCDIF_CTRL_ BYPASS_COUNT bit must be 0. The RUN bit is cleared automatically once the LCDIF has received/transmitted all the data as per the LCDIF_TRANSFER_COUNT register and has completed the transfer to the panel. The current transfer can be cancelled/aborted if the RUN bit is manually made 0.

13.2.3.6.1 Code Example to Initialize the LCDIF in MPU Write Mode

```
// Note: Common initialization steps in Initializing the LCDIF must also be
// executed along with the following code
BF_CS1(LCDIF_CTRL, DATA_SELECT, 1); // 0 if sending command, 1 if sending data. Note that the
                                    // idle state for LCD_RS signal is high, regardless of the
                                     // programming of the DATA_SELECT register.
BF_CS1 (LCDIF_CTRL, MODE86, 8080_MODE);
BF_CS1 (LCDIF_CTRL, READ_WRITEB, 0);
BF_CS1 (LCDIF_CTRL, BYPASS_COUNT, 0); //Must be 0 in MPU mode
BF_CS1 (LCDIF_CTRL1, BUSY_ENABLE, 1);//Only if LCD controller implements a busy line
BF_CS4 (LCDIF_TIMING, CMD_HOLD, 2, CMD_SETUP, 2, DATA_HOLD, 2, DATA_SETUP, 2);  //Values
based
                              // on DISPLAY CLOCK (pix_clk) frequency and timing requirements
of controller.
                             // Note that these register must be non-zero for correct
operation.
BF_CS2 (LCDIF_TRANSFER_COUNT, H_COUNT, 320, V_COUNT, 240); //For a 320 RGB x 240 display
BF_CS1 (LCDIF_CTRL, RUN, 1);
```

The LCDIF is now ready to receive data via bus master PIO write transactions using the LCDIF_DATA register. Note that when using the PIO write operations to the LCDIF_DATA register, the software will need to poll the FIFO STATUS bits to ensure that it does not overflow the LCDIF data buffers. When LCDIF is done transmitting H_COUNT x V_COUNT pixels, it will stop, turn off the RUN bit and assert the cur_frame_done interrupt.

13.2.3.7 VSYNC Interface

The VSYNC interface uses the same protocol as the MPU interface, with an additional signal VSYNC at the frame rate of the display, as shown in the figure given in MPU Interface section.

It is used in the moving picture display mode where data has to be written to the internal LCD buffer at a speed higher than the display rate and displayed in synchronization with the VSYNC signal. This mode is selected by setting the VSYNC_MODE bit in LCDIF_CTRL register. The VSYNC signal is programmable for period, polarity and direction. Many other programmable parameters are shared with the MPU interface. The VSYNC_OEB bit in LCDIF_VDCTRL0 register indicates whether the display controller will send the VSYNC signal, or whether it should be generated by LCDIF. The timing of the VSYNC signal is based on the DISPLAY CLOCK (pix_clk) (make sure VSYNC_PULSE_WIDTH_UNIT = VSYNC_PERIOD_UNIT = 0 and VSYNC_ONLY = 1) and it is determined by the VSYNC_PERIOD, VSYNC_PULSE_WIDTH and VSYNC_POL fields in LCDIF_VDCTRL0-4 registers. The SYNC_SIGNALS_ON bit in LCDIF_VDCTRL4 register must be set if the target requires the VSYNC signal to be generated by LCDIF. If the WAIT_FOR_VSYNC_EDGE bit in LCDIF_CTRL register is set, it indicates that the hardware should wait until it sees the leading VSYNC edge before starting the data transfer. The VERITCAL_WAIT_CNT indicates the number of DISPLAY CLOCK (pix_clk) cycles from the leading VSYNC edge after which data transfer will be started on the interface.

In the VSYNC interface mode, the LCDIF_CTRL_ BYPASS_COUNT bit must be 0. The RUN bit is cleared automatically once the LCDIF has received/transmitted all the data as per the LCDIF_TRANSFER_COUNT register and has completed the transfer to the panel. The current transfer can be cancelled/aborted if the RUN bit is manually made 0.

13.2.3.7.1 Code Example to Initialize LCDIF in VSYNC Mode

```
// Note: Common initialization steps in Initializing the LCDIF must also be
// executed along with the following code
BF_CS1 (LCDIF_CTRL, DATA_SELECT, 1); // 0 if sending command, 1 if sending data. Note that
//the idle state for LCD_RS signal is high, regardless of the programming of the DATA_SELECT
//register.

BF_CS1 (LCDIF_CTRL, MODE86, 8080_MODE);
BF_CS1 (LCDIF_CTRL, BYPASS_COUNT, 0); //Must be 0 in MPU mode
BF_CS1 (LCDIF_CTRL1, BUSY_ENABLE, 0);
BF_CS4 (LCDIF_TIMING, CMD_HOLD, 2, CMD_SETUP, 2, DATA_HOLD, 2, DATA_SETUP, 2);  //Values
//based on DISPLAY CLOCK (pix_clk) frequency and timing requirements of controller. Note
that these
//register must be non-zero for the MPU and VSYNC modes.
BF_CS2 (LCDIF_TRANSFER_COUNT, H_COUNT, 320, V_COUNT, 240);//For a 320 RGB x 240 display
//The following section indicates setting up the VSYNC signal timing when VSYNC is an output
BF_CS1 (LCDIF_VDCTRL0, VSYNC_OEB, 0); //Making VSYNC signal an output
BF_CS1 (LCDIF_VDCTRL4, VSYNC_ONLY, 1); //Only need to generate VSYNC signal
BF_CS1 (VDCTRL0, VSYNC_POL, 0); //Setting the polarity of VSYNC signal to be low during
//VSYNC_PULSE_WIDTH time
BF_CS2 (LCDIF_VDCTRL0, VSYNC_PERIOD_UNIT, 0, VSYNC_PULSE_WIDTH_UNIT, 0);
BF_CS2 (LCDIF_VDCTRL1, VSYNC_PERIOD, 400000, VSYNC_PULSE_WIDTH, 100);//Frame display rate in
//terms of number of DISPLAY CLOCKS (pix_clk).
BF_CS2 (LCDIF_VDCTRL2, HSYNC_PULSE_WIDTH, 0, HSYNC_PERIOD, 0);
BF_CS1 (LCDIF_VDCTRL3, VERTICAL_WAIT_CNT, 50);
BF_CS1 (LCDIF_VDCTRL4, SYNC_SIGNALS_ON, 1);
BF_CS2 (LCDIF_CTRL, VSYNC_MODE, 1, WAIT_FOR_VSYNC_EDGE, 1); //set WAIT_FOR_VSYNC_EDGE if
```

```
//software wishes to transfer the next frame after the VSYNC edge occurs.
BF_CS1 (LCDIF_CTRL, RUN, 1);
```

The LCDIF is now ready to receive data via bus master requests or PIO writes to the LCDIF_DATA register. When LCDIF is done transmitting H_COUNT x V_COUNT pixels, it will stop, turn off the RUN bit and assert the cur_frame_done interrupt.

13.2.3.8 DOTCLK Interface

The DOTCLK interface is another mode used in moving picture displays.

It includes the VSYNC, HSYNC, DOTCLK and (optional) ENABLE signals. The interface is popularly called the RGB interface if the ENABLE signal is present.

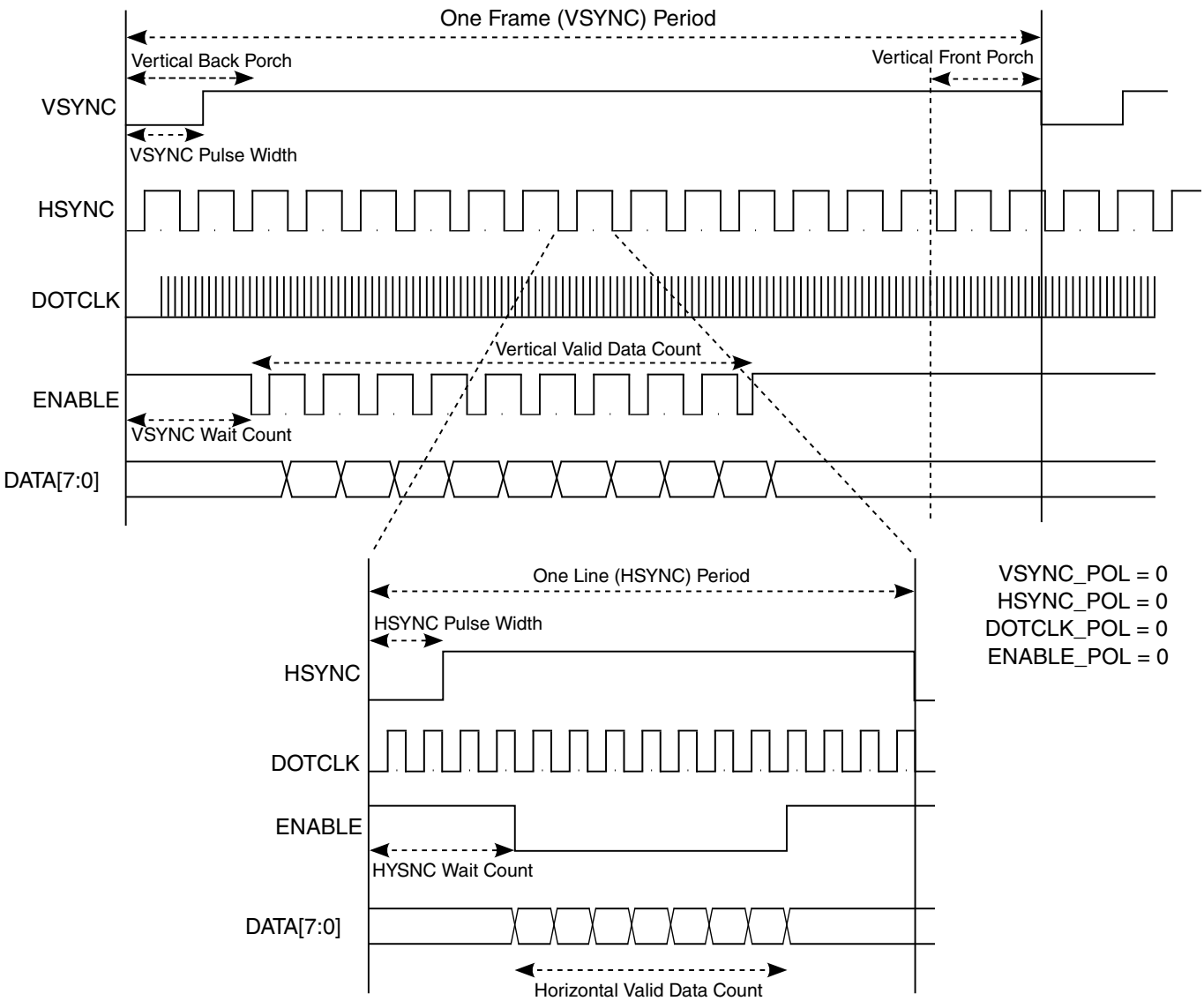


Figure 13-16. DOTCLK protocol with programmable parameters

The DOTCLK mode writes data at high speed to the LCD, and the display operation is synchronized with the VSYNC, HSYNC, ENABLE and DOTCLK signals. The polarities, periods and pulse-widths of the sync signals are programmable using the LCDIF_VDCTRL0-4 registers. The units for the VSYNC signal must be number of horizontal lines and can be selected using the VSYNC_PULSE_WIDTH_UNIT and VSYNC_PERIOD_UNIT bit fields. The VERTICAL_WAIT_CNT is by default given the same unit as the VSYNC_PERIOD. The DISPLAY CLOCK (pix_clk) frequency is managed by the CCM.

In DOTCLK mode, LCDIF_CTRL_BYPASS_COUNT bit must be set to 1. To end the current transfer, the software should make the DOTCLK_MODE bit 0, so that all data that is currently in the LCDIF LFIFO and TXFIFO is transmitted. Once that transfer is complete, the block will automatically clear the RUN bit and issue the cur_frame_done interrupt.

13.2.3.8.1 Code Example

The following code shows an example for programming a 320x240 display.

NOTE

Setting up the display must be done through the MPU mode or via SPI.

```
// Note: Common initialization steps in Initializing the LCDIF must also be
// executed along with the following code
BF_CS1 (LCDIF_CTRL, DOTCLK_MODE, 1);
BF_CS1 (LCDIF_CTRL, BYPASS_COUNT, 1); //Always for DOTCLK mode
BF_CS1 (LCDIF_VDCTRL0, VSYNC_OEB, 0); //Vsync is always an output in the DOTCLK mode
BF_CS4 (LCDIF_VDCTRL0, VSYNC_POL, 0, HSYNC_POL, 0, DOTCLK_POL, 0, ENABLE_POL, 0);
BF_CS1 (LCDIF_VDCTRL0, ENABLE_PRESENT, 1);
BF_CS2 (LCDIF_VDCTRL0, VSYNC_PERIOD_UNIT, 1, VSYNC_PULSE_WIDTH_UNIT, 1);
BF_CS1 (LCDIF_VDCTRL0, VSYNC_PULSE_WIDTH, 2);
BF_CS1 (LCDIF_VDCTRL1, VSYNC_PERIOD, 280);
BF_CS2 (LCDIF_VDCTRL2, HSYNC_PULSE_WIDTH, 10, HSYNC_PERIOD, 360); //Assuming
                                                    //  LCD_DATABUS_WIDTH is 24bit
BF_CS2 (LCDIF_VDCTRL3, VSYNC_ONLY, 0);
BF_CS2 (LCDIF_VDCTRL3, HORIZONTAL_WAIT_CNT, 20, VERTICAL_WAIT_CNT, 20);
BF_CS1 (LCDIF_VDCTRL4, DOTCLK_H_VALID_DATA_CNT, 320);//Note that DOTCLK_V_VALID_DATA_CNT is
                            //implicitly assumed to be HW_LCDIF_TRANSFER_COUNT_V_COUNT
BF_CS1 (LCDIF_VDCTRL4, SYNC_SIGNALS_ON, 1);
BF_CS1 (LCDIF_CTRL, RUN, 1);
```

To stop the transfer completely, the ideal way is to make DOTCLK_MODE = 0. In that case, the block will transmit the contents in the FIFO and reset the RUN bit.

13.2.3.9 Alpha Blending Interface

The LCDIF have the capability to add an extra overlay on the normal display buffer,LCDIF can fetch data from two buffers and combine them before display,one buffer data can have the alpha value with the RGB pixels. With LCDIF_AS_CTRL[AS_ENABLE] is set, the LCDIF will start fetching alpha surface buffer data in bus master mode and combine it with another buffer.

The LCDIF_AS_CTRL[ALPHA_CTRL] bits determines how the alpha value is constructed for the alpha surface.

13.2.3.9.1 Alpha Blending/Color Key

Regardless of pixel input format, the PS and AS pixels are normalized to 32-bits, organized as one alpha and three data bytes. Alpha blending occurs in the RGB space, if blending is required, PS pixels should be converted to RGB space. If no alpha blending is required, then YUV pixels can bypass the alpha blending ALU without color space conversion.

13.2.3.9.2 Alpha Blend

The alpha value for an individual pixel represents a mathematical weighting factor applied to the AS pixel. An alpha value of 0x00 corresponds to a transparent pixel and a value of 0xFF corresponds to an opaque pixel.

The effective alpha value for an AS pixel is determined by the AS_CTRL[ALPHA] and AS_CTRL[ALPHA_CTRL] register fields. If AS_CTRL[ALPHA_CTRL] = ALPHA_OVERRIDE , the alpha value for the pixel is taken from the AS_CTRL[ALPHA] . This can be useful for applying a constant alpha to an entire image or for image formats that don't include an alpha value. If AS_CTRL[ALPHA_CTRL] = ALPHA_MULTIPLY, the pixel's alpha value will be multiplied by the pixel's ALPHA value in order to allow scaling of the pixel's alpha or to provide better control for pixel formats such as RGB1555, which only contains a single bit of alpha.

For each color channel, the equation used to blend two source pixels is defined below:

Gá = PIO programmed global alpha (8-bit value).
Eá = Embedded alpha associated with AS pixel.
á = Gá * Eá + 0x80

The result for the red channel as an example:

R[7:0] = (á * PS.r) + ((1 - á) * AS.r)

When á is 0xff, the PS pixel will not be blended with the AS pixel, but PS will be passed as the output pixel and will not be blended with AS. In this case, AS will be discarded. Likewise, if á is 0x00 for a given pixel, PS will be loaded as the output pixel.

AS_CTRL[ALPHA_INVERT] provides the option to invert the final alpha value. This essentially inverts the effect the alpha value has on the AS and PS blending operation.

13.2.3.9.3 Color Key

The color key function is provided to create transparent effects on the output pixel.

Color keying is applied on the input pixels after they are converted to 8-bits for each red, green, and blue color channels (color keys are not applied directly to 16-bit pixel formats but to their corresponding 24-bit representation). A color key range is programmable for both PS and AS pixels. If the PS 24-bit pixel is within the PS color key range, then AS is passed through the pixel pipeline. In this case, alpha blending does NOT occur. Conversely, if PS is within the AS color key range, then PS is passed via the LCDIF data pipeline. If both PS and AS color key tests pass, then the back ground color register is passed onto following LCDIF processing components in the pipeline.

The condition for color keying to be satisfied is:

CK0.r.low <= PS.r <= CK0.r.high

CK0.g.low <= PS.g <= CK0.g.high

CK0.b.low <= PS.b <= CK0.b.high

For example, if the "red" 8-bit value for the PS pixel (or PS.r) is between the color key low and high values (CK0.r.l and CK0.r.h), the condition is true for the red color plane. When ALL three color planes meet this condition, then only the PS pixel is loaded into the output register.

To disable color keying, program the low color key register value to 0xff and the high value to 0x00. This will guarantee that the color key range test will never be true.

13.2.3.9.4 Color Key Processing (AS_CTRL)

The AS_CTRL register also contains an ENABLE_COLORKEY bit that can be used to enable or disable color key substitution for the AS.

When enabled, the pixel values are compared to the ASCOLORKEYLOW and ASCOLORKEYHIGH registers to determine if a match has occurred. When an AS pixel matches the color key range, the pixel from the AS image is considered transparent and the corresponding PS pixel is rendered. If both the PS and AS pixels match their corresponding color key ranges, the AS pixel is displayed unmodified.

AS color keys are handled in a manner similar to PS color keys. The same images used in the PS color key example could be used with the images swapped. In this case, matches on the AS image to the ASCOLORKEY register would display the PS pixels.

13.2.3.10 ITU-R BT.656 Digital Video Interface (DVI)

ITU-R BT.656 Digital Video Interface shown below transmits 4:2:2 YCbCr digital component video to a digital video encoder that can translate it into 525/60 or 625/50 analog TV signal.

Unique timing codes (timing reference signals) are embedded within the video stream to indicate the different timing events that would have been otherwise indicated by VSYNC, HSYNC and BLANK signals. The hardware supports 8-bit data transfers; the pins are shared with the lower 8 bits of LCD data bus. The LCD_RS pin is shared with the clock signal of the interface (called CCIRCLK here for uniqueness). CCIRCLK also can be obtained on the LCD_DOTCLK pin. The mode shares the write FIFO with the LCD interface and the associated pipeline. The programmable parameters in registers LCDIF_DVICTRL0-3 allow setting the total number of horizontal lines per frame, vertical and horizontal blanking interval, odd and even field start and end positions, and so on. In short, these parameters are provided to ensure that the hardware has enough flexibility to generate the right 525/60 or 625/50 data streams. Most of the initialization steps in Initializing the LCDIF such as data shifting, swizzle, and so on, are applicable to DVI mode also. The register descriptions in the programmable registers section at the end of this chapter include example code for programming the DVICTRL0-3 registers.

In DVI mode, LCDIF_CTRL_BYPASS_COUNT bit must be set to 1. To end the current transfer, the software should make the DVI_MODE bit the value 0, so that all data that is currently in the LCDIF LFIFO and TXFIFO is transmitted. Once that transfer is complete, the block will automatically clear the RUN bit and assert the cur_frame_done interrupt.

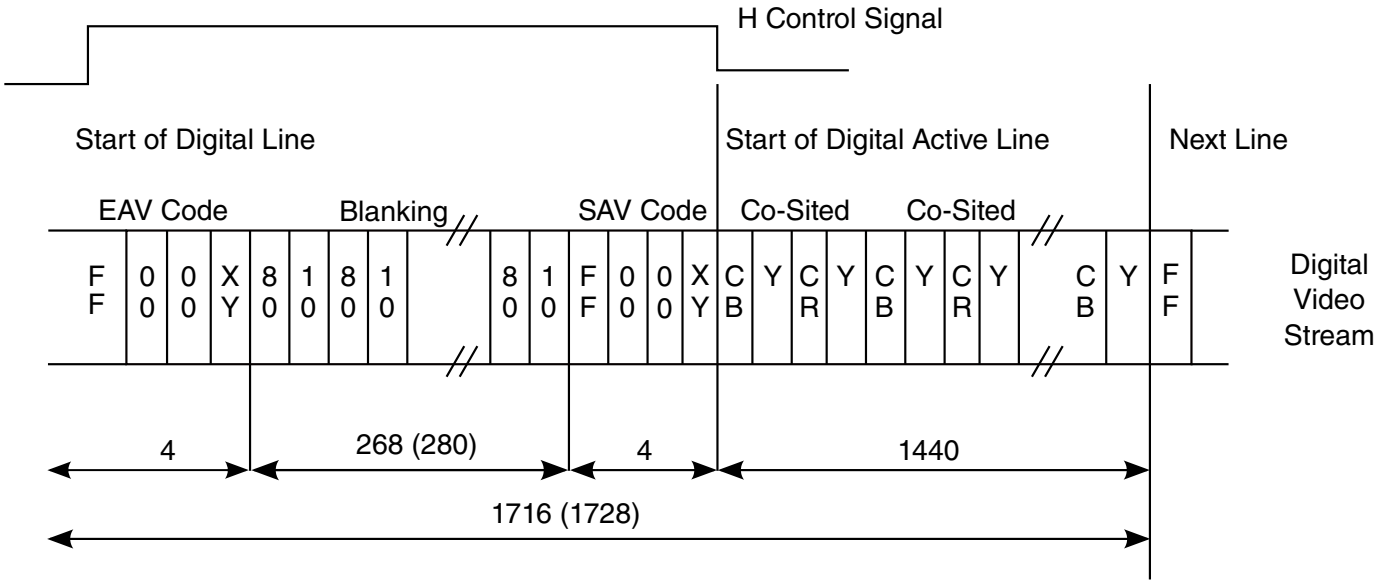


Figure 13-17. Digital Video Interface

13.2.3.11 LCDIF Pin Usage by Interface Mode

The following tables detail how the LCDIF level interface pins are used based on the desired mode of operation. The chip level I/Os should also be configured to be consistent with the desired LCDIF operating mode.

The VSYNC signal has been mapped onto two pins, LCD_BUSY and LCD_VSYNC. The pin multiplexing can be programmed to select either of those pins to function as VSYN.

NOTE

There is an option to internally mux the HSYNC, DOTCLK and ENABLE signals in the DOTCLK mode by setting the MUX_SYNC_SIGNALS bit in the VDCTRL0 register.

Table 13-5. Pin use in MPU Mode

PIN NAME	8-bit MPU LCD IF	16-bit MPU LCD IF	18-bit MPU LCD IF	24-bit MPU LCD IF
LCD_RS	LCD_ RS	LCD_ RS	LCD_ RS	LCD_RS
LCD_CS	LCD_CS	LCD_CS	LCD_ CS	LCD_CS
LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn
LCD_RD_E	LCD_RD_E	LCD_RD_E	LCD_RD_E	LCD_RD_E

Table continues on the next page...

Table 13-5. Pin use in MPU Mode (continued)

PIN NAME	8-bit MPU LCD IF	16-bit MPU LCD IF	18-bit MPU LCD IF	24-bit MPU LCD IF
LCD_VSYNC* (Two options)	X	X	X	X
LCD_HSYNC	X	X	X	X
LCD_DOTCLK	X	X	X	X
LCD_ENABLE	X	X	X	X
LCD_DATA23 (LCD_D23)	X	X	X	LCD_DATA23
LCD_DATA22 (LCD_D22)	X	X	X	LCD_DATA22
LCD_DATA21 (LCD_D21)	X	X	X	LCD_DATA21
LCD_DATA20 (LCD_D20)	X	X	X	LCD_DATA20
LCD_DATA19 (LCD_D19)	X	X	X	LCD_DATA19
LCD_DATA18 (LCD_D18)	X	X	X	LCD_DATA18
LCD_DATA17 (LCD_D17)	X	X	LCD_DATA17	LCD_DATA17
LCD_DATA16 (LCD_D16)	X	X	LCD_DATA16	LCD_DATA16
LCD_DATA15 (LCD_D15) / VSYNC*	X	LCD_DATA15	LCD_DATA15	LCD_DATA15
LCD_DATA14 (LCD_D14) / HSYNC**	X	LCD_DATA14	LCD_DATA14	LCD_DATA14
LCD_DATA13 (LCD_D13) / LCD_DOTCLK**	X	LCD_DATA13	LCD_DATA13	LCD_DATA13
LCD_DATA12 (LCD_D12) / ENABLE**	X	LCD_DATA12	LCD_DATA12	LCD_DATA12
LCD_DATA11 (LCD_D11)	X	LCD_DATA11	LCD_DATA11	LCD_DATA11
LCD_DATA10 (LCD_D10)	X	LCD_DATA10	LCD_DATA10	LCD_DATA10
LCD_DATA09 (LCD_D9)	X	LCD_DATA09	LCD_DATA09	LCD_DATA09
LCD_DATA08 (LCD_D8)	X	LCD_DATA08	LCD_DATA08	LCD_DATA08
LCD_DATA07 (LCD_D7)	LCD_DATA07	LCD_DATA07	LCD_DATA07	LCD_DATA07
LCD_DATA06 (LCD_D6)	LCD_DATA06	LCD_DATA06	LCD_DATA06	LCD_DATA06

Table continues on the next page...

Table 13-5. Pin use in MPU Mode (continued)

PIN NAME	8-bit MPU LCD IF	16-bit MPU LCD IF	18-bit MPU LCD IF	24-bit MPU LCD IF
LCD_DATA05 (LCD_D5)	LCD_DATA05	LCD_DATA05	LCD_DATA05	LCD_DATA05
LCD_DATA04 (LCD_D4)	LCD_DATA04	LCD_DATA04	LCD_DATA04	LCD_DATA04
LCD_DATA03 (LCD_D3)	LCD_DATA03	LCD_DATA03	LCD_DATA03	LCD_DATA03
LCD_DATA02 (LCD_D2)	LCD_DATA02	LCD_DATA02	LCD_DATA02	LCD_DATA02
LCD_DATA01 (LCD_D1)	LCD_DATA01	LCD_DATA01	LCD_DATA01	LCD_DATA01
LCD_DATA00 (LCD_D0)	LCD_DATA00	LCD_DATA00	LCD_DATA00	LCD_DATA00
LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET
LCD_BUSY / LCD_VSYNC	LCD_BUSY	LCD_BUSY	LCD_BUSY	LCD_BUSY

Table 13-6. Pin use in VSYNC Mode

PIN NAME	8-bit VSYNC LCD IF	16-bit VSYNC LCD IF	18-bit VSYNC LCD IF	24-bit VSYNC LCD IF
LCD_RS	LCD_RS	LCD_ RS	LCD_RS	LCD_RS
LCD_CS	LCD_CS	LCD_CS	LCD_ CS	LCD_CS
LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn	LCD_WR _RWn
LCD_RD_E	LCD_RD_E	LCD_RD_E	LCD_RD_E	LCD_RD_E
LCD_VSYNC* (Two options)	LCD_ VSYNC	LCD_ VSYNC	LCD_ VSYNC	LCD_ VSYNC
LCD_HSYNC	X	X	X	X
LCD_DOTCLK	X	X	X	X
LCD_ENABLE	X	X	X	X
LCD_DATA23 (LCD_D23)	X	X	X	LCD_DATA23
LCD_DATA22 (LCD_D22)	X	X	X	LCD_DATA22
LCD_DATA21 (LCD_D21)	X	X	X	LCD_DATA21
LCD_DATA20 (LCD_D20)	X	X	X	LCD_DATA20
LCD_DATA19 (LCD_D19)	X	X	X	LCD_DATA19
LCD_DATA18 (LCD_D18)	X	X	X	LCD_DATA18
LCD_DATA17 (LCD_D17)	X	X	LCD_DATA17	LCD_DATA17

Table continues on the next page...

Table 13-6. Pin use in VSYNC Mode (continued)

PIN NAME	8-bit VSYNC LCD IF	16-bit VSYNC LCD IF	18-bit VSYNC LCD IF	24-bit VSYNC LCD IF
LCD_DATA16 (LCD_D16)	X	X	LCD_DATA16	LCD_DATA16
LCD_DATA15 (LCD_D15) / VSYNC*	VSYNC (optional)	LCD_DATA15	VSYNC (optional)	LCD_DATA15
LCD_DATA14 (LCD_D14) / HSYNC**	X	LCD_DATA14	X	LCD_DATA14
LCD_DATA13 (LCD_D13) / LCD_DOTCLK**	X	LCD_DATA13	X	LCD_DATA13
LCD_DATA12 (LCD_D12) / ENABLE**	X	LCD_DATA12	X	LCD_DATA12
LCD_DATA11 (LCD_D11)	X	LCD_DATA11	X	LCD_DATA11
LCD_DATA10 (LCD_D10)	X	LCD_DATA10	X	LCD_DATA10
LCD_DATA09 (LCD_D9)	X	LCD_DATA09	X	LCD_DATA09
LCD_DATA08 (LCD_D8)	X	LCD_DATA08	X	LCD_DATA08
LCD_DATA07 (LCD_D7)	LCD_DATA07	LCD_DATA07	LCD_DATA07	LCD_DATA07
LCD_DATA06 (LCD_D6)	LCD_DATA06	LCD_DATA06	LCD_DATA06	LCD_DATA06
LCD_DATA05 (LCD_D5)	LCD_DATA05	LCD_DATA05	LCD_DATA05	LCD_DATA05
LCD_DATA04 (LCD_D4)	LCD_DATA04	LCD_DATA04	LCD_DATA04	LCD_DATA04
LCD_DATA03 (LCD_D3)	LCD_DATA03	LCD_DATA03	LCD_DATA03	LCD_DATA03
LCD_DATA02 (LCD_D2)	LCD_DATA02	LCD_DATA02	LCD_DATA02	LCD_DATA02
LCD_DATA01 (LCD_D1)	LCD_DATA01	LCD_DATA01	LCD_DATA01	LCD_DATA01
LCD_DATA00 (LCD_D0)	LCD_DATA00	LCD_DATA00	LCD_DATA00	LCD_DATA00
LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET
LCD_BUSY / LCD_VSYNC	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)

Table 13-7. Pin use in DOTCLK Mode

PIN NAME	8-bit DOTCLK LCD IF	16-bit DOTCLK LCD IF	18-bit DOTCLK LCD IF	24-bit DOTCLK LCD IF
LCD_VSYNC	LCD_VSYNC	LCD_VSYNC	LCD_VSYNC	LCD_VSYNC
LCD_HSYNC	LCD_HSYNC	LCD_HSYNC	LCD_HSYNC	LCD_HSYNC

Table continues on the next page...

Table 13-7. Pin use in DOTCLK Mode (continued)

PIN NAME	8-bit DOTCLK LCD IF	16-bit DOTCLK LCD IF	18-bit DOTCLK LCD IF	24-bit DOTCLK LCD IF
LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK	LCD_DOTCLK
LCD_ENABLE	LCD_ENABLE	LCD_ENABLE	LCD_ENABLE	LCD_ENABLE
LCD_DATA23 (LCD_D23)	X	X	X	LCD_DATA23
LCD_DATA22 (LCD_D22)	X	X	X	LCD_DATA22
LCD_DATA21 (LCD_D21)	X	X	X	LCD_DATA21
LCD_DATA20 (LCD_D20)	X	X	X	LCD_DATA20
LCD_DATA19 (LCD_D19)	X	X	X	LCD_DATA19
LCD_DATA18 (LCD_D18)	X	X	X	LCD_DATA18
LCD_DATA17 (LCD_D17)	X	X	LCD_DATA17	LCD_DATA17
LCD_DATA16 (LCD_D16)	X	X	LCD_DATA16	LCD_DATA16
LCD_DATA15 (LCD_D15)	X	LCD_DATA15	LCD_DATA15	LCD_DATA15
LCD_DATA14 (LCD_D14)	X	LCD_DATA14	LCD_DATA14	LCD_DATA14
LCD_DATA13 (LCD_D13)	X	LCD_DATA13	LCD_DATA13	LCD_DATA13
LCD_DATA12 (LCD_D12)	X	LCD_DATA12	LCD_DATA12	LCD_DATA12
LCD_DATA11 (LCD_D11)	X	LCD_DATA11	LCD_DATA11	LCD_DATA11
LCD_DATA10 (LCD_D10)	X	LCD_DATA10	LCD_DATA10	LCD_DATA10
LCD_DATA09 (LCD_D9)	X	LCD_DATA09	LCD_DATA09	LCD_DATA09
LCD_DATA08 (LCD_D8)	X	LCD_DATA08	LCD_DATA08	LCD_DATA08
LCD_DATA07 (LCD_D7)	LCD_DATA07	LCD_DATA07	LCD_DATA07	LCD_DATA07
LCD_DATA06 (LCD_D6)	LCD_DATA06	LCD_DATA06	LCD_DATA06	LCD_DATA06
LCD_DATA05 (LCD_D5)	LCD_DATA05	LCD_DATA05	LCD_DATA05	LCD_DATA05
LCD_DATA04 (LCD_D4)	LCD_DATA04	LCD_DATA04	LCD_DATA04	LCD_DATA04
LCD_DATA03 (LCD_D3)	LCD_DATA03	LCD_DATA03	LCD_DATA03	LCD_DATA03

Table continues on the next page...

Table 13-7. Pin use in DOTCLK Mode (continued)

PIN NAME	8-bit DOTCLK LCD IF	16-bit DOTCLK LCD IF	18-bit DOTCLK LCD IF	24-bit DOTCLK LCD IF
LCD_DATA02 (LCD_D2)	LCD_DATA02	LCD_DATA02	LCD_DATA02	LCD_DATA02
LCD_DATA01 (LCD_D1)	LCD_DATA01	LCD_DATA01	LCD_DATA01	LCD_DATA01
LCD_DATA00 (LCD_D0)	LCD_DATA00	LCD_DATA00	LCD_DATA00	LCD_DATA00
LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET	LCD_RESET
LCD_BUSY / LCD_VSYNC	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)	LCD_BUSY (OR optional LCD_VSYNC)

Table 13-8. Pin use in DVI Mode

PIN NAME	8-bit DVI LCD IF
LCD_RS	CCIR_CLK
LCD_CS	X
LCD_WR_RWn	X
LCD_RD_E	X
LCD_VSYNC* (Two options)	X
LCD_HSYNC	X
LCD_DOTCLK	X
LCD_ENABLE	X
LCD_DATA23 (LCD_D23)	X
LCD_DATA22 (LCD_D22)	X
LCD_DATA21 (LCD_D21)	X
LCD_DATA20 (LCD_D20)	X
LCD_DATA19 (LCD_D19)	X
LCD_DATA18 (LCD_D18)	X
LCD_DATA17 (LCD_D17)	X
LCD_DATA16 (LCD_D16)	X
LCD_DATA15 (LCD_D15) / VSYNC*	X
LCD_DATA14 (LCD_D14) / HSYNC**	X
LCD_DATA13 (LCD_D13) / LCD_DOTCLK**	X
LCD_DATA12 (LCD_D12) / ENABLE**	X

Table continues on the next page...

Table 13-8. Pin use in DVI Mode (continued)

PIN NAME	8-bit DVI LCD IF
LCD_DATA11 (LCD_D11)	X
LCD_DATA10 (LCD_D10)	X
LCD_DATA09 (LCD_D9)	X
LCD_DATA08 (LCD_D8)	X
LCD_DATA07 (LCD_D7)	LCD_DATA07
LCD_DATA06 (LCD_D6)	LCD_DATA06
LCD_DATA05 (LCD_D5)	LCD_DATA05
LCD_DATA04 (LCD_D4)	LCD_DATA04
LCD_DATA03 (LCD_D3)	LCD_DATA03
LCD_DATA02 (LCD_D2)	LCD_DATA02
LCD_DATA01 (LCD_D1)	LCD_DATA01
LCD_DATA00 (LCD_D0)	LCD_DATA00
LCD_RESET	X
LCD_BUSY / LCD_VSYNC	X

13.2.4 Behavior During Reset

BUS CLOCK (apb_clk) and DISPLAY CLOCK (pix_clk) must be running before making any changes to SFTRST or CLKGATE bits.

A soft reset (SFTRST) can take multiple clock periods to complete, so do not set CLKGATE when setting SFTRST.

The reset process gates the clocks automatically.

13.2.5 LCDIF Memory Map/Register Definition

Some of the LCDIF registers (XXX_SET, XXX_CLR, and XXX_TOG) allow direct bit field masking and access.

- When writing 1 to XXX_SET bit fields, these registers allow setting the masked 1 bit fields, while keeping unchanged all bit fields which remain on 0 logic state.

- When writing 1 to XXX_CLR bit fields, these registers allow clearing the masked 1 bit fields, while keeping unchanged all other bit fields which remained on 0 logic state.
- When writing 1 to XXX_TOG bit fields, these registers allow inverting the logic state of all masked 1 bit fields, while they keep unchanged the remaining bit fields which were kept on 0 logic state.

LCDIF Hardware Register Format Summary

LCDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3032_0000	LCDIF General Control Register (LCDIF_CTRL)	32	R/W	C000_0000h	13.2.5.1/ 3715
3032_0004	LCDIF General Control Register (LCDIF_CTRL_SET)	32	R/W	C000_0000h	13.2.5.1/ 3715
3032_0008	LCDIF General Control Register (LCDIF_CTRL_CLR)	32	R/W	C000_0000h	13.2.5.1/ 3715
3032_000C	LCDIF General Control Register (LCDIF_CTRL_TOG)	32	R/W	C000_0000h	13.2.5.1/ 3715
3032_0010	LCDIF General Control1 Register (LCDIF_CTRL1)	32	R/W	000F_0000h	13.2.5.2/ 3718
3032_0014	LCDIF General Control1 Register (LCDIF_CTRL1_SET)	32	R/W	000F_0000h	13.2.5.2/ 3718
3032_0018	LCDIF General Control1 Register (LCDIF_CTRL1_CLR)	32	R/W	000F_0000h	13.2.5.2/ 3718
3032_001C	LCDIF General Control1 Register (LCDIF_CTRL1_TOG)	32	R/W	000F_0000h	13.2.5.2/ 3718
3032_0020	LCDIF General Control2 Register (LCDIF_CTRL2)	32	R/W	0020_0000h	13.2.5.3/ 3720
3032_0024	LCDIF General Control2 Register (LCDIF_CTRL2_SET)	32	R/W	0020_0000h	13.2.5.3/ 3720
3032_0028	LCDIF General Control2 Register (LCDIF_CTRL2_CLR)	32	R/W	0020_0000h	13.2.5.3/ 3720
3032_002C	LCDIF General Control2 Register (LCDIF_CTRL2_TOG)	32	R/W	0020_0000h	13.2.5.3/ 3720
3032_0030	LCDIF Horizontal and Vertical Valid Data Count Register (LCDIF_TRANSFER_COUNT)	32	R/W	0001_0000h	13.2.5.4/ 3723
3032_0040	LCD Interface Current Buffer Address Register (LCDIF_CUR_BUF)	32	R/W	0000_0000h	13.2.5.5/ 3723
3032_0050	LCD Interface Next Buffer Address Register (LCDIF_NEXT_BUF)	32	R/W	0000_0000h	13.2.5.6/ 3724
3032_0060	LCD Interface Timing Register (LCDIF_TIMING)	32	R/W	0000_0000h	13.2.5.7/ 3724

Table continues on the next page...

LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3032_0070	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0)	32	R/W	0000_0000h	13.2.5.8/ 3725
3032_0074	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_SET)	32	R/W	0000_0000h	13.2.5.8/ 3725
3032_0078	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_CLR)	32	R/W	0000_0000h	13.2.5.8/ 3725
3032_007C	LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0_TOG)	32	R/W	0000_0000h	13.2.5.8/ 3725
3032_0080	LCDIF VSYNC Mode and Dotclk Mode Control Register1 (LCDIF_VDCTRL1)	32	R/W	0000_0000h	13.2.5.9/ 3726
3032_0090	LCDIF VSYNC Mode and Dotclk Mode Control Register2 (LCDIF_VDCTRL2)	32	R/W	0000_0000h	13.2.5.10/ 3727
3032_00A0	LCDIF VSYNC Mode and Dotclk Mode Control Register3 (LCDIF_VDCTRL3)	32	R/W	0000_0000h	13.2.5.11/ 3727
3032_00B0	LCDIF VSYNC Mode and Dotclk Mode Control Register4 (LCDIF_VDCTRL4)	32	R/W	0000_0000h	13.2.5.12/ 3728
3032_00C0	Digital Video Interface Control0 Register (LCDIF_DVICTRL0)	32	R/W	0000_0000h	13.2.5.13/ 3729
3032_00D0	Digital Video Interface Control1 Register (LCDIF_DVICTRL1)	32	R/W	0000_0000h	13.2.5.14/ 3730
3032_00E0	Digital Video Interface Control2 Register (LCDIF_DVICTRL2)	32	R/W	0000_0000h	13.2.5.15/ 3731
3032_00F0	Digital Video Interface Control3 Register (LCDIF_DVICTRL3)	32	R/W	0000_0000h	13.2.5.16/ 3732
3032_0100	Digital Video Interface Control4 Register (LCDIF_DVICTRL4)	32	R/W	0000_0000h	13.2.5.17/ 3733
3032_0110	RGB to YCbCr 4:2:2 CSC Coefficient0 Register (LCDIF_CSC_COEFF0)	32	R/W	0000_0000h	13.2.5.18/ 3734
3032_0120	RGB to YCbCr 4:2:2 CSC Coefficient1 Register (LCDIF_CSC_COEFF1)	32	R/W	0000_0000h	13.2.5.19/ 3735
3032_0130	RGB to YCbCr 4:2:2 CSC Coefficent2 Register (LCDIF_CSC_COEFF2)	32	R/W	0000_0000h	13.2.5.20/ 3735
3032_0140	RGB to YCbCr 4:2:2 CSC Coefficient3 Register (LCDIF_CSC_COEFF3)	32	R/W	0000_0000h	13.2.5.21/ 3736
3032_0150	RGB to YCbCr 4:2:2 CSC Coefficient4 Register (LCDIF_CSC_COEFF4)	32	R/W	0000_0000h	13.2.5.22/ 3737
3032_0160	RGB to YCbCr 4:2:2 CSC Offset Register (LCDIF_CSC_OFFSET)	32	R/W	0080_0010h	13.2.5.23/ 3738
3032_0170	RGB to YCbCr 4:2:2 CSC Limit Register (LCDIF_CSC_LIMIT)	32	R/W	00FF_00FFh	13.2.5.24/ 3738
3032_0180	LCD Interface Data Register (LCDIF_DATA)	32	R/W	0000_0000h	13.2.5.25/ 3739
3032_0190	Bus Master Error Status Register (LCDIF_BM_ERROR_STAT)	32	R/W	0000_0000h	13.2.5.26/ 3740

Table continues on the next page...

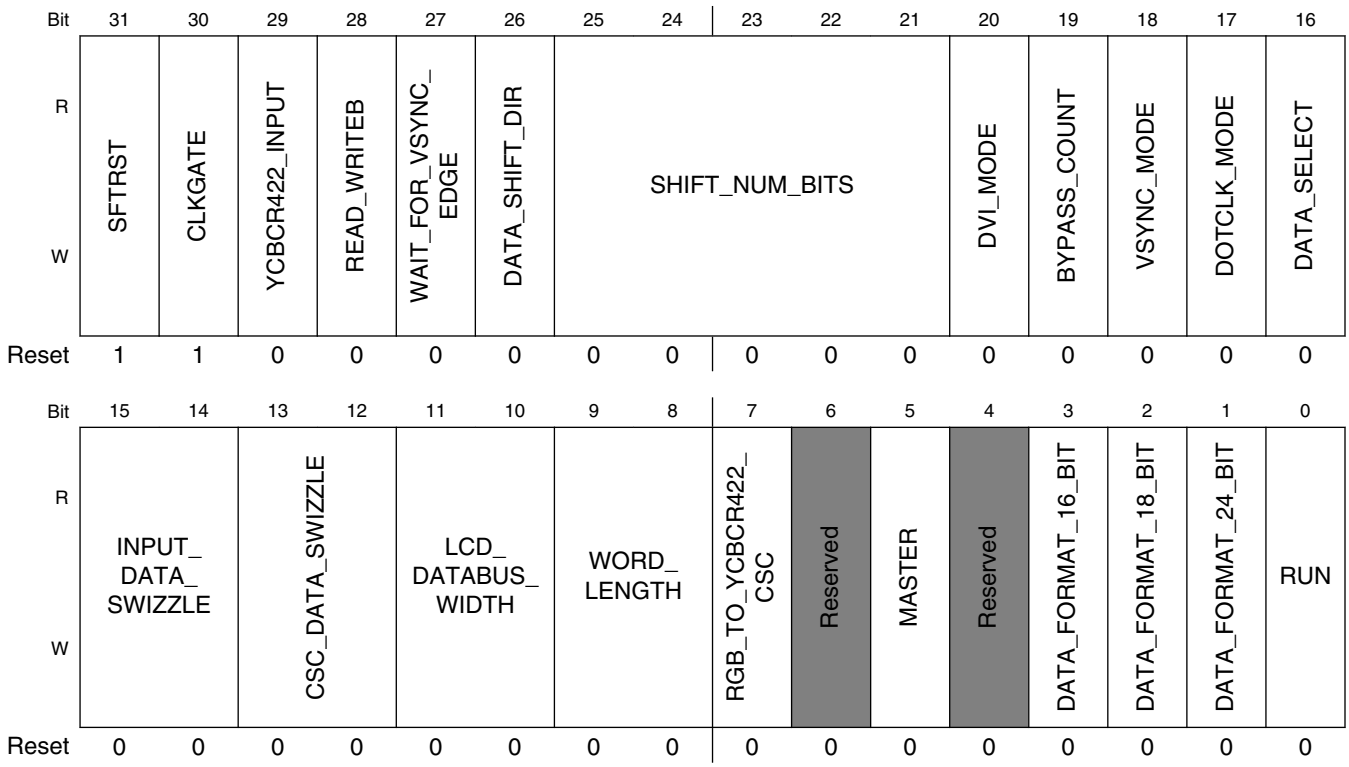
LCDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3032_01A0	CRC Status Register (LCDIF_CRC_STAT)	32	R/W	0000_0000h	13.2.5.27/ 3740
3032_01B0	LCD Interface Status Register (LCDIF_STAT)	32	R	9500_0000h	13.2.5.28/ 3741
3032_0200	LCDIF Threshold Register (LCDIF_THRES)	32	R/W	0100_000Fh	13.2.5.29/ 3743
3032_0210	LCDIF AS Buffer Control Register (LCDIF_AS_CTRL)	32	R/W	0000_0000h	13.2.5.30/ 3744
3032_0220	Alpha Surface Buffer Pointer (LCDIF_AS_BUF)	32	R/W	0000_0000h	13.2.5.31/ 3746
3032_0230	LCDIF_AS_NEXT_BUF	32	R/W	0000_0000h	13.2.5.32/ 3747
3032_0240	LCDIF Overlay Color Key Low (LCDIF_AS_CLRKEYLOW)	32	R/W	00FF_FFFFh	13.2.5.33/ 3747
3032_0250	LCDIF Overlay Color Key High (LCDIF_AS_CLRKEYHIGH)	32	R/W	0000_0000h	13.2.5.34/ 3748
3032_0260	LCD working insync mode with CSI for VSYNC delay (LCDIF_SYNC_DELAY)	32	R/W	0000_0000h	13.2.5.35/ 3748

13.2.5.1 LCDIF General Control Register (LCDIF_CTRL*n*)

The LCD Interface Control Register provides overall control of the LCDIF block. The LCDIF Control Register provides a variety of control functions to the programmer. These functions allow the interface to be very flexible to work with a variety of LCD controllers, and to minimize overhead and increase performance of LCD programming. The register has been organized such that switching between the different LCD modes can be done with minimum PIO writes.

Address: 3032_0000h base + 0h offset + (4d × i), where i=0d to 3d



LCDIF_CTRL*n* field descriptions

Field	Description
31 SFTRST	This bit must be set to zero to enable normal operation of the LCDIF. When set to one, it forces a block level reset.
30 CLKGATE	This bit must be set to zero for normal operation. When set to one it gates off the clocks to the block.
29 YCBCR422_INPUT	Zero implies input data is in RGB color space. One implies input data is in YCbCr 4:2:2 format, such that YCbYCr are packed in a 32-bit word. It also means that there are 2 pixels in 4 bytes. If this bit is set, software should program the H_COUNT field in the TRANSFER_COUNT register to the total number of pixels that will have to be fetched by the LCDIF block per line and the BYTE_PACKING_FORMAT should be 0xF. The WORD_LENGTH does not matter in this case.

Table continues on the next page...

LCDIF_CTRL*n* field descriptions (continued)

Field	Description
28 READ_WRITEB	By default, LCDIF is in the write mode. Setting this bit to 1 will make the hardware go into 6800/8080 MPU read mode. The LCDIF_MASTER bit must be 0, since bus master mode can only be used for writing the display.
27 WAIT_FOR_ VSYNC_EDGE	Setting this bit to 1 will make the hardware wait for the triggering VSYNC edge before starting write transfers to the LCD. Used only in the VSYNC mode of operation.
26 DATA_SHIFT_ DIR	Use this bit to determine the direction of shift of transmit data. In the DVI mode, it works only on the active data, not on the timing codes and ancillary data. 0x0 **TXDATA_SHIFT_LEFT** — Data to be transmitted is shifted LEFT by SHIFT_NUM_BITS bits. 0x1 **TXDATA_SHIFT_RIGHT** — Data to be transmitted is shifted RIGHT by SHIFT_NUM_BITS bits.
25–21 SHIFT_NUM_ BITS	The data to be transmitted is shifted left or right by this number of bits.
20 DVI_MODE	Set this bit to 1 to get into the ITU-R BT.656 digital video interface mode. Toggle this bit from 1 to 0 to make the hardware go out of DVI mode after completing all data transfer and after the RUN bit has been deasserted.
19 BYPASS_ COUNT	When this bit is 0, it means that LCDIF will stop the block operation and turn off the RUN bit after the amount of data indicated by the LCDIF_TRANSFER_COUNT register has been transferred out. When this bit is set to 1, the block will continue normal operation indefinitely until it is told to stop. This bit must be 0 in MPU and VSYNC modes, and must be 1 in DOTCLK and DVI modes of operation.
18 VSYNC_MODE	Setting this bit to 1 will make the LCDIF hardware go into VSYNC mode. WAIT_FOR_VSYNC_EDGE can be used only if this bit is set. If VSYNC signal is required to be an output from the block, SYNC_SIGNALS_ON bit in LCDIF_VDCTRL4 register must be set.
17 DOTCLK_MODE	Set this bit to 1 to make the hardware go into the DOTCLK mode, i.e. VSYNC/HSYNC/DOTCLK/ENABLE interface mode. ENABLE is optional, selected by the ENABLE_PRESENT bit. Toggle this bit from 1 to 0 to make the hardware go out of DOTCLK mode after completing all data transfer and deasserting the RUN bit.
16 DATA_SELECT	Command Mode polarity bit. This bit should only be changed when RUN is 0. 0x0 **CMD_MODE** — Command Mode. LCD_RS signal is Low. 0x1 **DATA_MODE** — Data Mode. LCD_RS signal is High.
15–14 INPUT_DATA_ SWIZZLE	This field specifies how to swap the bytes fetched by the bus master interface. The swizzle function is independent of the WORD_LENGTH bit. The supported swizzle configurations are: 0x0 **NO_SWAP** — No byte swapping.(Little endian) 0x0 **LITTLE_ENDIAN** — Little Endian byte ordering (same as NO_SWAP). 0x1 **BIG_ENDIAN_SWAP** — Big Endian swap (swap bytes 0,3 and 1,2). 0x1 **SWAP_ALL_BYTES** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
13–12 CSC_DATA_ SWIZZLE	This field specifies how to swap the bytes after the data has been converted into an internal representation of 24 bits per pixel and before it is transmitted over the LCD interface bus. The data is always transmitted with the least significant byte/hword (half word) first after the swizzle takes place. So, INPUT_DATA_SWIZZLE takes place first on the incoming data, and then CSC_DATA_SWIZZLE is applied. The swizzle function is independent of the WORD_LENGTH or the LCD_DATABUS_WIDTH fields. If RGB_TO_YCRCB422_CSC bit is set, the swizzle occurs on the Y, Cb, Cr values. The supported swizzle configurations are: 0x0 **NO_SWAP** — No byte swapping.(Little endian)

Table continues on the next page...

LCDIF_CTRL*n* field descriptions (continued)

Field	Description
	0x0 **LITTLE_ENDIAN** — Little Endian byte ordering (same as NO_SWAP). 0x1 **BIG_ENDIAN_SWAP** — Big Endian swap (swap bytes 0,3 and 1,2). 0x1 **SWAP_ALL_BYTES** — Swizzle all bytes, swap bytes 0,3 and 1,2 (aka Big Endian). 0x2 **HWD_SWAP** — Swap half-words. 0x3 **HWD_BYTE_SWAP** — Swap bytes within each half-word.
11–10 LCD_DATABUS_WIDTH	LCD Data bus transfer width. 0x0 **16_BIT** — 16-bit data bus mode. 0x1 **8_BIT** — 8-bit data bus mode. 0x2 **18_BIT** — 18-bit data bus mode. 0x3 **24_BIT** — 24-bit data bus mode.
9–8 WORD_LENGTH	Input data format. 0x0 **16_BIT** — Input data is 16 bits per pixel. 0x1 **8_BIT** — Input data is 8 bits wide. 0x2 **18_BIT** — Input data is 18 bits per pixel. 0x3 **24_BIT** — Input data is 24 bits per pixel.
7 RGB_TO_YCBCR422_CSC	Set this bit to 1 to enable conversion from RGB to YCbCr colorspace. See the LCDIF_CSC_ registers for further details.
6 -	This field is reserved. Reserved
5 MASTER	Set this bit to make the LCDIF act as a bus master.
4 RSRVD0	This field is reserved. Reserved bits. Write as 0.
3 DATA_FORMAT_16_BIT	When this bit is 1 and WORD_LENGTH = 0, it implies that the 16-bit data is in ARGB555 format. When this bit is 0 and WORD_LENGTH = 0, it implies that the 16-bit data is in RGB565 format. When WORD_LENGTH is not 0, this bit does not care.
2 DATA_FORMAT_18_BIT	Used only when WORD_LENGTH = 2, i.e. 18-bit. 0x0 **LOWER_18_BITS_VALID** — Data input to the block is in 18 bpp format, such that lower 18 bits contain RGB 666 and upper 14 bits do not contain any useful data. 0x1 **UPPER_18_BITS_VALID** — Data input to the block is in 18 bpp format, such that upper 18 bits contain RGB 666 and lower 14 bits do not contain any useful data.
1 DATA_FORMAT_24_BIT	Used only when WORD_LENGTH = 3, i.e. 24-bit. Note that this applies to both packed and unpacked 24-bit data. 0x0 **ALL_24_BITS_VALID** — Data input to the block is in 24 bpp format, such that all RGB 888 data is contained in 24 bits. 0x1 **DROP_UPPER_2_BITS_PER_BYTE** — Data input to the block is actually RGB 18 bpp, but there is 1 color per byte, hence the upper 2 bits in each byte do not contain any useful data, and should be dropped.
0 RUN	When this bit is set by software, the LCDIF will begin transferring data between the SoC and the display. This bit must remain set until the operation is complete.

13.2.5.2 LCDIF General Control1 Register (LCDIF_CTRL1*n*)

The LCDIF Control Register provides overall control of the LCDIF block.

The LCDIF Control1 Register provides additional programming to the LCDIF. It implements some bits which are unlikely to change often in a particular application. It also carries interrupt-related bits which are common across more than one mode of operation.

Address: 3032_0000h base + 10h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved	Reserved	Reserved		COMBINE_MPU_WR_STRB	BM_ERROR_IRQ_EN	BM_ERROR_IRQ	RECOVER_ON_UNDERFLOW	INTERLACE_FIELDS	START_INTERLACE_FROM_SECOND_FIELD	FIFO_CLEAR	IRQ_ON_ALTERNATE_FIELDS	BYTE_PACKING_FORMAT			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	OVERFLOW_IRQ_EN	UNDERFLOW_IRQ_EN	CUR_FRAME_DONE_IRQ_EN	VSYNC_EDGE_IRQ_EN	OVERFLOW_IRQ	UNDERFLOW_IRQ	CUR_FRAME_DONE_IRQ	VSYNC_EDGE_IRQ	Reserved					BUSY_ENABLE	MODE86	RESET
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CTRL1*n* field descriptions

Field	Description
31 -	This field is reserved. Reserved bits. Write as 0.
30 -	This field is reserved. Reserved bits. Write as 0.
29–28 -	This field is reserved. Reserved bits. Write as 0.
27 COMBINE_MPU_WR_STRB	If this bit is not set, the write strobe will be driven on LCD_WR_RWn pin in the 8080 mode and on the LCD_RD_E pin in the 6800 mode. If it is set, the write strobe of both the 6800 and 8080 modes will be driven only on the LCD_WR_RWn pin. Note that this does not work for read strobe.
26 BM_ERROR_IRQ_EN	This bit is set to enable bus master error interrupt in the LCDIF master mode.

Table continues on the next page...

LCDIF_CTRL1*n* field descriptions (continued)

Field	Description
25 BM_ERROR_ IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. This bit will be set when the LCDIF is in master mode and an error response was returned by the slave. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
24 RECOVER_ON_ UNDERFLOW	Set this bit to enable the LCDIF block to recover in the next field/frame if there was an underflow in the current field/frame.
23 INTERLACE_ FIELDS	Set this bit if it is required that the LCDIF block fetches odd lines in one field and even lines in the other field. It will work only in LCDIF_MASTER is set to 1.
22 START_ INTERLACE_ FROM_ SECOND_FIELD	The default is to grab the odd lines first and then the even lines. Set this bit if it is required to grab the even lines first and then the odd lines. (Line numbers start from 1, so odd lines are 1,3,5,etc. and even lines are 2,4,6, etc.)
21 FIFO_CLEAR	Set this bit to clear all the data in the latency FIFO (LFIFO), TXFIFO and the RXFIFO.
20 IRQ_ON_ ALTERNATE_ FIELDS	If this bit is set, the LCDIF block will assert the cur_frame_done interrupt only on alternate fields, otherwise it will issue the interrupt on both odd and even field. This bit is mostly relevant if INTERLACE_FIELDS is set.
19–16 BYTE_ PACKING_ FORMAT	This bitfield is used to show which data bytes in a 32-bit word are valid. Default value 0xf indicates that all bytes are valid. For 8-bit transfers, any combination in this bitfield will mean valid data is present in the corresponding bytes. In the 16-bit mode, a 16-bit half-word is valid only if adjacent bits [1:0] or [3:2] or both are 1. A value of 0x0 will mean that none of the bytes are valid and should not be used. For example, set the bit field value to 0x7 if the display data is arranged in the 24-bit unpacked format (A-R-G-B where A value does not have be transmitted). When input data is in YCbCr 4:2:2 format (YCBCR422_INPUT is 1), H_COUNT should be the number of pixels that should be fetched by the block and the BYTE_PACKING_FORMAT should be 0xF.(Note - YCBCR422_INPUT = 1 implies 2 pixels per 32 bits).
15 OVERFLOW_ IRQ_EN	This bit is set to enable an overflow interrupt in the TXFIFO in the write mode.
14 UNDERFLOW_ IRQ_EN	This bit is set to enable an underflow interrupt in the TXFIFO in the write mode.
13 CUR_FRAME_ DONE_IRQ_EN	This bit is set to 1 enable an interrupt every time the hardware enters in the vertical blanking state.
12 VSYNC_EDGE_ IRQ_EN	This bit is set to enable an interrupt every time the hardware encounters the leading VSYNC edge in the VSYNC and DOTCLK modes, or the beginning of every field in DVI mode.
11 OVERFLOW_ IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. A latency FIFO (LFIFO) overflow in the write mode (MPU/VSYNC/DOTCLK/DVI mode) was detected, data samples have been lost. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.

Table continues on the next page...

LCDIF_CTRL1*n* field descriptions (continued)

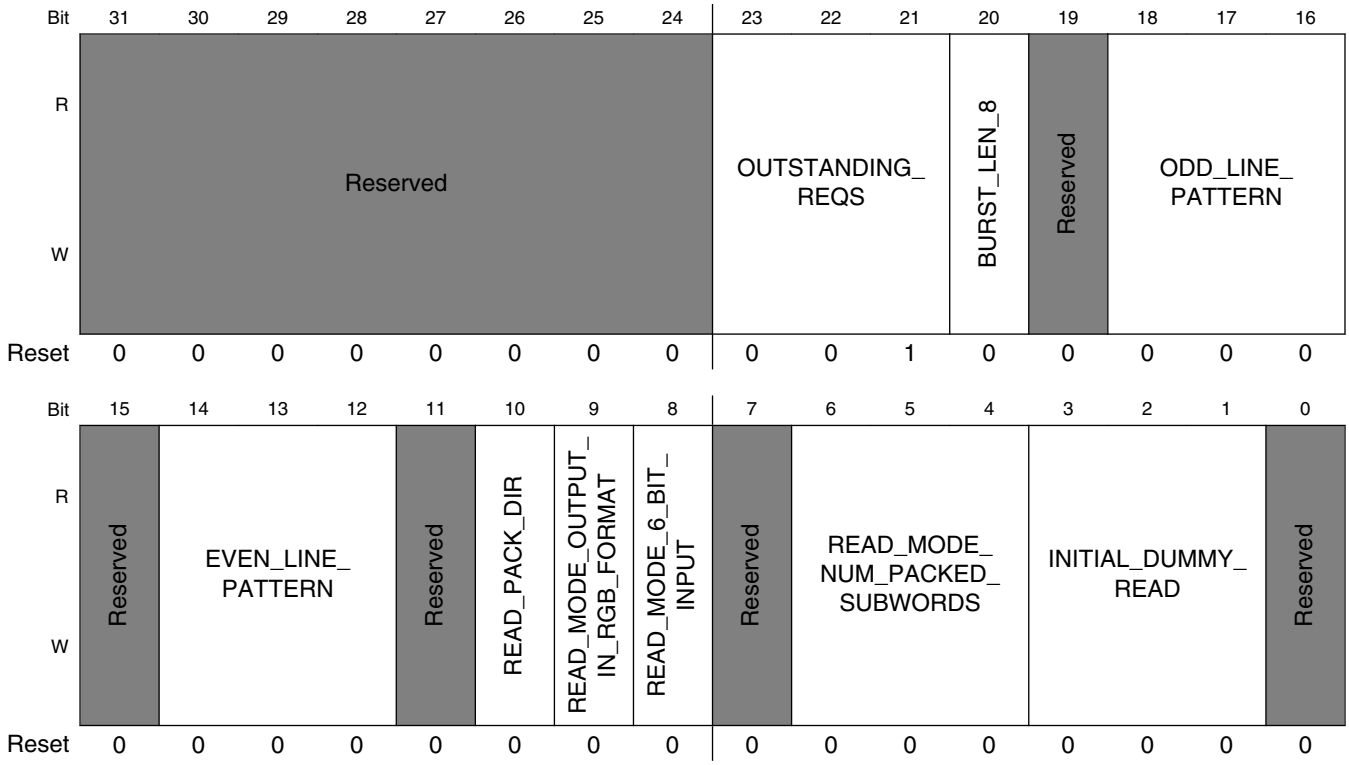
Field	Description
10 UNDERFLOW_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. A TXFIFO underflow in the write mode (MPU/VSYNC/ DOTCLK/DVI mode) was detected. Could produce an error in the DOTCLK / DVI modes. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
9 CUR_FRAME_DONE_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. It indicates that the hardware has completed transmitting the current frame and is in the vertical blanking period in the DOTCLK/DVI modes. In the MPU and VSYNC modes, this IRQ is asserted at the end of the data transfer indicated by LCDIF_TRANSFER_COUNT register. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
8 VSYNC_EDGE_IRQ	This bit is set to indicate that an interrupt is requested by the LCDIF block. This bit is cleared by software by writing a one to its SCT clear address. It is set whenever the leading VSYNC edge is detected in the VSYNC and DOTCLK modes. In the DVI mode, it is asserted every time the block enters a new field. 0x0 **NO_REQUEST** — No Interrupt Request Pending. 0x1 **REQUEST** — Interrupt Request Pending.
7–3 RSRVD0	This field is reserved. Reserved bits. Write as 0.
2 BUSY_ENABLE	This bit enables the use of the interface's busy signal input. This should be enabled for LCD controllers that implement a busy line (to stall the LCDIF from sending more data until ready). Otherwise this bit should be cleared. 0x0 **BUSY_DISABLED** — The busy signal from the LCD controller will be ignored. 0x1 **BUSY_ENABLED** — Enable the use of the busy signal from the LCD controller.
1 MODE86	This bit is used to select between the 8080 and 6800 series of microprocessor modes. This bit should only be changed when RUN is 0. 0x0 **8080_MODE** — Pins LCD_WR_RWn and LCD_RD_E function as active low WR and active low RD signals respectively. 0x1 **6800_MODE** — Pins LCD_WR_RWn and LCD_RD_E function as Read/Write and active high Enable signals respectively.
0 RESET	Reset bit for the external LCD controller. This bit can be changed at any time. It CANNOT be reset by SFTRST. 0x0 **LCDRESET_LOW** — LCD_RESET output signal is low. 0x1 **LCDRESET_HIGH** — LCD_RESET output signal is high.

13.2.5.3 LCDIF General Control2 Register (LCDIF_CTRL2*n*)

The LCDIF Control Register provides overall control of the LCDIF block.

The LCDIF Control2 Register provides additional programming to the LCDIF. It implements some bits which are unlikely to change often in a particular application.

Address: 3032_0000h base + 20h offset + (4d × i), where i=0d to 3d



LCDIF_CTRL2*n* field descriptions

Field	Description
31–24 RSRVD5	This field is reserved. Reserved bits. Write as 0.
23–21 OUTSTANDING_ REQS	This bitfield indicates the maximum number of outstanding transactions that LCDIF should request when it is acting as a bus master. Default is 2 outstanding transactions. 0x0 **REQ_1** — 0x1 **REQ_2** — 0x2 **REQ_4** — 0x3 **REQ_8** — 0x4 **REQ_16** —
20 BURST_LEN_8	By default, when the LCDIF is in the bus master mode, it will issue AXI bursts of length 16 (except when in packed 24 bpp mode, it will issue bursts of length 15). When this bit is set to 1, the block will issue bursts of length 8 (except when in packed 24 bpp mode, it will issue bursts of length 9). Note that this bitfield is only applicable when LCDIF_MASTER is set to 1.
19 RSRVD4	This field is reserved. Reserved bits. Write as 0.
18–16 ODD_LINE_ PATTERN	This field determines the order of the RGB components of each pixel in ODD lines (line numbers 1,3,5,..). This bitfield must be 0 in DVI mode. 0x0 **RGB** — 0x1 **RBG** — 0x2 **GBR** — 0x3 **GRB** —

Table continues on the next page...

LCDIF_CTRL2*n* field descriptions (continued)

Field	Description
	0x4 **BRG** — 0x5 **BGR** —
15 RSRVD3	This field is reserved. Reserved bits. Write as 0.
14–12 EVEN_LINE_ PATTERN	This field determines the order of the RGB components of each pixel in EVEN lines (line numbers 2,4,6,..). This bitfield must be 0 in DVI mode. 0x0 **RGB** — 0x1 **RBG** — 0x2 **GBR** — 0x3 **GRB** — 0x4 **BRG** — 0x5 **BGR** —
11 RSRVD2	This field is reserved. Reserved bits. Write as 0.
10 READ_PACK_DIR	The default value of 0 indicates data is stored in the little endian format. When LCD_DATABUS_WIDTH is 8-bit, this bit provides the option of rearranging the data byte-wise in the big endian format. For example, if READ_MODE_NUM_PACKED_SUBWORDS = 3 and the order of incoming data is 0x11, 0x22 and 0x33, then setting this bit to 1 will cause the data to be stored as 0x00112233 as opposed to the default 0x00332211. This operation occurs after the shifting operation done by SHIFT_NUM_BITS bitfield.
9 READ_MODE_ OUTPUT_IN_ RGB_FORMAT	Setting this bit will enable the LCDIF to convert the incoming data to the RGB format given by WORD_LENGTH bitfield. This feature is not available when WORD_LENGTH is set to 8 bits. LCDIF performs this operation of converting to RGB format after the endianness has been determined by the READ_PACK_DIR bitfield.
8 READ_MODE_6_ BIT_INPUT	Setting this bit to 1 indicates to LCDIF that even though LCD_DATABUS_WIDTH is set to 8 bits, the input data is actually only 6 bits wide and exists on D5-D0.
7 RSRVD1	This field is reserved. Reserved bits. Write as 0.
6–4 READ_MODE_ NUM_PACKED_ SUBWORDS	Indicates the number of valid 8/16/18/24-bit subwords that will be packed into the 32-bit word in read mode. The subword size (8,16, 18 or 24 bits) is determined by the LCD_DATABUS_WIDTH field. The swizzle operation is performed after READ_MODE_NUM_PACKED_SUBWORDS number of subwords has been received and stored in little-endian format. For example, if LCD_DATABUS_WIDTH is set to 8-bit and data to be read back has to be stored in memory in 24-bit unpacked RGB format, set READ_MODE_NUM_PACKED_SUBWORDS to 0x3 so that each 32-bit word will contain only 3 valid bytes (RGB). Maximum value of READ_MODE_NUM_PACKED_SUBWORDS is 4 for 8-bit databus, 2 for 16-bit databus and 1 for 18/24-bit databus.
3–1 INITIAL_DUMMY_ READ	The value in this field determines the number of dummy 8/16/18/24-bit subwords that have to be read back from the LCD panel/controller. They will then not be stored in the read FIFO.
0 RSRVD0	This field is reserved. Reserved bits. Write as 0.

13.2.5.4 LCDIF Horizontal and Vertical Valid Data Count Register (LCDIF_TRANSFER_COUNT)

This register tells the LCDIF how much data will be sent for this frame, or transaction. The total number of words is a product of the V_COUNT and H_COUNT fields. The word size is specified by the WORD_LENGTH field.

This register gives the dimensions of the input frame. For normal operation, but V_COUNT and H_COUNT should be non-zero.

Address: 3032_0000h base + 30h offset = 3032_0030h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V_COUNT																H_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_TRANSFER_COUNT field descriptions

Field	Description
31–16 V_COUNT	Number of horizontal lines per frame which contain valid data. In DOTCLK mode, V_COUNT should be the same as the number of active horizontal lines in a progressive frame. In DVI mode, V_COUNT should be the number of active horizontal lines per frame, and not per field.
H_COUNT	Total valid data (pixels) in each horizontal line. The data size is given by the WORD_LENGTH. When input data is in YCbCr 4:2:2 format (YCBCR422_INPUT is 1), H_COUNT should be the number of 32-bit words that should be fetched by the block and the BYTE_PACKING_FORMAT should be 0xF. In 24-bit packed format (WORD_LENGTH=0x3, BYTE_PACKING_FORMAT=0xF), the H_COUNT must be a multiple of 4 pixels. In 16-bit packed format (WORD_LENGTH=0x0, BYTE_PACKING_FORMAT=0xF), the H_COUNT must be a multiple of 2 pixels.

13.2.5.5 LCD Interface Current Buffer Address Register (LCDIF_CUR_BUF)

This register indicates the address of the current frame being transmitted by LCDIF.

When the LCDIF is behaving as a master, this address points to the address of the current frame of data being sent out via the LCDIF. When the current frame is done, the LCDIF block will assert the cur_frame_done interrupt for software to take action. The block will also copy the LCDIF_NEXT_BUF_ADDR into this bitfield so that the software can program the next frame address into the LCDIF_NEXT_BUF_ADDR bitfield. This address must always be double-word aligned.

Address: 3032_0000h base + 40h offset = 3032_0040h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CUR_BUF field descriptions

Field	Description
ADDR	Address of the current frame being transmitted by LCDIF.

13.2.5.6 LCD Interface Next Buffer Address Register (LCDIF_NEXT_BUF)

This register indicates the address of next frame that will be transmitted by LCDIF.

When the LCDIF is behaving as a master, this address points to the address of the next frame of data that will be sent out via the LCDIF. It is up to the software to make sure that this register is programmed before the end of the current frame, otherwise it might result in old data going out the LCDIF. This address must always be double-word aligned.

Address: 3032_0000h base + 50h offset = 3032_0050h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_NEXT_BUF field descriptions

Field	Description
ADDR	Address of the next frame that will be transmitted by LCDIF.

13.2.5.7 LCD Interface Timing Register (LCDIF_TIMING)

The LCD interface timing register controls the various setup and hold times enforced by the LCD interface in the 6800/8080 MPU and VSYNC modes of operation.

The values used in this register are dependent on the particular LCD controller used, consult the users manual for the particular controller for required timings. Each field of the register must be non-zero, therefore the minimum value is: 0x01010101. NOTE: the timings are not automatically adjusted if the DISPLAY CLOCK (pix_clk) frequency changes—it may be necessary to adjust the timings if DISPLAY CLOCK (pix_clk) changes. NOTE: Each field in this register must be non-zero for the MPU and VSYNC modes to function. The settings in this register do not affect the DOTCLK and DVI modes.

Address: 3032_0000h base + 60h offset = 3032_0060h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CMD_HOLD								CMD_SETUP								DATA_HOLD								DATA_SETUP							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_TIMING field descriptions

Field	Description
31–24 CMD_HOLD	Number of DISPLAY CLOCK (pix_clk) cycles that the LCD_RS signal is active after LCD_CS is deasserted.
23–16 CMD_SETUP	Number of DISPLAY CLOCK (pix_clk) cycles that the LCD_RS signal is active before LCD_CS is asserted.
15–8 DATA_HOLD	Data bus hold time in DISPLAY CLOCK (pix_clk) cycles. Also the time that the data strobe is deasserted in a cycle
DATA_SETUP	Data bus setup time in DISPLAY CLOCK (pix_clk) cycles. Also the time that the data strobe is asserted in a cycle.

13.2.5.8 LCDIF VSYNC Mode and Dotclk Mode Control Register0 (LCDIF_VDCTRL0*n*)

This register is used to control the VSYNC and DOTCLK modes of the LCDIF so as to work with different types of LCDs like moving picture displays and delta pixel displays.

This register gives general programmability to the VSYNC signal including polarity, direction, pulse width, etc.

Address: 3032_0000h base + 70h offset + (4d × i), where i=0d to 3d

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		VSYNC_OEB	ENABLE_PRESENT	VSYNC_POL	HSYNC_POL	DOTCLK_POL	ENABLE_POL	Reserved		VSYNC_PERIOD_UNIT	VSYNC_PULSE_WIDTH_UNIT	HALF_LINE	HALF_LINE_MODE	VSYNC_PULSE_WIDTH	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VSYNC_PULSE_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL0*n* field descriptions

Field	Description
31–30 RSRVD2	This field is reserved. Reserved bits. Write as 0.
29 VSYNC_OEB	0 means the VSYNC signal is an output, 1 means it is an input. Should be set to 0 in the DOTCLK mode. 0x0 **VSYNC_OUTPUT** — The VSYNC pin is in the output mode and the VSYNC signal has to be generated by the LCDIF block. 0x1 **VSYNC_INPUT** — The VSYNC pin is in the input mode and the LCD controller sends the VSYNC signal to the block.
28 ENABLE_ PRESENT	Setting this bit to 1 will make the hardware generate the ENABLE signal in the DOTCLK mode, thereby making it the true RGB interface along with the remaining three signals VSYNC, HSYNC and DOTCLK.
27 VSYNC_POL	Default 0 active low during VSYNC_PULSE_WIDTH time and will be high during the rest of the VSYNC period. Set it to 1 to invert the polarity.
26 HSYNC_POL	Default 0 active low during HSYNC_PULSE_WIDTH time and will be high during the rest of the HSYNC period. Set it to 1 to invert the polarity.
25 DOTCLK_POL	Default is data launched at negative edge of DOTCLK and captured at positive edge. Set it to 1 to invert the polarity. Set it to 0 in DVI mode.
24 ENABLE_POL	Default 0 active low during valid data transfer on each horizontal line.
23–22 RSRVD1	This field is reserved. Reserved bits. Write as 0.
21 VSYNC_ PERIOD_UNIT	Default 0 for counting VSYNC_PERIOD in terms of DISPLAY CLOCK (pix_clk) cycles. Set it to 1 to count in terms of complete horizontal lines. DISPLAY CLOCK (pix_clk) cycles should be used in the VSYNC mode, while horizontal line should be used in the DOTCLK mode.
20 VSYNC_PULSE_ WIDTH_UNIT	Default 0 for counting VSYNC_PULSE_WIDTH in terms of DISPLAY CLOCK (pix_clk) cycles. Set it to 1 to count in terms of complete horizontal lines.
19 HALF_LINE	Setting this bit to 1 will make the total VSYNC period equal to the VSYNC_PERIOD field plus half the HORIZONTAL_PERIOD field (i.e. VSYNC_PERIOD field plus half horizontal line), otherwise it is just VSYNC_PERIOD. Should be only used in the DOTCLK mode, not in the VSYNC interface mode.
18 HALF_LINE_ MODE	When this bit is 0, the first field (VSYNC period) will end in half a horizontal line and the second field will begin with half a horizontal line. When this bit is 1, all fields will end with half a horizontal line, and none will begin with half a horizontal line.
VSYNC_PULSE_ WIDTH	Number of units for which VSYNC signal is active. For the DOTCLK mode, the unit is determined by the VSYNC_PULSE_WIDTH_UNIT. If the VSYNC_PULSE_WIDTH_UNIT is 0 for DOTCLK mode, VSYNC_PULSE_WIDTH must be less than HSYNC_PERIOD. For the VSYNC interface mode, it should be in terms of number of DISPLAY CLOCK (pix_clk) cycles only.

13.2.5.9 LCDIF VSYNC Mode and Dotclk Mode Control Register1 (LCDIF_VDCTRL1)

This register is used to control the VSYNC signal in the VSYNC and DOTCLK modes of the block.

This register determines the period and duty cycle of the VSYNC signal when it is generated in the block.

Address: 3032_0000h base + 80h offset = 3032_0080h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	VSYNC_PERIOD																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL1 field descriptions

Field	Description
VSYNC_PERIOD	Total number of units between two positive or two negative edges of the VSYNC signal. If HALF_LINE is set, it is implicitly calculated to be VSYNC_PERIOD plus half HSYNC_PERIOD.

13.2.5.10 LCDIF VSYNC Mode and Dotclk Mode Control Register2 (LCDIF_VDCTRL2)

This register is used to control the HSYNC signal in the DOTCLK mode of the block.

This register determines the period and duty cycle of the HSYNC signal when it is generated in the block.

Address: 3032_0000h base + 90h offset = 3032_0090h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HSYNC_PULSE_WIDTH														HSYNC_PERIOD																	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_VDCTRL2 field descriptions

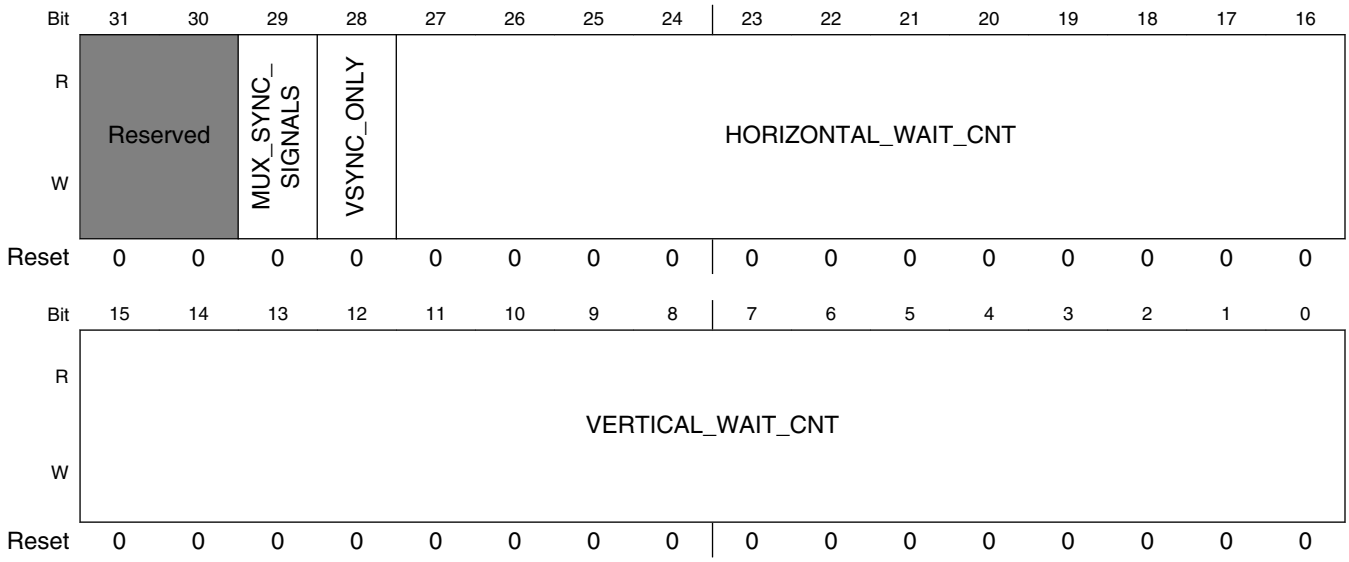
Field	Description
31–18 HSYNC_PULSE_ WIDTH	Number of DISPLAY CLOCK (pix_clk) cycles for which HSYNC signal is active.
HSYNC_PERIOD	Total number of DISPLAY CLOCK (pix_clk) cycles between two positive or two negative edges of the HSYNC signal.

13.2.5.11 LCDIF VSYNC Mode and Dotclk Mode Control Register3 (LCDIF_VDCTRL3)

This register is used to determine the vertical and horizontal wait counts.

This register determines the back porches of HSYNC and VSYNC signals when they are generated by the block.

Address: 3032_0000h base + A0h offset = 3032_00A0h



LCDIF_VDCTRL3 field descriptions

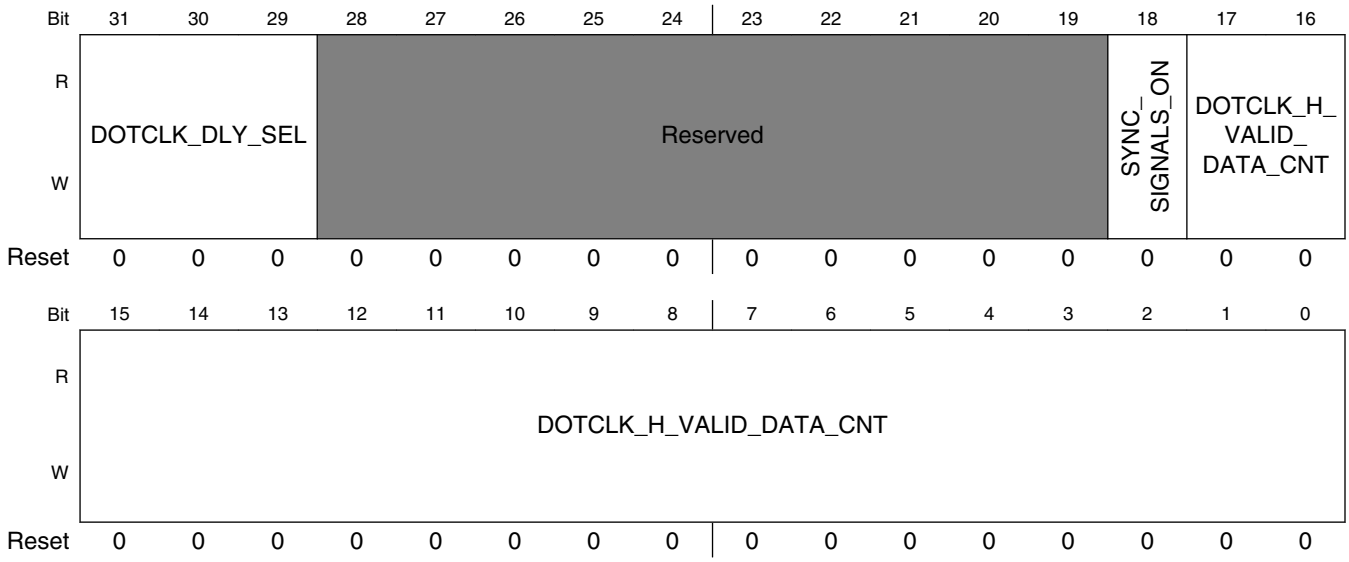
Field	Description
31–30 RSRVD0	This field is reserved. Reserved bits, write as 0.
29 MUX_SYNC_SIGNALS	When this bit is set, the LCDIF block will internally mux HSYNC with LCD_D14, DOTCLK with LCD_D13 and ENABLE with LCD_D12, otherwise these signals will go out on separate pins. This feature can be used to maintain backward compatible with 37xx.
28 VSYNC_ONLY	This bit must be set to 1 in the VSYNC mode of operation, and 0 in the DOTCLK mode of operation.
27–16 HORIZONTAL_WAIT_CNT	In the DOTCLK mode, wait for this number of clocks from falling edge (or rising if HSYNC_POL is 1) of HSYNC signal to account for horizontal back porch plus the number of DOTCLKs before the moving picture information begins.
VERTICAL_WAIT_CNT	In the VSYNC interface mode, wait for this number of DISPLAY CLOCK (pix_clk) cycles from the falling VSYNC edge (or rising if VSYNC_POL is 1) before starting LCD transactions and is applicable only if WAIT_FOR_VSYNC_EDGE is set. Minimum is CMD_SETUP+5. In the DOTCLK mode, it accounts for the vertical back porch lines plus the number of horizontal lines before the moving picture begins. The unit for this parameter is inherently the same as the VSYNC_PERIOD_UNIT.

13.2.5.12 LCDIF VSYNC Mode and Dotclk Mode Control Register4 (LCDIF_VDCTRL4)

This register is used to control the DOTCLK mode of the block.

This register determines the active data in each horizontal line in the DOTCLK mode. Note that the total number of active horizontal lines in the DOTCLK mode is the same as the V_COUNT bitfield in the LCDIF_TRANSFER_COUNT register.

Address: 3032_0000h base + B0h offset = 3032_00B0h



LCDIF_VDCTRL4 field descriptions

Field	Description
31–29 DOTCLK_DLY_ SEL	This bitfield selects the amount of time by which the DOTCLK signal should be delayed before coming out of the LCD_DOTCK pin. 0 = 2ns; 1=4ns;2=6ns;3=8ns. Remaining values are reserved.
28–19 RSRVD0	This field is reserved. Reserved bits, write as 0.
18 SYNC_ SIGNALS_ON	Set this field to 1 if the LCD controller requires that the VSYNC or VSYNC/HSYNC/DOTCLK control signals should be active at least one frame before the data transfers actually start and remain active at least one frame after the data transfers end. The hardware does not count the number of frames automatically. Rather, the VSYNC edge interrupt can be monitored by software to count the number of frames that have occurred after this bit is set and then the RUN bit can be set to start the data transactions. This bit must always be set in the DOTCLK mode of operation, and it must be set in the VSYNC mode of operation when VSYNC signal is an output.
DOTCLK_H_ VALID_DATA_ CNT	Total number of DISPLAY CLOCK (pix_clk) cycles on each horizontal line that carry valid data in DOTCLK mode.

13.2.5.13 Digital Video Interface Control0 Register (LCDIF_DVICTRL0)

The Digital Video interface Control0 register provides the overall control of the Digital Video interface.

This register gives information about the horizontal active, horizontal blanking and total number of lines in the ITU-R BT.656 interface.

EXAMPLE

```
//525/60 video system
```

```
HW_LCDIF_DVICTRL0_H_ACTIVE_CNT_WR(0x5A0);//1440
HW_LCDIF_DVICTRL0_H_BLANKING_CNT_WR(0x106);//262
//625/50 video system
HW_LCDIF_DVICTRL0_H_ACTIVE_CNT_WR(0x5A0);//1440
HW_LCDIF_DVICTRL0_H_BLANKING_CNT_WR(0x112);//274
```

Address: 3032_0000h base + C0h offset = 3032_00C0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				H_ACTIVE_CNT												Reserved				H_BLANKING_CNT											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_DVICTRL0 field descriptions

Field	Description
31–28 RSRVD1	This field is reserved. Reserved bits, write as 0.
27–16 H_ACTIVE_CNT	Number of active video samples to be transmitted. (Mostly will be 1440 for both PAL and NTSC). Must always be a multiple of 4.
15–12 RSRVD0	This field is reserved. Reserved bits, write as 0.
H_BLANKING_ CNT	Number of blanking samples to be inserted between EAV and SAV during horizontal blanking interval.

13.2.5.14 Digital Video Interface Control1 Register (LCDIF_DVICTRL1)

The Digital Video interface Control1 register provides the overall control of the Digital Video interface.

This register contains information about the Field1 start and end, and the Field2 start in the ITU-R BT.656 interface.

EXAMPLE

```
//525/60 video system
HW_LCDIF_DVICTRL1_F1_START_LINE_WR(0x4);//4
HW_LCDIF_DVICTRL1_F1_END_LINE_WR(0x109);//265
HW_LCDIF_DVICTRL1_F2_START_LINE_WR(0x10A);//266
//625/50 video system
HW_LCDIF_DVICTRL1_F1_START_LINE_WR(0x1);//1
HW_LCDIF_DVICTRL1_F1_END_LINE_WR(0x138);//312
HW_LCDIF_DVICTRL1_F2_START_LINE_WR(0x139);//313
```

Address: 3032_0000h base + D0h offset = 3032_00D0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		F1_START_LINE										F1_END_LINE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F1_END_LINE								F2_START_LINE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_DVICTRL1 field descriptions

Field	Description
31–30 RSRVD0	This field is reserved. Reserved bits, write as 0.
29–20 F1_START_LINE	Vertical line number from which Field 1 begins.
19–10 F1_END_LINE	Vertical line number at which Field1 ends.
F2_START_LINE	Vertical line number from which Field 2 begins.

13.2.5.15 Digital Video Interface Control2 Register (LCDIF_DVICTRL2)

The Digital Video interface Control2 register provides the overall control of the Digital Video interface.

This register contains information about the Field2 end, and the Vertical Blanking1 interval in the ITU-R BT.656 interface.

EXAMPLE

```
//525/60 video system
HW_LCDIF_DVICTRL2_F2_END_LINE_WR(0x3);//3
HW_LCDIF_DVICTRL2_V1_BLANK_START_LINE_WR(0x108);//264
HW_LCDIF_DVICTRL2_V1_BLANK_END_LINE_WR(0x11A);//282
//625/50 video system
HW_LCDIF_DVICTRL2_F2_END_LINE_WR(0x271);//625
HW_LCDIF_DVICTRL2_V1_BLANK_START_LINE_WR(0x137);//311
HW_LCDIF_DVICTRL2_V1_BLANK_END_LINE_WR(0x14F);//335
```

Address: 3032_0000h base + E0h offset = 3032_00E0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved		F2_END_LINE										V1_BLANK_START_LINE			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V1_BLANK_START_LINE								V1_BLANK_END_LINE							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_DVICTRL2 field descriptions

Field	Description
31–30 RSRVD0	This field is reserved. Reserved bits, write as 0.
29–20 F2_END_LINE	Vertical line number at which Field 2 ends.
19–10 V1_BLANK_ START_LINE	Vertical line number towards the end of Field1 where first Vertical Blanking interval starts.
V1_BLANK_ END_LINE	Vertical line number in the beginning part of Field2 where first Vertical Blanking interval ends.

13.2.5.16 Digital Video Interface Control3 Register (LCDIF_DVICTRL3)

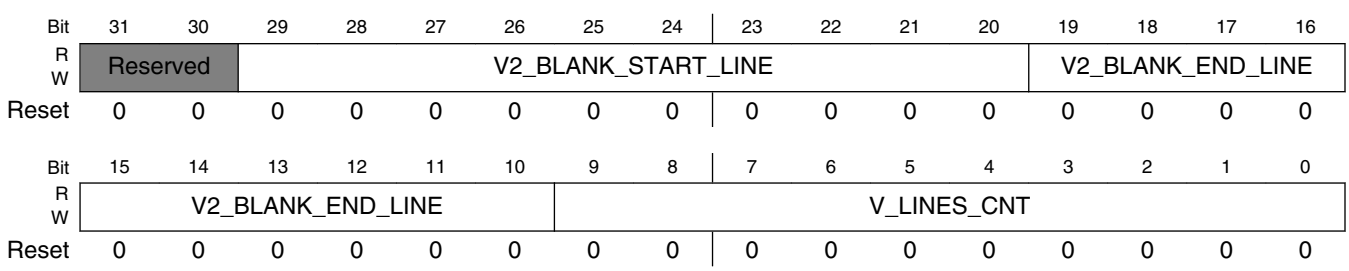
The Digital Video interface Control3 register provides the overall control of the Digital Video interface.

This register contains information about the Vertical Blanking2 interval in the ITU-R BT. 656 interface.

EXAMPLE

```
//525/60 video system
HW_LCDIF_DVICTRL3_V2_BLANK_START_LINE_WR(0x1);//1
HW_LCDIF_DVICTRL3_V2_BLANK_END_LINE_WR(0x13);//19
HW_LCDIF_DVICTRL0_V_LINES_CNT_WR(0x20D);//525
//625/50 video system
HW_LCDIF_DVICTRL3_V2_BLANK_START_LINE_WR(0x270);//624
HW_LCDIF_DVICTRL3_V2_BLANK_END_LINE_WR(0x16);//22
HW_LCDIF_DVICTRL0_V_LINES_CNT_WR(0x271);//625
```

Address: 3032_0000h base + F0h offset = 3032_00F0h



LCDIF_DVICTRL3 field descriptions

Field	Description
31–30 RSRVD0	This field is reserved. Reserved bits, write as 0.

Table continues on the next page...

LCDIF_DVICTRL3 field descriptions (continued)

Field	Description
29–20 V2_BLANK_ START_LINE	Vertical line number towards the end of Field2 where second Vertical Blanking interval starts.
19–10 V2_BLANK_ END_LINE	Vertical line number in the beginning part of Field1 where second Vertical Blanking interval ends.
V_LINES_CNT	Total number of vertical lines per frame (generally 525 or 625)

13.2.5.17 Digital Video Interface Control4 Register (LCDIF_DVICTRL4)

The Digital Video interface Control4 register provides the overall control of the Digital Video interface.

This register is used to add side borders to the output if the input frame width is less than 720 pixels.

EXAMPLE

```
                //If input frame has only 640 pixels per line, but output is supposed to have
720 pixels per line.
                HW_LCDIF_DVICTRL4_H_FILL_CNT_WR(0x50);//80
                HW_LCDIF_DVICTRL4_Y_FILL_VALUE_WR(0x10);//16
                HW_LCDIF_DVICTRL4_CB_FILL_VALUE_WR(0x80);//128
                HW_LCDIF_DVICTRL4_CR_FILL_VALUE_WR(0x80);//128
```

Address: 3032_0000h base + 100h offset = 3032_0100h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Y_FILL_VALUE								CB_FILL_VALUE								CR_FILL_VALUE								H_FILL_CNT							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_DVICTRL4 field descriptions

Field	Description
31–24 Y_FILL_VALUE	Value of Y component of filler data
23–16 CB_FILL_VALUE	Value of CB component of filler data
15–8 CR_FILL_VALUE	Value of CR component of filler data.
H_FILL_CNT	Number of active video samples that have to be filled with the filler data in the front and back portions of the active horizontal interval. Must be a multiple of 4. This field will have to be programmed if the input frame has less than 720 pixels per line.

13.2.5.18 RGB to YCbCr 4:2:2 CSC Coefficient0 Register (LCDIF_CSC_COEFF0)

LCDIF_CSC_COEFF0 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

EXAMPLE

```
HW_LCDIF_CSC_COEFF0_C0_WR(0x41);//0.257x256=65
HW_LCDIF_CSC_COEFF0_CSC_SUBSAMPLE_FILTER_WR(0x3);
```

Address: 3032_0000h base + 110h offset = 3032_0110h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								C0							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														CSC_ SUBSAMPL E_FILTER	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CSC_COEFF0 field descriptions

Field	Description
31–26 RSRVD1	This field is reserved. Reserved bits, write as 0.
25–16 C0	Two's complement red multiplier coefficient for Y
15–2 RSRVD0	This field is reserved. Reserved bits, write as 0.
CSC_ SUBSAMPLE_ FILTER	This register describes the filtering and subsampling scheme to be performed on the chroma components in order to convert from YCbCr 4:4:4 to YCbCr 4:2:2 space. Note that the following descriptions apply individually to Cb and Cr. 0x0 **SAMPLE_AND_HOLD** — No filtering, simply keep every chroma value for samples numbered 2n and discard chroma values associated with all samples numbered 2n+1. 0x1 **RSRVD** — Reserved 0x2 **INTERSTITIAL** — Chroma samples numbered 2n and 2n+1 are averaged (weights 1/2, 1/2) and that chroma value replaces the two chroma values at 2n and 2n+1. This chroma now exists horizontally halfway between the two luma samples. 0x3 **COSITED** — Chroma samples numbered 2n-1, 2n, and 2n+1 are averaged (weights 1/4, 1/2, 1/4) and that chroma value exists at the same site as the luma sample numbered 2n and the chroma samples at 2n+1 are discarded.

13.2.5.19 RGB to YCbCr 4:2:2 CSC Coefficient1 Register (LCDIF_CSC_COEFF1)

LCDIF_CSC_COEFF1 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

EXAMPLE

```
HW_LCDIF_CSC_COEFF1_C1_WR(0x81);//0.504x256=129
HW_LCDIF_CSC_COEFF1_C2_WR(0x19);//0.098x256=25
```

Address: 3032_0000h base + 120h offset = 3032_0120h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						C2										Reserved						C1									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CSC_COEFF1 field descriptions

Field	Description
31–26 RSRVD1	This field is reserved. Reserved bits, write as 0.
25–16 C2	Two's complement blue multiplier coefficient for Y
15–10 RSRVD0	This field is reserved. Reserved bits, write as 0.
C1	Two's complement green multiplier coefficient for Y

13.2.5.20 RGB to YCbCr 4:2:2 CSC Coefficent2 Register (LCDIF_CSC_COEFF2)

LCDIF_CSC_COEFF2 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

EXAMPLE

```
HW_LCDIF_CSC_COEFF2_C3_WR(0x3DB);//-0.148x256=-37
HW_LCDIF_CSC_COEFF2_C4_WR(0x3B6);//-0.291x256=-74
```

Address: 3032_0000h base + 130h offset = 3032_0130h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						C4										Reserved						C3									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CSC_COEFF2 field descriptions

Field	Description
31–26 RSRVD1	This field is reserved. Reserved bits, write as 0.
25–16 C4	Two's complement green multiplier coefficient for Cb
15–10 RSRVD0	This field is reserved. Reserved bits, write as 0.
C3	Two's complement red multiplier coefficient for Cb

13.2.5.21 RGB to YCbCr 4:2:2 CSC Coefficient3 Register (LCDIF_CSC_COEFF3)

LCDIF_CSC_COEFF3 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

EXAMPLE

```
HW_LCDIF_CSC_COEFF3_C5_WR(0x70);//0.439x256=112
HW_LCDIF_CSC_COEFF3_C6_WR(0x70);//0.439x256=112
```

Address: 3032_0000h base + 140h offset = 3032_0140h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						C6										Reserved						C5									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CSC_COEFF3 field descriptions

Field	Description
31–26 RSRVD1	This field is reserved. Reserved bits, write as 0.

Table continues on the next page...

LCDIF_CSC_COEFF3 field descriptions (continued)

Field	Description
25–16 C6	Two's complement red multiplier coefficient for Cr
15–10 RSRVD0	This field is reserved. Reserved bits, write as 0.
C5	Two's complement blue multiplier coefficient for Cb

13.2.5.22 RGB to YCbCr 4:2:2 CSC Coefficient4 Register (LCDIF_CSC_COEFF4)

LCDIF_CSC_COEFF4 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

EXAMPLE

```
HW_LCDIF_CSC_COEFF4_C7_WR(0x3A2);//-0.368x256=-94
HW_LCDIF_CSC_COEFF4_C8_WR(0x3EE);//-0.071x256=-18
```

Address: 3032_0000h base + 150h offset = 3032_0150h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						C8										Reserved						C7									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CSC_COEFF4 field descriptions

Field	Description
31–26 RSRVD1	This field is reserved. Reserved bits, write as 0.
25–16 C8	Two's complement blue multiplier coefficient for Cr
15–10 RSRVD0	This field is reserved. Reserved bits, write as 0.
C7	Two's complement green multiplier coefficient for Cr

13.2.5.23 RGB to YCbCr 4:2:2 CSC Offset Register (LCDIF_CSC_OFFSET)

LCDIF_CSC_ register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC.

Address: 3032_0000h base + 160h offset = 3032_0160h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							CBCR_OFFSET									Reserved							Y_OFFSET								
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

LCDIF_CSC_OFFSET field descriptions

Field	Description
31–25 RSRVD1	This field is reserved. Reserved bits, write as 0.
24–16 CBCR_OFFSET	Two's complement offset for the Cb and Cr components
15–9 RSRVD0	This field is reserved. Reserved bits, write as 0.
Y_OFFSET	Two's complement offset for the Y component

13.2.5.24 RGB to YCbCr 4:2:2 CSC Limit Register (LCDIF_CSC_LIMIT)

LCDIF_CSC_CTRL0 register provides overall control over color space conversion from RGB to 4:2:2 YCbCr. The equations for the conversion are given by: Y = C0*R + C1*G + C2*B + Y_offset Cb= C3*R + C4*G + C5*B + CbCr_offset Cr= C6*R + C7*G + C8*B + CbCr_offset

This register carries programming information about RGB to YCbCr 4:2:2 CSC. Note that the values in this register are unsigned.

EXAMPLE

```
HW_LCDIF_CSC_LIMIT_CBCR_MIN_WR(0x10);//16
HW_LCDIF_CSC_LIMIT_CBCR_MAX_WR(0xF0);//240
HW_LCDIF_CSC_LIMIT_Y_MIN_WR(0x10);//16
HW_LCDIF_CSC_LIMIT_Y_MAX_WR(0xEB);//235
```

Address: 3032_0000h base + 170h offset = 3032_0170h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CBCR_MIN								CBCR_MAX								Y_MIN								Y_MAX							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

LCDIF_CSC_LIMIT field descriptions

Field	Description
31–24 CBCR_MIN	Lower limit of Cb and Cr after RGB to 4:2:2 YCbCr conversion
23–16 CBCR_MAX	Upper limit of Cb and Cr after RGB to 4:2:2 YCbCr conversion
15–8 Y_MIN	Lower limit of Y after RGB to 4:2:2 YCbCr conversion
Y_MAX	Upper limit of Y after RGB to 4:2:2 YCbCr conversion

13.2.5.25 LCD Interface Data Register (LCDIF_DATA)

This register is used to transfer data using the PIO interface mode of operation. In MPU mode, data written to this register will be transferred out to the display device. When receiving data from the display, data is read from this register using PIO operations. During write operations, data can be written to this register (from the processor's perspective) as bytes, half-words (16 bits), or words (32 bits) as desired.

This register holds the 32-bit word written by the Arm platform into LCDIF. This data then gets sent out by the block across the interface.

Address: 3032_0000h base + 180h offset = 3032_0180h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DATA_THREE								DATA_TWO								DATA_ONE								DATA_ZERO							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_DATA field descriptions

Field	Description
31–24 DATA_THREE	Byte 3 (most significant byte) of data written to LCDIF.
23–16 DATA_TWO	Byte 2 of data written to LCDIF.
15–8 DATA_ONE	Byte 1 of data written to LCDIF.
DATA_ZERO	Byte 0 (least significant byte) of data written to LCDIF.

13.2.5.26 Bus Master Error Status Register (LCDIF_BM_ERROR_STAT)

This register reflects the virtual address at which the AXI master received an error response from the slave.

When the BM_ERROR_IRQ is asserted, the address of the bus error is updated in the register.

Address: 3032_0000h base + 190h offset = 3032_0190h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_BM_ERROR_STAT field descriptions

Field	Description
ADDR	Virtual address at which bus master error occurred.

13.2.5.27 CRC Status Register (LCDIF_CRC_STAT)

This register reflects the CRC value of each frame sent out by LCDIF. The CRC is done on the final output bus, so the value will be dependent on the LCD_DATABUS_WIDTH bitfield even if the input data is the same.

This register will be updated when the CUR_FRAME_DONE_IRQ is asserted. In the case of DVI mode, the CRC is calculated for the entire frame, not separately for each field in the frame.

Address: 3032_0000h base + 1A0h offset = 3032_01A0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CRC_VALUE																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_CRC_STAT field descriptions

Field	Description
CRC_VALUE	Calculated CRC value.

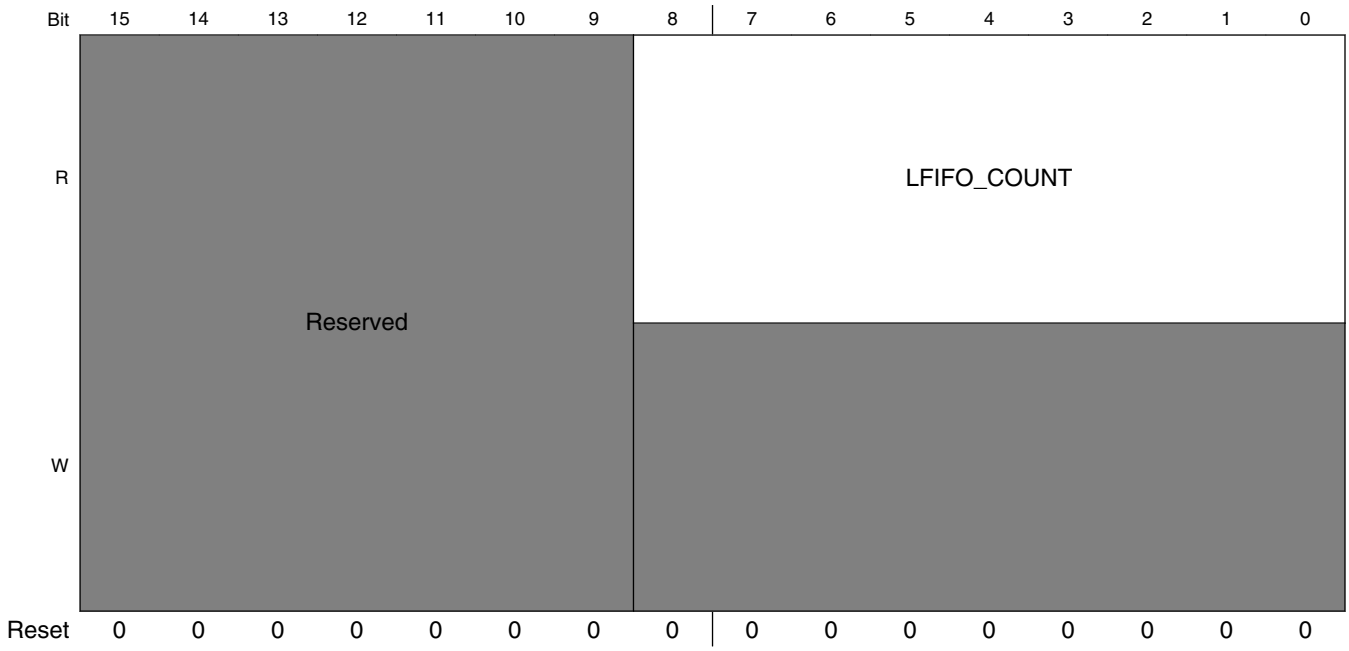
13.2.5.28 LCD Interface Status Register (LCDIF_STAT)

The LCD interface status register can be used to check the current status of the LCDIF block.

The LCD interface status register that contains read only views of some parameters or current state of the block.

Address: 3032_0000h base + 1B0h offset = 3032_01B0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	PRESENT	Reserved	LFIFO_FULL	LFIFO_EMPTY	TXFIFO_FULL	TXFIFO_EMPTY	BUSY	DVI_CURRENT_FIELD	Reserved							
W																
Reset	1	0	0	1	0	1	0	1	0	0	0	0	0	0	0	0



LCDIF_STAT field descriptions

Field	Description
31 PRESENT	0: LCDIF not present on this product 1: LCDIF is present.
30 -	This field is reserved. Reserved.
29 LFIFO_FULL	Read only view of the signals that indicates LCD LFIFO is full.
28 LFIFO_EMPTY	Read only view of the signals that indicates LCD LFIFO is empty.
27 TXFIFO_FULL	Read only view of the signals that indicates LCD TXFIFO is full.
26 TXFIFO_EMPTY	Read only view of the signals that indicates LCD TXFIFO is empty.
25 BUSY	Read only view of the input busy signal from the external LCD controller.
24 DVI_CURRENT_FIELD	Read only view of the current field being transmitted. DVI_CURRENT_FIELD = 0 means field 1. DVI_CURRENT_FIELD = 1 means field 2.
23–9 RSRVD0	This field is reserved. Reserved bits. Write as 0.
LFIFO_COUNT	Read only view of the current count in Latency buffer (LFIFO).

13.2.5.29 LCDIF Threshold Register (LCDIF_THRES)

This register is used to activate control signals when the number of pixels reaches the programmed threshold. These control signals, in turn, can be used to manipulate access priority or dynamically change the input clock frequency to meet the required pixel throughput.

Memory request priority threshold register.

Address: 3032_0000h base + 200h offset = 3032_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved							FASTCLOCK									Reserved							PANIC								
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1

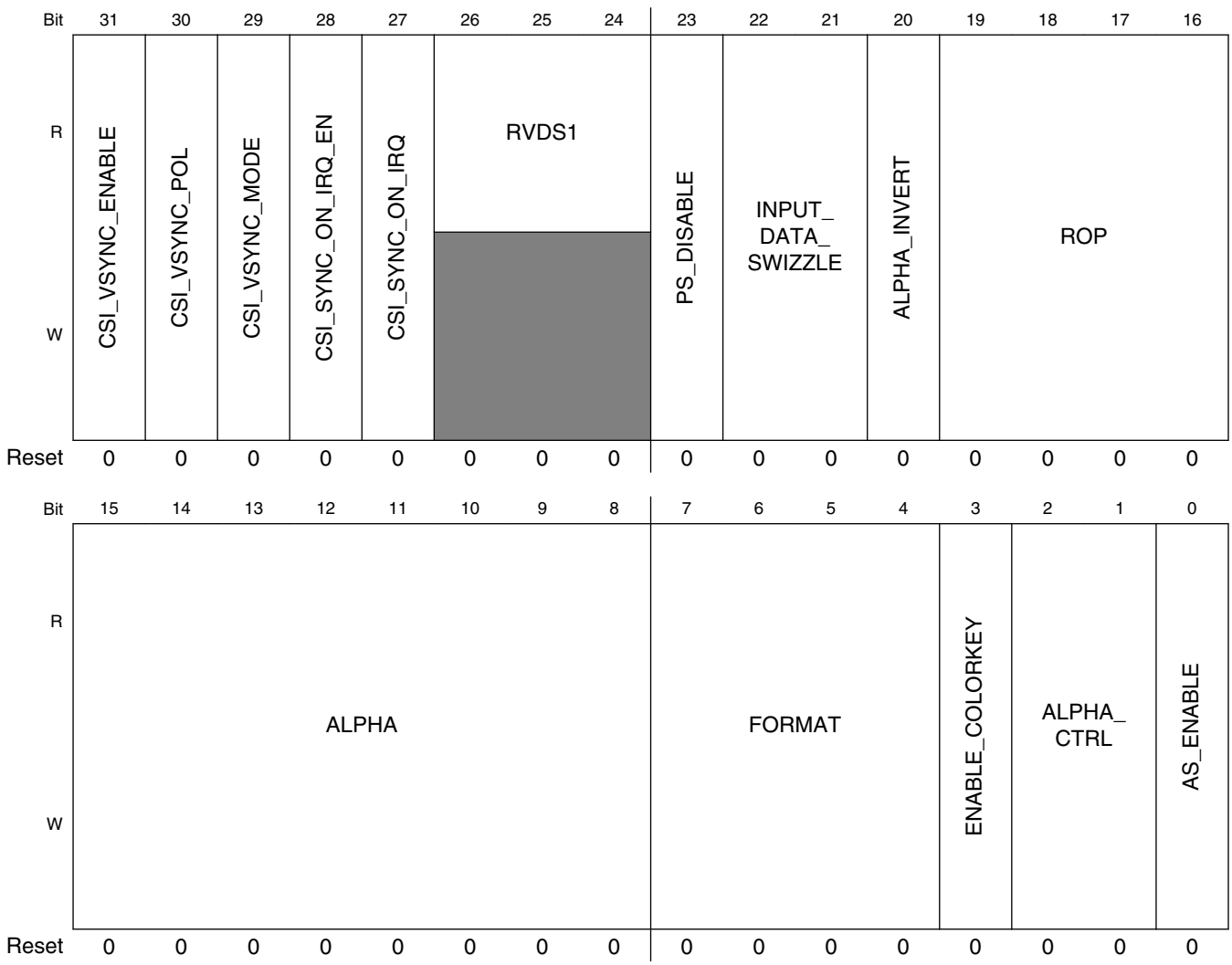
LCDIF_THRES field descriptions

Field	Description
31–25 RSRVD2	This field is reserved. Reserved bits. Write as 0.
24–16 FASTCLOCK	This value should be set to a value of pixels, from 0 to 511. When the number of pixels in the input pixel FIFO is LESS than this value, the fast clock control output will be raised. This signal can be used to reduce the system bus clock frequency to save power during horizontal or vertical blanking intervals. This value should also be programmed to a value that is greater than the "PANIC" threshold value. This will allow a faster clock to recover the number of pixels in the FIFO before a "panic" level is encountered.
15–9 RSRVD1	This field is reserved. Reserved bits. Write as 0.
PANIC	This value should be set to a value of pixels from 0 to 511. When the number of pixels in the input pixel FIFO is less than this value, the internal panic control output will be raised. This signal can be used to raise the access LCDIF's access priority.

13.2.5.30 LCDIF AS Buffer Control Register (LCDIF_AS_CTRL)

The Alpha Surface Parameter register provides additional controls for AS.

Address: 3032_0000h base + 210h offset = 3032_0210h



LCDIF_AS_CTRL field descriptions

Field	Description
31 CSI_VSYNC_ ENABLE	When this bit is set by software, the LCDIF work as sync mode with CSI input.
30 CSI_VSYNC_ POL	Default 0 active low during VSYNC_PULSE_WIDTH time and will be high during the rest of the VSYNC period. Set it to 1 to invert the polarity.

Table continues on the next page...

LCDIF_AS_CTRL field descriptions (continued)

Field	Description
29 CSI_VSYNC_MODE	this bit is set by software to decide which vsync generate mode. LCDIF vsync generate by internal counter when set to 0, LCDIF vsync delayed by each csi_vsync_in when set to 1; INT_SYNC_MODE = 0x0 LCDIF vsync generate by internal counter. EXT_SYNC_MODE = 0x1 LCDIF vsync delayed by each csi_vsync_in.
28 CSI_SYNC_ON_IRQ_EN	This bit is set to enable an interrupt when LCDIF lock with CSI vsync input.
27 CSI_SYNC_ON_IRQ	this bit is set by software to decide which vsync generate mode. LCDIF vsync generate by internal counter when set to 0, LCDIF vsync delayed by each csi_vsync_in when set to 1; INT_SYNC_MODE = 0x0 LCDIF vsync generate by internal counter. EXT_SYNC_MODE = 0x1 LCDIF vsync delayed by each csi_vsync_in.
26–24 RVDS1	Reserved, always set to zero.
23 PS_DISABLE	When this bit is set by software, the LCDIF will disable PS buffer data.
22–21 INPUT_DATA_SWIZZLE	This field specifies how to swap the bytes either in the HW_LCDIF_DATA register or those fetched by the AXI master part of LCDIF. The swizzle function is independent of the WORD_LENGTH bit. See the explanation of the HW_LCDIF_DATA below for names and definitions of data register fields. The supported swizzle configurations are: NO_SWAP = 0x0 No byte swapping.(Little endian) LITTLE_ENDIAN = 0x0 Little Endian byte ordering (same as NO_SWAP). BIG_ENDIAN_SWAP = 0x1 Big Endian swap (swap bytes 0, 3 and 1, 2). SWAP_ALL_BYTES = 0x1 Swizzle all bytes, swap bytes 0, 3 and 1, 2 (aka Big Endian). HWD_SWAP = 0x2 Swap half-words. HWD_BYTE_SWAP = 0x3 Swap bytes within each half-word.
20 ALPHA_INVERT	Setting this bit to logic 0 will not alter the alpha value. A logic 1 will invert the alpha value and apply (1-alpha) for image composition.
19–16 ROP	Indicates a raster operation to perform when enabled. Raster operations are enabled through the ALPHA_CTRL field. MASKAS = 0x0 AS AND PS MASKNOTAS = 0x1 nAS AND PS MASKASNOT = 0x2 AS AND nPS MERGEAS = 0x3 AS OR PS MERGENOTAS = 0x4 nAS OR PS MERGEASNOT = 0x5 AS OR nPS NOTCOPYAS = 0x6 nAS NOT = 0x7 nPS NOTMASKAS = 0x8 AS NAND PS NOTMERGEAS = 0x9 AS NOR PS XORAS = 0xA AS XOR PS NOTXORAS = 0xB AS XNOR PS

Table continues on the next page...

LCDIF_AS_CTRL field descriptions (continued)

Field	Description
15–8 ALPHA	Alpha modifier used when the ALPHA_MULTIPLY or ALPHA_OVERRIDE values are programmed in REG_AS_CTRL[ALPHA_CTRL]. The output alpha value will either be replaced (ALPHA_OVERRIDE) or scaled (ALPHA_MULTIPLY) when selected.
7–4 FORMAT	Indicates the input buffer format for AS. ARGB8888 = 0x0 32-bit pixels with alpha RGB888 = 0x4 32-bit pixels without alpha (unpacked 24-bit format) ARGB1555 = 0x8 16-bit pixels with alpha ARGB4444 = 0x9 16-bit pixels with alpha RGB555 = 0xC 16-bit pixels without alpha RGB444 = 0xD 16-bit pixels without alpha RGB565 = 0xE 16-bit pixels without alpha
3 ENABLE_ COLORKEY	Indicates that colorkey functionality is enabled for this alpha surface. Pixels found in the alpha surface colorkey range will be displayed as transparent (the PS pixel will be used).
2–1 ALPHA_CTRL	Determines how the alpha value is constructed for this alpha surface. Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Embedded = 0x0 Indicates that the AS pixel alpha value will be used to blend the AS with PS. The ALPHA field is ignored. Override = 0x1 Indicates that the value in the ALPHA field should be used instead of the alpha values present in the input pixels. Multiply = 0x2 Indicates that the value in the ALPHA field should be used to scale all pixel alpha values. Each pixel alpha is multiplied by the value in the ALPHA field. ROPs = 0x3 Enable ROPs. The ROP field indicates an operation to be performed on the alpha surface and PS pixels.
0 AS_ENABLE	When this bit is set by software, the LCDIF will start fetching AS buffer data in bus master mode and combine it with another buffer.

13.2.5.31 Alpha Surface Buffer Pointer (LCDIF_AS_BUF)

This register is used to indicate the base address of the AS buffer.

Address: 3032_0000h base + 220h offset = 3032_0220h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_AS_BUF field descriptions

Field	Description
ADDR	Address pointer for the alpha surface 0 buffer.

13.2.5.32 LCDIF_AS_NEXT_BUF

When the LCDIF is behaving as a master, this address points to the address of the next frame of data that will be sent out via the LCDIF. It is upto the software to make sure that this register is programmed before the end of the current frame, otherwise it might result in old data going out the LCDIF. This address must always be double-word aligned.

Address: 3032_0000h base + 230h offset = 3032_0230h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ADDR																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_AS_NEXT_BUF field descriptions

Field	Description
ADDR	Address of the next frame that will be transmitted by LCDIF.

13.2.5.33 LCDIF Overlay Color Key Low (LCDIF_AS_CLRKEYLOW)

If a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. Colorkey operations are higher priority than alpha or ROP operations.

Address: 3032_0000h base + 240h offset = 3032_0240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RSVD1								PIXEL																							
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

LCDIF_AS_CLRKEYLOW field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	Low range of RGB color key applied to AS buffer

13.2.5.34 LCDIF Overlay Color Key High (LCDIF_AS_CLRKEYHIGH)

If a pixel in the current overlay image with a color that falls in the range from the ASCOLORKEYLOW to ASCOLORKEYHIGH range, it will use the PS pixel value for that location. Colorkey operations are higher priority than alpha or ROP operations.

Address: 3032_0000h base + 250h offset = 3032_0250h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RSVD1								PIXEL																							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_AS_CLRKEYHIGH field descriptions

Field	Description
31–24 RSVD1	Reserved, always set to zero.
PIXEL	High range of RGB color key applied to AS buffer

13.2.5.35 LCD working insync mode with CSI for VSYNC delay (LCDIF_SYNC_DELAY)

The LCDIF DOTCLK mode VSYNC will delay from CSI_VSYNC as
(V_COUNT_DELAY * HSYNC_PERIOD + H_COUNT_DELAY) PIXCLK cycles

Address: 3032_0000h base + 260h offset = 3032_0260h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V_COUNT_DELAY																H_COUNT_DELAY															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

LCDIF_SYNC_DELAY field descriptions

Field	Description
31–16 V_COUNT_ DELAY	LCDIF VSYNC delayed counter for CSI_VSYNC.
H_COUNT_ DELAY	LCDIF VSYNC delayed counter for CSI_VSYNC.

13.3 Graphics Processing Unit (GPU)

13.3.1 Overview

The graphics processing unit (GPU) provides high performance, high quality graphics, low power consumption, and the smallest silicon footprint for its class. The dynamic power consumption is minimized by extensive use of multi-level hierarchical clock gating.

An optimized software stack, complete software development tools, and a growing application ecosystem are supported by a robust graphics pipeline designed for industry-standard APIs, and with full support for Android, Linux, and Windows embedded development platforms. The GPU supports the following graphics APIs:

- OpenGL ES 3.1/3.0
- OpenGL ES 2.0/1.1
- EGL 1.4
- DirectX 11 (9_3)
- OpenGL 3.1/3.0
- OpenGL 2.1/2.0
- OpenVG 1.1
- OpenCL 1.2 FP / 1.1 FP w/ Global memory

13.3.1.1 Block Diagram

A block diagram and a description of the functional blocks of the complete graphics pipeline is given below.

- Host Interface - Allows GCCORE to communicate with external memory and the CPU through AXI or AHB bus. In this block data crosses clock domain boundaries.
- Memory Controller - Internal memory management unit that controls the block-to-host memory request interface.
- Fetch Engine - Inserts high level primitives and commands into the graphics pipeline.
- Input Assembly - Assembles vertices and primitives for further processing.
- Compute Modules:
  - Fixed Function 3D Rendering Engine Units:
    - Setup/Raster - Converts triangles and lines into pixels. Computes slopes of color attributes and texture coordinates. Performs clipping.

        - Texture Engine - Retrieves texture information from memory upon request by the vertex shader or fragment shader. Performs interpolation and filtering, and transfers the computed value to the fragment shader or the vertex shader. The texture unit can process two pixels or two vertices/cycle.
        - Pixel Engine/Blit - The Pixel Engine does alpha blending and visible surface determination. The Blit Engine does resolve, clear, tiling and de-tiling as well as FSAA filtering. The Pixel Engine can process two pixels/cycle.
    - Programmable Engine Units:
        - Shader Execution Units - SIMD processor that performs as vertex shader and fragment shader. When used as a vertex shader, it performs geometry transformations and lighting computations. When used as a fragment shader it applies texture data and computes color values for each pixel. The shaders also act as Compute Units for OpenCL. The GPU has 4 such vec4 shaders.
- Universal Storage Cache - Cache shared between the Fetch Engine and the Compute Modules. A portion of this cache can be locked to stay on-chip.

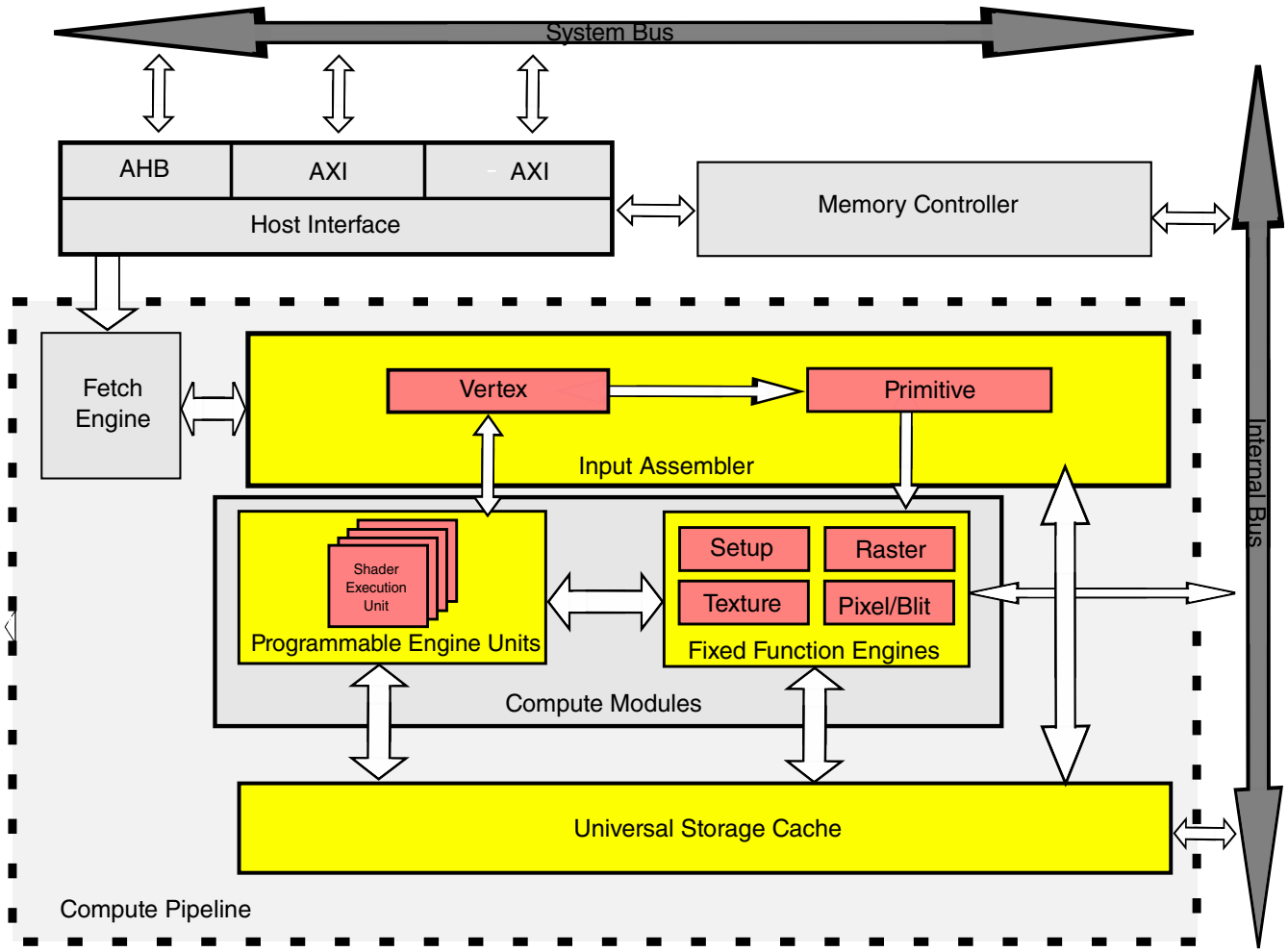


Figure 13-18. GPU Block Diagram

13.3.2 GPU Features

The features of the GPU 3D unit include:

- OpenGL ES 3.1 / 3.0 / 2.0 / 1.1 compliance, including extensions; OpenVG 1.1
- IEEE 32-bit floating-point pipeline
- Ultra-threaded, unified vertex and fragment (pixel) shaders
- Low bandwidth at both high and low data rates
- Low CPU loading
- Up to 16 programmable elements per vertex
- Dependent texture operation with high-performance
- Alpha blending
- Depth and stencil compare

- Point sampling, bi-linear sampling, tri-linear filtering, and cubic textures
- Resolve and fast clear
- 8k x 8k texture size and 8k x 8k rendering target
- 16 Vertex DMA streams
- Generates 2 texels and 2 pixels/cycle.
- Supports YUV formats for display output (YUV422, YUYV 4:2:2)
- 3D Composition engine
- MMU functionality supported
- Wide AA line support
- Primitive restart support
- Software programmable Big Endian support
- Performance Counters for DMA Profiling

The following table describes API support and select architectural features of the GPU:

Table 13-9. GPU Architecture Features

Feature	GPU Support
Primary API	OpenGL ES 3.1 / 3.0 / 2.0 / 1.1 OpenCL 1.1 / 1.2
Additional APIs	OpenVG 1.1 DirectX 11 (9_3) OpenGL 3.1 / 3.0 / 2.0 / 2.1
Other graphics support	EGL 1.4
Drivers	OpenGL ES 3.1 / 3.0 / 2.0 / 1.1 OpenVG 1.1 EGL 1.4 DirectX 11 (9_3) OpenGL 3.1 / 3.0 / 2.0 / 2.1 OpenCL 1.1 / 1.2
Operating systems	Windows Embedded Compact / Embedded CE 6 Embedded Linux Android 8.0/7.x / 6.0 /5.1 / 4.x
Z (depth)	Early Z support included
Stencil	Early stencil support included
GLSL ES Shader languages	3.1 / 3.0 / 1.0
DirectX Shader model compatibility	SM3.0
Shader types and execution units	Four programmable Scalable Ultra-threaded Unified Shaders (SIMD4:transcendental,ctl-flow,tx-load) one instruction issue per shader per clock; IEEE 32-bit floating-point pipeline supports long shader instructions
FSAA anti-aliasing mechanisms	High quality MSAA 4x

13.3.2.1 GPU Host and Memory Interface Features

Table 13-10. GPU host and memory interface features

Feature	GPU Support
AHB interface	32-bit
AXI interface	2 128-bit AXI / ACE-Lite interfaces for external memory access
Virtual memory support	Yes
Code and data memory location restrictions	Unrestricted; arbitrary memory reads and writes
Physical address	32 bits
Resource locks with CPU	Semaphore lock

13.3.2.2 GPU Power Management Features

Table 13-11. Power management features

Feature	GPU Support
Low power CMOS technology compatible	Yes
Dynamic frequency scaling based on current load	Yes
Automatic clock gating of flip flops and rams	Yes
Global clock gating of unused macro blocks	Yes
Independent clock gating for the shader core	Yes
Software controlled effective clock frequency without changing the PLL	Yes

13.3.2.3 GPU Command Processor Features

Table 13-12. GPU command processor features

Feature	GPU Support
Counters	Variety of hardware counters for performance profiling

13.3.2.4 OpenCL Support

Support for OpenCL includes the following capabilities in a full memory configuration:

Table 13-13. OpenCL Support

OpenCL Support	GPU Support
Profile	Full Profile with Global memory
Compute Devices (GPGPU cores)	1
Compute Units per device	1
Processing Elements per compute unit (vec4 sh cores)	4
Preferred work-group/ thread group size	8
Max count global work-items each dim	64KB
Max count work-items each dim per work-group	1 KB
Instruction Cache Size	512

13.3.2.5 Texture processing

Table 13-14. Texture Processing Features

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td>Fixed-point input texture formats</td><td>A8, L8, I8, A8L8, ARGB4, XRGB4, ARGB8, XRGB8, ABGR8, XBGR8, R5G6B5, A1RGB5, X1RGB5, YUY2, UYVY, D16, D24X8, A8_OES, DXT1, DXT2, DXT3, DXT4, DXT5, HDR7E3, HDR6E4, HDR5E5, HDR6E5, RGBE8, RGBE8F, RGB9E5, RGB9E5F, ETC1, ETC2_RGB8, ETC2_RGB8A1, ETC2_RGB8A8, EAC_R11_UNSIGNED, EAC__RG11_UNSIGNED, EAC__RG11_SIGNED, R8G8, RF16, RF16GF16, RF16GF16BGF16AF16, RF32, RF32GF32, R10G10B10A2, EAC__R11_SIGNED, R8__SNORM, RG8__SNORM, RGBX8__SNORM, RGBA8__SNORM, YUV_ASSEMBLY, RI8, RI8GI8, RI8GI8BI8AI8, RI16, RI16GI16, RI16GI16BI16AI16, RI32, RI32GI32, RF11GF11BF10, RF16GF16BF16AF16, RI10GI10BI10AI2, R8G8B8G8, G8R8G8B8, RI8GI8BI8GI8, GI8RI8GI8BI8, RF11GF11BF10. All fixed-point formats are filtered. sRGB conversion and Swizzling supported for color channels.
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha/X/s</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4 don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1 don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>sARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
</table>
</td></tr>
</table>

Table continues on the next page...

Table 13-14. Texture Processing Features (continued)

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td></td><td>
<table>
<tr><th>Planes</th><th>Format</th><th>Mode</th><th>Y</th><th>U</th><th>V</th><th>UV</th><th>YUYV</th><th>UYVY</th></tr>
<tr><td>1</td><td>YUY2(YUYV)</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td>1</td><td></td></tr>
<tr><td>1</td><td>UYVY</td><td>4:2:2</td><td></td><td></td><td></td><td></td><td></td><td>1</td></tr>
</table>
</td></tr>
<tr><td>Additional texture formats supported through Resolve conversion</td><td>Resolve converts planar to YUV 4:2:2 packed:
<table>
<tr><th>Planes</th><th>Format</th><th>Mode</th><th>Y</th><th>U</th><th>V</th><th>UV</th><th>YUYV</th><th>UYVY</th></tr>
<tr><td>3</td><td>YV12</td><td>4:2:0</td><td>1</td><td>1</td><td>1</td><td></td><td></td><td></td></tr>
<tr><td>2</td><td>NV12</td><td>4:2:0</td><td>1</td><td></td><td></td><td>1</td><td></td><td></td></tr>
</table>
</td></tr>
<tr><td>Texture compression</td><td>4 bits and 8 bits per texel</td></tr>
<tr><td>Compressed texture formats</td><td>DXT1, DXT2, DXT3, DXT4, DXT5, ETC1, ETC2, EAC11. All compressed formats are filtered.</td></tr>
<tr><td>Texture Formats for Optimal Performance</td><td>Formats with 4 bits per texel: DXT1, ETC1, EXT2_RGB8A1, EAC_R11_UNSIGNED, EAC_R11_SIGNED</td></tr>
<tr><td>Texture size maximum</td><td>8k x 8k</td></tr>
<tr><td>Addressing modes</td><td>wrap, mirror, clamp</td></tr>
<tr><td>Mipmap support</td><td>14 mipmap levels; programmable LOD biasing and replacement</td></tr>
<tr><td>Shadow texture</td><td>Depth texture PCF filtering</td></tr>
<tr><td>Texture coordinate fraction bits</td><td>5 bits</td></tr>
<tr><td>Texture samplers</td><td>32 samplers, indexable</td></tr>
<tr><td>Textures per fragment maximum</td><td>16 texture samplers</td></tr>
<tr><td>Dependent texture operation</td><td>High performance; unlimited dependent texture reads</td></tr>
<tr><td>Dependent tx per fragment max, relative sampling</td><td>No Limit</td></tr>
<tr><td>Texture repeat max</td><td>256</td></tr>
<tr><td>Texture types</td><td>2D, seamless cube map, 1D, projected, depth, bump map, displacement map, PCF, 3D, texture array</td></tr>
<tr><td>Texture filters</td><td>Point sample, bi-linear, fast tri-linear (2 pixels/cycle), quad-linear, Anisotropic</td></tr>
<tr><td>Texture component mapping: D3D, OGL ES options</td><td>Supports both D3D and OES options</td></tr>
<tr><td>Texture size types</td><td>Power-of-2, Non-square, Non-power-of-2</td></tr>
<tr><td>Texture swizzle</td><td>Yes</td></tr>
<tr><td>OES3 half-float support</td><td>Yes</td></tr>
</table>

13.3.2.6 Fragment Processing and Render Targets

Table 13-15. Fragment Processing and Render Target Features

<table>
<tr><th>Feature</th><th>GPU Support</th></tr>
<tr><td>FSAA mechanisms</td><td>MSAA 4x, SSAA 4x</td></tr>
<tr><td>Fragment color, alpha, Z, stencil precision</td><td>
<table>
<tr><th>Bits</th><th>Format</th><th>Alpha</th><th>R</th><th>B</th><th>G</th></tr>
<tr><td>16</td><td>ARGB4444</td><td>4</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>XRGB4444</td><td>4 don't care</td><td>4</td><td>4</td><td>4</td></tr>
<tr><td>16</td><td>ARGB1555</td><td>1</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>XRGB1555</td><td>1 don't care</td><td>5</td><td>5</td><td>5</td></tr>
<tr><td>16</td><td>RGB565</td><td>0</td><td>5</td><td>6</td><td>5</td></tr>
<tr><td>16</td><td>RF16</td><td>0</td><td>16</td><td>0</td><td>0</td></tr>
<tr><td>32</td><td>ARGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>sRGB8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XRGB8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>ABGR8888</td><td>8</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>XBGR8888</td><td>8 don't care</td><td>8</td><td>8</td><td>8</td></tr>
<tr><td>32</td><td>RGB10_a2ui</td><td>2</td><td>10</td><td>10</td><td>10</td></tr>
<tr><td>32</td><td>RF16GF16</td><td>0</td><td>16</td><td>16</td><td>0</td></tr>
<tr><td>32</td><td>RF32</td><td>0</td><td>32</td><td>0</td><td>0</td></tr>
<tr><td>64</td><td>RF32GF32</td><td>0</td><td>32</td><td>32</td><td>0</td></tr>
<tr><td>64</td><td>RF16GF16BF16AF16</td><td>16</td><td>16</td><td>16</td><td>16</td></tr>
</table>
<table>
<tr><th>Bits</th><th>Format</th><th>Depth</th><th>Stencil</th></tr>
<tr><td>16</td><td>D16</td><td>16</td><td>0</td></tr>
<tr><td>32</td><td>D24S8</td><td>24</td><td>8</td></tr>
</table>
Render Target no blend formats: A8, R8G8, RI8, RI16, RI32, RI8GI8, RI16GI16, RI32GI32, RI8GI8BI8AI8, RI16GI16BI16AI16, RI10GI10BI10AI2, RF11GF11BF10
</td></tr>
<tr><td>Fragment storage / Render Target formats</td><td>16-bit color and z, 32-bit color and z for each fragment. Lossless compression, no storage reduction.</td></tr>
<tr><td>Individual fragment alpha masking</td><td>Yes</td></tr>
<tr><td>Two sided stencil support</td><td>Yes</td></tr>
<tr><td>Compressed formats</td><td>Yes, 30% - 70%</td></tr>
<tr><td>Fast clear and initialization options</td><td>Yes</td></tr>
<tr><td>Multiple Render Target support</td><td>Yes, 8 Targets</td></tr>
</table>

13.3.2.7 Destination / Alpha Blending

Table 13-16. Destination / Alpha Blending Features

Feature	GPU Support					
Destination / Resolve (3DBlit) color formats	Bits	Format	Alpha	R	B	G
	16	ARGB4444	4	4	4	4
	16	ARGB1555	1	5	5	5
	16	RGB565	0	5	6	5
	32	ARGB8888	8	8	8	8
	32	sRGB8888	8	8	8	8
	32	ABGR8888	8	8	8	8
	32	RGB10_a2ui	2	10	10	10

13.3.3 Usage Mode

The GPU should be programmed through the NXP provided driver. NXP does not provide support for software that directly programs the GPU registers. APIs for programming the GPU through the software driver are described in separate driver documentation.

13.4 HD Display Transmitter Controller (HDMI TX)

13.4.1 Overview

The HD Display Transmitter Controller IP offers multi-protocol support of standards such as HDMI, DisplayPort, eDP, with one of these standards supported at a time.

the mentioned protocol options enable switching between the modes to be applied on a system level and performed by means of software configuration.

Protocols supported by the controller are:

- HDMI 1.4 Specification
- HDMI 2.0a Specification
- DisplayPort Specification, Version 1.3
- eDP Specification, Version, 1.4
- High-bandwidth Digital Content Protection system, Mapping HDCP to HDMI, Revision 2.2

- High-bandwidth Digital Content Protection system, Mapping HDCP to DisplayPort, Revision 2.2
- High-bandwidth Digital Content Protection system, Mapping HDCP to HDMI, Revision 1.4
- High-bandwidth Digital Content Protection system, Mapping HDCP to DsiplayPort, Revision 1.3
- CEA-861-F
- IEC60958
- ARM AMBA 3 APB Specification, Version 1.0

13.4.1.1 Features

The HDMI_TX includes the following generic features:

- Compliant with HDCP2.2 (and back compatible with HDCP1.3/HDCP1.4)
- Variety of video resolutions and formats
  - Supports up to 4k2k at 60Hz resolution
  - 25-600Mhz pixel clk
  - Supports 8, 10, 12, and 16bpp
  - Supports RGB, YCbCr422 formats
- Variety of audio formats – PCM and compressed, over I2S interface
  - I2S PCM samples are converted to comply with IEC60958 sub-frames
  - Supporting I2S-TDM
- Enhanced testability using uCPU debugger (over JTAG) and Integrated Pattern Generator

HDMI 2.0a and DisplayPort 1.3 specific features:

- In HDMI mode, supports up to 600Mhz pixel clk
- In DisplayPort mode, supports up to HBR2

DisplayPort-specific features:

- 1Mbps AUX channel
- Configurable number of lanes (1, 2, or 4)
- SSC injection is supported by PHY
- Support for Single Stream-only (SST)
- 6bpp support

eDP-specific features:

- Supports backlights and multi-touch commands
- eDP DPCD registers

- Variable link rate R162/R216/R243/R270/R324/R432/R540
- Fast link training

13.4.2 Block diagram

This is a high-level block diagram illustrating the HD Display Transmitter Controller IP.

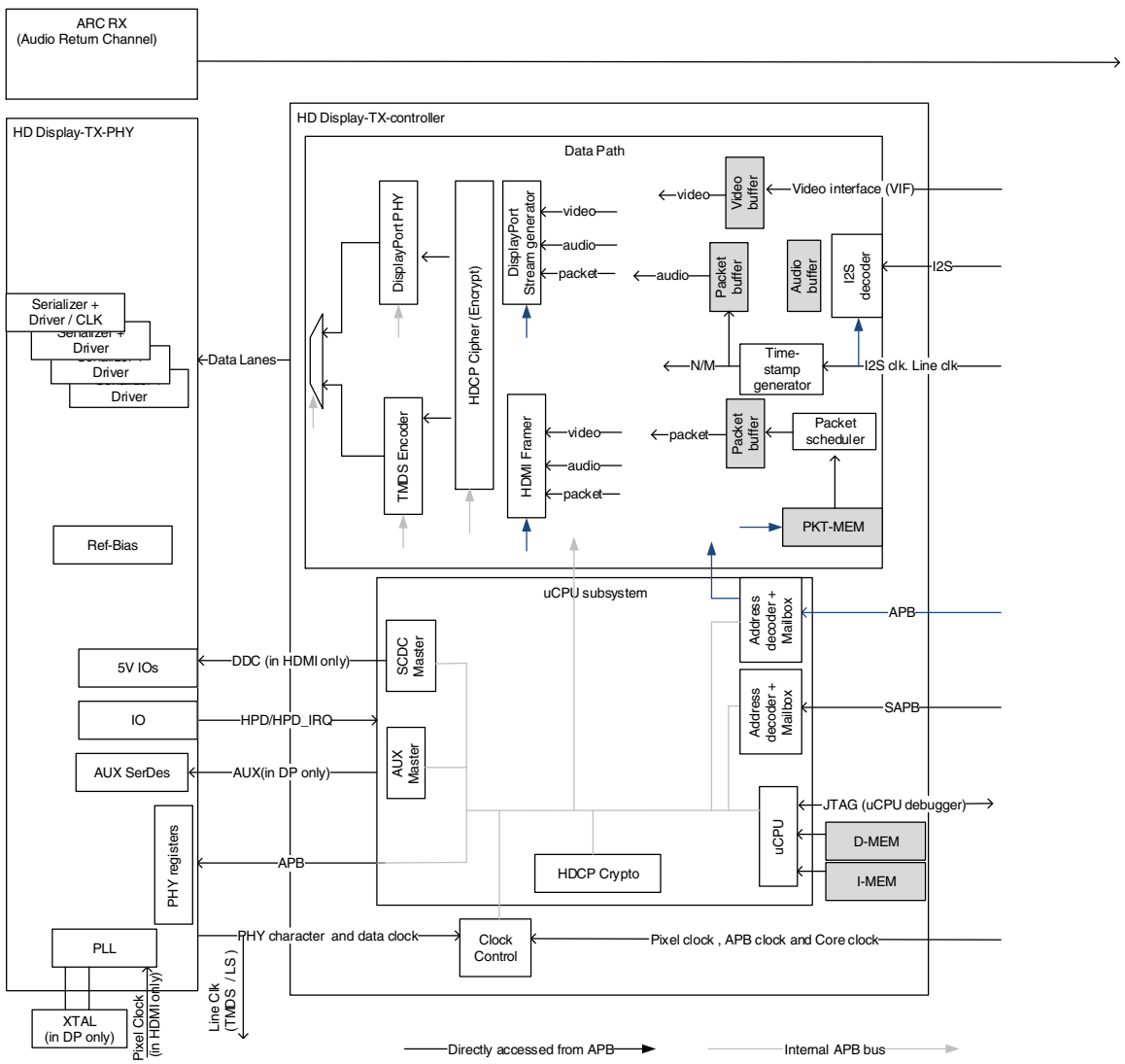


Figure 13-19. HDMI_TX block diagram

13.4.3 Video Packing

In HDMI, the Controller automatically detects the video timing from the video interface bus (VIF) and pack the video over the TMDS stream. The PHY PLL generates the character clock from the pixel clock configured by the HD Display Controller to a specific ratio (1:1, 1:1.25, 1:1.5, 1:2).

In DisplayPort, the Controller is configured by the external host with the video timing of the video interface bus (VIF) and packs the video over the main-stream. Video time stamp information (Nvid, Mvid) is generated by the Controller based on the ratio between the pixel clock and the ls clock.

The video data is transparently mapped to the video stream, where the Controller does not perform any color conversion.

13.4.4 Audio Packing

Audio info-frames and time-stamp information are generated by the Controller and encoded over the TMDS stream blanking period on HDMIsecondary stream in Displayport.

The Audio time-stamp information (NCTS in HDMI Maud and Naud in DisplayPort) is generated by the controller based on the ratio between the audio sample clock and the link clock (tmds in HDMIls clock in DisplayPort).

A peripheral module converting I2S stream into AIF bus can easily interface with the Controller.

13.4.5 Info-Frame Packing

Info-frames are encoded over the TMDS stream blanking period on HDMI the secondary stream in DisplayPort.

Time sensitive info-frames (i.e. GC-packet, CTS in HDMI) are generated autonomously by the Controller. Other info-frames (i.e. audio and video meta-data) are written into the PKT-MEM by the external host (directly accessed over the APB bus), to be then read and encoded by the Controller over the TMDS stream blanking period (on HDMI) the secondary stream (in DisplayPort).

Checksum (in HDMI) is parity bytes (in DisplayPort) are generated by the Controller.

13.4.6 HDCP

HDCP 2.2 is supported by the Controller. A dedicated HW provides the cipher and the cryptographic acceleration, the HDCP authentication state-machine is implemented in the FW (running over the uCPU).

13.4.7 Clock Control

The clock control block integrates clock-gates and provides clock metering functionality.

13.4.8 Functional Description

13.4.8.1 Functional description

The following sections describe functional details of the HDMI_TX module.

13.4.8.1.1 Management And Control Interface

The HD Display TX controller has the following APB slave ports that are controlled by the host processor:

- APB Slave port (APB):
  - Used as the main control interface between the host processor and the HD Display TX controller.
  - Used in different methods during boot.
  - Used for directly loading the FW into the memories.
  - Used for directly accessing several HW modules and for communicating the internal uCPU over the mailbox channel.
- SAPB Slave port (SAPB):
  - Used for secure connection, that is, HDCP key loading between the host processor and the uCPU through mailbox and pre-defined commands.

The internal APB bus has an output APB master port which is used to manage the PHY.

Each Mailxbox channel (APB or SAPB) generates an interrupt-level signal when its received FIFO is not empty or when its transmitted FIFO is full. The interrupt is cleared when the MAILBOX_INT_STATUS register is read. There are two sets of register, one for APB and one for SAPB.

13.4.8.1.2 Video Interface

The video interface (VIF) is compliant with CEA video timing and composed of 48-bit pixel data and the following control signals: data-enable, hsync, and vsync. The video timing of the video-interface bus (VIF) complies with the specification in CEA-861-F Annex L.

NOTE

Only a single video stream is supported

The video interface has the following control signals:

- VSYNC – Video Vertical Blanking indication signal as defined in the above standards.
- HSYNC - Video Horizontal Blanking indication signal as defined in the above standards.
- DATA ENABLE – Video data enable indication signal as defined in the above standards.
- DATA - 48 bits of data for supporting up to 16bpp.
- Clock – As defined for each format from 25MHz-600MHz.

13.4.8.1.2.1 Pixel Mapping

The following table shows the pixel mapping in different video formats.

Video Bus	RGB / YCbCr 4:4:4				4:2:0 / 4:2:2				Y Only
Bit Width	8	10	12	16	8	10	12	16 (4:2:0 only)	16/ 12/ 10/ 8
Channel 2	R/Cr				Cb/Cr				
[47:40]	Cr[15:8]	Cr[15:6]	Cr[15:8]	Cb[15:0]	C[15:8]	C[15:6]	C[15:4]	C[15:0]	
[39:38]									
[37:36]									
[35:32]									
Channel 1	G/Y				Y				
[31:24]	Y[15:8]	Y[15:6]	Y[15:4]	Y[15:0]	Y[15:8]	Y[15:6]	Y[15:4]	Y[15:0]	Y[15:0] Y[15:4] Y[15:6] Y[15:8]
[23:22]									
[21:20]									
[19:16]									
Channel 0	B/Cb								
[15:12]	Cb[15:8]	Cb[15:6]	Cb[15:4]	Cb[15:0]					
[11:8]									
[7:6]									
[5:4]									
[3:0]									

Figure 13-20. Pixel Mapping

NOTE

- In YCbCr4:2:2 or YCbCr4:2:0 the Cb and Cr are alternating as defined in timing diagram below.
- 6bpp in all formats and 16bpp in YCbCr422 format are supported only in DisplayPortoreDP.

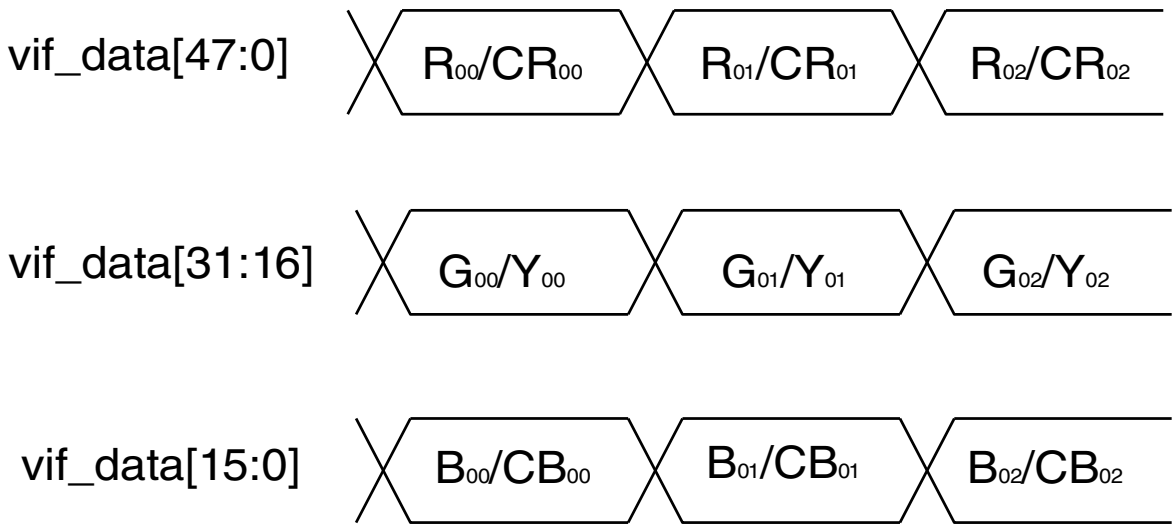


Figure 13-21. YCbCr 4:4:4 pixel mapping

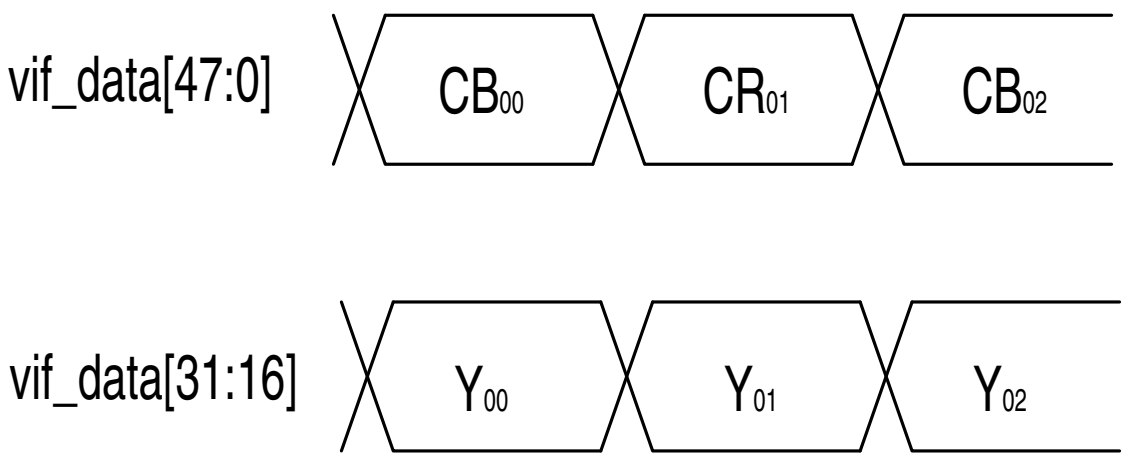


Figure 13-22. YCbCr 4:2:2 pixel mapping

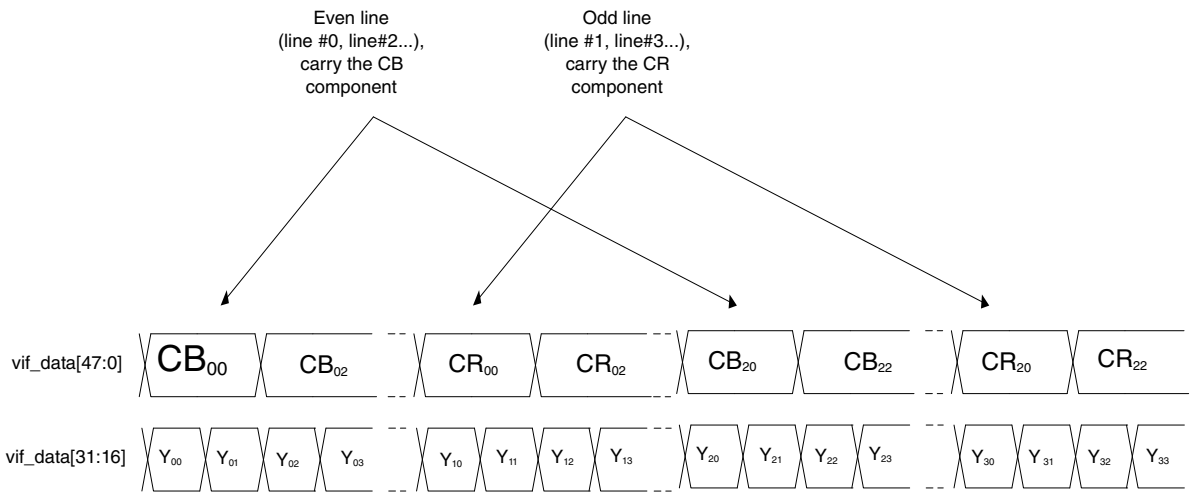


Figure 13-23. YCbCr 4:0:0 pixel mapping

13.4.8.1.2.2 Interlaced Video Timing

The HD Display TX controller expect the interlaced timing format to comply with the specification defined in CEA-861-F. The field information will be extracted from the input signal (hsync, vsync and de) and will be propagated towards the framer (i.e VB-ID field bit in DisplayPort).

13.4.8.1.2.3 3D Stereo Video Timing

The HD Display TX controller expect the stereo timing format to comply with one of the following formats:

- Frame-packing (in HD1.4)/Stacked-Top (in DP1.3).
- Top-Bottom (in HD1.4).
- Side-By-Side (in DP1.3 or HD1.4).

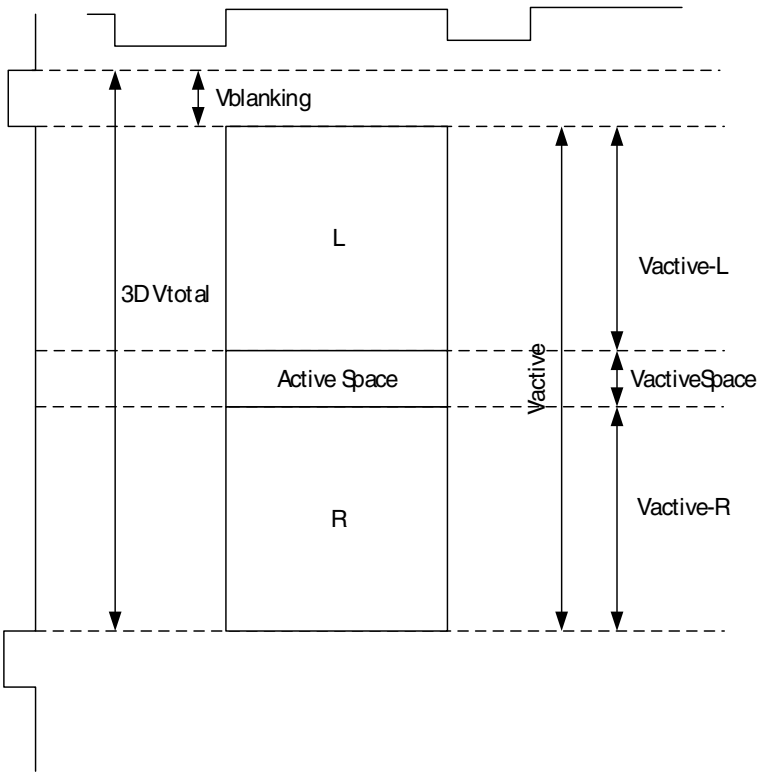


Figure 13-24. Frame-packing/Stacked-Top 3D Video Timing

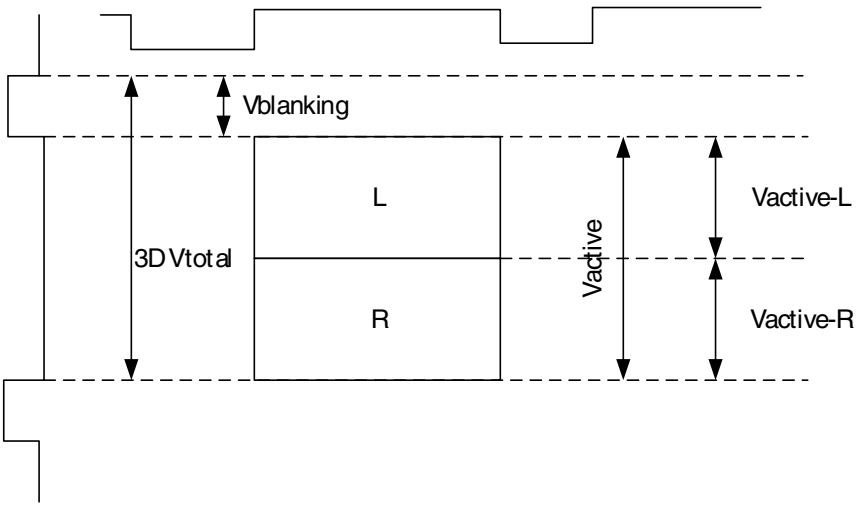


Figure 13-25. Top-Bottom 3D Video Timing

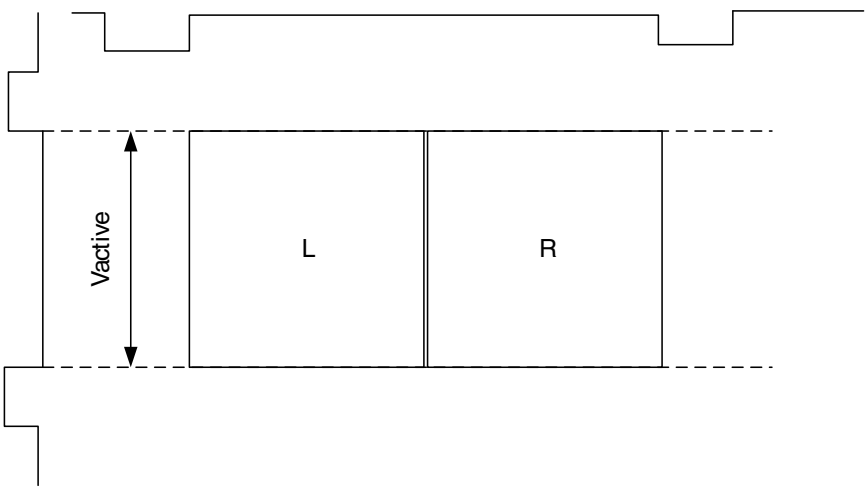


Figure 13-26. Side-By-Side 3D Video Timing

In the DisplayPort mode, the HD Display TX controller can convert the Frame-packing video timing into Field-sequential (in DP1.3). While converting into Field-sequential, the Vactive-space height should be equal to the Vertical-blank height.

13.4.8.1.2.4 Color Space Conversion Module

An add-on Color Space Conversion (CSC) module is optionally provided, it supports the video-timing defined in CEA-861-F.

The following conversion operations are supported:

- Color space conversion from RGB to YCbCr
- Color space conversion from YCbCr to RGB
- YCbCr up-sampling from 4:2:2 to 4:4:4
- YCbCr down-sampling from 4:4:4 to 4:2:2

The CSC module is configurable over its dedicated APB slave port.

13.4.8.1.3 Audio Interface

The Audio interface can be configured to support I2S.

When I2S is supported, the integrated I2S decoder extracts the PCM audio samples and converts them into 32-bit SPDIF sub-frame format compliant with IEC60958 to be carried over the data-stream. The Validity Flag (V), User Data (U), and the Channel Status (C) fields are configured by the external host. Parity bit (P) is calculated by the HD Display Transmitter Controller IP .

The I2S interface supports configurable slot time (16, 24, 32 bits), up to 8 channels, where each audio sample can occupy up to 24bits (right justified).

The interface is composed from three signals – I2S-data[3:0], I2S-bit-clk, and I2S-ws.

NOTE

Only a single audio stream is supported.

The sequence of operation is as follows:

1. Samples are decoded.
2. buffered (compensating for the video active period).
3. Mapped into an audio info-frame packet.

13.4.8.1.3.1 I2S Interface

The I2S interface could be configured to support multiple physical channels, i2s_data[n], (n=1, 2, 4), where channels could be configured with variable TDM time slots (m = 2 or 8). Time slot size is 32, 24 or 16 bits, supporting variable word length (28, 24, 20, and 16) configured as left or right justified.

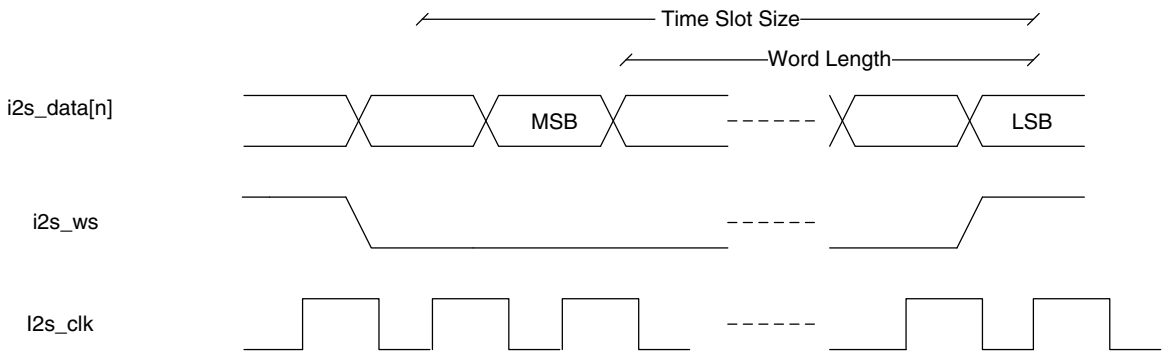


Figure 13-27. I2S bit Allocation (right justification)

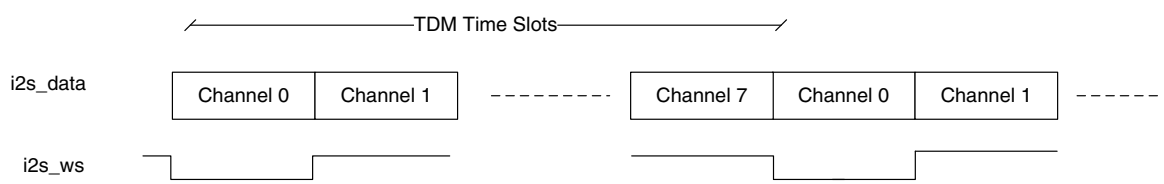


Figure 13-28. I2S TDM time allocation (M=8)

The I2S interface can carry audio samples in one of the following formats:

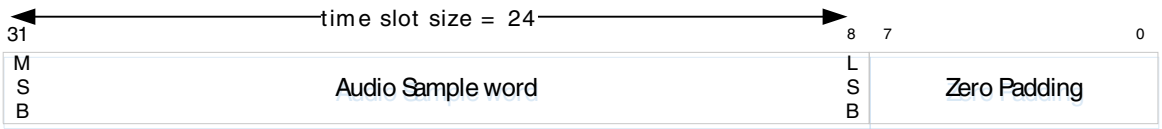


Figure 13-29. Audio L-PCM Sample Format

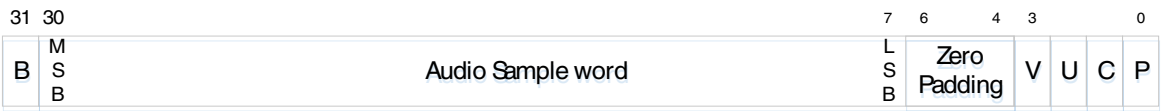


Figure 13-30. Audio SPDIF Sample Format

13.4.8.1.3.2 Audio Path

The following block diagram represents the audio path from I2S interface to creation of audio info-frame.

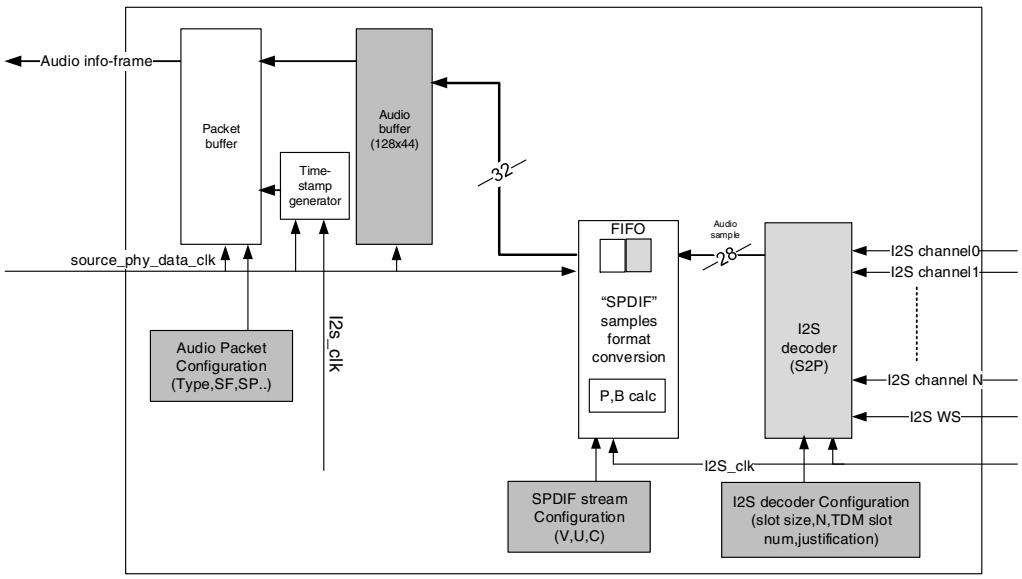


Figure 13-31. Audio Path

13.4.8.1.3.3 Audio Buffer Size

The Audio buffer is used to store audio samples during video active period, its default configuration is 128 samples (each sample occupy 44 bits).

You can configure its size to meet different worst case scenarios by using the following formula:

Audio buffer size (in samples)= $MAX[H_{active} * MaxFrameRate * 2]/F_{pixel}$

That is, if VIC10 (2880x480i) represents a worst-case scenario, its related video/audio parameters are (refer to Table 7-5 in HDMI1.4b):

Hactive = 2880, Fpixel = 54.05MHz, Max Frame Rate = 768Khz (i.e 8 ch L-PCM @192Khz)

This results in 82 samples.

13.4.8.1.4 Info-frame Processing

Info-frames are encoded over the video stream vertical blanking period.

Time-sensitive info-frames, that is, GC-packet and CTS in HDMI are generated autonomously by the HD Display TX controller.

Other info-frames, that is, audio and video meta-data are written into the PKT-MEM by the host processor (directly accessed over the APB bus) only when changed, to be then read and encoded by the HD Display TX controller packet-scheduler over the stream once per frame.

Checksum (in HDMI)Parity bytes (in DisplayPort) are generated by the HD Display TX controller.

Up to 16 info-frames are supported. When writing info-frames into the PKT-MEM the host must update the packet allocation table.

13.4.8.1.5 PHY Interface

The Controller interfaces to the PHY over interfaces described in the following sections:

13.4.8.1.5.1 Data Lanes

In HDMI, three data lanes are supported.

In DisplayPort, 1, 2, or 4 data lanes are active – actual number of active lanes is configured during link-training.

Each data lane is of 20-bit width is composed of two characters and is synchronous to phy_data_clk.

13.4.8.1.5.2 APB Master Interface

The Controller has control over the PHY by means of an APB interface.

13.4.8.1.5.3 AUX Interface

The AUX master supports various protocols: native-AUX, I2C over AUX.

13.4.8.1.5.4 DDC Interface

DDC interface is used for reading the sink EDID content and to carry the HDCP and SCDC commands.

13.4.8.1.5.5 HPD/HDP_IRQ

The HPD signal is input to the Controller and provides the HPD functionality in HDMIHPD_IRQ functionality in (DisplayPort mode).

Glitch filtering is integrated in the Controller.

13.4.8.1.6 APB Slave Interface

The Controller interfaces to the SoC as a slave over the APB bus. Up to three APB slave ports are supported.

The primary APB slave interface used for the main external host to IP communication where mailbox is used for communicating with the embedded FW (control and status), packet memory is directly mapped to the APB bus and I-MEM / D-MEM are directly accessed during boot mode (for FW download) . An interrupt signal is driven by the Controller.

The secondary APB slave is used to carry designated commands which are part of the HDCP functionality, i.e. key loading and content encryption control.

13.4.9 Clocks And Resets

13.4.9.1 Clocks Overview

The section describes the clocks and the resets of the HD Display TX IP.

13.4.9.1.1 Overview

The clock domains in the IP are as shown:

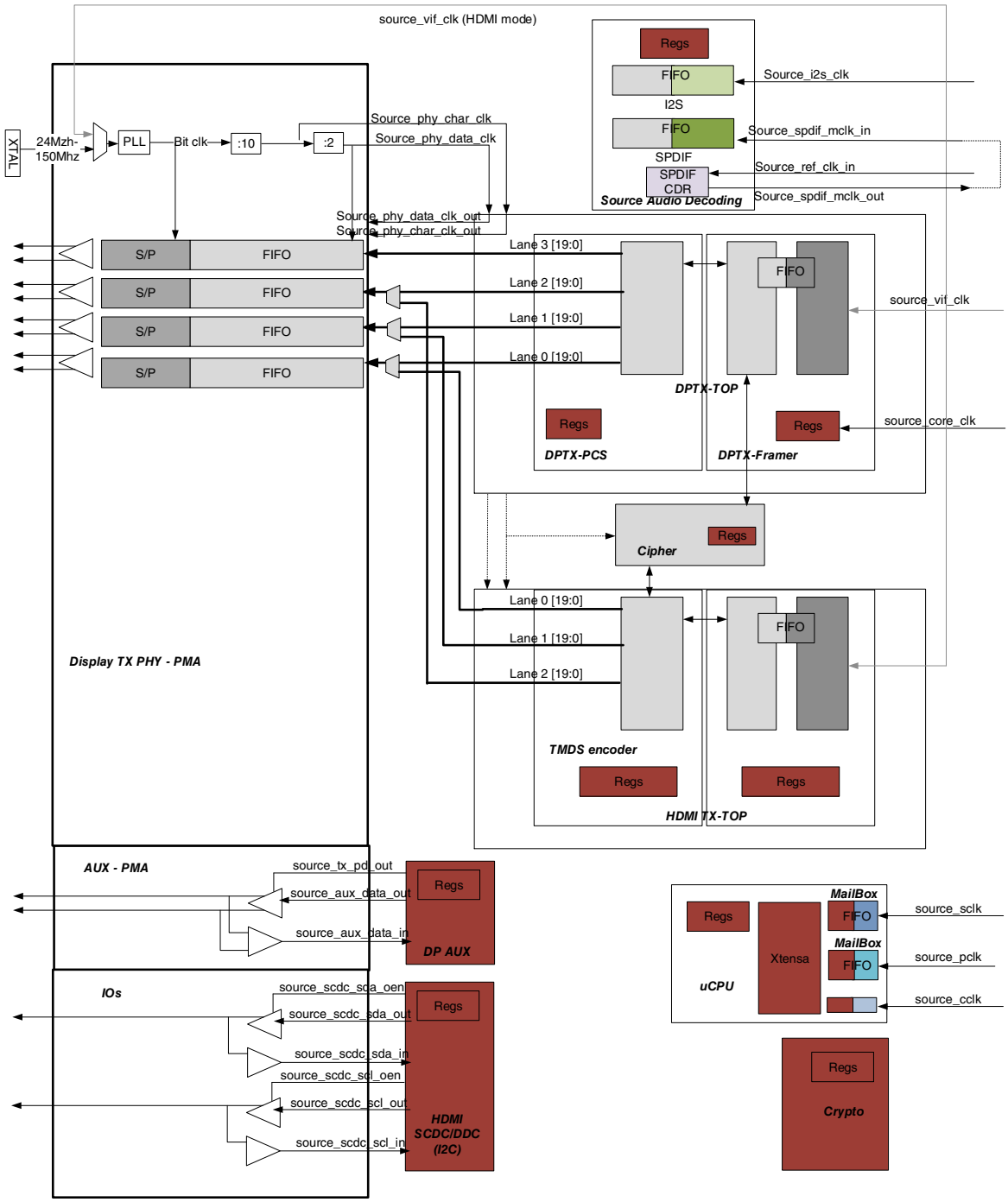


Figure 13-32. HD Display TX Clock Domains

NOTE

- Clock-gates are not represented in the clock scheme.
- Generated clocks are considered as the same clock domain.

The following table shows the list of all input clocks.

Table 13-17. Input Clock Signals

SignalName	Frequency	Source	Description
source_core_clk	100- 200Mhz	Host	main HDDisplay TX controller clock, usedby theuCPUandbythe internal register bus
source_pclk	50-200MHz	Host	APB slave clock
source_sclk	50-200MHz	Host	SAPB slave clock
source_cclk	50-200MHz	Host	CAPB slave clock
xt_jtck	<10Mhz	JTAG debugger	Debugger JTAG clock
source_ref_clk_in	200-400Mhz	Host	SPDIF CDR reference clock. Must be >10 times the SPDIF BMC rate (i.e for 192Khz , 10x192Khzx64x2 = 245.76Mhz)
source_spdif_mclk_in	<24.57Mhz	SPDIF CDR	SPDIF BMC decoder clock , need to be connected to source_spdif_mclk_out after scan clk mux insertion
source_i2s_clk	<50Mhz	Host	Audio I2S stream clock
source_vif_clk	24-600Mhz	Host	Video Pixel Clock
source_phy_char_clk	In DP: 162/270/540 /810MHZ In HDMI: <600Mhz	PHY	Lane clock (in DP:ls_clk) generated by the PHY and passed to use in the HD Display TX controller. In HDMI source_phy_char_clk is derived from source_vif_clk (ratio defined by HDMI standard) but its phase is independent , in DP independent clock
source_phy_data_clk	In DP: 81/135/270/ 405MHZ In HDMI: <300Mhz	PHY	Lane clock divided by two generated by the PHY and passed to use in the HD Display TX controller. source_phy_data_clk is derived from source_phy_char_clk but its phase is independent

The following table shows the clock signals that are internally generated in the CAR module.

Table 13-18. Generated Clock Signals

SignalName	Frequency	Source	Description
source_aux_clk_out	2MHz	source_core_clk	Not connected– for future use. Divided by configurable number according to the source_core_clk frequency
source_spdif_mclk_out	<24.57Mhz	source_ref_clk_in	Generated by SPDIF CDR, need to be fed-back (after scan clk muxing) to source_spdif_mclk_in
source_xt_pclk	100- 200Mhz	source_core_clk	PHY apb clock

13.4.10 Memory Map and register definition

This section includes the HDMI_TX module memory map and detailed descriptions of all registers.

13.4.10.1 HDMI TX register descriptions

13.4.10.1.1 HDMI_TX memory map

Offset	Register	Width (In bits)	Access	Reset value
0h	APB Control Register (APB_CTRL)	32	RW	0000_0007h
4h	Internal CPU Interrupt Polarity Control Register (XT_INT_CTRL)	32	RW	0000_0000h
8h	Mailboxes full indication register (MAILBOX_FULL_ADDR)	32	RO	0000_0000h
Ch	Mailboxes Empty indication register (MAILBOX_EMPTY_ADDR)	32	RO	0000_0001h
10h	Write Data to Mailbox register (MAILBOX0_WR_DATA)	32	RW	0000_0000h
14h	Mailbox Read data register (MAILBOX0_RD_DATA)	32	RO	0000_0000h
18h	Software keep alive counter (KEEP_ALIVE)	32	RO	0000_0000h
1Ch	Software Version LSB (VER_L)	32	RO	0000_0000h
20h	Software Version MSB (VER_H)	32	RO	0000_0000h
24h	Software Lib version written by CPU (LSB) (VER_LIB_L_ADDR)	32	RO	0000_0000h
28h	Software Lib version written by CPU (MSB) (VER_LIB_H_ADDR)	32	RO	0000_0000h
2Ch	Software Debug Register H (SW_DEBUG_L)	32	RO	0000_0000h
30h	Software Debug Register L (SW_DEBUG_H)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
34h	Mailbox Interupt mask register (MAILBOX_INT_MASK)	32	RW	0000_0000h
38h	Mailbox Interupt Status register (MAILBOX_INT_STATUS)	32	RO	0000_0000h
3Ch	Core Clock frequency (SW_CLK_L)	32	RW	0000_0000h
40h	Core Clock frequency (SW_CLK_H)	32	RW	0000_0064h
44h	SW Event 0 register (SW_EVENTS0)	32	RO	0000_0000h
48h	SW Event 1 register (SW_EVENTS1)	32	RO	0000_0000h
4Ch	SW Event 2 register (SW_EVENTS2)	32	RO	0000_0000h
50h	SW Event 3 register (SW_EVENTS3)	32	RO	0000_0000h
60h	Internal CPU - On Chip Debug (OCD) Ctrl Register (XT_OCD_CTRL)	32	RW	0000_0003h
64h	Internal CPU - OCD R0 mode configuration (XT_OCD_CTRL_RO)	32	RO	0000_0000h
6Ch	APB Interrupt Mask Register (APB_INT_MASK)	32	RW	0000_0007h
70h	APB interupt status register (APB_STATUS_MASK)	32	RO	0000_0000h
800h	HDMI shift pattern 3-0 (SHIFT_PATTERN_IN_3_0)	32	RW	0000_0000h
804h	HDMI shift pattern 4-7 (SHIFT_PATTERN_IN_4_7)	32	RW	0000_0000h
808h	HDMI shift pattern 9-8 with control bits (SHIFT_PATTERN_IN9_8)	32	RW	0000_0000h
80Ch	PRBS control (PRBS_CNTRL)	32	RW	0000_2222h
810h	PRBS error insertion (PRBS_ERR_INSERTION)	32	RW	0000_0000h
814h	Lane control register: swap, order, polarity (LANES_CONFIG)	32	RW	0060_001Bh
818h	PHY data select DP/HDMI and HDMI data source (PHY_DATA_SEL)	32	RW	0000_0000h
81Ch	Lane delay control (LANES_DEL_VAL)	32	RW	0000_6420h
900h	Register implemented only for configuration with HDMI. (SOURCE_HDTX_CAR)	32	RW	0000_0000h
904h	DP TX clock and reset ctrl register (SOURCE_DPTX_CAR)	32	RW	0000_0000h
908h	Source PHY clock and reset ctrl register (SOURCE_PHY_CAR)	32	RW	0000_0000h
90Ch	Register implemented only for configuration with HDMI. (SOURCE_CEC_CAR)	32	RW	0000_0000h
910h	CBUS clock and reset ctrl register. (SOURCE_CBUS_CAR)	32	RW	0000_0000h
918h	PKT clock and reset ctrl register (SOURCE_PKT_CAR)	32	RW	0000_0000h
91Ch	AIF clock and reset ctrl register (SOURCE_AIF_CAR)	32	RW	0000_0000h
920h	Cipher clock and reset ctrl register (SOURCE_CIPHER_CAR)	32	RW	0000_0000h
924h	Crypto clock and reset ctrl register (SOURCE_CRYPTO_CAR)	32	RW	0000_0000h
A00h	Clock Meter control (CM_CTRL)	32	RW	0000_0000h
A04h	I2S clock control (CM_I2S_CTRL)	32	RW	0400_1000h
A08h	SPDIF clock control (CM_SPDIF_CTRL)	32	RW	0400_1000h
A0Ch	Video clock control (CM_VID_CTRL)	32	RW	0400_8000h
A10h	Lane control (CM_LANE_CTRL)	32	RW	0000_0000h
A14h	I2S clock stable, audio clock measured (I2S_NM_STABLE)	32	RO	0000_0000h
A18h	I2S clock stable, lane clock measured (I2S_NCTS_STABLE)	32	RO	0000_0000h
A1Ch	SPDIF clock stable, audio clock measured (SPDIF_NM_STABLE)	32	RO	0000_0000h
A20h	SPDIF clock stable, lane clock measured (SPDIF_NCTS_STABLE)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
A24h	Video clock stable (NMVID_MEAS_STABLE)	32	RO	0000_0000h
A28h	Video cycles measure (CM_VID_MEAS)	32	RW	0000_0000h
A2Ch	Audio cycles measure (CM_AUD_MEAS)	32	RW	0000_0000h
A40h	I2S clock measurment HDMI (I2S_MEAS)	32	RO	0000_0000h
A44h	I2S clock measurment DP (I2S_DP_MEAS)	32	RO	0000_0000h
A80h	SPDIF clock measurment DP (SPDIF_DP_MEAS)	32	RO	0000_0000h
A84h	SPDIF clock measurment HDMI (SPDIF_MEAS)	32	RO	0000_0000h
AC0h	Video clock measurment (NMVID_MEAS)	32	RO	0000_0000h
B00h	Video Input Interface Setting Register (BND_HSYNC2VSYNC)	32	RW	0000_2000h
B04h	Status of HSYNC to VSYNC Distance Counter 1 (HSYNC2VSYNC_F1_L1)	32	RO	0000_0000h
B08h	Status of HSYNC to VSYNC Distance Counter 2 (HSYNC2VSYNC_F2_L1)	32	RO	0000_0000h
B0Ch	Video Interface Status Register (HSYNC2VSYNC_STATUS)	32	RO	0000_0000h
B10h	Setting Polarity of HSYNC and VSYNC (HSYNC2VSYNC_POL_CTRL)	32	RW	0000_0000h
1000h	Video HSize configuration (SCHEDULER_H_SIZE)	32	RW	FFFF_FFFFh
1004h	Video VSize configuration (SCHEDULER_V_SIZE)	32	RW	FFFF_FFFFh
1008h	Video Blank (Keep Out) Size configuration (SCHEDULER_KEEP_OUT)	32	RW	08E3_F83Ah
100Ch	Video Blank (Front) Size configuration (HDTX_SIGNAL_FRONT_WIDTH)	32	RW	FFFF_FFFFh
1010h	Video Sync Size configuration (HDTX_SIGNAL_SYNC_WIDTH)	32	RW	FFFF_FFFFh
1014h	Video Back Size configuration (HDTX_SIGNAL_BACK_WIDTH)	32	RW	FFFF_FFFFh
1018h	Video Mode configuration (HDTX_CONTROLLER)	32	RW	000C_1F91h
101Ch	HDMi HDCP configuration (HDTX_HDCP)	32	RW	0000_2100h
1020h	HDMI HPD configuration (HDTX_HPD)	32	RW	0000_000Fh
1024h	HDMI CLOCK configuration - REG0 (HDTX_CLOCK_REG_0)	32	RW	0000_7C1Fh
1028h	HDMI CLOCK configuration - REG1 (HDTX_CLOCK_REG_1)	32	RW	0000_7C1Fh
102Ch	HDMI HPD status (HPD_PLUG_IN)	32	RO	0000_0000h
1030h	HDMI status (HDCP_IN)	32	RO	0000_0100h
1034h	HDMI GCP coding configuration (GCP_FORCE_COLOR_DEPTH_CODING)	32	RW	0000_0000h
1038h	HDMI SSCP positionining (SSCP_POSITIONING)	32	RW	0002_0005h
103Ch	HDMI HDCP window opportunity (HDCP_WIN_OF_OPP_POSITION)	32	RW	0000_3DFFh
1800h	HDMI I2C configuration (CTRL)	32	RW	0000_0000h
1804h	HDMI I2C status (STATUS)	32	RO	0000_0000h
1808h	HDMI I2C Address configuration (I2C_ADDR)	32	RW	0000_0000h
180Ch	HDMI I2C data register (I2C_DATA)	32	RW	0000_0000h
1810h	HDMI I2C interrupt status (INT_STATUS)	32	RO	0000_0000h
1814h	HDMI I2C transaction size (TRANS_SIZE)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1818h	HDMI I2C Slave Monitor (SLAVE_MON)	32	RW	0000_0000h
181Ch	HDMI I2C Timeout (TIMEOUT)	32	RW	0000_001Fh
1820h	HDMI SCDC interrupt mask (INT_SCDC_MASK)	32	RO	0001_F2FFh
1824h	HDMI I2C interrupt enable (INT_ENABLE)	32	RW	0000_0000h
1828h	HDMI I2C interrupt disable (INT_DISABLE)	32	RW	0000_0000h
182Ch	HDMI I2C glitch filter configuration (GLITCH_FILTER_CTRL)	32	RW	0000_0000h
1830h	HDMI I2C RR configuration (RR_CTRL)	32	RW	0000_0000h
1834h	HDMI I2C RR timeout (RR_TIMEOUT)	32	RW	0000_208Dh
2000h	DPTX PHY control (DP_TX_PHY_CONFIG_REG)	32	RW	0000_0000h
2004h	DPTC PHY software reset (DP_TX_PHY_SW_RESET)	32	RW	0000_0000h
2008h	Scrambler seed (DP_TX_PHY_SCRAMBLER_SEED)	32	RW	0000_FFFFh
200Ch	Custom training value bytes 1-4 (DP_TX_PHY_TRAINING_01_04)	32	RW	0000_0000h
2010h	Custom training value bytes 5-8 (DP_TX_PHY_TRAINING_05_08)	32	RW	0000_0000h
2014h	Custom training value bytes 9-10 (DP_TX_PHY_TRAINING_09_10)	32	RW	0000_0000h
2018h	Custom CP2520 SR interval (DP_TX_PHY_SR_INTERVAL)	32	RW	0000_00FCh
2100h	HPD min timer for irq, define the minimum pclk cycles that the HPD pulse will be considered as IRQ (HPD_IRQ_DET_MIN_TIMER)	32	RW	0000_C350h
2104h	HPD max timer for irq, define the maximum pclk cycles that the HPD pulse will be considered as IRQ (HPD_IRQ_DET_MAX_TIMER)	32	RW	0001_86A0h
2108h	HPD min timer for HPD detect, define the minimum pclk cycles that the HPD is low (HPD_UNPLGED_DET_MIN_TIMER)	32	RW	0003_0D40h
210Ch	Timer for detecting HPD stable, count in pclk cycles (HPD_STABLE_TIMER)	32	RW	0001_86A0h
2110h	Timer for filtering small pulses on hpd input (HPD_FILTER_TIMER)	32	RW	0000_7A12h
2114h	HPD debouncer control (HPD_DBNC_TIMER)	32	RW	0000_0000h
211Ch	Mask of HPD interrupt and status (HPD_EVENT_MASK)	32	RW	0000_0000h
2120h	HPD interrupt and status (HPD_EVENT_DET)	32	R2C	0000_0004h
2200h	Global configuration of the framer module (DP_FRAMER_GLOBAL_CONFIG)	32	RW	0000_0023h
2204h	SW reset (DP_SW_RESET)	32	RW	0000_0000h
2208h	TU related configuration (DP_FRAMER_TU)	32	RW	0000_A002h
220Ch	Video pixel format configuration (DP_FRAMER_PXL_REPR)	32	RW	0000_0102h
2210h	Polarity and 3D configuration (DP_FRAMER_SP)	32	RW	0000_0000h
2214h	Audio packet configuration (AUDIO_PACK_CONTROL)	32	RW	0000_0000h
2218h	VC table values 0-3 (MST feature) (DP_VC_TABLE_0)	32	RW	0000_0000h
221Ch	VC table values 4-7 (MST feature) (DP_VC_TABLE_1)	32	RW	0000_0000h
2220h	VC table values 8-11 (MST feature) (DP_VC_TABLE_2)	32	RW	0000_0000h
2224h	VC table values 12-15 (MST feature) (DP_VC_TABLE_3)	32	RW	0000_0000h
2228h	VC table values 16-19 (MST feature) (DP_VC_TABLE_4)	32	RW	0000_0000h
222Ch	VC table values 20-23 (MST feature) (DP_VC_TABLE_5)	32	RW	0000_0000h
2230h	VC table values 24-27 (MST feature) (DP_VC_TABLE_6)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2234h	VC table values 28-31 (MST feature) (DP_VC_TABLE_7)	32	RW	0000_0000h
2238h	VC table values 32-35 (MST feature) (DP_VC_TABLE_8)	32	RW	0000_0000h
223Ch	VC table values 36-39 (MST feature) (DP_VC_TABLE_9)	32	RW	0000_0000h
2240h	VC table values 40-43 (MST feature) (DP_VC_TABLE_10)	32	RW	0000_0000h
2244h	VC table values 44-47 (MST feature) (DP_VC_TABLE_11)	32	RW	0000_0000h
2248h	VC table values 48-51 (MST feature) (DP_VC_TABLE_12)	32	RW	0000_0000h
224Ch	VC table values 52-55 (MST feature) (DP_VC_TABLE_13)	32	RW	0000_0000h
2250h	VC table values 56-59 (MST feature) (DP_VC_TABLE_14)	32	RW	0000_0000h
2254h	Video FIFO latency threshold (LINE_THRESH)	32	RW	0000_0020h
2258h	Vertical blanking ID (DP_VB_ID)	32	RW	0000_0009h
225Ch	LVP insertion in MTP header enable (MST feature) (DP_MTPH_LVP_CONTROL)	32	RW	0000_0000h
2260h	Values to be inserted in the MTP header (MST feature) (DP_MTPH_SYMBOL_VALUES)	32	RW	0000_0000h
2264h	ECF insertion in MTP header enable (MST feature) (DP_MTPH_ECF_CONTROL)	32	RW	0000_0000h
2268h	Supporting configuration to switch from top/bottom on the input to field sequential on the output (DP_FIELDSEQ_3D)	32	RW	0000_0000h
226Ch	MTP header status (MST feature) (DP_MTPH_STATUS)	32	RO	0000_5555h
2270h	Interrupt sources of the framer module, active high. (DP_INTERRUPT_SOURCE)	32	RO	0000_0000h
2274h	Masks for the interrupt sources in the DP_INTERRUPT_SOURCE register, when set high, these bits disable the corresponding interrupts (DP_INTERRUPT_MASK)	32	RW	0000_0000h
2278h	MSA parameters to be used in the HW (DP_FRONT_BACK_PORCH)	32	RW	0010_003Ch
227Ch	Number of bytes per lane/chunk parameters (DP_BYTE_COUNT)	32	RW	0000_02A4h
2280h	MSA horizontal parameters first part (MSA_HORIZONTAL_0)	32	RW	0000_0000h
2284h	MSA horizontal parameters second part (MSA_HORIZONTAL_1)	32	RW	0000_0000h
2288h	MSA vertical parameters first part (MSA_VERTICAL_0)	32	RW	0000_0000h
228Ch	MSA vertical parameters second part (MSA_VERTICAL_1)	32	RW	0000_0000h
2290h	MISC0 and MISC1 values of the MSA, as per DPv1. (MSA_MISC)	32	RW	0002_0000h
2294h	MSA stream number (MST feature) (STREAM_CONFIG)	32	RW	0000_0000h
2298h	Status signals for the audio pack (AUDIO_PACK_STATUS)	32	RW	0012_0002h
229Ch	Status signals for the VIF module (VIF_STATUS)	32	RO	0000_5555h
22A0h	Status signals for the video stuff module, part 1 (PCK_STUFF_STATUS_0)	32	RO	0101_0101h
22A4h	Status signals for the video stuff module, part 2 (PCK_STUFF_STATUS_1)	32	RO	0000_0101h
22A8h	Status signals for the info pack module, as well as final VB-ID value (INFO_PACK_STATUS)	32	RO	0000_0002h
22ACh	Rate governer status, as well as additional video configuration (RATE_GOVERNOR_STATUS)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
22B0h	Video line parameters to be used in the RTL calculations (DP_HORIZONTAL)	32	RW	0000_0000h
22B4h	Video frame parameters to be used in the RTL calculations, part 1 (DP_VERTICAL_0)	32	RW	0000_0000h
22B8h	Video frame parameters to be used in the RTL calculations, part 2 (DP_VERTICAL_1)	32	RW	0000_0000h
22BCh	(DP_BLOCK_SDP)	32	RW	0FBE_0505h
2300h	DPTX lane enable (DPTX_LANE_EN)	32	RW	0000_0000h
2304h	DPTX enhanced mode (DPTX_ENHNCD)	32	RW	0000_0000h
2308h	DPTX Interrupt mask (DPTX_INT_MASK)	32	RW	0000_0003h
230Ch	DPTX interrupt status (DPTX_INT_STATUS)	32	RO	0000_0000h
2400h	HDCP DP status register (HDCP_DP_STATUS)	32	RO	0000_0000h
2404h	HDCP DP config register (HDCP_DP_CONFIG)	32	RW	0000_0009h
2408h	HDCP DP software reset register (HDCP_DP_SW_RST)	32	RW	0000_0000h
240Ch	HDCP DP FIFO status register (HDCP_DP_FIFO_STATUS)	32	RO	0000_0000h
2800h	Control bits for DP_AUX (DP_AUX_HOST_CONTROL)	32	RW	0000_0004h
2804h	Status of the DP_AUX interrupt sources (DP_AUX_INTERRUPT_SOURCE)	32	RO	0000_0000h
2808h	Mask vector of the DP_AUX interrupt sources (DP_AUX_INTERRUPT_MASK)	32	RW	0000_0000h
280Ch	Ordering and inversion of transmit/receive on Auxiliary Channel (DP_AUX_SWAP_INVERSION_CONTROL)	32	RW	0000_0000h
2810h	NACK transaction send (DP_AUX_SEND_NACK_TRANSACTION)	32	RW	0000_0000h
2814h	RX bits clear (DP_AUX_CLEAR_RX)	32	RW	0000_0000h
2818h	TX bits clear (DP_AUX_CLEAR_TX)	32	RW	0000_0000h
281Ch	Stop timer operation (DP_AUX_TIMER_STOP)	32	RW	0000_0000h
2820h	Clear timer operation (DP_AUX_TIMER_CLEAR)	32	RW	0000_0000h
2824h	Soft reset of the DP_AUX (DP_AUX_RESET_SW)	32	RW	0000_0000h
2828h	SYS_CLK and 2 MHz clock ratio (DP_AUX_DIVIDE_2M)	32	RW	0000_000Bh
282Ch	Pre charge field length (DP_AUX_TX_PREACHARGE_LENGTH)	32	RW	0000_0010h
2830h	Maximum legal receiving frequency (DP_AUX_FREQUENCY_1M_MAX)	32	RW	0000_0022h
2834h	Minimum legal receiving frequency (DP_AUX_FREQUENCY_1M_MIN)	32	RW	0000_0009h
2838h	Minimum received preamble length (DP_AUX_RX_PRE_MIN)	32	RW	0000_001Ah
283Ch	Maximum received preamble length (DP_AUX_RX_PRE_MAX)	32	RW	0000_0020h
2840h	DP_AUX_MAIN start value (DP_AUX_TIMER_PRESET)	32	RW	0000_1D4Ch
2844h	Transmit pattern (DP_AUX_NACK_FORMAT)	32	RW	0000_0020h
2848h	Mailbox write data (DP_AUX_TX_DATA)	32	RW	0000_0000h
284Ch	Mailbox read data (DP_AUX_RX_DATA)	32	RO	0000_0000h
2850h	AUX_TX status (DP_AUX_TX_STATUS)	32	RO	0000_0101h
2854h	AUX_RX status (DP_AUX_RX_STATUS)	32	RO	0101_0141h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2858h	RX counter status (DP_AUX_RX_CYCLE_COUNTER)	32	RO	0000_0000h
285Ch	DP_AUX MAIN State Machines status (DP_AUX_MAIN_STATES)	32	RO	0000_0000h
2860h	DP_AUX MAIN timer status (DP_AUX_MAIN_TIMER)	32	RO	0000_0000h
2864h	Test mode configuration (DP_AUX_AFE_OUT)	32	RW	0000_0000h
3800h	CEC TX message header (TX_MSG_HEADER)	32	RW	0000_0000h
3804h	CEC TX message opcode (TX_MSG_OPCODE)	32	RW	0000_0000h
3808h	CEC TX message operand - byte 1 (TX_MSG_OP1)	32	RW	0000_0000h
380Ch	CEC TX message operand - byte 2 (TX_MSG_OP2)	32	RW	0000_0000h
3810h	CEC TX message operand - byte 3 (TX_MSG_OP3)	32	RW	0000_0000h
3814h	CEC TX message operand - byte 4 (TX_MSG_OP4)	32	RW	0000_0000h
3818h	CEC TX message operand - byte 5 (TX_MSG_OP5)	32	RW	0000_0000h
381Ch	CEC TX message operand - byte 6 (TX_MSG_OP6)	32	RW	0000_0000h
3820h	CEC TX message operand - byte 7 (TX_MSG_OP7)	32	RW	0000_0000h
3824h	CEC TX message operand - byte 8 (TX_MSG_OP8)	32	RW	0000_0000h
3828h	CEC TX message operand - byte 9 (TX_MSG_OP9)	32	RW	0000_0000h
382Ch	CEC TX message operand - byte 10 (TX_MSG_OP10)	32	RW	0000_0000h
3830h	CEC TX message operand - byte 11 (TX_MSG_OP11)	32	RW	0000_0000h
3834h	CEC TX message operand - byte 12 (TX_MSG_OP12)	32	RW	0000_0000h
3838h	CEC TX message operand - byte 13 (TX_MSG_OP13)	32	RW	0000_0000h
383Ch	CEC TX message operand - byte 14 (TX_MSG_OP14)	32	RW	0000_0000h
3840h	CEC TX message length (TX_MSG_LENGTH)	32	RW	0000_0000h
3844h	CEC TX message command (TX_MSG_CMD)	32	RW	0000_0000h
3848h	CEC TX buffer write enable (TX_WRITE_BUF)	32	RW	0000_0000h
384Ch	CEC TX buffer clear (TX_CLEAR_BUF)	32	RW	0000_0000h
3850h	CEC RX message command (RX_MSG_CMD)	32	RW	0000_0000h
3854h	CEC RX buffer clear (RX_CLEAR_BUF)	32	RW	0000_0000h
3858h	CEC Logical Address 0 (LOGICAL_ADDRESS_LA0)	32	RW	0000_000Fh
385Ch	CEC Logical Address 1 (LOGICAL_ADDRESS_LA1)	32	RW	0000_000Fh
3860h	CEC Logical Address 2 (LOGICAL_ADDRESS_LA2)	32	RW	0000_000Fh
3864h	CEC Logical Address 3 (LOGICAL_ADDRESS_LA3)	32	RW	0000_000Fh
3868h	CEC Logical Address 4 (LOGICAL_ADDRESS_LA4)	32	RW	0000_000Fh
386Ch	CEC Clock Divider - MSB part (CLK_DIV_MSB)	32	RW	0000_0003h
3870h	CEC Clock Divider - LSB part (CLK_DIV_LSB)	32	RW	0000_00E7h
387Ch	CDC mesage enable (cdc_msg)	32	RW	0000_0000h
3900h	CEC RX message header (RX_MSG_DATA1)	32	RO	0000_0000h
3904h	CEC RX message opcode (RX_MSG_DATA2)	32	RO	0000_0000h
3908h	CEC RX message operand 1 (RX_MSG_DATA3)	32	RO	0000_0000h
390Ch	CEC RX message operand 2 (RX_MSG_DATA4)	32	RO	0000_0000h
3910h	CEC RX message operand 3 (RX_MSG_DATA5)	32	RO	0000_0000h
3914h	CEC RX message operand 4 (RX_MSG_DATA6)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3918h	CEC RX message operand 5 (RX_MSG_DATA7)	32	RO	0000_0000h
391Ch	CEC RX message operand 6 (RX_MSG_DATA8)	32	RO	0000_0000h
3920h	CEC RX message operand 7 (RX_MSG_DATA9)	32	RO	0000_0000h
3924h	CEC RX message operand 8 (RX_MSG_DATA10)	32	RO	0000_0000h
3928h	CEC RX message operand 9 (RX_MSG_DATA11)	32	RO	0000_0000h
392Ch	CEC RX message operand 10 (RX_MSG_DATA12)	32	RO	0000_0000h
3930h	CEC RX message operand 11 (RX_MSG_DATA13)	32	RO	0000_0000h
3934h	CEC RX message operand 12 (RX_MSG_DATA14)	32	RO	0000_0000h
3938h	CEC RX message operand 13 (RX_MSG_DATA15)	32	RO	0000_0000h
393Ch	CEC RX message operand 14 (RX_MSG_DATA16)	32	RO	0000_0000h
3940h	CEC RX message length (RX_MSG_LENGTH)	32	RO	0000_0000h
3944h	CEC RX status (RX_MSG_STATUS)	32	RO	0000_0000h
3948h	CEC RX status - number of received messages (NUM_OF_MSG_RX_BUF)	32	RO	0000_0000h
394Ch	CEC TX status (TX_MSG_STATUS)	32	RO	0000_0000h
3950h	CEC TX status - number of messages in TX buffer (NUMBER_OF_MSG_IN_TX_BUF)	32	RO	0000_0000h
3954h	CEC Free-Time counter (FREE_TIME_CNTR_CONFIG)	32	RW	0000_0022h
3958h	CEC Interrupt Mask (INT_CEC_MASK)	32	RW	0000_0003h
395Ch	CEC Interrupt status (INT_STAT)	32	RO	0000_0000h
3980h	CEC DB L timer (DB_L_TIMER)	32	RW	0000_0000h
3984h	CEC DB M timer (DB_M_TIMER)	32	RW	0000_0000h
3988h	CEC DB H timer (DB_H_TIMER)	32	RW	0000_0000h
4000h	Contains the revision of the internal HDCP 1. (CRYPTO_HDCP_REVISION)	32	RO	0000_0000h
4004h	Contains global configuration information for the HDCP Crypto module (HDCP_CRYPTO_CONFIG)	32	RW	0000_0000h
4008h	Contains the status of the HDCP interrupt sources (CRYPTO_INTERRUPT_SOURCE)	32	RO	0000_0000h
400Ch	CContains the mask vector of the HDCP interrupt sources (CRYPTO_INTERRUPT_MASK)	32	RW	0000_0703h
4018h	Contains global configuration information for the HDCP 2. (CRYPTO22_CONFIG)	32	RW	0000_0000h
401Ch	Crypto 2. (CRYPTO22_STATUS)	32	RO	0000_0000h
403Ch	Holds 32-bit input data word of the SHA-256 module (SHA_256_DATA_IN)	32	RW	0000_0000h
4050h	Result of operation SHA-256 - 1' dw (SHA_256_DATA_OUT_0)	32	RO	0000_0000h
4054h	Result of operation SHA-256 - 2' dw (SHA_256_DATA_OUT_1)	32	RO	0000_0000h
4058h	Result of operation SHA-256 - 3' dw (SHA_256_DATA_OUT_2)	32	RO	0000_0000h
405Ch	Result of operation SHA-256 - 4' dw (SHA_256_DATA_OUT_3)	32	RO	0000_0000h
4060h	Result of operation SHA-256 - 5' dw (SHA_256_DATA_OUT_4)	32	RO	0000_0000h
4064h	Result of operation SHA-256 - 6' dw (SHA_256_DATA_OUT_5)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4068h	Result of operation SHA-256 - 7' dw (SHA_256_DATA_OUT_6)	32	RO	0000_0000h
406Ch	Result of operation SHA-256 - 8' dw (SHA_256_DATA_OUT_7)	32	RO	0000_0000h
4070h	Input key word of the AES-32 module - 1' dw (AES_32_KEY_0)	32	RW	0000_0000h
4074h	Input key word of the AES-32 module - 2' dw (AES_32_KEY_1)	32	RW	0000_0000h
4078h	Input key word of the AES-32 module - 3' dw (AES_32_KEY_2)	32	RW	0000_0000h
407Ch	Input key word of the AES-32 module - 4' dw (AES_32_KEY_3)	32	RW	0000_0000h
4080h	Input data word to the AES-32 module (AES_32_DATA_IN)	32	RW	0000_0000h
4084h	AES-32 module - 128-bits output data word - 1' dw (AES_32_DATA_OUT_0)	32	RO	0000_0000h
4088h	AES-32 module - 128-bits output data word - 2' dw (AES_32_DATA_OUT_1)	32	RO	0000_0000h
408Ch	AES-32 module - 128-bits output data word - 3' dw (AES_32_DATA_OUT_2)	32	RO	0000_0000h
4090h	AES-32 module - 128-bits output data word - 4' dw (AES_32_DATA_OUT_3)	32	RO	0000_0000h
40A0h	Contains global configuration information for the HDCP 1. (CRYPTO14_CONFIG)	32	RW	0000_0000h
40A4h	Contains global status information for the HDCP 1. (CRYPTO14_STATUS)	32	RO	0000_0000h
40A8h	Contains 24-bit pseudo random data (CRYPTO14_PRNM_OUT)	32	RO	0000_0000h
40ACh	Contains the first word of the Km value (CRYPTO14_KM_0)	32	RO	0000_0000h
40B0h	Contains the most significant 3 bytes of the Km value (CRYPTO14_KM_1)	32	RO	0000_0000h
40B4h	First word of An value generated by hdcpRngCipher operation. (CRYPTO14_AN_0)	32	RW	0000_0000h
40B8h	Second word of An value generated by hdcpRngCipher operation (CRYPTO14_AN_1)	32	RW	0000_0000h
40BCh	First 32 bits of the KSV from the other HDCP device (CRYPTO14_YOUR_KSV_0)	32	RW	0000_0000h
40C0h	Last byte of the KSV from other HDCP device (CRYPTO14_YOUR_KSV_1)	32	RW	0000_0000h
40C4h	Mi value - 1' dw (CRYPTO14_MI_0)	32	RO	0000_0000h
40C8h	Mi value - 2' dw (CRYPTO14_MI_1)	32	RO	0000_0000h
40CCh	Ti value (CRYPTO14_TI_0)	32	RO	0000_0000h
40D0h	First 32 bits of the Ki frame key from this HDCP device (CRYPTO14_KI_0)	32	RO	0000_0000h
40D4h	Last 3 bytes of the Ki frame key from this HDCP device. (CRYPTO14_KI_1)	32	RO	0000_0000h
40D8h	This register defines number of iterations for SHA-1 calculations (CRYPTO14_BLOCKS_NUM)	32	RW	0000_0000h
40DCh	Key memory control register. (CRYPTO14_KEY_MEM_DATA_0)	32	RW	0000_0000h
40E0h	Key memory control register. (CRYPTO14_KEY_MEM_DATA_1)	32	RW	0000_0000h
40E4h	SHA1 message control register. (CRYPTO14_SHA1_MSG_DATA)	32	RW	0000_0000h
40E8h	SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_0)	32	RO	6745_2301h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
40ECh	SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_1)	32	RO	EFCD_AB89h
40F0h	SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_2)	32	RO	98BA_DCFEh
40F4h	SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_3)	32	RO	1032_5476h
40F8h	SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_4)	32	RO	C3D2_E1F0h
40FCh	TRNG control (TRNG_CTRL)	32	RW	0000_0000h
4100h	TRNG status (TRNG_DATA_RDY)	32	RO	0000_0000h
4104h	TRNG data register (TRNG_DATA)	32	RO	0000_0000h
3_0000h	Audio source control (AUDIO_SRC_CNTL)	32	RW	0000_0000h
3_0004h	Audio source configuration (AUDIO_SRC_CNFG)	32	RW	0000_0000h
3_0008h	Common channels configuration (COM_CH_STTS_BITS)	32	RW	0000_0000h
3_000Ch	Channels 0,1 configuration (STTS_BIT_CH01)	32	RW	0000_0000h
3_0010h	Channels 2,3 configuration (STTS_BIT_CH23)	32	RW	0000_0000h
3_0014h	Channels 4,5 configuration (STTS_BIT_CH45)	32	RW	0000_0000h
3_0018h	Channels 6,7 configuration (STTS_BIT_CH67)	32	RW	0000_0000h
3_001Ch	Channels 8,9 configuration (STTS_BIT_CH89)	32	RW	0000_0000h
3_0020h	Channels 10,11 configuration (STTS_BIT_CH1011)	32	RW	0000_0000h
3_0024h	Channels 12,13 configuration (STTS_BIT_CH1213)	32	RW	0000_0000h
3_0028h	Channels 14,15 configuration (STTS_BIT_CH1415)	32	RW	0000_0000h
3_002Ch	Channels 16,17 configuration (STTS_BIT_CH1617)	32	RW	0000_0000h
3_0030h	Channels 18,19 configuration (STTS_BIT_CH1819)	32	RW	0000_0000h
3_0034h	Channels 20,21 configuration (STTS_BIT_CH2021)	32	RW	0000_0000h
3_0038h	Channels 22,23 configuration (STTS_BIT_CH2223)	32	RW	0000_0000h
3_003Ch	Channels 24,25 configuration (STTS_BIT_CH2425)	32	RW	0000_0000h
3_0040h	Channels 26,27 configuration (STTS_BIT_CH2627)	32	RW	0000_0000h
3_0044h	Channels 28,29 configuration (STTS_BIT_CH2829)	32	RW	0000_0000h
3_0048h	Channels 30,31 configuration (STTS_BIT_CH3031)	32	RW	0000_0000h
3_004Ch	SPDIF control (SPDIF_CTRL_ADDR)	32	RW	0000_0000h
3_0050h	SPDIF channel 1 status [31:00] (SPDIF_CH1_CS_3100_ADDR)	32	RO	0000_0000h
3_0054h	SPDIF channel 1 status [63:32] (SPDIF_CH1_CS_6332_ADDR)	32	RO	0000_0000h
3_0058h	SPDIF channel 1 status [95:64] (SPDIF_CH1_CS_9564_ADDR)	32	RO	0000_0000h
3_005Ch	SPDIF channel 1 status [127:96] (SPDIF_CH1_CS_12796_ADDR)	32	RO	0000_0000h
3_0060h	SPDIF channel 1 status [159:128] (SPDIF_CH1_CS_159128_ADDR)	32	RO	0000_0000h
3_0064h	SPDIF channel 1 status [191:160] (SPDIF_CH1_CS_191160_ADDR)	32	RO	0000_0000h
3_0068h	SPDIF channel 2 status [31:00] (SPDIF_CH2_CS_3100_ADDR)	32	RO	0000_0000h
3_006Ch	SPDIF channel 2 status [63:32] (SPDIF_CH2_CS_6332_ADDR)	32	RO	0000_0000h
3_0070h	SPDIF channel 2 status [95:64] (SPDIF_CH2_CS_9564_ADDR)	32	RO	0000_0000h
3_0074h	SPDIF channel 2 status [127:96] (SPDIF_CH2_CS_12796_ADDR)	32	RO	0000_0000h
3_0078h	SPDIF channel 2 status [159:128] (SPDIF_CH2_CS_159128_ADDR)	32	RO	0000_0000h
3_007Ch	SPDIF channel 2 status [191:160] (SPDIF_CH2_CS_191160_ADDR)	32	RO	0000_0000h
3_0080h	Sample 2 Packets Control Register (SMPL2PKT_CNTL)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3_0084h	Sample 2 Packets Config Register (SMPL2PKT_CNFG)	32	RW	0000_0800h
3_0088h	FIFO control register (FIFO_CNTL)	32	RW	0000_0000h
3_008Ch	FIFO Status register (FIFO_STTS)	32	RO	0000_0000h
3_0090h	SUB Packet Threshold register (SUB_PCKT_THRSH)	32	RW	0030_2010h
3_0800h	4 MSB of the packet memory address in which the data is written. (SOURCE_PIF_WR_ADDR)	32	RW	0000_0000h
3_0804h	Write request bit for the host write transaction. (SOURCE_PIF_WR_REQ)	32	RW	0000_0000h
3_0808h	4 MSB of the packet memory address from which the data is read. (SOURCE_PIF_RD_ADDR)	32	RW	0000_0000h
3_080Ch	Read request bit for the host read transaction. (SOURCE_PIF_RD_REQ)	32	RW	0000_0000h
3_0810h	The 32 bits of the data to be written to the packet memory. (SOURCE_PIF_DATA_WR)	32	RW	0000_0000h
3_0814h	The 32 bits of the data to be read from the packet memory. (SOURCE_PIF_DATA_RD)	32	RO	0000_0000h
3_0818h	Fifo1 flush (SOURCE_PIF_FIFO1_FLUSH)	32	RW	0000_0000h
3_081Ch	Fifo2 flush (SOURCE_PIF_FIFO2_FLUSH)	32	RW	0000_0000h
3_0820h	Status bits for the PIF module (SOURCE_PIF_STATUS)	32	RO	0000_0010h
3_0824h	Interrupt sources of the PIF module, active high. (SOURCE_PIF_INTERRUPT_SOURCE)	32	RO	0000_0000h
3_0828h	Masks for the interrupt sources in the SOURCE_PIF_INTERRUPT_SOURCE register, when set high, these bits disable the corresponding interrupts (SOURCE_PIF_INTERRUPT_MASK)	32	RW	0000_0000h
3_082Ch	Packet configuration to be stored in the allocation table (SOURCE_PIF_PKT_ALLOC_REG)	32	RW	0000_0000h
3_0830h	Enable bit for writing to the allocation table (SOURCE_PIF_PKT_ALLOC_WR_EN)	32	RW	0000_0000h
3_0834h	Software reset. (SOURCE_PIF_SW_RESET)	32	RW	0000_0000h
6_0000h	Contains the revision information of the HDCP Cipher module (HDCP_REVISION)	32	RO	0000_0000h
6_0004h	Contains the status of the HDCP Cipher interrupt sources (INTERRUPT_SOURCE)	32	RO	0000_0000h
6_0008h	Contains the mask vector of the HDCP Cipher interrupt sources. (INTERRUPT_MASK)	32	RW	0000_0000h
6_000Ch	Contains the configuration details for the Cipher core module (HDCP_CIPHER_CONFIG)	32	RW	0000_0000h
6_0010h	Holds the least significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_0)	32	RW	0000_0000h
6_0014h	Holds the second significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_1)	32	RW	0000_0000h
6_0018h	Holds the third significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_2)	32	RW	0000_0000h

Table continues on the next page...

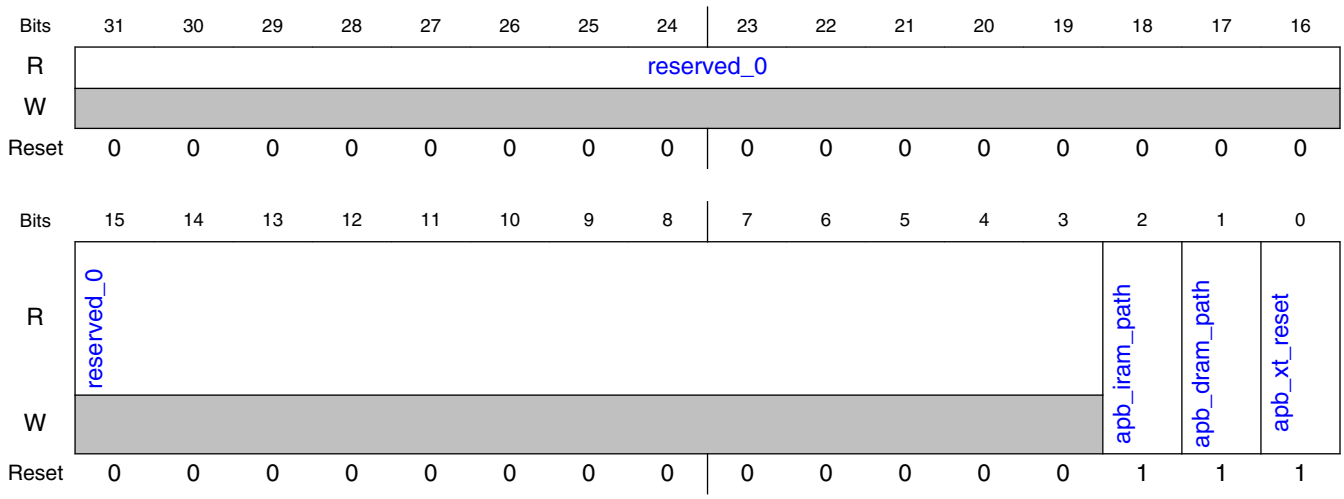
Offset	Register	Width (In bits)	Access	Reset value
6_001Ch	Holds the most significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_3)	32	RW	0000_0000h
6_0020h	Holds the least significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_RANDOM_0)	32	RW	0000_0000h
6_0024h	Holds the most significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_RANDOM_1)	32	RW	0000_0000h
6_0028h	Holds the first word of the Km value (CIPHER14_KM_0)	32	RW	0000_0000h
6_002Ch	Holds the most significant 3 bytes of the Km value (CIPHER14_KM_1)	32	RW	0000_0000h
6_0030h	Cipher 1. (CIPHER14_STATUS)	32	RO	0000_0000h
6_0034h	Cipher 1. (CIPHER14_RI_PJ_STATUS)	32	RO	0000_0000h
6_0038h	Cipher mode status register (CIPHER_MODE)	32	RO	0000_0000h
6_003Ch	Holds the First word of An value generated by hdcpRngCipher operation. (CIPHER14_AN_0)	32	RW	0000_0000h
6_0040h	Holds the second word of An value generated by hdcpRngCipher operation (CIPHER14_AN_1)	32	RW	0000_0000h
6_0044h	HDCP2. (CIPHER22_AUTH)	32	RW	0000_0000h
6_0048h	This register contains Ri value in A5: Link Integrity Check state [HDCP 1. (CIPHER14_R0_DP_STATUS)	32	RO	0000_0000h
6_004Ch	Cipher for HDCP1. (CIPHER14_BOOTSTRAP)	32	RW	0000_0000h

13.4.10.1.2 APB Control Register (APB_CTRL)

13.4.10.1.2.1 Offset

Register	Offset
APB_CTRL	0h

13.4.10.1.2.2 Diagram



13.4.10.1.2.3 Fields

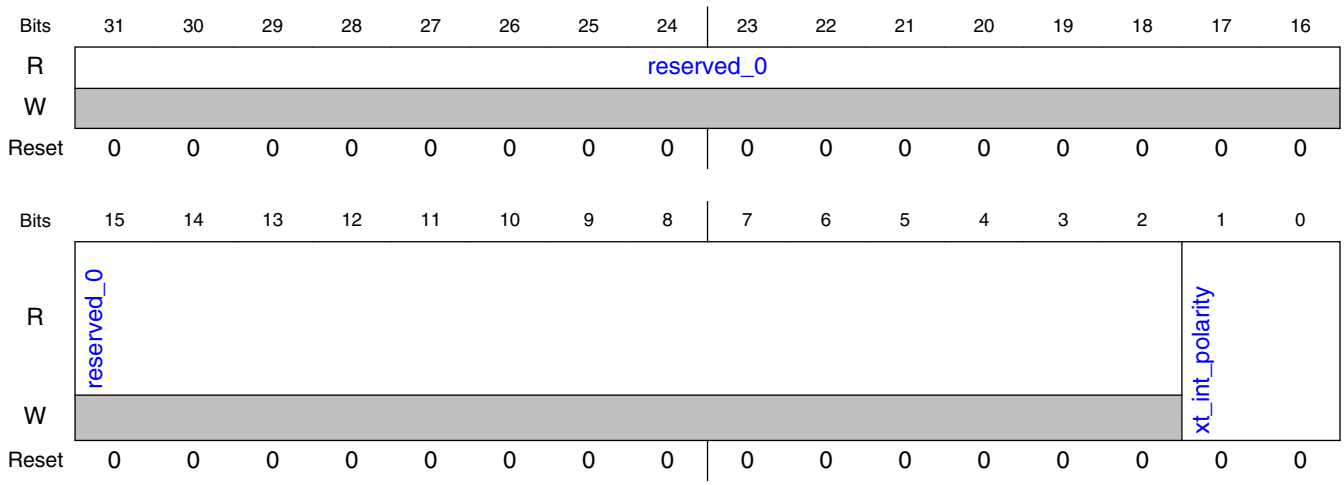
Field	Function
31-3 reserved_0	
2 apb_iram_path	When 1 then enable APB to R/W the IRAM
1 apb_dram_path	When 1 then enable APB to R/W the DRAM
0 apb_xt_reset	APB Control on the CPU reset. APB Control on the CPU reset. Active High

13.4.10.1.3 Internal CPU Interrupt Polarity Control Register (XT_INT_CTRL)

13.4.10.1.3.1 Offset

Register	Offset
XT_INT_CTRL	4h

13.4.10.1.3.2 Diagram



13.4.10.1.3.3 Fields

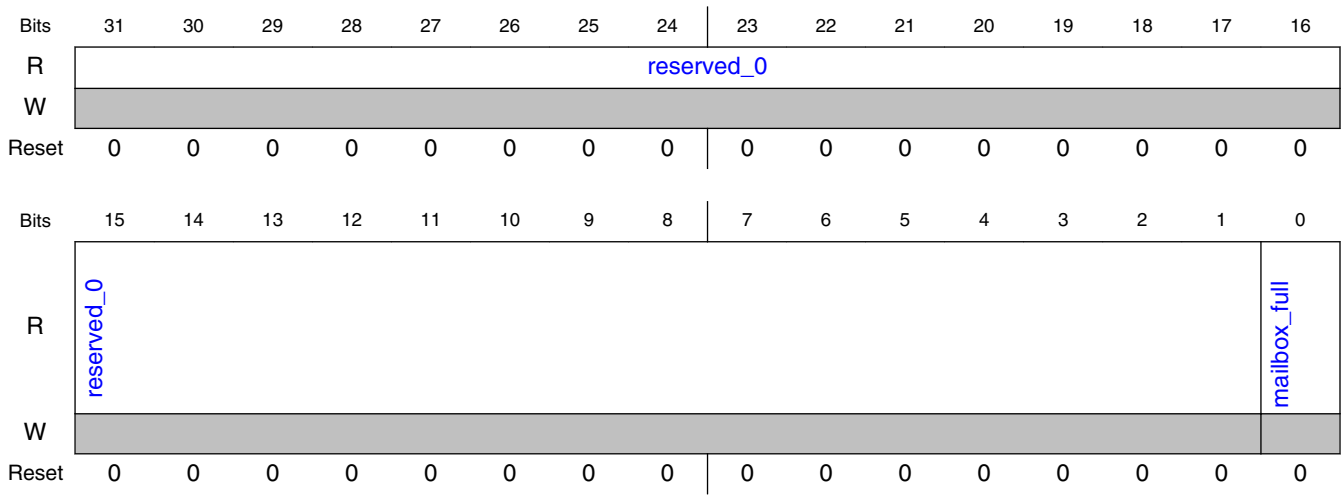
Field	Function
31-2 reserved_0	
1-0 xt_int_polarity	Each bit inverts appropriate interrupt signal provided do internal CPU interrupt input

13.4.10.1.4 Mailboxes full indication register (MAILBOX_FULL_ADDR)

13.4.10.1.4.1 Offset

Register	Offset
MAILBOX_FULL_ADDR	8h

13.4.10.1.4.2 Diagram



13.4.10.1.4.3 Fields

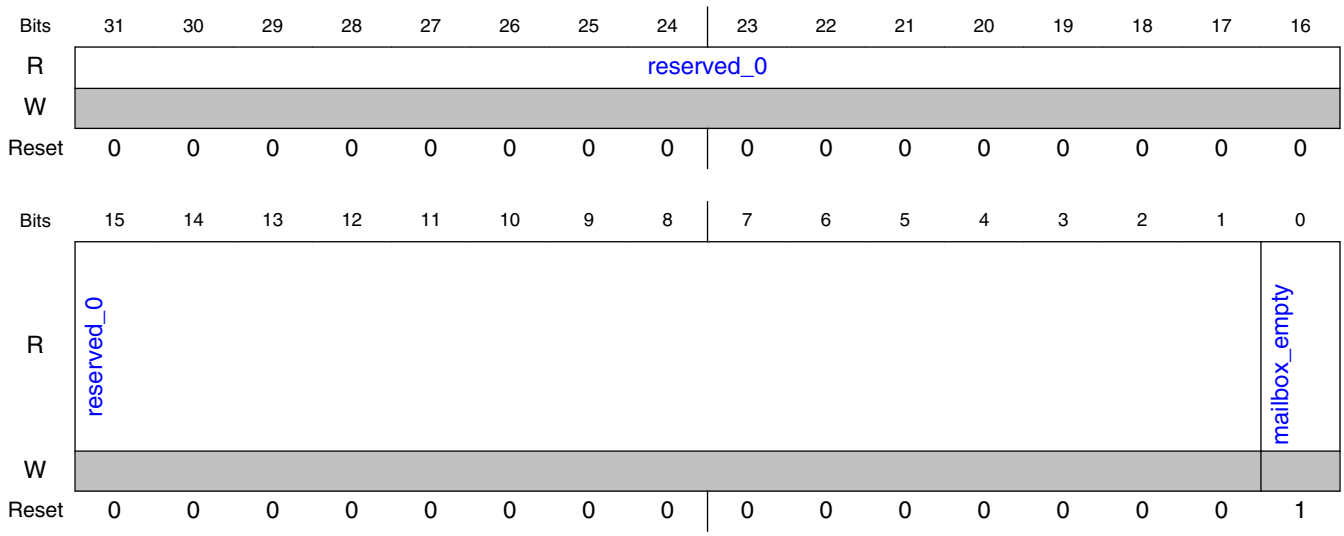
Field	Function
31-1 reserved_0	
0 mailbox_full	Mailboxes full indication

13.4.10.1.5 Mailboxes Empty indication register (MAILBOX_EMPTY_ADDR)

13.4.10.1.5.1 Offset

Register	Offset
MAILBOX_EMPTY_ADDR	Ch

13.4.10.1.5.2 Diagram



13.4.10.1.5.3 Fields

Field	Function
31-1 reserved_0	
0 mailbox_empty	Mailboxes Empty indication

13.4.10.1.6 Write Data to Mailbox register (MAILBOX0_WR_DATA)

13.4.10.1.6.1 Offset

Register	Offset
MAILBOX0_WR_DATA	10h

13.4.10.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								mailbox0_wr_data							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.6.3 Fields

Field	Function
31-8 reserved_0	
7-0 mailbox0_wr_data	Write Data to Mailbox

13.4.10.1.7 Mailbox Read data register (MAILBOX0_RD_DATA)

13.4.10.1.7.1 Offset

Register	Offset
MAILBOX0_RD_DATA	14h

13.4.10.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								mailbox0_rd_data							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.7.3 Fields

Field	Function
31-8 reserved_0	
7-0 mailbox0_rd_data	Mailbox Read data

13.4.10.1.8 Software keep alive counter (KEEP_ALIVE)

13.4.10.1.8.1 Offset

Register	Offset
KEEP_ALIVE	18h

13.4.10.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								keep_alive_cnt							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.8.3 Fields

Field	Function
31-8 reserved_0	
7-0 keep_alive_cnt	Software keep alive counter. Software keep alive counter. Counter is initialized to 0x0 after reset and incremented by 0x1 with every FW kernel loop. It can be used to determine if internal CPU started running correctly.

13.4.10.1.9 Software Version LSB (VER_L)

13.4.10.1.9.1 Offset

Register	Offset
VER_L	1Ch

13.4.10.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								ver_lsb							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.9.3 Fields

Field	Function
31-8 reserved_0	
7-0 ver_lsb	Software Version LSB. Software Version LSB. Loaded by FW

13.4.10.1.10 Software Version MSB (VER_H)

13.4.10.1.10.1 Offset

Register	Offset
VER_H	20h

13.4.10.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								ver_msb							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.10.3 Fields

Field	Function
31-8 reserved_0	
7-0 ver_msb	Software Version MSB. Software Version MSB. Loaded by FW

13.4.10.1.11 Software Lib version written by CPU (LSB) (VER_LIB_L_ADDR)

13.4.10.1.11.1 Offset

Register	Offset
VER_LIB_L_ADDR	24h

13.4.10.1.11.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_lib_ver_l							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.11.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_lib_ver_l	Software Lib version written by CPU (FW)

13.4.10.1.12 Software Lib version written by CPU (MSB) (VER_LIB_H_ADDR)

13.4.10.1.12.1 Offset

Register	Offset
VER_LIB_H_ADDR	28h

13.4.10.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_lib_ver_h							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.12.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_lib_ver_h	Software Lib version written by CPU (FW)

13.4.10.1.13 Software Debug Register H (SW_DEBUG_L)

13.4.10.1.13.1 Offset

Register	Offset
SW_DEBUG_L	2Ch

13.4.10.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_debug_7_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.13.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_debug_7_0	For software debug purposes. For software debug purposes. Not used during normal operation.

13.4.10.1.14 Software Debug Register L (SW_DEBUG_H)

13.4.10.1.14.1 Offset

Register	Offset
SW_DEBUG_H	30h

13.4.10.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_debug_15_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.14.3 Fields

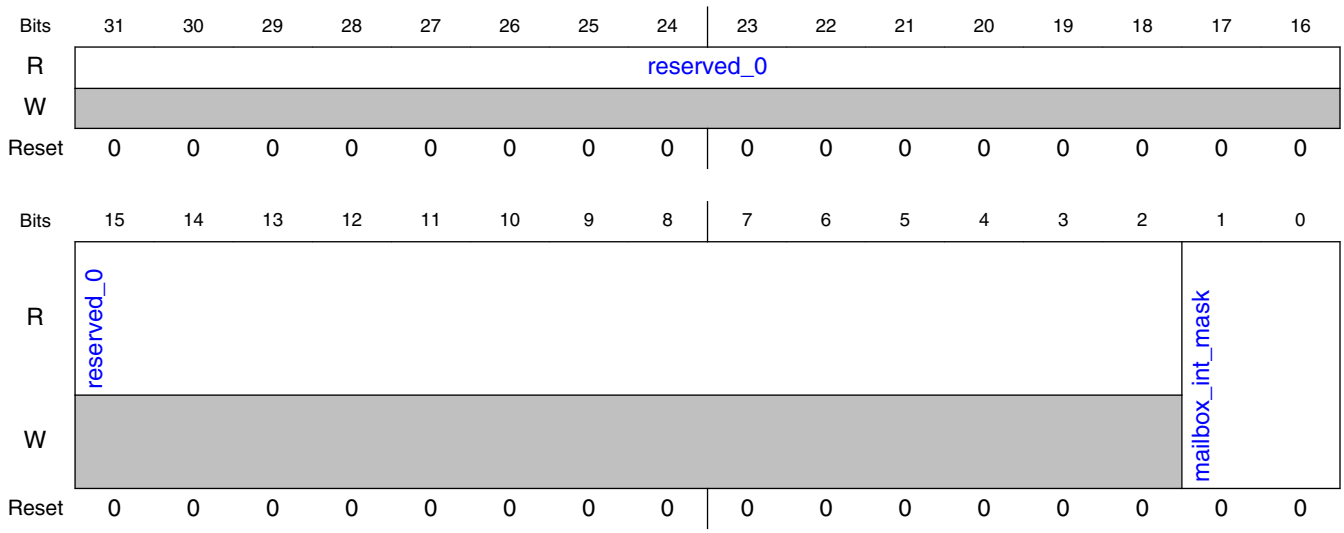
Field	Function
31-8 reserved_0	
7-0 sw_debug_15_8	For software debug purposes. For software debug purposes. Not used during normal operation.

13.4.10.1.15 Mailbox Interupt mask register (MAILBOX_INT_MASK)

13.4.10.1.15.1 Offset

Register	Offset
MAILBOX_INT_MASK	34h

13.4.10.1.15.2 Diagram



13.4.10.1.15.3 Fields

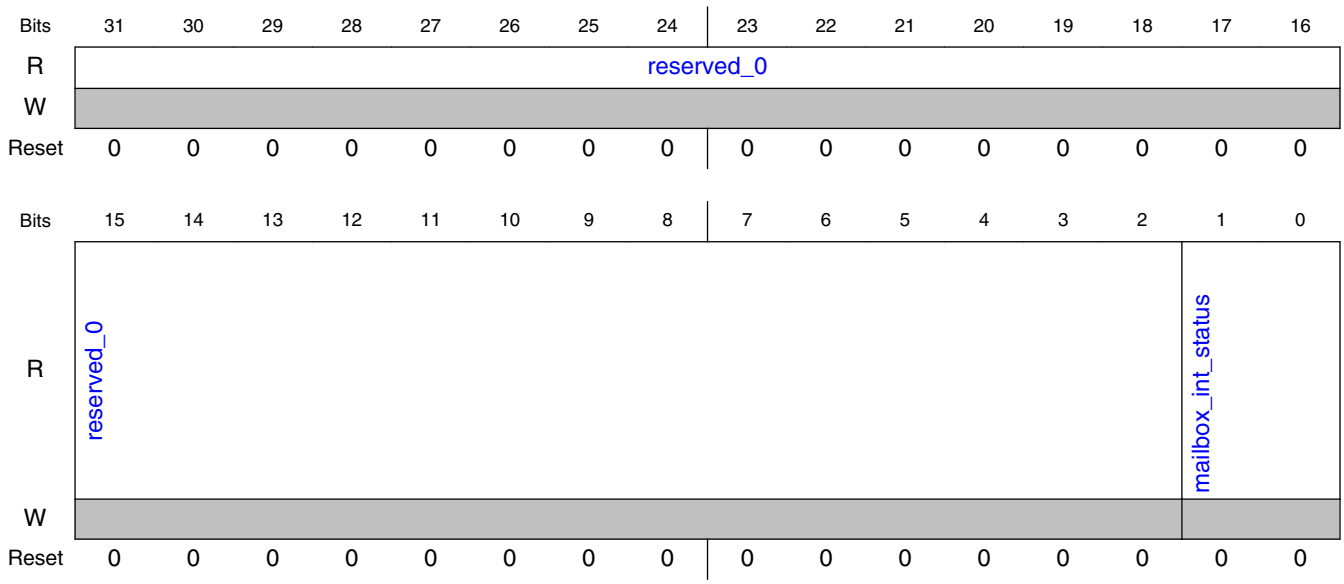
Field	Function
31-2 reserved_0	
1-0 mailbox_int_mask	Mailbox Interupt mask-Bit[0] - Empty-Bit[1] - Full

13.4.10.1.16 Mailbox Interupt Status register (MAILBOX_INT_STATUS)

13.4.10.1.16.1 Offset

Register	Offset
MAILBOX_INT_STATUS	38h

13.4.10.1.16.2 Diagram



13.4.10.1.16.3 Fields

Field	Function
31-2 reserved_0	
1-0 mailbox_int_status	Mailbox Interupt Status-Bit[0] - Empty-Bit[1] - Full-

13.4.10.1.17 Core Clock frequency (SW_CLK_L)

13.4.10.1.17.1 Offset

Register	Offset
SW_CLK_L	3Ch

13.4.10.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_clock_val_l							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.17.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_clock_val_l	Fractial of the clock decimal value. Fractial of the clock decimal value. Should be loaded by API to the value that reflects the frequency of clock provided to core clock input.

13.4.10.1.18 Core Clock frequency (SW_CLK_H)

13.4.10.1.18.1 Offset

Register	Offset
SW_CLK_H	40h

13.4.10.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_clock_val_h							
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	1	0	0

13.4.10.1.18.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_clock_val_h	Clock frequency in decimal values. Clock frequency in decimal values. Should be loaded by API to the value that reflects the frequency of clock provided to core clock input.

13.4.10.1.19 SW Event 0 register (SW_EVENTS0)

13.4.10.1.19.1 Offset

Register	Offset
SW_EVENTS0	44h

13.4.10.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_events7_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.19.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_events7_0	When SW writes it updates just the extra event bit. When SW writes it updates just the extra event bit. When Host reads it is cleared

13.4.10.1.20 SW Event 1 register (SW_EVENTS1)

13.4.10.1.20.1 Offset

Register	Offset
SW_EVENTS1	48h

13.4.10.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_events15_8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.20.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_events15_8	When SW writes it updates just the extra event bit. When SW writes it updates just the extra event bit. When Host reads it is cleared

13.4.10.1.21 SW Event 2 register (SW_EVENTS2)

13.4.10.1.21.1 Offset

Register	Offset
SW_EVENTS2	4Ch

13.4.10.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_events23_16							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.21.3 Fields

Field	Function
31-8 reserved_0	
7-0 sw_events23_16	When SW writes it updates just the extra event bit. When SW writes it updates just the extra event bit. When Host reads it is cleared

13.4.10.1.22 SW Event 3 register (SW_EVENTS3)

13.4.10.1.22.1 Offset

Register	Offset
SW_EVENTS3	50h

13.4.10.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								sw_events31_24							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.22.3 Fields

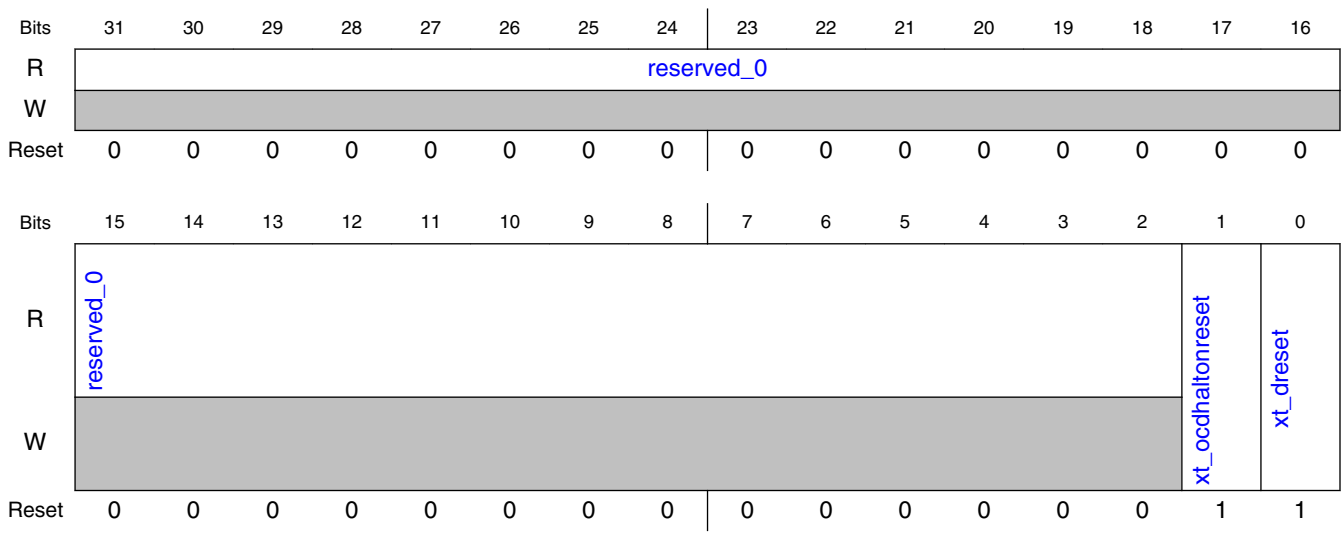
Field	Function
31-8 reserved_0	
7-0 sw_events31_24	When SW writes it updates just the extra event bit. When SW writes it updates just the extra event bit. When Host reads it is cleared

13.4.10.1.23 Internal CPU - On Chip Debug (OCD) Ctrl Register (XT_OCD_CTRL)

13.4.10.1.23.1 Offset

Register	Offset
XT_OCD_CTRL	60h

13.4.10.1.23.2 Diagram



13.4.10.1.23.3 Fields

Field	Function
31-2	

Table continues on the next page...

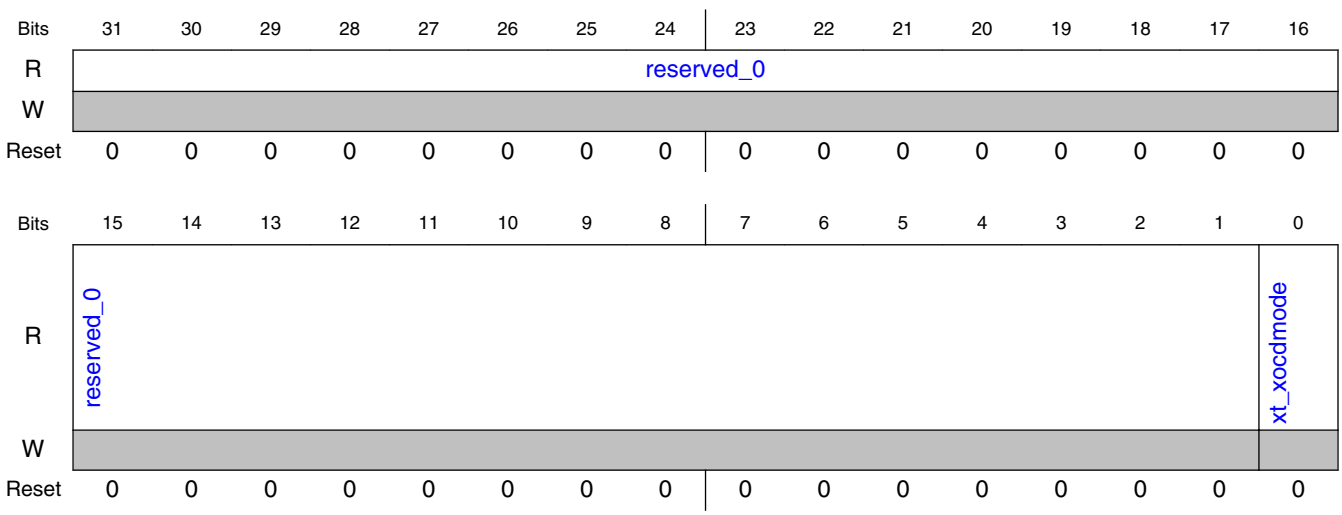
Field	Function
reserved_0	
1 xt_ocdhaltonreset	Internal CPU - Halt On Reget configuration register
0 xt_dreset	Internal CPU - Dreset control register

13.4.10.1.24 Internal CPU - OCD R0 mode configuration (XT_OCD_CTRL_RO)

13.4.10.1.24.1 Offset

Register	Offset
XT_OCD_CTRL_RO	64h

13.4.10.1.24.2 Diagram



13.4.10.1.24.3 Fields

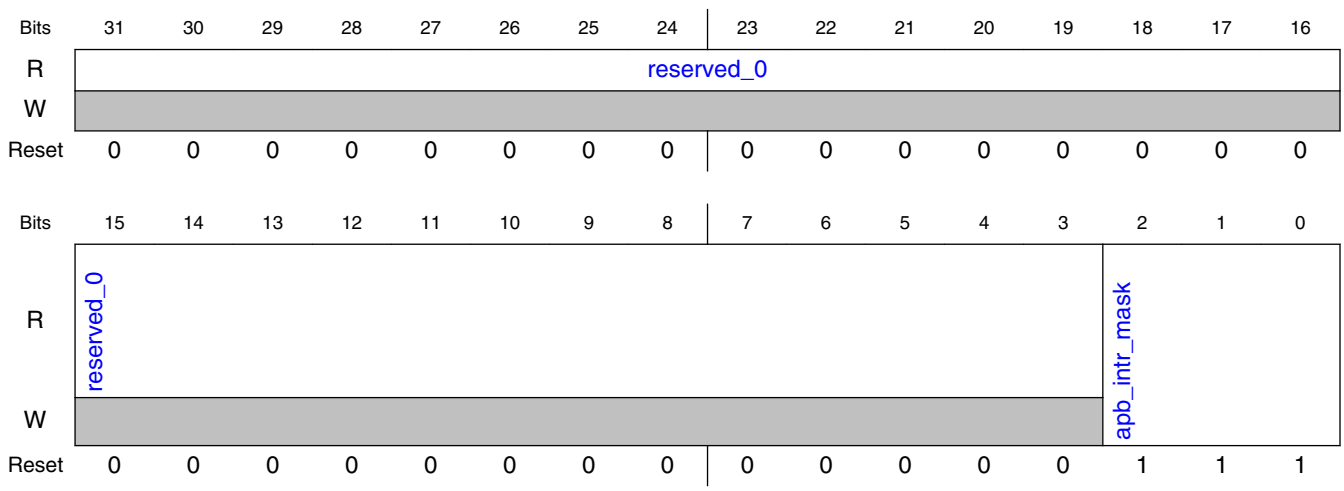
Field	Function
31-1 reserved_0	
0 xt_xocdmode	Internal CPU - OCD mode configuration

13.4.10.1.25 APB Interrupt Mask Register (APB_INT_MASK)

13.4.10.1.25.1 Offset

Register	Offset
APB_INT_MASK	6Ch

13.4.10.1.25.2 Diagram



13.4.10.1.25.3 Fields

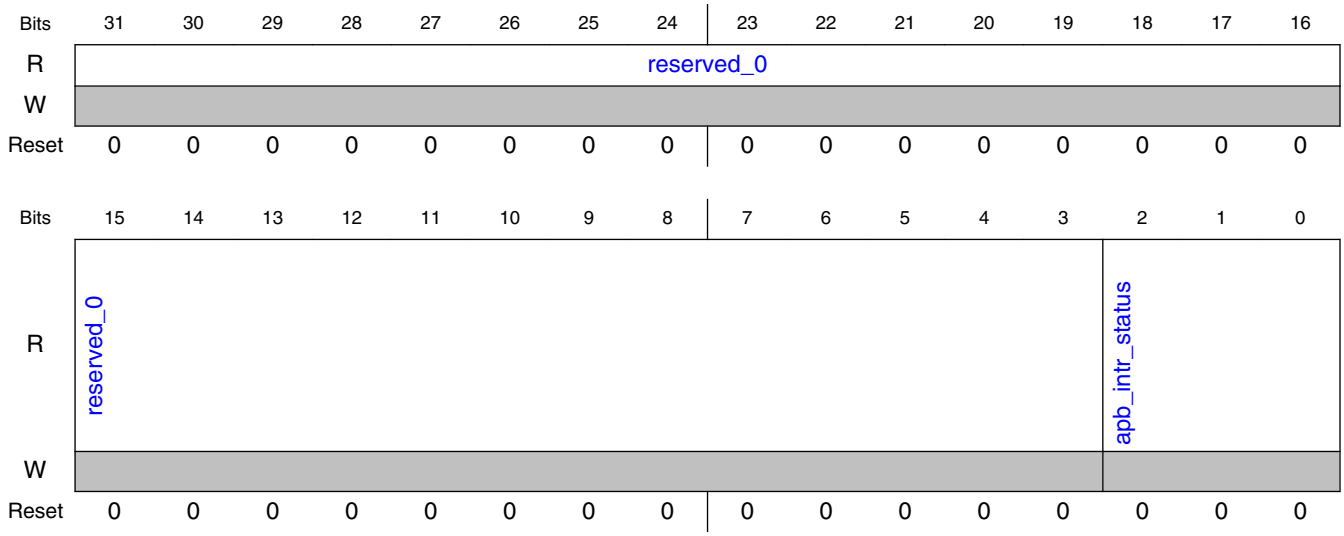
Field	Function
31-3 reserved_0	
2-0 apb_intr_mask	Mask the APB interupt-Bit0 - Mailbox Interupt-Bit1 - PIF Interupt-Bit2 - CEC Interupt

13.4.10.1.26 APB interupt status register (APB_STATUS_MASK)

13.4.10.1.26.1 Offset

Register	Offset
APB_STATUS_MASK	70h

13.4.10.1.26.2 Diagram



13.4.10.1.26.3 Fields

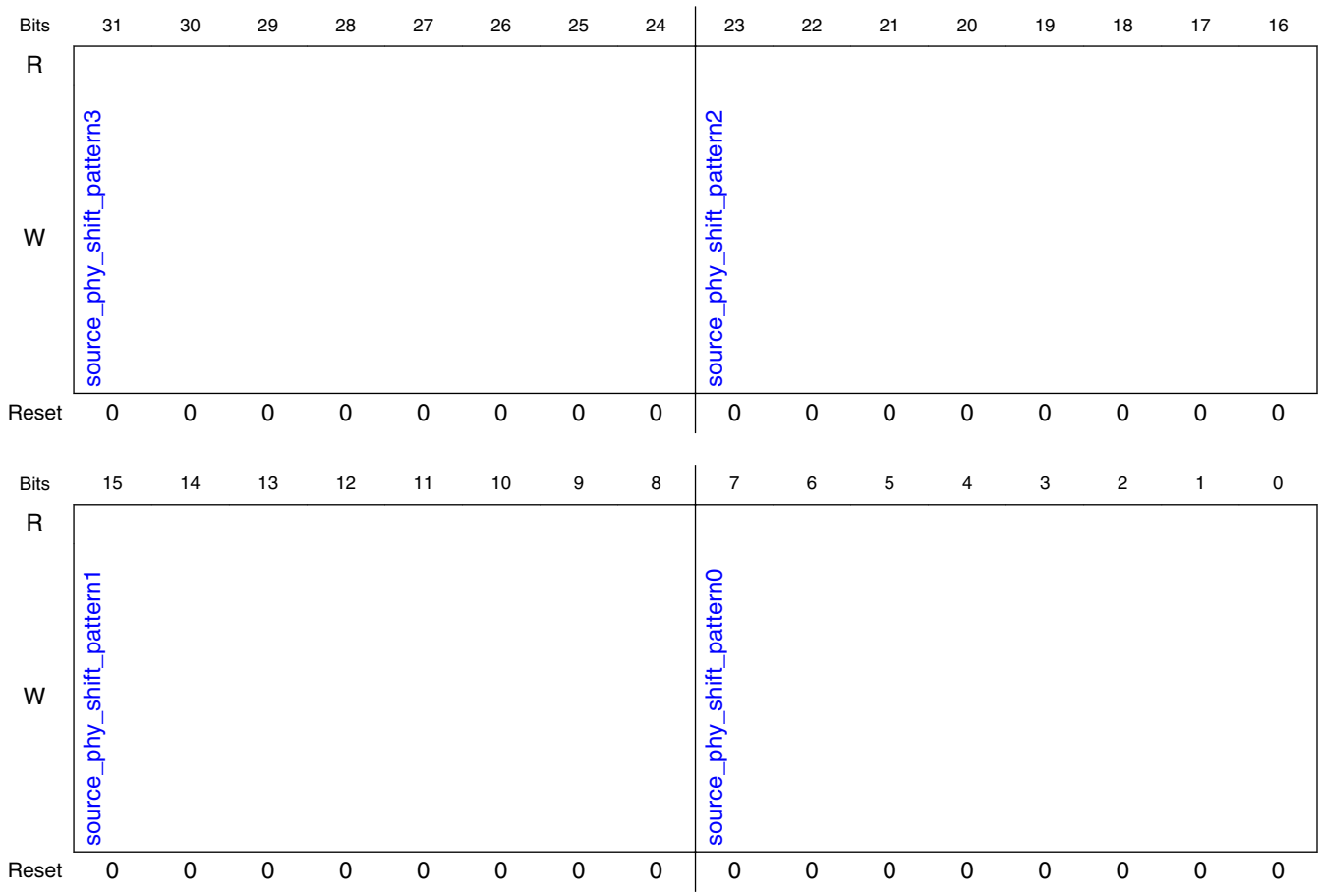
Field	Function
31-3 reserved_0	
2-0 apb_intr_status	APB interupt STATUS-Bit0 - Mailbox Interupt-Bit1 - PIF Interupt-Bit2 - CEC Interupt

13.4.10.1.27 HDMI shift pattern 3-0 (SHIFT_PATTERN_IN_3_0)

13.4.10.1.27.1 Offset

Register	Offset
SHIFT_PATTERN_IN_3_0	800h

13.4.10.1.27.2 Diagram



13.4.10.1.27.3 Fields

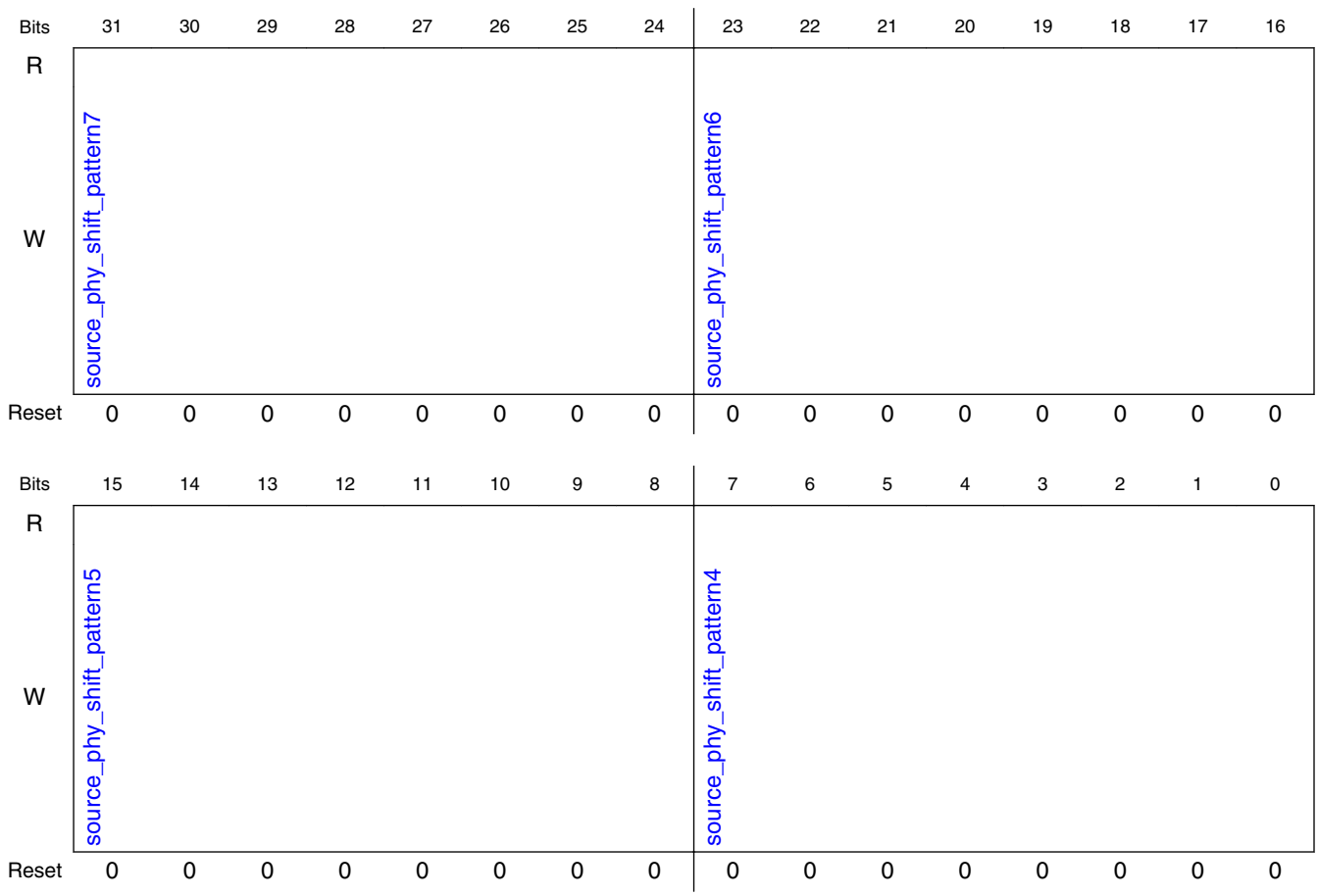
Field	Function
31-24 source_phy_shift_pattern3	Input to hdmi_pattern_shift
23-16 source_phy_shift_pattern2	Input to hdmi_pattern_shift
15-8 source_phy_shift_pattern1	Input to hdmi_pattern_shift
7-0 source_phy_shift_pattern0	Input to hdmi_pattern_shift

13.4.10.1.28 HDMI shift pattern 4-7 (SHIFT_PATTERN_IN_4_7)

13.4.10.1.28.1 Offset

Register	Offset
SHIFT_PATTERN_IN_4_7	804h

13.4.10.1.28.2 Diagram



13.4.10.1.28.3 Fields

Field	Function
31-24 source_phy_shift_pattern7	Input to hdmi_pattern_shift

Table continues on the next page...

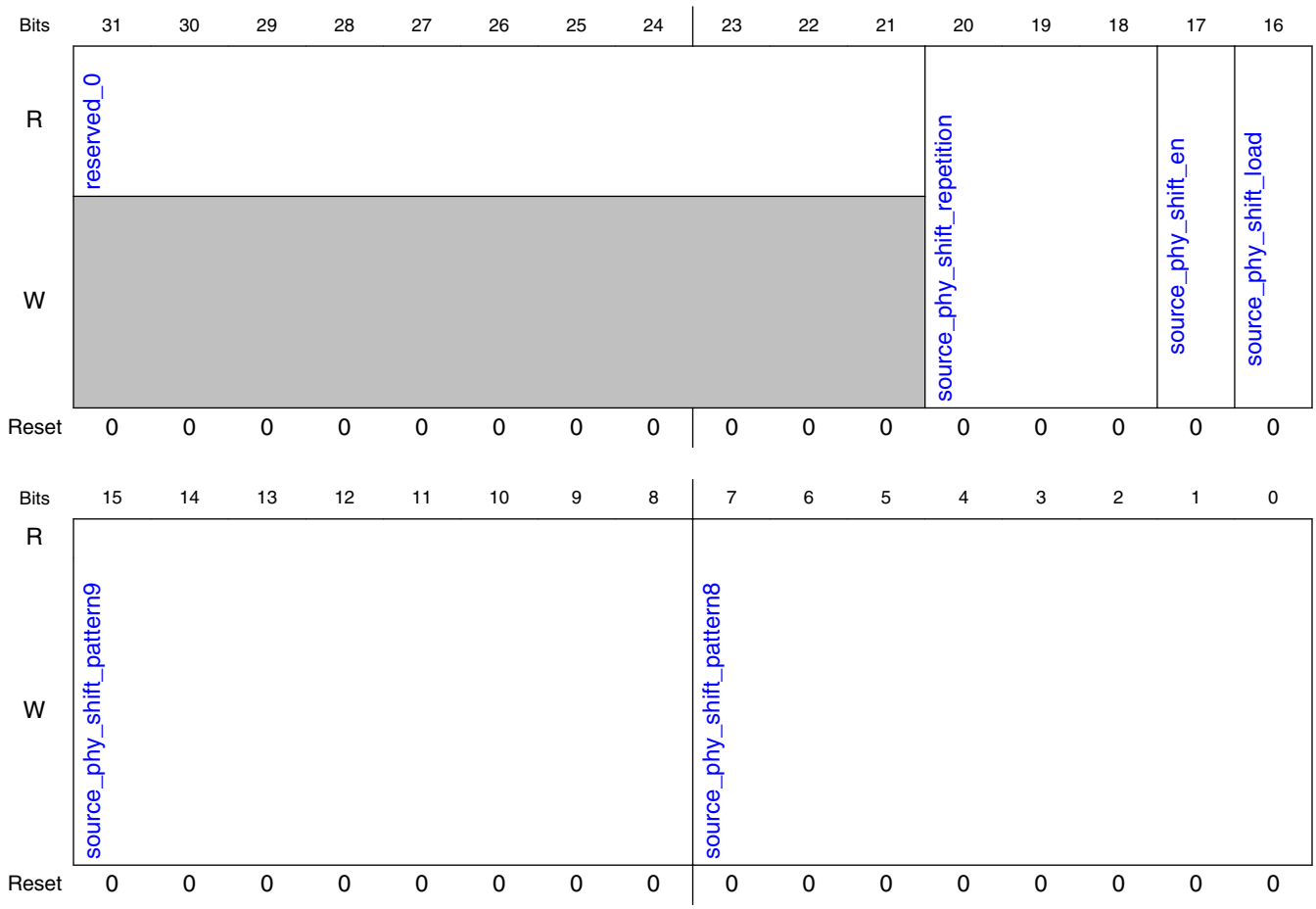
Field	Function
23-16 source_phy_shift_pattern6	Input to hdmi_pattern_shift
15-8 source_phy_shift_pattern5	Input to hdmi_pattern_shift
7-0 source_phy_shift_pattern4	Input to hdmi_pattern_shift

13.4.10.1.29 HDMI shift pattern 9-8 with control bits (SHIFT_PATTERN_IN9_8)

13.4.10.1.29.1 Offset

Register	Offset
SHIFT_PATTERN_IN9_8	808h

13.4.10.1.29.2 Diagram



13.4.10.1.29.3 Fields

Field	Function
31-21 reserved_0	
20-18 source_phy_shift_repetition	Shift repetition Number
17 source_phy_shift_en	When 1 enable the Shift pattern Mechanism
16 source_phy_shift_load	When 1 load the 80 bits of data
15-8	Input to hdmi_pattern_shift

Table continues on the next page...

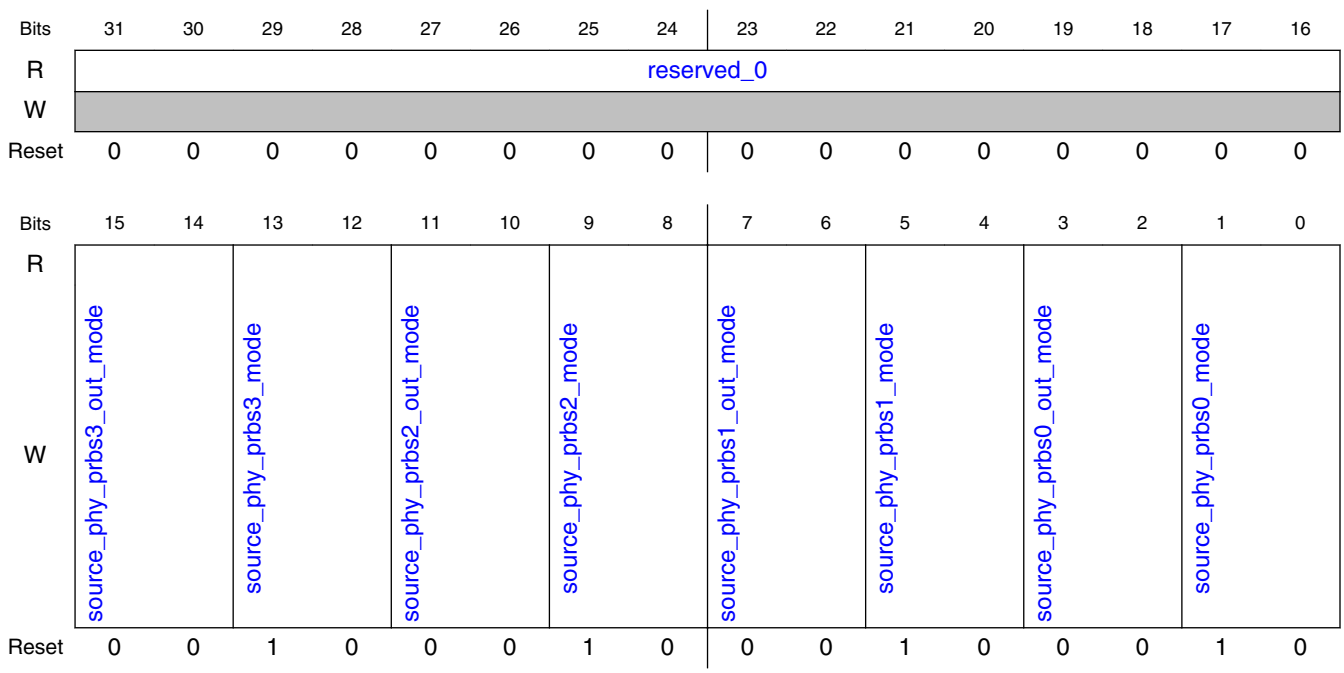
Field	Function
source_phy_shift_pattern9	
7-0 source_phy_shift_pattern8	Input to hdmi_pattern_shift

13.4.10.1.30 PRBS control (PRBS_CNTRL)

13.4.10.1.30.1 Offset

Register	Offset
PRBS_CNTRL	80Ch

13.4.10.1.30.2 Diagram



13.4.10.1.30.3 Fields

Field	Function
31-16 reserved_0	

Table continues on the next page...

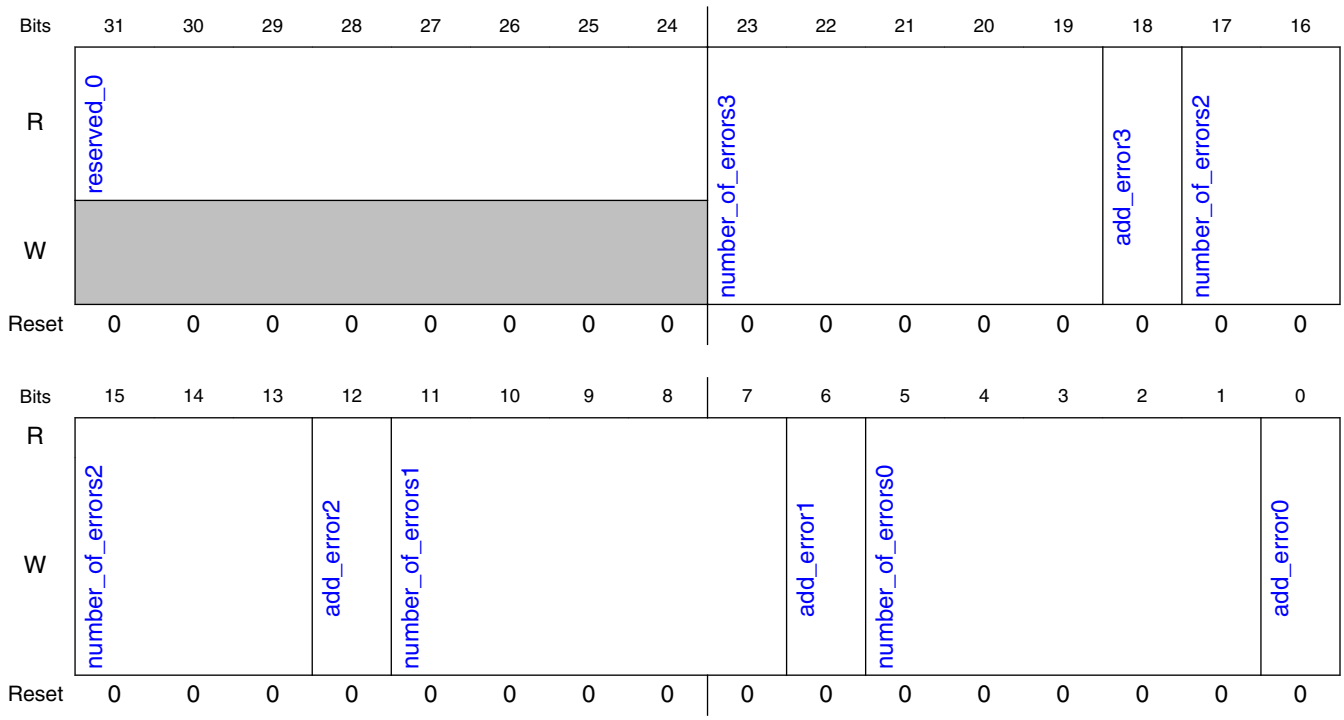
Field	Function
15-14 source_phy_prbs3_out_mode	00 = idle, output all zeros -01 = output 8 bits on pattern[7:0]10 = output 1 bit on pattern[9] and output inverted bit on pattern[8] -11 = output 10 bits on pattern[9:0]
13-12 source_phy_prbs3_mode	00 = PRBS11 -01 = PRBS15 -10 = PRBS7 -11 = PRBS31
11-10 source_phy_prbs2_out_mode	00 = idle, output all zeros01 = output 8 bits on pattern[7:0] -10 = output 1 bit on pattern[9] and output inverted bit on pattern[8] -11 = output 10 bits on pattern[9:0]
9-8 source_phy_prbs2_mode	00 = PRBS11 -01 = PRBS15 -10 = PRBS7 -11 = PRBS31
7-6 source_phy_prbs1_out_mode	00 = idle, output all zeros -01 = output 8 bits on pattern[7:0] -10 = output 1 bit on pattern[9] and output inverted bit on pattern[8] -11 = output 10 bits on pattern[9:0]
5-4 source_phy_prbs1_mode	00 = PRBS11 -01 = PRBS15 -10 = PRBS7 -11 = PRBS31
3-2 source_phy_prbs0_out_mode	00 = idle, output all zeros01 = output 8 bits on pattern[7:0]10 = output 1 bit on pattern[9] and output inverted bit on pattern[8]11 = output 10 bits on pattern[9:0]
1-0 source_phy_prbs0_mode	00 = PRBS1101 = PRBS15 -10 = PRBS7 -11 = PRBS31

13.4.10.1.31 PRBS error insertion (PRBS_ERR_INSERTION)

13.4.10.1.31.1 Offset

Register	Offset
PRBS_ERR_INSERTION	810h

13.4.10.1.31.2 Diagram



13.4.10.1.31.3 Fields

Field	Function
31-24 reserved_0	
23-19 number_of_errors3	The number of errors to be inserted when add_error is high. The number of errors to be inserted when add_error is high.
18 add_error3	When high the PRBS generator inserts the number of errors written in number_of_errors field. When high the PRBS generator inserts the number of errors written in number_of_errors field.
17-13 number_of_errors2	The number of errors to be inserted when add_error is high. The number of errors to be inserted when add_error is high.
12 add_error2	When high the PRBS generator inserts the number of errors written in number_of_errors field. When high the PRBS generator inserts the number of errors written in number_of_errors field.
11-7 number_of_errors1	The number of errors to be inserted when add_error is high. The number of errors to be inserted when add_error is high.
6 add_error1	When high the PRBS generator inserts the number of errors written in number_of_errors field
5-1	The number of errors to be inserted when add_error is high.

Table continues on the next page...

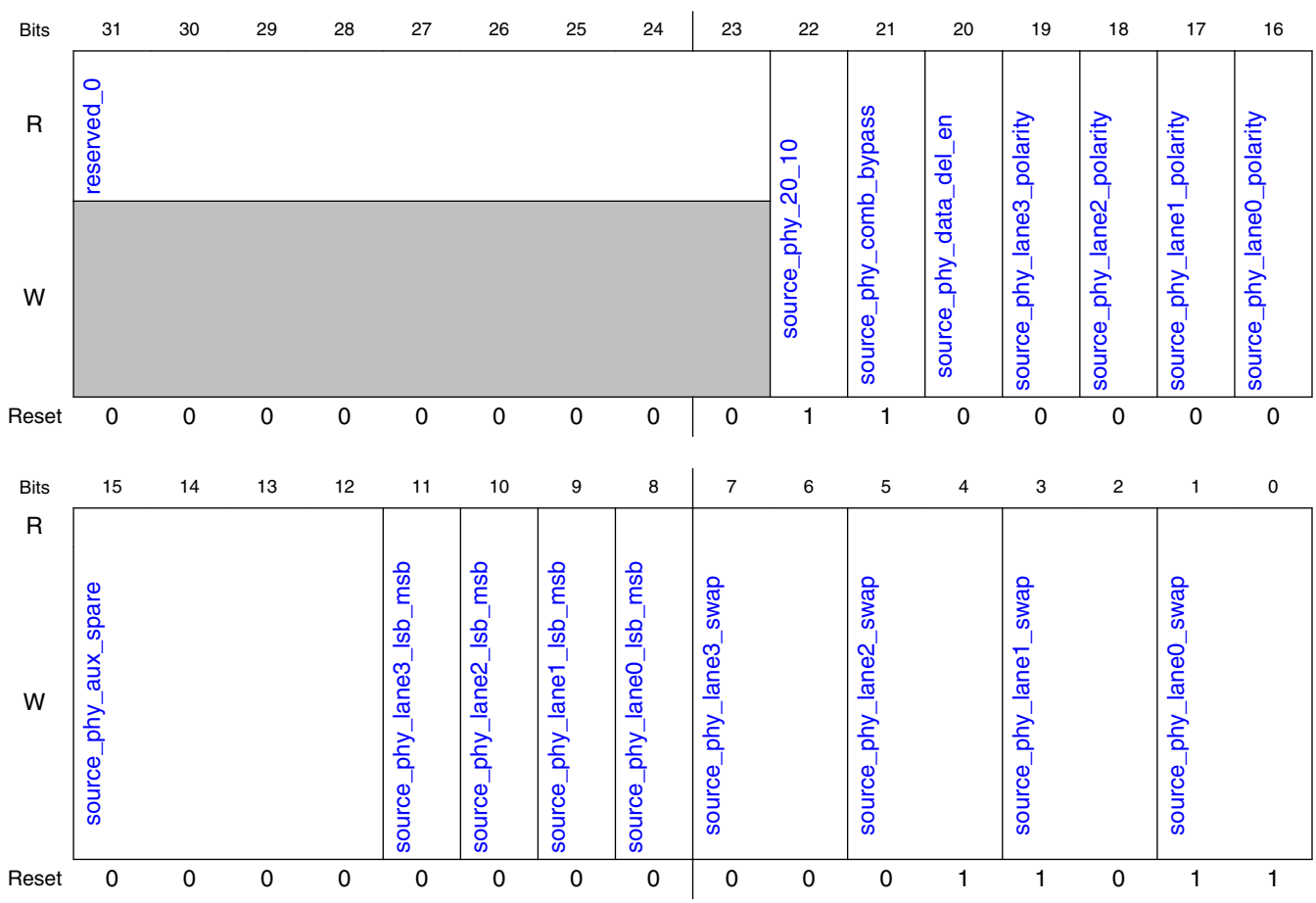
Field	Function
number_of_errors0	The number of errors to be inserted when add_error is high.
0 add_error0	When high the PRBS generator inserts the number of errors written in number_of_errors field

13.4.10.1.32 Lane control register: swap, order, polarity (LANES_CONFIG)

13.4.10.1.32.1 Offset

Register	Offset
LANES_CONFIG	814h

13.4.10.1.32.2 Diagram



13.4.10.1.32.3 Fields

Field	Function
31-23 reserved_0	
22 source_phy_20_10	1'b0: Data to PHY is 10 bit with char clock -1'd1: Data to PHY is 20 bit with data clock
21 source_phy_comb_bypass	Bypass swap invert and all combination
20 source_phy_data_del_en	enable configurable delay of lanes to be activated. enable configurable delay of lanes to be activated.if this bit is 0 the delay is only activated for DisplayPort mode with source_phy_data_sel=prbs or shift-mem
19 source_phy_lane3_polarity	Reverse polarity of data, lane3
18 source_phy_lane2_polarity	Reverse polarity of data, lane2
17 source_phy_lane1_polarity	Reverse polarity of data, lane1
16 source_phy_lane0_polarity	Reverse polarity of data, lane0
15-12 source_phy_aux_spare	Spare bits for aux **1. Spare bits for aux **1.1**
11 source_phy_lane3_lsb_msb	Reverse order of data, lane3
10 source_phy_lane2_lsb_msb	Reverse order of data, lane2
9 source_phy_lane1_lsb_msb	Reverse order of data, lane1
8 source_phy_lane0_lsb_msb	Reverse order of data, lane0
7-6 source_phy_lane3_swap	Swap control lane3
5-4	Swap control lane2

Table continues on the next page...

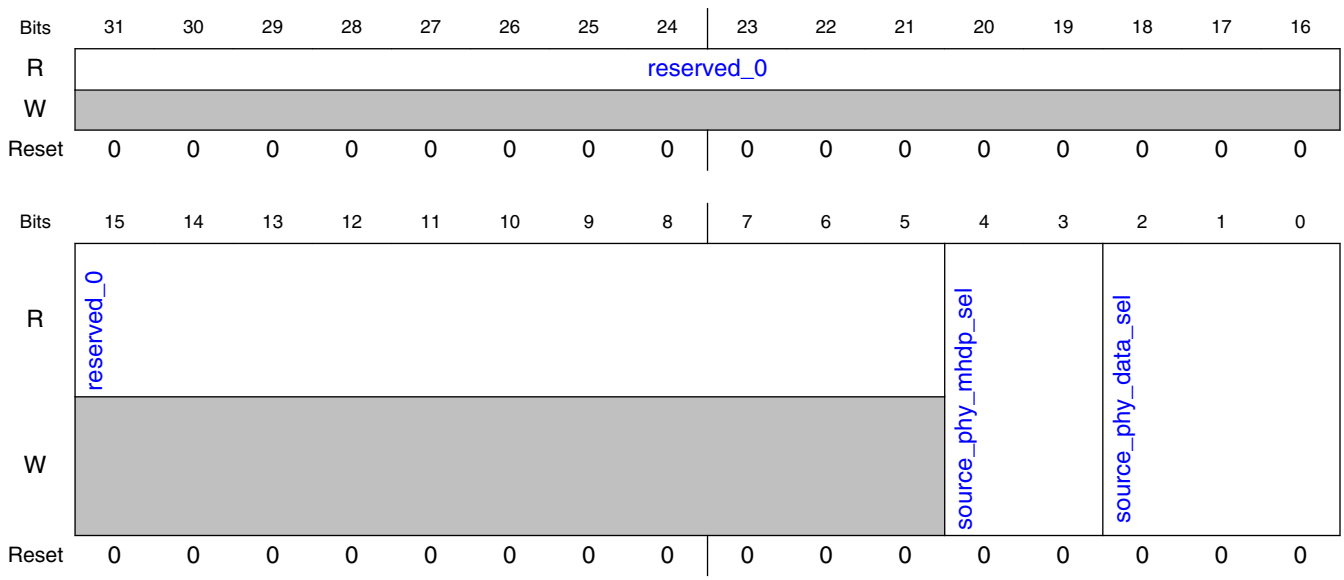
Field	Function
source_phy_lane2_swap	
3-2 source_phy_lane1_swap	Swap control lane1
1-0 source_phy_lane0_swap	Swap control lane0

13.4.10.1.33 PHY data select DP/HDMI and HDMI data source (PHY_DATA_SEL)

13.4.10.1.33.1 Offset

Register	Offset
PHY_DATA_SEL	818h

13.4.10.1.33.2 Diagram



13.4.10.1.33.3 Fields

Field	Function
31-5	

Table continues on the next page...

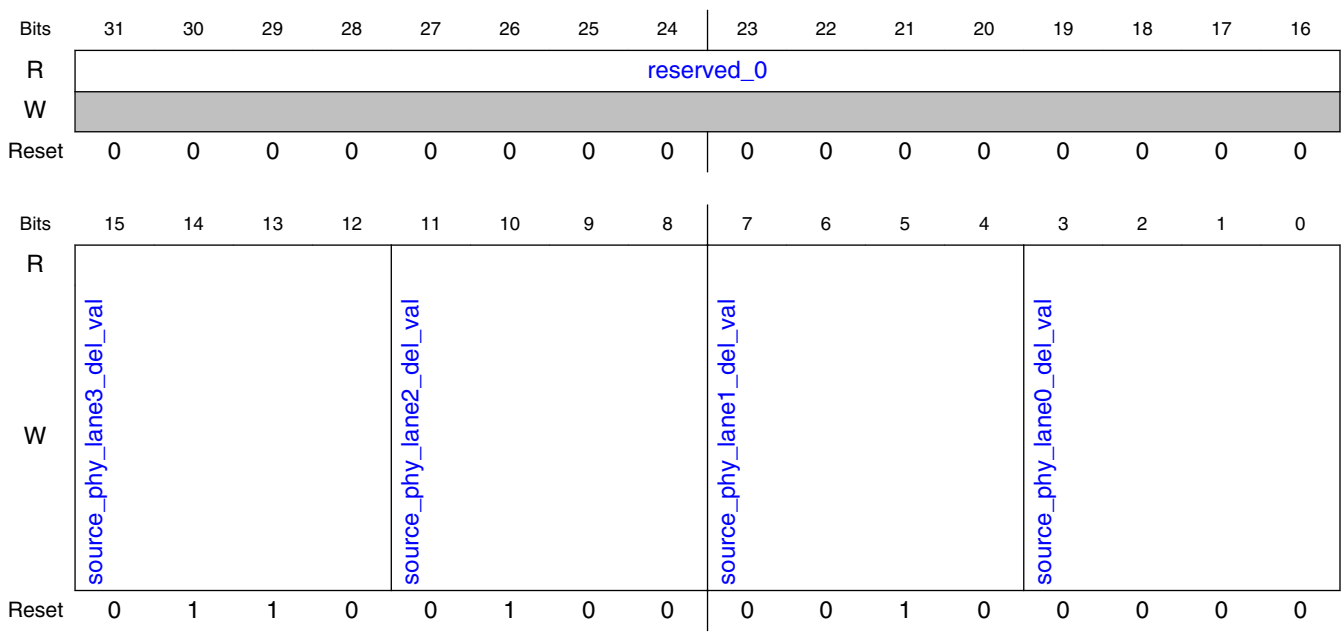
Field	Function
reserved_0	
4-3 source_phy_mhdp_sel	3'd0: tx_data = DP -3'd1: tx_data = HDMI -3'd2: tx_data = RSRV -3'd3: tx_data = RSRV
2-0 source_phy_data_sel	3'd0: tx_data = phy_dout -3'd1: tx_data = phy_dout_bypass -3'd2: tx_data = source_phy_prbs_pout -3'd3: tx_data = source_phy_shift_pout

13.4.10.1.34 Lane delay control (LANES_DEL_VAL)

13.4.10.1.34.1 Offset

Register	Offset
LANES_DEL_VAL	81Ch

13.4.10.1.34.2 Diagram



13.4.10.1.34.3 Fields

Field	Function
31-16	

Table continues on the next page...

Field	Function
reserved_0	
15-12 source_phy_lane3_del_val	delay for lane 3 -this parameter can take values from 0 up to 8. delay for lane 3 -this parameter can take values from 0 up to 8. -All other values are reserved
11-8 source_phy_lane2_del_val	delay for lane 2 -this parameter can take values from 0 up to 8. delay for lane 2 -this parameter can take values from 0 up to 8. -All other values are reserved
7-4 source_phy_lane1_del_val	delay for lane 1 -this parameter can take values from 0 up to 8. delay for lane 1 -this parameter can take values from 0 up to 8. -All other values are reserved
3-0 source_phy_lane0_del_val	delay for lane 0 -this parameter can take values from 0 up to 8. delay for lane 0 -this parameter can take values from 0 up to 8. -All other values are reserved

13.4.10.1.35 Register implemented only for configuration with HDMI. (SOURCE_HDTX_CAR)

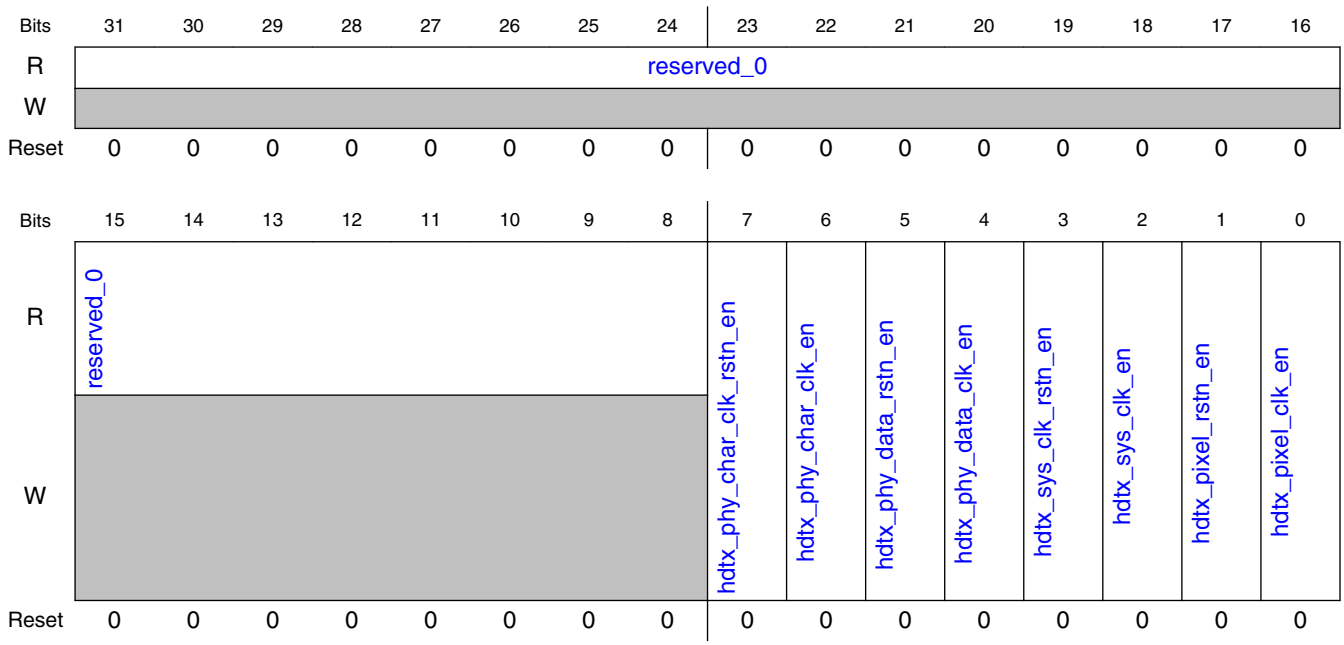
13.4.10.1.35.1 Offset

Register	Offset
SOURCE_HDTX_CAR	900h

13.4.10.1.35.2 Function

Register implemented only for configuration with HDMI. Each bit is responsible for clock or reset enable

13.4.10.1.35.3 Diagram



13.4.10.1.35.4 Fields

Field	Function
31-8 reserved_0	Reserved - ignored on write
7 hdtx_phy_char_clk_rstn_en	hdtx_phy_char_clk_rstn enable - active low (only when HDMI used)
6 hdtx_phy_char_clk_en	hdtx_phy_char_clk enable - active high (only when HDMI used)
5 hdtx_phy_data_rstn_en	hdtx_phy_data_rstn enable - active low (only when HDMI used)
4 hdtx_phy_data_clk_en	hdtx_phy_data_clk enable - active high (only when HDMI used)
3 hdtx_sys_clk_rstn_en	hdtx_sys_clk_rstn enable - active low (only when HDMI used)
2 hdtx_sys_clk_en	hdtx_sys_clk enable - active high (only when HDMI used)
1	hdtx_pixel_rstn enable - active low (only when HDMI used)

Table continues on the next page...

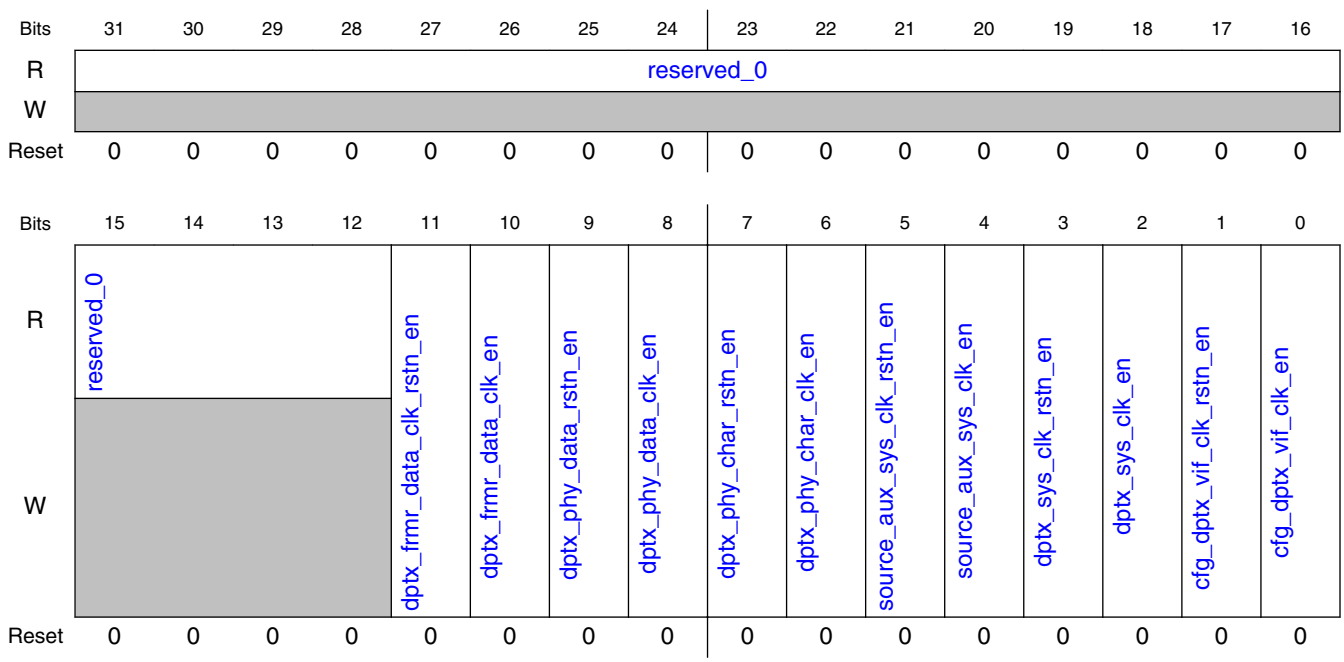
Field	Function
hdtx_pixel_rstn_en	
0 hdtx_pixel_clk_en	hdtx_pixel_clk enable - active high (only when HDMI used)

13.4.10.1.36 DP TX clock and reset ctrl register (SOURCE_DPTX_CAR)

13.4.10.1.36.1 Offset

Register	Offset
SOURCE_DPTX_CAR	904h

13.4.10.1.36.2 Diagram



13.4.10.1.36.3 Fields

Field	Function
31-12 reserved_0	Reserved - ignored on write

Table continues on the next page...

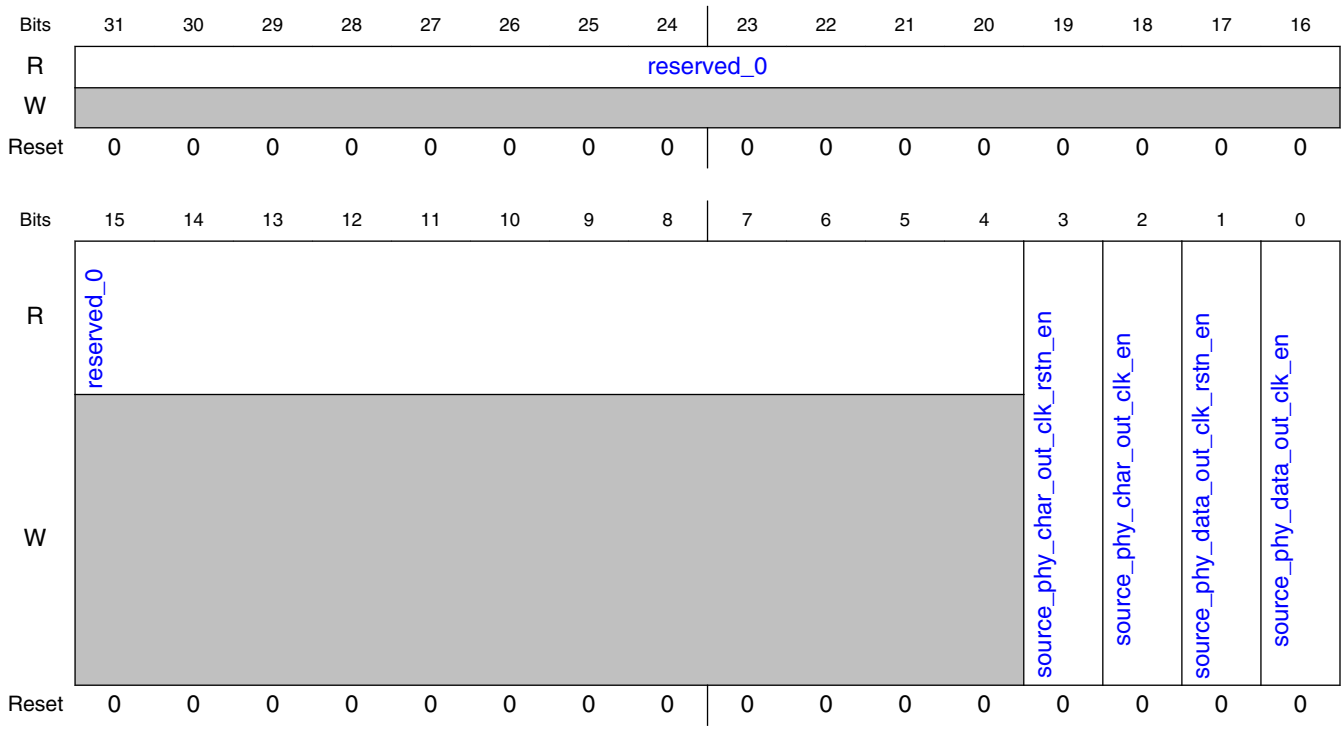
Field	Function
11 dptx_frmr_data_clk_rstn_en	dptx_frmr_data_clk_rstn enable - active low
10 dptx_frmr_data_clk_en	dptx_frmr_data_clk enable - active high
9 dptx_phy_data_rstn_en	dptx_phy_data_rstn enable - active low
8 dptx_phy_data_clk_en	dptx_phy_data_clk enable - active high
7 dptx_phy_char_rstn_en	dptx_phy_char_rstn enable - active low
6 dptx_phy_char_clk_en	dptx_phy_char_clk enable - active high
5 source_aux_sys_clk_rstn_en	source_aux_sys_clk_rstn enable - active low
4 source_aux_sys_clk_en	source_aux_sys_clk enable - active high
3 dptx_sys_clk_rstn_en	dptx_sys_clk_rstn enable - active low
2 dptx_sys_clk_en	dptx_sys_clk enable - active high
1 cfg_dptx_vif_clk_rstn_en	dptx_vif_clk_rstn enable - active low
0 cfg_dptx_vif_clk_en	dptx_vif_clk enable - active high

13.4.10.1.37 Source PHY clock and reset ctrl register (SOURCE_PHY_CAR)

13.4.10.1.37.1 Offset

Register	Offset
SOURCE_PHY_CAR	908h

13.4.10.1.37.2 Diagram



13.4.10.1.37.3 Fields

Field	Function
31-4 reserved_0	Reserved - ignored on write
3 source_phy_char_out_clk_rstn_en	source_phy_char_out_clk_rstn enable - active low
2 source_phy_char_out_clk_en	source_phy_char_out_clk enable - active high
1	source_phy_data_out_clk_rstn enable - active low

Table continues on the next page...

Field	Function
source_phy_data_out_clk_rstn_en	
0 source_phy_data_out_clk_en	source_phy_data_out_clk enable - active high

13.4.10.1.38 Register implemented only for configuration with HDMI. (SOURCE_CEC_CAR)

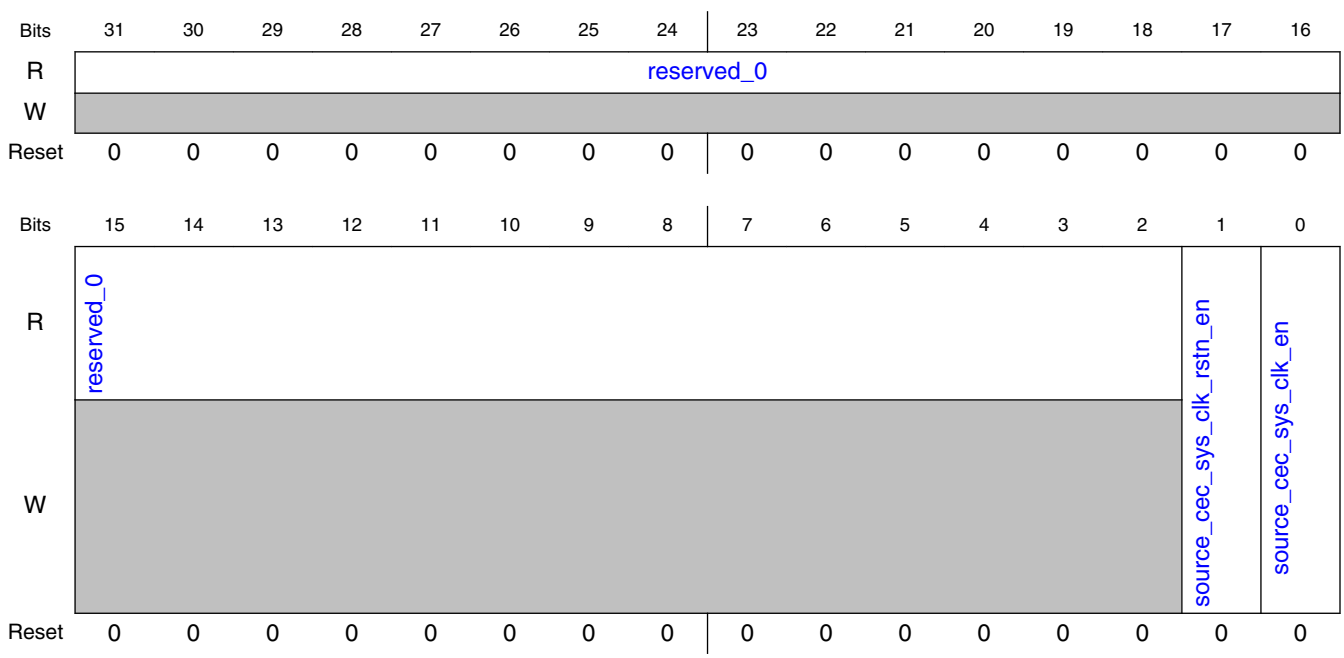
13.4.10.1.38.1 Offset

Register	Offset
SOURCE_CEC_CAR	90Ch

13.4.10.1.38.2 Function

Register implemented only for configuration with HDMI. Each bit is responsible for clock or reset enable

13.4.10.1.38.3 Diagram



13.4.10.1.38.4 Fields

Field	Function
31-2 reserved_0	Reserved - ignored on write (only when HDMI used)
1 source_cec_sys_clk_rstn_en	source_cec_sys_clk_rstn enable - active low (only when HDMI used)
0 source_cec_sys_clk_en	source_cec_sys_clk enable - active high (only when HDMI used)

13.4.10.1.39 CBUS clock and reset ctrl register. (SOURCE_CBUS_CAR)

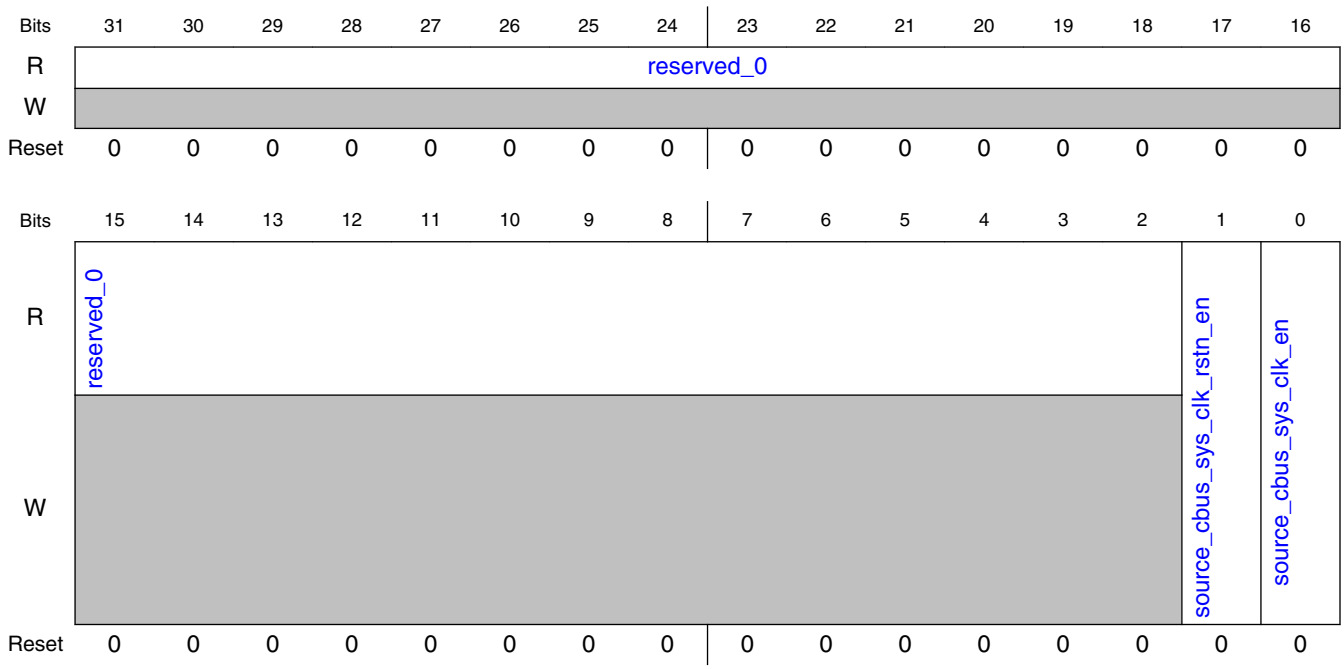
13.4.10.1.39.1 Offset

Register	Offset
SOURCE_CBUS_CAR	910h

13.4.10.1.39.2 Function

CBUS clock and reset ctrl register. Implemented only if CAPB is implemented

13.4.10.1.39.3 Diagram



13.4.10.1.39.4 Fields

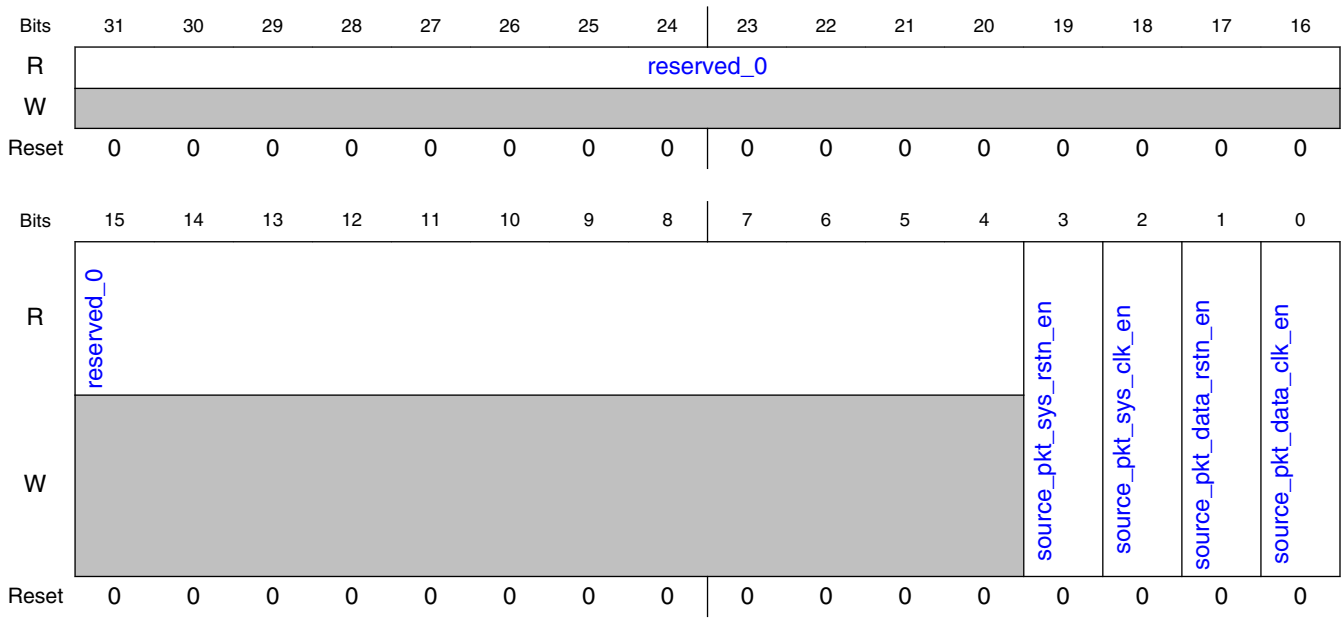
Field	Function
31-2 reserved_0	Reserved - ignored on write
1 source_cbus_sys_clk_rstn_en	source_cbus_sys_clk_rstn enable - active low
0 source_cbus_sys_clk_en	source_cbus_sys_clk enable - active high

13.4.10.1.40 PKT clock and reset ctrl register (SOURCE_PKT_CAR)

13.4.10.1.40.1 Offset

Register	Offset
SOURCE_PKT_CAR	918h

13.4.10.1.40.2 Diagram



13.4.10.1.40.3 Fields

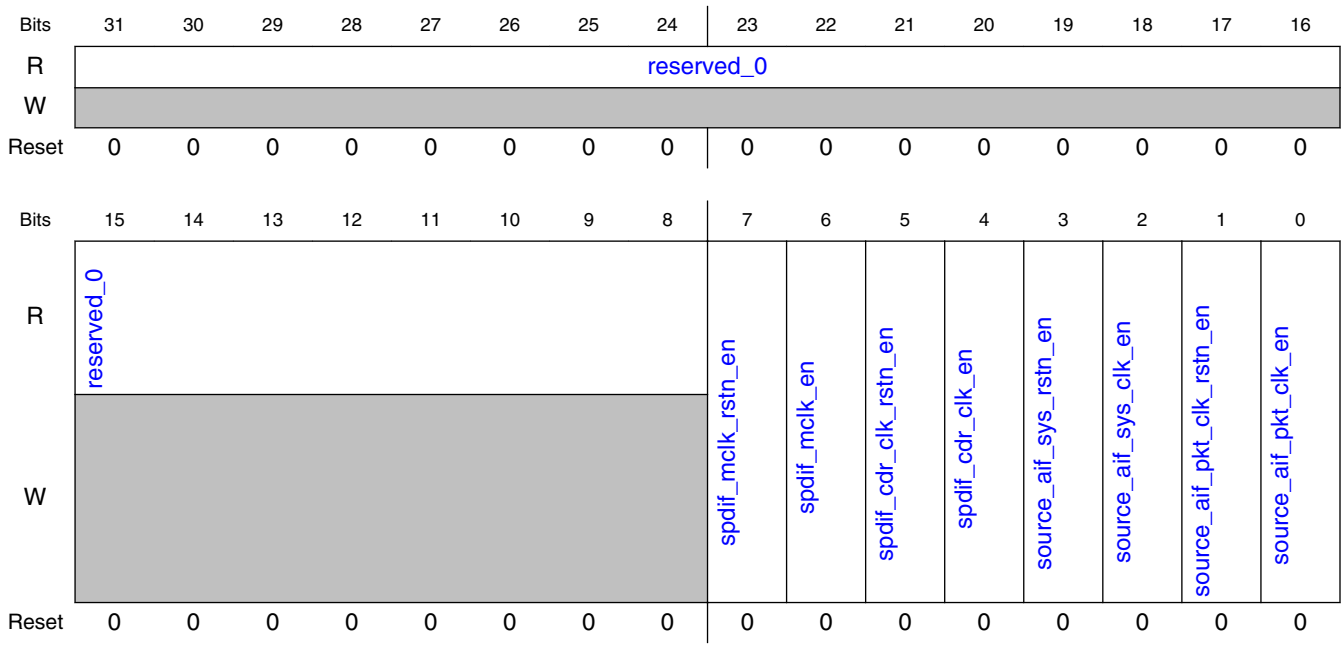
Field	Function
31-4 reserved_0	Reserved - ignored on write
3 source_pkt_sys_rstn_en	source_pkt_sys_rstn enable - active low
2 source_pkt_sys_clk_en	source_pkt_sys_clk enable - active high
1 source_pkt_data_rstn_en	source_pkt_data_rstn enable - active low
0 source_pkt_data_clk_en	source_pkt_data_clk enable - active high

13.4.10.1.41 AIF clock and reset ctrl register (SOURCE_AIF_CAR)

13.4.10.1.41.1 Offset

Register	Offset
SOURCE_AIF_CAR	91Ch

13.4.10.1.41.2 Diagram



13.4.10.1.41.3 Fields

Field	Function
31-8 reserved_0	Reserved - ignored on write
7 spdif_mclk_rstn_en	spdif_mclk_rstn enable - active low
6 spdif_mclk_en	spdif_mclk enable - active high
5 spdif_cdr_clk_rstn_en	spdif_cdr_clk_rstn enable - active low
4 spdif_cdr_clk_en	spdif_cdr_clk enable - active high

Table continues on the next page...

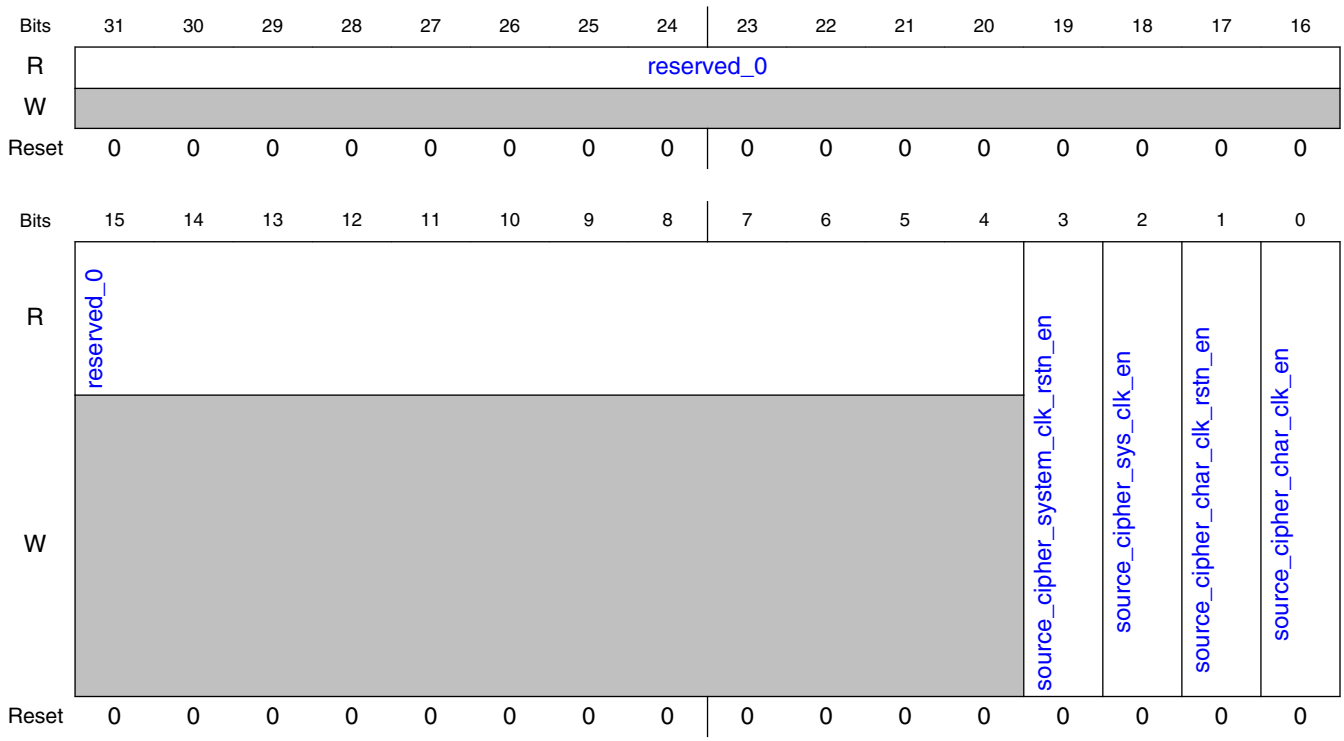
Field	Function
3 source_aif_sys_rstn_en	source_aif_sys_rstn enable - active low
2 source_aif_sys_clk_en	source_aif_sys_clk enable - active high
1 source_aif_pkt_clk_rstn_en	source_aif_pkt_clk_rstn enable - active low
0 source_aif_pkt_clk_en	source_aif_pkt_clk enable - active high

13.4.10.1.42 Cipher clock and reset ctrl register (SOURCE_CIPHER_CAR)

13.4.10.1.42.1 Offset

Register	Offset
SOURCE_CIPHER_CAR	920h

13.4.10.1.42.2 Diagram



13.4.10.1.42.3 Fields

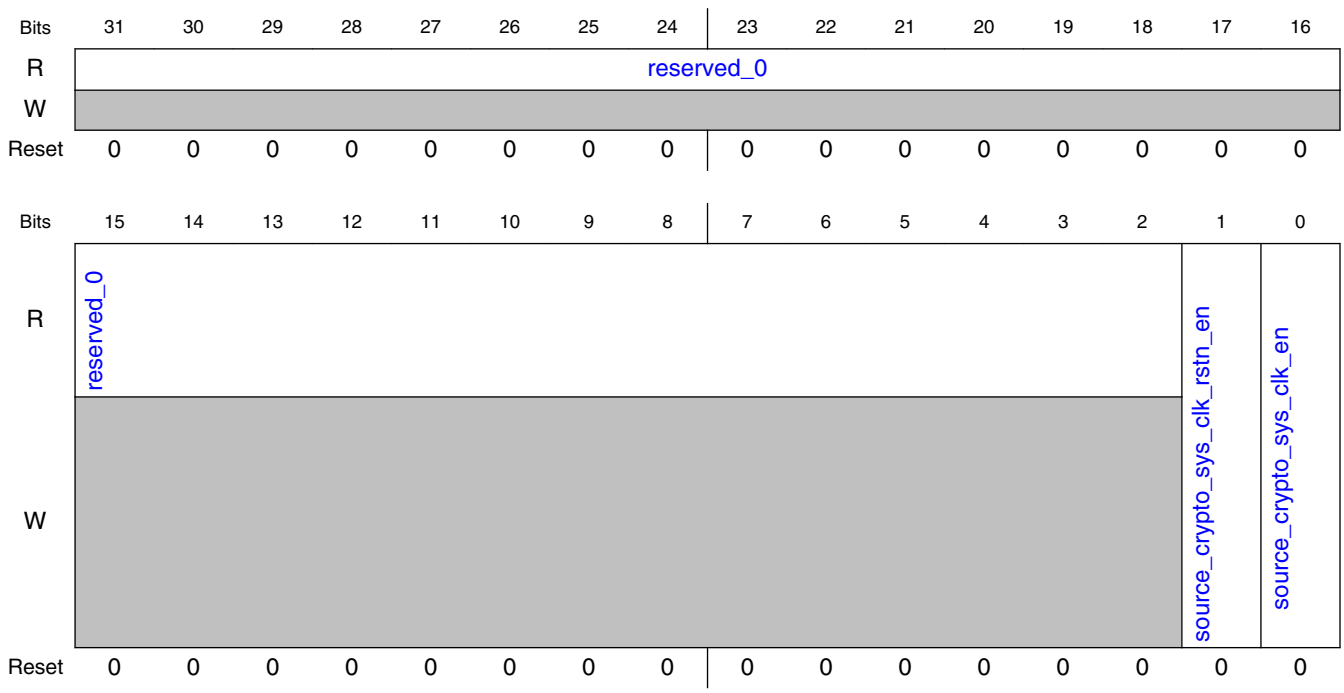
Field	Function
31-4 reserved_0	Reserved - ignored on write
3 source_cipher_system_clk_rstn_en	source_cipher_system_clk_rstn enable - active low (Only when HDCP used)
2 source_cipher_sys_clk_en	source_cipher_sys_clk enable - active high (Only when HDCP used)
1 source_cipher_char_clk_rstn_en	source_cipher_char_clk_rstn enable - active low (Only when HDCP used)
0 source_cipher_char_clk_en	source_cipher_char_clk enable - active high (Only when HDCP used)

13.4.10.1.43 Crypto clock and reset ctrl register (SOURCE_CRYPTO_CAR)

13.4.10.1.43.1 Offset

Register	Offset
SOURCE_CRYPTO_CAR	924h

13.4.10.1.43.2 Diagram



13.4.10.1.43.3 Fields

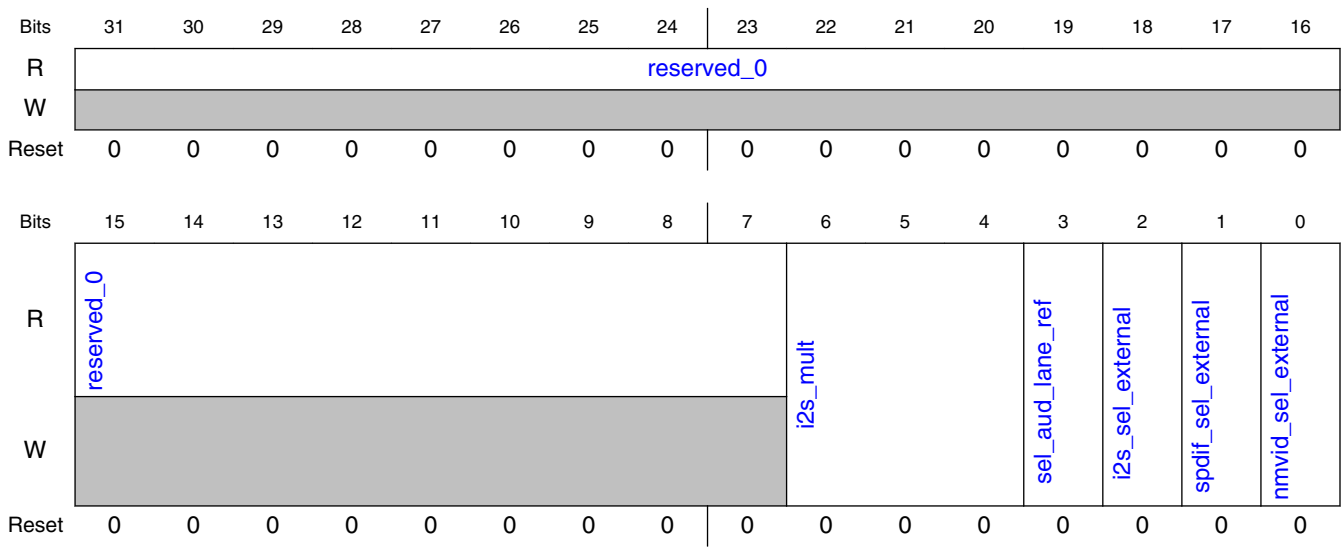
Field	Function
31-2 reserved_0	Reserved - ignored on write
1 source_crypto_sys_clk_rstn_en	source_crypto_sys_clk_rstn enable - active low (Only when HDCP used)
0 source_crypto_sys_clk_en	source_crypto_sys_clk enable - active high (Only when HDCP used)

13.4.10.1.44 Clock Meter control (CM_CTRL)

13.4.10.1.44.1 Offset

Register	Offset
CM_CTRL	A00h

13.4.10.1.44.2 Diagram



13.4.10.1.44.3 Fields

Field	Function
31-7 reserved_0	
6-4 i2s_mult	Select the division of N value for different I2S TDM configuration
3 sel_aud_lane_ref	When 1 Select Audio CLK as a refrence (HDMI) -When 0 Select LANE CLK as a refrence (DP) -
2 i2s_sel_external	When 1 Select external values of NMAUD (N/A) for I2S
1 spdif_sel_external	When 1 Select external values of NMAUD (N/A) for SPDIF

Table continues on the next page...

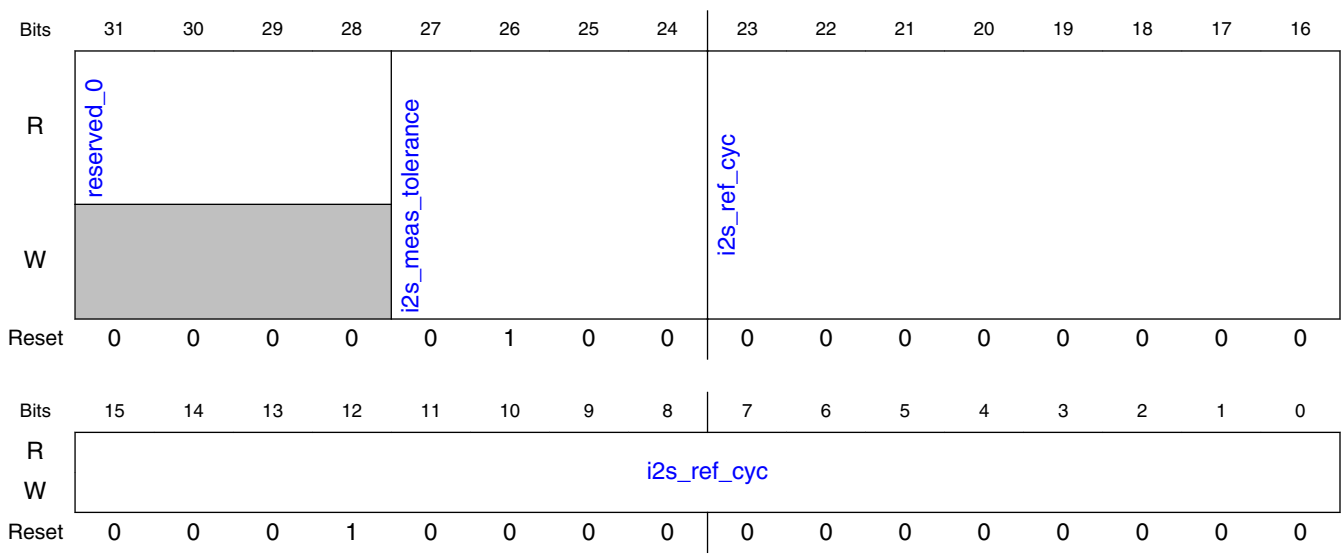
Field	Function
0 nmvid_sel_external	When 1 Select external values of NMVID (N/A)

13.4.10.1.45 I2S clock control (CM_I2S_CTRL)

13.4.10.1.45.1 Offset

Register	Offset
CM_I2S_CTRL	A04h

13.4.10.1.45.2 Diagram



13.4.10.1.45.3 Fields

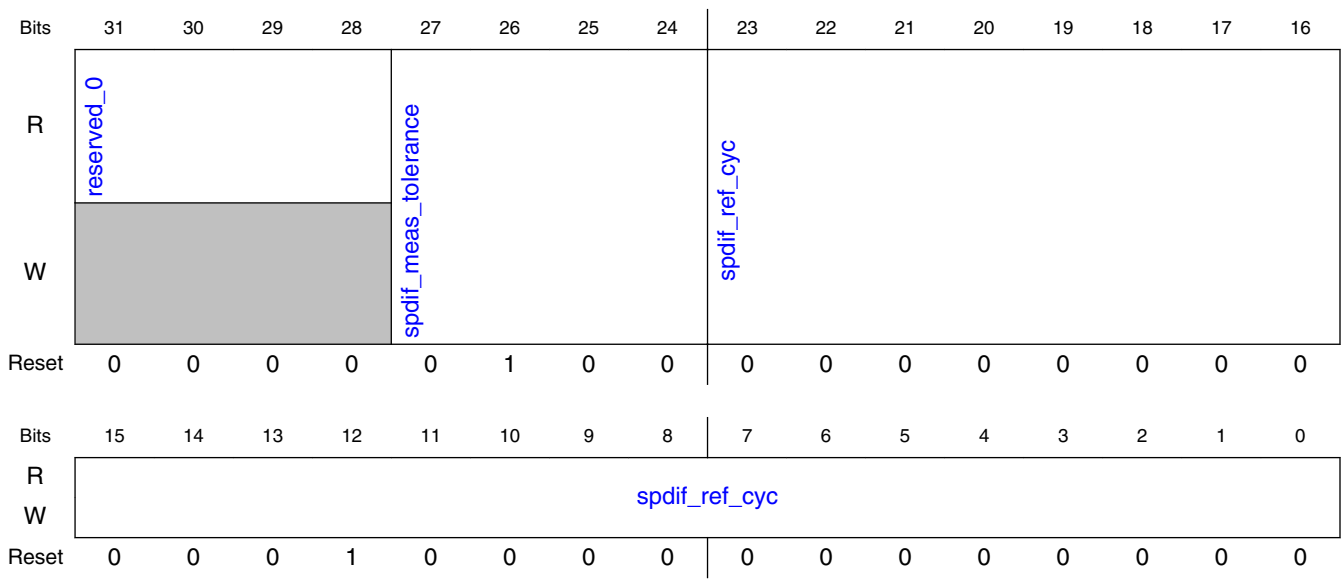
Field	Function
31-28 reserved_0	
27-24 i2s_meas_tolerance	Measurement tolerance of Audio clock to be stable in clocks
23-0 i2s_ref_cyc	Refernce cycles for I2S Audio meter

13.4.10.1.46 SPDIF clock control (CM_SPDIF_CTRL)

13.4.10.1.46.1 Offset

Register	Offset
CM_SPDIF_CTRL	A08h

13.4.10.1.46.2 Diagram



13.4.10.1.46.3 Fields

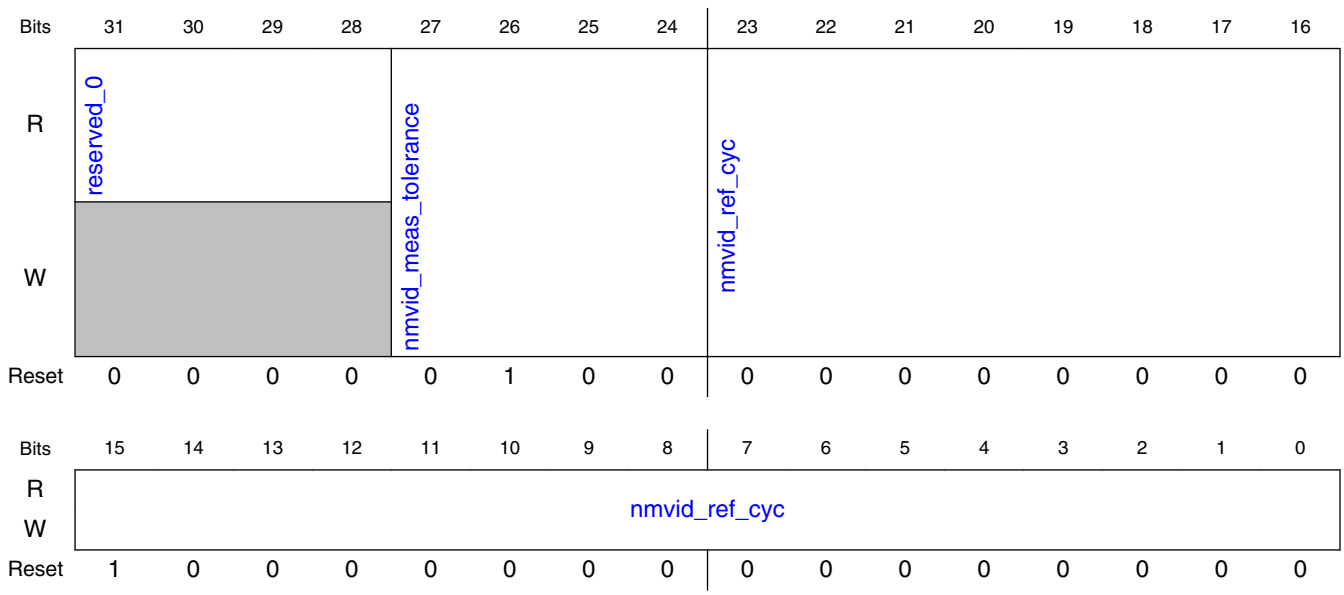
Field	Function
31-28 reserved_0	
27-24 spdif_meas_tolerance	SPDIF measurement tolearnce to be stable in clocks
23-0 spdif_ref_cyc	Refernce cycles of SPDIF measurment

13.4.10.1.47 Video clock control (CM_VID_CTRL)

13.4.10.1.47.1 Offset

Register	Offset
CM_VID_CTRL	A0Ch

13.4.10.1.47.2 Diagram



13.4.10.1.47.3 Fields

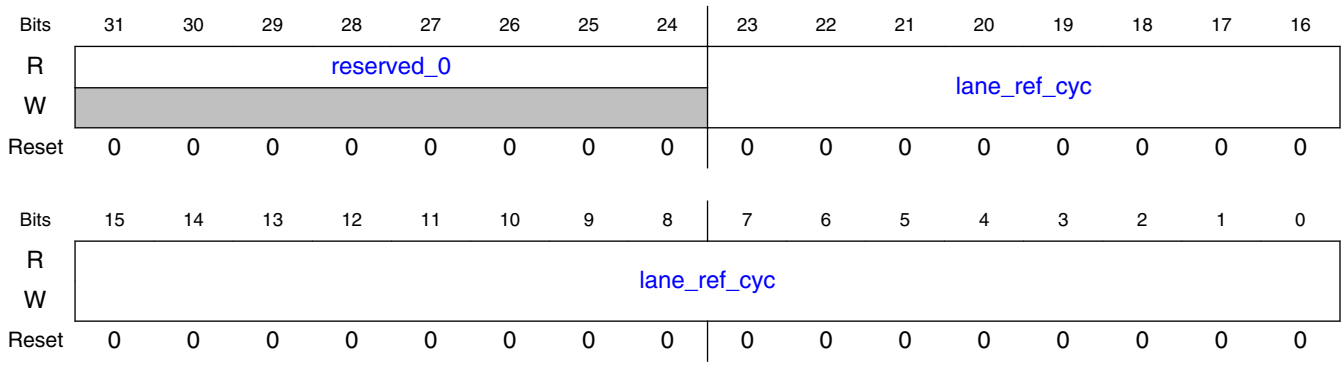
Field	Function
31-28 reserved_0	
27-24 nmvid_meas_tolerance	Video measurment tolerance in pixel clock cycles
23-0 nmvid_ref_cyc	Video refernce cycles

13.4.10.1.48 Lane control (CM_LANE_CTRL)

13.4.10.1.48.1 Offset

Register	Offset
CM_LANE_CTRL	A10h

13.4.10.1.48.2 Diagram



13.4.10.1.48.3 Fields

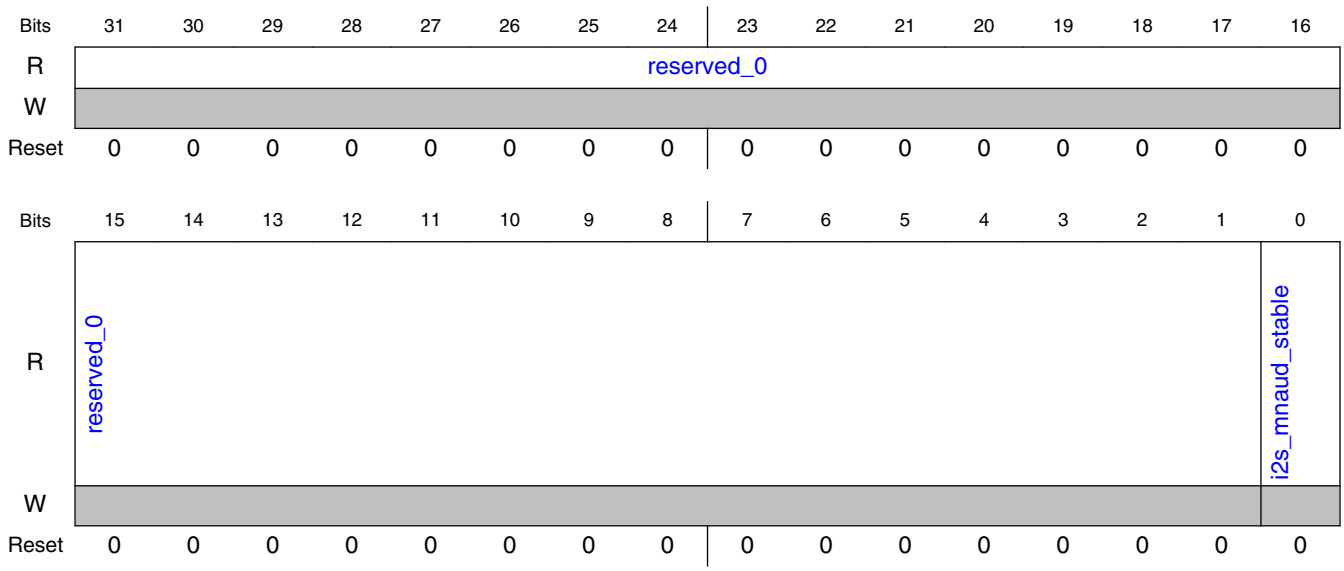
Field	Function
31-24 reserved_0	
23-0 lane_ref_cyc	Refernce cycles when using lane clock as refernce (DP)

13.4.10.1.49 I2S clock stable, audio clock measured (I2S_NM_STABLE)

13.4.10.1.49.1 Offset

Register	Offset
I2S_NM_STABLE	A14h

13.4.10.1.49.2 Diagram



13.4.10.1.49.3 Fields

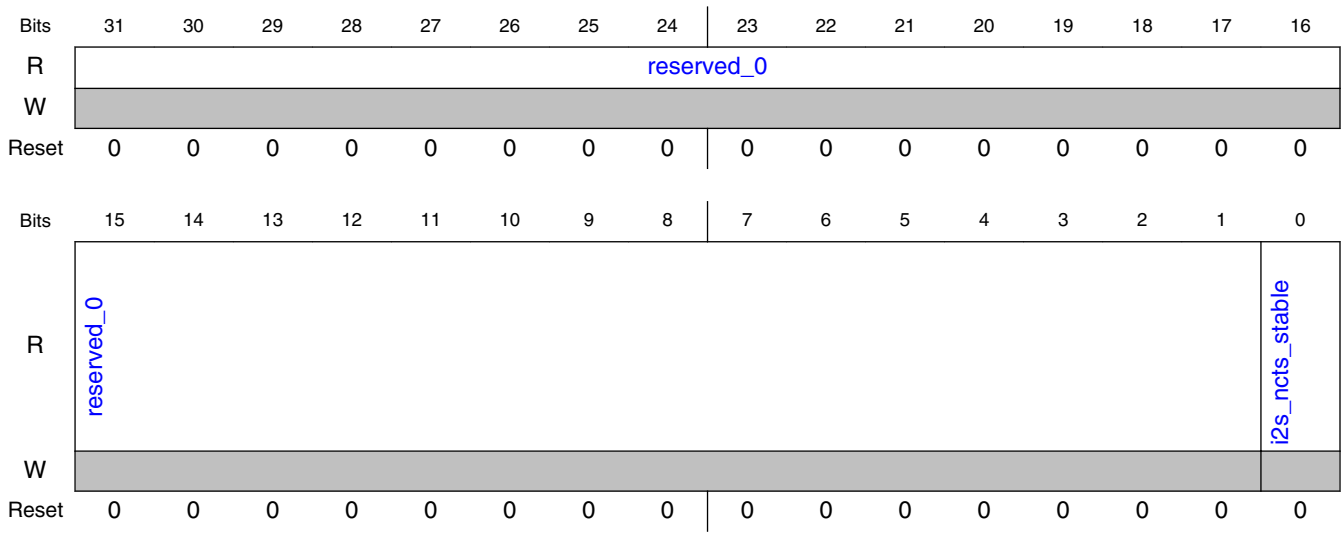
Field	Function
31-1 reserved_0	
0 i2s_mnaud_stable	I2S NMAUD Mesurment stable

13.4.10.1.50 I2S clock stable, lane clock measured (I2S_NCTS_STABLE)

13.4.10.1.50.1 Offset

Register	Offset
I2S_NCTS_STABLE	A18h

13.4.10.1.50.2 Diagram



13.4.10.1.50.3 Fields

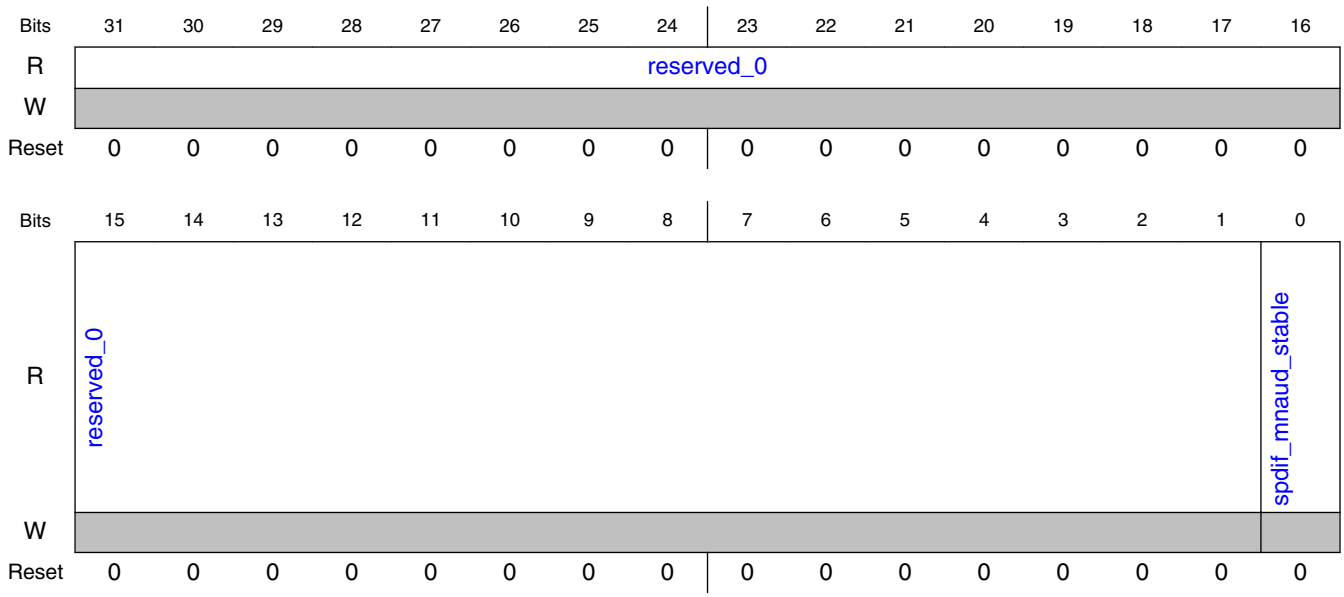
Field	Function
31-1 reserved_0	
0 i2s_ncts_stable	i2s CTS measurment stable

13.4.10.1.51 SPDIF clock stable, audio clock measured (SPDIF_NM_STABLE)

13.4.10.1.51.1 Offset

Register	Offset
SPDIF_NM_STABLE	A1Ch

13.4.10.1.51.2 Diagram



13.4.10.1.51.3 Fields

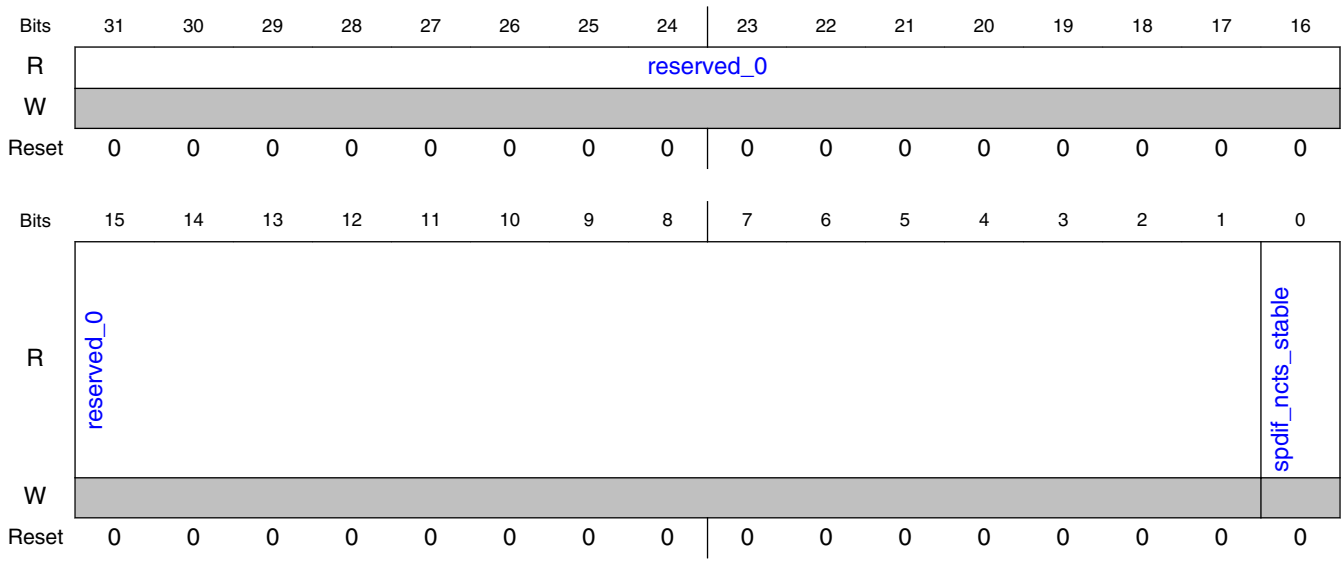
Field	Function
31-1 reserved_0	
0 spdif_mnaud_stable	SPDIF NMAUD measurment stable

13.4.10.1.52 SPDIF clock stable, lane clock measured (SPDIF_NCTS_STABLE)

13.4.10.1.52.1 Offset

Register	Offset
SPDIF_NCTS_STABLE	A20h

13.4.10.1.52.2 Diagram



13.4.10.1.52.3 Fields

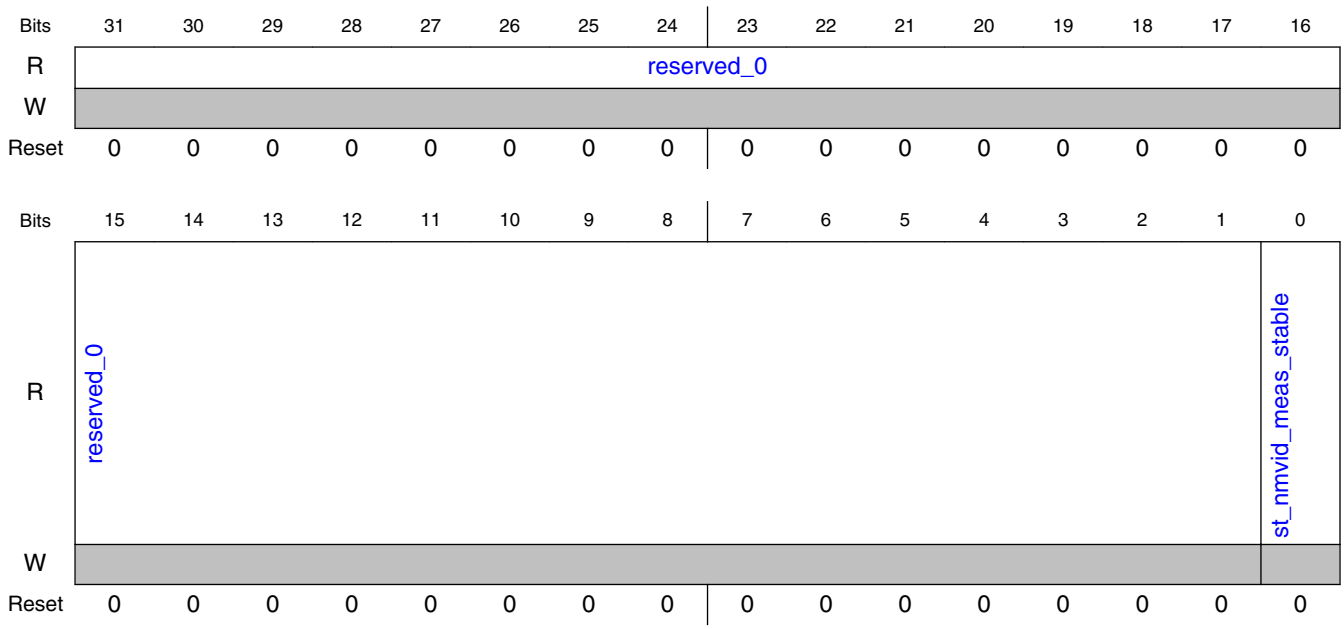
Field	Function
31-1 reserved_0	
0 spdif_ncts_stable	SPDIF CTS measurment stable

13.4.10.1.53 Video clock stable (NMVID_MEAS_STABLE)

13.4.10.1.53.1 Offset

Register	Offset
NMVID_MEAS_STABLE	A24h

13.4.10.1.53.2 Diagram



13.4.10.1.53.3 Fields

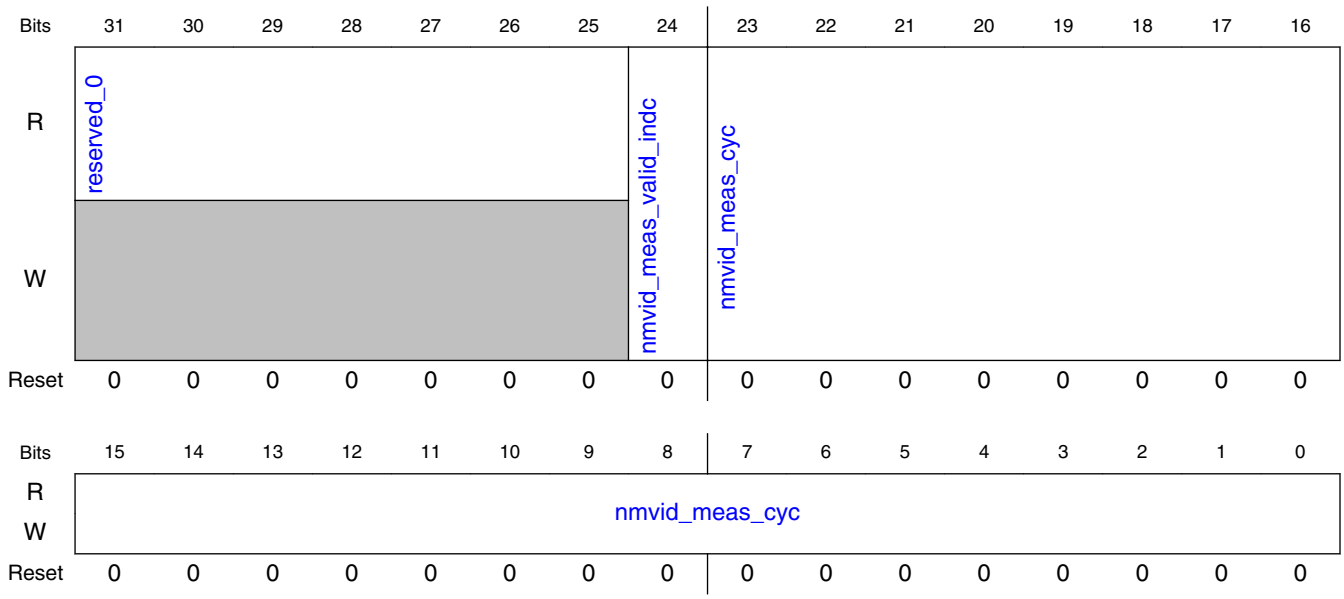
Field	Function
31-1 reserved_0	
0 st_nmvid_meas_stable	Pixel clock NMVID measurment stable

13.4.10.1.54 Video cycles measure (CM_VID_MEAS)

13.4.10.1.54.1 Offset

Register	Offset
CM_VID_MEAS	A28h

13.4.10.1.54.2 Diagram



13.4.10.1.54.3 Fields

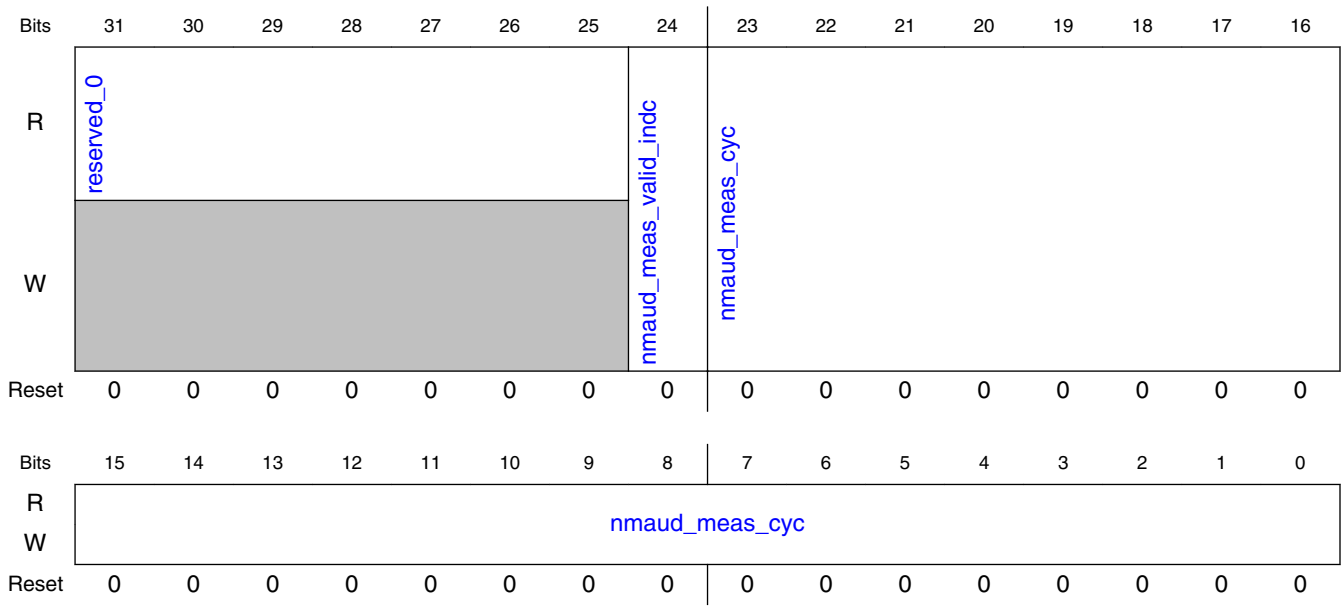
Field	Function
31-25 reserved_0	
24 nmvid_meas_valid_indc	When Toggle Valid pulse is generated to sample MNVID fix value
23-0 nmvid_meas_cyc	Fixed Value for NVID , The MVID is nmvid_ref_cyc[23:0]

13.4.10.1.55 Audio cycles measure (CM_AUD_MEAS)

13.4.10.1.55.1 Offset

Register	Offset
CM_AUD_MEAS	A2Ch

13.4.10.1.55.2 Diagram



13.4.10.1.55.3 Fields

Field	Function
31-25 reserved_0	
24 nmaud_meas_valid_indc	When Toggle Valid pulse is generated to sample MNAUD fix value
23-0 nmaud_meas_cyc	Fixed Value for NAUD, The MAUD is lane_ref_cyc[23:0]

13.4.10.1.56 I2S clock measurment HDMI (I2S_MEAS)

13.4.10.1.56.1 Offset

Register	Offset
I2S_MEAS	A40h

13.4.10.1.56.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								i2_meas							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	i2_meas															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.56.3 Fields

Field	Function
31-24 reserved_0	
23-0 i2_meas	I2S measurment value

13.4.10.1.57 I2S clock measurment DP (I2S_DP_MEAS)

13.4.10.1.57.1 Offset

Register	Offset
I2S_DP_MEAS	A44h

13.4.10.1.57.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								i2_dp_meas							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	i2_dp_meas															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.57.3 Fields

Field	Function
31-24 reserved_0	
23-0 i2_dp_meas	I2S Clock Meter measurment value (in DP)

13.4.10.1.58 SPDIF clock measurment DP (SPDIF_DP_MEAS)

13.4.10.1.58.1 Offset

Register	Offset
SPDIF_DP_MEAS	A80h

13.4.10.1.58.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								spdif_dp_meas							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_dp_meas															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.58.3 Fields

Field	Function
31-24 reserved_0	
23-0 spdif_dp_meas	SPDIF measurment value

13.4.10.1.59 SPDIF clock measurment HDMI (SPDIF_MEAS)

13.4.10.1.59.1 Offset

Register	Offset
SPDIF_MEAS	A84h

13.4.10.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								spdif_meas							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_meas															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.59.3 Fields

Field	Function
31-24 reserved_0	
23-0 spdif_meas	SPDIF Clock Meter measurment value (in DP)

13.4.10.1.60 Video clock measurment (NMVID_MEAS)

13.4.10.1.60.1 Offset

Register	Offset
NMVID_MEAS	AC0h

13.4.10.1.60.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								nmvid_meas							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	nmvid_meas															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.60.3 Fields

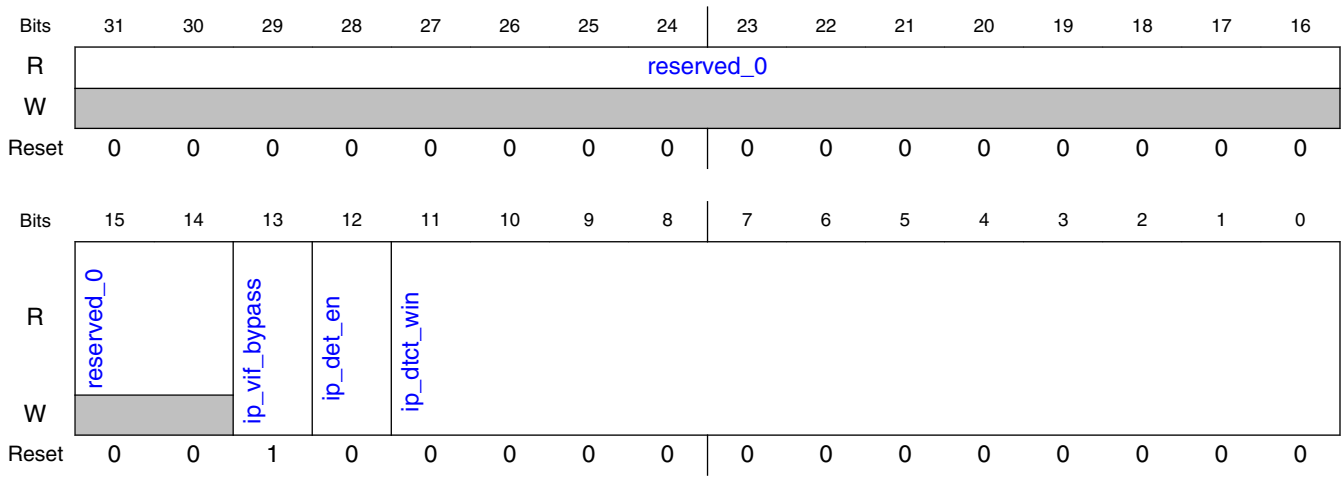
Field	Function
31-24 reserved_0	
23-0 nmvid_meas	Video clock measurment value

13.4.10.1.61 Video Input Interface Setting Register (BND_HSYNC2VSYNC)

13.4.10.1.61.1 Offset

Register	Offset
BND_HSYNC2VSYNC	B00h

13.4.10.1.61.2 Diagram



13.4.10.1.61.3 Fields

Field	Function
31-14 reserved_0	Reserved ignored on write. Reserved ignored on write.
13 ip_vif_bypass	Bypass video interface. Bypass video interface.
12 ip_det_en	Enable detection of Interlace formats after decided if the polarity is Automatic or Manual detection. Enable detection of Interlace formats after decided if the polarity is Automatic or Manual detection.
11-0 ip_dtct_win	Bound for HSYNC to VSYNC for all fields. Bound for HSYNC to VSYNC for all fields.

13.4.10.1.62 Status of HSYNC to VSYNC Distance Counter 1 (HSYNC2VSYNC_F1_L1)

13.4.10.1.62.1 Offset

Register	Offset
HSYNC2VSYNC_F1_L1	B04h

13.4.10.1.62.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ip_dtct_hsync2vsync_f1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.62.3 Fields

Field	Function
31-16 reserved_0	Reserved
15-0 ip_dtct_hsync2vsync_f1	Value of HSYNC to VSYNC field 1. Value of HSYNC to VSYNC field 1.

13.4.10.1.63 Status of HSYNC to VSYNC Distance Counter 2 (HSYNC2VSYNC_F2_L1)

13.4.10.1.63.1 Offset

Register	Offset
HSYNC2VSYNC_F2_L1	B08h

13.4.10.1.63.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	ip_dtct_hsync2vsync_f2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.63.3 Fields

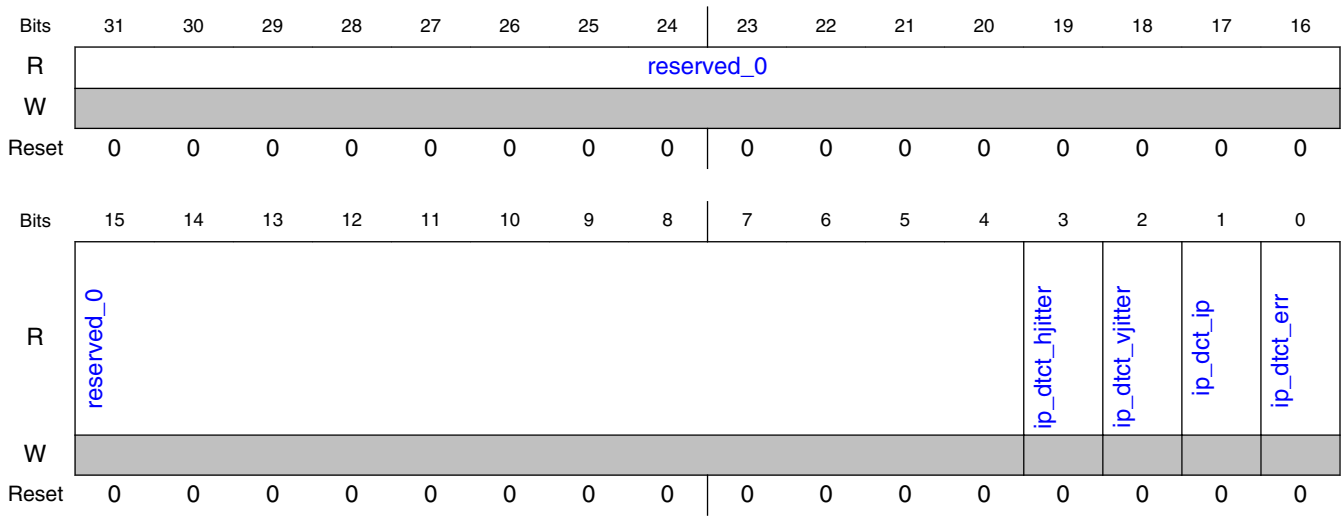
Field	Function
31-16 reserved_0	Reserved
15-0 ip_dtct_hsync2vsync_f2	Value of HSYNC to VSYNC field 2. Value of HSYNC to VSYNC field 2.

13.4.10.1.64 Video Interface Status Register (HSYNC2VSYNC_STATUS)

13.4.10.1.64.1 Offset

Register	Offset
HSYNC2VSYNC_STATUS	B0Ch

13.4.10.1.64.2 Diagram



13.4.10.1.64.3 Fields

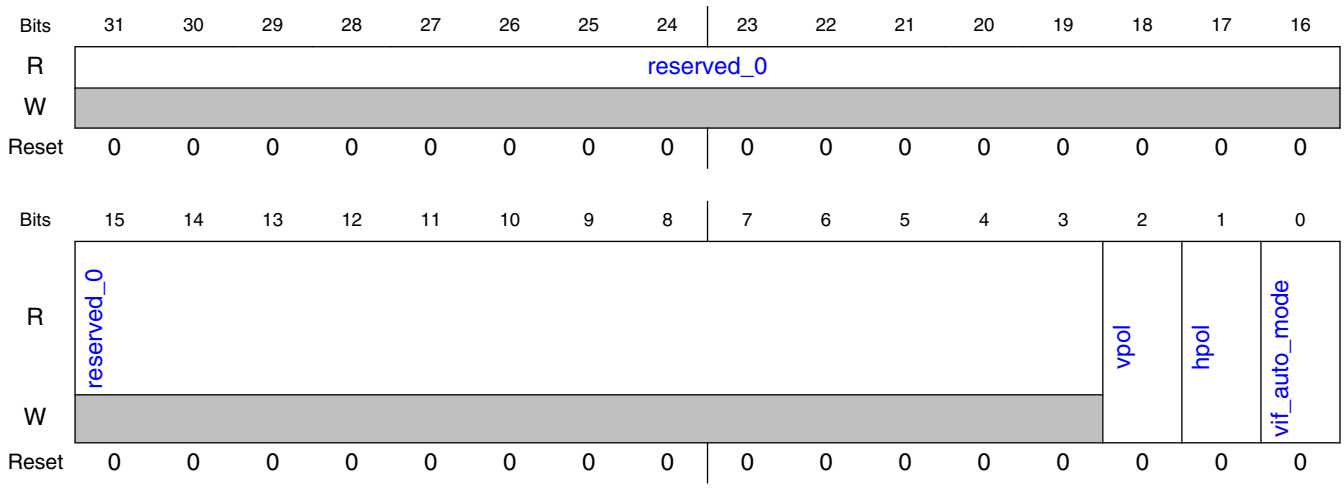
Field	Function
31-4 reserved_0	Reserved
3 ip_dtct_hjitter	Asserted when jitter is observed on htotal i. Asserted when jitter is observed on htotal i.e. HSYNC rising edge to next HSYNC rising edge delay count. Clear on Read.
2 ip_dtct_vjitter	Asserted when jitter is observed on vtotal i. Asserted when jitter is observed on vtotal i.e. VSYNC rising edge to next VSYNC rising edge delay count. Clear on Read.
1 ip_dct_ip	When asserted interlaced format is detected else progressive format. When asserted interlaced format is detected else progressive format.
0 ip_dtct_err	Asserted when HSYNC to VSYNC bound is violated. Asserted when HSYNC to VSYNC bound is violated. Clear on Read.

13.4.10.1.65 Setting Polarity of HSYNC and VSYNC (HSYNC2VSYNC_POL_CTRL)

13.4.10.1.65.1 Offset

Register	Offset
HSYNC2VSYNC_POL_CTRL	B10h

13.4.10.1.65.2 Diagram



13.4.10.1.65.3 Fields

Field	Function
31-3 reserved_0	Reserved ignored on write. Reserved ignored on write.
2 vpol	VSYNC polarity: 0-active HIGH 1-active LOW. VSYNC polarity: 0-active HIGH 1-active LOW.
1 hpol	HSYNC polarity: 0-active HIGH 1-active LOW. HSYNC polarity: 0-active HIGH 1-active LOW.
0 vif_auto_mode	Automatic or Manual configuration of the polarity: 0-vpol and hpol settings are used 1-automatic detection of polarity of input VSYNC and HSYNC

13.4.10.1.66 Video HSize configuration (SCHEDULER_H_SIZE)

13.4.10.1.66.1 Offset

Register	Offset
SCHEDULER_H_SIZE	1000h

13.4.10.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	h_active_size															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	h_blank_size															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.66.3 Fields

Field	Function
31-16 h_active_size	H-Active Size of video
15-0 h_blank_size	H-Blank Size of video

13.4.10.1.67 Video VSize configuration (SCHEDULER_V_SIZE)

13.4.10.1.67.1 Offset

Register	Offset
SCHEDULER_V_SIZE	1004h

13.4.10.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	v_active_size															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	v_blank_size															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.67.3 Fields

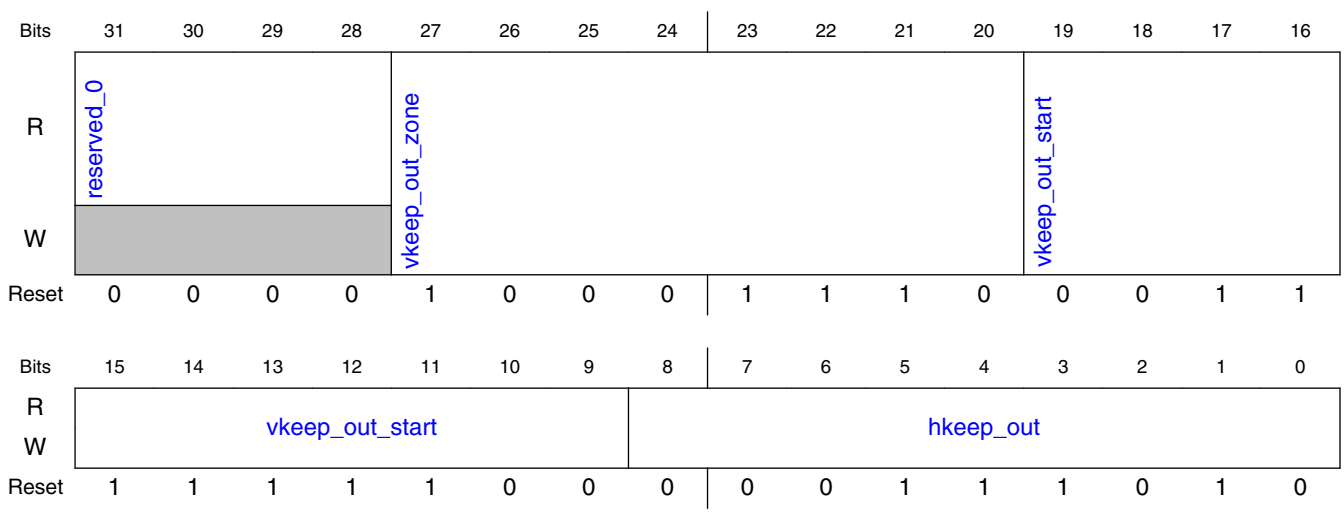
Field	Function
31-16 v_active_size	V-Active Size of video
15-0 v_blank_size	V-Blank Size of video

13.4.10.1.68 Video Blank (Keep Out) Size configuration (SCHEDULER_KEEP_OUT)

13.4.10.1.68.1 Offset

Register	Offset
SCHEDULER_KEEP_OUT	1008h

13.4.10.1.68.2 Diagram



13.4.10.1.68.3 Fields

Field	Function
31-28	

Table continues on the next page...

Field	Function
reserved_0	
27-20 vkeep_out_zone	V Blank Keep out zone
19-9 vkeep_out_start	V Blank Keep out buffer from rise of VSYNC
8-0 hkeep_out	H Blank Keep out size

13.4.10.1.69 Video Blank (Front) Size configuration (HDTX_SIGNAL_FRONT_WIDTH)

13.4.10.1.69.1 Offset

Register	Offset
HDTX_SIGNAL_FRONT_WIDTH	100Ch

13.4.10.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	vfront															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	hfront															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.69.3 Fields

Field	Function
31-16 vfront	V Front Size. V Front Size.
15-0 hfront	H Front Size. H Front Size.

13.4.10.1.70 Video Sync Size configuration (HDTX_SIGNAL_SYNC_WIDTH)

13.4.10.1.70.1 Offset

Register	Offset
HDTX_SIGNAL_SYNC_WIDTH	1010h

13.4.10.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	vsync															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	hsync															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.70.3 Fields

Field	Function
31-16 vsync	V Sync Size. V Sync Size.
15-0 hsync	H Sync Size. H Sync Size.

13.4.10.1.71 Video Back Size configuration (HDTX_SIGNAL_BACK_WIDTH)

13.4.10.1.71.1 Offset

Register	Offset
HDTX_SIGNAL_BACK_WIDTH	1014h

13.4.10.1.71.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	vback															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	hback															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.71.3 Fields

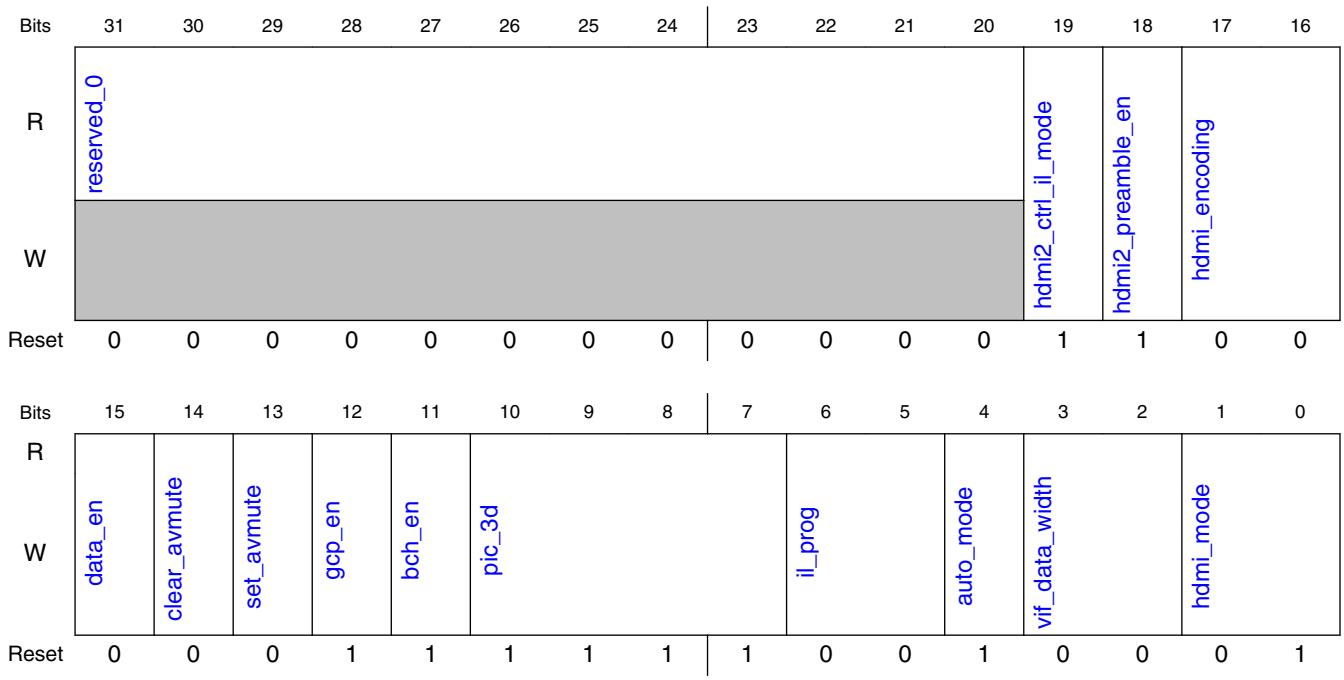
Field	Function
31-16 vback	V Back Size. V Back Size.
15-0 hback	H Back Size. H Back Size.

13.4.10.1.72 Video Mode configuration (HDTX_CONTROLLER)

13.4.10.1.72.1 Offset

Register	Offset
HDTX_CONTROLLER	1018h

13.4.10.1.72.2 Diagram



13.4.10.1.72.3 Fields

Field	Function
31-20 reserved_0	
19 hdmi2_ctrl_il_mode	HDMI2 Interlace mode: 1 - SSCP is generated after every vsync in frame 0 - SSCP is generated after first vsync of frame only
18 hdmi2_preamble_en	HDMI2 preamble enabled:
17-16 hdmi_encoding	HDMI pixel encoding
15 data_en	Data Transmission Enable:
14 clear_avmute	Clear_AVMute Value
13 set_avmute	Set_AVMute Value
12 gcp_en	Transmit GCP Enable:
11	BCH Error Correction Algorithm is enabled:

Table continues on the next page...

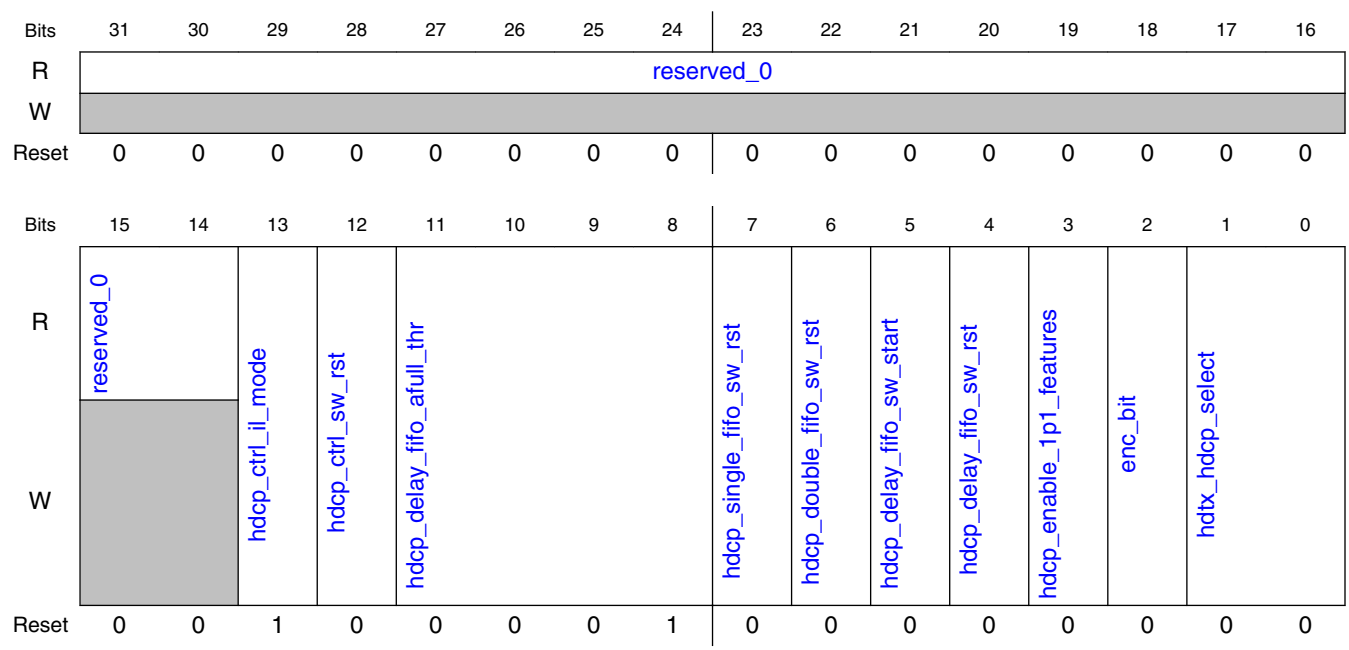
Field	Function
bch_en	
10-7 pic_3d	Picture in 3d mode:
6-5 il_prog	Interlaced or progressive video mode:
4 auto_mode	Frame sizing auto mode:
3-2 vif_data_width	Video I/F Data width - color depth feature:
1-0 hdmi_mode	HDMI mode operation

13.4.10.1.73 HDMi HDCP configuration (HDTX_HDCP)

13.4.10.1.73.1 Offset

Register	Offset
HDTX_HDCP	101Ch

13.4.10.1.73.2 Diagram



13.4.10.1.73.3 Fields

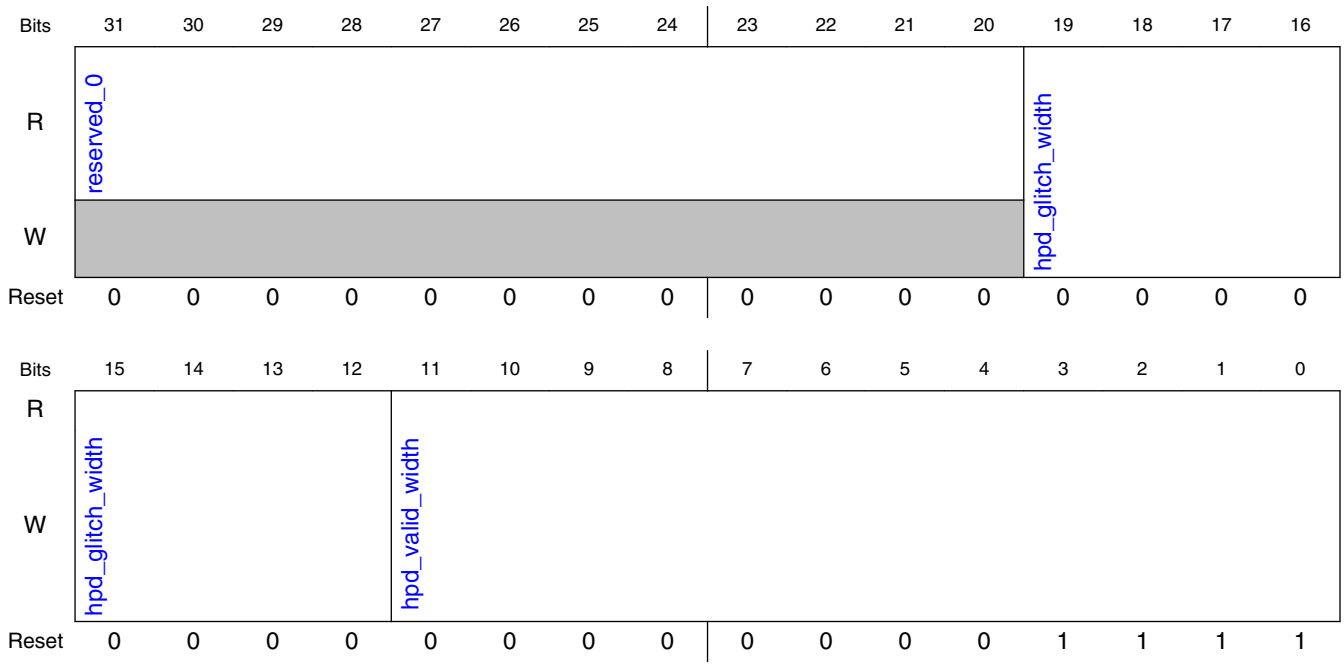
Field	Function
31-14 reserved_0	
13 hdcp_ctrl_il_mode	HDCP Interlace mode: 1 - encoding and vertical keep out on all vsync's in frame 0 - encoding and vertical keep out on first vsync of frame only
12 hdcp_ctrl_sw_rst	HDCP Control path SW reset
11-8 hdcp_delay_fifo_afull_thr	hdcp delay fifo full threshold
7 hdcp_single_fifo_sw_rst	hdcp single fifo sw reset
6 hdcp_double_fifo_sw_rst	hdcp double fifo sw reset
5 hdcp_delay_fifo_sw_start	hdcp delay fifo sw start
4 hdcp_delay_fifo_sw_rst	hdcp delay fifo sw reset
3 hdcp_enable_1p1_features	hdcp enabling 1p1
2 enc_bit	hdcp encoding bit
1-0 hdtx_hdcp_select	HDCP Select

13.4.10.1.74 HDMI HPD configuration (HDTX_HPD)

13.4.10.1.74.1 Offset

Register	Offset
HDTX_HPD	1020h

13.4.10.1.74.2 Diagram



13.4.10.1.74.3 Fields

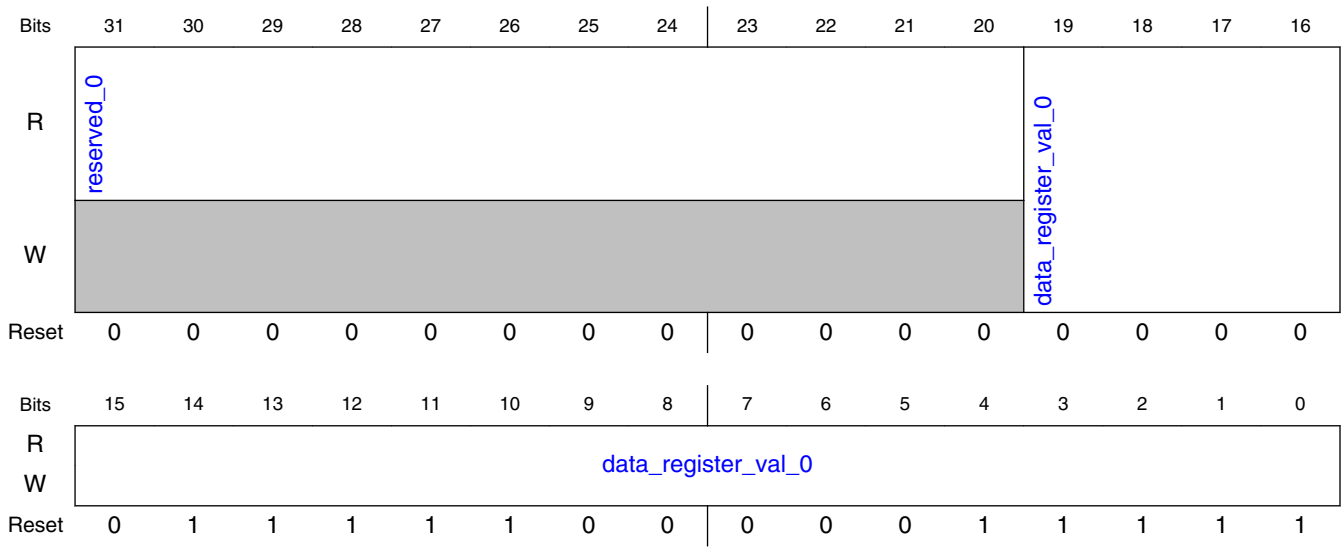
Field	Function
31-20 reserved_0	
19-12 hpd_glitch_width	hdp glitch valid
11-0 hpd_valid_width	hdp valid valid

13.4.10.1.75 HDMI CLOCK configuration - REG0 (HDTX_CLOCK_REG_0)

13.4.10.1.75.1 Offset

Register	Offset
HDTX_CLOCK_REG_0	1024h

13.4.10.1.75.2 Diagram



13.4.10.1.75.3 Fields

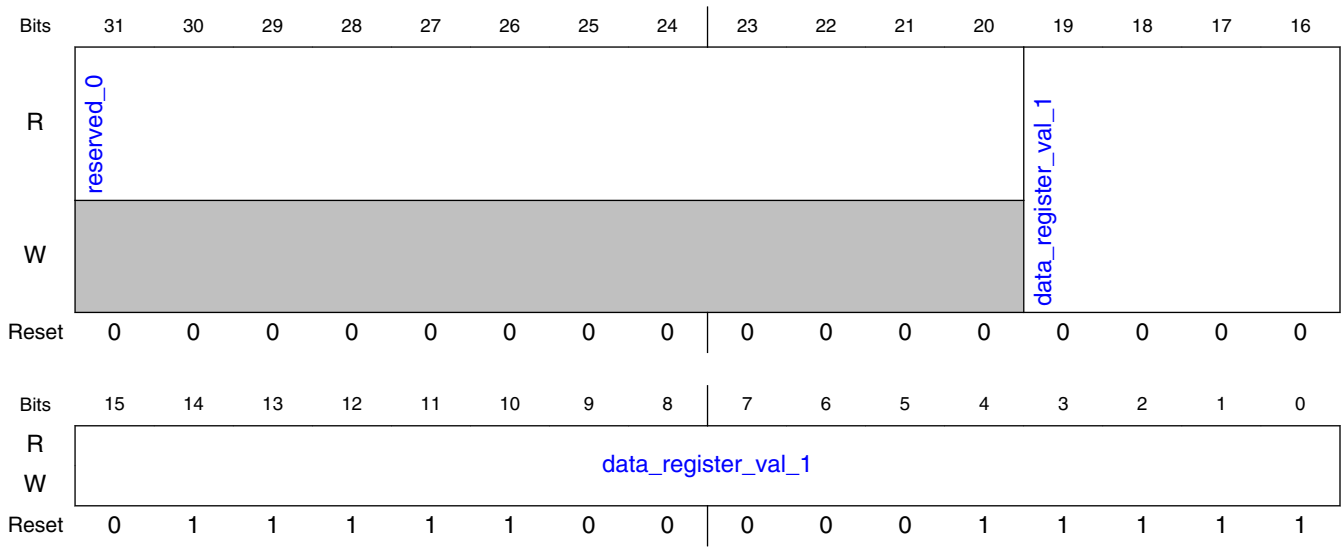
Field	Function
31-20 reserved_0	
19-0 data_register_val_0	data register 0 - 10 LSB is used for character clock

13.4.10.1.76 HDMI CLOCK configuration - REG1 (HDTX_CLOCK_REG_1)

13.4.10.1.76.1 Offset

Register	Offset
HDTX_CLOCK_REG_1	1028h

13.4.10.1.76.2 Diagram



13.4.10.1.76.3 Fields

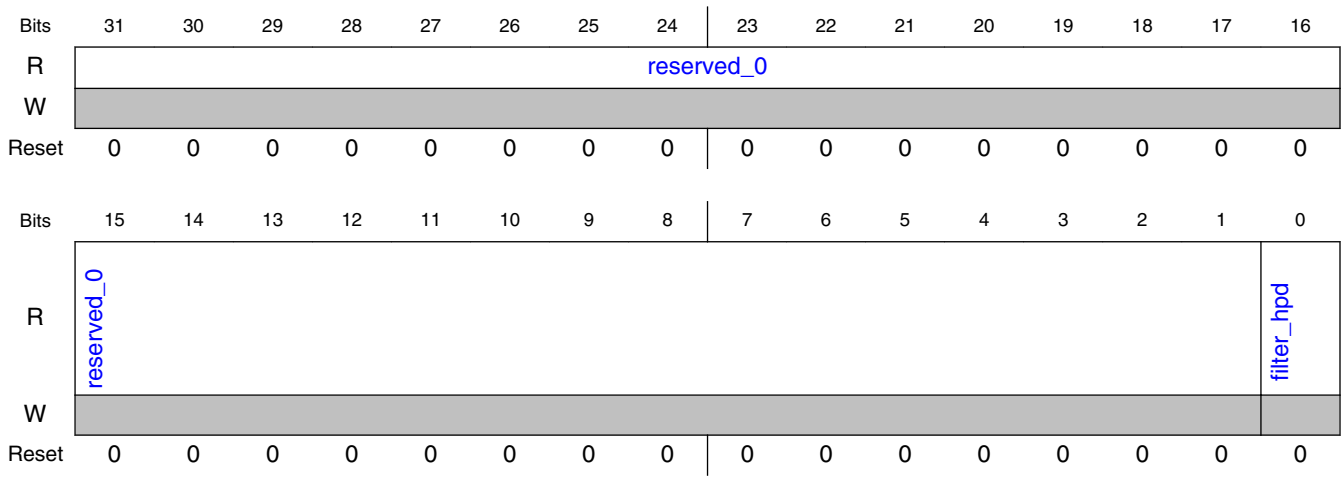
Field	Function
31-20 reserved_0	
19-0 data_register_val_1	data register 1

13.4.10.1.77 HDMI HPD status (HPD_PLUG_IN)

13.4.10.1.77.1 Offset

Register	Offset
HPD_PLUG_IN	102Ch

13.4.10.1.77.2 Diagram



13.4.10.1.77.3 Fields

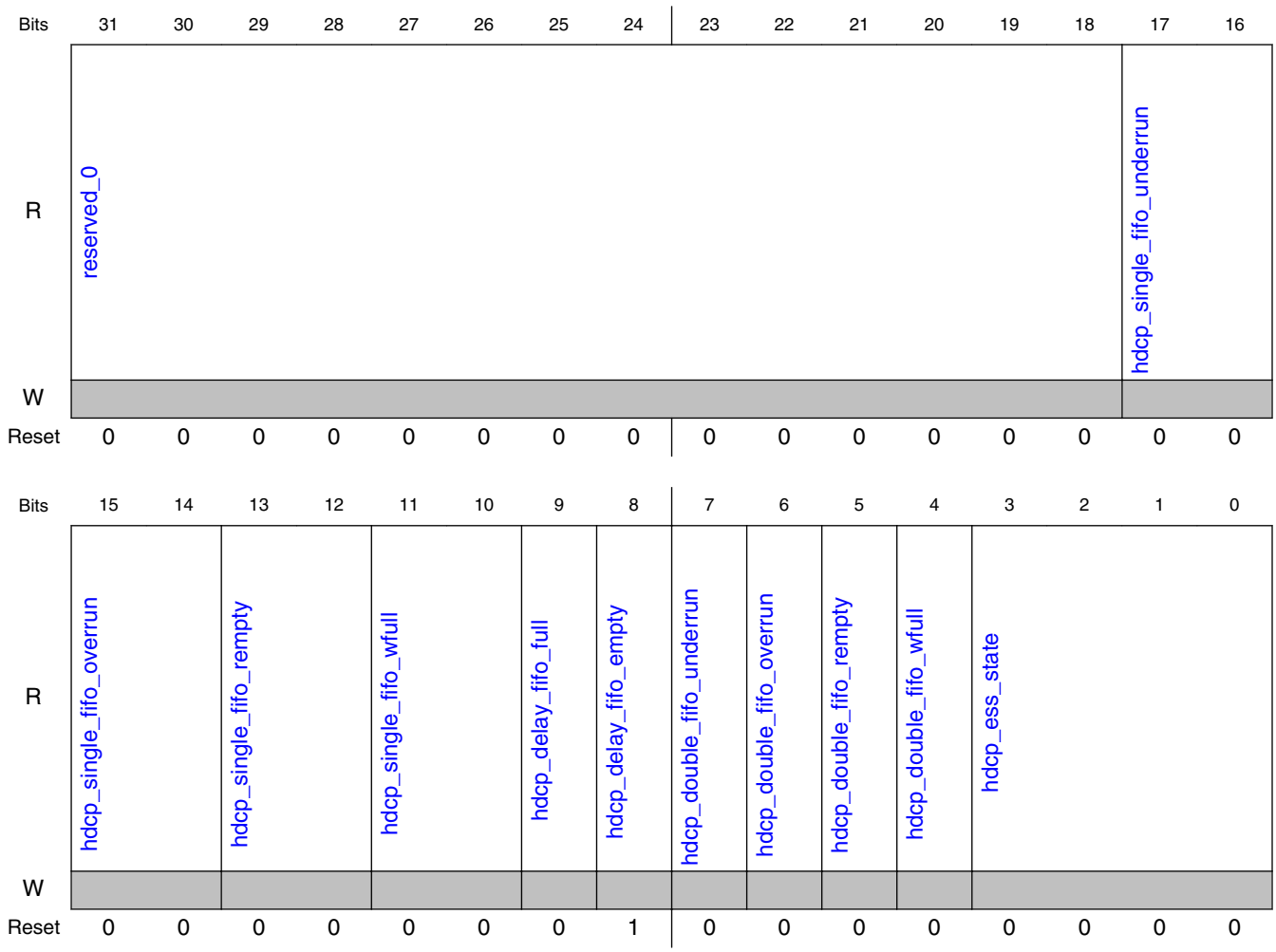
Field	Function
31-1 reserved_0	
0 filter_hpd	Filter HPD input

13.4.10.1.78 HDMI status (HDCP_IN)

13.4.10.1.78.1 Offset

Register	Offset
HDCP_IN	1030h

13.4.10.1.78.2 Diagram



13.4.10.1.78.3 Fields

Field	Function
31-18 reserved_0	
17-16 hdcp_single_fifo_underrun	
15-14 hdcp_single_fifo_overrun	
13-12 hdcp_single_fifo_rempty	

Table continues on the next page...

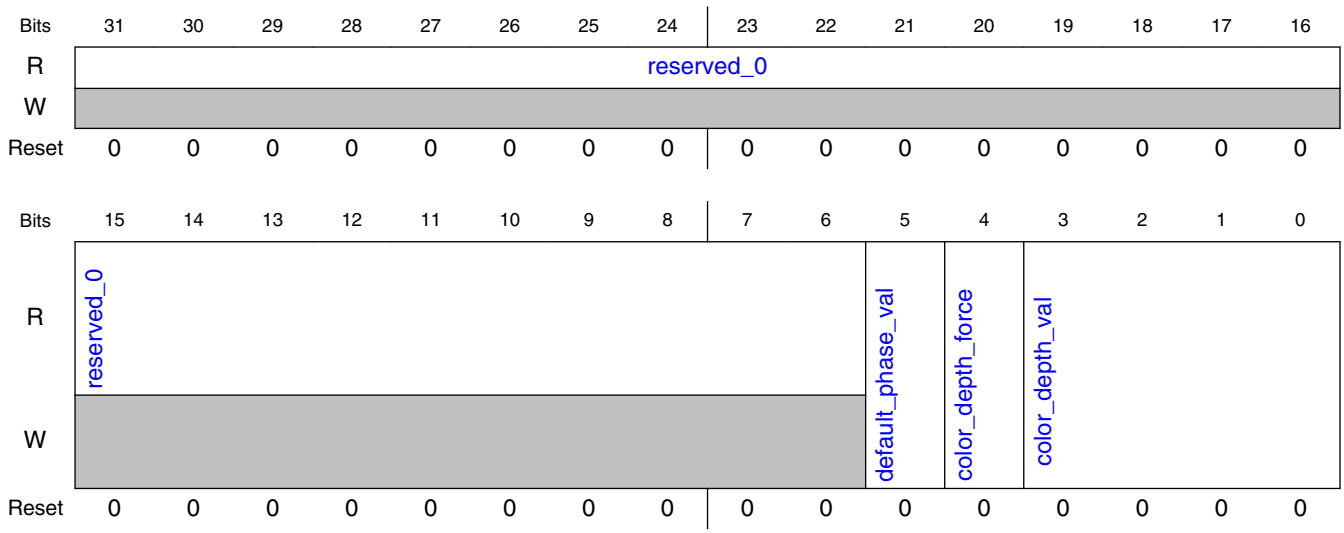
Field	Function
11-10 hdcp_single_fifo_wfull	
9 hdcp_delay_fifo_full	
8 hdcp_delay_fifo_empty	
7 hdcp_double_fifo_underrun	
6 hdcp_double_fifo_overrun	
5 hdcp_double_fifo_rempty	
4 hdcp_double_fifo_wfull	
3-0 hdcp_ess_state	ESS State Machine

13.4.10.1.79 HDMI GCP coding configuration (GCP_FORCE_COLOR_DEPTH_CODING)

13.4.10.1.79.1 Offset

Register	Offset
GCP_FORCE_COLOR_DEPTH_CODING	1034h

13.4.10.1.79.2 Diagram



13.4.10.1.79.3 Fields

Field	Function
31-6 reserved_0	
5 default_phase_val	Default Phase Value Forced
4 color_depth_force	Enable Forcing Color Depth Value. Enable Forcing Color Depth Value.
3-0 color_depth_val	Color Depth Value Forced

13.4.10.1.80 HDMI SSCP positionining (SSCP_POSITIONING)

13.4.10.1.80.1 Offset

Register	Offset
SSCP_POSITIONING	1038h

13.4.10.1.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sscp_col_val															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	sscp_row_val															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1

13.4.10.1.80.3 Fields

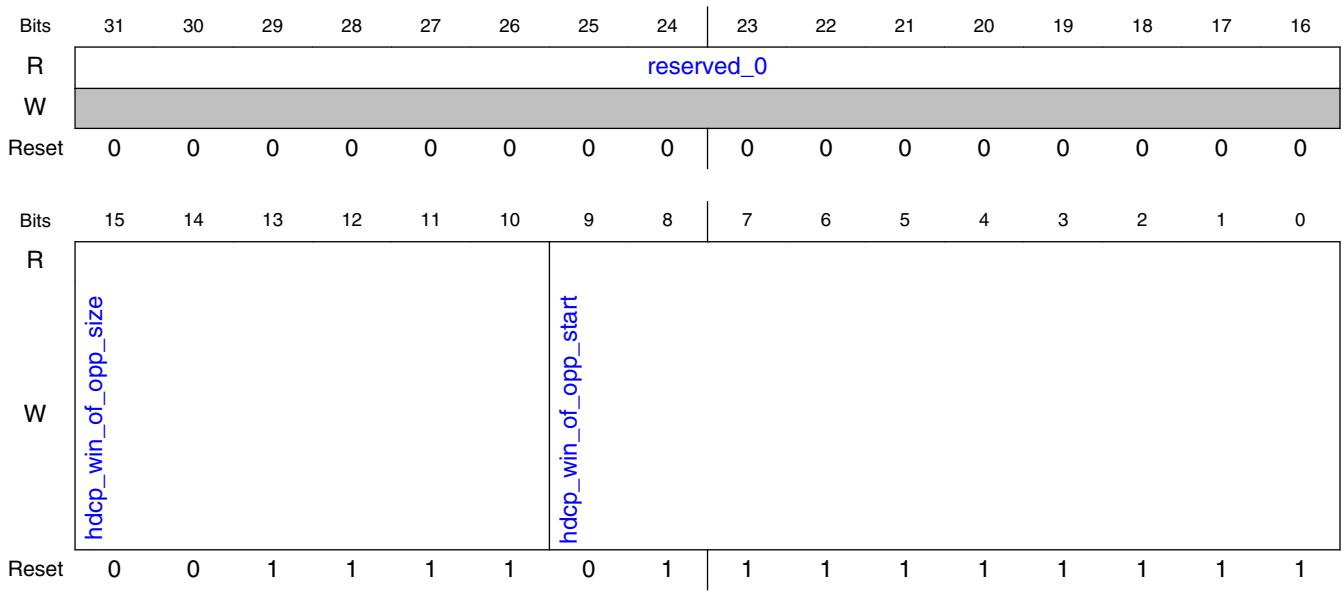
Field	Function
31-16 sscp_col_val	Colomn value of start of SSCP from end of Default
15-0 sscp_row_val	Row value of DE of the SSCP

13.4.10.1.81 HDMI HDCP window opportunity (HDCP_WIN_OF_OPP_POSITION)

13.4.10.1.81.1 Offset

Register	Offset
HDCP_WIN_OF_OPP_POSITION	103Ch

13.4.10.1.81.2 Diagram



13.4.10.1.81.3 Fields

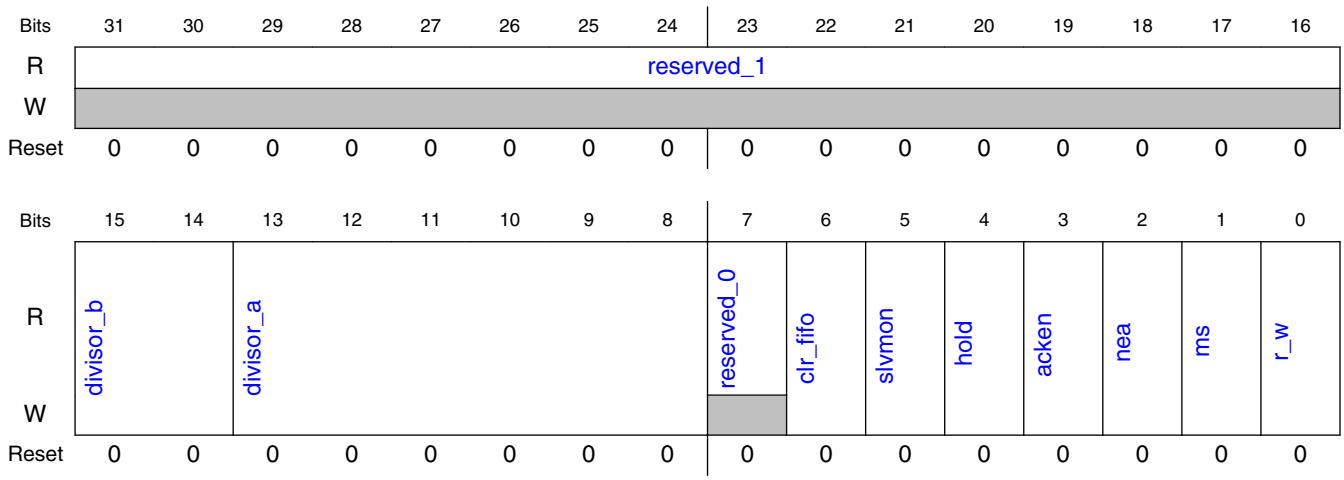
Field	Function
31-16 reserved_0	
15-10 hdcp_win_of_opp_size	Size of window of opportunity
9-0 hdcp_win_of_opp_start	Position of start of window of opportunity

13.4.10.1.82 HDMI I2C configuration (CTRL)

13.4.10.1.82.1 Offset

Register	Offset
CTRL	1800h

13.4.10.1.82.2 Diagram



13.4.10.1.82.3 Fields

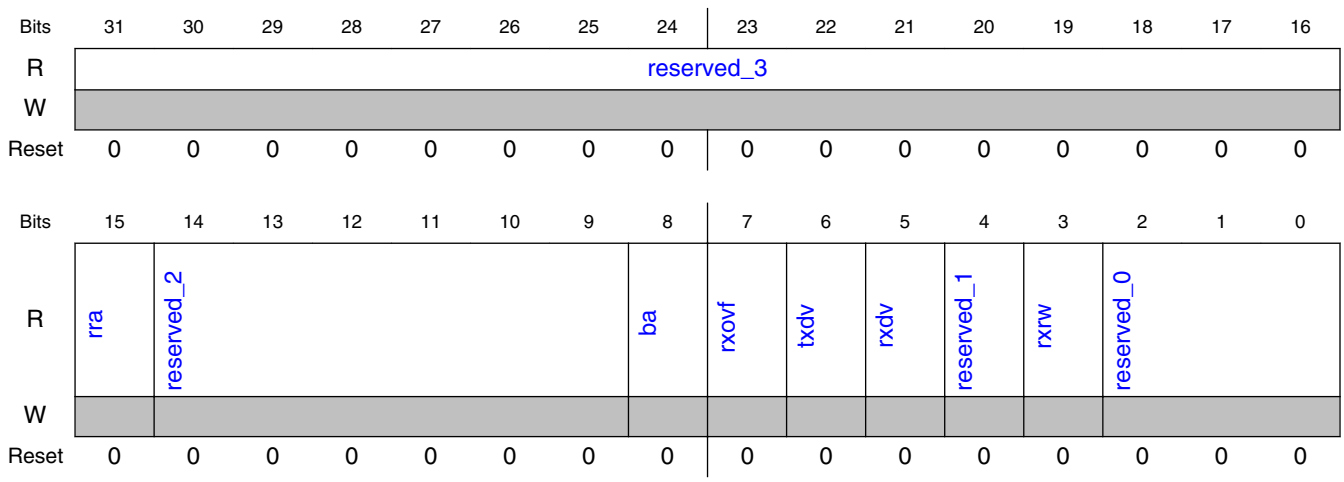
Field	Function
31-16 reserved_1	
15-14 divisor_b	
13-8 divisor_a	
7 reserved_0	
6 clr_fifo	
5 slvmon	
4 hold	
3 acken	
2 nea	
1 ms	
0 r_w	

13.4.10.1.83 HDMI I2C status (STATUS)

13.4.10.1.83.1 Offset

Register	Offset
STATUS	1804h

13.4.10.1.83.2 Diagram



13.4.10.1.83.3 Fields

Field	Function
31-16 reserved_3	
15 rra	
14-9 reserved_2	
8 ba	
7 rxovf	
6 txdv	
5	

Table continues on the next page...

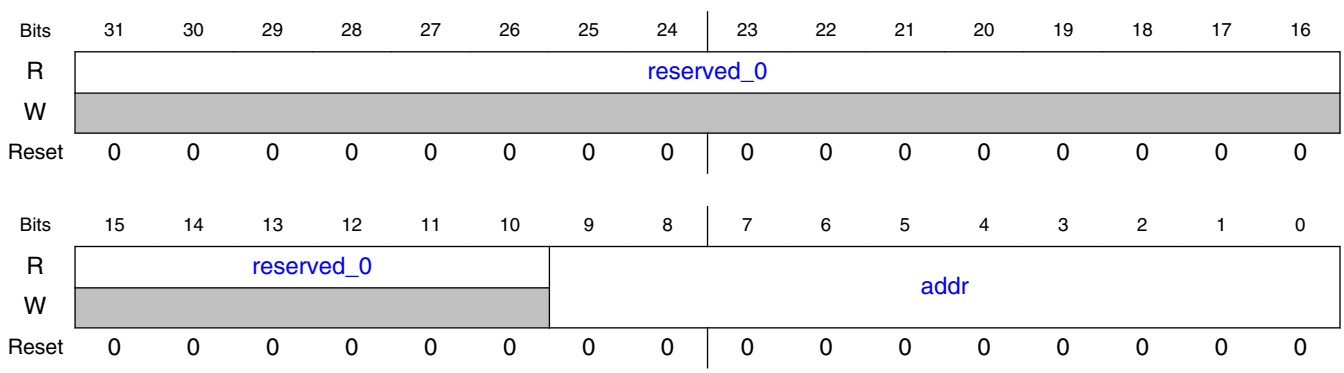
Field	Function
rxdv	
4 reserved_1	
3 rxrw	
2-0 reserved_0	

13.4.10.1.84 HDMI I2C Address configuration (I2C_ADDR)

13.4.10.1.84.1 Offset

Register	Offset
I2C_ADDR	1808h

13.4.10.1.84.2 Diagram



13.4.10.1.84.3 Fields

Field	Function
31-10 reserved_0	
9-0 addr	

13.4.10.1.85 HDMI I2C data register (I2C_DATA)

13.4.10.1.85.1 Offset

Register	Offset
I2C_DATA	180Ch

13.4.10.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								data							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.85.3 Fields

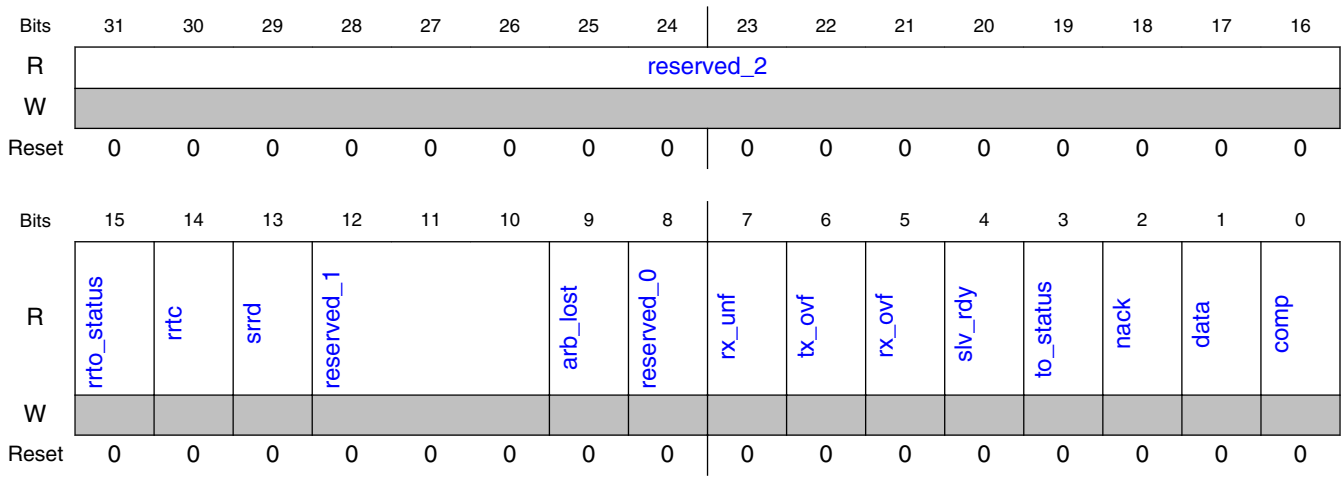
Field	Function
31-8 reserved_0	
7-0 data	

13.4.10.1.86 HDMI I2C interrupt status (INT_STATUS)

13.4.10.1.86.1 Offset

Register	Offset
INT_STATUS	1810h

13.4.10.1.86.2 Diagram



13.4.10.1.86.3 Fields

Field	Function
31-16 reserved_2	
15 rrto_status	
14 rrtc	
13 srrd	
12-10 reserved_1	
9 arb_lost	
8 reserved_0	
7 rx_unf	
6 tx_ovf	
5 rx_ovf	
4 slv_rdy	
3	

Table continues on the next page...

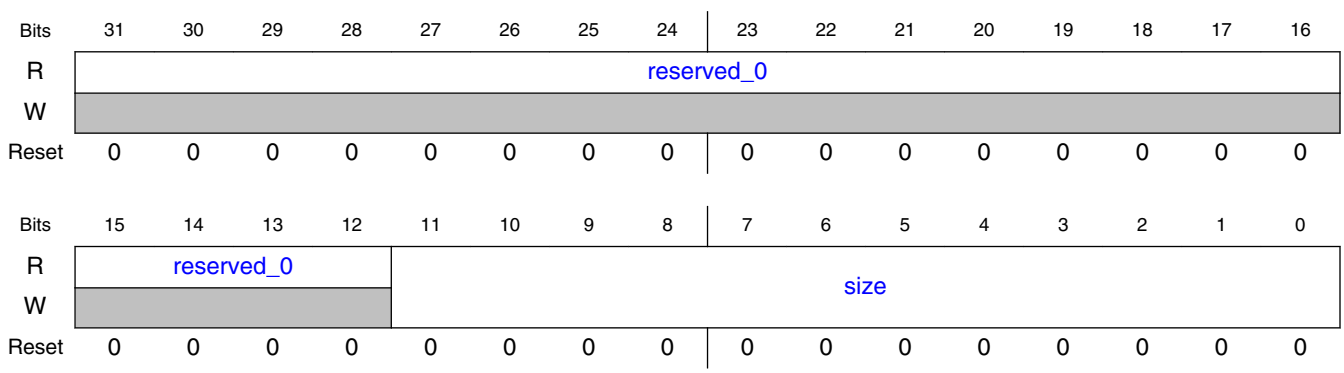
Field	Function
to_status	
2 nack	
1 data	
0 comp	

13.4.10.1.87 HDMI I2C transaction size (TRANS_SIZE)

13.4.10.1.87.1 Offset

Register	Offset
TRANS_SIZE	1814h

13.4.10.1.87.2 Diagram



13.4.10.1.87.3 Fields

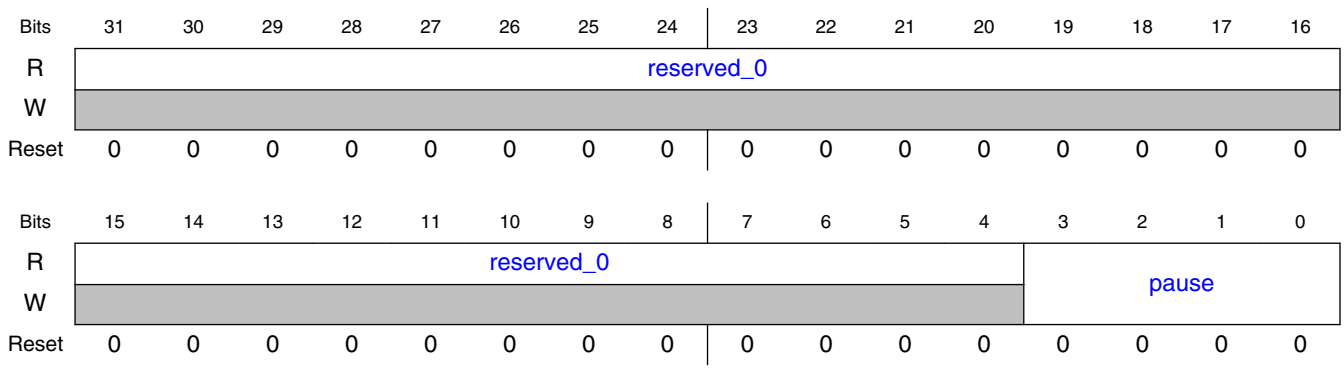
Field	Function
31-12 reserved_0	
11-0 size	

13.4.10.1.88 HDMI I2C Slave Monitor (SLAVE_MON)

13.4.10.1.88.1 Offset

Register	Offset
SLAVE_MON	1818h

13.4.10.1.88.2 Diagram



13.4.10.1.88.3 Fields

Field	Function
31-4 reserved_0	
3-0 pause	

13.4.10.1.89 HDMI I2C Timeout (TIMEOUT)

13.4.10.1.89.1 Offset

Register	Offset
TIMEOUT	181Ch

13.4.10.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								to_timeout							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

13.4.10.1.89.3 Fields

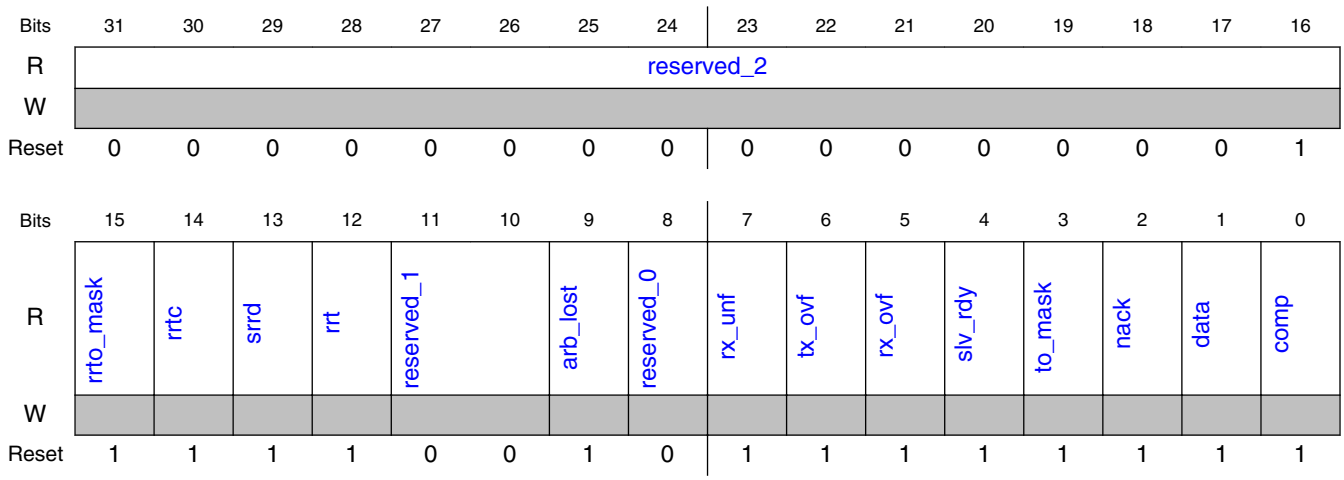
Field	Function
31-8 reserved_0	
7-0 to_timeout	

13.4.10.1.90 HDMI SCDC interrupt mask (INT_SCDC_MASK)

13.4.10.1.90.1 Offset

Register	Offset
INT_SCDC_MASK	1820h

13.4.10.1.90.2 Diagram



13.4.10.1.90.3 Fields

Field	Function
31-16 reserved_2	
15 rrto_mask	
14 rrtc	
13 srrd	
12 rrt	
11-10 reserved_1	
9 arb_lost	
8 reserved_0	
7 rx_unf	
6 tx_ovf	
5 rx_ovf	
4	

Table continues on the next page...

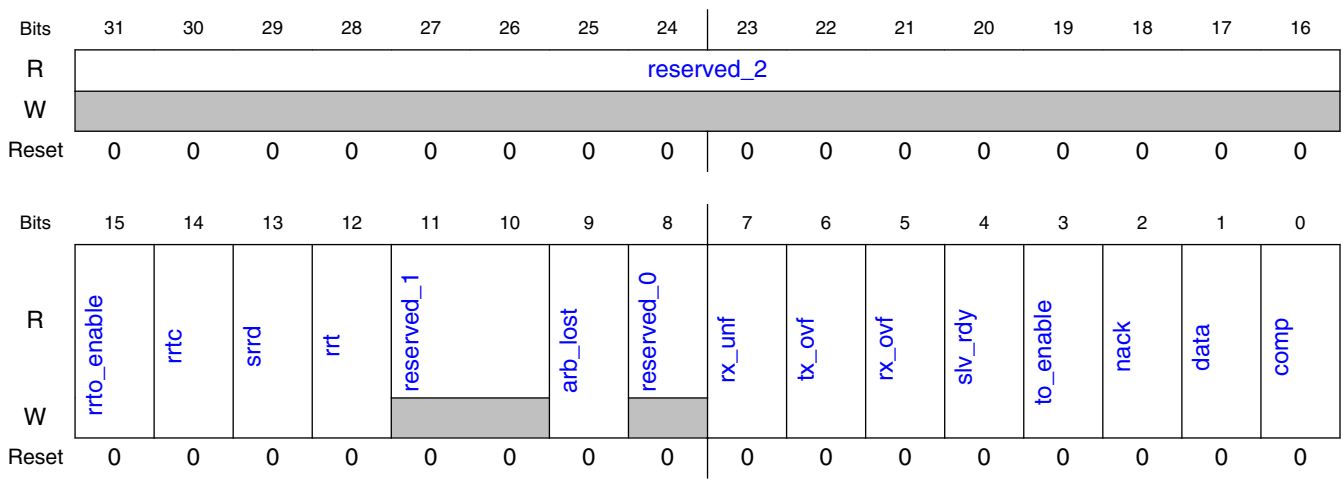
Field	Function
slv_rdy	
3 to_mask	
2 nack	
1 data	
0 comp	

13.4.10.1.91 HDMI I2C interrupt enable (INT_ENABLE)

13.4.10.1.91.1 Offset

Register	Offset
INT_ENABLE	1824h

13.4.10.1.91.2 Diagram



13.4.10.1.91.3 Fields

Field	Function
31-16	

Table continues on the next page...

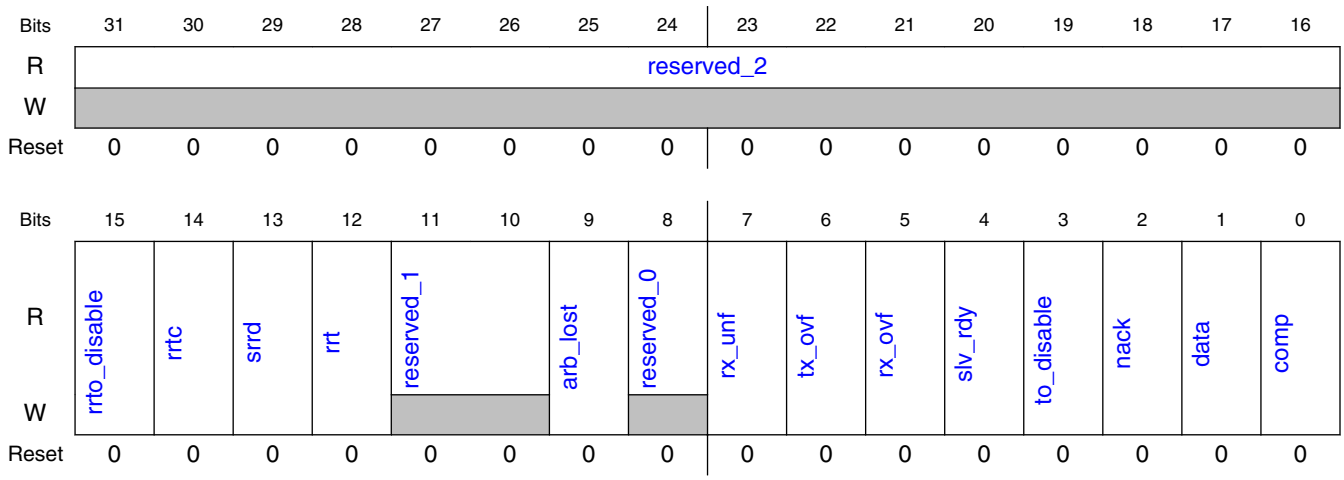
Field	Function
reserved_2	
15 rrto_enable	
14 rrtc	
13 srrd	
12 rrt	
11-10 reserved_1	
9 arb_lost	
8 reserved_0	
7 rx_unf	
6 tx_ovf	
5 rx_ovf	
4 slv_rdy	
3 to_enable	
2 nack	
1 data	
0 comp	

13.4.10.1.92 HDMI I2C interrupt disable (INT_DISABLE)

13.4.10.1.92.1 Offset

Register	Offset
INT_DISABLE	1828h

13.4.10.1.92.2 Diagram



13.4.10.1.92.3 Fields

Field	Function
31-16 reserved_2	
15 rrto_disable	
14 rrtc	
13 srrd	
12 rrt	
11-10 reserved_1	
9 arb_lost	
8 reserved_0	
7 rx_unf	
6 tx_ovf	
5 rx_ovf	

Table continues on the next page...

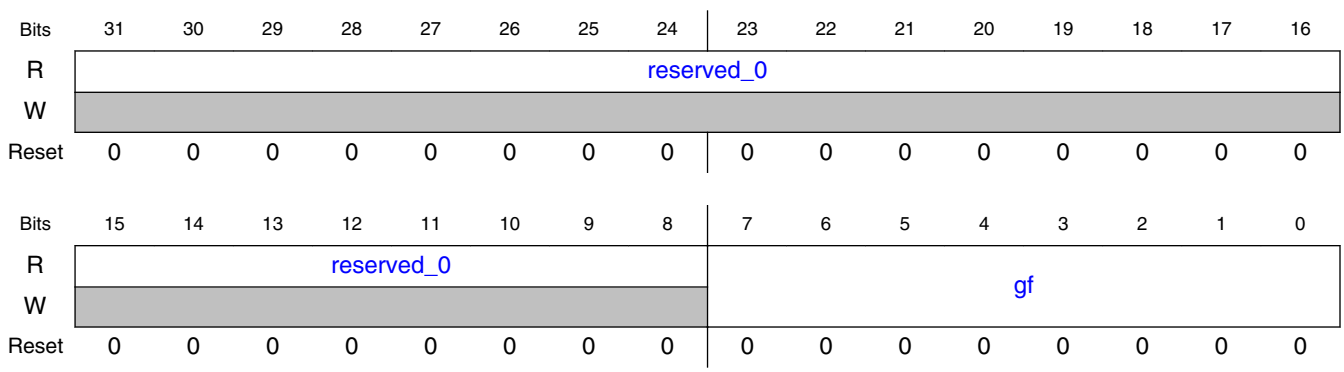
Field	Function
4 slv_rdy	
3 to_disable	
2 nack	
1 data	
0 comp	

13.4.10.1.93 HDMI I2C glitch filter configuration (GLITCH_FILTER_CTRL)

13.4.10.1.93.1 Offset

Register	Offset
GLITCH_FILTER_CTRL	182Ch

13.4.10.1.93.2 Diagram



13.4.10.1.93.3 Fields

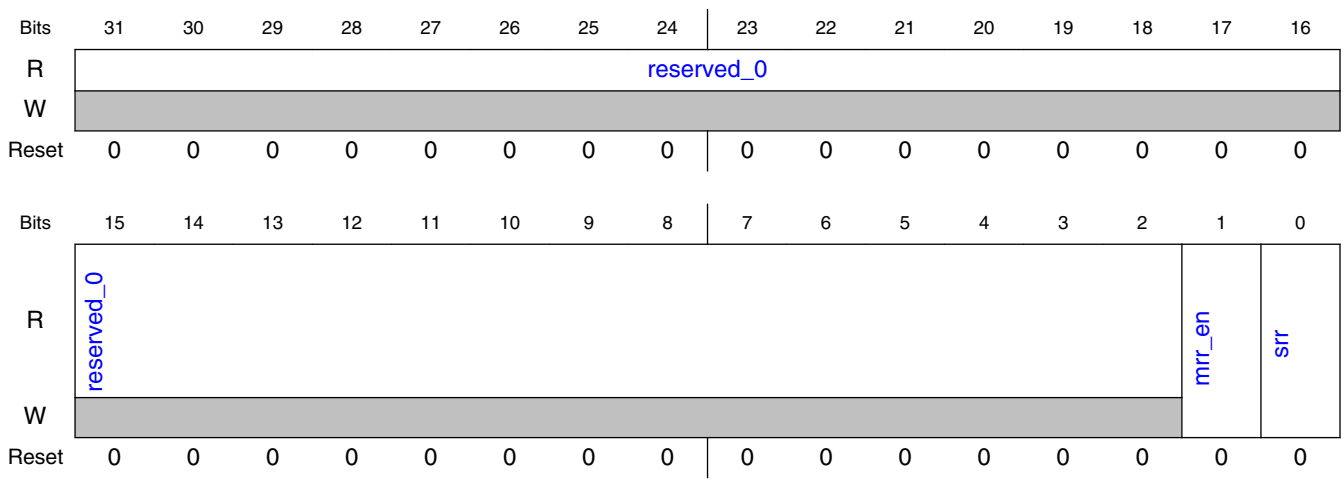
Field	Function
31-8 reserved_0	
7-0 gf	

13.4.10.1.94 HDMI I2C RR configuration (RR_CTRL)

13.4.10.1.94.1 Offset

Register	Offset
RR_CTRL	1830h

13.4.10.1.94.2 Diagram



13.4.10.1.94.3 Fields

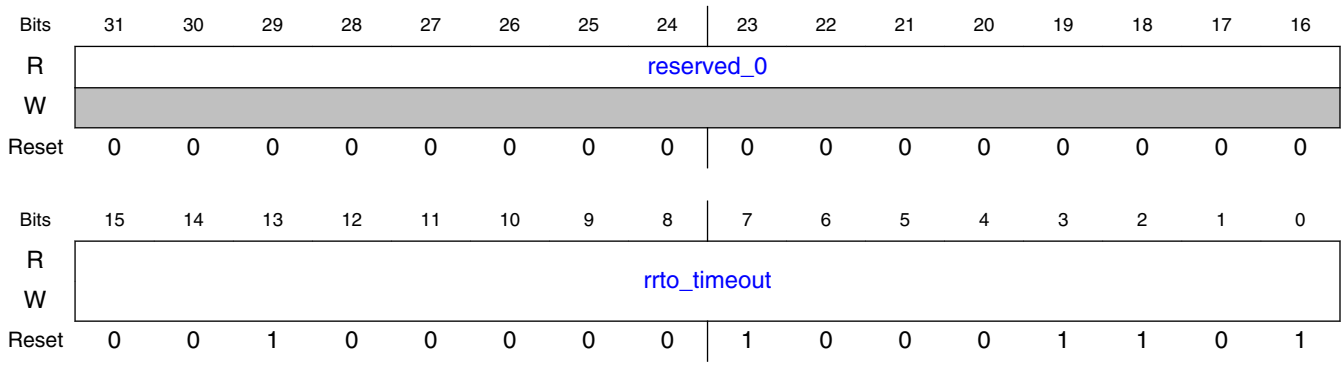
Field	Function
31-2 reserved_0	
1 mrr_en	
0 srr	

13.4.10.1.95 HDMI I2C RR timeout (RR_TIMEOUT)

13.4.10.1.95.1 Offset

Register	Offset
RR_TIMEOUT	1834h

13.4.10.1.95.2 Diagram



13.4.10.1.95.3 Fields

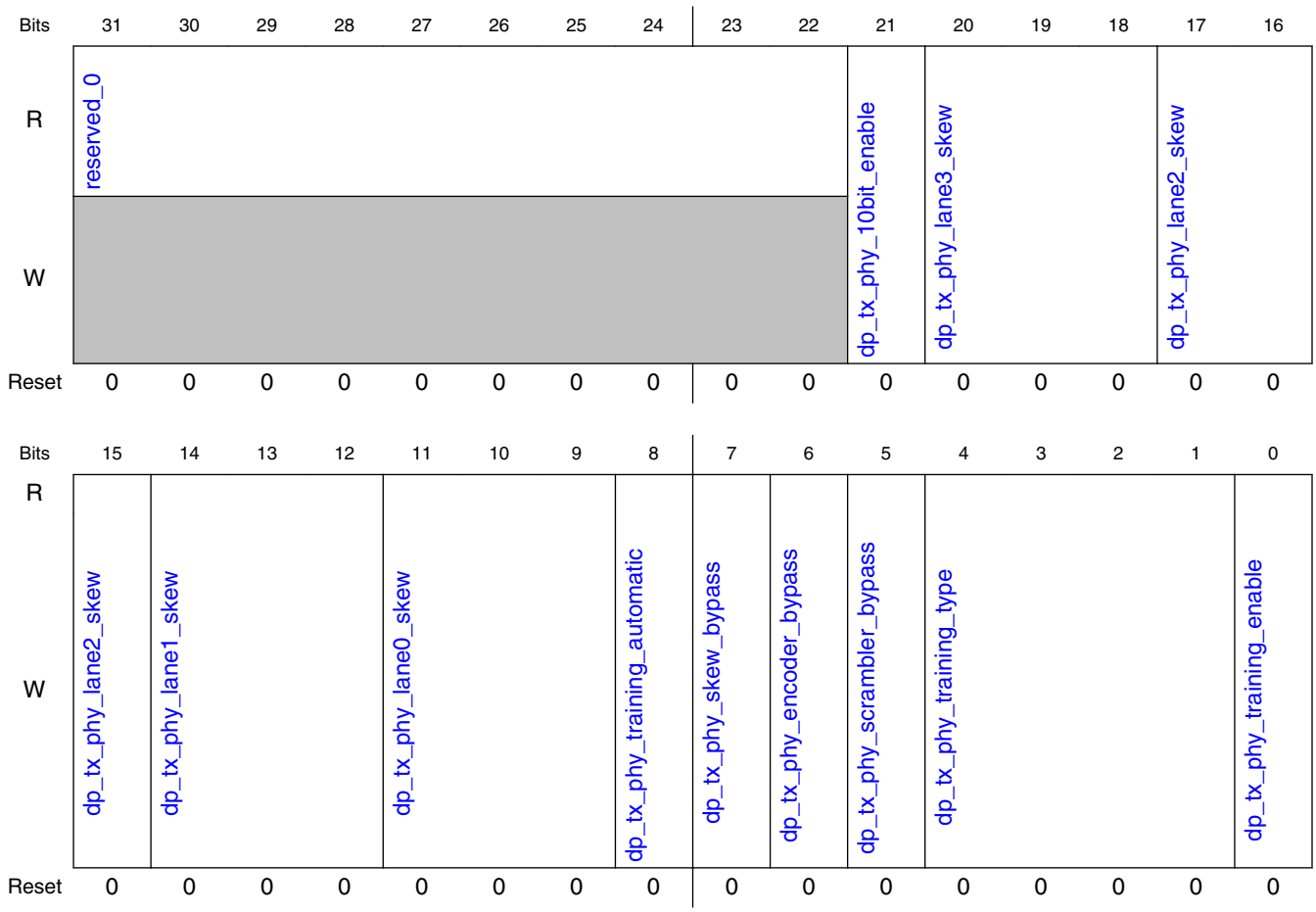
Field	Function
31-16 reserved_0	
15-0 rrto_timeout	

13.4.10.1.96 DPTX PHY control (DP_TX_PHY_CONFIG_REG)

13.4.10.1.96.1 Offset

Register	Offset
DP_TX_PHY_CONFIG_REG	2000h

13.4.10.1.96.2 Diagram



13.4.10.1.96.3 Fields

Field	Function
31-22 reserved_0	
21 dp_tx_phy_10bit_enable	Used to enable the 10-bit mode. Used to enable the 10-bit mode. Active high.
20-18 dp_tx_phy_lane3_skew	Specifies the programmable lane3 skew. Specifies the programmable lane3 skew.
17-15 dp_tx_phy_lane2_skew	Specifies the programmable lane2 skew. Specifies the programmable lane2 skew.
14-12	Specifies the programmable lane1 skew. Specifies the programmable lane1 skew.

Table continues on the next page...

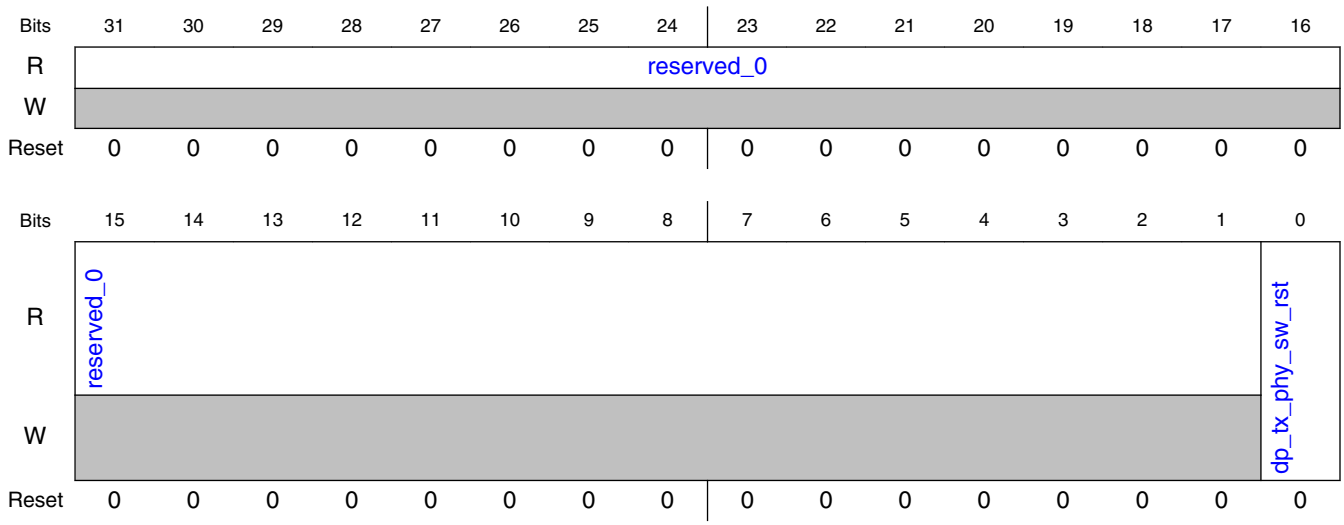
Field	Function
dp_tx_phy_lane 1_skew	
11-9 dp_tx_phy_lane 0_skew	Specifies the programmable lane0 skew. Specifies the programmable lane0 skew.
8 dp_tx_phy_traini ng_automatic	Used to bypass 8/10 encoder and scrambler during link training. Used to bypass 8/10 encoder and scrambler during link training. Bit is sampled only when dp_tx_phy_training_enable bit is set.
7 dp_tx_phy_skew _bypass	Used to bypass the lane skew. Used to bypass the lane skew. Active high.
6 dp_tx_phy_enco der_bypass	Used to bypass the encoder. Used to bypass the encoder. Active high.
5 dp_tx_phy_scra mbler_bypass	Used to bypass the scrambler. Used to bypass the scrambler. Active high.
4-1 dp_tx_phy_traini ng_type	Specifies the training pattern type used as follows: 0000 PRBS7, 0001 TPS1, 0010 TPS2, 0011 TPS3, 0100 TPS4, 0101 custom 80-bit pattern, 0110 D10. Specifies the training pattern type used as follows: 0000 PRBS7, 0001 TPS1, 0010 TPS2, 0011 TPS3, 0100 TPS4, 0101 custom 80-bit pattern, 0110 D10.2 training pattern, 0111 Symbol Error Rate Measurement pattern, 1000 CP2520 pattern1, 1001 CP2520 pattern2, 1010 CP2520 pattern3, others reserved
0 dp_tx_phy_traini ng_enable	Enables the training sequence (when set to 1). Enables the training sequence (when set to 1).

13.4.10.1.97 DPTC PHY software reset (DP_TX_PHY_SW_RESET)

13.4.10.1.97.1 Offset

Register	Offset
DP_TX_PHY_SW_RES ET	2004h

13.4.10.1.97.2 Diagram



13.4.10.1.97.3 Fields

Field	Function
31-1 reserved_0	
0 dp_tx_phy_sw_rst	Software reset. Software reset. Active high.

13.4.10.1.98 Scrambler seed (DP_TX_PHY_SCRAMBLER_SEED)

13.4.10.1.98.1 Offset

Register	Offset
DP_TX_PHY_SCRAMBLER_SEED	2008h

13.4.10.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dp_tx_phy_scrambler_seed															
W																
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.1.98.3 Fields

Field	Function
31-16 reserved_0	
15-0 dp_tx_phy_scrambler_seed	Scrambler seed range 0-0xFFFF

13.4.10.1.99 Custom training value bytes 1-4 (DP_TX_PHY_TRAINING_01_04)

13.4.10.1.99.1 Offset

Register	Offset
DP_TX_PHY_TRAINING_01_04	200Ch

13.4.10.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dp_tx_phy_training_04								dp_tx_phy_training_03							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dp_tx_phy_training_02								dp_tx_phy_training_01							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.99.3 Fields

Field	Function
31-24 dp_tx_phy_training_04	Byte 4 of the 80-bit custom training data. Byte 4 of the 80-bit custom training data.
23-16 dp_tx_phy_training_03	Byte 3 of the 80-bit custom training data. Byte 3 of the 80-bit custom training data.
15-8 dp_tx_phy_training_02	Byte 2 of the 80-bit custom training data. Byte 2 of the 80-bit custom training data.
7-0 dp_tx_phy_training_01	Byte 1 of the 80-bit custom training data. Byte 1 of the 80-bit custom training data.

13.4.10.1.100 Custom training value bytes 5-8 (DP_TX_PHY_TRAINING_05_08)

13.4.10.1.100.1 Offset

Register	Offset
DP_TX_PHY_TRAINING_05_08	2010h

13.4.10.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	dp_tx_phy_training_08								dp_tx_phy_training_07							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	dp_tx_phy_training_06								dp_tx_phy_training_05							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.100.3 Fields

Field	Function
31-24 dp_tx_phy_training_08	Byte 8 of the 80-bit custom training data. Byte 8 of the 80-bit custom training data.
23-16 dp_tx_phy_training_07	Byte 7 of the 80-bit custom training data. Byte 7 of the 80-bit custom training data.
15-8 dp_tx_phy_training_06	Byte 6 of the 80-bit custom training data. Byte 6 of the 80-bit custom training data.
7-0 dp_tx_phy_training_05	Byte 5 of the 80-bit custom training data. Byte 5 of the 80-bit custom training data.

13.4.10.1.101 Custom training value bytes 9-10 (DP_TX_PHY_TRAINING_09_10)

13.4.10.1.101.1 Offset

Register	Offset
DP_TX_PHY_TRAINING_09_10	2014h

13.4.10.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dp_tx_phy_training_10								dp_tx_phy_training_09							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.101.3 Fields

Field	Function
31-16 reserved_0	
15-8 dp_tx_phy_training_10	Byte 10 of the 80-bit custom training data. Byte 10 of the 80-bit custom training data.
7-0 dp_tx_phy_training_09	Byte 9 of the 80-bit custom training data. Byte 9 of the 80-bit custom training data.

13.4.10.1.102 Custom CP2520 SR interval (DP_TX_PHY_SR_INTERVAL)

13.4.10.1.102.1 Offset

Register	Offset
DP_TX_PHY_SR_INTERVAL	2018h

13.4.10.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	dp_tx_phy_sr_interval															
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	1	1	1	0	0

13.4.10.1.102.3 Fields

Field	Function
31-16 reserved_0	
15-0 dp_tx_phy_sr_interval	CP2520 test pattern SR Interval definition

13.4.10.1.103 HPD min timer for irq, define the minimum pclk cycles that the HPD pulse will be considered as IRQ (HPD_IRQ_DET_MIN_TIMER)

13.4.10.1.103.1 Offset

Register	Offset
HPD_IRQ_DET_MIN_TIMER	2100h

13.4.10.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								hpd_irq_det_min_timer							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hpd_irq_det_min_timer															
W																
Reset	1	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0

13.4.10.1.103.3 Fields

Field	Function
31-24 reserved_0	
23-0 hpd_irq_det_min_timer	hpd min timer for irq

13.4.10.1.104 HPD max timer for irq, define the maximum pclk cycles that the HPD pulse will be considered as IRQ (HPD_IRQ_DET_MAX_TIMER)

13.4.10.1.104.1 Offset

Register	Offset
HPD_IRQ_DET_MAX_TIMER	2104h

13.4.10.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								hpd_irq_det_max_timer							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hpd_irq_det_max_timer															
W																
Reset	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0

13.4.10.1.104.3 Fields

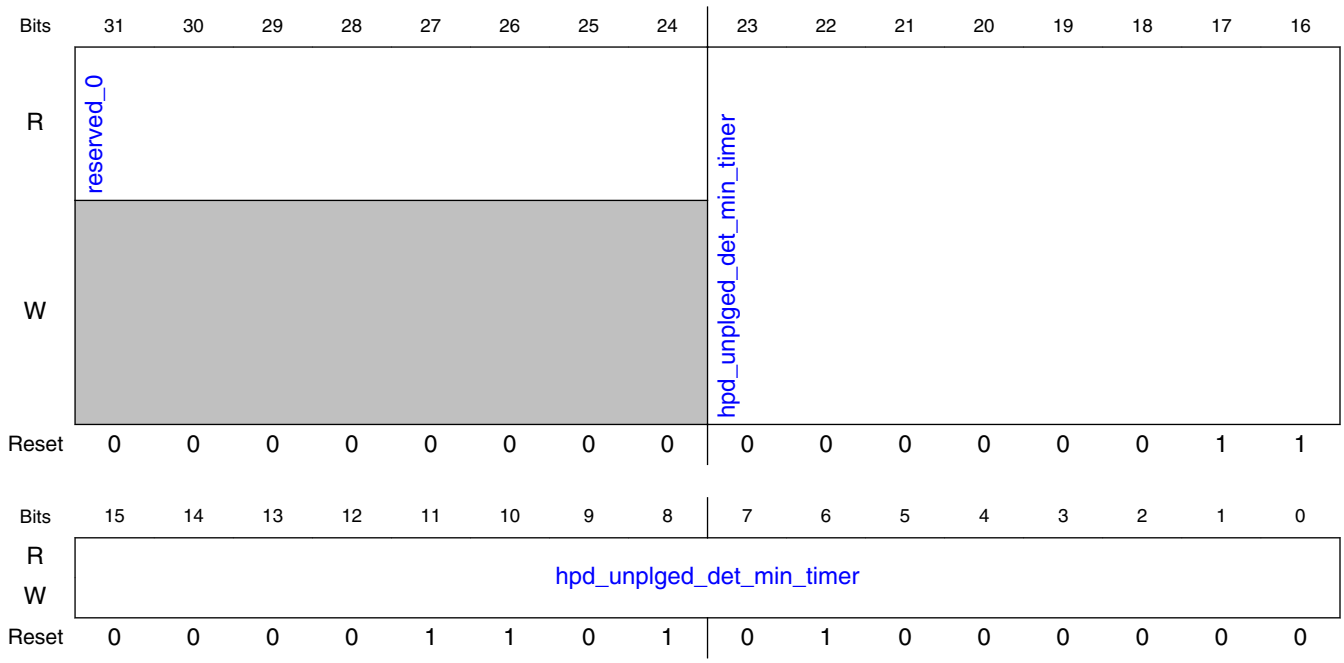
Field	Function
31-24 reserved_0	
23-0 hpd_irq_det_max_timer	hpd max timer

13.4.10.1.105 HPD min timer for HPD detect, define the minimum pclk cycles that the HPD is low (HPD_UNPLGED_DET_MIN_TIMER)

13.4.10.1.105.1 Offset

Register	Offset
HPD_UNPLGED_DET_MIN_TIMER	2108h

13.4.10.1.105.2 Diagram



13.4.10.1.105.3 Fields

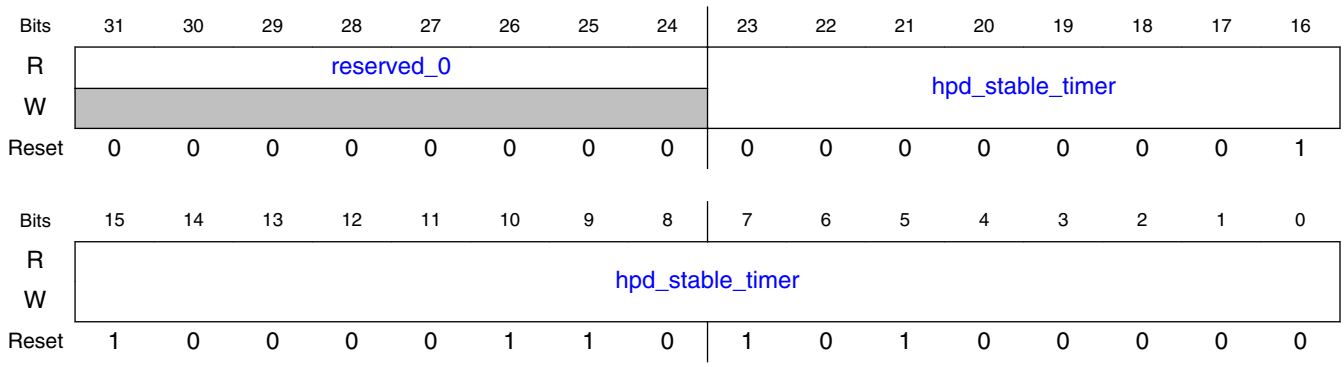
Field	Function
31-24 reserved_0	
23-0 hpd_unplged_det_min_timer	hpd unplugged timer

13.4.10.1.106 Timer for detecting HPD stable, count in pclk cycles (HPD_STABLE_TIMER)

13.4.10.1.106.1 Offset

Register	Offset
HPD_STABLE_TIMER	210Ch

13.4.10.1.106.2 Diagram



13.4.10.1.106.3 Fields

Field	Function
31-24 reserved_0	
23-0 hpd_stable_timer	hpd stable timer bits

13.4.10.1.107 Timer for filtering small pulses on hpd input (HPD_FILTER_TIMER)

13.4.10.1.107.1 Offset

Register	Offset
HPD_FILTER_TIMER	2110h

13.4.10.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								hpd_filter_timer							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hpd_filter_timer															
W																
Reset	0	1	1	1	1	0	1	0	0	0	0	1	0	0	1	0

13.4.10.1.107.3 Fields

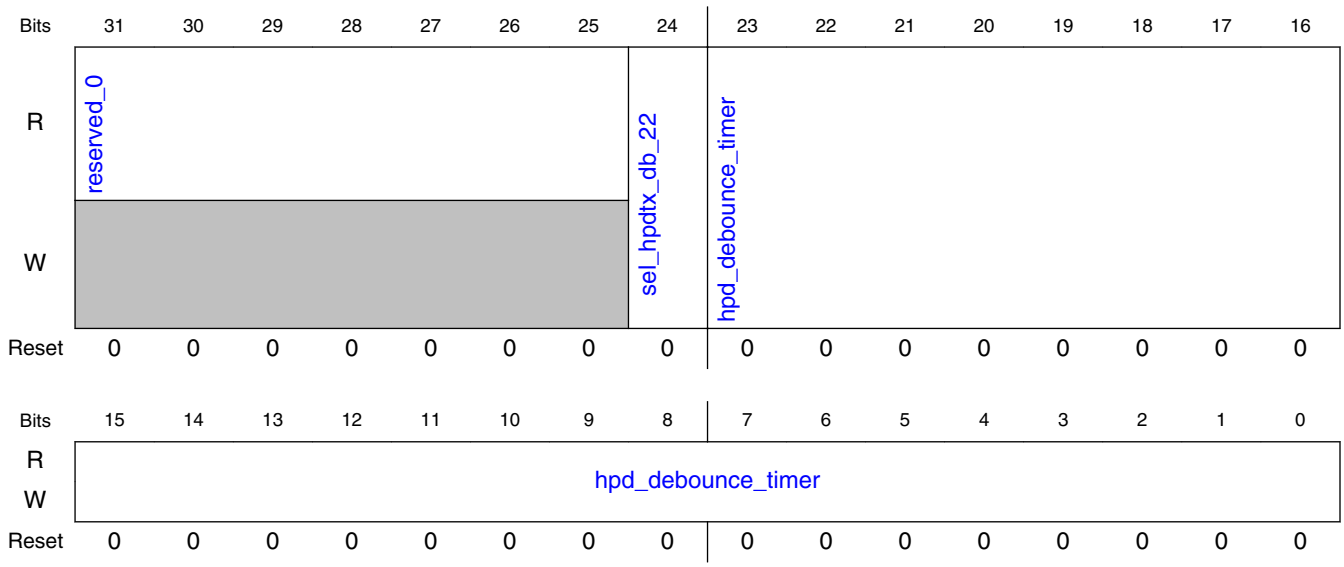
Field	Function
31-24 reserved_0	
23-0 hpd_filter_timer	hpd glitch filter timer bits

13.4.10.1.108 HPD debouncer control (HPD_DBNC_TIMER)

13.4.10.1.108.1 Offset

Register	Offset
HPD_DBNC_TIMER	2114h

13.4.10.1.108.2 Diagram



13.4.10.1.108.3 Fields

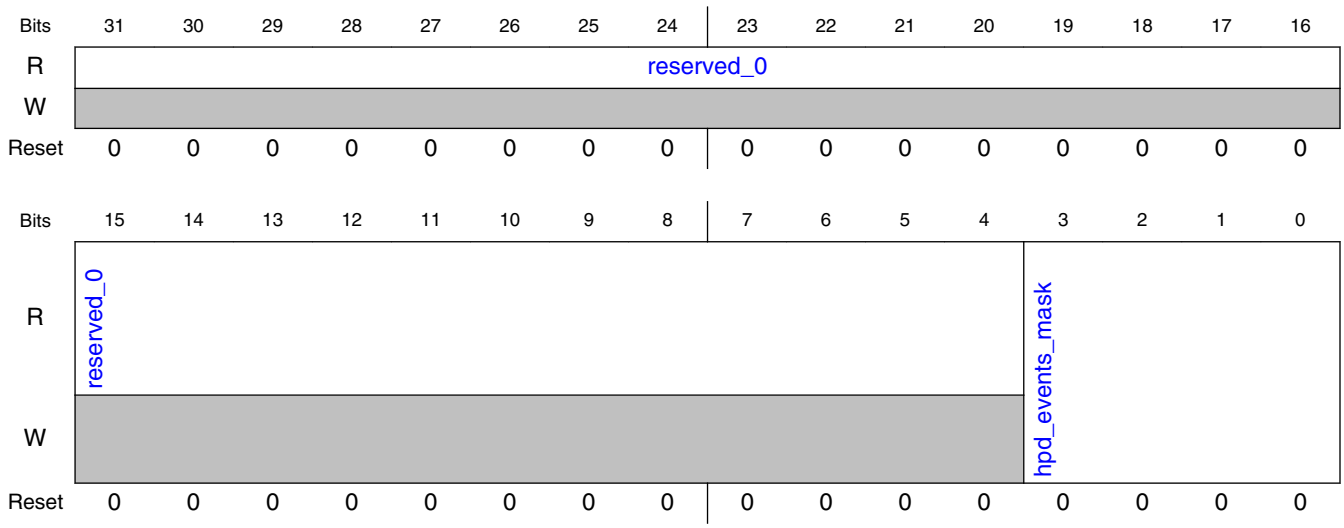
Field	Function
31-25 reserved_0	
24 sel_hpdtx_db_22	use debouncer
23-0 hpd_debounce_timer	hpd debounce timer bits

13.4.10.1.109 Mask of HPD interrupt and status (HPD_EVENT_MASK)

13.4.10.1.109.1 Offset

Register	Offset
HPD_EVENT_MASK	211Ch

13.4.10.1.109.2 Diagram



13.4.10.1.109.3 Fields

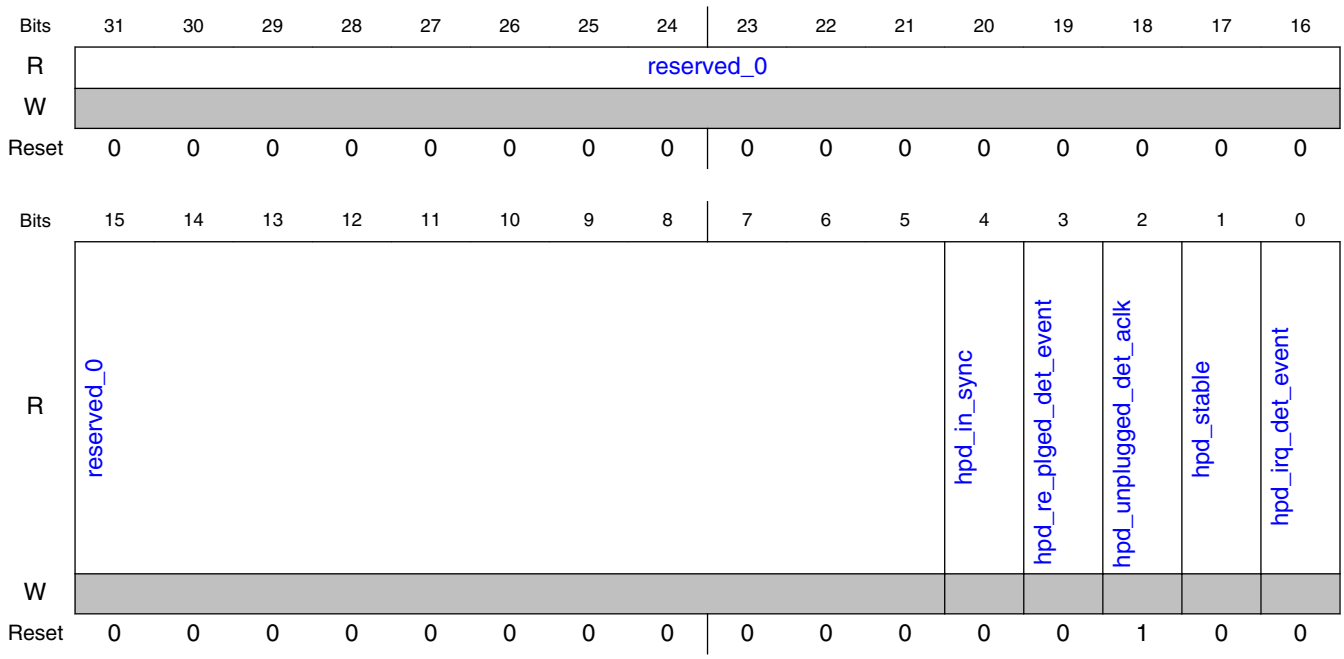
Field	Function
31-4 reserved_0	
3-0 hpd_events_mask	hpd mask events

13.4.10.1.110 HPD interrupt and status (HPD_EVENT_DET)

13.4.10.1.110.1 Offset

Register	Offset
HPD_EVENT_DET	2120h

13.4.10.1.110.2 Diagram



13.4.10.1.110.3 Fields

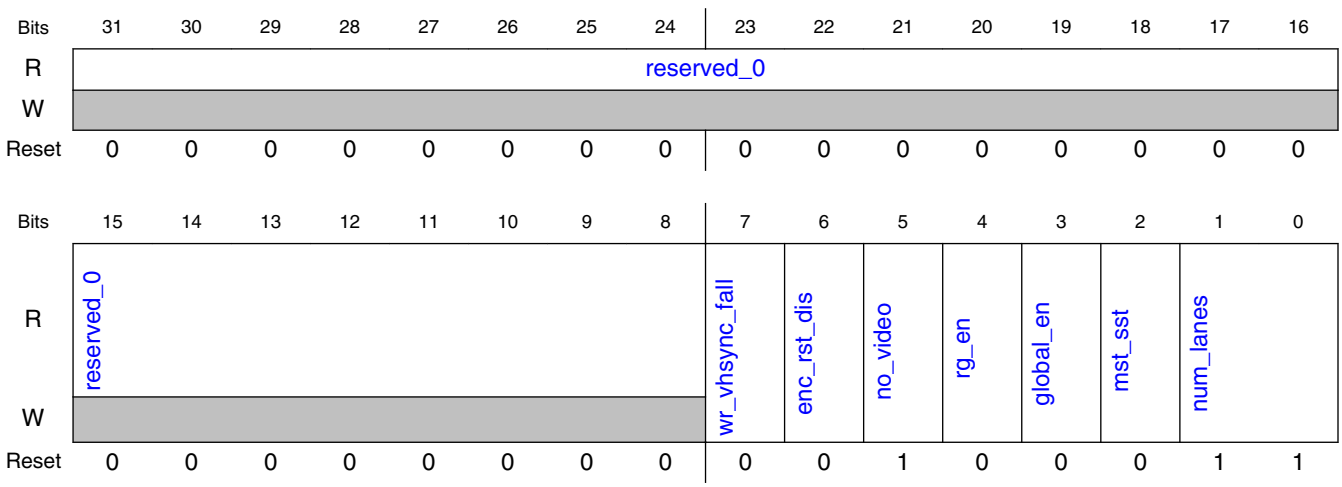
Field	Function
31-5 reserved_0	
4 hpd_in_sync	HDP in sync detected After a read operation, the field's value clears to 0.
3 hpd_re_plged_det_event	HPD Re-Plugged event detected. HPD Re-Plugged event detected. After a read operation, the field's value clears to 0.
2 hpd_unplugged_det_aclk	HPD Un-Plugged event detected. HPD Un-Plugged event detected. After a read operation, the field's value clears to 0.
1 hpd_stable	HPD Stable indication After a read operation, the field's value clears to 0.
0 hpd_irq_det_event	Bit 0 - HPD irq event After a read operation, the field's value clears to 0.

13.4.10.1.111 Global configuration of the framer module (DP_FRAMER_GLOBAL_CONFIG)

13.4.10.1.111.1 Offset

Register	Offset
DP_FRAMER_GLOBAL_CONFIG	2200h

13.4.10.1.111.2 Diagram



13.4.10.1.111.3 Fields

Field	Function
31-8 reserved_0	
7 wr_vhsync_fall	When set to 1 change the write state machine to sync on falling edge of vsync
6 enc_rst_dis	Disable reset of phy by the framer (during no video mode), 1 - reset disabled, 0 - reset enabled
5 no_video	No_video mode configuration bit, active high
4 rg_en	Rate governor enable bit, active high
3 global_en	Global enable for complete Framer module, active high. Global enable for complete Framer module, active high. It is deasserted during configuration phase. Once configuration is finished, it is asserted.

Table continues on the next page...

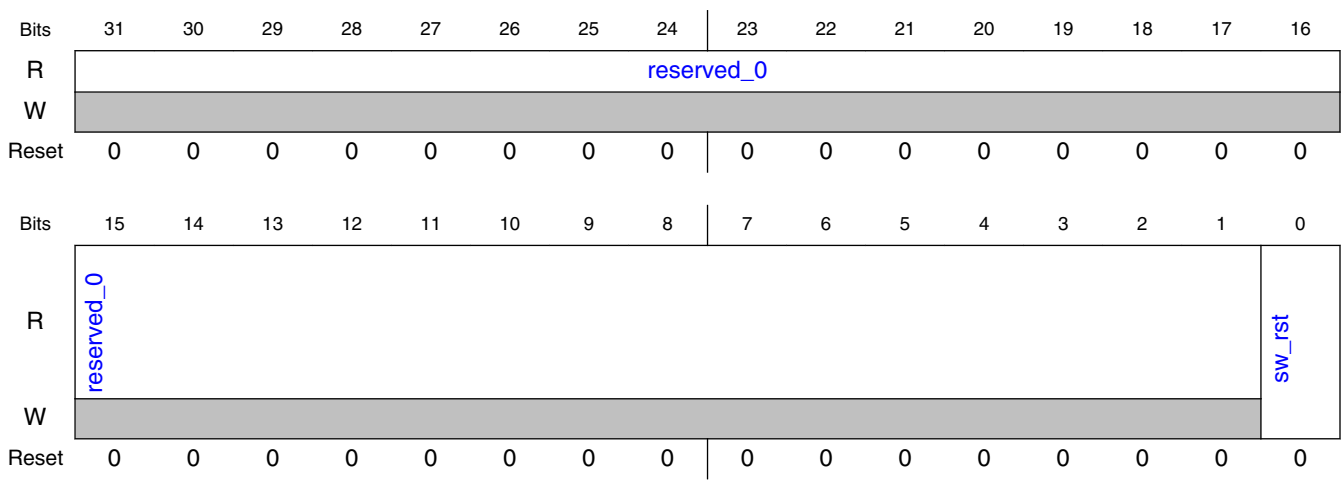
Field	Function
2 mst_sst	Mode select: 0 - SST mode 1 - MST mode
1-0 num_lanes	Number of lanes: 0h - One lane (Lane 0 only) 1h - Two lanes (Lanes 0 and 1 only) 2h - Reserved 3h - Four lanes (Lanes 0, 1, 2, and 3)

13.4.10.1.112 SW reset (DP_SW_RESET)

13.4.10.1.112.1 Offset

Register	Offset
DP_SW_RESET	2204h

13.4.10.1.112.2 Diagram



13.4.10.1.112.3 Fields

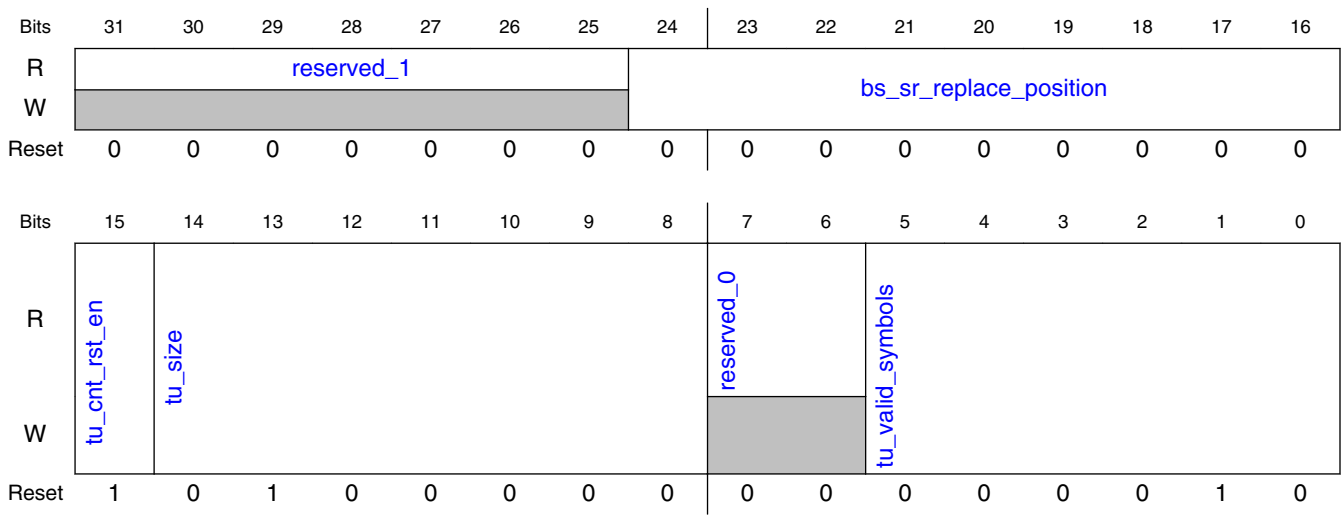
Field	Function
31-1 reserved_0	
0 sw_rst	SW reset for the entire framer

13.4.10.1.113 TU related configuration (DP_FRAMER_TU)

13.4.10.1.113.1 Offset

Register	Offset
DP_FRAMER_TU	2208h

13.4.10.1.113.2 Diagram



13.4.10.1.113.3 Fields

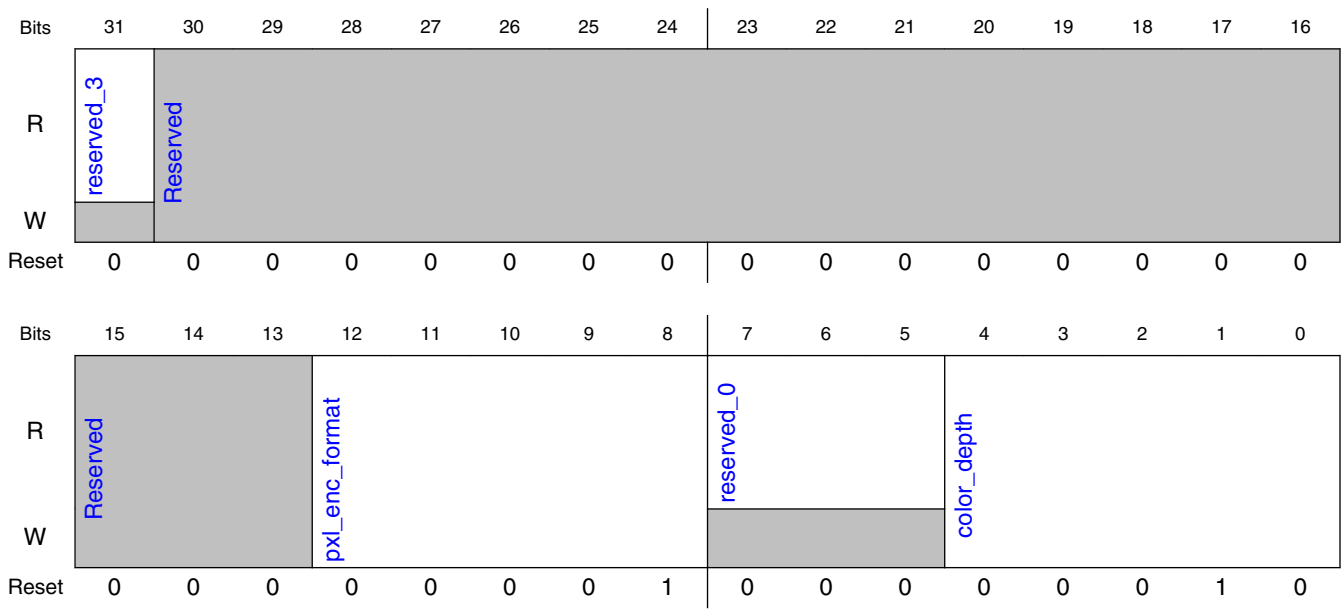
Field	Function
31-25 reserved_1	
24-16 bs_sr_replace_position	When set to non-zero value, the RTL BS counter will be reinitialized to this value. When set to non-zero value, the RTL BS counter will be reinitialized to this value.
15 tu_cnt_rst_en	When set high allows VIF logic to reset the TU_start counter in order to synchronize it to the VIF read fsm
14-8 tu_size	TU size (min=32, max=64)
7-6 reserved_0	
5-0 tu_valid_symbols	Number of valid symbols in TU, rounded to lower integer value (refer to equation in standard). Number of valid symbols in TU, rounded to lower integer value (refer to equation in standard). Always lower or equal to TU size

13.4.10.1.114 Video pixel format configuration (DP_FRAMER_PXL_REPR)

13.4.10.1.114.1 Offset

Register	Offset
DP_FRAMER_PXL_REPR	220Ch

13.4.10.1.114.2 Diagram



13.4.10.1.114.3 Fields

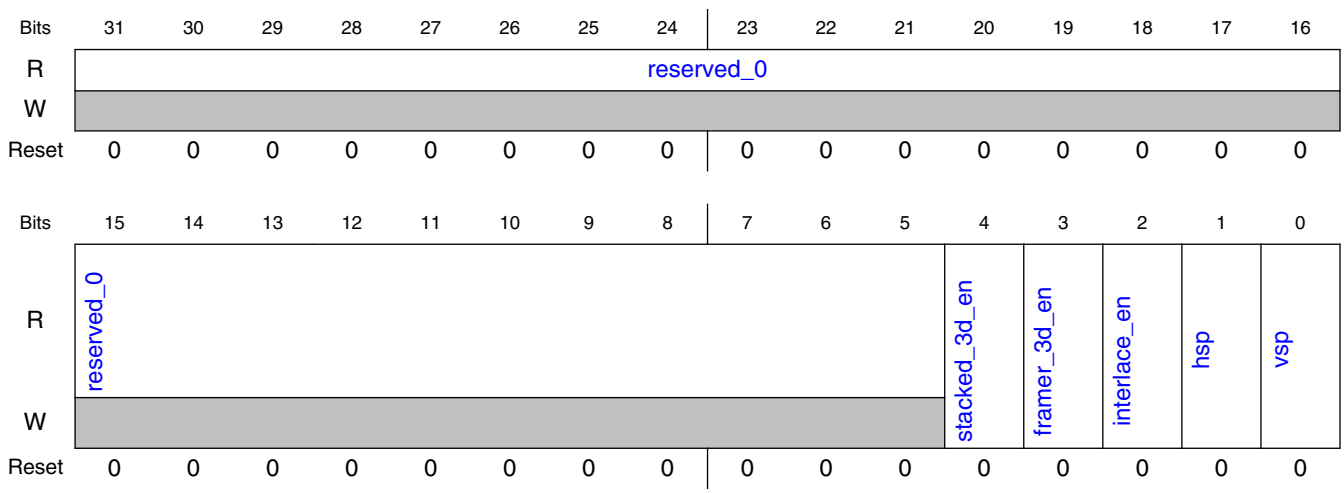
Field	Function
31 reserved_3	
30-13 —	Reserved
12-8 pxl_enc_format	Pixel encoding format: 1h - RGB2h - YCbCr 4:4:4, 4h - YCbCr 4:2:2, 8h - Y CbCr 4:2:0, 10h - Y only, All other values - RESERVED
7-5 reserved_0	
4-0 color_depth	Color depth: 1h - 6 bpc, 2h - 8 bpc, 4h - 10 bpc, 8h - 12 bpc, 10h - 16 bpc, All other values - RESERVED

13.4.10.1.115 Polarity and 3D configuration (DP_FRAMER_SP)

13.4.10.1.115.1 Offset

Register	Offset
DP_FRAMER_SP	2210h

13.4.10.1.115.2 Diagram



13.4.10.1.115.3 Fields

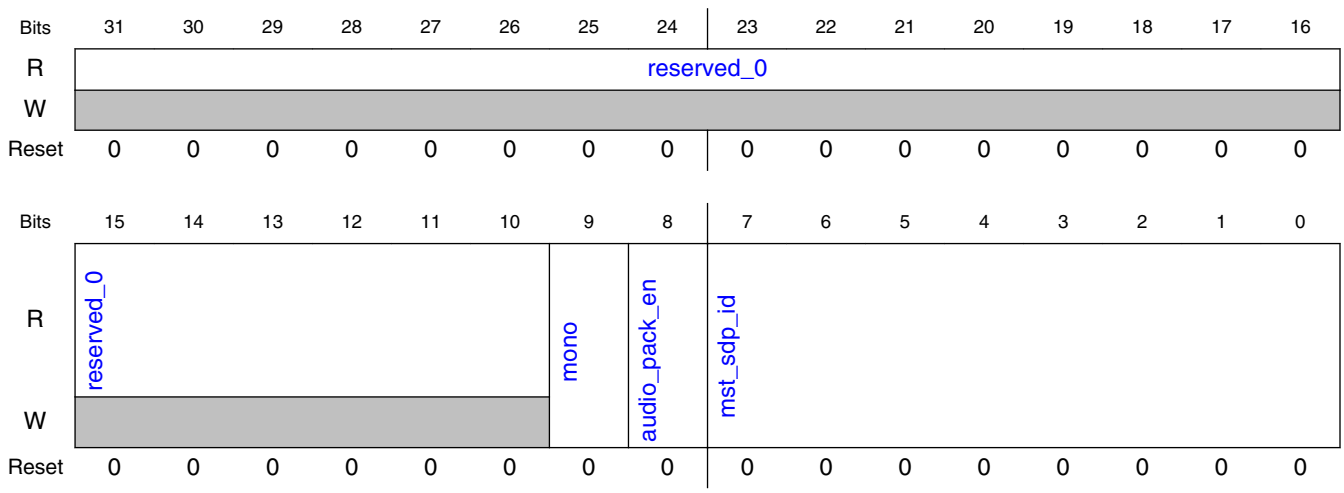
Field	Function
31-5 reserved_0	
4 stacked_3d_en	3D mode, 1 - stacked, 0 - field sequential
3 framer_3d_en	3D video enabled, active high
2 interlace_en	Interlaced video enabled, active high
1 hsp	HSYNC polarity: 0 - active high, 1 - active low
0 vsp	VSYNC polarity: 0 - active high, 1 - active low

13.4.10.1.116 Audio packet configuration (AUDIO_PACK_CONTROL)

13.4.10.1.116.1 Offset

Register	Offset
AUDIO_PACK_CONTROL	2214h

13.4.10.1.116.2 Diagram



13.4.10.1.116.3 Fields

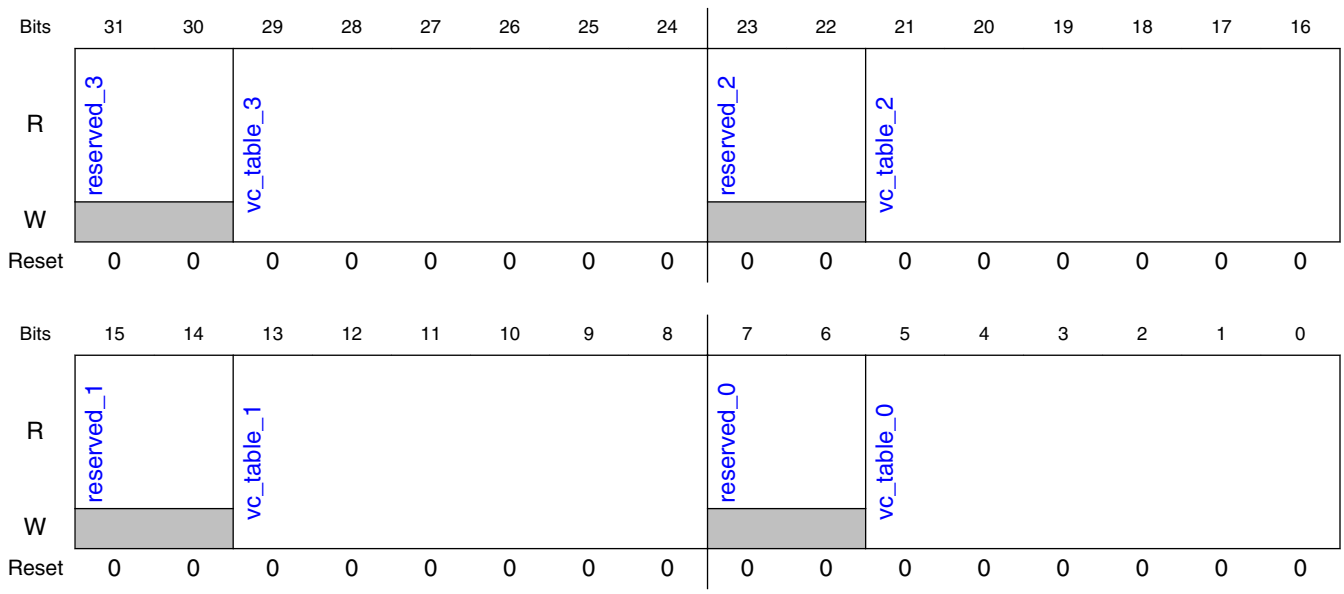
Field	Function
31-10 reserved_0	
9 mono	In case of 2-channel layout and one lane configuration SW decides whether it is a stereo or mono transfer. In case of 2-channel layout and one lane configuration SW decides whether it is a stereo or mono transfer. Relevant for SDP Header: ChannelCount field
8 audio_pack_en	Enables the Audio_Pack module, active high
7-0 mst_sdp_id	Secondary-Data Packet ID

13.4.10.1.117 VC table values 0-3 (MST feature) (DP_VC_TABLE_0)

13.4.10.1.117.1 Offset

Register	Offset
DP_VC_TABLE_0	2218h

13.4.10.1.117.2 Diagram



13.4.10.1.117.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_3	VC Table value for time slot 3
23-22 reserved_2	
21-16 vc_table_2	VC Table value for time slot 2
15-14 reserved_1	
13-8	VC Table value for time slot 1

Table continues on the next page...

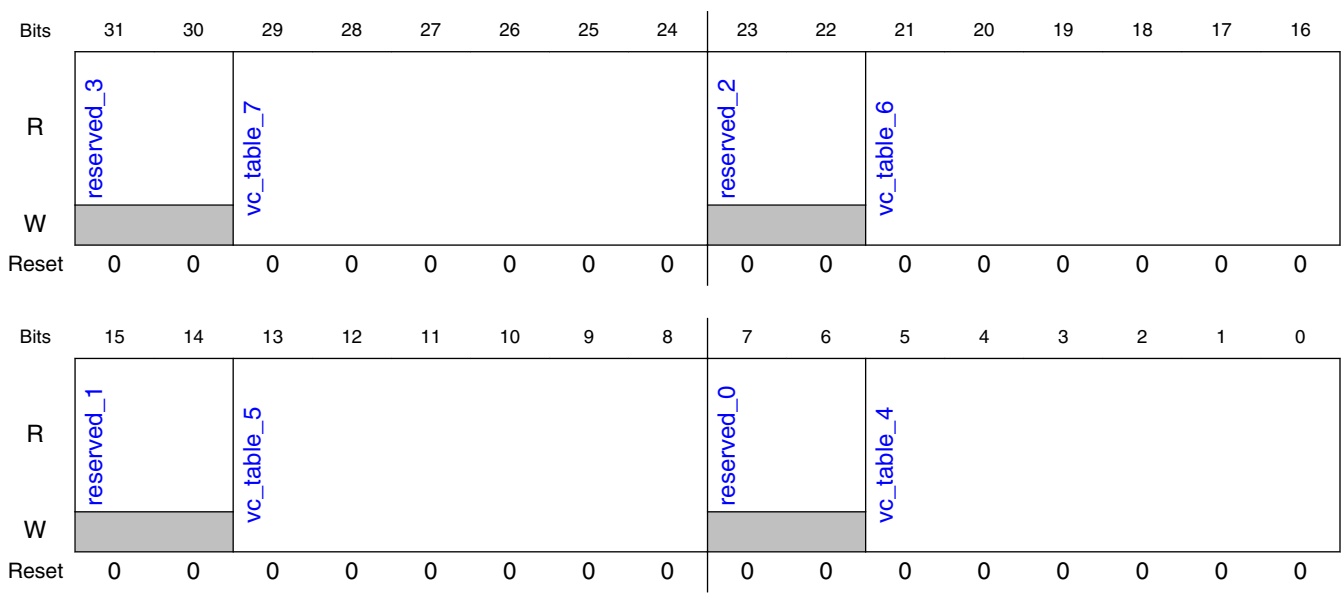
Field	Function
vc_table_1	
7-6 reserved_0	
5-0 vc_table_0	VC Table value for time slot 0

13.4.10.1.118 VC table values 4-7 (MST feature) (DP_VC_TABLE_1)

13.4.10.1.118.1 Offset

Register	Offset
DP_VC_TABLE_1	221Ch

13.4.10.1.118.2 Diagram



13.4.10.1.118.3 Fields

Field	Function
31-30 reserved_3	
29-24	VC Table value for time slot 7

Table continues on the next page...

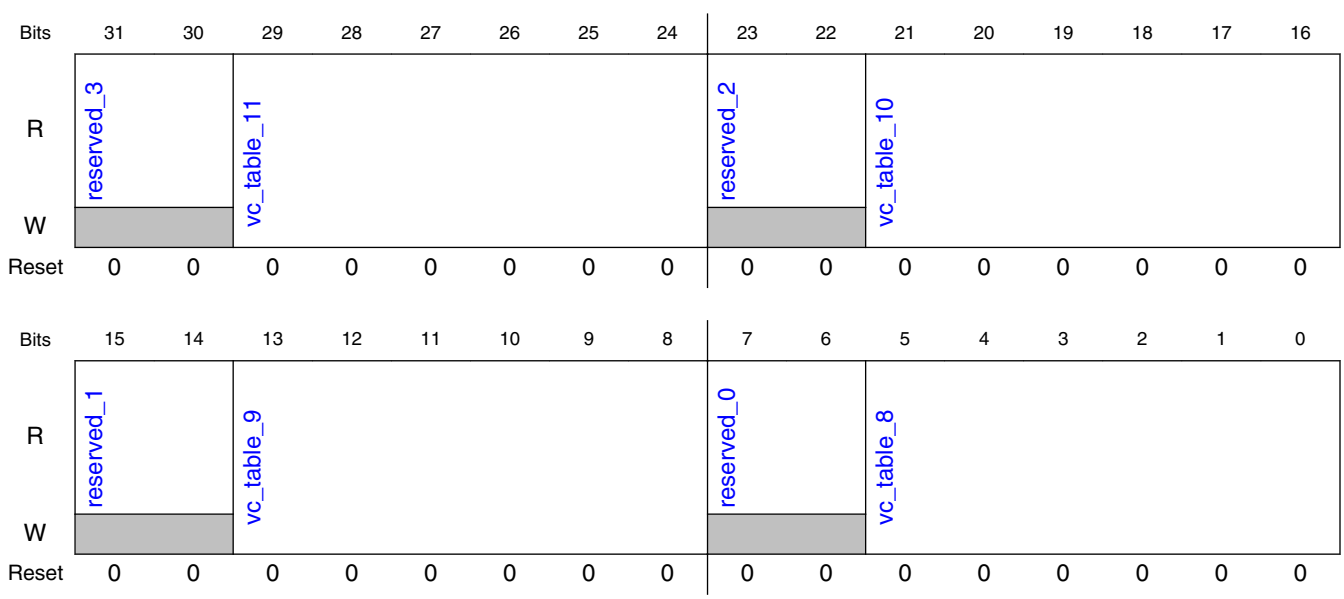
Field	Function
vc_table_7	
23-22 reserved_2	
21-16 vc_table_6	VC Table value for time slot 6
15-14 reserved_1	
13-8 vc_table_5	VC Table value for time slot 5
7-6 reserved_0	
5-0 vc_table_4	VC Table value for time slot 4

13.4.10.1.119 VC table values 8-11 (MST feature) (DP_VC_TABLE_2)

13.4.10.1.119.1 Offset

Register	Offset
DP_VC_TABLE_2	2220h

13.4.10.1.119.2 Diagram



13.4.10.1.119.3 Fields

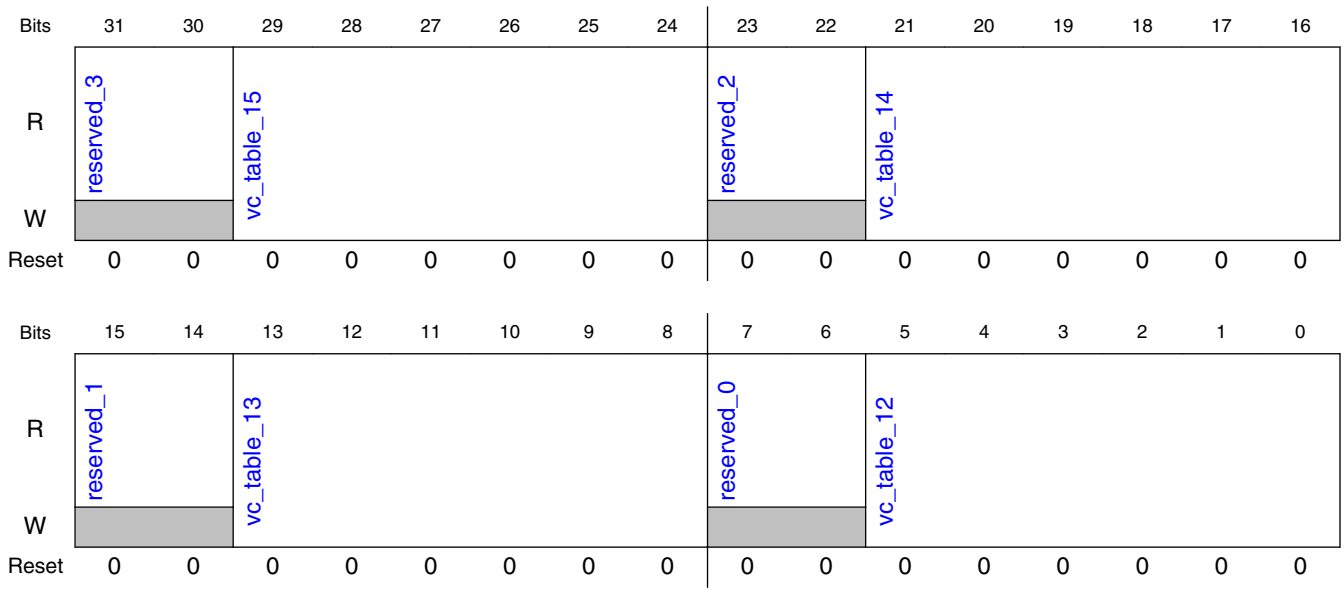
Field	Function
31-30 reserved_3	
29-24 vc_table_11	VC Table value for time slot 11
23-22 reserved_2	
21-16 vc_table_10	VC Table value for time slot 10
15-14 reserved_1	
13-8 vc_table_9	VC Table value for time slot 9
7-6 reserved_0	
5-0 vc_table_8	VC Table value for time slot 8

13.4.10.1.120 VC table values 12-15 (MST feature) (DP_VC_TABLE_3)

13.4.10.1.120.1 Offset

Register	Offset
DP_VC_TABLE_3	2224h

13.4.10.1.120.2 Diagram



13.4.10.1.120.3 Fields

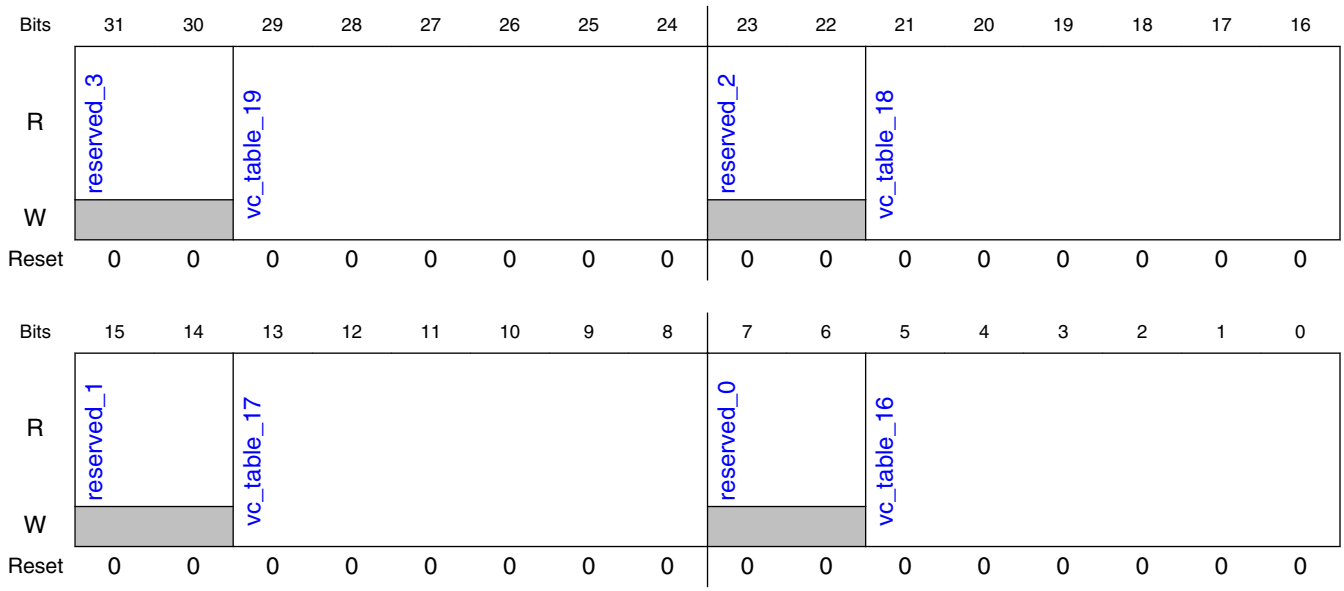
Field	Function
31-30 reserved_3	
29-24 vc_table_15	VC Table value for time slot 15
23-22 reserved_2	
21-16 vc_table_14	VC Table value for time slot 14
15-14 reserved_1	
13-8 vc_table_13	VC Table value for time slot 13
7-6 reserved_0	
5-0 vc_table_12	VC Table value for time slot 12

13.4.10.1.121 VC table values 16-19 (MST feature) (DP_VC_TABLE_4)

13.4.10.1.121.1 Offset

Register	Offset
DP_VC_TABLE_4	2228h

13.4.10.1.121.2 Diagram



13.4.10.1.121.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_19	VC Table value for time slot 19
23-22 reserved_2	
21-16 vc_table_18	VC Table value for time slot 18
15-14 reserved_1	
13-8 vc_table_17	VC Table value for time slot 17
7-6 reserved_0	

Table continues on the next page...

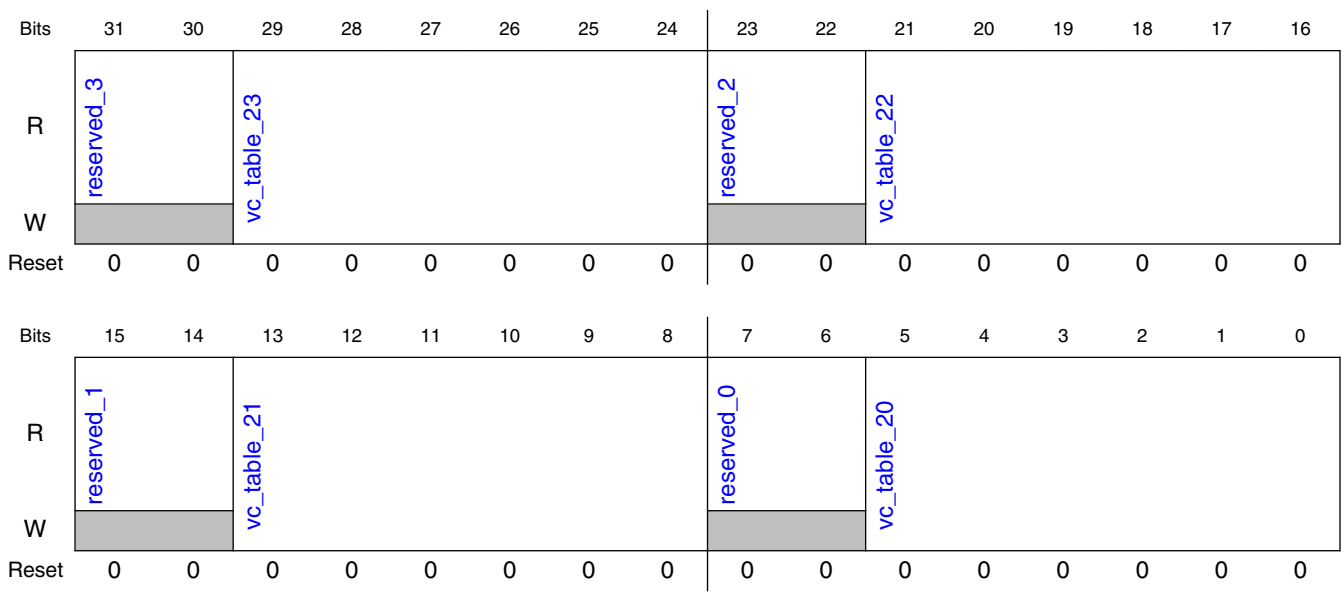
Field	Function
5-0 vc_table_16	VC Table value for time slot 16

13.4.10.1.122 VC table values 20-23 (MST feature) (DP_VC_TABLE_5)

13.4.10.1.122.1 Offset

Register	Offset
DP_VC_TABLE_5	222Ch

13.4.10.1.122.2 Diagram



13.4.10.1.122.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_23	VC Table value for time slot 23
23-22 reserved_2	

Table continues on the next page...

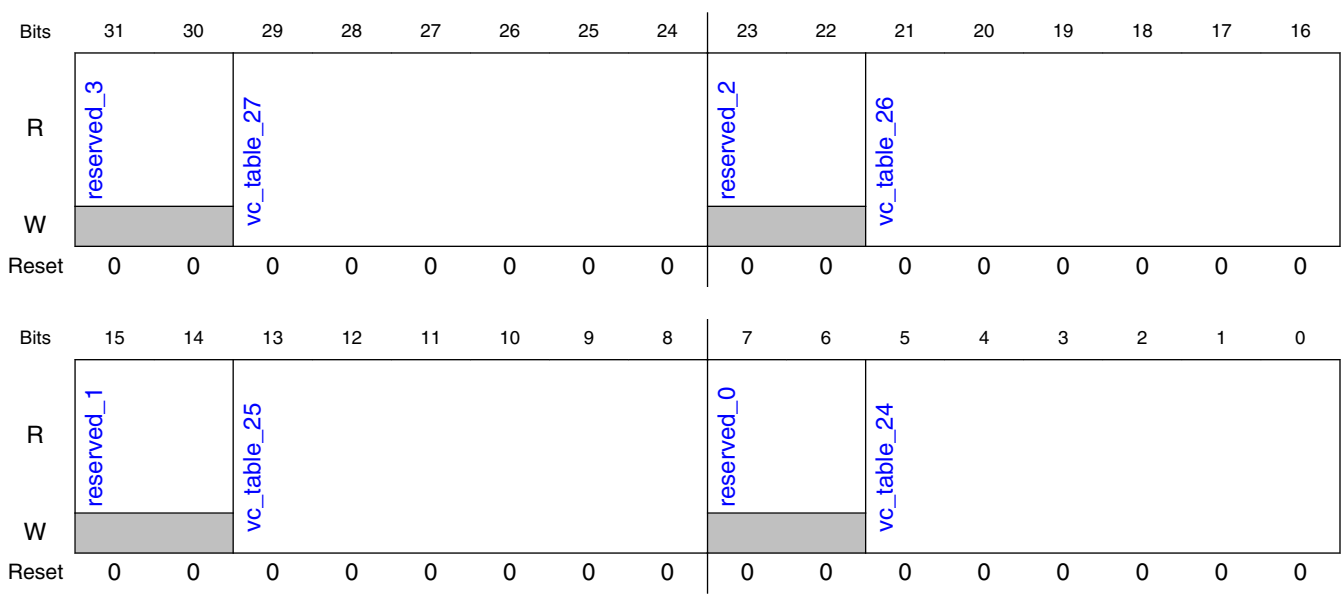
Field	Function
21-16 vc_table_22	VC Table value for time slot 22
15-14 reserved_1	
13-8 vc_table_21	VC Table value for time slot 21
7-6 reserved_0	
5-0 vc_table_20	VC Table value for time slot 20

13.4.10.1.123 VC table values 24-27 (MST feature) (DP_VC_TABLE_6)

13.4.10.1.123.1 Offset

Register	Offset
DP_VC_TABLE_6	2230h

13.4.10.1.123.2 Diagram



13.4.10.1.123.3 Fields

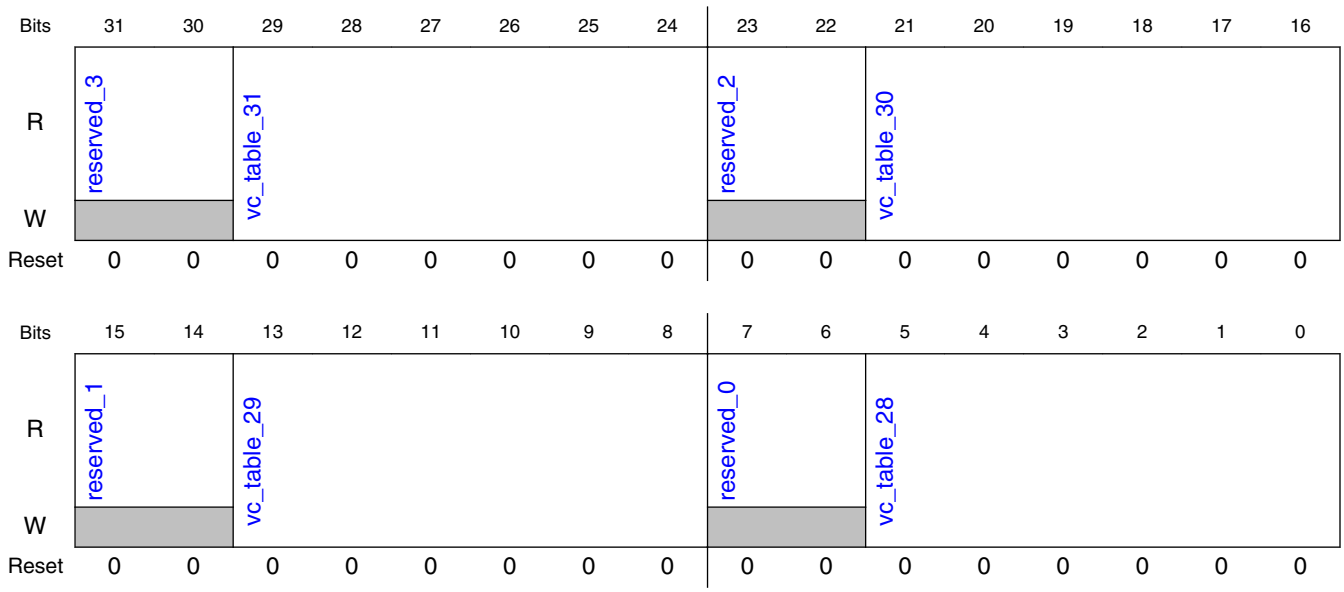
Field	Function
31-30 reserved_3	
29-24 vc_table_27	VC Table value for time slot 27
23-22 reserved_2	
21-16 vc_table_26	VC Table value for time slot 26
15-14 reserved_1	
13-8 vc_table_25	VC Table value for time slot 25
7-6 reserved_0	
5-0 vc_table_24	VC Table value for time slot 24

13.4.10.1.124 VC table values 28-31 (MST feature) (DP_VC_TABLE_7)

13.4.10.1.124.1 Offset

Register	Offset
DP_VC_TABLE_7	2234h

13.4.10.1.124.2 Diagram



13.4.10.1.124.3 Fields

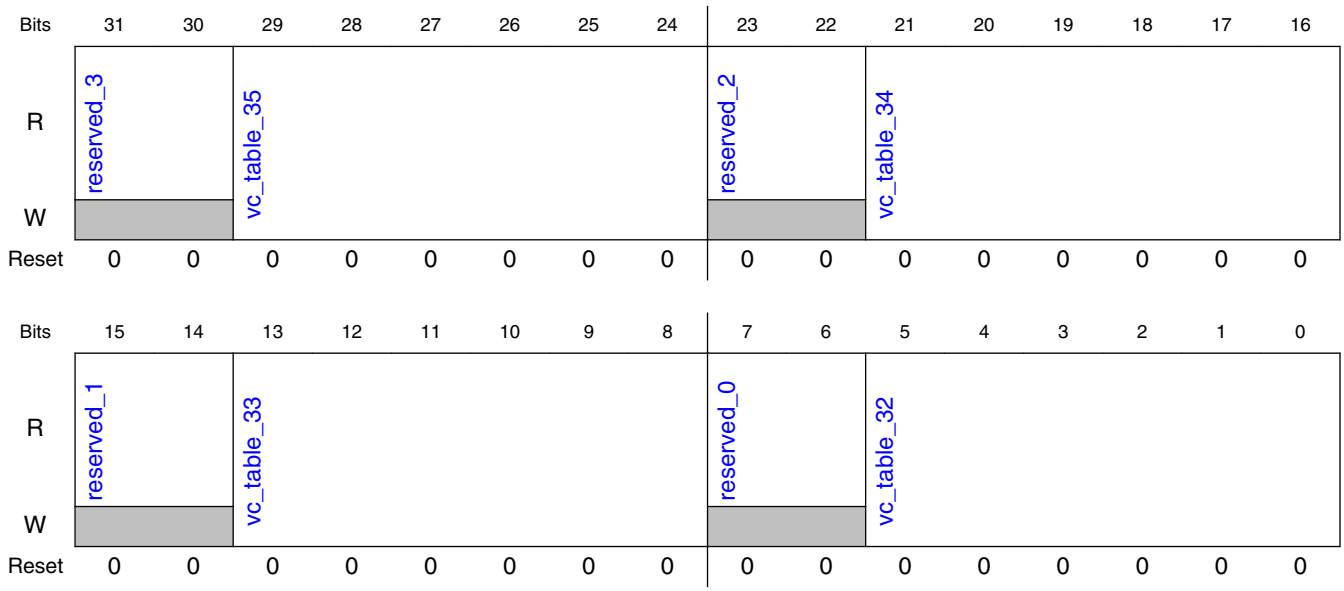
Field	Function
31-30 reserved_3	
29-24 vc_table_31	VC Table value for time slot 31
23-22 reserved_2	
21-16 vc_table_30	VC Table value for time slot 30
15-14 reserved_1	
13-8 vc_table_29	VC Table value for time slot 29
7-6 reserved_0	
5-0 vc_table_28	VC Table value for time slot 28

13.4.10.1.125 VC table values 32-35 (MST feature) (DP_VC_TABLE_8)

13.4.10.1.125.1 Offset

Register	Offset
DP_VC_TABLE_8	2238h

13.4.10.1.125.2 Diagram



13.4.10.1.125.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_35	VC Table value for time slot 35
23-22 reserved_2	
21-16 vc_table_34	VC Table value for time slot 34
15-14 reserved_1	
13-8 vc_table_33	VC Table value for time slot 33
7-6 reserved_0	

Table continues on the next page...

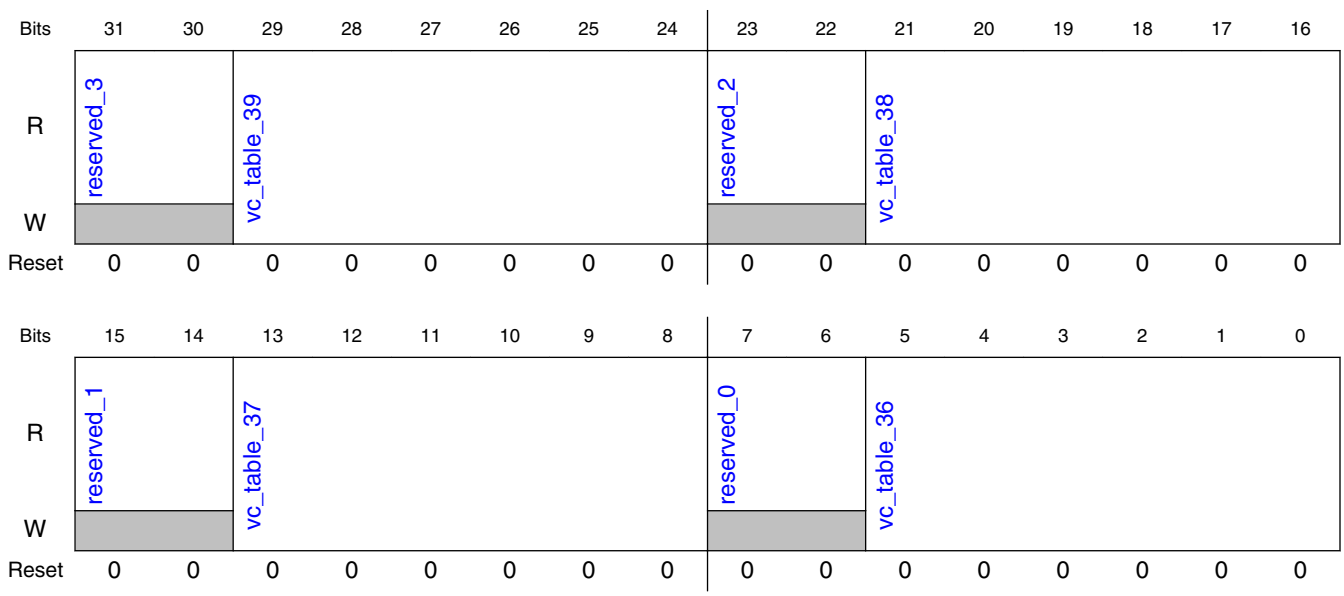
Field	Function
5-0 vc_table_32	VC Table value for time slot 32

13.4.10.1.126 VC table values 36-39 (MST feature) (DP_VC_TABLE_9)

13.4.10.1.126.1 Offset

Register	Offset
DP_VC_TABLE_9	223Ch

13.4.10.1.126.2 Diagram



13.4.10.1.126.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_39	VC Table value for time slot 39
23-22 reserved_2	

Table continues on the next page...

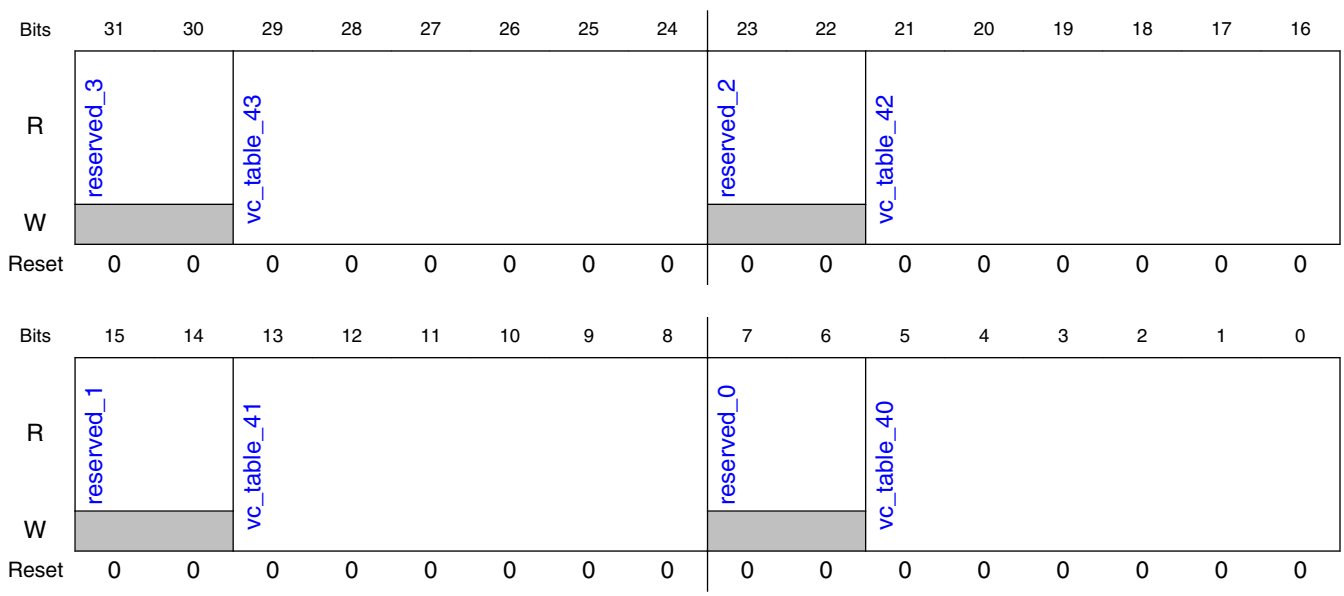
Field	Function
21-16 vc_table_38	VC Table value for time slot 38
15-14 reserved_1	
13-8 vc_table_37	VC Table value for time slot 37
7-6 reserved_0	
5-0 vc_table_36	VC Table value for time slot 36

13.4.10.1.127 VC table values 40-43 (MST feature) (DP_VC_TABLE_10)

13.4.10.1.127.1 Offset

Register	Offset
DP_VC_TABLE_10	2240h

13.4.10.1.127.2 Diagram



13.4.10.1.127.3 Fields

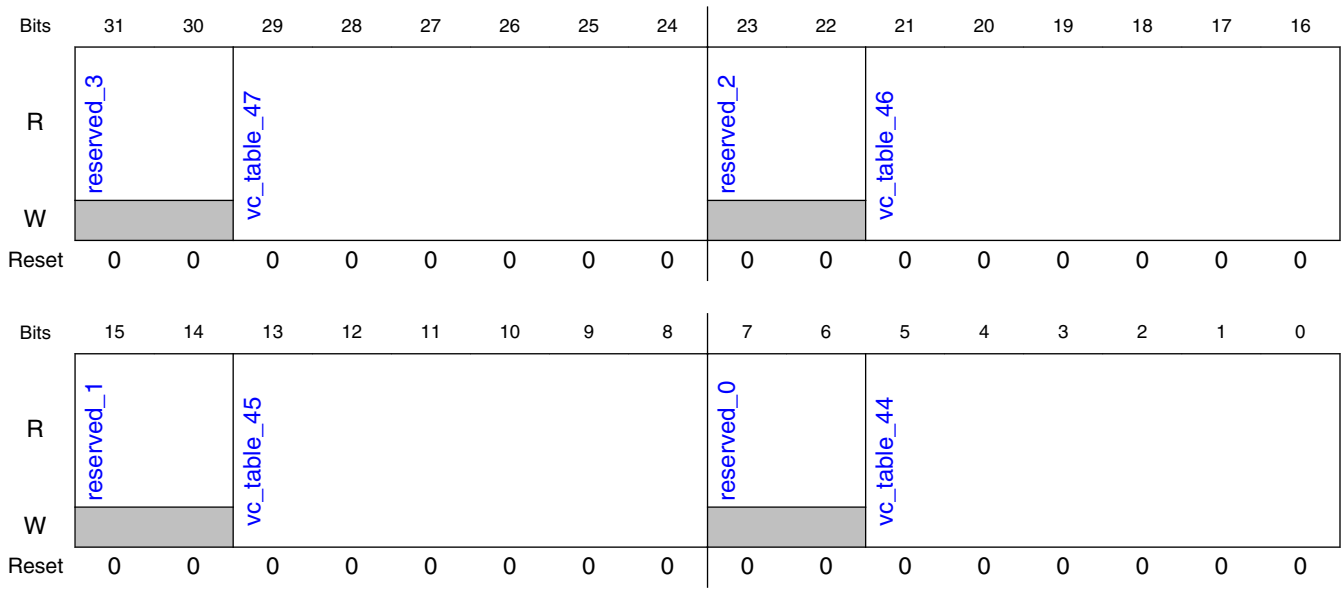
Field	Function
31-30 reserved_3	
29-24 vc_table_43	VC Table value for time slot 43
23-22 reserved_2	
21-16 vc_table_42	VC Table value for time slot 42
15-14 reserved_1	
13-8 vc_table_41	VC Table value for time slot 41
7-6 reserved_0	
5-0 vc_table_40	VC Table value for time slot 40

13.4.10.1.128 VC table values 44-47 (MST feature) (DP_VC_TABLE_11)

13.4.10.1.128.1 Offset

Register	Offset
DP_VC_TABLE_11	2244h

13.4.10.1.128.2 Diagram



13.4.10.1.128.3 Fields

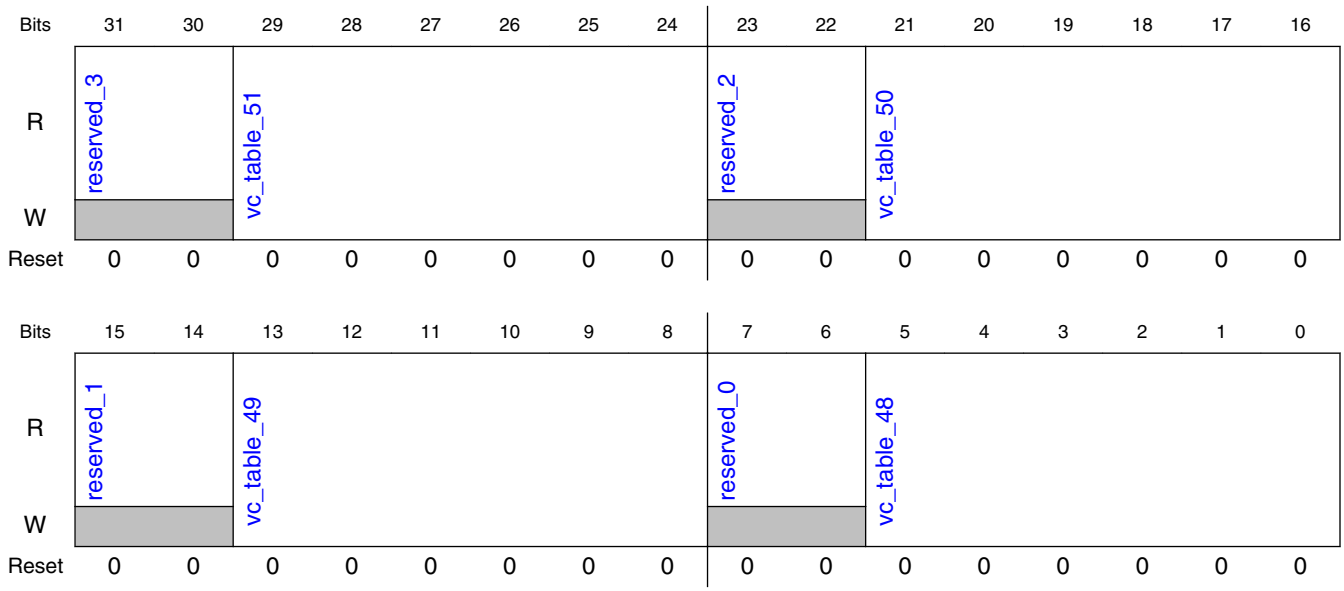
Field	Function
31-30 reserved_3	
29-24 vc_table_47	VC Table value for time slot 47
23-22 reserved_2	
21-16 vc_table_46	VC Table value for time slot 46
15-14 reserved_1	
13-8 vc_table_45	VC Table value for time slot 45
7-6 reserved_0	
5-0 vc_table_44	VC Table value for time slot 44

13.4.10.1.129 VC table values 48-51 (MST feature) (DP_VC_TABLE_12)

13.4.10.1.129.1 Offset

Register	Offset
DP_VC_TABLE_12	2248h

13.4.10.1.129.2 Diagram



13.4.10.1.129.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_51	VC Table value for time slot 51
23-22 reserved_2	
21-16 vc_table_50	VC Table value for time slot 50
15-14 reserved_1	
13-8 vc_table_49	VC Table value for time slot 49
7-6 reserved_0	

Table continues on the next page...

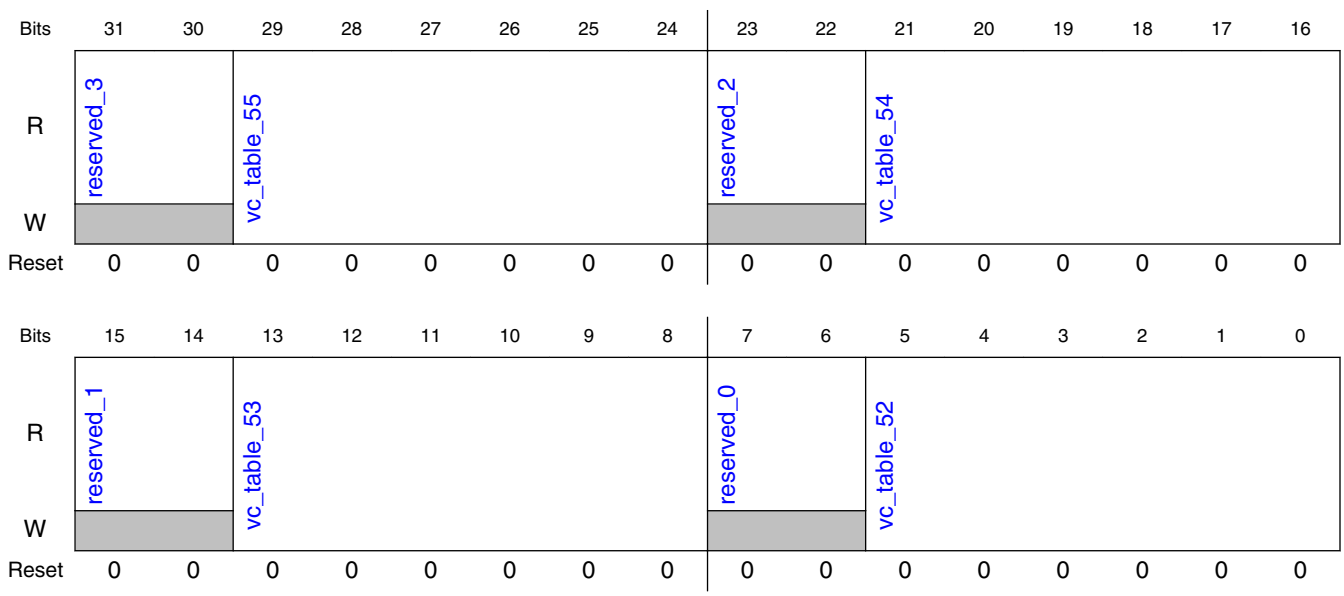
Field	Function
5-0 vc_table_48	VC Table value for time slot 48

13.4.10.1.130 VC table values 52-55 (MST feature) (DP_VC_TABLE_13)

13.4.10.1.130.1 Offset

Register	Offset
DP_VC_TABLE_13	224Ch

13.4.10.1.130.2 Diagram



13.4.10.1.130.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_55	VC Table value for time slot 55
23-22 reserved_2	

Table continues on the next page...

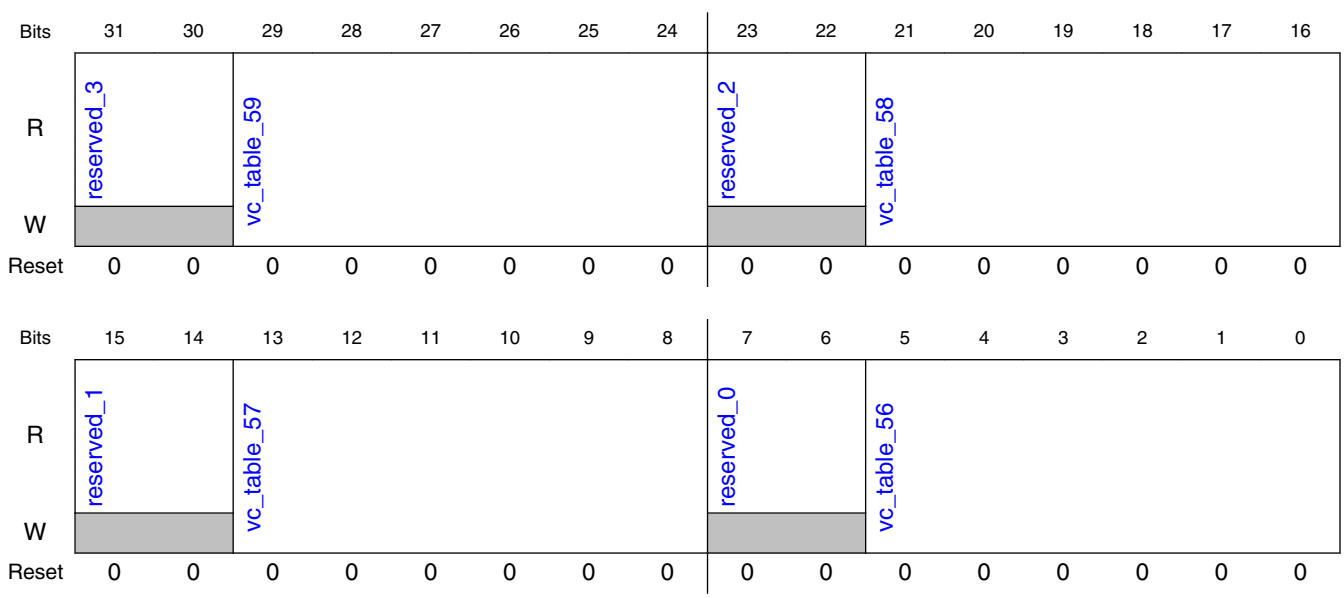
Field	Function
21-16 vc_table_54	VC Table value for time slot 54
15-14 reserved_1	
13-8 vc_table_53	VC Table value for time slot 53
7-6 reserved_0	
5-0 vc_table_52	VC Table value for time slot 52

13.4.10.1.131 VC table values 56-59 (MST feature) (DP_VC_TABLE_14)

13.4.10.1.131.1 Offset

Register	Offset
DP_VC_TABLE_14	2250h

13.4.10.1.131.2 Diagram



13.4.10.1.131.3 Fields

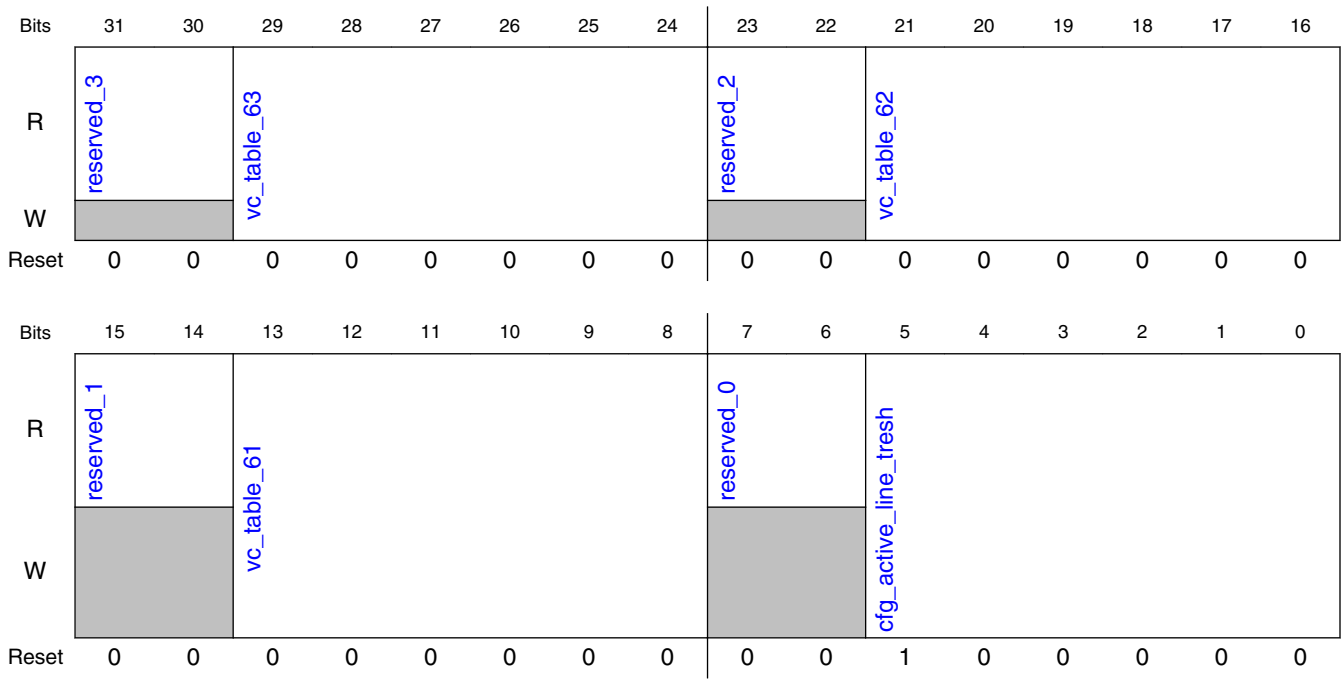
Field	Function
31-30 reserved_3	
29-24 vc_table_59	VC Table value for time slot 59
23-22 reserved_2	
21-16 vc_table_58	VC Table value for time slot 58
15-14 reserved_1	
13-8 vc_table_57	VC Table value for time slot 57
7-6 reserved_0	
5-0 vc_table_56	VC Table value for time slot 56

13.4.10.1.132 Video FIFO latency threshold (LINE_THRESH)

13.4.10.1.132.1 Offset

Register	Offset
LINE_THRESH	2254h

13.4.10.1.132.2 Diagram



13.4.10.1.132.3 Fields

Field	Function
31-30 reserved_3	
29-24 vc_table_63	VC Table value for time slot 63
23-22 reserved_2	
21-16 vc_table_62	VC Table value for time slot 62
15-14 reserved_1	
13-8 vc_table_61	VC Table value for time slot 61
7-6 reserved_0	
5-0 cfg_active_line_tresh	Video Fifo Latency threshold. Video Fifo Latency threshold. Defines the number of FIFO rows before reading starts.

13.4.10.1.133 Vertical blanking ID (DP_VB_ID)

13.4.10.1.133.1 Offset

Register	Offset
DP_VB_ID	2258h

13.4.10.1.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								vb_id							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

13.4.10.1.133.3 Fields

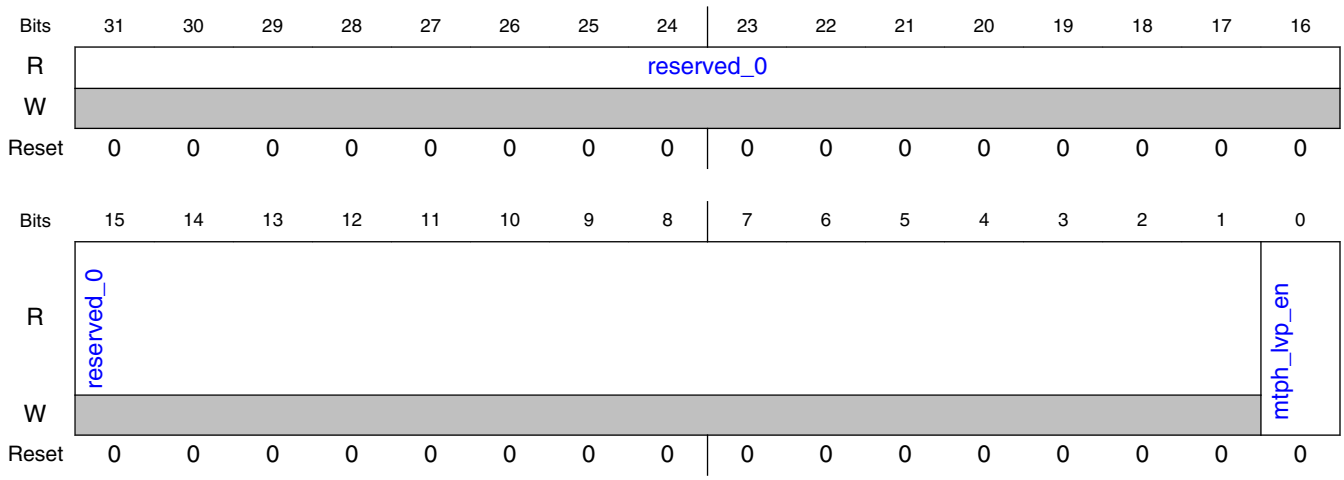
Field	Function
31-8 reserved_0	
7-0 vb_id	Vertical blanking ID

13.4.10.1.134 LVP insertion in MTP header enable (MST feature) (DP_MTPH_LVP_CONTROL)

13.4.10.1.134.1 Offset

Register	Offset
DP_MTPH_LVP_CONTROL	225Ch

13.4.10.1.134.2 Diagram



13.4.10.1.134.3 Fields

Field	Function
31-1 reserved_0	
0 mtph_lvp_en	This bit is used in MST mode. This bit is used in MST mode. When set high insertion of LVP symbols in MTP Header time slots is enabled.

13.4.10.1.135 Values to be inserted in the MTP header (MST feature) (DP_MTPH_SYMBOL_VALUES)

13.4.10.1.135.1 Offset

Register	Offset
DP_MTPH_SYMBOL_VALUES	2260h

13.4.10.1.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								mtph_mtph_sym							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	mtph_ecf_sym								mpth_lvp_sym							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.135.3 Fields

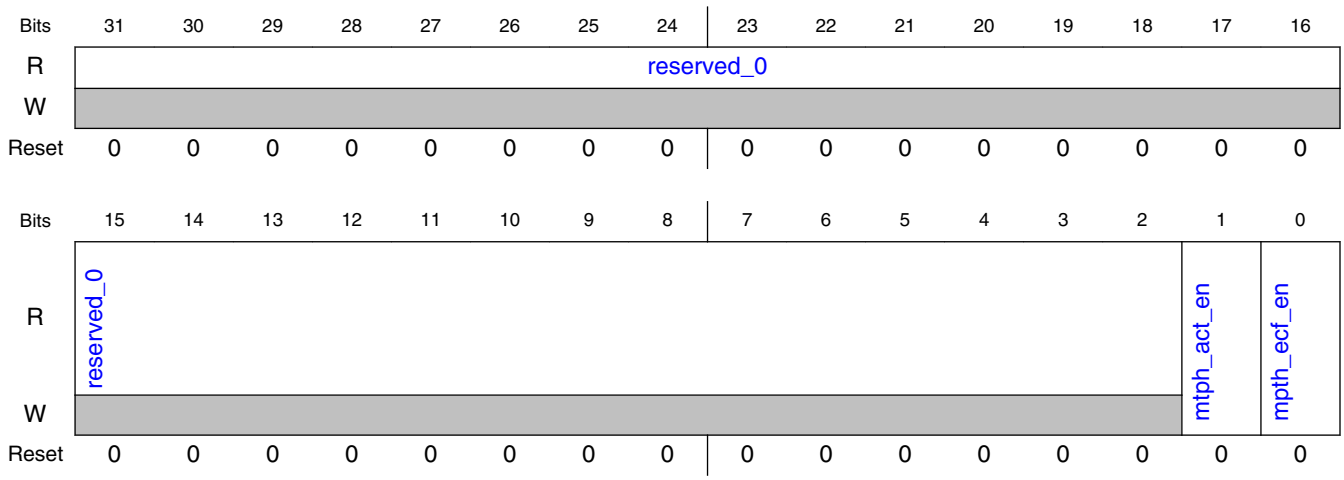
Field	Function
31-24 reserved_0	
23-16 mtph_mtph_sym	Value of this vector is inserted in MTP Header time slot when neither SR, ACT, LVP or ECF symbols are inserted in MTP Header time slot. Value of this vector is inserted in MTP Header time slot when neither SR, ACT, LVP or ECF symbols are inserted in MTP Header time slot.
15-8 mtph_ecf_sym	Value of this vector is inserted in MTP Header time slot when mtph_ecf_en signal is active
7-0 mpth_lvp_sym	Value of this vector is inserted in MTP Header time slot when mtph_lvp_en signal is active

13.4.10.1.136 ECF insertion in MTP header enable (MST feature) (DP_MTPH_ECF_CONTROL)

13.4.10.1.136.1 Offset

Register	Offset
DP_MTPH_ECF_CONTROL	2264h

13.4.10.1.136.2 Diagram



13.4.10.1.136.3 Fields

Field	Function
31-2 reserved_0	
1 mtph_act_en	MST feature
0 mpth_ecf_en	This bit is used in MST mode. This bit is used in MST mode. When set high insertion of ECF symbols in MTP Header time slots is enabled.

13.4.10.1.137 Supporting configuration to switch from top/bottom on the input to field sequential on the output (DP_FIELDSEQ_3D)

13.4.10.1.137.1 Offset

Register	Offset
DP_FIELDSEQ_3D	2268h

13.4.10.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	field_seq_end															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	field_seq_start															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.137.3 Fields

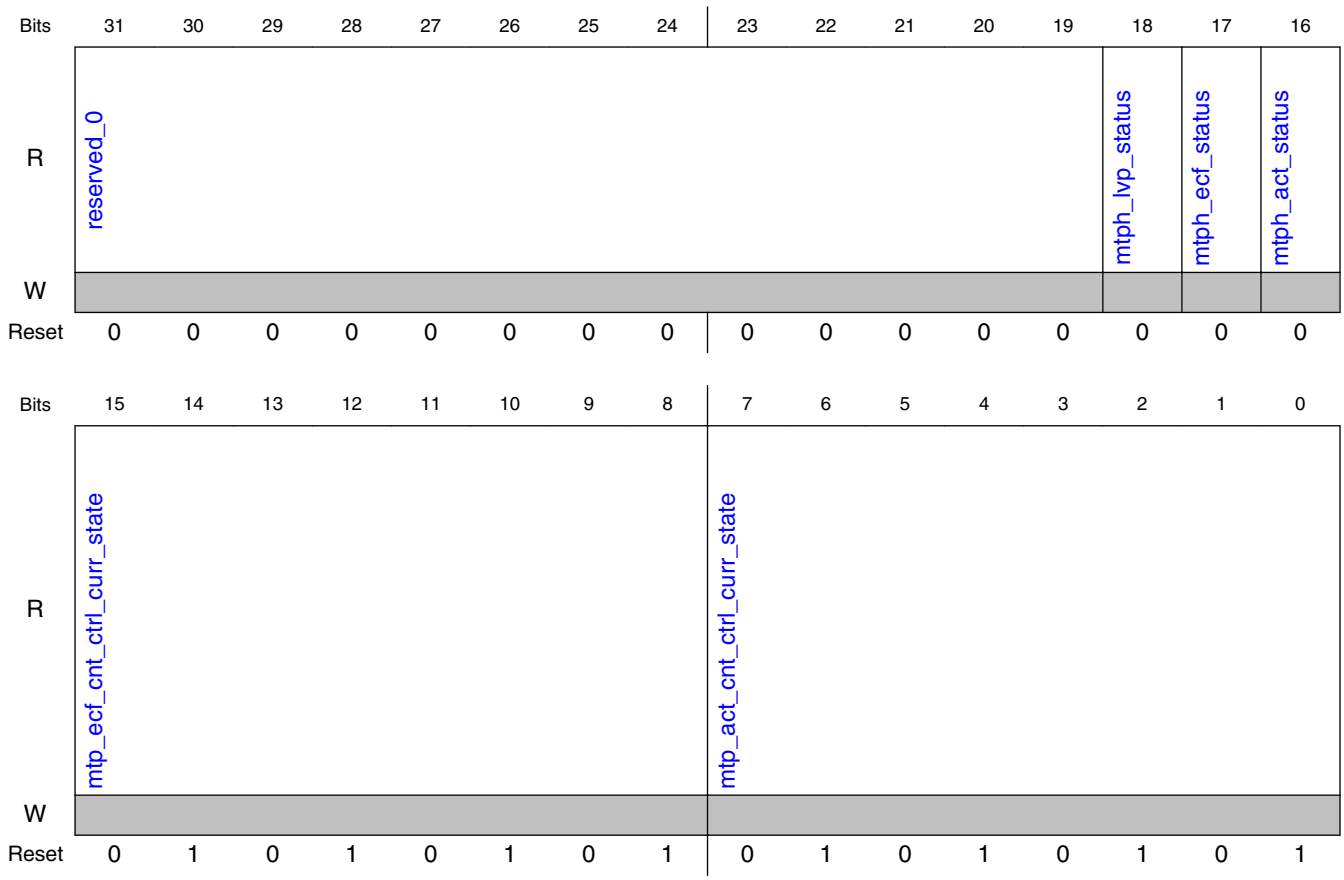
Field	Function
31-16 field_seq_end	Number of line in the frame where the Vblank part in the field sequential format ends
15-0 field_seq_start	Number of line in the frame where the Vblank part in the field sequential format starts

13.4.10.1.138 MTP header status (MST feature) (DP_MTPH_STATUS)

13.4.10.1.138.1 Offset

Register	Offset
DP_MTPH_STATUS	226Ch

13.4.10.1.138.2 Diagram



13.4.10.1.138.3 Fields

Field	Function
31-19 reserved_0	
18 mtph_lvp_status	Status of mtph_lvp
17 mtph_ecf_status	Status of mtph_ecf
16 mtph_act_status	Status of mtph_act
15-8 mtp_ecf_cnt_ctrl_curr_state	mtp_ecf fsm current state
7-0 mtp_act_cnt_ctrl_curr_state	mtp_act fsm current state

13.4.10.1.139 Interrupt sources of the framer module, active high. (DP_INTERRUPT_SOURCE)

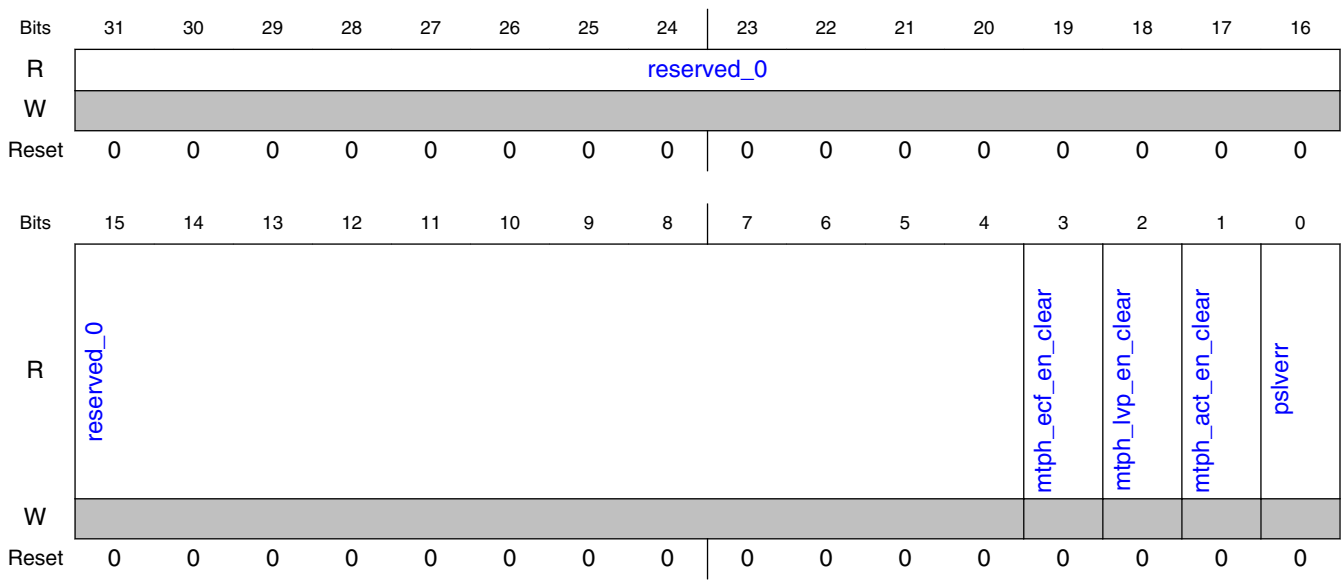
13.4.10.1.139.1 Offset

Register	Offset
DP_INTERRUPT_SOURCE	2270h

13.4.10.1.139.2 Function

Interrupt sources of the framer module, active high. Automatically cleared on read.

13.4.10.1.139.3 Diagram



13.4.10.1.139.4 Fields

Field	Function
31-4 reserved_0	
3 mtph_ecf_en_clear	Indication to clear mtph_ecf_en field in configuration register, active high

Table continues on the next page...

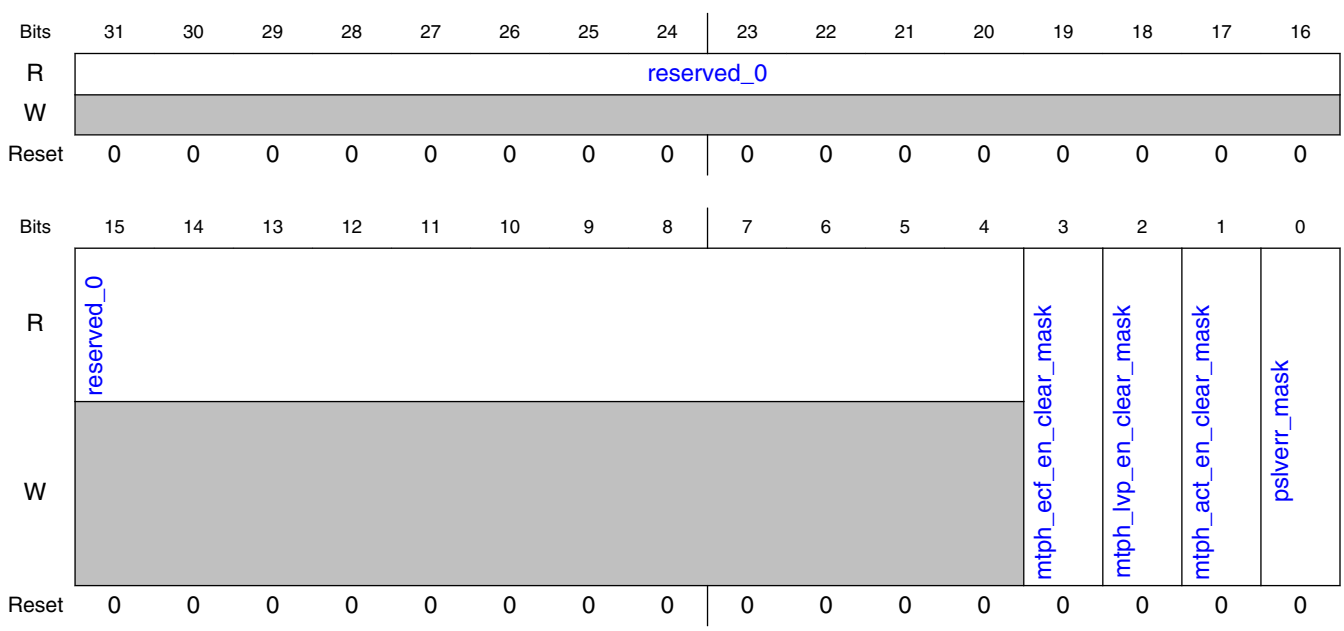
Field	Function
2 mtph_lvp_en_clear	Indication to clear mtph_lvp_en field in configuration register, active high
1 mtph_act_en_clear	Indication to clear mtph_act_en field in configuration register, active high
0 pslverr	APB slave error

13.4.10.1.140 Masks for the interrupt sources in the DP_INTERRUPT_SOURCE register, when set high, these bits disable the corresponding interrupts (DP_INTERRUPT_MASK)

13.4.10.1.140.1 Offset

Register	Offset
DP_INTERRUPT_MASK	2274h

13.4.10.1.140.2 Diagram



13.4.10.1.140.3 Fields

Field	Function
31-4 reserved_0	
3 mtph_ecf_en_clear_mask	Bit that masks mtph_ecf_en_clear interrupt
2 mtph_lvp_en_clear_mask	Bit that masks mtph_lvp_en_clear interrupt
1 mtph_act_en_clear_mask	Bit that masks mtph_act_en_clear interrupt
0 pslverr_mask	Bit that masks pslverr interrupt

13.4.10.1.141 MSA parameters to be used in the HW (DP_FRONT_BACK_PORCH)

13.4.10.1.141.1 Offset

Register	Offset
DP_FRONT_BACK_PORCH	2278h

13.4.10.1.141.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	front_porch															
Reset	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	back_porch															
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0

13.4.10.1.141.3 Fields

Field	Function
31-16 front_porch	Value of the front porch
15-0 back_porch	Value of the back porch

13.4.10.1.142 Number of bytes per lane/chunk parameters (DP_BYTE_COUNT)

13.4.10.1.142.1 Offset

Register	Offset
DP_BYTE_COUNT	227Ch

13.4.10.1.142.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	byte_count															
W																
Reset	0	0	0	0	0	0	1	0	1	0	1	0	0	1	0	0

13.4.10.1.142.3 Fields

Field	Function
31-16 reserved_0	
15-0 byte_count	Total number of bytes in a line in case of non-DSC video. Total number of bytes in a line in case of non-DSC video. When DSC is enabled should be total number of bytes in a line *per lane*, including the additional EOC symbol(s).

13.4.10.1.143 MSA horizontal parameters first part (MSA_HORIZONTAL_0)

13.4.10.1.143.1 Offset

Register	Offset
MSA_HORIZONTAL_0	2280h

13.4.10.1.143.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	pck_stuff_hstart															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	pck_stuff_htotal															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.143.3 Fields

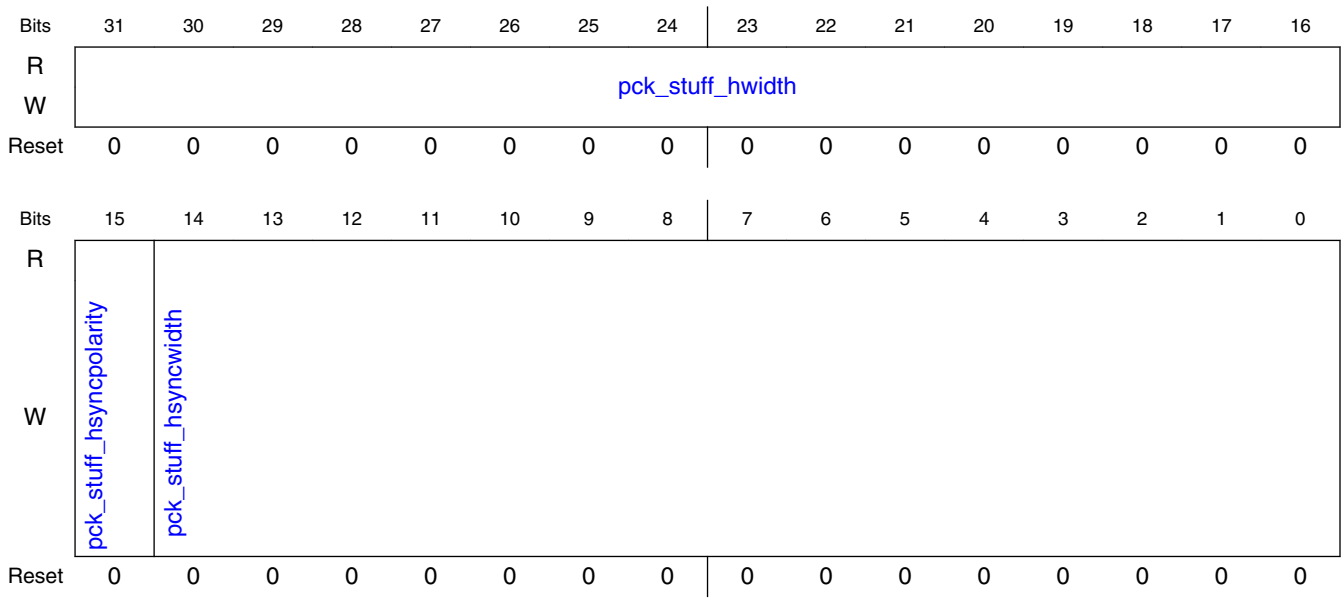
Field	Function
31-16 pck_stuff_hstart	Horizontal Active Start, index of pixel in a line where active region begins
15-0 pck_stuff_htotal	Horizontal Total, number of pixels per line

13.4.10.1.144 MSA horizontal parameters second part (MSA_HORIZONTAL_1)

13.4.10.1.144.1 Offset

Register	Offset
MSA_HORIZONTAL_1	2284h

13.4.10.1.144.2 Diagram



13.4.10.1.144.3 Fields

Field	Function
31-16 pck_stuff_hwidth	Horizontal Active Video Width, number of active pixels per line
15 pck_stuff_hsync polarity	Horizontal Sync Polarity: 0 - active high, 1 - active low
14-0 pck_stuff_hsync width	Horizontal Sync Width, number of pixels Hsync is active

13.4.10.1.145 MSA vertical parameters first part (MSA_VERTICAL_0)

13.4.10.1.145.1 Offset

Register	Offset
MSA_VERTICAL_0	2288h

13.4.10.1.145.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	pck_stuff_vstart															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	pck_stuff_vtotal															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.145.3 Fields

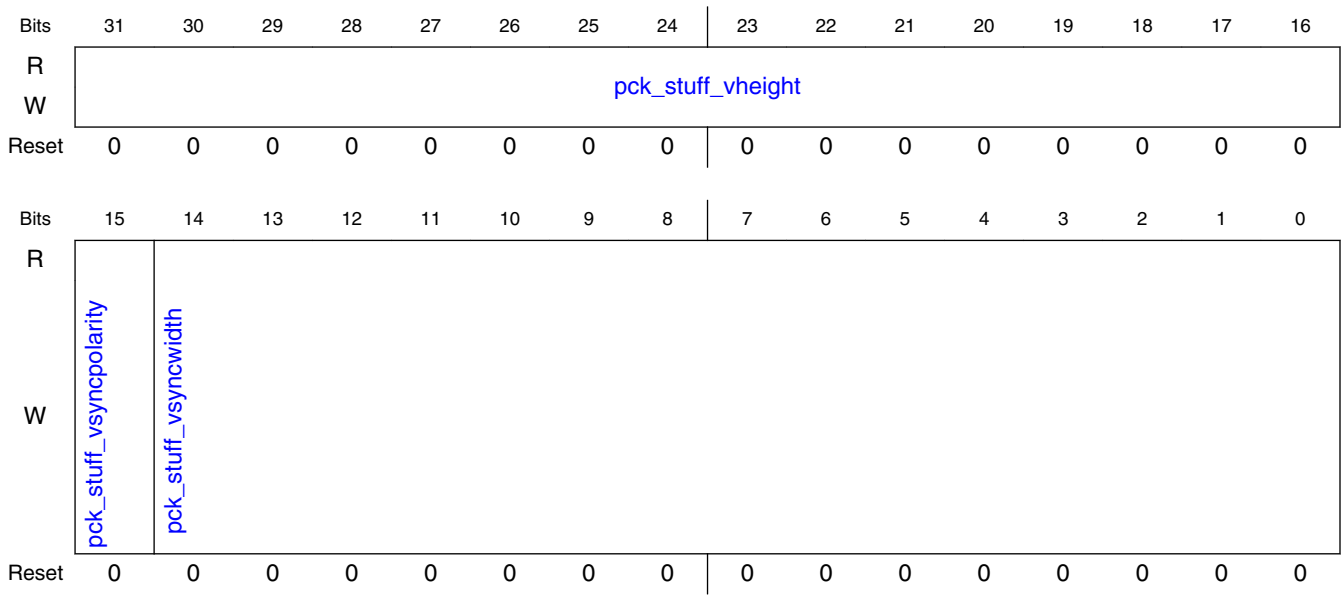
Field	Function
31-16 pck_stuff_vstart	Vertical Active Start, index of the first active line in a frame
15-0 pck_stuff_vtotal	Vertical Total, number of lines per frame

13.4.10.1.146 MSA vertical parameters second part (MSA_VERTICAL_1)

13.4.10.1.146.1 Offset

Register	Offset
MSA_VERTICAL_1	228Ch

13.4.10.1.146.2 Diagram



13.4.10.1.146.3 Fields

Field	Function
31-16 pck_stuff_vheight	Vertical Active Video Height, number of active lines in a frame
15 pck_stuff_vsyncpolarity	Vertical Sync Polarity, 0 - active high, 1- active low
14-0 pck_stuff_vsyncwidth	Vertical Sync Width, number of lines Vsync is active

13.4.10.1.147 MISC0 and MISC1 values of the MSA, as per DPv1. (MSA_MISC)

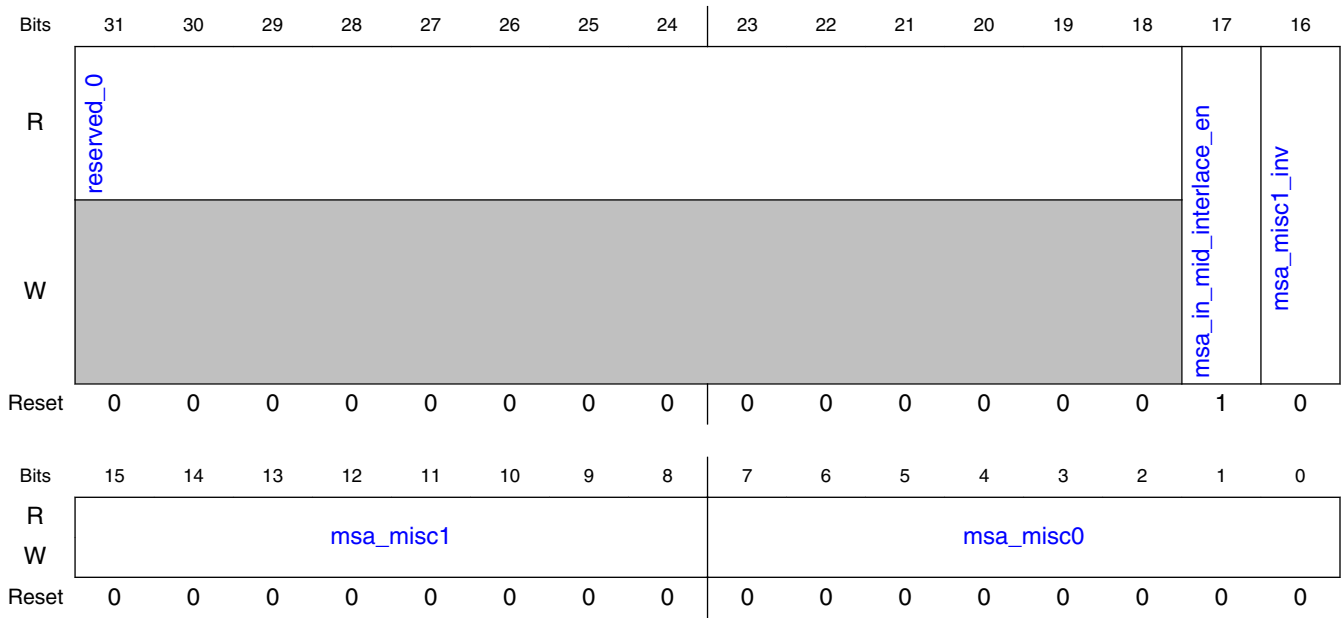
13.4.10.1.147.1 Offset

Register	Offset
MSA_MISC	2290h

13.4.10.1.147.2 Function

MISC0 and MISC1 values of the MSA, as per DPv1.4 standard

13.4.10.1.147.3 Diagram



13.4.10.1.147.4 Fields

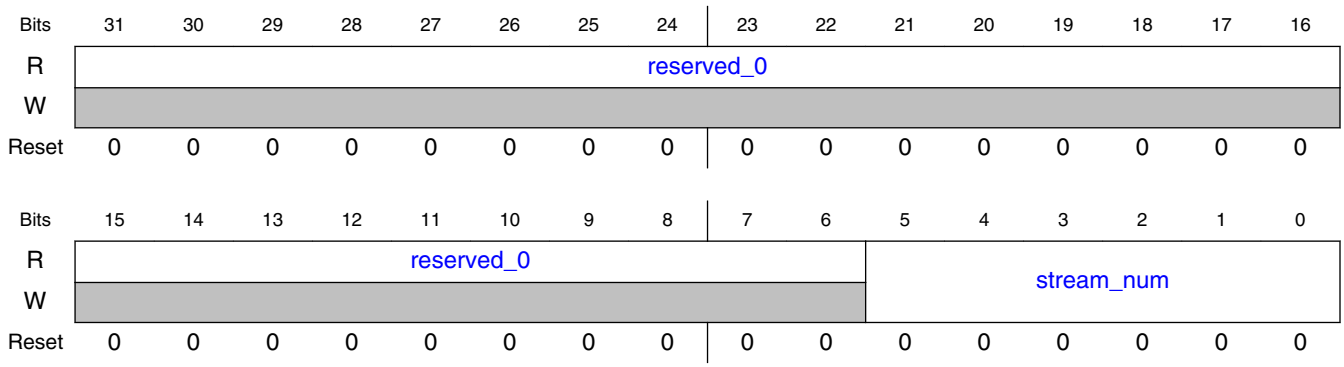
Field	Function
31-18 reserved_0	
17 msa_in_mid_interlace_en	0 - enable transmission of MSA on each field , 1 - MSA is transmitted on Top only
16 msa_misc1_inv	L/R toggle for interlaced and field sequential video, 0 - left, 1 - right
15-8 msa_misc1	Miscellaneous1 value
7-0 msa_misc0	Miscellaneous0 value

13.4.10.1.148 MSA stream number (MST feature) (STREAM_CONFIG)

13.4.10.1.148.1 Offset

Register	Offset
STREAM_CONFIG	2294h

13.4.10.1.148.2 Diagram



13.4.10.1.148.3 Fields

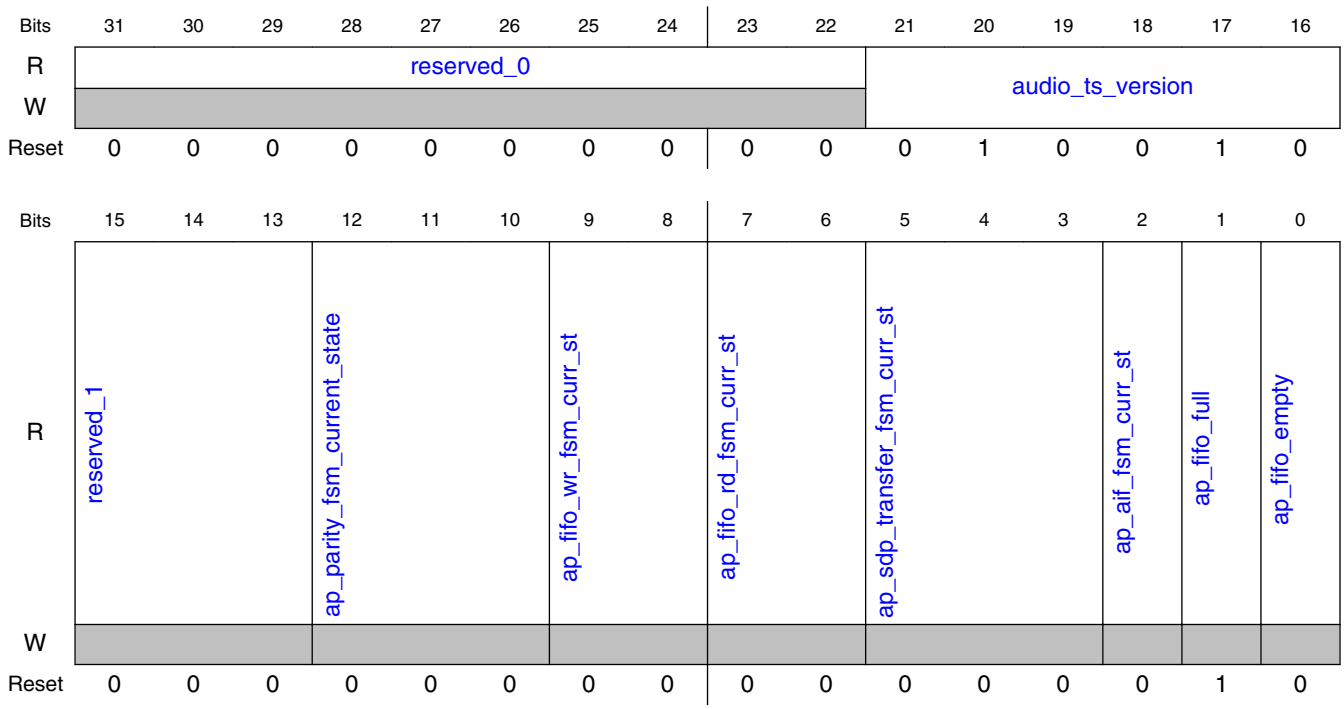
Field	Function
31-6 reserved_0	
5-0 stream_num	Number of current data stream. Number of current data stream.

13.4.10.1.149 Status signals for the audio pack (AUDIO_PACK_STATUS)

13.4.10.1.149.1 Offset

Register	Offset
AUDIO_PACK_STATUS	2298h

13.4.10.1.149.2 Diagram



13.4.10.1.149.3 Fields

Field	Function
31-22 reserved_0	
21-16 audio_ts_version	Audio timestamp version
15-13 reserved_1	
12-10 ap_parity_fsm_current_state	Audio pack parity calc fsm state
9-8 ap_fifo_wr_fsm_curr_st	Audio pack FIFO write fsm state
7-6 ap_fifo_rd_fsm_curr_st	Audio pack FIFO read fsm state
5-3 ap_sdp_transfer_fsm_curr_st	Audio pack sdp transfer fsm state

Table continues on the next page...

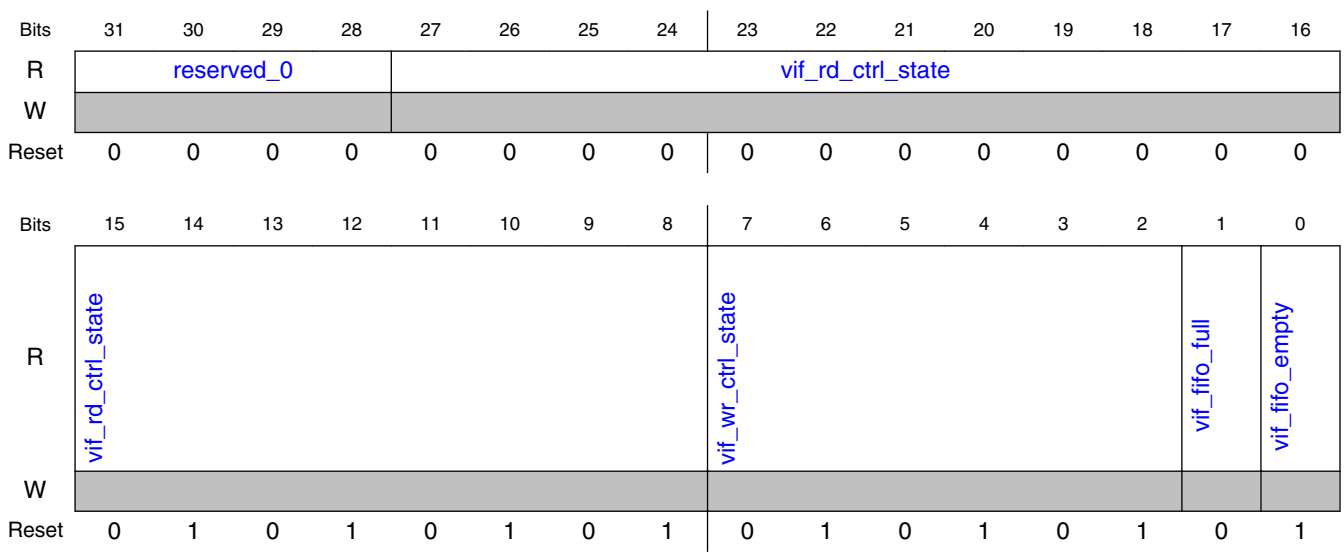
Field	Function
2 ap_aif_fsm_curr_st	Audio pack aif fsm state
1 ap_fifo_full	Audio Pack Sync FIFO full flag, active high
0 ap_fifo_empty	Audio Pack Sync FIFO empty flag, active high

13.4.10.1.150 Status signals for the VIF module (VIF_STATUS)

13.4.10.1.150.1 Offset

Register	Offset
VIF_STATUS	229Ch

13.4.10.1.150.2 Diagram



13.4.10.1.150.3 Fields

Field	Function
31-28 reserved_0	

Table continues on the next page...

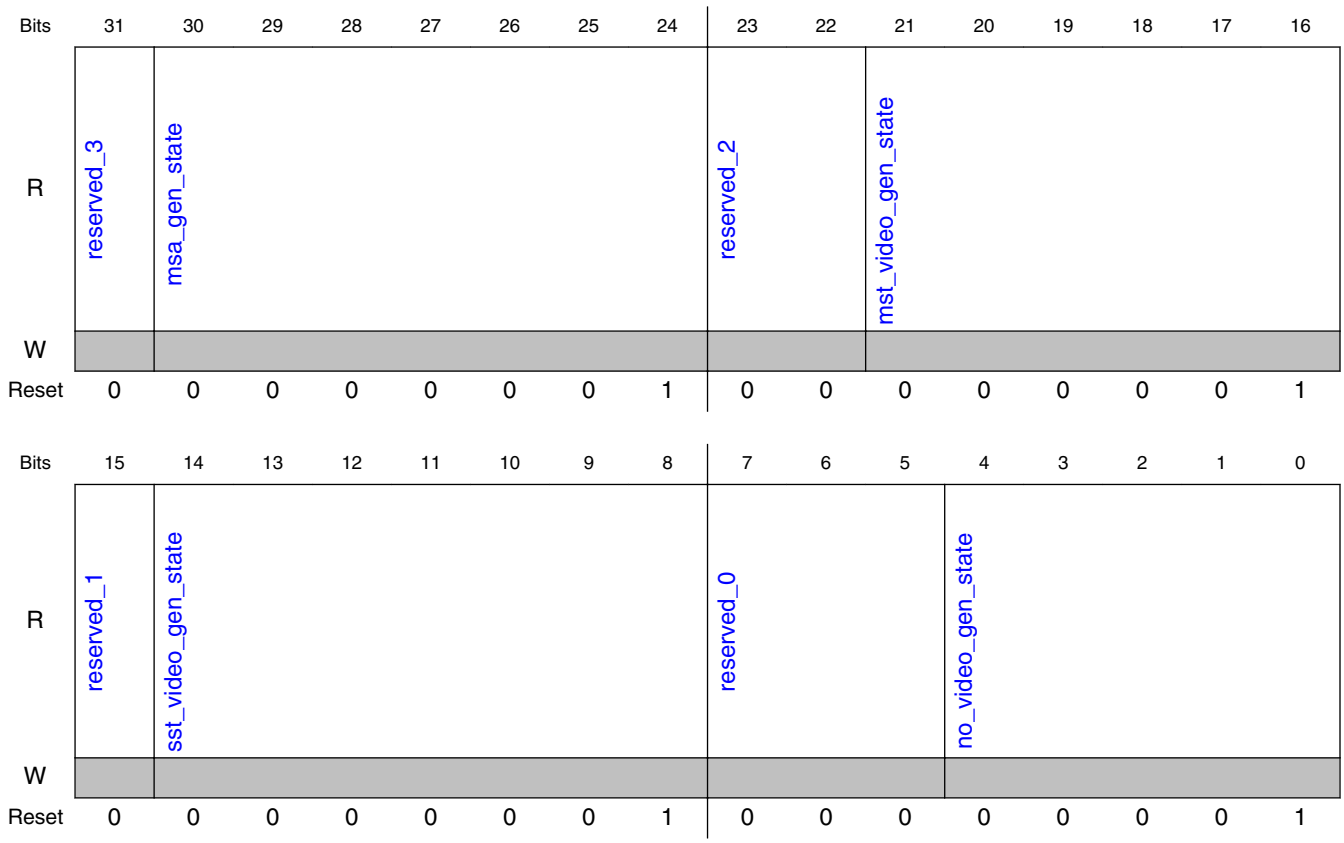
Field	Function
27-8 vif_rd_ctrl_state	VIF rd fsm current state
7-2 vif_wr_ctrl_state	VIF wr fsm current state
1 vif_fifo_full	VIF ASync FIFO full flag, active high
0 vif_fifo_empty	VIF ASync FIFO empty flag, active high

13.4.10.1.151 Status signals for the video stuff module, part 1 (PCK_STUFF_STATUS_0)

13.4.10.1.151.1 Offset

Register	Offset
PCK_STUFF_STATUS_0	22A0h

13.4.10.1.151.2 Diagram



13.4.10.1.151.3 Fields

Field	Function
31 reserved_3	
30-24 msa_gen_state	Secondary Data generator FSM status
23-22 reserved_2	
21-16 mst_video_gen_state	MST video generator FSM status. MST video generator FSM status.
15 reserved_1	
14-8 sst_video_gen_state	SST video generator FSM status. SST video generator FSM status.
7-5	

Table continues on the next page...

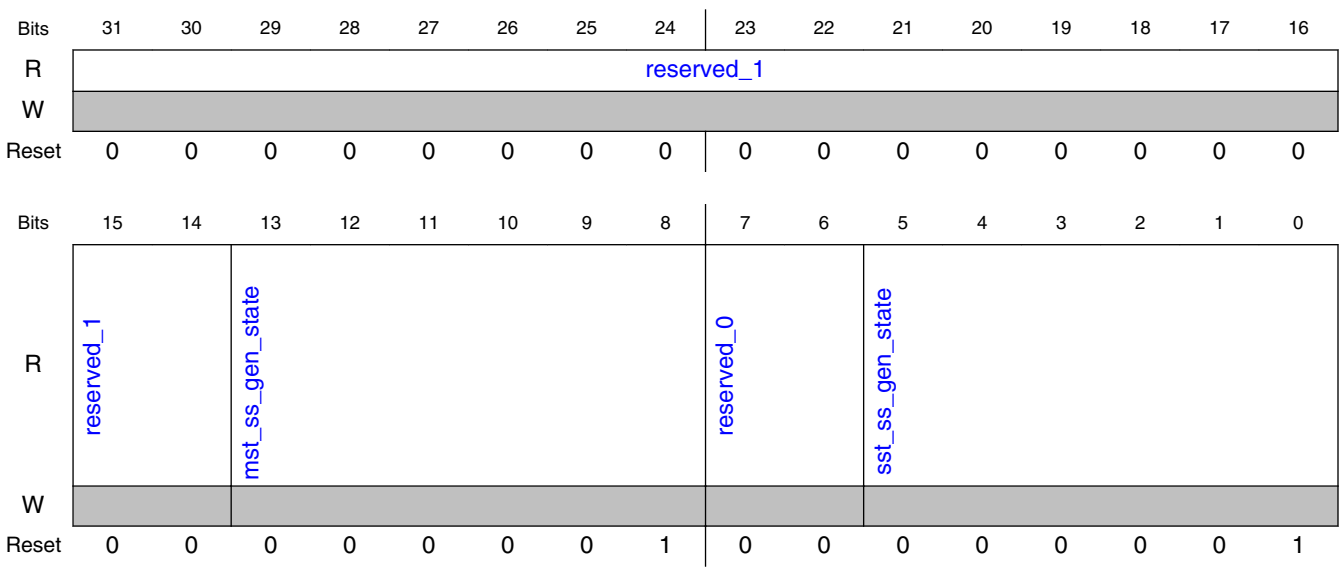
Field	Function
reserved_0	
4-0 no_video_gen_state	No video generator FSM status. No video generator FSM status.

13.4.10.1.152 Status signals for the video stuff module, part 2 (PCK_STUFF_STATUS_1)

13.4.10.1.152.1 Offset

Register	Offset
PCK_STUFF_STATUS_1	22A4h

13.4.10.1.152.2 Diagram



13.4.10.1.152.3 Fields

Field	Function
31-14 reserved_1	
13-8	MST SS generator FSM status

Table continues on the next page...

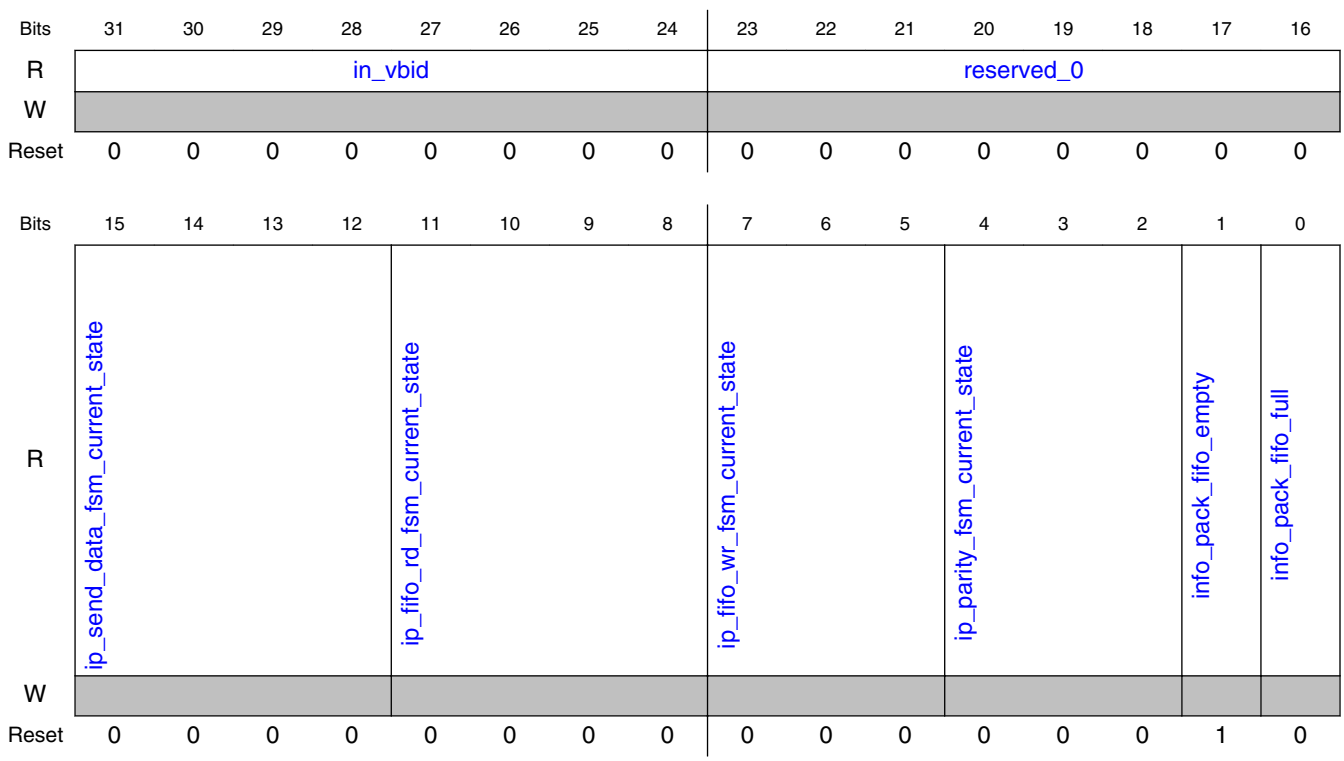
Field	Function
mst_ss_gen_state	
7-6 reserved_0	
5-0 sst_ss_gen_state	MSA generator FSM status

13.4.10.1.153 Status signals for the info pack module, as well as final VB-ID value (INFO_PACK_STATUS)

13.4.10.1.153.1 Offset

Register	Offset
INFO_PACK_STATUS	22A8h

13.4.10.1.153.2 Diagram



13.4.10.1.153.3 Fields

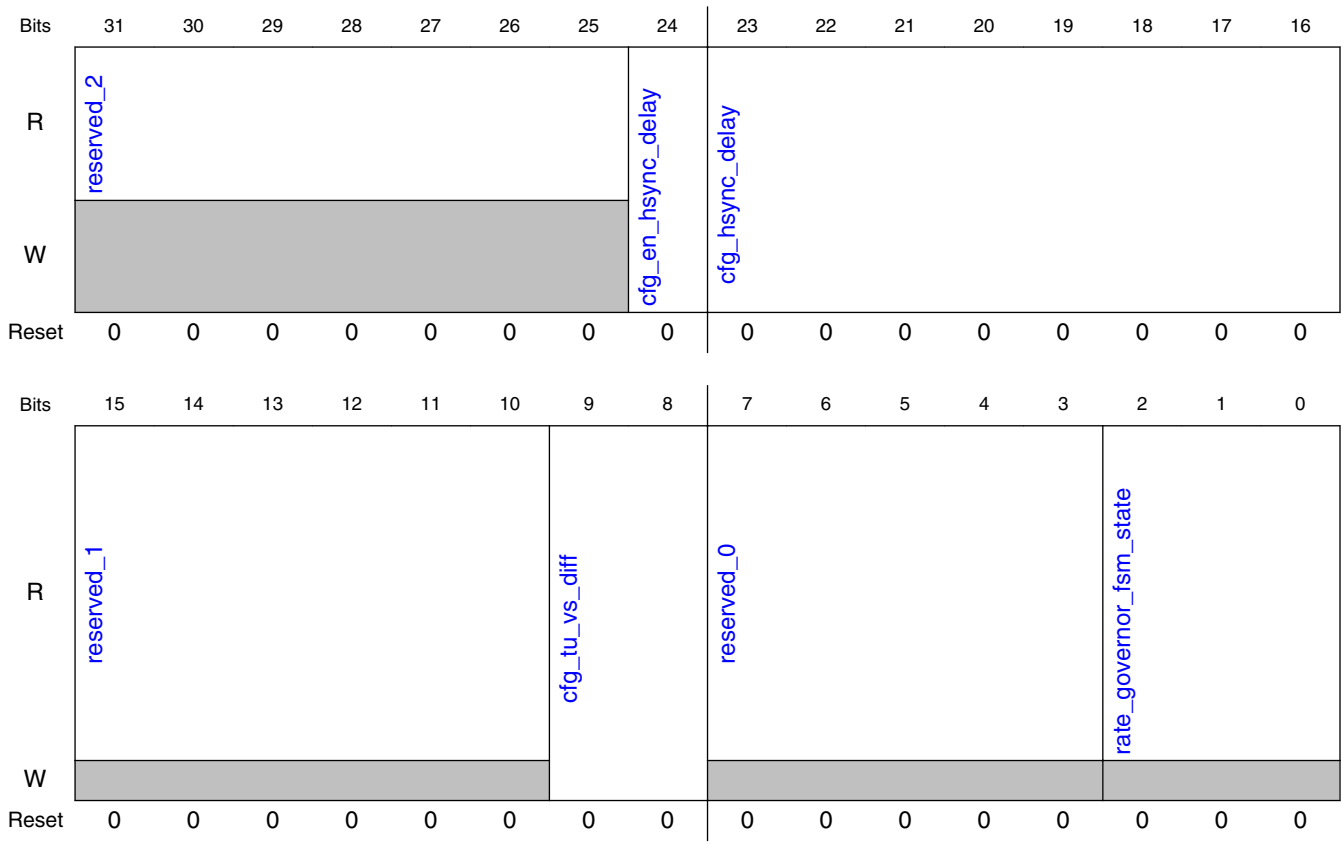
Field	Function
31-24 in_vbid	Value of the sent VB-ID (vb_id_final)
23-16 reserved_0	
15-12 ip_send_data_fsm_current_state	State of the send_data fsm
11-8 ip_fifo_rd_fsm_current_state	State of the fifo_rd fsm
7-5 ip_fifo_wr_fsm_current_state	State of the fifo_wr fsm
4-2 ip_parity_fsm_current_state	State of the parity fsm
1 info_pack_fifo_empty	Info_pack fifo empty flag, active high
0 info_pack_fifo_full	Info_pack fifo full flag, active high

13.4.10.1.154 Rate governer status, as well as additional video configuration (RATE_GOVERNOR_STATUS)

13.4.10.1.154.1 Offset

Register	Offset
RATE_GOVERNOR_STATUS	22ACh

13.4.10.1.154.2 Diagram



13.4.10.1.154.3 Fields

Field	Function
31-25 reserved_2	
24 cfg_en_hsync_delay	Hsync Delay Enable. Hsync Delay Enable. Active high
23-16 cfg_hsync_delay	Hsync delay value (in pixel clocks). Hsync delay value (in pixel clocks). Actual delay is: (cfg_hsync_delay+2) pixel clocks.
15-10 reserved_1	
9-8 cfg_tu_vs_diff	TU-VS difference. TU-VS difference. b'00: TU-VS>3, b'01:NA, b'10:TU-VS=2, b'11:TU-VS=3
7-3 reserved_0	
2-0	Status of the governer FSM, relevant only for MST

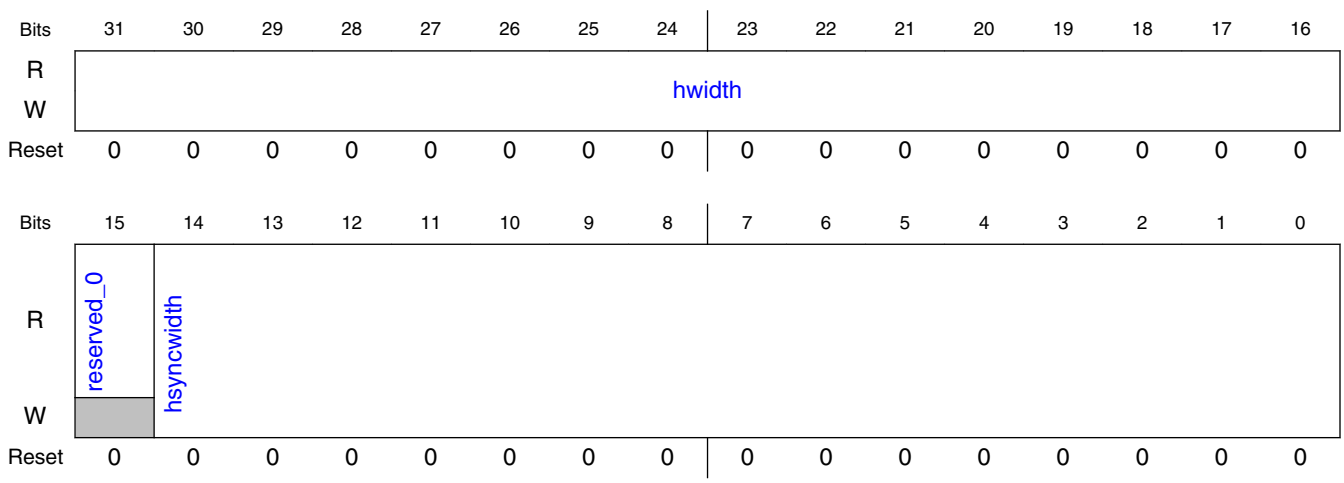
Field	Function
rate_governor_fsm_state	

13.4.10.1.155 Video line parameters to be used in the RTL calculations (DP_HORIZONTAL)

13.4.10.1.155.1 Offset

Register	Offset
DP_HORIZONTAL	22B0h

13.4.10.1.155.2 Diagram



13.4.10.1.155.3 Fields

Field	Function
31-16 hwidth	Horizontal Active Video Width, number of active pixels per line
15 reserved_0	
14-0 hsyncwidth	Horizontal Sync Width, number of pixels Hsync is active

13.4.10.1.156 Video frame parameters to be used in the RTL calculations, part 1 (DP_VERTICAL_0)

13.4.10.1.156.1 Offset

Register	Offset
DP_VERTICAL_0	22B4h

13.4.10.1.156.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	vstart															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	vheight															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.156.3 Fields

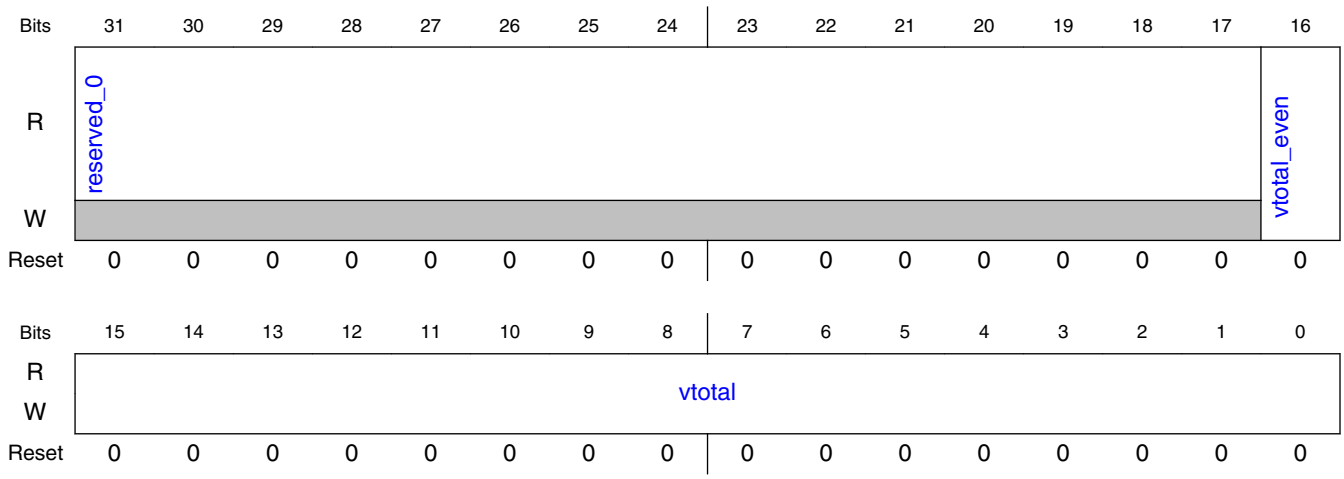
Field	Function
31-16 vstart	Vertical Active Start, index of the first active line in a frame
15-0 vheight	Vertical Active Video Height, number of active lines in a frame

13.4.10.1.157 Video frame parameters to be used in the RTL calculations, part 2 (DP_VERTICAL_1)

13.4.10.1.157.1 Offset

Register	Offset
DP_VERTICAL_1	22B8h

13.4.10.1.157.2 Diagram



13.4.10.1.157.3 Fields

Field	Function
31-17 reserved_0	
16 vtotal_even	Indicate Vtotal is Even Number like MISC, active high
15-0 vtotal	Vertical Total, number of lines per frame

13.4.10.1.158 (DP_BLOCK_SDP)

13.4.10.1.158.1 Offset

Register	Offset
DP_BLOCK_SDP	22BCh

13.4.10.1.158.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0				no_video_block_sdp											
W																
Reset	0	0	0	0	1	1	1	1	1	0	1	1	1	1	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	block_sdp_be								block_sdp_bs							
W																
Reset	0	0	0	0	0	1	0	1	0	0	0	0	0	1	0	1

13.4.10.1.158.3 Fields

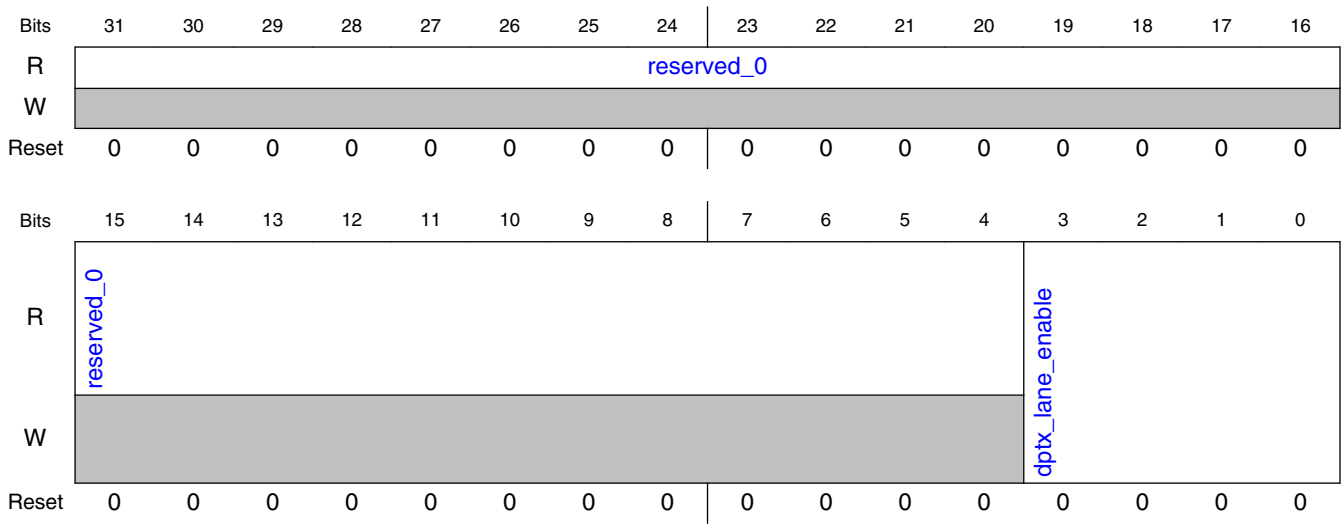
Field	Function
31-28 reserved_0	
27-16 no_video_block_sdp	
15-8 block_sdp_be	
7-0 block_sdp_bs	

13.4.10.1.159 DPTX lane enable (DPTX_LANE_EN)

13.4.10.1.159.1 Offset

Register	Offset
DPTX_LANE_EN	2300h

13.4.10.1.159.2 Diagram



13.4.10.1.159.3 Fields

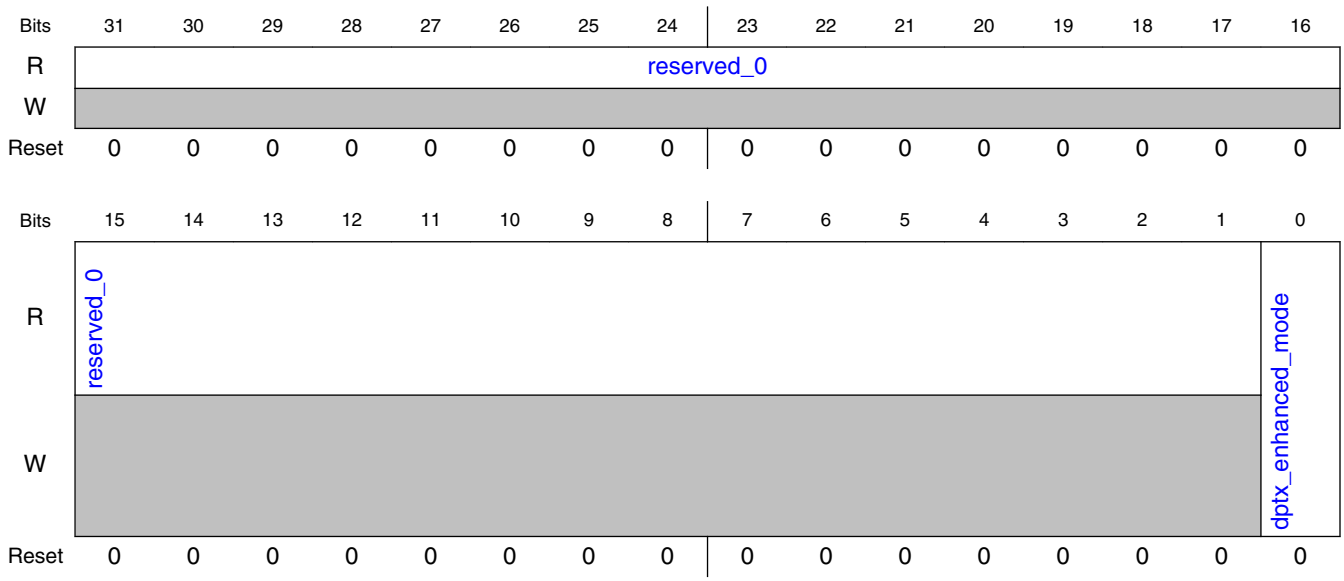
Field	Function
31-4 reserved_0	
3-0 dptx_lane_enable	DPTX lane enable each lane as a bit when 1 lane is enabled

13.4.10.1.160 DPTX enhanced mode (DPTX_ENHNCD)

13.4.10.1.160.1 Offset

Register	Offset
DPTX_ENHNCD	2304h

13.4.10.1.160.2 Diagram



13.4.10.1.160.3 Fields

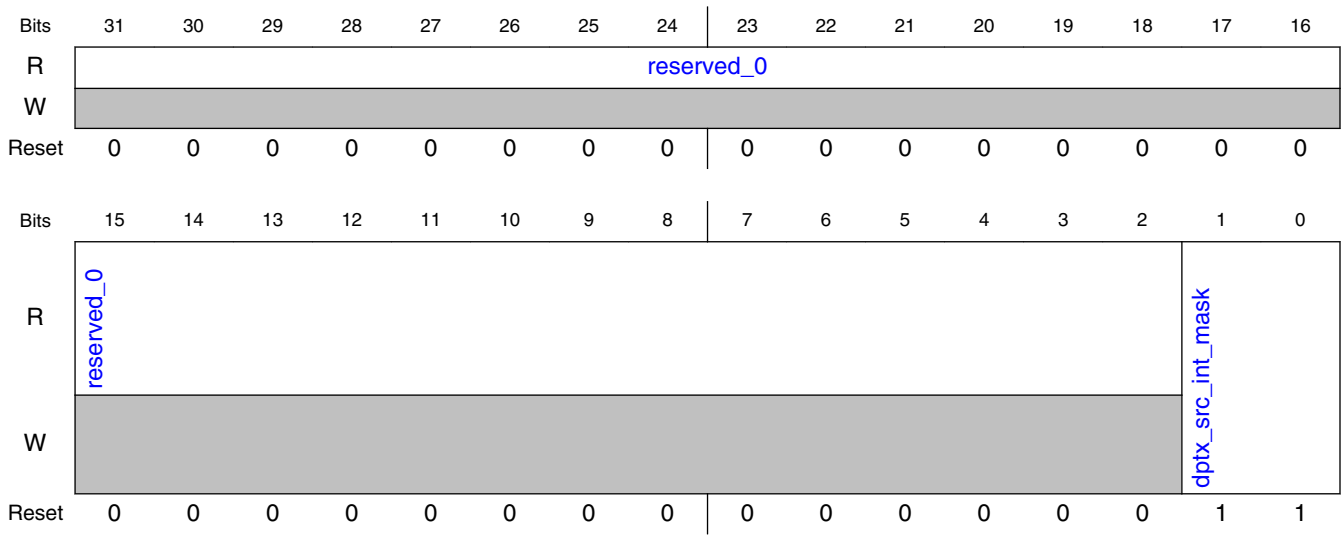
Field	Function
31-1 reserved_0	
0 dptx_enhanced_mode	DPTX is in enhanced mode when 1

13.4.10.1.161 DPTX Interrupt mask (DPTX_INT_MASK)

13.4.10.1.161.1 Offset

Register	Offset
DPTX_INT_MASK	2308h

13.4.10.1.161.2 Diagram



13.4.10.1.161.3 Fields

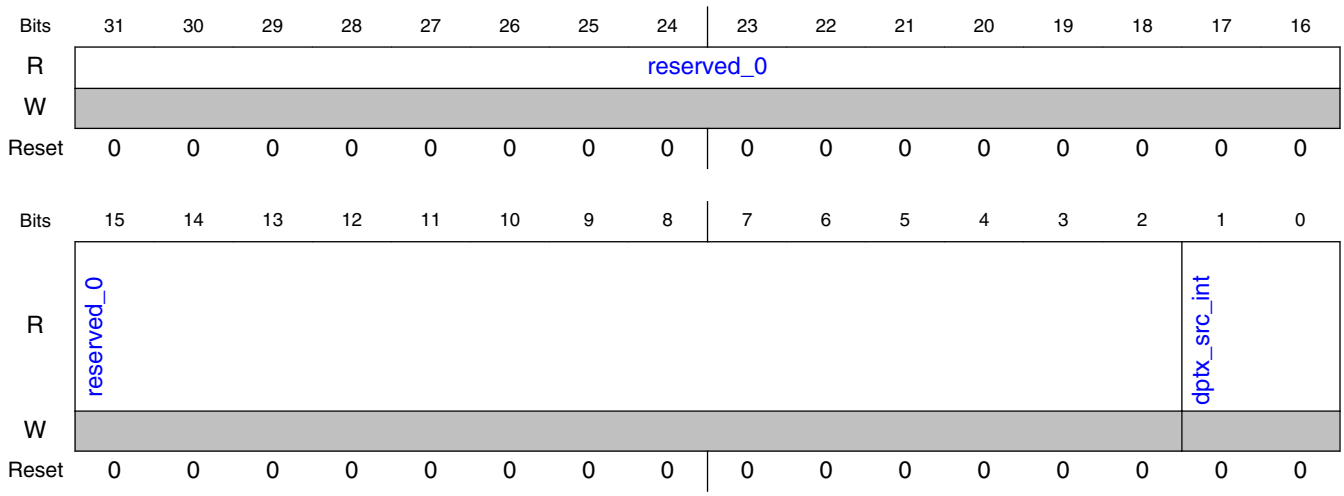
Field	Function
31-2 reserved_0	
1-0 dptx_src_int_mask	DPTX mask interupts :-0-HPD interupt Mask -1-Framer Interupt Mask

13.4.10.1.162 DPTX interrupt status (DPTX_INT_STATUS)

13.4.10.1.162.1 Offset

Register	Offset
DPTX_INT_STATUS	230Ch

13.4.10.1.162.2 Diagram



13.4.10.1.162.3 Fields

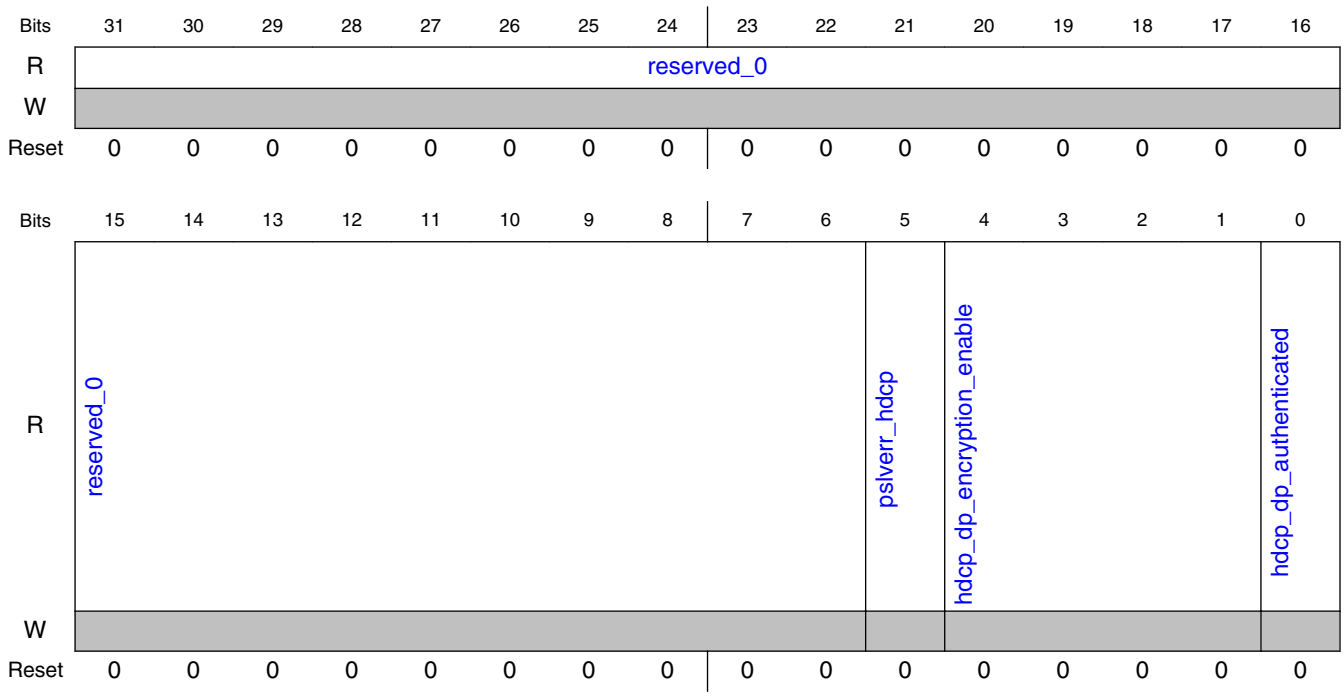
Field	Function
31-2 reserved_0	
1-0 dptx_src_int	HPD event Status-FRAMER event Status

13.4.10.1.163 HDCP DP status register (HDCP_DP_STATUS)

13.4.10.1.163.1 Offset

Register	Offset
HDCP_DP_STATUS	2400h

13.4.10.1.163.2 Diagram



13.4.10.1.163.3 Fields

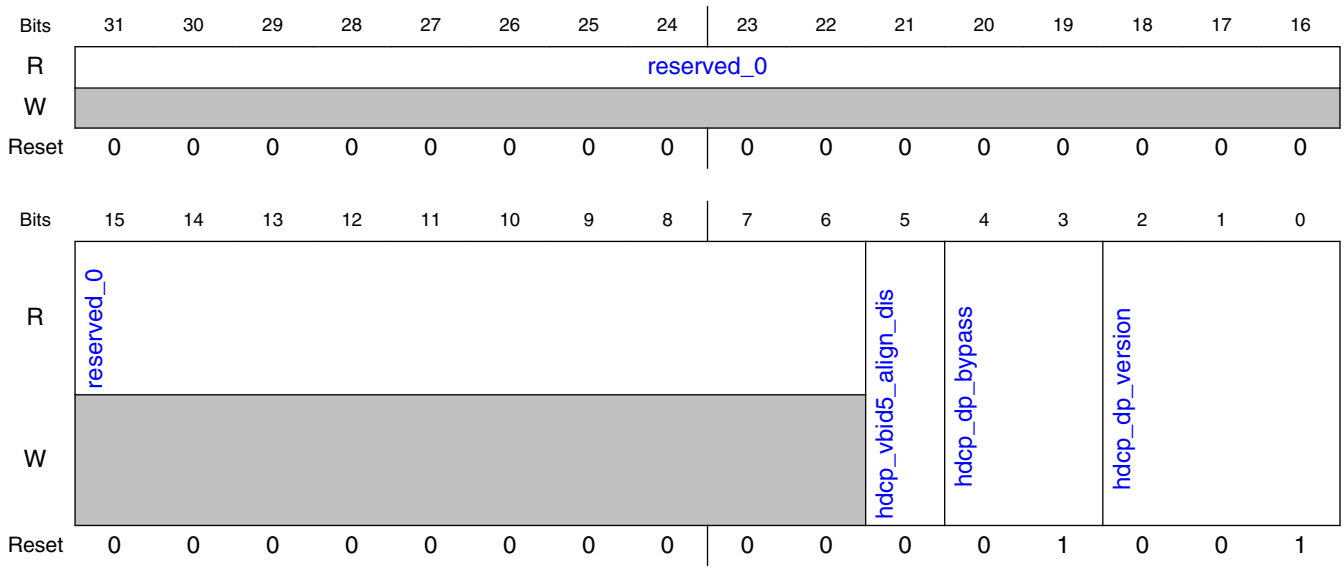
Field	Function
31-6 reserved_0	
5 pslverr_hdcp	APB slave error status from HDCP module. APB slave error status from HDCP module.
4-1 hdcp_dp_encryption_enable	Encryption enabled
0 hdcp_dp_authenticated	Authentication enabled HDCP 1. Authentication enabled HDCP 1.4

13.4.10.1.164 HDCP DP config register (HDCP_DP_CONFIG)

13.4.10.1.164.1 Offset

Register	Offset
HDCP_DP_CONFIG	2404h

13.4.10.1.164.2 Diagram



13.4.10.1.164.3 Fields

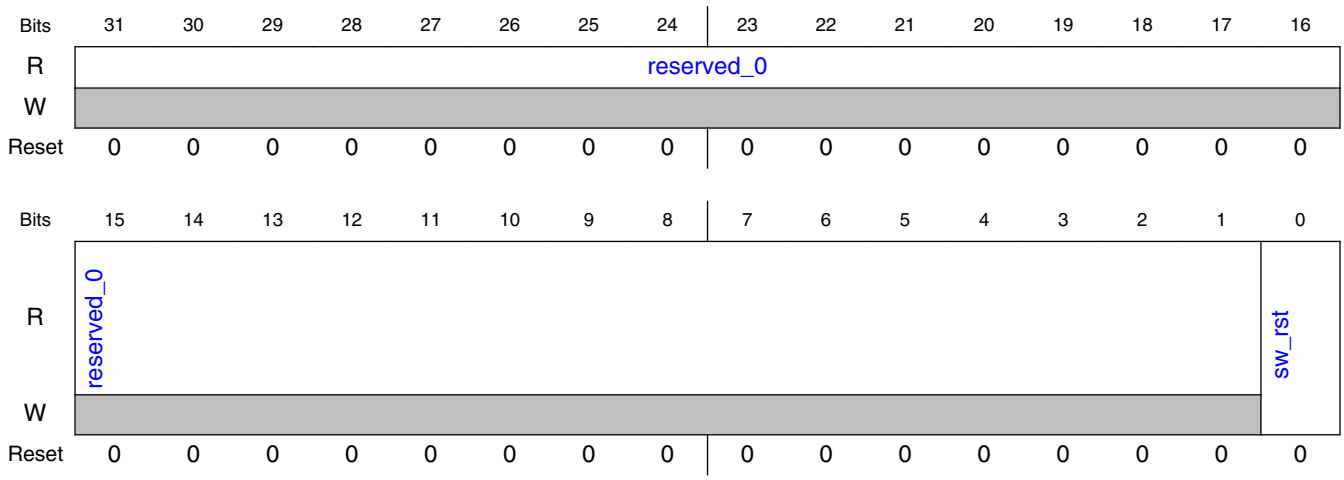
Field	Function
31-6 reserved_0	
5 hdcp_vbid5_align_dis	
4-3 hdcp_dp_bypass	HDCP DP bypass. HDCP DP bypass. 0x0-No bypass(HDCP enabled); 0x1-Bypass enabled; 0x2-Data from FIFO6 (debug, not intended for normal usage); 0x3-Data from FIFO4 (debug, not intended for normal usage).
2-0 hdcp_dp_version	HDCP version. HDCP version. 0x1-HDCP2.2; 0x2-HDCP1.4; Other-Reserved

13.4.10.1.165 HDCP DP software reset register (HDCP_DP_SW_RST)

13.4.10.1.165.1 Offset

Register	Offset
HDCP_DP_SW_RST	2408h

13.4.10.1.165.2 Diagram



13.4.10.1.165.3 Fields

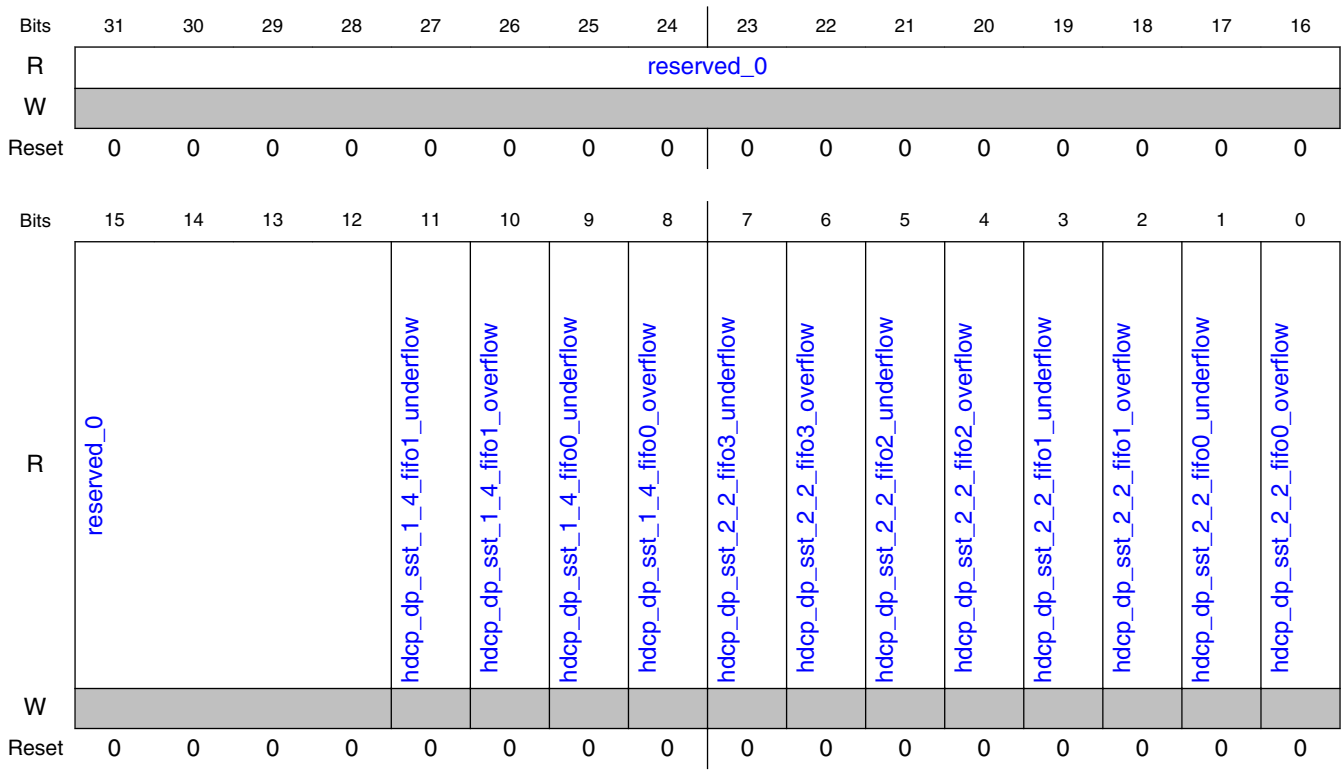
Field	Function
31-1 reserved_0	
0 sw_rst	Software reset

13.4.10.1.166 HDCP DP FIFO status register (HDCP_DP_FIFO_STATUS)

13.4.10.1.166.1 Offset

Register	Offset
HDCP_DP_FIFO_STATUS	240Ch

13.4.10.1.166.2 Diagram



13.4.10.1.166.3 Fields

Field	Function
31-12 reserved_0	
11 hdcp_dp_sst_1_4_fifo1_underflow	SST HDCP1. SST HDCP1.4 fifo1 underflow
10 hdcp_dp_sst_1_4_fifo1_overflow	SST HDCP1. SST HDCP1.4 fifo1 overflow
9 hdcp_dp_sst_1_4_fifo0_underflow	SST HDCP1. SST HDCP1.4 fifo0 underflow
8 hdcp_dp_sst_1_4_fifo0_overflow	SST HDCP1. SST HDCP1.4 fifo0 overflow
7	SST HDCP2. SST HDCP2.2 fifo3 underflow

Table continues on the next page...

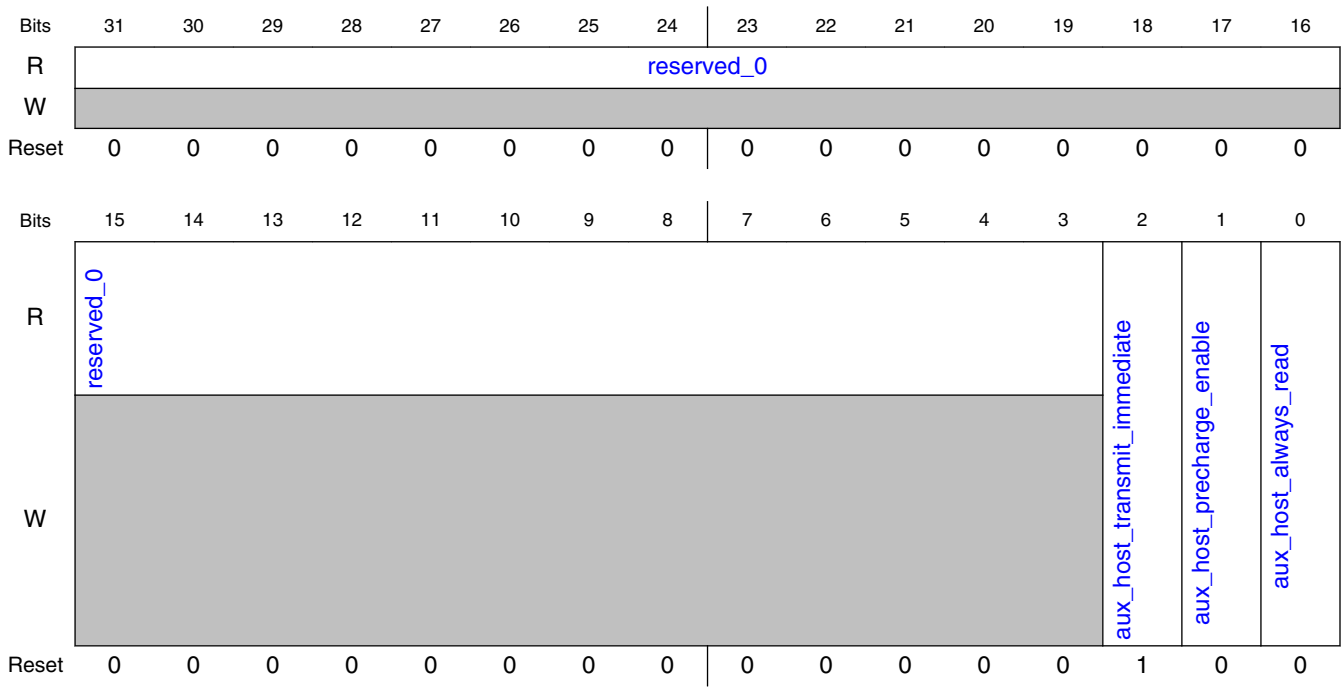
Field	Function
hdcp_dp_sst_2_2_fifo3_underflow	
6 hdcp_dp_sst_2_2_fifo3_overflow	SST HDCP2. SST HDCP2.2 fifo3 overflow
5 hdcp_dp_sst_2_2_fifo2_underflow	SST HDCP2. SST HDCP2.2 fifo2 underflow
4 hdcp_dp_sst_2_2_fifo2_overflow	SST HDCP2. SST HDCP2.2 fifo2 overflow
3 hdcp_dp_sst_2_2_fifo1_underflow	SST HDCP2. SST HDCP2.2 fifo1 underflow
2 hdcp_dp_sst_2_2_fifo1_overflow	SST HDCP2. SST HDCP2.2 fifo1 overflow
1 hdcp_dp_sst_2_2_fifo0_underflow	SST HDCP2. SST HDCP2.2 fifo0 underflow
0 hdcp_dp_sst_2_2_fifo0_overflow	SST HDCP2. SST HDCP2.2 fifo0 overflow

13.4.10.1.167 Control bits for DP_AUX (DP_AUX_HOST_CONTROL)

13.4.10.1.167.1 Offset

Register	Offset
DP_AUX_HOST_CONTROL	2800h

13.4.10.1.167.2 Diagram



13.4.10.1.167.3 Fields

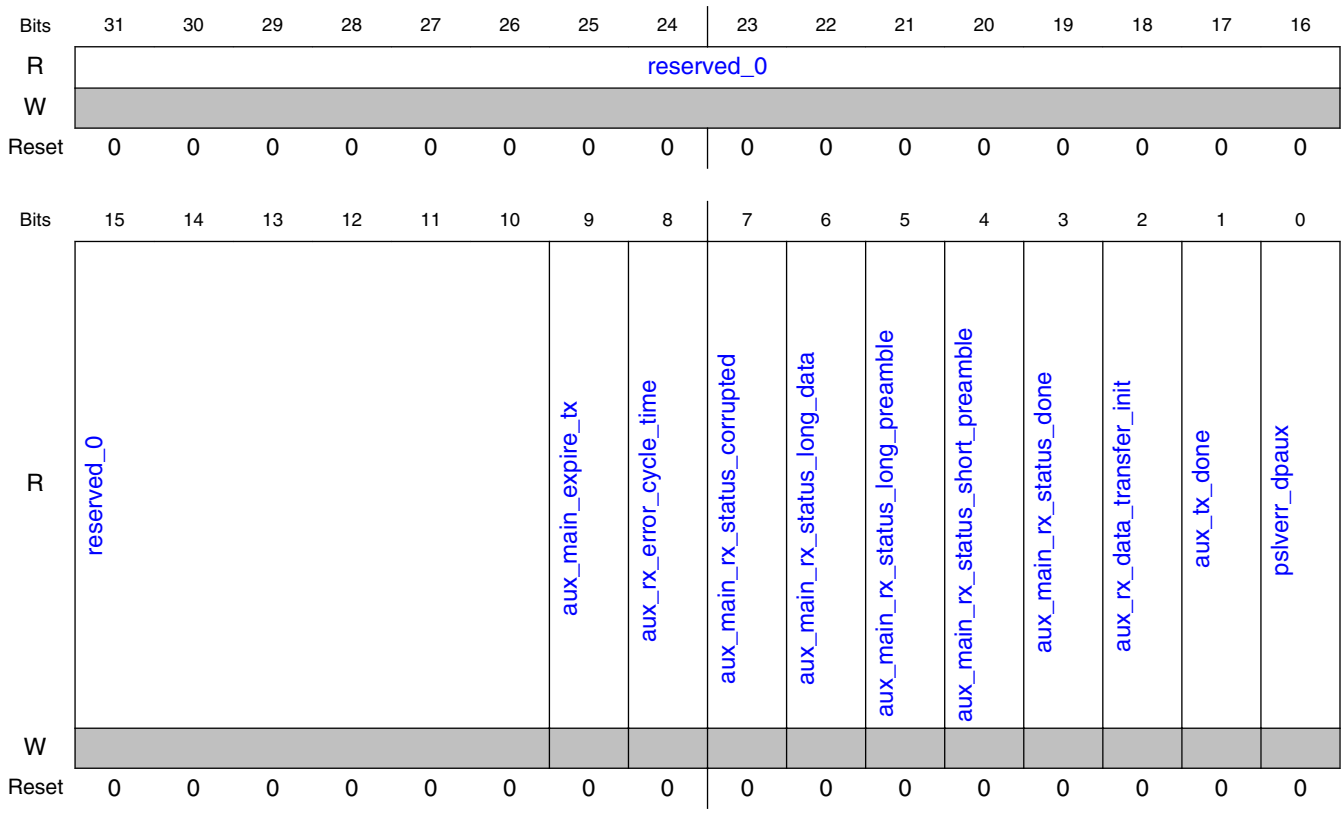
Field	Function
31-3 reserved_0	
2 aux_host_transmit_immediate	This bit is used only in DP_OUT mode. This bit is used only in DP_OUT mode. If SET, a transaction that comes from the adapter will be sent immediately without waiting for send_external_transaction pulse. If CLEAR, the MC controls the traffic to/from the adapter.
1 aux_host_precharge_enable	According to the current standard the tx precharge is done by sending 10 to 16 data_0 on the line before the SYNC. According to the current standard the tx precharge is done by sending 10 to 16 data_0 on the line before the SYNC. Old standard define the precharge by forcing the AFE to be in precharge mode before transmitting the SYNC.
0 aux_host_always_read	Normally, the aux_rx is disabled during transmit. Normally, the aux_rx is disabled during transmit. Setting this bit allow loopback operation and all transmit transactions will go to the receiver. Used for debug.

13.4.10.1.168 Status of the DP_AUX interrupt sources (DP_AUX_INTERRUPT_SOURCE)

13.4.10.1.168.1 Offset

Register	Offset
DP_AUX_INTERRUPT_SOURCE	2804h

13.4.10.1.168.2 Diagram



13.4.10.1.168.3 Fields

Field	Function
31-10 reserved_0	
9 aux_main_expire_tx	timer expire (external) in DP_OUT
8 aux_rx_error_cycle_time	Cycle time error. Cycle time error. Asserted if aux_rx_last_cycle is less then aux_host_1m_min or greater then aux_host_1m_max.
7	The received transaction corrupted during the data phase (bad STOP, or unaligned STOP).

Table continues on the next page...

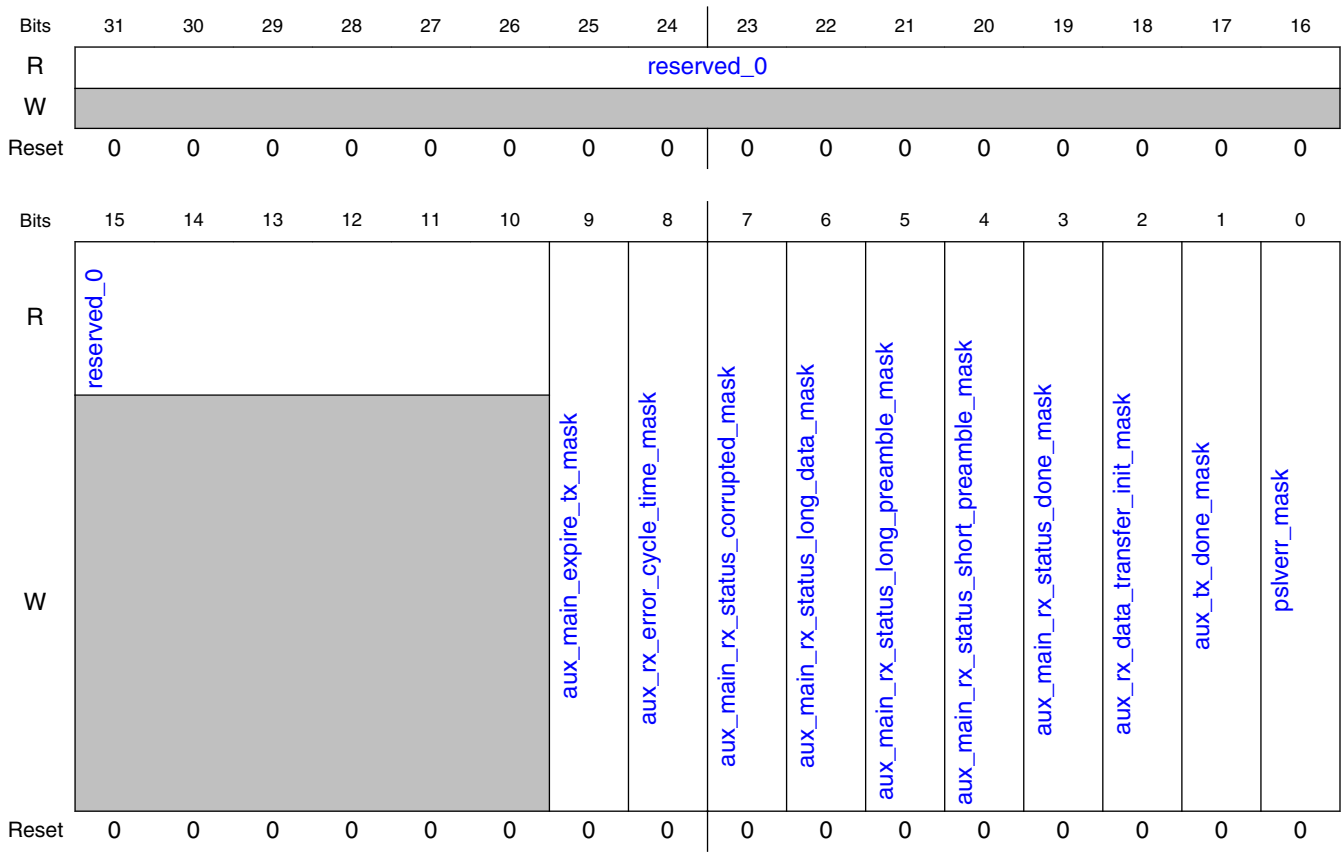
Field	Function
aux_main_rx_status_corrupted	The received transaction corrupted during the data phase (bad STOP, or unaligned STOP).
6 aux_main_rx_status_long_data	The received transaction had more than 20 data bytes. The received transaction had more than 20 data bytes.
5 aux_main_rx_status_long_preamble	The received transaction had preamble greater than the preamble_max. The received transaction had preamble greater than the preamble_max.
4 aux_main_rx_status_short_preamble	The received transaction had preamble shorter than the preamble_max. The received transaction had preamble shorter than the preamble_max.
3 aux_main_rx_status_done	This module control the packet extraction and packet read from the memory. This module control the packet extraction and packet read from the memory.
2 aux_rx_data_transfer_init	Rx data transfer may be initiated. Rx data transfer may be initiated. Falling edge of the aux_mailbox_empty.
1 aux_tx_done	Tx data transfer finished. Tx data transfer finished.
0 pslverr_dpaux	APB slave error interrupt from DP AUX module. APB slave error interrupt from DP AUX module.

13.4.10.1.169 Mask vector of the DP_AUX interrupt sources (DP_AUX_INTERRUPT_MASK)

13.4.10.1.169.1 Offset

Register	Offset
DP_AUX_INTERRUPT_MASK	2808h

13.4.10.1.169.2 Diagram



13.4.10.1.169.3 Fields

Field	Function
31-10 reserved_0	
9 aux_main_expire_tx_mask	aux_main_expire_external
8 aux_rx_error_cycle_time_mask	aux_rx_error_cycle_time mask
7 aux_main_rx_status_corrupted_mask	aux_main_rx_status_corrupted_mask
6 aux_main_rx_status_long_data_mask	aux_main_rx_status_long_data_mask

Table continues on the next page...

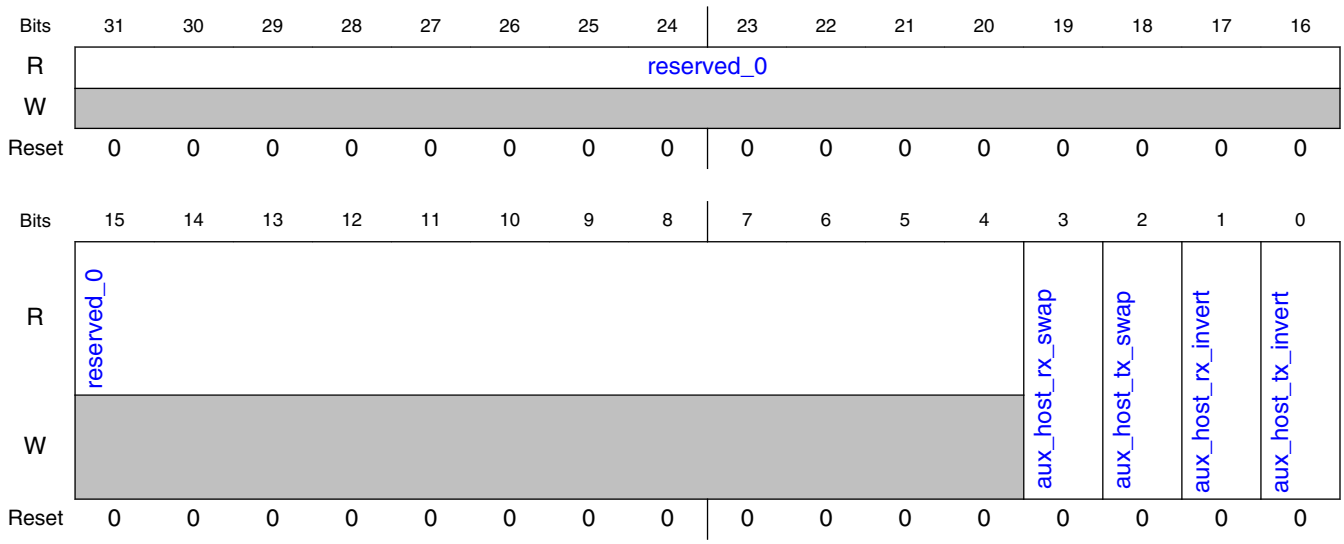
Field	Function
5 aux_main_rx_status_long_preamble_mask	aux_main_rx_status_long_preamble_mask
4 aux_main_rx_status_short_preamble_mask	aux_main_rx_status_short_preamble_mask
3 aux_main_rx_status_done_mask	aux_main_rx_status_done_mask
2 aux_rx_data_transfer_init_mask	rx_data_transfer_init mask
1 aux_tx_done_mask	aux_tx_done_mask
0 pslverr_mask	Mask for pslverr_dpaux interrupt

13.4.10.1.170 Ordering and inversion of transmit/receive on Auxiliary Channel (DP_AUX_SWAP_INVERSION_CONTROL)

13.4.10.1.170.1 Offset

Register	Offset
DP_AUX_SWAP_INVERSION_CONTROL	280Ch

13.4.10.1.170.2 Diagram



13.4.10.1.170.3 Fields

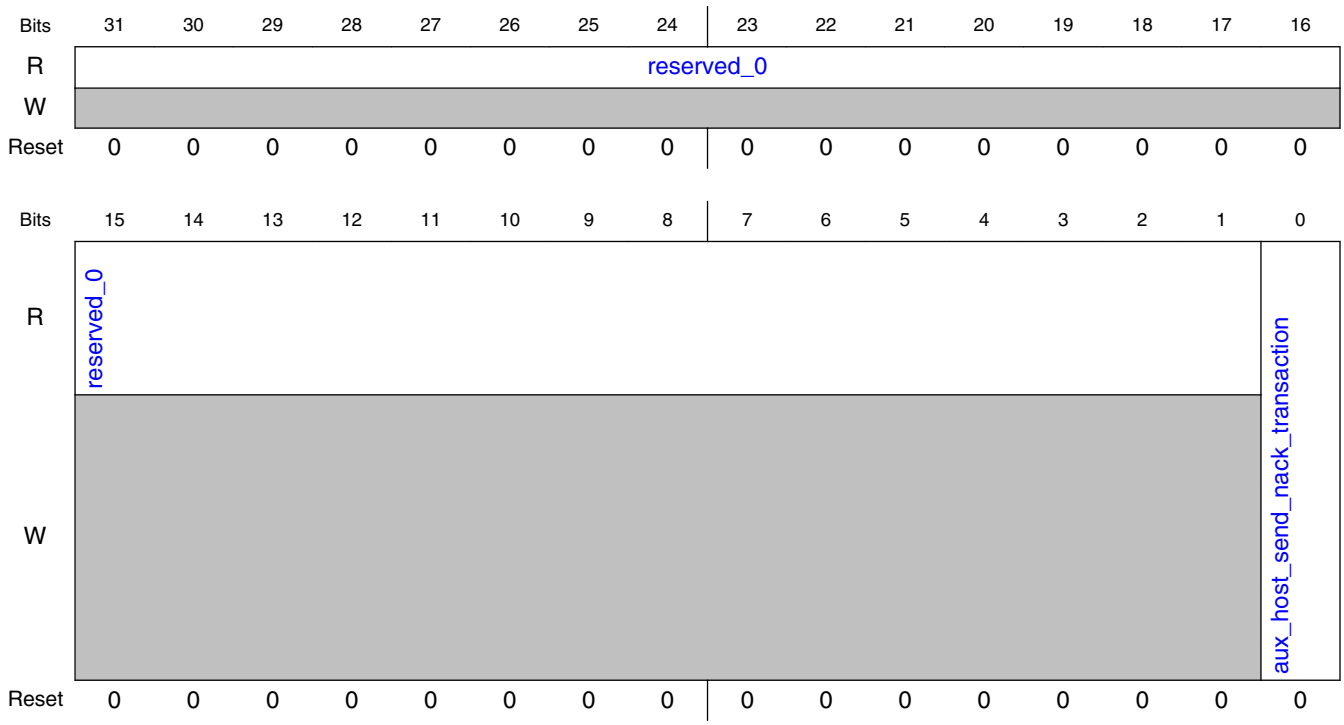
Field	Function
31-4 reserved_0	
3 aux_host_rx_swap	Shift right (LSB first) of the income data
2 aux_host_tx_swap	Shift right the output data (LSB first)
1 aux_host_rx_invert	Invert rx input and output data to AUXILIARY CHANNEL
0 aux_host_tx_invert	Invert tx input and output data to AUXILIARY CHANNEL

13.4.10.1.171 NACK transaction send (DP_AUX_SEND_NACK_TRANSACTION)

13.4.10.1.171.1 Offset

Register	Offset
DP_AUX_SEND_NACK_TRANSACTION	2810h

13.4.10.1.171.2 Diagram



13.4.10.1.171.3 Fields

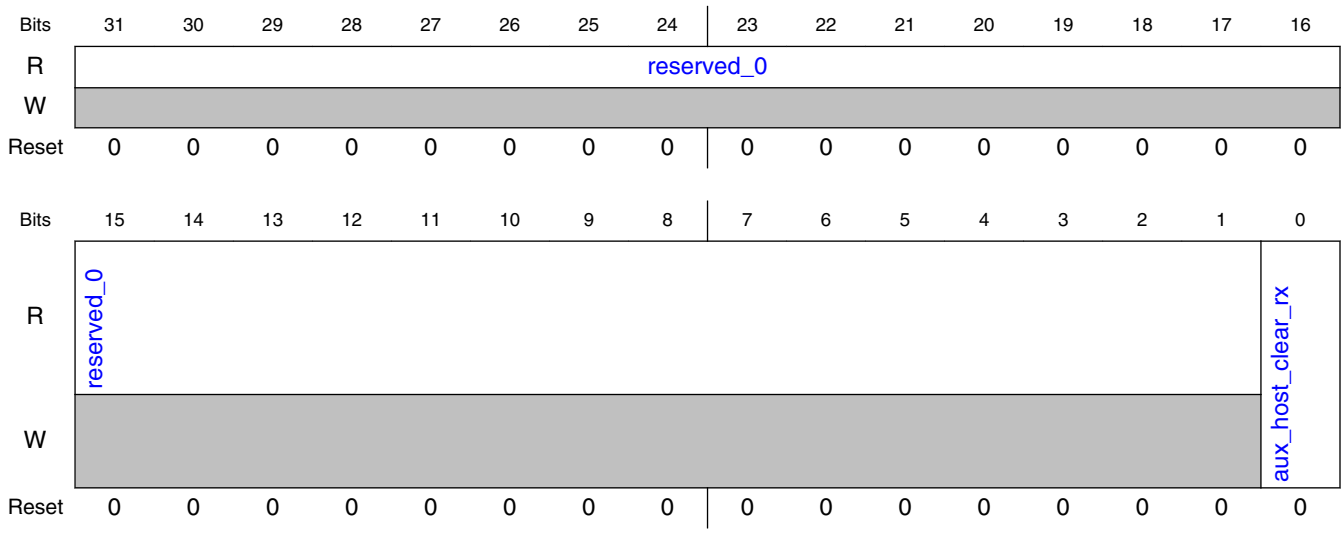
Field	Function
31-1 reserved_0	
0 aux_host_send_nack_transaction	Send nack transaction by AUX_TX. Send nack transaction by AUX_TX. This bit is automaticaly cleared when operation in completed.

13.4.10.1.172 RX bits clear (DP_AUX_CLEAR_RX)

13.4.10.1.172.1 Offset

Register	Offset
DP_AUX_CLEAR_RX	2814h

13.4.10.1.172.2 Diagram



13.4.10.1.172.3 Fields

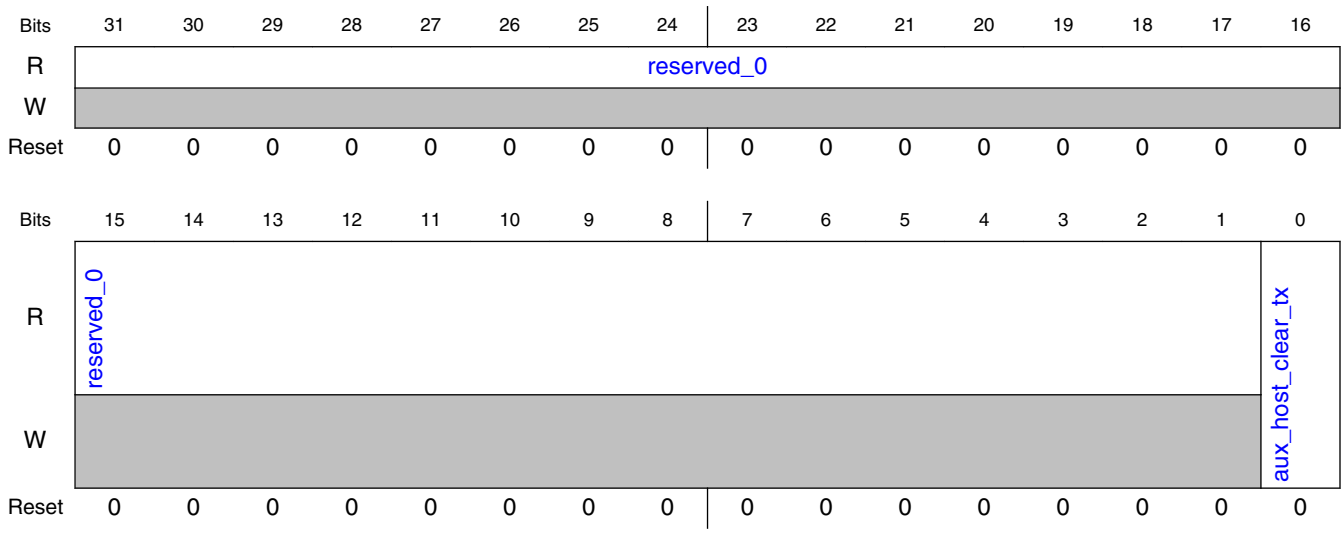
Field	Function
31-1 reserved_0	
0 aux_host_clear_rx	Clear all rx bits in register 64,65. Clear all rx bits in register 64,65.This command is an indication that the processing of last receive transaction was completed and the AUX_RX can start looking for new receive transaction. This bit is automaticaly cleared when operation in completed.

13.4.10.1.173 TX bits clear (DP_AUX_CLEAR_TX)

13.4.10.1.173.1 Offset

Register	Offset
DP_AUX_CLEAR_TX	2818h

13.4.10.1.173.2 Diagram



13.4.10.1.173.3 Fields

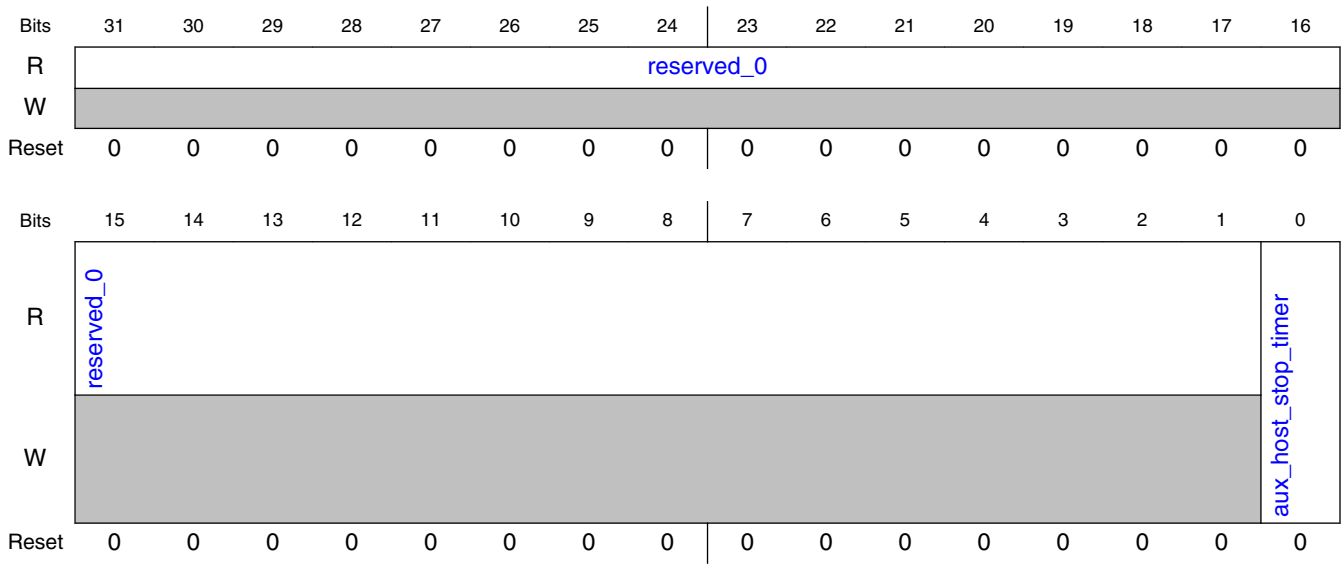
Field	Function
31-1 reserved_0	
0 aux_host_clear_tx	Clear all external bits in registers 64,67. Clear all external bits in registers 64,67.This command used in DP_IN mode. It is an indication that the processing of last external transaction was completed and the DP_AUX can start receive new external transaction from the adapter. This bit is automaticaly cleared when operation in completed.

13.4.10.1.174 Stop timer operation (DP_AUX_TIMER_STOP)

13.4.10.1.174.1 Offset

Register	Offset
DP_AUX_TIMER_STOP	281Ch

13.4.10.1.174.2 Diagram



13.4.10.1.174.3 Fields

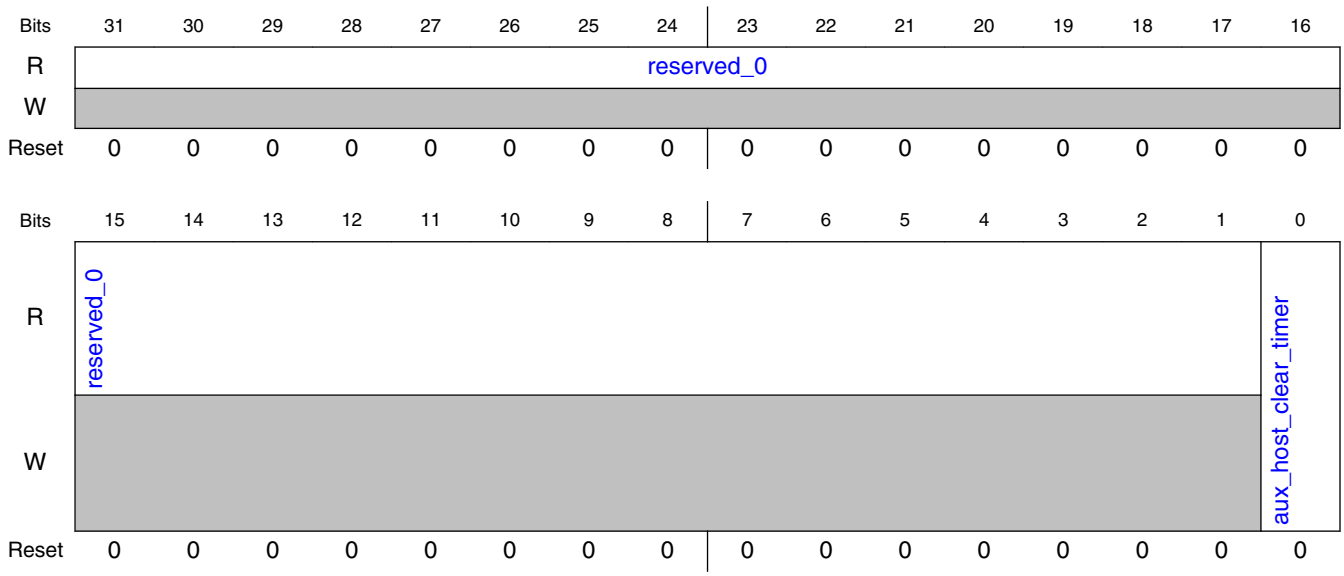
Field	Function
31-1 reserved_0	
0 aux_host_stop_timer	Stop timer operation

13.4.10.1.175 Clear timer operation (DP_AUX_TIMER_CLEAR)

13.4.10.1.175.1 Offset

Register	Offset
DP_AUX_TIMER_CLEAR	2820h

13.4.10.1.175.2 Diagram



13.4.10.1.175.3 Fields

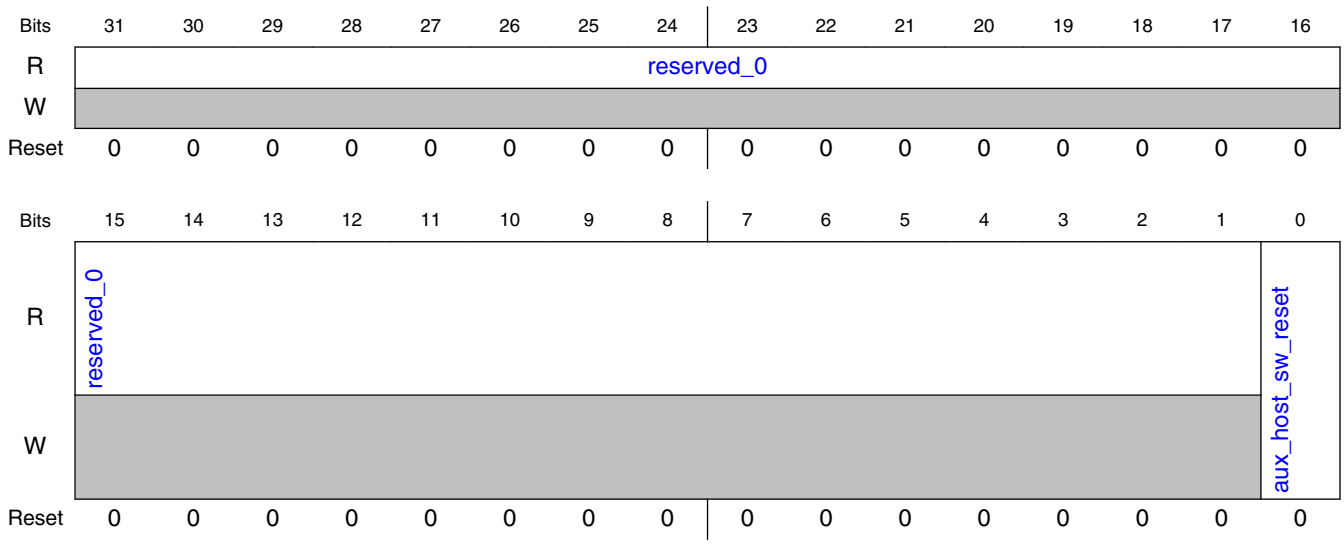
Field	Function
31-1 reserved_0	
0 aux_host_clear_timer	Stop timer operation. Stop timer operation. This bit is automaticaly cleared when operation in completed.

13.4.10.1.176 Soft reset of the DP_AUX (DP_AUX_RESET_SW)

13.4.10.1.176.1 Offset

Register	Offset
DP_AUX_RESET_SW	2824h

13.4.10.1.176.2 Diagram



13.4.10.1.176.3 Fields

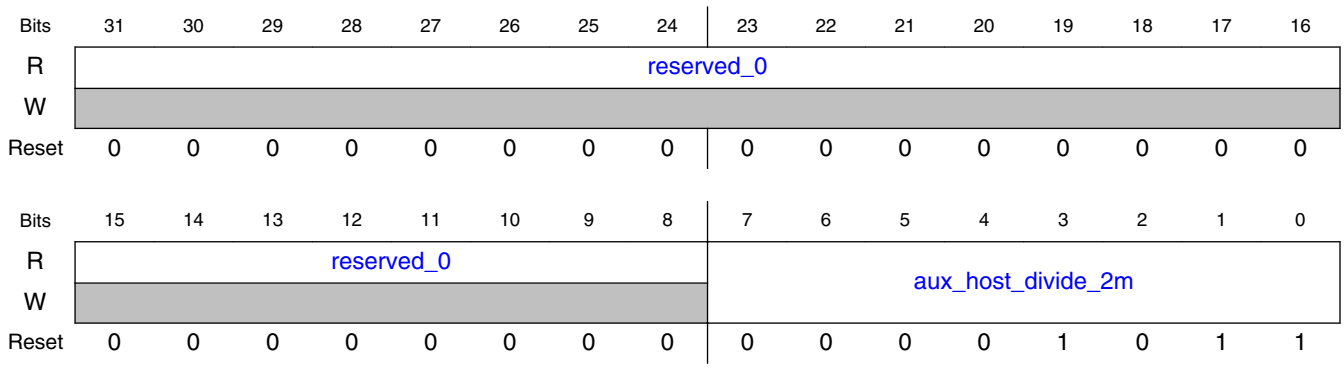
Field	Function
31-1 reserved_0	
0 aux_host_sw_reset	Reset all DP_AUX state machines and clear all the status bits. Reset all DP_AUX state machines and clear all the status bits. The registers value remains. (S/W reset). This bit is automaticaly cleared when operation in completed.

13.4.10.1.177 SYS_CLK and 2 MHz clock ratio (DP_AUX_DIVIDE_2M)

13.4.10.1.177.1 Offset

Register	Offset
DP_AUX_DIVIDE_2M	2828h

13.4.10.1.177.2 Diagram



13.4.10.1.177.3 Fields

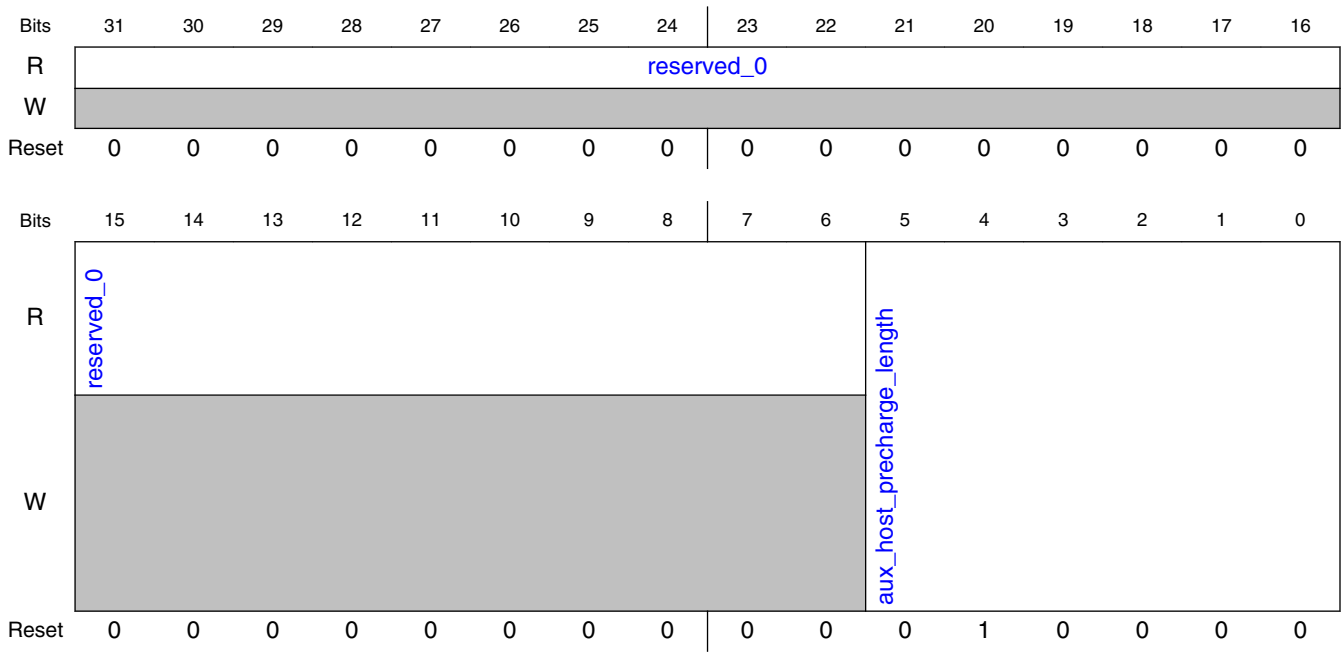
Field	Function
31-8 reserved_0	
7-0 aux_host_divide_2m	The ratio between sys_clk and 2MHz, [(sys_clk frequency/2MHz) - 1], for 25MHz sys_clk the value is 11. The ratio between sys_clk and 2MHz, [(sys_clk frequency/2MHz) - 1], for 25MHz sys_clk the value is 11. This register is used by AUX_TX for generating the AUX_TX clock.

13.4.10.1.178 Pre charge field length (DP_AUX_TX_PREACHARGE_LENGTH)

13.4.10.1.178.1 Offset

Register	Offset
DP_AUX_TX_PREACHARGE_LENGTH	282Ch

13.4.10.1.178.2 Diagram



13.4.10.1.178.3 Fields

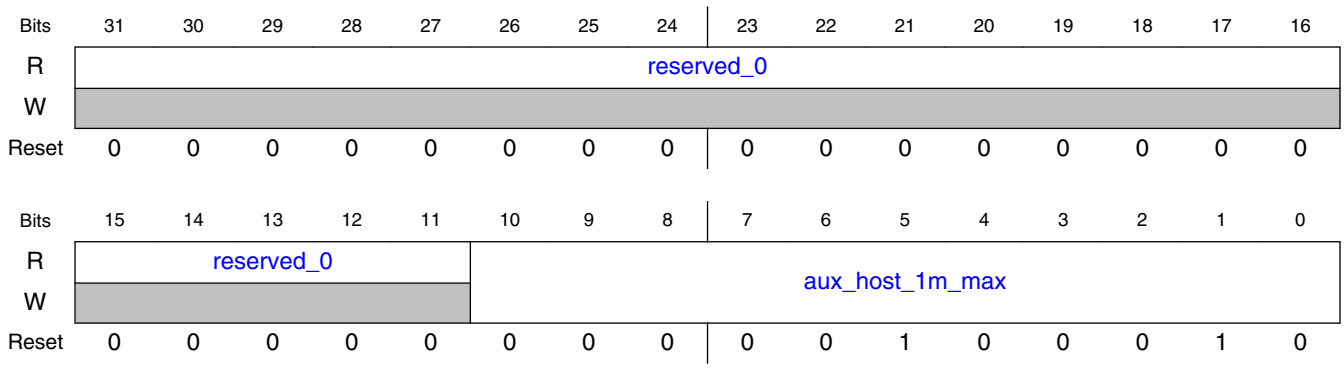
Field	Function
31-6 reserved_0	
5-0 aux_host_precharge_length	Length of pre charge field, standard definition is 10 to 16 bits/clocks

13.4.10.1.179 Maximum legal receiving frequency (DP_AUX_FREQUENCY_1M_MAX)

13.4.10.1.179.1 Offset

Register	Offset
DP_AUX_FREQUENCY_1M_MAX	2830h

13.4.10.1.179.2 Diagram



13.4.10.1.179.3 Fields

Field	Function
31-11 reserved_0	
10-0 aux_host_1m_max	The maximum legal frequency receiving from the line by the standard is 1. The maximum legal frequency receiving from the line by the standard is 1.25MHz.The calculation is:(1.25 MHz cycle time)/(sys_clk(-15%) cycle time) 800/46 =17

13.4.10.1.180 Minimum legal receiving frequency (DP_AUX_FREQUENCY_1M_MIN)

13.4.10.1.180.1 Offset

Register	Offset
DP_AUX_FREQUENCY_1M_MIN	2834h

13.4.10.1.180.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0					aux_host_1m_min										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1

13.4.10.1.180.3 Fields

Field	Function
31-11 reserved_0	
10-0 aux_host_1m_min	The minimum legal frequency receiving from the line by the standard is 0. The minimum legal frequency receiving from the line by the standard is 0.83MHz.The calculation is:(0.83 MHz cycle time)/(sys_clk(+15%) cycle time) 1200/34 =35

13.4.10.1.181 Minimum received preamble length (DP_AUX_RX_PRE_MIN)

13.4.10.1.181.1 Offset

Register	Offset
DP_AUX_RX_PRE_MIN	2838h

13.4.10.1.181.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0										aux_host_pre_min					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0

13.4.10.1.181.3 Fields

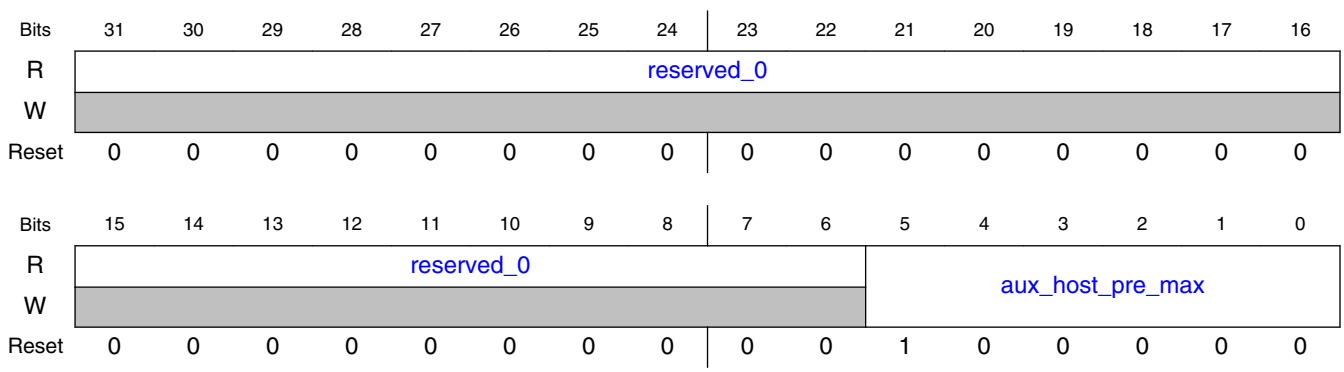
Field	Function
31-6 reserved_0	
5-0 aux_host_pre_min	Valid minimum length of preamble during receive. Valid minimum length of preamble during receive. The standard defines pre_min=26 The value of this register should be greater then the average_number_of_cycles defined in reg 0 (2 , 4 or 8)

13.4.10.1.182 Maximum received preamble length (DP_AUX_RX_PRE_MAX)

13.4.10.1.182.1 Offset

Register	Offset
DP_AUX_RX_PRE_MAX	283Ch

13.4.10.1.182.2 Diagram



13.4.10.1.182.3 Fields

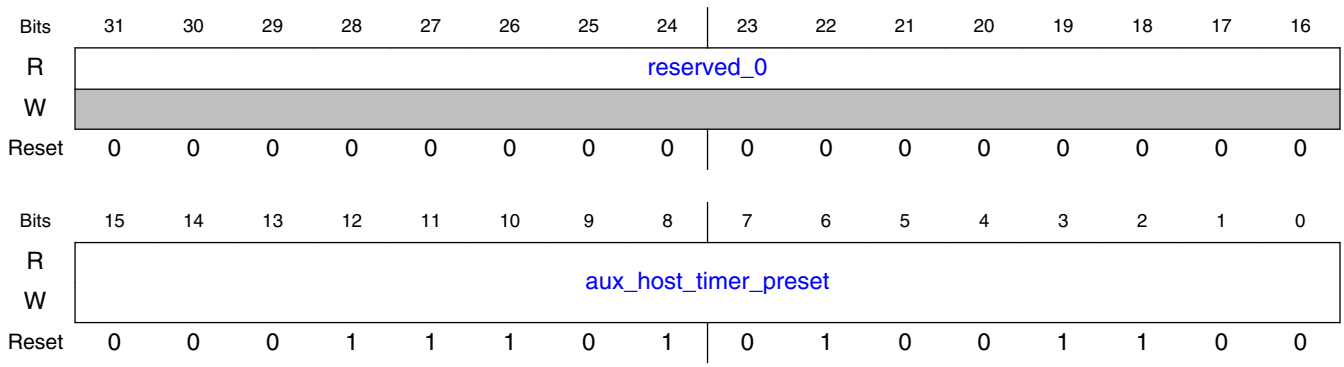
Field	Function
31-6 reserved_0	
5-0 aux_host_pre_max	Valid maximum length of preamble during receive. Valid maximum length of preamble during receive. The standard defines pre_max = 32

13.4.10.1.183 DP_AUX_MAIN start value (DP_AUX_TIMER_PRESET)

13.4.10.1.183.1 Offset

Register	Offset
DP_AUX_TIMER_PRESET	2840h

13.4.10.1.183.2 Diagram



13.4.10.1.183.3 Fields

Field	Function
31-16 reserved_0	
15-0 aux_host_timer_preset	The preset value of the timer in DP_IN mode. The preset value of the timer in DP_IN mode. With sys_clk= 25MHz the Timer can measure up to ~2500 micro seconds. The defaults value is 300 microseconds (1D4c in Hex)

13.4.10.1.184 Transmit pattern (DP_AUX_NACK_FORMAT)

13.4.10.1.184.1 Offset

Register	Offset
DP_AUX_NACK_FORMAT	2844h

13.4.10.1.184.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								aux_host_nack_format							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

13.4.10.1.184.3 Fields

Field	Function
31-8 reserved_0	
7-0 aux_host_nack_format	Nack or defer pattern for transmit (00100000 for defer, 00010000 for nack)

13.4.10.1.185 Mailbox write data (DP_AUX_TX_DATA)

13.4.10.1.185.1 Offset

Register	Offset
DP_AUX_TX_DATA	2848h

13.4.10.1.185.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0						mailbox_tx_data									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.185.3 Fields

Field	Function
31-10 reserved_0	
9-0 mailbox_tx_data	TX data byte written to the mailbox. TX data byte written to the mailbox. It is written 20 times and directly transferred into TX mailbox. First 8 bits are regular data. When the first data is transferred into mailbox mailbox_tx_data[8] (frame start) is set to 1. When the last data is transferred into mailbox, mailbox_tx_data[9] (frame_end) is set to 1.

13.4.10.1.186 Mailbox read data (DP_AUX_RX_DATA)

13.4.10.1.186.1 Offset

Register	Offset
DP_AUX_RX_DATA	284Ch

13.4.10.1.186.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0						mailbox_rx_data									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.186.3 Fields

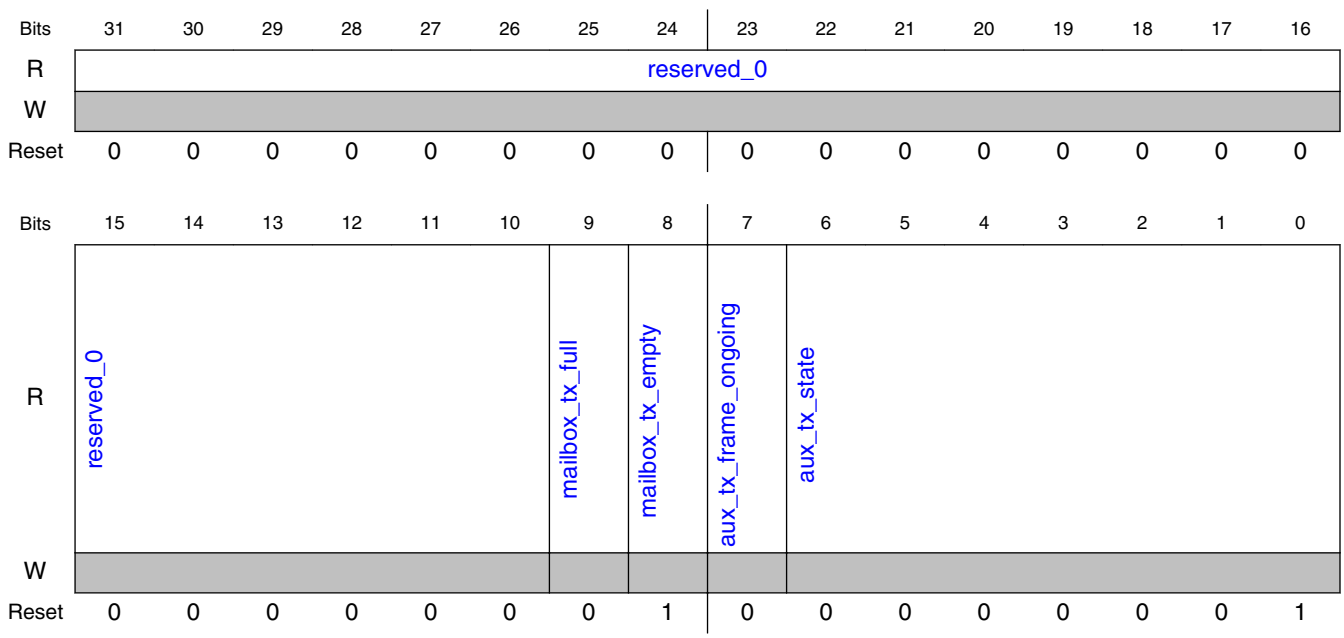
Field	Function
31-10 reserved_0	
9-0 mailbox_rx_data	Read data from the mailbox. Read data from the mailbox. Whenever read to this register occurs, aux_mailbox_read to RX Mailbox shall be asserted for one clock cycle.

13.4.10.1.187 AUX_TX status (DP_AUX_TX_STATUS)

13.4.10.1.187.1 Offset

Register	Offset
DP_AUX_TX_STATUS	2850h

13.4.10.1.187.2 Diagram



13.4.10.1.187.3 Fields

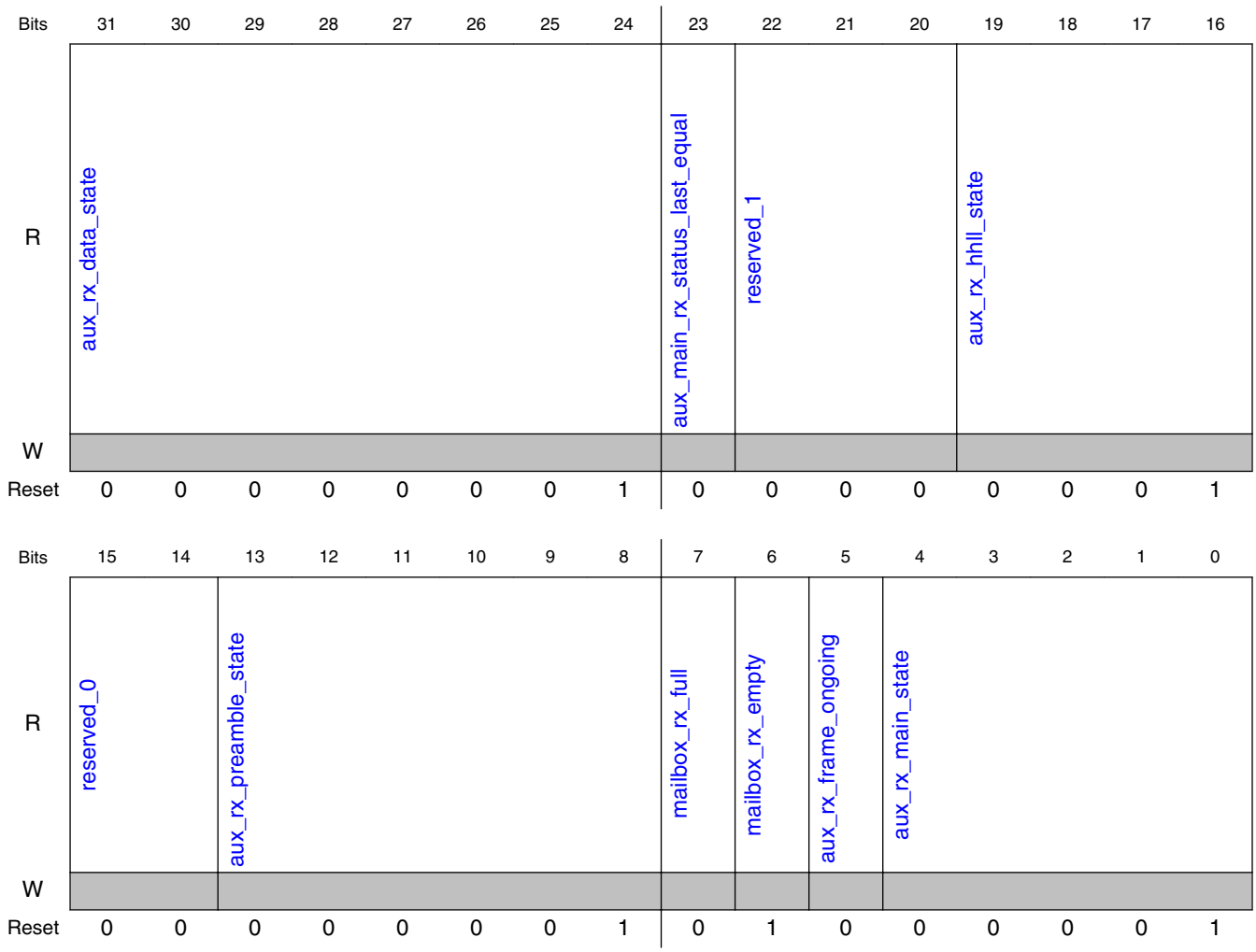
Field	Function
31-10 reserved_0	
9 mailbox_tx_full	Mailbox TX full flag
8 mailbox_tx_empty	Mailbox TX empty flag
7 aux_tx_frame_ongoing	Frame transmission status
6-0 aux_tx_state	Aux_tx state machine register

13.4.10.1.188 AUX_RX status (DP_AUX_RX_STATUS)

13.4.10.1.188.1 Offset

Register	Offset
DP_AUX_RX_STATUS	2854h

13.4.10.1.188.2 Diagram



13.4.10.1.188.3 Fields

Field	Function
31-24 aux_rx_data_state	AUX_RX SM state
23 aux_main_rx_status_last_equal	The receive transaction is equal to the previous transaction. The receive transaction is equal to the previous transaction. (THIS IS NOT ERROR BIT !!)
22-20 reserved_1	
19-16 aux_rx_hhll_state	AUX_RX hhll state machine register

Table continues on the next page...

Field	Function
15-14 reserved_0	
13-8 aux_rx_preamble_state	AUX_RX preamble state machine register
7 mailbox_rx_full	Mailbox RX full flag
6 mailbox_rx_empty	Mailbox RX empty flag
5 aux_rx_frame_ongoing	Frame reception status
4-0 aux_rx_main_state	AUX_RX main state machine register

13.4.10.1.189 RX counter status (DP_AUX_RX_CYCLE_COUNTER)

13.4.10.1.189.1 Offset

Register	Offset
DP_AUX_RX_CYCLE_COUNTER	2858h

13.4.10.1.189.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0					aux_rx_cycle_counter										
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.189.3 Fields

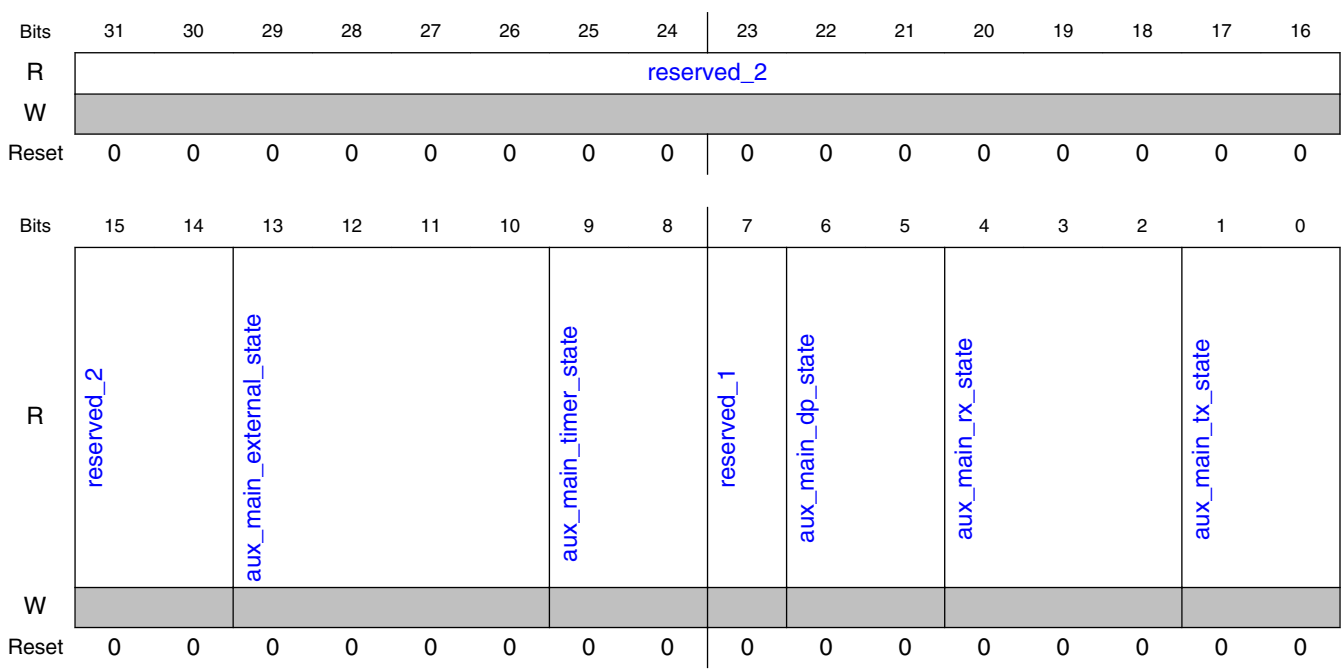
Field	Function
31-11 reserved_0	
10-0 aux_rx_cycle_counter	Count system clocks from last change in the auxiliary line input

13.4.10.1.190 DP_AUX MAIN State Machines status (DP_AUX_MAIN_STATES)

13.4.10.1.190.1 Offset

Register	Offset
DP_AUX_MAIN_STATES	285Ch

13.4.10.1.190.2 Diagram



13.4.10.1.190.3 Fields

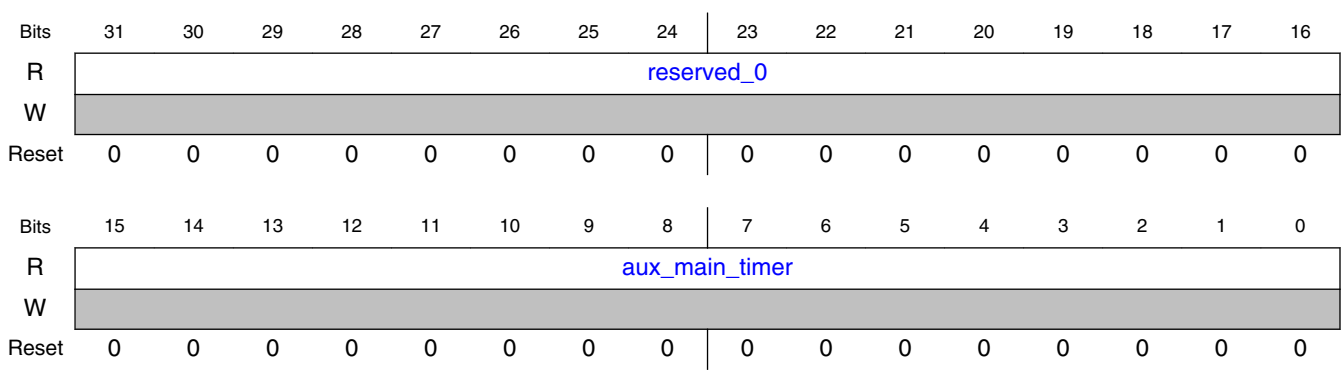
Field	Function
31-14 reserved_2	
13-10 aux_main_external_state	AUX_MAIN external state machine register
9-8 aux_main_timer_state	AUX_MAIN timer state machine
7 reserved_1	
6-5 aux_main_dp_state	AUX_MAIN dp state machine
4-2 aux_main_rx_state	AUX_MAIN rx state machine
1-0 aux_main_tx_state	AUX_MAIN tx state machine

13.4.10.1.191 DP_AUX MAIN timer status (DP_AUX_MAIN_TIMER)

13.4.10.1.191.1 Offset

Register	Offset
DP_AUX_MAIN_TIMER	2860h

13.4.10.1.191.2 Diagram



13.4.10.1.191.3 Fields

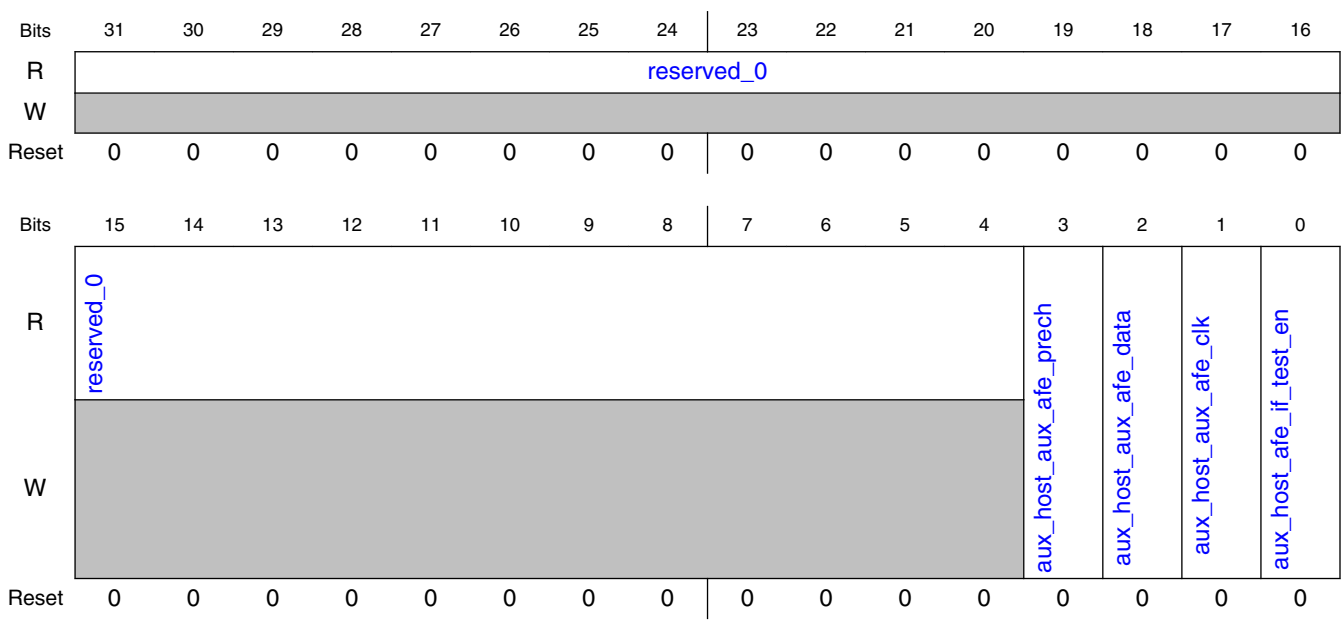
Field	Function
31-16 reserved_0	
15-0 aux_main_timer	DP_AUX MAIN timer status

13.4.10.1.192 Test mode configuration (DP_AUX_AFE_OUT)

13.4.10.1.192.1 Offset

Register	Offset
DP_AUX_AFE_OUT	2864h

13.4.10.1.192.2 Diagram



13.4.10.1.192.3 Fields

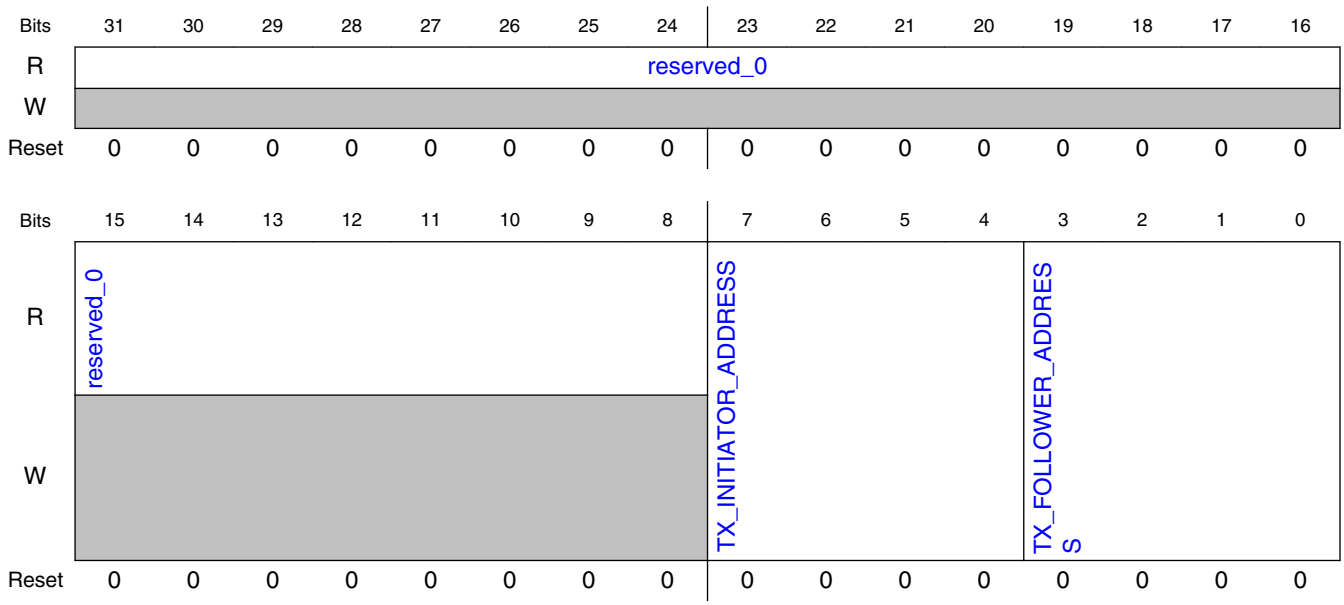
Field	Function
31-4 reserved_0	
3 aux_host_aux_afe_prech	Drive the aux_data_prech output to the AFE when aux_host_afe_if_test_en (bit 0) is set
2 aux_host_aux_afe_data	Drive the aux_data_out output to the AFE when aux_host_afe_if_test_en (bit 0) is set
1 aux_host_aux_afe_clk	Drive the aux_clk_out output to the AFE when aux_host_afe_if_test_en (bit 0) is set
0 aux_host_afe_if_test_en	TESTER mode enable. TESTER mode enable. Give the TESTER direct interface to the AFE_AUX.

13.4.10.1.193 CEC TX message header (TX_MSG_HEADER)

13.4.10.1.193.1 Offset

Register	Offset
TX_MSG_HEADER	3800h

13.4.10.1.193.2 Diagram



13.4.10.1.193.3 Fields

Field	Function
31-8 reserved_0	
7-4 TX_INITIATOR_ADDRESS	The 4 bit logical address of the initiator
3-0 TX_FOLLOWER_ADDRESS	The 4 bit logical address of the follower

13.4.10.1.194 CEC TX message opcode (TX_MSG_OPCODE)

13.4.10.1.194.1 Offset

Register	Offset
TX_MSG_OPCODE	3804h

13.4.10.1.194.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OPCODE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.194.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OPCODE	The opcode of the current message

13.4.10.1.195 CEC TX message operand - byte 1 (TX_MSG_OP1)

13.4.10.1.195.1 Offset

Register	Offset
TX_MSG_OP1	3808h

13.4.10.1.195.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.195.3 Fields

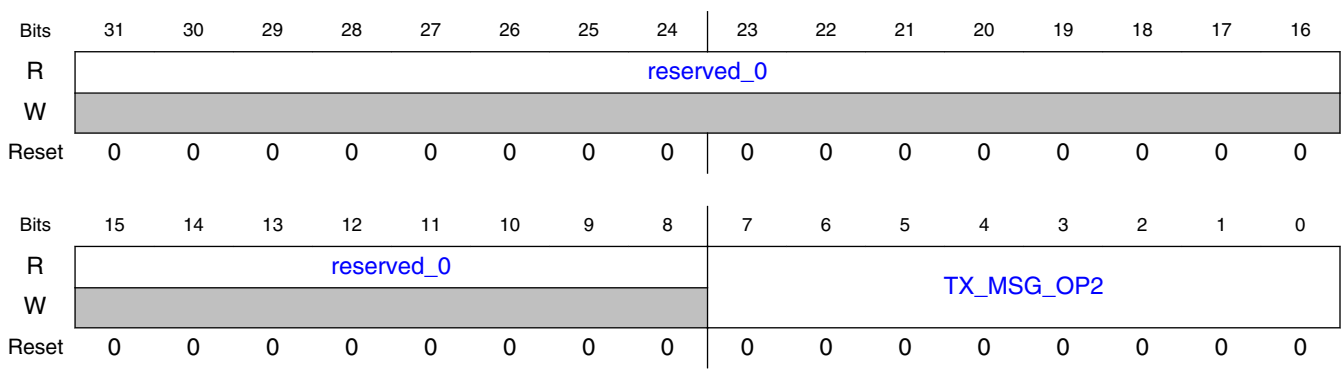
Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP1	The first operand required by the current opcode. The first operand required by the current opcode.

13.4.10.1.196 CEC TX message operand - byte 2 (TX_MSG_OP2)

13.4.10.1.196.1 Offset

Register	Offset
TX_MSG_OP2	380Ch

13.4.10.1.196.2 Diagram



13.4.10.1.196.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP2	The second operand required by the current opcode. The second operand required by the current opcode.

13.4.10.1.197 CEC TX message operand - byte 3 (TX_MSG_OP3)

13.4.10.1.197.1 Offset

Register	Offset
TX_MSG_OP3	3810h

13.4.10.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.197.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP3	The third operand required by the current opcode. The third operand required by the current opcode.

13.4.10.1.198 CEC TX message operand - byte 4 (TX_MSG_OP4)

13.4.10.1.198.1 Offset

Register	Offset
TX_MSG_OP4	3814h

13.4.10.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.198.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP4	The fourth operand required by the current opcode. The fourth operand required by the current opcode.

13.4.10.1.199 CEC TX message operand - byte 5 (TX_MSG_OP5)

13.4.10.1.199.1 Offset

Register	Offset
TX_MSG_OP5	3818h

13.4.10.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.199.3 Fields

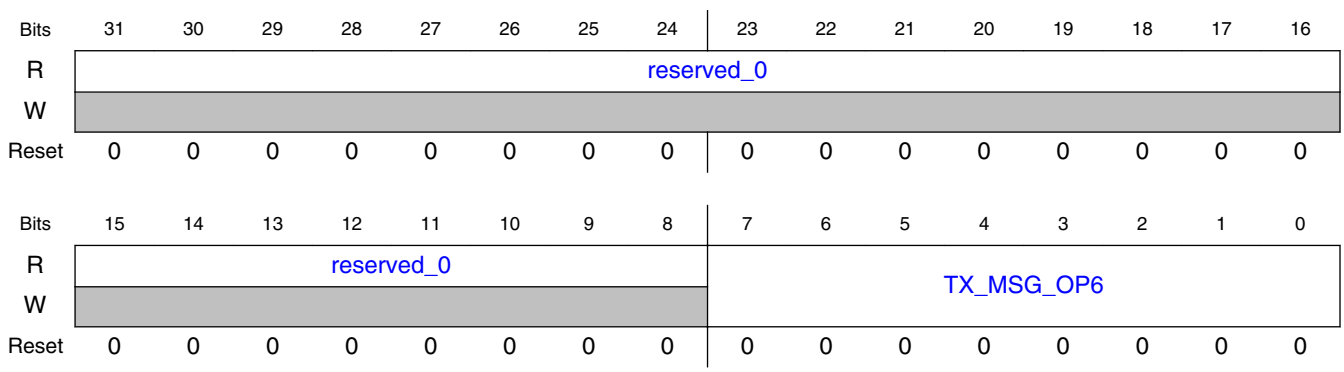
Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP5	The fifth operand required by the current opcode. The fifth operand required by the current opcode.

13.4.10.1.200 CEC TX message operand - byte 6 (TX_MSG_OP6)

13.4.10.1.200.1 Offset

Register	Offset
TX_MSG_OP6	381Ch

13.4.10.1.200.2 Diagram



13.4.10.1.200.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP6	The sixth operand required by the current opcode. The sixth operand required by the current opcode.

13.4.10.1.201 CEC TX message operand - byte 7 (TX_MSG_OP7)

13.4.10.1.201.1 Offset

Register	Offset
TX_MSG_OP7	3820h

13.4.10.1.201.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.201.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP7	The seventh operand required by the current opcode. The seventh operand required by the current opcode.

13.4.10.1.202 CEC TX message operand - byte 8 (TX_MSG_OP8)

13.4.10.1.202.1 Offset

Register	Offset
TX_MSG_OP8	3824h

13.4.10.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.202.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP8	The eighth operand required by the current opcode. The eighth operand required by the current opcode.

13.4.10.1.203 CEC TX message operand - byte 9 (TX_MSG_OP9)

13.4.10.1.203.1 Offset

Register	Offset
TX_MSG_OP9	3828h

13.4.10.1.203.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP9							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.203.3 Fields

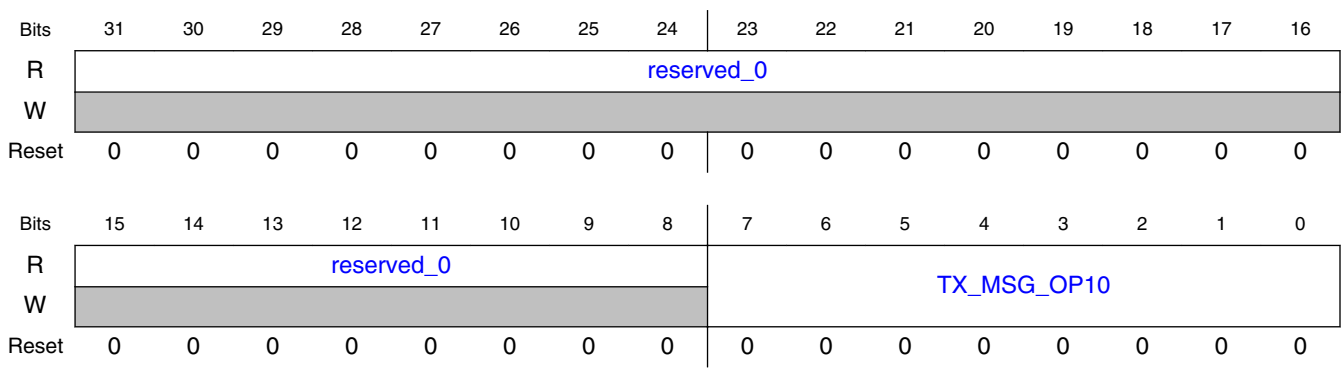
Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP9	The nineth operand required by the current opcode. The nineth operand required by the current opcode.

13.4.10.1.204 CEC TX message operand - byte 10 (TX_MSG_OP10)

13.4.10.1.204.1 Offset

Register	Offset
TX_MSG_OP10	382Ch

13.4.10.1.204.2 Diagram



13.4.10.1.204.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP10	The tenth operand required by the current opcode. The tenth operand required by the current opcode.

13.4.10.1.205 CEC TX message operand - byte 11 (TX_MSG_OP11)

13.4.10.1.205.1 Offset

Register	Offset
TX_MSG_OP11	3830h

13.4.10.1.205.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP11							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.205.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP11	The eleventh operand required by the current opcode. The eleventh operand required by the current opcode.

13.4.10.1.206 CEC TX message operand - byte 12 (TX_MSG_OP12)

13.4.10.1.206.1 Offset

Register	Offset
TX_MSG_OP12	3834h

13.4.10.1.206.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP12							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.206.3 Fields

Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP12	The twelvth operand required by the current opcode. The twelvth operand required by the current opcode.

13.4.10.1.207 CEC TX message operand - byte 13 (TX_MSG_OP13)

13.4.10.1.207.1 Offset

Register	Offset
TX_MSG_OP13	3838h

13.4.10.1.207.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								TX_MSG_OP13							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.207.3 Fields

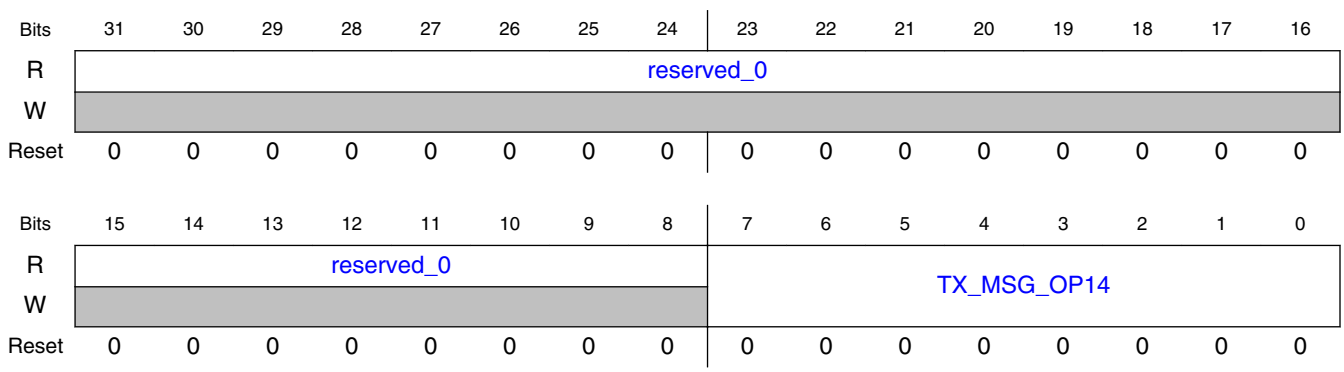
Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP13	The thirteenth operand required by the current opcode. The thirteenth operand required by the current opcode.

13.4.10.1.208 CEC TX message operand - byte 14 (TX_MSG_OP14)

13.4.10.1.208.1 Offset

Register	Offset
TX_MSG_OP14	383Ch

13.4.10.1.208.2 Diagram



13.4.10.1.208.3 Fields

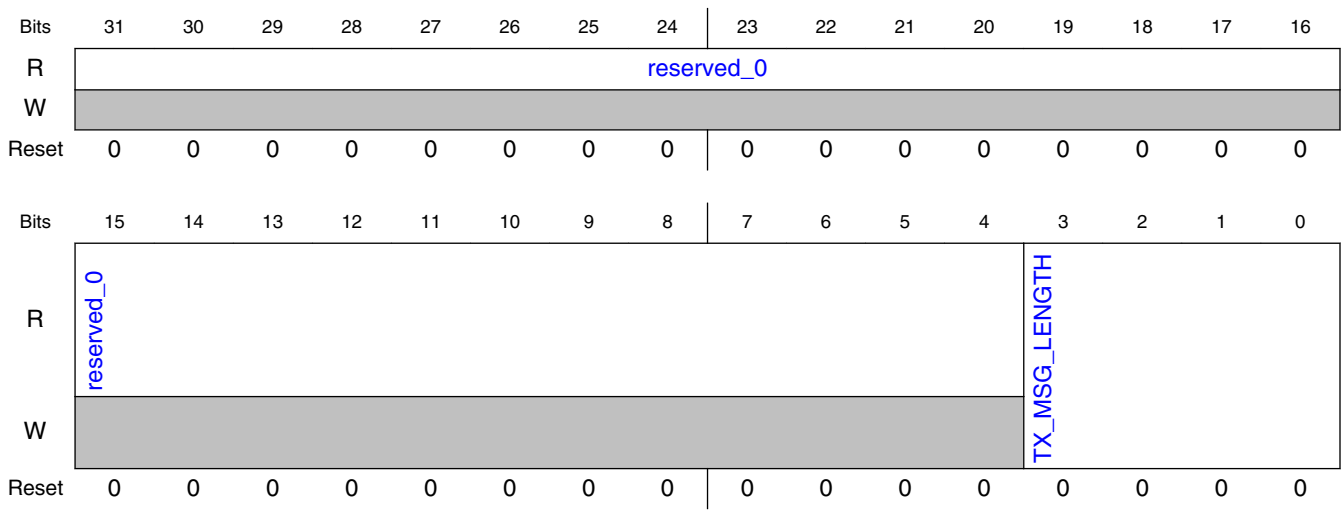
Field	Function
31-8 reserved_0	
7-0 TX_MSG_OP14	The fourteenth operand required by the current opcode. The fourteenth operand required by the current opcode.

13.4.10.1.209 CEC TX message length (TX_MSG_LENGTH)

13.4.10.1.209.1 Offset

Register	Offset
TX_MSG_LENGTH	3840h

13.4.10.1.209.2 Diagram



13.4.10.1.209.3 Fields

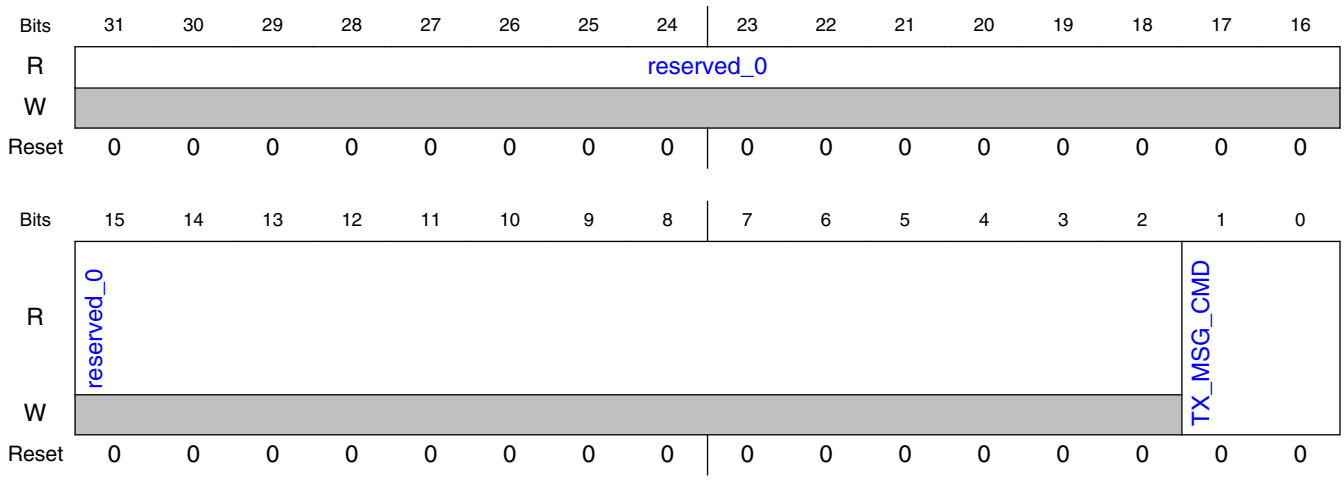
Field	Function
31-4 reserved_0	
3-0 TX_MSG_LENGTH	The number of bytes in the current message. The number of bytes in the current message. Zero means only header.

13.4.10.1.210 CEC TX message command (TX_MSG_CMD)

13.4.10.1.210.1 Offset

Register	Offset
TX_MSG_CMD	3844h

13.4.10.1.210.2 Diagram



13.4.10.1.210.3 Fields

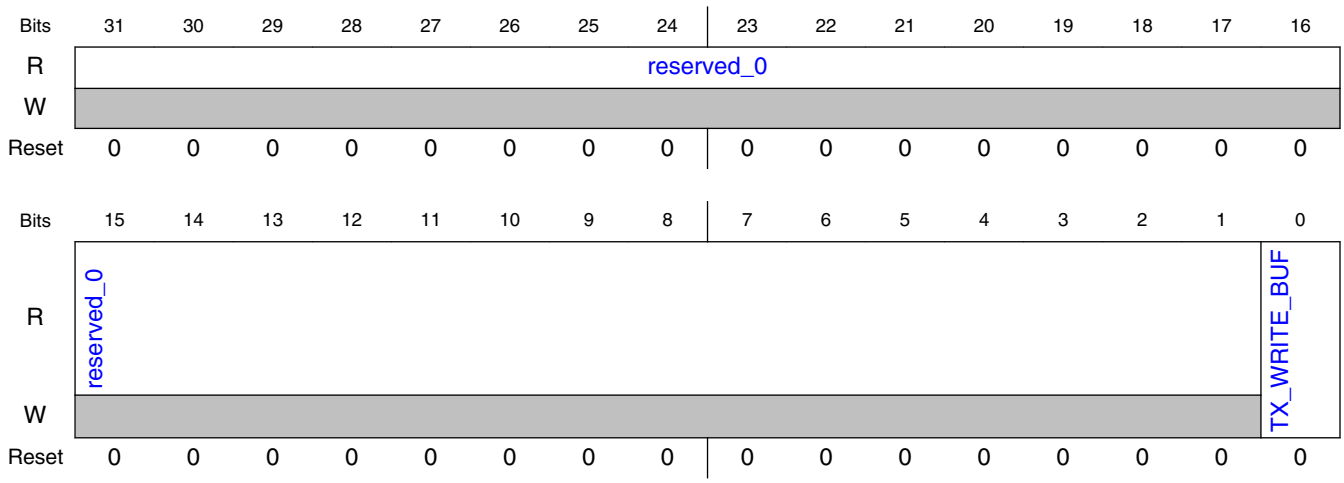
Field	Function
31-2 reserved_0	
1-0 TX_MSG_CMD	00: No transaction, 01: transmit earliest message in buffer, 10: abort transmitting earliest message, 11: overwrite earliest message and transmit next.

13.4.10.1.211 CEC TX buffer write enable (TX_WRITE_BUF)

13.4.10.1.211.1 Offset

Register	Offset
TX_WRITE_BUF	3848h

13.4.10.1.211.2 Diagram



13.4.10.1.211.3 Fields

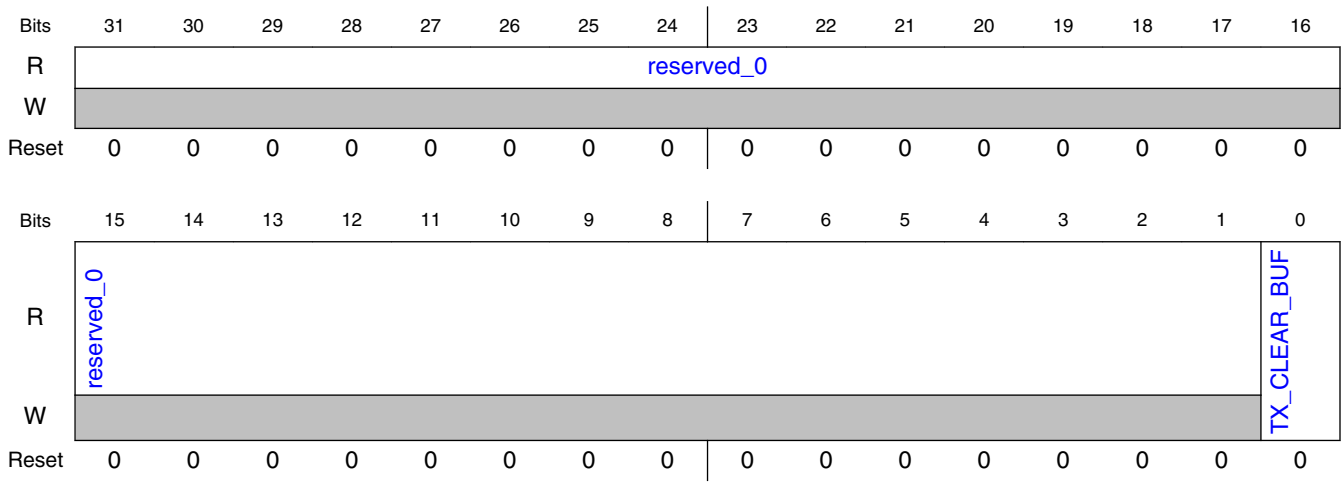
Field	Function
31-1 reserved_0	
0 TX_WRITE_BUF	Write enable (from host). Write enable (from host). -- Used only if BUFDEPTH > 1

13.4.10.1.212 CEC TX buffer clear (TX_CLEAR_BUF)

13.4.10.1.212.1 Offset

Register	Offset
TX_CLEAR_BUF	384Ch

13.4.10.1.212.2 Diagram



13.4.10.1.212.3 Fields

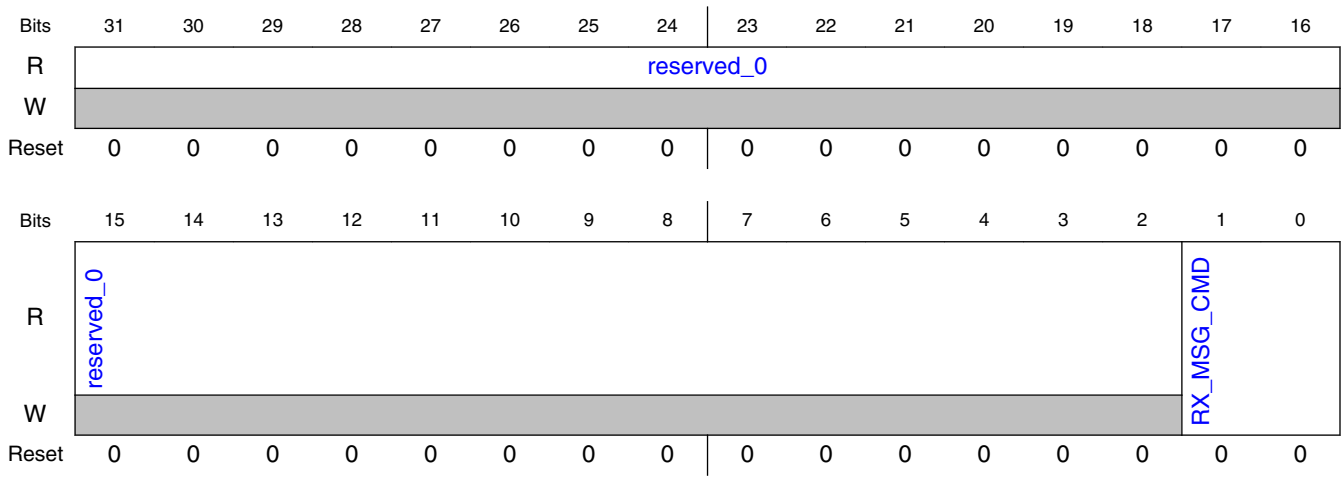
Field	Function
31-1 reserved_0	
0 TX_CLEAR_BUF	Clear all messages in Tx buf. Clear all messages in Tx buf. (from host). -- Used only if BUFDEPTH > 1.

13.4.10.1.213 CEC RX message command (RX_MSG_CMD)

13.4.10.1.213.1 Offset

Register	Offset
RX_MSG_CMD	3850h

13.4.10.1.213.2 Diagram



13.4.10.1.213.3 Fields

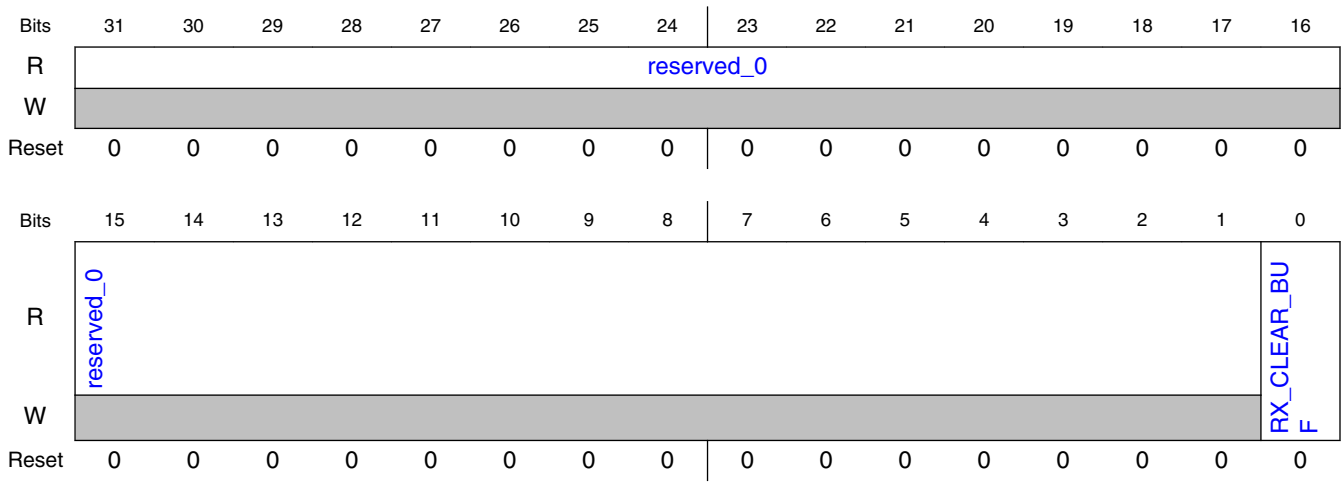
Field	Function
31-2 reserved_0	
1-0 RX_MSG_CMD	00: No transaction, 01: read earliest message in buffer, 10: disable receiving latest message, 11: clear earliest message and read next.

13.4.10.1.214 CEC RX buffer clear (RX_CLEAR_BUF)

13.4.10.1.214.1 Offset

Register	Offset
RX_CLEAR_BUF	3854h

13.4.10.1.214.2 Diagram



13.4.10.1.214.3 Fields

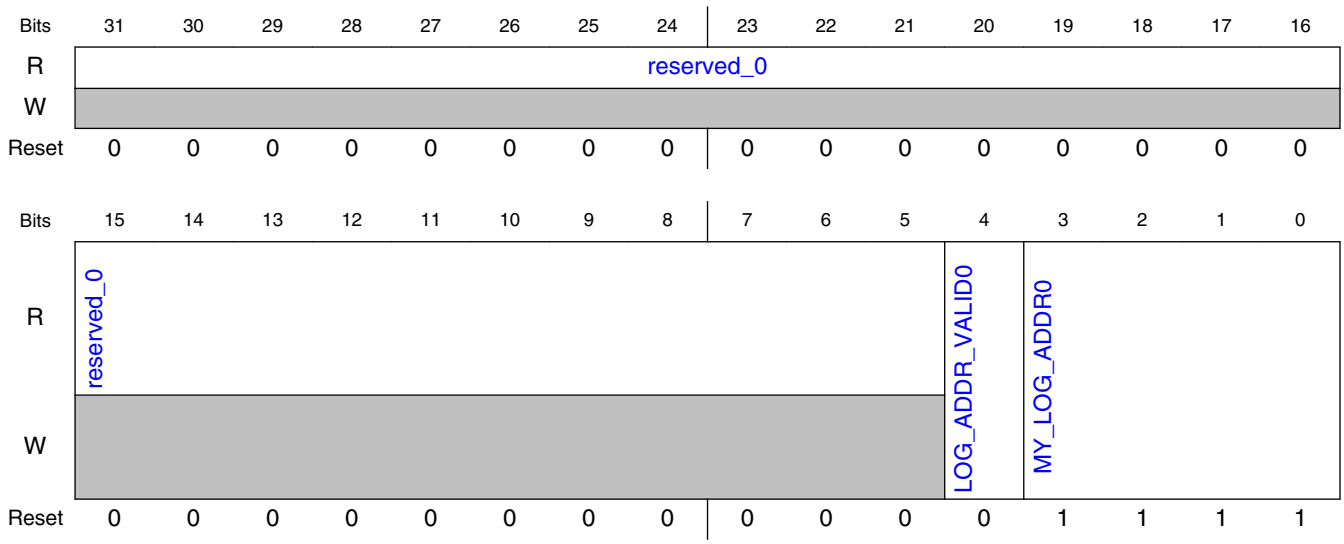
Field	Function
31-1 reserved_0	
0 RX_CLEAR_BUF	Clear all messages in Rx buf. Clear all messages in Rx buf.

13.4.10.1.215 CEC Logical Address 0 (LOGICAL_ADDRESS_LA0)

13.4.10.1.215.1 Offset

Register	Offset
LOGICAL_ADDRESS_LA0	3858h

13.4.10.1.215.2 Diagram



13.4.10.1.215.3 Fields

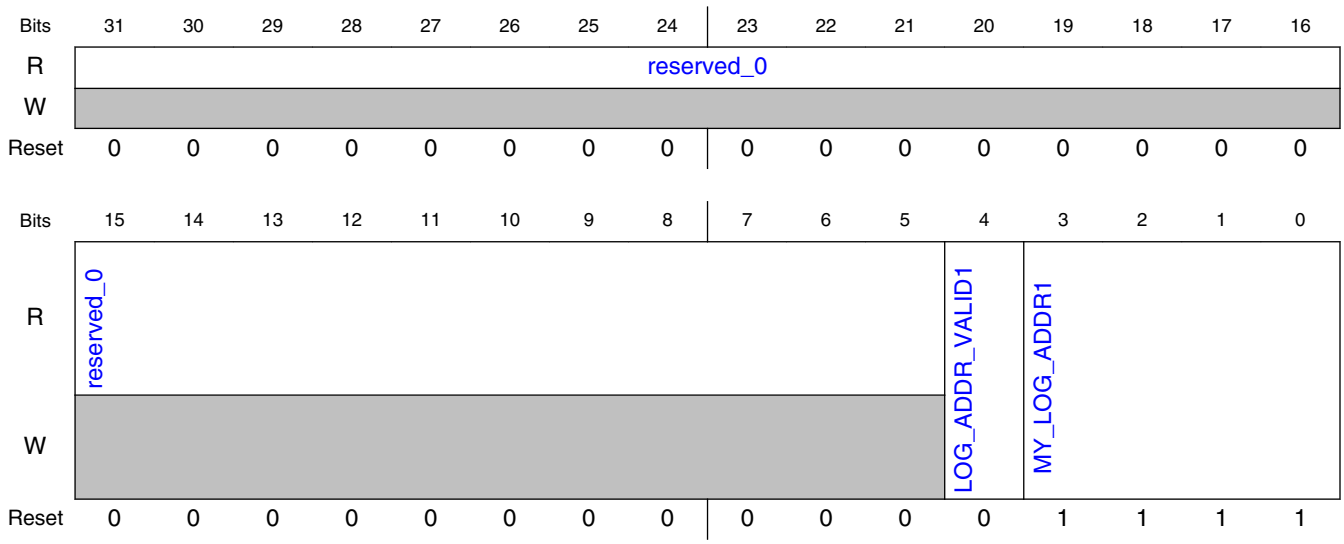
Field	Function
31-5 reserved_0	
4 LOG_ADDR_VALID0	Status of logical address ('1' valid, '0' not valid). Status of logical address ('1' valid, '0' not valid).
3-0 MY_LOG_ADDR0	The logical address for the first device. The logical address for the first device.

13.4.10.1.216 CEC Logical Address 1 (LOGICAL_ADDRESS_LA1)

13.4.10.1.216.1 Offset

Register	Offset
LOGICAL_ADDRESS_LA1	385Ch

13.4.10.1.216.2 Diagram



13.4.10.1.216.3 Fields

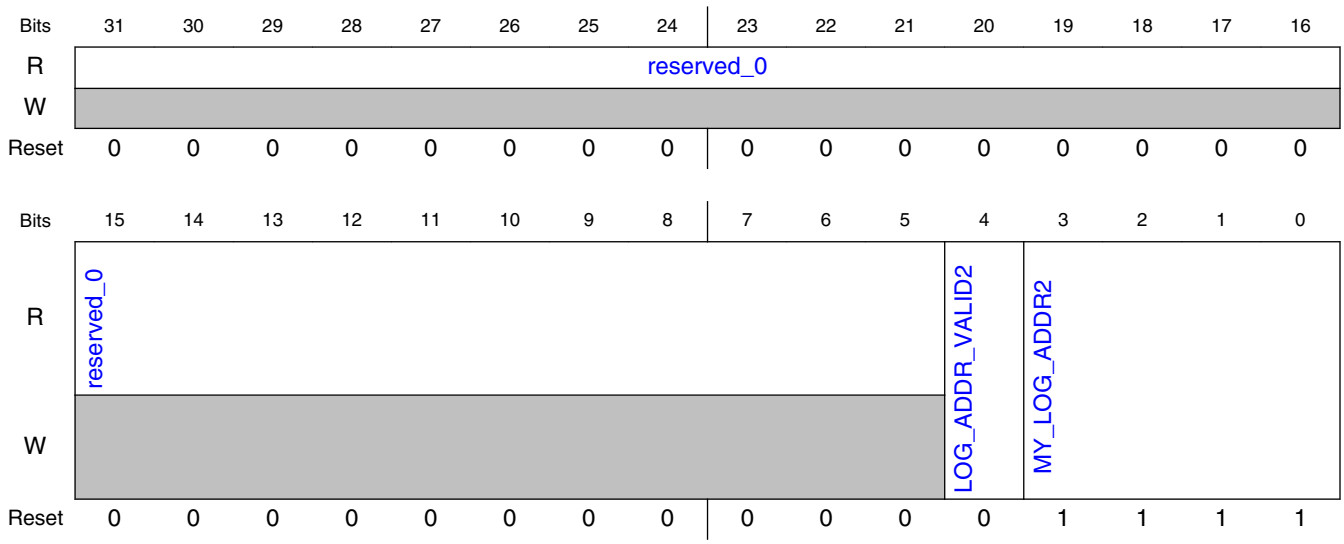
Field	Function
31-5 reserved_0	
4 LOG_ADDR_VALID1	Status of logical address ('1' valid, '0' not valid). Status of logical address ('1' valid, '0' not valid).
3-0 MY_LOG_ADDR1	The logical address for the second device. The logical address for the second device.

13.4.10.1.217 CEC Logical Address 2 (LOGICAL_ADDRESS_LA2)

13.4.10.1.217.1 Offset

Register	Offset
LOGICAL_ADDRESS_LA2	3860h

13.4.10.1.217.2 Diagram



13.4.10.1.217.3 Fields

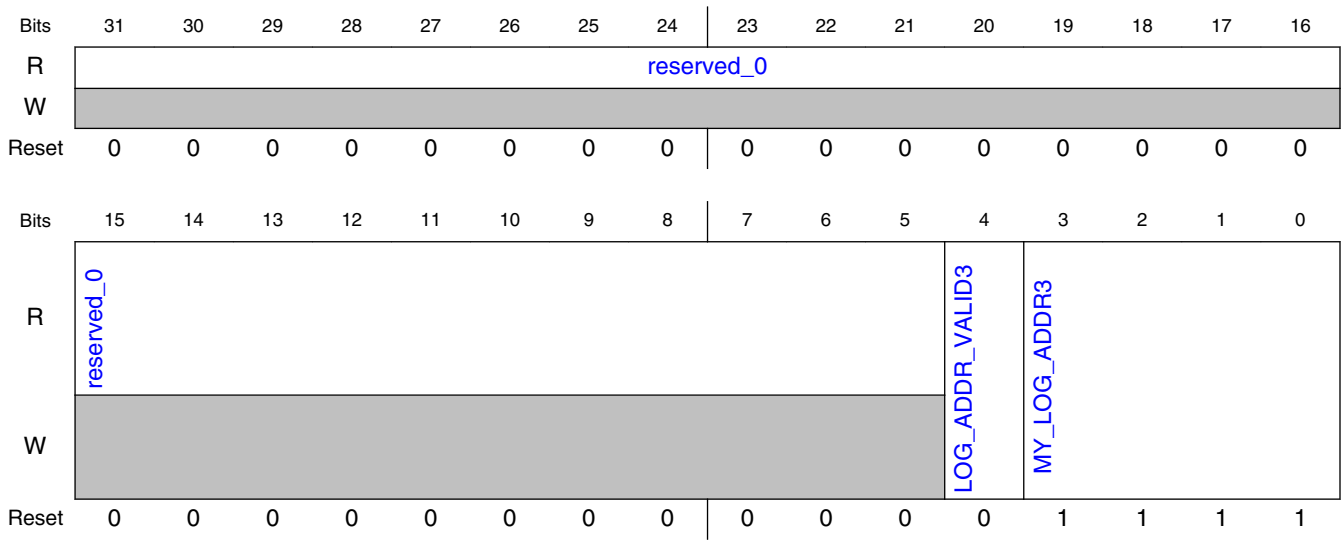
Field	Function
31-5 reserved_0	
4 LOG_ADDR_VALID2	Status of logical address ('1' valid, '0' not valid). Status of logical address ('1' valid, '0' not valid).
3-0 MY_LOG_ADDR2	The logical address for the third device. The logical address for the third device.

13.4.10.1.218 CEC Logical Address 3 (LOGICAL_ADDRESS_LA3)

13.4.10.1.218.1 Offset

Register	Offset
LOGICAL_ADDRESS_LA3	3864h

13.4.10.1.218.2 Diagram



13.4.10.1.218.3 Fields

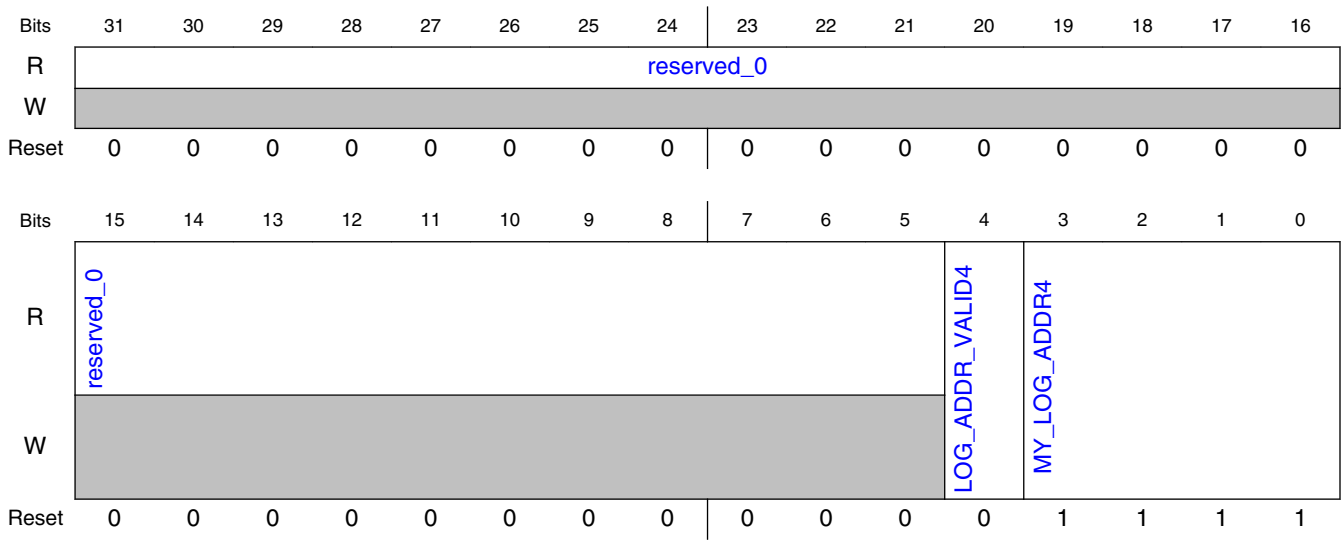
Field	Function
31-5 reserved_0	
4 LOG_ADDR_VALID3	Status of logical address ('1' valid, '0' not valid). Status of logical address ('1' valid, '0' not valid).
3-0 MY_LOG_ADDR3	The logical address for the fourth device. The logical address for the fourth device.

13.4.10.1.219 CEC Logical Address 4 (LOGICAL_ADDRESS_LA4)

13.4.10.1.219.1 Offset

Register	Offset
LOGICAL_ADDRESS_LA4	3868h

13.4.10.1.219.2 Diagram



13.4.10.1.219.3 Fields

Field	Function
31-5 reserved_0	
4 LOG_ADDR_VALID4	Status of logical address ('1' valid, '0' not valid). Status of logical address ('1' valid, '0' not valid).
3-0 MY_LOG_ADDR4	The logical address for the fifth device. The logical address for the fifth device.

13.4.10.1.220 CEC Clock Divider - MSB part (CLK_DIV_MSB)

13.4.10.1.220.1 Offset

Register	Offset
CLK_DIV_MSB	386Ch

13.4.10.1.220.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								CLK_DIV_MSB							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1

13.4.10.1.220.3 Fields

Field	Function
31-8 reserved_0	
7-0 CLK_DIV_MSB	Clock divider ratio. Clock divider ratio. It divides the system clock in order to have enable every 0.01ms. Default is #h3e7 for 100MHz clock.

13.4.10.1.221 CEC Clock Divider - LSB part (CLK_DIV_LSB)

13.4.10.1.221.1 Offset

Register	Offset
CLK_DIV_LSB	3870h

13.4.10.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								CLK_DIV_LSB							
W																
Reset	0	0	0	0	0	0	0	0	1	1	1	0	0	1	1	1

13.4.10.1.221.3 Fields

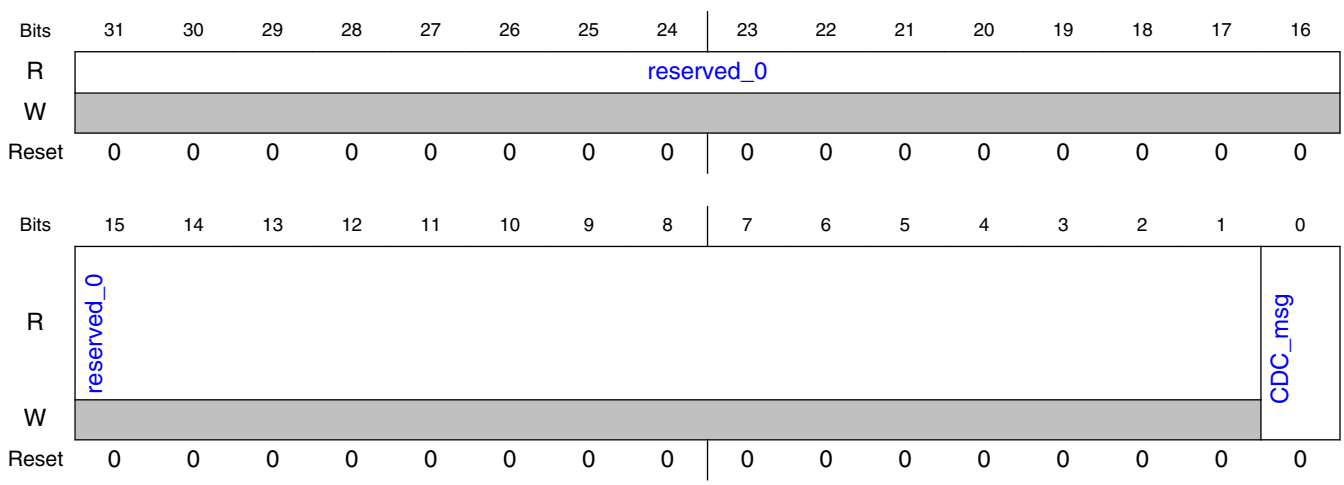
Field	Function
31-8 reserved_0	
7-0 CLK_DIV_LSB	

13.4.10.1.222 CDC mesage enable (cdc_msg)

13.4.10.1.222.1 Offset

Register	Offset
cdc_msg	387Ch

13.4.10.1.222.2 Diagram



13.4.10.1.222.3 Fields

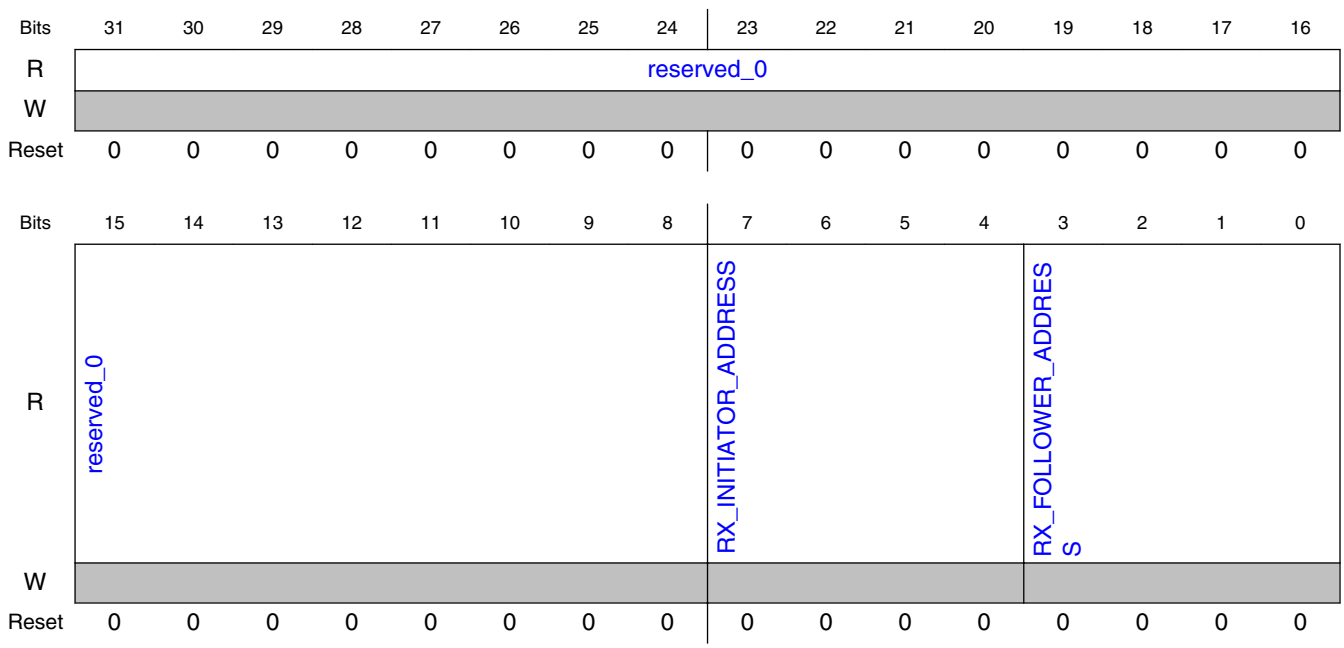
Field	Function
31-1 reserved_0	
0 CDC_msg	0: regular CEC message, 1: CDC message. 0: regular CEC message, 1: CDC message.

13.4.10.1.223 CEC RX message header (RX_MSG_DATA1)

13.4.10.1.223.1 Offset

Register	Offset
RX_MSG_DATA1	3900h

13.4.10.1.223.2 Diagram



13.4.10.1.223.3 Fields

Field	Function
31-8 reserved_0	
7-4 RX_INITIATOR_ADDRESS	
3-0 RX_FOLLOWER_ADDRESS	Header of the received message. Header of the received message.

13.4.10.1.224 CEC RX message opcode (RX_MSG_DATA2)

13.4.10.1.224.1 Offset

Register	Offset
RX_MSG_DATA2	3904h

13.4.10.1.224.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OPCODE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.224.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OPCODE	Opcode of the received message. Opcode of the received message.

13.4.10.1.225 CEC RX message operand 1 (RX_MSG_DATA3)

13.4.10.1.225.1 Offset

Register	Offset
RX_MSG_DATA3	3908h

13.4.10.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.225.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP1	First operand for the opcode of the received message. First operand for the opcode of the received message.

13.4.10.1.226 CEC RX message operand 2 (RX_MSG_DATA4)

13.4.10.1.226.1 Offset

Register	Offset
RX_MSG_DATA4	390Ch

13.4.10.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP2							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.226.3 Fields

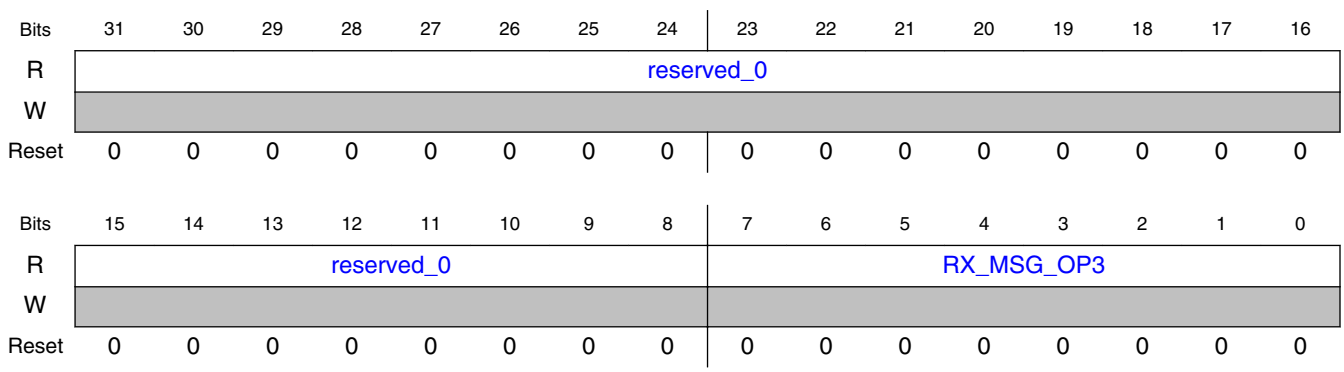
Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP2	Second operand for the opcode of the received message. Second operand for the opcode of the received message.

13.4.10.1.227 CEC RX message operand 3 (RX_MSG_DATA5)

13.4.10.1.227.1 Offset

Register	Offset
RX_MSG_DATA5	3910h

13.4.10.1.227.2 Diagram



13.4.10.1.227.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP3	Third operand for the opcode of the received message. Third operand for the opcode of the received message.

13.4.10.1.228 CEC RX message operand 4 (RX_MSG_DATA6)

13.4.10.1.228.1 Offset

Register	Offset
RX_MSG_DATA6	3914h

13.4.10.1.228.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP4							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.228.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP4	Fourth operand for the opcode of the received message. Fourth operand for the opcode of the received message.

13.4.10.1.229 CEC RX message operand 5 (RX_MSG_DATA7)

13.4.10.1.229.1 Offset

Register	Offset
RX_MSG_DATA7	3918h

13.4.10.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP5							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.229.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP5	Fifth operand for the opcode of the received message. Fifth operand for the opcode of the received message.

13.4.10.1.230 CEC RX message operand 6 (RX_MSG_DATA8)

13.4.10.1.230.1 Offset

Register	Offset
RX_MSG_DATA8	391Ch

13.4.10.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP6							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.230.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP6	Sixth operand for the opcode of the received message. Sixth operand for the opcode of the received message.

13.4.10.1.231 CEC RX message operand 7 (RX_MSG_DATA9)

13.4.10.1.231.1 Offset

Register	Offset
RX_MSG_DATA9	3920h

13.4.10.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP7							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.231.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP7	Seventh operand for the opcode of the received message. Seventh operand for the opcode of the received message.

13.4.10.1.232 CEC RX message operand 8 (RX_MSG_DATA10)

13.4.10.1.232.1 Offset

Register	Offset
RX_MSG_DATA10	3924h

13.4.10.1.232.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP8							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.232.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP8	Eighth operand for the opcode of the received message. Eighth operand for the opcode of the received message.

13.4.10.1.233 CEC RX message operand 9 (RX_MSG_DATA11)

13.4.10.1.233.1 Offset

Register	Offset
RX_MSG_DATA11	3928h

13.4.10.1.233.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP9							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.233.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP9	Ninth operand for the opcode of the received message. Ninth operand for the opcode of the received message.

13.4.10.1.234 CEC RX message operand 10 (RX_MSG_DATA12)

13.4.10.1.234.1 Offset

Register	Offset
RX_MSG_DATA12	392Ch

13.4.10.1.234.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP10							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.234.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP10	Tenth operand for the opcode of the received message. Tenth operand for the opcode of the received message.

13.4.10.1.235 CEC RX message operand 11 (RX_MSG_DATA13)

13.4.10.1.235.1 Offset

Register	Offset
RX_MSG_DATA13	3930h

13.4.10.1.235.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP11							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.235.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP11	The eleventh operand for the opcode of the received message. The eleventh operand for the opcode of the received message.

13.4.10.1.236 CEC RX message operand 12 (RX_MSG_DATA14)

13.4.10.1.236.1 Offset

Register	Offset
RX_MSG_DATA14	3934h

13.4.10.1.236.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP12							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.236.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP12	The twelfth operand for the opcode of the received message. The twelfth operand for the opcode of the received message.

13.4.10.1.237 CEC RX message operand 13 (RX_MSG_DATA15)

13.4.10.1.237.1 Offset

Register	Offset
RX_MSG_DATA15	3938h

13.4.10.1.237.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP13							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.237.3 Fields

Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP13	The thirteenth operand for the opcode of the received message. The thirteenth operand for the opcode of the received message.

13.4.10.1.238 CEC RX message operand 14 (RX_MSG_DATA16)

13.4.10.1.238.1 Offset

Register	Offset
RX_MSG_DATA16	393Ch

13.4.10.1.238.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								RX_MSG_OP14							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.238.3 Fields

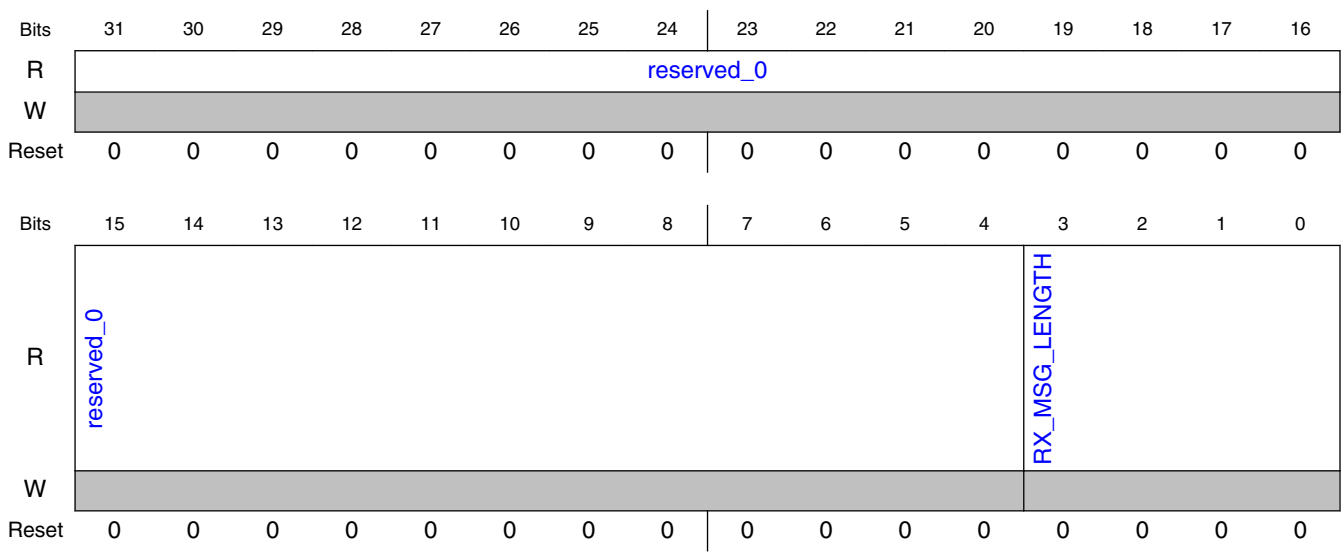
Field	Function
31-8 reserved_0	
7-0 RX_MSG_OP14	The fourteenth operand for the opcode of the received message. The fourteenth operand for the opcode of the received message.

13.4.10.1.239 CEC RX message length (RX_MSG_LENGTH)

13.4.10.1.239.1 Offset

Register	Offset
RX_MSG_LENGTH	3940h

13.4.10.1.239.2 Diagram



13.4.10.1.239.3 Fields

Field	Function
31-4 reserved_0	

Table continues on the next page...

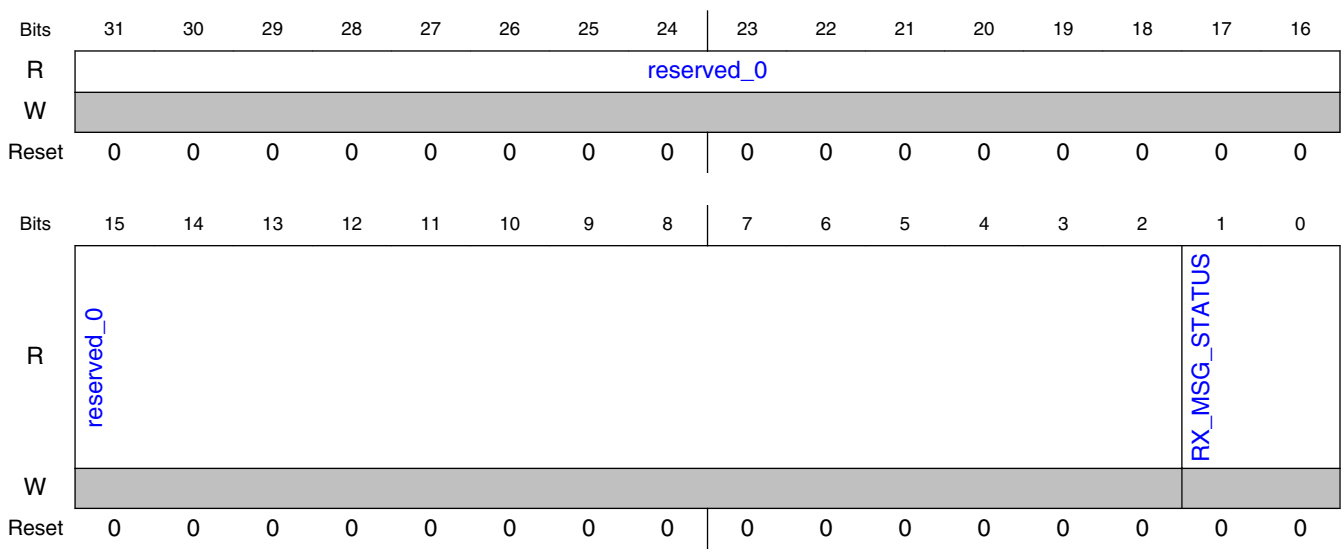
Field	Function
3-0 RX_MSG_LENGTH	The number of bytes in the received message. The number of bytes in the received message.

13.4.10.1.240 CEC RX status (RX_MSG_STATUS)

13.4.10.1.240.1 Offset

Register	Offset
RX_MSG_STATUS	3944h

13.4.10.1.240.2 Diagram



13.4.10.1.240.3 Fields

Field	Function
31-2 reserved_0	
1-0 RX_MSG_STATUS	00: no transaction, 01: busy, 10: message received successfully, 11: message received with error. 00: no transaction, 01: busy, 10: message received successfully, 11: message received with error.

13.4.10.1.241 CEC RX status - number of received messages (NUM_OF_MSG_RX_BUF)

13.4.10.1.241.1 Offset

Register	Offset
NUM_OF_MSG_RX_BUF	3948h

13.4.10.1.241.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0												RX_NUM_MSG			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.241.3 Fields

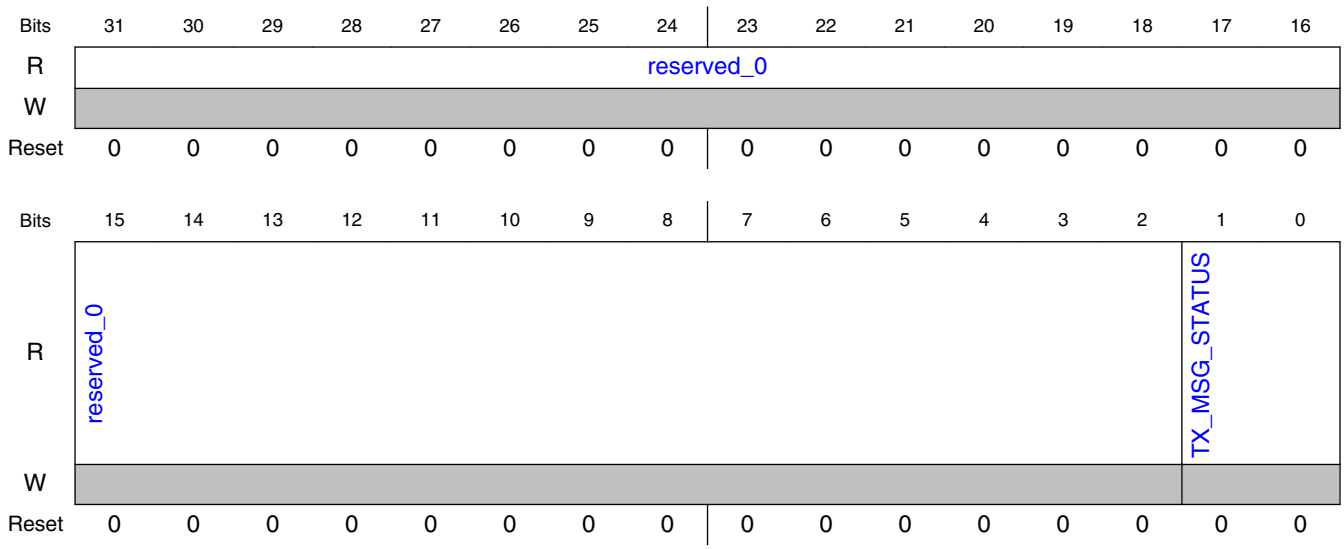
Field	Function
31-4 reserved_0	
3-0 RX_NUM_MSG	The number of received messages in Rx buffer waiting to be read. The number of received messages in Rx buffer waiting to be read.

13.4.10.1.242 CEC TX status (TX_MSG_STATUS)

13.4.10.1.242.1 Offset

Register	Offset
TX_MSG_STATUS	394Ch

13.4.10.1.242.2 Diagram



13.4.10.1.242.3 Fields

Field	Function
31-2 reserved_0	
1-0 TX_MSG_STATUS	00: no transaction, 01: busy, 10: message transmitted successfully, 11: message transmitted with error. 00: no transaction, 01: busy, 10: message transmitted successfully, 11: message transmitted with error.

13.4.10.1.243 CEC TX status - number of messages in TX buffer (NUMBER_OF_MSG_IN_TX_BUF)

13.4.10.1.243.1 Offset

Register	Offset
NUMBER_OF_MSG_IN_TX_BUF	3950h

13.4.10.1.243.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0												TX_NUM_MSG			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.243.3 Fields

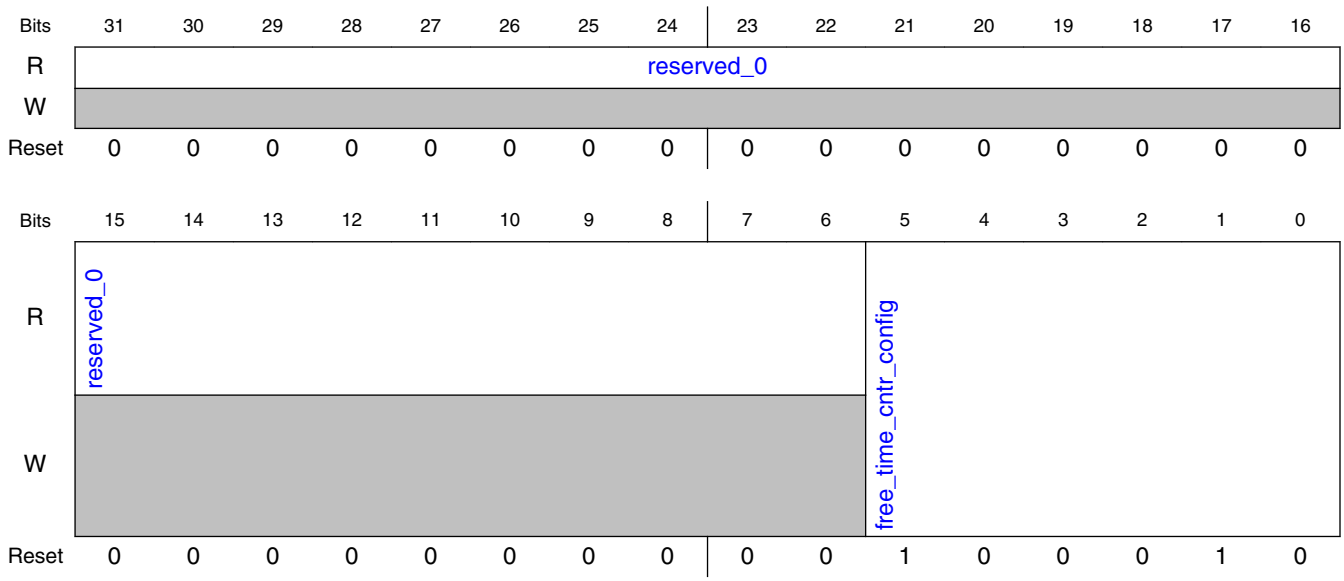
Field	Function
31-4 reserved_0	
3-0 TX_NUM_MSG	

13.4.10.1.244 CEC Free-Time counter (FREE_TIME_CNTR_CONFIG)

13.4.10.1.244.1 Offset

Register	Offset
FREE_TIME_CNTR_CONFIG	3954h

13.4.10.1.244.2 Diagram



13.4.10.1.244.3 Fields

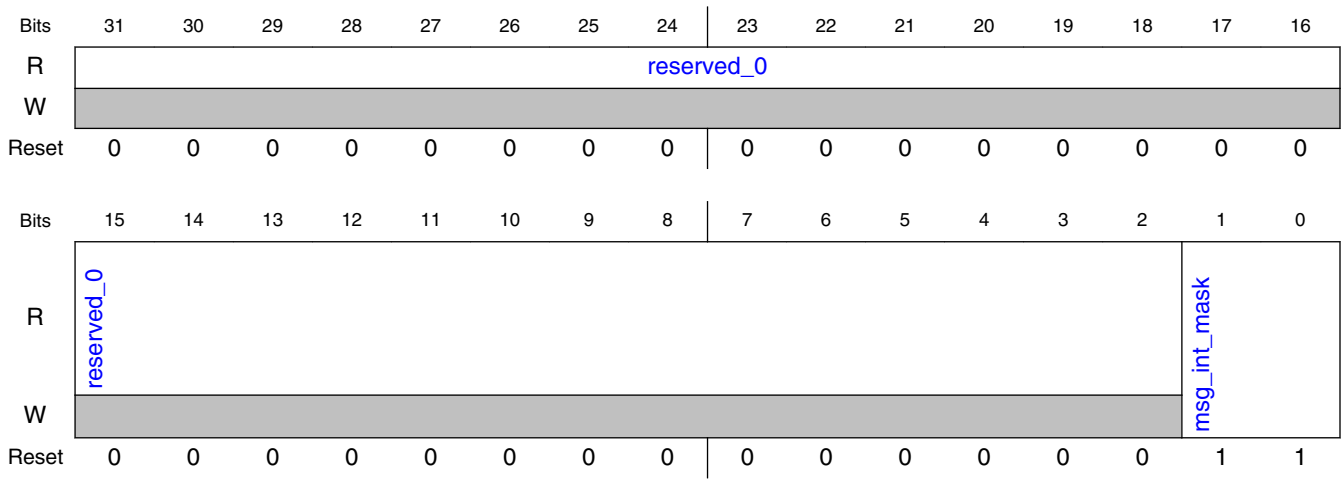
Field	Function
31-6 reserved_0	
5-0 free_time_cntr_config	Parallel load for 6 MSB in free_time_cntr counter. Parallel load for 6 MSB in free_time_cntr counter.

13.4.10.1.245 CEC Interrupt Mask (INT_CEC_MASK)

13.4.10.1.245.1 Offset

Register	Offset
INT_CEC_MASK	3958h

13.4.10.1.245.2 Diagram



13.4.10.1.245.3 Fields

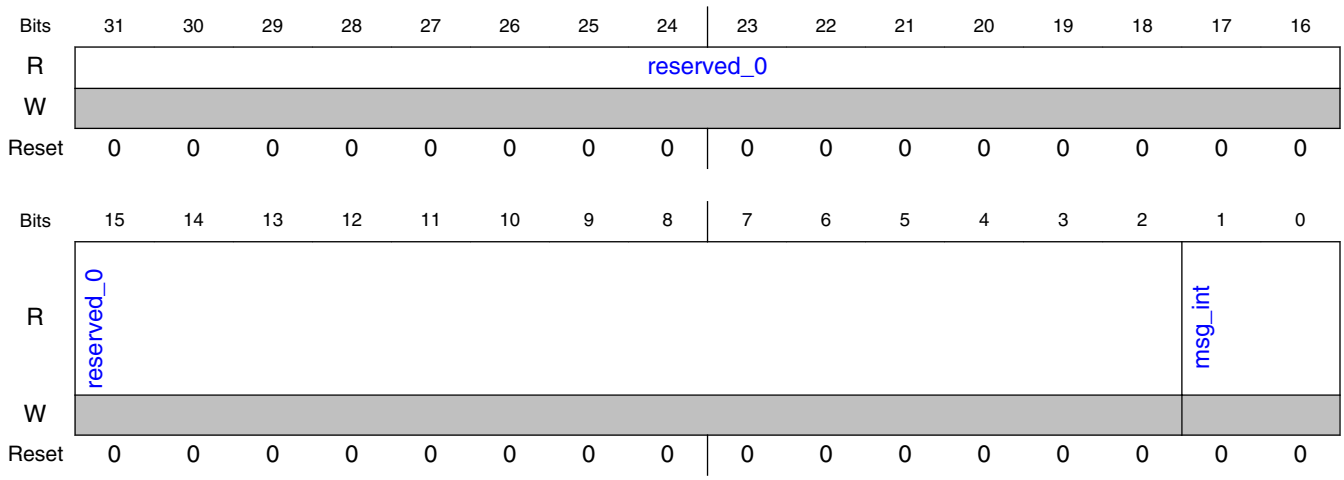
Field	Function
31-2 reserved_0	
1-0 msg_int_mask	Interrupt Mask. Interrupt Mask.

13.4.10.1.246 CEC Interrupt status (INT_STAT)

13.4.10.1.246.1 Offset

Register	Offset
INT_STAT	395Ch

13.4.10.1.246.2 Diagram



13.4.10.1.246.3 Fields

Field	Function
31-2 reserved_0	
1-0 msg_int	Interrupt Stat. Interrupt Stat.

13.4.10.1.247 CEC DB L timer (DB_L_TIMER)

13.4.10.1.247.1 Offset

Register	Offset
DB_L_TIMER	3980h

13.4.10.1.247.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								DB_L_TIMER							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.247.3 Fields

Field	Function
31-8 reserved_0	
7-0 DB_L_TIMER	

13.4.10.1.248 CEC DB M timer (DB_M_TIMER)

13.4.10.1.248.1 Offset

Register	Offset
DB_M_TIMER	3984h

13.4.10.1.248.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	reserved_0								DB_M_TIMER							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.248.3 Fields

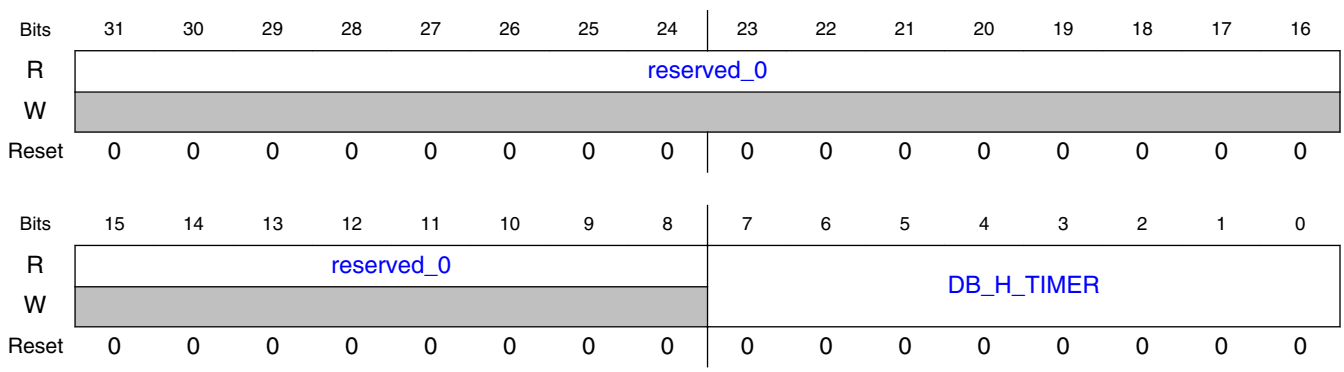
Field	Function
31-8 reserved_0	
7-0 DB_M_TIMER	

13.4.10.1.249 CEC DB H timer (DB_H_TIMER)

13.4.10.1.249.1 Offset

Register	Offset
DB_H_TIMER	3988h

13.4.10.1.249.2 Diagram



13.4.10.1.249.3 Fields

Field	Function
31-8 reserved_0	
7-0 DB_H_TIMER	

13.4.10.1.250 Contains the revision of the internal HDCP 1. (CRYPTO_HDCP_REVISION)

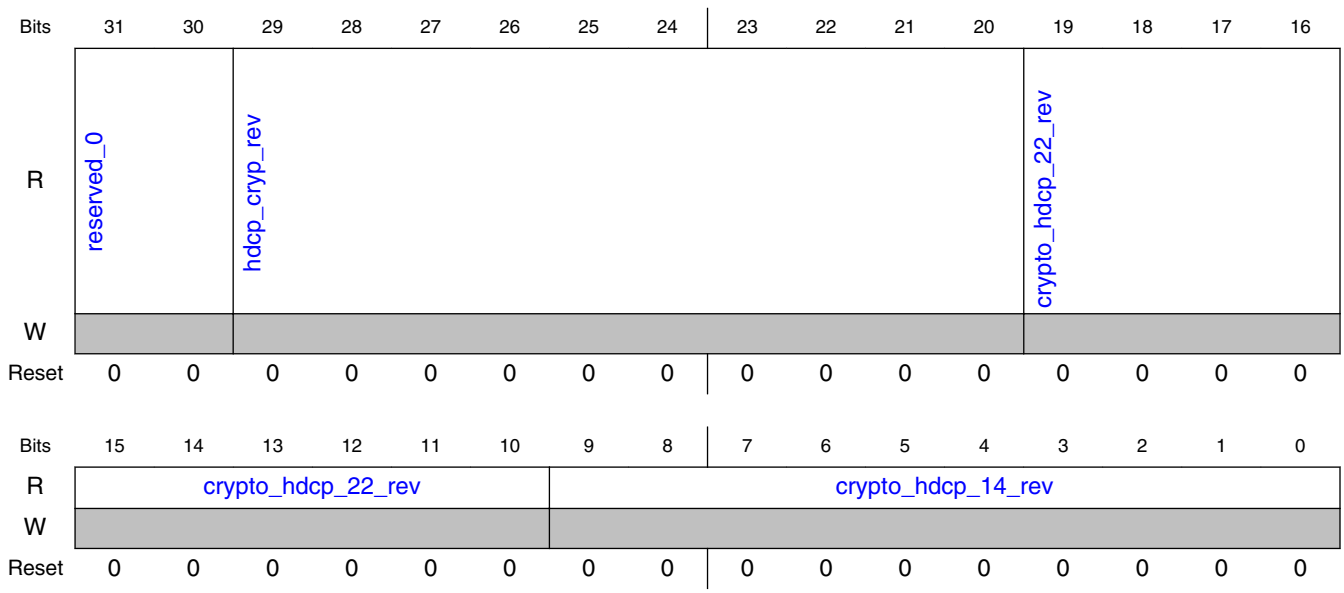
13.4.10.1.250.1 Offset

Register	Offset
CRYPTO_HDCP_REVISION	4000h

13.4.10.1.250.2 Function

Contains the revision of the internal HDCP 1.4 and 2.2 module

13.4.10.1.250.3 Diagram



13.4.10.1.250.4 Fields

Field	Function
31-30 reserved_0	
29-20 hdcp_cryp_rev	Revision of the HDCP Crypto block. Revision of the HDCP Crypto block.
19-10	Revision of the HDCP Crypto 2. Revision of the HDCP Crypto 2.2 block.

Table continues on the next page...

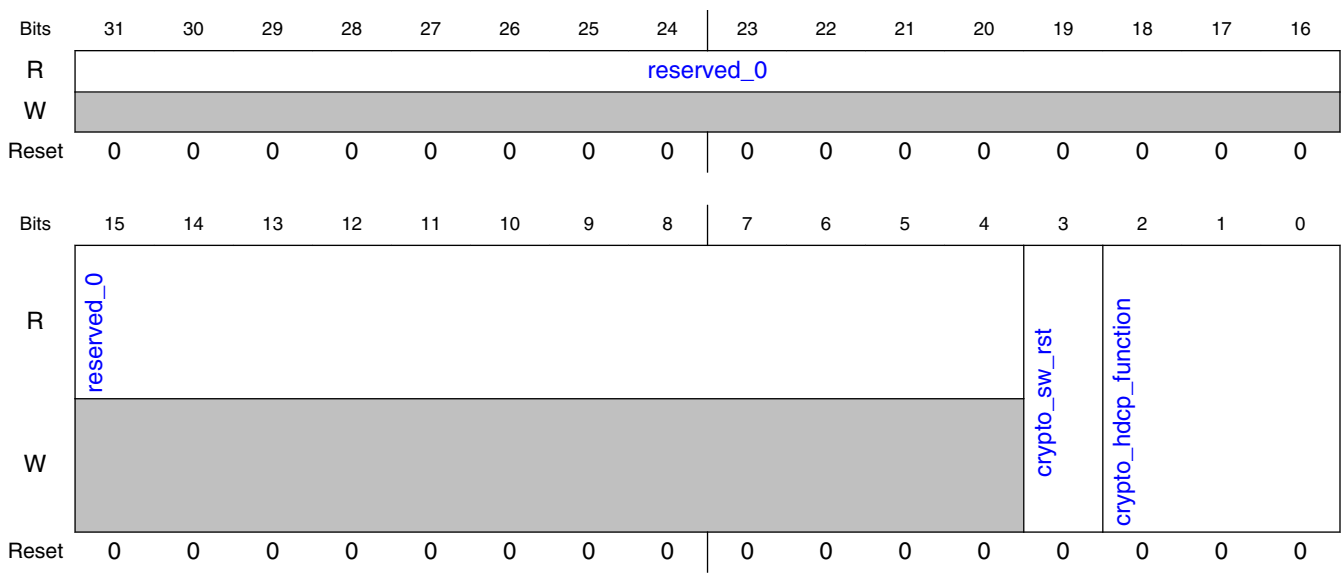
Field	Function
crypto_hdcp_22_rev	
9-0 crypto_hdcp_14_rev	Revision of the HDCP Crypto 1. Revision of the HDCP Crypto 1.4 block.

13.4.10.1.251 Contains global configuration information for the HDCP Crypto module (HDCP_CRYPTO_CONFIG)

13.4.10.1.251.1 Offset

Register	Offset
HDCP_CRYPTO_CONFIG	4004h

13.4.10.1.251.2 Diagram



13.4.10.1.251.3 Fields

Field	Function
31-4 reserved_0	

Table continues on the next page...

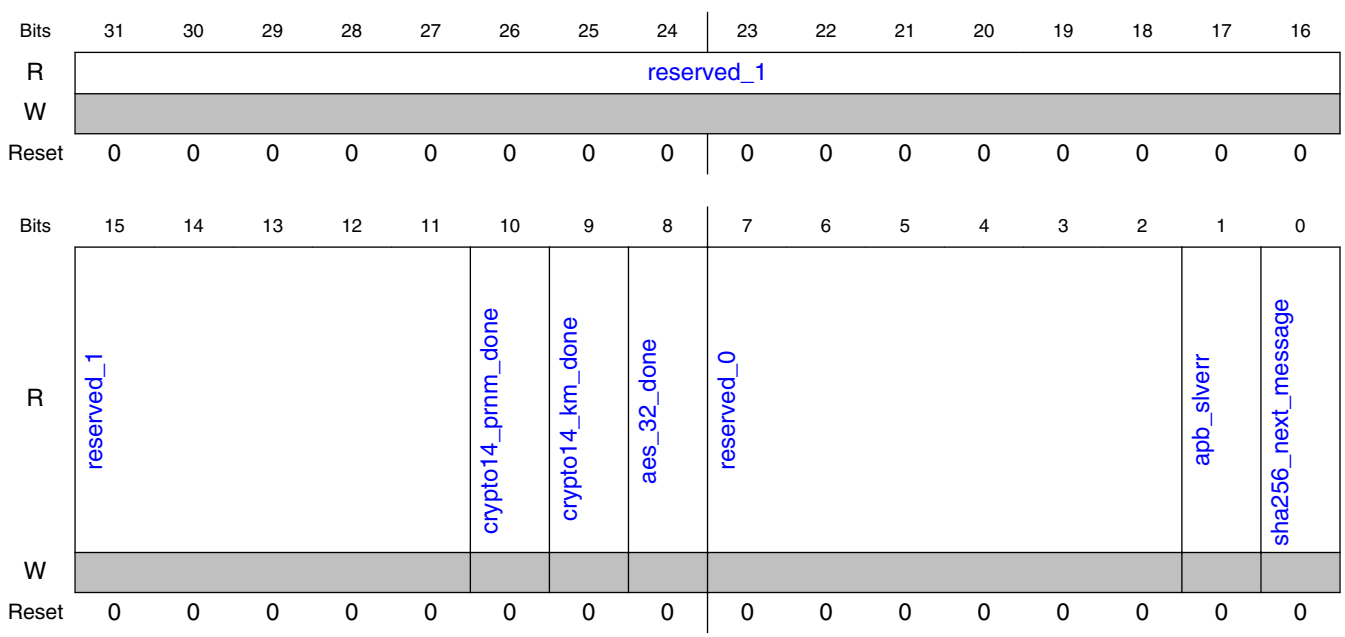
Field	Function
3 crypto_sw_rst	Software reset for the Crypto module. Software reset for the Crypto module.
2-0 crypto_hdcp_function	Enables a version of the Crypto function: 0: HDCP 1. Enables a version of the Crypto function: 0: HDCP 1.4 1: HDCP 2.2 Others: Reserved

13.4.10.1.252 Contains the status of the HDCP interrupt sources (CRYPTO_INTERRUPT_SOURCE)

13.4.10.1.252.1 Offset

Register	Offset
CRYPTO_INTERRUPT_SOURCE	4008h

13.4.10.1.252.2 Diagram



13.4.10.1.252.3 Fields

Field	Function
31-11	

Table continues on the next page...

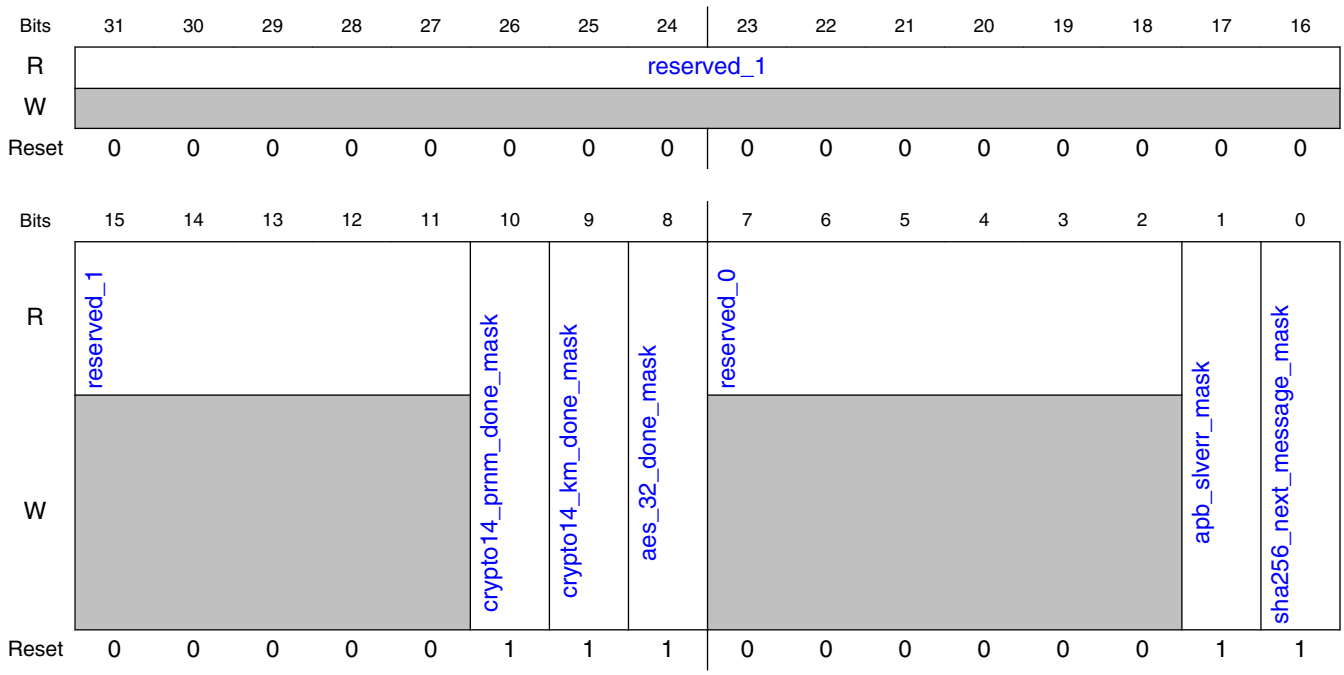
Field	Function
reserved_1	
10 crypto14_prnm_done	LFSR and block output finished calculation. LFSR and block output finished calculation.
9 crypto14_km_done	Done reading/calculating Km. Done reading/calculating Km.
8 aes_32_done	Asserted when the rising edge of the AES-32 done output is detected. Asserted when the rising edge of the AES-32 done output is detected.
7-2 reserved_0	
1 apb_slverr	APB slave error. APB slave error. Asserted when APB address is out of address range.
0 sha256_next_message	Asserted when the rising edge of the SHA256 done output is detected. Asserted when the rising edge of the SHA256 done output is detected.

13.4.10.1.253 CContains the mask vector of the HDCP interrupt sources (CRYPTO_INTERRUPT_MASK)

13.4.10.1.253.1 Offset

Register	Offset
CRYPTO_INTERRUPT_MASK	400Ch

13.4.10.1.253.2 Diagram



13.4.10.1.253.3 Fields

Field	Function
31-11 reserved_1	
10 crypto14_prnm_done_mask	Set to 1 to mask the crypto14_prnm_done interrupt. Set to 1 to mask the crypto14_prnm_done interrupt.
9 crypto14_km_done_mask	Set to 1 to mask the crypto14_km_done interrupt. Set to 1 to mask the crypto14_km_done interrupt.
8 aes_32_done_mask	Set to 1 to mask the AES32_done interrupt. Set to 1 to mask the AES32_done interrupt.
7-2 reserved_0	
1 apb_slverr_mask	Set to 1 for the apb_slverr interrupt. Set to 1 for the apb_slverr interrupt.
0 sha256_next_message_mask	Set to 1 to mask the SHA256_done interrupt. Set to 1 to mask the SHA256_done interrupt.

13.4.10.1.254 Contains global configuration information for the HDCP 2. (CRYPTO22_CONFIG)

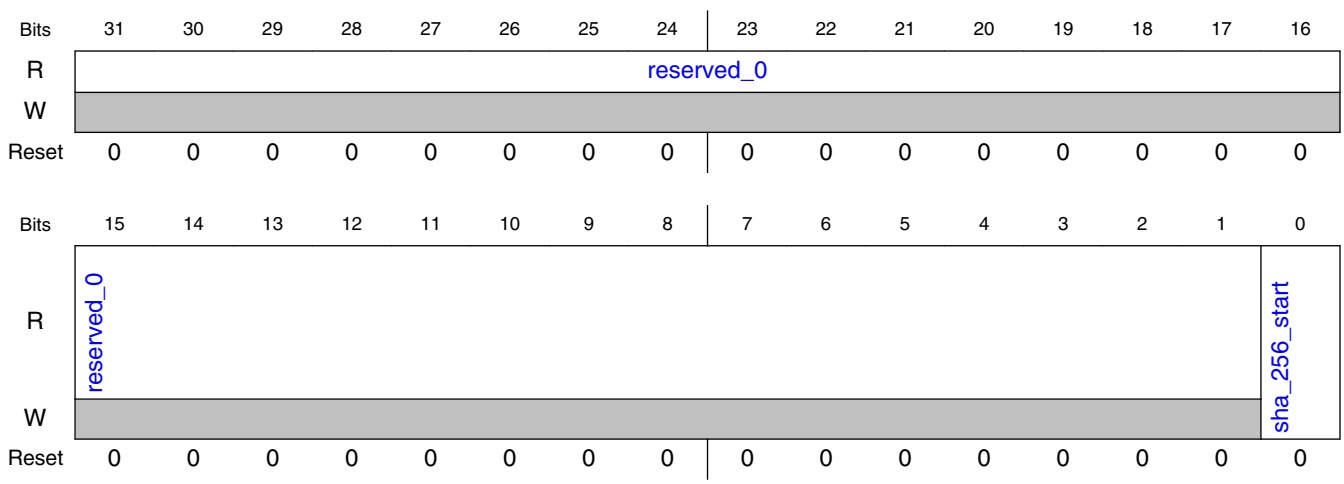
13.4.10.1.254.1 Offset

Register	Offset
CRYPTO22_CONFIG	4018h

13.4.10.1.254.2 Function

Contains global configuration information for the HDCP 2.2 Crypto module

13.4.10.1.254.3 Diagram



13.4.10.1.254.4 Fields

Field	Function
31-1 reserved_0	
0 sha_256_start	Set to 1 for Sha-256 start

13.4.10.1.255 Crypto 2. (CRYPTO22_STATUS)

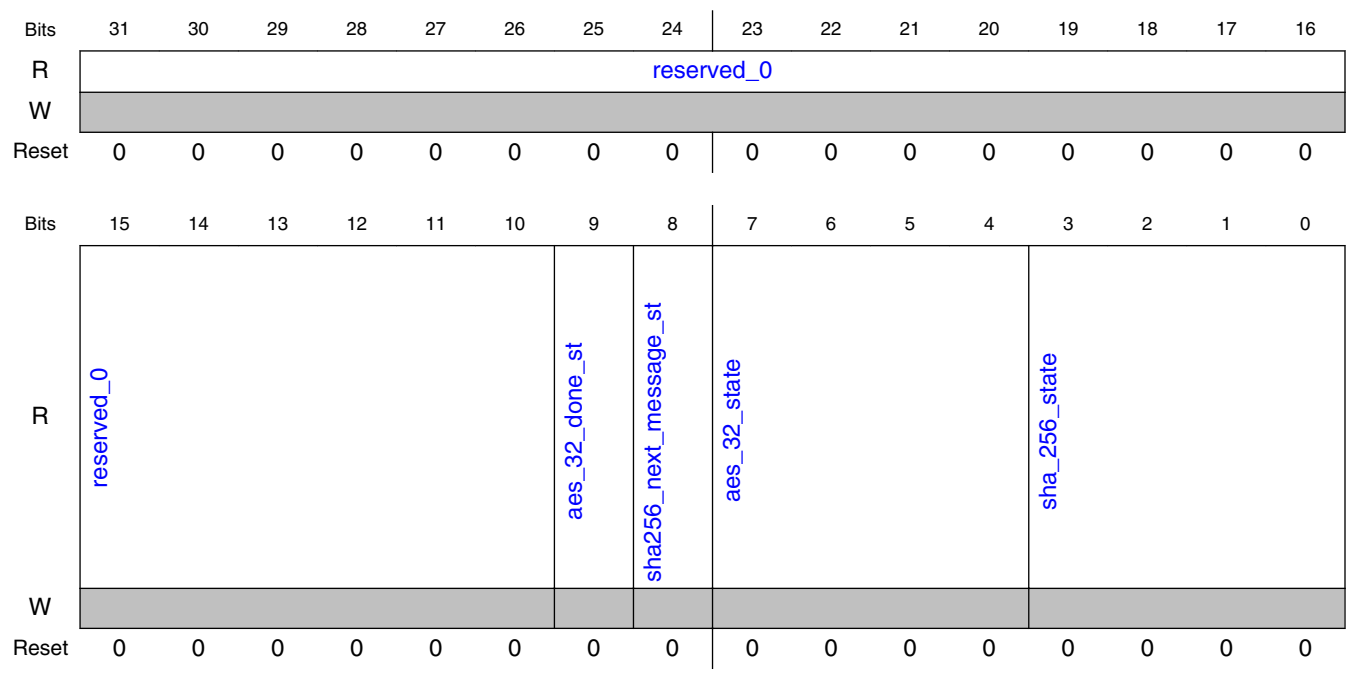
13.4.10.1.255.1 Offset

Register	Offset
CRYPTO22_STATUS	401Ch

13.4.10.1.255.2 Function

Crypto 2.2 global status register

13.4.10.1.255.3 Diagram



13.4.10.1.255.4 Fields

Field	Function
31-10 reserved_0	
9 aes_32_done_st	Asserted when the rising edge of the AES-32 done output is detected. Asserted when the rising edge of the AES-32 done output is detected.
8 sha256_next_message_st	Asserted when the SHA-256 module is ready to receive the next message
7-4	AES-32 current state

Table continues on the next page...

Field	Function
aes_32_state	
3-0 sha_256_state	SHA-256 current state

13.4.10.1.256 Holds 32-bit input data word of the SHA-256 module (SHA_256_DATA_IN)

13.4.10.1.256.1 Offset

Register	Offset
SHA_256_DATA_IN	403Ch

13.4.10.1.256.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sha_256_data_in															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	sha_256_data_in															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.256.3 Fields

Field	Function
31-0 sha_256_data_in	Holds the 32-bit input data word of the SHA-256 module. Holds the 32-bit input data word of the SHA-256 module.

13.4.10.1.257 Result of operation SHA-256 - 1' dw (SHA_256_DATA_OUT_0)

13.4.10.1.257.1 Offset

Register	Offset
SHA_256_DATA_OUT_0	4050h

13.4.10.1.257.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.257.3 Fields

Field	Function
31-0 sha_256_data_out_0	Holds the least significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the least significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.258 Result of operation SHA-256 - 2' dw (SHA_256_DATA_OUT_1)

13.4.10.1.258.1 Offset

Register	Offset
SHA_256_DATA_OUT_1	4054h

13.4.10.1.258.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.258.3 Fields

Field	Function
31-0 sha_256_data_out_1	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.259 Result of operation SHA-256 - 3' dw (SHA_256_DATA_OUT_2)

13.4.10.1.259.1 Offset

Register	Offset
SHA_256_DATA_OUT_2	4058h

13.4.10.1.259.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.259.3 Fields

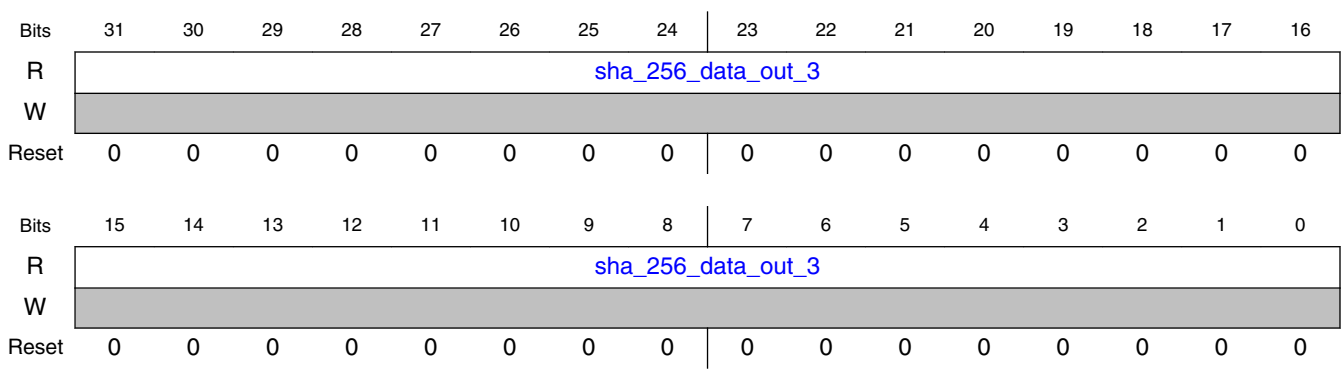
Field	Function
31-0 sha_256_data_out_2	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module

13.4.10.1.260 Result of operation SHA-256 - 4' dw (SHA_256_DATA_OUT_3)

13.4.10.1.260.1 Offset

Register	Offset
SHA_256_DATA_OUT_3	405Ch

13.4.10.1.260.2 Diagram



13.4.10.1.260.3 Fields

Field	Function
31-0 sha_256_data_out_3	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.261 Result of operation SHA-256 - 5' dw (SHA_256_DATA_OUT_4)

13.4.10.1.261.1 Offset

Register	Offset
SHA_256_DATA_OUT_4	4060h

13.4.10.1.261.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_4															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.261.3 Fields

Field	Function
31-0 sha_256_data_out_4	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.262 Result of operation SHA-256 - 6' dw (SHA_256_DATA_OUT_5)

13.4.10.1.262.1 Offset

Register	Offset
SHA_256_DATA_OUT_5	4064h

13.4.10.1.262.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_5															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.262.3 Fields

Field	Function
31-0 sha_256_data_out_5	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.263 Result of operation SHA-256 - 7' dw (SHA_256_DATA_OUT_6)

13.4.10.1.263.1 Offset

Register	Offset
SHA_256_DATA_OUT_6	4068h

13.4.10.1.263.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_6															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_6															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.263.3 Fields

Field	Function
31-0 sha_256_data_out_6	Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the next significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.264 Result of operation SHA-256 - 8' dw (SHA_256_DATA_OUT_7)

13.4.10.1.264.1 Offset

Register	Offset
SHA_256_DATA_OUT_7	406Ch

13.4.10.1.264.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	sha_256_data_out_7															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	sha_256_data_out_7															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.264.3 Fields

Field	Function
31-0 sha_256_data_out_7	Holds the most significant 32-bits word of the 256-bits output data word of the SHA-256 module. Holds the most significant 32-bits word of the 256-bits output data word of the SHA-256 module.

13.4.10.1.265 Input key word of the AES-32 module - 1' dw (AES_32_KEY_0)

13.4.10.1.265.1 Offset

Register	Offset
AES_32_KEY_0	4070h

13.4.10.1.265.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_32_key_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_32_key_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.265.3 Fields

Field	Function
31-0 aes_32_key_0	Holds the least significant 32-bits word of the 128-bits input key word of the AES-32 module. Holds the least significant 32-bits word of the 128-bits input key word of the AES-32 module.

13.4.10.1.266 Input key word of the AES-32 module - 2' dw (AES_32_KEY_1)

13.4.10.1.266.1 Offset

Register	Offset
AES_32_KEY_1	4074h

13.4.10.1.266.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_32_key_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_32_key_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.266.3 Fields

Field	Function
31-0 aes_32_key_1	Holds the next significant 32-bits word of the 128-bits input key word of the AES-32 module. Holds the next significant 32-bits word of the 128-bits input key word of the AES-32 module.

13.4.10.1.267 Input key word of the AES-32 module - 3' dw (AES_32_KEY_2)

13.4.10.1.267.1 Offset

Register	Offset
AES_32_KEY_2	4078h

13.4.10.1.267.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_32_key_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_32_key_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.267.3 Fields

Field	Function
31-0 aes_32_key_2	Holds the next significant 32-bits word of the 128-bits input key word of the AES-32 module. Holds the next significant 32-bits word of the 128-bits input key word of the AES-32 module.

13.4.10.1.268 Input key word of the AES-32 module - 4' dw (AES_32_KEY_3)

13.4.10.1.268.1 Offset

Register	Offset
AES_32_KEY_3	407Ch

13.4.10.1.268.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_32_key_3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_32_key_3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.268.3 Fields

Field	Function
31-0 aes_32_key_3	Holds the most significant 32-bits word of the 128-bits input key word of the AES-32 module. Holds the most significant 32-bits word of the 128-bits input key word of the AES-32 module.

13.4.10.1.269 Input data word to the AES-32 module (AES_32_DATA_IN)

13.4.10.1.269.1 Offset

Register	Offset
AES_32_DATA_IN	4080h

13.4.10.1.269.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_32_data_in															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_32_data_in															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.269.3 Fields

Field	Function
31-0 aes_32_data_in	Holds the input data word to the AES-32 module. Holds the input data word to the AES-32 module.

13.4.10.1.270 AES-32 module - 128-bits output data word - 1' dw (AES_32_DATA_OUT_0)

13.4.10.1.270.1 Offset

Register	Offset
AES_32_DATA_OUT_0	4084h

13.4.10.1.270.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	aes_32_data_out_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	aes_32_data_out_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.270.3 Fields

Field	Function
31-0 aes_32_data_out_0	Holds the least significant 32-bits word of the 128-bits output data word of the AES-32 module. Holds the least significant 32-bits word of the 128-bits output data word of the AES-32 module.

13.4.10.1.271 AES-32 module - 128-bits output data word - 2' dw (AES_32_DATA_OUT_1)

13.4.10.1.271.1 Offset

Register	Offset
AES_32_DATA_OUT_1	4088h

13.4.10.1.271.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	aes_32_data_out_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	aes_32_data_out_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.271.3 Fields

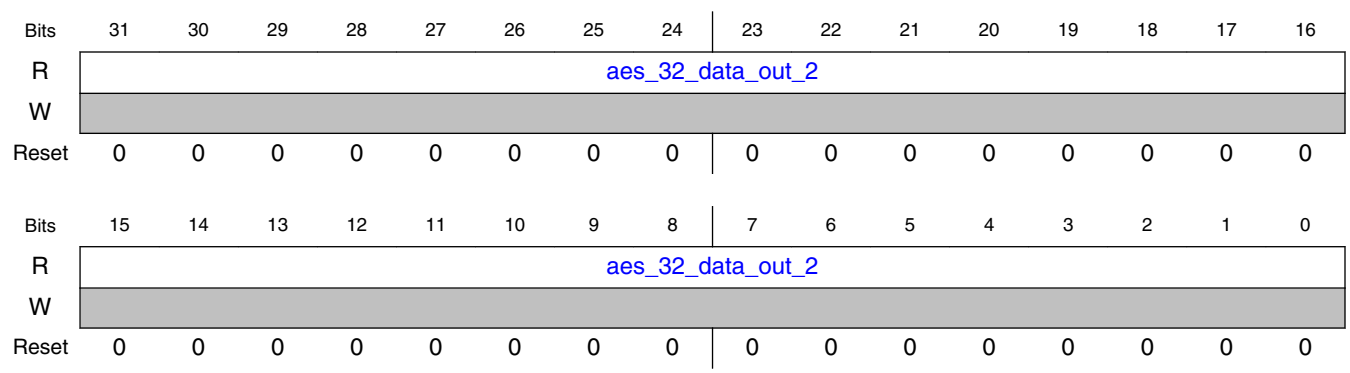
Field	Function
31-0 aes_32_data_out_1	Holds the next significant 32-bits word of the 128-bits output data word of the AES-32 module. Holds the next significant 32-bits word of the 128-bits output data word of the AES-32 module.

13.4.10.1.272 AES-32 module - 128-bits output data word - 3' dw (AES_32_DATA_OUT_2)

13.4.10.1.272.1 Offset

Register	Offset
AES_32_DATA_OUT_2	408Ch

13.4.10.1.272.2 Diagram



13.4.10.1.272.3 Fields

Field	Function
31-0 aes_32_data_out_2	Holds the next significant 32-bits word of the 128-bits output data word of the AES-32 module. Holds the next significant 32-bits word of the 128-bits output data word of the AES-32 module.

13.4.10.1.273 AES-32 module - 128-bits output data word - 4' dw (AES_32_DATA_OUT_3)

13.4.10.1.273.1 Offset

Register	Offset
AES_32_DATA_OUT_3	4090h

13.4.10.1.273.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	aes_32_data_out_3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	aes_32_data_out_3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.273.3 Fields

Field	Function
31-0 aes_32_data_out_3	Holds the most significant 32-bits word of the 128-bits output data word of the AES-32 module. Holds the most significant 32-bits word of the 128-bits output data word of the AES-32 module.

13.4.10.1.274 Contains global configuration information for the HDCP 1. (CRYPTO14_CONFIG)

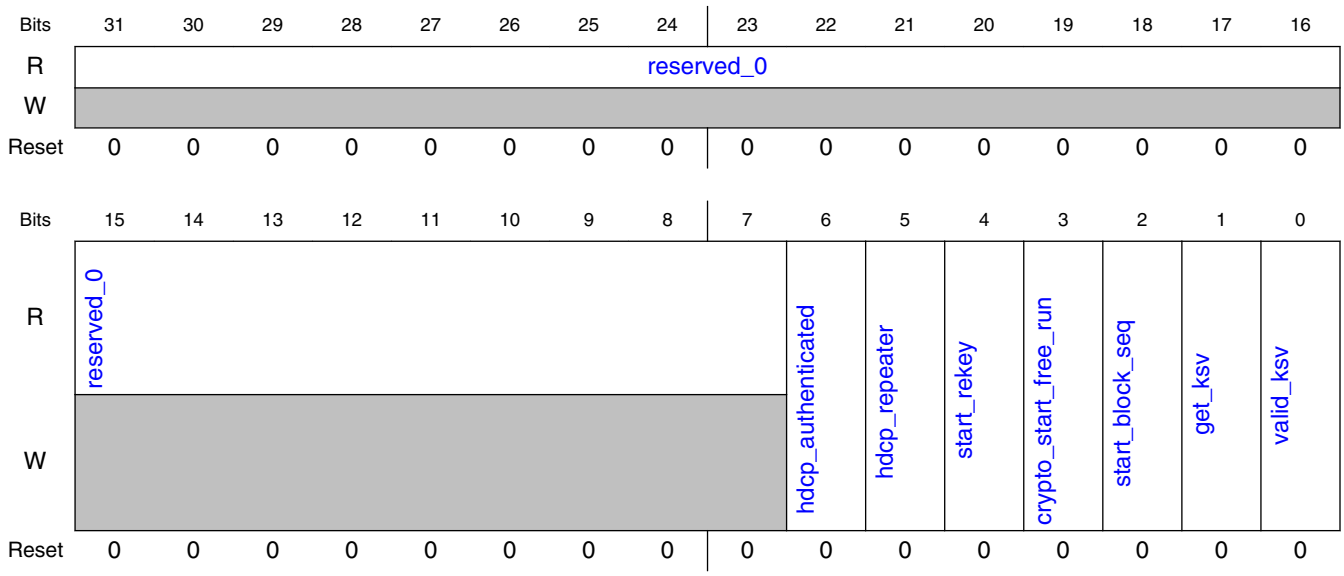
13.4.10.1.274.1 Offset

Register	Offset
CRYPTO14_CONFIG	40A0h

13.4.10.1.274.2 Function

Contains global configuration information for the HDCP 1.4 Crypto module

13.4.10.1.274.3 Diagram



13.4.10.1.274.4 Fields

Field	Function
31-7 reserved_0	
6 hdcp_authenticated	Authenticated finished. Authenticated finished.
5 hdcp_repeater	Repeater bit : 0: for the receiver, 1: for the repeater
4 start_rekey	Crypto 1. Crypto 1.4 command to start hdcpRekeyCipher
3 crypto_start_free_run	Crypto 1. Crypto 1.4 command to start free running enable for operation hdcpRngCipher
2 start_block_seq	Crypto 1. Crypto 1.4 command to start LFSR calculation
1 get_ksv	Read it's own KSV enable bit. Read it's own KSV enable bit.'0' reading not allowed'1' start reading.
0 valid_ksv	Enable for Km calculation. Enable for Km calculation. When high,start calculating Km. Indicates a good moment for ri_out sampling.

13.4.10.1.275 Contains global status information for the HDCP 1. (CRYPTO14_STATUS)

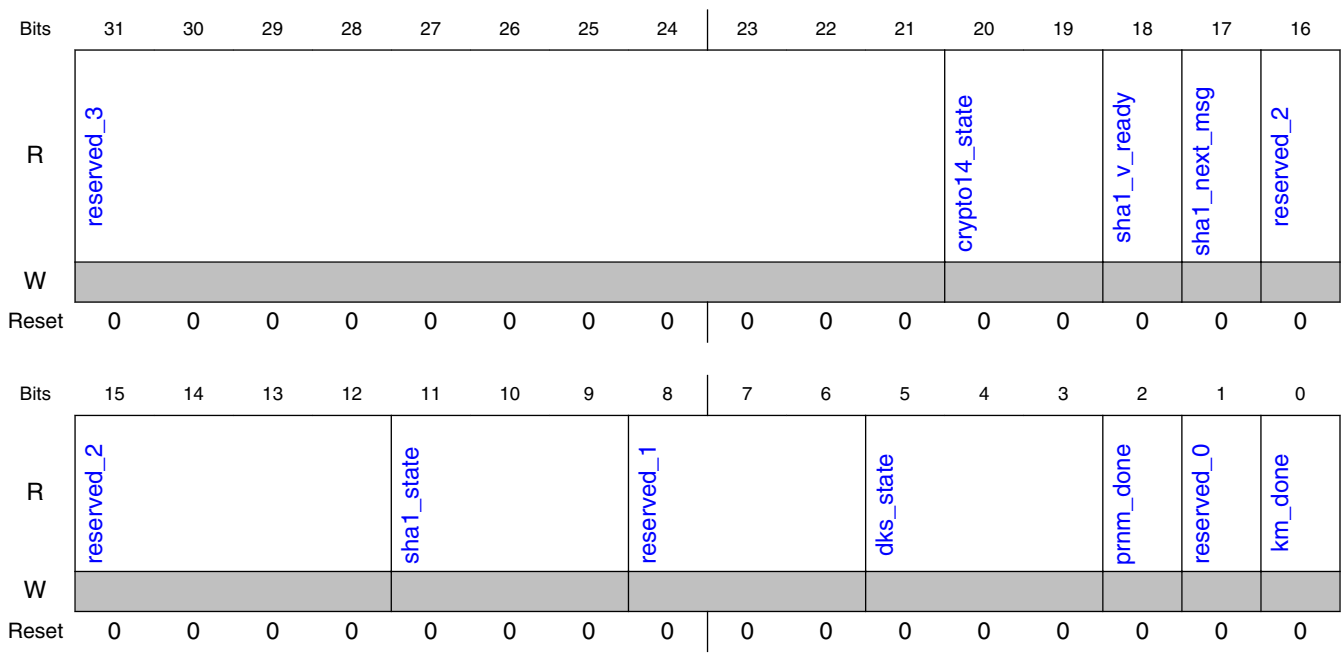
13.4.10.1.275.1 Offset

Register	Offset
CRYPTO14_STATUS	40A4h

13.4.10.1.275.2 Function

Contains global status information for the HDCP 1.4 Crypto module

13.4.10.1.275.3 Diagram



13.4.10.1.275.4 Fields

Field	Function
31-21 reserved_3	
20-19	Crypto operation SM state: Possible values:00- HDCP_RNG_CIPHER; 01 - HDCP_BLOCK_CIPHER; 10 - HDCP_STREAM_CIPHER; 11 - HDCP_REKEY_CIPHER

Table continues on the next page...

Field	Function
crypto14_state	
18 sha1_v_ready	Indication that V value from SHA-1 CRYPTO14_SHA1_V_VALUE_4 is ready. Indication that V value from SHA-1 CRYPTO14_SHA1_V_VALUE_4 is ready.
17 sha1_next_msg	Request for the next message block. Request for the next message block. When set high, CRYPTO14_SHA1_MSG_DATA_0-15 registers shall be written.
16-12 reserved_2	
11-9 sha1_state	Current state for Crypto 1. Current state for Crypto 1.4 SHA-1 FSM. Used for debug purpose.Possible values:000 - IDLE; 001 - PREPARE; 010 - CALCULATE; 011 - RESULT; 100 - BLOCK_WAIT
8-6 reserved_1	
5-3 dks_state	Crypto 1. Crypto 1.4 DKS current state. Used for debug purpose.Possible values:000 - HDCP_IDLE_KSV; 010 - HDCP_IDLE;100 - HDCP_PRECALC; 101 - HDCP_POSTCALC; 110 - HDCP_CALC; 111 - HDCP_READY
2 prnm_done	LFSR and block output finished calculation. LFSR and block output finished calculation.
1 reserved_0	
0 km_done	Done reading/calculating Km. Done reading/calculating Km. Used as interrupt event.

13.4.10.1.276 Contains 24-bit pseudo random data (CRYPTO14_PRNM_OUT)

13.4.10.1.276.1 Offset

Register	Offset
CRYPTO14_PRNM_OUT	40A8h

13.4.10.1.276.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								prnm_out							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	prnm_out															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.276.3 Fields

Field	Function
31-24 reserved_0	
23-0 prnm_out	24-bit pseudo-random data. 24-bit pseudo-random data.

13.4.10.1.277 Contains the first word of the Km value (CRYPTO14_KM_0)

13.4.10.1.277.1 Offset

Register	Offset
CRYPTO14_KM_0	40ACh

13.4.10.1.277.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	crypto14_km_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_km_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.277.3 Fields

Field	Function
31-0 crypto14_km_0	Holds the first word of the Km value. Holds the first word of the Km value.

13.4.10.1.278 Contains the most significant 3 bytes of the Km value (CRYPTO14_KM_1)

13.4.10.1.278.1 Offset

Register	Offset
CRYPTO14_KM_1	40B0h

13.4.10.1.278.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								crypto14_km_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_km_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.278.3 Fields

Field	Function
31-24 reserved_0	
23-0 crypto14_km_1	Holds the most significant 3 bytes of the Km value. Holds the most significant 3 bytes of the Km value.

13.4.10.1.279 First word of An value generated by hdcpRngCipher operation. (CRYPTO14_AN_0)

13.4.10.1.279.1 Offset

Register	Offset
CRYPTO14_AN_0	40B4h

13.4.10.1.279.2 Function

First word of An value generated by hdcpRngCipher operation.

13.4.10.1.279.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	crypto14_an_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	crypto14_an_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.279.4 Fields

Field	Function
31-0 crypto14_an_0	Holds the first 4 bytes of the An value. Holds the first 4 bytes of the An value.

13.4.10.1.280 Second word of An value generated by hdcpRngCipher operation (CRYPTO14_AN_1)

13.4.10.1.280.1 Offset

Register	Offset
CRYPTO14_AN_1	40B8h

13.4.10.1.280.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	crypto14_an_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	crypto14_an_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.280.3 Fields

Field	Function
31-0 crypto14_an_1	Holds the most significant 4 bytes of the An value. Holds the most significant 4 bytes of the An value.

13.4.10.1.281 First 32 bits of the KSV from the other HDCP device (CRYPTO14_YOUR_KSV_0)

13.4.10.1.281.1 Offset

Register	Offset
CRYPTO14_YOUR_KSV_0	40BCh

13.4.10.1.281.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	crypto14_your_ksv_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	crypto14_your_ksv_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.281.3 Fields

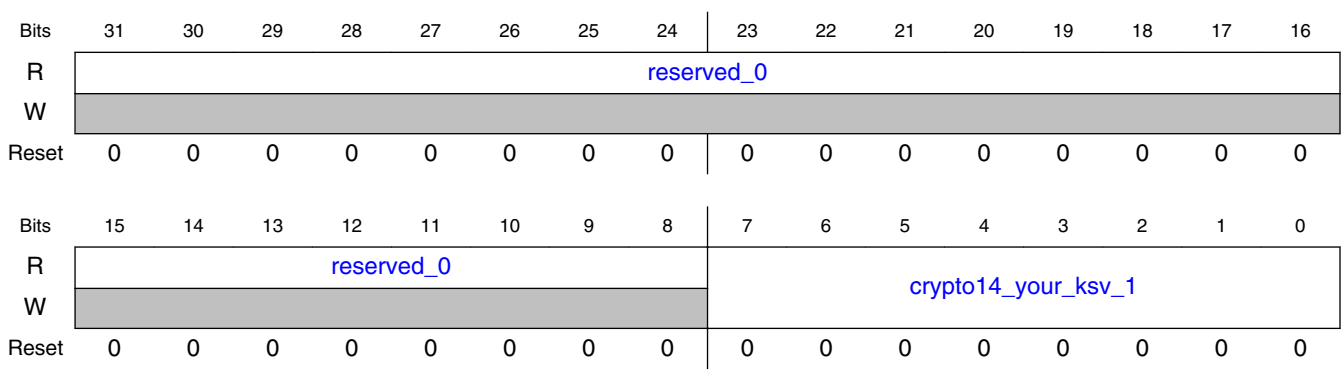
Field	Function
31-0 crypto14_your_ksv_0	Holds the first 32 bits of the KSV from the other HDCP device. Holds the first 32 bits of the KSV from the other HDCP device.

13.4.10.1.282 Last byte of the KSV from other HDCP device (CRYPTO14_YOUR_KSV_1)

13.4.10.1.282.1 Offset

Register	Offset
CRYPTO14_YOUR_KSV_1	40C0h

13.4.10.1.282.2 Diagram



13.4.10.1.282.3 Fields

Field	Function
31-8 reserved_0	
7-0 crypto14_your_ksv_1	Holds the last byte of the KSV from other HDCP device

13.4.10.1.283 Mi value - 1' dw (CRYPTO14_MI_0)

13.4.10.1.283.1 Offset

Register	Offset
CRYPTO14_MI_0	40C4h

13.4.10.1.283.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	crypto14_mi_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_mi_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.283.3 Fields

Field	Function
31-0 crypto14_mi_0	Mi value first 32 bits

13.4.10.1.284 Mi value - 2' dw (CRYPTO14_MI_1)

13.4.10.1.284.1 Offset

Register	Offset
CRYPTO14_MI_1	40C8h

13.4.10.1.284.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	crypto14_mi_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_mi_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.284.3 Fields

Field	Function
31-0 crypto14_mi_1	Mi value second 32 bits

13.4.10.1.285 Ti value (CRYPTO14_TI_0)

13.4.10.1.285.1 Offset

Register	Offset
CRYPTO14_TI_0	40CCh

13.4.10.1.285.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_ti_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.285.3 Fields

Field	Function
31-16 reserved_0	
15-0 crypto14_ti_0	Ti value

13.4.10.1.286 First 32 bits of the Ki frame key from this HDCP device (CRYPTO14_KI_0)

13.4.10.1.286.1 Offset

Register	Offset
CRYPTO14_KI_0	40D0h

13.4.10.1.286.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	crypto14_ki_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_ki_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.286.3 Fields

Field	Function
31-0 crypto14_ki_0	Holds the first 32 bits of the Ki frame key from this HDCP device. Holds the first 32 bits of the Ki frame key from this HDCP device.

13.4.10.1.287 Last 3 bytes of the Ki frame key from this HDCP device. (CRYPTO14_KI_1)

13.4.10.1.287.1 Offset

Register	Offset
CRYPTO14_KI_1	40D4h

13.4.10.1.287.2 Function

Last 3 bytes of the Ki frame key from this HDCP device.

13.4.10.1.287.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								crypto14_ki_1							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	crypto14_ki_1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.287.4 Fields

Field	Function
31-24 reserved_0	
23-0 crypto14_ki_1	Holds the last 3 bytes of the Ki frame key from this HDCP device. Holds the last 3 bytes of the Ki frame key from this HDCP device.

13.4.10.1.288 This register defines number of iterations for SHA-1 calculations (CRYPTO14_BLOCKS_NUM)

13.4.10.1.288.1 Offset

Register	Offset
CRYPTO14_BLOCKS_NUM	40D8h

13.4.10.1.288.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	blocks_num															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	blocks_num															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.288.3 Fields

Field	Function
31-0 blocks_num	Number of iterations for SHA-1 calculation. Number of iterations for SHA-1 calculation.

13.4.10.1.289 Key memory control register. (CRYPTO14_KEY_MEM_DATA_0)

13.4.10.1.289.1 Offset

Register	Offset
CRYPTO14_KEY_MEM_DATA_0	40DCh

13.4.10.1.289.2 Function

Key memory control register. Writing data to this register transfers data to the key RAM. This is input for DKS block. First 32 bits

13.4.10.1.289.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	key_mem_data_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	key_mem_data_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.289.4 Fields

Field	Function
31-0 key_mem_data_0	Output data from keys RAM. Output data from keys RAM. Input for DKS block. First 32 bits.

13.4.10.1.290 Key memory control register. (CRYPTO14_KEY_MEM_DATA_1)

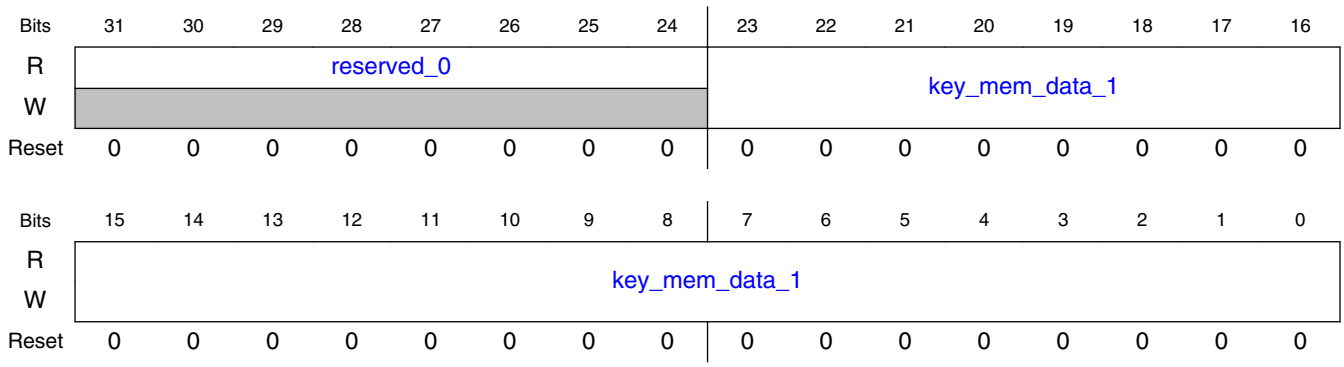
13.4.10.1.290.1 Offset

Register	Offset
CRYPTO14_KEY_MEM_DATA_1	40E0h

13.4.10.1.290.2 Function

Key memory control register. Writing data to this register transfers data to the key RAM. When data is written to this register, pulse key_mem_vld shall be also generated to the core_clk clock domain, as indication of new key_mem_data. This is input for DKS block. Last 3 bytes

13.4.10.1.290.3 Diagram



13.4.10.1.290.4 Fields

Field	Function
31-24 reserved_0	
23-0 key_mem_data_1	Output data from keys RAM. Output data from keys RAM. Input for DKS block. Last 3 bytes.

13.4.10.1.291 SHA1 message control register. (CRYPTO14_SHA1_MSG_DATA)

13.4.10.1.291.1 Offset

Register	Offset
CRYPTO14_SHA1_MSG_DATA	40E4h

13.4.10.1.291.2 Function

SHA1 message control register. 32-bit word for SHA-1 message. Input for SHA-1 block. Writing data to this register transfers data to the SHA-1 block. Write to this register shall be repeated 16 times.

13.4.10.1.291.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	sha1_msg_data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	sha1_msg_data															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.291.4 Fields

Field	Function
31-0 sha1_msg_data	32-bit word for SHA-1 message. 32-bit word for SHA-1 message. Input for SHA-1 block.

13.4.10.1.292 SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_0)

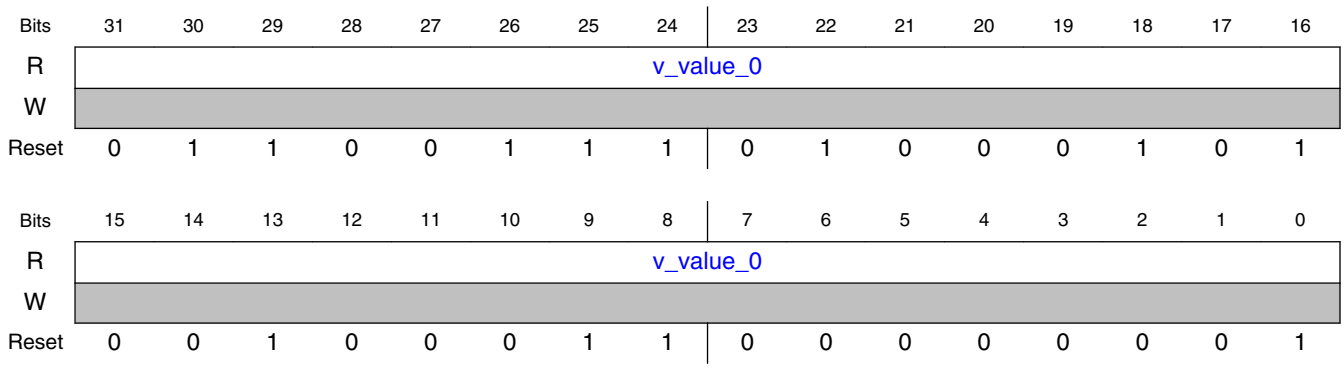
13.4.10.1.292.1 Offset

Register	Offset
CRYPTO14_SHA1_V_VALUE_0	40E8h

13.4.10.1.292.2 Function

SHA1 message status register. First 32-bit word for SHA-1 calculation. Output from SHA-1 block

13.4.10.1.292.3 Diagram



13.4.10.1.292.4 Fields

Field	Function
31-0 v_value_0	First 32-bit word for SHA-1 calculation value. First 32-bit word for SHA-1 calculation value. Output from SHA-1 block.

13.4.10.1.293 SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_1)

13.4.10.1.293.1 Offset

Register	Offset
CRYPTO14_SHA1_V_VALUE_1	40ECh

13.4.10.1.293.2 Function

SHA1 message status register. Second 32-bit word for SHA-1 calculation. Output from SHA-1 block

13.4.10.1.293.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	v_value_1															
W																
Reset	1	1	1	0	1	1	1	1	1	1	0	0	1	1	0	1

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	v_value_1															
W																
Reset	1	0	1	0	1	0	1	1	1	0	0	0	1	0	0	1

13.4.10.1.293.4 Fields

Field	Function
31-0 v_value_1	Second 32-bit word for SHA-1 calculation value. Second 32-bit word for SHA-1 calculation value. Output from SHA-1 block.

13.4.10.1.294 SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_2)

13.4.10.1.294.1 Offset

Register	Offset
CRYPTO14_SHA1_V_VALUE_2	40F0h

13.4.10.1.294.2 Function

SHA1 message status register. Third 32-bit word for SHA-1 calculation. Output from SHA-1 block

13.4.10.1.294.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	v_value_2															
W																
Reset	1	0	0	1	1	0	0	0	1	0	1	1	1	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	v_value_2															
W																
Reset	1	1	0	1	1	1	0	0	1	1	1	1	1	1	1	0

13.4.10.1.294.4 Fields

Field	Function
31-0 v_value_2	Third 32-bit word for SHA-1 calculation value. Third 32-bit word for SHA-1 calculation value. Output from SHA-1 block.

13.4.10.1.295 SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_3)

13.4.10.1.295.1 Offset

Register	Offset
CRYPTO14_SHA1_V_VALUE_3	40F4h

13.4.10.1.295.2 Function

SHA1 message status register. Third 32-bit word for SHA-1 calculation. Output from SHA-1 block

13.4.10.1.295.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	v_value_3															
W																
Reset	0	0	0	1	0	0	0	0	0	0	1	1	0	0	1	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	v_value_3															
W																
Reset	0	1	0	1	0	1	0	0	0	1	1	1	0	1	1	0

13.4.10.1.295.4 Fields

Field	Function
31-0 v_value_3	4th 32-bit word for SHA-1 calculation value. 4th 32-bit word for SHA-1 calculation value. Output from SHA-1 block.

13.4.10.1.296 SHA1 message status register. (CRYPTO14_SHA1_V_VALUE_4)

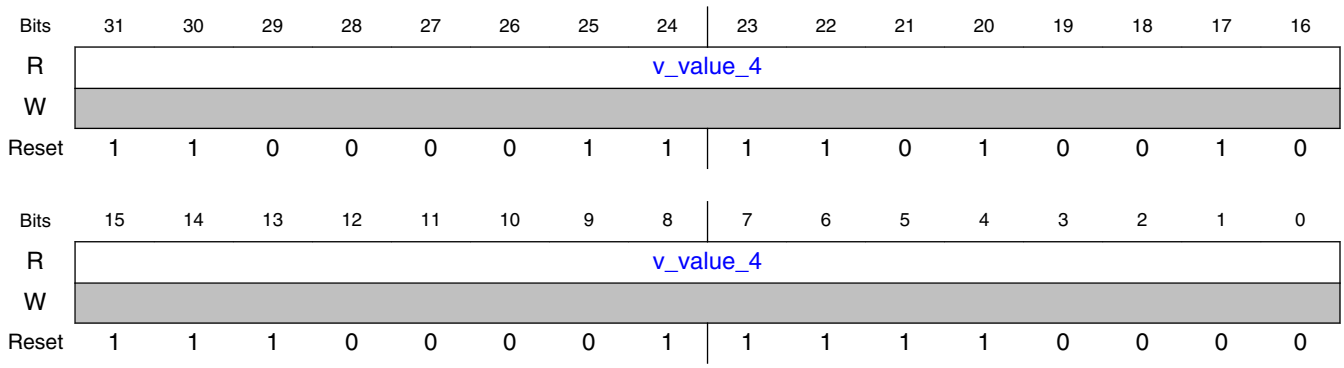
13.4.10.1.296.1 Offset

Register	Offset
CRYPTO14_SHA1_V_VALUE_4	40F8h

13.4.10.1.296.2 Function

SHA1 message status register. 5th 32-bit word for SHA-1 calculation. Output from SHA-1 block

13.4.10.1.296.3 Diagram



13.4.10.1.296.4 Fields

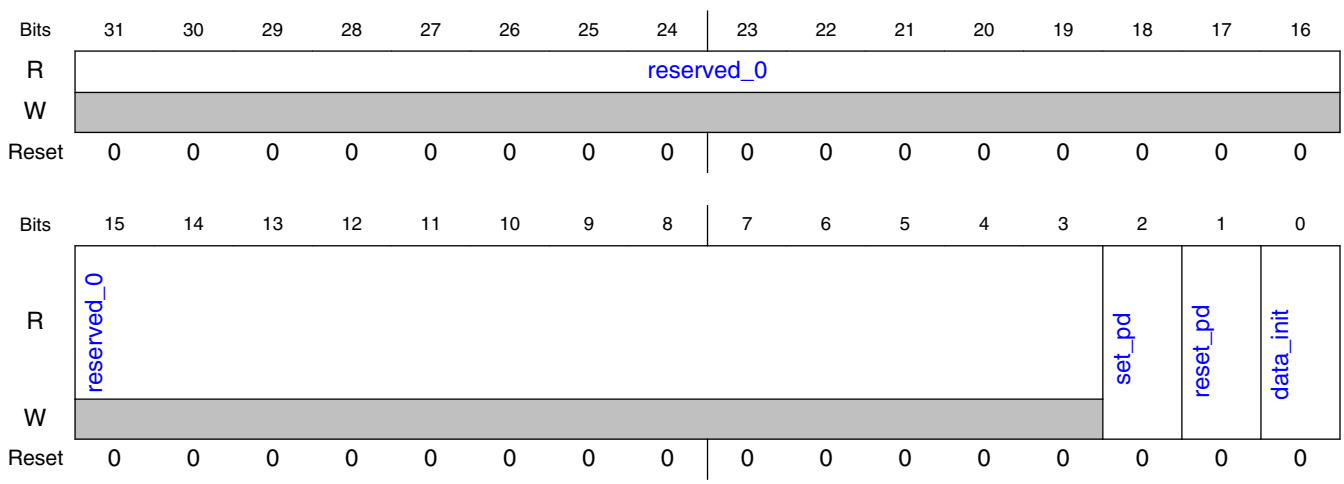
Field	Function
31-0 v_value_4	5th 32-bit word for SHA-1 calculation value. 5th 32-bit word for SHA-1 calculation value. Output from SHA-1 block.

13.4.10.1.297 TRNG control (TRNG_CTRL)

13.4.10.1.297.1 Offset

Register	Offset
TRNG_CTRL	40FCh

13.4.10.1.297.2 Diagram



13.4.10.1.297.3 Fields

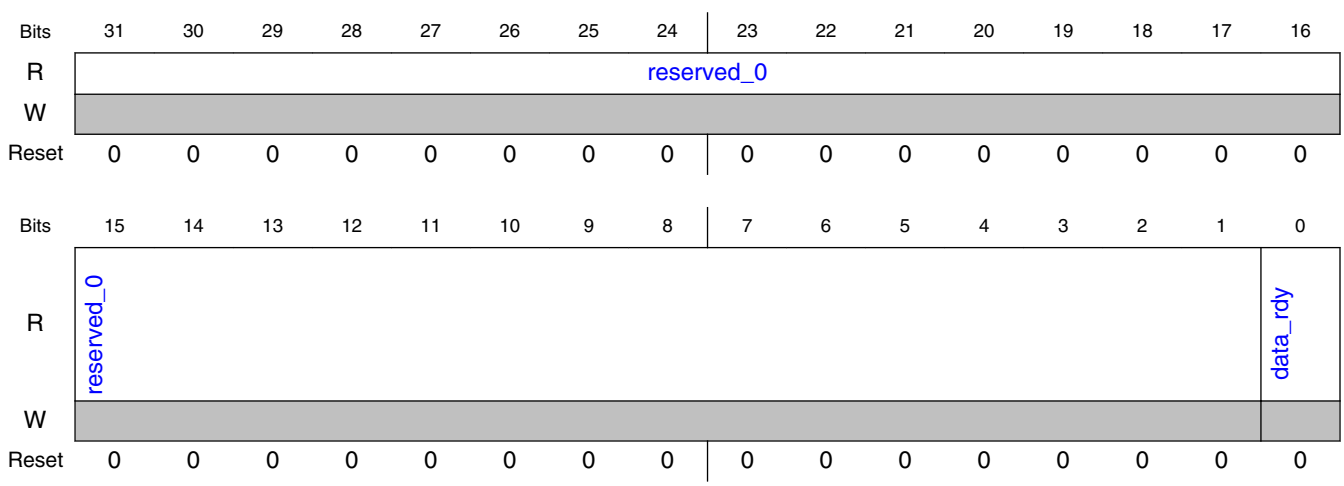
Field	Function
31-3 reserved_0	
2 set_pd	set pd
1 reset_pd	resetpd output
0 data_init	init trng read

13.4.10.1.298 TRNG status (TRNG_DATA_RDY)

13.4.10.1.298.1 Offset

Register	Offset
TRNG_DATA_RDY	4100h

13.4.10.1.298.2 Diagram



13.4.10.1.298.3 Fields

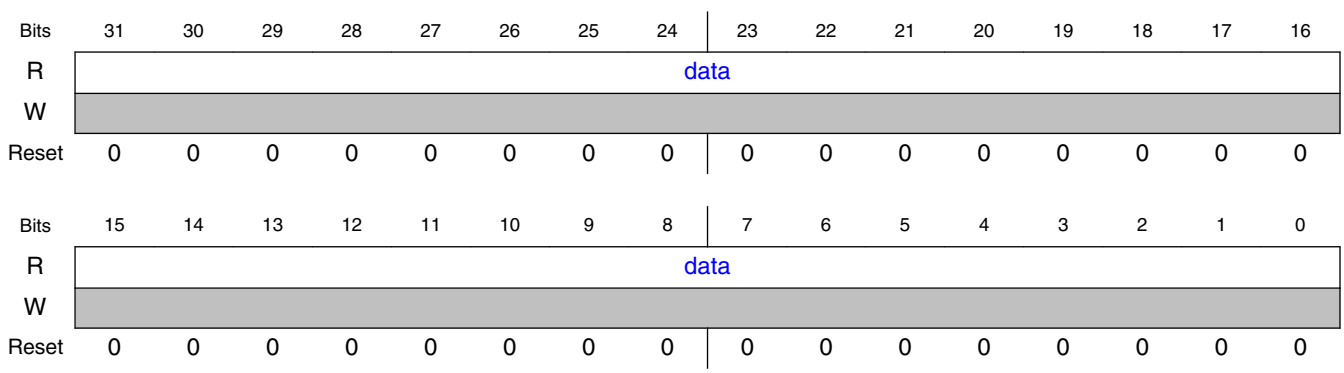
Field	Function
31-1 reserved_0	
0 data_rdy	

13.4.10.1.299 TRNG data register (TRNG_DATA)

13.4.10.1.299.1 Offset

Register	Offset
TRNG_DATA	4104h

13.4.10.1.299.2 Diagram



13.4.10.1.299.3 Fields

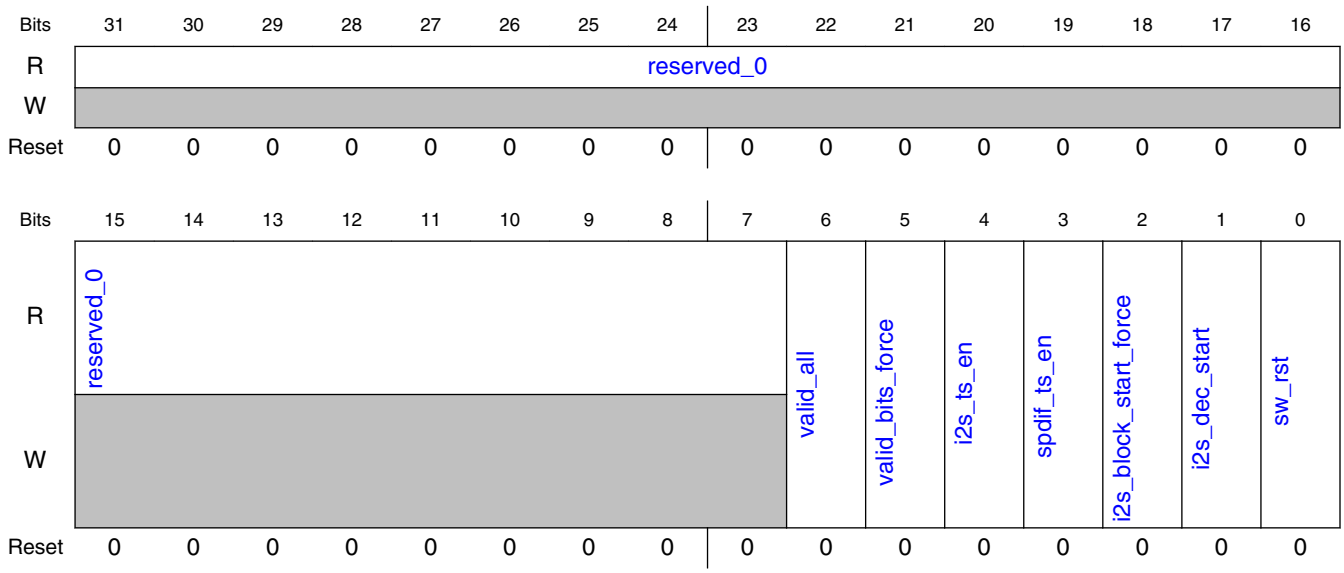
Field	Function
31-0 data	dat to read

13.4.10.1.300 Audio source control (AUDIO_SRC_CNTL)

13.4.10.1.300.1 Offset

Register	Offset
AUDIO_SRC_CNTL	3_0000h

13.4.10.1.300.2 Diagram



13.4.10.1.300.3 Fields

Field	Function
31-7 reserved_0	
6 valid_all	valid bit for all samples
5 valid_bits_force	Force valid bits of the channels
4 i2s_ts_en	Enble I2S Time Stamp when decoders are disabled
3 spdif_ts_en	Enble SPDIF Time Stamp when decoders are disabled
2 i2s_block_start_force	Force a Block Start in the audio stream
1 i2s_dec_start	When high Source Decoder starts. When high Source Decoder starts.

Table continues on the next page...

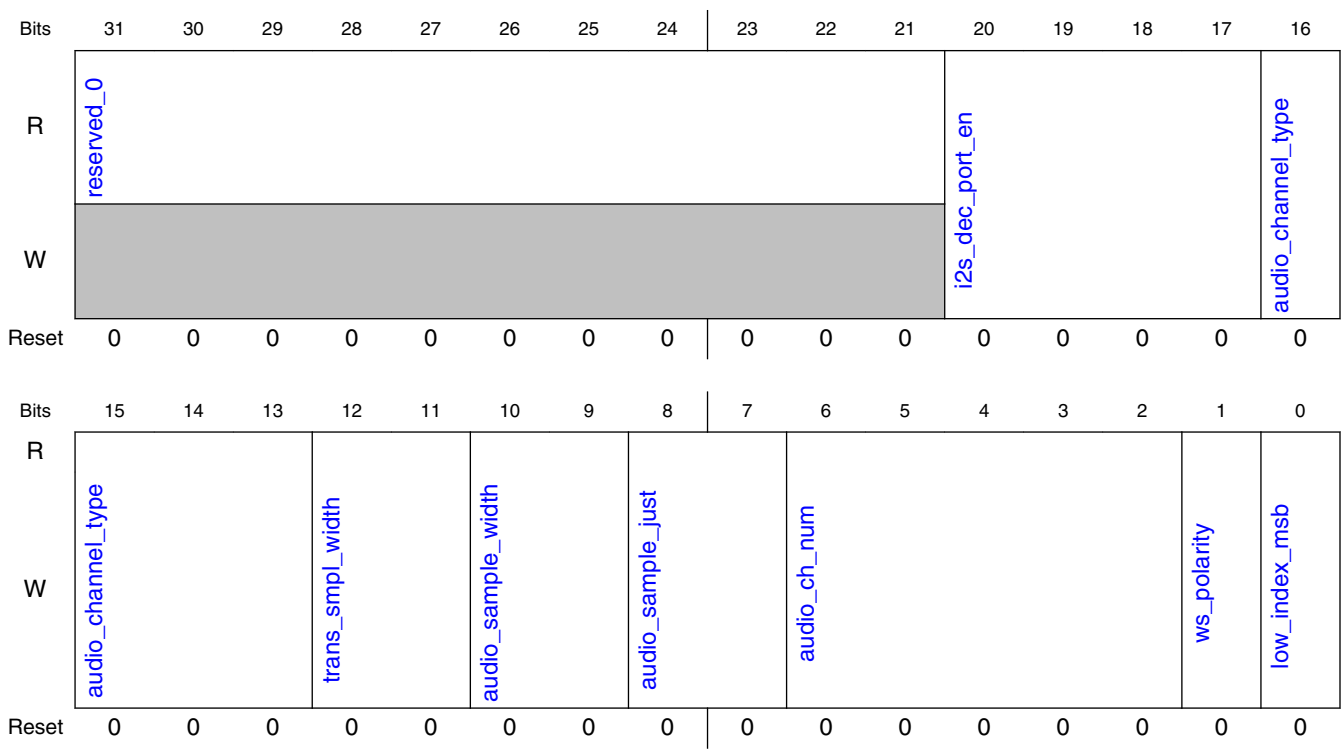
Field	Function
0 sw_rst	Software reset. Software reset. Active high.

13.4.10.1.301 Audio source configuration (AUDIO_SRC_CNFG)

13.4.10.1.301.1 Offset

Register	Offset
AUDIO_SRC_CNFG	3_0004h

13.4.10.1.301.2 Diagram



13.4.10.1.301.3 Fields

Field	Function
31-21 reserved_0	

Table continues on the next page...

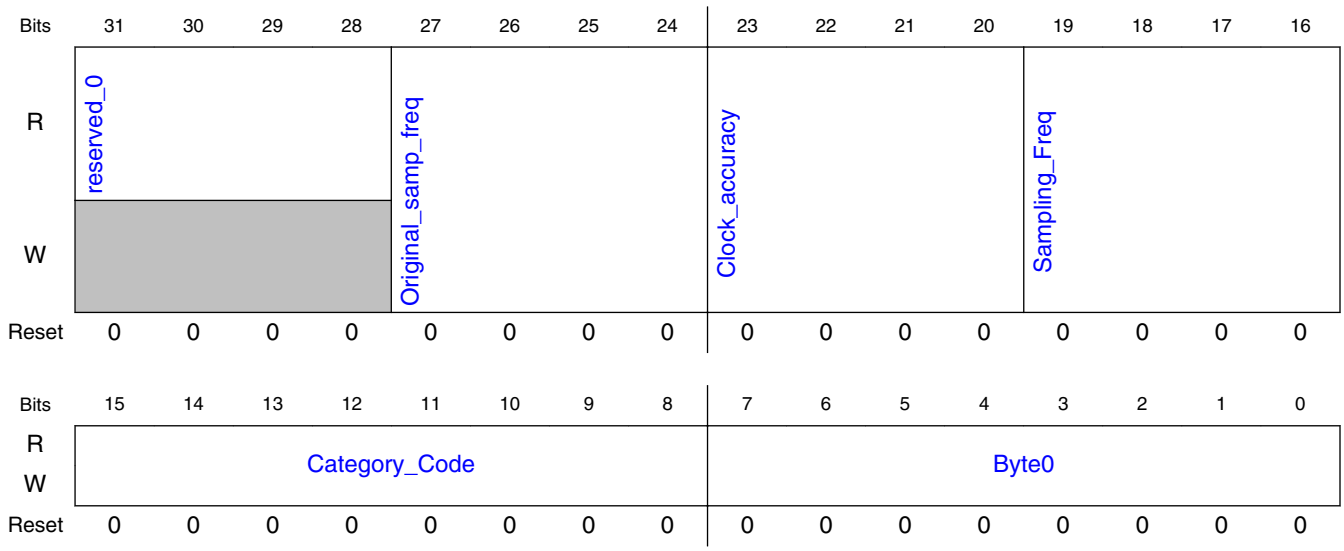
Field	Function
20-17 i2s_dec_port_en	Enables the I2S Decoder ports. Enables the I2S Decoder ports. Allowed values are:0001 - I2S port 0 is enabled.0011 - I2S ports 0,1 are enabled.1111 - I2S ports 0,1,2,3 are enabled. No other values are allowed.
16-13 audio_channel_type	Set the transmission type. Set the transmission type.
12-11 trans_smpl_width	Decoder Word Select width: 00-16 bit, 01-24 bit, 10-32 bit
10-9 audio_sample_width	Decoder sample width:00-16 bit, 01-24 bit, 10-32 bit
8-7 audio_sample_just	Data justification setting:00 left-justified, 01 right-justified
6-2 audio_ch_num	Number of channels to decode
1 ws_polarity	Word Select Polarity. Word Select Polarity. 0: No change, 1: Inverted.
0 low_index_msb	When low MSB is transmitted first. When low MSB is transmitted first. When high LSB is transmitted first.

13.4.10.1.302 Common channels configuration (COM_CH_STTS_BITS)

13.4.10.1.302.1 Offset

Register	Offset
COM_CH_STTS_BITS	3_0008h

13.4.10.1.302.2 Diagram



13.4.10.1.302.3 Fields

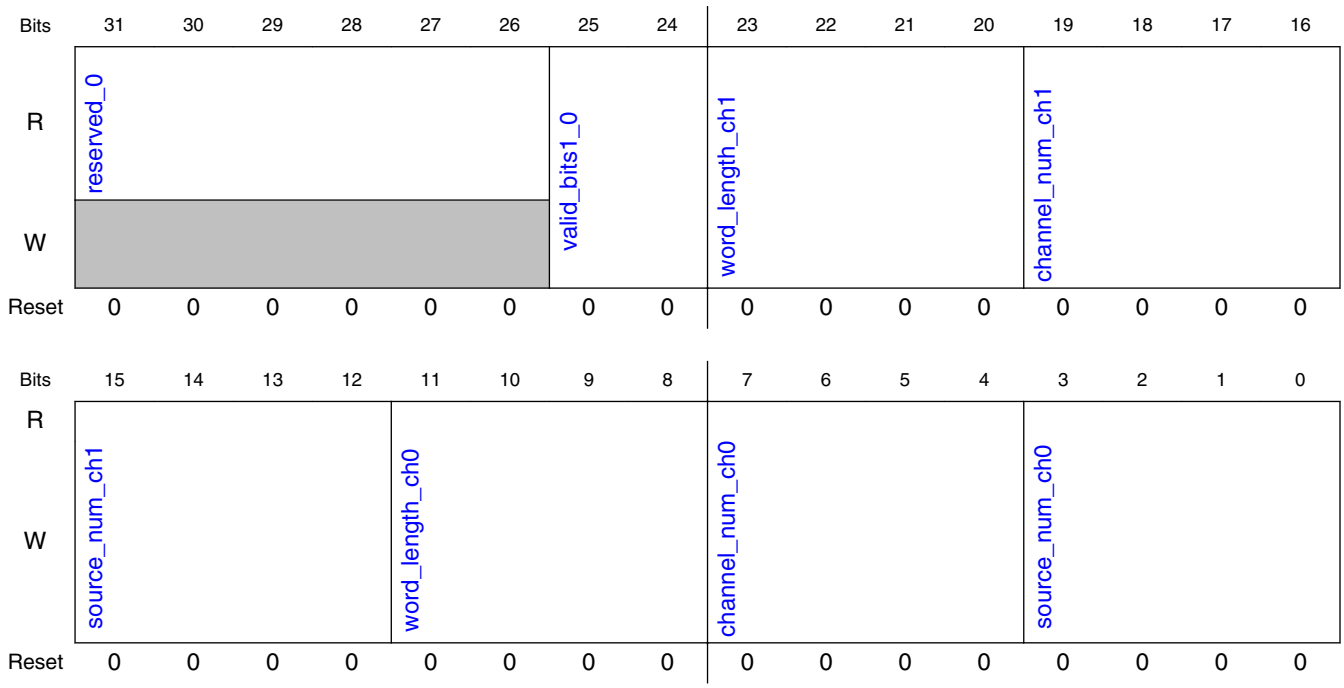
Field	Function
31-28 reserved_0	
27-24 Original_samp_freq	Original Sampling Freq. Original Sampling Freq. of transmitted channel. Same for all channels.
23-20 Clock_accuracy	Clock Accuracy of transmitted channel. Clock Accuracy of transmitted channel. Same for all channels.
19-16 Sampling_Freq	Sampling Frequency of transmitted channel. Sampling Frequency of transmitted channel. Same for all channels.
15-8 Category_Code	Category Code of transmitted channel. Category Code of transmitted channel. Same for all channels.
7-0 Byte0	Byte 0 of transmitted channel. Byte 0 of transmitted channel. Same for all channels.

13.4.10.1.303 Channels 0,1 configuration (STTS_BIT_CH01)

13.4.10.1.303.1 Offset

Register	Offset
STTS_BIT_CH01	3_000Ch

13.4.10.1.303.2 Diagram



13.4.10.1.303.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits1_0	Valid Bits for channel 1 and 0 if force is enabled
23-20 word_length_ch1	Channel 1 word length. Channel 1 word length.
19-16 channel_num_ch1	Channel 1 channel number. Channel 1 channel number.
15-12 source_num_ch1	Channel 1 Source number. Channel 1 Source number.
11-8 word_length_ch0	Channel 0 word length. Channel 0 word length.
7-4	Channel 0 channel number. Channel 0 channel number.

Table continues on the next page...

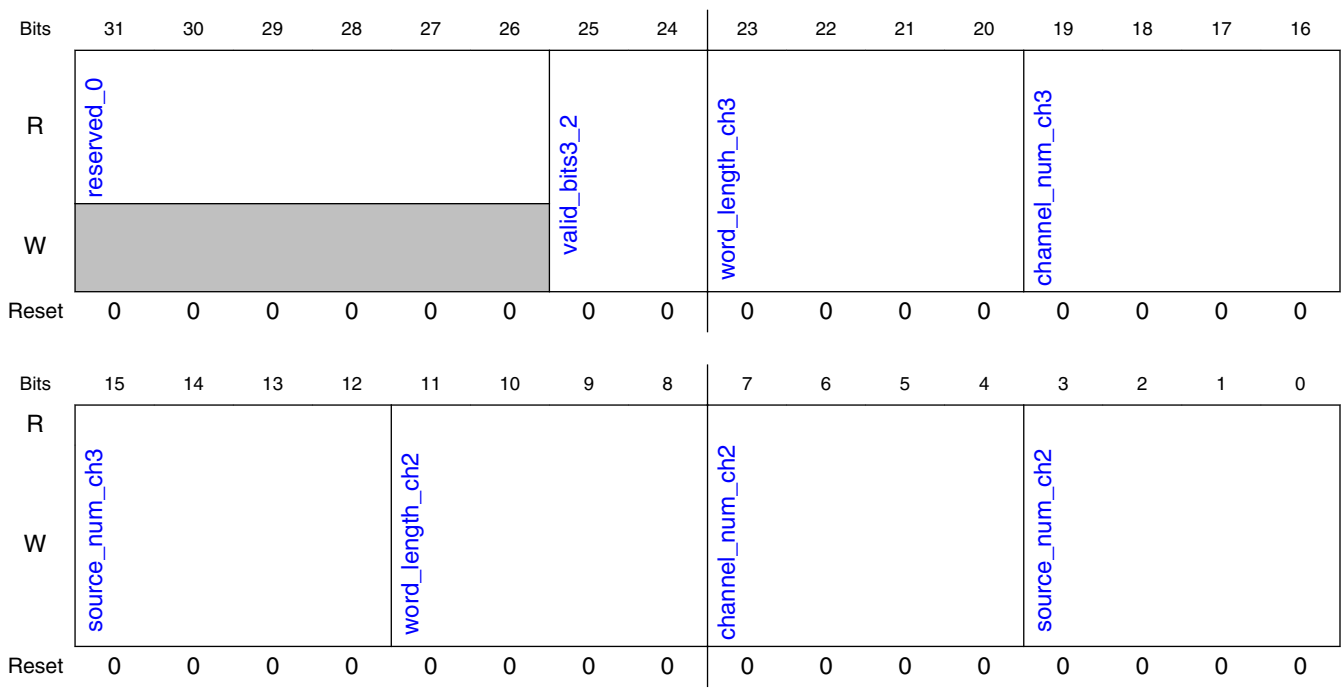
Field	Function
channel_num_ch0	
3-0 source_num_ch0	Channel 0 Source number. Channel 0 Source number.

13.4.10.1.304 Channels 2,3 configuration (STTS_BIT_CH23)

13.4.10.1.304.1 Offset

Register	Offset
STTS_BIT_CH23	3_0010h

13.4.10.1.304.2 Diagram



13.4.10.1.304.3 Fields

Field	Function
31-26	

Table continues on the next page...

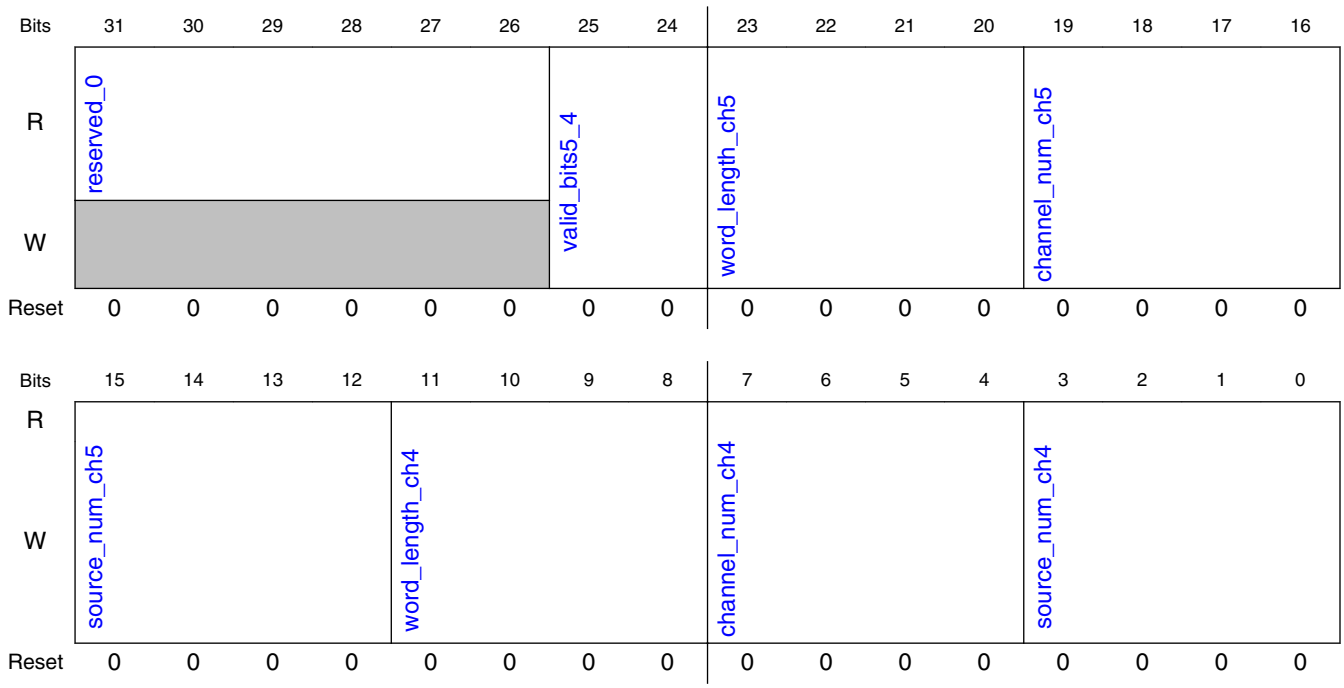
Field	Function
reserved_0	
25-24 valid_bits3_2	Valid Bits for channel 3 and 2 if force is enabled
23-20 word_length_ch3	Channel 3 word length. Channel 3 word length.
19-16 channel_num_ch3	Channel 3 channel number. Channel 3 channel number.
15-12 source_num_ch3	Channel 3 Source number. Channel 3 Source number.
11-8 word_length_ch2	Channel 2 word length. Channel 2 word length.
7-4 channel_num_ch2	Channel 2 channel number. Channel 2 channel number.
3-0 source_num_ch2	Channel 2 Source number. Channel 2 Source number.

13.4.10.1.305 Channels 4,5 configuration (STTS_BIT_CH45)

13.4.10.1.305.1 Offset

Register	Offset
STTS_BIT_CH45	3_0014h

13.4.10.1.305.2 Diagram



13.4.10.1.305.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits5_4	Valid Bits for channel 5 and 4 if force is enabled
23-20 word_length_ch5	Channel 5 word length. Channel 5 word length.
19-16 channel_num_ch5	Channel 5 channel number. Channel 5 channel number.
15-12 source_num_ch5	Channel 5 Source number. Channel 5 Source number.
11-8 word_length_ch4	Channel 4 word length. Channel 4 word length.
7-4 channel_num_ch4	Channel 4 channel number. Channel 4 channel number.

Table continues on the next page...

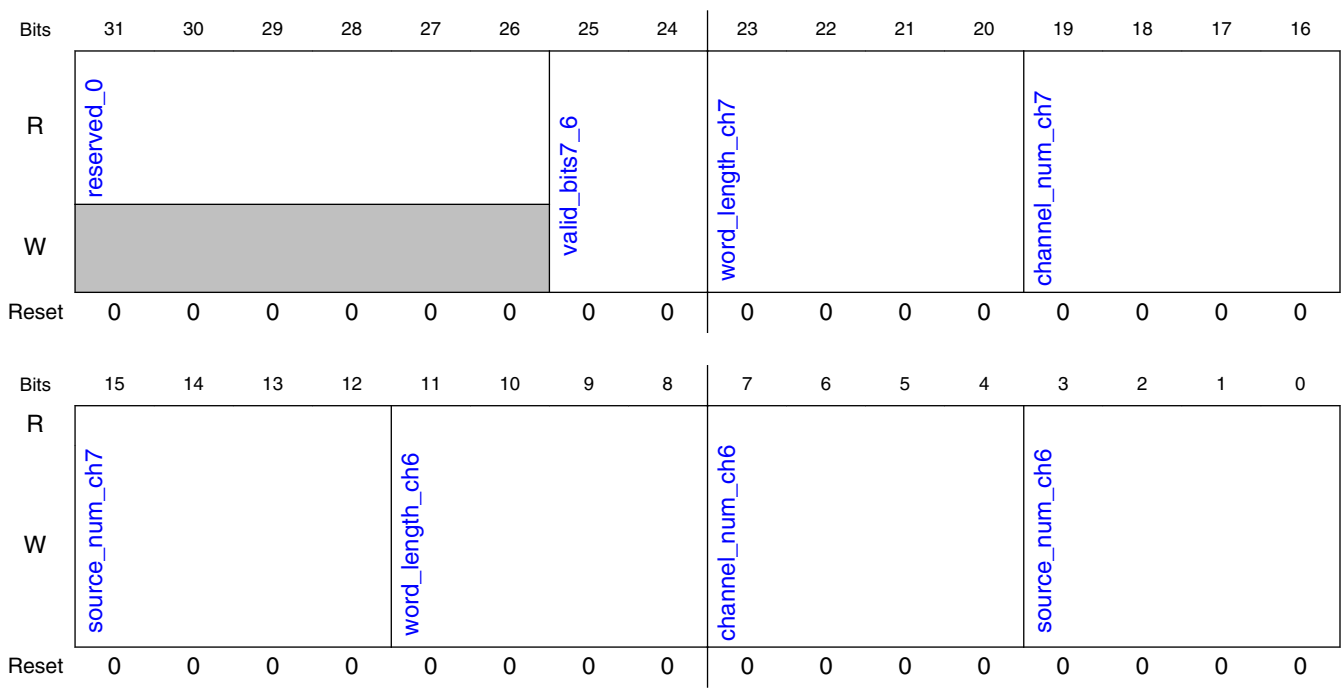
Field	Function
3-0 source_num_ch4	Channel 4 Source number. Channel 4 Source number.

13.4.10.1.306 Channels 6,7 configuration (STTS_BIT_CH67)

13.4.10.1.306.1 Offset

Register	Offset
STTS_BIT_CH67	3_0018h

13.4.10.1.306.2 Diagram



13.4.10.1.306.3 Fields

Field	Function
31-26 reserved_0	
25-24	Valid Bits for channel 7 and 6 if force is enabled

Table continues on the next page...

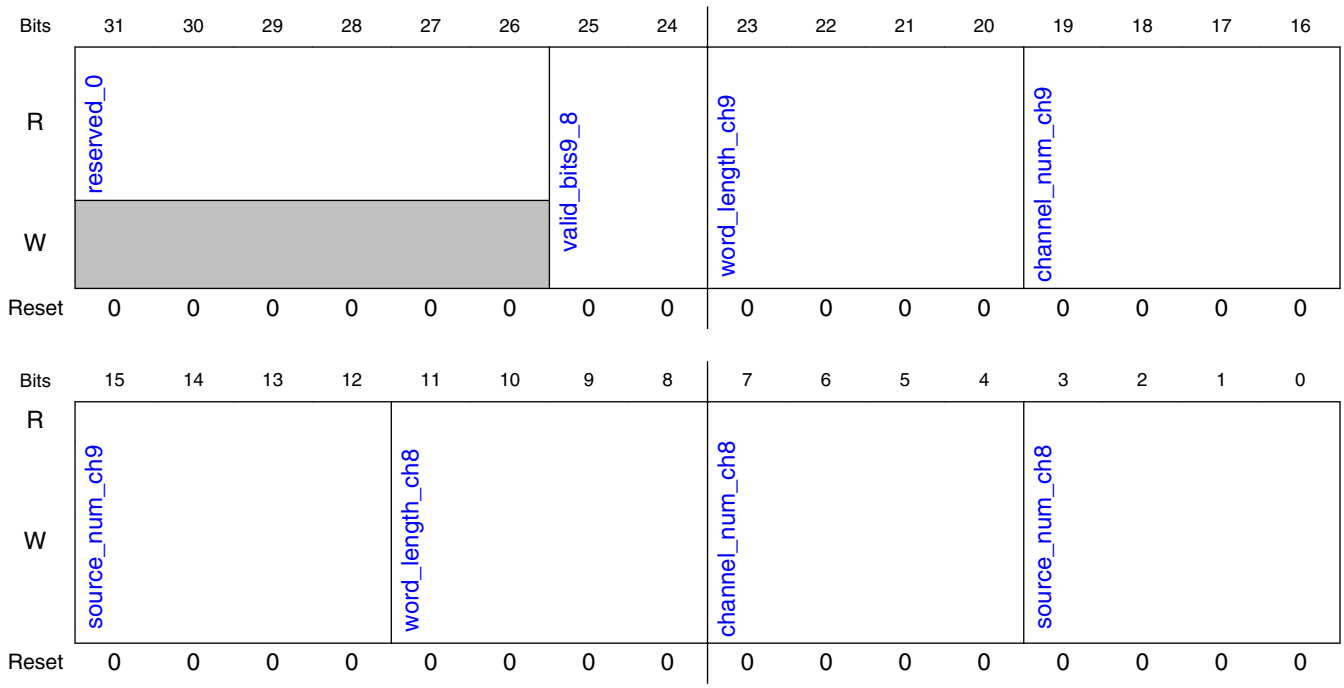
Field	Function
valid_bits7_6	
23-20 word_length_ch7	Channel 7 word length. Channel 7 word length.
19-16 channel_num_ch7	Channel 7 channel number. Channel 7 channel number.
15-12 source_num_ch7	Channel 7 Source number. Channel 7 Source number.
11-8 word_length_ch6	Channel 6 word length. Channel 6 word length.
7-4 channel_num_ch6	Channel 6 channel number. Channel 6 channel number.
3-0 source_num_ch6	Channel 6 Source number. Channel 6 Source number.

13.4.10.1.307 Channels 8,9 configuration (STTS_BIT_CH89)

13.4.10.1.307.1 Offset

Register	Offset
STTS_BIT_CH89	3_001Ch

13.4.10.1.307.2 Diagram



13.4.10.1.307.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits9_8	Valid Bits for channel 9 and 8 if force is enabled
23-20 word_length_ch9	Channel 9 word length. Channel 9 word length.
19-16 channel_num_ch9	Channel 9 channel number. Channel 9 channel number.
15-12 source_num_ch9	Channel 9 Source number. Channel 9 Source number.
11-8 word_length_ch8	Channel 8 word length. Channel 8 word length.
7-4 channel_num_ch8	Channel 8 channel number. Channel 8 channel number.

Table continues on the next page...

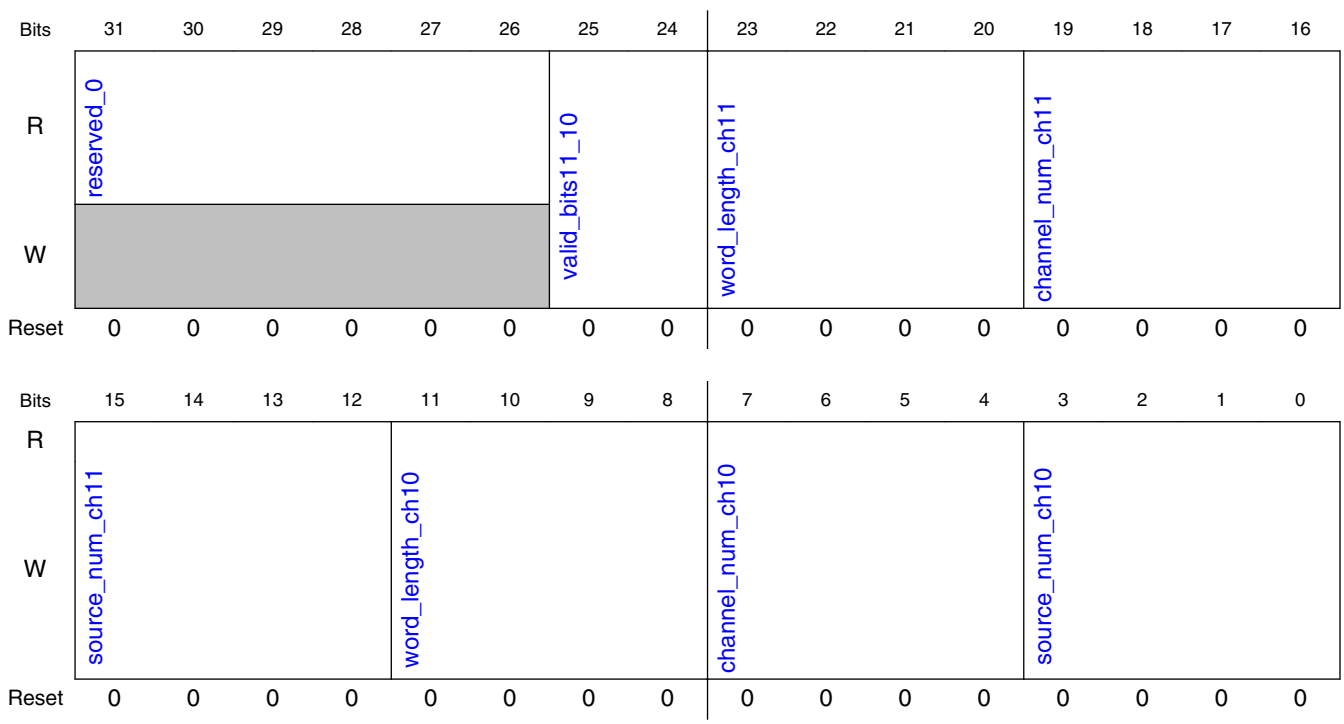
Field	Function
3-0 source_num_ch8	Channel 8 Source number. Channel 8 Source number.

13.4.10.1.308 Channels 10,11 configuration (STTS_BIT_CH1011)

13.4.10.1.308.1 Offset

Register	Offset
STTS_BIT_CH1011	3_0020h

13.4.10.1.308.2 Diagram



13.4.10.1.308.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

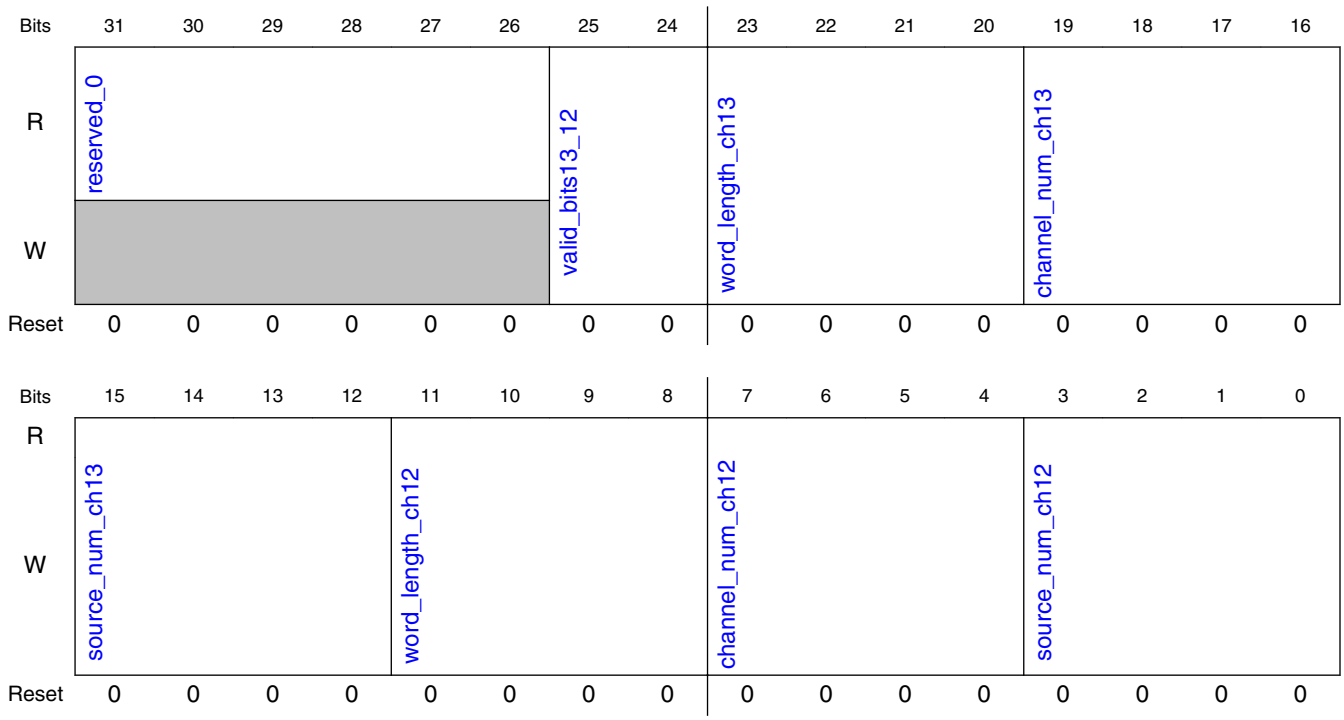
Field	Function
25-24 valid_bits11_10	Valid Bits for channel 11 and 10 if force is enabled
23-20 word_length_ch11	Channel 11 word length. Channel 11 word length.
19-16 channel_num_ch11	Channel 11 channel number. Channel 11 channel number.
15-12 source_num_ch11	Channel 11 Source number. Channel 11 Source number.
11-8 word_length_ch10	Channel 10 word length. Channel 10 word length.
7-4 channel_num_ch10	Channel 10 channel number. Channel 10 channel number.
3-0 source_num_ch10	Channel 10 Source number. Channel 10 Source number.

13.4.10.1.309 Channels 12,13 configuration (STTS_BIT_CH1213)

13.4.10.1.309.1 Offset

Register	Offset
STTS_BIT_CH1213	3_0024h

13.4.10.1.309.2 Diagram



13.4.10.1.309.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits13_12	Valid Bits for channel 13 and 12 if force is enabled
23-20 word_length_ch13	Channel 13 word length. Channel 13 word length.
19-16 channel_num_ch13	Channel 13 channel number. Channel 13 channel number.
15-12 source_num_ch13	Channel 13 Source number. Channel 13 Source number.
11-8 word_length_ch12	Channel 12 word length. Channel 12 word length.
7-4 channel_num_ch12	Channel 12 channel number. Channel 12 channel number.

Table continues on the next page...

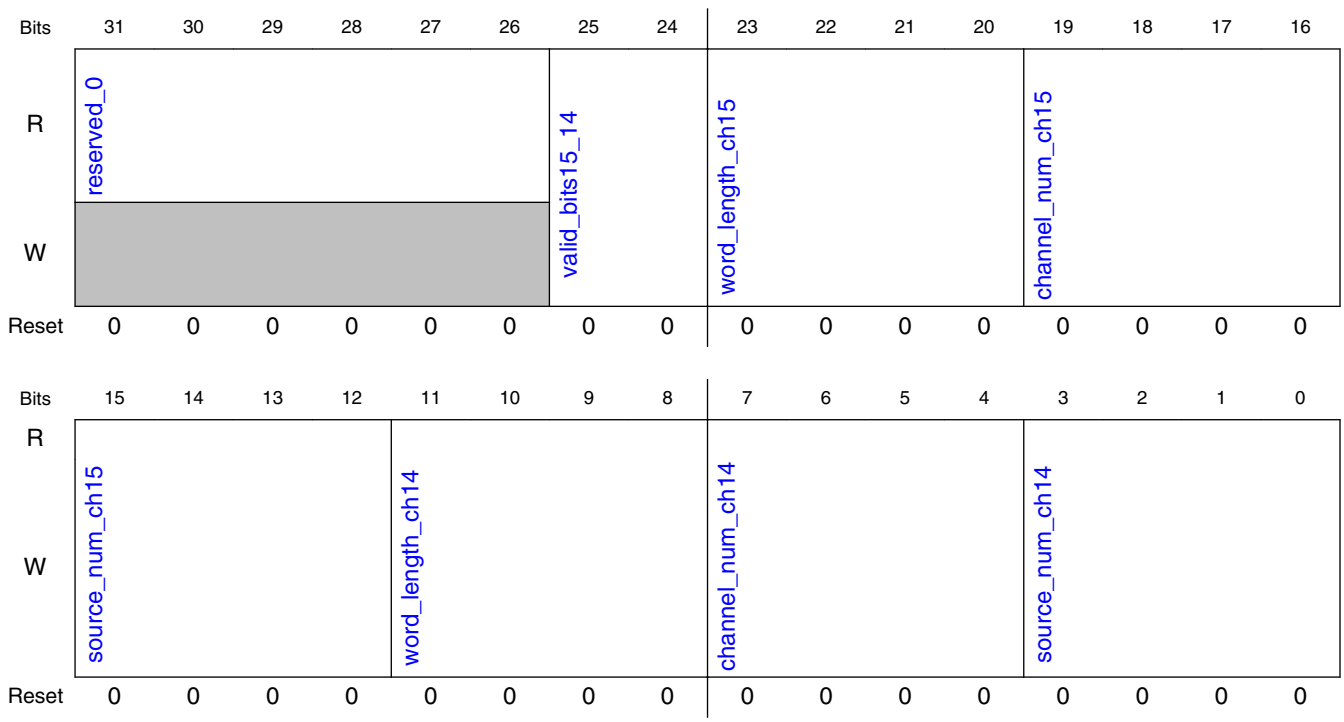
Field	Function
3-0 source_num_ch12	Channel 12 Source number. Channel 12 Source number.

13.4.10.1.310 Channels 14,15 configuration (STTS_BIT_CH1415)

13.4.10.1.310.1 Offset

Register	Offset
STTS_BIT_CH1415	3_0028h

13.4.10.1.310.2 Diagram



13.4.10.1.310.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

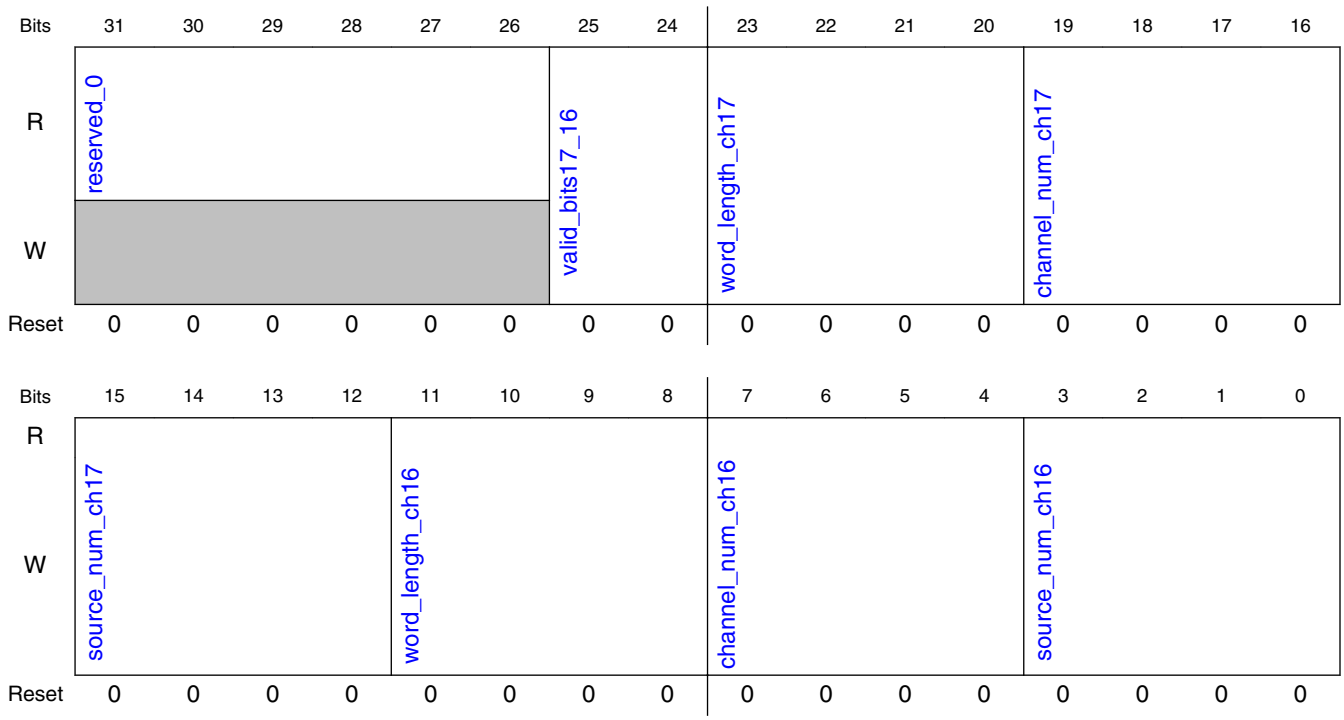
Field	Function
25-24 valid_bits15_14	Valid Bits for channel 15 and 14 if force is enabled
23-20 word_length_ch15	Channel 15 word length. Channel 15 word length.
19-16 channel_num_ch15	Channel 15 channel number. Channel 15 channel number.
15-12 source_num_ch15	Channel 15 Source number. Channel 15 Source number.
11-8 word_length_ch14	Channel 14 word length. Channel 14 word length.
7-4 channel_num_ch14	Channel 14 channel number. Channel 14 channel number.
3-0 source_num_ch14	Channel 14 Source number. Channel 14 Source number.

13.4.10.1.311 Channels 16,17 configuration (STTS_BIT_CH1617)

13.4.10.1.311.1 Offset

Register	Offset
STTS_BIT_CH1617	3_002Ch

13.4.10.1.311.2 Diagram



13.4.10.1.311.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits17_16	Valid Bits for channel 17 and 16 if force is enabled
23-20 word_length_ch17	Channel 17 word length. Channel 17 word length.
19-16 channel_num_ch17	Channel 17 channel number. Channel 17 channel number.
15-12 source_num_ch17	Channel 17 Source number. Channel 17 Source number.
11-8 word_length_ch16	Channel 16 word length. Channel 16 word length.
7-4 channel_num_ch16	Channel 16 channel number. Channel 16 channel number.

Table continues on the next page...

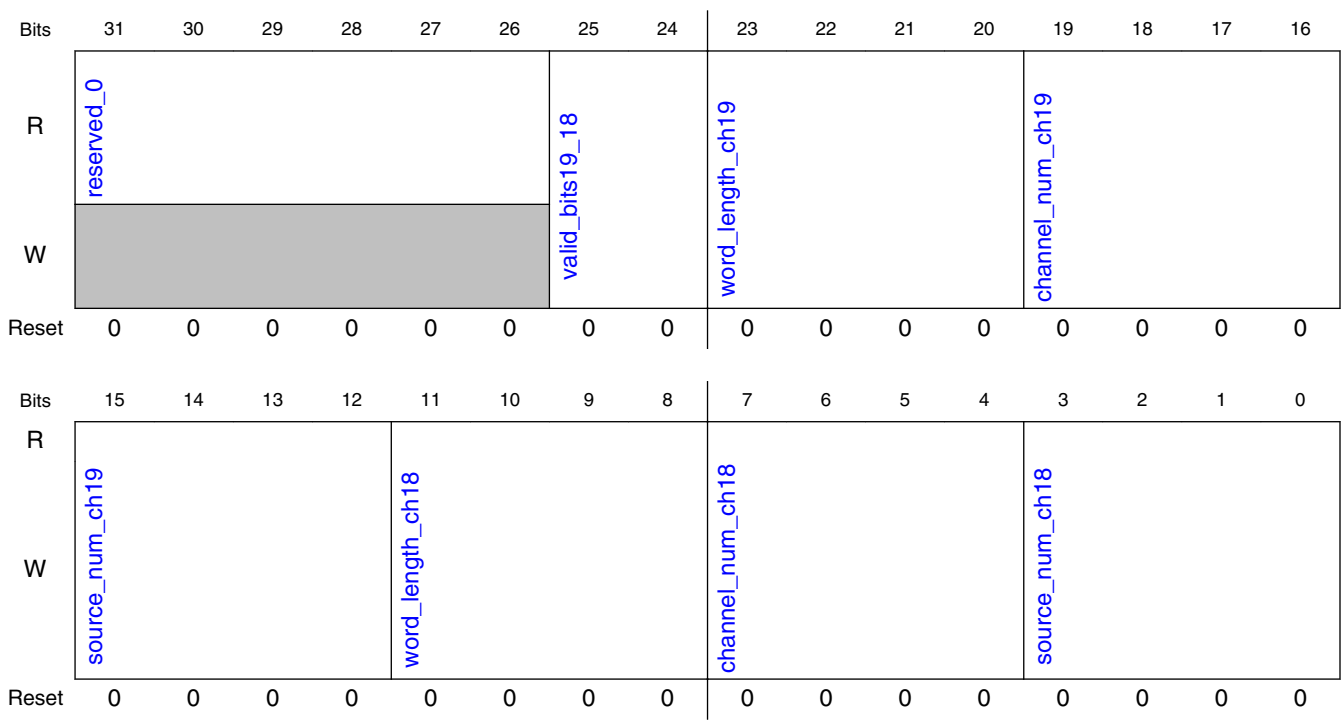
Field	Function
3-0 source_num_ch16	Channel 16 Source number. Channel 16 Source number.

13.4.10.1.312 Channels 18,19 configuration (STTS_BIT_CH1819)

13.4.10.1.312.1 Offset

Register	Offset
STTS_BIT_CH1819	3_0030h

13.4.10.1.312.2 Diagram



13.4.10.1.312.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

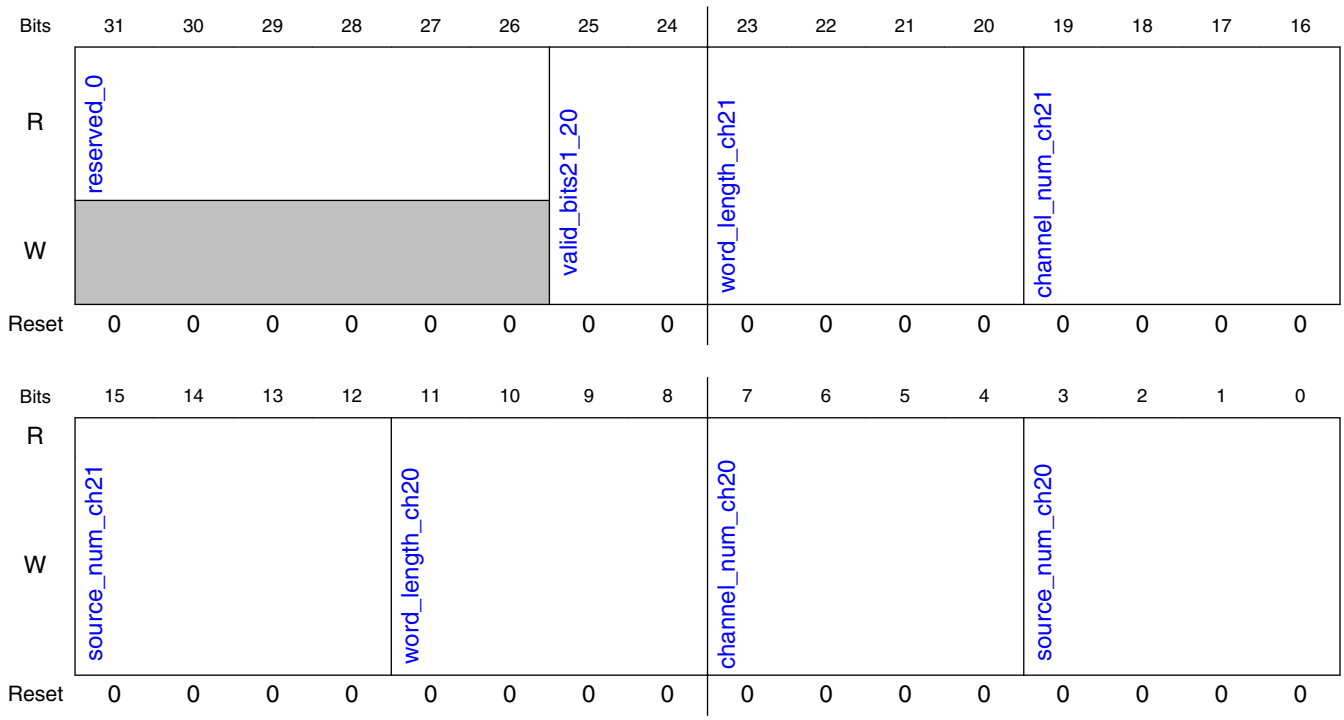
Field	Function
25-24 valid_bits19_18	Valid Bits for channel 19 and 18 if force is enabled
23-20 word_length_ch19	Channel 19 word length. Channel 19 word length.
19-16 channel_num_ch19	Channel 19 channel number. Channel 19 channel number.
15-12 source_num_ch19	Channel 19 Source number. Channel 19 Source number.
11-8 word_length_ch18	Channel 18 word length. Channel 18 word length.
7-4 channel_num_ch18	Channel 18 channel number. Channel 18 channel number.
3-0 source_num_ch18	Channel 18 Source number. Channel 18 Source number.

13.4.10.1.313 Channels 20,21 configuration (STTS_BIT_CH2021)

13.4.10.1.313.1 Offset

Register	Offset
STTS_BIT_CH2021	3_0034h

13.4.10.1.313.2 Diagram



13.4.10.1.313.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits21_20	Valid Bits for channel 21 and 20 if force is enabled
23-20 word_length_ch21	Channel 21 word length. Channel 21 word length.
19-16 channel_num_ch21	Channel 21 channel number. Channel 21 channel number.
15-12 source_num_ch21	Channel 21 Source number. Channel 21 Source number.
11-8 word_length_ch20	Channel 20 word length. Channel 20 word length.
7-4 channel_num_ch20	Channel 20 channel number. Channel 20 channel number.

Table continues on the next page...

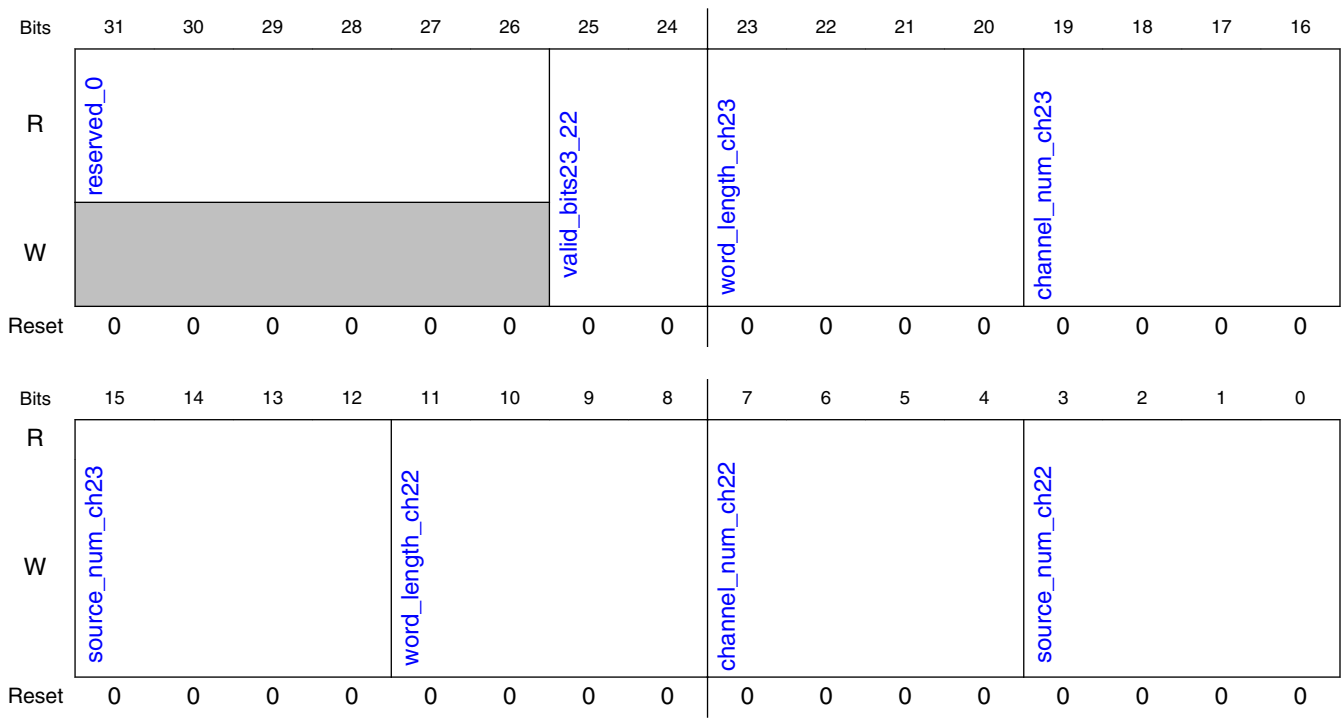
Field	Function
3-0 source_num_ch20	Channel 20 Source number. Channel 20 Source number.

13.4.10.1.314 Channels 22,23 configuration (STTS_BIT_CH2223)

13.4.10.1.314.1 Offset

Register	Offset
STTS_BIT_CH2223	3_0038h

13.4.10.1.314.2 Diagram



13.4.10.1.314.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

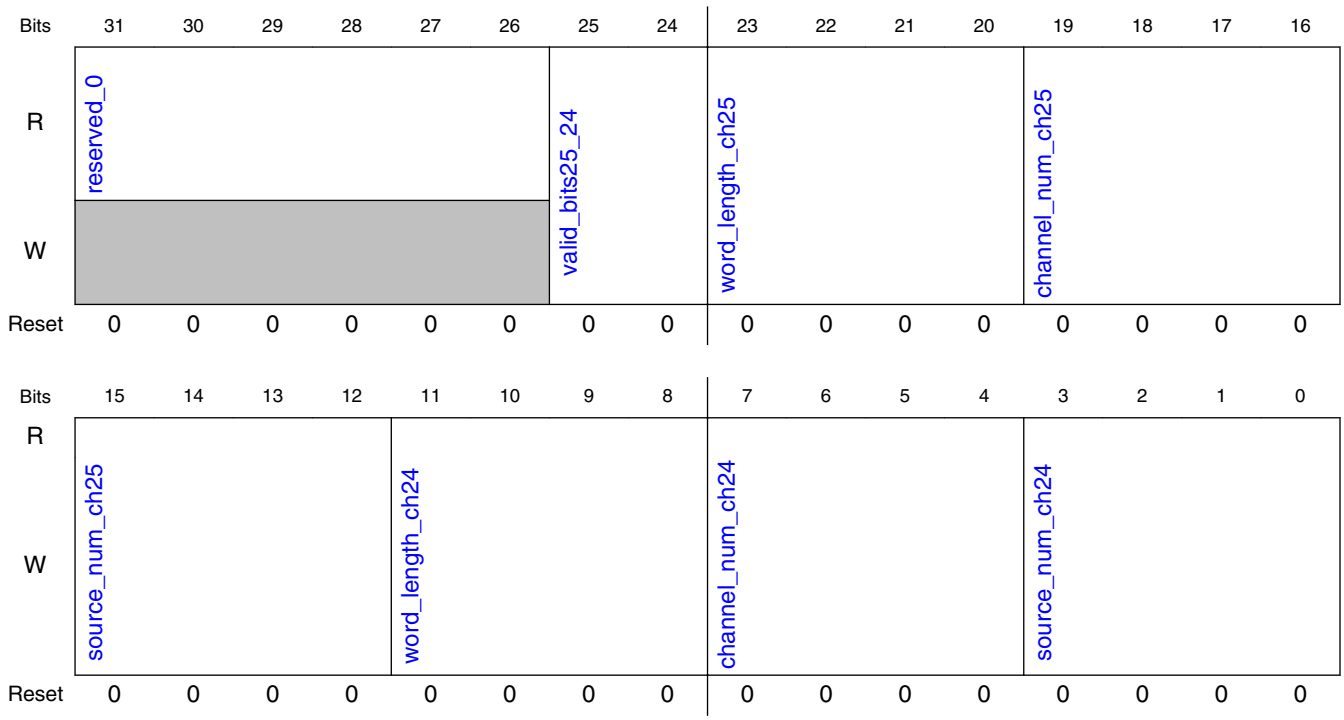
Field	Function
25-24 valid_bits23_22	Valid Bits for channel 23 and 22 if force is enabled
23-20 word_length_ch23	Channel 23 word length. Channel 23 word length.
19-16 channel_num_ch23	Channel 23 channel number. Channel 23 channel number.
15-12 source_num_ch23	Channel 23 Source number. Channel 23 Source number.
11-8 word_length_ch22	Channel 22 word length. Channel 22 word length.
7-4 channel_num_ch22	Channel 22 channel number. Channel 22 channel number.
3-0 source_num_ch22	Channel 22 Source number. Channel 22 Source number.

13.4.10.1.315 Channels 24,25 configuration (STTS_BIT_CH2425)

13.4.10.1.315.1 Offset

Register	Offset
STTS_BIT_CH2425	3_003Ch

13.4.10.1.315.2 Diagram



13.4.10.1.315.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits25_24	Valid Bits for channel 25 and 24 if force is enabled
23-20 word_length_ch25	Channel 25 word length. Channel 25 word length.
19-16 channel_num_ch25	Channel 25 channel number. Channel 25 channel number.
15-12 source_num_ch25	Channel 25 Source number. Channel 25 Source number.
11-8 word_length_ch24	Channel 24 word length. Channel 24 word length.
7-4 channel_num_ch24	Channel 24 channel number. Channel 24 channel number.

Table continues on the next page...

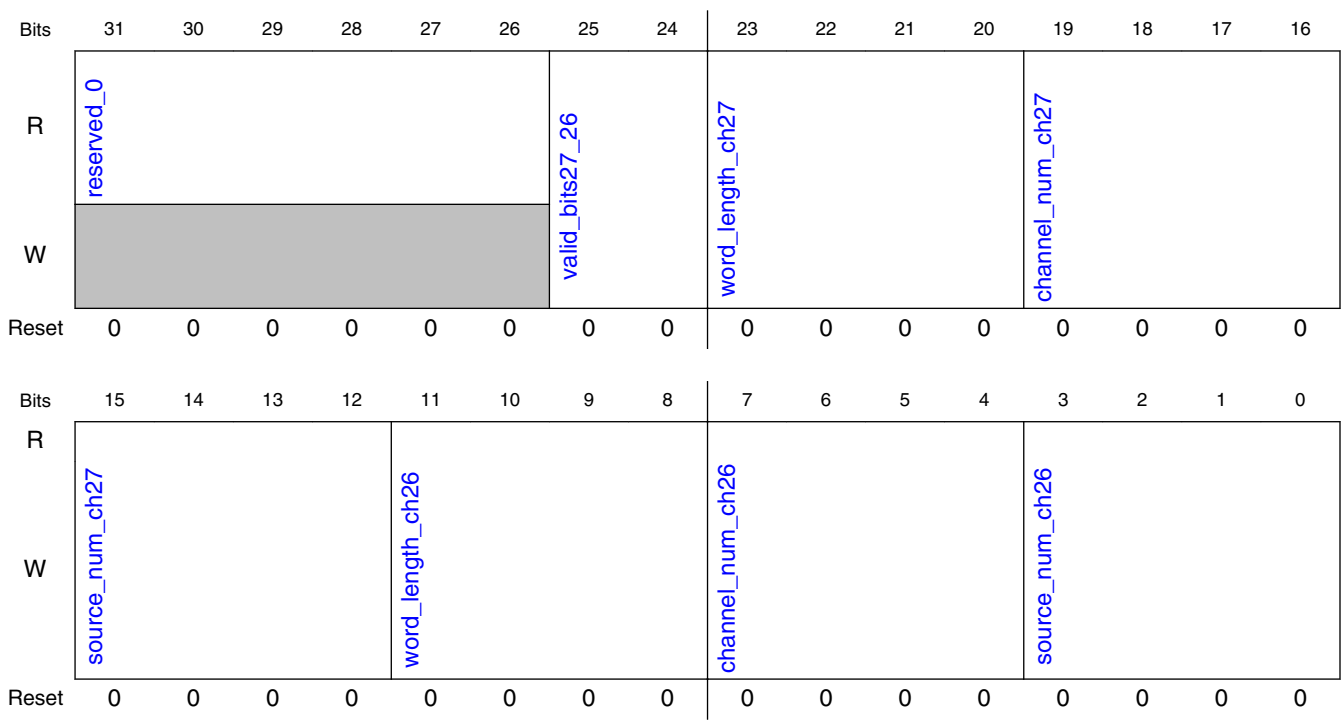
Field	Function
3-0 source_num_ch24	Channel 24 Source number. Channel 24 Source number.

13.4.10.1.316 Channels 26,27 configuration (STTS_BIT_CH2627)

13.4.10.1.316.1 Offset

Register	Offset
STTS_BIT_CH2627	3_0040h

13.4.10.1.316.2 Diagram



13.4.10.1.316.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

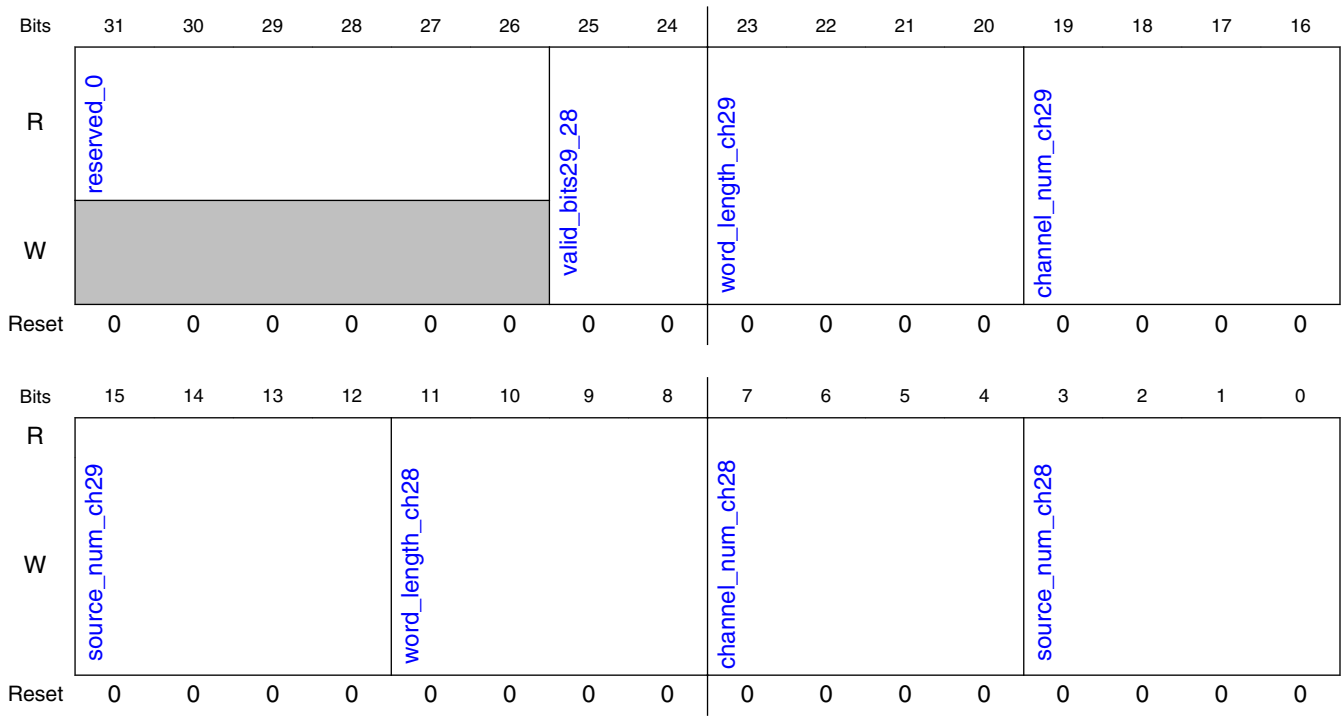
Field	Function
25-24 valid_bits27_26	Valid Bits for channel 27 and 26 if force is enabled
23-20 word_length_ch27	Channel 27 word length. Channel 27 word length.
19-16 channel_num_ch27	Channel 27 channel number. Channel 27 channel number.
15-12 source_num_ch27	Channel 27 Source number. Channel 27 Source number.
11-8 word_length_ch26	Channel 26 word length. Channel 26 word length.
7-4 channel_num_ch26	Channel 26 channel number. Channel 26 channel number.
3-0 source_num_ch26	Channel 26 Source number. Channel 26 Source number.

13.4.10.1.317 Channels 28,29 configuration (STTS_BIT_CH2829)

13.4.10.1.317.1 Offset

Register	Offset
STTS_BIT_CH2829	3_0044h

13.4.10.1.317.2 Diagram



13.4.10.1.317.3 Fields

Field	Function
31-26 reserved_0	
25-24 valid_bits29_28	Valid Bits for channel 29 and 28 if force is enabled
23-20 word_length_ch29	Channel 29 word length. Channel 29 word length.
19-16 channel_num_ch29	Channel 29 channel number. Channel 29 channel number.
15-12 source_num_ch29	Channel 29 Source number. Channel 29 Source number.
11-8 word_length_ch28	Channel 28 word length. Channel 28 word length.
7-4 channel_num_ch28	Channel 28 channel number. Channel 28 channel number.

Table continues on the next page...

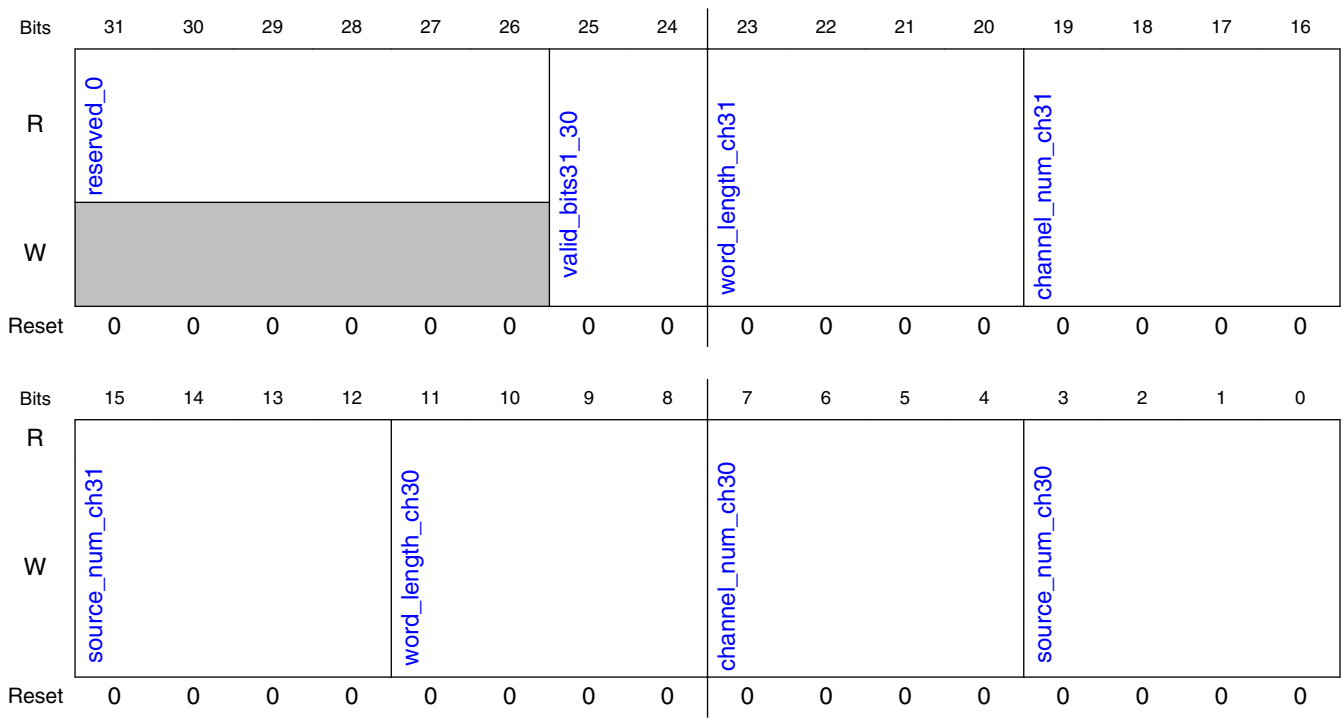
Field	Function
3-0 source_num_ch28	Channel 28 Source number. Channel 28 Source number.

13.4.10.1.318 Channels 30,31 configuration (STTS_BIT_CH3031)

13.4.10.1.318.1 Offset

Register	Offset
STTS_BIT_CH3031	3_0048h

13.4.10.1.318.2 Diagram



13.4.10.1.318.3 Fields

Field	Function
31-26 reserved_0	

Table continues on the next page...

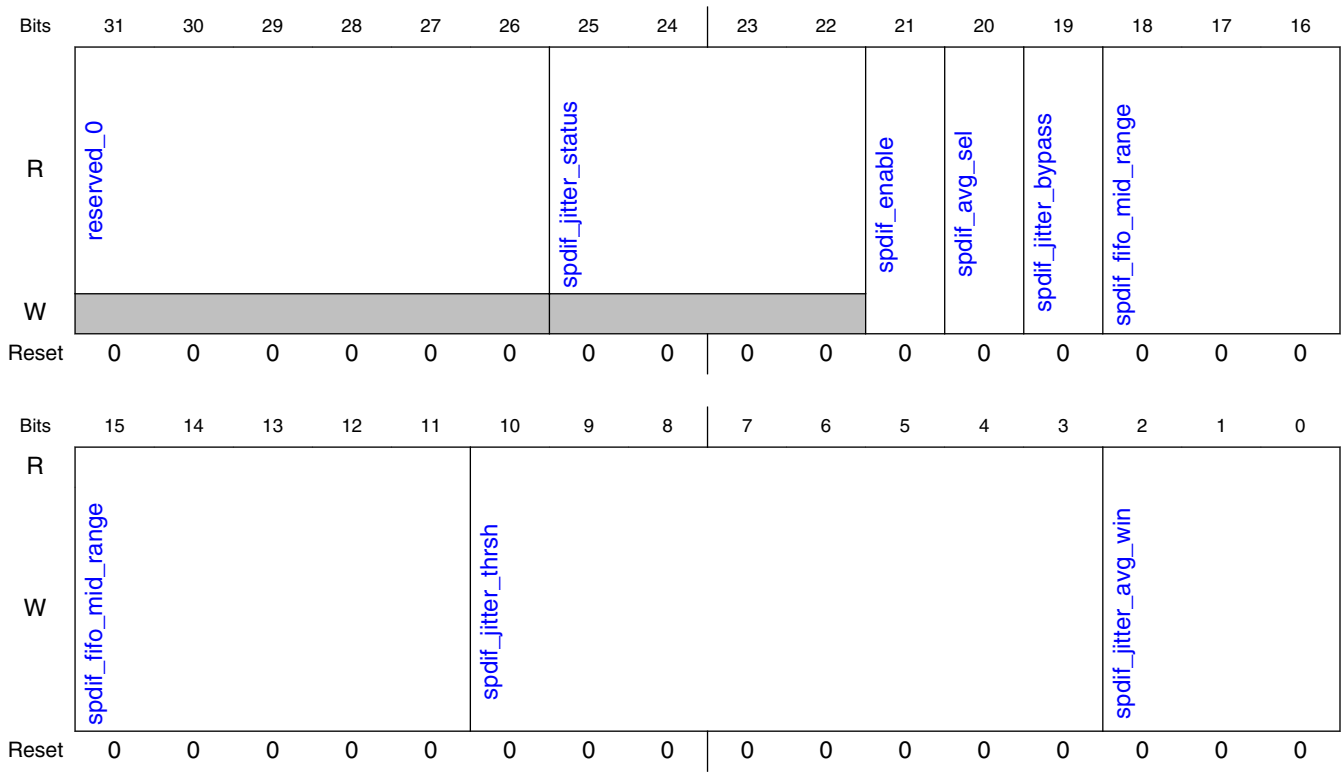
Field	Function
25-24 valid_bits31_30	Valid Bits for channel 31 and 30 if force is enabled
23-20 word_length_ch31	Channel 31 word length. Channel 31 word length.
19-16 channel_num_ch31	Channel 31 channel number. Channel 31 channel number.
15-12 source_num_ch31	Channel 31 Source number. Channel 31 Source number.
11-8 word_length_ch30	Channel 30 word length. Channel 30 word length.
7-4 channel_num_ch30	Channel 30 channel number. Channel 30 channel number.
3-0 source_num_ch30	Channel 30 Source number. Channel 30 Source number.

13.4.10.1.319 SPDIF control (SPDIF_CTRL_ADDR)

13.4.10.1.319.1 Offset

Register	Offset
SPDIF_CTRL_ADDR	3_004Ch

13.4.10.1.319.2 Diagram



13.4.10.1.319.3 Fields

Field	Function
31-26 reserved_0	
25-22 spdif_jitter_status	SPDIF Jitter Status
21 spdif_enable	SPDIF Enable
20 spdif_avg_sel	SPDIF average Select
19 spdif_jitter_bypass	SPDIF Jitter Bypass
18-11 spdif_fifo_mid_range	SPDIF fifo mid range
10-3 spdif_jitter_thrsh	SPDIF Jitter threshold

Table continues on the next page...

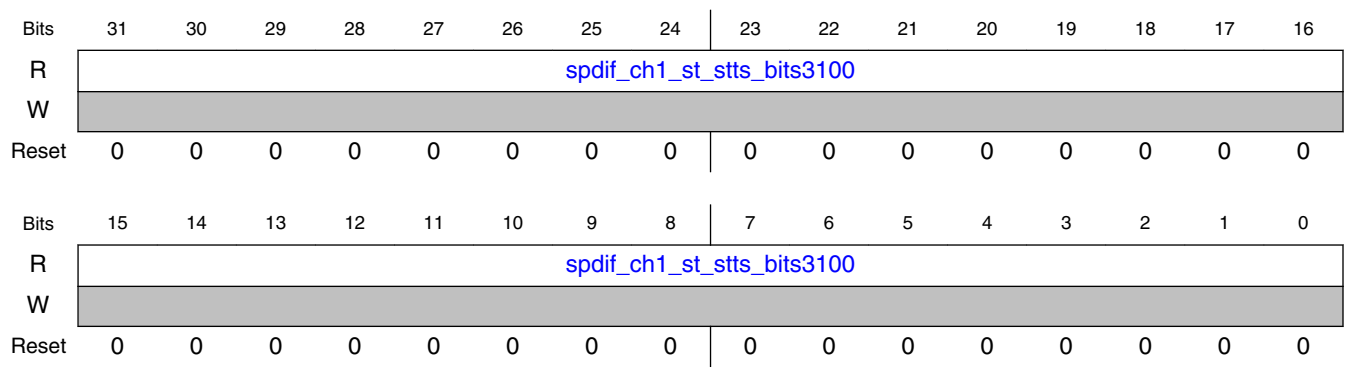
Field	Function
2-0 spdif_jitter_avg_win	Spdif Jitter AVG Window

13.4.10.1.320 SPDIF channel 1 status [31:00] (SPDIF_CH1_CS_3100_ADDR)

13.4.10.1.320.1 Offset

Register	Offset
SPDIF_CH1_CS_3100_ADDR	3_0050h

13.4.10.1.320.2 Diagram



13.4.10.1.320.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits3100	SPDIF Channel 1 Status bits[31:0]

13.4.10.1.321 SPDIF channel 1 status [63:32] (SPDIF_CH1_CS_6332_ADDR)

13.4.10.1.321.1 Offset

Register	Offset
SPDIF_CH1_CS_6332_ADDR	3_0054h

13.4.10.1.321.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch1_st_stts_bits6332															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch1_st_stts_bits6332															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.321.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits6332	SPDIF Channel 1 Status bits[63:32]

13.4.10.1.322 SPDIF channel 1 status [95:64] (SPDIF_CH1_CS_9564_ADDR)

13.4.10.1.322.1 Offset

Register	Offset
SPDIF_CH1_CS_9564_ADDR	3_0058h

13.4.10.1.322.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch1_st_stts_bits9564															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch1_st_stts_bits9564															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.322.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits9564	SPDIF Channel 1 Status bits[95:64]

13.4.10.1.323 SPDIF channel 1 status [127:96] (SPDIF_CH1_CS_12796_ADDR)

13.4.10.1.323.1 Offset

Register	Offset
SPDIF_CH1_CS_12796_ADDR	3_005Ch

13.4.10.1.323.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch1_st_stts_bits12796															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch1_st_stts_bits12796															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.323.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits12796	SPDIF Channel 1 Status bits[127:96]

13.4.10.1.324 SPDIF channel 1 status [159:128] (SPDIF_CH1_CS_159128_ADDR)

13.4.10.1.324.1 Offset

Register	Offset
SPDIF_CH1_CS_159128_ADDR	3_0060h

13.4.10.1.324.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch1_st_stts_bits159128															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch1_st_stts_bits159128															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.324.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits159128	SPDIF Channel 1 Status bits[159:128]

13.4.10.1.325 SPDIF channel 1 status [191:160] (SPDIF_CH1_CS_191160_ADDR)

13.4.10.1.325.1 Offset

Register	Offset
SPDIF_CH1_CS_191160_ADDR	3_0064h

13.4.10.1.325.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch1_st_stts_bits191160															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch1_st_stts_bits191160															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.325.3 Fields

Field	Function
31-0 spdif_ch1_st_stts_bits191160	SPDIF Channel 1 Status bits[191:160]

13.4.10.1.326 SPDIF channel 2 status [31:00] (SPDIF_CH2_CS_3100_ADDR)

13.4.10.1.326.1 Offset

Register	Offset
SPDIF_CH2_CS_3100_ADDR	3_0068h

13.4.10.1.326.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits3100															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits3100															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.326.3 Fields

Field	Function
31-0 spdif_ch2_st_stts_bits3100	SPDIF Channel 2 Status bits[31:0]

13.4.10.1.327 SPDIF channel 2 status [63:32] (SPDIF_CH2_CS_6332_ADDR)

13.4.10.1.327.1 Offset

Register	Offset
SPDIF_CH2_CS_6332_ADDR	3_006Ch

13.4.10.1.327.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits6332															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits6332															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.327.3 Fields

Field	Function
31-0 spdif_ch2_st_stts_bits6332	SPDIF Channel 2 Status bits[63:32]

13.4.10.1.328 SPDIF channel 2 status [95:64] (SPDIF_CH2_CS_9564_ADDR)

13.4.10.1.328.1 Offset

Register	Offset
SPDIF_CH2_CS_9564_ADDR	3_0070h

13.4.10.1.328.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits9564															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits9564															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.328.3 Fields

Field	Function
31-0 spdif_ch2_st_stts_bits9564	SPDIF Channel 2 Status bits[95:64]

13.4.10.1.329 SPDIF channel 2 status [127:96] (SPDIF_CH2_CS_12796_ADDR)

13.4.10.1.329.1 Offset

Register	Offset
SPDIF_CH2_CS_12796_ADDR	3_0074h

13.4.10.1.329.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits12796															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits12796															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.329.3 Fields

Field	Function
31-0 spdif_ch2_st_stts_bits12796	SPDIF Channel 2 Status bits[127:96]

13.4.10.1.330 SPDIF channel 2 status [159:128] (SPDIF_CH2_CS_159128_ADDR)

13.4.10.1.330.1 Offset

Register	Offset
SPDIF_CH2_CS_159128_ADDR	3_0078h

13.4.10.1.330.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits159128															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits159128															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.330.3 Fields

Field	Function
31-0 spdif_ch2_st_stts_bits159128	SPDIF Channel 2 Status bits[159:128]

13.4.10.1.331 SPDIF channel 2 status [191:160] (SPDIF_CH2_CS_191160_ADDR)

13.4.10.1.331.1 Offset

Register	Offset
SPDIF_CH2_CS_191160_ADDR	3_007Ch

13.4.10.1.331.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	spdif_ch2_st_stts_bits191160															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	spdif_ch2_st_stts_bits191160															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.331.3 Fields

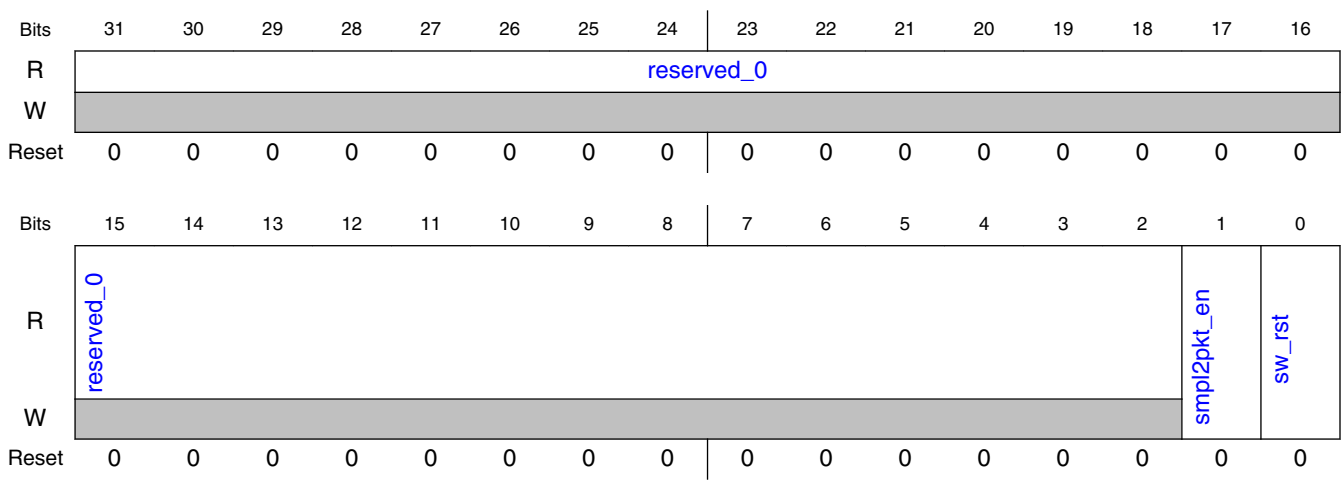
Field	Function
31-0 spdif_ch2_st_stts_bits191160	SPDIF Channel 2 Status bits[191:160]

13.4.10.1.332 Sample 2 Packets Control Register (SMPL2PKT_CNTL)

13.4.10.1.332.1 Offset

Register	Offset
SMPL2PKT_CNTL	3_0080h

13.4.10.1.332.2 Diagram



13.4.10.1.332.3 Fields

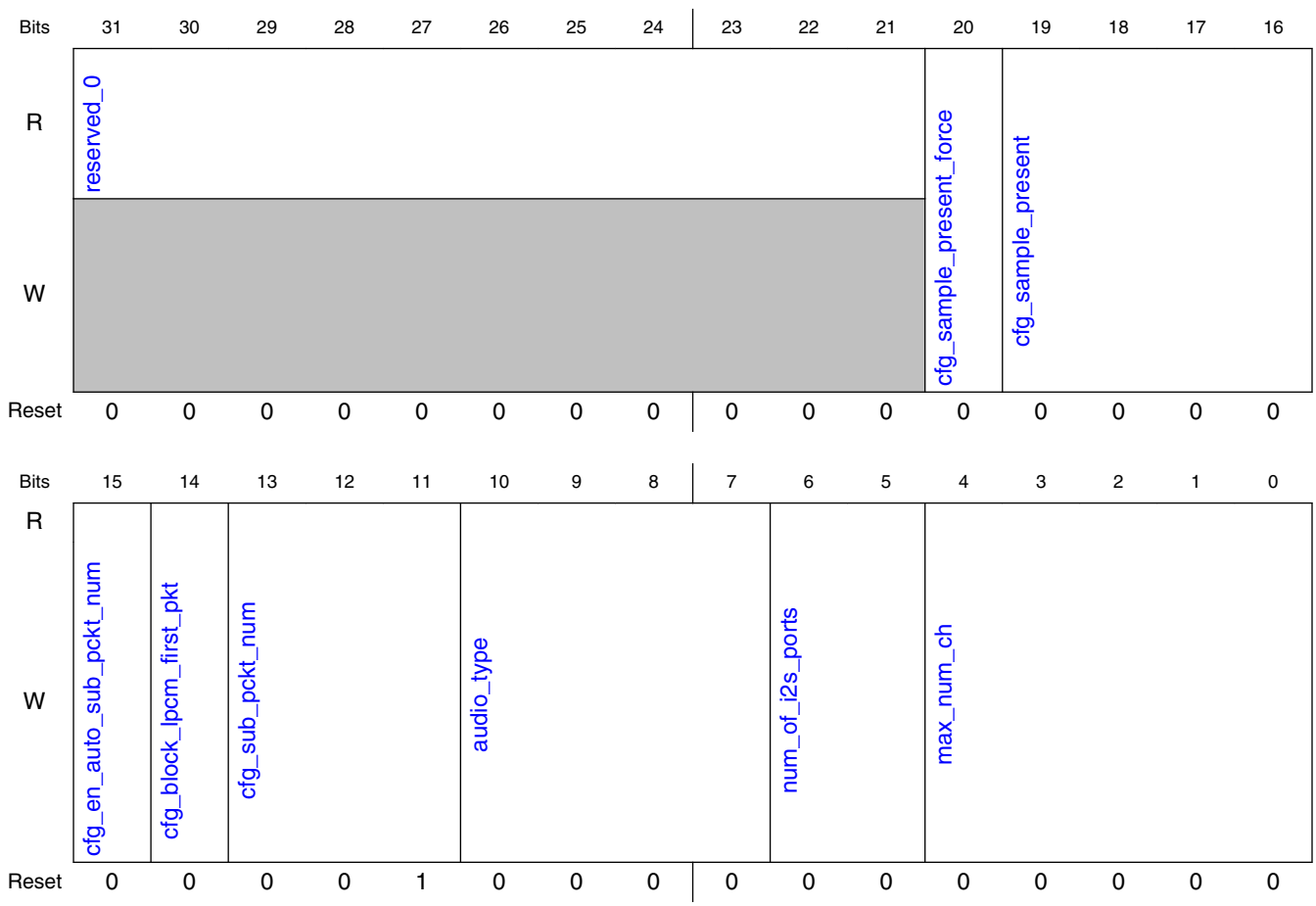
Field	Function
31-2 reserved_0	
1 smpl2pkt_en	When high Sample to Packets Block starts. When high Sample to Packets Block starts.
0 sw_rst	Software reset. Software reset. Active high.

13.4.10.1.333 Sample 2 Packets Config Register (SMPL2PKT_CNFG)

13.4.10.1.333.1 Offset

Register	Offset
SMPL2PKT_CNFG	3_0084h

13.4.10.1.333.2 Diagram



13.4.10.1.333.3 Fields

Field	Function
31-21 reserved_0	

Table continues on the next page...

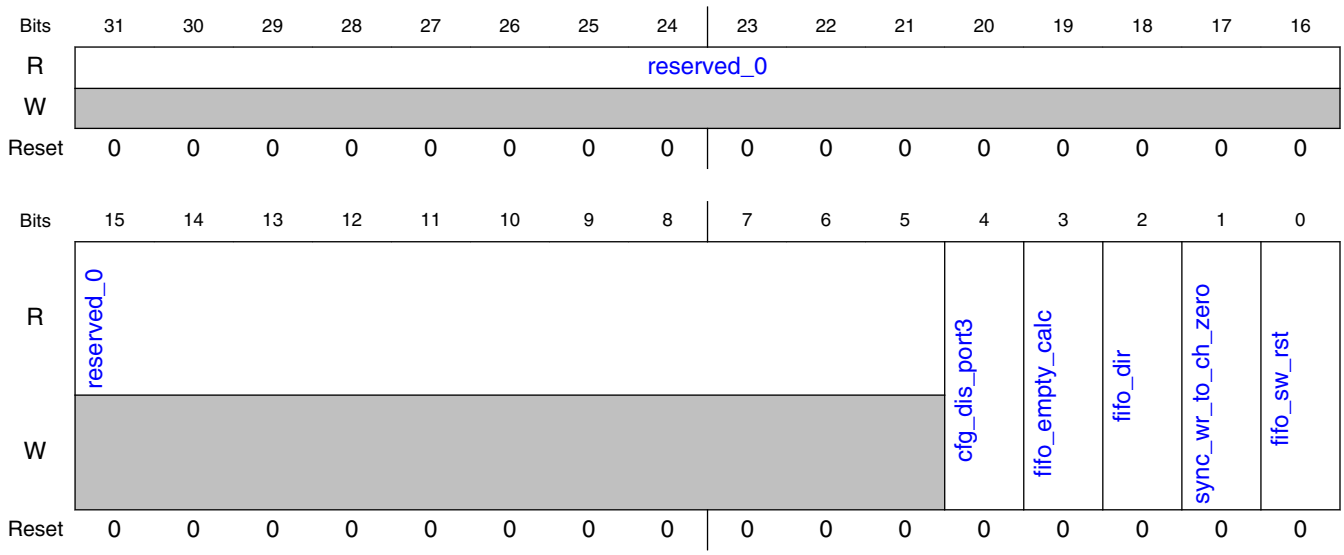
Field	Function
20 cfg_sample_present_force	Force sample present bits
19-16 cfg_sample_present	Sample present bits if force them is active
15 cfg_en_auto_sub_pckt_num	Enable automatics sub packet number. Enable automatics sub packet number. When enabled number of sub-packts will be set according to MEM FIFO number of samples.
14 cfg_block_lpcm_first_pkt	0 - All packets behave the same. 0 - All packets behave the same. 1- First lpcm audio packet is sent with 1 - SP.
13-11 cfg_sub_pckt_num	Number of sub-packets in HDMI audio 2-ch packet. Number of sub-packets in HDMI audio 2-ch packet. 00: 1-SP, 01: 2-SP, 10: 3-SP, 11: 4-SP.100-111: NA.
10-7 audio_type	Audio Type setting. Audio Type setting. Packet is structured according to audio type.
6-5 num_of_i2s_ports	Number ofactive I2S ports. Number ofactive I2S ports. 00- 1 port, 01-2 ports, 11- 4 ports, 11 -NA.
4-0 max_num_ch	Number of channels to decode. Number of channels to decode. 0: 1 channel, 31: 32 channels

13.4.10.1.334 FIFO control register (FIFO_CNTL)

13.4.10.1.334.1 Offset

Register	Offset
FIFO_CNTL	3_0088h

13.4.10.1.334.2 Diagram



13.4.10.1.334.3 Fields

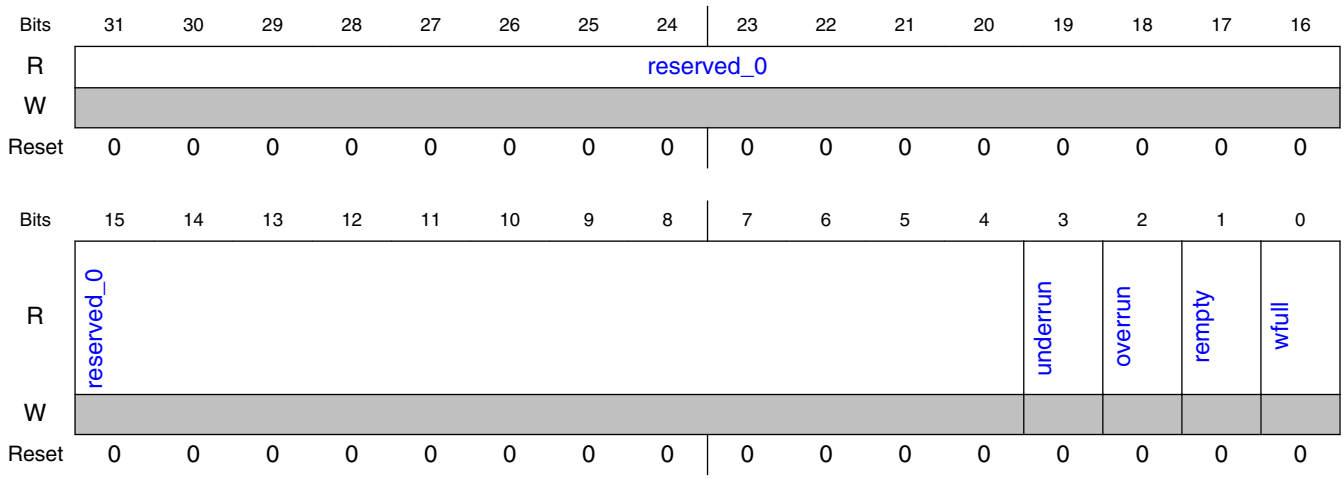
Field	Function
31-5 reserved_0	
4 cfg_dis_port3	0 - Normal Operation. 0 - Normal Operation. 1 - I2S port 3 is disabled (user should ignore its outputs). This allows for 24-ch, 12-ch, 6-ch transfer.
3 fifo_empty_calc	0- Empty is a function of read address. 0- Empty is a function of read address. 1 - Empty is a function of BASE read address.
2 fifo_dir	0 - smpl2pkt (inc_step=number of I2S ports), 1 - pkt2smpl (inc_step=num_ch_per_port)
1 sync_wr_to_ch_zero	When high the last channel index synchronizes the write addresses (to the next channel group)
0 fifo_sw_rst	Resets Fifo's write and read pointers. Resets Fifo's write and read pointers. When FIFO configuration bits change this signal should be high (due to synchronization issues).

13.4.10.1.335 FIFO Status register (FIFO_STTS)

13.4.10.1.335.1 Offset

Register	Offset
FIFO_STTS	3_008Ch

13.4.10.1.335.2 Diagram



13.4.10.1.335.3 Fields

Field	Function
31-4 reserved_0	
3 underrun	Indicates a FIFO underrun error has occured - FIFO read when it was empty. Indicates a FIFO underrun error has occured - FIFO read when it was empty. For debug purposes, not synchronized.
2 overrun	Indicates a FIFO overrun error has occured - FIFO written to when it was full. Indicates a FIFO overrun error has occured - FIFO written to when it was full. For debug purposes, not synchronized.
1 rempty	Indicates FIFO Empty. Indicates FIFO Empty. For debug purposes, not synchronized.
0 wfull	Indicates FIFO Full. Indicates FIFO Full. For debug purposes, not synchronized.

13.4.10.1.336 SUB Packet Threshold register (SUB_PCKT_THRSH)

13.4.10.1.336.1 Offset

Register	Offset
SUB_PCKT_THRSH	3_0090h

13.4.10.1.336.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0								cfg_mem_fifo_thrsh3							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cfg_mem_fifo_thrsh2								cfg_mem_fifo_thrsh1							
W																
Reset	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0

13.4.10.1.336.3 Fields

Field	Function
31-24 reserved_0	
23-16 cfg_mem_fifo_thrsh3	If number of samples in MEM FIFO is below Threshold 3: Each Packet will contain only 3 subpacket. If number of samples in MEM FIFO is below Threshold 3: Each Packet will contain only 3 subpacket.
15-8 cfg_mem_fifo_thrsh2	If number of samples in MEM FIFO is below Threshold2: Each Packet will contain only 2 subpacket. If number of samples in MEM FIFO is below Threshold2: Each Packet will contain only 2 subpacket.
7-0 cfg_mem_fifo_thrsh1	If number of samples in MEM FIFO is below Threshold 1: Each Packet will contain only 1 subpacket. If number of samples in MEM FIFO is below Threshold 1: Each Packet will contain only 1 subpacket.

13.4.10.1.337 4 MSB of the packet memory address in which the data is written. (SOURCE_PIF_WR_ADDR)

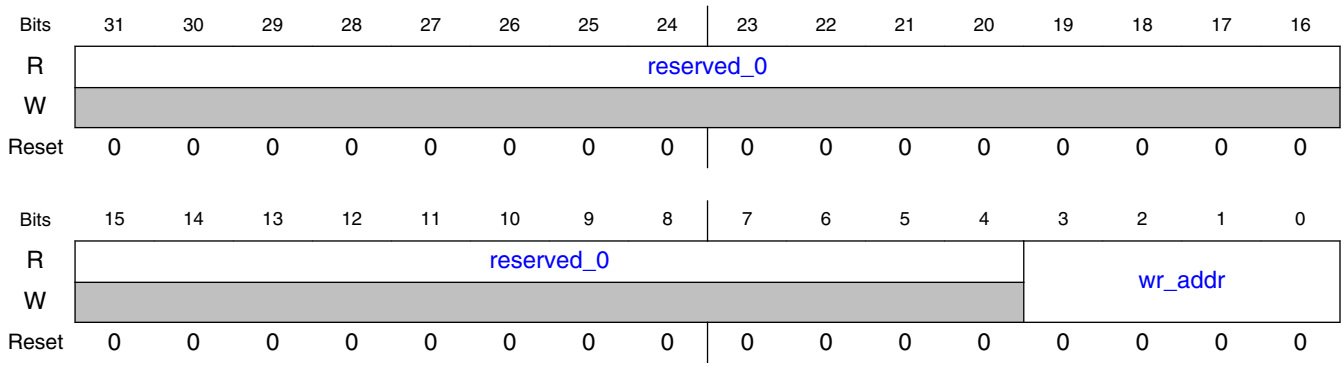
13.4.10.1.337.1 Offset

Register	Offset
SOURCE_PIF_WR_ADDR	3_0800h

13.4.10.1.337.2 Function

4 MSB of the packet memory address in which the data is written.

13.4.10.1.337.3 Diagram



13.4.10.1.337.4 Fields

Field	Function
31-4 reserved_0	
3-0 wr_addr	4 MSB of the packet memory address in which the data is written. 4 MSB of the packet memory address in which the data is written.

13.4.10.1.338 Write request bit for the host write transaction. (SOURCE_PIF_WR_REQ)

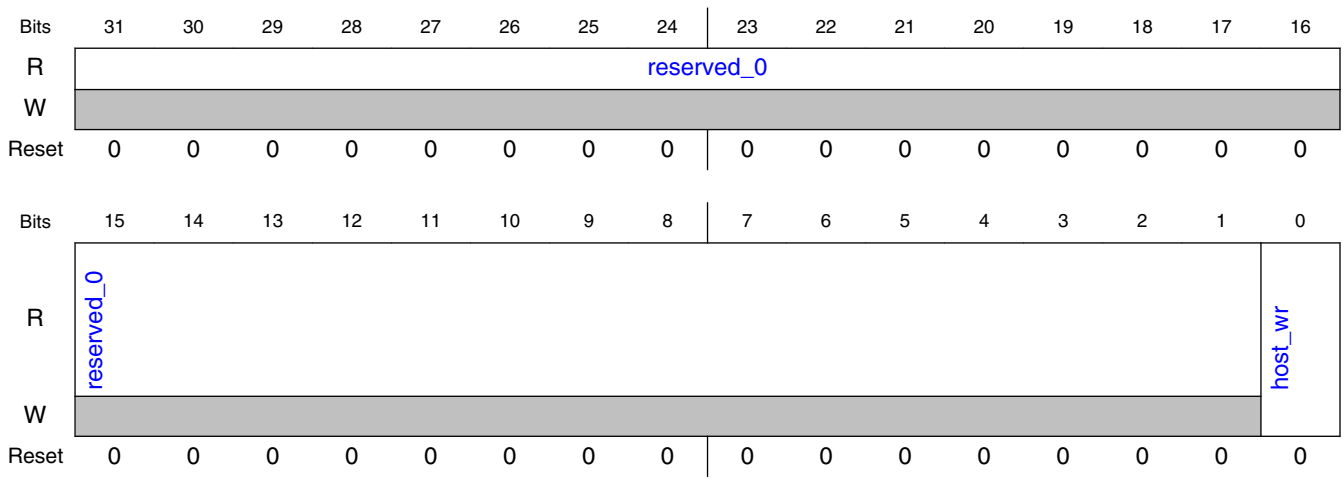
13.4.10.1.338.1 Offset

Register	Offset
SOURCE_PIF_WR_REQ	3_0804h

13.4.10.1.338.2 Function

Write request bit for the host write transaction.

13.4.10.1.338.3 Diagram



13.4.10.1.338.4 Fields

Field	Function
31-1 reserved_0	
0 host_wr	Write request bit for the host write transaction, active high. Write request bit for the host write transaction, active high. Bit is automaticaly cleared when operation is completed.

13.4.10.1.339 4 MSB of the packet memory address from which the data is read. (SOURCE_PIF_RD_ADDR)

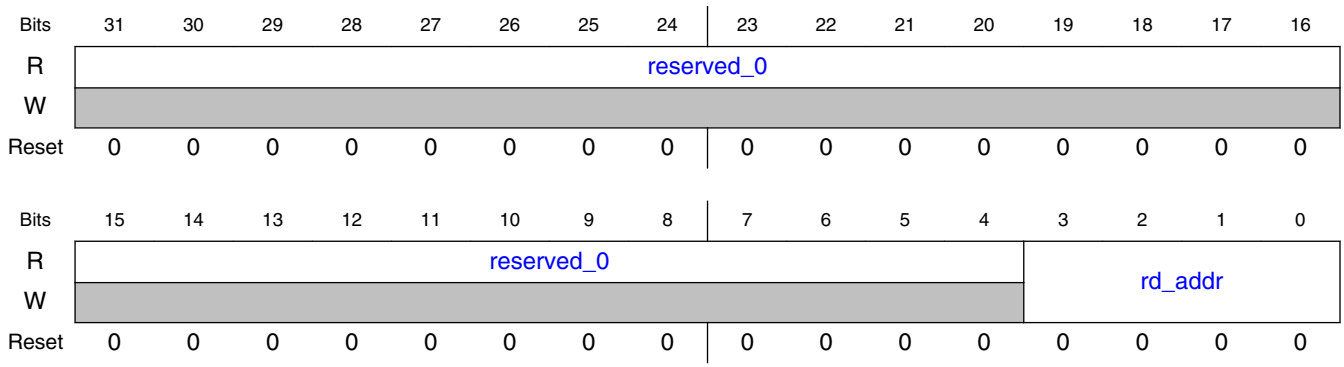
13.4.10.1.339.1 Offset

Register	Offset
SOURCE_PIF_RD_ADDR	3_0808h

13.4.10.1.339.2 Function

4 MSB of the packet memory address from which the data is read.

13.4.10.1.339.3 Diagram



13.4.10.1.339.4 Fields

Field	Function
31-4 reserved_0	
3-0 rd_addr	4 MSB of the packet memory address from which the data is read. 4 MSB of the packet memory address from which the data is read.

13.4.10.1.340 Read request bit for the host read transaction. (SOURCE_PIF_RD_REQ)

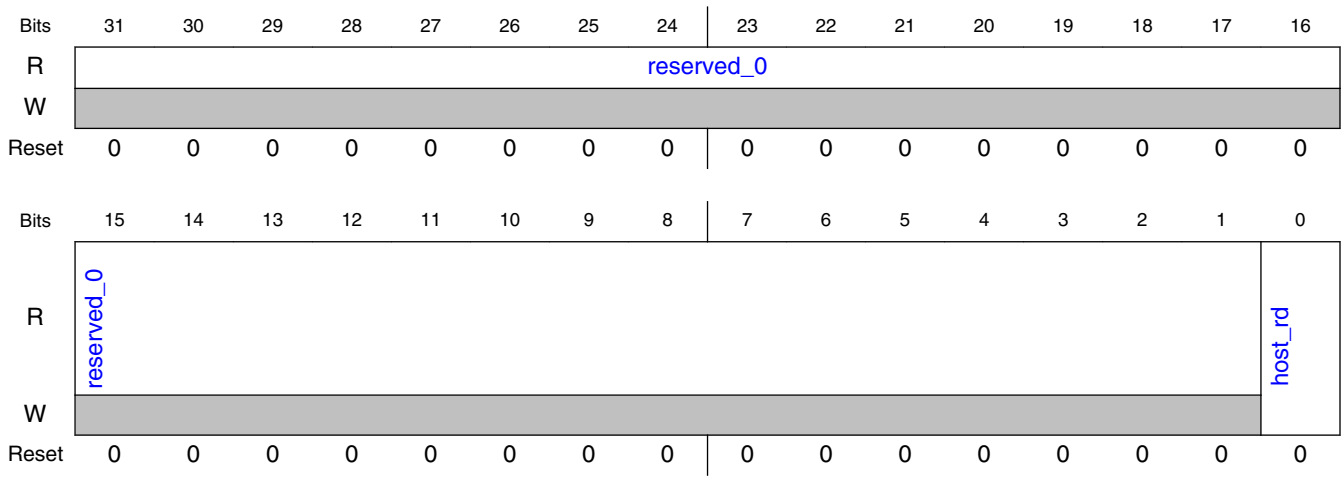
13.4.10.1.340.1 Offset

Register	Offset
SOURCE_PIF_RD_REQ	3_080Ch

13.4.10.1.340.2 Function

Read request bit for the host read transaction.

13.4.10.1.340.3 Diagram



13.4.10.1.340.4 Fields

Field	Function
31-1 reserved_0	
0 host_rd	Read request bit for the host read transaction, active high. Read request bit for the host read transaction, active high. Bit is automaticaly cleared when operation is completed.

13.4.10.1.341 The 32 bits of the data to be written to the packet memory. (SOURCE_PIF_DATA_WR)

13.4.10.1.341.1 Offset

Register	Offset
SOURCE_PIF_DATA_WR	3_0810h

13.4.10.1.341.2 Function

The 32 bits of the data to be written to the packet memory.

13.4.10.1.341.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	data_wr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	data_wr															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.341.4 Fields

Field	Function
31-0 data_wr	The 32 bits of the data to be written to the packet memory. The 32 bits of the data to be written to the packet memory. When written to this register fifo1_wr_enable will automatically be asserted and the data is stored in FIFO.

13.4.10.1.342 The 32 bits of the data to be read from the packet memory. (SOURCE_PIF_DATA_RD)

13.4.10.1.342.1 Offset

Register	Offset
SOURCE_PIF_DATA_RD	3_0814h

13.4.10.1.342.2 Function

The 32 bits of the data to be read from the packet memory.

13.4.10.1.342.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	fifo2_data_out															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	fifo2_data_out															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.342.4 Fields

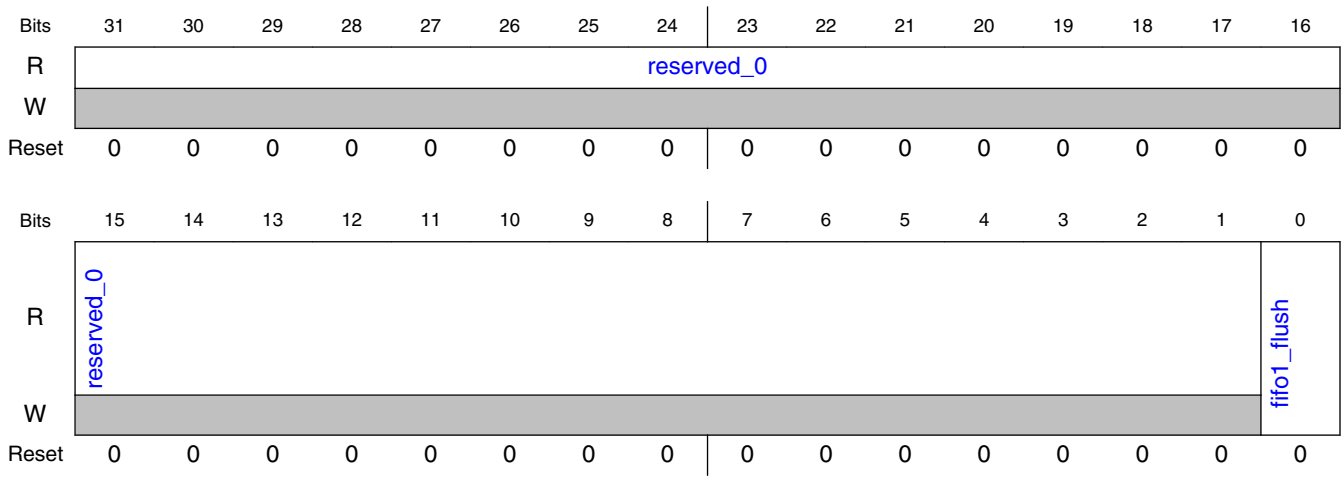
Field	Function
31-0 fifo2_data_out	The 32 bits of the data to be read from the packet memory. The 32 bits of the data to be read from the packet memory. When read from this register fifo2_rd_enable will automatically be asserted and the data is read from the FIFO.

13.4.10.1.343 Fifo1 flush (SOURCE_PIF_FIFO1_FLUSH)

13.4.10.1.343.1 Offset

Register	Offset
SOURCE_PIF_FIFO1_FLUSH	3_0818h

13.4.10.1.343.2 Diagram



13.4.10.1.343.3 Fields

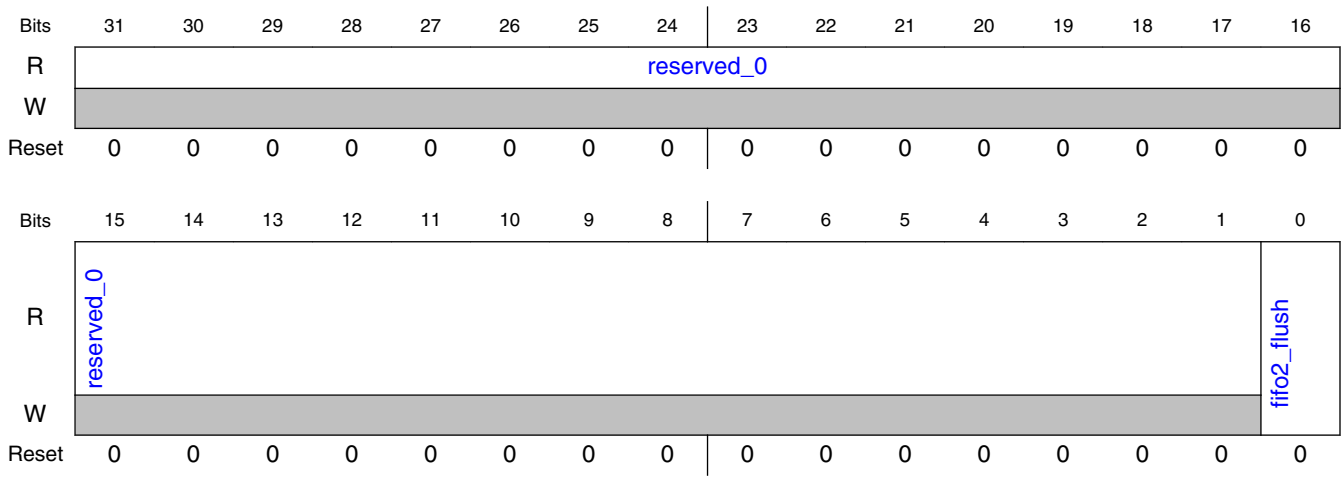
Field	Function
31-1 reserved_0	
0 fifo1_flush	Fifo1 flush bit, active high. Fifo1 flush bit, active high. Bit is automaticaly cleared when operation is completed.

13.4.10.1.344 Fifo2 flush (SOURCE_PIF_FIFO2_FLUSH)

13.4.10.1.344.1 Offset

Register	Offset
SOURCE_PIF_FIFO2_FLUSH	3_081Ch

13.4.10.1.344.2 Diagram



13.4.10.1.344.3 Fields

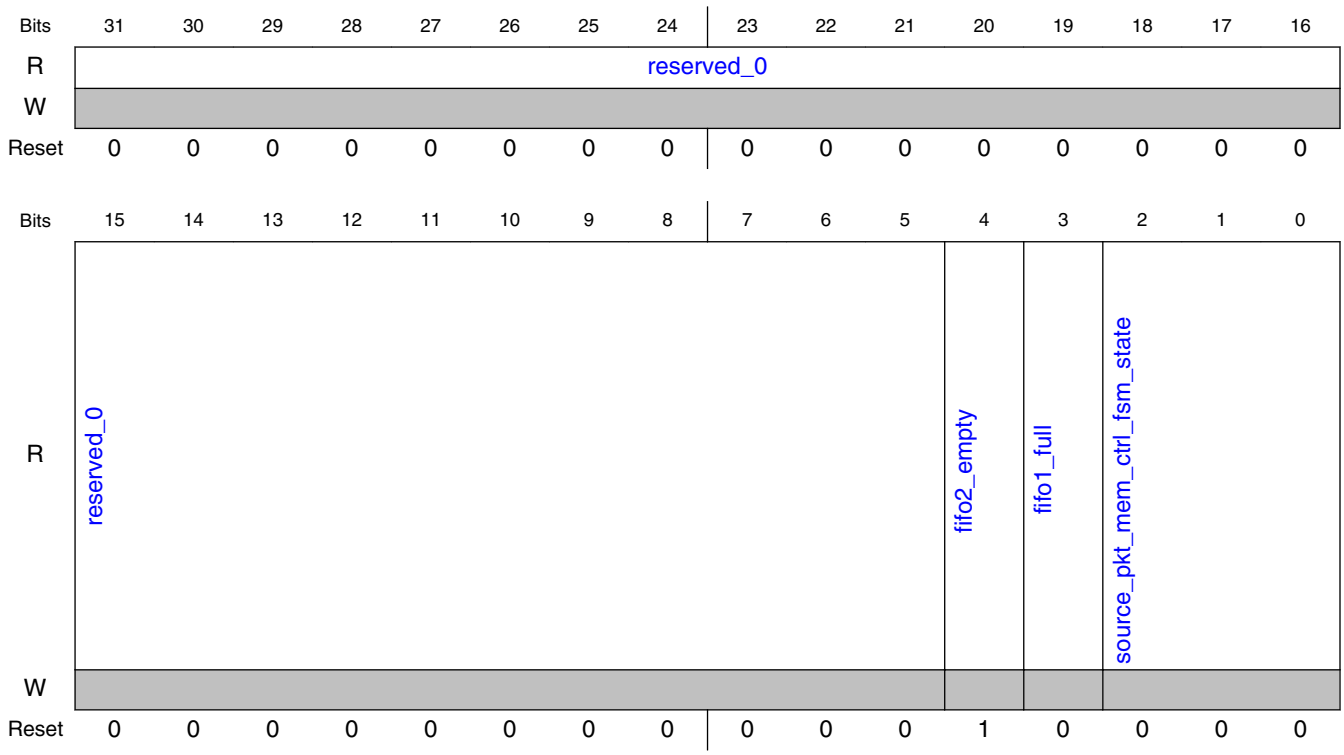
Field	Function
31-1 reserved_0	
0 fifo2_flush	Fifo2 flush bit, active high. Fifo2 flush bit, active high. Bit is automaticaly cleared when operation is completed.

13.4.10.1.345 Status bits for the PIF module (SOURCE_PIF_STATUS)

13.4.10.1.345.1 Offset

Register	Offset
SOURCE_PIF_STATUS	3_0820h

13.4.10.1.345.2 Diagram



13.4.10.1.345.3 Fields

Field	Function
31-5 reserved_0	
4 fifo2_empty	Fifo2 empty indication, when high indicates that FIFO2 is empty
3 fifo1_full	Fifo1 full indication, when high indicates that FIFO1 is full
2-0 source_pkt_mem_ctrl_fsm_state	State of the FSM that controls packet memory transactions. State of the FSM that controls packet memory transactions.

13.4.10.1.346 Interrupt sources of the PIF module, active high. (SOURCE_PIF_INTERRUPT_SOURCE)

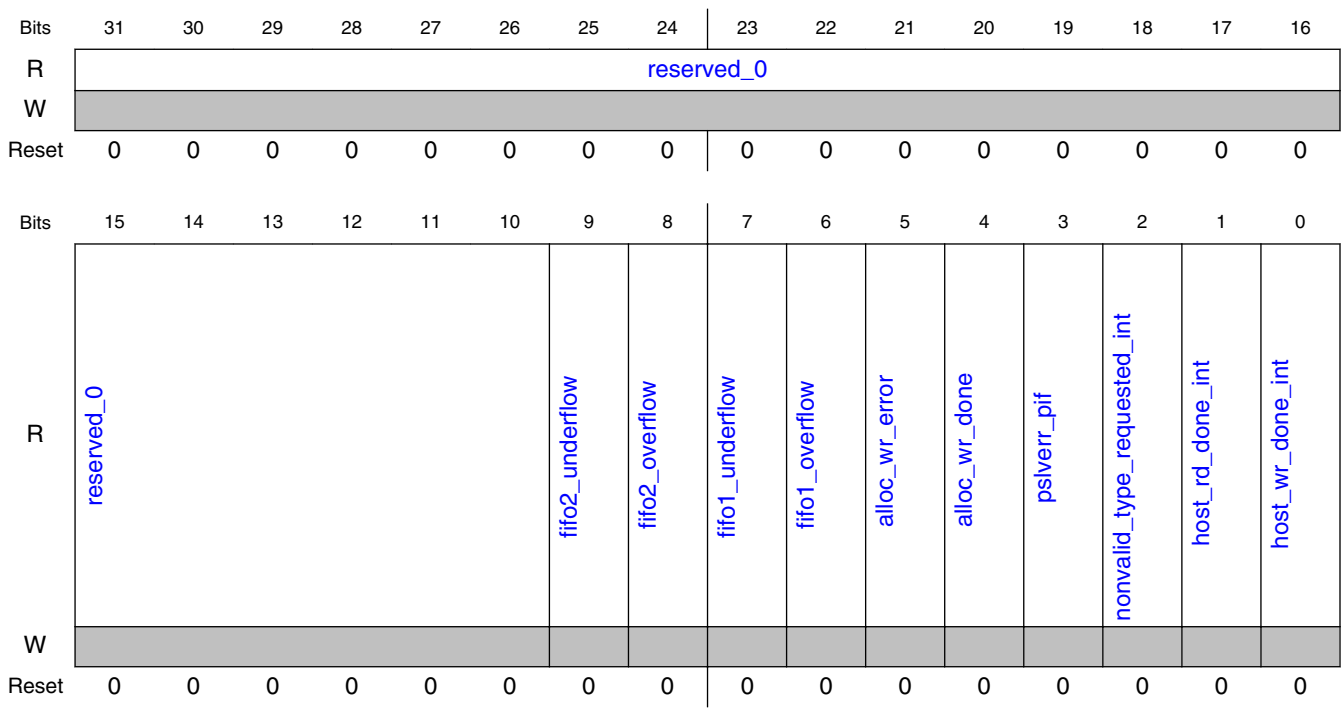
13.4.10.1.346.1 Offset

Register	Offset
SOURCE_PIF_INTERRUPT_SOURCE	3_0824h

13.4.10.1.346.2 Function

Interrupt sources of the PIF module, active high. Automatically cleared on read.

13.4.10.1.346.3 Diagram



13.4.10.1.346.4 Fields

Field	Function
31-10 reserved_0	
9 fifo2_underflow	Fifo2 underflow indication
8 fifo2_overflow	Fifo2 overflow indication

Table continues on the next page...

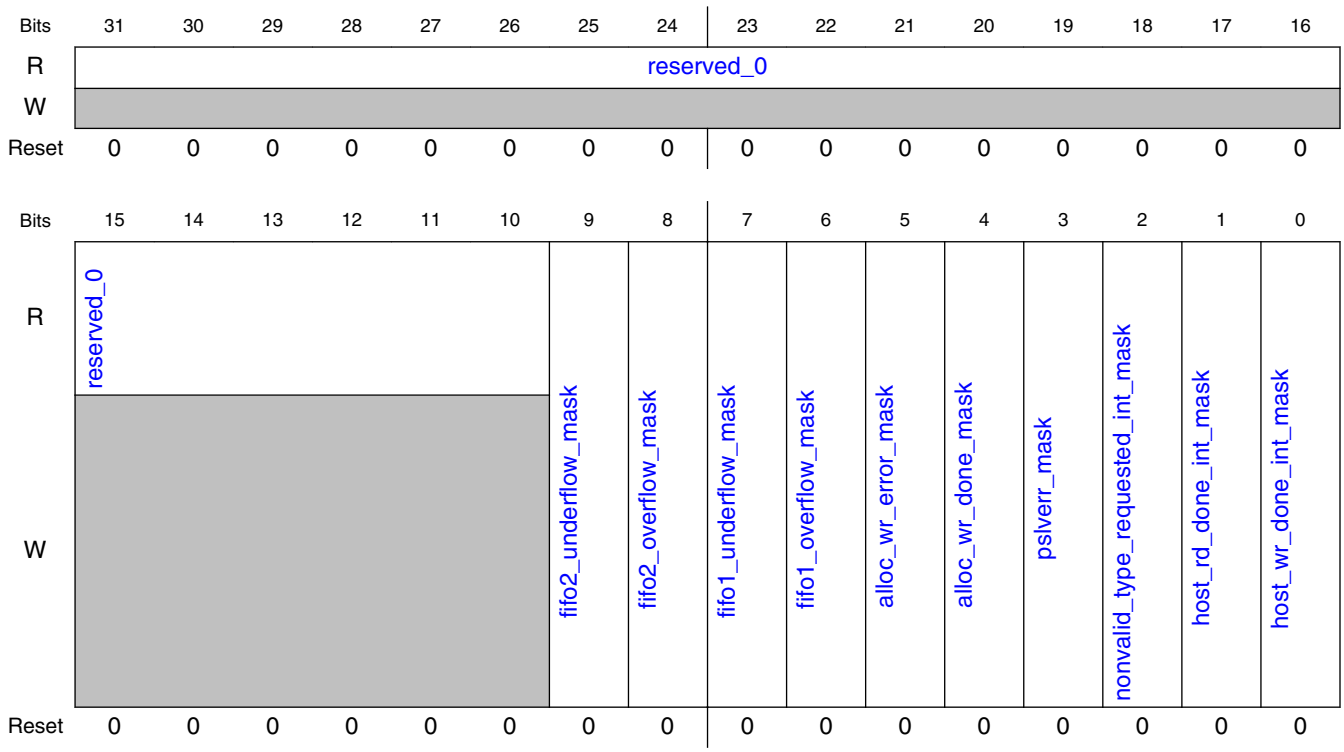
Field	Function
7 fifo1_underflow	Fifo1 underflow indication
6 fifo1_overflow	Fifo1 overflow indication
5 alloc_wr_error	Error happened, invalid write to the allocation table. Error happened, invalid write to the allocation table.
4 alloc_wr_done	Successful write to the allocation table. Successful write to the allocation table.
3 pslverr_pif	APB slave error interrupt from Packet Interface
2 nonvalid_type_requested_int	Indication that nonvalid type of packet is requested by the packet interface. Indication that nonvalid type of packet is requested by the packet interface.
1 host_rd_done_int	Indication that the host read transaction finished. Indication that the host read transaction finished.
0 host_wr_done_int	Indication that the host write transaction finished. Indication that the host write transaction finished.

13.4.10.1.347 Masks for the interrupt sources in the SOURCE_PIF_INTERRUPT_SOURCE register, when set high, these bits disable the corresponding interrupts (SOURCE_PIF_INTERRUPT_MASK)

13.4.10.1.347.1 Offset

Register	Offset
SOURCE_PIF_INTERRUPT_MASK	3_0828h

13.4.10.1.347.2 Diagram



13.4.10.1.347.3 Fields

Field	Function
31-10 reserved_0	
9 fifo2_underflow_mask	Masks the fifo2_underflow interrupt
8 fifo2_overflow_mask	Masks the fifo2_overflow interrupt
7 fifo1_underflow_mask	Masks the fifo1_underflow interrupt
6 fifo1_overflow_mask	Masks the fifo1_overflow interrupt
5 alloc_wr_error_mask	Masks the alloc_wr_error interrupt
4	Masks the alloc_wr_done interrupt

Table continues on the next page...

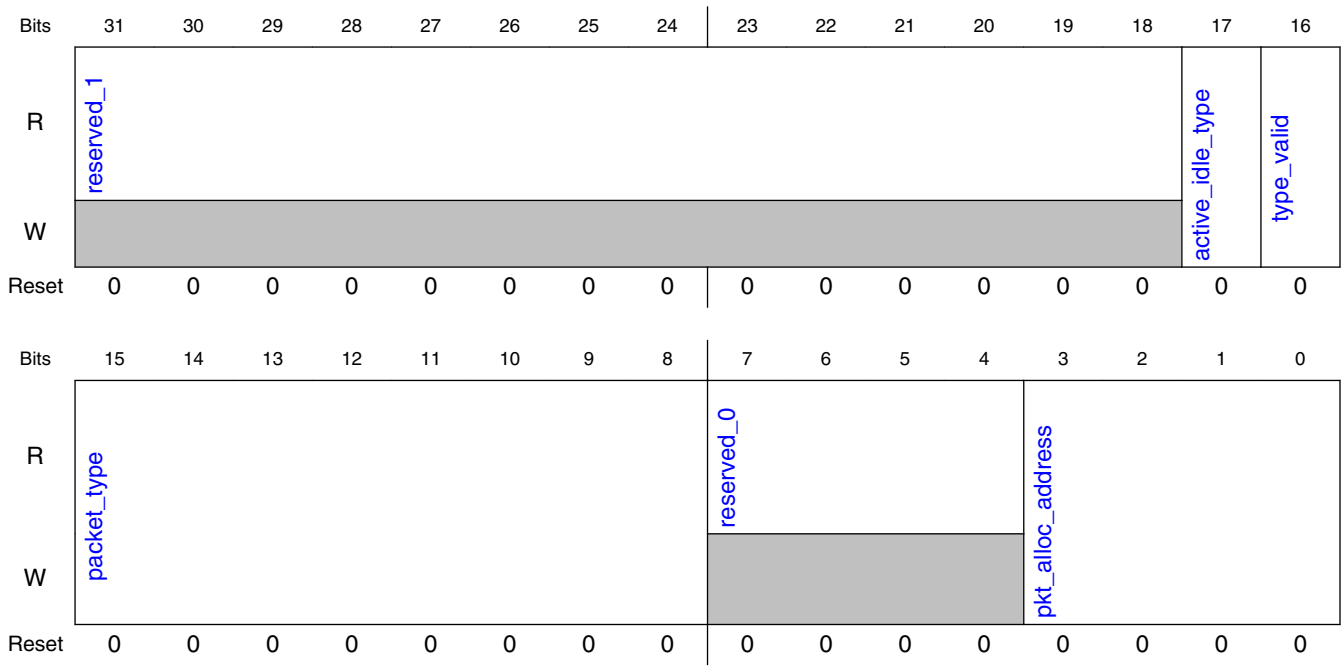
Field	Function
alloc_wr_done_mask	
3 pslverr_mask	Masks the pslverr_pif interrupt
2 nonvalid_type_requested_int_mask	Masks the nonvalid_type_requested_int interrupt
1 host_rd_done_int_mask	Masks the host_rd_done_int interrupt
0 host_wr_done_int_mask	Masks the host_wr_done_int interrupt

13.4.10.1.348 Packet configuration to be stored in the allocation table (SOURCE_PIF_PKT_ALLOC_REG)

13.4.10.1.348.1 Offset

Register	Offset
SOURCE_PIF_PKT_ALLOC_REG	3_082Ch

13.4.10.1.348.2 Diagram



13.4.10.1.348.3 Fields

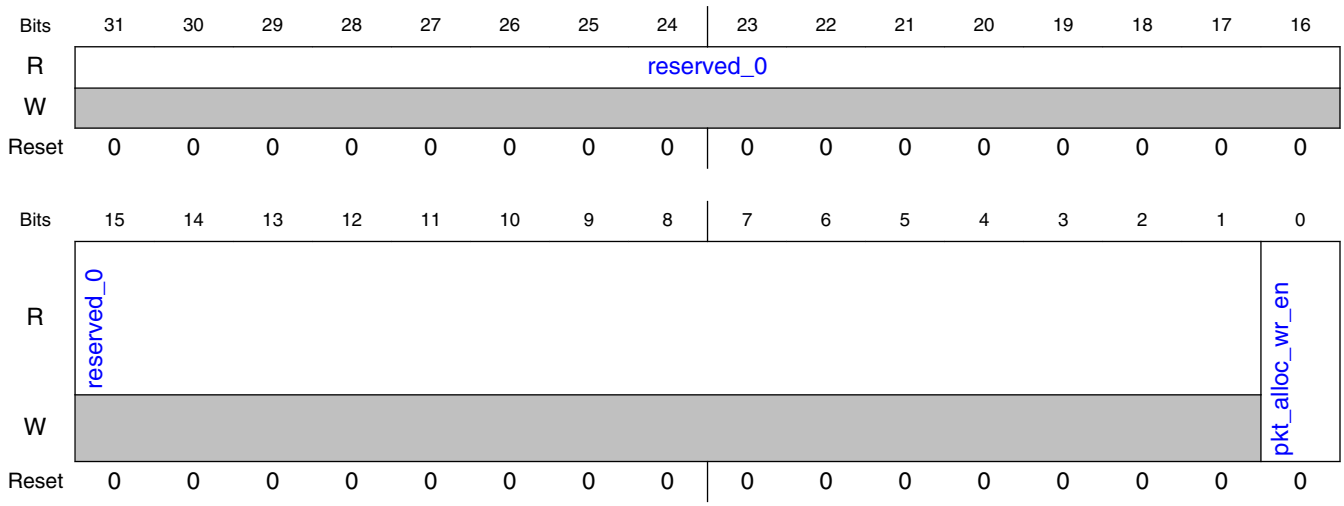
Field	Function
31-18 reserved_1	
17 active_idle_type	Indicates in which mode the SDP will be sent. Indicates in which mode the SDP will be sent. 0- no_video mode, 1- video mode
16 type_valid	1 for valid, 0 for nonvalid
15-8 packet_type	8-bit value of the packet type
7-4 reserved_0	
3-0 pkt_alloc_address	Address of the register in the source allocation table

13.4.10.1.349 Enable bit for writing to the allocation table (SOURCE_PIF_PKT_ALLOC_WR_EN)

13.4.10.1.349.1 Offset

Register	Offset
SOURCE_PIF_PKT_ALLOC_WR_EN	3_0830h

13.4.10.1.349.2 Diagram



13.4.10.1.349.3 Fields

Field	Function
31-1 reserved_0	
0 pkt_alloc_wr_en	Enable bit for writing to the allocation table, active high

13.4.10.1.350 Software reset. (SOURCE_PIF_SW_RESET)

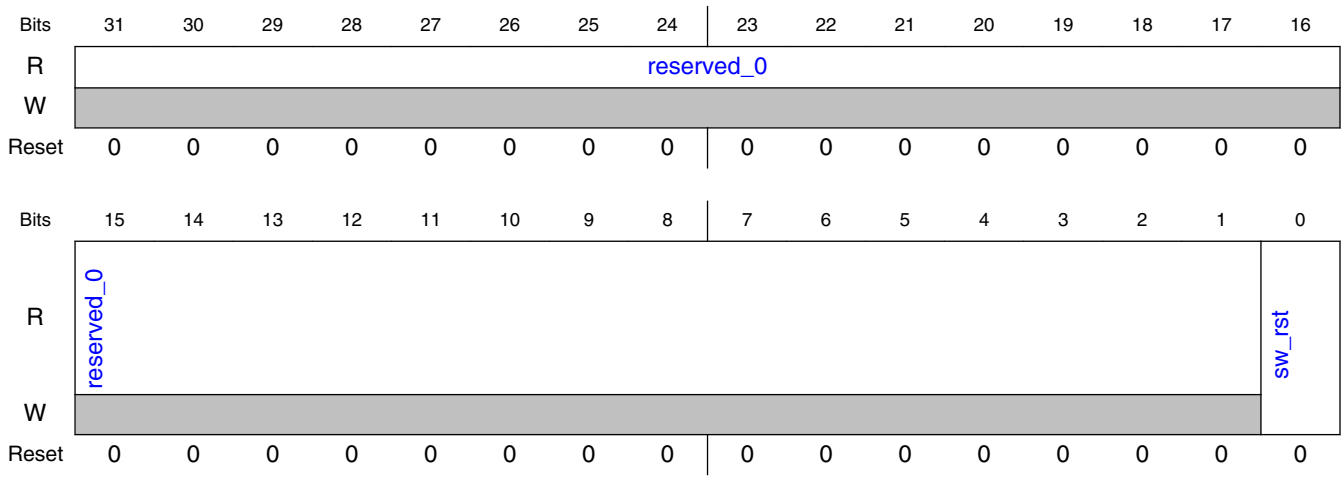
13.4.10.1.350.1 Offset

Register	Offset
SOURCE_PIF_SW_RESET	3_0834h

13.4.10.1.350.2 Function

Software reset.

13.4.10.1.350.3 Diagram



13.4.10.1.350.4 Fields

Field	Function
31-1 reserved_0	
0 sw_rst	Software reset, active high. Software reset, active high. Bit is automaticaly cleared when operation is completed.

13.4.10.1.351 Contains the revision information of the HDCP Cipher module (HDCP_REVISION)

13.4.10.1.351.1 Offset

Register	Offset
HDCP_REVISION	6_0000h

13.4.10.1.351.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reg_file_rev								hdcp_22_rev							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	hdcp_14_rev								top_hdcp_rev							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.351.3 Fields

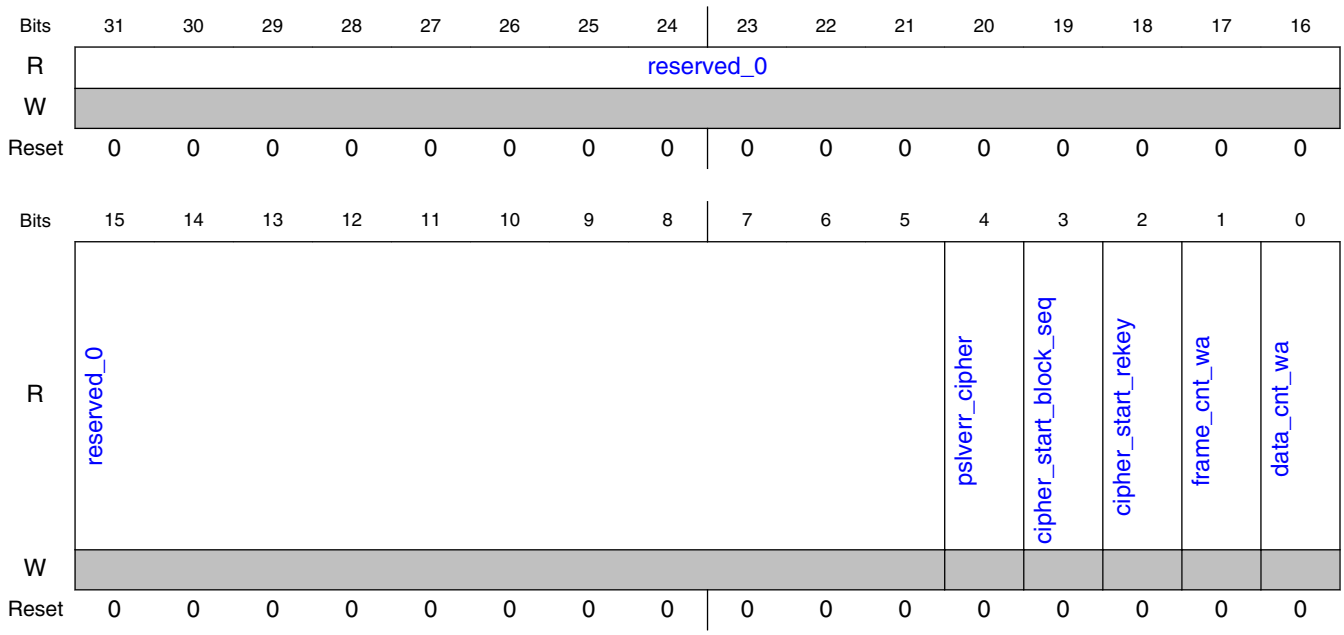
Field	Function
31-24 reg_file_rev	Revision number for the register file
23-16 hdcp_22_rev	Revision number for the HDCP 2. Revision number for the HDCP 2.2 engine
15-8 hdcp_14_rev	Revision number for the HDCP 1. Revision number for the HDCP 1.4 engine
7-0 top_hdcp_rev	Revision number for the top level RTL code

13.4.10.1.352 Contains the status of the HDCP Cipher interrupt sources (INTERRUPT_SOURCE)

13.4.10.1.352.1 Offset

Register	Offset
INTERRUPT_SOURCE	6_0004h

13.4.10.1.352.2 Diagram



13.4.10.1.352.3 Fields

Field	Function
31-5 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
4 pslverr_cipher	APB slave error from Cipher module. APB slave error from Cipher module. Generated when APB address is out of the range
3 cipher_start_block_seq	Block start event
2 cipher_start_rekey	Start rekey event
1 frame_cnt_wa	Frame counter wraparound event. Frame counter wraparound event. Active high
0 data_cnt_wa	Data counter wraparound event. Data counter wraparound event. Active high

13.4.10.1.353 Contains the mask vector of the HDCP Cipher interrupt sources. (INTERRUPT_MASK)

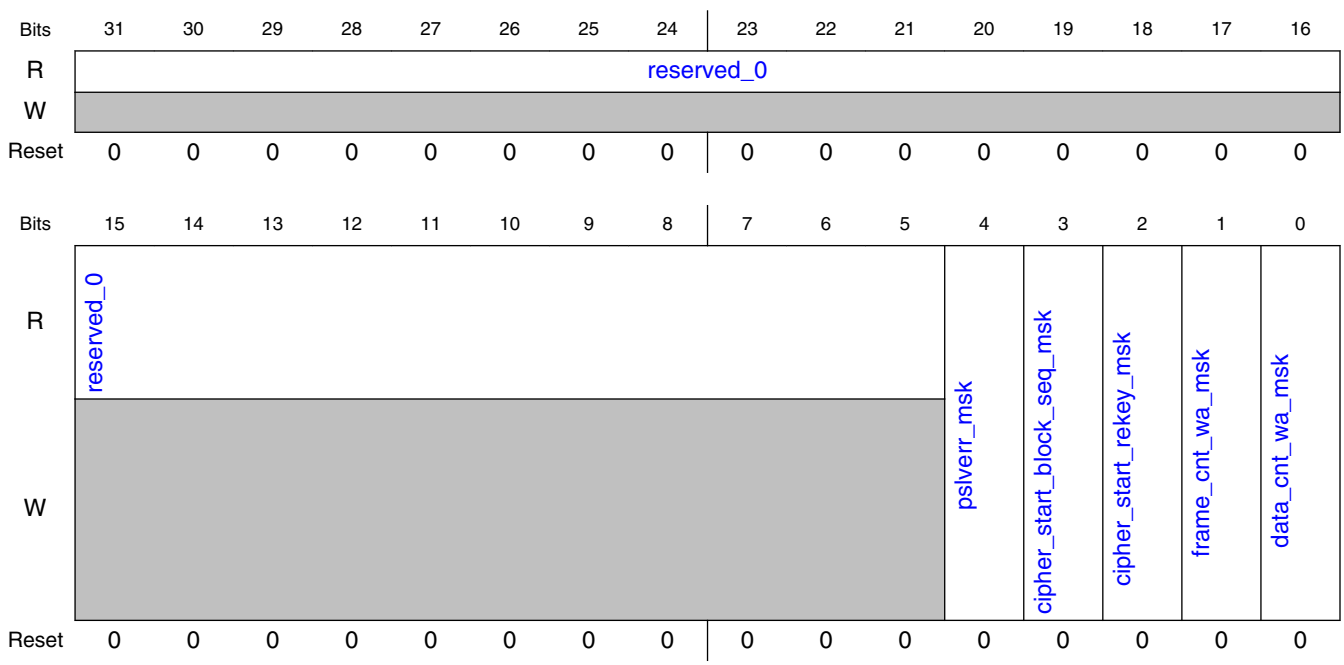
13.4.10.1.353.1 Offset

Register	Offset
INTERRUPT_MASK	6_0008h

13.4.10.1.353.2 Function

Contains the mask vector of the HDCP Cipher interrupt sources.

13.4.10.1.353.3 Diagram



13.4.10.1.353.4 Fields

Field	Function
31-5 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
4 pslverr_msk	Set to 1 to mask the pslverr_cipher interrupt
3 cipher_start_block_seq_msk	Set to 1 mask the BLOCK_START interrupt
2	Set to 1 mask the START_REKEY interrupt

Table continues on the next page...

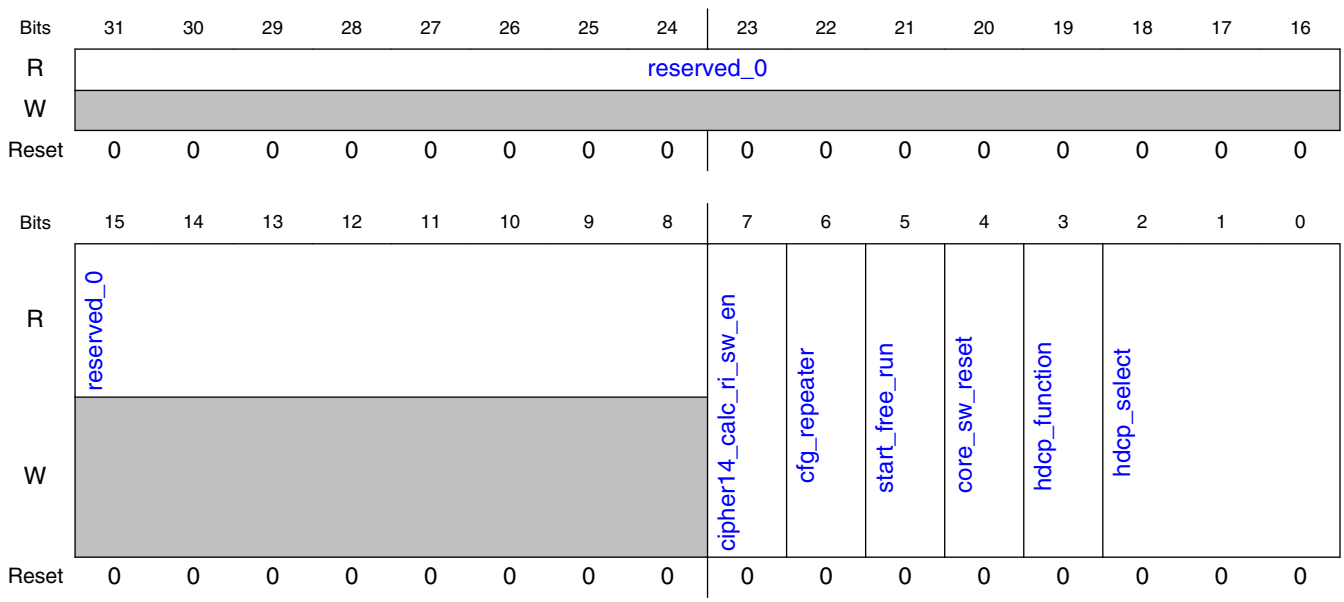
Field	Function
cipher_start_rekey_msk	
1 frame_cnt_wa_msk	Set to 1 mask the FRAME_COUNT_WA interrupt
0 data_cnt_wa_msk	Set to 1 mask the DATA_COUNT_WA interrupt

13.4.10.1.354 Contains the configuration details for the Cipher core module (HDCP_CIPHER_CONFIG)

13.4.10.1.354.1 Offset

Register	Offset
HDCP_CIPHER_CONFIG	6_000Ch

13.4.10.1.354.2 Diagram



13.4.10.1.354.3 Fields

Field	Function
31-8 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
7 cipher14_calc_ri_sw_en	For RX; calculate Ri
6 cfg_repeater	HDCP Repeater capability bit : 0 connection from receiver. HDCP Repeater capability bit : 0 connection from receiver. HDCP receiver is not an HDCP repeater,1 connection from repeater. HDCP receiver is an HDCP repeater
5 start_free_run	Cipher 1. Cipher 1.4 command to start free running enable for operation hdcpRngCipher
4 core_sw_reset	Software reset signal for the Cipher core. Software reset signal for the Cipher core. Active high
3 hdcp_function	Selects the HDCP function performed. Selects the HDCP function performed.- 0x0 - Transmitter- 0x1 - Receiver
2-0 hdcp_select	Selects the HDCP core version support. Selects the HDCP core version support.- 0x0 - HDCP 1.4- 0x1 - HDCP 2.2- Other: Reserved

13.4.10.1.355 Holds the least significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_0)

13.4.10.1.355.1 Offset

Register	Offset
AES_128_KEY_0	6_0010h

13.4.10.1.355.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_key_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_key_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.355.3 Fields

Field	Function
31-0 aes_128_key_0	Holds the [31:0] of the Ks^Lc128 128 bits results as defined in HDCP2. Holds the [31:0] of the Ks^Lc128 128 bits results as defined in HDCP2.2 Figure 3-3- input key word of the AES-128 module

13.4.10.1.356 Holds the second significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_1)

13.4.10.1.356.1 Offset

Register	Offset
AES_128_KEY_1	6_0014h

13.4.10.1.356.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_key_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_key_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.356.3 Fields

Field	Function
31-0 aes_128_key_1	Holds the [63:32] of the Ks^Lc128 128 bits results as defined in HDCP2. Holds the [63:32] of the Ks^Lc128 128 bits results as defined in HDCP2.2 Figure 3-3- input key word of the AES-128 module

13.4.10.1.357 Holds the third significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_2)

13.4.10.1.357.1 Offset

Register	Offset
AES_128_KEY_2	6_0018h

13.4.10.1.357.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_key_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_key_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.357.3 Fields

Field	Function
31-0 aes_128_key_2	Holds the [95:64] of the Ks^Lc128 128 bits results as defined in HDCP2. Holds the [95:64] of the Ks^Lc128 128 bits results as defined in HDCP2.2 Figure 3-3- input key word of the AES-128 module

13.4.10.1.358 Holds the most significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_KEY_3)

13.4.10.1.358.1 Offset

Register	Offset
AES_128_KEY_3	6_001Ch

13.4.10.1.358.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_key_3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_key_3															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.358.3 Fields

Field	Function
31-0 aes_128_key_3	Holds the [127:96] of the Ks^Lc128 128 bits results as defined in HDCP2. Holds the [127:96] of the Ks^Lc128 128 bits results as defined in HDCP2.2 Figure 3-3- input key word of the AES-128 module

13.4.10.1.359 Holds the least significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_RANDOM_0)

13.4.10.1.359.1 Offset

Register	Offset
AES_128_RANDOM_0	6_0020h

13.4.10.1.359.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_random_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_random_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.359.3 Fields

Field	Function
31-0 aes_128_random_0	Holds the RiV[31:0] as defined in HDCP2. Holds the RiV[31:0] as defined in HDCP2.2 Figure 3-3-input random number for the AES-128 module

13.4.10.1.360 Holds the most significant 32-bits word of the 128-bits input key word of the AES-128 module (AES_128_RANDOM_1)

13.4.10.1.360.1 Offset

Register	Offset
AES_128_RANDOM_1	6_0024h

13.4.10.1.360.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	aes_128_random_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	aes_128_random_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.360.3 Fields

Field	Function
31-0 aes_128_random_1	Holds the RiV[64:32] as defined in HDCP2. Holds the RiV[64:32] as defined in HDCP2.2 Figure 3-3-input random number for the AES-128 module

13.4.10.1.361 Holds the first word of the Km value (CIPHER14_KM_0)

13.4.10.1.361.1 Offset

Register	Offset
CIPHER14_KM_0	6_0028h

13.4.10.1.361.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	cipher14_km_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cipher14_km_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.361.3 Fields

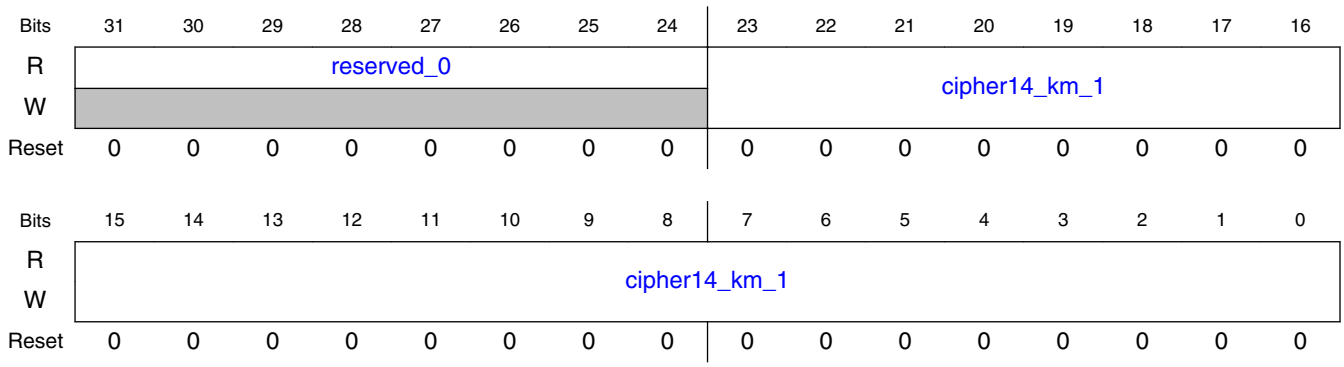
Field	Function
31-0 cipher14_km_0	Holds the first word of the Km value

13.4.10.1.362 Holds the most significant 3 bytes of the Km value (CIPHER14_KM_1)

13.4.10.1.362.1 Offset

Register	Offset
CIPHER14_KM_1	6_002Ch

13.4.10.1.362.2 Diagram



13.4.10.1.362.3 Fields

Field	Function
31-24 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
23-0 cipher14_km_1	Holds the most significant 3 bytes of the Km value

13.4.10.1.363 Cipher 1. (CIPHER14_STATUS)

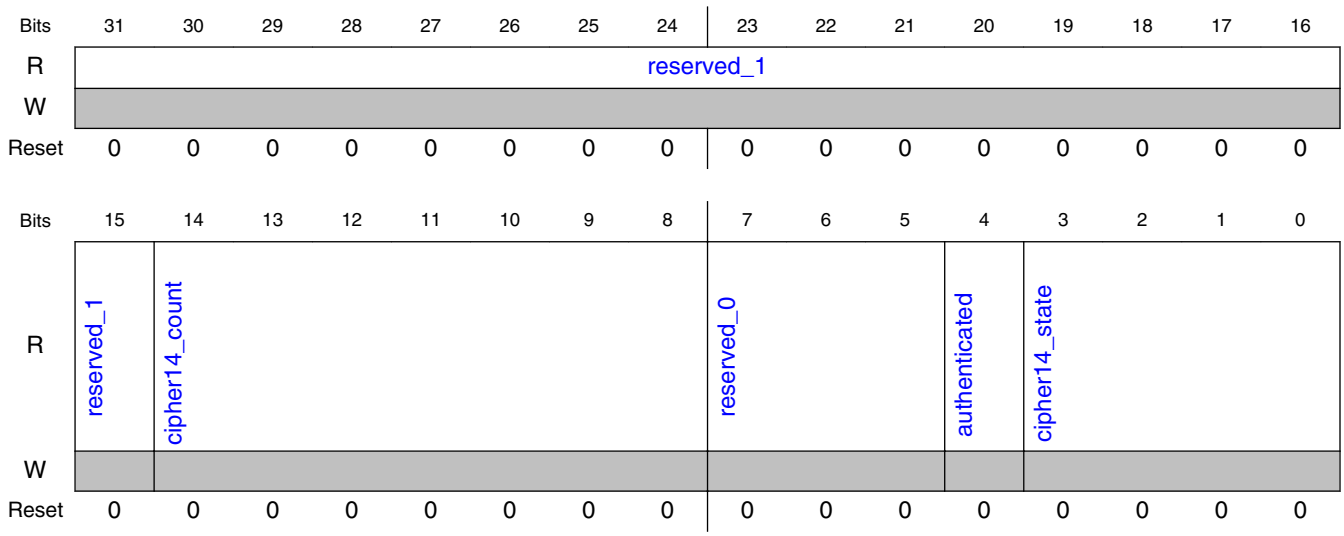
13.4.10.1.363.1 Offset

Register	Offset
CIPHER14_STATUS	6_0030h

13.4.10.1.363.2 Function

Cipher 1.4 status register

13.4.10.1.363.3 Diagram



13.4.10.1.363.4 Fields

Field	Function
31-15 reserved_1	Reserved Reserved. Writes are ignored. 0x0 when read
14-8 cipher14_count	Cipher 1. Cipher 1.4 counter output status.
7-5 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
4 authenticated	Authenticated status bit. Authenticated status bit.Set to 1 when first block starts generating R0
3-0 cipher14_state	Current state for Cipher FSM. Current state for Cipher FSM.Possible values:0001 - HDCPRNGCIPHER,0010 - HDCPBLOCKCIPHER, 0100 - HDCPSTREAMCIPHER state,1000 - HDCPREKEYCIPHER

13.4.10.1.364 Cipher 1. (CIPHER14_RI_PJ_STATUS)

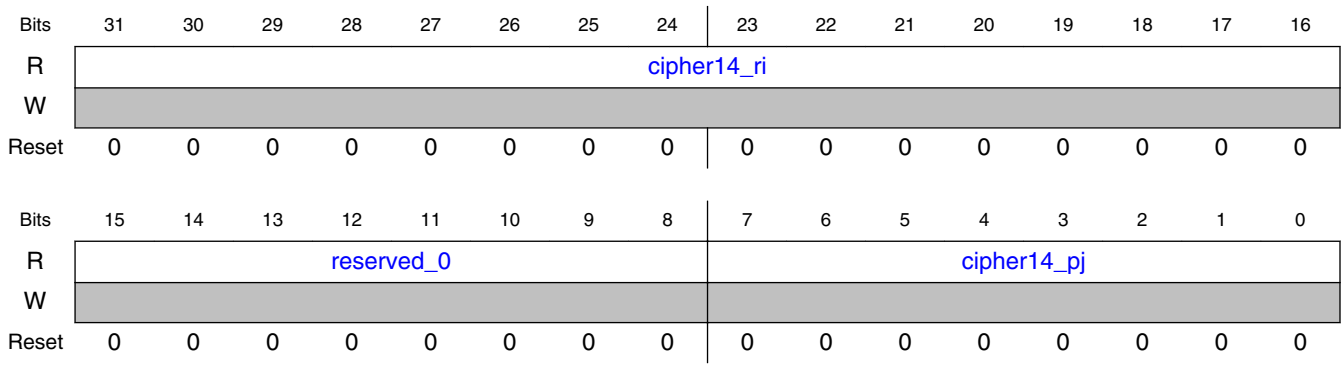
13.4.10.1.364.1 Offset

Register	Offset
CIPHER14_RI_PJ_STATUS	6_0034h

13.4.10.1.364.2 Function

Cipher 1.4 status register for Ri and Pj values

13.4.10.1.364.3 Diagram



13.4.10.1.364.4 Fields

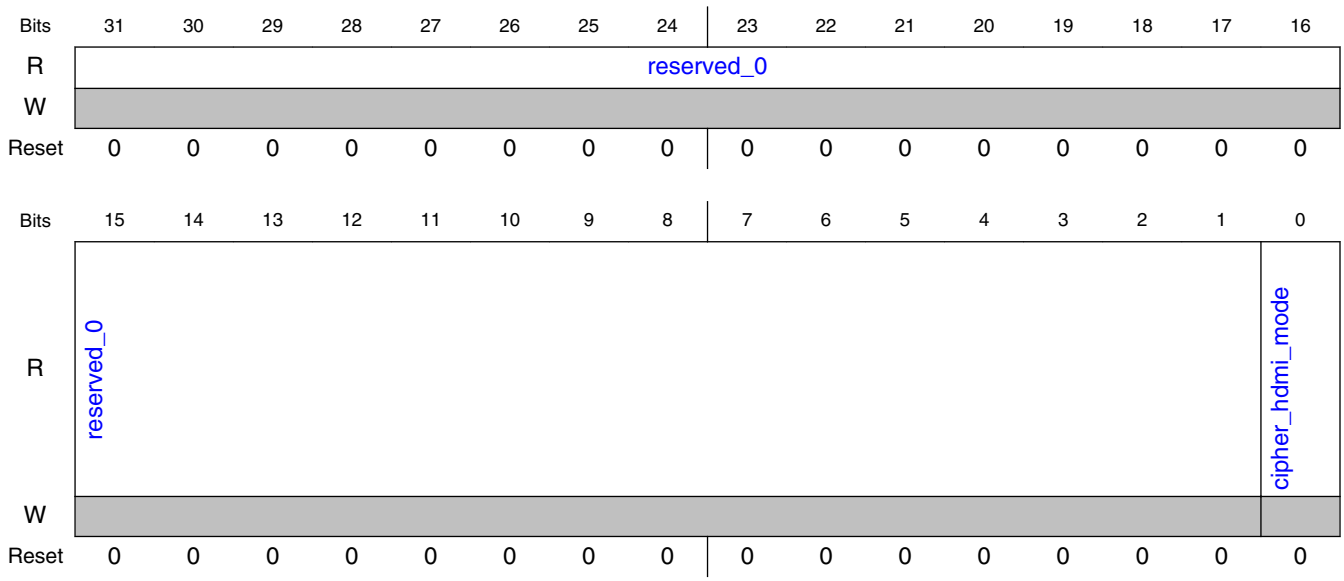
Field	Function
31-16 cipher14_ri	HDCP 1. HDCP 1.4 Ri data.
15-8 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
7-0 cipher14_pj	HDCP 1. HDCP 1.4 Pj data.

13.4.10.1.365 Cipher mode status register (CIPHER_MODE)

13.4.10.1.365.1 Offset

Register	Offset
CIPHER_MODE	6_0038h

13.4.10.1.365.2 Diagram



13.4.10.1.365.3 Fields

Field	Function
31-1 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
0 cipher_hdmi_mode	HDMI mode status (input cipher_hdmi_mode). HDMI mode status (input cipher_hdmi_mode).When 1 - HDMI -When 0 - DVI

13.4.10.1.366 Holds the First word of An value generated by hdcpRngCipher operation. (CIPHER14_AN_0)

13.4.10.1.366.1 Offset

Register	Offset
CIPHER14_AN_0	6_003Ch

13.4.10.1.366.2 Function

Holds the First word of An value generated by hdcpRngCipher operation.

13.4.10.1.366.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	cipher14_an_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cipher14_an_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.366.4 Fields

Field	Function
31-0 cipher14_an_0	Holds the First word of An value generated by hdcpRngCipher operation. Holds the First word of An value generated by hdcpRngCipher operation.

13.4.10.1.367 Holds the second word of An value generated by hdcpRngCipher operation (CIPHER14_AN_1)

13.4.10.1.367.1 Offset

Register	Offset
CIPHER14_AN_1	6_0040h

13.4.10.1.367.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	cipher14_an_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cipher14_an_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.367.3 Fields

Field	Function
31-0 cipher14_an_1	Holds the second word of An value generated by hdcpRngCipher operation

13.4.10.1.368 HDCP2. (CIPHER22_AUTH)

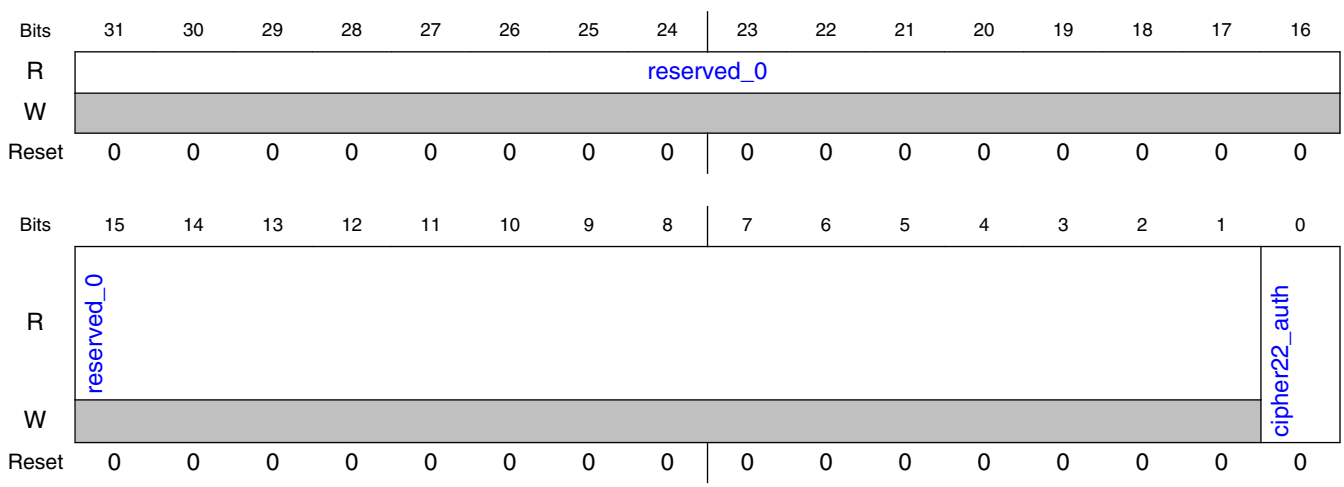
13.4.10.1.368.1 Offset

Register	Offset
CIPHER22_AUTH	6_0044h

13.4.10.1.368.2 Function

HDCP2.2 authentication status register. Set to 1 by firmware when authentication is finished successfuly. Cleared by FW when Sink is not authenticated (start of operation, Sink disconnection, authentication restart).

13.4.10.1.368.3 Diagram



13.4.10.1.368.4 Fields

Field	Function
31-1	Reserved

Table continues on the next page...

Field	Function
reserved_0	Reserved. Writes are ignored. 0x0 when read
0 cipher22_auth	Cipher 2. Cipher 2.2 authentication status. 1 - authenticated, 0 - not authenticated

13.4.10.1.369 This register contains Ri value in A5: Link Integrity Check state [HDCP 1. (CIPHER14_R0_DP_STATUS)

13.4.10.1.369.1 Offset

Register	Offset
CIPHER14_R0_DP_STATUS	6_0048h

13.4.10.1.369.2 Function

This register contains Ri value in A5: Link Integrity Check state [HDCP 1.4]

13.4.10.1.369.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	reserved_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cipher14_r0_dp															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.1.369.4 Fields

Field	Function
31-16 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
15-0 cipher14_r0_dp	Ri value checked in A5 state

13.4.10.1.370 Cipher for HDCP1. (CIPHER14_BOOTSTRAP)

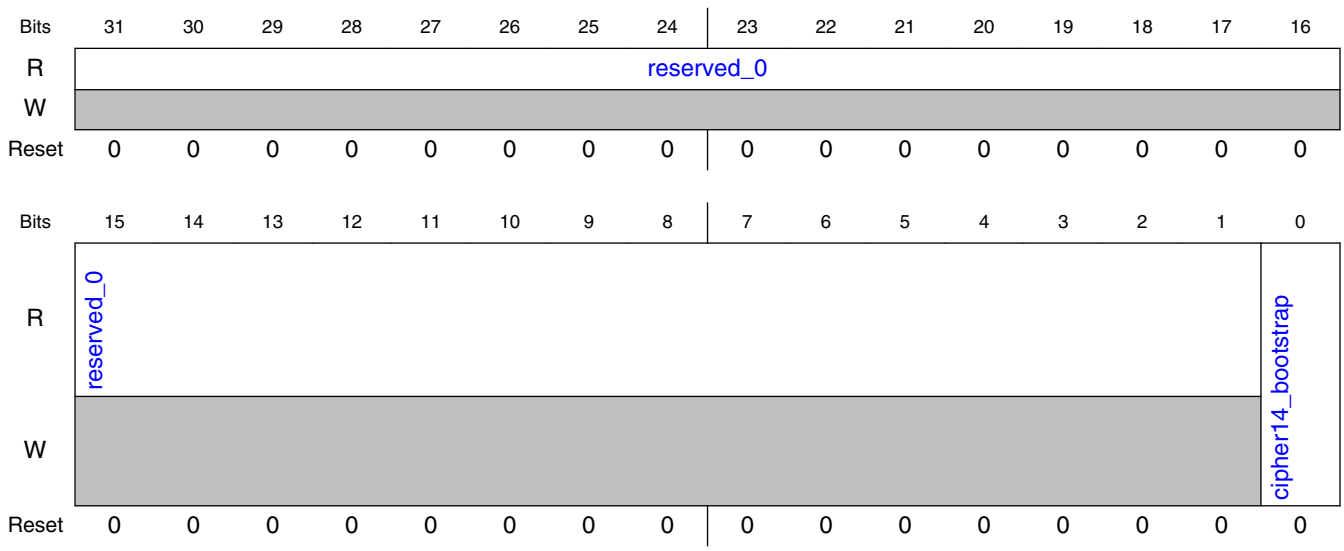
13.4.10.1.370.1 Offset

Register	Offset
CIPHER14_BOOTSTRAP	6_004Ch

13.4.10.1.370.2 Function

Cipher for HDCP1.4 bootrstap

13.4.10.1.370.3 Diagram



13.4.10.1.370.4 Fields

Field	Function
31-1 reserved_0	Reserved Reserved. Writes are ignored. 0x0 when read
0 cipher14_bootstrap	Set to 1 when authentication is finished succesfuly. Set to 1 when authentication is finished succesfuly. Otherwise 0

13.4.10.2 HDMI_TX_PHY register descriptions

13.4.10.2.1 HDMI_TX_PHY memory map

HDMI_TX_PHY base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
4000h	Power state machine control register (lane0_xcvr_psm_ctrl)	16	RW	0201h
4001h	Power state machine reset control register (lane0_xcvr_psm_rctrl)	16	RW	BCFCh
4002h	PSM calibration delay timer register (lane0_xcvr_psm_cal_tmr)	16	RW	0152h
4003h	A0 in delay timer register (lane0_xcvr_psm_a0in_tmr)	16	RW	0152h
4004h	A0 in bypass timer register (lane0_xcvr_psm_a0byp_tmr)	16	RW	0001h
4005h	A1 in delay timer register (lane0_xcvr_psm_a1in_tmr)	16	RW	0008h
4006h	A2 in delay timer register (lane0_xcvr_psm_a2in_tmr)	16	RW	0008h
4007h	A3 in delay timer register (lane0_xcvr_psm_a3in_tmr)	16	RW	0008h
4008h	A4 in delay timer register (lane0_xcvr_psm_a4in_tmr)	16	RW	0008h
4009h	A5 in delay timer register (lane0_xcvr_psm_a5in_tmr)	16	RW	0008h
400Ah	A0 out delay timer register (lane0_xcvr_psm_a0out_tmr)	16	RW	0001h
400Bh	A1 out delay timer register (lane0_xcvr_psm_a1out_tmr)	16	RW	0001h
400Ch	A2 out delay timer register (lane0_xcvr_psm_a2out_tmr)	16	RW	0001h
400Dh	A3 out delay timer register (lane0_xcvr_psm_a3out_tmr)	16	RW	0001h
400Eh	A4 out delay timer register (lane0_xcvr_psm_a4out_tmr)	16	RW	0001h
400Fh	A5 out delay timer register (lane0_xcvr_psm_a5out_tmr)	16	RW	0001h
4010h	Power state machine diagnostic register (lane0_xcvr_psm_diag)	16	RW	0000h
401Fh	Power state machine user defined control register (lane0_xcvr_psm_user_def_ctrl)	16	RW	0002h
4040h	TX coefficient controller control register (lane0_tx_txcc_ctrl)	16	RO	0000h
4041h	TX pre-cursor override register (lane0_tx_txcc_pre_ovrd)	16	RW	0000h
4042h	TX main-cursor override register (lane0_tx_txcc_main_ovrd)	16	RW	0000h
4043h	TX post-cursor override register (lane0_tx_txcc_post_ovrd)	16	RW	0000h
4044h	TX pre-cursor current value register (lane0_tx_txcc_pre_cval)	16	RO	0000h
4045h	TX main-cursor current value register (lane0_tx_txcc_main_cval)	16	RO	0000h
4046h	TX post-cursor current value register (lane0_tx_txcc_post_cval)	16	RO	0000h
4047h	Resistor calibration code scaler multiplier value register (lane0_tx_txcc_cal_sclr_mult)	16	RW	0080h
4048h	Calculated pre emphasis multiplier value 00 register (lane0_tx_txcc_cpre_mult_00)	16	RW	0000h
4049h	Calculated pre emphasis multiplier value 01 register (lane0_tx_txcc_cpre_mult_01)	16	RW	0000h
404Ah	Calculated pre emphasis multiplier value 10 register (lane0_tx_txcc_cpre_mult_10)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
404Bh	Calculated pre emphasis multiplier value 11 register (lane0_tx_txcc_cpre_mult_11)	16	RW	0000h
404Ch	Calculated post emphasis multiplier value 00 register (lane0_tx_txcc_cpost_mult_00)	16	RW	0020h
404Dh	Calculated post emphasis multiplier value 01 register (lane0_tx_txcc_cpost_mult_01)	16	RW	0015h
404Eh	Calculated post emphasis multiplier value 10 register (lane0_tx_txcc_cpost_mult_10)	16	RW	0000h
404Fh	Calculated post emphasis multiplier value 11 register (lane0_tx_txcc_cpost_mult_11)	16	RW	0000h
4050h	Margin full swing multiplier value 000 register (lane0_tx_txcc_mgnfs_mult_000)	16	RW	0000h
4051h	Margin full swing multiplier value 001 register (lane0_tx_txcc_mgnfs_mult_001)	16	RW	0000h
4052h	Margin full swing multiplier value 010 register (lane0_tx_txcc_mgnfs_mult_010)	16	RW	0001h
4053h	Margin full swing multiplier value 011 register (lane0_tx_txcc_mgnfs_mult_011)	16	RW	0008h
4054h	Margin full swing multiplier value 100 register (lane0_tx_txcc_mgnfs_mult_100)	16	RW	0011h
4055h	Margin full swing multiplier value 101 register (lane0_tx_txcc_mgnfs_mult_101)	16	RW	0019h
4056h	Margin full swing multiplier value 110 register (lane0_tx_txcc_mgnfs_mult_110)	16	RW	0021h
4057h	Margin full swing multiplier value 111 register (lane0_tx_txcc_mgnfs_mult_111)	16	RW	0029h
4058h	Margin half swing multiplier value 000 register (lane0_tx_txcc_mgnls_mult_000)	16	RW	0018h
4059h	Margin half swing multiplier value 001 register (lane0_tx_txcc_mgnls_mult_001)	16	RW	0018h
405Ah	Margin half swing multiplier value 010 register (lane0_tx_txcc_mgnls_mult_010)	16	RW	001Dh
405Bh	Margin half swing multiplier value 011 register (lane0_tx_txcc_mgnls_mult_011)	16	RW	0021h
405Ch	Margin half swing multiplier value 100 register (lane0_tx_txcc_mgnls_mult_100)	16	RW	0026h
405Dh	Margin half swing multiplier value 101 register (lane0_tx_txcc_mgnls_mult_101)	16	RW	002Bh
405Eh	Margin half swing multiplier value 110 register (lane0_tx_txcc_mgnls_mult_110)	16	RW	002Fh
405Fh	Margin half swing multiplier value 111 register (lane0_tx_txcc_mgnls_mult_111)	16	RW	0034h
40E0h	Transceiver PLL data rate clock control register (lane0_xcvr_diag_plldrc_ctrl)	16	RW	5112h
40E1h	Transceiver high speed clock select register (lane0_xcvr_diag_hsclk_sel)	16	RW	1001h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
40E2h	Transceiver high speed clock A divider control register (lane0_xcvr_diag_hsclka_dctrl)	16	RW	0000h
40E3h	Transceiver high speed clock B divider control register (lane0_xcvr_diag_hsclkb_dctrl)	16	RW	0000h
40E7h	Transceiver control reset diagnostic register (lane0_xcvr_diag_rst_diag)	16	RO	0000h
40E8h	Transceiver bidirectional control register (lane0_xcvr_diag_bidi_ctrl)	16	RW	007Fh
40E9h	Transceiver power island control register (lane0_xcvr_diag_pwr_ctrl)	16	RW	0000h
40EAh	RX Lane calibration reset timer register (lane0_xcvr_diag_rx_lane_cal_rst_tmr)	16	RW	0095h
40F0h	Lane fast common mode enable timeout register (lane0_xcvr_diag_lane_fcm_en_to)	16	RW	0000h
40F1h	Lane fast common mode enable sample wait timer register (lane0_xcvr_diag_lane_fcm_en_swait_tmr)	16	RW	0000h
40F2h	Lane fast common mode enable margin timer register (lane0_xcvr_diag_lane_fcm_en_mgn_tmr)	16	RW	0096h
40F3h	Lane fast common mode enable tuning register (lane0_xcvr_diag_lane_fcm_en_tune)	16	RW	0031h
40FFh	Transceiver digital cover your alternatives register (lane0_xcvr_diag_dcya)	16	RW	0000h
4100h	Transmitter A0 power state definition register (lane0_tx_psc_a0)	16	RW	6799h
4101h	Transmitter A1 power state definition register (lane0_tx_psc_a1)	16	RW	6798h
4102h	Transmitter A2 power state definition register (lane0_tx_psc_a2)	16	RW	0098h
4103h	Transmitter A3 power state definition register (lane0_tx_psc_a3)	16	RW	0098h
4104h	Transmitter A4 power state definition register (lane0_tx_psc_a4)	16	RW	0020h
4105h	Transmitter A5 power state definition register (lane0_tx_psc_a5)	16	RW	0000h
4106h	Transmitter calibration power state definition register (lane0_tx_psc_cal)	16	RW	0000h
4107h	Transmitter ready power state definition register (lane0_tx_psc_rdy)	16	RW	0000h
4120h	Transmit receiver detect control register (lane0_tx_rcvdet_ctrl)	16	RW	0000h
4121h	Transmit receiver detect override register (lane0_tx_rcvdet_ovrd)	16	RW	0000h
4122h	Transmit receiver detect enable timer register (lane0_tx_rcvdet_en_tmr)	16	RW	09C4h
4123h	Transmit receiver detect start timer register (lane0_tx_rcvdet_st_tmr)	16	RW	0032h
4140h	Transmit BIST control register (lane0_tx_bist_ctrl)	16	RW	0000h
4141h	Transmit BIST user defined data write register (lane0_tx_bist_uddwr)	16	WO	0000h
4142h	Transmit BIST PRBS seed 0 register (lane0_tx_bist_seed0)	16	RW	0001h
4143h	Transmit BIST PRBS seed 1 register (lane0_tx_bist_seed1)	16	RW	0000h
41E0h	TX control register (lane0_tx_diag_tx_ctrl)	16	RW	0000h
41E1h	TX driver control register (lane0_tx_diag_tx_drv)	16	RW	0000h
41E2h	TX electrical idle diagnostic register (lane0_tx_diag_elec_idle)	16	RW	0033h
41E3h	TX sync FIFO diagnostic control register (lane0_tx_diag_sfifo_ctrl)	16	RW	0008h
41E4h	TX sync FIFO diagnostic timer register (lane0_tx_diag_sfifo_tmr)	16	RW	060Ch

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
41E5h	TX receiver detect tuning register (lane0_tx_diag_rdvdet_tune)	16	RW	0000h
41E6h	Transmitter control reset diagnostic register (lane0_tx_diag_rst_diag)	16	RO	0000h
41E7h	TX bandgap reference and pre-drive enable delay register (lane0_tx_diag_bgref_predrv_delay)	16	RW	007Dh
41F0h	TX MPHY control register 1 (lane0_tx_diag_mphy_ctrl1)	16	RW	0000h
41F1h	TX MPHY control register 2 (lane0_tx_diag_mphy_ctrl2)	16	RW	0000h
41F4h	TX driver LDO programming register (lane0_tx_diag_drv_ldo_prog)	16	RW	0000h
41F5h	TX extra enable control override register (lane0_tx_diag_ectrl_ovrd)	16	RW	0000h
41FEh	Transmitter digital cover your alternatives register (lane0_tx_diag_dcya)	16	RW	0000h
41FFh	Transmitter analog cover your alternatives register (lane0_tx_diag_acya)	16	RW	0000h
4400h	Power state machine control register (lane1_xcvr_psm_ctrl)	16	RW	0201h
4401h	Power state machine reset control register (lane1_xcvr_psm_rctrl)	16	RW	BCFCh
4402h	PSM calibration delay timer register (lane1_xcvr_psm_cal_tmr)	16	RW	0152h
4403h	A0 in delay timer register (lane1_xcvr_psm_a0in_tmr)	16	RW	0152h
4404h	A0 in bypass timer register (lane1_xcvr_psm_a0byp_tmr)	16	RW	0001h
4405h	A1 in delay timer register (lane1_xcvr_psm_a1in_tmr)	16	RW	0008h
4406h	A2 in delay timer register (lane1_xcvr_psm_a2in_tmr)	16	RW	0008h
4407h	A3 in delay timer register (lane1_xcvr_psm_a3in_tmr)	16	RW	0008h
4408h	A4 in delay timer register (lane1_xcvr_psm_a4in_tmr)	16	RW	0008h
4409h	A5 in delay timer register (lane1_xcvr_psm_a5in_tmr)	16	RW	0008h
440Ah	A0 out delay timer register (lane1_xcvr_psm_a0out_tmr)	16	RW	0001h
440Bh	A1 out delay timer register (lane1_xcvr_psm_a1out_tmr)	16	RW	0001h
440Ch	A2 out delay timer register (lane1_xcvr_psm_a2out_tmr)	16	RW	0001h
440Dh	A3 out delay timer register (lane1_xcvr_psm_a3out_tmr)	16	RW	0001h
440Eh	A4 out delay timer register (lane1_xcvr_psm_a4out_tmr)	16	RW	0001h
440Fh	A5 out delay timer register (lane1_xcvr_psm_a5out_tmr)	16	RW	0001h
4410h	Power state machine diagnostic register (lane1_xcvr_psm_diag)	16	RW	0000h
441Fh	Power state machine user defined control register (lane1_xcvr_psm_user_def_ctrl)	16	RW	0002h
4440h	TX coefficient controller control register (lane1_tx_txcc_ctrl)	16	RO	0000h
4441h	TX pre-cursor override register (lane1_tx_txcc_pre_ovrd)	16	RW	0000h
4442h	TX main-cursor override register (lane1_tx_txcc_main_ovrd)	16	RW	0000h
4443h	TX post-cursor override register (lane1_tx_txcc_post_ovrd)	16	RW	0000h
4444h	TX pre-cursor current value register (lane1_tx_txcc_pre_cval)	16	RO	0000h
4445h	TX main-cursor current value register (lane1_tx_txcc_main_cval)	16	RO	0000h
4446h	TX post-cursor current value register (lane1_tx_txcc_post_cval)	16	RO	0000h
4447h	Resistor calibration code scaler multiplier value register (lane1_tx_txcc_cal_sclr_mult)	16	RW	0080h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4448h	Calculated pre emphasis multiplier value 00 register (lane1_tx_txcc_cpre_mult_00)	16	RW	0000h
4449h	Calculated pre emphasis multiplier value 01 register (lane1_tx_txcc_cpre_mult_01)	16	RW	0000h
444Ah	Calculated pre emphasis multiplier value 10 register (lane1_tx_txcc_cpre_mult_10)	16	RW	0000h
444Bh	Calculated pre emphasis multiplier value 11 register (lane1_tx_txcc_cpre_mult_11)	16	RW	0000h
444Ch	Calculated post emphasis multiplier value 00 register (lane1_tx_txcc_cpost_mult_00)	16	RW	0020h
444Dh	Calculated post emphasis multiplier value 01 register (lane1_tx_txcc_cpost_mult_01)	16	RW	0015h
444Eh	Calculated post emphasis multiplier value 10 register (lane1_tx_txcc_cpost_mult_10)	16	RW	0000h
444Fh	Calculated post emphasis multiplier value 11 register (lane1_tx_txcc_cpost_mult_11)	16	RW	0000h
4450h	Margin full swing multiplier value 000 register (lane1_tx_txcc_mgnfs_mult_000)	16	RW	0000h
4451h	Margin full swing multiplier value 001 register (lane1_tx_txcc_mgnfs_mult_001)	16	RW	0000h
4452h	Margin full swing multiplier value 010 register (lane1_tx_txcc_mgnfs_mult_010)	16	RW	0001h
4453h	Margin full swing multiplier value 011 register (lane1_tx_txcc_mgnfs_mult_011)	16	RW	0008h
4454h	Margin full swing multiplier value 100 register (lane1_tx_txcc_mgnfs_mult_100)	16	RW	0011h
4455h	Margin full swing multiplier value 101 register (lane1_tx_txcc_mgnfs_mult_101)	16	RW	0019h
4456h	Margin full swing multiplier value 110 register (lane1_tx_txcc_mgnfs_mult_110)	16	RW	0021h
4457h	Margin full swing multiplier value 111 register (lane1_tx_txcc_mgnfs_mult_111)	16	RW	0029h
4458h	Margin half swing multiplier value 000 register (lane1_tx_txcc_mgnls_mult_000)	16	RW	0018h
4459h	Margin half swing multiplier value 001 register (lane1_tx_txcc_mgnls_mult_001)	16	RW	0018h
445Ah	Margin half swing multiplier value 010 register (lane1_tx_txcc_mgnls_mult_010)	16	RW	001Dh
445Bh	Margin half swing multiplier value 011 register (lane1_tx_txcc_mgnls_mult_011)	16	RW	0021h
445Ch	Margin half swing multiplier value 100 register (lane1_tx_txcc_mgnls_mult_100)	16	RW	0026h
445Dh	Margin half swing multiplier value 101 register (lane1_tx_txcc_mgnls_mult_101)	16	RW	002Bh
445Eh	Margin half swing multiplier value 110 register (lane1_tx_txcc_mgnls_mult_110)	16	RW	002Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
445Fh	Margin half swing multiplier value 111 register (lane1_tx_txcc_mgnls_mult_111)	16	RW	0034h
44E0h	Transceiver PLL data rate clock control register (lane1_xcvr_diag_plldrc_ctrl)	16	RW	5112h
44E1h	Transceiver high speed clock select register (lane1_xcvr_diag_hsclk_sel)	16	RW	1001h
44E2h	Transceiver high speed clock A divider control register (lane1_xcvr_diag_hsclka_dctrl)	16	RW	0000h
44E3h	Transceiver high speed clock B divider control register (lane1_xcvr_diag_hsclkb_dctrl)	16	RW	0000h
44E7h	Transceiver control reset diagnostic register (lane1_xcvr_diag_rst_diag)	16	RO	0000h
44E8h	Transceiver bidirectional control register (lane1_xcvr_diag_bidi_ctrl)	16	RW	007Fh
44E9h	Transceiver power island control register (lane1_xcvr_diag_pwr_ctrl)	16	RW	0000h
44EAh	RX Lane calibration reset timer register (lane1_xcvr_diag_rx_lane_cal_rst_tmr)	16	RW	0095h
44F0h	Lane fast common mode enable timeout register (lane1_xcvr_diag_lane_fcm_en_to)	16	RW	0000h
44F1h	Lane fast common mode enable sample wait timer register (lane1_xcvr_diag_lane_fcm_en_swait_tmr)	16	RW	0000h
44F2h	Lane fast common mode enable margin timer register (lane1_xcvr_diag_lane_fcm_en_mgn_tmr)	16	RW	0096h
44F3h	Lane fast common mode enable tuning register (lane1_xcvr_diag_lane_fcm_en_tune)	16	RW	0031h
44FFh	Transceiver digital cover your alternatives register (lane1_xcvr_diag_dcya)	16	RW	0000h
4500h	Transmitter A0 power state definition register (lane1_tx_psc_a0)	16	RW	6799h
4501h	Transmitter A1 power state definition register (lane1_tx_psc_a1)	16	RW	6798h
4502h	Transmitter A2 power state definition register (lane1_tx_psc_a2)	16	RW	0098h
4503h	Transmitter A3 power state definition register (lane1_tx_psc_a3)	16	RW	0098h
4504h	Transmitter A4 power state definition register (lane1_tx_psc_a4)	16	RW	0020h
4505h	Transmitter A5 power state definition register (lane1_tx_psc_a5)	16	RW	0000h
4506h	Transmitter calibration power state definition register (lane1_tx_psc_cal)	16	RW	0000h
4507h	Transmitter ready power state definition register (lane1_tx_psc_rdy)	16	RW	0000h
4520h	Transmit receiver detect control register (lane1_tx_rcvdet_ctrl)	16	RW	0000h
4521h	Transmit receiver detect override register (lane1_tx_rcvdet_ovrd)	16	RW	0000h
4522h	Transmit receiver detect enable timer register (lane1_tx_rcvdet_en_tmr)	16	RW	09C4h
4523h	Transmit receiver detect start timer register (lane1_tx_rcvdet_st_tmr)	16	RW	0032h
4540h	Transmit BIST control register (lane1_tx_bist_ctrl)	16	RW	0000h
4541h	Transmit BIST user defined data write register (lane1_tx_bist_uddwr)	16	WO	0000h
4542h	Transmit BIST PRBS seed 0 register (lane1_tx_bist_seed0)	16	RW	0001h
4543h	Transmit BIST PRBS seed 1 register (lane1_tx_bist_seed1)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
45E0h	TX control register (lane1_tx_diag_tx_ctrl)	16	RW	0000h
45E1h	TX driver control register (lane1_tx_diag_tx_drv)	16	RW	0000h
45E2h	TX electrical idle diagnostic register (lane1_tx_diag_elec_idle)	16	RW	0033h
45E3h	TX sync FIFO diagnostic control register (lane1_tx_diag_sfifo_ctrl)	16	RW	0008h
45E4h	TX sync FIFO diagnostic timer register (lane1_tx_diag_sfifo_tmr)	16	RW	060Ch
45E5h	TX receiver detect tuning register (lane1_tx_diag_rdvdet_tune)	16	RW	0000h
45E6h	Transmitter control reset diagnostic register (lane1_tx_diag_rst_diag)	16	RO	0000h
45E7h	TX bandgap reference and pre-drive enable delay register (lane1_tx_diag_bgref_predrv_delay)	16	RW	007Dh
45F0h	TX MPHY control register 1 (lane1_tx_diag_mphy_ctrl1)	16	RW	0000h
45F1h	TX MPHY control register 2 (lane1_tx_diag_mphy_ctrl2)	16	RW	0000h
45F4h	TX driver LDO programming register (lane1_tx_diag_drv_ldo_prog)	16	RW	0000h
45F5h	TX extra enable control override register (lane1_tx_diag_ectrl_ovrd)	16	RW	0000h
45FEh	Transmitter digital cover your alternatives register (lane1_tx_diag_dcya)	16	RW	0000h
45FFh	Transmitter analog cover your alternatives register (lane1_tx_diag_acya)	16	RW	0000h
4800h	Power state machine control register (lane2_xcvr_psm_ctrl)	16	RW	0201h
4801h	Power state machine reset control register (lane2_xcvr_psm_rctrl)	16	RW	BCFCh
4802h	PSM calibration delay timer register (lane2_xcvr_psm_cal_tmr)	16	RW	0152h
4803h	A0 in delay timer register (lane2_xcvr_psm_a0in_tmr)	16	RW	0152h
4804h	A0 in bypass timer register (lane2_xcvr_psm_a0byp_tmr)	16	RW	0001h
4805h	A1 in delay timer register (lane2_xcvr_psm_a1in_tmr)	16	RW	0008h
4806h	A2 in delay timer register (lane2_xcvr_psm_a2in_tmr)	16	RW	0008h
4807h	A3 in delay timer register (lane2_xcvr_psm_a3in_tmr)	16	RW	0008h
4808h	A4 in delay timer register (lane2_xcvr_psm_a4in_tmr)	16	RW	0008h
4809h	A5 in delay timer register (lane2_xcvr_psm_a5in_tmr)	16	RW	0008h
480Ah	A0 out delay timer register (lane2_xcvr_psm_a0out_tmr)	16	RW	0001h
480Bh	A1 out delay timer register (lane2_xcvr_psm_a1out_tmr)	16	RW	0001h
480Ch	A2 out delay timer register (lane2_xcvr_psm_a2out_tmr)	16	RW	0001h
480Dh	A3 out delay timer register (lane2_xcvr_psm_a3out_tmr)	16	RW	0001h
480Eh	A4 out delay timer register (lane2_xcvr_psm_a4out_tmr)	16	RW	0001h
480Fh	A5 out delay timer register (lane2_xcvr_psm_a5out_tmr)	16	RW	0001h
4810h	Power state machine diagnostic register (lane2_xcvr_psm_diag)	16	RW	0000h
481Fh	Power state machine user defined control register (lane2_xcvr_psm_user_def_ctrl)	16	RW	0002h
4840h	TX coefficient controller control register (lane2_tx_txcc_ctrl)	16	RO	0000h
4841h	TX pre-cursor override register (lane2_tx_txcc_pre_ovrd)	16	RW	0000h
4842h	TX main-cursor override register (lane2_tx_txcc_main_ovrd)	16	RW	0000h
4843h	TX post-cursor override register (lane2_tx_txcc_post_ovrd)	16	RW	0000h
4844h	TX pre-cursor current value register (lane2_tx_txcc_pre_cval)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4845h	TX main-cursor current value register (lane2_tx_txcc_main_cval)	16	RO	0000h
4846h	TX post-cursor current value register (lane2_tx_txcc_post_cval)	16	RO	0000h
4847h	Resistor calibration code scaler multiplier value register (lane2_tx_txcc_cal_sclr_mult)	16	RW	0080h
4848h	Calculated pre emphasis multiplier value 00 register (lane2_tx_txcc_cpre_mult_00)	16	RW	0000h
4849h	Calculated pre emphasis multiplier value 01 register (lane2_tx_txcc_cpre_mult_01)	16	RW	0000h
484Ah	Calculated pre emphasis multiplier value 10 register (lane2_tx_txcc_cpre_mult_10)	16	RW	0000h
484Bh	Calculated pre emphasis multiplier value 11 register (lane2_tx_txcc_cpre_mult_11)	16	RW	0000h
484Ch	Calculated post emphasis multiplier value 00 register (lane2_tx_txcc_cpost_mult_00)	16	RW	0020h
484Dh	Calculated post emphasis multiplier value 01 register (lane2_tx_txcc_cpost_mult_01)	16	RW	0015h
484Eh	Calculated post emphasis multiplier value 10 register (lane2_tx_txcc_cpost_mult_10)	16	RW	0000h
484Fh	Calculated post emphasis multiplier value 11 register (lane2_tx_txcc_cpost_mult_11)	16	RW	0000h
4850h	Margin full swing multiplier value 000 register (lane2_tx_txcc_mgnfs_mult_000)	16	RW	0000h
4851h	Margin full swing multiplier value 001 register (lane2_tx_txcc_mgnfs_mult_001)	16	RW	0000h
4852h	Margin full swing multiplier value 010 register (lane2_tx_txcc_mgnfs_mult_010)	16	RW	0001h
4853h	Margin full swing multiplier value 011 register (lane2_tx_txcc_mgnfs_mult_011)	16	RW	0008h
4854h	Margin full swing multiplier value 100 register (lane2_tx_txcc_mgnfs_mult_100)	16	RW	0011h
4855h	Margin full swing multiplier value 101 register (lane2_tx_txcc_mgnfs_mult_101)	16	RW	0019h
4856h	Margin full swing multiplier value 110 register (lane2_tx_txcc_mgnfs_mult_110)	16	RW	0021h
4857h	Margin full swing multiplier value 111 register (lane2_tx_txcc_mgnfs_mult_111)	16	RW	0029h
4858h	Margin half swing multiplier value 000 register (lane2_tx_txcc_mgnls_mult_000)	16	RW	0018h
4859h	Margin half swing multiplier value 001 register (lane2_tx_txcc_mgnls_mult_001)	16	RW	0018h
485Ah	Margin half swing multiplier value 010 register (lane2_tx_txcc_mgnls_mult_010)	16	RW	001Dh
485Bh	Margin half swing multiplier value 011 register (lane2_tx_txcc_mgnls_mult_011)	16	RW	0021h
485Ch	Margin half swing multiplier value 100 register (lane2_tx_txcc_mgnls_mult_100)	16	RW	0026h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
485Dh	Margin half swing multiplier value 101 register (lane2_tx_txcc_mgnls_mult_101)	16	RW	002Bh
485Eh	Margin half swing multiplier value 110 register (lane2_tx_txcc_mgnls_mult_110)	16	RW	002Fh
485Fh	Margin half swing multiplier value 111 register (lane2_tx_txcc_mgnls_mult_111)	16	RW	0034h
48E0h	Transceiver PLL data rate clock control register (lane2_xcvr_diag_plldrc_ctrl)	16	RW	5112h
48E1h	Transceiver high speed clock select register (lane2_xcvr_diag_hsclk_sel)	16	RW	1001h
48E2h	Transceiver high speed clock A divider control register (lane2_xcvr_diag_hsclka_dctrl)	16	RW	0000h
48E3h	Transceiver high speed clock B divider control register (lane2_xcvr_diag_hsclkb_dctrl)	16	RW	0000h
48E7h	Transceiver control reset diagnostic register (lane2_xcvr_diag_rst_diag)	16	RO	0000h
48E8h	Transceiver bidirectional control register (lane2_xcvr_diag_bidi_ctrl)	16	RW	007Fh
48E9h	Transceiver power island control register (lane2_xcvr_diag_pwr_ctrl)	16	RW	0000h
48EAh	RX Lane calibration reset timer register (lane2_xcvr_diag_rx_lane_cal_rst_tmr)	16	RW	0095h
48F0h	Lane fast common mode enable timeout register (lane2_xcvr_diag_lane_fcm_en_to)	16	RW	0000h
48F1h	Lane fast common mode enable sample wait timer register (lane2_xcvr_diag_lane_fcm_en_swait_tmr)	16	RW	0000h
48F2h	Lane fast common mode enable margin timer register (lane2_xcvr_diag_lane_fcm_en_mgn_tmr)	16	RW	0096h
48F3h	Lane fast common mode enable tuning register (lane2_xcvr_diag_lane_fcm_en_tune)	16	RW	0031h
48FFh	Transceiver digital cover your alternatives register (lane2_xcvr_diag_dcya)	16	RW	0000h
4900h	Transmitter A0 power state definition register (lane2_tx_psc_a0)	16	RW	6799h
4901h	Transmitter A1 power state definition register (lane2_tx_psc_a1)	16	RW	6798h
4902h	Transmitter A2 power state definition register (lane2_tx_psc_a2)	16	RW	0098h
4903h	Transmitter A3 power state definition register (lane2_tx_psc_a3)	16	RW	0098h
4904h	Transmitter A4 power state definition register (lane2_tx_psc_a4)	16	RW	0020h
4905h	Transmitter A5 power state definition register (lane2_tx_psc_a5)	16	RW	0000h
4906h	Transmitter calibration power state definition register (lane2_tx_psc_cal)	16	RW	0000h
4907h	Transmitter ready power state definition register (lane2_tx_psc_rdy)	16	RW	0000h
4920h	Transmit receiver detect control register (lane2_tx_rcvdet_ctrl)	16	RW	0000h
4921h	Transmit receiver detect override register (lane2_tx_rcvdet_ovrd)	16	RW	0000h
4922h	Transmit receiver detect enable timer register (lane2_tx_rcvdet_en_tmr)	16	RW	09C4h
4923h	Transmit receiver detect start timer register (lane2_tx_rcvdet_st_tmr)	16	RW	0032h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4940h	Transmit BIST control register (lane2_tx_bist_ctrl)	16	RW	0000h
4941h	Transmit BIST user defined data write register (lane2_tx_bist_uddwr)	16	WO	0000h
4942h	Transmit BIST PRBS seed 0 register (lane2_tx_bist_seed0)	16	RW	0001h
4943h	Transmit BIST PRBS seed 1 register (lane2_tx_bist_seed1)	16	RW	0000h
49E0h	TX control register (lane2_tx_diag_tx_ctrl)	16	RW	0000h
49E1h	TX driver control register (lane2_tx_diag_tx_drv)	16	RW	0000h
49E2h	TX electrical idle diagnostic register (lane2_tx_diag_elec_idle)	16	RW	0033h
49E3h	TX sync FIFO diagnostic control register (lane2_tx_diag_sfifo_ctrl)	16	RW	0008h
49E4h	TX sync FIFO diagnostic timer register (lane2_tx_diag_sfifo_tmr)	16	RW	060Ch
49E5h	TX receiver detect tuning register (lane2_tx_diag_rdvdet_tune)	16	RW	0000h
49E6h	Transmitter control reset diagnostic register (lane2_tx_diag_rst_diag)	16	RO	0000h
49E7h	TX bandgap reference and pre-drive enable delay register (lane2_tx_diag_bgref_predrv_delay)	16	RW	007Dh
49F0h	TX MPHY control register 1 (lane2_tx_diag_mphy_ctrl1)	16	RW	0000h
49F1h	TX MPHY control register 2 (lane2_tx_diag_mphy_ctrl2)	16	RW	0000h
49F4h	TX driver LDO programming register (lane2_tx_diag_drv_ldo_prog)	16	RW	0000h
49F5h	TX extra enable control override register (lane2_tx_diag_ectrl_ovrd)	16	RW	0000h
49FEh	Transmitter digital cover your alternatives register (lane2_tx_diag_dcya)	16	RW	0000h
49FFh	Transmitter analog cover your alternatives register (lane2_tx_diag_acya)	16	RW	0000h
4C00h	Power state machine control register (lane3_xcvr_psm_ctrl)	16	RW	0201h
4C01h	Power state machine reset control register (lane3_xcvr_psm_rctrl)	16	RW	BCFCh
4C02h	PSM calibration delay timer register (lane3_xcvr_psm_cal_tmr)	16	RW	0152h
4C03h	A0 in delay timer register (lane3_xcvr_psm_a0in_tmr)	16	RW	0152h
4C04h	A0 in bypass timer register (lane3_xcvr_psm_a0byp_tmr)	16	RW	0001h
4C05h	A1 in delay timer register (lane3_xcvr_psm_a1in_tmr)	16	RW	0008h
4C06h	A2 in delay timer register (lane3_xcvr_psm_a2in_tmr)	16	RW	0008h
4C07h	A3 in delay timer register (lane3_xcvr_psm_a3in_tmr)	16	RW	0008h
4C08h	A4 in delay timer register (lane3_xcvr_psm_a4in_tmr)	16	RW	0008h
4C09h	A5 in delay timer register (lane3_xcvr_psm_a5in_tmr)	16	RW	0008h
4C0Ah	A0 out delay timer register (lane3_xcvr_psm_a0out_tmr)	16	RW	0001h
4C0Bh	A1 out delay timer register (lane3_xcvr_psm_a1out_tmr)	16	RW	0001h
4C0Ch	A2 out delay timer register (lane3_xcvr_psm_a2out_tmr)	16	RW	0001h
4C0Dh	A3 out delay timer register (lane3_xcvr_psm_a3out_tmr)	16	RW	0001h
4C0Eh	A4 out delay timer register (lane3_xcvr_psm_a4out_tmr)	16	RW	0001h
4C0Fh	A5 out delay timer register (lane3_xcvr_psm_a5out_tmr)	16	RW	0001h
4C10h	Power state machine diagnostic register (lane3_xcvr_psm_diag)	16	RW	0000h
4C1Fh	Power state machine user defined control register (lane3_xcvr_psm_user_def_ctrl)	16	RW	0002h
4C40h	TX coefficient controller control register (lane3_tx_txcc_ctrl)	16	RO	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4C41h	TX pre-cursor override register (lane3_tx_txcc_pre_ovrd)	16	RW	0000h
4C42h	TX main-cursor override register (lane3_tx_txcc_main_ovrd)	16	RW	0000h
4C43h	TX post-cursor override register (lane3_tx_txcc_post_ovrd)	16	RW	0000h
4C44h	TX pre-cursor current value register (lane3_tx_txcc_pre_cval)	16	RO	0000h
4C45h	TX main-cursor current value register (lane3_tx_txcc_main_cval)	16	RO	0000h
4C46h	TX post-cursor current value register (lane3_tx_txcc_post_cval)	16	RO	0000h
4C47h	Resistor calibration code scaler multiplier value register (lane3_tx_txcc_cal_sclr_mult)	16	RW	0080h
4C48h	Calculated pre emphasis multiplier value 00 register (lane3_tx_txcc_cpre_mult_00)	16	RW	0000h
4C49h	Calculated pre emphasis multiplier value 01 register (lane3_tx_txcc_cpre_mult_01)	16	RW	0000h
4C4Ah	Calculated pre emphasis multiplier value 10 register (lane3_tx_txcc_cpre_mult_10)	16	RW	0000h
4C4Bh	Calculated pre emphasis multiplier value 11 register (lane3_tx_txcc_cpre_mult_11)	16	RW	0000h
4C4Ch	Calculated post emphasis multiplier value 00 register (lane3_tx_txcc_cpost_mult_00)	16	RW	0020h
4C4Dh	Calculated post emphasis multiplier value 01 register (lane3_tx_txcc_cpost_mult_01)	16	RW	0015h
4C4Eh	Calculated post emphasis multiplier value 10 register (lane3_tx_txcc_cpost_mult_10)	16	RW	0000h
4C4Fh	Calculated post emphasis multiplier value 11 register (lane3_tx_txcc_cpost_mult_11)	16	RW	0000h
4C50h	Margin full swing multiplier value 000 register (lane3_tx_txcc_mgnfs_mult_000)	16	RW	0000h
4C51h	Margin full swing multiplier value 001 register (lane3_tx_txcc_mgnfs_mult_001)	16	RW	0000h
4C52h	Margin full swing multiplier value 010 register (lane3_tx_txcc_mgnfs_mult_010)	16	RW	0001h
4C53h	Margin full swing multiplier value 011 register (lane3_tx_txcc_mgnfs_mult_011)	16	RW	0008h
4C54h	Margin full swing multiplier value 100 register (lane3_tx_txcc_mgnfs_mult_100)	16	RW	0011h
4C55h	Margin full swing multiplier value 101 register (lane3_tx_txcc_mgnfs_mult_101)	16	RW	0019h
4C56h	Margin full swing multiplier value 110 register (lane3_tx_txcc_mgnfs_mult_110)	16	RW	0021h
4C57h	Margin full swing multiplier value 111 register (lane3_tx_txcc_mgnfs_mult_111)	16	RW	0029h
4C58h	Margin half swing multiplier value 000 register (lane3_tx_txcc_mgnls_mult_000)	16	RW	0018h
4C59h	Margin half swing multiplier value 001 register (lane3_tx_txcc_mgnls_mult_001)	16	RW	0018h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4C5Ah	Margin half swing multiplier value 010 register (lane3_tx_txcc_mgnls_mult_010)	16	RW	001Dh
4C5Bh	Margin half swing multiplier value 011 register (lane3_tx_txcc_mgnls_mult_011)	16	RW	0021h
4C5Ch	Margin half swing multiplier value 100 register (lane3_tx_txcc_mgnls_mult_100)	16	RW	0026h
4C5Dh	Margin half swing multiplier value 101 register (lane3_tx_txcc_mgnls_mult_101)	16	RW	002Bh
4C5Eh	Margin half swing multiplier value 110 register (lane3_tx_txcc_mgnls_mult_110)	16	RW	002Fh
4C5Fh	Margin half swing multiplier value 111 register (lane3_tx_txcc_mgnls_mult_111)	16	RW	0034h
4CE0h	Transceiver PLL data rate clock control register (lane3_xcvr_diag_plldrc_ctrl)	16	RW	5112h
4CE1h	Transceiver high speed clock select register (lane3_xcvr_diag_hsclk_sel)	16	RW	1001h
4CE2h	Transceiver high speed clock A divider control register (lane3_xcvr_diag_hsclka_dctrl)	16	RW	0000h
4CE3h	Transceiver high speed clock B divider control register (lane3_xcvr_diag_hsclkb_dctrl)	16	RW	0000h
4CE7h	Transceiver control reset diagnostic register (lane3_xcvr_diag_rst_diag)	16	RO	0000h
4CE8h	Transceiver bidirectional control register (lane3_xcvr_diag_bidi_ctrl)	16	RW	007Fh
4CE9h	Transceiver power island control register (lane3_xcvr_diag_pwr_ctrl)	16	RW	0000h
4CEAh	RX Lane calibration reset timer register (lane3_xcvr_diag_rx_lane_cal_rst_tmr)	16	RW	0095h
4CF0h	Lane fast common mode enable timeout register (lane3_xcvr_diag_lane_fcm_en_to)	16	RW	0000h
4CF1h	Lane fast common mode enable sample wait timer register (lane3_xcvr_diag_lane_fcm_en_swait_tmr)	16	RW	0000h
4CF2h	Lane fast common mode enable margin timer register (lane3_xcvr_diag_lane_fcm_en_mgn_tmr)	16	RW	0096h
4CF3h	Lane fast common mode enable tuning register (lane3_xcvr_diag_lane_fcm_en_tune)	16	RW	0031h
4CFFh	Transceiver digital cover your alternatives register (lane3_xcvr_diag_dcya)	16	RW	0000h
4D00h	Transmitter A0 power state definition register (lane3_tx_psc_a0)	16	RW	6799h
4D01h	Transmitter A1 power state definition register (lane3_tx_psc_a1)	16	RW	6798h
4D02h	Transmitter A2 power state definition register (lane3_tx_psc_a2)	16	RW	0098h
4D03h	Transmitter A3 power state definition register (lane3_tx_psc_a3)	16	RW	0098h
4D04h	Transmitter A4 power state definition register (lane3_tx_psc_a4)	16	RW	0020h
4D05h	Transmitter A5 power state definition register (lane3_tx_psc_a5)	16	RW	0000h
4D06h	Transmitter calibration power state definition register (lane3_tx_psc_cal)	16	RW	0000h
4D07h	Transmitter ready power state definition register (lane3_tx_psc_rdy)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
4D20h	Transmit receiver detect control register (lane3_tx_rcvdet_ctrl)	16	RW	0000h
4D21h	Transmit receiver detect override register (lane3_tx_rcvdet_ovrd)	16	RW	0000h
4D22h	Transmit receiver detect enable timer register (lane3_tx_rcvdet_en_tmr)	16	RW	09C4h
4D23h	Transmit receiver detect start timer register (lane3_tx_rcvdet_st_tmr)	16	RW	0032h
4D40h	Transmit BIST control register (lane3_tx_bist_ctrl)	16	RW	0000h
4D41h	Transmit BIST user defined data write register (lane3_tx_bist_uddwr)	16	WO	0000h
4D42h	Transmit BIST PRBS seed 0 register (lane3_tx_bist_seed0)	16	RW	0001h
4D43h	Transmit BIST PRBS seed 1 register (lane3_tx_bist_seed1)	16	RW	0000h
4DE0h	TX control register (lane3_tx_diag_tx_ctrl)	16	RW	0000h
4DE1h	TX driver control register (lane3_tx_diag_tx_drv)	16	RW	0000h
4DE2h	TX electrical idle diagnostic register (lane3_tx_diag_elec_idle)	16	RW	0033h
4DE3h	TX sync FIFO diagnostic control register (lane3_tx_diag_sfifo_ctrl)	16	RW	0008h
4DE4h	TX sync FIFO diagnostic timer register (lane3_tx_diag_sfifo_tmr)	16	RW	060Ch
4DE5h	TX receiver detect tuning register (lane3_tx_diag_rdvdet_tune)	16	RW	0000h
4DE6h	Transmitter control reset diagnostic register (lane3_tx_diag_rst_diag)	16	RO	0000h
4DE7h	TX bandgap reference and pre-drive enable delay register (lane3_tx_diag_bgref_predrv_delay)	16	RW	007Dh
4DF0h	TX MPHY control register 1 (lane3_tx_diag_mphy_ctrl1)	16	RW	0000h
4DF1h	TX MPHY control register 2 (lane3_tx_diag_mphy_ctrl2)	16	RW	0000h
4DF4h	TX driver LDO programming register (lane3_tx_diag_drv_ldo_prog)	16	RW	0000h
4DF5h	TX extra enable control override register (lane3_tx_diag_ectrl_ovrd)	16	RW	0000h
4DFEh	Transmitter digital cover your alternatives register (lane3_tx_diag_dcya)	16	RW	0000h
4DFFh	Transmitter analog cover your alternatives register (lane3_tx_diag_acya)	16	RW	0000h
8000h	Receiver A0 power state definition register (lane0_rx_psc_a0)	16	RW	8BFDh
8001h	Receiver A1 power state definition register (lane0_rx_psc_a1)	16	RW	8BFDh
8002h	Receiver A2 power state definition register (lane0_rx_psc_a2)	16	RW	8910h
8003h	Receiver A3 power state definition register (lane0_rx_psc_a3)	16	RW	0000h
8004h	Receiver A4 power state definition register (lane0_rx_psc_a4)	16	RW	1000h
8005h	Receiver A5 power state definition register (lane0_rx_psc_a5)	16	RW	0000h
8006h	Receiver calibration power state definition register (lane0_rx_psc_cal)	16	RW	03FFh
8007h	Receiver ready power state definition register (lane0_rx_psc_rdy)	16	RW	0000h
8020h	RX IQ PI ILL calibration control register (lane0_rx_iqpi_ill_cal_ctrl)	16	RW	0018h
8021h	RX IQ PI ILL calibration start point register (lane0_rx_iqpi_ill_cal_start)	16	RW	1018h
8022h	RX IQ PI ILL calibration timer control register (lane0_rx_iqpi_ill_cal_tctrl)	16	RW	0003h
8023h	RX IQ PI ILL calibration override register (lane0_rx_iqpi_ill_cal_ovrd)	16	RW	0000h
8024h	RX IQ PI ILL calibration initialization timer register (lane0_rx_iqpi_ill_cal_init_tmr)	16	RW	02EEh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8025h	RX IQ PI ILL calibration iteration timer register (lane0_rx_iqpi_ill_cal_iter_tmr)	16	RW	0080h
8026h	RX IQ PI ILL lock reference timer start value register (lane0_rx_iqpi_ill_lock_reftmr_start)	16	RW	01FFh
8028h	RX IQ PI ILL lock calibration counter start value standard mode 0 register (lane0_rx_iqpi_ill_lock_calcnt_start_0)	16	RW	007Fh
8029h	RX IQ PI ILL lock calibration counter start value standard mode 1 register (lane0_rx_iqpi_ill_lock_calcnt_start_1)	16	RW	00FFh
802Ah	RX IQ PI ILL lock calibration counter start value standard mode 2 register (lane0_rx_iqpi_ill_lock_calcnt_start_2)	16	RW	00FFh
802Bh	RX IQ PI ILL lock calibration counter start value standard mode 3 register (lane0_rx_iqpi_ill_lock_calcnt_start_3)	16	RW	0000h
8030h	RX E PI ILL calibration control register (lane0_rx_epi_ill_cal_ctrl)	16	RW	0018h
8031h	RX E PI ILL calibration start point register (lane0_rx_epi_ill_cal_start)	16	RW	1018h
8032h	RX E PI ILL calibration timer control register (lane0_rx_epi_ill_cal_tctrl)	16	RW	0003h
8033h	RX E PI ILL calibration override register (lane0_rx_epi_ill_cal_ovrd)	16	RW	0000h
8034h	RX E PI ILL calibration initialization timer register (lane0_rx_epi_ill_cal_init_tmr)	16	RW	02EEh
8035h	RX E PI ILL calibration iteration timer register (lane0_rx_epi_ill_cal_iter_tmr)	16	RW	0080h
8036h	RX E PI ILL lock reference timer start value register (lane0_rx_epi_ill_lock_reftmr_start)	16	RW	01FFh
8038h	RX E PI ILL lock calibration counter start value standard mode 0 register (lane0_rx_epi_ill_lock_calcnt_start_0)	16	RW	007Fh
8039h	RX E PI ILL lock calibration counter start value standard mode 1 register (lane0_rx_epi_ill_lock_calcnt_start_1)	16	RW	00FFh
803Ah	RX E PI ILL lock calibration counter start value standard mode 2 register (lane0_rx_epi_ill_lock_calcnt_start_2)	16	RW	00FFh
803Bh	RX E PI ILL lock calibration counter start value standard mode 3 register (lane0_rx_epi_ill_lock_calcnt_start_3)	16	RW	0000h
8040h	Signal detect calibration 0 control register (lane0_rx_sdcal0_ctrl)	16	RW	0000h
8041h	Signal detect calibration 0 override register (lane0_rx_sdcal0_ovrd)	16	RW	0000h
8042h	Signal detect calibration 0 start register (lane0_rx_sdcal0_start)	16	RW	0000h
8043h	Signal detect calibration 0 tune register (lane0_rx_sdcal0_tune)	16	RW	0000h
8044h	Signal detect calibration 0 initialization timer register (lane0_rx_sdcal0_init_tmr)	16	RW	0019h
8045h	Signal detect calibration 0 iteration timer register (lane0_rx_sdcal0_iter_tmr)	16	RW	007Dh
8048h	Signal detect calibration 1 control register (lane0_rx_sdcal1_ctrl)	16	RW	0000h
8049h	Signal detect calibration 1 override register (lane0_rx_sdcal1_ovrd)	16	RW	0000h
804Ah	Signal detect calibration 1 start register (lane0_rx_sdcal1_start)	16	RW	0000h
804Bh	Signal detect calibration 1 tune register (lane0_rx_sdcal1_tune)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
804Ch	Signal detect calibration 1 initialization timer register (lane0_rx_sdcal1_init_tmr)	16	RW	0019h
804Dh	Signal detect calibration 1 iteration timer register (lane0_rx_sdcal1_iter_tmr)	16	RW	007Dh
8058h	Sampler error DAC control register (lane0_rx_samp_dac_ctrl)	16	RW	0014h
8080h	CDRLF configuration register (lane0_rx_cdrlf_cnfg)	16	RW	20B3h
8081h	CDRLF configuration register 2 (lane0_rx_cdrlf_cnfg2)	16	RW	0000h
8082h	CDRLF margin diagnostic register (lane0_rx_cdrlf_mgn_diag)	16	RW	0000h
8083h	CDRLF fast phase lock timer value register 0 (lane0_rx_cdrlf_fpl_tmr0)	16	RW	0071h
8084h	CDRLF fast phase lock timer value register 1 (lane0_rx_cdrlf_fpl_tmr1)	16	RW	0731h
8085h	CDRLF fast frequency lock timer value register (lane0_rx_cdrlf_ffl_tmr)	16	RW	0018h
8088h	CDRLF fast frequency lock step 0 control register (lane0_rx_cdrlf_ffl0_ctrl)	16	RW	460Ch
8089h	CDRLF fast frequency lock step 1 control register (lane0_rx_cdrlf_ffl1_ctrl)	16	RW	4A0Ah
808Ah	CDRLF fast frequency lock step 2 control register (lane0_rx_cdrlf_ffl2_ctrl)	16	RW	5414h
808Bh	CDRLF fast frequency lock step 3 control register (lane0_rx_cdrlf_ffl3_ctrl)	16	RW	5E1Eh
808Ch	CDRLF fast frequency lock step 4 control register (lane0_rx_cdrlf_ffl4_ctrl)	16	RW	9E1Eh
8090h	Receiver signal detect filter high to low filter timer register (lane0_rx_sigdet_hl_filt_tmr)	16	RW	0019h
8091h	Receiver signal detect filter high to low delay timer register (lane0_rx_sigdet_hl_dly_tmr)	16	RW	0000h
8092h	Receiver signal detect filter high to low min timer register (lane0_rx_sigdet_hl_min_tmr)	16	RW	0000h
8093h	Receiver signal detect filter high to low init timer register (lane0_rx_sigdet_hl_init_tmr)	16	RW	03E8h
8094h	Receiver signal detect filter low to high filter timer register (lane0_rx_sigdet_lh_filt_tmr)	16	RW	0004h
8095h	Receiver signal detect filter low to high delay timer register (lane0_rx_sigdet_lh_dly_tmr)	16	RW	0004h
8096h	Receiver signal detect filter low to high min timer register (lane0_rx_sigdet_lh_min_tmr)	16	RW	0000h
8097h	Receiver signal detect filter low to high init timer register (lane0_rx_sigdet_lh_init_tmr)	16	RW	03E8h
8098h	Receiver LFPS detect filter filter timer register (lane0_rx_lfpsdet_filt_tmr)	16	RW	0008h
8099h	Receiver LFPS detect filter delay timer register (lane0_rx_lfpsdet_dly_tmr)	16	RW	000Bh
809Ah	Receiver LFPS detect min timer register (lane0_rx_lfpsdet_min_tmr)	16	RW	0013h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
809Bh	Receiver LFPS detect init timer register (lane0_rx_lfpsdet_init_tmr)	16	RW	0000h
80A0h	Eye surf control register (lane0_rx_eyesurf_ctrl)	16	RW	0000h
80A4h	Eye surf timer delay low register (lane0_rx_eyesurf_tmr_dellow)	16	RW	0000h
80A5h	Eye surf timer delay high register (lane0_rx_eyesurf_tmr_delhigh)	16	RW	0000h
80A6h	Eye surf timer test low register (lane0_rx_eyesurf_tmr_testlow)	16	RW	0000h
80A7h	Eye surf timer test high register (lane0_rx_eyesurf_tmr_testhigh)	16	RW	0000h
80A8h	Eye surf north south test point coordinate register (lane0_rx_eyesurf_ns_coord)	16	RW	0000h
80A9h	Eye surf east west test point coordinate register (lane0_rx_eyesurf_ew_coord)	16	RW	0000h
80AAh	Eye surf bit error count register (lane0_rx_eyesurf_errcnt)	16	RO	0000h
80B0h	Receiver BIST control register (lane0_rx_bist_ctrl)	16	RW	0000h
80B1h	Receiver BIST sync count register (lane0_rx_bist_synccnt)	16	RW	0000h
80B2h	Receiver BIST user defined data write register (lane0_rx_bist_uddwr)	16	WO	0000h
80B3h	Receiver BIST error count register (lane0_rx_bist_errcnt)	16	RO	0000h
80B4h	Clock frequency measurement control register (lane0_xcvr_cmsmt_clk_freq_msmt_ctrl)	16	RW	0000h
80B5h	Test clock selection register (lane0_xcvr_cmsmt_test_clk_sel)	16	RW	0000h
80B6h	Reference clock timer value register (lane0_xcvr_cmsmt_ref_clk_tmr_value)	16	RW	0000h
80B7h	Test clock counter value register (lane0_xcvr_cmsmt_test_clk_cnt_value)	16	RO	0000h
80E0h	RX sampler latch calibration control register (lane0_rx_slc_ctrl)	16	RW	0EFFh
80E1h	RX sampler latch calibration enable initialization timer value register (lane0_rx_slc_en_init_tmr)	16	RW	0004h
80E2h	RX sampler latch calibration unit initialization timer value register (lane0_rx_slc_cu_init_tmr)	16	RW	0010h
80E3h	RX sampler latch calibration unit iteration timer value register (lane0_rx_slc_cu_iter_tmr)	16	RW	0001h
80E4h	RX sampler latch calibration I even data mask register (lane0_rx_slc_ie_mask)	16	RW	03FFh
80E5h	RX sampler latch calibration I odd data mask register (lane0_rx_slc_io_mask)	16	RW	03FFh
80E6h	RX sampler latch calibration Q even data mask register (lane0_rx_slc_qe_mask)	16	RW	03FFh
80E7h	RX sampler latch calibration Q odd data mask register (lane0_rx_slc_qo_mask)	16	RW	03FFh
80E8h	RX sampler latch calibration E even data mask register (lane0_rx_slc_ee_mask)	16	RW	03FFh
80E9h	RX sampler latch calibration E odd data mask register (lane0_rx_slc_eo_mask)	16	RW	03FFh
80EAh	RX sampler latch calibration data threshold register (lane0_rx_slc_data_thr)	16	RW	0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8100h	RX sampler latch I odd positive 0 calibration unit control register (lane0_rx_slc_iop0_ctrl)	16	RW	0021h
8101h	RX sampler latch I odd positive 0 calibration unit override register (lane0_rx_slc_iop0_ovrd)	16	RW	0000h
8102h	RX sampler latch I odd positive 0 calibration unit start register (lane0_rx_slc_iop0_start)	16	RW	0021h
8103h	RX sampler latch I odd positive 0 calibration unit tune register (lane0_rx_slc_iop0_tune)	16	RW	0000h
8104h	RX sampler latch I odd positive 1 calibration unit control register (lane0_rx_slc_iop1_ctrl)	16	RW	001Fh
8105h	RX sampler latch I odd positive 1 calibration unit override register (lane0_rx_slc_iop1_ovrd)	16	RW	0000h
8106h	RX sampler latch I odd positive 1 calibration unit start register (lane0_rx_slc_iop1_start)	16	RW	801Fh
8107h	RX sampler latch I odd positive 1 calibration unit tune register (lane0_rx_slc_iop1_tune)	16	RW	0000h
8108h	RX sampler latch Q odd positive 0 calibration unit control register (lane0_rx_slc_qop0_ctrl)	16	RW	0021h
8109h	RX sampler latch Q odd positive 0 calibration unit override register (lane0_rx_slc_qop0_ovrd)	16	RW	0000h
810Ah	RX sampler latch Q odd positive 0 calibration unit start register (lane0_rx_slc_qop0_start)	16	RW	0021h
810Bh	RX sampler latch Q odd positive 0 calibration unit tune register (lane0_rx_slc_qop0_tune)	16	RW	0000h
810Ch	RX sampler latch Q odd positive 1 calibration unit control register (lane0_rx_slc_qop1_ctrl)	16	RW	001Fh
810Dh	RX sampler latch Q odd positive 1 calibration unit override register (lane0_rx_slc_qop1_ovrd)	16	RW	0000h
810Eh	RX sampler latch Q odd positive 1 calibration unit start register (lane0_rx_slc_qop1_start)	16	RW	801Fh
810Fh	RX sampler latch Q odd positive 1 calibration unit tune register (lane0_rx_slc_qop1_tune)	16	RW	0000h
8110h	RX sampler latch E odd positive 0 calibration unit control register (lane0_rx_slc_eop0_ctrl)	16	RW	0021h
8111h	RX sampler latch E odd positive 0 calibration unit override register (lane0_rx_slc_eop0_ovrd)	16	RW	0000h
8112h	RX sampler latch E odd positive 0 calibration unit start register (lane0_rx_slc_eop0_start)	16	RW	0021h
8113h	RX sampler latch E odd positive 0 calibration unit tune register (lane0_rx_slc_eop0_tune)	16	RW	0000h
8114h	RX sampler latch E odd positive 1 calibration unit control register (lane0_rx_slc_eop1_ctrl)	16	RW	001Fh
8115h	RX sampler latch E odd positive 1 calibration unit override register (lane0_rx_slc_eop1_ovrd)	16	RW	0000h
8116h	RX sampler latch E odd positive 1 calibration unit start register (lane0_rx_slc_eop1_start)	16	RW	801Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8117h	RX sampler latch E odd positive 1 calibration unit tune register (lane 0_rx_slc_eop1_tune)	16	RW	0000h
8118h	RX sampler latch I odd negative 0 calibration unit control register (lane0_rx_slc_ion0_ctrl)	16	RW	0021h
8119h	RX sampler latch I odd negative 0 calibration unit override register (lane0_rx_slc_ion0_ovrd)	16	RW	0000h
811Ah	RX sampler latch I odd negative 0 calibration unit start register (lane 0_rx_slc_ion0_start)	16	RW	0021h
811Bh	RX sampler latch I odd negative 0 calibration unit tune register (lane 0_rx_slc_ion0_tune)	16	RW	0000h
811Ch	RX sampler latch I odd negative 1 calibration unit control register (lane0_rx_slc_ion1_ctrl)	16	RW	001Fh
811Dh	RX sampler latch I odd negative 1 calibration unit override register (lane0_rx_slc_ion1_ovrd)	16	RW	0000h
811Eh	RX sampler latch I odd negative 1 calibration unit start register (lane 0_rx_slc_ion1_start)	16	RW	801Fh
811Fh	RX sampler latch I odd negative 1 calibration unit tune register (lane 0_rx_slc_ion1_tune)	16	RW	0000h
8120h	RX sampler latch Q odd negative 0 calibration unit control register (lane0_rx_slc_qon0_ctrl)	16	RW	0021h
8121h	RX sampler latch Q odd negative 0 calibration unit override register (lane0_rx_slc_qon0_ovrd)	16	RW	0000h
8122h	RX sampler latch Q odd negative 0 calibration unit start register (lane 0_rx_slc_qon0_start)	16	RW	0021h
8123h	RX sampler latch Q odd negative 0 calibration unit tune register (lane 0_rx_slc_qon0_tune)	16	RW	0000h
8124h	RX sampler latch Q odd negative 1 calibration unit control register (lane0_rx_slc_qon1_ctrl)	16	RW	001Fh
8125h	RX sampler latch Q odd negative 1 calibration unit override register (lane0_rx_slc_qon1_ovrd)	16	RW	0000h
8126h	RX sampler latch Q odd negative 1 calibration unit start register (lane 0_rx_slc_qon1_start)	16	RW	801Fh
8127h	RX sampler latch Q odd negative 1 calibration unit tune register (lane 0_rx_slc_qon1_tune)	16	RW	0000h
8128h	RX sampler latch E odd negative 0 calibration unit control register (lane0_rx_slc_eon0_ctrl)	16	RW	0021h
8129h	RX sampler latch E odd negative 0 calibration unit override register (lane0_rx_slc_eon0_ovrd)	16	RW	0000h
812Ah	RX sampler latch E odd negative 0 calibration unit start register (lane 0_rx_slc_eon0_start)	16	RW	0021h
812Bh	RX sampler latch E odd negative 0 calibration unit tune register (lane 0_rx_slc_eon0_tune)	16	RW	0000h
812Ch	RX sampler latch E odd negative 1 calibration unit control register (lane0_rx_slc_eon1_ctrl)	16	RW	001Fh
812Dh	RX sampler latch E odd negative 1 calibration unit override register (lane0_rx_slc_eon1_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
812Eh	RX sampler latch E odd negative 1 calibration unit start register (lane0_rx_slc_eon1_start)	16	RW	801Fh
812Fh	RX sampler latch E odd negative 1 calibration unit tune register (lane0_rx_slc_eon1_tune)	16	RW	0000h
8130h	RX sampler latch I even positive 0 calibration unit control register (lane0_rx_slc_iep0_ctrl)	16	RW	0021h
8131h	RX sampler latch I even positive 0 calibration unit override register (lane0_rx_slc_iep0_ovrd)	16	RW	0000h
8132h	RX sampler latch I even positive 0 calibration unit start register (lane0_rx_slc_iep0_start)	16	RW	0021h
8133h	RX sampler latch I even positive 0 calibration unit tune register (lane0_rx_slc_iep0_tune)	16	RW	0000h
8134h	RX sampler latch I even positive 1 calibration unit control register (lane0_rx_slc_iep1_ctrl)	16	RW	001Fh
8135h	RX sampler latch I even positive 1 calibration unit override register (lane0_rx_slc_iep1_ovrd)	16	RW	0000h
8136h	RX sampler latch I even positive 1 calibration unit start register (lane0_rx_slc_iep1_start)	16	RW	801Fh
8137h	RX sampler latch I even positive 1 calibration unit tune register (lane0_rx_slc_iep1_tune)	16	RW	0000h
8138h	RX sampler latch Q even positive 0 calibration unit control register (lane0_rx_slc_qep0_ctrl)	16	RW	0021h
8139h	RX sampler latch Q even positive 0 calibration unit override register (lane0_rx_slc_qep0_ovrd)	16	RW	0000h
813Ah	RX sampler latch Q even positive 0 calibration unit start register (lane0_rx_slc_qep0_start)	16	RW	0021h
813Bh	RX sampler latch Q even positive 0 calibration unit tune register (lane0_rx_slc_qep0_tune)	16	RW	0000h
813Ch	RX sampler latch Q even positive 1 calibration unit control register (lane0_rx_slc_qep1_ctrl)	16	RW	001Fh
813Dh	RX sampler latch Q even positive 1 calibration unit override register (lane0_rx_slc_qep1_ovrd)	16	RW	0000h
813Eh	RX sampler latch Q even positive 1 calibration unit start register (lane0_rx_slc_qep1_start)	16	RW	801Fh
813Fh	RX sampler latch Q even positive 1 calibration unit tune register (lane0_rx_slc_qep1_tune)	16	RW	0000h
8140h	RX sampler latch E even positive 0 calibration unit control register (lane0_rx_slc_eep0_ctrl)	16	RW	0021h
8141h	RX sampler latch E even positive 0 calibration unit override register (lane0_rx_slc_eep0_ovrd)	16	RW	0000h
8142h	RX sampler latch E even positive 0 calibration unit start register (lane0_rx_slc_eep0_start)	16	RW	0021h
8143h	RX sampler latch E even positive 0 calibration unit tune register (lane0_rx_slc_eep0_tune)	16	RW	0000h
8144h	RX sampler latch E even positive 1 calibration unit control register (lane0_rx_slc_eep1_ctrl)	16	RW	001Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8145h	RX sampler latch E even positive 1 calibration unit override register (lane0_rx_slc_eep1_ovrd)	16	RW	0000h
8146h	RX sampler latch E even positive 1 calibration unit start register (lane 0_rx_slc_eep1_start)	16	RW	801Fh
8147h	RX sampler latch E even positive 1 calibration unit tune register (lane 0_rx_slc_eep1_tune)	16	RW	0000h
8148h	RX sampler latch I even negative 0 calibration unit control register (lane0_rx_slc_ien0_ctrl)	16	RW	0021h
8149h	RX sampler latch I even negative 0 calibration unit override register (lane0_rx_slc_ien0_ovrd)	16	RW	0000h
814Ah	RX sampler latch I even negative 0 calibration unit start register (lane 0_rx_slc_ien0_start)	16	RW	0021h
814Bh	RX sampler latch I even negative 0 calibration unit tune register (lane 0_rx_slc_ien0_tune)	16	RW	0000h
814Ch	RX sampler latch I even negative 1 calibration unit control register (lane0_rx_slc_ien1_ctrl)	16	RW	001Fh
814Dh	RX sampler latch I even negative 1 calibration unit override register (lane0_rx_slc_ien1_ovrd)	16	RW	0000h
814Eh	RX sampler latch I even negative 1 calibration unit start register (lane 0_rx_slc_ien1_start)	16	RW	801Fh
814Fh	RX sampler latch I even negative 1 calibration unit tune register (lane 0_rx_slc_ien1_tune)	16	RW	0000h
8150h	RX sampler latch Q even negative 0 calibration unit control register (lane0_rx_slc_qen0_ctrl)	16	RW	0021h
8151h	RX sampler latch Q even negative 0 calibration unit override register (lane0_rx_slc_qen0_ovrd)	16	RW	0000h
8152h	RX sampler latch Q even negative 0 calibration unit start register (lane0_rx_slc_qen0_start)	16	RW	0021h
8153h	RX sampler latch Q even negative 0 calibration unit tune register (lane0_rx_slc_qen0_tune)	16	RW	0000h
8154h	RX sampler latch Q even negative 1 calibration unit control register (lane0_rx_slc_qen1_ctrl)	16	RW	001Fh
8155h	RX sampler latch Q even negative 1 calibration unit override register (lane0_rx_slc_qen1_ovrd)	16	RW	0000h
8156h	RX sampler latch Q even negative 1 calibration unit start register (lane0_rx_slc_qen1_start)	16	RW	801Fh
8157h	RX sampler latch Q even negative 1 calibration unit tune register (lane0_rx_slc_qen1_tune)	16	RW	0000h
8158h	RX sampler latch E even negative 0 calibration unit control register (lane0_rx_slc_een0_ctrl)	16	RW	0021h
8159h	RX sampler latch E even negative 0 calibration unit override register (lane0_rx_slc_een0_ovrd)	16	RW	0000h
815Ah	RX sampler latch E even negative 0 calibration unit start register (lane0_rx_slc_een0_start)	16	RW	0021h
815Bh	RX sampler latch E even negative 0 calibration unit tune register (lane0_rx_slc_een0_tune)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
815Ch	RX sampler latch E even negative 1 calibration unit control register (lane0_rx_slc_een1_ctrl)	16	RW	001Fh
815Dh	RX sampler latch E even negative 1 calibration unit override register (lane0_rx_slc_een1_ovrd)	16	RW	0000h
815Eh	RX sampler latch E even negative 1 calibration unit start register (lane0_rx_slc_een1_start)	16	RW	801Fh
815Fh	RX sampler latch E even negative 1 calibration unit tune register (lane0_rx_slc_een1_tune)	16	RW	0000h
8160h	REE USB 3 general control state machine control register (lane0_rx_ree_u3gcsm_ctrl)	16	RW	0001h
8161h	REE USB 3 general control state machine phase 1 equalization enable mask register (lane0_rx_ree_u3gcsm_eqenm_ph1)	16	RW	00C7h
8162h	REE USB 3 general control state machine phase 2 equalization enable mask register (lane0_rx_ree_u3gcsm_eqenm_ph2)	16	RW	00C7h
8163h	REE USB 3 general control state machine start timer value register (lane0_rx_ree_u3gcsm_start_tmr)	16	RW	0125h
8164h	REE USB 3 general control state machine run phase 1 timer value register (lane0_rx_ree_u3gcsm_run_ph1_tmr)	16	RW	07A2h
8165h	REE USB 3 general control state machine run phase 2 timer value register (lane0_rx_ree_u3gcsm_run_ph2_tmr)	16	RW	009Dh
8168h	REE PCIe Gen 2 general control state machine control register (lane0_rx_ree_g2gcsm_ctrl)	16	RW	0001h
8169h	REE PCIe Gen 2 general control state machine phase 1 equalization enable mask register (lane0_rx_ree_g2gcsm_eqenm_ph1)	16	RW	00C7h
816Ah	REE PCIe Gen 2 general control state machine phase 2 equalization enable mask register (lane0_rx_ree_g2gcsm_eqenm_ph2)	16	RW	00C7h
816Bh	REE PCIe Gen 2 general control state machine start timer value register (lane0_rx_ree_g2gcsm_start_tmr)	16	RW	0000h
816Ch	REE PCIe Gen 2 general control state machine run phase 1 timer value register (lane0_rx_ree_g2gcsm_run_ph1_tmr)	16	RW	0F43h
816Dh	REE PCIe Gen 2 general control state machine run phase 2 timer value register (lane0_rx_ree_g2gcsm_run_ph2_tmr)	16	RW	009Dh
8178h	REE periodic general control state machine control register (lane0_rx_ree_pergcsm_ctrl)	16	RW	0001h
8179h	REE periodic general control state machine phase 1 equalization enable mask register (lane0_rx_ree_pergcsm_eqenm_ph1)	16	RW	0080h
817Ah	REE periodic general control state machine phase 2 equalization enable mask register (lane0_rx_ree_pergcsm_eqenm_ph2)	16	RW	0080h
817Bh	REE periodic general control state machine start timer value register (lane0_rx_ree_pergcsm_start_tmr)	16	RW	0000h
817Ch	REE periodic general control state machine run phase 1 timer value register (lane0_rx_ree_pergcsm_run_ph1_tmr)	16	RW	FFFFh
817Dh	REE periodic general control state machine run phase 2 timer value register (lane0_rx_ree_pergcsm_run_ph2_tmr)	16	RW	009Dh
8180h	REE tap 1 control register (lane0_rx_ree_tap1_ctrl)	16	RW	0000h
8181h	REE tap 1 override register (lane0_rx_ree_tap1_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8182h	REE tap 1 diagnostics register (lane0_rx_ree_tap1_diag)	16	RW	0000h
8184h	REE tap 2 control register (lane0_rx_ree_tap2_ctrl)	16	RW	0400h
8185h	REE tap 2 override register (lane0_rx_ree_tap2_ovrd)	16	RW	0000h
8186h	REE tap 2 diagnostics register (lane0_rx_ree_tap2_diag)	16	RW	0000h
8188h	REE tap 3 control register (lane0_rx_ree_tap3_ctrl)	16	RW	0400h
8189h	REE tap 3 override register (lane0_rx_ree_tap3_ovrd)	16	RW	0000h
818Ah	REE tap 3 diagnostics register (lane0_rx_ree_tap3_diag)	16	RW	0000h
8194h	REE analog enable control state machine delay timer value register (lane0_rx_ree_anaensm_del_tmr)	16	RW	02EEh
8198h	REE peaking amp control register (lane0_rx_ree_peak_ctrl)	16	RW	0701h
8199h	REE peaking amp code control register (lane0_rx_ree_peak_code_ctrl)	16	RW	2A0Eh
819Ah	REE peaking amp upper threshold register (lane0_rx_ree_peak_uthr)	16	RW	0004h
819Bh	REE peaking amp lower threshold register (lane0_rx_ree_peak_lthr)	16	RW	0000h
819Ch	REE peaking amp input override register (lane0_rx_ree_peak_iovrd)	16	RW	0000h
819Dh	REE peaking amp code override register (lane0_rx_ree_peak_covrd)	16	RW	0000h
819Eh	REE peaking amp diagnostics register (lane0_rx_ree_peak_diag)	16	RW	0020h
81A0h	REE attenuation control register (lane0_rx_ree_atten_ctrl)	16	RW	0005h
81A1h	REE attenuation threshold register (lane0_rx_ree_atten_thr)	16	RW	0C02h
81A2h	REE attenuation counter register (lane0_rx_ree_atten_cnt)	16	RW	2100h
81A3h	REE attenuation override register (lane0_rx_ree_atten_ovrd)	16	RW	0000h
81A4h	REE attenuation diagnostics register (lane0_rx_ree_atten_diag)	16	RO	0000h
81A8h	REE low frequency equalizer control register (lane0_rx_ree_lfeq_ctrl)	16	RW	0000h
81A9h	REE low frequency equalizer override register (lane0_rx_ree_lfeq_ovrd)	16	RW	0000h
81AAh	REE low frequency equalizer diagnostics register (lane0_rx_ree_lfeq_diag)	16	RW	0000h
81ACh	REE VGA gain control register (lane0_rx_ree_vga_gain_ctrl)	16	RW	1001h
81ADh	REE VGA gain override register (lane0_rx_ree_vga_gain_ovrd)	16	RW	0000h
81AEh	REE VGA gain diagnostics register (lane0_rx_ree_vga_gain_diag)	16	RW	000Ah
81AFh	REE VGA gain target adjust diagnostics register (lane0_rx_ree_vga_gain_tgt_diag)	16	RO	0000h
81B0h	REE offset correction control register (lane0_rx_ree_off_cor_ctrl)	16	RW	0001h
81B1h	REE offset correction override register (lane0_rx_ree_off_cor_ovrd)	16	RW	0000h
81B2h	REE offset correction diagnostics register (lane0_rx_ree_off_cor_diag)	16	RW	0000h
81B8h	REE adder configuration register (lane0_rx_ree_addr_cfg)	16	RW	0001h
81B9h	REE tap 1 clip control register (lane0_rx_ree_tap1_clip)	16	RW	0519h
81BAh	REE taps 2 and 3 clip control register (lane0_rx_ree_tap2ton_clip)	16	RW	0519h
81BBh	REE control data mask register (lane0_rx_ree_ctrl_data_mask)	16	RW	4000h
81BCh	REE diagnostic control register (lane0_rx_ree_diag_ctrl)	16	RW	0040h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
81BDh	REE control state machine gen mode control register 1 (lane0_rx_ree_smgm_ctrl1)	16	RW	0702h
81BEh	REE control state machine gen mode control register 2 (lane0_rx_ree_smgm_ctrl2)	16	RW	0000h
81C0h	RX ILL diagnostic control register (lane0_rx_diag_ill_ctrl)	16	RW	0000h
81C1h	RX ILL IQ trim 0 register (lane0_rx_diag_ill_iq_trim0)	16	RW	533Fh
81C2h	RX ILL E trim 0 register (lane0_rx_diag_ill_e_trim0)	16	RW	533Fh
81C3h	RX ILL IQ trim 1 register (lane0_rx_diag_ill_iq_trim1)	16	RW	0002h
81C4h	RX ILL E trim 1 register (lane0_rx_diag_ill_e_trim1)	16	RW	0002h
81C5h	RX ILL IQ E trim 2 register (lane0_rx_diag_ill_iqe_trim2)	16	RW	3E00h
81C6h	RX ILL IQ E trim 3 register (lane0_rx_diag_ill_iqe_trim3)	16	RW	199Fh
81C7h	RX ILL IQ E trim 4 register (lane0_rx_diag_ill_iqe_trim4)	16	RW	0014h
81C8h	RX ILL IQ E trim 5 register (lane0_rx_diag_ill_iqe_trim5)	16	RW	1504h
81C9h	RX ILL IQ E trim 6 register (lane0_rx_diag_ill_iqe_trim6)	16	RW	0015h
81D0h	DFE amp fine tuning register (lane0_rx_diag_dfe_amp_tune)	16	RW	4CCCh
81D1h	DFE amp fine tuning 2 register (lane0_rx_diag_dfe_amp_tune_2)	16	RW	0C21h
81D2h	REE DAC control register (lane0_rx_diag_ree_dac_ctrl)	16	RW	0004h
81D3h	Receiver DFE control register 1 (lane0_rx_diag_dfe_ctrl1)	16	RW	0007h
81D4h	Receiver DFE control register 2 (lane0_rx_diag_dfe_ctrl2)	16	RW	003Eh
81D5h	Receiver DFE control register 3 (lane0_rx_diag_dfe_ctrl3)	16	RW	0ACAh
81D6h	Nyquist control register (lane0_rx_diag_nqst_ctrl)	16	RW	0998h
81D7h	Low frequency equalizer tuning register (lane0_rx_diag_lfeq_tune)	16	RW	0014h
81D8h	RX control register (lane0_rx_diag_rxctrl)	16	RW	0060h
81D9h	Receiver control reset diagnostic register (lane0_rx_diag_rst_diag)	16	RO	0000h
81DCh	RX signal detect tuning and control register (lane0_rx_diag_sigdet_tune)	16	RW	1005h
81DDh	RX LFPS detect tuning and control register (lane0_rx_diag_lfpsdet_tune)	16	RW	2410h
81DEh	Signal detect test register (lane0_rx_diag_sd_test)	16	RW	0000h
81E0h	RX sampler diagnostic control register (lane0_rx_diag_samp_ctrl)	16	RW	0001h
81E1h	RX Sampler CML TO CMOS enable delay register (lane0_rx_diag_sc2c_delay)	16	RW	012Ch
81E4h	MPHY control register 1 (lane0_rx_diag_mphy_ctrl_1)	16	RW	0000h
81E5h	MPHY control register 2 (lane0_rx_diag_mphy_ctrl_2)	16	RW	0000h
81E8h	RX loopback controller register (lane0_rx_diag_lpbk_ctrl)	16	RW	0000h
81E9h	RX extra enable control override register (lane0_rx_diag_ectrl_ovrd)	16	RW	0000h
81F0h	CML to CMOS bias trim register (lane0_rx_diag_cml2cmos_btrim)	16	RW	0000h
81F1h	RX bias gen control register 1 (lane0_rx_diag_bias_gen_ctrl1)	16	RW	0000h
81F2h	RX bias gen control register 2 (lane0_rx_diag_bias_gen_ctrl2)	16	RW	0000h
81F3h	RX bias gen control register 3 (lane0_rx_diag_bias_gen_ctrl3)	16	RW	0000h
81F4h	RX bias gen control register 4 (lane0_rx_diag_bias_gen_ctrl4)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
81F5h	RX boundary scan test mode register (lane0_rx_diag_bs_tm)	16	RW	0000h
81F6h	RX receiver front end test mode register 1 (lane0_rx_diag_rxfe_tm1)	16	RW	0000h
81F7h	RX receiver front end test mode register 2 (lane0_rx_diag_rxfe_tm2)	16	RW	0000h
81FEh	Receiver digital cover your alternatives register (lane0_rx_diag_dcya)	16	RW	0000h
81FFh	Receiver analog cover your alternatives register (lane0_rx_diag_ac ya)	16	RW	0000h
8400h	Receiver A0 power state definition register (lane1_rx_psc_a0)	16	RW	8BFDh
8401h	Receiver A1 power state definition register (lane1_rx_psc_a1)	16	RW	8BFDh
8402h	Receiver A2 power state definition register (lane1_rx_psc_a2)	16	RW	8910h
8403h	Receiver A3 power state definition register (lane1_rx_psc_a3)	16	RW	0000h
8404h	Receiver A4 power state definition register (lane1_rx_psc_a4)	16	RW	1000h
8405h	Receiver A5 power state definition register (lane1_rx_psc_a5)	16	RW	0000h
8406h	Receiver calibration power state definition register (lane1_rx_psc_cal)	16	RW	03FFh
8407h	Receiver ready power state definition register (lane1_rx_psc_rdy)	16	RW	0000h
8420h	RX IQ PI ILL calibration control register (lane1_rx_iqpi_ill_cal_ctrl)	16	RW	0018h
8421h	RX IQ PI ILL calibration start point register (lane1_rx_iqpi_ill_cal_st art)	16	RW	1018h
8422h	RX IQ PI ILL calibration timer control register (lane1_rx_iqpi_ill_cal_tc trl)	16	RW	0003h
8423h	RX IQ PI ILL calibration override register (lane1_rx_iqpi_ill_cal_ovrd)	16	RW	0000h
8424h	RX IQ PI ILL calibration initialization timer register (lane1_rx_iqpi_ill_ cal_init_tmr)	16	RW	02EEh
8425h	RX IQ PI ILL calibration iteration timer register (lane1_rx_iqpi_ill_cal_ iter_tmr)	16	RW	0080h
8426h	RX IQ PI ILL lock reference timer start value register (lane1_rx_iqpi_il l_lock_reftmr_start)	16	RW	01FFh
8428h	RX IQ PI ILL lock calibration counter start value standard mode 0 register (lane1_rx_iqpi_ill_lock_calcnt_start_0)	16	RW	007Fh
8429h	RX IQ PI ILL lock calibration counter start value standard mode 1 register (lane1_rx_iqpi_ill_lock_calcnt_start_1)	16	RW	00FFh
842Ah	RX IQ PI ILL lock calibration counter start value standard mode 2 register (lane1_rx_iqpi_ill_lock_calcnt_start_2)	16	RW	00FFh
842Bh	RX IQ PI ILL lock calibration counter start value standard mode 3 register (lane1_rx_iqpi_ill_lock_calcnt_start_3)	16	RW	0000h
8430h	RX E PI ILL calibration control register (lane1_rx_epi_ill_cal_ctrl)	16	RW	0018h
8431h	RX E PI ILL calibration start point register (lane1_rx_epi_ill_cal_start)	16	RW	1018h
8432h	RX E PI ILL calibration timer control register (lane1_rx_epi_ill_cal_tct rl)	16	RW	0003h
8433h	RX E PI ILL calibration override register (lane1_rx_epi_ill_cal_ovrd)	16	RW	0000h
8434h	RX E PI ILL calibration initialization timer register (lane1_rx_epi_ill_ cal_init_tmr)	16	RW	02EEh
8435h	RX E PI ILL calibration iteration timer register (lane1_rx_epi_ill_cal_ iter_tmr)	16	RW	0080h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8436h	RX E PI ILL lock reference timer start value register (lane1_rx_epi_ill_lock_reftmr_start)	16	RW	01FFh
8438h	RX E PI ILL lock calibration counter start value standard mode 0 register (lane1_rx_epi_ill_lock_calcnt_start_0)	16	RW	007Fh
8439h	RX E PI ILL lock calibration counter start value standard mode 1 register (lane1_rx_epi_ill_lock_calcnt_start_1)	16	RW	00FFh
843Ah	RX E PI ILL lock calibration counter start value standard mode 2 register (lane1_rx_epi_ill_lock_calcnt_start_2)	16	RW	00FFh
843Bh	RX E PI ILL lock calibration counter start value standard mode 3 register (lane1_rx_epi_ill_lock_calcnt_start_3)	16	RW	0000h
8440h	Signal detect calibration 0 control register (lane1_rx_sdcal0_ctrl)	16	RW	0000h
8441h	Signal detect calibration 0 override register (lane1_rx_sdcal0_ovrd)	16	RW	0000h
8442h	Signal detect calibration 0 start register (lane1_rx_sdcal0_start)	16	RW	0000h
8443h	Signal detect calibration 0 tune register (lane1_rx_sdcal0_tune)	16	RW	0000h
8444h	Signal detect calibration 0 initialization timer register (lane1_rx_sdcal0_init_tmr)	16	RW	0019h
8445h	Signal detect calibration 0 iteration timer register (lane1_rx_sdcal0_iter_tmr)	16	RW	007Dh
8448h	Signal detect calibration 1 control register (lane1_rx_sdcal1_ctrl)	16	RW	0000h
8449h	Signal detect calibration 1 override register (lane1_rx_sdcal1_ovrd)	16	RW	0000h
844Ah	Signal detect calibration 1 start register (lane1_rx_sdcal1_start)	16	RW	0000h
844Bh	Signal detect calibration 1 tune register (lane1_rx_sdcal1_tune)	16	RW	0000h
844Ch	Signal detect calibration 1 initialization timer register (lane1_rx_sdcal1_init_tmr)	16	RW	0019h
844Dh	Signal detect calibration 1 iteration timer register (lane1_rx_sdcal1_iter_tmr)	16	RW	007Dh
8458h	Sampler error DAC control register (lane1_rx_samp_dac_ctrl)	16	RW	0014h
8480h	CDRLF configuration register (lane1_rx_cdrlf_cnfg)	16	RW	20B3h
8481h	CDRLF configuration register 2 (lane1_rx_cdrlf_cnfg2)	16	RW	0000h
8482h	CDRLF margin diagnostic register (lane1_rx_cdrlf_mgn_diag)	16	RW	0000h
8483h	CDRLF fast phase lock timer value register 0 (lane1_rx_cdrlf_fpl_tmr0)	16	RW	0071h
8484h	CDRLF fast phase lock timer value register 1 (lane1_rx_cdrlf_fpl_tmr1)	16	RW	0731h
8485h	CDRLF fast frequency lock timer value register (lane1_rx_cdrlf_ffl_tmr)	16	RW	0018h
8488h	CDRLF fast frequency lock step 0 control register (lane1_rx_cdrlf_ffl0_ctrl)	16	RW	460Ch
8489h	CDRLF fast frequency lock step 1 control register (lane1_rx_cdrlf_ffl1_ctrl)	16	RW	4A0Ah
848Ah	CDRLF fast frequency lock step 2 control register (lane1_rx_cdrlf_ffl2_ctrl)	16	RW	5414h
848Bh	CDRLF fast frequency lock step 3 control register (lane1_rx_cdrlf_ffl3_ctrl)	16	RW	5E1Eh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
848Ch	CDRLF fast frequency lock step 4 control register (lane1_rx_cdrlf_ffl4_ctrl)	16	RW	9E1Eh
8490h	Receiver signal detect filter high to low filter timer register (lane1_rx_sigdet_hl_filt_tmr)	16	RW	0019h
8491h	Receiver signal detect filter high to low delay timer register (lane1_rx_sigdet_hl_dly_tmr)	16	RW	0000h
8492h	Receiver signal detect filter high to low min timer register (lane1_rx_sigdet_hl_min_tmr)	16	RW	0000h
8493h	Receiver signal detect filter high to low init timer register (lane1_rx_sigdet_hl_init_tmr)	16	RW	03E8h
8494h	Receiver signal detect filter low to high filter timer register (lane1_rx_sigdet_lh_filt_tmr)	16	RW	0004h
8495h	Receiver signal detect filter low to high delay timer register (lane1_rx_sigdet_lh_dly_tmr)	16	RW	0004h
8496h	Receiver signal detect filter low to high min timer register (lane1_rx_sigdet_lh_min_tmr)	16	RW	0000h
8497h	Receiver signal detect filter low to high init timer register (lane1_rx_sigdet_lh_init_tmr)	16	RW	03E8h
8498h	Receiver LFPS detect filter filter timer register (lane1_rx_lfpsdet_filt_tmr)	16	RW	0008h
8499h	Receiver LFPS detect filter delay timer register (lane1_rx_lfpsdet_dly_tmr)	16	RW	000Bh
849Ah	Receiver LFPS detect min timer register (lane1_rx_lfpsdet_min_tmr)	16	RW	0013h
849Bh	Receiver LFPS detect init timer register (lane1_rx_lfpsdet_init_tmr)	16	RW	0000h
84A0h	Eye surf control register (lane1_rx_eyesurf_ctrl)	16	RW	0000h
84A4h	Eye surf timer delay low register (lane1_rx_eyesurf_tmr_dellow)	16	RW	0000h
84A5h	Eye surf timer delay high register (lane1_rx_eyesurf_tmr_delhigh)	16	RW	0000h
84A6h	Eye surf timer test low register (lane1_rx_eyesurf_tmr_testlow)	16	RW	0000h
84A7h	Eye surf timer test high register (lane1_rx_eyesurf_tmr_testhigh)	16	RW	0000h
84A8h	Eye surf north south test point coordinate register (lane1_rx_eyesurf_ns_coord)	16	RW	0000h
84A9h	Eye surf east west test point coordinate register (lane1_rx_eyesurf_ew_coord)	16	RW	0000h
84AAh	Eye surf bit error count register (lane1_rx_eyesurf_errcnt)	16	RO	0000h
84B0h	Receiver BIST control register (lane1_rx_bist_ctrl)	16	RW	0000h
84B1h	Receiver BIST sync count register (lane1_rx_bist_synccnt)	16	RW	0000h
84B2h	Receiver BIST user defined data write register (lane1_rx_bist_uddwr)	16	WO	0000h
84B3h	Receiver BIST error count register (lane1_rx_bist_errcnt)	16	RO	0000h
84B4h	Clock frequency measurement control register (lane1_xcvr_cmsmt_clk_freq_msmt_ctrl)	16	RW	0000h
84B5h	Test clock selection register (lane1_xcvr_cmsmt_test_clk_sel)	16	RW	0000h
84B6h	Reference clock timer value register (lane1_xcvr_cmsmt_ref_clk_tmr_value)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
84B7h	Test clock counter value register (lane1_xcvr_cmsmt_test_clk_cnt_value)	16	RO	0000h
84E0h	RX sampler latch calibration control register (lane1_rx_slc_ctrl)	16	RW	0EFFh
84E1h	RX sampler latch calibration enable initialization timer value register (lane1_rx_slc_en_init_tmr)	16	RW	0004h
84E2h	RX sampler latch calibration unit initialization timer value register (lane1_rx_slc_cu_init_tmr)	16	RW	0010h
84E3h	RX sampler latch calibration unit iteration timer value register (lane1_rx_slc_cu_iter_tmr)	16	RW	0001h
84E4h	RX sampler latch calibration I even data mask register (lane1_rx_slc_ie_mask)	16	RW	03FFh
84E5h	RX sampler latch calibration I odd data mask register (lane1_rx_slc_io_mask)	16	RW	03FFh
84E6h	RX sampler latch calibration Q even data mask register (lane1_rx_slc_qe_mask)	16	RW	03FFh
84E7h	RX sampler latch calibration Q odd data mask register (lane1_rx_slc_qo_mask)	16	RW	03FFh
84E8h	RX sampler latch calibration E even data mask register (lane1_rx_slc_ee_mask)	16	RW	03FFh
84E9h	RX sampler latch calibration E odd data mask register (lane1_rx_slc_eo_mask)	16	RW	03FFh
84EAh	RX sampler latch calibration data threshold register (lane1_rx_slc_data_thr)	16	RW	0005h
8500h	RX sampler latch I odd positive 0 calibration unit control register (lane1_rx_slc_iop0_ctrl)	16	RW	0021h
8501h	RX sampler latch I odd positive 0 calibration unit override register (lane1_rx_slc_iop0_ovrd)	16	RW	0000h
8502h	RX sampler latch I odd positive 0 calibration unit start register (lane1_rx_slc_iop0_start)	16	RW	0021h
8503h	RX sampler latch I odd positive 0 calibration unit tune register (lane1_rx_slc_iop0_tune)	16	RW	0000h
8504h	RX sampler latch I odd positive 1 calibration unit control register (lane1_rx_slc_iop1_ctrl)	16	RW	001Fh
8505h	RX sampler latch I odd positive 1 calibration unit override register (lane1_rx_slc_iop1_ovrd)	16	RW	0000h
8506h	RX sampler latch I odd positive 1 calibration unit start register (lane1_rx_slc_iop1_start)	16	RW	801Fh
8507h	RX sampler latch I odd positive 1 calibration unit tune register (lane1_rx_slc_iop1_tune)	16	RW	0000h
8508h	RX sampler latch Q odd positive 0 calibration unit control register (lane1_rx_slc_qop0_ctrl)	16	RW	0021h
8509h	RX sampler latch Q odd positive 0 calibration unit override register (lane1_rx_slc_qop0_ovrd)	16	RW	0000h
850Ah	RX sampler latch Q odd positive 0 calibration unit start register (lane1_rx_slc_qop0_start)	16	RW	0021h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
850Bh	RX sampler latch Q odd positive 0 calibration unit tune register (lane 1_rx_slc_qop0_tune)	16	RW	0000h
850Ch	RX sampler latch Q odd positive 1 calibration unit control register (lane1_rx_slc_qop1_ctrl)	16	RW	001Fh
850Dh	RX sampler latch Q odd positive 1 calibration unit override register (lane1_rx_slc_qop1_ovrd)	16	RW	0000h
850Eh	RX sampler latch Q odd positive 1 calibration unit start register (lane 1_rx_slc_qop1_start)	16	RW	801Fh
850Fh	RX sampler latch Q odd positive 1 calibration unit tune register (lane 1_rx_slc_qop1_tune)	16	RW	0000h
8510h	RX sampler latch E odd positive 0 calibration unit control register (lane1_rx_slc_eop0_ctrl)	16	RW	0021h
8511h	RX sampler latch E odd positive 0 calibration unit override register (lane1_rx_slc_eop0_ovrd)	16	RW	0000h
8512h	RX sampler latch E odd positive 0 calibration unit start register (lane 1_rx_slc_eop0_start)	16	RW	0021h
8513h	RX sampler latch E odd positive 0 calibration unit tune register (lane 1_rx_slc_eop0_tune)	16	RW	0000h
8514h	RX sampler latch E odd positive 1 calibration unit control register (lane1_rx_slc_eop1_ctrl)	16	RW	001Fh
8515h	RX sampler latch E odd positive 1 calibration unit override register (lane1_rx_slc_eop1_ovrd)	16	RW	0000h
8516h	RX sampler latch E odd positive 1 calibration unit start register (lane 1_rx_slc_eop1_start)	16	RW	801Fh
8517h	RX sampler latch E odd positive 1 calibration unit tune register (lane 1_rx_slc_eop1_tune)	16	RW	0000h
8518h	RX sampler latch I odd negative 0 calibration unit control register (lane1_rx_slc_ion0_ctrl)	16	RW	0021h
8519h	RX sampler latch I odd negative 0 calibration unit override register (lane1_rx_slc_ion0_ovrd)	16	RW	0000h
851Ah	RX sampler latch I odd negative 0 calibration unit start register (lane 1_rx_slc_ion0_start)	16	RW	0021h
851Bh	RX sampler latch I odd negative 0 calibration unit tune register (lane 1_rx_slc_ion0_tune)	16	RW	0000h
851Ch	RX sampler latch I odd negative 1 calibration unit control register (lane1_rx_slc_ion1_ctrl)	16	RW	001Fh
851Dh	RX sampler latch I odd negative 1 calibration unit override register (lane1_rx_slc_ion1_ovrd)	16	RW	0000h
851Eh	RX sampler latch I odd negative 1 calibration unit start register (lane 1_rx_slc_ion1_start)	16	RW	801Fh
851Fh	RX sampler latch I odd negative 1 calibration unit tune register (lane 1_rx_slc_ion1_tune)	16	RW	0000h
8520h	RX sampler latch Q odd negative 0 calibration unit control register (lane1_rx_slc_qon0_ctrl)	16	RW	0021h
8521h	RX sampler latch Q odd negative 0 calibration unit override register (lane1_rx_slc_qon0_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8522h	RX sampler latch Q odd negative 0 calibration unit start register (lane 1_rx_slc_qon0_start)	16	RW	0021h
8523h	RX sampler latch Q odd negative 0 calibration unit tune register (lane 1_rx_slc_qon0_tune)	16	RW	0000h
8524h	RX sampler latch Q odd negative 1 calibration unit control register (lane1_rx_slc_qon1_ctrl)	16	RW	001Fh
8525h	RX sampler latch Q odd negative 1 calibration unit override register (lane1_rx_slc_qon1_ovrd)	16	RW	0000h
8526h	RX sampler latch Q odd negative 1 calibration unit start register (lane 1_rx_slc_qon1_start)	16	RW	801Fh
8527h	RX sampler latch Q odd negative 1 calibration unit tune register (lane 1_rx_slc_qon1_tune)	16	RW	0000h
8528h	RX sampler latch E odd negative 0 calibration unit control register (lane1_rx_slc_eon0_ctrl)	16	RW	0021h
8529h	RX sampler latch E odd negative 0 calibration unit override register (lane1_rx_slc_eon0_ovrd)	16	RW	0000h
852Ah	RX sampler latch E odd negative 0 calibration unit start register (lane 1_rx_slc_eon0_start)	16	RW	0021h
852Bh	RX sampler latch E odd negative 0 calibration unit tune register (lane 1_rx_slc_eon0_tune)	16	RW	0000h
852Ch	RX sampler latch E odd negative 1 calibration unit control register (lane1_rx_slc_eon1_ctrl)	16	RW	001Fh
852Dh	RX sampler latch E odd negative 1 calibration unit override register (lane1_rx_slc_eon1_ovrd)	16	RW	0000h
852Eh	RX sampler latch E odd negative 1 calibration unit start register (lane 1_rx_slc_eon1_start)	16	RW	801Fh
852Fh	RX sampler latch E odd negative 1 calibration unit tune register (lane 1_rx_slc_eon1_tune)	16	RW	0000h
8530h	RX sampler latch I even positive 0 calibration unit control register (lane1_rx_slc_iep0_ctrl)	16	RW	0021h
8531h	RX sampler latch I even positive 0 calibration unit override register (lane1_rx_slc_iep0_ovrd)	16	RW	0000h
8532h	RX sampler latch I even positive 0 calibration unit start register (lane 1_rx_slc_iep0_start)	16	RW	0021h
8533h	RX sampler latch I even positive 0 calibration unit tune register (lane 1_rx_slc_iep0_tune)	16	RW	0000h
8534h	RX sampler latch I even positive 1 calibration unit control register (lane1_rx_slc_iep1_ctrl)	16	RW	001Fh
8535h	RX sampler latch I even positive 1 calibration unit override register (lane1_rx_slc_iep1_ovrd)	16	RW	0000h
8536h	RX sampler latch I even positive 1 calibration unit start register (lane 1_rx_slc_iep1_start)	16	RW	801Fh
8537h	RX sampler latch I even positive 1 calibration unit tune register (lane 1_rx_slc_iep1_tune)	16	RW	0000h
8538h	RX sampler latch Q even positive 0 calibration unit control register (lane1_rx_slc_qep0_ctrl)	16	RW	0021h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8539h	RX sampler latch Q even positive 0 calibration unit override register (lane1_rx_slc_qep0_ovrd)	16	RW	0000h
853Ah	RX sampler latch Q even positive 0 calibration unit start register (lane 1_rx_slc_qep0_start)	16	RW	0021h
853Bh	RX sampler latch Q even positive 0 calibration unit tune register (lane 1_rx_slc_qep0_tune)	16	RW	0000h
853Ch	RX sampler latch Q even positive 1 calibration unit control register (lane1_rx_slc_qep1_ctrl)	16	RW	001Fh
853Dh	RX sampler latch Q even positive 1 calibration unit override register (lane1_rx_slc_qep1_ovrd)	16	RW	0000h
853Eh	RX sampler latch Q even positive 1 calibration unit start register (lane 1_rx_slc_qep1_start)	16	RW	801Fh
853Fh	RX sampler latch Q even positive 1 calibration unit tune register (lane 1_rx_slc_qep1_tune)	16	RW	0000h
8540h	RX sampler latch E even positive 0 calibration unit control register (lane1_rx_slc_eep0_ctrl)	16	RW	0021h
8541h	RX sampler latch E even positive 0 calibration unit override register (lane1_rx_slc_eep0_ovrd)	16	RW	0000h
8542h	RX sampler latch E even positive 0 calibration unit start register (lane 1_rx_slc_eep0_start)	16	RW	0021h
8543h	RX sampler latch E even positive 0 calibration unit tune register (lane 1_rx_slc_eep0_tune)	16	RW	0000h
8544h	RX sampler latch E even positive 1 calibration unit control register (lane1_rx_slc_eep1_ctrl)	16	RW	001Fh
8545h	RX sampler latch E even positive 1 calibration unit override register (lane1_rx_slc_eep1_ovrd)	16	RW	0000h
8546h	RX sampler latch E even positive 1 calibration unit start register (lane 1_rx_slc_eep1_start)	16	RW	801Fh
8547h	RX sampler latch E even positive 1 calibration unit tune register (lane 1_rx_slc_eep1_tune)	16	RW	0000h
8548h	RX sampler latch I even negative 0 calibration unit control register (lane1_rx_slc_ien0_ctrl)	16	RW	0021h
8549h	RX sampler latch I even negative 0 calibration unit override register (lane1_rx_slc_ien0_ovrd)	16	RW	0000h
854Ah	RX sampler latch I even negative 0 calibration unit start register (lane 1_rx_slc_ien0_start)	16	RW	0021h
854Bh	RX sampler latch I even negative 0 calibration unit tune register (lane 1_rx_slc_ien0_tune)	16	RW	0000h
854Ch	RX sampler latch I even negative 1 calibration unit control register (lane1_rx_slc_ien1_ctrl)	16	RW	001Fh
854Dh	RX sampler latch I even negative 1 calibration unit override register (lane1_rx_slc_ien1_ovrd)	16	RW	0000h
854Eh	RX sampler latch I even negative 1 calibration unit start register (lane 1_rx_slc_ien1_start)	16	RW	801Fh
854Fh	RX sampler latch I even negative 1 calibration unit tune register (lane 1_rx_slc_ien1_tune)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8550h	RX sampler latch Q even negative 0 calibration unit control register (lane1_rx_slc_qen0_ctrl)	16	RW	0021h
8551h	RX sampler latch Q even negative 0 calibration unit override register (lane1_rx_slc_qen0_ovrd)	16	RW	0000h
8552h	RX sampler latch Q even negative 0 calibration unit start register (lane1_rx_slc_qen0_start)	16	RW	0021h
8553h	RX sampler latch Q even negative 0 calibration unit tune register (lane1_rx_slc_qen0_tune)	16	RW	0000h
8554h	RX sampler latch Q even negative 1 calibration unit control register (lane1_rx_slc_qen1_ctrl)	16	RW	001Fh
8555h	RX sampler latch Q even negative 1 calibration unit override register (lane1_rx_slc_qen1_ovrd)	16	RW	0000h
8556h	RX sampler latch Q even negative 1 calibration unit start register (lane1_rx_slc_qen1_start)	16	RW	801Fh
8557h	RX sampler latch Q even negative 1 calibration unit tune register (lane1_rx_slc_qen1_tune)	16	RW	0000h
8558h	RX sampler latch E even negative 0 calibration unit control register (lane1_rx_slc_een0_ctrl)	16	RW	0021h
8559h	RX sampler latch E even negative 0 calibration unit override register (lane1_rx_slc_een0_ovrd)	16	RW	0000h
855Ah	RX sampler latch E even negative 0 calibration unit start register (lane1_rx_slc_een0_start)	16	RW	0021h
855Bh	RX sampler latch E even negative 0 calibration unit tune register (lane1_rx_slc_een0_tune)	16	RW	0000h
855Ch	RX sampler latch E even negative 1 calibration unit control register (lane1_rx_slc_een1_ctrl)	16	RW	001Fh
855Dh	RX sampler latch E even negative 1 calibration unit override register (lane1_rx_slc_een1_ovrd)	16	RW	0000h
855Eh	RX sampler latch E even negative 1 calibration unit start register (lane1_rx_slc_een1_start)	16	RW	801Fh
855Fh	RX sampler latch E even negative 1 calibration unit tune register (lane1_rx_slc_een1_tune)	16	RW	0000h
8560h	REE USB 3 general control state machine control register (lane1_rx_ree_u3gcsm_ctrl)	16	RW	0001h
8561h	REE USB 3 general control state machine phase 1 equalization enable mask register (lane1_rx_ree_u3gcsm_eqenm_ph1)	16	RW	00C7h
8562h	REE USB 3 general control state machine phase 2 equalization enable mask register (lane1_rx_ree_u3gcsm_eqenm_ph2)	16	RW	00C7h
8563h	REE USB 3 general control state machine start timer value register (lane1_rx_ree_u3gcsm_start_tmr)	16	RW	0125h
8564h	REE USB 3 general control state machine run phase 1 timer value register (lane1_rx_ree_u3gcsm_run_ph1_tmr)	16	RW	07A2h
8565h	REE USB 3 general control state machine run phase 2 timer value register (lane1_rx_ree_u3gcsm_run_ph2_tmr)	16	RW	009Dh
8568h	REE PCIe Gen 2 general control state machine control register (lane 1_rx_ree_g2gcsm_ctrl)	16	RW	0001h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8569h	REE PCIe Gen 2 general control state machine phase 1 equalization enable mask register (lane1_rx_ree_g2gcsm_eqenm_ph1)	16	RW	00C7h
856Ah	REE PCIe Gen 2 general control state machine phase 2 equalization enable mask register (lane1_rx_ree_g2gcsm_eqenm_ph2)	16	RW	00C7h
856Bh	REE PCIe Gen 2 general control state machine start timer value register (lane1_rx_ree_g2gcsm_start_tmr)	16	RW	0000h
856Ch	REE PCIe Gen 2 general control state machine run phase 1 timer value register (lane1_rx_ree_g2gcsm_run_ph1_tmr)	16	RW	0F43h
856Dh	REE PCIe Gen 2 general control state machine run phase 2 timer value register (lane1_rx_ree_g2gcsm_run_ph2_tmr)	16	RW	009Dh
8578h	REE periodic general control state machine control register (lane1_rx_ree_pergcsm_ctrl)	16	RW	0001h
8579h	REE periodic general control state machine phase 1 equalization enable mask register (lane1_rx_ree_pergcsm_eqenm_ph1)	16	RW	0080h
857Ah	REE periodic general control state machine phase 2 equalization enable mask register (lane1_rx_ree_pergcsm_eqenm_ph2)	16	RW	0080h
857Bh	REE periodic general control state machine start timer value register (lane1_rx_ree_pergcsm_start_tmr)	16	RW	0000h
857Ch	REE periodic general control state machine run phase 1 timer value register (lane1_rx_ree_pergcsm_run_ph1_tmr)	16	RW	FFFFh
857Dh	REE periodic general control state machine run phase 2 timer value register (lane1_rx_ree_pergcsm_run_ph2_tmr)	16	RW	009Dh
8580h	REE tap 1 control register (lane1_rx_ree_tap1_ctrl)	16	RW	0000h
8581h	REE tap 1 override register (lane1_rx_ree_tap1_ovrd)	16	RW	0000h
8582h	REE tap 1 diagnostics register (lane1_rx_ree_tap1_diag)	16	RW	0000h
8584h	REE tap 2 control register (lane1_rx_ree_tap2_ctrl)	16	RW	0400h
8585h	REE tap 2 override register (lane1_rx_ree_tap2_ovrd)	16	RW	0000h
8586h	REE tap 2 diagnostics register (lane1_rx_ree_tap2_diag)	16	RW	0000h
8588h	REE tap 3 control register (lane1_rx_ree_tap3_ctrl)	16	RW	0400h
8589h	REE tap 3 override register (lane1_rx_ree_tap3_ovrd)	16	RW	0000h
858Ah	REE tap 3 diagnostics register (lane1_rx_ree_tap3_diag)	16	RW	0000h
8594h	REE analog enable control state machine delay timer value register (lane1_rx_ree_anaensm_del_tmr)	16	RW	02EEh
8598h	REE peaking amp control register (lane1_rx_ree_peak_ctrl)	16	RW	0701h
8599h	REE peaking amp code control register (lane1_rx_ree_peak_code_ctrl)	16	RW	2A0Eh
859Ah	REE peaking amp upper threshold register (lane1_rx_ree_peak_uthr)	16	RW	0004h
859Bh	REE peaking amp lower threshold register (lane1_rx_ree_peak_lthr)	16	RW	0000h
859Ch	REE peaking amp input override register (lane1_rx_ree_peak_iovrd)	16	RW	0000h
859Dh	REE peaking amp code override register (lane1_rx_ree_peak_covrd)	16	RW	0000h
859Eh	REE peaking amp diagnostics register (lane1_rx_ree_peak_diag)	16	RW	0020h
85A0h	REE attenuation control register (lane1_rx_ree_atten_ctrl)	16	RW	0005h
85A1h	REE attenuation threshold register (lane1_rx_ree_atten_thr)	16	RW	0C02h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
85A2h	REE attenuation counter register (lane1_rx_ree_atten_cnt)	16	RW	2100h
85A3h	REE attenuation override register (lane1_rx_ree_atten_ovrd)	16	RW	0000h
85A4h	REE attenuation diagnostics register (lane1_rx_ree_atten_diag)	16	RO	0000h
85A8h	REE low frequency equalizer control register (lane1_rx_ree_lfeq_ctrl)	16	RW	0000h
85A9h	REE low frequency equalizer override register (lane1_rx_ree_lfeq_ovrd)	16	RW	0000h
85AAh	REE low frequency equalizer diagnostics register (lane1_rx_ree_lfeq_diag)	16	RW	0000h
85ACh	REE VGA gain control register (lane1_rx_ree_vga_gain_ctrl)	16	RW	1001h
85ADh	REE VGA gain override register (lane1_rx_ree_vga_gain_ovrd)	16	RW	0000h
85AEh	REE VGA gain diagnostics register (lane1_rx_ree_vga_gain_diag)	16	RW	000Ah
85AFh	REE VGA gain target adjust diagnostics register (lane1_rx_ree_vga_gain_tgt_diag)	16	RO	0000h
85B0h	REE offset correction control register (lane1_rx_ree_off_cor_ctrl)	16	RW	0001h
85B1h	REE offset correction override register (lane1_rx_ree_off_cor_ovrd)	16	RW	0000h
85B2h	REE offset correction diagnostics register (lane1_rx_ree_off_cor_diag)	16	RW	0000h
85B8h	REE adder configuration register (lane1_rx_ree_addr_cfg)	16	RW	0001h
85B9h	REE tap 1 clip control register (lane1_rx_ree_tap1_clip)	16	RW	0519h
85BAh	REE taps 2 and 3 clip control register (lane1_rx_ree_tap2ton_clip)	16	RW	0519h
85BBh	REE control data mask register (lane1_rx_ree_ctrl_data_mask)	16	RW	4000h
85BCh	REE diagnostic control register (lane1_rx_ree_diag_ctrl)	16	RW	0040h
85BDh	REE control state machine gen mode control register 1 (lane1_rx_ree_smgm_ctrl1)	16	RW	0702h
85BEh	REE control state machine gen mode control register 2 (lane1_rx_ree_smgm_ctrl2)	16	RW	0000h
85C0h	RX ILL diagnostic control register (lane1_rx_diag_ill_ctrl)	16	RW	0000h
85C1h	RX ILL IQ trim 0 register (lane1_rx_diag_ill_iq_trim0)	16	RW	533Fh
85C2h	RX ILL E trim 0 register (lane1_rx_diag_ill_e_trim0)	16	RW	533Fh
85C3h	RX ILL IQ trim 1 register (lane1_rx_diag_ill_iq_trim1)	16	RW	0002h
85C4h	RX ILL E trim 1 register (lane1_rx_diag_ill_e_trim1)	16	RW	0002h
85C5h	RX ILL IQ E trim 2 register (lane1_rx_diag_ill_iqe_trim2)	16	RW	3E00h
85C6h	RX ILL IQ E trim 3 register (lane1_rx_diag_ill_iqe_trim3)	16	RW	199Fh
85C7h	RX ILL IQ E trim 4 register (lane1_rx_diag_ill_iqe_trim4)	16	RW	0014h
85C8h	RX ILL IQ E trim 5 register (lane1_rx_diag_ill_iqe_trim5)	16	RW	1504h
85C9h	RX ILL IQ E trim 6 register (lane1_rx_diag_ill_iqe_trim6)	16	RW	0015h
85D0h	DFE amp fine tuning register (lane1_rx_diag_dfe_amp_tune)	16	RW	4CCCh
85D1h	DFE amp fine tuning 2 register (lane1_rx_diag_dfe_amp_tune_2)	16	RW	0C21h
85D2h	REE DAC control register (lane1_rx_diag_ree_dac_ctrl)	16	RW	0004h
85D3h	Receiver DFE control register 1 (lane1_rx_diag_dfe_ctrl1)	16	RW	0007h
85D4h	Receiver DFE control register 2 (lane1_rx_diag_dfe_ctrl2)	16	RW	003Eh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
85D5h	Receiver DFE control register 3 (lane1_rx_diag_dfe_ctrl3)	16	RW	0ACAh
85D6h	Nyquist control register (lane1_rx_diag_nqst_ctrl)	16	RW	0998h
85D7h	Low frequency equalizer tuning register (lane1_rx_diag_lfeq_tune)	16	RW	0014h
85D8h	RX control register (lane1_rx_diag_rxctrl)	16	RW	0060h
85D9h	Receiver control reset diagnostic register (lane1_rx_diag_rst_diag)	16	RO	0000h
85DCh	RX signal detect tuning and control register (lane1_rx_diag_sigdet_tune)	16	RW	1005h
85DDh	RX LFPS detect tuning and control register (lane1_rx_diag_lfpsdet_tune)	16	RW	2410h
85DEh	Signal detect test register (lane1_rx_diag_sd_test)	16	RW	0000h
85E0h	RX sampler diagnostic control register (lane1_rx_diag_samp_ctrl)	16	RW	0001h
85E1h	RX Sampler CML TO CMOS enable delay register (lane1_rx_diag_sc2c_delay)	16	RW	012Ch
85E4h	MPHY control register 1 (lane1_rx_diag_mphy_ctrl_1)	16	RW	0000h
85E5h	MPHY control register 2 (lane1_rx_diag_mphy_ctrl_2)	16	RW	0000h
85E8h	RX loopback controller register (lane1_rx_diag_lpbk_ctrl)	16	RW	0000h
85E9h	RX extra enable control override register (lane1_rx_diag_ectrl_ovrd)	16	RW	0000h
85F0h	CML to CMOS bias trim register (lane1_rx_diag_cml2cmos_btrim)	16	RW	0000h
85F1h	RX bias gen control register 1 (lane1_rx_diag_bias_gen_ctrl1)	16	RW	0000h
85F2h	RX bias gen control register 2 (lane1_rx_diag_bias_gen_ctrl2)	16	RW	0000h
85F3h	RX bias gen control register 3 (lane1_rx_diag_bias_gen_ctrl3)	16	RW	0000h
85F4h	RX bias gen control register 4 (lane1_rx_diag_bias_gen_ctrl4)	16	RW	0000h
85F5h	RX boundary scan test mode register (lane1_rx_diag_bs_tm)	16	RW	0000h
85F6h	RX receiver front end test mode register 1 (lane1_rx_diag_rxfe_tm1)	16	RW	0000h
85F7h	RX receiver front end test mode register 2 (lane1_rx_diag_rxfe_tm2)	16	RW	0000h
85FEh	Receiver digital cover your alternatives register (lane1_rx_diag_dcya)	16	RW	0000h
85FFh	Receiver analog cover your alternatives register (lane1_rx_diag_acya)	16	RW	0000h
8800h	Receiver A0 power state definition register (lane2_rx_psc_a0)	16	RW	8BFDh
8801h	Receiver A1 power state definition register (lane2_rx_psc_a1)	16	RW	8BFDh
8802h	Receiver A2 power state definition register (lane2_rx_psc_a2)	16	RW	8910h
8803h	Receiver A3 power state definition register (lane2_rx_psc_a3)	16	RW	0000h
8804h	Receiver A4 power state definition register (lane2_rx_psc_a4)	16	RW	1000h
8805h	Receiver A5 power state definition register (lane2_rx_psc_a5)	16	RW	0000h
8806h	Receiver calibration power state definition register (lane2_rx_psc_cal)	16	RW	03FFh
8807h	Receiver ready power state definition register (lane2_rx_psc_rdy)	16	RW	0000h
8820h	RX IQ PI ILL calibration control register (lane2_rx_iqpi_ill_cal_ctrl)	16	RW	0018h
8821h	RX IQ PI ILL calibration start point register (lane2_rx_iqpi_ill_cal_start)	16	RW	1018h
8822h	RX IQ PI ILL calibration timer control register (lane2_rx_iqpi_ill_cal_tctrl)	16	RW	0003h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8823h	RX IQ PI ILL calibration override register (lane2_rx_iqpi_ill_cal_ovrd)	16	RW	0000h
8824h	RX IQ PI ILL calibration initialization timer register (lane2_rx_iqpi_ill_cal_init_tmr)	16	RW	02EEh
8825h	RX IQ PI ILL calibration iteration timer register (lane2_rx_iqpi_ill_cal_iter_tmr)	16	RW	0080h
8826h	RX IQ PI ILL lock reference timer start value register (lane2_rx_iqpi_ill_lock_reftmr_start)	16	RW	01FFh
8828h	RX IQ PI ILL lock calibration counter start value standard mode 0 register (lane2_rx_iqpi_ill_lock_calcnt_start_0)	16	RW	007Fh
8829h	RX IQ PI ILL lock calibration counter start value standard mode 1 register (lane2_rx_iqpi_ill_lock_calcnt_start_1)	16	RW	00FFh
882Ah	RX IQ PI ILL lock calibration counter start value standard mode 2 register (lane2_rx_iqpi_ill_lock_calcnt_start_2)	16	RW	00FFh
882Bh	RX IQ PI ILL lock calibration counter start value standard mode 3 register (lane2_rx_iqpi_ill_lock_calcnt_start_3)	16	RW	0000h
8830h	RX E PI ILL calibration control register (lane2_rx_epi_ill_cal_ctrl)	16	RW	0018h
8831h	RX E PI ILL calibration start point register (lane2_rx_epi_ill_cal_start)	16	RW	1018h
8832h	RX E PI ILL calibration timer control register (lane2_rx_epi_ill_cal_tctrl)	16	RW	0003h
8833h	RX E PI ILL calibration override register (lane2_rx_epi_ill_cal_ovrd)	16	RW	0000h
8834h	RX E PI ILL calibration initialization timer register (lane2_rx_epi_ill_cal_init_tmr)	16	RW	02EEh
8835h	RX E PI ILL calibration iteration timer register (lane2_rx_epi_ill_cal_iter_tmr)	16	RW	0080h
8836h	RX E PI ILL lock reference timer start value register (lane2_rx_epi_ill_lock_reftmr_start)	16	RW	01FFh
8838h	RX E PI ILL lock calibration counter start value standard mode 0 register (lane2_rx_epi_ill_lock_calcnt_start_0)	16	RW	007Fh
8839h	RX E PI ILL lock calibration counter start value standard mode 1 register (lane2_rx_epi_ill_lock_calcnt_start_1)	16	RW	00FFh
883Ah	RX E PI ILL lock calibration counter start value standard mode 2 register (lane2_rx_epi_ill_lock_calcnt_start_2)	16	RW	00FFh
883Bh	RX E PI ILL lock calibration counter start value standard mode 3 register (lane2_rx_epi_ill_lock_calcnt_start_3)	16	RW	0000h
8840h	Signal detect calibration 0 control register (lane2_rx_sdcal0_ctrl)	16	RW	0000h
8841h	Signal detect calibration 0 override register (lane2_rx_sdcal0_ovrd)	16	RW	0000h
8842h	Signal detect calibration 0 start register (lane2_rx_sdcal0_start)	16	RW	0000h
8843h	Signal detect calibration 0 tune register (lane2_rx_sdcal0_tune)	16	RW	0000h
8844h	Signal detect calibration 0 initialization timer register (lane2_rx_sdcal0_init_tmr)	16	RW	0019h
8845h	Signal detect calibration 0 iteration timer register (lane2_rx_sdcal0_iter_tmr)	16	RW	007Dh
8848h	Signal detect calibration 1 control register (lane2_rx_sdcal1_ctrl)	16	RW	0000h
8849h	Signal detect calibration 1 override register (lane2_rx_sdcal1_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
884Ah	Signal detect calibration 1 start register (lane2_rx_sdcal1_start)	16	RW	0000h
884Bh	Signal detect calibration 1 tune register (lane2_rx_sdcal1_tune)	16	RW	0000h
884Ch	Signal detect calibration 1 initialization timer register (lane2_rx_sdcal1_init_tmr)	16	RW	0019h
884Dh	Signal detect calibration 1 iteration timer register (lane2_rx_sdcal1_iter_tmr)	16	RW	007Dh
8858h	Sampler error DAC control register (lane2_rx_samp_dac_ctrl)	16	RW	0014h
8880h	CDRLF configuration register (lane2_rx_cdrlf_cnfg)	16	RW	20B3h
8881h	CDRLF configuration register 2 (lane2_rx_cdrlf_cnfg2)	16	RW	0000h
8882h	CDRLF margin diagnostic register (lane2_rx_cdrlf_mgn_diag)	16	RW	0000h
8883h	CDRLF fast phase lock timer value register 0 (lane2_rx_cdrlf_fpl_tmr0)	16	RW	0071h
8884h	CDRLF fast phase lock timer value register 1 (lane2_rx_cdrlf_fpl_tmr1)	16	RW	0731h
8885h	CDRLF fast frequency lock timer value register (lane2_rx_cdrlf_ffl_tmr)	16	RW	0018h
8888h	CDRLF fast frequency lock step 0 control register (lane2_rx_cdrlf_ffl0_ctrl)	16	RW	460Ch
8889h	CDRLF fast frequency lock step 1 control register (lane2_rx_cdrlf_ffl1_ctrl)	16	RW	4A0Ah
888Ah	CDRLF fast frequency lock step 2 control register (lane2_rx_cdrlf_ffl2_ctrl)	16	RW	5414h
888Bh	CDRLF fast frequency lock step 3 control register (lane2_rx_cdrlf_ffl3_ctrl)	16	RW	5E1Eh
888Ch	CDRLF fast frequency lock step 4 control register (lane2_rx_cdrlf_ffl4_ctrl)	16	RW	9E1Eh
8890h	Receiver signal detect filter high to low filter timer register (lane2_rx_sigdet_hl_filt_tmr)	16	RW	0019h
8891h	Receiver signal detect filter high to low delay timer register (lane2_rx_sigdet_hl_dly_tmr)	16	RW	0000h
8892h	Receiver signal detect filter high to low min timer register (lane2_rx_sigdet_hl_min_tmr)	16	RW	0000h
8893h	Receiver signal detect filter high to low init timer register (lane2_rx_sigdet_hl_init_tmr)	16	RW	03E8h
8894h	Receiver signal detect filter low to high filter timer register (lane2_rx_sigdet_lh_filt_tmr)	16	RW	0004h
8895h	Receiver signal detect filter low to high delay timer register (lane2_rx_sigdet_lh_dly_tmr)	16	RW	0004h
8896h	Receiver signal detect filter low to high min timer register (lane2_rx_sigdet_lh_min_tmr)	16	RW	0000h
8897h	Receiver signal detect filter low to high init timer register (lane2_rx_sigdet_lh_init_tmr)	16	RW	03E8h
8898h	Receiver LFPS detect filter filter timer register (lane2_rx_lfpsdet_filt_tmr)	16	RW	0008h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8899h	Receiver LFPS detect filter delay timer register (lane2_rx_lfpsdet_dly_tmr)	16	RW	000Bh
889Ah	Receiver LFPS detect min timer register (lane2_rx_lfpsdet_min_tmr)	16	RW	0013h
889Bh	Receiver LFPS detect init timer register (lane2_rx_lfpsdet_init_tmr)	16	RW	0000h
88A0h	Eye surf control register (lane2_rx_eyesurf_ctrl)	16	RW	0000h
88A4h	Eye surf timer delay low register (lane2_rx_eyesurf_tmr_dellow)	16	RW	0000h
88A5h	Eye surf timer delay high register (lane2_rx_eyesurf_tmr_delhigh)	16	RW	0000h
88A6h	Eye surf timer test low register (lane2_rx_eyesurf_tmr_testlow)	16	RW	0000h
88A7h	Eye surf timer test high register (lane2_rx_eyesurf_tmr_testhigh)	16	RW	0000h
88A8h	Eye surf north south test point coordinate register (lane2_rx_eyesurf_ns_coord)	16	RW	0000h
88A9h	Eye surf east west test point coordinate register (lane2_rx_eyesurf_ew_coord)	16	RW	0000h
88AAh	Eye surf bit error count register (lane2_rx_eyesurf_errcnt)	16	RO	0000h
88B0h	Receiver BIST control register (lane2_rx_bist_ctrl)	16	RW	0000h
88B1h	Receiver BIST sync count register (lane2_rx_bist_synccnt)	16	RW	0000h
88B2h	Receiver BIST user defined data write register (lane2_rx_bist_uddwr)	16	WO	0000h
88B3h	Receiver BIST error count register (lane2_rx_bist_errcnt)	16	RO	0000h
88B4h	Clock frequency measurement control register (lane2_xcvr_cmsmt_clk_freq_msmt_ctrl)	16	RW	0000h
88B5h	Test clock selection register (lane2_xcvr_cmsmt_test_clk_sel)	16	RW	0000h
88B6h	Reference clock timer value register (lane2_xcvr_cmsmt_ref_clk_tmr_value)	16	RW	0000h
88B7h	Test clock counter value register (lane2_xcvr_cmsmt_test_clk_cnt_value)	16	RO	0000h
88E0h	RX sampler latch calibration control register (lane2_rx_slc_ctrl)	16	RW	0EFFh
88E1h	RX sampler latch calibration enable initialization timer value register (lane2_rx_slc_en_init_tmr)	16	RW	0004h
88E2h	RX sampler latch calibration unit initialization timer value register (lane2_rx_slc_cu_init_tmr)	16	RW	0010h
88E3h	RX sampler latch calibration unit iteration timer value register (lane2_rx_slc_cu_iter_tmr)	16	RW	0001h
88E4h	RX sampler latch calibration I even data mask register (lane2_rx_slc_ie_mask)	16	RW	03FFh
88E5h	RX sampler latch calibration I odd data mask register (lane2_rx_slc_io_mask)	16	RW	03FFh
88E6h	RX sampler latch calibration Q even data mask register (lane2_rx_slc_qe_mask)	16	RW	03FFh
88E7h	RX sampler latch calibration Q odd data mask register (lane2_rx_slc_qo_mask)	16	RW	03FFh
88E8h	RX sampler latch calibration E even data mask register (lane2_rx_slc_ee_mask)	16	RW	03FFh
88E9h	RX sampler latch calibration E odd data mask register (lane2_rx_slc_eo_mask)	16	RW	03FFh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
88EAh	RX sampler latch calibration data threshold register (lane2_rx_slc_data_thr)	16	RW	0005h
8900h	RX sampler latch I odd positive 0 calibration unit control register (lane2_rx_slc_iop0_ctrl)	16	RW	0021h
8901h	RX sampler latch I odd positive 0 calibration unit override register (lane2_rx_slc_iop0_ovrd)	16	RW	0000h
8902h	RX sampler latch I odd positive 0 calibration unit start register (lane2_rx_slc_iop0_start)	16	RW	0021h
8903h	RX sampler latch I odd positive 0 calibration unit tune register (lane2_rx_slc_iop0_tune)	16	RW	0000h
8904h	RX sampler latch I odd positive 1 calibration unit control register (lane2_rx_slc_iop1_ctrl)	16	RW	001Fh
8905h	RX sampler latch I odd positive 1 calibration unit override register (lane2_rx_slc_iop1_ovrd)	16	RW	0000h
8906h	RX sampler latch I odd positive 1 calibration unit start register (lane2_rx_slc_iop1_start)	16	RW	801Fh
8907h	RX sampler latch I odd positive 1 calibration unit tune register (lane2_rx_slc_iop1_tune)	16	RW	0000h
8908h	RX sampler latch Q odd positive 0 calibration unit control register (lane2_rx_slc_qop0_ctrl)	16	RW	0021h
8909h	RX sampler latch Q odd positive 0 calibration unit override register (lane2_rx_slc_qop0_ovrd)	16	RW	0000h
890Ah	RX sampler latch Q odd positive 0 calibration unit start register (lane2_rx_slc_qop0_start)	16	RW	0021h
890Bh	RX sampler latch Q odd positive 0 calibration unit tune register (lane2_rx_slc_qop0_tune)	16	RW	0000h
890Ch	RX sampler latch Q odd positive 1 calibration unit control register (lane2_rx_slc_qop1_ctrl)	16	RW	001Fh
890Dh	RX sampler latch Q odd positive 1 calibration unit override register (lane2_rx_slc_qop1_ovrd)	16	RW	0000h
890Eh	RX sampler latch Q odd positive 1 calibration unit start register (lane2_rx_slc_qop1_start)	16	RW	801Fh
890Fh	RX sampler latch Q odd positive 1 calibration unit tune register (lane2_rx_slc_qop1_tune)	16	RW	0000h
8910h	RX sampler latch E odd positive 0 calibration unit control register (lane2_rx_slc_eop0_ctrl)	16	RW	0021h
8911h	RX sampler latch E odd positive 0 calibration unit override register (lane2_rx_slc_eop0_ovrd)	16	RW	0000h
8912h	RX sampler latch E odd positive 0 calibration unit start register (lane2_rx_slc_eop0_start)	16	RW	0021h
8913h	RX sampler latch E odd positive 0 calibration unit tune register (lane2_rx_slc_eop0_tune)	16	RW	0000h
8914h	RX sampler latch E odd positive 1 calibration unit control register (lane2_rx_slc_eop1_ctrl)	16	RW	001Fh
8915h	RX sampler latch E odd positive 1 calibration unit override register (lane2_rx_slc_eop1_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8916h	RX sampler latch E odd positive 1 calibration unit start register (lane2_rx_slc_eop1_start)	16	RW	801Fh
8917h	RX sampler latch E odd positive 1 calibration unit tune register (lane2_rx_slc_eop1_tune)	16	RW	0000h
8918h	RX sampler latch I odd negative 0 calibration unit control register (lane2_rx_slc_ion0_ctrl)	16	RW	0021h
8919h	RX sampler latch I odd negative 0 calibration unit override register (lane2_rx_slc_ion0_ovrd)	16	RW	0000h
891Ah	RX sampler latch I odd negative 0 calibration unit start register (lane2_rx_slc_ion0_start)	16	RW	0021h
891Bh	RX sampler latch I odd negative 0 calibration unit tune register (lane2_rx_slc_ion0_tune)	16	RW	0000h
891Ch	RX sampler latch I odd negative 1 calibration unit control register (lane2_rx_slc_ion1_ctrl)	16	RW	001Fh
891Dh	RX sampler latch I odd negative 1 calibration unit override register (lane2_rx_slc_ion1_ovrd)	16	RW	0000h
891Eh	RX sampler latch I odd negative 1 calibration unit start register (lane2_rx_slc_ion1_start)	16	RW	801Fh
891Fh	RX sampler latch I odd negative 1 calibration unit tune register (lane2_rx_slc_ion1_tune)	16	RW	0000h
8920h	RX sampler latch Q odd negative 0 calibration unit control register (lane2_rx_slc_qon0_ctrl)	16	RW	0021h
8921h	RX sampler latch Q odd negative 0 calibration unit override register (lane2_rx_slc_qon0_ovrd)	16	RW	0000h
8922h	RX sampler latch Q odd negative 0 calibration unit start register (lane2_rx_slc_qon0_start)	16	RW	0021h
8923h	RX sampler latch Q odd negative 0 calibration unit tune register (lane2_rx_slc_qon0_tune)	16	RW	0000h
8924h	RX sampler latch Q odd negative 1 calibration unit control register (lane2_rx_slc_qon1_ctrl)	16	RW	001Fh
8925h	RX sampler latch Q odd negative 1 calibration unit override register (lane2_rx_slc_qon1_ovrd)	16	RW	0000h
8926h	RX sampler latch Q odd negative 1 calibration unit start register (lane2_rx_slc_qon1_start)	16	RW	801Fh
8927h	RX sampler latch Q odd negative 1 calibration unit tune register (lane2_rx_slc_qon1_tune)	16	RW	0000h
8928h	RX sampler latch E odd negative 0 calibration unit control register (lane2_rx_slc_eon0_ctrl)	16	RW	0021h
8929h	RX sampler latch E odd negative 0 calibration unit override register (lane2_rx_slc_eon0_ovrd)	16	RW	0000h
892Ah	RX sampler latch E odd negative 0 calibration unit start register (lane2_rx_slc_eon0_start)	16	RW	0021h
892Bh	RX sampler latch E odd negative 0 calibration unit tune register (lane2_rx_slc_eon0_tune)	16	RW	0000h
892Ch	RX sampler latch E odd negative 1 calibration unit control register (lane2_rx_slc_eon1_ctrl)	16	RW	001Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
892Dh	RX sampler latch E odd negative 1 calibration unit override register (lane2_rx_slc_eon1_ovrd)	16	RW	0000h
892Eh	RX sampler latch E odd negative 1 calibration unit start register (lane 2_rx_slc_eon1_start)	16	RW	801Fh
892Fh	RX sampler latch E odd negative 1 calibration unit tune register (lane 2_rx_slc_eon1_tune)	16	RW	0000h
8930h	RX sampler latch I even positive 0 calibration unit control register (lane2_rx_slc_iep0_ctrl)	16	RW	0021h
8931h	RX sampler latch I even positive 0 calibration unit override register (lane2_rx_slc_iep0_ovrd)	16	RW	0000h
8932h	RX sampler latch I even positive 0 calibration unit start register (lane 2_rx_slc_iep0_start)	16	RW	0021h
8933h	RX sampler latch I even positive 0 calibration unit tune register (lane 2_rx_slc_iep0_tune)	16	RW	0000h
8934h	RX sampler latch I even positive 1 calibration unit control register (lane2_rx_slc_iep1_ctrl)	16	RW	001Fh
8935h	RX sampler latch I even positive 1 calibration unit override register (lane2_rx_slc_iep1_ovrd)	16	RW	0000h
8936h	RX sampler latch I even positive 1 calibration unit start register (lane 2_rx_slc_iep1_start)	16	RW	801Fh
8937h	RX sampler latch I even positive 1 calibration unit tune register (lane 2_rx_slc_iep1_tune)	16	RW	0000h
8938h	RX sampler latch Q even positive 0 calibration unit control register (lane2_rx_slc_qep0_ctrl)	16	RW	0021h
8939h	RX sampler latch Q even positive 0 calibration unit override register (lane2_rx_slc_qep0_ovrd)	16	RW	0000h
893Ah	RX sampler latch Q even positive 0 calibration unit start register (lane 2_rx_slc_qep0_start)	16	RW	0021h
893Bh	RX sampler latch Q even positive 0 calibration unit tune register (lane 2_rx_slc_qep0_tune)	16	RW	0000h
893Ch	RX sampler latch Q even positive 1 calibration unit control register (lane2_rx_slc_qep1_ctrl)	16	RW	001Fh
893Dh	RX sampler latch Q even positive 1 calibration unit override register (lane2_rx_slc_qep1_ovrd)	16	RW	0000h
893Eh	RX sampler latch Q even positive 1 calibration unit start register (lane 2_rx_slc_qep1_start)	16	RW	801Fh
893Fh	RX sampler latch Q even positive 1 calibration unit tune register (lane 2_rx_slc_qep1_tune)	16	RW	0000h
8940h	RX sampler latch E even positive 0 calibration unit control register (lane2_rx_slc_eep0_ctrl)	16	RW	0021h
8941h	RX sampler latch E even positive 0 calibration unit override register (lane2_rx_slc_eep0_ovrd)	16	RW	0000h
8942h	RX sampler latch E even positive 0 calibration unit start register (lane 2_rx_slc_eep0_start)	16	RW	0021h
8943h	RX sampler latch E even positive 0 calibration unit tune register (lane 2_rx_slc_eep0_tune)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8944h	RX sampler latch E even positive 1 calibration unit control register (lane2_rx_slc_eep1_ctrl)	16	RW	001Fh
8945h	RX sampler latch E even positive 1 calibration unit override register (lane2_rx_slc_eep1_ovrd)	16	RW	0000h
8946h	RX sampler latch E even positive 1 calibration unit start register (lane 2_rx_slc_eep1_start)	16	RW	801Fh
8947h	RX sampler latch E even positive 1 calibration unit tune register (lane 2_rx_slc_eep1_tune)	16	RW	0000h
8948h	RX sampler latch I even negative 0 calibration unit control register (lane2_rx_slc_ien0_ctrl)	16	RW	0021h
8949h	RX sampler latch I even negative 0 calibration unit override register (lane2_rx_slc_ien0_ovrd)	16	RW	0000h
894Ah	RX sampler latch I even negative 0 calibration unit start register (lane 2_rx_slc_ien0_start)	16	RW	0021h
894Bh	RX sampler latch I even negative 0 calibration unit tune register (lane 2_rx_slc_ien0_tune)	16	RW	0000h
894Ch	RX sampler latch I even negative 1 calibration unit control register (lane2_rx_slc_ien1_ctrl)	16	RW	001Fh
894Dh	RX sampler latch I even negative 1 calibration unit override register (lane2_rx_slc_ien1_ovrd)	16	RW	0000h
894Eh	RX sampler latch I even negative 1 calibration unit start register (lane 2_rx_slc_ien1_start)	16	RW	801Fh
894Fh	RX sampler latch I even negative 1 calibration unit tune register (lane 2_rx_slc_ien1_tune)	16	RW	0000h
8950h	RX sampler latch Q even negative 0 calibration unit control register (lane2_rx_slc_qen0_ctrl)	16	RW	0021h
8951h	RX sampler latch Q even negative 0 calibration unit override register (lane2_rx_slc_qen0_ovrd)	16	RW	0000h
8952h	RX sampler latch Q even negative 0 calibration unit start register (lane2_rx_slc_qen0_start)	16	RW	0021h
8953h	RX sampler latch Q even negative 0 calibration unit tune register (lane2_rx_slc_qen0_tune)	16	RW	0000h
8954h	RX sampler latch Q even negative 1 calibration unit control register (lane2_rx_slc_qen1_ctrl)	16	RW	001Fh
8955h	RX sampler latch Q even negative 1 calibration unit override register (lane2_rx_slc_qen1_ovrd)	16	RW	0000h
8956h	RX sampler latch Q even negative 1 calibration unit start register (lane2_rx_slc_qen1_start)	16	RW	801Fh
8957h	RX sampler latch Q even negative 1 calibration unit tune register (lane2_rx_slc_qen1_tune)	16	RW	0000h
8958h	RX sampler latch E even negative 0 calibration unit control register (lane2_rx_slc_een0_ctrl)	16	RW	0021h
8959h	RX sampler latch E even negative 0 calibration unit override register (lane2_rx_slc_een0_ovrd)	16	RW	0000h
895Ah	RX sampler latch E even negative 0 calibration unit start register (lane2_rx_slc_een0_start)	16	RW	0021h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
895Bh	RX sampler latch E even negative 0 calibration unit tune register (lane2_rx_slc_een0_tune)	16	RW	0000h
895Ch	RX sampler latch E even negative 1 calibration unit control register (lane2_rx_slc_een1_ctrl)	16	RW	001Fh
895Dh	RX sampler latch E even negative 1 calibration unit override register (lane2_rx_slc_een1_ovrd)	16	RW	0000h
895Eh	RX sampler latch E even negative 1 calibration unit start register (lane2_rx_slc_een1_start)	16	RW	801Fh
895Fh	RX sampler latch E even negative 1 calibration unit tune register (lane2_rx_slc_een1_tune)	16	RW	0000h
8960h	REE USB 3 general control state machine control register (lane2_rx_ree_u3gcsm_ctrl)	16	RW	0001h
8961h	REE USB 3 general control state machine phase 1 equalization enable mask register (lane2_rx_ree_u3gcsm_eqenm_ph1)	16	RW	00C7h
8962h	REE USB 3 general control state machine phase 2 equalization enable mask register (lane2_rx_ree_u3gcsm_eqenm_ph2)	16	RW	00C7h
8963h	REE USB 3 general control state machine start timer value register (lane2_rx_ree_u3gcsm_start_tmr)	16	RW	0125h
8964h	REE USB 3 general control state machine run phase 1 timer value register (lane2_rx_ree_u3gcsm_run_ph1_tmr)	16	RW	07A2h
8965h	REE USB 3 general control state machine run phase 2 timer value register (lane2_rx_ree_u3gcsm_run_ph2_tmr)	16	RW	009Dh
8968h	REE PCIe Gen 2 general control state machine control register (lane2_rx_ree_g2gcsm_ctrl)	16	RW	0001h
8969h	REE PCIe Gen 2 general control state machine phase 1 equalization enable mask register (lane2_rx_ree_g2gcsm_eqenm_ph1)	16	RW	00C7h
896Ah	REE PCIe Gen 2 general control state machine phase 2 equalization enable mask register (lane2_rx_ree_g2gcsm_eqenm_ph2)	16	RW	00C7h
896Bh	REE PCIe Gen 2 general control state machine start timer value register (lane2_rx_ree_g2gcsm_start_tmr)	16	RW	0000h
896Ch	REE PCIe Gen 2 general control state machine run phase 1 timer value register (lane2_rx_ree_g2gcsm_run_ph1_tmr)	16	RW	0F43h
896Dh	REE PCIe Gen 2 general control state machine run phase 2 timer value register (lane2_rx_ree_g2gcsm_run_ph2_tmr)	16	RW	009Dh
8978h	REE periodic general control state machine control register (lane2_rx_ree_pergcsm_ctrl)	16	RW	0001h
8979h	REE periodic general control state machine phase 1 equalization enable mask register (lane2_rx_ree_pergcsm_eqenm_ph1)	16	RW	0080h
897Ah	REE periodic general control state machine phase 2 equalization enable mask register (lane2_rx_ree_pergcsm_eqenm_ph2)	16	RW	0080h
897Bh	REE periodic general control state machine start timer value register (lane2_rx_ree_pergcsm_start_tmr)	16	RW	0000h
897Ch	REE periodic general control state machine run phase 1 timer value register (lane2_rx_ree_pergcsm_run_ph1_tmr)	16	RW	FFFFh
897Dh	REE periodic general control state machine run phase 2 timer value register (lane2_rx_ree_pergcsm_run_ph2_tmr)	16	RW	009Dh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8980h	REE tap 1 control register (lane2_rx_ree_tap1_ctrl)	16	RW	0000h
8981h	REE tap 1 override register (lane2_rx_ree_tap1_ovrd)	16	RW	0000h
8982h	REE tap 1 diagnostics register (lane2_rx_ree_tap1_diag)	16	RW	0000h
8984h	REE tap 2 control register (lane2_rx_ree_tap2_ctrl)	16	RW	0400h
8985h	REE tap 2 override register (lane2_rx_ree_tap2_ovrd)	16	RW	0000h
8986h	REE tap 2 diagnostics register (lane2_rx_ree_tap2_diag)	16	RW	0000h
8988h	REE tap 3 control register (lane2_rx_ree_tap3_ctrl)	16	RW	0400h
8989h	REE tap 3 override register (lane2_rx_ree_tap3_ovrd)	16	RW	0000h
898Ah	REE tap 3 diagnostics register (lane2_rx_ree_tap3_diag)	16	RW	0000h
8994h	REE analog enable control state machine delay timer value register (lane2_rx_ree_anaensm_del_tmr)	16	RW	02EEh
8998h	REE peaking amp control register (lane2_rx_ree_peak_ctrl)	16	RW	0701h
8999h	REE peaking amp code control register (lane2_rx_ree_peak_code_ctrl)	16	RW	2A0Eh
899Ah	REE peaking amp upper threshold register (lane2_rx_ree_peak_uthr)	16	RW	0004h
899Bh	REE peaking amp lower threshold register (lane2_rx_ree_peak_lthr)	16	RW	0000h
899Ch	REE peaking amp input override register (lane2_rx_ree_peak_iovrd)	16	RW	0000h
899Dh	REE peaking amp code override register (lane2_rx_ree_peak_covrd)	16	RW	0000h
899Eh	REE peaking amp diagnostics register (lane2_rx_ree_peak_diag)	16	RW	0020h
89A0h	REE attenuation control register (lane2_rx_ree_atten_ctrl)	16	RW	0005h
89A1h	REE attenuation threshold register (lane2_rx_ree_atten_thr)	16	RW	0C02h
89A2h	REE attenuation counter register (lane2_rx_ree_atten_cnt)	16	RW	2100h
89A3h	REE attenuation override register (lane2_rx_ree_atten_ovrd)	16	RW	0000h
89A4h	REE attenuation diagnostics register (lane2_rx_ree_atten_diag)	16	RO	0000h
89A8h	REE low frequency equalizer control register (lane2_rx_ree_lfeq_ctrl)	16	RW	0000h
89A9h	REE low frequency equalizer override register (lane2_rx_ree_lfeq_ovrd)	16	RW	0000h
89AAh	REE low frequency equalizer diagnostics register (lane2_rx_ree_lfeq_diag)	16	RW	0000h
89ACh	REE VGA gain control register (lane2_rx_ree_vga_gain_ctrl)	16	RW	1001h
89ADh	REE VGA gain override register (lane2_rx_ree_vga_gain_ovrd)	16	RW	0000h
89AEh	REE VGA gain diagnostics register (lane2_rx_ree_vga_gain_diag)	16	RW	000Ah
89AFh	REE VGA gain target adjust diagnostics register (lane2_rx_ree_vga_gain_tgt_diag)	16	RO	0000h
89B0h	REE offset correction control register (lane2_rx_ree_off_cor_ctrl)	16	RW	0001h
89B1h	REE offset correction override register (lane2_rx_ree_off_cor_ovrd)	16	RW	0000h
89B2h	REE offset correction diagnostics register (lane2_rx_ree_off_cor_diag)	16	RW	0000h
89B8h	REE adder configuration register (lane2_rx_ree_addr_cfg)	16	RW	0001h
89B9h	REE tap 1 clip control register (lane2_rx_ree_tap1_clip)	16	RW	0519h
89BAh	REE taps 2 and 3 clip control register (lane2_rx_ree_tap2ton_clip)	16	RW	0519h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
89BBh	REE control data mask register (lane2_rx_ree_ctrl_data_mask)	16	RW	4000h
89BCh	REE diagnostic control register (lane2_rx_ree_diag_ctrl)	16	RW	0040h
89BDh	REE control state machine gen mode control register 1 (lane2_rx_ree_smgm_ctrl1)	16	RW	0702h
89BEh	REE control state machine gen mode control register 2 (lane2_rx_ree_smgm_ctrl2)	16	RW	0000h
89C0h	RX ILL diagnostic control register (lane2_rx_diag_ill_ctrl)	16	RW	0000h
89C1h	RX ILL IQ trim 0 register (lane2_rx_diag_ill_iq_trim0)	16	RW	533Fh
89C2h	RX ILL E trim 0 register (lane2_rx_diag_ill_e_trim0)	16	RW	533Fh
89C3h	RX ILL IQ trim 1 register (lane2_rx_diag_ill_iq_trim1)	16	RW	0002h
89C4h	RX ILL E trim 1 register (lane2_rx_diag_ill_e_trim1)	16	RW	0002h
89C5h	RX ILL IQ E trim 2 register (lane2_rx_diag_ill_iqe_trim2)	16	RW	3E00h
89C6h	RX ILL IQ E trim 3 register (lane2_rx_diag_ill_iqe_trim3)	16	RW	199Fh
89C7h	RX ILL IQ E trim 4 register (lane2_rx_diag_ill_iqe_trim4)	16	RW	0014h
89C8h	RX ILL IQ E trim 5 register (lane2_rx_diag_ill_iqe_trim5)	16	RW	1504h
89C9h	RX ILL IQ E trim 6 register (lane2_rx_diag_ill_iqe_trim6)	16	RW	0015h
89D0h	DFE amp fine tuning register (lane2_rx_diag_dfe_amp_tune)	16	RW	4CCCh
89D1h	DFE amp fine tuning 2 register (lane2_rx_diag_dfe_amp_tune_2)	16	RW	0C21h
89D2h	REE DAC control register (lane2_rx_diag_ree_dac_ctrl)	16	RW	0004h
89D3h	Receiver DFE control register 1 (lane2_rx_diag_dfe_ctrl1)	16	RW	0007h
89D4h	Receiver DFE control register 2 (lane2_rx_diag_dfe_ctrl2)	16	RW	003Eh
89D5h	Receiver DFE control register 3 (lane2_rx_diag_dfe_ctrl3)	16	RW	0ACAh
89D6h	Nyquist control register (lane2_rx_diag_nqst_ctrl)	16	RW	0998h
89D7h	Low frequency equalizer tuning register (lane2_rx_diag_lfeq_tune)	16	RW	0014h
89D8h	RX control register (lane2_rx_diag_rxctrl)	16	RW	0060h
89D9h	Receiver control reset diagnostic register (lane2_rx_diag_rst_diag)	16	RO	0000h
89DCh	RX signal detect tuning and control register (lane2_rx_diag_sigdet_tune)	16	RW	1005h
89DDh	RX LFPS detect tuning and control register (lane2_rx_diag_lfpsdet_tune)	16	RW	2410h
89DEh	Signal detect test register (lane2_rx_diag_sd_test)	16	RW	0000h
89E0h	RX sampler diagnostic control register (lane2_rx_diag_samp_ctrl)	16	RW	0001h
89E1h	RX Sampler CML TO CMOS enable delay register (lane2_rx_diag_sc2c_delay)	16	RW	012Ch
89E4h	MPHY control register 1 (lane2_rx_diag_mphy_ctrl_1)	16	RW	0000h
89E5h	MPHY control register 2 (lane2_rx_diag_mphy_ctrl_2)	16	RW	0000h
89E8h	RX loopback controller register (lane2_rx_diag_lpbk_ctrl)	16	RW	0000h
89E9h	RX extra enable control override register (lane2_rx_diag_ectrl_ovrd)	16	RW	0000h
89F0h	CML to CMOS bias trim register (lane2_rx_diag_cml2cmos_btrim)	16	RW	0000h
89F1h	RX bias gen control register 1 (lane2_rx_diag_bias_gen_ctrl1)	16	RW	0000h
89F2h	RX bias gen control register 2 (lane2_rx_diag_bias_gen_ctrl2)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
89F3h	RX bias gen control register 3 (lane2_rx_diag_bias_gen_ctrl3)	16	RW	0000h
89F4h	RX bias gen control register 4 (lane2_rx_diag_bias_gen_ctrl4)	16	RW	0000h
89F5h	RX boundary scan test mode register (lane2_rx_diag_bs_tm)	16	RW	0000h
89F6h	RX receiver front end test mode register 1 (lane2_rx_diag_rxfe_tm1)	16	RW	0000h
89F7h	RX receiver front end test mode register 2 (lane2_rx_diag_rxfe_tm2)	16	RW	0000h
89FEh	Receiver digital cover your alternatives register (lane2_rx_diag_dcya)	16	RW	0000h
89FFh	Receiver analog cover your alternatives register (lane2_rx_diag_ac ya)	16	RW	0000h
8C00h	Receiver A0 power state definition register (lane3_rx_psc_a0)	16	RW	8BFDh
8C01h	Receiver A1 power state definition register (lane3_rx_psc_a1)	16	RW	8BFDh
8C02h	Receiver A2 power state definition register (lane3_rx_psc_a2)	16	RW	8910h
8C03h	Receiver A3 power state definition register (lane3_rx_psc_a3)	16	RW	0000h
8C04h	Receiver A4 power state definition register (lane3_rx_psc_a4)	16	RW	1000h
8C05h	Receiver A5 power state definition register (lane3_rx_psc_a5)	16	RW	0000h
8C06h	Receiver calibration power state definition register (lane3_rx_psc_cal)	16	RW	03FFh
8C07h	Receiver ready power state definition register (lane3_rx_psc_rdy)	16	RW	0000h
8C20h	RX IQ PI ILL calibration control register (lane3_rx_iqpi_ill_cal_ctrl)	16	RW	0018h
8C21h	RX IQ PI ILL calibration start point register (lane3_rx_iqpi_ill_cal_st art)	16	RW	1018h
8C22h	RX IQ PI ILL calibration timer control register (lane3_rx_iqpi_ill_cal_tc trl)	16	RW	0003h
8C23h	RX IQ PI ILL calibration override register (lane3_rx_iqpi_ill_cal_ovrd)	16	RW	0000h
8C24h	RX IQ PI ILL calibration initialization timer register (lane3_rx_iqpi_ill_ cal_init_tmr)	16	RW	02EEh
8C25h	RX IQ PI ILL calibration iteration timer register (lane3_rx_iqpi_ill_cal_ iter_tmr)	16	RW	0080h
8C26h	RX IQ PI ILL lock reference timer start value register (lane3_rx_iqpi_il l_lock_reftmr_start)	16	RW	01FFh
8C28h	RX IQ PI ILL lock calibration counter start value standard mode 0 register (lane3_rx_iqpi_ill_lock_calcnt_start_0)	16	RW	007Fh
8C29h	RX IQ PI ILL lock calibration counter start value standard mode 1 register (lane3_rx_iqpi_ill_lock_calcnt_start_1)	16	RW	00FFh
8C2Ah	RX IQ PI ILL lock calibration counter start value standard mode 2 register (lane3_rx_iqpi_ill_lock_calcnt_start_2)	16	RW	00FFh
8C2Bh	RX IQ PI ILL lock calibration counter start value standard mode 3 register (lane3_rx_iqpi_ill_lock_calcnt_start_3)	16	RW	0000h
8C30h	RX E PI ILL calibration control register (lane3_rx_epi_ill_cal_ctrl)	16	RW	0018h
8C31h	RX E PI ILL calibration start point register (lane3_rx_epi_ill_cal_start)	16	RW	1018h
8C32h	RX E PI ILL calibration timer control register (lane3_rx_epi_ill_cal_tct rl)	16	RW	0003h
8C33h	RX E PI ILL calibration override register (lane3_rx_epi_ill_cal_ovrd)	16	RW	0000h
8C34h	RX E PI ILL calibration initialization timer register (lane3_rx_epi_ill_ cal_init_tmr)	16	RW	02EEh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8C35h	RX E PI ILL calibration iteration timer register (lane3_rx_epi_ill_cal_iter_tmr)	16	RW	0080h
8C36h	RX E PI ILL lock reference timer start value register (lane3_rx_epi_ill_lock_reftmr_start)	16	RW	01FFh
8C38h	RX E PI ILL lock calibration counter start value standard mode 0 register (lane3_rx_epi_ill_lock_calcnt_start_0)	16	RW	007Fh
8C39h	RX E PI ILL lock calibration counter start value standard mode 1 register (lane3_rx_epi_ill_lock_calcnt_start_1)	16	RW	00FFh
8C3Ah	RX E PI ILL lock calibration counter start value standard mode 2 register (lane3_rx_epi_ill_lock_calcnt_start_2)	16	RW	00FFh
8C3Bh	RX E PI ILL lock calibration counter start value standard mode 3 register (lane3_rx_epi_ill_lock_calcnt_start_3)	16	RW	0000h
8C40h	Signal detect calibration 0 control register (lane3_rx_sdcal0_ctrl)	16	RW	0000h
8C41h	Signal detect calibration 0 override register (lane3_rx_sdcal0_ovrd)	16	RW	0000h
8C42h	Signal detect calibration 0 start register (lane3_rx_sdcal0_start)	16	RW	0000h
8C43h	Signal detect calibration 0 tune register (lane3_rx_sdcal0_tune)	16	RW	0000h
8C44h	Signal detect calibration 0 initialization timer register (lane3_rx_sdcal0_init_tmr)	16	RW	0019h
8C45h	Signal detect calibration 0 iteration timer register (lane3_rx_sdcal0_iter_tmr)	16	RW	007Dh
8C48h	Signal detect calibration 1 control register (lane3_rx_sdcal1_ctrl)	16	RW	0000h
8C49h	Signal detect calibration 1 override register (lane3_rx_sdcal1_ovrd)	16	RW	0000h
8C4Ah	Signal detect calibration 1 start register (lane3_rx_sdcal1_start)	16	RW	0000h
8C4Bh	Signal detect calibration 1 tune register (lane3_rx_sdcal1_tune)	16	RW	0000h
8C4Ch	Signal detect calibration 1 initialization timer register (lane3_rx_sdcal1_init_tmr)	16	RW	0019h
8C4Dh	Signal detect calibration 1 iteration timer register (lane3_rx_sdcal1_iter_tmr)	16	RW	007Dh
8C58h	Sampler error DAC control register (lane3_rx_samp_dac_ctrl)	16	RW	0014h
8C80h	CDRLF configuration register (lane3_rx_cdrlf_cnfg)	16	RW	20B3h
8C81h	CDRLF configuration register 2 (lane3_rx_cdrlf_cnfg2)	16	RW	0000h
8C82h	CDRLF margin diagnostic register (lane3_rx_cdrlf_mgn_diag)	16	RW	0000h
8C83h	CDRLF fast phase lock timer value register 0 (lane3_rx_cdrlf_fpl_tmr0)	16	RW	0071h
8C84h	CDRLF fast phase lock timer value register 1 (lane3_rx_cdrlf_fpl_tmr1)	16	RW	0731h
8C85h	CDRLF fast frequency lock timer value register (lane3_rx_cdrlf_ffl_tmr)	16	RW	0018h
8C88h	CDRLF fast frequency lock step 0 control register (lane3_rx_cdrlf_ffl0_ctrl)	16	RW	460Ch
8C89h	CDRLF fast frequency lock step 1 control register (lane3_rx_cdrlf_ffl1_ctrl)	16	RW	4A0Ah
8C8Ah	CDRLF fast frequency lock step 2 control register (lane3_rx_cdrlf_ffl2_ctrl)	16	RW	5414h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8C8Bh	CDRLF fast frequency lock step 3 control register (lane3_rx_cdrlf_ffl3_ctrl)	16	RW	5E1Eh
8C8Ch	CDRLF fast frequency lock step 4 control register (lane3_rx_cdrlf_ffl4_ctrl)	16	RW	9E1Eh
8C90h	Receiver signal detect filter high to low filter timer register (lane3_rx_sigdet_hl_filt_tmr)	16	RW	0019h
8C91h	Receiver signal detect filter high to low delay timer register (lane3_rx_sigdet_hl_dly_tmr)	16	RW	0000h
8C92h	Receiver signal detect filter high to low min timer register (lane3_rx_sigdet_hl_min_tmr)	16	RW	0000h
8C93h	Receiver signal detect filter high to low init timer register (lane3_rx_sigdet_hl_init_tmr)	16	RW	03E8h
8C94h	Receiver signal detect filter low to high filter timer register (lane3_rx_sigdet_lh_filt_tmr)	16	RW	0004h
8C95h	Receiver signal detect filter low to high delay timer register (lane3_rx_sigdet_lh_dly_tmr)	16	RW	0004h
8C96h	Receiver signal detect filter low to high min timer register (lane3_rx_sigdet_lh_min_tmr)	16	RW	0000h
8C97h	Receiver signal detect filter low to high init timer register (lane3_rx_sigdet_lh_init_tmr)	16	RW	03E8h
8C98h	Receiver LFPS detect filter filter timer register (lane3_rx_lfpsdet_filt_tmr)	16	RW	0008h
8C99h	Receiver LFPS detect filter delay timer register (lane3_rx_lfpsdet_dly_tmr)	16	RW	000Bh
8C9Ah	Receiver LFPS detect min timer register (lane3_rx_lfpsdet_min_tmr)	16	RW	0013h
8C9Bh	Receiver LFPS detect init timer register (lane3_rx_lfpsdet_init_tmr)	16	RW	0000h
8CA0h	Eye surf control register (lane3_rx_eyesurf_ctrl)	16	RW	0000h
8CA4h	Eye surf timer delay low register (lane3_rx_eyesurf_tmr_dellow)	16	RW	0000h
8CA5h	Eye surf timer delay high register (lane3_rx_eyesurf_tmr_delhigh)	16	RW	0000h
8CA6h	Eye surf timer test low register (lane3_rx_eyesurf_tmr_testlow)	16	RW	0000h
8CA7h	Eye surf timer test high register (lane3_rx_eyesurf_tmr_testhigh)	16	RW	0000h
8CA8h	Eye surf north south test point coordinate register (lane3_rx_eyesurf_ns_coord)	16	RW	0000h
8CA9h	Eye surf east west test point coordinate register (lane3_rx_eyesurf_ew_coord)	16	RW	0000h
8CAAh	Eye surf bit error count register (lane3_rx_eyesurf_errcnt)	16	RO	0000h
8CB0h	Receiver BIST control register (lane3_rx_bist_ctrl)	16	RW	0000h
8CB1h	Receiver BIST sync count register (lane3_rx_bist_synccnt)	16	RW	0000h
8CB2h	Receiver BIST user defined data write register (lane3_rx_bist_uddwr)	16	WO	0000h
8CB3h	Receiver BIST error count register (lane3_rx_bist_errcnt)	16	RO	0000h
8CB4h	Clock frequency measurement control register (lane3_xcvr_cmsmt_clk_freq_msmt_ctrl)	16	RW	0000h
8CB5h	Test clock selection register (lane3_xcvr_cmsmt_test_clk_sel)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8CB6h	Reference clock timer value register (lane3_xcvr_cmsmt_ref_clk_tmr_value)	16	RW	0000h
8CB7h	Test clock counter value register (lane3_xcvr_cmsmt_test_clk_cnt_value)	16	RO	0000h
8CE0h	RX sampler latch calibration control register (lane3_rx_slc_ctrl)	16	RW	0EFFh
8CE1h	RX sampler latch calibration enable initialization timer value register (lane3_rx_slc_en_init_tmr)	16	RW	0004h
8CE2h	RX sampler latch calibration unit initialization timer value register (lane3_rx_slc_cu_init_tmr)	16	RW	0010h
8CE3h	RX sampler latch calibration unit iteration timer value register (lane3_rx_slc_cu_iter_tmr)	16	RW	0001h
8CE4h	RX sampler latch calibration I even data mask register (lane3_rx_slc_ie_mask)	16	RW	03FFh
8CE5h	RX sampler latch calibration I odd data mask register (lane3_rx_slc_io_mask)	16	RW	03FFh
8CE6h	RX sampler latch calibration Q even data mask register (lane3_rx_slc_qe_mask)	16	RW	03FFh
8CE7h	RX sampler latch calibration Q odd data mask register (lane3_rx_slc_qo_mask)	16	RW	03FFh
8CE8h	RX sampler latch calibration E even data mask register (lane3_rx_slc_ee_mask)	16	RW	03FFh
8CE9h	RX sampler latch calibration E odd data mask register (lane3_rx_slc_eo_mask)	16	RW	03FFh
8CEAh	RX sampler latch calibration data threshold register (lane3_rx_slc_data_thr)	16	RW	0005h
8D00h	RX sampler latch I odd positive 0 calibration unit control register (lane3_rx_slc_iop0_ctrl)	16	RW	0021h
8D01h	RX sampler latch I odd positive 0 calibration unit override register (lane3_rx_slc_iop0_ovrd)	16	RW	0000h
8D02h	RX sampler latch I odd positive 0 calibration unit start register (lane3_rx_slc_iop0_start)	16	RW	0021h
8D03h	RX sampler latch I odd positive 0 calibration unit tune register (lane3_rx_slc_iop0_tune)	16	RW	0000h
8D04h	RX sampler latch I odd positive 1 calibration unit control register (lane3_rx_slc_iop1_ctrl)	16	RW	001Fh
8D05h	RX sampler latch I odd positive 1 calibration unit override register (lane3_rx_slc_iop1_ovrd)	16	RW	0000h
8D06h	RX sampler latch I odd positive 1 calibration unit start register (lane3_rx_slc_iop1_start)	16	RW	801Fh
8D07h	RX sampler latch I odd positive 1 calibration unit tune register (lane3_rx_slc_iop1_tune)	16	RW	0000h
8D08h	RX sampler latch Q odd positive 0 calibration unit control register (lane3_rx_slc_qop0_ctrl)	16	RW	0021h
8D09h	RX sampler latch Q odd positive 0 calibration unit override register (lane3_rx_slc_qop0_ovrd)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8D0Ah	RX sampler latch Q odd positive 0 calibration unit start register (lane3_rx_slc_qop0_start)	16	RW	0021h
8D0Bh	RX sampler latch Q odd positive 0 calibration unit tune register (lane3_rx_slc_qop0_tune)	16	RW	0000h
8D0Ch	RX sampler latch Q odd positive 1 calibration unit control register (lane3_rx_slc_qop1_ctrl)	16	RW	001Fh
8D0Dh	RX sampler latch Q odd positive 1 calibration unit override register (lane3_rx_slc_qop1_ovrd)	16	RW	0000h
8D0Eh	RX sampler latch Q odd positive 1 calibration unit start register (lane3_rx_slc_qop1_start)	16	RW	801Fh
8D0Fh	RX sampler latch Q odd positive 1 calibration unit tune register (lane3_rx_slc_qop1_tune)	16	RW	0000h
8D10h	RX sampler latch E odd positive 0 calibration unit control register (lane3_rx_slc_eop0_ctrl)	16	RW	0021h
8D11h	RX sampler latch E odd positive 0 calibration unit override register (lane3_rx_slc_eop0_ovrd)	16	RW	0000h
8D12h	RX sampler latch E odd positive 0 calibration unit start register (lane3_rx_slc_eop0_start)	16	RW	0021h
8D13h	RX sampler latch E odd positive 0 calibration unit tune register (lane3_rx_slc_eop0_tune)	16	RW	0000h
8D14h	RX sampler latch E odd positive 1 calibration unit control register (lane3_rx_slc_eop1_ctrl)	16	RW	001Fh
8D15h	RX sampler latch E odd positive 1 calibration unit override register (lane3_rx_slc_eop1_ovrd)	16	RW	0000h
8D16h	RX sampler latch E odd positive 1 calibration unit start register (lane3_rx_slc_eop1_start)	16	RW	801Fh
8D17h	RX sampler latch E odd positive 1 calibration unit tune register (lane3_rx_slc_eop1_tune)	16	RW	0000h
8D18h	RX sampler latch I odd negative 0 calibration unit control register (lane3_rx_slc_ion0_ctrl)	16	RW	0021h
8D19h	RX sampler latch I odd negative 0 calibration unit override register (lane3_rx_slc_ion0_ovrd)	16	RW	0000h
8D1Ah	RX sampler latch I odd negative 0 calibration unit start register (lane3_rx_slc_ion0_start)	16	RW	0021h
8D1Bh	RX sampler latch I odd negative 0 calibration unit tune register (lane3_rx_slc_ion0_tune)	16	RW	0000h
8D1Ch	RX sampler latch I odd negative 1 calibration unit control register (lane3_rx_slc_ion1_ctrl)	16	RW	001Fh
8D1Dh	RX sampler latch I odd negative 1 calibration unit override register (lane3_rx_slc_ion1_ovrd)	16	RW	0000h
8D1Eh	RX sampler latch I odd negative 1 calibration unit start register (lane3_rx_slc_ion1_start)	16	RW	801Fh
8D1Fh	RX sampler latch I odd negative 1 calibration unit tune register (lane3_rx_slc_ion1_tune)	16	RW	0000h
8D20h	RX sampler latch Q odd negative 0 calibration unit control register (lane3_rx_slc_qon0_ctrl)	16	RW	0021h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8D21h	RX sampler latch Q odd negative 0 calibration unit override register (lane3_rx_slc_qon0_ovrd)	16	RW	0000h
8D22h	RX sampler latch Q odd negative 0 calibration unit start register (lane 3_rx_slc_qon0_start)	16	RW	0021h
8D23h	RX sampler latch Q odd negative 0 calibration unit tune register (lane 3_rx_slc_qon0_tune)	16	RW	0000h
8D24h	RX sampler latch Q odd negative 1 calibration unit control register (lane3_rx_slc_qon1_ctrl)	16	RW	001Fh
8D25h	RX sampler latch Q odd negative 1 calibration unit override register (lane3_rx_slc_qon1_ovrd)	16	RW	0000h
8D26h	RX sampler latch Q odd negative 1 calibration unit start register (lane 3_rx_slc_qon1_start)	16	RW	801Fh
8D27h	RX sampler latch Q odd negative 1 calibration unit tune register (lane 3_rx_slc_qon1_tune)	16	RW	0000h
8D28h	RX sampler latch E odd negative 0 calibration unit control register (lane3_rx_slc_eon0_ctrl)	16	RW	0021h
8D29h	RX sampler latch E odd negative 0 calibration unit override register (lane3_rx_slc_eon0_ovrd)	16	RW	0000h
8D2Ah	RX sampler latch E odd negative 0 calibration unit start register (lane 3_rx_slc_eon0_start)	16	RW	0021h
8D2Bh	RX sampler latch E odd negative 0 calibration unit tune register (lane 3_rx_slc_eon0_tune)	16	RW	0000h
8D2Ch	RX sampler latch E odd negative 1 calibration unit control register (lane3_rx_slc_eon1_ctrl)	16	RW	001Fh
8D2Dh	RX sampler latch E odd negative 1 calibration unit override register (lane3_rx_slc_eon1_ovrd)	16	RW	0000h
8D2Eh	RX sampler latch E odd negative 1 calibration unit start register (lane 3_rx_slc_eon1_start)	16	RW	801Fh
8D2Fh	RX sampler latch E odd negative 1 calibration unit tune register (lane 3_rx_slc_eon1_tune)	16	RW	0000h
8D30h	RX sampler latch I even positive 0 calibration unit control register (lane3_rx_slc_iep0_ctrl)	16	RW	0021h
8D31h	RX sampler latch I even positive 0 calibration unit override register (lane3_rx_slc_iep0_ovrd)	16	RW	0000h
8D32h	RX sampler latch I even positive 0 calibration unit start register (lane 3_rx_slc_iep0_start)	16	RW	0021h
8D33h	RX sampler latch I even positive 0 calibration unit tune register (lane 3_rx_slc_iep0_tune)	16	RW	0000h
8D34h	RX sampler latch I even positive 1 calibration unit control register (lane3_rx_slc_iep1_ctrl)	16	RW	001Fh
8D35h	RX sampler latch I even positive 1 calibration unit override register (lane3_rx_slc_iep1_ovrd)	16	RW	0000h
8D36h	RX sampler latch I even positive 1 calibration unit start register (lane 3_rx_slc_iep1_start)	16	RW	801Fh
8D37h	RX sampler latch I even positive 1 calibration unit tune register (lane 3_rx_slc_iep1_tune)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8D38h	RX sampler latch Q even positive 0 calibration unit control register (lane3_rx_slc_qep0_ctrl)	16	RW	0021h
8D39h	RX sampler latch Q even positive 0 calibration unit override register (lane3_rx_slc_qep0_ovrd)	16	RW	0000h
8D3Ah	RX sampler latch Q even positive 0 calibration unit start register (lane 3_rx_slc_qep0_start)	16	RW	0021h
8D3Bh	RX sampler latch Q even positive 0 calibration unit tune register (lane 3_rx_slc_qep0_tune)	16	RW	0000h
8D3Ch	RX sampler latch Q even positive 1 calibration unit control register (lane3_rx_slc_qep1_ctrl)	16	RW	001Fh
8D3Dh	RX sampler latch Q even positive 1 calibration unit override register (lane3_rx_slc_qep1_ovrd)	16	RW	0000h
8D3Eh	RX sampler latch Q even positive 1 calibration unit start register (lane 3_rx_slc_qep1_start)	16	RW	801Fh
8D3Fh	RX sampler latch Q even positive 1 calibration unit tune register (lane 3_rx_slc_qep1_tune)	16	RW	0000h
8D40h	RX sampler latch E even positive 0 calibration unit control register (lane3_rx_slc_eep0_ctrl)	16	RW	0021h
8D41h	RX sampler latch E even positive 0 calibration unit override register (lane3_rx_slc_eep0_ovrd)	16	RW	0000h
8D42h	RX sampler latch E even positive 0 calibration unit start register (lane 3_rx_slc_eep0_start)	16	RW	0021h
8D43h	RX sampler latch E even positive 0 calibration unit tune register (lane 3_rx_slc_eep0_tune)	16	RW	0000h
8D44h	RX sampler latch E even positive 1 calibration unit control register (lane3_rx_slc_eep1_ctrl)	16	RW	001Fh
8D45h	RX sampler latch E even positive 1 calibration unit override register (lane3_rx_slc_eep1_ovrd)	16	RW	0000h
8D46h	RX sampler latch E even positive 1 calibration unit start register (lane 3_rx_slc_eep1_start)	16	RW	801Fh
8D47h	RX sampler latch E even positive 1 calibration unit tune register (lane 3_rx_slc_eep1_tune)	16	RW	0000h
8D48h	RX sampler latch I even negative 0 calibration unit control register (lane3_rx_slc_ien0_ctrl)	16	RW	0021h
8D49h	RX sampler latch I even negative 0 calibration unit override register (lane3_rx_slc_ien0_ovrd)	16	RW	0000h
8D4Ah	RX sampler latch I even negative 0 calibration unit start register (lane 3_rx_slc_ien0_start)	16	RW	0021h
8D4Bh	RX sampler latch I even negative 0 calibration unit tune register (lane 3_rx_slc_ien0_tune)	16	RW	0000h
8D4Ch	RX sampler latch I even negative 1 calibration unit control register (lane3_rx_slc_ien1_ctrl)	16	RW	001Fh
8D4Dh	RX sampler latch I even negative 1 calibration unit override register (lane3_rx_slc_ien1_ovrd)	16	RW	0000h
8D4Eh	RX sampler latch I even negative 1 calibration unit start register (lane 3_rx_slc_ien1_start)	16	RW	801Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8D4Fh	RX sampler latch I even negative 1 calibration unit tune register (lane3_rx_slc_ien1_tune)	16	RW	0000h
8D50h	RX sampler latch Q even negative 0 calibration unit control register (lane3_rx_slc_qen0_ctrl)	16	RW	0021h
8D51h	RX sampler latch Q even negative 0 calibration unit override register (lane3_rx_slc_qen0_ovrd)	16	RW	0000h
8D52h	RX sampler latch Q even negative 0 calibration unit start register (lane3_rx_slc_qen0_start)	16	RW	0021h
8D53h	RX sampler latch Q even negative 0 calibration unit tune register (lane3_rx_slc_qen0_tune)	16	RW	0000h
8D54h	RX sampler latch Q even negative 1 calibration unit control register (lane3_rx_slc_qen1_ctrl)	16	RW	001Fh
8D55h	RX sampler latch Q even negative 1 calibration unit override register (lane3_rx_slc_qen1_ovrd)	16	RW	0000h
8D56h	RX sampler latch Q even negative 1 calibration unit start register (lane3_rx_slc_qen1_start)	16	RW	801Fh
8D57h	RX sampler latch Q even negative 1 calibration unit tune register (lane3_rx_slc_qen1_tune)	16	RW	0000h
8D58h	RX sampler latch E even negative 0 calibration unit control register (lane3_rx_slc_een0_ctrl)	16	RW	0021h
8D59h	RX sampler latch E even negative 0 calibration unit override register (lane3_rx_slc_een0_ovrd)	16	RW	0000h
8D5Ah	RX sampler latch E even negative 0 calibration unit start register (lane3_rx_slc_een0_start)	16	RW	0021h
8D5Bh	RX sampler latch E even negative 0 calibration unit tune register (lane3_rx_slc_een0_tune)	16	RW	0000h
8D5Ch	RX sampler latch E even negative 1 calibration unit control register (lane3_rx_slc_een1_ctrl)	16	RW	001Fh
8D5Dh	RX sampler latch E even negative 1 calibration unit override register (lane3_rx_slc_een1_ovrd)	16	RW	0000h
8D5Eh	RX sampler latch E even negative 1 calibration unit start register (lane3_rx_slc_een1_start)	16	RW	801Fh
8D5Fh	RX sampler latch E even negative 1 calibration unit tune register (lane3_rx_slc_een1_tune)	16	RW	0000h
8D60h	REE USB 3 general control state machine control register (lane3_rx_ree_u3gcsm_ctrl)	16	RW	0001h
8D61h	REE USB 3 general control state machine phase 1 equalization enable mask register (lane3_rx_ree_u3gcsm_eqenm_ph1)	16	RW	00C7h
8D62h	REE USB 3 general control state machine phase 2 equalization enable mask register (lane3_rx_ree_u3gcsm_eqenm_ph2)	16	RW	00C7h
8D63h	REE USB 3 general control state machine start timer value register (lane3_rx_ree_u3gcsm_start_tmr)	16	RW	0125h
8D64h	REE USB 3 general control state machine run phase 1 timer value register (lane3_rx_ree_u3gcsm_run_ph1_tmr)	16	RW	07A2h
8D65h	REE USB 3 general control state machine run phase 2 timer value register (lane3_rx_ree_u3gcsm_run_ph2_tmr)	16	RW	009Dh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8D68h	REE PCIe Gen 2 general control state machine control register (lane3_rx_ree_g2gcsm_ctrl)	16	RW	0001h
8D69h	REE PCIe Gen 2 general control state machine phase 1 equalization enable mask register (lane3_rx_ree_g2gcsm_eqenm_ph1)	16	RW	00C7h
8D6Ah	REE PCIe Gen 2 general control state machine phase 2 equalization enable mask register (lane3_rx_ree_g2gcsm_eqenm_ph2)	16	RW	00C7h
8D6Bh	REE PCIe Gen 2 general control state machine start timer value register (lane3_rx_ree_g2gcsm_start_tmr)	16	RW	0000h
8D6Ch	REE PCIe Gen 2 general control state machine run phase 1 timer value register (lane3_rx_ree_g2gcsm_run_ph1_tmr)	16	RW	0F43h
8D6Dh	REE PCIe Gen 2 general control state machine run phase 2 timer value register (lane3_rx_ree_g2gcsm_run_ph2_tmr)	16	RW	009Dh
8D78h	REE periodic general control state machine control register (lane3_rx_ree_pergcsm_ctrl)	16	RW	0001h
8D79h	REE periodic general control state machine phase 1 equalization enable mask register (lane3_rx_ree_pergcsm_eqenm_ph1)	16	RW	0080h
8D7Ah	REE periodic general control state machine phase 2 equalization enable mask register (lane3_rx_ree_pergcsm_eqenm_ph2)	16	RW	0080h
8D7Bh	REE periodic general control state machine start timer value register (lane3_rx_ree_pergcsm_start_tmr)	16	RW	0000h
8D7Ch	REE periodic general control state machine run phase 1 timer value register (lane3_rx_ree_pergcsm_run_ph1_tmr)	16	RW	FFFFh
8D7Dh	REE periodic general control state machine run phase 2 timer value register (lane3_rx_ree_pergcsm_run_ph2_tmr)	16	RW	009Dh
8D80h	REE tap 1 control register (lane3_rx_ree_tap1_ctrl)	16	RW	0000h
8D81h	REE tap 1 override register (lane3_rx_ree_tap1_ovrd)	16	RW	0000h
8D82h	REE tap 1 diagnostics register (lane3_rx_ree_tap1_diag)	16	RW	0000h
8D84h	REE tap 2 control register (lane3_rx_ree_tap2_ctrl)	16	RW	0400h
8D85h	REE tap 2 override register (lane3_rx_ree_tap2_ovrd)	16	RW	0000h
8D86h	REE tap 2 diagnostics register (lane3_rx_ree_tap2_diag)	16	RW	0000h
8D88h	REE tap 3 control register (lane3_rx_ree_tap3_ctrl)	16	RW	0400h
8D89h	REE tap 3 override register (lane3_rx_ree_tap3_ovrd)	16	RW	0000h
8D8Ah	REE tap 3 diagnostics register (lane3_rx_ree_tap3_diag)	16	RW	0000h
8D94h	REE analog enable control state machine delay timer value register (lane3_rx_ree_anaensm_del_tmr)	16	RW	02EEh
8D98h	REE peaking amp control register (lane3_rx_ree_peak_ctrl)	16	RW	0701h
8D99h	REE peaking amp code control register (lane3_rx_ree_peak_code_ctrl)	16	RW	2A0Eh
8D9Ah	REE peaking amp upper threshold register (lane3_rx_ree_peak_uthr)	16	RW	0004h
8D9Bh	REE peaking amp lower threshold register (lane3_rx_ree_peak_lthr)	16	RW	0000h
8D9Ch	REE peaking amp input override register (lane3_rx_ree_peak_iovrd)	16	RW	0000h
8D9Dh	REE peaking amp code override register (lane3_rx_ree_peak_covrd)	16	RW	0000h
8D9Eh	REE peaking amp diagnostics register (lane3_rx_ree_peak_diag)	16	RW	0020h
8DA0h	REE attenuation control register (lane3_rx_ree_atten_ctrl)	16	RW	0005h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8DA1h	REE attenuation threshold register (lane3_rx_ree_atten_thr)	16	RW	0C02h
8DA2h	REE attenuation counter register (lane3_rx_ree_atten_cnt)	16	RW	2100h
8DA3h	REE attenuation override register (lane3_rx_ree_atten_ovrd)	16	RW	0000h
8DA4h	REE attenuation diagnostics register (lane3_rx_ree_atten_diag)	16	RO	0000h
8DA8h	REE low frequency equalizer control register (lane3_rx_ree_lfeq_ctrl)	16	RW	0000h
8DA9h	REE low frequency equalizer override register (lane3_rx_ree_lfeq_ovrd)	16	RW	0000h
8DAAh	REE low frequency equalizer diagnostics register (lane3_rx_ree_lfeq_diag)	16	RW	0000h
8DACh	REE VGA gain control register (lane3_rx_ree_vga_gain_ctrl)	16	RW	1001h
8DADh	REE VGA gain override register (lane3_rx_ree_vga_gain_ovrd)	16	RW	0000h
8DAEh	REE VGA gain diagnostics register (lane3_rx_ree_vga_gain_diag)	16	RW	000Ah
8DAFh	REE VGA gain target adjust diagnostics register (lane3_rx_ree_vga_gain_tgt_diag)	16	RO	0000h
8DB0h	REE offset correction control register (lane3_rx_ree_off_cor_ctrl)	16	RW	0001h
8DB1h	REE offset correction override register (lane3_rx_ree_off_cor_ovrd)	16	RW	0000h
8DB2h	REE offset correction diagnostics register (lane3_rx_ree_off_cor_diag)	16	RW	0000h
8DB8h	REE adder configuration register (lane3_rx_ree_addr_cfg)	16	RW	0001h
8DB9h	REE tap 1 clip control register (lane3_rx_ree_tap1_clip)	16	RW	0519h
8DBAh	REE taps 2 and 3 clip control register (lane3_rx_ree_tap2ton_clip)	16	RW	0519h
8DBBh	REE control data mask register (lane3_rx_ree_ctrl_data_mask)	16	RW	4000h
8DBCh	REE diagnostic control register (lane3_rx_ree_diag_ctrl)	16	RW	0040h
8DBDh	REE control state machine gen mode control register 1 (lane3_rx_ree_smgm_ctrl1)	16	RW	0702h
8DBEh	REE control state machine gen mode control register 2 (lane3_rx_ree_smgm_ctrl2)	16	RW	0000h
8DC0h	RX ILL diagnostic control register (lane3_rx_diag_ill_ctrl)	16	RW	0000h
8DC1h	RX ILL IQ trim 0 register (lane3_rx_diag_ill_iq_trim0)	16	RW	533Fh
8DC2h	RX ILL E trim 0 register (lane3_rx_diag_ill_e_trim0)	16	RW	533Fh
8DC3h	RX ILL IQ trim 1 register (lane3_rx_diag_ill_iq_trim1)	16	RW	0002h
8DC4h	RX ILL E trim 1 register (lane3_rx_diag_ill_e_trim1)	16	RW	0002h
8DC5h	RX ILL IQ E trim 2 register (lane3_rx_diag_ill_iqe_trim2)	16	RW	3E00h
8DC6h	RX ILL IQ E trim 3 register (lane3_rx_diag_ill_iqe_trim3)	16	RW	199Fh
8DC7h	RX ILL IQ E trim 4 register (lane3_rx_diag_ill_iqe_trim4)	16	RW	0014h
8DC8h	RX ILL IQ E trim 5 register (lane3_rx_diag_ill_iqe_trim5)	16	RW	1504h
8DC9h	RX ILL IQ E trim 6 register (lane3_rx_diag_ill_iqe_trim6)	16	RW	0015h
8DD0h	DFE amp fine tuning register (lane3_rx_diag_dfe_amp_tune)	16	RW	4CCCh
8DD1h	DFE amp fine tuning 2 register (lane3_rx_diag_dfe_amp_tune_2)	16	RW	0C21h
8DD2h	REE DAC control register (lane3_rx_diag_ree_dac_ctrl)	16	RW	0004h
8DD3h	Receiver DFE control register 1 (lane3_rx_diag_dfe_ctrl1)	16	RW	0007h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8DD4h	Receiver DFE control register 2 (lane3_rx_diag_dfe_ctrl2)	16	RW	003Eh
8DD5h	Receiver DFE control register 3 (lane3_rx_diag_dfe_ctrl3)	16	RW	0ACAh
8DD6h	Nyquist control register (lane3_rx_diag_nqst_ctrl)	16	RW	0998h
8DD7h	Low frequency equalizer tuning register (lane3_rx_diag_lfeq_tune)	16	RW	0014h
8DD8h	RX control register (lane3_rx_diag_rxctrl)	16	RW	0060h
8DD9h	Receiver control reset diagnostic register (lane3_rx_diag_rst_diag)	16	RO	0000h
8DDCh	RX signal detect tuning and control register (lane3_rx_diag_sigdet_tune)	16	RW	1005h
8DDDh	RX LFPS detect tuning and control register (lane3_rx_diag_lfpsdet_tune)	16	RW	2410h
8DDEh	Signal detect test register (lane3_rx_diag_sd_test)	16	RW	0000h
8DE0h	RX sampler diagnostic control register (lane3_rx_diag_samp_ctrl)	16	RW	0001h
8DE1h	RX Sampler CML TO CMOS enable delay register (lane3_rx_diag_sc2c_delay)	16	RW	012Ch
8DE4h	MPHY control register 1 (lane3_rx_diag_mphy_ctrl_1)	16	RW	0000h
8DE5h	MPHY control register 2 (lane3_rx_diag_mphy_ctrl_2)	16	RW	0000h
8DE8h	RX loopback controller register (lane3_rx_diag_lpbk_ctrl)	16	RW	0000h
8DE9h	RX extra enable control override register (lane3_rx_diag_ectrl_ovrd)	16	RW	0000h
8DF0h	CML to CMOS bias trim register (lane3_rx_diag_cml2cmos_btrim)	16	RW	0000h
8DF1h	RX bias gen control register 1 (lane3_rx_diag_bias_gen_ctrl1)	16	RW	0000h
8DF2h	RX bias gen control register 2 (lane3_rx_diag_bias_gen_ctrl2)	16	RW	0000h
8DF3h	RX bias gen control register 3 (lane3_rx_diag_bias_gen_ctrl3)	16	RW	0000h
8DF4h	RX bias gen control register 4 (lane3_rx_diag_bias_gen_ctrl4)	16	RW	0000h
8DF5h	RX boundary scan test mode register (lane3_rx_diag_bs_tm)	16	RW	0000h
8DF6h	RX receiver front end test mode register 1 (lane3_rx_diag_rxfe_tm1)	16	RW	0000h
8DF7h	RX receiver front end test mode register 2 (lane3_rx_diag_rxfe_tm2)	16	RW	0000h
8DFEh	Receiver digital cover your alternatives register (lane3_rx_diag_dcya)	16	RW	0000h
8DFFh	Receiver analog cover your alternatives register (lane3_rx_diag_acya)	16	RW	0000h
C408h	HDP Lane Configuration register (lane0_phy_hdp_tx_ctl)	16	RW	0000h
C41Ch	DP Tx data low isolation register (lane0_phy_dp_iso_tx_data_lo)	16	RW	0000h
C41Dh	DP Tx data high isolation register (lane0_phy_dp_iso_tx_data_hi)	16	RW	0000h
C448h	HDP Lane Configuration register (lane1_phy_hdp_tx_ctl)	16	RW	0000h
C45Ch	DP Tx data low isolation register (lane1_phy_dp_iso_tx_data_lo)	16	RW	0000h
C45Dh	DP Tx data high isolation register (lane1_phy_dp_iso_tx_data_hi)	16	RW	0000h
C488h	HDP Lane Configuration register (lane2_phy_hdp_tx_ctl)	16	RW	0000h
C49Ch	DP Tx data low isolation register (lane2_phy_dp_iso_tx_data_lo)	16	RW	0000h
C49Dh	DP Tx data high isolation register (lane2_phy_dp_iso_tx_data_hi)	16	RW	0000h
C4C8h	HDP Lane Configuration register (lane3_phy_hdp_tx_ctl)	16	RW	0000h
C4DCh	DP Tx data low isolation register (lane3_phy_dp_iso_tx_data_lo)	16	RW	0000h
C4DDh	DP Tx data high isolation register (lane3_phy_dp_iso_tx_data_hi)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
CC00h	PMA transceiver control register (lane0_phy_pma_xcvr_ctrl)	16	RW	0000h
CC01h	PMA loopback control register (lane0_phy_pma_xcvr_lpbk)	16	RW	0000h
CC04h	PMA PSM current state lower register (lane0_phy_pma_psm_state_lo)	16	RO	0000h
CC05h	PMA PSM current state higher register (lane0_phy_pma_psm_state_hi)	16	RO	0000h
CC10h	PMA Isolation Transceiver control register (lane0_phy_pma_iso_xcvr_ctrl)	16	RW	0000h
CC11h	PMA TX configuration register (lane0_phy_pma_iso_tx_cfg)	16	RW	0000h
CC12h	PMA Isolation mode control register (lane0_phy_pma_iso_link_mode)	16	RW	C000h
CC13h	PMA Isolation power state control register (lane0_phy_pma_iso_pwrst_ctrl)	16	RW	0000h
CC14h	PMA transmit low data isolation register (lane0_phy_pma_iso_tx_data_lo)	16	RW	0000h
CC15h	PMA transmit high data isolation register (lane0_phy_pma_iso_tx_data_hi)	16	RW	0000h
CC40h	PMA transceiver control register (lane1_phy_pma_xcvr_ctrl)	16	RW	0000h
CC41h	PMA loopback control register (lane1_phy_pma_xcvr_lpbk)	16	RW	0000h
CC44h	PMA PSM current state lower register (lane1_phy_pma_psm_state_lo)	16	RO	0000h
CC45h	PMA PSM current state higher register (lane1_phy_pma_psm_state_hi)	16	RO	0000h
CC50h	PMA Isolation Transceiver control register (lane1_phy_pma_iso_xcvr_ctrl)	16	RW	0000h
CC51h	PMA TX configuration register (lane1_phy_pma_iso_tx_cfg)	16	RW	0000h
CC52h	PMA Isolation mode control register (lane1_phy_pma_iso_link_mode)	16	RW	C000h
CC53h	PMA Isolation power state control register (lane1_phy_pma_iso_pwrst_ctrl)	16	RW	0000h
CC54h	PMA transmit low data isolation register (lane1_phy_pma_iso_tx_data_lo)	16	RW	0000h
CC55h	PMA transmit high data isolation register (lane1_phy_pma_iso_tx_data_hi)	16	RW	0000h
CC80h	PMA transceiver control register (lane2_phy_pma_xcvr_ctrl)	16	RW	0000h
CC81h	PMA loopback control register (lane2_phy_pma_xcvr_lpbk)	16	RW	0000h
CC84h	PMA PSM current state lower register (lane2_phy_pma_psm_state_lo)	16	RO	0000h
CC85h	PMA PSM current state higher register (lane2_phy_pma_psm_state_hi)	16	RO	0000h
CC90h	PMA Isolation Transceiver control register (lane2_phy_pma_iso_xcvr_ctrl)	16	RW	0000h
CC91h	PMA TX configuration register (lane2_phy_pma_iso_tx_cfg)	16	RW	0000h
CC92h	PMA Isolation mode control register (lane2_phy_pma_iso_link_mode)	16	RW	C000h
CC93h	PMA Isolation power state control register (lane2_phy_pma_iso_pwrst_ctrl)	16	RW	0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
CC94h	PMA transmit low data isolation register (lane2_phy_pma_iso_tx_data_lo)	16	RW	0000h
CC95h	PMA transmit high data isolation register (lane2_phy_pma_iso_tx_data_hi)	16	RW	0000h
CCC0h	PMA transceiver control register (lane3_phy_pma_xcvr_ctrl)	16	RW	0000h
CCC1h	PMA loopback control register (lane3_phy_pma_xcvr_lpbk)	16	RW	0000h
CCC4h	PMA PSM current state lower register (lane3_phy_pma_psm_state_lo)	16	RO	0000h
CCC5h	PMA PSM current state higher register (lane3_phy_pma_psm_state_hi)	16	RO	0000h
CCD0h	PMA Isolation Transceiver control register (lane3_phy_pma_iso_xcvr_ctrl)	16	RW	0000h
CCD1h	PMA TX configuration register (lane3_phy_pma_iso_tx_cfg)	16	RW	0000h
CCD2h	PMA Isolation mode control register (lane3_phy_pma_iso_link_mode)	16	RW	C000h
CCD3h	PMA Isolation power state control register (lane3_phy_pma_iso_pwrst_ctrl)	16	RW	0000h
CCD4h	PMA transmit low data isolation register (lane3_phy_pma_iso_tx_data_lo)	16	RW	0000h
CCD5h	PMA transmit high data isolation register (lane3_phy_pma_iso_tx_data_hi)	16	RW	0000h
8_0000h	Product type ID register (cmn_pid_type)	16	RO	7364h
8_0001h	Product number 1 2 ID register (cmn_pid_num12)	16	RO	0000h
8_0002h	Product number 3 4 ID register (cmn_pid_num34)	16	RO	0000h
8_0003h	Product number 5 6 ID register (cmn_pid_num56)	16	RO	0000h
8_0004h	Product number 7 8 ID register (cmn_pid_num78)	16	RO	0000h
8_0005h	Product number 9 10 ID register (cmn_pid_num910)	16	RO	0000h
8_0006h	Product number 11 12 ID register (cmn_pid_num1112)	16	RO	0000h
8_0007h	Product revision ID register (cmn_pid_rev)	16	RO	0100h
8_0008h	Product technology manufacturer ID register (cmn_pid_mfg)	16	RO	0074h
8_0009h	Product technology process node ID register (cmn_pid_node)	16	RO	0028h
8_000Ah	Product technology process flavor ID register 0 (cmn_pid_flv0)	16	RO	0000h
8_000Bh	Product technology process flavor ID register 1 (cmn_pid_flv1)	16	RO	7400h
8_000Ch	Product I/O voltage ID register (cmn_pid_iov)	16	RO	0180h
8_000Dh	Product SerDes lanes ID register (cmn_pid_lanes)	16	RO	0202h
8_0010h	Product metal layers ID register 0 (cmn_pid_metal0)	16	RO	0000h
8_0011h	Product metal layers ID register 1 (cmn_pid_metal1)	16	RO	0000h
8_0012h	Product metal layers ID register 2 (cmn_pid_metal2)	16	RO	0000h
8_0013h	Product metal layers ID register 3 (cmn_pid_metal3)	16	RO	0000h
8_0014h	Product metal layer direction ID register (cmn_pid_metald)	16	RO	0000h
8_0020h	Startup state machine control register (cmn_ssm_sm_ctrl)	16	RW	0001h
8_0021h	Bandgap enable timer register (cmn_ssm_bandgap_tmr)	16	RW	0003h
8_0022h	Bias enable timer register (cmn_ssm_bias_tmr)	16	RW	007Dh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8_0027h	Startup state machine user defined control register (cmn_ssm_user_def_ctrl)	16	RW	0000h
8_0028h	PLL 0 control state machine control register (cmn_pllsm0_sm_ctrl)	16	RW	0000h
8_0029h	PLL 0 enable timer register (cmn_pllsm0_pllen_tmr)	16	RW	0019h
8_002Ah	PLL 0 pre-charge timer register (cmn_pllsm0_pllpre_tmr)	16	RW	0032h
8_002Bh	PLL 0 VREF delay timer register (cmn_pllsm0_pllvref_tmr)	16	RW	0019h
8_002Ch	PLL 0 lock delay timer register (cmn_pllsm0_plllock_tmr)	16	RW	0064h
8_002Dh	PLL 0 clock disable delay timer register (cmn_pllsm0_pllclkdis_tmr)	16	RW	0001h
8_002Fh	PLL 0 control state machine user defined control register (cmn_pllsm0_user_def_ctrl)	16	RW	1000h
8_0030h	PLL 1 control state machine control register (cmn_pllsm1_sm_ctrl)	16	RW	0000h
8_0031h	PLL 1 enable timer register (cmn_pllsm1_pllen_tmr)	16	RW	0019h
8_0032h	PLL 1 pre-charge timer register (cmn_pllsm1_pllpre_tmr)	16	RW	0032h
8_0033h	PLL 1 VREF delay timer register (cmn_pllsm1_pllvref_tmr)	16	RW	0019h
8_0034h	PLL 1 lock delay timer register (cmn_pllsm1_plllock_tmr)	16	RW	0064h
8_0035h	PLL 1 clock disable delay timer register (cmn_pllsm1_pllclkdis_tmr)	16	RW	0001h
8_0037h	PLL 1 control state machine user defined control register (cmn_pllsm1_user_def_ctrl)	16	RW	1000h
8_0060h	Common control power island control register (cmn_cdiag_pwr_ctrl)	16	RW	0012h
8_0061h	Common PSM clock control register (cmn_psm_clk_ctrl)	16	RW	0006h
8_0062h	Reference clock receiver control register (cmn_cdiag_refclk_ctrl)	16	RW	1000h
8_0064h	Common control CDB diagnostic register (cmn_cdiag_cdb_diag)	16	WO	0000h
8_0066h	Common control reset diagnostic register (cmn_cdiag_rst_diag)	16	RO	0000h
8_007Fh	Common control cover your alternatives register (cmn_cdiag_dcya)	16	RW	0000h
8_0080h	PLL 0 VCO calibration control register (cmn_pll0_vcocal_ctrl)	16	RW	00A0h
8_0081h	PLL 0 VCO calibration start point register (cmn_pll0_vcocal_start)	16	RW	30A0h
8_0082h	PLL 0 VCO calibration timer control register (cmn_pll0_vcocal_tctrl)	16	RW	0003h
8_0083h	PLL 0 VCO calibration override register (cmn_pll0_vcocal_ovrd)	16	RW	0000h
8_0084h	PLL 0 VCO calibration initialization timer register (cmn_pll0_vcocal_init_tmr)	16	RW	03E8h
8_0085h	PLL 0 VCO calibration iteration timer register (cmn_pll0_vcocal_iter_tmr)	16	RW	0064h
8_0086h	PLL 0 VCO calibration reference clock timer start value register (cmn_pll0_vcocal_reftim_start)	16	RW	0200h
8_0088h	PLL 0 VCO calibration PLL clock counter start value register (cmn_pll0_vcocal_pllcnt_start)	16	RW	0200h
8_0090h	PLL 0 lock reference counter start value register (cmn_pll0_lock_refcnt_start)	16	RW	00C8h
8_0091h	PLL 0 lock reference counter idle value register (cmn_pll0_lock_refcnt_idle)	16	RW	0004h
8_0092h	PLL 0 lock PLL counter start value register (cmn_pll0_lock_pllcnt_start)	16	RW	00C8h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8_0093h	PLL 0 lock PLL counter threshold value register (cmn_pll0_lock_pllcnt_thr)	16	RW	0003h
8_0094h	PLL 0 feedback divider integer register (cmn_pll0_intdiv)	16	RW	0000h
8_0095h	PLL 0 feedback divider fractional register (cmn_pll0_fracdiv)	16	RW	0000h
8_0096h	PLL 0 feedback divider high threshold register (cmn_pll0_high_thr)	16	RW	0000h
8_0097h	PLL 0 delta sigma modulator diagnostics register (cmn_pll0_dsm_diag)	16	RW	8020h
8_0098h	PLL 0 spread spectrum control register 1 (cmn_pll0_ss_ctrl1)	16	RW	8000h
8_0099h	PLL 0 spread spectrum control register 2 (cmn_pll0_ss_ctrl2)	16	RW	0000h
8_009Ch	PLL 0 VCO calibration V2I control register (cmn_pll0_vcocal_v2i_ctrl)	16	RW	0003h
8_00A0h	PLL 1 VCO calibration control register (cmn_pll1_vcocal_ctrl)	16	RW	00A0h
8_00A1h	PLL 1 VCO calibration start point register (cmn_pll1_vcocal_start)	16	RW	30A0h
8_00A2h	PLL 1 VCO calibration timer control register (cmn_pll1_vcocal_tctrl)	16	RW	0003h
8_00A3h	PLL 1 VCO calibration override register (cmn_pll1_vcocal_ovrd)	16	RW	0000h
8_00A4h	PLL 1 VCO calibration initialization timer register (cmn_pll1_vcocal_init_tmr)	16	RW	03E8h
8_00A5h	PLL 1 VCO calibration iteration timer register (cmn_pll1_vcocal_iter_tmr)	16	RW	0064h
8_00A6h	PLL 1 VCO calibration reference clock timer start value register (cmn_pll1_vcocal_reftim_start)	16	RW	0200h
8_00A8h	PLL 1 VCO calibration PLL clock counter start value register (cmn_pll1_vcocal_pllcnt_start)	16	RW	0200h
8_00B0h	PLL 1 lock reference counter start value register (cmn_pll1_lock_refcnt_start)	16	RW	00C8h
8_00B1h	PLL 1 lock reference counter idle value register (cmn_pll1_lock_refcnt_idle)	16	RW	0004h
8_00B2h	PLL 1 lock PLL counter start value register (cmn_pll1_lock_pllcnt_start)	16	RW	00C8h
8_00B3h	PLL 1 lock PLL counter threshold value register (cmn_pll1_lock_pllcnt_thr)	16	RW	0003h
8_00B4h	PLL 1 feedback divider integer register (cmn_pll1_intdiv)	16	RW	0000h
8_00B5h	PLL 1 feedback divider fractional register (cmn_pll1_fracdiv)	16	RW	0000h
8_00B6h	PLL 1 feedback divider high threshold register (cmn_pll1_high_thr)	16	RW	0000h
8_00B7h	PLL 1 delta sigma modulator diagnostics register (cmn_pll1_dsm_diag)	16	RW	8020h
8_00B8h	PLL 1 spread spectrum control register 1 (cmn_pll1_ss_ctrl1)	16	RW	8000h
8_00B9h	PLL 1 spread spectrum control register 2 (cmn_pll1_ss_ctrl2)	16	RW	0000h
8_00BCh	PLL 1 VCO calibration V2I control register (cmn_pll1_vcocal_v2i_ctrl)	16	RW	0003h
8_00C0h	Current calibration control register (cmn_ical_ctrl)	16	RW	0010h
8_00C1h	Current calibration override register (cmn_ical_ovrd)	16	RW	0000h
8_00C2h	Current calibration start register (cmn_ical_start)	16	RW	0010h
8_00C3h	Current calibration tune register (cmn_ical_tune)	16	RW	0000h
8_00C4h	Current calibration initialization timer register (cmn_ical_init_tmr)	16	RW	003Fh

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8_00C5h	Current calibration iteration timer register (cmn_ical_iter_tmr)	16	RW	0006h
8_00D0h	RX resistor calibration control register (cmn_rxcal_ctrl)	16	RW	0008h
8_00D1h	RX resistor calibration override register (cmn_rxcal_ovrd)	16	RW	0000h
8_00D2h	RX resistor calibration start register (cmn_rxcal_start)	16	RW	0008h
8_00D3h	RX resistor calibration tune register (cmn_rxcal_tune)	16	RW	0000h
8_00D4h	RX resistor calibration initialization timer register (cmn_rxcal_init_tmr)	16	RW	0020h
8_00D5h	RX resistor calibration iteration timer register (cmn_rxcal_iter_tmr)	16	RW	0006h
8_00E0h	TX pull-up resistor calibration control register (cmn_txpucal_ctrl)	16	RW	0028h
8_00E1h	TX pull-up resistor calibration override register (cmn_txpucal_ovrd)	16	RW	0000h
8_00E2h	TX pull-up resistor calibration start register (cmn_txpucal_start)	16	RW	0028h
8_00E3h	TX pull-up resistor calibration tune register (cmn_txpucal_tune)	16	RW	0000h
8_00E4h	TX pull-up resistor calibration initialization timer register (cmn_txpucal_init_tmr)	16	RW	0020h
8_00E5h	TX pull-up resistor calibration iteration timer register (cmn_txpucal_iter_tmr)	16	RW	0006h
8_00F0h	TX pull-down resistor calibration control register (cmn_txpdcal_ctrl)	16	RW	0028h
8_00F1h	TX pull-down resistor calibration override register (cmn_txpdcal_ovrd)	16	RW	0000h
8_00F2h	TX pull-down resistor calibration start register (cmn_txpdcal_start)	16	RW	0028h
8_00F3h	TX pull-down resistor calibration tune register (cmn_txpdcal_tune)	16	RW	0000h
8_00F4h	TX pull-down resistor calibration initialization timer register (cmn_txpdcal_init_tmr)	16	RW	0020h
8_00F5h	TX pull-down resistor calibration iteration timer register (cmn_txpdcal_iter_tmr)	16	RW	0006h
8_0100h	Current calibration adjust control register (cmn_ical_adj_ctrl)	16	RW	0000h
8_0101h	Current calibration adjust count register (cmn_ical_adj_cnt)	16	RW	0001h
8_0102h	Current calibration adjust initialization timer register (cmn_ical_adj_init_tmr)	16	RW	0020h
8_0103h	Current calibration adjust iteration timer register (cmn_ical_adj_iter_tmr)	16	RW	0006h
8_0104h	RX resistor calibration adjust control register (cmn_rx_adj_ctrl)	16	RW	0000h
8_0105h	RX resistor calibration adjust count register (cmn_rx_adj_cnt)	16	RW	0001h
8_0106h	RX resistor calibration adjust initialization timer register (cmn_rx_adj_init_tmr)	16	RW	0020h
8_0107h	RX resistor calibration adjust iteration timer register (cmn_rx_adj_iter_tmr)	16	RW	0006h
8_0108h	TX pull up resistor calibration adjust control register (cmn_txpu_adj_ctrl)	16	RW	0000h
8_0109h	TX pull up resistor calibration adjust count register (cmn_txpu_adj_cnt)	16	RW	0001h
8_010Ah	TX pull up resistor calibration adjust initialization timer register (cmn_txpu_adj_init_tmr)	16	RW	0020h
8_010Bh	TX pull up resistor calibration adjust iteration timer register (cmn_txpu_adj_iter_tmr)	16	RW	0006h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8_010Ch	TX pull down resistor calibration adjust control register (cmn_txpd_adj_ctrl)	16	RW	0000h
8_010Dh	TX pull down resistor calibration adjust count register (cmn_txpd_adj_cnt)	16	RW	0001h
8_010Eh	TX pull down resistor calibration adjust initialization timer register (cmn_txpd_adj_init_tmr)	16	RW	0020h
8_010Fh	TX pull down resistor calibration adjust iteration timer register (cmn_txpd_adj_iter_tmr)	16	RW	0006h
8_01A0h	Clock frequency measurement control register (cmn_cmsmt_clk_freq_msmt_ctrl)	16	RW	0000h
8_01A1h	Test clock selection register (cmn_cmsmt_test_clk_sel)	16	RW	0000h
8_01A2h	Reference clock timer value register (cmn_cmsmt_ref_clk_tmr_value)	16	RW	0000h
8_01A3h	Test clock counter value register (cmn_cmsmt_test_clk_cnt_value)	16	RO	0000h
8_01C0h	PLL 0 feedback divider value high override register (cmn_diag_pll0_fbh_ovrd)	16	RW	8016h
8_01C1h	PLL 0 feedback divider value low override register (cmn_diag_pll0_fbl_ovrd)	16	RW	8018h
8_01C2h	PLL 0 override register (cmn_diag_pll0_ovrd)	16	RW	0010h
8_01C3h	PLL 0 LDO control register (cmn_diag_pll0_ldo_ctrl)	16	RW	0000h
8_01C4h	PLL 0 test mode register (cmn_diag_pll0_test_mode)	16	RW	0000h
8_01C5h	PLL 0 voltage to current unit tuning register (cmn_diag_pll0_v2i_tune)	16	RW	0037h
8_01C6h	PLL 0 charge pump tuning register (cmn_diag_pll0_cp_tune)	16	RW	0072h
8_01C7h	PLL 0 loop filter programmability register (cmn_diag_pll0_lf_prog)	16	RW	0004h
8_01C8h	PLL 0 PTAT current slope tuning register 1 (cmn_diag_pll0_ptatis_tune1)	16	RW	0001h
8_01C9h	PLL 0 PTAT current slope tuning register 2 (cmn_diag_pll0_ptatis_tune2)	16	RW	0001h
8_01CAh	PLL 0 input clock control register (cmn_diag_pll0_inclk_ctrl)	16	RW	0001h
8_01D0h	PLL 1 feedback divider value high override register (cmn_diag_pll1_fbh_ovrd)	16	RW	8026h
8_01D1h	PLL 1 feedback divider value low override register (cmn_diag_pll1_fbl_ovrd)	16	RW	8026h
8_01D2h	PLL 1 override register (cmn_diag_pll1_ovrd)	16	RW	0010h
8_01D3h	PLL 1 LDO control register (cmn_diag_pll0_ld1_ctrl)	16	RW	0000h
8_01D4h	PLL 1 test mode register (cmn_diag_pll1_test_mode)	16	RW	0000h
8_01D5h	PLL 1 voltage to current unit tuning register (cmn_diag_pll1_v2i_tune)	16	RW	0037h
8_01D6h	PLL 1 charge pump tuning register (cmn_diag_pll1_cp_tune)	16	RW	0072h
8_01D7h	PLL 1 loop filter programmability register (cmn_diag_pll1_lf_prog)	16	RW	0004h
8_01D8h	PLL 1 PTAT current slope tuning register 1 (cmn_diag_pll1_ptatis_tune1)	16	RW	0001h
8_01D9h	PLL 1 PTAT current slope tuning register 2 (cmn_diag_pll1_ptatis_tune2)	16	RW	0001h
8_01DAh	PLL 1 input clock control register (cmn_diag_pll1_inclk_ctrl)	16	RW	0001h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
8_01E0h	Common high speed clock select register (cmn_diag_hsclk_sel)	16	RW	0010h
8_01E1h	Common calibration clock control register (cmn_diag_calclk_ctrl)	16	RW	0004h
8_01E2h	Common high speed reset release state machine control register (cmn_diag_hsrrsm_ctrl)	16	RW	0001h
8_01E3h	Common functions reset diagnostic register (cmn_diag_rst_diag)	16	RO	0000h
8_01E8h	Bandgap override register (cmn_diag_bandgap_ovrd)	16	RW	0000h
8_01E9h	Bias override register (cmn_diag_bias_ovrd)	16	RW	0000h
8_01ECh	Common periodic calibration adjust control register (cmn_diag_per_cal_adj)	16	RW	00F4h
8_01EDh	Common calibration control register (cmn_diag_cal_ctrl)	16	RW	0000h
8_01F0h	ATB control register 1 (cmn_diag_atb_ctrl1)	16	RW	0000h
8_01F1h	ATB control register 2 (cmn_diag_atb_ctrl2)	16	RW	0000h
8_01F4h	Common bandgap PTAT current control register (cmn_diag_bgi_ptat_ctrl)	16	RW	0000h
8_01F5h	Common bandgap current control register 1 (cmn_diag_bgi_ctrl1)	16	RW	0000h
8_01F6h	Common bandgap current control register 2 (cmn_diag_bgi_ctrl2)	16	RW	0000h
8_01F7h	Common control current control register 1 (cmn_diag_ctrli_ctrl1)	16	RW	0000h
8_01F8h	Common control current control register 2 (cmn_diag_ctrli_ctrl2)	16	RW	0000h
8_01FEh	Common digital functions cover your alternatives register (cmn_diag_dcya)	16	RW	0000h
8_01FFh	Common analog cover your alternatives register (cmn_diag_acya)	16	RW	0000h
8_5020h	DP Aux analog control 1 (tx_ana_ctrl_reg_1)	16	RW	0000h
8_5021h	DP Aux analog control 2 (tx_ana_ctrl_reg_2)	16	RW	0000h
8_5022h	Tx_coef_calc module Inputs (txda_coeff_calc_ctrl)	16	RW	0000h
8_5023h	Tx dig control reg 1 (tx_dig_ctrl_reg_1)	16	RW	0000h
8_5024h	Tx dig control reg 2 (tx_dig_ctrl_reg_2)	16	RW	0000h
8_5025h	DP Aux analog control 3 (txda_cya_auxda_cya)	16	RW	0000h
8_5026h	DP Aux analog control 4 (tx_ana_ctrl_reg_3)	16	RW	0000h
8_5027h	DP Aux analog control 5 (tx_ana_ctrl_reg_4)	16	RW	0000h
8_5028h	DP Aux analog status 1 (tx_ana_status_reg_1)	16	RO	0000h
8_5029h	DP Aux analog status 6 (tx_ana_ctrl_reg_5)	16	RW	0000h
8_9060h	rx_aux analog control1 (rx_ana_ctrl_reg_1)	16	RW	0000h
8_C008h	HDP Mode Control register (phy_hdp_mode_ctl)	16	RW	C004h
8_C009h	HDP Clock Control register (phy_hdp_clk_ctl)	16	RW	1205h
8_C00Fh	PHY status register (phy_sts)	16	RO	0000h
8_C010h	PHY common control signal isolation register (phy_iso_cmn_ctrl)	16	RW	0000h
8_C800h	PMA common control1 register (phy_pma_cmn_ctrl1)	16	RW	0000h
8_C801h	PMA common control2 register (phy_pma_cmn_ctrl2)	16	RO	000Ch
8_C802h	PMA SSM current state register (phy_pma_ssm_state)	16	RO	0020h
8_C803h	PMA PLL State Machine current state register (phy_pma_iso_pll_sm_state)	16	RO	0300h

Table continues on the next page...

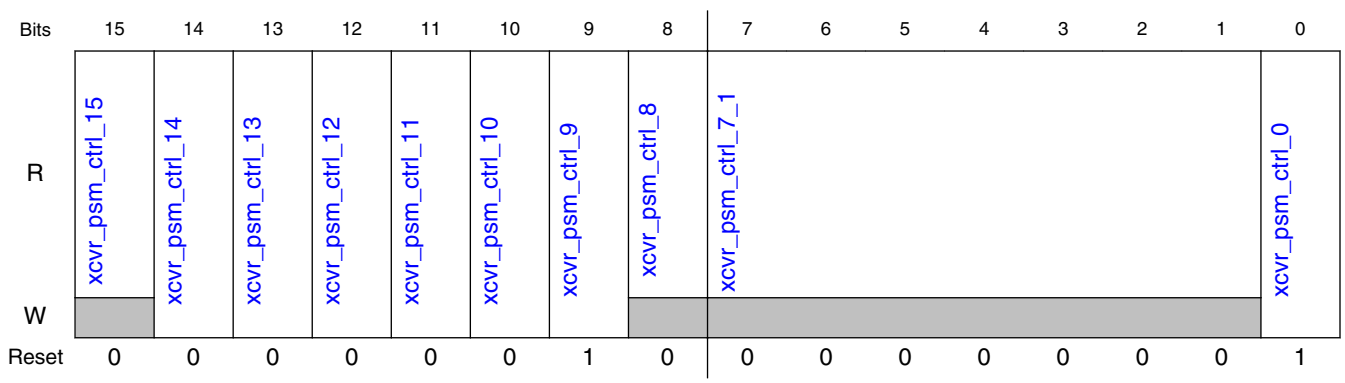
Offset	Register	Width (In bits)	Access	Reset value
8_C810h	PMA common control signal isolation register (phy_pma_iso_cmn_ctrl)	16	RW	0001h
8_C811h	PMA PLL control0 isolation register (phy_pma_iso_pll_ctrl0)	16	RW	000Fh
8_C812h	PMA PLL control1 isolation register (phy_pma_iso_pll_ctrl1)	16	RW	1122h
8_C81Fh	Isolation control register (phy_pma_isolation_ctrl)	16	RW	0000h

13.4.10.2.2 Power state machine control register (lane0_xcvr_psm_ctrl - lane3_xcvr_psm_ctrl)

13.4.10.2.2.1 Offset

Register	Offset
lane0_xcvr_psm_ctrl	4000h
lane1_xcvr_psm_ctrl	4400h
lane2_xcvr_psm_ctrl	4800h
lane3_xcvr_psm_ctrl	4C00h

13.4.10.2.2.2 Diagram



13.4.10.2.2.3 Fields

Field	Function
15 xcvr_psm_ctrl_15	Reserved

Table continues on the next page...

Field	Function
14 xcvr_psm_ctrl_14	Bypass A0 in delay from PSM ready : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the PSM ready state to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
13 xcvr_psm_ctrl_13	Bypass A0 in delay from A5 : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the A5 to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
12 xcvr_psm_ctrl_12	Bypass A0 in delay from A4 : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the A4 to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
11 xcvr_psm_ctrl_11	Bypass A0 in delay from A3 : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the A3 to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
10 xcvr_psm_ctrl_10	Bypass A0 in delay from A2 : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the A2 to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
9 xcvr_psm_ctrl_9	Bypass A0 in delay from A1 : When this bit is active (1b1), the A0 input delay is bypassed when transitioning from the A1 to the A0 power state. The result of this is the amount of time spent in the A0 in delay state is the value specified in the
8 xcvr_psm_ctrl_8	Reserved
7-1 xcvr_psm_ctrl_7_1	Reserved
0 xcvr_psm_ctrl_0	Skip lane re-calibration : When this bit is active (1b1), the lane calibration state will be skipped if it was previously run, unless the macro is disabled or reset.

13.4.10.2.3 Power state machine reset control register (lane0_xcvr_psm_rctrl - lane3_xcvr_psm_rctrl)

13.4.10.2.3.1 Offset

Register	Offset
lane0_xcvr_psm_rctrl	4001h
lane1_xcvr_psm_rctrl	4401h
lane2_xcvr_psm_rctrl	4801h
lane3_xcvr_psm_rctrl	4C01h

13.4.10.2.3.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	xcvr_psm_rctrl_15	xcvr_psm_rctrl_14	xcvr_psm_rctrl_13	xcvr_psm_rctrl_12	xcvr_psm_rctrl_11	xcvr_psm_rctrl_10	xcvr_psm_rctrl_9	xcvr_psm_rctrl_8	xcvr_psm_rctrl_7	xcvr_psm_rctrl_6	xcvr_psm_rctrl_5	xcvr_psm_rctrl_4	xcvr_psm_rctrl_3	xcvr_psm_rctrl_2	xcvr_psm_rctrl_1	xcvr_psm_rctrl_0
Reset	1	0	1	1	1	1	0	0	1	1	1	1	1	1	0	0

13.4.10.2.3.3 Fields

Field	Function
15 xcvr_psm_rctrl_15	RX reset active ready : Controls the state the receiver reset is changed to when in the ready power state.
14 xcvr_psm_rctrl_14	RX reset active calibration : Controls the state the receiver reset is changed to when in the calibration power state.
13 xcvr_psm_rctrl_13	RX reset active A5 : Controls the state the receiver reset is changed to when in the A5 entry power state.
12 xcvr_psm_rctrl_12	RX reset active A4 : Controls the state the receiver reset is changed to when in the A4 entry power state.
11 xcvr_psm_rctrl_11	RX reset active A3 : Controls the state the receiver reset is changed to when in the A3 entry power state.
10 xcvr_psm_rctrl_10	RX reset active A2 : Controls the state the receiver reset is changed to when in the A2 entry power state.
9 xcvr_psm_rctrl_9	RX reset active A1 : Controls the state the receiver reset is changed to when in the A1 entry power state.
8 xcvr_psm_rctrl_8	RX reset active A0 : Controls the state the receiver reset is changed to when in the A0 entry power state.
7 xcvr_psm_rctrl_7	TX reset active ready : Controls the state the transmitter reset is changed to when in the ready power state.
6	TX reset active calibration : Controls the state the transmitter reset is changed to when in the calibration power state.

Table continues on the next page...

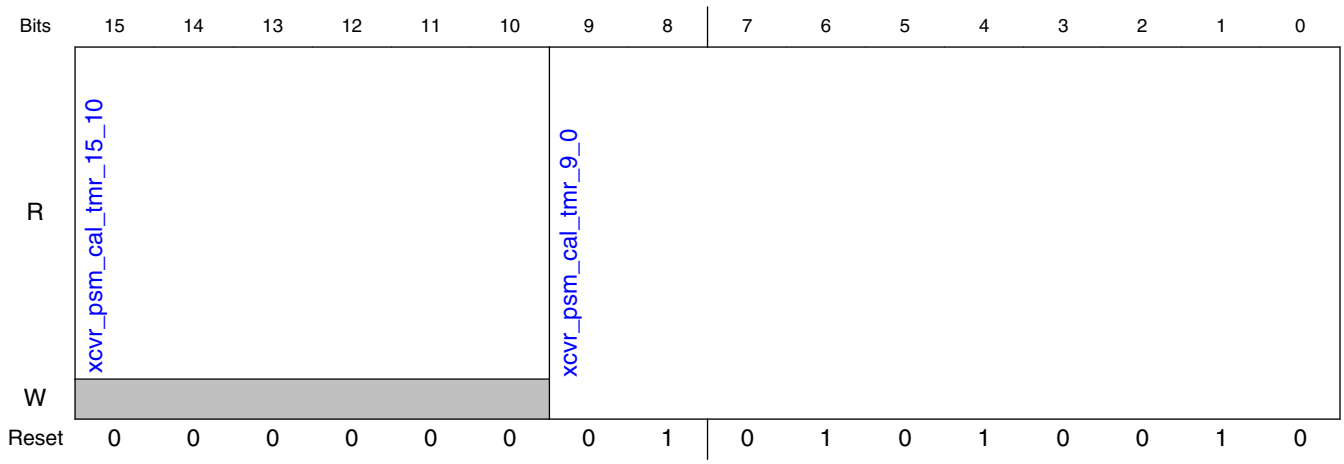
Field	Function
xcvr_psm_rctrl_6	
5 xcvr_psm_rctrl_5	TX reset active A5 : Controls the state the transmitter reset is changed to when in the A5 entry power state.
4 xcvr_psm_rctrl_4	TX reset active A4 : Controls the state the transmitter reset is changed to when in the A4 entry power state.
3 xcvr_psm_rctrl_3	TX reset active A3 : Controls the state the transmitter reset is changed to when in the A3 entry power state.
2 xcvr_psm_rctrl_2	TX reset active A2 : Controls the state the transmitter reset is changed to when in the A2 entry power state.
1 xcvr_psm_rctrl_1	TX reset active A1 : Controls the state the transmitter reset is changed to when in the A1 entry power state.
0 xcvr_psm_rctrl_0	TX reset active A0 : Controls the state the transmitter reset is changed to when in the A0 entry power state.

13.4.10.2.4 PSM calibration delay timer register (lane0_xcvr_psm_cal_tmr - lane3_xcvr_psm_cal_tmr)

13.4.10.2.4.1 Offset

Register	Offset
lane0_xcvr_psm_cal_tmr	4002h
lane1_xcvr_psm_cal_tmr	4402h
lane2_xcvr_psm_cal_tmr	4802h
lane3_xcvr_psm_cal_tmr	4C02h

13.4.10.2.4.2 Diagram



13.4.10.2.4.3 Fields

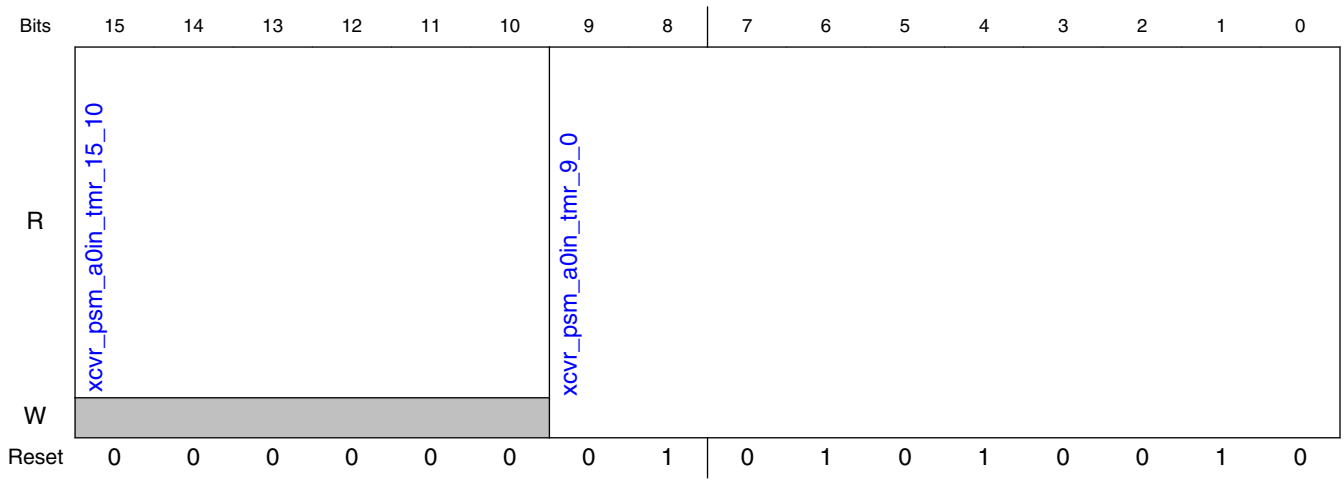
Field	Function
15-10 xcvr_psm_cal_tmr_15_10	Reserved
9-0 xcvr_psm_cal_tmr_9_0	PSM calibration delay state timer value : Value used for the timer when the power state machine is in the PSM calibration delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_lane_cal_del.

13.4.10.2.5 A0 in delay timer register (lane0_xcvr_psm_a0in_tmr - lane3_xcvr_psm_a0in_tmr)

13.4.10.2.5.1 Offset

Register	Offset
lane0_xcvr_psm_a0in_tmr	4003h
lane1_xcvr_psm_a0in_tmr	4403h
lane2_xcvr_psm_a0in_tmr	4803h
lane3_xcvr_psm_a0in_tmr	4C03h

13.4.10.2.5.2 Diagram



13.4.10.2.5.3 Fields

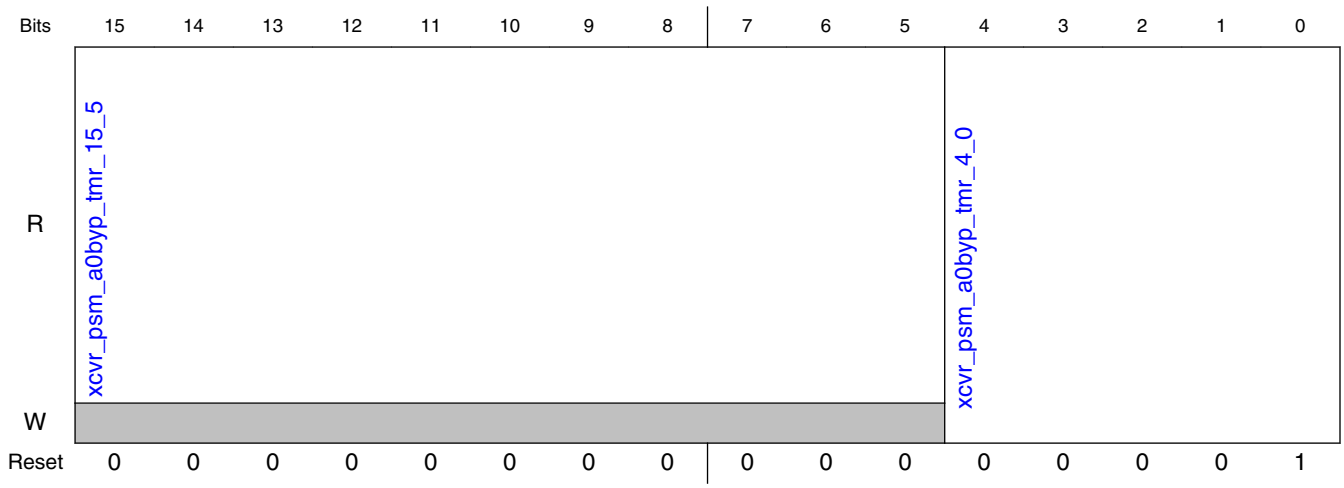
Field	Function
15-10 xcvr_psm_a0in_tmr_15_10	Reserved
9-0 xcvr_psm_a0in_tmr_9_0	A0 in delay state timer value : Value used for the timer when the power state machine is in the A0 in delay state, unless the timer is bypassed under the control of the bypass A0 bits in the

13.4.10.2.6 A0 in bypass timer register (lane0_xcvr_psm_a0byp_tmr - lane 3_xcvr_psm_a0byp_tmr)

13.4.10.2.6.1 Offset

Register	Offset
lane0_xcvr_psm_a0byp_tmr	4004h
lane1_xcvr_psm_a0byp_tmr	4404h
lane2_xcvr_psm_a0byp_tmr	4804h
lane3_xcvr_psm_a0byp_tmr	4C04h

13.4.10.2.6.2 Diagram



13.4.10.2.6.3 Fields

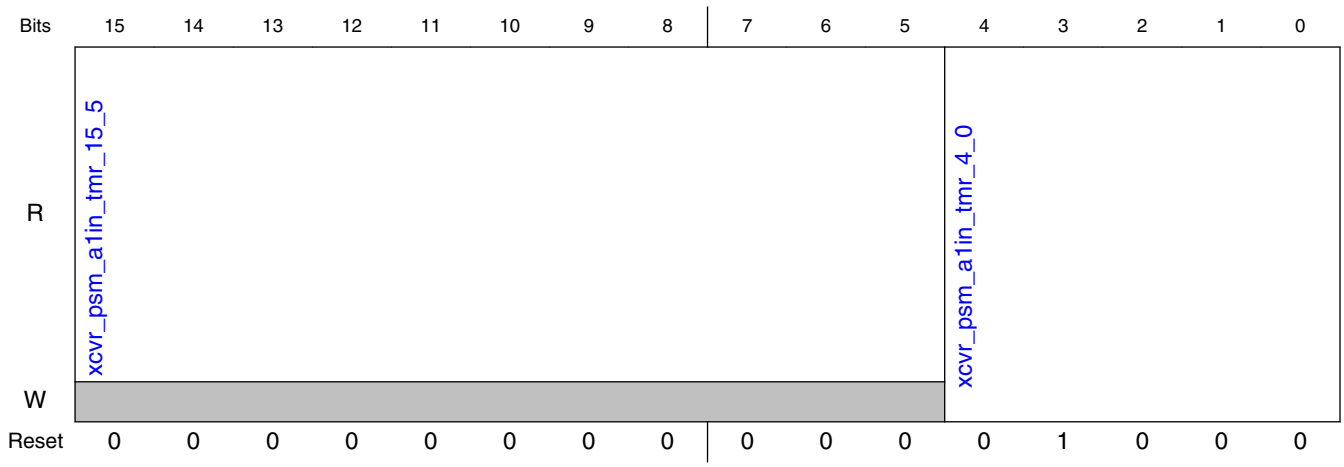
Field	Function
15-5 xcvr_psm_a0byp_tmr_15_5	Reserved
4-0 xcvr_psm_a0byp_tmr_4_0	A0 in delay state bypass timer value : Value used for the timer when the power state machine is in the A0 in delay state and the timer is bypassed under the control of the bypass A0 bits in the

13.4.10.2.7 A1 in delay timer register (lane0_xcvr_psm_a1in_tmr - lane3_xcvr_psm_a1in_tmr)

13.4.10.2.7.1 Offset

Register	Offset
lane0_xcvr_psm_a1in_tmr	4005h
lane1_xcvr_psm_a1in_tmr	4405h
lane2_xcvr_psm_a1in_tmr	4805h
lane3_xcvr_psm_a1in_tmr	4C05h

13.4.10.2.7.2 Diagram



13.4.10.2.7.3 Fields

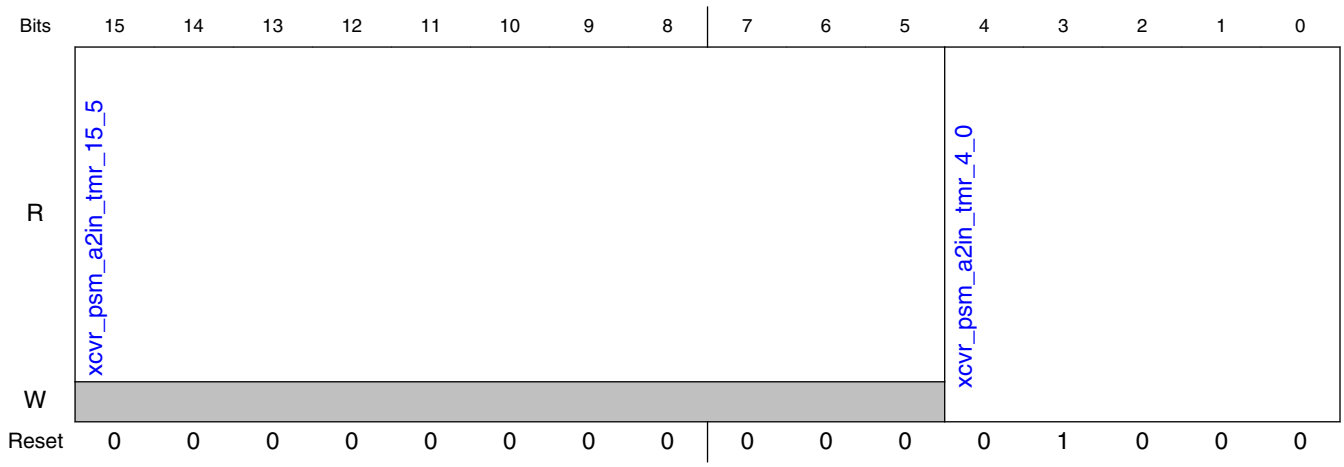
Field	Function
15-5 xcvr_psm_a1in_tmr_15_5	Reserved
4-0 xcvr_psm_a1in_tmr_4_0	A1 in delay state timer value : Value used for the timer when the power state machine is in the A1 in delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A1_in_del.

13.4.10.2.8 A2 in delay timer register (lane0_xcvr_psm_a2in_tmr - lane3_xcvr_psm_a2in_tmr)

13.4.10.2.8.1 Offset

Register	Offset
lane0_xcvr_psm_a2in_tmr	4006h
lane1_xcvr_psm_a2in_tmr	4406h
lane2_xcvr_psm_a2in_tmr	4806h
lane3_xcvr_psm_a2in_tmr	4C06h

13.4.10.2.8.2 Diagram



13.4.10.2.8.3 Fields

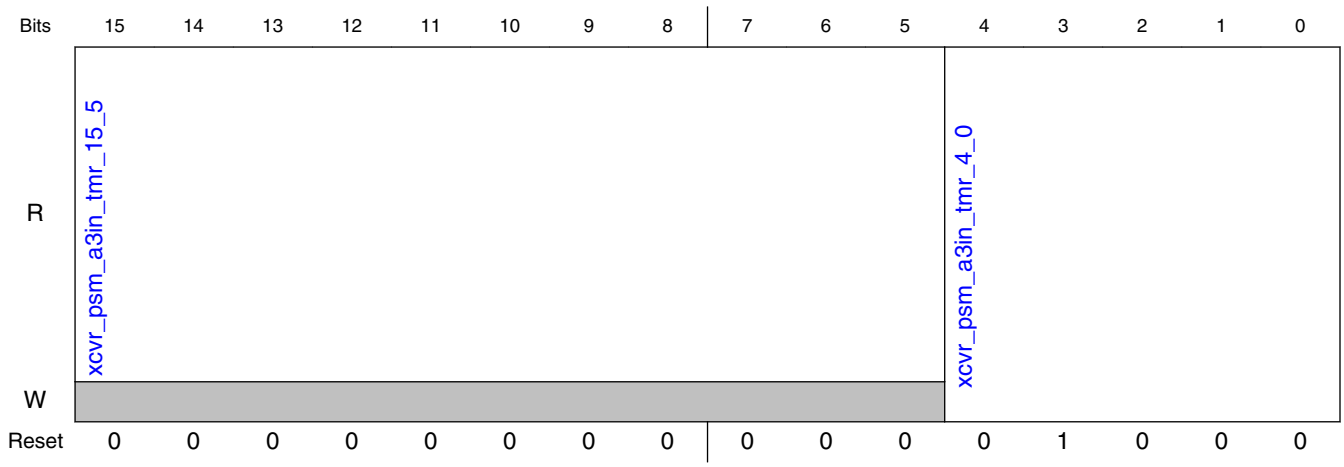
Field	Function
15-5 xcvr_psm_a2in_tmr_15_5	Reserved
4-0 xcvr_psm_a2in_tmr_4_0	A2 in delay state timer value : Value used for the timer when the power state machine is in the A2 in delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A2_in_del.

13.4.10.2.9 A3 in delay timer register (lane0_xcvr_psm_a3in_tmr - lane3_xcvr_psm_a3in_tmr)

13.4.10.2.9.1 Offset

Register	Offset
lane0_xcvr_psm_a3in_tmr	4007h
lane1_xcvr_psm_a3in_tmr	4407h
lane2_xcvr_psm_a3in_tmr	4807h
lane3_xcvr_psm_a3in_tmr	4C07h

13.4.10.2.9.2 Diagram



13.4.10.2.9.3 Fields

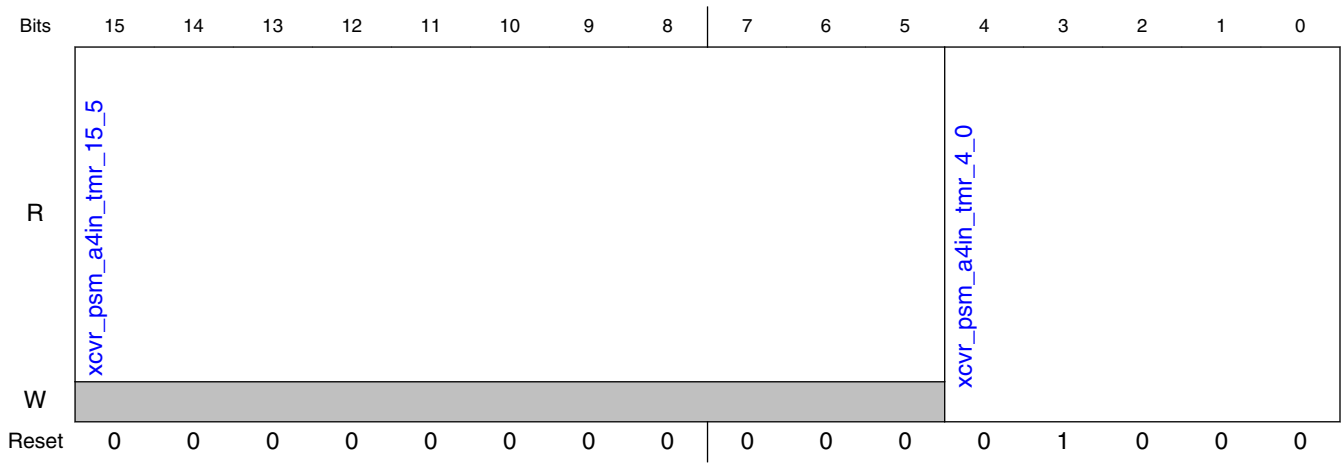
Field	Function
15-5 xcvr_psm_a3in_tmr_15_5	Reserved
4-0 xcvr_psm_a3in_tmr_4_0	A3 in delay state timer value : Value used for the timer when the power state machine is in the A3 in delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A3_in_del.

13.4.10.2.10 A4 in delay timer register (lane0_xcvr_psm_a4in_tmr - lane3_xcvr_psm_a4in_tmr)

13.4.10.2.10.1 Offset

Register	Offset
lane0_xcvr_psm_a4in_tmr	4008h
lane1_xcvr_psm_a4in_tmr	4408h
lane2_xcvr_psm_a4in_tmr	4808h
lane3_xcvr_psm_a4in_tmr	4C08h

13.4.10.2.10.2 Diagram



13.4.10.2.10.3 Fields

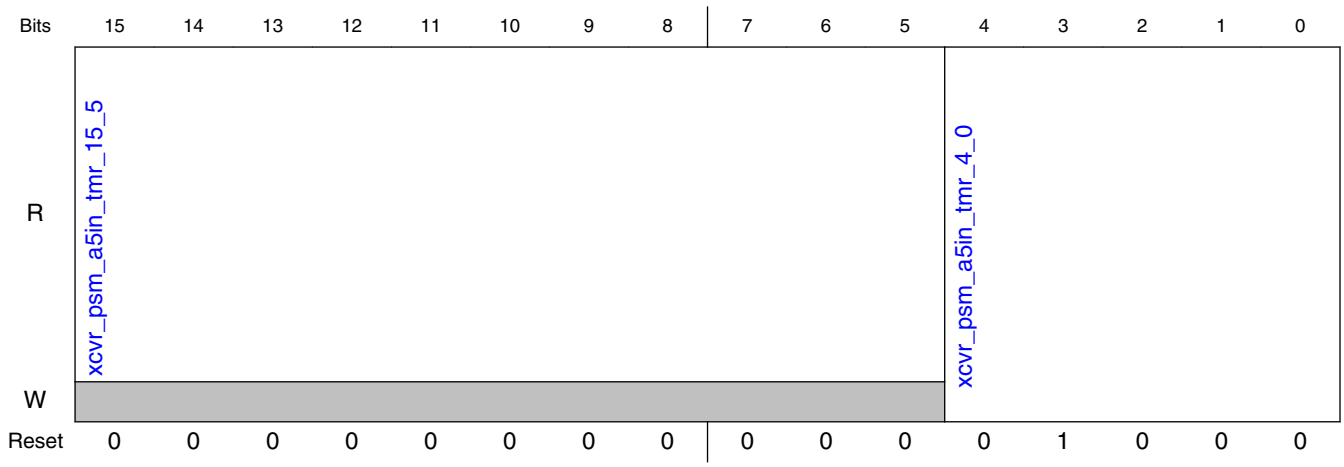
Field	Function
15-5 xcvr_psm_a4in_tmr_15_5	Reserved
4-0 xcvr_psm_a4in_tmr_4_0	A4 in delay state timer value : Value used for the timer when the power state machine is in the A4 in delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A4_in_del.

13.4.10.2.11 A5 in delay timer register (lane0_xcvr_psm_a5in_tmr - lane3_xcvr_psm_a5in_tmr)

13.4.10.2.11.1 Offset

Register	Offset
lane0_xcvr_psm_a5in_tmr	4009h
lane1_xcvr_psm_a5in_tmr	4409h
lane2_xcvr_psm_a5in_tmr	4809h
lane3_xcvr_psm_a5in_tmr	4C09h

13.4.10.2.11.2 Diagram



13.4.10.2.11.3 Fields

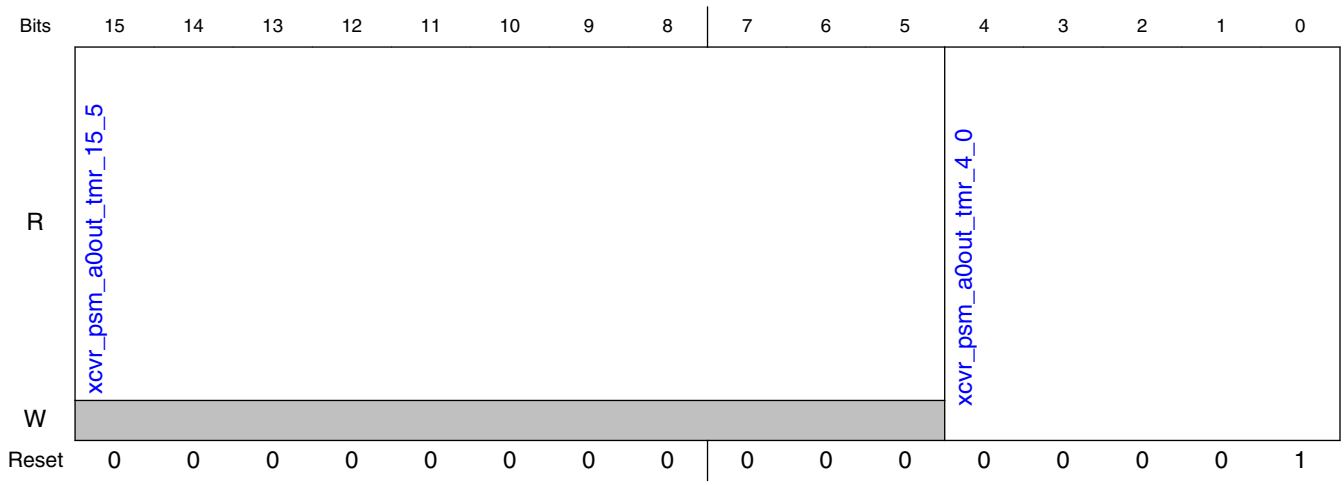
Field	Function
15-5 xcvr_psm_a5in_tmr_15_5	Reserved
4-0 xcvr_psm_a5in_tmr_4_0	A5 in delay state timer value : Value used for the timer when the power state machine is in the A5 in delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A5_in_del.

13.4.10.2.12 A0 out delay timer register (lane0_xcvr_psm_a0out_tmr - lane 3_xcvr_psm_a0out_tmr)

13.4.10.2.12.1 Offset

Register	Offset
lane0_xcvr_psm_a0out_tmr	400Ah
lane1_xcvr_psm_a0out_tmr	440Ah
lane2_xcvr_psm_a0out_tmr	480Ah
lane3_xcvr_psm_a0out_tmr	4C0Ah

13.4.10.2.12.2 Diagram



13.4.10.2.12.3 Fields

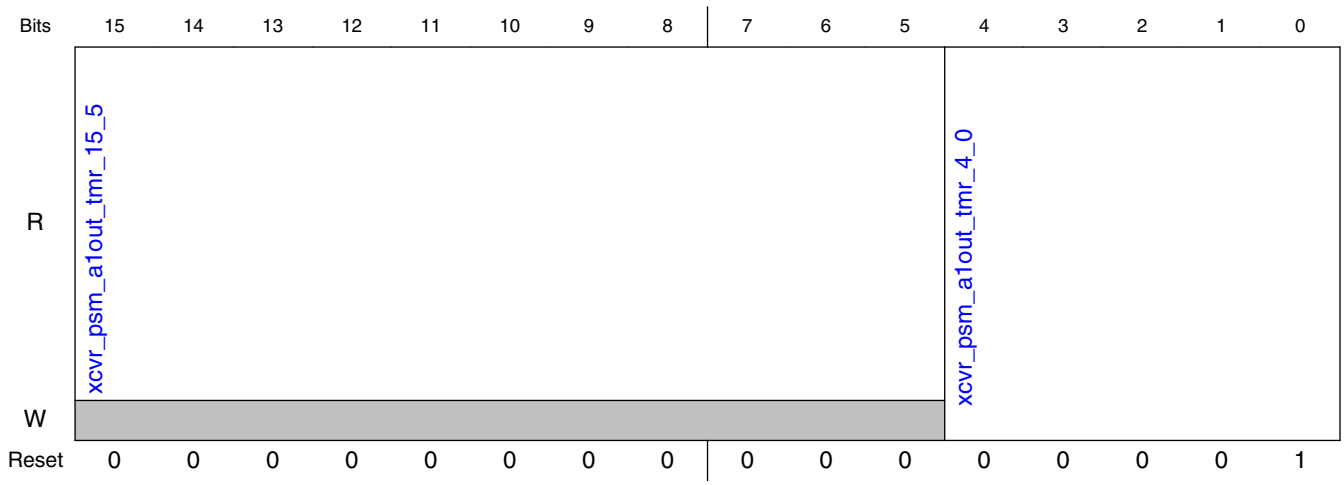
Field	Function
15-5 xcvr_psm_a0out_tmr_15_5	Reserved
4-0 xcvr_psm_a0out_tmr_4_0	A0 out delay state timer value : Value used for the timer when the power state machine is in the A0 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A0_out_del.

13.4.10.2.13 A1 out delay timer register (lane0_xcvr_psm_a1out_tmr - lane 3_xcvr_psm_a1out_tmr)

13.4.10.2.13.1 Offset

Register	Offset
lane0_xcvr_psm_a1out_tmr	400Bh
lane1_xcvr_psm_a1out_tmr	440Bh
lane2_xcvr_psm_a1out_tmr	480Bh
lane3_xcvr_psm_a1out_tmr	4C0Bh

13.4.10.2.13.2 Diagram



13.4.10.2.13.3 Fields

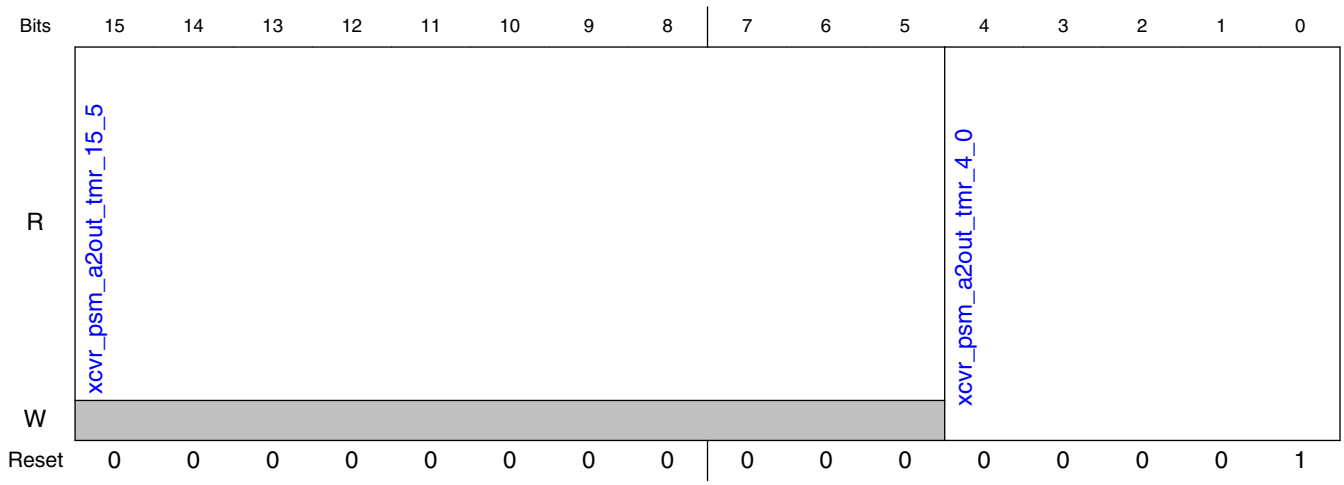
Field	Function
15-5 xcvr_psm_a1out_tmr_15_5	Reserved
4-0 xcvr_psm_a1out_tmr_4_0	A1 out delay state timer value : Value used for the timer when the power state machine is in the A1 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A1_out_del.

13.4.10.2.14 A2 out delay timer register (lane0_xcvr_psm_a2out_tmr - lane3_xcvr_psm_a2out_tmr)

13.4.10.2.14.1 Offset

Register	Offset
lane0_xcvr_psm_a2out_tmr	400Ch
lane1_xcvr_psm_a2out_tmr	440Ch
lane2_xcvr_psm_a2out_tmr	480Ch
lane3_xcvr_psm_a2out_tmr	4C0Ch

13.4.10.2.14.2 Diagram



13.4.10.2.14.3 Fields

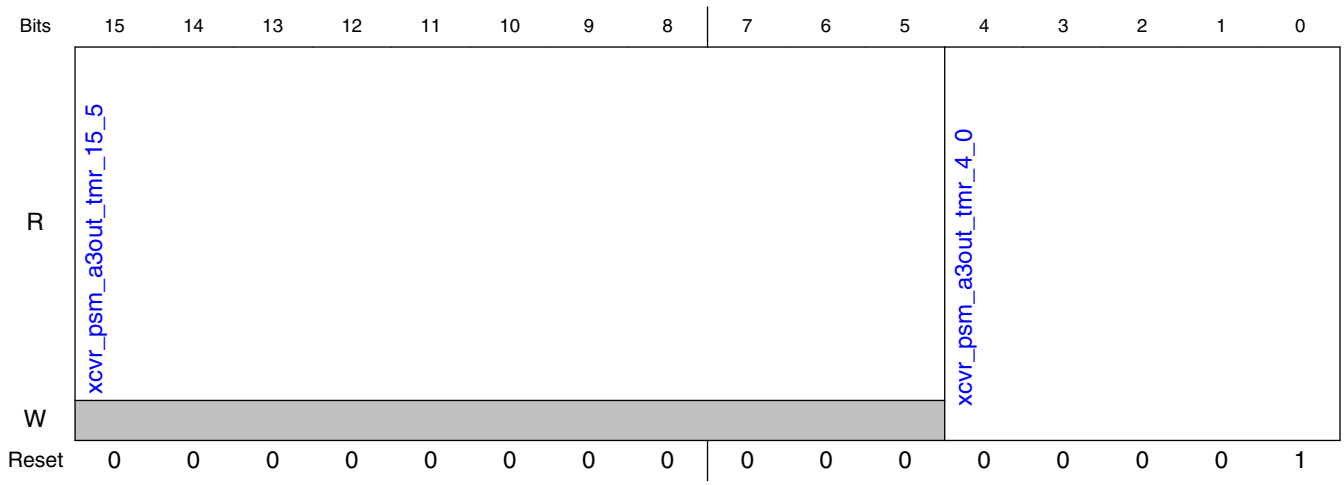
Field	Function
15-5 xcvr_psm_a2out_tmr_15_5	Reserved
4-0 xcvr_psm_a2out_tmr_4_0	A2 out delay state timer value : Value used for the timer when the power state machine is in the A2 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A2_out_del.

13.4.10.2.15 A3 out delay timer register (lane0_xcvr_psm_a3out_tmr - lane 3_xcvr_psm_a3out_tmr)

13.4.10.2.15.1 Offset

Register	Offset
lane0_xcvr_psm_a3out_tmr	400Dh
lane1_xcvr_psm_a3out_tmr	440Dh
lane2_xcvr_psm_a3out_tmr	480Dh
lane3_xcvr_psm_a3out_tmr	4C0Dh

13.4.10.2.15.2 Diagram



13.4.10.2.15.3 Fields

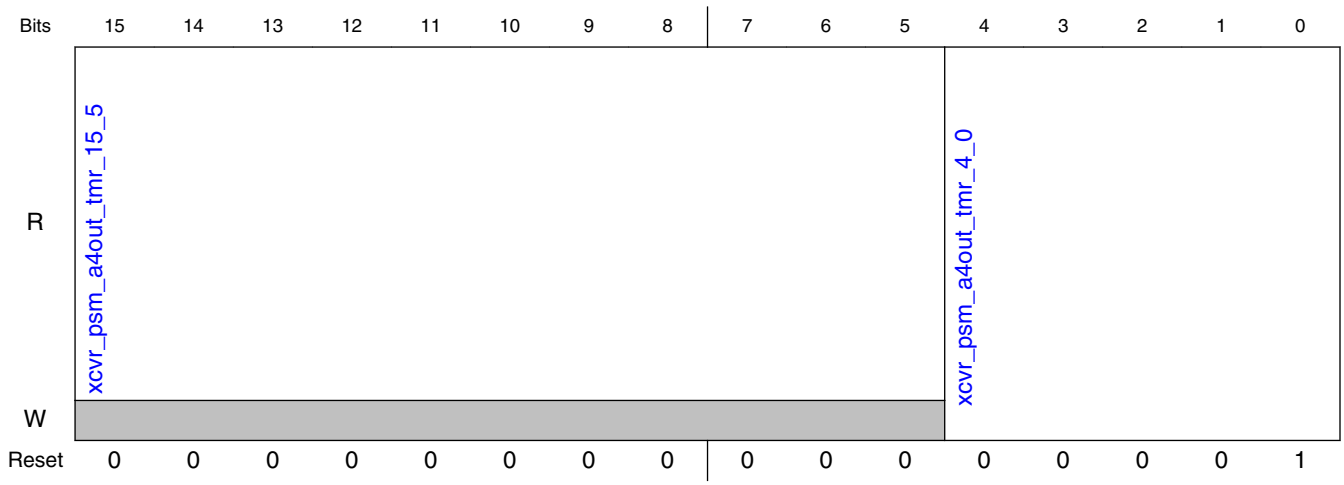
Field	Function
15-5 xcvr_psm_a3out_tmr_15_5	Reserved
4-0 xcvr_psm_a3out_tmr_4_0	A3 out delay state timer value : Value used for the timer when the power state machine is in the A3 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A3_out_del.

13.4.10.2.16 A4 out delay timer register (lane0_xcvr_psm_a4out_tmr - lane 3_xcvr_psm_a4out_tmr)

13.4.10.2.16.1 Offset

Register	Offset
lane0_xcvr_psm_a4out_tmr	400Eh
lane1_xcvr_psm_a4out_tmr	440Eh
lane2_xcvr_psm_a4out_tmr	480Eh
lane3_xcvr_psm_a4out_tmr	4C0Eh

13.4.10.2.16.2 Diagram



13.4.10.2.16.3 Fields

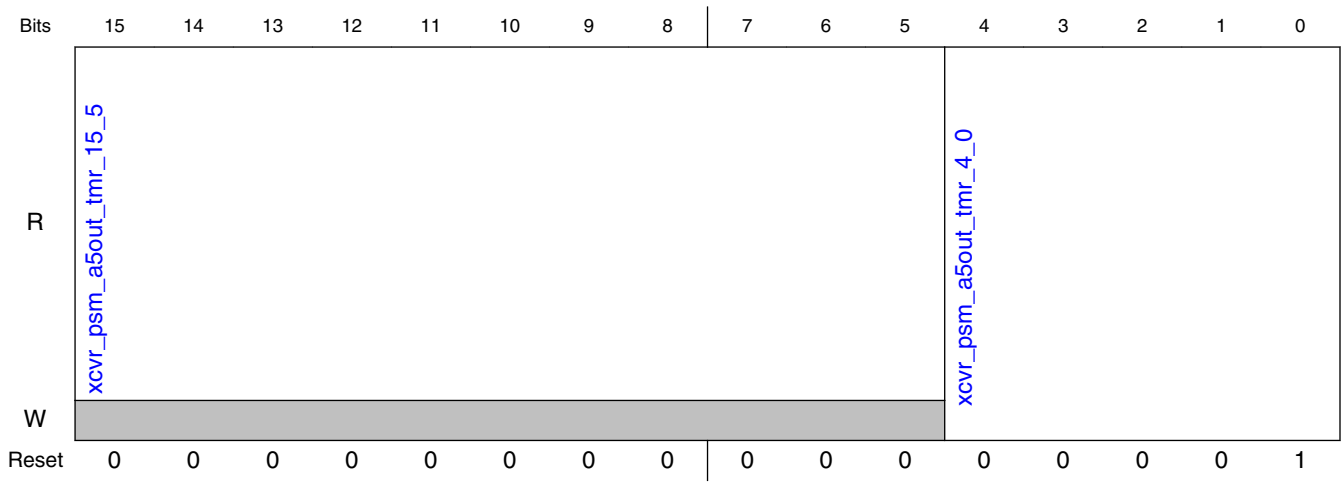
Field	Function
15-5 xcvr_psm_a4out_tmr_15_5	Reserved
4-0 xcvr_psm_a4out_tmr_4_0	A4 out delay state timer value : Value used for the timer when the power state machine is in the A4 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A4_out_del.

13.4.10.2.17 A5 out delay timer register (lane0_xcvr_psm_a5out_tmr - lane 3_xcvr_psm_a5out_tmr)

13.4.10.2.17.1 Offset

Register	Offset
lane0_xcvr_psm_a5out_tmr	400Fh
lane1_xcvr_psm_a5out_tmr	440Fh
lane2_xcvr_psm_a5out_tmr	480Fh
lane3_xcvr_psm_a5out_tmr	4C0Fh

13.4.10.2.17.2 Diagram



13.4.10.2.17.3 Fields

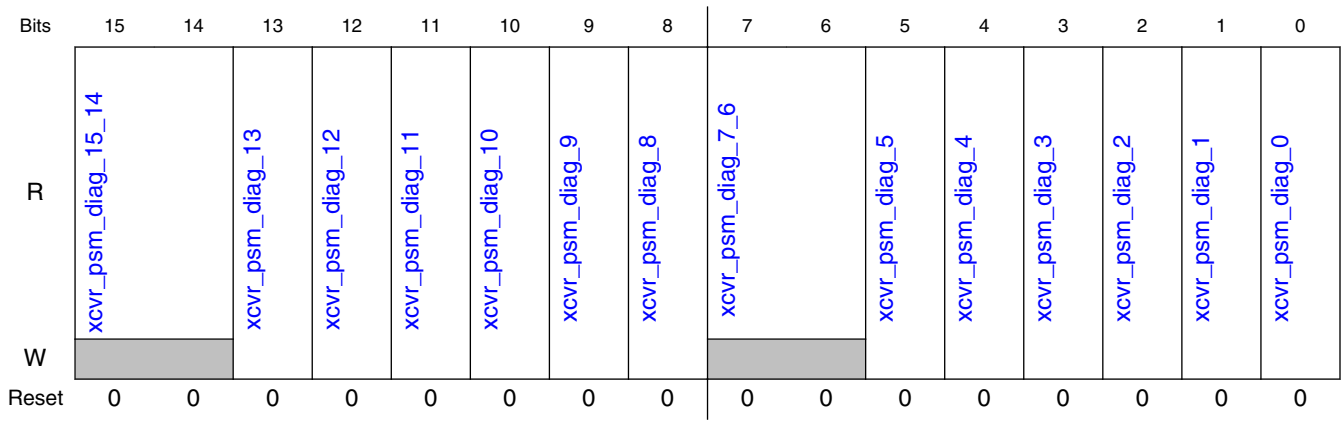
Field	Function
15-5 xcvr_psm_a5out_tmr_15_5	Reserved
4-0 xcvr_psm_a5out_tmr_4_0	A5 out delay state timer value : Value used for the timer when the power state machine is in the A5 out delay state. This timer delay is specified as the number of PSM clocks to count to implement the time required for tpsm_A5_out_del.

13.4.10.2.18 Power state machine diagnostic register (lane0_xcvr_psm_diag - lane3_xcvr_psm_diag)

13.4.10.2.18.1 Offset

Register	Offset
lane0_xcvr_psm_diag	4010h
lane1_xcvr_psm_diag	4410h
lane2_xcvr_psm_diag	4810h
lane3_xcvr_psm_diag	4C10h

13.4.10.2.18.2 Diagram



13.4.10.2.18.3 Fields

Field	Function
15-14 xcvr_psm_diag_15_14	Reserved
13 xcvr_psm_diag_13	Force A5 exit acknowledge : Setting this bit to 1b1 forces the
12 xcvr_psm_diag_12	Force A4 exit acknowledge : Setting this bit to 1b1 forces the
11 xcvr_psm_diag_11	Force A3 exit acknowledge : Setting this bit to 1b1 forces the
10 xcvr_psm_diag_10	Force A2 exit acknowledge : Setting this bit to 1b1 forces the
9 xcvr_psm_diag_9	Force A1 exit acknowledge : Setting this bit to 1b1 forces the
8 xcvr_psm_diag_8	Force A0 exit acknowledge : Setting this bit to 1b1 forces the
7-6 xcvr_psm_diag_7_6	Reserved
5 xcvr_psm_diag_5	Force A5 entry acknowledge : Setting this bit to 1b1 forces the

Table continues on the next page...

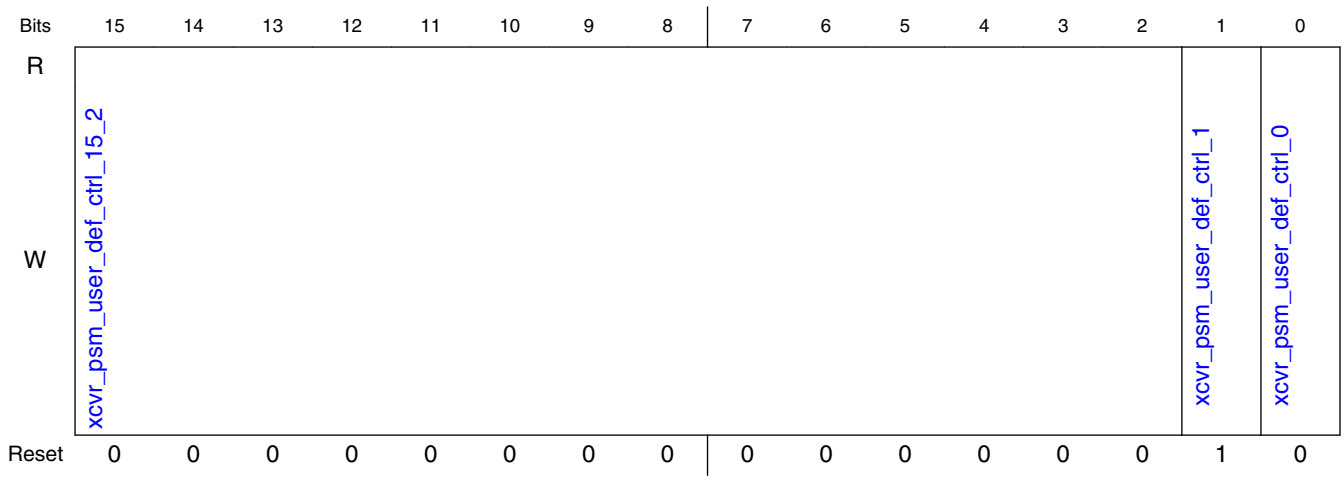
Field	Function
4 xcvr_psm_diag_4	Force A4 entry acknowledge : Setting this bit to 1b1 forces the
3 xcvr_psm_diag_3	Force A3 entry acknowledge : Setting this bit to 1b1 forces the
2 xcvr_psm_diag_2	Force A2 entry acknowledge : Setting this bit to 1b1 forces the
1 xcvr_psm_diag_1	Force A1 entry acknowledge : Setting this bit to 1b1 forces the
0 xcvr_psm_diag_0	Force A0 entry acknowledge : Setting this bit to 1b1 forces the

13.4.10.2.19 Power state machine user defined control register (lane0_xcvr_psm_user_def_ctrl - lane3_xcvr_psm_user_def_ctrl)

13.4.10.2.19.1 Offset

Register	Offset
lane0_xcvr_psm_user_def_ctrl	401Fh
lane1_xcvr_psm_user_def_ctrl	441Fh
lane2_xcvr_psm_user_def_ctrl	481Fh
lane3_xcvr_psm_user_def_ctrl	4C1Fh

13.4.10.2.19.2 Diagram



13.4.10.2.19.3 Fields

Field	Function
15-2 xcvr_psm_user_def_ctrl_15_2	Reserved
1 xcvr_psm_user_def_ctrl_1	Disable PSM clock gating: This bit controls the PSM clock gating function. By disabling the PSM clock gating, faster state transition times can be achieved, at the expense of this clock always being enabled.
0 xcvr_psm_user_def_ctrl_0	Disable early A0 acknowledge response : Setting this bit to 1b1, will disable the function where an A0 acknowledge is generated early, when transitioning from A1 to A0, relative to the PSM actually being in the A0 power state.

13.4.10.2.20 TX coefficient controller control register (lane0_tx_txcc_ctrl - lane3_tx_txcc_ctrl)

13.4.10.2.20.1 Offset

Register	Offset
lane0_tx_txcc_ctrl	4040h
lane1_tx_txcc_ctrl	4440h
lane2_tx_txcc_ctrl	4840h
lane3_tx_txcc_ctrl	4C40h

13.4.10.2.20.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_txcc_ctrl_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.20.3 Fields

Field	Function
15-0 tx_txcc_ctrl_15_0	Reserved

13.4.10.2.21 TX pre-cursor override register (lane0_tx_txcc_pre_ovrd - lane3_tx_txcc_pre_ovrd)

13.4.10.2.21.1 Offset

Register	Offset
lane0_tx_txcc_pre_ovrd	4041h
lane1_tx_txcc_pre_ovrd	4441h
lane2_tx_txcc_pre_ovrd	4841h
lane3_tx_txcc_pre_ovrd	4C41h

13.4.10.2.21.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_txcc_pre_ovrd_15_9							tx_txcc_pre_ovrd_8	tx_txcc_pre_ovrd_7_6		tx_txcc_pre_ovrd_5_0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.21.3 Fields

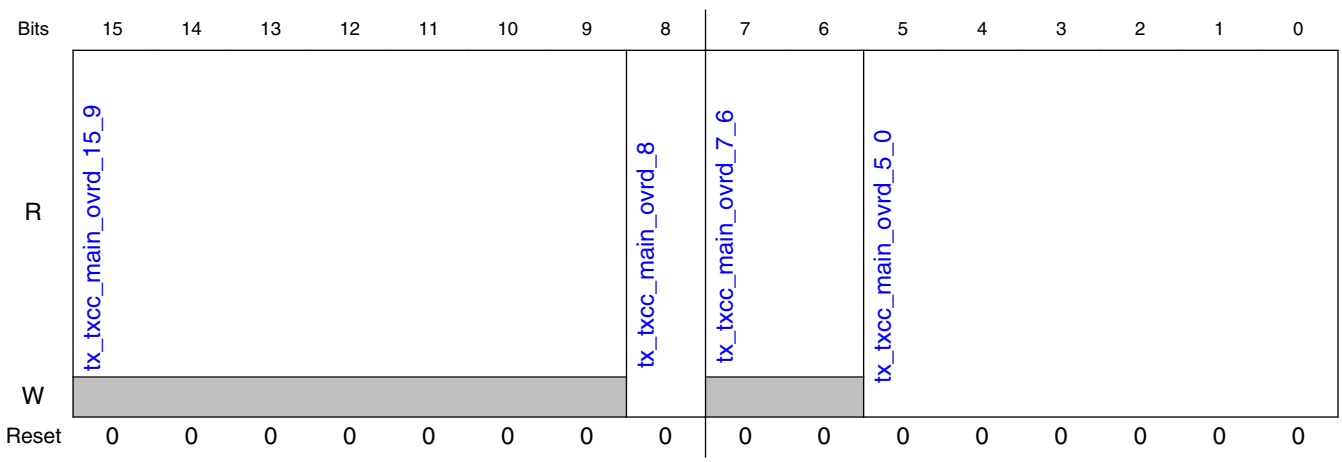
Field	Function
15-9 tx_txcc_pre_ovrd_15_9	Reserved
8 tx_txcc_pre_ovrd_8	Pre-cursor override enable: When enabled, the pre-cursor field in this register is used to override the pre-cursor value.
7-6 tx_txcc_pre_ovrd_7_6	Reserved
5-0 tx_txcc_pre_ovrd_5_0	Pre-cursor override value: When enabled by the pre-cursor override enable bit in this register, the value in this field is used to override the pre-cursor value. Note that this field is 6 bits wide, to match the pre-cursor data input width on the

13.4.10.2.22 TX main-cursor override register (lane0_tx_txcc_main_ovrd - lane3_tx_txcc_main_ovrd)

13.4.10.2.22.1 Offset

Register	Offset
lane0_tx_txcc_main_ovrd	4042h
lane1_tx_txcc_main_ovrd	4442h
lane2_tx_txcc_main_ovrd	4842h
lane3_tx_txcc_main_ovrd	4C42h

13.4.10.2.22.2 Diagram



13.4.10.2.22.3 Fields

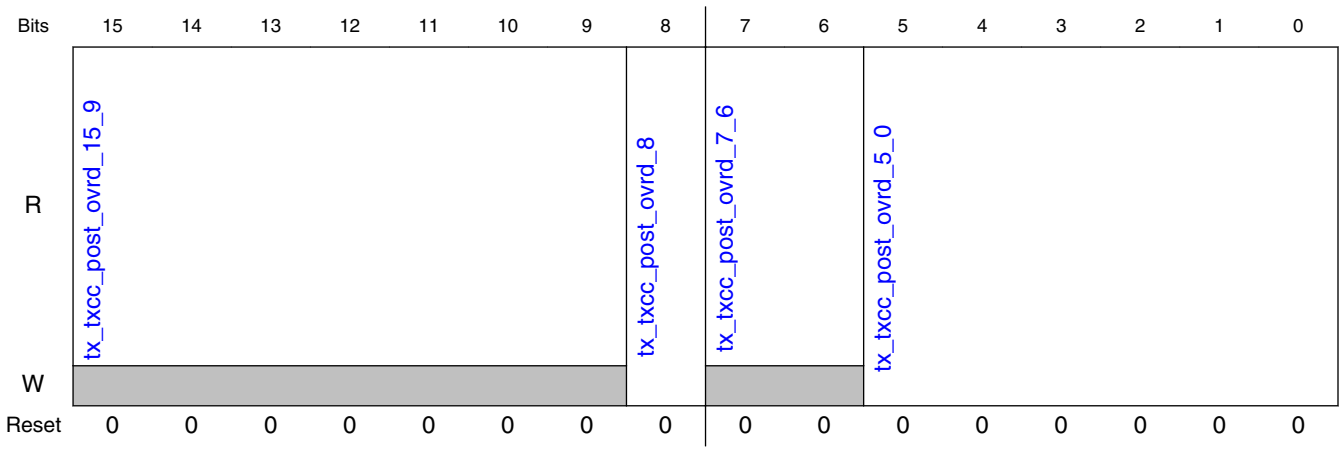
Field	Function
15-9 tx_txcc_main_ovrd_15_9	Reserved
8 tx_txcc_main_ovrd_8	Main-cursor override enable: When enabled, the main-cursor field in this register is used to override the main-cursor value.
7-6 tx_txcc_main_ovrd_7_6	Reserved
5-0 tx_txcc_main_ovrd_5_0	Main-cursor override value: When enabled by the main-cursor override enable bit in this register, the value in this field is used to override the main-cursor value. Note that this field is 6 bits wide, to match the main-cursor data input width on the

13.4.10.2.23 TX post-cursor override register (lane0_tx_txcc_post_ovrd - lane3_tx_txcc_post_ovrd)

13.4.10.2.23.1 Offset

Register	Offset
lane0_tx_txcc_post_ovrd	4043h
lane1_tx_txcc_post_ovrd	4443h
lane2_tx_txcc_post_ovrd	4843h
lane3_tx_txcc_post_ovrd	4C43h

13.4.10.2.23.2 Diagram



13.4.10.2.23.3 Fields

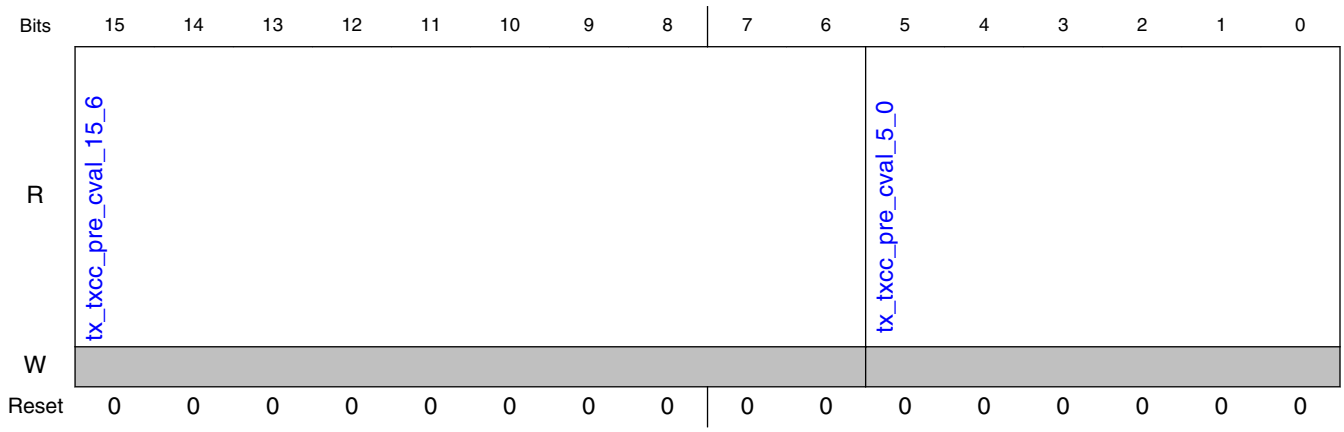
Field	Function
15-9 tx_txcc_post_ovrd_15_9	Reserved
8 tx_txcc_post_ovrd_8	Post-cursor override enable: When enabled, the post-cursor field in this register is used to override the post-cursor value.
7-6 tx_txcc_post_ovrd_7_6	Reserved
5-0 tx_txcc_post_ovrd_5_0	Post-cursor override value: When enabled by the post-cursor override enable bit in this register, the value in this field is used to override the post-cursor value. Note that this field is 6 bits wide, to match the post-cursor data input width on the

13.4.10.2.24 TX pre-cursor current value register (lane0_tx_txcc_pre_cval - lane3_tx_txcc_pre_cval)

13.4.10.2.24.1 Offset

Register	Offset
lane0_tx_txcc_pre_cval	4044h
lane1_tx_txcc_pre_cval	4444h
lane2_tx_txcc_pre_cval	4844h
lane3_tx_txcc_pre_cval	4C44h

13.4.10.2.24.2 Diagram



13.4.10.2.24.3 Fields

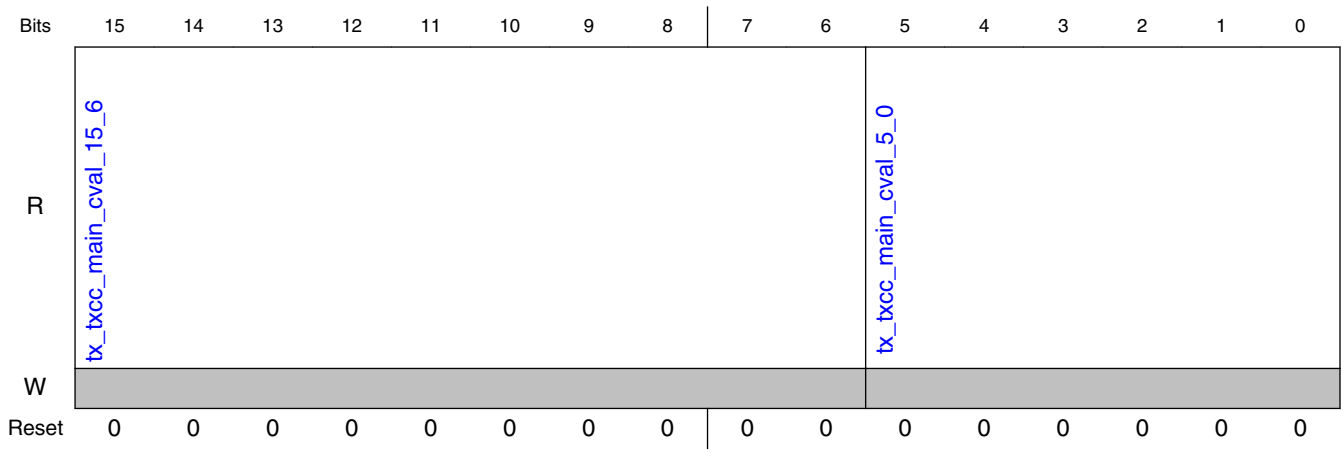
Field	Function
15-6 tx_txcc_pre_cval_15_6	Reserved
5-0 tx_txcc_pre_cval_5_0	Pre-cursor value: The value in this field indicates the current value of the pre-cursor (C-1) coefficient. The value of this field can be any of the following, depending on the current mode of operation.

13.4.10.2.25 TX main-cursor current value register (lane0_tx_txcc_main_cval - lane3_tx_txcc_main_cval)

13.4.10.2.25.1 Offset

Register	Offset
lane0_tx_txcc_main_cval	4045h
lane1_tx_txcc_main_cval	4445h
lane2_tx_txcc_main_cval	4845h
lane3_tx_txcc_main_cval	4C45h

13.4.10.2.25.2 Diagram



13.4.10.2.25.3 Fields

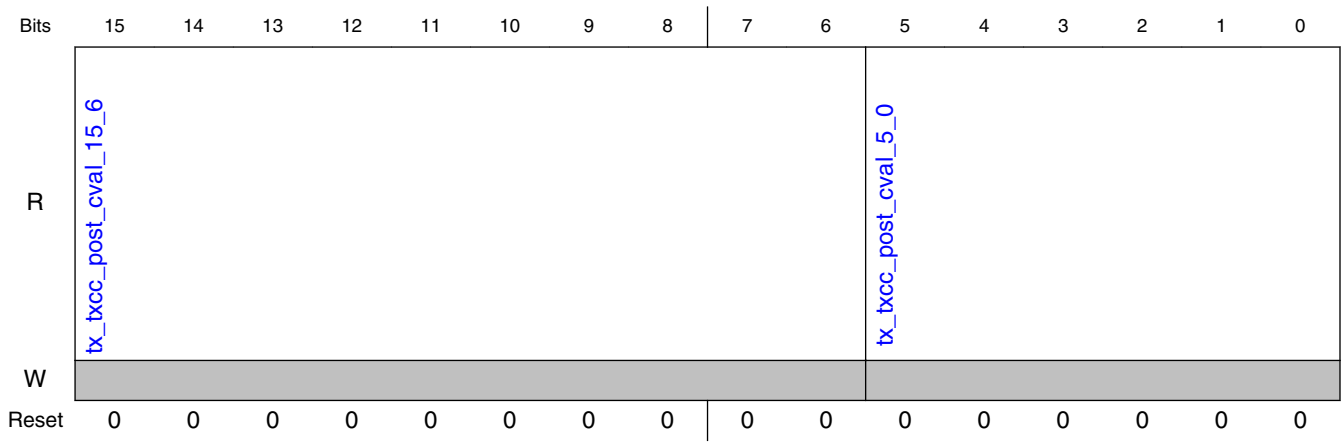
Field	Function
15-6 tx_txcc_main_cval_15_6	Reserved
5-0 tx_txcc_main_cval_5_0	Main-cursor value: The value in this field indicates the current value of the main-cursor (C0) coefficient. The value of this field can be any of the following, depending on the current mode of operation.

13.4.10.2.26 TX post-cursor current value register (lane0_tx_txcc_post_cval - lane3_tx_txcc_post_cval)

13.4.10.2.26.1 Offset

Register	Offset
lane0_tx_txcc_post_cval	4046h
lane1_tx_txcc_post_cval	4446h
lane2_tx_txcc_post_cval	4846h
lane3_tx_txcc_post_cval	4C46h

13.4.10.2.26.2 Diagram



13.4.10.2.26.3 Fields

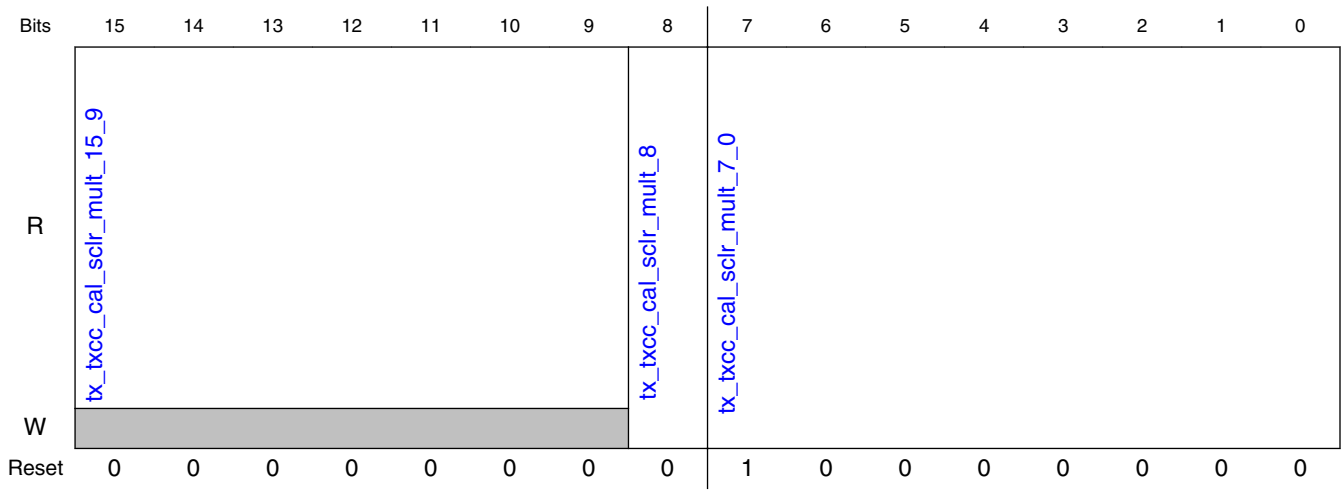
Field	Function
15-6 tx_txcc_post_cval_15_6	Reserved
5-0 tx_txcc_post_cval_5_0	Post-cursor value: The value in this field indicates the current value of the post-cursor (C+1) coefficient. The value of this field can be any of the following, depending on the current mode of operation.

13.4.10.2.27 Resistor calibration code scaler multiplier value register (lane 0_tx_txcc_cal_sclr_mult - lane3_tx_txcc_cal_sclr_mult)

13.4.10.2.27.1 Offset

Register	Offset
lane0_tx_txcc_cal_sclr_mult	4047h
lane1_tx_txcc_cal_sclr_mult	4447h
lane2_tx_txcc_cal_sclr_mult	4847h
lane3_tx_txcc_cal_sclr_mult	4C47h

13.4.10.2.27.2 Diagram



13.4.10.2.27.3 Fields

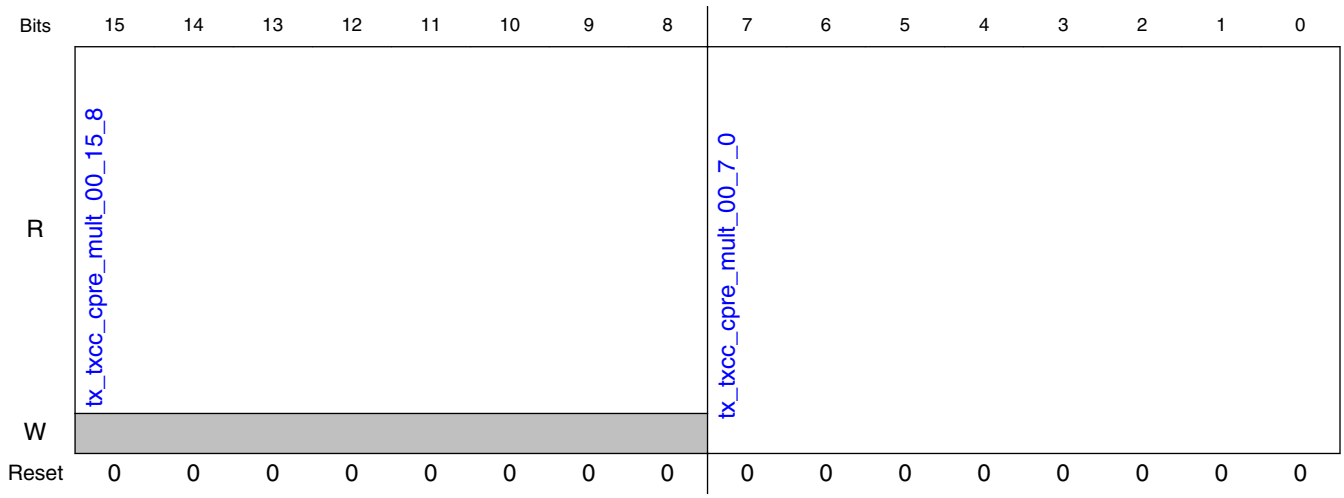
Field	Function
15-9 tx_txcc_cal_sclr_mult_15_9	Reserved
8 tx_txcc_cal_sclr_mult_8	Scaled resistor calibration code add: When this bit is set to
7-0 tx_txcc_cal_sclr_mult_7_0	Resistor calibration multiplier value: The value in this field specifies the multiplier value used to scale the resistor calibration code by. The following describes the multiplier value each bit in this field corresponds to.

13.4.10.2.28 Calculated pre emphasis multiplier value 00 register (lane0_tx_txcc_cpre_mult_00 - lane3_tx_txcc_cpre_mult_00)

13.4.10.2.28.1 Offset

Register	Offset
lane0_tx_txcc_cpre_mult_00	4048h
lane1_tx_txcc_cpre_mult_00	4448h
lane2_tx_txcc_cpre_mult_00	4848h
lane3_tx_txcc_cpre_mult_00	4C48h

13.4.10.2.28.2 Diagram



13.4.10.2.28.3 Fields

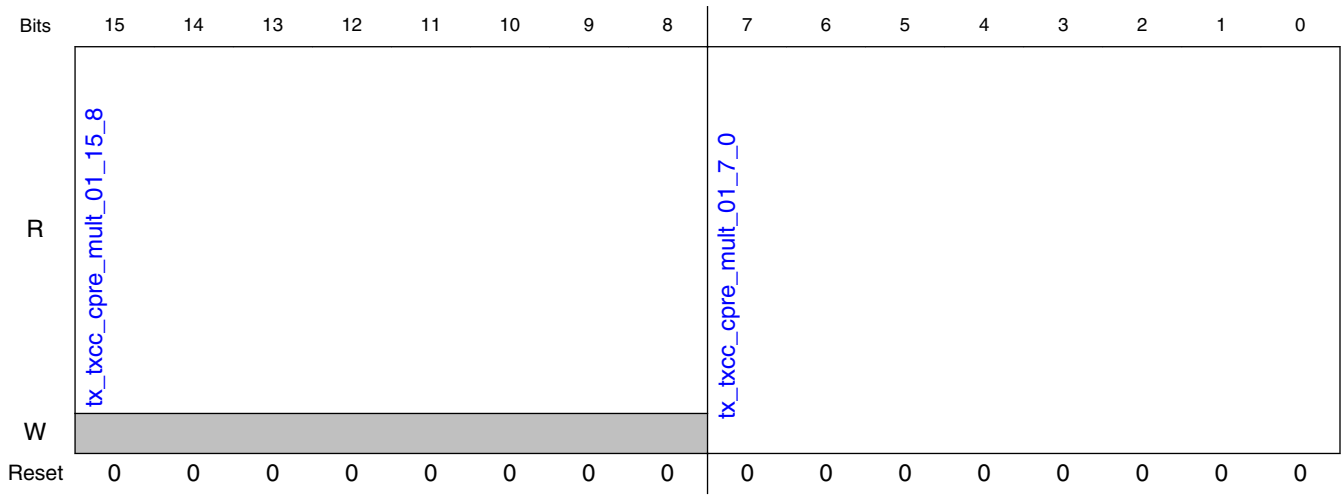
Field	Function
15-8 tx_txcc_cpre_mult_00_15_8	Reserved
7-0 tx_txcc_cpre_mult_00_7_0	Calculated pre emphasis multiplier value 00: The value in this field specifies the multiplier value used to generate the calculated pre emphasis value from the FS value when

13.4.10.2.29 Calculated pre emphasis multiplier value 01 register (lane0_tx_txcc_cpre_mult_01 - lane3_tx_txcc_cpre_mult_01)

13.4.10.2.29.1 Offset

Register	Offset
lane0_tx_txcc_cpre_mult_01	4049h
lane1_tx_txcc_cpre_mult_01	4449h
lane2_tx_txcc_cpre_mult_01	4849h
lane3_tx_txcc_cpre_mult_01	4C49h

13.4.10.2.29.2 Diagram



13.4.10.2.29.3 Fields

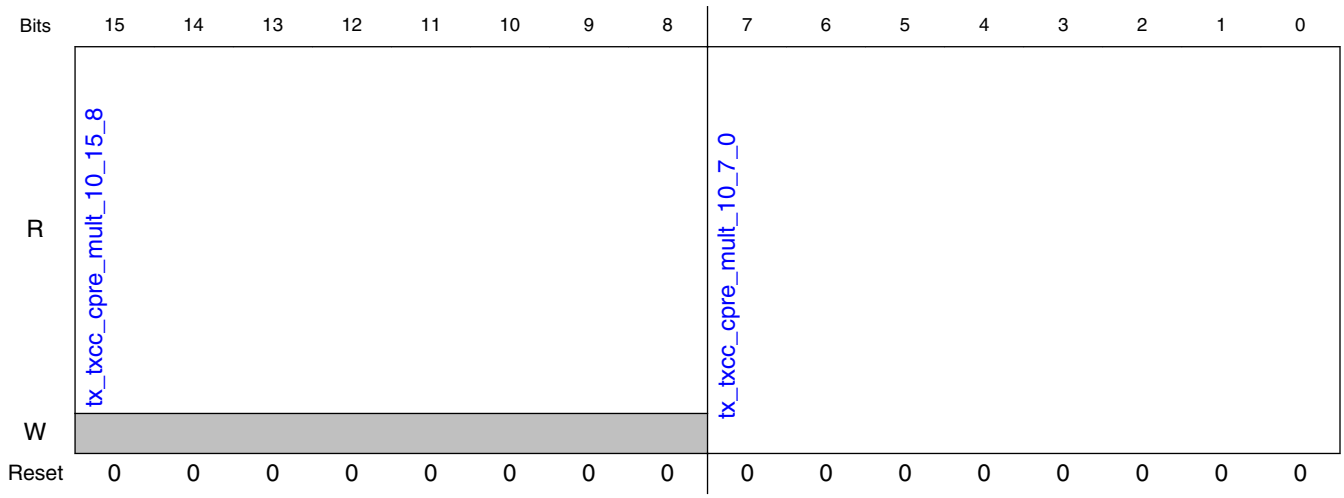
Field	Function
15-8 tx_txcc_cpre_mult_01_15_8	Reserved
7-0 tx_txcc_cpre_mult_01_7_0	Calculated pre emphasis multiplier value 01: The value in this field specifies the multiplier value used to generate the calculated pre emphasis value from the FS value when

13.4.10.2.30 Calculated pre emphasis multiplier value 10 register (lane0_tx_txcc_cpre_mult_10 - lane3_tx_txcc_cpre_mult_10)

13.4.10.2.30.1 Offset

Register	Offset
lane0_tx_txcc_cpre_mult_10	404Ah
lane1_tx_txcc_cpre_mult_10	444Ah
lane2_tx_txcc_cpre_mult_10	484Ah
lane3_tx_txcc_cpre_mult_10	4C4Ah

13.4.10.2.30.2 Diagram



13.4.10.2.30.3 Fields

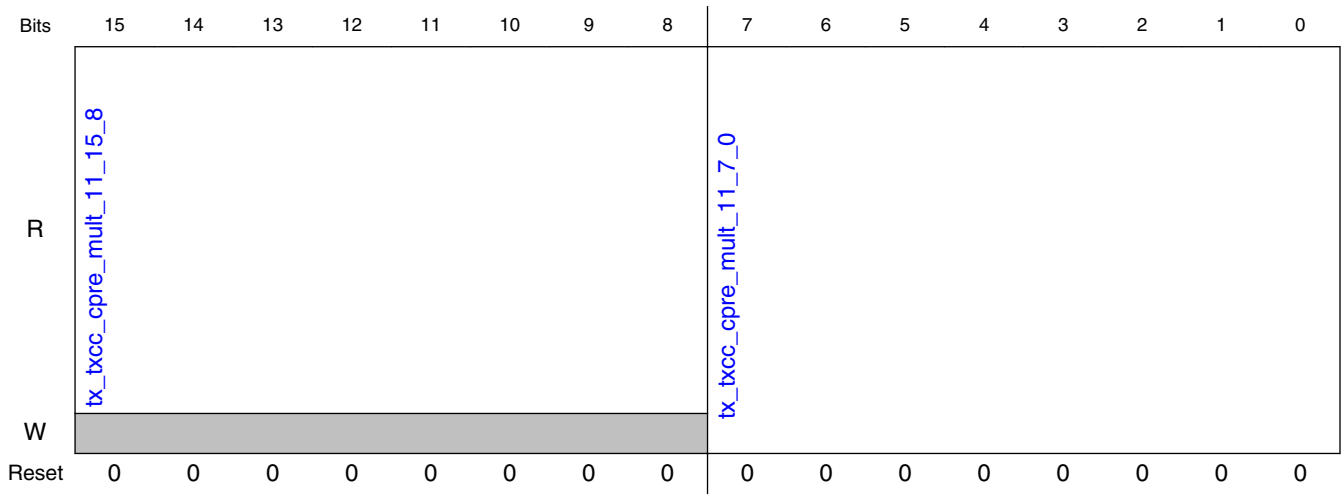
Field	Function
15-8 tx_txcc_cpre_mult_10_15_8	Reserved
7-0 tx_txcc_cpre_mult_10_7_0	Calculated pre emphasis multiplier value 10: The value in this field specifies the multiplier value used to generate the calculated pre emphasis value from the FS value when

13.4.10.2.31 Calculated pre emphasis multiplier value 11 register (lane0_tx_txcc_cpre_mult_11 - lane3_tx_txcc_cpre_mult_11)

13.4.10.2.31.1 Offset

Register	Offset
lane0_tx_txcc_cpre_mult_11	404Bh
lane1_tx_txcc_cpre_mult_11	444Bh
lane2_tx_txcc_cpre_mult_11	484Bh
lane3_tx_txcc_cpre_mult_11	4C4Bh

13.4.10.2.31.2 Diagram



13.4.10.2.31.3 Fields

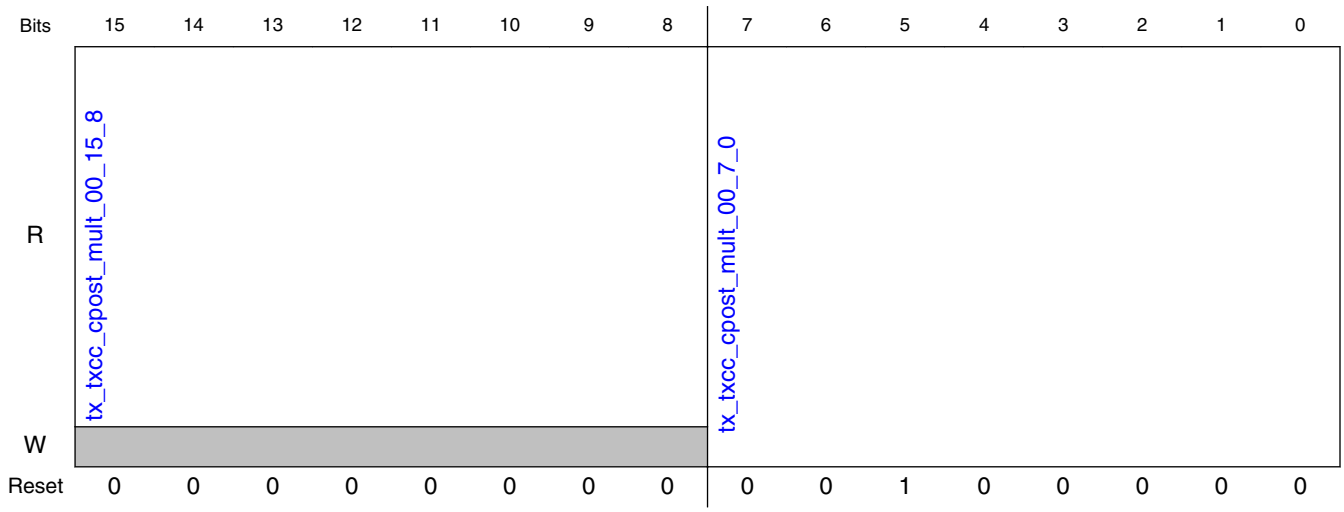
Field	Function
15-8 tx_txcc_cpre_mult_11_15_8	Reserved
7-0 tx_txcc_cpre_mult_11_7_0	Calculated pre emphasis multiplier value 11: The value in this field specifies the multiplier value used to generate the calculated pre emphasis value from the FS value when

13.4.10.2.32 Calculated post emphasis multiplier value 00 register (lane0_tx_txcc_cpost_mult_00 - lane3_tx_txcc_cpost_mult_00)

13.4.10.2.32.1 Offset

Register	Offset
lane0_tx_txcc_cpost_mult_00	404Ch
lane1_tx_txcc_cpost_mult_00	444Ch
lane2_tx_txcc_cpost_mult_00	484Ch
lane3_tx_txcc_cpost_mult_00	4C4Ch

13.4.10.2.32.2 Diagram



13.4.10.2.32.3 Fields

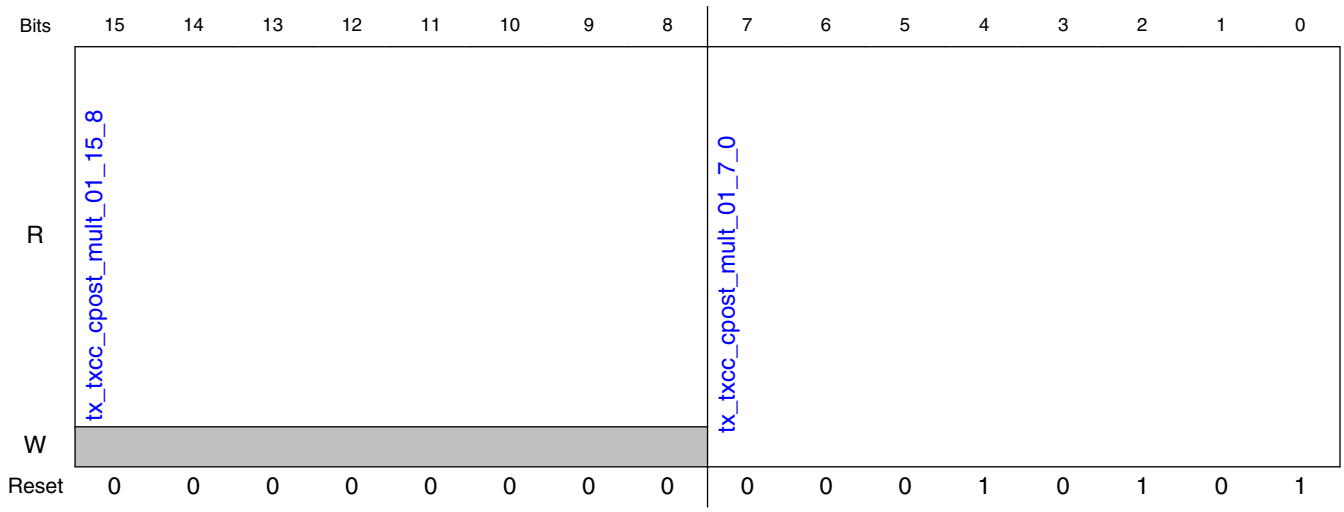
Field	Function
15-8 tx_txcc_cpost_mult_00_15_8	Reserved
7-0 tx_txcc_cpost_mult_00_7_0	Calculated post emphasis multiplier value 00: The value in this field specifies the multiplier value used to generate the calculated post emphasis value from the FS value when

13.4.10.2.33 Calculated post emphasis multiplier value 01 register (lane0_tx_txcc_cpost_mult_01 - lane3_tx_txcc_cpost_mult_01)

13.4.10.2.33.1 Offset

Register	Offset
lane0_tx_txcc_cpost_mult_01	404Dh
lane1_tx_txcc_cpost_mult_01	444Dh
lane2_tx_txcc_cpost_mult_01	484Dh
lane3_tx_txcc_cpost_mult_01	4C4Dh

13.4.10.2.33.2 Diagram



13.4.10.2.33.3 Fields

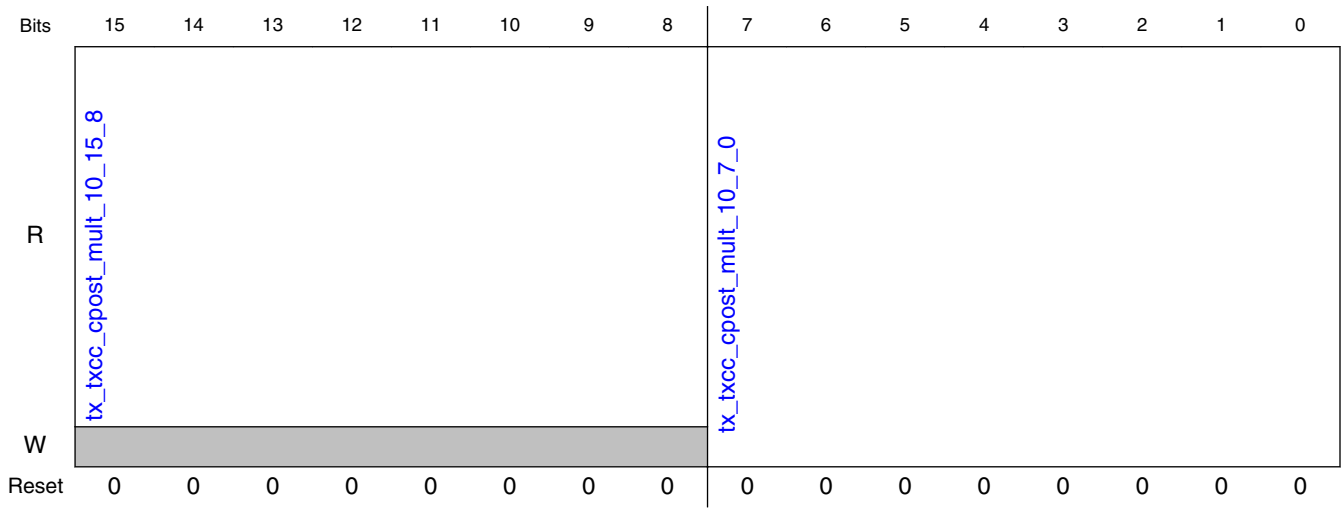
Field	Function
15-8 tx_txcc_cpost_mult_01_15_8	Reserved
7-0 tx_txcc_cpost_mult_01_7_0	Calculated post emphasis multiplier value 01: The value in this field specifies the multiplier value used to generate the calculated post emphasis value from the FS value when

13.4.10.2.34 Calculated post emphasis multiplier value 10 register (lane0_tx_txcc_cpost_mult_10 - lane3_tx_txcc_cpost_mult_10)

13.4.10.2.34.1 Offset

Register	Offset
lane0_tx_txcc_cpost_mult_10	404Eh
lane1_tx_txcc_cpost_mult_10	444Eh
lane2_tx_txcc_cpost_mult_10	484Eh
lane3_tx_txcc_cpost_mult_10	4C4Eh

13.4.10.2.34.2 Diagram



13.4.10.2.34.3 Fields

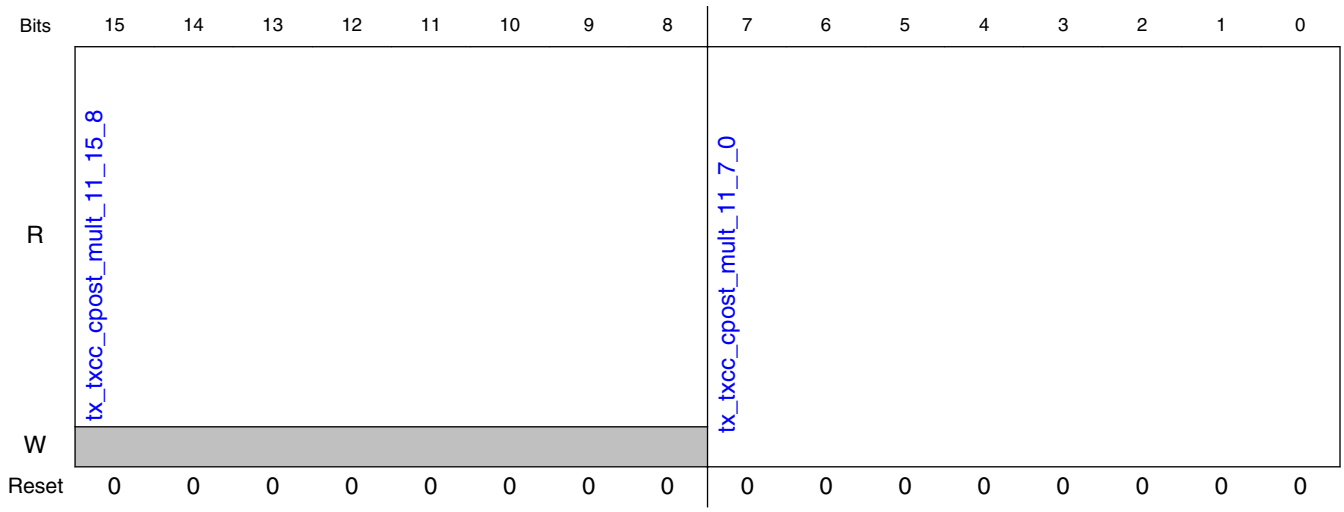
Field	Function
15-8 tx_txcc_cpost_mult_10_15_8	Reserved
7-0 tx_txcc_cpost_mult_10_7_0	Calculated post emphasis multiplier value 10: The value in this field specifies the multiplier value used to generate the calculated post emphasis value from the FS value when

13.4.10.2.35 Calculated post emphasis multiplier value 11 register (lane0_tx_txcc_cpost_mult_11 - lane3_tx_txcc_cpost_mult_11)

13.4.10.2.35.1 Offset

Register	Offset
lane0_tx_txcc_cpost_mult_11	404Fh
lane1_tx_txcc_cpost_mult_11	444Fh
lane2_tx_txcc_cpost_mult_11	484Fh
lane3_tx_txcc_cpost_mult_11	4C4Fh

13.4.10.2.35.2 Diagram



13.4.10.2.35.3 Fields

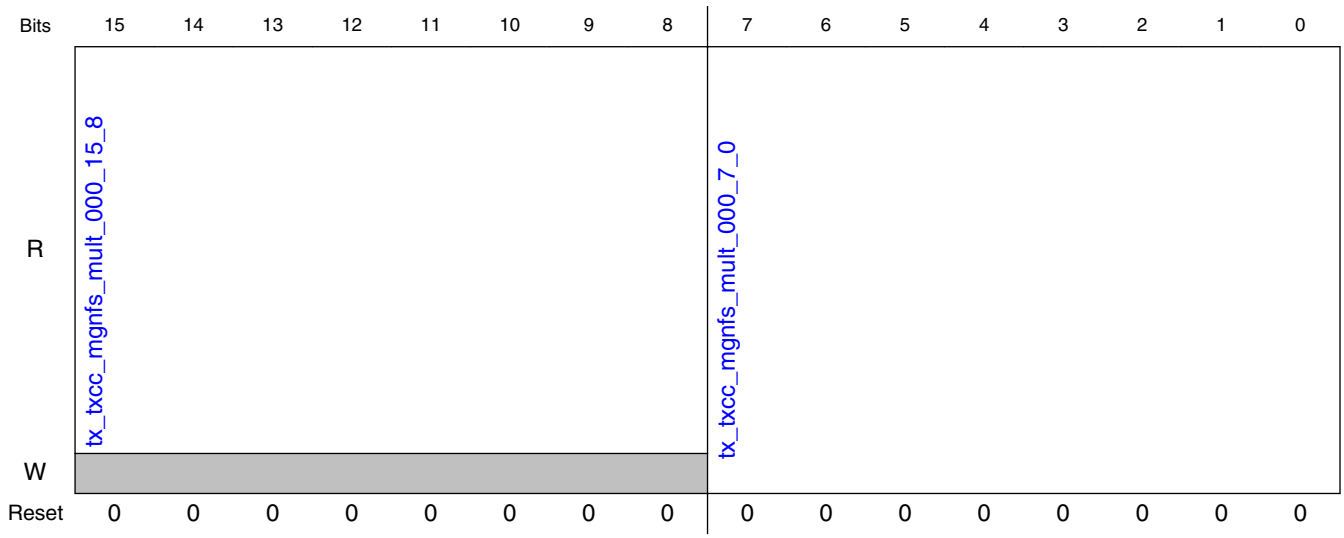
Field	Function
15-8 tx_txcc_cpost_mult_11_15_8	Reserved
7-0 tx_txcc_cpost_mult_11_7_0	Calculated post emphasis multiplier value 11: The value in this field specifies the multiplier value used to generate the calculated post emphasis value from the FS value when

13.4.10.2.36 Margin full swing multiplier value 000 register (lane0_tx_txcc_mgnfs_mult_000 - lane3_tx_txcc_mgnfs_mult_000)

13.4.10.2.36.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_000	4050h
lane1_tx_txcc_mgnfs_mult_000	4450h
lane2_tx_txcc_mgnfs_mult_000	4850h
lane3_tx_txcc_mgnfs_mult_000	4C50h

13.4.10.2.36.2 Diagram



13.4.10.2.36.3 Fields

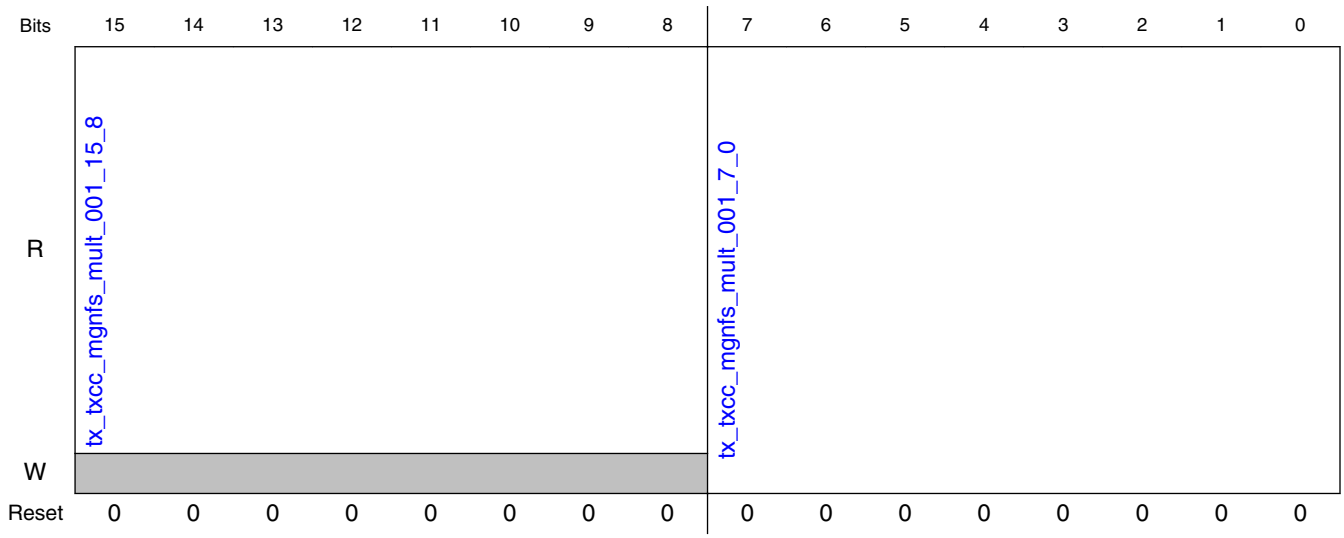
Field	Function
15-8 tx_txcc_mgnfs_mult_000_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_000_7_0	Margin full swing multiplier value 000: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.37 Margin full swing multiplier value 001 register (lane0_tx_txcc_mgnfs_mult_001 - lane3_tx_txcc_mgnfs_mult_001)

13.4.10.2.37.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_001	4051h
lane1_tx_txcc_mgnfs_mult_001	4451h
lane2_tx_txcc_mgnfs_mult_001	4851h
lane3_tx_txcc_mgnfs_mult_001	4C51h

13.4.10.2.37.2 Diagram



13.4.10.2.37.3 Fields

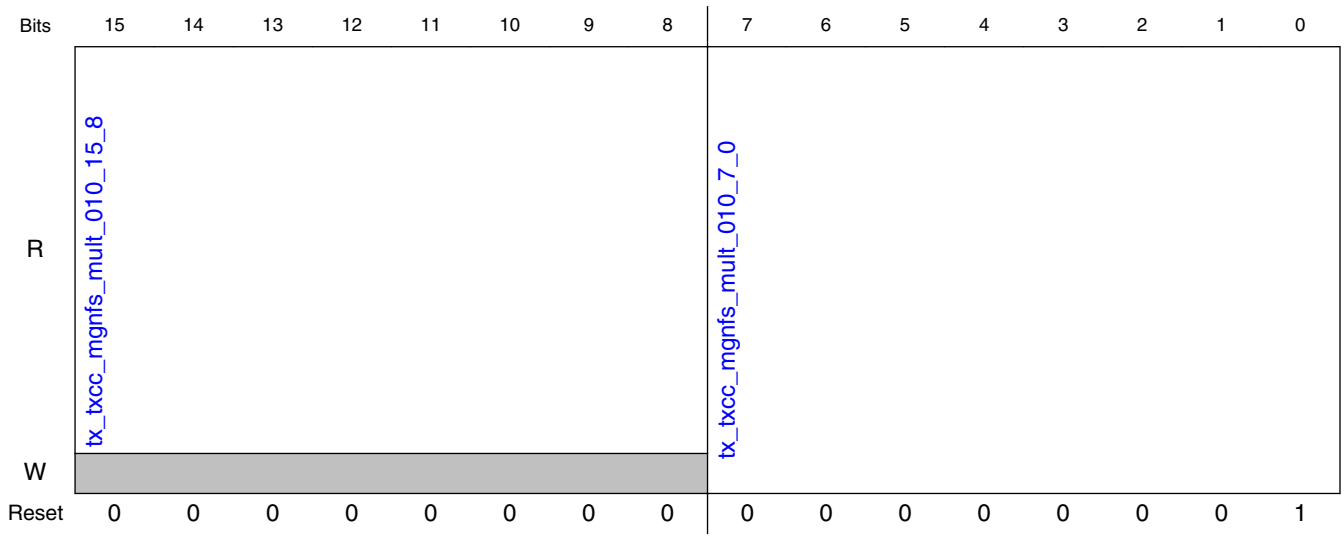
Field	Function
15-8 tx_txcc_mgnfs_mult_001_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_001_7_0	Margin full swing multiplier value 001: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.38 Margin full swing multiplier value 010 register (lane0_tx_txcc_mgnfs_mult_010 - lane3_tx_txcc_mgnfs_mult_010)

13.4.10.2.38.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_010	4052h
lane1_tx_txcc_mgnfs_mult_010	4452h
lane2_tx_txcc_mgnfs_mult_010	4852h
lane3_tx_txcc_mgnfs_mult_010	4C52h

13.4.10.2.38.2 Diagram



13.4.10.2.38.3 Fields

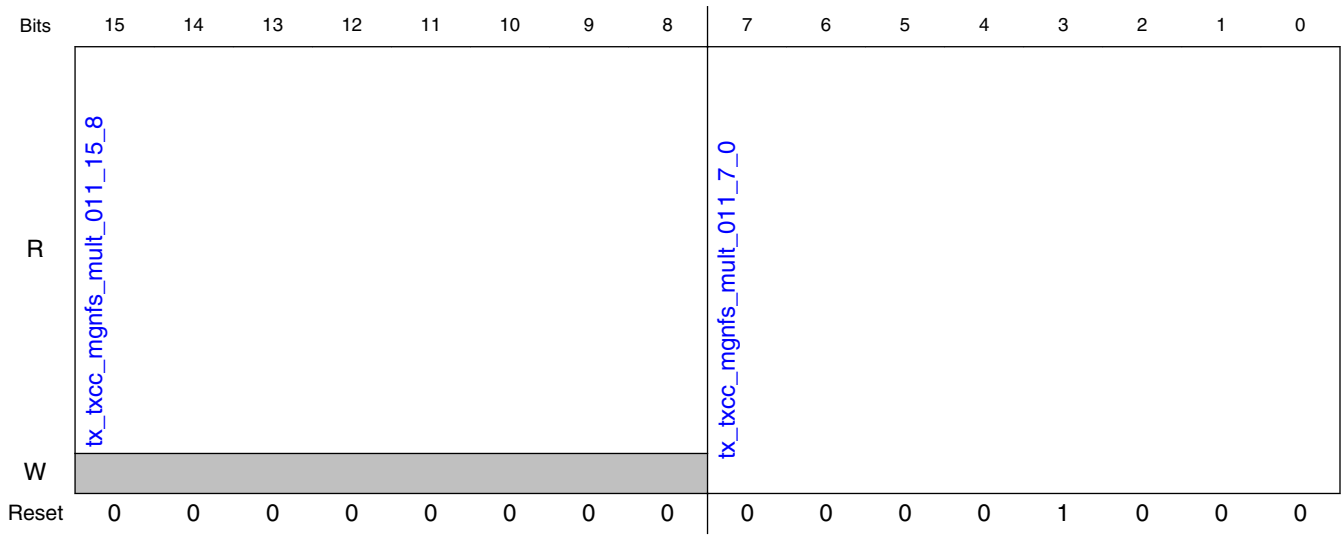
Field	Function
15-8 tx_txcc_mgnfs_mult_010_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_010_7_0	Margin full swing multiplier value 010: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.39 Margin full swing multiplier value 011 register (lane0_tx_txcc_mgnfs_mult_011 - lane3_tx_txcc_mgnfs_mult_011)

13.4.10.2.39.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_011	4053h
lane1_tx_txcc_mgnfs_mult_011	4453h
lane2_tx_txcc_mgnfs_mult_011	4853h
lane3_tx_txcc_mgnfs_mult_011	4C53h

13.4.10.2.39.2 Diagram



13.4.10.2.39.3 Fields

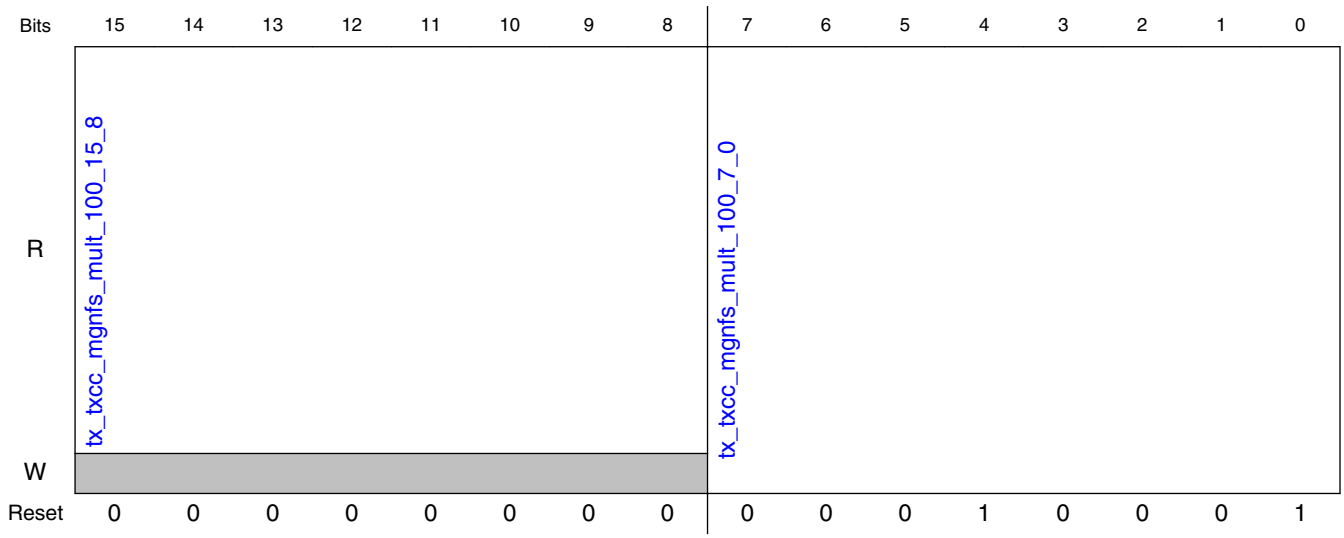
Field	Function
15-8 tx_txcc_mgnfs_mult_011_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_011_7_0	Margin full swing multiplier value 011: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.40 Margin full swing multiplier value 100 register (lane0_tx_txcc_mgnfs_mult_100 - lane3_tx_txcc_mgnfs_mult_100)

13.4.10.2.40.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_100	4054h
lane1_tx_txcc_mgnfs_mult_100	4454h
lane2_tx_txcc_mgnfs_mult_100	4854h
lane3_tx_txcc_mgnfs_mult_100	4C54h

13.4.10.2.40.2 Diagram



13.4.10.2.40.3 Fields

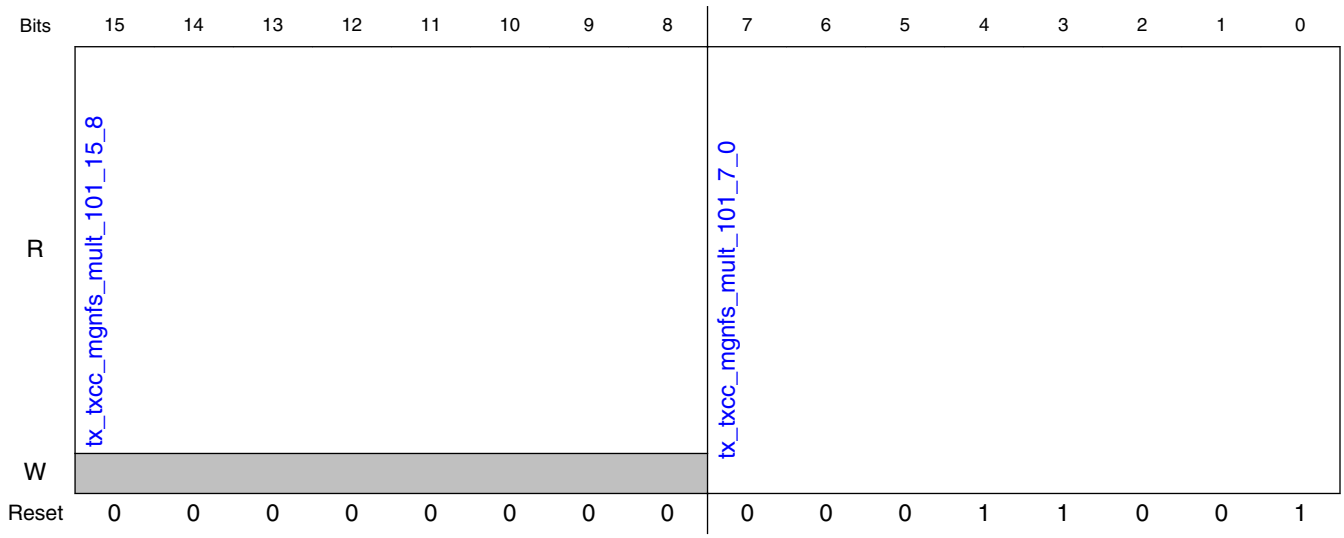
Field	Function
15-8 tx_txcc_mgnfs_mult_100_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_100_7_0	Margin full swing multiplier value 100: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.41 Margin full swing multiplier value 101 register (lane0_tx_txcc_mgnfs_mult_101 - lane3_tx_txcc_mgnfs_mult_101)

13.4.10.2.41.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_101	4055h
lane1_tx_txcc_mgnfs_mult_101	4455h
lane2_tx_txcc_mgnfs_mult_101	4855h
lane3_tx_txcc_mgnfs_mult_101	4C55h

13.4.10.2.41.2 Diagram



13.4.10.2.41.3 Fields

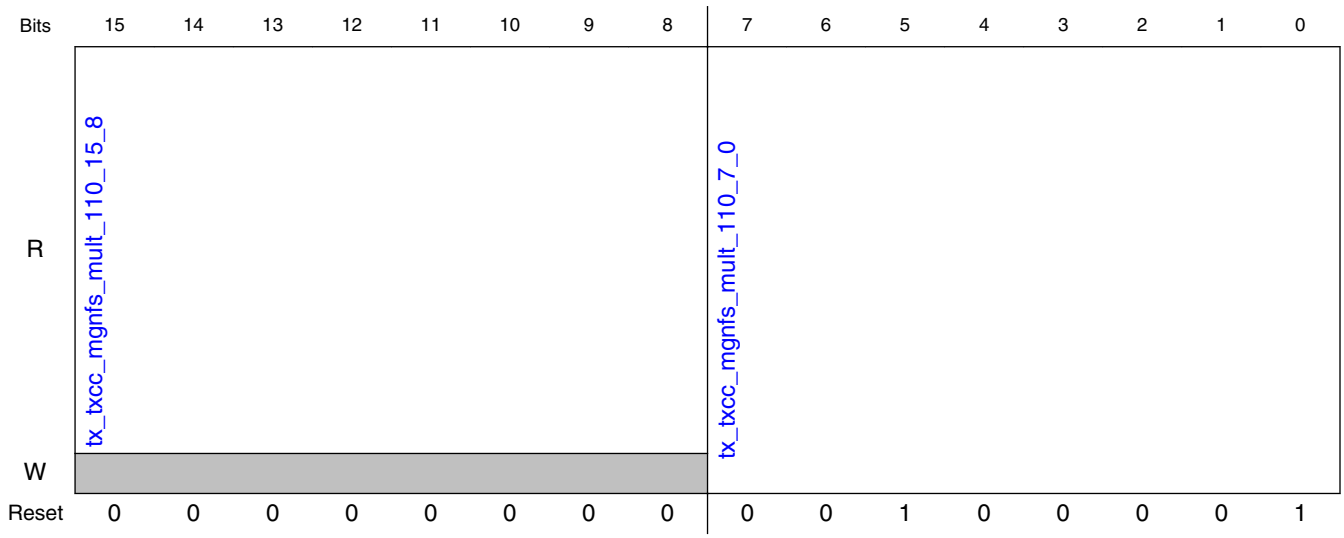
Field	Function
15-8 tx_txcc_mgnfs_mult_101_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_101_7_0	Margin full swing multiplier value 101: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.42 Margin full swing multiplier value 110 register (lane0_tx_txcc_mgnfs_mult_110 - lane3_tx_txcc_mgnfs_mult_110)

13.4.10.2.42.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_110	4056h
lane1_tx_txcc_mgnfs_mult_110	4456h
lane2_tx_txcc_mgnfs_mult_110	4856h
lane3_tx_txcc_mgnfs_mult_110	4C56h

13.4.10.2.42.2 Diagram



13.4.10.2.42.3 Fields

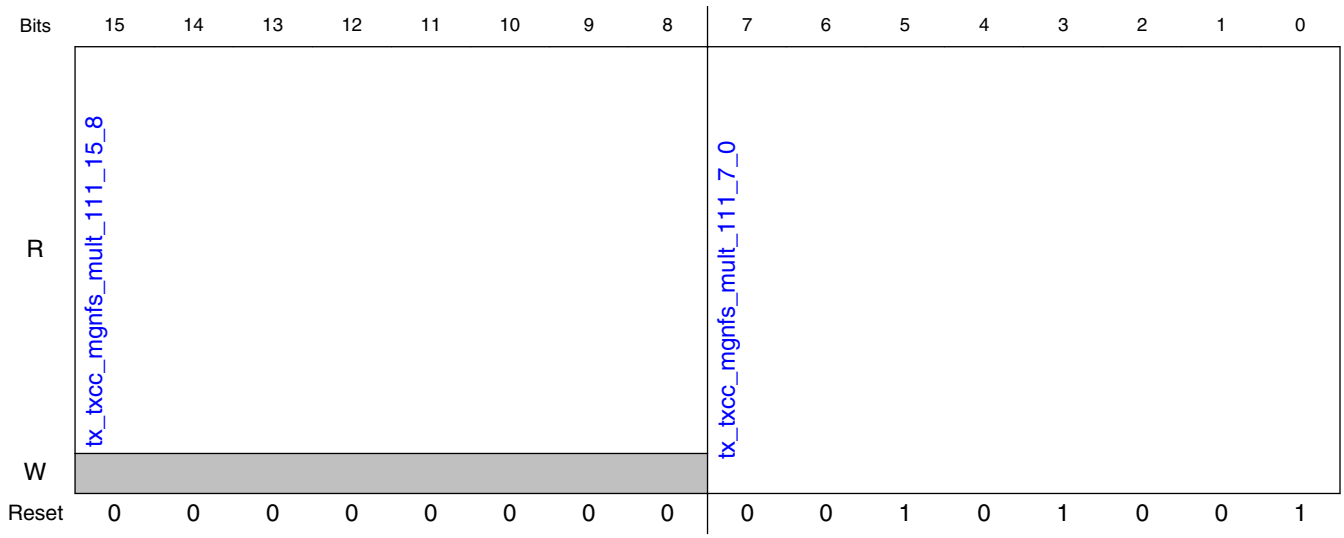
Field	Function
15-8 tx_txcc_mgnfs_mult_110_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_110_7_0	Margin full swing multiplier value 110: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.43 Margin full swing multiplier value 111 register (lane0_tx_txcc_mgnfs_mult_111 - lane3_tx_txcc_mgnfs_mult_111)

13.4.10.2.43.1 Offset

Register	Offset
lane0_tx_txcc_mgnfs_mult_111	4057h
lane1_tx_txcc_mgnfs_mult_111	4457h
lane2_tx_txcc_mgnfs_mult_111	4857h
lane3_tx_txcc_mgnfs_mult_111	4C57h

13.4.10.2.43.2 Diagram



13.4.10.2.43.3 Fields

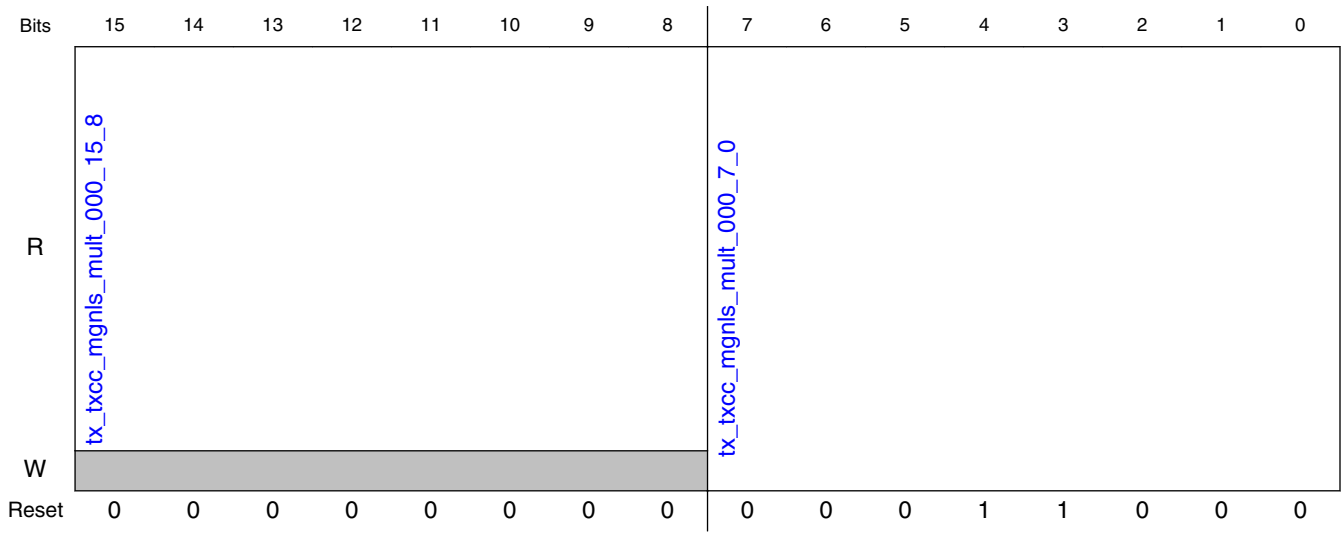
Field	Function
15-8 tx_txcc_mgnfs_mult_111_15_8	Reserved
7-0 tx_txcc_mgnfs_mult_111_7_0	Margin full swing multiplier value 111: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.44 Margin half swing multiplier value 000 register (lane0_tx_txcc_mgnls_mult_000 - lane3_tx_txcc_mgnls_mult_000)

13.4.10.2.44.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_000	4058h
lane1_tx_txcc_mgnls_mult_000	4458h
lane2_tx_txcc_mgnls_mult_000	4858h
lane3_tx_txcc_mgnls_mult_000	4C58h

13.4.10.2.44.2 Diagram



13.4.10.2.44.3 Fields

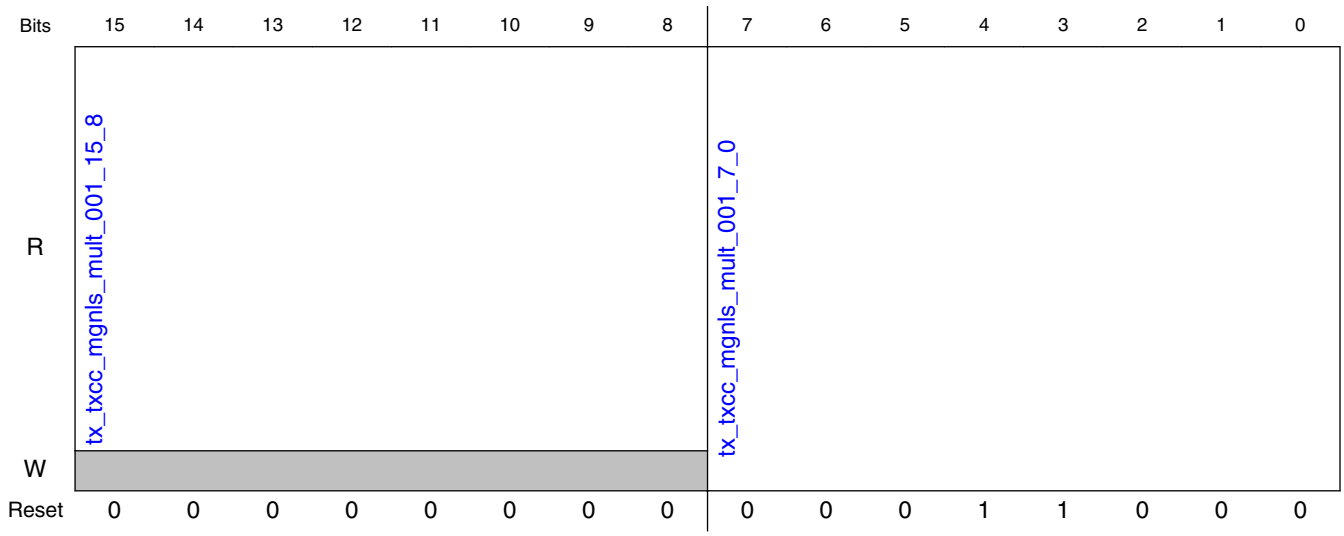
Field	Function
15-8 tx_txcc_mgnls_mult_000_15_8	Reserved
7-0 tx_txcc_mgnls_mult_000_7_0	Margin half swing multiplier value 000: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.45 Margin half swing multiplier value 001 register (lane0_tx_txcc_mgnls_mult_001 - lane3_tx_txcc_mgnls_mult_001)

13.4.10.2.45.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_001	4059h
lane1_tx_txcc_mgnls_mult_001	4459h
lane2_tx_txcc_mgnls_mult_001	4859h
lane3_tx_txcc_mgnls_mult_001	4C59h

13.4.10.2.45.2 Diagram



13.4.10.2.45.3 Fields

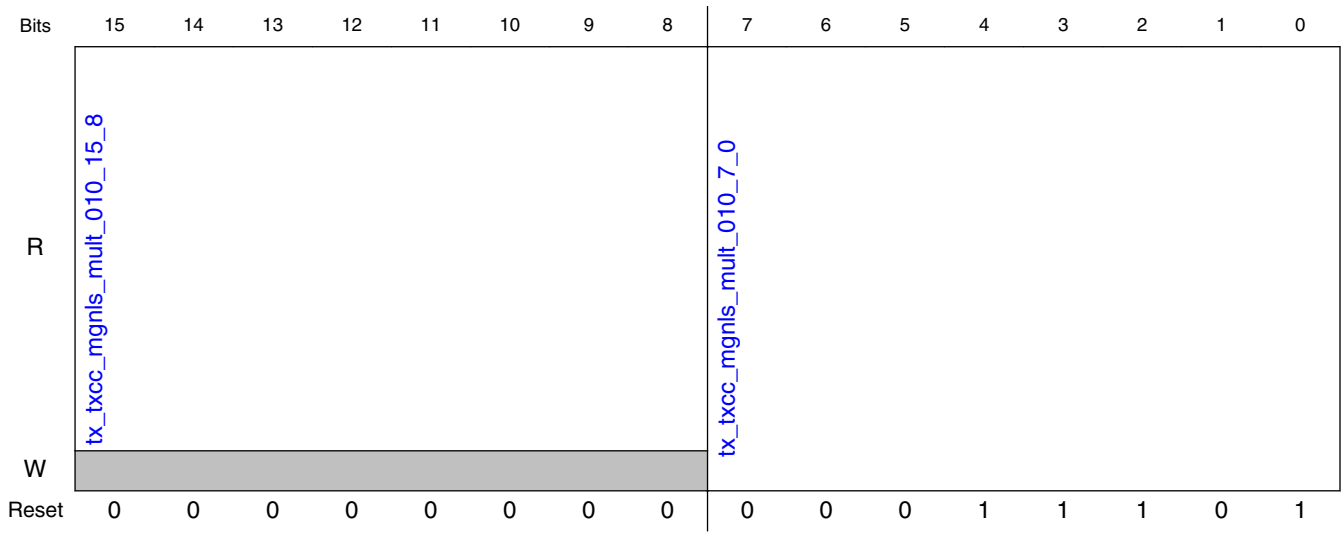
Field	Function
15-8 tx_txcc_mgnls_mult_001_15_8	Reserved
7-0 tx_txcc_mgnls_mult_001_7_0	Margin half swing multiplier value 001: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.46 Margin half swing multiplier value 010 register (lane0_tx_txcc_mgnls_mult_010 - lane3_tx_txcc_mgnls_mult_010)

13.4.10.2.46.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_010	405Ah
lane1_tx_txcc_mgnls_mult_010	445Ah
lane2_tx_txcc_mgnls_mult_010	485Ah
lane3_tx_txcc_mgnls_mult_010	4C5Ah

13.4.10.2.46.2 Diagram



13.4.10.2.46.3 Fields

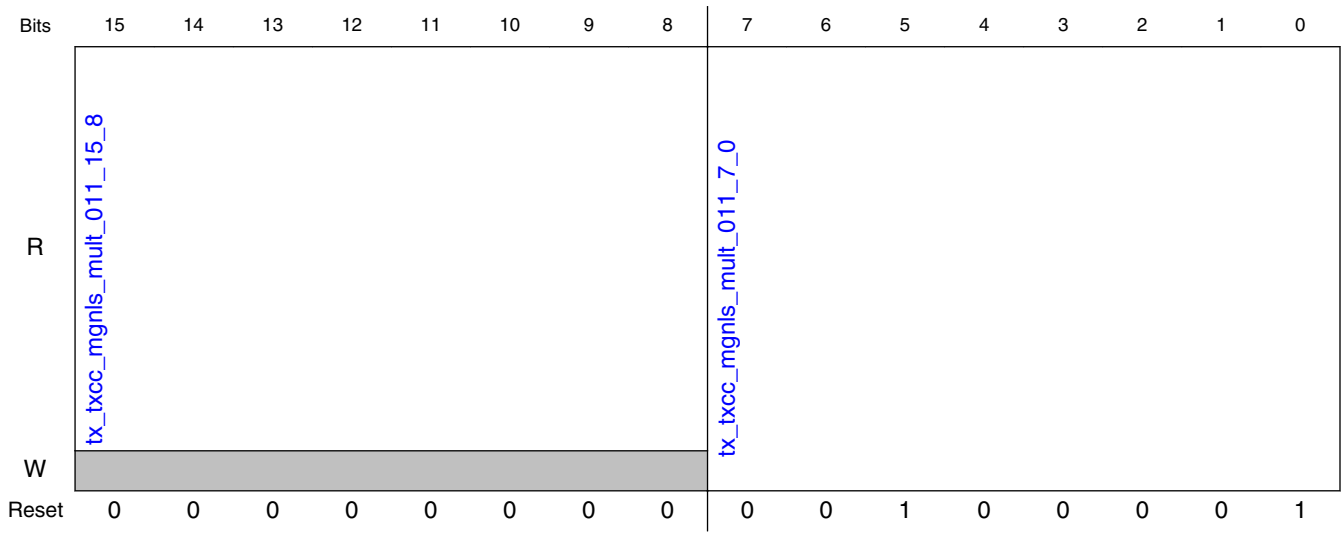
Field	Function
15-8 tx_txcc_mgnls_mult_010_15_8	Reserved
7-0 tx_txcc_mgnls_mult_010_7_0	Margin half swing multiplier value 010: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.47 Margin half swing multiplier value 011 register (lane0_tx_txcc_mgnls_mult_011 - lane3_tx_txcc_mgnls_mult_011)

13.4.10.2.47.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_011	405Bh
lane1_tx_txcc_mgnls_mult_011	445Bh
lane2_tx_txcc_mgnls_mult_011	485Bh
lane3_tx_txcc_mgnls_mult_011	4C5Bh

13.4.10.2.47.2 Diagram



13.4.10.2.47.3 Fields

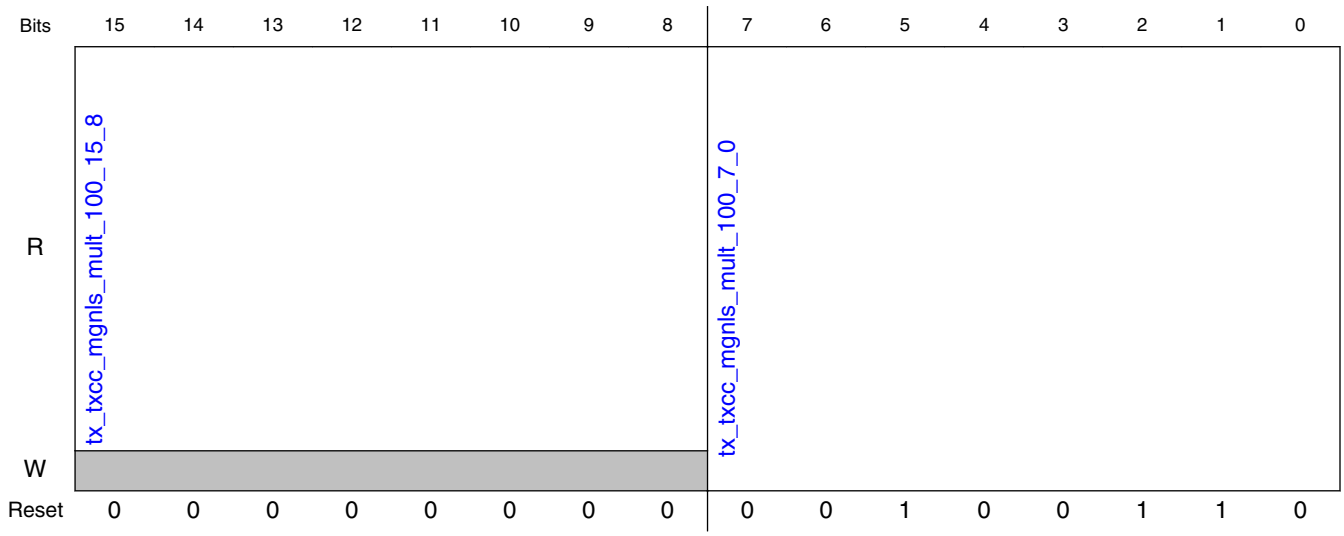
Field	Function
15-8 tx_txcc_mgnls_mult_011_15_8	Reserved
7-0 tx_txcc_mgnls_mult_011_7_0	Margin half swing multiplier value 011: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.48 Margin half swing multiplier value 100 register (lane0_tx_txcc_mgnls_mult_100 - lane3_tx_txcc_mgnls_mult_100)

13.4.10.2.48.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_100	405Ch
lane1_tx_txcc_mgnls_mult_100	445Ch
lane2_tx_txcc_mgnls_mult_100	485Ch
lane3_tx_txcc_mgnls_mult_100	4C5Ch

13.4.10.2.48.2 Diagram



13.4.10.2.48.3 Fields

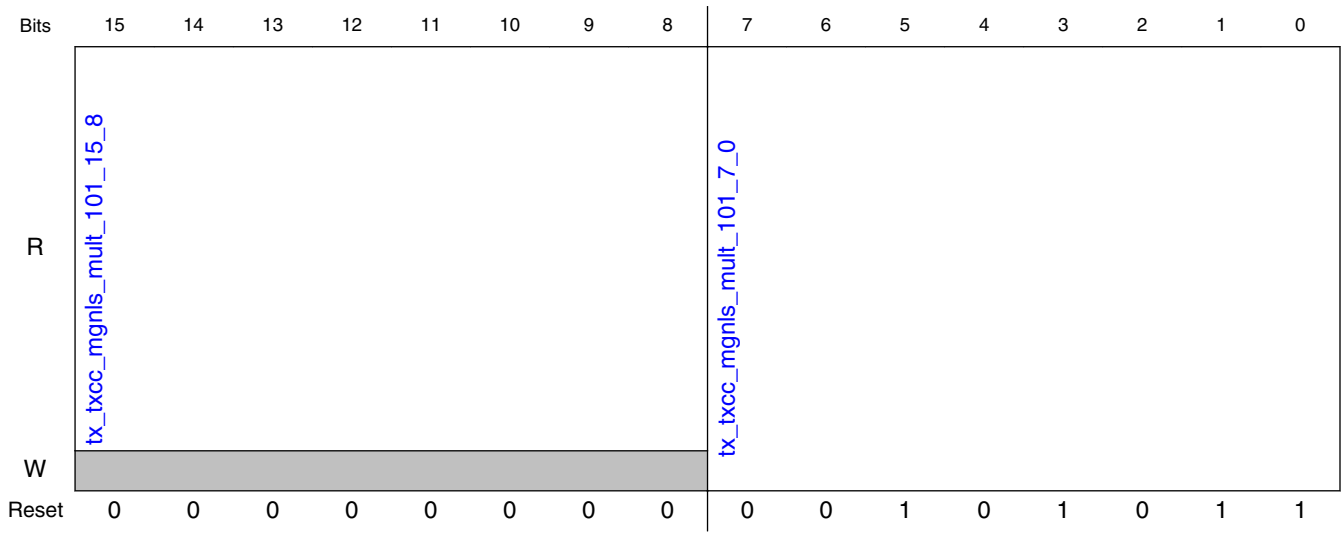
Field	Function
15-8 tx_txcc_mgnls_mult_100_15_8	Reserved
7-0 tx_txcc_mgnls_mult_100_7_0	Margin half swing multiplier value 100: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.49 Margin half swing multiplier value 101 register (lane0_tx_txcc_mgnls_mult_101 - lane3_tx_txcc_mgnls_mult_101)

13.4.10.2.49.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_101	405Dh
lane1_tx_txcc_mgnls_mult_101	445Dh
lane2_tx_txcc_mgnls_mult_101	485Dh
lane3_tx_txcc_mgnls_mult_101	4C5Dh

13.4.10.2.49.2 Diagram



13.4.10.2.49.3 Fields

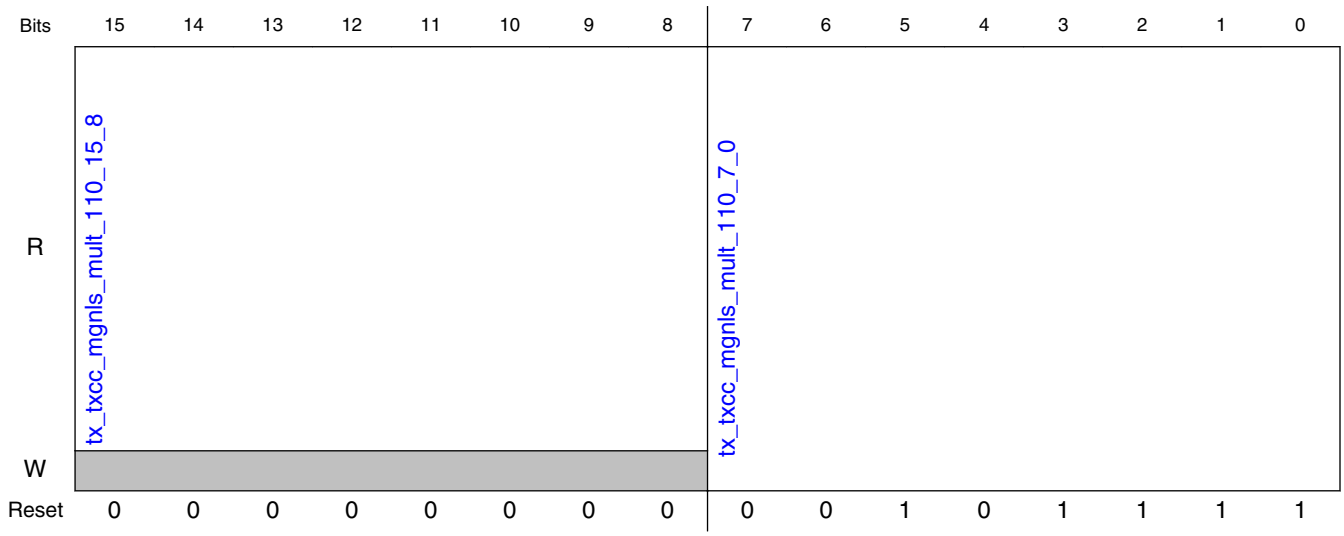
Field	Function
15-8 tx_txcc_mgnls_mult_101_15_8	Reserved
7-0 tx_txcc_mgnls_mult_101_7_0	Margin half swing multiplier value 101: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.50 Margin half swing multiplier value 110 register (lane0_tx_txcc_mgnls_mult_110 - lane3_tx_txcc_mgnls_mult_110)

13.4.10.2.50.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_110	405Eh
lane1_tx_txcc_mgnls_mult_110	445Eh
lane2_tx_txcc_mgnls_mult_110	485Eh
lane3_tx_txcc_mgnls_mult_110	4C5Eh

13.4.10.2.50.2 Diagram



13.4.10.2.50.3 Fields

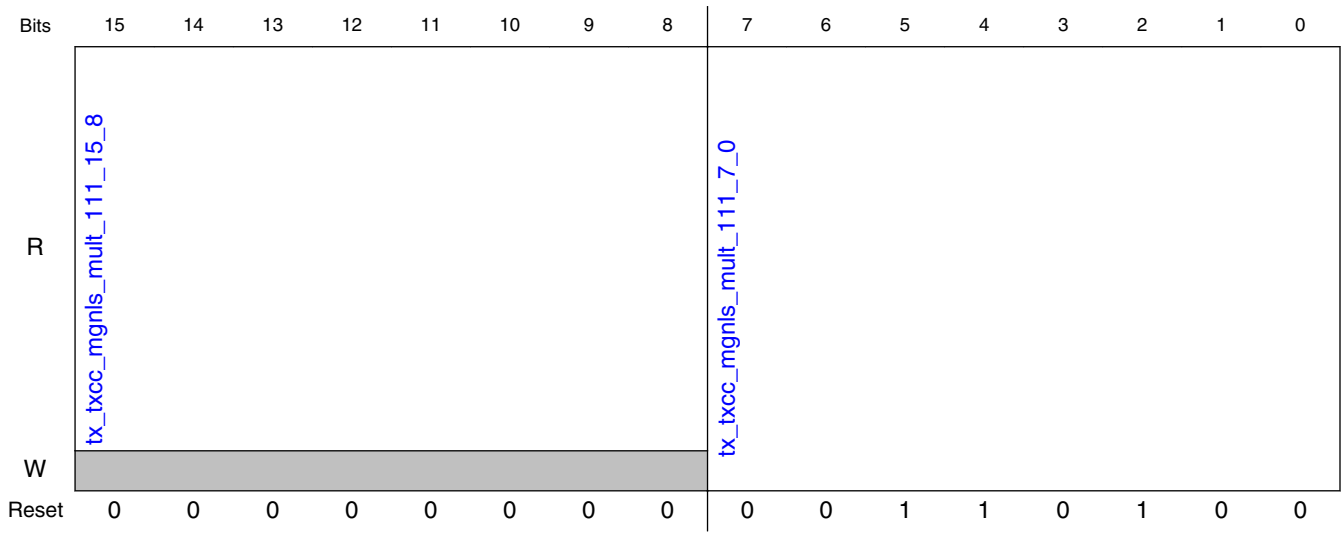
Field	Function
15-8 tx_txcc_mgnls_mult_110_15_8	Reserved
7-0 tx_txcc_mgnls_mult_110_7_0	Margin half swing multiplier value 110: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.51 Margin half swing multiplier value 111 register (lane0_tx_txcc_mgnls_mult_111 - lane3_tx_txcc_mgnls_mult_111)

13.4.10.2.51.1 Offset

Register	Offset
lane0_tx_txcc_mgnls_mult_111	405Fh
lane1_tx_txcc_mgnls_mult_111	445Fh
lane2_tx_txcc_mgnls_mult_111	485Fh
lane3_tx_txcc_mgnls_mult_111	4C5Fh

13.4.10.2.51.2 Diagram



13.4.10.2.51.3 Fields

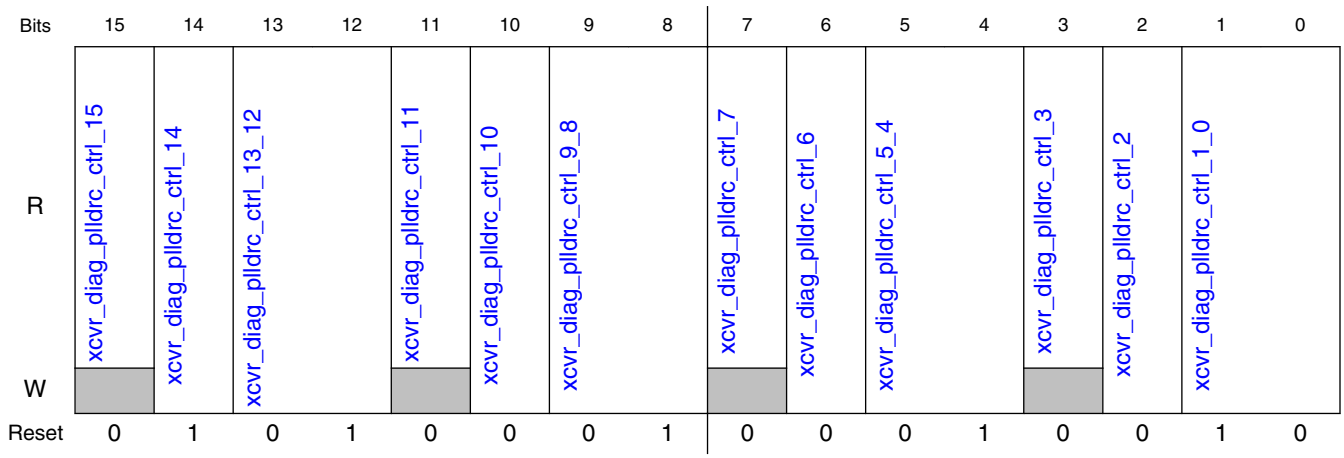
Field	Function
15-8 tx_txcc_mgnls_mult_111_15_8	Reserved
7-0 tx_txcc_mgnls_mult_111_7_0	Margin half swing multiplier value 111: The value in this field specifies the multiplier value used to generate the margin value from the resistor calibration value when

13.4.10.2.52 Transceiver PLL data rate clock control register (lane0_xcvr_diag_plldrc_ctrl - lane3_xcvr_diag_plldrc_ctrl)

13.4.10.2.52.1 Offset

Register	Offset
lane0_xcvr_diag_plldrc_ctrl	40E0h
lane1_xcvr_diag_plldrc_ctrl	44E0h
lane2_xcvr_diag_plldrc_ctrl	48E0h
lane3_xcvr_diag_plldrc_ctrl	4CE0h

13.4.10.2.52.2 Diagram



13.4.10.2.52.3 Fields

Field	Function
15 xcvr_diag_plldrc_ctrl_15	Reserved
14 xcvr_diag_plldrc_ctrl_14	Digital PLL clock select standard mode 3: This bit controls which 500 MHz PLL clock is selected, when
13-12 xcvr_diag_plldrc_ctrl_13_12	Digital PLL data rate divider standard mode 3 value: This field will directly control the xcvr_pll_clk_datart_div signal, which controls which divided clock is selected when generating the
11 xcvr_diag_plldrc_ctrl_11	Reserved
10 xcvr_diag_plldrc_ctrl_10	Digital PLL clock select standard mode 2: This bit controls which 500 MHz PLL clock is selected, when
9-8 xcvr_diag_plldrc_ctrl_9_8	Digital PLL data rate divider standard mode 2 value: This field will directly control the xcvr_pll_clk_datart_div signal, which controls which divided clock is selected when generating the
7 xcvr_diag_plldrc_ctrl_7	Reserved
6 xcvr_diag_plldrc_ctrl_6	Digital PLL clock select standard mode 1: This bit controls which 500 MHz PLL clock is selected, when

Table continues on the next page...

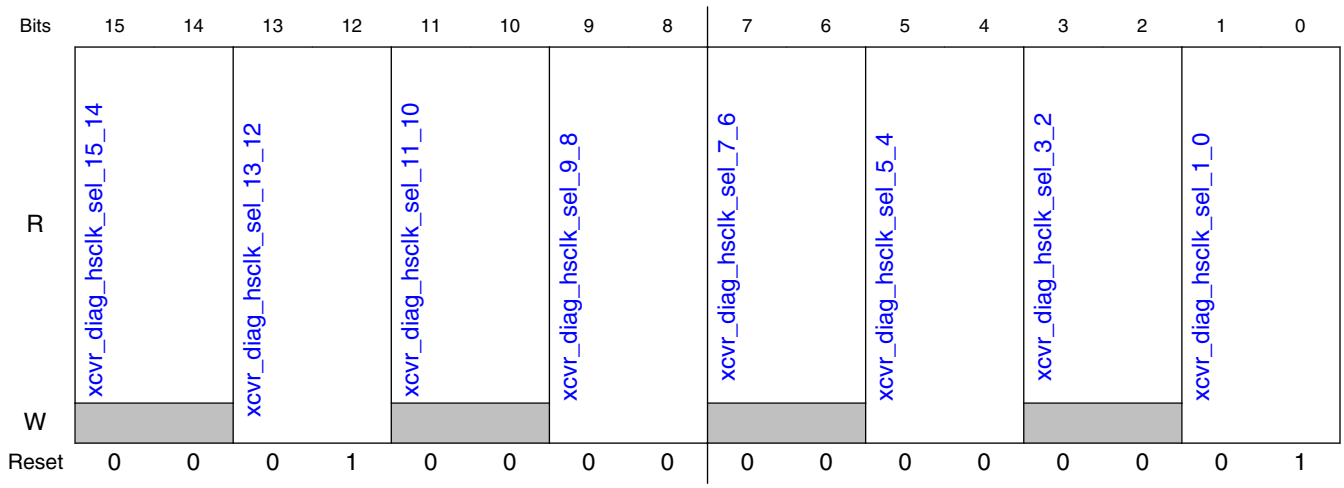
Field	Function
5-4 xcvr_diag_plldrc_ctrl_5_4	Digital PLL data rate divider standard mode 1 value: This field will directly control the xcvr_pll_clk_datart_div signal, which controls which divided clock is selected when generating the
3 xcvr_diag_plldrc_ctrl_3	Reserved
2 xcvr_diag_plldrc_ctrl_2	Digital PLL clock select standard mode 0: This bit controls which 500 MHz PLL clock is selected, when
1-0 xcvr_diag_plldrc_ctrl_1_0	Digital PLL data rate divider standard mode 0 value: This field will directly control the xcvr_pll_clk_datart_div signal, which controls which divided clock is selected when generating the

13.4.10.2.53 Transceiver high speed clock select register (lane0_xcvr_diag_hsclk_sel - lane3_xcvr_diag_hsclk_sel)

13.4.10.2.53.1 Offset

Register	Offset
lane0_xcvr_diag_hsclk_sel	40E1h
lane1_xcvr_diag_hsclk_sel	44E1h
lane2_xcvr_diag_hsclk_sel	48E1h
lane3_xcvr_diag_hsclk_sel	4CE1h

13.4.10.2.53.2 Diagram



13.4.10.2.53.3 Fields

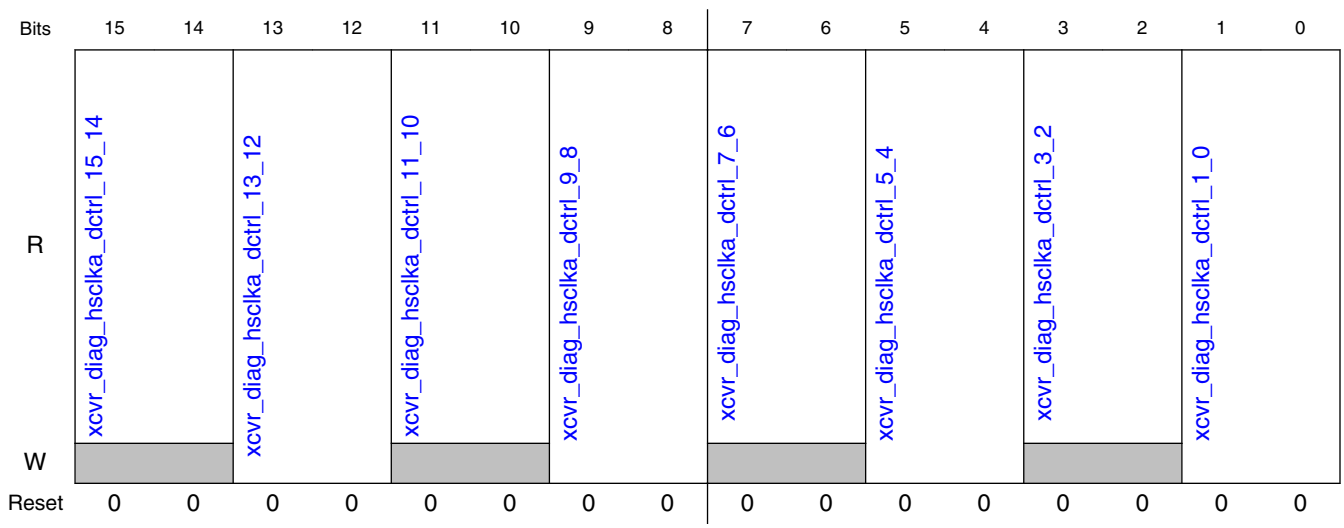
Field	Function
15-14 xcvr_diag_hsclk_sel_15_14	Reserved
13-12 xcvr_diag_hsclk_sel_13_12	High speed clock select standard mode 3: This specifies which analog high speed clock is selected when
11-10 xcvr_diag_hsclk_sel_11_10	Reserved
9-8 xcvr_diag_hsclk_sel_9_8	High speed clock select standard mode 2: This specifies which analog high speed clock is selected when
7-6 xcvr_diag_hsclk_sel_7_6	Reserved
5-4 xcvr_diag_hsclk_sel_5_4	High speed clock select standard mode 1: This specifies which analog high speed clock is selected when
3-2 xcvr_diag_hsclk_sel_3_2	Reserved
1-0 xcvr_diag_hsclk_sel_1_0	High speed clock select standard mode 0: This specifies which analog high speed clock is selected when

13.4.10.2.54 Transceiver high speed clock A divider control register (lane 0_xcvr_diag_hsclka_dctrl - lane3_xcvr_diag_hsclka_dctrl)

13.4.10.2.54.1 Offset

Register	Offset
lane0_xcvr_diag_hsclka_dctrl	40E2h
lane1_xcvr_diag_hsclka_dctrl	44E2h
lane2_xcvr_diag_hsclka_dctrl	48E2h
lane3_xcvr_diag_hsclka_dctrl	4CE2h

13.4.10.2.54.2 Diagram



13.4.10.2.54.3 Fields

Field	Function
15-14 xcvr_diag_hsclka_dctrl_15_14	Reserved
13-12 xcvr_diag_hsclka_dctrl_13_12	Transceiver clock A (transmitter) divider control standard mode 3: Controls the divider for clock A (which is the transmitter clock) in the analog transceiver clock distribution module, when

Table continues on the next page...

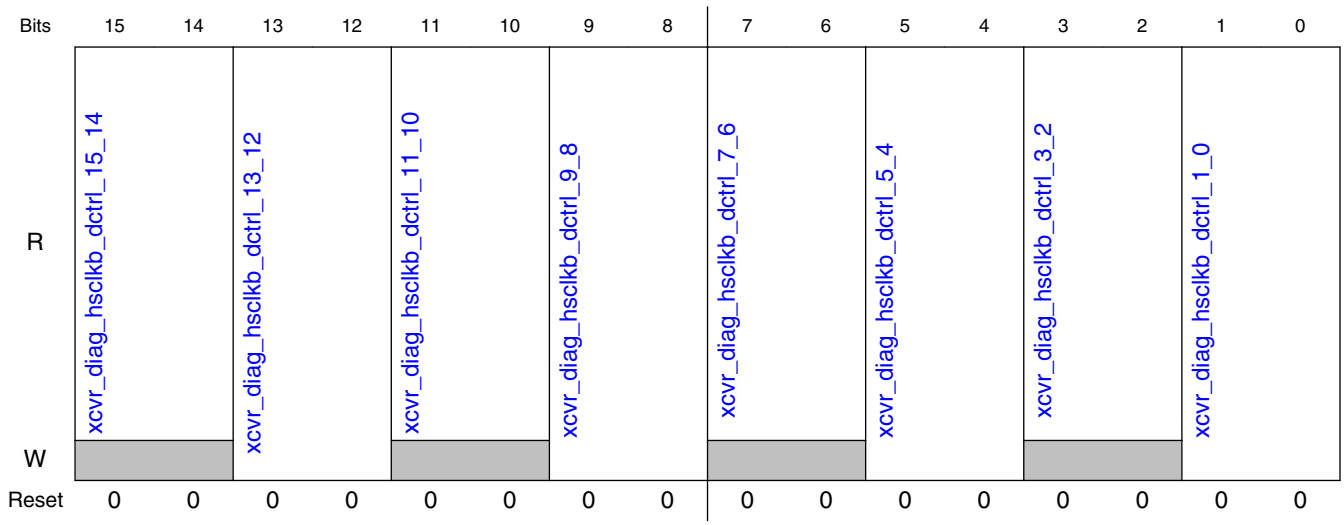
Field	Function
11-10 xcvr_diag_hsclka_dctrl_11_10	Reserved
9-8 xcvr_diag_hsclka_dctrl_9_8	Transceiver clock A (transmitter) divider control standard mode 2: Controls the divider for clock A (which is the transmitter clock) in the analog transceiver clock distribution module, when
7-6 xcvr_diag_hsclka_dctrl_7_6	Reserved
5-4 xcvr_diag_hsclka_dctrl_5_4	Transceiver clock A (transmitter) divider control standard mode 1: Controls the divider for clock A (which is the transmitter clock) in the analog transceiver clock distribution module, when
3-2 xcvr_diag_hsclka_dctrl_3_2	Reserved
1-0 xcvr_diag_hsclka_dctrl_1_0	Transceiver clock A (transmitter) divider control standard mode 0: Controls the divider for clock A (which is the transmitter clock) in the analog transceiver clock distribution module, when

13.4.10.2.55 Transceiver high speed clock B divider control register (lane 0_xcvr_diag_hsclkb_dctrl - lane3_xcvr_diag_hsclkb_dctrl)

13.4.10.2.55.1 Offset

Register	Offset
lane0_xcvr_diag_hsclkb_dctrl	40E3h
lane1_xcvr_diag_hsclkb_dctrl	44E3h
lane2_xcvr_diag_hsclkb_dctrl	48E3h
lane3_xcvr_diag_hsclkb_dctrl	4CE3h

13.4.10.2.55.2 Diagram



13.4.10.2.55.3 Fields

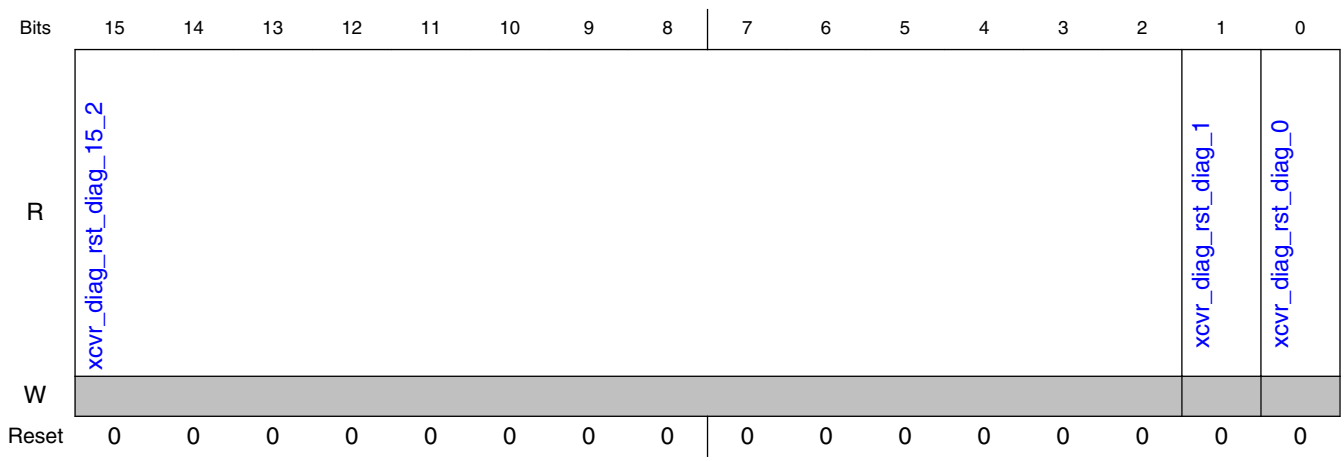
Field	Function
15-14 xcvr_diag_hsclkb_dctrl_15_14	Reserved
13-12 xcvr_diag_hsclkb_dctrl_13_12	Transceiver clock B (receiver) divider control standard mode 3: Controls the divider for clock B (which is the receiver clock) in the analog transceiver clock distribution module, when
11-10 xcvr_diag_hsclkb_dctrl_11_10	Reserved
9-8 xcvr_diag_hsclkb_dctrl_9_8	Transceiver clock B (receiver) divider control standard mode 2: Controls the divider for clock B (which is the receiver clock) in the analog transceiver clock distribution module, when
7-6 xcvr_diag_hsclkb_dctrl_7_6	Reserved
5-4 xcvr_diag_hsclkb_dctrl_5_4	Transceiver clock B (receiver) divider control standard mode 1: Controls the divider for clock B (which is the receiver clock) in the analog transceiver clock distribution module, when
3-2 xcvr_diag_hsclkb_dctrl_3_2	Reserved
1-0 xcvr_diag_hsclkb_dctrl_1_0	Transceiver clock B (receiver) divider control standard mode 0: Controls the divider for clock B (which is the receiver clock) in the analog transceiver clock distribution module, when

13.4.10.2.56 Transceiver control reset diagnostic register (lane0_xcvr_diag_rst_diag - lane3_xcvr_diag_rst_diag)

13.4.10.2.56.1 Offset

Register	Offset
lane0_xcvr_diag_rst_diag	40E7h
lane1_xcvr_diag_rst_diag	44E7h
lane2_xcvr_diag_rst_diag	48E7h
lane3_xcvr_diag_rst_diag	4CE7h

13.4.10.2.56.2 Diagram



13.4.10.2.56.3 Fields

Field	Function
15-2 xcvr_diag_rst_diag_15_2	Reserved
1 xcvr_diag_rst_diag_1	Current state of the xcvr_psm_reset_n reset.
0 xcvr_diag_rst_diag_0	Current state of the xcvr_ref_clk_reset_n reset.

13.4.10.2.57 Transceiver bidirectional control register (lane0_xcvr_diag_bidi_ctrl - lane3_xcvr_diag_bidi_ctrl)

13.4.10.2.57.1 Offset

Register	Offset
lane0_xcvr_diag_bidi_ctrl	40E8h
lane1_xcvr_diag_bidi_ctrl	44E8h
lane2_xcvr_diag_bidi_ctrl	48E8h
lane3_xcvr_diag_bidi_ctrl	4CE8h

13.4.10.2.57.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	xcvr_diag_bidi_ctrl_15_8								xcvr_diag_bidi_ctrl_7	xcvr_diag_bidi_ctrl_6	xcvr_diag_bidi_ctrl_5	xcvr_diag_bidi_ctrl_4	xcvr_diag_bidi_ctrl_3	xcvr_diag_bidi_ctrl_2	xcvr_diag_bidi_ctrl_1	xcvr_diag_bidi_ctrl_0
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1

13.4.10.2.57.3 Fields

Field	Function
15-8 xcvr_diag_bidi_ctrl_15_8	Reserved
7 xcvr_diag_bidi_ctrl_7	Receiver enable standard mode 3: This bit is a global enable for the receiver function, when
6 xcvr_diag_bidi_ctrl_6	Receiver enable standard mode 2: This bit is a global enable for the receiver function, when
5 xcvr_diag_bidi_ctrl_5	Receiver enable standard mode 1: This bit is a global enable for the receiver function, when

Table continues on the next page...

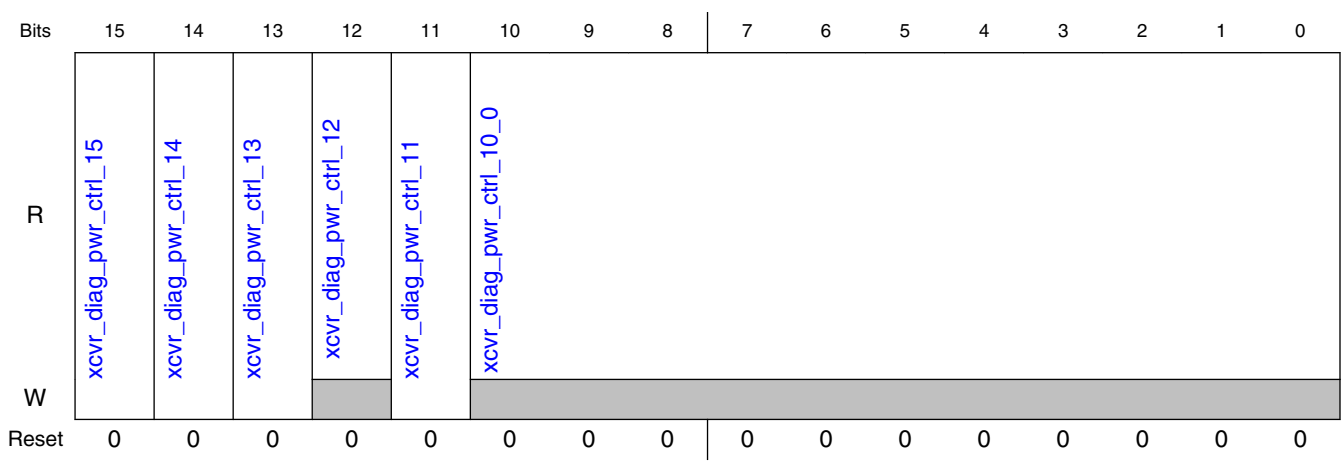
Field	Function
4 xcvr_diag_bidi_ctrl_4	Receiver enable standard mode 0: This bit is a global enable for the receiver function, when
3 xcvr_diag_bidi_ctrl_3	Transmitter enable standard mode 3: This bit is a global enable for the transmitter function, when
2 xcvr_diag_bidi_ctrl_2	Transmitter enable standard mode 2: This bit is a global enable for the transmitter function, when
1 xcvr_diag_bidi_ctrl_1	Transmitter enable standard mode 1: This bit is a global enable for the transmitter function, when
0 xcvr_diag_bidi_ctrl_0	Transmitter enable standard mode 0: This bit is a global enable for the transmitter function, when

13.4.10.2.58 Transceiver power island control register (lane0_xcvr_diag_pwr_ctrl - lane3_xcvr_diag_pwr_ctrl)

13.4.10.2.58.1 Offset

Register	Offset
lane0_xcvr_diag_pwr_ctrl	40E9h
lane1_xcvr_diag_pwr_ctrl	44E9h
lane2_xcvr_diag_pwr_ctrl	48E9h
lane3_xcvr_diag_pwr_ctrl	4CE9h

13.4.10.2.58.2 Diagram



13.4.10.2.58.3 Fields

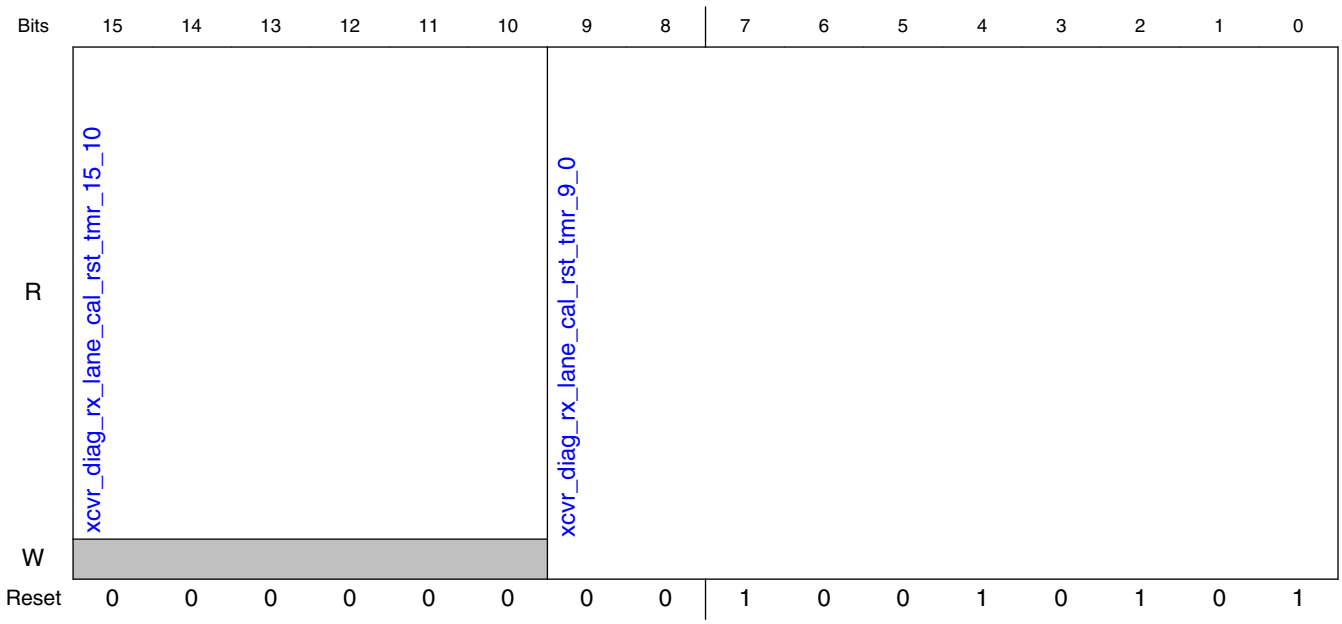
Field	Function
15 xcvr_diag_pwr_ctrl_15	Transceiver Dsync power down disable: Setting this bit to 1b1 will disable turning off the transceiver dsync power island when the macro is in a state that would normally switch the power island off.
14 xcvr_diag_pwr_ctrl_14	Transceiver calibration one time power down disable: Setting this bit to 1b1 will disable turning off the transceiver calibration one time power island when the macro is in a state that would normally switch the power island off.
13 xcvr_diag_pwr_ctrl_13	Transceiver calibration multiples time power down disable: Setting this bit to 1b1 will disable turning off the transceiver calibration multiple times power island when the macro is in a state that would normally switch the power island off.
12 xcvr_diag_pwr_ctrl_12	Reserved
11 xcvr_diag_pwr_ctrl_11	Transceiver test functions power enable: Controls the enabling of the transceiver test functions power island.
10-0 xcvr_diag_pwr_ctrl_10_0	Reserved

13.4.10.2.59 RX Lane calibration reset timer register (lane0_xcvr_diag_rx_lane_cal_rst_tmr - lane3_xcvr_diag_rx_lane_cal_rst_tmr)

13.4.10.2.59.1 Offset

Register	Offset
lane0_xcvr_diag_rx_lane_cal_rst_tmr	40EAh
lane1_xcvr_diag_rx_lane_cal_rst_tmr	44EAh
lane2_xcvr_diag_rx_lane_cal_rst_tmr	48EAh
lane3_xcvr_diag_rx_lane_cal_rst_tmr	4CEAh

13.4.10.2.59.2 Diagram



13.4.10.2.59.3 Fields

Field	Function
15-10 xcvr_diag_rx_lane_cal_rst_tmr_15_10	Reserved
9-0 xcvr_diag_rx_lane_cal_rst_tmr_9_0	Lane calibration receiver reset timer value: Value of the lane calibration receiver reset timer. This specifies the number of

13.4.10.2.60 Lane fast common mode enable timeout register (lane0_xcvr_diag_lane_fcm_en_to - lane3_xcvr_diag_lane_fcm_en_to)

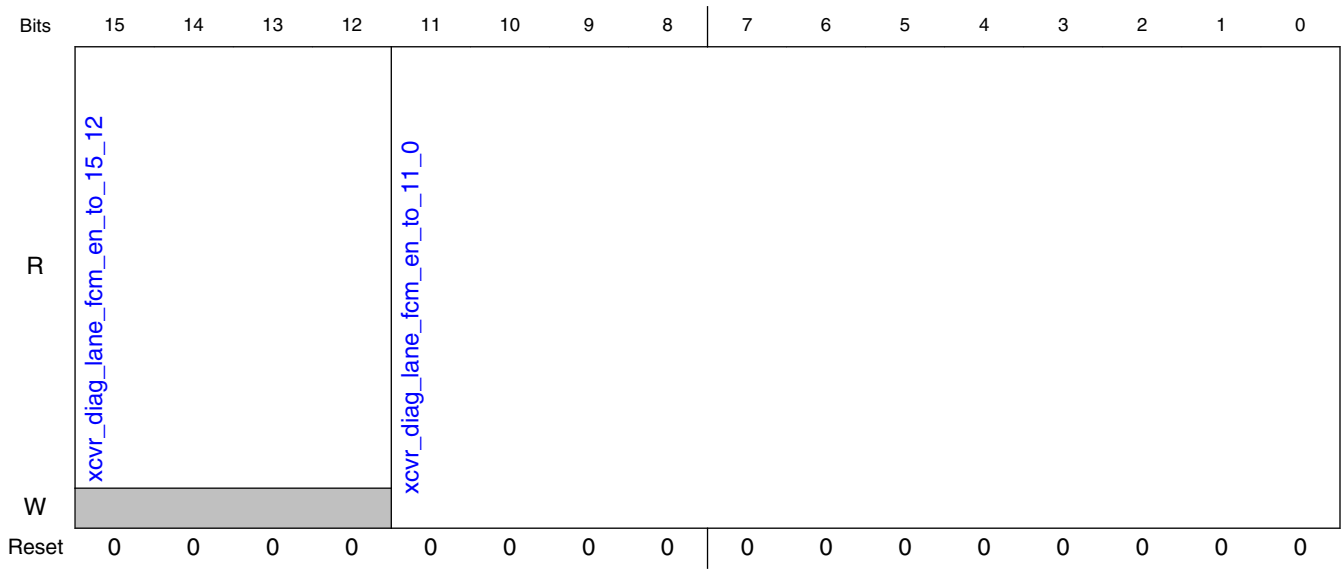
13.4.10.2.60.1 Offset

Register	Offset
lane0_xcvr_diag_lane_fcm_en_to	40F0h
lane1_xcvr_diag_lane_fcm_en_to	44F0h

Table continues on the next page...

Register	Offset
lane2_xcvr_diag_lane_fcm_en_to	48F0h
lane3_xcvr_diag_lane_fcm_en_to	4CF0h

13.4.10.2.60.2 Diagram



13.4.10.2.60.3 Fields

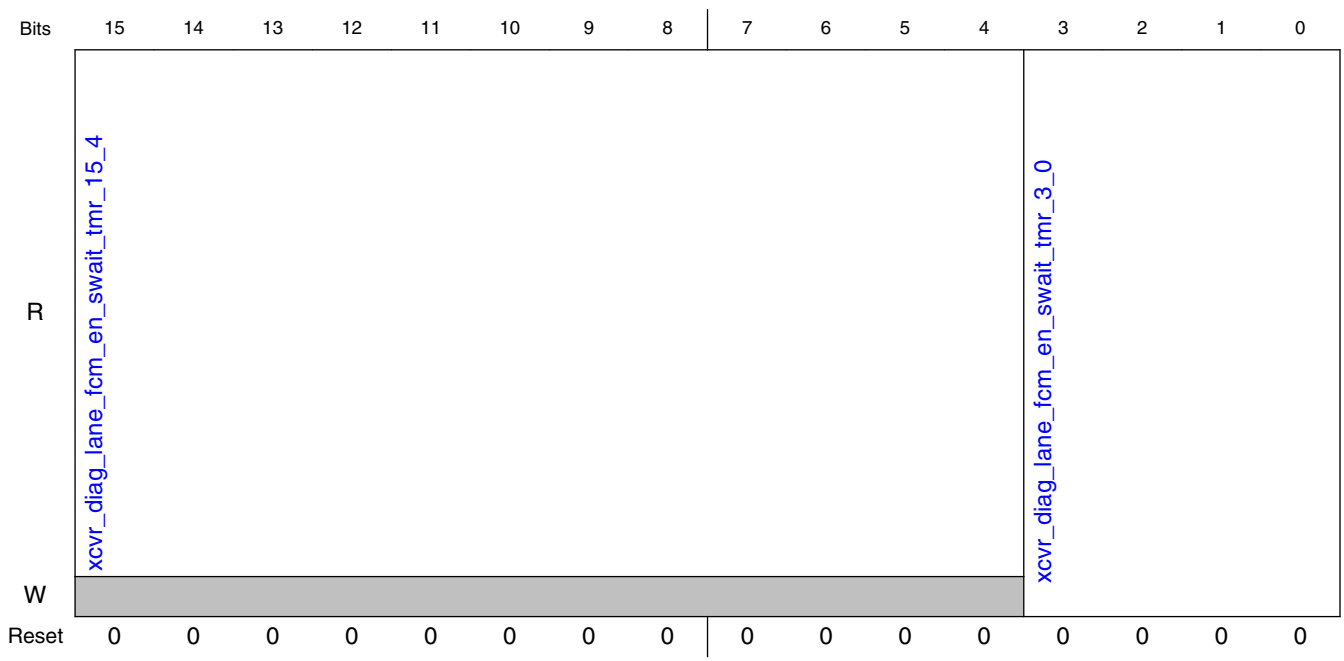
Field	Function
15-12 xcvr_diag_lane_fcm_en_to_15_12	Reserved
11-0 xcvr_diag_lane_fcm_en_to_11_0	Lane fast common mode enable timeout value: This specifies the maximum number of reference clock cycles the fast establishment of common mode process will wait for the

13.4.10.2.61 Lane fast common mode enable sample wait timer register (lane0_xcvr_diag_lane_fcm_en_swait_tmr - lane3_xcvr_diag_lane_fcm_en_swait_tmr)

13.4.10.2.61.1 Offset

Register	Offset
lane0_xcvr_diag_lane_fcm_en_swait_tmr	40F1h
lane1_xcvr_diag_lane_fcm_en_swait_tmr	44F1h
lane2_xcvr_diag_lane_fcm_en_swait_tmr	48F1h
lane3_xcvr_diag_lane_fcm_en_swait_tmr	4CF1h

13.4.10.2.61.2 Diagram



13.4.10.2.61.3 Fields

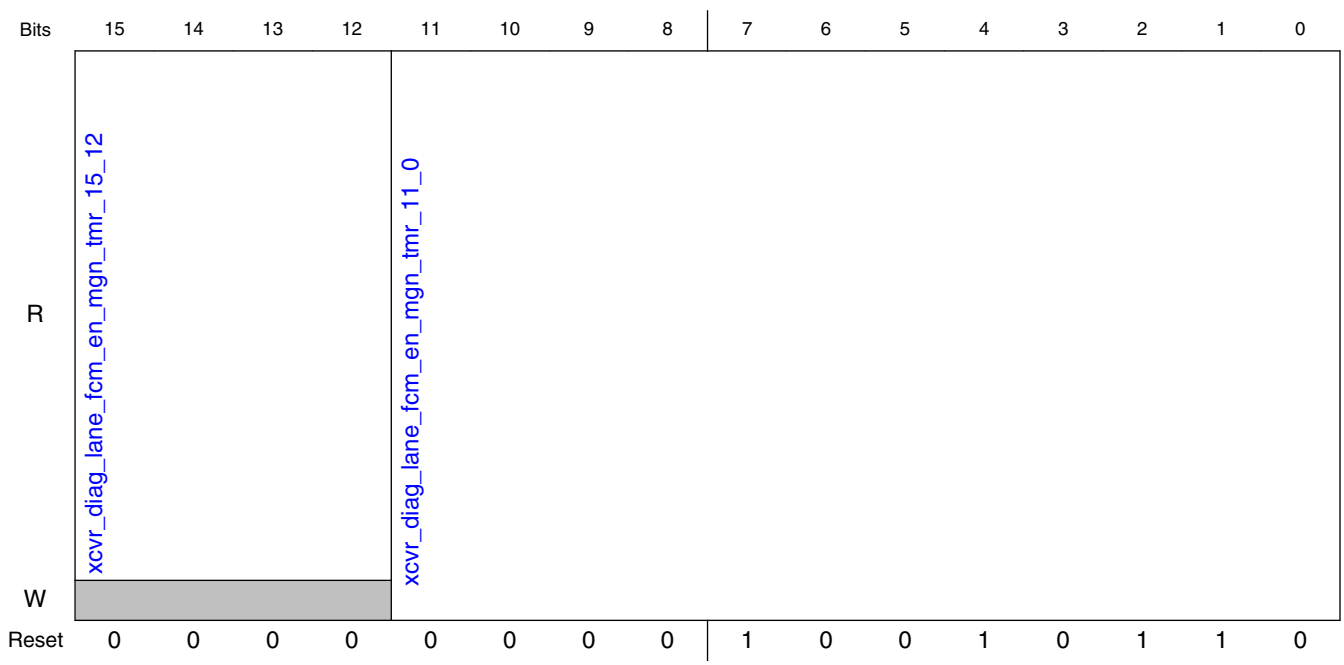
Field	Function
15-4 xcvr_diag_lane_fcm_en_swait_tmr_15_4	Reserved
3-0 xcvr_diag_lane_fcm_en_swait_tmr_3_0	Lane fast common mode enable sample wait timer value: This specifies the number of reference clock cycles the fast establishment of common mode process will wait between changing the state of the signals controlling the analog common mode sense circuits, and testing the results of the new state.

13.4.10.2.62 Lane fast common mode enable margin timer register (lane0_xcvr_diag_lane_fcm_en_mgn_tmr - lane3_xcvr_diag_lane_fcm_en_mgn_tmr)

13.4.10.2.62.1 Offset

Register	Offset
lane0_xcvr_diag_lane_fcm_en_mgn_tmr	40F2h
lane1_xcvr_diag_lane_fcm_en_mgn_tmr	44F2h
lane2_xcvr_diag_lane_fcm_en_mgn_tmr	48F2h
lane3_xcvr_diag_lane_fcm_en_mgn_tmr	4CF2h

13.4.10.2.62.2 Diagram



13.4.10.2.62.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

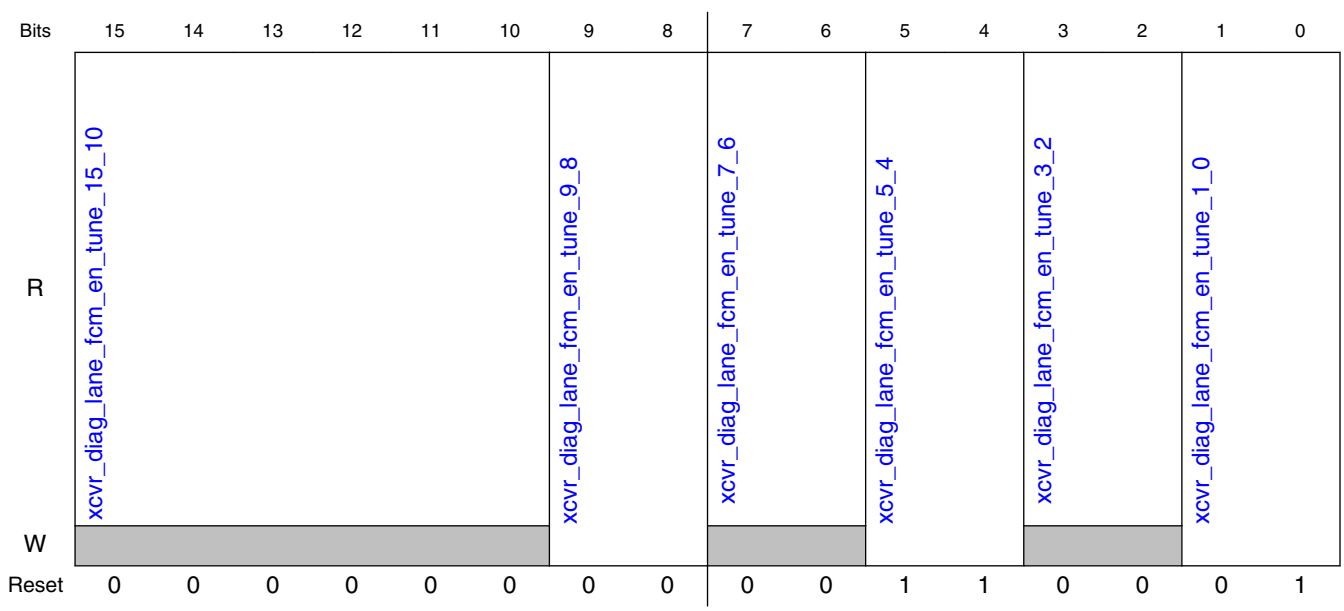
Field	Function
xcvr_diag_lane_fcm_en_mgn_tmr_15_12	
11-0 xcvr_diag_lane_fcm_en_mgn_tmr_11_0	Lane fast common mode enable margin timer value: This specifies the number of reference clock cycles the fast establishment of common mode process will enable all the margin segments for.

13.4.10.2.63 Lane fast common mode enable tuning register (lane0_xcvr_diag_lane_fcm_en_tune - lane3_xcvr_diag_lane_fcm_en_tune)

13.4.10.2.63.1 Offset

Register	Offset
lane0_xcvr_diag_lane_fcm_en_tune	40F3h
lane1_xcvr_diag_lane_fcm_en_tune	44F3h
lane2_xcvr_diag_lane_fcm_en_tune	48F3h
lane3_xcvr_diag_lane_fcm_en_tune	4CF3h

13.4.10.2.63.2 Diagram



13.4.10.2.63.3 Fields

Field	Function
15-10 xcvr_diag_lane_fcm_en_tune_15_10	Reserved
9-8 xcvr_diag_lane_fcm_en_tune_9_8	Common mode sense reference DAC voltage initial test: This field sets the common mode detect reference voltage for the common mode detect comparator, when the fast establishment of common mode function is checking the initial state of the
7-6 xcvr_diag_lane_fcm_en_tune_7_6	Reserved
5-4 xcvr_diag_lane_fcm_en_tune_5_4	Common mode sense reference DAC voltage high test: This field sets the common mode detect reference voltage for the common mode detect comparator, when the fast establishment of common mode function is waiting for the
3-2 xcvr_diag_lane_fcm_en_tune_3_2	Reserved
1-0 xcvr_diag_lane_fcm_en_tune_1_0	Common mode sense reference DAC voltage low test: This field sets the common mode detect reference voltage for the common mode detect comparator, when the fast establishment of common mode function is waiting for the

13.4.10.2.64 Transceiver digital cover your alternatives register (lane0_xcvr_diag_dcya - lane3_xcvr_diag_dcya)

13.4.10.2.64.1 Offset

Register	Offset
lane0_xcvr_diag_dcya	40FFh
lane1_xcvr_diag_dcya	44FFh
lane2_xcvr_diag_dcya	48FFh
lane3_xcvr_diag_dcya	4CFFh

13.4.10.2.64.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	xcvr_diag_dcya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.64.3 Fields

Field	Function
15-0 xcvr_diag_dcya_15_0	Reserved

13.4.10.2.65 Transmitter A0 power state definition register (lane0_tx_psc_a0 - lane3_tx_psc_a0)

13.4.10.2.65.1 Offset

Register	Offset
lane0_tx_psc_a0	4100h
lane1_tx_psc_a0	4500h
lane2_tx_psc_a0	4900h
lane3_tx_psc_a0	4D00h

13.4.10.2.65.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a0_15	tx_psc_a0_14	tx_psc_a0_13	tx_psc_a0_12	tx_psc_a0_11	tx_psc_a0_10	tx_psc_a0_9	tx_psc_a0_8	tx_psc_a0_7	tx_psc_a0_6	tx_psc_a0_5	tx_psc_a0_4	tx_psc_a0_3	tx_psc_a0_2	tx_psc_a0_1	tx_psc_a0_0
W																
Reset	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	1

13.4.10.2.65.3 Fields

Field	Function
15 tx_psc_a0_15	Reserved
14 tx_psc_a0_14	TX DCAP enable: Enables the transmitter DCAP, via the
13 tx_psc_a0_13	TX UPHY supply enable: Enables the transmitter UPHY components supply, via the
12 tx_psc_a0_12	LFPS controller enable: Enables the LFPS controller function.
11 tx_psc_a0_11	MPHY pre-driver enable: Enables the MPHY pre-driver components.
10 tx_psc_a0_10	TX serializer clock enable: Enables the serializer clock divider circuits.
9 tx_psc_a0_9	TX serializer enable: Enables the serializer circuits.
8 tx_psc_a0_8	TX clock enable: Enables the transmitter clock and MUX circuits in the transceiver clock distribution, via the
7 tx_psc_a0_7	Transmitter low current mode: Enables a low current consumption mode within the common mode voltage circuit in the driver, via the
6 tx_psc_a0_6	Transmitter mission mode enable: Enables the analog circuits in the driver required to run in mission mode, via the
5 tx_psc_a0_5	TX driver common mode enable extend control: Specifies which power states the
4 tx_psc_a0_4	TX driver common mode enable: Enables the common mode voltage circuits in the driver.
3 tx_psc_a0_3	TX post-emphasis enable: Enables the transmitter circuits related to the post-emphasis function.
2 tx_psc_a0_2	TX pre-emphasis enable: Enables the transmitter circuits related to the pre-emphasis function.
1 tx_psc_a0_1	TX driver LDO enable: Enables the transmitter circuits related to the driver LDO function, but driving the
0 tx_psc_a0_0	TX driver enable: Enables the transmitter driver, via the H bridge driver controller. Note that this also controls whether the

13.4.10.2.66 Transmitter A1 power state definition register (lane0_tx_psc_a1 - lane3_tx_psc_a1)

13.4.10.2.66.1 Offset

Register	Offset
lane0_tx_psc_a1	4101h
lane1_tx_psc_a1	4501h
lane2_tx_psc_a1	4901h
lane3_tx_psc_a1	4D01h

13.4.10.2.66.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a1_15	tx_psc_a1_14	tx_psc_a1_13	tx_psc_a1_12	tx_psc_a1_11	tx_psc_a1_10	tx_psc_a1_9	tx_psc_a1_8	tx_psc_a1_7	tx_psc_a1_6	tx_psc_a1_5	tx_psc_a1_4	tx_psc_a1_3	tx_psc_a1_2	tx_psc_a1_1	tx_psc_a1_0
W																
Reset	0	1	1	0	0	1	1	1	1	0	0	1	1	0	0	0

13.4.10.2.66.3 Fields

Field	Function
15 tx_psc_a1_15	Reserved
14 tx_psc_a1_14	TX DCAP enable
13 tx_psc_a1_13	TX UPHY supply enable
12 tx_psc_a1_12	LFPS controller enable
11 tx_psc_a1_11	MPHY pre-driver enable
10 tx_psc_a1_10	TX serializer clock enable
9 tx_psc_a1_9	TX serializer enable
8 tx_psc_a1_8	TX clock enable
7	Transmitter low current mode

Table continues on the next page...

Field	Function
tx_psc_a1_7	
6 tx_psc_a1_6	Transmitter mission mode enable
5 tx_psc_a1_5	TX driver common mode enable extend control
4 tx_psc_a1_4	TX driver common mode enable
3 tx_psc_a1_3	TX post-emphasis enable
2 tx_psc_a1_2	TX pre-emphasis enable
1 tx_psc_a1_1	TX driver LDO enable
0 tx_psc_a1_0	TX driver enable

13.4.10.2.67 Transmitter A2 power state definition register (lane0_tx_psc_a2 - lane3_tx_psc_a2)

13.4.10.2.67.1 Offset

Register	Offset
lane0_tx_psc_a2	4102h
lane1_tx_psc_a2	4502h
lane2_tx_psc_a2	4902h
lane3_tx_psc_a2	4D02h

13.4.10.2.67.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a2_15	tx_psc_a2_14	tx_psc_a2_13	tx_psc_a2_12	tx_psc_a2_11	tx_psc_a2_10	tx_psc_a2_9	tx_psc_a2_8	tx_psc_a2_7	tx_psc_a2_6	tx_psc_a2_5	tx_psc_a2_4	tx_psc_a2_3	tx_psc_a2_2	tx_psc_a2_1	tx_psc_a2_0
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0

13.4.10.2.67.3 Fields

Field	Function
15 tx_psc_a2_15	Reserved
14 tx_psc_a2_14	TX DCAP enable
13 tx_psc_a2_13	TX UPHY supply enable
12 tx_psc_a2_12	LFPS controller enable
11 tx_psc_a2_11	MPHY pre-driver enable
10 tx_psc_a2_10	TX serializer clock enable
9 tx_psc_a2_9	TX serializer enable
8 tx_psc_a2_8	TX clock enable
7 tx_psc_a2_7	Transmitter low current mode
6 tx_psc_a2_6	Transmitter mission mode enable
5 tx_psc_a2_5	TX driver common mode enable extend control
4 tx_psc_a2_4	TX driver common mode enable
3 tx_psc_a2_3	TX post-emphasis enable
2 tx_psc_a2_2	TX pre-emphasis enable
1 tx_psc_a2_1	TX driver LDO enable
0 tx_psc_a2_0	TX driver enable

13.4.10.2.68 Transmitter A3 power state definition register (lane0_tx_psc_a3 - lane3_tx_psc_a3)

13.4.10.2.68.1 Offset

Register	Offset
lane0_tx_psc_a3	4103h
lane1_tx_psc_a3	4503h
lane2_tx_psc_a3	4903h
lane3_tx_psc_a3	4D03h

13.4.10.2.68.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a3_15	tx_psc_a3_14	tx_psc_a3_13	tx_psc_a3_12	tx_psc_a3_11	tx_psc_a3_10	tx_psc_a3_9	tx_psc_a3_8	tx_psc_a3_7	tx_psc_a3_6	tx_psc_a3_5	tx_psc_a3_4	tx_psc_a3_3	tx_psc_a3_2	tx_psc_a3_1	tx_psc_a3_0
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0

13.4.10.2.68.3 Fields

Field	Function
15 tx_psc_a3_15	Reserved
14 tx_psc_a3_14	TX DCAP enable
13 tx_psc_a3_13	TX UPHY supply enable
12 tx_psc_a3_12	LFPS controller enable
11 tx_psc_a3_11	MPHY pre-driver enable
10 tx_psc_a3_10	TX serializer clock enable
9 tx_psc_a3_9	TX serializer enable
8 tx_psc_a3_8	TX clock enable
7	Transmitter low current mode

Table continues on the next page...

Field	Function
tx_psc_a3_7	
6 tx_psc_a3_6	Transmitter mission mode enable
5 tx_psc_a3_5	TX driver common mode enable extend control
4 tx_psc_a3_4	TX driver common mode enable
3 tx_psc_a3_3	TX post-emphasis enable
2 tx_psc_a3_2	TX pre-emphasis enable
1 tx_psc_a3_1	TX driver LDO enable
0 tx_psc_a3_0	TX driver enable

13.4.10.2.69 Transmitter A4 power state definition register (lane0_tx_psc_a4 - lane3_tx_psc_a4)

13.4.10.2.69.1 Offset

Register	Offset
lane0_tx_psc_a4	4104h
lane1_tx_psc_a4	4504h
lane2_tx_psc_a4	4904h
lane3_tx_psc_a4	4D04h

13.4.10.2.69.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a4_15	tx_psc_a4_14	tx_psc_a4_13	tx_psc_a4_12	tx_psc_a4_11	tx_psc_a4_10	tx_psc_a4_9	tx_psc_a4_8	tx_psc_a4_7	tx_psc_a4_6	tx_psc_a4_5	tx_psc_a4_4	tx_psc_a4_3	tx_psc_a4_2	tx_psc_a4_1	tx_psc_a4_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0

13.4.10.2.69.3 Fields

Field	Function
15 tx_psc_a4_15	Reserved
14 tx_psc_a4_14	TX DCAP enable
13 tx_psc_a4_13	TX UPHY supply enable
12 tx_psc_a4_12	LFPS controller enable
11 tx_psc_a4_11	MPHY pre-driver enable
10 tx_psc_a4_10	TX serializer clock enable
9 tx_psc_a4_9	TX serializer enable
8 tx_psc_a4_8	TX clock enable
7 tx_psc_a4_7	Transmitter low current mode
6 tx_psc_a4_6	Transmitter mission mode enable
5 tx_psc_a4_5	TX driver common mode enable extend control
4 tx_psc_a4_4	TX driver common mode enable
3 tx_psc_a4_3	TX post-emphasis enable
2 tx_psc_a4_2	TX pre-emphasis enable
1 tx_psc_a4_1	TX driver LDO enable
0 tx_psc_a4_0	TX driver enable

13.4.10.2.70 Transmitter A5 power state definition register (lane0_tx_psc_a5 - lane3_tx_psc_a5)

13.4.10.2.70.1 Offset

Register	Offset
lane0_tx_psc_a5	4105h
lane1_tx_psc_a5	4505h
lane2_tx_psc_a5	4905h
lane3_tx_psc_a5	4D05h

13.4.10.2.70.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_a5_15	tx_psc_a5_14	tx_psc_a5_13	tx_psc_a5_12	tx_psc_a5_11	tx_psc_a5_10	tx_psc_a5_9	tx_psc_a5_8	tx_psc_a5_7	tx_psc_a5_6	tx_psc_a5_5	tx_psc_a5_4	tx_psc_a5_3	tx_psc_a5_2	tx_psc_a5_1	tx_psc_a5_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.70.3 Fields

Field	Function
15 tx_psc_a5_15	Reserved
14 tx_psc_a5_14	TX DCAP enable
13 tx_psc_a5_13	TX UPHY supply enable
12 tx_psc_a5_12	LFPS controller enable
11 tx_psc_a5_11	MPHY pre-driver enable
10 tx_psc_a5_10	TX serializer clock enable
9 tx_psc_a5_9	TX serializer enable
8 tx_psc_a5_8	TX clock enable
7	Transmitter low current mode

Table continues on the next page...

Field	Function
tx_psc_a5_7	
6 tx_psc_a5_6	Transmitter mission mode enable
5 tx_psc_a5_5	TX driver common mode enable extend control
4 tx_psc_a5_4	TX driver common mode enable
3 tx_psc_a5_3	TX post-emphasis enable
2 tx_psc_a5_2	TX pre-emphasis enable
1 tx_psc_a5_1	TX driver LDO enable
0 tx_psc_a5_0	TX driver enable

13.4.10.2.71 Transmitter calibration power state definition register (lane0_tx_psc_cal - lane3_tx_psc_cal)

13.4.10.2.71.1 Offset

Register	Offset
lane0_tx_psc_cal	4106h
lane1_tx_psc_cal	4506h
lane2_tx_psc_cal	4906h
lane3_tx_psc_cal	4D06h

13.4.10.2.71.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_cal_15	tx_psc_cal_14	tx_psc_cal_13	tx_psc_cal_12	tx_psc_cal_11	tx_psc_cal_10	tx_psc_cal_9	tx_psc_cal_8	tx_psc_cal_7	tx_psc_cal_6	tx_psc_cal_5	tx_psc_cal_4	tx_psc_cal_3	tx_psc_cal_2	tx_psc_cal_1	tx_psc_cal_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.71.3 Fields

Field	Function
15 tx_psc_cal_15	Reserved
14 tx_psc_cal_14	TX DCAP enable
13 tx_psc_cal_13	TX UPHY supply enable
12 tx_psc_cal_12	LFPS controller enable
11 tx_psc_cal_11	MPHY pre-driver enable
10 tx_psc_cal_10	TX serializer clock enable
9 tx_psc_cal_9	TX serializer enable
8 tx_psc_cal_8	TX clock enable
7 tx_psc_cal_7	Transmitter low current mode
6 tx_psc_cal_6	Transmitter mission mode enable
5 tx_psc_cal_5	TX driver common mode enable extend control
4 tx_psc_cal_4	TX driver common mode enable
3 tx_psc_cal_3	TX post-emphasis enable
2 tx_psc_cal_2	TX pre-emphasis enable
1 tx_psc_cal_1	TX driver LDO enable
0 tx_psc_cal_0	TX driver enable

13.4.10.2.72 Transmitter ready power state definition register (lane0_tx_psc_rdy - lane3_tx_psc_rdy)

13.4.10.2.72.1 Offset

Register	Offset
lane0_tx_psc_rdy	4107h
lane1_tx_psc_rdy	4507h
lane2_tx_psc_rdy	4907h
lane3_tx_psc_rdy	4D07h

13.4.10.2.72.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_psc_rdy_15	tx_psc_rdy_14	tx_psc_rdy_13	tx_psc_rdy_12	tx_psc_rdy_11	tx_psc_rdy_10	tx_psc_rdy_9	tx_psc_rdy_8	tx_psc_rdy_7	tx_psc_rdy_6	tx_psc_rdy_5	tx_psc_rdy_4	tx_psc_rdy_3	tx_psc_rdy_2	tx_psc_rdy_1	tx_psc_rdy_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.72.3 Fields

Field	Function
15 tx_psc_rdy_15	Reserved
14 tx_psc_rdy_14	TX DCAP enable
13 tx_psc_rdy_13	TX UPHY supply enable
12 tx_psc_rdy_12	LFPS controller enable
11 tx_psc_rdy_11	MPHY pre-driver enable
10 tx_psc_rdy_10	TX serializer clock enable
9 tx_psc_rdy_9	TX serializer enable
8 tx_psc_rdy_8	TX clock enable
7	Transmitter low current mode

Table continues on the next page...

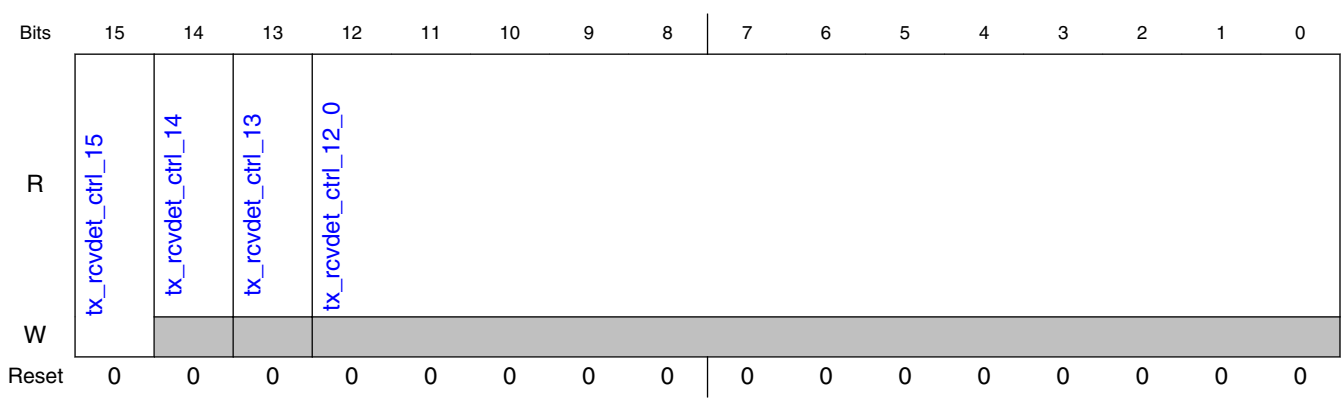
Field	Function
tx_psc_rdy_7	
6 tx_psc_rdy_6	Transmitter mission mode enable
5 tx_psc_rdy_5	TX driver common mode enable extend control
4 tx_psc_rdy_4	TX driver common mode enable
3 tx_psc_rdy_3	TX post-emphasis enable
2 tx_psc_rdy_2	TX pre-emphasis enable
1 tx_psc_rdy_1	TX driver LDO enable
0 tx_psc_rdy_0	TX driver enable

13.4.10.2.73 Transmit receiver detect control register (lane0_tx_rcvdet_ctrl - lane3_tx_rcvdet_ctrl)

13.4.10.2.73.1 Offset

Register	Offset
lane0_tx_rcvdet_ctrl	4120h
lane1_tx_rcvdet_ctrl	4520h
lane2_tx_rcvdet_ctrl	4920h
lane3_tx_rcvdet_ctrl	4D20h

13.4.10.2.73.2 Diagram



13.4.10.2.73.3 Fields

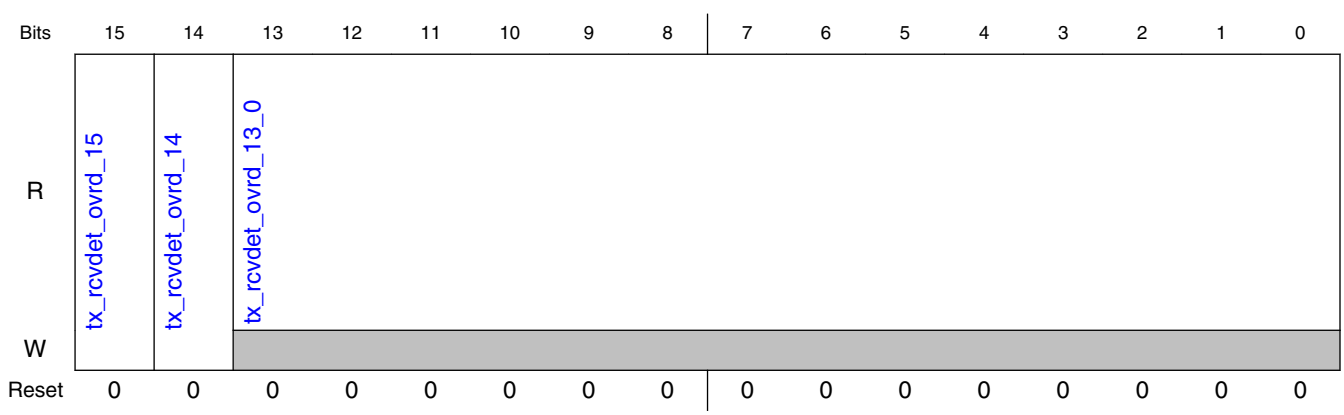
Field	Function
15 tx_rcvdet_ctrl_15	Start receiver detect: Activating (1b1) this bit will start the receiver detect process. This bit must remain active until the receiver detect process is complete, as indicated by the receiver detect process done bit in this register. To start another receiver detect process, this register must first be set inactive (1b0) until the receiver detect process done bit in this register is cleared.
14 tx_rcvdet_ctrl_14	Receiver detect process done: This bit will be set to 1b1 when the receiver detect process is complete. It will be cleared by
13 tx_rcvdet_ctrl_13	Receiver detected: When the receiver detect process is complete, this register bit will indicate the current state of the
12-0 tx_rcvdet_ctrl_12_0	Reserved

13.4.10.2.74 Transmit receiver detect override register (lane0_tx_rcvdet_ovrd - lane3_tx_rcvdet_ovrd)

13.4.10.2.74.1 Offset

Register	Offset
lane0_tx_rcvdet_ovrd	4121h
lane1_tx_rcvdet_ovrd	4521h
lane2_tx_rcvdet_ovrd	4921h
lane3_tx_rcvdet_ovrd	4D21h

13.4.10.2.74.2 Diagram



13.4.10.2.74.3 Fields

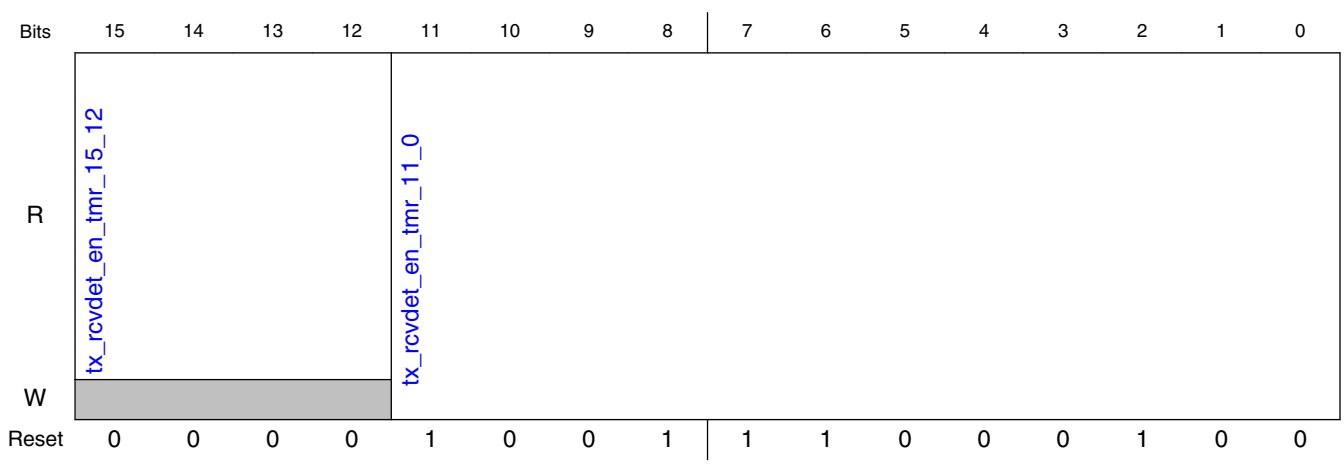
Field	Function
15 tx_rcvdet_ovrd_15	Receiver detect override enable: Activation (1b1) of this register bit enables the
14 tx_rcvdet_ovrd_14	Receiver detect override: When the receiver detect override enable bit in this register is active (1b1), this bit will directly control the
13-0 tx_rcvdet_ovrd_13_0	Reserved

13.4.10.2.75 Transmit receiver detect enable timer register (lane0_tx_rcvdet_en_tmr - lane3_tx_rcvdet_en_tmr)

13.4.10.2.75.1 Offset

Register	Offset
lane0_tx_rcvdet_en_tmr	4122h
lane1_tx_rcvdet_en_tmr	4522h
lane2_tx_rcvdet_en_tmr	4922h
lane3_tx_rcvdet_en_tmr	4D22h

13.4.10.2.75.2 Diagram



13.4.10.2.75.3 Fields

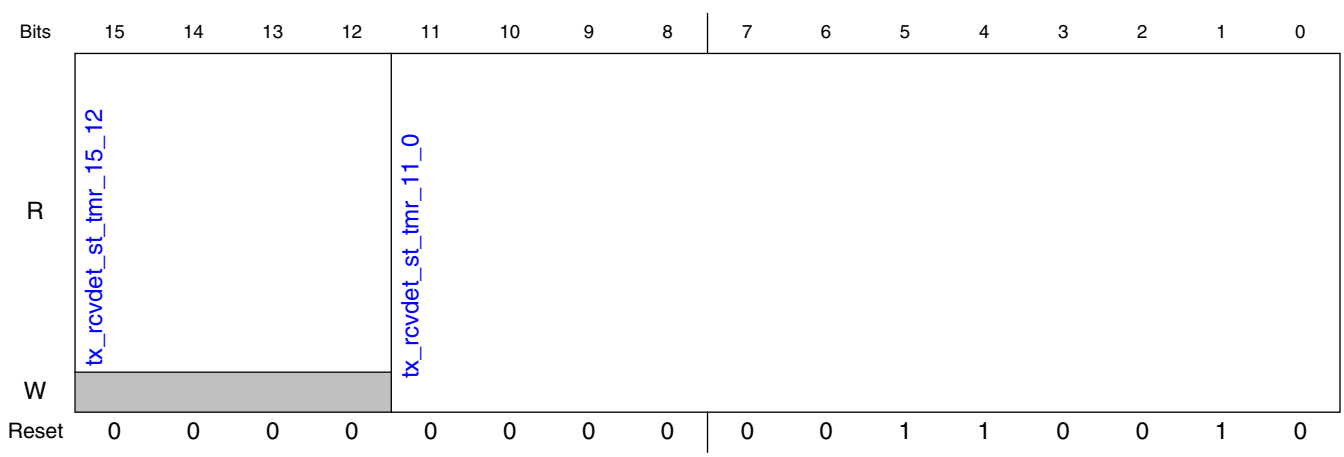
Field	Function
15-12 tx_rcvdet_en_tmr_15_12	Reserved
11-0 tx_rcvdet_en_tmr_11_0	Enable wait time value: This is the number of clocks the receiver detect state machine waits between driving the

13.4.10.2.76 Transmit receiver detect start timer register (lane0_tx_rcvdet_st_tmr - lane3_tx_rcvdet_st_tmr)

13.4.10.2.76.1 Offset

Register	Offset
lane0_tx_rcvdet_st_tmr	4123h
lane1_tx_rcvdet_st_tmr	4523h
lane2_tx_rcvdet_st_tmr	4923h
lane3_tx_rcvdet_st_tmr	4D23h

13.4.10.2.76.2 Diagram



13.4.10.2.76.3 Fields

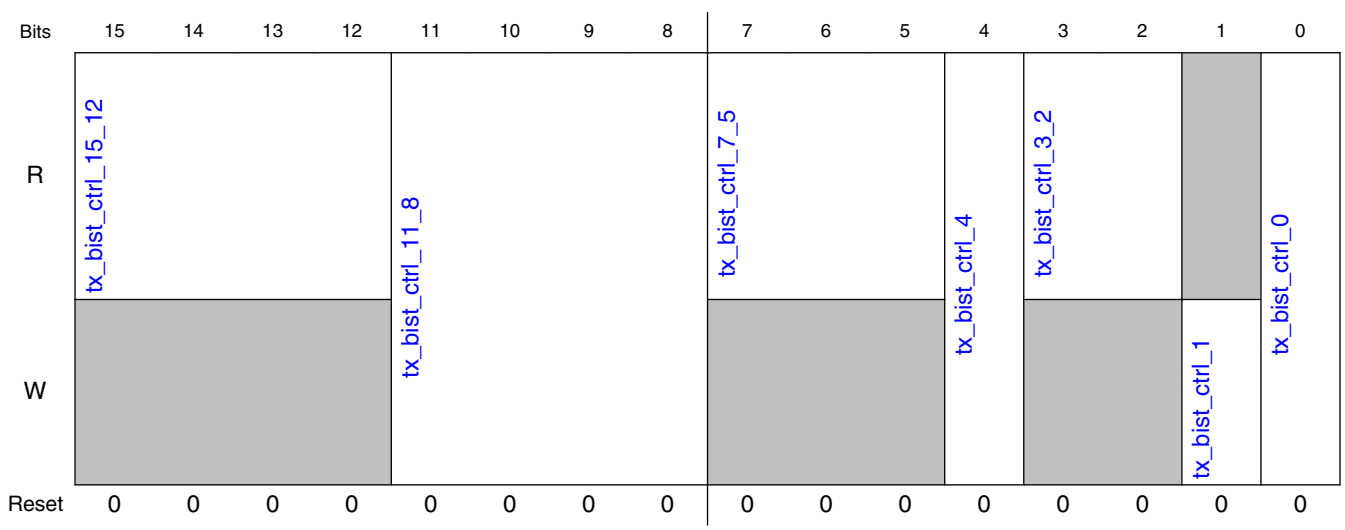
Field	Function
15-12 tx_rcvdet_st_tmr_15_12	Reserved
11-0 tx_rcvdet_st_tmr_11_0	Start wait time value: This is the number of clocks the receiver detect state machine waits between driving the

13.4.10.2.77 Transmit BIST control register (lane0_tx_bist_ctrl - lane3_tx_bist_ctrl)

13.4.10.2.77.1 Offset

Register	Offset
lane0_tx_bist_ctrl	4140h
lane1_tx_bist_ctrl	4540h
lane2_tx_bist_ctrl	4940h
lane3_tx_bist_ctrl	4D40h

13.4.10.2.77.2 Diagram



13.4.10.2.77.3 Fields

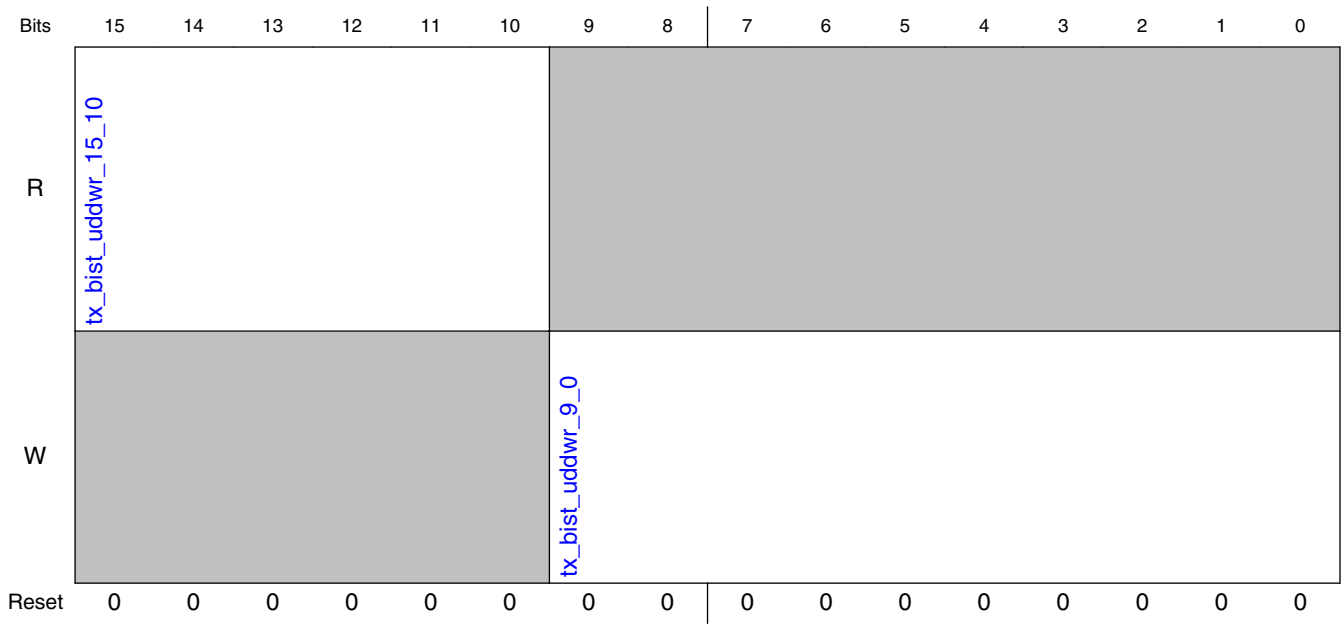
Field	Function
15-12 tx_bist_ctrl_15_12	Reserved
11-8 tx_bist_ctrl_11_8	Transmitter BIST mode: Controls which mode the BIST will operate in. The value of this field must match the corresponding field for the receive BIST controller. The following are the values used for this field, and what BIST mode they correspond to.
7-5 tx_bist_ctrl_7_5	Reserved
4 tx_bist_ctrl_4	Transmitter BIST force error: When this bit transitions from 1b0 to 1b1, the transmit BIST controller will force an error to be transmitted from the BIST logic, by inverting one of the parallel data bits.
3-2 tx_bist_ctrl_3_2	Reserved
1 tx_bist_ctrl_1	Transmitter BIST user defined data FIFO clear: Writing a 1b1 to this bit will clear the transmitter BIST user defined data FIFO.
0 tx_bist_ctrl_0	Transmitter BIST enable: This bit enables the transmitter BIST function.

13.4.10.2.78 Transmit BIST user defined data write register (lane0_tx_bist_uddwr - lane3_tx_bist_uddwr)

13.4.10.2.78.1 Offset

Register	Offset
lane0_tx_bist_uddwr	4141h
lane1_tx_bist_uddwr	4541h
lane2_tx_bist_uddwr	4941h
lane3_tx_bist_uddwr	4D41h

13.4.10.2.78.2 Diagram



13.4.10.2.78.3 Fields

Field	Function
15-10 tx_bist_uddwr_15_10	Reserved
9-0 tx_bist_uddwr_9_0	Transmitter BIST user defined data: Writing a data word to this field will result in that data word being placed in the next available position in the transmitter BIST user defined data FIFO. Note, when in 20 bit mode, all 10 of these bits are used. When in 16 bit mode, only the least significant 8 bits are used. Note that the FIFO in this implementation is 18 words deep. It is up to the user to not write more than 18 words of data into this FIFO. Exceeding this amount of data will generate unpredictable results.

13.4.10.2.79 Transmit BIST PRBS seed 0 register (lane0_tx_bist_seed0 - lane3_tx_bist_seed0)

13.4.10.2.79.1 Offset

Register	Offset
lane0_tx_bist_seed0	4142h
lane1_tx_bist_seed0	4542h
lane2_tx_bist_seed0	4942h
lane3_tx_bist_seed0	4D42h

13.4.10.2.79.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_bist_seed0_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

13.4.10.2.79.3 Fields

Field	Function
15-0 tx_bist_seed0_15_0	Transmitter BIST PRBS seed (15:0): When the BIST is in PRBS mode, this field provides a seed for the PRBS, such that different lanes can have different BIST patterns. Note that this field contains the least significant 16 bits of the full 31 bit seed value.

13.4.10.2.80 Transmit BIST PRBS seed 1 register (lane0_tx_bist_seed1 - lane3_tx_bist_seed1)

13.4.10.2.80.1 Offset

Register	Offset
lane0_tx_bist_seed1	4143h
lane1_tx_bist_seed1	4543h
lane2_tx_bist_seed1	4943h
lane3_tx_bist_seed1	4D43h

13.4.10.2.80.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_bist_seed1_15	tx_bist_seed1_14_0														
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.80.3 Fields

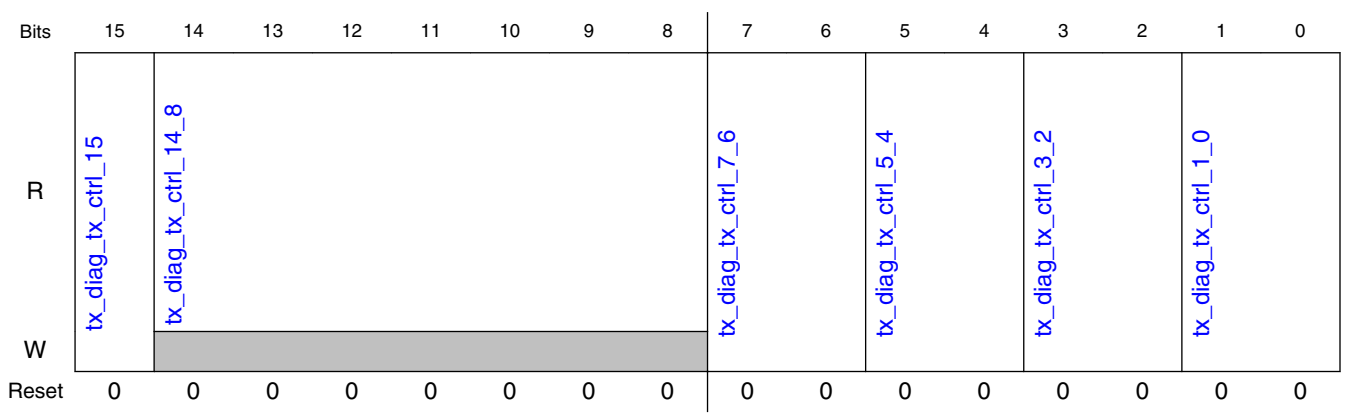
Field	Function
15 tx_bist_seed1_15	Reserved
14-0 tx_bist_seed1_14_0	Transmitter BIST PRBS seed (30:16): When the BIST is in PRBS mode, this field provides a seed for the PRBS, such that different lanes can have different BIST patterns. Note that this field contains the most significant 15 bits of the full 31 bit seed value.

13.4.10.2.81 TX control register (lane0_tx_diag_tx_ctrl - lane3_tx_diag_tx_ctrl)

13.4.10.2.81.1 Offset

Register	Offset
lane0_tx_diag_tx_ctrl	41E0h
lane1_tx_diag_tx_ctrl	45E0h
lane2_tx_diag_tx_ctrl	49E0h
lane3_tx_diag_tx_ctrl	4DE0h

13.4.10.2.81.2 Diagram



13.4.10.2.81.3 Fields

Field	Function
15	TX serializer clock invert: This bit is used to optionally invert the serializer clock (

Table continues on the next page...

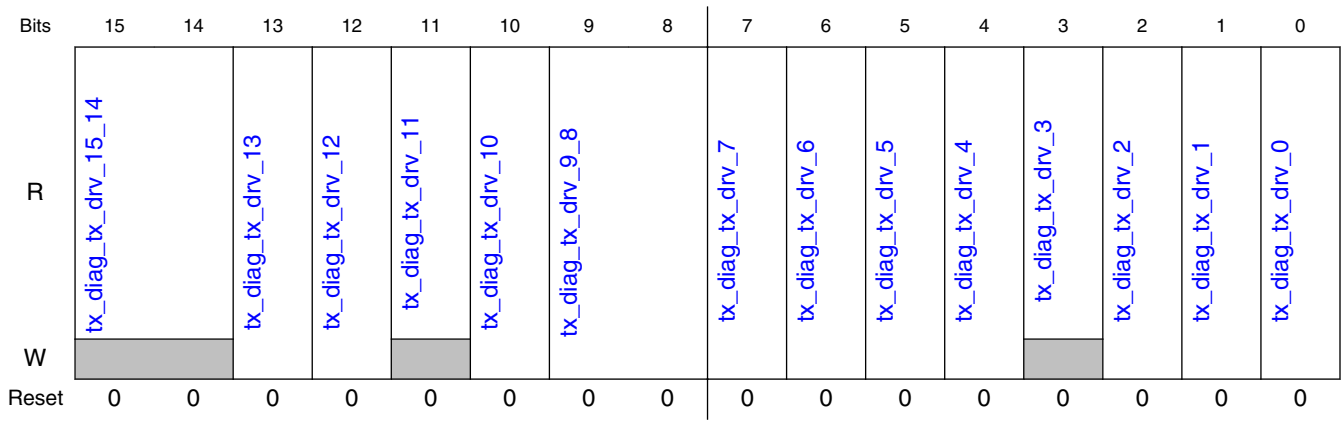
Field	Function
tx_diag_tx_ctrl_15	
14-8 tx_diag_tx_ctrl_14_8	Reserved
7-6 tx_diag_tx_ctrl_7_6	TX interface subrate standard mode 3: This field specifies the transmitter subrate when generating the
5-4 tx_diag_tx_ctrl_5_4	TX interface subrate standard mode 2: This field specifies the transmitter subrate when generating the
3-2 tx_diag_tx_ctrl_3_2	TX interface subrate standard mode 1: This field specifies the transmitter subrate when generating the
1-0 tx_diag_tx_ctrl_1_0	TX interface subrate standard mode 0: This field specifies the transmitter subrate when generating the

13.4.10.2.82 TX driver control register (lane0_tx_diag_tx_drv - lane3_tx_diag_tx_drv)

13.4.10.2.82.1 Offset

Register	Offset
lane0_tx_diag_tx_drv	41E1h
lane1_tx_diag_tx_drv	45E1h
lane2_tx_diag_tx_drv	49E1h
lane3_tx_diag_tx_drv	4DE1h

13.4.10.2.82.2 Diagram



13.4.10.2.82.3 Fields

Field	Function
15-14 tx_diag_tx_drv_15_14	Reserved
13 tx_diag_tx_drv_13	Transmitter reset pull down override enable: When this bit is active (1b1), the transmitter reset pull down override bit in this register will drive the
12 tx_diag_tx_drv_12	Transmitter reset pull down override : When enabled by the transmitter reset pull down override enable bit in this register, this bit will drive the
11 tx_diag_tx_drv_11	Reserved
10 tx_diag_tx_drv_10	TX driver programmable boost enable: Enables the programmable boost function.
9-8 tx_diag_tx_drv_9_8	TX driver programmable boost level: When the programmable boost function is enabled, this signal controls the level of the boost.
7 tx_diag_tx_drv_7	TX driver LDO bandgap dependent feedback reference enable: Enables BG dependent feedback reference to LDO.
6 tx_diag_tx_drv_6	TX driver LDO bandgap dependent reference enable: Enables BG dependent reference to LDO.
5 tx_diag_tx_drv_5	TX driver LDO VDD dependent feedback reference enable: Enables VDD dependent feedback reference to LDO.

Table continues on the next page...

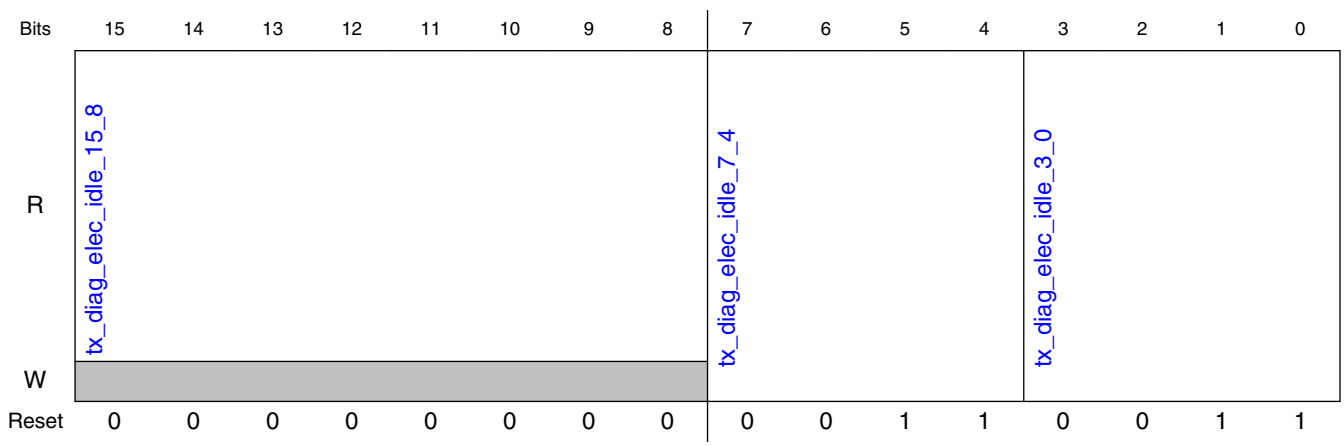
Field	Function
4 tx_diag_tx_drv_4	TX driver LDO VDD dependent reference enable: Enables VDD dependent reference to LDO.
3 tx_diag_tx_drv_3	Reserved
2 tx_diag_tx_drv_2	TD driver polarity control: Controls the polarity of the driver pull-up/down.
1 tx_diag_tx_drv_1	TX pre-driver pull up control: When the pre-driver is disabled, this bit controls the state of the pre-driver output, by controlling the
0 tx_diag_tx_drv_0	TX driver margin type: Selects the margining type the driver will operate in, by controlling the

13.4.10.2.83 TX electrical idle diagnostic register (lane0_tx_diag_elec_idle - lane3_tx_diag_elec_idle)

13.4.10.2.83.1 Offset

Register	Offset
lane0_tx_diag_elec_idle	41E2h
lane1_tx_diag_elec_idle	45E2h
lane2_tx_diag_elec_idle	49E2h
lane3_tx_diag_elec_idle	4DE2h

13.4.10.2.83.2 Diagram



13.4.10.2.83.3 Fields

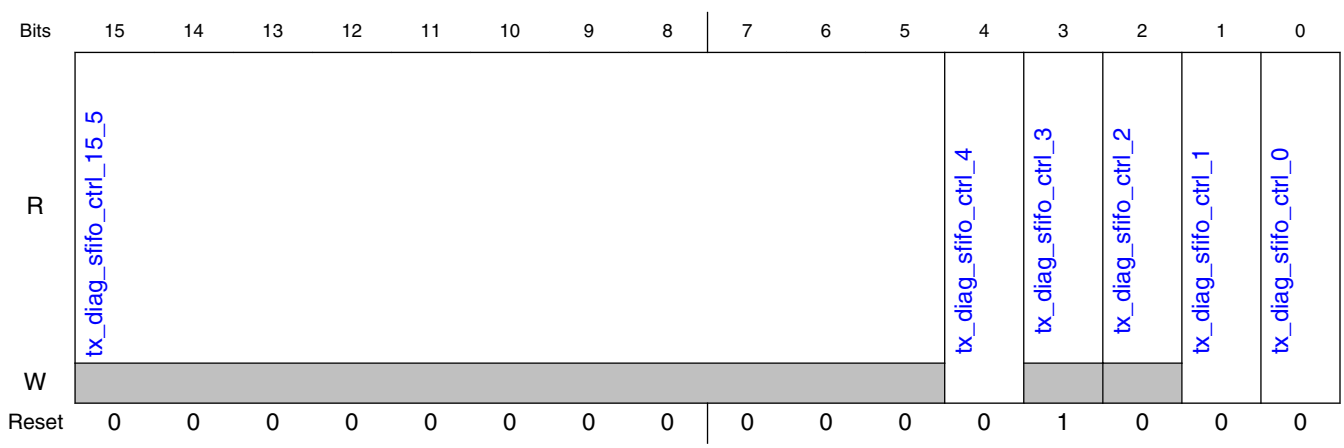
Field	Function
15-8 tx_diag_elec_idle_15_8	Reserved
7-4 tx_diag_elec_idle_7_4	TX electrical idle exit delay : This field controls the amount of additional delay added to the electrical idle signal after the sync FIFO, when exiting the electrical idle state. The following are the valid values that can be used for this field, and the corresponding delays. Note that in the cases of 1/2 clock cycle delays, this is implemented on the falling edge of the clock, so the result is a function of the duty cycle of the clock.
3-0 tx_diag_elec_idle_3_0	TX electrical idle entry delay : This field controls the amount of additional delay added to the electrical idle signal after the sync FIFO, when entering the electrical idle state. The following are the valid values that can be used for this field, and the corresponding delays. Note that in the cases of 1/2 clock cycle delays, this is implemented on the falling edge of the clock, so the result is a function of the duty cycle of the clock.

13.4.10.2.84 TX sync FIFO diagnostic control register (lane0_tx_diag_sfifo_ctrl - lane3_tx_diag_sfifo_ctrl)

13.4.10.2.84.1 Offset

Register	Offset
lane0_tx_diag_sfifo_ctrl	41E3h
lane1_tx_diag_sfifo_ctrl	45E3h
lane2_tx_diag_sfifo_ctrl	49E3h
lane3_tx_diag_sfifo_ctrl	4DE3h

13.4.10.2.84.2 Diagram



13.4.10.2.84.3 Fields

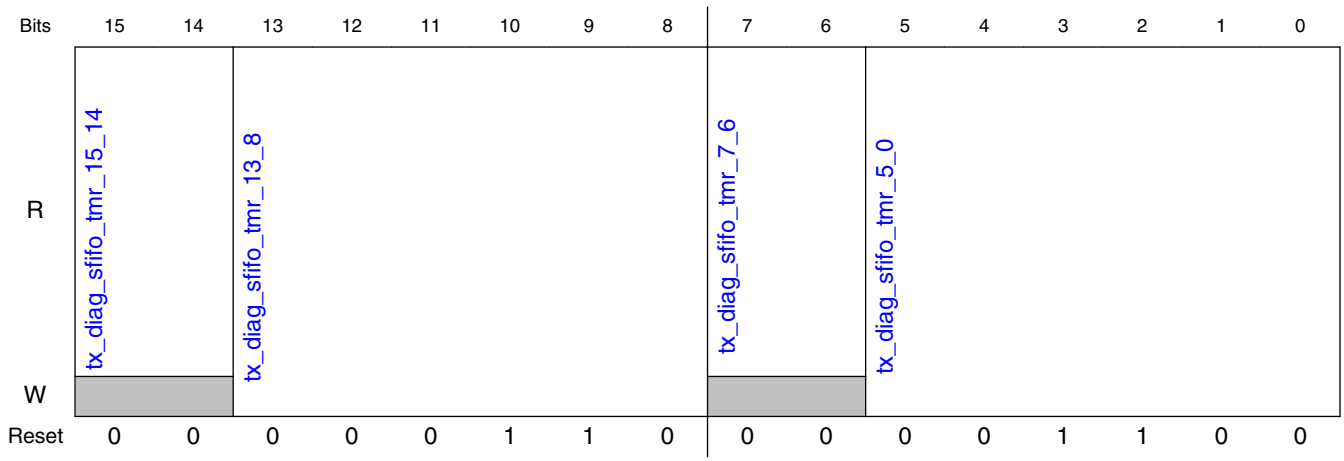
Field	Function
15-5 tx_diag_sfifo_ctrl_15_5	Reserved
4 tx_diag_sfifo_ctrl_4	FIFO enqueue pointer bump: This bit can be used to decrement the enqueue pointer relative to the dequeue pointer, for diagnostic purposes. Changing this bit from a value of 1b0 to 1b1 will trigger a single decrement of the enqueue pointer.
3 tx_diag_sfifo_ctrl_3	FIFO alignment error: This bit indicates if a FIFO overflow or FIFO underflow condition currently exists. Note that during the FIFO alignment process, this bit will toggle. This bit is driven directly by the
2 tx_diag_sfifo_ctrl_2	FIFO alignment acknowledge: This bit indicates that the FIFO alignment process is complete, as initiated either automatically by the hardware of the FIFO alignment enable override bits in this register. This bit is driven directly by the
1 tx_diag_sfifo_ctrl_1	FIFO alignment enable override enable: This bit enables the FIFO alignment enable override register to drive the
0 tx_diag_sfifo_ctrl_0	FIFO alignment enable override: When enabled by the FIFO alignment enable override enable bit in this register, this bit directly controls the

13.4.10.2.85 TX sync FIFO diagnostic timer register (lane0_tx_diag_sfifo_tmr - lane3_tx_diag_sfifo_tmr)

13.4.10.2.85.1 Offset

Register	Offset
lane0_tx_diag_sfifo_tmr	41E4h
lane1_tx_diag_sfifo_tmr	45E4h
lane2_tx_diag_sfifo_tmr	49E4h
lane3_tx_diag_sfifo_tmr	4DE4h

13.4.10.2.85.2 Diagram



13.4.10.2.85.3 Fields

Field	Function
15-14 tx_diag_sfifo_tmr_15_14	Reserved
13-8 tx_diag_sfifo_tmr_13_8	FIFO alignment settle delay: This field specifies the number of clocks to wait for a prior change to the enqueue pointer to complete before initiating the check phase of the alignment procedure in the sync FIFO. It drives the
7-6 tx_diag_sfifo_tmr_7_6	Reserved
5-0 tx_diag_sfifo_tmr_5_0	FIFO alignment detect delay: This field specifies the number of clocks to wait in the delay state for each phase of the alignment procedure in the sync FIFO. It drives the

13.4.10.2.86 TX receiver detect tuning register (lane0_tx_diag_rdvdet_tune - lane3_tx_diag_rdvdet_tune)

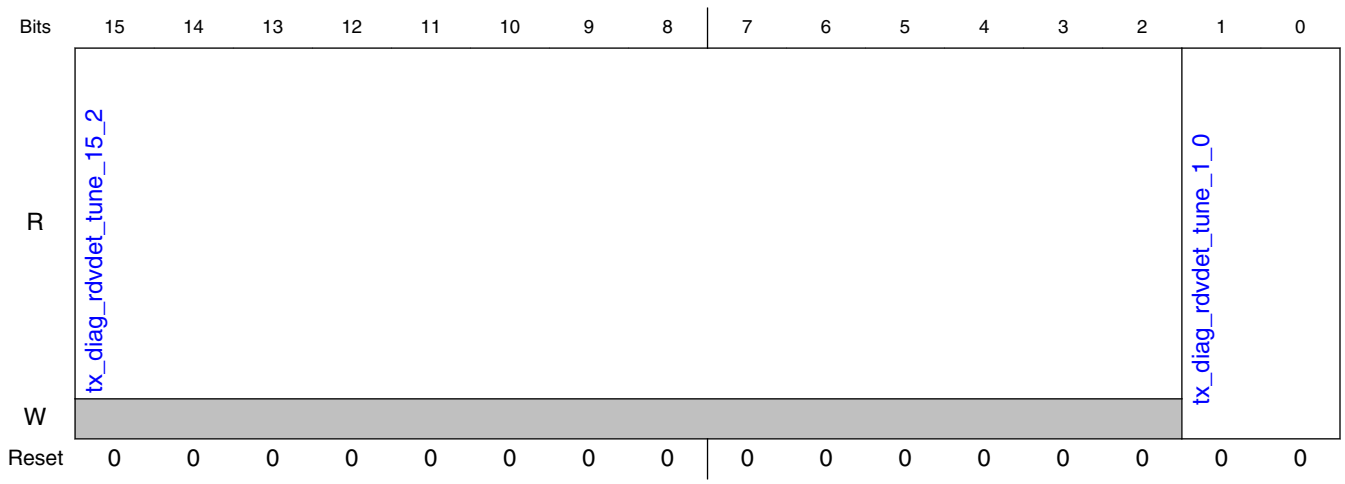
13.4.10.2.86.1 Offset

Register	Offset
lane0_tx_diag_rdvdet_tune	41E5h
lane1_tx_diag_rdvdet_tune	45E5h

Table continues on the next page...

Register	Offset
lane2_tx_diag_rdvdet_tune	49E5h
lane3_tx_diag_rdvdet_tune	4DE5h

13.4.10.2.86.2 Diagram



13.4.10.2.86.3 Fields

Field	Function
15-2 tx_diag_rdvdet_tune_15_2	Reserved
1-0 tx_diag_rdvdet_tune_1_0	Receiver detect reference DAC voltage: This field sets the receiver detect reference voltage for the receiver detect comparator. This field drives the

13.4.10.2.87 Transmitter control reset diagnostic register (lane0_tx_diag_rst_diag - lane3_tx_diag_rst_diag)

13.4.10.2.87.1 Offset

Register	Offset
lane0_tx_diag_rst_diag	41E6h
lane1_tx_diag_rst_diag	45E6h

Table continues on the next page...

Register	Offset
lane2_tx_diag_rst_diag	49E6h
lane3_tx_diag_rst_diag	4DE6h

13.4.10.2.87.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_diag_rst_diag_15_9							tx_diag_rst_diag_8	tx_diag_rst_diag_7	tx_diag_rst_diag_6	tx_diag_rst_diag_5	tx_diag_rst_diag_4	tx_diag_rst_diag_3	tx_diag_rst_diag_2	tx_diag_rst_diag_1	tx_diag_rst_diag_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.87.3 Fields

Field	Function
15-9 tx_diag_rst_diag_15_9	Reserved
8 tx_diag_rst_diag_8	Current state of the dsync_power_reset_n reset. Note that when the power islands are enabled by driving
7 tx_diag_rst_diag_7	Current state of the tfunc_power_reset_n reset. Note that when the power islands are enabled by driving
6 tx_diag_rst_diag_6	Current state of the xcal1_power_reset_n reset. Note that when the power islands are enabled by driving
5 tx_diag_rst_diag_5	Current state of the xcaln_power_reset_n reset. Note that when the power islands are enabled by driving
4 tx_diag_rst_diag_4	Current state of the txda_tx_clk_reset_n reset.
3	Current state of the tx_dig_reset_n reset.

Table continues on the next page...

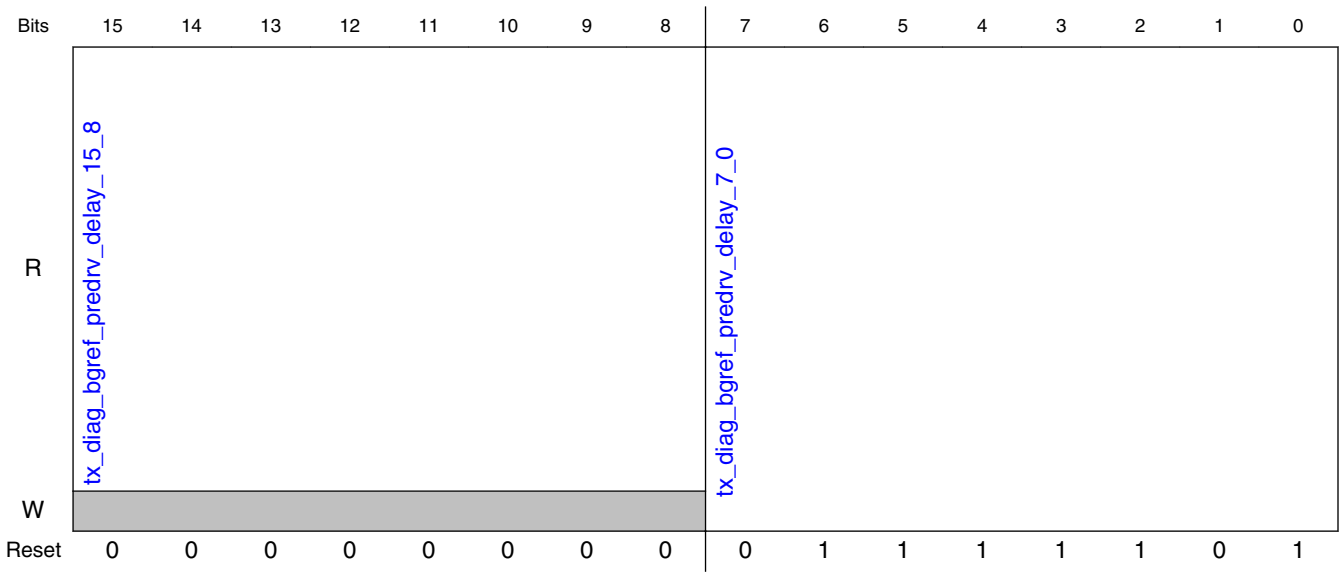
Field	Function
tx_diag_rst_diag_3	
2 tx_diag_rst_diag_2	Current state of the tx_sync_fifo_deq_rst_n reset.
1 tx_diag_rst_diag_1	Current state of the tx_sync_fifo_enq_rst_n reset.
0 tx_diag_rst_diag_0	Current state of the tx_lfps_reset_n reset.

13.4.10.2.88 TX bandgap reference and pre-drive enable delay register (lane0_tx_diag_bgref_predrv_delay - lane3_tx_diag_bgref_predrv_delay)

13.4.10.2.88.1 Offset

Register	Offset
lane0_tx_diag_bgref_predrv_delay	41E7h
lane1_tx_diag_bgref_predrv_delay	45E7h
lane2_tx_diag_bgref_predrv_delay	49E7h
lane3_tx_diag_bgref_predrv_delay	4DE7h

13.4.10.2.88.2 Diagram



13.4.10.2.88.3 Fields

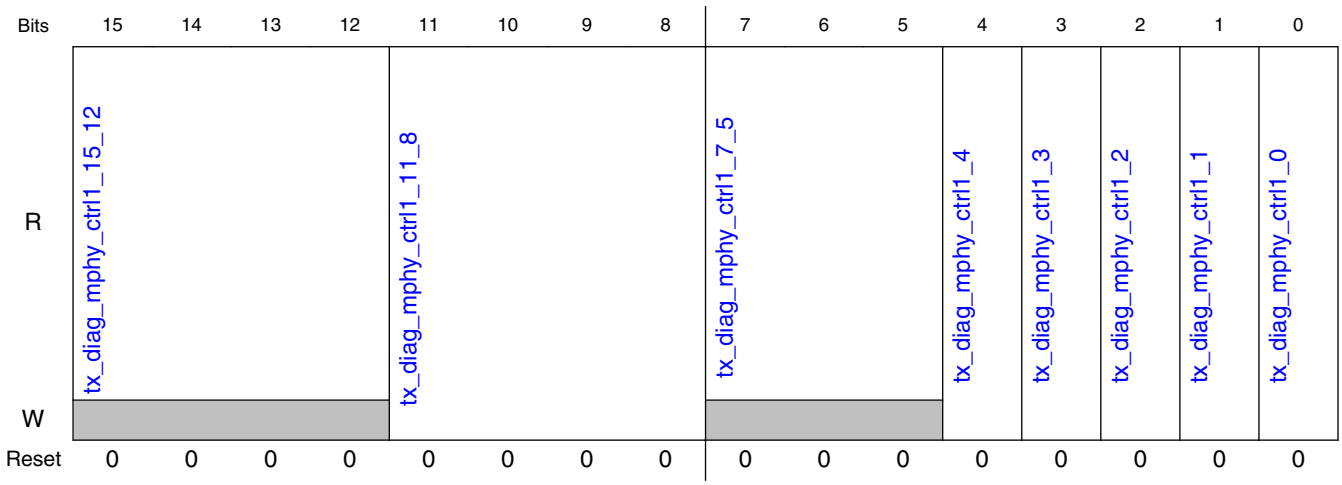
Field	Function
15-8 tx_diag_bgref_predrv_delay_15_8	Reserved
7-0 tx_diag_bgref_predrv_delay_7_0	TX bandgap reference and pre-drive enable delay: This field specifies the number of PSM clocks between when

13.4.10.2.89 TX MPHY control register 1 (lane0_tx_diag_mphy_ctrl1 - lane 3_tx_diag_mphy_ctrl1)

13.4.10.2.89.1 Offset

Register	Offset
lane0_tx_diag_mphy_ctrl1	41F0h
lane1_tx_diag_mphy_ctrl1	45F0h
lane2_tx_diag_mphy_ctrl1	49F0h
lane3_tx_diag_mphy_ctrl1	4DF0h

13.4.10.2.89.2 Diagram



13.4.10.2.89.3 Fields

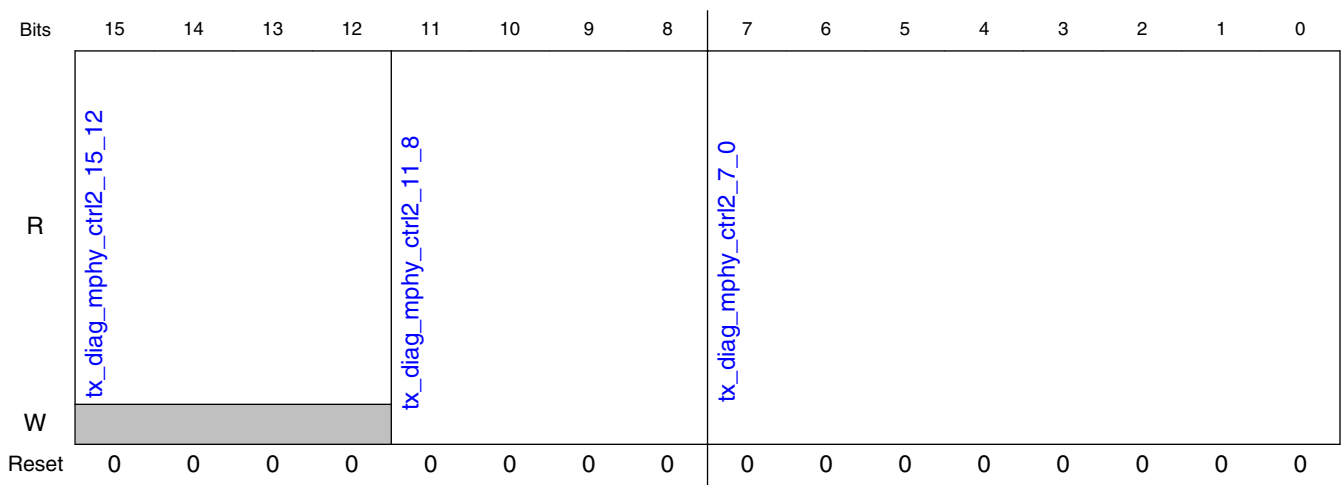
Field	Function
15-12 tx_diag_mphy_ctrl1_15_12	Reserved
11-8 tx_diag_mphy_ctrl1_11_8	Register definition to be provided by the analog team (Jira CCUPHY-2654)
7-5 tx_diag_mphy_ctrl1_7_5	Reserved
4 tx_diag_mphy_ctrl1_4	MPHY small amplitude mode: Enables the driver LDO small amplitude mode, which is used for eDP, AUX, ARC and MPHY, by driving the
3 tx_diag_mphy_ctrl1_3	AUX bias current enable: Enables the bias from the bias block of the analog boost circuit for the AUX receiver, by driving the
2 tx_diag_mphy_ctrl1_2	LDO no-load current reduce: Reduces the no-load current from the driver LDO in order to reduce the settling time, by driving the
1 tx_diag_mphy_ctrl1_1	Register definition to be provided by the analog team (Jira CCUPHY-2654)
0 tx_diag_mphy_ctrl1_0	MPHY high load current mode enable: Enables the high load current mode for the driver LDO. This is intended for non-terminated (NT) HS mode operation, by driving the

13.4.10.2.90 TX MPHY control register 2 (lane0_tx_diag_mphy_ctrl2 - lane 3_tx_diag_mphy_ctrl2)

13.4.10.2.90.1 Offset

Register	Offset
lane0_tx_diag_mphy_ctrl2	41F1h
lane1_tx_diag_mphy_ctrl2	45F1h
lane2_tx_diag_mphy_ctrl2	49F1h
lane3_tx_diag_mphy_ctrl2	4DF1h

13.4.10.2.90.2 Diagram



13.4.10.2.90.3 Fields

Field	Function
15-12 tx_diag_mphy_ctrl2_15_12	Reserved
11-8 tx_diag_mphy_ctrl2_11_8	Register definition to be provided by the analog team (Jira CCUPHY-2654)

Table continues on the next page...

Field	Function
7-0 tx_diag_mphy_ctrl2_7_0	MPHY slew rate control: Controls the slew rate of the line in MPHY-G1 mode and MHL-eCBUS mode, by driving

13.4.10.2.91 TX driver LDO programming register (lane0_tx_diag_drv_ldo_prog - lane3_tx_diag_drv_ldo_prog)

13.4.10.2.91.1 Offset

Register	Offset
lane0_tx_diag_drv_ldo_prog	41F4h
lane1_tx_diag_drv_ldo_prog	45F4h
lane2_tx_diag_drv_ldo_prog	49F4h
lane3_tx_diag_drv_ldo_prog	4DF4h

13.4.10.2.91.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_diag_drv_ldo_prog_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.91.3 Fields

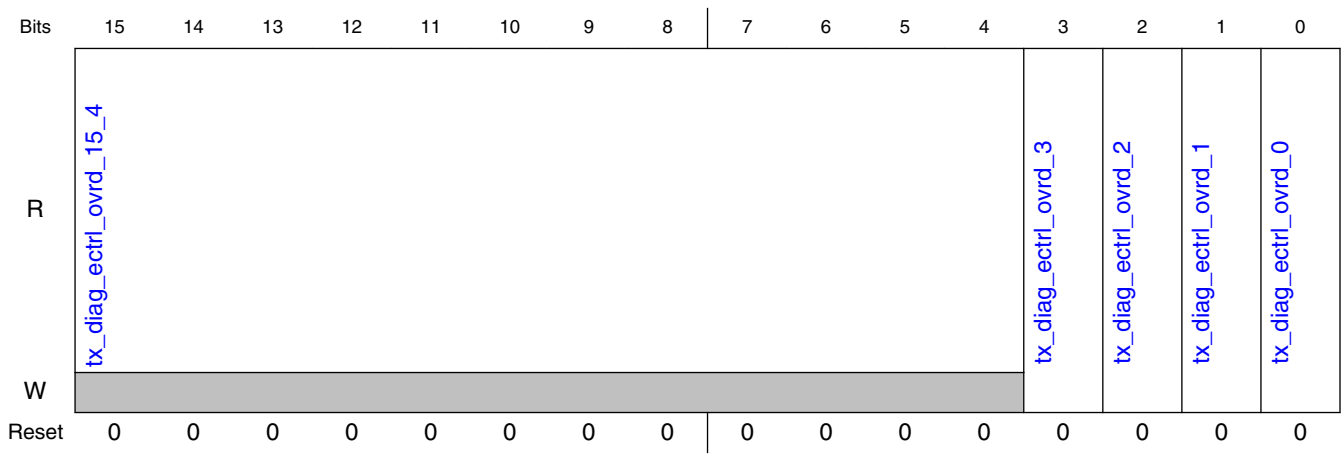
Field	Function
15-0 tx_diag_drv_ldo_prog_15_0	Register definition to be provided by the analog team (Jira CCUPHY-2654)

13.4.10.2.92 TX extra enable control override register (lane0_tx_diag_ectrl_ovrd - lane3_tx_diag_ectrl_ovrd)

13.4.10.2.92.1 Offset

Register	Offset
lane0_tx_diag_ectrl_ovrd	41F5h
lane1_tx_diag_ectrl_ovrd	45F5h
lane2_tx_diag_ectrl_ovrd	49F5h
lane3_tx_diag_ectrl_ovrd	4DF5h

13.4.10.2.92.2 Diagram



13.4.10.2.92.3 Fields

Field	Function
15-4 tx_diag_ectrl_ovrd_15_4	Reserved
3 tx_diag_ectrl_ovrd_3	Driver pre-drive enable override enable : When active (1b1), the driver pre-drive enable override bit in this register will drive the
2 tx_diag_ectrl_ovrd_2	Driver pre-drive enable override : When enabled by the driver pre-drive enable override enable bit in this register, this bit will drive the
1 tx_diag_ectrl_ovrd_1	Bandgap refrence enable override enable : When active (1b1), the bandgap reference enable override bit in this register will drive the
0 tx_diag_ectrl_ovrd_0	Bandgap refrence enable override : When enabled by the bandgap reference enable override enable bit in this register, this bit will drive the

13.4.10.2.93 Transmitter digital cover your alternatives register (lane0_tx_diag_dcya - lane3_tx_diag_dcya)

13.4.10.2.93.1 Offset

Register	Offset
lane0_tx_diag_dcya	41FEh
lane1_tx_diag_dcya	45FEh
lane2_tx_diag_dcya	49FEh
lane3_tx_diag_dcya	4DFEh

13.4.10.2.93.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_diag_dcya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.93.3 Fields

Field	Function
15-0 tx_diag_dcya_15_0	Reserved

13.4.10.2.94 Transmitter analog cover your alternatives register (lane0_tx_diag_acya - lane3_tx_diag_acya)

13.4.10.2.94.1 Offset

Register	Offset
lane0_tx_diag_acya	41FFh
lane1_tx_diag_acya	45FFh
lane2_tx_diag_acya	49FFh
lane3_tx_diag_acya	4DFFh

13.4.10.2.94.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_diag_acya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.94.3 Fields

Field	Function
15-0 tx_diag_acya_15_0	Reserved

13.4.10.2.95 Receiver A0 power state definition register (lane0_rx_psc_a0 - lane3_rx_psc_a0)

13.4.10.2.95.1 Offset

Register	Offset
lane0_rx_psc_a0	8000h
lane1_rx_psc_a0	8400h
lane2_rx_psc_a0	8800h
lane3_rx_psc_a0	8C00h

13.4.10.2.95.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a0_15	rx_psc_a0_14	rx_psc_a0_13	rx_psc_a0_12	rx_psc_a0_11	rx_psc_a0_10	rx_psc_a0_9	rx_psc_a0_8	rx_psc_a0_7	rx_psc_a0_6	rx_psc_a0_5	rx_psc_a0_4	rx_psc_a0_3	rx_psc_a0_2	rx_psc_a0_1	rx_psc_a0_0
Reset	1	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1

13.4.10.2.95.3 Fields

Field	Function
15 rx_psc_a0_15	RX LFPS / signal detect filter mode: Selects the mode of operation of the LFPS and signal detect filters.
14 rx_psc_a0_14	RX squelch enable: Enables the receiver squelch function. This drives the
13 rx_psc_a0_13	RX LFPS detect enable: Enables the receiver LFPS detect function. This drives the
12 rx_psc_a0_12	RX signal detect extend valid: Indicates the power state where the
11 rx_psc_a0_11	RX signal detect filter enable: Enables the receiver signal detect filter function in the digital receiver controller.
10 rx_psc_a0_10	RX LFPS detect filter enable: Enables the receiver LFPS detect filter function in the digital receiver controller.
9 rx_psc_a0_9	RX clock enable: Enables the receiver circuits related to the high speed clocks.
8 rx_psc_a0_8	RX signal detect enable: Enables the receiver signal detect function. This drives the
7 rx_psc_a0_7	RX equalization enable: Enables the receiver equalization circuits, via the
6 rx_psc_a0_6	RX sampler enable: Enables the receiver circuits related to the sampler.
5 rx_psc_a0_5	RX CDRLF enable: Enables the receiver circuits related to the CDRLF.
4 rx_psc_a0_4	RX bias enable: Enables the receiver circuits related to the bias.
3 rx_psc_a0_3	RX DFE equalization enable: Enables the receiver DFE equalization circuits, via the
2 rx_psc_a0_2	RX PI enable: Enables the receiver circuits related to the PI and associated clocking components.
1 rx_psc_a0_1	RX e path enable (calibration and eye surf only) : Enables the receiver circuits related to the eye plot PI and e path deserializer for calibration and eye surf.
0 rx_psc_a0_0	RX enable: Enables the receiver circuits related to the FE, and Deserializer.

13.4.10.2.96 Receiver A1 power state definition register (lane0_rx_psc_a1 - lane3_rx_psc_a1)

13.4.10.2.96.1 Offset

Register	Offset
lane0_rx_psc_a1	8001h
lane1_rx_psc_a1	8401h
lane2_rx_psc_a1	8801h
lane3_rx_psc_a1	8C01h

13.4.10.2.96.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a1_15	rx_psc_a1_14	rx_psc_a1_13	rx_psc_a1_12	rx_psc_a1_11	rx_psc_a1_10	rx_psc_a1_9	rx_psc_a1_8	rx_psc_a1_7	rx_psc_a1_6	rx_psc_a1_5	rx_psc_a1_4	rx_psc_a1_3	rx_psc_a1_2	rx_psc_a1_1	rx_psc_a1_0
Reset	1	0	0	0	1	0	1	1	1	1	1	1	1	1	0	1

13.4.10.2.96.3 Fields

Field	Function
15 rx_psc_a1_15	RX LFPS / signal detect filter mode
14 rx_psc_a1_14	RX squelch enable
13 rx_psc_a1_13	RX LFPS detect enable
12 rx_psc_a1_12	RX signal detect extend valid
11 rx_psc_a1_11	RX signal detect filter enable
10 rx_psc_a1_10	RX LFPS detect filter enable
9 rx_psc_a1_9	RX clock enable
8 rx_psc_a1_8	RX signal detect enable
7 rx_psc_a1_7	RX equalization enable

Table continues on the next page...

Field	Function
6 rx_psc_a1_6	RX sampler enable
5 rx_psc_a1_5	RX CDRLF enable
4 rx_psc_a1_4	RX bias enable
3 rx_psc_a1_3	RX DFE equalization enable
2 rx_psc_a1_2	RX PI enable
1 rx_psc_a1_1	RX e path enable (calibration and eye surf only)
0 rx_psc_a1_0	RX enable

13.4.10.2.97 Receiver A2 power state definition register (lane0_rx_psc_a2 - lane3_rx_psc_a2)

13.4.10.2.97.1 Offset

Register	Offset
lane0_rx_psc_a2	8002h
lane1_rx_psc_a2	8402h
lane2_rx_psc_a2	8802h
lane3_rx_psc_a2	8C02h

13.4.10.2.97.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a2_15	rx_psc_a2_14	rx_psc_a2_13	rx_psc_a2_12	rx_psc_a2_11	rx_psc_a2_10	rx_psc_a2_9	rx_psc_a2_8	rx_psc_a2_7	rx_psc_a2_6	rx_psc_a2_5	rx_psc_a2_4	rx_psc_a2_3	rx_psc_a2_2	rx_psc_a2_1	rx_psc_a2_0
Reset	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0	0

13.4.10.2.97.3 Fields

Field	Function
15 rx_psc_a2_15	RX LFPS / signal detect filter mode
14 rx_psc_a2_14	RX squelch enable
13 rx_psc_a2_13	RX LFPS detect enable
12 rx_psc_a2_12	RX signal detect extend valid
11 rx_psc_a2_11	RX signal detect filter enable
10 rx_psc_a2_10	RX LFPS detect filter enable
9 rx_psc_a2_9	RX clock enable
8 rx_psc_a2_8	RX signal detect enable
7 rx_psc_a2_7	RX equalization enable
6 rx_psc_a2_6	RX sampler enable
5 rx_psc_a2_5	RX CDRLF enable
4 rx_psc_a2_4	RX bias enable
3 rx_psc_a2_3	RX DFE equalization enable
2 rx_psc_a2_2	RX PI enable
1 rx_psc_a2_1	RX e path enable (calibration and eye surf only)
0 rx_psc_a2_0	RX enable

13.4.10.2.98 Receiver A3 power state definition register (lane0_rx_psc_a3 - lane3_rx_psc_a3)

13.4.10.2.98.1 Offset

Register	Offset
lane0_rx_psc_a3	8003h
lane1_rx_psc_a3	8403h
lane2_rx_psc_a3	8803h
lane3_rx_psc_a3	8C03h

13.4.10.2.98.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a3_15	rx_psc_a3_14	rx_psc_a3_13	rx_psc_a3_12	rx_psc_a3_11	rx_psc_a3_10	rx_psc_a3_9	rx_psc_a3_8	rx_psc_a3_7	rx_psc_a3_6	rx_psc_a3_5	rx_psc_a3_4	rx_psc_a3_3	rx_psc_a3_2	rx_psc_a3_1	rx_psc_a3_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.98.3 Fields

Field	Function
15 rx_psc_a3_15	RX LFPS / signal detect filter mode
14 rx_psc_a3_14	RX squelch enable
13 rx_psc_a3_13	RX LFPS detect enable
12 rx_psc_a3_12	RX signal detect extend valid
11 rx_psc_a3_11	RX signal detect filter enable
10 rx_psc_a3_10	RX LFPS detect filter enable
9 rx_psc_a3_9	RX clock enable
8 rx_psc_a3_8	RX signal detect enable
7 rx_psc_a3_7	RX equalization enable

Table continues on the next page...

Field	Function
6 rx_psc_a3_6	RX sampler enable
5 rx_psc_a3_5	RX CDRLF enable
4 rx_psc_a3_4	RX bias enable
3 rx_psc_a3_3	RX DFE equalization enable
2 rx_psc_a3_2	RX PI enable
1 rx_psc_a3_1	RX e path enable (calibration and eye surf only)
0 rx_psc_a3_0	RX enable

13.4.10.2.99 Receiver A4 power state definition register (lane0_rx_psc_a4 - lane3_rx_psc_a4)

13.4.10.2.99.1 Offset

Register	Offset
lane0_rx_psc_a4	8004h
lane1_rx_psc_a4	8404h
lane2_rx_psc_a4	8804h
lane3_rx_psc_a4	8C04h

13.4.10.2.99.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a4_15	rx_psc_a4_14	rx_psc_a4_13	rx_psc_a4_12	rx_psc_a4_11	rx_psc_a4_10	rx_psc_a4_9	rx_psc_a4_8	rx_psc_a4_7	rx_psc_a4_6	rx_psc_a4_5	rx_psc_a4_4	rx_psc_a4_3	rx_psc_a4_2	rx_psc_a4_1	rx_psc_a4_0
Reset	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.99.3 Fields

Field	Function
15 rx_psc_a4_15	RX LFPS / signal detect filter mode
14 rx_psc_a4_14	RX squelch enable
13 rx_psc_a4_13	RX LFPS detect enable
12 rx_psc_a4_12	RX signal detect extend valid
11 rx_psc_a4_11	RX signal detect filter enable
10 rx_psc_a4_10	RX LFPS detect filter enable
9 rx_psc_a4_9	RX clock enable
8 rx_psc_a4_8	RX signal detect enable
7 rx_psc_a4_7	RX equalization enable
6 rx_psc_a4_6	RX sampler enable
5 rx_psc_a4_5	RX CDRLF enable
4 rx_psc_a4_4	RX bias enable
3 rx_psc_a4_3	RX DFE equalization enable
2 rx_psc_a4_2	RX PI enable
1 rx_psc_a4_1	RX e path enable (calibration and eye surf only)
0 rx_psc_a4_0	RX enable

13.4.10.2.100 Receiver A5 power state definition register (lane0_rx_psc_a5 - lane3_rx_psc_a5)

13.4.10.2.100.1 Offset

Register	Offset
lane0_rx_psc_a5	8005h
lane1_rx_psc_a5	8405h
lane2_rx_psc_a5	8805h
lane3_rx_psc_a5	8C05h

13.4.10.2.100.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_a5_15	rx_psc_a5_14	rx_psc_a5_13	rx_psc_a5_12	rx_psc_a5_11	rx_psc_a5_10	rx_psc_a5_9	rx_psc_a5_8	rx_psc_a5_7	rx_psc_a5_6	rx_psc_a5_5	rx_psc_a5_4	rx_psc_a5_3	rx_psc_a5_2	rx_psc_a5_1	rx_psc_a5_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.100.3 Fields

Field	Function
15 rx_psc_a5_15	RX LFPS / signal detect filter mode
14 rx_psc_a5_14	RX squelch enable
13 rx_psc_a5_13	RX LFPS detect enable
12 rx_psc_a5_12	RX signal detect extend valid
11 rx_psc_a5_11	RX signal detect filter enable
10 rx_psc_a5_10	RX LFPS detect filter enable
9 rx_psc_a5_9	RX clock enable
8 rx_psc_a5_8	RX signal detect enable
7 rx_psc_a5_7	RX equalization enable

Table continues on the next page...

Field	Function
6 rx_psc_a5_6	RX sampler enable
5 rx_psc_a5_5	RX CDRLF enable
4 rx_psc_a5_4	RX bias enable
3 rx_psc_a5_3	RX DFE equalization enable
2 rx_psc_a5_2	RX PI enable
1 rx_psc_a5_1	RX e path enable (calibration and eye surf only)
0 rx_psc_a5_0	RX enable

13.4.10.2.101 Receiver calibration power state definition register (lane0_rx_psc_cal - lane3_rx_psc_cal)

13.4.10.2.101.1 Offset

Register	Offset
lane0_rx_psc_cal	8006h
lane1_rx_psc_cal	8406h
lane2_rx_psc_cal	8806h
lane3_rx_psc_cal	8C06h

13.4.10.2.101.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_cal_15	rx_psc_cal_14	rx_psc_cal_13	rx_psc_cal_12	rx_psc_cal_11	rx_psc_cal_10	rx_psc_cal_9	rx_psc_cal_8	rx_psc_cal_7	rx_psc_cal_6	rx_psc_cal_5	rx_psc_cal_4	rx_psc_cal_3	rx_psc_cal_2	rx_psc_cal_1	rx_psc_cal_0
Reset	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1

13.4.10.2.101.3 Fields

Field	Function
15 rx_psc_cal_15	RX LFPS / signal detect filter mode
14 rx_psc_cal_14	RX squelch enable
13 rx_psc_cal_13	RX LFPS detect enable
12 rx_psc_cal_12	RX signal detect extend valid
11 rx_psc_cal_11	RX signal detect filter enable
10 rx_psc_cal_10	RX LFPS detect filter enable
9 rx_psc_cal_9	RX clock enable
8 rx_psc_cal_8	RX signal detect enable
7 rx_psc_cal_7	RX equalization enable
6 rx_psc_cal_6	RX sampler enable
5 rx_psc_cal_5	RX CDRLF enable
4 rx_psc_cal_4	RX bias enable
3 rx_psc_cal_3	RX DFE equalization enable
2 rx_psc_cal_2	RX PI enable
1 rx_psc_cal_1	RX e path enable (calibration and eye surf only)
0 rx_psc_cal_0	RX enable

13.4.10.2.102 Receiver ready power state definition register (lane0_rx_psc_rdy - lane3_rx_psc_rdy)

13.4.10.2.102.1 Offset

Register	Offset
lane0_rx_psc_rdy	8007h
lane1_rx_psc_rdy	8407h
lane2_rx_psc_rdy	8807h
lane3_rx_psc_rdy	8C07h

13.4.10.2.102.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_psc_rdy_15	rx_psc_rdy_14	rx_psc_rdy_13	rx_psc_rdy_12	rx_psc_rdy_11	rx_psc_rdy_10	rx_psc_rdy_9	rx_psc_rdy_8	rx_psc_rdy_7	rx_psc_rdy_6	rx_psc_rdy_5	rx_psc_rdy_4	rx_psc_rdy_3	rx_psc_rdy_2	rx_psc_rdy_1	rx_psc_rdy_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.102.3 Fields

Field	Function
15 rx_psc_rdy_15	RX LFPS / signal detect filter mode
14 rx_psc_rdy_14	RX squelch enable
13 rx_psc_rdy_13	RX LFPS detect enable
12 rx_psc_rdy_12	RX signal detect extend valid
11 rx_psc_rdy_11	RX signal detect filter enable
10 rx_psc_rdy_10	RX LFPS detect filter enable
9 rx_psc_rdy_9	RX clock enable
8 rx_psc_rdy_8	RX signal detect enable
7 rx_psc_rdy_7	RX equalization enable

Table continues on the next page...

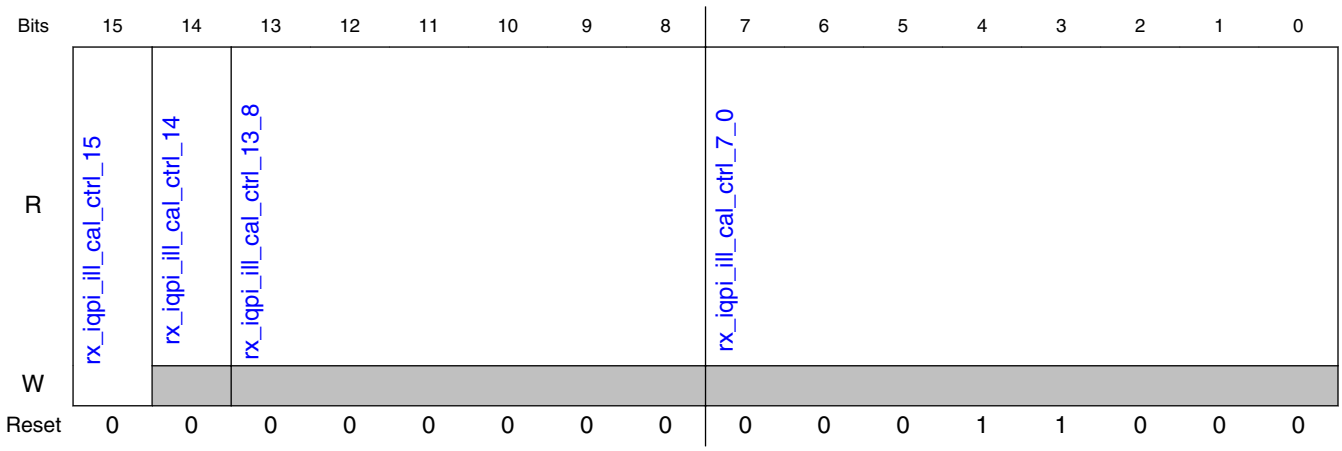
Field	Function
6 rx_psc_rdy_6	RX sampler enable
5 rx_psc_rdy_5	RX CDRLF enable
4 rx_psc_rdy_4	RX bias enable
3 rx_psc_rdy_3	RX DFE equalization enable
2 rx_psc_rdy_2	RX PI enable
1 rx_psc_rdy_1	RX e path enable (calibration and eye surf only)
0 rx_psc_rdy_0	RX enable

13.4.10.2.103 RX IQ PI ILL calibration control register (lane0_rx_iqpi_ill_cal_ctrl - lane3_rx_iqpi_ill_cal_ctrl)

13.4.10.2.103.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_ctrl	8020h
lane1_rx_iqpi_ill_cal_ctrl	8420h
lane2_rx_iqpi_ill_cal_ctrl	8820h
lane3_rx_iqpi_ill_cal_ctrl	8C20h

13.4.10.2.103.2 Diagram



13.4.10.2.103.3 Fields

Field	Function
15 rx_iqpi_ill_cal_ctrl_15	Start ILL calibration: Activating (1b1) this bit will start an ILL calibration process. This bit must remain active until the ILL calibration process is complete (as indicated by the ILL calibration process done bit in this register). To start another ILL calibration process, the ILL calibration process done bit must have gone inactive from any prior calibration process.
14 rx_iqpi_ill_cal_ctrl_14	ILL calibration process done: This bit will be set to 1b1 when the ILL calibration process is complete. It will be cleared by the receiver being reset, or by the deactivation of the Start ILL calibration bit in this register.
13-8 rx_iqpi_ill_cal_ctrl_13_8	Reserved
7-0 rx_iqpi_ill_cal_ctrl_7_0	ILL calibration code: This is the calibration code that was determined by the ILL calibration process. This signal is valid when the ILL calibration process is complete. This field specifies the number of resistors that are switched in. The following are the values for this code:

13.4.10.2.104 RX IQ PI ILL calibration start point register (lane0_rx_iqpi_ill_cal_start - lane3_rx_iqpi_ill_cal_start)

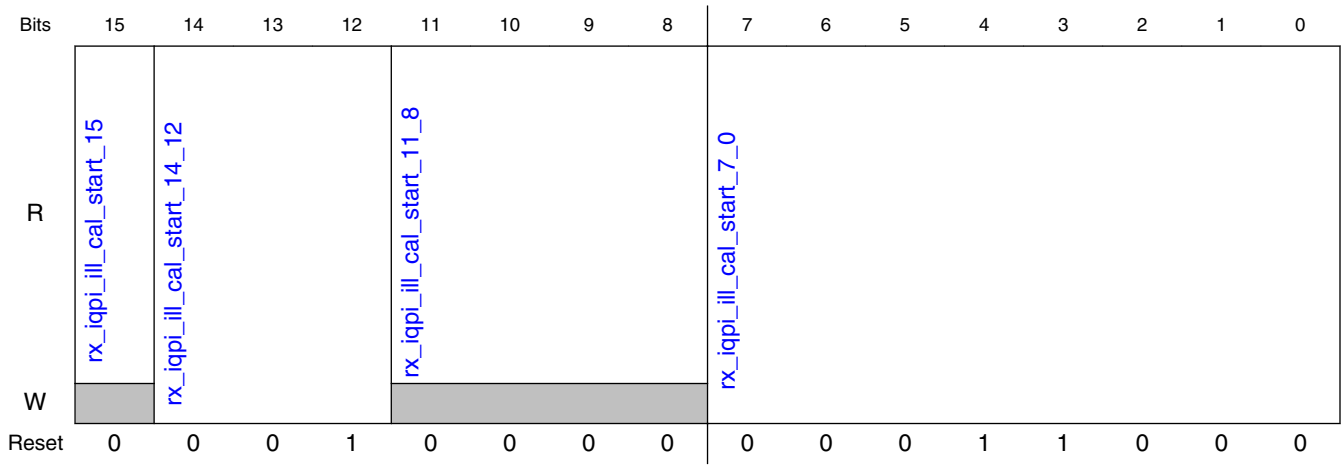
13.4.10.2.104.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_start	8021h
lane1_rx_iqpi_ill_cal_start	8421h

Table continues on the next page...

Register	Offset
lane2_rx_iqpi_ill_cal_start	8821h
lane3_rx_iqpi_ill_cal_start	8C21h

13.4.10.2.104.2 Diagram



13.4.10.2.104.3 Fields

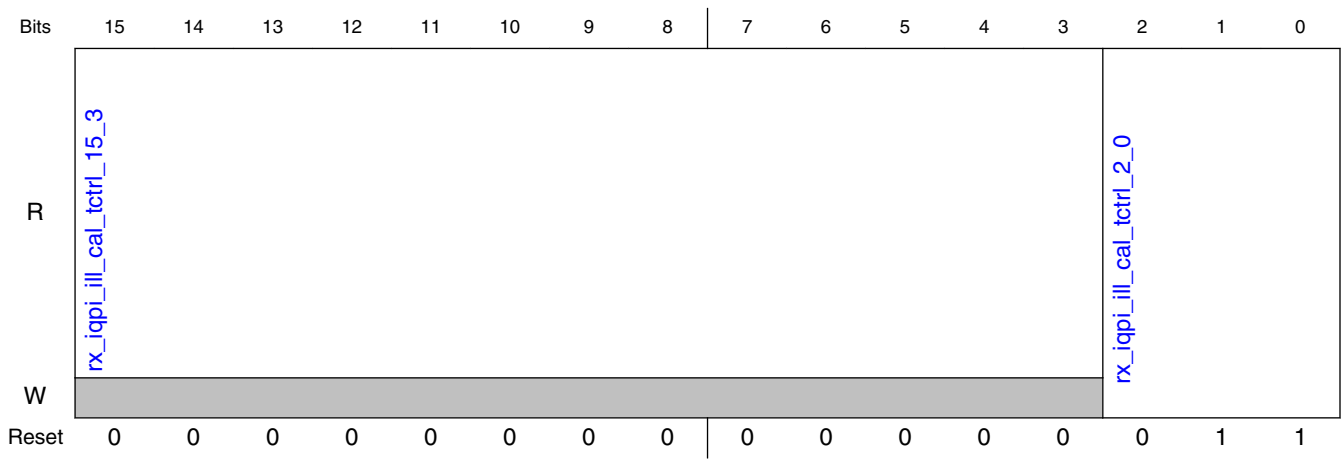
Field	Function
15 rx_iqpi_ill_cal_start_15	Reserved
14-12 rx_iqpi_ill_cal_start_14_12	ILL calibration initial step size control: This field specifies the initial step size for the ILL calibration state machine. The following are the values that can be used in this field, and the corresponding step sizes.
11-8 rx_iqpi_ill_cal_start_11_8	Reserved
7-0 rx_iqpi_ill_cal_start_7_0	ILL calibration code starting point value: This field specifies the starting ILL code that is used by the ILL calibration state machine. The purpose of this value is such that the ILL calibration process starts at a point that is, on average, relatively close to the final calibration point. This allows the calibration time to be reduced, on average. The following are the values for this code:

13.4.10.2.105 RX IQ PI ILL calibration timer control register (lane0_rx_iqpi_ill_cal_tctrl - lane3_rx_iqpi_ill_cal_tctrl)

13.4.10.2.105.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_tctrl	8022h
lane1_rx_iqpi_ill_cal_tctrl	8422h
lane2_rx_iqpi_ill_cal_tctrl	8822h
lane3_rx_iqpi_ill_cal_tctrl	8C22h

13.4.10.2.105.2 Diagram



13.4.10.2.105.3 Fields

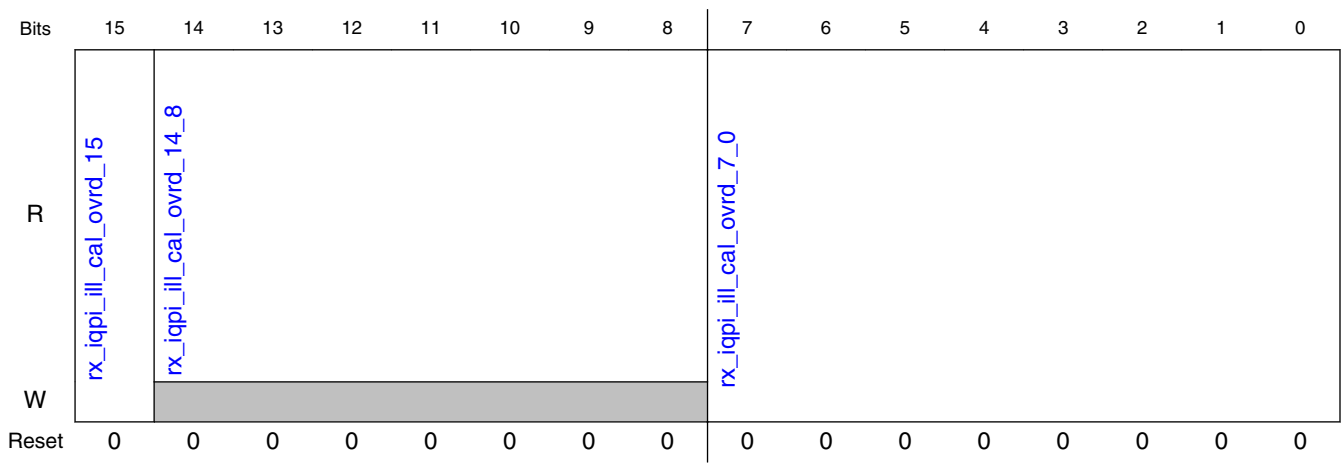
Field	Function
15-3 rx_iqpi_ill_cal_tctrl_15_3	Reserved
2-0 rx_iqpi_ill_cal_tctrl_2_0	ILL calibration initial time scale control: This field specifies the calibration start time scaling factor applied to the ILL calibration when running the initial step size for the calibration code is not set to 1. Setting this value to a value other than 1 will reduce the calibration measurement time by the amount specified below. Then when the final calibration steps are made the full calibration time will be used to get the appropriate amount of resolution.

13.4.10.2.106 RX IQ PI ILL calibration override register (lane0_rx_iqpi_ill_cal_ovrd - lane3_rx_iqpi_ill_cal_ovrd)

13.4.10.2.106.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_ovrd	8023h
lane1_rx_iqpi_ill_cal_ovrd	8423h
lane2_rx_iqpi_ill_cal_ovrd	8823h
lane3_rx_iqpi_ill_cal_ovrd	8C23h

13.4.10.2.106.2 Diagram



13.4.10.2.106.3 Fields

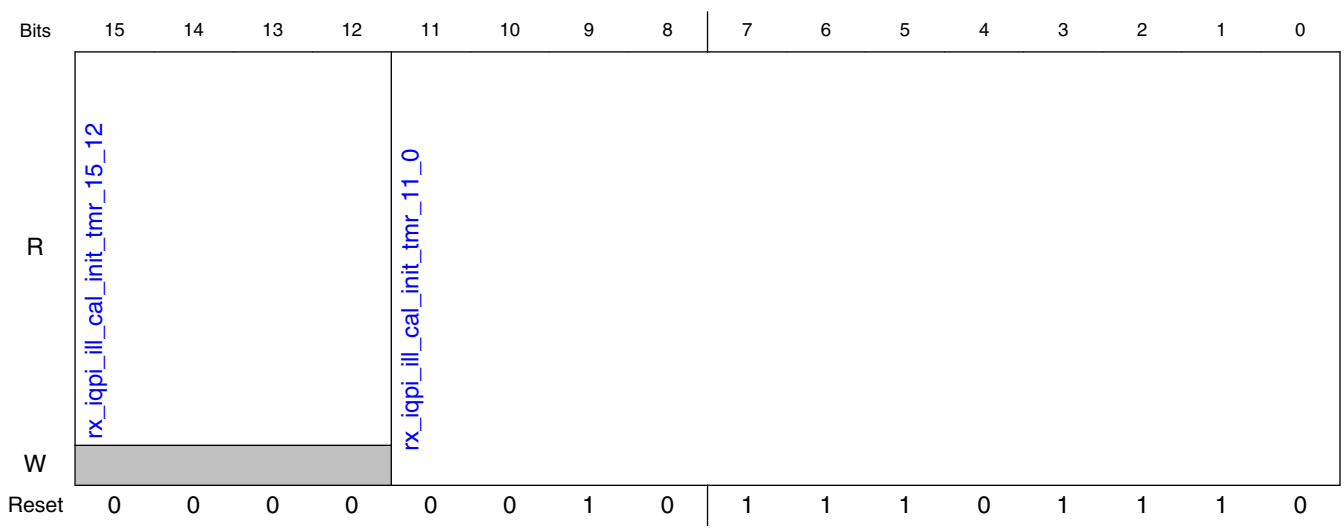
Field	Function
15 rx_iqpi_ill_cal_ovrd_15	ILL calibration code override enable: Activating (1b1) this bit allows the ILL code determined during the automatic ILL calibration process to be overridden by the value driven by the ILL calibration code override value field in this register.
14-8 rx_iqpi_ill_cal_ovrd_14_8	Reserved
7-0 rx_iqpi_ill_cal_ovrd_7_0	ILL calibration code override value: This field is used to override the ILL code determined during the automatic ILL calibration process. The ILL code driven on this field is valid when the ILL calibration code override enable bit in this register is active. The following are the values for this code:

13.4.10.2.107 RX IQ PI ILL calibration initialization timer register (lane0_rx_iqpi_ill_cal_init_tmr - lane3_rx_iqpi_ill_cal_init_tmr)

13.4.10.2.107.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_init_tmr	8024h
lane1_rx_iqpi_ill_cal_init_tmr	8424h
lane2_rx_iqpi_ill_cal_init_tmr	8824h
lane3_rx_iqpi_ill_cal_init_tmr	8C24h

13.4.10.2.107.2 Diagram



13.4.10.2.107.3 Fields

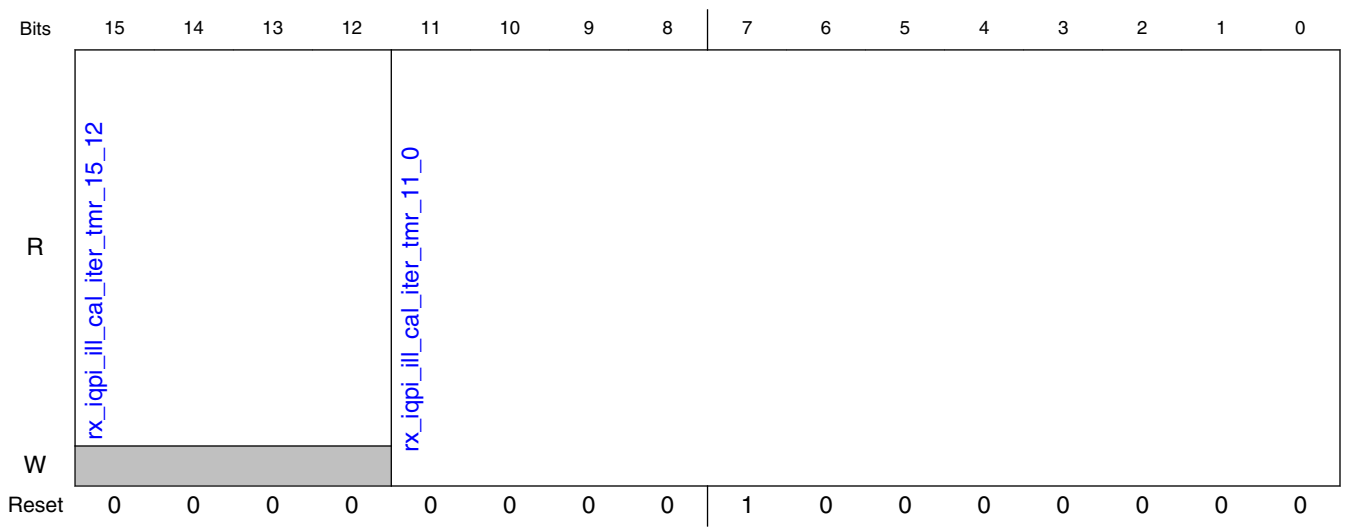
Field	Function
15-12 rx_iqpi_ill_cal_init_tmr_15_12	Reserved
11-0 rx_iqpi_ill_cal_init_tmr_11_0	Initialization wait timer value: This is the number of

13.4.10.2.108 RX IQ PI ILL calibration iteration timer register (lane0_rx_iqpi_ill_cal_iter_tmr - lane3_rx_iqpi_ill_cal_iter_tmr)

13.4.10.2.108.1 Offset

Register	Offset
lane0_rx_iqpi_ill_cal_iter_tmr	8025h
lane1_rx_iqpi_ill_cal_iter_tmr	8425h
lane2_rx_iqpi_ill_cal_iter_tmr	8825h
lane3_rx_iqpi_ill_cal_iter_tmr	8C25h

13.4.10.2.108.2 Diagram



13.4.10.2.108.3 Fields

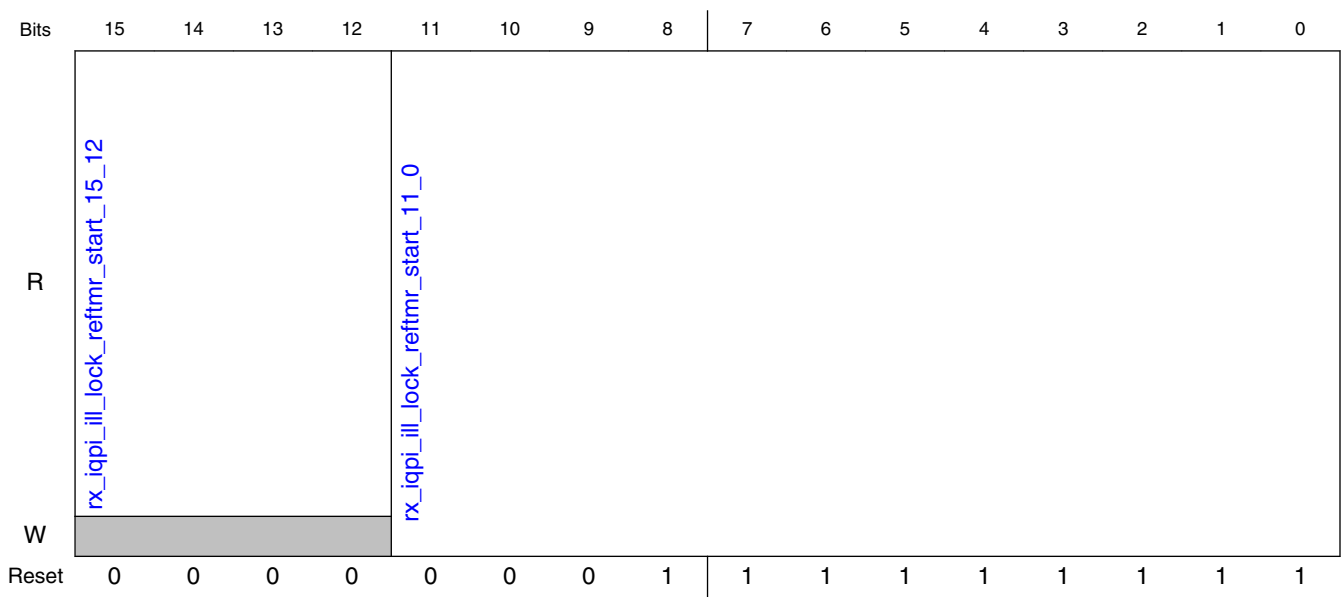
Field	Function
15-12 rx_iqpi_ill_cal_iter_tmr_15_12	Reserved
11-0 rx_iqpi_ill_cal_iter_tmr_11_0	Iteration wait timer value: This is the number of

13.4.10.2.109 RX IQ PI ILL lock reference timer start value register (lane0_rx_iqpi_ill_lock_reftmr_start - lane3_rx_iqpi_ill_lock_reftmr_start)

13.4.10.2.109.1 Offset

Register	Offset
lane0_rx_iqpi_ill_lock_reftmr_start	8026h
lane1_rx_iqpi_ill_lock_reftmr_start	8426h
lane2_rx_iqpi_ill_lock_reftmr_start	8826h
lane3_rx_iqpi_ill_lock_reftmr_start	8C26h

13.4.10.2.109.2 Diagram



13.4.10.2.109.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

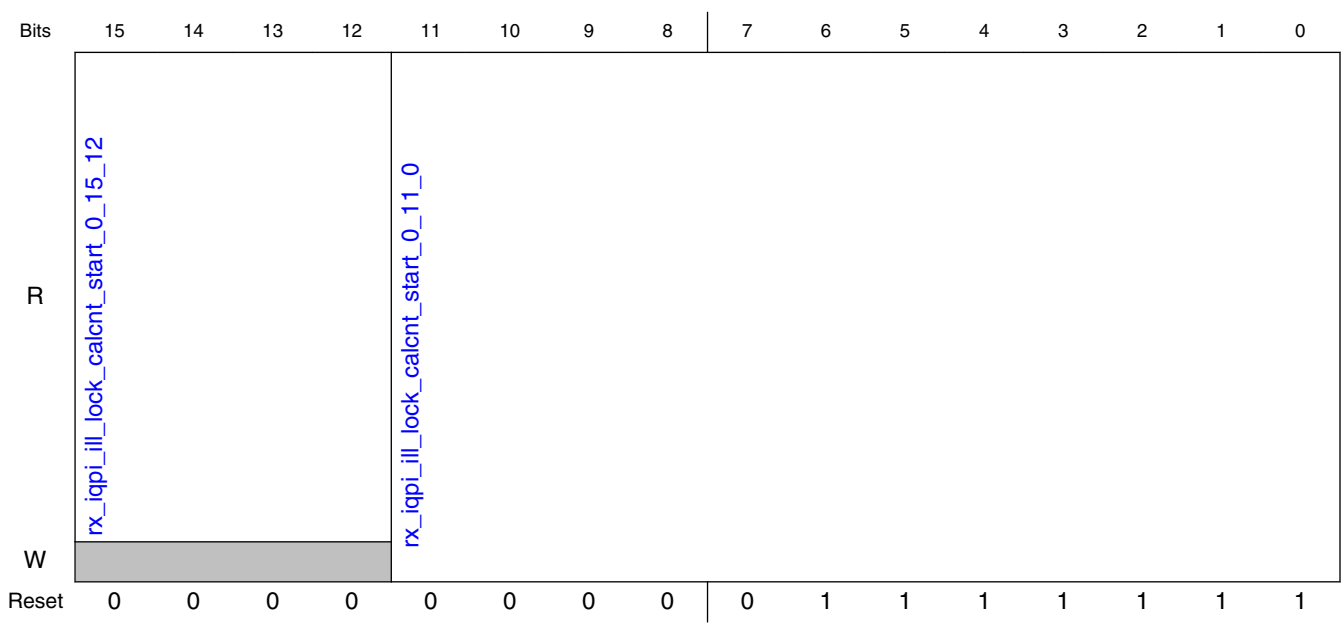
Field	Function
rx_iqpi_ill_lock_reftmr_start_15_12	
11-0 rx_iqpi_ill_lock_reftmr_start_11_0	ILL lock reference timer start value : This is the value that is loaded into the ILL lock reference timer as the starting point for that timer, when checking for ILL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. Note that the counter counts down to 0, therefore this is a value of 1 less than value required for the timer function.

13.4.10.2.110 RX IQ PI ILL lock calibration counter start value standard mode 0 register (lane0_rx_iqpi_ill_lock_calcnt_start_0 - lane 3_rx_iqpi_ill_lock_calcnt_start_0)

13.4.10.2.110.1 Offset

Register	Offset
lane0_rx_iqpi_ill_lock_calcnt_start_0	8028h
lane1_rx_iqpi_ill_lock_calcnt_start_0	8428h
lane2_rx_iqpi_ill_lock_calcnt_start_0	8828h
lane3_rx_iqpi_ill_lock_calcnt_start_0	8C28h

13.4.10.2.110.2 Diagram



13.4.10.2.110.3 Fields

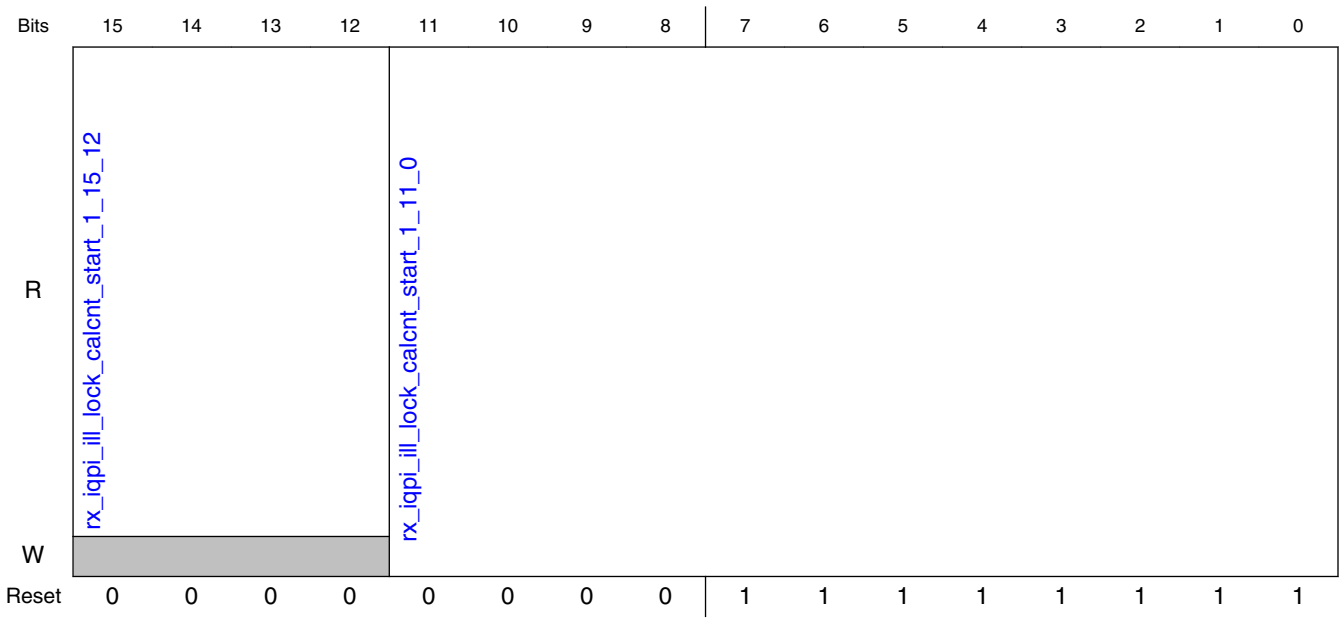
Field	Function
15-12 rx_iqpi_ill_lock_calcnt_start_0_15_12	Reserved
11-0 rx_iqpi_ill_lock_calcnt_start_0_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.111 RX IQ PI ILL lock calibration counter start value standard mode 1 register (lane0_rx_iqpi_ill_lock_calcnt_start_1 - lane 3_rx_iqpi_ill_lock_calcnt_start_1)

13.4.10.2.111.1 Offset

Register	Offset
lane0_rx_iqpi_ill_lock_calcnt_start_1	8029h
lane1_rx_iqpi_ill_lock_calcnt_start_1	8429h
lane2_rx_iqpi_ill_lock_calcnt_start_1	8829h
lane3_rx_iqpi_ill_lock_calcnt_start_1	8C29h

13.4.10.2.111.2 Diagram



13.4.10.2.111.3 Fields

Field	Function
15-12 rx_iqpi_ill_lock_calcnt_start_1_15_12	Reserved
11-0 rx_iqpi_ill_lock_calcnt_start_1_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.112 RX IQ PI ILL lock calibration counter start value standard mode 2 register (lane0_rx_iqpi_ill_lock_calcnt_start_2 - lane 3_rx_iqpi_ill_lock_calcnt_start_2)

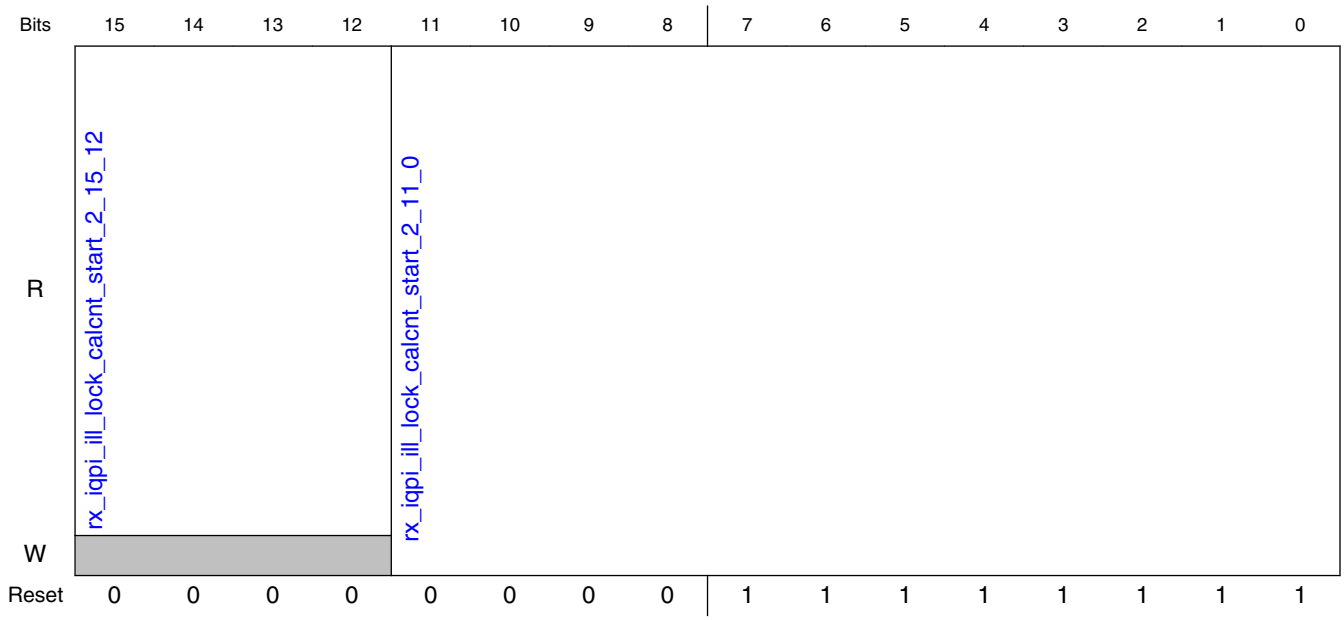
13.4.10.2.112.1 Offset

Register	Offset
lane0_rx_iqpi_ill_lock_calcnt_start_2	802Ah
lane1_rx_iqpi_ill_lock_calcnt_start_2	842Ah

Table continues on the next page...

Register	Offset
lane2_rx_iqpi_ill_lock_calcnt_start_2	882Ah
lane3_rx_iqpi_ill_lock_calcnt_start_2	8C2Ah

13.4.10.2.112.2 Diagram



13.4.10.2.112.3 Fields

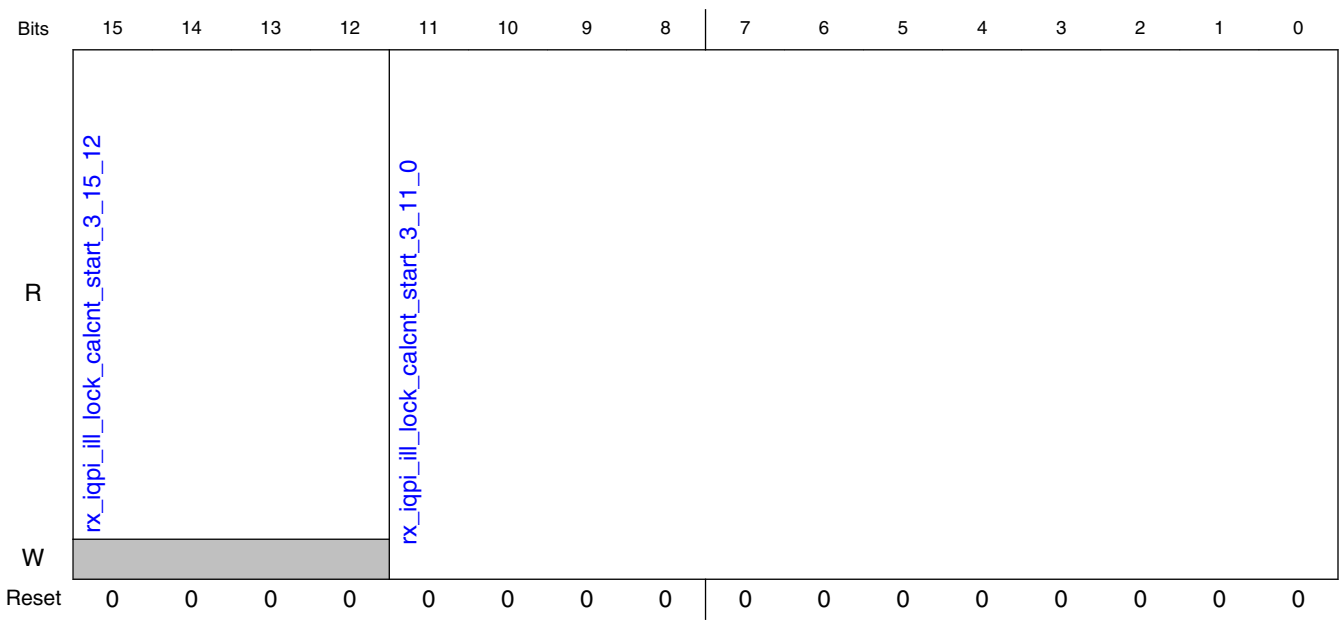
Field	Function
15-12 rx_iqpi_ill_lock_calcnt_start_2_15_12	Reserved
11-0 rx_iqpi_ill_lock_calcnt_start_2_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.113 RX IQ PI ILL lock calibration counter start value standard mode 3 register (lane0_rx_iqpi_ill_lock_calcnt_start_3 - lane 3_rx_iqpi_ill_lock_calcnt_start_3)

13.4.10.2.113.1 Offset

Register	Offset
lane0_rx_iqpi_ill_lock_calcnt_start_3	802Bh
lane1_rx_iqpi_ill_lock_calcnt_start_3	842Bh
lane2_rx_iqpi_ill_lock_calcnt_start_3	882Bh
lane3_rx_iqpi_ill_lock_calcnt_start_3	8C2Bh

13.4.10.2.113.2 Diagram



13.4.10.2.113.3 Fields

Field	Function
15-12 rx_iqpi_ill_lock_calcnt_start_3_15_12	Reserved
11-0 rx_iqpi_ill_lock_calcnt_start_3_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.114 RX E PI ILL calibration control register (lane0_rx_epi_ill_cal_ctrl - lane3_rx_epi_ill_cal_ctrl)

13.4.10.2.114.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_ctrl	8030h
lane1_rx_epi_ill_cal_ctrl	8430h
lane2_rx_epi_ill_cal_ctrl	8830h
lane3_rx_epi_ill_cal_ctrl	8C30h

13.4.10.2.114.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_epi_ill_cal_ctrl_15	rx_epi_ill_cal_ctrl_14	rx_epi_ill_cal_ctrl_13_8						rx_epi_ill_cal_ctrl_7_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0

13.4.10.2.114.3 Fields

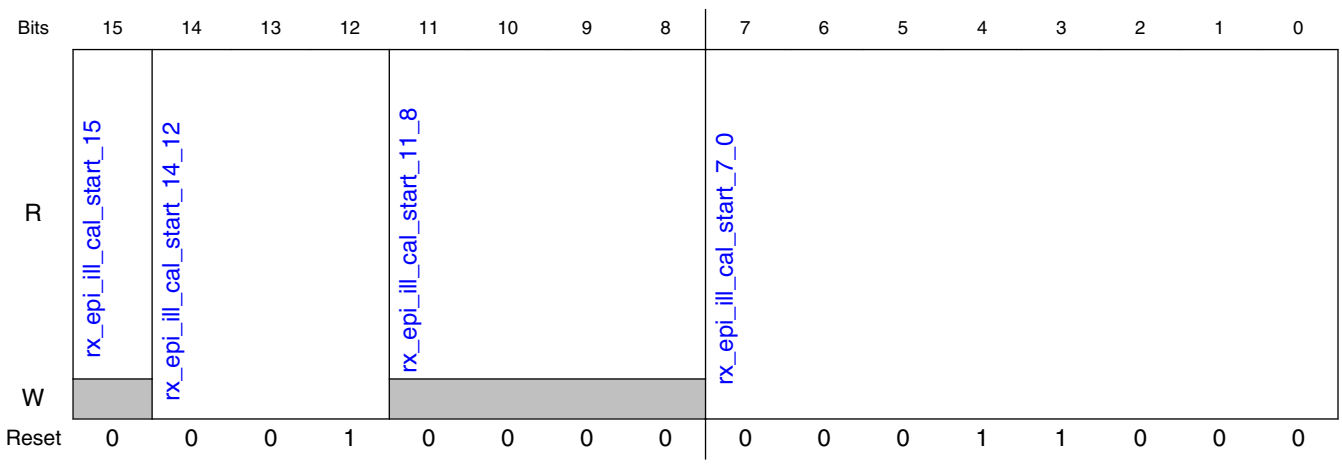
Field	Function
15 rx_epi_ill_cal_ctrl_15	Start ILL calibration: Activating (1b1) this bit will start an ILL calibration process. This bit must remain active until the ILL calibration process is complete (as indicated by the ILL calibration process done bit in this register). To start another ILL calibration process, the ILL calibration process done bit must have gone inactive from any prior calibration process.
14 rx_epi_ill_cal_ctrl_14	ILL calibration process done: This bit will be set to 1b1 when the ILL calibration process is complete. It will be cleared by the receiver being reset, or by the deactivation of the Start ILL calibration bit in this register.
13-8 rx_epi_ill_cal_ctrl_13_8	Reserved
7-0 rx_epi_ill_cal_ctrl_7_0	ILL calibration code: This is the calibration code that was determined by the ILL calibration process. This signal is valid when the ILL calibration process is complete. This field specifies the number of resistors that are switched in. The following are the values for this code:

13.4.10.2.115 RX E PI ILL calibration start point register (lane0_rx_epi_ill_cal_start - lane3_rx_epi_ill_cal_start)

13.4.10.2.115.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_start	8031h
lane1_rx_epi_ill_cal_start	8431h
lane2_rx_epi_ill_cal_start	8831h
lane3_rx_epi_ill_cal_start	8C31h

13.4.10.2.115.2 Diagram



13.4.10.2.115.3 Fields

Field	Function
15 rx_epi_ill_cal_start_15	Reserved
14-12 rx_epi_ill_cal_start_14_12	ILL calibration initial step size control: This field specifies the initial step size for the ILL calibration state machine. The following are the values that can be used in this field, and the corresponding step sizes.
11-8 rx_epi_ill_cal_start_11_8	Reserved

Table continues on the next page...

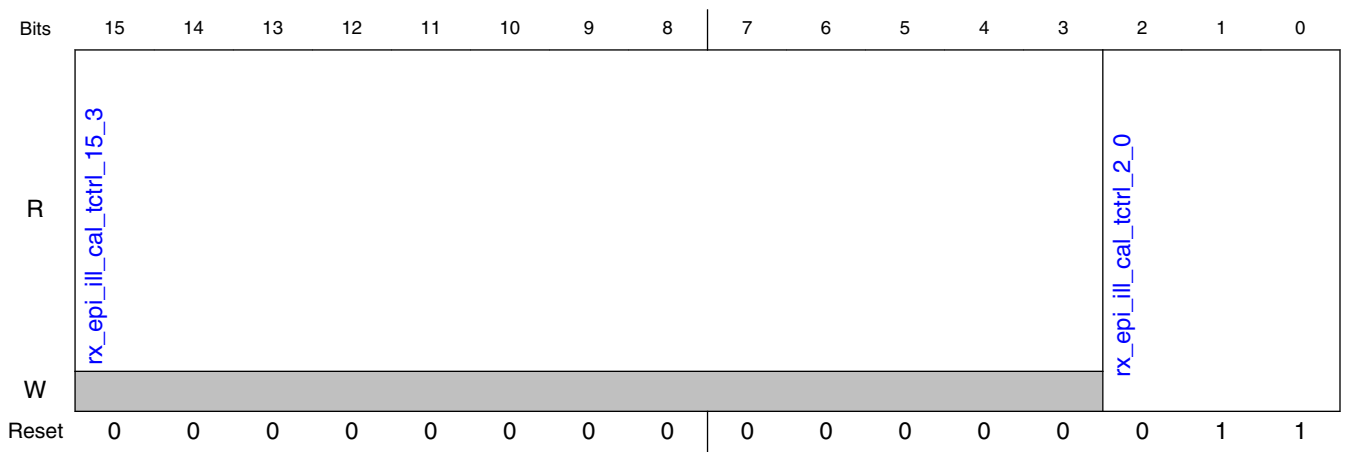
Field	Function
7-0 rx_epi_ill_cal_start_7_0	ILL calibration code starting point value: This field specifies the starting ILL code that is used by the ILL calibration state machine. The purpose of this value is such that the ILL calibration process starts at a point that is, on average, relatively close to the final calibration point. This allows the calibration time to be reduced, on average. The following are the values for this code:

13.4.10.2.116 RX E PI ILL calibration timer control register (lane0_rx_epi_ill_cal_tctrl - lane3_rx_epi_ill_cal_tctrl)

13.4.10.2.116.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_tctrl	8032h
lane1_rx_epi_ill_cal_tctrl	8432h
lane2_rx_epi_ill_cal_tctrl	8832h
lane3_rx_epi_ill_cal_tctrl	8C32h

13.4.10.2.116.2 Diagram



13.4.10.2.116.3 Fields

Field	Function
15-3 rx_epi_ill_cal_tctrl_15_3	Reserved
2-0	ILL calibration initial time scale control: This field specifies the calibration start time scaling factor applied to the ILL calibration when running the initial step size for the calibration code is not set to 1. Setting this

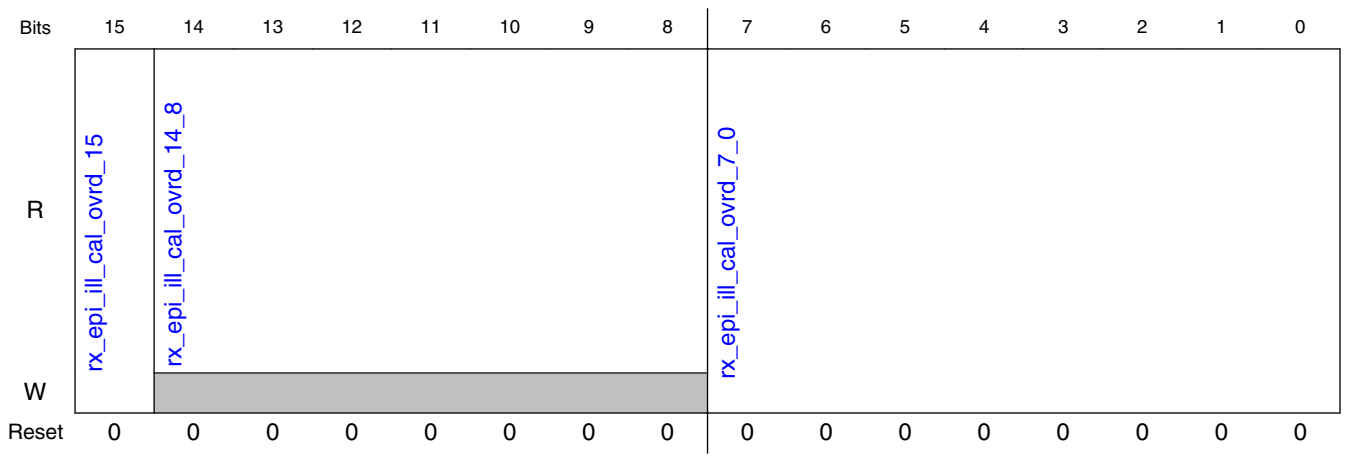
Field	Function
rx_epi_ill_cal_tctrl_2_0	value to a value other than 1 will reduce the calibration measurement time by the amount specified below. Then when the final calibration steps are made the full calibration time will be used to get the appropriate amount of resolution.

13.4.10.2.117 RX E PI ILL calibration override register (lane0_rx_epi_ill_cal_ovrd - lane3_rx_epi_ill_cal_ovrd)

13.4.10.2.117.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_ovrd	8033h
lane1_rx_epi_ill_cal_ovrd	8433h
lane2_rx_epi_ill_cal_ovrd	8833h
lane3_rx_epi_ill_cal_ovrd	8C33h

13.4.10.2.117.2 Diagram



13.4.10.2.117.3 Fields

Field	Function
15 rx_epi_ill_cal_ovrd_15	ILL calibration code override enable: Activating (1b1) this bit allows the ILL code determined during the automatic ILL calibration process to be overridden by the value driven by the ILL calibration code override value field in this register.
14-8	Reserved

Table continues on the next page...

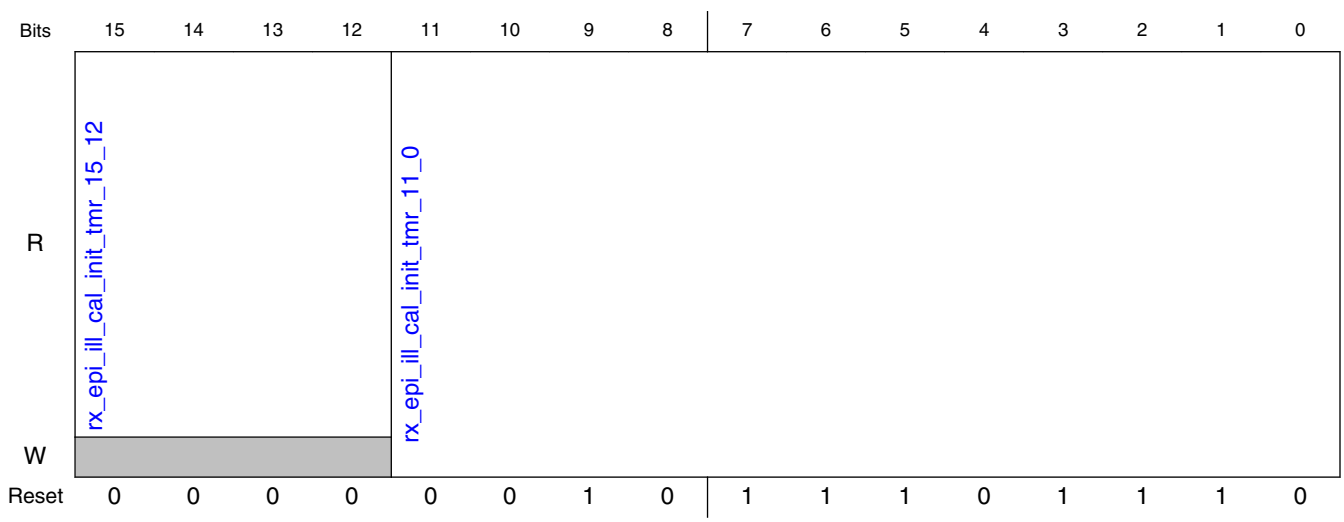
Field	Function
rx_epi_ill_cal_ovrd_14_8	
7-0 rx_epi_ill_cal_ovrd_7_0	ILL calibration code override value: This field is used to override the ILL code determined during the automatic ILL calibration process. The ILL code driven on this field is valid when the ILL calibration code override enable bit in this register is active. The following are the values for this code:

13.4.10.2.118 RX E PI ILL calibration initialization timer register (lane0_rx_epi_ill_cal_init_tmr - lane3_rx_epi_ill_cal_init_tmr)

13.4.10.2.118.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_init_tmr	8034h
lane1_rx_epi_ill_cal_init_tmr	8434h
lane2_rx_epi_ill_cal_init_tmr	8834h
lane3_rx_epi_ill_cal_init_tmr	8C34h

13.4.10.2.118.2 Diagram



13.4.10.2.118.3 Fields

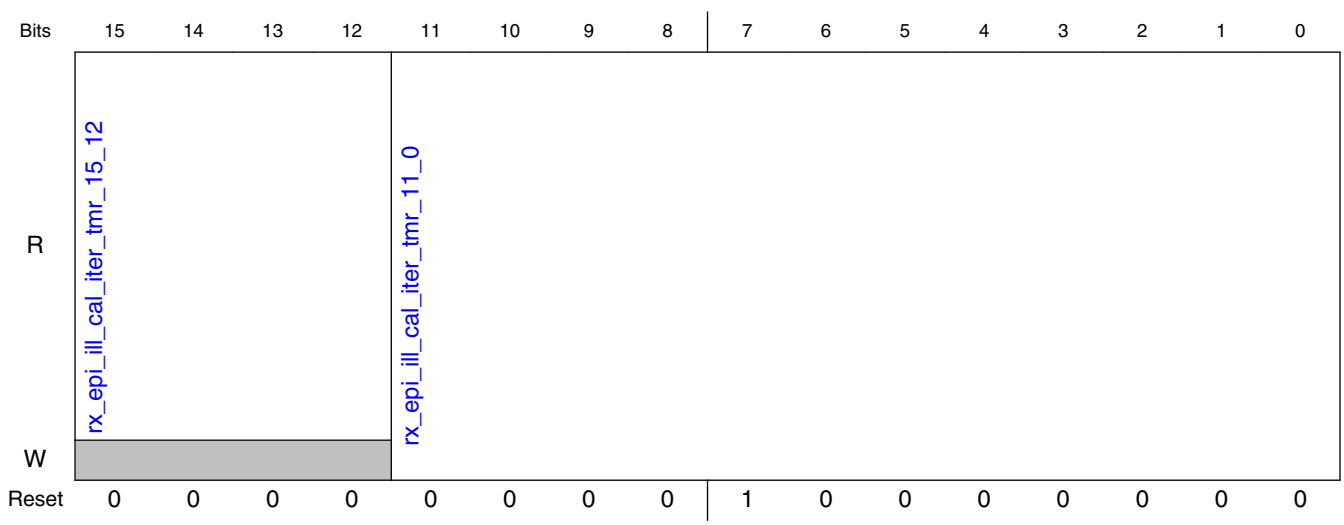
Field	Function
15-12 rx_epi_ill_cal_init_tmr_15_12	Reserved
11-0 rx_epi_ill_cal_init_tmr_11_0	Initialization wait timer value: This is the number of

13.4.10.2.119 RX E PI ILL calibration iteration timer register (lane0_rx_epi_ill_cal_iter_tmr - lane3_rx_epi_ill_cal_iter_tmr)

13.4.10.2.119.1 Offset

Register	Offset
lane0_rx_epi_ill_cal_iter_tmr	8035h
lane1_rx_epi_ill_cal_iter_tmr	8435h
lane2_rx_epi_ill_cal_iter_tmr	8835h
lane3_rx_epi_ill_cal_iter_tmr	8C35h

13.4.10.2.119.2 Diagram



13.4.10.2.119.3 Fields

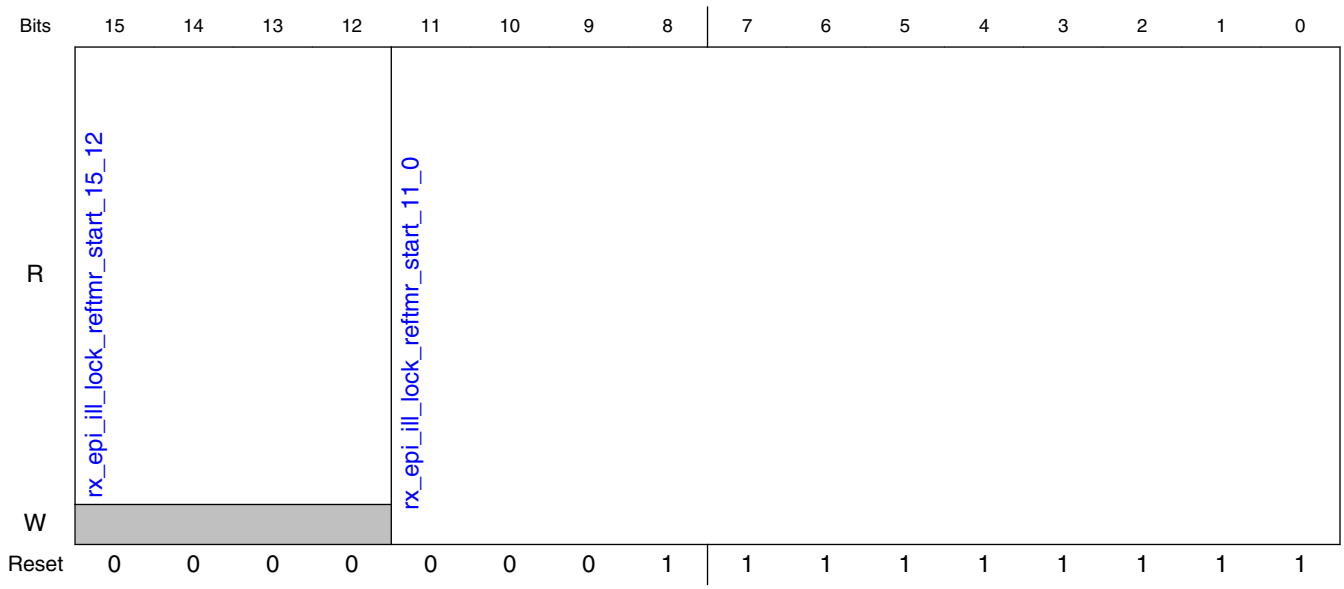
Field	Function
15-12 rx_epi_ill_cal_iter_tmr_15_12	Reserved
11-0 rx_epi_ill_cal_iter_tmr_11_0	Iteration wait timer value: This is the number of

13.4.10.2.120 RX E PI ILL lock reference timer start value register (lane0_rx_epi_ill_lock_reftmr_start - lane3_rx_epi_ill_lock_reftmr_start)

13.4.10.2.120.1 Offset

Register	Offset
lane0_rx_epi_ill_lock_reftmr_start	8036h
lane1_rx_epi_ill_lock_reftmr_start	8436h
lane2_rx_epi_ill_lock_reftmr_start	8836h
lane3_rx_epi_ill_lock_reftmr_start	8C36h

13.4.10.2.120.2 Diagram



13.4.10.2.120.3 Fields

Field	Function
15-12 rx_epi_ill_lock_reftmr_start_15_12	Reserved
11-0 rx_epi_ill_lock_reftmr_start_11_0	ILL lock reference timer start value : This is the value that is loaded into the ILL lock timer as the starting point for that timer, when checking for ILL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. Note that the counter counts down to 0, therefore this is a value of 1 less than value required for the timer function.

13.4.10.2.121 RX E PI ILL lock calibration counter start value standard mode 0 register (lane0_rx_epi_ill_lock_calcnt_start_0 - lane 3_rx_epi_ill_lock_calcnt_start_0)

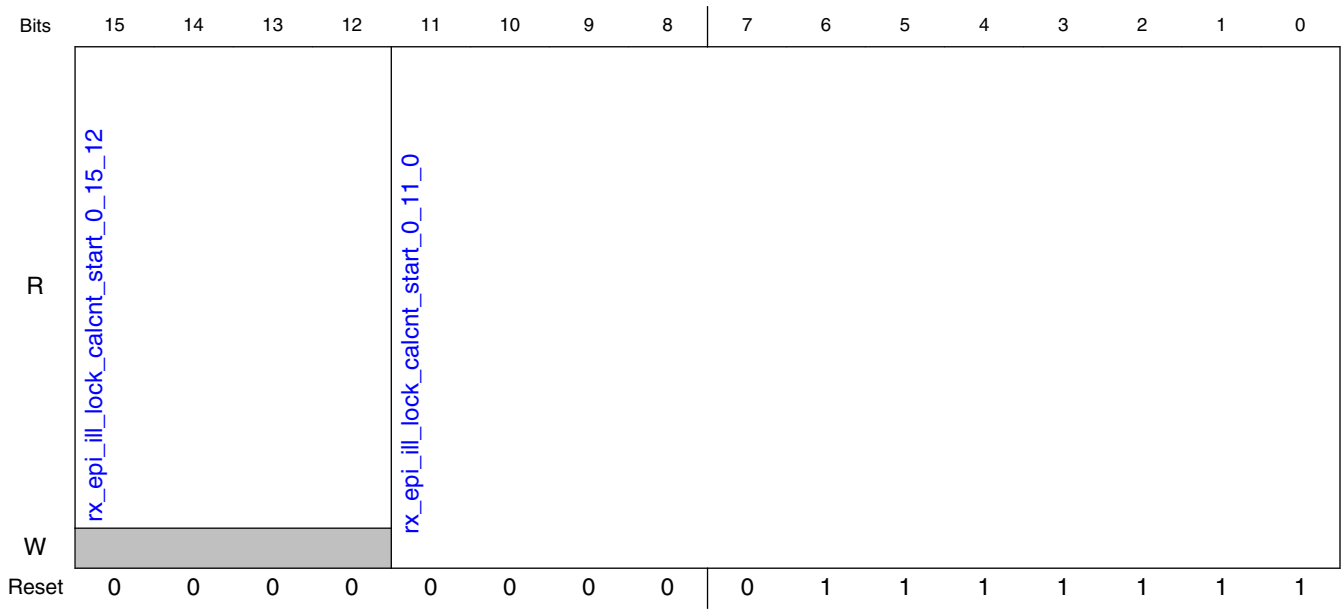
13.4.10.2.121.1 Offset

Register	Offset
lane0_rx_epi_ill_lock_calcnt_start_0	8038h
lane1_rx_epi_ill_lock_calcnt_start_0	8438h

Table continues on the next page...

Register	Offset
lane2_rx_epi_ill_lock_calcnt_start_0	8838h
lane3_rx_epi_ill_lock_calcnt_start_0	8C38h

13.4.10.2.121.2 Diagram



13.4.10.2.121.3 Fields

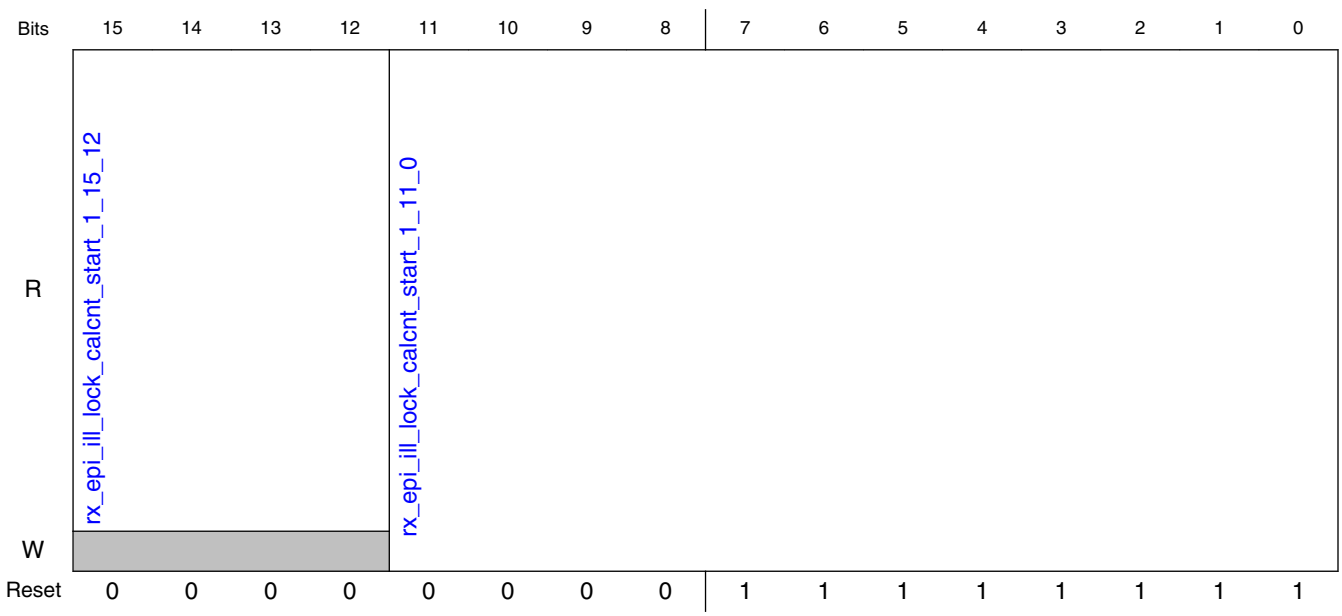
Field	Function
15-12 rx_epi_ill_lock_calcnt_start_0_15_12	Reserved
11-0 rx_epi_ill_lock_calcnt_start_0_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.122 RX E PI ILL lock calibration counter start value standard mode 1 register (lane0_rx_epi_ill_lock_calcnt_start_1 - lane 3_rx_epi_ill_lock_calcnt_start_1)

13.4.10.2.122.1 Offset

Register	Offset
lane0_rx_epi_ill_lock_calcnt_start_1	8039h
lane1_rx_epi_ill_lock_calcnt_start_1	8439h
lane2_rx_epi_ill_lock_calcnt_start_1	8839h
lane3_rx_epi_ill_lock_calcnt_start_1	8C39h

13.4.10.2.122.2 Diagram



13.4.10.2.122.3 Fields

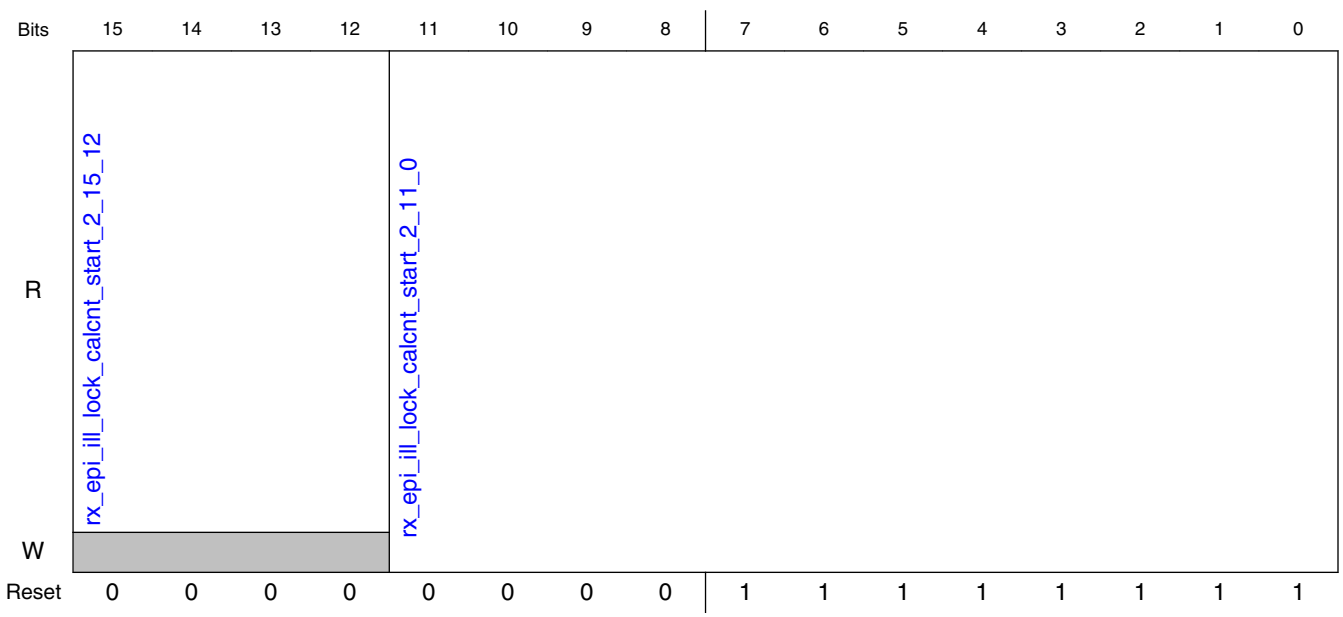
Field	Function
15-12 rx_epi_ill_lock_calcnt_start_1_15_12	Reserved
11-0 rx_epi_ill_lock_calcnt_start_1_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking for ILL lock when

13.4.10.2.123 RX E PI ILL lock calibration counter start value standard mode 2 register (lane0_rx_epi_ill_lock_calcnt_start_2 - lane 3_rx_epi_ill_lock_calcnt_start_2)

13.4.10.2.123.1 Offset

Register	Offset
lane0_rx_epi_ill_lock_calcnt_start_2	803Ah
lane1_rx_epi_ill_lock_calcnt_start_2	843Ah
lane2_rx_epi_ill_lock_calcnt_start_2	883Ah
lane3_rx_epi_ill_lock_calcnt_start_2	8C3Ah

13.4.10.2.123.2 Diagram



13.4.10.2.123.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

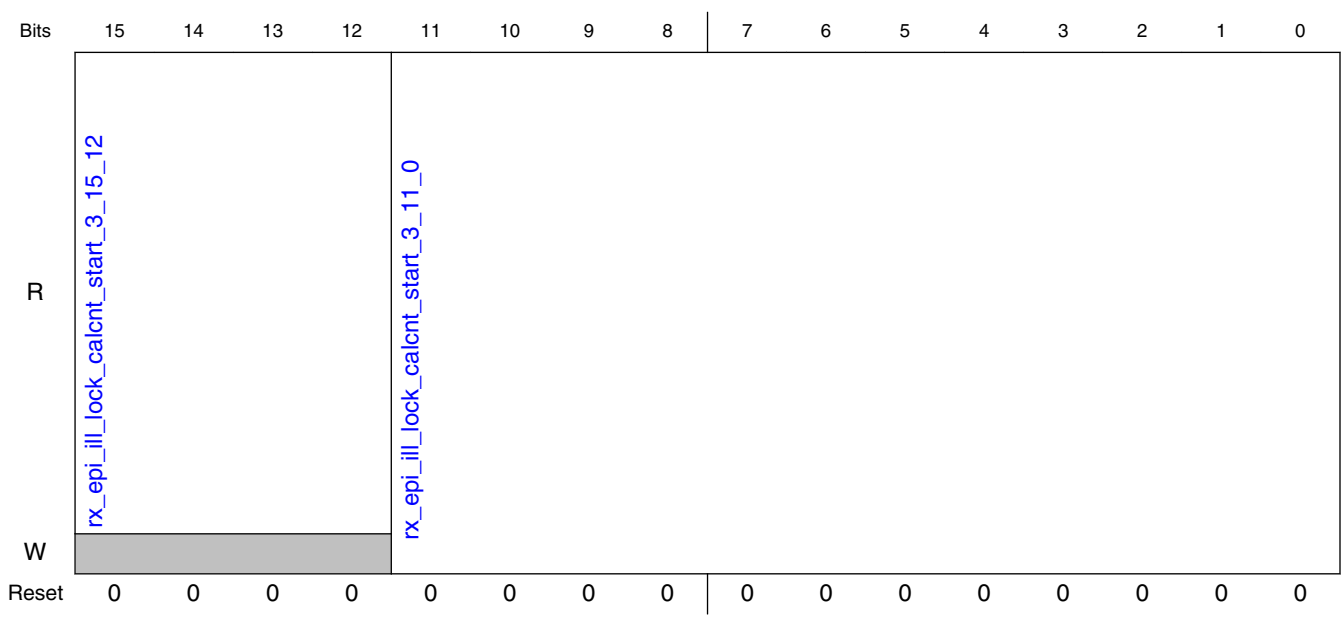
Field	Function
rx_epi_ill_lock_calcnt_start_2_15_12	
11-0 rx_epi_ill_lock_calcnt_start_2_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking when

13.4.10.2.124 RX E PI ILL lock calibration counter start value standard mode 3 register (lane0_rx_epi_ill_lock_calcnt_start_3 - lane 3_rx_epi_ill_lock_calcnt_start_3)

13.4.10.2.124.1 Offset

Register	Offset
lane0_rx_epi_ill_lock_calcnt_start_3	803Bh
lane1_rx_epi_ill_lock_calcnt_start_3	843Bh
lane2_rx_epi_ill_lock_calcnt_start_3	883Bh
lane3_rx_epi_ill_lock_calcnt_start_3	8C3Bh

13.4.10.2.124.2 Diagram



13.4.10.2.124.3 Fields

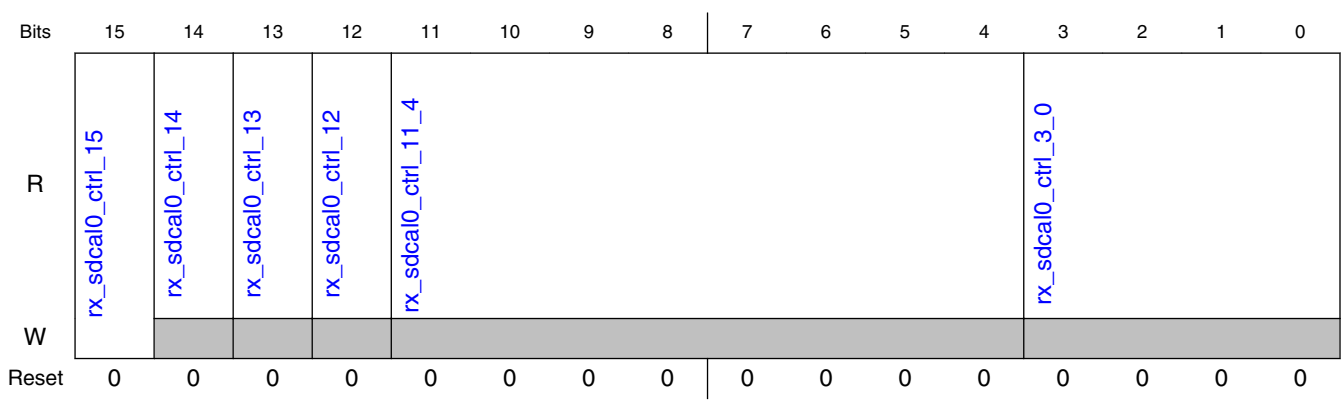
Field	Function
15-12 rx_epi_ill_lock_calcnt_start_3_15_12	Reserved
11-0 rx_epi_ill_lock_calcnt_start_3_11_0	ILL lock calibration counter start value : This is the value that is loaded into the ILL lock calibration counter as the starting point for that counter, when checking when

13.4.10.2.125 Signal detect calibration 0 control register (lane0_rx_sdcal0_ctrl - lane3_rx_sdcal0_ctrl)

13.4.10.2.125.1 Offset

Register	Offset
lane0_rx_sdcal0_ctrl	8040h
lane1_rx_sdcal0_ctrl	8440h
lane2_rx_sdcal0_ctrl	8840h
lane3_rx_sdcal0_ctrl	8C40h

13.4.10.2.125.2 Diagram



13.4.10.2.125.3 Fields

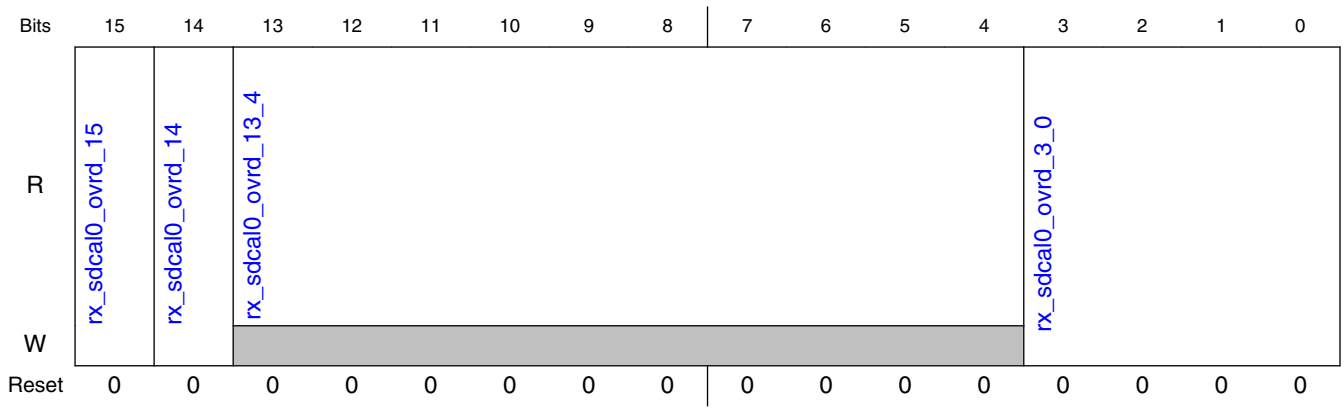
Field	Function
15 rx_sdcal0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the bandbap calibration process done bit in this register is cleared.
14 rx_sdcal0_ctrl_14	Calibration process done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_sdcal0_ctrl_13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.
12 rx_sdcal0_ctrl_12	Current analog comparator response: This is the current state of the analog comparator response signal (
11-4 rx_sdcal0_ctrl_11_4	Reserved
3-0 rx_sdcal0_ctrl_3_0	Calibration code: This is the calibration code that was determined by the calibration process. The following is the encoding of this field, and how it maps to the

13.4.10.2.126 Signal detect calibration 0 override register (lane0_rx_sdcal0_ovrd - lane3_rx_sdcal0_ovrd)

13.4.10.2.126.1 Offset

Register	Offset
lane0_rx_sdcal0_ovrd	8041h
lane1_rx_sdcal0_ovrd	8441h
lane2_rx_sdcal0_ovrd	8841h
lane3_rx_sdcal0_ovrd	8C41h

13.4.10.2.126.2 Diagram



13.4.10.2.126.3 Fields

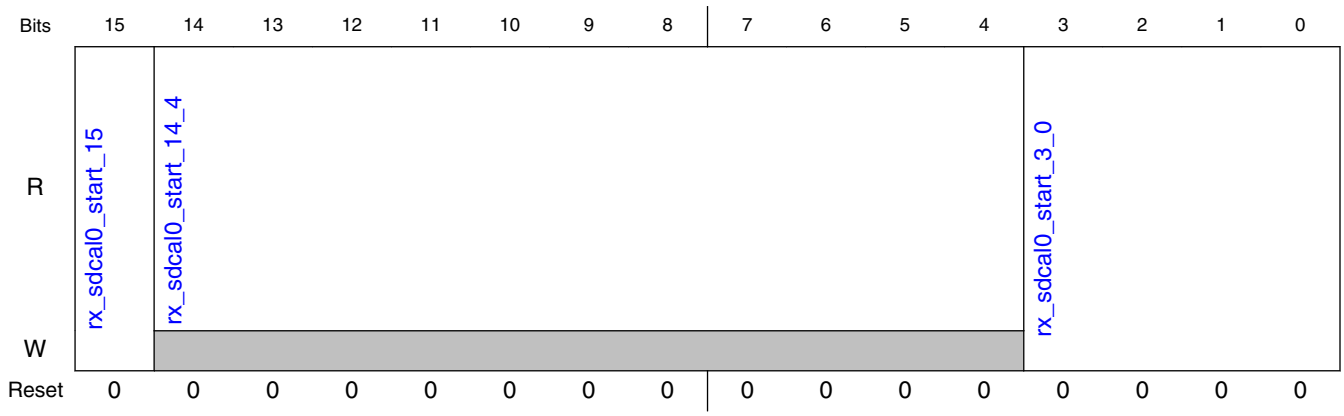
Field	Function
15 rx_sdcal0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the codes determined during the automatic calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_sdcal0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-4 rx_sdcal0_ovrd_13_4	Reserved
3-0 rx_sdcal0_ovrd_3_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic calibration process. The code written to these bits is valid when the calibration code override enable bit in this register is active.

13.4.10.2.127 Signal detect calibration 0 start register (lane0_rx_sdcal0_start - lane3_rx_sdcal0_start)

13.4.10.2.127.1 Offset

Register	Offset
lane0_rx_sdcal0_start	8042h
lane1_rx_sdcal0_start	8442h
lane2_rx_sdcal0_start	8842h
lane3_rx_sdcal0_start	8C42h

13.4.10.2.127.2 Diagram



13.4.10.2.127.3 Fields

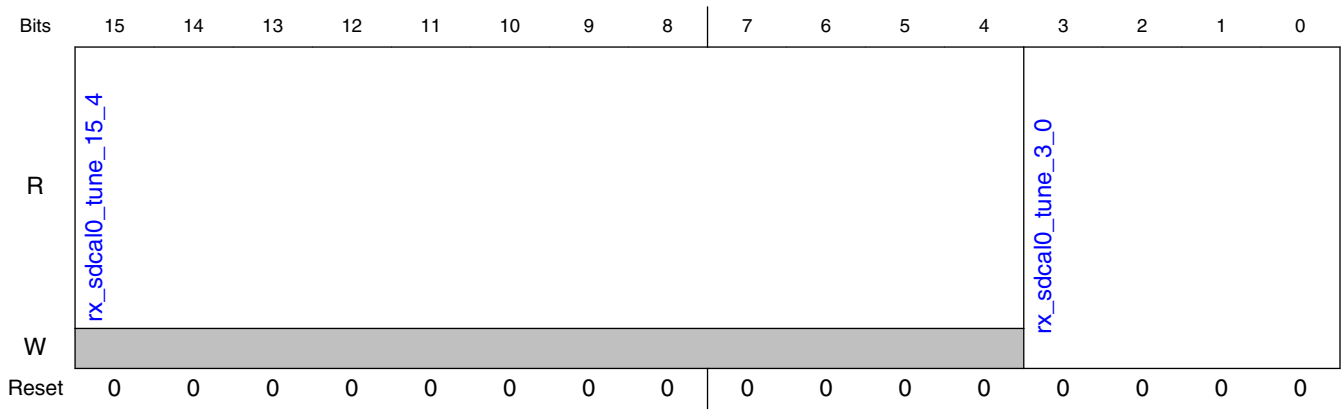
Field	Function
15 rx_sdcal0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-4 rx_sdcal0_start_14_4	Reserved
3-0 rx_sdcal0_start_3_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run.

13.4.10.2.128 Signal detect calibration 0 tune register (lane0_rx_sdcal0_tune - lane3_rx_sdcal0_tune)

13.4.10.2.128.1 Offset

Register	Offset
lane0_rx_sdcal0_tune	8043h
lane1_rx_sdcal0_tune	8443h
lane2_rx_sdcal0_tune	8843h
lane3_rx_sdcal0_tune	8C43h

13.4.10.2.128.2 Diagram



13.4.10.2.128.3 Fields

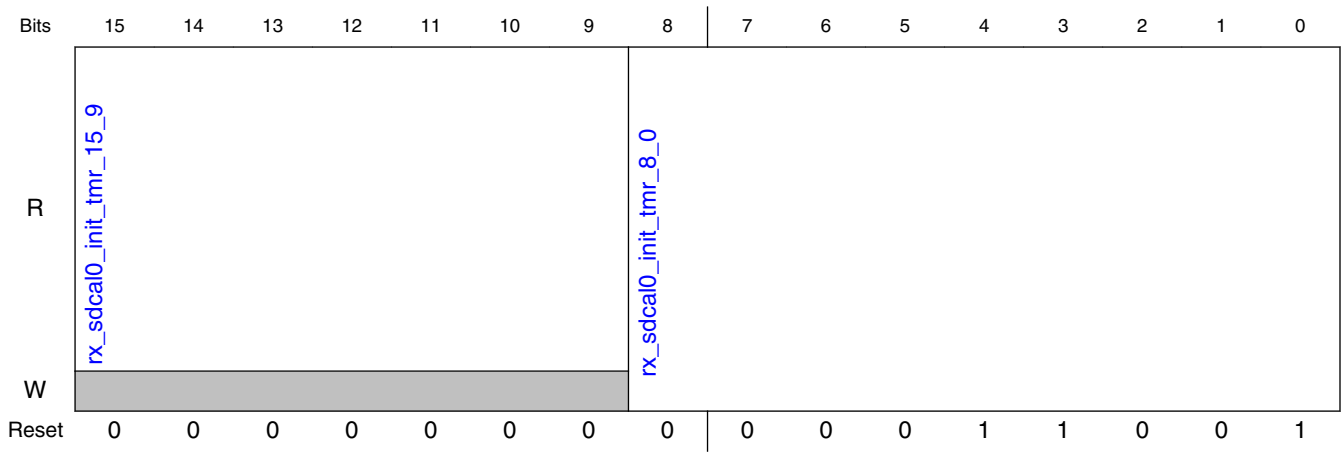
Field	Function
15-4 rx_sdcal0_tune_15_4	Reserved
3-0 rx_sdcal0_tune_3_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.129 Signal detect calibration 0 initialization timer register (lane0_rx_sdcal0_init_tmr - lane3_rx_sdcal0_init_tmr)

13.4.10.2.129.1 Offset

Register	Offset
lane0_rx_sdcal0_init_tmr	8044h
lane1_rx_sdcal0_init_tmr	8444h
lane2_rx_sdcal0_init_tmr	8844h
lane3_rx_sdcal0_init_tmr	8C44h

13.4.10.2.129.2 Diagram



13.4.10.2.129.3 Fields

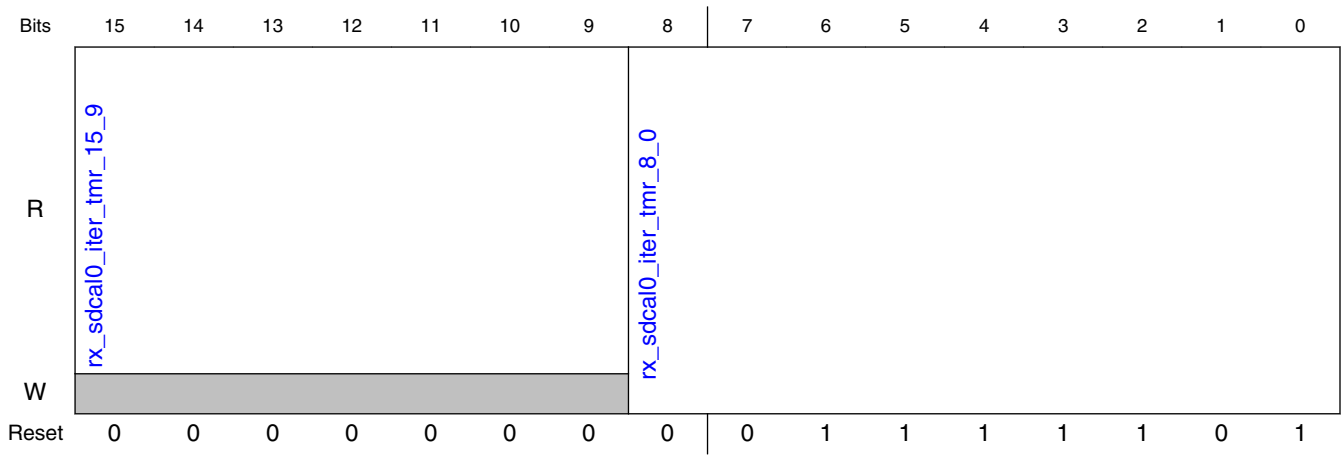
Field	Function
15-9 rx_sdcal0_init_tmr_15_9	Reserved
8-0 rx_sdcal0_init_tmr_8_0	Initialization wait timer value: This is the number of

13.4.10.2.130 Signal detect calibration 0 iteration timer register (lane0_rx_sdcal0_iter_tmr - lane3_rx_sdcal0_iter_tmr)

13.4.10.2.130.1 Offset

Register	Offset
lane0_rx_sdcal0_iter_tmr	8045h
lane1_rx_sdcal0_iter_tmr	8445h
lane2_rx_sdcal0_iter_tmr	8845h
lane3_rx_sdcal0_iter_tmr	8C45h

13.4.10.2.130.2 Diagram



13.4.10.2.130.3 Fields

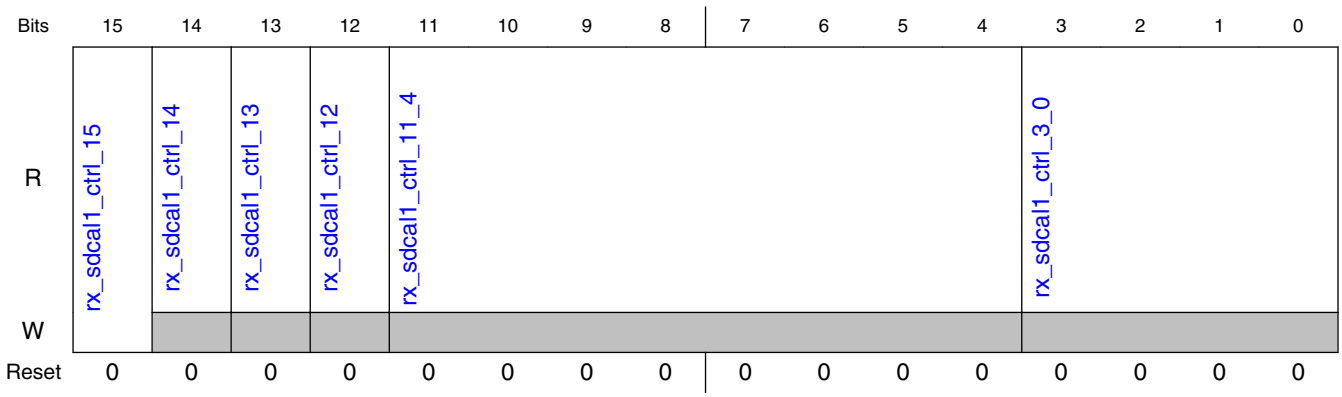
Field	Function
15-9 rx_sdcal0_iter_tmr_15_9	Reserved
8-0 rx_sdcal0_iter_tmr_8_0	Iteration wait timer value: This is the number of

13.4.10.2.131 Signal detect calibration 1 control register (lane0_rx_sdcal1_ctrl - lane3_rx_sdcal1_ctrl)

13.4.10.2.131.1 Offset

Register	Offset
lane0_rx_sdcal1_ctrl	8048h
lane1_rx_sdcal1_ctrl	8448h
lane2_rx_sdcal1_ctrl	8848h
lane3_rx_sdcal1_ctrl	8C48h

13.4.10.2.131.2 Diagram



13.4.10.2.131.3 Fields

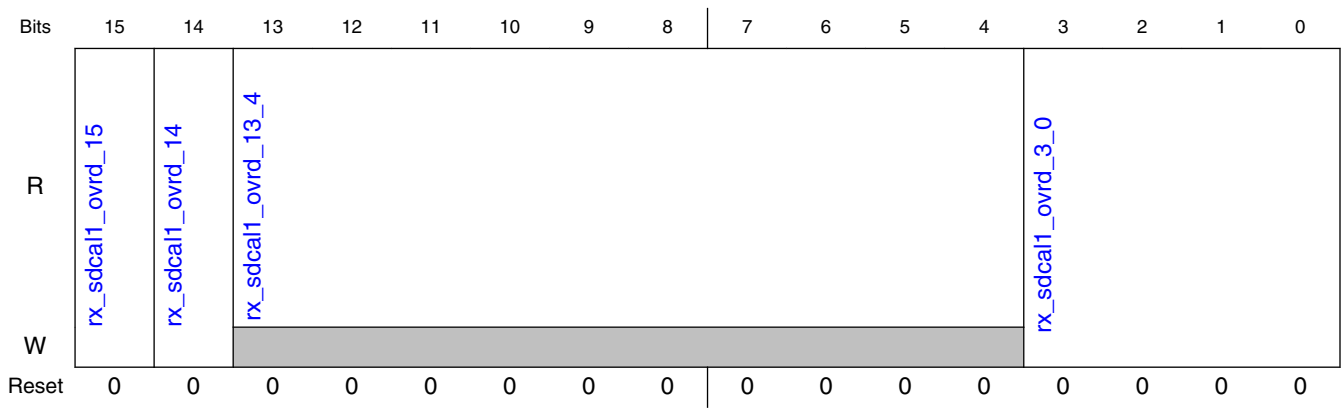
Field	Function
15 rx_sdcal1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the bandbap calibration process done bit in this register is cleared.
14 rx_sdcal1_ctrl_14	Calibration process done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_sdcal1_ctrl_13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.
12 rx_sdcal1_ctrl_12	Current analog comparator response: This is the current state of the analog comparator response signal (
11-4 rx_sdcal1_ctrl_11_4	Reserved
3-0 rx_sdcal1_ctrl_3_0	Calibration code: This is the calibration code that was determined by the calibration process. The following is the encoding of this field, and how it maps to the

13.4.10.2.132 Signal detect calibration 1 override register (lane0_rx_sdcal1_ovrd - lane3_rx_sdcal1_ovrd)

13.4.10.2.132.1 Offset

Register	Offset
lane0_rx_sdcal1_ovrd	8049h
lane1_rx_sdcal1_ovrd	8449h
lane2_rx_sdcal1_ovrd	8849h
lane3_rx_sdcal1_ovrd	8C49h

13.4.10.2.132.2 Diagram



13.4.10.2.132.3 Fields

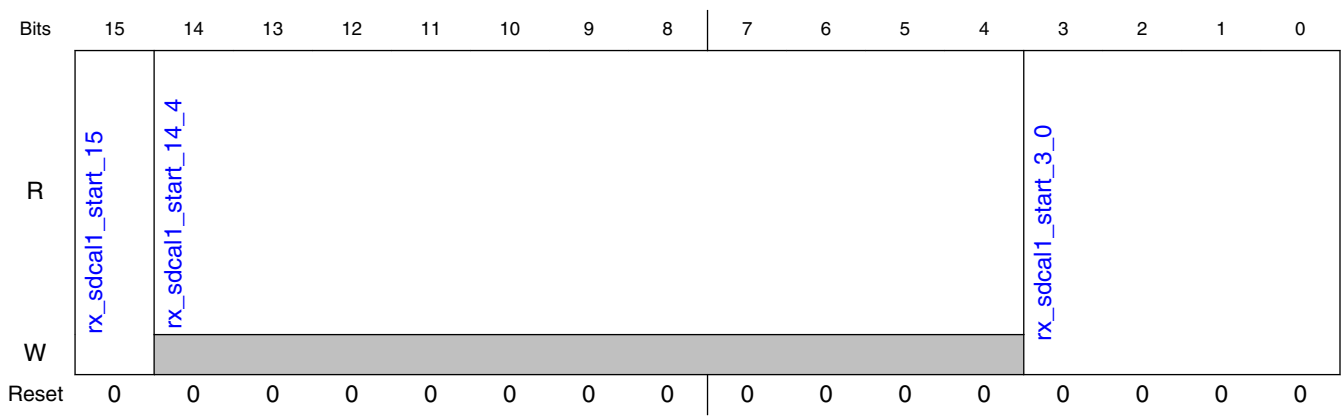
Field	Function
15 rx_sdcal1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_sdcal1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-4 rx_sdcal1_ovrd_13_4	Reserved
3-0 rx_sdcal1_ovrd_3_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic calibration process. The code written to these bits is valid when the calibration code override enable bit in this register is active.

13.4.10.2.133 Signal detect calibration 1 start register (lane0_rx_sdcal1_start - lane3_rx_sdcal1_start)

13.4.10.2.133.1 Offset

Register	Offset
lane0_rx_sdcal1_start	804Ah
lane1_rx_sdcal1_start	844Ah
lane2_rx_sdcal1_start	884Ah
lane3_rx_sdcal1_start	8C4Ah

13.4.10.2.133.2 Diagram



13.4.10.2.133.3 Fields

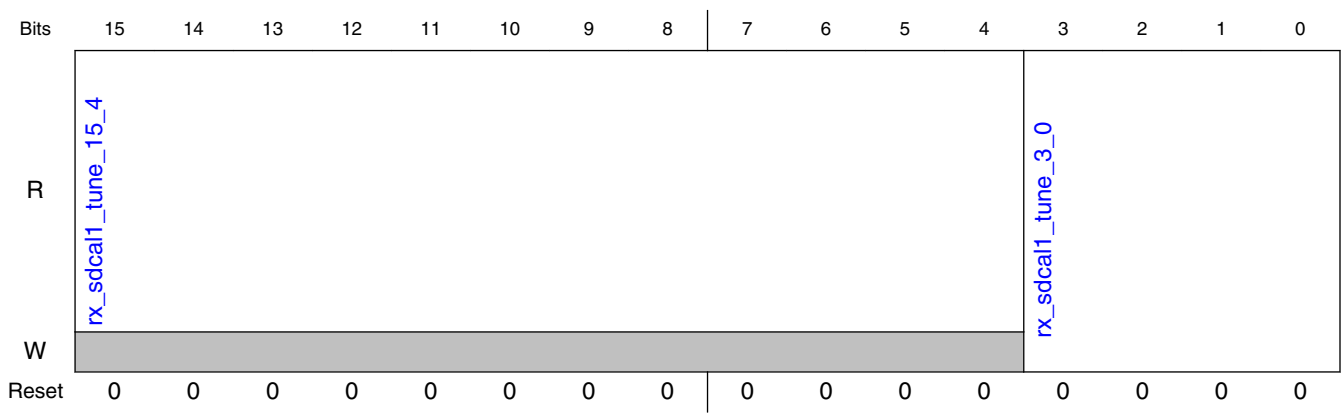
Field	Function
15 rx_sdcal1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-4 rx_sdcal1_start_14_4	Reserved
3-0 rx_sdcal1_start_3_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run.

13.4.10.2.134 Signal detect calibration 1 tune register (lane0_rx_sdcal1_tune - lane3_rx_sdcal1_tune)

13.4.10.2.134.1 Offset

Register	Offset
lane0_rx_sdcal1_tune	804Bh
lane1_rx_sdcal1_tune	844Bh
lane2_rx_sdcal1_tune	884Bh
lane3_rx_sdcal1_tune	8C4Bh

13.4.10.2.134.2 Diagram



13.4.10.2.134.3 Fields

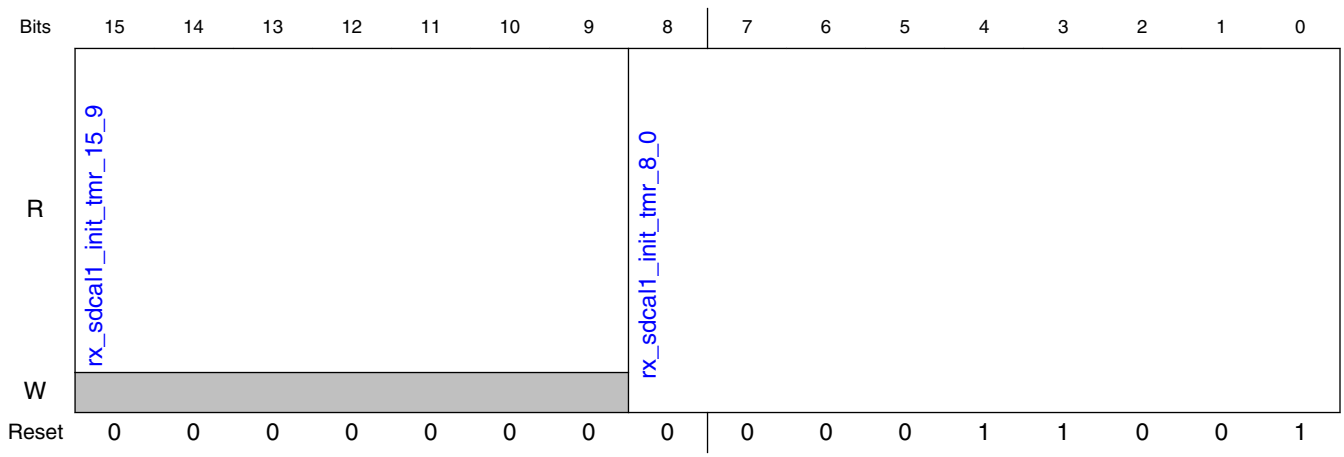
Field	Function
15-4 rx_sdcal1_tune_15_4	Reserved
3-0 rx_sdcal1_tune_3_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.135 Signal detect calibration 1 initialization timer register (lane0_rx_sdcal1_init_tmr - lane3_rx_sdcal1_init_tmr)

13.4.10.2.135.1 Offset

Register	Offset
lane0_rx_sdcal1_init_tmr	804Ch
lane1_rx_sdcal1_init_tmr	844Ch
lane2_rx_sdcal1_init_tmr	884Ch
lane3_rx_sdcal1_init_tmr	8C4Ch

13.4.10.2.135.2 Diagram



13.4.10.2.135.3 Fields

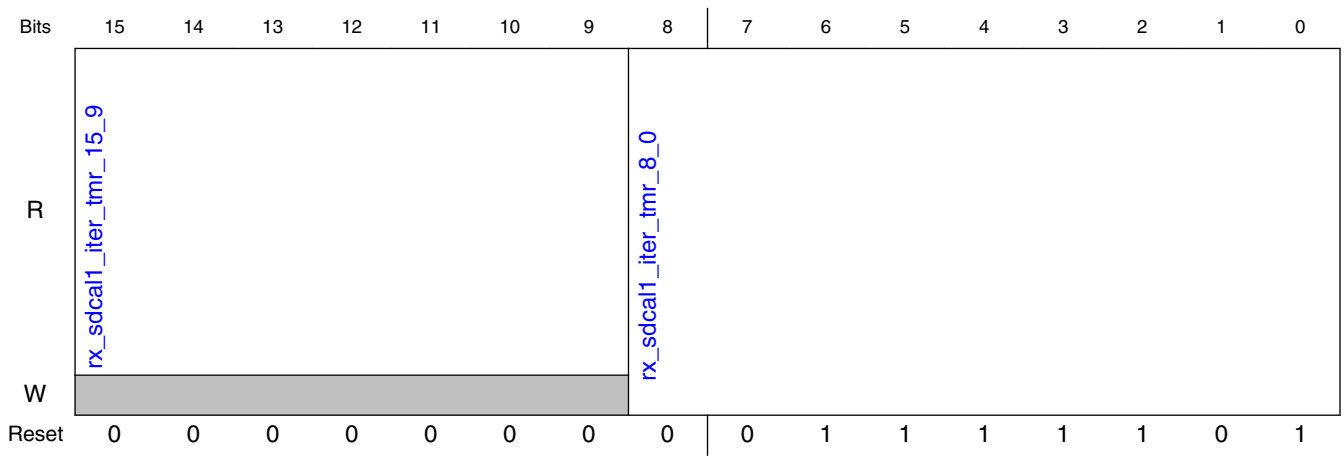
Field	Function
15-9 rx_sdcal1_init_tmr_15_9	Reserved
8-0 rx_sdcal1_init_tmr_8_0	Initialization wait timer value: This is the number of

13.4.10.2.136 Signal detect calibration 1 iteration timer register (lane0_rx_sdcal1_iter_tmr - lane3_rx_sdcal1_iter_tmr)

13.4.10.2.136.1 Offset

Register	Offset
lane0_rx_sdcal1_iter_tmr	804Dh
lane1_rx_sdcal1_iter_tmr	844Dh
lane2_rx_sdcal1_iter_tmr	884Dh
lane3_rx_sdcal1_iter_tmr	8C4Dh

13.4.10.2.136.2 Diagram



13.4.10.2.136.3 Fields

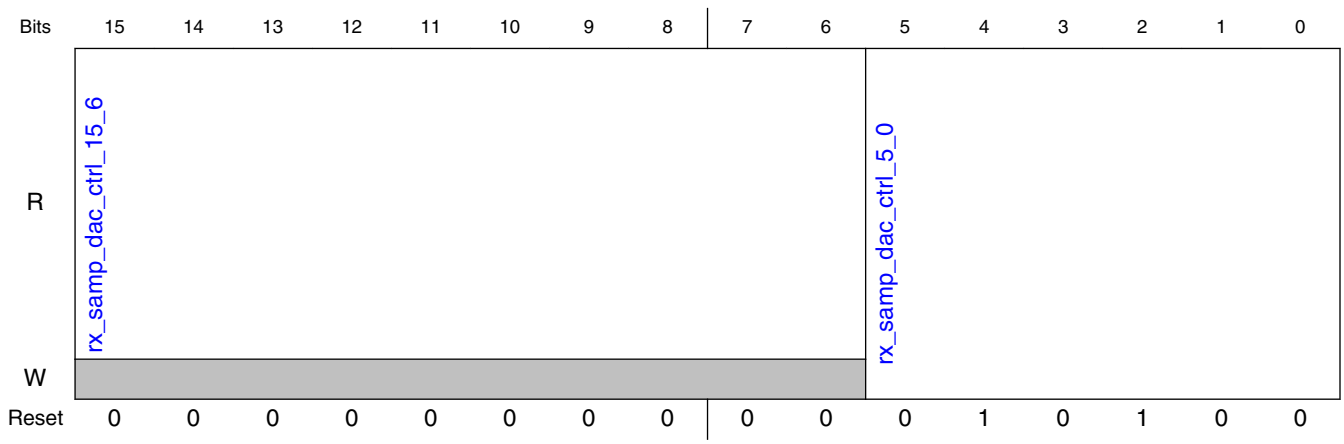
Field	Function
15-9 rx_sdcal1_iter_tmr_15_9	Reserved
8-0 rx_sdcal1_iter_tmr_8_0	Iteration wait timer value: This is the number of

13.4.10.2.137 Sampler error DAC control register (lane0_rx_samp_dac_ctrl - lane3_rx_samp_dac_ctrl)

13.4.10.2.137.1 Offset

Register	Offset
lane0_rx_samp_dac_ctrl	8058h
lane1_rx_samp_dac_ctrl	8458h
lane2_rx_samp_dac_ctrl	8858h
lane3_rx_samp_dac_ctrl	8C58h

13.4.10.2.137.2 Diagram



13.4.10.2.137.3 Fields

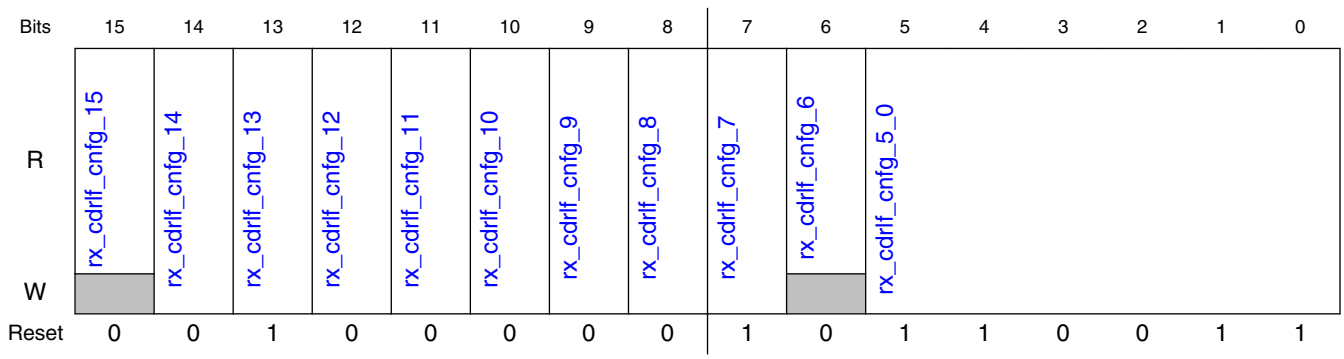
Field	Function
15-6 rx_samp_dac_ctrl_15_6	Reserved
5-0 rx_samp_dac_ctrl_5_0	Sampler error DAC value: Specifies the input value to the sampler error DAC. This value is a twos complement binary number, with the range specified below.

13.4.10.2.138 CDRLF configuration register (lane0_rx_cdrlf_cnfg - lane3_rx_cdrlf_cnfg)

13.4.10.2.138.1 Offset

Register	Offset
lane0_rx_cdrlf_cnfg	8080h
lane1_rx_cdrlf_cnfg	8480h
lane2_rx_cdrlf_cnfg	8880h
lane3_rx_cdrlf_cnfg	8C80h

13.4.10.2.138.2 Diagram



13.4.10.2.138.3 Fields

Field	Function
15 rx_cdrlf_cnfg_15	CDRLF fast phase lock locked detected: This register bit is the current status of the
14 rx_cdrlf_cnfg_14	CDRLF fast phase lock diagnostic enable: Setting this bit will force the
13 rx_cdrlf_cnfg_13	CDRLF fast phase lock enable: Controls the enable of the fast phase lock function.
12 rx_cdrlf_cnfg_12	CDRLF fast frequency lock enable: Controls the enable of the fast frequency lock function.
11 rx_cdrlf_cnfg_11	CDRLF second order loop integrator max clear enable: This signal enables the function in the CDRLF where the second order loop integrator is cleared when it reaches the maximum value.
10 rx_cdrlf_cnfg_10	CDRLF reset on CDRLF PM Accumulator Max: Activating (1b1) this bit will force the CDRLF to be reset when the PM accumulator in the CDRLF reaches is maximum absolute value (the largest positive or negative value).
9 rx_cdrlf_cnfg_9	CDRLF freeze on electrical idle detect: Activating (1b1) this bit will force the CDRLF to be freeze in its current state when the receiver signal detect detects an electrical idle. When high speed data is detected, the freeze will be released.

Table continues on the next page...

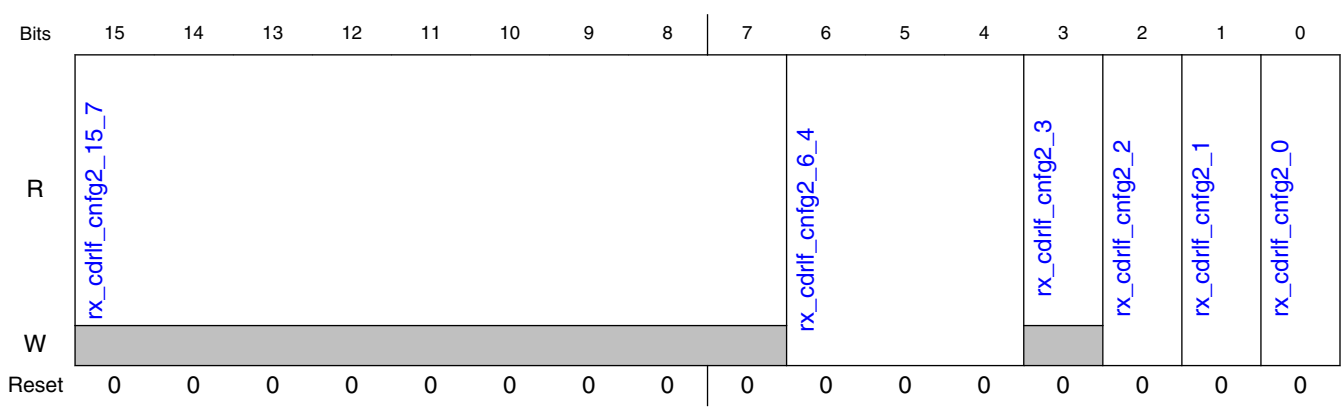
Field	Function
8 rx_cdrlf_cnfg_8	CDRLF reset on electrical idle detect: Activating (1b1) this bit will force the CDRLF to be reset when the receiver signal detect detects an electrical idle. When high speed data is detected, the reset will be released.
7 rx_cdrlf_cnfg_7	CDRLF data filter enable : Enables the filter function for the data that feeds into the CDRLF from the deserializer, using the
6 rx_cdrlf_cnfg_6	Reserved
5-0 rx_cdrlf_cnfg_5_0	CDRLF second order loop integrator threshold : This value is the maximum magnitude the CDRLF second order loop integrator will be allowed to go to.

13.4.10.2.139 CDRLF configuration register 2 (lane0_rx_cdrlf_cnfg2 - lane 3_rx_cdrlf_cnfg2)

13.4.10.2.139.1 Offset

Register	Offset
lane0_rx_cdrlf_cnfg2	8081h
lane1_rx_cdrlf_cnfg2	8481h
lane2_rx_cdrlf_cnfg2	8881h
lane3_rx_cdrlf_cnfg2	8C81h

13.4.10.2.139.2 Diagram



13.4.10.2.139.3 Fields

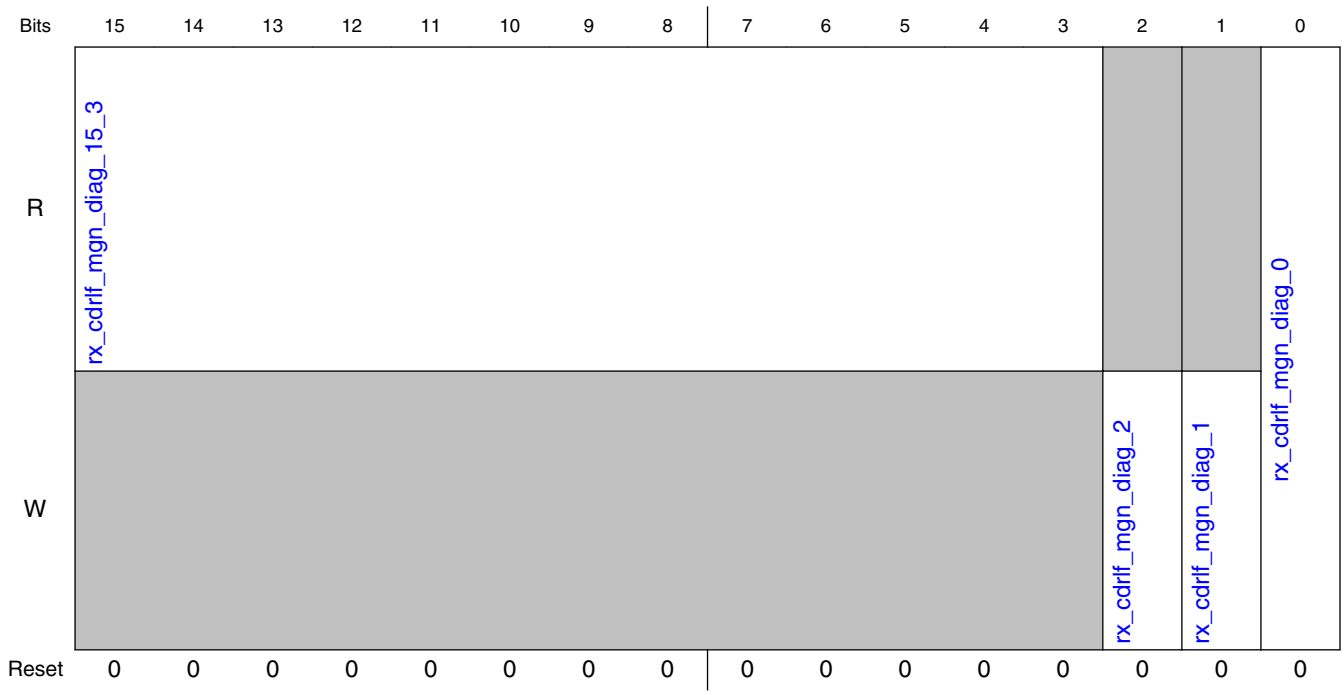
Field	Function
15-7 rx_cdrlf_cnfg2_15_7	Reserved
6-4 rx_cdrlf_cnfg2_6_4	CDRLF diagnostic mode control: This field controls the information driven on the rx_pi_val_ln_{15:0}[7:0] signal, when in diagnostics mode.
3 rx_cdrlf_cnfg2_3	Reserved
2 rx_cdrlf_cnfg2_2	CDLRF reset hold: When active (1b1), the CDRLF will be held in reset beyond the time that it would normally be released by its asynchronous release signals. The CDRLF will be held in reset until this bit is deactivated (1b0).
1 rx_cdrlf_cnfg2_1	CDRLF second order loop disable: Activating (1b1) this bit will disable the CDRLF second order loop.
0 rx_cdrlf_cnfg2_0	CDRLF first order loop disable: Activating (1b1) this bit will disable the CDRLF first order loop.

13.4.10.2.140 CDRLF margin diagnostic register (lane0_rx_cdrlf_mgn_diag - lane3_rx_cdrlf_mgn_diag)

13.4.10.2.140.1 Offset

Register	Offset
lane0_rx_cdrlf_mgn_diag	8082h
lane1_rx_cdrlf_mgn_diag	8482h
lane2_rx_cdrlf_mgn_diag	8882h
lane3_rx_cdrlf_mgn_diag	8C82h

13.4.10.2.140.2 Diagram



13.4.10.2.140.3 Fields

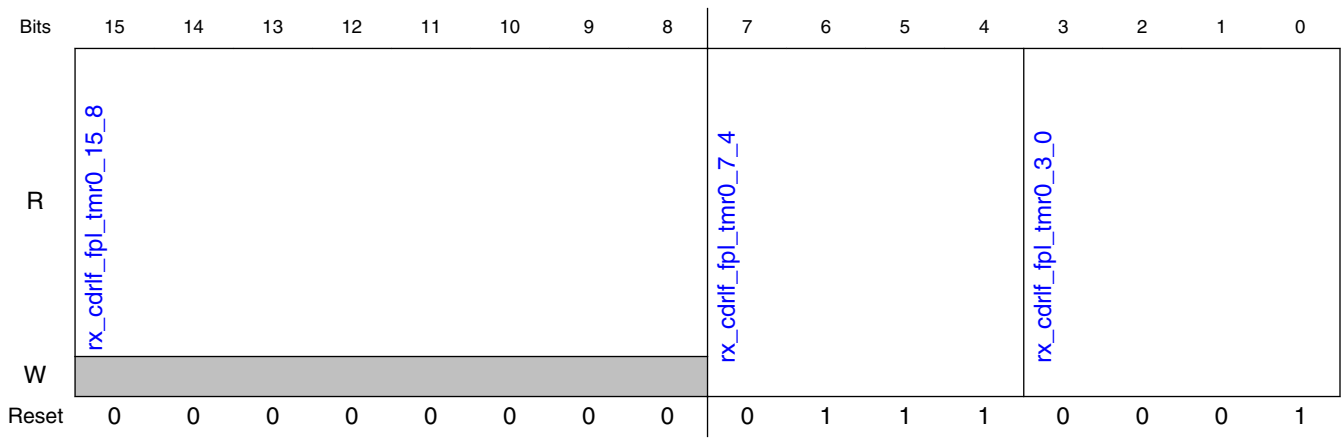
Field	Function
15-3 rx_cdrlf_mgn_diag_15_3	Reserved
2 rx_cdrlf_mgn_diag_2	CDRLF PI override down : When the CDRLF PI override enable function is enabled, writing a 1b1 to this bit will force a down to be generated in the CDRLF PI interface logic.
1 rx_cdrlf_mgn_diag_1	CDRLF PI override up : When the CDRLF PI override enable function is enabled, writing a 1b1 to this bit will force an up to be generated in the CDRLF PI interface logic.
0 rx_cdrlf_mgn_diag_0	CDRLF PI override enable : Setting this bit to 1b1 will enable the CDRLF PI override function, which will allow ups and downs to be forced to the CDRLF PI interface logic from the up and down override bits in this register.

13.4.10.2.141 CDRLF fast phase lock timer value register 0 (lane0_rx_cdrlf_fpl_tmr0 - lane3_rx_cdrlf_fpl_tmr0)

13.4.10.2.141.1 Offset

Register	Offset
lane0_rx_cdrlf_fpl_tmr0	8083h
lane1_rx_cdrlf_fpl_tmr0	8483h
lane2_rx_cdrlf_fpl_tmr0	8883h
lane3_rx_cdrlf_fpl_tmr0	8C83h

13.4.10.2.141.2 Diagram



13.4.10.2.141.3 Fields

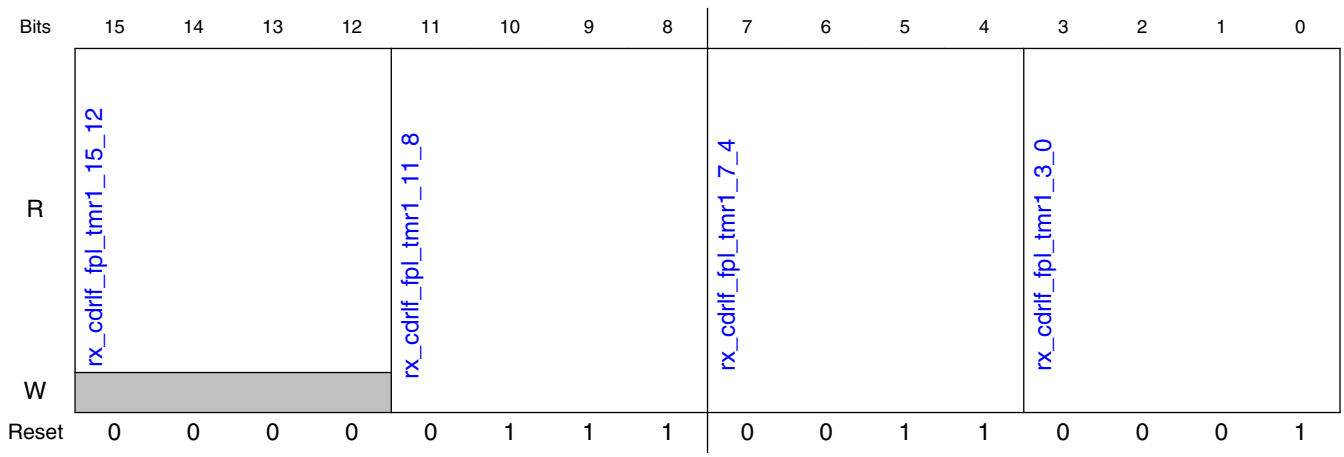
Field	Function
15-8 rx_cdrlf_fpl_tmr0_15_8	Reserved
7-4 rx_cdrlf_fpl_tmr0_7_4	Fast phase lock timer accumulate state time value : Specifies the number of clock cycles minus 1 that the fast phase lock state machine will remain in the accumulate state.
3-0 rx_cdrlf_fpl_tmr0_3_0	Fast phase lock timer delay state time value : Specifies the number of clock cycles minus 1 that the fast phase lock state machine will remain in the delay state.

13.4.10.2.142 CDRLF fast phase lock timer value register 1 (lane0_rx_cdrlf_fpl_tmr1 - lane3_rx_cdrlf_fpl_tmr1)

13.4.10.2.142.1 Offset

Register	Offset
lane0_rx_cdrlf_fpl_tmr1	8084h
lane1_rx_cdrlf_fpl_tmr1	8484h
lane2_rx_cdrlf_fpl_tmr1	8884h
lane3_rx_cdrlf_fpl_tmr1	8C84h

13.4.10.2.142.2 Diagram



13.4.10.2.142.3 Fields

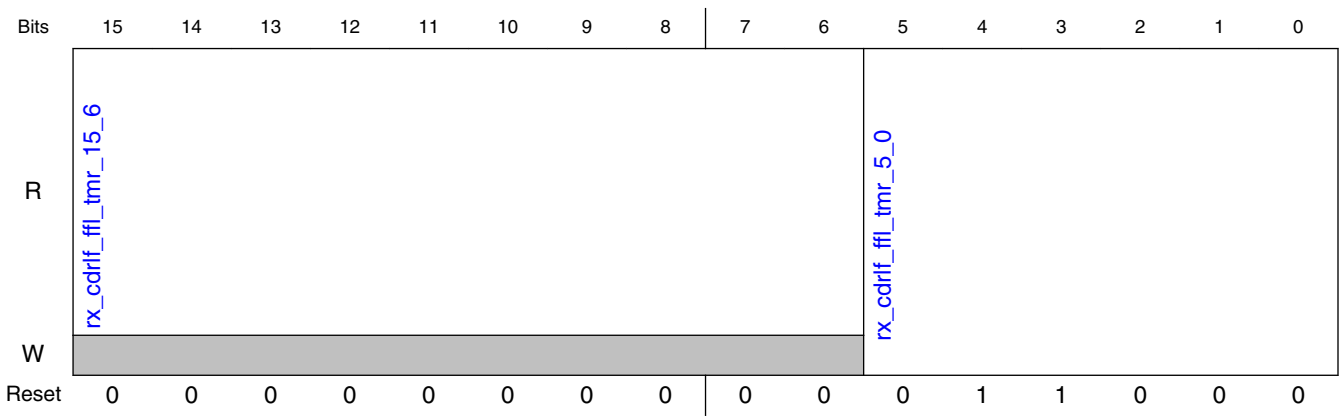
Field	Function
15-12 rx_cdrlf_fpl_tmr1_15_12	Reserved
11-8 rx_cdrlf_fpl_tmr1_11_8	Fast phase lock timer trigger 1 state time value : Specifies the number of clock cycles minus 1 that the fast phase lock state machine will remain in the trigger state the first time it is in that state. The recommended value for this is the number of PI steps required to move the PI 1/4 UI.
7-4 rx_cdrlf_fpl_tmr1_7_4	Fast phase lock timer trigger 2 state time value : Specifies the number of clock cycles minus 1 that the fast phase lock state machine will remain in the trigger state the second time it is in that state. The recommended value for this is the number of PI steps required to move the PI 1/8 UI.
3-0 rx_cdrlf_fpl_tmr1_3_0	Fast phase lock timer trigger 3 state time value : Specifies the number of clock cycles minus 1 that the fast phase lock state machine will remain in the trigger state the third time it is in that state. The recommended value for this is the number of PI steps required to move the PI 1/16 UI.

13.4.10.2.143 CDRLF fast frequency lock timer value register (lane0_rx_cdrlf_ffl_tmr - lane3_rx_cdrlf_ffl_tmr)

13.4.10.2.143.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl_tmr	8085h
lane1_rx_cdrlf_ffl_tmr	8485h
lane2_rx_cdrlf_ffl_tmr	8885h
lane3_rx_cdrlf_ffl_tmr	8C85h

13.4.10.2.143.2 Diagram



13.4.10.2.143.3 Fields

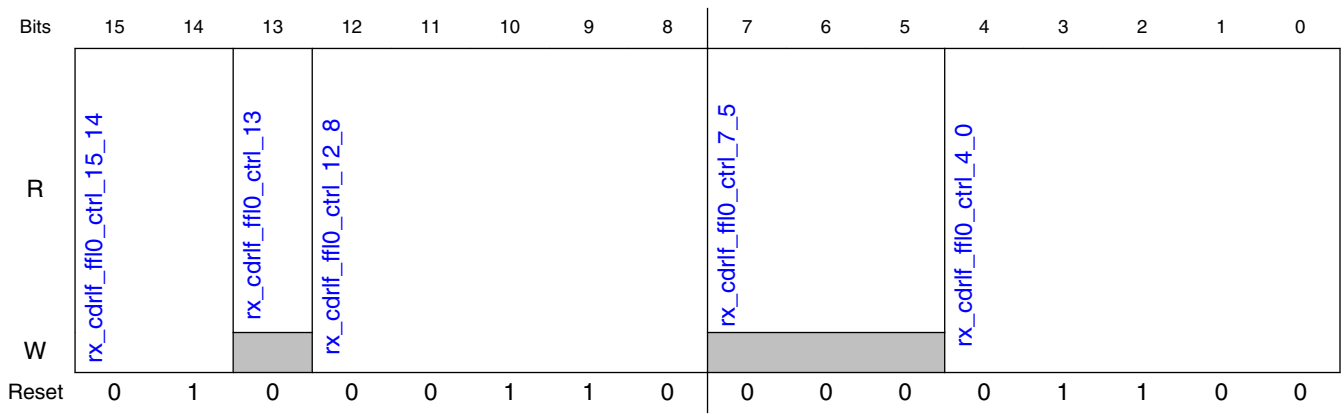
Field	Function
15-6 rx_cdrlf_ffl_tmr_15_6	Reserved
5-0 rx_cdrlf_ffl_tmr_5_0	Fast frequency lock step timer value : Specifies the number of clock cycles for each step of the fast frequency lock process.

13.4.10.2.144 CDRLF fast frequency lock step 0 control register (lane0_rx_cdrlf_ffl0_ctrl - lane3_rx_cdrlf_ffl0_ctrl)

13.4.10.2.144.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl0_ctrl	8088h
lane1_rx_cdrlf_ffl0_ctrl	8488h
lane2_rx_cdrlf_ffl0_ctrl	8888h
lane3_rx_cdrlf_ffl0_ctrl	8C88h

13.4.10.2.144.2 Diagram



13.4.10.2.144.3 Fields

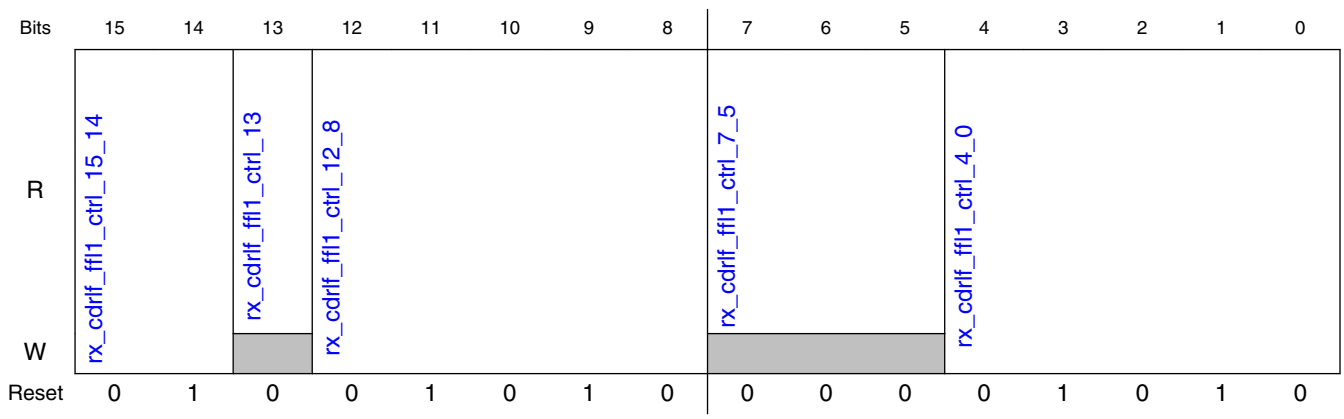
Field	Function
15-14 rx_cdrlf_ffl0_ctrl_15_14	FFL step 0 CDRLF second order loop integrator scaler: Scaler value for the second order loop integrator input. The output of the second order loop sigma delta accumulator is multiplied by the value specified by this signal before it is added to the second order loop integrator.
13 rx_cdrlf_ffl0_ctrl_13	Reserved
12-8 rx_cdrlf_ffl0_ctrl_12_8	FFL step 0 CDRLF second order loop sigma delta update rate: This is the value that is added to or subtracted from the second order loop accumulator register when the serial data sample clock is detected as being out of phase with the serial data on a given parallel data word cycle. This is part of the CDRLF averaging function.
7-5 rx_cdrlf_ffl0_ctrl_7_5	Reserved
4-0 rx_cdrlf_ffl0_ctrl_4_0	FFL step 0 CDRLF first order loop sigma delta update rate: This is the value that is added to or subtracted from the first order loop accumulator register when the serial data sample clock is detected as being out of phase with the serial data on a given parallel data word cycle. This is part of the CDRLF averaging function.

13.4.10.2.145 CDRLF fast frequency lock step 1 control register (lane0_rx_cdrlf_ffl1_ctrl - lane3_rx_cdrlf_ffl1_ctrl)

13.4.10.2.145.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl1_ctrl	8089h
lane1_rx_cdrlf_ffl1_ctrl	8489h
lane2_rx_cdrlf_ffl1_ctrl	8889h
lane3_rx_cdrlf_ffl1_ctrl	8C89h

13.4.10.2.145.2 Diagram



13.4.10.2.145.3 Fields

Field	Function
15-14 rx_cdrlf_ffl1_ctrl_15_14	FFL step 1 CDRLF second order loop integrator scaler
13 rx_cdrlf_ffl1_ctrl_13	Reserved
12-8 rx_cdrlf_ffl1_ctrl_12_8	FFL step 1 CDRLF second order loop sigma delta update rate
7-5 rx_cdrlf_ffl1_ctrl_7_5	Reserved

Table continues on the next page...

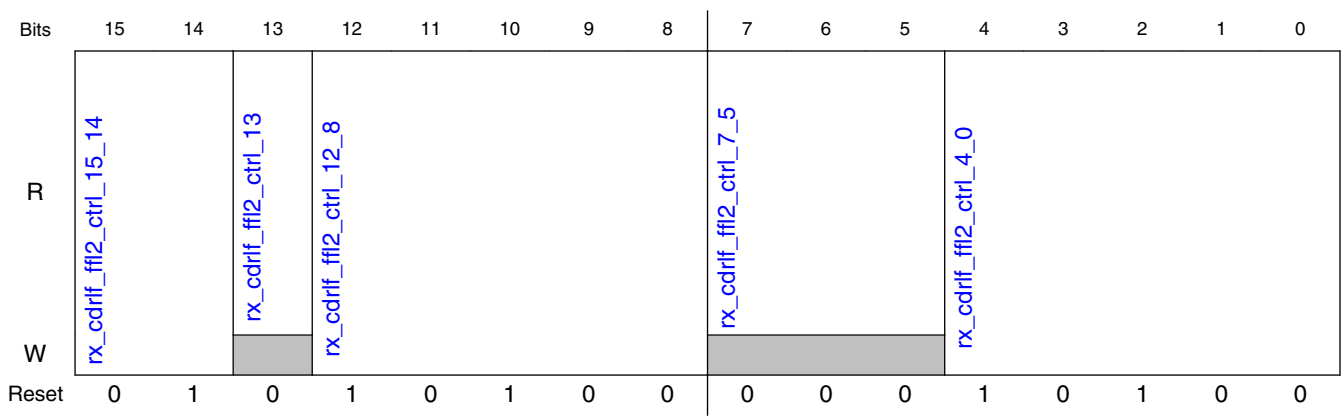
Field	Function
4-0 rx_cdrlf_ffl1_ctrl_4_0	FFL step 1 CDRLF first order loop sigma delta update rate

13.4.10.2.146 CDRLF fast frequency lock step 2 control register (lane0_rx_cdrlf_ffl2_ctrl - lane3_rx_cdrlf_ffl2_ctrl)

13.4.10.2.146.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl2_ctrl	808Ah
lane1_rx_cdrlf_ffl2_ctrl	848Ah
lane2_rx_cdrlf_ffl2_ctrl	888Ah
lane3_rx_cdrlf_ffl2_ctrl	8C8Ah

13.4.10.2.146.2 Diagram



13.4.10.2.146.3 Fields

Field	Function
15-14 rx_cdrlf_ffl2_ctrl_15_14	FFL step 2 CDRLF second order loop integrator scaler
13 rx_cdrlf_ffl2_ctrl_13	Reserved

Table continues on the next page...

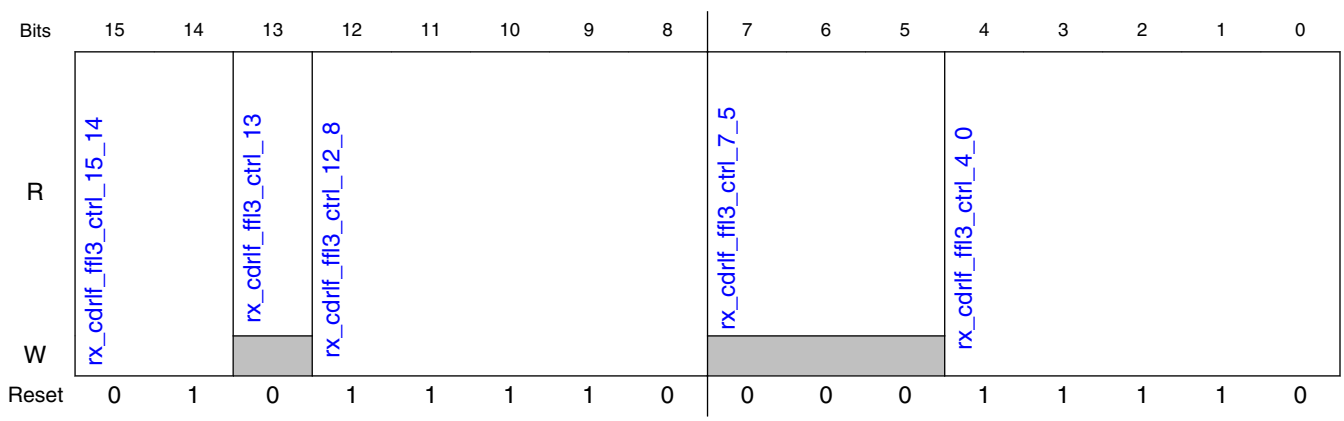
Field	Function
12-8 rx_cdrlf_ffl2_ctrl_12_8	FFL step 2 CDRLF second order loop sigma delta update rate
7-5 rx_cdrlf_ffl2_ctrl_7_5	Reserved
4-0 rx_cdrlf_ffl2_ctrl_4_0	FFL step 2 CDRLF first order loop sigma delta update rate

13.4.10.2.147 CDRLF fast frequency lock step 3 control register (lane0_rx_cdrlf_ffl3_ctrl - lane3_rx_cdrlf_ffl3_ctrl)

13.4.10.2.147.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl3_ctrl	808Bh
lane1_rx_cdrlf_ffl3_ctrl	848Bh
lane2_rx_cdrlf_ffl3_ctrl	888Bh
lane3_rx_cdrlf_ffl3_ctrl	8C8Bh

13.4.10.2.147.2 Diagram



13.4.10.2.147.3 Fields

Field	Function
15-14	FFL step 3 CDRLF second order loop integrator scaler

Table continues on the next page...

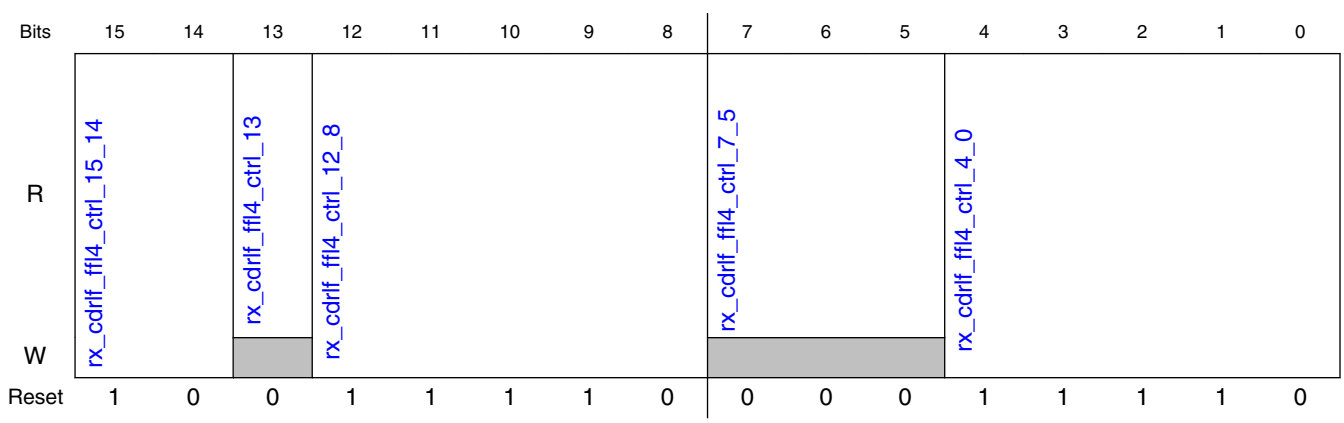
Field	Function
rx_cdrlf_ffl3_ctrl _15_14	
13 rx_cdrlf_ffl3_ctrl _13	Reserved
12-8 rx_cdrlf_ffl3_ctrl _12_8	FFL step 3 CDRLF second order loop sigma delta update rate
7-5 rx_cdrlf_ffl3_ctrl _7_5	Reserved
4-0 rx_cdrlf_ffl3_ctrl _4_0	FFL step 3 CDRLF first order loop sigma delta update rate

13.4.10.2.148 CDRLF fast frequency lock step 4 control register (lane0_rx_cdrlf_ffl4_ctrl - lane3_rx_cdrlf_ffl4_ctrl)

13.4.10.2.148.1 Offset

Register	Offset
lane0_rx_cdrlf_ffl4_ctrl	808Ch
lane1_rx_cdrlf_ffl4_ctrl	848Ch
lane2_rx_cdrlf_ffl4_ctrl	888Ch
lane3_rx_cdrlf_ffl4_ctrl	8C8Ch

13.4.10.2.148.2 Diagram



13.4.10.2.148.3 Fields

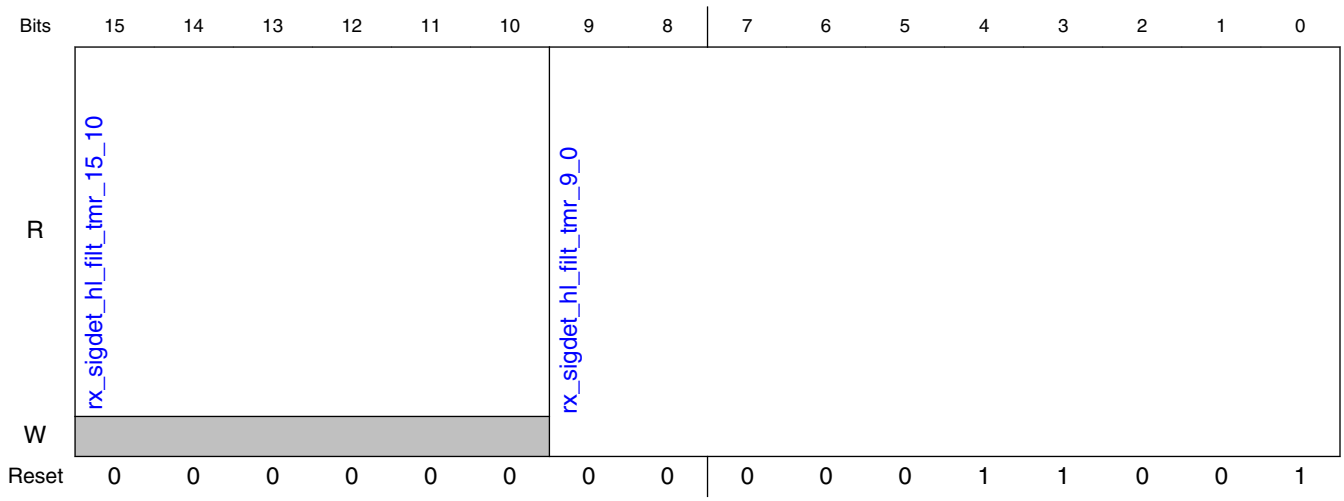
Field	Function
15-14 rx_cdrlf_ffl4_ctrl_15_14	FFL step 4 CDRLF second order loop integrator scaler
13 rx_cdrlf_ffl4_ctrl_13	Reserved
12-8 rx_cdrlf_ffl4_ctrl_12_8	FFL step 4 CDRLF second order loop sigma delta update rate
7-5 rx_cdrlf_ffl4_ctrl_7_5	Reserved
4-0 rx_cdrlf_ffl4_ctrl_4_0	FFL step 4 CDRLF first order loop sigma delta update rate

13.4.10.2.149 Receiver signal detect filter high to low filter timer register (lane0_rx_sigdet_hl_filt_tmr - lane3_rx_sigdet_hl_filt_tmr)

13.4.10.2.149.1 Offset

Register	Offset
lane0_rx_sigdet_hl_filt_tmr	8090h
lane1_rx_sigdet_hl_filt_tmr	8490h
lane2_rx_sigdet_hl_filt_tmr	8890h
lane3_rx_sigdet_hl_filt_tmr	8C90h

13.4.10.2.149.2 Diagram



13.4.10.2.149.3 Fields

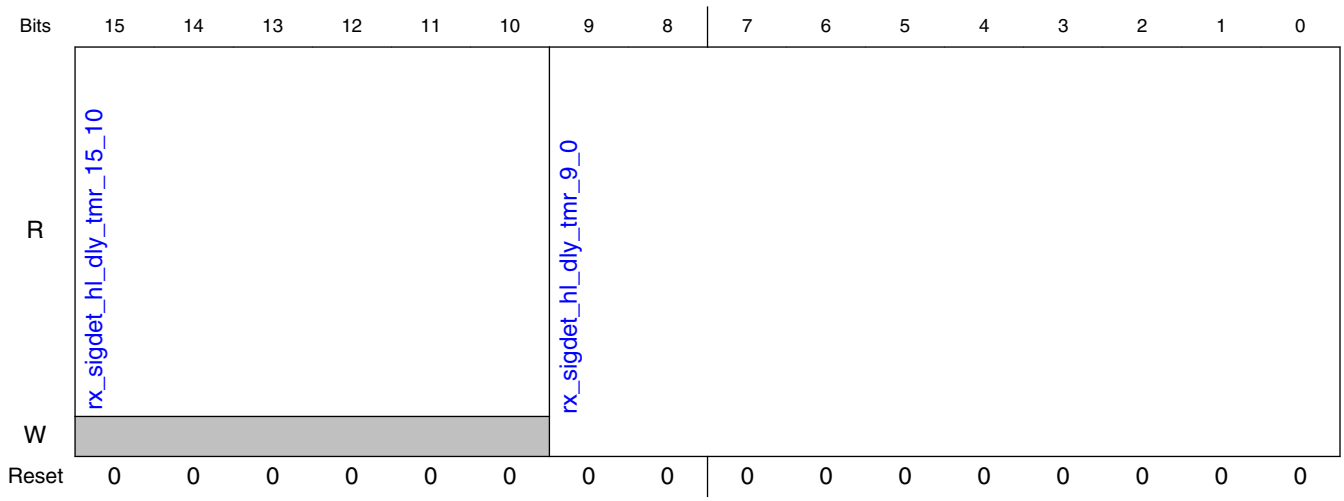
Field	Function
15-10 rx_sigdet_hl_filt_tmr_15_10	Reserved
9-0 rx_sigdet_hl_filt_tmr_9_0	Signal detect filter high to low filter timer value: This is the value loaded into the filter timer in the signal detect high to low filter circuit. This should be set to 1 less than the number of clocks of desired filter time.

13.4.10.2.150 Receiver signal detect filter high to low delay timer register (lane0_rx_sigdet_hl_dly_tmr - lane3_rx_sigdet_hl_dly_tmr)

13.4.10.2.150.1 Offset

Register	Offset
lane0_rx_sigdet_hl_dly_tmr	8091h
lane1_rx_sigdet_hl_dly_tmr	8491h
lane2_rx_sigdet_hl_dly_tmr	8891h
lane3_rx_sigdet_hl_dly_tmr	8C91h

13.4.10.2.150.2 Diagram



13.4.10.2.150.3 Fields

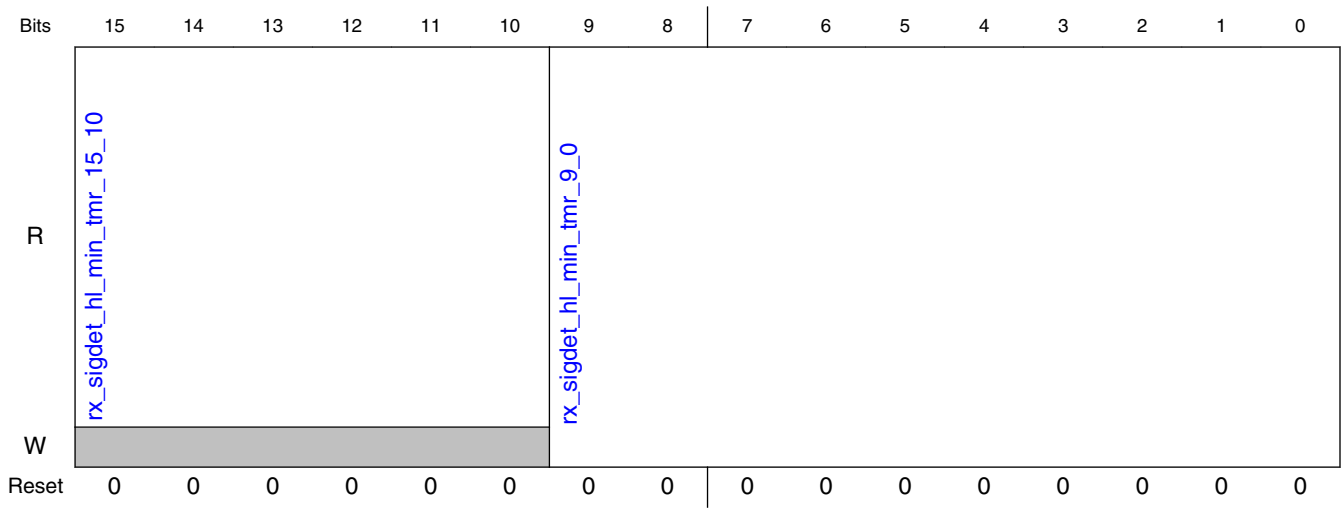
Field	Function
15-10 rx_sigdet_hl_dly_tmr_15_10	Reserved
9-0 rx_sigdet_hl_dly_tmr_9_0	Signal detect filter high to low delay timer value: This is the value loaded into the delay timer in the signal detect high to low filter circuit. This should be set to 1 less than the number of clocks of desired delay time.

13.4.10.2.151 Receiver signal detect filter high to low min timer register (lane0_rx_sigdet_hl_min_tmr - lane3_rx_sigdet_hl_min_tmr)

13.4.10.2.151.1 Offset

Register	Offset
lane0_rx_sigdet_hl_min_tmr	8092h
lane1_rx_sigdet_hl_min_tmr	8492h
lane2_rx_sigdet_hl_min_tmr	8892h
lane3_rx_sigdet_hl_min_tmr	8C92h

13.4.10.2.151.2 Diagram



13.4.10.2.151.3 Fields

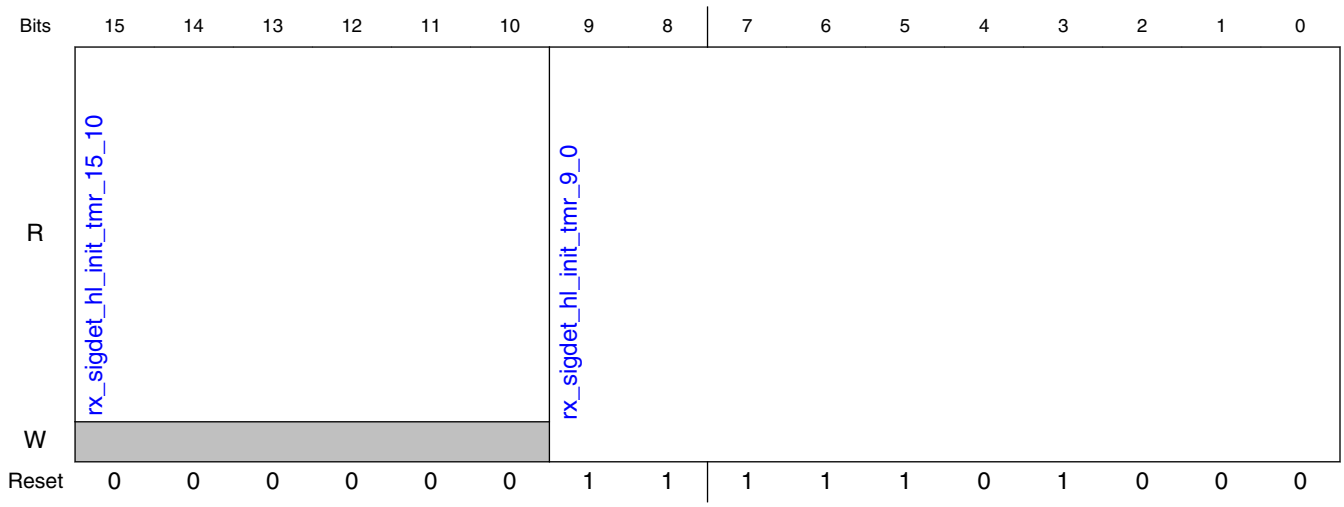
Field	Function
15-10 rx_sigdet_hl_min_tmr_15_10	Reserved
9-0 rx_sigdet_hl_min_tmr_9_0	Signal detect filter high to low min timer value: This is the value loaded into the min timer in the signal detect high to low filter circuit. This should be set to 1 less than the number of clocks of desired min time.

13.4.10.2.152 Receiver signal detect filter high to low init timer register (lane0_rx_sigdet_hl_init_tmr - lane3_rx_sigdet_hl_init_tmr)

13.4.10.2.152.1 Offset

Register	Offset
lane0_rx_sigdet_hl_init_tmr	8093h
lane1_rx_sigdet_hl_init_tmr	8493h
lane2_rx_sigdet_hl_init_tmr	8893h
lane3_rx_sigdet_hl_init_tmr	8C93h

13.4.10.2.152.2 Diagram



13.4.10.2.152.3 Fields

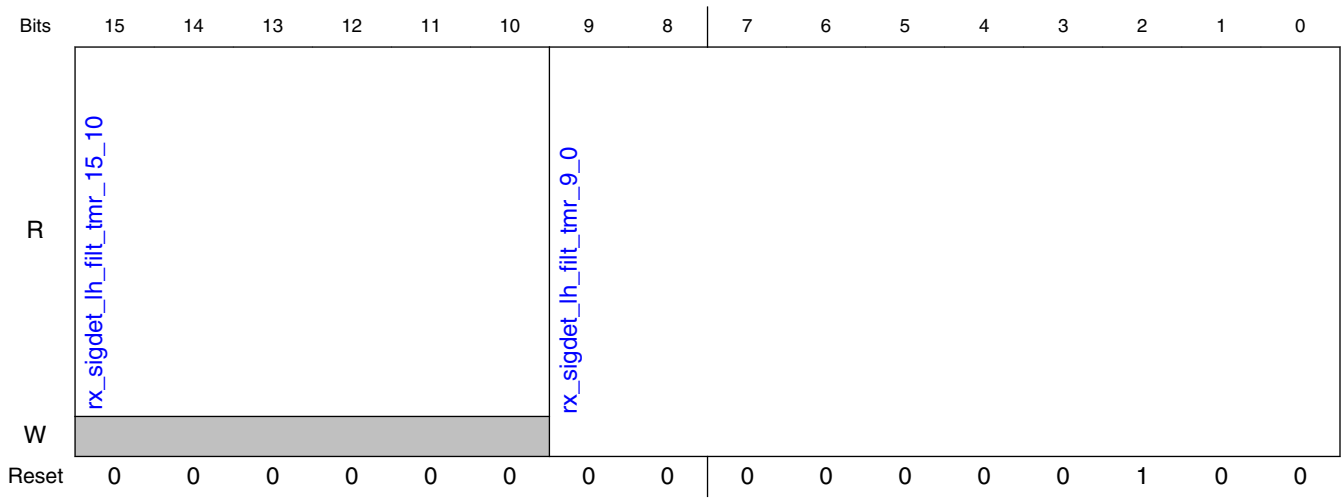
Field	Function
15-10 rx_sigdet_hl_init_tmr_15_10	Reserved
9-0 rx_sigdet_hl_init_tmr_9_0	Signal detect init timer value: This is the value loaded into the initialization timer in the signal detect filter high to low filter circuit.

13.4.10.2.153 Receiver signal detect filter low to high filter timer register (lane0_rx_sigdet_lh_filt_tmr - lane3_rx_sigdet_lh_filt_tmr)

13.4.10.2.153.1 Offset

Register	Offset
lane0_rx_sigdet_lh_filt_tmr	8094h
lane1_rx_sigdet_lh_filt_tmr	8494h
lane2_rx_sigdet_lh_filt_tmr	8894h
lane3_rx_sigdet_lh_filt_tmr	8C94h

13.4.10.2.153.2 Diagram



13.4.10.2.153.3 Fields

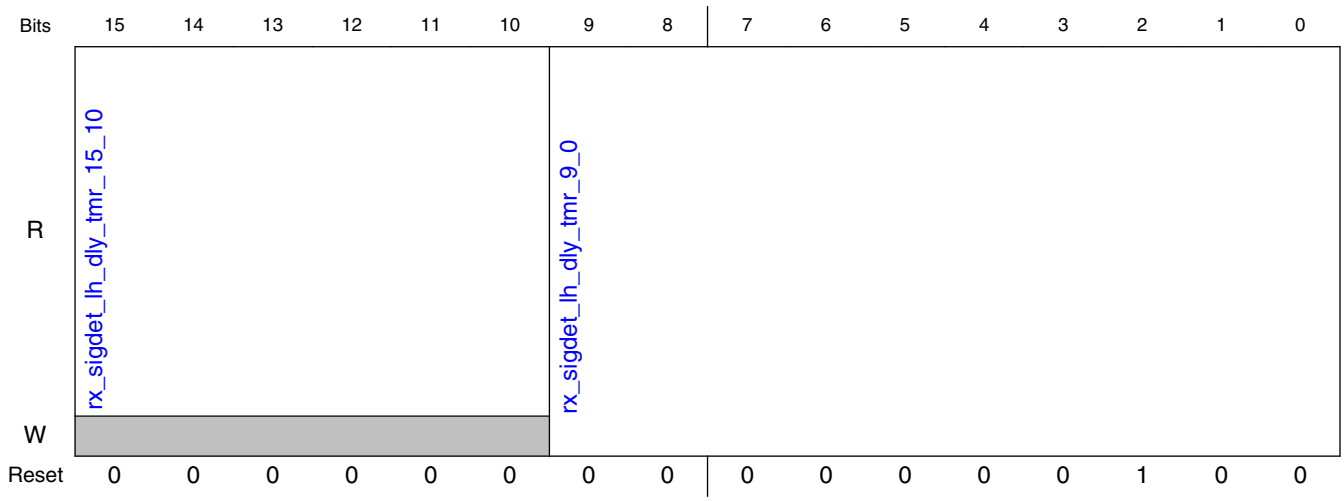
Field	Function
15-10 rx_sigdet_lh_filt_tmr_15_10	Reserved
9-0 rx_sigdet_lh_filt_tmr_9_0	Signal detect filter low to high filter timer value: This is the value loaded into the filter timer in the signal detect low to high filter circuit. This should be set to 6 less than the number of clocks of desired filter time. This takes into account the synchronizing of the signal to be filtered (4 clocks), the time to load the filter timer (1 clock), and that the timer counts to 0. Therefore, the smallest granularity of filtering available here is 6 clocks. This is more of an issue of latency through the filter circuit as the signal initially rises.

13.4.10.2.154 Receiver signal detect filter low to high delay timer register (lane0_rx_sigdet_lh_dly_tmr - lane3_rx_sigdet_lh_dly_tmr)

13.4.10.2.154.1 Offset

Register	Offset
lane0_rx_sigdet_lh_dly_tmr	8095h
lane1_rx_sigdet_lh_dly_tmr	8495h
lane2_rx_sigdet_lh_dly_tmr	8895h
lane3_rx_sigdet_lh_dly_tmr	8C95h

13.4.10.2.154.2 Diagram



13.4.10.2.154.3 Fields

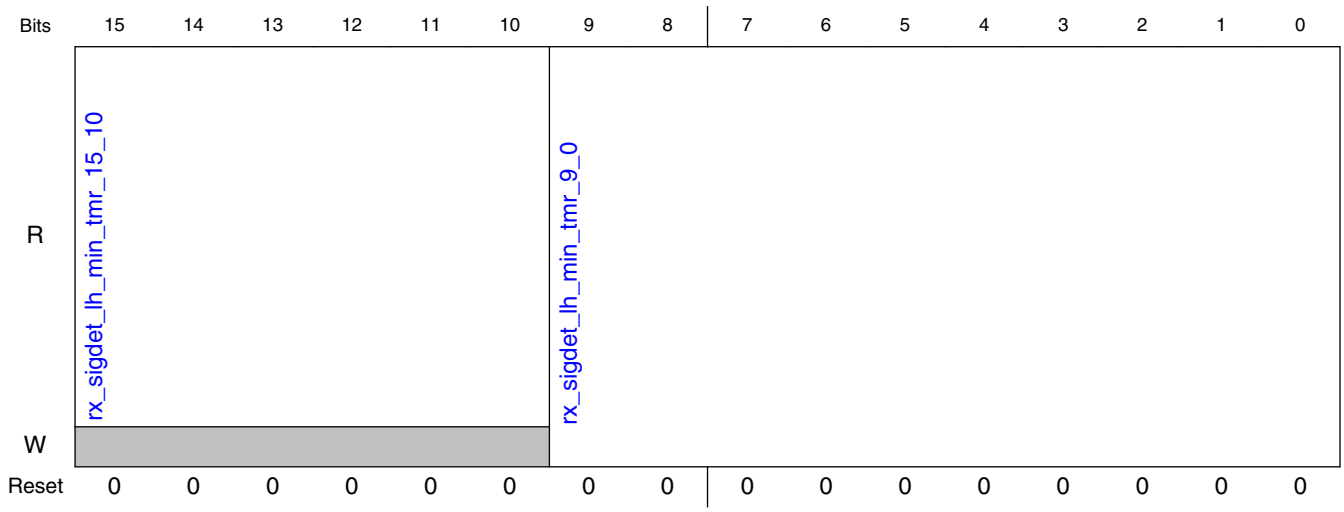
Field	Function
15-10 rx_sigdet_lh_dly_tmr_15_10	Reserved
9-0 rx_sigdet_lh_dly_tmr_9_0	Signal detect filter low to high delay timer value: This is the value loaded into the delay timer in the signal detect low to high filter circuit. This should be set to 1 less than the number of clocks of desired delay time.

13.4.10.2.155 Receiver signal detect filter low to high min timer register (lane0_rx_sigdet_lh_min_tmr - lane3_rx_sigdet_lh_min_tmr)

13.4.10.2.155.1 Offset

Register	Offset
lane0_rx_sigdet_lh_min_tmr	8096h
lane1_rx_sigdet_lh_min_tmr	8496h
lane2_rx_sigdet_lh_min_tmr	8896h
lane3_rx_sigdet_lh_min_tmr	8C96h

13.4.10.2.155.2 Diagram



13.4.10.2.155.3 Fields

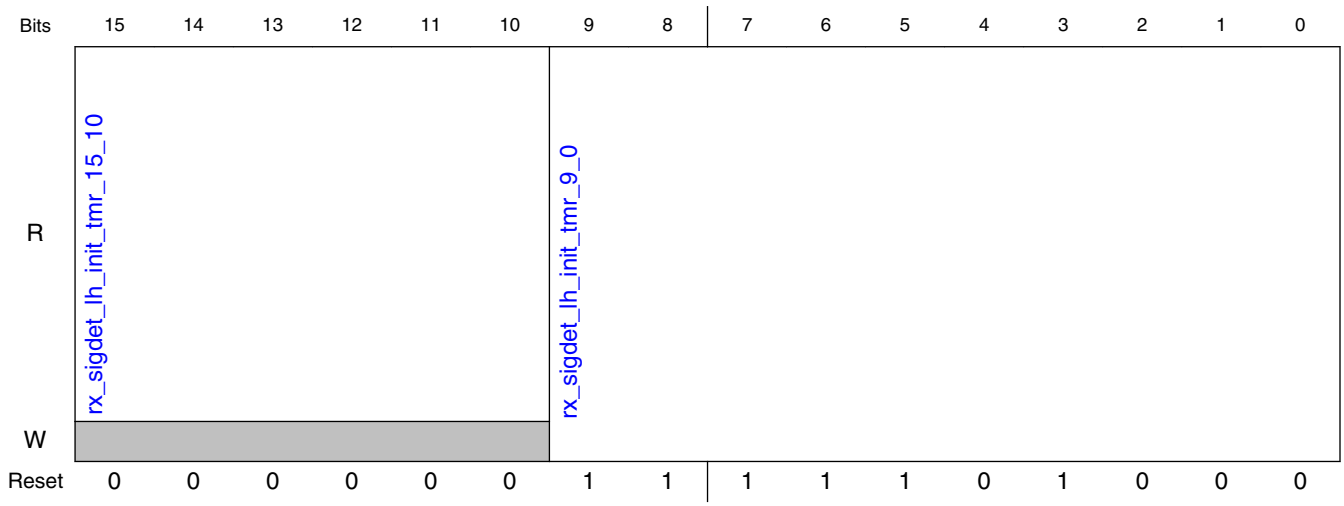
Field	Function
15-10 rx_sigdet_lh_min_tmr_15_10	Reserved
9-0 rx_sigdet_lh_min_tmr_9_0	Signal detect filter low to high min timer value: This is the value loaded into the min timer in the signal detect low to high filter circuit. This should be set to 1 less than the number of clocks of desired min time.

13.4.10.2.156 Receiver signal detect filter low to high init timer register (lane0_rx_sigdet_lh_init_tmr - lane3_rx_sigdet_lh_init_tmr)

13.4.10.2.156.1 Offset

Register	Offset
lane0_rx_sigdet_lh_init_tmr	8097h
lane1_rx_sigdet_lh_init_tmr	8497h
lane2_rx_sigdet_lh_init_tmr	8897h
lane3_rx_sigdet_lh_init_tmr	8C97h

13.4.10.2.156.2 Diagram



13.4.10.2.156.3 Fields

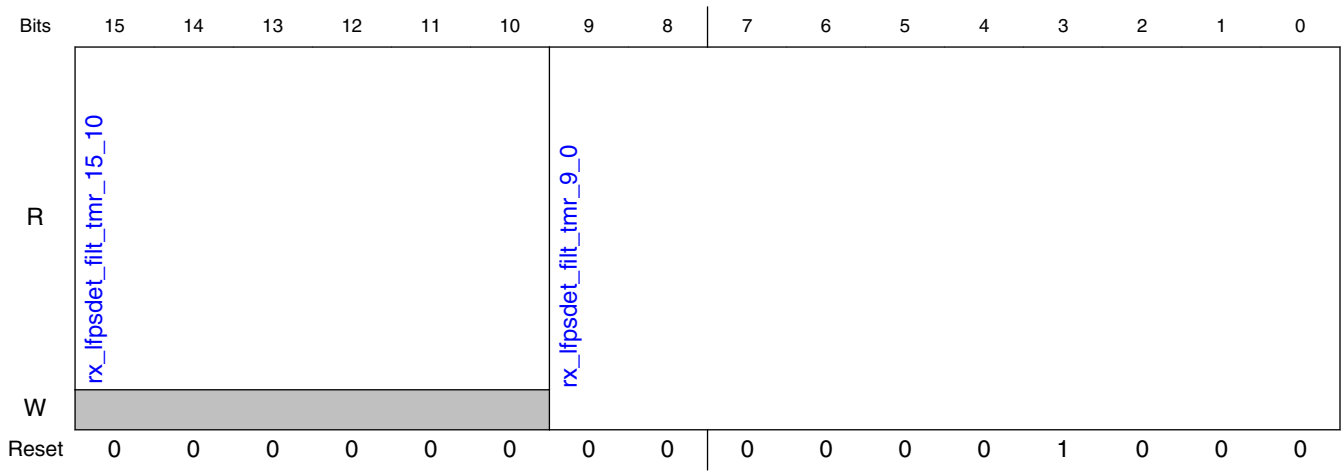
Field	Function
15-10 rx_sigdet_lh_init_tmr_15_10	Reserved
9-0 rx_sigdet_lh_init_tmr_9_0	Signal detect init timer value: This is the value loaded into the initialization timer in the signal detect filter low to high filter circuit.

13.4.10.2.157 Receiver LFPS detect filter filter timer register (lane0_rx_lfpsdet_filt_tmr - lane3_rx_lfpsdet_filt_tmr)

13.4.10.2.157.1 Offset

Register	Offset
lane0_rx_lfpsdet_filt_tmr	8098h
lane1_rx_lfpsdet_filt_tmr	8498h
lane2_rx_lfpsdet_filt_tmr	8898h
lane3_rx_lfpsdet_filt_tmr	8C98h

13.4.10.2.157.2 Diagram



13.4.10.2.157.3 Fields

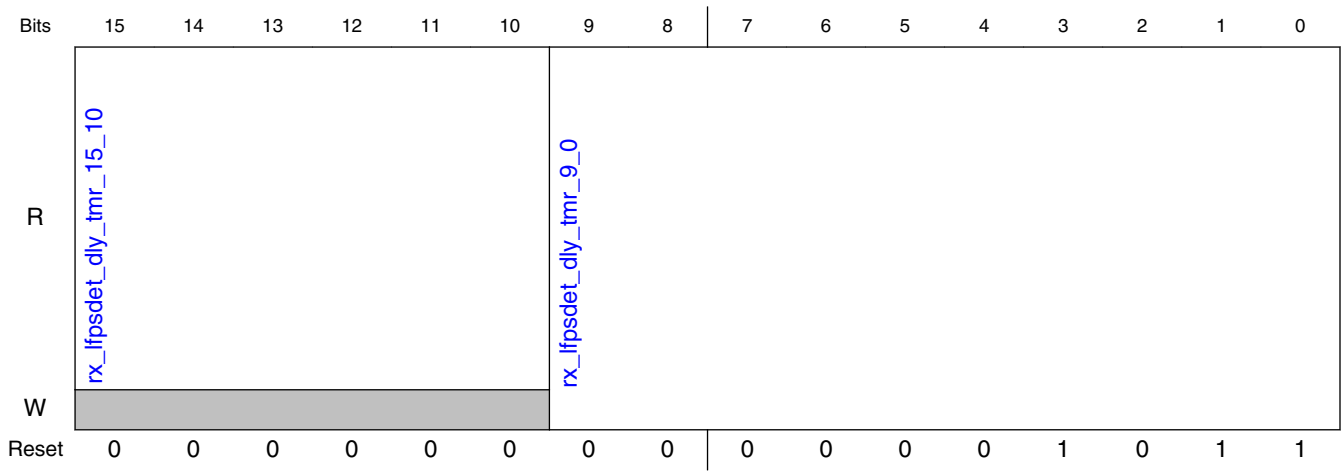
Field	Function
15-10 rx_lfpsdet_filt_tmr_15_10	Reserved
9-0 rx_lfpsdet_filt_tmr_9_0	LFPS detect filter timer value: This is the value loaded into the filter timer in the signal detect filter low circuit. This should be set to 6 less than the number of clocks of desired filter time.

13.4.10.2.158 Receiver LFPS detect filter delay timer register (lane0_rx_lfpsdet_dly_tmr - lane3_rx_lfpsdet_dly_tmr)

13.4.10.2.158.1 Offset

Register	Offset
lane0_rx_lfpsdet_dly_tmr	8099h
lane1_rx_lfpsdet_dly_tmr	8499h
lane2_rx_lfpsdet_dly_tmr	8899h
lane3_rx_lfpsdet_dly_tmr	8C99h

13.4.10.2.158.2 Diagram



13.4.10.2.158.3 Fields

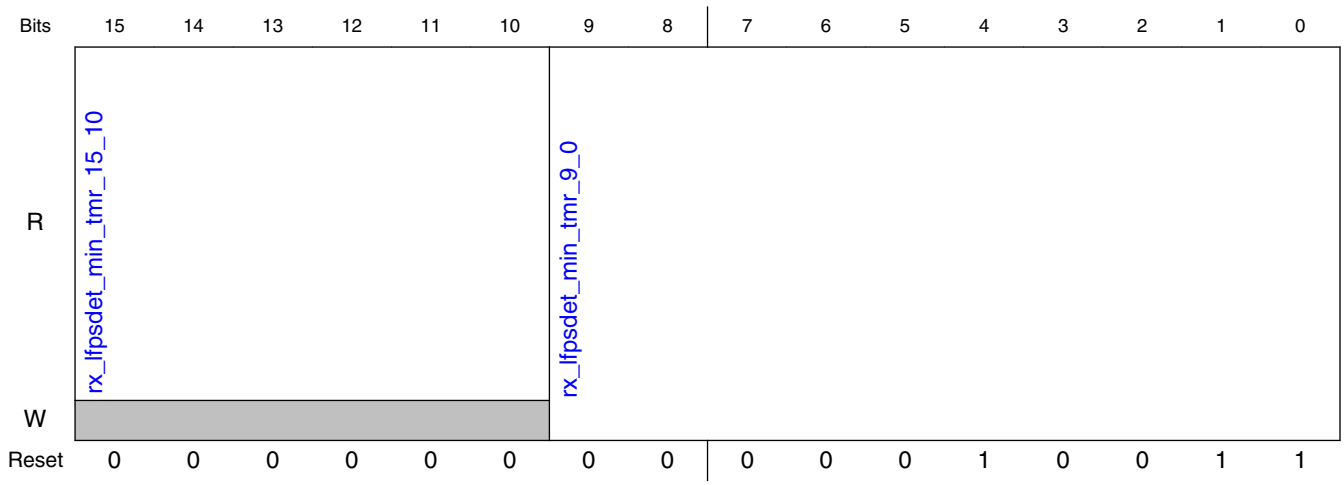
Field	Function
15-10 rx_lfpsdet_dly_tmr_15_10	Reserved
9-0 rx_lfpsdet_dly_tmr_9_0	LFPS detect filter delay timer value: This is the value loaded into the delay timer in the signal detect low to high filter circuit. This should be set to 1 less than the number of clocks of desired delay time.

13.4.10.2.159 Receiver LFPS detect min timer register (lane0_rx_lfpsdet_min_tmr - lane3_rx_lfpsdet_min_tmr)

13.4.10.2.159.1 Offset

Register	Offset
lane0_rx_lfpsdet_min_tmr	809Ah
lane1_rx_lfpsdet_min_tmr	849Ah
lane2_rx_lfpsdet_min_tmr	889Ah
lane3_rx_lfpsdet_min_tmr	8C9Ah

13.4.10.2.159.2 Diagram



13.4.10.2.159.3 Fields

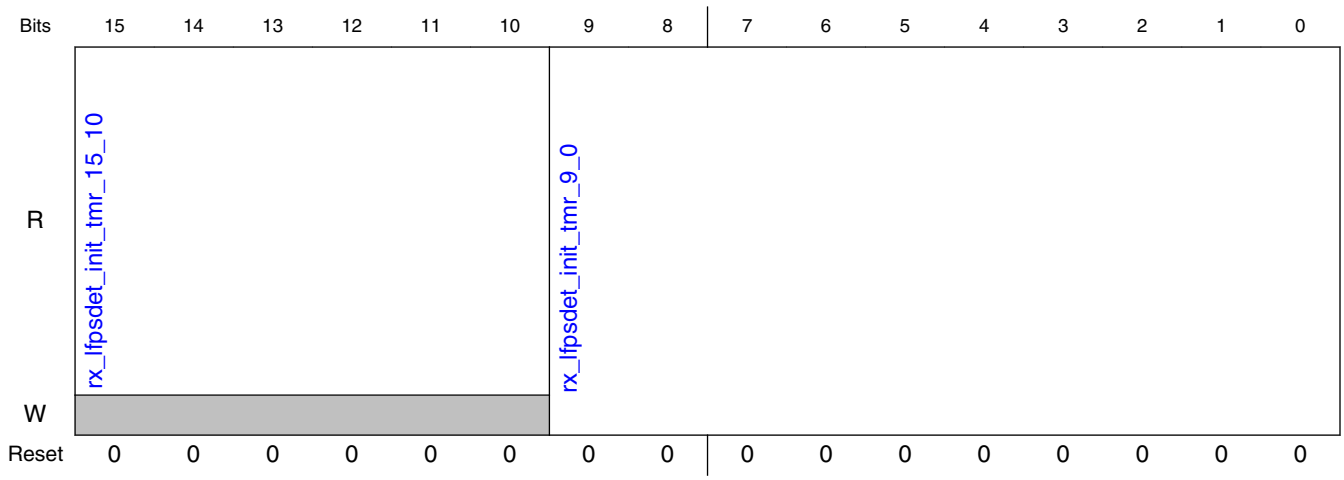
Field	Function
15-10 rx_lfpsdet_min_tmr_15_10	Reserved
9-0 rx_lfpsdet_min_tmr_9_0	LFPS detect min timer value: This is the value loaded into the min timer in the signal detect filter low circuit. This should be set to 1 less than the number of clocks of desired min time.

13.4.10.2.160 Receiver LFPS detect init timer register (lane0_rx_lfpsdet_init_tmr - lane3_rx_lfpsdet_init_tmr)

13.4.10.2.160.1 Offset

Register	Offset
lane0_rx_lfpsdet_init_tmr	809Bh
lane1_rx_lfpsdet_init_tmr	849Bh
lane2_rx_lfpsdet_init_tmr	889Bh
lane3_rx_lfpsdet_init_tmr	8C9Bh

13.4.10.2.160.2 Diagram



13.4.10.2.160.3 Fields

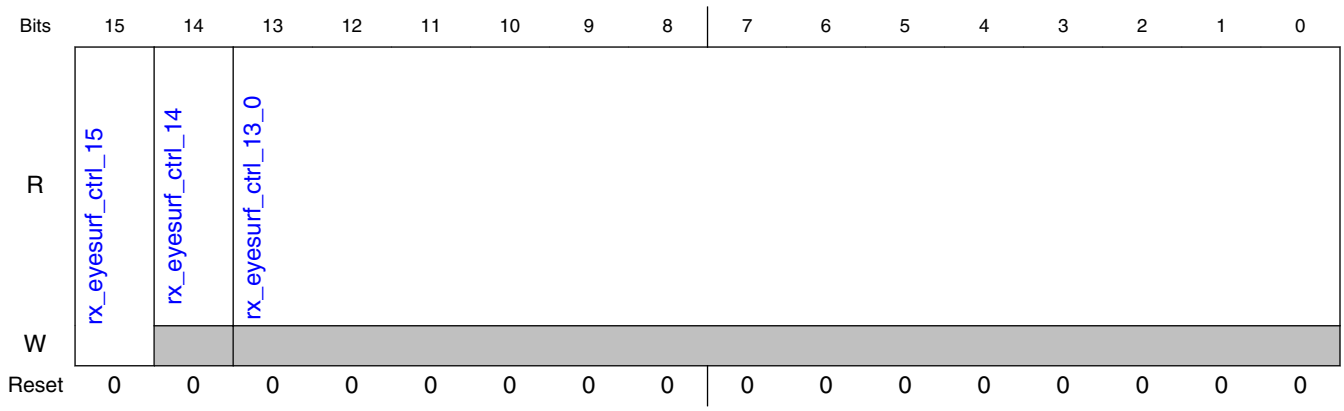
Field	Function
15-10 rx_lfpsdet_init_tmr_15_10	Reserved
9-0 rx_lfpsdet_init_tmr_9_0	LFPS detect init timer value: This is the value loaded into the initialization timer in the signal detect filter low circuit.

13.4.10.2.161 Eye surf control register (lane0_rx_eyesurf_ctrl - lane3_rx_eyesurf_ctrl)

13.4.10.2.161.1 Offset

Register	Offset
lane0_rx_eyesurf_ctrl	80A0h
lane1_rx_eyesurf_ctrl	84A0h
lane2_rx_eyesurf_ctrl	88A0h
lane3_rx_eyesurf_ctrl	8CA0h

13.4.10.2.161.2 Diagram



13.4.10.2.161.3 Fields

Field	Function
15 rx_eyesurf_ctrl_15	Eye surf run: Setting this bit to 1b1 will initiate the eye surf process. This bit must remain set to 1b1 until the eye surf done bit is set.
14 rx_eyesurf_ctrl_14	Eye surf done: When this bit is set to 1b1, the eye surf process has completed. This bit will be cleared after the eye surf run bit is cleared.
13-0 rx_eyesurf_ctrl_13_0	Reserved

13.4.10.2.162 Eye surf timer delay low register (lane0_rx_eyesurf_tmr_del low - lane3_rx_eyesurf_tmr_dellow)

13.4.10.2.162.1 Offset

Register	Offset
lane0_rx_eyesurf_tmr_dellow	80A4h
lane1_rx_eyesurf_tmr_dellow	84A4h
lane2_rx_eyesurf_tmr_dellow	88A4h
lane3_rx_eyesurf_tmr_dellow	8CA4h

13.4.10.2.162.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_eyesurf_tmr_dellow_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.162.3 Fields

Field	Function
15-0 rx_eyesurf_tmr_dellow_15_0	Least significant 16 bits of the delay time: The delay time specifies the number of clock cycles to wait between when a coordinate test point is set, and when to start testing the i and e data.

13.4.10.2.163 Eye surf timer delay high register (lane0_rx_eyesurf_tmr_delhigh - lane3_rx_eyesurf_tmr_delhigh)

13.4.10.2.163.1 Offset

Register	Offset
lane0_rx_eyesurf_tmr_delhigh	80A5h
lane1_rx_eyesurf_tmr_delhigh	84A5h
lane2_rx_eyesurf_tmr_delhigh	88A5h
lane3_rx_eyesurf_tmr_delhigh	8CA5h

13.4.10.2.163.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_eyesurf_tmr_delhigh_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.163.3 Fields

Field	Function
15-0 rx_eyesurf_tmr_delhigh_15_0	Most significant 16 bits of the delay time: The delay time specifies the number of clock cycles to wait between when a coordinate test point is set, and when to start testing the i and e data.

13.4.10.2.164 Eye surf timer test low register (lane0_rx_eyesurf_tmr_testlow - lane3_rx_eyesurf_tmr_testlow)

13.4.10.2.164.1 Offset

Register	Offset
lane0_rx_eyesurf_tmr_testlow	80A6h
lane1_rx_eyesurf_tmr_testlow	84A6h
lane2_rx_eyesurf_tmr_testlow	88A6h
lane3_rx_eyesurf_tmr_testlow	8CA6h

13.4.10.2.164.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_eyesurf_tmr_testlow_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.164.3 Fields

Field	Function
15-0 rx_eyesurf_tmr_testlow_15_0	Least significant 16 bits of the test time: The test time specifies the number of clock cycles to test the i and e data at a given coordinate test point.

13.4.10.2.165 Eye surf timer test high register (lane0_rx_eyesurf_tmr_testhigh - lane3_rx_eyesurf_tmr_testhigh)

13.4.10.2.165.1 Offset

Register	Offset
lane0_rx_eyesurf_tmr_testhigh	80A7h
lane1_rx_eyesurf_tmr_testhigh	84A7h
lane2_rx_eyesurf_tmr_testhigh	88A7h
lane3_rx_eyesurf_tmr_testhigh	8CA7h

13.4.10.2.165.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_eyesurf_tmr_testhigh_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.165.3 Fields

Field	Function
15-0 rx_eyesurf_tmr_testhigh_15_0	Most significant 16 bits of the test time: The test time specifies the number of clock cycles to test the i and e data at a given coordinate test point.

13.4.10.2.166 Eye surf north south test point coordinate register (lane0_rx_eyesurf_ns_coord - lane3_rx_eyesurf_ns_coord)

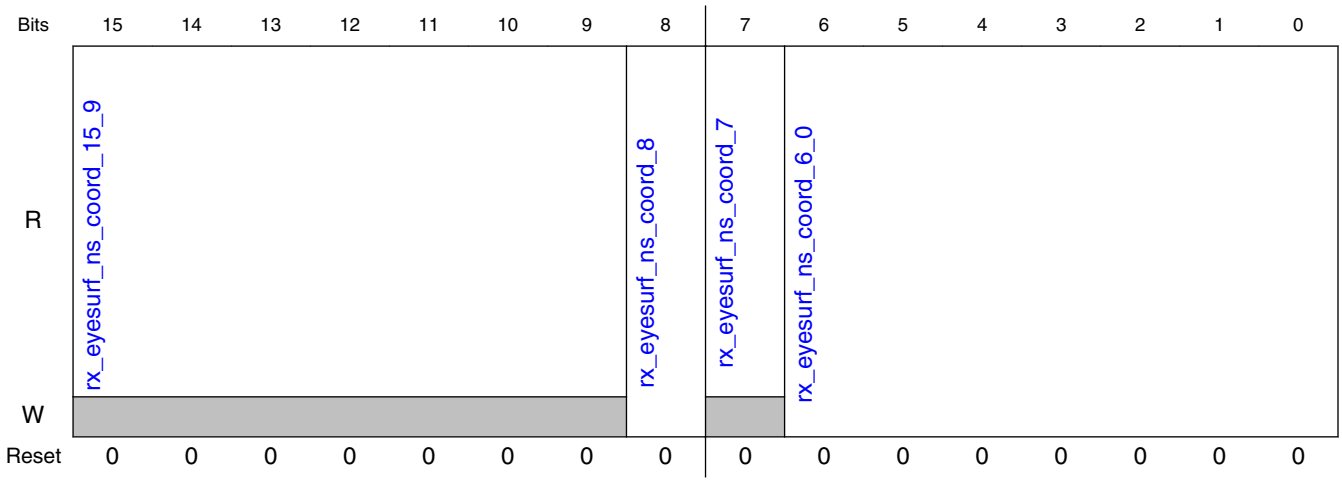
13.4.10.2.166.1 Offset

Register	Offset
lane0_rx_eyesurf_ns_coord	80A8h
lane1_rx_eyesurf_ns_coord	84A8h

Table continues on the next page...

Register	Offset
lane2_rx_eyesurf_ns_coord	88A8h
lane3_rx_eyesurf_ns_coord	8CA8h

13.4.10.2.166.2 Diagram



13.4.10.2.166.3 Fields

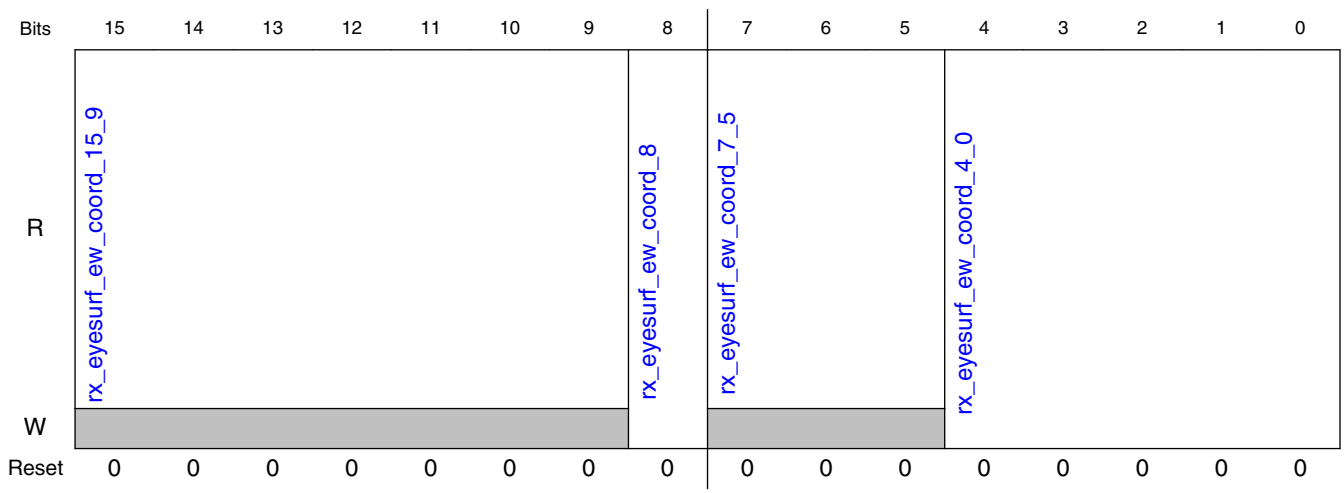
Field	Function
15-9 rx_eyesurf_ns_coord_15_9	Reserved
8 rx_eyesurf_ns_coord_8	Test point coordinate north south direction : Indicates whether the desired test point is in the
7 rx_eyesurf_ns_coord_7	Reserved
6-0 rx_eyesurf_ns_coord_6_0	Test point coordinate north south offset : Indicates how many steps in the north or south

13.4.10.2.167 Eye surf east west test point coordinate register (lane0_rx_eyesurf_ew_coord - lane3_rx_eyesurf_ew_coord)

13.4.10.2.167.1 Offset

Register	Offset
lane0_rx_eyesurf_ew_coord	80A9h
lane1_rx_eyesurf_ew_coord	84A9h
lane2_rx_eyesurf_ew_coord	88A9h
lane3_rx_eyesurf_ew_coord	8CA9h

13.4.10.2.167.2 Diagram



13.4.10.2.167.3 Fields

Field	Function
15-9 rx_eyesurf_ew_coord_15_9	Reserved
8 rx_eyesurf_ew_coord_8	Test point coordinate east west direction : Indicates whether the desired test point is in the
7-5 rx_eyesurf_ew_coord_7_5	Reserved
4-0 rx_eyesurf_ew_coord_4_0	Test point coordinate east west offset : Indicates how many steps in the east or west

13.4.10.2.168 Eye surf bit error count register (lane0_rx_eyesurf_errcnt - lane3_rx_eyesurf_errcnt)

13.4.10.2.168.1 Offset

Register	Offset
lane0_rx_eyesurf_errcnt	80AAh
lane1_rx_eyesurf_errcnt	84AAh
lane2_rx_eyesurf_errcnt	88AAh
lane3_rx_eyesurf_errcnt	8CAAh

13.4.10.2.168.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_eyesurf_errcnt_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.168.3 Fields

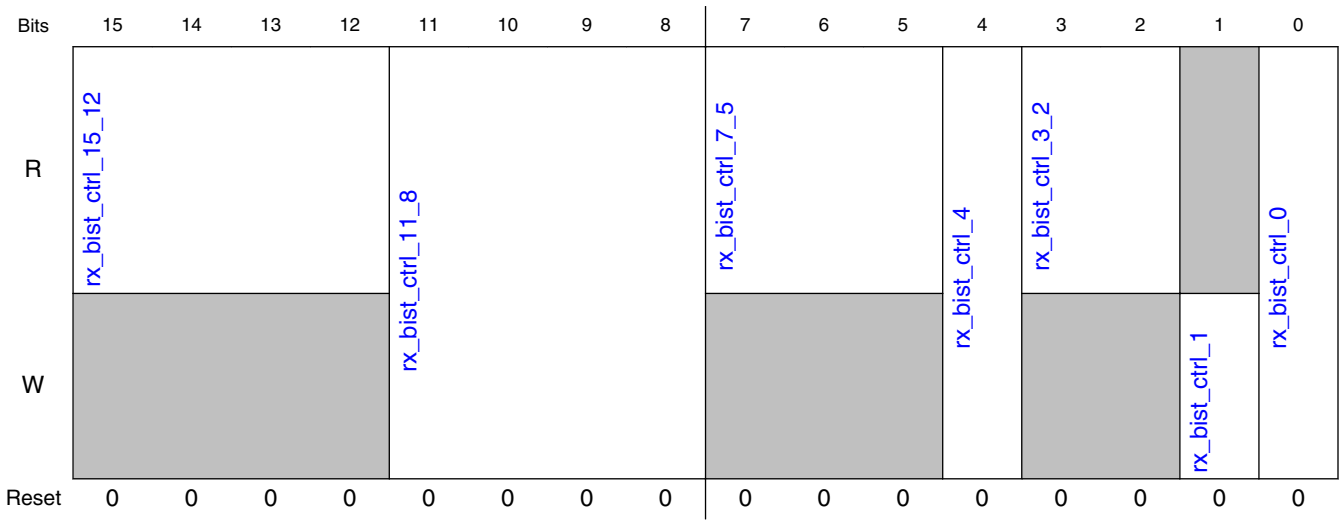
Field	Function
15-0 rx_eyesurf_errcnt_15_0	Test point bit error count : The total number of bit errors that were detected for a given run of the eye surf function. This register is only valid after the eye surf done bit in the

13.4.10.2.169 Receiver BIST control register (lane0_rx_bist_ctrl - lane3_rx_bist_ctrl)

13.4.10.2.169.1 Offset

Register	Offset
lane0_rx_bist_ctrl	80B0h
lane1_rx_bist_ctrl	84B0h
lane2_rx_bist_ctrl	88B0h
lane3_rx_bist_ctrl	8CB0h

13.4.10.2.169.2 Diagram



13.4.10.2.169.3 Fields

Field	Function
15-12 rx_bist_ctrl_15_12	Reserved
11-8 rx_bist_ctrl_11_8	Receiver BIST mode: Controls which mode the BIST will operate in. The value of this field must match the corresponding field for the receive BIST controller. The following are the values used for this field, and what BIST mode they correspond to.
7-5 rx_bist_ctrl_7_5	Reserved
4 rx_bist_ctrl_4	Receiver BIST error reset: Writing this bit is set to a 1b1 will hold the error indicators in the receive BIST logic in reset. When this signal goes active, the
3-2 rx_bist_ctrl_3_2	Reserved
1 rx_bist_ctrl_1	Receiver BIST user defined data FIFO clear: Writing a 1b1 to this bit will clear the receiver BIST user defined data FIFO.
0 rx_bist_ctrl_0	Receiver BIST enable: This bit enables the receiver BIST function.

13.4.10.2.170 Receiver BIST sync count register (lane0_rx_bist_synccnt - lane3_rx_bist_synccnt)

13.4.10.2.170.1 Offset

Register	Offset
lane0_rx_bist_synccnt	80B1h
lane1_rx_bist_synccnt	84B1h
lane2_rx_bist_synccnt	88B1h
lane3_rx_bist_synccnt	8CB1h

13.4.10.2.170.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_bist_synccnt_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.170.3 Fields

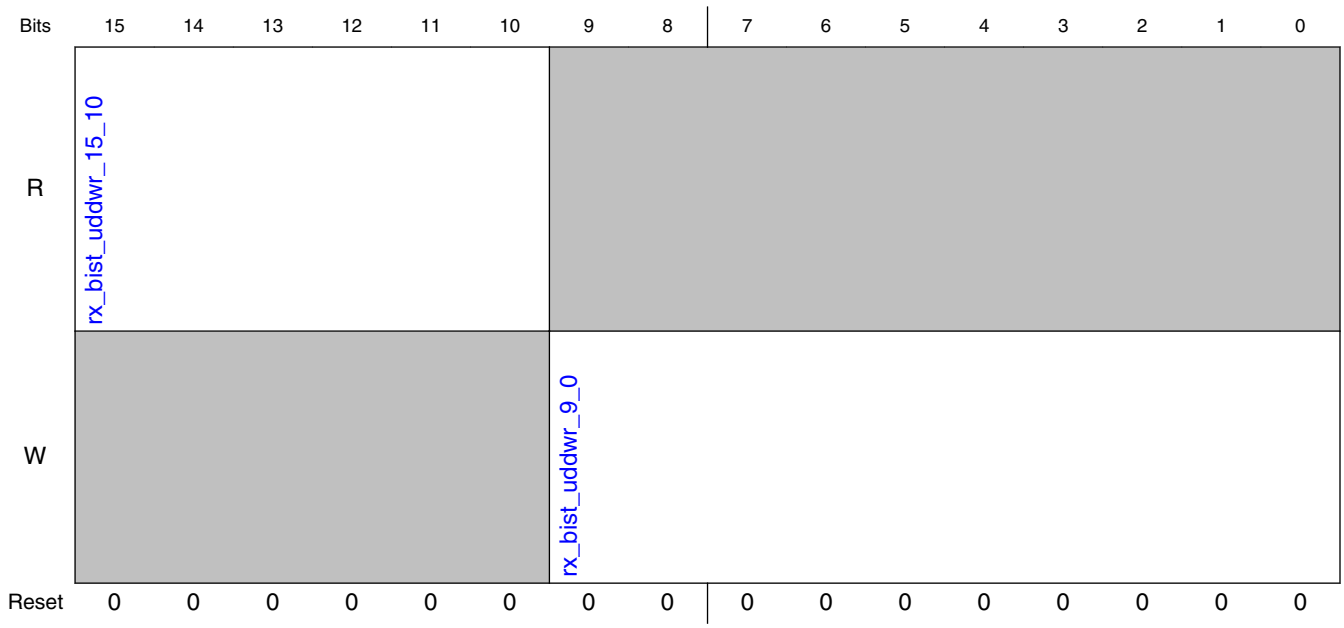
Field	Function
15-0 rx_bist_synccnt_15_0	Receiver BIST sync count: This field controls the value of the RX BIST sync count. The sync count indicates the number of consecutive received data words with no BIST bit errors that must be received in order for the RX BIST module to be considered synced to the input data stream.

13.4.10.2.171 Receiver BIST user defined data write register (lane0_rx_bist_uddwr - lane3_rx_bist_uddwr)

13.4.10.2.171.1 Offset

Register	Offset
lane0_rx_bist_uddwr	80B2h
lane1_rx_bist_uddwr	84B2h
lane2_rx_bist_uddwr	88B2h
lane3_rx_bist_uddwr	8CB2h

13.4.10.2.171.2 Diagram



13.4.10.2.171.3 Fields

Field	Function
15-10 rx_bist_uddwr_15_10	Reserved
9-0 rx_bist_uddwr_9_0	Receiver BIST user defined data: Writing a data word to this field will result in that data word being placed in the next available position in the receiver BIST user defined data FIFO. Note, when in 20 bit mode, all 10 of these bits are used. When in 16 bit mode, only the least significant 8 bits are used. Note that the FIFO in this implementation is 18 words deep. It is up to the user to not write more than 18 words of data into this FIFO. Exceeding this amount of data will generate unpredictable results.

13.4.10.2.172 Receiver BIST error count register (lane0_rx_bist_errcnt - lane3_rx_bist_errcnt)

13.4.10.2.172.1 Offset

Register	Offset
lane0_rx_bist_errcnt	80B3h
lane1_rx_bist_errcnt	84B3h
lane2_rx_bist_errcnt	88B3h
lane3_rx_bist_errcnt	8CB3h

13.4.10.2.172.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_bist_errcnt_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.172.3 Fields

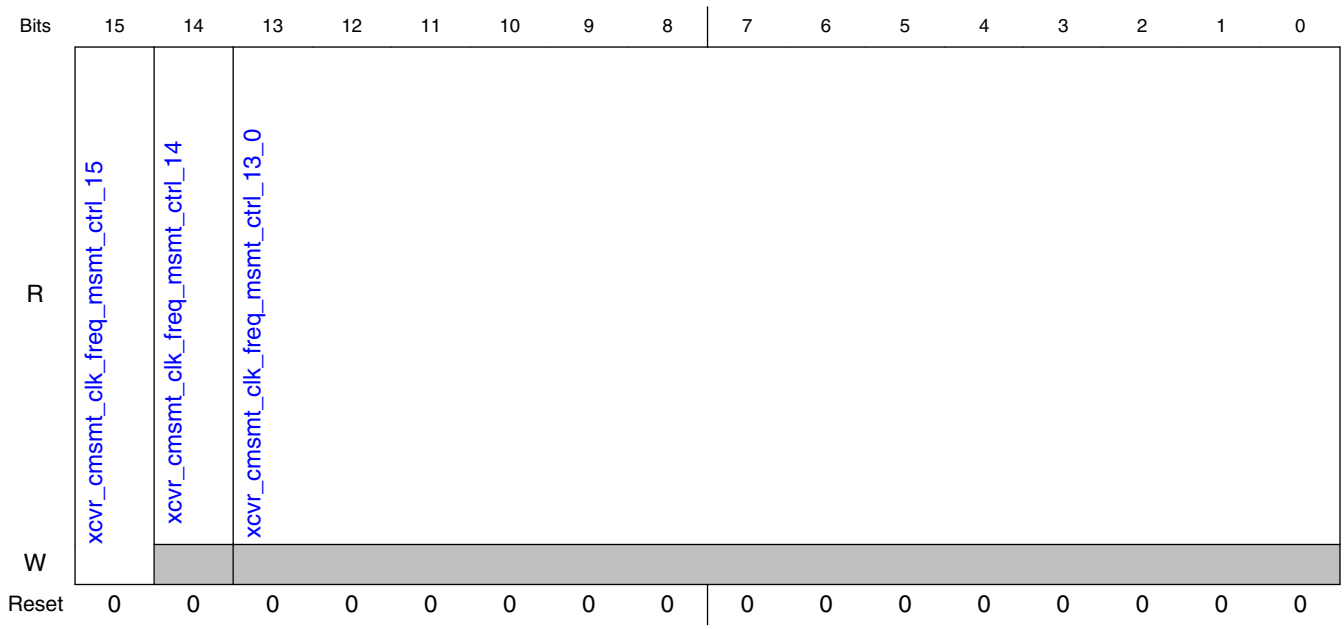
Field	Function
15-0 rx_bist_errcnt_15_0	Receiver BIST error count: Indicates the number of BIST errors that have been observed by the receive BIST logic, since the last time the BIST error indicator logic was reset or restarted. This counter increments up to a maximum value of 16hFFFF.

13.4.10.2.173 Clock frequency measurement control register (lane0_xcvr_cmsmt_clk_freq_msmt_ctrl - lane3_xcvr_cmsmt_clk_freq_msmt_ctrl)

13.4.10.2.173.1 Offset

Register	Offset
lane0_xcvr_cmsmt_clk_freq_msmt_ctrl	80B4h
lane1_xcvr_cmsmt_clk_freq_msmt_ctrl	84B4h
lane2_xcvr_cmsmt_clk_freq_msmt_ctrl	88B4h
lane3_xcvr_cmsmt_clk_freq_msmt_ctrl	8CB4h

13.4.10.2.173.2 Diagram



13.4.10.2.173.3 Fields

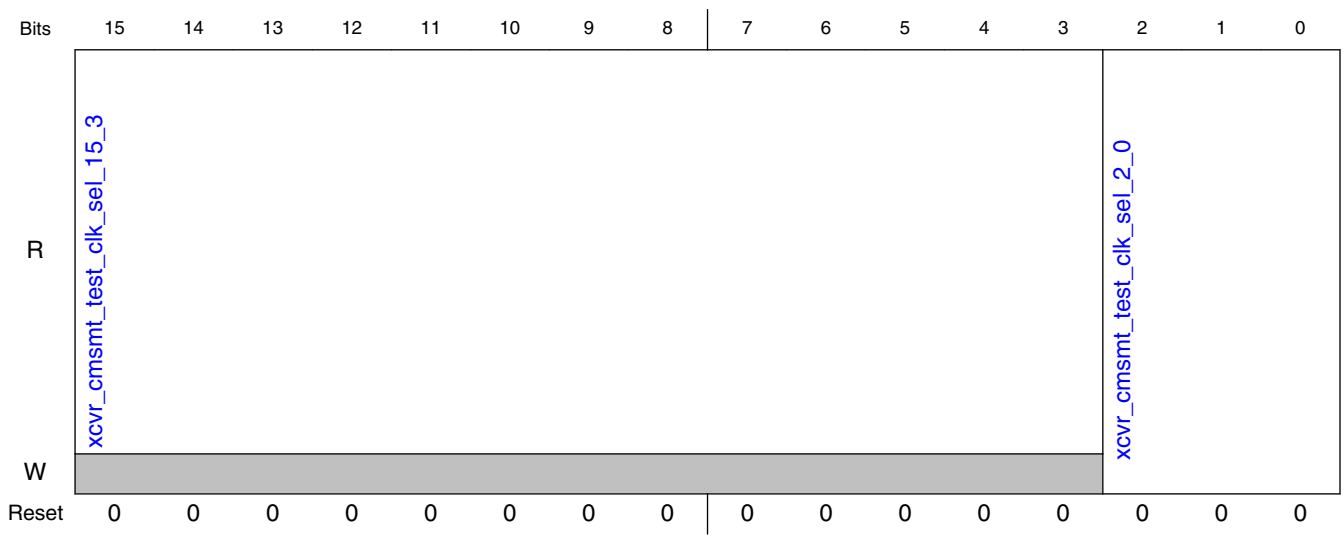
Field	Function
15 xcvr_cmsmt_clk_freq_msmt_ctrl_15	Run test clock measurement: Activating (1b1) this bit will run the test clock measurement process. This bit must remain active until the test clock measurement process is complete, as indicated by the test clock measurement done bit in this register. To start another measurement process, this bit must first be driven inactive then driven active again.
14 xcvr_cmsmt_clk_freq_msmt_ctrl_14	Test clock measurement done: This bit will be set to 1b1 when the test clock measurement process is complete. It will be cleared by the deactivation of the start test clock measurement bit in this register.
13-0 xcvr_cmsmt_clk_freq_msmt_ctrl_13_0	Reserved

13.4.10.2.174 Test clock selection register (lane0_xcvr_cmsmt_test_clk_sel - lane3_xcvr_cmsmt_test_clk_sel)

13.4.10.2.174.1 Offset

Register	Offset
lane0_xcvr_cmsmt_test_clk_sel	80B5h
lane1_xcvr_cmsmt_test_clk_sel	84B5h
lane2_xcvr_cmsmt_test_clk_sel	88B5h
lane3_xcvr_cmsmt_test_clk_sel	8CB5h

13.4.10.2.174.2 Diagram



13.4.10.2.174.3 Fields

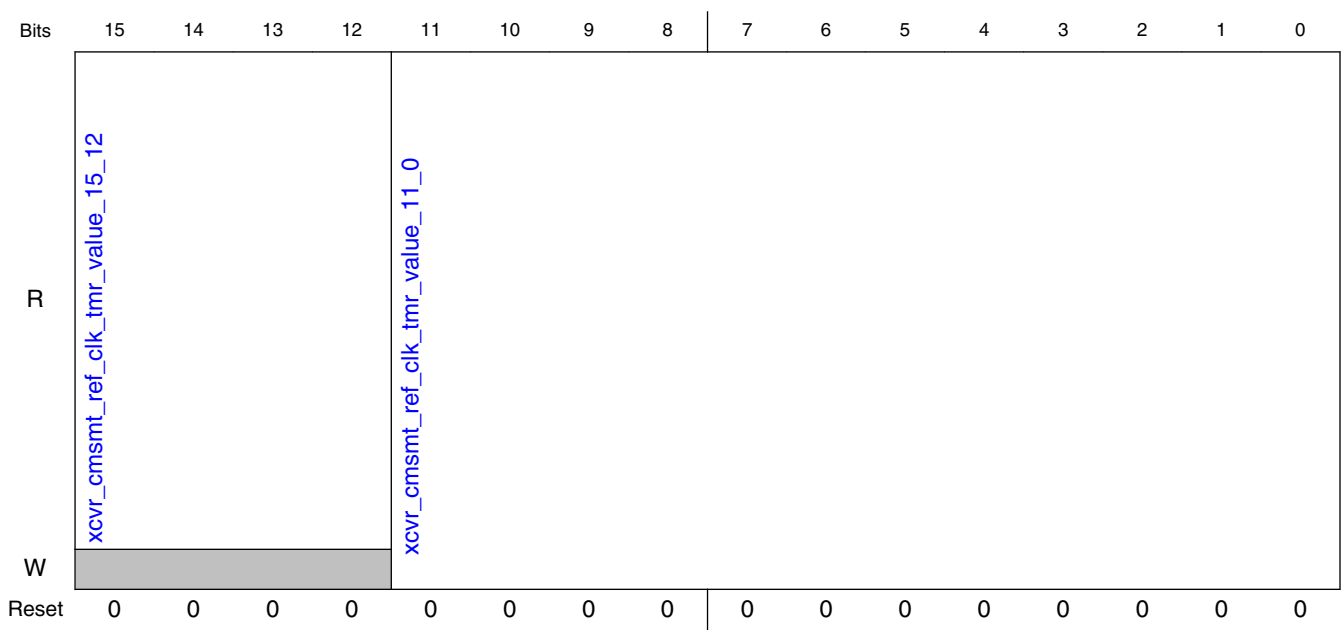
Field	Function
15-3 xcvr_cmsmt_test_clk_sel_15_3	Reserved
2-0 xcvr_cmsmt_test_clk_sel_2_0	Test clock select: This field drives the

13.4.10.2.175 Reference clock timer value register (lane0_xcvr_cmsmt_ref_clk_tmr_value - lane3_xcvr_cmsmt_ref_clk_tmr_value)

13.4.10.2.175.1 Offset

Register	Offset
lane0_xcvr_cmsmt_ref_clk_tmr_value	80B6h
lane1_xcvr_cmsmt_ref_clk_tmr_value	84B6h
lane2_xcvr_cmsmt_ref_clk_tmr_value	88B6h
lane3_xcvr_cmsmt_ref_clk_tmr_value	8CB6h

13.4.10.2.175.2 Diagram



13.4.10.2.175.3 Fields

Field	Function
15-12 xcvr_cmsmt_ref_clk_tmr_value_15_12	Reserved

Table continues on the next page...

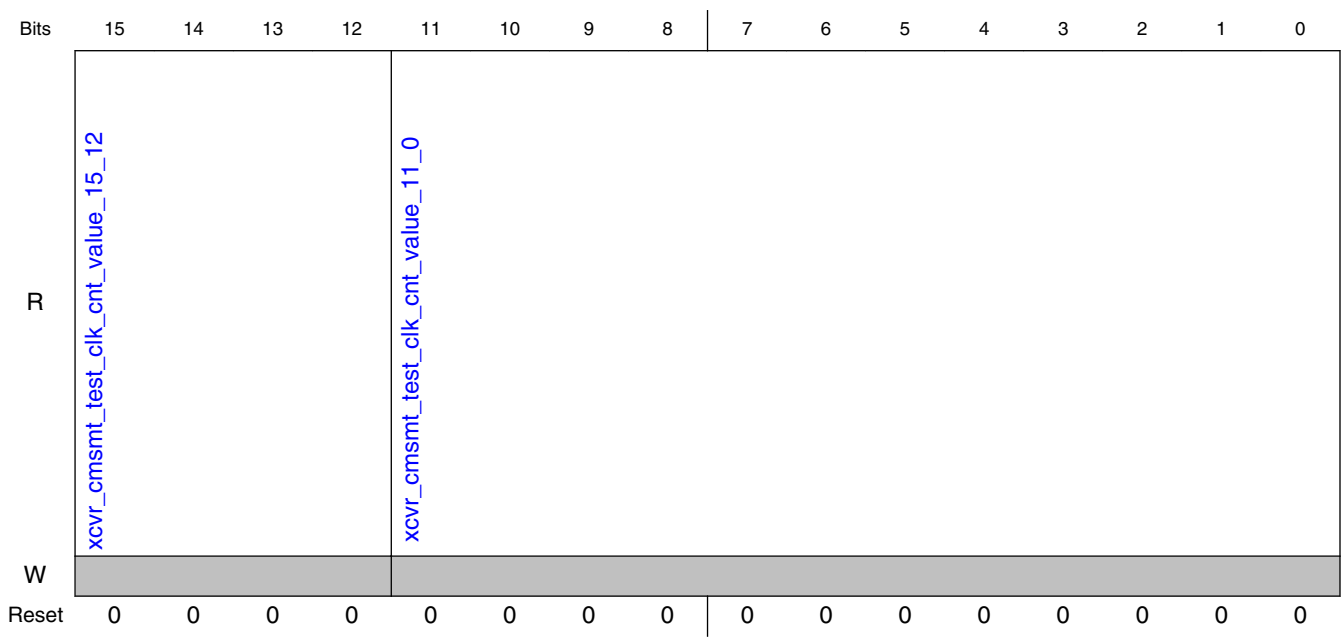
Field	Function
11-0 xcvr_cmsmt_ref_clk_tmr_value_11_0	Reference clock timer value : This specifies the amount of time, in reference clock cycles, to count test clock cycles. This value minus 1 is loaded into the reference clock timer. A value of 0 for this field is not valid when running this function.

13.4.10.2.176 Test clock counter value register (lane0_xcvr_cmsmt_test_clk_cnt_value - lane3_xcvr_cmsmt_test_clk_cnt_value)

13.4.10.2.176.1 Offset

Register	Offset
lane0_xcvr_cmsmt_test_clk_cnt_value	80B7h
lane1_xcvr_cmsmt_test_clk_cnt_value	84B7h
lane2_xcvr_cmsmt_test_clk_cnt_value	88B7h
lane3_xcvr_cmsmt_test_clk_cnt_value	8CB7h

13.4.10.2.176.2 Diagram



13.4.10.2.176.3 Fields

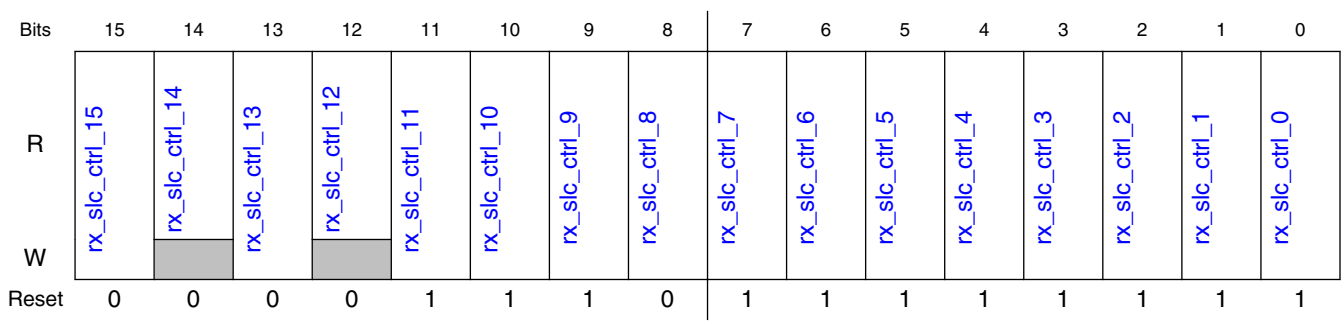
Field	Function
15-12 xcvr_cmsmt_test_clk_cnt_value_15_12	Reserved
11-0 xcvr_cmsmt_test_clk_cnt_value_11_0	Test clock counter value: When the test clock measurement process is complete, the value in this field specifies the number of test clock cycles that were counted in the time specified by the reference clock timer value. This field is only valid while the test clock measurement done bit in the

13.4.10.2.177 RX sampler latch calibration control register (lane0_rx_slc_ctrl - lane3_rx_slc_ctrl)

13.4.10.2.177.1 Offset

Register	Offset
lane0_rx_slc_ctrl	80E0h
lane1_rx_slc_ctrl	84E0h
lane2_rx_slc_ctrl	88E0h
lane3_rx_slc_ctrl	8CE0h

13.4.10.2.177.2 Diagram



13.4.10.2.177.3 Fields

Field	Function
15 rx_slc_ctrl_15	Start RX sampler latch calibration: Activating (1b1) this bit will start the RX sampler latch calibration process. This bit should remain active until the RX sampler latch calibration process is complete. To start

Table continues on the next page...

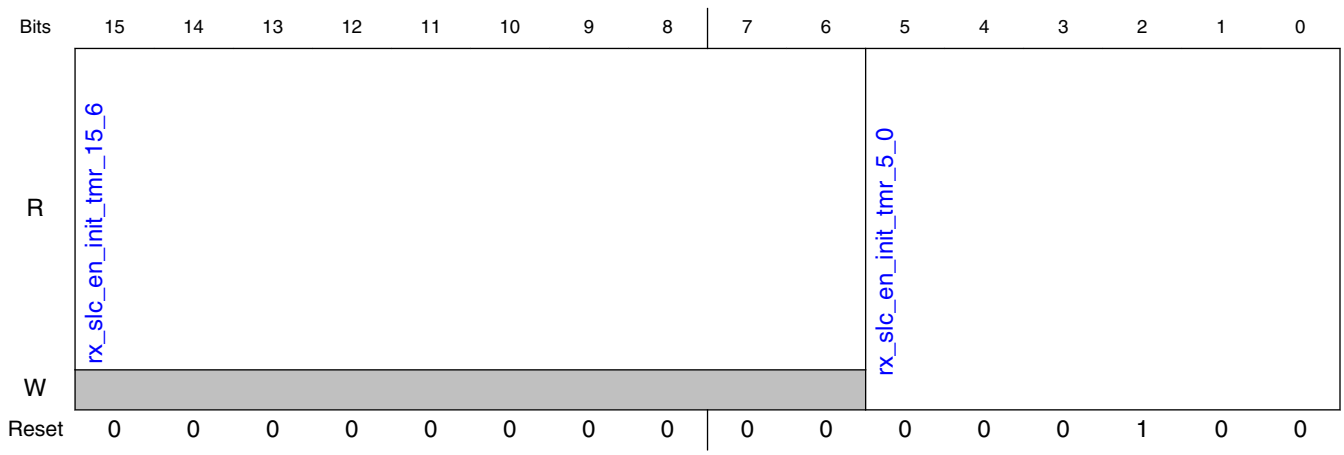
Field	Function
	another RX sampler latch calibration process, this bit must first be set inactive until the RX sampler latch calibration done bit is read as inactive.
14 rx_slc_ctrl_14	RX sampler latch calibration done: This bit will be set to 1b1 when the RX sampler latch calibration process is complete. It will be cleared by
13 rx_slc_ctrl_13	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
12 rx_slc_ctrl_12	Reserved
11 rx_slc_ctrl_11	I odd positive calibration unit enable.
10 rx_slc_ctrl_10	Q odd positive calibration unit enable.
9 rx_slc_ctrl_9	E odd positive calibration unit enable.
8 rx_slc_ctrl_8	I odd negative calibration unit enable.
7 rx_slc_ctrl_7	Q odd negative calibration unit enable.
6 rx_slc_ctrl_6	E odd negative calibration unit enable.
5 rx_slc_ctrl_5	I even positive calibration unit enable.
4 rx_slc_ctrl_4	Q even positive calibration unit enable.
3 rx_slc_ctrl_3	E even positive calibration unit enable.
2 rx_slc_ctrl_2	I even negative calibration unit enable.
1 rx_slc_ctrl_1	Q even negative calibration unit enable.
0 rx_slc_ctrl_0	E even negative calibration unit enable.

13.4.10.2.178 RX sampler latch calibration enable initialization timer value register (lane0_rx_slc_en_init_tmr - lane3_rx_slc_en_init_tmr)

13.4.10.2.178.1 Offset

Register	Offset
lane0_rx_slc_en_init_tmr	80E1h
lane1_rx_slc_en_init_tmr	84E1h
lane2_rx_slc_en_init_tmr	88E1h
lane3_rx_slc_en_init_tmr	8CE1h

13.4.10.2.178.2 Diagram



13.4.10.2.178.3 Fields

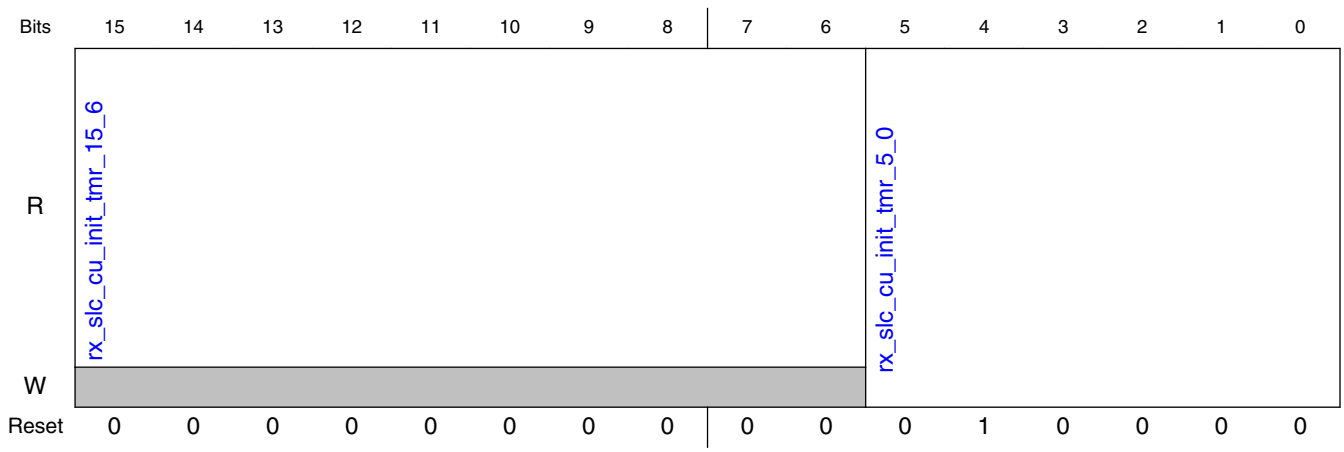
Field	Function
15-6 rx_slc_en_init_tmr_15_6	Reserved
5-0 rx_slc_en_init_tmr_5_0	RX sampler latch calibration enable initialization timer value : This is the value that is used for the RX sampler latch calibration enable initialization timer, which controls the time the

13.4.10.2.179 RX sampler latch calibration unit initialization timer value register (lane0_rx_slc_cu_init_tmr - lane3_rx_slc_cu_init_tmr)

13.4.10.2.179.1 Offset

Register	Offset
lane0_rx_slc_cu_init_tmr	80E2h
lane1_rx_slc_cu_init_tmr	84E2h
lane2_rx_slc_cu_init_tmr	88E2h
lane3_rx_slc_cu_init_tmr	8CE2h

13.4.10.2.179.2 Diagram



13.4.10.2.179.3 Fields

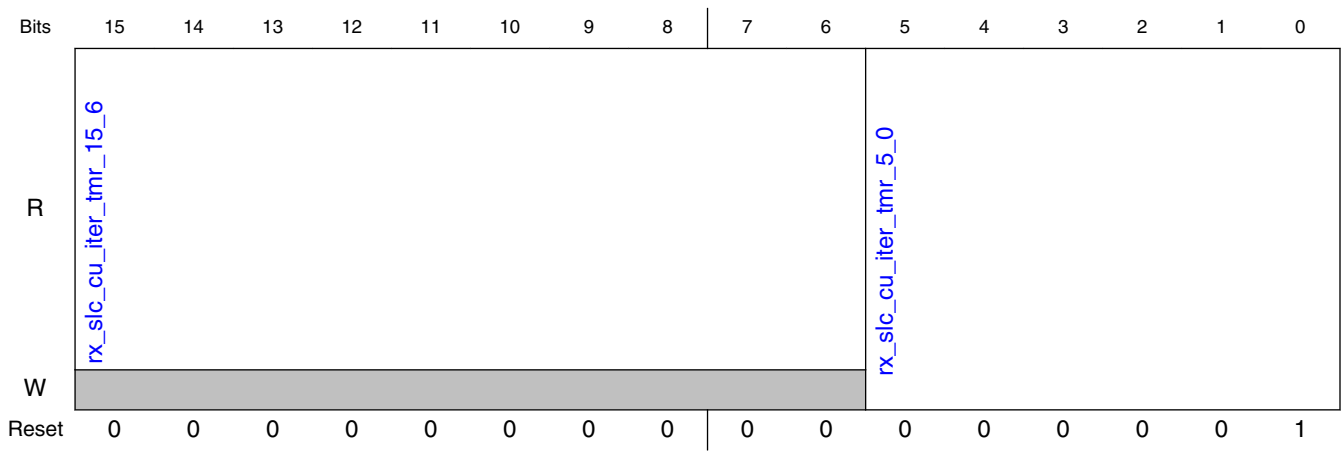
Field	Function
15-6 rx_slc_cu_init_tmr_15_6	Reserved
5-0 rx_slc_cu_init_tmr_5_0	RX sampler latch calibration unit initialization timer value : This is the value that is used for each of the calibration units initialization timers.

13.4.10.2.180 RX sampler latch calibration unit iteration timer value register (lane0_rx_slc_cu_iter_tmr - lane3_rx_slc_cu_iter_tmr)

13.4.10.2.180.1 Offset

Register	Offset
lane0_rx_slc_cu_iter_tmr	80E3h
lane1_rx_slc_cu_iter_tmr	84E3h
lane2_rx_slc_cu_iter_tmr	88E3h
lane3_rx_slc_cu_iter_tmr	8CE3h

13.4.10.2.180.2 Diagram



13.4.10.2.180.3 Fields

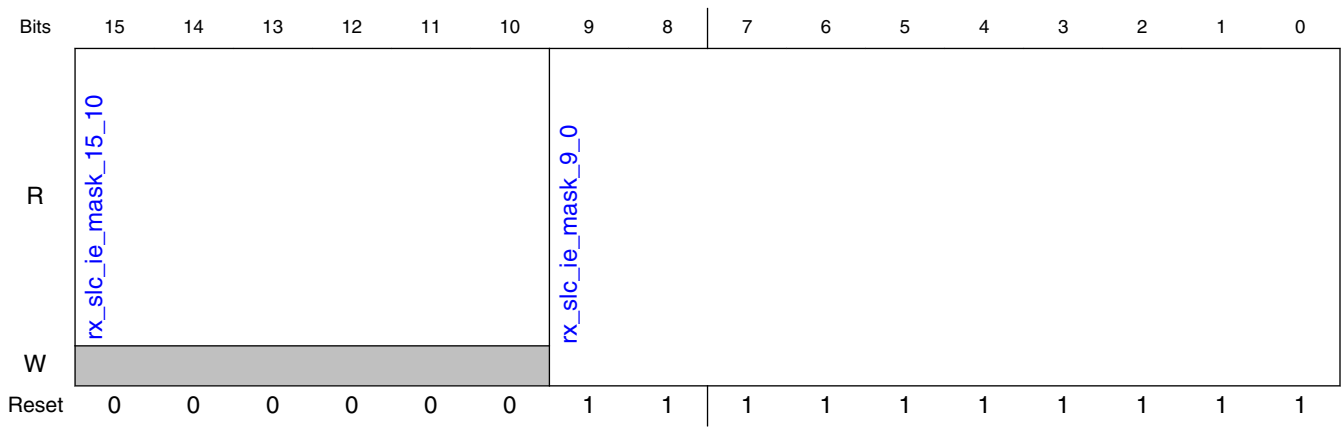
Field	Function
15-6 rx_slc_cu_iter_tmr_15_6	Reserved
5-0 rx_slc_cu_iter_tmr_5_0	RX sampler latch calibration unit iteration timer value : This is the value that is used for each of the calibration units iteration timers.

13.4.10.2.181 RX sampler latch calibration I even data mask register (lane 0_rx_slc_ie_mask - lane3_rx_slc_ie_mask)

13.4.10.2.181.1 Offset

Register	Offset
lane0_rx_slc_ie_mask	80E4h
lane1_rx_slc_ie_mask	84E4h
lane2_rx_slc_ie_mask	88E4h
lane3_rx_slc_ie_mask	8CE4h

13.4.10.2.181.2 Diagram



13.4.10.2.181.3 Fields

Field	Function
15-10 rx_slc_ie_mask_15_10	Reserved
9-0 rx_slc_ie_mask_9_0	I even data mask: The bits in this field can be used to mask the I data even bits in the data unit component.

13.4.10.2.182 RX sampler latch calibration I odd data mask register (lane 0_rx_slc_io_mask - lane3_rx_slc_io_mask)

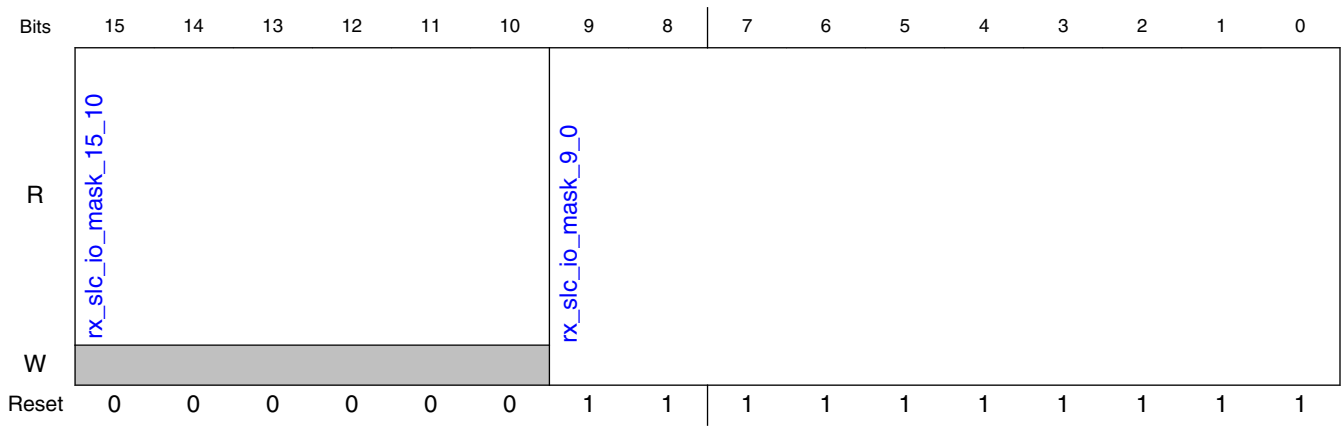
13.4.10.2.182.1 Offset

Register	Offset
lane0_rx_slc_io_mask	80E5h

Table continues on the next page...

Register	Offset
lane1_rx_slc_io_mask	84E5h
lane2_rx_slc_io_mask	88E5h
lane3_rx_slc_io_mask	8CE5h

13.4.10.2.182.2 Diagram



13.4.10.2.182.3 Fields

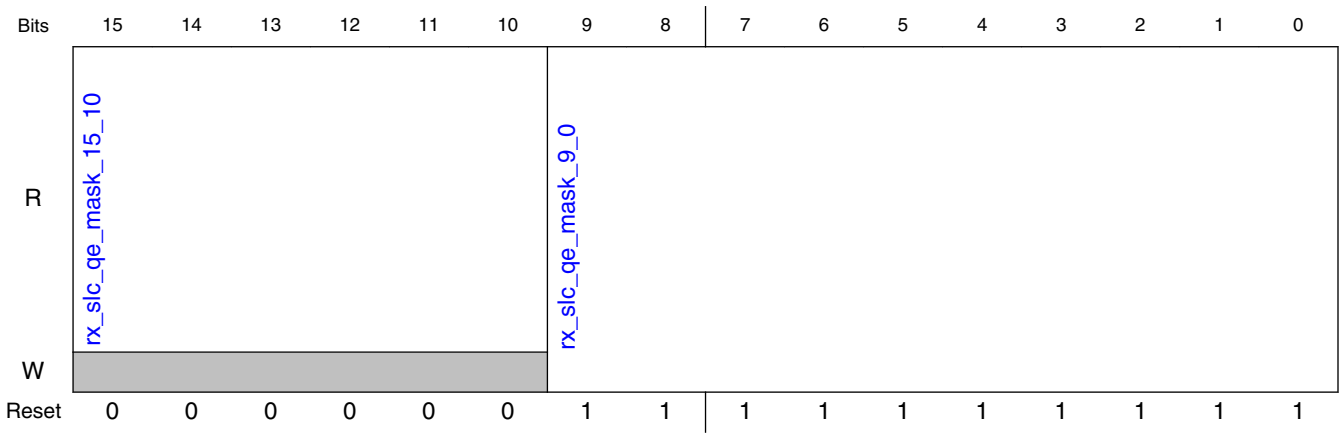
Field	Function
15-10 rx_slc_io_mask_15_10	Reserved
9-0 rx_slc_io_mask_9_0	I odd data mask: The bits in this field can be used to mask the I data odd bits in the data unit component.

13.4.10.2.183 RX sampler latch calibration Q even data mask register (lane 0_rx_slc_qe_mask - lane3_rx_slc_qe_mask)

13.4.10.2.183.1 Offset

Register	Offset
lane0_rx_slc_qe_mask	80E6h
lane1_rx_slc_qe_mask	84E6h
lane2_rx_slc_qe_mask	88E6h
lane3_rx_slc_qe_mask	8CE6h

13.4.10.2.183.2 Diagram



13.4.10.2.183.3 Fields

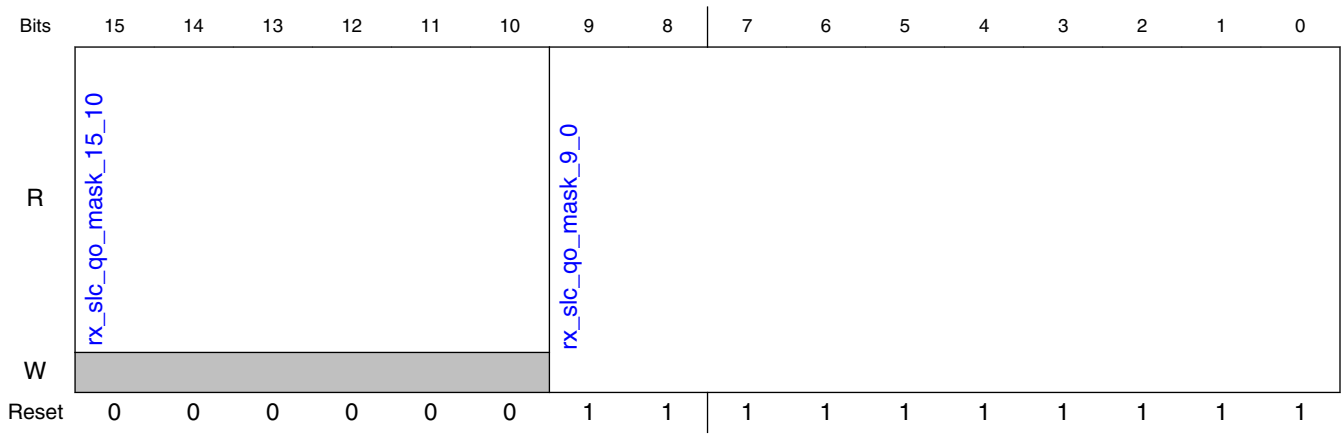
Field	Function
15-10 rx_slc_qe_mask_15_10	Reserved
9-0 rx_slc_qe_mask_9_0	Q even data mask: The bits in this field can be used to mask the Q data even bits in the data unit component.

13.4.10.2.184 RX sampler latch calibration Q odd data mask register (lane 0_rx_slc_qo_mask - lane3_rx_slc_qo_mask)

13.4.10.2.184.1 Offset

Register	Offset
lane0_rx_slc_qo_mask	80E7h
lane1_rx_slc_qo_mask	84E7h
lane2_rx_slc_qo_mask	88E7h
lane3_rx_slc_qo_mask	8CE7h

13.4.10.2.184.2 Diagram



13.4.10.2.184.3 Fields

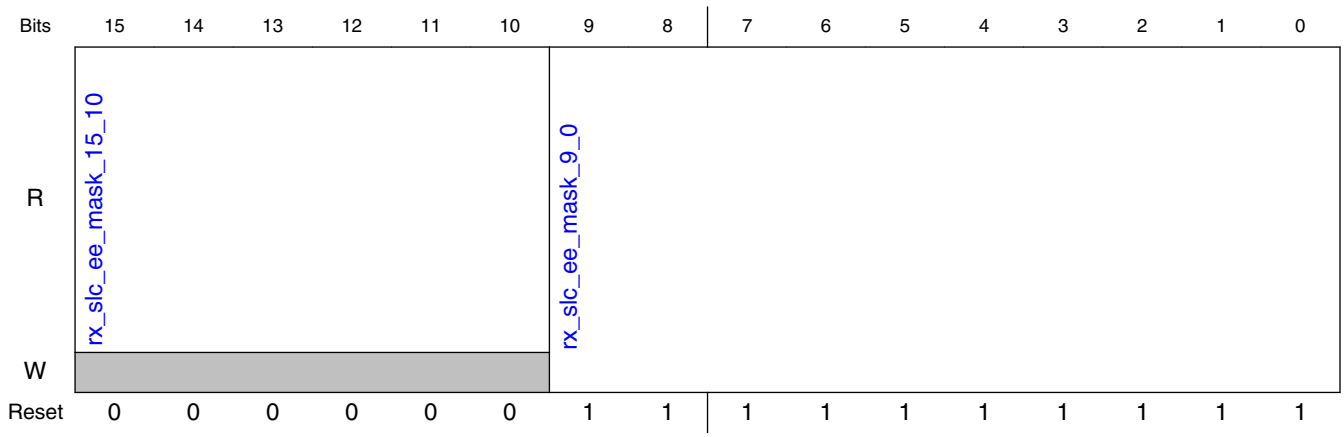
Field	Function
15-10 rx_slc_qo_mask_15_10	Reserved
9-0 rx_slc_qo_mask_9_0	Q odd data mask: The bits in this field can be used to mask the Q data odd bits in the data unit component.

13.4.10.2.185 RX sampler latch calibration E even data mask register (lane 0_rx_slc_ee_mask - lane3_rx_slc_ee_mask)

13.4.10.2.185.1 Offset

Register	Offset
lane0_rx_slc_ee_mask	80E8h
lane1_rx_slc_ee_mask	84E8h
lane2_rx_slc_ee_mask	88E8h
lane3_rx_slc_ee_mask	8CE8h

13.4.10.2.185.2 Diagram



13.4.10.2.185.3 Fields

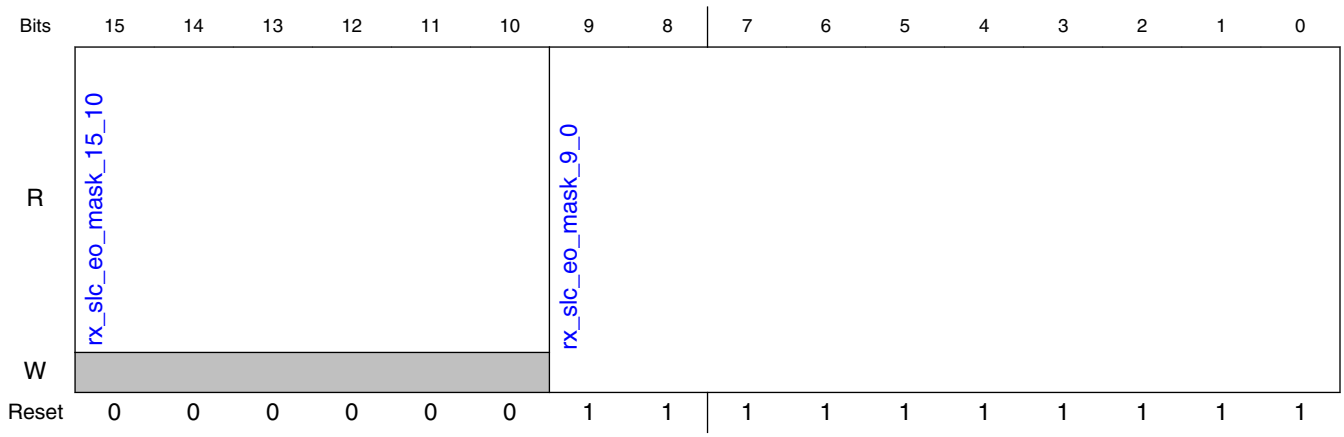
Field	Function
15-10 rx_slc_ee_mask_15_10	Reserved
9-0 rx_slc_ee_mask_9_0	E even data mask: The bits in this field can be used to mask the E data even bits in the data unit component.

13.4.10.2.186 RX sampler latch calibration E odd data mask register (lane 0_rx_slc_eo_mask - lane3_rx_slc_eo_mask)

13.4.10.2.186.1 Offset

Register	Offset
lane0_rx_slc_eo_mask	80E9h
lane1_rx_slc_eo_mask	84E9h
lane2_rx_slc_eo_mask	88E9h
lane3_rx_slc_eo_mask	8CE9h

13.4.10.2.186.2 Diagram



13.4.10.2.186.3 Fields

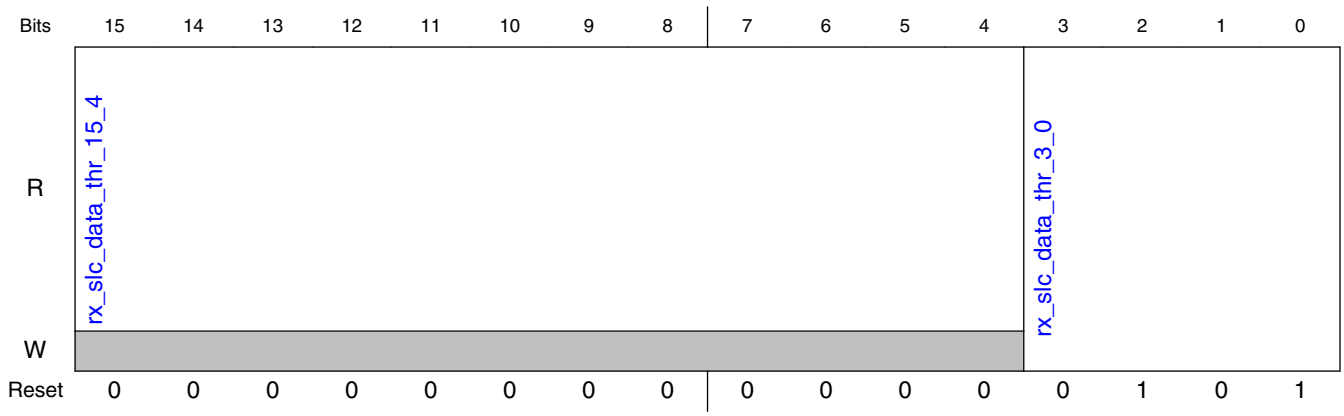
Field	Function
15-10 rx_slc_eo_mask_15_10	Reserved
9-0 rx_slc_eo_mask_9_0	E odd data mask: The bits in this field can be used to mask the E data odd bits in the data unit component.

13.4.10.2.187 RX sampler latch calibration data threshold register (lane0_rx_slc_data_thr - lane3_rx_slc_data_thr)

13.4.10.2.187.1 Offset

Register	Offset
lane0_rx_slc_data_thr	80EAh
lane1_rx_slc_data_thr	84EAh
lane2_rx_slc_data_thr	88EAh
lane3_rx_slc_data_thr	8CEAh

13.4.10.2.187.2 Diagram



13.4.10.2.187.3 Fields

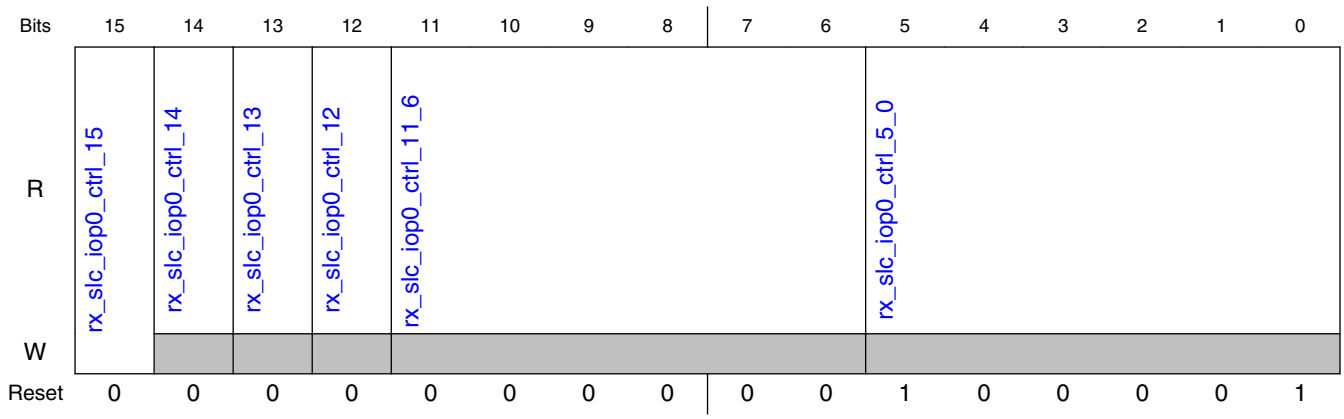
Field	Function
15-4 rx_slc_data_thr_15_4	Reserved
3-0 rx_slc_data_thr_3_0	Data threshold value: Specifies the value that the RX sampler latch calibration data unit compares the sum of 1s for each parallel data input against to create the inputs to each of the calibration units. The range of values for this field is as follows.

13.4.10.2.188 RX sampler latch I odd positive 0 calibration unit control register (lane0_rx_slc_iop0_ctrl - lane3_rx_slc_iop0_ctrl)

13.4.10.2.188.1 Offset

Register	Offset
lane0_rx_slc_iop0_ctrl	8100h
lane1_rx_slc_iop0_ctrl	8500h
lane2_rx_slc_iop0_ctrl	8900h
lane3_rx_slc_iop0_ctrl	8D00h

13.4.10.2.188.2 Diagram



13.4.10.2.188.3 Fields

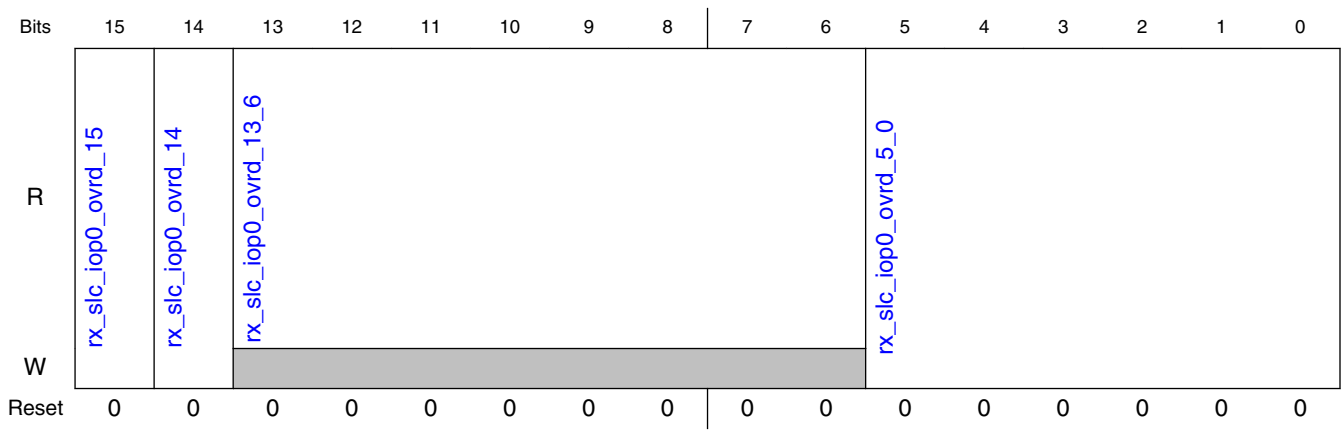
Field	Function
15 rx_slc_iop0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_iop0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_iop0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_iop0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_iop0_ctrl_11_6	Reserved
5-0 rx_slc_iop0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.189 RX sampler latch I odd positive 0 calibration unit override register (lane0_rx_slc_iop0_ovrd - lane3_rx_slc_iop0_ovrd)

13.4.10.2.189.1 Offset

Register	Offset
lane0_rx_slc_iop0_ovrd	8101h
lane1_rx_slc_iop0_ovrd	8501h
lane2_rx_slc_iop0_ovrd	8901h
lane3_rx_slc_iop0_ovrd	8D01h

13.4.10.2.189.2 Diagram



13.4.10.2.189.3 Fields

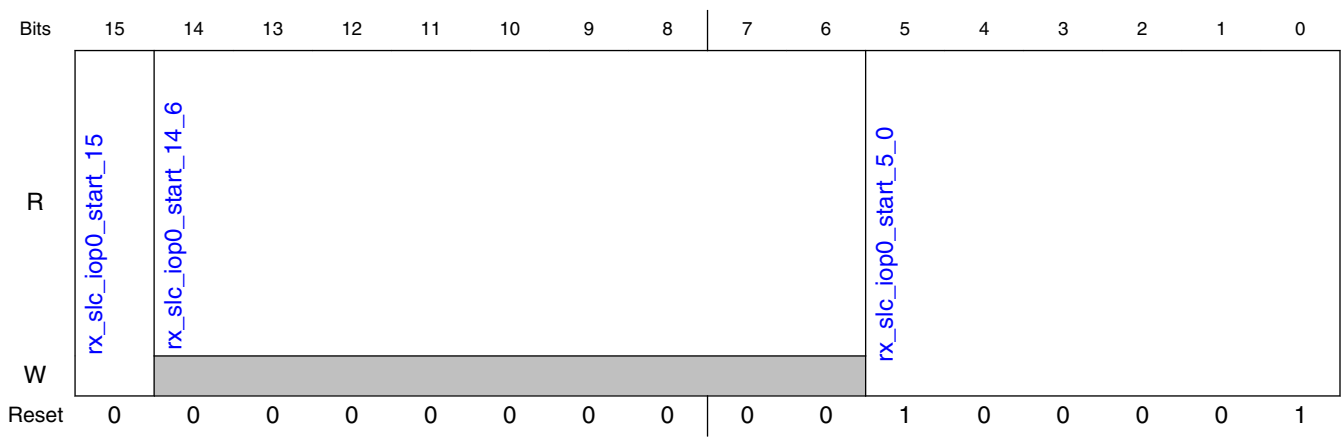
Field	Function
15 rx_slc_iop0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_iop0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_iop0_ovrd_13_6	Reserved
5-0 rx_slc_iop0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.190 RX sampler latch I odd positive 0 calibration unit start register (lane0_rx_slc_iop0_start - lane3_rx_slc_iop0_start)

13.4.10.2.190.1 Offset

Register	Offset
lane0_rx_slc_iop0_start	8102h
lane1_rx_slc_iop0_start	8502h
lane2_rx_slc_iop0_start	8902h
lane3_rx_slc_iop0_start	8D02h

13.4.10.2.190.2 Diagram



13.4.10.2.190.3 Fields

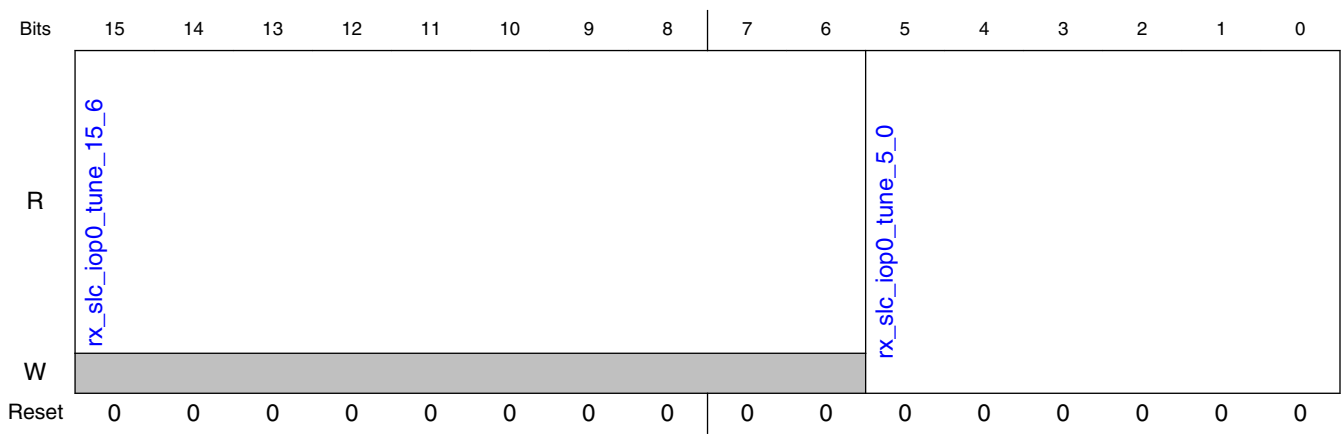
Field	Function
15 rx_slc_iop0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_iop0_start_14_6	Reserved
5-0 rx_slc_iop0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.191 RX sampler latch I odd positive 0 calibration unit tune register (lane0_rx_slc_iop0_tune - lane3_rx_slc_iop0_tune)

13.4.10.2.191.1 Offset

Register	Offset
lane0_rx_slc_iop0_tune	8103h
lane1_rx_slc_iop0_tune	8503h
lane2_rx_slc_iop0_tune	8903h
lane3_rx_slc_iop0_tune	8D03h

13.4.10.2.191.2 Diagram



13.4.10.2.191.3 Fields

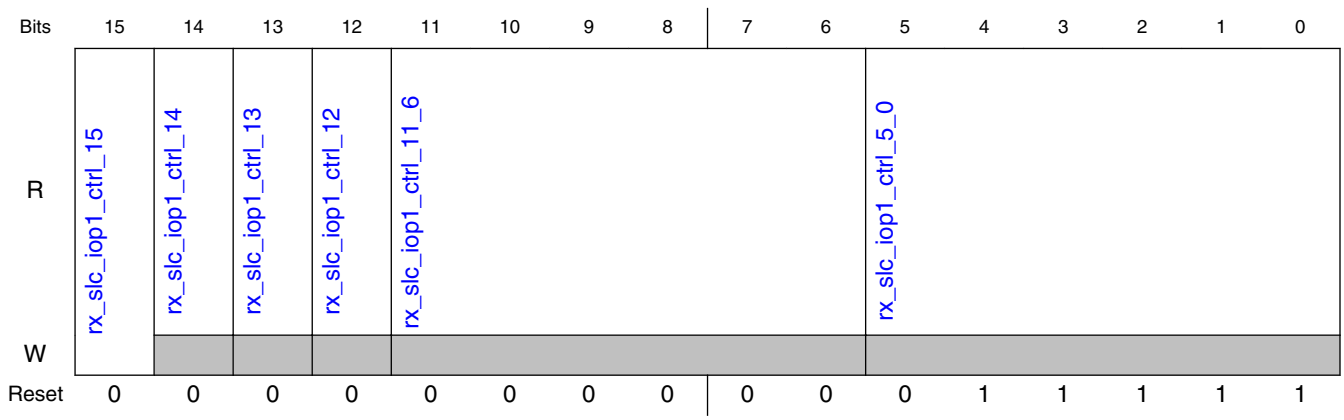
Field	Function
15-6 rx_slc_iop0_tune_15_6	Reserved
5-0 rx_slc_iop0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.192 RX sampler latch I odd positive 1 calibration unit control register (lane0_rx_slc_iop1_ctrl - lane3_rx_slc_iop1_ctrl)

13.4.10.2.192.1 Offset

Register	Offset
lane0_rx_slc_iop1_ctrl	8104h
lane1_rx_slc_iop1_ctrl	8504h
lane2_rx_slc_iop1_ctrl	8904h
lane3_rx_slc_iop1_ctrl	8D04h

13.4.10.2.192.2 Diagram



13.4.10.2.192.3 Fields

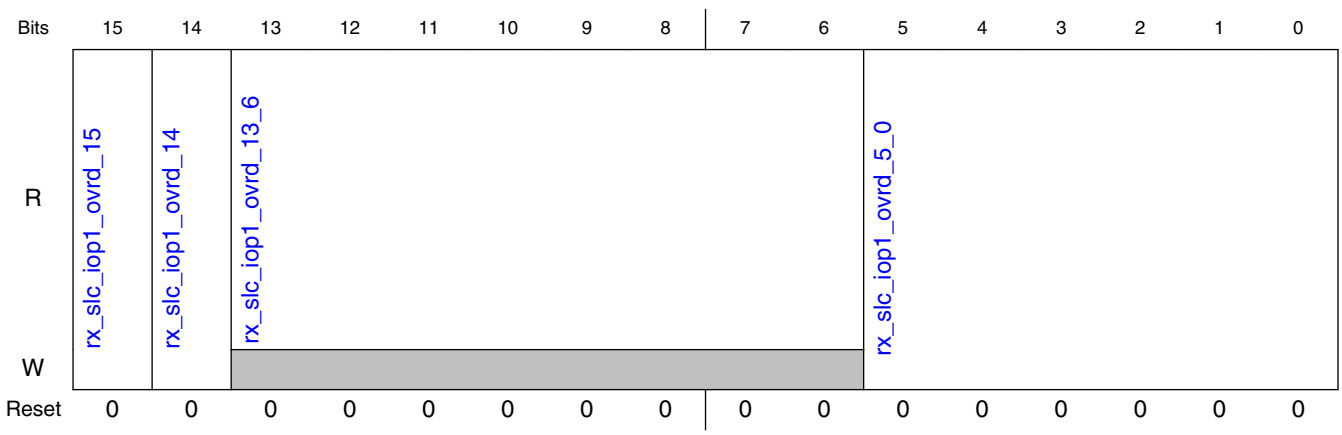
Field	Function
15 rx_slc_iop1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_iop1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_iop1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_iop1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_iop1_ctrl_11_6	Reserved
5-0 rx_slc_iop1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.193 RX sampler latch I odd positive 1 calibration unit override register (lane0_rx_slc_iop1_ovrd - lane3_rx_slc_iop1_ovrd)

13.4.10.2.193.1 Offset

Register	Offset
lane0_rx_slc_iop1_ovrd	8105h
lane1_rx_slc_iop1_ovrd	8505h
lane2_rx_slc_iop1_ovrd	8905h
lane3_rx_slc_iop1_ovrd	8D05h

13.4.10.2.193.2 Diagram



13.4.10.2.193.3 Fields

Field	Function
15 rx_slc_iop1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_iop1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_iop1_ovrd_13_6	Reserved
5-0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the

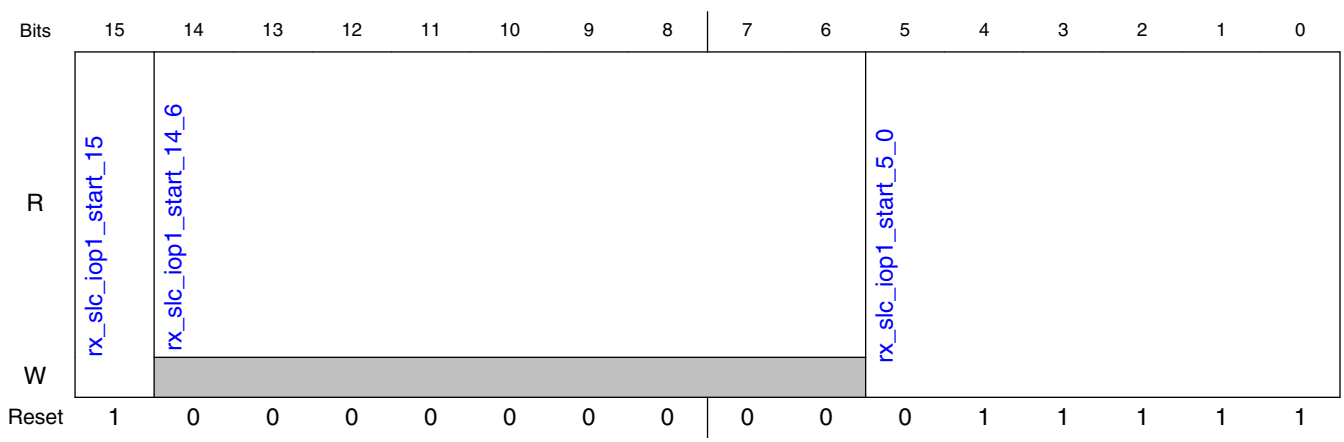
Field	Function
rx_slc_iop1_ovrd_5_0	calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.194 RX sampler latch I odd positive 1 calibration unit start register (lane0_rx_slc_iop1_start - lane3_rx_slc_iop1_start)

13.4.10.2.194.1 Offset

Register	Offset
lane0_rx_slc_iop1_start	8106h
lane1_rx_slc_iop1_start	8506h
lane2_rx_slc_iop1_start	8906h
lane3_rx_slc_iop1_start	8D06h

13.4.10.2.194.2 Diagram



13.4.10.2.194.3 Fields

Field	Function
15 rx_slc_iop1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_iop1_start_14_6	Reserved

Table continues on the next page...

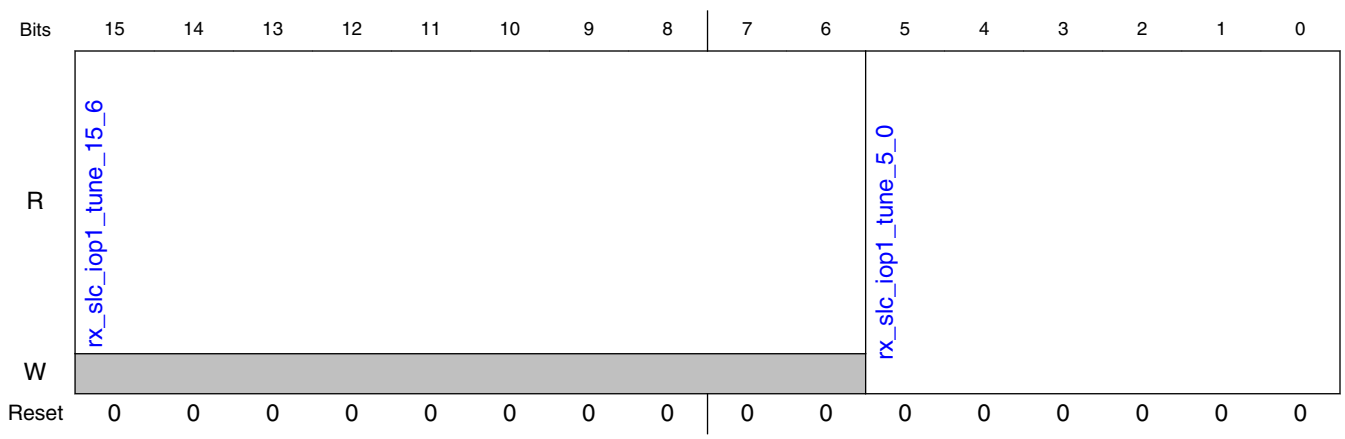
Field	Function
5-0 rx_slc_iop1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.195 RX sampler latch I odd positive 1 calibration unit tune register (lane0_rx_slc_iop1_tune - lane3_rx_slc_iop1_tune)

13.4.10.2.195.1 Offset

Register	Offset
lane0_rx_slc_iop1_tune	8107h
lane1_rx_slc_iop1_tune	8507h
lane2_rx_slc_iop1_tune	8907h
lane3_rx_slc_iop1_tune	8D07h

13.4.10.2.195.2 Diagram



13.4.10.2.195.3 Fields

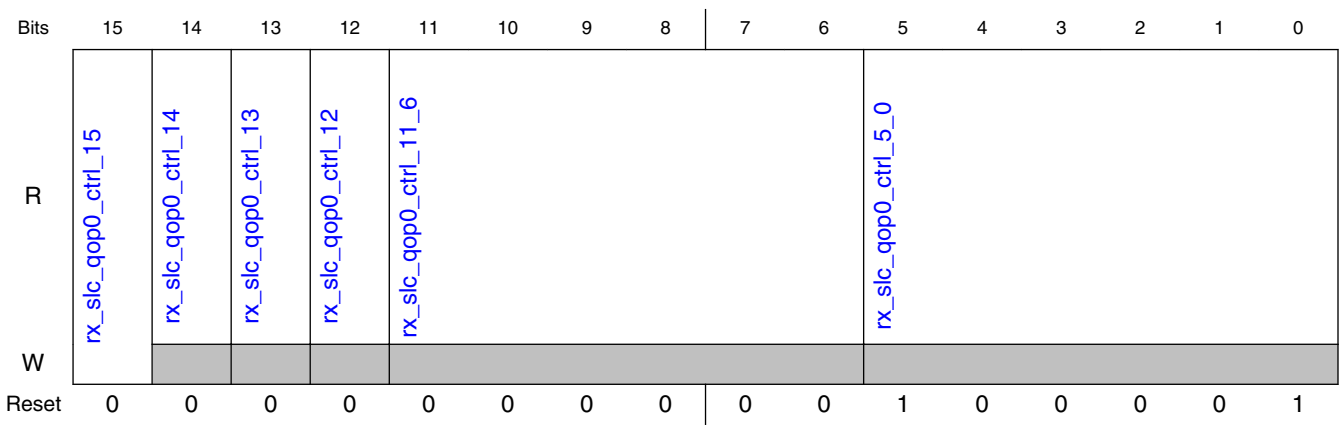
Field	Function
15-6 rx_slc_iop1_tune_15_6	Reserved
5-0 rx_slc_iop1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.196 RX sampler latch Q odd positive 0 calibration unit control register (lane0_rx_slc_qop0_ctrl - lane3_rx_slc_qop0_ctrl)

13.4.10.2.196.1 Offset

Register	Offset
lane0_rx_slc_qop0_ctrl	8108h
lane1_rx_slc_qop0_ctrl	8508h
lane2_rx_slc_qop0_ctrl	8908h
lane3_rx_slc_qop0_ctrl	8D08h

13.4.10.2.196.2 Diagram



13.4.10.2.196.3 Fields

Field	Function
15 rx_slc_qop0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qop0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qop0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.

Table continues on the next page...

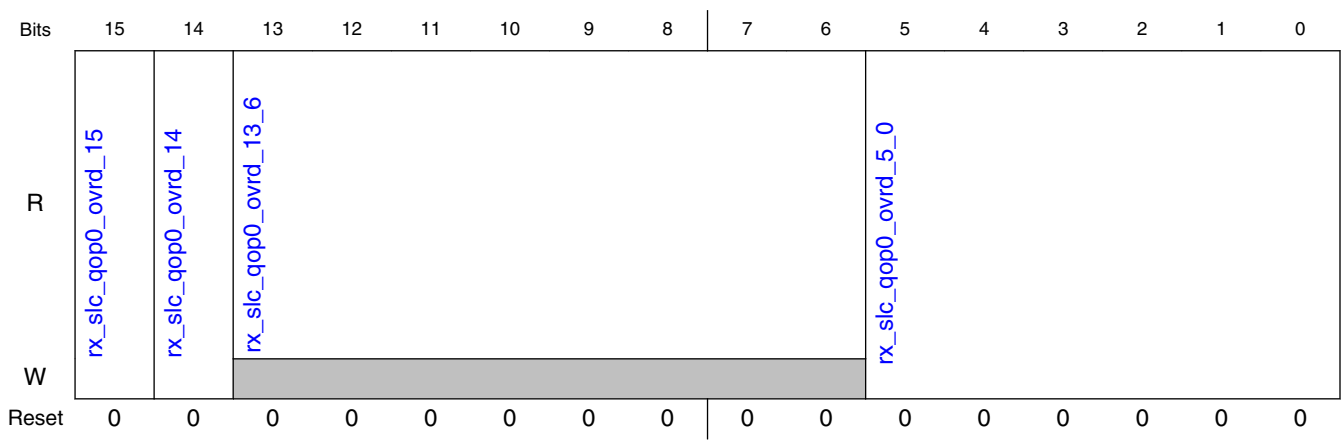
Field	Function
rx_slc_qop0_ctrl_12	
11-6 rx_slc_qop0_ctrl_11_6	Reserved
5-0 rx_slc_qop0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.197 RX sampler latch Q odd positive 0 calibration unit override register (lane0_rx_slc_qop0_ovrd - lane3_rx_slc_qop0_ovrd)

13.4.10.2.197.1 Offset

Register	Offset
lane0_rx_slc_qop0_ovrd	8109h
lane1_rx_slc_qop0_ovrd	8509h
lane2_rx_slc_qop0_ovrd	8909h
lane3_rx_slc_qop0_ovrd	8D09h

13.4.10.2.197.2 Diagram



13.4.10.2.197.3 Fields

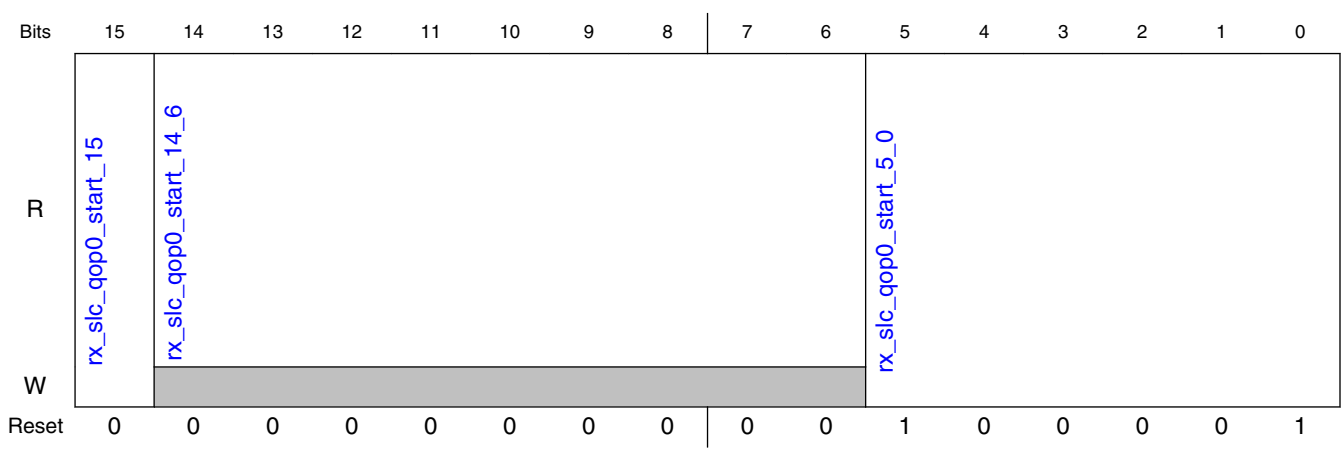
Field	Function
15 rx_slc_qop0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qop0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qop0_ovrd_13_6	Reserved
5-0 rx_slc_qop0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.198 RX sampler latch Q odd positive 0 calibration unit start register (lane0_rx_slc_qop0_start - lane3_rx_slc_qop0_start)

13.4.10.2.198.1 Offset

Register	Offset
lane0_rx_slc_qop0_start	810Ah
lane1_rx_slc_qop0_start	850Ah
lane2_rx_slc_qop0_start	890Ah
lane3_rx_slc_qop0_start	8D0Ah

13.4.10.2.198.2 Diagram



13.4.10.2.198.3 Fields

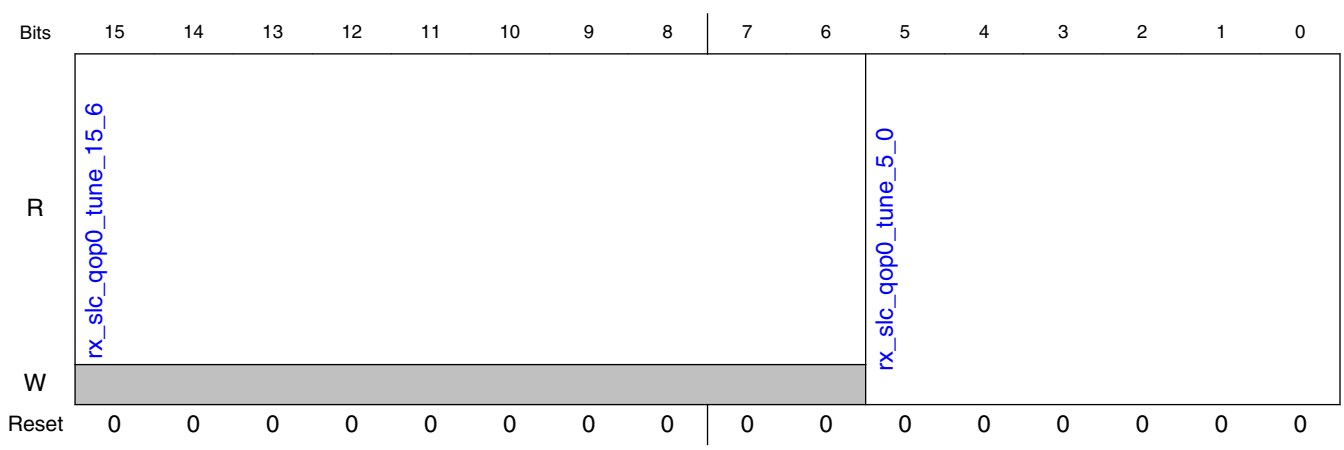
Field	Function
15 rx_slc_qop0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qop0_start_14_6	Reserved
5-0 rx_slc_qop0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.199 RX sampler latch Q odd positive 0 calibration unit tune register (lane0_rx_slc_qop0_tune - lane3_rx_slc_qop0_tune)

13.4.10.2.199.1 Offset

Register	Offset
lane0_rx_slc_qop0_tune	810Bh
lane1_rx_slc_qop0_tune	850Bh
lane2_rx_slc_qop0_tune	890Bh
lane3_rx_slc_qop0_tune	8D0Bh

13.4.10.2.199.2 Diagram



13.4.10.2.199.3 Fields

Field	Function
15-6 rx_slc_qop0_tune_15_6	Reserved
5-0 rx_slc_qop0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.200 RX sampler latch Q odd positive 1 calibration unit control register (lane0_rx_slc_qop1_ctrl - lane3_rx_slc_qop1_ctrl)

13.4.10.2.200.1 Offset

Register	Offset
lane0_rx_slc_qop1_ctrl	810Ch
lane1_rx_slc_qop1_ctrl	850Ch
lane2_rx_slc_qop1_ctrl	890Ch
lane3_rx_slc_qop1_ctrl	8D0Ch

13.4.10.2.200.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_slc_qop1_ctrl_15	rx_slc_qop1_ctrl_14	rx_slc_qop1_ctrl_13	rx_slc_qop1_ctrl_12	rx_slc_qop1_ctrl_11_6						rx_slc_qop1_ctrl_5_0					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1

13.4.10.2.200.3 Fields

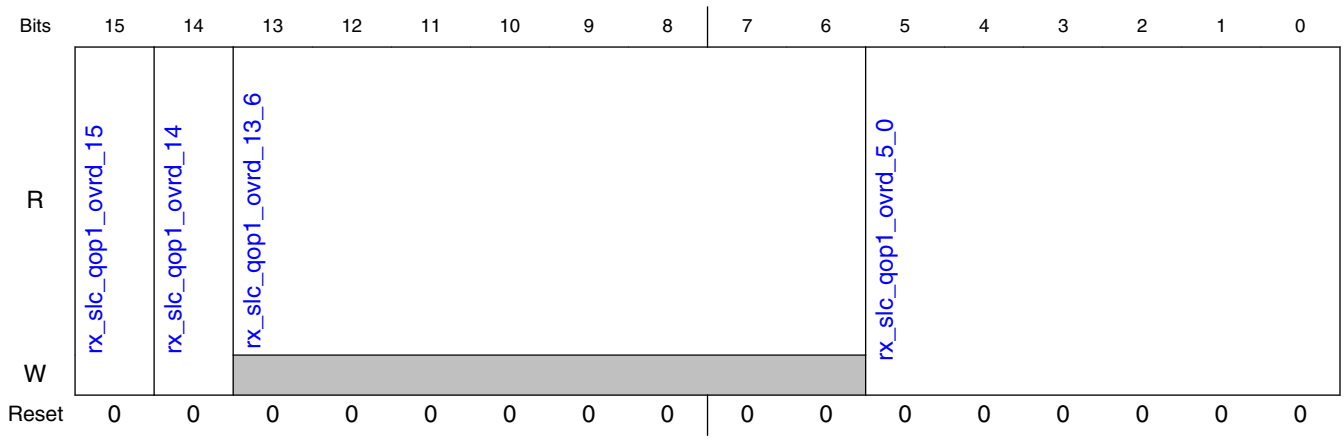
Field	Function
15 rx_slc_qop1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qop1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qop1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qop1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qop1_ctrl_11_6	Reserved
5-0 rx_slc_qop1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.201 RX sampler latch Q odd positive 1 calibration unit override register (lane0_rx_slc_qop1_ovrd - lane3_rx_slc_qop1_ovrd)

13.4.10.2.201.1 Offset

Register	Offset
lane0_rx_slc_qop1_ovrd	810Dh
lane1_rx_slc_qop1_ovrd	850Dh
lane2_rx_slc_qop1_ovrd	890Dh
lane3_rx_slc_qop1_ovrd	8D0Dh

13.4.10.2.201.2 Diagram



13.4.10.2.201.3 Fields

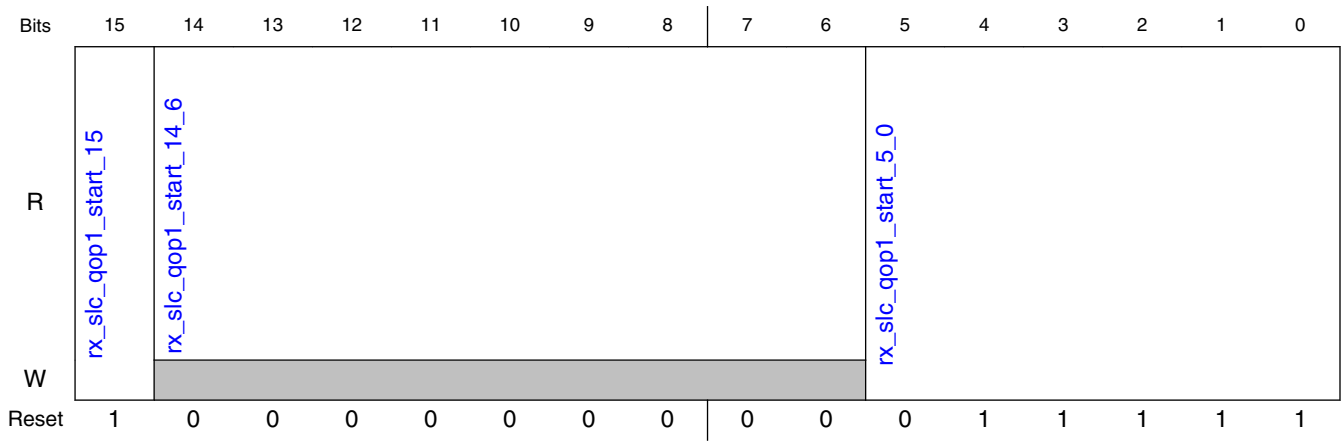
Field	Function
15 rx_slc_qop1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qop1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qop1_ovrd_13_6	Reserved
5-0 rx_slc_qop1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.202 RX sampler latch Q odd positive 1 calibration unit start register (lane0_rx_slc_qop1_start - lane3_rx_slc_qop1_start)

13.4.10.2.202.1 Offset

Register	Offset
lane0_rx_slc_qop1_start	810Eh
lane1_rx_slc_qop1_start	850Eh
lane2_rx_slc_qop1_start	890Eh
lane3_rx_slc_qop1_start	8D0Eh

13.4.10.2.202.2 Diagram



13.4.10.2.202.3 Fields

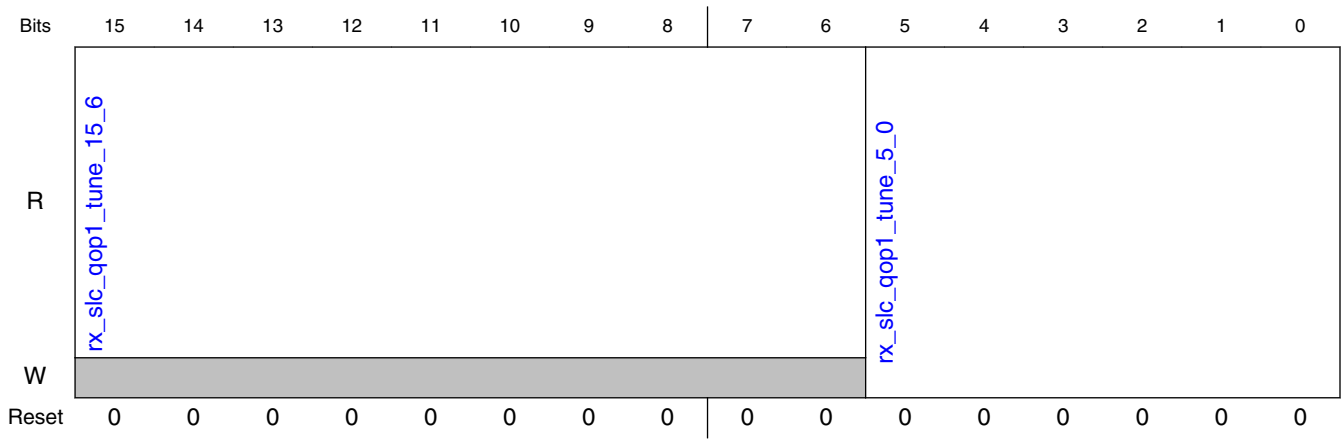
Field	Function
15 rx_slc_qop1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qop1_start_14_6	Reserved
5-0 rx_slc_qop1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.203 RX sampler latch Q odd positive 1 calibration unit tune register (lane0_rx_slc_qop1_tune - lane3_rx_slc_qop1_tune)

13.4.10.2.203.1 Offset

Register	Offset
lane0_rx_slc_qop1_tune	810Fh
lane1_rx_slc_qop1_tune	850Fh
lane2_rx_slc_qop1_tune	890Fh
lane3_rx_slc_qop1_tune	8D0Fh

13.4.10.2.203.2 Diagram



13.4.10.2.203.3 Fields

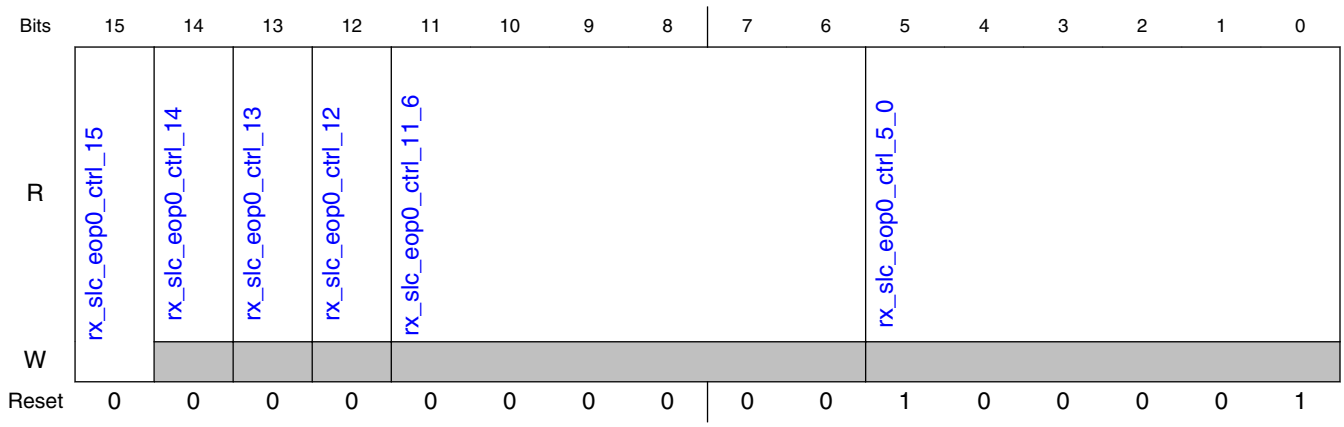
Field	Function
15-6 rx_slc_qop1_tune_15_6	Reserved
5-0 rx_slc_qop1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.204 RX sampler latch E odd positive 0 calibration unit control register (lane0_rx_slc_eop0_ctrl - lane3_rx_slc_eop0_ctrl)

13.4.10.2.204.1 Offset

Register	Offset
lane0_rx_slc_eop0_ctrl	8110h
lane1_rx_slc_eop0_ctrl	8510h
lane2_rx_slc_eop0_ctrl	8910h
lane3_rx_slc_eop0_ctrl	8D10h

13.4.10.2.204.2 Diagram



13.4.10.2.204.3 Fields

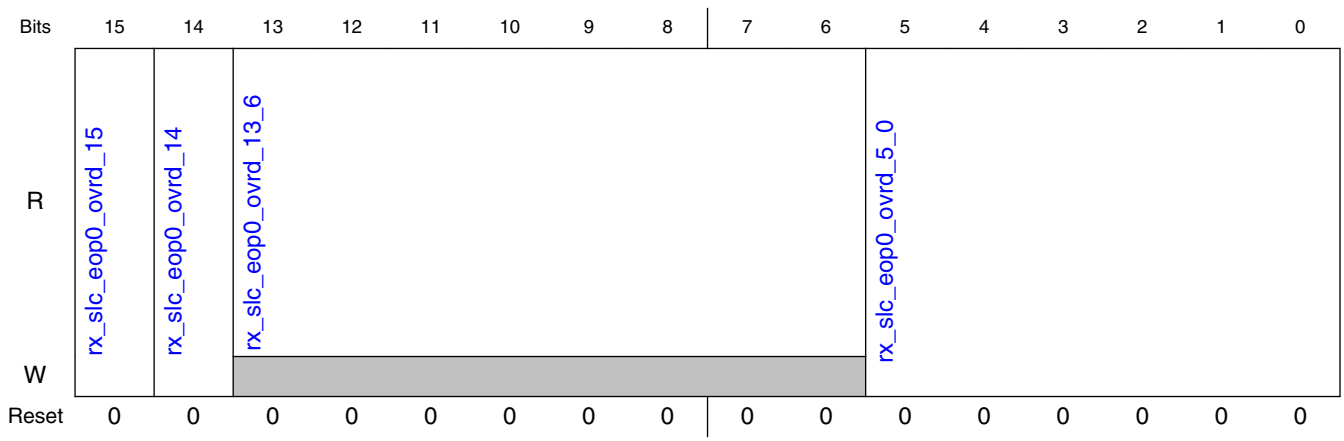
Field	Function
15 rx_slc_eop0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_eop0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_eop0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eop0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eop0_ctrl_11_6	Reserved
5-0 rx_slc_eop0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.205 RX sampler latch E odd positive 0 calibration unit override register (lane0_rx_slc_eop0_ovrd - lane3_rx_slc_eop0_ovrd)

13.4.10.2.205.1 Offset

Register	Offset
lane0_rx_slc_eop0_ovrd	8111h
lane1_rx_slc_eop0_ovrd	8511h
lane2_rx_slc_eop0_ovrd	8911h
lane3_rx_slc_eop0_ovrd	8D11h

13.4.10.2.205.2 Diagram



13.4.10.2.205.3 Fields

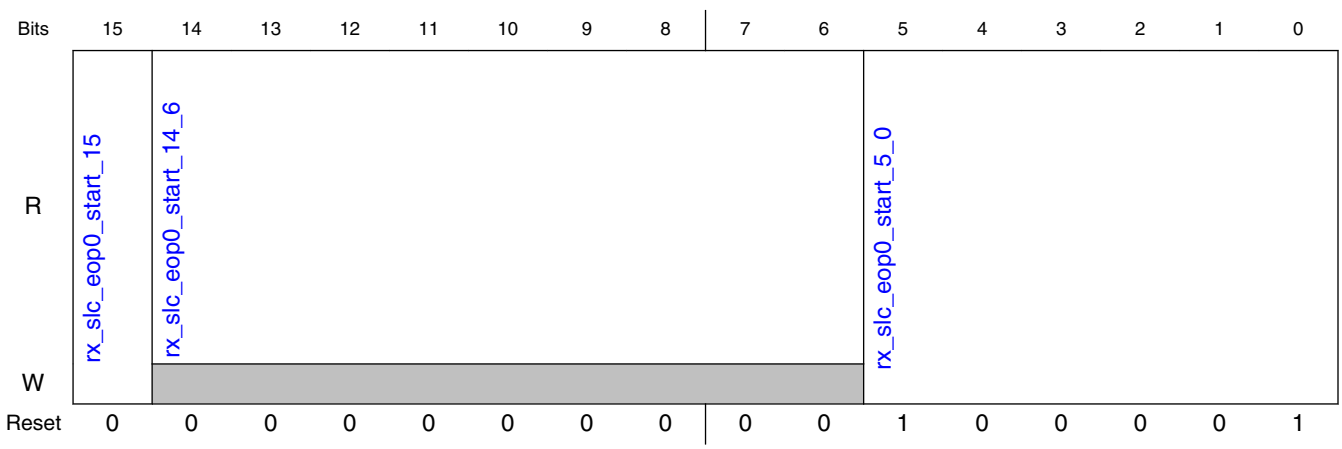
Field	Function
15 rx_slc_eop0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eop0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_eop0_ovrd_13_6	Reserved
5-0 rx_slc_eop0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.206 RX sampler latch E odd positive 0 calibration unit start register (lane0_rx_slc_eop0_start - lane3_rx_slc_eop0_start)

13.4.10.2.206.1 Offset

Register	Offset
lane0_rx_slc_eop0_start	8112h
lane1_rx_slc_eop0_start	8512h
lane2_rx_slc_eop0_start	8912h
lane3_rx_slc_eop0_start	8D12h

13.4.10.2.206.2 Diagram



13.4.10.2.206.3 Fields

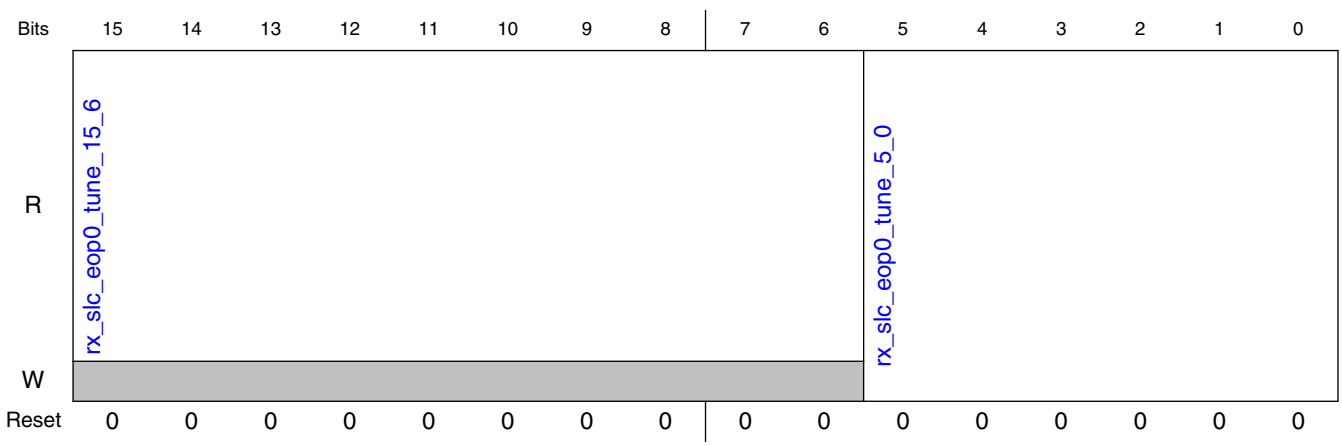
Field	Function
15 rx_slc_eop0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_eop0_start_14_6	Reserved
5-0 rx_slc_eop0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.207 RX sampler latch E odd positive 0 calibration unit tune register (lane0_rx_slc_eop0_tune - lane3_rx_slc_eop0_tune)

13.4.10.2.207.1 Offset

Register	Offset
lane0_rx_slc_eop0_tune	8113h
lane1_rx_slc_eop0_tune	8513h
lane2_rx_slc_eop0_tune	8913h
lane3_rx_slc_eop0_tune	8D13h

13.4.10.2.207.2 Diagram



13.4.10.2.207.3 Fields

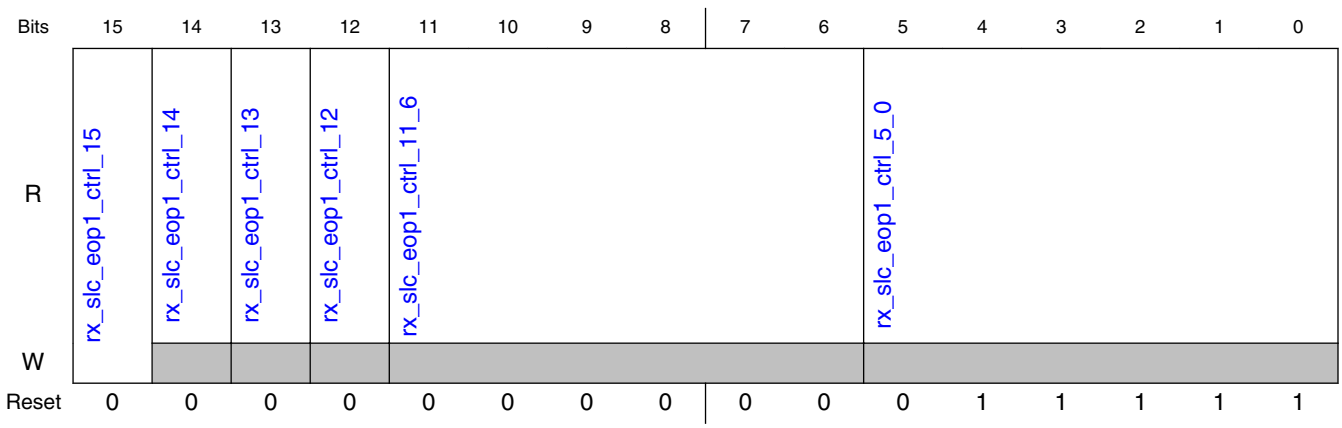
Field	Function
15-6 rx_slc_eop0_tune_15_6	Reserved
5-0 rx_slc_eop0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.208 RX sampler latch E odd positive 1 calibration unit control register (lane0_rx_slc_eop1_ctrl - lane3_rx_slc_eop1_ctrl)

13.4.10.2.208.1 Offset

Register	Offset
lane0_rx_slc_eop1_ctrl	8114h
lane1_rx_slc_eop1_ctrl	8514h
lane2_rx_slc_eop1_ctrl	8914h
lane3_rx_slc_eop1_ctrl	8D14h

13.4.10.2.208.2 Diagram



13.4.10.2.208.3 Fields

Field	Function
15 rx_slc_eop1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_eop1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_eop1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eop1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eop1_ctrl_11_6	Reserved
5-0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

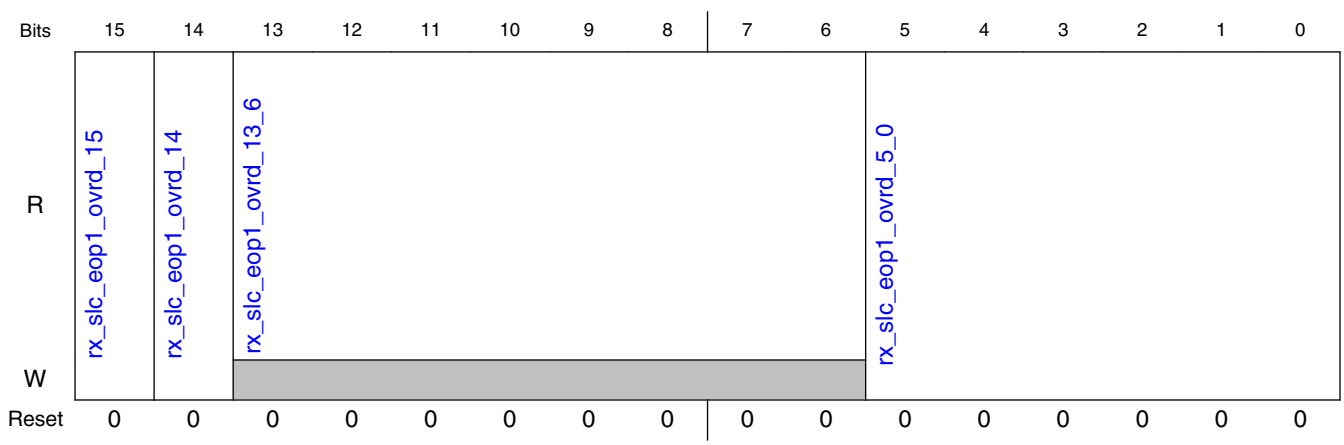
Field	Function
rx_slc_eop1_ctrl_5_0	

13.4.10.2.209 RX sampler latch E odd positive 1 calibration unit override register (lane0_rx_slc_eop1_ovrd - lane3_rx_slc_eop1_ovrd)

13.4.10.2.209.1 Offset

Register	Offset
lane0_rx_slc_eop1_ovrd	8115h
lane1_rx_slc_eop1_ovrd	8515h
lane2_rx_slc_eop1_ovrd	8915h
lane3_rx_slc_eop1_ovrd	8D15h

13.4.10.2.209.2 Diagram



13.4.10.2.209.3 Fields

Field	Function
15 rx_slc_eop1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eop1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the

Table continues on the next page...

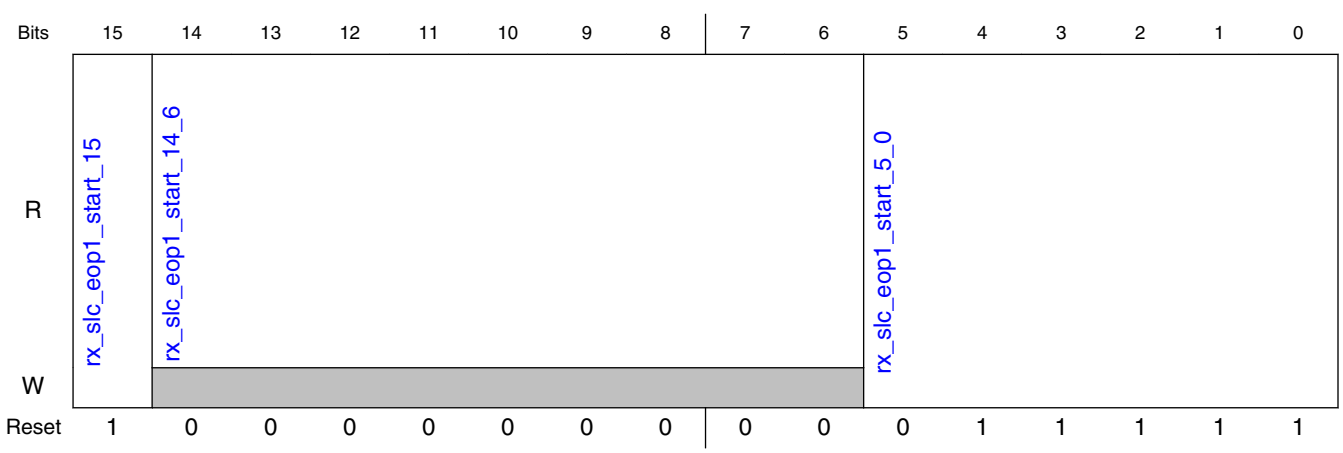
Field	Function
13-6 rx_slc_eop1_ovrd_13_6	Reserved
5-0 rx_slc_eop1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.210 RX sampler latch E odd positive 1 calibration unit start register (lane0_rx_slc_eop1_start - lane3_rx_slc_eop1_start)

13.4.10.2.210.1 Offset

Register	Offset
lane0_rx_slc_eop1_start	8116h
lane1_rx_slc_eop1_start	8516h
lane2_rx_slc_eop1_start	8916h
lane3_rx_slc_eop1_start	8D16h

13.4.10.2.210.2 Diagram



13.4.10.2.210.3 Fields

Field	Function
15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.

Table continues on the next page...

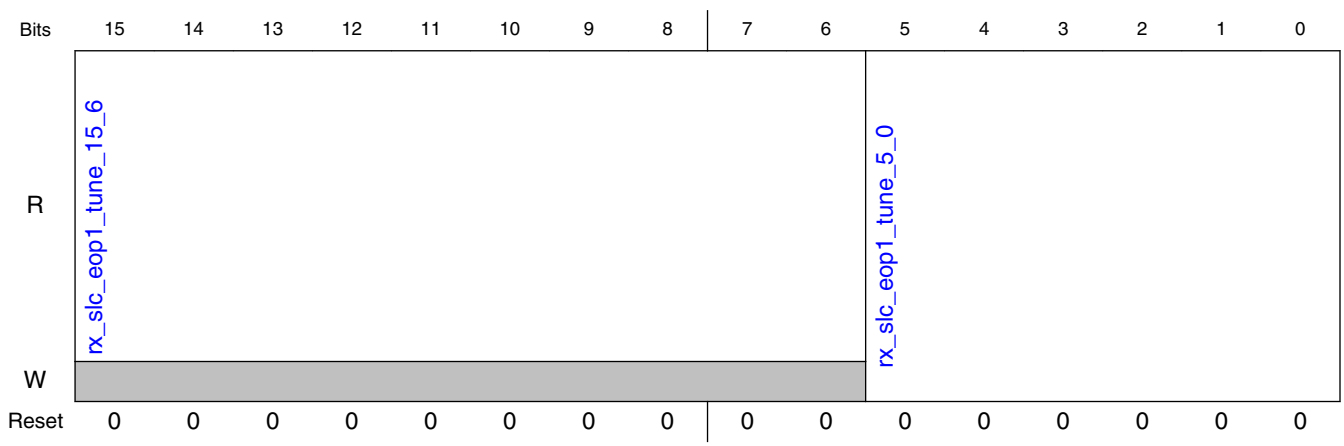
Field	Function
rx_slc_eop1_start_15	
14-6 rx_slc_eop1_start_14_6	Reserved
5-0 rx_slc_eop1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.211 RX sampler latch E odd positive 1 calibration unit tune register (lane0_rx_slc_eop1_tune - lane3_rx_slc_eop1_tune)

13.4.10.2.211.1 Offset

Register	Offset
lane0_rx_slc_eop1_tune	8117h
lane1_rx_slc_eop1_tune	8517h
lane2_rx_slc_eop1_tune	8917h
lane3_rx_slc_eop1_tune	8D17h

13.4.10.2.211.2 Diagram



13.4.10.2.211.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

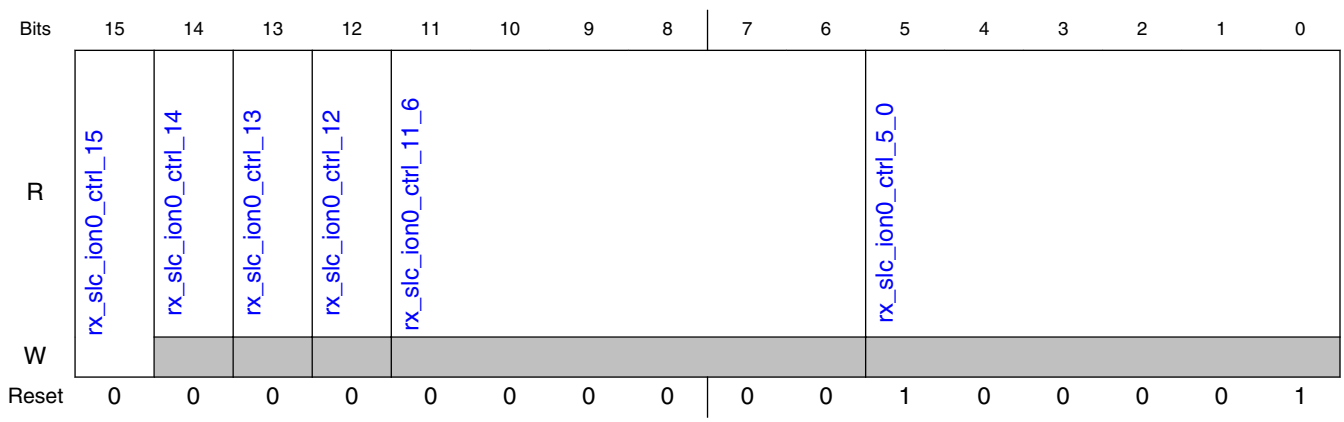
Field	Function
rx_slc_eop1_tune_15_6	
5-0 rx_slc_eop1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.212 RX sampler latch I odd negative 0 calibration unit control register (lane0_rx_slc_ion0_ctrl - lane3_rx_slc_ion0_ctrl)

13.4.10.2.212.1 Offset

Register	Offset
lane0_rx_slc_ion0_ctrl	8118h
lane1_rx_slc_ion0_ctrl	8518h
lane2_rx_slc_ion0_ctrl	8918h
lane3_rx_slc_ion0_ctrl	8D18h

13.4.10.2.212.2 Diagram



13.4.10.2.212.3 Fields

Field	Function
15 rx_slc_ion0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by

Table continues on the next page...

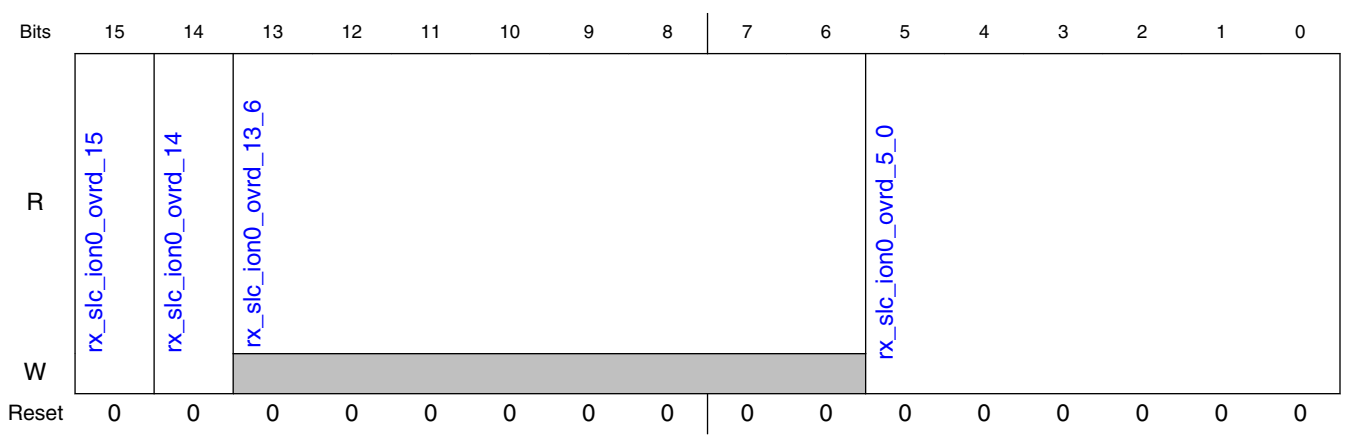
Field	Function
rx_slc_ion0_ctrl_14	
13 rx_slc_ion0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_ion0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_ion0_ctrl_11_6	Reserved
5-0 rx_slc_ion0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.213 RX sampler latch I odd negative 0 calibration unit override register (lane0_rx_slc_ion0_ovrd - lane3_rx_slc_ion0_ovrd)

13.4.10.2.213.1 Offset

Register	Offset
lane0_rx_slc_ion0_ovrd	8119h
lane1_rx_slc_ion0_ovrd	8519h
lane2_rx_slc_ion0_ovrd	8919h
lane3_rx_slc_ion0_ovrd	8D19h

13.4.10.2.213.2 Diagram



13.4.10.2.213.3 Fields

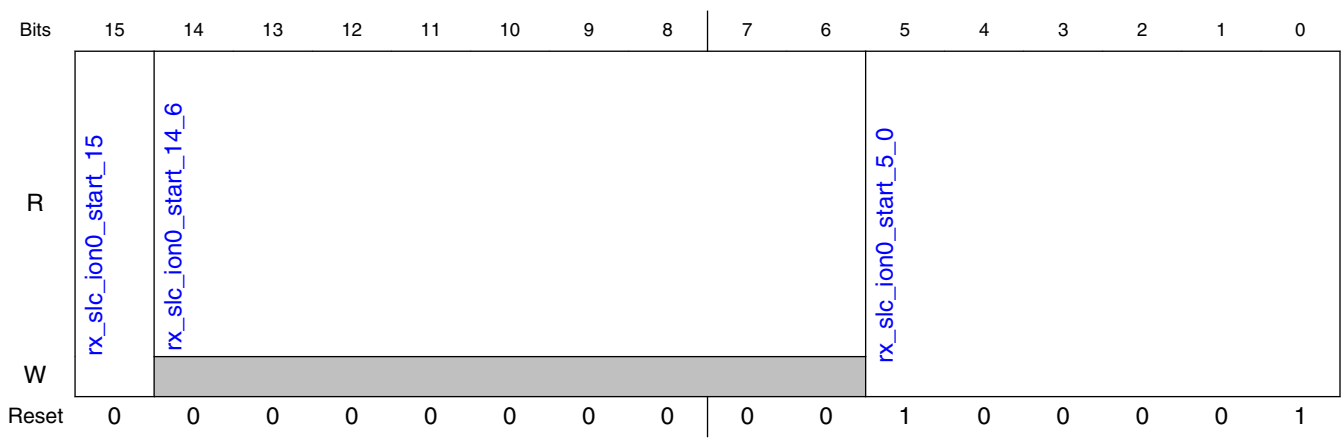
Field	Function
15 rx_slc_ion0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_ion0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_ion0_ovrd_13_6	Reserved
5-0 rx_slc_ion0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.214 RX sampler latch I odd negative 0 calibration unit start register (lane0_rx_slc_ion0_start - lane3_rx_slc_ion0_start)

13.4.10.2.214.1 Offset

Register	Offset
lane0_rx_slc_ion0_start	811Ah
lane1_rx_slc_ion0_start	851Ah
lane2_rx_slc_ion0_start	891Ah
lane3_rx_slc_ion0_start	8D1Ah

13.4.10.2.214.2 Diagram



13.4.10.2.214.3 Fields

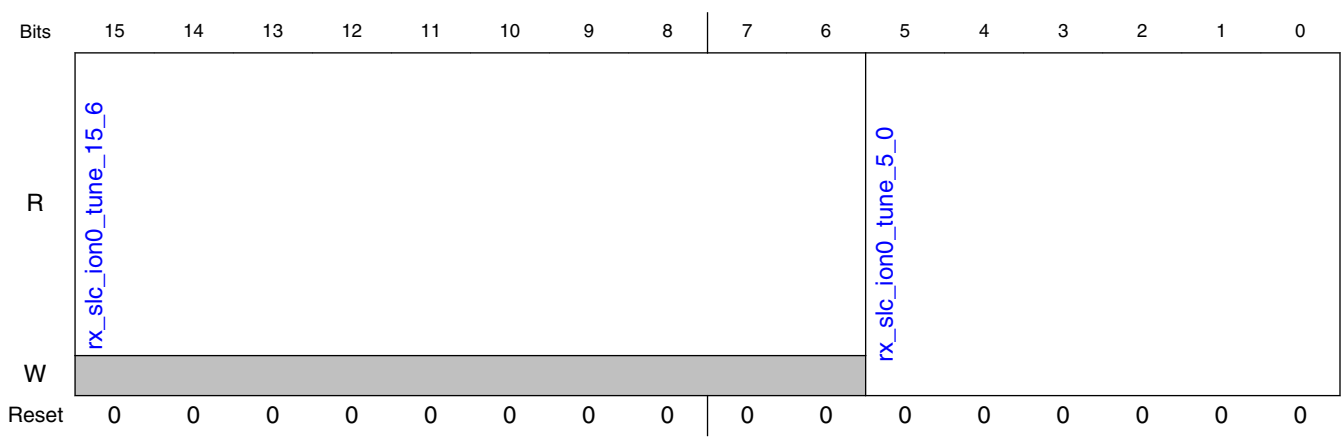
Field	Function
15 rx_slc_ion0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_ion0_start_14_6	Reserved
5-0 rx_slc_ion0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.215 RX sampler latch I odd negative 0 calibration unit tune register (lane0_rx_slc_ion0_tune - lane3_rx_slc_ion0_tune)

13.4.10.2.215.1 Offset

Register	Offset
lane0_rx_slc_ion0_tune	811Bh
lane1_rx_slc_ion0_tune	851Bh
lane2_rx_slc_ion0_tune	891Bh
lane3_rx_slc_ion0_tune	8D1Bh

13.4.10.2.215.2 Diagram



13.4.10.2.215.3 Fields

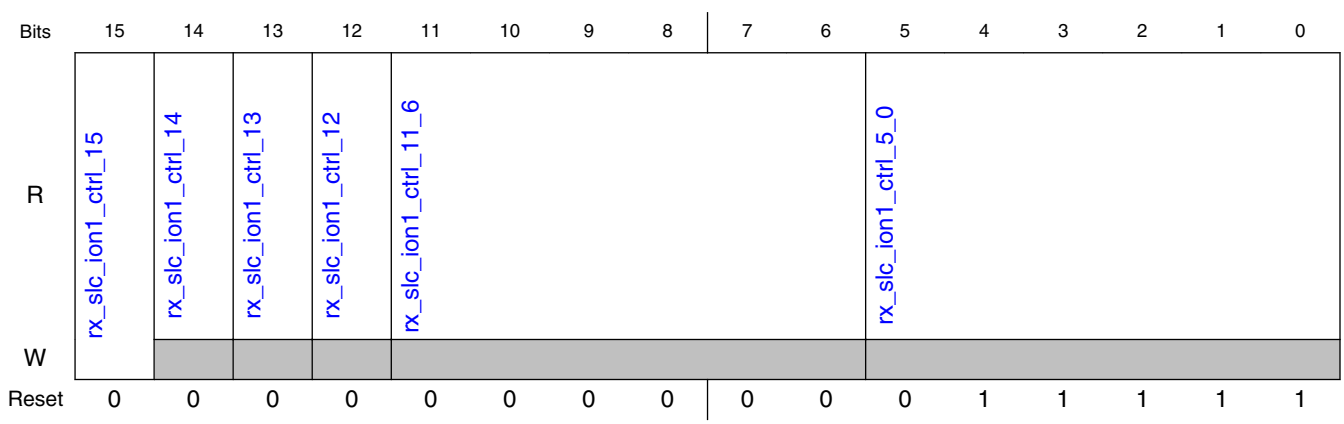
Field	Function
15-6 rx_slc_ion0_tune_15_6	Reserved
5-0 rx_slc_ion0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.216 RX sampler latch I odd negative 1 calibration unit control register (lane0_rx_slc_ion1_ctrl - lane3_rx_slc_ion1_ctrl)

13.4.10.2.216.1 Offset

Register	Offset
lane0_rx_slc_ion1_ctrl	811Ch
lane1_rx_slc_ion1_ctrl	851Ch
lane2_rx_slc_ion1_ctrl	891Ch
lane3_rx_slc_ion1_ctrl	8D1Ch

13.4.10.2.216.2 Diagram



13.4.10.2.216.3 Fields

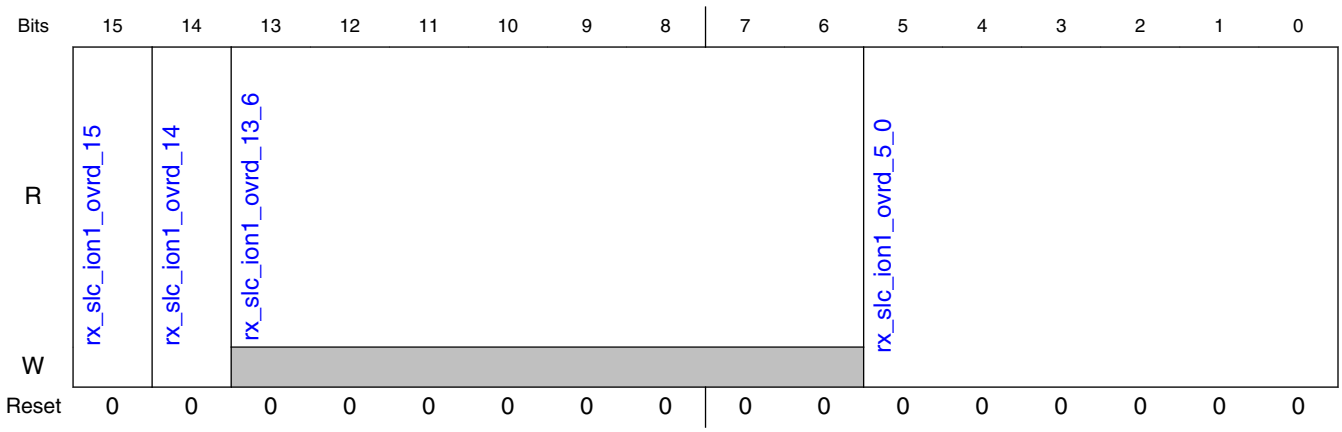
Field	Function
15 rx_slc_ion1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_ion1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_ion1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_ion1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_ion1_ctrl_11_6	Reserved
5-0 rx_slc_ion1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.217 RX sampler latch I odd negative 1 calibration unit override register (lane0_rx_slc_ion1_ovrd - lane3_rx_slc_ion1_ovrd)

13.4.10.2.217.1 Offset

Register	Offset
lane0_rx_slc_ion1_ovrd	811Dh
lane1_rx_slc_ion1_ovrd	851Dh
lane2_rx_slc_ion1_ovrd	891Dh
lane3_rx_slc_ion1_ovrd	8D1Dh

13.4.10.2.217.2 Diagram



13.4.10.2.217.3 Fields

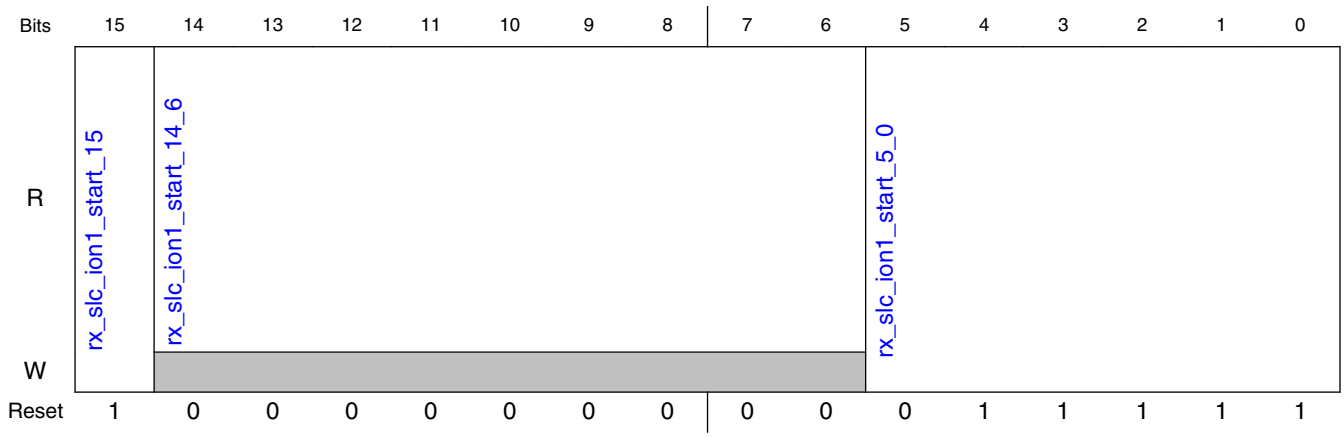
Field	Function
15 rx_slc_ion1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_ion1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_ion1_ovrd_13_6	Reserved
5-0 rx_slc_ion1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.218 RX sampler latch I odd negative 1 calibration unit start register (lane0_rx_slc_ion1_start - lane3_rx_slc_ion1_start)

13.4.10.2.218.1 Offset

Register	Offset
lane0_rx_slc_ion1_start	811Eh
lane1_rx_slc_ion1_start	851Eh
lane2_rx_slc_ion1_start	891Eh
lane3_rx_slc_ion1_start	8D1Eh

13.4.10.2.218.2 Diagram



13.4.10.2.218.3 Fields

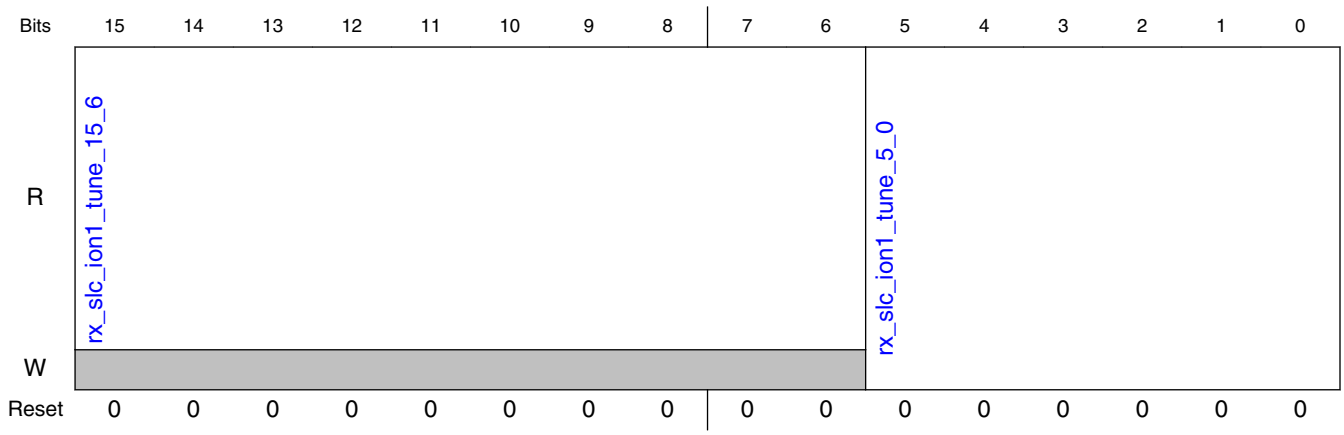
Field	Function
15 rx_slc_ion1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_ion1_start_14_6	Reserved
5-0 rx_slc_ion1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.219 RX sampler latch I odd negative 1 calibration unit tune register (lane0_rx_slc_ion1_tune - lane3_rx_slc_ion1_tune)

13.4.10.2.219.1 Offset

Register	Offset
lane0_rx_slc_ion1_tune	811Fh
lane1_rx_slc_ion1_tune	851Fh
lane2_rx_slc_ion1_tune	891Fh
lane3_rx_slc_ion1_tune	8D1Fh

13.4.10.2.219.2 Diagram



13.4.10.2.219.3 Fields

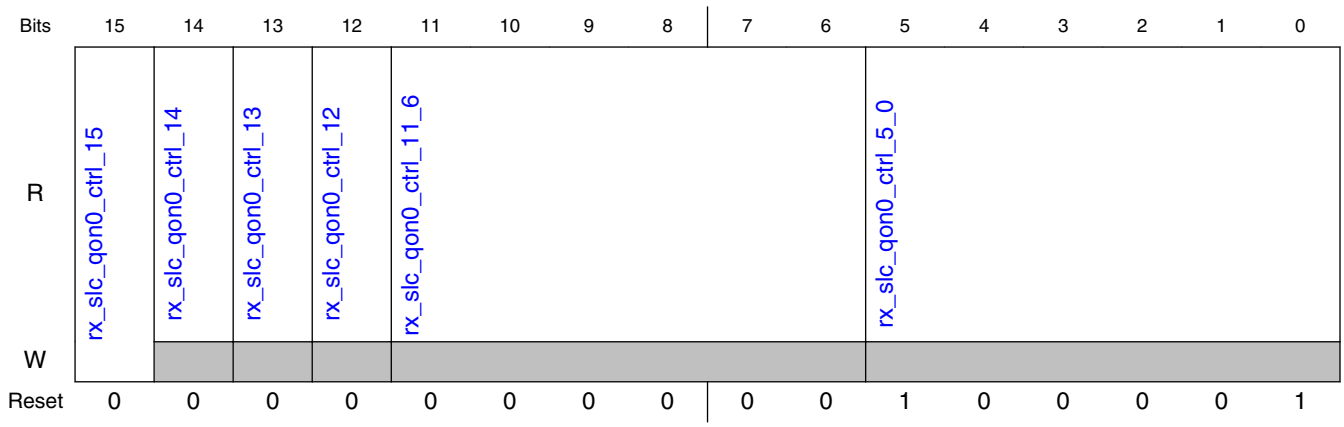
Field	Function
15-6 rx_slc_ion1_tune_15_6	Reserved
5-0 rx_slc_ion1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.220 RX sampler latch Q odd negative 0 calibration unit control register (lane0_rx_slc_qon0_ctrl - lane3_rx_slc_qon0_ctrl)

13.4.10.2.220.1 Offset

Register	Offset
lane0_rx_slc_qon0_ctrl	8120h
lane1_rx_slc_qon0_ctrl	8520h
lane2_rx_slc_qon0_ctrl	8920h
lane3_rx_slc_qon0_ctrl	8D20h

13.4.10.2.220.2 Diagram



13.4.10.2.220.3 Fields

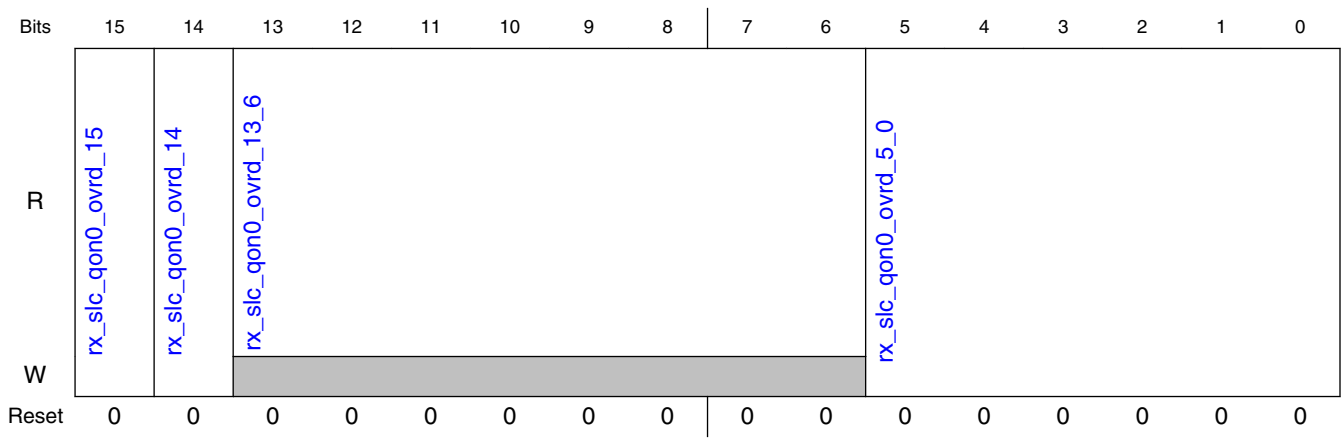
Field	Function
15 rx_slc_qon0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qon0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qon0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qon0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qon0_ctrl_11_6	Reserved
5-0 rx_slc_qon0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.221 RX sampler latch Q odd negative 0 calibration unit override register (lane0_rx_slc_qon0_ovrd - lane3_rx_slc_qon0_ovrd)

13.4.10.2.221.1 Offset

Register	Offset
lane0_rx_slc_qon0_ovrd	8121h
lane1_rx_slc_qon0_ovrd	8521h
lane2_rx_slc_qon0_ovrd	8921h
lane3_rx_slc_qon0_ovrd	8D21h

13.4.10.2.221.2 Diagram



13.4.10.2.221.3 Fields

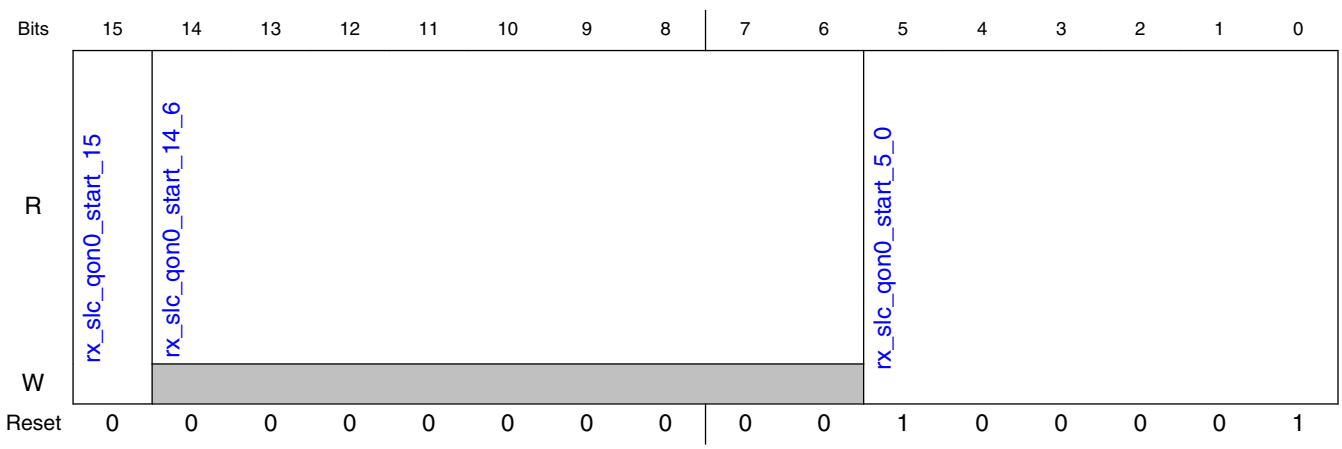
Field	Function
15 rx_slc_qon0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qon0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qon0_ovrd_13_6	Reserved
5-0 rx_slc_qon0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.222 RX sampler latch Q odd negative 0 calibration unit start register (lane0_rx_slc_qon0_start - lane3_rx_slc_qon0_start)

13.4.10.2.222.1 Offset

Register	Offset
lane0_rx_slc_qon0_start	8122h
lane1_rx_slc_qon0_start	8522h
lane2_rx_slc_qon0_start	8922h
lane3_rx_slc_qon0_start	8D22h

13.4.10.2.222.2 Diagram



13.4.10.2.222.3 Fields

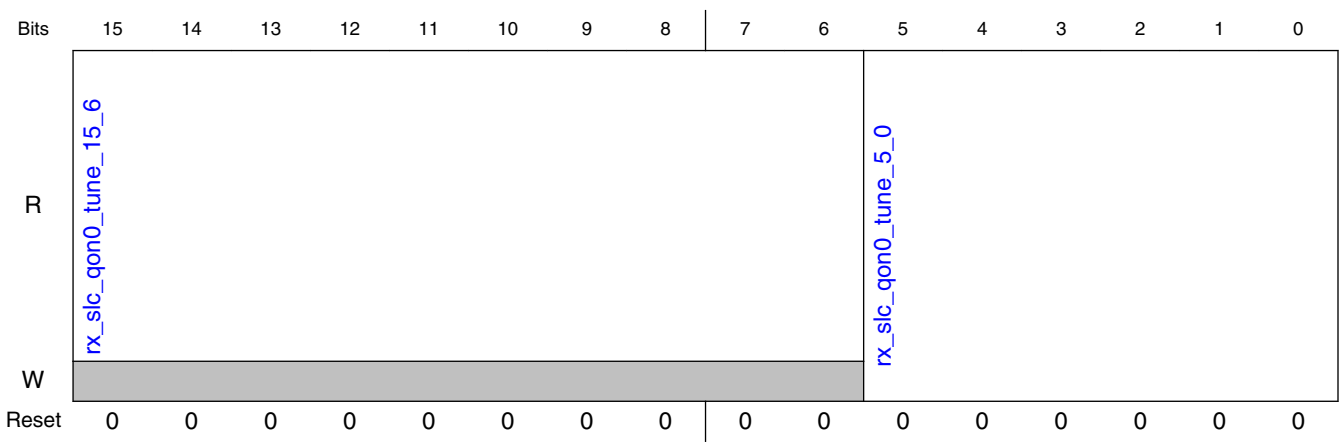
Field	Function
15 rx_slc_qon0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qon0_start_14_6	Reserved
5-0 rx_slc_qon0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.223 RX sampler latch Q odd negative 0 calibration unit tune register (lane0_rx_slc_qon0_tune - lane3_rx_slc_qon0_tune)

13.4.10.2.223.1 Offset

Register	Offset
lane0_rx_slc_qon0_tune	8123h
lane1_rx_slc_qon0_tune	8523h
lane2_rx_slc_qon0_tune	8923h
lane3_rx_slc_qon0_tune	8D23h

13.4.10.2.223.2 Diagram



13.4.10.2.223.3 Fields

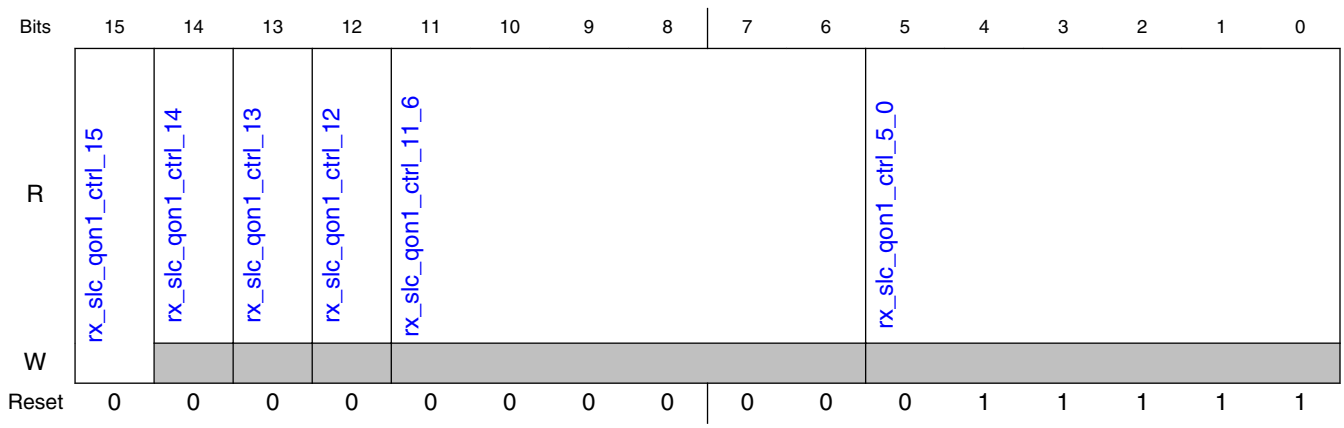
Field	Function
15-6 rx_slc_qon0_tune_15_6	Reserved
5-0 rx_slc_qon0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.224 RX sampler latch Q odd negative 1 calibration unit control register (lane0_rx_slc_qon1_ctrl - lane3_rx_slc_qon1_ctrl)

13.4.10.2.224.1 Offset

Register	Offset
lane0_rx_slc_qon1_ctrl	8124h
lane1_rx_slc_qon1_ctrl	8524h
lane2_rx_slc_qon1_ctrl	8924h
lane3_rx_slc_qon1_ctrl	8D24h

13.4.10.2.224.2 Diagram



13.4.10.2.224.3 Fields

Field	Function
15 rx_slc_qon1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qon1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qon1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qon1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qon1_ctrl_11_6	Reserved
5-0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

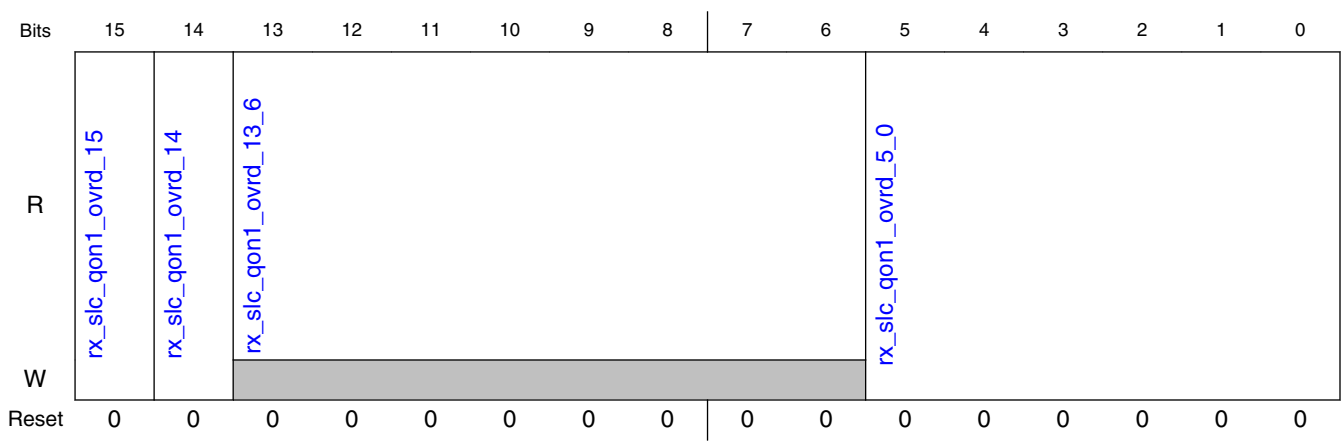
Field	Function
rx_slc_qon1_ctrl_5_0	

13.4.10.2.225 RX sampler latch Q odd negative 1 calibration unit override register (lane0_rx_slc_qon1_ovrd - lane3_rx_slc_qon1_ovrd)

13.4.10.2.225.1 Offset

Register	Offset
lane0_rx_slc_qon1_ovrd	8125h
lane1_rx_slc_qon1_ovrd	8525h
lane2_rx_slc_qon1_ovrd	8925h
lane3_rx_slc_qon1_ovrd	8D25h

13.4.10.2.225.2 Diagram



13.4.10.2.225.3 Fields

Field	Function
15 rx_slc_qon1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qon1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the

Table continues on the next page...

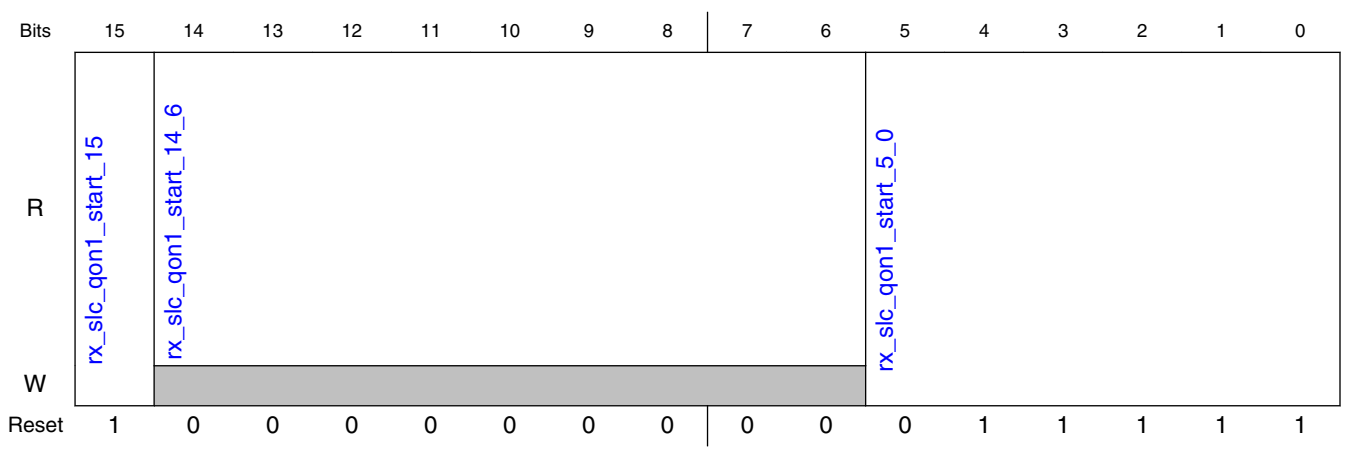
Field	Function
13-6 rx_slc_qon1_ovrd_13_6	Reserved
5-0 rx_slc_qon1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.226 RX sampler latch Q odd negative 1 calibration unit start register (lane0_rx_slc_qon1_start - lane3_rx_slc_qon1_start)

13.4.10.2.226.1 Offset

Register	Offset
lane0_rx_slc_qon1_start	8126h
lane1_rx_slc_qon1_start	8526h
lane2_rx_slc_qon1_start	8926h
lane3_rx_slc_qon1_start	8D26h

13.4.10.2.226.2 Diagram



13.4.10.2.226.3 Fields

Field	Function
15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.

Table continues on the next page...

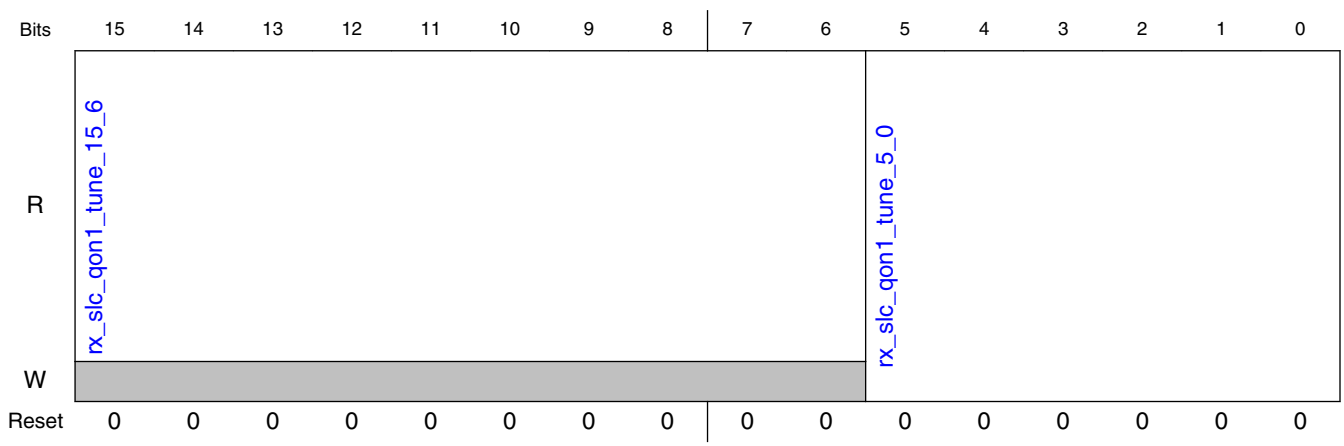
Field	Function
rx_slc_qon1_start_15	
14-6 rx_slc_qon1_start_14_6	Reserved
5-0 rx_slc_qon1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.227 RX sampler latch Q odd negative 1 calibration unit tune register (lane0_rx_slc_qon1_tune - lane3_rx_slc_qon1_tune)

13.4.10.2.227.1 Offset

Register	Offset
lane0_rx_slc_qon1_tune	8127h
lane1_rx_slc_qon1_tune	8527h
lane2_rx_slc_qon1_tune	8927h
lane3_rx_slc_qon1_tune	8D27h

13.4.10.2.227.2 Diagram



13.4.10.2.227.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

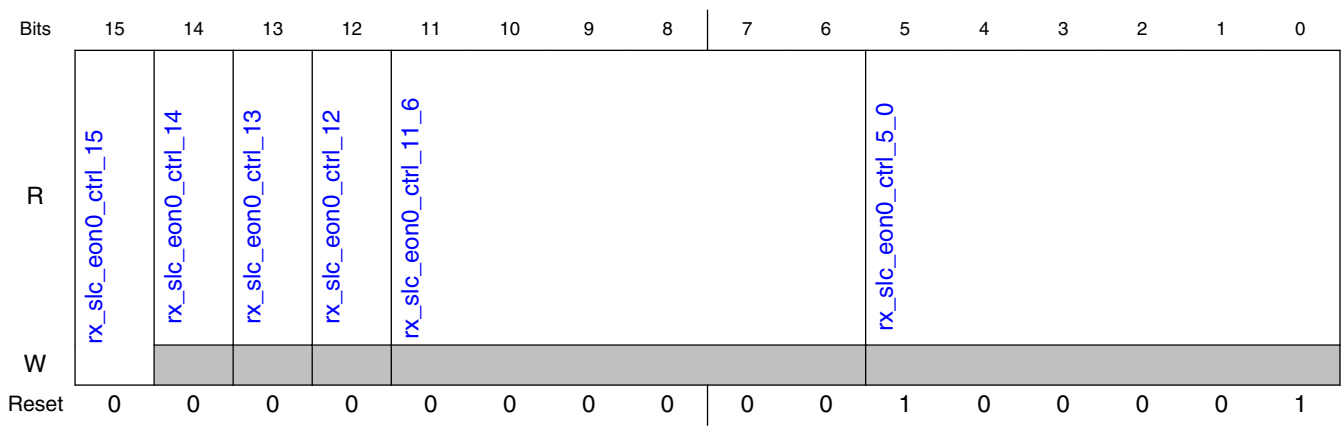
Field	Function
rx_slc_qon1_tune_15_6	
5-0 rx_slc_qon1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.228 RX sampler latch E odd negative 0 calibration unit control register (lane0_rx_slc_eon0_ctrl - lane3_rx_slc_eon0_ctrl)

13.4.10.2.228.1 Offset

Register	Offset
lane0_rx_slc_eon0_ctrl	8128h
lane1_rx_slc_eon0_ctrl	8528h
lane2_rx_slc_eon0_ctrl	8928h
lane3_rx_slc_eon0_ctrl	8D28h

13.4.10.2.228.2 Diagram



13.4.10.2.228.3 Fields

Field	Function
15 rx_slc_eon0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by

Table continues on the next page...

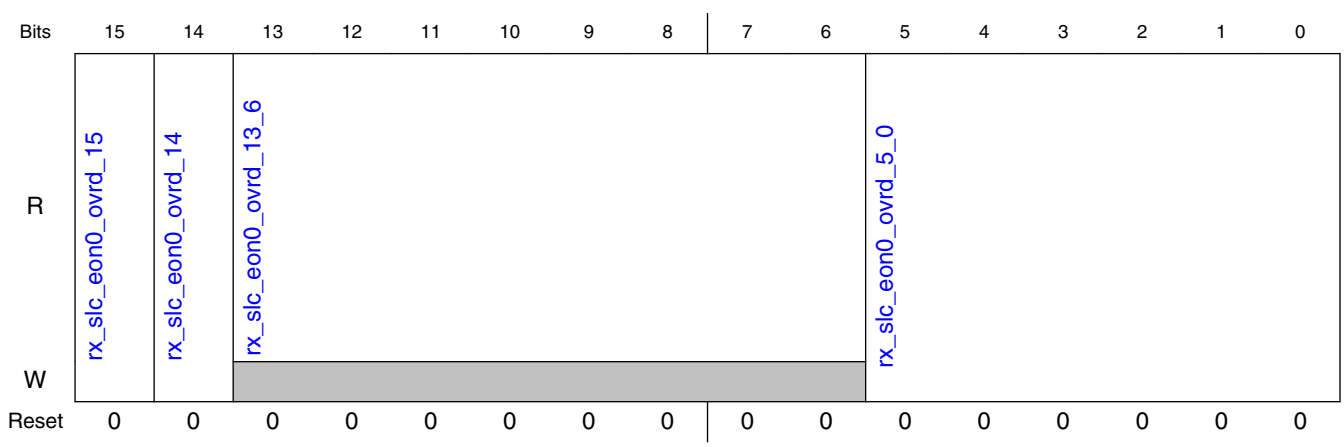
Field	Function
rx_slc_eon0_ctrl_14	
13 rx_slc_eon0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eon0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eon0_ctrl_11_6	Reserved
5-0 rx_slc_eon0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.229 RX sampler latch E odd negative 0 calibration unit override register (lane0_rx_slc_eon0_ovrd - lane3_rx_slc_eon0_ovrd)

13.4.10.2.229.1 Offset

Register	Offset
lane0_rx_slc_eon0_ovrd	8129h
lane1_rx_slc_eon0_ovrd	8529h
lane2_rx_slc_eon0_ovrd	8929h
lane3_rx_slc_eon0_ovrd	8D29h

13.4.10.2.229.2 Diagram



13.4.10.2.229.3 Fields

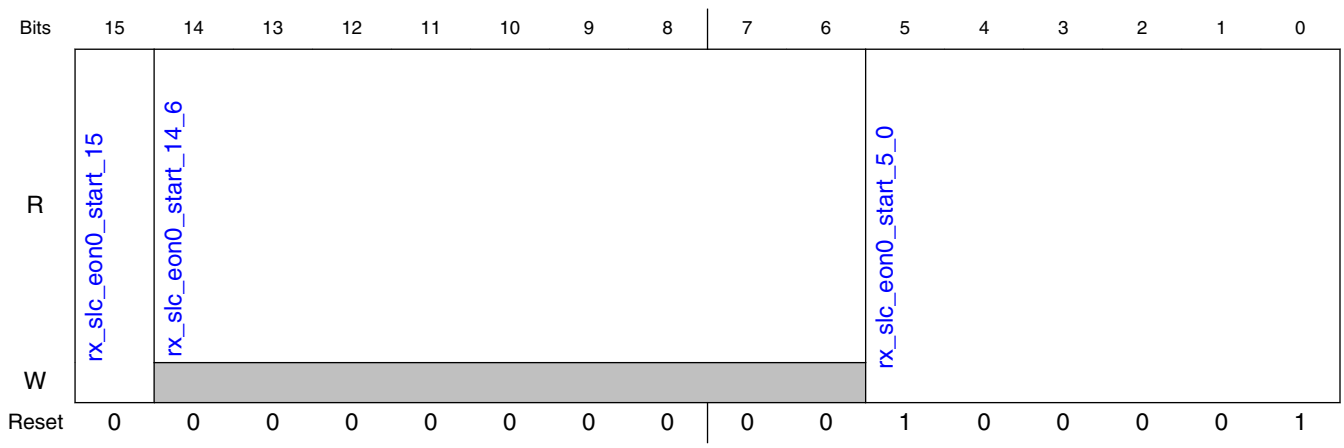
Field	Function
15 rx_slc_eon0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eon0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_eon0_ovrd_13_6	Reserved
5-0 rx_slc_eon0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.230 RX sampler latch E odd negative 0 calibration unit start register (lane0_rx_slc_eon0_start - lane3_rx_slc_eon0_start)

13.4.10.2.230.1 Offset

Register	Offset
lane0_rx_slc_eon0_start	812Ah
lane1_rx_slc_eon0_start	852Ah
lane2_rx_slc_eon0_start	892Ah
lane3_rx_slc_eon0_start	8D2Ah

13.4.10.2.230.2 Diagram



13.4.10.2.230.3 Fields

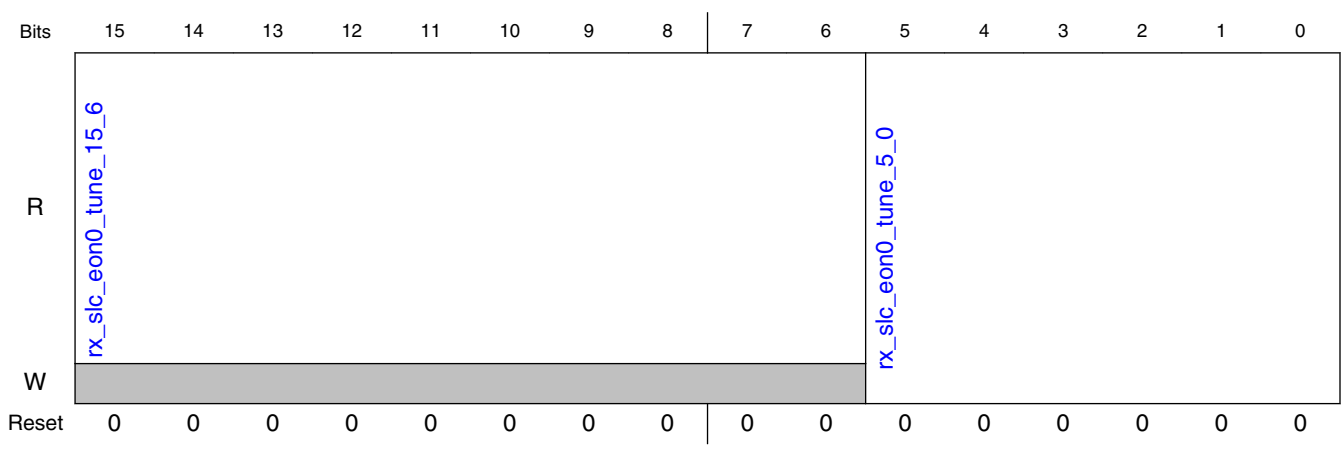
Field	Function
15 rx_slc_eon0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_eon0_start_14_6	Reserved
5-0 rx_slc_eon0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.231 RX sampler latch E odd negative 0 calibration unit tune register (lane0_rx_slc_eon0_tune - lane3_rx_slc_eon0_tune)

13.4.10.2.231.1 Offset

Register	Offset
lane0_rx_slc_eon0_tune	812Bh
lane1_rx_slc_eon0_tune	852Bh
lane2_rx_slc_eon0_tune	892Bh
lane3_rx_slc_eon0_tune	8D2Bh

13.4.10.2.231.2 Diagram



13.4.10.2.231.3 Fields

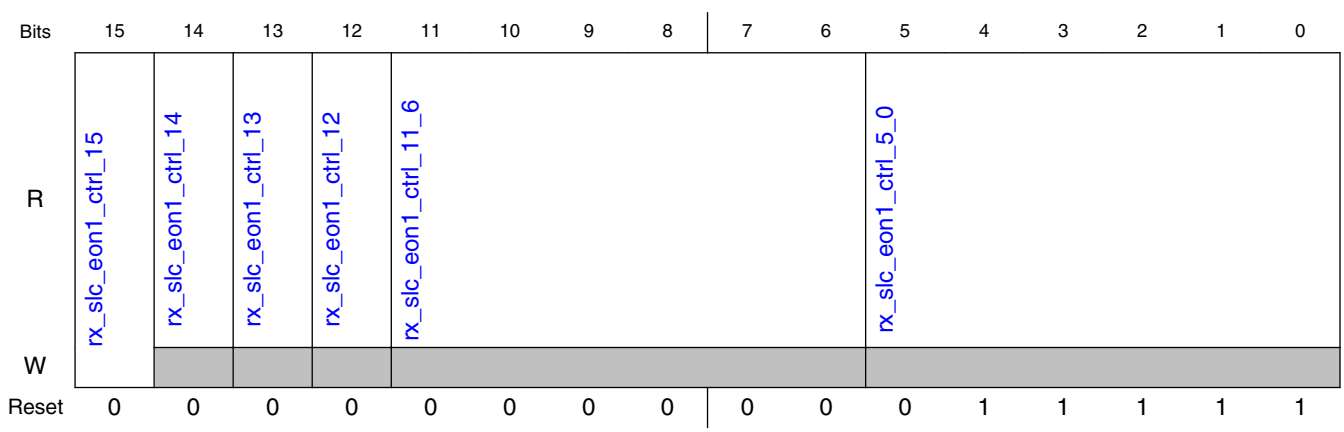
Field	Function
15-6 rx_slc_eon0_tune_15_6	Reserved
5-0 rx_slc_eon0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.232 RX sampler latch E odd negative 1 calibration unit control register (lane0_rx_slc_eon1_ctrl - lane3_rx_slc_eon1_ctrl)

13.4.10.2.232.1 Offset

Register	Offset
lane0_rx_slc_eon1_ctrl	812Ch
lane1_rx_slc_eon1_ctrl	852Ch
lane2_rx_slc_eon1_ctrl	892Ch
lane3_rx_slc_eon1_ctrl	8D2Ch

13.4.10.2.232.2 Diagram



13.4.10.2.232.3 Fields

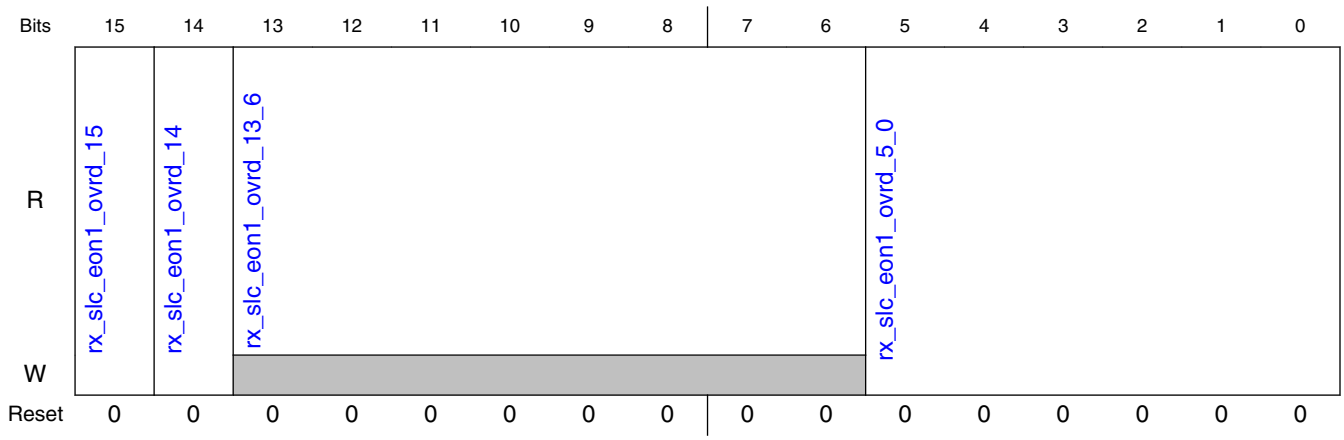
Field	Function
15 rx_slc_eon1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_eon1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_eon1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eon1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eon1_ctrl_11_6	Reserved
5-0 rx_slc_eon1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.233 RX sampler latch E odd negative 1 calibration unit override register (lane0_rx_slc_eon1_ovrd - lane3_rx_slc_eon1_ovrd)

13.4.10.2.233.1 Offset

Register	Offset
lane0_rx_slc_eon1_ovrd	812Dh
lane1_rx_slc_eon1_ovrd	852Dh
lane2_rx_slc_eon1_ovrd	892Dh
lane3_rx_slc_eon1_ovrd	8D2Dh

13.4.10.2.233.2 Diagram



13.4.10.2.233.3 Fields

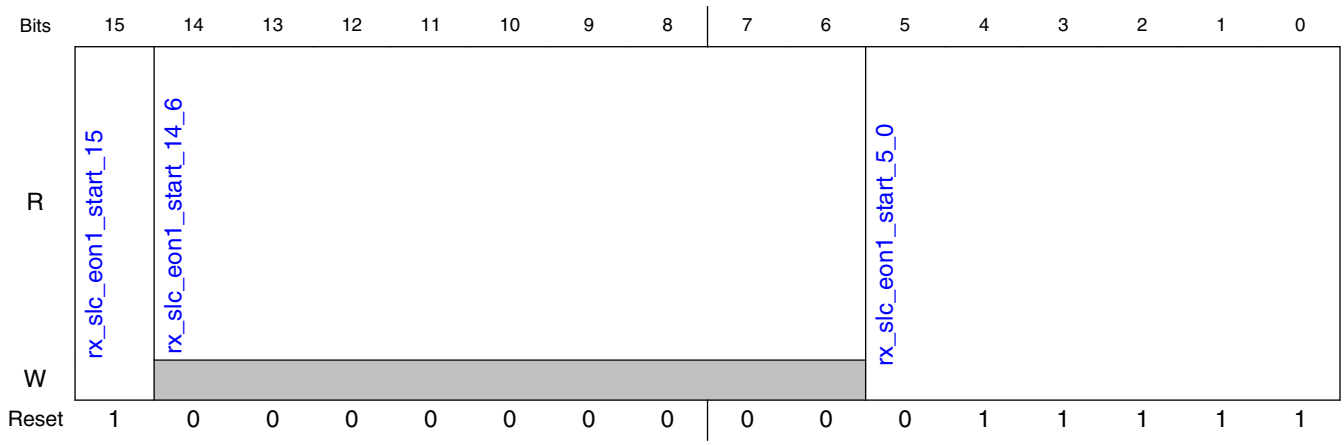
Field	Function
15 rx_slc_eon1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eon1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_eon1_ovrd_13_6	Reserved
5-0 rx_slc_eon1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.234 RX sampler latch E odd negative 1 calibration unit start register (lane0_rx_slc_eon1_start - lane3_rx_slc_eon1_start)

13.4.10.2.234.1 Offset

Register	Offset
lane0_rx_slc_eon1_start	812Eh
lane1_rx_slc_eon1_start	852Eh
lane2_rx_slc_eon1_start	892Eh
lane3_rx_slc_eon1_start	8D2Eh

13.4.10.2.234.2 Diagram



13.4.10.2.234.3 Fields

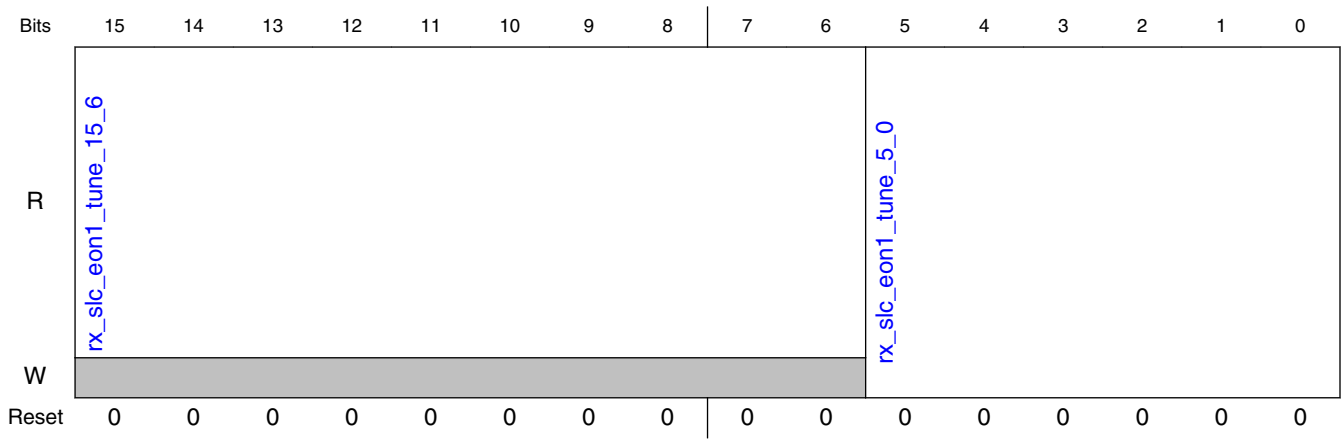
Field	Function
15 rx_slc_eon1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_eon1_start_14_6	Reserved
5-0 rx_slc_eon1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.235 RX sampler latch E odd negative 1 calibration unit tune register (lane0_rx_slc_eon1_tune - lane3_rx_slc_eon1_tune)

13.4.10.2.235.1 Offset

Register	Offset
lane0_rx_slc_eon1_tune	812Fh
lane1_rx_slc_eon1_tune	852Fh
lane2_rx_slc_eon1_tune	892Fh
lane3_rx_slc_eon1_tune	8D2Fh

13.4.10.2.235.2 Diagram



13.4.10.2.235.3 Fields

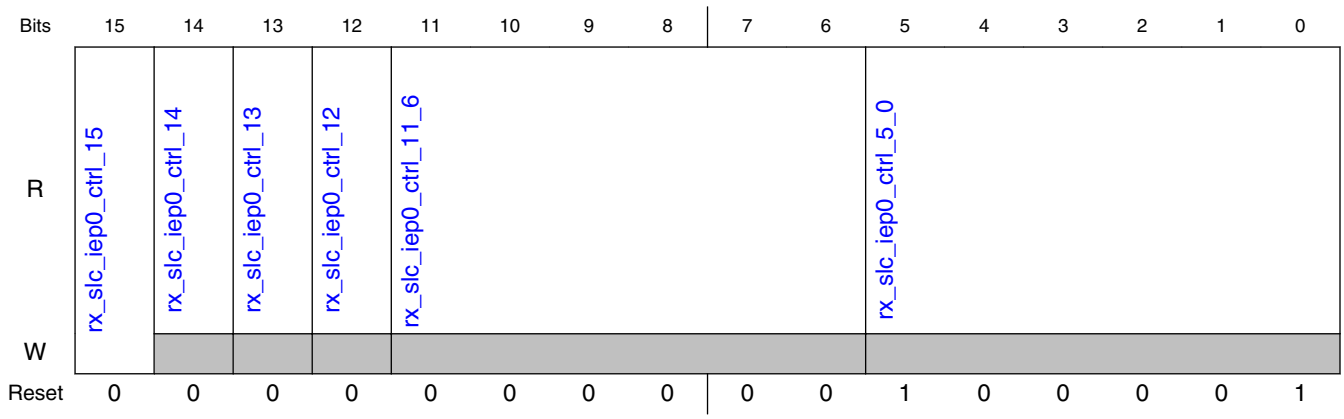
Field	Function
15-6 rx_slc_eon1_tune_15_6	Reserved
5-0 rx_slc_eon1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.236 RX sampler latch I even positive 0 calibration unit control register (lane0_rx_slc_iep0_ctrl - lane3_rx_slc_iep0_ctrl)

13.4.10.2.236.1 Offset

Register	Offset
lane0_rx_slc_iep0_ctrl	8130h
lane1_rx_slc_iep0_ctrl	8530h
lane2_rx_slc_iep0_ctrl	8930h
lane3_rx_slc_iep0_ctrl	8D30h

13.4.10.2.236.2 Diagram



13.4.10.2.236.3 Fields

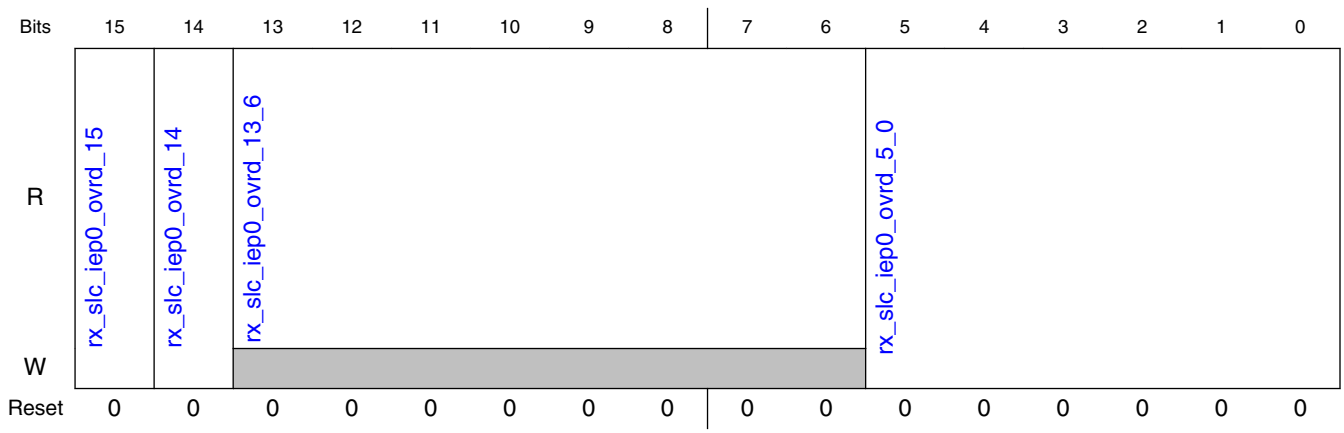
Field	Function
15 rx_slc_iep0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_iep0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_iep0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_iep0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_iep0_ctrl_11_6	Reserved
5-0 rx_slc_iep0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.237 RX sampler latch I even positive 0 calibration unit override register (lane0_rx_slc_iep0_ovrd - lane3_rx_slc_iep0_ovrd)

13.4.10.2.237.1 Offset

Register	Offset
lane0_rx_slc_iep0_ovrd	8131h
lane1_rx_slc_iep0_ovrd	8531h
lane2_rx_slc_iep0_ovrd	8931h
lane3_rx_slc_iep0_ovrd	8D31h

13.4.10.2.237.2 Diagram



13.4.10.2.237.3 Fields

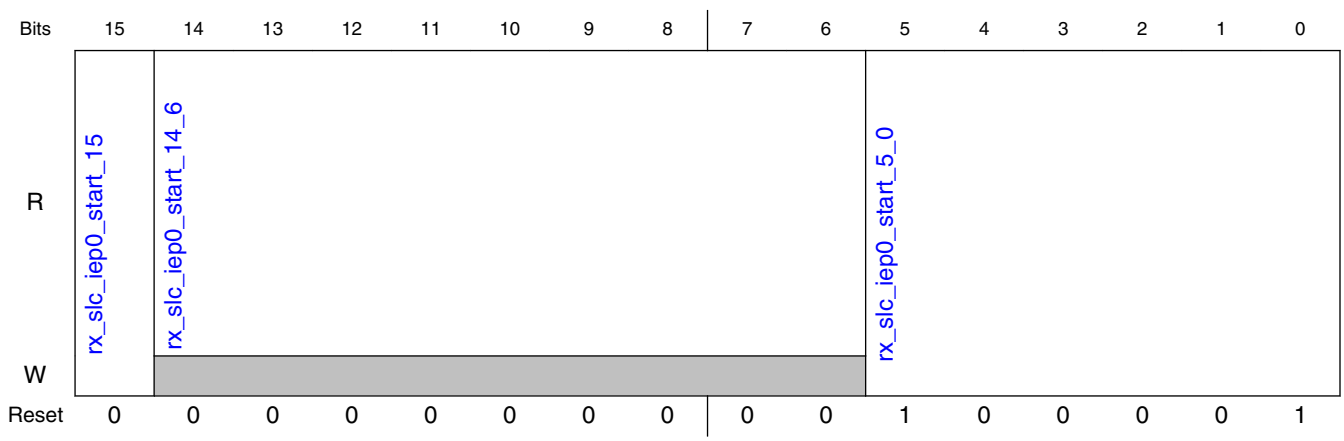
Field	Function
15 rx_slc_iep0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_iep0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_iep0_ovrd_13_6	Reserved
5-0 rx_slc_iep0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.238 RX sampler latch I even positive 0 calibration unit start register (lane0_rx_slc_iep0_start - lane3_rx_slc_iep0_start)

13.4.10.2.238.1 Offset

Register	Offset
lane0_rx_slc_iep0_start	8132h
lane1_rx_slc_iep0_start	8532h
lane2_rx_slc_iep0_start	8932h
lane3_rx_slc_iep0_start	8D32h

13.4.10.2.238.2 Diagram



13.4.10.2.238.3 Fields

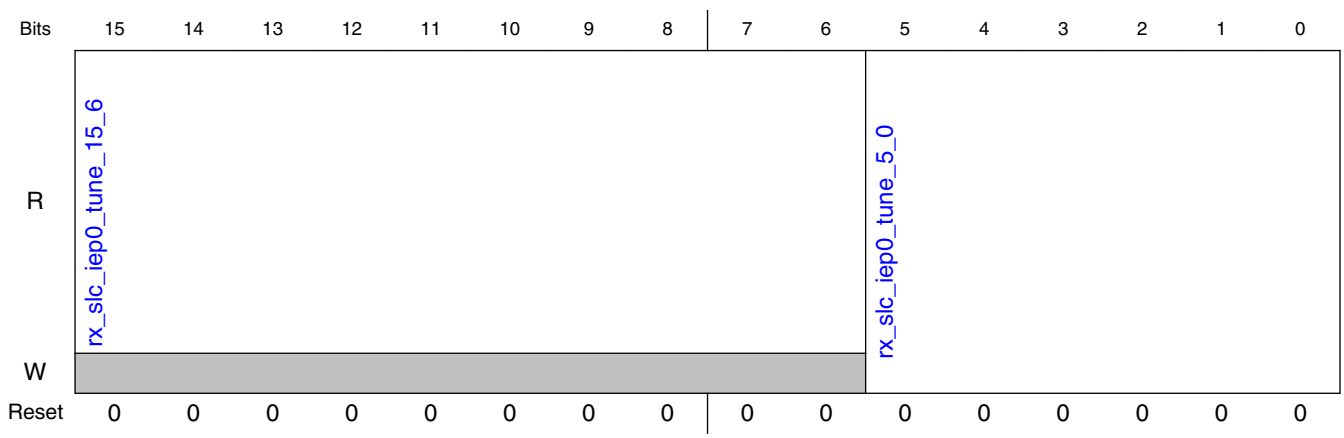
Field	Function
15 rx_slc_iep0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_iep0_start_14_6	Reserved
5-0 rx_slc_iep0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.239 RX sampler latch I even positive 0 calibration unit tune register (lane0_rx_slc_iep0_tune - lane3_rx_slc_iep0_tune)

13.4.10.2.239.1 Offset

Register	Offset
lane0_rx_slc_iep0_tune	8133h
lane1_rx_slc_iep0_tune	8533h
lane2_rx_slc_iep0_tune	8933h
lane3_rx_slc_iep0_tune	8D33h

13.4.10.2.239.2 Diagram



13.4.10.2.239.3 Fields

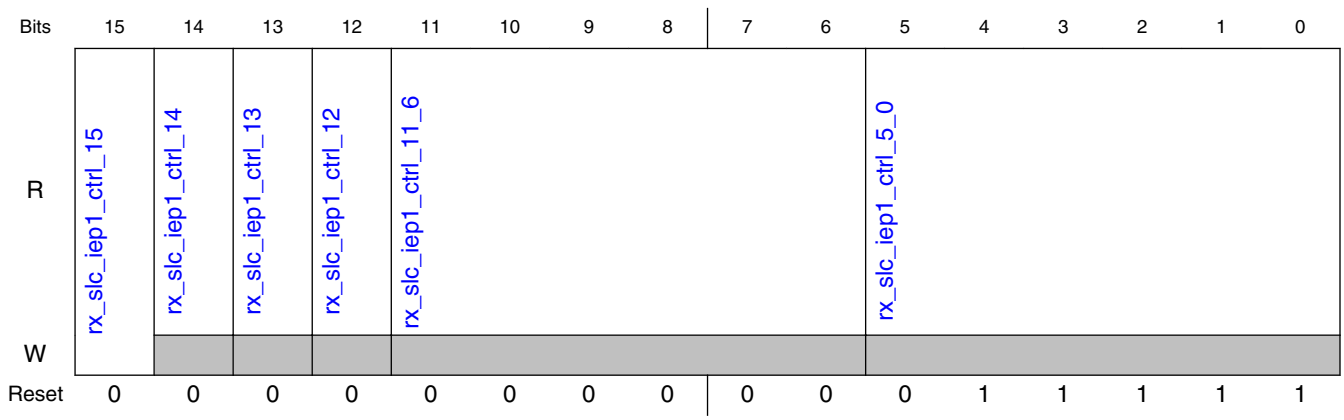
Field	Function
15-6 rx_slc_iep0_tune_15_6	Reserved
5-0 rx_slc_iep0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.240 RX sampler latch I even positive 1 calibration unit control register (lane0_rx_slc_iep1_ctrl - lane3_rx_slc_iep1_ctrl)

13.4.10.2.240.1 Offset

Register	Offset
lane0_rx_slc_iep1_ctrl	8134h
lane1_rx_slc_iep1_ctrl	8534h
lane2_rx_slc_iep1_ctrl	8934h
lane3_rx_slc_iep1_ctrl	8D34h

13.4.10.2.240.2 Diagram



13.4.10.2.240.3 Fields

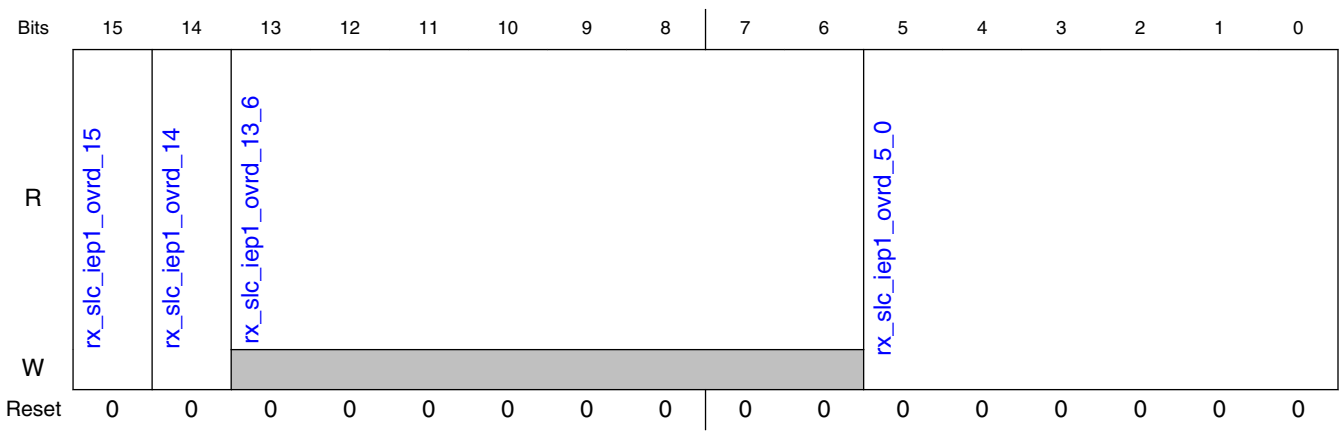
Field	Function
15 rx_slc_iep1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_iep1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_iep1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_iep1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_iep1_ctrl_11_6	Reserved
5-0 rx_slc_iep1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.241 RX sampler latch I even positive 1 calibration unit override register (lane0_rx_slc_iep1_ovrd - lane3_rx_slc_iep1_ovrd)

13.4.10.2.241.1 Offset

Register	Offset
lane0_rx_slc_iep1_ovrd	8135h
lane1_rx_slc_iep1_ovrd	8535h
lane2_rx_slc_iep1_ovrd	8935h
lane3_rx_slc_iep1_ovrd	8D35h

13.4.10.2.241.2 Diagram



13.4.10.2.241.3 Fields

Field	Function
15 rx_slc_iep1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_iep1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_iep1_ovrd_13_6	Reserved
5-0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the

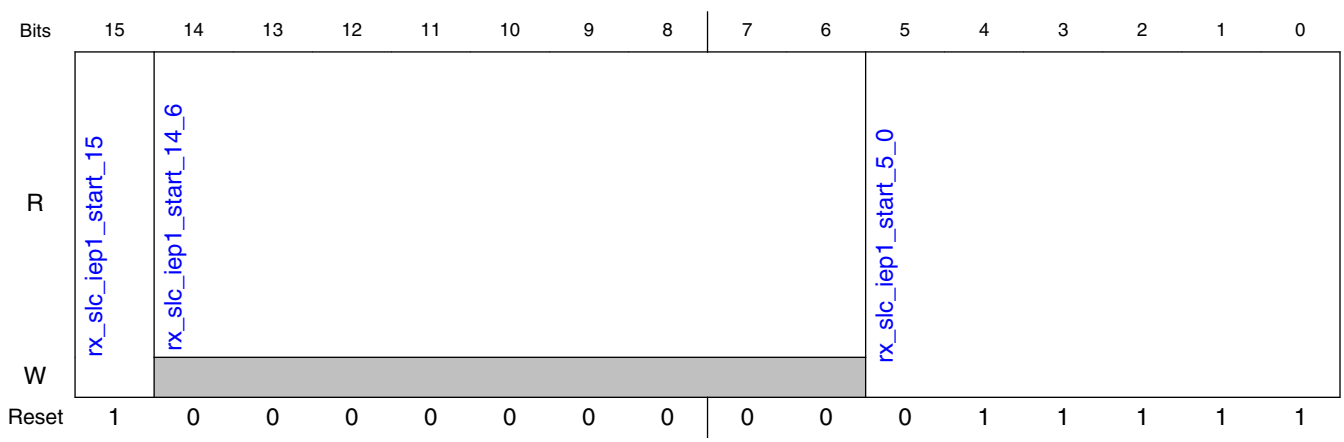
Field	Function
rx_slc_iep1_ovrd_5_0	calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.242 RX sampler latch I even positive 1 calibration unit start register (lane0_rx_slc_iep1_start - lane3_rx_slc_iep1_start)

13.4.10.2.242.1 Offset

Register	Offset
lane0_rx_slc_iep1_start	8136h
lane1_rx_slc_iep1_start	8536h
lane2_rx_slc_iep1_start	8936h
lane3_rx_slc_iep1_start	8D36h

13.4.10.2.242.2 Diagram



13.4.10.2.242.3 Fields

Field	Function
15 rx_slc_iep1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_iep1_start_14_6	Reserved

Table continues on the next page...

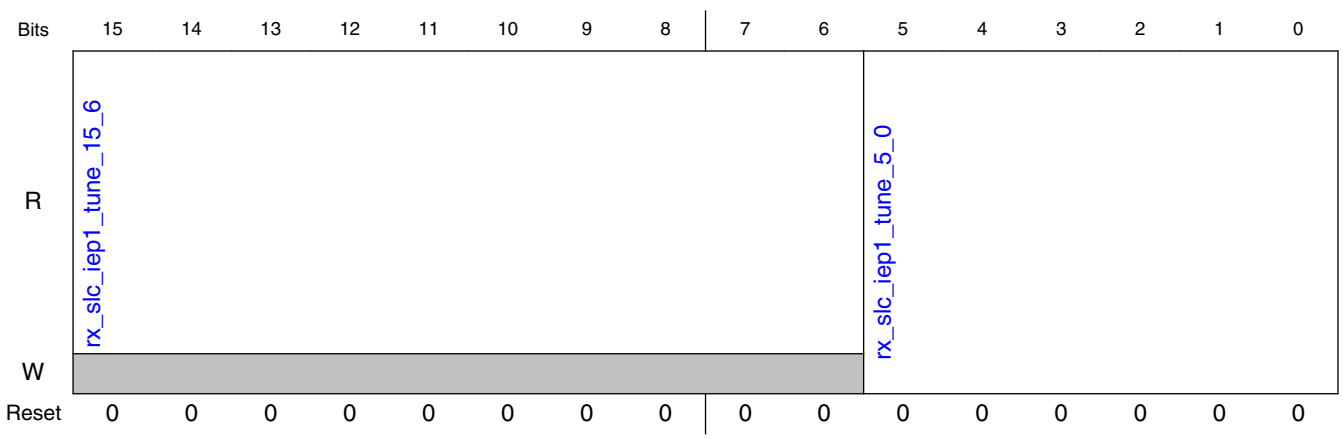
Field	Function
5-0 rx_slc_iep1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.243 RX sampler latch I even positive 1 calibration unit tune register (lane0_rx_slc_iep1_tune - lane3_rx_slc_iep1_tune)

13.4.10.2.243.1 Offset

Register	Offset
lane0_rx_slc_iep1_tune	8137h
lane1_rx_slc_iep1_tune	8537h
lane2_rx_slc_iep1_tune	8937h
lane3_rx_slc_iep1_tune	8D37h

13.4.10.2.243.2 Diagram



13.4.10.2.243.3 Fields

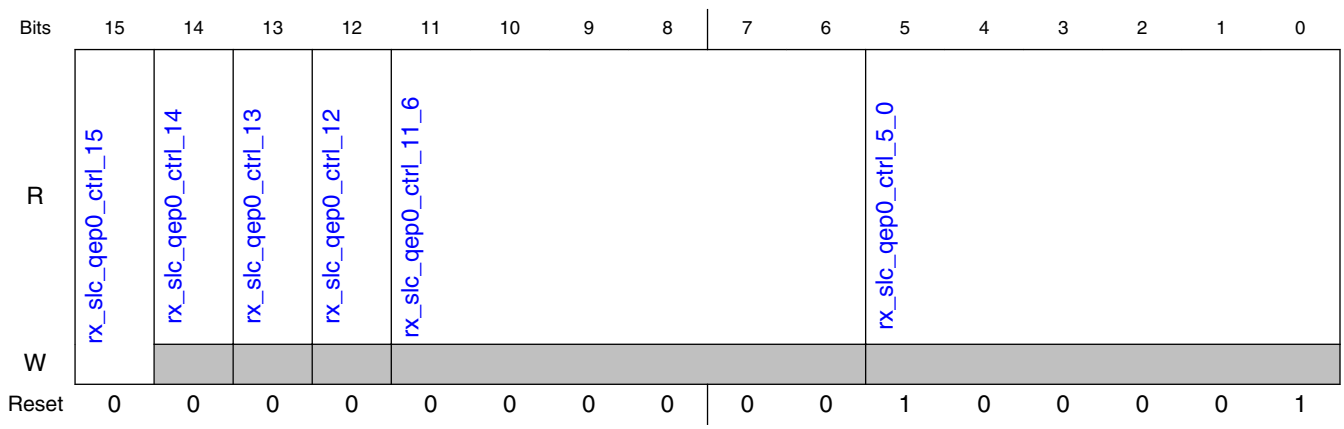
Field	Function
15-6 rx_slc_iep1_tune_15_6	Reserved
5-0 rx_slc_iep1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.244 RX sampler latch Q even positive 0 calibration unit control register (lane0_rx_slc_qep0_ctrl - lane3_rx_slc_qep0_ctrl)

13.4.10.2.244.1 Offset

Register	Offset
lane0_rx_slc_qep0_ctrl	8138h
lane1_rx_slc_qep0_ctrl	8538h
lane2_rx_slc_qep0_ctrl	8938h
lane3_rx_slc_qep0_ctrl	8D38h

13.4.10.2.244.2 Diagram



13.4.10.2.244.3 Fields

Field	Function
15 rx_slc_qep0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qep0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qep0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.

Table continues on the next page...

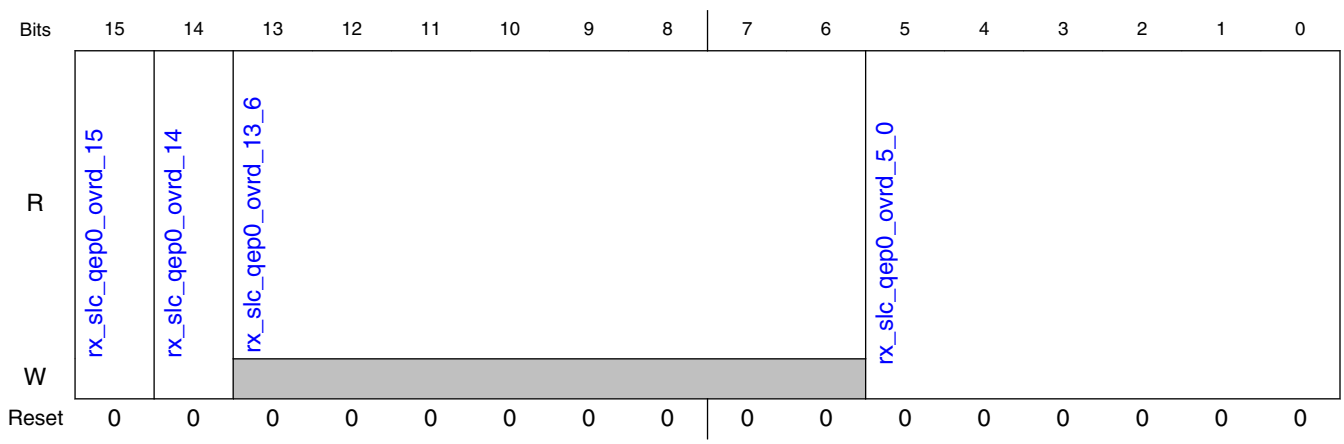
Field	Function
rx_slc_qep0_ctrl_12	
11-6 rx_slc_qep0_ctrl_11_6	Reserved
5-0 rx_slc_qep0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.245 RX sampler latch Q even positive 0 calibration unit override register (lane0_rx_slc_qep0_ovrd - lane3_rx_slc_qep0_ovrd)

13.4.10.2.245.1 Offset

Register	Offset
lane0_rx_slc_qep0_ovrd	8139h
lane1_rx_slc_qep0_ovrd	8539h
lane2_rx_slc_qep0_ovrd	8939h
lane3_rx_slc_qep0_ovrd	8D39h

13.4.10.2.245.2 Diagram



13.4.10.2.245.3 Fields

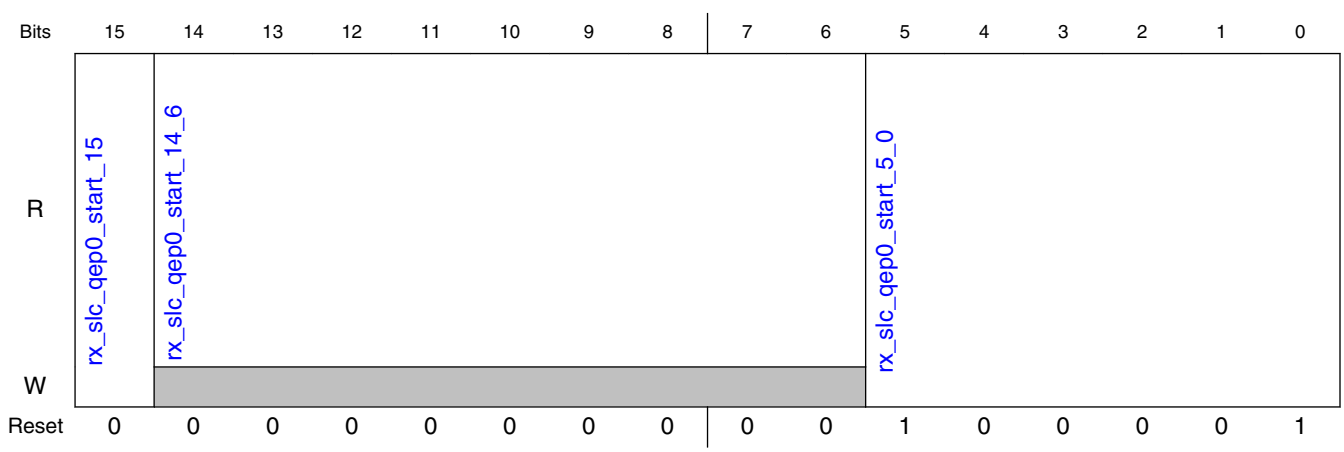
Field	Function
15 rx_slc_qep0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qep0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qep0_ovrd_13_6	Reserved
5-0 rx_slc_qep0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.246 RX sampler latch Q even positive 0 calibration unit start register (lane0_rx_slc_qep0_start - lane3_rx_slc_qep0_start)

13.4.10.2.246.1 Offset

Register	Offset
lane0_rx_slc_qep0_start	813Ah
lane1_rx_slc_qep0_start	853Ah
lane2_rx_slc_qep0_start	893Ah
lane3_rx_slc_qep0_start	8D3Ah

13.4.10.2.246.2 Diagram



13.4.10.2.246.3 Fields

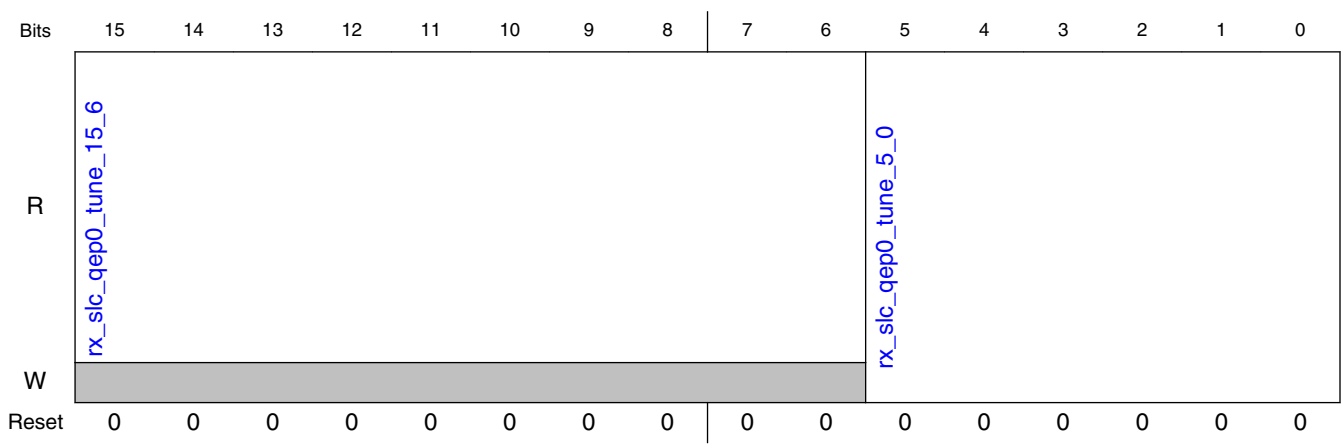
Field	Function
15 rx_slc_qep0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qep0_start_14_6	Reserved
5-0 rx_slc_qep0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.247 RX sampler latch Q even positive 0 calibration unit tune register (lane0_rx_slc_qep0_tune - lane3_rx_slc_qep0_tune)

13.4.10.2.247.1 Offset

Register	Offset
lane0_rx_slc_qep0_tune	813Bh
lane1_rx_slc_qep0_tune	853Bh
lane2_rx_slc_qep0_tune	893Bh
lane3_rx_slc_qep0_tune	8D3Bh

13.4.10.2.247.2 Diagram



13.4.10.2.247.3 Fields

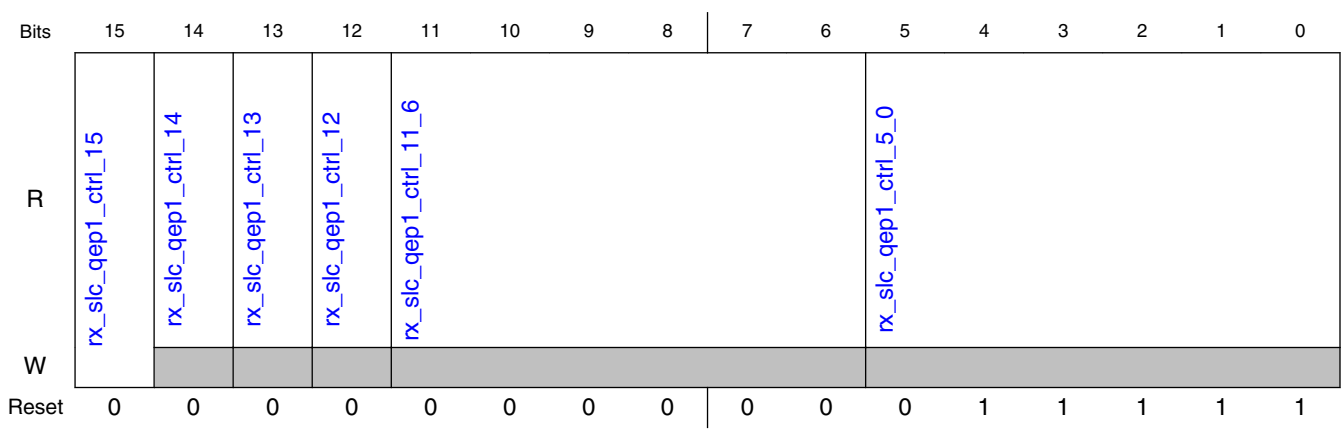
Field	Function
15-6 rx_slc_qep0_tune_15_6	Reserved
5-0 rx_slc_qep0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.248 RX sampler latch Q even positive 1 calibration unit control register (lane0_rx_slc_qep1_ctrl - lane3_rx_slc_qep1_ctrl)

13.4.10.2.248.1 Offset

Register	Offset
lane0_rx_slc_qep1_ctrl	813Ch
lane1_rx_slc_qep1_ctrl	853Ch
lane2_rx_slc_qep1_ctrl	893Ch
lane3_rx_slc_qep1_ctrl	8D3Ch

13.4.10.2.248.2 Diagram



13.4.10.2.248.3 Fields

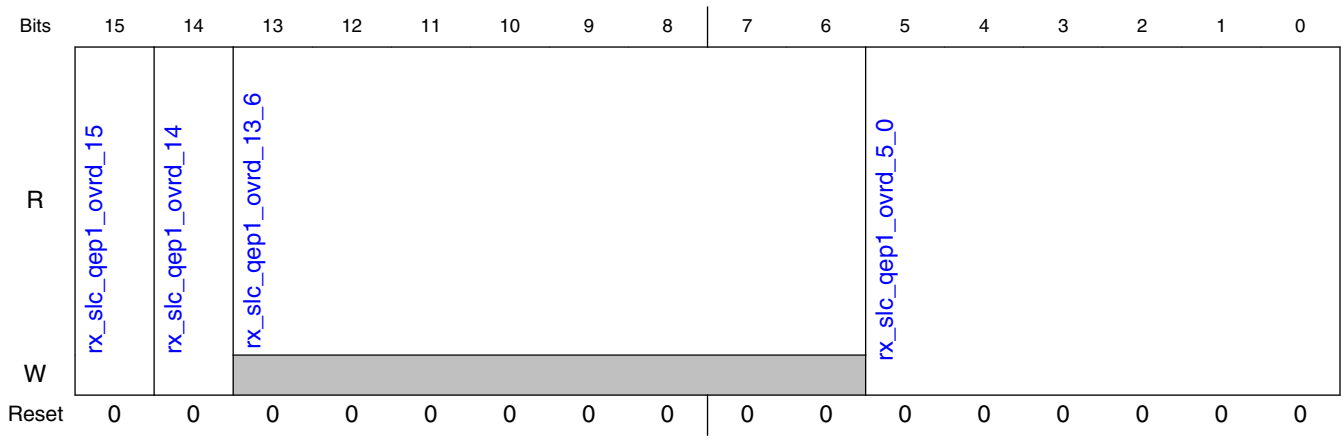
Field	Function
15 rx_slc_qep1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qep1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qep1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qep1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qep1_ctrl_11_6	Reserved
5-0 rx_slc_qep1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.249 RX sampler latch Q even positive 1 calibration unit override register (lane0_rx_slc_qep1_ovrd - lane3_rx_slc_qep1_ovrd)

13.4.10.2.249.1 Offset

Register	Offset
lane0_rx_slc_qep1_ovrd	813Dh
lane1_rx_slc_qep1_ovrd	853Dh
lane2_rx_slc_qep1_ovrd	893Dh
lane3_rx_slc_qep1_ovrd	8D3Dh

13.4.10.2.249.2 Diagram



13.4.10.2.249.3 Fields

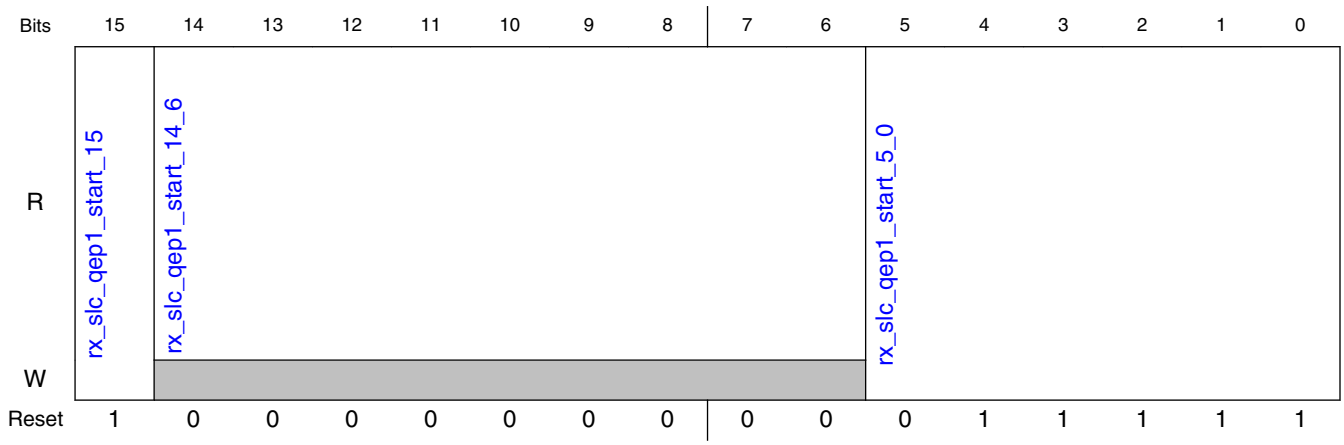
Field	Function
15 rx_slc_qep1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qep1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qep1_ovrd_13_6	Reserved
5-0 rx_slc_qep1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.250 RX sampler latch Q even positive 1 calibration unit start register (lane0_rx_slc_qep1_start - lane3_rx_slc_qep1_start)

13.4.10.2.250.1 Offset

Register	Offset
lane0_rx_slc_qep1_start	813Eh
lane1_rx_slc_qep1_start	853Eh
lane2_rx_slc_qep1_start	893Eh
lane3_rx_slc_qep1_start	8D3Eh

13.4.10.2.250.2 Diagram



13.4.10.2.250.3 Fields

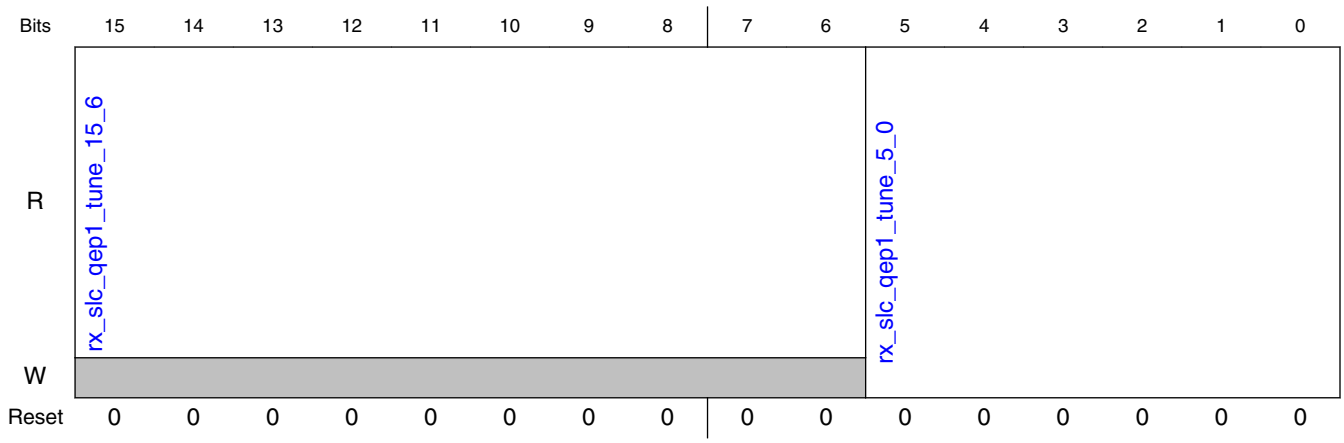
Field	Function
15 rx_slc_qep1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qep1_start_14_6	Reserved
5-0 rx_slc_qep1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.251 RX sampler latch Q even positive 1 calibration unit tune register (lane0_rx_slc_qep1_tune - lane3_rx_slc_qep1_tune)

13.4.10.2.251.1 Offset

Register	Offset
lane0_rx_slc_qep1_tune	813Fh
lane1_rx_slc_qep1_tune	853Fh
lane2_rx_slc_qep1_tune	893Fh
lane3_rx_slc_qep1_tune	8D3Fh

13.4.10.2.251.2 Diagram



13.4.10.2.251.3 Fields

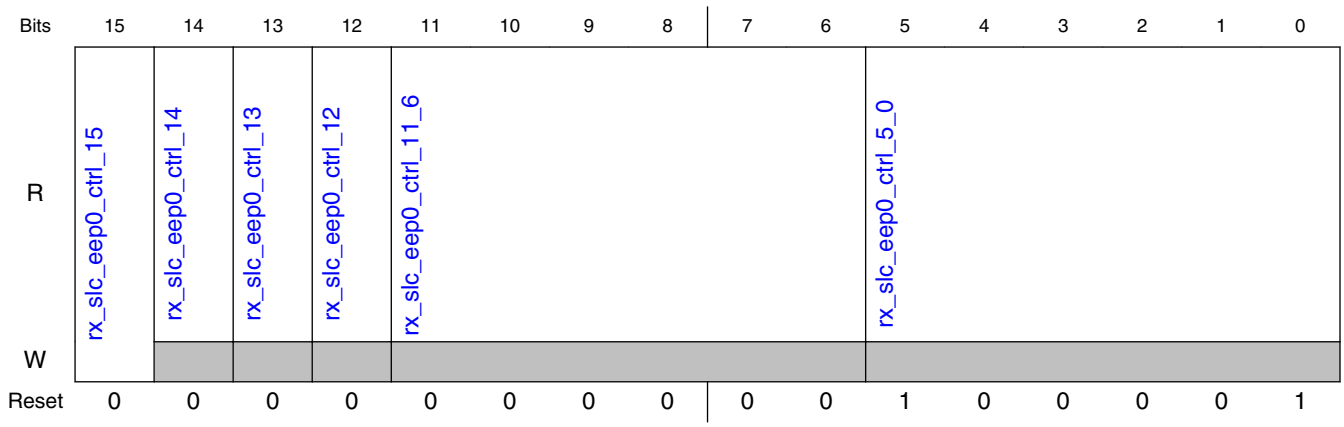
Field	Function
15-6 rx_slc_qep1_tune_15_6	Reserved
5-0 rx_slc_qep1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.252 RX sampler latch E even positive 0 calibration unit control register (lane0_rx_slc_eep0_ctrl - lane3_rx_slc_eep0_ctrl)

13.4.10.2.252.1 Offset

Register	Offset
lane0_rx_slc_eep0_ctrl	8140h
lane1_rx_slc_eep0_ctrl	8540h
lane2_rx_slc_eep0_ctrl	8940h
lane3_rx_slc_eep0_ctrl	8D40h

13.4.10.2.252.2 Diagram



13.4.10.2.252.3 Fields

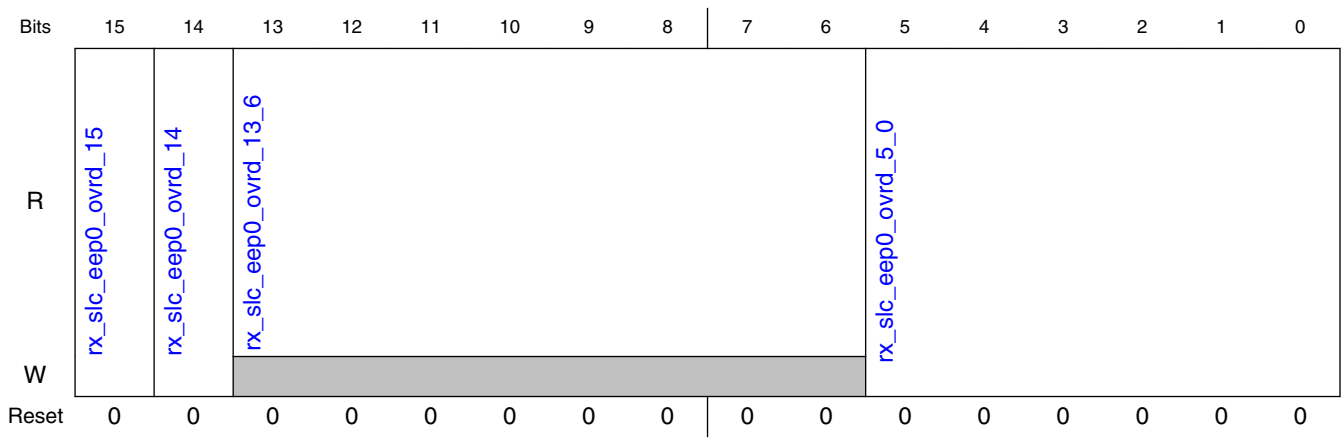
Field	Function
15 rx_slc_eep0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_eep0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_eep0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eep0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eep0_ctrl_11_6	Reserved
5-0 rx_slc_eep0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.253 RX sampler latch E even positive 0 calibration unit override register (lane0_rx_slc_eep0_ovrd - lane3_rx_slc_eep0_ovrd)

13.4.10.2.253.1 Offset

Register	Offset
lane0_rx_slc_eep0_ovrd	8141h
lane1_rx_slc_eep0_ovrd	8541h
lane2_rx_slc_eep0_ovrd	8941h
lane3_rx_slc_eep0_ovrd	8D41h

13.4.10.2.253.2 Diagram



13.4.10.2.253.3 Fields

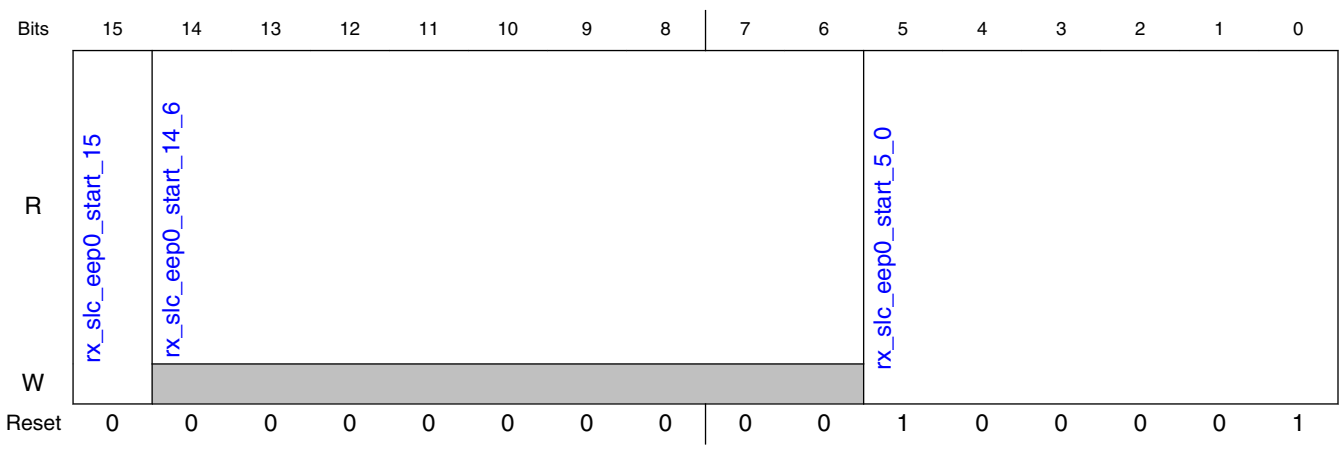
Field	Function
15 rx_slc_eep0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eep0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_eep0_ovrd_13_6	Reserved
5-0 rx_slc_eep0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.254 RX sampler latch E even positive 0 calibration unit start register (lane0_rx_slc_eep0_start - lane3_rx_slc_eep0_start)

13.4.10.2.254.1 Offset

Register	Offset
lane0_rx_slc_eep0_start	8142h
lane1_rx_slc_eep0_start	8542h
lane2_rx_slc_eep0_start	8942h
lane3_rx_slc_eep0_start	8D42h

13.4.10.2.254.2 Diagram



13.4.10.2.254.3 Fields

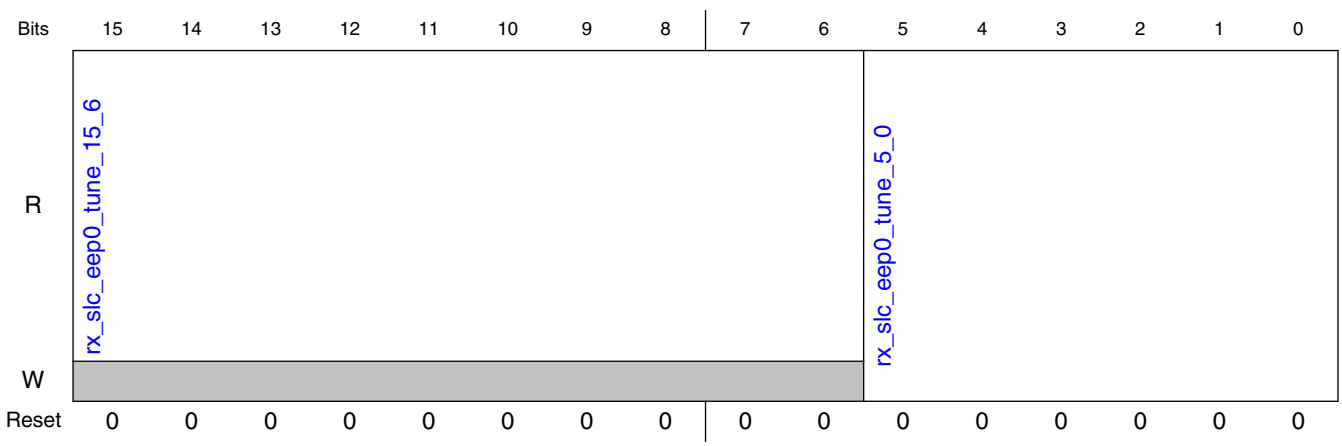
Field	Function
15 rx_slc_eep0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_eep0_start_14_6	Reserved
5-0 rx_slc_eep0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.255 RX sampler latch E even positive 0 calibration unit tune register (lane0_rx_slc_eep0_tune - lane3_rx_slc_eep0_tune)

13.4.10.2.255.1 Offset

Register	Offset
lane0_rx_slc_eep0_tune	8143h
lane1_rx_slc_eep0_tune	8543h
lane2_rx_slc_eep0_tune	8943h
lane3_rx_slc_eep0_tune	8D43h

13.4.10.2.255.2 Diagram



13.4.10.2.255.3 Fields

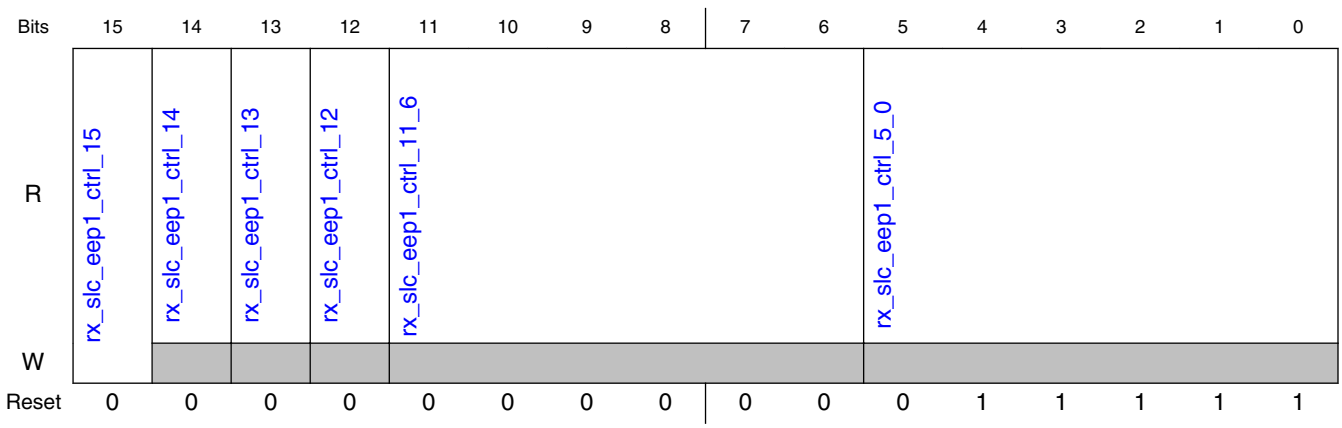
Field	Function
15-6 rx_slc_eep0_tune_15_6	Reserved
5-0 rx_slc_eep0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.256 RX sampler latch E even positive 1 calibration unit control register (lane0_rx_slc_eep1_ctrl - lane3_rx_slc_eep1_ctrl)

13.4.10.2.256.1 Offset

Register	Offset
lane0_rx_slc_eep1_ctrl	8144h
lane1_rx_slc_eep1_ctrl	8544h
lane2_rx_slc_eep1_ctrl	8944h
lane3_rx_slc_eep1_ctrl	8D44h

13.4.10.2.256.2 Diagram



13.4.10.2.256.3 Fields

Field	Function
15 rx_slc_eep1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_eep1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_eep1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_eep1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_eep1_ctrl_11_6	Reserved
5-0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

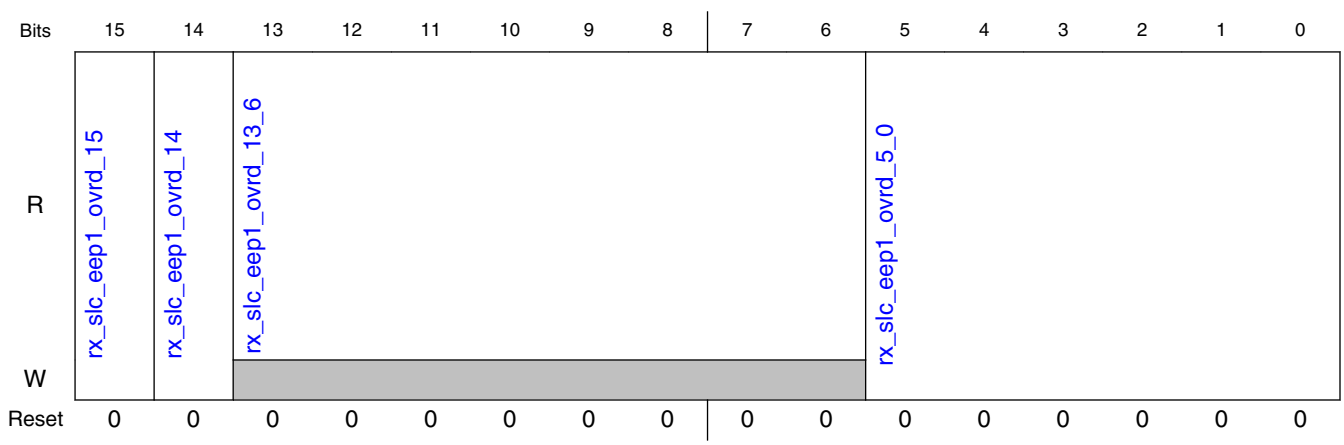
Field	Function
rx_slc_eep1_ctrl_5_0	

13.4.10.2.257 RX sampler latch E even positive 1 calibration unit override register (lane0_rx_slc_eep1_ovrd - lane3_rx_slc_eep1_ovrd)

13.4.10.2.257.1 Offset

Register	Offset
lane0_rx_slc_eep1_ovrd	8145h
lane1_rx_slc_eep1_ovrd	8545h
lane2_rx_slc_eep1_ovrd	8945h
lane3_rx_slc_eep1_ovrd	8D45h

13.4.10.2.257.2 Diagram



13.4.10.2.257.3 Fields

Field	Function
15 rx_slc_eep1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_eep1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the

Table continues on the next page...

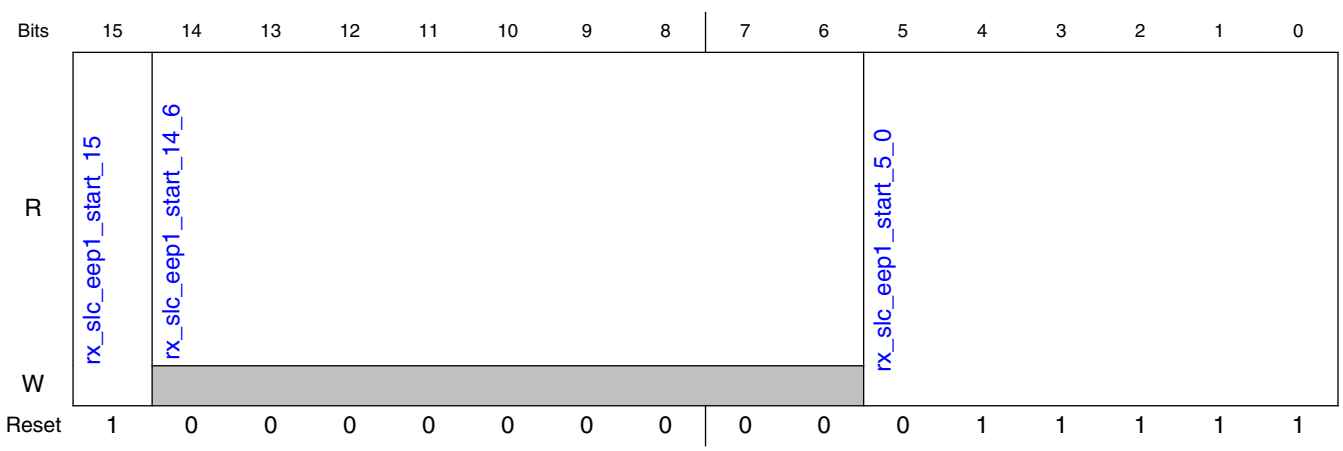
Field	Function
13-6 rx_slc_eep1_ovrd_13_6	Reserved
5-0 rx_slc_eep1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.258 RX sampler latch E even positive 1 calibration unit start register (lane0_rx_slc_eep1_start - lane3_rx_slc_eep1_start)

13.4.10.2.258.1 Offset

Register	Offset
lane0_rx_slc_eep1_start	8146h
lane1_rx_slc_eep1_start	8546h
lane2_rx_slc_eep1_start	8946h
lane3_rx_slc_eep1_start	8D46h

13.4.10.2.258.2 Diagram



13.4.10.2.258.3 Fields

Field	Function
15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.

Table continues on the next page...

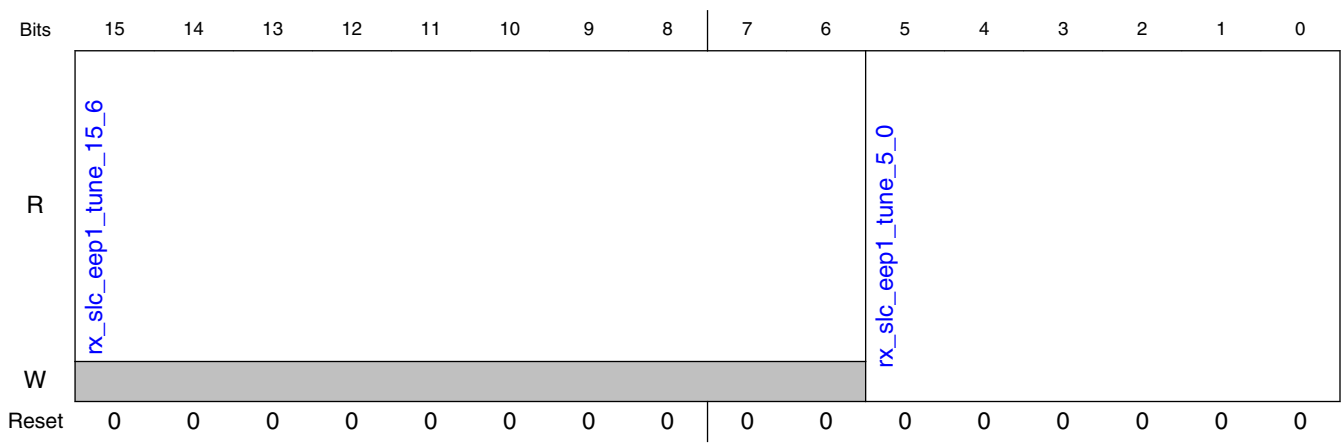
Field	Function
rx_slc_eep1_start_15	
14-6 rx_slc_eep1_start_14_6	Reserved
5-0 rx_slc_eep1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.259 RX sampler latch E even positive 1 calibration unit tune register (lane0_rx_slc_eep1_tune - lane3_rx_slc_eep1_tune)

13.4.10.2.259.1 Offset

Register	Offset
lane0_rx_slc_eep1_tune	8147h
lane1_rx_slc_eep1_tune	8547h
lane2_rx_slc_eep1_tune	8947h
lane3_rx_slc_eep1_tune	8D47h

13.4.10.2.259.2 Diagram



13.4.10.2.259.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

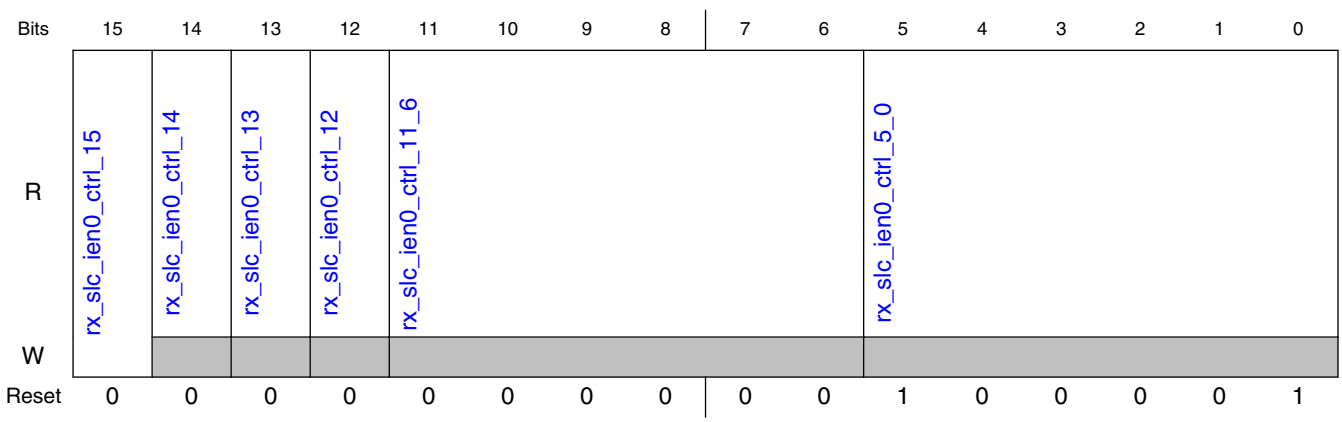
Field	Function
rx_slc_eep1_tune_15_6	
5-0 rx_slc_eep1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.260 RX sampler latch I even negative 0 calibration unit control register (lane0_rx_slc_ien0_ctrl - lane3_rx_slc_ien0_ctrl)

13.4.10.2.260.1 Offset

Register	Offset
lane0_rx_slc_ien0_ctrl	8148h
lane1_rx_slc_ien0_ctrl	8548h
lane2_rx_slc_ien0_ctrl	8948h
lane3_rx_slc_ien0_ctrl	8D48h

13.4.10.2.260.2 Diagram



13.4.10.2.260.3 Fields

Field	Function
15 rx_slc_ien0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by

Table continues on the next page...

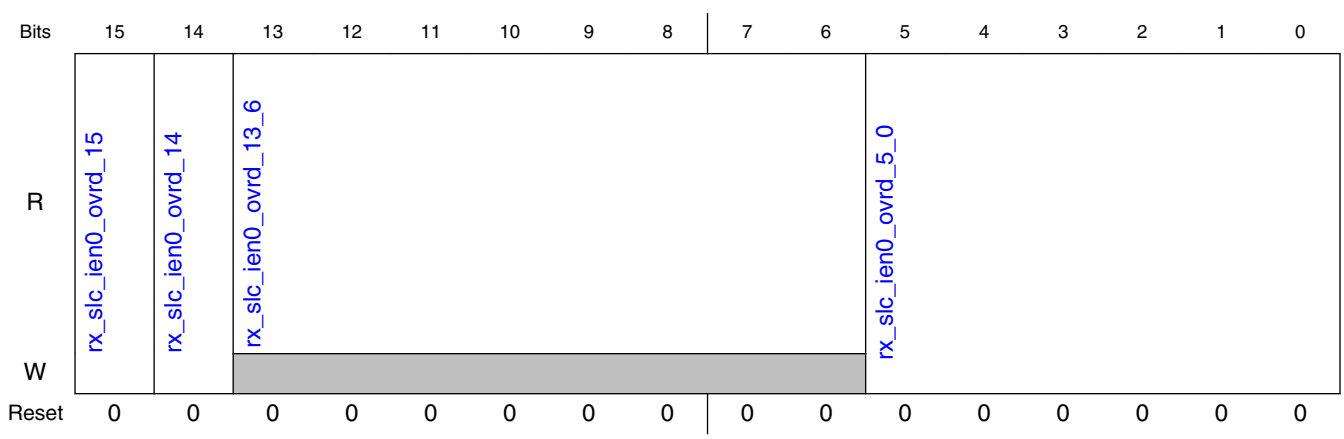
Field	Function
rx_slc_ien0_ctrl_14	
13 rx_slc_ien0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_ien0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_ien0_ctrl_11_6	Reserved
5-0 rx_slc_ien0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.261 RX sampler latch I even negative 0 calibration unit override register (lane0_rx_slc_ien0_ovrd - lane3_rx_slc_ien0_ovrd)

13.4.10.2.261.1 Offset

Register	Offset
lane0_rx_slc_ien0_ovrd	8149h
lane1_rx_slc_ien0_ovrd	8549h
lane2_rx_slc_ien0_ovrd	8949h
lane3_rx_slc_ien0_ovrd	8D49h

13.4.10.2.261.2 Diagram



13.4.10.2.261.3 Fields

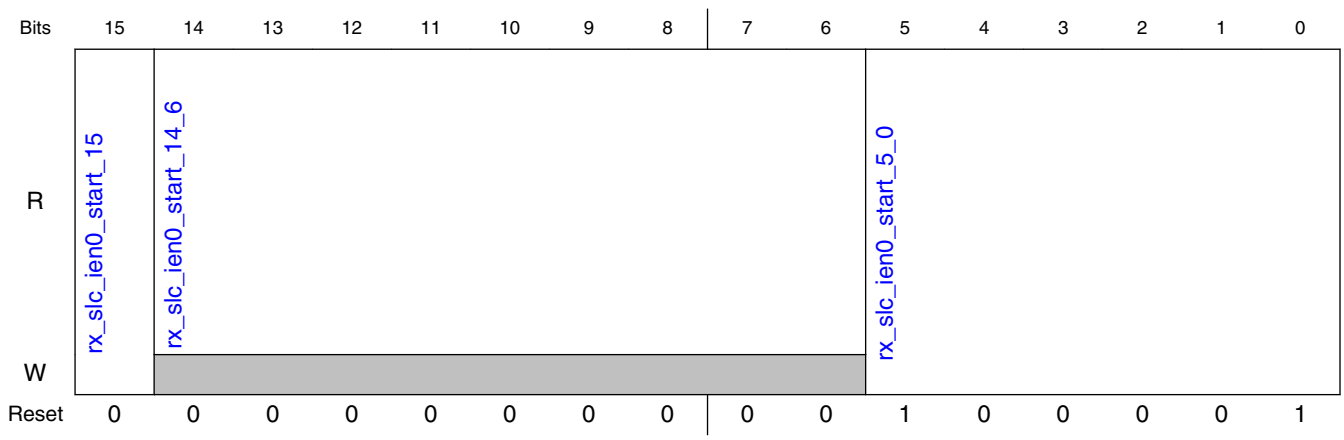
Field	Function
15 rx_slc_ien0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_ien0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_ien0_ovrd_13_6	Reserved
5-0 rx_slc_ien0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.262 RX sampler latch I even negative 0 calibration unit start register (lane0_rx_slc_ien0_start - lane3_rx_slc_ien0_start)

13.4.10.2.262.1 Offset

Register	Offset
lane0_rx_slc_ien0_start	814Ah
lane1_rx_slc_ien0_start	854Ah
lane2_rx_slc_ien0_start	894Ah
lane3_rx_slc_ien0_start	8D4Ah

13.4.10.2.262.2 Diagram



13.4.10.2.262.3 Fields

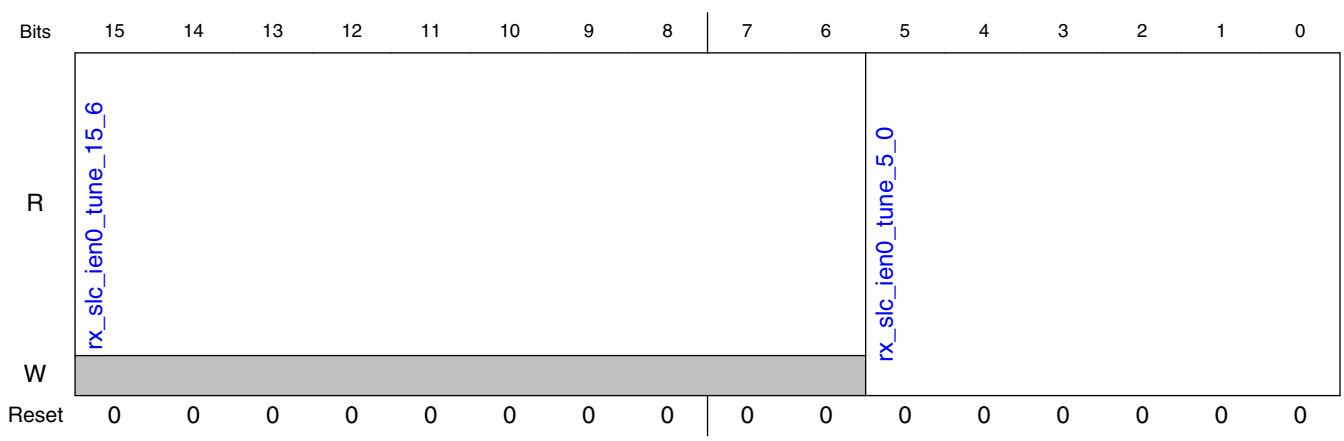
Field	Function
15 rx_slc_ien0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_ien0_start_14_6	Reserved
5-0 rx_slc_ien0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.263 RX sampler latch I even negative 0 calibration unit tune register (lane0_rx_slc_ien0_tune - lane3_rx_slc_ien0_tune)

13.4.10.2.263.1 Offset

Register	Offset
lane0_rx_slc_ien0_tune	814Bh
lane1_rx_slc_ien0_tune	854Bh
lane2_rx_slc_ien0_tune	894Bh
lane3_rx_slc_ien0_tune	8D4Bh

13.4.10.2.263.2 Diagram



13.4.10.2.263.3 Fields

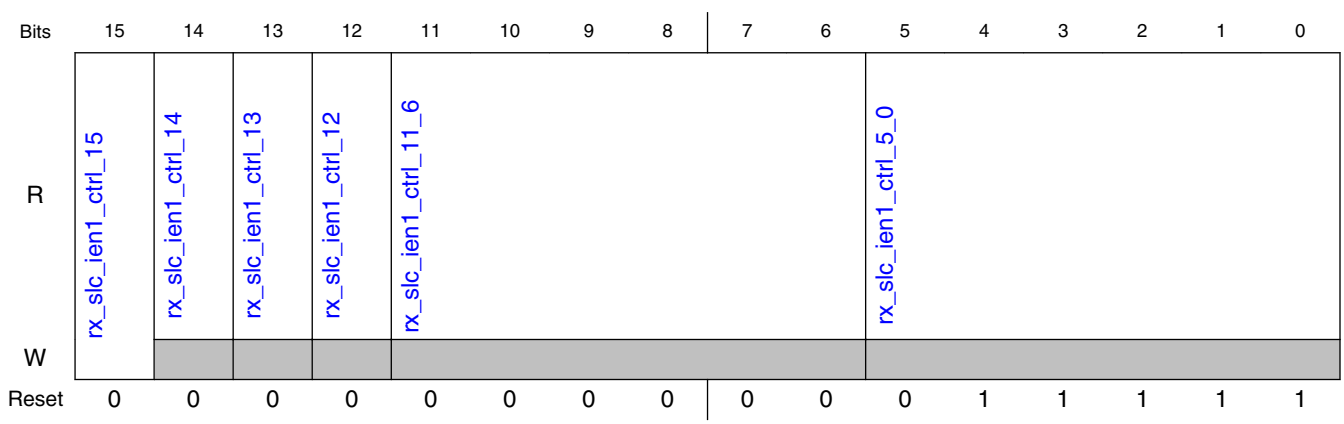
Field	Function
15-6 rx_slc_ien0_tune_15_6	Reserved
5-0 rx_slc_ien0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.264 RX sampler latch I even negative 1 calibration unit control register (lane0_rx_slc_ien1_ctrl - lane3_rx_slc_ien1_ctrl)

13.4.10.2.264.1 Offset

Register	Offset
lane0_rx_slc_ien1_ctrl	814Ch
lane1_rx_slc_ien1_ctrl	854Ch
lane2_rx_slc_ien1_ctrl	894Ch
lane3_rx_slc_ien1_ctrl	8D4Ch

13.4.10.2.264.2 Diagram



13.4.10.2.264.3 Fields

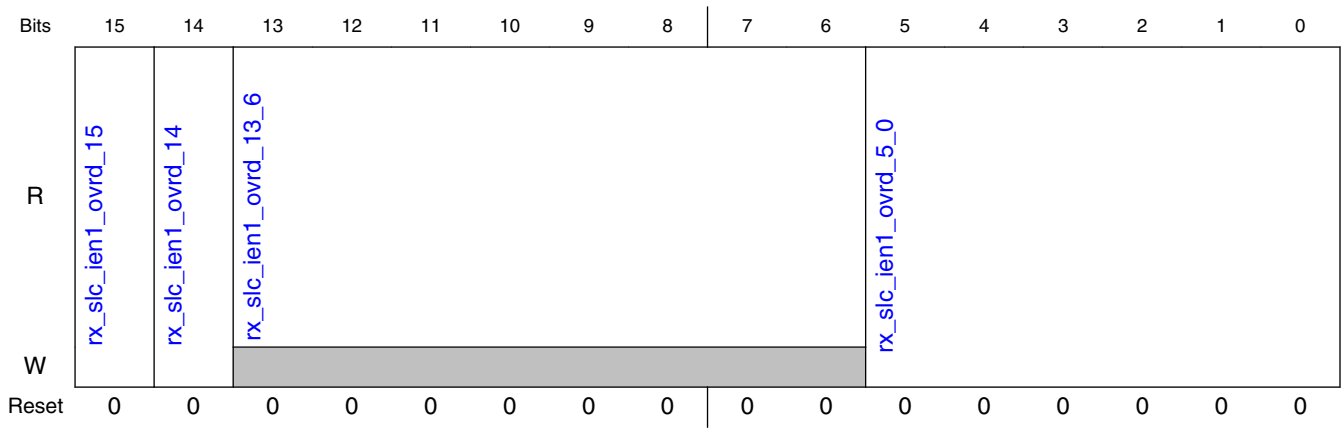
Field	Function
15 rx_slc_ien1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_ien1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_ien1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_ien1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_ien1_ctrl_11_6	Reserved
5-0 rx_slc_ien1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.265 RX sampler latch I even negative 1 calibration unit override register (lane0_rx_slc_ien1_ovrd - lane3_rx_slc_ien1_ovrd)

13.4.10.2.265.1 Offset

Register	Offset
lane0_rx_slc_ien1_ovrd	814Dh
lane1_rx_slc_ien1_ovrd	854Dh
lane2_rx_slc_ien1_ovrd	894Dh
lane3_rx_slc_ien1_ovrd	8D4Dh

13.4.10.2.265.2 Diagram



13.4.10.2.265.3 Fields

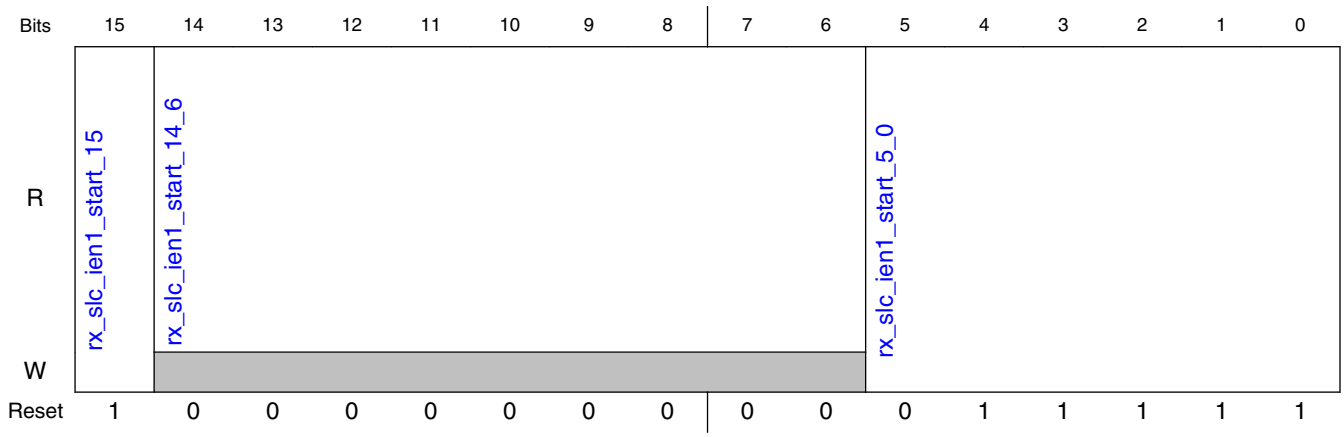
Field	Function
15 rx_slc_ien1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_ien1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_ien1_ovrd_13_6	Reserved
5-0 rx_slc_ien1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.266 RX sampler latch I even negative 1 calibration unit start register (lane0_rx_slc_ien1_start - lane3_rx_slc_ien1_start)

13.4.10.2.266.1 Offset

Register	Offset
lane0_rx_slc_ien1_start	814Eh
lane1_rx_slc_ien1_start	854Eh
lane2_rx_slc_ien1_start	894Eh
lane3_rx_slc_ien1_start	8D4Eh

13.4.10.2.266.2 Diagram



13.4.10.2.266.3 Fields

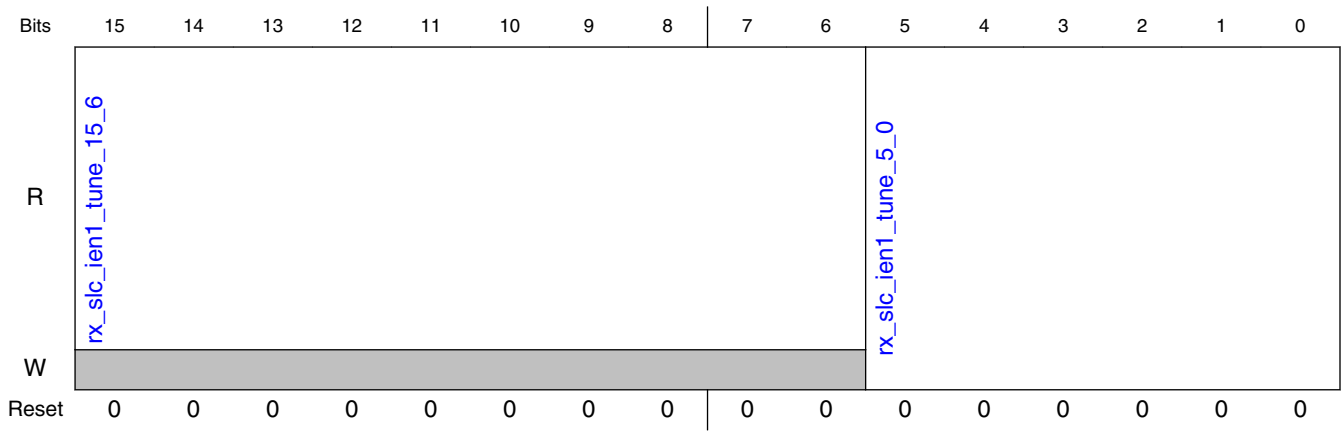
Field	Function
15 rx_slc_ien1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_ien1_start_14_6	Reserved
5-0 rx_slc_ien1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.267 RX sampler latch I even negative 1 calibration unit tune register (lane0_rx_slc_ien1_tune - lane3_rx_slc_ien1_tune)

13.4.10.2.267.1 Offset

Register	Offset
lane0_rx_slc_ien1_tune	814Fh
lane1_rx_slc_ien1_tune	854Fh
lane2_rx_slc_ien1_tune	894Fh
lane3_rx_slc_ien1_tune	8D4Fh

13.4.10.2.267.2 Diagram



13.4.10.2.267.3 Fields

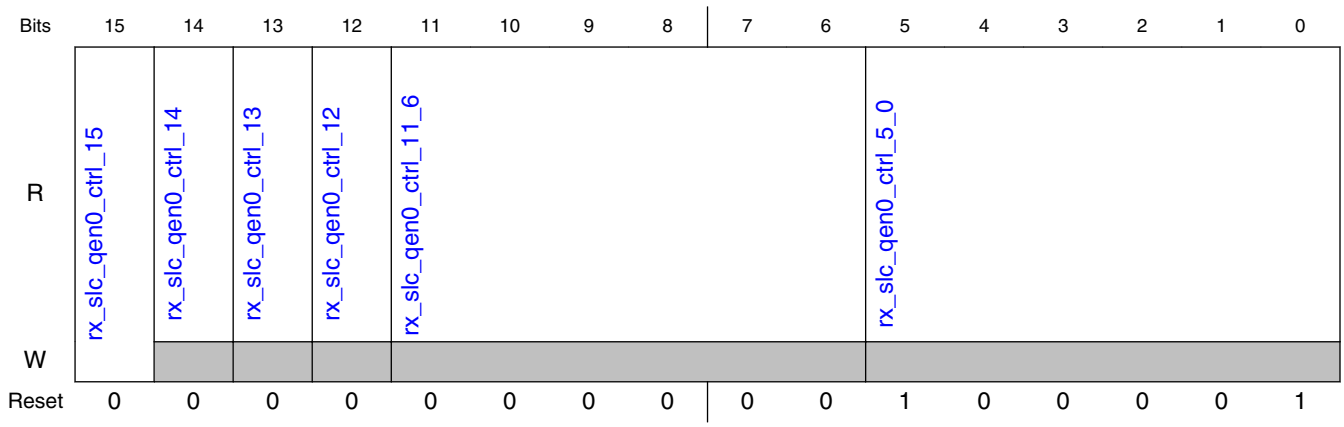
Field	Function
15-6 rx_slc_ien1_tune_15_6	Reserved
5-0 rx_slc_ien1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.268 RX sampler latch Q even negative 0 calibration unit control register (lane0_rx_slc_qen0_ctrl - lane3_rx_slc_qen0_ctrl)

13.4.10.2.268.1 Offset

Register	Offset
lane0_rx_slc_qen0_ctrl	8150h
lane1_rx_slc_qen0_ctrl	8550h
lane2_rx_slc_qen0_ctrl	8950h
lane3_rx_slc_qen0_ctrl	8D50h

13.4.10.2.268.2 Diagram



13.4.10.2.268.3 Fields

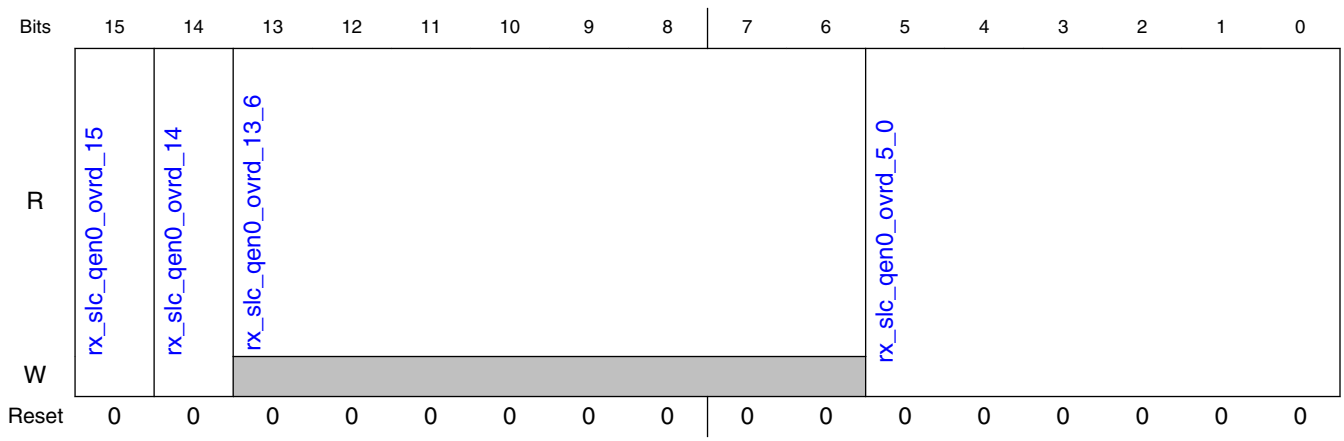
Field	Function
15 rx_slc_qen0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qen0_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qen0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qen0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qen0_ctrl_11_6	Reserved
5-0 rx_slc_qen0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.269 RX sampler latch Q even negative 0 calibration unit override register (lane0_rx_slc_qen0_ovrd - lane3_rx_slc_qen0_ovrd)

13.4.10.2.269.1 Offset

Register	Offset
lane0_rx_slc_qen0_ovrd	8151h
lane1_rx_slc_qen0_ovrd	8551h
lane2_rx_slc_qen0_ovrd	8951h
lane3_rx_slc_qen0_ovrd	8D51h

13.4.10.2.269.2 Diagram



13.4.10.2.269.3 Fields

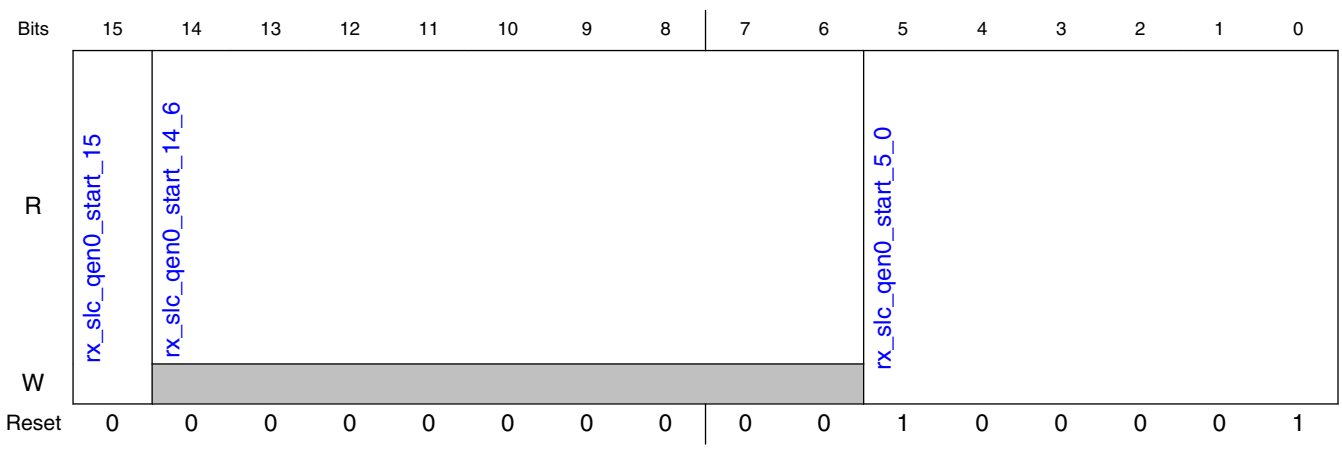
Field	Function
15 rx_slc_qen0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qen0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_qen0_ovrd_13_6	Reserved
5-0 rx_slc_qen0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.270 RX sampler latch Q even negative 0 calibration unit start register (lane0_rx_slc_qen0_start - lane3_rx_slc_qen0_start)

13.4.10.2.270.1 Offset

Register	Offset
lane0_rx_slc_qen0_start	8152h
lane1_rx_slc_qen0_start	8552h
lane2_rx_slc_qen0_start	8952h
lane3_rx_slc_qen0_start	8D52h

13.4.10.2.270.2 Diagram



13.4.10.2.270.3 Fields

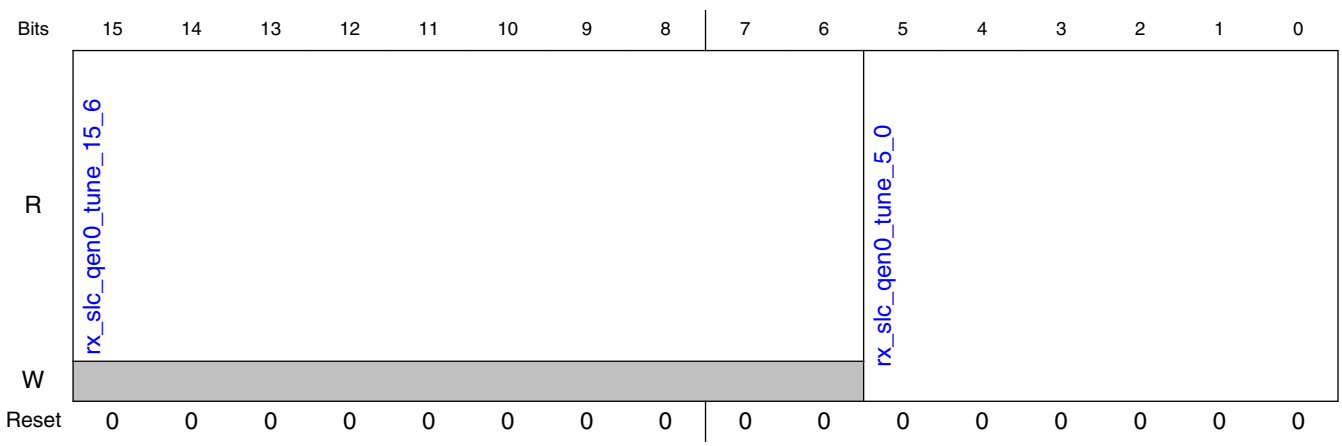
Field	Function
15 rx_slc_qen0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_qen0_start_14_6	Reserved
5-0 rx_slc_qen0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.271 RX sampler latch Q even negative 0 calibration unit tune register (lane0_rx_slc_qen0_tune - lane3_rx_slc_qen0_tune)

13.4.10.2.271.1 Offset

Register	Offset
lane0_rx_slc_qen0_tune	8153h
lane1_rx_slc_qen0_tune	8553h
lane2_rx_slc_qen0_tune	8953h
lane3_rx_slc_qen0_tune	8D53h

13.4.10.2.271.2 Diagram



13.4.10.2.271.3 Fields

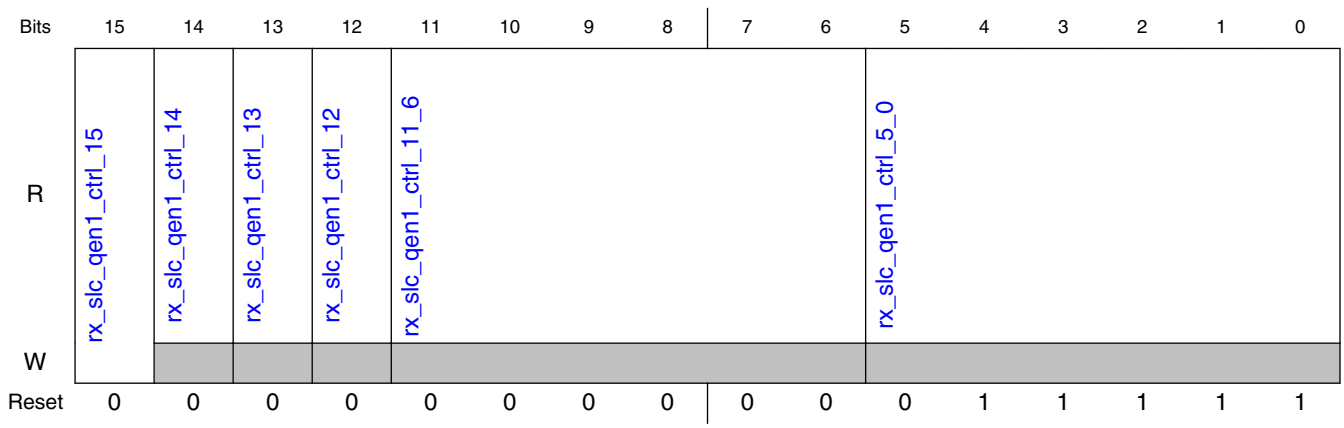
Field	Function
15-6 rx_slc_qen0_tune_15_6	Reserved
5-0 rx_slc_qen0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.272 RX sampler latch Q even negative 1 calibration unit control register (lane0_rx_slc_qen1_ctrl - lane3_rx_slc_qen1_ctrl)

13.4.10.2.272.1 Offset

Register	Offset
lane0_rx_slc_qen1_ctrl	8154h
lane1_rx_slc_qen1_ctrl	8554h
lane2_rx_slc_qen1_ctrl	8954h
lane3_rx_slc_qen1_ctrl	8D54h

13.4.10.2.272.2 Diagram



13.4.10.2.272.3 Fields

Field	Function
15 rx_slc_qen1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_qen1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_qen1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_qen1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_qen1_ctrl_11_6	Reserved
5-0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

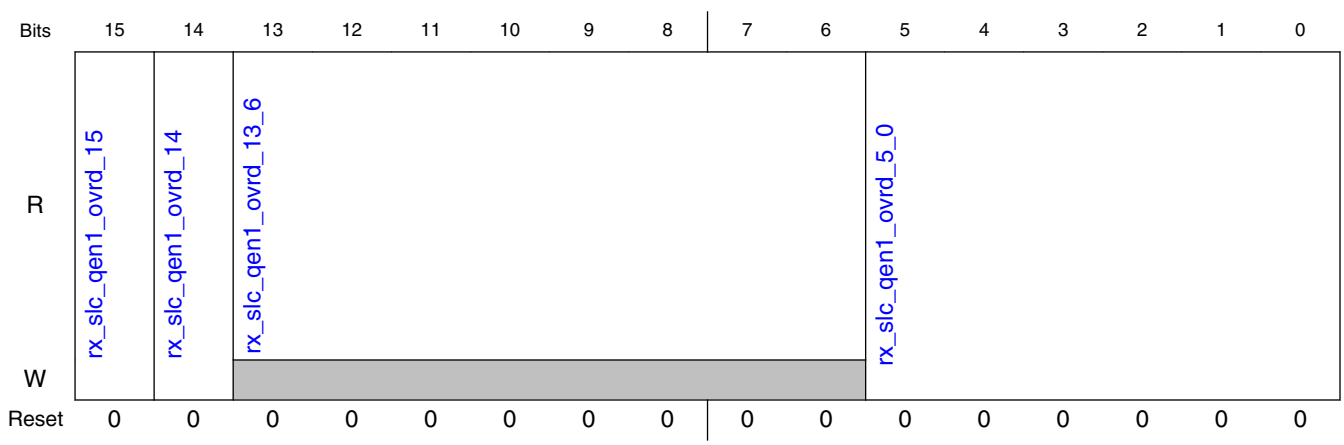
Field	Function
rx_slc_qen1_ctrl_5_0	

13.4.10.2.273 RX sampler latch Q even negative 1 calibration unit override register (lane0_rx_slc_qen1_ovrd - lane3_rx_slc_qen1_ovrd)

13.4.10.2.273.1 Offset

Register	Offset
lane0_rx_slc_qen1_ovrd	8155h
lane1_rx_slc_qen1_ovrd	8555h
lane2_rx_slc_qen1_ovrd	8955h
lane3_rx_slc_qen1_ovrd	8D55h

13.4.10.2.273.2 Diagram



13.4.10.2.273.3 Fields

Field	Function
15 rx_slc_qen1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_qen1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the

Table continues on the next page...

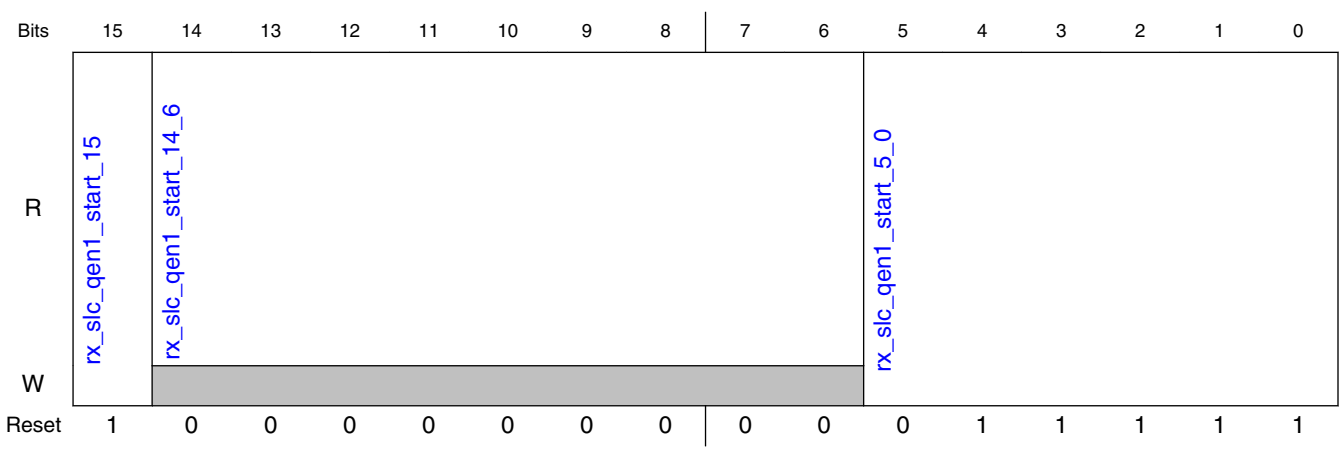
Field	Function
13-6 rx_slc_qen1_ovrd_13_6	Reserved
5-0 rx_slc_qen1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.274 RX sampler latch Q even negative 1 calibration unit start register (lane0_rx_slc_qen1_start - lane3_rx_slc_qen1_start)

13.4.10.2.274.1 Offset

Register	Offset
lane0_rx_slc_qen1_start	8156h
lane1_rx_slc_qen1_start	8556h
lane2_rx_slc_qen1_start	8956h
lane3_rx_slc_qen1_start	8D56h

13.4.10.2.274.2 Diagram



13.4.10.2.274.3 Fields

Field	Function
15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.

Table continues on the next page...

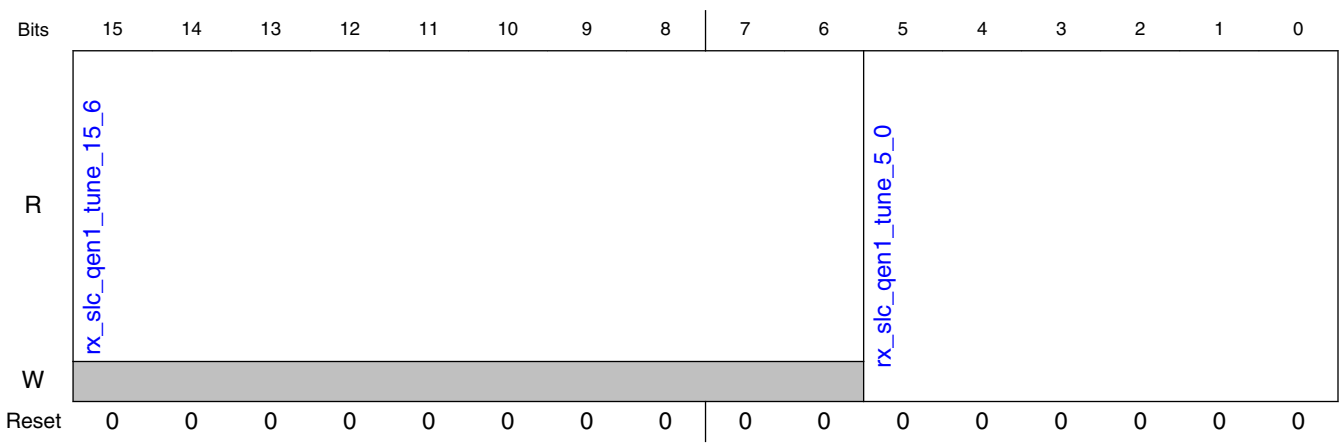
Field	Function
rx_slc_qen1_start_15	
14-6 rx_slc_qen1_start_14_6	Reserved
5-0 rx_slc_qen1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.275 RX sampler latch Q even negative 1 calibration unit tune register (lane0_rx_slc_qen1_tune - lane3_rx_slc_qen1_tune)

13.4.10.2.275.1 Offset

Register	Offset
lane0_rx_slc_qen1_tune	8157h
lane1_rx_slc_qen1_tune	8557h
lane2_rx_slc_qen1_tune	8957h
lane3_rx_slc_qen1_tune	8D57h

13.4.10.2.275.2 Diagram



13.4.10.2.275.3 Fields

Field	Function
15-6	Reserved

Table continues on the next page...

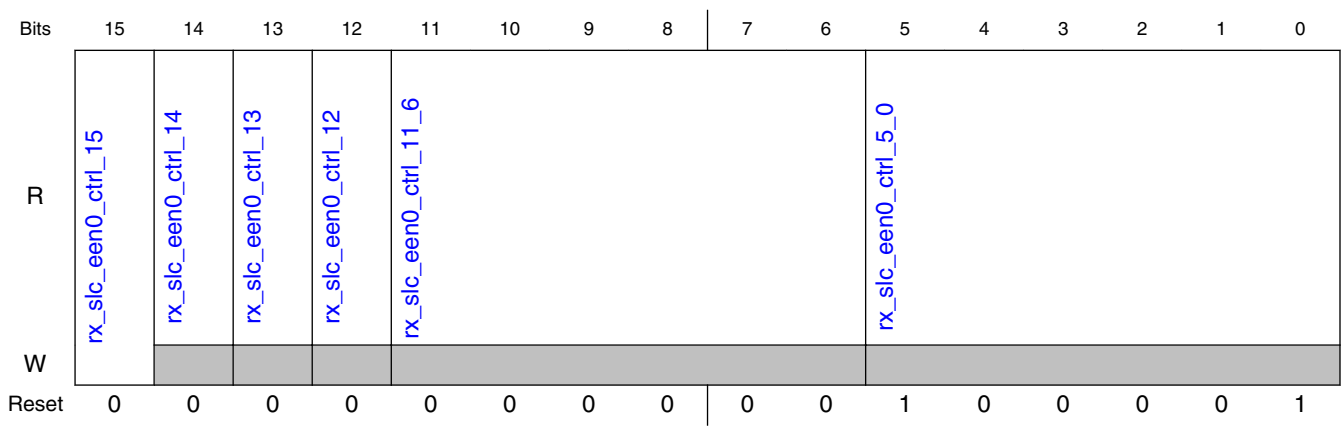
Field	Function
rx_slc_qen1_tune_15_6	
5-0 rx_slc_qen1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.276 RX sampler latch E even negative 0 calibration unit control register (lane0_rx_slc_een0_ctrl - lane3_rx_slc_een0_ctrl)

13.4.10.2.276.1 Offset

Register	Offset
lane0_rx_slc_een0_ctrl	8158h
lane1_rx_slc_een0_ctrl	8558h
lane2_rx_slc_een0_ctrl	8958h
lane3_rx_slc_een0_ctrl	8D58h

13.4.10.2.276.2 Diagram



13.4.10.2.276.3 Fields

Field	Function
15 rx_slc_een0_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by

Table continues on the next page...

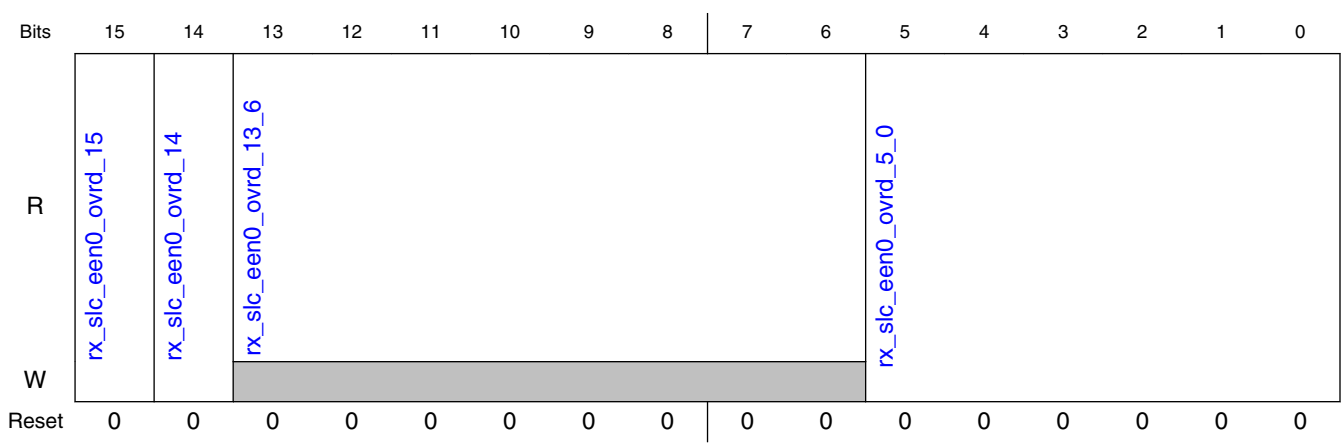
Field	Function
rx_slc_een0_ctrl_14	
13 rx_slc_een0_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_een0_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_een0_ctrl_11_6	Reserved
5-0 rx_slc_een0_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.277 RX sampler latch E even negative 0 calibration unit override register (lane0_rx_slc_een0_ovrd - lane3_rx_slc_een0_ovrd)

13.4.10.2.277.1 Offset

Register	Offset
lane0_rx_slc_een0_ovrd	8159h
lane1_rx_slc_een0_ovrd	8559h
lane2_rx_slc_een0_ovrd	8959h
lane3_rx_slc_een0_ovrd	8D59h

13.4.10.2.277.2 Diagram



13.4.10.2.277.3 Fields

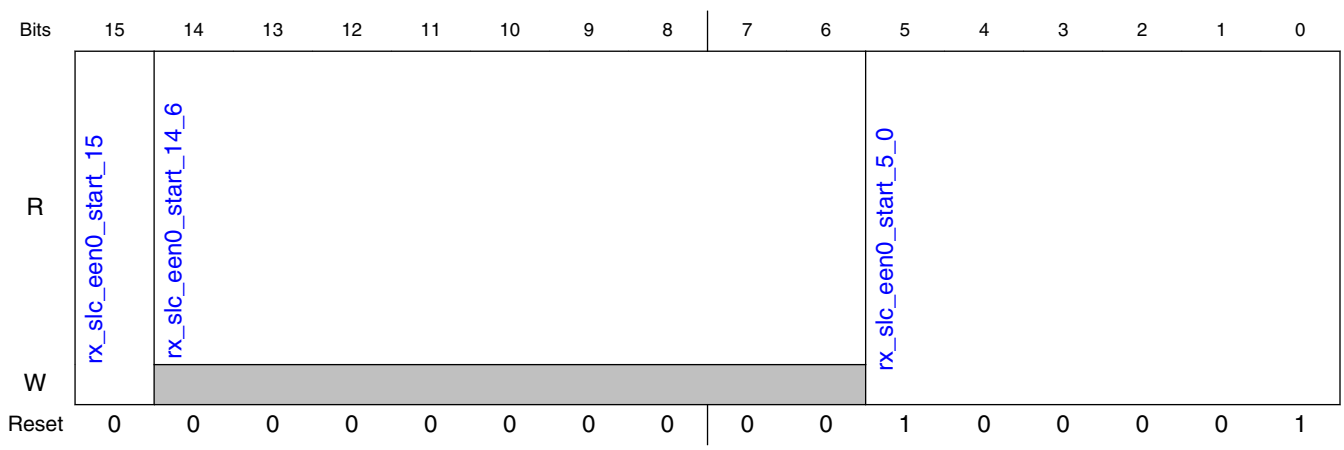
Field	Function
15 rx_slc_een0_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_een0_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_een0_ovrd_13_6	Reserved
5-0 rx_slc_een0_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.278 RX sampler latch E even negative 0 calibration unit start register (lane0_rx_slc_een0_start - lane3_rx_slc_een0_start)

13.4.10.2.278.1 Offset

Register	Offset
lane0_rx_slc_een0_start	815Ah
lane1_rx_slc_een0_start	855Ah
lane2_rx_slc_een0_start	895Ah
lane3_rx_slc_een0_start	8D5Ah

13.4.10.2.278.2 Diagram



13.4.10.2.278.3 Fields

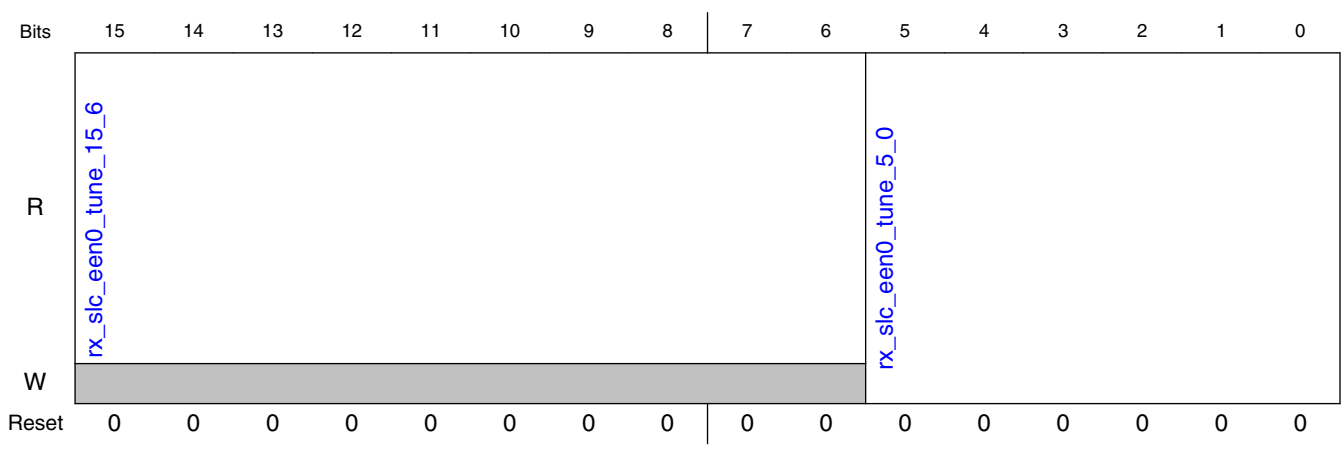
Field	Function
15 rx_slc_een0_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_een0_start_14_6	Reserved
5-0 rx_slc_een0_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.279 RX sampler latch E even negative 0 calibration unit tune register (lane0_rx_slc_een0_tune - lane3_rx_slc_een0_tune)

13.4.10.2.279.1 Offset

Register	Offset
lane0_rx_slc_een0_tune	815Bh
lane1_rx_slc_een0_tune	855Bh
lane2_rx_slc_een0_tune	895Bh
lane3_rx_slc_een0_tune	8D5Bh

13.4.10.2.279.2 Diagram



13.4.10.2.279.3 Fields

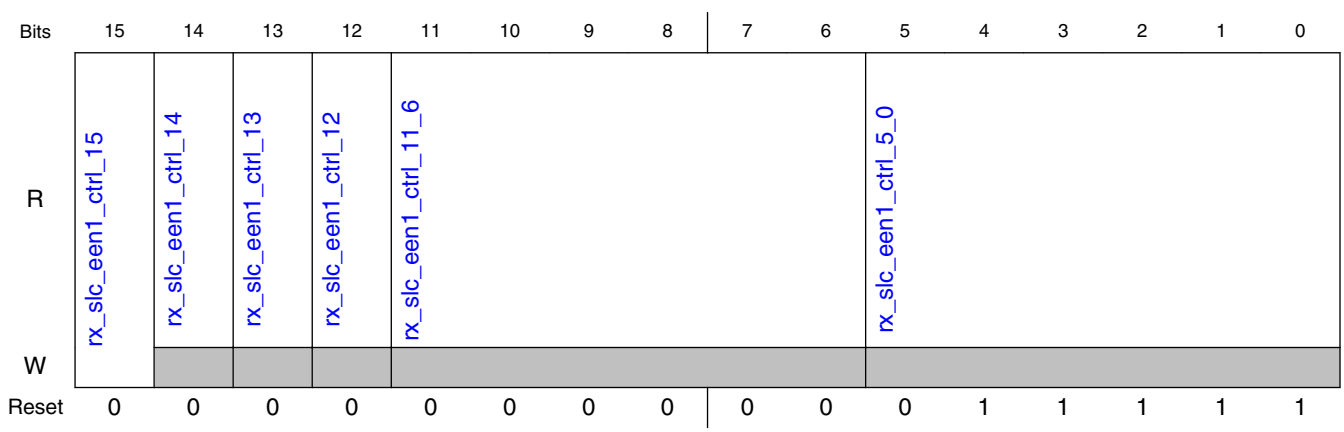
Field	Function
15-6 rx_slc_een0_tune_15_6	Reserved
5-0 rx_slc_een0_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.280 RX sampler latch E even negative 1 calibration unit control register (lane0_rx_slc_een1_ctrl - lane3_rx_slc_een1_ctrl)

13.4.10.2.280.1 Offset

Register	Offset
lane0_rx_slc_een1_ctrl	815Ch
lane1_rx_slc_een1_ctrl	855Ch
lane2_rx_slc_een1_ctrl	895Ch
lane3_rx_slc_een1_ctrl	8D5Ch

13.4.10.2.280.2 Diagram



13.4.10.2.280.3 Fields

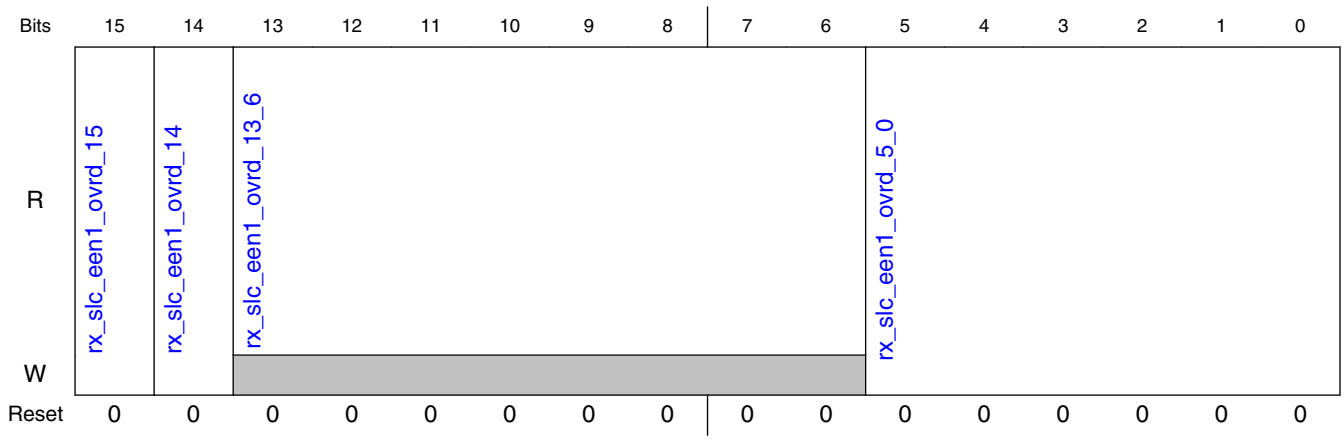
Field	Function
15 rx_slc_een1_ctrl_15	Start calibration: Activating (1b1) this bit will start the calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the calibration process done bit in this register is cleared.
14 rx_slc_een1_ctrl_14	Calibration done: This bit will be set to 1b1 when the calibration process is complete. It will be cleared by
13 rx_slc_een1_ctrl_13	No calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the comparator has not responded indicating that a valid calibration value has been reached.
12 rx_slc_een1_ctrl_12	Current comparator response: This is the current state of the comparator response signal. This signal is not synchronized, and is provided for diagnostic purposes only.
11-6 rx_slc_een1_ctrl_11_6	Reserved
5-0 rx_slc_een1_ctrl_5_0	Calibration code: This is the calibration code that was determined by the calibration process. The following are the values for the code:

13.4.10.2.281 RX sampler latch E even negative 1 calibration unit override register (lane0_rx_slc_een1_ovrd - lane3_rx_slc_een1_ovrd)

13.4.10.2.281.1 Offset

Register	Offset
lane0_rx_slc_een1_ovrd	815Dh
lane1_rx_slc_een1_ovrd	855Dh
lane2_rx_slc_een1_ovrd	895Dh
lane3_rx_slc_een1_ovrd	8D5Dh

13.4.10.2.281.2 Diagram



13.4.10.2.281.3 Fields

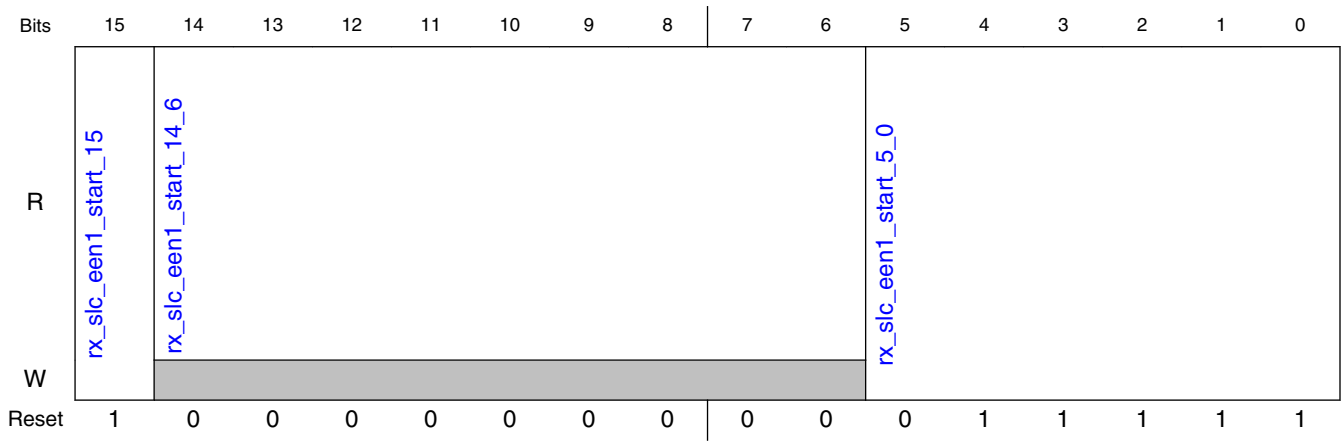
Field	Function
15 rx_slc_een1_ovrd_15	Calibration code override enable: Activation (1b1) of this register bit allows the calibration code determined during the automatic resistor calibration process to be overridden. The override value is specified using the calibration code override value field of this register.
14 rx_slc_een1_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-6 rx_slc_een1_ovrd_13_6	Reserved
5-0 rx_slc_een1_ovrd_5_0	Calibration code override value: These bits are used to override the calibration code determined during the automatic resistor calibration process. The calibration code written to these bits is valid when the calibration code override enable bit in this register is active. See the control register description for valid codes.

13.4.10.2.282 RX sampler latch E even negative 1 calibration unit start register (lane0_rx_slc_een1_start - lane3_rx_slc_een1_start)

13.4.10.2.282.1 Offset

Register	Offset
lane0_rx_slc_een1_start	815Eh
lane1_rx_slc_een1_start	855Eh
lane2_rx_slc_een1_start	895Eh
lane3_rx_slc_een1_start	8D5Eh

13.4.10.2.282.2 Diagram



13.4.10.2.282.3 Fields

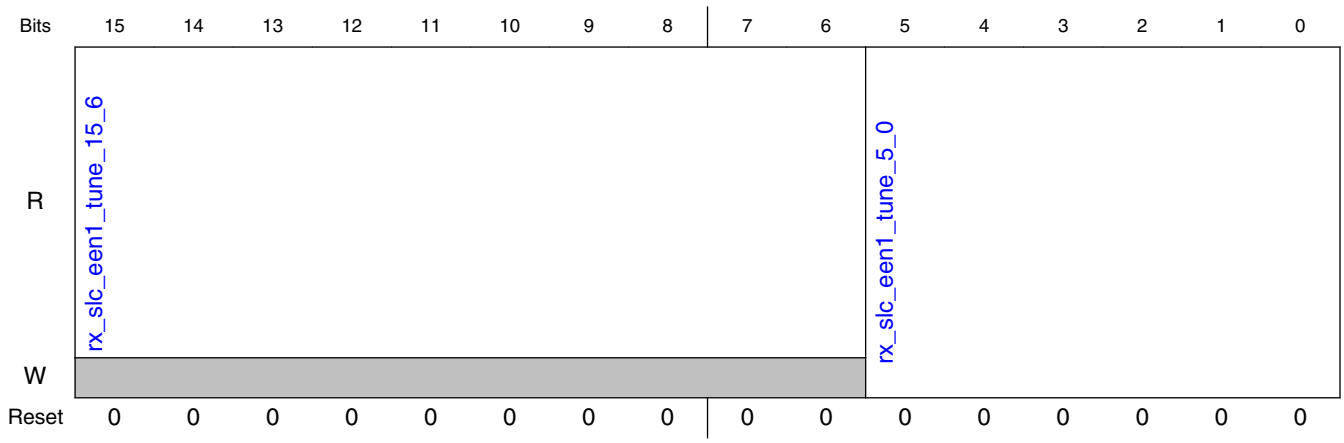
Field	Function
15 rx_slc_een1_start_15	Calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 rx_slc_een1_start_14_6	Reserved
5-0 rx_slc_een1_start_5_0	Start calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run. See the control register description for valid codes.

13.4.10.2.283 RX sampler latch E even negative 1 calibration unit tune register (lane0_rx_slc_een1_tune - lane3_rx_slc_een1_tune)

13.4.10.2.283.1 Offset

Register	Offset
lane0_rx_slc_een1_tune	815Fh
lane1_rx_slc_een1_tune	855Fh
lane2_rx_slc_een1_tune	895Fh
lane3_rx_slc_een1_tune	8D5Fh

13.4.10.2.283.2 Diagram



13.4.10.2.283.3 Fields

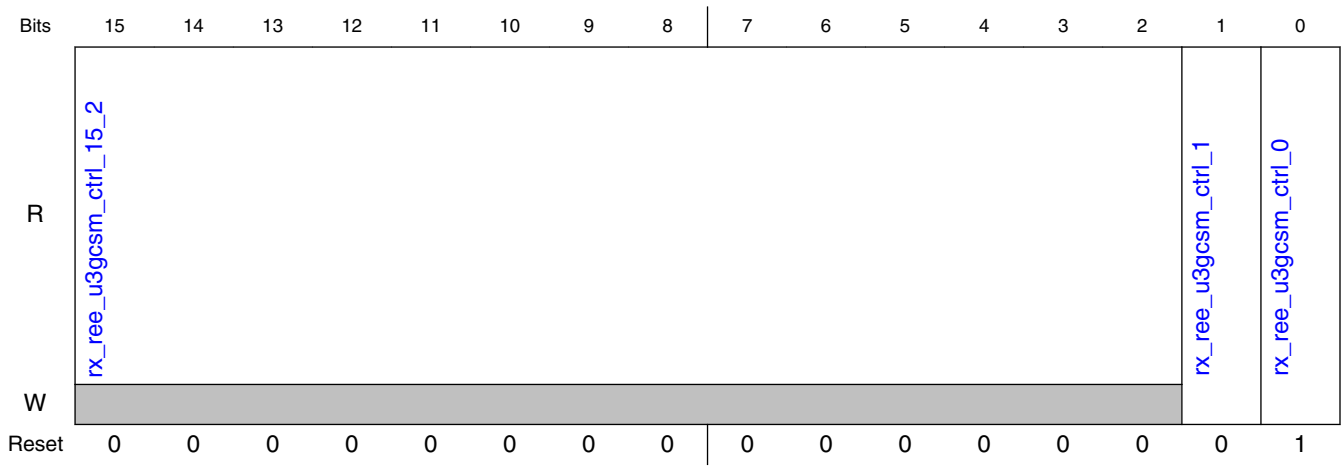
Field	Function
15-6 rx_slc_een1_tune_15_6	Reserved
5-0 rx_slc_een1_tune_5_0	Calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.284 REE USB 3 general control state machine control register (lane0_rx_ree_u3gcsm_ctrl - lane3_rx_ree_u3gcsm_ctrl)

13.4.10.2.284.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_ctrl	8160h
lane1_rx_ree_u3gcsm_ctrl	8560h
lane2_rx_ree_u3gcsm_ctrl	8960h
lane3_rx_ree_u3gcsm_ctrl	8D60h

13.4.10.2.284.2 Diagram



13.4.10.2.284.3 Fields

Field	Function
15-2 rx_ree_u3gcsm_ctrl_15_2	Reserved
1 rx_ree_u3gcsm_ctrl_1	Force run equalization: Setting this bit to a 1b1, will force the general control state machine to run, independent of the macro functions that normally run the equalization.
0 rx_ree_u3gcsm_ctrl_0	Enable: This bit enables the general control state machine function.

13.4.10.2.285 REE USB 3 general control state machine phase 1 equalization enable mask register (lane0_rx_ree_u3gcsm_eqenm_ph1 - lane3_rx_ree_u3gcsm_eqenm_ph1)

13.4.10.2.285.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_eqenm_ph1	8161h
lane1_rx_ree_u3gcsm_eqenm_ph1	8561h
lane2_rx_ree_u3gcsm_eqenm_ph1	8961h

Table continues on the next page...

Register	Offset
lane3_rx_ree_u3gcsm_eqenm_ph1	8D61h

13.4.10.2.285.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	rx_ree_u3gcsm_eqenm_ph1_15	rx_ree_u3gcsm_eqenm_ph1_14	rx_ree_u3gcsm_eqenm_ph1_13	rx_ree_u3gcsm_eqenm_ph1_12	rx_ree_u3gcsm_eqenm_ph1_11	rx_ree_u3gcsm_eqenm_ph1_10	rx_ree_u3gcsm_eqenm_ph1_9	rx_ree_u3gcsm_eqenm_ph1_8	rx_ree_u3gcsm_eqenm_ph1_7	rx_ree_u3gcsm_eqenm_ph1_6	rx_ree_u3gcsm_eqenm_ph1_5	rx_ree_u3gcsm_eqenm_ph1_4	rx_ree_u3gcsm_eqenm_ph1_3	rx_ree_u3gcsm_eqenm_ph1_2	rx_ree_u3gcsm_eqenm_ph1_1	rx_ree_u3gcsm_eqenm_ph1_0
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1

13.4.10.2.285.3 Fields

Field	Function
15 rx_ree_u3gcsm_eqenm_ph1_15	Reserved
14 rx_ree_u3gcsm_eqenm_ph1_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
13 rx_ree_u3gcsm_eqenm_ph1_13	Reserved
12 rx_ree_u3gcsm_eqenm_ph1_12	Reserved
11 rx_ree_u3gcsm_eqenm_ph1_11	Reserved
10 rx_ree_u3gcsm_eqenm_ph1_10	Reserved

Table continues on the next page...

Field	Function
9 rx_ree_u3gcsm_eqenm_ph1_9	RX attenuation : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
8 rx_ree_u3gcsm_eqenm_ph1_8	RX VGA gain : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
7 rx_ree_u3gcsm_eqenm_ph1_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
6 rx_ree_u3gcsm_eqenm_ph1_6	RX peaking amp gain : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
5 rx_ree_u3gcsm_eqenm_ph1_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
4 rx_ree_u3gcsm_eqenm_ph1_4	Reserved
3 rx_ree_u3gcsm_eqenm_ph1_3	Reserved
2 rx_ree_u3gcsm_eqenm_ph1_2	RX tap 3 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
1 rx_ree_u3gcsm_eqenm_ph1_1	RX tap 2 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
0 rx_ree_u3gcsm_eqenm_ph1_0	RX tap 1 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.

13.4.10.2.286 REE USB 3 general control state machine phase 2 equalization enable mask register (lane0_rx_ree_u3gcsm_eqenm_ph2 - lane3_rx_ree_u3gcsm_eqenm_ph2)

13.4.10.2.286.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_eqenm_ph2	8162h
lane1_rx_ree_u3gcsm_eqenm_ph2	8562h

Table continues on the next page...

Register	Offset
lane2_rx_ree_u3gcsm_eqenm_ph2	8962h
lane3_rx_ree_u3gcsm_eqenm_ph2	8D62h

13.4.10.2.286.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	rx_ree_u3gcsm_eqenm_ph2_15	rx_ree_u3gcsm_eqenm_ph2_14	rx_ree_u3gcsm_eqenm_ph2_13	rx_ree_u3gcsm_eqenm_ph2_12	rx_ree_u3gcsm_eqenm_ph2_11	rx_ree_u3gcsm_eqenm_ph2_10	rx_ree_u3gcsm_eqenm_ph2_9	rx_ree_u3gcsm_eqenm_ph2_8	rx_ree_u3gcsm_eqenm_ph2_7	rx_ree_u3gcsm_eqenm_ph2_6	rx_ree_u3gcsm_eqenm_ph2_5	rx_ree_u3gcsm_eqenm_ph2_4	rx_ree_u3gcsm_eqenm_ph2_3	rx_ree_u3gcsm_eqenm_ph2_2	rx_ree_u3gcsm_eqenm_ph2_1	rx_ree_u3gcsm_eqenm_ph2_0
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1

13.4.10.2.286.3 Fields

Field	Function
15 rx_ree_u3gcsm_eqenm_ph2_15	Reserved
14 rx_ree_u3gcsm_eqenm_ph2_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
13 rx_ree_u3gcsm_eqenm_ph2_13	Reserved
12 rx_ree_u3gcsm_eqenm_ph2_12	Reserved
11 rx_ree_u3gcsm_eqenm_ph2_11	Reserved
10	Reserved

Table continues on the next page...

Field	Function
rx_ree_u3gcsm_eqenm_ph2_10	
9 rx_ree_u3gcsm_eqenm_ph2_9	RX attenuation : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
8 rx_ree_u3gcsm_eqenm_ph2_8	RX VGA gain : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
7 rx_ree_u3gcsm_eqenm_ph2_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
6 rx_ree_u3gcsm_eqenm_ph2_6	RX peaking amp gain : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
5 rx_ree_u3gcsm_eqenm_ph2_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
4 rx_ree_u3gcsm_eqenm_ph2_4	Reserved
3 rx_ree_u3gcsm_eqenm_ph2_3	Reserved
2 rx_ree_u3gcsm_eqenm_ph2_2	RX tap 3 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
1 rx_ree_u3gcsm_eqenm_ph2_1	RX tap 2 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.
0 rx_ree_u3gcsm_eqenm_ph2_0	RX tap 1 : When set to 1b1, this function is enabled when the USB 3.0 general control state machine is controlling the REE.

13.4.10.2.287 REE USB 3 general control state machine start timer value register (lane0_rx_ree_u3gcsm_start_tmr - lane3_rx_ree_u3gcsm_start_tmr)

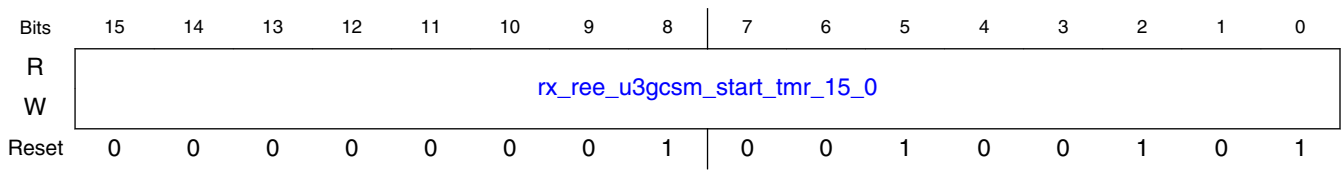
13.4.10.2.287.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_start_tmr	8163h

Table continues on the next page...

Register	Offset
lane1_rx_ree_u3gcsm_start_tmr	8563h
lane2_rx_ree_u3gcsm_start_tmr	8963h
lane3_rx_ree_u3gcsm_start_tmr	8D63h

13.4.10.2.287.2 Diagram



13.4.10.2.287.3 Fields

Field	Function
15-0 rx_ree_u3gcsm_start_tmr_15_0	Start timer value : The number of clock cycles the state machine will wait in the Start Delay state.

13.4.10.2.288 REE USB 3 general control state machine run phase 1 timer value register (lane0_rx_ree_u3gcsm_run_ph1_tmr - lane3_rx_ree_u3gcsm_run_ph1_tmr)

13.4.10.2.288.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_run_ph1_tmr	8164h
lane1_rx_ree_u3gcsm_run_ph1_tmr	8564h
lane2_rx_ree_u3gcsm_run_ph1_tmr	8964h
lane3_rx_ree_u3gcsm_run_ph1_tmr	8D64h

13.4.10.2.288.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_u3gcsm_run_ph1_tmr_15_0															
Reset	0	0	0	0	0	1	1	1	1	0	1	0	0	0	1	0

13.4.10.2.288.3 Fields

Field	Function
15-0 rx_ree_u3gcsm_run_ph1_tmr_15_0	Run phase 1 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 1 state.

13.4.10.2.289 REE USB 3 general control state machine run phase 2 timer value register (lane0_rx_ree_u3gcsm_run_ph2_tmr - lane3_rx_ree_u3gcsm_run_ph2_tmr)

13.4.10.2.289.1 Offset

Register	Offset
lane0_rx_ree_u3gcsm_run_ph2_tmr	8165h
lane1_rx_ree_u3gcsm_run_ph2_tmr	8565h
lane2_rx_ree_u3gcsm_run_ph2_tmr	8965h
lane3_rx_ree_u3gcsm_run_ph2_tmr	8D65h

13.4.10.2.289.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_u3gcsm_run_ph2_tmr_15_0															
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1

13.4.10.2.289.3 Fields

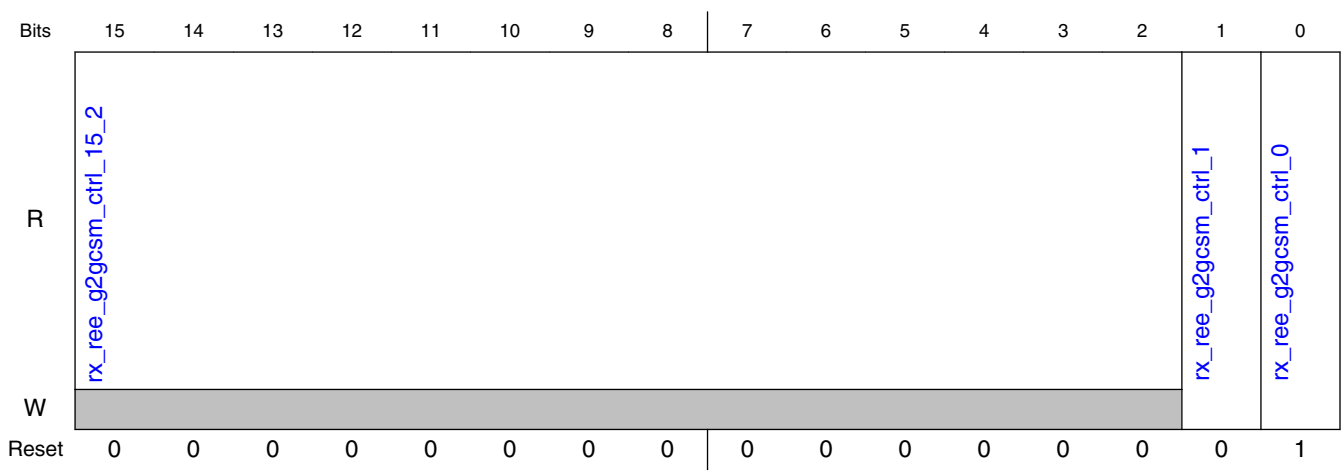
Field	Function
15-0 rx_ree_u3gcsm_run_ph2_tmr_15_0	Run phase 2 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 2 state.

13.4.10.2.290 REE PCIe Gen 2 general control state machine control register (lane0_rx_ree_g2gcsm_ctrl - lane3_rx_ree_g2gcsm_ctrl)

13.4.10.2.290.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_ctrl	8168h
lane1_rx_ree_g2gcsm_ctrl	8568h
lane2_rx_ree_g2gcsm_ctrl	8968h
lane3_rx_ree_g2gcsm_ctrl	8D68h

13.4.10.2.290.2 Diagram



13.4.10.2.290.3 Fields

Field	Function
15-2 rx_ree_g2gcsm_ctrl_15_2	Reserved
1 rx_ree_g2gcsm_ctrl_1	Force run equalization: Setting this bit to a 1b1, will force the general control state machine to run, independent of the macro functions that normally run the equalization.
0 rx_ree_g2gcsm_ctrl_0	Enable: This bit enables the general control state machine function.

13.4.10.2.291 REE PCIe Gen 2 general control state machine phase 1 equalization enable mask register (lane0_rx_ree_g2gcsm_eqenm_ph1 - lane3_rx_ree_g2gcsm_eqenm_ph1)

13.4.10.2.291.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_eqenm_ph1	8169h
lane1_rx_ree_g2gcsm_eqenm_ph1	8569h
lane2_rx_ree_g2gcsm_eqenm_ph1	8969h
lane3_rx_ree_g2gcsm_eqenm_ph1	8D69h

13.4.10.2.291.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_ree_g2gcsm_eqenm_ph1_15	rx_ree_g2gcsm_eqenm_ph1_14	rx_ree_g2gcsm_eqenm_ph1_13	rx_ree_g2gcsm_eqenm_ph1_12	rx_ree_g2gcsm_eqenm_ph1_11	rx_ree_g2gcsm_eqenm_ph1_10	rx_ree_g2gcsm_eqenm_ph1_9	rx_ree_g2gcsm_eqenm_ph1_8	rx_ree_g2gcsm_eqenm_ph1_7	rx_ree_g2gcsm_eqenm_ph1_6	rx_ree_g2gcsm_eqenm_ph1_5	rx_ree_g2gcsm_eqenm_ph1_4	rx_ree_g2gcsm_eqenm_ph1_3	rx_ree_g2gcsm_eqenm_ph1_2	rx_ree_g2gcsm_eqenm_ph1_1	rx_ree_g2gcsm_eqenm_ph1_0
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1

13.4.10.2.291.3 Fields

Field	Function
15 rx_ree_g2gcsm_eqenm_ph1_15	Reserved
14 rx_ree_g2gcsm_eqenm_ph1_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
13 rx_ree_g2gcsm_eqenm_ph1_13	Reserved
12 rx_ree_g2gcsm_eqenm_ph1_12	Reserved
11 rx_ree_g2gcsm_eqenm_ph1_11	Reserved
10 rx_ree_g2gcsm_eqenm_ph1_10	Reserved
9 rx_ree_g2gcsm_eqenm_ph1_9	RX attenuation : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
8 rx_ree_g2gcsm_eqenm_ph1_8	RX VGA gain : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.

Table continues on the next page...

Field	Function
7 rx_ree_g2gcsm_eqenm_ph1_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
6 rx_ree_g2gcsm_eqenm_ph1_6	RX peaking amp gain : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
5 rx_ree_g2gcsm_eqenm_ph1_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
4 rx_ree_g2gcsm_eqenm_ph1_4	Reserved
3 rx_ree_g2gcsm_eqenm_ph1_3	Reserved
2 rx_ree_g2gcsm_eqenm_ph1_2	RX tap 3 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
1 rx_ree_g2gcsm_eqenm_ph1_1	RX tap 2 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
0 rx_ree_g2gcsm_eqenm_ph1_0	RX tap 1 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.

13.4.10.2.292 REE PCIe Gen 2 general control state machine phase 2 equalization enable mask register (lane0_rx_ree_g2gcsm_eqenm_ph2 - lane3_rx_ree_g2gcsm_eqenm_ph2)

13.4.10.2.292.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_eqenm_ph2	816Ah
lane1_rx_ree_g2gcsm_eqenm_ph2	856Ah
lane2_rx_ree_g2gcsm_eqenm_ph2	896Ah
lane3_rx_ree_g2gcsm_eqenm_ph2	8D6Ah

13.4.10.2.292.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_ree_g2gcsm_eqenm_ph2_15	rx_ree_g2gcsm_eqenm_ph2_14	rx_ree_g2gcsm_eqenm_ph2_13	rx_ree_g2gcsm_eqenm_ph2_12	rx_ree_g2gcsm_eqenm_ph2_11	rx_ree_g2gcsm_eqenm_ph2_10	rx_ree_g2gcsm_eqenm_ph2_9	rx_ree_g2gcsm_eqenm_ph2_8	rx_ree_g2gcsm_eqenm_ph2_7	rx_ree_g2gcsm_eqenm_ph2_6	rx_ree_g2gcsm_eqenm_ph2_5	rx_ree_g2gcsm_eqenm_ph2_4	rx_ree_g2gcsm_eqenm_ph2_3	rx_ree_g2gcsm_eqenm_ph2_2	rx_ree_g2gcsm_eqenm_ph2_1	rx_ree_g2gcsm_eqenm_ph2_0
W																
Reset	0	0	0	0	0	0	0	0	1	1	0	0	0	1	1	1

13.4.10.2.292.3 Fields

Field	Function
15 rx_ree_g2gcsm_eqenm_ph2_15	Reserved
14 rx_ree_g2gcsm_eqenm_ph2_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
13 rx_ree_g2gcsm_eqenm_ph2_13	Reserved
12 rx_ree_g2gcsm_eqenm_ph2_12	Reserved
11 rx_ree_g2gcsm_eqenm_ph2_11	Reserved
10 rx_ree_g2gcsm_eqenm_ph2_10	Reserved
9 rx_ree_g2gcsm_eqenm_ph2_9	RX attenuation : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
8 rx_ree_g2gcsm_eqenm_ph2_8	RX VGA gain : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.

Table continues on the next page...

Field	Function
7 rx_ree_g2gcsm_eqenm_ph2_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
6 rx_ree_g2gcsm_eqenm_ph2_6	RX peaking amp gain : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
5 rx_ree_g2gcsm_eqenm_ph2_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
4 rx_ree_g2gcsm_eqenm_ph2_4	Reserved
3 rx_ree_g2gcsm_eqenm_ph2_3	Reserved
2 rx_ree_g2gcsm_eqenm_ph2_2	RX tap 3 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
1 rx_ree_g2gcsm_eqenm_ph2_1	RX tap 2 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.
0 rx_ree_g2gcsm_eqenm_ph2_0	RX tap 1 : When set to 1b1, this function is enabled when the PCIe Gen 2 general control state machine is controlling the REE.

13.4.10.2.293 REE PCIe Gen 2 general control state machine start timer value register (lane0_rx_ree_g2gcsm_start_tmr - lane3_rx_ree_g2gcsm_start_tmr)

13.4.10.2.293.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_start_tmr	816Bh
lane1_rx_ree_g2gcsm_start_tmr	856Bh
lane2_rx_ree_g2gcsm_start_tmr	896Bh
lane3_rx_ree_g2gcsm_start_tmr	8D6Bh

13.4.10.2.293.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_g2gcsm_start_tmr_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.293.3 Fields

Field	Function
15-0 rx_ree_g2gcsm_start_tmr_15_0	Start timer value : The number of clock cycles the state machine will wait in the Start Delay state.

13.4.10.2.294 REE PCIe Gen 2 general control state machine run phase 1 timer value register (lane0_rx_ree_g2gcsm_run_ph1_tmr - lane3_rx_ree_g2gcsm_run_ph1_tmr)

13.4.10.2.294.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_run_ph1_tmr	816Ch
lane1_rx_ree_g2gcsm_run_ph1_tmr	856Ch
lane2_rx_ree_g2gcsm_run_ph1_tmr	896Ch
lane3_rx_ree_g2gcsm_run_ph1_tmr	8D6Ch

13.4.10.2.294.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_g2gcsm_run_ph1_tmr_15_0															
Reset	0	0	0	0	1	1	1	1	0	1	0	0	0	0	1	1

13.4.10.2.294.3 Fields

Field	Function
15-0 rx_ree_g2gcsm_run_ph1_tmr_15_0	Run phase 1 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 1 state.

13.4.10.2.295 REE PCIe Gen 2 general control state machine run phase 2 timer value register (lane0_rx_ree_g2gcsm_run_ph2_tmr - lane3_rx_ree_g2gcsm_run_ph2_tmr)

13.4.10.2.295.1 Offset

Register	Offset
lane0_rx_ree_g2gcsm_run_ph2_tmr	816Dh
lane1_rx_ree_g2gcsm_run_ph2_tmr	856Dh
lane2_rx_ree_g2gcsm_run_ph2_tmr	896Dh
lane3_rx_ree_g2gcsm_run_ph2_tmr	8D6Dh

13.4.10.2.295.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_g2gcsm_run_ph2_tmr_15_0															
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1

13.4.10.2.295.3 Fields

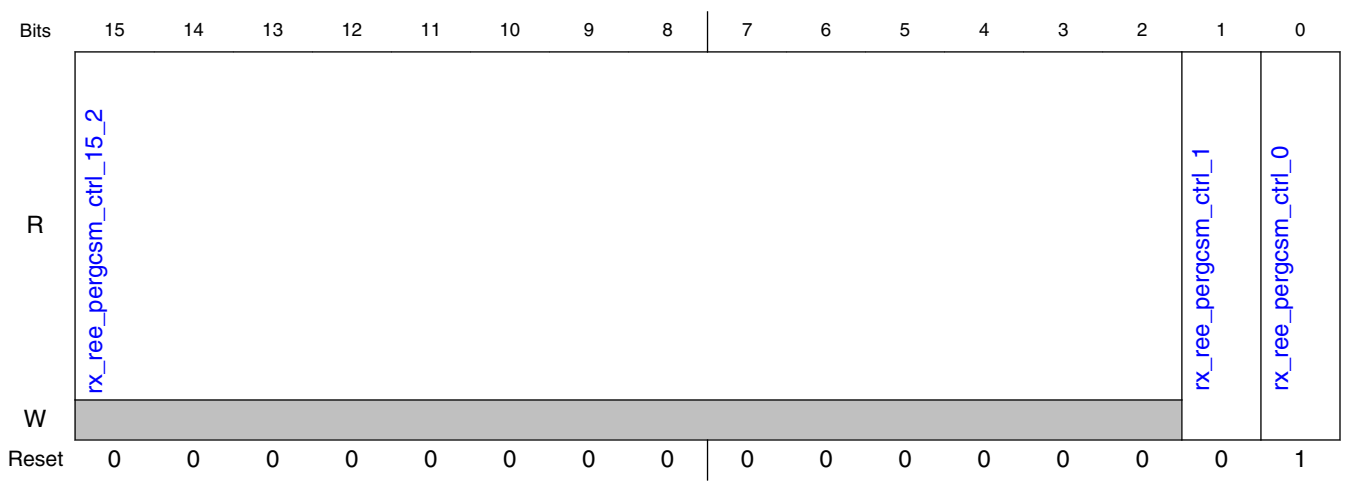
Field	Function
15-0 rx_ree_g2gcsm_run_ph2_tmr_15_0	Run phase 2 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 2 state.

13.4.10.2.296 REE periodic general control state machine control register (lane0_rx_ree_pergcsm_ctrl - lane3_rx_ree_pergcsm_ctrl)

13.4.10.2.296.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_ctrl	8178h
lane1_rx_ree_pergcsm_ctrl	8578h
lane2_rx_ree_pergcsm_ctrl	8978h
lane3_rx_ree_pergcsm_ctrl	8D78h

13.4.10.2.296.2 Diagram



13.4.10.2.296.3 Fields

Field	Function
15-2 rx_ree_pergcsm_ctrl_15_2	Reserved
1 rx_ree_pergcsm_ctrl_1	Force run equalization: Setting this bit to a 1b1, will force the general control state machine to run, independent of the macro functions that normally run the equalization.
0 rx_ree_pergcsm_ctrl_0	Enable: This bit enables the general control state machine function.

13.4.10.2.297 REE periodic general control state machine phase 1 equalization enable mask register (lane0_rx_ree_pergcsm_eqenm_ph1 - lane3_rx_ree_pergcsm_eqenm_ph1)

13.4.10.2.297.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_eqenm_ph1	8179h
lane1_rx_ree_pergcsm_eqenm_ph1	8579h
lane2_rx_ree_pergcsm_eqenm_ph1	8979h
lane3_rx_ree_pergcsm_eqenm_ph1	8D79h

13.4.10.2.297.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	rx_ree_pergcsm_eqenm_ph1_15	rx_ree_pergcsm_eqenm_ph1_14	rx_ree_pergcsm_eqenm_ph1_13	rx_ree_pergcsm_eqenm_ph1_12	rx_ree_pergcsm_eqenm_ph1_11	rx_ree_pergcsm_eqenm_ph1_10	rx_ree_pergcsm_eqenm_ph1_9	rx_ree_pergcsm_eqenm_ph1_8	rx_ree_pergcsm_eqenm_ph1_7	rx_ree_pergcsm_eqenm_ph1_6	rx_ree_pergcsm_eqenm_ph1_5	rx_ree_pergcsm_eqenm_ph1_4	rx_ree_pergcsm_eqenm_ph1_3	rx_ree_pergcsm_eqenm_ph1_2	rx_ree_pergcsm_eqenm_ph1_1	rx_ree_pergcsm_eqenm_ph1_0
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

13.4.10.2.297.3 Fields

Field	Function
15 rx_ree_pergcsm_eqenm_ph1_15	Reserved

Table continues on the next page...

Field	Function
14 rx_ree_pergcsm_eqenm_ph1_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
13 rx_ree_pergcsm_eqenm_ph1_13	Reserved
12 rx_ree_pergcsm_eqenm_ph1_12	Reserved
11 rx_ree_pergcsm_eqenm_ph1_11	Reserved
10 rx_ree_pergcsm_eqenm_ph1_10	Reserved
9 rx_ree_pergcsm_eqenm_ph1_9	RX attenuation : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
8 rx_ree_pergcsm_eqenm_ph1_8	RX VGA gain : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
7 rx_ree_pergcsm_eqenm_ph1_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
6 rx_ree_pergcsm_eqenm_ph1_6	RX peaking amp gain : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
5 rx_ree_pergcsm_eqenm_ph1_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
4 rx_ree_pergcsm_eqenm_ph1_4	Reserved
3 rx_ree_pergcsm_eqenm_ph1_3	Reserved
2 rx_ree_pergcsm_eqenm_ph1_2	RX tap 3 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.

Table continues on the next page...

Field	Function
1 rx_ree_pergcsm_eqenm_ph1_1	RX tap 2 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
0 rx_ree_pergcsm_eqenm_ph1_0	RX tap 1 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.

13.4.10.2.298 REE periodic general control state machine phase 2 equalization enable mask register (lane0_rx_ree_pergcsm_eqenm_ph2 - lane3_rx_ree_pergcsm_eqenm_ph2)

13.4.10.2.298.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_eqenm_ph2	817Ah
lane1_rx_ree_pergcsm_eqenm_ph2	857Ah
lane2_rx_ree_pergcsm_eqenm_ph2	897Ah
lane3_rx_ree_pergcsm_eqenm_ph2	8D7Ah

13.4.10.2.298.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_ree_pergcsm_eqenm_ph2_15	rx_ree_pergcsm_eqenm_ph2_14	rx_ree_pergcsm_eqenm_ph2_13	rx_ree_pergcsm_eqenm_ph2_12	rx_ree_pergcsm_eqenm_ph2_11	rx_ree_pergcsm_eqenm_ph2_10	rx_ree_pergcsm_eqenm_ph2_9	rx_ree_pergcsm_eqenm_ph2_8	rx_ree_pergcsm_eqenm_ph2_7	rx_ree_pergcsm_eqenm_ph2_6	rx_ree_pergcsm_eqenm_ph2_5	rx_ree_pergcsm_eqenm_ph2_4	rx_ree_pergcsm_eqenm_ph2_3	rx_ree_pergcsm_eqenm_ph2_2	rx_ree_pergcsm_eqenm_ph2_1	rx_ree_pergcsm_eqenm_ph2_0
W																
Reset	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0

13.4.10.2.298.3 Fields

Field	Function
15 rx_ree_pergcsm_eqenm_ph2_15	Reserved
14 rx_ree_pergcsm_eqenm_ph2_14	Ignore 1010 controller : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
13 rx_ree_pergcsm_eqenm_ph2_13	Reserved
12 rx_ree_pergcsm_eqenm_ph2_12	Reserved
11 rx_ree_pergcsm_eqenm_ph2_11	Reserved
10 rx_ree_pergcsm_eqenm_ph2_10	Reserved
9 rx_ree_pergcsm_eqenm_ph2_9	RX attenuation : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
8 rx_ree_pergcsm_eqenm_ph2_8	RX VGA gain : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
7 rx_ree_pergcsm_eqenm_ph2_7	RX offset correction coefficient : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
6 rx_ree_pergcsm_eqenm_ph2_6	RX peaking amp gain : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
5 rx_ree_pergcsm_eqenm_ph2_5	RX low frequency equalizer adaptive control : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
4 rx_ree_pergcsm_eqenm_ph2_4	Reserved
3	Reserved

Table continues on the next page...

Field	Function
rx_ree_pergcsm_eqenm_ph2_3	
2 rx_ree_pergcsm_eqenm_ph2_2	RX tap 3 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
1 rx_ree_pergcsm_eqenm_ph2_1	RX tap 2 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.
0 rx_ree_pergcsm_eqenm_ph2_0	RX tap 1 : When set to 1b1, this function is enabled when the periodic general control state machine is controlling the REE.

13.4.10.2.299 REE periodic general control state machine start timer value register (lane0_rx_ree_pergcsm_start_tmr - lane3_rx_ree_pergcsm_start_tmr)

13.4.10.2.299.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_start_tmr	817Bh
lane1_rx_ree_pergcsm_start_tmr	857Bh
lane2_rx_ree_pergcsm_start_tmr	897Bh
lane3_rx_ree_pergcsm_start_tmr	8D7Bh

13.4.10.2.299.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_pergcsm_start_tmr_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.299.3 Fields

Field	Function
15-0	Start timer value : The number of clock cycles the state machine will wait in the Start Delay state.

Field	Function
rx_ree_pergcsm_start_tmr_15_0	

13.4.10.2.300 REE periodic general control state machine run phase 1 timer value register (lane0_rx_ree_pergcsm_run_ph1_tmr - lane3_rx_ree_pergcsm_run_ph1_tmr)

13.4.10.2.300.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_run_ph1_tmr	817Ch
lane1_rx_ree_pergcsm_run_ph1_tmr	857Ch
lane2_rx_ree_pergcsm_run_ph1_tmr	897Ch
lane3_rx_ree_pergcsm_run_ph1_tmr	8D7Ch

13.4.10.2.300.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_pergcsm_run_ph1_tmr_15_0															
Reset	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

13.4.10.2.300.3 Fields

Field	Function
15-0 rx_ree_pergcsm_run_ph1_tmr_15_0	Run phase 1 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 1 state.

13.4.10.2.301 REE periodic general control state machine run phase 2 timer value register (lane0_rx_ree_pergcsm_run_ph2_tmr - lane3_rx_ree_pergcsm_run_ph2_tmr)

13.4.10.2.301.1 Offset

Register	Offset
lane0_rx_ree_pergcsm_run_ph2_tmr	817Dh
lane1_rx_ree_pergcsm_run_ph2_tmr	857Dh
lane2_rx_ree_pergcsm_run_ph2_tmr	897Dh
lane3_rx_ree_pergcsm_run_ph2_tmr	8D7Dh

13.4.10.2.301.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_pergcsm_run_ph2_tmr_15_0															
Reset	0	0	0	0	0	0	0	0	1	0	0	1	1	1	0	1

13.4.10.2.301.3 Fields

Field	Function
15-0 rx_ree_pergcsm_run_ph2_tmr_15_0	Run phase 2 timer value : This specifies the number of clock cycles the state machine will wait in the Run Equalization Phase 2 state.

13.4.10.2.302 REE tap 1 control register (lane0_rx_ree_tap1_ctrl - lane3_rx_ree_tap1_ctrl)

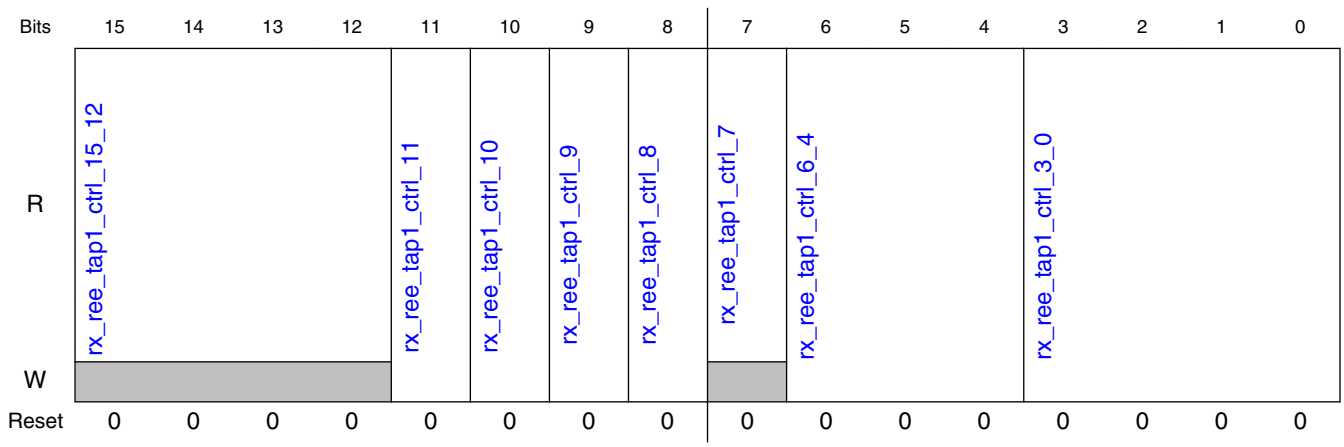
13.4.10.2.302.1 Offset

Register	Offset
lane0_rx_ree_tap1_ctrl	8180h
lane1_rx_ree_tap1_ctrl	8580h

Table continues on the next page...

Register	Offset
lane2_rx_ree_tap1_ctrl	8980h
lane3_rx_ree_tap1_ctrl	8D80h

13.4.10.2.302.2 Diagram



13.4.10.2.302.3 Fields

Field	Function
15-12 rx_ree_tap1_ctrl_15_12	Reserved
11 rx_ree_tap1_ctrl_11	Tap coefficient combinational logic zero crossing enable:
10 rx_ree_tap1_ctrl_10	Tap coefficient combinational logic non zero crossing enable:
9 rx_ree_tap1_ctrl_9	Tap coefficient combinational logic bit 0 only enable:
8 rx_ree_tap1_ctrl_8	Receiver DFE tap coefficient disable: This bit disables the
7 rx_ree_tap1_ctrl_7	Reserved
6-4	Tap integrator accumulator scaler value: Specifies the amount to scale the input to the tap integrator accumulator by. The following are the valid settings for this field:

Table continues on the next page...

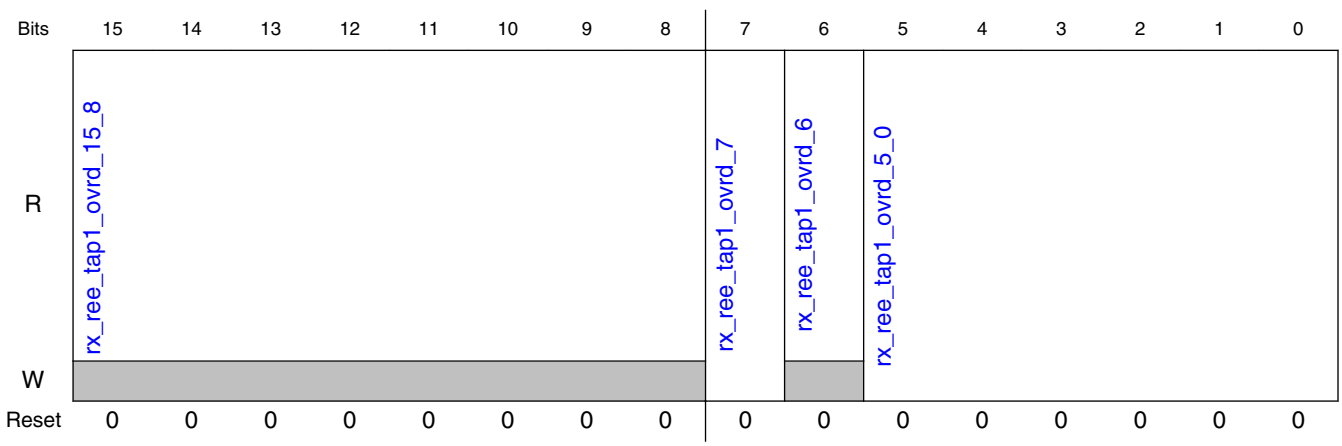
Field	Function
rx_ree_tap1_ctrl_6_4	
3-0 rx_ree_tap1_ctrl_3_0	Tap sigma delta accumulator scaler value: Specifies the amount to scale the input to the tap sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.303 REE tap 1 override register (lane0_rx_ree_tap1_ovrd - lane 3_rx_ree_tap1_ovrd)

13.4.10.2.303.1 Offset

Register	Offset
lane0_rx_ree_tap1_ovrd	8181h
lane1_rx_ree_tap1_ovrd	8581h
lane2_rx_ree_tap1_ovrd	8981h
lane3_rx_ree_tap1_ovrd	8D81h

13.4.10.2.303.2 Diagram



13.4.10.2.303.3 Fields

Field	Function
15-8 rx_ree_tap1_ovrd_15_8	Reserved

Table continues on the next page...

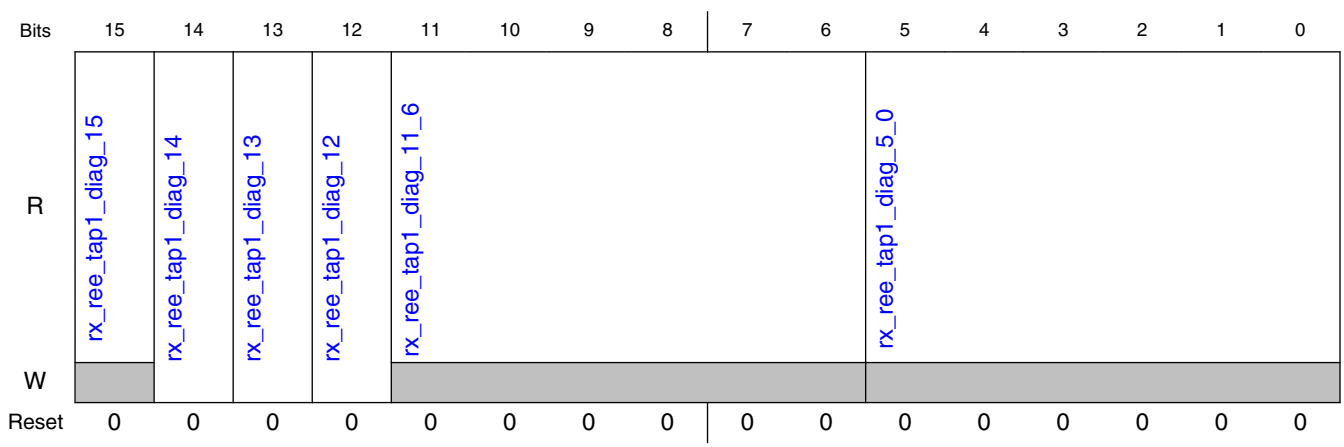
Field	Function
7 rx_ree_tap1_ovrd_7	Tap override enable: Setting this bit to a 1b1 will enable the tap override field in this register to override the tap integrator accumulator functions.
6 rx_ree_tap1_ovrd_6	Reserved
5-0 rx_ree_tap1_ovrd_5_0	Tap override value: When the tap override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.304 REE tap 1 diagnostics register (lane0_rx_ree_tap1_diag - lane3_rx_ree_tap1_diag)

13.4.10.2.304.1 Offset

Register	Offset
lane0_rx_ree_tap1_diag	8182h
lane1_rx_ree_tap1_diag	8582h
lane2_rx_ree_tap1_diag	8982h
lane3_rx_ree_tap1_diag	8D82h

13.4.10.2.304.2 Diagram



13.4.10.2.304.3 Fields

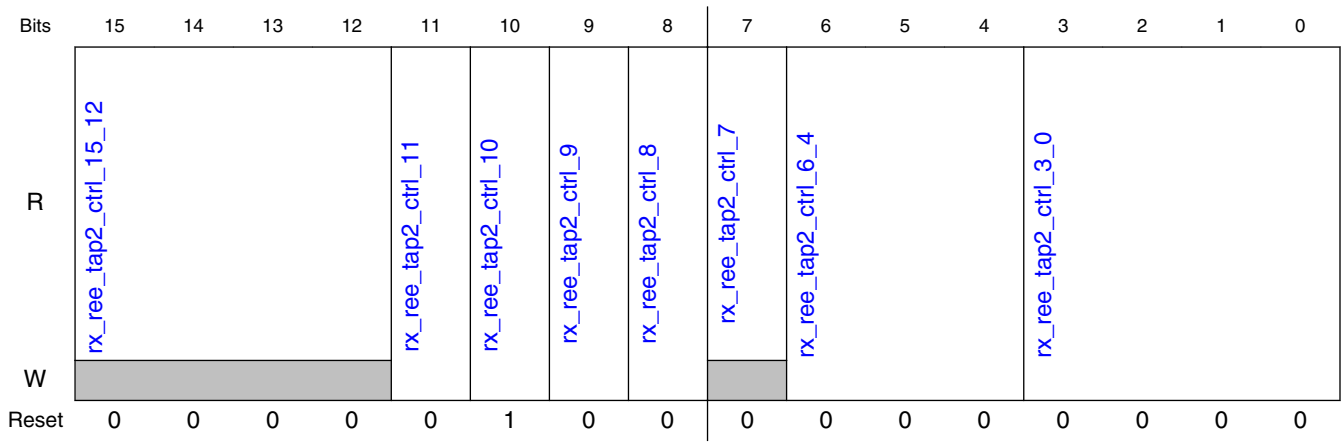
Field	Function
15 rx_ree_tap1_diag_15	Reserved
14 rx_ree_tap1_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the tap voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_tap1_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the tap voter function to activate the voter pos signal for a single clock cycle.
12 rx_ree_tap1_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the tap.
11-6 rx_ree_tap1_diag_11_6	Reserved
5-0 rx_ree_tap1_diag_5_0	Current tap integrator accumulator: Current value of the tap integrator accumulator.

13.4.10.2.305 REE tap 2 control register (lane0_rx_ree_tap2_ctrl - lane3_rx_ree_tap2_ctrl)

13.4.10.2.305.1 Offset

Register	Offset
lane0_rx_ree_tap2_ctrl	8184h
lane1_rx_ree_tap2_ctrl	8584h
lane2_rx_ree_tap2_ctrl	8984h
lane3_rx_ree_tap2_ctrl	8D84h

13.4.10.2.305.2 Diagram



13.4.10.2.305.3 Fields

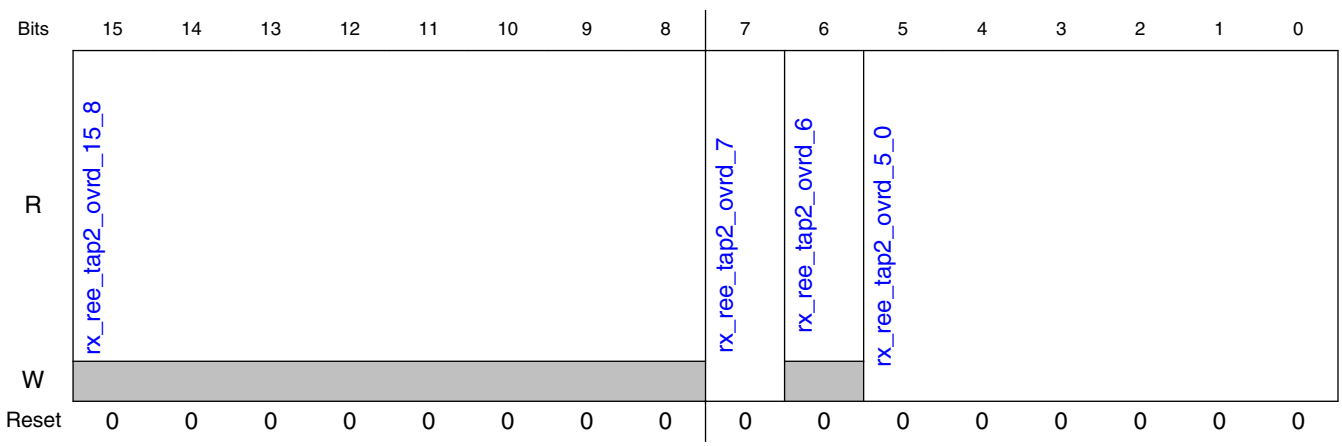
Field	Function
15-12 rx_ree_tap2_ctrl_15_12	Reserved
11 rx_ree_tap2_ctrl_11	Tap coefficient combinational logic zero crossing enable:
10 rx_ree_tap2_ctrl_10	Tap coefficient combinational logic non zero crossing enable:
9 rx_ree_tap2_ctrl_9	Tap coefficient combinational logic bit 0 only enable:
8 rx_ree_tap2_ctrl_8	Receiver DFE tap coefficient disable: This bit disables the
7 rx_ree_tap2_ctrl_7	Reserved
6-4 rx_ree_tap2_ctrl_6_4	Tap integrator accumulator scaler value: Specifies the amount to scale the input to the tap integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_tap2_ctrl_3_0	Tap sigma delta accumulator scaler value: Specifies the amount to scale the input to the tap sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.306 REE tap 2 override register (lane0_rx_ree_tap2_ovrd - lane 3_rx_ree_tap2_ovrd)

13.4.10.2.306.1 Offset

Register	Offset
lane0_rx_ree_tap2_ovrd	8185h
lane1_rx_ree_tap2_ovrd	8585h
lane2_rx_ree_tap2_ovrd	8985h
lane3_rx_ree_tap2_ovrd	8D85h

13.4.10.2.306.2 Diagram



13.4.10.2.306.3 Fields

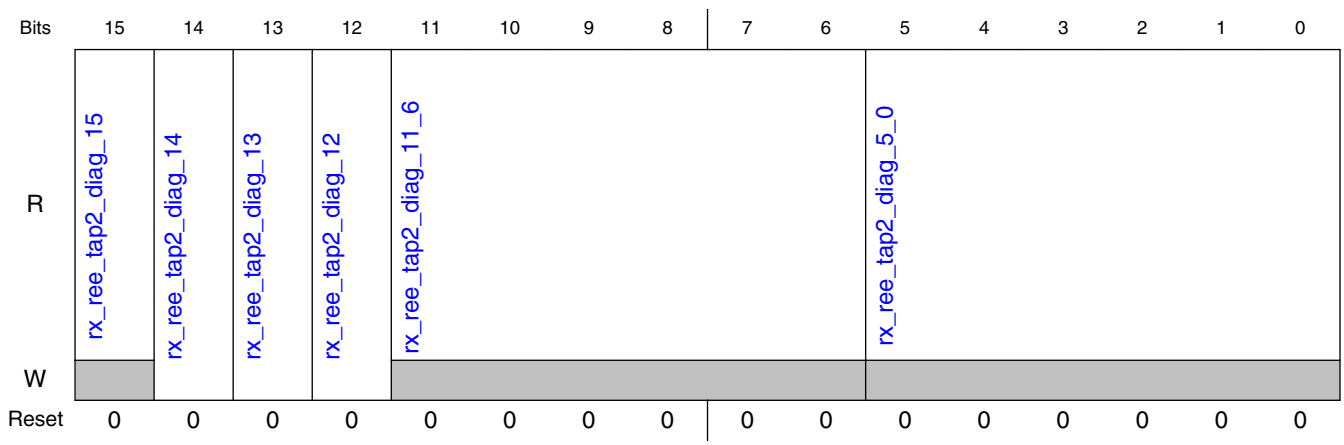
Field	Function
15-8 rx_ree_tap2_ovrd_15_8	Reserved
7 rx_ree_tap2_ovrd_7	Tap override enable: Setting this bit to a 1b1 will enable the tap override field in this register to override the tap integrator accumulator functions.
6 rx_ree_tap2_ovrd_6	Reserved
5-0 rx_ree_tap2_ovrd_5_0	Tap override value: When the tap override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.307 REE tap 2 diagnostics register (lane0_rx_ree_tap2_diag - lane3_rx_ree_tap2_diag)

13.4.10.2.307.1 Offset

Register	Offset
lane0_rx_ree_tap2_diag	8186h
lane1_rx_ree_tap2_diag	8586h
lane2_rx_ree_tap2_diag	8986h
lane3_rx_ree_tap2_diag	8D86h

13.4.10.2.307.2 Diagram



13.4.10.2.307.3 Fields

Field	Function
15 rx_ree_tap2_diag_15	Reserved
14 rx_ree_tap2_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the tap voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_tap2_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the tap voter function to activate the voter pos signal for a single clock cycle.
12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the tap.

Table continues on the next page...

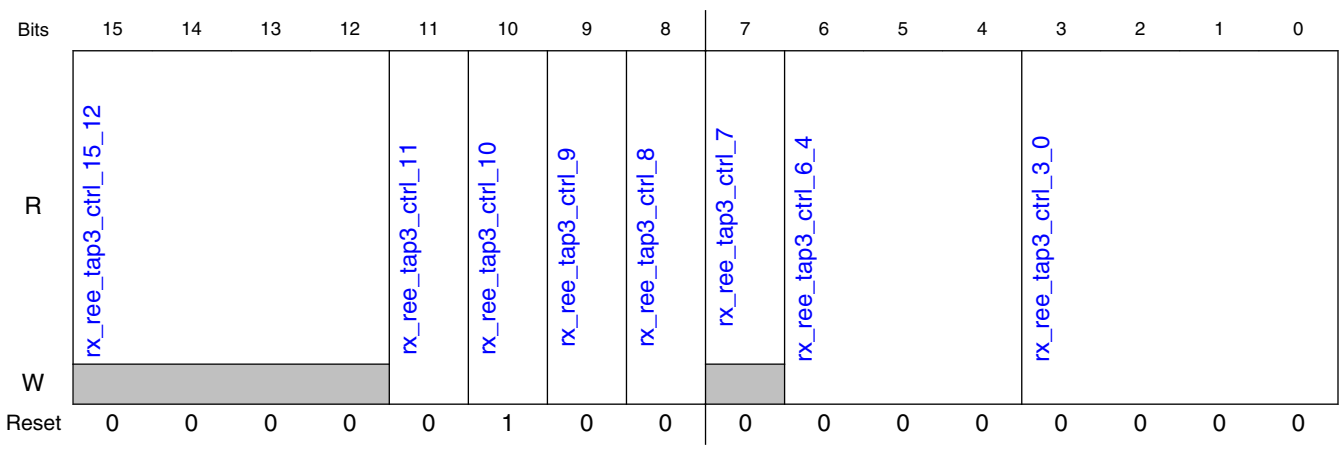
Field	Function
rx_ree_tap2_diag_12	
11-6 rx_ree_tap2_diag_11_6	Reserved
5-0 rx_ree_tap2_diag_5_0	Current tap integrator accumulator: Current value of the tap integrator accumulator.

13.4.10.2.308 REE tap 3 control register (lane0_rx_ree_tap3_ctrl - lane3_rx_ree_tap3_ctrl)

13.4.10.2.308.1 Offset

Register	Offset
lane0_rx_ree_tap3_ctrl	8188h
lane1_rx_ree_tap3_ctrl	8588h
lane2_rx_ree_tap3_ctrl	8988h
lane3_rx_ree_tap3_ctrl	8D88h

13.4.10.2.308.2 Diagram



13.4.10.2.308.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

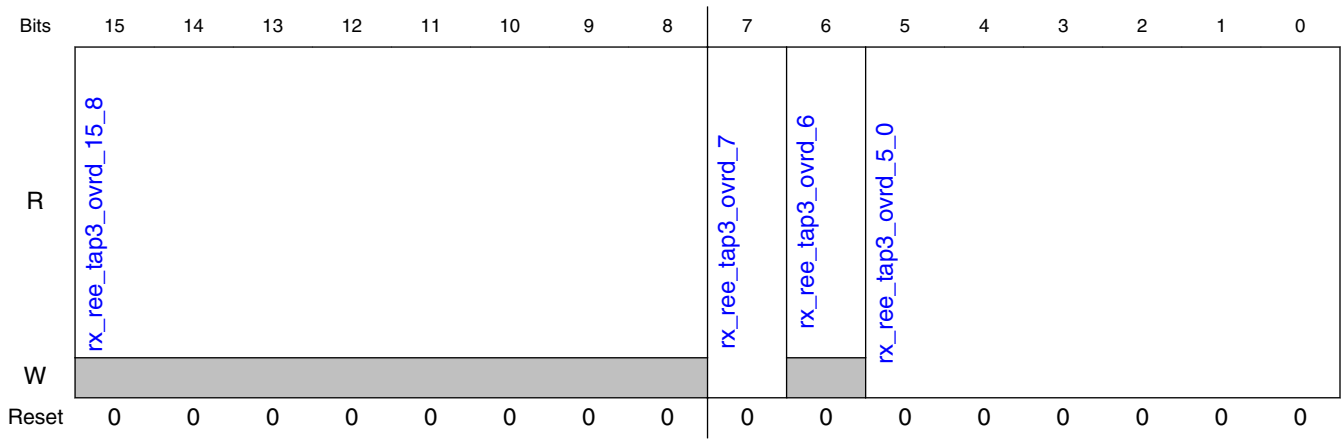
Field	Function
rx_ree_tap3_ctrl _15_12	
11 rx_ree_tap3_ctrl _11	Tap coefficient combinational logic zero crossing enable:
10 rx_ree_tap3_ctrl _10	Tap coefficient combinational logic non zero crossing enable:
9 rx_ree_tap3_ctrl _9	Tap coefficient combinational logic bit 0 only enable:
8 rx_ree_tap3_ctrl _8	Receiver DFE tap coefficient disable: This bit disables the
7 rx_ree_tap3_ctrl _7	Reserved
6-4 rx_ree_tap3_ctrl _6_4	Tap integrator accumulator scaler value: Specifies the amount to scale the input to the tap integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_tap3_ctrl _3_0	Tap sigma delta accumulator scaler value: Specifies the amount to scale the input to the tap sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.309 REE tap 3 override register (lane0_rx_ree_tap3_ovrd - lane 3_rx_ree_tap3_ovrd)

13.4.10.2.309.1 Offset

Register	Offset
lane0_rx_ree_tap3_ovrd	8189h
lane1_rx_ree_tap3_ovrd	8589h
lane2_rx_ree_tap3_ovrd	8989h
lane3_rx_ree_tap3_ovrd	8D89h

13.4.10.2.309.2 Diagram



13.4.10.2.309.3 Fields

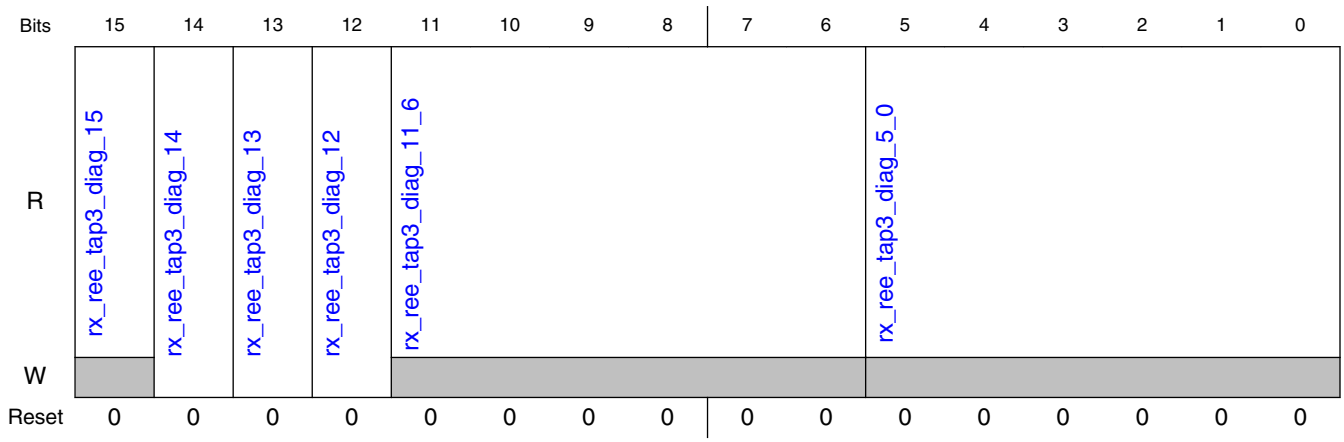
Field	Function
15-8 rx_ree_tap3_ovrd_15_8	Reserved
7 rx_ree_tap3_ovrd_7	Tap override enable: Setting this bit to a 1b1 will enable the tap override field in this register to override the tap integrator accumulator functions.
6 rx_ree_tap3_ovrd_6	Reserved
5-0 rx_ree_tap3_ovrd_5_0	Tap override value: When the tap override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.310 REE tap 3 diagnostics register (lane0_rx_ree_tap3_diag - lane3_rx_ree_tap3_diag)

13.4.10.2.310.1 Offset

Register	Offset
lane0_rx_ree_tap3_diag	818Ah
lane1_rx_ree_tap3_diag	858Ah
lane2_rx_ree_tap3_diag	898Ah
lane3_rx_ree_tap3_diag	8D8Ah

13.4.10.2.310.2 Diagram



13.4.10.2.310.3 Fields

Field	Function
15 rx_ree_tap3_diag_15	Reserved
14 rx_ree_tap3_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the tap voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_tap3_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the tap voter function to activate the voter pos signal for a single clock cycle.
12 rx_ree_tap3_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the tap.
11-6 rx_ree_tap3_diag_11_6	Reserved
5-0 rx_ree_tap3_diag_5_0	Current tap integrator accumulator: Current value of the tap integrator accumulator.

13.4.10.2.311 REE analog enable control state machine delay timer value register (lane0_rx_ree_anaensm_del_tmr - lane3_rx_ree_anaensm_del_tmr)

13.4.10.2.311.1 Offset

Register	Offset
lane0_rx_ree_anaensm_del_tmr	8194h
lane1_rx_ree_anaensm_del_tmr	8594h
lane2_rx_ree_anaensm_del_tmr	8994h
lane3_rx_ree_anaensm_del_tmr	8D94h

13.4.10.2.311.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_anaensm_del_tmr_15_0															
Reset	0	0	0	0	0	0	1	0	1	1	1	0	1	1	1	0

13.4.10.2.311.3 Fields

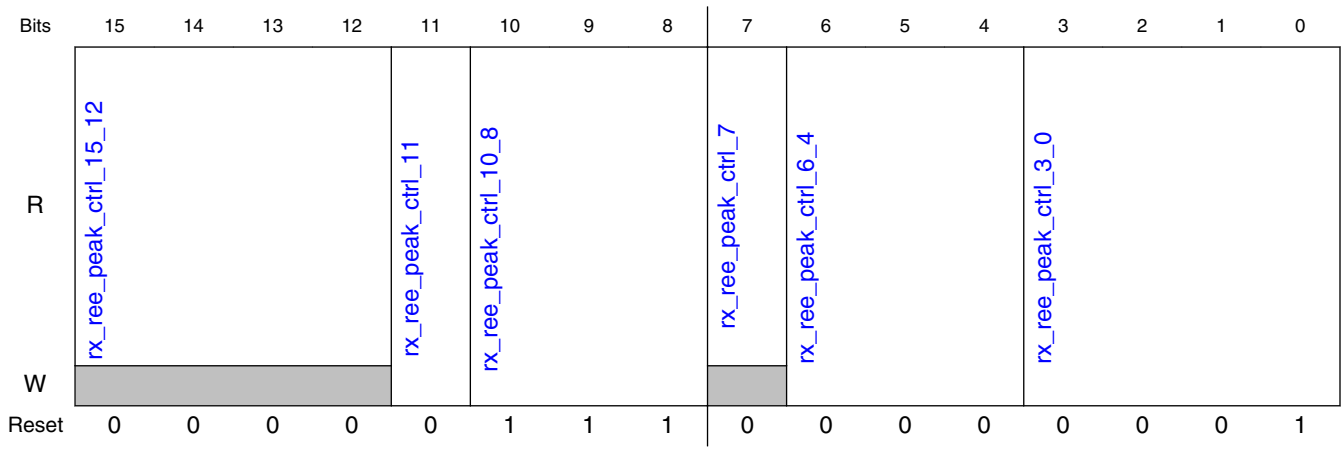
Field	Function
15-0 rx_ree_anaensm_del_tmr_15_0	Analog enable delay timer value : The number of clock cycles the state machine will wait in the Analog Enable Delay state. The time specified here is the number of clock cycles to wait between when the analog enable signal (

13.4.10.2.312 REE peaking amp control register (lane0_rx_ree_peak_ctrl - lane3_rx_ree_peak_ctrl)

13.4.10.2.312.1 Offset

Register	Offset
lane0_rx_ree_peak_ctrl	8198h
lane1_rx_ree_peak_ctrl	8598h
lane2_rx_ree_peak_ctrl	8998h
lane3_rx_ree_peak_ctrl	8D98h

13.4.10.2.312.2 Diagram



13.4.10.2.312.3 Fields

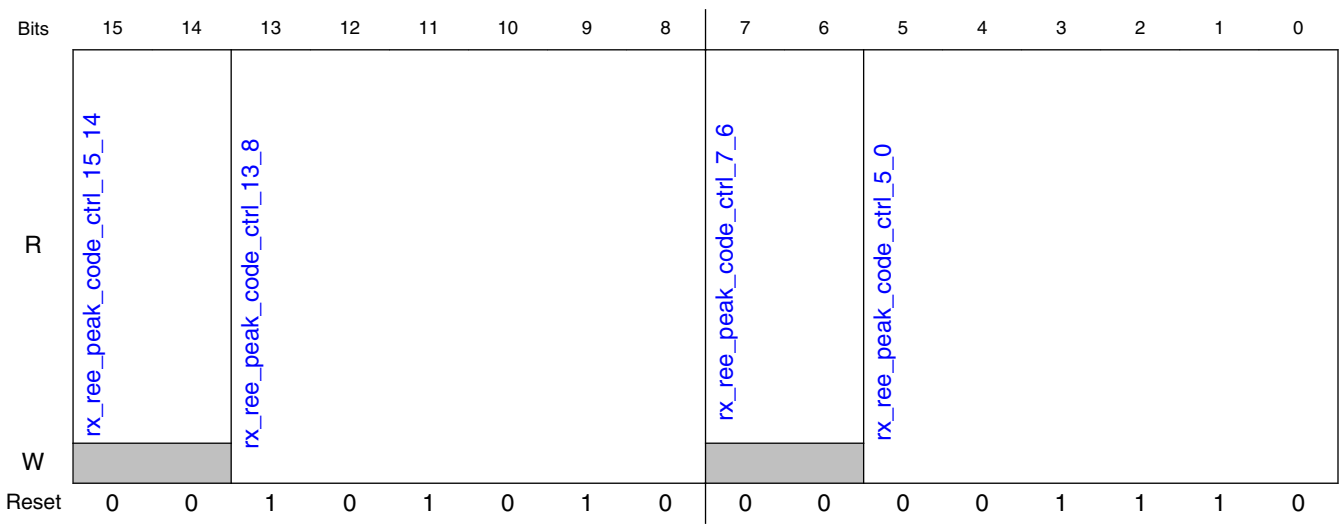
Field	Function
15-12 rx_ree_peak_ctrl_15_12	Reserved
11 rx_ree_peak_ctrl_11	Peaking amp feedback path enable: Enables the peaking amp feedback path.
10-8 rx_ree_peak_ctrl_10_8	Peaking amp feedback scaler value: Specifies the amount to scale the peaking amp feedback by. The following are the valid settings for this signal:
7 rx_ree_peak_ctrl_7	Reserved
6-4 rx_ree_peak_ctrl_6_4	Peaking amp integrator accumulator scaler value: Specifies the amount to scale the input to the peaking amp integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_peak_ctrl_3_0	Peaking amp sigma delta accumulator scaler value: Specifies the amount to scale the input to the peaking amp sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.313 REE peaking amp code control register (lane0_rx_ree_peak_code_ctrl - lane3_rx_ree_peak_code_ctrl)

13.4.10.2.313.1 Offset

Register	Offset
lane0_rx_ree_peak_code_ctrl	8199h
lane1_rx_ree_peak_code_ctrl	8599h
lane2_rx_ree_peak_code_ctrl	8999h
lane3_rx_ree_peak_code_ctrl	8D99h

13.4.10.2.313.2 Diagram



13.4.10.2.313.3 Fields

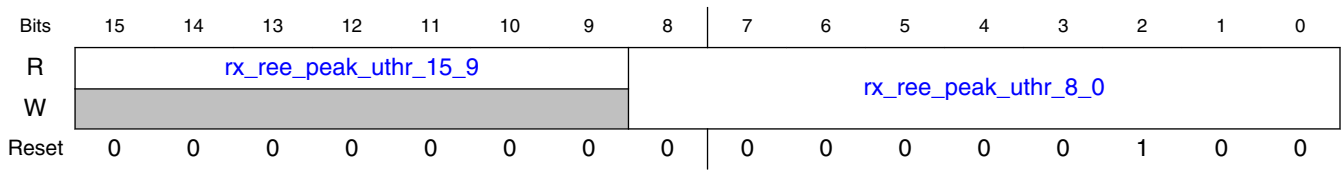
Field	Function
15-14 rx_ree_peak_code_ctrl_15_14	Reserved
13-8 rx_ree_peak_code_ctrl_13_8	Peaking amp code maximum value: This is the maximum value that the peaking amp code will be allowed to increase to.
7-6 rx_ree_peak_code_ctrl_7_6	Reserved
5-0 rx_ree_peak_code_ctrl_5_0	Peaking amp initial code: Initial value the peaking amp code is set to when training starts.

13.4.10.2.314 REE peaking amp upper threshold register (lane0_rx_ree_peak_uthr - lane3_rx_ree_peak_uthr)

13.4.10.2.314.1 Offset

Register	Offset
lane0_rx_ree_peak_uthr	819Ah
lane1_rx_ree_peak_uthr	859Ah
lane2_rx_ree_peak_uthr	899Ah
lane3_rx_ree_peak_uthr	8D9Ah

13.4.10.2.314.2 Diagram



13.4.10.2.314.3 Fields

Field	Function
15-9 rx_ree_peak_uthr_15_9	Reserved
8-0 rx_ree_peak_uthr_8_0	Peaking amp algorithm upper threshold: This is the upper threshold value used in the peaking amp algorithm.

13.4.10.2.315 REE peaking amp lower threshold register (lane0_rx_ree_peak_lthr - lane3_rx_ree_peak_lthr)

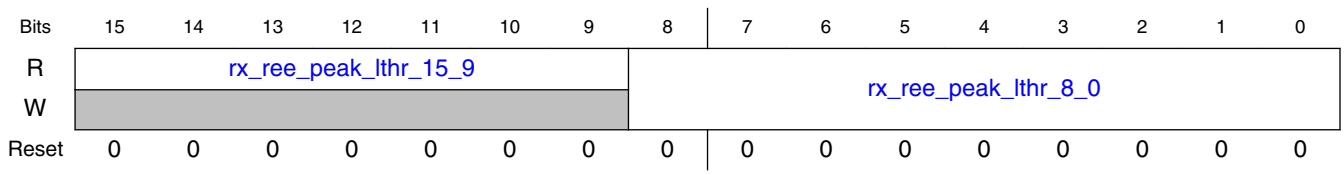
13.4.10.2.315.1 Offset

Register	Offset
lane0_rx_ree_peak_lthr	819Bh
lane1_rx_ree_peak_lthr	859Bh

Table continues on the next page...

Register	Offset
lane2_rx_ree_peak_lthr	899Bh
lane3_rx_ree_peak_lthr	8D9Bh

13.4.10.2.315.2 Diagram



13.4.10.2.315.3 Fields

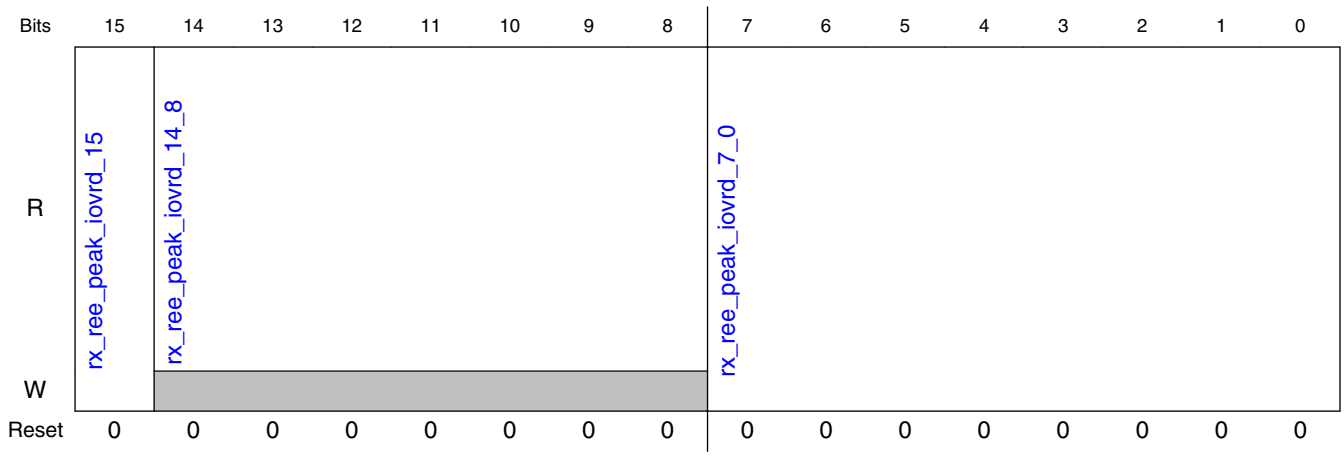
Field	Function
15-9 rx_ree_peak_lthr_15_9	Reserved
8-0 rx_ree_peak_lthr_8_0	Peaking amp algorithm lower threshold: This is the lower threshold value used in the peaking amp algorithm.

13.4.10.2.316 REE peaking amp input override register (lane0_rx_ree_peak_iovrd - lane3_rx_ree_peak_iovrd)

13.4.10.2.316.1 Offset

Register	Offset
lane0_rx_ree_peak_iovrd	819Ch
lane1_rx_ree_peak_iovrd	859Ch
lane2_rx_ree_peak_iovrd	899Ch
lane3_rx_ree_peak_iovrd	8D9Ch

13.4.10.2.316.2 Diagram



13.4.10.2.316.3 Fields

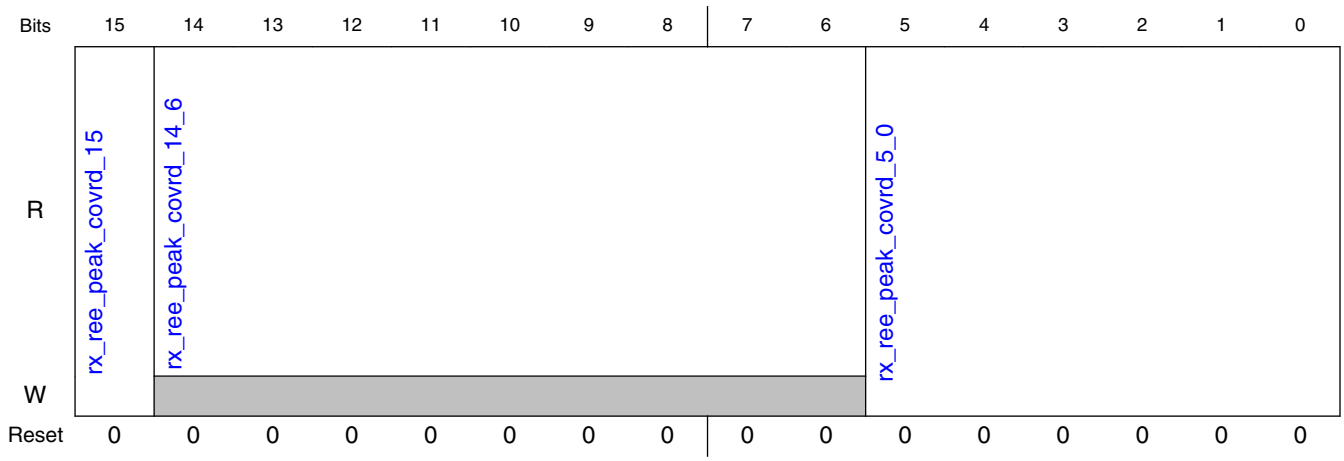
Field	Function
15 rx_ree_peak_iovrd_15	Peaking amp tap accumulator input override enable: Setting this bit to a 1b1 will allow the tap accumulator input in the peaking amp gain algorithm to be overridden by the peaking amp tap accumulator input override field in this register.
14-8 rx_ree_peak_iovrd_14_8	Reserved
7-0 rx_ree_peak_iovrd_7_0	Peaking amp tap accumulator input override : Value that will override the tap accumulator input in the peaking amp gain algorithm, when the Peaking amp tap accumulator input override enable bit is active.

13.4.10.2.317 REE peaking amp code override register (lane0_rx_ree_peak_covrd - lane3_rx_ree_peak_covrd)

13.4.10.2.317.1 Offset

Register	Offset
lane0_rx_ree_peak_covrd	819Dh
lane1_rx_ree_peak_covrd	859Dh
lane2_rx_ree_peak_covrd	899Dh
lane3_rx_ree_peak_covrd	8D9Dh

13.4.10.2.317.2 Diagram



13.4.10.2.317.3 Fields

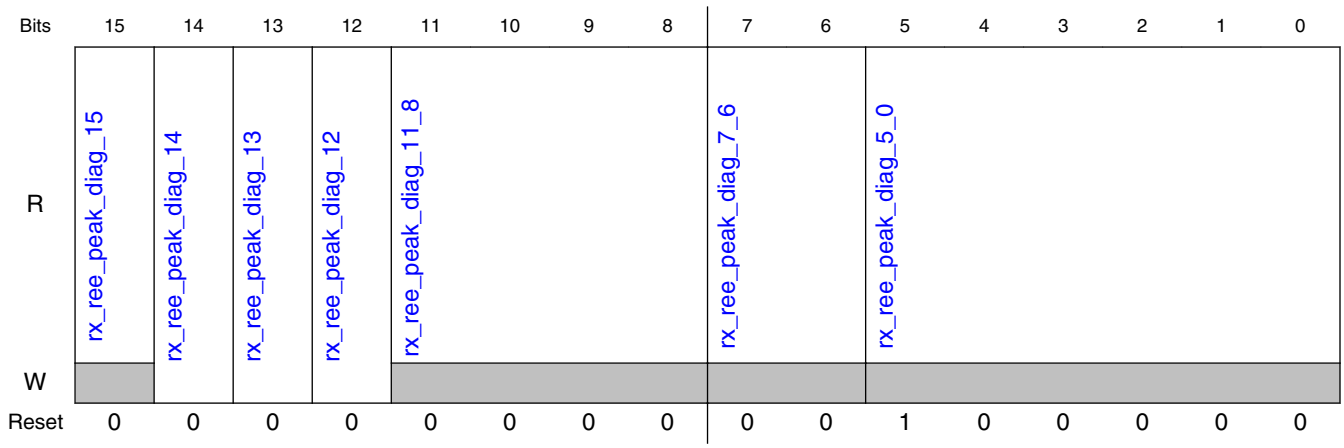
Field	Function
15 rx_ree_peak_covrd_15	Peaking amp code override enable: Setting this bit to a 1b1 will allow the peaking amp code to be overridden by the peaking amp code override value field in this register. This allows the
14-6 rx_ree_peak_covrd_14_6	Reserved
5-0 rx_ree_peak_covrd_5_0	Peaking amp code override value: Value that will override the peaking amp code when the peaking amp code override enable bit in this register is active.

13.4.10.2.318 REE peaking amp diagnostics register (lane0_rx_ree_peak_diag - lane3_rx_ree_peak_diag)

13.4.10.2.318.1 Offset

Register	Offset
lane0_rx_ree_peak_diag	819Eh
lane1_rx_ree_peak_diag	859Eh
lane2_rx_ree_peak_diag	899Eh
lane3_rx_ree_peak_diag	8D9Eh

13.4.10.2.318.2 Diagram



13.4.10.2.318.3 Fields

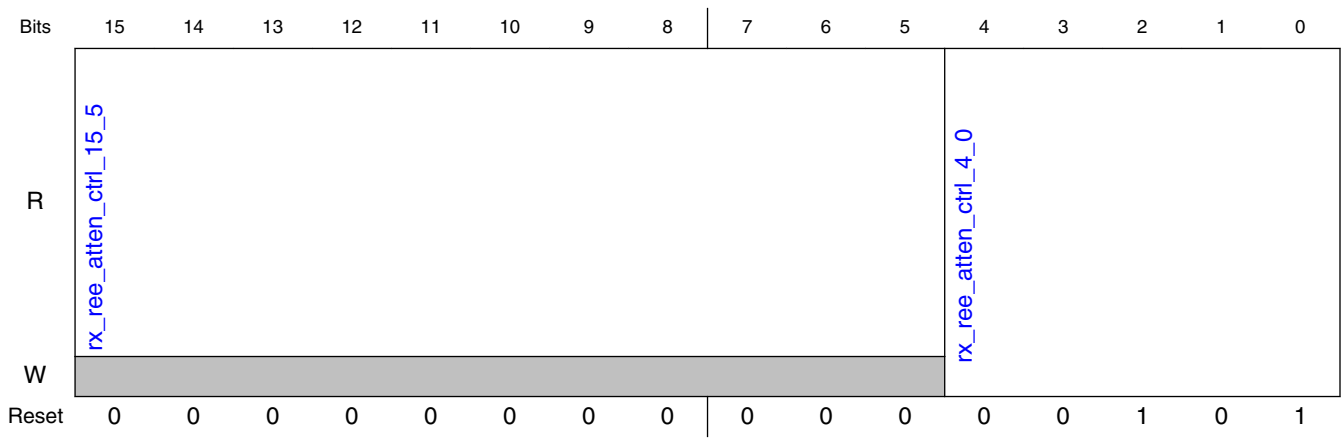
Field	Function
15 rx_ree_peak_diag_15	Reserved
14 rx_ree_peak_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the peaking amp voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_peak_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the peaking amp voter function to activate the voter pos signal for a single clock cycle.
12 rx_ree_peak_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the peaking amp.
11-8 rx_ree_peak_diag_11_8	Reserved
7-6 rx_ree_peak_diag_7_6	Reserved
5-0 rx_ree_peak_diag_5_0	Current peaking amp integrator accumulator: Current value of the tap integrator accumulator, without the unused sign bit.

13.4.10.2.319 REE attenuation control register (lane0_rx_ree_atten_ctrl - lane3_rx_ree_atten_ctrl)

13.4.10.2.319.1 Offset

Register	Offset
lane0_rx_ree_atten_ctrl	81A0h
lane1_rx_ree_atten_ctrl	85A0h
lane2_rx_ree_atten_ctrl	89A0h
lane3_rx_ree_atten_ctrl	8DA0h

13.4.10.2.319.2 Diagram



13.4.10.2.319.3 Fields

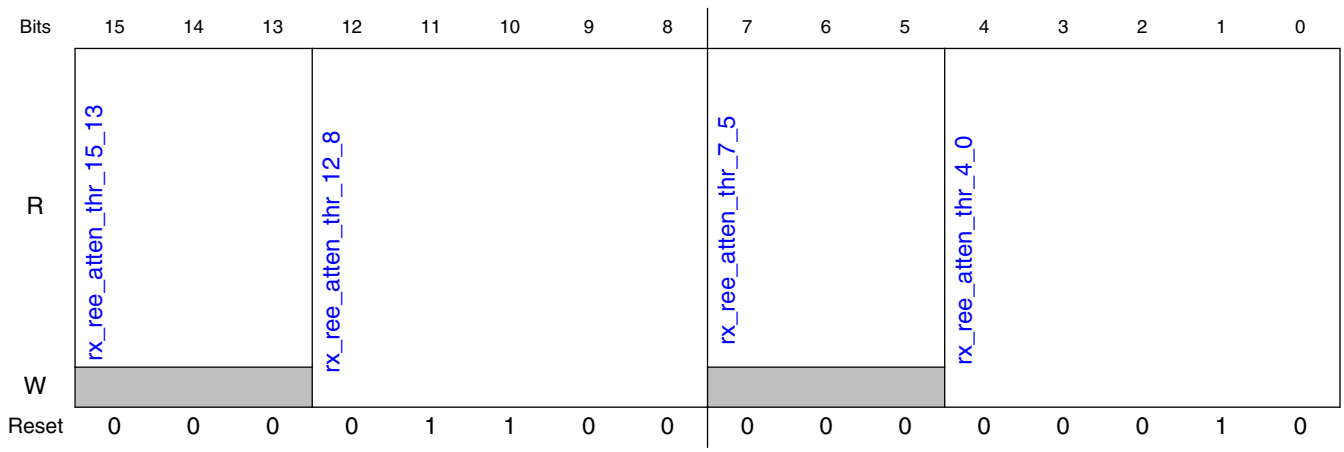
Field	Function
15-5 rx_ree_atten_ctrl_15_5	Reserved
4-0 rx_ree_atten_ctrl_4_0	Receiver DFE attenuation maximum value: The maximum value the

13.4.10.2.320 REE attenuation threshold register (lane0_rx_ree_atten_thr - lane3_rx_ree_atten_thr)

13.4.10.2.320.1 Offset

Register	Offset
lane0_rx_ree_atten_thr	81A1h
lane1_rx_ree_atten_thr	85A1h
lane2_rx_ree_atten_thr	89A1h
lane3_rx_ree_atten_thr	8DA1h

13.4.10.2.320.2 Diagram



13.4.10.2.320.3 Fields

Field	Function
15-13 rx_ree_atten_thr_15_13	Reserved
12-8 rx_ree_atten_thr_12_8	Attenuation high threshold value: High threshold value to compare against the VGA gain accumulator value. Note that the value on this field is not a twos complement value (There is no sign bit and it is always a positive number).
7-5 rx_ree_atten_thr_7_5	Reserved
4-0 rx_ree_atten_thr_4_0	Attenuation low threshold value: Low threshold value to compare against the VGA gain accumulator value. Note that the value on this field is not a twos complement value (There is no sign bit and it is always a positive number).

13.4.10.2.321 REE attenuation counter register (lane0_rx_ree_atten_cnt - lane3_rx_ree_atten_cnt)

13.4.10.2.321.1 Offset

Register	Offset
lane0_rx_ree_atten_cnt	81A2h
lane1_rx_ree_atten_cnt	85A2h
lane2_rx_ree_atten_cnt	89A2h
lane3_rx_ree_atten_cnt	8DA2h

13.4.10.2.321.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ree_atten_cnt_15_0															
Reset	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0

13.4.10.2.321.3 Fields

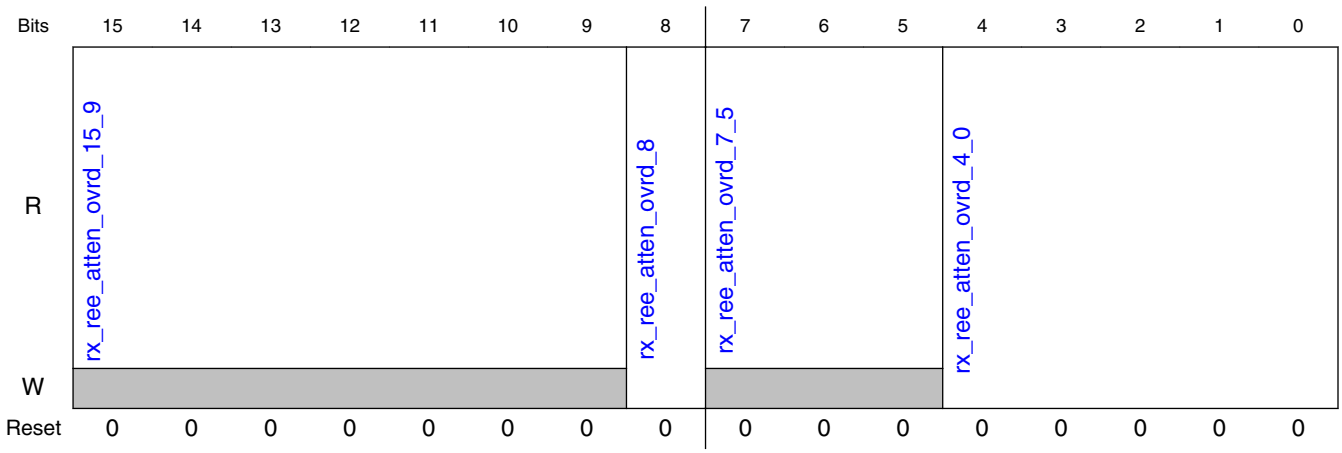
Field	Function
15-0 rx_ree_atten_cnt_15_0	Attenuation counter max: Value used to specify the maximum number of consecutive words above or below the specified thresholds which will result in triggering an increase or decrease in the

13.4.10.2.322 REE attenuation override register (lane0_rx_ree_atten_ovrd - lane3_rx_ree_atten_ovrd)

13.4.10.2.322.1 Offset

Register	Offset
lane0_rx_ree_atten_ovrd	81A3h
lane1_rx_ree_atten_ovrd	85A3h
lane2_rx_ree_atten_ovrd	89A3h
lane3_rx_ree_atten_ovrd	8DA3h

13.4.10.2.322.2 Diagram



13.4.10.2.322.3 Fields

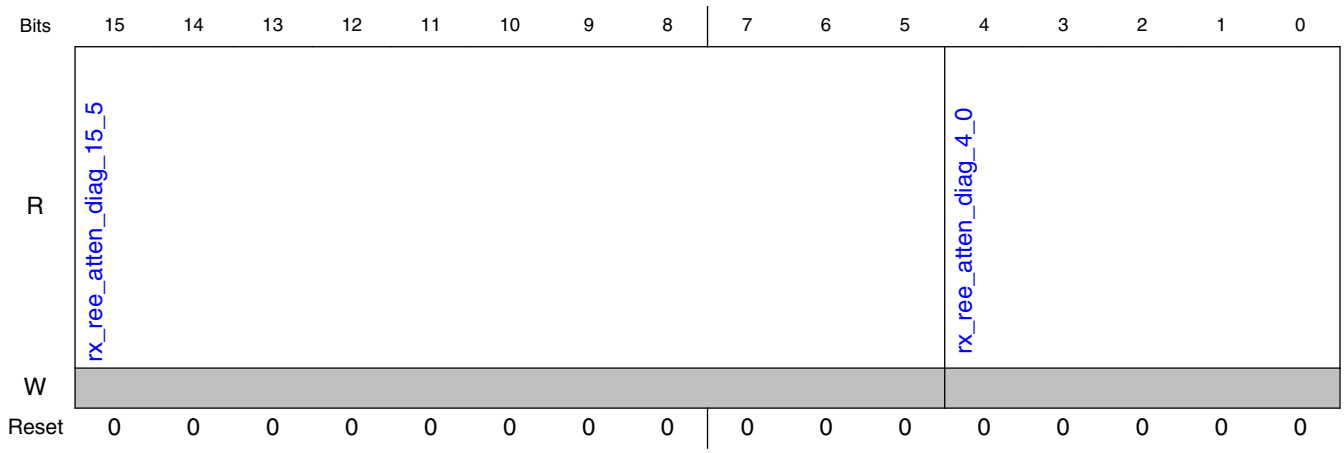
Field	Function
15-9 rx_ree_atten_ovrd_15_9	Reserved
8 rx_ree_atten_ovrd_8	Attenuation override enable: Setting this bit to a 1b1 will allow the
7-5 rx_ree_atten_ovrd_7_5	Reserved
4-0 rx_ree_atten_ovrd_4_0	Attenuation override value: When enabled by the attenuation override enable bit in this register, this value will override the current attenuation value on the

13.4.10.2.323 REE attenuation diagnostics register (lane0_rx_ree_atten_diag - lane3_rx_ree_atten_diag)

13.4.10.2.323.1 Offset

Register	Offset
lane0_rx_ree_atten_diag	81A4h
lane1_rx_ree_atten_diag	85A4h
lane2_rx_ree_atten_diag	89A4h
lane3_rx_ree_atten_diag	8DA4h

13.4.10.2.323.2 Diagram



13.4.10.2.323.3 Fields

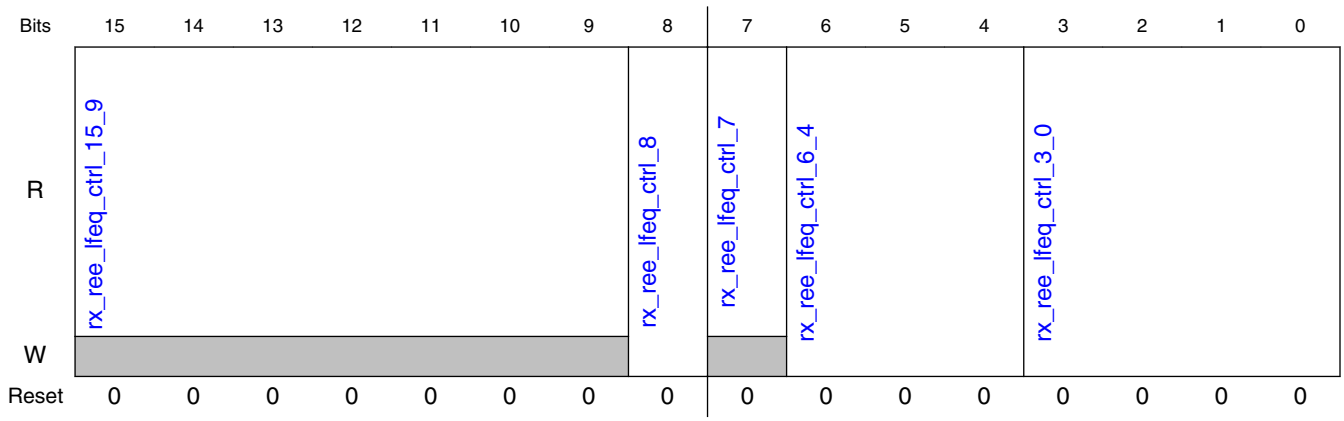
Field	Function
15-5 rx_ree_atten_diag_15_5	Reserved
4-0 rx_ree_atten_diag_4_0	Current attenuation value: Current value of the attenuation.

13.4.10.2.324 REE low frequency equalizer control register (lane0_rx_ree_lfeq_ctrl - lane3_rx_ree_lfeq_ctrl)

13.4.10.2.324.1 Offset

Register	Offset
lane0_rx_ree_lfeq_ctrl	81A8h
lane1_rx_ree_lfeq_ctrl	85A8h
lane2_rx_ree_lfeq_ctrl	89A8h
lane3_rx_ree_lfeq_ctrl	8DA8h

13.4.10.2.324.2 Diagram



13.4.10.2.324.3 Fields

Field	Function
15-9 rx_ree_lfeq_ctrl_15_9	Reserved
8 rx_ree_lfeq_ctrl_8	Receiver DFE coefficient disable: This bit disables the
7 rx_ree_lfeq_ctrl_7	Reserved
6-4 rx_ree_lfeq_ctrl_6_4	Integrator accumulator scaler value: Specifies the amount to scale the input to the integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_lfeq_ctrl_3_0	Sigma delta accumulator scaler value: Specifies the amount to scale the input to the sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.325 REE low frequency equalizer override register (lane0_rx_ree_lfeq_ovrd - lane3_rx_ree_lfeq_ovrd)

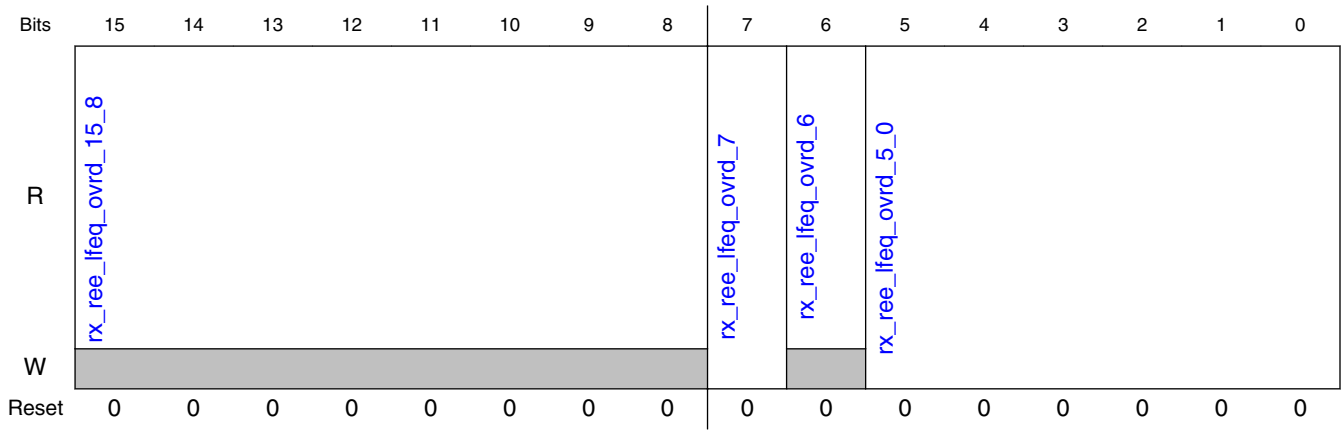
13.4.10.2.325.1 Offset

Register	Offset
lane0_rx_ree_lfeq_ovrd	81A9h
lane1_rx_ree_lfeq_ovrd	85A9h

Table continues on the next page...

Register	Offset
lane2_rx_ree_lfeq_ovrd	89A9h
lane3_rx_ree_lfeq_ovrd	8DA9h

13.4.10.2.325.2 Diagram



13.4.10.2.325.3 Fields

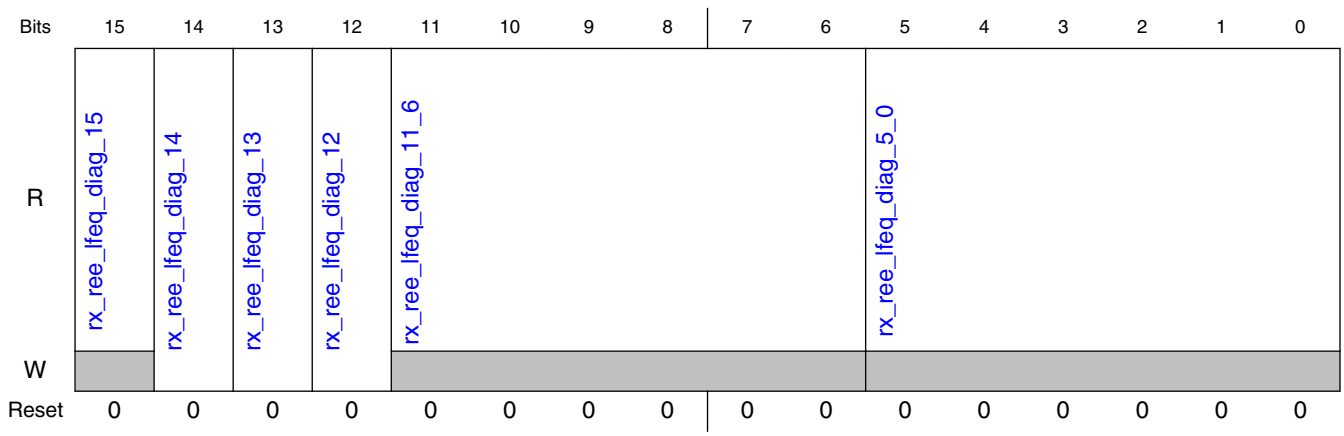
Field	Function
15-8 rx_ree_lfeq_ovrd_15_8	Reserved
7 rx_ree_lfeq_ovrd_7	Override enable: Setting this bit to a 1b1 will enable the override field in this register to override the integrator accumulator functions.
6 rx_ree_lfeq_ovrd_6	Reserved
5-0 rx_ree_lfeq_ovrd_5_0	Override value: When the override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.326 REE low frequency equalizer diagnostics register (lane0_rx_ree_lfeq_diag - lane3_rx_ree_lfeq_diag)

13.4.10.2.326.1 Offset

Register	Offset
lane0_rx_ree_lfeq_diag	81AAh
lane1_rx_ree_lfeq_diag	85AAh
lane2_rx_ree_lfeq_diag	89AAh
lane3_rx_ree_lfeq_diag	8DAAh

13.4.10.2.326.2 Diagram



13.4.10.2.326.3 Fields

Field	Function
15 rx_ree_lfeq_diag_15	Reserved
14 rx_ree_lfeq_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_lfeq_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the voter function to activate the voter pos signal for a single cycle.
12 rx_ree_lfeq_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter.
11-6 rx_ree_lfeq_diag_11_6	Reserved
5-0	Current integrator accumulator: Current value of the integrator accumulator.

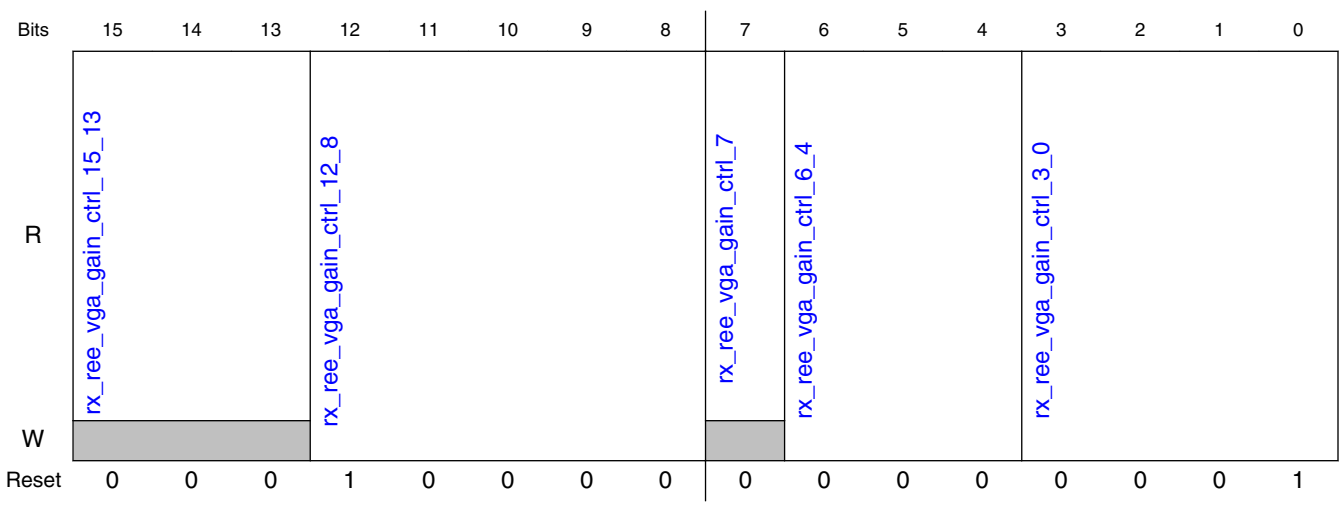
Field	Function
rx_ree_lfeq_diag_5_0	

13.4.10.2.327 REE VGA gain control register (lane0_rx_ree_vga_gain_ctrl - lane3_rx_ree_vga_gain_ctrl)

13.4.10.2.327.1 Offset

Register	Offset
lane0_rx_ree_vga_gain_ctrl	81ACh
lane1_rx_ree_vga_gain_ctrl	85ACh
lane2_rx_ree_vga_gain_ctrl	89ACh
lane3_rx_ree_vga_gain_ctrl	8DACh

13.4.10.2.327.2 Diagram



13.4.10.2.327.3 Fields

Field	Function
15-13	Reserved

Table continues on the next page...

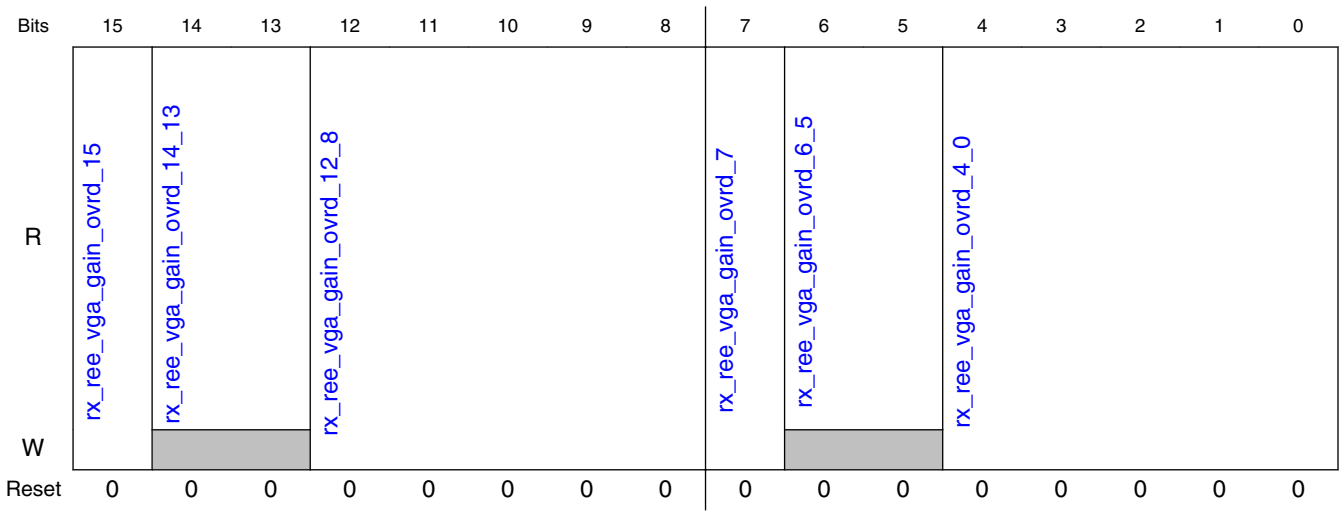
Field	Function
rx_ree_vga_gain_ctrl_15_13	
12-8 rx_ree_vga_gain_ctrl_12_8	VGA gain max: Specifies the maximum value of the VGA gain integrator accumulator, and therefore also the maximum number of bits in the rxda_dfe_vga_gain thermometer code that will be set.
7 rx_ree_vga_gain_ctrl_7	Reserved
6-4 rx_ree_vga_gain_ctrl_6_4	VGA gain integrator accumulator scaler value: Specifies the amount to scale the input to the VGA gain integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_vga_gain_ctrl_3_0	VGA gain sigma delta accumulator scaler value: Specifies the amount to scale the input to the VGA gain sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.328 REE VGA gain override register (lane0_rx_ree_vga_gain_ovrd - lane3_rx_ree_vga_gain_ovrd)

13.4.10.2.328.1 Offset

Register	Offset
lane0_rx_ree_vga_gain_ovrd	81ADh
lane1_rx_ree_vga_gain_ovrd	85ADh
lane2_rx_ree_vga_gain_ovrd	89ADh
lane3_rx_ree_vga_gain_ovrd	8DADh

13.4.10.2.328.2 Diagram



13.4.10.2.328.3 Fields

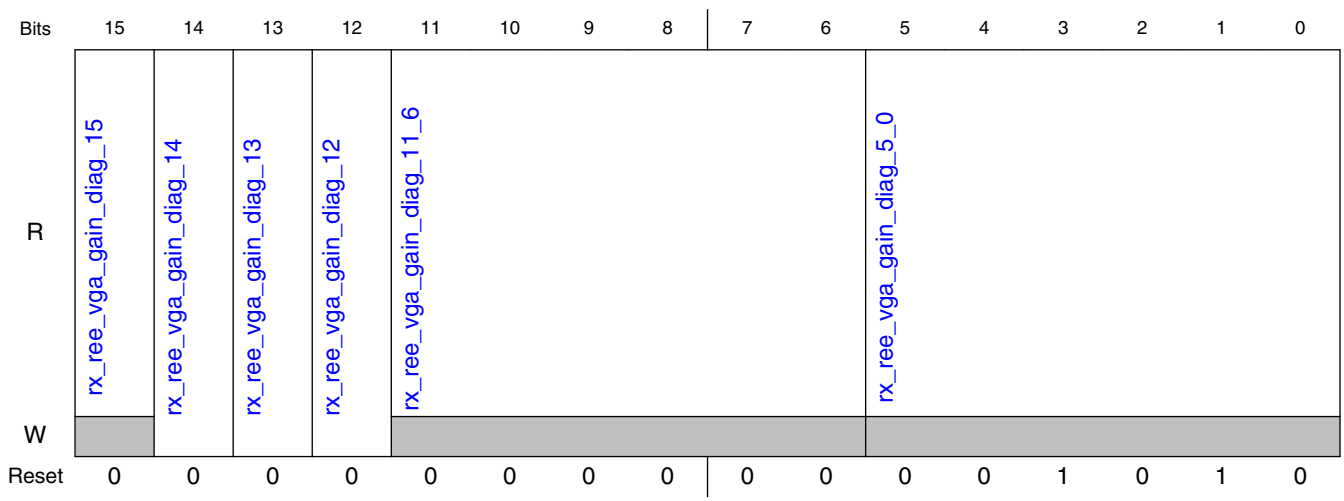
Field	Function
15 rx_ree_vga_gain_ovrd_15	VGA gain target adjust override enable: Setting this bit to a 1b1 will enable the VGA gain target adjust override field in this register to override the VGA gain target adjust accumulator functions.
14-13 rx_ree_vga_gain_ovrd_14_13	Reserved
12-8 rx_ree_vga_gain_ovrd_12_8	VGA gain target adjust override value: When the VGA gain target adjust override enable bit in this register is active, the value in this field will override the accumulator value used to drive the
7 rx_ree_vga_gain_ovrd_7	VGA gain override enable: Setting this bit to a 1b1 will enable the VGA gain override field in this register to override the VGA gain integrator accumulator functions.
6-5 rx_ree_vga_gain_ovrd_6_5	Reserved
4-0 rx_ree_vga_gain_ovrd_4_0	VGA gain override value: When the VGA gain override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.329 REE VGA gain diagnostics register (lane0_rx_ree_vga_gain_diag - lane3_rx_ree_vga_gain_diag)

13.4.10.2.329.1 Offset

Register	Offset
lane0_rx_ree_vga_gain_diag	81AEh
lane1_rx_ree_vga_gain_diag	85AEh
lane2_rx_ree_vga_gain_diag	89AEh
lane3_rx_ree_vga_gain_diag	8DAEh

13.4.10.2.329.2 Diagram



13.4.10.2.329.3 Fields

Field	Function
15 rx_ree_vga_gain_diag_15	Reserved
14 rx_ree_vga_gain_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the VGA gain voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_vga_gain_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the VGA gain voter function to activate the voter pos signal for a single clock cycle.
12 rx_ree_vga_gain_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the VGA gain.

Table continues on the next page...

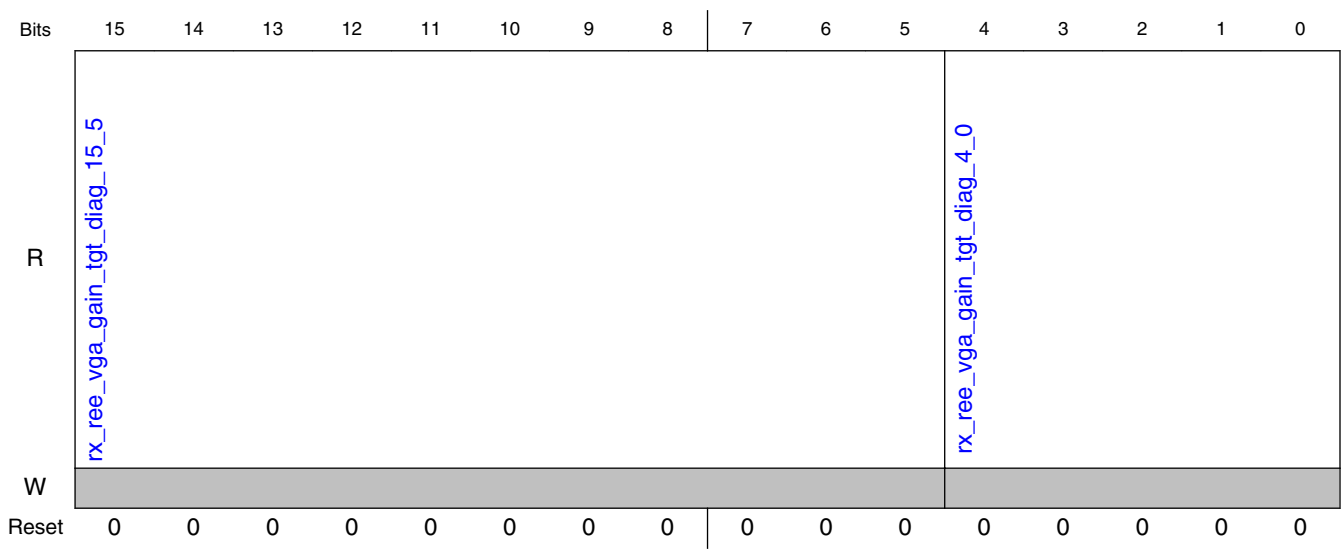
Field	Function
11-6 rx_ree_vga_gain_diag_11_6	Reserved
5-0 rx_ree_vga_gain_diag_5_0	Current VGA gain integrator accumulator: Current value of the VGA gain integrator accumulator.

13.4.10.2.330 REE VGA gain target adjust diagnostics register (lane0_rx_ree_vga_gain_tgt_diag - lane3_rx_ree_vga_gain_tgt_diag)

13.4.10.2.330.1 Offset

Register	Offset
lane0_rx_ree_vga_gain_tgt_diag	81AFh
lane1_rx_ree_vga_gain_tgt_diag	85AFh
lane2_rx_ree_vga_gain_tgt_diag	89AFh
lane3_rx_ree_vga_gain_tgt_diag	8DAFh

13.4.10.2.330.2 Diagram



13.4.10.2.330.3 Fields

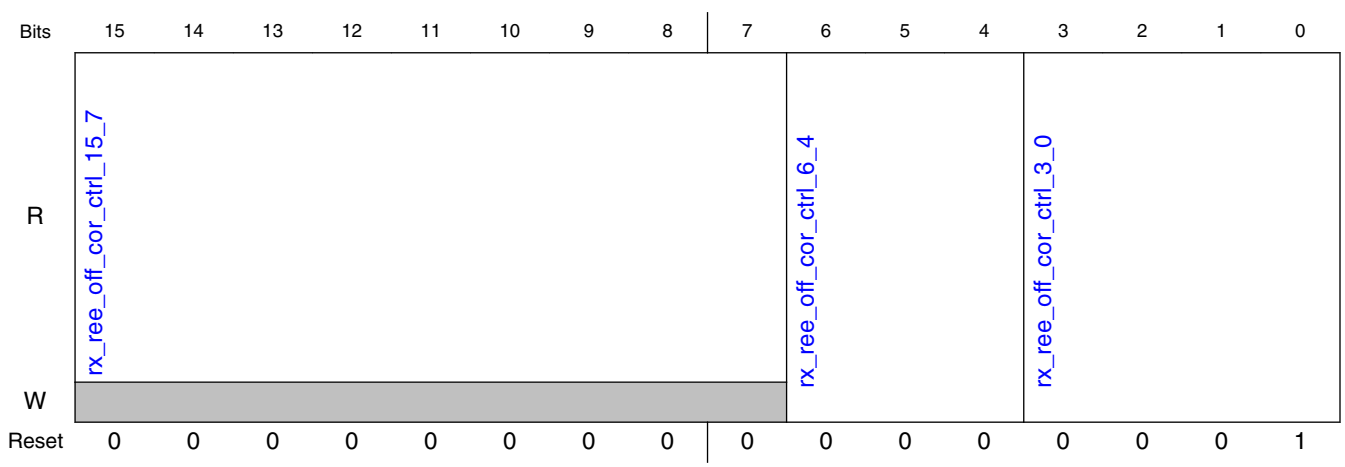
Field	Function
15-5 rx_ree_vga_gain_tgt_diag_15_5	Reserved
4-0 rx_ree_vga_gain_tgt_diag_4_0	Current VGA gain integrator accumulator: Current value of the VGA gain integrator accumulator.

13.4.10.2.331 REE offset correction control register (lane0_rx_ree_off_cor_ctrl - lane3_rx_ree_off_cor_ctrl)

13.4.10.2.331.1 Offset

Register	Offset
lane0_rx_ree_off_cor_ctrl	81B0h
lane1_rx_ree_off_cor_ctrl	85B0h
lane2_rx_ree_off_cor_ctrl	89B0h
lane3_rx_ree_off_cor_ctrl	8DB0h

13.4.10.2.331.2 Diagram



13.4.10.2.331.3 Fields

Field	Function
15-7	Reserved

Table continues on the next page...

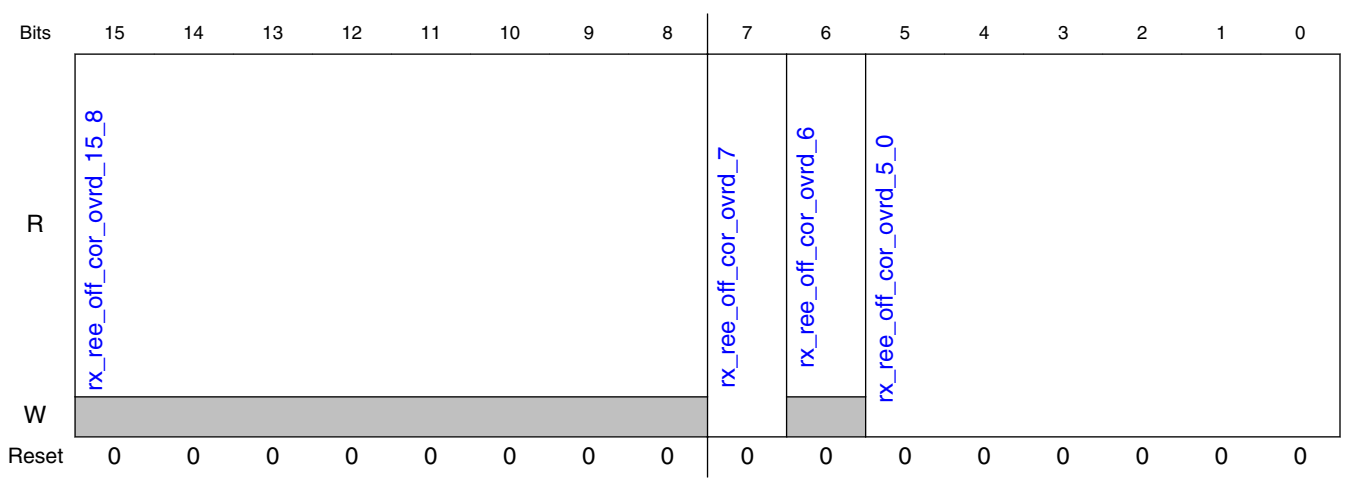
Field	Function
rx_ree_off_cor_ctrl_15_7	
6-4 rx_ree_off_cor_ctrl_6_4	Offset correction integrator accumulator scaler value: Specifies the amount to scale the input to the offset correction integrator accumulator by. The following are the valid settings for this field:
3-0 rx_ree_off_cor_ctrl_3_0	Offset correction sigma delta accumulator scaler value: Specifies the amount to scale the input to the offset correction sigma delta accumulator by. The following are the valid settings for this field:

13.4.10.2.332 REE offset correction override register (lane0_rx_ree_off_cor_ovrd - lane3_rx_ree_off_cor_ovrd)

13.4.10.2.332.1 Offset

Register	Offset
lane0_rx_ree_off_cor_ovrd	81B1h
lane1_rx_ree_off_cor_ovrd	85B1h
lane2_rx_ree_off_cor_ovrd	89B1h
lane3_rx_ree_off_cor_ovrd	8DB1h

13.4.10.2.332.2 Diagram



13.4.10.2.332.3 Fields

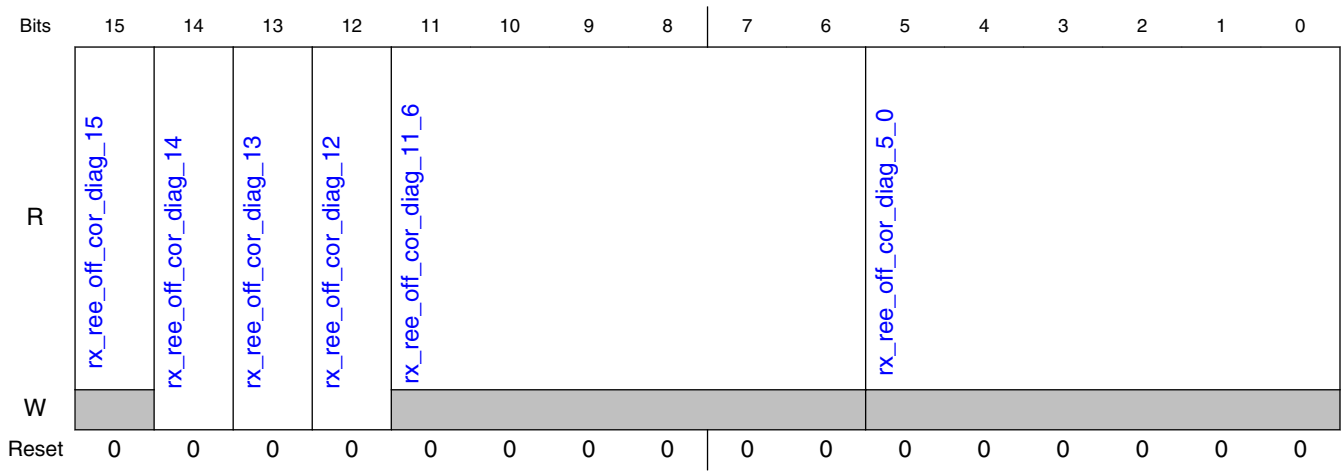
Field	Function
15-8 rx_ree_off_cor_ovrd_15_8	Reserved
7 rx_ree_off_cor_ovrd_7	Offset correction override enable: Setting this bit to a 1b1 will enable the offset correction override field in this register to override the offset correction integrator accumulator functions.
6 rx_ree_off_cor_ovrd_6	Reserved
5-0 rx_ree_off_cor_ovrd_5_0	Offset correction override value: When the offset correction override enable bit in this register is active, the value in this field will override the integrator accumulator value, as well as the input to the binary to thermometer encoder.

13.4.10.2.333 REE offset correction diagnostics register (lane0_rx_ree_off_cor_diag - lane3_rx_ree_off_cor_diag)

13.4.10.2.333.1 Offset

Register	Offset
lane0_rx_ree_off_cor_diag	81B2h
lane1_rx_ree_off_cor_diag	85B2h
lane2_rx_ree_off_cor_diag	89B2h
lane3_rx_ree_off_cor_diag	8DB2h

13.4.10.2.333.2 Diagram



13.4.10.2.333.3 Fields

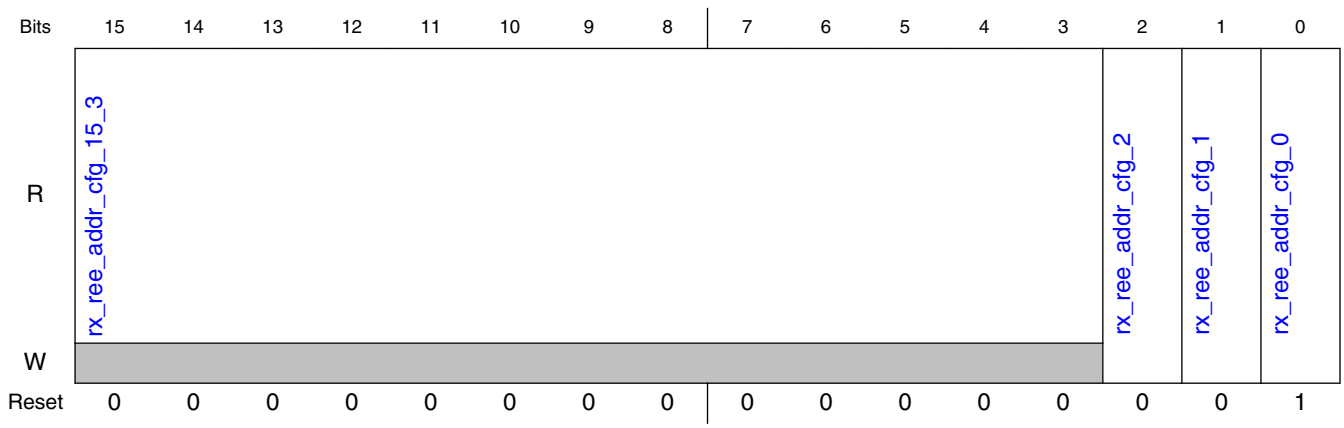
Field	Function
15 rx_ree_off_cor_diag_15	Reserved
14 rx_ree_off_cor_diag_14	Voter override neg : Writing a 1b1 in this register bit will force the offset correction voter function to activate the voter neg signal for a single clock cycle.
13 rx_ree_off_cor_diag_13	Voter override pos : Writing a 1b1 in this register bit will force the offset correction voter function to activate the voter pos signal for a single clock cycle.
12 rx_ree_off_cor_diag_12	Voter override enable : Setting this bit to a 1b1 will allow only the voter override pos and voter override neg bits in this register to control the voter in the offset correction.
11-6 rx_ree_off_cor_diag_11_6	Reserved
5-0 rx_ree_off_cor_diag_5_0	Current offset correction integrator accumulator: Current value of the offset correction integrator accumulator.

13.4.10.2.334 REE adder configuration register (lane0_rx_ree_addr_cfg - lane3_rx_ree_addr_cfg)

13.4.10.2.334.1 Offset

Register	Offset
lane0_rx_ree_addr_cfg	81B8h
lane1_rx_ree_addr_cfg	85B8h
lane2_rx_ree_addr_cfg	89B8h
lane3_rx_ree_addr_cfg	8DB8h

13.4.10.2.334.2 Diagram



13.4.10.2.334.3 Fields

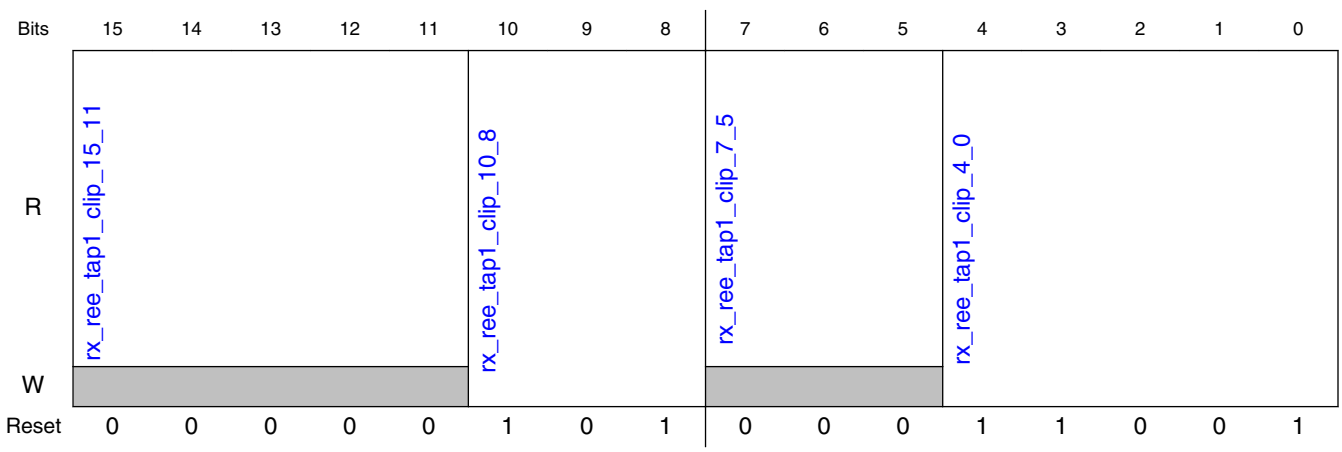
Field	Function
15-3 rx_ree_addr_cfg_15_3	Reserved
2 rx_ree_addr_cfg_2	RX peaking tap 3 adder enable: Setting this bit to 1b1, enables the results of tap 3 to be added to the RX peaking amp gain input.
1 rx_ree_addr_cfg_1	RX peaking tap 2 adder enable: Setting this bit to 1b1, enables the results of tap 2 to be added to the RX peaking amp gain input.
0 rx_ree_addr_cfg_0	RX peaking tap 1 adder enable: Setting this bit to 1b1, enables the results of tap 1 to be added to the RX peaking amp gain input.

13.4.10.2.335 REE tap 1 clip control register (lane0_rx_ree_tap1_clip - lane 3_rx_ree_tap1_clip)

13.4.10.2.335.1 Offset

Register	Offset
lane0_rx_ree_tap1_clip	81B9h
lane1_rx_ree_tap1_clip	85B9h
lane2_rx_ree_tap1_clip	89B9h
lane3_rx_ree_tap1_clip	8DB9h

13.4.10.2.335.2 Diagram



13.4.10.2.335.3 Fields

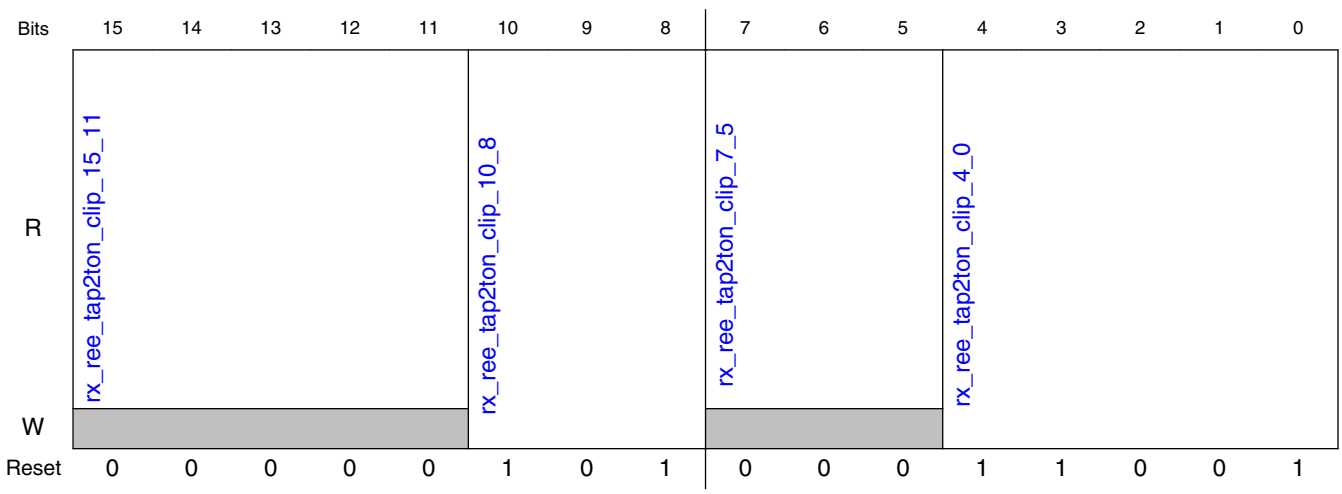
Field	Function
15-11 rx_ree_tap1_clip_15_11	Reserved
10-8 rx_ree_tap1_clip_10_8	VGA target gain adjust multiplier: Controls how much to multiply the VGA target gain adjust by, when calculating the tap threshold. The following are valid values, and the corresponding multiplier values.
7-5 rx_ree_tap1_clip_7_5	Reserved
4-0 rx_ree_tap1_clip_4_0	Threshold adjust: Controls how much the threshold can be adjusted by, after multiplying the VGA target gain adjust multiplier with the VGA target gain adjust. Note that this field is a positive number (not twos complement).

13.4.10.2.336 REE taps 2 and 3 clip control register (lane0_rx_ree_tap2ton_clip - lane3_rx_ree_tap2ton_clip)

13.4.10.2.336.1 Offset

Register	Offset
lane0_rx_ree_tap2ton_clip	81BAh
lane1_rx_ree_tap2ton_clip	85BAh
lane2_rx_ree_tap2ton_clip	89BAh
lane3_rx_ree_tap2ton_clip	8DBAh

13.4.10.2.336.2 Diagram



13.4.10.2.336.3 Fields

Field	Function
15-11 rx_ree_tap2ton_clip_15_11	Reserved
10-8 rx_ree_tap2ton_clip_10_8	VGA target gain adjust multiplier: Controls how much to multiply the VGA target gain adjust by, when calculating the tap threshold. The following are valid values, and the corresponding multiplier values.
7-5	Reserved

Table continues on the next page...

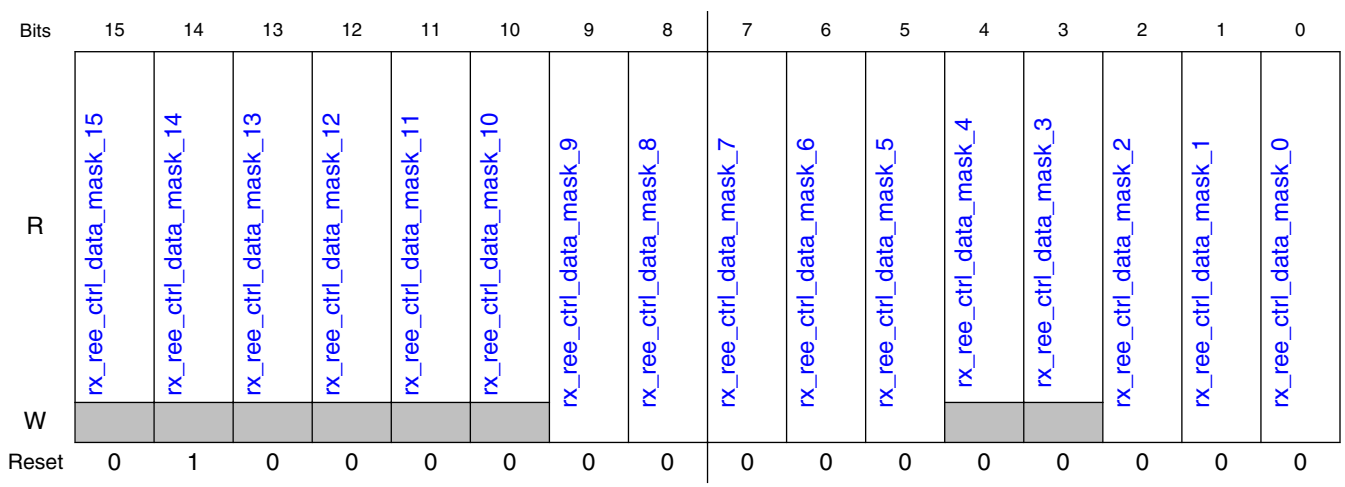
Field	Function
rx_ree_tap2ton_clip_7_5	
4-0 rx_ree_tap2ton_clip_4_0	Threshold adjust: Controls how much the threshold can be adjusted by, after multiplying the VGA target gain adjust multiplier with the VGA target gain adjust. Note that this field is a positive number (not twos complement).

13.4.10.2.337 REE control data mask register (lane0_rx_ree_ctrl_data_mask - lane3_rx_ree_ctrl_data_mask)

13.4.10.2.337.1 Offset

Register	Offset
lane0_rx_ree_ctrl_data_mask	81BBh
lane1_rx_ree_ctrl_data_mask	85BBh
lane2_rx_ree_ctrl_data_mask	89BBh
lane3_rx_ree_ctrl_data_mask	8DBBh

13.4.10.2.337.2 Diagram



13.4.10.2.337.3 Fields

Field	Function
15 rx_ree_ctrl_data_mask_15	Reserved
14 rx_ree_ctrl_data_mask_14	Ignore 1010 controller - Note that this is read only. This should never be disabled.
13 rx_ree_ctrl_data_mask_13	Reserved
12 rx_ree_ctrl_data_mask_12	Reserved
11 rx_ree_ctrl_data_mask_11	Reserved
10 rx_ree_ctrl_data_mask_10	Reserved
9 rx_ree_ctrl_data_mask_9	RX attenuation: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
8 rx_ree_ctrl_data_mask_8	RX VGA gain: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
7 rx_ree_ctrl_data_mask_7	RX offset correction coefficient: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
6 rx_ree_ctrl_data_mask_6	RX peaking amp gain: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
5 rx_ree_ctrl_data_mask_5	RX low frequency equalizer adaptive control: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
4 rx_ree_ctrl_data_mask_4	Reserved
3 rx_ree_ctrl_data_mask_3	Reserved
2 rx_ree_ctrl_data_mask_2	RX tap 3: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the

Table continues on the next page...

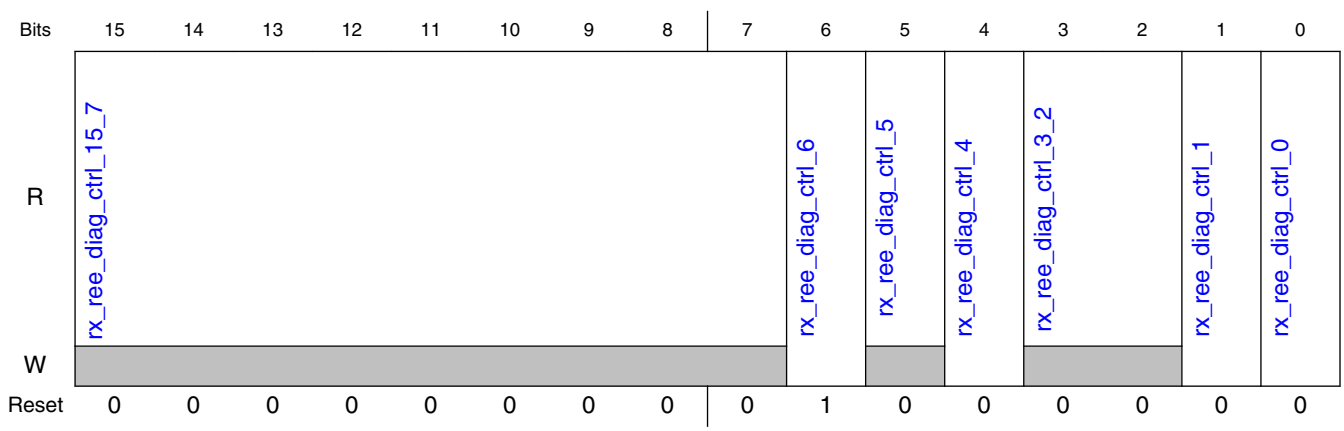
Field	Function
1 rx_ree_ctrl_data_mask_1	RX tap 2: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the
0 rx_ree_ctrl_data_mask_0	RX tap 1: When set to 1b0, this REE component will be turned off when control data is being received, as indicated by either the

13.4.10.2.338 REE diagnostic control register (lane0_rx_ree_diag_ctrl - lane3_rx_ree_diag_ctrl)

13.4.10.2.338.1 Offset

Register	Offset
lane0_rx_ree_diag_ctrl	81BCh
lane1_rx_ree_diag_ctrl	85BCh
lane2_rx_ree_diag_ctrl	89BCh
lane3_rx_ree_diag_ctrl	8DBCh

13.4.10.2.338.2 Diagram



13.4.10.2.338.3 Fields

Field	Function
15-7 rx_ree_diag_ctrl_15_7	Reserved

Table continues on the next page...

Field	Function
6 rx_ree_diag_ctrl_6	Hold periodic equalization while RX idle: When this bit is set to 1b1, a detection of electrical idle on the receiver (
5 rx_ree_diag_ctrl_5	Reserved
4 rx_ree_diag_ctrl_4	Hold gen 2 equalization while RX idle: When this bit is set to 1b1, a detection of electrical idle on the receiver (
3-2 rx_ree_diag_ctrl_3_2	Reserved
1 rx_ree_diag_ctrl_1	Force REE controller clock on : When active, the REE controller clock gate will allow the clock to run.
0 rx_ree_diag_ctrl_0	Force REE function clock on : When active, the REE function clock gate will allow the clock to run.

13.4.10.2.339 REE control state machine gen mode control register 1 (lane 0_rx_ree_smgm_ctrl1 - lane3_rx_ree_smgm_ctrl1)

13.4.10.2.339.1 Offset

Register	Offset
lane0_rx_ree_smgm_ctrl1	81BDh
lane1_rx_ree_smgm_ctrl1	85BDh
lane2_rx_ree_smgm_ctrl1	89BDh
lane3_rx_ree_smgm_ctrl1	8DBDh

13.4.10.2.339.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_ree_smgm_ctrl1_15	rx_ree_smgm_ctrl1_14	rx_ree_smgm_ctrl1_13	rx_ree_smgm_ctrl1_12	rx_ree_smgm_ctrl1_11	rx_ree_smgm_ctrl1_10	rx_ree_smgm_ctrl1_9	rx_ree_smgm_ctrl1_8	rx_ree_smgm_ctrl1_7	rx_ree_smgm_ctrl1_6	rx_ree_smgm_ctrl1_5	rx_ree_smgm_ctrl1_4	rx_ree_smgm_ctrl1_3	rx_ree_smgm_ctrl1_2	rx_ree_smgm_ctrl1_1	rx_ree_smgm_ctrl1_0
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	1	0

13.4.10.2.339.3 Fields

Field	Function
15 rx_ree_smgm_ctrl1_15	REE Periodic general control state machine E path enable standard mode 3: This bit controls when the REE periodic general control state machine will enable the analog E path when running REE functions, when
14 rx_ree_smgm_ctrl1_14	REE Periodic general control state machine E path enable standard mode 2: This bit controls when the REE periodic general control state machine will enable the analog E path when running REE functions, when
13 rx_ree_smgm_ctrl1_13	REE Periodic general control state machine E path enable standard mode 1: This bit controls when the REE periodic general control state machine will enable the analog E path when running REE functions, when
12 rx_ree_smgm_ctrl1_12	REE Periodic general control state machine E path enable standard mode 0: This bit controls when the REE periodic general control state machine will enable the analog E path when running REE functions, when
11 rx_ree_smgm_ctrl1_11	REE Periodic general control state machine enable standard mode 3: This bit will control if the REE periodic general control state machine will run, when
10 rx_ree_smgm_ctrl1_10	REE Periodic general control state machine enable standard mode 2: This bit will control if the REE periodic general control state machine will run, when
9 rx_ree_smgm_ctrl1_9	REE Periodic general control state machine enable standard mode 1: This bit will control if the REE periodic general control state machine will run, when
8 rx_ree_smgm_ctrl1_8	REE Periodic general control state machine enable standard mode 0: This bit will control if the REE periodic general control state machine will run, when
7 rx_ree_smgm_ctrl1_7	REE Gen 2 general control state machine E path enable standard mode 3: This bit controls when the REE Gen 2 general control state machine will enable the analog E path when running REE functions, when

Table continues on the next page...

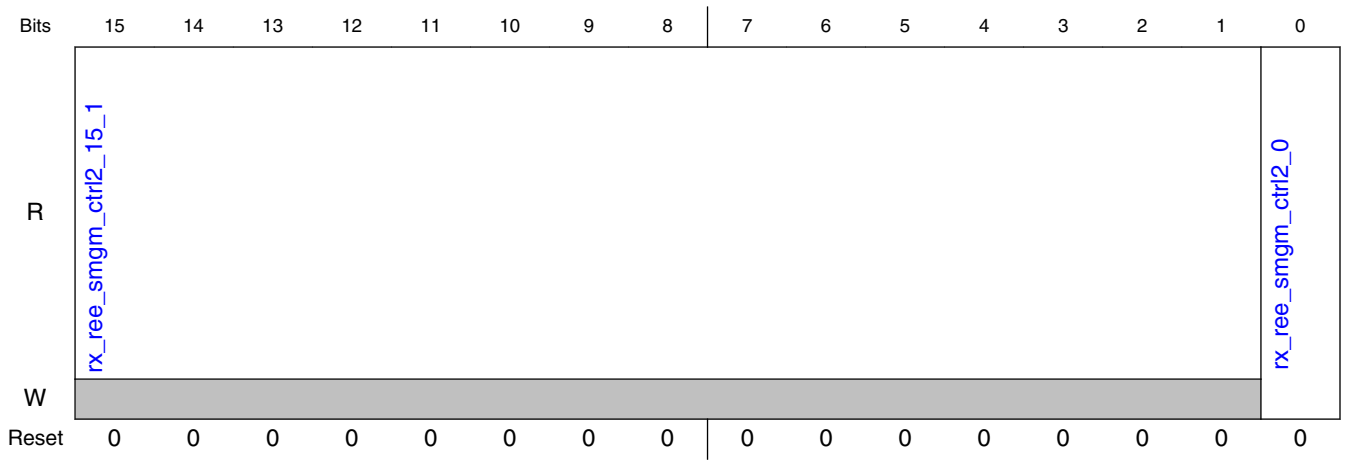
Field	Function
6 rx_ree_smgm_ctrl1_6	REE Gen 2 general control state machine E path enable standard mode 2: This bit controls when the REE Gen 2 general control state machine will enable the analog E path when running REE functions, when
5 rx_ree_smgm_ctrl1_5	REE Gen 2 general control state machine E path enable standard mode 1: This bit controls when the REE Gen 2 general control state machine will enable the analog E path when running REE functions, when
4 rx_ree_smgm_ctrl1_4	REE Gen 2 general control state machine E path enable standard mode 0: This bit controls when the REE Gen 2 general control state machine will enable the analog E path when running REE functions, when
3 rx_ree_smgm_ctrl1_3	REE Gen 2 general control state machine enable standard mode 3: This bit will control if the REE Gen 2 general control state machine will run, when
2 rx_ree_smgm_ctrl1_2	REE Gen 2 general control state machine enable standard mode 2: This bit will control if the REE Gen 2 general control state machine will run, when
1 rx_ree_smgm_ctrl1_1	REE Gen 2 general control state machine enable standard mode 1: This bit will control if the REE Gen 2 general control state machine will run, when
0 rx_ree_smgm_ctrl1_0	REE Gen 2 general control state machine enable standard mode 0: This bit will control if the REE Gen 2 general control state machine will run, when

13.4.10.2.340 REE control state machine gen mode control register 2 (lane 0_rx_ree_smgm_ctrl2 - lane3_rx_ree_smgm_ctrl2)

13.4.10.2.340.1 Offset

Register	Offset
lane0_rx_ree_smgm_ctrl2	81BEh
lane1_rx_ree_smgm_ctrl2	85BEh
lane2_rx_ree_smgm_ctrl2	89BEh
lane3_rx_ree_smgm_ctrl2	8DBEh

13.4.10.2.340.2 Diagram



13.4.10.2.340.3 Fields

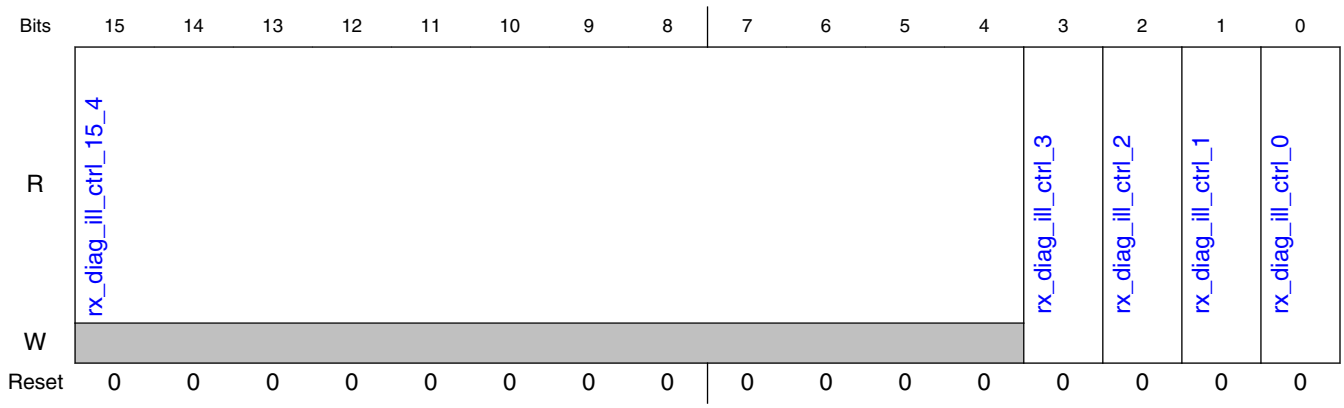
Field	Function
15-1 rx_ree_smgm_ctrl2_15_1	Reserved
0 rx_ree_smgm_ctrl2_0	REE USB 3 general control state machine E path enable: This bit controls when the REE USB 3 general control state machine will enable the analog E path when running REE functions.

13.4.10.2.341 RX ILL diagnostic control register (lane0_rx_diag_ill_ctrl - lane3_rx_diag_ill_ctrl)

13.4.10.2.341.1 Offset

Register	Offset
lane0_rx_diag_ill_ctrl	81C0h
lane1_rx_diag_ill_ctrl	85C0h
lane2_rx_diag_ill_ctrl	89C0h
lane3_rx_diag_ill_ctrl	8DC0h

13.4.10.2.341.2 Diagram



13.4.10.2.341.3 Fields

Field	Function
15-4 rx_diag_ill_ctrl_15_4	Reserved
3 rx_diag_ill_ctrl_3	IQ PI ILL calibration enable override enable: When active (1b1), the IQ PI ILL calibration enable override bit in this register, can be used to directly control the enable of the IQ PI ILL calibration function in the IQ PI ILL (instead of the IQ PI ILL calibration module).
2 rx_diag_ill_ctrl_2	IQ PI ILL calibration enable override: When enabled by the IQ PI ILL calibration enable override enable bit in this register, this bit will directly control the enable of the IQ PI ILL calibration function in the IQ PI ILL.
1 rx_diag_ill_ctrl_1	E PI ILL calibration enable override enable: When active (1b1), the E PI ILL calibration enable override bit in this register, can be used to directly control the enable of the E PI ILL calibration function in the E PI ILL (instead of the E PI ILL calibration module).
0 rx_diag_ill_ctrl_0	E PI ILL calibration enable override: When enabled by the E PI ILL calibration enable override enable bit in this register, this bit will directly control the enable of the E PI ILL calibration function in the E PI ILL.

13.4.10.2.342 RX ILL IQ trim 0 register (lane0_rx_diag_ill_iq_trim0 - lane3_rx_diag_ill_iq_trim0)

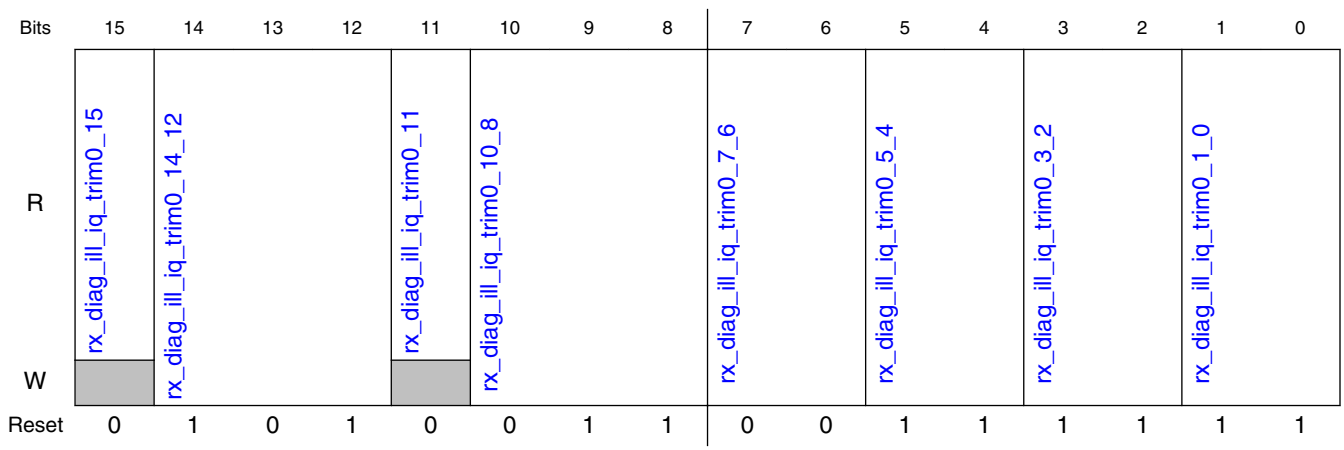
13.4.10.2.342.1 Offset

Register	Offset
lane0_rx_diag_ill_iq_trim0	81C1h

Table continues on the next page...

Register	Offset
lane1_rx_diag_ill_iq_trim0	85C1h
lane2_rx_diag_ill_iq_trim0	89C1h
lane3_rx_diag_ill_iq_trim0	8DC1h

13.4.10.2.342.2 Diagram



13.4.10.2.342.3 Fields

Field	Function
15 rx_diag_ill_iq_trim0_15	Reserved
14-12 rx_diag_ill_iq_trim0_14_12	rx_diag_ill_iq_trim0_14_12
11 rx_diag_ill_iq_trim0_11	Reserved
10-8 rx_diag_ill_iq_trim0_10_8	rx_diag_ill_iq_trim0_10_8
7-6 rx_diag_ill_iq_trim0_7_6	rx_diag_ill_iq_trim0_7_6
5-4	rx_diag_ill_iq_trim0_5_4

Table continues on the next page...

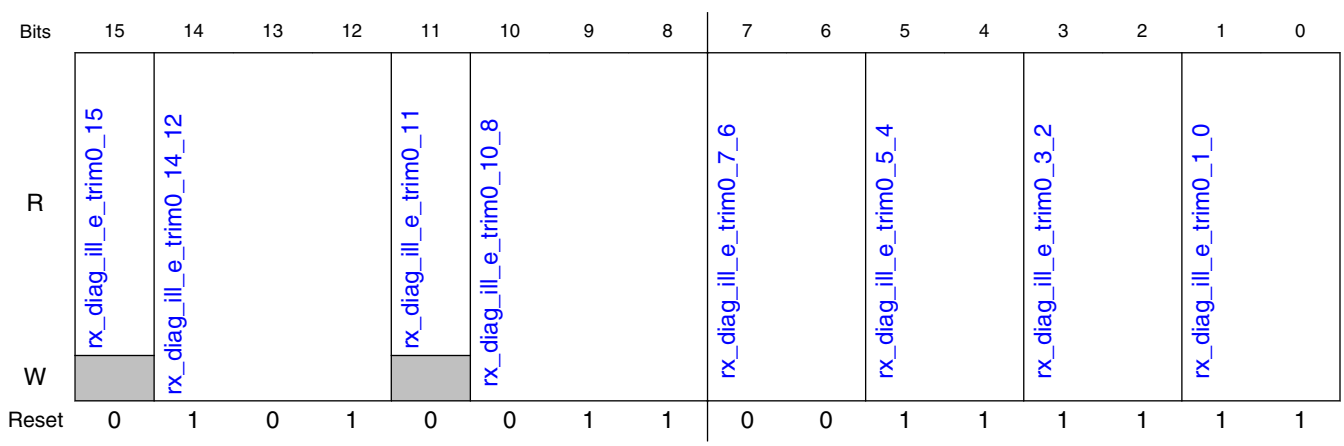
Field	Function
rx_diag_ill_iq_trim0_5_4	
3-2 rx_diag_ill_iq_trim0_3_2	rx_diag_ill_iq_trim0_3_2
1-0 rx_diag_ill_iq_trim0_1_0	rx_diag_ill_iq_trim0_1_0

13.4.10.2.343 RX ILL E trim 0 register (lane0_rx_diag_ill_e_trim0 - lane3_rx_diag_ill_e_trim0)

13.4.10.2.343.1 Offset

Register	Offset
lane0_rx_diag_ill_e_trim0	81C2h
lane1_rx_diag_ill_e_trim0	85C2h
lane2_rx_diag_ill_e_trim0	89C2h
lane3_rx_diag_ill_e_trim0	8DC2h

13.4.10.2.343.2 Diagram



13.4.10.2.343.3 Fields

Field	Function
15	Reserved

Table continues on the next page...

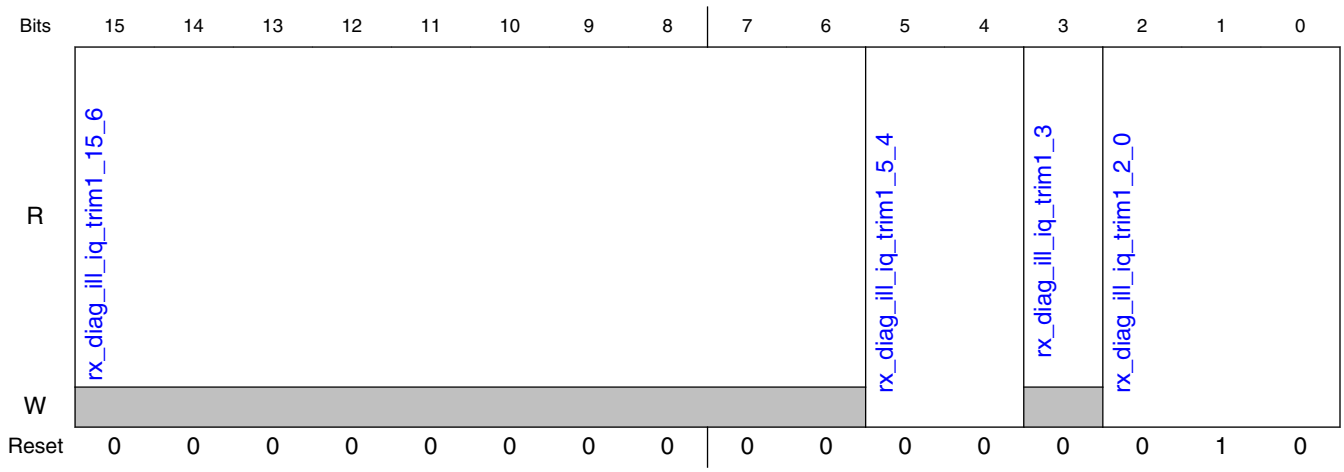
Field	Function
rx_diag_ill_e_trim0_15	
14-12 rx_diag_ill_e_trim0_14_12	rx_diag_ill_e_trim0_14_12
11 rx_diag_ill_e_trim0_11	Reserved
10-8 rx_diag_ill_e_trim0_10_8	rx_diag_ill_e_trim0_10_8
7-6 rx_diag_ill_e_trim0_7_6	rx_diag_ill_e_trim0_7_6
5-4 rx_diag_ill_e_trim0_5_4	rx_diag_ill_e_trim0_5_4
3-2 rx_diag_ill_e_trim0_3_2	rx_diag_ill_e_trim0_3_2
1-0 rx_diag_ill_e_trim0_1_0	rx_diag_ill_e_trim0_1_0

13.4.10.2.344 RX ILL IQ trim 1 register (lane0_rx_diag_ill_iq_trim1 - lane3_rx_diag_ill_iq_trim1)

13.4.10.2.344.1 Offset

Register	Offset
lane0_rx_diag_ill_iq_trim1	81C3h
lane1_rx_diag_ill_iq_trim1	85C3h
lane2_rx_diag_ill_iq_trim1	89C3h
lane3_rx_diag_ill_iq_trim1	8DC3h

13.4.10.2.344.2 Diagram



13.4.10.2.344.3 Fields

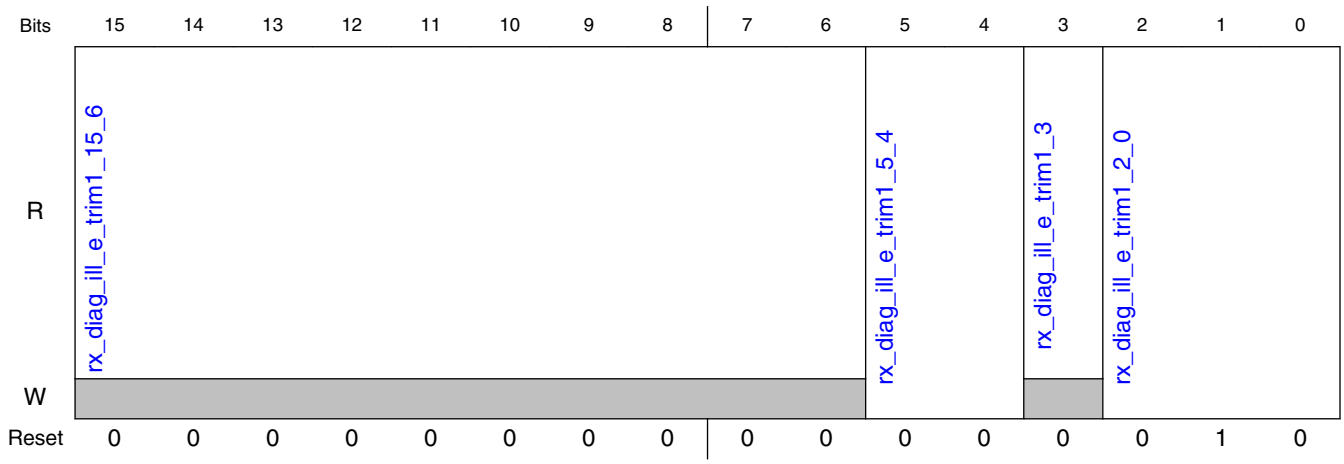
Field	Function
15-6 rx_diag_ill_iq_trim1_15_6	Reserved
5-4 rx_diag_ill_iq_trim1_5_4	Drives the
3 rx_diag_ill_iq_trim1_3	Reserved
2-0 rx_diag_ill_iq_trim1_2_0	Drives the

13.4.10.2.345 RX ILL E trim 1 register (lane0_rx_diag_ill_e_trim1 - lane3_rx_diag_ill_e_trim1)

13.4.10.2.345.1 Offset

Register	Offset
lane0_rx_diag_ill_e_trim1	81C4h
lane1_rx_diag_ill_e_trim1	85C4h
lane2_rx_diag_ill_e_trim1	89C4h
lane3_rx_diag_ill_e_trim1	8DC4h

13.4.10.2.345.2 Diagram



13.4.10.2.345.3 Fields

Field	Function
15-6 rx_diag_ill_e_trim1_15_6	Reserved
5-4 rx_diag_ill_e_trim1_5_4	Drives the
3 rx_diag_ill_e_trim1_3	Reserved
2-0 rx_diag_ill_e_trim1_2_0	Drives the

13.4.10.2.346 RX ILL IQ E trim 2 register (lane0_rx_diag_ill_iqe_trim2 - lane3_rx_diag_ill_iqe_trim2)

13.4.10.2.346.1 Offset

Register	Offset
lane0_rx_diag_ill_iqe_trim2	81C5h

Table continues on the next page...

Register	Offset
lane1_rx_diag_ill_iqe_trim2	85C5h
lane2_rx_diag_ill_iqe_trim2	89C5h
lane3_rx_diag_ill_iqe_trim2	8DC5h

13.4.10.2.346.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R																
W	rx_diag_ill_iqe_trim2_15_14		rx_diag_ill_iqe_trim2_13_12		rx_diag_ill_iqe_trim2_11_10		rx_diag_ill_iqe_trim2_9_8		rx_diag_ill_iqe_trim2_7_6		rx_diag_ill_iqe_trim2_5_4		rx_diag_ill_iqe_trim2_3_2		rx_diag_ill_iqe_trim2_1_0	
Reset	0	0	1	1	1	1	1	0	0	0	0	0	0	0	0	0

13.4.10.2.346.3 Fields

Field	Function
15-14 rx_diag_ill_iqe_trim2_15_14	rx_diag_ill_iqe_trim2_15_14
13-12 rx_diag_ill_iqe_trim2_13_12	rx_diag_ill_iqe_trim2_13_12
11-10 rx_diag_ill_iqe_trim2_11_10	rx_diag_ill_iqe_trim2_11_10
9-8 rx_diag_ill_iqe_trim2_9_8	rx_diag_ill_iqe_trim2_9_8
7-6 rx_diag_ill_iqe_trim2_7_6	rx_diag_ill_iqe_trim2_7_6
5-4	rx_diag_ill_iqe_trim2_5_4

Table continues on the next page...

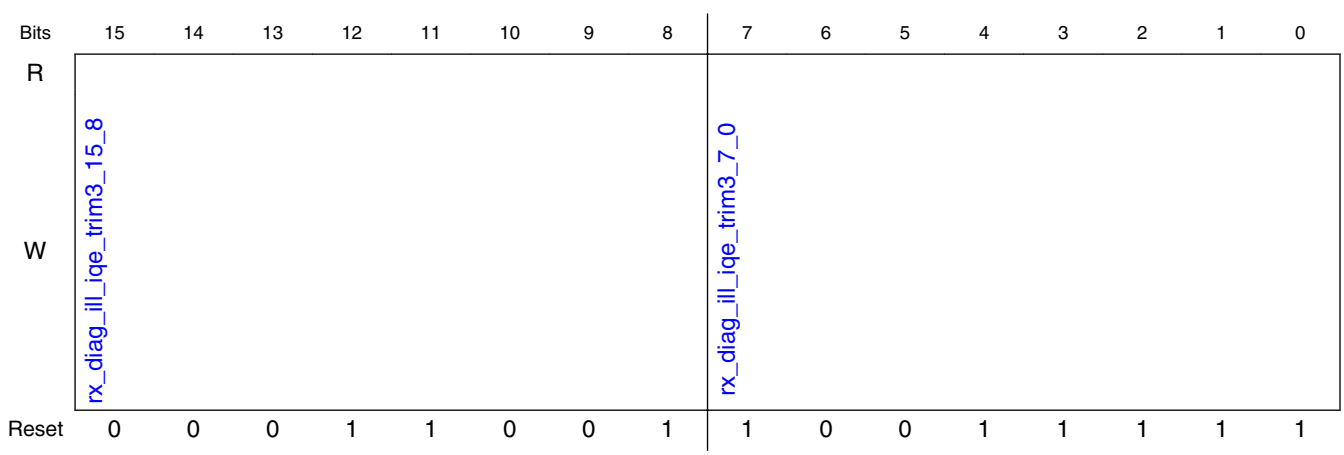
Field	Function
rx_diag_ill_iqe_trim2_5_4	
3-2 rx_diag_ill_iqe_trim2_3_2	rx_diag_ill_iqe_trim2_3_2
1-0 rx_diag_ill_iqe_trim2_1_0	rx_diag_ill_iqe_trim2_1_0

13.4.10.2.347 RX ILL IQ E trim 3 register (lane0_rx_diag_ill_iqe_trim3 - lane3_rx_diag_ill_iqe_trim3)

13.4.10.2.347.1 Offset

Register	Offset
lane0_rx_diag_ill_iqe_trim3	81C6h
lane1_rx_diag_ill_iqe_trim3	85C6h
lane2_rx_diag_ill_iqe_trim3	89C6h
lane3_rx_diag_ill_iqe_trim3	8DC6h

13.4.10.2.347.2 Diagram



13.4.10.2.347.3 Fields

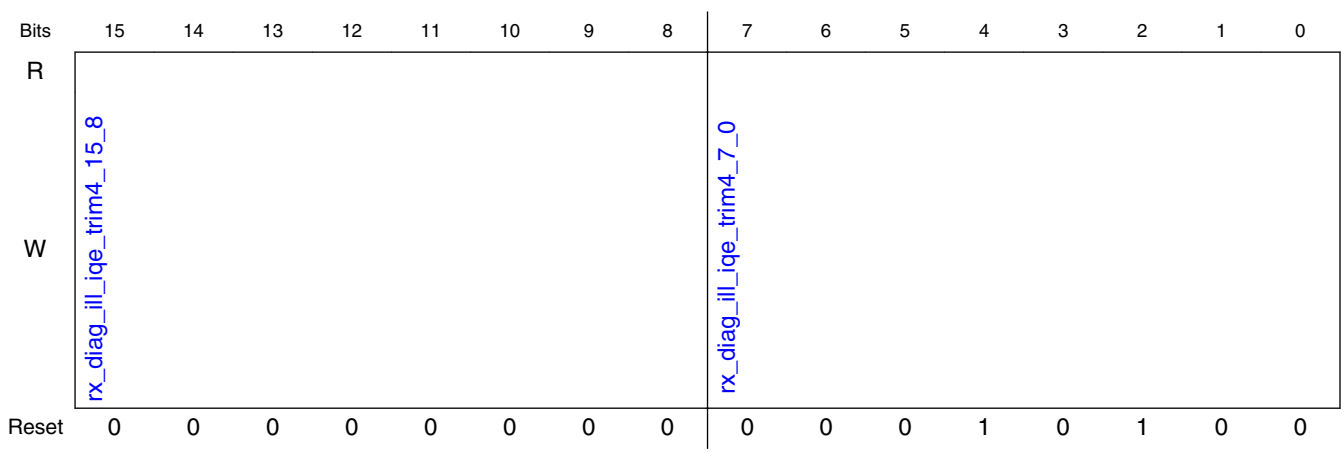
Field	Function
15-8 rx_diag_ill_iqe_trim3_15_8	rx_diag_ill_iqe_trim3_15_8
7-0 rx_diag_ill_iqe_trim3_7_0	rx_diag_ill_iqe_trim3_7_0

13.4.10.2.348 RX ILL IQ E trim 4 register (lane0_rx_diag_ill_iqe_trim4 - lane3_rx_diag_ill_iqe_trim4)

13.4.10.2.348.1 Offset

Register	Offset
lane0_rx_diag_ill_iqe_trim4	81C7h
lane1_rx_diag_ill_iqe_trim4	85C7h
lane2_rx_diag_ill_iqe_trim4	89C7h
lane3_rx_diag_ill_iqe_trim4	8DC7h

13.4.10.2.348.2 Diagram



13.4.10.2.348.3 Fields

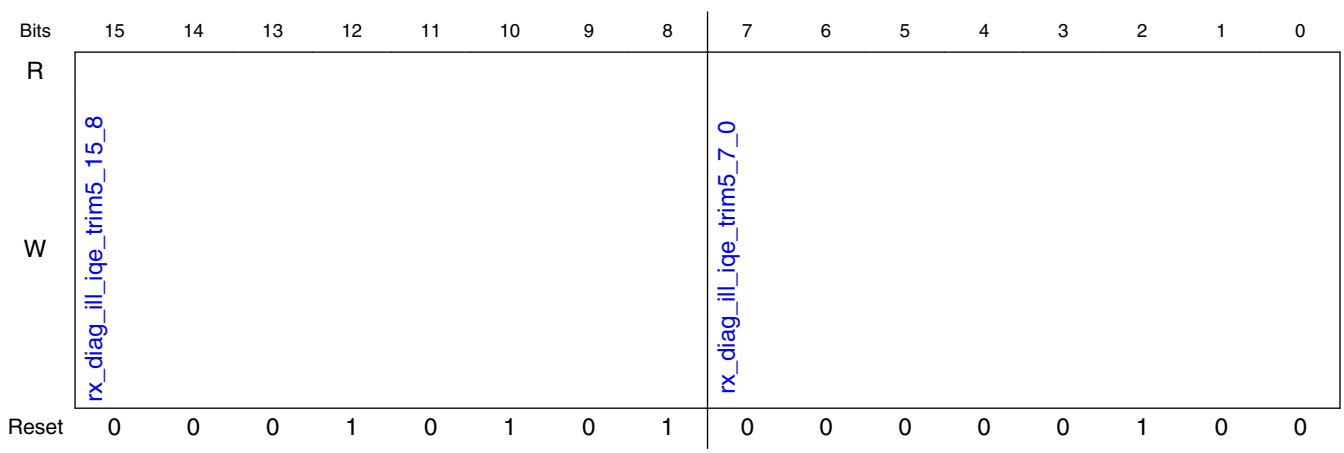
Field	Function
15-8 rx_diag_ill_iqe_trim4_15_8	rx_diag_ill_iqe_trim4_15_8
7-0 rx_diag_ill_iqe_trim4_7_0	rx_diag_ill_iqe_trim4_7_0

13.4.10.2.349 RX ILL IQ E trim 5 register (lane0_rx_diag_ill_iqe_trim5 - lane3_rx_diag_ill_iqe_trim5)

13.4.10.2.349.1 Offset

Register	Offset
lane0_rx_diag_ill_iqe_trim5	81C8h
lane1_rx_diag_ill_iqe_trim5	85C8h
lane2_rx_diag_ill_iqe_trim5	89C8h
lane3_rx_diag_ill_iqe_trim5	8DC8h

13.4.10.2.349.2 Diagram



13.4.10.2.349.3 Fields

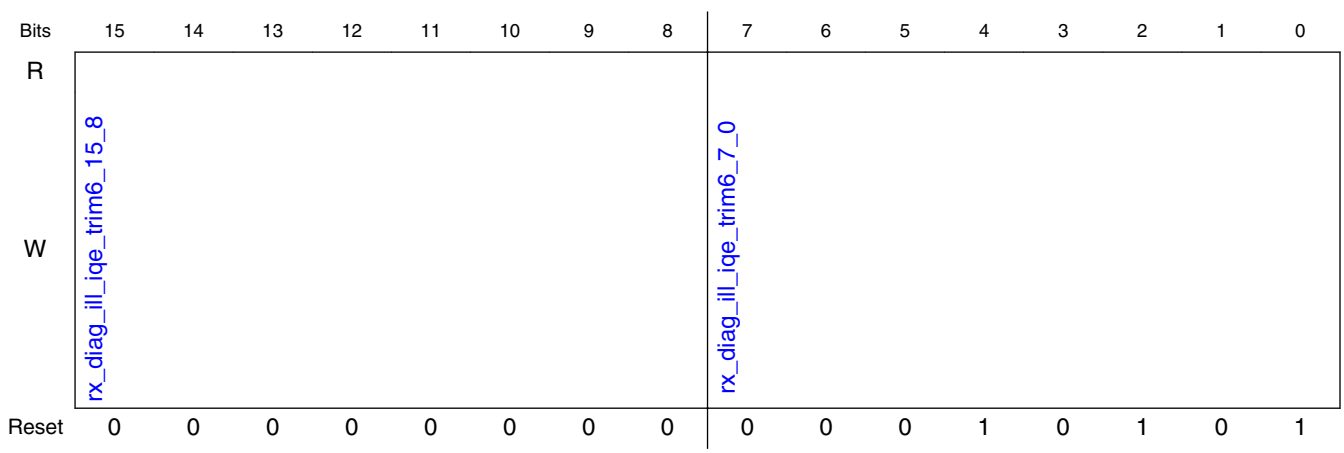
Field	Function
15-8 rx_diag_ill_iqe_trim5_15_8	rx_diag_ill_iqe_trim5_15_8
7-0 rx_diag_ill_iqe_trim5_7_0	rx_diag_ill_iqe_trim5_7_0

13.4.10.2.350 RX ILL IQ E trim 6 register (lane0_rx_diag_ill_iqe_trim6 - lane3_rx_diag_ill_iqe_trim6)

13.4.10.2.350.1 Offset

Register	Offset
lane0_rx_diag_ill_iqe_trim6	81C9h
lane1_rx_diag_ill_iqe_trim6	85C9h
lane2_rx_diag_ill_iqe_trim6	89C9h
lane3_rx_diag_ill_iqe_trim6	8DC9h

13.4.10.2.350.2 Diagram



13.4.10.2.350.3 Fields

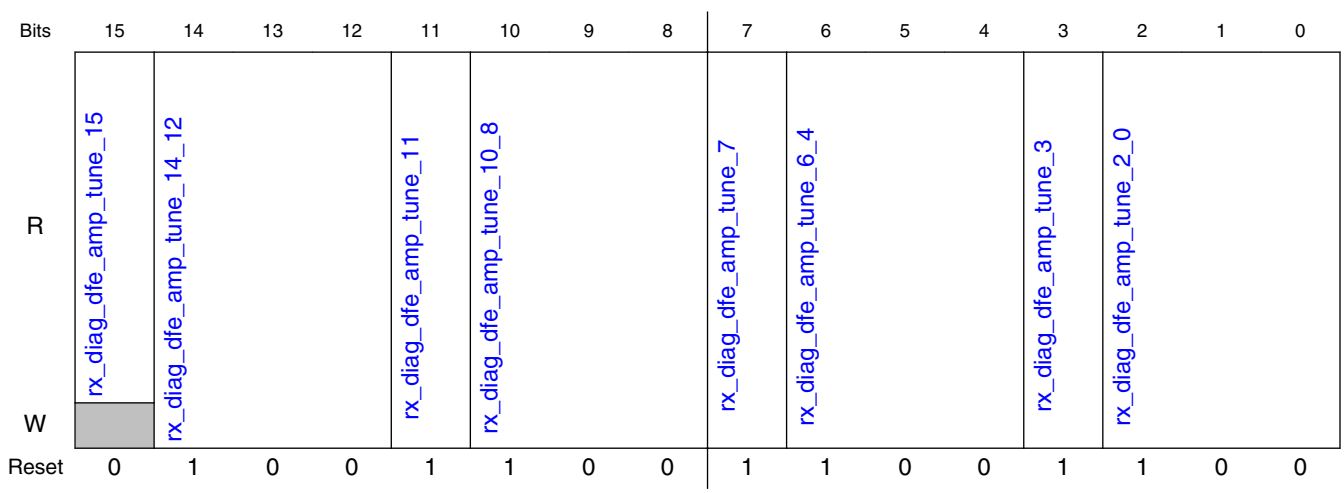
Field	Function
15-8 rx_diag_ill_iqe_trim6_15_8	rx_diag_ill_iqe_trim6_15_8
7-0 rx_diag_ill_iqe_trim6_7_0	rx_diag_ill_iqe_trim6_7_0

13.4.10.2.351 DFE amp fine tuning register (lane0_rx_diag_dfe_amp_tune - lane3_rx_diag_dfe_amp_tune)

13.4.10.2.351.1 Offset

Register	Offset
lane0_rx_diag_dfe_amp_tune	81D0h
lane1_rx_diag_dfe_amp_tune	85D0h
lane2_rx_diag_dfe_amp_tune	89D0h
lane3_rx_diag_dfe_amp_tune	8DD0h

13.4.10.2.351.2 Diagram



13.4.10.2.351.3 Fields

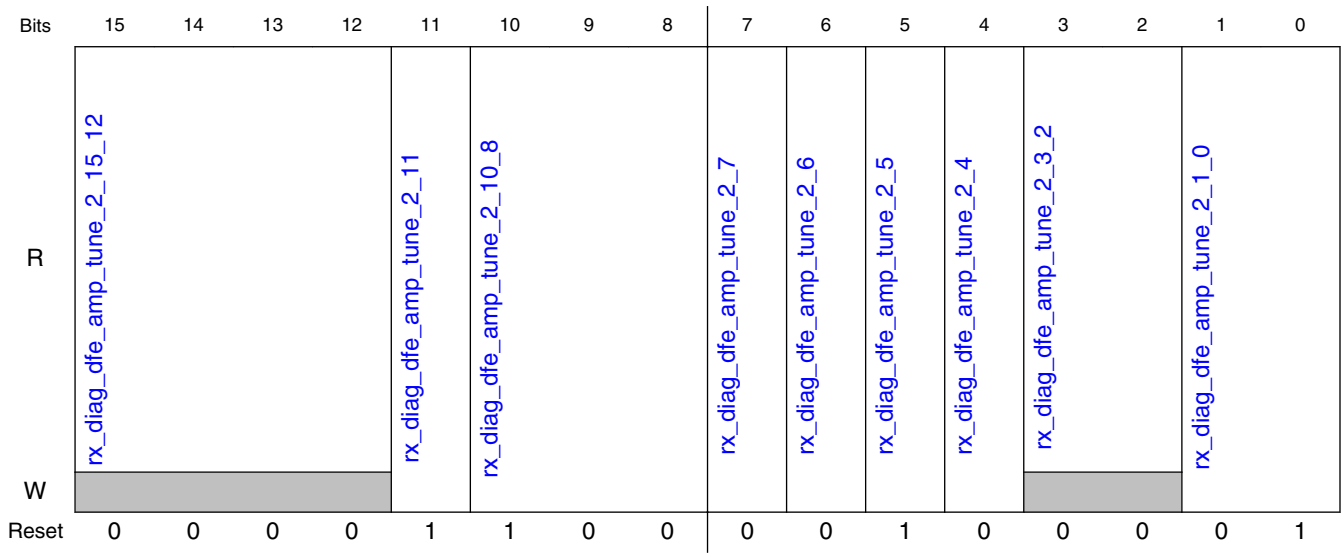
Field	Function
15 rx_diag_dfe_amp_tune_15	Reserved
14-12 rx_diag_dfe_amp_tune_14_12	DFE constant gm bias tune: Adjusts the constant gm bias.
11 rx_diag_dfe_amp_tune_11	DFE VGA constant gm bias enable: Enables the VGA constant gm bias. This bit drives the
10-8 rx_diag_dfe_amp_tune_10_8	DFE VGA amp current adjust: Adjusts the current for the DFE VGA amp, using the
7 rx_diag_dfe_amp_tune_7	DFE peaking constant gm bias enable: Enables the peaking constant gm bias. This bit drives the
6-4 rx_diag_dfe_amp_tune_6_4	DFE peaking amp current adjust: Adjusts the current for the DFE peaking amp, using the
3 rx_diag_dfe_amp_tune_3	DFE summing constant gm bias enable: Enables the summing constant gm bias. This bit drives the
2-0 rx_diag_dfe_amp_tune_2_0	DFE summing amp current adjust: Adjusts the current for the DFE summing amp, using the

13.4.10.2.352 DFE amp fine tuning 2 register (lane0_rx_diag_dfe_amp_tune_2 - lane3_rx_diag_dfe_amp_tune_2)

13.4.10.2.352.1 Offset

Register	Offset
lane0_rx_diag_dfe_amp_tune_2	81D1h
lane1_rx_diag_dfe_amp_tune_2	85D1h
lane2_rx_diag_dfe_amp_tune_2	89D1h
lane3_rx_diag_dfe_amp_tune_2	8DD1h

13.4.10.2.352.2 Diagram



13.4.10.2.352.3 Fields

Field	Function
15-12 rx_diag_dfe_amp_tune_2_15_12	Reserved
11 rx_diag_dfe_amp_tune_2_11	DFE low frequency equalizer constant gm bias enable: Enables the low frequency equalizer constant gm bias. This bit drives the
10-8 rx_diag_dfe_amp_tune_2_10_8	DFE low frequency equalizer current adjust: Adjusts the current for the DFE low frequency equalizer, using the
7 rx_diag_dfe_amp_tune_2_7	DFE peaking amp boost: Enables the active inductors boost function in the peaking amp, for high data rates. This bit controls the
6 rx_diag_dfe_amp_tune_2_6	DFE VGA stage 1 boost: Enables the active inductors boost function in stage 1 of the VGA, for high data rates. This bit controls the
5 rx_diag_dfe_amp_tune_2_5	DFE VGA stage 2 boost: Enables the active inductors boost function in stage 2 of the VGA, for high data rates. This bit controls the
4 rx_diag_dfe_amp_tune_2_4	DFE RX Tap 1 DAC Range Select: Controls the tap 1 DAC range in the analog DFE. This bit controls the

Table continues on the next page...

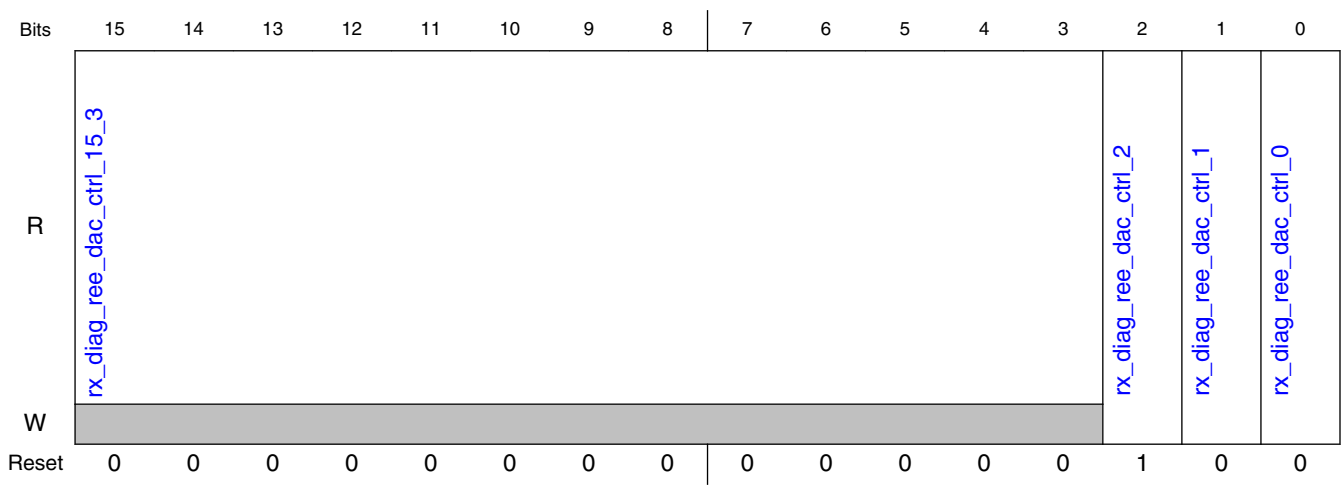
Field	Function
3-2 rx_diag_dfe_amp_tune_2_3_2	Reserved
1-0 rx_diag_dfe_amp_tune_2_1_0	DFE RX amp current adjust: Adjusts the mix of constant-gm and External Current for RX front end amplifiers. This controls the

13.4.10.2.353 REE DAC control register (lane0_rx_diag_ree_dac_ctrl - lane 3_rx_diag_ree_dac_ctrl)

13.4.10.2.353.1 Offset

Register	Offset
lane0_rx_diag_ree_dac_ctrl	81D2h
lane1_rx_diag_ree_dac_ctrl	85D2h
lane2_rx_diag_ree_dac_ctrl	89D2h
lane3_rx_diag_ree_dac_ctrl	8DD2h

13.4.10.2.353.2 Diagram



13.4.10.2.353.3 Fields

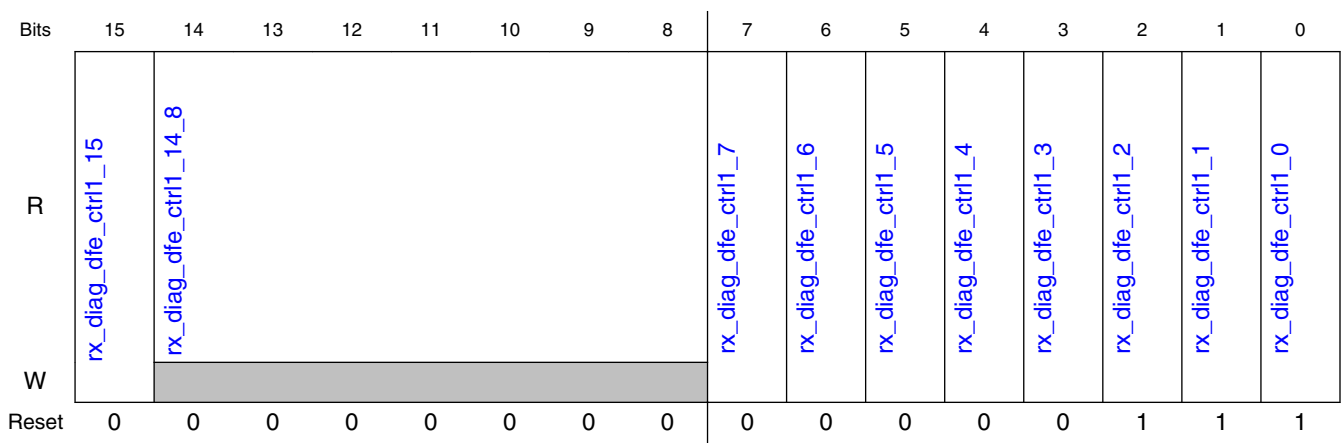
Field	Function
15-3 rx_diag_ree_dac_ctrl_15_3	Reserved
2 rx_diag_ree_dac_ctrl_2	DFE Offset DAC enable: Enables the DFE offset DAC associated with the VGA amp.
1 rx_diag_ree_dac_ctrl_1	DFE Offset DAC attenuation: Adds attenuation to the DFE offset DAC associated with the VGA amp.
0 rx_diag_ree_dac_ctrl_0	DFE DAC attenuation: Adds attenuation to the DFE DACs associated with the summing amp.

13.4.10.2.354 Receiver DFE control register 1 (lane0_rx_diag_dfe_ctrl1 - lane3_rx_diag_dfe_ctrl1)

13.4.10.2.354.1 Offset

Register	Offset
lane0_rx_diag_dfe_ctrl1	81D3h
lane1_rx_diag_dfe_ctrl1	85D3h
lane2_rx_diag_dfe_ctrl1	89D3h
lane3_rx_diag_dfe_ctrl1	8DD3h

13.4.10.2.354.2 Diagram



13.4.10.2.354.3 Fields

Field	Function
15 rx_diag_dfe_ctrl1_15	DFE Tap 1 deserializer MUX select: When the DFE is enabled, there are two samplers that ultimately provide the input data to the deserializer. In cases where the DFE is not enabled, only one of the samplers can be used to provide data to the deserializer. This bit specifies which of these paths provides this data, by driving the
14-8 rx_diag_dfe_ctrl1_14_8	Reserved
7 rx_diag_dfe_ctrl1_7	Receiver DFE low frequency equalization enable value standard mode 3: This bit controls the
6 rx_diag_dfe_ctrl1_6	Receiver DFE low frequency equalization enable value standard mode 2: This bit controls the
5 rx_diag_dfe_ctrl1_5	Receiver DFE low frequency equalization enable value standard mode 1: This bit controls the
4 rx_diag_dfe_ctrl1_4	Receiver DFE low frequency equalization enable value standard mode 0: This bit controls the
3 rx_diag_dfe_ctrl1_3	Receiver DFE equalization enable mask value standard mode 3: This bit will controls the rx_dfe_eq_enable_std_mask digital signal, which can be used to mask off the
2 rx_diag_dfe_ctrl1_2	Receiver DFE equalization enable mask value standard mode 2: This bit will controls the rx_dfe_eq_enable_std_mask digital signal, which can be used to mask off the
1 rx_diag_dfe_ctrl1_1	Receiver DFE equalization enable mask value standard mode 1: This bit will controls the rx_dfe_eq_enable_std_mask digital signal, which can be used to mask off the
0 rx_diag_dfe_ctrl1_0	Receiver DFE equalization enable mask value standard mode 0: This bit will controls the rx_dfe_eq_enable_std_mask digital signal, which can be used to mask off the

13.4.10.2.355 Receiver DFE control register 2 (lane0_rx_diag_dfe_ctrl2 - lane3_rx_diag_dfe_ctrl2)

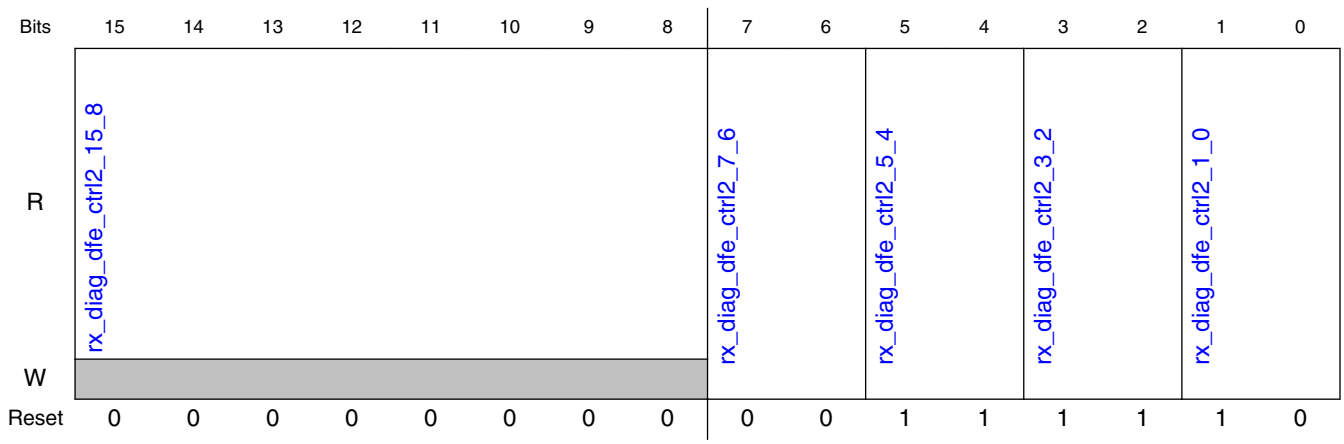
13.4.10.2.355.1 Offset

Register	Offset
lane0_rx_diag_dfe_ctrl2	81D4h

Table continues on the next page...

Register	Offset
lane1_rx_diag_dfe_ctrl2	85D4h
lane2_rx_diag_dfe_ctrl2	89D4h
lane3_rx_diag_dfe_ctrl2	8DD4h

13.4.10.2.355.2 Diagram



13.4.10.2.355.3 Fields

Field	Function
15-8 rx_diag_dfe_ctrl2_15_8	Reserved
7-6 rx_diag_dfe_ctrl2_7_6	RX equalizer range select standard mode 3: Thermometer encoded field that sets the equalizer to the required operating frequency range, by driving the
5-4 rx_diag_dfe_ctrl2_5_4	RX equalizer range select standard mode 2: Thermometer encoded field that sets the equalizer to the required operating frequency range, by driving the
3-2 rx_diag_dfe_ctrl2_3_2	RX equalizer range select standard mode 1: Thermometer encoded field that sets the equalizer to the required operating frequency range, by driving the
1-0 rx_diag_dfe_ctrl2_1_0	RX equalizer range select standard mode 0: Thermometer encoded field that sets the equalizer to the required operating frequency range, by driving the

13.4.10.2.356 Receiver DFE control register 3 (lane0_rx_diag_dfe_ctrl3 - lane3_rx_diag_dfe_ctrl3)

13.4.10.2.356.1 Offset

Register	Offset
lane0_rx_diag_dfe_ctrl3	81D5h
lane1_rx_diag_dfe_ctrl3	85D5h
lane2_rx_diag_dfe_ctrl3	89D5h
lane3_rx_diag_dfe_ctrl3	8DD5h

13.4.10.2.356.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_diag_dfe_ctrl3_15_12				rx_diag_dfe_ctrl3_11_8				rx_diag_dfe_ctrl3_7_4				rx_diag_dfe_ctrl3_3_0			
W																
Reset	0	0	0	0	1	0	1	0	1	1	0	0	1	0	1	0

13.4.10.2.356.3 Fields

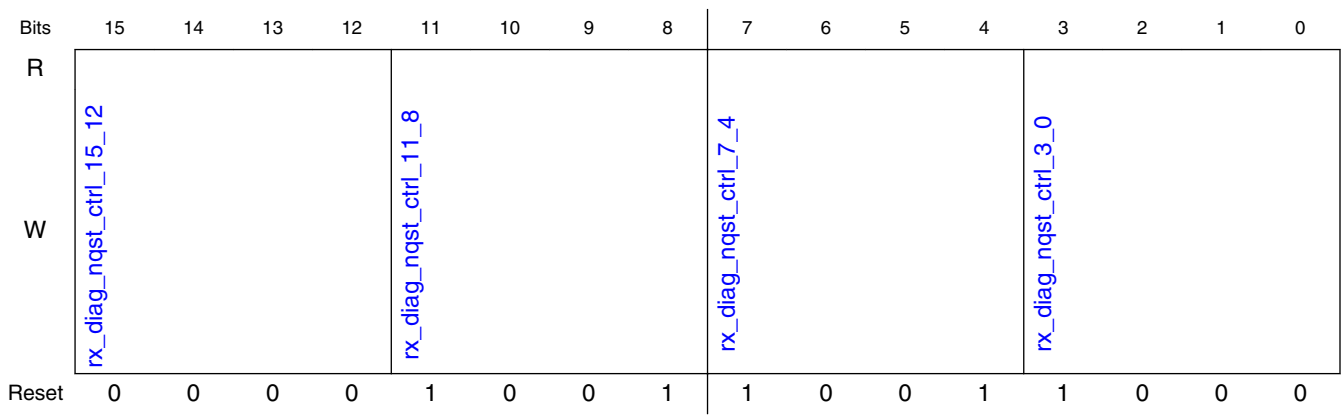
Field	Function
15-12 rx_diag_dfe_ctrl3_15_12	RX DFE peaking resistor code select standard mode 3: Thermometer encoded field used to program the DC gain/relative peaking of the equalizer, by driving the
11-8 rx_diag_dfe_ctrl3_11_8	RX DFE peaking resistor code select standard mode 2: Thermometer encoded field used to program the DC gain/relative peaking of the equalizer, by driving the
7-4 rx_diag_dfe_ctrl3_7_4	RX DFE peaking resistor code select standard mode 1: Thermometer encoded field used to program the DC gain/relative peaking of the equalizer, by driving the
3-0 rx_diag_dfe_ctrl3_3_0	RX DFE peaking resistor code select standard mode 0: Thermometer encoded field used to program the DC gain/relative peaking of the equalizer, by driving the

13.4.10.2.357 Nyquist control register (lane0_rx_diag_nqst_ctrl - lane3_rx_diag_nqst_ctrl)

13.4.10.2.357.1 Offset

Register	Offset
lane0_rx_diag_nqst_ctrl	81D6h
lane1_rx_diag_nqst_ctrl	85D6h
lane2_rx_diag_nqst_ctrl	89D6h
lane3_rx_diag_nqst_ctrl	8DD6h

13.4.10.2.357.2 Diagram



13.4.10.2.357.3 Fields

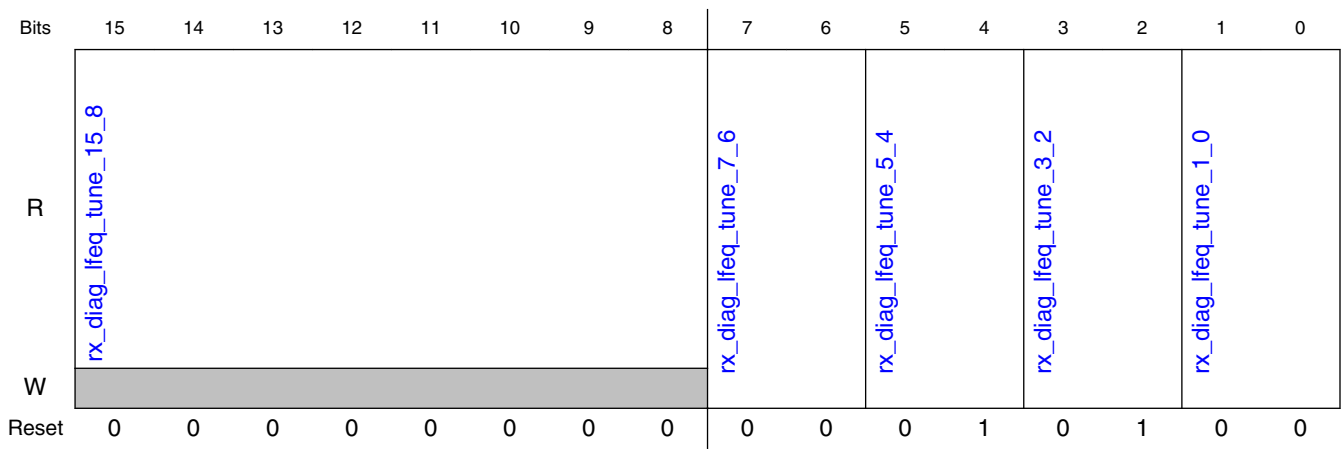
Field	Function
15-12 rx_diag_nqst_ctrl_15_12	RX nyquist select value standard mode 3: This field specifies the receiver DFE nyquist select value on the
11-8 rx_diag_nqst_ctrl_11_8	RX nyquist select value standard mode 2: This field specifies the receiver DFE nyquist select value on the
7-4 rx_diag_nqst_ctrl_7_4	RX nyquist select value standard mode 1: This field specifies the receiver DFE nyquist select value on the
3-0 rx_diag_nqst_ctrl_3_0	RX nyquist select value standard mode 0: This field specifies the receiver DFE nyquist select value on the

13.4.10.2.358 Low frequency equalizer tuning register (lane0_rx_diag_lfeq_tune - lane3_rx_diag_lfeq_tune)

13.4.10.2.358.1 Offset

Register	Offset
lane0_rx_diag_lfeq_tune	81D7h
lane1_rx_diag_lfeq_tune	85D7h
lane2_rx_diag_lfeq_tune	89D7h
lane3_rx_diag_lfeq_tune	8DD7h

13.4.10.2.358.2 Diagram



13.4.10.2.358.3 Fields

Field	Function
15-8 rx_diag_lfeq_tune_15_8	Reserved
7-6 rx_diag_lfeq_tune_7_6	RX low frequency equalization zero frequency value standard mode 3: This field specifies the receiver zero frequency setting for the low frequency equalization function, by driving the
5-4 rx_diag_lfeq_tune_5_4	RX low frequency equalization zero frequency value standard mode 2: This field specifies the receiver zero frequency setting for the low frequency equalization function, by driving the
3-2	RX low frequency equalization zero frequency value standard mode 1: This field specifies the receiver zero frequency setting for the low frequency equalization function, by driving the

Table continues on the next page...

Field	Function
rx_diag_lfeq_tune_3_2	
1-0 rx_diag_lfeq_tune_1_0	RX low frequency equalization zero frequency value standard mode 0: This field specifies the receiver zero frequency setting for the low frequency equalization function, by driving the

13.4.10.2.359 RX control register (lane0_rx_diag_rxctrl - lane3_rx_diag_rxctrl)

13.4.10.2.359.1 Offset

Register	Offset
lane0_rx_diag_rxctrl	81D8h
lane1_rx_diag_rxctrl	85D8h
lane2_rx_diag_rxctrl	89D8h
lane3_rx_diag_rxctrl	8DD8h

13.4.10.2.359.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_diag_rxctrl_15	rx_diag_rxctrl_14_12			rx_diag_rxctrl_11	rx_diag_rxctrl_10	rx_diag_rxctrl_9	rx_diag_rxctrl_8	rx_diag_rxctrl_7	rx_diag_rxctrl_6	rx_diag_rxctrl_5	rx_diag_rxctrl_4	rx_diag_rxctrl_3	rx_diag_rxctrl_2	rx_diag_rxctrl_1	rx_diag_rxctrl_0
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0

13.4.10.2.359.3 Fields

Field	Function
15 rx_diag_rxctrl_15	RX deserializer clock invert: This bit is used to optionally invert the deserializer clock (
14-12	Reserved

Table continues on the next page...

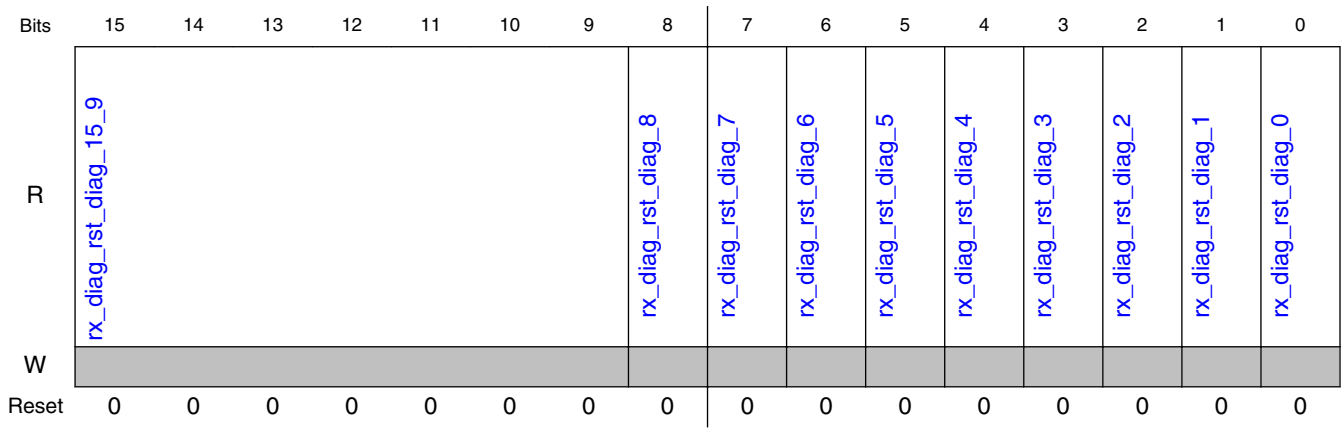
Field	Function
rx_diag_rxctrl_14_12	
11 rx_diag_rxctrl_11	PI output clock divider enable standard mode 3: This bit controls the
10 rx_diag_rxctrl_10	PI output clock divider enable standard mode 2: This bit controls the
9 rx_diag_rxctrl_9	PI output clock divider enable standard mode 1: This bit controls the
8 rx_diag_rxctrl_8	PI output clock divider enable standard mode 0: This bit controls the
7 rx_diag_rxctrl_7	Receiver CML to CMOS rate select value standard mode 3: This bit will drive the
6 rx_diag_rxctrl_6	Receiver CML to CMOS rate select value standard mode 2: This bit will drive the
5 rx_diag_rxctrl_5	Receiver CML to CMOS rate select value standard mode 1: This bit will drive the
4 rx_diag_rxctrl_4	Receiver CML to CMOS rate select value standard mode 0: This bit will drive the
3 rx_diag_rxctrl_3	RX interface subrate standard mode 3: This field specifies the receiver subrate, when
2 rx_diag_rxctrl_2	RX interface subrate standard mode 2: This field specifies the receiver subrate, when
1 rx_diag_rxctrl_1	RX interface subrate standard mode 1: This field specifies the receiver subrate, when
0 rx_diag_rxctrl_0	RX interface subrate standard mode 0: This field specifies the receiver subrate, when

13.4.10.2.360 Receiver control reset diagnostic register (lane0_rx_diag_rst_diag - lane3_rx_diag_rst_diag)

13.4.10.2.360.1 Offset

Register	Offset
lane0_rx_diag_rst_diag	81D9h
lane1_rx_diag_rst_diag	85D9h
lane2_rx_diag_rst_diag	89D9h
lane3_rx_diag_rst_diag	8DD9h

13.4.10.2.360.2 Diagram



13.4.10.2.360.3 Fields

Field	Function
15-9 rx_diag_rst_diag_15_9	Reserved
8 rx_diag_rst_diag_8	Current state of the rxda_clk_reset_n reset.
7 rx_diag_rst_diag_7	Current state of the rx_dig_reset_n reset.
6 rx_diag_rst_diag_6	Current state of the rxda_cdrlf_reset_n reset.
5 rx_diag_rst_diag_5	Current state of the rx_ree_reset_n reset.
4 rx_diag_rst_diag_4	Current state of the rx_lfps_det_filter_reset_n reset.
3 rx_diag_rst_diag_3	Current state of the rx_epi_ill_cal_lock_det_clk_reset_n reset.
2 rx_diag_rst_diag_2	Current state of the rx_epi_ill_cal_ref_clk_reset_n reset.
1	Current state of the rx_iqpi_ill_cal_lock_det_clk_reset_n reset.

Table continues on the next page...

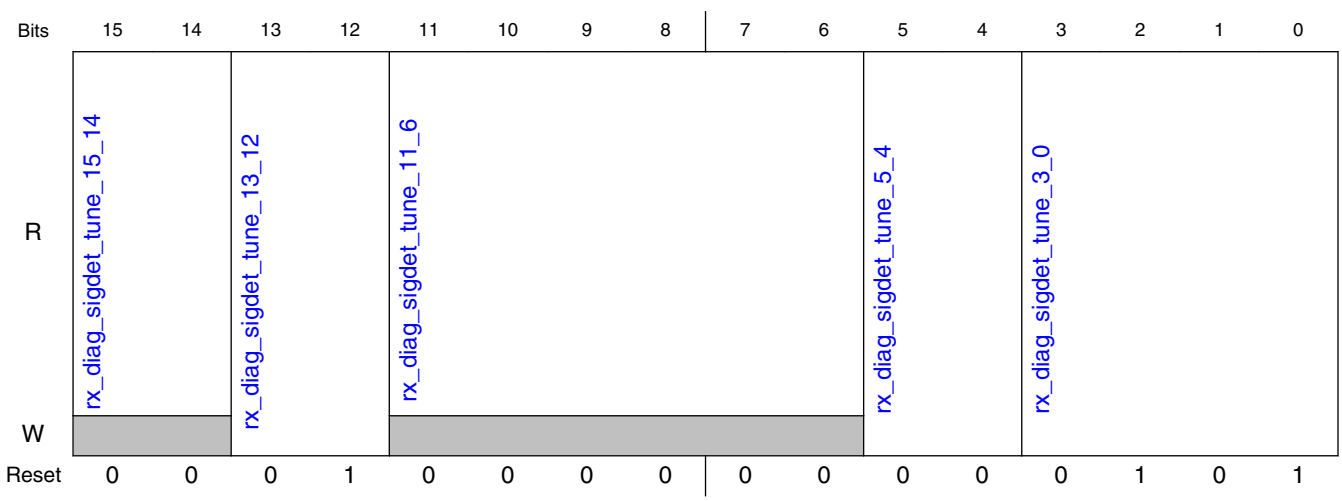
Field	Function
rx_diag_rst_diag_1	
0 rx_diag_rst_diag_0	Current state of the rx_iqpi_ill_cal_ref_clk_reset_n reset.

13.4.10.2.361 RX signal detect tuning and control register (lane0_rx_diag_sigdet_tune - lane3_rx_diag_sigdet_tune)

13.4.10.2.361.1 Offset

Register	Offset
lane0_rx_diag_sigdet_tune	81DCh
lane1_rx_diag_sigdet_tune	85DCh
lane2_rx_diag_sigdet_tune	89DCh
lane3_rx_diag_sigdet_tune	8DDCh

13.4.10.2.361.2 Diagram



13.4.10.2.361.3 Fields

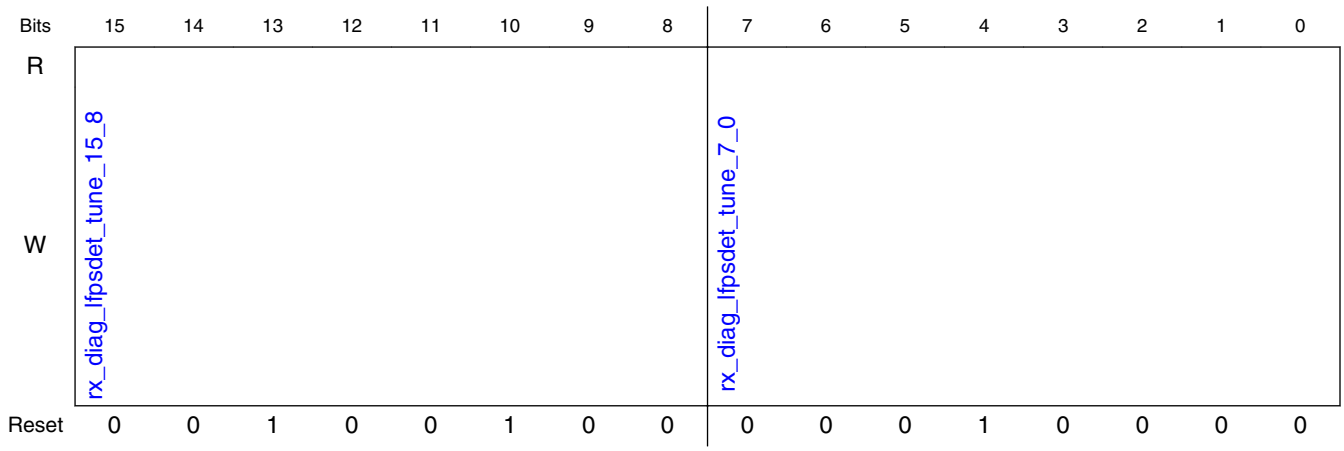
Field	Function
15-14 rx_diag_sigdet_tune_15_14	Reserved
13-12 rx_diag_sigdet_tune_13_12	Signal detect filter function select: Selects which of the two RX signal detect filter functions are enabled. As specified below.
11-6 rx_diag_sigdet_tune_11_6	Reserved
5-4 rx_diag_sigdet_tune_5_4	Signal definition to be provided by the analog team (Jira CCUPHY-2954)
3-0 rx_diag_sigdet_tune_3_0	Signal detect level: Sets the reference voltage level at which the comparators will detect a signal by driving the

13.4.10.2.362 RX LFPS detect tuning and control register (lane0_rx_diag_lfpsdet_tune - lane3_rx_diag_lfpsdet_tune)

13.4.10.2.362.1 Offset

Register	Offset
lane0_rx_diag_lfpsdet_tune	81DDh
lane1_rx_diag_lfpsdet_tune	85DDh
lane2_rx_diag_lfpsdet_tune	89DDh
lane3_rx_diag_lfpsdet_tune	8DDDh

13.4.10.2.362.2 Diagram



13.4.10.2.362.3 Fields

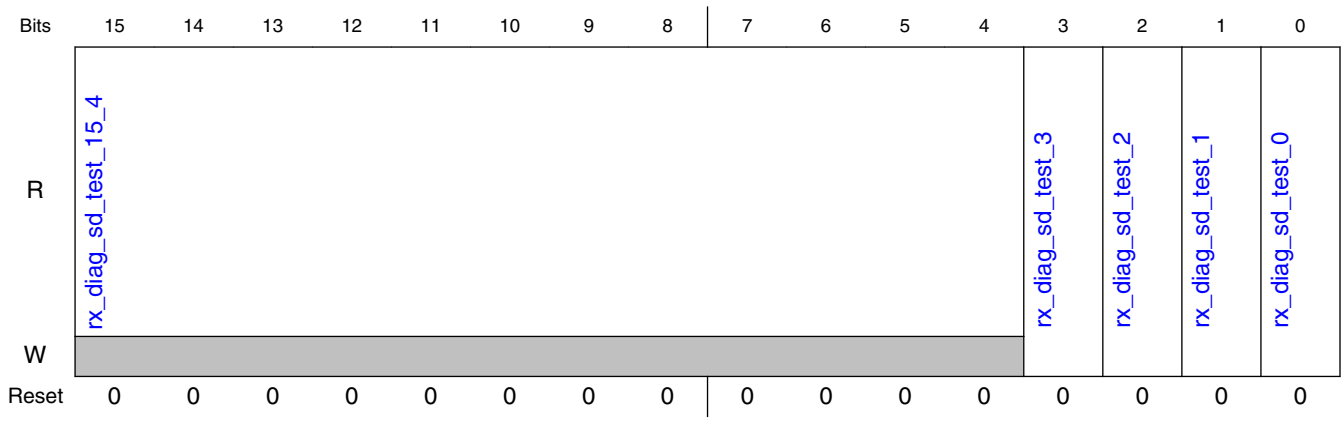
Field	Function
15-8 rx_diag_lfpsdet_tune_15_8	Signal definition to be provided by the analog team (Jira CCUPHY-2954)
7-0 rx_diag_lfpsdet_tune_7_0	LFPS detect level: Sets the reference voltage level at which the LFPS detect comparators will detect a signal. This register field drives the

13.4.10.2.363 Signal detect test register (lane0_rx_diag_sd_test - lane3_rx_diag_sd_test)

13.4.10.2.363.1 Offset

Register	Offset
lane0_rx_diag_sd_test	81DEh
lane1_rx_diag_sd_test	85DEh
lane2_rx_diag_sd_test	89DEh
lane3_rx_diag_sd_test	8DDEh

13.4.10.2.363.2 Diagram



13.4.10.2.363.3 Fields

Field	Function
15-4 rx_diag_sd_test_15_4	Reserved
3 rx_diag_sd_test_3	LFPS detected low test bit: This bit can be used to detect if the
2 rx_diag_sd_test_2	LFPS detected high test bit: This bit can be used to detect if the
1 rx_diag_sd_test_1	Signal detected low test bit: This bit can be used to detect if the
0 rx_diag_sd_test_0	Signal detected high test bit: This bit can be used to detect if the

13.4.10.2.364 RX sampler diagnostic control register (lane0_rx_diag_samp_ctrl - lane3_rx_diag_samp_ctrl)

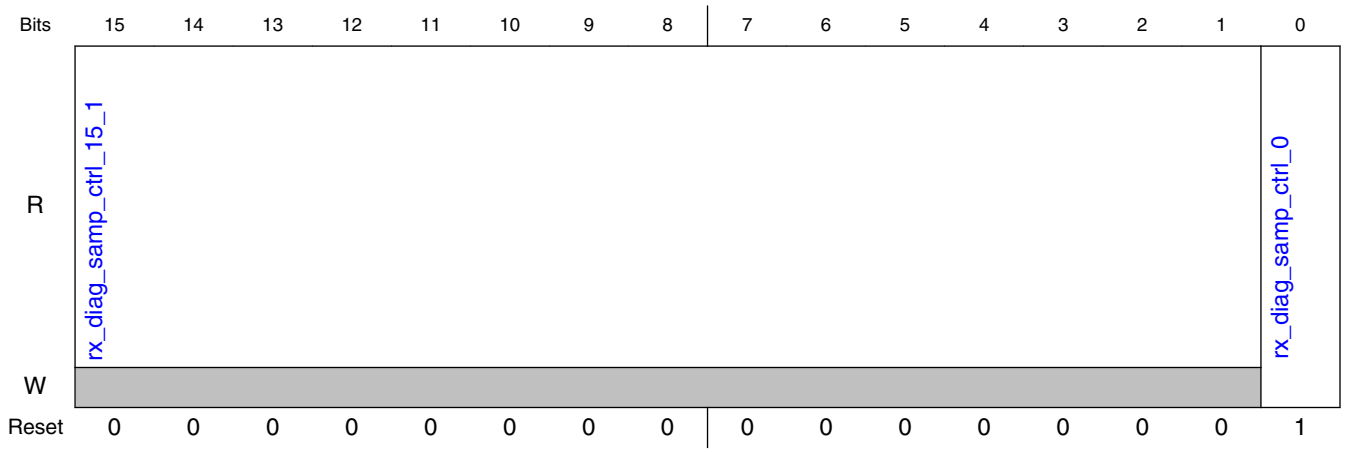
13.4.10.2.364.1 Offset

Register	Offset
lane0_rx_diag_samp_ctrl	81E0h
lane1_rx_diag_samp_ctrl	85E0h

Table continues on the next page...

Register	Offset
lane2_rx_diag_samp_ctrl	89E0h
lane3_rx_diag_samp_ctrl	8DE0h

13.4.10.2.364.2 Diagram



13.4.10.2.364.3 Fields

Field	Function
15-1 rx_diag_samp_ctrl_15_1	Reserved
0 rx_diag_samp_ctrl_0	Analog sampler

13.4.10.2.365 RX Sampler CML TO CMOS enable delay register (lane0_rx_diag_sc2c_delay - lane3_rx_diag_sc2c_delay)

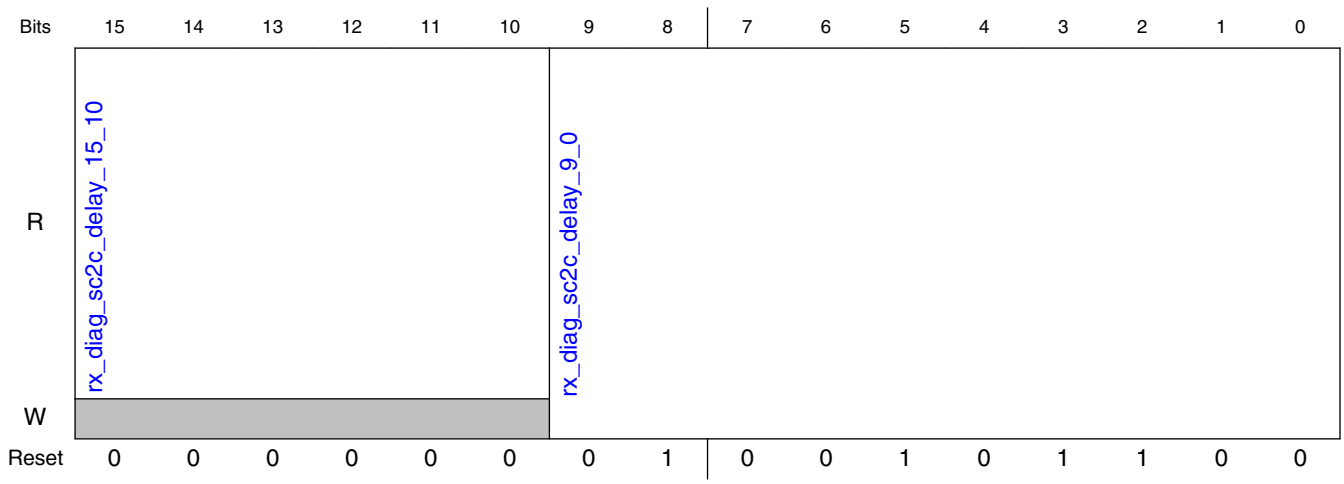
13.4.10.2.365.1 Offset

Register	Offset
lane0_rx_diag_sc2c_delay	81E1h
lane1_rx_diag_sc2c_delay	85E1h

Table continues on the next page...

Register	Offset
lane2_rx_diag_sc2c_delay	89E1h
lane3_rx_diag_sc2c_delay	8DE1h

13.4.10.2.365.2 Diagram



13.4.10.2.365.3 Fields

Field	Function
15-10 rx_diag_sc2c_delay_15_10	Reserved
9-0 rx_diag_sc2c_delay_9_0	Sampler CML to CMOS enable delay: This field specifies the number of PSM clocks between when

13.4.10.2.366 MPHY control register 1 (lane0_rx_diag_mphy_ctrl_1 - lane 3_rx_diag_mphy_ctrl_1)

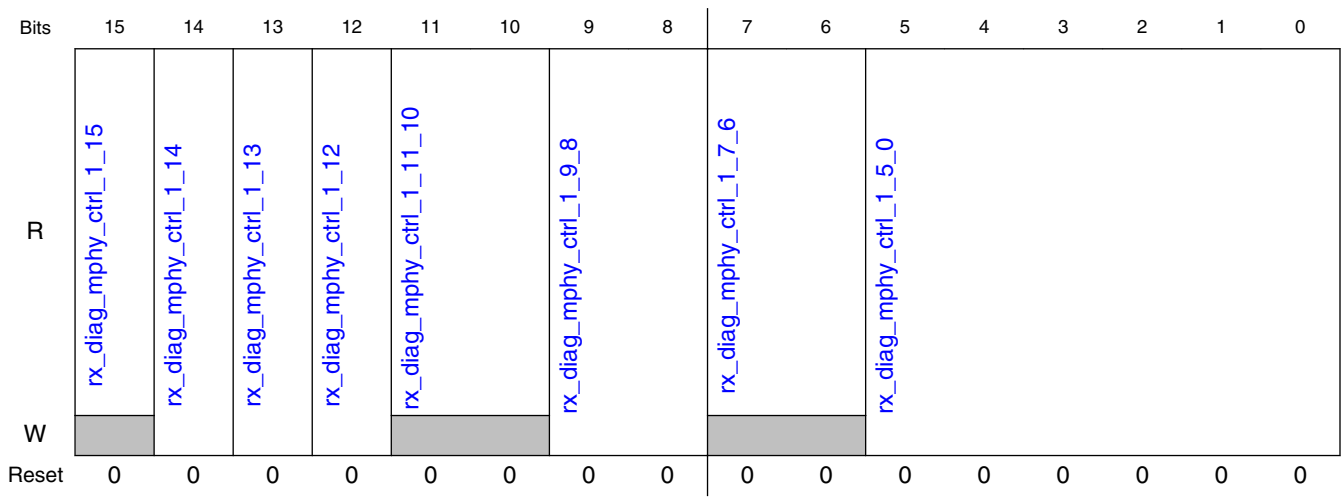
13.4.10.2.366.1 Offset

Register	Offset
lane0_rx_diag_mphy_ctrl_1	81E4h

Table continues on the next page...

Register	Offset
lane1_rx_diag_mphy_ctrl_1	85E4h
lane2_rx_diag_mphy_ctrl_1	89E4h
lane3_rx_diag_mphy_ctrl_1	8DE4h

13.4.10.2.366.2 Diagram



13.4.10.2.366.3 Fields

Field	Function
15 rx_diag_mphy_ctrl_1_15	Reserved
14 rx_diag_mphy_ctrl_1_14	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
13 rx_diag_mphy_ctrl_1_13	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
12 rx_diag_mphy_ctrl_1_12	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
11-10 rx_diag_mphy_ctrl_1_11_10	Reserved

Table continues on the next page...

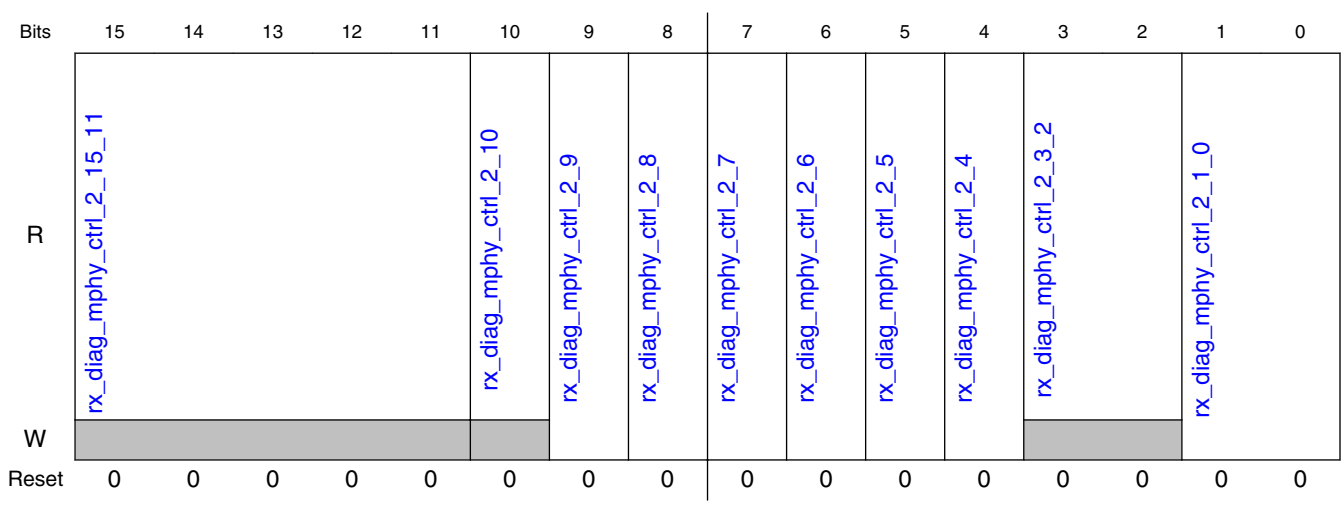
Field	Function
9-8 rx_diag_mphy_ctrl_1_9_8	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
7-6 rx_diag_mphy_ctrl_1_7_6	Reserved
5-0 rx_diag_mphy_ctrl_1_5_0	Signal definition to be provided by the analog team (Jira CCUPHY-2673)

13.4.10.2.367 MPHY control register 2 (lane0_rx_diag_mphy_ctrl_2 - lane 3_rx_diag_mphy_ctrl_2)

13.4.10.2.367.1 Offset

Register	Offset
lane0_rx_diag_mphy_ctrl_2	81E5h
lane1_rx_diag_mphy_ctrl_2	85E5h
lane2_rx_diag_mphy_ctrl_2	89E5h
lane3_rx_diag_mphy_ctrl_2	8DE5h

13.4.10.2.367.2 Diagram



13.4.10.2.367.3 Fields

Field	Function
15-11 rx_diag_mphy_ctrl_2_15_11	Reserved
10 rx_diag_mphy_ctrl_2_10	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
9 rx_diag_mphy_ctrl_2_9	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
8 rx_diag_mphy_ctrl_2_8	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
7 rx_diag_mphy_ctrl_2_7	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
6 rx_diag_mphy_ctrl_2_6	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
5 rx_diag_mphy_ctrl_2_5	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
4 rx_diag_mphy_ctrl_2_4	Signal definition to be provided by the analog team (Jira CCUPHY-2673)
3-2 rx_diag_mphy_ctrl_2_3_2	Reserved
1-0 rx_diag_mphy_ctrl_2_1_0	Signal definition to be provided by the analog team (Jira CCUPHY-2673)

13.4.10.2.368 RX loopback controller register (lane0_rx_diag_lpbk_ctrl - lane3_rx_diag_lpbk_ctrl)

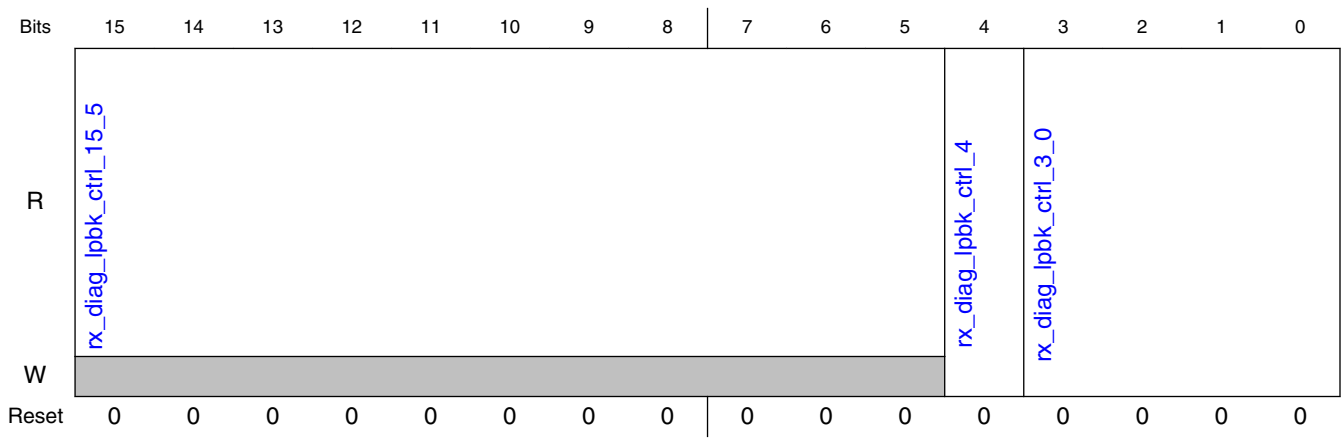
13.4.10.2.368.1 Offset

Register	Offset
lane0_rx_diag_lpbk_ctrl	81E8h

Table continues on the next page...

Register	Offset
lane1_rx_diag_lpbk_ctrl	85E8h
lane2_rx_diag_lpbk_ctrl	89E8h
lane3_rx_diag_lpbk_ctrl	8DE8h

13.4.10.2.368.2 Diagram



13.4.10.2.368.3 Fields

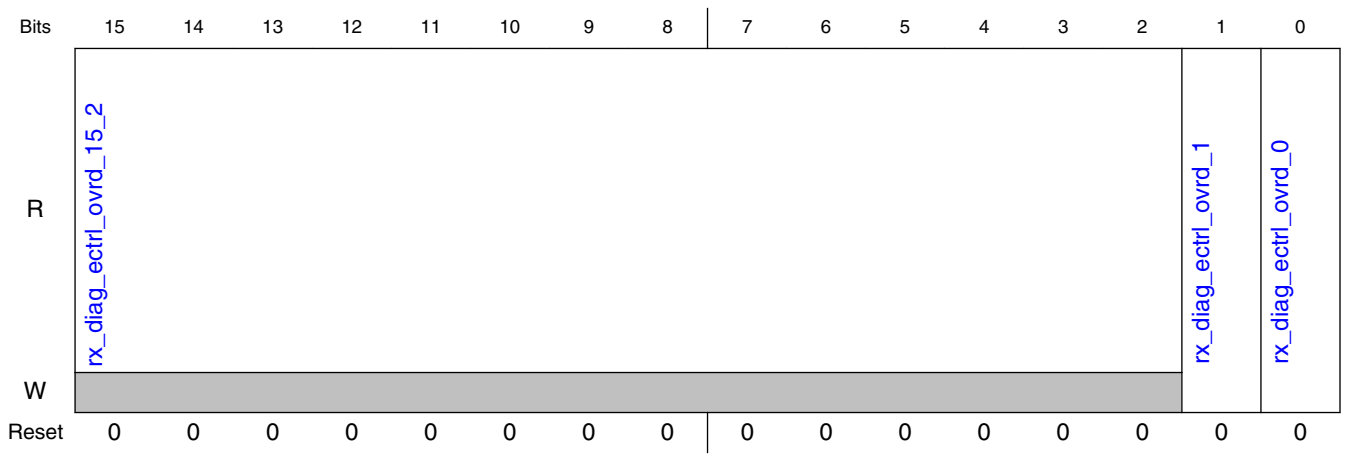
Field	Function
15-5 rx_diag_lpbk_ctrl_15_5	Reserved
4 rx_diag_lpbk_ctrl_4	Recovered clock loopback select: Selects which recovered clock to use when recovered clock loopback is enabled.
3-0 rx_diag_lpbk_ctrl_3_0	Attenuation settings: Sets the attenuation for the ISI generation loopback filter, as specified below.

13.4.10.2.369 RX extra enable control override register (lane0_rx_diag_ectrl_ovrd - lane3_rx_diag_ectrl_ovrd)

13.4.10.2.369.1 Offset

Register	Offset
lane0_rx_diag_ectrl_ovrd	81E9h
lane1_rx_diag_ectrl_ovrd	85E9h
lane2_rx_diag_ectrl_ovrd	89E9h
lane3_rx_diag_ectrl_ovrd	8DE9h

13.4.10.2.369.2 Diagram



13.4.10.2.369.3 Fields

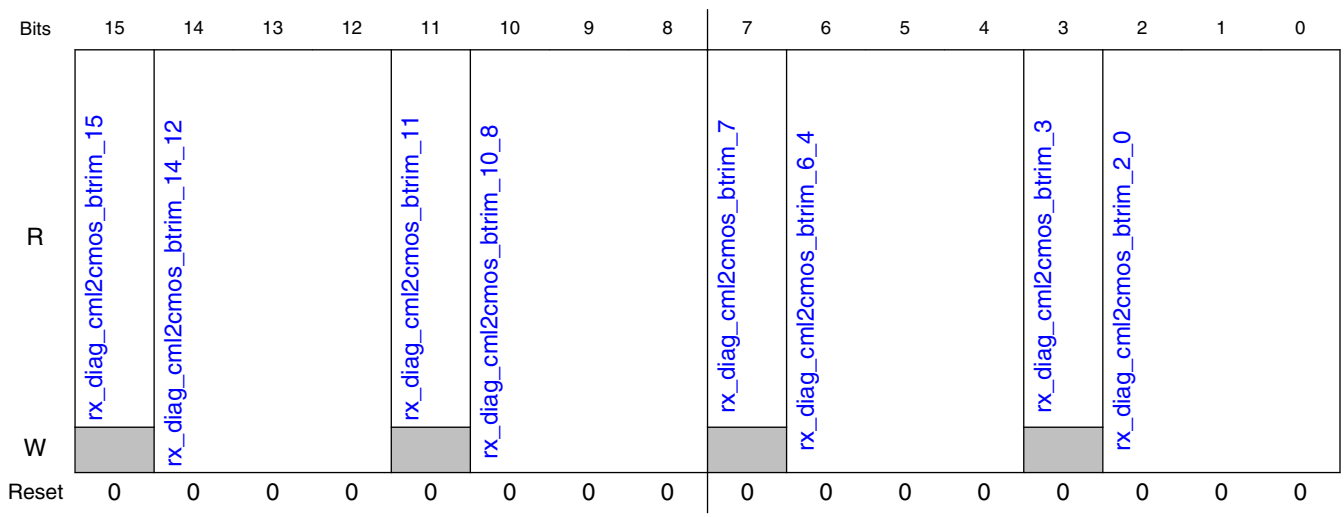
Field	Function
15-2 rx_diag_ectrl_ovrd_15_2	Reserved
1 rx_diag_ectrl_ovrd_1	Sampler CML to CMOS enable override enable : When active (1b1), the sampler CML to CMOS enable override bit in this register will drive the
0 rx_diag_ectrl_ovrd_0	Sampler CML to CMOS enable override : When enabled by the sampler CML to CMOS enable override enable bit in this register, this bit will drive the

13.4.10.2.370 CML to CMOS bias trim register (lane0_rx_diag_cml2cmos_btrim - lane3_rx_diag_cml2cmos_btrim)

13.4.10.2.370.1 Offset

Register	Offset
lane0_rx_diag_cml2cmos_btrim	81F0h
lane1_rx_diag_cml2cmos_btrim	85F0h
lane2_rx_diag_cml2cmos_btrim	89F0h
lane3_rx_diag_cml2cmos_btrim	8DF0h

13.4.10.2.370.2 Diagram



13.4.10.2.370.3 Fields

Field	Function
15 rx_diag_cml2cmos_btrim_15	Reserved
14-12 rx_diag_cml2cmos_btrim_14_12	CML to CMOS IQ bias sink current trim: Controls the
11 rx_diag_cml2cmos_btrim_11	Reserved
10-8	CML to CMOS IQ bias source current trim: Controls the

Table continues on the next page...

Field	Function
rx_diag_cml2cmos_btrim_10_8	
7 rx_diag_cml2cmos_btrim_7	Reserved
6-4 rx_diag_cml2cmos_btrim_6_4	CML to CMOS E bias sink current trim: Controls the
3 rx_diag_cml2cmos_btrim_3	Reserved
2-0 rx_diag_cml2cmos_btrim_2_0	CML to CMOS E bias source current trim: Controls the

13.4.10.2.371 RX bias gen control register 1 (lane0_rx_diag_bias_gen_ctrl1 - lane3_rx_diag_bias_gen_ctrl1)

13.4.10.2.371.1 Offset

Register	Offset
lane0_rx_diag_bias_gen_ctrl1	81F1h
lane1_rx_diag_bias_gen_ctrl1	85F1h
lane2_rx_diag_bias_gen_ctrl1	89F1h
lane3_rx_diag_bias_gen_ctrl1	8DF1h

13.4.10.2.371.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_diag_bias_gen_ctrl1_15_14		rx_diag_bias_gen_ctrl1_13_12		rx_diag_bias_gen_ctrl1_11_10		rx_diag_bias_gen_ctrl1_9_8		rx_diag_bias_gen_ctrl1_7_6		rx_diag_bias_gen_ctrl1_5_4		rx_diag_bias_gen_ctrl1_3_2		rx_diag_bias_gen_ctrl1_1_0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.371.3 Fields

Field	Function
15-14 rx_diag_bias_gen_ctrl1_15_14	Current programmability on the biasgen current supplying block xxx, by driving
13-12 rx_diag_bias_gen_ctrl1_13_12	Current programmability on the biasgen current supplying block xxx, by driving
11-10 rx_diag_bias_gen_ctrl1_11_10	Current programmability on the biasgen current supplying block xxx, by driving
9-8 rx_diag_bias_gen_ctrl1_9_8	Current programmability on the biasgen current supplying block xxx, by driving
7-6 rx_diag_bias_gen_ctrl1_7_6	Current programmability on the biasgen current supplying block xxx, by driving
5-4 rx_diag_bias_gen_ctrl1_5_4	Current programmability on the biasgen current supplying block xxx, by driving
3-2 rx_diag_bias_gen_ctrl1_3_2	Current programmability on the biasgen current supplying block xxx, by driving
1-0 rx_diag_bias_gen_ctrl1_1_0	Current programmability on the biasgen current supplying block xxx, by driving

13.4.10.2.372 RX bias gen control register 2 (lane0_rx_diag_bias_gen_ctrl2 - lane3_rx_diag_bias_gen_ctrl2)

13.4.10.2.372.1 Offset

Register	Offset
lane0_rx_diag_bias_gen_ctrl2	81F2h
lane1_rx_diag_bias_gen_ctrl2	85F2h
lane2_rx_diag_bias_gen_ctrl2	89F2h
lane3_rx_diag_bias_gen_ctrl2	8DF2h

13.4.10.2.372.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	rx_diag_bias_gen_ctrl2_15_14		rx_diag_bias_gen_ctrl2_13_12		rx_diag_bias_gen_ctrl2_11_10		rx_diag_bias_gen_ctrl2_9_8		rx_diag_bias_gen_ctrl2_7_6		rx_diag_bias_gen_ctrl2_5_4		rx_diag_bias_gen_ctrl2_3_2		rx_diag_bias_gen_ctrl2_1_0	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.372.3 Fields

Field	Function
15-14 rx_diag_bias_gen_ctrl2_15_14	Current programmability on the biasgen current supplying block xxx, by driving
13-12 rx_diag_bias_gen_ctrl2_13_12	Current programmability on the biasgen current supplying block xxx, by driving
11-10	Current programmability on the biasgen current supplying block xxx, by driving

Table continues on the next page...

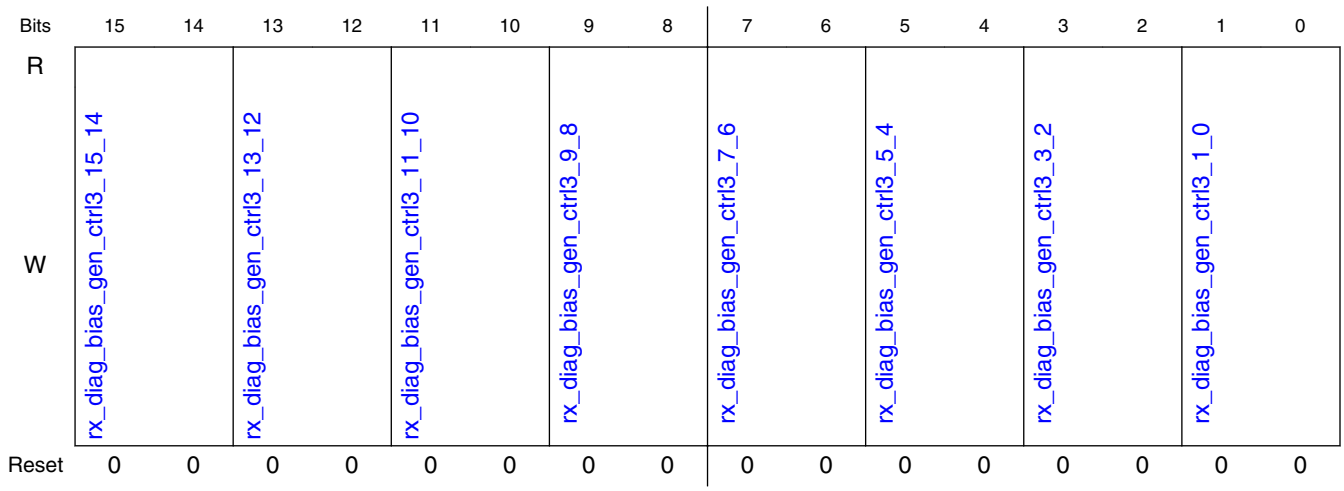
Field	Function
rx_diag_bias_gen_ctrl2_11_10	
9-8 rx_diag_bias_gen_ctrl2_9_8	Current programmability on the biasgen current supplying block xxx, by driving
7-6 rx_diag_bias_gen_ctrl2_7_6	Current programmability on the biasgen current supplying block xxx, by driving
5-4 rx_diag_bias_gen_ctrl2_5_4	Current programmability on the biasgen current supplying block xxx, by driving
3-2 rx_diag_bias_gen_ctrl2_3_2	Current programmability on the biasgen current supplying block xxx, by driving
1-0 rx_diag_bias_gen_ctrl2_1_0	Current programmability on the biasgen current supplying block xxx, by driving

13.4.10.2.373 RX bias gen control register 3 (lane0_rx_diag_bias_gen_ctrl3 - lane3_rx_diag_bias_gen_ctrl3)

13.4.10.2.373.1 Offset

Register	Offset
lane0_rx_diag_bias_gen_ctrl3	81F3h
lane1_rx_diag_bias_gen_ctrl3	85F3h
lane2_rx_diag_bias_gen_ctrl3	89F3h
lane3_rx_diag_bias_gen_ctrl3	8DF3h

13.4.10.2.373.2 Diagram



13.4.10.2.373.3 Fields

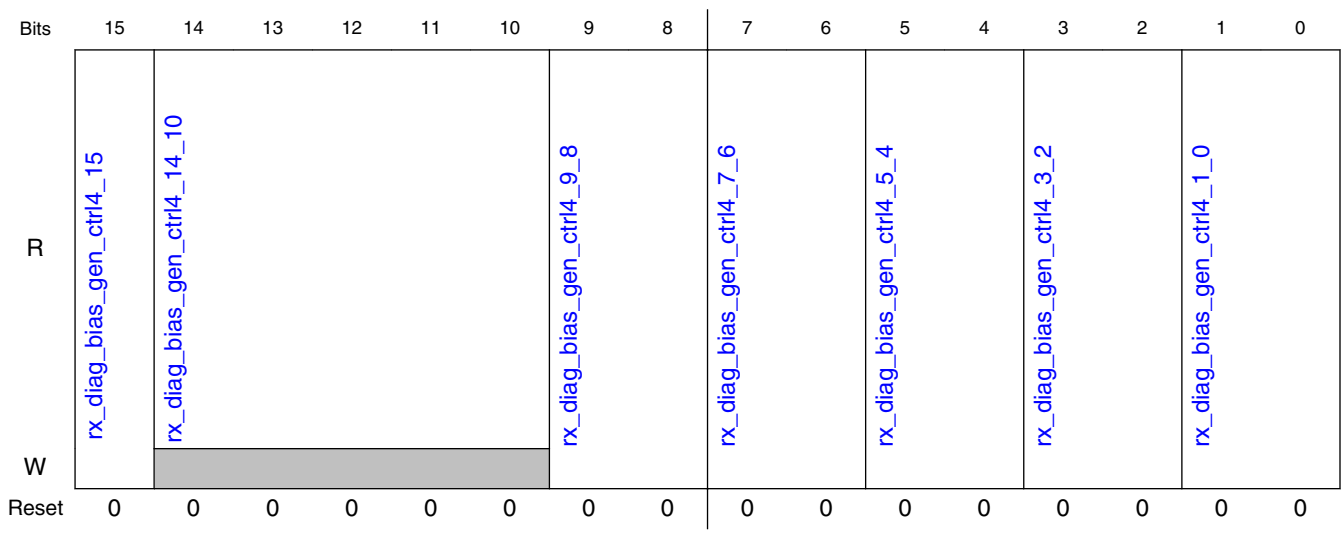
Field	Function
15-14 rx_diag_bias_gen_ctrl3_15_14	Current programmability on the biasgen current supplying block xxx, by driving
13-12 rx_diag_bias_gen_ctrl3_13_12	Current programmability on the biasgen current supplying block xxx, by driving
11-10 rx_diag_bias_gen_ctrl3_11_10	Current programmability on the biasgen current supplying block xxx, by driving
9-8 rx_diag_bias_gen_ctrl3_9_8	Current programmability on the biasgen current supplying block xxx, by driving
7-6 rx_diag_bias_gen_ctrl3_7_6	Current programmability on the biasgen current supplying block xxx, by driving
5-4 rx_diag_bias_gen_ctrl3_5_4	Current programmability on the biasgen current supplying block xxx, by driving
3-2 rx_diag_bias_gen_ctrl3_3_2	Current programmability on the biasgen current supplying block xxx, by driving
1-0 rx_diag_bias_gen_ctrl3_1_0	Current programmability on the biasgen current supplying block xxx, by driving

13.4.10.2.374 RX bias gen control register 4 (lane0_rx_diag_bias_gen_ctrl4 - lane3_rx_diag_bias_gen_ctrl4)

13.4.10.2.374.1 Offset

Register	Offset
lane0_rx_diag_bias_gen_ctrl4	81F4h
lane1_rx_diag_bias_gen_ctrl4	85F4h
lane2_rx_diag_bias_gen_ctrl4	89F4h
lane3_rx_diag_bias_gen_ctrl4	8DF4h

13.4.10.2.374.2 Diagram



13.4.10.2.374.3 Fields

Field	Function
15 rx_diag_bias_gen_ctrl4_15	Enable the base unit on all the current outputs from the RX bias generation block, by driving
14-10 rx_diag_bias_gen_ctrl4_14_10	Reserved

Table continues on the next page...

Field	Function
9-8 rx_diag_bias_gen_ctrl4_9_8	Current programmability on the biasgen current supplying block xxx, by driving
7-6 rx_diag_bias_gen_ctrl4_7_6	Current programmability on the biasgen current supplying block xxx, by driving
5-4 rx_diag_bias_gen_ctrl4_5_4	Current programmability on the biasgen current supplying block xxx, by driving
3-2 rx_diag_bias_gen_ctrl4_3_2	Current programmability on the biasgen current supplying block xxx, by driving
1-0 rx_diag_bias_gen_ctrl4_1_0	Current programmability on the biasgen current supplying block xxx, by driving

13.4.10.2.375 RX boundary scan test mode register (lane0_rx_diag_bs_tm - lane3_rx_diag_bs_tm)

13.4.10.2.375.1 Offset

Register	Offset
lane0_rx_diag_bs_tm	81F5h
lane1_rx_diag_bs_tm	85F5h
lane2_rx_diag_bs_tm	89F5h
lane3_rx_diag_bs_tm	8DF5h

13.4.10.2.375.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_diag_bs_tm_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.375.3 Fields

Field	Function
15-0	Register definition to be provided by the analog team (Jira CCUPHY-2954)

Field	Function
rx_diag_bs_tm_15_0	

13.4.10.2.376 RX receiver front end test mode register 1 (lane0_rx_diag_rxfe_tm1 - lane3_rx_diag_rxfe_tm1)

13.4.10.2.376.1 Offset

Register	Offset
lane0_rx_diag_rxfe_tm1	81F6h
lane1_rx_diag_rxfe_tm1	85F6h
lane2_rx_diag_rxfe_tm1	89F6h
lane3_rx_diag_rxfe_tm1	8DF6h

13.4.10.2.376.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_diag_rxfe_tm1_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.376.3 Fields

Field	Function
15-0 rx_diag_rxfe_tm1_15_0	Register definition to be provided by the analog team (Jira CCUPHY-2954)

13.4.10.2.377 RX receiver front end test mode register 2 (lane0_rx_diag_rxfe_tm2 - lane3_rx_diag_rxfe_tm2)

13.4.10.2.377.1 Offset

Register	Offset
lane0_rx_diag_rxfe_tm2	81F7h

Table continues on the next page...

Register	Offset
lane1_rx_diag_rxfe_tm2	85F7h
lane2_rx_diag_rxfe_tm2	89F7h
lane3_rx_diag_rxfe_tm2	8DF7h

13.4.10.2.377.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_diag_rxfe_tm2_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.377.3 Fields

Field	Function
15-0 rx_diag_rxfe_tm2_15_0	Register definition to be provided by the analog team (Jira CCUPHY-2954)

13.4.10.2.378 Receiver digital cover your alternatives register (lane0_rx_diag_dcya - lane3_rx_diag_dcya)

13.4.10.2.378.1 Offset

Register	Offset
lane0_rx_diag_dcya	81FEh
lane1_rx_diag_dcya	85FEh
lane2_rx_diag_dcya	89FEh
lane3_rx_diag_dcya	8DFEh

13.4.10.2.378.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_diag_dcya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.378.3 Fields

Field	Function
15-0 rx_diag_dcya_15_0	Reserved

13.4.10.2.379 Receiver analog cover your alternatives register (lane0_rx_diag_acya - lane3_rx_diag_acya)

13.4.10.2.379.1 Offset

Register	Offset
lane0_rx_diag_acya	81FFh
lane1_rx_diag_acya	85FFh
lane2_rx_diag_acya	89FFh
lane3_rx_diag_acya	8DFFh

13.4.10.2.379.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_diag_acya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.379.3 Fields

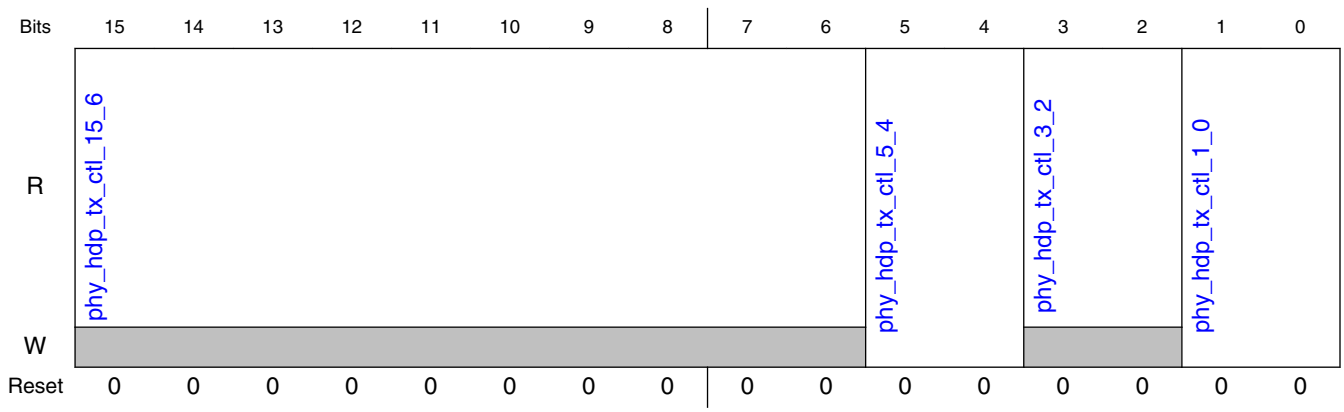
Field	Function
15-0 rx_diag_acya_15_0	Reserved

13.4.10.2.380 HDP Lane Configuration register (lane0_phy_hdp_tx_ctl - lane3_phy_hdp_tx_ctl)

13.4.10.2.380.1 Offset

Register	Offset
lane0_phy_hdp_tx_ctl	C408h
lane1_phy_hdp_tx_ctl	C448h
lane2_phy_hdp_tx_ctl	C488h
lane3_phy_hdp_tx_ctl	C4C8h

13.4.10.2.380.2 Diagram



13.4.10.2.380.3 Fields

Field	Function
15-6 phy_hdp_tx_ctl_15_6	Reserved
5-4 phy_hdp_tx_ctl_5_4	Tx Voltage Level - Drives tx_vmargin PMA input for the mapped PMA lane (for functional and isolation modes). This field is used to set the DP Voltage Swing Level (0b00 = Level 0, 0b01 = Level 1, 0b10 = Level 2 and 0b11 = Level 3).
3-2 phy_hdp_tx_ctl_3_2	Reserved
1-0 phy_hdp_tx_ctl_1_0	Tx De-emphasis setting - Drives tx_deemphasis PMA input for the mapped PMA lane (for functional and isolation modes). This field is used to set the DP Pre-emphasis Level (0b00 = Level 0, 0b01 = Level 1, 0b10 = Level 2 and 0b11 = Level 3).

13.4.10.2.381 DP Tx data low isolation register (lane0_phy_dp_iso_tx_data_lo - lane3_phy_dp_iso_tx_data_lo)

13.4.10.2.381.1 Offset

Register	Offset
lane0_phy_dp_iso_tx_data_lo	C41Ch
lane1_phy_dp_iso_tx_data_lo	C45Ch
lane2_phy_dp_iso_tx_data_lo	C49Ch
lane3_phy_dp_iso_tx_data_lo	C4DCh

13.4.10.2.381.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	phy_dp_iso_tx_data_lo_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.381.3 Fields

Field	Function
15-0 phy_dp_iso_tx_data_lo_15_0	Drives phy_pma_tx_data_ln_XX[15:0] PHY input for the associated lane when in PHY and PMA isolation modes. (Not re-synchronized)

13.4.10.2.382 DP Tx data high isolation register (lane0_phy_dp_iso_tx_data_hi - lane3_phy_dp_iso_tx_data_hi)

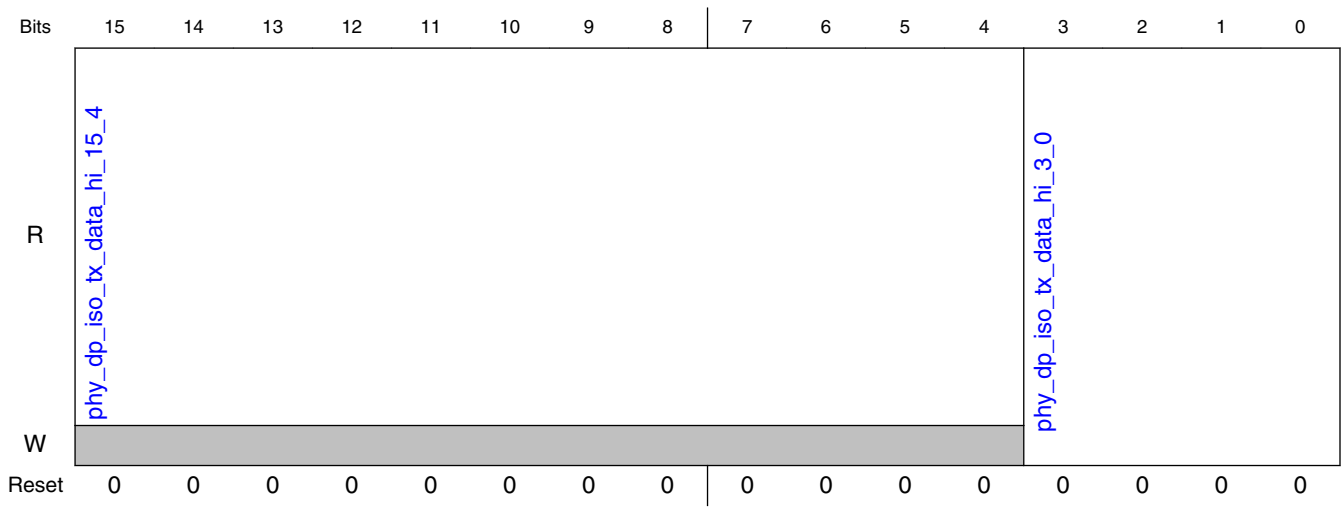
13.4.10.2.382.1 Offset

Register	Offset
lane0_phy_dp_iso_tx_data_hi	C41Dh
lane1_phy_dp_iso_tx_data_hi	C45Dh

Table continues on the next page...

Register	Offset
lane2_phy_dp_iso_tx_data_hi	C49Dh
lane3_phy_dp_iso_tx_data_hi	C4DDh

13.4.10.2.382.2 Diagram



13.4.10.2.382.3 Fields

Field	Function
15-4 phy_dp_iso_tx_data_hi_15_4	Reserved
3-0 phy_dp_iso_tx_data_hi_3_0	Drives phy_pma_tx_data_ln_XX[19:16] PHY input for the associated lane when in PHY isolation modes. (Not re-synchronized)

13.4.10.2.383 PMA transceiver control register (lane0_phy_pma_xcvr_ctrl - lane3_phy_pma_xcvr_ctrl)

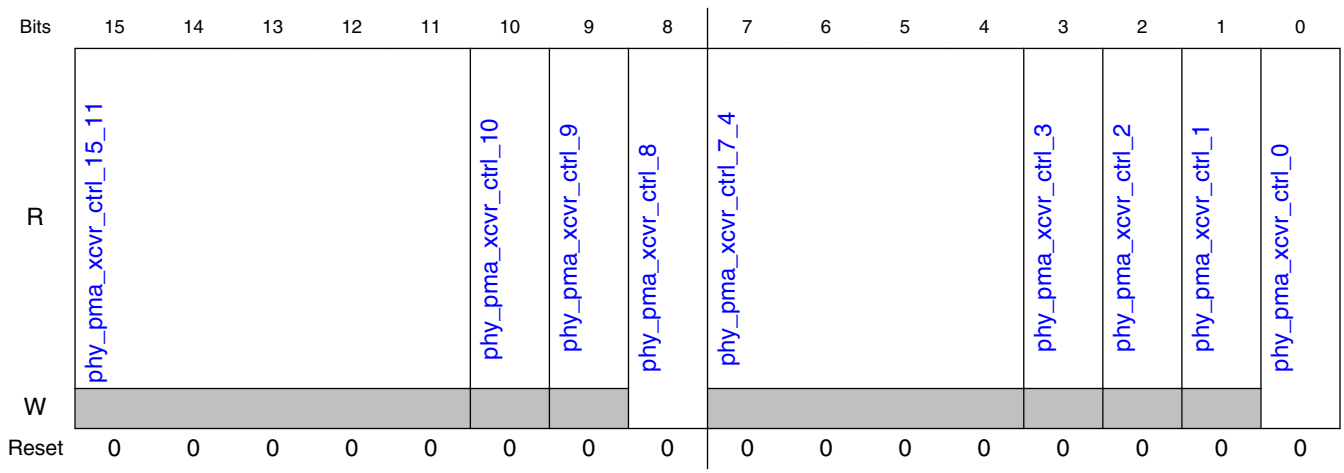
13.4.10.2.383.1 Offset

Register	Offset
lane0_phy_pma_xcvr_ctrl	CC00h

Table continues on the next page...

Register	Offset
lane1_phy_pma_xcvr_ctrl	CC40h
lane2_phy_pma_xcvr_ctrl	CC80h
lane3_phy_pma_xcvr_ctrl	CCC0h

13.4.10.2.383.2 Diagram



13.4.10.2.383.3 Fields

Field	Function
15-11 phy_pma_xcvr_ctrl_15_11	Reserved
10 phy_pma_xcvr_ctrl_10	Current value of xcvr_lane_en_ack PMA output for the associated lane
9 phy_pma_xcvr_ctrl_9	Reserved
8 phy_pma_xcvr_ctrl_8	Drives the tx_differential_invert PMA input for the associated lane.
7-4 phy_pma_xcvr_ctrl_7_4	Reserved
3 phy_pma_xcvr_ctrl_3	Current value of rx_bist_status_ln_{nnn} PMA output for the associated lane.

Table continues on the next page...

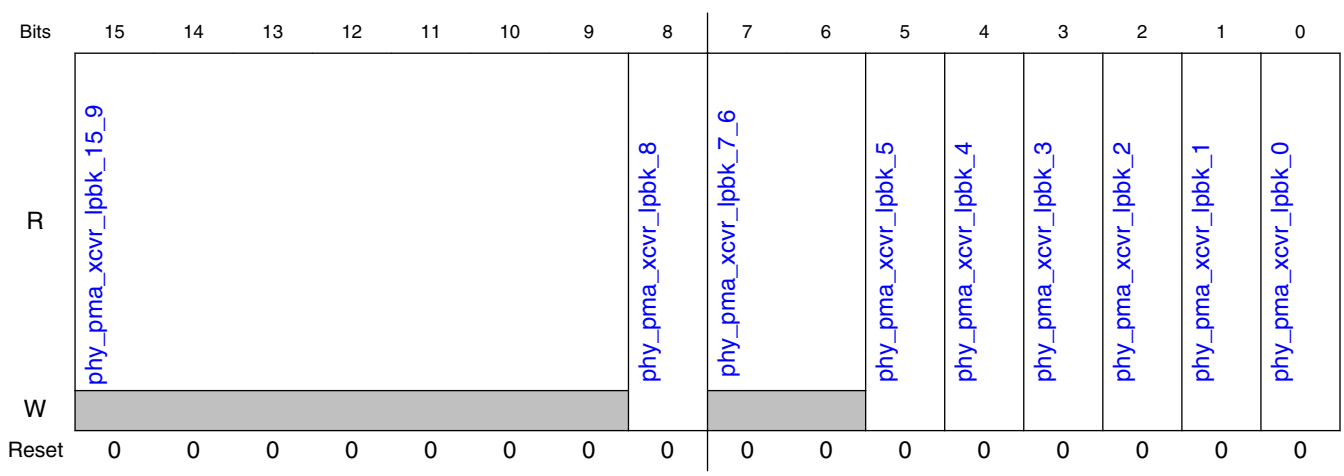
Field	Function
2 phy_pma_xcvr_ctrl_2	Current value of rx_bist_err_toggle_ln_{nnn} PMA output for the associated lane.
1 phy_pma_xcvr_ctrl_1	Current value of rx_bist_sync_ln_{nnn} PMA output for the associated lane.
0 phy_pma_xcvr_ctrl_0	Reserved

13.4.10.2.384 PMA loopback control register (lane0_phy_pma_xcvr_lpbk - lane3_phy_pma_xcvr_lpbk)

13.4.10.2.384.1 Offset

Register	Offset
lane0_phy_pma_xcvr_lpbk	CC01h
lane1_phy_pma_xcvr_lpbk	CC41h
lane2_phy_pma_xcvr_lpbk	CC81h
lane3_phy_pma_xcvr_lpbk	CCC1h

13.4.10.2.384.2 Diagram



13.4.10.2.384.3 Fields

Field	Function
15-9 phy_pma_xcvr_lpbk_15_9	Reserved
8 phy_pma_xcvr_lpbk_8	Drives the tx_bist_hold PMA input for all lanes in the associated link. Synchronized to transmit data rate clock. This signal can be used during test to start each lanes BIST engine on the same clock cycle to test lane to lane transmit skew.
7-6 phy_pma_xcvr_lpbk_7_6	Reserved
5 phy_pma_xcvr_lpbk_5	Drives the txrx_fe_parallel_lpbk_en_ln_{nnn} PMA input for the associated lane.
4 phy_pma_xcvr_lpbk_4	Drives the txrx_ne_parallel_lpbk_en_ln_{nnn} PMA input for the associated lane.
3 phy_pma_xcvr_lpbk_3	Drives the txrx_recovered_clk_lpbk_en_ln_{nnn} PMA input for the associated lane.
2 phy_pma_xcvr_lpbk_2	Drives the txrx_line_lpbk_en_ln_{nnn} PMA input for the associated lane.
1 phy_pma_xcvr_lpbk_1	Drives the txrx_isi_gen_lpbk_en_ln_{nnn} PMA input for the associated lane.
0 phy_pma_xcvr_lpbk_0	Drives the txrx_serial_lpbk_en_ln_{nnn} PMA input for the associated lane.

13.4.10.2.385 PMA PSM current state lower register (lane0_phy_pma_psm_state_lo - lane3_phy_pma_psm_state_lo)

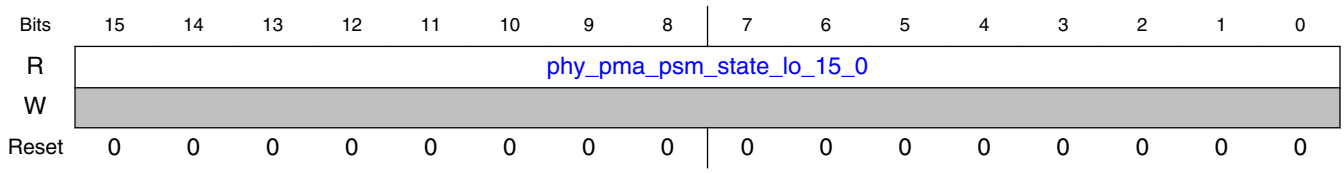
13.4.10.2.385.1 Offset

Register	Offset
lane0_phy_pma_psm_state_lo	CC04h
lane1_phy_pma_psm_state_lo	CC44h
lane2_phy_pma_psm_state_lo	CC84h

Table continues on the next page...

Register	Offset
lane3_phy_pma_psm_state_lo	CCC4h

13.4.10.2.385.2 Diagram



13.4.10.2.385.3 Fields

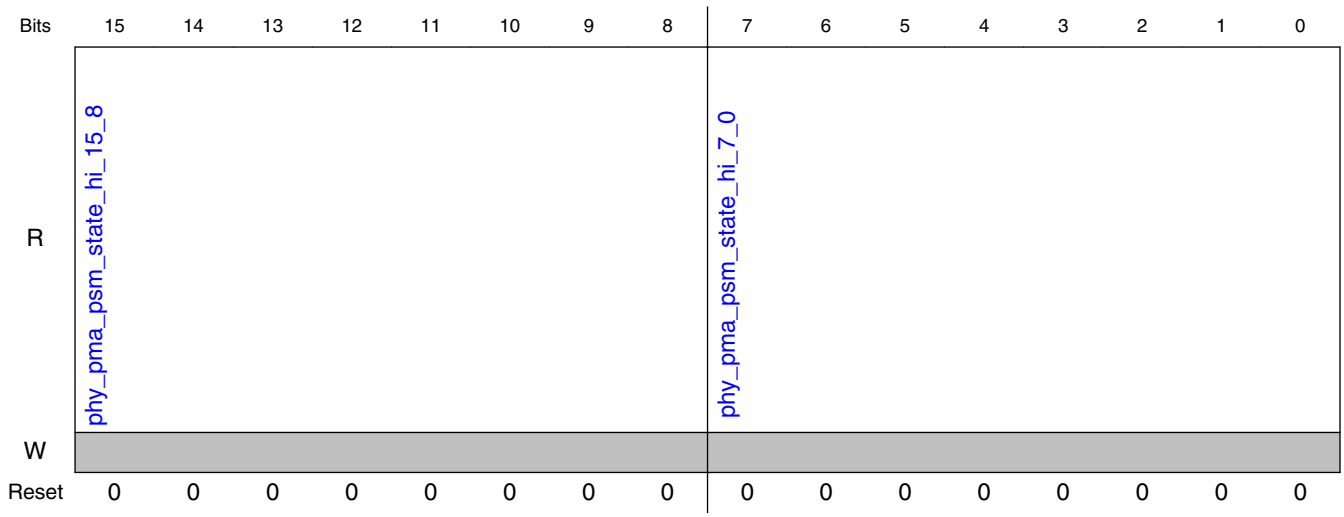
Field	Function
15-0 phy_pma_psm_state_lo_15_0	PMA PSM[15:0] : Current state of the PMA power state machine. (Not re-synchronized to apb_pclk)

13.4.10.2.386 PMA PSM current state higher register (lane0_phy_pma_psm_state_hi - lane3_phy_pma_psm_state_hi)

13.4.10.2.386.1 Offset

Register	Offset
lane0_phy_pma_psm_state_hi	CC05h
lane1_phy_pma_psm_state_hi	CC45h
lane2_phy_pma_psm_state_hi	CC85h
lane3_phy_pma_psm_state_hi	CCC5h

13.4.10.2.386.2 Diagram



13.4.10.2.386.3 Fields

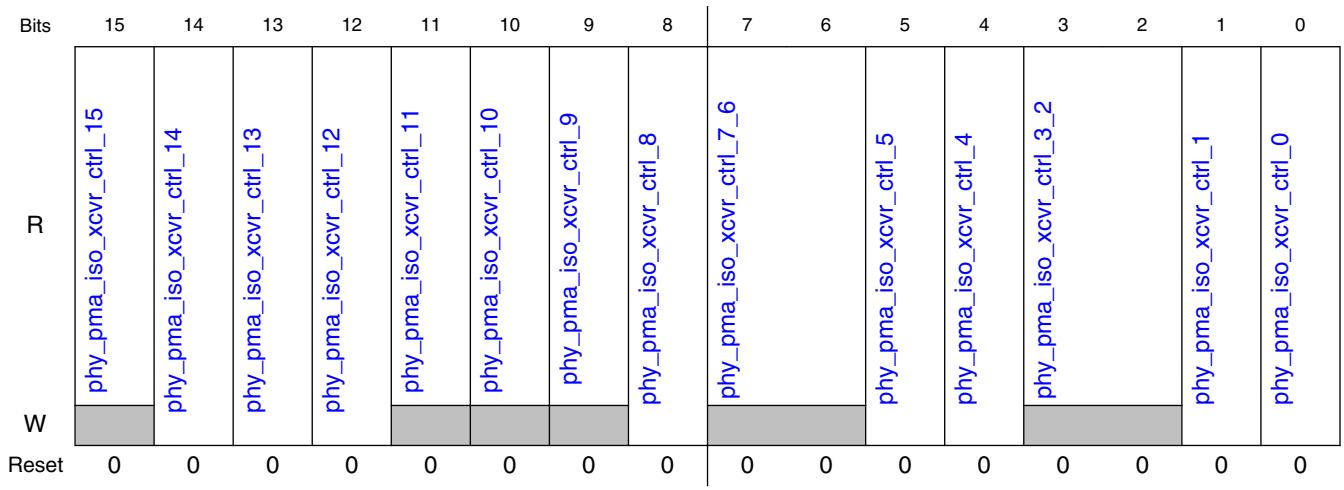
Field	Function
15-8 phy_pma_psm_state_hi_15_8	Reserved
7-0 phy_pma_psm_state_hi_7_0	PMA PSM[23:16] : Current state of the PMA power state machine. (Not re-synchronized to apb_pclk)

13.4.10.2.387 PMA Isolation Transceiver control register (lane0_phy_pma_iso_xcvr_ctrl - lane3_phy_pma_iso_xcvr_ctrl)

13.4.10.2.387.1 Offset

Register	Offset
lane0_phy_pma_iso_xcvr_ctrl	CC10h
lane1_phy_pma_iso_xcvr_ctrl	CC50h
lane2_phy_pma_iso_xcvr_ctrl	CC90h
lane3_phy_pma_iso_xcvr_ctrl	CCD0h

13.4.10.2.387.2 Diagram



13.4.10.2.387.3 Fields

Field	Function
15 phy_pma_iso_xcvr_ctrl_15	Current value of xcvr_pll_clk_en_ack PMA output for the associated lane.
14 phy_pma_iso_xcvr_ctrl_14	Drives xcvr_pll_clk_en PMA input for the associated lane when in PMA isolation mode.
13 phy_pma_iso_xcvr_ctrl_13	Drives tx_lfps_en PMA input for the associated lane when in PMA isolation mode.
12 phy_pma_iso_xcvr_ctrl_12	Drives tx_elec_idle PMA input for the associated lane when in PMA isolation mode.
11 phy_pma_iso_xcvr_ctrl_11	Current value of xcvr_psm_ready PMA output for the associated lane.
10 phy_pma_iso_xcvr_ctrl_10	Current value of tx_rcv_detected PMA output for the associated lane when in PMA isolation mode. (Not re-synchronized to apb_pclk)
9 phy_pma_iso_xcvr_ctrl_9	Current value of tx_rcv_detect_done PMA output for the associated lane.
8 phy_pma_iso_xcvr_ctrl_8	Drives tx_rcv_detect_en PMA input for the associated lane when in PMA isolation mode.

Table continues on the next page...

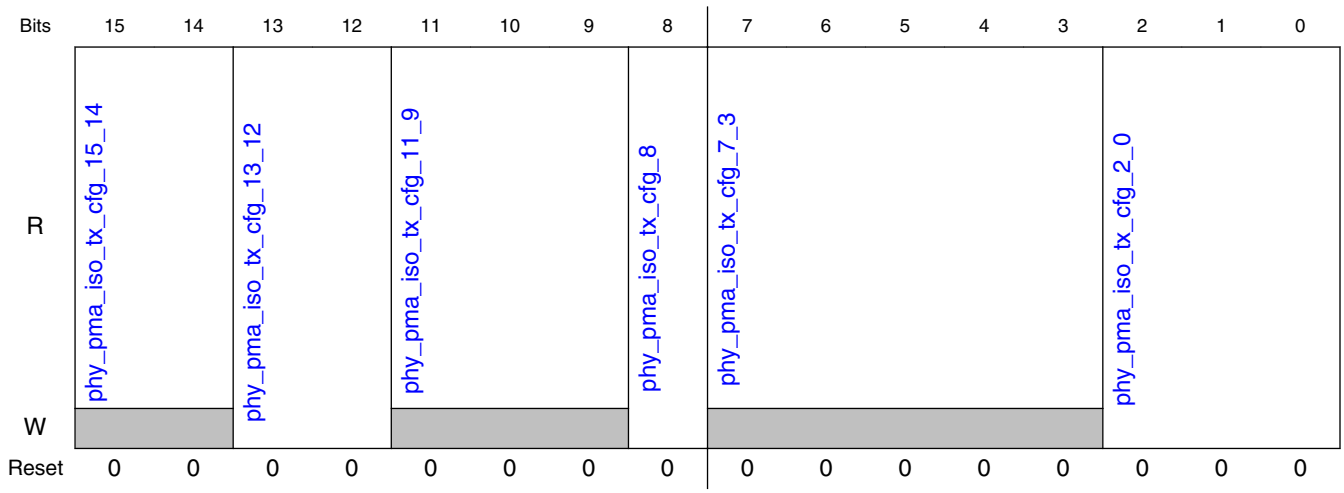
Field	Function
7-6 phy_pma_iso_xcvr_ctrl_7_6	Reserved
5 phy_pma_iso_xcvr_ctrl_5	Drives xcvr_link_reset_n PMA input for the associated lane when in PMA isolation mode.
4 phy_pma_iso_xcvr_ctrl_4	Drives xcvr_lane_suspend PMA input for the associated lane when in PMA isolation mode.
3-2 phy_pma_iso_xcvr_ctrl_3_2	Reserved
1 phy_pma_iso_xcvr_ctrl_1	Reserved
0 phy_pma_iso_xcvr_ctrl_0	Drives xcvr_lane_en PMA input for the associated lane when in PMA isolation mode.

13.4.10.2.388 PMA TX configuration register (lane0_phy_pma_iso_tx_cfg - lane3_phy_pma_iso_tx_cfg)

13.4.10.2.388.1 Offset

Register	Offset
lane0_phy_pma_iso_tx_cfg	CC11h
lane1_phy_pma_iso_tx_cfg	CC51h
lane2_phy_pma_iso_tx_cfg	CC91h
lane3_phy_pma_iso_tx_cfg	CCD1h

13.4.10.2.388.2 Diagram



13.4.10.2.388.3 Fields

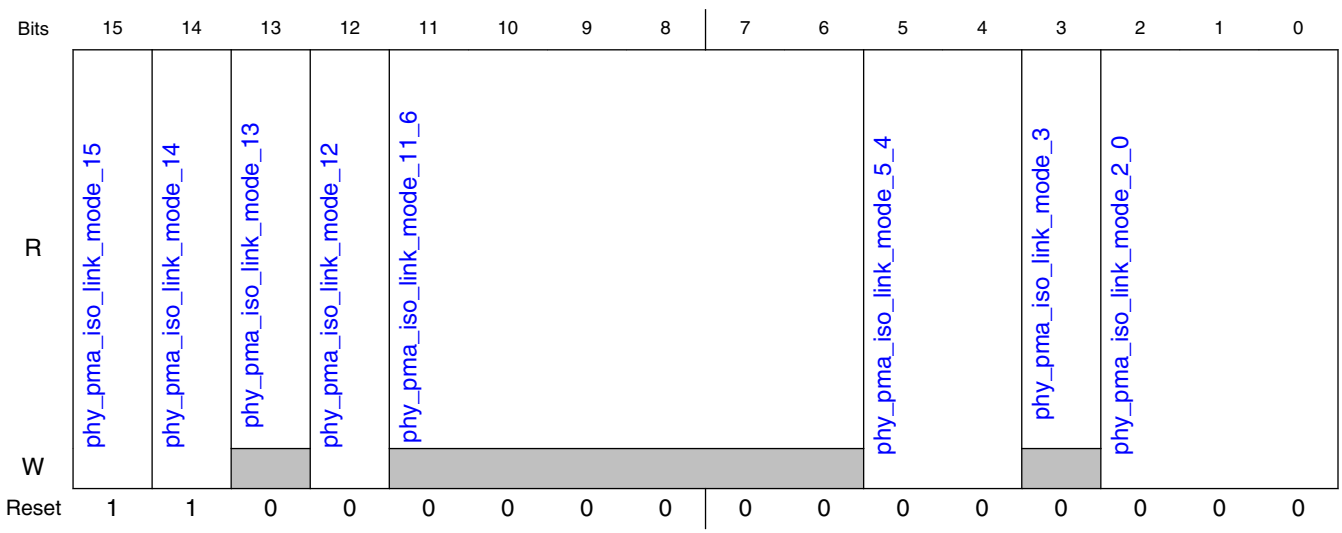
Field	Function
15-14 phy_pma_iso_tx_cfg_15_14	Reserved
13-12 phy_pma_iso_tx_cfg_13_12	Drives tx_deemphasis PMA input for the associated lane when in PMA isolation mode.
11-9 phy_pma_iso_tx_cfg_11_9	Reserved
8 phy_pma_iso_tx_cfg_8	Drives tx_low_power_swing_en PMA input for the associated lane when in PMA isolation mode.
7-3 phy_pma_iso_tx_cfg_7_3	Reserved
2-0 phy_pma_iso_tx_cfg_2_0	Drives tx_vmargin PMA input for the associated lane when in PMA isolation mode.

13.4.10.2.389 PMA Isolation mode control register (lane0_phy_pma_iso_link_mode - lane3_phy_pma_iso_link_mode)

13.4.10.2.389.1 Offset

Register	Offset
lane0_phy_pma_iso_link_mode	CC12h
lane1_phy_pma_iso_link_mode	CC52h
lane2_phy_pma_iso_link_mode	CC92h
lane3_phy_pma_iso_link_mode	CCD2h

13.4.10.2.389.2 Diagram



13.4.10.2.389.3 Fields

Field	Function
15 phy_pma_iso_link_mode_15	Drives tx_reset_n PMA input for the associated lane when in PMA isolation mode.
14 phy_pma_iso_link_mode_14	Reserved
13 phy_pma_iso_link_mode_13	Reserved
12	Drives the tx_high_z PMA input for the associated lane when in PMA isolation mode

Table continues on the next page...

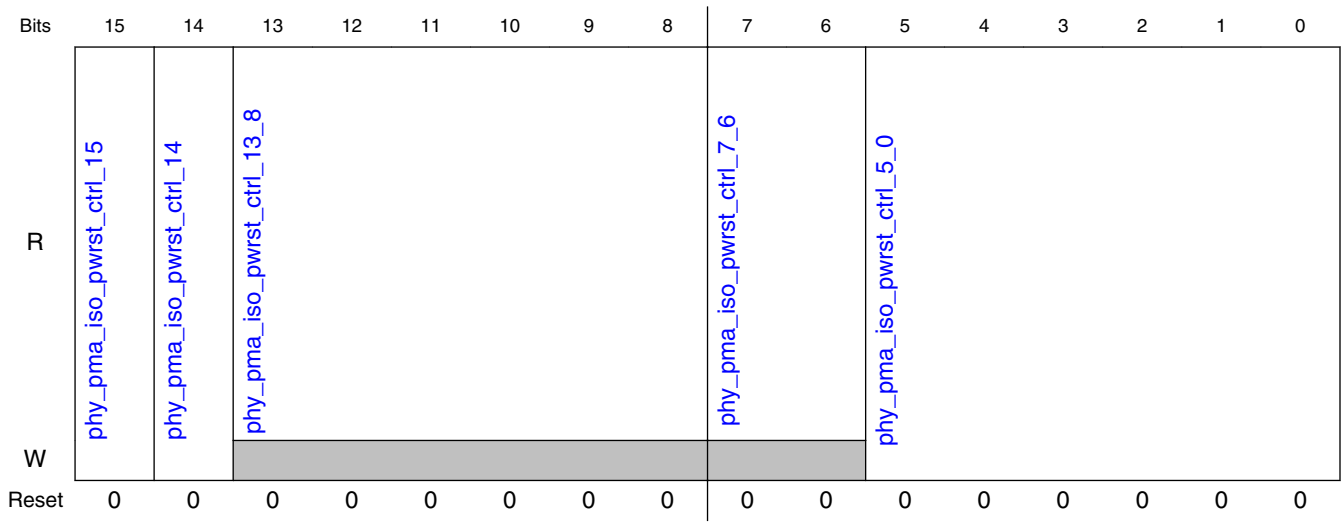
Field	Function
phy_pma_iso_link_mode_12	
11-6 phy_pma_iso_link_mode_11_6	Reserved
5-4 phy_pma_iso_link_mode_5_4	Drives xcvr_standard_mode PMA input for the associated lane when in PMA isolation mode.
3 phy_pma_iso_link_mode_3	Reserved
2-0 phy_pma_iso_link_mode_2_0	Drives xcvr_data_width PMA input for the associated lane when in PMA isolation mode.

13.4.10.2.390 PMA Isolation power state control register (lane0_phy_pma_iso_pwrst_ctrl - lane3_phy_pma_iso_pwrst_ctrl)

13.4.10.2.390.1 Offset

Register	Offset
lane0_phy_pma_iso_pwrst_ctrl	CC13h
lane1_phy_pma_iso_pwrst_ctrl	CC53h
lane2_phy_pma_iso_pwrst_ctrl	CC93h
lane3_phy_pma_iso_pwrst_ctrl	CCD3h

13.4.10.2.390.2 Diagram



13.4.10.2.390.3 Fields

Field	Function
15 phy_pma_iso_pwrst_ctrl_15	Reserved
14 phy_pma_iso_pwrst_ctrl_14	tx_cmn_mode_en_ext PMA input for the associated lane when in PMA isolation mode. (Used for PCIe)
13-8 phy_pma_iso_pwrst_ctrl_13_8	Current value of xcvr_power_state_ack PMA output for the associated lane.
7-6 phy_pma_iso_pwrst_ctrl_7_6	Reserved
5-0 phy_pma_iso_pwrst_ctrl_5_0	Drives xcvr_power_state_req PMA input for the associated lane when in PMA isolation mode.

13.4.10.2.391 PMA transmit low data isolation register (lane0_phy_pma_iso_tx_data_lo - lane3_phy_pma_iso_tx_data_lo)

13.4.10.2.391.1 Offset

Register	Offset
lane0_phy_pma_iso_tx_data_lo	CC14h
lane1_phy_pma_iso_tx_data_lo	CC54h
lane2_phy_pma_iso_tx_data_lo	CC94h
lane3_phy_pma_iso_tx_data_lo	CCD4h

13.4.10.2.391.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	phy_pma_iso_tx_data_lo_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.391.3 Fields

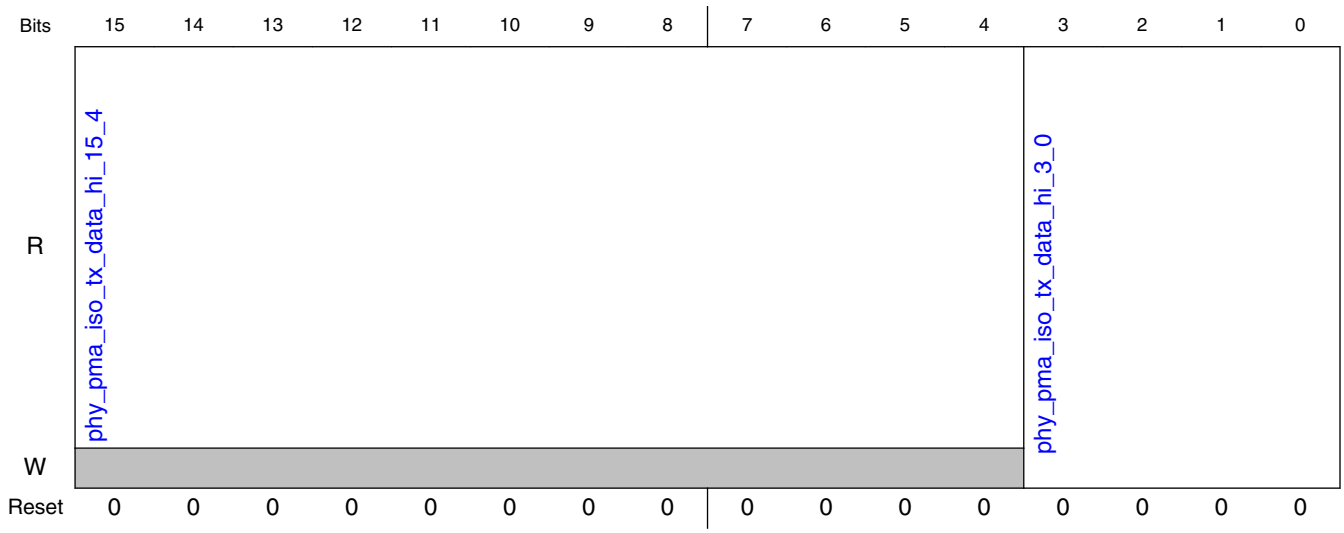
Field	Function
15-0 phy_pma_iso_tx_data_lo_15_0	Drives tx_td[15:0] PMA input for the associated lane when in PMA isolation mode. (Not re-synchronized to apb_pclk).

13.4.10.2.392 PMA transmit high data isolation register (lane0_phy_pma_iso_tx_data_hi - lane3_phy_pma_iso_tx_data_hi)

13.4.10.2.392.1 Offset

Register	Offset
lane0_phy_pma_iso_tx_data_hi	CC15h
lane1_phy_pma_iso_tx_data_hi	CC55h
lane2_phy_pma_iso_tx_data_hi	CC95h
lane3_phy_pma_iso_tx_data_hi	CCD5h

13.4.10.2.392.2 Diagram



13.4.10.2.392.3 Fields

Field	Function
15-4 phy_pma_iso_tx_data_hi_15_4	Reserved
3-0 phy_pma_iso_tx_data_hi_3_0	Drives tx_td[19:16] PMA input for the associated lane when in PMA isolation mode. (Not re-synchronized to apb_pclk).

13.4.10.2.393 Product type ID register (cmn_pid_type)

13.4.10.2.393.1 Offset

Register	Offset
cmn_pid_type	8_0000h

13.4.10.2.393.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_type_15_0															
W																
Reset	0	1	1	1	0	0	1	1	0	1	1	0	0	1	0	0

13.4.10.2.393.3 Fields

Field	Function
15-0 cmn_pid_type_15_0	Product type : This field contains the ASCII codes that represent the product type sd for SerDes.

13.4.10.2.394 Product number 1 2 ID register (cmn_pid_num12)

13.4.10.2.394.1 Offset

Register	Offset
cmn_pid_num12	8_0001h

13.4.10.2.394.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_num12_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.394.3 Fields

Field	Function
15-0 cmn_pid_num12_15_0	Product number : This field contains the ASCII codes that represent the characters 1 and 2 of the product part number.

13.4.10.2.395 Product number 3 4 ID register (cmn_pid_num34)

13.4.10.2.395.1 Offset

Register	Offset
cmn_pid_num34	8_0002h

13.4.10.2.395.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_num34_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.395.3 Fields

Field	Function
15-0 cmn_pid_num34_15_0	Product number : If necessary, this field contains the ASCII codes that represent the characters 3 and 4 of the product part number.

13.4.10.2.396 Product number 5 6 ID register (cmn_pid_num56)

13.4.10.2.396.1 Offset

Register	Offset
cmn_pid_num56	8_0003h

13.4.10.2.396.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_num56_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.396.3 Fields

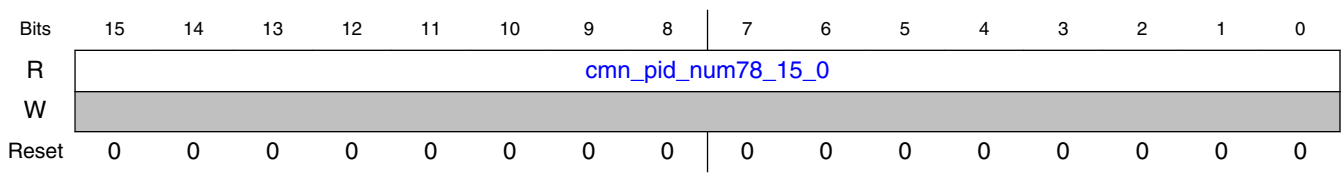
Field	Function
15-0 cmn_pid_num56_15_0	Product number : If necessary, this field contains the ASCII codes that represent the characters 5 and 6 of the product part number.

13.4.10.2.397 Product number 7 8 ID register (cmn_pid_num78)

13.4.10.2.397.1 Offset

Register	Offset
cmn_pid_num78	8_0004h

13.4.10.2.397.2 Diagram



13.4.10.2.397.3 Fields

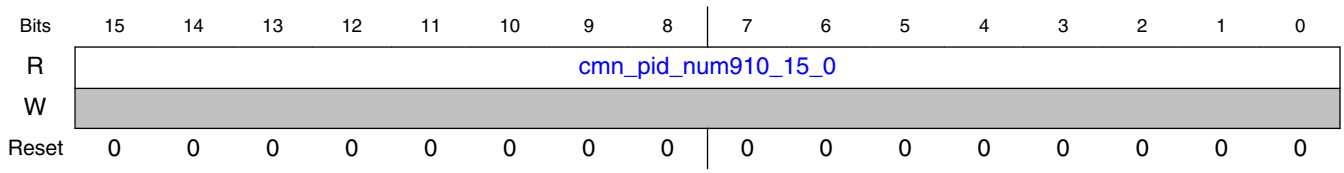
Field	Function
15-0 cmn_pid_num78_15_0	Product number : If necessary, this field contains the ASCII codes that represent the characters 7 and 8 of the product part number.

13.4.10.2.398 Product number 9 10 ID register (cmn_pid_num910)

13.4.10.2.398.1 Offset

Register	Offset
cmn_pid_num910	8_0005h

13.4.10.2.398.2 Diagram



13.4.10.2.398.3 Fields

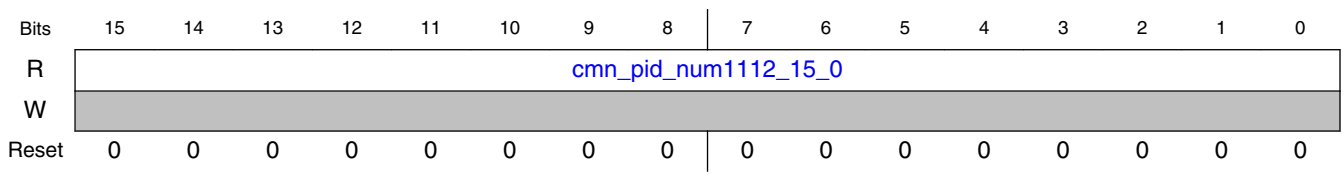
Field	Function
15-0 cmn_pid_num910_15_0	Product number : If necessary, this field contains the ASCII codes that represent the characters 9 and 10 of the product part number.

13.4.10.2.399 Product number 11 12 ID register (cmn_pid_num1112)

13.4.10.2.399.1 Offset

Register	Offset
cmn_pid_num1112	8_0006h

13.4.10.2.399.2 Diagram



13.4.10.2.399.3 Fields

Field	Function
15-0 cmn_pid_num1112_15_0	Product number : If necessary, this field contains the ASCII codes that represent the characters 11 and 12 of the product part number.

13.4.10.2.400 Product revision ID register (cmn_pid_rev)

13.4.10.2.400.1 Offset

Register	Offset
cmn_pid_rev	8_0007h

13.4.10.2.400.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_rev_15_0															
W																
Reset	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0

13.4.10.2.400.3 Fields

Field	Function
15-0 cmn_pid_rev_15_0	Product revision : This field contains the binary coded decimal numbers that represent the product revision 0100.

13.4.10.2.401 Product technology manufacturer ID register (cmn_pid_mfg)

13.4.10.2.401.1 Offset

Register	Offset
cmn_pid_mfg	8_0008h

13.4.10.2.401.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_mfg_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	1	1	1	0	1	0	0

13.4.10.2.401.3 Fields

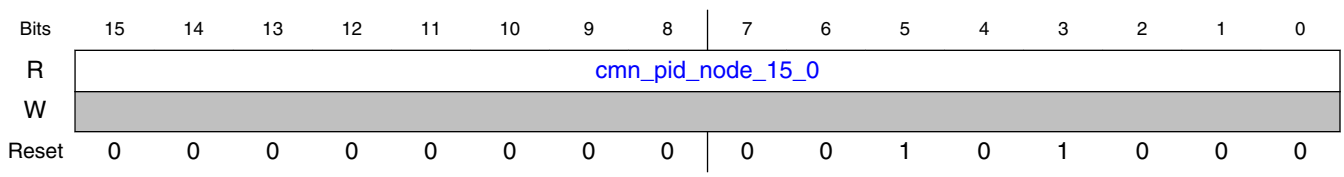
Field	Function
15-0 cmn_pid_mfg_15_0	Product technology manufacturer : This field contains the ASCII codes that represent the product technology manufacturer.

13.4.10.2.402 Product technology process node ID register (cmn_pid_node)

13.4.10.2.402.1 Offset

Register	Offset
cmn_pid_node	8_0009h

13.4.10.2.402.2 Diagram



13.4.10.2.402.3 Fields

Field	Function
15-0 cmn_pid_node_15_0	Product technology process node : This field contains the binary coded decimal numbers that represent the product technology node 28nm

13.4.10.2.403 Product technology process flavor ID register 0 (cmn_pid_flv0)

13.4.10.2.403.1 Offset

Register	Offset
cmn_pid_flv0	8_000Ah

13.4.10.2.403.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_flv0_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.403.3 Fields

Field	Function
15-0 cmn_pid_flv0_15_0	Product technology flavor : This field contains the ASCII codes that represent the first two characters of the product technology flavor.

13.4.10.2.404 Product technology process flavor ID register 1 (cmn_pid_flv1)

13.4.10.2.404.1 Offset

Register	Offset
cmn_pid_flv1	8_000Bh

13.4.10.2.404.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_flv1_15_0															
W																
Reset	0	1	1	1	0	1	0	0	0	0	0	0	0	0	0	0

13.4.10.2.404.3 Fields

Field	Function
15-0 cmn_pid_flv1_15_0	Product technology flavor : This field contains the ASCII codes that represent the second two characters of the product technology flavor (when applicable).

13.4.10.2.405 Product I/O voltage ID register (cmn_pid_iov)

13.4.10.2.405.1 Offset

Register	Offset
cmn_pid_iov	8_000Ch

13.4.10.2.405.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_iov_15_0															
W																
Reset	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0

13.4.10.2.405.3 Fields

Field	Function
15-0 cmn_pid_iov_15_0	Product I/O voltage : This field contains the binary coded decimal numbers that represent the product I/O voltage. The most significant byte represents the value to the left of the decimal point, and the least significant byte represents the value to the right of the decimal point. For example, 1.5V is represented as 0x0150.

13.4.10.2.406 Product SerDes lanes ID register (cmn_pid_lanes)

13.4.10.2.406.1 Offset

Register	Offset
cmn_pid_lanes	8_000Dh

13.4.10.2.406.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_lanes_15_8								cmn_pid_lanes_7_0							
W																
Reset	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0

13.4.10.2.406.3 Fields

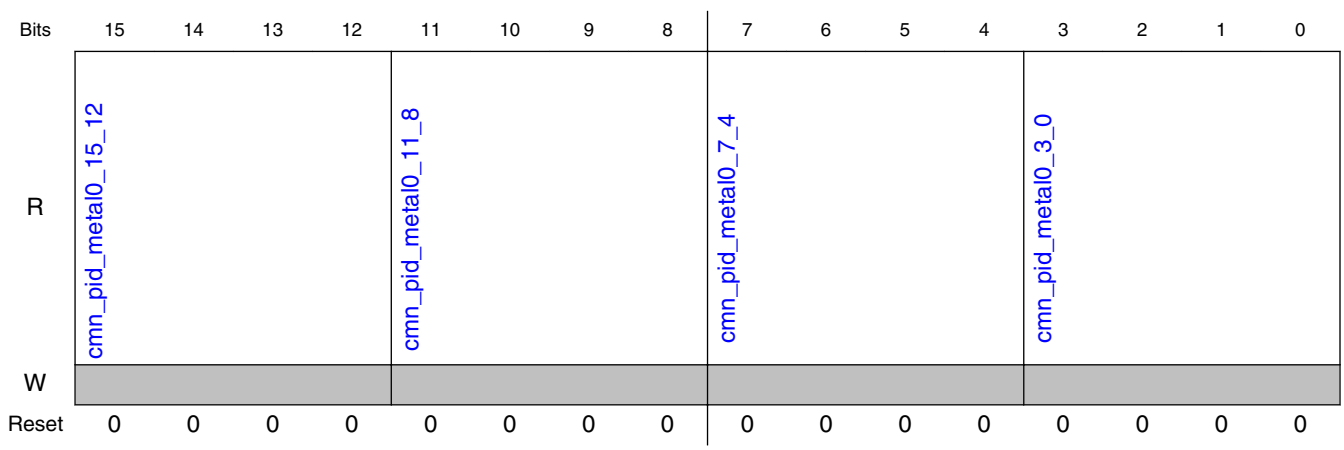
Field	Function
15-8 cmn_pid_lanes_15_8	Product SerDes lanes left of common : This field contains the binary coded decimal numbers that represent the number of lanes implemented in this SerDes product on the left side of the common module.
7-0 cmn_pid_lanes_7_0	Product SerDes lanes right of common : This field contains the binary coded decimal numbers that represent the number of lanes implemented in this SerDes product on the right side of the common module.

13.4.10.2.407 Product metal layers ID register 0 (cmn_pid_metal0)

13.4.10.2.407.1 Offset

Register	Offset
cmn_pid_metal0	8_0010h

13.4.10.2.407.2 Diagram



13.4.10.2.407.3 Fields

Field	Function
15-12 cmn_pid_metal0_15_12	Product X metal layers : This field contains the binary coded decimal number that represent the number of xd metal layers used for this product.

Table continues on the next page...

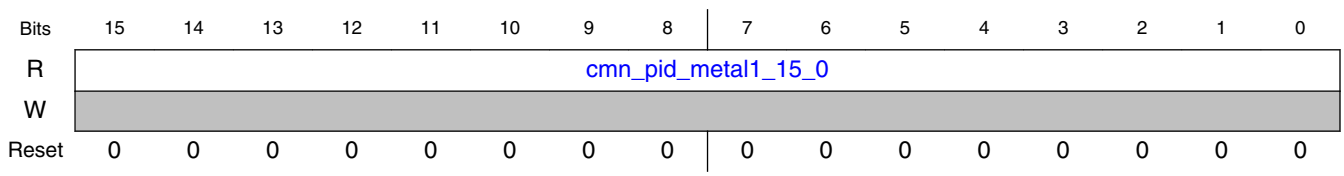
Field	Function
11-8 cmn_pid_metal0_11_8	Product Y metal layers : This field contains the binary coded decimal number that represent the number of xc metal layers used for this product.
7-4 cmn_pid_metal0_7_4	Product Z metal layers : This field contains the binary coded decimal number that represent the number of xa metal layers used for this product.
3-0 cmn_pid_metal0_3_0	Product R/U metal layers : This field contains the binary coded decimal number that represent the number of x metal layers used for this product.

13.4.10.2.408 Product metal layers ID register 1 (cmn_pid_metal1)

13.4.10.2.408.1 Offset

Register	Offset
cmn_pid_metal1	8_0011h

13.4.10.2.408.2 Diagram



13.4.10.2.408.3 Fields

Field	Function
15-0 cmn_pid_metal1_15_0	Reserved

13.4.10.2.409 Product metal layers ID register 2 (cmn_pid_metal2)

13.4.10.2.409.1 Offset

Register	Offset
cmn_pid_metal2	8_0012h

13.4.10.2.409.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_metal2_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.409.3 Fields

Field	Function
15-0 cmn_pid_metal2_15_0	Reserved

13.4.10.2.410 Product metal layers ID register 3 (cmn_pid_metal3)

13.4.10.2.410.1 Offset

Register	Offset
cmn_pid_metal3	8_0013h

13.4.10.2.410.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pid_metal3_15_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.410.3 Fields

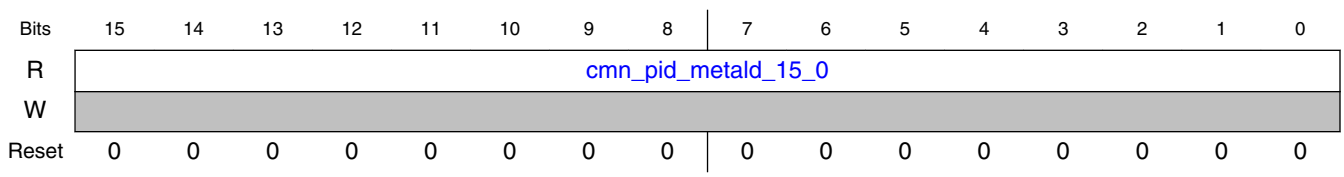
Field	Function
15-0 cmn_pid_metal3_15_0	Reserved

13.4.10.2.411 Product metal layer direction ID register (cmn_pid_metald)

13.4.10.2.411.1 Offset

Register	Offset
cmn_pid_metald	8_0014h

13.4.10.2.411.2 Diagram



13.4.10.2.411.3 Fields

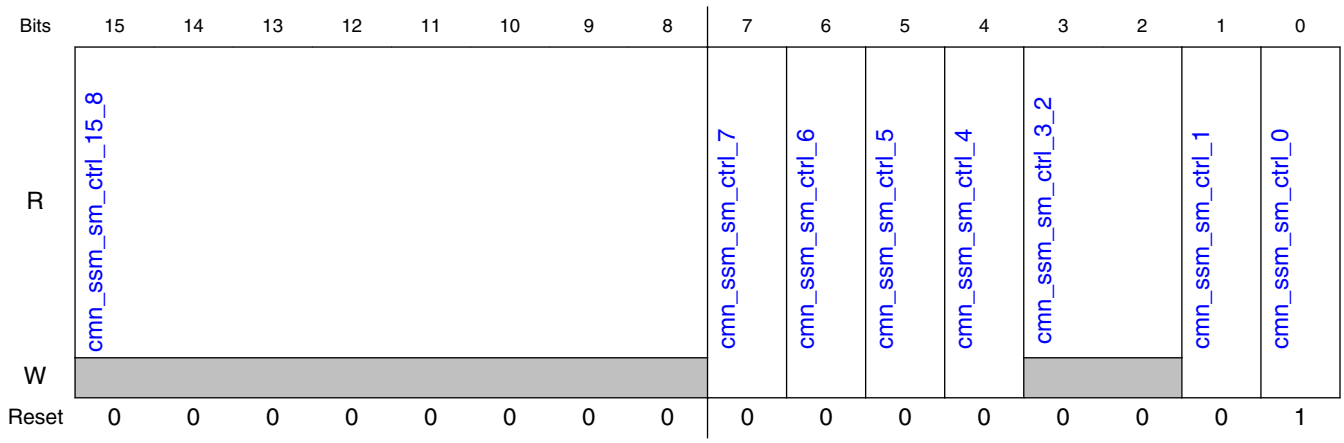
Field	Function
15-0 cmn_pid_metald_15_0	Reserved

13.4.10.2.412 Startup state machine control register (cmn_ssm_sm_ctrl)

13.4.10.2.412.1 Offset

Register	Offset
cmn_ssm_sm_ctrl	8_0020h

13.4.10.2.412.2 Diagram



13.4.10.2.412.3 Fields

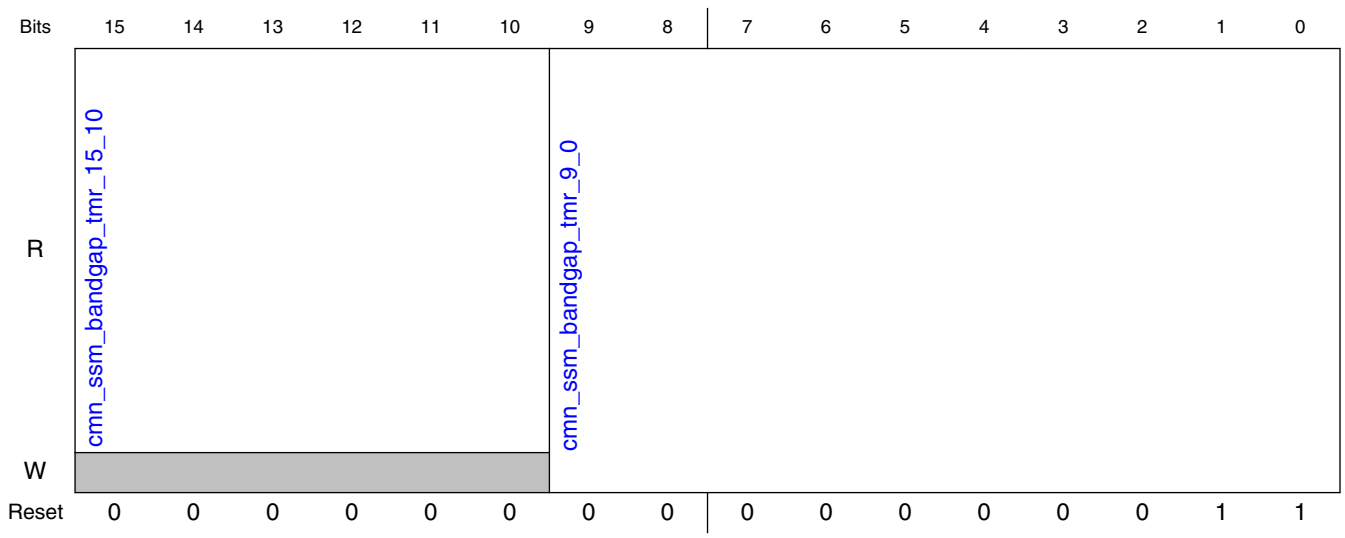
Field	Function
15-8 cmn_ssm_sm_ctrl_15_8	Reserved
7 cmn_ssm_sm_ctrl_7	Bandgap enable override enable : When active (1b1), the bandgap enable override bit in this register will drive the
6 cmn_ssm_sm_ctrl_6	Bandgap enable override : When enabled by the bandgap enable override enable bit in this register, this bit will drive the
5 cmn_ssm_sm_ctrl_5	Bias enable override enable : When active (1b1), the bias enable override bit in this register will drive the
4 cmn_ssm_sm_ctrl_4	Bias enable override : When enabled by the bias enable override enable bit in this register, this bit will drive the
3-2 cmn_ssm_sm_ctrl_3_2	Reserved
1 cmn_ssm_sm_ctrl_1	Skip post bandgap enable re-calibration : When this bit is active (1b1), the post bandgap enable calibration state will be skipped if it was previously run, unless the macro is disabled or reset.
0 cmn_ssm_sm_ctrl_0	Skip auto re-calibration : When this bit is active (1b1), the auto calibration state will be skipped if it was previously run, unless the macro is disabled or reset.

13.4.10.2.413 Bandgap enable timer register (cmn_ssm_bandgap_tmr)

13.4.10.2.413.1 Offset

Register	Offset
cmn_ssm_bandgap_tmr	8_0021h

13.4.10.2.413.2 Diagram



13.4.10.2.413.3 Fields

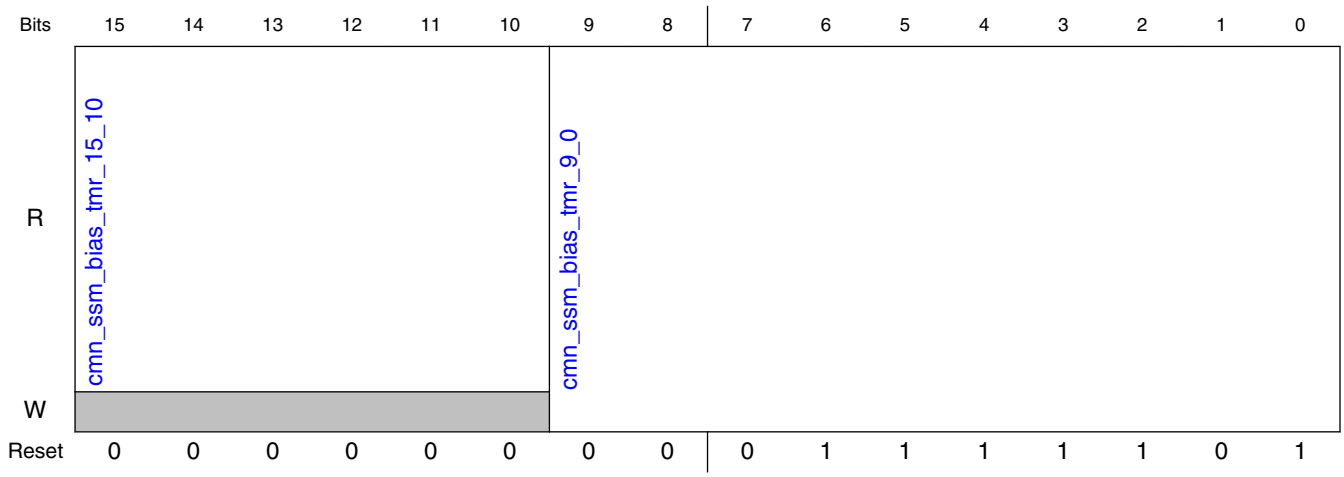
Field	Function
15-10 cmn_ssm_bandgap_tmr_15_10	Reserved
9-0 cmn_ssm_bandgap_tmr_9_0	Bandgap enable state timer value : Value used for the timer when the startup state machine is in the bandgap enable state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.414 Bias enable timer register (cmn_ssm_bias_tmr)

13.4.10.2.414.1 Offset

Register	Offset
cmn_ssm_bias_tmr	8_0022h

13.4.10.2.414.2 Diagram



13.4.10.2.414.3 Fields

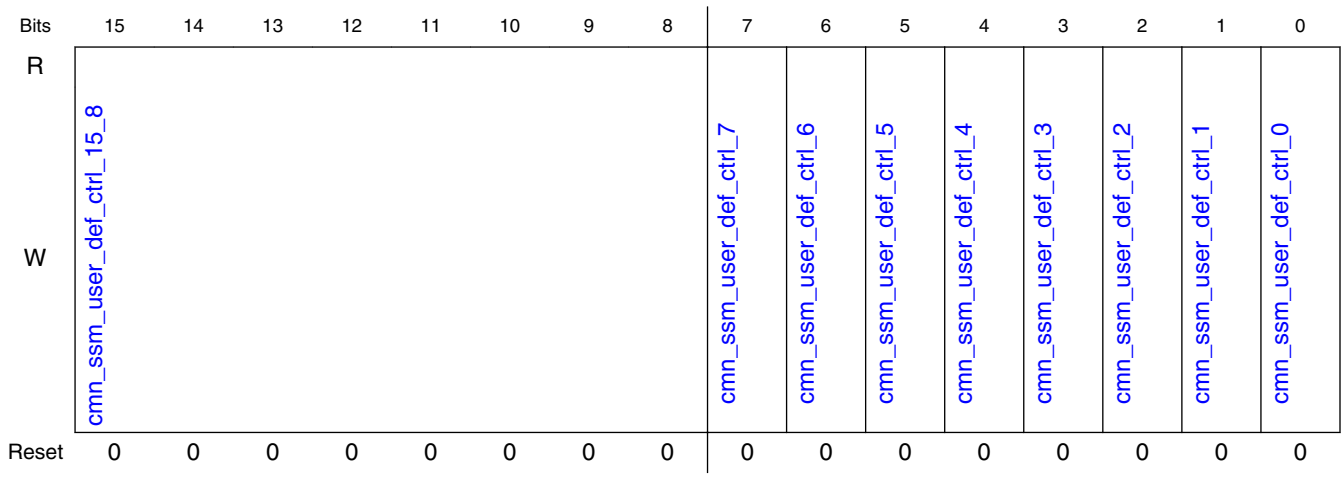
Field	Function
15-10 cmn_ssm_bias_tmr_15_10	Reserved
9-0 cmn_ssm_bias_tmr_9_0	Bias enable state timer value : Value used for the timer when the startup state machine is in the bias enable state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.415 Startup state machine user defined control register (cmn_ssm_user_def_ctrl)

13.4.10.2.415.1 Offset

Register	Offset
cmn_ssm_user_def_ctrl	8_0027h

13.4.10.2.415.2 Diagram



13.4.10.2.415.3 Fields

Field	Function
15-8 cmn_ssm_user_def_ctrl_15_8	Reserved
7 cmn_ssm_user_def_ctrl_7	Analog reference clock enable override enable : When active (1b1), the analog reference clock enable override bit in this register will drive the
6 cmn_ssm_user_def_ctrl_6	Analog reference clock enable override : When enabled by the analog reference clock enable override enable bit in this register, this bit will drive the
5 cmn_ssm_user_def_ctrl_5	Calibration iconst enable override enable : When active (1b1), the calibration iconst enable override bit in this register will drive the
4 cmn_ssm_user_def_ctrl_4	Calibration iconst enable override : When enabled by the calibration power enable override enable bit in this register, this bit will drive the
3 cmn_ssm_user_def_ctrl_3	Calibration power enable override enable : When active (1b1), the calibration iconst enable override bit in this register will drive the
2 cmn_ssm_user_def_ctrl_2	Calibration power enable override : When enabled by the calibration power enable override enable bit in this register, this bit will drive the
1 cmn_ssm_user_def_ctrl_1	Bandgap enable override enable : When active (1b1), the bandgap enable override bit in this register will drive the
0	Bandgap enable override : When enabled by the bandgap enable override enable bit in this register, this bit will drive the

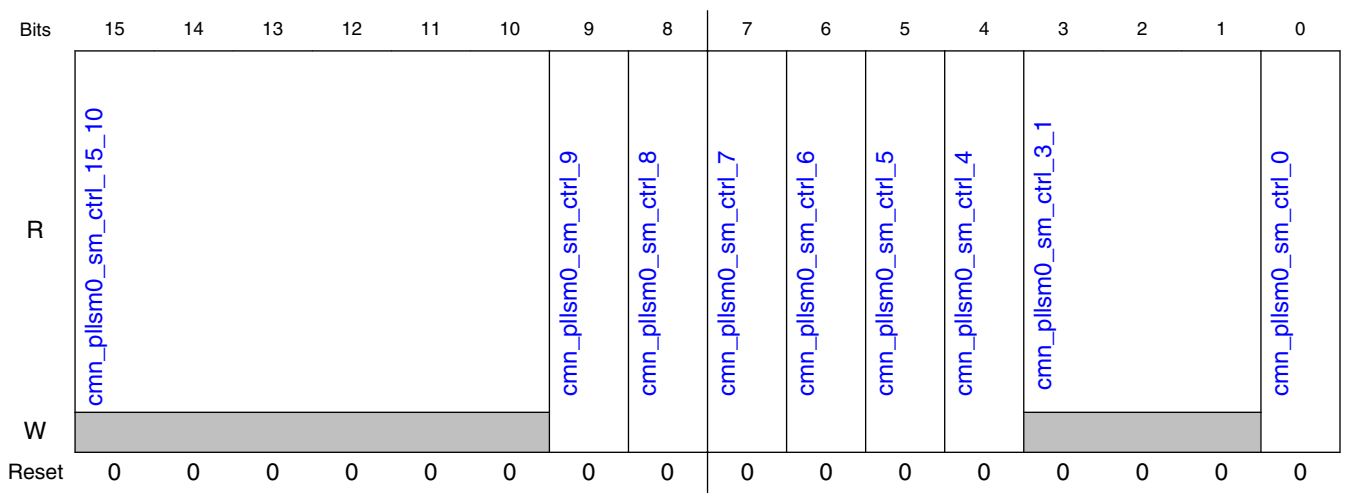
Field	Function
cmn_ssm_user_def_ctrl_0	

13.4.10.2.416 PLL 0 control state machine control register (cmn_pllsm0_sm_ctrl)

13.4.10.2.416.1 Offset

Register	Offset
cmn_pllsm0_sm_ctrl	8_0028h

13.4.10.2.416.2 Diagram



13.4.10.2.416.3 Fields

Field	Function
15-10 cmn_pllsm0_sm_ctrl_15_10	Reserved
9 cmn_pllsm0_sm_ctrl_9	PLL enable override enable : When active (1b1), the PLL enable override bit in this register will drive the
8	PLL enable override : When enabled by the PLL enable override enable bit in this register, this bit will drive the

Table continues on the next page...

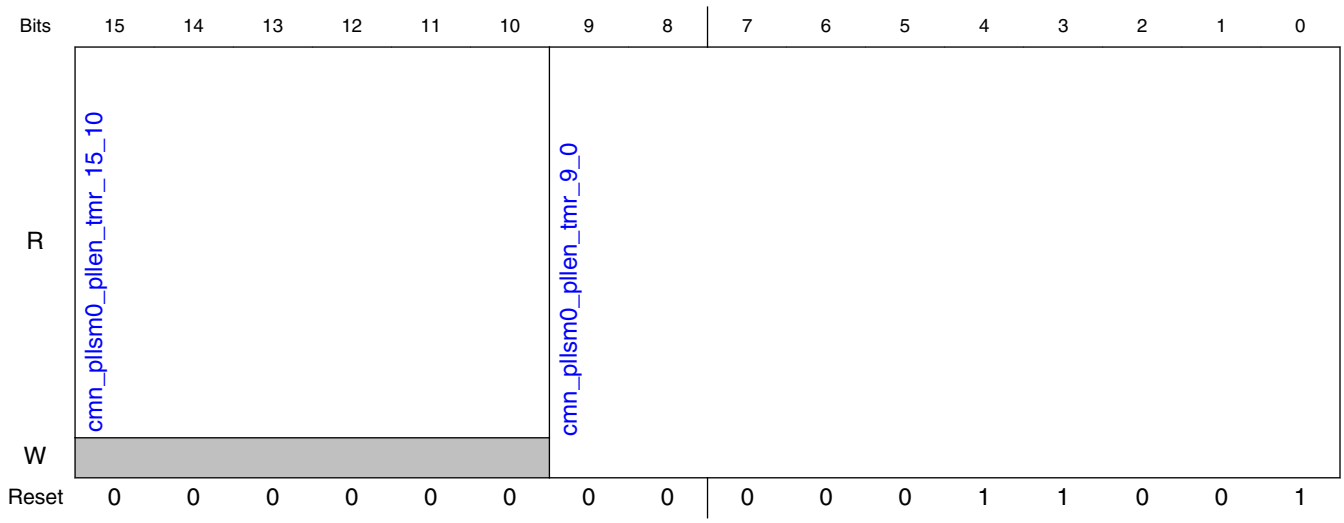
Field	Function
cmn_pllsm0_sm_ctrl_8	
7 cmn_pllsm0_sm_ctrl_7	PLL VCO LDO reference override enable : When active (1b1), the PLL VCO LDO reference override bit in this register will drive the
6 cmn_pllsm0_sm_ctrl_6	PLL VCO LDO reference override : When enabled by the PLL PLL VCO LDO reference override enable bit in this register, this bit will drive the
5 cmn_pllsm0_sm_ctrl_5	PLL VCO LDO reference charge pulse override enable : When active (1b1), the PLL VCO LDO reference charge pulse override bit in this register will drive the
4 cmn_pllsm0_sm_ctrl_4	PLL VCO LDO reference charge pulse override : When enabled by the PLL VCO LDO reference charge pulse override enable bit in this register, this bit will drive the
3-1 cmn_pllsm0_sm_ctrl_3_1	Reserved
0 cmn_pllsm0_sm_ctrl_0	Skip PLL re-calibration : When this bit is active (1b1), the PLL calibration state will be skipped if it was previously run, unless the PLL is disabled or resetting the state machine.

13.4.10.2.417 PLL 0 enable timer register (cmn_pllsm0_pllen_tmr)

13.4.10.2.417.1 Offset

Register	Offset
cmn_pllsm0_pllen_tmr	8_0029h

13.4.10.2.417.2 Diagram



13.4.10.2.417.3 Fields

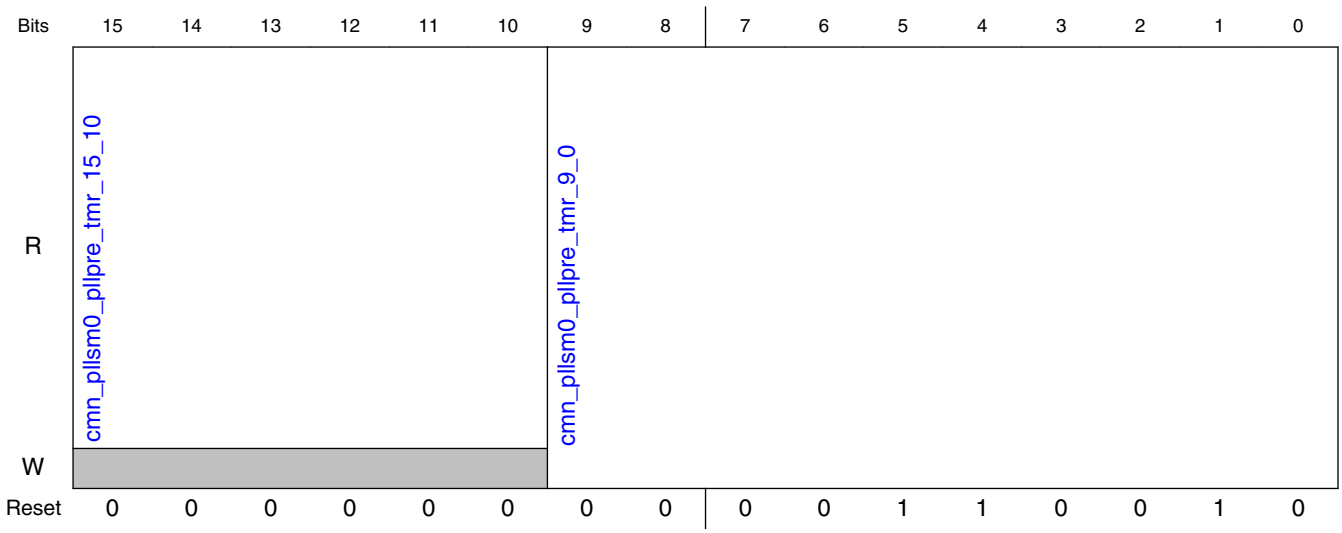
Field	Function
15-10 cmn_pllsm0_pllen_tmr_15_10	Reserved
9-0 cmn_pllsm0_pllen_tmr_9_0	PLL enable state timer value : Value used for the timer when the startup state machine is in the PLL enable state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.418 PLL 0 pre-charge timer register (cmn_pllsm0_pllpre_tmr)

13.4.10.2.418.1 Offset

Register	Offset
cmn_pllsm0_pllpre_tmr	8_002Ah

13.4.10.2.418.2 Diagram



13.4.10.2.418.3 Fields

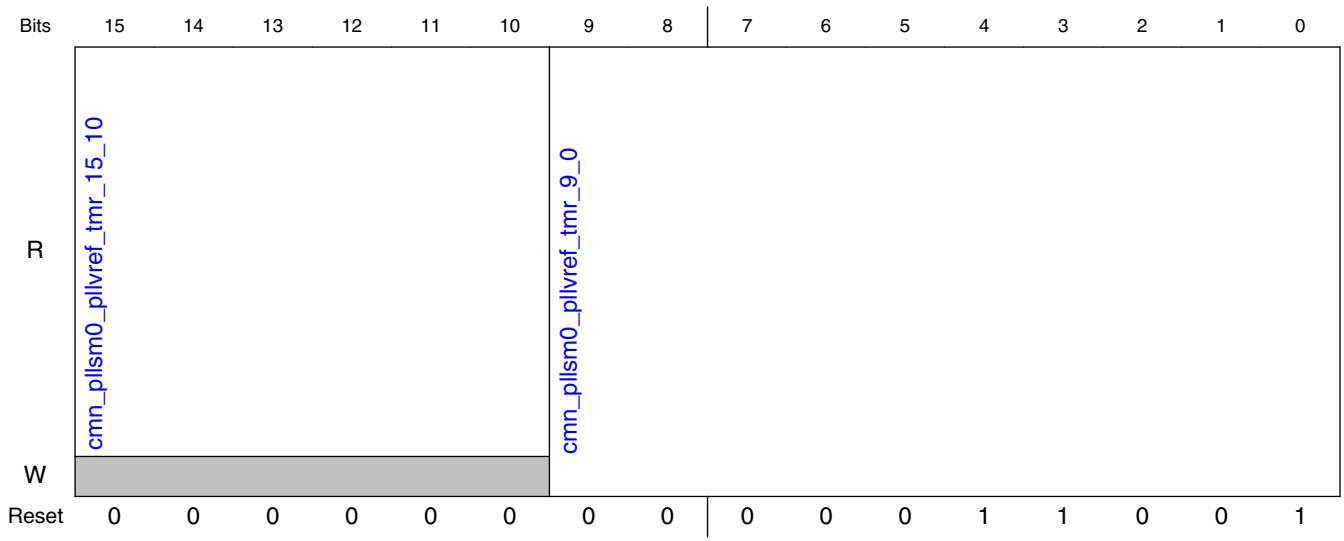
Field	Function
15-10 cmn_pllsm0_pllpre_tmr_15_10	Reserved
9-0 cmn_pllsm0_pllpre_tmr_9_0	PLL pre-charge state timer value : Value used for the timer when the startup state machine is in the PLL pre-charge state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.419 PLL 0 VREF delay timer register (cmn_pllsm0_pllvref_tmr)

13.4.10.2.419.1 Offset

Register	Offset
cmn_pllsm0_pllvref_tmr	8_002Bh

13.4.10.2.419.2 Diagram



13.4.10.2.419.3 Fields

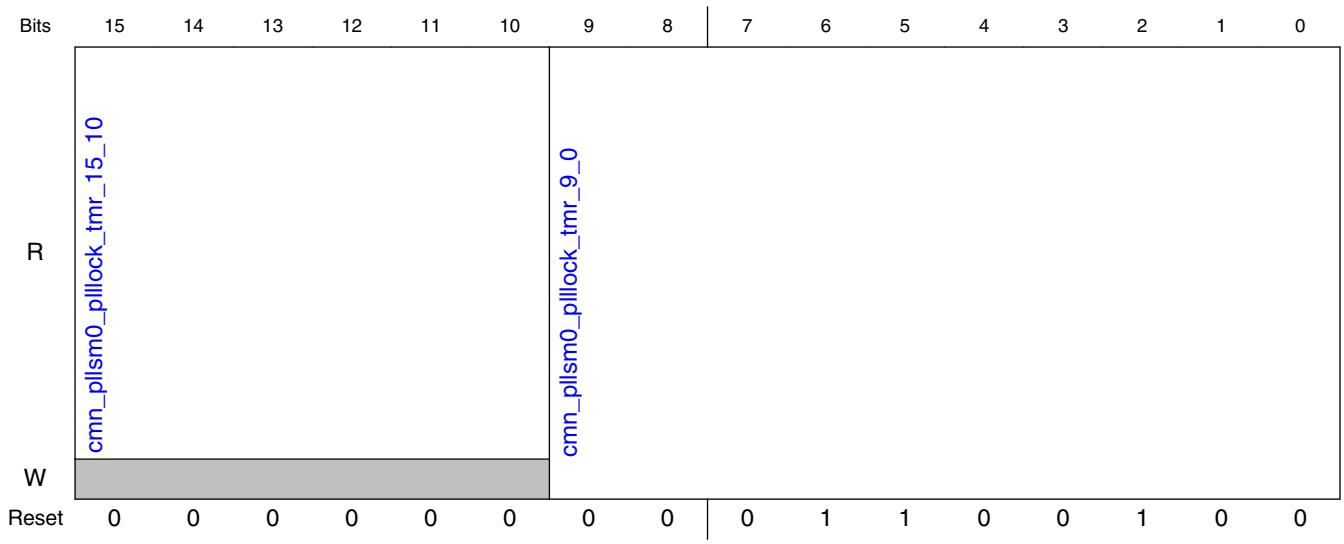
Field	Function
15-10 cmn_pllsm0_pllvref_tmr_15_10	Reserved
9-0 cmn_pllsm0_pllvref_tmr_9_0	PLL VREF delay state timer value : Value used for the timer when the startup state machine is in the PLL VREF delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.420 PLL 0 lock delay timer register (cmn_pllsm0_plllock_tmr)

13.4.10.2.420.1 Offset

Register	Offset
cmn_pllsm0_plllock_tmr	8_002Ch

13.4.10.2.420.2 Diagram



13.4.10.2.420.3 Fields

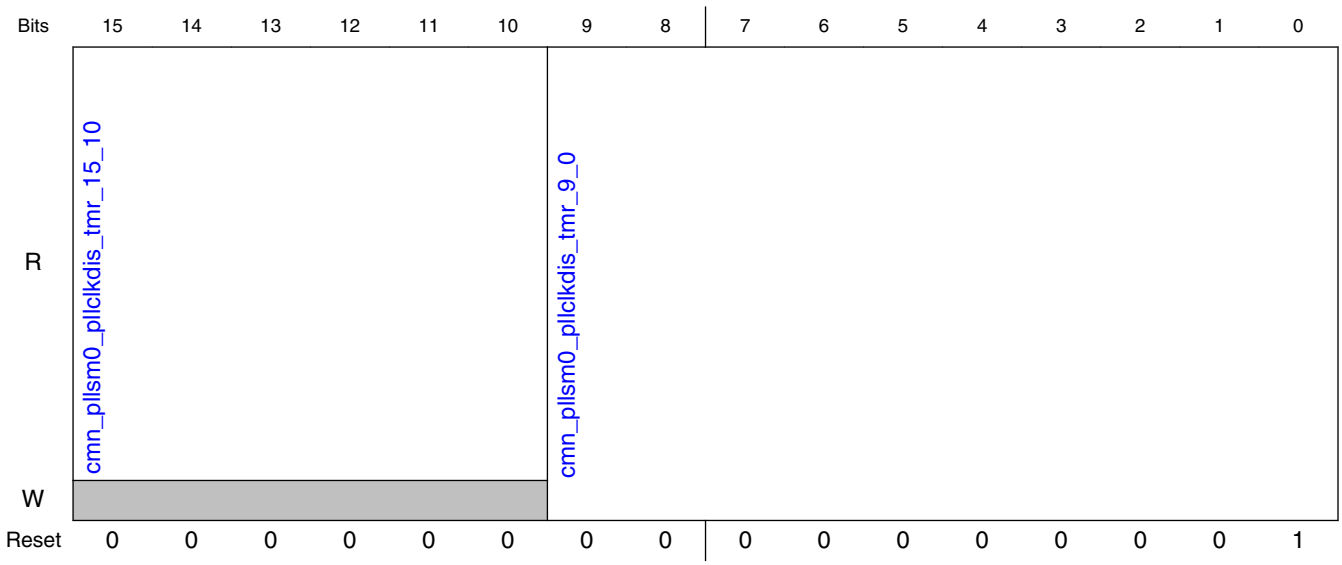
Field	Function
15-10 cmn_pllsm0_pllock_tmr_15_10	Reserved
9-0 cmn_pllsm0_pllock_tmr_9_0	PLL lock delay state timer value : Value used for the timer when the startup state machine is in the PLL lock delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.421 PLL 0 clock disable delay timer register (cmn_pllsm0_pllclkdis_tmr)

13.4.10.2.421.1 Offset

Register	Offset
cmn_pllsm0_pllclkdis_tmr	8_002Dh

13.4.10.2.421.2 Diagram



13.4.10.2.421.3 Fields

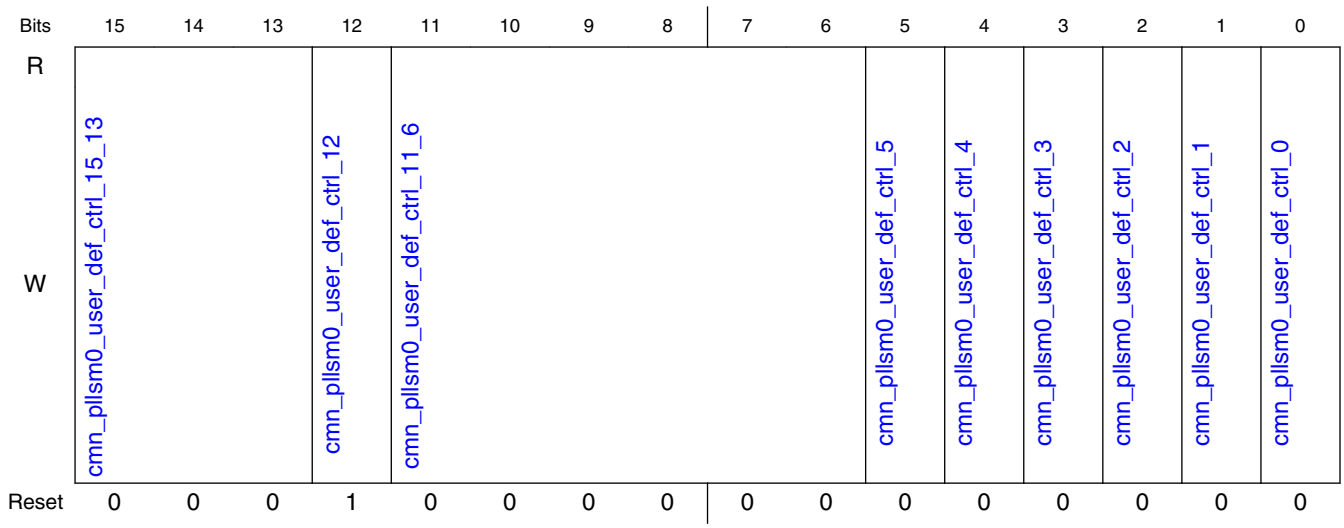
Field	Function
15-10 cmn_pllsm0_pllclkdis_tmr_15_10	Reserved
9-0 cmn_pllsm0_pllclkdis_tmr_9_0	PLL clock disable delay state timer value : Value used for the timer when the startup state machine is in the PLL clock disable delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.422 PLL 0 control state machine user defined control register (cmn_pllsm0_user_def_ctrl)

13.4.10.2.422.1 Offset

Register	Offset
cmn_pllsm0_user_def_ctrl	8_002Fh

13.4.10.2.422.2 Diagram



13.4.10.2.422.3 Fields

Field	Function
15-13 cmn_pllsm0_user_def_ctrl_15_13	Reserved
12 cmn_pllsm0_user_def_ctrl_12	PLL oscillator ring select: Selects which oscillator ring is enabled.
11-6 cmn_pllsm0_user_def_ctrl_11_6	Reserved
5 cmn_pllsm0_user_def_ctrl_5	PLL bias enable override enable : When active (1b1), the PLL bias enable override bit in this register will drive the
4 cmn_pllsm0_user_def_ctrl_4	PLL bias enable override : When enabled by the PLL bias enable override enable bit in this register, this bit will drive the
3 cmn_pllsm0_user_def_ctrl_3	PLL 0 start loop override enable : When active (1b1), the PLL 0 start loop override bit in this register will drive the
2 cmn_pllsm0_user_def_ctrl_2	PLL 0 start loop enable override : When enabled by the PLL 0 start loop override enable bit in this register, this bit will drive the
1	PLL 0 VCO LDO enable override enable : When active (1b1), the PLL 0 VCO LDO enable override bit in this register will drive the

Table continues on the next page...

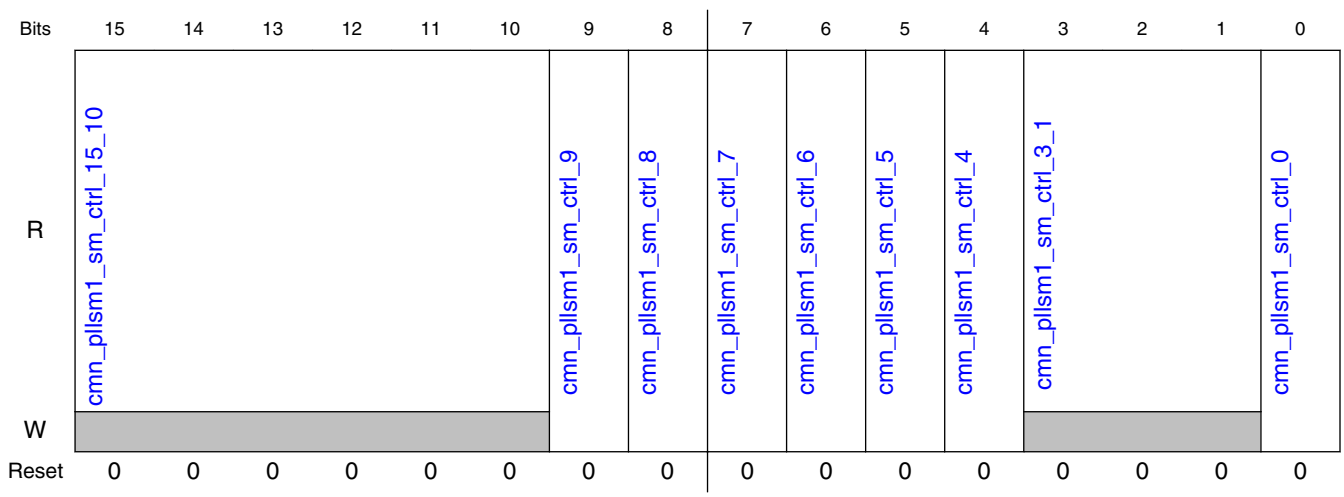
Field	Function
cmn_pllsm0_user_def_ctrl_1	
0 cmn_pllsm0_user_def_ctrl_0	PLL 0 VCO LDO enable override : When enabled by the PLL 0 VCO LDO enable override enable bit in this register, this bit will drive the

13.4.10.2.423 PLL 1 control state machine control register (cmn_pllsm1_sm_ctrl)

13.4.10.2.423.1 Offset

Register	Offset
cmn_pllsm1_sm_ctrl	8_0030h

13.4.10.2.423.2 Diagram



13.4.10.2.423.3 Fields

Field	Function
15-10 cmn_pllsm1_sm_ctrl_15_10	Reserved
9	PLL enable override enable : When active (1b1), the PLL enable override bit in this register will drive the

Table continues on the next page...

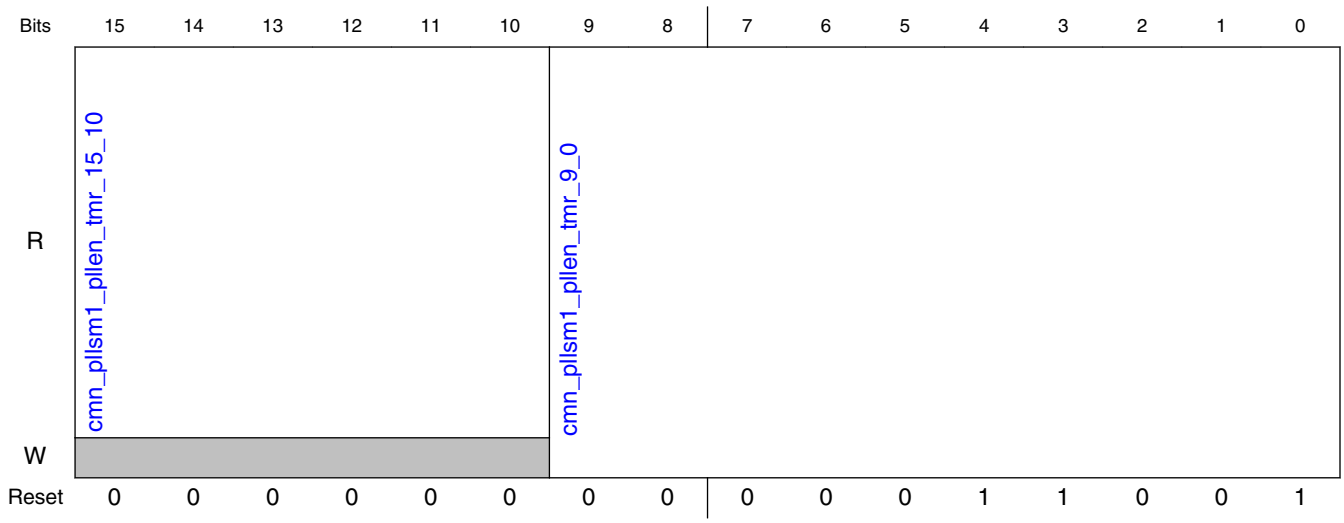
Field	Function
cmn_pllsm1_sm_ctrl_9	
8 cmn_pllsm1_sm_ctrl_8	PLL enable override : When enabled by the PLL enable override enable bit in this register, this bit will drive the
7 cmn_pllsm1_sm_ctrl_7	PLL VCO LDO reference override enable : When active (1b1), the PLL VCO LDO reference override bit in this register will drive the
6 cmn_pllsm1_sm_ctrl_6	PLL VCO LDO reference override : When enabled by the PLL PLL VCO LDO reference override enable bit in this register, this bit will drive the
5 cmn_pllsm1_sm_ctrl_5	PLL VCO LDO reference charge pulse override enable : When active (1b1), the PLL VCO LDO reference charge pulse override bit in this register will drive the
4 cmn_pllsm1_sm_ctrl_4	PLL VCO LDO reference charge pulse override : When enabled by the PLL VCO LDO reference charge pulse override enable bit in this register, this bit will drive the
3-1 cmn_pllsm1_sm_ctrl_3_1	Reserved
0 cmn_pllsm1_sm_ctrl_0	Skip PLL re-calibration : When this bit is active (1b1), the PLL calibration state will be skipped if it was previously run, unless the PLL is disabled or resetting the state machine.

13.4.10.2.424 PLL 1 enable timer register (cmn_pllsm1_pllen_tmr)

13.4.10.2.424.1 Offset

Register	Offset
cmn_pllsm1_pllen_tmr	8_0031h

13.4.10.2.424.2 Diagram



13.4.10.2.424.3 Fields

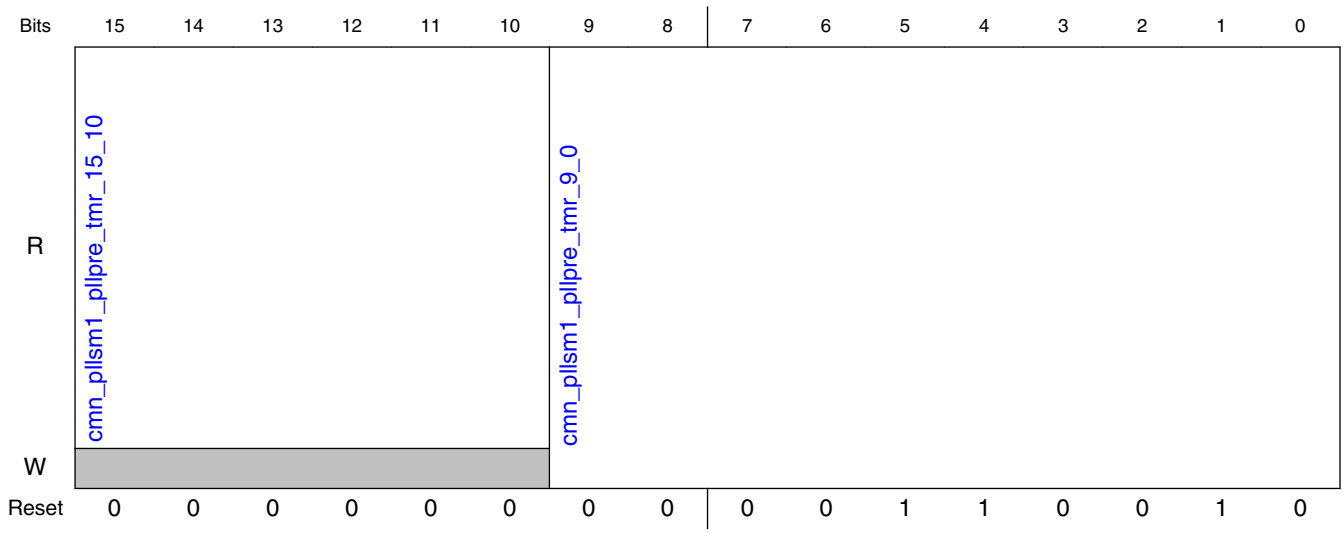
Field	Function
15-10 cmn_pllsm1_pllen_tmr_15_10	Reserved
9-0 cmn_pllsm1_pllen_tmr_9_0	PLL enable state timer value : Value used for the timer when the startup state machine is in the PLL enable state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.425 PLL 1 pre-charge timer register (cmn_pllsm1_pllpre_tmr)

13.4.10.2.425.1 Offset

Register	Offset
cmn_pllsm1_pllpre_tmr	8_0032h

13.4.10.2.425.2 Diagram



13.4.10.2.425.3 Fields

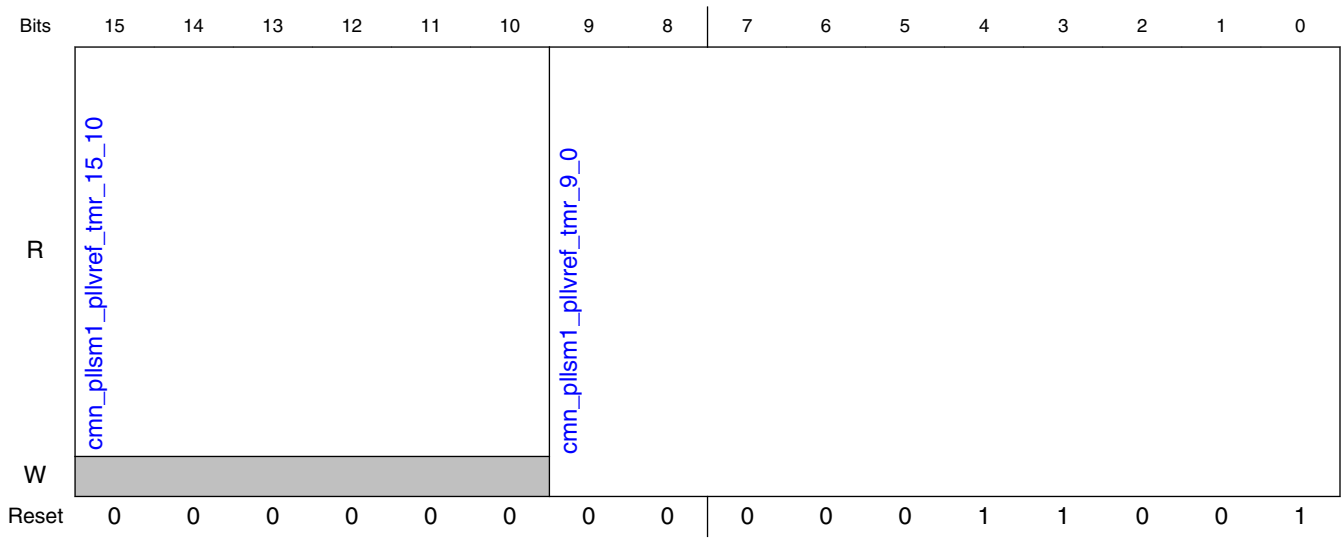
Field	Function
15-10 cmn_pllsm1_pllpre_tmr_15_10	Reserved
9-0 cmn_pllsm1_pllpre_tmr_9_0	PLL pre-charge state timer value : Value used for the timer when the startup state machine is in the PLL pre-charge state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.426 PLL 1 VREF delay timer register (cmn_pllsm1_pllvref_tmr)

13.4.10.2.426.1 Offset

Register	Offset
cmn_pllsm1_pllvref_tmr	8_0033h

13.4.10.2.426.2 Diagram



13.4.10.2.426.3 Fields

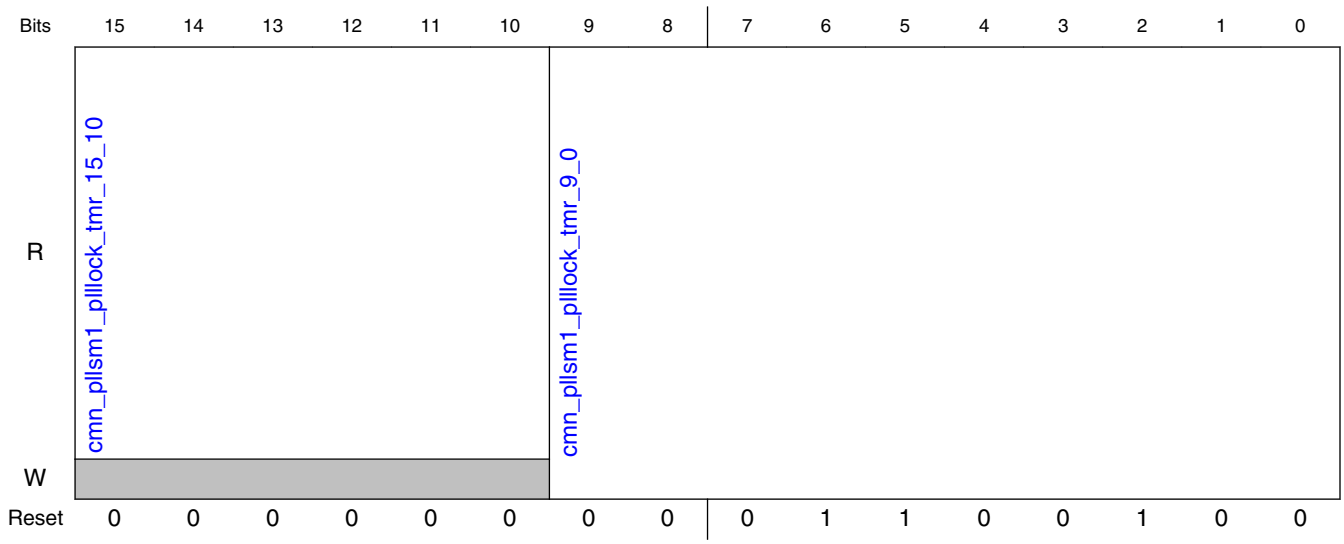
Field	Function
15-10 cmn_pllsm1_pllvref_tmr_15_10	Reserved
9-0 cmn_pllsm1_pllvref_tmr_9_0	PLL VREF delay state timer value : Value used for the timer when the startup state machine is in the PLL VREF delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.427 PLL 1 lock delay timer register (cmn_pllsm1_plllock_tmr)

13.4.10.2.427.1 Offset

Register	Offset
cmn_pllsm1_plllock_tmr	8_0034h

13.4.10.2.427.2 Diagram



13.4.10.2.427.3 Fields

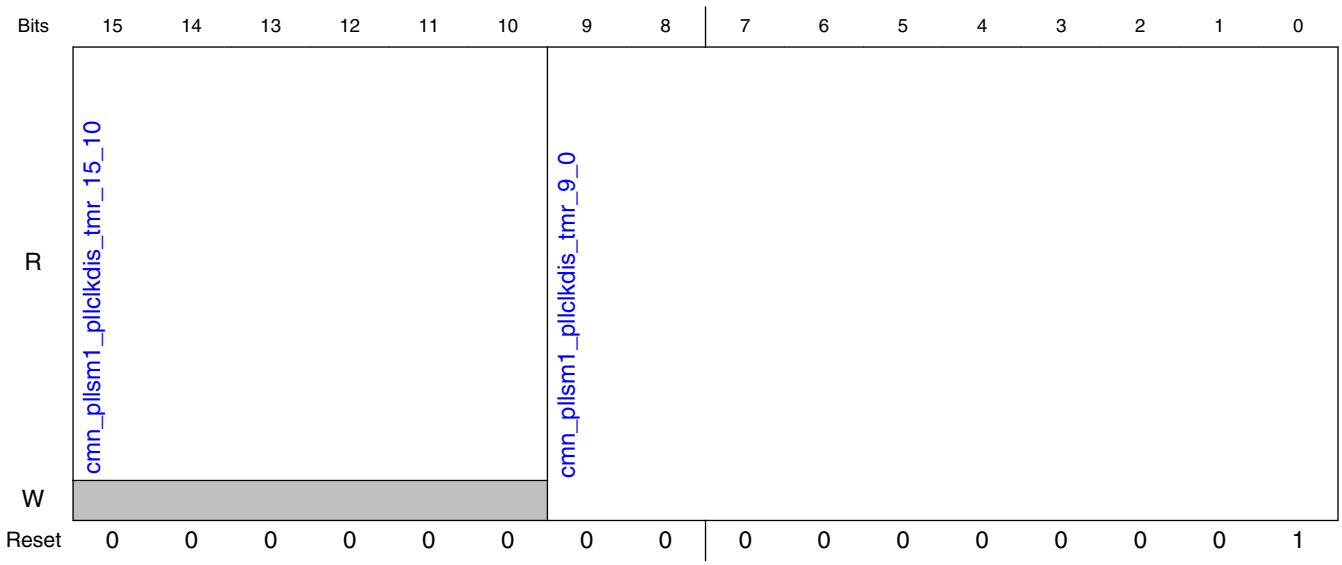
Field	Function
15-10 cmn_pllsm1_pllock_tmr_15_10	Reserved
9-0 cmn_pllsm1_pllock_tmr_9_0	PLL lock delay state timer value : Value used for the timer when the startup state machine is in the PLL lock delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.428 PLL 1 clock disable delay timer register (cmn_pllsm1_pllclkdis_tmr)

13.4.10.2.428.1 Offset

Register	Offset
cmn_pllsm1_pllclkdis_tmr	8_0035h

13.4.10.2.428.2 Diagram



13.4.10.2.428.3 Fields

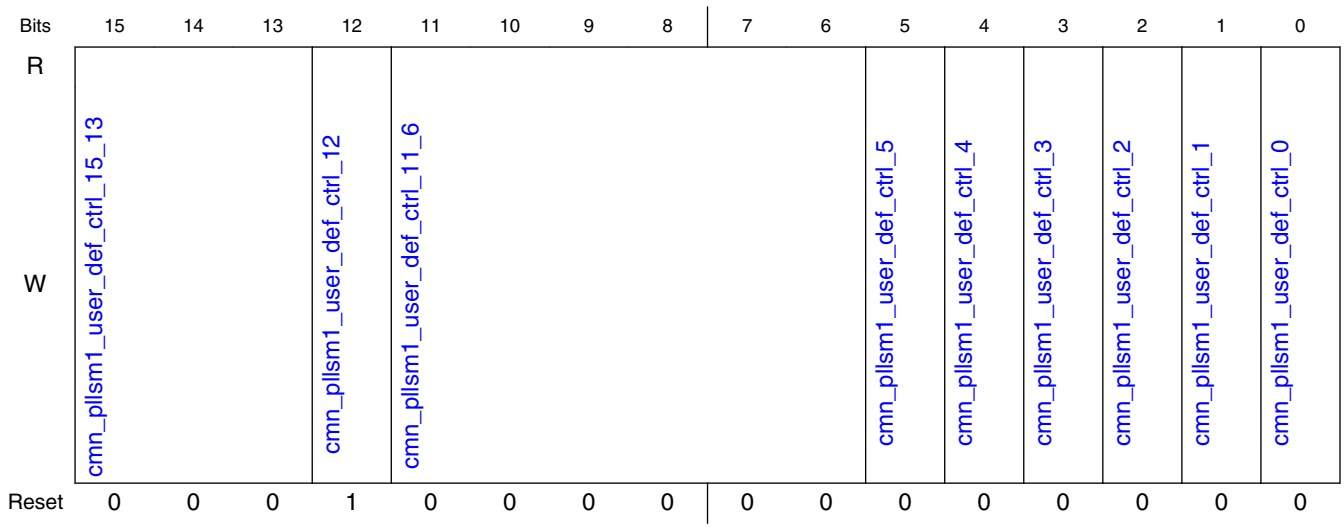
Field	Function
15-10 cmn_pllsm1_pllclkdis_tmr_15_10	Reserved
9-0 cmn_pllsm1_pllclkdis_tmr_9_0	PLL clock disable delay state timer value : Value used for the timer when the startup state machine is in the PLL clock disable delay state. This timer delay is specified as the number of reference clocks to count.

13.4.10.2.429 PLL 1 control state machine user defined control register (cmn_pllsm1_user_def_ctrl)

13.4.10.2.429.1 Offset

Register	Offset
cmn_pllsm1_user_def_ctrl	8_0037h

13.4.10.2.429.2 Diagram



13.4.10.2.429.3 Fields

Field	Function
15-13 cmn_pllsm1_user_def_ctrl_15_13	Reserved
12 cmn_pllsm1_user_def_ctrl_12	PLL oscillator ring select: Selects which oscillator ring is enabled.
11-6 cmn_pllsm1_user_def_ctrl_11_6	Reserved
5 cmn_pllsm1_user_def_ctrl_5	PLL bias enable override enable : When active (1b1), the PLL bias enable override bit in this register will drive the
4 cmn_pllsm1_user_def_ctrl_4	PLL bias enable override : When enabled by the PLL bias enable override enable bit in this register, this bit will drive the
3 cmn_pllsm1_user_def_ctrl_3	PLL 1 start loop override enable : When active (1b1), the PLL 1 start loop override bit in this register will drive the
2 cmn_pllsm1_user_def_ctrl_2	PLL 1 start loop enable override : When enabled by the PLL 1 start loop override enable bit in this register, this bit will drive the
1	PLL 1 VCO LDO enable override enable : When active (1b1), the PLL 1 VCO LDO enable override bit in this register will drive the

Table continues on the next page...

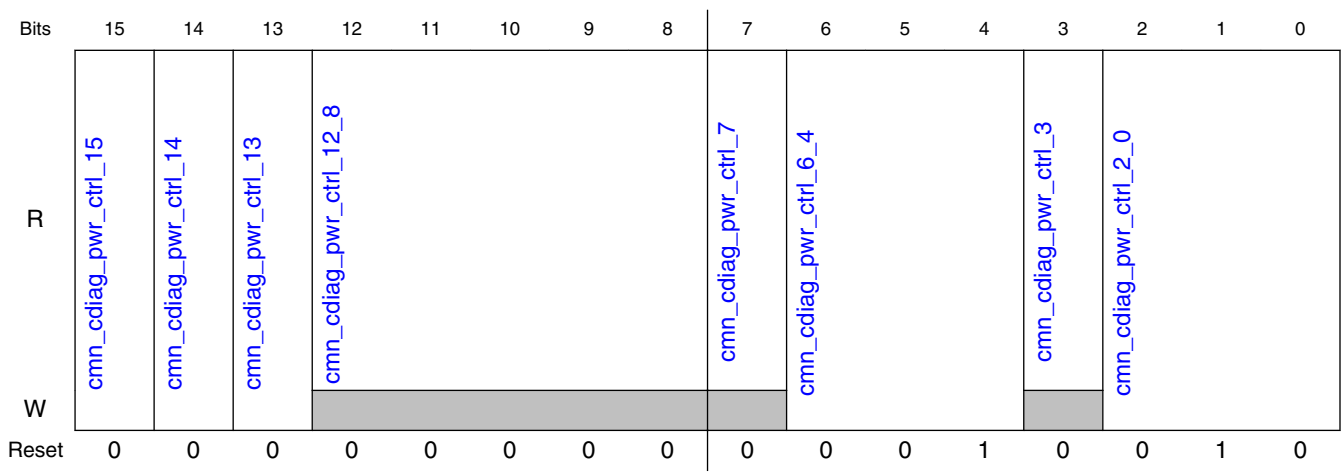
Field	Function
cmn_pllsm1_user_def_ctrl_1	
0 cmn_pllsm1_user_def_ctrl_0	PLL 1 VCO LDO enable override : When enabled by the PLL 1 VCO LDO enable override enable bit in this register, this bit will drive the

13.4.10.2.430 Common control power island control register (cmn_cdiag_pwr_ctrl)

13.4.10.2.430.1 Offset

Register	Offset
cmn_cdiag_pwr_ctrl	8_0060h

13.4.10.2.430.2 Diagram



13.4.10.2.430.3 Fields

Field	Function
15 cmn_cdiag_pwr_ctrl_15	Startup State Machine Auto Calibration Power down disable: Setting this bit to 1b1 will disable turning off the startup state machine auto calibration power island when the macro is in a state that would normally switch the power islands off.
14 cmn_cdiag_pwr_ctrl_14	PLL Control 1 State Machine Auto Calibration Power down disable: Setting this bit to 1b1 will disable turning off the PLL control 1 state machine calibration power island when the macro is in a state that would normally switch the power islands off.

Table continues on the next page...

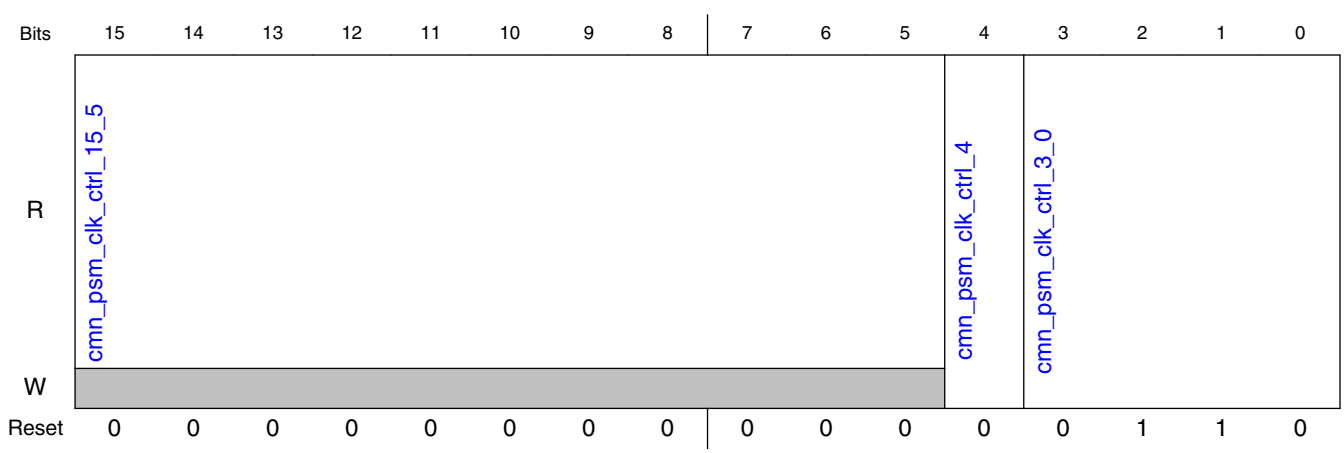
Field	Function
13 cmn_cdiag_pwr_ctrl_13	PLL Control 0 State Machine Auto Calibration Power down disable: Setting this bit to 1b1 will disable turning off the PLL control 0 state machine calibration power island when the macro is in a state that would normally switch the power islands off.
12-8 cmn_cdiag_pwr_ctrl_12_8	Reserved
7 cmn_cdiag_pwr_ctrl_7	Reserved
6-4 cmn_cdiag_pwr_ctrl_6_4	Power enable phase 2 timer value: This specifies the number of reference clock cycles all the power island control state machines will wait in the power phase 2 enable states, in order to allow enough time for the second phase of the switched domain to power up, before deactivating the isolation functions.
3 cmn_cdiag_pwr_ctrl_3	Reserved
2-0 cmn_cdiag_pwr_ctrl_2_0	Power enable phase 1 timer value: This specifies the number of reference clock cycles all the power island control state machine will wait in the power phase 1 enable states, in order to allow enough time for the first phase of the switched domain to power up, before enabling the second phase of the power up.

13.4.10.2.431 Common PSM clock control register (cmn_psm_clk_ctrl)

13.4.10.2.431.1 Offset

Register	Offset
cmn_psm_clk_ctrl	8_0061h

13.4.10.2.431.2 Diagram



13.4.10.2.431.3 Fields

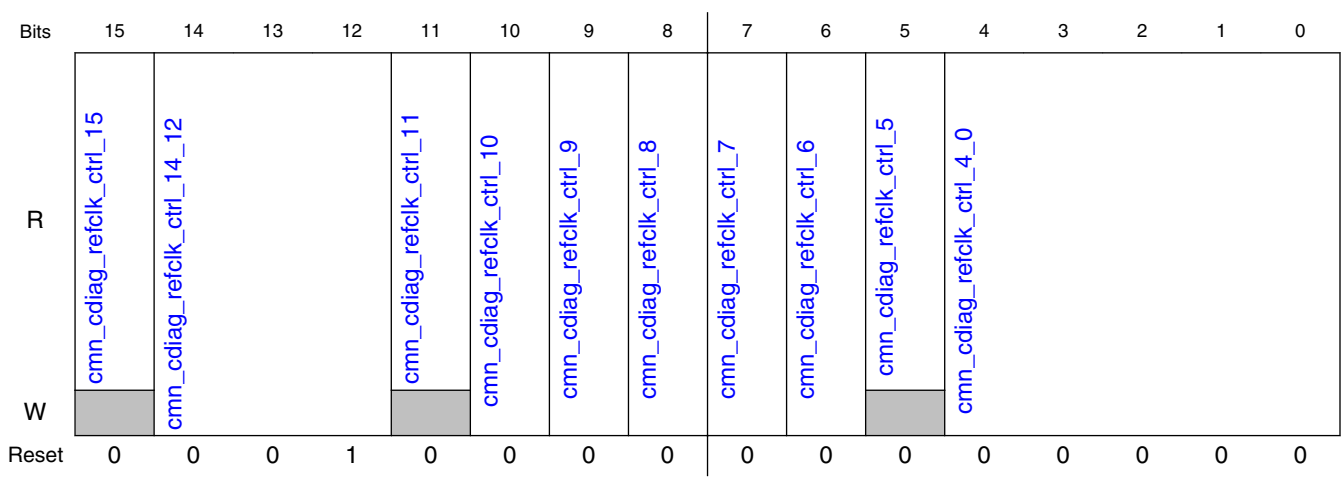
Field	Function
15-5 cmn_psm_clk_ctrl_15_5	Reserved
4 cmn_psm_clk_ctrl_4	PSM clock select: This bit selects the source of the PSM clock.
3-0 cmn_psm_clk_ctrl_3_0	PLL clock divider value for PSM clock: This specifies the value of the clock divider associated with PLL 0 that is used to generate the PSM clock.

13.4.10.2.432 Reference clock receiver control register (cmn_cdiag_refclk_ctrl)

13.4.10.2.432.1 Offset

Register	Offset
cmn_cdiag_refclk_ctrl	8_0062h

13.4.10.2.432.2 Diagram



13.4.10.2.432.3 Fields

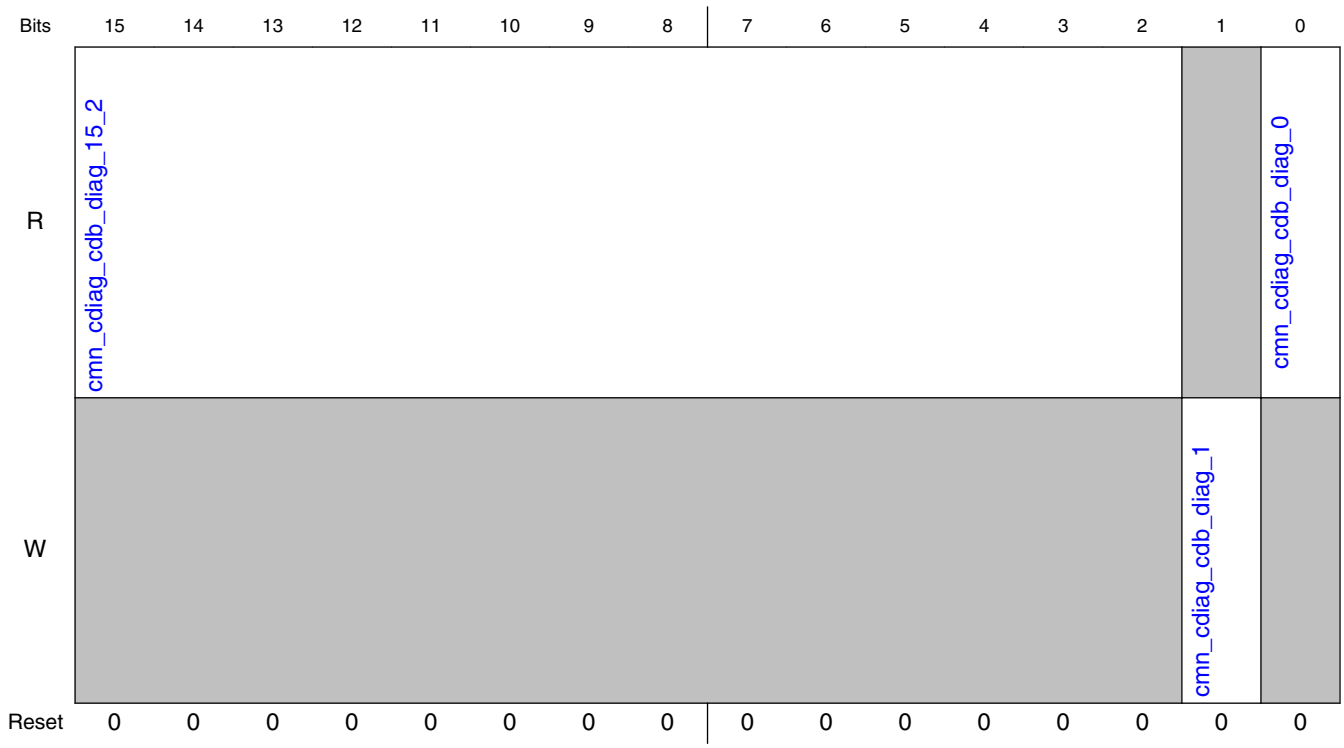
Field	Function
15 cmn_cdiag_refclk_ctrl_15	Reserved
14-12 cmn_cdiag_refclk_ctrl_14_12	Digital reference clock divider scaler: Scales the reference clock digital divider control signal (
11 cmn_cdiag_refclk_ctrl_11	Reserved
10 cmn_cdiag_refclk_ctrl_10	Reference clock analog clock test mode: Test mode signal that allows the analog path reference clock to drive the digital, by driving the
9 cmn_cdiag_refclk_ctrl_9	Reference clock digital clock test mode: Test mode signal that allows the digital path reference clock to drive the PLL, by driving the
8 cmn_cdiag_refclk_ctrl_8	Reference clock hysteresis enable: Enables the hysteresis in the digital slicer output, by driving the
7 cmn_cdiag_refclk_ctrl_7	Reference clock termination enable override enable: When this bit is set, the reference clock termination enable override bit in this register will override the reference clock termination enable, by driving the
6 cmn_cdiag_refclk_ctrl_6	Reference clock termination enable override: When the reference clock termination enable override enable bit in this register is set to 1b1, this bit will override the reference clock termination enable, by driving the
5 cmn_cdiag_refclk_ctrl_5	Reserved
4-0 cmn_cdiag_refclk_ctrl_4_0	Reference clock termination code: Controls the integrated reference clock termination impedance, by driving the

13.4.10.2.433 Common control CDB diagnostic register (cmn_cdiag_cdb_diag)

13.4.10.2.433.1 Offset

Register	Offset
cmn_cdiag_cdb_diag	8_0064h

13.4.10.2.433.2 Diagram



13.4.10.2.433.3 Fields

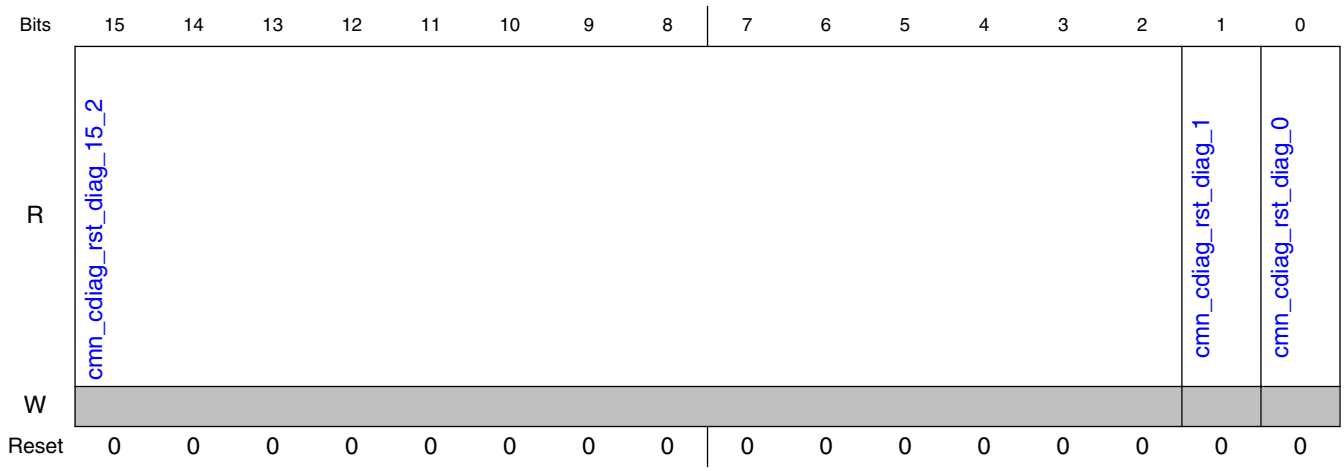
Field	Function
15-2 cmn_cdiag_cdb_diag_15_2	Reserved
1 cmn_cdiag_cdb_diag_1	Force clear of the CDB first access detected register bit : Writing a 1b1 to this register will force the clearing of the CDB first access detected register bit, which is the register bit that overrides the
0 cmn_cdiag_cdb_diag_0	CDB bus error: This bit will be set when the internal CDB watchdog timer expires. It is automatically cleared on a read of this register.

13.4.10.2.434 Common control reset diagnostic register (cmn_cdiag_rst_diag)

13.4.10.2.434.1 Offset

Register	Offset
cmn_cdiag_rst_diag	8_0066h

13.4.10.2.434.2 Diagram



13.4.10.2.434.3 Fields

Field	Function
15-2 cmn_cdiag_rst_diag_15_2	Reserved
1 cmn_cdiag_rst_diag_1	Current state of the macro_pwr_reset_sync_n reset.
0 cmn_cdiag_rst_diag_0	Current state of the cmn_reset_sync_n reset.

13.4.10.2.435 Common control cover your alternatives register (cmn_cdiag_dcya)

13.4.10.2.435.1 Offset

Register	Offset
cmn_cdiag_dcya	8_007Fh

13.4.10.2.435.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cmn_cdiag_dcya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.435.3 Fields

Field	Function
15-0 cmn_cdiag_dcya_15_0	Reserved

13.4.10.2.436 PLL 0 VCO calibration control register (cmn_pll0_vcocal_ctrl)

13.4.10.2.436.1 Offset

Register	Offset
cmn_pll0_vcocal_ctrl	8_0080h

13.4.10.2.436.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_pll0_vcocal_ctrl_15	cmn_pll0_vcocal_ctrl_14	cmn_pll0_vcocal_ctrl_13_9					cmn_pll0_vcocal_ctrl_8_0								
W																
Reset	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0

13.4.10.2.436.3 Fields

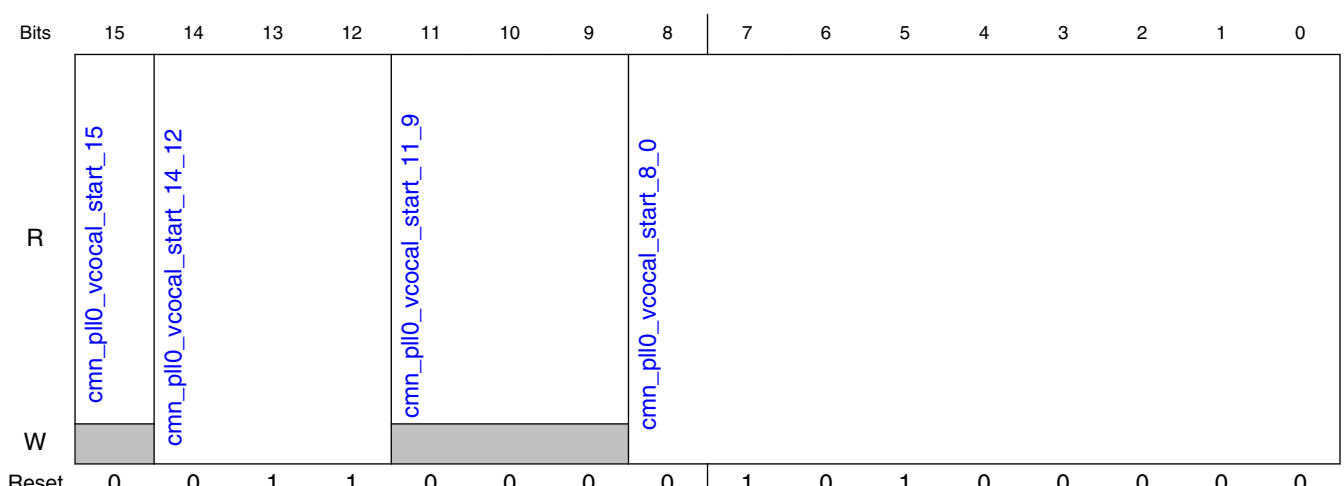
Field	Function
15 cmn_pll0_vcocal_ctrl_15	Start VCO calibration: Activating (1b1) this bit will start a VCO calibration process. This bit must remain active until the VCO calibration process is complete (as indicated by the VCO calibration process done bit in this register). To start another VCO calibration process, the VCO calibration process done bit must have gone inactive from any prior calibration process.
14 cmn_pll0_vcocal_ctrl_14	VCO calibration process done: This bit will be set to 1b1 when the VCO calibration process is complete. It will be cleared by the deactivation of the Start VCO calibration bit in this register.
13-9 cmn_pll0_vcocal_ctrl_13_9	Reserved
8-0 cmn_pll0_vcocal_ctrl_8_0	VCO calibration code: This is the calibration code that was determined by the VCO calibration process. This signal is valid when the VCO calibration process is complete. The values of this field correspond to different frequency bands the VCO will operate in. The frequency bands are controlled by the number of capacitors that are switched in the VCO analog circuit. This field specifies the number of capacitors that are switched in. The following are the values for this code:

13.4.10.2.437 PLL 0 VCO calibration start point register (cmn_pll0_vcocal_start)

13.4.10.2.437.1 Offset

Register	Offset
cmn_pll0_vcocal_start	8_0081h

13.4.10.2.437.2 Diagram



13.4.10.2.437.3 Fields

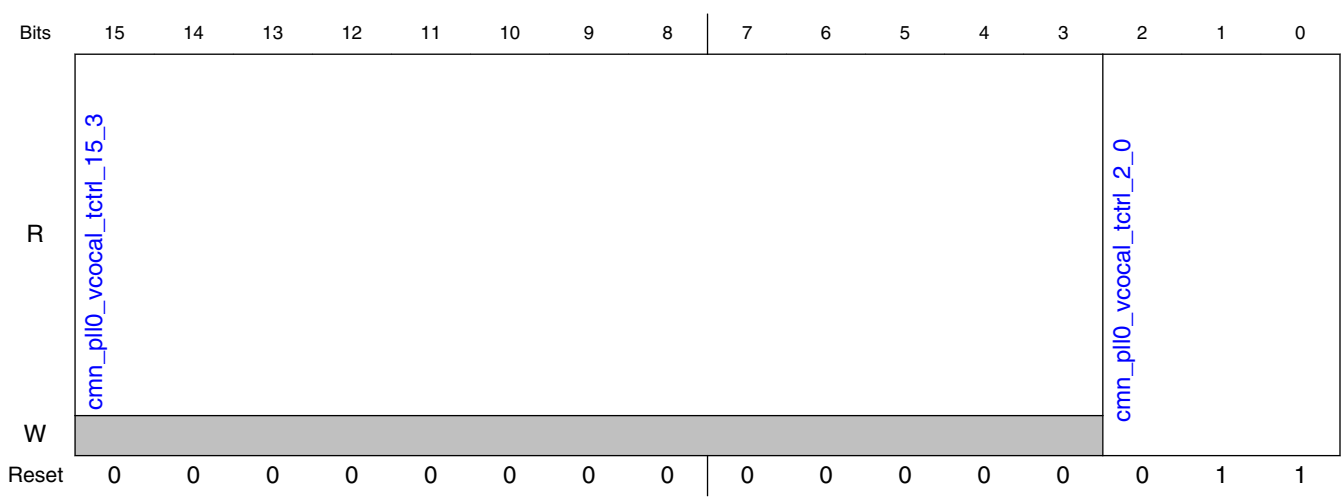
Field	Function
15 cmn_pll0_vcocal_start_15	Reserved
14-12 cmn_pll0_vcocal_start_14_12	VCO calibration initial step size control: This field specifies the initial step size for the VCO calibration state machine. The following are the values that can be used in this field, and the corresponding step sizes.
11-9 cmn_pll0_vcocal_start_11_9	Reserved
8-0 cmn_pll0_vcocal_start_8_0	VCO calibration code starting point value: This field specifies the starting VCO code that is used by the VCO calibration state machine. The purpose of this value is such that the VCO calibration process starts at a point that is, on average, relatively close to the final calibration point. This allows the calibration time to be reduced, on average.

13.4.10.2.438 PLL 0 VCO calibration timer control register (cmn_pll0_vcocal_tctrl)

13.4.10.2.438.1 Offset

Register	Offset
cmn_pll0_vcocal_tctrl	8_0082h

13.4.10.2.438.2 Diagram



13.4.10.2.438.3 Fields

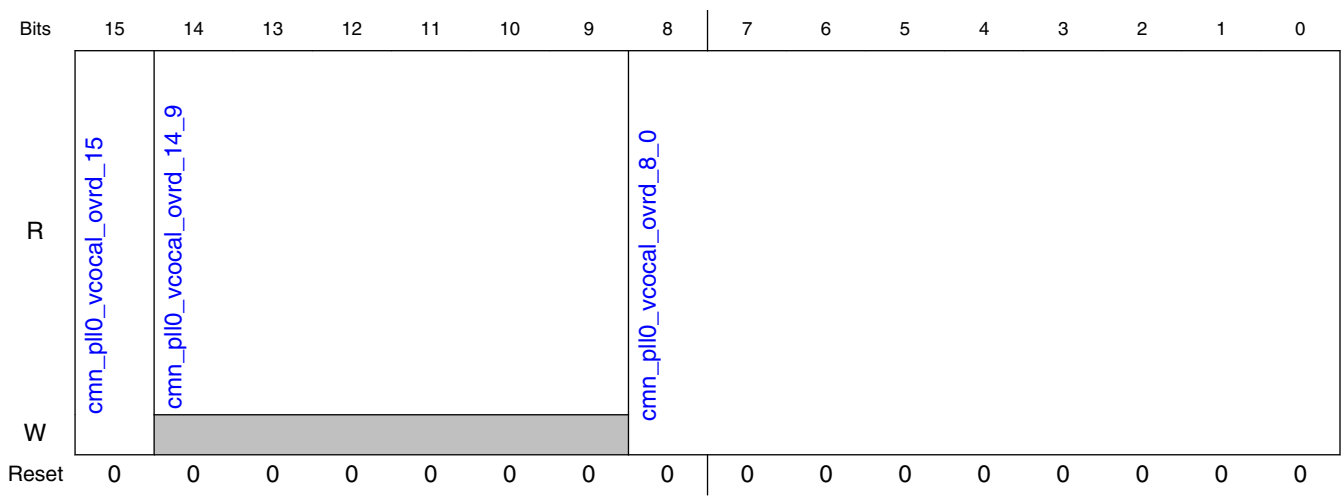
Field	Function
15-3 cmn_pll0_vcocal_tctrl_15_3	Reserved
2-0 cmn_pll0_vcocal_tctrl_2_0	VCO calibration initial time scale control: This field specifies the calibration start time scaling factor applied to the VCO calibration when running the initial step size for the calibration code is not set to 1. Setting this value to a value other than 1 will reduce the calibration measurement time by the amount specified below. Then when the final calibration steps are made the full calibration time will be used to get the appropriate amount of resolution.

13.4.10.2.439 PLL 0 VCO calibration override register (cmn_pll0_vcocal_ovrd)

13.4.10.2.439.1 Offset

Register	Offset
cmn_pll0_vcocal_ovrd	8_0083h

13.4.10.2.439.2 Diagram



13.4.10.2.439.3 Fields

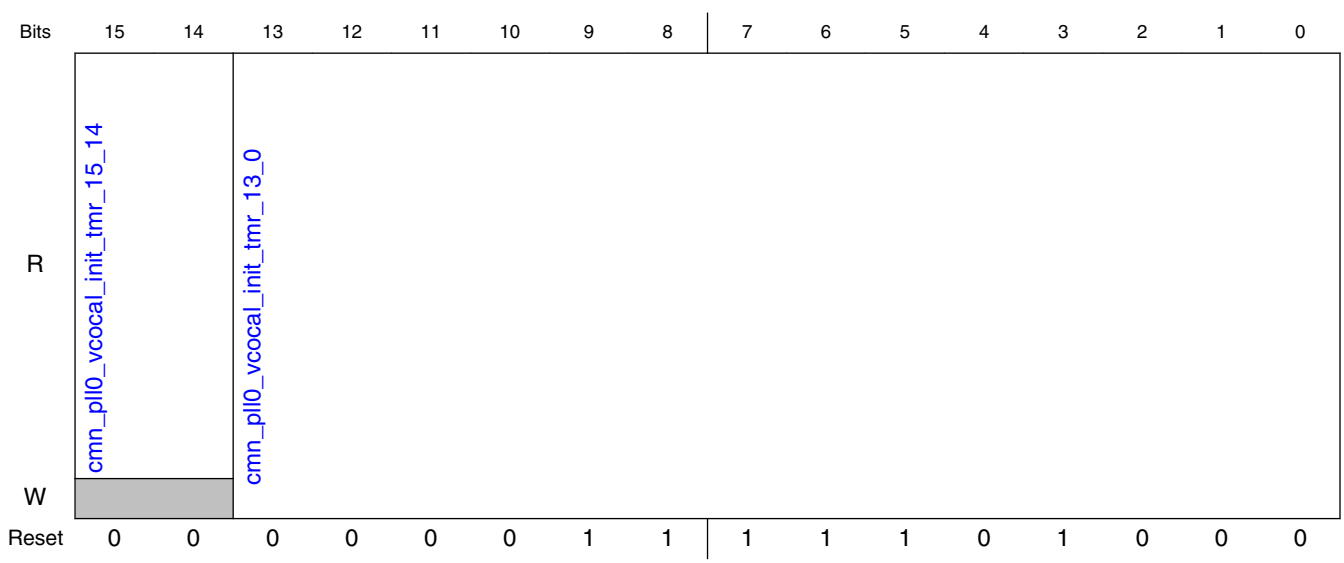
Field	Function
15 cmn_pll0_vcocal_ovrd_15	VCO calibration code override enable: Activating (1b1) this bit allows the VCO code determined during the automatic VCO calibration process to be overridden by the value driven by the VCO calibration code override value field in this register.
14-9 cmn_pll0_vcocal_ovrd_14_9	Reserved
8-0 cmn_pll0_vcocal_ovrd_8_0	VCO calibration code override value: This field is used to override the VCO code determined during the automatic VCO calibration process. The VCO code driven on this field is valid when the VCO calibration code override enable bit in this register is active.

13.4.10.2.440 PLL 0 VCO calibration initialization timer register (cmn_pll0_vcocal_init_tmr)

13.4.10.2.440.1 Offset

Register	Offset
cmn_pll0_vcocal_init_tmr	8_0084h

13.4.10.2.440.2 Diagram



13.4.10.2.440.3 Fields

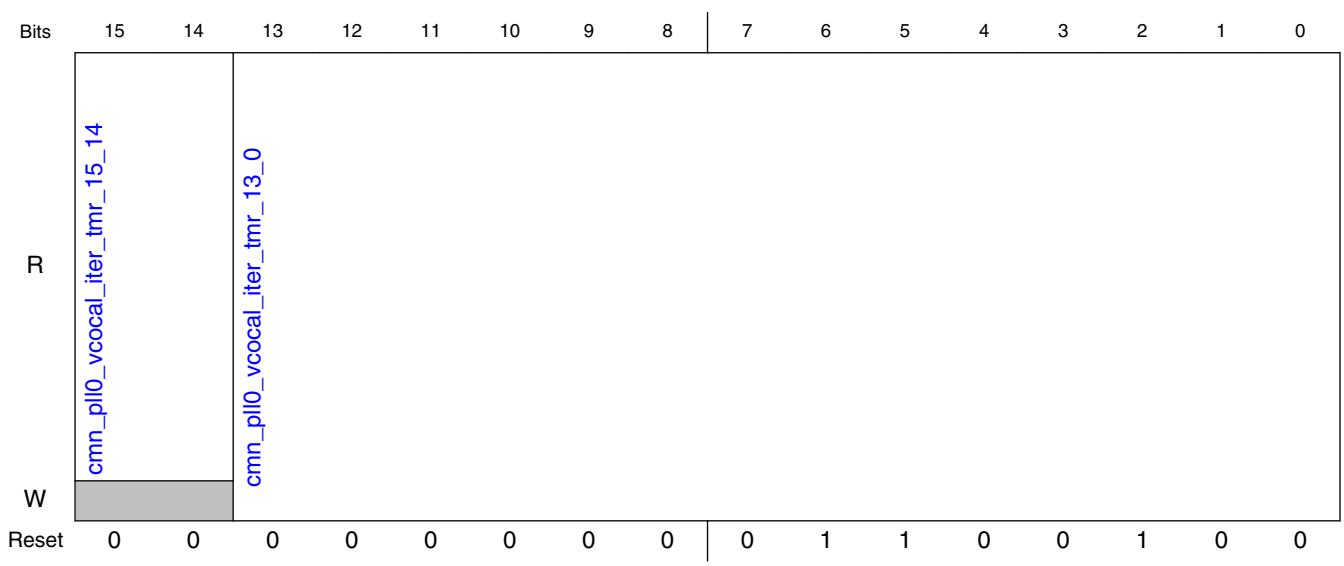
Field	Function
15-14 cmn_pll0_vcocal_init_tmr_15_14	Reserved
13-0 cmn_pll0_vcocal_init_tmr_13_0	Initialization wait timer value: This is the number of clocks to wait between when the analog VCO calibration circuits are enabled, and when the first calibration code is driven to the analog.

13.4.10.2.441 PLL 0 VCO calibration iteration timer register (cmn_pll0_vcocal_iter_tmr)

13.4.10.2.441.1 Offset

Register	Offset
cmn_pll0_vcocal_iter_tmr	8_0085h

13.4.10.2.441.2 Diagram



13.4.10.2.441.3 Fields

Field	Function
15-14	Reserved

Table continues on the next page...

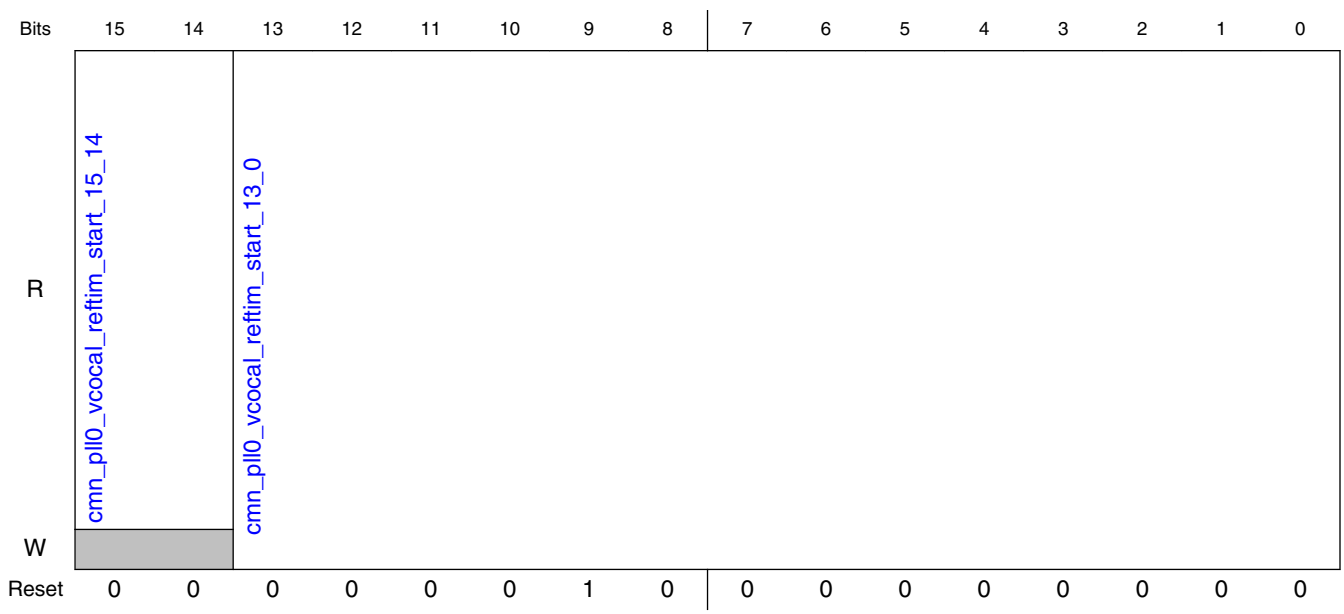
Field	Function
cmn_pll0_vcocal_iter_tmr_15_14	
13-0 cmn_pll0_vcocal_iter_tmr_13_0	Iteration wait timer value: This is the number of clocks to wait between when a calibration code is driven to the analog, and when the clock rates are measured.

13.4.10.2.442 PLL 0 VCO calibration reference clock timer start value register (cmn_pll0_vcocal_reftim_start)

13.4.10.2.442.1 Offset

Register	Offset
cmn_pll0_vcocal_reftim_start	8_0086h

13.4.10.2.442.2 Diagram



13.4.10.2.442.3 Fields

Field	Function
15-14	Reserved

Table continues on the next page...

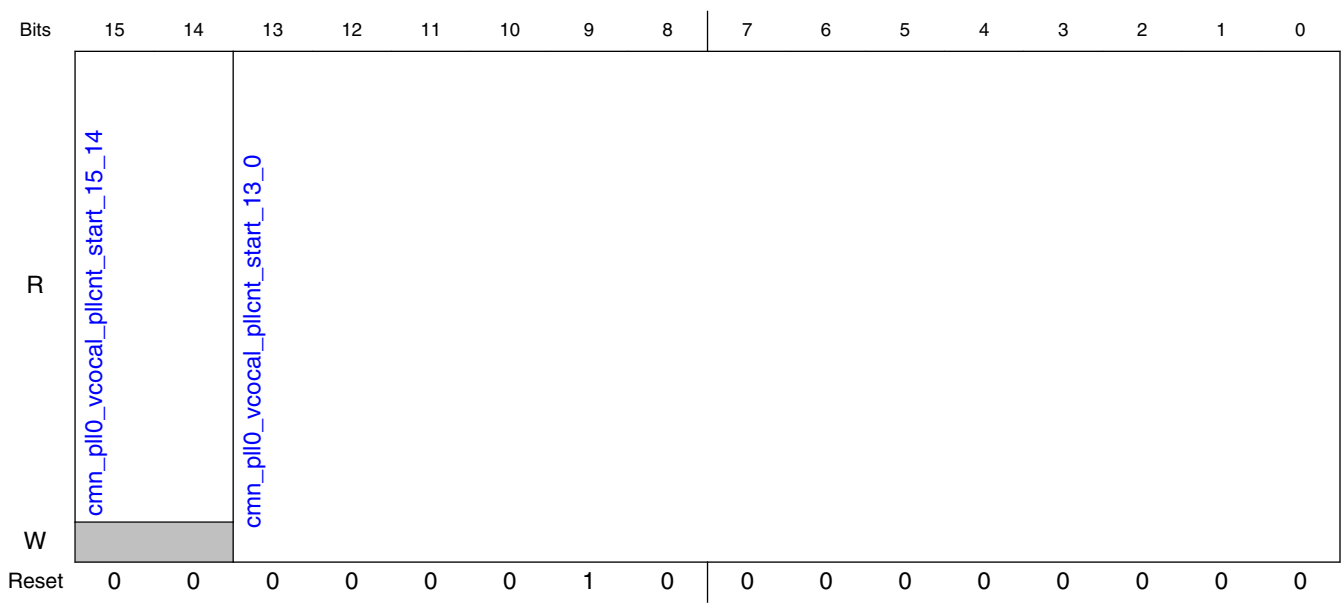
Field	Function
cmn_pll0_vcocal_reftim_start_15_14	
13-0 cmn_pll0_vcocal_reftim_start_13_0	PLL VCO calibration reference clock timer start value : This is the value that is loaded into the reference clock timer as the starting point for that timer, when running VCO calibration. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes.

13.4.10.2.443 PLL 0 VCO calibration PLL clock counter start value register (cmn_pll0_vcocal_pllcnt_start)

13.4.10.2.443.1 Offset

Register	Offset
cmn_pll0_vcocal_pllcnt_start	8_0088h

13.4.10.2.443.2 Diagram



13.4.10.2.443.3 Fields

Field	Function
15-14	Reserved

Table continues on the next page...

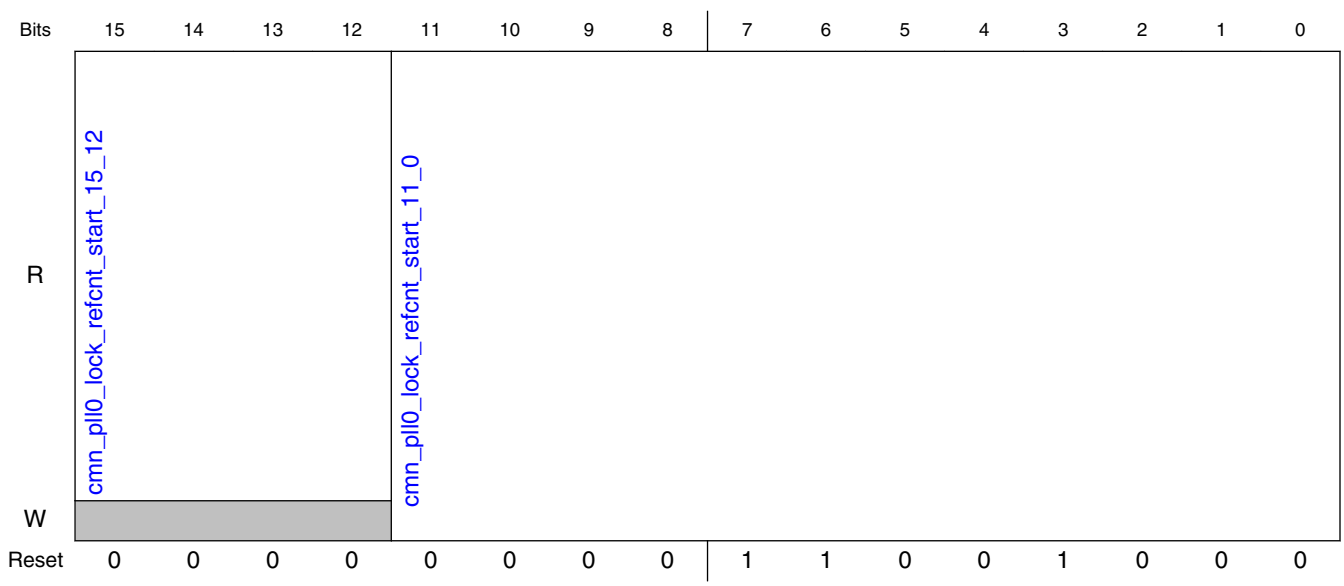
Field	Function
cmn_pll0_vcocal_pllcnt_start_15_14	
13-0 cmn_pll0_vcocal_pllcnt_start_13_0	PLL VCO calibration PLL clock counter start value : This is the value that is loaded into the PLL clock counter as the starting point for that counter, when running VCO calibration. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes.

13.4.10.2.444 PLL 0 lock reference counter start value register (cmn_pll0_lock_refcnt_start)

13.4.10.2.444.1 Offset

Register	Offset
cmn_pll0_lock_refcnt_start	8_0090h

13.4.10.2.444.2 Diagram



13.4.10.2.444.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

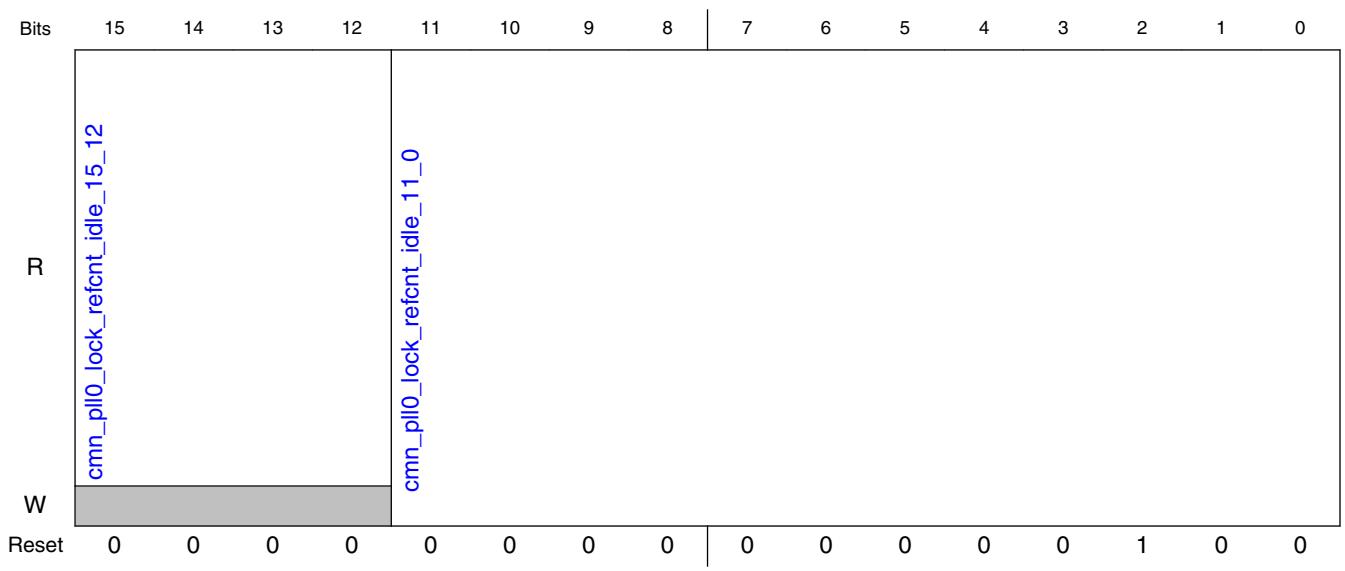
Field	Function
cmn_pll0_lock_refcnt_start_15_12	
11-0 cmn_pll0_lock_refcnt_start_11_0	PLL lock reference counter start value : This is the value that is loaded into the PLL lock detect reference counter as the starting point for that counter, when checking for PLL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. This is set to 200 clocks by default.

13.4.10.2.445 PLL 0 lock reference counter idle value register (cmn_pll0_lock_refcnt_idle)

13.4.10.2.445.1 Offset

Register	Offset
cmn_pll0_lock_refcnt_idle	8_0091h

13.4.10.2.445.2 Diagram



13.4.10.2.445.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

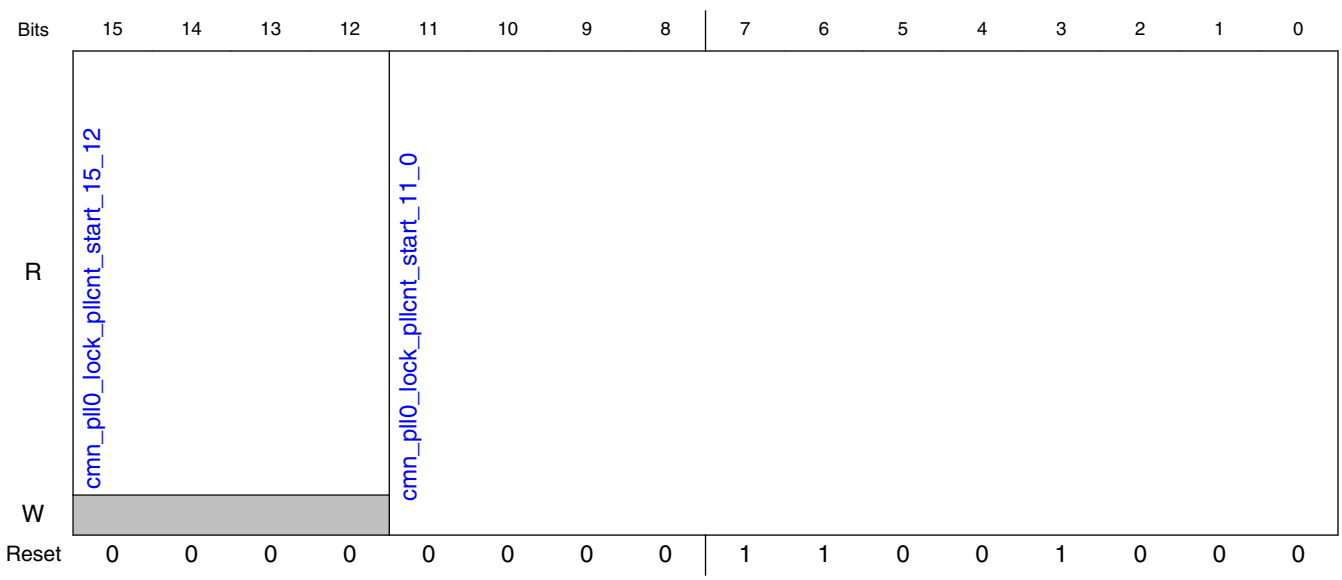
Field	Function
cmn_pll0_lock_refcnt_idle_15_12	
11-0 cmn_pll0_lock_refcnt_idle_11_0	PLL lock reference counter idle value : This is the value used by the PLL lock detection logic to specify the number of reference clocks between each phase of counting PLL clocks.

13.4.10.2.446 PLL 0 lock PLL counter start value register (cmn_pll0_lock_pllcnt_start)

13.4.10.2.446.1 Offset

Register	Offset
cmn_pll0_lock_pllcnt_start	8_0092h

13.4.10.2.446.2 Diagram



13.4.10.2.446.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

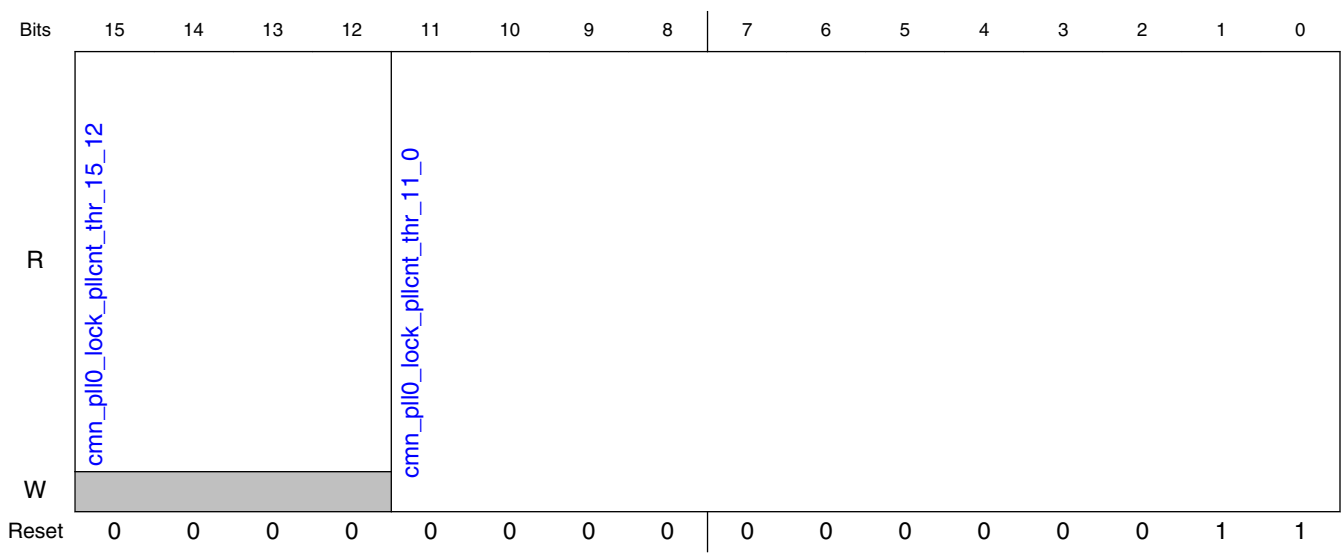
Field	Function
cmn_pll0_lock_pllcnt_start_15_12	
11-0 cmn_pll0_lock_pllcnt_start_11_0	PLL lock PLL counter start value : This is the value that is loaded into the PLL lock detect PLL counter as the starting point for that counter, when checking for PLL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. This is set to 200 clocks by default.

13.4.10.2.447 PLL 0 lock PLL counter threshold value register (cmn_pll0_lock_pllcnt_thr)

13.4.10.2.447.1 Offset

Register	Offset
cmn_pll0_lock_pllcnt_thr	8_0093h

13.4.10.2.447.2 Diagram



13.4.10.2.447.3 Fields

Field	Function
15-12 cmn_pll0_lock_pllcnt_thr_15_12	Reserved

Table continues on the next page...

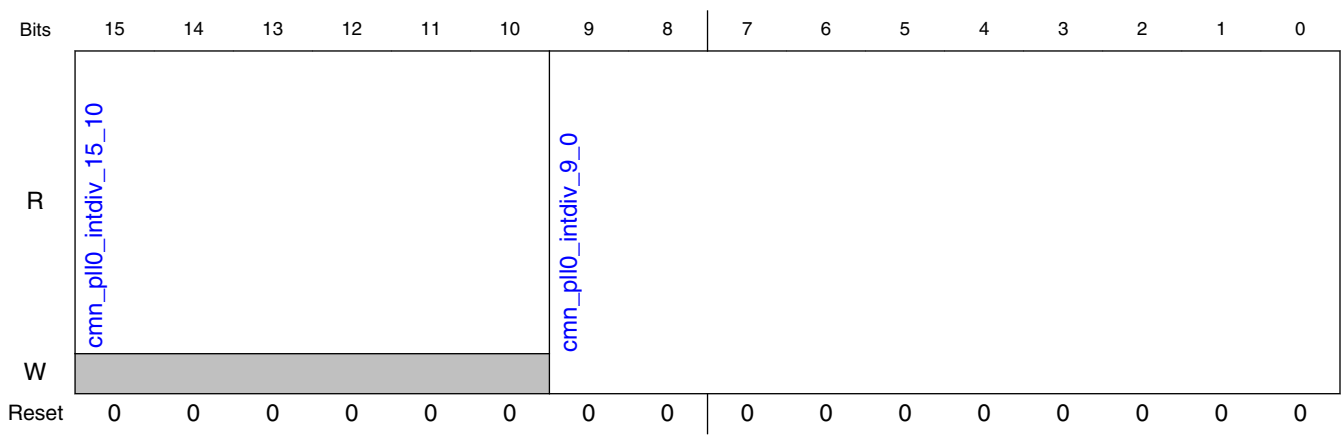
Field	Function
11-0 cmn_pll0_lock_pllcnt_thr_11_0	PLL lock counter threshold value : This is the value used by the PLL lock detection logic to determine if the PLL has locked. If the two counters in the PLL lock detection logic differ by less than this value, the PLL is considered locked. This is set to 3 clocks by default.

13.4.10.2.448 PLL 0 feedback divider integer register (cmn_pll0_intdiv)

13.4.10.2.448.1 Offset

Register	Offset
cmn_pll0_intdiv	8_0094h

13.4.10.2.448.2 Diagram



13.4.10.2.448.3 Fields

Field	Function
15-10 cmn_pll0_intdiv_15_10	Reserved
9-0 cmn_pll0_intdiv_9_0	pll_fb_div_integer value: Value of the pll_fb_div_integer signal.

13.4.10.2.449 PLL 0 feedback divider fractional register (cmn_pll0_fracdiv)

13.4.10.2.449.1 Offset

Register	Offset
cmn_pll0_fracdiv	8_0095h

13.4.10.2.449.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cmn_pll0_fracdiv_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.449.3 Fields

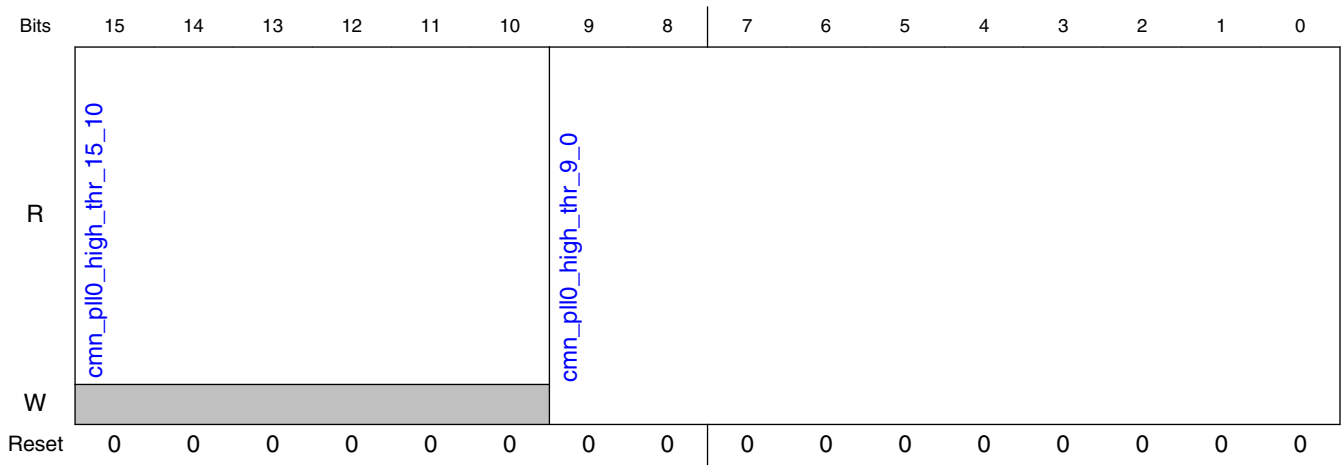
Field	Function
15-0 cmn_pll0_fracdiv_15_0	pll_fb_div_fractional: Value of the pll_fb_div_fractional signal.

13.4.10.2.450 PLL 0 feedback divider high threshold register (cmn_pll0_high_thr)

13.4.10.2.450.1 Offset

Register	Offset
cmn_pll0_high_thr	8_0096h

13.4.10.2.450.2 Diagram



13.4.10.2.450.3 Fields

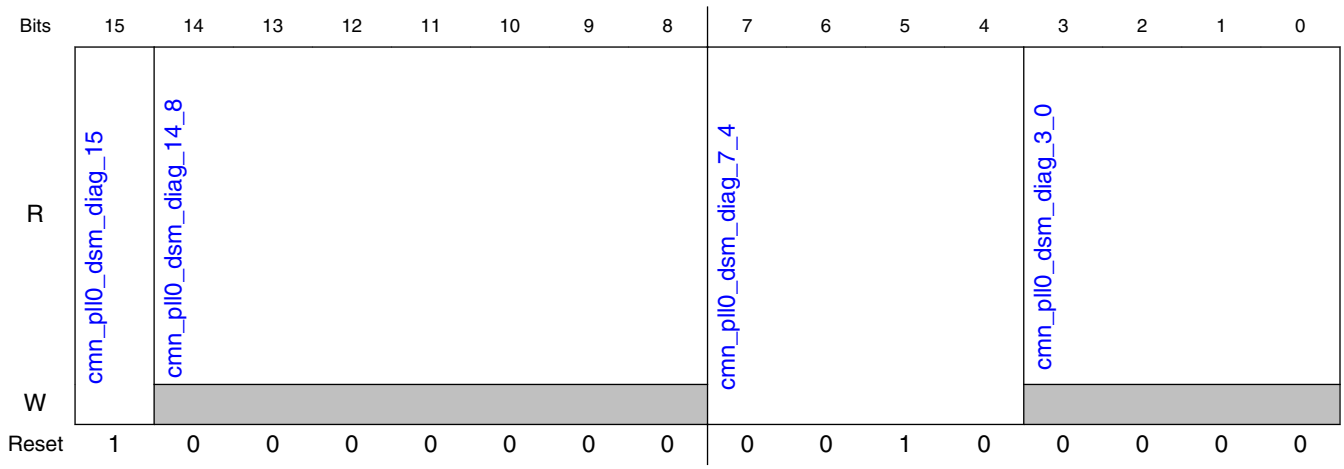
Field	Function
15-10 cmn_pll0_high_thr_15_10	Reserved
9-0 cmn_pll0_high_thr_9_0	pll_fb_div_high_theshold: Value of the pll_fb_div_high_threshold signal.

13.4.10.2.451 PLL 0 delta sigma modulator diagnostics register (cmn_pll0_dsm_diag)

13.4.10.2.451.1 Offset

Register	Offset
cmn_pll0_dsm_diag	8_0097h

13.4.10.2.451.2 Diagram



13.4.10.2.451.3 Fields

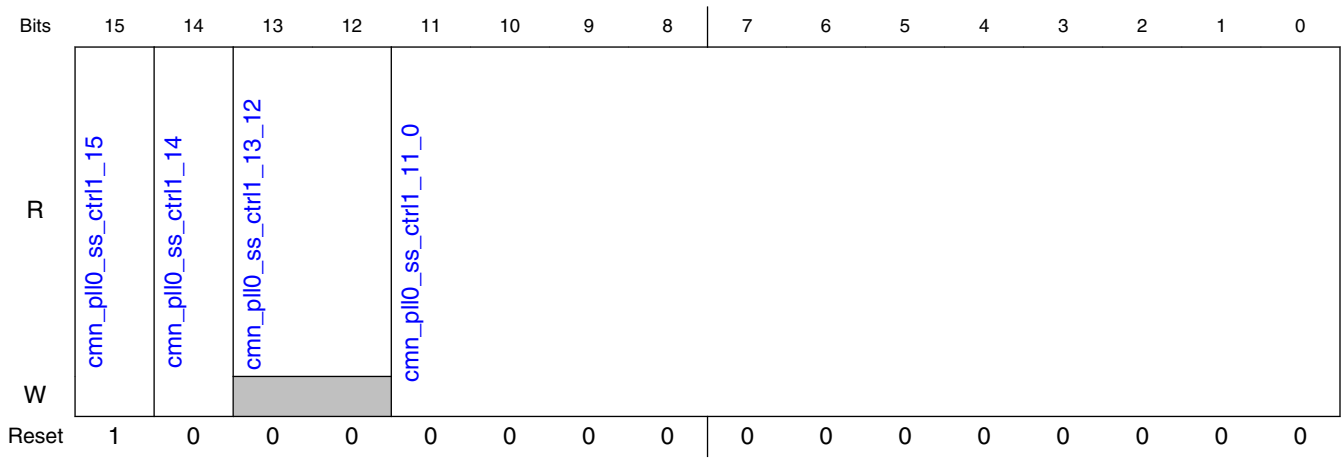
Field	Function
15 cmn_pll0_dsm_diag_15	Delta sigma bypass enable: When set to 1b1, the delta sigma modulator will be bypassed, and the output will be the value specified for the internal pll_fb_div_integer signal.
14-8 cmn_pll0_dsm_diag_14_8	Reserved
7-4 cmn_pll0_dsm_diag_7_4	PLL feedback divider latency adjustment: This signal specifies a value to be subtracted from the feedback divider settings before they are output on the
3-0 cmn_pll0_dsm_diag_3_0	Reserved

13.4.10.2.452 PLL 0 spread spectrum control register 1 (cmn_pll0_ss_ctrl 1)

13.4.10.2.452.1 Offset

Register	Offset
cmn_pll0_ss_ctrl1	8_0098h

13.4.10.2.452.2 Diagram



13.4.10.2.452.3 Fields

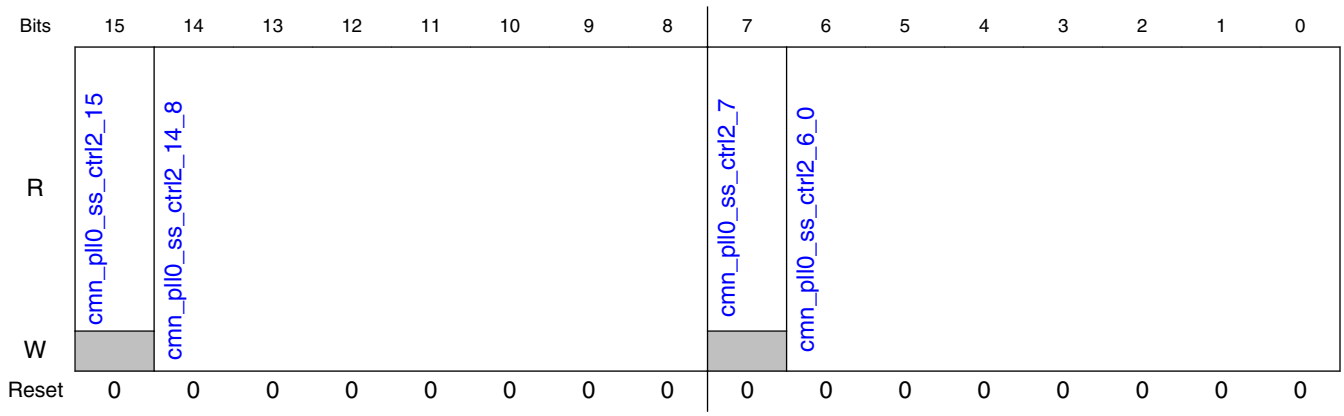
Field	Function
15 cmn_pll0_ss_ctrl1_15	Spread spectrum waveform generator disable: Setting this bit to a 1b1 will disable the spread spectrum waveform generator.
14 cmn_pll0_ss_ctrl1_14	Spread spectrum enable during VCO calibration : Setting this bit to a 1b1 will enable the spread spectrum function while VCO calibration is taking place.
13-12 cmn_pll0_ss_ctrl1_13_12	Reserved
11-0 cmn_pll0_ss_ctrl1_11_0	Amplitude step size: Value of the amplitude_step_size pin on the spread spectrum waveform generator.

13.4.10.2.453 PLL 0 spread spectrum control register 2 (cmn_pll0_ss_ctrl2)

13.4.10.2.453.1 Offset

Register	Offset
cmn_pll0_ss_ctrl2	8_0099h

13.4.10.2.453.2 Diagram



13.4.10.2.453.3 Fields

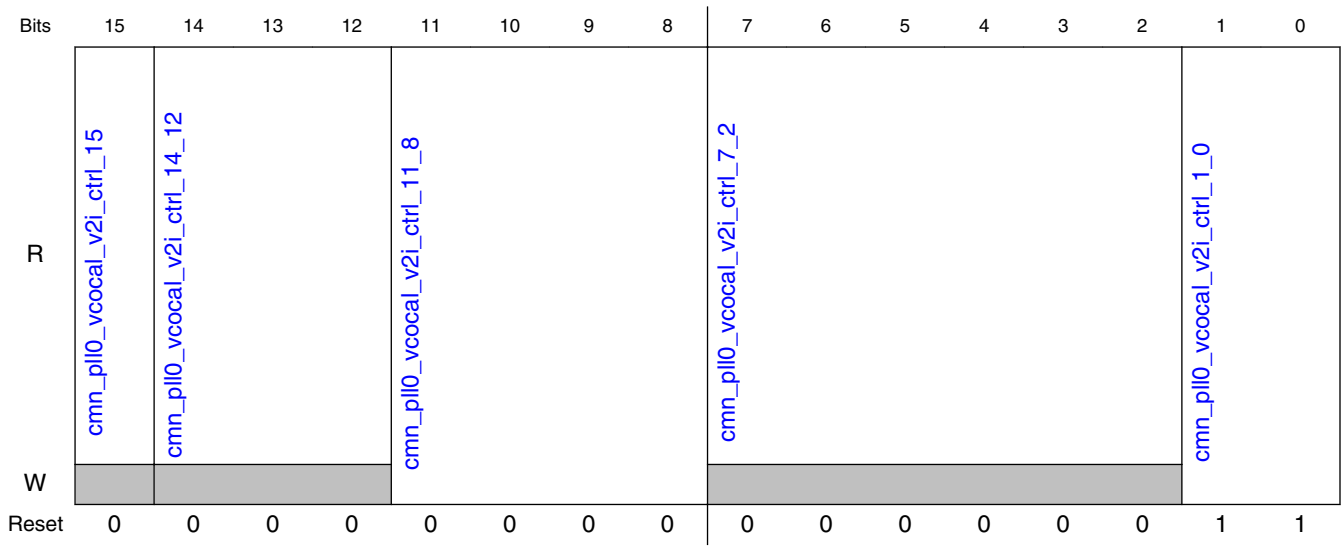
Field	Function
15 cmn_pll0_ss_ctrl2_15	Reserved
14-8 cmn_pll0_ss_ctrl2_14_8	Number of steps: Value of the num_steps pin on the spread spectrum waveform generator.
7 cmn_pll0_ss_ctrl2_7	Reserved
6-0 cmn_pll0_ss_ctrl2_6_0	Time step size: Value for the time_step_size pin on the spread spectrum waveform generator.

13.4.10.2.454 PLL 0 VCO calibration V2I control register (cmn_pll0_vcocal_v2i_ctrl)

13.4.10.2.454.1 Offset

Register	Offset
cmn_pll0_vcocal_v2i_ctrl	8_009Ch

13.4.10.2.454.2 Diagram



13.4.10.2.454.3 Fields

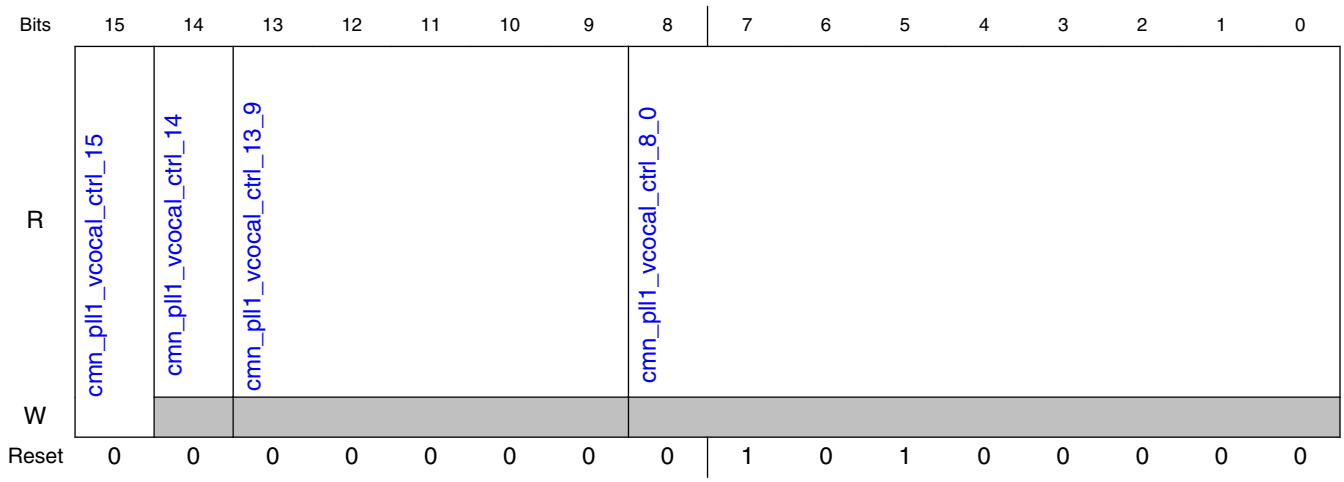
Field	Function
15 cmn_pll0_vcocal_v2i_ctrl_15	VCO calibration code for voltage to current DAC override enable: Activating (1b1) this bit allows the
14-12 cmn_pll0_vcocal_v2i_ctrl_14_12	Reserved
11-8 cmn_pll0_vcocal_v2i_ctrl_11_8	VCO calibration code for voltage to current DAC override value: This field is used to override the
7-2 cmn_pll0_vcocal_v2i_ctrl_7_2	Reserved
1-0 cmn_pll0_vcocal_v2i_ctrl_1_0	VCO calibration code for voltage to current DAC scaling factor: This field is used to control the scaling function in the logic used in the internal calculations related to the VCO calibration code for voltage to current DAC. The following is the encoding used for this field.

13.4.10.2.455 PLL 1 VCO calibration control register (cmn_pll1_vcocal_ctrl)

13.4.10.2.455.1 Offset

Register	Offset
cmn_pll1_vcocal_ctrl	8_00A0h

13.4.10.2.455.2 Diagram



13.4.10.2.455.3 Fields

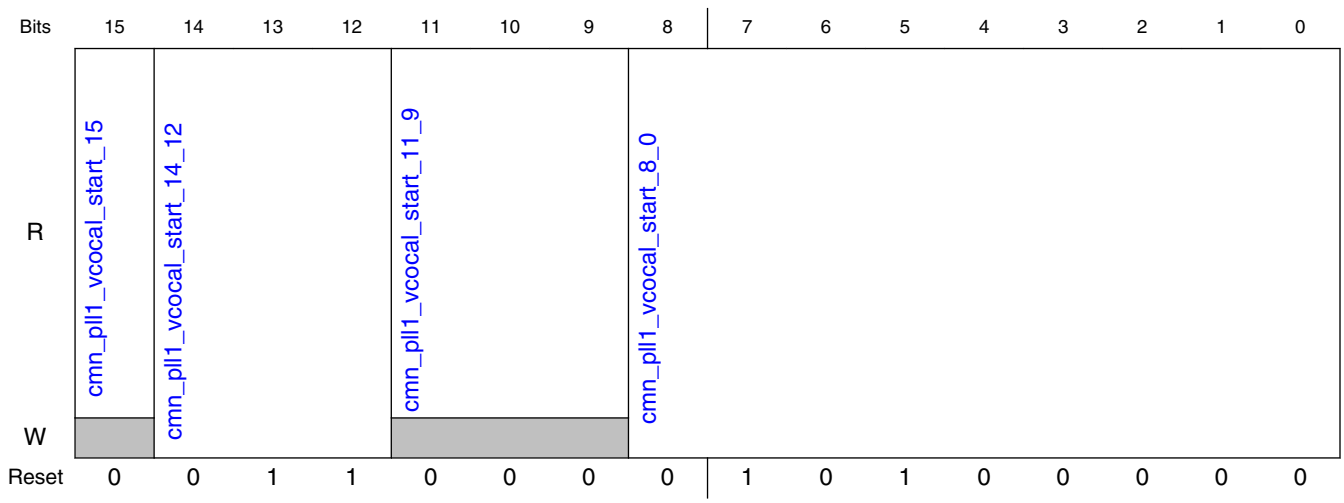
Field	Function
15 cmn_pll1_vcocal_ctrl_15	Start VCO calibration: Activating (1b1) this bit will start a VCO calibration process. This bit must remain active until the VCO calibration process is complete (as indicated by the VCO calibration process done bit in this register). To start another VCO calibration process, the VCO calibration process done bit must have gone inactive from any prior calibration process.
14 cmn_pll1_vcocal_ctrl_14	VCO calibration process done: This bit will be set to 1b1 when the VCO calibration process is complete. It will be cleared by the deactivation of the Start VCO calibration bit in this register.
13-9 cmn_pll1_vcocal_ctrl_13_9	Reserved
8-0 cmn_pll1_vcocal_ctrl_8_0	VCO calibration code: This is the calibration code that was determined by the VCO calibration process. This signal is valid when the VCO calibration process is complete. The values of this field correspond to different frequency bands the VCO will operate in. The frequency bands are controlled by the number of capacitors that are switched in the VCO analog circuit. This field specifies the number of capacitors that are switched in. The following are the values for this code:

13.4.10.2.456 PLL 1 VCO calibration start point register (cmn_pll1_vcocal_start)

13.4.10.2.456.1 Offset

Register	Offset
cmn_pll1_vcocal_start	8_00A1h

13.4.10.2.456.2 Diagram



13.4.10.2.456.3 Fields

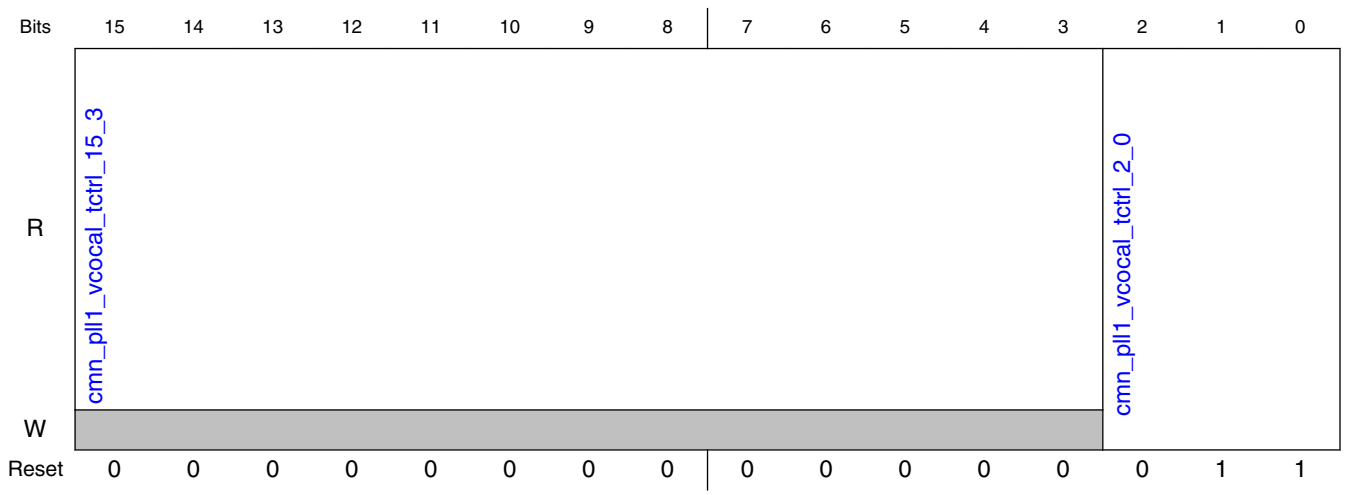
Field	Function
15 cmn_pll1_vcocal_start_15	Reserved
14-12 cmn_pll1_vcocal_start_14_12	VCO calibration initial step size control: This field specifies the initial step size for the VCO calibration state machine. The following are the values that can be used in this field, and the corresponding step sizes.
11-9 cmn_pll1_vcocal_start_11_9	Reserved
8-0 cmn_pll1_vcocal_start_8_0	VCO calibration code starting point value: This field specifies the starting VCO code that is used by the VCO calibration state machine. The purpose of this value is such that the VCO calibration process starts at a point that is, on average, relatively close to the final calibration point. This allows the calibration time to be reduced, on average.

13.4.10.2.457 PLL 1 VCO calibration timer control register (cmn_pll1_vcocal_tctrl)

13.4.10.2.457.1 Offset

Register	Offset
cmn_pll1_vcocal_tctrl	8_00A2h

13.4.10.2.457.2 Diagram



13.4.10.2.457.3 Fields

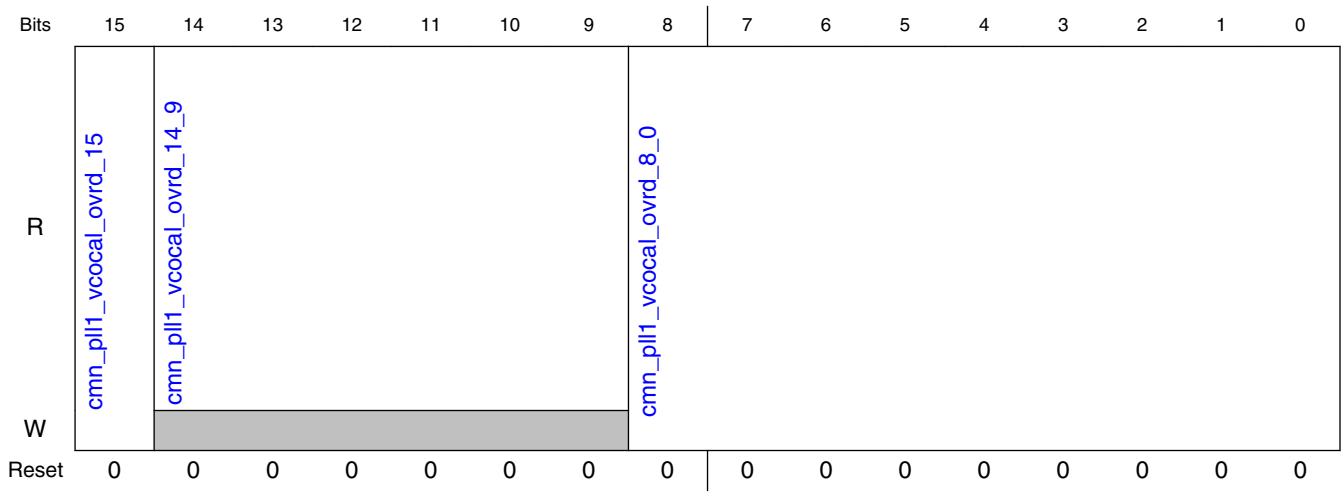
Field	Function
15-3 cmn_pll1_vcocal_tctrl_15_3	Reserved
2-0 cmn_pll1_vcocal_tctrl_2_0	VCO calibration initial time scale control: This field specifies the calibration start time scaling factor applied to the VCO calibration when running the initial step size for the calibration code is not set to 1. Setting this value to a value other than 1 will reduce the calibration measurement time by the amount specified below. Then when the final calibration steps are made the full calibration time will be used to get the appropriate amount of resolution.

13.4.10.2.458 PLL 1 VCO calibration override register (cmn_pll1_vcocal_ovrd)

13.4.10.2.458.1 Offset

Register	Offset
cmn_pll1_vcocal_ovrd	8_00A3h

13.4.10.2.458.2 Diagram



13.4.10.2.458.3 Fields

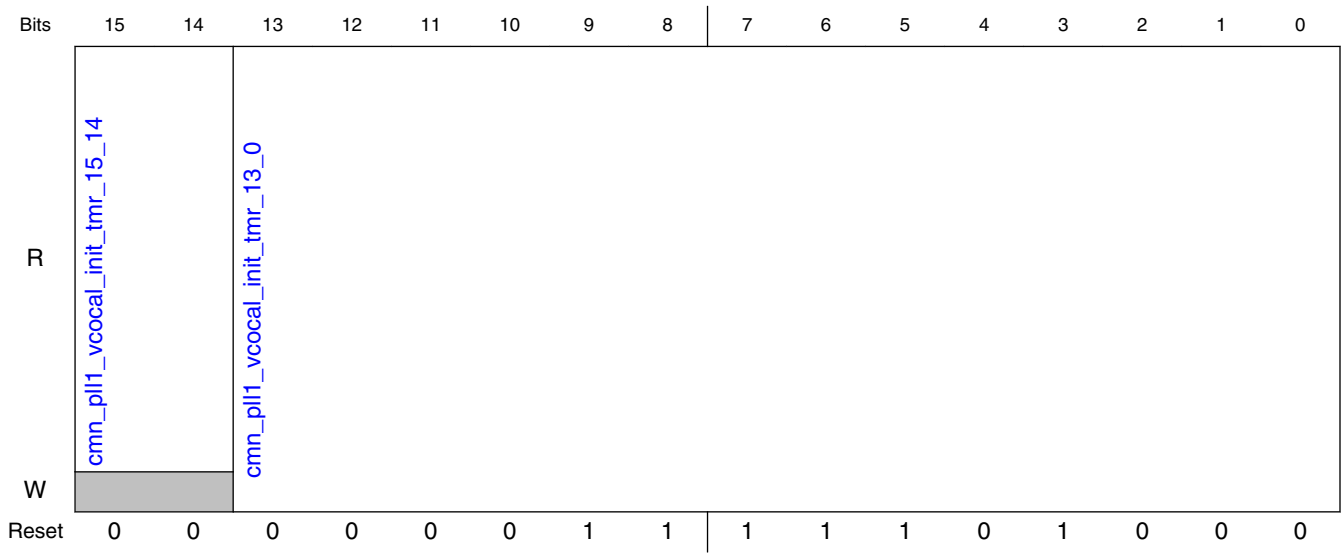
Field	Function
15 cmn_pll1_vcocal_ovrd_15	VCO calibration code override enable: Activating (1b1) this bit allows the VCO code determined during the automatic VCO calibration process to be overridden by the value driven by the VCO calibration code override value field in this register.
14-9 cmn_pll1_vcocal_ovrd_14_9	Reserved
8-0 cmn_pll1_vcocal_ovrd_8_0	VCO calibration code override value: This field is used to override the VCO code determined during the automatic VCO calibration process. The VCO code driven on this field is valid when the VCO calibration code override enable bit in this register is active.

13.4.10.2.459 PLL 1 VCO calibration initialization timer register (cmn_pll1_vcocal_init_tmr)

13.4.10.2.459.1 Offset

Register	Offset
cmn_pll1_vcocal_init_tmr	8_00A4h

13.4.10.2.459.2 Diagram



13.4.10.2.459.3 Fields

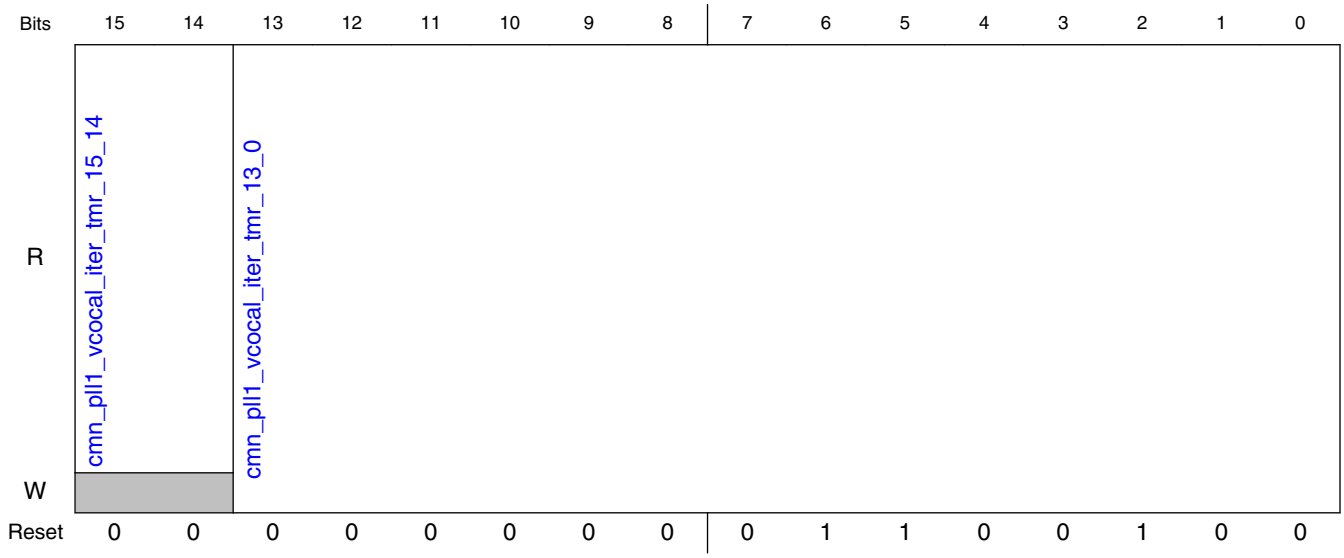
Field	Function
15-14 cmn_pll1_vcocal_init_tmr_15_14	Reserved
13-0 cmn_pll1_vcocal_init_tmr_13_0	Initialization wait timer value: This is the number of clocks to wait between when the analog VCO calibration circuits are enabled, and when the first calibration code is driven to the analog.

13.4.10.2.460 PLL 1 VCO calibration iteration timer register (cmn_pll1_vcocal_iter_tmr)

13.4.10.2.460.1 Offset

Register	Offset
cmn_pll1_vcocal_iter_tmr	8_00A5h

13.4.10.2.460.2 Diagram



13.4.10.2.460.3 Fields

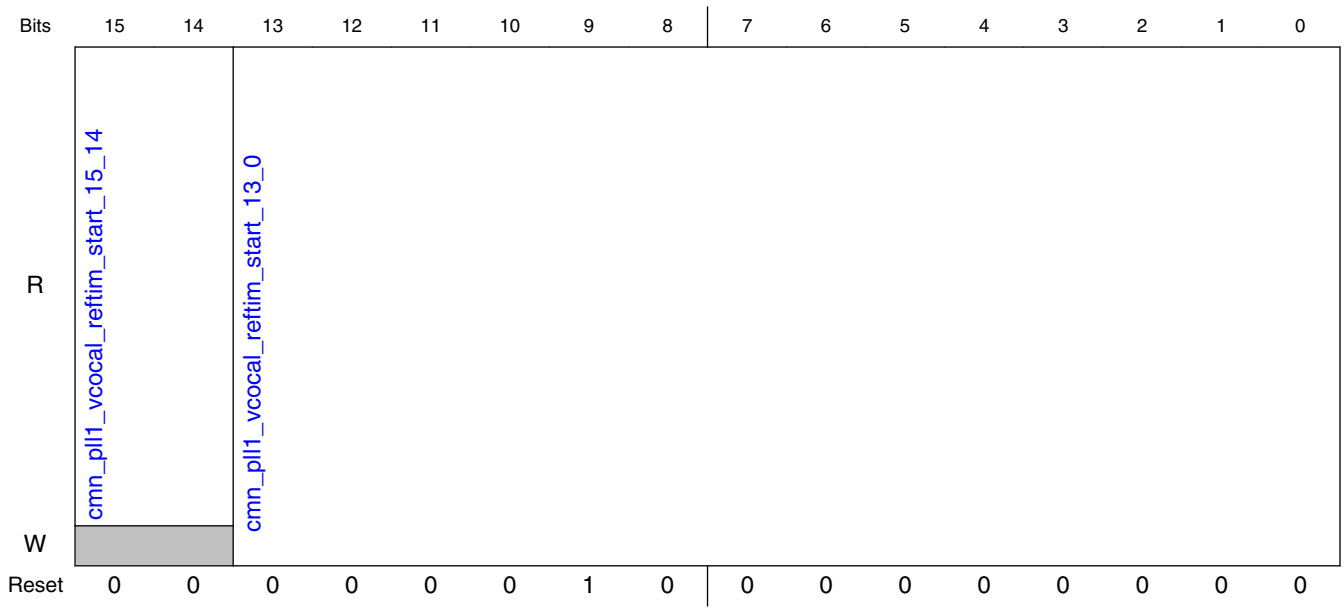
Field	Function
15-14 cmn_pll1_vcocal_iter_tmr_15_14	Reserved
13-0 cmn_pll1_vcocal_iter_tmr_13_0	Iteration wait timer value: This is the number of clocks to wait between when a calibration code is driven to the analog, and when the clock rates are measured.

13.4.10.2.461 PLL 1 VCO calibration reference clock timer start value register (cmn_pll1_vcocal_reftim_start)

13.4.10.2.461.1 Offset

Register	Offset
cmn_pll1_vcocal_reftim_start	8_00A6h

13.4.10.2.461.2 Diagram



13.4.10.2.461.3 Fields

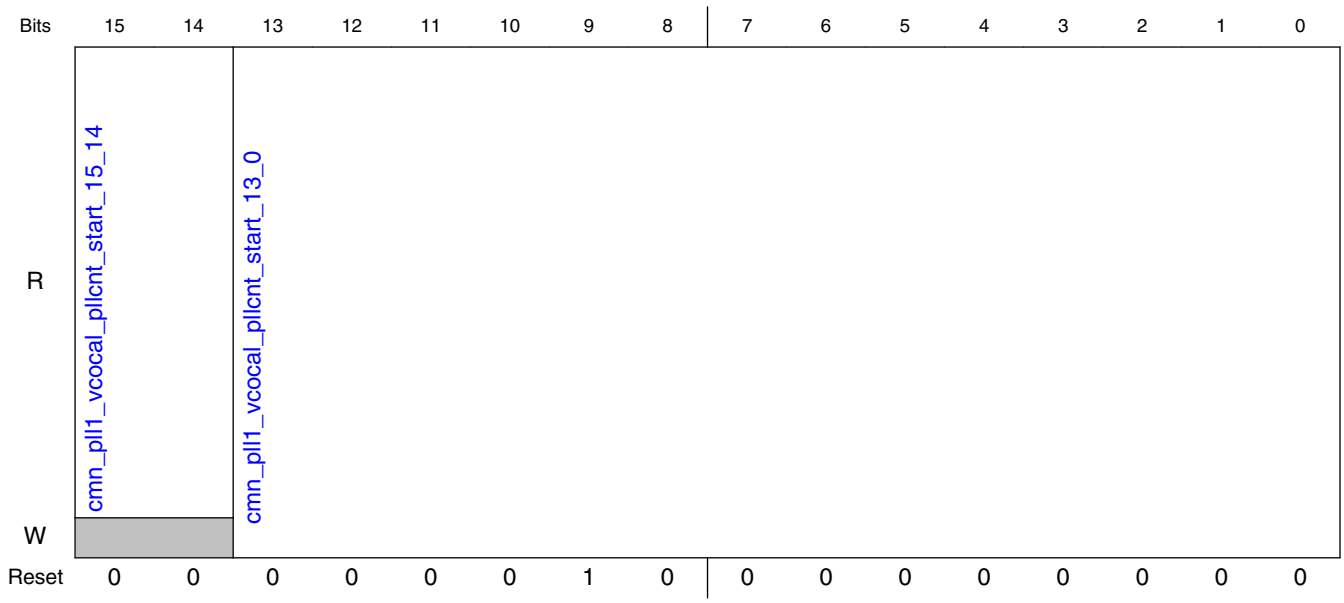
Field	Function
15-14 cmn_pll1_vcocal_reftim_start_15_14	Reserved
13-0 cmn_pll1_vcocal_reftim_start_13_0	PLL VCO calibration reference clock timer start value : This is the value that is loaded into the reference clock timer as the starting point for that timer, when running VCO calibration. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes.

13.4.10.2.462 PLL 1 VCO calibration PLL clock counter start value register (cmn_pll1_vcocal_pllcnt_start)

13.4.10.2.462.1 Offset

Register	Offset
cmn_pll1_vcocal_pllcnt_start	8_00A8h

13.4.10.2.462.2 Diagram



13.4.10.2.462.3 Fields

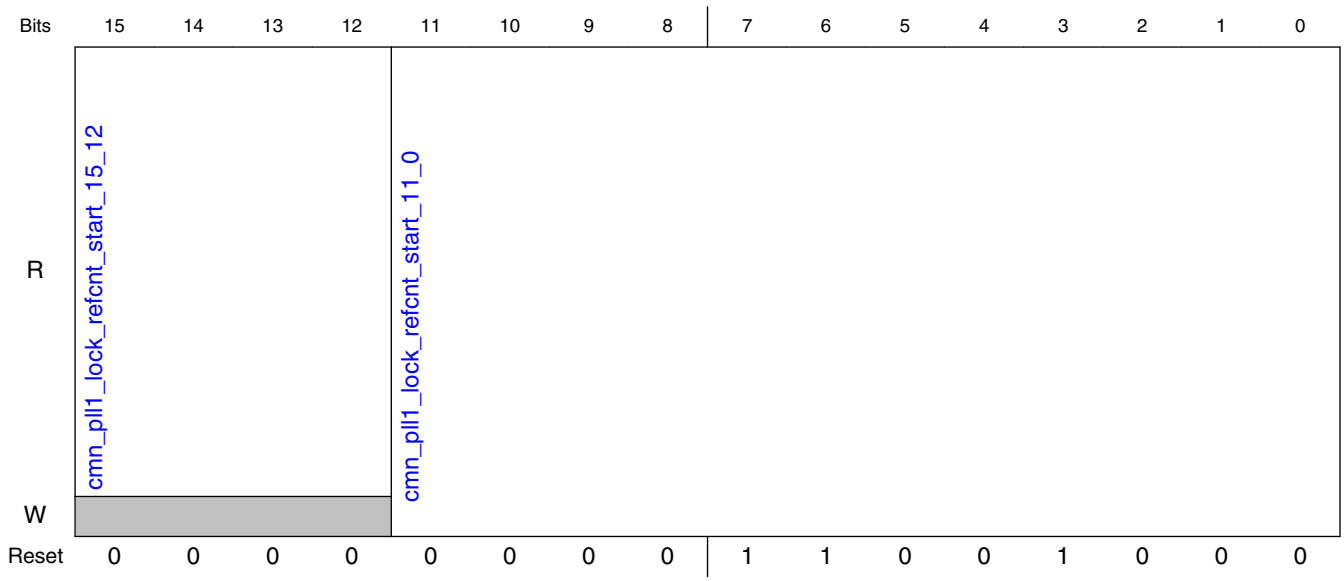
Field	Function
15-14 cmn_pll1_vcocal_pllcnt_start_15_14	Reserved
13-0 cmn_pll1_vcocal_pllcnt_start_13_0	PLL VCO calibration PLL clock counter start value : This is the value that is loaded into the PLL clock counter as the starting point for that counter, when running VCO calibration. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes.

13.4.10.2.463 PLL 1 lock reference counter start value register (cmn_pll1_lock_refcnt_start)

13.4.10.2.463.1 Offset

Register	Offset
cmn_pll1_lock_refcnt_start	8_00B0h

13.4.10.2.463.2 Diagram



13.4.10.2.463.3 Fields

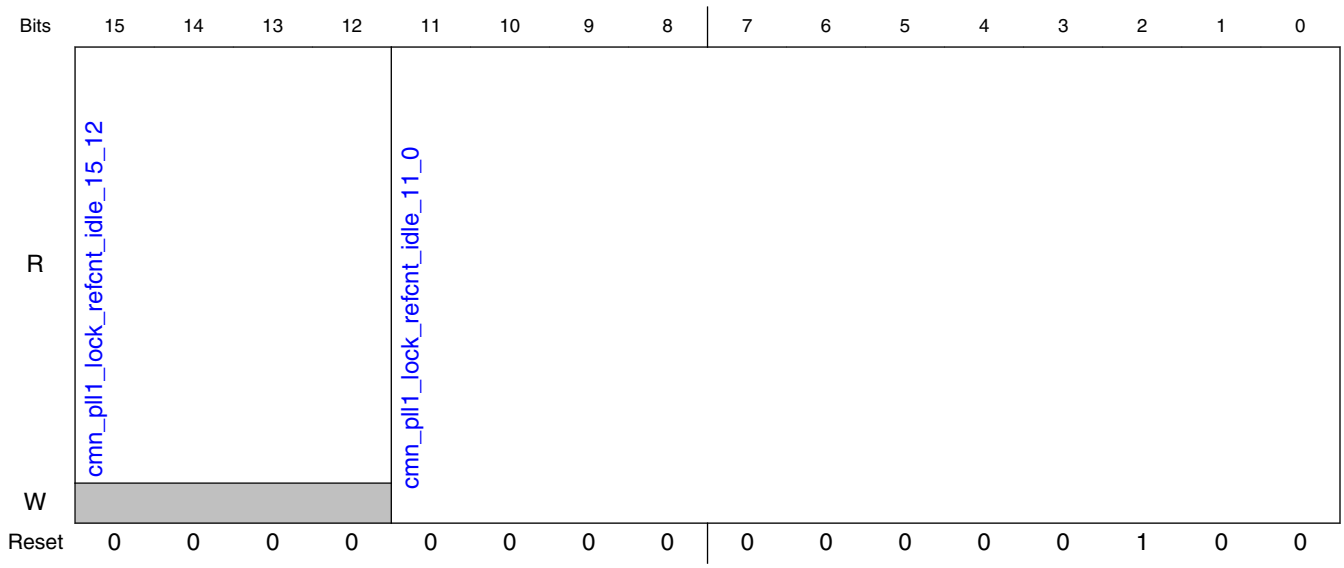
Field	Function
15-12 cmn_pll1_lock_refcnt_start_15_12	Reserved
11-0 cmn_pll1_lock_refcnt_start_11_0	PLL lock reference counter start value : This is the value that is loaded into the PLL lock detect reference counter as the starting point for that counter, when checking for PLL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. This is set to 200 clocks by default.

13.4.10.2.464 PLL 1 lock reference counter idle value register (cmn_pll1_lock_refcnt_idle)

13.4.10.2.464.1 Offset

Register	Offset
cmn_pll1_lock_refcnt_idle	8_00B1h

13.4.10.2.464.2 Diagram



13.4.10.2.464.3 Fields

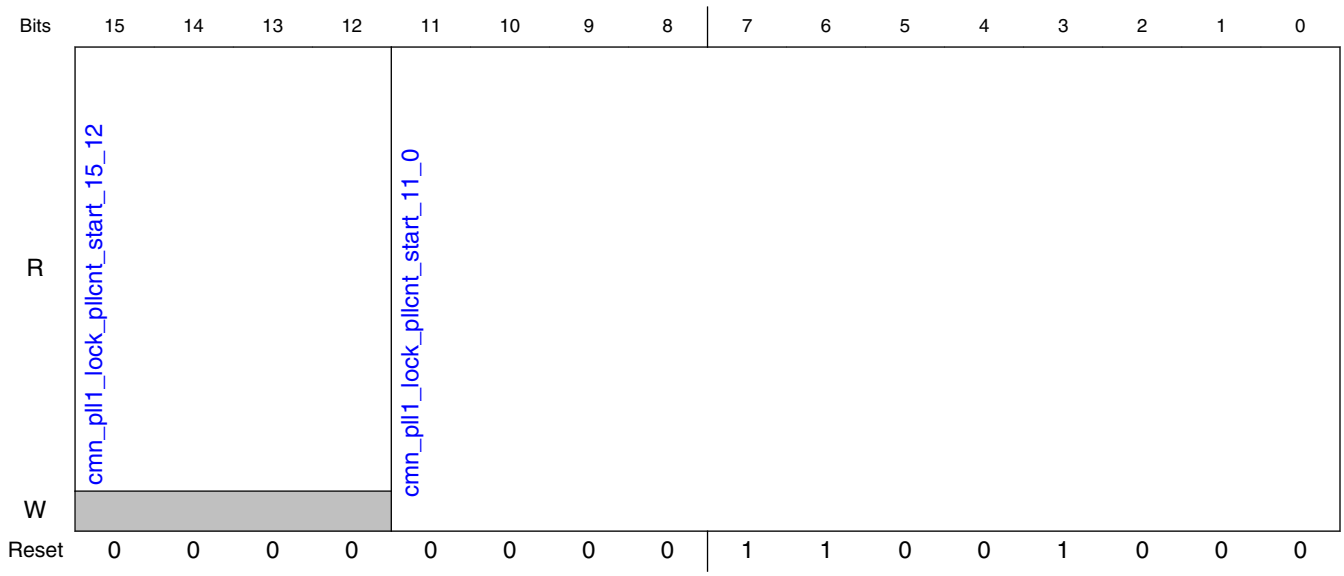
Field	Function
15-12 cmn_pll1_lock_refcnt_idle_15_12	Reserved
11-0 cmn_pll1_lock_refcnt_idle_11_0	PLL lock reference counter idle value : This is the value used by the PLL lock detection logic to specify the number of reference clocks between each phase of counting PLL clocks.

13.4.10.2.465 PLL 1 lock PLL counter start value register (cmn_pll1_lock_pllcnt_start)

13.4.10.2.465.1 Offset

Register	Offset
cmn_pll1_lock_pllcnt_start	8_00B2h

13.4.10.2.465.2 Diagram



13.4.10.2.465.3 Fields

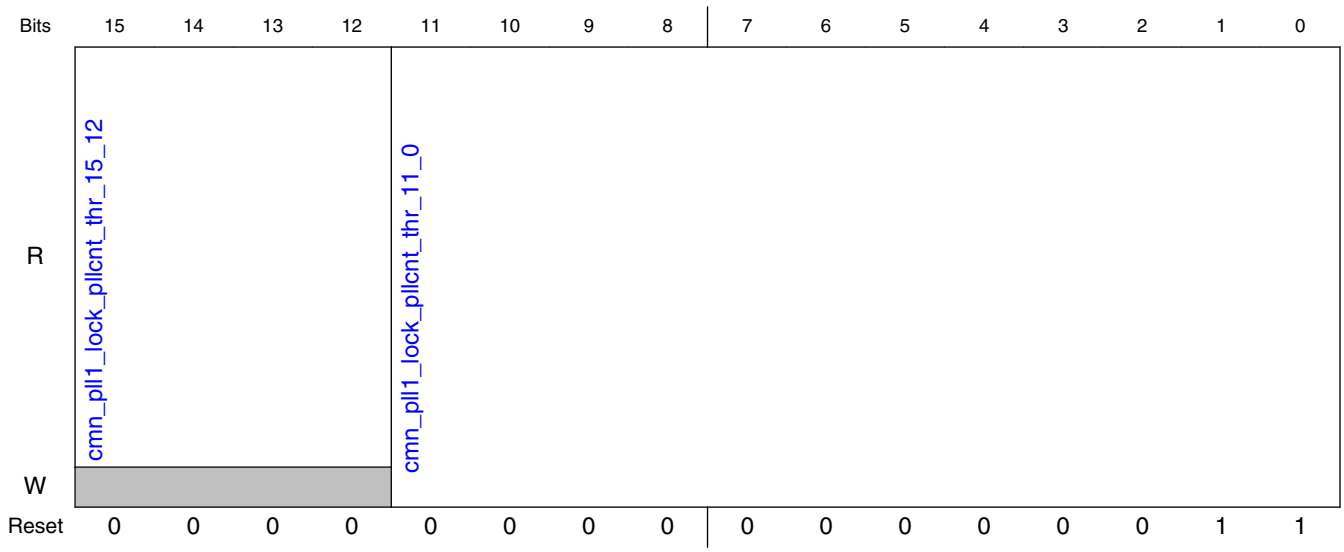
Field	Function
15-12 cmn_pll1_lock_pllcnt_start_15_12	Reserved
11-0 cmn_pll1_lock_pllcnt_start_11_0	PLL lock PLL counter start value : This is the value that is loaded into the PLL lock detect PLL counter as the starting point for that counter, when checking for PLL lock. This register is provided for simulation speedup and diagnostic purposes only. It is not intended to be for functional purposes. This is set to 200 clocks by default.

13.4.10.2.466 PLL 1 lock PLL counter threshold value register (cmn_pll1_lock_pllcnt_thr)

13.4.10.2.466.1 Offset

Register	Offset
cmn_pll1_lock_pllcnt_thr	8_00B3h

13.4.10.2.466.2 Diagram



13.4.10.2.466.3 Fields

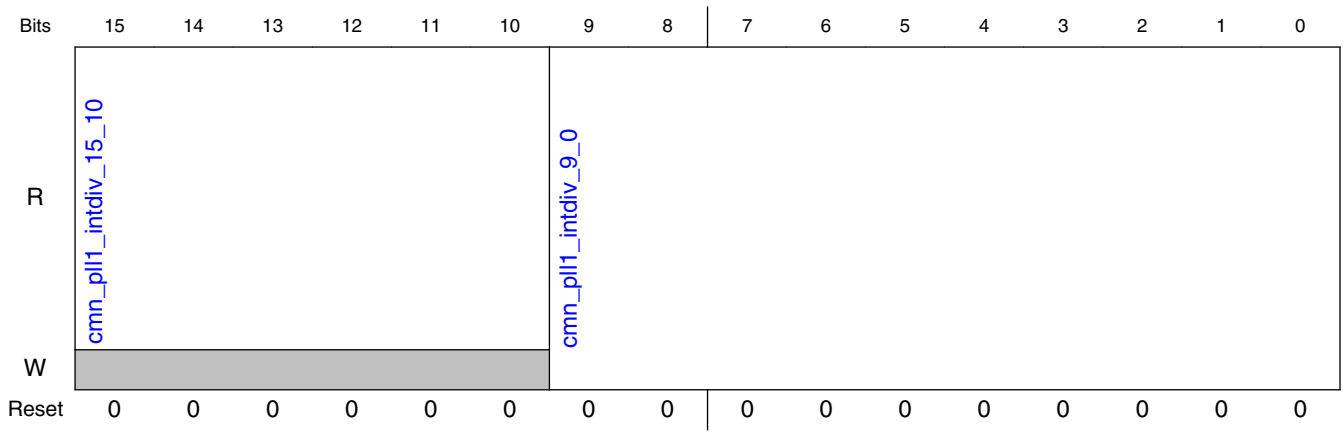
Field	Function
15-12 cmn_pll1_lock_pllcnt_thr_15_12	Reserved
11-0 cmn_pll1_lock_pllcnt_thr_11_0	PLL lock counter threshold value : This is the value used by the PLL lock detection logic to determine if the PLL has locked. If the two counters in the PLL lock detection logic differ by less than this value, the PLL is considered locked. This is set to 3 clocks by default.

13.4.10.2.467 PLL 1 feedback divider integer register (cmn_pll1_intdiv)

13.4.10.2.467.1 Offset

Register	Offset
cmn_pll1_intdiv	8_00B4h

13.4.10.2.467.2 Diagram



13.4.10.2.467.3 Fields

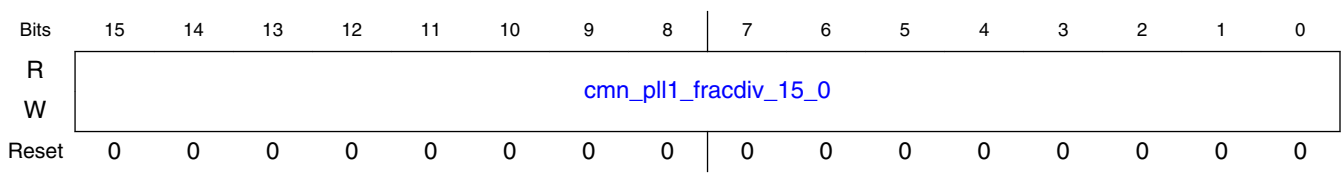
Field	Function
15-10 cmn_pll1_intdiv_15_10	Reserved
9-0 cmn_pll1_intdiv_9_0	pll_fb_div_integer value: Value of the pll_fb_div_integer signal.

13.4.10.2.468 PLL 1 feedback divider fractional register (cmn_pll1_fracdiv)

13.4.10.2.468.1 Offset

Register	Offset
cmn_pll1_fracdiv	8_00B5h

13.4.10.2.468.2 Diagram



13.4.10.2.468.3 Fields

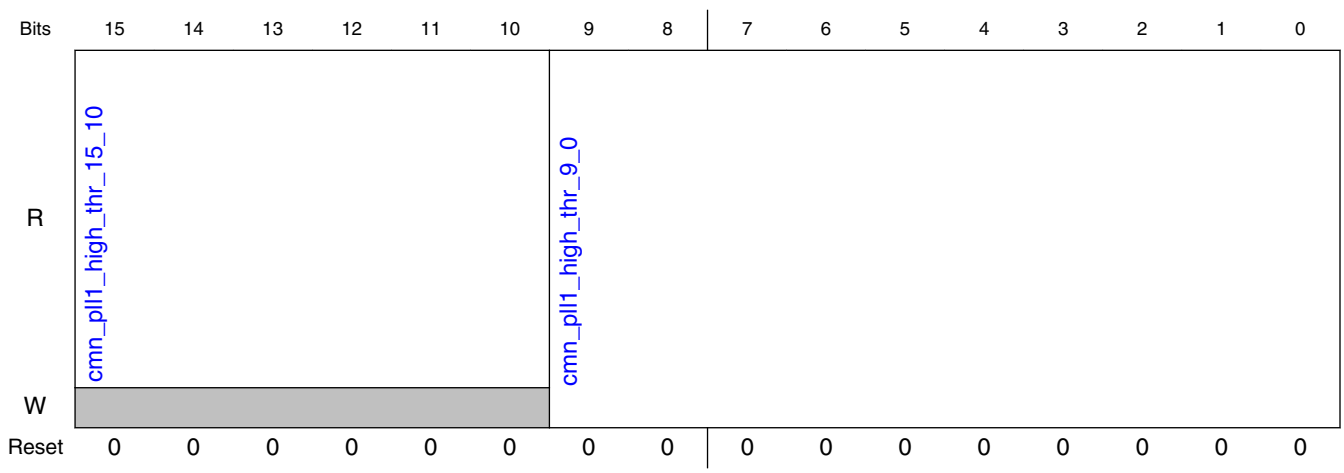
Field	Function
15-0 cmn_pll1_fracdiv_15_0	pll_fb_div_fractional: Value of the pll_fb_div_fractional signal.

13.4.10.2.469 PLL 1 feedback divider high threshold register (cmn_pll1_high_thr)

13.4.10.2.469.1 Offset

Register	Offset
cmn_pll1_high_thr	8_00B6h

13.4.10.2.469.2 Diagram



13.4.10.2.469.3 Fields

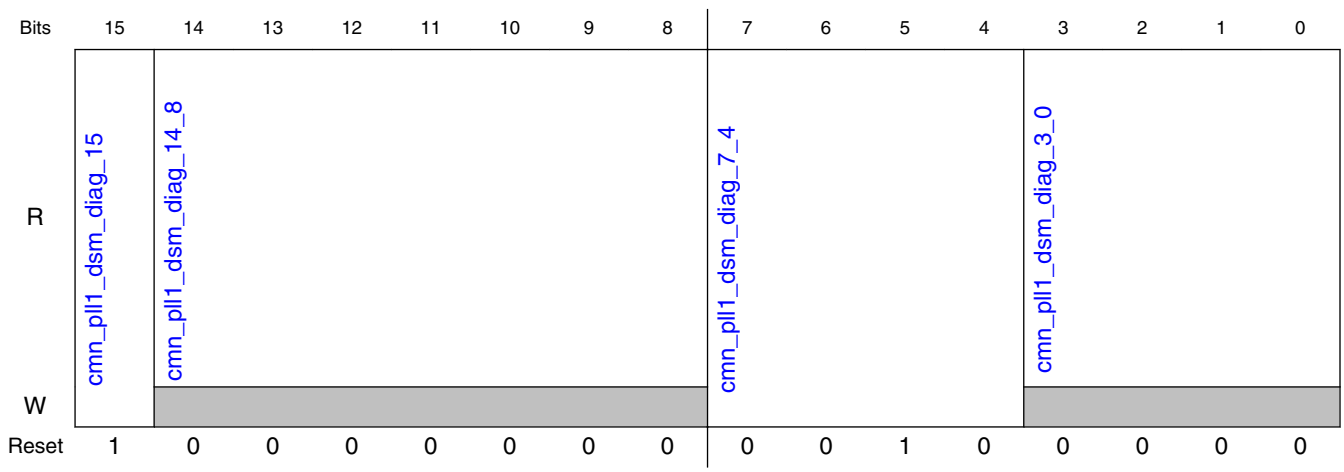
Field	Function
15-10 cmn_pll1_high_thr_15_10	Reserved
9-0 cmn_pll1_high_thr_9_0	pll_fb_div_high_theshold: Value of the pll_fb_div_high_threshold signal.

13.4.10.2.470 PLL 1 delta sigma modulator diagnostics register (cmn_pll1_dsm_diag)

13.4.10.2.470.1 Offset

Register	Offset
cmn_pll1_dsm_diag	8_00B7h

13.4.10.2.470.2 Diagram



13.4.10.2.470.3 Fields

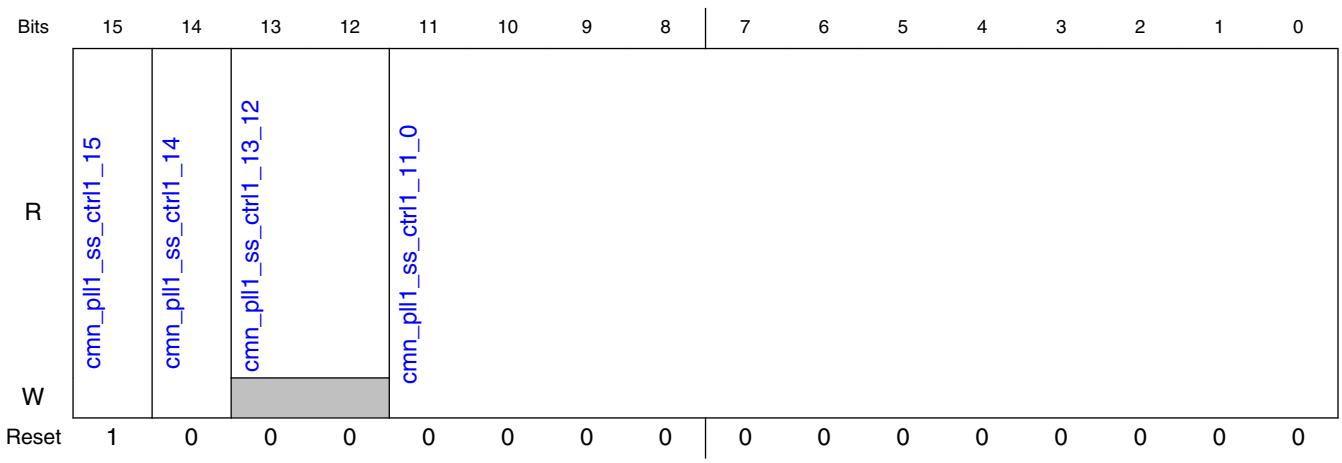
Field	Function
15 cmn_pll1_dsm_diag_15	Delta sigma bypass enable: When set to 1b1, the delta sigma modulator will be bypassed, and the output will be the value specified for the internal pll_fb_div_integer signal.
14-8 cmn_pll1_dsm_diag_14_8	Reserved
7-4 cmn_pll1_dsm_diag_7_4	PLL feedback divider latency adjustment: This signal specifies a value to be subtracted from the feedback divider settings before they are output on the
3-0 cmn_pll1_dsm_diag_3_0	Reserved

13.4.10.2.471 PLL 1 spread spectrum control register 1 (cmn_pll1_ss_ctrl 1)

13.4.10.2.471.1 Offset

Register	Offset
cmn_pll1_ss_ctrl1	8_00B8h

13.4.10.2.471.2 Diagram



13.4.10.2.471.3 Fields

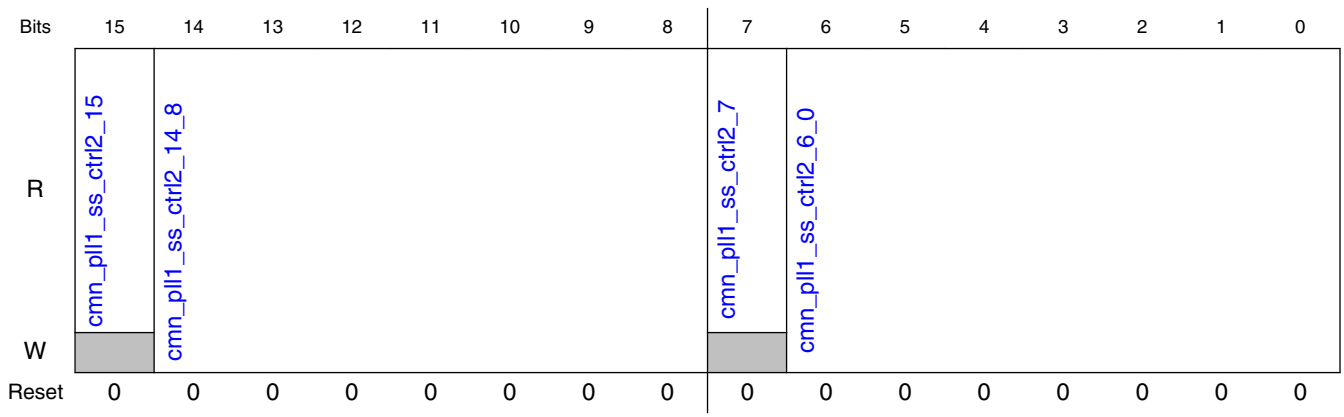
Field	Function
15 cmn_pll1_ss_ctrl 1_15	Spread spectrum waveform generator disable: Setting this bit to a 1b1 will disable the spread spectrum waveform generator.
14 cmn_pll1_ss_ctrl 1_14	Spread spectrum enable during VCO calibration : Setting this bit to a 1b1 will enable the spread spectrum function while VCO calibration is taking place.
13-12 cmn_pll1_ss_ctrl 1_13_12	Reserved
11-0 cmn_pll1_ss_ctrl 1_11_0	Amplitude step size: Value of the amplitude_step_size pin on the spread spectrum waveform generator.

13.4.10.2.472 PLL 1 spread spectrum control register 2 (cmn_pll1_ss_ctrl 2)

13.4.10.2.472.1 Offset

Register	Offset
cmn_pll1_ss_ctrl2	8_00B9h

13.4.10.2.472.2 Diagram



13.4.10.2.472.3 Fields

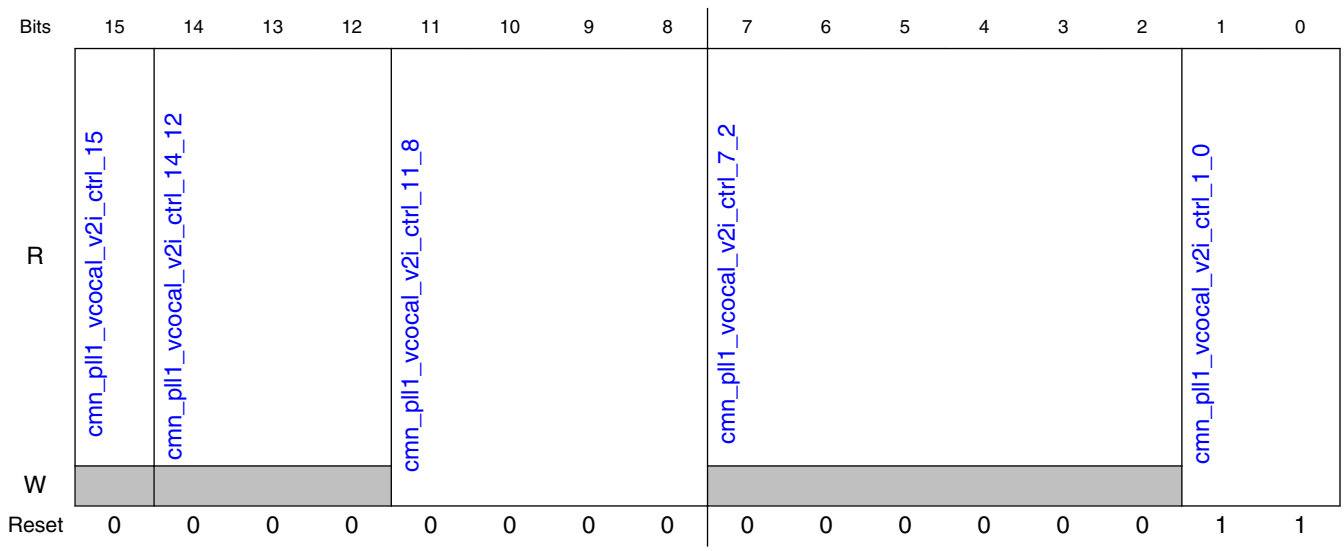
Field	Function
15 cmn_pll1_ss_ctrl 2_15	Reserved
14-8 cmn_pll1_ss_ctrl 2_14_8	Number of steps: Value of the num_steps pin on the spread spectrum waveform generator.
7 cmn_pll1_ss_ctrl 2_7	Reserved
6-0 cmn_pll1_ss_ctrl 2_6_0	Time step size: Value for the time_step_size pin on the spread spectrum waveform generator.

13.4.10.2.473 PLL 1 VCO calibration V2I control register (cmn_pll1_vcocal_v2i_ctrl)

13.4.10.2.473.1 Offset

Register	Offset
cmn_pll1_vcocal_v2i_ctrl	8_00BCh

13.4.10.2.473.2 Diagram



13.4.10.2.473.3 Fields

Field	Function
15 cmn_pll1_vcocal_v2i_ctrl_15	VCO calibration code for voltage to current DAC override enable: Activating (1b1) this bit allows the
14-12 cmn_pll1_vcocal_v2i_ctrl_14_12	Reserved
11-8 cmn_pll1_vcocal_v2i_ctrl_11_8	VCO calibration code for voltage to current DAC override value: This field is used to override the
7-2 cmn_pll1_vcocal_v2i_ctrl_7_2	Reserved

Table continues on the next page...

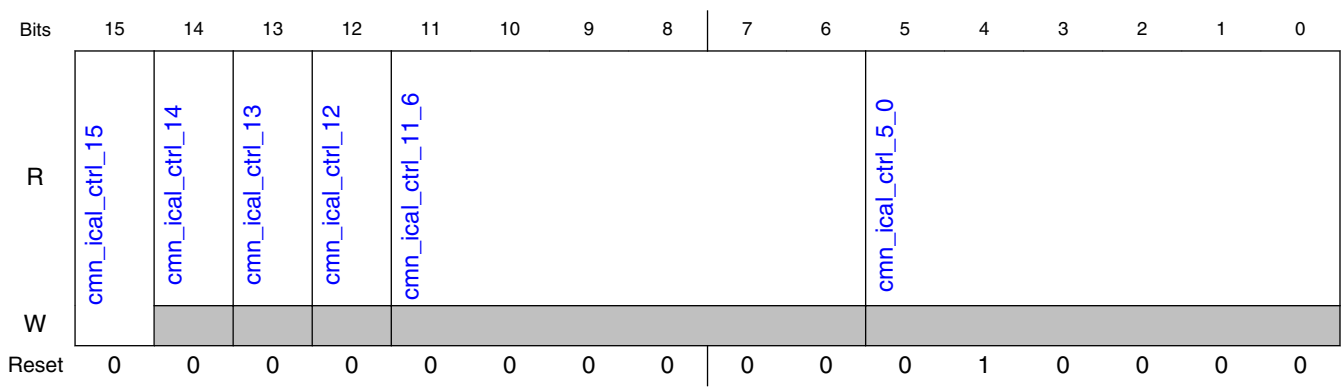
Field	Function
1-0 cmn_pll1_vcocal_v2i_ctrl_1_0	VCO calibration code for voltage to current DAC scaling factor: This field is used to control the scaling function in the logic used in the internal calculations related to the VCO calibration code for voltage to current DAC. The following is the encoding used for this field.

13.4.10.2.474 Current calibration control register (cmn_ical_ctrl)

13.4.10.2.474.1 Offset

Register	Offset
cmn_ical_ctrl	8_00C0h

13.4.10.2.474.2 Diagram



13.4.10.2.474.3 Fields

Field	Function
15 cmn_ical_ctrl_15	Start current calibration: Activating (1b1) this bit will start the current calibration process. This signal must remain active until the calibration process is complete. To start another calibration process, this register must first be set inactive (1b0) until the bandbap calibration process done bit in this register is cleared.
14 cmn_ical_ctrl_14	Current calibration process done: This bit will be set to 1b1 when the current calibration process is complete. It will be cleared by
13 cmn_ical_ctrl_13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.
12	Current analog comparator response: This is the current state of the analog comparator response signal (

Table continues on the next page...

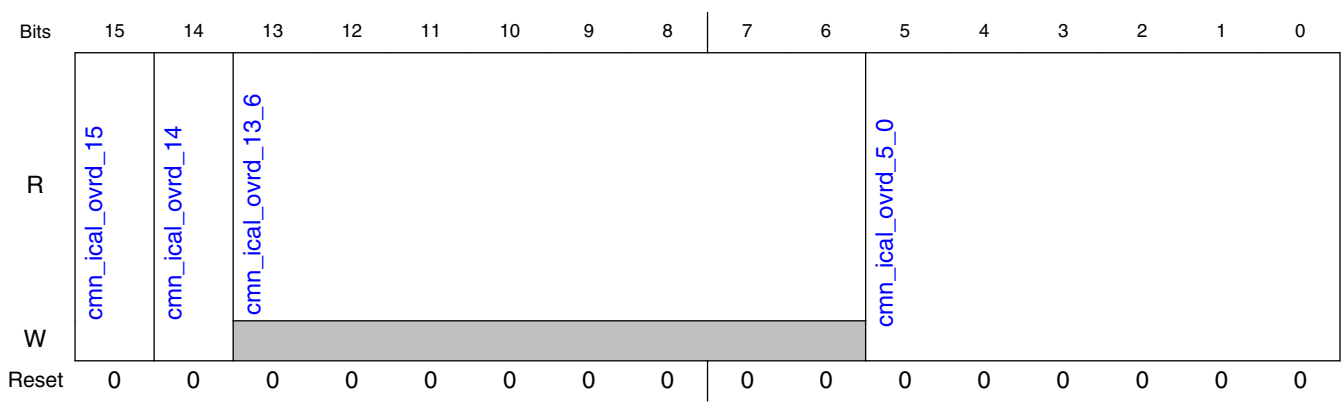
Field	Function
cmn_ical_ctrl_12	
11-6 cmn_ical_ctrl_11_6	Reserved
5-0 cmn_ical_ctrl_5_0	Current calibration code: This is the calibration code that was determined by the current calibration process. The following are the values for the code.

13.4.10.2.475 Current calibration override register (cmn_ical_ovrd)

13.4.10.2.475.1 Offset

Register	Offset
cmn_ical_ovrd	8_00C1h

13.4.10.2.475.2 Diagram



13.4.10.2.475.3 Fields

Field	Function
15 cmn_ical_ovrd_15	Current code override enable: Activation (1b1) of this register bit allows the current codes determined during the automatic calibration process to be overridden. The override value is specified using the current code override value field of this register.
14 cmn_ical_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the

Table continues on the next page...

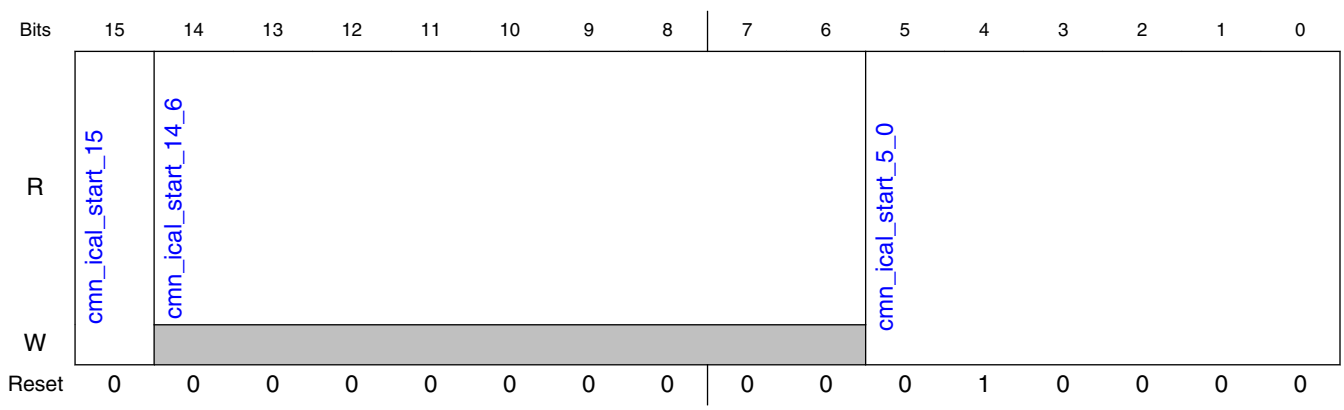
Field	Function
13-6 cmn_ical_ovrd_13_6	Reserved
5-0 cmn_ical_ovrd_5_0	Current code override value: These bits are used to override the current code determined during the automatic calibration process. The code written to these bits is valid when the current code override enable bit in this register is active.

13.4.10.2.476 Current calibration start register (cmn_ical_start)

13.4.10.2.476.1 Offset

Register	Offset
cmn_ical_start	8_00C2h

13.4.10.2.476.2 Diagram



13.4.10.2.476.3 Fields

Field	Function
15 cmn_ical_start_15	Current calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-6 cmn_ical_start_14_6	Reserved
5-0	Start current calibration code: This is the calibration code that the calibration process starts with when automatic calibration is run.

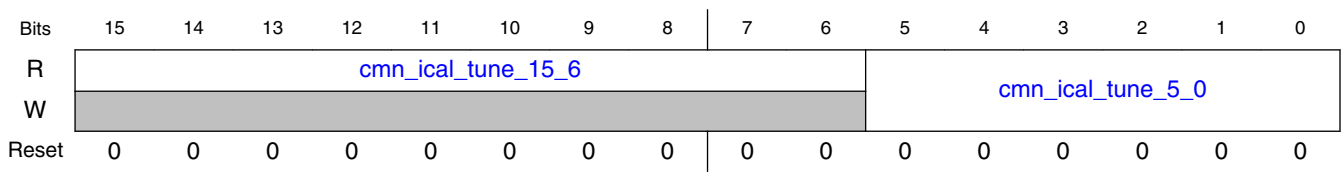
Field	Function
cmn_ical_start_5_0	

13.4.10.2.477 Current calibration tune register (cmn_ical_tune)

13.4.10.2.477.1 Offset

Register	Offset
cmn_ical_tune	8_00C3h

13.4.10.2.477.2 Diagram



13.4.10.2.477.3 Fields

Field	Function
15-6 cmn_ical_tune_15_6	Reserved
5-0 cmn_ical_tune_5_0	Current calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.478 Current calibration initialization timer register (cmn_ical_init_tmr)

13.4.10.2.478.1 Offset

Register	Offset
cmn_ical_init_tmr	8_00C4h

13.4.10.2.478.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_ical_init_tmr_15_8								cmn_ical_init_tmr_7_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1

13.4.10.2.478.3 Fields

Field	Function
15-8 cmn_ical_init_tmr_15_8	Reserved
7-0 cmn_ical_init_tmr_7_0	Initialization wait timer value: This is the number of

13.4.10.2.479 Current calibration iteration timer register (cmn_ical_iter_tmr)

13.4.10.2.479.1 Offset

Register	Offset
cmn_ical_iter_tmr	8_00C5h

13.4.10.2.479.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	cmn_ical_iter_tmr_15_8								cmn_ical_iter_tmr_7_0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0

13.4.10.2.479.3 Fields

Field	Function
15-8	Reserved

Table continues on the next page...

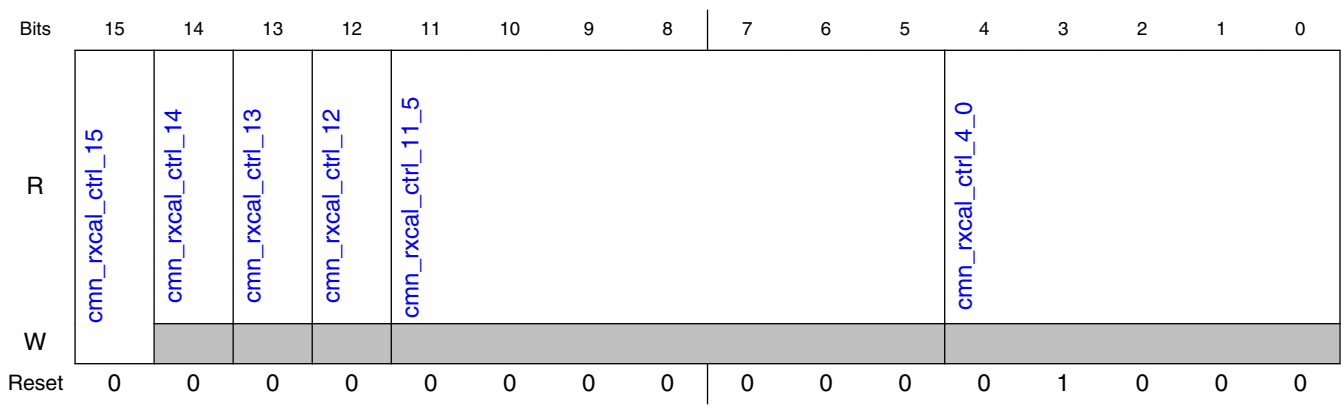
Field	Function
cmn_ical_iter_tmr_15_8	
7-0 cmn_ical_iter_tmr_7_0	Iteration wait timer value: This is the number of

13.4.10.2.480 RX resistor calibration control register (cmn_rxcal_ctrl)

13.4.10.2.480.1 Offset

Register	Offset
cmn_rxcal_ctrl	8_00D0h

13.4.10.2.480.2 Diagram



13.4.10.2.480.3 Fields

Field	Function
15 cmn_rxcal_ctrl_15	Start resistor calibration: Activating (1b1) this bit will start the resistor calibration process. This signal must remain active until the resistor calibration process is complete. To start another resistor calibration process, this register must first be set inactive (1b0) until the resistor calibration process done bit in this register is cleared.
14 cmn_rxcal_ctrl_14	Resistor calibration process done: This bit will be set to 1b1 when the resistor calibration process is complete. It will be cleared by
13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.

Table continues on the next page...

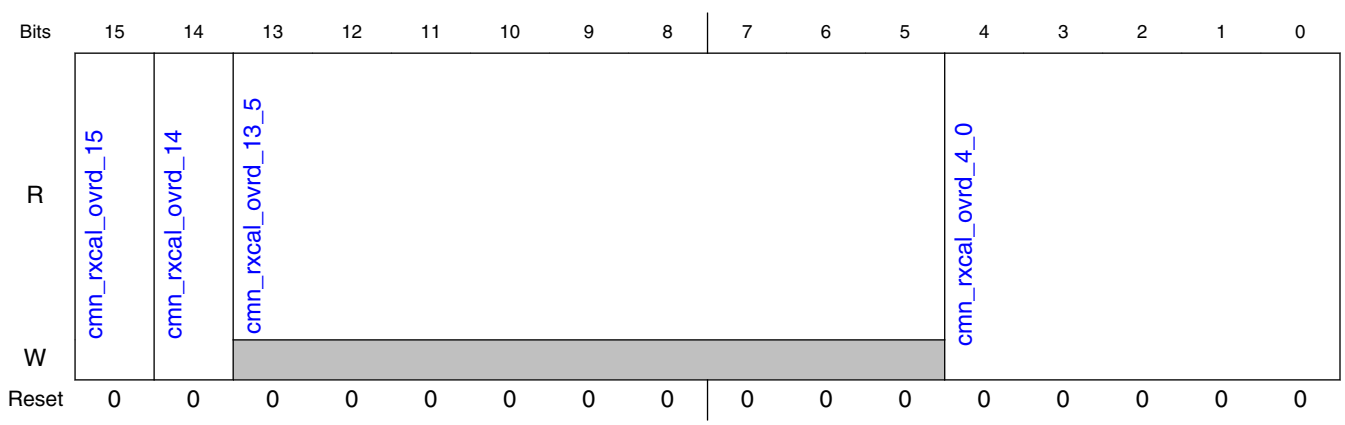
Field	Function
cmn_rxcal_ctrl_13	
12 cmn_rxcal_ctrl_12	Current analog comparator response: This is the current state of the analog comparator response signal (
11-5 cmn_rxcal_ctrl_11_5	Reserved
4-0 cmn_rxcal_ctrl_4_0	Resistor calibration code: This is the calibration code that was determined by the resistor calibration process. The following are the values for the code:

13.4.10.2.481 RX resistor calibration override register (cmn_rxcal_ovrd)

13.4.10.2.481.1 Offset

Register	Offset
cmn_rxcal_ovrd	8_00D1h

13.4.10.2.481.2 Diagram



13.4.10.2.481.3 Fields

Field	Function
15	Resistor code override enable: Activation (1b1) of this register bit allows the resistor codes determined during the automatic resistor calibration process to be overridden. The override value is specified using the resistor code override value field of this register.

Table continues on the next page...

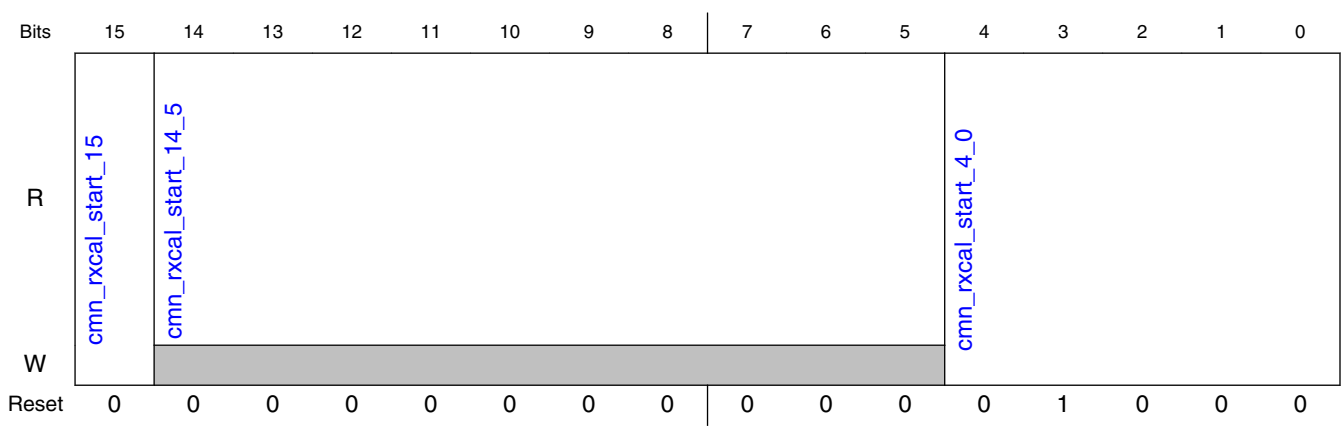
Field	Function
cmn_rxcal_ovrd_15	
14 cmn_rxcal_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-5 cmn_rxcal_ovrd_13_5	Reserved
4-0 cmn_rxcal_ovrd_4_0	Resistor code override value: These bits are used to override the resistor code determined during the automatic resistor calibration process. The resistor code written to these bits is valid when the resistor code override enable bit in this register is active. The following are the values for the code:

13.4.10.2.482 RX resistor calibration start register (cmn_rxcal_start)

13.4.10.2.482.1 Offset

Register	Offset
cmn_rxcal_start	8_00D2h

13.4.10.2.482.2 Diagram



13.4.10.2.482.3 Fields

Field	Function
15	Resistor calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.

Table continues on the next page...

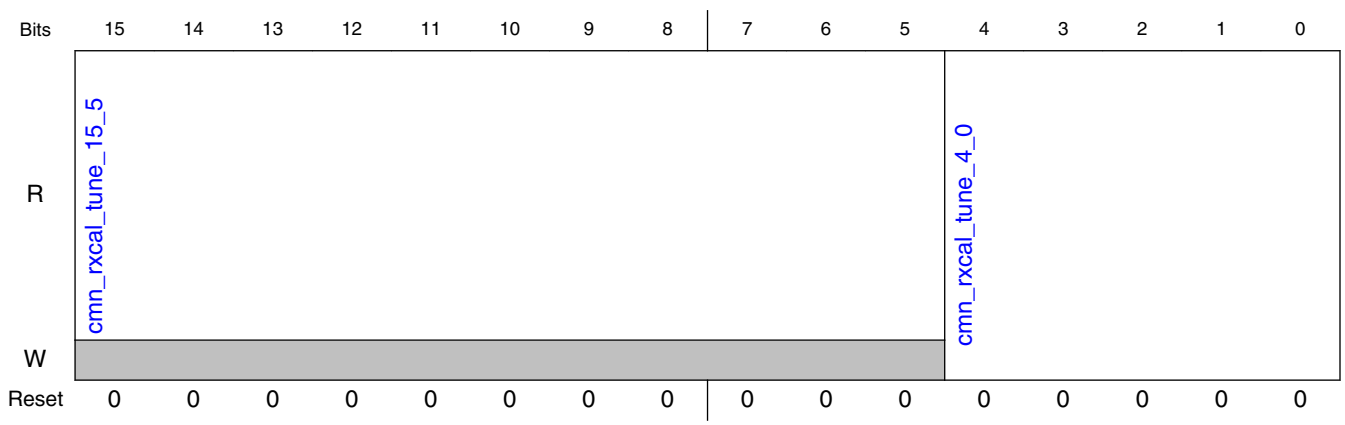
Field	Function
cmn_rxcal_start_15	
14-5 cmn_rxcal_start_14_5	Reserved
4-0 cmn_rxcal_start_4_0	Start resistor calibration code: This is the calibration code that the resistor calibration process starts with when automatic calibration is run. The following are the values for the code.

13.4.10.2.483 RX resistor calibration tune register (cmn_rxcal_tune)

13.4.10.2.483.1 Offset

Register	Offset
cmn_rxcal_tune	8_00D3h

13.4.10.2.483.2 Diagram



13.4.10.2.483.3 Fields

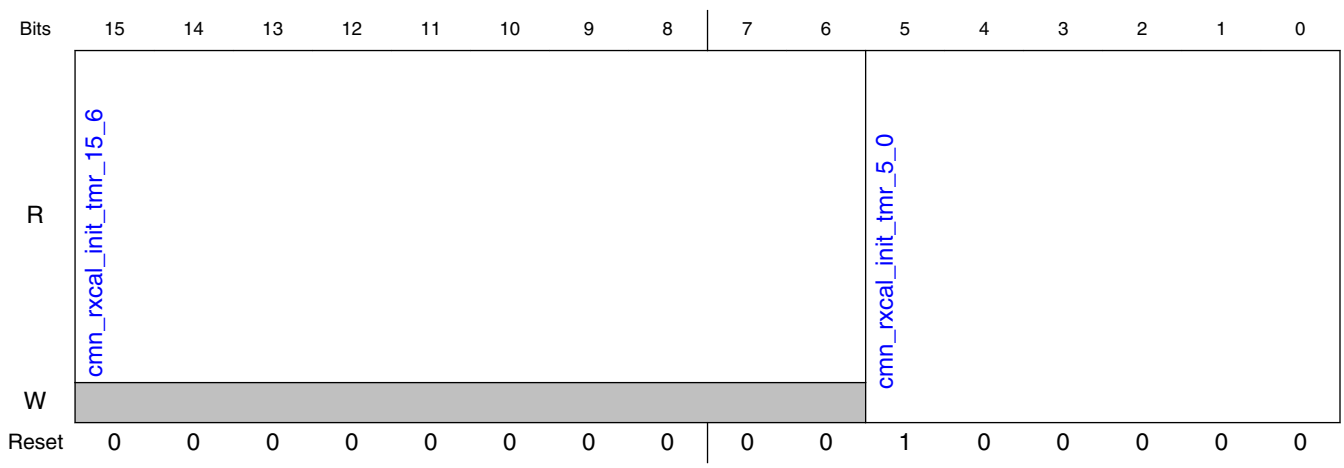
Field	Function
15-5 cmn_rxcal_tune_15_5	Reserved
4-0 cmn_rxcal_tune_4_0	Resistor calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.484 RX resistor calibration initialization timer register (cmn_rxcal_init_tmr)

13.4.10.2.484.1 Offset

Register	Offset
cmn_rxcal_init_tmr	8_00D4h

13.4.10.2.484.2 Diagram



13.4.10.2.484.3 Fields

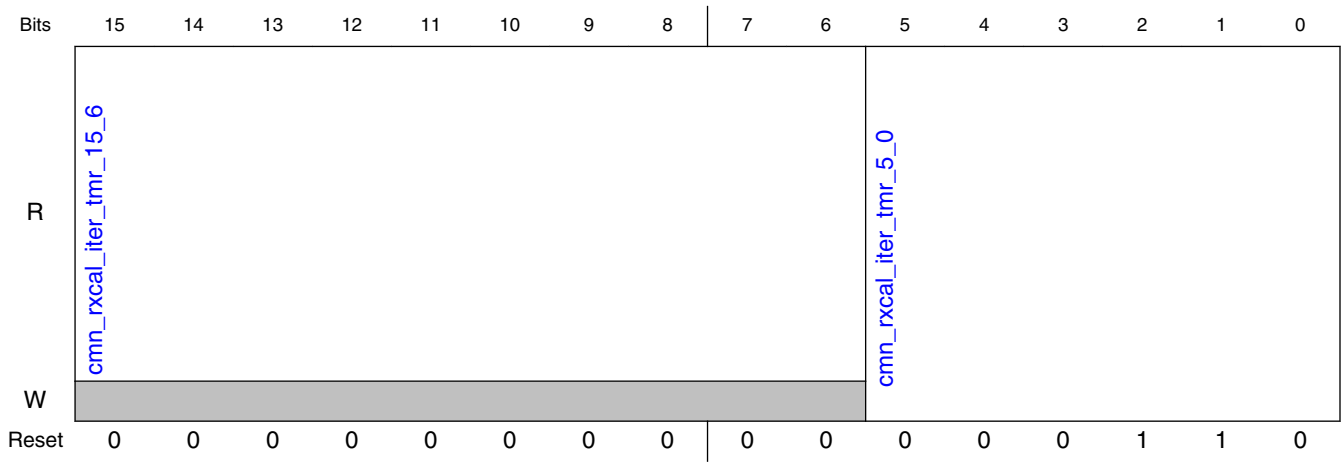
Field	Function
15-6 cmn_rxcal_init_tmr_15_6	Reserved
5-0 cmn_rxcal_init_tmr_5_0	Initialization wait timer value: This is the number of

13.4.10.2.485 RX resistor calibration iteration timer register (cmn_rxcal_iter_tmr)

13.4.10.2.485.1 Offset

Register	Offset
cmn_rxcal_iter_tmr	8_00D5h

13.4.10.2.485.2 Diagram



13.4.10.2.485.3 Fields

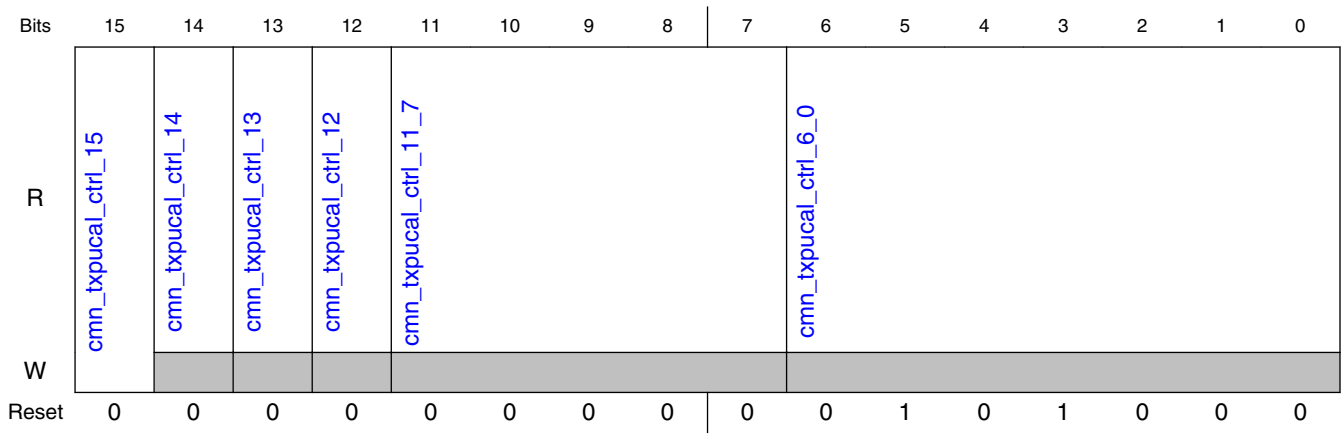
Field	Function
15-6 cmn_rxcal_iter_tmr_15_6	Reserved
5-0 cmn_rxcal_iter_tmr_5_0	Iteration wait timer value: This is the number of

13.4.10.2.486 TX pull-up resistor calibration control register (cmn_txpucal_ctrl)

13.4.10.2.486.1 Offset

Register	Offset
cmn_txpucal_ctrl	8_00E0h

13.4.10.2.486.2 Diagram



13.4.10.2.486.3 Fields

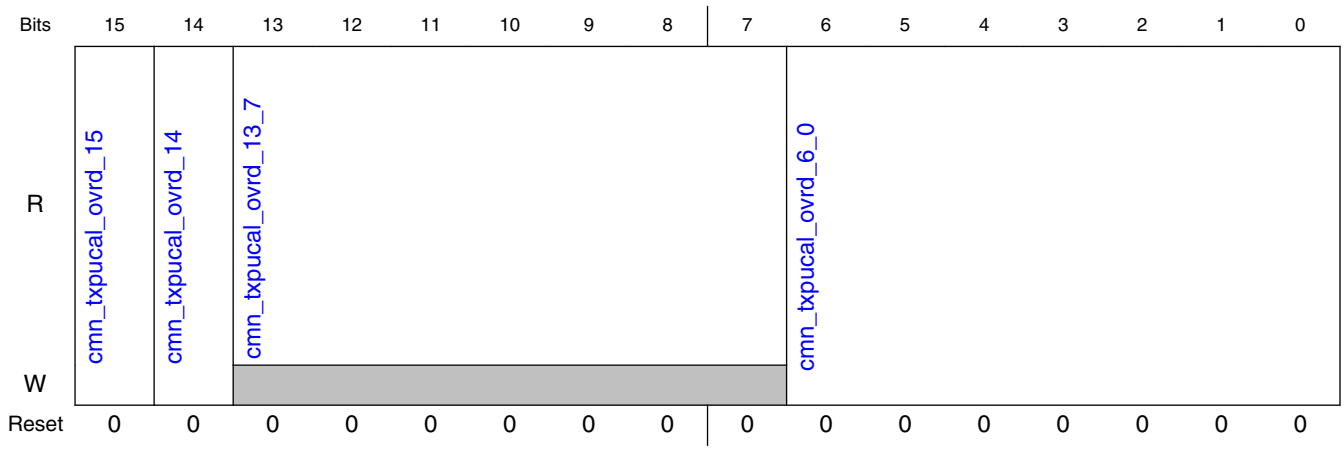
Field	Function
15 cmn_txpucal_ctrl_15	Start resistor calibration: Activating (1b1) this bit will start the resistor calibration process. This signal must remain active until the resistor calibration process is complete. To start another resistor calibration process, this register must first be set inactive (1b0) until the resistor calibration process done bit in this register is cleared.
14 cmn_txpucal_ctrl_14	Resistor calibration process done: This bit will be set to 1b1 when the resistor calibration process is complete. It will be cleared by
13 cmn_txpucal_ctrl_13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.
12 cmn_txpucal_ctrl_12	Current analog comparator response: This is the current state of the analog comparator response signal (
11-7 cmn_txpucal_ctrl_11_7	Reserved
6-0 cmn_txpucal_ctrl_6_0	Resistor calibration code: This is the calibration code that was determined by the resistor calibration process. The following are the values for the code.

13.4.10.2.487 TX pull-up resistor calibration override register (cmn_txpucal_ovrd)

13.4.10.2.487.1 Offset

Register	Offset
cmn_txpucal_ovrd	8_00E1h

13.4.10.2.487.2 Diagram



13.4.10.2.487.3 Fields

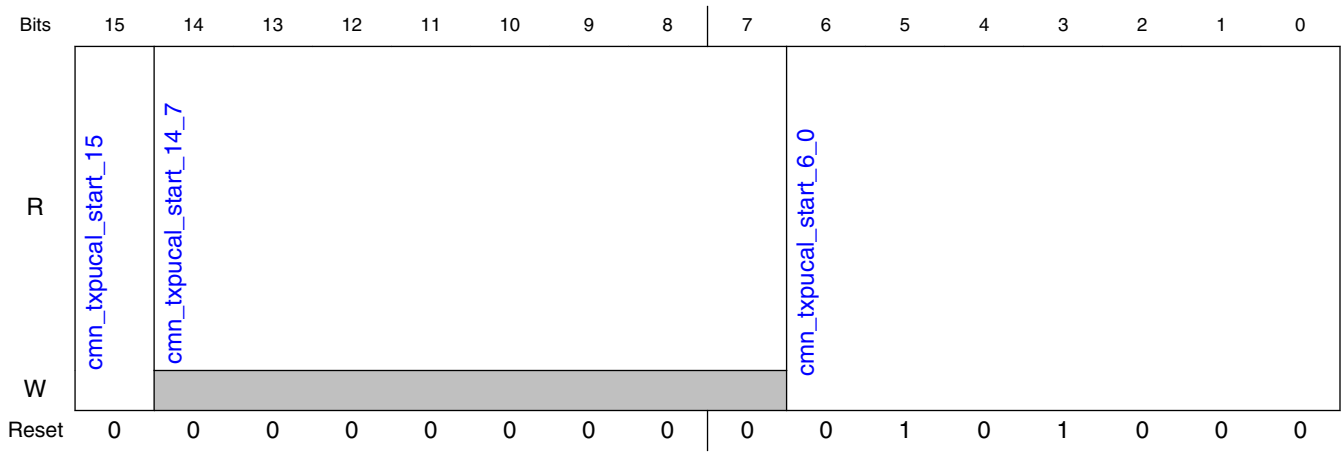
Field	Function
15 cmn_txpucal_ovrd_15	Resistor code override enable: Activation (1b1) of this register bit allows the resistor codes determined during the automatic resistor calibration process to be overridden. The override value is specified using the resistor code override value field of this register.
14 cmn_txpucal_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-7 cmn_txpucal_ovrd_13_7	Reserved
6-0 cmn_txpucal_ovrd_6_0	Resistor code override value: These bits are used to override the resistor code determined during the automatic resistor calibration process. The resistor code written to these bits is valid when the resistor code override enable bit in this register is active.

13.4.10.2.488 TX pull-up resistor calibration start register (cmn_txpucal_start)

13.4.10.2.488.1 Offset

Register	Offset
cmn_txpucal_start	8_00E2h

13.4.10.2.488.2 Diagram



13.4.10.2.488.3 Fields

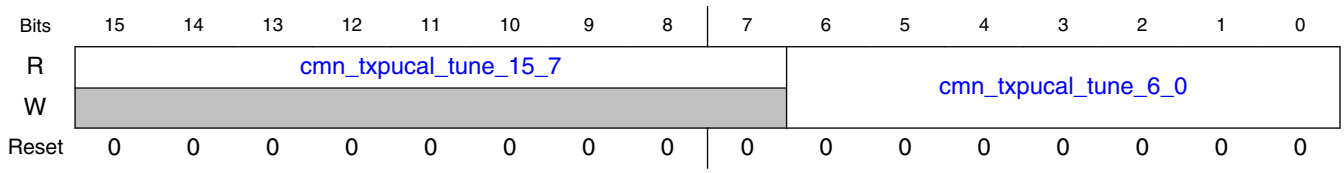
Field	Function
15 cmn_txpucal_start_15	Resistor calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-7 cmn_txpucal_start_14_7	Reserved
6-0 cmn_txpucal_start_6_0	Start resistor calibration code: This is the calibration code that the resistor calibration process starts with when automatic calibration is run. The following are the values for the code.

13.4.10.2.489 TX pull-up resistor calibration tune register (cmn_txpucal_tune)

13.4.10.2.489.1 Offset

Register	Offset
cmn_txpucal_tune	8_00E3h

13.4.10.2.489.2 Diagram



13.4.10.2.489.3 Fields

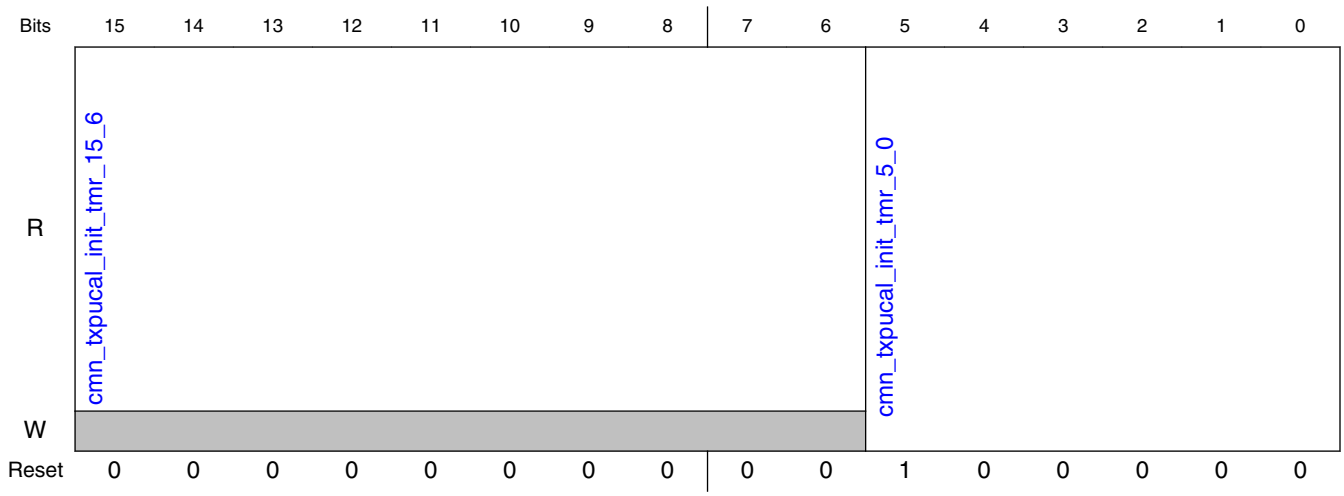
Field	Function
15-7 cmn_txpucal_tune_15_7	Reserved
6-0 cmn_txpucal_tune_6_0	Resistor calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.490 TX pull-up resistor calibration initialization timer register (cmn_txpucal_init_tmr)

13.4.10.2.490.1 Offset

Register	Offset
cmn_txpucal_init_tmr	8_00E4h

13.4.10.2.490.2 Diagram



13.4.10.2.490.3 Fields

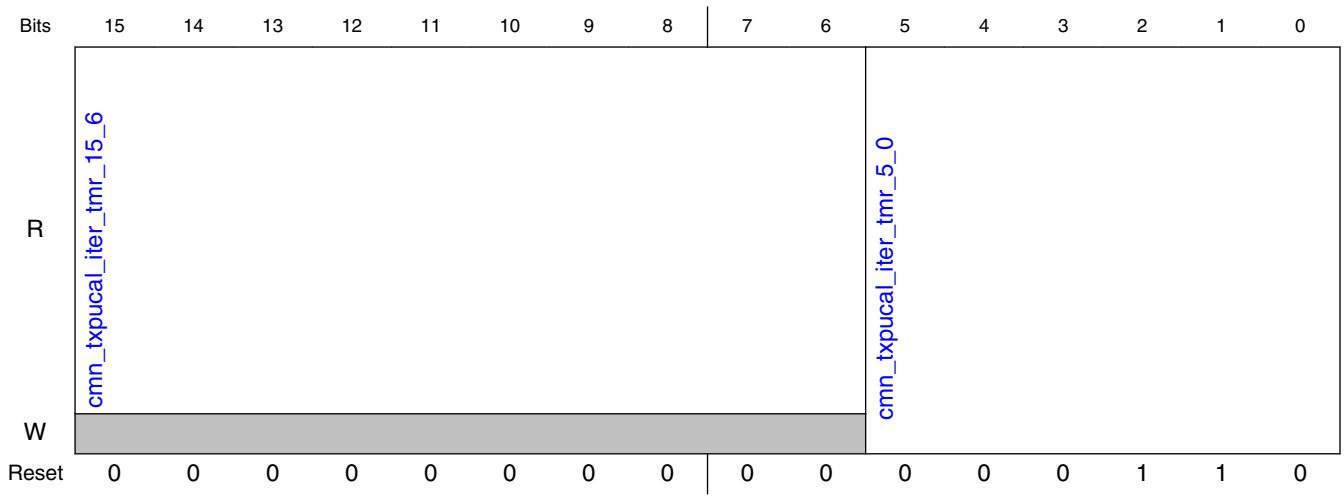
Field	Function
15-6 cmn_txpucal_init_tmr_15_6	Reserved
5-0 cmn_txpucal_init_tmr_5_0	Initialization wait timer value: This is the number of

13.4.10.2.491 TX pull-up resistor calibration iteration timer register (cmn_txpucal_iter_tmr)

13.4.10.2.491.1 Offset

Register	Offset
cmn_txpucal_iter_tmr	8_00E5h

13.4.10.2.491.2 Diagram



13.4.10.2.491.3 Fields

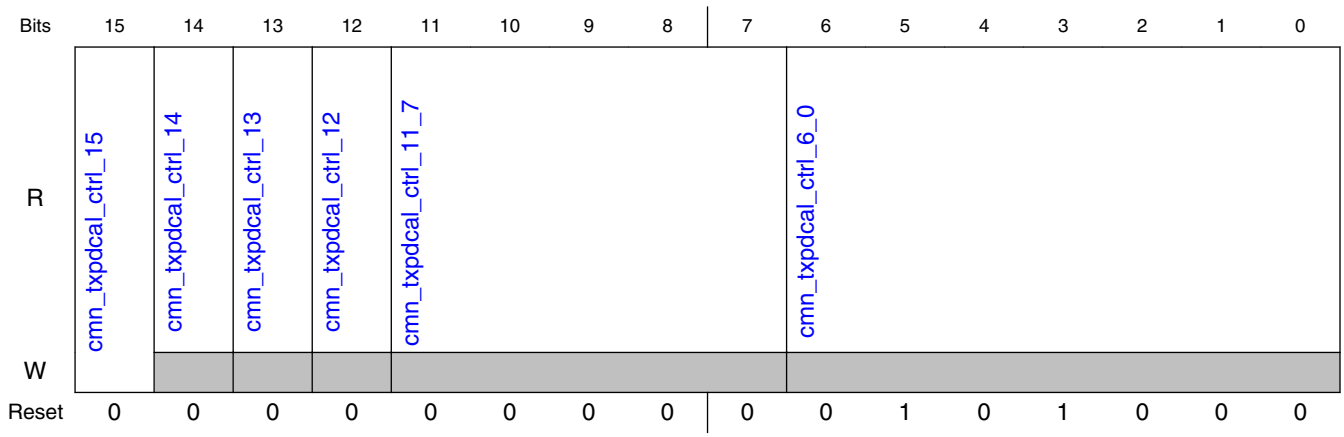
Field	Function
15-6 cmn_txpucal_iter_tmr_15_6	Reserved
5-0 cmn_txpucal_iter_tmr_5_0	Iteration wait timer value: This is the number of

13.4.10.2.492 TX pull-down resistor calibration control register (cmn_txpdcal_ctrl)

13.4.10.2.492.1 Offset

Register	Offset
cmn_txpdcal_ctrl	8_00F0h

13.4.10.2.492.2 Diagram



13.4.10.2.492.3 Fields

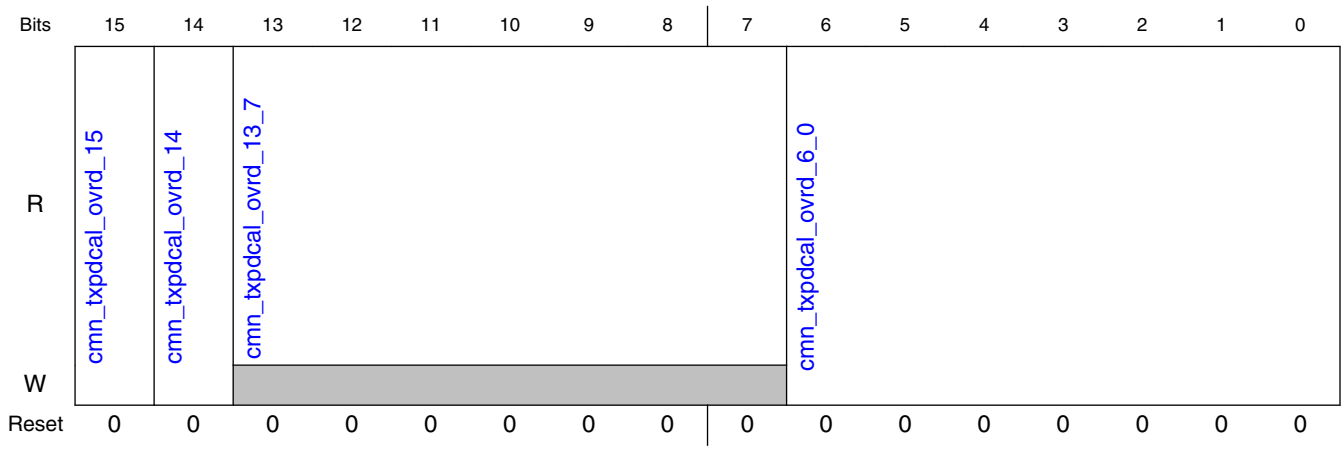
Field	Function
15 cmn_txpdcal_ctrl_15	Start resistor calibration: Activating (1b1) this bit will start the resistor calibration process. This signal must remain active until the resistor calibration process is complete. To start another resistor calibration process, this register must first be set inactive (1b0) until the resistor calibration process done bit in this register is cleared.
14 cmn_txpdcal_ctrl_14	Resistor calibration process done: This bit will be set to 1b1 when the resistor calibration process is complete. It will be cleared by
13 cmn_txpdcal_ctrl_13	No analog calibration response : This signal indicates that the calibration function has gone through the entire calibration process, reached the final calibration value, and the analog has not responded indicating that a valid calibration value has been reached.
12 cmn_txpdcal_ctrl_12	Current analog comparator response: This is the current state of the analog comparator response signal (
11-7 cmn_txpdcal_ctrl_11_7	Reserved
6-0 cmn_txpdcal_ctrl_6_0	Resistor calibration code: This is the calibration code that was determined by the resistor calibration process. The following are the values for the code.

13.4.10.2.493 TX pull-down resistor calibration override register (cmn_txpdcal_ovrd)

13.4.10.2.493.1 Offset

Register	Offset
cmn_txpdcal_ovrd	8_00F1h

13.4.10.2.493.2 Diagram



13.4.10.2.493.3 Fields

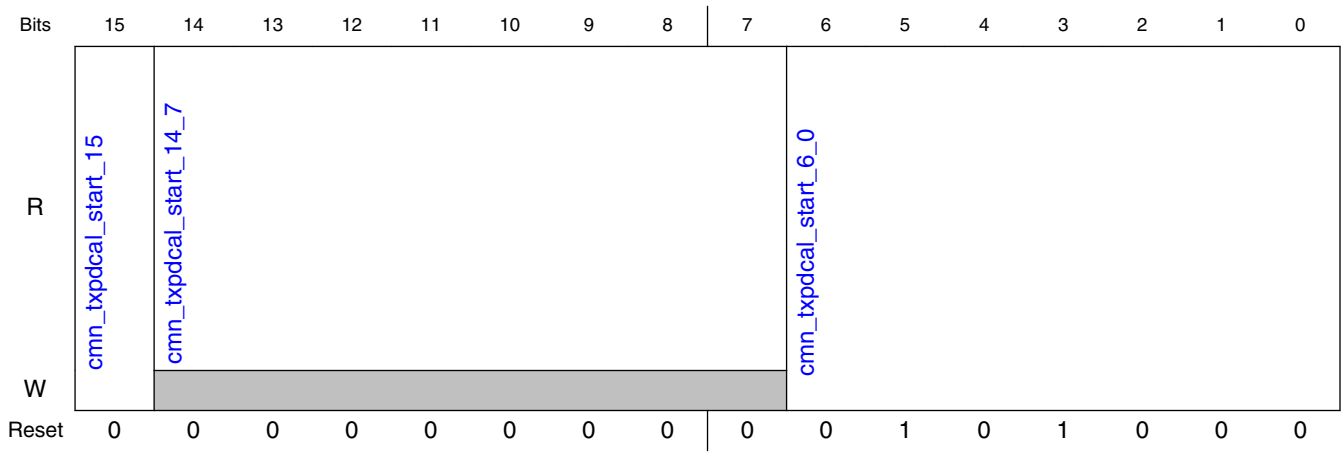
Field	Function
15 cmn_txpdcal_ovrd_15	Resistor code override enable: Activation (1b1) of this register bit allows the resistor codes determined during the automatic resistor calibration process to be overridden. The override value is specified using the resistor code override value field of this register.
14 cmn_txpdcal_ovrd_14	Analog calibration enable override: Activation (1b1) of this register bit will force the analog calibration circuits to be enabled by activating the
13-7 cmn_txpdcal_ovrd_13_7	Reserved
6-0 cmn_txpdcal_ovrd_6_0	Resistor code override value: These bits are used to override the resistor code determined during the automatic resistor calibration process. The resistor code written to these bits is valid when the resistor code override enable bit in this register is active. The following are the values for the code.

13.4.10.2.494 TX pull-down resistor calibration start register (cmn_txpdcal_start)

13.4.10.2.494.1 Offset

Register	Offset
cmn_txpdcal_start	8_00F2h

13.4.10.2.494.2 Diagram



13.4.10.2.494.3 Fields

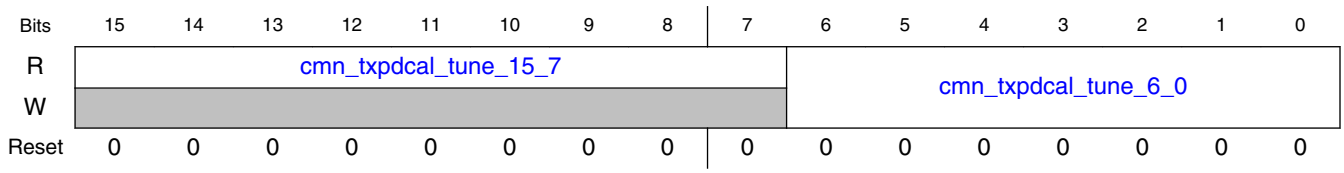
Field	Function
15 cmn_txpdcal_start_15	Resistor calibration direction: This controls the direction that the automatic calibration process steps the calibration codes in.
14-7 cmn_txpdcal_start_14_7	Reserved
6-0 cmn_txpdcal_start_6_0	Start resistor calibration code: This is the calibration code that the resistor calibration process starts with when automatic calibration is run. The following are the values for the code.

13.4.10.2.495 TX pull-down resistor calibration tune register (cmn_txpdcal_tune)

13.4.10.2.495.1 Offset

Register	Offset
cmn_txpdcal_tune	8_00F3h

13.4.10.2.495.2 Diagram



13.4.10.2.495.3 Fields

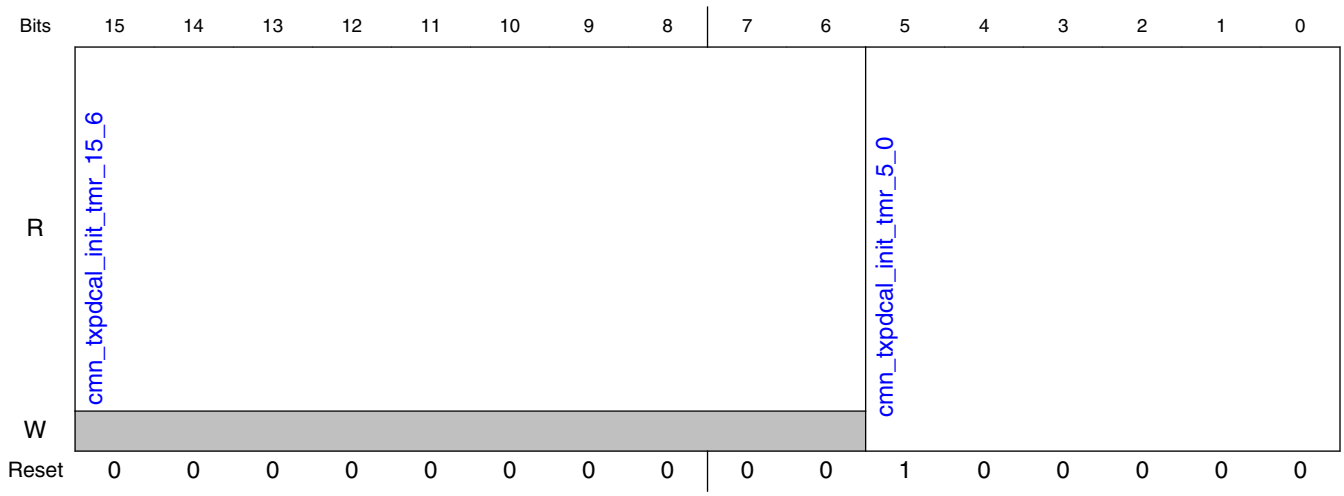
Field	Function
15-7 cmn_txpdcal_tune_15_7	Reserved
6-0 cmn_txpdcal_tune_6_0	Resistor calibration tune value: The value of this field is added to the automatically calibrated code, or the override code if override is enabled. Note that this value is a twos complement value, so the calibrated code can be increased or decreased.

13.4.10.2.496 TX pull-down resistor calibration initialization timer register (cmn_txpdcal_init_tmr)

13.4.10.2.496.1 Offset

Register	Offset
cmn_txpdcal_init_tmr	8_00F4h

13.4.10.2.496.2 Diagram



13.4.10.2.496.3 Fields

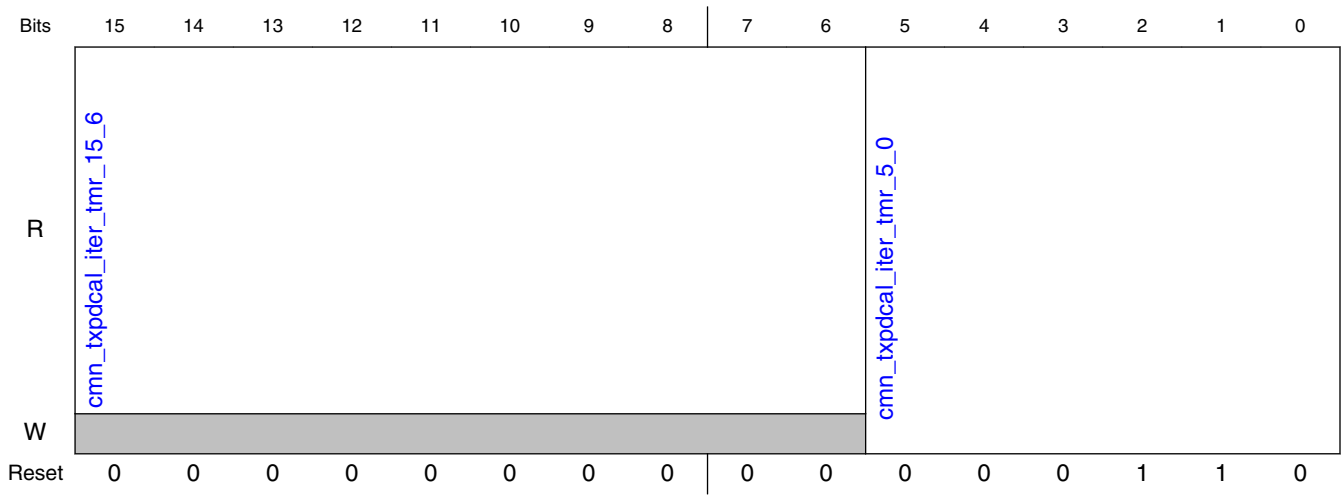
Field	Function
15-6 cmn_txpdcal_init_tmr_15_6	Reserved
5-0 cmn_txpdcal_init_tmr_5_0	Initialization wait timer value: This is the number of

13.4.10.2.497 TX pull-down resistor calibration iteration timer register (cmn_txpdcal_iter_tmr)

13.4.10.2.497.1 Offset

Register	Offset
cmn_txpdcal_iter_tmr	8_00F5h

13.4.10.2.497.2 Diagram



13.4.10.2.497.3 Fields

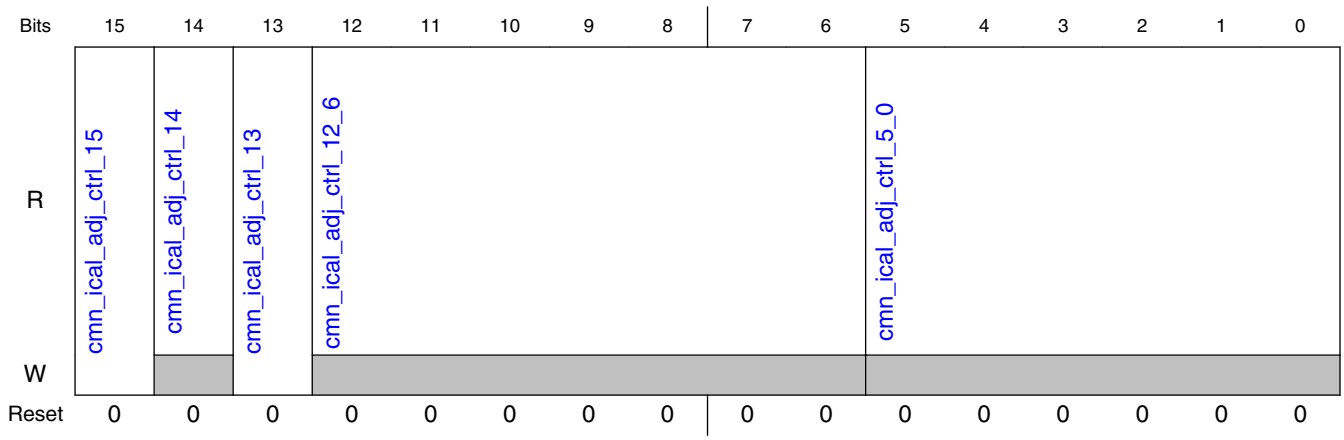
Field	Function
15-6 cmn_txpdcal_iter_tmr_15_6	Reserved
5-0 cmn_txpdcal_iter_tmr_5_0	Iteration wait timer value: This is the number of

13.4.10.2.498 Current calibration adjust control register (cmn_ical_adj_ctrl)

13.4.10.2.498.1 Offset

Register	Offset
cmn_ical_adj_ctrl	8_0100h

13.4.10.2.498.2 Diagram



13.4.10.2.498.3 Fields

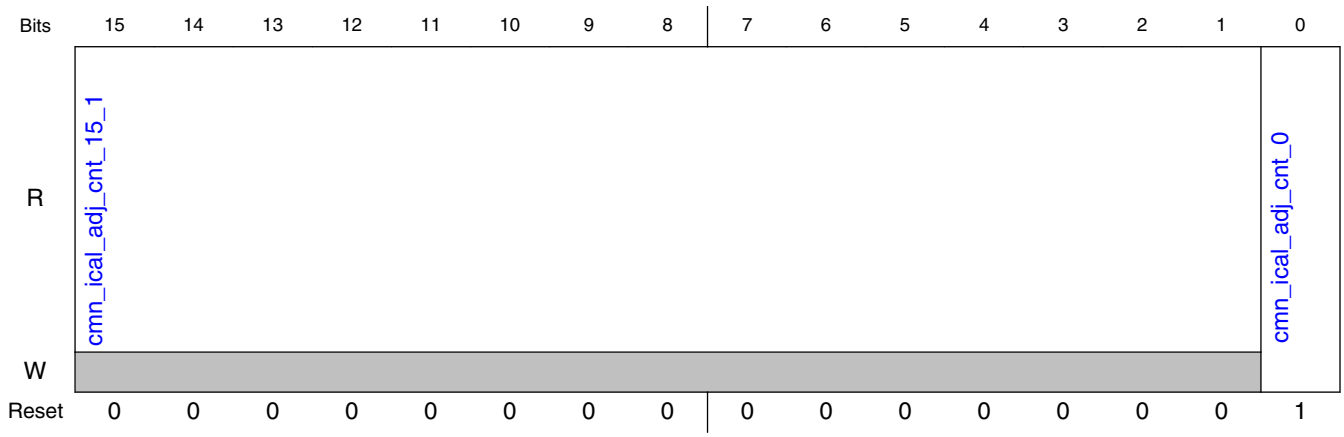
Field	Function
15 cmn_ical_adj_ctrl_15	Start calibration adjust: Activating (1b1) this bit will start the calibration adjust process. This signal must remain active until the calibration adjust process is complete. To start another calibration adjust process, this register must first be set inactive (1b0) until the calibration adjust process done bit in this register is cleared.
14 cmn_ical_adj_ctrl_14	Calibration adjust process done: This bit will be set to 1b1 when the calibration adjust process is complete. It will be cleared by
13 cmn_ical_adj_ctrl_13	Calibration adjust direction: This controls the direction that the automatic calibration process steps the calibration codes in.
12-6 cmn_ical_adj_ctrl_12_6	Reserved
5-0 cmn_ical_adj_ctrl_5_0	Calibration adjust code: This is the value that the calibration adjust function is currently attempting to adjust the calibration code by. Note that this is a signed twos complement value. The following are the values for the code:

13.4.10.2.499 Current calibration adjust count register (cmn_ical_adj_cnt)

13.4.10.2.499.1 Offset

Register	Offset
cmn_ical_adj_cnt	8_0101h

13.4.10.2.499.2 Diagram



13.4.10.2.499.3 Fields

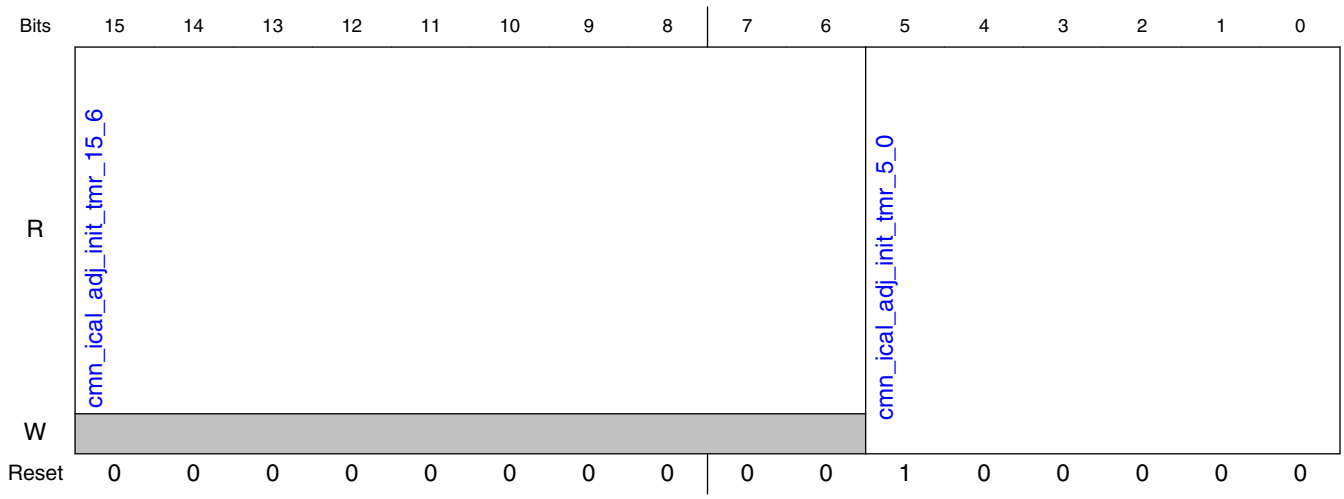
Field	Function
15-1 cmn_ical_adj_cnt_15_1	Reserved
0 cmn_ical_adj_cnt_0	Calibration adjust count value: Specifies the number of calibration adjust iterations that will be performed every time a calibration adjust is requested.

13.4.10.2.500 Current calibration adjust initialization timer register (cmn_ical_adj_init_tmr)

13.4.10.2.500.1 Offset

Register	Offset
cmn_ical_adj_init_tmr	8_0102h

13.4.10.2.500.2 Diagram



13.4.10.2.500.3 Fields

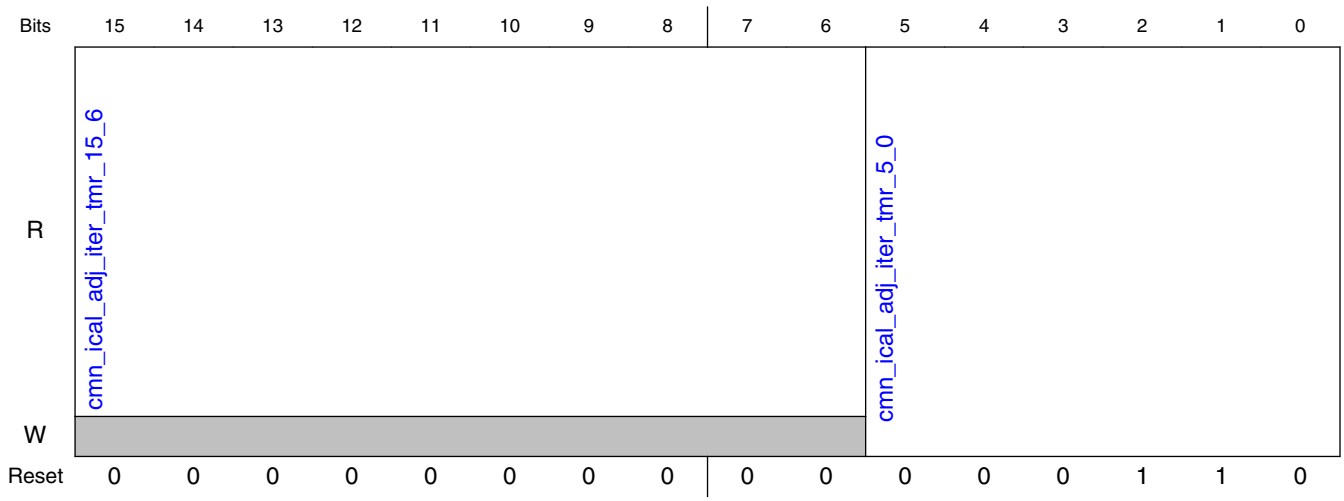
Field	Function
15-6 cmn_ical_adj_init_tmr_15_6	Reserved
5-0 cmn_ical_adj_init_tmr_5_0	Initialization wait timer value: This is the number of clock cycles to wait between when the analog calibration circuits are enabled, and when the first calibration adjust test value is placed on the calibration code signal, going to the analog.

13.4.10.2.501 Current calibration adjust iteration timer register (cmn_ical_adj_iter_tmr)

13.4.10.2.501.1 Offset

Register	Offset
cmn_ical_adj_iter_tmr	8_0103h

13.4.10.2.501.2 Diagram



13.4.10.2.501.3 Fields

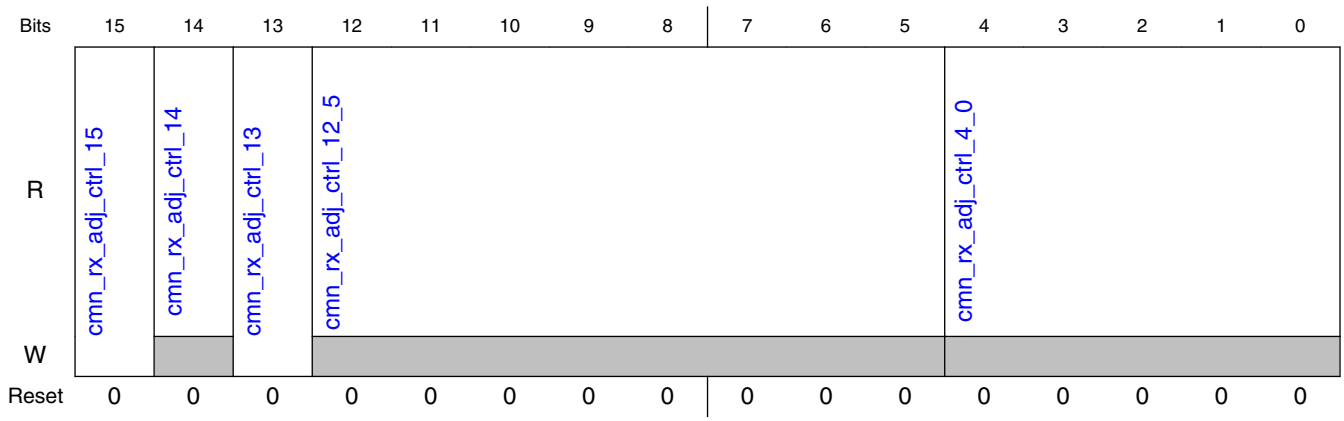
Field	Function
15-6 cmn_ical_adj_iter_tmr_15_6	Reserved
5-0 cmn_ical_adj_iter_tmr_5_0	Iteration wait timer value: This is the number of clock cycles to wait between when an adjusted calibration value is placed on the calibration signal going to the analog, and when the comparator value coming from the analog circuits can be checked.

13.4.10.2.502 RX resistor calibration adjust control register (cmn_rx_adj_ctrl)

13.4.10.2.502.1 Offset

Register	Offset
cmn_rx_adj_ctrl	8_0104h

13.4.10.2.502.2 Diagram



13.4.10.2.502.3 Fields

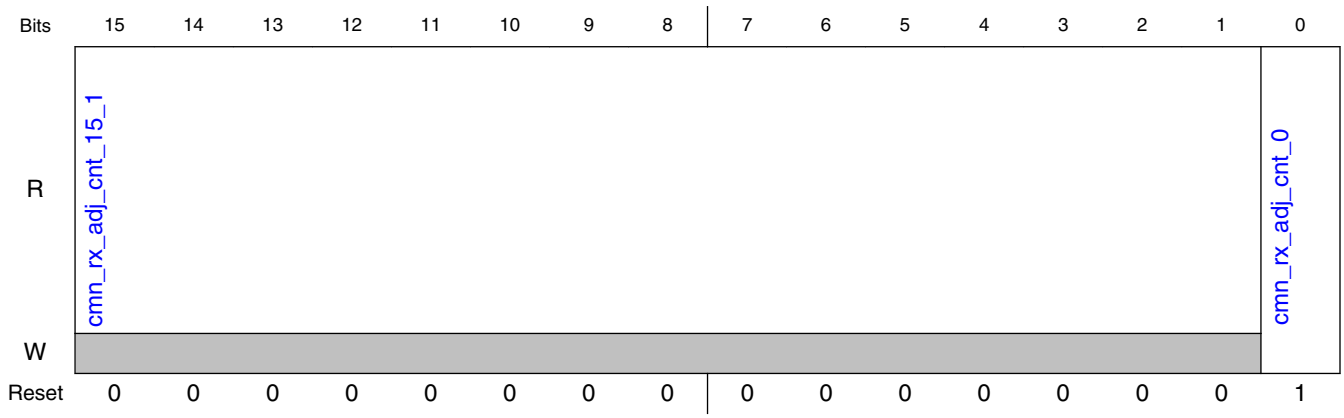
Field	Function
15 cmn_rx_adj_ctrl_15	Start calibration adjust: Activating (1b1) this bit will start the calibration adjust process. This signal must remain active until the calibration adjust process is complete. To start another calibration adjust process, this register must first be set inactive (1b0) until the calibration adjust process done bit in this register is cleared.
14 cmn_rx_adj_ctrl_14	Calibration adjust process done: This bit will be set to 1b1 when the calibration adjust process is complete. It will be cleared by
13 cmn_rx_adj_ctrl_13	Calibration adjust direction: This controls the direction that the automatic calibration process steps the calibration codes in.
12-5 cmn_rx_adj_ctrl_12_5	Reserved
4-0 cmn_rx_adj_ctrl_4_0	Calibration adjust code: This is the value that the calibration adjust function is currently attempting to adjust the calibration code by. Note that this is a signed twos compliment value. The following are the values for the code:

13.4.10.2.503 RX resistor calibration adjust count register (cmn_rx_adj_cnt)

13.4.10.2.503.1 Offset

Register	Offset
cmn_rx_adj_cnt	8_0105h

13.4.10.2.503.2 Diagram



13.4.10.2.503.3 Fields

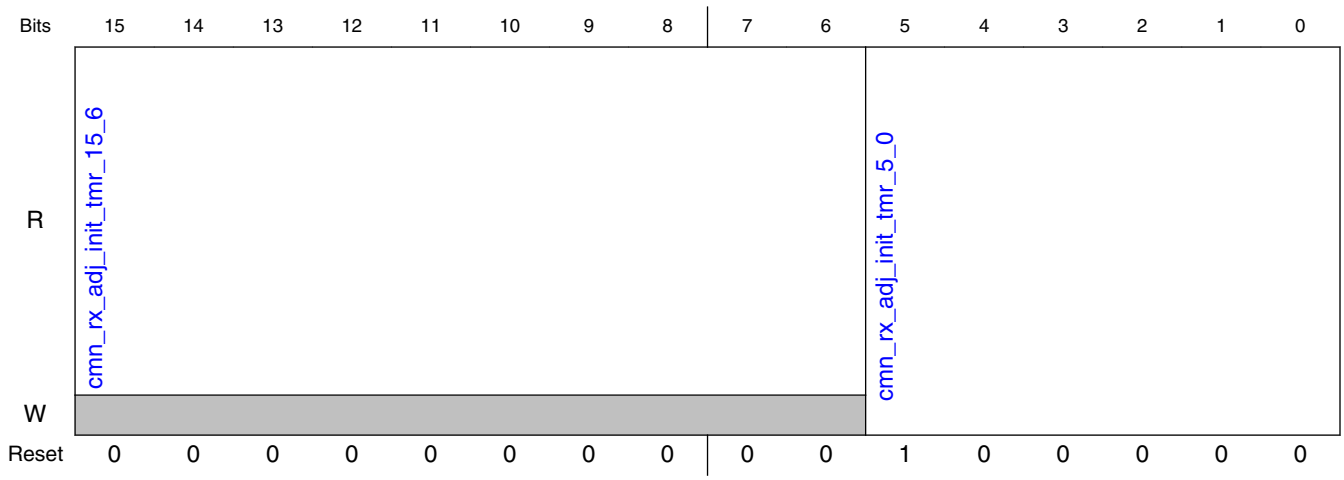
Field	Function
15-1 cmn_rx_adj_cnt_15_1	Reserved
0 cmn_rx_adj_cnt_0	Calibration adjust count value: Specifies the number of calibration adjust itinerations that will be performed every time a calibration adjust is requested.

13.4.10.2.504 RX resistor calibration adjust initialization timer register (cmn_rx_adj_init_tmr)

13.4.10.2.504.1 Offset

Register	Offset
cmn_rx_adj_init_tmr	8_0106h

13.4.10.2.504.2 Diagram



13.4.10.2.504.3 Fields

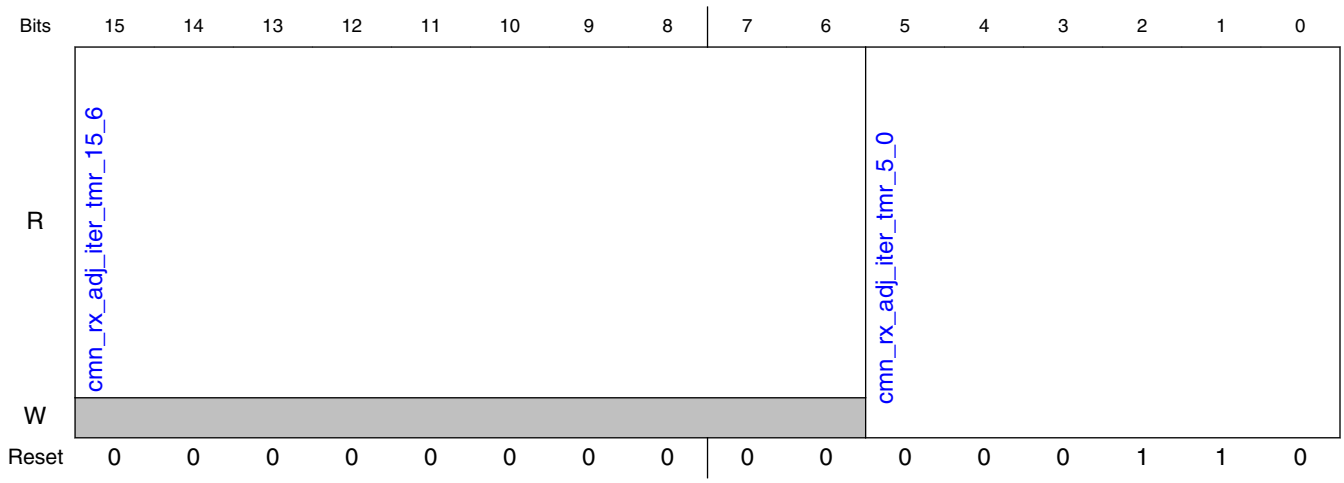
Field	Function
15-6 cmn_rx_adj_init_tmr_15_6	Reserved
5-0 cmn_rx_adj_init_tmr_5_0	Initialization wait timer value: This is the number of clock cycles to wait between when the analog calibration circuits are enabled, and when the first calibration adjust test value is placed on the calibration code signal, going to the analog.

13.4.10.2.505 RX resistor calibration adjust iteration timer register (cmn_rx_adj_iter_tmr)

13.4.10.2.505.1 Offset

Register	Offset
cmn_rx_adj_iter_tmr	8_0107h

13.4.10.2.505.2 Diagram



13.4.10.2.505.3 Fields

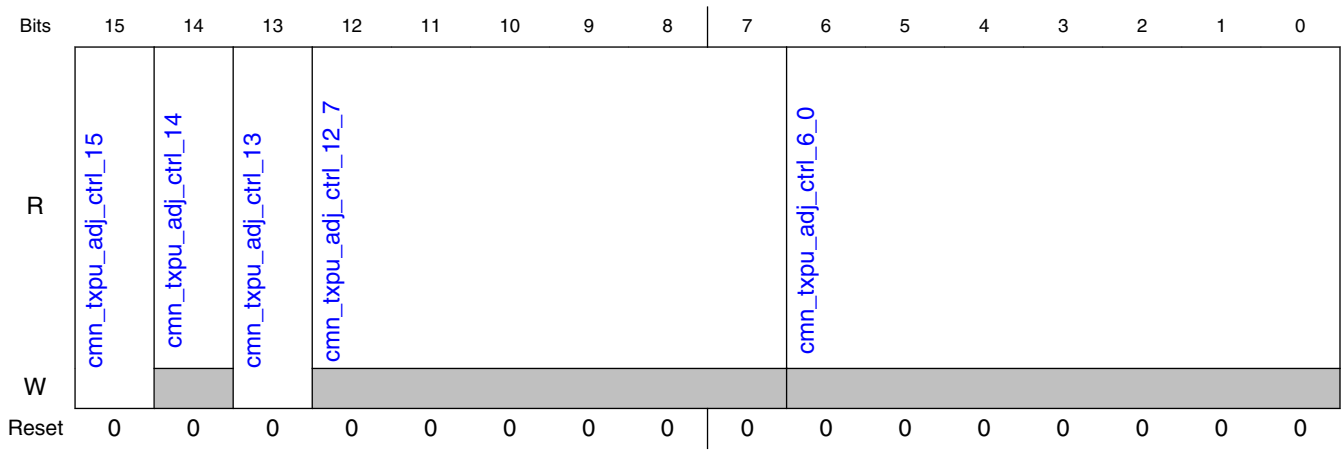
Field	Function
15-6 cmn_rx_adj_iter_tmr_15_6	Reserved
5-0 cmn_rx_adj_iter_tmr_5_0	Iteration wait timer value: This is the number of clock cycles to wait between when an adjusted calibration value is placed on the calibration signal going to the analog, and when the comparator value coming from the analog circuits can be checked.

13.4.10.2.506 TX pull up resistor calibration adjust control register (cmn_txpu_adj_ctrl)

13.4.10.2.506.1 Offset

Register	Offset
cmn_txpu_adj_ctrl	8_0108h

13.4.10.2.506.2 Diagram



13.4.10.2.506.3 Fields

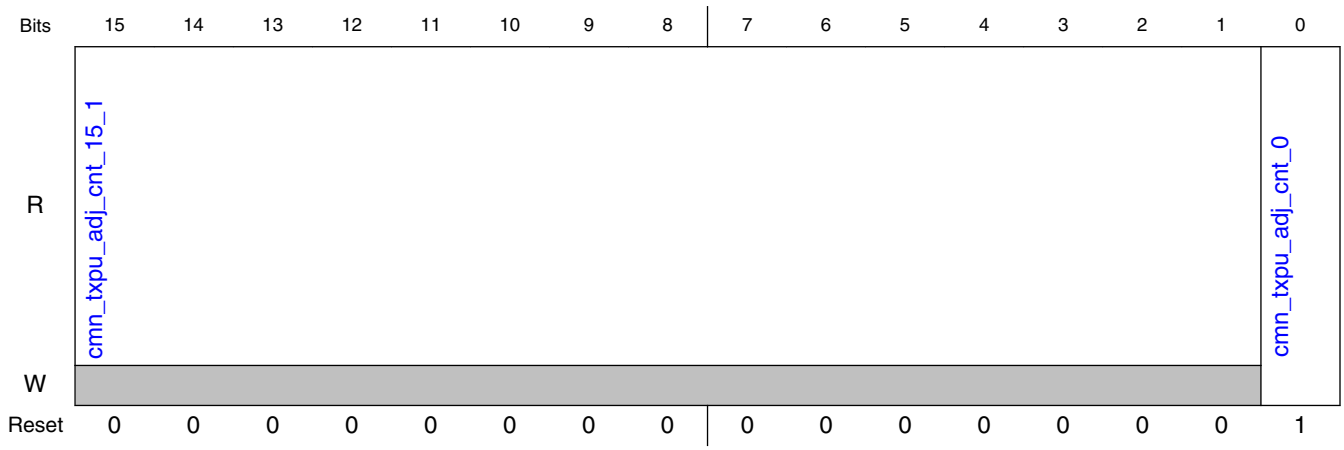
Field	Function
15 cmn_txpu_adj_ctrl_15	Start calibration adjust: Activating (1b1) this bit will start the calibration adjust process. This signal must remain active until the calibration adjust process is complete. To start another calibration adjust process, this register must first be set inactive (1b0) until the calibration adjust process done bit in this register is cleared.
14 cmn_txpu_adj_ctrl_14	Calibration adjust process done: This bit will be set to 1b1 when the calibration adjust process is complete. It will be cleared by
13 cmn_txpu_adj_ctrl_13	Calibration adjust direction: This controls the direction that the automatic calibration process steps the calibration codes in.
12-7 cmn_txpu_adj_ctrl_12_7	Reserved
6-0 cmn_txpu_adj_ctrl_6_0	Calibration adjust code: This is the value that the calibration adjust function is currently attempting to adjust the calibration code by. Note that this is a signed twos compliment value. The following are the values for the code:

13.4.10.2.507 TX pull up resistor calibration adjust count register (cmn_txpu_adj_cnt)

13.4.10.2.507.1 Offset

Register	Offset
cmn_txpu_adj_cnt	8_0109h

13.4.10.2.507.2 Diagram



13.4.10.2.507.3 Fields

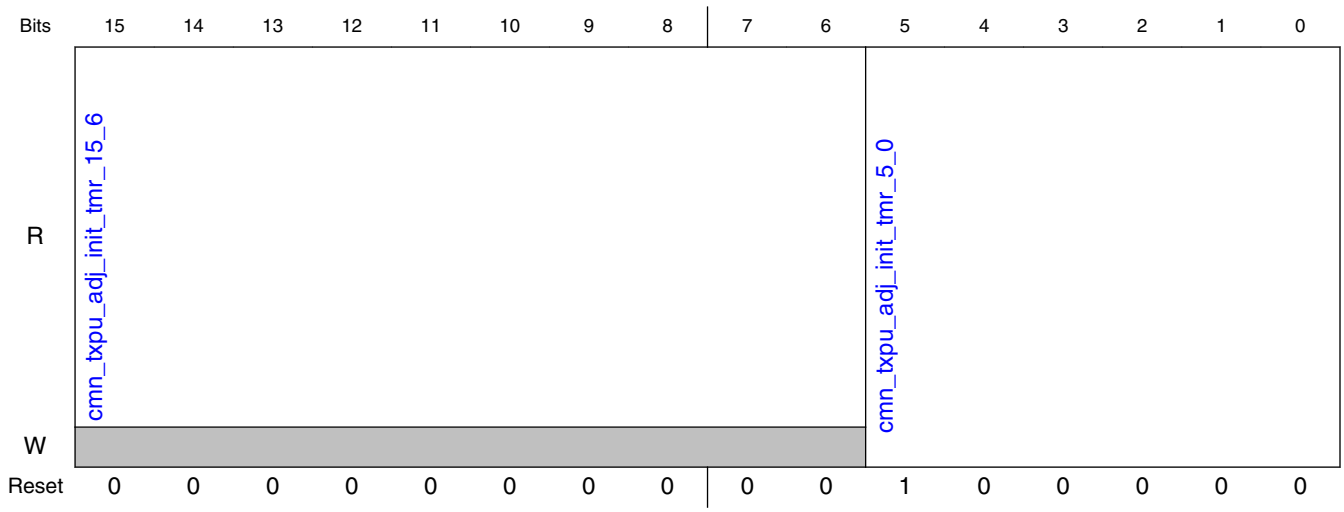
Field	Function
15-1 cmn_txpu_adj_cnt_15_1	Reserved
0 cmn_txpu_adj_cnt_0	Calibration adjust count value: Specifies the number of calibration adjust iterations that will be performed every time a calibration adjust is requested.

13.4.10.2.508 TX pull up resistor calibration adjust initialization timer register (cmn_txpu_adj_init_tmr)

13.4.10.2.508.1 Offset

Register	Offset
cmn_txpu_adj_init_tmr	8_010Ah

13.4.10.2.508.2 Diagram



13.4.10.2.508.3 Fields

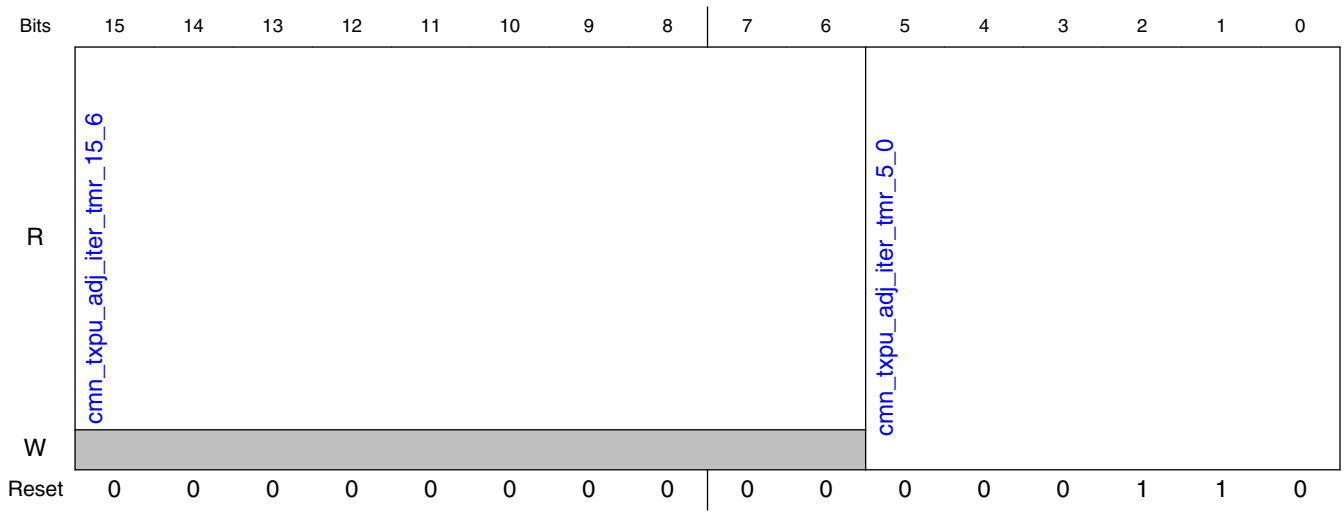
Field	Function
15-6 cmn_txpu_adj_init_tmr_15_6	Reserved
5-0 cmn_txpu_adj_init_tmr_5_0	Initialization wait timer value: This is the number of clock cycles to wait between when the analog calibration circuits are enabled, and when the first calibration adjust test value is placed on the calibration code signal, going to the analog.

13.4.10.2.509 TX pull up resistor calibration adjust iteration timer register (cmn_txpu_adj_iter_tmr)

13.4.10.2.509.1 Offset

Register	Offset
cmn_txpu_adj_iter_tmr	8_010Bh

13.4.10.2.509.2 Diagram



13.4.10.2.509.3 Fields

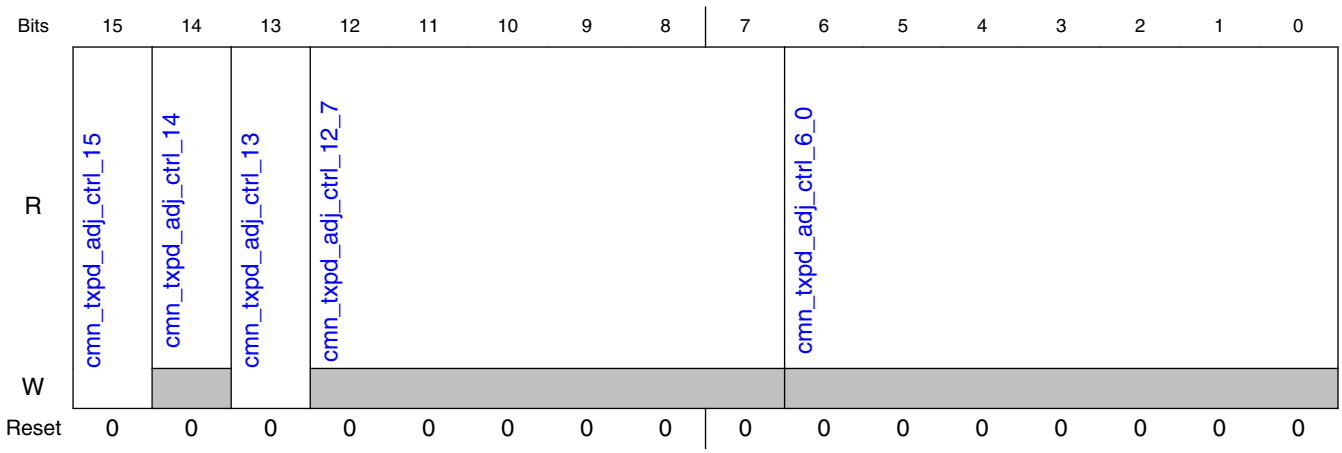
Field	Function
15-6 cmn_txpu_adj_iter_tmr_15_6	Reserved
5-0 cmn_txpu_adj_iter_tmr_5_0	Iteration wait timer value: This is the number of clock cycles to wait between when an adjusted calibration value is placed on the calibration signal going to the analog, and when the comparator value coming from the analog circuits can be checked.

13.4.10.2.510 TX pull down resistor calibration adjust control register (cmn_txpd_adj_ctrl)

13.4.10.2.510.1 Offset

Register	Offset
cmn_txpd_adj_ctrl	8_010Ch

13.4.10.2.510.2 Diagram



13.4.10.2.510.3 Fields

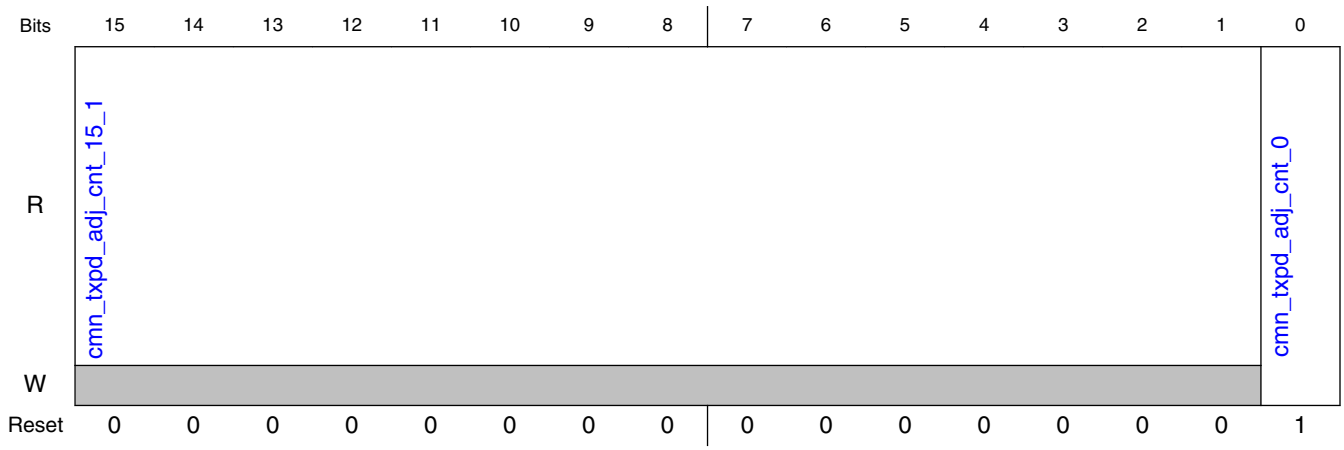
Field	Function
15 cmn_txpd_adj_ctrl_15	Start calibration adjust: Activating (1b1) this bit will start the calibration adjust process. This signal must remain active until the calibration adjust process is complete. To start another calibration adjust process, this register must first be set inactive (1b0) until the calibration adjust process done bit in this register is cleared.
14 cmn_txpd_adj_ctrl_14	Calibration adjust process done: This bit will be set to 1b1 when the calibration adjust process is complete. It will be cleared by
13 cmn_txpd_adj_ctrl_13	Calibration adjust direction: This controls the direction that the automatic calibration process steps the calibration codes in.
12-7 cmn_txpd_adj_ctrl_12_7	Reserved
6-0 cmn_txpd_adj_ctrl_6_0	Calibration adjust code: This is the value that the calibration adjust function is currently attempting to adjust the calibration code by. Note that this is a signed twos compliment value. The following are the values for the code:

13.4.10.2.511 TX pull down resistor calibration adjust count register (cmn_txpd_adj_cnt)

13.4.10.2.511.1 Offset

Register	Offset
cmn_txpd_adj_cnt	8_010Dh

13.4.10.2.511.2 Diagram



13.4.10.2.511.3 Fields

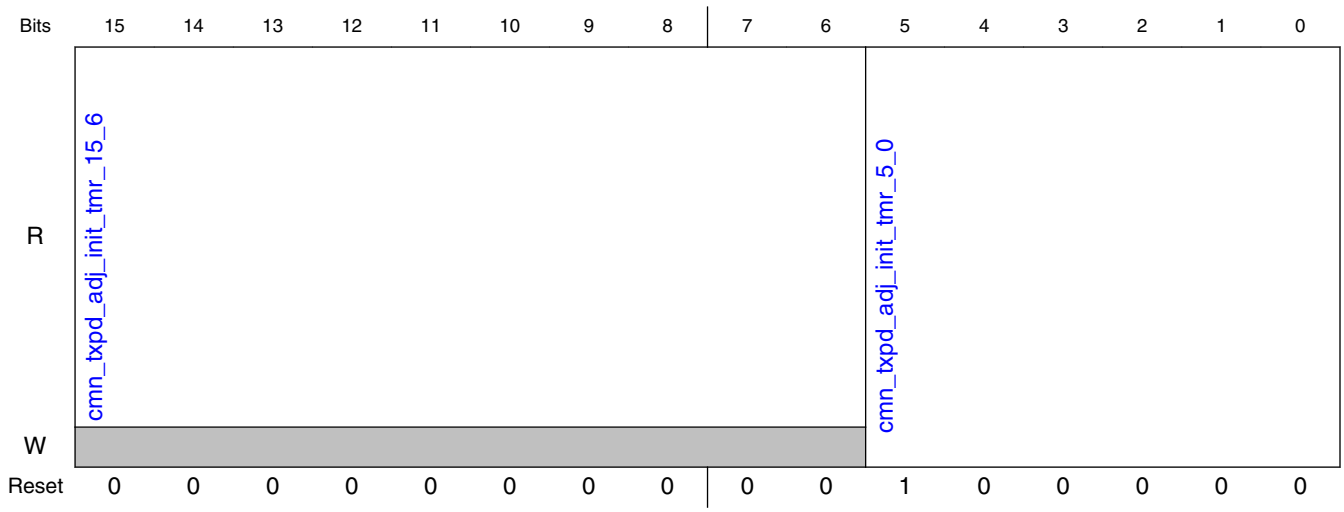
Field	Function
15-1 cmn_txpd_adj_cnt_15_1	Reserved
0 cmn_txpd_adj_cnt_0	Calibration adjust count value: Specifies the number of calibration adjust itinerations that will be performed every time a calibration adjust is requested.

13.4.10.2.512 TX pull down resistor calibration adjust initialization timer register (cmn_txpd_adj_init_tmr)

13.4.10.2.512.1 Offset

Register	Offset
cmn_txpd_adj_init_tmr	8_010Eh

13.4.10.2.512.2 Diagram



13.4.10.2.512.3 Fields

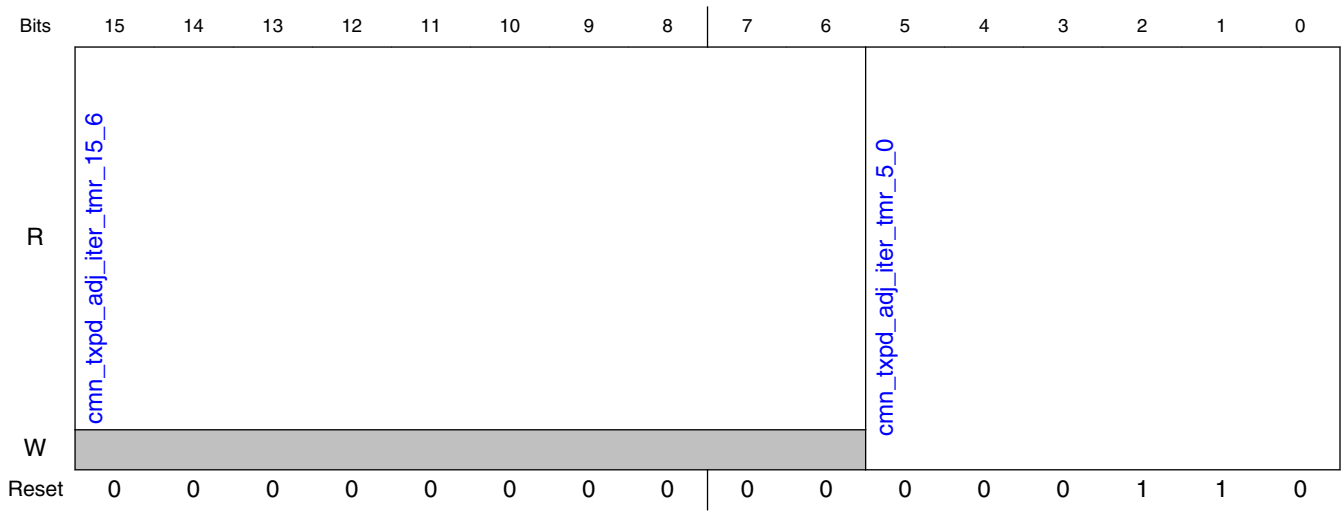
Field	Function
15-6 cmn_txpd_adj_init_tmr_15_6	Reserved
5-0 cmn_txpd_adj_init_tmr_5_0	Initialization wait timer value: This is the number of clock cycles to wait between when the analog calibration circuits are enabled, and when the first calibration adjust test value is placed on the calibration code signal, going to the analog.

13.4.10.2.513 TX pull down resistor calibration adjust iteration timer register (cmn_txpd_adj_iter_tmr)

13.4.10.2.513.1 Offset

Register	Offset
cmn_txpd_adj_iter_tmr	8_010Fh

13.4.10.2.513.2 Diagram



13.4.10.2.513.3 Fields

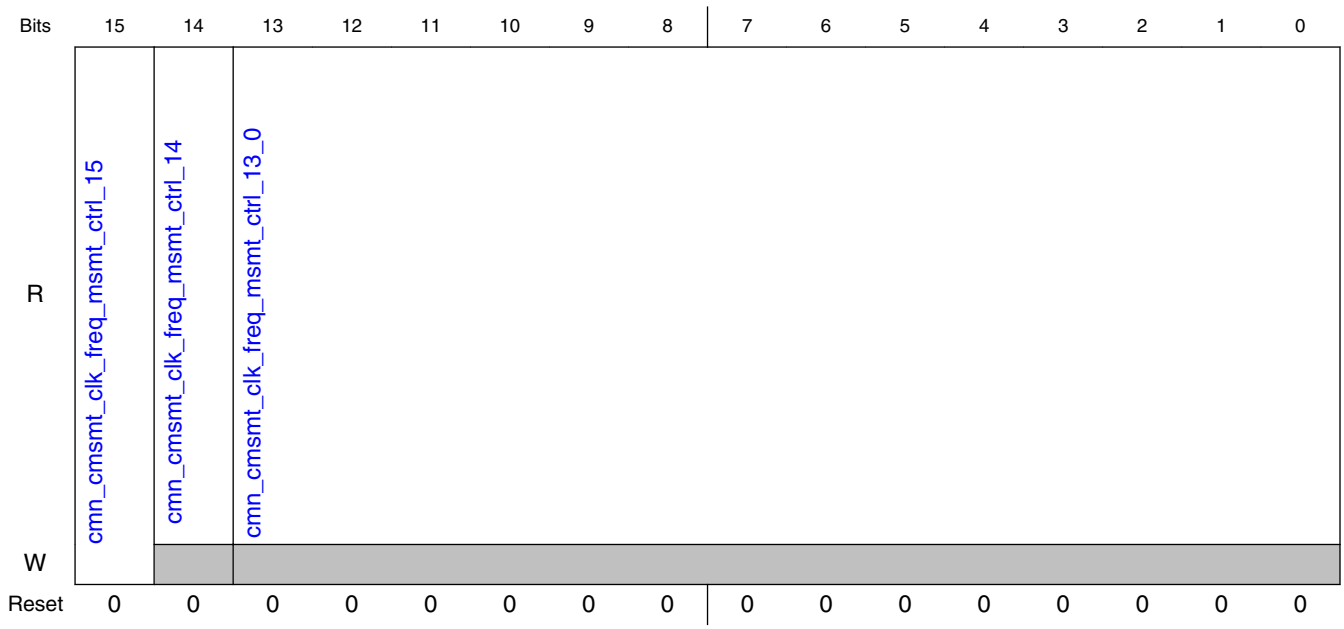
Field	Function
15-6 cmn_txpd_adj_iter_tmr_15_6	Reserved
5-0 cmn_txpd_adj_iter_tmr_5_0	Iteration wait timer value: This is the number of clock cycles to wait between when an adjusted calibration value is placed on the calibration signal going to the analog, and when the comparator value coming from the analog circuits can be checked.

13.4.10.2.514 Clock frequency measurement control register (cmn_cmsmt_clk_freq_msmt_ctrl)

13.4.10.2.514.1 Offset

Register	Offset
cmn_cmsmt_clk_freq_msmt_ctrl	8_01A0h

13.4.10.2.514.2 Diagram



13.4.10.2.514.3 Fields

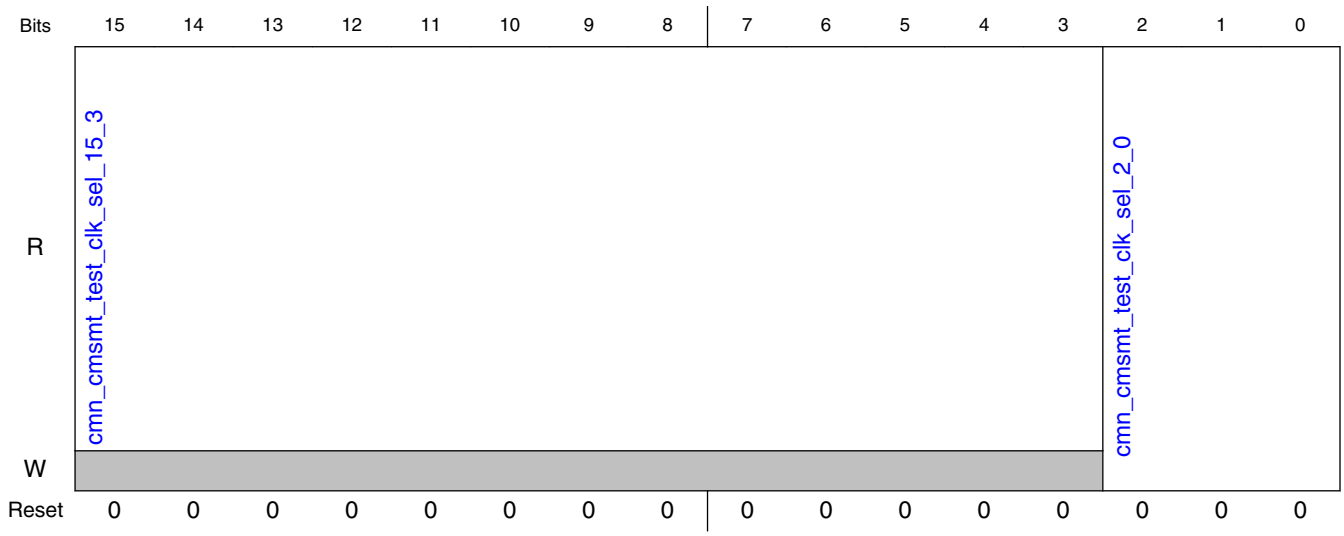
Field	Function
15 cmn_cmsmt_clk_freq_msmt_ctrl_15	Run test clock measurement: Activating (1b1) this bit will run the test clock measurement process. This bit must remain active until the test clock measurement process is complete, as indicated by the test clock measurement done bit in this register. To start another measurement process, this bit must first be driven inactive then driven active again.
14 cmn_cmsmt_clk_freq_msmt_ctrl_14	Test clock measurement done: This bit will be set to 1b1 when the test clock measurement process is complete. It will be cleared by the deactivation of the start test clock measurement bit in this register.
13-0 cmn_cmsmt_clk_freq_msmt_ctrl_13_0	Reserved

13.4.10.2.515 Test clock selection register (cmn_cmsmt_test_clk_sel)

13.4.10.2.515.1 Offset

Register	Offset
cmn_cmsmt_test_clk_sel	8_01A1h

13.4.10.2.515.2 Diagram



13.4.10.2.515.3 Fields

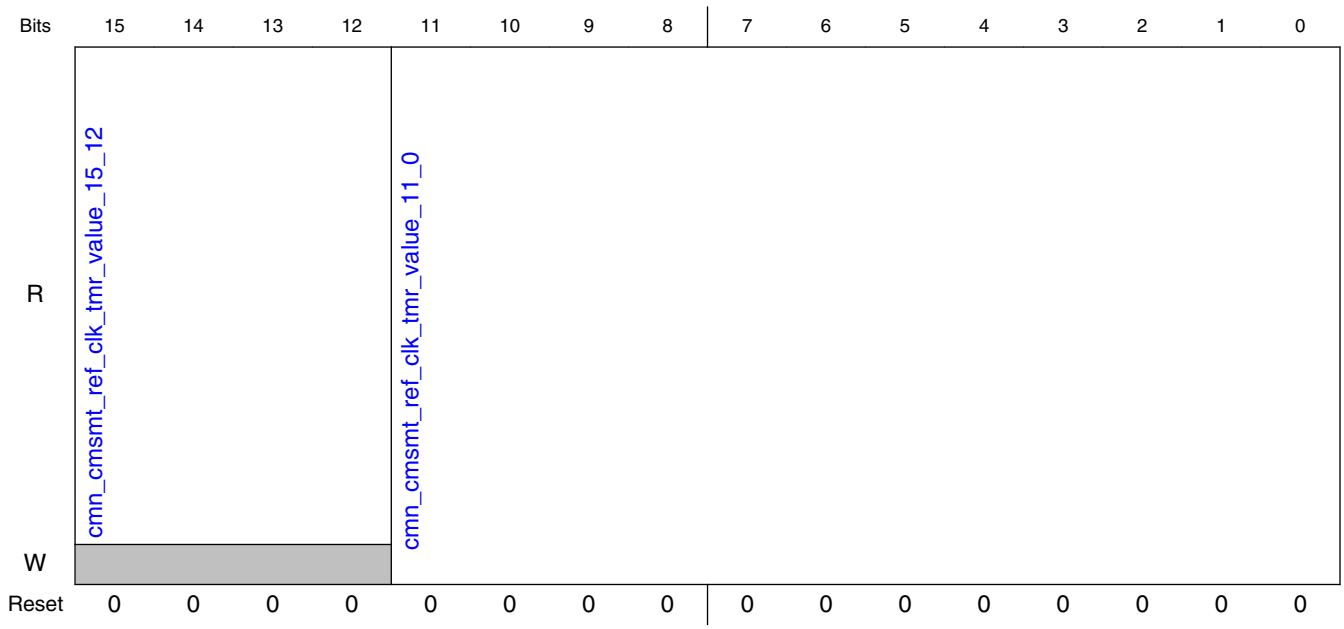
Field	Function
15-3 cmn_cmsmt_test_clk_sel_15_3	Reserved
2-0 cmn_cmsmt_test_clk_sel_2_0	Test clock select: This field drives the

13.4.10.2.516 Reference clock timer value register (cmn_cmsmt_ref_clk_tmr_value)

13.4.10.2.516.1 Offset

Register	Offset
cmn_cmsmt_ref_clk_tmr_value	8_01A2h

13.4.10.2.516.2 Diagram



13.4.10.2.516.3 Fields

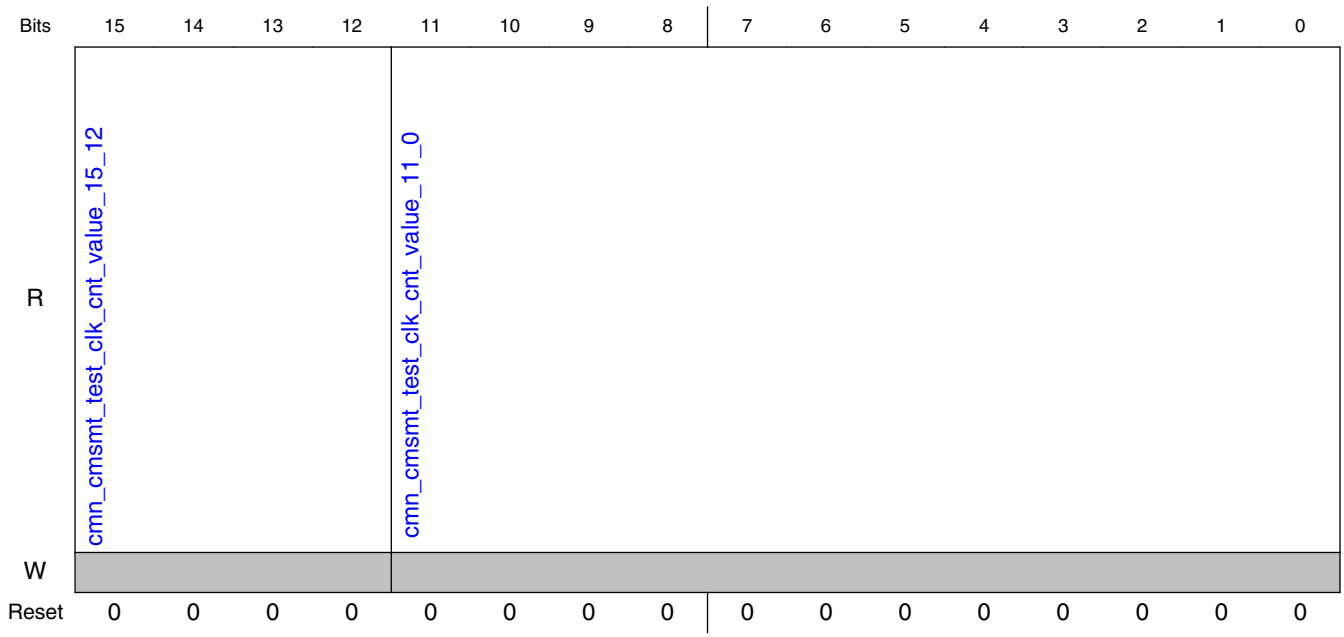
Field	Function
15-12 cmn_cmsmt_ref_clk_tmr_value_15_12	Reserved
11-0 cmn_cmsmt_ref_clk_tmr_value_11_0	Reference clock timer value : This specifies the amount of time, in reference clock cycles, to count test clock cycles. This value minus 1 is loaded into the reference clock timer. A value of 0 for this field is not valid when running this function.

13.4.10.2.517 Test clock counter value register (cmn_cmsmt_test_clk_cnt_value)

13.4.10.2.517.1 Offset

Register	Offset
cmn_cmsmt_test_clk_cnt_value	8_01A3h

13.4.10.2.517.2 Diagram



13.4.10.2.517.3 Fields

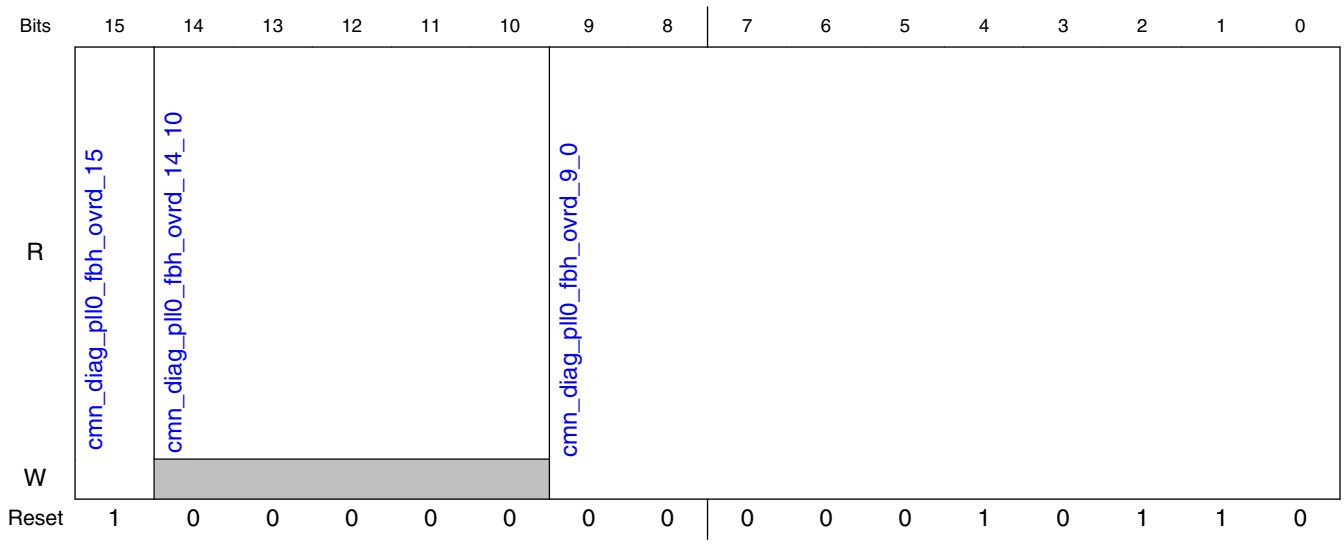
Field	Function
15-12 cmn_cmsmt_test_clk_cnt_value_15_12	Reserved
11-0 cmn_cmsmt_test_clk_cnt_value_11_0	Test clock counter value: When the test clock measurement process is complete, the value in this field specifies the number of test clock cycles that were counted in the time specified by the reference clock timer value. This field is only valid while the test clock measurement done bit in the

13.4.10.2.518 PLL 0 feedback divider value high override register (cmn_diag_pll0_fbh_ovrd)

13.4.10.2.518.1 Offset

Register	Offset
cmn_diag_pll0_fbh_ovrd	8_01C0h

13.4.10.2.518.2 Diagram



13.4.10.2.518.3 Fields

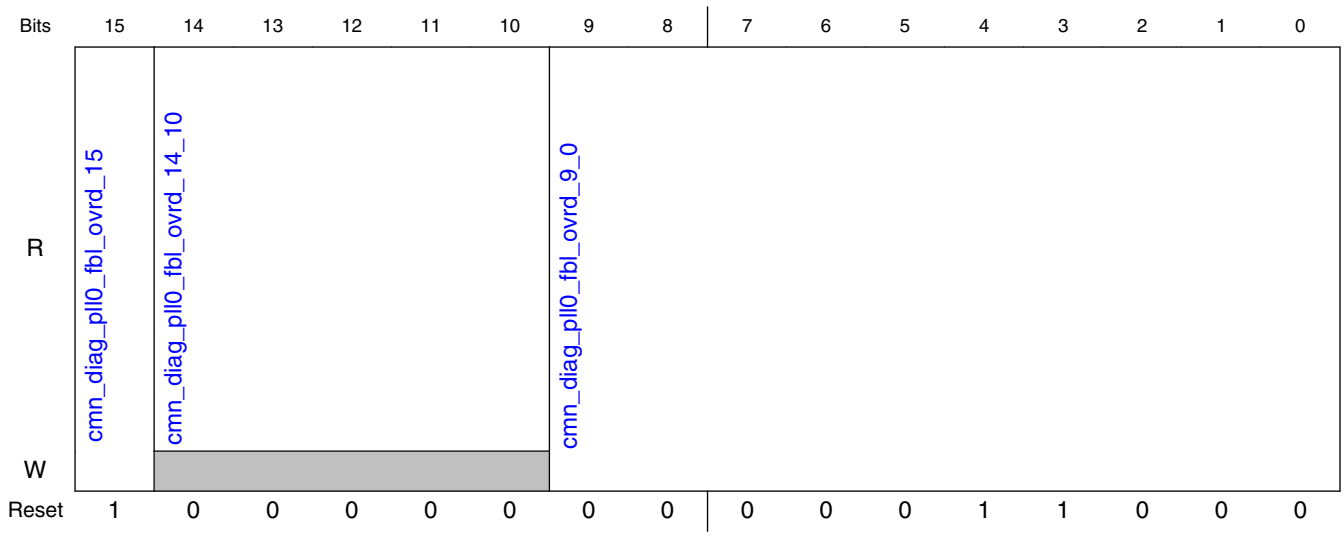
Field	Function
15 cmn_diag_pll0_fbh_ovrd_15	PLL feedback divider high override enable : When active (1b1), the PLL feedback divider high override value field of this register will be used to override the value on the
14-10 cmn_diag_pll0_fbh_ovrd_14_10	Reserved
9-0 cmn_diag_pll0_fbh_ovrd_9_0	PLL feedback divider high override value : When enabled, the value in this field will be used to override the value on the

13.4.10.2.519 PLL 0 feedback divider value low override register (cmn_diag_pll0_fbl_ovrd)

13.4.10.2.519.1 Offset

Register	Offset
cmn_diag_pll0_fbl_ovrd	8_01C1h

13.4.10.2.519.2 Diagram



13.4.10.2.519.3 Fields

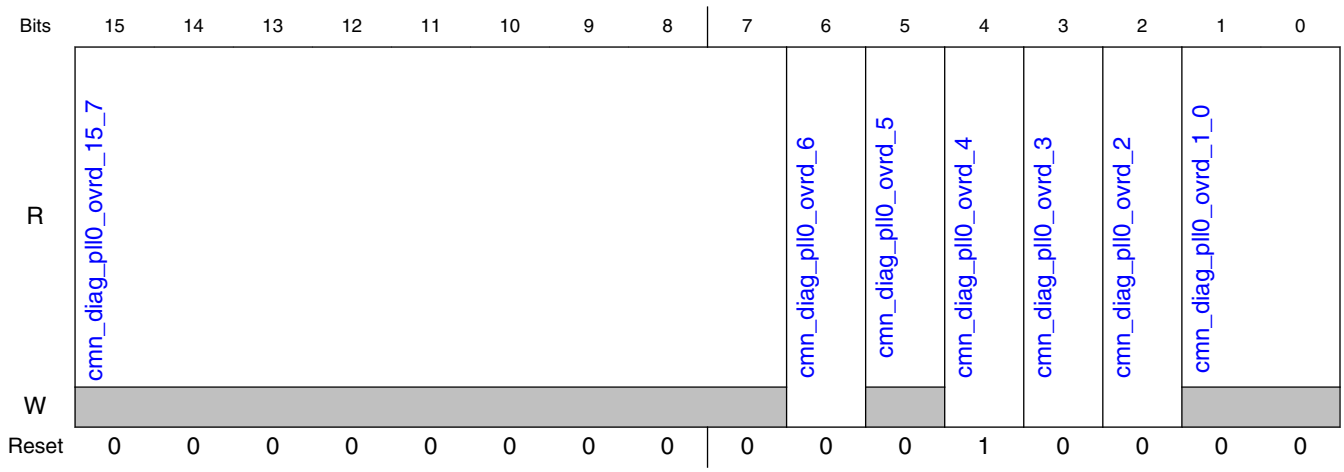
Field	Function
15 cmn_diag_pll0_fbl_ovrd_15	PLL feedback divider low override enable : When active (1b1), the PLL feedback divider low override value field of this register will be used to override the value on the
14-10 cmn_diag_pll0_fbl_ovrd_14_10	Reserved
9-0 cmn_diag_pll0_fbl_ovrd_9_0	PLL feedback divider low override value : When enabled, the value in this field will be used to override the value on the

13.4.10.2.520 PLL 0 override register (cmn_diag_pll0_ovrd)

13.4.10.2.520.1 Offset

Register	Offset
cmn_diag_pll0_ovrd	8_01C2h

13.4.10.2.520.2 Diagram



13.4.10.2.520.3 Fields

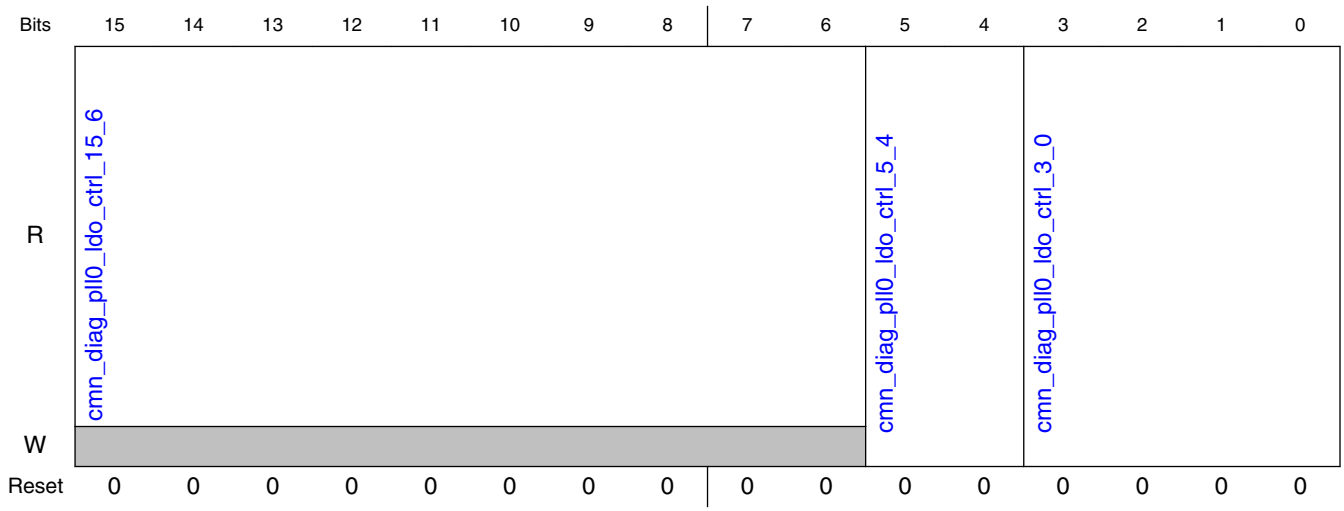
Field	Function
15-7 cmn_diag_pll0_ovrd_15_7	Reserved
6 cmn_diag_pll0_ovrd_6	PLL lock override: When active (1b1), this bit will force the PLL lock indication active.
5 cmn_diag_pll0_ovrd_5	Reserved
4 cmn_diag_pll0_ovrd_4	PLL PCIe mode: Selects corresponding PCIe CP and Loop filter block sections, by driving the
3 cmn_diag_pll0_ovrd_3	PLL VCO calibration enable override enable: When active (1b1), the PLL VCO calibration enable override bit in this register, can be used to directly control the enable of the VCO calibration function in the PLL (instead of the VCO calibration module).
2 cmn_diag_pll0_ovrd_2	PLL VCO calibration enable override: When enabled by the PLL VCO calibration enable override enable bit in this register, this bit will directly control the enable of the VCO calibration function in the PLL.
1-0 cmn_diag_pll0_ovrd_1_0	Reserved

13.4.10.2.521 PLL 0 LDO control register (cmn_diag_pll0_ldo_ctrl)

13.4.10.2.521.1 Offset

Register	Offset
cmn_diag_pll0_ldo_ctrl	8_01C3h

13.4.10.2.521.2 Diagram



13.4.10.2.521.3 Fields

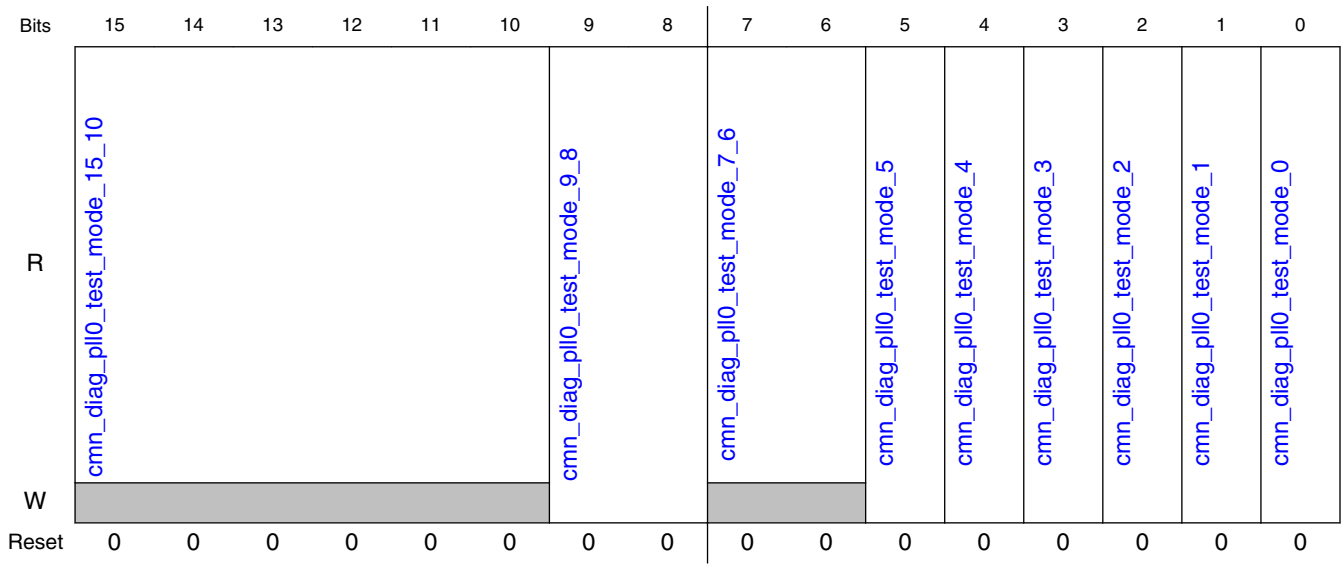
Field	Function
15-6 cmn_diag_pll0_ldo_ctrl_15_6	Reserved
5-4 cmn_diag_pll0_ldo_ctrl_5_4	PLL 0 LDO output bypass selection: When the optional PLL voltage regulator is present, this binary code controls the output bypass selection.
3-0 cmn_diag_pll0_ldo_ctrl_3_0	PLL 0 LDO nominal output voltage: When the optional PLL voltage regulator is present in the design, this code controls the regulator's nominal output voltage.

13.4.10.2.522 PLL 0 test mode register (cmn_diag_pll0_test_mode)

13.4.10.2.522.1 Offset

Register	Offset
cmn_diag_pll0_test_mode	8_01C4h

13.4.10.2.522.2 Diagram



13.4.10.2.522.3 Fields

Field	Function
15-10 cmn_diag_pll0_test_mode_15_10	Reserved
9-8 cmn_diag_pll0_test_mode_9_8	PLL 0 secondary loop bandwidth control: Controls the bandwidth of the second order loop, by driving the
7-6 cmn_diag_pll0_test_mode_7_6	Reserved
5 cmn_diag_pll0_test_mode_5	PLL 0 VCO clock divider test mode select: During VCO calibration, the VCO clock is divided by 2 before being sent to the feedback divider, by driving the
4 cmn_diag_pll0_test_mode_4	PLL 0 secondary loop bypass: Controls the bypass secondary loop output filter, by driving the

Table continues on the next page...

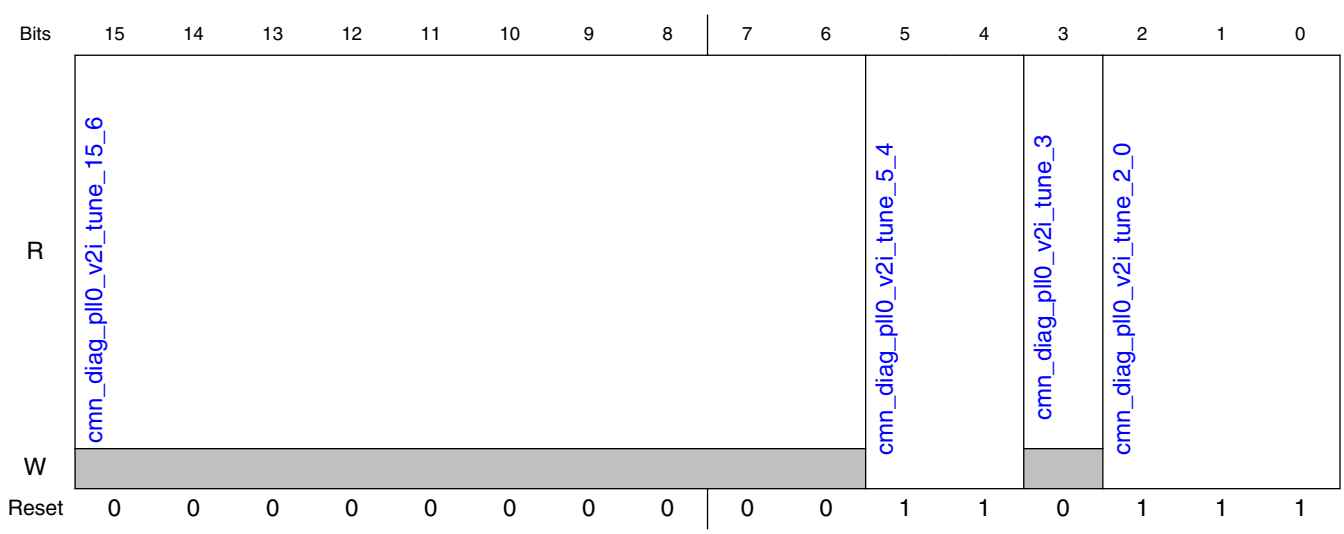
Field	Function
3 cmn_diag_pll0_test_mode_3	PLL 0 high speed ripple divider select: Test mode control of the VCO/2 clock to HS ripple divider, by driving the
2 cmn_diag_pll0_test_mode_2	PLL 0 secondary loop amp bypass enable: Test mode control of the secondary loop amp, by driving the
1 cmn_diag_pll0_test_mode_1	PLL 0 secondary loop gain control: Controls the secondary loop gain, by driving the
0 cmn_diag_pll0_test_mode_0	PLL 0 test mode charge pump current select: Test mode control of the charge pump current for the ATB cell, by driving the

13.4.10.2.523 PLL 0 voltage to current unit tuning register (cmn_diag_pll0_v2i_tune)

13.4.10.2.523.1 Offset

Register	Offset
cmn_diag_pll0_v2i_tune	8_01C5h

13.4.10.2.523.2 Diagram



13.4.10.2.523.3 Fields

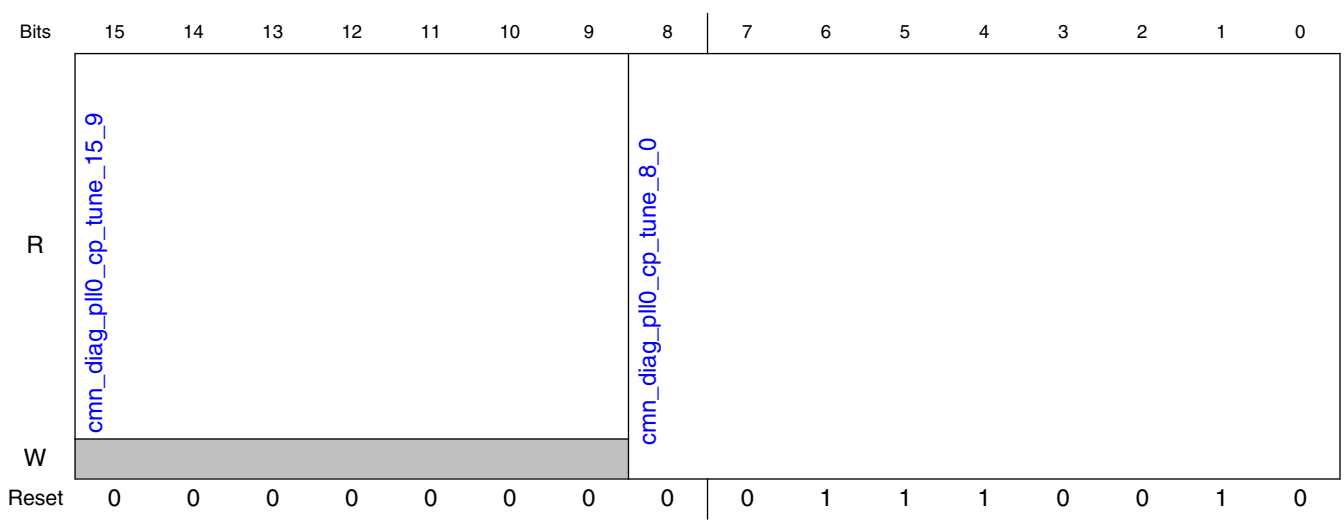
Field	Function
15-6 cmn_diag_pll0_v2i_tune_15_6	Reserved
5-4 cmn_diag_pll0_v2i_tune_5_4	PLL 0 voltage to current program code: Voltage to current program code used to multiply the V2I unit current, by driving the
3 cmn_diag_pll0_v2i_tune_3	Reserved
2-0 cmn_diag_pll0_v2i_tune_2_0	PLL 0 voltage to current coarse program code: Voltage to current coarse program code used to program CDAC/V2I unit current, by driving the

13.4.10.2.524 PLL 0 charge pump tuning register (cmn_diag_pll0_cp_tune)

13.4.10.2.524.1 Offset

Register	Offset
cmn_diag_pll0_cp_tune	8_01C6h

13.4.10.2.524.2 Diagram



13.4.10.2.524.3 Fields

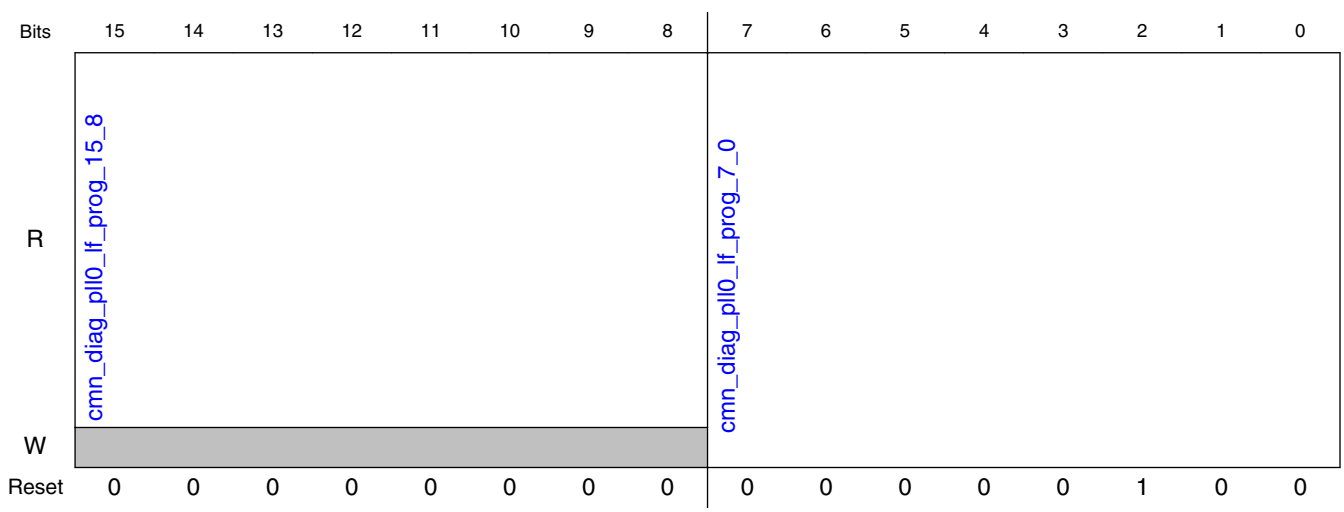
Field	Function
15-9 cmn_diag_pll0_cp_tune_15_9	Reserved
8-0 cmn_diag_pll0_cp_tune_8_0	PLL 0 charge pump gain: Controls the charge pump gain in PLL loop, by driving the

13.4.10.2.525 PLL 0 loop filter programmability register (cmn_diag_pll0_lf_prog)

13.4.10.2.525.1 Offset

Register	Offset
cmn_diag_pll0_lf_prog	8_01C7h

13.4.10.2.525.2 Diagram



13.4.10.2.525.3 Fields

Field	Function
15-8	Reserved

Table continues on the next page...

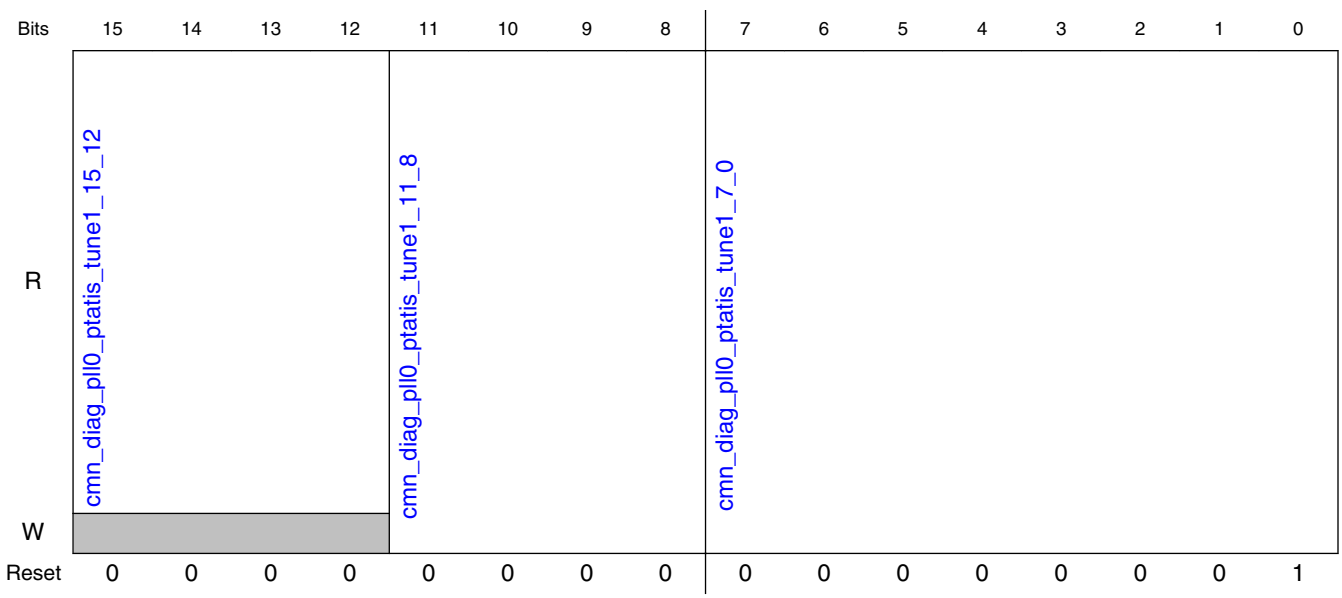
Field	Function
cmn_diag_pll0_lf_prog_15_8	
7-0 cmn_diag_pll0_lf_prog_7_0	PLL 0 loop filter programmability: Loop filter binary programmability is used for following purposes.

13.4.10.2.526 PLL 0 PTAT current slope tuning register 1 (cmn_diag_pll0_ptatis_tune1)

13.4.10.2.526.1 Offset

Register	Offset
cmn_diag_pll0_ptatis_tune1	8_01C8h

13.4.10.2.526.2 Diagram



13.4.10.2.526.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

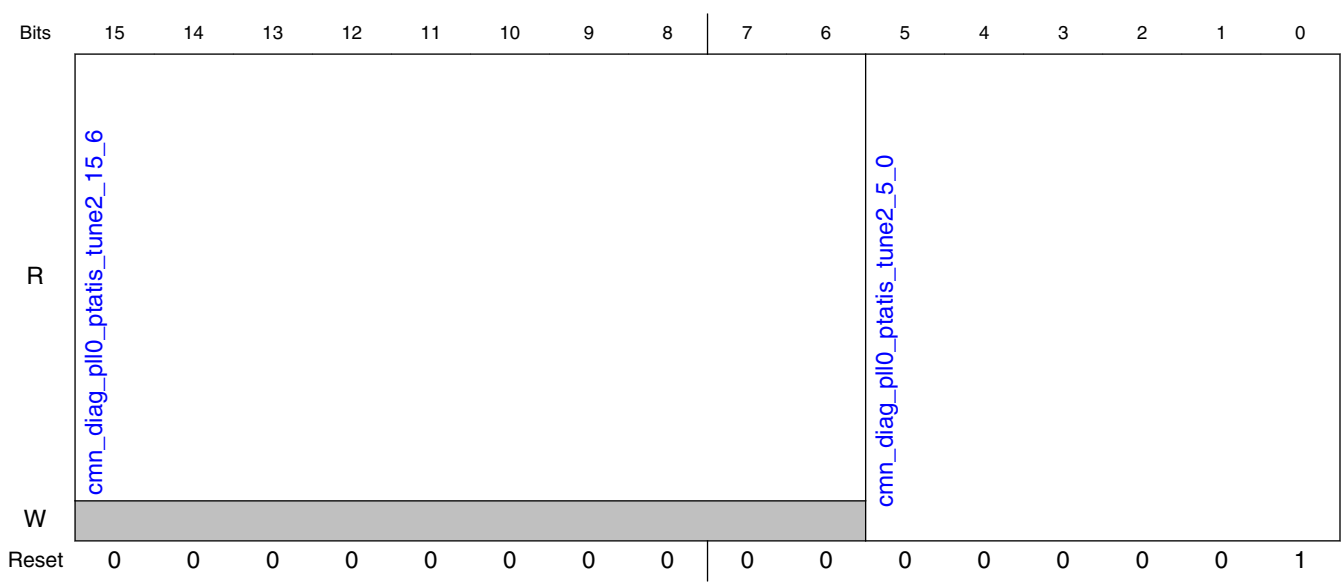
Field	Function
cmn_diag_pll0_ptatis_tune1_15_12	
11-8 cmn_diag_pll0_ptatis_tune1_11_8	PLL 0 NDAC control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/ from bandgap PTAT current. Amount of subtracted bandgap VBGbyR current is controlled by this field, by driving the
7-0 cmn_diag_pll0_ptatis_tune1_7_0	PLL 0 PMOS control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/ from bandgap PTAT current. Amount of subtracted bandgap VBGbyR current is controlled by this field, by driving the

13.4.10.2.527 PLL 0 PTAT current slope tuning register 2 (cmn_diag_pll0_ptatis_tune2)

13.4.10.2.527.1 Offset

Register	Offset
cmn_diag_pll0_ptatis_tune2	8_01C9h

13.4.10.2.527.2 Diagram



13.4.10.2.527.3 Fields

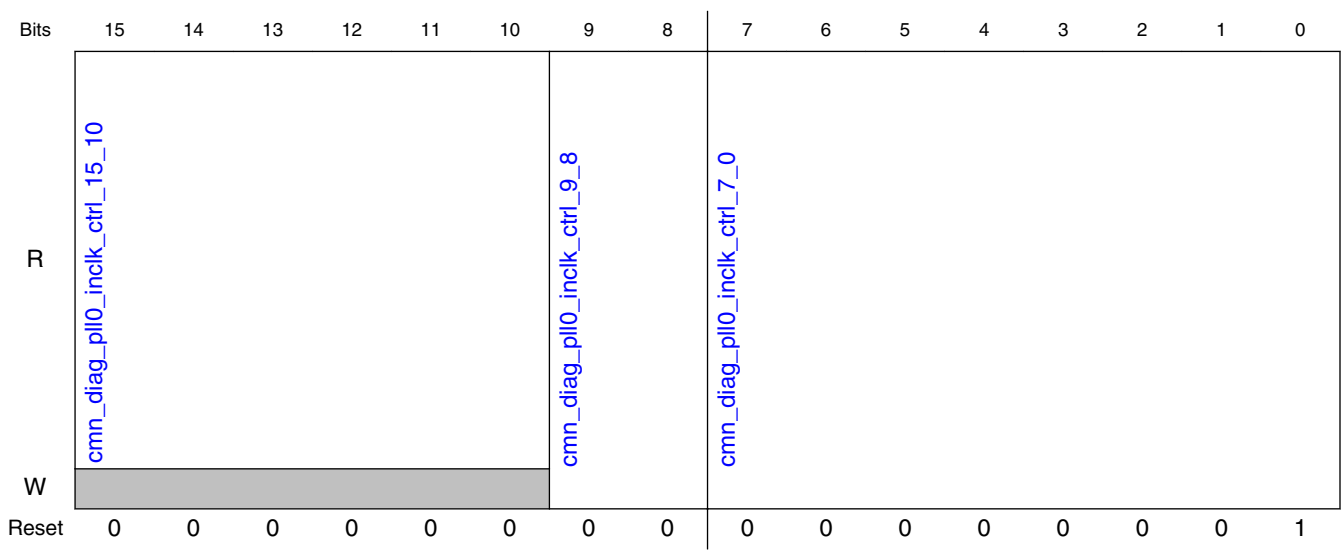
Field	Function
15-6 cmn_diag_pll0_ptatis_tune2_15_6	Reserved
5-0 cmn_diag_pll0_ptatis_tune2_5_0	PLL 0 PTAT NDAC control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/from bandgap PTAT current. After addition/subtraction the final current sent PTAT NDAC DAC to make sure output typical current is 10uA. This field controls this function, by driving the

13.4.10.2.528 PLL 0 input clock control register (cmn_diag_pll0_inclk_ctrl)

13.4.10.2.528.1 Offset

Register	Offset
cmn_diag_pll0_inclk_ctrl	8_01CAh

13.4.10.2.528.2 Diagram



13.4.10.2.528.3 Fields

Field	Function
15-10	Reserved

Table continues on the next page...

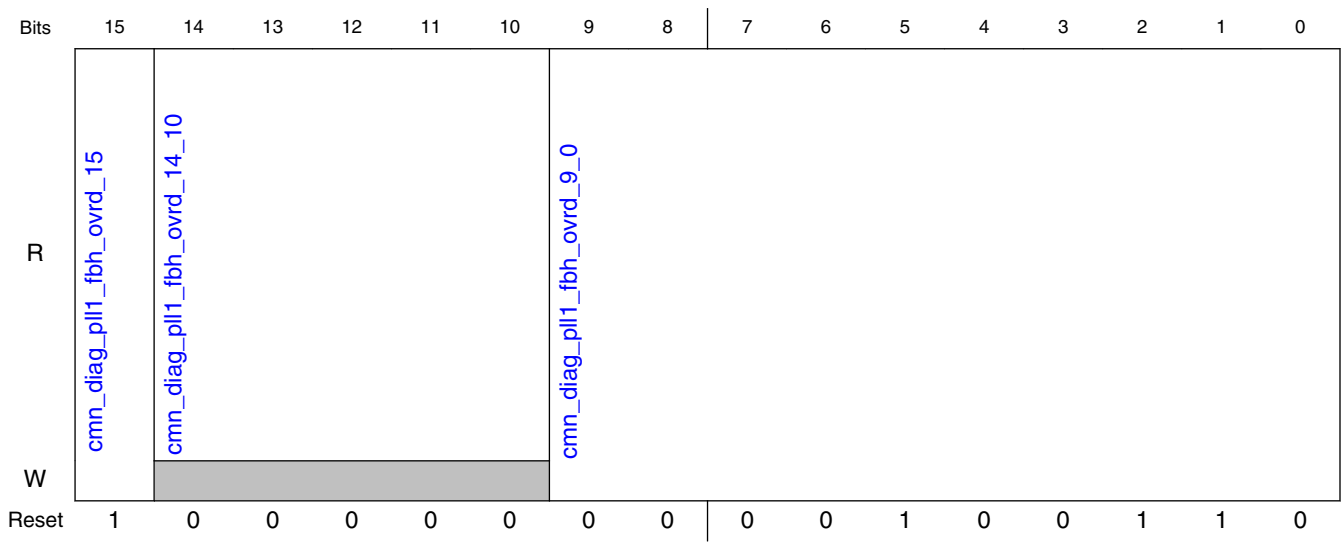
Field	Function
cmn_diag_pll0_inclk_ctrl_15_10	
9-8 cmn_diag_pll0_inclk_ctrl_9_8	PLL 0 low speed output clock divider control: Controls the division value for the input clock for the divide by 5 function used to generate the
7-0 cmn_diag_pll0_inclk_ctrl_7_0	PLL 0 input clock divider control: Controls the division value for the input reference clock divider which provides a divided reference clock to the PLL, by driving the

13.4.10.2.529 PLL 1 feedback divider value high override register (cmn_diag_pll1_fbh_ovrd)

13.4.10.2.529.1 Offset

Register	Offset
cmn_diag_pll1_fbh_ovrd	8_01D0h

13.4.10.2.529.2 Diagram



13.4.10.2.529.3 Fields

Field	Function
15	PLL feedback divider high override enable : When active (1b1), the PLL feedback divider high override value field of this register will be used to override the value on the

Table continues on the next page...

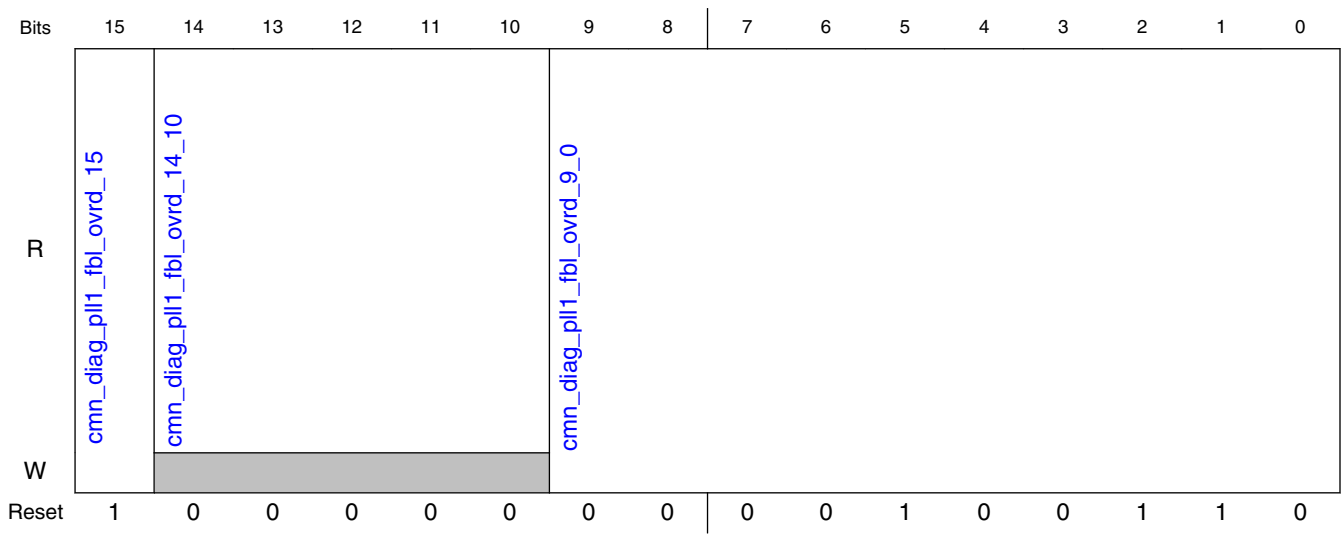
Field	Function
cmn_diag_pll1_fbh_ovrd_15	
14-10 cmn_diag_pll1_fbh_ovrd_14_10	Reserved
9-0 cmn_diag_pll1_fbh_ovrd_9_0	PLL feedback divider high override value : When enabled, the value in this field will be used to override the value on the

13.4.10.2.530 PLL 1 feedback divider value low override register (cmn_diag_pll1_fbl_ovrd)

13.4.10.2.530.1 Offset

Register	Offset
cmn_diag_pll1_fbl_ovrd	8_01D1h

13.4.10.2.530.2 Diagram



13.4.10.2.530.3 Fields

Field	Function
15	PLL feedback divider low override enable : When active (1b1), the PLL feedback divider low override value field of this register will be used to override the value on the

Table continues on the next page...

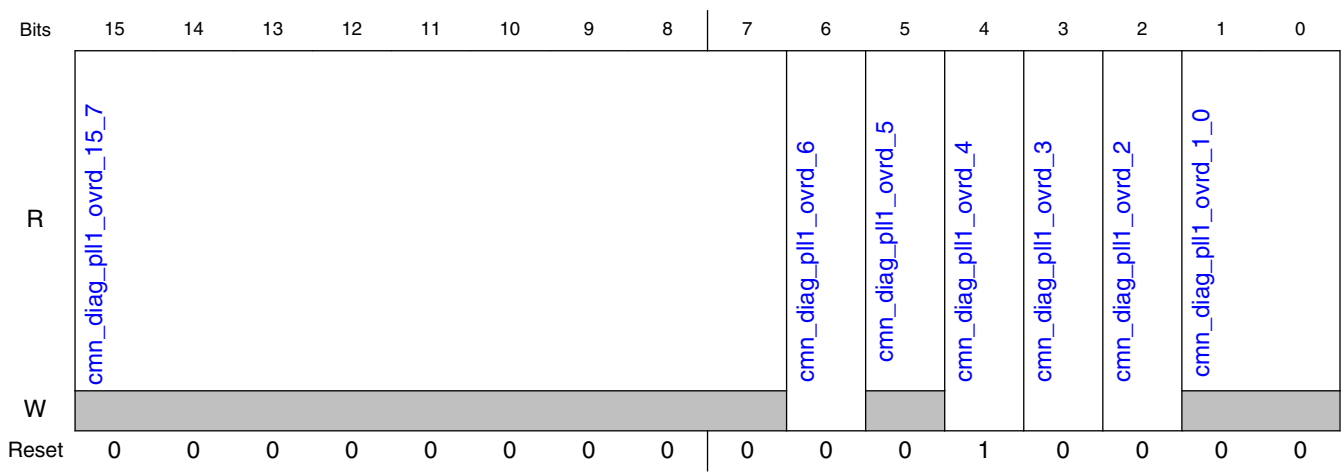
Field	Function
cmn_diag_pll1_fbl_ovrd_15	
14-10 cmn_diag_pll1_fbl_ovrd_14_10	Reserved
9-0 cmn_diag_pll1_fbl_ovrd_9_0	PLL feedback divider low override value : When enabled, the value in this field will be used to override the value on the

13.4.10.2.531 PLL 1 override register (cmn_diag_pll1_ovrd)

13.4.10.2.531.1 Offset

Register	Offset
cmn_diag_pll1_ovrd	8_01D2h

13.4.10.2.531.2 Diagram



13.4.10.2.531.3 Fields

Field	Function
15-7 cmn_diag_pll1_ovrd_15_7	Reserved
6	PLL lock override: When active (1b1), this bit will force the PLL lock indication active.

Table continues on the next page...

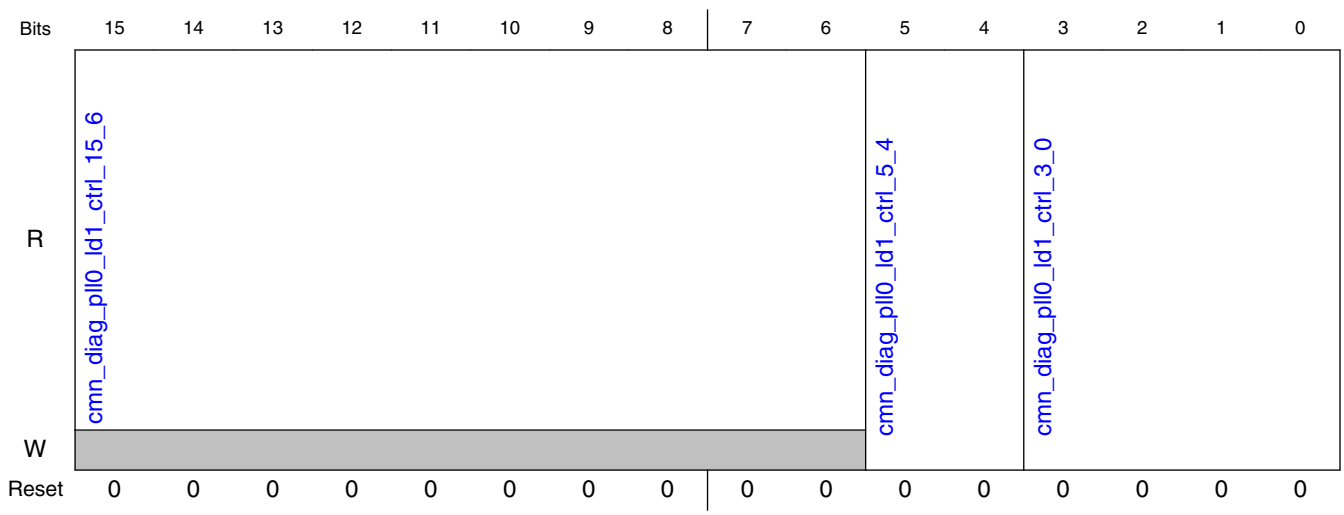
Field	Function
cmn_diag_pll1_ovrd_6	
5 cmn_diag_pll1_ovrd_5	Reserved
4 cmn_diag_pll1_ovrd_4	PLL PCIe mode: Selects corresponding PCIe CP and Loop filter block sections, by driving the
3 cmn_diag_pll1_ovrd_3	PLL VCO calibration enable override enable: When active (1b1), the PLL VCO calibration enable override bit in this register, can be used to directly control the enable of the VCO calibration function in the PLL (instead of the VCO calibration module).
2 cmn_diag_pll1_ovrd_2	PLL VCO calibration enable override: When enabled by the PLL VCO calibration enable override enable bit in this register, this bit will directly control the enable of the VCO calibration function in the PLL.
1-0 cmn_diag_pll1_ovrd_1_0	Reserved

13.4.10.2.532 PLL 1 LDO control register (cmn_diag_pll0_ld1_ctrl)

13.4.10.2.532.1 Offset

Register	Offset
cmn_diag_pll0_ld1_ctrl	8_01D3h

13.4.10.2.532.2 Diagram



13.4.10.2.532.3 Fields

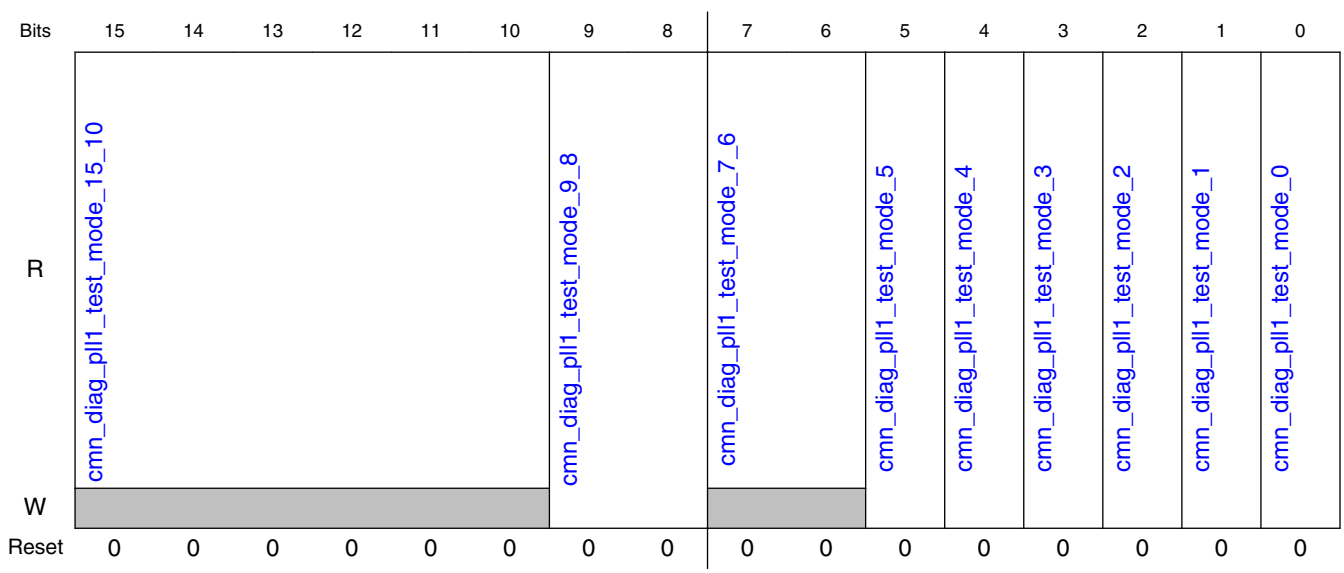
Field	Function
15-6 cmn_diag_pll0_ld1_ctrl_15_6	Reserved
5-4 cmn_diag_pll0_ld1_ctrl_5_4	PLL 1 LDO output bypass selection: When the optional PLL voltage regulator is present, this binary code controls the output bypass selection.
3-0 cmn_diag_pll0_ld1_ctrl_3_0	PLL 1 LDO nominal output voltage: When the optional PLL voltage regulator is present in the design, this code controls the regulator's nominal output voltage.

13.4.10.2.533 PLL 1 test mode register (cmn_diag_pll1_test_mode)

13.4.10.2.533.1 Offset

Register	Offset
cmn_diag_pll1_test_mode	8_01D4h

13.4.10.2.533.2 Diagram



13.4.10.2.533.3 Fields

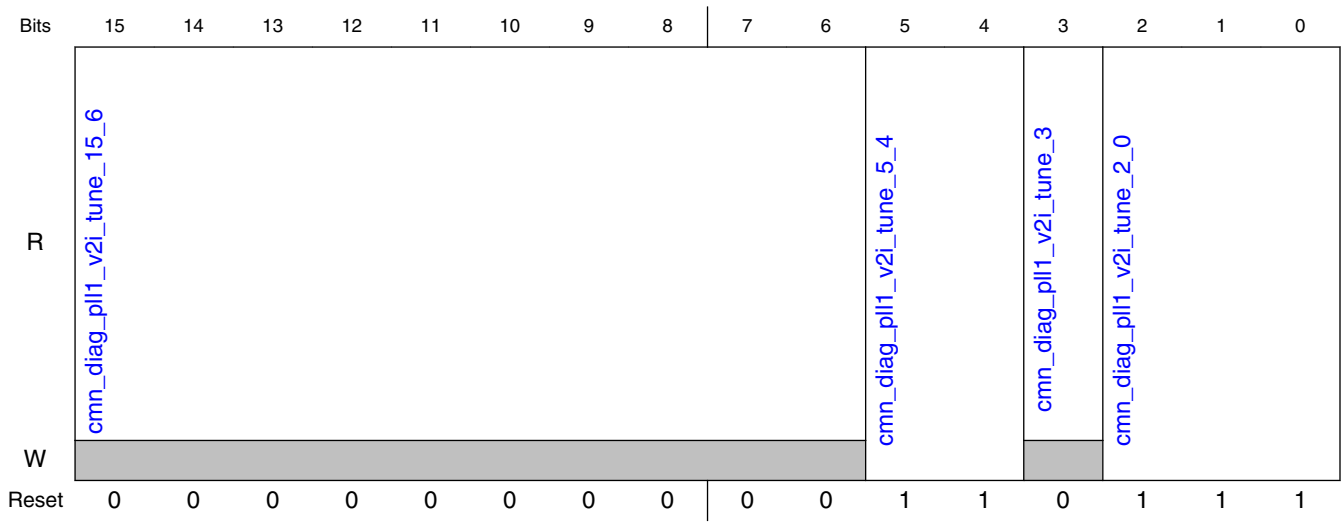
Field	Function
15-10 cmn_diag_pll1_test_mode_15_10	Reserved
9-8 cmn_diag_pll1_test_mode_9_8	PLL 1 secondary loop bandwidth control: Controls the bandwidth of the second order loop, by driving the
7-6 cmn_diag_pll1_test_mode_7_6	Reserved
5 cmn_diag_pll1_test_mode_5	PLL 1 VCO clock divider test mode select: During VCO calibration, the VCO clock is divided by 2 before being sent to the feedback divider, by driving the
4 cmn_diag_pll1_test_mode_4	PLL 1 secondary loop bypass: Controls the bypass secondary loop output filter, by driving the
3 cmn_diag_pll1_test_mode_3	PLL 1 high speed ripple divider select: Test mode control of the VCO/2 clock to HS ripple divider, by driving the
2 cmn_diag_pll1_test_mode_2	PLL 1 secondary loop amp bypass enable: Test mode control of the secondary loop amp, by driving the
1 cmn_diag_pll1_test_mode_1	PLL 1 secondary loop gain control: Controls the secondary loop gain, by driving the
0 cmn_diag_pll1_test_mode_0	PLL 1 test mode charge pump current select: Test mode control of the charge pump current for the ATB cell, by driving the

13.4.10.2.534 PLL 1 voltage to current unit tuning register (cmn_diag_pll 1_v2i_tune)

13.4.10.2.534.1 Offset

Register	Offset
cmn_diag_pll1_v2i_tune	8_01D5h

13.4.10.2.534.2 Diagram



13.4.10.2.534.3 Fields

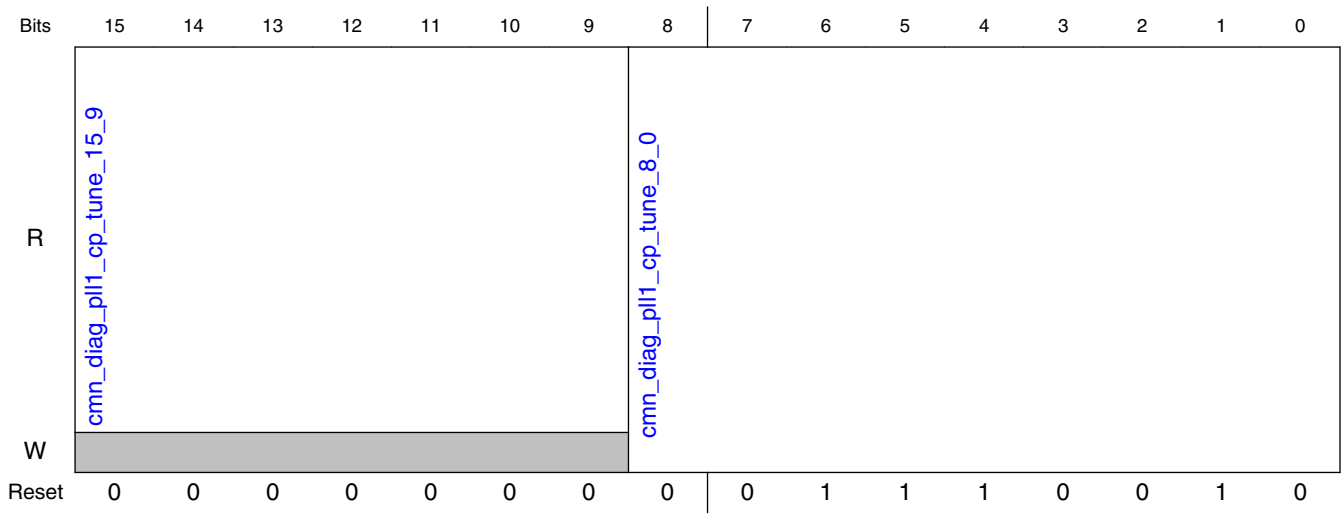
Field	Function
15-6 cmn_diag_pll1_v2i_tune_15_6	Reserved
5-4 cmn_diag_pll1_v2i_tune_5_4	PLL 0 voltage to current program code: Voltage to current program code used to multiply the V2I unit current, by driving the
3 cmn_diag_pll1_v2i_tune_3	Reserved
2-0 cmn_diag_pll1_v2i_tune_2_0	PLL 0 voltage to current coarse program code: Voltage to current coarse program code used to program CDAC/V2I unit current, by driving the

13.4.10.2.535 PLL 1 charge pump tuning register (cmn_diag_pll1_cp_tune)

13.4.10.2.535.1 Offset

Register	Offset
cmn_diag_pll1_cp_tune	8_01D6h

13.4.10.2.535.2 Diagram



13.4.10.2.535.3 Fields

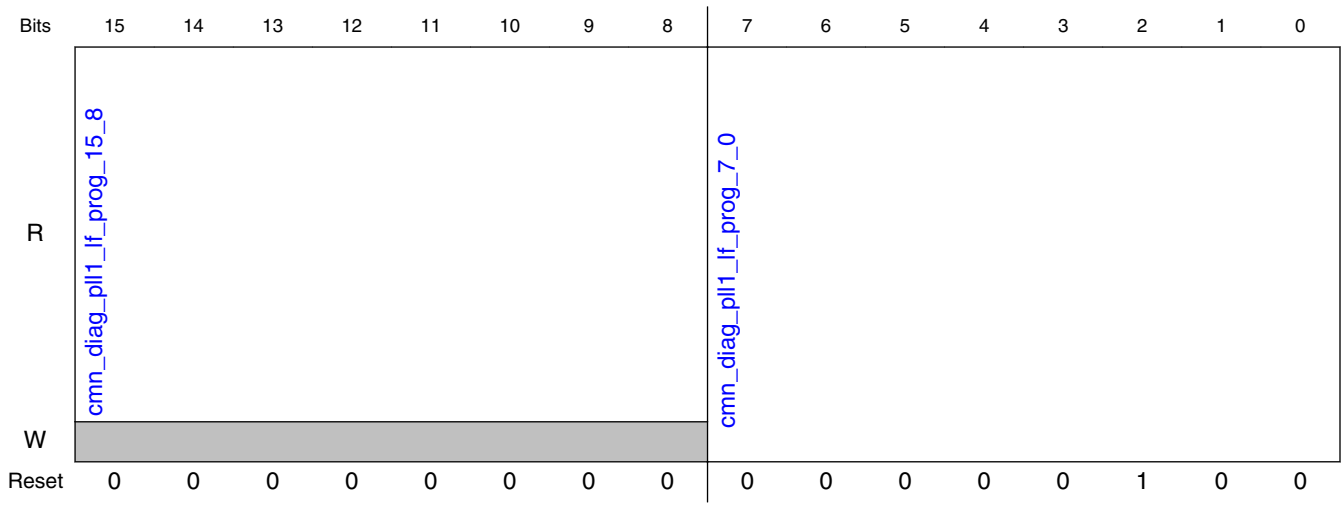
Field	Function
15-9 cmn_diag_pll1_cp_tune_15_9	Reserved
8-0 cmn_diag_pll1_cp_tune_8_0	PLL 1 charge pump gain: Controls the charge pump gain in PLL loop, by driving the

13.4.10.2.536 PLL 1 loop filter programmability register (cmn_diag_pll1_lf_prog)

13.4.10.2.536.1 Offset

Register	Offset
cmn_diag_pll1_lf_prog	8_01D7h

13.4.10.2.536.2 Diagram



13.4.10.2.536.3 Fields

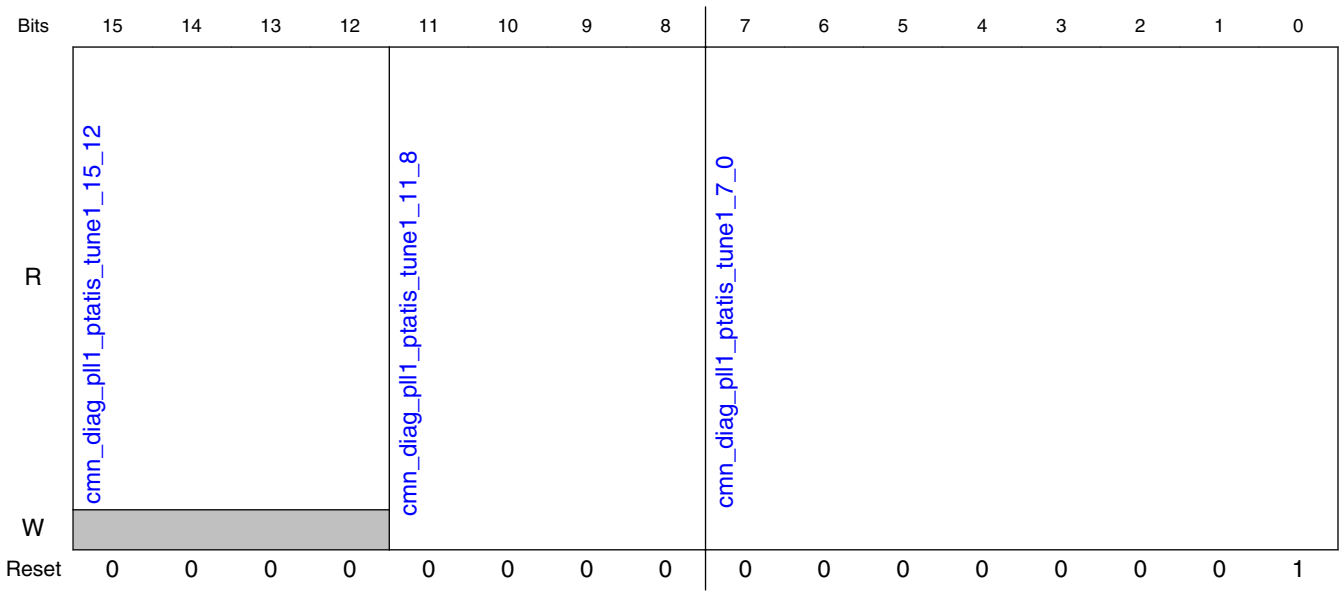
Field	Function
15-8 cmn_diag_pll1_lf_prog_15_8	Reserved
7-0 cmn_diag_pll1_lf_prog_7_0	PLL 1 loop filter programmability: Loop filter binary programmability is used for following purposes.

13.4.10.2.537 PLL 1 PTAT current slope tuning register 1 (cmn_diag_pll1_ptatis_tune1)

13.4.10.2.537.1 Offset

Register	Offset
cmn_diag_pll1_ptatis_tune1	8_01D8h

13.4.10.2.537.2 Diagram



13.4.10.2.537.3 Fields

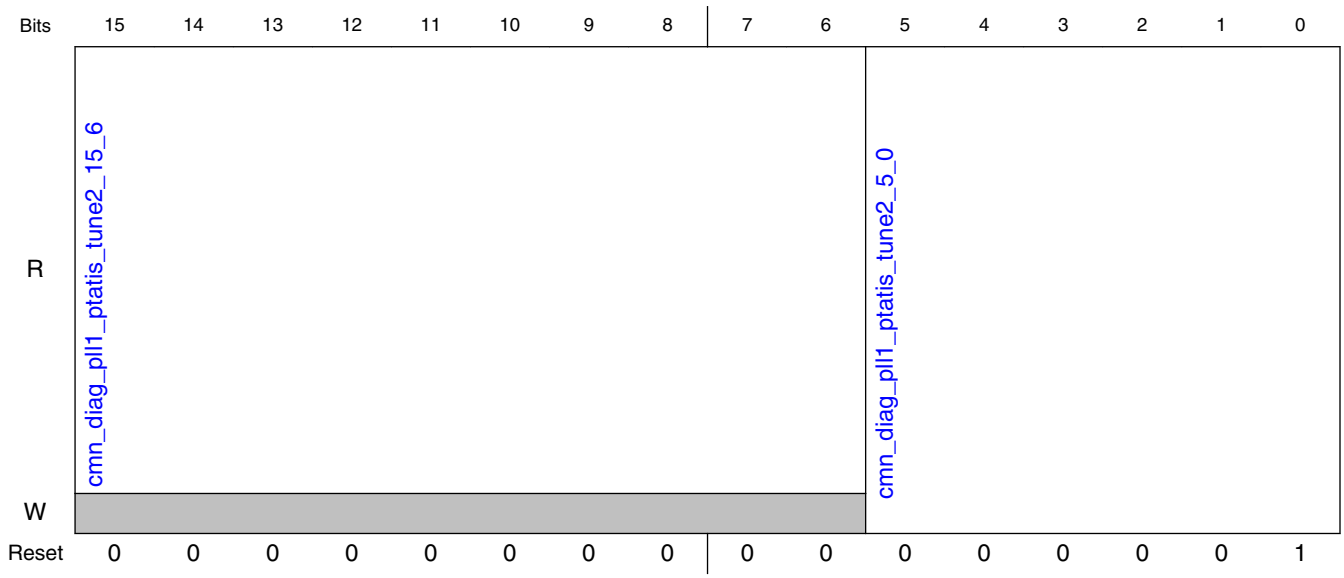
Field	Function
15-12 cmn_diag_pll1_ptatis_tune1_15_12	Reserved
11-8 cmn_diag_pll1_ptatis_tune1_11_8	PLL 1 NDAC control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/ from bandgap PTAT current. Amount of subtracted bandgap VBGbyR current is controlled by this field, by driving the
7-0 cmn_diag_pll1_ptatis_tune1_7_0	PLL 1 PMOS control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/ from bandgap PTAT current. Amount of subtracted bandgap VBGbyR current is controlled by this field, by driving the

13.4.10.2.538 PLL 1 PTAT current slope tuning register 2 (cmn_diag_pll1_ptatis_tune2)

13.4.10.2.538.1 Offset

Register	Offset
cmn_diag_pll1_ptatis_tune2	8_01D9h

13.4.10.2.538.2 Diagram



13.4.10.2.538.3 Fields

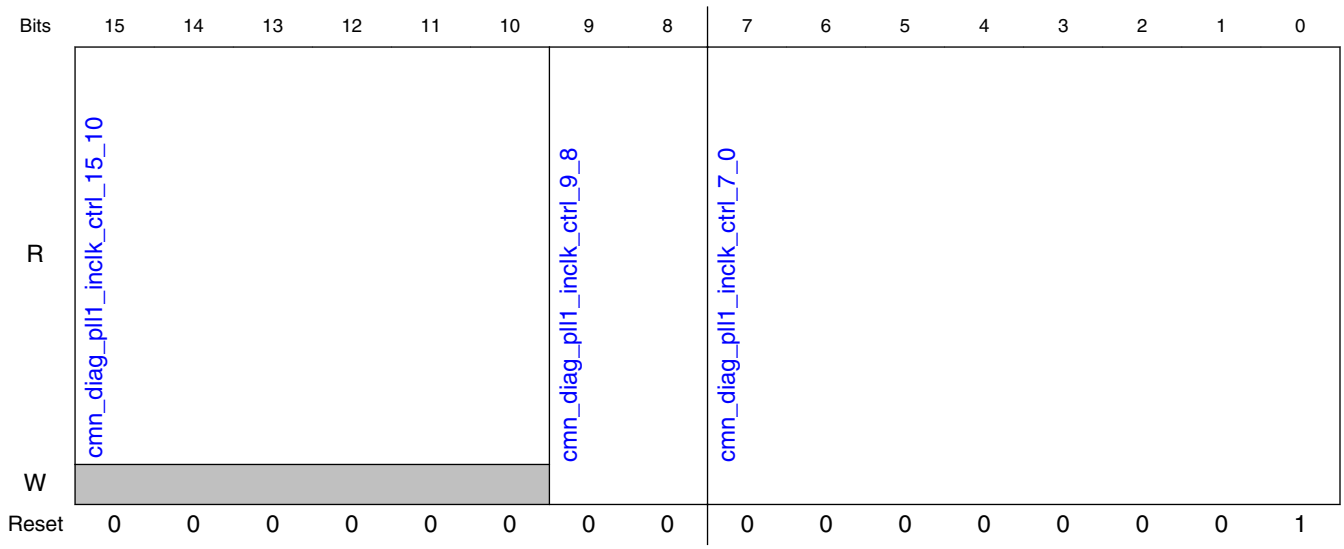
Field	Function
15-6 cmn_diag_pll1_ptatis_tune2_15_6	Reserved
5-0 cmn_diag_pll1_ptatis_tune2_5_0	PLL 1 PTAT NDAC control: PTAT current slope is varied by adding/subtracting bandgap VBGbyR current to/from bandgap PTAT current. After addition/subtraction the final current sent PTAT NDAC DAC to make sure output typical current is 10uA. This field controls this function, by driving the

13.4.10.2.539 PLL 1 input clock control register (cmn_diag_pll1_inclk_ctrl)

13.4.10.2.539.1 Offset

Register	Offset
cmn_diag_pll1_inclk_ctrl	8_01DAh

13.4.10.2.539.2 Diagram



13.4.10.2.539.3 Fields

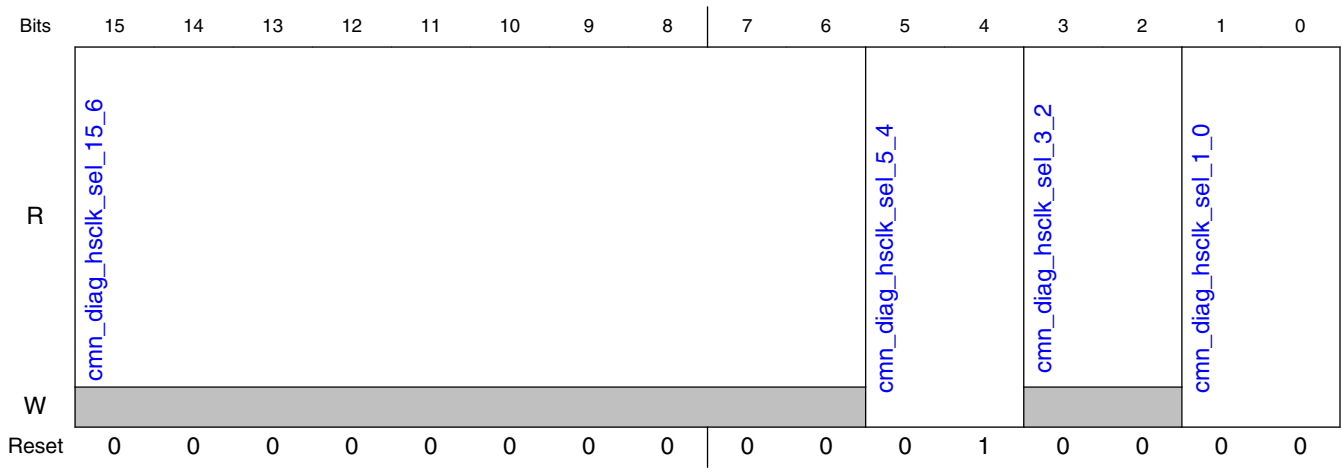
Field	Function
15-10 cmn_diag_pll1_inclk_ctrl_15_10	Reserved
9-8 cmn_diag_pll1_inclk_ctrl_9_8	PLL 1 low speed output clock divider control: Controls the division value for the input clock for the divide by 5 function used to generate the
7-0 cmn_diag_pll1_inclk_ctrl_7_0	PLL 1 input clock divider control: Controls the division value for the input reference clock divider which provides a divided reference clock to the PLL, by driving the

13.4.10.2.540 Common high speed clock select register (cmn_diag_hsclk_sel)

13.4.10.2.540.1 Offset

Register	Offset
cmn_diag_hsclk_sel	8_01E0h

13.4.10.2.540.2 Diagram



13.4.10.2.540.3 Fields

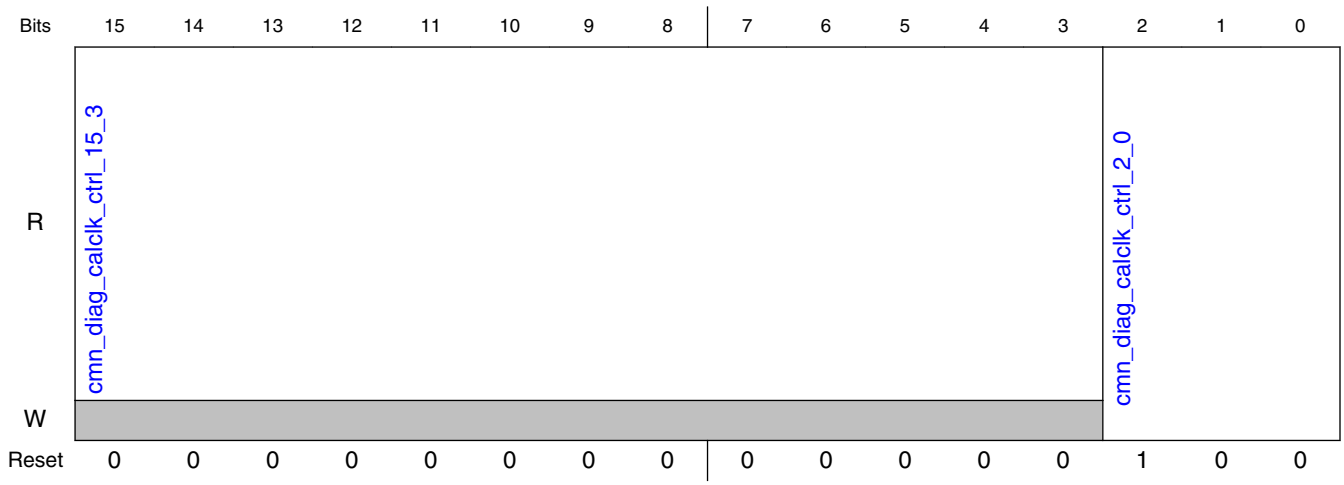
Field	Function
15-6 cmn_diag_hsclk_sel_15_6	Reserved
5-4 cmn_diag_hsclk_sel_5_4	High speed clock 1 select: Selects which PLL clock will be driven on the analog high speed clock 1.
3-2 cmn_diag_hsclk_sel_3_2	Reserved
1-0 cmn_diag_hsclk_sel_1_0	High speed clock 0 select: Selects which PLL clock will be driven on the analog high speed clock 0.

13.4.10.2.541 Common calibration clock control register (cmn_diag_cal clk_ctrl)

13.4.10.2.541.1 Offset

Register	Offset
cmn_diag_calclk_ctrl	8_01E1h

13.4.10.2.541.2 Diagram



13.4.10.2.541.3 Fields

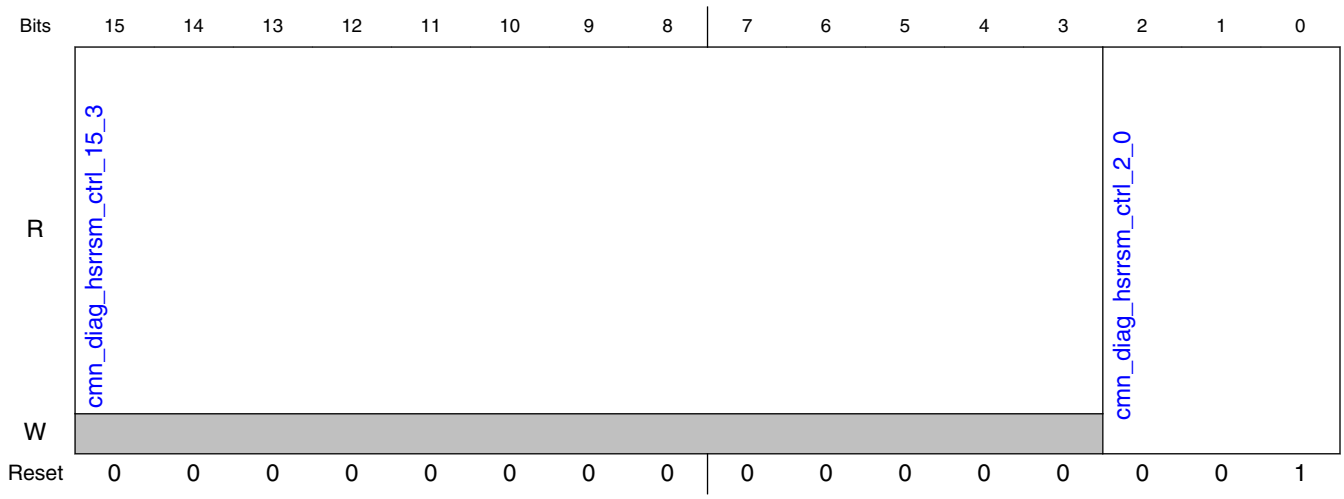
Field	Function
15-3 cmn_diag_calclk_ctrl_15_3	Reserved
2-0 cmn_diag_calclk_ctrl_2_0	Calibration clock divider select: Selects the divider setting for the calibration clock.

13.4.10.2.542 Common high speed reset release state machine control register (cmn_diag_hsrrsm_ctrl)

13.4.10.2.542.1 Offset

Register	Offset
cmn_diag_hsrrsm_ctrl	8_01E2h

13.4.10.2.542.2 Diagram



13.4.10.2.542.3 Fields

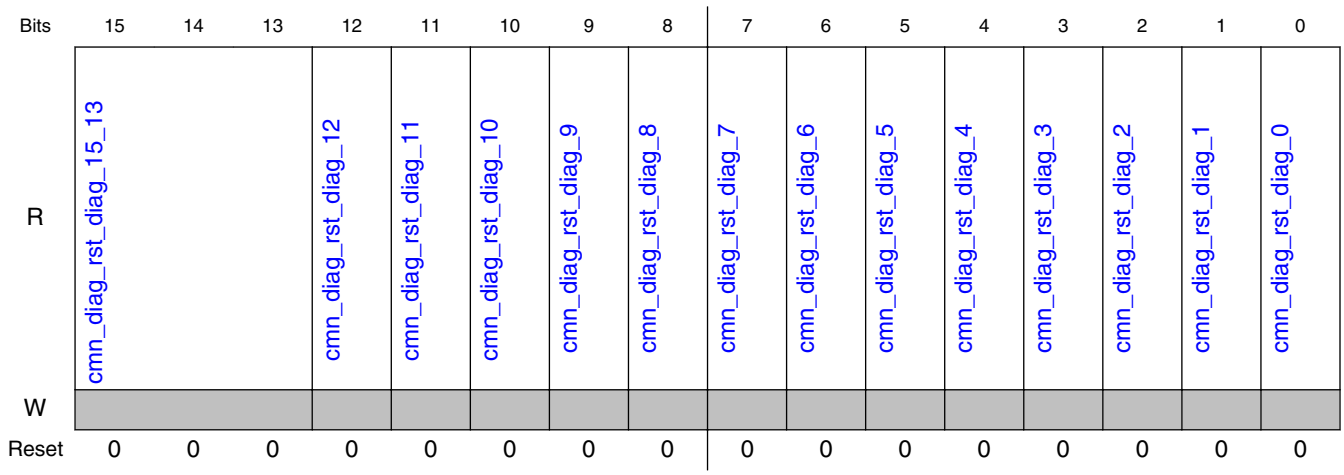
Field	Function
15-3 cmn_diag_hsrrsm_ctrl_15_3	Reserved
2-0 cmn_diag_hsrrsm_ctrl_2_0	Transmitter reset delay : Species the number of PSM clock cycles the transmitter common high speed reset state machine stays in the delay state.

13.4.10.2.543 Common functions reset diagnostic register (cmn_diag_rst_diag)

13.4.10.2.543.1 Offset

Register	Offset
cmn_diag_rst_diag	8_01E3h

13.4.10.2.543.2 Diagram



13.4.10.2.543.3 Fields

Field	Function
15-13 cmn_diag_rst_diag_15_13	Reserved
12 cmn_diag_rst_diag_12	Current state of the ssmac_power_reset_n reset.
11 cmn_diag_rst_diag_11	Reserved
10 cmn_diag_rst_diag_10	Reserved
9 cmn_diag_rst_diag_9	Current state of the cmn_pll1_dsm_reset_n reset.
8 cmn_diag_rst_diag_8	Current state of the cmn_pll0_dsm_reset_n reset.
7 cmn_diag_rst_diag_7	Current state of the cmn_pll1_vco_cal_fbdiv_clk_reset_n reset.
6 cmn_diag_rst_diag_6	Current state of the cmn_pll1_lock_det_fbdiv_clk_reset_n reset.
5	Current state of the cmn_pll1_vco_cal_ref_clk_reset_n reset.

Table continues on the next page...

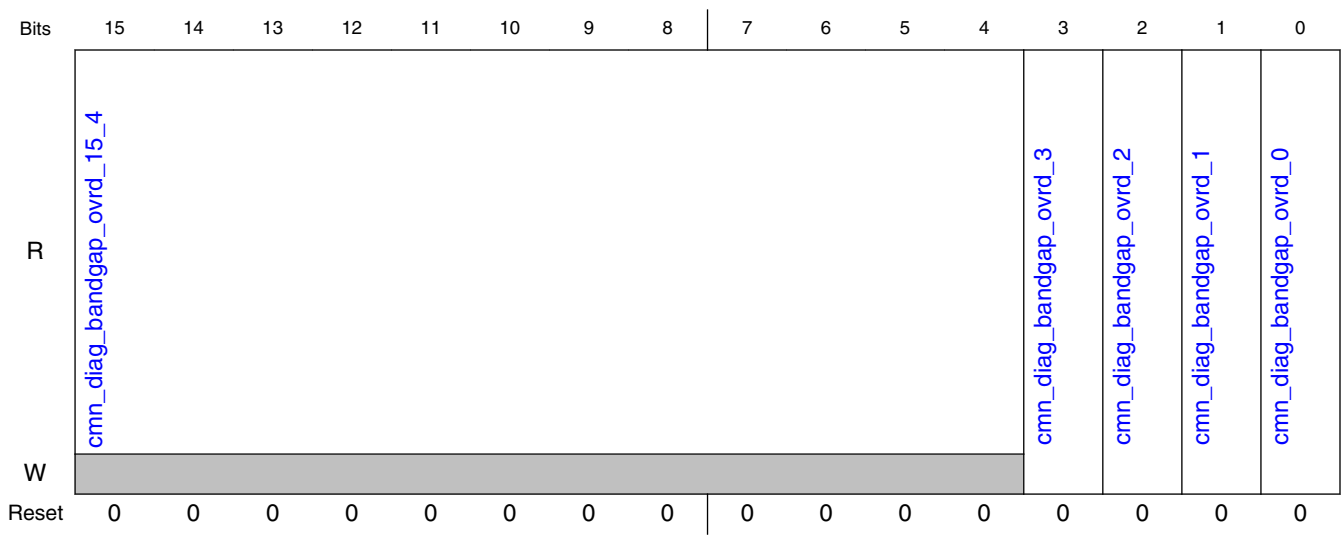
Field	Function
cmn_diag_rst_diag_5	
4 cmn_diag_rst_diag_4	Current state of the cmn_pll1_lock_det_ref_clk_reset_n reset.
3 cmn_diag_rst_diag_3	Current state of the cmn_pll0_vco_cal_fbdiv_clk_reset_n reset.
2 cmn_diag_rst_diag_2	Current state of the cmn_pll0_lock_det_fbdiv_clk_reset_n reset.
1 cmn_diag_rst_diag_1	Current state of the cmn_pll0_vco_cal_ref_clk_reset_n reset.
0 cmn_diag_rst_diag_0	Current state of the cmn_pll0_lock_det_ref_clk_reset_n reset.

13.4.10.2.544 Bandgap override register (cmn_diag_bandgap_ovrd)

13.4.10.2.544.1 Offset

Register	Offset
cmn_diag_bandgap_ovrd	8_01E8h

13.4.10.2.544.2 Diagram



13.4.10.2.544.3 Fields

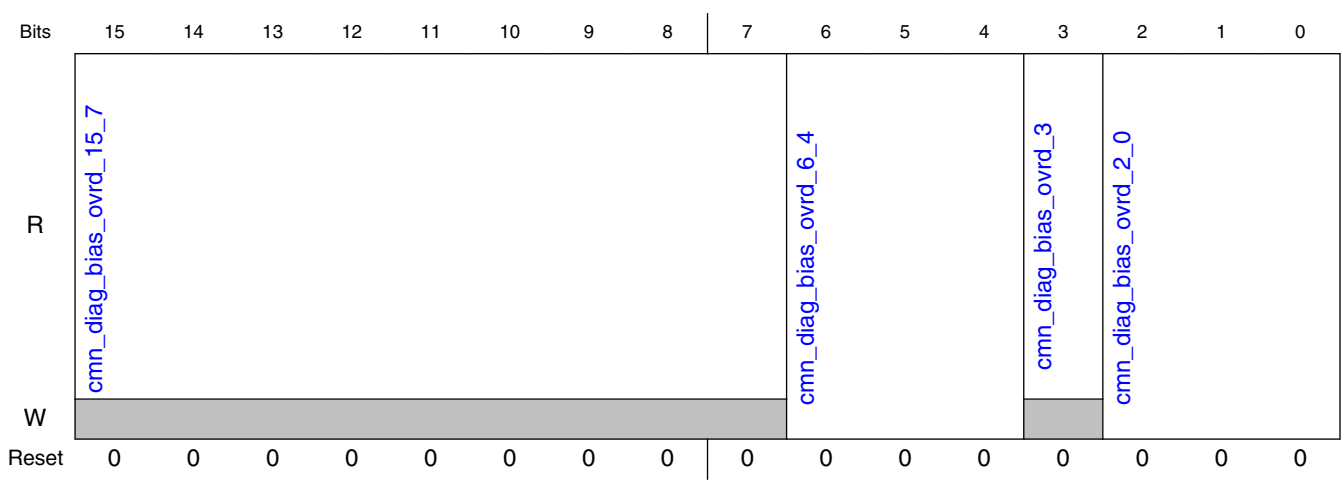
Field	Function
15-4 cmn_diag_bandgap_ovrd_15_4	Reserved
3 cmn_diag_bandgap_ovrd_3	Bandgap reference current increase: Increases the value of the crude reference current used in the bandgap startup block to determine when the bandgap startup circuitry can be disabled.
2 cmn_diag_bandgap_ovrd_2	Bandgap reference current decrease: Decreases the value of the crude reference current used in the bandgap startup block to determine when the bandgap startup circuitry can be disabled.
1 cmn_diag_bandgap_ovrd_1	Bandgap startup force on: Forces the bandgap startup circuitry to turn on. When the startup circuitry is turned on in this manner, the bandgap startup force off bit in this register shall be used to turn it off after a delay of at least 2 uSec.
0 cmn_diag_bandgap_ovrd_0	Bandgap startup force off: Forces the bandgap startup circuitry to turn off. This signal has higher priority than bandgap startup force on bit in this register.

13.4.10.2.545 Bias override register (cmn_diag_bias_ovrd)

13.4.10.2.545.1 Offset

Register	Offset
cmn_diag_bias_ovrd	8_01E9h

13.4.10.2.545.2 Diagram



13.4.10.2.545.3 Fields

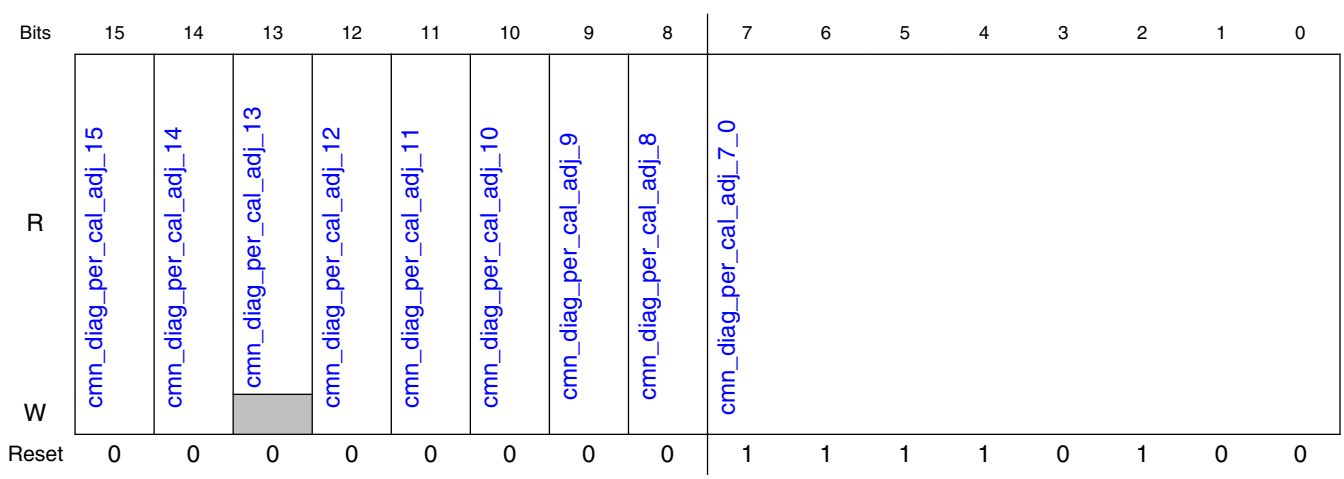
Field	Function
15-7 cmn_diag_bias_ovrd_15_7	Reserved
6-4 cmn_diag_bias_ovrd_6_4	Receiver resistor calibration current adjust: This field is used to adjust the receiver resistor calibration bias current. It drives the
3 cmn_diag_bias_ovrd_3	Reserved
2-0 cmn_diag_bias_ovrd_2_0	Transmitter resistor calibration current adjust: This field is used to adjust the transmitter resistor calibration bias current. It drives the

13.4.10.2.546 Common periodic calibration adjust control register (cmn_diag_per_cal_adj)

13.4.10.2.546.1 Offset

Register	Offset
cmn_diag_per_cal_adj	8_01ECh

13.4.10.2.546.2 Diagram



13.4.10.2.546.3 Fields

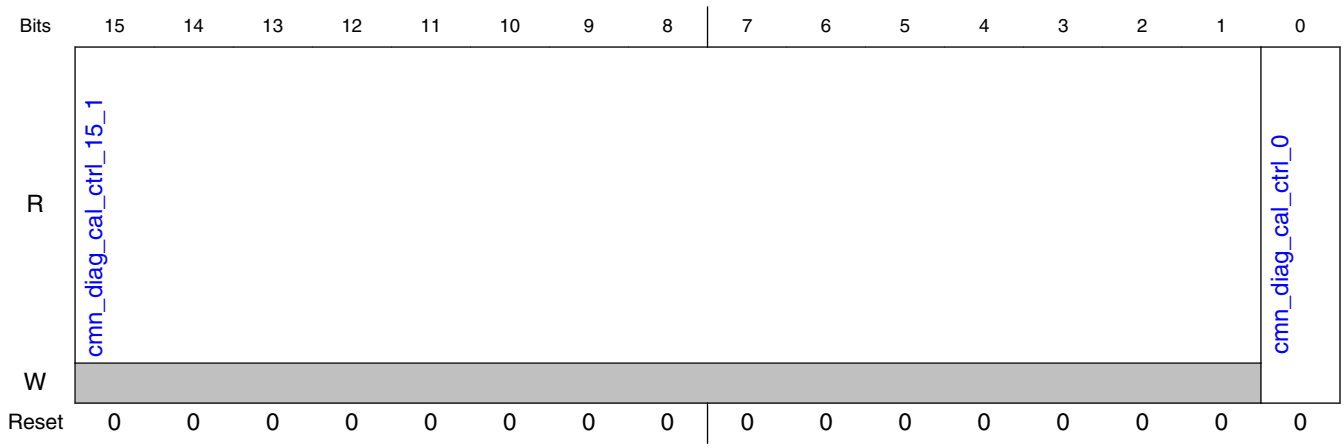
Field	Function
15 cmn_diag_per_cal_adj_15	Calibration latch enable override enable : When active (1b1), the calibration latch enable override bit in this register will drive the
14 cmn_diag_per_cal_adj_14	Calibration latch enable override : When enabled by the calibration latch enable override enable bit in this register, this bit will drive the
13 cmn_diag_per_cal_adj_13	Reserved
12 cmn_diag_per_cal_adj_12	Transmitter termination pull up calibration adjust enable: Enables the transmitter pull up termination calibration adjust function, when the periodic calibration adjust function is enabled.
11 cmn_diag_per_cal_adj_11	Transmitter termination pull down calibration adjust enable: Enables the transmitter pull down termination calibration adjust function, when the periodic calibration adjust function is enabled.
10 cmn_diag_per_cal_adj_10	Receiver termination calibration adjust enable: Enables the receiver termination calibration adjust function, when the periodic calibration adjust function is enabled.
9 cmn_diag_per_cal_adj_9	Current calibration adjust enable: Enables the current calibration adjust function, when the periodic calibration adjust function is enabled.
8 cmn_diag_per_cal_adj_8	Periodic calibration adjust enable: Enables the periodic calibration adjust function.
7-0 cmn_diag_per_cal_adj_7_0	Periodic calibration timer value: This is the value of the most significant 8 bits of the 16 bit periodic calibration timer.

13.4.10.2.547 Common calibration control register (cmn_diag_cal_ctrl)

13.4.10.2.547.1 Offset

Register	Offset
cmn_diag_cal_ctrl	8_01EDh

13.4.10.2.547.2 Diagram



13.4.10.2.547.3 Fields

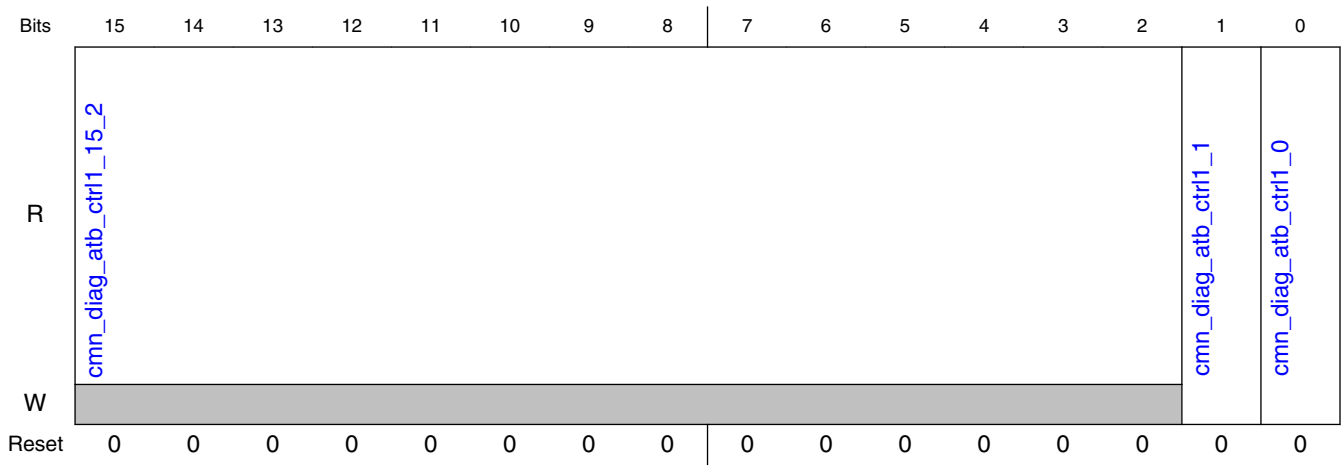
Field	Function
15-1 cmn_diag_cal_ctrl_15_1	Reserved
0 cmn_diag_cal_ctrl_0	Resistor calibration low swing select: In some protocols the TX calibration needs to be performed for a low swing output. This bit indicates that low swing calibration should be performed, by driving the

13.4.10.2.548 ATB control register 1 (cmn_diag_atb_ctrl1)

13.4.10.2.548.1 Offset

Register	Offset
cmn_diag_atb_ctrl1	8_01F0h

13.4.10.2.548.2 Diagram



13.4.10.2.548.3 Fields

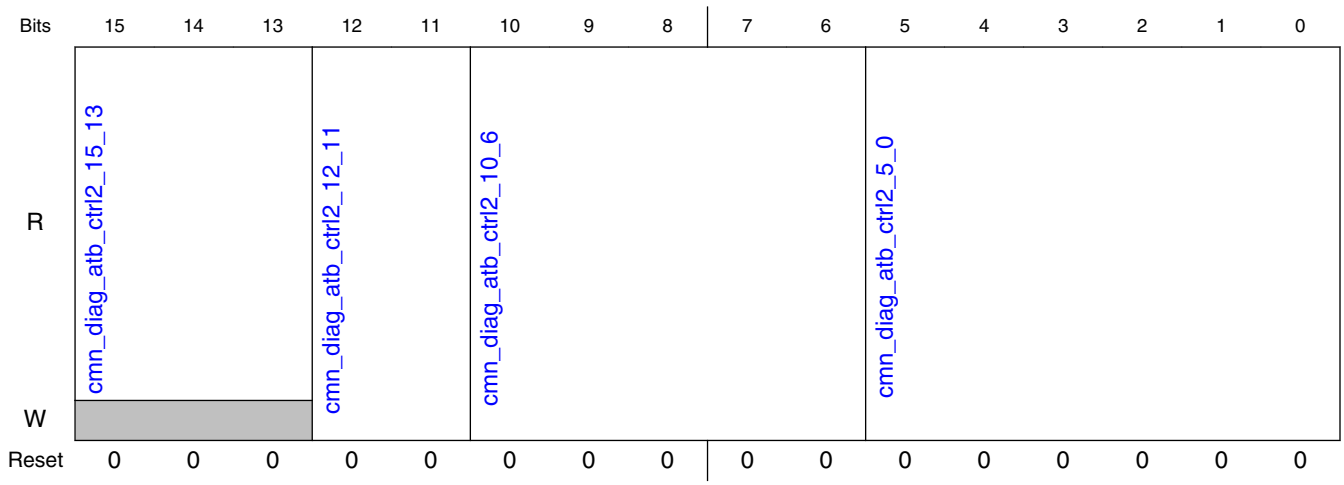
Field	Function
15-2 cmn_diag_atb_ctrl1_15_2	Reserved
1 cmn_diag_atb_ctrl1_1	ATB source select: Selects between the two source domains that can drive the ATB signals, by driving the
0 cmn_diag_atb_ctrl1_0	ATB enable: When active (1b1), the ATB test function is enabled.

13.4.10.2.549 ATB control register 2 (cmn_diag_atb_ctrl2)

13.4.10.2.549.1 Offset

Register	Offset
cmn_diag_atb_ctrl2	8_01F1h

13.4.10.2.549.2 Diagram



13.4.10.2.549.3 Fields

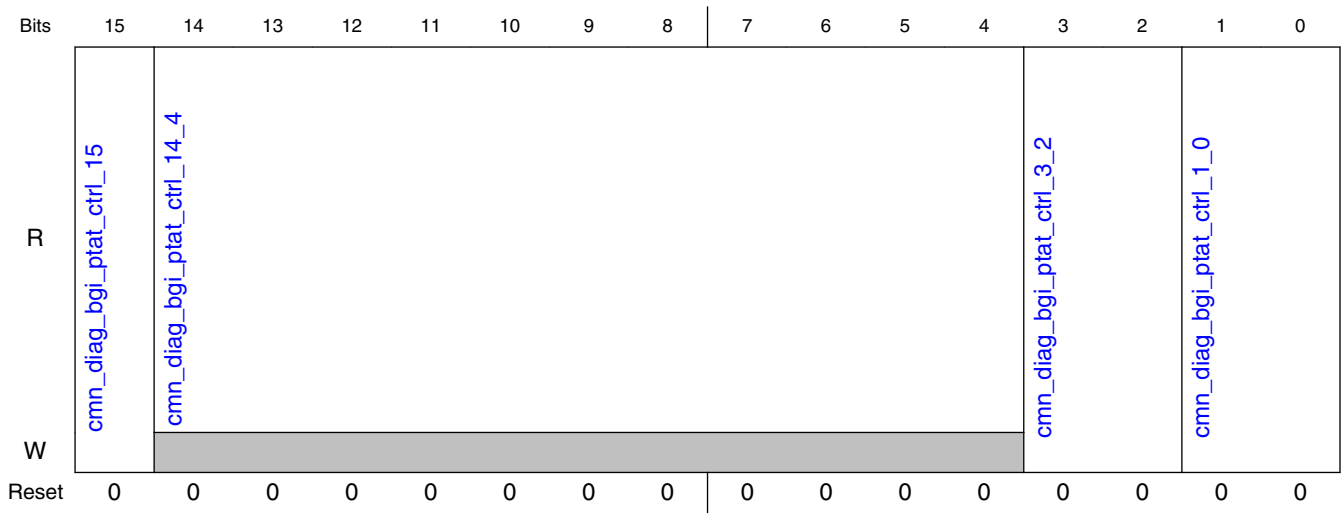
Field	Function
15-13 cmn_diag_atb_ctrl2_15_13	Reserved
12-11 cmn_diag_atb_ctrl2_12_11	ATB component type select: These bits specify which component type is currently selected by the ATB, as specified below.
10-6 cmn_diag_atb_ctrl2_10_6	ATB component sub address: Specifies the sub address of the component being selected. In this design, the sub address must be 0 when accessing the common, and the lane number when accessing a transmitter or receiver.
5-0 cmn_diag_atb_ctrl2_5_0	ATB test point address: Specifies the exact point in the selected analog component to be observed.

13.4.10.2.550 Common bandgap PTAT current control register (cmn_diag_bgi_ptat_ctrl)

13.4.10.2.550.1 Offset

Register	Offset
cmn_diag_bgi_ptat_ctrl	8_01F4h

13.4.10.2.550.2 Diagram



13.4.10.2.550.3 Fields

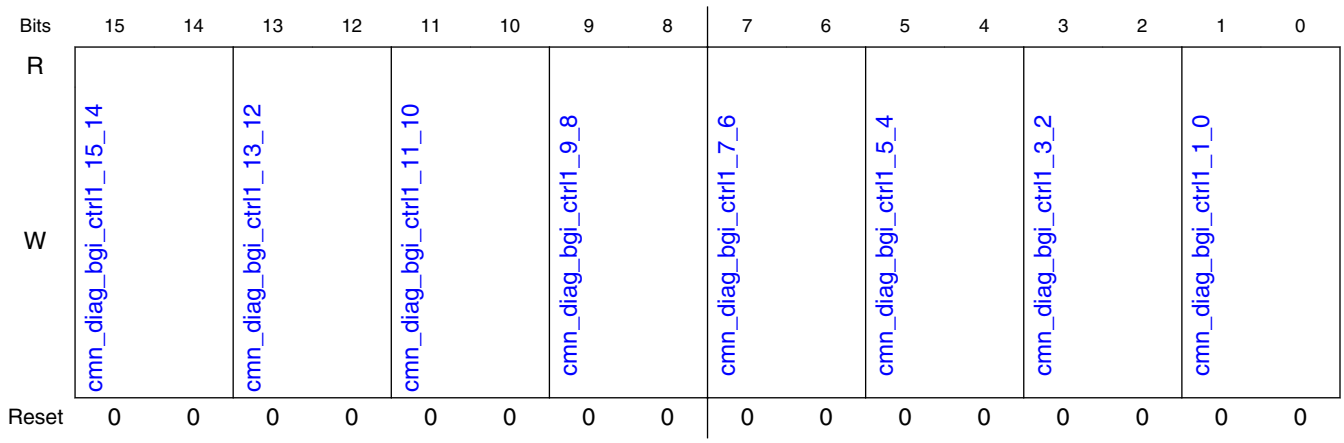
Field	Function
15 cmn_diag_bgi_ptat_ctrl_15	Bandgap PTAT current base unit enable: Enables the base unit on all the PTAT current outputs from the bandgap.
14-4 cmn_diag_bgi_ptat_ctrl_14_4	Reserved
3-2 cmn_diag_bgi_ptat_ctrl_3_2	Current programmability on the bandgap PTAT current supplying block xxx, by driving
1-0 cmn_diag_bgi_ptat_ctrl_1_0	Current programmability on the bandgap PTAT current supplying block xxx, by driving

13.4.10.2.551 Common bandgap current control register 1 (cmn_diag_bgi_ctrl1)

13.4.10.2.551.1 Offset

Register	Offset
cmn_diag_bgi_ctrl1	8_01F5h

13.4.10.2.551.2 Diagram



13.4.10.2.551.3 Fields

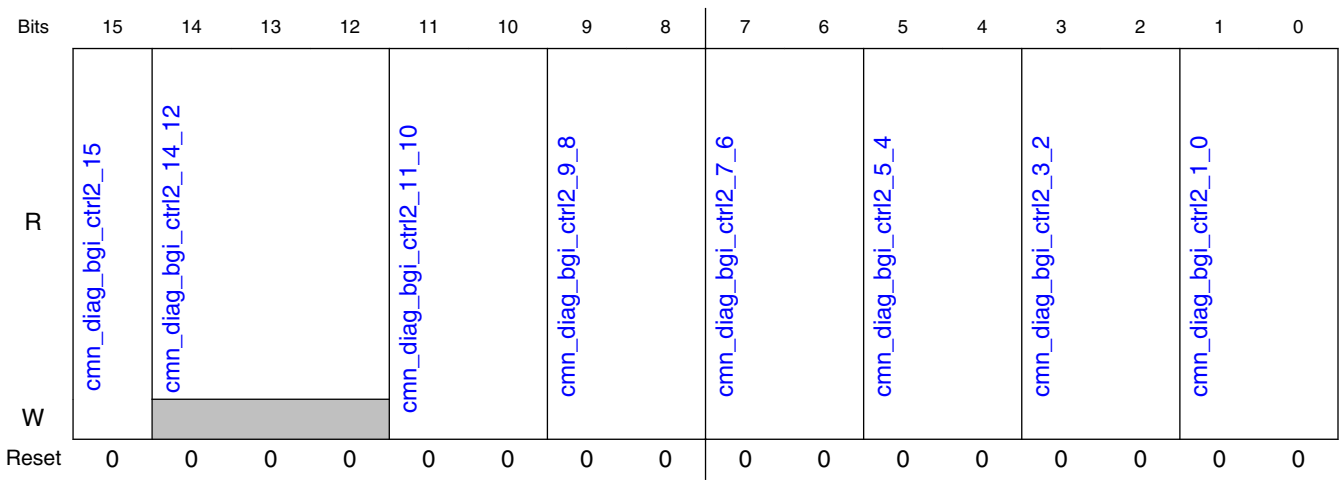
Field	Function
15-14 cmn_diag_bgi_ctrl1_15_14	Current programmability on the bandgap normal current supplying block xxx, by driving
13-12 cmn_diag_bgi_ctrl1_13_12	Current programmability on the bandgap normal current supplying block xxx, by driving
11-10 cmn_diag_bgi_ctrl1_11_10	Current programmability on the bandgap normal current supplying block xxx, by driving
9-8 cmn_diag_bgi_ctrl1_9_8	Current programmability on the bandgap normal current supplying block xxx, by driving
7-6 cmn_diag_bgi_ctrl1_7_6	Current programmability on the bandgap normal current supplying block xxx, by driving
5-4 cmn_diag_bgi_ctrl1_5_4	Current programmability on the bandgap normal current supplying block xxx, by driving
3-2 cmn_diag_bgi_ctrl1_3_2	Current programmability on the bandgap normal current supplying block xxx, by driving
1-0 cmn_diag_bgi_ctrl1_1_0	Current programmability on the bandgap normal current supplying block xxx, by driving

13.4.10.2.552 Common bandgap current control register 2 (cmn_diag_bgi_ctrl2)

13.4.10.2.552.1 Offset

Register	Offset
cmn_diag_bgi_ctrl2	8_01F6h

13.4.10.2.552.2 Diagram



13.4.10.2.552.3 Fields

Field	Function
15 cmn_diag_bgi_ctrl2_15	Bandgap normal current base unit enable: Enables the base unit on all the normal current outputs from the bandgap.
14-12 cmn_diag_bgi_ctrl2_14_12	Reserved
11-10 cmn_diag_bgi_ctrl2_11_10	Current programmability on the bandgap normal current supplying block xxx, by driving
9-8 cmn_diag_bgi_ctrl2_9_8	Current programmability on the bandgap normal current supplying block xxx, by driving
7-6 cmn_diag_bgi_ctrl2_7_6	Current programmability on the bandgap normal current supplying block xxx, by driving

Table continues on the next page...

Field	Function
5-4 cmn_diag_bgi_ctrl2_5_4	Current programmability on the bandgap normal current supplying block xxx, by driving
3-2 cmn_diag_bgi_ctrl2_3_2	Current programmability on the bandgap normal current supplying block xxx, by driving
1-0 cmn_diag_bgi_ctrl2_1_0	Current programmability on the bandgap normal current supplying block xxx, by driving

13.4.10.2.553 Common control current control register 1 (cmn_diag_ctrli_ctrl1)

13.4.10.2.553.1 Offset

Register	Offset
cmn_diag_ctrli_ctrl1	8_01F7h

13.4.10.2.553.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cmn_diag_ctrli_ctrl1_15_14		cmn_diag_ctrli_ctrl1_13_12		cmn_diag_ctrli_ctrl1_11_10		cmn_diag_ctrli_ctrl1_9_8		cmn_diag_ctrli_ctrl1_7_6		cmn_diag_ctrli_ctrl1_5_4		cmn_diag_ctrli_ctrl1_3_2		cmn_diag_ctrli_ctrl1_1_0	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.553.3 Fields

Field	Function
15-14	Current programmability on the constant current from the calibration block supplying block xxx, by driving

Table continues on the next page...

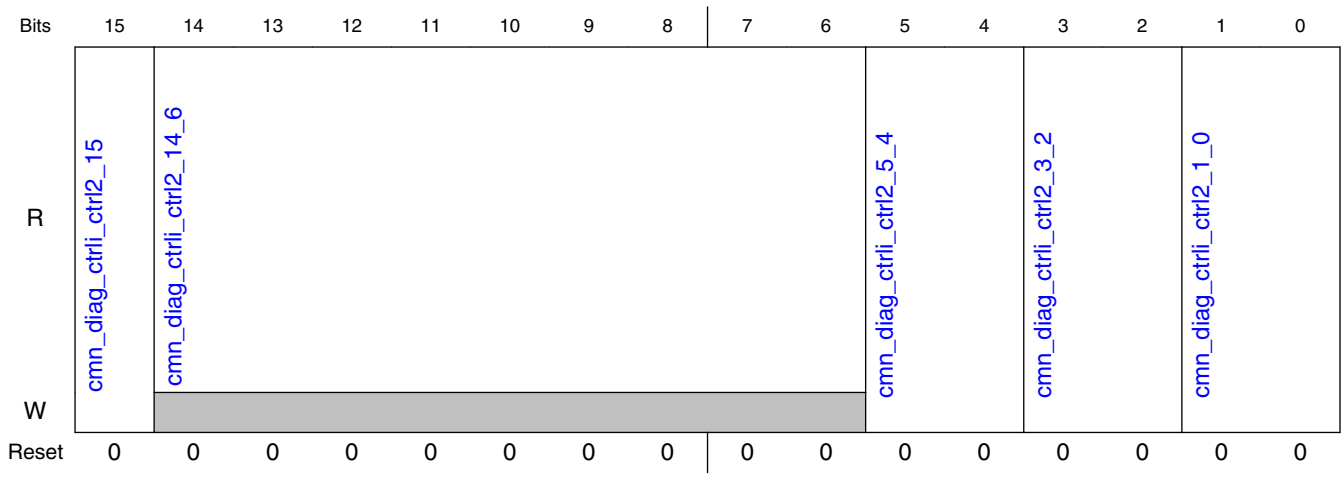
Field	Function
cmn_diag_ctrli_ctrl1_15_14	
13-12 cmn_diag_ctrli_ctrl1_13_12	Current programmability on the constant current from the calibration block supplying block xxx, by driving
11-10 cmn_diag_ctrli_ctrl1_11_10	Current programmability on the constant current from the calibration block supplying block xxx, by driving
9-8 cmn_diag_ctrli_ctrl1_9_8	Current programmability on the constant current from the calibration block supplying block xxx, by driving
7-6 cmn_diag_ctrli_ctrl1_7_6	Current programmability on the constant current from the calibration block supplying block xxx, by driving
5-4 cmn_diag_ctrli_ctrl1_5_4	Current programmability on the constant current from the calibration block supplying block xxx, by driving
3-2 cmn_diag_ctrli_ctrl1_3_2	Current programmability on the constant current from the calibration block supplying block xxx, by driving
1-0 cmn_diag_ctrli_ctrl1_1_0	Current programmability on the constant current from the calibration block supplying block xxx, by driving

13.4.10.2.554 Common control current control register 2 (cmn_diag_ctrli_ctrl2)

13.4.10.2.554.1 Offset

Register	Offset
cmn_diag_ctrli_ctrl2	8_01F8h

13.4.10.2.554.2 Diagram



13.4.10.2.554.3 Fields

Field	Function
15 cmn_diag_ctrli_ctrl2_15	Constant current base unit enable: Enables the base unit on all the normal current outputs from the bandgap.
14-6 cmn_diag_ctrli_ctrl2_14_6	Reserved
5-4 cmn_diag_ctrli_ctrl2_5_4	Current programmability on the constant current from the calibration block supplying block xxx, by driving
3-2 cmn_diag_ctrli_ctrl2_3_2	Current programmability on the constant current from the calibration block supplying block xxx, by driving
1-0 cmn_diag_ctrli_ctrl2_1_0	Current programmability on the constant current from the calibration block supplying block xxx, by driving

13.4.10.2.555 Common digital functions cover your alternatives register (cmn_diag_dcya)

13.4.10.2.555.1 Offset

Register	Offset
cmn_diag_dcya	8_01FEh

13.4.10.2.555.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cmn_diag_dcya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.555.3 Fields

Field	Function
15-0 cmn_diag_dcya_15_0	Reserved

13.4.10.2.556 Common analog cover your alternatives register (cmn_diag_acya)

13.4.10.2.556.1 Offset

Register	Offset
cmn_diag_acya	8_01FFh

13.4.10.2.556.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cmn_diag_acya_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.556.3 Fields

Field	Function
15-0 cmn_diag_acya_15_0	Reserved

13.4.10.2.557 DP Aux analog control 1 (tx_ana_ctrl_reg_1)

13.4.10.2.557.1 Offset

Register	Offset
tx_ana_ctrl_reg_1	8_5020h

13.4.10.2.557.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	tx_ana_ctrl_reg_1_15	tx_ana_ctrl_reg_1_14	tx_ana_ctrl_reg_1_13	tx_ana_ctrl_reg_1_12	tx_ana_ctrl_reg_1_11	tx_ana_ctrl_reg_1_10	tx_ana_ctrl_reg_1_9	tx_ana_ctrl_reg_1_8	tx_ana_ctrl_reg_1_7	tx_ana_ctrl_reg_1_6	tx_ana_ctrl_reg_1_5	tx_ana_ctrl_reg_1_4	tx_ana_ctrl_reg_1_3	tx_ana_ctrl_reg_1_2	tx_ana_ctrl_reg_1_1	tx_ana_ctrl_reg_1_0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.557.3 Fields

Field	Function
15 tx_ana_ctrl_reg_1_15	Controls txda_dp_aux_en,
14 tx_ana_ctrl_reg_1_14	Controls auxda_se_en
13 tx_ana_ctrl_reg_1_13	Controls txda_cal_latch_en, Enable signal for latch that sample and holds calibration values. Activate this signal for 1 clock cycle to sample new calibration values.
12 tx_ana_ctrl_reg_1_12	Controls auxda_polarity, which selects the polarity of the xcvr
11 tx_ana_ctrl_reg_1_11	Controls txda_drv_power_isolation_en, Power island isolation signal.This signal is currently unused.
10	Controls txda_drv_power_en_ph_2_n, Power island enable signals.This signal is currently unused.

Table continues on the next page...

Field	Function
tx_ana_ctrl_reg_1_10	
9 tx_ana_ctrl_reg_1_9	Comtrols txda_drv_power_en_ph_1_n, Power island enable signals.This signal is currently unused.
8 tx_ana_ctrl_reg_1_8	Controls txda_bgref_en, which enables Bandgap Reference Circuit. Must be asserted after(X=5us) txda_drv_ldo_en, else held to default value
7 tx_ana_ctrl_reg_1_7	Controls txda_drv_ldo_en, which enables the transmitter circuits related to the driver LDO function
6 tx_ana_ctrl_reg_1_6	Controls txda_decap_en_del, delayed version(tX=100ns) of tx_decap_en
5 tx_ana_ctrl_reg_1_5	Controls txda_decap_en, which enables the analog decap circuits.
4 tx_ana_ctrl_reg_1_4	Controls txda_uphy_supply_en_del, delayed version (tX=100ns) of txda_uphy_supply_en
3 tx_ana_ctrl_reg_1_3	Controls txda_uphy_supply_en, which enables the transmitter UPHY components supply
2 tx_ana_ctrl_reg_1_2	Controls txda_low_leakage_en,
1 tx_ana_ctrl_reg_1_1	Controls txda_drv_idle_lowi_en,
0 tx_ana_ctrl_reg_1_0	Controls txda_drv_cmn_mode_en, which enables the analog circuits required, to hold the driver output at a common mode voltage when required.

13.4.10.2.558 DP Aux analog control 2 (tx_ana_ctrl_reg_2)

13.4.10.2.558.1 Offset

Register	Offset
tx_ana_ctrl_reg_2	8_5021h

13.4.10.2.558.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_ana_ctrl_reg_2_15	tx_ana_ctrl_reg_2_14	tx_ana_ctrl_reg_2_13	tx_ana_ctrl_reg_2_12	tx_ana_ctrl_reg_2_11	tx_ana_ctrl_reg_2_10	tx_ana_ctrl_reg_2_9	tx_ana_ctrl_reg_2_8	tx_ana_ctrl_reg_2_7	tx_ana_ctrl_reg_2_6	tx_ana_ctrl_reg_2_5	tx_ana_ctrl_reg_2_4	tx_ana_ctrl_reg_2_3	tx_ana_ctrl_reg_2_2	tx_ana_ctrl_reg_2_1	tx_ana_ctrl_reg_2_0
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.558.3 Fields

Field	Function
15 tx_ana_ctrl_reg_2_15	Controls auxda_debouncing_clk, required in MHL mode, for data reception
14 tx_ana_ctrl_reg_2_14	Controls txda_lpbk_recovered_clk_en,
13 tx_ana_ctrl_reg_2_13	Controls txda_lpbk_isi_gen_en
12 tx_ana_ctrl_reg_2_12	Controls txda_lpbk_serial_en, when enabled (1b1), the serial loopback path from a lanes serializer to its deserializer is activated.
11 tx_ana_ctrl_reg_2_11	Controls txda_lpbk_line_en, when enabled (1b1), the line side loopback path from a lanes serial receiver to its serial transmitter is activated.
10 tx_ana_ctrl_reg_2_10	Controls txda_drv_ldo_redc_sinkiq, when enabled (1'b1) reduces the no-load current from the driver LDO in order to reduce the settling time
9 tx_ana_ctrl_reg_2_9	Controls xcvr_decap_en_del. Delayed verion (tX=100ns) of xcvr_decap_en
8 tx_ana_ctrl_reg_2_8	Controls xcvr_decap_en, which must be enabled when the common module initially powers up, or exits the suspend mode.
7 tx_ana_ctrl_reg_2_7	Controls txda_mphy_enable_hs_nt
6	Controls txda_mphy_sa_mode, which enables the driver LDO small amplitude mode.

Table continues on the next page...

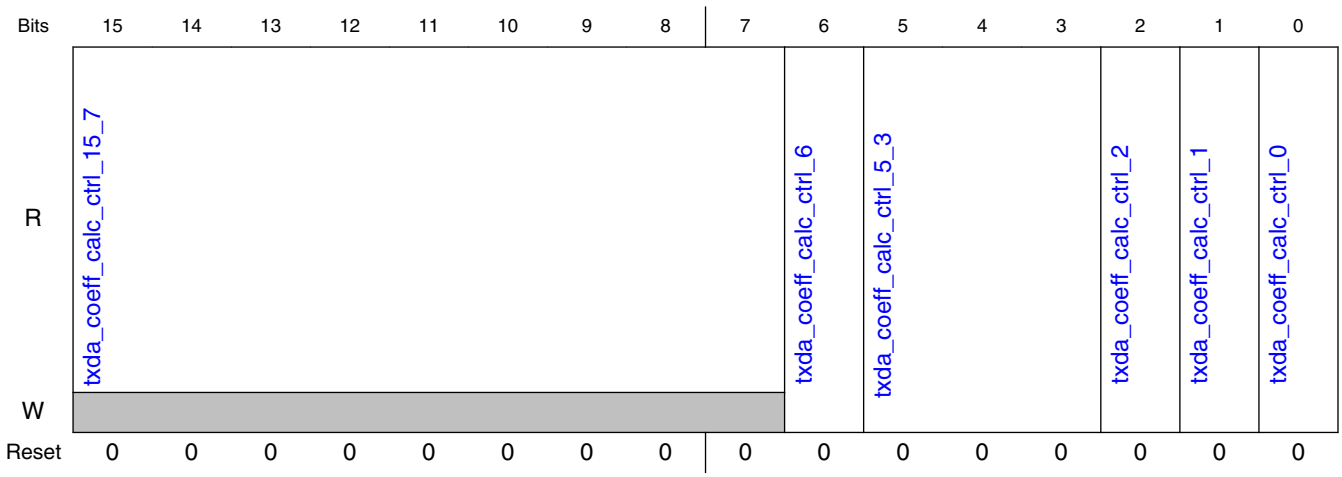
Field	Function
tx_ana_ctrl_reg_2_6	
5 tx_ana_ctrl_reg_2_5	Controls txda_drv_ldo_rbyr_fb_en, TX driver LDO VDD dependent feedback reference enable: Enables VDD dependent feedback reference to LDO.
4 tx_ana_ctrl_reg_2_4	Controls txda_drv_rst_pull_down
3 tx_ana_ctrl_reg_2_3	Controls txda_drv_ldo_bg_fb_en,
2 tx_ana_ctrl_reg_2_2	Controls txda_drv_ldo_bg_ref_en,
1 tx_ana_ctrl_reg_2_1	Controls txda_drv_predrv_en_del, delayed version (tX=100ns) os txda_predrv_en
0 tx_ana_ctrl_reg_2_0	Controls txda_drv_predrv_en,

13.4.10.2.559 Tx_coef_calc module Inputs (txda_coeff_calc_ctrl)

13.4.10.2.559.1 Offset

Register	Offset
txda_coeff_calc_ctrl	8_5022h

13.4.10.2.559.2 Diagram



13.4.10.2.559.3 Fields

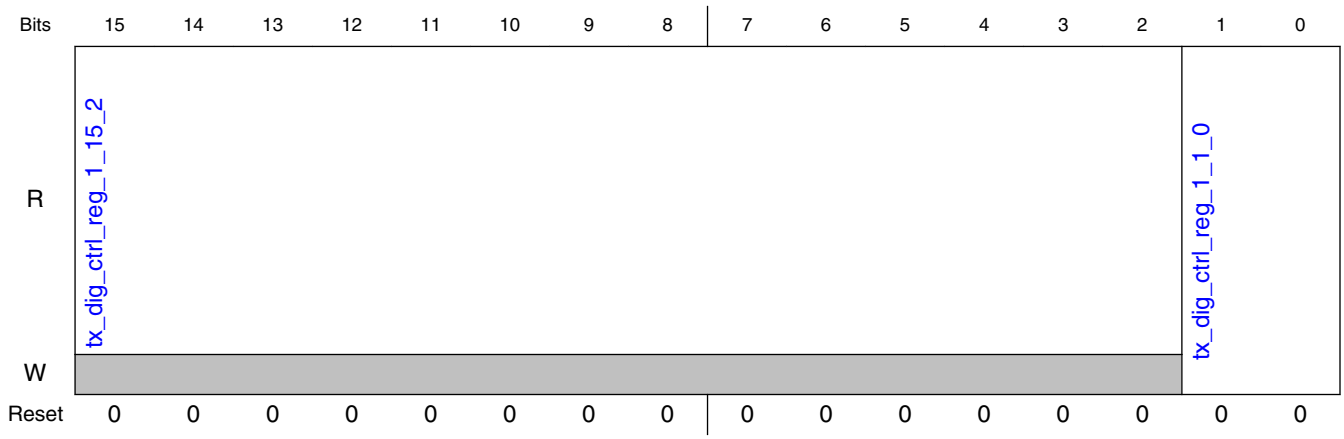
Field	Function
15-7 txda_coeff_calc_ctrl_15_7	Reserved
6 txda_coeff_calc_ctrl_6	Controls tx_high_z, when enabled (1'b1), puts the analog driver in a high impedance state.
5-3 txda_coeff_calc_ctrl_5_3	Controls, tx_vmargin, the voltage margining of the transmitter for the lane, as specified below.
2 txda_coeff_calc_ctrl_2	Controls low_power_swing_en, the voltage swing of the driver. The values below are peak to peak (differential) values.
1 txda_coeff_calc_ctrl_1	Controls tx_fcm_drv_main_en, which enable the driver with no emphasis
0 txda_coeff_calc_ctrl_0	Controls tx_fcm_full_margin, which enables the maximum margining possible

13.4.10.2.560 Tx dig control reg 1 (tx_dig_ctrl_reg_1)

13.4.10.2.560.1 Offset

Register	Offset
tx_dig_ctrl_reg_1	8_5023h

13.4.10.2.560.2 Diagram



13.4.10.2.560.3 Fields

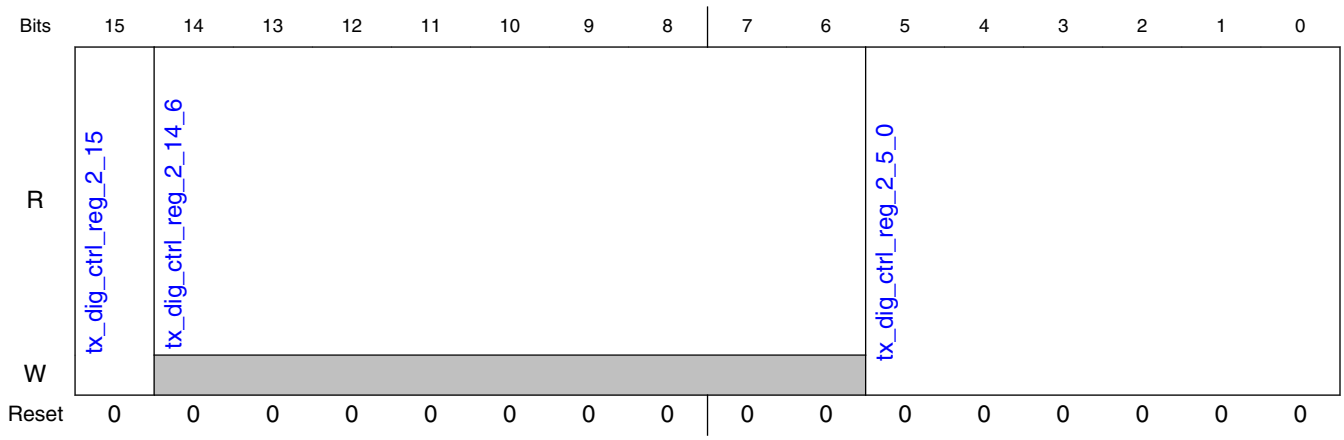
Field	Function
15-2 tx_dig_ctrl_reg_1_15_2	Reserved
1-0 tx_dig_ctrl_reg_1_1_0	Controls tx_deemphasis[1:0], the de-emphasis level of the transmitter as specified below.

13.4.10.2.561 Tx dig control reg 2 (tx_dig_ctrl_reg_2)

13.4.10.2.561.1 Offset

Register	Offset
tx_dig_ctrl_reg_2	8_5024h

13.4.10.2.561.2 Diagram



13.4.10.2.561.3 Fields

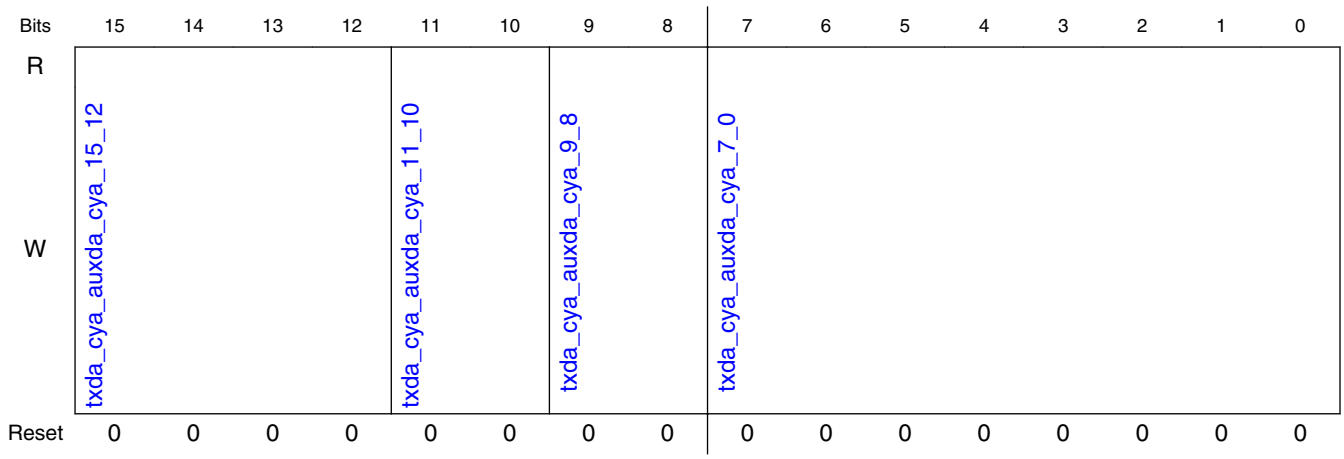
Field	Function
15 tx_dig_ctrl_reg_2_15	Controls tx_high_z_tm_en,
14-6 tx_dig_ctrl_reg_2_14_6	Reserved
5-0 tx_dig_ctrl_reg_2_5_0	tx_rescal_code required for hbdc and tx_coef_calc, average of the pull up and pull down resistor calibration values.

13.4.10.2.562 DP Aux analog control 3 (txda_cya_auxda_cya)

13.4.10.2.562.1 Offset

Register	Offset
txda_cya_auxda_cya	8_5025h

13.4.10.2.562.2 Diagram



13.4.10.2.562.3 Fields

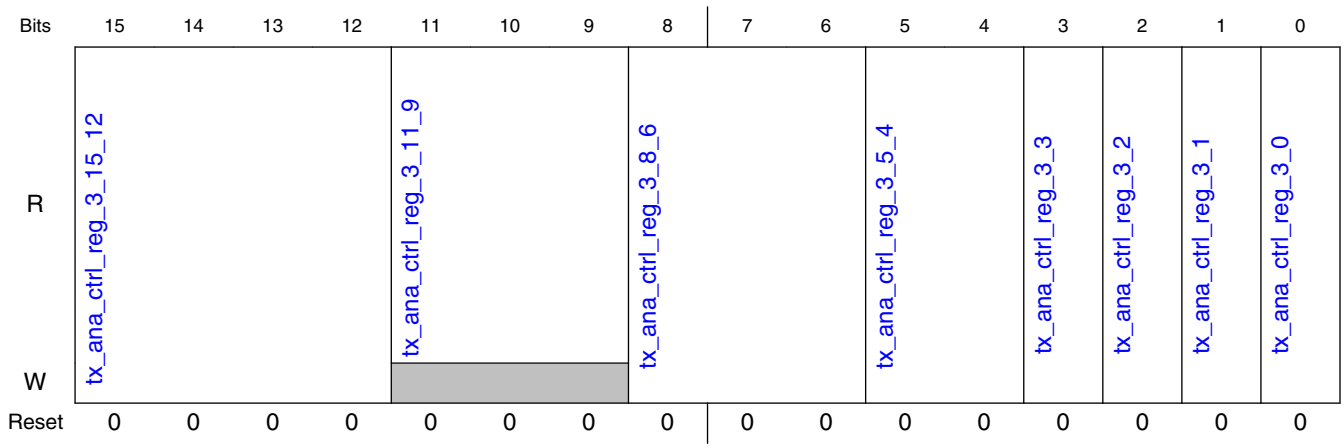
Field	Function
15-12 txda_cya_auxda_cya_15_12	txda_cya_auxda_cya_15_12
11-10 txda_cya_auxda_cya_11_10	txda_cya_auxda_cya_11_10
9-8 txda_cya_auxda_cya_9_8	txda_cya_auxda_cya_9_8
7-0 txda_cya_auxda_cya_7_0	Controls txda_cya, the TX testmode signal (cover your alternatives signals).These signals are reserved for reserved for last minute analog modifications that require CDB register control

13.4.10.2.563 DP Aux analog control 4 (tx_ana_ctrl_reg_3)

13.4.10.2.563.1 Offset

Register	Offset
tx_ana_ctrl_reg_3	8_5026h

13.4.10.2.563.2 Diagram



13.4.10.2.563.3 Fields

Field	Function
15-12 tx_ana_ctrl_reg_3_15_12	Programs txda_vcmhold_prog[3:0]
11-9 tx_ana_ctrl_reg_3_11_9	Reserved
8-6 tx_ana_ctrl_reg_3_8_6	txda_drv_boost_prog[2:0], When the programmable boost function is enabled, this signal controls the level of the boost.
5-4 tx_ana_ctrl_reg_3_5_4	Programs txda_cm_sense_vref_dac[1:0]
3 tx_ana_ctrl_reg_3_3	Controls txda_drv_mission_en,
2 tx_ana_ctrl_reg_3_2	Controls txda_drv_predrv_pullup, when the pre-driver is disabled via the txda_drv_predrv_en signal, this signal controls the state of the predriver output.
1 tx_ana_ctrl_reg_3_1	Controls txda_drv_margin_type,TX driver margin type: Selects the margining type the driver will operate in
0 tx_ana_ctrl_reg_3_0	Controls txda_cm_sense_en, Common mode sense enable, which enables the common mode sense analog circuits. This signal must be active during the entire fast establishment of common mode process.

13.4.10.2.564 DP Aux analog control 5 (tx_ana_ctrl_reg_4)

13.4.10.2.564.1 Offset

Register	Offset
tx_ana_ctrl_reg_4	8_5027h

13.4.10.2.564.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	tx_ana_ctrl_reg_4_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.564.3 Fields

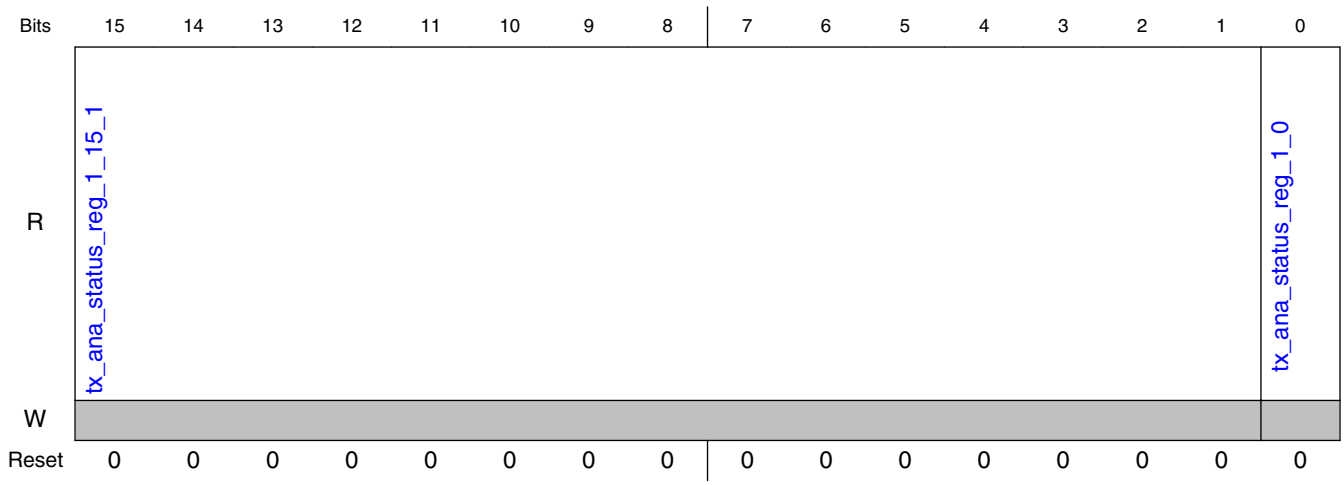
Field	Function
15-0 tx_ana_ctrl_reg_4_15_0	Programs txda_drv_ldo_prog[15:0], Sets driver LDO voltage

13.4.10.2.565 DP Aux analog status 1 (tx_ana_status_reg_1)

13.4.10.2.565.1 Offset

Register	Offset
tx_ana_status_reg_1	8_5028h

13.4.10.2.565.2 Diagram



13.4.10.2.565.3 Fields

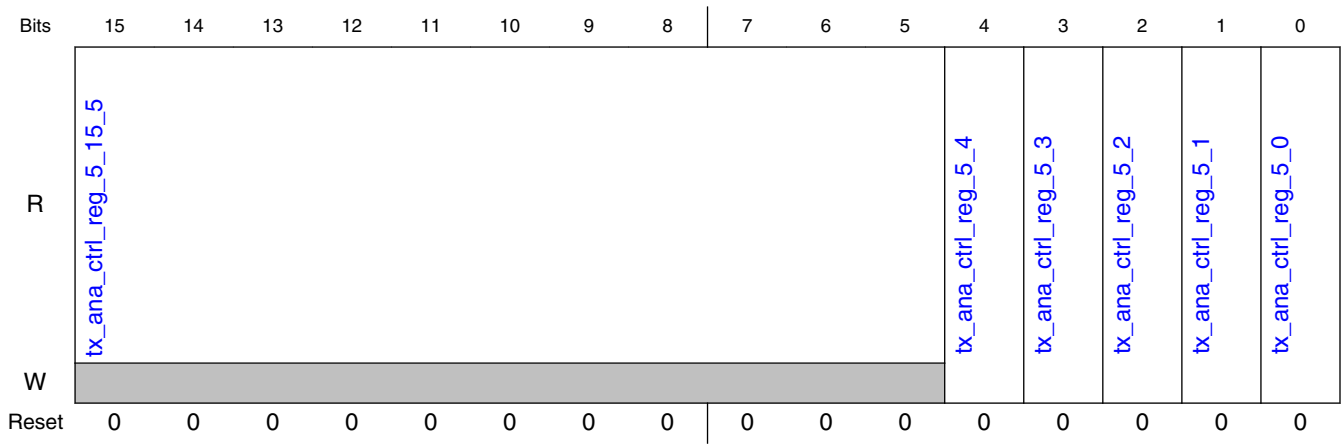
Field	Function
15-1 tx_ana_status_reg_1_15_1	Reserved
0 tx_ana_status_reg_1_0	Gives out txda_cm_sense_out, Common mode sense comparator output: Indicates the level of the tx_{p,m} signals relative to the reference voltage, controlled by the txda_cm_sense_vref_dac signal.

13.4.10.2.566 DP Aux analog status 6 (tx_ana_ctrl_reg_5)

13.4.10.2.566.1 Offset

Register	Offset
tx_ana_ctrl_reg_5	8_5029h

13.4.10.2.566.2 Diagram



13.4.10.2.566.3 Fields

Field	Function
15-5 tx_ana_ctrl_reg_5_15_5	Reserved
4 tx_ana_ctrl_reg_5_4	Programs auxda_arc_en, HDMI_ARC mode transmission/reception
3 tx_ana_ctrl_reg_5_3	Programs auxda_mhl_en, MHL/eCBUS mode transmission/reception
2 tx_ana_ctrl_reg_5_2	Programs txda_bidi_term_en
1 tx_ana_ctrl_reg_5_1	Controls txda_drv_boost_prog_en, which enables the programmable boost function
0 tx_ana_ctrl_reg_5_0	Controls txda_drv_ldo_vdd_ref_en,

13.4.10.2.567 rx_aux analog control1 (rx_ana_ctrl_reg_1)

13.4.10.2.567.1 Offset

Register	Offset
rx_ana_ctrl_reg_1	8_9060h

13.4.10.2.567.2 Diagram

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	rx_ana_ctrl_reg_1_15_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.4.10.2.567.3 Fields

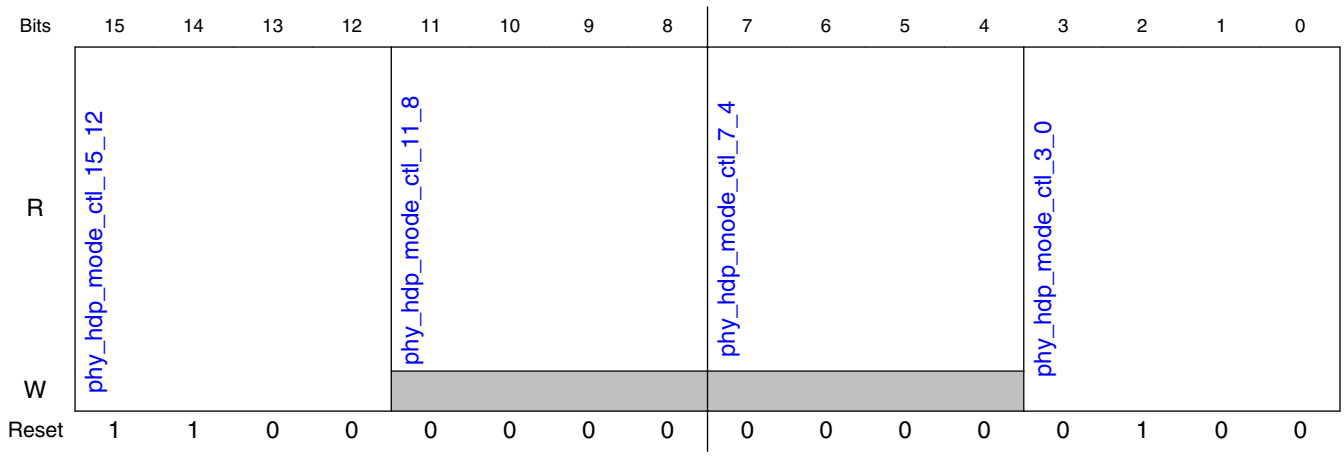
Field	Function
15-0 rx_ana_ctrl_reg_1_15_0	Unused

13.4.10.2.568 HDP Mode Control register (phy_hdp_mode_ctl)

13.4.10.2.568.1 Offset

Register	Offset
phy_hdp_mode_ctl	8_C008h

13.4.10.2.568.2 Diagram



13.4.10.2.568.3 Fields

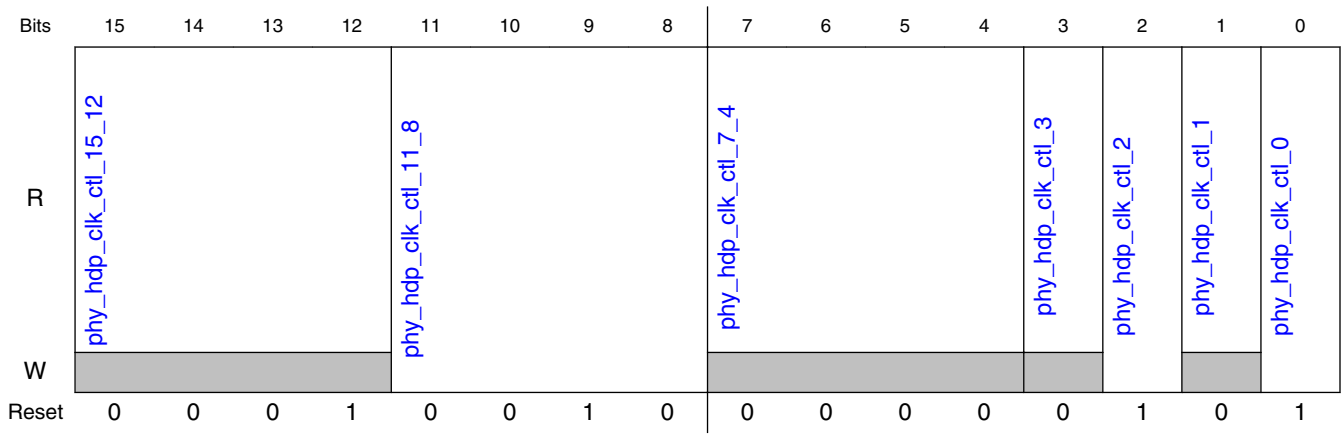
Field	Function
15-12 phy_hdp_mode_ctl_15_12	PHY HDP lane disable - 0 = enable associated PHY HDP lane; 1 = disable/powerdown the associated PHY DP lane. This field is used to disable PHY lanes when not used. For example, DP can operate on a sub-set of the lanes (1 or 2) instead of all 4. HDMI requires all 4 lanes at all times. The PMA lane(s) mapped to the disabled HDP lane(s) will be powered down as well. Value can only be changed when phy_reset_n is asserted low or as described by the DP lane configuration change procedure.
11-8 phy_hdp_mode_ctl_11_8	Reserved
7-4 phy_hdp_mode_ctl_7_4	HDP Power State Acknowledgement - power state acknowledgement for HDP lanes. (Re-synchronized to APB clock.)
3-0 phy_hdp_mode_ctl_3_0	HDP Power State - power state for HDP lanes. Direct mapping to the PMAs A0 to A3 power states (A0 = 0b0001, A1 = 0b0010, A2 = 0b0100 and A3 = 0b1000). (Re-synchronized to PSM clock.)

13.4.10.2.569 HDP Clock Control register (phy_hdp_clk_ctl)

13.4.10.2.569.1 Offset

Register	Offset
phy_hdp_clk_ctl	8_C009h

13.4.10.2.569.2 Diagram



13.4.10.2.569.3 Fields

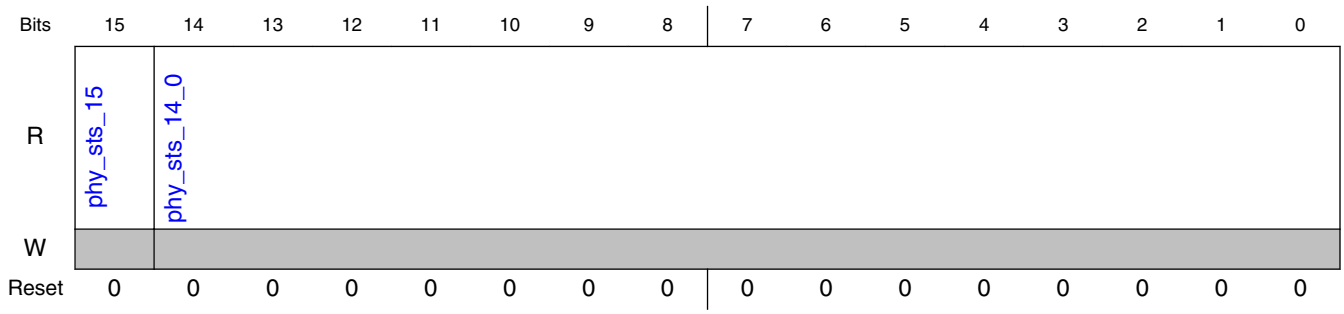
Field	Function
15-12 phy_hdp_clk_ctl_15_12	DP PLL data rate 1 clock divider value. Divider value for the PLL clock to generate phy_pma_char_clk_out. (HBR2 = 1, RBR/HBR = 2.)
11-8 phy_hdp_clk_ctl_11_8	DP PLL data rate 0 clock divider value. Divider value for the PLL clock to generate phy_pma_tx_data_clk_out. (HBR2 = 2, RBR/HBR = 4.)
7-4 phy_hdp_clk_ctl_7_4	Reserved
3 phy_hdp_clk_ctl_3	HDP PLL clock enable acknowledge - Indicates whether HDP PLLs data rate and full rate clocks are active/enabled. 1 = clocks enabled/active, 0 = clocks disabled/gated.
2 phy_hdp_clk_ctl_2	HDP PLL clock enable - Clock enable for HDP PLLs data rate and full rate clocks out of PMA. 1 = enable PLL data rate and full rate clocks, 0 = gate PLL data rate and full rate clocks
1 phy_hdp_clk_ctl_1	HDP PLL ready - HDP PLLs ready indication for high speed clocks. 1 = PLL ready, 0 = PLL not ready.
0 phy_hdp_clk_ctl_0	HDP PLL enable - HDP PLLs enable for high speed clocks. 1 = enabled, 0 = disabled.

13.4.10.2.570 PHY status register (phy_sts)

13.4.10.2.570.1 Offset

Register	Offset
phy_sts	8_C00Fh

13.4.10.2.570.2 Diagram



13.4.10.2.570.3 Fields

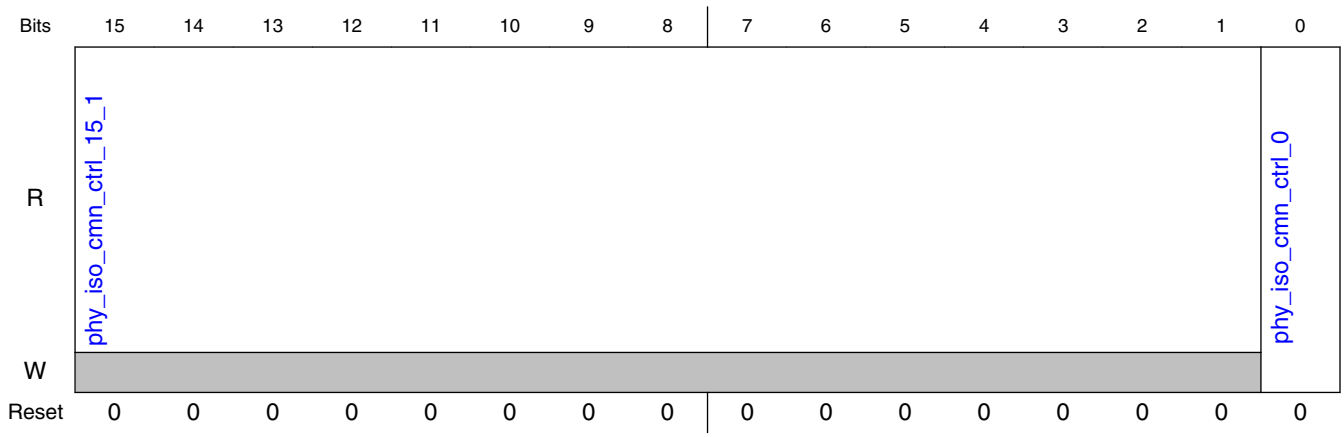
Field	Function
15 phy_sts_15	PHY APB access timeout: When set, an APB read/write request to PHY level registers failed (i.e. timed out). When set, this bit is cleared upon read.
14-0 phy_sts_14_0	Reserved

13.4.10.2.571 PHY common control signal isolation register (phy_iso_cmn_ctrl)

13.4.10.2.571.1 Offset

Register	Offset
phy_iso_cmn_ctrl	8_C010h

13.4.10.2.571.2 Diagram



13.4.10.2.571.3 Fields

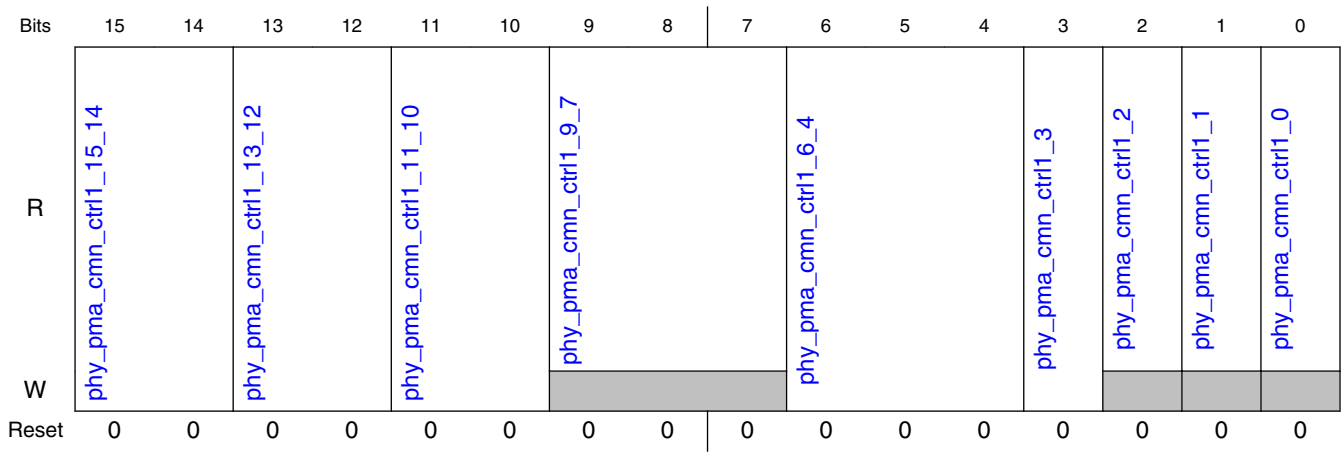
Field	Function
15-1 phy_iso_cmn_ctrl_15_1	Reserved
0 phy_iso_cmn_ctrl_0	Drives phy_reset_n PHY input when in PHY and PMA isolation modes.

13.4.10.2.572 PMA common control1 register (phy_pma_cmn_ctrl1)

13.4.10.2.572.1 Offset

Register	Offset
phy_pma_cmn_ctrl1	8_C800h

13.4.10.2.572.2 Diagram



13.4.10.2.572.3 Fields

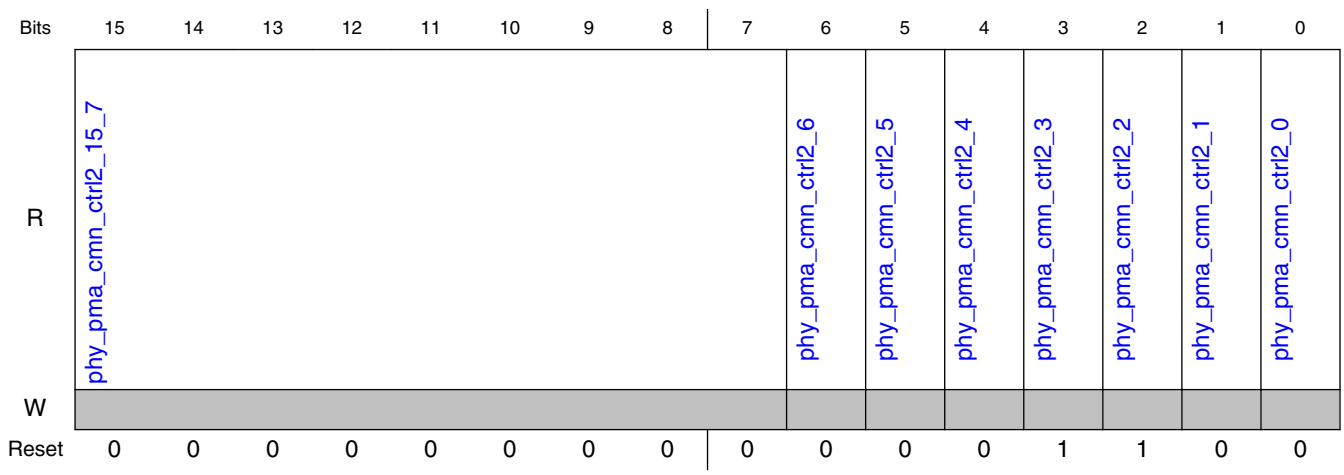
Field	Function
15-14 phy_pma_cmn_ctrl1_15_14	Drives cmn_ref_clk_ana_div PMA input
13-12 phy_pma_cmn_ctrl1_13_12	Drives cmn_ref_clk_dig_div PMA input
11-10 phy_pma_cmn_ctrl1_11_10	Drives cmn_psm_clk_dig_div PMA input
9-7 phy_pma_cmn_ctrl1_9_7	Reserved
6-4 phy_pma_cmn_ctrl1_6_4	Drives cmn_ref_clk_sel PMA input
3 phy_pma_cmn_ctrl1_3	Drives cmn_ref_clk_rcv_en PMA input
2 phy_pma_cmn_ctrl1_2	Current value of cmn_macro_suspend_ack PMA output
1 phy_pma_cmn_ctrl1_1	Current value of cmn_refclk_active PMA output
0 phy_pma_cmn_ctrl1_0	Current value of cmn_ready PMA output

13.4.10.2.573 PMA common control2 register (phy_pma_cmn_ctrl2)

13.4.10.2.573.1 Offset

Register	Offset
phy_pma_cmn_ctrl2	8_C801h

13.4.10.2.573.2 Diagram



13.4.10.2.573.3 Fields

Field	Function
15-7 phy_pma_cmn_ctrl2_15_7	Reserved
6 phy_pma_cmn_ctrl2_6	Current value of cmn_pll0_locked PMA output
5 phy_pma_cmn_ctrl2_5	Reserved
4 phy_pma_cmn_ctrl2_4	Current value of cmn_pll0_clk_datart_en_ack PMA output
3	Reserved

Table continues on the next page...

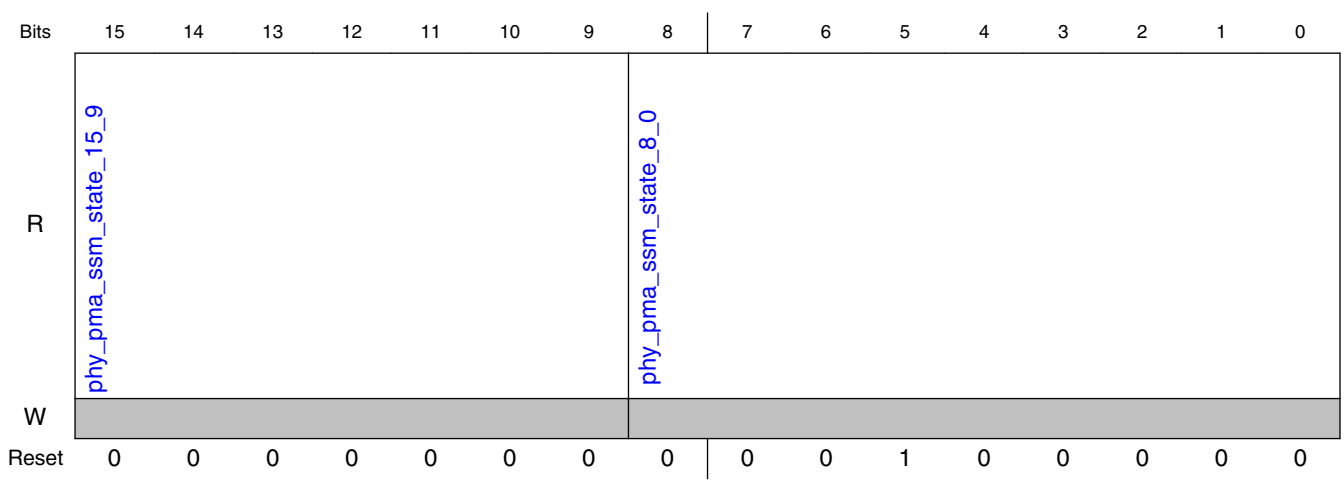
Field	Function
phy_pma_cmn_ctrl2_3	
2 phy_pma_cmn_ctrl2_2	Current value of cmn_pll0_disabled PMA output
1 phy_pma_cmn_ctrl2_1	Reserved
0 phy_pma_cmn_ctrl2_0	Current value of cmn_pll0_ready PMA output

13.4.10.2.574 PMA SSM current state register (phy_pma_ssm_state)

13.4.10.2.574.1 Offset

Register	Offset
phy_pma_ssm_state	8_C802h

13.4.10.2.574.2 Diagram



13.4.10.2.574.3 Fields

Field	Function
15-9	Reserved

Table continues on the next page...

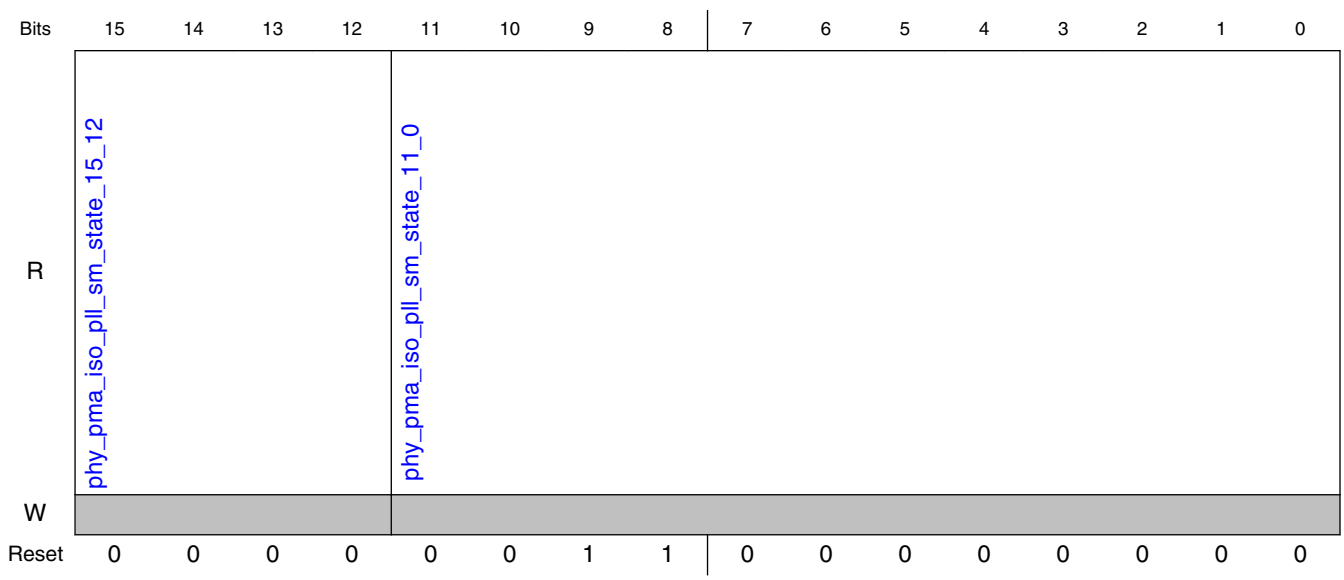
Field	Function
phy_pma_ssm_state_15_9	
8-0 phy_pma_ssm_state_8_0	PMA SSM : Current state of the PMA startup state machine. PMA output (Not re-synchronized to apb_pclk)

13.4.10.2.575 PMA PLL State Machine current state register (phy_pma_iso_pll_sm_state)

13.4.10.2.575.1 Offset

Register	Offset
phy_pma_iso_pll_sm_state	8_C803h

13.4.10.2.575.2 Diagram



13.4.10.2.575.3 Fields

Field	Function
15-12	Reserved

Table continues on the next page...

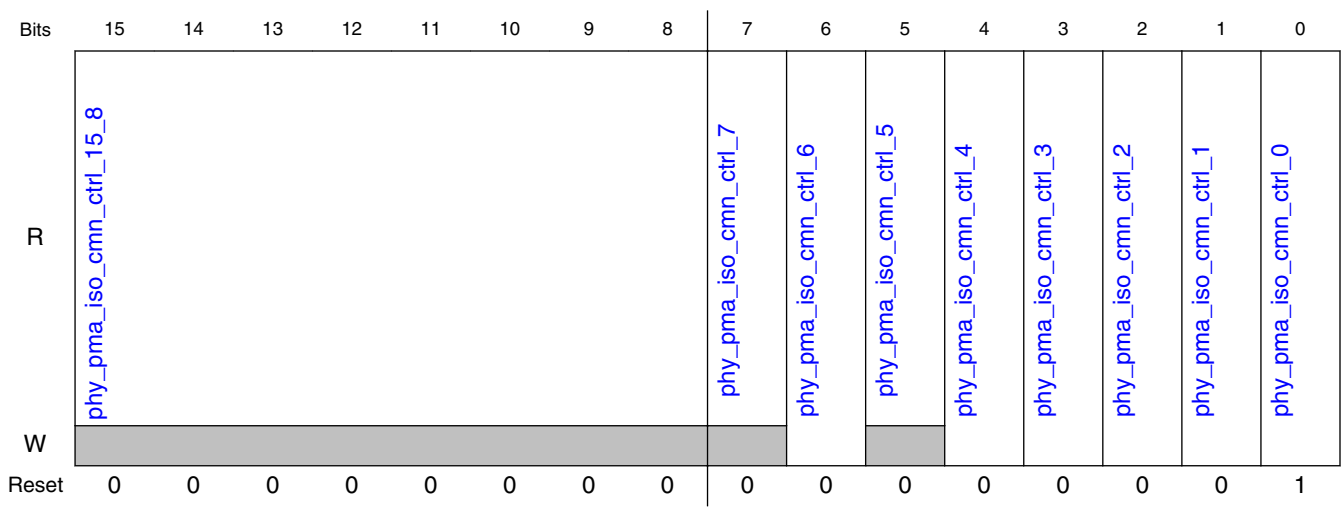
Field	Function
phy_pma_iso_pll_sm_state_15_12	
11-0 phy_pma_iso_pll_sm_state_11_0	Current value of cmn_pllsm0_state[11:0]. PMA output (Debug only: Not re-synchronized)

13.4.10.2.576 PMA common control signal isolation register (phy_pma_iso_cmn_ctrl)

13.4.10.2.576.1 Offset

Register	Offset
phy_pma_iso_cmn_ctrl	8_C810h

13.4.10.2.576.2 Diagram



13.4.10.2.576.3 Fields

Field	Function
15-8 phy_pma_iso_cmn_ctrl_15_8	Reserved

Table continues on the next page...

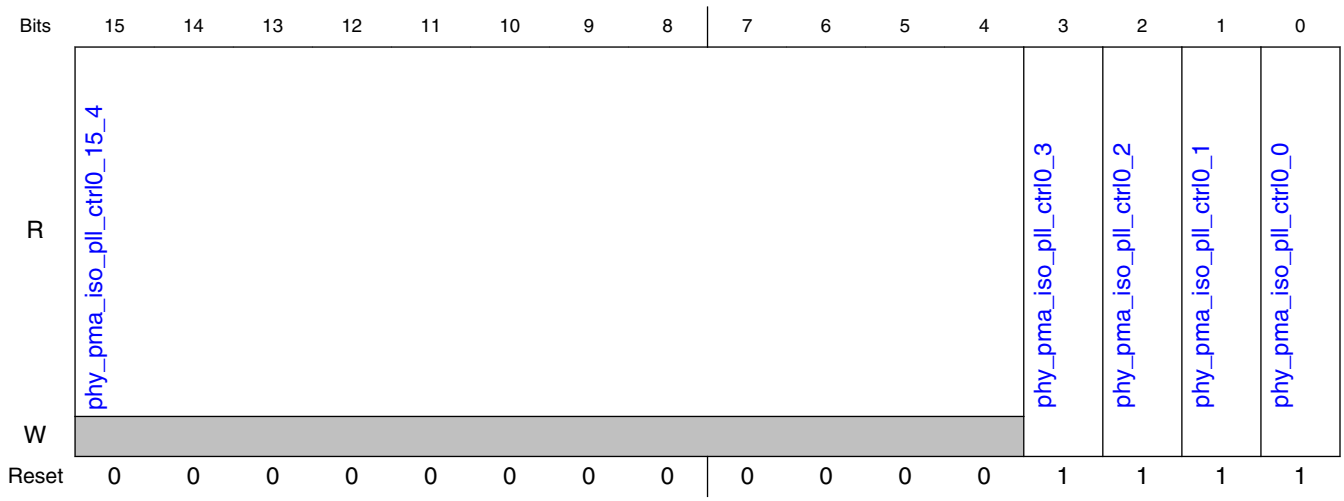
Field	Function
7 phy_pma_iso_cmn_ctrl_7	Current value of cmn_clock_stop_ack PMA output.
6 phy_pma_iso_cmn_ctrl_6	Drives cmn_clock_stop_req PMA input when in PMA isolation mode
5 phy_pma_iso_cmn_ctrl_5	Current value of cmn_macro_pwr_en_ack PMA output.
4 phy_pma_iso_cmn_ctrl_4	Drives cmn_macro_pwr_en PMA input when in PHY macro and PMA isolation modes
3 phy_pma_iso_cmn_ctrl_3	Drives cmn_refclk_disable PMA input when in PMA isolation mode.
2 phy_pma_iso_cmn_ctrl_2	Drives macro_suspend_req PMA input when in PMA isolation mode.
1 phy_pma_iso_cmn_ctrl_1	Drives cmn_macro_en PMA input when in PMA isolation mode.
0 phy_pma_iso_cmn_ctrl_0	Drives cmn_reset_n PMA input when in PMA isolation mode.

13.4.10.2.577 PMA PLL control0 isolation register (phy_pma_iso_pll_ctrl0)

13.4.10.2.577.1 Offset

Register	Offset
phy_pma_iso_pll_ctrl0	8_C811h

13.4.10.2.577.2 Diagram



13.4.10.2.577.3 Fields

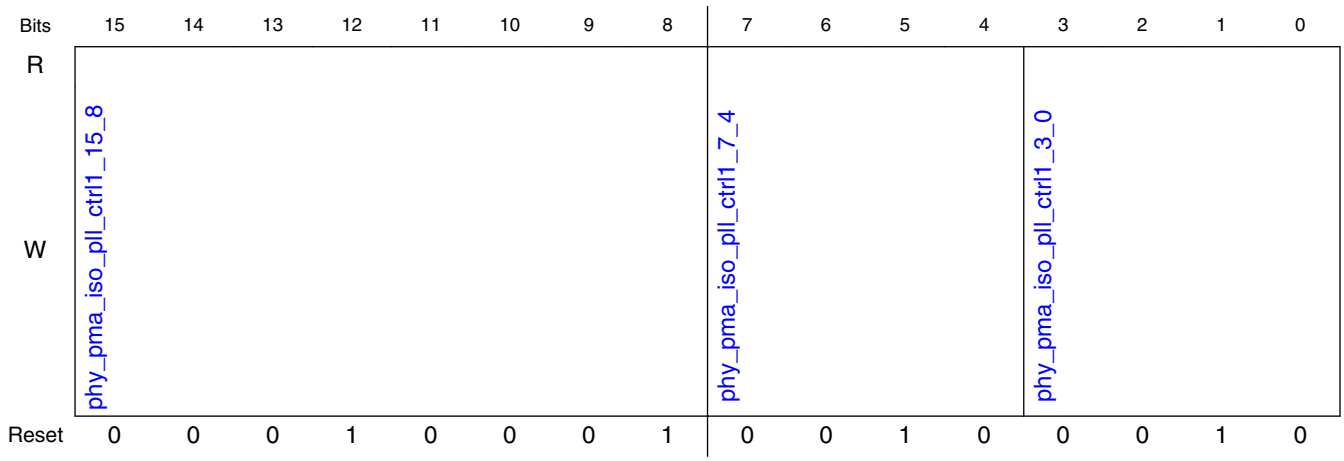
Field	Function
15-4 phy_pma_iso_pll_ctrl0_15_4	Reserved
3 phy_pma_iso_pll_ctrl0_3	Reserved
2 phy_pma_iso_pll_ctrl0_2	Drives cmn_pll0_clk_datart_en PMA input when in PMA isolation mode
1 phy_pma_iso_pll_ctrl0_1	Reserved
0 phy_pma_iso_pll_ctrl0_0	Drives cmn_pll0_en PMA input when in PMA isolation mode

13.4.10.2.578 PMA PLL control1 isolation register (phy_pma_iso_pll_ctrl1)

13.4.10.2.578.1 Offset

Register	Offset
phy_pma_iso_pll_ctrl1	8_C812h

13.4.10.2.578.2 Diagram



13.4.10.2.578.3 Fields

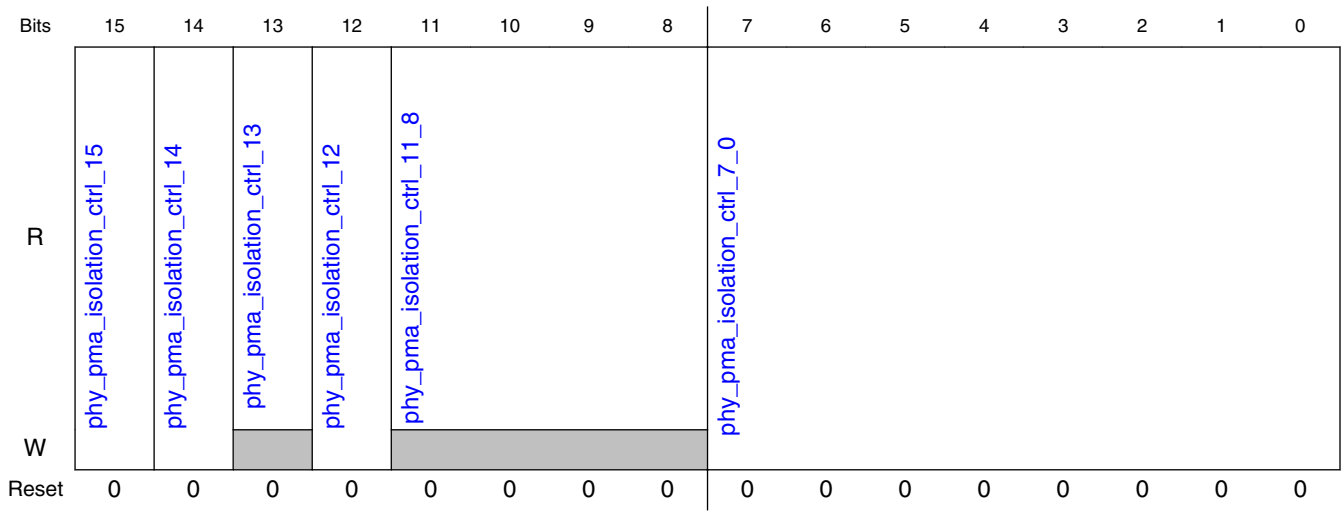
Field	Function
15-8 phy_pma_iso_pll_ctrl1_15_8	Reserved
7-4 phy_pma_iso_pll_ctrl1_7_4	Drives cmn_pll0_clk_datart1_div PMA input when in PMA isolation mode
3-0 phy_pma_iso_pll_ctrl1_3_0	Drives cmn_pll0_clk_datart0_div PMA input when in PMA isolation mode

13.4.10.2.579 Isolation control register (phy_pma_isolation_ctrl)

13.4.10.2.579.1 Offset

Register	Offset
phy_pma_isolation_ctrl	8_C81Fh

13.4.10.2.579.2 Diagram



13.4.10.2.579.3 Fields

Field	Function
15 phy_pma_isolation_ctrl_15	PHY/PMA isolation enable (isolation_en) - When set, enables isolation (PHY or PMA).
14 phy_pma_isolation_ctrl_14	PHY/PMA common isolation enable (cmn_isolation_en) - When in PHY Macro Isolation Mode, the PHY common isolation register(s) are selected. When in PMA Isolation Mode, the PMA common isolation register(s) are selected.
13 phy_pma_isolation_ctrl_13	Reserved
12 phy_pma_isolation_ctrl_12	PHY/PMA isolation mode select (isolation_mode_sel) - When isolation_en is set, this bit selects between PHY isolation and PMA isolation mode. 0 = PHY isolation mode; 1 = PMA isolation mode.
11-8 phy_pma_isolation_ctrl_11_8	Reserved
7-0 phy_pma_isolation_ctrl_7_0	PHY/PMA lane isolation enable (ln_isolation_en) - When in PHY Macro Isolation Mode, the selected PHY lane(s) isolation registers are selected. When in PMA Isolation Mode, the selected PMA lane(s) isolation registers are selected.

13.5 HDMI / DisplayPort Transmit PHY (HDMI PHY)

13.5.1 Introduction

13.5.1.1 Overview

The HDMI and DisplayPort Transmit PHY consisting of a multi-protocol PMA, AUX/ ARC channel, and supporting components (APB interface, TAP controller, PHY and PMA isolation and control registers and clock and reset logic). In regards to the top level PHY specification, it is assumed that the reader of this document is familiar with these specifications.

Figure 13-33 illustrates a top-level block diagram for the PHY macro, and how it interfaces with an external MAC. The following are high level points worth noting in this fig- ure.

- The PMA isolation and control logic
  - It contains registers for controlling the PHY/PMA's power state and mode of operation.
- The Async APB Interface Slave module is part of a two module system that allows the internal CDB registers to be written and read by either the external APB interface or the integrated TAP interface. Both interfaces can access the internal CDB, with the TAP having higher priority.

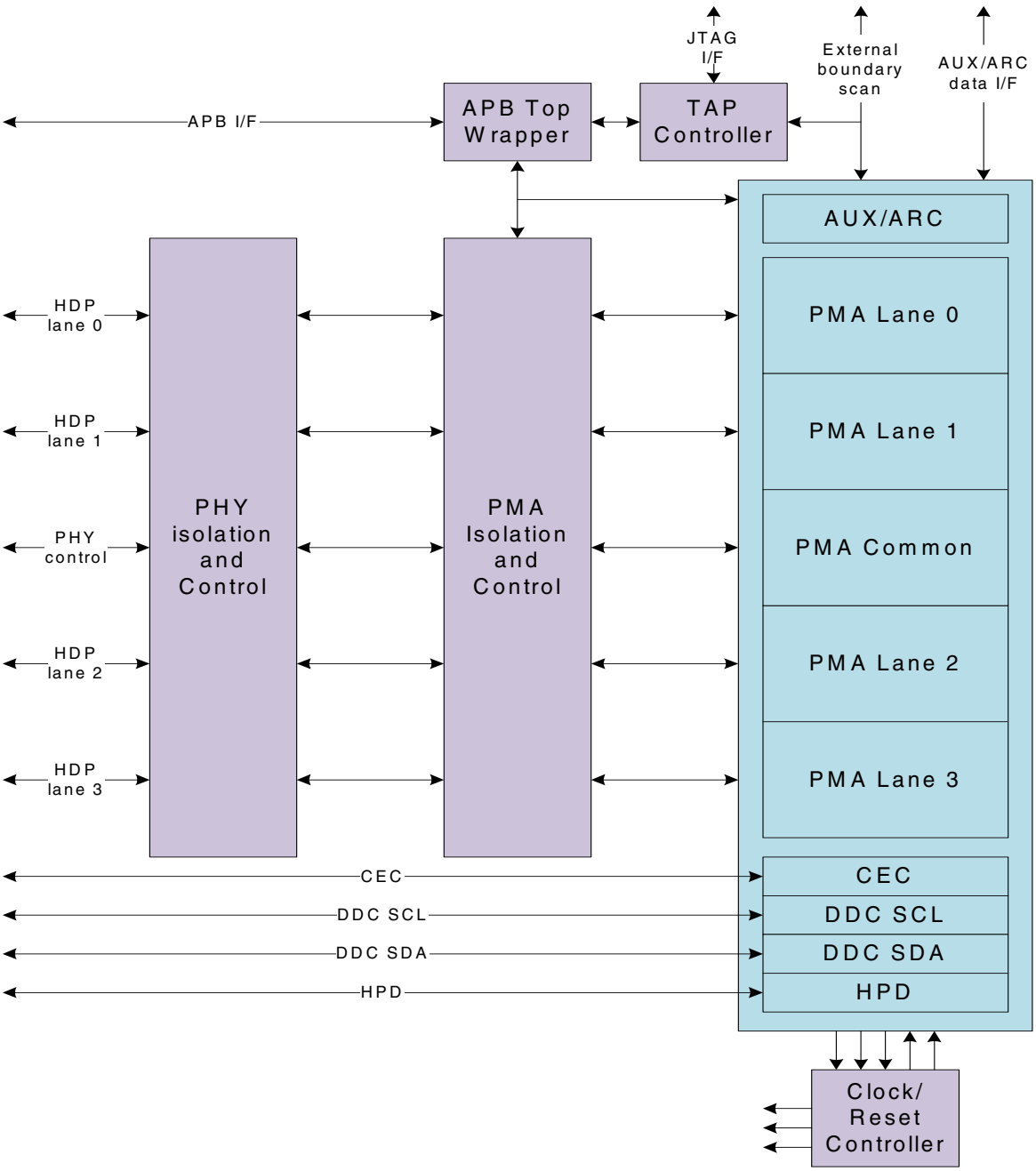


Figure 13-33. HDP PHY block diagram

13.5.1.2 Features

The features of the HDMI and DisplayPort Transmit PHY are listed below:

- Supported standards:
  - HDMI v2.0/1.4
  - DisplayPort 1.3 (RDR, HDR and HDR2 rates)
  - Embedded DisplayPort v1.4

NOTE

DP 1.3 HBR3, 8.1 Gbps, data rate not supported.

- 4 PMA TX lanes
- Supports DisplayPort AUX channel and HDMI ARC channel
- PMA and AUX/ARC
- 20-bit data interface width for both DP and HDMI
- Includes HPD, DDC SDA and SCL and CEC pads

13.5.2 External Signals

The external signals are listed in the table below:

Table 13-19. External signals

Signal Name	Direction	Type	Description
refclk_p refclk_m	input	Analog	PMA external reference clock. Only valid for DP. The HDMI pixel clock cannot be driven on these pins. For DP, supports nominal frequencies of 19.2, 20, 24, 27, 54 and 108 MHz. The following external reference clock sources are supported: • AC coupled differential low swing clock (HCSL levels) • DC single ended clock on refclk_p pin. In this mode refclk_m should be tied to ground. This mode is for test purposes only. For DP, a reference clock must be provided on either these external pins or on the refclock internal SoC-side pin.
tx_p_ln_{3:0} tx_m_ln_{3:0}	output	Diff	PMA transmitter serial data
rext	input	Analog	PMA external calibration resistor
aux_p aux_m	inout	Analog	AUX differential Tx/Rx serial data. For ARC, aux_p pin is used as input only.
hpd	inout	Analog	HDMI Hot Plug Detect
ddc_sda	inout	Analog	HDMI DDC serial data
ddc_scl	inout	Analog	HDMI DDC serial clock
cec	inout	Analog	HDMI CEC

13.5.3 On-Chip Functional Pin List

On-chip functional signals are listed in the table below:

Table 13-20. On-chip functional pin list

Signal Name	Direction	Type	Timing	Description
External APB Interface				
apb_pclk	input	CMOS	n/a	APB bus clock.
apb_preset_n	input	CMOS	async	APB bus reset.
apb_penable	input	CMOS	apb_pclk	APB bus enable.
apb_psel	input	CMOS	apb_pclk	APB bus device select.
apb_pwrite	input	CMOS	apb_pclk	APB bus write.
apb_paddr[15:0]	input	CMOS	apb_pclk	APB bus address.
apb_pwdata[15:0]	input	CMOS	apb_pclk	APB bus write data.
apb_prdata[15:0]	output	CMOS	apb_pclk	APB bus read data.
apb_pready	output	CMOS	apb_pclk	APB bus ready.
HDP Interface				
phy_reset_n	input	CMOS	async	**PHY reset** : Asserting this signal low will reset the PHY, with the exception of the APB registers.
phy_pma_tx_data_clk_in	input	CMOS	n/a	**Tx Data clock**. Transmit data clock. The Tx data is synchronous to this clock.
phy_pma_tx_data_clk_out	output	CMOS	n/a	**PHY data rate clock out** - 20-bit data rate clock out from PHY which provides the clock source for phy_pma_tx_data_clk_in, so that the PHY transmit inter- face is source synchronous.
phy_pma_tx_data_ln_{3:0}[19:0]	input	CMOS	phy_pma_tx_data_clk_in	**Transmit data**. HDMI/ Display Tx display data. Synchro- nous to phy_pma_tx_data_clk_in.
phy_pma_char_clk_out	output	CMOS	n/a	**PHY character clock out** - 10-bit free running clock from PHY. Runs at 2x frequency of phy_pma_tx_data_clk_out. Frequency aligned, but not edge aligned

Table continues on the next page...

Table 13-20. On-chip functional pin list (continued)

Signal Name	Direction	Type	Timing	Description
				(not synchronous) to phy_pma_tx_data_clk_out.
AUX / ARC Interface				
source_aux_arc_oen	input	CMOS	async	**AUX/ARC output enable** : 1 : Receive mode 0 : Transmit mode For ARC, the AUX/ARC block will always be in receive mode. Therefore, this pin shall be tied high for HDMI operation.
source_aux_data_in	input	CMOS	async	**AUX transmit data -** For HDMI Tx, ARC is only used as a receiver and thus there is not ARC transmit data.
source_aux_arc_data_out	output	CMOS	async	**AUX/ARC receive data**
PMA Interface				
refclock	input	CMOS	n/a	**PMA internal reference clock input -** Single-ended alternate reference clock input. For DP, supports nominal frequencies of 19.2, 20, 24, 27, 54 and 108 MHz. For HDMI, the pixel clock must be driven on this pin with a frequency range from 25 to 600 MHz. PHY_PMA_CMN_CTRL1[6:4] is used to select between refclock and refclk_<p/m> as the reference clock source. refclock must be selected as the reference clock source for HDMI.
pma_refclk_rcv_out	output	CMOS	n/a	**PMA common reference clock receiver output -** Clock output from the analog reference clock receiver circuits. Frequency is always the same as the selected reference

Table continues on the next page...

Table 13-20. On-chip functional pin list (continued)

Signal Name	Direction	Type	Timing	Description
				clock input. Once enabled, this clock is free-running and is active in all power states (provided the input reference clock is running). This clock is enabled by setting PHY_PMA_CMN_CTRL1[3]. This clock is provided for use by the SoC, if needed.
HPD Interface				
hpd_in_from_pad	output	CMOS	n/a	**HPD pad data in**
hpd_out_high_to_pad	input	CMOS	n/a	**HPD pad output drive high**
hpd_out_low_to_pad	input	CMOS	n/a	**HPD pad output drive low**
hpd_out_drv_strength[1:0]	input	CMOS	static	**HPD pad output drive strength -** Selects output drive strength for HPD pad.
hpd_out_en_to_pad	input	CMOS	n/a	**HPD pad output enable -** 1 = output mode 0 = input mode
hpd_power_down_to_pad	input	CMOS	n/a	**HPD pad power down -** 1 = power down HPD pad 0 = normal mode
phy_pma_pixel_clk_out	output			PHY Pixel clock out - Pixel clock output from PHY.
DDC SDA Interface				
ddc_sda_in_from_pad	output	CMOS	n/a	**DDC SDA pad in**
ddc_sda_out_high_to_pad	input	CMOS	n/a	**DDC SDA pad output drive high**
ddc_sda_out_low_to_pad	input	CMOS	n/a	**DDC SDA pad output drive low**
ddc_sda_out_drv_strength[1:0]	input	CMOS	n/a	**DDC SDA pad output drive strength -** Selects output drive strength for DDC SDA pad.
ddc_sda_out_en_to_pad	input	CMOS	n/a	**DDC SDA pad output enable -**

Table continues on the next page...

Table 13-20. On-chip functional pin list (continued)

Signal Name	Direction	Type	Timing	Description
				1 = output mode 0 = input mode
ddc_sda_power_down_to_pad	input	CMOS	n/a	**DDC SDA pad power down -** 1 = power down DDC SDA pad 0 = normal mode
DDC SCL Interface				
ddc_scl_in_from_pad	output	CMOS	n/a	**DDC SCL pad in**
ddc_scl_out_high_to_pad	input	CMOS	n/a	**DDC SCL pad output drive high**
ddc_scl_out_low_to_pad	input	CMOS	n/a	**DDC SCL pad output drive low**
ddc_scl_out_drv_strength[1:0]	input	CMOS	n/a	**DDC SCL pad output drive strength -** Selects output drive strength for DDC SDA pad.
ddc_scl_out_en_to_pad	input	CMOS	n/a	**DDC SCL pad output enable -** 1 = output mode 0 = input mode
ddc_scl_power_down_to_pad	input	CMOS	n/a	**DDC SCL pad power down -** 1 = power down DDC SCL pad 0 = normal mode
CEC Interface				
cec_in_from_pad	output	CMOS	n/a	**CEC pad in**
cec_out_high_to_pad	input	CMOS	n/a	**CEC pad output drive high**
cec_out_low_to_pad	input	CMOS	n/a	**CEC pad output drive low**
cec_out_drv_strength[1:0]	input	CMOS	n/a	**CEC pad output drive strength -** Selects output drive strength for DDC SDA pad.
cec_out_en_to_pad	input	CMOS	n/a	**CEC pad output enable -** 1 = output mode 0 = input mode
cec_power_down_to_pad	input	CMOS	n/a	**CEC pad power down -**

Table 13-20. On-chip functional pin list

Signal Name	Direction	Type	Timing	Description
				1 = power down CEC pad 0 = normal mode

13.5.4 PHY Clocking and Reset

This section describes the details of the PHY clocking and reset functions. Note that what is described here is the top level function.

13.5.4.1 Reference and Data Path Clocking

Figure 13-34 illustrates the clocking associated with the reference clock and data path clocks. The following points should be noted about this function.

- The selection of the internal (refclock) or external (refclk_p / refclk_m) clock as the source for the PMA clock, and ulti- mately generates cmn_pllclk_datart0 PMA clock output is controlled by a PMA APB register.
- phy_pma_tx_data_clk_in must be driven from phy_pma_tx_data_clk_out. The DP/HDMI transmit data for all lanes (phy_pma_tx_data_ln_<>[19:0]) is synchronous to phy_pma_tx_data_clk_in.

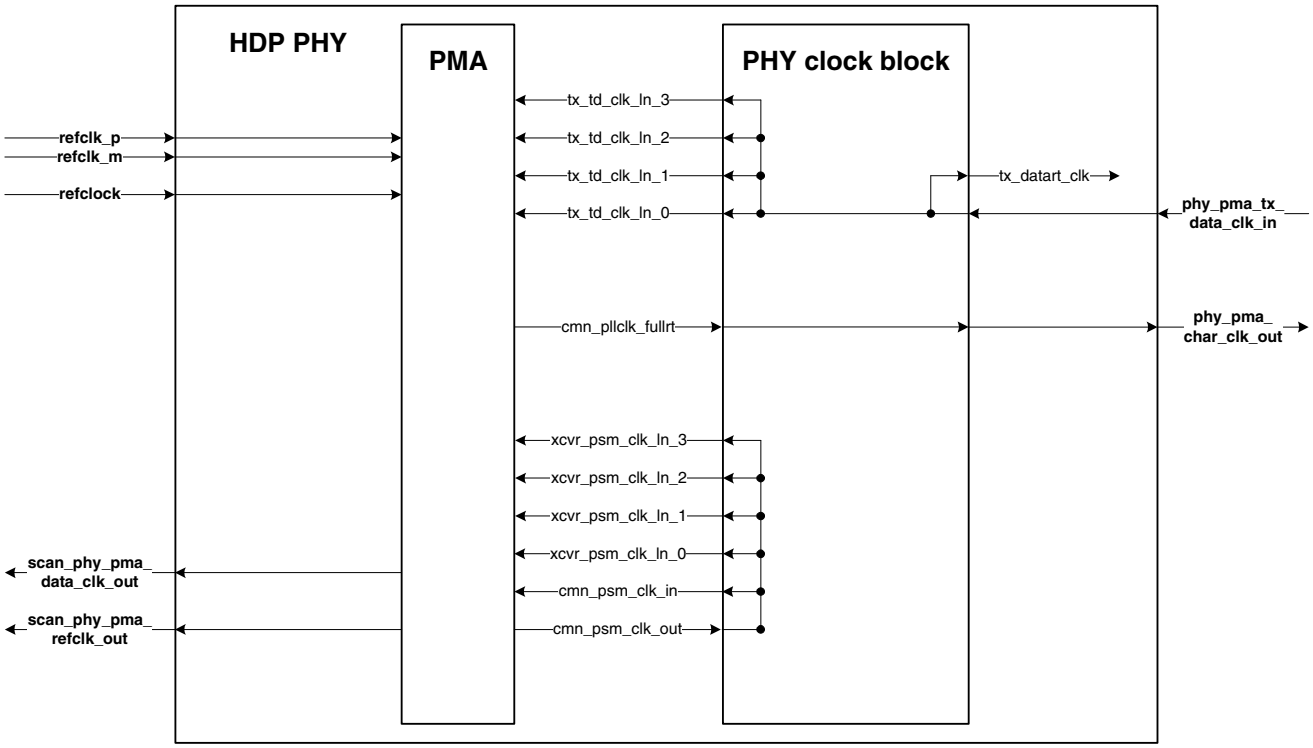


Figure 13-34. PHY clocking block diagram

13.5.4.2 Reset Distribution

All resets are active low, asynchronously activated, and synchronously deactivated (synchronized internally).

phy_reset_n is master reset for entire PHY. Asserting phy_reset_n will drive reset to entire PHY, except for APB and TAP controller registers.

When phy_reset_n is released, a reset sequencer controls the release of the various resets for the entire macro, as follows:

- After phy_reset_n is released, cmn_reset_n going to the PMA is released.
- The PHY reset sequencer waits for cmn_ready from the PMA to go active, indicating common modules are powered up, calibrated, and the PLL has locked.
- At this point, all other resets going to logic in the PHY are synchronously released (1 for each internal PHY clock domain).

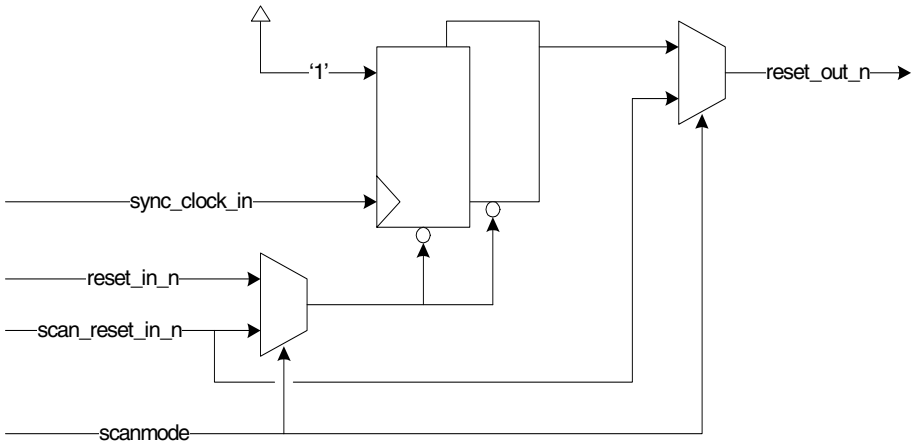


Figure 13-35. Reset synchronizer

13.5.4.3 PHY Clocking

The following table shows the clock names and their associated frequencies in PHY.

Table 13-21. Clock and frequency details

Clock Name	Maximum Frequency (MHz)
refclk_p	600
refclk_m	600
apb_pclk	200
phy_pma_tx_data_clk_in	300
phy_pma_tx_data_clk_out	300
phy_pma_char_clk_out	600
refclock	600
pma_refclk_rcv_out	600
tap_tck	100
scan_phy_pma_refclk_in	200
scan_phy_pma_data_clk_in	300
scan_phy_pma_refclk_out	600
scan_phy_pma_data_clk_out	300
tx_bscan_ext_clockdr	100
tx_bscan_ext_updatedr	100
phy_pma_pixel_clk_out	600

13.5.4.4 Reset Pins

The following table shows the reset names and their polarities.

Table 13-22. Reset pins and their polarities

Reset Name	Polarity
apb_preset_n	Active Low
phy_reset_n	Active Low
tap_trst_n	Active Low

13.5.5 PHY Register Read through TAP

- As required, write PHY macro diagnostic control register with 0x3 to source APB clock from TAP clock.
- Write address of APB register to PHY macro diagnostic APB access address register.
- Write 0x2 to PHY macro diagnostic APB access control register.
- Read PHY macro diagnostic APB access control register. If [0] is set high, continue. Otherwise, read PHY macro diagnostic APB access control register again.
- Read PHY macro diagnostic APB access read data register, which contains the read data.

13.5.6 PHY Register Write through TAP

- As required, write PHY macro diagnostic control register with 0x3 to source APB clock from TAP clock.
- Write address of APB register to PHY macro diagnostic APB access address register.
- Write data to be written to PHY macro diagnostic APB access write data register.
- Write 0x6 to PHY macro diagnostic APB access control register.
- Read PHY macro diagnostic APB access control register. If [0] is set high, the write access is complete. Otherwise, read PHY macro diagnostic APB access control register again.

13.5.7 Functional Description

This section describes PHY functional operation, such as start-up, disable, transmitting data, changing power state and changing the data rate.

13.5.7.1 Start-up Sequence Timing Diagram

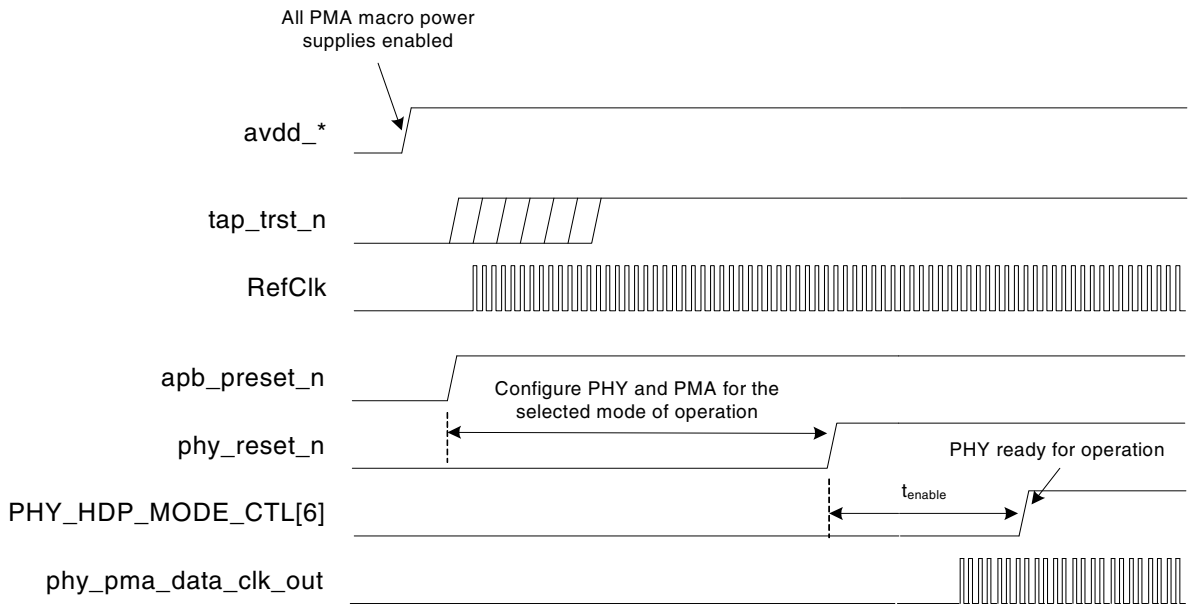


Figure 13-36. Start-up Sequence

Prior to de-assertion of phy_reset_n, write PHY_HDP_MODE_CTL[3:0] with 0x4 (0b0100). Upon start-up, the initial power state for each PMA lane must be A2. The PHY indicates ready for operation upon assertion (high) of the corresponding PHY_HDP_MODE_CTL[6] acknowledgement bit.

13.5.7.2 PHY Disable Description

To disable the PHY, assert apb_preset_n and phy_reset_n low and disable PMA reference clock (refclk_<p/m>/refclock) and apb_pclk. Note: apb_preset_n must be asserted at the same time as or after phy_reset_n is asserted. To re-enable the PHY follow the start-up sequence described above.

13.5.7.3 DP Transmit Data Path Timing Diagram

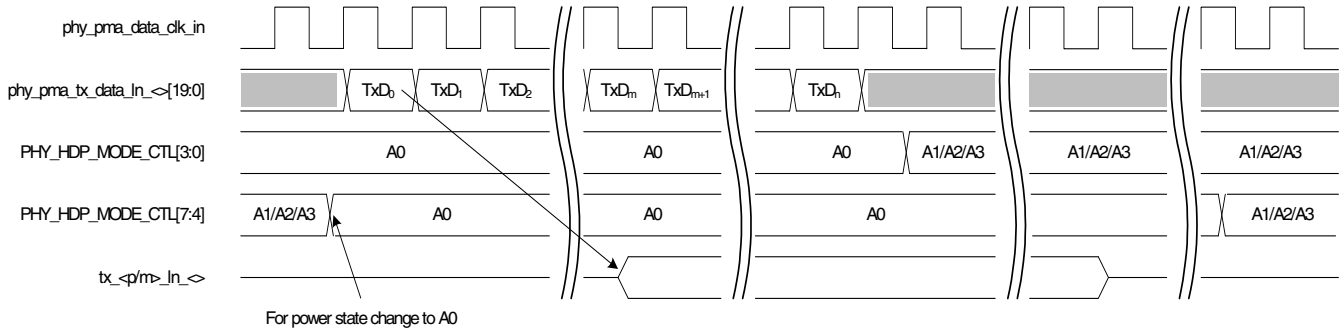


Figure 13-37. DP transmit data path

The HDP PHY must be in the A0 power state in order to transmit data. When in the A0 power state, the data provided on phy_pma_tx_data_ln_* is transmitted on tx_<p/m>_ln_*. When exiting the A0 power state, there is a race condition between the last data provided on phy_pma_tx_data_ln_* traversing the PMA and the PMA changing its power state. This may result in this data not being transmitted. Therefore, it is recommend to send a minimum of 4 dummy words into the PHY after the last valid data word prior to changing the PHY's power state.

13.5.7.3.1 HDMI TMDS Clock Lane

For the HDMI TMDS clock lane, the 20-bit data pattern depends on the HDMI data rate.

- For HDMI 1.4 data rates, a fixed 20-bit data pattern shall be driven on phy_pma_tx_data_ln_<> : 20'b1111_1000_0011_1110_0000 (5 1s, 5 0s, 5 1s, 5 0s).
- For HDMI 2.0 data rates, a repeating pattern that toggles every phy_pma_tx_data_clk_in clock cycle shall be driven on phy_pma_tx_data_ln_<> : toggle between 20'hF_FFFF and 20'h0_0000 (repeating all 1s, then all 0s).

13.5.7.4 Power State Change

The procedure for changing the PHY's power state is:

- Write PHY_HDP_MODE_CTL[3:0] with one-hot encoded value for the new power state (0b0001 = A0, 0b0010 = A1, 0b0100 = A2 and 0b1000 = A3).
- Wait for PHY_HDP_MODE_CTL[7:4] = value for the selected power state.

A0 is the active power state. A1, A2 and A3 are various power down states, with A1 having the fastest recovery to A0 time, but the lowest power savings and A3 having the slowest recovery time, but largest power savings.

13.5.7.5 Data Rate/DP Lane Configuration Change

The procedure for changing the PHY's power state and/or the DisplayPort lane configuration is:

- Write PHY_HDP_MODE_CTL[3:0] with 0b1000. (Place the PHY lanes in the A3 power state.)
- Wait for PHY_HDP_MODE_CTL[7:4] == 0b1000.
- Clear PHY_HDP_CLK_CTL[2]. Gate the PLL clocks from PMA.
- Wait for PHY_HDP_CLK_CTL[3] == 0.
- Clear PHY_HDP_CLK_CTL[0]. Disable the PLL.
- Wait for PHY_HDP_CLK_CTL[1] == 0.
- Re-configure PMA registers for the new data rate (as defined in the PMA specification / programmer's guide) and/or DP lane configuration.
- Set PHY_HDP_CLK_CTL[0]. Enable the PLL.
- Wait for PHY_HDP_CLK_CTL[1] == 1.
- Set PHY_HDP_CLK_CTL[2]. Enable PMA PLL clocks.
- Wait for PHY_HDP_CLK_CTL[3] == 1.
- Write PHY_HDP_MODE_CTL[3:0] with 0b0100 (A2 power state). The PMA must go through the A2 power state upon a data rate change.
- Wait for PHY_HDP_MODE_CTL[7:4] == 0b0100.
- As required, change the PHY power state to A0.

13.5.7.6 Lane Mapping

For the HDP PHY, the PHY lanes have a fixed mapping to the PMA lanes as follows:

- PHY HDP lane 0 → PMA lane 0 (master lane)
- PHY HDP lane 1 → PMA lane 1
- PHY HDP lane 2 → PMA lane 2
- PHY HDP lane 3 → PMA lane 3

13.5.7.7 PMA TX Sync FIFO control

Each PMA lane outputs the following Tx FIFO control signals: tx_sfifo_enq_reset_n_out_ln_<> and tx_sfifo_align_bump_out_ln_<>. Similarly, each PMA lane has a tx_sfifo_enq_reset_n_in_ln_<> and tx_sfifo_align_bump_in_ln_<>. These signals are used to properly align the Tx Sync FIFOs for lanes combined into multi- lane links in order to meet the Tx inter-lane skew requirements. There are 2

aspects to this: (1) the Tx FIFOs for each DP lane must be released from reset on the same clock cycle (through use of a common, re-synchronized tx_sfifo_enq_reset_n_out_ln_<> signal) and (20 the write pointers must be adjusted by the same amount offset from the read pointer by a common tx_sfifo_align_bump_in_<> signal.

To accomplish this, one of the PMA lanes must be selected as the master lanes. The lowest numbered enabled PMA lane (as configured in PHY_DP_MODE_CTL[15:12]) is used as the master lane.

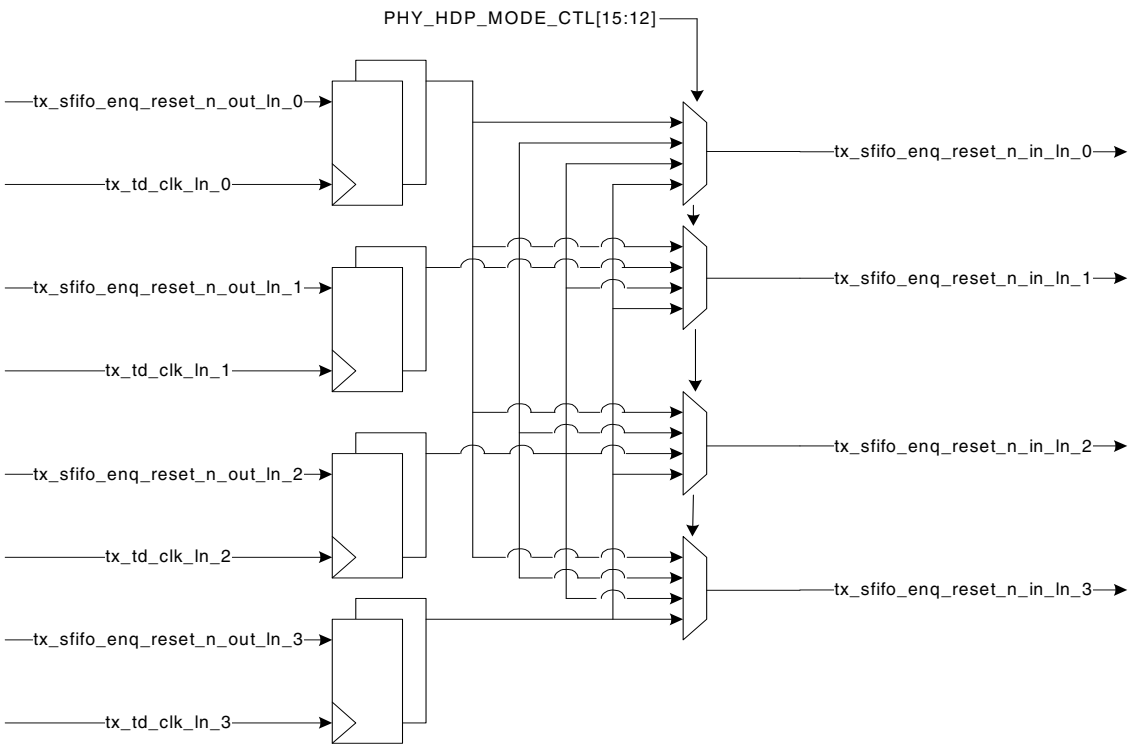


Figure 13-38. tx_sfifo_enq_reset_n_in_ln_<> generation

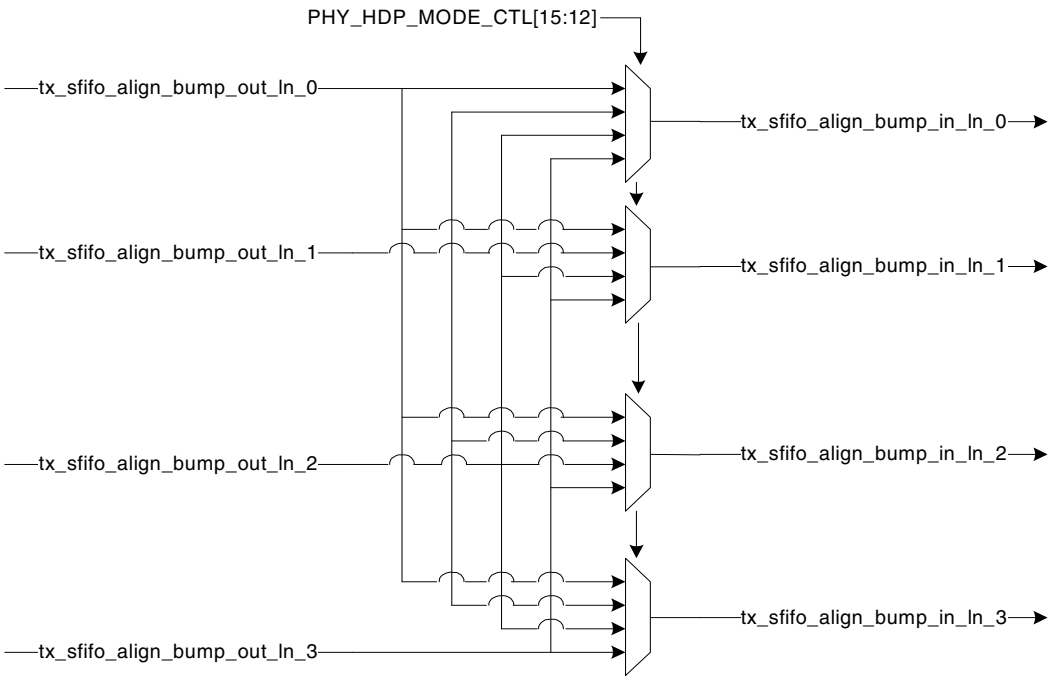


Figure 13-39. tx_sfifo_align_bump_in_ln_<> generation

13.6 MIPI DSI Host Controller (MIPI_DSI)

13.6.1 Overview

The Mobile Industry Processor Interface (MIPI) Display Serial Interface (DSI) controller is a flexible, high-performance, and easy-to-use digital core that implements all protocol functions defined in the MIPI DSI Specification. The MIPI DSI controller provides an interface that allows communication with MIPI DSI-compliant peripherals.

The MIPI DSI D-PHY is a high frequency, low power, low-cost, source-synchronous, physical layer supporting the MIPI Alliance standard for D-PHY.

13.6.2 Features

Key features of the MIPI DSI Controller Core include:

- Implements all three DSI Layers (Pixel to Byte packing, Low Level Protocol, Lane Management)
- Support for Command and Video Modes
- Host Version
- Scalable data lane support, 1 to 4 Data Lanes

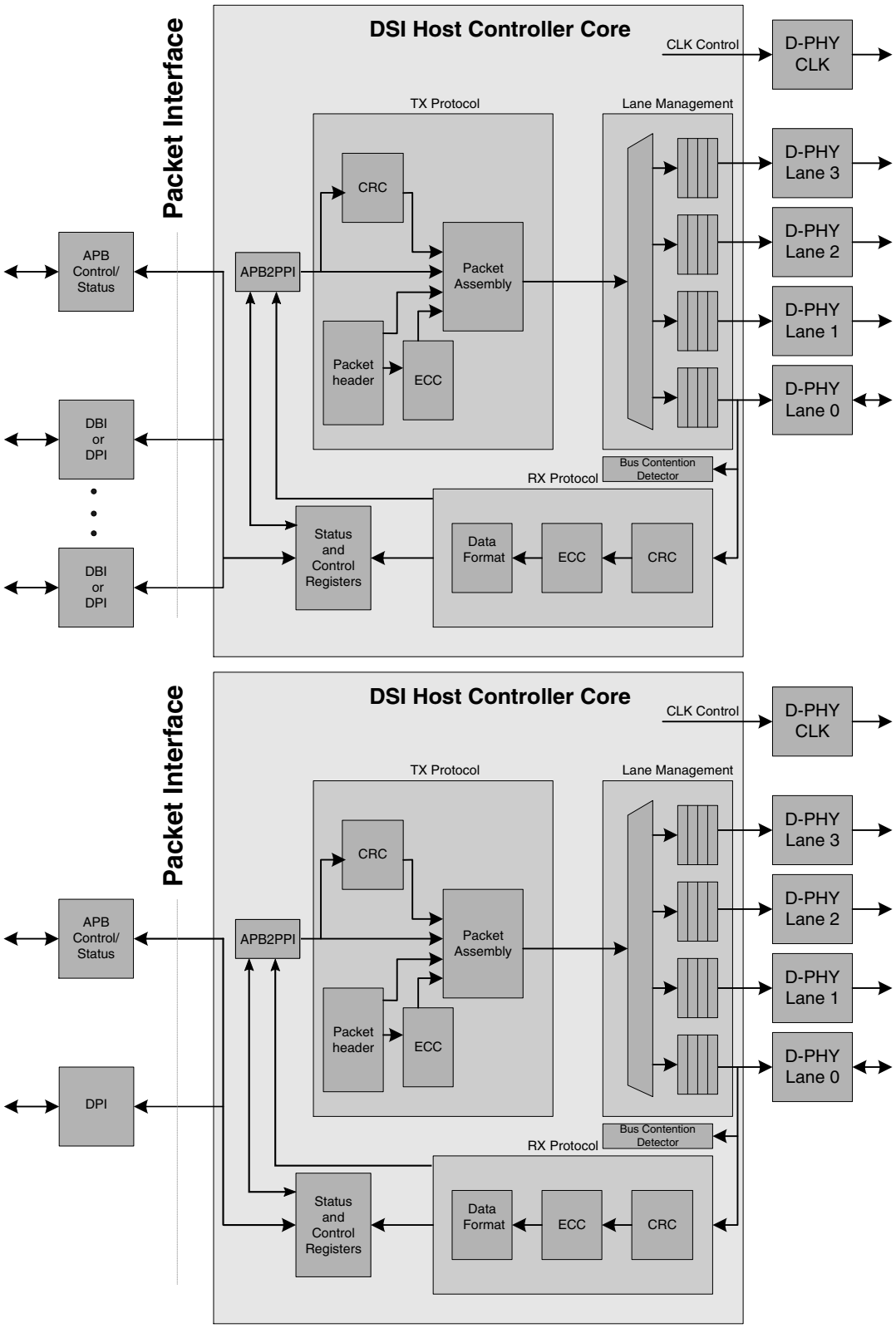
    - Optional bidirectional support on lane 0
- Supports High Speed and Low Power operation
- Support for all DSI data types and formats
- Virtual Channel support
- Supports ULPS mode
- Full Low-Level Protocol Error and Contention detection and reporting
- Supports continuous and non-continuous Clock Lane operation
- Supports multiple packets per transmission
- Support for all three Video Mode packet sequences
    - Non-Burst Mode with Sync Pulses
    - Non-Burst Mode with Sync Events
    - Burst mode
- Support for bus turnaround signaling
- Flexible packet based user interface
    - APB interface option (status and control)
    - Display Pixel Interface Core (DPI-2) option
    - Display Bus Interface Core (DBI-2) option
- Supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs
- MIPI Alliance Specification for Display Serial Interface Version 1.1 compliant

The following are the key features of the MIPI DSI D-PHY:

- 1 Clock lane
- 4 data lanes
- Supports MIPI Standard for D-PHY
- Supports both high speed and low power modes
- High Speed Serializers and Deserializers
- Resistance termination calibrator

13.6.3 DSI Host Controller Core

The figure below illustrates the DSI Host Controller Core structure. The DSI Host Controller Core operates on the host (transmit) side of a DSI link.


DSI Host Controller Core
Packet Interface
CLK Control
D-PHY CLK
TX Protocol
Lane Management
CRC
Packet Assembly
APB Control/ Status
APB2PPI
D-PHY Lane 3
D-PHY Lane 2
D-PHY Lane 1
D-PHY Lane 0
Packet header
ECC
DBI or DPI
DBI or DPI
Bus Contention Detector
RX Protocol
Status and Control Registers
Data Format
ECC
CRC
DSI Host Controller Core
Packet Interface
CLK Control
D-PHY CLK
TX Protocol
Lane Management
CRC
Packet Assembly
APB Control/ Status
APB2PPI
D-PHY Lane 3
D-PHY Lane 2
D-PHY Lane 1
D-PHY Lane 0
Packet header
ECC
DPI
Bus Contention Detector
RX Protocol
Status and Control Registers
Data Format
ECC
CRC

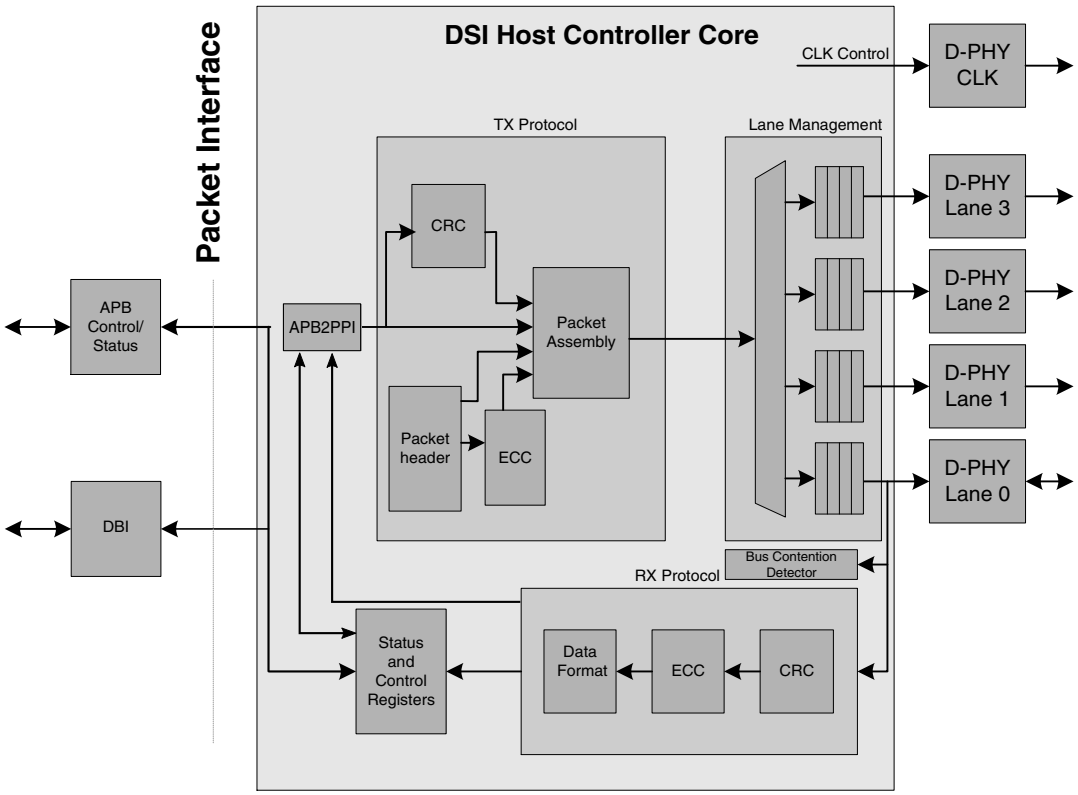


Figure 13-40. DSI Host Controller Core Block Diagram

The DSI Host Controller Core implements all three layers defined by the DSI Specification: Pixel to Byte Packing in the Application layer, Low Level Protocol, and Lane Management. The DSI Host Controller Core sends and receives DSI commands via the Packet Interface. The Packet Interface can be connected to a DBI/DPI translator or to an APB to PPI gasket.

The D-PHY interface of the DSI Host Controller Core supports up to four PHY Protocol Interface (PPI) compatible MIPI D-PHYs.

The Packet Interface is an easy-to-use data interface that accepts commands and data, and sends it over the DSI link. It supports 1 to 4 virtual channels, and the use of 1-4 D-PHY lanes. The DSI Host Controller Core takes care of all packet formatting details and transmission over the MIPI bus.

The DBI/DPI Translator connects to the DSI Host Controller Core via the Packet Interface. DBI/DPI masters may connect directly to the DBI/DPI Translator to send commands across the DSI link.

Sending commands and receiving information through the DSI link is also possible through the APB to PPI gasket. See MIPI DSI Host APB PKT IF Memory Map/Register section for details about the DSI registers which allow the APB to PPI setup and communication.

13.6.3.1 D-PHY Interface

The DSI Host Controller Core D-PHY Interface connects directly to MIPI PPI compliant D-PHYs. The DSI Host Controller Core uses this interface to transmit data, send Escape sequences, receive and transmit triggers, and detect and report D-PHY error conditions.

13.6.3.1.1 High-Speed Transmit Interface

The High-Speed Transmit interface is used to transmit High-Speed data across the MIPI interface. All signals are synchronous to the PPI TX byte clock, TxByteClkHS, and are PPI compliant.

13.6.3.1.2 Escape Mode Transmit Interface

The DSI Host Controller Core uses the Escape Mode Transmit Interface to generate Escape sequences on the MIPI Interface via the D-PHY. Escape sequences are used to change the DPHY transmit mode, go into low-power mode, or initiate a bus turnaround. The Controller generates Escape sequence requests when necessary to implement the DSI protocol. These signals are synchronous to the clk_esc.

13.6.3.1.3 Escape Mode Receive Interface

The DSI Host Controller Core uses the Escape Mode Receive Interface to receive Escape sequences on the MIPI Interface via the D-PHY. The Controller receives Escape sequence requests and Low Power data when necessary to implement the DSI protocol.

The DSI protocol uses Lane 0 in LP mode for returning data from the peripheral to the host. Receive Interface signals are only applicable to the MIPI D-PHY Lane 0. These signals are synchronous to the RxClkEsc_ln0.

13.6.3.2 Packet Interface

The DSI Host Controller Core Packet Interface consists of Transmit, Receive, and Control and Status sections. Through these interfaces, the user application can take complete control of the DSI interface, sending all video timing, sending DSI commands, receiving DSI reads, monitoring the status of the interface and responding to error reporting.

User application may have access to DSI Host Controller Core Packet Interface through DSI_HOST_APB_PKT_IF registers. For details refer to the section MIPI_DSI_HOST_APB_PKT_IF

13.6.3.2.1 Transmit Ports

The Transmit Packet Interface is the mechanism with which the user creates packets to send over the MIPI Interface.

For Long Packets, the user provides the Virtual Channel (VC) number, Data Type (DT), and Word Count (WC) to the controller. The Controller then creates a packet header and pulls the packet data from the Packet Interface and out to the D-PHY to transmit.

For Short Packets, the user provides the Virtual Channel (VC) number, Data Type (DT), and required parameters (if any) to the controller. The Controller then creates the short packet and sends it to the D-PHY to transmit. This interface enables the user application to transmit and receive any type of DSI packet.

The Packet Interface Transmit Interface ports are listed in the table below. Signals are synchronous to the Tx Byte clock.

NOTE

Only a subset of the Packet Interface Signals are accessible by user application through DSI_HOST_APB_PKT_IFx registers or chip-specific registers. For reset_byte_n and reset_esc_n signals, refer to chip-specific MIPI DSI information. For tx_payload, tx_cmd_data_type, tx_cmd_vc, and tx_cmd_byte_count signals, refer to DSI_HOST_APB_PKT_IFx registers.

Table 13-23. DSI Host Controller Core Transmit Packet Interface

PORT	TYPE	DESCRIPTION
clk_byte	Input	Byte clock input. The D-PHY PPI interface, the tx_cmd and tx_payload interfaces are synchronous to clk_byte. This clock can be independent and unrelated to clk_esc.
clk_esc	Input	The low-speed D-PHY Escape Mode clock, which becomes TxClkEsc on the D-PHY interface. This clock can be independent and unrelated to clk_byte.
reset_byte_n	Input	Asynchronous reset, active low. This reset applies to all logic in the clk_byte clock domain.
reset_esc_n	Input	Asynchronous reset, active low. This reset applies to all logic in the clk_esc clock domain.
tx_payload[31:0]	Input	Packet data input.

Table continues on the next page...

Table 13-23. DSI Host Controller Core Transmit Packet Interface (continued)

PORT	TYPE	DESCRIPTION
tx_payload_en	Output	Packet data read enable. This active high signal indicates that the controller requires a valid packet during the next clk_byte period.
tx_payload_en_last	Output	Last packet read enable, active high signals last cycle of tx_payload_en.
tx_cmd_data_type[5:0]	Input	Transmit packet DSI data type. It is written into the command buffer when tx_cmd_ack is asserted high.
tx_cmd_vc[1:0]	Input	Transmit packet command virtual channel. It is written into the command buffer when tx_cmd_ack is asserted high.
tx_cmd_byte_count[15:0]	Input	Transmit packet payload byte count. It is written into the command buffer when tx_cmd_ack is asserted high. For DSI Long packet types, tx_cmd_byte_count defines the number of bytes of packet data to pull from the tx_payload port. For DSI Short packets, the format of tx_cmd_byte_count contains any optional parameters. If the SDI Short packet type does not have any parameters, it is recommended to set tx_cmd_byte_count to all 0s.
tx_cmd_req	Input	Transmit packet command request. This active high signal informs the controller that the packet command is valid. The packet command consists of the ports tx_cmd_data_type, tx_cmd_vc, and tx_cmd_byte_count. The controller will assert tx_cmd_ack when it accepts the command, after which, the user should either update port values for the next transmit packet command or deassert tx_cmd_req.
tx_cmd_ack	Output	Transmit packet command request acknowledge. This active high signal indicates that the controller has accepted the TX packet request and the user logic should either submit a new request or deassert tx_cmd_req on the next rising edge of clk_byte.
trigger_req	Input	Transmit trigger request. This active high signal informs the controller that the trigger number on trigger_send is valid. The controller will assert trigger_ack when it accepts the command, after which, the user should either put update trigger_ack with the values for the next transmit packet or deassert trigger_req.
trigger_ack	Output	Transmit trigger request acknowledge. This active high signal indicates that the controller has accepted the trigger request and the user logic should either submit a new request or deassert trigger_req on the next rising edge of user_clk.
trigger_send[1:0]	Input	Transmit trigger. The trigger number on trigger_send is sampled when trigger_ack is asserted high. The format of trigger_send is as follows: 1’b00 = Trigger 0 (Reset-Trigger) 1’b01 = Trigger 1 ([Reserved]) 1’b10 = Trigger 2 ([Reserved]) 1’b11 = Trigger 3 ([Reserved])

13.6.3.2.2 Receive Ports

The Receive Packet Interface returns data from the Peripheral to the user. The user is provided the Virtual Channel (VC) number, Data Type (DT), Word Count (WC), and Rx Payload Data.

This interface enables the user application to receive any type of DSI packet. These signals are synchronous to the clk_byte clock. The Packet Interface Receive Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals are accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- rx_payload
- rx_cmd_vc
- rx_cmd_data_type
- rx_cmd_byte_count
- ecc_one_bit_error
- ecc_two_bit_error
- ecc_one_bit_error_pos[4:0]
- ecc_err
- ecc_err_pos[2:0]
- crc_err

See DSI_HOST_APB_PKT_IFx and DSI_HOST device sections for details.

Table 13-24. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
rx_payload[31:0]	Output	Received packet data output. The Host Receive Packet Interface presents 4 bytes at a time. Bytes are valid in this interface according to the rx_cmd_byte_count signal, beginning with the lowest byte.
rx_payload_valid	Output	Packet data valid. This active high signal indicates that the controller is presenting valid packet during the next clk_byte period.
rx_payload_valid_last	Output	This data is the last of the packet, active high signals last cycle of rx_payload_valid.
rx_cmd_valid	Output	Packet header data is valid on the packet header ports below when this signal is asserted.
rx_cmd_vc[1:0]	Output	Packet virtual channel number, valid when rx_cmd_valid is asserted.
rx_cmd_data_type[5:0]	Output	Packet data type, valid when rx_cmd_valid is asserted. See the MIPI DSI-2 specification for a definition of possible values.

Table continues on the next page...

Table 13-24. DSI Host Controller Core Receive Packet interface (continued)

PORT	TYPE	DESCRIPTION
rx_cmd_byte_count[15:0]	Output	Packet Word Count (byte count). Contains the number of bytes of data in the received packet. Valid when rx_cmd_valid is asserted.
ecc_one_bit_error	Output	Single bit error in the packet header was detected and corrected. Active high. Valid when rx_cmd_valid is high.
ecc_two_bit_error	Output	Two packet header bit errors were detected and not corrected, active high. Valid when rx_cmd_valid is high.
ecc_one_bit_error_pos[4: 0]	Output	Position of the corrected single bit error in the packet header. Valid when ecc_one_bit_error is high.
ecc_err	Output	Error detected in the ECC bits. Active high. Valid when rx_cmd_valid is high.
ecc_err_pos[2:0]	Output	Position of the erroneous bit in the ECC bits, valid when ecc_err is asserted.
crc_err	Output	Asserts high when the CRC calculated on the received data does not reach the end of the packet.

13.6.3.3 Status Interface Ports

The Status Interface Core provides for reading the Status of the DSI Host Controller Core via a simple interface. All signals in the Status Interface are synchronous to the clk_byte clock. The Status Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals are accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- status_out
- hs_tx_timeout
- lp_rx_timeout

Refer to DSI_HOST_APB_PKT_IFx and DSI_HOST device sections for details.

Table 13-25. DSI Host Controller Status Interface Port

PORT	TYPE	DESCRIPTION
status_rd	Input	Initiates a read of the status register. Asserting this input high will result in the contents of the status register being presented on status_out[15:0] along with status_out_valid asserting high to indicate that status_out[15:0] contains a valid value.

Table continues on the next page...

Table 13-25. DSI Host Controller Status Interface Port (continued)

PORT	TYPE	DESCRIPTION
status_addr[3:0]	Input	When status_rd is asserted, indicates which status register is to be returned. Address 0x0 is a valid address while 0x1-0xF are reserved.
status_out[31:0]	Output	Contains the contents of the addressed status register.
status_out_valid	Output	When asserted high, indicates that the data on status_out is valid.
tx_active	Output	tx_active asserts high when the Host Controller is actively transmitting data or when it has accepted a request from the user but has not yet started transmitting.
hs_tx_timeout	Output	Asserts high for one clk_byte period to indicate that a High Speed transmit has timed out.
lp_rx_timeout	Output	Asserts high for one clk_byte period to indicate that a Low Power RX curred.

13.6.3.4 Control and Status Ports

The Control and Status Packet Interface allows the user to control and receive the status of the underlying link between the DSI Host and DSI Peripheral. It works on the same clk_byte clock as the Host Transmit Packet Interface. The Packet Interface Control and Status Interface ports are listed in the table below.

NOTE

Only a subset of the Packet Interface signals may be accessible by user application through DSI_HOST_APB_PKT_IFx registers and DSI_HOST device registers:

- dphy_direction
- tx_hs_mode
- dphy_turnaround

Please refer to DSI_HOST_APB_PKT_IFx and DSI_HOST device sections for details.

Table 13-26. DSI Host Controller Core Control And Status Packet Interface

PORT	TYPE	DESCRIPTION
tx_hs_mode	Input	Switches the DPHY into High Speed Data Transfer mode or Low Power Data Transfer mode. 1'b1 = request HS mode 1'b0 = request LP mode. The Packet interface will not acknowledge packet commands or data while switching modes.

Table continues on the next page...

Table 13-26. DSI Host Controller Core Control And Status Packet Interface (continued)

PORT	TYPE	DESCRIPTION
dphy_turnaround	Input	Requests bus turnaround. 1'b1 = Request reverse direction LP mode, from Host TX to Host RX. 1'b0 = No effect. This signal is ignored if the bus is already in Reverse (Host RX) direction.
dphy_direction	Output	Reports the current bus direction. 1'b1 = Bus is in Reverse direction (Host RX). 1'b0 = Bus is in Forward direction (Host TX).

13.6.3.5 Interface Cores

There are several interface cores for the DSI Host Controller Core. These interface cores provide industry standard interfaces to the DSI Host Controller Core to simplify interfacing and integration the DSI Host Controller Core to your application. When selected, these interface cores provide additional ports to the DSI Host Controller core.

13.6.3.5.1 DPI-2 Interface Host Bridge Core

The DPI-2 Interface Host Bridge Core provides a DPI-2 compliant interface to the DSI Host controller. The DPI-2 Interface Host Bridge Core handles converting the DPI-2 interface into a packet format that is compatible with the DSI Host Controller Core's Packet Interface, and that adheres to the DSI specification. The DPI-2 Interface Core handles all pixel-to-byte packing for all DPI data types.

The DSI Host Controller DPI-2 Interface Core provides the following features:

- Support for Type 2,3, and 4 displays
- Support for 16-, 18- and 24- bit Pixel data and all alignment configurations
- Support for 16, 18, 24, and 18 bit loosely packed DSI data types
- Supports DSI video modes
  - Non-Burst Mode with Sync Pulses
  - Non-Burst Mode with Sync Events
  - Burst Mode
- Supports normal or inverted HSYNC and VSYNC signals
- Handles clock domain crossing from DPI Pixel clock to the Host controller TX Byte clock
- Interfaces directly to the Host Controller's DSI Packet Interface
- Comes already integrated with the DSI Host Controller

A block diagram of the DPI-2 Interface Host Bridge Core is shown below. The DPI-2 Host Bridge Core accepts a MIPI compliant DPI-2 set of signals (vsync, hsync, pixel data), creates DSI packets for the video timing events and video data and transmits those packets via the DSI Host Controller's Packet Interface.

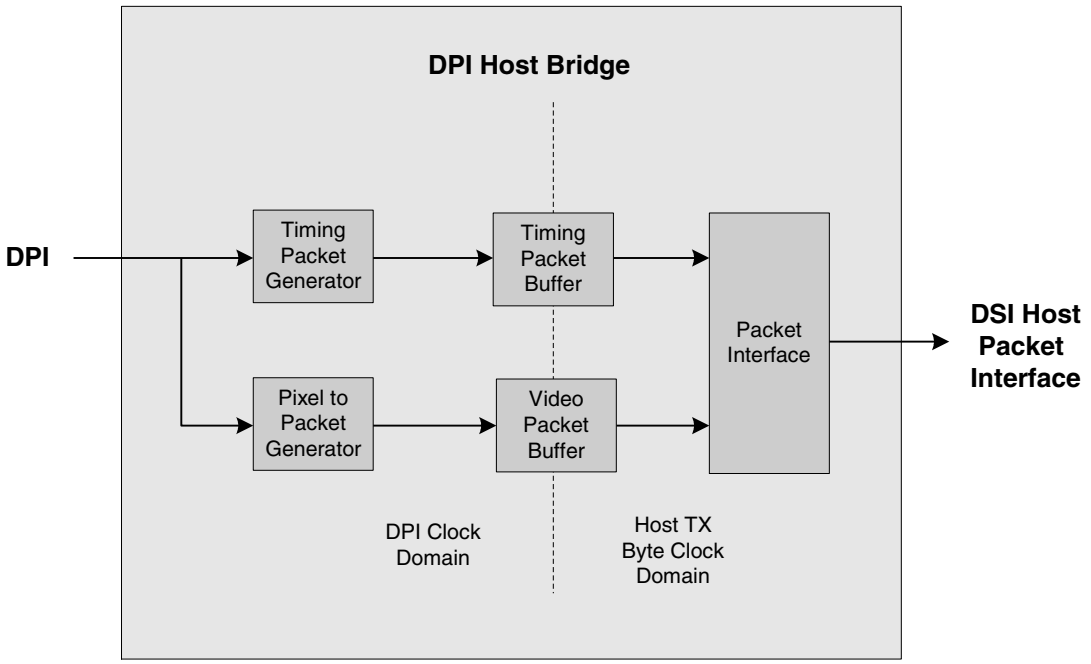


Figure 13-41. DPI-2 Host Bridge Core Block Diagram

13.6.3.5.1.1 DPI-2 Interface Ports

The DPI-2 Interface module provides a MIPI compliant DPI-2 interface to the DSI Host Controller Core. In addition to the standard DPI-2 interface signals (prefixed with a dpi_) there are several configuration and status ports that allow the user to adjust the behavior of the DPI-2 Interface module to better suit a particular application.

NOTE

The DPI configuration signals herein described are accessible to User Application through chip-specific registers (please refer to chip-specific MIPI-DSI information section for details) and DSI_HOST_DPI_INTFC device registers (please refer to MIPI DSI Host Memory Map/Register section for details).

Table 13-27. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
dpi_sd	Input	Shut Down – Control to shutdown display (type 4 only) 1'b1= Send shutdown command. 1'b0= No effect

Table continues on the next page...

Table 13-27. DSI Host Controller Core Receive Packet interface (continued)

PORT	TYPE	DESCRIPTION
dpi_cm	Input	Color Mode control. 1'b0== Normal Mode 1'b1== Low-color Mode
cfg_dpi_pixel_payload_size [15:0]	Input	Maximum number of pixels that should send as one DSI packet. Recommended to be evenly divisible by the line size (in pixels).
cfg_dpi_pixel_fifo_send_ level[15:0]	Input	In order to optimize DSI utility, the DPI bridge buffers a certain number of DPI pixels before initiating a DSI packet. This configuration port controls the level at which the DPI Host bridge begins sending pixels.
cfg_dpi_interface_color_ coding[2:0]	Input	Sets the distribution of RGB bits within the 24-bit d bus, as specified by the DPI specification. 0= 16-bit Configuration 1 1= 16-bit Configuration 2 2= 16-bit Configuration 3 3= 18-bit Configuration 1 4= 18-bit Configuration 2 5= 24-bit
cfg_dpi_pixel_format[1:0]	Input	Sets the DSI packet type of the pixels. 0= 16-bit 1= 18-bit 2= 18-bit loosely packed, 3= 24-bit
dpi_host_underrun_err	Output	During DSI Host transmission of DPI data insufficient DPI data was received. This may indicate that DPI_CLK is too slow, or that the cfg_dpi_* parameters are incorrectly set.
cfg_dpi_vsync_polarity	Input	Sets polarity of dpi_vsync input, 0 – active low, 1 active high
cfg_dpi_hsync_polarity	Input	Sets Polarity of dpi_hsync input, 0 – active low, 1 – active high
cfg_host_dpi_video_mode[1:0]	Input	Select DSI video mode that the host DPI module should generate packets for. 2'b00 – Non-Burst mode with Sync Pulses 2'b01 – Non-Burst mode with Sync Events 2'b10 – Burst mode 2'b11 – Reserved, not valid
cfg_host_dpi_hfp[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal front porch blanking packet.
cfg_host_dpi_hbp[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal back porch blanking packet.
cfg_host_dpi_hsa[15:0]	Input	Sets the DSI packet payload size, in bytes, of the horizontal sync g packet.
cfg_enable_mult_packets	Input	Enable Multiple packets per video line. When enabled, cfg_dpi_pixel_payload_size must be set to exactly half the size of the video line. 0 – Video Line is sent in a single packet 1 – Video Line is sent in two packets
cfg_bllp_mode	Input	Optimize bllp periods to Low Power mode when possible 0 – blanking packets are sent during BLLP periods 1 – LP mode is used for BLLP periods

13.6.3.5.1.2 DPI-2 Video Timing Recreation over DSI

The DSI Host DPI-2 module is designed to preserve the receive video timing by properly spacing the DSI packets that carry the DPI video information and by inserting blanking packets into the packet stream. In order to achieve this goal, the DSI Host controller needs to have the cfg_host_dpi_hfp, cfg_host_dpi_hbp, and cfg_host_dpi_hsa ports

properly set. The values of these ports reflect the size of the hfp, hbp, and has in terms of DSI packet bytes, values such that the number of DSI bytes when transmitted takes approximately the same amount of absolute time as the video event took on the DPI interface.

To relate pixel sizes on the DPI interface to DSI packet bytes, use the following relationship:

```
      Time of DPI event - time to transmit x number of bytes on the DSI interface

      dpi_event_size * dpi_pclk_period = number_of_dsi_bytes * 8 * dsi_hs_bit_period /
cfg_num_lanes
```

Solving for number_of_dsi_bytes:

```
      number_of_dsi_bytes = dpi_event_size * dpi_pclk_period * cfg_num_lanes / (8 *
dsi_hs_bit_period)

      number_of_dsi_bytes - value to set the cfg_host_dpi_hfp (hbp, or hsa) port to
      dpi_event_size - size of the video event in pixels (hfp, hbp, hsa)
      dpi_pclk_period - period of dpi_pclk
      dsi_hs_bit_period - bit period of the mipi data lanes in High Speed mode
      cfg_num_lanes - host controller setting for number of active lanes, 0 = 1 lane, 1 = 2
lanes, 2 = 3 lanes, 3 = 4 lanes)
```

13.6.3.5.2 DBI-2 Interface Core

The DBI-2 Interface Core provides a DBI-2 compliant interface that attaches to a Packet Interface Port on the DSI Host controller. The DBI-2 Interface Core handles converting the DBI-2 interface into a packet format that is compatible with the DSI Host Controller Core's Packet Interface and adheres to the DSI specification. The DBI-2 Interface Core handles all pixel-to-byte packing for all DBI data types.

The DSI Host Controller DBI-2 Interface Core provides the following features:

- Support for Type 1, 2, and 3 displays
- Support for DBI-2 Type B and C Interfaces
- Support for 8-, 9- and 16- bit data (Type B)
- Interfaces directly to the Host Controller's Packet Interface
- Comes already integrated with the DSI Host Controller
- Source Code License available (Verilog)
- Provided with expert technical support
- Customization and integration services available

13.6.3.5.2.1 DBI-2 Host Bridge Module Interface Ports

The DBI-2 Interface module provides a MIPI compliant DBI-2 interface to the DSI Host Controller Core. In addition to the standard DBI-2 interface signals (prefixed with a dbi_) there are several configuration and status ports that allow the user to tune adjust the behavior of the DBI-2 Interface module to better suit a particular application.

Table 13-28. DSI Host Controller Core Receive Packet interface

PORT	TYPE	DESCRIPTION
dbi_csx	Input	DBI Chip Select, active low.
dbi_wrx	Input	DBI Write enable. Write data/command is registered at the rising edge of dbi_wrx.
dbi_rdx	Input	DBI Read enable. Read data is read at the rising edge of dbi_rdx.
dbi_d_write[15:0]	Input	8-,9- or 16-bit Input Data for Writes
dbi_d_read[15:0]	Output	8-,9- or 16-bit Output Data for Reads
dbi_dcx	Input	Data/Command indicator. 0 – Command cycle 1 – Data cycle
dbi_resx	Input	Active low DBI Display Module reset
dbi_te	Output	Tearing Effect (optional support)
dbi_host_underrun_err	Output	During DSI Host transmission of DBI data insufficient DBI data was received. This may indicate that the rate of data received on dbi_d_in was insufficient to keep up with the outgoing data rate on the MIPI interface, or that the cfg_dbi_* parameters are incorrectly set.
cfg_dbi_pixel_payload_size[15:0]	Input	Maximum number of pixels that should send as one DSI packet. Recommended to be evenly divisible by the line size (in pixels).
cfg_dbi_pixel_fifo_send_level[15:0]	Input	In order to optimize DSI utility, the DBI bridge buffers a certain number of DBI pixels before initiating a DSI packet. This configuration port controls the level at which the DBI Host bridge begins sending pixels.

13.6.3.5.2.2 DBI-2 Host Bridge Module Write and Read Waveforms

The figure below shows a DBI-2 Write transaction. The command portion of the write is registered with the rising edge of dbi_wrx when dbi_dcx and dbi_csx are low. The write data is registered with the rising edge of dbi_wrx when dbi_dcx is high and dbi_csx is low.

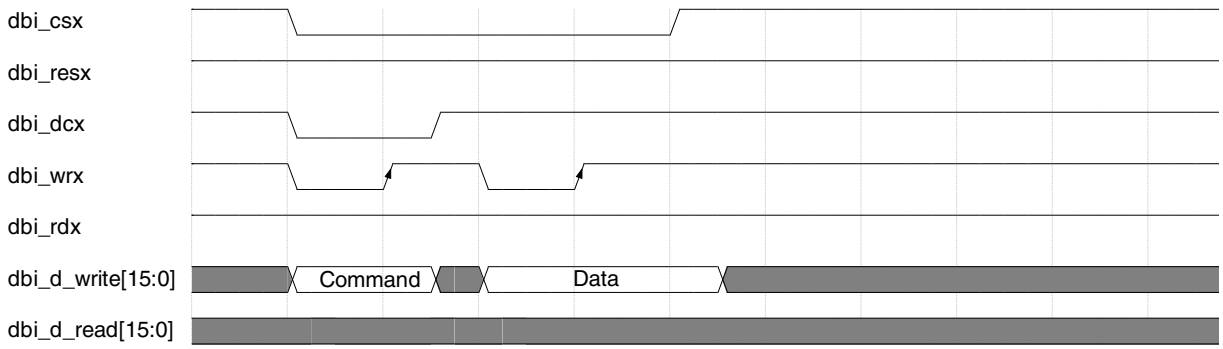


Figure 13-42. DBI-2 Write Waveform

The figure below shows a DBI-2 Read Transaction. The command portion of the read is registered with the rising edge of dbi_wrx when dbi_dcx and dbi_csx are low. The first cycle of read data is invalid with the first rising edge of dbi_rdx while the second cycle of read data is valid with the rising edge of dbi_rdx.

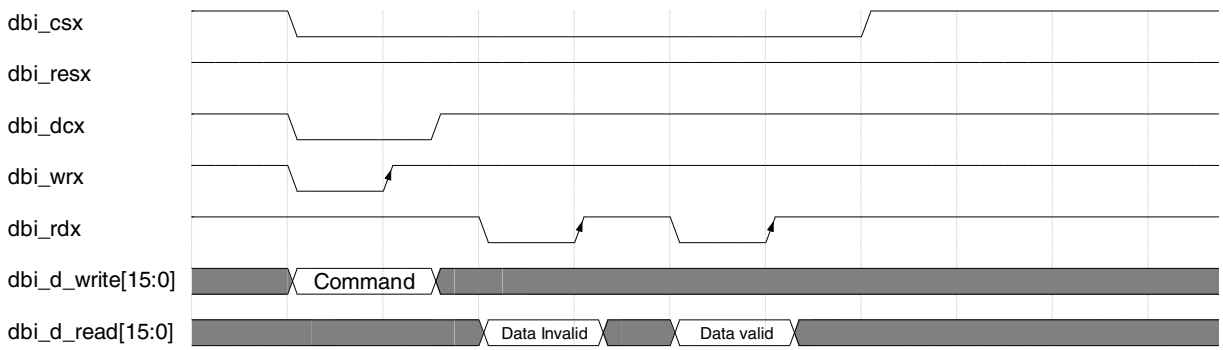


Figure 13-43. DBI-2 Read Waveform

13.6.3.6 RAM Usage

The DSI Host Controller Core requires on-chip RAM arrays. These RAM arrays and their corresponding sizes are listed in the table below.

All RAM arrays used in this core are configured as two-port, having the following attributes:

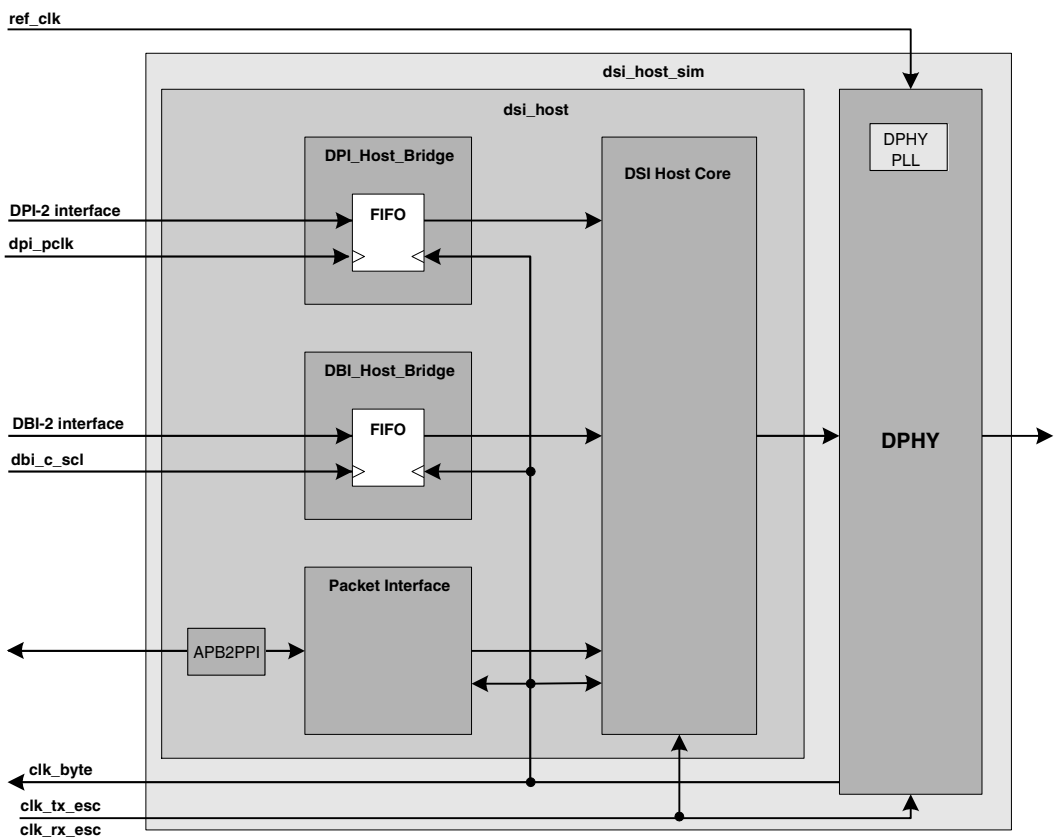
- Two address busses, one for the write port and one for read port
- Two data busses, one for write data on the write port and one for read data on the read port
- Two clock inputs, one for the write port and one for the read port. (Note: In some cases, the core logic will connect the same clock to both the read and write clock inputs of the RAM.)

Ram Purpose	Size (Depth x Width)
DPI Core Pixel Data FIFO	2048 x 32
APB to Packet Interface TX FIFO	64 x 32
APB to Packet Interface RX FIFO	64 x 32

13.6.3.7 DSI Host Clocking

The DSI Host Controller requires the following clocks:

- Reference clock (ref_clk) - used by the DPHY (DPHY PLL generates byte clock)
- Byte clock (clk_byte) - generated by the DPHY, used for packet generation and transfer to the DPHY
- TX escape mode clock (clk_tx_esc) - for internal LPDT state machines and low-power transmissions;
- RX escape mode clock (clk_rx_esc) - used by DPHY for reverse low-power reception.
- Pixel clock (dpi_pclk)
- Serial clock (dbi_c_scl)



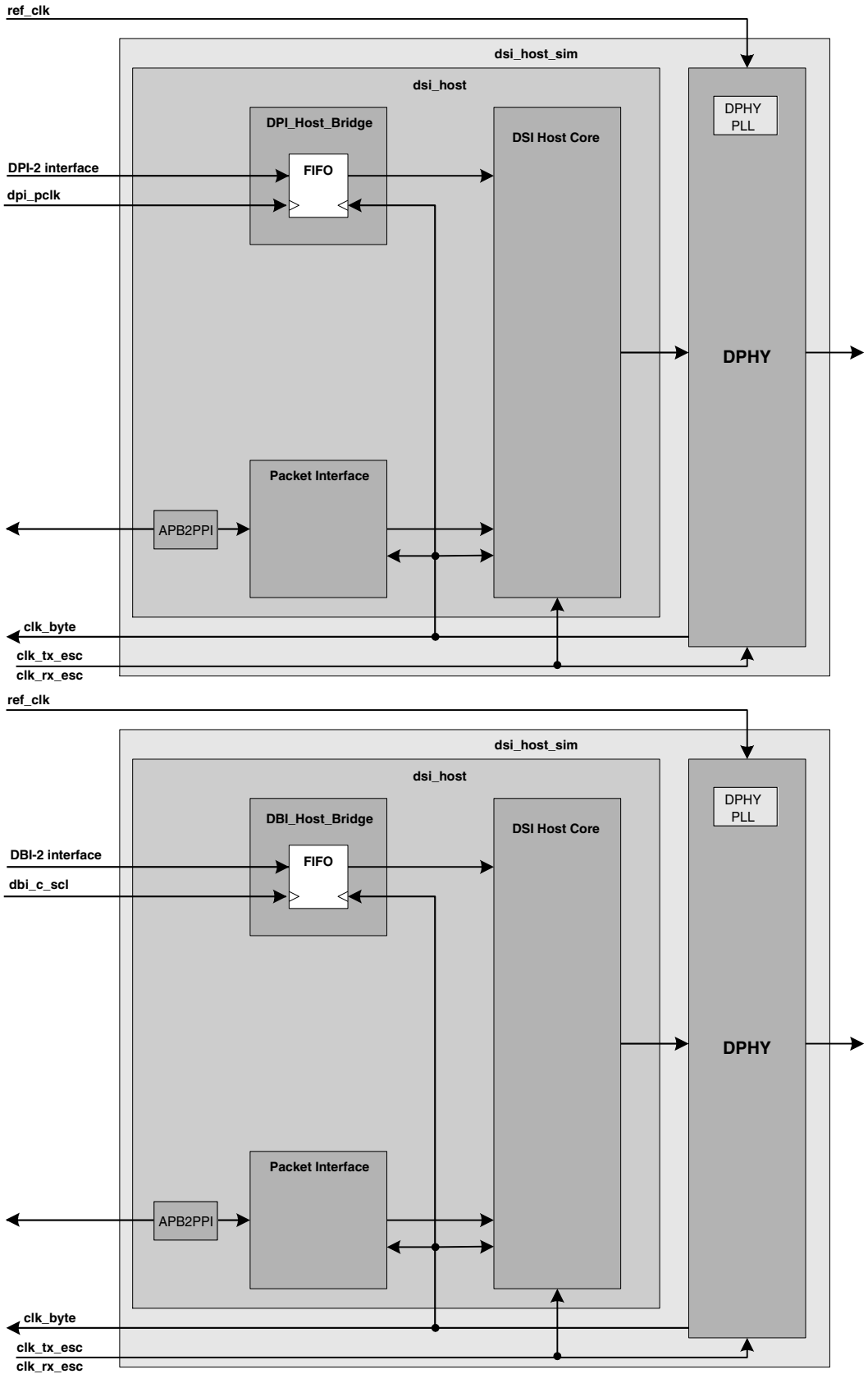


Figure 13-44. DSI Host Controller Clocking Block Diagram

13.6.3.7.1 ref_clk

The reference clock is used by the DPHY (DPHY PLL) to generate the High Speed MIPI clock and High Speed Data Lane signaling. The frequency of the resulting clock generated from DPHY PLL is typically equal to the high speed bit rate of the data lanes. The DPHY will also use this clock to generate the clk_byte clock output used by the Host DSI Controller and user application logic.

The MIPI DPHY PLL must be setup through the MIPI_DSI_HOST_FSL_IP1_DPHY INTFCx registers, prior to starting DSI Host Controller and DPHY operation. For details, please refer to MIPI DSI Host IP1 DPHY INTFC Memory section.

13.6.3.7.2 clk_byte

The clk_byte clock is generated by the DPHY (DPHY PLL) and the frequency is 1/8th the data rate of the DPHY Data Lane High Speed data rate. A DPHY configured to generate High Speed data at 1Gbps/lane would generate a 125MHz clk_byte clock.

The clk_byte clock is used by the DSI Host Controller to interface to the MIPI Tx DPHY High Speed PPI interface. The DSI Host Controller's Packet Interface is synchronous to the clk_byte clock.

13.6.3.7.3 dpi_pclk

The dpi_pclk clock is used on the Host DPI-2 interface. All of the Host DPI-2 signals are synchronous to this clock. The DSI Host Controller's DPI Bridge module handles transferring video data received in the dpi_pclk clock domain over to the clk_byte clock domain.

The dpi_pclk and clk_byte frequencies are related by the following formula:
clk_byte_freq >= dpi_pclk_freq * DPI_pixel_size / (8 * (cfg_num_lanes + 1))

cfg_num_lanes = the configuration port setting that selects the number of active MIPI DPHY data lanes clk_byte_freq = frequency of clk_byte which is 1/8th the High Speed data lane rate. dpi_pclk_freq = frequency of the dpi_pclk clock on the DPI-2 interface. DPI_pixel_size = size of pixels, in bits, on the DPI-2 interface

If the clk_byte frequency does not meet this requirement, the MIPI interface will not be able to keep up with the video stream on the DPI-2 interface resulting in dropped video lines.

13.6.3.7.4 clk_tx_esc

The clk_tx_esc clock is used by the MIPI Tx DPHY for state control and low power data transmission. The DSI Host Controller also uses clk_tx_esc for the portion of controller logic that interfaces to the MIPI Tx DPHY that are synchronous to clk_tx_esc. The frequency of clk_tx_esc is defined by the requirements of the MIPI interface and the MIPI Tx DPHY. For details about the clk_tx_esc source, please refer to the chip-specific MIPI-DSI clocking section.

13.6.3.7.5 clk_rx_esc

The clk_rx_esc clock is used by the MIPI Tx DPHY for reverse low power data reception. The frequency of clk_rx_esc is defined by the Tx DPHY and MIPI DPHY interface timing requirements.

13.6.3.7.6 dbi_c_scl

The dbi_c_scl clock is used by the optional DBI-C interface. The MIPI DBI-2 specification defines the minimum period for dbi_c_scl to be 80ns. The DSI Host Controller DBI-2 bridge module handles the clock domain crossing from dbi_c_scl clock domain to the clk_byte clock domain.

13.6.3.8 Reset and Initialization

Reset and Initialization procedure for the DSI Host Controller is as follows:

1. Assert all resets through the chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n
2. Setup DSI related clocks through chip-specific registers provided for this purpose (refer to CCM for details).
3. Setup MIPI DSI application by programming all DSI Host Controller and DSI DPHY required parameters (refer to MIPI DSI Host Memory Map/Register for details).
4. Deassert all remaining resets through the Chip-specific registers provided for this purpose:
   - reset_dpi_n
   - reset_byte_n
   - reset_esc_n
5. DSI Host Controller should be ready for use

13.6.4 DSI D-PHY

The figure below shows the block diagram of the MIPI DSI D-PHY. It consists of a clock lane module and four data lane modules. In addition to the data lanes, MIPI DSI D-PHY includes a calibrator module for calibration of termination resistance. Each of these PHY Lane Modules communicates through a differential line to a complementary PHY at the other side of the lane interconnect.

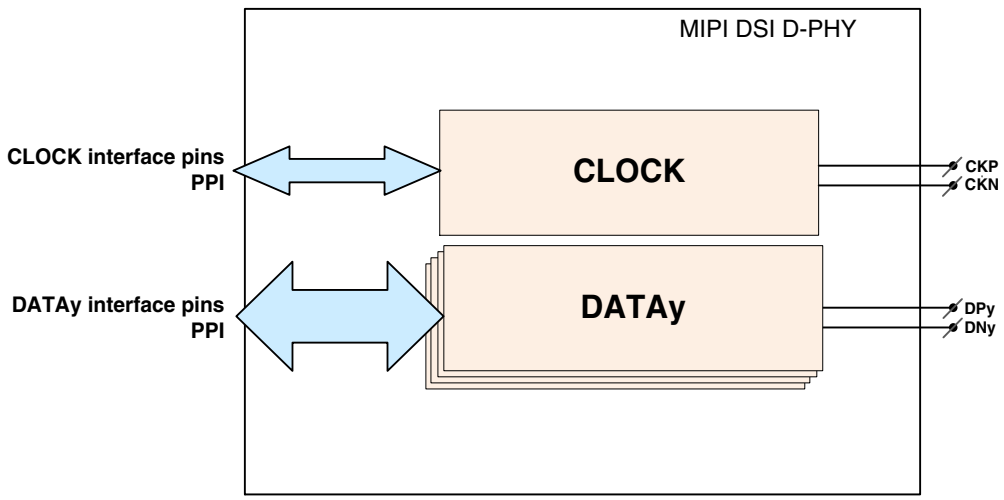


Figure 13-45. MIPI D-PHY Block Diagram

The figure below shows the D-PHY Lane components:

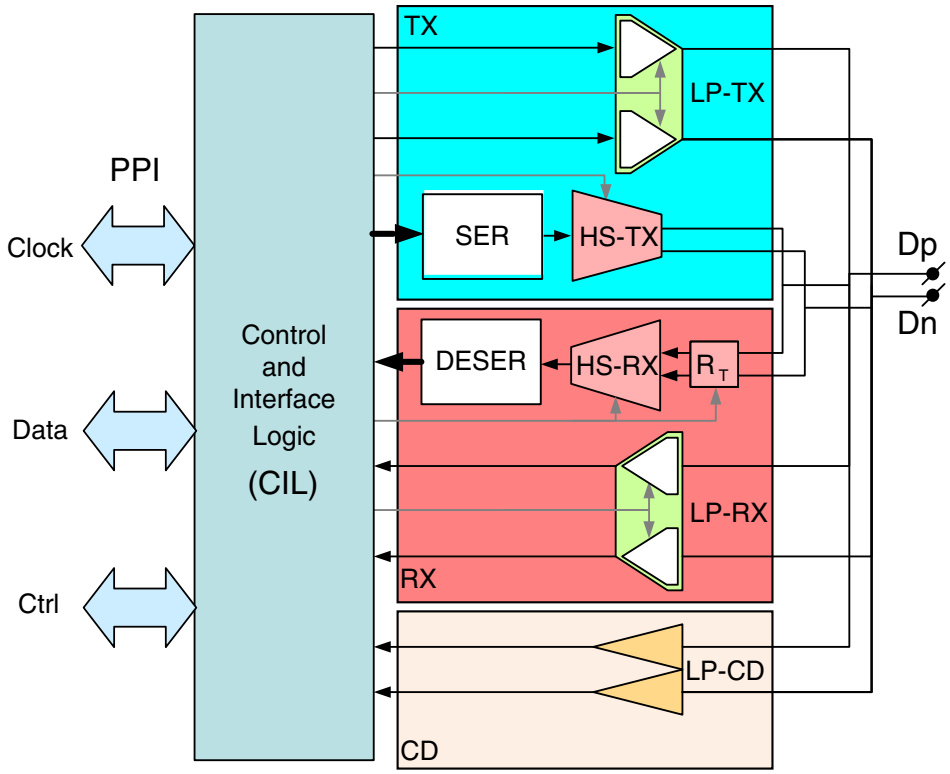


Figure 13-46. D-PHY Lane Diagram

The DATAy are bidirectional lanes and consist of High-Speed Receiver (HS-RX), Low-Power Receiver (LP-RX), High-Speed Transmitter (HS-TX), Low-Power Transmitter (LP-TX), Serializer (SER), De-Serializer (DESER), and Low- Power Contention Detector (LP-CD).

The Clock lane is a bidirectional lane and consists of High-Speed Receiver (HS-RX), Low-Power Receiver (LP-RX), High-Speed Transmitter (HS-TX), and Low-Power Transmitter (LP-TX). The B is optional and is used to calibrate the HS-TX and HS-RX termination, which requires connecting the pin REXT with a precise external resistor.

The Control and Interface Logic (CIL) module interfaces with the Protocol and determines the global operation of the Lane Module. The interface between the D-PHY and the protocol is called the PHY-Protocol Interface (PPI).

During normal operation, a Lane switches between Low-Power and High-Speed mode. Bidirectional Lanes can also switch communication direction. The change of operating mode or direction requires enabling and disabling of certain electrical functions. These enable and disable events do not cause glitches on the Lines that would result in a detection of an incorrect signal level. Therefore, all mode and direction changes are smooth to always ensure a proper detection of the Line signals.

13.6.4.1 High-Speed Transmitter (HS-TX)

The differential output voltage VOD is defined as the difference of the voltages V_{DP} and V_{DN} at the Dp and Dn pins, respectively.

VOD =VDP −VDN

The output voltages V_{DP} and V_{DN} at the Dp and Dn pins will not exceed the high-speed output high voltage V_{OHHS}. The common-mode voltage V_{CMTX} is defined as the arithmetic mean value of the voltages at the Dp and Dn pins:

VCMTX = (VDP + V DN)/2

V_{OD} and V_{CMTX} are graphically shown in Figure 13-47 for ideal HS signals. Figure 13-48 shows single ended HS signals with the possible kinds of distortion of the differential output and common mode voltages. V_{OD} and V_{CMTX} may be slightly different for driving a Differential-1 or a Differential-0 on the pins.

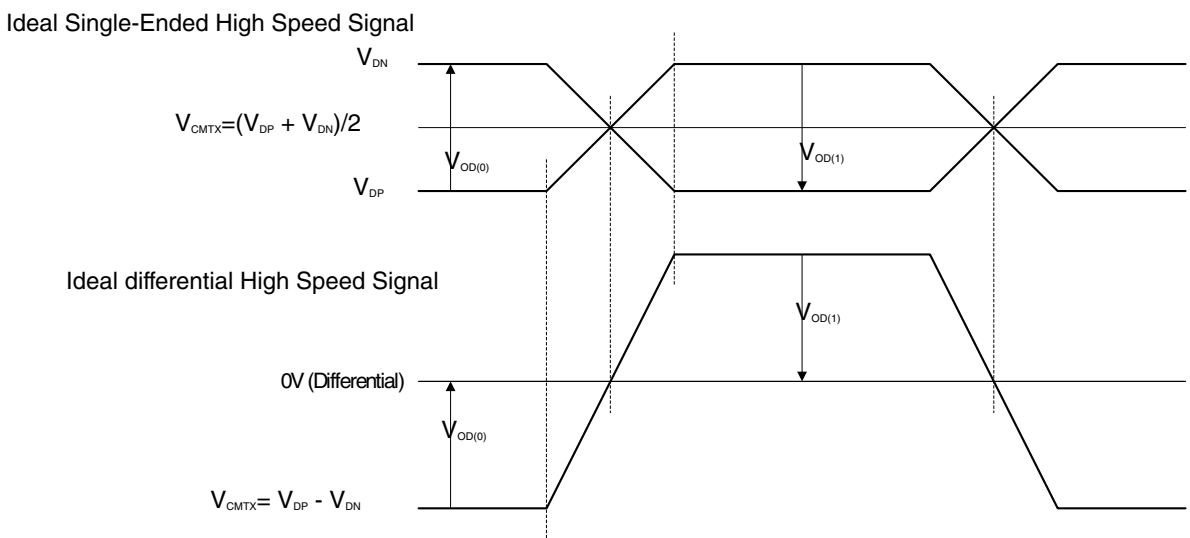


Figure 13-47. Ideal Single-ended and Resulting Differential HS Signals

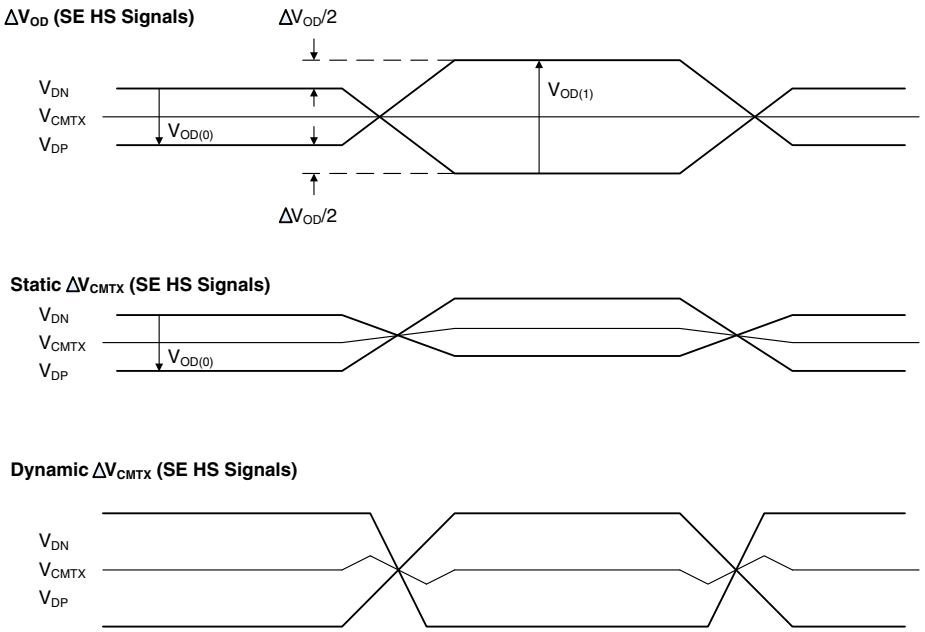


Figure 13-48. Possible ΔVcmtx and ΔVod Distortions of the Single-Ended HS Signals

The output differential voltage mismatch ΔVOD is defined as the difference of the absolute values of the differential output voltage in the Differential-1 state VOD(1) and the differential output voltage in the Differential-0 state VOD(0). This is expressed by:

$$\Delta V = |V_{OD(1)}| - |V_{OD(0)}|$$

If VCMTX(1) and VCMTX(0) are the common-mode voltages for static Differential-1 and Differential- 0 states respectively, then the common-mode reference voltage is defined by:

$$V_{CMTX,\,REF} = (V_{\,CMTX(1)} + V_{\,CMTX(0)})/2$$

The transient common-mode voltage variation is defined by:

$$\Delta V_{CMTX\,(t)} = V_{CMTX\,(t)} - V_{\,CMTX\,,REF(t)}$$

The static common-mode voltage mismatch between the Differential-1 and Differential-0 state is given by:

$$\Delta V_{CMTX\,(1,0\,)} = (V_{CMTX\,(1)} - V\,CMTX\,(0))/2$$

A test circuit for the measurement of VOD and VCMTX is shown:

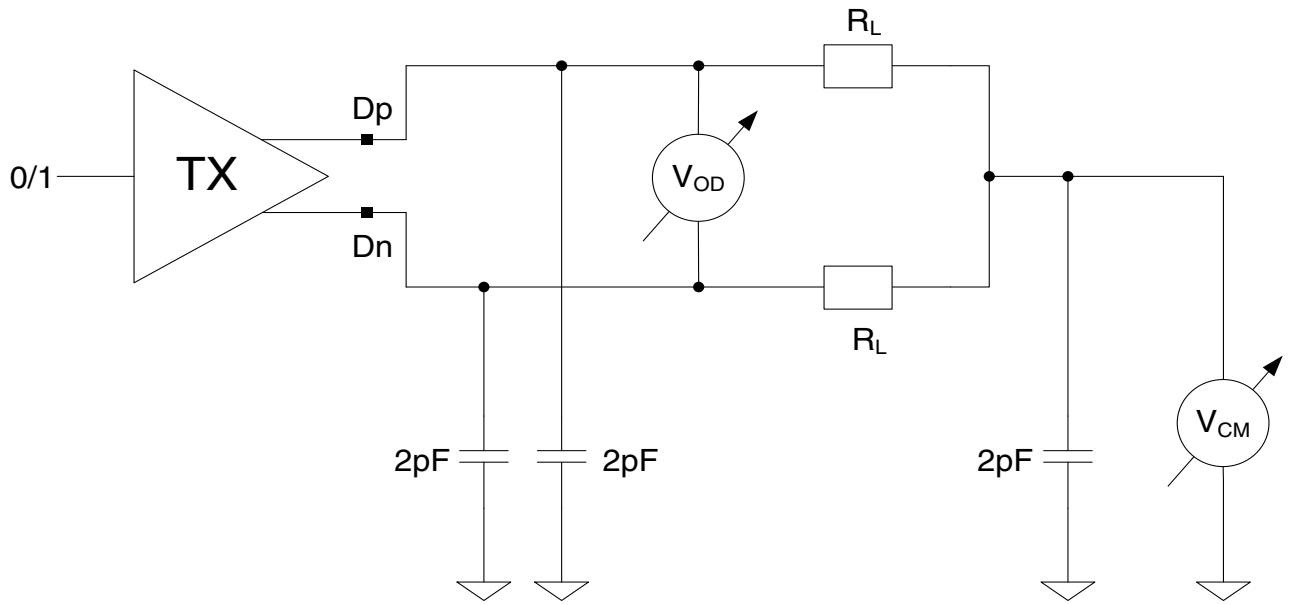


Figure 13-49. Test Circuit for Vcmtx and Vod measurement

The single-ended output impedance of the transmitter at both the Dp and Dn pins is denoted by Z_{OS}. ΔZ_{OS} is the mismatch of the single ended output impedances at the Dp and Dn pins, denoted by ZOSDP and ZOSDN respectively. This mismatch is defined as the ratio of the absolute value of the difference of Z_{OSDP} and Z_{OSDN} and the average of those impedances:

$$\Delta Z_{OS} = (2*|Z_{OSDP} - Z_{OSDN}|)/(Z_{OSDP} + Z_{OSDN})$$

The output impedance ZOS and the output impedance mismatch ΔZOS is compliant with High Speed Transmitter DC Specifications for both the Differential-0 and Differential-1 states for all allowed loading conditions. It is recommended the design keep the output impedance during state transitions as close as possible to the steady state value. The output impedance ZOS can be determined by injecting an AC current into the Dp and Dn pins and measuring the peak-to-peak voltage amplitude. The test circuit for such a measurement is shown in the following figure, where RL are load resistors and VTEST,PP is a test signal applied at the common-mode node via an AC coupling capacitor CAC. The frequency of the test signal is fTEST.

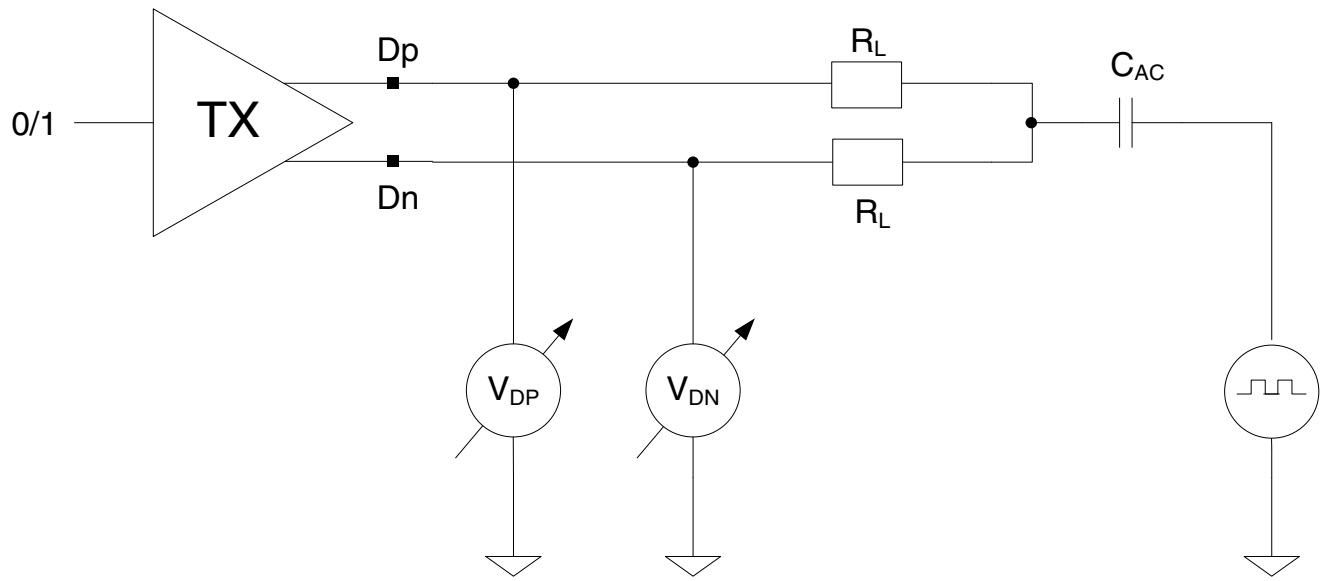


Figure 13-50. Test Circuit for Output Impedance measurements

Assuming negligible effect of the coupling capacitor, the example circuit gives the following relation between peak-to-peak pin voltage VPIN,PP and ZOS. This equation can be used for both Dn and Dp pins individually.

$Z_{OS} = (V_{PIN,PP} * RL)/(V_{TEST,PP} - Z_{PIN,PP})$

Where VPIN,PP is the peak-to-peak voltage at the pin Dp or Dn.

13.6.4.2 High-Speed Receiver (HS-RX)

The HS receiver is a differential line receiver. It contains an on-die switchable parallel input termination, ZID, between the positive input pin Dp and the negative input pin Dn.

The differential input high and low threshold voltages of the HS receiver are denoted by V_{IDTH} and V_{IDTL}, respectively. V_{CMRXDC} is the differential input common mode voltage. The HS receiver is able to detect differential signals at its Dp and Dn input signal pins when both signal voltages, VDP and VDN, are within the common mode voltage range and if the voltage difference of V_{DP} and V_{DN} exceeds either V_{IDTH} or V_{IDTL}.

During operation of the HS receiver, termination impedance is required between the Dx_P and Dx_N pins of the HS receiver. Z_{ID} is disabled when the module is not in the HS receive mode. CCM is the common mode AC termination, which ensures a proper termination of the receiver at higher frequencies.

13.6.4.3 Low-Power Transmitter (LP-TX)

The Low-Power transmitter is a slew-rate controlled push-pull driver. It is used for driving the Lines in all Low-Power operating modes. The LP transmitter has been optimized to minimize static power consumption. The slew-rate of signal transitions is bounded in order to keep EMI low. The LP transmitter will not drive the pad pin potential statically beyond the maximum value of V_{OH}. V_{OH} is the thevenin output, high-level voltage in the high-level state, when the pad pin is not loaded. V_{OL} is the thevenin output, low-level voltage in the LP transmit mode. This is the voltage at an unloaded pad pin in the low-level state.

13.6.4.4 Low-Power Receiver (LP-RX)

The low power receiver is an un-terminated, single-ended receiver circuit. The LP receiver is used to detect the Low-Power state on each pin. For high robustness, the LP receiver filters out noise pulses and RF interference. The LP receiver is optimized for low power operation.

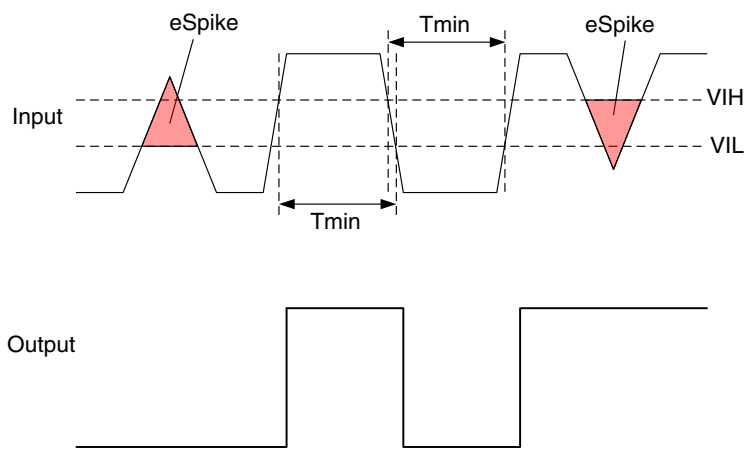


Figure 13-51. Input Glitch Rejection of Low-Power Receiver

The input low-level voltage, VIL, is the voltage at which the receiver is required to detect a low state in the input signal. VIL is larger than the maximum single-ended Line voltage during HS transmission. Therefore, both LP receivers will detect low during HS signaling. The input high-level voltage, VIH, is the voltage at which the receiver is required to detect a high state in the input signal. In order to reduce noise sensitivity on the received signal, the LP receiver incorporates a hysteresis, VHYST.

The LP receiver also rejects any input glitch when the glitch is smaller than eSPIKE. The filter allows pulses wider than TMIN to propagate through the LP receiver. Additionally, the LP receivers are tolerant of super-positioned RF interference on top of the wanted

Line signals. The LP receiver meets all specifications for interference with amplitude VINT and frequency fINT. The interference does not cause glitches or incorrect operation during signal transitions.

13.6.4.5 Low-Power Contention Detector (LP-CD)

Contention can be inferred from any of the following conditions:

- LP high fault is detected when the LP transmitter is driving high and the pin voltage is less than VIL.
- LP low fault is detected when the LP transmitter is driving low and the pad pin voltage is greater than VIHCD. An LP low fault will not be detected when the pin voltage is less than VILCD.

The general operation of a contention detector is similar to that of an LP receiver with lower threshold voltages. Although the DC specifications differ, the AC specifications of the LP-CD are defined to match those of the LP receiver. The LP-CD sufficiently filters the input signal to avoid false triggering on short events

The LP-CD threshold voltages (VILCD, VIHCD) are shown along with the normal signaling voltages:

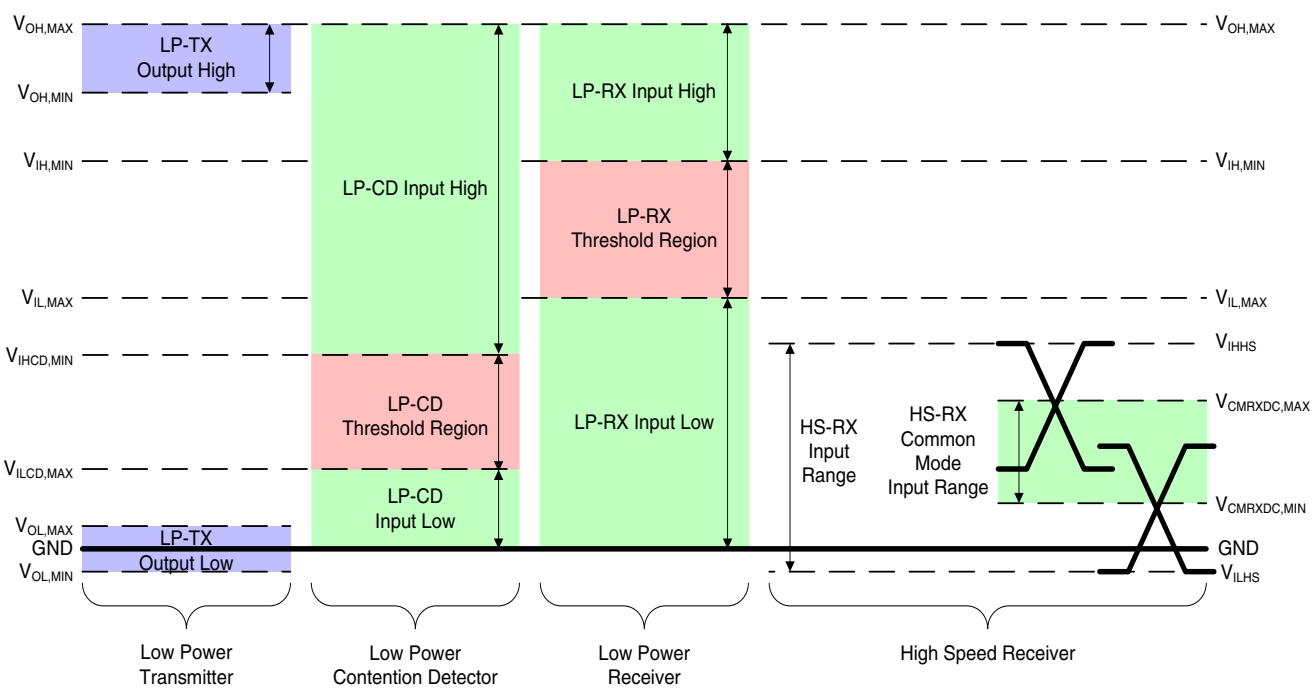


Figure 13-52. Signaling and MIPI Contention Voltage Levels

13.6.4.6 Input Characteristics

No structure within the PHY may be damaged when a DC signal that is within the signal voltage range VPIN is applied to a pad pin for an indefinite period of time. $V_{PIN(absmax)}$ is the maximum transient output voltage at the transmitter pin. The voltage on the transmitter's output pin will not excel $V_{PIN,MAX}$ for a period greater than $T_{VPIN(absmax)}$. When the PHY is in the low-power receive mode the pad pin leakage current will be I_{LEAK} when the pad signal voltage is within the signal voltage range of V_{PIN}. The specification of ILEAK assures interoperability of any PHY in the LP mode by restricting the maximum load current of an LP transmitter. The test circuit for leakage current measurement is shown:

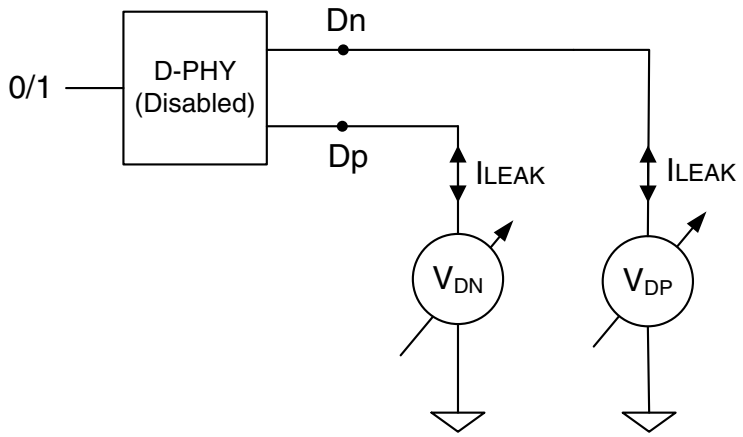


Figure 13-53. Pin-Leakage Measurement DC specification

13.6.4.7 High-Speed Serializer

The HS-Serializer converts 8-bits parallel data into 1 bit data stream to be transmitted by HSTX module. The CIL engine should clock out 8-bits parallel data, HSTX_DATA[7:0], on D0_HS_BYTE_CLKS rising edges. Then the Serializer will capture HSTX_DATA[7:0] on D0_HS_BYTE_CLKS rising edges and the signal connected to HSTX_DATA[0] is transmitted first. There is setup timing constraint and hold timing constraint designated to the rising edge of D0_HS_BYTE_CLKS as constrained in the liberty view provided.

The figure shows the relationship between HS_TXCLKP, HS_TXCLKN, HS_SER_LD, HS_BYTE_CLKS and HSTX_DATA[7:0]. It also shows how the 8-bits of parallel bits are mapped into the serial link:

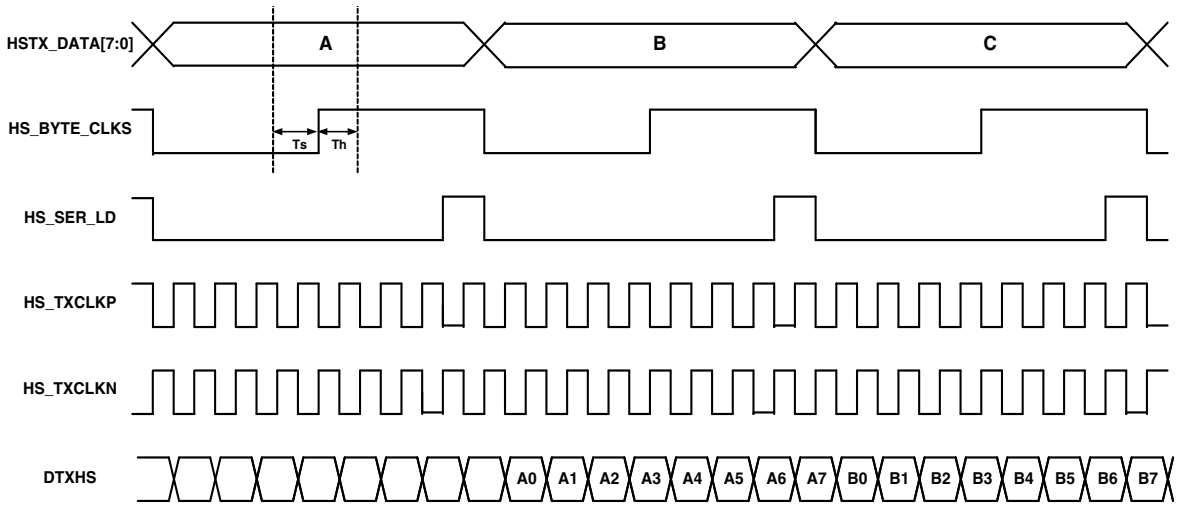


Figure 13-54. Serializer Timing Diagram

13.6.4.8 High-Speed Deserializer

The HS-Deserializer includes a Token Detector to detect SoT leader sequence '011101'. After proper match found (any single bit error allowed) for Sync sequence in HS input data stream, Deserializer then converts 1 bit data stream received by HS-RX module into 8-bits parallel data and HS_BYTE_CLKD becomes active.

The High speed input data stream from HS-RX module, DRXHSP/DRXHSN, is sampled by a differential DDR (half-rate) clock, HS_RXCLKP/HS_RXCLKN. The following bits after the Sync leader sequence are payload bits. The signal connected to HSRX_DATA[0] was received first and HSRX_DATA[7:0] can be captured on rising edges of HS_BYTE_CLKD. The transmitter will ensure that a rising edge of the DDR clock is sent during the first payload bit of a transmission burst such that the first payload bit can be sampled by the receiver on the rising clock edge, the second bit can be sampled on the falling edge, and all following bits can be sampled on alternating rising and falling edges.

HS-Deserializer also provides 3 synchronization condition signals. If sync sequence is properly matched, "SYNC" signal will be asserted. If sync sequence is matched with single bit error, "ERRSYNC" signal will be asserted. If the leader sequence is corrupted in a way that proper synchronization cannot be achieved (more than single-bit error), "NOSYNC" signal will be asserted.

The figures show the relationship between HS_RXCLKP/HS_RXCLKN, DRXHSP/DRXHSN, DRXLPP/DRXLPN, SYNC/ERRSYNC/NOSYNC, HS_BYTE_CLKD and HSRX_DATA[7:0]:

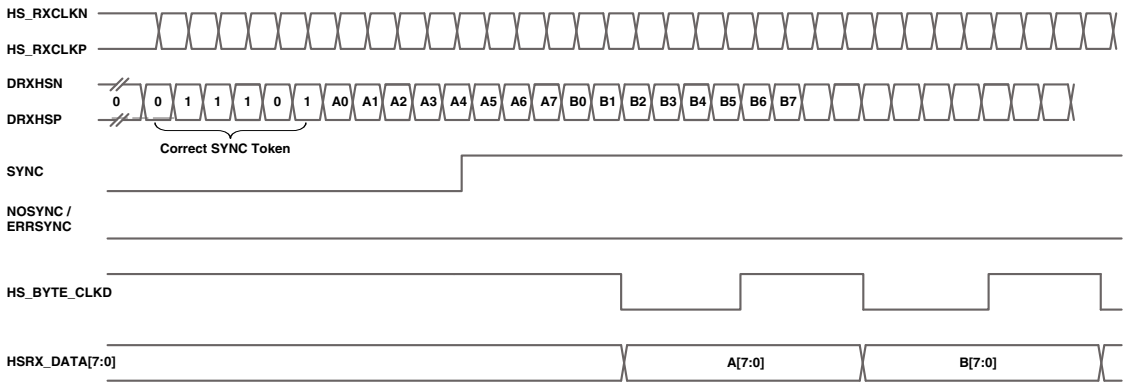


Figure 13-55. Deserializer Timing Diagram with Correct Sync Token

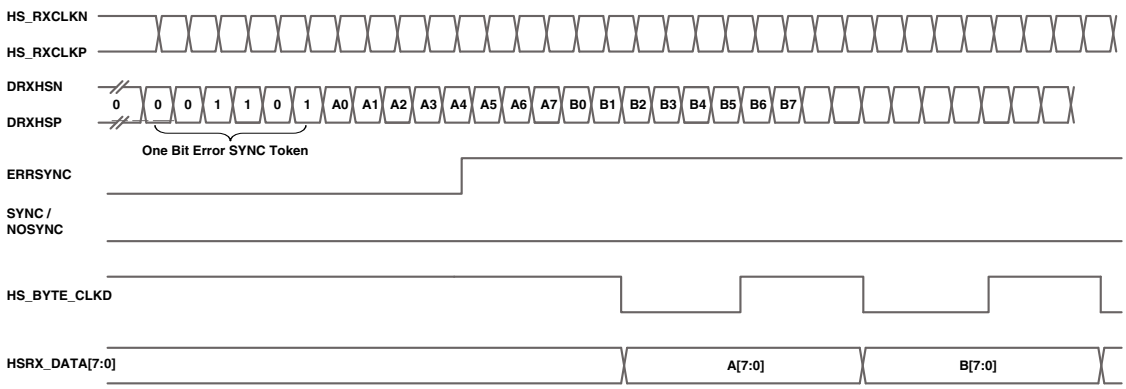


Figure 13-56. Deserializer Timing Diagram with One-Bit-Error Sync Token

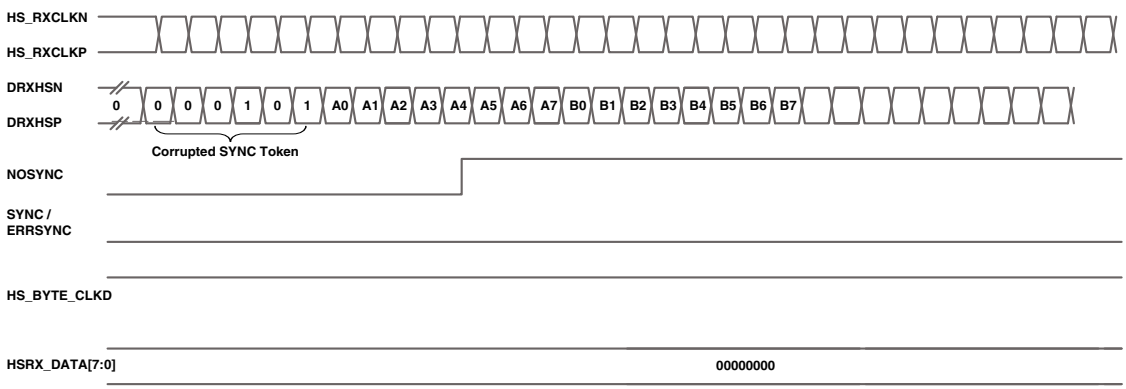


Figure 13-57. Deserializer Timing Diagram with Corrupted Sync Token

13.6.4.9 Termination Calibrator

An optional calibration circuit is implemented inside D-PHY to ensure that the HS-TX and HS-RX meet the required specifications with process corner and temperature variations. A precise external resistor is used to calibrate a scaled replica of the termination resistor, inside the calibrator. The auto calibration procedure is performed every time the D-PHY is powered-up.

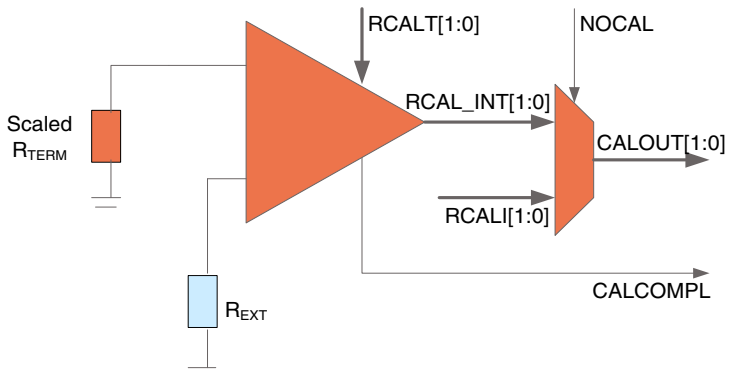


Figure 13-58. Calibrator Block Diagram

The D-PHY supports one of the following receiver termination schemes:

- **Auto Calibration mode:**Termination resistors are automatically configured by calibration circuit with reference resistor connected between REXT and VSSA.
- **User programmable mode:**RCALI[1:0] is programmed by the user through chip registers. In this case, the user has to control NOCAL signal for manual calibration.

13.6.4.10 DSI D-PHY Signals

This section details the interface signals for MIPI DSI D-PHY. For detailed signal spec, please refer to the chip datasheet.

Table 13-29. DSI DPHY PPI Signals

Signal	Type	Description
TxClkEsc	Clock	Escape mode Transmit Clock. This clock is directly used to generate escape sequences. The period of this clock determines the symbol time for low power signals. It is therefore constrained by the normative part of the DPHY specification. **NOTE:** The max frequency of TxClkEsc is 20 MHz **NOTE:** The Min frequency of TxClkEsc is 12 MHz. TxClkEsc is synchronous with RxClkInEsc.

Table continues on the next page...

Table 13-29. DSI DPHY PPI Signals (continued)

Signal	Type	Description
RxClkInEsc_ln0	Clock	Escape mode Clk for RX. Minimum Clk frequency should be 60 MHz in order to correctly detect the LP states, which are minimum 50 ns long but the minimum LP pulse duration could be as low as 20 ns as per D-PHY CTS. RxClkInEsc is synchronous with TxClkEsc
TxByteClkHS	Clock	High-Speed Transmit Byte Clock. This is used to synchronize PPI signals in the High-Speed transmit clock domain. It is recommended that all transmitting Data Lane Modules share on TxByteClkHS signal. The frequency of TxByteClkHS is exactly 1/8 the High-Speed bit rate.

13.6.4.11 Operating Modes

13.6.4.11.1 High-Speed Transmit from the Master Side

The figure shows an example of a High-Speed transmission on the Master side. While TxRequestHS is low, the Lane Module ignores the value of TxDataHS. To begin transmission, the protocol drives TxDataHS with the first byte of data and asserts TxRequestHS. This data byte is accepted by the PHY on the first rising edge of TxByteClkHS with TxReadyHS also asserted. At this point, the protocol logic drives the next data byte onto TxDataHS. After every rising clock cycle with TxReadyHS active, the protocol supplies a new valid data byte or ends the transmission. After the last data byte has been transferred to the Lane Module, TxRequestHS is driven low to cause the Lane Module to stop the transmission and enter Stop state. The minimum number of bytes transmitted could be as small as one.

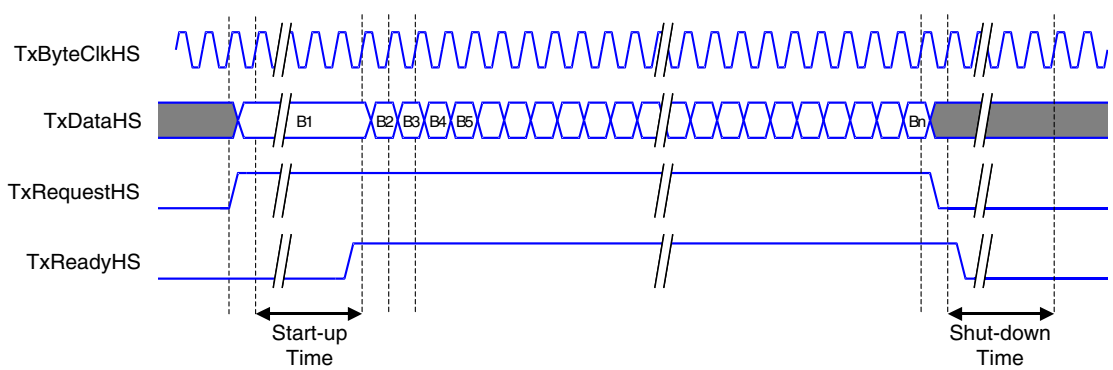


Figure 13-59. High-Speed Transmission from the Master Side

13.6.4.11.2 Low-Power Data Transmission

For Low-Power data transmission the TxClkEsc is used. The Protocol directs the Data Lane to enter Low-Power data transmission Escape mode by asserting TxRequestEsc with TxLpdtEsc high. The Low-Power transmit data is transferred on the TxDataEsc lines when TxValidEsc and TxReadyEsc are both active at a rising edge of TxClkEsc. The byte is transmitted in the time after the TxDataEsc is accepted by the Lane Module (TxValidEsc = TxReadyEsc = high). The Protocol knows the byte transmission is finished when TxReadyEsc is asserted. After the last byte has been transmitted, the protocol de-asserts TxRequestEsc to end the Low-Power data transmission. This causes TxReadyEsc to return low. Whenever TxRequestEsc transitions from high-to-low, it always remains in the low state for a minimum of two TxClkEsc clock cycles. The figure shows an example Low-Power data transmission operation.

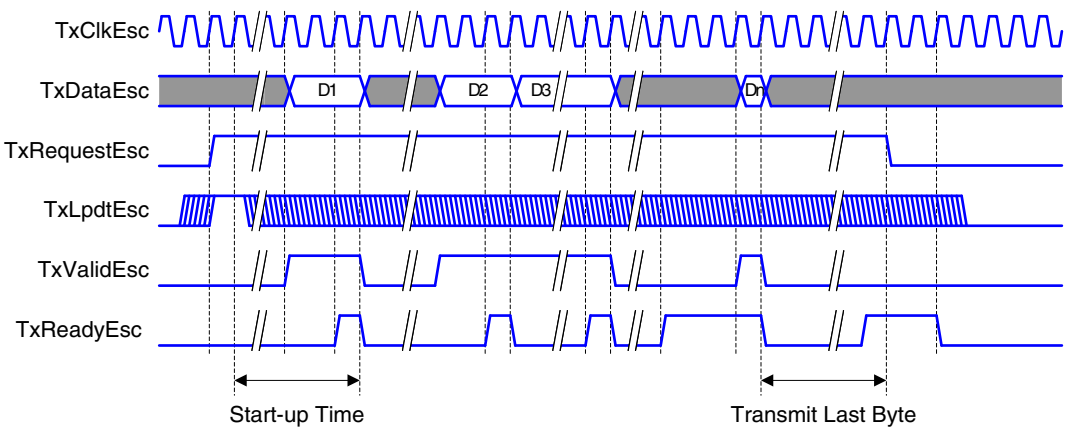


Figure 13-60. Low-Power Data Transmission

13.6.4.11.3 Low-Power Data Reception

The figure shows an example Low-Power data reception. In this example, a Low-Power escape “clock” is generated from the Lane Interconnect by the logical exclusive-OR of the Dp and Dn lines. This “clock” is used within the Lane Module to capture the transmitted data. In this example, the “clock” is also used to generate RxClkEsc. The signal RxLpdtEsc is asserted when the escape entry command is detected and stays high until the Lane returns to Stop state, indicating that the transmission has finished. It is important to note that because of the asynchronous nature of Escape mode transmission, the RxClkEsc signal can stop at anytime in either the high or low state. This is most likely to happen just after a byte has been received, but it could happen at other times as well

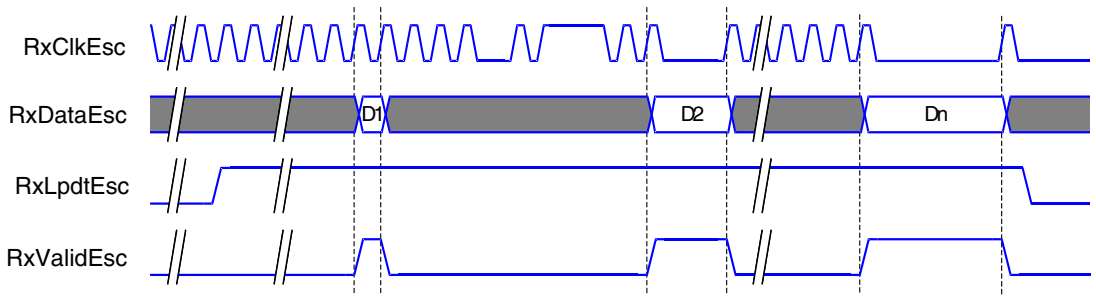


Figure 13-61. Low-Power Data Reception

13.6.4.11.4 Turn-around

If the Master side and Slave side Lane Modules are both bi-directional, it is possible to turn around the Link for Escape mode signaling. Which side is allowed to transmit is determined by passing a “token” back and forth. That is, the side currently transmitting passes the token to the receiving side. If the receiving side acknowledges the turn-around request, as indicated by driving the appropriate line state, the direction is switched. The figure shows an example of two turn-around events. At the beginning, the local side is the transmitter, as shown by Direction=0. When the protocol on this side wishes to turn the Lane around (i.e. give the token to the other side), it asserts TurnRequest for at least one cycle of TxClkEsc. This initiates the turn around procedure. The remote side acknowledges the turn-around request by driving the appropriate states on the Lines. When this happens, the local Direction signal changes from transmit (0) to receive (1). The remote side initiates a turn-around request, passing the token back to the local side. When this happens, the local Direction signal changes back to transmit (0). The transmitter is in control of the Link direction and decides when to turn the Link around, passing control to the receiver. If the remote side does not acknowledge the turnaround request, the Direction signal does not change.

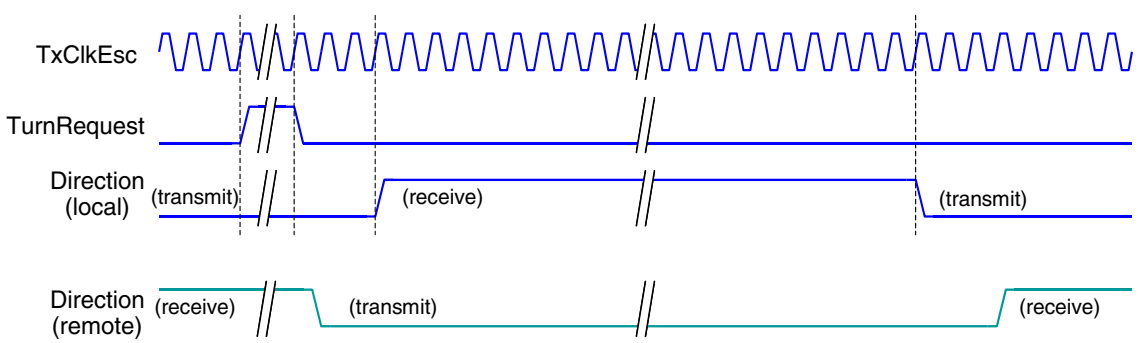


Figure 13-62. Turn-around Actions Transmit-to-Receive and Back to Transmit

13.6.5 DPHY PLL

The MIPI DSI DPHY PLL is a high performance PLL based frequency synthesizer that incorporates a lock detector, independent output divider, and supports power down modes. The PLL block diagram is shown below.

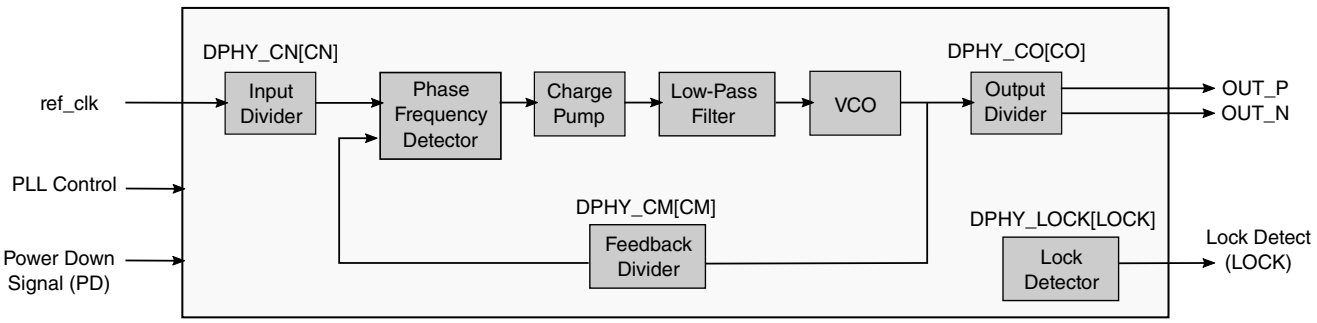


Figure 13-63. MIPI DSI DPHY PLL Block Diagram

The DPHY PLL input clock is ref_clk. The PLL output multiplies the input frequency by `(CM / (CN * CO))`.

NOTE

The input frequency ranges from 24 MHz till 200 MHz. The input divider has to be programmed such that the frequency after the input divider ranges from 24 MHz till 30 MHz.

NOTE

The VCO maximum output frequency is 1.5 GHz.

The power down signal (PD) is active-high and used to power down the PLL. Asserting PD will reset the PLL to it's initial state. To ensure proper PLL functionality, ref_clk needs to be stable before PLL power up. The expected power-on sequence for the DSI DPHY PLL is illustrated below.

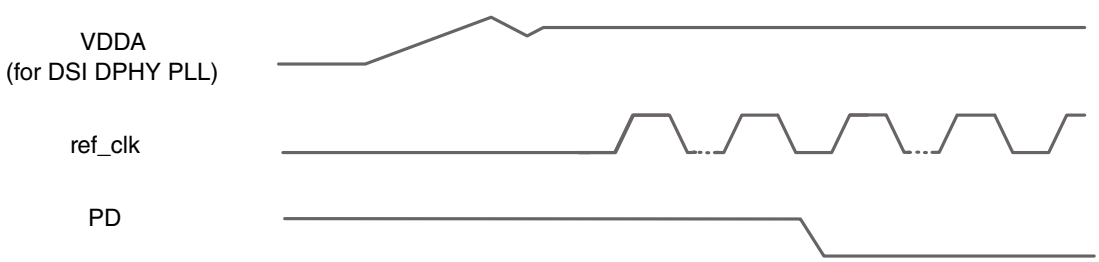


Figure 13-64. MIPI DSI DPHY PLL Power-on Sequence

13.6.6 DPHY Programming

The following sections detail the programming of the MIPI DSI DPHY high-speed timers.

13.6.6.1 High-Speed Prepare Timer ($T_{HS-PREPARE}$)

Calculate the value for clock lane high-speed prepare timer (mc_PRG_HS_PREPARE) using the formula below:

Assuming TxClkEsc = 20 MHz (T_{per} = 50 ns)
When mc_PRG_HS_PREPARE = 0, $T_{CLK-PREPARE}$= 1 * TxClkEsc Period = 50 ns
When mc_PRG_HS_PREPARE = 1, $T_{CLK-PREPARE}$ = 1.5 * TxClkEsc Period = 75 ns

Calculate the value for data lane high-speed prepare timer (m_PRG_HS_PREPARE) using the formula below:

Assuming TxClkEsc (TLPX) = 20 MHz (T_{per} = 50 ns)
When m_PRG_HS_PREPARE = 00, $T_{HS-PREPARE}$ = 1 * TxClkEsc Period = 50 ns
When m_PRG_HS_PREPARE = 01, $T_{HS-PREPARE}$ = 1.5 * TxClkEsc Period = 75 ns
When m_PRG_HS_PREPARE = 10, $T_{HS-PREPARE}$ = 2 * TxClkEsc Period = 100 ns
When m_PRG_HS_PREPARE = 11, $T_{HS-PREPARE}$ = 2.5 * TxClkEsc Period = 125 ns

13.6.6.2 High-Speed Zero Timer($T_{HS-ZERO}$)

Calculate the value for clock lane high-speed zero timer (mc_PRG_HS_ZERO) using the formula below:

$T_{CLK-ZERO}$= (mc_PRG_HS_ZERO + 3)*(TxByteClkHS Period)
TxByteClkHS = 1/8 Data Rate
When mc_PRG_HS_ZERO = 000110 @ 250 Mb/s, $T_{CLK-ZERO}$ = (6+3) * 32 ns = 288 ns
When mc_PRG_HS_ZERO = 100000 @ 1 Gb/s, $T_{CLK-ZERO}$ = (32+3) * 8 ns = 280 ns

Calculate the value for data lane high-speed zero timer (m_PRG_HS_ZERO) using the formula below:

$T_{HS-ZERO}$ = (m_PRG_HS_ZERO + 6)*(TxByteClkHS Period)
TxByteClkHS = 1/8 Data Rate
When m_PRG_HS_ZERO = 00001 @ 250 Mb/s, $T_{HS-ZERO}$ = (1+6) * 32 ns = 224 ns
When m_PRG_HS_ZERO = 01001 @ 1 Gb/s, $T_{HS-ZERO}$ = (9+6) * 8 ns = 120 ns

13.6.6.3 High-Speed Trail Timer ($T_{HS-TRAIL}$)

Calculate the value for the high-speed trail timer (PRG_HS_TRAIL) using the formula below:

$T_{HS-TRAIL}$= (PRG_HS_TRAIL)*(TxByteClkHS Period)
TxByteClkHS = 1/8 Data Rate
When PRG_HS_TRAIL = 0100 @ 250 Mb/s, >$T_{HS-TRAIL}$ = 4 * 32 ns = 128 ns
When PRG_HS_TRAIL = 1100 @ 1 Gb/s, >$T_{HS-TRAIL}$ = 12 * 8 ns = 96 ns

13.6.7 Packet Data and Pixel Formats

The DSI Host Controller Core and the external DSI Peripheral explicitly handle and exchanges 32-bit words encapsulated as packet data. The mapping between pixel formats and pixel data must be done on both sides by user application on both sides of the DSI link. The DSI spec gives many examples of common pixel formats. The DSI Host Controller Core and external DSI Peripheral have no preference or limitation on which pixel formats it will handle. The IP which provides video source packet data through DPI-2 interface (e.g. LCDIF) onto DSI should also be configured by the user application to match the pixel formats expected by the DSI Host Controller.

The DPI-2/DBI-2 interface core handles pixel to byte packing and byte to pixel unpacking according to the MIPI DSI specification.

13.6.8 MIPI DSI Host Memory Map/Register Definition

MIPI_DSI_HOST memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0000	MIPI_DSI_HOST_DSI_HOST_CFG_NUM_LANES	32	R	0000_0000h	13.6.8.1/ 4903
30A0_0004	MIPI_DSI_HOST_DSI_HOST_CFG_NONCONTINUOUS_CLK	32	R	0000_0000h	13.6.8.2/ 4904
30A0_0008	MIPI_DSI_HOST_DSI_HOST_CFG_T_PRE	32	R	0000_0000h	13.6.8.3/ 4905
30A0_000C	MIPI_DSI_HOST_DSI_HOST_CFG_T_POST	32	R	0000_0000h	13.6.8.4/ 4905
30A0_0010	MIPI_DSI_HOST_DSI_HOST_CFG_TX_GAP	32	R	0000_0000h	13.6.8.5/ 4906

Table continues on the next page...

MIPI_DSI_HOST memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0014	MIPI_DSI_HOST_DSI_HOST_CFG_AUTOINSERT_EOTP	32	R	0000_0000h	13.6.8.6/ 4907
30A0_0018	MIPI_DSI_HOST_DSI_HOST_CFG_EXTRA_CMDS_AFTER_EOTP	32	R	0000_0000h	13.6.8.7/ 4907
30A0_001C	MIPI_DSI_HOST_DSI_HOST_CFG_HTX_TO_COUNT	32	R	0000_0000h	13.6.8.8/ 4908
30A0_0020	MIPI_DSI_HOST_DSI_HOST_CFG_LRX_H_TO_COUNT	32	R	0000_0000h	13.6.8.9/ 4908
30A0_0024	MIPI_DSI_HOST_DSI_HOST_CFG_BTA_H_TO_COUNT	32	R	0000_0000h	13.6.8.10/ 4909
30A0_0028	MIPI_DSI_HOST_DSI_HOST_CFG_TWAKEUP	32	R	0000_0000h	13.6.8.11/ 4909
30A0_002C	MIPI_DSI_HOST_DSI_HOST_CFG_STATUS_OUT	32	R	0000_0000h	13.6.8.12/ 4910
30A0_0030	MIPI_DSI_HOST_DSI_HOST_RX_ERROR_STATUS	32	R	0000_0000h	13.6.8.13/ 4912

13.6.8.1 MIPI_DSI_HOST_DSI_HOST_CFG_NUM_LANES

Address: 30A0_0000h base + 0h offset = 30A0_0000h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

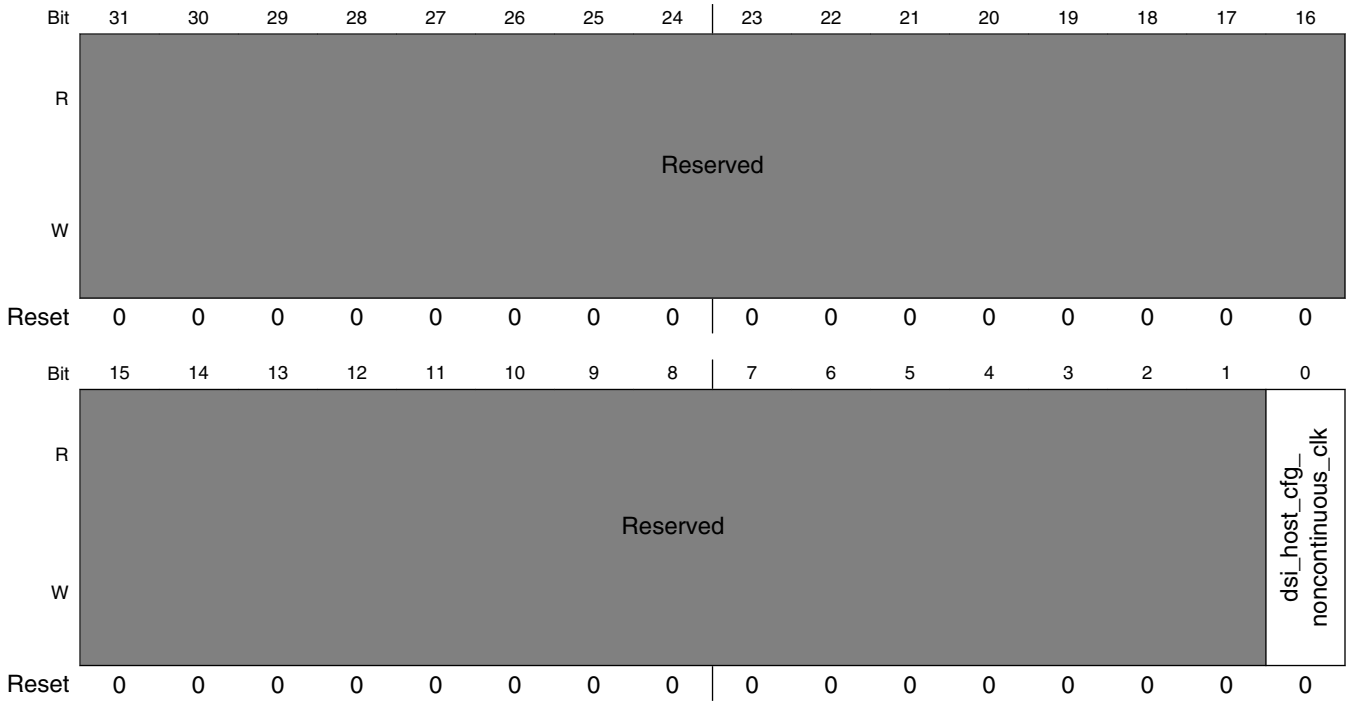
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dsi_host_cfg_num_lanes	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_NUM_LANES field descriptions

Field	Description
31–2 -	This field is reserved.
dsi_host_cfg_num_lanes	Sets the number of active lanes that are to be used for transmitting data. • 2'b00 - 1 Lane • 2'b01 - 2 Lanes • 2'b10 - 3 Lanes • 2'b11 - 4 Lanes

13.6.8.2 MIPI_DSI_HOST_DSI_HOST_CFG_NONCONTINUOUS_CLK

Address: 30A0_0000h base + 4h offset = 30A0_0004h



MIPI_DSI_HOST_DSI_HOST_CFG_NONCONTINUOUS_CLK field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_ noncontinuous_clk	Sets the Host Controller into non-continuous MIPI clock mode. When in non-continuous clock mode, the high speed clock will transistion into low power mode between transmissions. • 1'b0 - Continuous high speed clock • 1'b1 - Non-Continuous high speed clock

13.6.8.3 MIPI_DSI_HOST_DSI_HOST_CFG_T_PRE

NOTE

Please refer to the MIPI Alliance Specification for D-PHY (Global Operation Timing Parameters) for details on calculating the parameter described below

Address: 30A0_0000h base + 8h offset = 30A0_0008h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																									dsi_host_cfg_t_pre							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_T_PRE field descriptions

Field	Description
31–7 -	This field is reserved.
dsi_host_cfg_t_pre	Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after enabling the clock lane for HS operation before enabling the data lanes for HS operation. This setting represents the TCLK-PRE parameter. The minimum value for this port is 1.

13.6.8.4 MIPI_DSI_HOST_DSI_HOST_CFG_T_POST

NOTE

Please refer to the MIPI Alliance Specification for D-PHY (Global Operation Timing Parameters) for details on calculating the parameter described below

Address: 30A0_0000h base + Ch offset = 30A0_000Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																									dsi_host_cfg_t_post							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_T_POST field descriptions

Field	Description
31–7 -	This field is reserved.

Table continues on the next page...

MIPI_DSI_HOST_DSI_HOST_CFG_T_POST field descriptions (continued)

Field	Description
dsi_host_cfg_t_post	Sets the number of byte clock periods ('clk_byte' input) to wait before putting the clock lane into LP mode after the data lanes have been detected to be in Stop State. This setting represents the DPHY timing parameters TLPX (TxClkEsc) + TCLK-PREPARE + TCLK-ZERO + TCLK-PRE requirement for the clock lane before the data lane is allowed to change from LP11 to start a high speed transmission. The minimum value for this port is 1.

13.6.8.5 MIPI_DSI_HOST_DSI_HOST_CFG_TX_GAP

NOTE

Please refer to the MIPI Alliance Specification for D-PHY (Global Operation Timing Parameters) for details on calculating the parameter described below

Address: 30A0_0000h base + 10h offset = 30A0_0010h

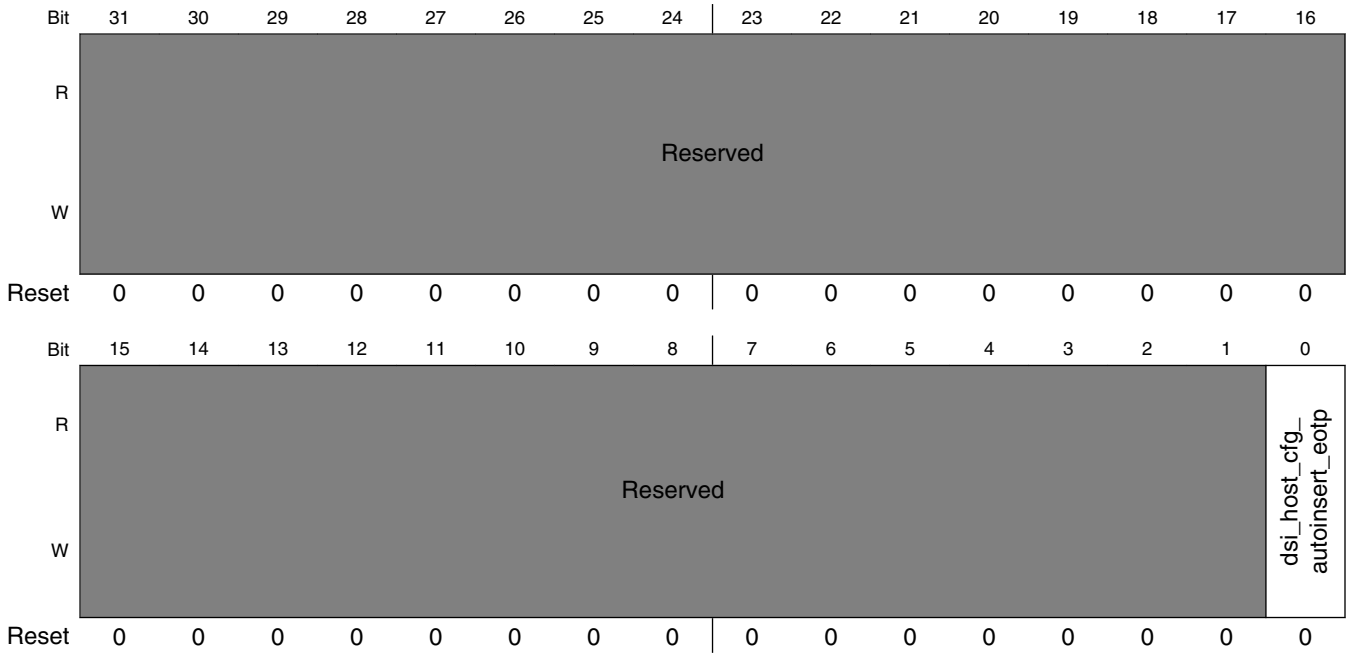
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																									dsi_host_cfg_tx_gap							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_TX_GAP field descriptions

Field	Description
31–7 -	This field is reserved.
dsi_host_cfg_tx_gap	Sets the number of byte clock periods ('clk_byte' input) that the controller will wait after the clock lane has been put into LP mode before enabling the clock lane for HS mode again. This setting represents the THS-EXIT parameter. The minimum value for this port is 1.

13.6.8.6 MIPI_DSI_HOST_DSI_HOST_CFG_AUTOINSERT_EOTP

Address: 30A0_0000h base + 14h offset = 30A0_0014h

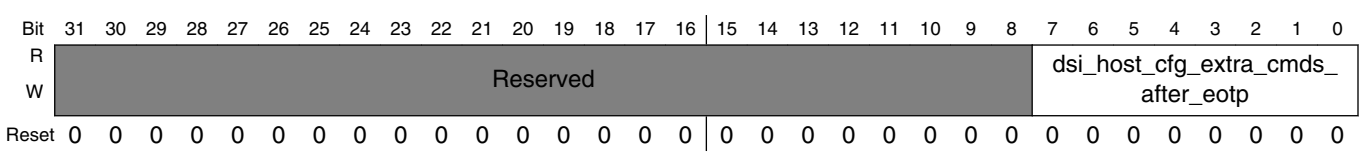


MIPI_DSI_HOST_DSI_HOST_CFG_AUTOINSERT_EOTP field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_autoinsert_eotp	Enables the Host Controller to automatically insert an EoTp short packet when switching from HS to LP mode. • 1'b0 - EoTp is not automatically inserted • 1'b1 - EoTp is automatically inserted

13.6.8.7 MIPI_DSI_HOST_DSI_HOST_CFG_EXTRA_CMDS_AFTER_EOTP

Address: 30A0_0000h base + 18h offset = 30A0_0018h



MIPI_DSI_HOST_DSI_HOST_CFG_EXTRA_CMDS_AFTER_EOTP field descriptions

Field	Description
31–8 -	This field is reserved.
dsi_host_cfg_ extra_cmds_ after_eotp	Configures the DSI Host Controller to send extra End Of Transmission Packets after the end of a packet. The value is the number of extra EOTP packets sent.

13.6.8.8 MIPI_DSI_HOST_DSI_HOST_CFG_HTX_TO_COUNT

Address: 30A0_0000h base + 1Ch offset = 30A0_001Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								dsi_host_cfg_htx_to_count																								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_HTX_TO_COUNT field descriptions

Field	Description
31–24 -	This field is reserved.
dsi_host_cfg_ htx_to_count	Sets the value of the DSI Host High Speed TX timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification. This timeout parameter should be configured to represent the time taken to transmit the biggest HS data payload. If this timeout is reached the DSI byte count is cleared and the HS transmission is aborted. This timer can be also disabled, when set to 0.

13.6.8.9 MIPI_DSI_HOST_DSI_HOST_CFG_LRX_H_TO_COUNT

Address: 30A0_0000h base + 20h offset = 30A0_0020h

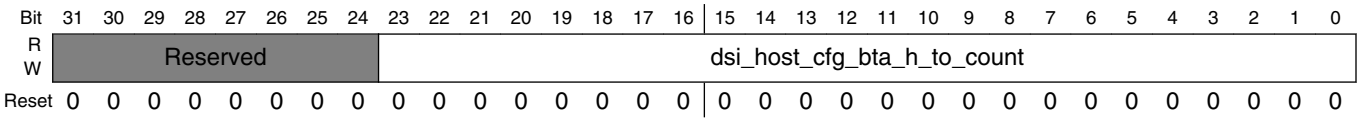
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								dsi_host_cfg_lrx_h_to_count																								
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DSI_HOST_CFG_LRX_H_TO_COUNT field descriptions

Field	Description
31–24 -	This field is reserved.
dsi_host_cfg_lrx_ h_to_count	Sets the value of the DSI Host low power RX timeout count in clk_byte clock periods that once reached will initiate a timeout error and follow the recovery procedure documented in the DSI specification. This timeout parameter should be configured to represent the time taken to receive the biggest LP (Escape mode) data payload. If this timeout is reached, the DSI byte count is cleared and the LP reception is aborted. This timer can be also disabled, when set to 0

13.6.8.10 MIPI_DSI_HOST_DSI_HOST_CFG_BTA_H_TO_COUNT

Address: 30A0_0000h base + 24h offset = 30A0_0024h

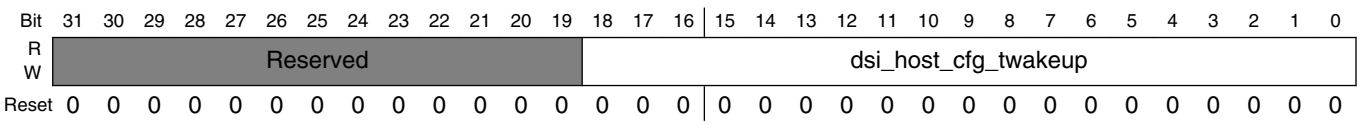


MIPI_DSI_HOST_DSI_HOST_CFG_BTA_H_TO_COUNT field descriptions

Field	Description
31–24 -	This field is reserved.
dsi_host_cfg_ bta_h_to_count	Sets the value of the DSI Host Bus Turn Around (BTA) timeout in clk_byte clock periods that once reached will initiate a timeout error.

13.6.8.11 MIPI_DSI_HOST_DSI_HOST_CFG_TWAKEUP

Address: 30A0_0000h base + 28h offset = 30A0_0028h

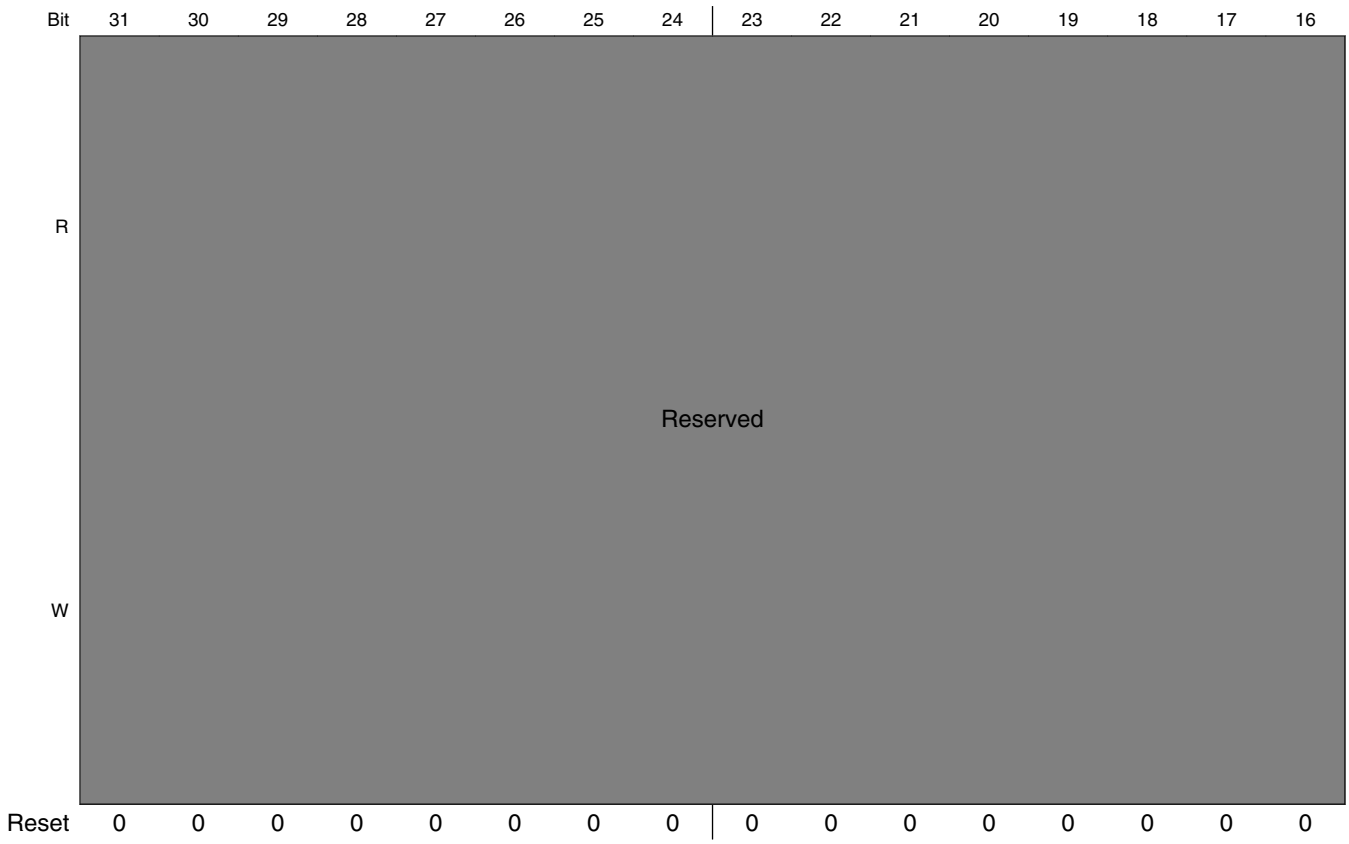


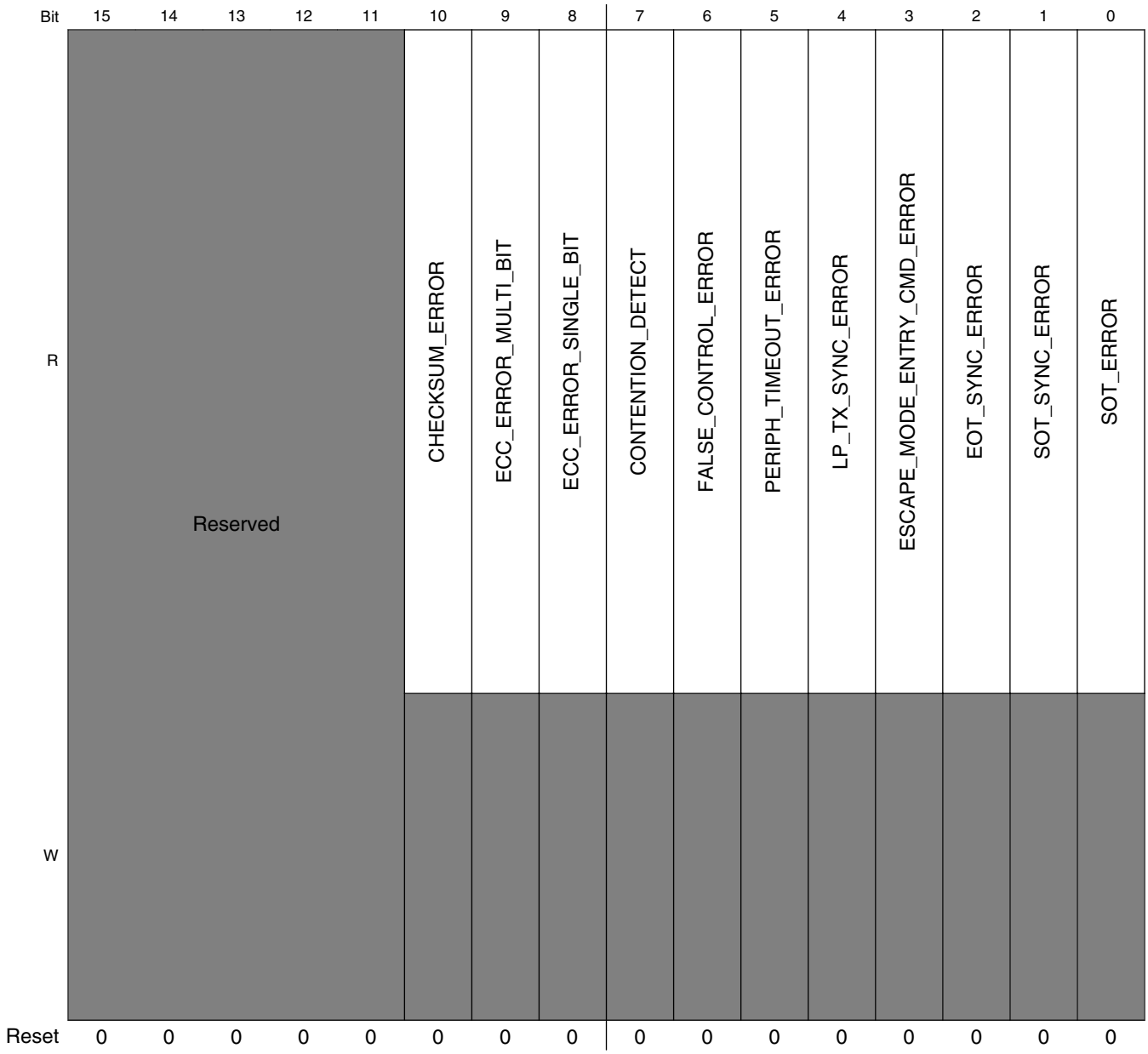
MIPI_DSI_HOST_DSI_HOST_CFG_TWAKEUP field descriptions

Field	Description
31–19 -	This field is reserved.
dsi_host_cfg_ twakeup	DPHY Twakeup timing parameter. Sets the number of clk_esc clock periods to keep a clock or data lane in Mark-1 state after exiting ULPS. The MIPI DPHY spec requires a minimum of 1ms in Mark-1 state after leaving ULPS.

13.6.8.12 MIPI_DSI_HOST_DSI_HOST_CFG_STATUS_OUT

Address: 30A0_0000h base + 2Ch offset = 30A0_002Ch





MIPI_DSI_HOST_DSI_HOST_CFG_STATUS_OUT field descriptions

Field	Description
31–11 -	This field is reserved. Reserved
10 CHECKSUM_ ERROR	Checksum Error (long packet only) – Checksum error from peripheral error report, cleared upon read
9 ECC_ERROR_ MULTI_BIT	ECC Error, multi-bit (detected, not corrected) – ECC multi-bit error from peripheral error report, cleared upon read

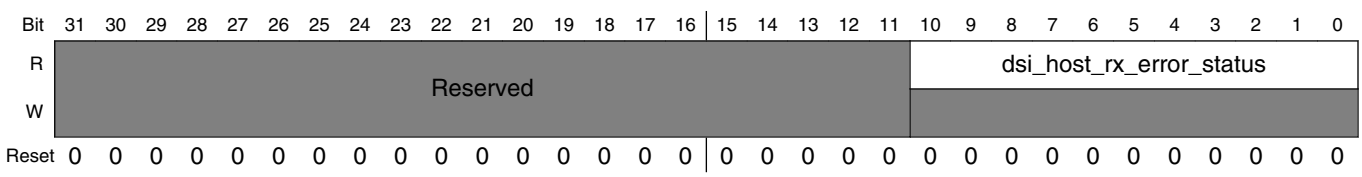
Table continues on the next page...

MIPI_DSI_HOST_DSI_HOST_CFG_STATUS_OUT field descriptions (continued)

Field	Description
8 ECC_ERROR_ SINGLE_BIT	ECC single bit error from peripheral error report, cleared upon read.
7 CONTENTION_ DETECT	Contention Detected – Contention Detection from peripheral error report, cleared upon read
6 FALSE_ CONTROL_ ERROR	False Control Error – False Control Error from peripheral error report, cleared upon read
5 PERIPH_ TIMEOUT_ ERROR	Peripheral Timeout Error – Peripheral Timeout error from peripheral error report, cleared upon read
4 LP_TX_SYNC_ ERROR	Low-Power Transmit Sync Error – Low Power Transmit Sync error from peripheral error report, cleared upon read.
3 ESCAPE_ MODE_ENTRY_ CMD_ERROR	Escape Mode Entry Command Error – Escape Mode Entry Command Error from peripheral error report, cleared upon read
2 EOT_SYNC_ ERROR	EoT Sync Error – End of Transmission (EoT) Sync Error from peripheral error report, cleared upon read
1 SOT_SYNC_ ERROR	SoT Sync Error – Start of Transmission (SoT) Sync Error from peripheral error report, cleared upon read
0 SOT_ERROR	SoT Error – Start of Transmission (SoT) Error from peripheral error report, cleared upon read.

13.6.8.13 MIPI_DSI_HOST_DSI_HOST_RX_ERROR_STATUS

Address: 30A0_0000h base + 30h offset = 30A0_0030h



MIPI_DSI_HOST_DSI_HOST_RX_ERROR_STATUS field descriptions

Field	Description
31–11 -	This field is reserved.
dsi_host_rx_ error_status	Status Register for Host receive error detection, ECC errors, CRC errors and for timeout indicators

Table continues on the next page...

MIPI_DSI_HOST_DSI_HOST_RX_ERROR_STATUS field descriptions (continued)

Field	Description
	• [0] ECC single bit error detected • [1] ECC multi bit error detected • [6:2] Errored bit position for single bit ECC error • [7] CRC error detected • [8] High Speed forward TX timeout detected • [9] Reverse Low power data receive timeout detected • [10] BTA timeout detected

13.6.9 MIPI DSI HOST DPI INTFC Memory Map/Register Definition

MIPI_DSI_HOST_DPI_INTFC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0200	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_PAYLOAD_SIZE	32	R	0000_0000h	13.6.9.1/ 4914
30A0_0204	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FIFO_SEND_LEVEL	32	R	0000_0000h	13.6.9.2/ 4914
30A0_0208	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_INTERFACE_COLOR_CODING	32	R	0000_0000h	13.6.9.3/ 4915
30A0_020C	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FORMAT	32	R	0000_0000h	13.6.9.4/ 4915
30A0_0210	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VSYNC_POLARITY	32	R	0000_0000h	13.6.9.5/ 4916
30A0_0214	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSYNC_POLARITY	32	R	0000_0000h	13.6.9.6/ 4917
30A0_0218	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VIDEO_MODE	32	R	0000_0000h	13.6.9.7/ 4918
30A0_021C	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HFP	32	R	0000_0000h	13.6.9.8/ 4918
30A0_0220	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HBP	32	R	0000_0000h	13.6.9.9/ 4918
30A0_0224	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSA	32	R	0000_0000h	13.6.9.10/ 4919
30A0_0228	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_ENABLE_MULT_PKTS	32	R	0000_0000h	13.6.9.11/ 4920
30A0_022C	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VBP	32	R	0000_0000h	13.6.9.12/ 4920
30A0_0230	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VFP	32	R	0000_0000h	13.6.9.13/ 4921
30A0_0234	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_BLLP_MODE	32	R	0000_0000h	13.6.9.14/ 4921
30A0_0238	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_USE_NULL_PKT_BLLP	32	R	0000_0000h	13.6.9.15/ 4922

Table continues on the next page...

MIPI_DSI_HOST_DPI_INTFC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_023C	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VACTIVE	32	R	0000_0000h	13.6.9.16/ 4923
30A0_0240	MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VC	32	R	0000_0000h	13.6.9.17/ 4923

13.6.9.1 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_PAYLOAD

Address: 30A0_0000h base + 200h offset = 30A0_0200h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																dsi_host_cfg_dpi_pixel_payload_size															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_PAYLOAD_SIZE field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_cfg_dpi_pixel_payload_size	Maximum number of pixels that should be sent as one DSI packet. Recommended that the line size (in pixels) is evenly divisible by this parameter (packet payload size in pixels).

13.6.9.2 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FIFO_SEN

Address: 30A0_0000h base + 204h offset = 30A0_0204h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																dsi_host_cfg_dpi_pixel_fifo_send_level															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FIFO_SEND_LEVEL field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_cfg_dpi_pixel_fifo_send_level	In order to optimize DSI utility, the DPI bridge buffers a cerntain number of DPI pixels before initiating a DSI packet. This configuration port controls the level at which the DPI Host bridge begins sending pixels.

13.6.9.3 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_INTERFACE_COL

Address: 30A0_0000h base + 208h offset = 30A0_0208h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved													dsi_host_cfg_dpi_interface_color_coding		
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_INTERFACE_COLOR_CODING field descriptions

Field	Description
31–3 -	This field is reserved.
dsi_host_cfg_dpi_interface_color_coding	Sets the distribution of RGB bits within the 24-bit d bus, as specified by the DPI specification. 0= 16-bit Configuration 1 1= 16-bit Configuration 2 2= 16-bit Configuration 3 3= 18-bit Configuration 1 4= 18-bit Configuration 2 5= 24-bit

13.6.9.4 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FORMAT

Address: 30A0_0000h base + 20Ch offset = 30A0_020Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

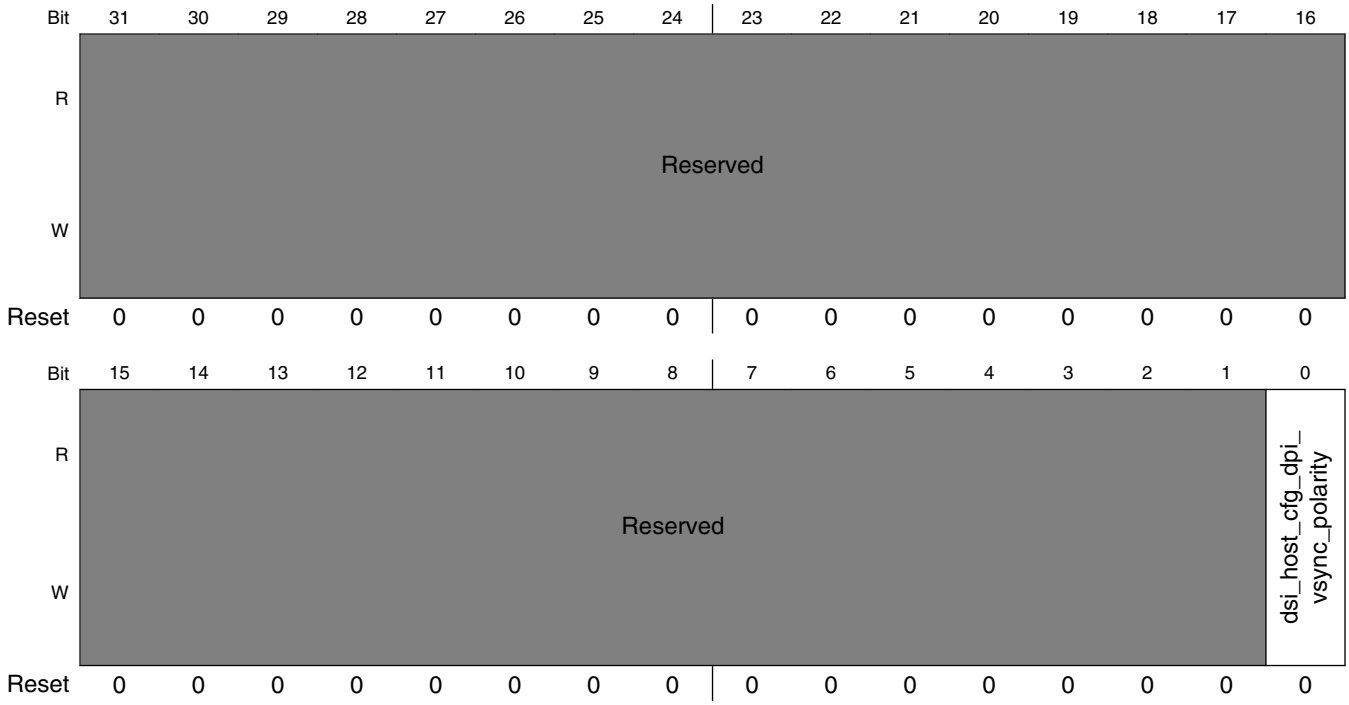
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dsi_host_cfg_dpi_pixel_format	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_PIXEL_FORMAT field descriptions

Field	Description
31–2 -	This field is reserved.
dsi_host_cfg_dpi_pixel_format	Sets the DSI packet type of the pixels. 0 - 16 bit 1 - 18 bit 2 - 18 bit loosely packed 3 - 24 bit

13.6.9.5 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VSYNC_POLARIT

Address: 30A0_0000h base + 210h offset = 30A0_0210h

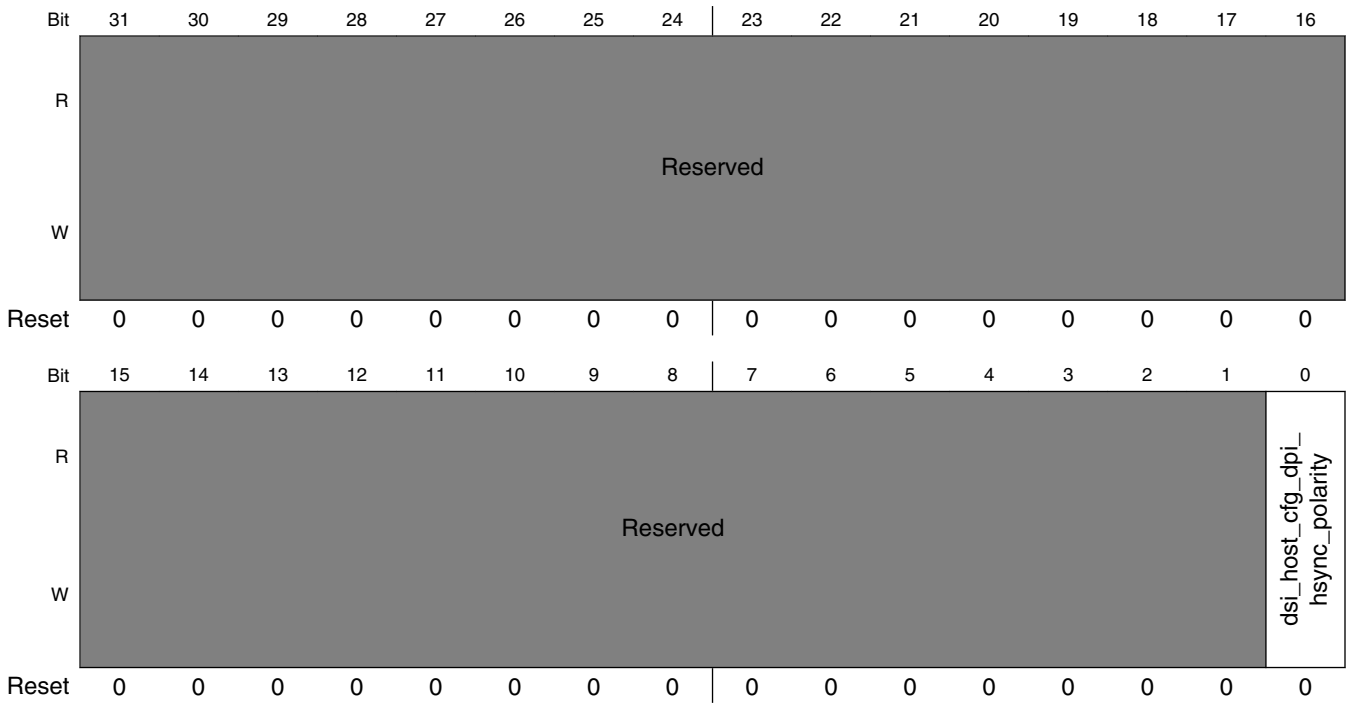


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VSYNC_POLARITY field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_dpi_vsync_polarity	Sets polarity of dpi_vsync_input 0 - active low 1 - active high

13.6.9.6 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSYNC_POLARIT

Address: 30A0_0000h base + 214h offset = 30A0_0214h



MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSYNC_POLARITY field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_dpi_hsync_polarity	Sets polarity of dpi_hsync_input 0 - active low 1 - active high

13.6.9.7 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VIDEO_MODE

Address: 30A0_0000h base + 218h offset = 30A0_0218h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dsi_host_cfg_dpi_video_mode	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VIDEO_MODE field descriptions

Field	Description
31–2 -	This field is reserved.
dsi_host_cfg_dpi_video_mode	Select DSI video mode that the host DPI module should generate packets for. 2'b00 - Non-Burst mode with Sync Pulses 2'b01 - Non-Burst mode with Sync Events 2'b10 - Burst mode 2'b11 - Reserved, not valid

13.6.9.8 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HFP

Address: 30A0_0000h base + 21Ch offset = 30A0_021Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																dsi_host_cfg_dpi_hfp															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HFP field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_cfg_dpi_hfp	Sets the DSI packet payload size, in bytes, of the horizontal front porch blanking packet.

13.6.9.9 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HBP

Address: 30A0_0000h base + 220h offset = 30A0_0220h

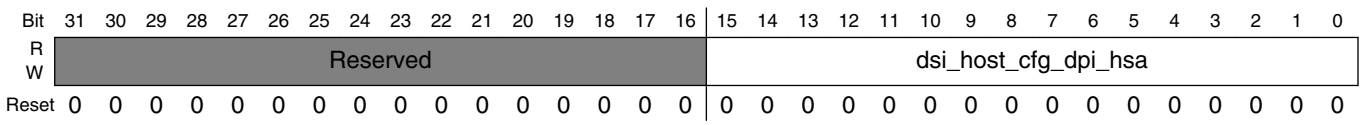
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																dsi_host_cfg_dpi_hbp															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HBP field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_cfg_ dpi_hbp	Sets the DSI packet payload size, in bytes, of the horizontal back porch blanking packet.

13.6.9.10 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSA

Address: 30A0_0000h base + 224h offset = 30A0_0224h

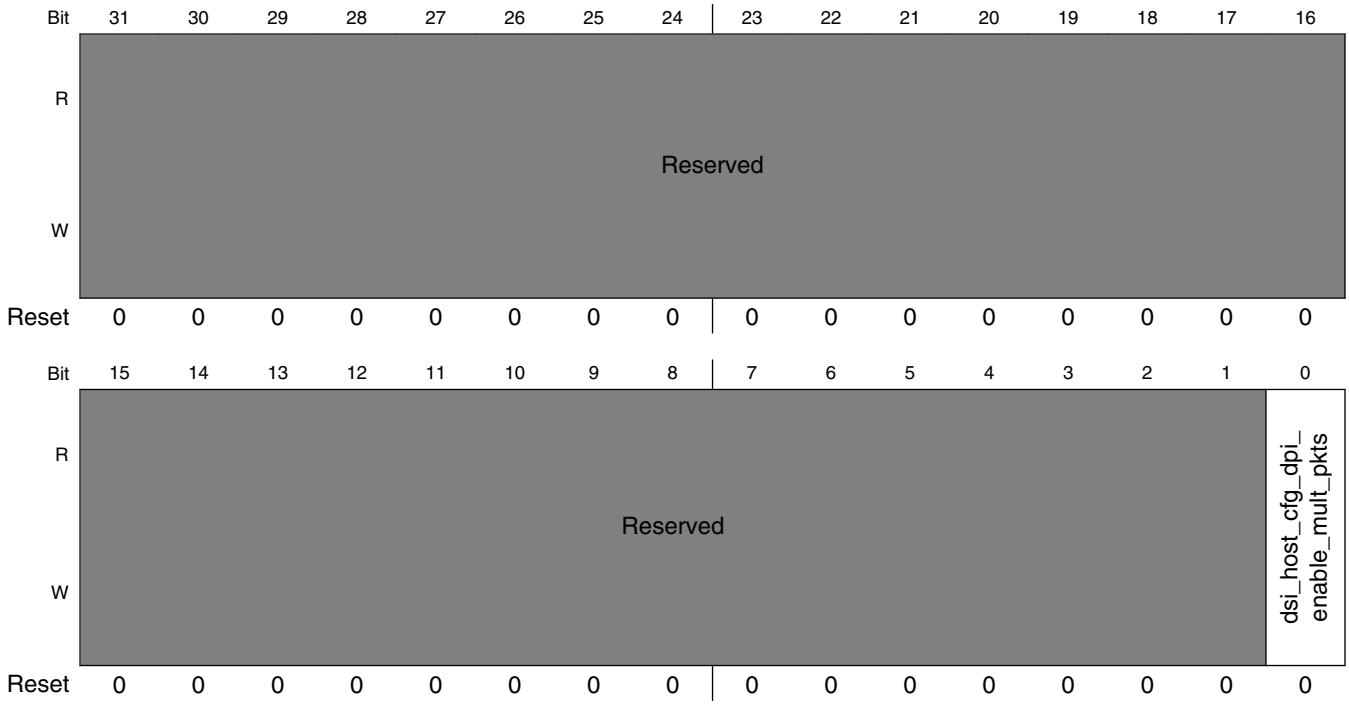


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_HSA field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_cfg_ dpi_hsa	Sets the DSI packet payload size, in bytes, of the horizontal sync width filler blanking packet.

13.6.9.11 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_ENABLE_MULT_

Address: 30A0_0000h base + 228h offset = 30A0_0228h

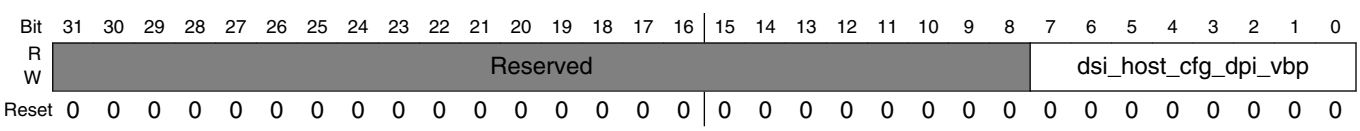


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_ENABLE_MULT_PKTS field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_dpi_enable_mult_pkts	Enable Multiple packets per video line. When enabled, cfg_dpi_pixel_payload_size must be set to exactly half the size of the video line. 0 - Video Line is sent in a single packet 1 - Video Line is sent in two packets

13.6.9.12 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VBP

Address: 30A0_0000h base + 22Ch offset = 30A0_022Ch

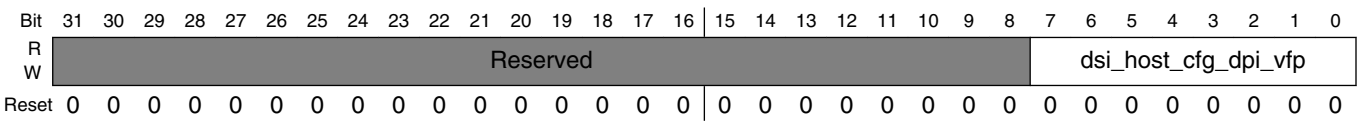


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VBP field descriptions

Field	Description
31–8 -	This field is reserved.
dsi_host_cfg_ dpi_vbp	Sets the number of lines in the vertical back porch.

13.6.9.13 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VFP

Address: 30A0_0000h base + 230h offset = 30A0_0230h

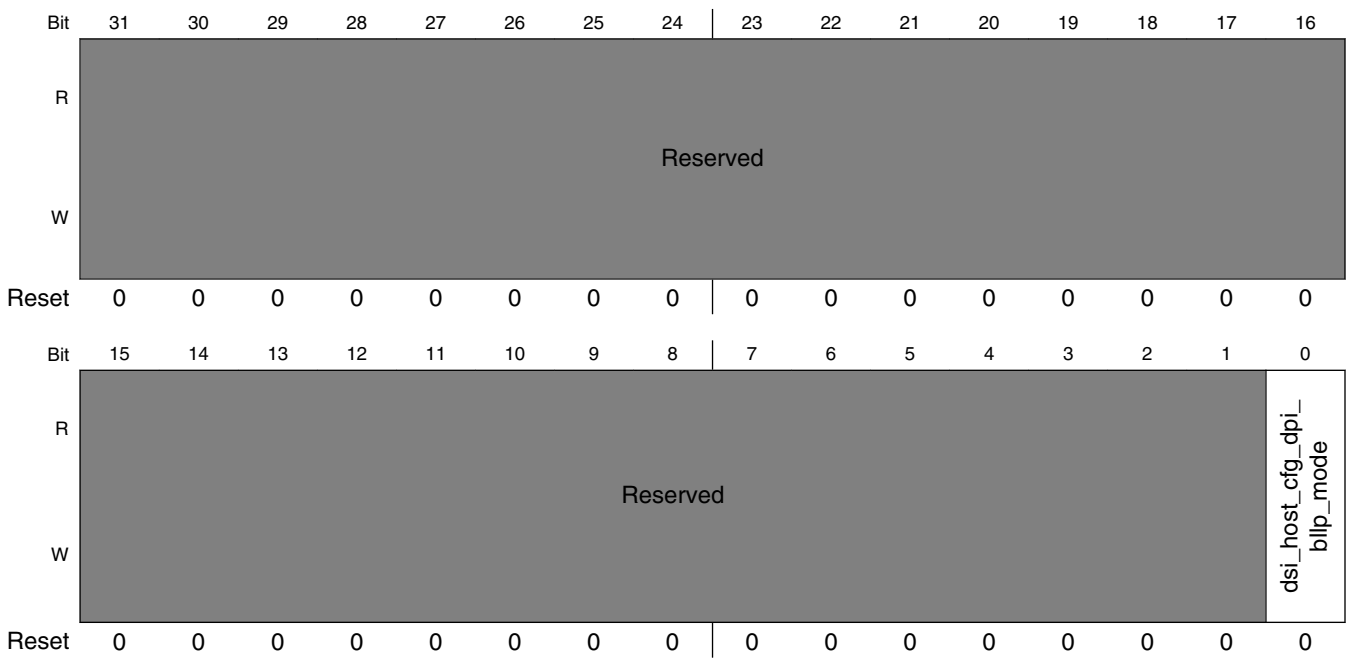


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VFP field descriptions

Field	Description
31–8 -	This field is reserved.
dsi_host_cfg_ dpi_vfp	Sets the number of lines in the vertical front porch.

13.6.9.14 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_BLLP_MODE

Address: 30A0_0000h base + 234h offset = 30A0_0234h

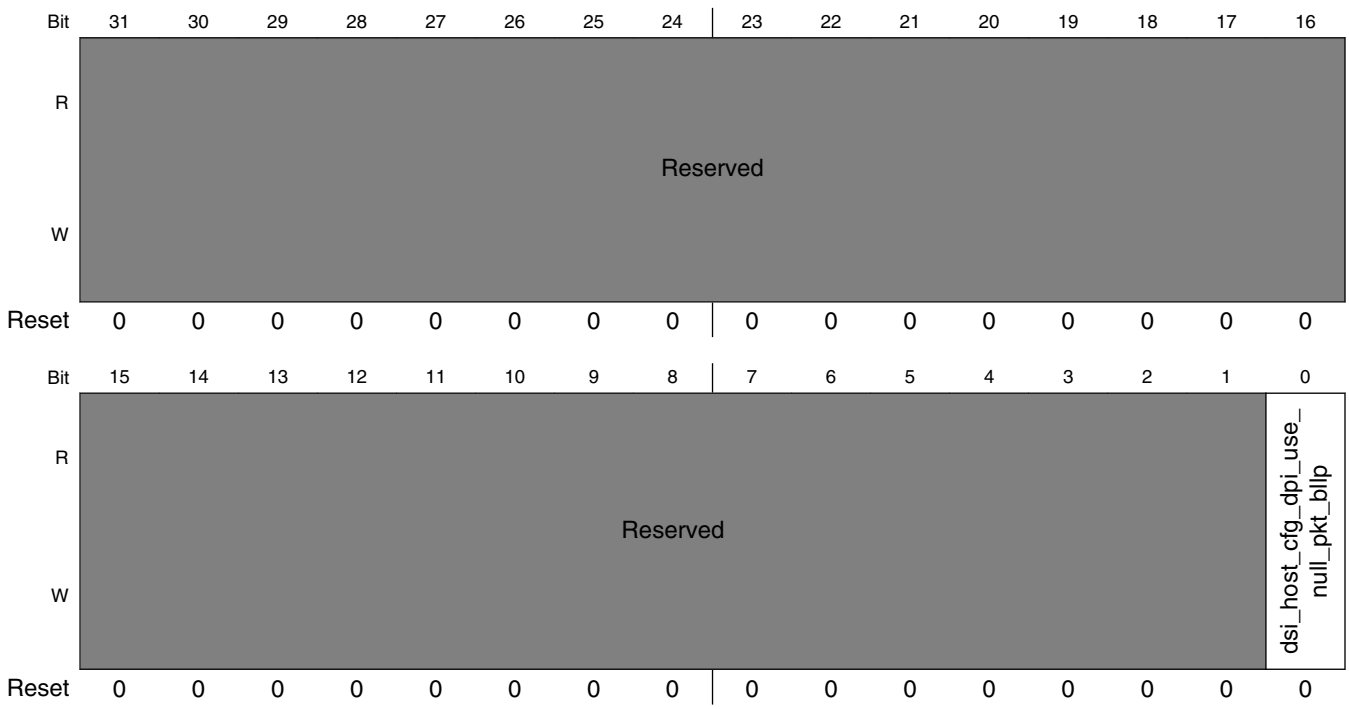


MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_BLLP_MODE field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_ dpi_bllp_mode	Optimize bllp periods to Low Power mode when possible 0 - blanking packets are sent during BLLP periods 1 - LP mode is used for BLLP periods

13.6.9.15 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_USE_NULL_PKT

Address: 30A0_0000h base + 238h offset = 30A0_0238h



MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_USE_NULL_PKT_BLLP field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_cfg_ dpi_use_null_ pkt_bllp	Selects type of blanking packet to be sent during bllp region 0 - Blanking packet used in bllp region 1 - Null packet used in bllp region

13.6.9.16 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VACTIVE

Address: 30A0_0000h base + 23Ch offset = 30A0_023Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																dsi_host_cfg_dpi_vactive															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VACTIVE field descriptions

Field	Description
31–14 -	This field is reserved.
dsi_host_cfg_ dpi_vactive	Sets the number of lines in the vertical active area. This field is equivalent to (real vertical size) - 1. For example, for an image of size 640x480, the bit field should be set as 479.

13.6.9.17 MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VC

Address: 30A0_0000h base + 240h offset = 30A0_0240h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dsi_host_ cfg_dpi_vc	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_DPI_INTFC_DSI_HOST_CFG_DPI_VC field descriptions

Field	Description
31–2 -	This field is reserved.
dsi_host_cfg_ dpi_vc	Sets the Virtual Channel (VC) of packets that will be sent to the receive packet interface. Packets with VC not equal to this value are discarded and the "DSI VC ID Invalid" bit (bit 12) in the DSI error report is set.

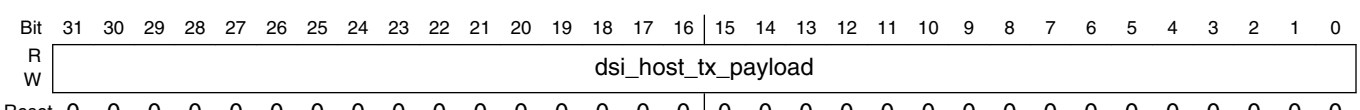
13.6.10 MIPI DSI Host APB PKT IF Memory Map/Register Definition

MIPI_DSI_HOST_APB_PKT_IF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0280	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_TX_PAYLOAD	32	R/W	0000_0000h	13.6.10.1/ 4924
30A0_0284	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_CONTROL	32	R/W	0000_0000h	13.6.10.2/ 4925
30A0_0288	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_SEND_PACKET	32	R/W	0000_0000h	13.6.10.3/ 4925
30A0_028C	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_STATUS	32	R	0000_0000h	13.6.10.4/ 4926
30A0_0290	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_WR_LEVEL	32	R	0000_0000h	13.6.10.5/ 4926
30A0_0294	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_RD_LEVEL	32	R	0000_0000h	13.6.10.6/ 4927
30A0_0298	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PAYLOAD	32	R	0000_0000h	13.6.10.7/ 4927
30A0_029C	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PKT_HEADER	32	R	0000_0000h	13.6.10.8/ 4928
30A0_02A0	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS	32	R	0000_0000h	13.6.10.9/ 4928
30A0_02A4	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS2	32	R	0000_0000h	13.6.10.10/ 4929
30A0_02A8	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK	32	R/W	0000_0000h	13.6.10.11/ 4929
30A0_02AC	MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK2	32	R/W	0000_0000h	13.6.10.12/ 4930

13.6.10.1 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_TX_PAYLOAD

Address: 30A0_0000h base + 280h offset = 30A0_0280h

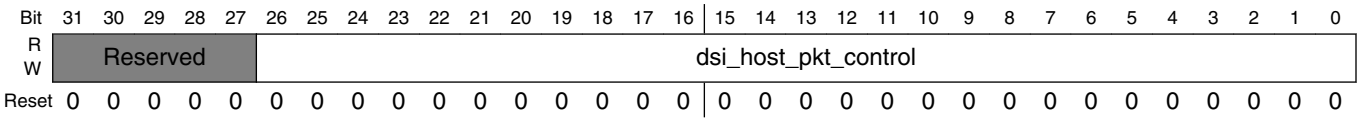


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_TX_PAYLOAD field descriptions

Field	Description
dsi_host_tx_payload	Tx Payload data write register. Writes to this registers load the payload fifo with 32 bit values.

13.6.10.2 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_CONTROL

Address: 30A0_0000h base + 284h offset = 30A0_0284h

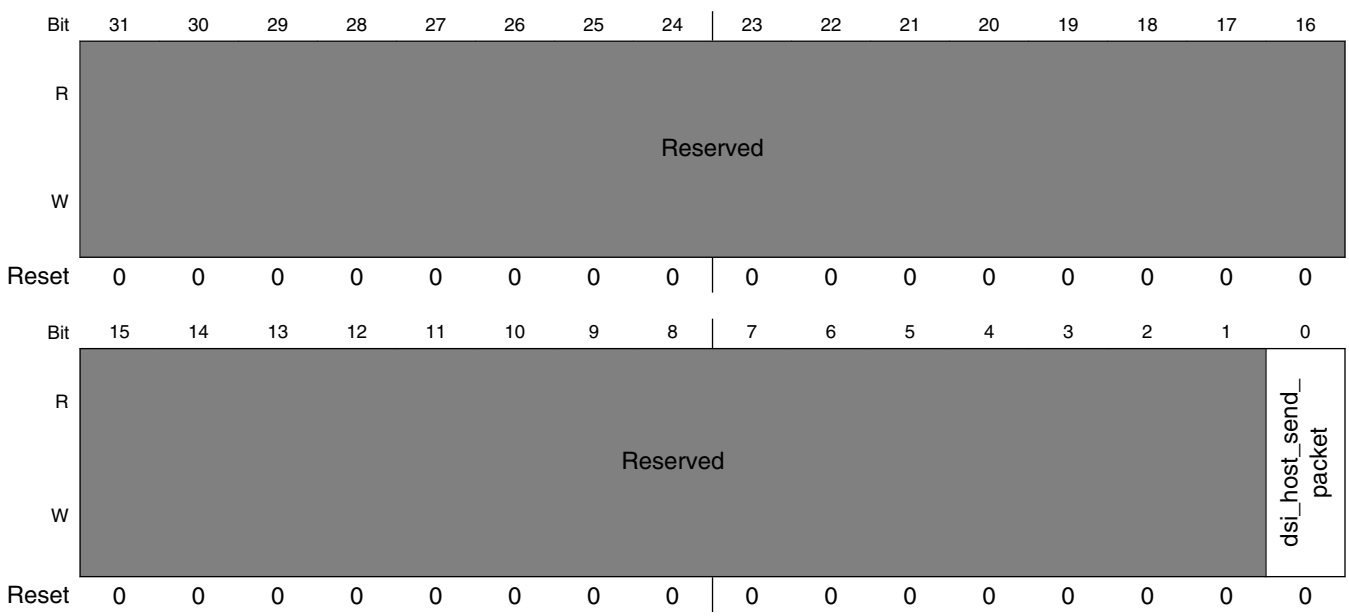


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_CONTROL field descriptions

Field	Description
31–27 -	This field is reserved.
dsi_host_pkt_ control	Tx packet control register. • [15:0] - Packet word count • [17:16] - Packet Virtual Channel • [23:18] - Packet Header DSI Data Type • [24] - Lp or HS select. 0 - LP mode, 1 - HS mode • [25] - perform BTA after packet is sent • [26] - perform BTA only, no packet tx

13.6.10.3 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_SEND_PACKET

Address: 30A0_0000h base + 288h offset = 30A0_0288h

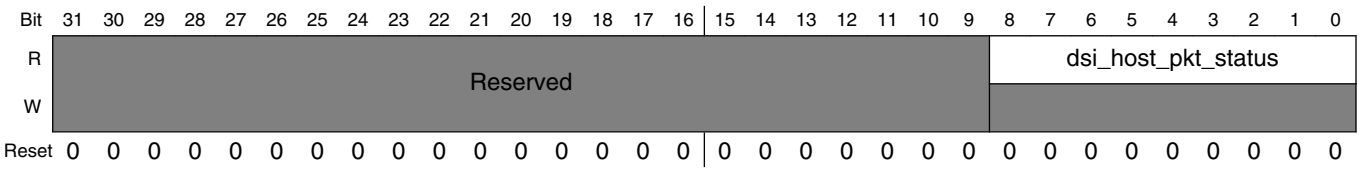


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_SEND_PACKET field descriptions

Field	Description
31–1 -	This field is reserved.
0 dsi_host_send_packet	Tx send packet. Writing to this register causes the packet described in dsi_host_pkt_control to be sent.

13.6.10.4 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_STATUS

Address: 30A0_0000h base + 28Ch offset = 30A0_028Ch

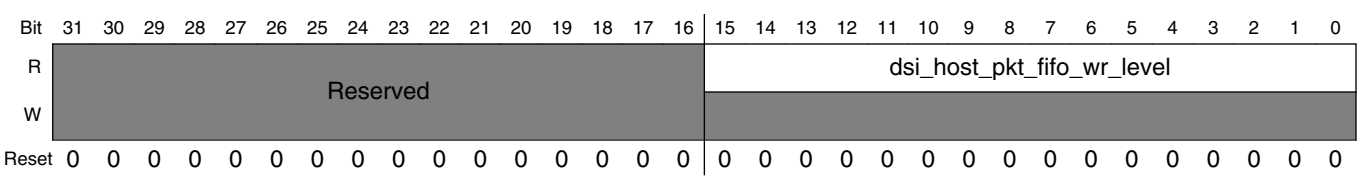


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_STATUS field descriptions

Field	Description
31–9 -	This field is reserved.
dsi_host_pkt_status	Status of APB to packet interface • [0] - state machine not idle • [1] - Tx packet done • [2] - dphy direction 0 - tx had control, 1 - rx has control • [3] - tx fifo overflow • [4] - tx fifo underflow • [5] - rx fifo overflow • [6] - rx fifo underflow • [7] - rx packet header has been received • [8] - all rx packet payload data has been receive d

13.6.10.5 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_WR_LEVEL

Address: 30A0_0000h base + 290h offset = 30A0_0290h

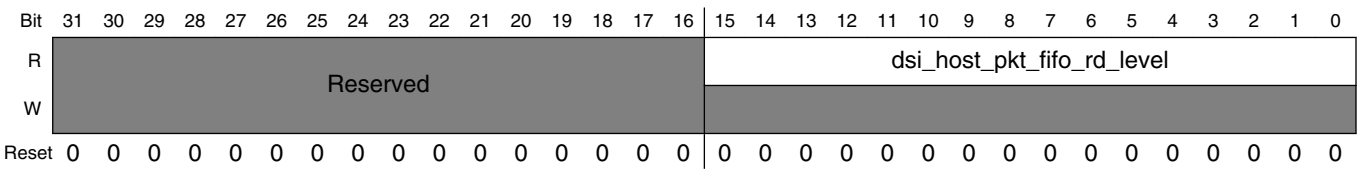


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_WR_LEVEL field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_pkt_ fifo_wr_level	Write level of APB to pkt interface fifo

13.6.10.6 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_RD_LEVEL

Address: 30A0_0000h base + 294h offset = 30A0_0294h

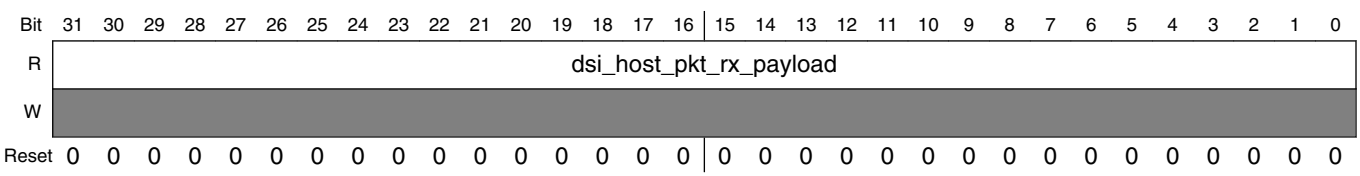


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_FIFO_RD_LEVEL field descriptions

Field	Description
31–16 -	This field is reserved.
dsi_host_pkt_ fifo_rd_level	Read level of APB to pkt interface fifo

13.6.10.7 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PAYLOAD

Address: 30A0_0000h base + 298h offset = 30A0_0298h

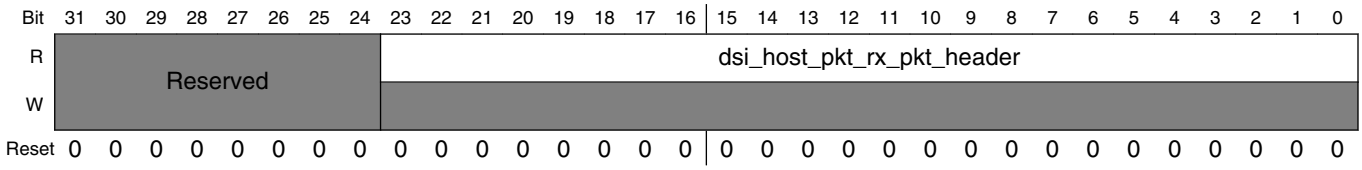


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PAYLOAD field descriptions

Field	Description
dsi_host_pkt_rx_ payload	APB to pkt interface rx payload read

13.6.10.8 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PKT_HEADER

Address: 30A0_0000h base + 29Ch offset = 30A0_029Ch

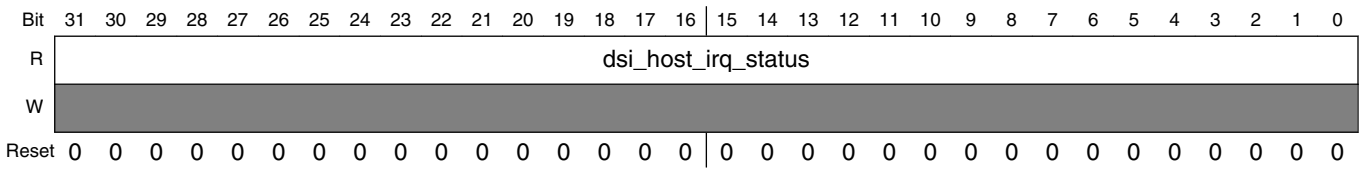


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_PKT_RX_PKT_HEADER field descriptions

Field	Description
31–24 -	This field is reserved.
dsi_host_pkt_rx_ pkt_header	APB to pkt interface rx packet header • [15:0] word count • [21:16] data type • [23:22] Virtual Channel

13.6.10.9 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS

Address: 30A0_0000h base + 2A0h offset = 30A0_02A0h

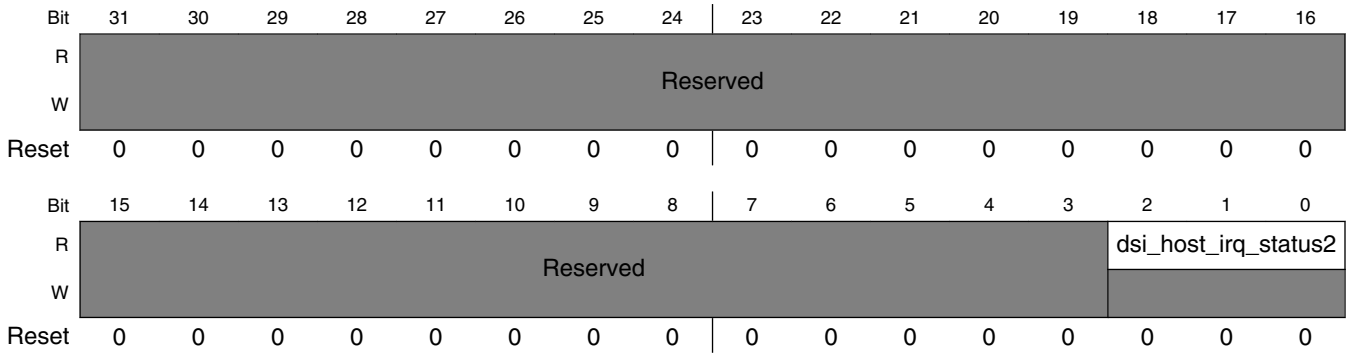


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS field descriptions

Field	Description
dsi_host_irq_ status	Status of APB to packet interface • [0] - state machine not idle • [1] - Tx packet done • [2] - dphy direction 0 - tx had control, 1 - rx has control • [3] - tx fifo overflow • [4] - tx fifo underflow • [5] - rx fifo overflow • [6] - rx fifo underflow • [7] - rx packet header has been received • [8] - all rx packet payload data has been received • [28:9] - map directory to dsi host controller status_out port bit descriptions • [29] - host bta timeout, host controller host_bta_timeout port • [30] - low power rx timeout, host controller lp_rx_timeout port • [31] - high speed tx timeout, host controller hs_tx_timeout port

13.6.10.10 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS2

Address: 30A0_0000h base + 2A4h offset = 30A0_02A4h

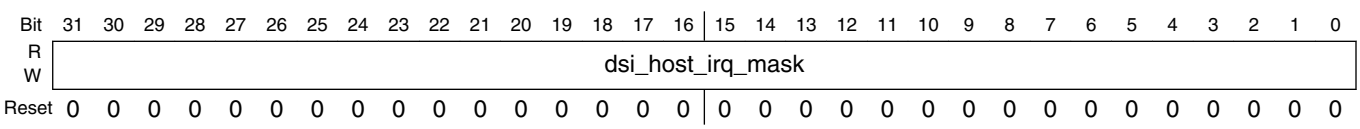


MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_STATUS2 field descriptions

Field	Description
31–3 -	This field is reserved.
dsi_host_irq_ status2	Status of APB to packet interface part 2. Read part 2 first then dsi_host_irq_status. Reading dsi_host_irq_status will clear both status and status 2. • [0] - single bit ecc error • [1] - multi bit ecc error • [2] - crc error

13.6.10.11 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK

Address: 30A0_0000h base + 2A8h offset = 30A0_02A8h



MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK field descriptions

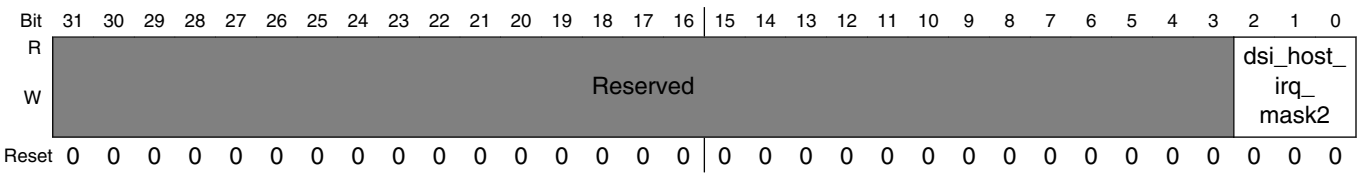
Field	Description
dsi_host_irq_ mask	irq mask • [0] - state machine not idle • [1] - Tx packet done • [2] - dphy direction 0 - tx had control, 1 - rx has control • [3] - tx fifo overflow • [4] - tx fifo underflow • [5] - rx fifo overflow • [6] - rx fifo underflow • [7] - rx packet header has been received • [8] - all rx packet payload data has been received • [28:9] - map directory to dsi host controller status_out port bit descriptions

MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK field descriptions (continued)

Field	Description
	• [29] - host bta timeout, host controller host_bta_timeout port • [30] - low power rx timeout, host controller lp_rx_timeout port • [31] - high speed tx timeout, host controller hs_tx_timeout port

13.6.10.12 MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK2

Address: 30A0_0000h base + 2ACh offset = 30A0_02ACh



MIPI_DSI_HOST_APB_PKT_IF_DSI_HOST_IRQ_MASK2 field descriptions

Field	Description
31–3 -	This field is reserved.
dsi_host_irq_mask2	irq mask 2 • [0] - single bit ecc error • [1] - multi bit ecc error • [2] - crc error

13.6.11 MIPI DSI Host IP1 DPHY INTFC Memory Map/Register Definition

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0300	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_DPHY	32	R	0000_0001h	13.6.11.1/ 4932
30A0_0304	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_PREPARE	32	R	0000_0000h	13.6.11.2/ 4932
30A0_0308	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_PREPARE	32	R	0000_0000h	13.6.11.3/ 4933
30A0_030C	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_ZERO	32	R	0000_0000h	13.6.11.4/ 4934
30A0_0310	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_ZERO	32	R	0000_0000h	13.6.11.5/ 4935

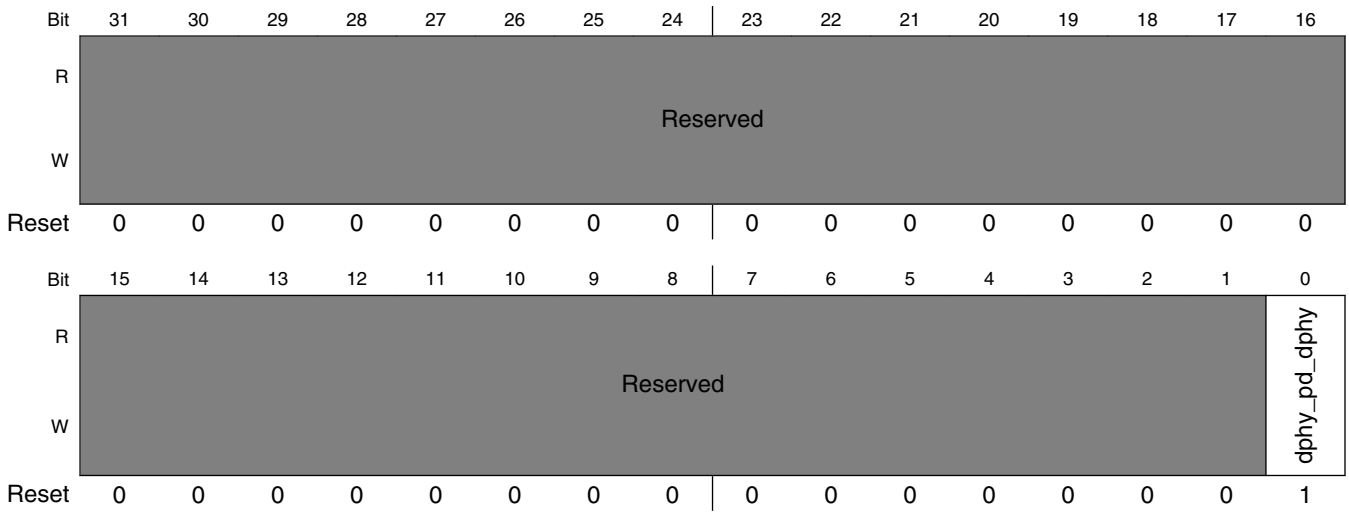
Table continues on the next page...

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A0_0314	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_TRAIL	32	R	0000_0000h	13.6.11.6/ 4936
30A0_0318	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_TRAIL	32	R	0000_0000h	13.6.11.7/ 4936
30A0_031C	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_PLL	32	R	0000_0001h	13.6.11.8/ 4937
30A0_0320	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_TST	32	R	0000_0025h	13.6.11.9/ 4938
30A0_0324	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CN	32	R	0000_0000h	13.6.11.10/ 4938
30A0_0328	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CM	32	R	0000_0000h	13.6.11.11/ 4939
30A0_032C	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CO	32	R	0000_0000h	13.6.11.12/ 4940
30A0_0330	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK	32	R	0000_0000h	13.6.11.13/ 4941
30A0_0334	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK_BYP	32	R	0000_0000h	13.6.11.14/ 4942
30A0_0338	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RTERM_SEL	32	R	0000_0000h	13.6.11.15/ 4942
30A0_033C	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_AUTO_PD_EN	32	R	0000_0000h	13.6.11.16/ 4943
30A0_0340	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXLPRP	32	R	0000_0000h	13.6.11.17/ 4944
30A0_0344	MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXCDRP	32	R	0000_0000h	13.6.11.18/ 4944

13.6.11.1 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_DPHY

Address: 30A0_0000h base + 300h offset = 30A0_0300h

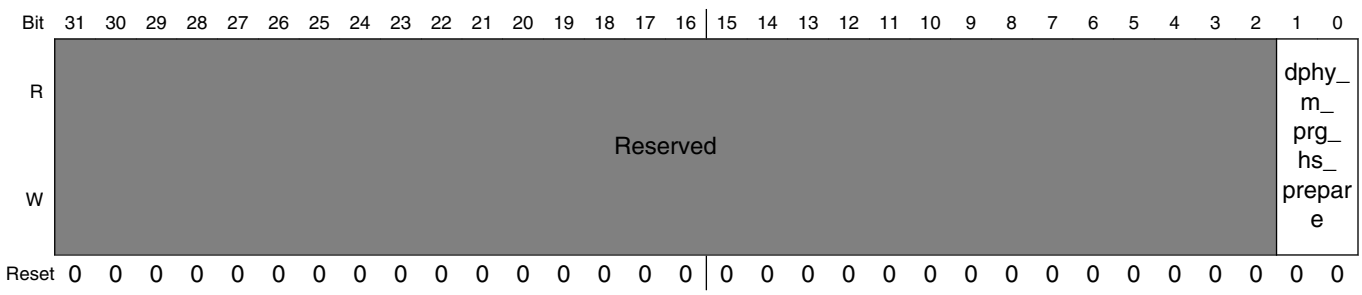


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_DPHY field descriptions

Field	Description
31–1 -	This field is reserved.
0 dphy_pd_dphy	DPHY PD_DPHY input control. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.2 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_PREPA

Address: 30A0_0000h base + 304h offset = 30A0_0304h



MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_PREPARE field descriptions

Field	Description
31–2 -	This field is reserved.

Table continues on the next page...

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_PREPARE field descriptions (continued)

Field	Description
dphy_m_prg_hs_prepare	DPHY m_PRG_HS_PREPARE input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 00 1 01 1.5 10 2 11 2.5

13.6.11.3 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_PREP

Address: 30A0_0000h base + 308h offset = 30A0_0308h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

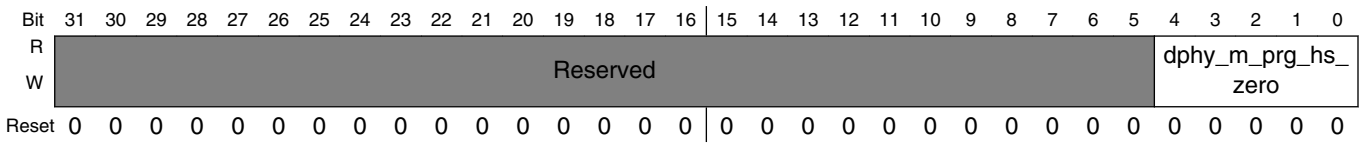
Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															dphy_mc_prg_hs_prepare
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_PREPARE field descriptions

Field	Description
31–1 -	This field is reserved.
0 dphy_mc_prg_hs_prepare	DPHY mc_PRG_HS_PREPARE input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 0 1 1 1.5

13.6.11.4 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_ZERO

Address: 30A0_0000h base + 30Ch offset = 30A0_030Ch

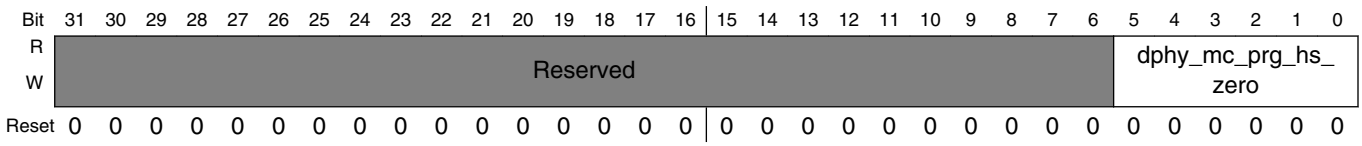


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_ZERO field descriptions

Field	Description
31–5 -	This field is reserved.
dphy_m_prg_hs_zero	DPHY m_PRG_HS_ZERO input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 00000 0 00001 1 00010 2 00011 3 00100 4 00101 5 00110 6 00111 7 01000 8 01001 9 01010 10 01011 11 01100 12 01101 13 01110 14 01111 15 10000 16 10001 17 10010 18 10011 19 10100 20 10101 21 10110 22 10111 23 11000 24 11001 25 11010 26 11011 27 11100 28 11101 29 11110 30 11111 31

13.6.11.5 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_ZERO

Address: 30A0_0000h base + 310h offset = 30A0_0310h



MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_ZERO field descriptions

Field	Description
31–6 -	This field is reserved.
dphy_mc_prg_hs_zero	DPHY mc_PRG_HS_ZERO input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 100000 32 100001 33 100010 34 100011 35 100100 36 100101 37 100110 38 100111 39 101000 40 101001 41 101010 42 101011 43 101100 44 101101 45 101110 46 101111 47 110000 48 110001 49 110010 50 110011 51 110100 52 110101 53 110110 54 110111 55 111000 56 111001 57 111010 58 111011 59 111100 60 111101 61 111110 62 111111 63

13.6.11.6 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_TRAIL

Address: 30A0_0000h base + 314h offset = 30A0_0314h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																												dphy_m_prg_hs_trail			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_M_PRG_HS_TRAIL field descriptions

Field	Description
31–4 -	This field is reserved.
dphy_m_prg_hs_trail	DPHY m_PRG_HS_TRAIL input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 0000 0 0001 1 0010 2 0011 3 0100 4 0101 5 0110 6 0111 7 1000 8 1001 9 1010 10 1011 11 1100 12 1101 13 1110 14 1111 15

13.6.11.7 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_TRAI

Address: 30A0_0000h base + 318h offset = 30A0_0318h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																												dphy_mc_prg_hs_trail			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_MC_PRG_HS_TRAIL field descriptions

Field	Description
31–4 -	This field is reserved.
dphy_mc_prg_hs_trail	DPHY mc_PRG_HS_TRAIL input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 0000 0 0001 1 0010 2 0011 3 0100 4 0101 5 0110 6 0111 7 1000 8 1001 9 1010 10 1011 11 1100 12 1101 13 1110 14 1111 15

13.6.11.8 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_PLL

Address: 30A0_0000h base + 31Ch offset = 30A0_031Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															PD
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_PD_PLL field descriptions

Field	Description
31–1 -	This field is reserved.
0 PD	DPHY PD_PLL input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.9 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_TST

Address: 30A0_0000h base + 320h offset = 30A0_0320h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																										TST					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_TST field descriptions

Field	Description
31–6 -	This field is reserved.
TST	DPHY TST input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.10 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CN

Address: 30A0_0000h base + 324h offset = 30A0_0324h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																											CN				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CN field descriptions

Field	Description
31–5 -	This field is reserved.
CN	DPHY PLL Input Divider. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 11111 Divide by 1 00000 Divide by 2 10000 Divide by 3 11000 Divide by 4 11100 Divide by 5 01110 Divide by 6 00111 Divide by 7 10011 Divide by 8 01001 Divide by 9 00100 Divide by 10 00010 Divide by 11 10001 Divide by 12 01000 Divide by 13 10100 Divide by 14

Table continues on the next page...

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CN field descriptions (continued)

Field	Description
	01010 Divide by 15 10101 Divide by 16 11010 Divide by 17 11101 Divide by 18 11110 Divide by 19 01111 Divide by 20 10111 Divide by 21 11011 Divide by 22 01101 Divide by 23 10110 Divide by 24 01011 Divide by 25 00101 Divide by 26 10010 Divide by 27 11001 Divide by 28 01100 Divide by 29 00110 Divide by 30 00011 Divide by 31 00001 Divide by 32

13.6.11.11 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CM

Address: 30A0_0000h base + 328h offset = 30A0_0328h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved																								CM							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CM field descriptions

Field	Description
31–8 -	This field is reserved.
CM	DPHY PLL Feedback Divider. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 111x0000 Divide by 16 : : 111x1111 Divide by 31 11000000 Divide by 32 : : 11011111 Divide by 63 10000000 Divide by 64 : : 10111111 Divide by 127 00000000 Divide by 128

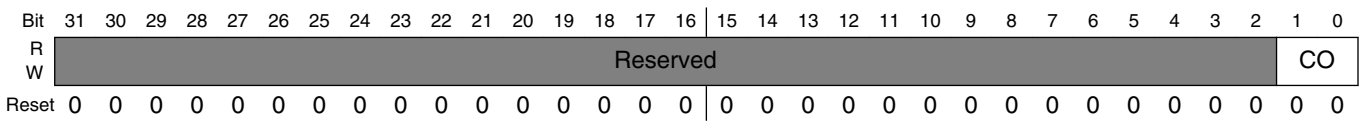
Table continues on the next page...

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CM field descriptions (continued)

Field	Description
	: 01111111 Divide by 255

13.6.11.12 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CO

Address: 30A0_0000h base + 32Ch offset = 30A0_032Ch

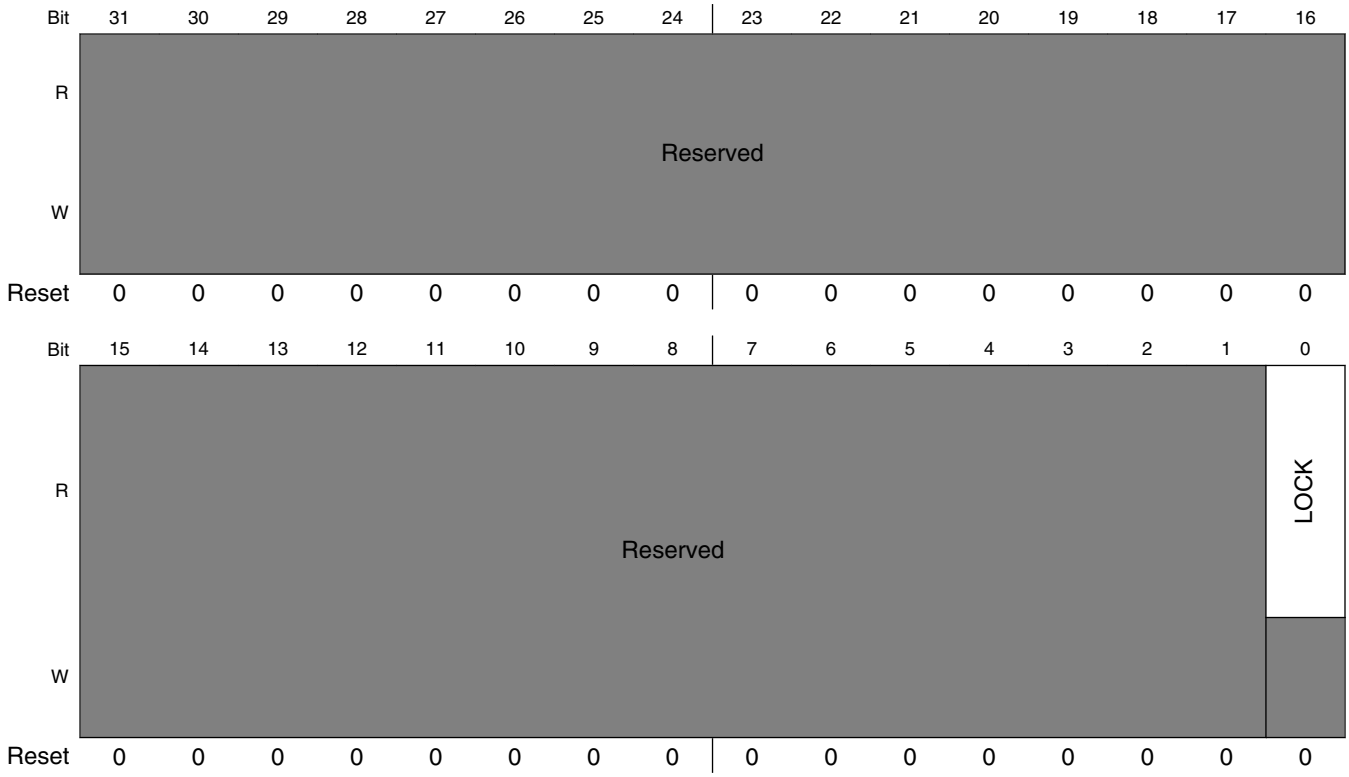


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_CO field descriptions

Field	Description
31–2 -	This field is reserved.
CO	DPHY PLL Output Divider. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section. 00 Divide by 1 01 Divide by 2 10 Divide by 4 11 Divide by 8

13.6.11.13 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK

Address: 30A0_0000h base + 330h offset = 30A0_0330h

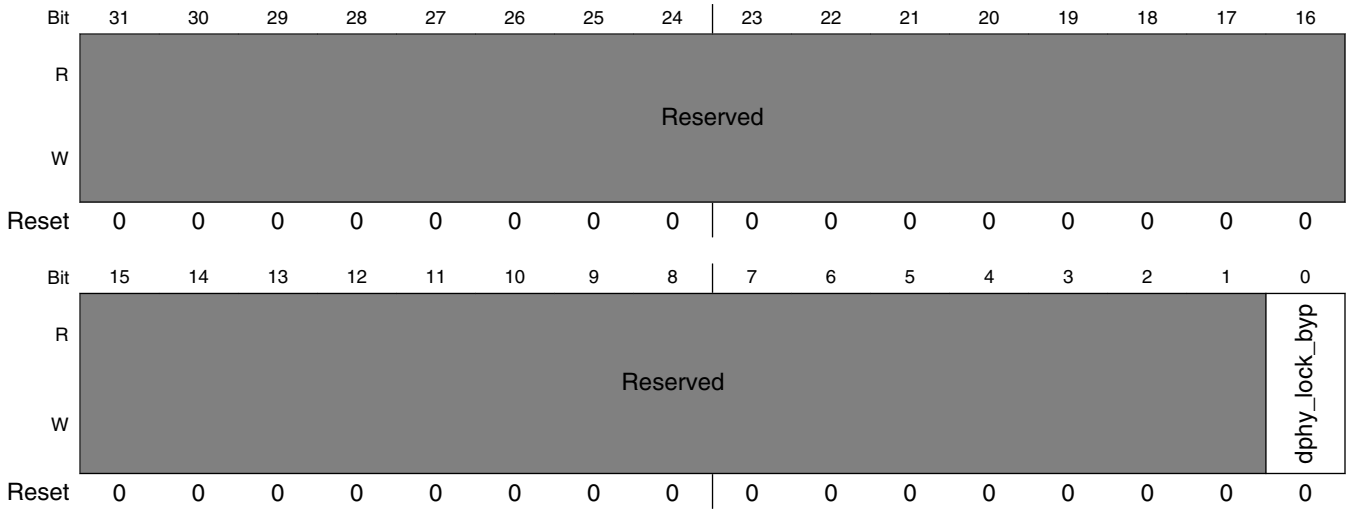


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK field descriptions

Field	Description
31–1 -	This field is reserved.
0 LOCK	DPHY PLL LOCK output. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.14 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK_BYP

Address: 30A0_0000h base + 334h offset = 30A0_0334h

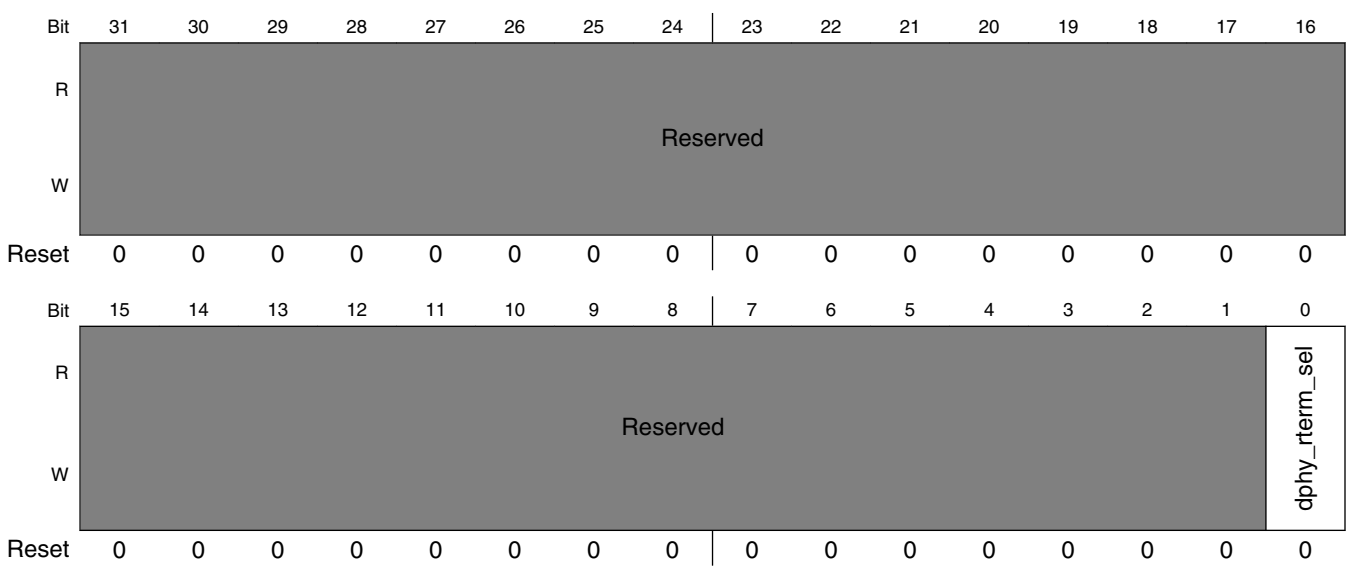


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_LOCK_BYP field descriptions

Field	Description
31–1 -	This field is reserved.
0 dphy_lock_byp	DPHY LOCK_BYP input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.15 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RTERM_SEL

Address: 30A0_0000h base + 338h offset = 30A0_0338h

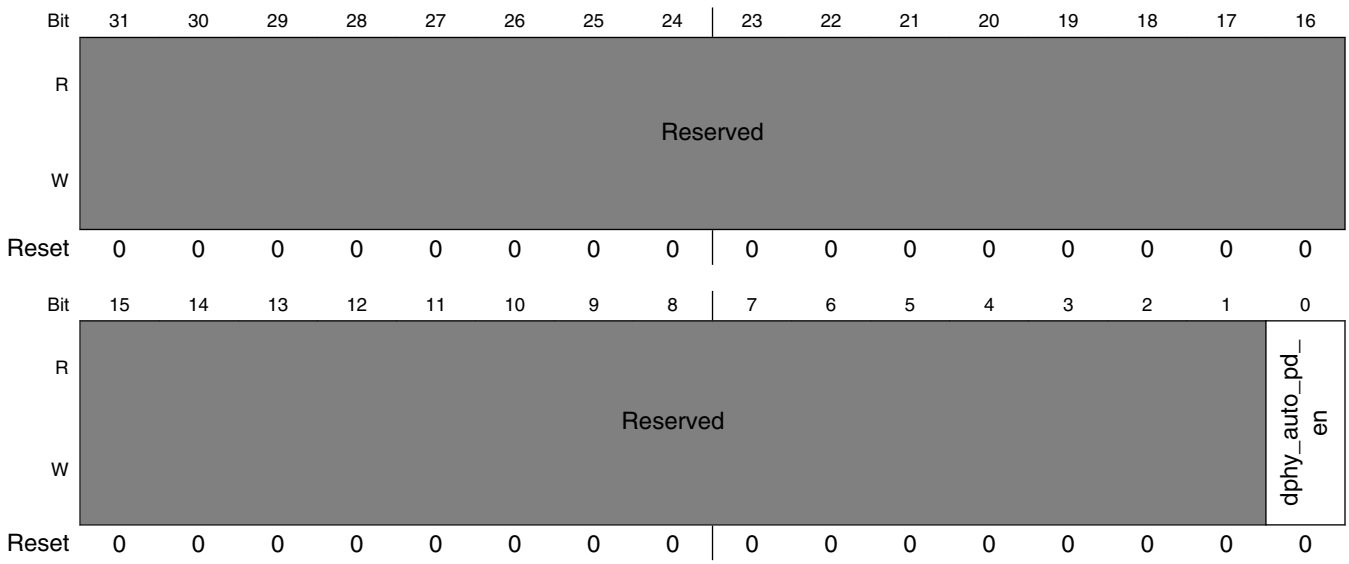


MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RTERM_SEL field descriptions

Field	Description
31–1 -	This field is reserved.
0 dphy_rterm_sel	DPHY RTERM_SEL input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.16 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_AUTO_PD_EN

Address: 30A0_0000h base + 33Ch offset = 30A0_033Ch



MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_AUTO_PD_EN field descriptions

Field	Description
31–1 -	This field is reserved.
0 dphy_auto_pd_en	DPHY AUTO_PD_EN input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.17 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXLPRP

Address: 30A0_0000h base + 340h offset = 30A0_0340h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dphy_rxlprp	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXLPRP field descriptions

Field	Description
31–2 -	This field is reserved.
dphy_rxlprp	DPHY RXLPRP input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.6.11.18 MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXCDRP

Address: 30A0_0000h base + 344h offset = 30A0_0344h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved														dphy_rxcdrp	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

MIPI_DSI_HOST_FSL_IP1_DPHY_INTFC_DPHY_RXCDRP field descriptions

Field	Description
31–2 -	This field is reserved.
dphy_rxcdrp	DPHY RXCDRP input. Detailed information about this parameter programming is available in the MIPI-DSI DPHY section.

13.7 CSI Bridge (CSI)

13.7.1 Overview

This chapter presents the CMOS Sensor Interface (CSI) architecture, operation principles, and programming model.

The CSI enables the chip to connect directly to external CMOS image sensors. CMOS image sensors are separated into two classes, dumb and smart. Dumb sensors are those that support only traditional sensor timing (Vertical SYNC and Horizontal SYNC) and output only Bayer and statistics data, while smart sensors support CCIR656 video decoder formats and perform additional processing of the image (for example, image compression, image pre-filtering, and various data output formats).

The capabilities of the CSI include:

- Configurable interface logic to support most commonly available CMOS sensors.
- Support for CCIR656 video interface as well as traditional sensor interface.
- 8-bit / 16-bit / 24-bit data port for YCbCr, YUV, or RGB data input.
- 8-bit / 10-bit / 16-bit data port for Bayer data input.
- Full control of 8-bit/pixel, 10-bit/pixel or 16-bit / pixel data format to 64-bit receive FIFO packing.
- 256 x 64 FIFO to store received image pixel data.
- Reveive FIFO overrun protection mechanism.
- Embedded DMA controllers to transfer data from receive FIFO or statistic FIFO through AHB bus.
- Support 2D DMA transfer from the receive FIFO to the frame buffers in the external memory.
- Support double bufferring two frames in the external memory.
- Single interrupt source to interrupt controller from maskable interrupt sources: Start of Frame, End of Frame, Change of Field, FIFO full, FIFO overrun, DMA transfer done, CCIR error and AHB bus response error.
- Configurable master clock frequency output to sensor.
- Statistic data generation for Auto Exposure (AE) and Auto White Balance (AWB) control of the camera (only for Bayer data and 8-bit/pixel format).
- Supports simple deinterlacing of interlaced input.

13.7.2 Clocks

The following table describes the clock sources for CSI.

Table 13-30. CSI Clocks

Clock name	Clock Root	Description
csi_hclk		Module clock
ipg_clk		Peripheral clock
ipg_clk_s		Peripheral access clock
ipg_clk_s_raw		Peripheral raw data clock

13.7.3 Principles of Operation

The information found here describes the modes of operation of the sensor interface.

The CSI is designed to support generic sensor interface timing as well as CCIR656 video interface timing. Traditional CMOS sensors typically use VSYNC (SOF), HSYNC (BLANK), and PIXCLK signals to output Bayer or YUV data. Smart CMOS sensors, that come with on-chip imaging processing, usually support video mode transfer. They use an embedded timing codec to replace the VSYNC and HSYNC signal. The timing codec is defined by the CCIR656 standard.

The CSI can support connection with the sensor as follows.

- To connect with one 8-bit sensor, the sensor data interface should connect to CSI_DATA[9:2].
- To connect with one 10-bit sensor, the sensor data interface should connect to CSI_DATA[9:0].
- To connect with one 16-bit sensor, the sensor data interface should connect to CSI_DATA[15:0].

Table 13-31. CSI input data format

Signal Name	TVdecoder YCbCr 1 Cycle	RGB888 1 Cycle	RGB888/ YUV4444 3 Cycle	RGB666 1 Cycle	RGB565 1 Cycle	YCbCr422 1 Cycle	YCbCr422 2 Cycle	Generic 10 bit	CCIR656
ipp_csi_d[23]	Y[7]	R[7]		R[5]					
ipp_csi_d[22]	Y[6]	R[6]		R[4]					
ipp_csi_d[21]	Y[5]	R[5]		R[3]					

Table continues on the next page...

Table 13-31. CSI input data format (continued)

Signal Name	TVdecoder YCbCr 1 Cycle	RGB888 1 Cycle	RGB888/ YUV4444 3 Cycle	RGB666 1 Cycle	RGB565 1 Cycle	YCbCr422 1 Cycle	YCbCr422 2 Cycle	Generic 10 bit	CCIR656
ipp_csi_d[20]	Y[4]	R[4]		R[2]					
ipp_csi_d[19]	Y[3]	R[3]		R[1]					
ipp_csi_d[18]	Y[2]	R[2]		R[0]					
ipp_csi_d[17]	Y[1]	R[1]		Y[5]					
ipp_csi_d[16]	Y[0]	R[0]		R[4]					
ipp_csi_d[15]	Cb[7]	G[7]		G[5]	R[4]	Y[7]			
ipp_csi_d[14]	Cb[6]	G[6]		G[4]	R[3]	Y[6]			
ipp_csi_d[13]	Cb[5]	G[5]		G[3]	R[2]	Y[5]			
ipp_csi_d[12]	Cb[4]	G[4]		G[2]	R[1]	Y[4]			
ipp_csi_d[11]	Cb[3]	G[3]		G[1]	R[0]	Y[3]			
ipp_csi_d[10]	Cb[2]	G[2]		G[0]	G[5]	Y[2]			
ipp_csi_d[9]	Cb[1]	G[1]	R/G/B[7]	G[5]	G[4]	Y[1]	Y/C[7]	Ge[9]	C/Y[7]
ipp_csi_d[8]	Cb[0]	G[0]	R/G/B[6]	G[4]	G[3]	Y[0]	Y/C[6]	Ge[8]	C/Y[6]
ipp_csi_d[7]	Cr[7]	B[7]	R/G/B[5]	B[5]	G[2]	C[7]	Y/C[5]	Ge[7]	C/Y[5]
ipp_csi_d[6]	Cr[6]	B[6]	R/G/B[4]	B[4]	G[1]	C[6]	Y/C[4]	Ge[6]	C/Y[4]
ipp_csi_d[5]	Cr[5]	B[5]	R/G/B[3]	B[3]	G[0]	C[5]	Y/C[3]	Ge[5]	C/Y[3]
ipp_csi_d[4]	Cr[4]	B[4]	R/G/B[2]	B[2]	B[4]	C[4]	Y/C[2]	Ge[4]	C/Y[2]
ipp_csi_d[3]	Cr[3]	B[3]	R/G/B[1]	B[1]	B[3]	C[3]	Y/C[1]	Ge[3]	C/Y[1]
ipp_csi_d[2]	Cr[2]	B[2]	R/G/B[0]	B[0]	B[2]	C[2]	Y/C[0]	Ge[2]	C/Y[0]
ipp_csi_d[1]	Cr[1]	B[1]		B[5]	B[1]	C[1]		Ge[1]	
ipp_csi_d[0]	Cr[0]	B[0]		B[4]	B[0]	C[0]		Ge[0]	

13.7.3.1 Data Transfer with the Embedded DMA Controllers

The CSI has two embedded DMA controllers, one for the receive FIFO and the other for the statistic FIFO. It supports 2D DMA transfer from the receive FIFO to the frame buffers in the external memory and linear DMA transfer from the statistic FIFO.

To transfer data from the RxFIFO to the external memory, the user should set the start address in the frame buffer where the transferred data is stored, the parameters of the frame buffers, and the parameters of the image coming from the sensor. The user can have two frame buffers in the external memory. Each one will store a frame of image coming from the sensor. The embedded DMA controller will first write the frame buffer1 and then frame buffer2. These two frame buffers will be written by turns. The start address should be aligned in double words and set in the CSIDMASA-FB1 and CSIDMASA-FB2 registers. In the CSIFBUF_PARA register, the user should set the stride of the frame buffer to show how many double words to skip before starting to write the next row of the image. In the CSIIMAG_PARA register, the user should set the width and height of the image coming from the sensor. The RxFF_LEVEL and DMA_REQ_EN_RFF bits in CSICR3 registers also need to be set before the data transfer starts. When the number of the data in the RxFIFO reaches the trigger level, a DMA request will be sent to the embedded DMA controller and the data will be read out from the RxFIFO and written through AHB bus into the external frame buffers. The burst type of transfer can be INCR4, INCR8 and INCR16 by setting DMA_BURST_TYPE_RFF bits in CSICR2 register. After all data in an image frame are transferred, the DMA_TSF_DONE_FB1 or DMA_TSF_DONE_FB2 bit will be set in CSISR register and the interrupt can be triggered if the corresponding enable bit is set in CSICR1 register. The DMA_REFLASH_RFF bit in CSICR3 can be used to activate or restart the embedded DMA controller.

The RxFIFO has the overrun protection mechanism in case the RxFIFO is overrun during data transfer. If the RxFIFO is full and more data needs to be received during the data transfer, the RxFIFO will be overwritten continuously and all 128 words of data in the RxFIFO before overrun occurred will be discarded; the corresponding 128 words memory space in the frame buffer will keep the previous values.

To transfer data from the statistic FIFO to the external memory, the user should set the start address of the externl memory where the transferred data is stored and the total transfer sizes. The start address and the transfer sizes are all aligned in double words and should be set in the CSIDMASA-STATFIFO and CSIDMATS-STATFIFO registers. The STATFF_LEVEL and DMA_REQ_EN_SFF bits in CSICR3 registers should also be set before the data transfer starts. When the number of the data in the STATFIFO reaches the trigger level, a dma request will be sent to the embedded DMA controller and the data will be read out from the STATFIFO and written through AHB bus into the external

memory. The burst type of transfer can be INCR4, INCR8 and INCR16 by setting DMA_BURST_TYPE_SFF bits in CSICR2 register. After all expected data (defined by the total transfer sizes) are transferred, the DMA_TSF_DONE_SFF bit will be set in CSISR register and an interrupt can be triggered if the SFF_DMA_DONE_INTEN is enabled in CSICR1 register. The DMA_REFLASH_SFF bit in CSICR3 can be used to activate or re-start the embedded DMA controller.

13.7.3.2 Gated Clock Mode

VSYNC, HSYNC, and PIXCLK signals are used in gated clock mode.

A frame starts with an active edge on VSYNC, then HSYNC asserts and holds for the entire line. The Pixel clock is valid as long as HSYNC is asserted. Data is latched at the active edge of the valid pixel clocks. HSYNC deasserts at the end of line. Pixel clocks then become invalid and CSI stops receiving data from the stream. For the next line the HSYNC timing repeats. For the next frame the VSYNC timing repeats.

13.7.3.3 Non-Gated Clock Mode

In non-gated clock mode, only the VSYNC and PIXCLK signals are used; the HSYNC signal is ignored.

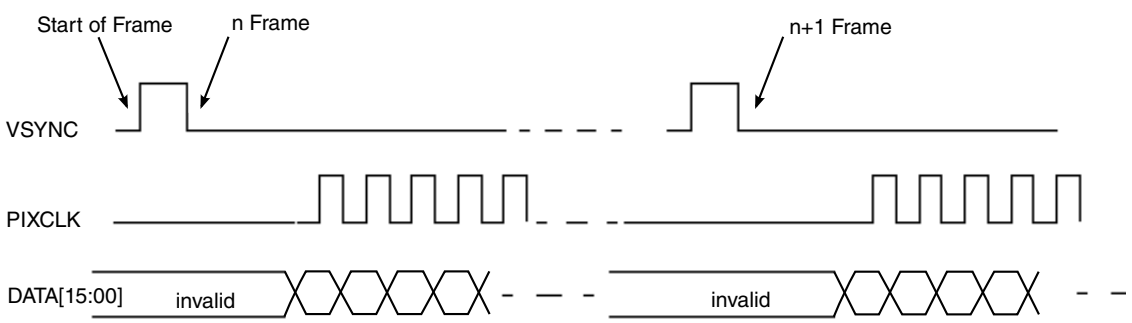


Figure 13-65. Non-Gated Clock Mode Timing Diagram

The overall timing of non-gated mode is the same as the gated-clock mode, except for the HSYNC signal. HSYNC signal is ignored by the CSI. All incoming pixel clocks are valid and cause data to be latched into RxFIFO. The PIXCLK signal is inactive (states low) until valid data is ready to be transmitted over the bus.

Figure 13-65 shows the timing of a typical sensor. Other sensors may have the slightly different timing from that shown. The CSI can be programed to support rising/falling-edge triggered VSYNC, active-high/low HSYNC, and rising/falling-edge triggered PIXCLK.

13.7.3.4 CCIR656 Interlace Mode

In CCIR656 interlace mode, only the PIXCLK and CSI_DATA[9:2] signals are used. The start of frame and blank signals are replaced by a timing codec which is embedded in the data stream. Each active line starts with an Start of Active Video (SAV) code and ends with an End of Active Video (EAV) code. In some cases, digital blanking is inserted in between EAV and SAV code. The CSI decodes and filters out the timing-coding from the data stream, recovering VSYNC and HSYNC signals for internal use, such as statistical block control. Data is forwarded to the data receive and packing block in a sequential manner without reordering-that is, field 1 followed by field 2. The fields must be reordered in software to get back the original image.

Change of Field (COF) interrupt is triggered upon every field change. The interrupt service routine reads the status register to check for the current field.

According to the CCIR656 specification, the image must be in 625/50 PAL or 525/60 NTSC format. In addition, the image is interlaced into odd and even fields with vertical and horizontal blank data being filled into certain lines. Data must be in YCbCr422 format, each pixel contains 2 bytes, either Y + Cr or Y + Cb. These requirements are set for TV systems. The CSI module supports PAL and NTSC format only.

The following figure describes the frame structure in PAL system, showing vertical and horizontal blanking.

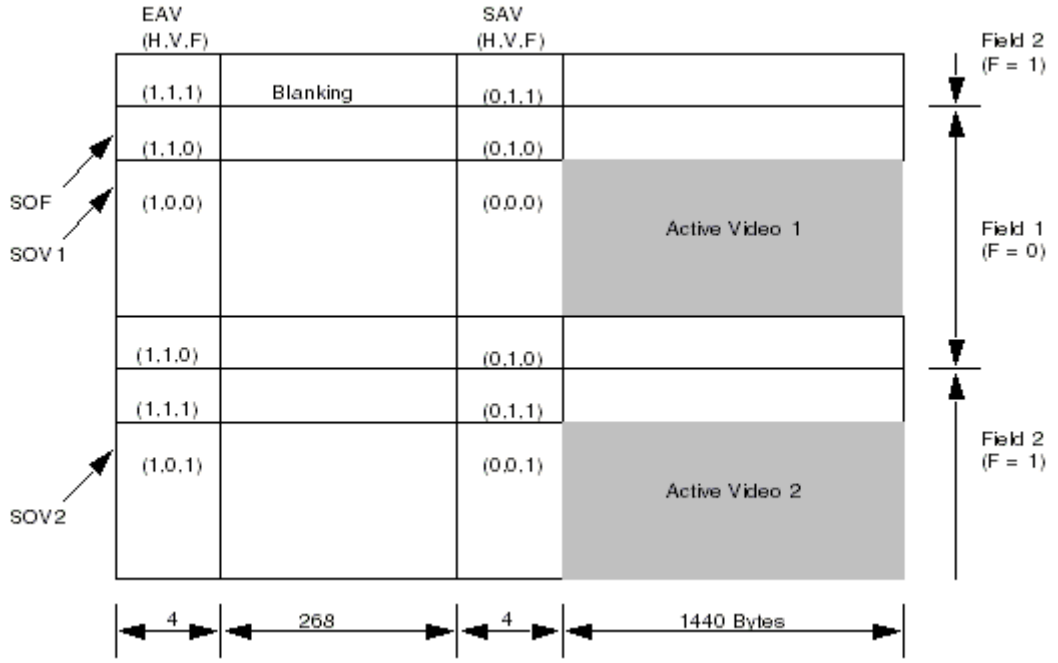


Figure 13-66. CCIR656 Interlace Mode (PAL)

The following figure describes the general timing for a single line, showing SAV and EAV.

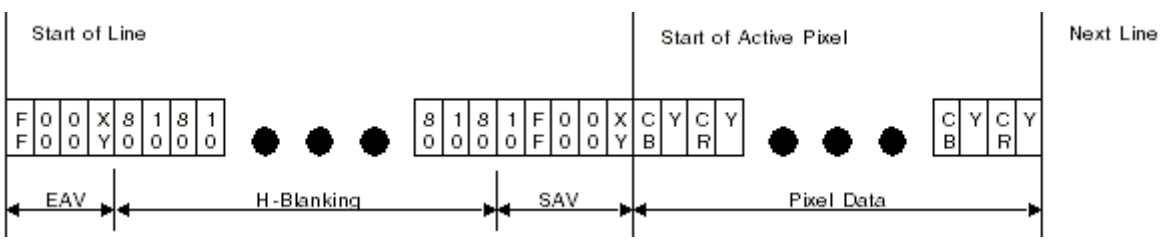


Figure 13-67. CCIR656 General Line Timing

The coding tables recommended by the CCIR656 specification are shown below. It is used in the CCIR656 mode to decode the video stream. An interrupt is generated for SOF, which is decoded from the embedded timing codec.

Table 13-32. Coding for SAV and EAV

Data Bit Number	1st Byte 0xFF	2nd Byte 0x00	3rd Byte 0x00	4th Byte 0xXY
7 (MSB)	1	0	0	1
6	1	0	0	F
5	1	0	0	V
4	1	0	0	H
3	1	0	0	P3
2	1	0	0	P2

Table continues on the next page...

Table 13-32. Coding for SAV and EAV (continued)

Data Bit Number	1st Byte 0xFF	2nd Byte 0x00	3rd Byte 0x00	4th Byte 0xXY
1	1	0	0	P1
0	1	0	0	P0

Table 13-33. Codes with Protection bits for Error Detection/Correction

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

Table 13-34. Representations by F-Bit

F-Bit	Representations
0	ODD FIELD (FIELD 1)
1	EVEN FIELD (FIELD 2)

13.7.3.5 CCIR656 Progressive Mode

For a CMOS camera system of VGA or CIF resolution, strict adherence to the interlace requirements stated in the CIR standard is not required.

The image is considered to have only 1 active field which is scanned in a progressive manner. This active field is regarded as field 1 and the F-bit in the timing codec is ignored by the decoder. Most sensors support CCIR timing in this mode (progressive) by default.

The following figure shows the typical flow of progressive mode.

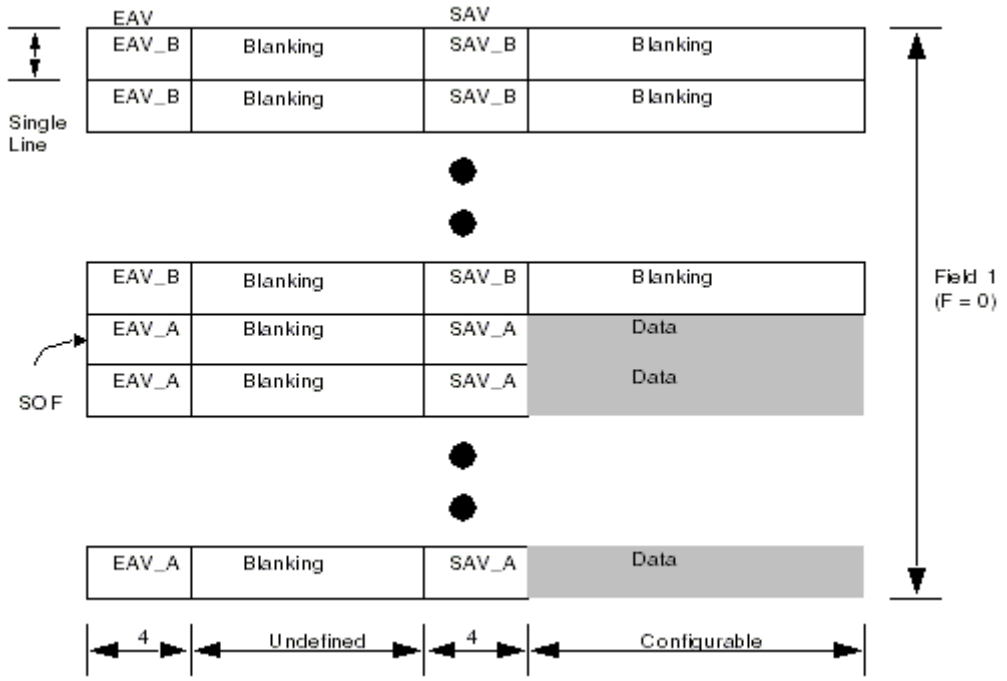


Figure 13-68. CCIR656 Progressive Mode (General Case)

An interrupt is generated for SOF but not for COF. In the general case, when SOF information is retrieved from the embedded coding, it is known as internal VSYNC mode. In other cases, when the VSYNC signal is provided by the sensor, it is known as external VSYNC mode. The CSI can be operated in internal or external VSYNC mode.

13.7.3.6 Error Correction for CCIR656 Coding

According to the algorithm for CCIR coding, protection bits in the SAV and EAV are encoded in the way that allows a 1-bit error to be corrected, or a 2-bit error to be detected by the decoder. This feature is supported by the interlace mode CCIR decoder in CSI.

For the 1-bit error case, users can select the error to be corrected automatically, or simply shown as a status flag instead. For the 2-bit error case, because the decoder is unable to make a correction, the error would be shown as a status flag only.

An interrupt can be generated upon the detection of an error. This signal can be enabled or disabled without affecting the operation of the status bit.

13.7.3.7 Deinterlacer

Deinterlacing is the process of converting interlaced video, such as CCIR656 input, into a non-interlaced form.

The CSI uses the weaving method to do deinterlacing. Weaving is done by adding consecutive fields together. CSI uses top-field detection function. No matter input is NTSC or PAL mode, the combined frame will always put the top-field first and then bottom-field.

13.7.4 Interrupt Generation

The information found here describes CSI events that generate interrupts.

13.7.4.1 Start Of Frame Interrupt (SOF_INT)

The source of an SOF interrupt is dependent on the mode of operation.

In traditional mode, VSYNC signal is taken from sensor and SOF_INT is generated at the rising or falling edge (programmable) of VSYNC.

In CCIR interlace mode, the SOF interrupt information is retrieved from the embedded coding and SOF_INT is generated.

In CCIR progressive mode, there are two sources of an SOF interrupt:

- In *internal* VSYNC mode, SOF is retrieved from the embedded coding.
- In *external* VSYNC mode, VSYNC is taken from the sensor and SOF is generated at the rising edge of VSYNC.

13.7.4.2 End Of Frame Interrupt (EOF_INT)

An EOF interrupt is generated when the frame ends and the complete frame data in RXFIFO is read.

The EOF event triggering works with the RX count register (CSIRXCNT). Software sets the RX count register to the frame size (in words). The CSI RX logic then counts the number of pixel data being received and compares it with the RX count. If the preset value is reached, an EOF interrupt is generated and the data in the RXFIFO are read. If a SOF event is detected before this happens, the EOF interrupt is not generated.

13.7.4.3 Change Of Field Interrupt (COF_INT)

The Change of Field interrupt is only valid in CCIR Interlace mode. The COF interrupt is generated when the field toggles, either from field 1 to field 2, or field 2 to field 1.

Software should first check COF_INT bit in the CSI Status Register (CSISTAT) before checking that F1_INT or F2_INT is turned on.

In PAL systems, the field changes at the beginning of the frame and coincides with SOF. For the first field, a COF interrupt is not generated, only an SOF. The COF interrupt is generated for the second field.

13.7.4.4 CCIR Error Interrupt (ECC_INT)

The CCIR Error Interrupt is only valid for CCIR Interlace mode. An ECC interrupt is generated when an error is found on the SAV or EAV codes in the incoming stream. When this happens, the ECC_INT status bit is set.

13.7.4.5 RxFIFO Full Interrupt (RxFF_INT)

A RxFIFO full interrupt is generated when the number of data in RXFIFO reaches the water mark defined by RxFF_LEVEL in CSICR3.

13.7.4.6 Statistic FIFO Full Interrupt (STATFF_INT)

A StatFIFO full interrupt is generated when the number of data in STATFIFO reaches the water mark defined by STATFF_LEVEL in CSICR3.

13.7.4.7 RxFIFO Overrun Interrupt (RFF_OR_INT)

A RxFIFO Overrun interrupt is generated when the RxFIFO has 128 words data and more data is being written in.

13.7.4.8 Statistic FIFO Overrun Interrupt (SFF_OR_INT)

A StatFIFO Overrun interrupt is generated when the STATFIFO has 64 words data and more data is being written in.

13.7.4.9 Frame Buffer1 DMA Transfer Done Interrupt (DMA_TSF_DONE_FB1)

A DMA transfer done interrupt of frame buffer1 is generated when one frame of data are transferred from RxFIFO to the frame buffer1 in the external memory.

13.7.4.10 Frame Buffer2 DMA Transfer Done Interrupt (DMA_TSF_DONE_FB2)

A DMA transfer done interrupt of frame buffer2 is generated when one frame of data are transferred from RxFIFO to the frame buffer2 in the external memory.

13.7.4.11 Statistic FIFO DMA Transfer Done Interrupt (DMA_TSF_DONE_SFF)

A StatFIFO DMA transfer done interrupt is generated when all the data are transferred from StatFIFO to the external memory. The transfer size is defined in the STATFIFO DMA Transfer Size Register.

13.7.4.12 AHB Bus Response Error Interrupt (HRESP_ERR_INT)

An AHB Bus response error interrupt is generated when a bus error is detected.

13.7.5 Data Packing Style

Careful attention to endianess is needed given the different port sizes at different stages of the image capture path.

To enable flexible packing of image data before storage in the FIFOs, the CSI module can swap data fields by use of the PACK_DIR and the SWAP16_EN bit in CSI Control Register 1 (CSICR1).

The CSI module accepts 8-bit, 10-bit or 16-bit data from the sensor by configuring PIXEL_BIT bit in CSI Control Register 1 (CSICR1) and TWO_8BIT_SENSOR bit in CSI Control Register3 (CSICR3). The input data is packed according to the setting of PACK_DIR bit. The packed data is stored in the RX FIFO according to the setting of the SWAP16_EN bit.

For 10-bit per pixel data format, each pixel is expanded to 16 bits by appending 6 zeros bits to the most significant bit.

13.7.5.1 RX FIFO Path

13.7.5.1.1 Bayer Data

Bayer data is a type of raw data from the image sensor. This byte-wide data must be converted to the RGB space or YUV space by software. The data path for Bayer data is from the CSI to memory. If the system is in little endian, then the PACK_DIR bit should be set to 0. 8-bit data format from a sensor is packed to 64 bits as P7.P6.P5.P4.P3.P2.P1.P0, where P0 is the pixel coming in time slot 0 (first data) and P3 is the pixel coming in time slot 3 (the last data in the 64-bit word). When the data is addressed as bytes by software, P0 is transferred first, P1 is transferred next, and so on. 10-bit data format is packed to 64 bits as 000000.P3.000000.P2.000000.P1.000000.P0, where P0 is the 10-bit data coming in time slot 0 (first pixel) and P3 is the 10-bit data coming in time slot 3 (fourth pixel). 16-bit data is packed to 64 bits as P7.P6.P5.P4.P3.P2.P1.P0.

13.7.5.1.2 RGB565 Data

RGB565 data is processed data from the image sensor, which can be put directly into the display buffer. The data is 16 bits wide. The data path is from CSI to memory to the display controller. On the sensor side, data must be transmitted as P0 first, followed by P1, and so on. For each pixel, whether the MSB or LSB is sent first depends on the endianness of the sensor. Data is 16 bits wide with the MSB labeled RG, and the LSB labeled GB. P0 is represented as RG0 and GB0.

CSI receives data in one of the following sequence:

- RG0, GB0, RG1, GB1, while RG0 comes out at time slot 0 (first data), and GB1 comes out at time slot 3 (last data)
- GB0, RG0, GB1, RG1

Using the first sequence as an example, and assuming the system is running in little endian, the data is presented as:

- 8-bit data from sensor: RG0, GB0, RG1, GB1, …
- 64-bit data before storage in the CSI RX FIFO (PACK_DIR bit = 1): RG0GB0RG1GB1RG2GB2RG3GB3
- 64-bitdata in CSI RX FIFO (SWAP16_EN bit enabled): RG3GB3RG2GB2RG1GB1RG0GB0
- 64-bit transfer to system memory: RG3GB3RG2GB2RG1GB1RG0GB0
- 16-bit read by display controller: RG0GB0, RG1GB1

13.7.5.1.3 RGB888 Data

This is another kind of processed data from image sensor, which can be used for further image processing directly. Each of the data consist of 8-bit Red, 8-bit Green, and 8-bit Blue data. An example of timing scheme is shown in the following figure.

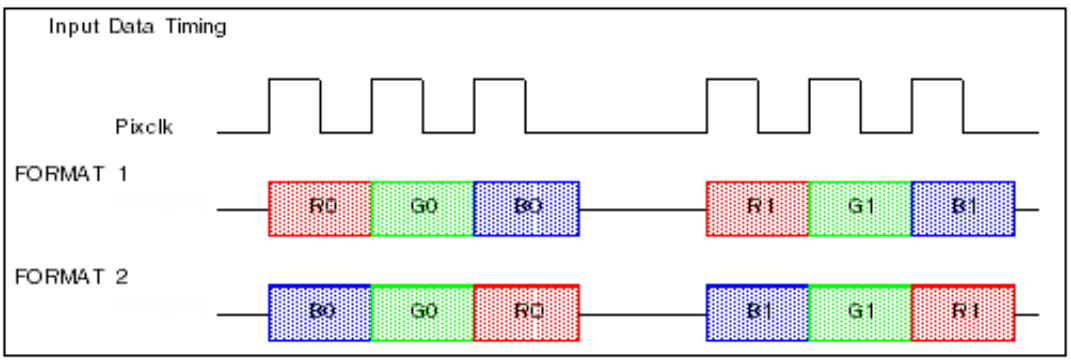
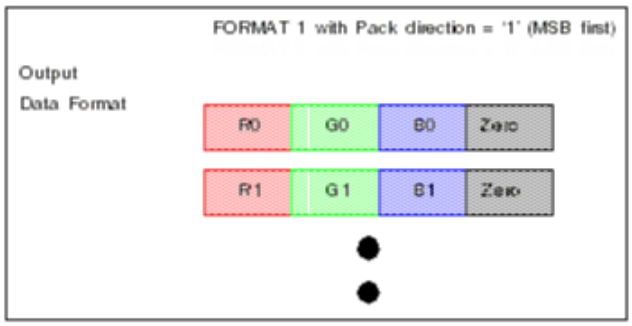


Figure 13-69. Sample Timing Diagram for RGB888 8 bits/cycle Data

An optional scheme to pack a dummy byte is provided. For every group of 3 bytes data, a dummy zero is packed to form a 32-bit word as shown in the following figure. The dummy zero can be packed at the LSB position or MSB position. Using RGB888A_FORMAT_SEL in CSI_CSICR18[18] to determine to put the dummy bytes packed at LSB or MSB position.


FORMAT 1 with Pack direction = '1' (MSB first)
Output
Data Format
R0
G0
B0
Zero
R1
G1
B1
Zero

13.7.5.2 STAT FIFO Path

Statistics only works for Bayer data in 8-bit per pixel format. It generates 16-bit statistical output from the 8-bit Bayer input (CSI_DATA[13:6]). The outputs are Sum of Green (G), Sum of Red (R), Sum of Blue (B), and Auto Focus (F). Each output is 16-bits wide.

The settings of PACK_DIR and SWAP16_EN bits in the CSICR1 register have no effect on the input path. The PACK_DIR only controls how the 16-bit stat output is packed into the 32-bit STAT FIFO.

When the PACK_DIR bit = 1, the stat data is packed as:

First 32-bit: RG

Second 32-bit: BF

…

When the PACK_DIR bit = 0, the stat data is packed as:

First 32-bit GR

Second 32-bit: FB

...

13.7.6 CSI Memory Map/Register Definition

All the 32-bit registers of the CSI module are summarized in the Memory Map below:

CSI memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
0	CSI Control Register 1 (CSI_CSICR1)	32	R/W	4000_0800h	13.7.6.1/ 4962
4	CSI Control Register 2 (CSI_CSICR2)	32	R/W	0000_0000h	13.7.6.2/ 4966
8	CSI Control Register 3 (CSI_CSICR3)	32	R/W	0000_0000h	13.7.6.3/ 4968
C	CSI Statistic FIFO Register (CSI_CSISTATFIFO)	32	R	0000_0000h	13.7.6.4/ 4970

Table continues on the next page...

CSI memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
10	CSI RX FIFO Register (CSI_CSIRFIFO)	32	R	0000_0000h	13.7.6.5/ 4970
14	CSI RX Count Register (CSI_CSIRXCNT)	32	R/W	0000_9600h	13.7.6.6/ 4971
18	CSI Status Register (CSI_CSISR)	32	R/W	0000_4000h	13.7.6.7/ 4972
20	CSI DMA Start Address Register - for STATFIFO (CSI_CSIDMASA_STATFIFO)	32	R/W	0000_0000h	13.7.6.8/ 4975
24	CSI DMA Transfer Size Register - for STATFIFO (CSI_CSIDMATS_STATFIFO)	32	R/W	0000_0000h	13.7.6.9/ 4975
28	CSI DMA Start Address Register - for Frame Buffer1 (CSI_CSIDMASA_FB1)	32	R/W	0000_0000h	13.7.6.10/ 4976
2C	CSI DMA Transfer Size Register - for Frame Buffer2 (CSI_CSIDMASA_FB2)	32	R/W	0000_0000h	13.7.6.11/ 4977
30	CSI Frame Buffer Parameter Register (CSI_CSIFBUF_PARA)	32	R/W	0000_0000h	13.7.6.12/ 4977
34	CSI Image Parameter Register (CSI_CSIIMAG_PARA)	32	R/W	0000_0000h	13.7.6.13/ 4978
48	CSI Control Register 18 (CSI_CSICR18)	32	R/W	0002_D000h	13.7.6.14/ 4979

13.7.6.1 CSI Control Register 1 (CSI_CSICR1)

This register controls the sensor interface timing and interrupt generation. The interrupt enable bits in this register control the interrupt signals and the status bits. That means status bits will only function when the corresponding interrupt bits are enabled.

Address: 0h base + 0h offset = 0h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SWAP16_EN	EXT_VSYNC	EOF_INT_EN	PrP_IF_EN	VIDEO_MODE	COF_INT_EN	SF_OR_INTEN	RF_OR_INTEN	Reserved	SFF_DMA_DONE_INTEN	STATFF_INTEN	FB2_DMA_DONE_INTEN	FB1_DMA_DONE_INTEN	RXFF_INTEN	SOF_POL	SOF_INTEN
Reset	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				HSYNC_POL	CCIR_EN	Reserved	FCC	PACK_DIR	CLR_STATFIFO	CLR_RXFIFO	GCLK_MODE	INV_DATA	INV_PCLK	REDGE	PIXEL_BIT
Reset	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0

CSI_CSICR1 field descriptions

Field	Description
31 SWAP16_EN	SWAP 16-Bit Enable. This bit enables the swapping of 16-bit data. Data is packed from 8-bit or 10-bit to 32-bit first (according to the setting of PACK_DIR) and then swapped as 16-bit words before being put into the RX FIFO. The action of the bit only affects the RX FIFO and has no affect on the STAT FIFO. **NOTE:** Example of swapping enabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x 33441122 **NOTE:** Example of swapping disabled: Data input to FIFO = 0x11223344 Data in RX FIFO = 0x11223344

Table continues on the next page...

CSI_CSICR1 field descriptions (continued)

Field	Description
	0 Disable swapping 1 Enable swapping
30 EXT_VSYNC	External VSYNC Enable. This bit controls the operational VSYNC mode. **NOTE:** This only works when the CSI is in CCIR progressive mode. 0 Internal VSYNC mode 1 External VSYNC mode
29 EOF_INT_EN	End-of-Frame Interrupt Enable. This bit enables and disables the EOF interrupt. 0 EOF interrupt is disabled. 1 EOF interrupt is generated when RX count value is reached.
28 PrP_IF_EN	CSI-PrP Interface Enable. This bit controls the CSI to PrP bus. When enabled the RxFIFO is detached from the AHB bus and connected to PrP. All CPU reads or DMA accesses to the RxFIFO register are ignored. All CSI interrupts are also masked. 0 CSI to PrP bus is disabled 1 CSI to PrP bus is enabled
27 VIDEO_MODE	Video mode select. This bit controls the video mode in CCIR mode. 0 Progressive mode is selected 1 Interlace mode is selected
26 COF_INT_EN	Change Of Image Field (COF) Interrupt Enable. This bit enables the COF interrupt. This bit works only in CCIR interlace mode which is when CCIR_EN = 1 and CCIR_MODE = 1. 0 COF interrupt is disabled 1 COF interrupt is enabled
25 SF_OR_INTEN	STAT FIFO Overrun Interrupt Enable. This bit enables the STATFIFO overrun interrupt. 0 STATFIFO overrun interrupt is disabled 1 STATFIFO overrun interrupt is enabled
24 RF_OR_INTEN	RxFIFO Overrun Interrupt Enable. This bit enables the RX FIFO overrun interrupt. 0 RxFIFO overrun interrupt is disabled 1 RxFIFO overrun interrupt is enabled
23 -	This field is reserved. Reserved. This bit is reserved and should read 0.
22 SFF_DMA_ DONE_INTEN	STATFIFO DMA Transfer Done Interrupt Enable. This bit enables the interrupt of STATFIFO DMA transfer done. 0 STATFIFO DMA Transfer Done interrupt disable 1 STATFIFO DMA Transfer Done interrupt enable
21 STATFF_INTEN	STATFIFO Full Interrupt Enable. This bit enables the STAT FIFO interrupt. 0 STATFIFO full interrupt disable 1 STATFIFO full interrupt enable
20 FB2_DMA_ DONE_INTEN	Frame Buffer2 DMA Transfer Done Interrupt Enable. This bit enables the interrupt of Frame Buffer2 DMA transfer done.

Table continues on the next page...

CSI_CSICR1 field descriptions (continued)

Field	Description
	0 Frame Buffer2 DMA Transfer Done interrupt disable 1 Frame Buffer2 DMA Transfer Done interrupt enable
19 FB1_DMA_DONE_INTEN	Frame Buffer1 DMA Transfer Done Interrupt Enable. This bit enables the interrupt of Frame Buffer1 DMA transfer done. 0 Frame Buffer1 DMA Transfer Done interrupt disable 1 Frame Buffer1 DMA Transfer Done interrupt enable
18 RXFF_INTEN	RxFIFO Full Interrupt Enable. This bit enables the RxFIFO full interrupt. 0 RxFIFO full interrupt disable 1 RxFIFO full interrupt enable
17 SOF_POL	SOF Interrupt Polarity. This bit controls the condition that generates an SOF interrupt. 0 SOF interrupt is generated on SOF falling edge 1 SOF interrupt is generated on SOF rising edge
16 SOF_INTEN	Start Of Frame (SOF) Interrupt Enable. This bit enables the SOF interrupt. 0 SOF interrupt disable 1 SOF interrupt enable
15–12 Reserved	This field is reserved. Reserved. This field is reserved.
11 HSYNC_POL	HSYNC Polarity Select. This bit controls the polarity of HSYNC. This bit only works in gated-clock-that is, GCLK_MODE = 1 and CCIR_EN = 0. 0 HSYNC is active low 1 HSYNC is active high
10 CCIR_EN	CCIR656 Interface Enable. This bit selects the type of interface used. When the CCIR656 timing decoder is enabled, it replaces the function of timing interface logic. 0 Traditional interface is selected. Timing interface logic is used to latch data. 1 CCIR656 interface is selected.
9 Reserved	This field is reserved. This field is reserved.
8 FCC	FIFO Clear Control. This bit determines how the RXFIFO and STATFIFO are cleared. When Synchronous FIFO clear is selected the RXFIFO and STATFIFO are cleared, and STAT block is reset, on every SOF. FIFOs and STAT block restarts immediately after reset. For information on the operation when Asynchronous FIFO clear is selected, refer to the descriptions for the CLR_RXFIFO and CLR_STATFIFO bits. 0 Asynchronous FIFO clear is selected. 1 Synchronous FIFO clear is selected.
7 PACK_DIR	Data Packing Direction. This bit Controls how 8-bit/10-bit image data is packed into 32-bit RX FIFO, and how 16-bit statistical data is packed into 32-bit STAT FIFO. 0 Pack from LSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x44332211 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xBBBBAAAA in STAT FIFO. 1 Pack from MSB first. For image data, 0x11, 0x22, 0x33, 0x44, it will appear as 0x11223344 in RX FIFO. For stat data, 0xAAAA, 0xBBBB, it will appear as 0xAAAABBBB in STAT FIFO.

Table continues on the next page...

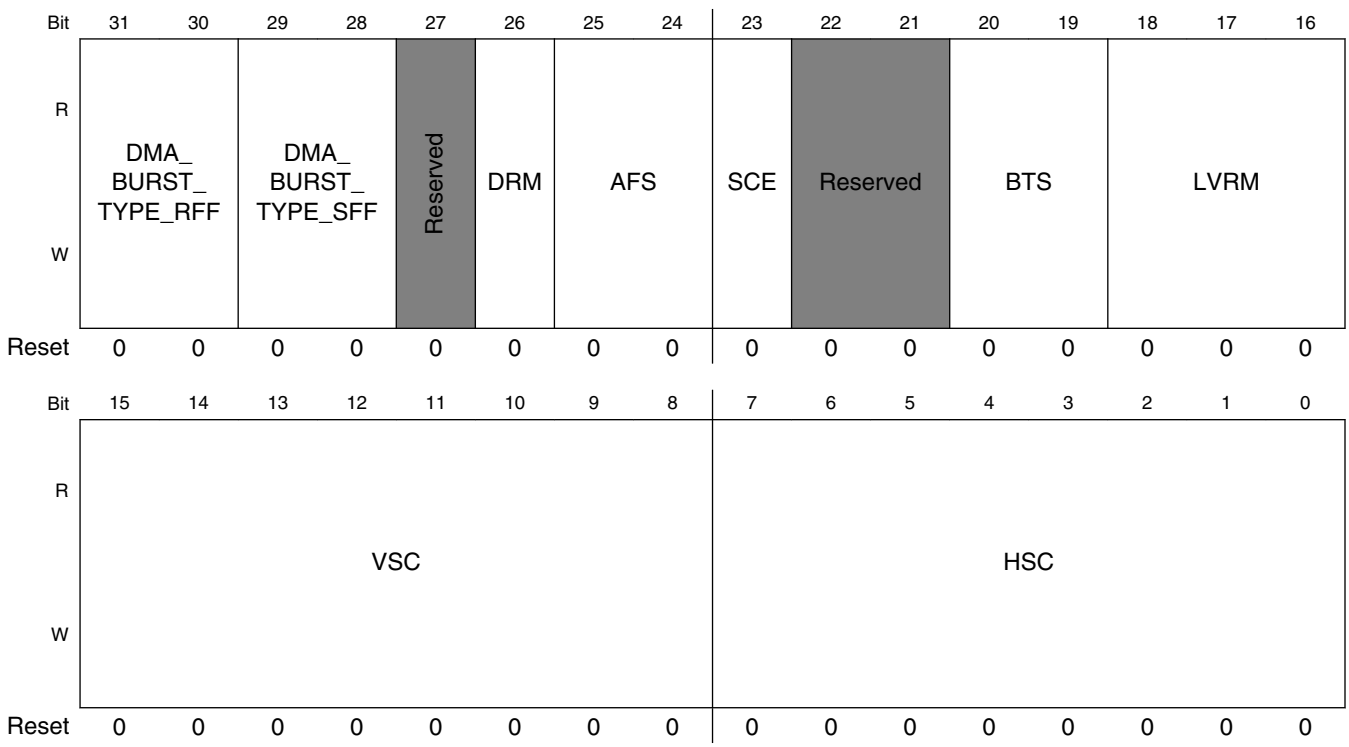
CSI_CSICR1 field descriptions (continued)

Field	Description
6 CLR_STATFIFO	Asynchronous STATFIFO Clear. This bit clears the STATFIFO and Reset STAT block. This bit works only in async FIFO clear mode-that is, FCC = 0. Otherwise this bit is ignored. Writing 1 will clear STATFIFO and reset STAT block immediately, STATFIFO and STAT block then wait and restart after the arrival of next SOF. The bit is restored to 0 automatically after finish. Normally reads 0.
5 CLR_RXFIFO	Asynchronous RXFIFO Clear. This bit clears the RXFIFO. This bit works only in async FIFO clear mode-that is, FCC = 0. Otherwise this bit is ignored. Writing 1 clears the RXFIFO immediately, RXFIFO restarts immediately after that. The bit is restored to 0 automatically after finish. Normally reads 0.
4 GCLK_MODE	Gated Clock Mode Enable. Controls if CSI is working in gated or non-gated mode. This bit works only in traditional mode-that is, CCIR_EN = 0. Otherwise this bit is ignored. 0 Non-gated clock mode. All incoming pixel clocks are valid. HSYNC is ignored. 1 Gated clock mode. Pixel clock signal is valid only when HSYNC is active.
3 INV_DATA	Invert Data Input. This bit enables or disables internal inverters on the data lines. 0 CSI_D[7:0] data lines are directly applied to internal circuitry 1 CSI_D[7:0] data lines are inverted before applied to internal circuitry
2 INV_PCLK	Invert Pixel Clock Input. This bit determines if the Pixel Clock (CSI_PIXCLK) is inverted before it is applied to the CSI module. 0 CSI_PIXCLK is directly applied to internal circuitry 1 CSI_PIXCLK is inverted before applied to internal circuitry
1 REDGE	Valid Pixel Clock Edge Select. Selects which edge of the CSI_PIXCLK is used to latch the pixel data. 0 Pixel data is latched at the falling edge of CSI_PIXCLK 1 Pixel data is latched at the rising edge of CSI_PIXCLK
0 PIXEL_BIT	Pixel Bit. This bit indicates the bayer data width for each pixel. This bit should be configured before activating or re-starting the embedded DMA controller. 0 8-bit data for each pixel 1 10-bit data for each pixel

13.7.6.2 CSI Control Register 2 (CSI_CSICR2)

This register provides the statistic block with data about which live view resolution is being used, and the starting sensor pixel of the Bayer pattern. It also contains the horizontal and vertical count used to determine the number of pixels to skip between the 64 x 64 blocks of statistics when generating statistics on live view image that are greater than 512 x 384.

Address: 0h base + 4h offset = 4h



CSI_CSICR2 field descriptions

Field	Description
31–30 DMA_BURST_ TYPE_RFF	Burst Type of DMA Transfer from RxFIFO. Selects the burst type of DMA transfer from RxFIFO. X0 INCR8 01 INCR4 11 INCR16
29–28 DMA_BURST_ TYPE_SFF	Burst Type of DMA Transfer from STATFIFO. Selects the burst type of DMA transfer from STATFIFO. X0 INCR8 01 INCR4 11 INCR16

Table continues on the next page...

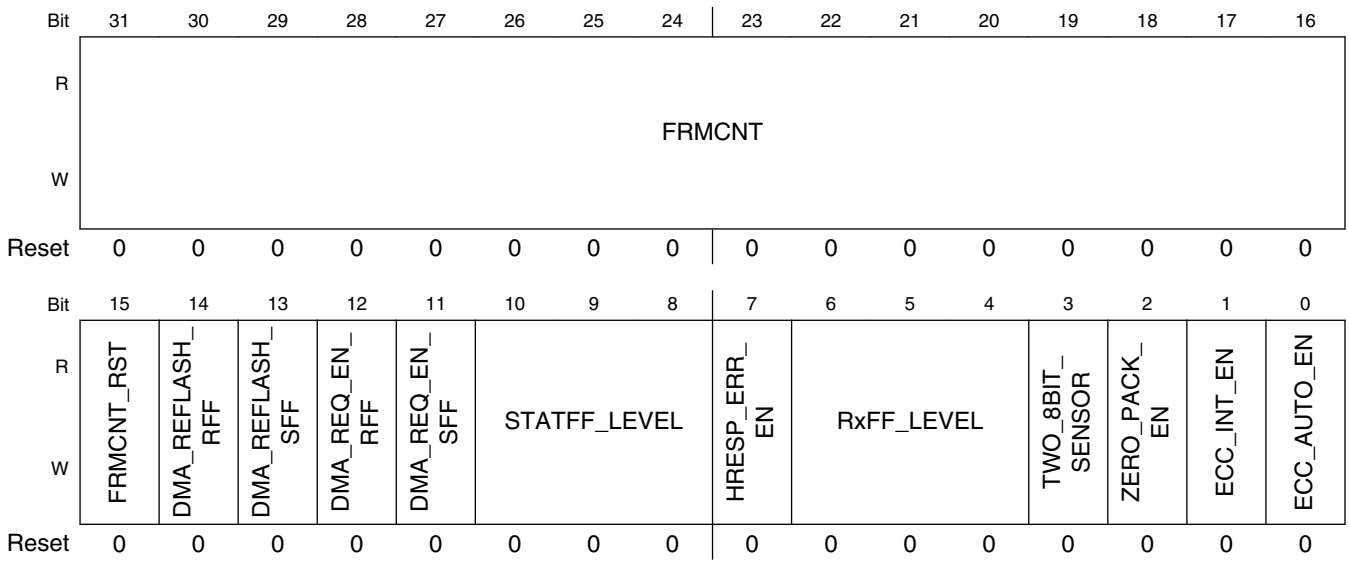
CSI_CSICR2 field descriptions (continued)

Field	Description
27 -	This field is reserved. Reserved. These bit is reserved and should read 0.
26 DRM	Double Resolution Mode. Controls size of statistics grid. 0 Stats grid of 8 x 6 1 Stats grid of 8 x 12
25–24 AFS	Auto Focus Spread. Selects which green pixels are used for auto-focus. 00 Abs Diff on consecutive green pixels 01 Abs Diff on every third green pixels 1x Abs Diff on every four green pixels
23 SCE	Skip Count Enable. Enables or disables the skip count feature. 0 Skip count disable 1 Skip count enable
22–21 -	This field is reserved. Reserved. These bits are reserved and should read 0.
20–19 BTS	Bayer Tile Start. Controls the Bayer pattern starting point. 00 GR 01 RG 10 BG 11 GB
18–16 LVRM	Live View Resolution Mode. Selects the grid size used for live view resolution. 0 512 x 384 1 448 x 336 2 384 x 288 3 384 x 256 4 320 x 240 5 288 x 216 6 400 x 300
15–8 VSC	Vertical Skip Count. Contains the number of rows to skip. SCE must be 1, otherwise VSC is ignored. 0-255 Number of rows to skip minus 1
HSC	Horizontal Skip Count. Contains the number of pixels to skip. SCE must be 1, otherwise HSC is ignored. 0-255 Number of pixels to skip minus 1

13.7.6.3 CSI Control Register 3 (CSI_CSICR3)

This read/write register acts as an extension of the functionality of the CSI Control register 1, adding additional control and features.

Address: 0h base + 8h offset = 8h



CSI_CSICR3 field descriptions

Field	Description
31–16 FRMCNT	Frame Counter. This is a 16-bit Frame Counter (Wraps around automatically after reaching the maximum)
15 FRMCNT_RST	Frame Count Reset. Resets the Frame Counter. (Cleared automatically after reset is done) 0 Do not reset 1 Reset frame counter immediately
14 DMA_REFLASH_RFF	Reflash DMA Controller for RxFIFO. This bit reflash the embedded DMA controller for RxFIFO. It should be reflashed before the embedded DMA controller starts to work. (Cleared automatically after reflashing is done) 0 No reflashing 1 Reflash the embedded DMA controller
13 DMA_REFLASH_SFF	Reflash DMA Controller for STATFIFO. This bit reflash the embedded DMA controller for STATFIFO. It should be reflashed before the embedded DMA controller starts to work. (Cleared automatically after reflashing is done) 0 No reflashing 1 Reflash the embedded DMA controller

Table continues on the next page...

CSI_CSICR3 field descriptions (continued)

Field	Description
12 DMA_REQ_EN_RFF	DMA Request Enable for RxFIFO. This bit enables the dma request from RxFIFO to the embedded DMA controller. 0 Disable the dma request 1 Enable the dma request
11 DMA_REQ_EN_SFF	DMA Request Enable for STATFIFO. This bit enables the dma request from STATFIFO to the embedded DMA controller. 0 Disable the dma request 1 Enable the dma request
10–8 STATFF_LEVEL	STATFIFO Full Level. When the number of data in STATFIFO reach this level, STATFIFO full interrupt is generated, or STATFIFO DMA request is sent. 000 4 Double words 001 8 Double words 010 12 Double words 011 16 Double words 100 24 Double words 101 32 Double words 110 48 Double words 111 64 Double words
7 HRESP_ERR_EN	Hresponse Error Enable. This bit enables the hresponse error interrupt. 0 Disable hresponse error interrupt 1 Enable hresponse error interrupt
6–4 RxFF_LEVEL	**RxFIFO Full Level**. When the number of data in RxFIFO reaches this level, a RxFIFO full interrupt is generated, or an RXFIFO DMA request is sent. 000 4 Double words 001 8 Double words 010 16 Double words 011 24 Double words 100 32 Double words 101 48 Double words 110 64 Double words 111 96 Double words
3 TWO_8BIT_SENSOR	16-bit Sensor Mode. This bit indicates one 16-bit sensor connected to the 16-bit data ports. This bit should be set if there is one 16-bit sensor connected. This bit should be configured before activating or restarting the embedded DMA controller. 0 Only one 8-bit sensor is connected. 1 One 16-bit sensor is connected.
2 ZERO_PACK_EN	Dummy Zero Packing Enable. This bit causes a dummy zero to be packed with every 3 incoming bytes, forming a 32-bit word. The dummy zero is always packed to the LSB position. This packing function is only available in 8-bit/pixel mode. 0 Zero packing disabled 1 Zero packing enabled

Table continues on the next page...

CSI_CSICR3 field descriptions (continued)

Field	Description
1 ECC_INT_EN	Error Detection Interrupt Enable. This bit enables and disables the error detection interrupt. This feature only works in CCIR interlace mode. 0 No interrupt is generated when error is detected. Only the status bit ECC_INT is set. 1 Interrupt is generated when error is detected.
0 ECC_AUTO_EN	Automatic Error Correction Enable. This bit enables and disables the automatic error correction. If an error occurs and error correction is disabled only the ECC_INT status bit is set. This feature only works in CCIR interlace mode. 0 Auto Error correction is disabled. 1 Auto Error correction is enabled.

13.7.6.4 CSI Statistic FIFO Register (CSI_CSISTATFIFO)

The StatFIFO is a read-only register containing statistic data from the sensor. Writing to this register has no effect.

Address: 0h base + Ch offset = Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STAT																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSISTATFIFO field descriptions

Field	Description
STAT	Static data from sensor

13.7.6.5 CSI RX FIFO Register (CSI_CSIRFIFO)

This read-only register contains received image data. Writing to this register has no effect.

Address: 0h base + 10h offset = 10h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	IMAGE																															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIRFIFO field descriptions

Field	Description
IMAGE	Received image data

13.7.6.6 CSI RX Count Register (CSI_CSIRXCNT)

This register works for EOF interrupt generation. It should be set to the number of words to receive that would generate an EOF interrupt.

There is an internal counter that counts the number of words read from the RX FIFO. Whenever the RX FIFO is being read, by either the CPU or the embedded DMA controller, the counter value is updated and compared with this register. If the values match, then an EOF interrupt is triggered.

Address: 0h base + 14h offset = 14h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved										RXCNT																					
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0

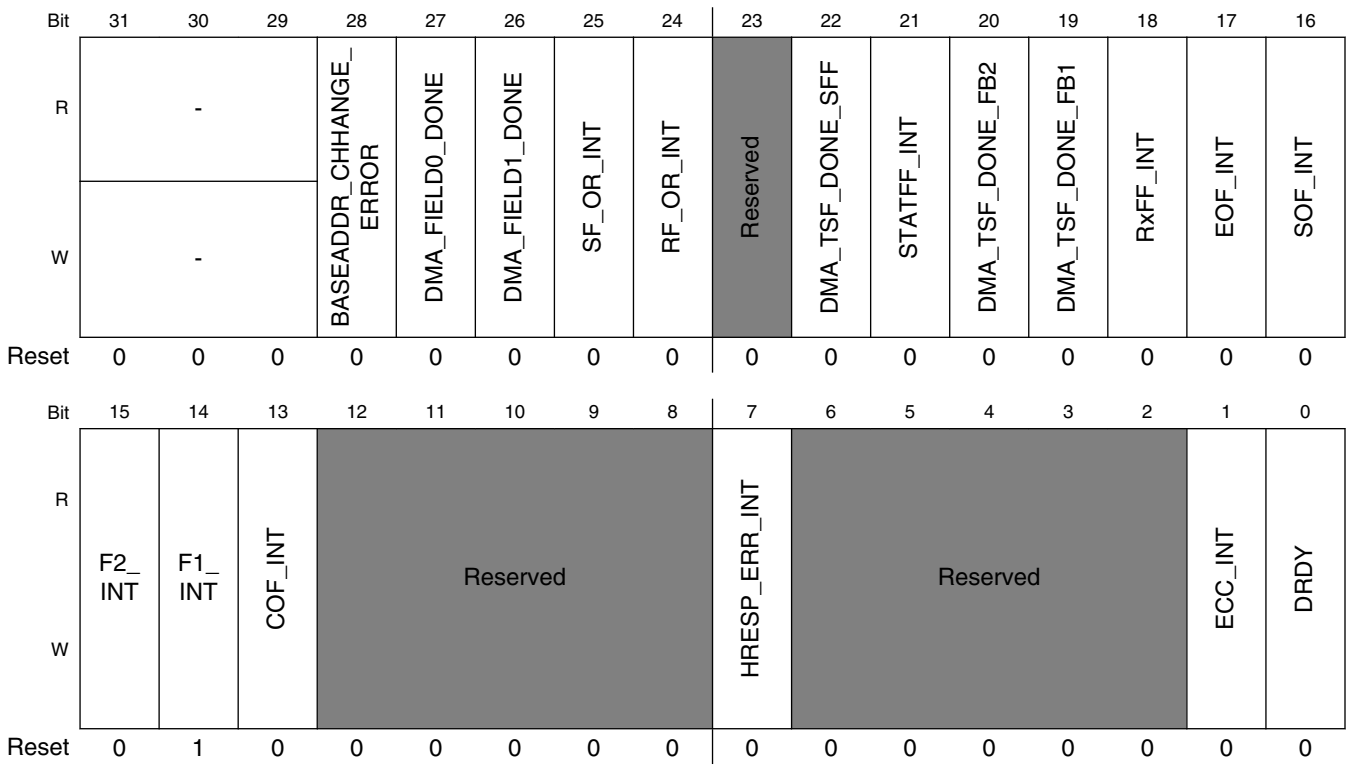
CSI_CSIRXCNT field descriptions

Field	Description
31–22 -	This field is reserved. Reserved. These bits are reserved and should read 0.
RXCNT	RxFIFO Count. This 22-bit counter for RXFIFO is updated each time the RXFIFO is read by CPU or DMA.This counter should be set to the expected number of words to receive that would generate an EOF interrupt.

13.7.6.7 CSI Status Register (CSI_CSISR)

This read/write register shows sensor interface status, and which kind of interrupt is being generated. The corresponding interrupt bits must be set for the status bit to function. Status bits should function normally even if the corresponding interrupt enable bits are not enabled.

Address: 0h base + 18h offset = 18h



CSI_CSISR field descriptions

Field	Description
31–29 -	Reserved. These bits are reserved and should read 0.
28 BASEADDR_CHHANGE_ERROR	When using base address switching enable, this bit will be 1 when switching occur before DMA complete. This bit will be clear by writing 1. When this interrupt happens, follow the steps listed below. 1. Unassert the CSI enable, CSIx_CSICR18 bit31, 2. Reflash the DMA, assert the CSIX_CSICR3 bit 14, 3. Assert the CSI enable, CSIx_CSICR18 bit31.
27 DMA_FIELD0_DONE	When DMA field 0 is complete, this bit will be set to 1(clear by writing 1).

Table continues on the next page...

CSI_CSISR field descriptions (continued)

Field	Description
26 DMA_FIELD1_ DONE	When DMA field 0 is complete, this bit will be set to 1(clear by writing 1).
25 SF_OR_INT	STATFIFO Overrun Interrupt Status. Indicates the overflow status of the STATFIFO register. (Cleared by writing 1) 0 STATFIFO has not overflowed. 1 STATFIFO has overflowed.
24 RF_OR_INT	RxFIFO Overrun Interrupt Status. Indicates the overflow status of the RxFIFO register. (Cleared by writing 1) 0 RXFIFO has not overflowed. 1 RXFIFO has overflowed.
23 -	This field is reserved. Reserved. This bit is reserved and should read 0.
22 DMA_TSF_ DONE_SFF	DMA Transfer Done from StatFIFO. Indicates that the dma transfer from StatFIFO is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by writting 1 or reflashing the StatFIFO dma controller in CSICR3.(Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
21 STATFF_INT	STATFIFO Full Interrupt Status. Indicates the number of data in the STATFIFO reaches the trigger level. (this bit is cleared automatically by reading the STATFIFO) 0 STATFIFO is not full. 1 STATFIFO is full.
20 DMA_TSF_ DONE_FB2	DMA Transfer Done in Frame Buffer2. Indicates that the DMA transfer from RxFIFO to Frame Buffer2 is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by by writting 1 or reflashing the RxFIFO dma controller in CSICR3. (Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
19 DMA_TSF_ DONE_FB1	DMA Transfer Done in Frame Buffer1. Indicates that the DMA transfer from RxFIFO to Frame Buffer1 is completed. It can trigger an interrupt if the corresponding enable bit is set in CSICR1. This bit can be cleared by by writting 1 or reflashing the RxFIFO dma controller in CSICR3. (Cleared by writing 1) 0 DMA transfer is not completed. 1 DMA transfer is completed.
18 RxFF_INT	RXFIFO Full Interrupt Status. Indicates the number of data in the RxFIFO reaches the trigger level. (this bit is cleared automatically by reading the RxFIFO) 0 RxFIFO is not full. 1 RxFIFO is full.
17 EOF_INT	End of Frame (EOF) Interrupt Status. Indicates when EOF is detected. (Cleared by writing 1) 0 EOF is not detected. 1 EOF is detected.
16 SOF_INT	Start of Frame Interrupt Status. Indicates when SOF is detected. (Cleared by writing 1)

Table continues on the next page...

CSI_CSISR field descriptions (continued)

Field	Description
	0 SOF is not detected. 1 SOF is detected.
15 F2_INT	CCIR Field 2 Interrupt Status. Indicates the presence of field 2 of video in CCIR mode. (Cleared automatically when current field does not match) **NOTE:** Only works in CCIR Interlace mode. 0 Field 2 of video is not detected 1 Field 2 of video is about to start
14 F1_INT	CCIR Field 1 Interrupt Status. Indicates the presence of field 1 of video in CCIR mode. (Cleared automatically when current field does not match) **NOTE:** Only works in CCIR Interlace mode. 0 Field 1 of video is not detected. 1 Field 1 of video is about to start.
13 COF_INT	Change Of Field Interrupt Status. Indicates that a change of the video field has been detected. Only works in CCIR Interlace mode. Software should read this bit first and then dispatch the new field from F1_INT and F2_INT. (Cleared by writing 1) 0 Video field has no change. 1 Change of video field is detected.
12–8 -	This field is reserved. Reserved. These bits are reserved and should read 0.
7 HRESP_ERR_INT	Hresponse Error Interrupt Status. Indicates that a hresponse error has been detected. (Cleared by writing 1) 0 No hresponse error. 1 Hresponse error is detected.
6–2 -	This field is reserved. Reserved. These bits are reserved and should read 0.
1 ECC_INT	CCIR Error Interrupt. This bit indicates an error has occurred. This only works in CCIR Interlace mode. (Cleared by writing 1) 0 No error detected 1 Error is detected in CCIR coding
0 DRDY	RXFIFO Data Ready. Indicates the presence of data that is ready for transfer in the RxFIFO. (Cleared automatically by reading FIFO) 0 No data (word) is ready 1 At least 1 datum (word) is ready in RXFIFO.

13.7.6.8 CSI DMA Start Address Register - for STATFIFO (CSI_CSIDMASA_STATFIFO)

This register provides the start address for the embedded DMA controller of STATFIFO. The embedded DMA controller will read data from STATFIFO and write it to the external memory from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 20h offset = 20h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_START_ADDR_SFF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_START_ADDR_SFF														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIDMASA_STATFIFO field descriptions

Field	Description
31–2 DMA_START_ ADDR_SFF	DMA Start Address for STATFIFO. Indicates the start address to write data. The embedded DMA controller will read data from STATFIFO and write it from this address through AHB bus. The address should be double words aligned.
-	This field is reserved. Reserved. These bits are reserved and should read 0.

13.7.6.9 CSI DMA Transfer Size Register - for STATFIFO (CSI_CSIDMATS_STATFIFO)

This register provides the total transfer size for the embedded DMA controller of STATFIFO. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 24h offset = 24h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_TSF_SIZE_SFF																															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIDMATS_STATFIFO field descriptions

Field	Description
DMA_TSF_SIZE_SFF	DMA Transfer Size for STATFIFO. Indicates how many words to be transfered by the embedded DMA controller. The size should be double words aligned.

13.7.6.10 CSI DMA Start Address Register - for Frame Buffer1 (CSI_CSIDMASA_FB1)

This register provides the start address in the frame buffer1 for the embedded DMA controller of RxFIFO. The embedded DMA controller will read data from RxFIFO and write it to the frame buffer1 from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 28h offset = 28h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_START_ADDR_FB1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_START_ADDR_FB1														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIDMASA_FB1 field descriptions

Field	Description
31–2 DMA_START_ADDR_FB1	DMA Start Address in Frame Buffer1. Indicates the start address to write data. The embedded DMA controller will read data from RxFIFO and write it from this address through AHB bus. The address should be double words aligned.
-	This field is reserved. Reserved. These bits are reserved and should read 0.

13.7.6.11 CSI DMA Transfer Size Register - for Frame Buffer2 (CSI_CSIDMASA_FB2)

This register provides the start address in the frame buffer2 for the embedded DMA controller of RxFIFO. The embedded DMA controller will read data from RxFIFO and write it to the frame buffer2 from the start address. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 2Ch offset = 2Ch

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DMA_START_ADDR_FB2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DMA_START_ADDR_FB2														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIDMASA_FB2 field descriptions

Field	Description
31–2 DMA_START_ ADDR_FB2	DMA Start Address in Frame Buffer2. Indicates the start address to write data. The embedded DMA controller will read data from RxFIFO and write it from this address through AHB bus. The address should be double words aligned.
-	This field is reserved. Reserved. These bits are reserved and should read 0.

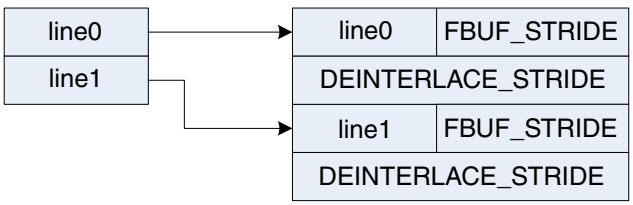
13.7.6.12 CSI Frame Buffer Parameter Register (CSI_CSIFBUF_PARA)

This register provides the stride of the frame buffer to show how many words to skip before starting to write the next row of the image. The width of the frame buffer minus the width of the image is the stride. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 30h offset = 30h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEINTERLACE_STRIDE																FBUF_STRIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

CSI_CSIFBUF_PARA field descriptions

Field	Description
31–16 DEINTERLACE_ STRIDE	DEINTERLACE_STRIDE is only used in the deinterlace mode. If line stride feature is supported in deinterlace mode, FBUF_STRIDE and DEINTERLACE_STRIDE need to be configured at the same time. DEINTERLACE_STRIDE is configured the same as line width. In normal line stride feature, only FBUF_STRIDE needs to be configured. 
FBUF_STRIDE	Frame Buffer Parameter. Indicates the stride of the frame buffer. The width of the frame buffer(in double words) minus the width of the image(in double words) is the stride. The stride should be double words aligned. The embedded DMA controller will skip the stride before starting to write the next row of the image.

13.7.6.13 CSI Image Parameter Register (CSI_CSIIMAG_PARA)

This register provides the width and the height of the image from the sensor. The width and height should be aligned in pixel. The width of the image multiplied by the height is the total pixel size that will be transfered in a frame by the embedded DMA controller. This register should be configured before activating or restarting the embedded DMA controller.

Address: 0h base + 34h offset = 34h

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	IMAGE_WIDTH																IMAGE_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

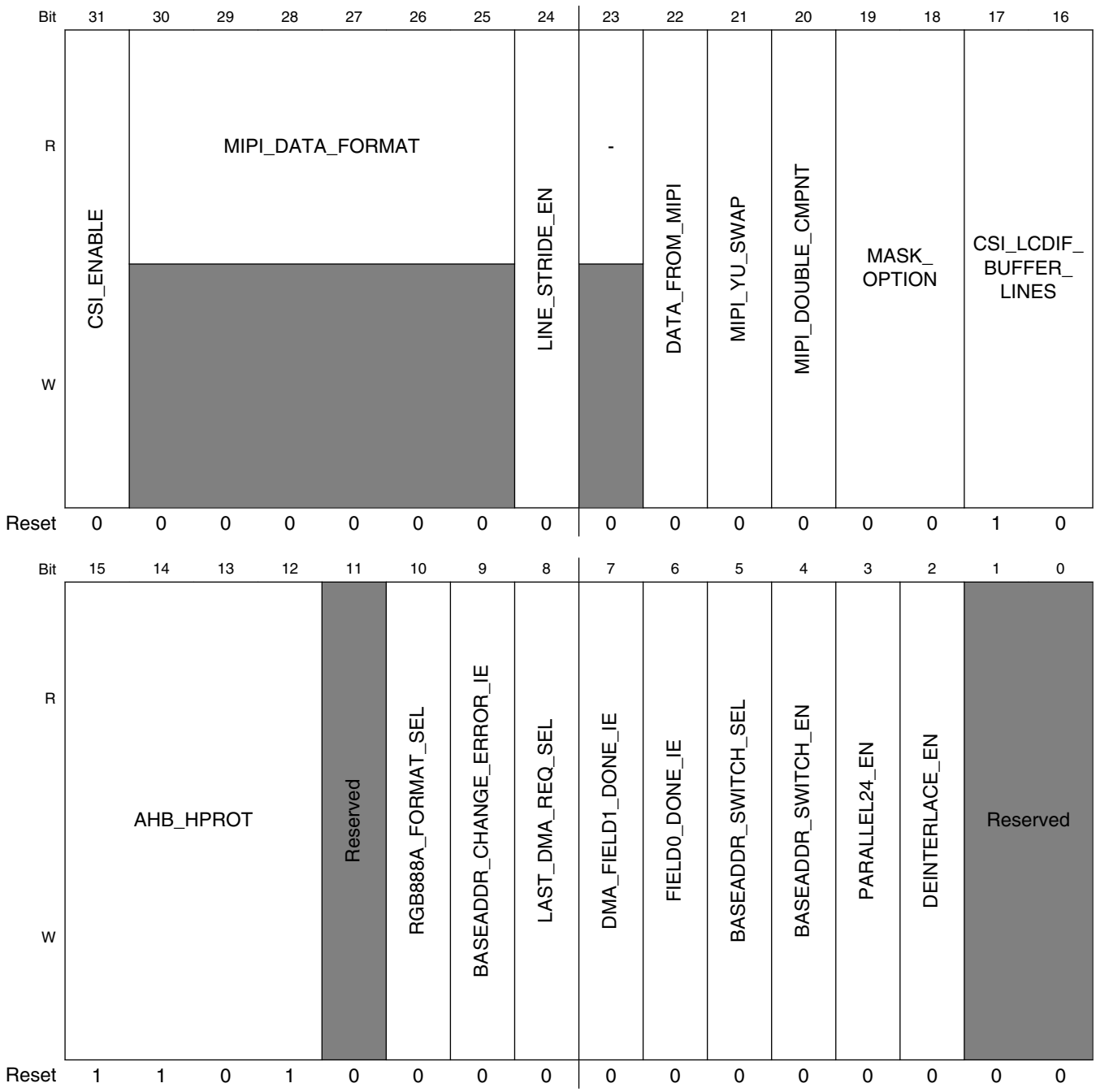
CSI_CSIIMAG_PARA field descriptions

Field	Description
31–16 IMAGE_WIDTH	Image Width. Indicates how many pixels in a line of the image from the sensor. If the input data from the sensor is 8-bit/pixel format, the IMAGE_WIDTH should be a multiple of 8 pixels. If the input data from the sensor is 10-bit/pixel or 16-bit/pixel format, the IMAGE_WIDTH should be a multiple of 4 pixels.
IMAGE_HEIGHT	Image Height. Indicates how many pixels in a column of the image from the sensor.

13.7.6.14 CSI Control Register 18 (CSI_CSICR18)

This read/write register acts as an extension of the functionality of the CSI Control register 1

Address: 0h base + 48h offset = 48h



CSI_CSICR18 field descriptions

Field	Description
31 CSI_ENABLE	CSI global enable signal. Only when this bit is 1, CSI can start to receive the data and store to memory.
30–25 MIPI_DATA_FORMAT	Image Data Format Generic Short Packet: 0x08 ~ 0x0F Embedded 8-bit non Image: 0x12 YUV420 (8-bit) 0x18 YUV420 (10-bit) 0x19 YUV420 (8-bit legacy) 0x1A YUV420 (8-bit CSPS) 0x1C YUV420 (10-bit CSPS) 0x1D YUV422 (8-bit) 0x1E YUV422 (10-bit) 0x1F RGB444 0x20 (Not support) RGB555 0x21 (Not support) RGB565 0x22 RGB666 0x23 RGB888 0x24 RAW6 0x28 RAW7 0x29 RAW8 0x2A RAW10 0x2B RAW12 0x2C RAW14 0x2D User defined 1 0x30 User defined 2 0x31 User defined 3 0x32 User defined 4 0x33 User defined 5 0x34 User defined 6 0x35 User defined 7 0x36 User defined 8 0x37
24 LINE_STRIDE_EN	When the line width are not the multiple of the burst length, assert this bit.
23 -	Reserved
22 DATA_FROM_MIPI	0 Data from parallel sensor 1 Data from MIPI
21 MIPI_YU_SWAP	It only works in MIPI CSI YUV422 double component mode.
20 MIPI_DOUBLE_CMPNT	Double component per clock cycle in YUV422 formats. 0 Single component per clock cycle

Table continues on the next page...

CSI_CSICR18 field descriptions (continued)

Field	Description
	(half pixel per clock cycle) 1 Double component per clock cycle (a pixel per clock cycle)
19–18 MASK_OPTION	These bits used to choose the method to mask the CSI input. 00 Writing to memory from first completely frame, when using this option, the CSI_ENABLE should be 1. 01 Writing to memory when CSI_ENABLE is 1. 02 Writing to memory from second completely frame, when using this option, the CSI_ENABLE should be 1. 03 Writing to memory when data comes in, not matter the CSI_ENABLE is 1 or 0.
17–16 CSI_LCDIF_ BUFFER_LINES	The number of lines are used in handshake mode with LCDIF. 00 4 lines 01 8 lines 02 16 lines 03 16 lines
15–12 AHB_HPROT	Hprot value in AHB bus protocol.
11 -	This field is reserved.
10 RGB888A_ FORMAT_SEL	Output is 32-bit format. 0 {8'h0, data[23:0]} 1 {data[23:0], 8'h0}
9 BASEADDR_ CHANGE_ ERROR_IE	Base address change error interrupt enable signal.
8 LAST_DMA_ REQ_SEL	Choosing the last DMA request condition. 0 fifo_full_level 1 hburst_length
7 DMA_FIELD1_ DONE_IE	When in interlace mode, field 1 done interrupt enable. 0 Interrupt disabled 1 Interrupt enabled
6 FIELD0_DONE_ IE	In interlace mode, fileld 0 means interrupt enabled. 0 Interrupt disabled 1 Interrupt enabled
5 BASEADDR_ SWITCH_SEL	CSI 2 base addresses switching method. When using this bit, BASEADDR_SWITCH_EN is 1. 0 Switching base address at the edge of the vsync 1 Switching base address at the edge of the first data of each frame
4 BASEADDR_ SWITCH_EN	When this bit is enabled, CSI DMA will switch the base address according to BASEADDR_SWITCH_SEL rather than atomically by DMA completed.

Table continues on the next page...

CSI_CSICR18 field descriptions (continued)

Field	Description
3 PARALLEL24_ EN	When input is parallel rgb888/yuv444 24bit, this bit can be enabled.
2 DEINTERLACE_ EN	This bit is used to select the output method When input is standard CCIR656 video. 0 Deinterlace disabled 1 Deinterlace enabled
-	This field is reserved. Reserved.

13.8 MIPI CSI Host Controller (MIPI_CSI)

13.8.1 Overview

This section introduces the MIPI CSI-2 RX subsystem with the CSI-2 RX PHY and host controller. This subsystem handles the sensor/image input and process for all the input imaging devices.

13.8.1.1 Features

The MIPI-CSI2 Controller has the following key features:

- Implements all three CSI-2 MIPI layers (Pixel to byte packing, low level protocol, Lane management)
- Supports unidirectional Master operation
- Transmitter and receiver versions
- Scalable data lane support, 1 to 4 Data Lanes
- Supports high speed mode(80Mbps - 1.5Gbps) per lane, providing 4K@30fps capability for the 4 lanes
- Supports 10Mbps data rate in low power mode
- Includes high speed deserializers
- Loopback testability support
- Support for all CSI-2 data types
- Support for DPHY Ultra Low Power State (ULPS)
- Error collection support (Rx Only)
- Flexible pixel-based user interface
    - Supports user generated packets
    - Supports single interface

- Supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs
  - Delivered fully integrate and verified with target MIPI D-PHY
- RX Video Interface
- APB Control and Status Register (CSR) interface with IRQ support
- Easy configuration and control via core ports

13.8.2 Reset

Reset and Initialization procedure for the CSI-2 Rx Controller is as follows:

1. Assert all resets.
2. Wait until clocks to the Rx Controller are stable and ensure DPHY interface is idle (Stop State on all lanes)
3. De-assert reset to CSR, pclk_reset_n, if CSR is included with the controller. If no CSR go to step 4.
4. Program CSR registers if CSR is present. If CSR is not present, set values to all configuration ports (cfg_* inputs) to appropriate values.
5. De-assert all remaining resets.
6. CSI-2 Rx Controller is ready for use by the next rising edge of clk_ui.

13.8.3 Functional description

The CSI-2 Rx Controller Core implements all three layers defined by the CSI-2 Specification: Pixel to Byte Packing, Low Level Protocol, and Lane Management. The D-PHY interface of the CSI-2 Rx Controller Core supports PHY Protocol Interface (PPI) compatible MIPI D-PHYs. The Local Interface is an easy to use pixel based interface that supports all data types. The Local interface runs at the User Interface clock rate for all implementations. The CSI-2 Rx Controller Core takes care of all packet formatting details and transmission over the MIPI bus. The CSI-2 Rx Controller Core also includes an Rx Video Interface. This interface watches the local interface of the controller and generates a simplified video output interface. Image packet types which are supported by the Pixel Format block are assembled into a Video output bus which carries the pixel data which is received by the controller. Valid video output data is indicated using the data enable output(s). Horizontal and vertical sync outputs are also provided.

13.8.3.1 CSI-2 RX Block diagram

The block diagram for the MIPI CSI-2 RX block is given below:

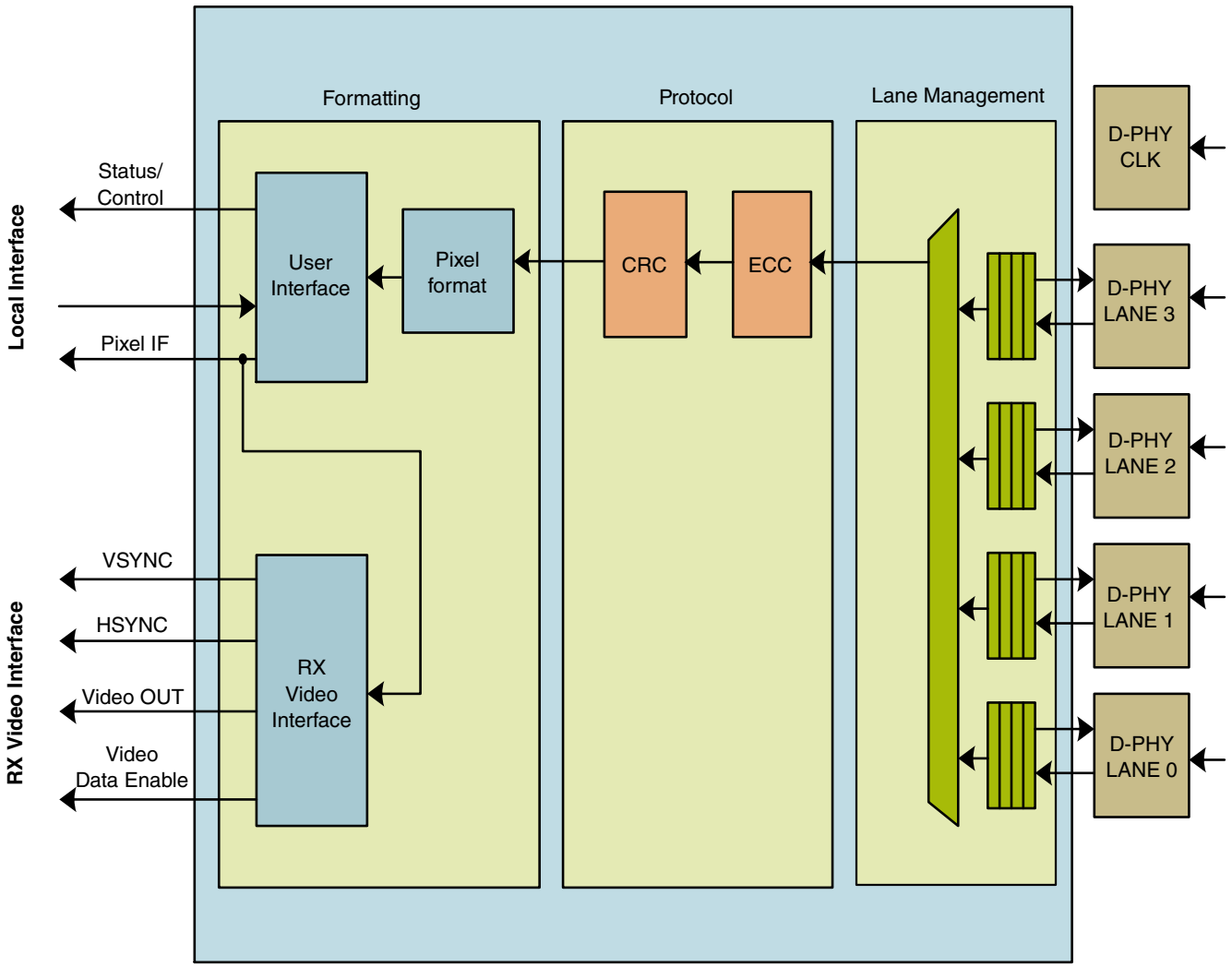


Figure 13-71. MIPI CSI-2 RX Block Diagram

13.8.3.2 Rx Controller Core Configuration Port Descriptions

The CSI-2 Rx Controller Core has configuration ports that can control the configuration of the core in real time. The ports are listed and described in the table below. When the CSR module is included with the CSI-2 RX Controller, the cfg_* ports are removed from the top level and connected to ports on the CSR module, making them accessible through the APB interface.

Table 13-35. Rx Controller Core Configuration Port Descriptions

Port	Type	Description
cfg_num_lanes[1:0]	Input	Sets the number of active lanes that are to be used for receiving MIPI data.

Table continues on the next page...

Table 13-35. Rx Controller Core Configuration Port Descriptions (continued)

		2'b00 –1 Lane 2'b01 – 2 Lanes 2'b10 – 3 Lanes 2'b11 – 4 Lanes
cfg_disable_payload[48:0]	Input	Disables payload data for the selected data type. When a bit is set that represents a supported data type, only the packet header will be presented at the user interface, along with the sop and eop indication where the payload data would have appeared if enabled. [0] – NULL [1] –BLANK [2] –EMBEDDED [9:3] –Reserved [10] – Legacy YUV420 8 bit [13:11] – Reserved [14] – YUV422 8 bit [15] – YUV422 10 bit [16] – RGB444 [17] – RGB555 [18] – RGB565 [19] – RGB666 [20] – RGB888 [23:21] – Reserved [24] –RAW6 [25] –RAW7 [26] –RAW8 [27] –RAW10 [28] –RAW12 [29] –RAW14 [31:30] – Reserved [32] – User defined type 0x30 [33] – User defined type 0x31 [34] – User defined type 0x32 [35] – User defined type 0x33 [36] – User defined type 0x34 [37] – User defined type 0x35 [38] – User defined type 0x36 [39] – User defined type 0x37 [47:40] – Reserved [48] – Unsupported data types
cfg_disable_data_lanes	[3:0]	Setting bits to a'1' value causes the DPHY Enable signal to deassert. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.3.3 CSI-2 RX Controller Core Local Interface

The CSI-2 Rx Controller Core Local Interface consists of a Receive Interface.

Packets are received from the MIPI Interface via the Receive Local Interface. The Receive Interface is designed to easily interface to a FIFO on the user side that holds the pixels that are received until the user application can process them.

13.8.3.3.1 RX Local Interface Description

The Receive Local Interface is the mechanism with which the user receives packets from the MIPI Interface.

The Receive Local Interface first announces that a packet is being or has been received by asserting the pkt_hdr_valid signal with the other packet information output ports containing details about the packet. Next, the sop_out signal will assert indicating that data is about to arrive at the Receive Local Interface. Dav_out will assert with every cycle of pixel data on the data_out port. The last cycle of receive data is signaled by the assertion of eop_out.

With a single pixel interface, data_out contains a single pixel every clock that dav_out (data valid) is asserted. The single pixel interface must run at a frequency that is equal to number of MIPI lanes times the byte lane frequency.

Table 13-36. Receive Local Interface

Port	Type	Description
clk	Input	RX Controller Core Clock input. This clock must be exactly equal to or faster than the receive byteclock, RxByteClkHS_ln0, from the RX DPHY.
clk_ui	Input	User interface clock. The frequency of clk_ui must be such that the data received on the data_out output is greater than or equal to the total bandwidth of the physical MIPI interface. Clk_ui has no relationship requirement with regards to ‘clk’ other than the bandwidth requirement mentioned previously.
clk_esc	Input	Rx Escape Clock. This must be the same escape clock that the RX DPHY receives.
reset_n	Input	Aysnc reset, active low. This reset applies to all logic in the Controller Core that is in the ‘clk’ clock domain.

Table continues on the next page...

Table 13-36. Receive Local Interface (continued)

clk_ui_reset_n	Input	Async reset, active low. This reset applies to all logic in the Controller Core that is in the 'clk_ui' clock domain.
clk_esc_reset_n	Input	Async reset, active low. This reset applies to all logic in the Controller Core that is in the clk_esc domain.
byte_clk_reset_n	Input	Asnyc reset, active low. This reset applies to all logic in the Controller Core that is in the byte clock (RxByteClkHS_lnx clock inputs) that is received from the RX DPHY.
data_out[n-1:0]	Output	Pixel Data output. The Rx Controller can present pixel data in single pixels or in pairs. The width of the data_out port is dependent on the largest pixel size possible and whether the controller is configured for single, double, or quad pixel mode. Pixel data types narrower than 'n' bits are justified down into the least significant bits (ie, bits [7:0] in a 12 bit field [11:0]). When in double pixel mode, the first pixel resides at [n-1:0] while the second pixel received is at [n*2-1:n]. When in quad pixel mode, the first pixel resides at [n-1:0], second pixel received is at [n*2-1:n], the third pixel is at [n*3-1:n*2] and the fourth is at [n*4-1:n*3] where 'n' is the maximum pixel width supported.
dav_out	Output	Data Valid Out. When asserted high, data_out, sop_out, and eop_out outputs are valid.
sop_out	Output	Start of packet signal. This active high signal indicates the first cycle of data on dataout. Valid when dav_out asserts high
eop_out[3:0]	Output	Last pixel indicator. Asserts to indicate the last pixel in a transmission. When in single pixel mode, eop_out[0] is used. When in double pixel mode two bits are used, [1:0]. 2'b01 indicates the last pixel is on the lower pixel path. 2'b10 indicates the last pixel is on the upper pixel path. When in quad pixel mode, all four bits are utilized. A value of 4'b0001 indicates the last pixel is on the lowest pixel path portion of data_out. 4'b0010 indicates the last pixel is on the second pixel path of data_out. 4'b0100 indicates the last pixel is on the third pixel path of data_out. 4'b1000 indicates the last pixel is on the fourth and highest pixel path of data_out. Valid when dav_out asserts high.

Table continues on the next page...

Table 13-36. Receive Local Interface (continued)

pkt_hdr_valid	Output	Packet header data is valid on the packet header ports below when this signal is asserted.
pkt_hdr_data_type[4:0]	Output	Packet data type, valid when pkt_hdr_valid is asserted. See the CSI-2 specification for a definition of possible values.
pkt_hdr_wc[15:0]	Output	Packet Word Count. Number of bytes of pixel data in the received payload for long packet data types. See the MIPI CSI-2 specification for meaning for short packet. The value on pkt_hdr_wc is the Word count field from the received packet header. The RX controller does not modify the value in any way. Valid when pkt_hdr_valid is asserted.
pixel_cnt_out[15:0]	Output	Number of pixels in receive packet. The pixel_cnt_out value is updated one clock after eop_out asserts and represents the packet of pixels there were in the just received packet. The pixel_cnt_out value is only valid for long packets.
ecc_one_bit_error	Output	Single bit error in the packet header was detected and corrected. Active high and is valid when pkt_hdr_valid is high.
ecc_two_bit_error	Output	Two packet header bit errors were detected and not corrected, active high and is valid when pkt_hdr_is high..
ecc_one_bit_error_pos[4:0]	Output	Position of the corrected single bit error in the packet header. Valid when pkt_hdr_valid is asserted high.
ecc_err	Output	Error detected in the ECC bits. This signal is no longer supported.
ecc_err_pos[2:0]	Output	Position of the bit in the ECC bits, valid when ecc_err is asserted. This signal is no longer supported.
crc_err	Output	Asserts high when the CRC calculated on the received data does not match the CRC the transmitter sent at the end of the packet. Valid when the eop_out output asserts.
ulps_active[4:0]	Output	Receive UltraLow Power State active. Bits assert high when corresponding clock or data lane is in ULPS mode. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1 Bit [3] – Data Lane 2 Bit [4] – Data Lane 3

Table continues on the next page...

Table 13-36. Receive Local Interface (continued)

ulps_mark_active[4:0]	Output	Receive UltraLow Power State Mark status. Bits assert high when corresponding clock or data lane has exited ULPS and entered Mark-1 state. Lanes will remain in Mark-1 state for a minimum of 1ms as per the MIPI DPHY specification. While in Mark-1 state, no other active, like High Speed data transmission, is allowed. Bit [0] – Clock Lane Bit [1] – Data Lane 0 Bit [2] – Data Lane 1 Bit [3] – Data Lane 2 Bit [4] – Data Lane 3
rx_enable	Input	Receive Enable, active high. When deasserted, the RX controller will pause data reception at the next packet boundary. Since the CSI-2 protocol does not allow the Receiver to pause data, when rx_enable is deasserted the RX Controller still receives data from the RX DPHY but it does not forward the receive packets to the user interface but instead discards the received data. When rx_enable is asserted, the RX controller will wait for the start of the next packet before allowing data to be sent out over the user interface.

13.8.3.3.2 Receive Interface Example Transaction

High Speed receive packets appear immediately on the Receive local interface as they are received.

13.8.3.3.2.1 Long packet receive with RAW 8 data and one DPHY

Initially, pkt_hdr_valid asserts along with the packet header information. The packet header ports, pkt_hdr_data_type, pkt_hdr_wc, and pkt_hdr_vc indicate that the packet contains RAW8 data, has 4 btes total and a Virtual Channel of zero. Several clocks later sop_out asserts with dav_out indicating the first cycle of receive data. Dav_out asserts for 4 clocks, framing the 4 bytes (pixels) of RAW8 pixel data on data_out[7:0]. During the last cycle of dav_out eop_out[0] asserts indicating the last cycle of received RAW8 data. When the pixel data has a width less than the 24 bit width of the data_out port the data is

justified down into the lower bits. In the example previously described, the RAW8 data will appear on data_out[7:0]. If the data had been RAW10 pixels, the RAW10 data would be contained on data_out[9:0].

The figure below shows a waveform of the received packet on the local interface.

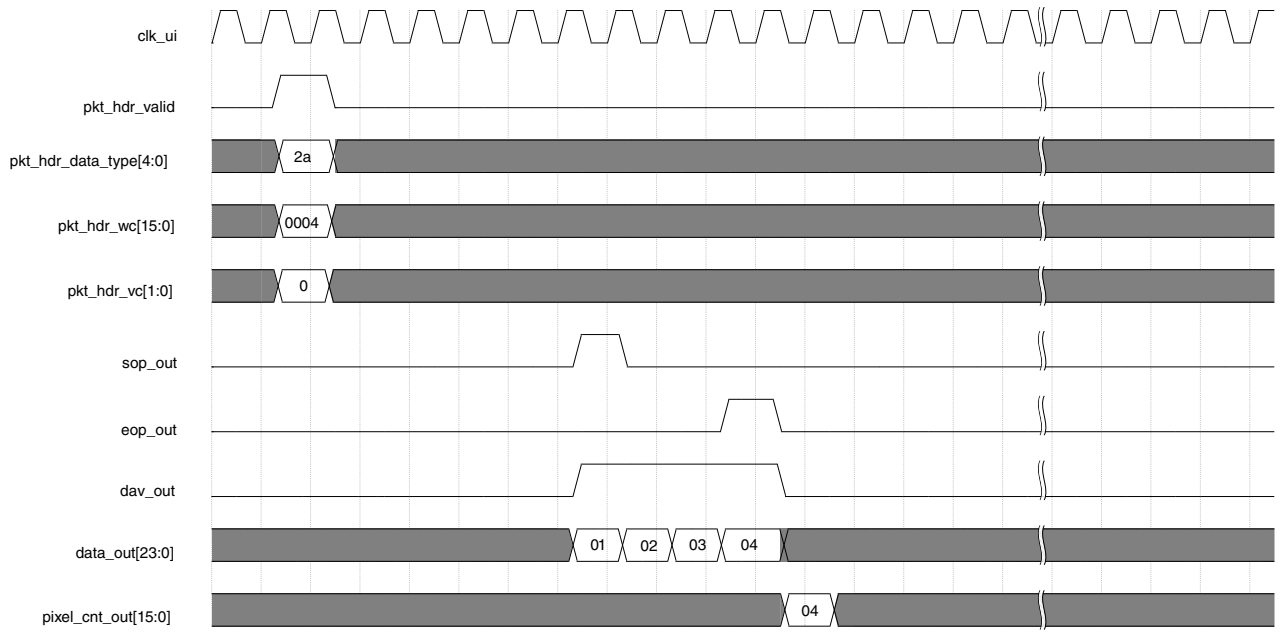


Figure 13-72. Receive Packet with a payload of 8 RAW8 pixels, single DPHY lane, and Virtual Channel = 0 single pixel configuration

13.8.3.3.2.2 Two back to back long packet request with RAW8 data and two DPHY lanes

The figure shows two packet requests that occur back to back. L_tx_pkt_cmd_req is asserted along with the packet parameters on l_tx_pkt_cmd. The request is acknowledged by the controller during the next clock period when l_tx_pkt_cmd_ack is asserted. The second request is immediate submitted by keeping l_tx_pkt_cmd_req asserted and changing the value on l_tx_pkt_cmd, indicating that the next packet will contain 4 bytes rather than the 3 that the previous packet contained. Note that it is not necessary to change l_tx_pkt_cmd in order to request another packet immediately following the first packet.

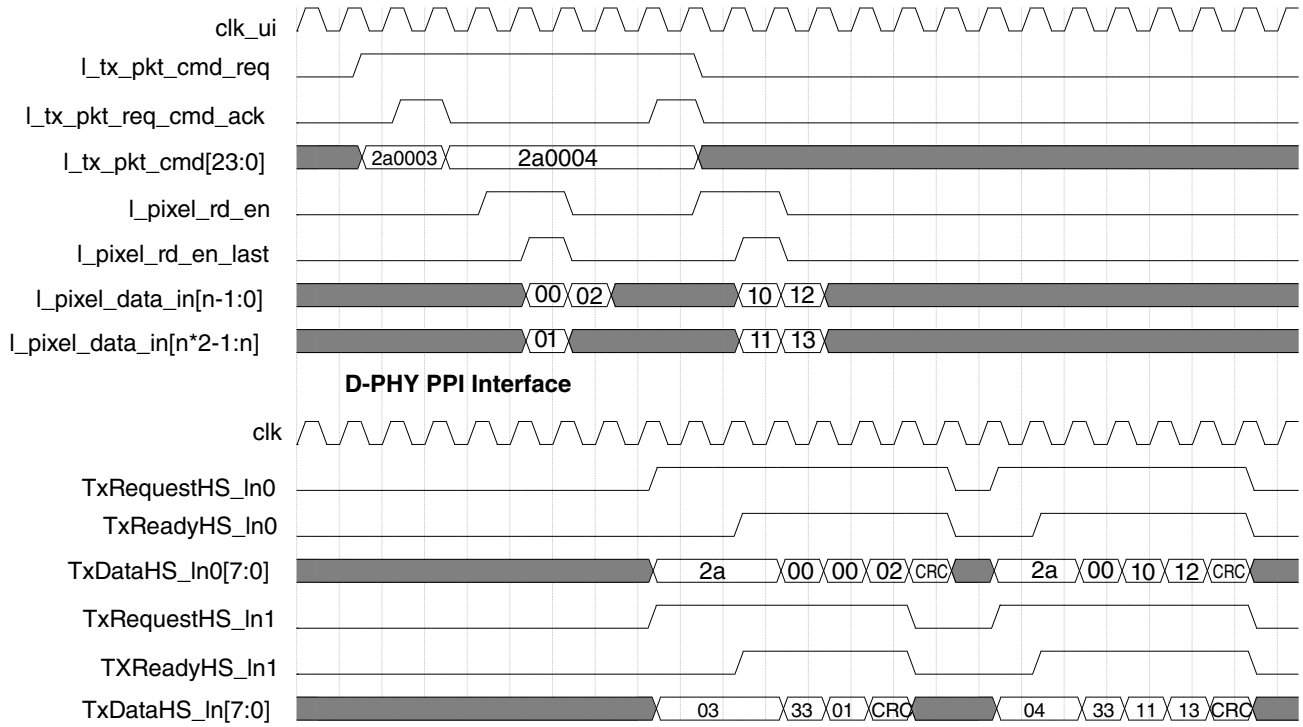


Figure 13-73. Two Transmit Packets, payloads of 3 and 4 pixels, two DPHY lanes, Virtual Channel=0, Double Pixel Mode

NOTE

The relationship of l_tx_pkt_cmd_req, l_tx_pkt_cmd_ack to TxRequestHS , TxReadyHS and l_pixel_data_in to TxDataHS is not defined. Furthermore, the relationship shown in the above figure is compressed for formatting considerations.

13.8.3.3.2.3 Short Packet Receive

The short packets do not contain any data. All the information in a short packet is contained in the 32 bit header. Pkt_hdr_valid asserts indicating that the packet header information contained on pkt_hdr_data_type, pkt_hdr_wc, and pkt_hdr_vc is valid. Short packets are defined as having a pkt_hdr_data_type value that is less than 0x10. With a short packet the CSI-2 Rx Controller Core does not assert sop_out, eop_out, nor dav_out.

The figure below shows the reception of a CSI-2 Short Packet type.

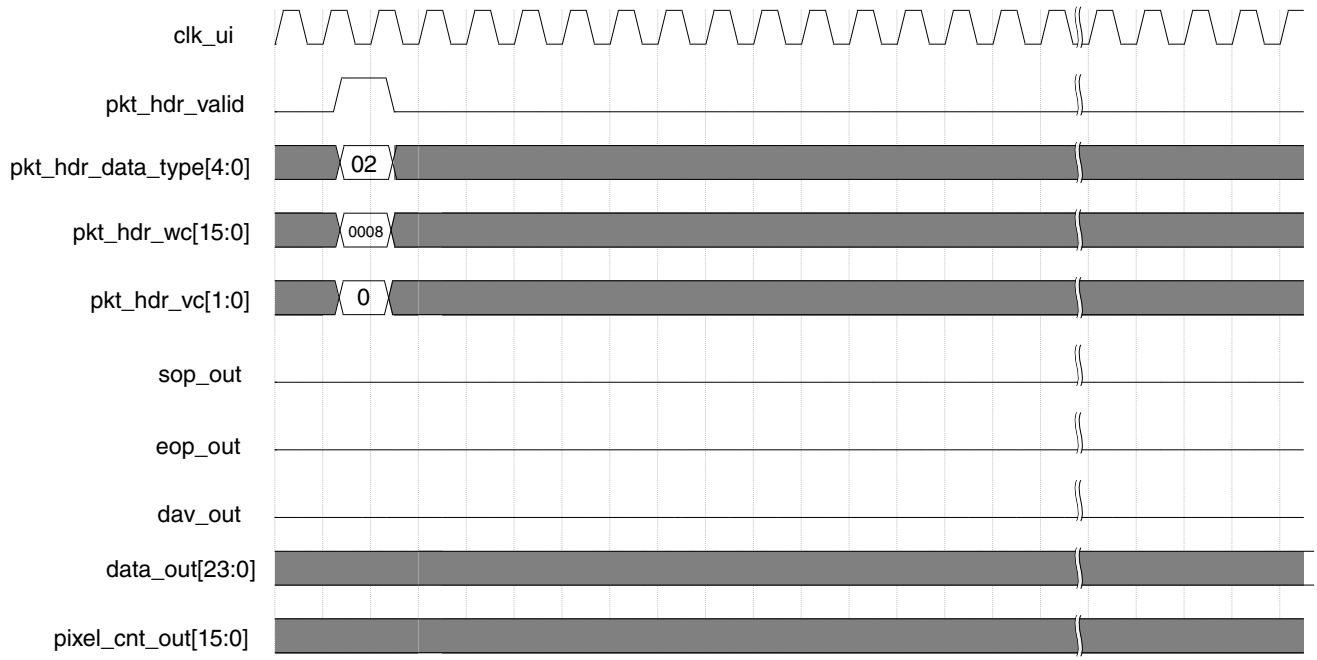


Figure 13-74. Short Packet receive with a data type of Line Start, Line Number of 8 , and Virtual Channel = 0 single pixel configuration

In this example, the data type ix 0x30, a user defined byte data type. Initially, pkt_hdr_valid asserts along with the packet header information. The packet header ports, pkt_hdr_data_type, pkt_hdr_wc, and pkt_hdr_vc indicate that the packet is a user defined data type with a payload of 4 bytes. For unsupported types, the Rx Controller outputs the entire packet minus the two byte CRC value at the end of the packet. The first four bytes are the packet header followed immediately by the 4 bytes of payload data.

The figure below shows a waveform diagram of a received packet on the local interface for an unsupported data type.

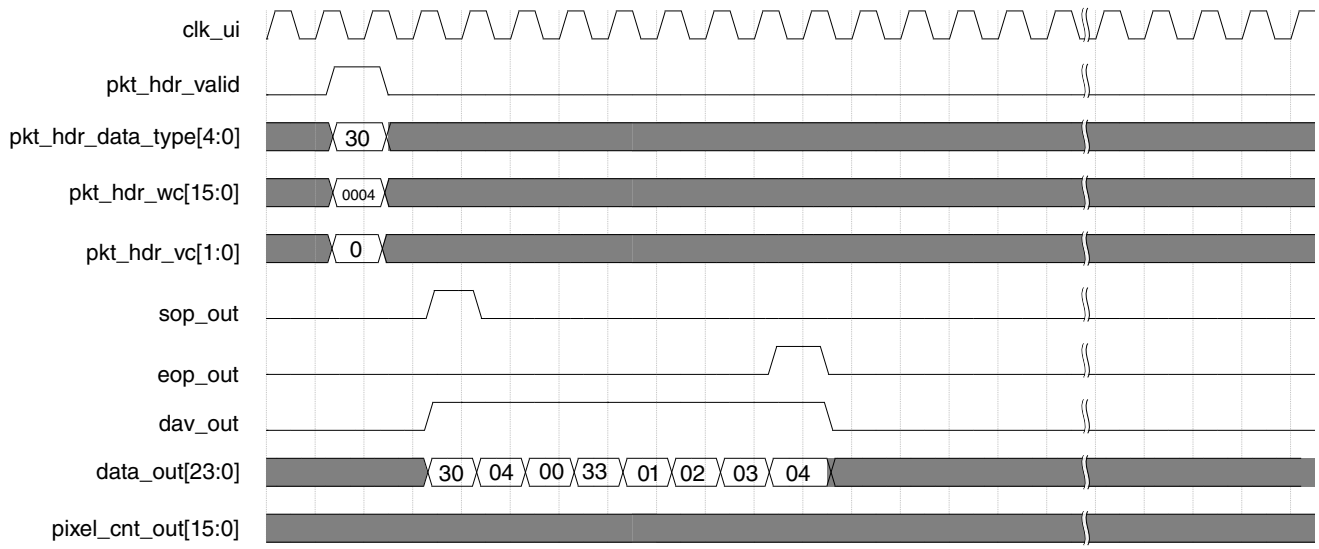


Figure 13-75. Received packet of an unsupported data type, 0x30, user defined data, single pixel configuration

13.8.3.4 CSI-2 Rx Controller Core Clocking

The RX controller core requires two clocks for proper operation clk_ui and clk. The clk_ui clock is the clock that the user interface is synchronous to, with all inputs and outputs referenced to the rising edge of clk_ui. The clk input is the clock that the CSI-2 Rx Controller Core uses to process data received from the D-PHY. Clk_ui must be of a sufficient frequency to keep up with the incoming data from the CSI-2 MIPI interface.

The data from the D-PHY is clocked into shallow FIFOs (one per DPHY lane) using a receive clock that the D-PHY generates. There is no direct timing requirements between the D-PHY generated receive clock and the CSI-2 Rx Controller Core's clk or clk_ui input. The frequency of the clock used for the RX controller's clk input must be of a sufficient frequency that the RX controller can keep up with the data from DPHY. The Rx Controller processes the Rx D-PHY data in a 32 bit wide data sufficient frequency that the Rx Controller can keep up with the receive data from the D-PHY. The Rx Controller processes the Rx D-PHY data in a 32 bit wide data path regardless of whether the CSI-2 MIPI receive data is coming across one, two, three, or four lanes. The frequency of clk must be exactly equal to or greater than the RX byte clock coming from the RX DPHY.

The CSI-2 Rx Controller Core can supply pixel data to the User application either one, two or four pixels per clock.

When in single Pixel Mode, the minimum frequency for clk_ui must be high enough to keep up with the incoming data rate of the CSI-2 MIPI interface. It is important to realize that the user interface is pixel based and therefore may be wider than the incoming byte data from the MIPI RX DPHY. As an example, if RGB888 data is being received, the user interface data width will be 24 bits wide (1 pixel per clk_ui rising edge and each pixel is 24 bits). As an example, let the MIPI interface run at 1000Mbps with a single lane. The data rate into the CSI-2 RX controller will be 125MBps so clk must run at 125MHz or higher. Since the user interface is 3 bytes wide, clk_ui must run at 125MHz/3 or higher to keep up with the incoming data.

the figure below shows one possible clocking methodology. In this example, the user Clk_Gen module generates the edge aligned clk_ui and clk clocks. It also generates a reference clock that the D-PHY might require. The Reference clock has no relationship to any of the CSI-2 Rx Controller Core's clocks.

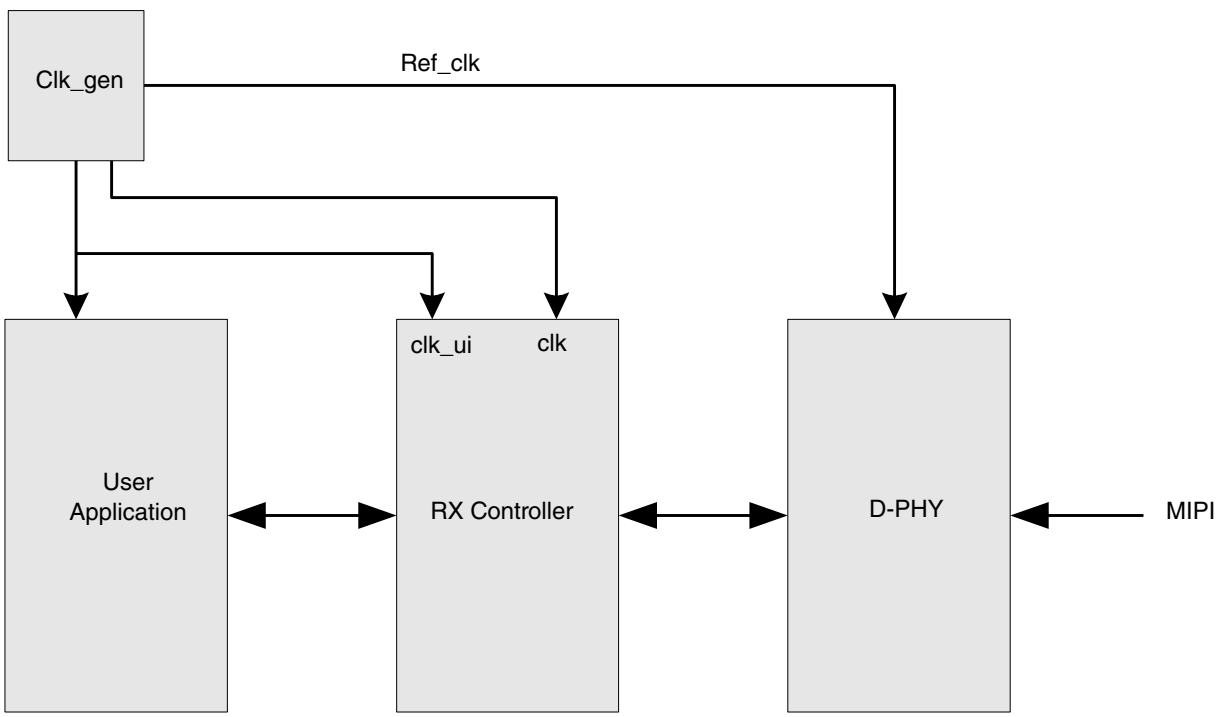


Figure 13-76. CSI-2 Rx Controller Core clocking example

13.8.3.5 CSI-2 Controller Core Configurations

The CSI-2 RX Controller Core User Interface supports either a single, double, or quad pixel wide data path. The double and quad wide pixel modes can help reduce the frequency the user interface must run at while the single pixel wide mode can be easier to interface to for some applications.

This chip supports the following:

- Single Pixel Configuration

13.8.3.5.1 Single Pixel Configuration

The CSI-2 Rx Controller Core User Interface supports a configuration where a single pixel of received data is processed every clock period.

The CSI-2 Rx Controller Core contains minimal pixel buffering (16 pixels or less) so the frequency of that the user interface runs at must be at least high enough to receive the data from the controller at a rate that is equal to or greater than the highest transmission data rate on the MIPI interface for all lanes combined. For example, if the total bandwidth of the physical MIPI interface will be 800Mbps with a data type of 8RAW then the user interface must be clocked at a minimum of 100MHz.

13.8.3.6 CSI-2 Rx Controller Core Data Types Formatting

The CSI-2 Rx Controller Core supports the various CSI-2 Data Types via the data_out port. The following subsections describe how the controller presents the various received data types on the data_out output port.

The CSI2 Controller supports the Single Pixel Configuration.

13.8.3.6.1 RAW6 (Data Type = 0x28)

With RAW6 data type (0x28) and the Rx Controller configured for single pixel mode, RAW6 data is contained on data_out[5:0].

13.8.3.6.2 RAW7 (Data Type = 0x29)

With RAW7 data type (0x29) and the Rx Controller configured for single pixel mode, RAW7 data is contained on data_out[6:0].

13.8.3.6.3 RAW8 (Data Type = 0x2A)

With RAW8 data type (0x2A) and the Rx Controller configured for single pixel mode, RAW8 data is contained on data_out[7:0].

13.8.3.6.4 RAW10 (Data Type = 0x2B)

With RAW10 data type (0x2B) and the Rx Controller configured for single pixel mode, the received RAW10 data is contained on data_out[9:0].

13.8.3.6.5 RAW12 (Data Type = 0x2C)

With RAW12 data type (0x2C) and the Rx Controller configured for single pixel mode, the received RAW12 data is contained on data_out[11:0].

13.8.3.6.6 RAW14 (Data Type = 0x2D)

With RAW14 data type (0x2D) and the Rx Controller configured for single pixel mode, the received RAW14 data is contained on data_out[13:0].

13.8.3.6.7 RGB444 (Data Type = 0x20)

With RGB444 data type (0x20) and the Rx Controller configured for single pixel mode, the blue data is on data_out[3:0], green on data_out[7:4], and red on data_out[11:8].

13.8.3.6.8 RGB555 (Data Type = 0x21)

With RGB555 data type (0x21) and the Rx Controller configured for single pixel mode the blue data is on data_out[4:0], green on data_out[9:5], and red on data_out[14:10].

13.8.3.6.9 RGB565 (Data Type = 0x22)

With RGB565 data type (0x22) and the Rx Controller configured for single pixel mode the blue data is on data_out[4:0], green on data_out[10:5], and red on data_out[15:11].

13.8.3.6.10 RGB666 (Data Type = 0x23)

With RGB666 data type (0x23) and the Rx Controller configured for single pixel mode, the blue data is on data_out[5:0], green on data_out[11:6], and red on data_out[17:12].

13.8.3.6.11 RGB888 (Data Type = 0x24)

With RGB888 data type (0x24) and the Rx Controller configured for single pixel mode, the blue data is on data_out[7:0], green on data_out[15:8], and red on data_out[23:16].

13.8.3.6.12 Legacy YUV420 8 bit (Data Type = 0x1A)

With Legacy YUV420 8 bit and the Rx Controller configured for single pixel mode the video data is received in UYY…/VYY sequences in odd/even lines. The U component is received with YY in a UYY group on odd lines while the V component is received with the same YY pairs in VYY groups on the even lines. See the MIPI CSI-2 Specification for a more complete description. When receiving an odd line data_out[7:0] contains the U component, data_out[15:8] Y1 and data_out[23:16] Y2. On even lines data_out[7:0] contains the V component, data_out[15:8] Y1 and data_out[23:16] Y2.

13.8.3.6.13 YUV420 8 bit (Data Type = 0x18,1C)

With YUV420 8 bit and YUV420 8 bit chroma shifted is received as YYYY (odd lines) UYVY (even lines) sequence. With the Rx controller configured for single pixel mode and receiving even numbered lines, data_out[7:0] contains the U1 component, bits [15:8] Y1. On the next data valid cycle bits [7:0] V1, bits [15:8] Y2. When receiving odd lines, data_out[7:0] contains the Y1 component, bits [15:8] Y2. On the next data valid cycle bits [7:0] Y3, bits [15:8] Y4.

13.8.3.6.14 YUV420 10 bit (Data Type = 0x19, 0x1D)

With YUV420 10 bit and YUV420 10 bit chroma shifted is received as YYYY (odd numbered lines) and UYVY (even numbered lines) sequence with each component represented by 10 bits. With the Rx controller configured for single pixel mode and receiving even numbered lines, data_out[9:0] contains the U1 component, bits [19:10] Y1. On the next data valid cycle bits [9:0] V1, bits [19:10] Y2. When receiving odd lines, data_out[9:0] contains the Y1 component, bits [19:10] Y2. On the next data valid cycle bits [9:0] Y3, bits [19:10] Y4.

13.8.3.6.15 YUV422 8 bit (Data Type = 0x1E)

With YUV422 8 bit is received as UYVY sequence. With the Rx controller configured for single pixel mode, data_out[7:0] contains the U1 component, bits [15:8] Y1. On the next data valid cycle bits [7:0] V1, bits [15:8] Y2.

13.8.3.6.16 YUV422 10 bit (Data Type = 0x1F)

With YUV422 10 bit is received as a UYVY sequence with each component represented by 10 bits. With the Rx controller configured for single pixel mode, data_out[9:0] contains the U1 component, bits [19:10] Y1. On the next RX data valid cycle, bits [9:0] V1, bits [19:10] Y2.

13.8.3.6.17 User Defined Data Types, (Data Type = 0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37)

With User Defined Data Types and the Rx Controller configured for single pixel mode, User Defined data is contained on data_out[7:0].

13.8.3.6.18 Null Data Type, (Data Type = 0x10)

With User Null Data Type and the Rx Controller configured for single pixel mode, Null data is contained on data_out[7:0].

13.8.3.6.19 Blanking Data Type, (Data Type = 0x11)

With Blanking Data Type and the Rx Controller configured for single pixel mode, Blanking data is contained on data_out[7:0].

13.8.3.6.20 Embedded Data Type, (Data Type = 0x12)

With User Null Data Type and the Rx Controller configured for single pixel mode, Embedded data is contained on data_out[7:0].

When the Rx Controller is configured for double pixel mode, the Embedded data that is received is contained on data_out[7:0] and data_out[MAX_PIXEL_WIDTH +7:MAX_PIXEL_WIDTH] with data_out[7:0] having been received first.

When the Rx Controller is configured for quad pixel mode, the first Embedded data that is received is contained on data_out[7:0], the second on data_out[MAX_PIXEL_WIDTH +7:MAX_PIXEL_WIDTH], the third on data_out[MAX_PIXEL_WIDTH*2+7:MAX_PIXEL_WIDTH*2],

13.8.3.6.21 Unsupported Data Types

With unsupported data types and the Rx Controller configured for single pixel mode, the raw packet data (including the header but minus the CRC) is contained on data_out[7:0].

13.8.3.7 RX Video Interface

The CSI-2 Rx Video Interface is connected internally to the local interface outputs of the CSI-2 Rx Controller Core. This interface provides both the packet header information and the pixel data (after processing by the Pixel Format block). CSI-2 Rx Controller Cores may be built with any number of Pixel Formats supported. The CSI-2 Rx Video Interface only recognizes video formats for which the CSI-2 Rx Controller Core has been configured to support.

The general timing of the video interface is determined by the timing of the packets received by the CSI-2 RX controller which are used to determine when to generate a Vertical Sync and when to output a line of video data.

The CSI-2 Rx Video Interface may receive several different types of packet data types as shown in the table below.

Packet Type	Data ID	Function
Frame Start Code (FS)	0x00	Indicates start of frame
Frame End Code (FE)	0x01	Indicates end of frame
Line Start Code (LS)	0x02	Indicates start of line
Line End Code (LS)	0x02	Indicates end of line
Image Data Codes (IMG)	various	Image data packets carry the sample data. The CSI-2 Rx Video Interface will respond to those Data types which are enabled in the CSI-2 Rx Controller Core. See CSI-2 Rx Controller Core Data Types Formatting

13.8.3.7.1 Video Interface Output Raster Timing

The output timing of the raster is formed by the packets sent to the CSI-2 Rx Controller Core and is determined by the relative timing of these packets. This timing is controlled by the transmitter and is not determined by the CSI-2 Rx Controller Core. As shown in Figure 4.8 below, the basic sequence of packets sent by the transmitter (often a sensor or camera) consists of a Frame Start packet (short packet), followed by some number of Image Data packets (long packets) with one Image data packet per video line and the followed by a Frame End packet (short packet). The Frame End packet is followed by pause which determines the Blanking Lines interval. The length of the Image Data packets is determined by the horizontal resolution of the sensor and each line's samples are carried by a single Image Data packet.

NOTE

Note that each Image Data packet may optionally be preceded by a Line Start packet and followed by an optional Line End packet. The Line Start and Line End packets are ignored by the CSI-2 Rx Video Interface.

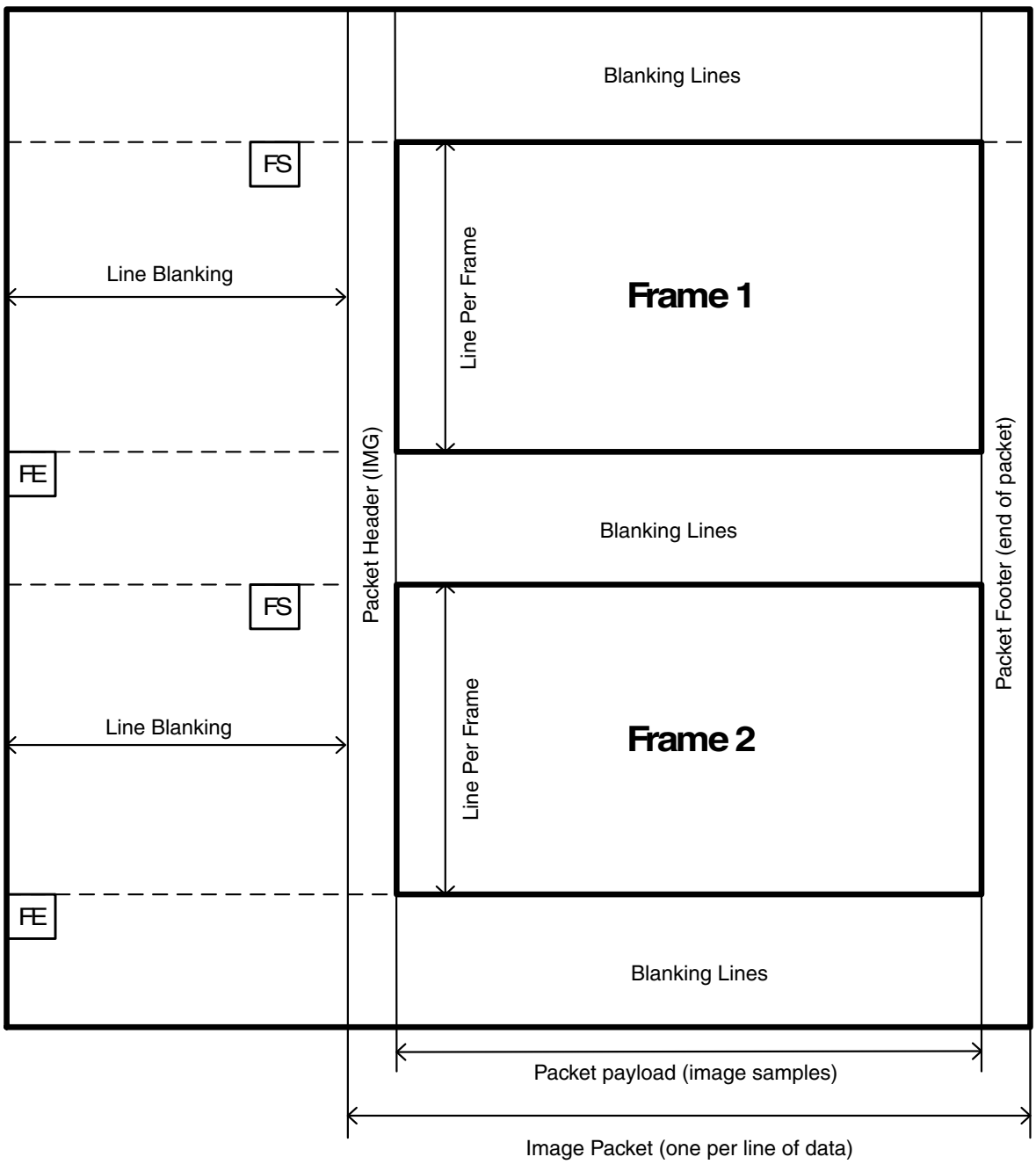


Figure 13-77. Rx Video Interface Raster Timing

NOTE

- FS- Frame Start Packet

- FE- Frame End Packet
- IMG- Image Data Packet

Note that the interface does not generate Horizontal sync pulses during the Blanking Lines.

The CSI-2 Rx Video controller generates a Vertical sync pulse upon the occurrence of every Frame Start packet. Each line of video sample data is preceded by a Horizontal sync pulse.

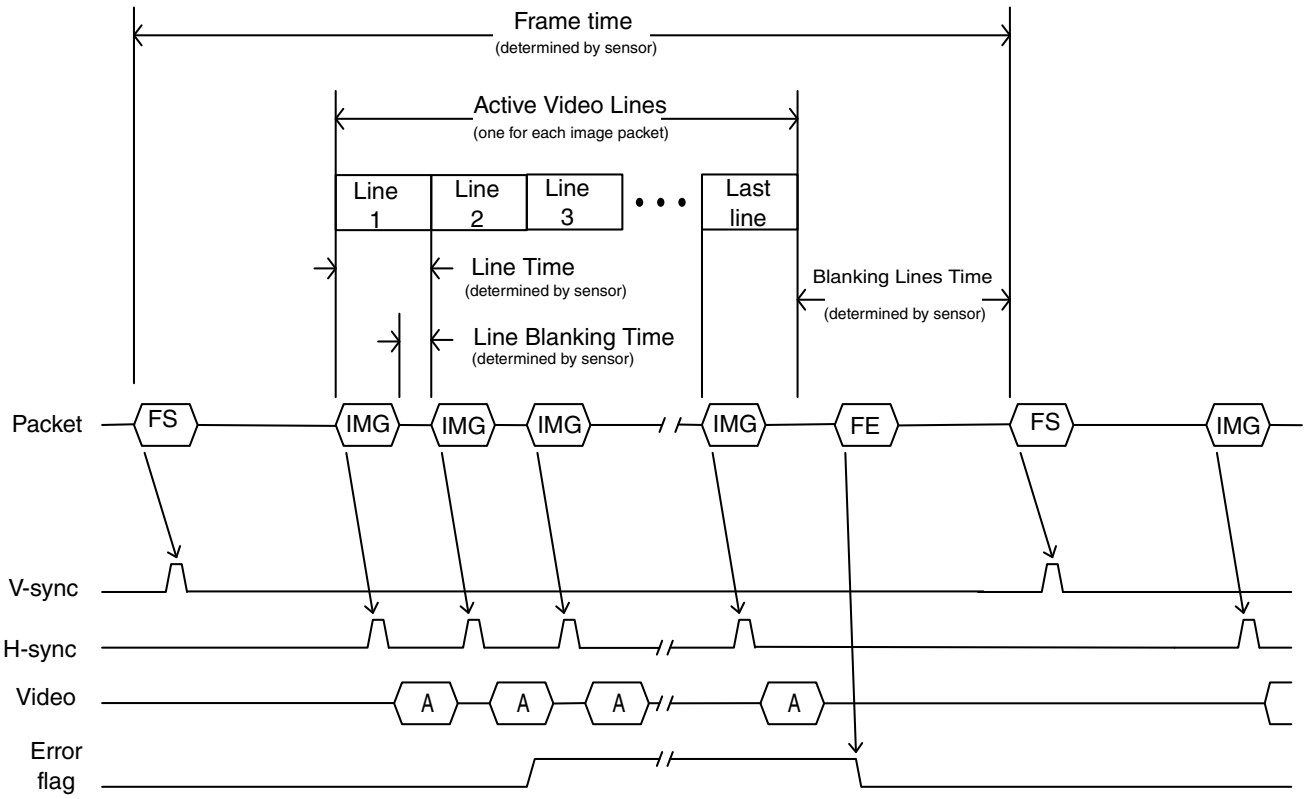


Figure 13-78. Relationship of Packet Stream to Video Interface

NOTE

- FS - Frame start Packet (causes V-sync)
- IMG - Image Data Packet(causes H-sync and Video after FIFO reaches send level)
- FE - Frame End Packet (causes any pending Error Flags which were set in the previous frame)
- A - Time period when video samples are valid (follows H-sync)

The video sample output bus (vid_out) is logically divided up into pixel lanes. The number of pixel lanes is determined by the pixel width of the CSI-2 Controller Core. This width may be one, two or four pixels wide (single, double or quad pixel interface). The bit width of each pixel lane is determined the widest pixel width supported by the core controller's Pixel Format block. When transferring image data whose pixels are narrower than the maximum required width unused more significant bits in each pixel lane will be output as zero.

To determine the validity of the data presented on each pixel lane the CSI-2 Rx Video Interface supplies a data enable output (vid_de) whose width is equal to the number of pixel lanes in the interface. When a data enable bit is high its pixel lane is valid.

Thus as a design example:

1. Controller supports a double pixel interface and data types RAW8, RAW10 and RAW12. In this case the maximum pixel width is 12 bits.
2. For this interface the total width of the video output bus is 24 bits (2 * 12) This bus is logically divided into two pixel lanes each supplying 12 bits. That sample which arrived earlier is delivered on the low half (bits[11:0]) of this bus while the following sample is delivered on the high half (bits[23:12]).
3. The vid_de bus would be 2 bits with the least significant bit enabling the low pixel lane and the most significant bit enabling the high pixel lane

The situation is similar for a single pixel controller interface. In this case there is only a single pixel lane in the video output bus and a single video enable bit. For a four pixel controller interface there would be four pixel lanes and a four bit vid_de bus.

The only time at which all vid_de bits will not be asserted is on the final transfer of video pixels in a scan line. At this time only some of the lower order bits in the vid_de bus will be asserted only if the number of samples per line is not evenly divisible by the number of pixel lanes.

Thus as a design example:

1. Assume a double pixel interface which results in a two video data lanes. Image data packets are 1023 samples long.
2. This would result in 511 transfers of video out with both bits of vid_de set (1022 samples). The last transfer would only have the least significant vid_de bits set to transfer the final sample to achieve the 1023 samples in the scan line.

13.8.3.7.2 Video Interface Video Port Descriptions

The Video output interface outputs all the signals associated with video port. All output signals in the interface are synchronous to clk_vid.

PORT	TYPE	DESCRIPTION
clk_vid	Input	Clock for all signals associated with the video port
clk_vid_reset_n	Input	Asynchronous active low reset for all logic clocked by clk_vid
vid_intfc_enb	Input	Active high enable for the video interface.
vid_h_sync	Output	Video H Sync pulse
vid_v_sync	Output	Video V Sync pulse
vid_out [VID_PIXEL_WIDTH*RX_PIXELS_PER_CLK-1:0]	Output	Video data bus. Contains 1, 2 or 4 pixel lanes as set by the CSI-2 Rx Controller port configuration (single, double or quad pixel interface). The width of each pixel lane is set by the widest pixel with of all the data types enabled in the controller. See CSI-2 Rx Controller Core Data Types Formatting SinglePixelInterface: Pixel lane 0= vid_out[VID_PIXEL_WIDTH-1:0] Dual Pixel Interface: Pixel lane 0= vid_out[VID_PIXEL_WIDTH-1:0] Pixel lane 1= vid_out[(2*VID_PIXEL_WIDTH)-1:VID_PIXEL_WIDTH] Quad Pixel Interface: Pixel lane 0= vid_out[VID_PIXEL_WIDTH-1:0] Pixel lane 1= vid_out[(2*VID_PIXEL_WIDTH)-1:VID_PIXEL_WIDTH] Pixel lane 2 = vid_out[(3*VID_PIXEL_WIDTH)-1:2*VID_PIXEL_WIDTH] Pixel lane 3 = vid_out[(4*VID_PIXEL_WIDTH)-1:3*VID_PIXEL_WIDTH]
vid_de [0]	Output	Video data pixel lane valid. One bit per Pixel lane. Logic 1 indicates a valid pixel value. SinglePixelInterface: Bit 0: Pixel lane 0 Dual Pixel Interface: Bit 0 : Pixel lane 0 Bit 1: Pixel lane 1 Dual Pixel Interface: Bit 0 : Pixel lane 0 Bit 1: Pixel lane 1 Bit 2: Pixel lane 2 Bit 3: Pixel lane 3

13.8.3.7.3 Video Output Port Waveforms

The vid_de port is used to enable capture of valid video pixels from the vid_out bus. The vid_de will have a single bit for each pixel lane in the vid_out bus. Thus for a single pixel interface (one pixel lane) this bus will be a single signal.

The vid_de port will assert all its signals high for all transfers except possibly the last transfer. This last transfer may only assert some number of lower order bits to one and the remaining higher order bits to zero. The number of valid pixel lanes in the last transfer is determined by the number of pixels transferred by the image packet which generated the video line.

The ending value may be predicted by taking number of pixels transferred in an image packet modulo number of pixel lanes. Any non-zero remainder indicates some number of vid_de bits will be zero on the last transfer.

The following table shows the last vid_de value as a function of number of pixel lanes and packet size:

MODULO	
	Single Pixel Interface
0	1
1	NA
2	NA
3	NA

NOTE

MODULO = Number_of_pixels_in_image_packet MOD number_of_pixel_lanes

13.8.3.7.4 Video Interface Error Port Descriptions

The video interface outputs three error flags which can assist in hardware debugging or be used to notify software of problems. Whenever one of these error conditions occurs the video interface stops responding to input packets until such time a Frame End packet is detected. At this time the error flags are reset and the interface tries again on the next frame. Assertion of these error conditions can mean a misconfigured interface or bad input packets.

All these flags are synchronous to the controller clk_ui. The interface must be enabled (vid_intfc_enb = 1) for these errors to be detected.

Table 13-37. CSI-2 Rx Video Interface Error Port Descriptions

Port	Type	Description
vid_err_send_level	Output	This flag indicates the video interface pixel fifo did not accumulate enough sample to trigger its send level setting before the end of an image data packet was detected. This flag would generally mean that the cfg_vid_p_fifo_send_level is set to high.
vid_err_fifo_wr_ovfl	Output	This flag indicates the video interface pixel fifo overflowed on a pixel write. This means the output video port is not removing pixels quickly enough to keep up with the input data rate. This flag would generally mean the clk_vid rate is too low.

The figure below shows the error flag interface waveforms:

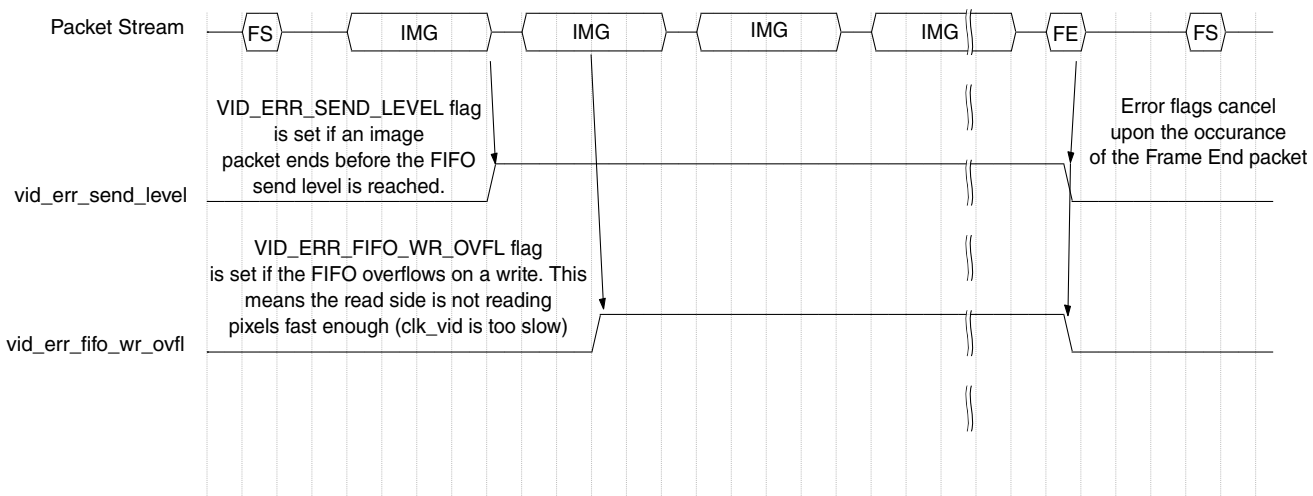


Figure 13-79. Error Flag Interface Waveforms

13.8.3.8 Rx Control and Status Register (CSR) APB Interface

The CSR provides an APB compatible interface that enables control of the controller's configuration inputs via registers accessible via the APB interface.

13.8.4 Memory Map and register definition

This section includes the memory map and detailed descriptions of all registers.

13.8.4.1 register descriptions

13.8.4.1.1 CSI2RX memory map

csi2_rx_csr_1 base address: 30A7_0000h

csi2_rx_csr_2 base address: 30B6_0000h

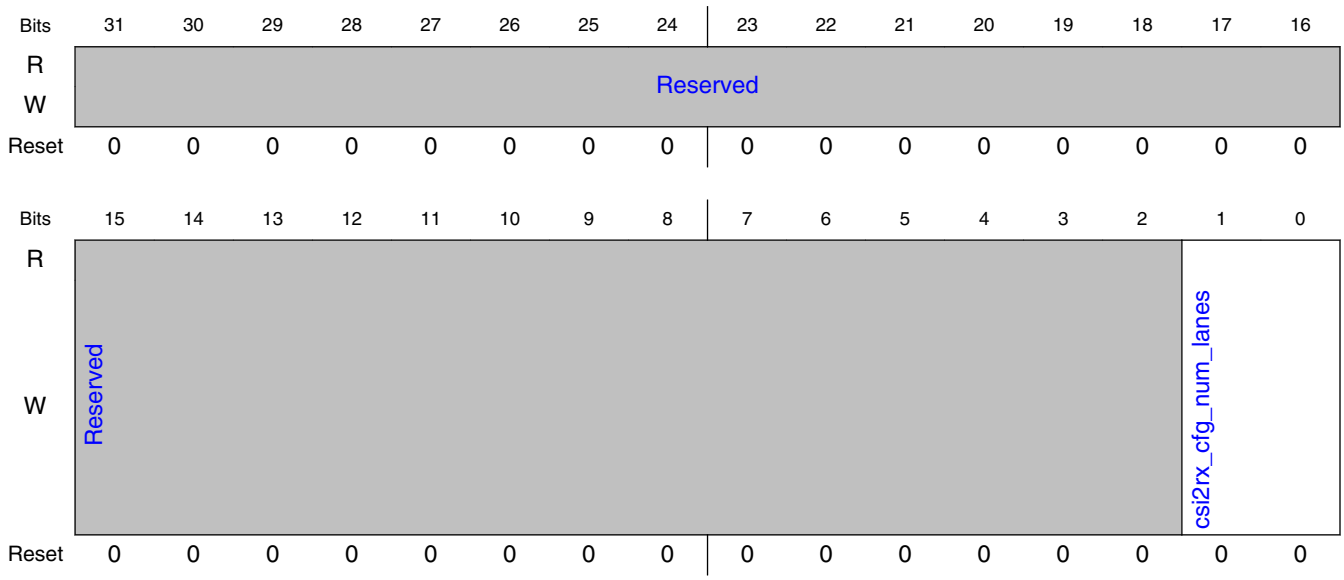
Offset	Register	Width (In bits)	Access	Reset value
100h	Lane Configuration Register (CSI2RX_CFG_NUM_LANES)	32	RW	0000_0000h
104h	Disable Data Lane Register (CSI2RX_CFG_DISABLE_DATA_LANES)	32	RW	0000_000Fh
108h	ECC and CRC Error Status Register (CSI2RX_BIT_ERR)	32	RO	0000_0000h
10Ch	IRQ Status Register (CSI2RX_IRQ_STATUS)	32	RO	0000_0000h
110h	IRQ Mask Setting Regsiter (CSI2RX_IRQ_MASK)	32	RW	0000_0000h
114h	ULPS Status Register (CSI2RX_ULPS_STATUS)	32	RO	0000_0000h
118h	ERRSot HS Status Register (CSI2RX_PPI_ERRSOT_HS)	32	RO	0000_0000h
11Ch	ErrSotSync HS Status Register (CSI2RX_PPI_ERRSOTSYNC_HS)	32	RO	0000_0000h
120h	ErrEsc Status Register (CSI2RX_PPI_ERRESC)	32	RO	0000_0000h
124h	ErrSyncEsc Status Register (CSI2RX_PPI_ERRSYNCESC)	32	RO	0000_0000h
128h	ErrControl Status Register (CSI2RX_PPI_ERRCONTROL)	32	RO	0000_0000h
12Ch	Disable Payload 0 Register (CSI2RX_CFG_DISABLE_PAYLOAD_0)	32	RW	0000_0000h
130h	Disable Payload 1 Register (CSI2RX_CFG_DISABLE_PAYLOAD_1)	32	RW	0000_0000h
188h	FIFO Send Level Configuration Register (CSI2RX_CFG_VID_P_FIFO_SEND_LEVEL)	32	RW	0000_0000h
18Ch	VSYNC Configuration Register (CSI2RX_CFG_VID_VSYNC)	32	RW	0000_0000h
190h	Start of HSYNC Delay control Register (CSI2RX_CFG_VID_HSYNC_FP)	32	RW	0000_0000h
194h	HSYNC Configuration Register (CSI2RX_CFG_VID_HSYNC)	32	RW	0000_0000h
198h	End of HSYNC Delay Control Register (CSI2RX_CFG_VID_HSYNC_BP)	32	RW	0000_0000h

13.8.4.1.2 Lane Configuration Register (CSI2RX_CFG_NUM_LANES)

13.8.4.1.2.1 Offset

Register	Offset
CSI2RX_CFG_NUM_LANES	100h

13.8.4.1.2.2 Diagram



13.8.4.1.2.3 Fields

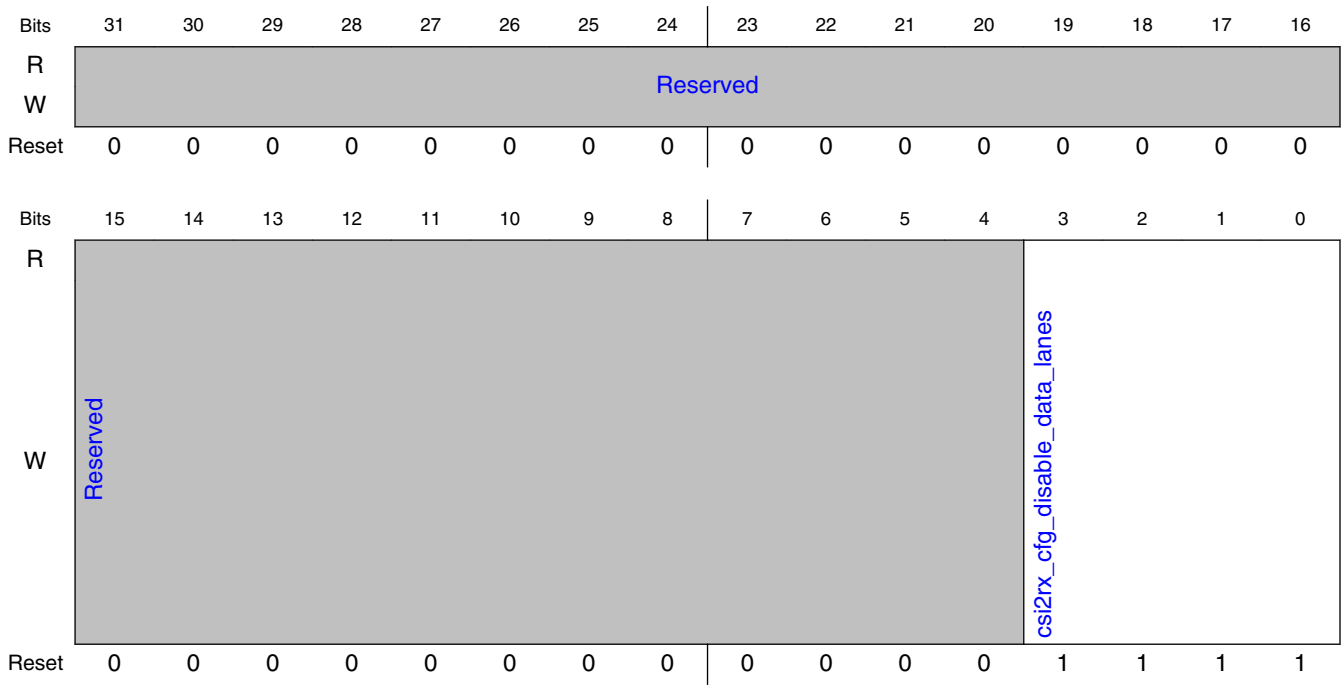
Field	Function
31-2 —	Reserved
1-0 csi2rx_cfg_num_lanes	Sets the number of active lanes that are to be used for receiving data. 00b - 1 Lane 01b - 2 Lane 10b - 3 Lane 11b - 4 Lane

13.8.4.1.3 Disable Data Lane Register (CSI2RX_CFG_DISABLE_DATA_LANES)

13.8.4.1.3.1 Offset

Register	Offset
CSI2RX_CFG_DISABLE_DATA_LANES	104h

13.8.4.1.3.2 Diagram



13.8.4.1.3.3 Fields

Field	Function
31-4 —	Reserved
3-0 csi2rx_cfg_disable_data_lanes	Setting bits to a '1' value causes the DPHY Enable signal to deassert. 0001b - Data Lane 0 0010b - Data Lane 1 0100b - Data Lane 2 1000b - Data Lane 3

13.8.4.1.4 ECC and CRC Error Status Register (CSI2RX_BIT_ERR)

13.8.4.1.4.1 Offset

Register	Offset
CSI2RX_BIT_ERR	108h

13.8.4.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						csi2rx_bit_err									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.4.3 Fields

Field	Function
31-10 —	Reserved
9-0 csi2rx_bit_err	BIT_ERR: CSI-2 RX Controller ECC and CRC error status. Bit [0] – value of '1' indicates ECC two bit error has occurred since the last read of this register Bit [1] – value of '1' indicates ECC one bit error has occurred since the last read of this register Bit [6:2] – ECC one bit error bit position that occurred since the last read of this register Bit [7] – value of '1' indicates a CRC error has occurred since the last read of this register Bit [8] – value of '1' indicates that rx video interface signal vid_err_send_level has asserted since the last read of this register. Register reads 0 when rx video interface included. Bit [9] – value of '1' indicates that rx video interface signal vid_err_fifo_wr_ovfl has asserted since the last read of this register. Register reads 0 when rx video interface included.

13.8.4.1.5 IRQ Status Register (CSI2RX_IRQ_STATUS)

13.8.4.1.5.1 Offset

Register	Offset
CSI2RX_IRQ_STATUS	10Ch

13.8.4.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							csi2rx_irq_status								
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.5.3 Fields

Field	Function
31-9 —	Reserved
8-0 csi2rx_irq_status	CSI2 RX IRQ status [0] – crc error [1] – one bit ecc error [2] – two bit ecc error [3] – ULPS status change [4] – DPHY ErrSotHS has occurred [5] – DPHY ErrSotSync_HS has occurred [6] – DPHY ErrEsc has occurred [7] – DPHY ErrSyncEsc has occurred [8] – DPHY ErrControl has occurred

13.8.4.1.6 IRQ Mask Setting Regsiter (CSI2RX_IRQ_MASK)

13.8.4.1.6.1 Offset

Register	Offset
CSI2RX_IRQ_MASK	110h

13.8.4.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								csi2rx_irq_mask							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.6.3 Fields

Field	Function
31-9 —	Reserved
8-0 csi2rx_irq_mask	CSI2 RX IRQ Mask setting Each bit in IRQ_MASK corresponds to each bit in IRQ_STATUS. Setting a bit in IRQ_MASK to 1 will mask the corresponding bit in IRQ_STATUS from causing irq_out to assert.

13.8.4.1.7 ULPS Status Register (CSI2RX_ULPS_STATUS)

13.8.4.1.7.1 Offset

Register	Offset
CSI2RX_ULPS_STATUS	114h

13.8.4.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						csi2rx_ulps_status									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.7.3 Fields

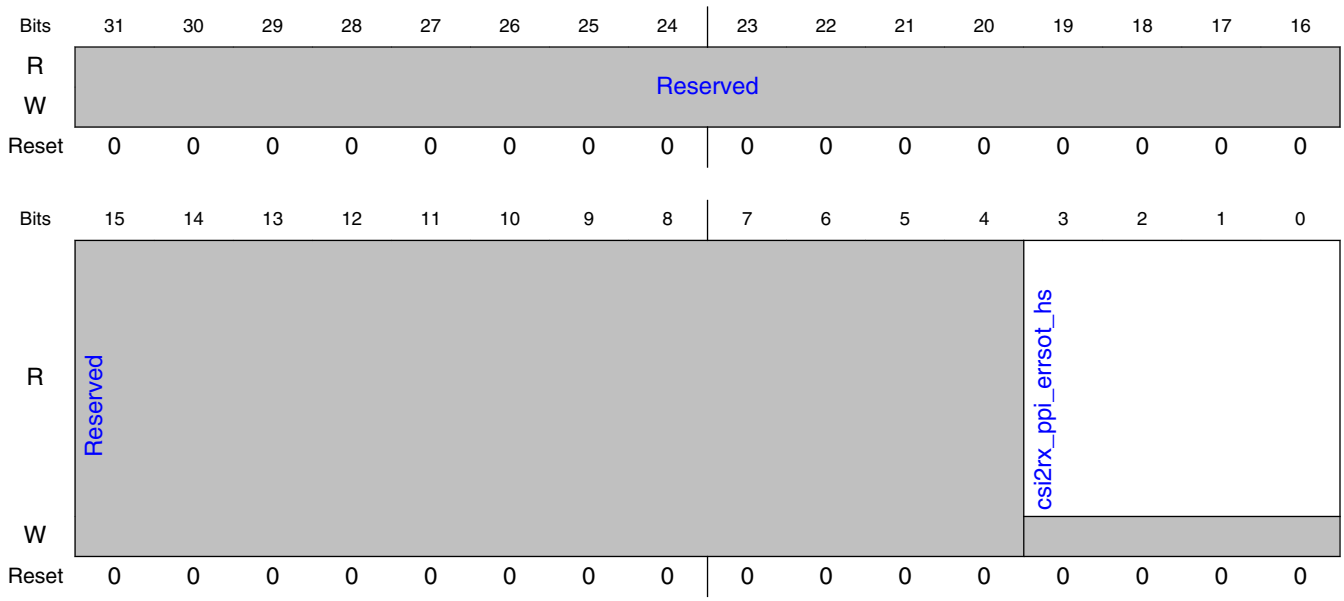
Field	Function
31-10 —	Reserved
9-0 csi2rx_ulps_status	Status of RX DPHY ULPS state [0] – Value '1' indicates Clock Lane in ULPS state [1] – Value '1' indicates Data Lane 0 in ULPS state [2] – Value '1' indicates Data Lane 1 in ULPS state [3] – Value '1' indicates Data Lane 2 in ULPS state [4] – Value '1' indicates Data Lane 3 in ULPS state [5] – Value '1' indicates Clock Lane in Mark state [6] – Value '1' indicates Data Lane 0 in Mark state [7] – Value '1' indicates Data Lane 1 in Mark state [8] – Value '1' indicates Data Lane 2 in Mark state [9] – Value '1' indicates Data Lane 3 in Mark state

13.8.4.1.8 ERRSot HS Status Register (CSI2RX_PPI_ERRSOT_HS)

13.8.4.1.8.1 Offset

Register	Offset
CSI2RX_PPI_ERRSOT_HS	118h

13.8.4.1.8.2 Diagram



13.8.4.1.8.3 Fields

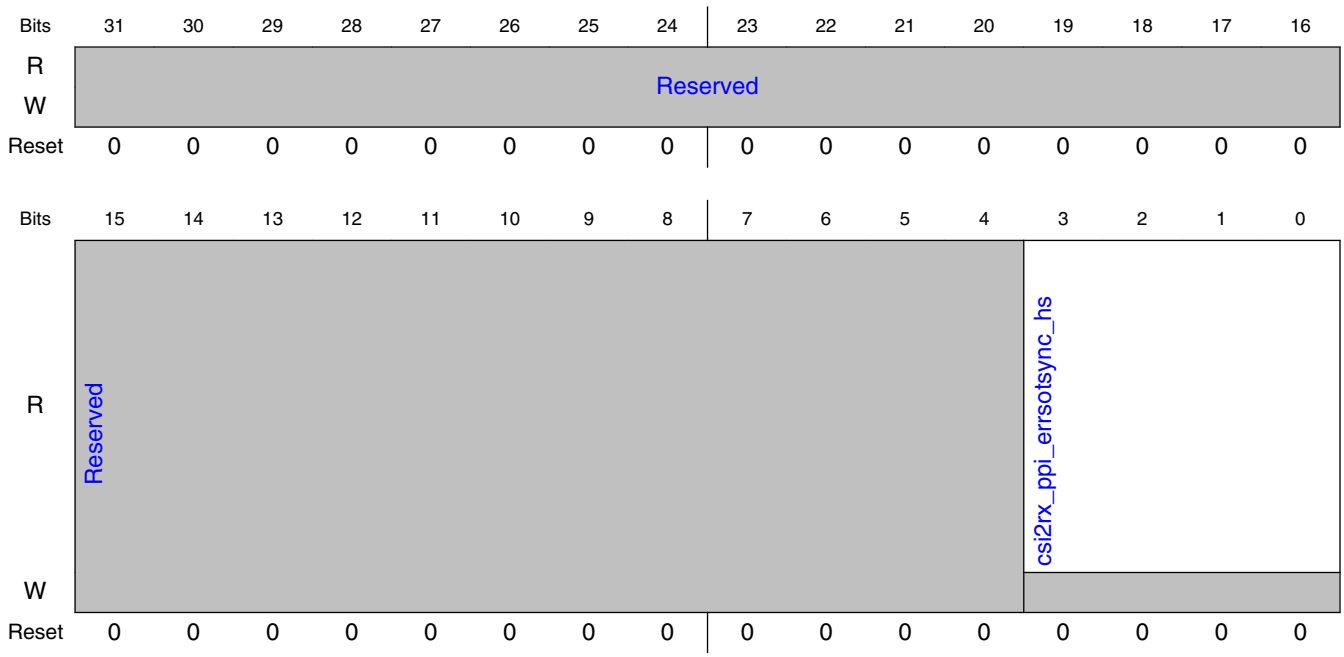
Field	Function
31-4 —	Reserved
3-0 csi2rx_ppi_errsot_hs	CSI2 RX DPHY PPI ErrSotHS captured status from the DPHY. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.4.1.9 ErrSotSync HS Status Register (CSI2RX_PPI_ERRSOTSYNC_HS)

13.8.4.1.9.1 Offset

Register	Offset
CSI2RX_PPI_ERRSOTSYNC_HS	11Ch

13.8.4.1.9.2 Diagram



13.8.4.1.9.3 Fields

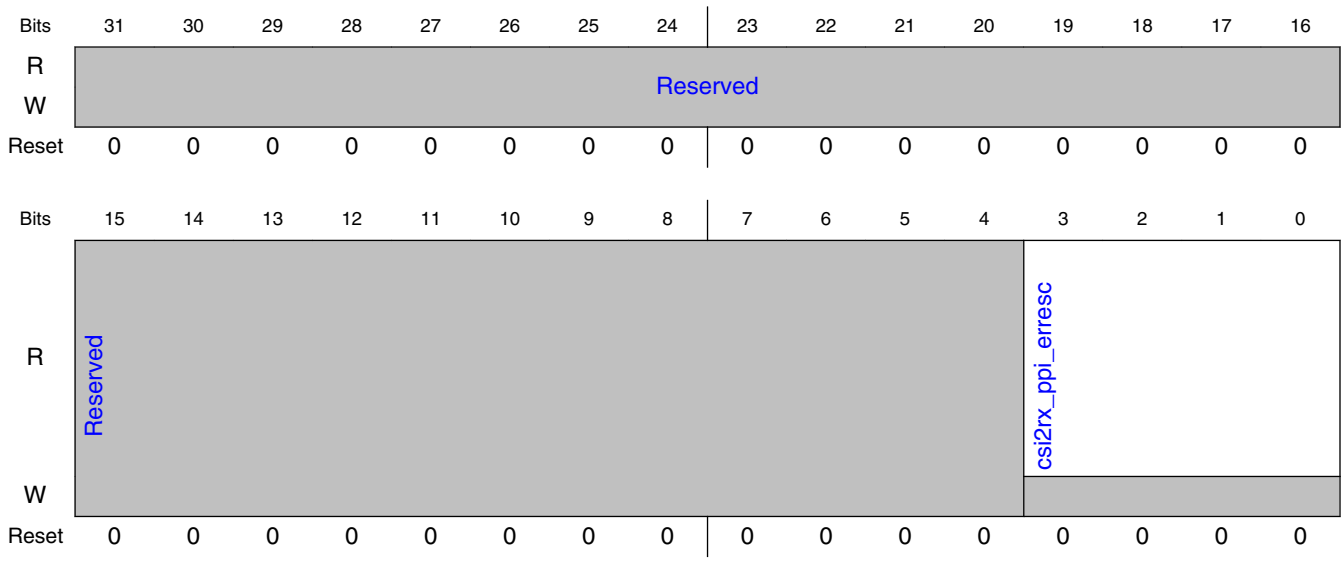
Field	Function
31-4 —	Reserved
3-0 csi2rx_ppi_errsotsync_hs	CSI2 RX DPHY PPI ErrSotSync_HS captured status from the DPHY. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.4.1.10 ErrEsc Status Register (CSI2RX_PPI_ERRESC)

13.8.4.1.10.1 Offset

Register	Offset
CSI2RX_PPI_ERRESC	120h

13.8.4.1.10.2 Diagram



13.8.4.1.10.3 Fields

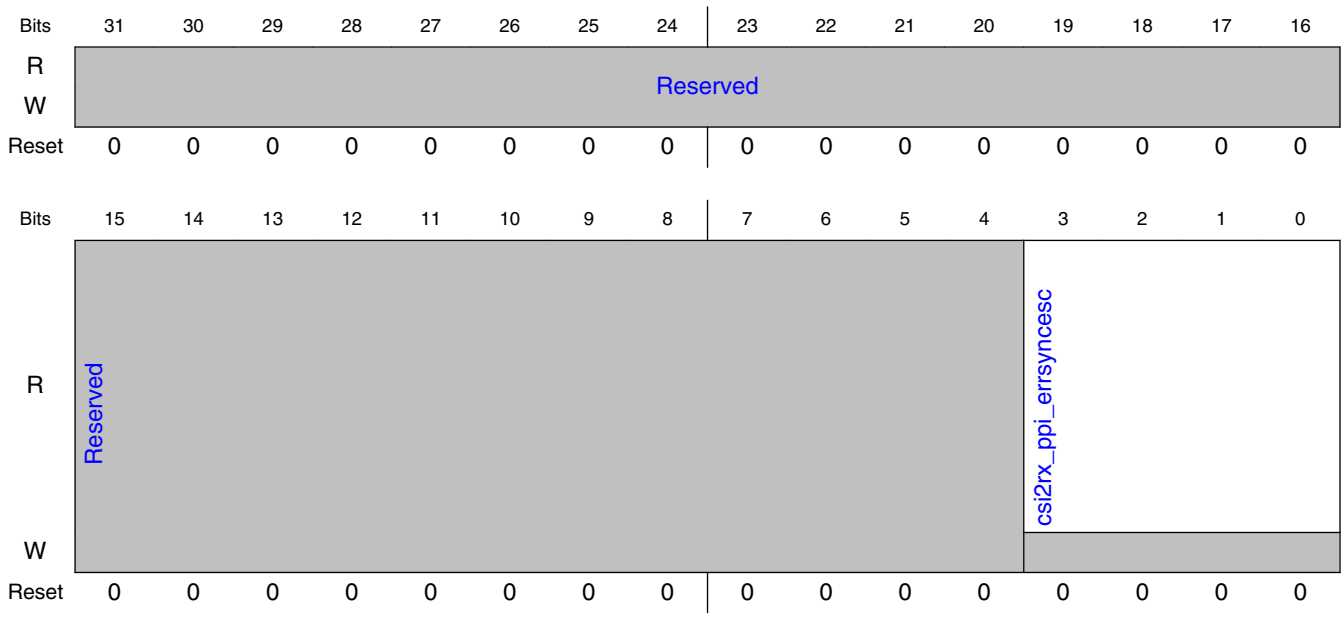
Field	Function
31-4 —	Reserved
3-0 csi2rx_ppi_erresc	CSI2 RX DPHY PPI ErrEsc captured status from the DPHY. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.4.1.11 ErrSyncEsc Status Register (CSI2RX_PPI_ERRSYNCESC)

13.8.4.1.11.1 Offset

Register	Offset
CSI2RX_PPI_ERRSYNCESC	124h

13.8.4.1.11.2 Diagram



13.8.4.1.11.3 Fields

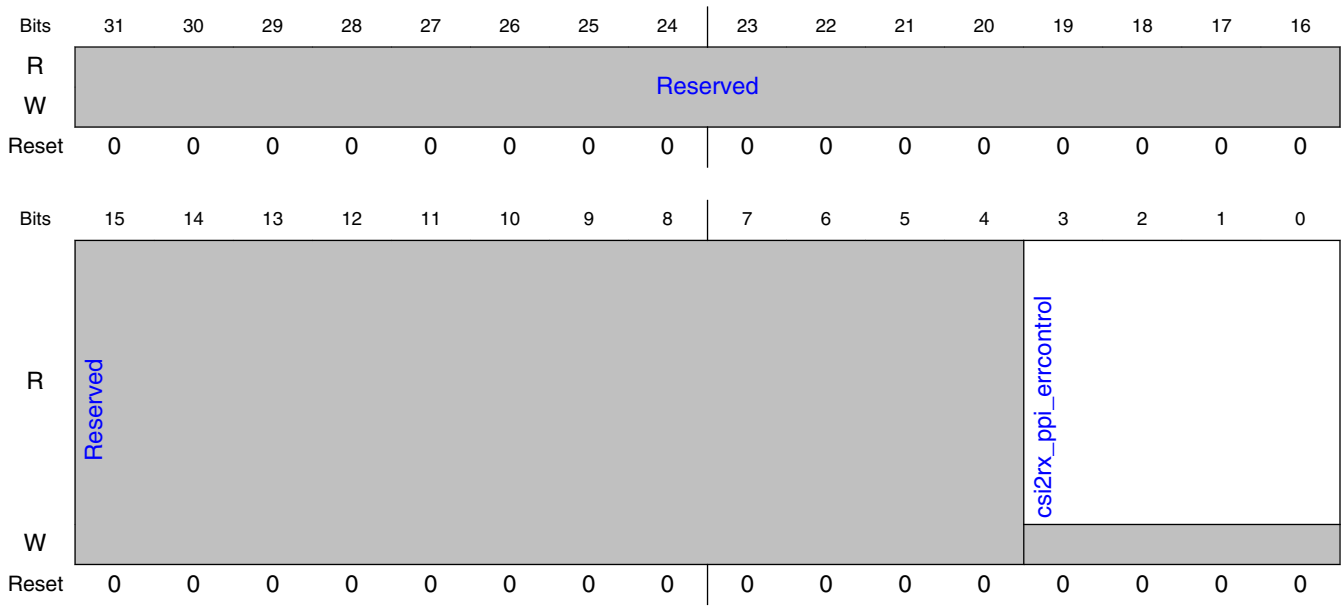
Field	Function
31-4 —	Reserved
3-0 csi2rx_ppi_errsyncesc	CSI2 RX DPHY PPI ErrSyncEsc captured status from the DPHY. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.4.1.12 ErrControl Status Register (CSI2RX_PPI_ERRCONTROL)

13.8.4.1.12.1 Offset

Register	Offset
CSI2RX_PPI_ERRCONTROL	128h

13.8.4.1.12.2 Diagram



13.8.4.1.12.3 Fields

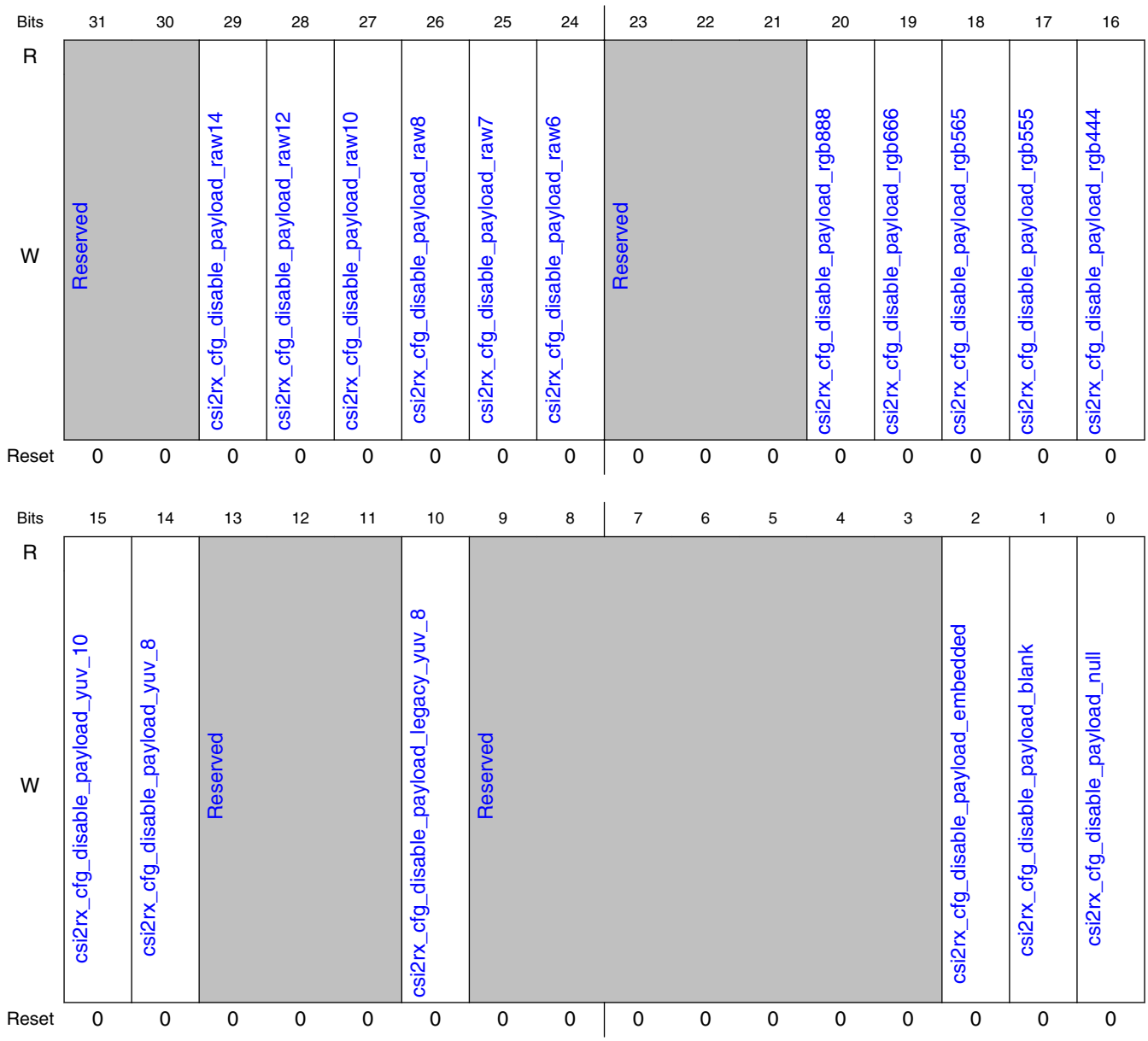
Field	Function
31-4 —	Reserved
3-0 csi2rx_ppi_errcontrol	CSI2 RX DPHY PPI ErrControl captured status from the DPHY. [0] – data lane 0 [1] – data lane 1 [2] – data lane 2 [3] – data lane 3

13.8.4.1.13 Disable Payload 0 Register (CSI2RX_CFG_DISABLE_PAYLOAD_0)

13.8.4.1.13.1 Offset

Register	Offset
CSI2RX_CFG_DISABLE_PAYLOAD_0	12Ch

13.8.4.1.13.2 Diagram



13.8.4.1.13.3 Fields

Field	Function
31-30 —	Reserved
29 csi2rx_cfg_disable_payload_raw14	RAW14

Table continues on the next page...

Field	Function
28 csi2rx_cfg_disable_payload_raw12	RAW12
27 csi2rx_cfg_disable_payload_raw10	RAW10
26 csi2rx_cfg_disable_payload_raw8	RAW8
25 csi2rx_cfg_disable_payload_raw7	RAW7
24 csi2rx_cfg_disable_payload_raw6	RAW6
23-21 —	Reserved
20 csi2rx_cfg_disable_payload_rgb888	RGB888
19 csi2rx_cfg_disable_payload_rgb666	RGB666
18 csi2rx_cfg_disable_payload_rgb565	RGB565
17 csi2rx_cfg_disable_payload_rgb555	RGB555
16 csi2rx_cfg_disable_payload_rgb444	RGB444
15 csi2rx_cfg_disable_payload_yuv_10	YUV422 10 bit

Table continues on the next page...

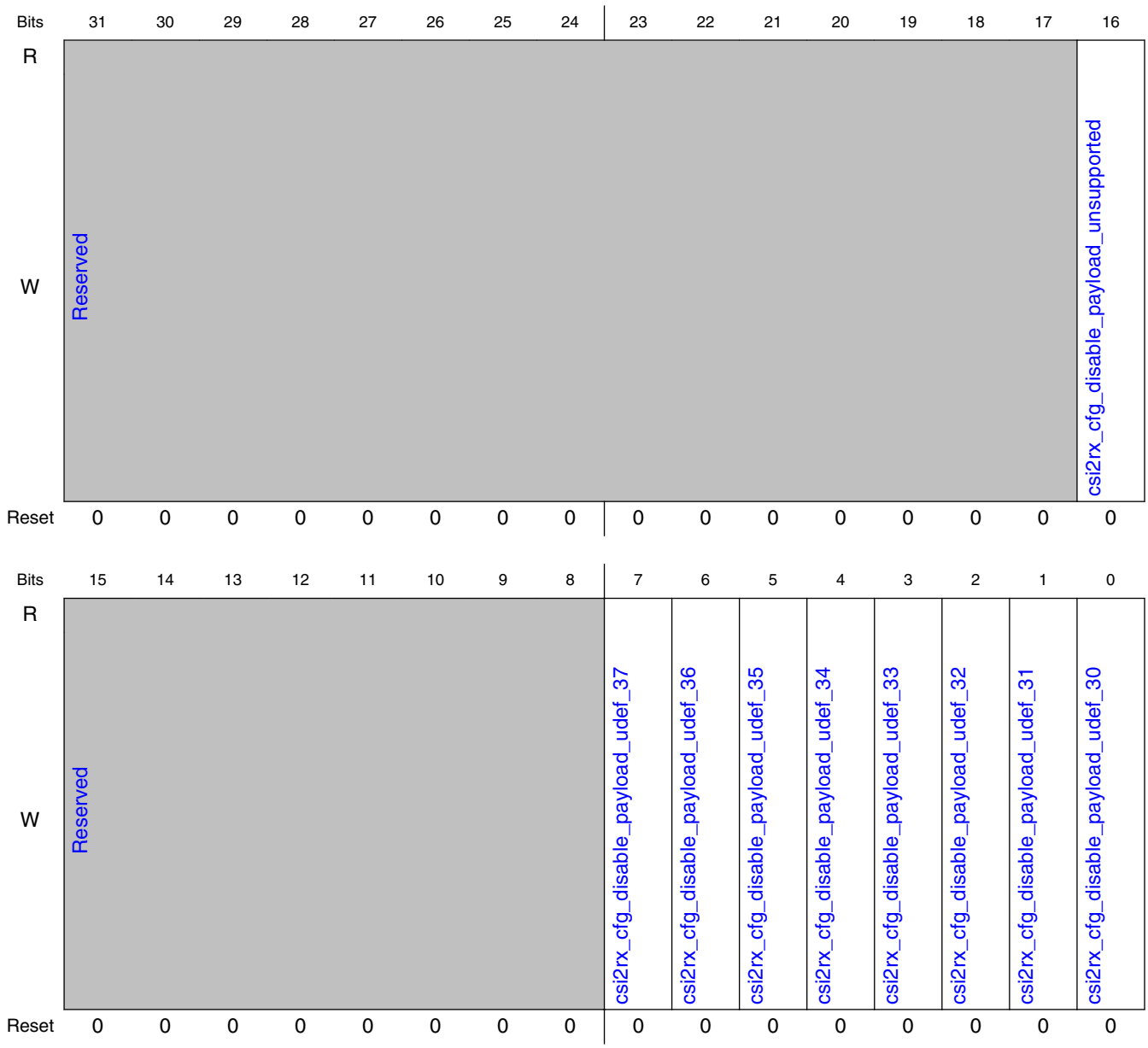
Field	Function
14 csi2rx_cfg_disable_payload_yuv_8	YUV422 8 bit
13-11 —	Reserved
10 csi2rx_cfg_disable_payload_legacy_yuv_8	Legacy YUV 420 8 bit
9-3 —	Reserved
2 csi2rx_cfg_disable_payload_embedded	Embedded
1 csi2rx_cfg_disable_payload_blank	Blank
0 csi2rx_cfg_disable_payload_null	Null

13.8.4.1.14 Disable Payload 1 Register (CSI2RX_CFG_DISABLE_PAYLOAD_1)

13.8.4.1.14.1 Offset

Register	Offset
CSI2RX_CFG_DISABLE_PAYLOAD_1	130h

13.8.4.1.14.2 Diagram



13.8.4.1.14.3 Fields

Field	Function
31-17 —	Reserved
16 csi2rx_cfg_disable_payload_unsupported	Unsupported Data Types

Table continues on the next page...

Field	Function
15-8 —	Reserved
7 csi2rx_cfg_disable_payload_udef_37	User defined type 0x37
6 csi2rx_cfg_disable_payload_udef_36	User defined type 0x36
5 csi2rx_cfg_disable_payload_udef_35	User defined type 0x35
4 csi2rx_cfg_disable_payload_udef_34	User defined type 0x35
3 csi2rx_cfg_disable_payload_udef_33	User defined type 0x34
2 csi2rx_cfg_disable_payload_udef_32	User defined type 0x33
1 csi2rx_cfg_disable_payload_udef_31	User defined type 0x32
0 csi2rx_cfg_disable_payload_udef_30	User defined type 0x31

13.8.4.1.15 FIFO Send Level Configuration Register (CSI2RX_CFG_VID_P_FIFO_SEND_LEVEL)

13.8.4.1.15.1 Offset

Register	Offset
CSI2RX_CFG_VID_P_FIFO_SEND_LEVEL	188h

13.8.4.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	cfg_vid_p_fifo_send_level															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.15.3 Fields

Field	Function
31-16 —	Reserved
15-0 cfg_vid_p_fifo_send_level	This input configures the number of entries that must accumulate in the Pixel FIFO before the data will be transferred to the video output. The exact value needed for this configuration is dependent on the rate at which the sensor transfers the data to the RX controller and the user video clock.

13.8.4.1.16 VSYNC Configuration Register (CSI2RX_CFG_VID_VSYNC)

13.8.4.1.16.1 Offset

Register	Offset
CSI2RX_CFG_VID_VSYNC	18Ch

13.8.4.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								cfg_vid_vsync							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.16.3 Fields

Field	Function
31-8 —	Reserved
7-0 cfg_vid_vsync	Width of VSYNC This input controls the width of the interface VSYNC pulse in video clocks. Pulse width = N+1 video clocks where N is the value of cfg_vid_vsync.

13.8.4.1.17 Start of HSYNC Delay control Register (CSI2RX_CFG_VID_HSYNC_FP)

13.8.4.1.17.1 Offset

Register	Offset
CSI2RX_CFG_VID_HSYNC_FP	190h

13.8.4.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								cfg_vid_hsync_fp							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.17.3 Fields

Field	Function
31-8 —	Reserved
7-0 cfg_vid_hsync_fp	Delay control for beginning of HSYNC pulse This input controls an arbitrary delay from when the fifo send level calls for a line of samples to be output and beginning of the HSYNC associated with the line. Pulse width = N+7 video clocks where N is the value of cfg_vid_hsync_fp.

13.8.4.1.18 HSYNC Configuration Register (CSI2RX_CFG_VID_HSYNC)

13.8.4.1.18.1 Offset

Register	Offset
CSI2RX_CFG_VID_HSYNC	194h

13.8.4.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								cfg_vid_hsync							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.18.3 Fields

Field	Function
31-8 —	Reserved
7-0 cfg_vid_hsync	Width of HSYNC This input controls the width of the interface HSYNC pulse in video clocks. Pulse width = N+1 video clocks where N is the value of cfg_vid_hsync.

13.8.4.1.19 End of HSYNC Delay Control Register (CSI2RX_CFG_VID_HSYNC_BP)

13.8.4.1.19.1 Offset

Register	Offset
CSI2RX_CFG_VID_HSYNC_BP	198h

13.8.4.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								cfg_vid_hsync_bp							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.8.4.1.19.3 Fields

Field	Function
31-8 —	Reserved
7-0 cfg_vid_hsync_bp	Delay Control for end of HSYNC pulse This input controls an arbitrary delay from the end of the H sync pulse to the assertion of the first data enable(EN). Pulse width = N+5 video clocks where N is the value of cfg_vid_hsync_bp.

13.9 Sony/Philips Digital Interface (SPDIF)

13.9.1 Overview

The Sony/Philips Digital Interface (SPDIF) audio block is a stereo transceiver that allows the processor to receive and transmit digital audio.

The SPDIF transceiver allows the handling of both SPDIF channel status (CS) and User (U) data and includes a frequency measurement block that allows the precise measurement of an incoming sampling frequency.

A recovered clock is provided to drive both internal components in the system, such as SAI ports, and external components, such as A/Ds or D/As, with clocking control provided via related registers.

As the SPDIF internal data width is 24-bit, the eight most-significant bits of all registers return zeros.

The figure below shows a block diagram of the SPDIF transceiver data paths (receiver and transmitter) and its interface.

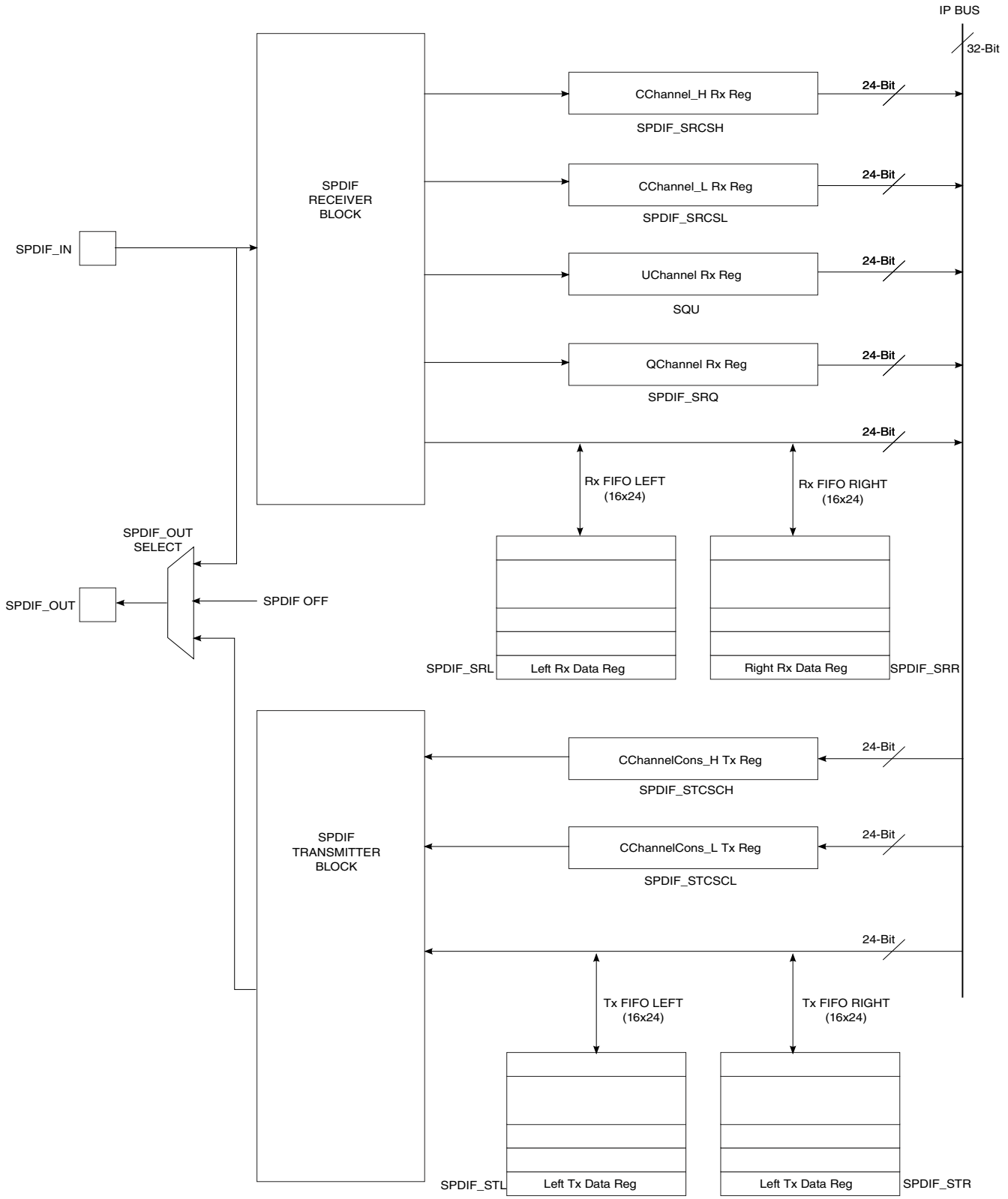


Figure 13-80. SPDIF Transceiver Data Interface Block Diagram

13.9.2 Clocks

The table found here describes the clock sources for SPDIF.

Please see clock control block for clock setting, configuration and gating information.

Table 13-38. SPDIF Clocks

Clock name	Clock Root	Description
gclkw_t0	ipg_clk_root	Global clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
tx_clk	spdif0_clk_root	Module Tx clock

13.9.3 Functional Description

The SPDIF is composed of two parts: SPDIF Receiver and SPDIF Transmitter.

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in the SPDIF Rx left and right FIFOs. The Channel Status and User Bits are also extracted from each frame and placed in the corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

For the SPDIF transmitter, the audio data is provided by the processor via the SPDIFTxLeft and SPDIFTxRight registers. The Channel Status bits are also provided via the corresponding registers. The SPDIF transmitter generates a SPDIF output bitstream in the biphase mark format (IEC60958), which consists of audio data, channel status and user bits.

In the SPDIF transmitter, the IEC60958 biphase bit stream is generated on both edges of the SPDIF Transmit clock. The SPDIF Transmit clock is generated by the SPDIF internal clock generate block and the sources are from outside of the SPDIF block. For the SPDIF receiver, it can recover the SPDIF Rx clock. Figure 1 shows the clock structure of the SPDIF transceiver.

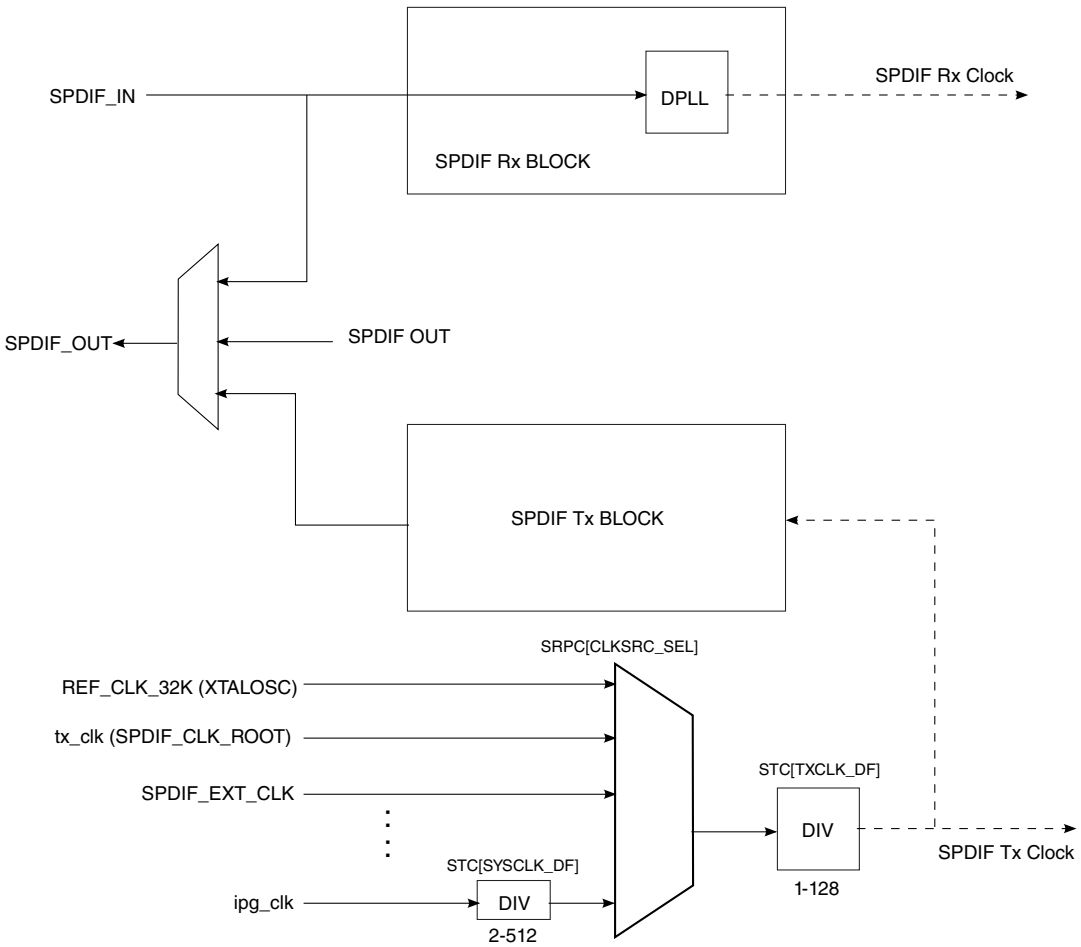


Figure 13-81. SPDIF Transceiver Clock Diagram

13.9.3.1 SPDIF Receiver

The SPDIF receiver extracts the audio data from each SPDIF frame and places the data in Rx left and right FIFOs.

The Tx left and right FIFOs are 16-deep and 24-bit-wide (equal to the audio data width). The Channel Status and User Bits are also extracted from each frame and placed in corresponding registers. The SPDIF receiver also provides a bypass option for direct transfer of the SPDIF input signal to the SPDIF transmitter.

The SPDIF receiver handles the main data audio stream and recovers the bit clock from the SPDIF input signal. The sample rate can be determined from the frequency measuring block. Additionally, the receiver supports the SPDIF C and U channels. The SPDIF C and U channel data is interfaced directly to memory-mapped registers.

All the data registers are controlled by the Interrupt Control Block and transferred to the memory-mapped IP bus.

The following functions are performed by the SPDIF receiver:

- Audio Data Reception see Audio Data Reception
- Channel Status bits Reception see Channel Status Reception
- U Channel bits Reception see User Bit Reception
- Validity Flag Reception see Validity Flag Reception
- SPDIF Receiver Exception support see SPDIF Receiver
- SPDIF Lock Detection

13.9.3.1.1 Audio Data Reception

The SPDIF Receiver block extracts the audio data from the IEC60958 stream, and outputs this via Rx left and right FIFOs to the memory-mapped registers SPDIFRxLeft and SPDIFRxRight.

Data from the SPDIF receiver is buffered in receive FIFO, and can be read by the processor from the memory-mapped registers.

- **SPDIF receiver data registers - Behavior on overrun, underrun**

The SPDIF Data Receive registers (SPDIFRxLeft and SPDIFRxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFOs. To prevent this from happening, hardware has been added to the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If a SPDIF Data Rx FIFO overrun occurs on e.g. the right half of the FIFO, the sample that caused the overrun is not written to the right half (due to overrun). Special hardware will make sure the next sample is not written to the left half of the FIFO. If the overrun occurs on the left half of the FIFO, the next sample is not written to the right half of the FIFO.

- **SPDIF receiver data registers - Automatic resynchronization of FIFOs**

An automatic FIFO resynchronization feature is available. It can be enabled and disabled separately for every FIFO. If it is enabled, the hardware will check to see if the left and right FIFOs are in sync. If that is not the case, it will set the filling pointer of the right FIFO to be equal to the filling pointer of the left FIFO.

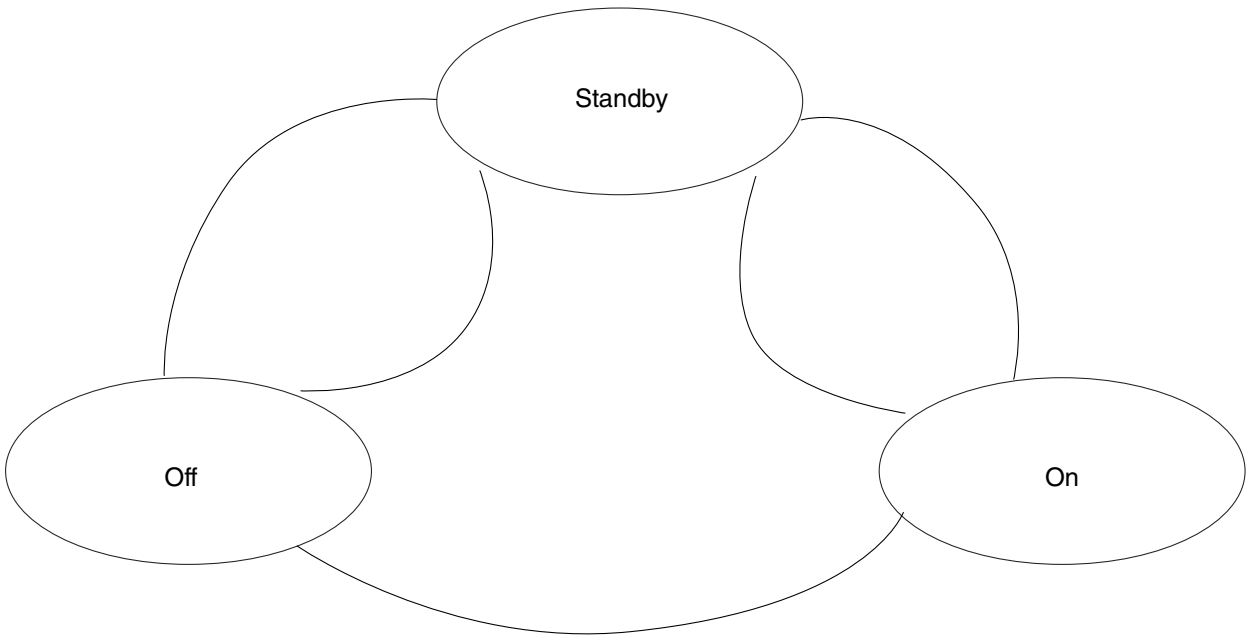


Figure 13-82. FIFO Auto-resync Controller State Machine

The operation is explained from the state diagram shown above. Every FIFO auto-resync controller has a state machine with 3 states: Off, StandBy and On. In the On state, the filling of the left FIFO is compared with the filling of right, and if they are not equal, right is made equal to left, and an interrupt is generated.

The controller will stay in Off state when the feature is disabled. When not disabled, the state machine will go to Off state on any processor read or write to the FIFO. It will go from On or Off to Standby on any left sample read from SPDIF Tx FIFOs, or on any left sample write to SPDIF Rx FIFOs. The controller will go from Standby to On on any right sample read from SPDIF Tx FIFO, or on any right sample write to SPDIF Rx FIFO. There is a control bit in the SPDIFConfig register to enable/disable the feature for the SPDIF Rx FIFO and SPDIF Tx FIFO.

13.9.3.1.1.1 Application Note

The automatic FIFO resynchronization can be switched on, and will avoid all mismatches between left and right FIFOs, if the software obeys the following rules: 1. When the left data is read or written to the left FIFO, in the same place of the program, data must be read or written to the right FIFO. Maximum time difference between left and right is 1/2 sample clock. (E.g. if sample frequency is 44 Khz, approximately 10 micro-seconds. For 88 Khz, approximately 5 micro-seconds.) 2. Write/read data to FIFO s at least 2 samples

at the time. If there is a mismatch Left-Right, the resync logic may go on only 1 sample clock after last data is read/written to the FIFO. Also acceptable is polling the FIFO, if at least part of the time 2 samples will be read/written to it.

- **SPDIF receiver - Additional features**

There are three exceptions associated with the SPDIF Receivers FIFOs

- full
- under/overrun
- resync

When the "full" condition is set for processor data input registers, the processor should read data from the FIFO, before overrun occurs. When "full" is set, and the FIFO contains e.g. 6 samples, it is acceptable for the software to read first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 6 samples from the RIGHT address, followed by 6 samples from the LEFT address, or 1 sample LEFT, followed by 1 sample RIGHT repeated 6 times. There is no order specified.

The implementation for SPDIF Rx is a double FIFO, one for left and one for right. "full" is set when both FIFOs are full. "underrun, overrun" are set when one of the FIFOs do underrun or do overrun. The resync interrupt means hardware took special action to resynchronize left and right FIFOs.

The FIFO level at which the "full" interrupt is generated, is programmable via the Full Select field in the SPDIFConfigReg register.

Rx FIFO on and Rx FIFO reset.

Two additional control fields of the SPDIF Rx FIFO are the on/off select and FIFO reset fields.

If on/off select is set to off, all-zero will be read from the FIFO, irrespective of the data received over the SPDIF interface.

If FIFO reset is set, the FIFO is blocked at "1 sample in FIFO". In this, the full interrupt will be on if FullSelect is set to "00". If FullSelect is set to any other value, interrupt will be off. The other interrupts are always off.

13.9.3.1.2 Channel Status Reception

A total of 48 channel status bits are received in two registers. No interpretation is performed by the SPDIF receiver block.

Channel Status Bits are ordered first bit left. CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFRxCChannel_h. CS-channel bit "23" is considered the LSB bit 0 in the register. C-channel bit 24 to 47 is seen as [23:0] bits of register SPDIFRxCChannel_l.

13.9.3.1.2.1 Channel Status Interrupt

When the value of a new SPDIF "CS" channel status frame is loaded in the register, an interrupt is generated. The interrupt is cleared when the processor writes the corresponding bit in the InterruptStat register.

13.9.3.1.3 User Bit Reception

There are two modes for U Channel reception, CD and non-CD. As is decided by USyncMode (bit 1 of CDText_Control register).

- **Behavior of U Channel receive interface on incoming CD U Channel Sub-code in SPDIF receiver**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set "1".

The CD sub-code stream embedded into the SPDIF U channel consists of a sequence of packets. Every packet is made up 98 "symbols". The first two symbols of every packet are "sync symbols", the other 96 symbols are "data symbols".

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

Data symbols are coming in MSB first. The MSB is the leading one.

When a "long pause" is seen between 2 subsequent "data symbols", the SPDIF receiver will assume the reception of one or more "sync symbols". Table below gives details.

Table 13-39. Sync Control Bits

Number of U Channel zero bits	Corresponding number of sync symbols
0-1	Unpredictable, not allowed
2-10	0
11-22	1
23-34	2
35-46	3
>45	Unpredictable, not allowed

The recognition of the number of sync symbols derives from the fact that the U channel transmitter in the CD channel decoder will transmit one symbol on average every 12 SPDIF channel bits. On this average rate, there is a maximum tolerance of 5%.

The SPDIF receiver is tolerant of symbol errors. Due to the physical nature of the transmission of the data over the CD disc, not more than 1 out of any 5 consecutive user channel symbols may be in error. The error may cause a change in data value, which is not detected by this interface, or it may cause a data symbol to be seen as a sync symbol, or a sync symbol to be seen as a data symbol. However, not more than 1 out of any 5 consecutive user channel symbols should be affected in this way.

The SPDIF U channel circuitry recognizes the 98-symbol packet structure, and sends the 96 symbol payload to the processor application. The 96 symbol payload is transmitted to the processor via 2 registers:

- The SPDIFRxUChannel register. In this register, data is presented 3 symbols at the time to the processor. Every time 3 new valid symbols, received on the SPDIF U Channel are present, the UChannelRxFull interrupt is asserted. For one 98-symbol packet, 96 symbols are carried across SPDIFRxUChannel. To transfer all this data, 32 UChannelRxFull interrupts are generated.
- The QChannelReceive register. In this register, only the Q bit of the packet is accumulated.Operation is similar to UChannelReceive. Because only Q-bit is transferred, only 96 Q-bits are transferred for any 98-symbol packet. To transfer this data, 4 QChannelRxFull interrupts are generated. When QChannelRxFull occurs, it is coincident with UChannelRxFull. There is only one QChannelRxFull for every 8 UChannelRxFull. The convention is that most significant data is transmitted first, and is left-aligned in the registers.
- Timing regarding packet boundary is extracted by hardware. The last UChannelRxFull corresponding to a given packet should be coincident with the last QChannelRxFull. In this last U, Q channel interrupt, symbols 95-98 are received, Q channel bits 67-98. The interrupts are coincident with UQSyncFound, flagging last symbols of the current frame.
- When the start of the new packet is found before the current packet is complete (less than 98 symbols in the packet), the UQFrameError interrupt is set. The application software should read out UChannelReceive and QchannelReceive registers, discard the value, and assume the start of a new packet.
- As already said, packet sync extraction is tolerant for single-symbol errors. Packet sync detection is based on the recognition of the sequence data-sync-sync-data in the symbol stream, because this is the only syncing sequence that is not affected by single errors. If the sync symbols are not found 98 symbols after the previous

occurrence, it is assumed to be destroyed by channel error, and a new sync symbols is interpolated.
- Normally, only data bytes are passed to the application software. Every databyte will have its most significant bit set. If sync symbols are passed to the application software, they are seen as all-zero symbols. Sync symbols can only end up in the data stream due to channel error.

- **Behavior of U Channel receive interface on incoming non-CD data**.

This mode is selected if UsyncMode, bit 1 in register CD Text control is set'0'.

In non-CD mode, the SPDIF U channel stream is recognized as a sequence of "data symbols". No packet recognition is done.

Any sequence found in the SPDIF U channel stream starting with a leading one, followed by 7 information bits, is recognized as a "data symbol". Subsequent data symbols are separated by "pauses". During the "pause", "zero bits" are seen on the SPDIF U channel.

3 consecutive data symbols seen in the SPDIF U Channel stream are grouped together into the SPDIFRxUChannel register. First symbol is left, last symbol is right aligned. When SPDIFRxUChannel contains 3 new data symbols, UChannelRxFull is asserted.

In this mode, the operation of QchannelRx and associated interrupt QchannelRxFull is reserved, undefined. And the operation of UQFrameError and UQSyncFound is also reserved, undefined.

The U channel is extracted, and output by the SPDIF Rx on SPDIFRxUChannel-Stream.

When incoming SPDIF data parity error or bit error is detected, and if the next SPDIF word for that channel is error-free, the SPDIF word in error is replaced with the average of the previous word and next word. When incoming SPDIF data parity error or bit error is detected, and the next SPDIF word is in error, the previous SPDIF word is repeated.

13.9.3.1.4 Validity Flag Reception

An interrupt is associated with the Validity flag. (interrupt 16 - SPDIFValNoGood). This interrupt is set every time a frame is seen on the SPDIF interface with the validity bit set to "invalid".

13.9.3.1.5 SPDIF Receiver Interrupt Exception Definition

Several SPDIF exceptions can trigger an interrupt.

They are:

- Control Status channel change. Set when SPDIFRxCChannel_1 register is updated. The register is updated for every new C-Channel received. The exception is reset on write to InterruptClear register.
- SPDIF Illegal Symbol. Set on reception of illegal symbol during SPDIF receive. Reset by writing register InterruptClear.[1]
- SPDIF bit error. Set on reception of bit error. (Parity bit does not match). Reset on write to InterruptClear register.
- Receive data FIFO full. Set when SPDIF receive data FIFO is full.
- Receive data FIFO underrun/overrun. Set when there is a underrun/overrun on the SPDIF receive data FIFO.
- Receive data FIFO resynchronization. Set when a resynchronization event occurs on the SPDIF receive data FIFO.
- Receive U Channel buffer full. Set when next 24 bits of U channel code are available.
- Receive Q Channel buffer overrun. Set when Q channel buffer overrun.
- Receive U Channel buffer overrun. Set on U channel buffer overrun.
- Receive Q Channel buffer full. Set when next 24 bits of Q channel code are available.
- Receive UQ sync found. Set when UQ channel sync found.
- Receive UQ frame error. Set when UQ frame error found.

13.9.3.1.6 Standards Compliance

The SPDIF interface is compatible with the Tech 3250-E standard of the European Broadcasting Union, except clause 6.3.3 and the IEC60958-3 Ed2 for relevant topics.

Supported input frequency range is 12 Khz up to 96 Khz. (fully compliant) and 96 Khz up to 176 Khz (Can interface with compliant SPDIF transmitter within same cabinet, making reasonable assumptions on jitter added due to interconnecting wire.)

Tolerated jitter on SPDIF input signals are 0.25 bit peak-peak for high frequencies. There is no jitter limit for low frequencies. The user channel extraction in CD mode is capable of coping with single-symbol errors, and still retrieve U channel frames on correct boundaries. This capability is required for reliable reception of CD-Text from some Philips CD channel decoders. This capability was deemed more important than compliance with the IEC60958 annex A.3 standard, and for this reason user channel

1. The SPDIF input is a biphase/mark modulated signal. The time between any two successive transitions of the SPDIF signal is always 1, 2 or 3 SPDIF symbol periods long. The SPDIF receiver will parse the stream, and split it in so-called symbols. It recognizes s1, s2 and s3 symbols, depending on the length of the symbols. Not all sequences of these symbols are allowed. To give an example, a sequence s2-s1-s1-s1-s2 cannot occur in a no-error SPDIF signal. If the receiver finds such an illegal sequence, the illegal symbol interrupt is set. No corrective action is undertaken. When the interrupt occurs, this means that(a) The SPDIF signal is destroyed by noise (b) The SPDIF frequency changed.

reception is not compliant with IEC60958 annex A.3. However, the interface is capable to receive U channel inserted by a typical CD channel decoder. Also, in this case, it is more robust and tolerant for channel error than what is required by IEC60958 annex A.3.

13.9.3.1.7 SPDIF PLOCK Detection and Rxclk Output

Using the high speed system clock, the internal DPLL can extract the bit clock (advanced pulse) from the input bitstream. When this internal DPLL is locked, the LOCK bit of PhaseConfig Register will be set, and the SPDIF Lock output pin SPDIF_LOCK will be asserted.

After DPLL has locked, the pulses are generated, and the average pulse rate is 128 x the sampling frequency.(For a 44.1 Khz input sampling frequency, the average pulse rate = 128 x 44.1 Khz.) The pulse signal is used in the FreqMeas circuit to generate the frequency measurement result.

13.9.3.1.8 Measuring Frequency of SPDIF_RxClk

The internal DPLL can extract the bit clock (advanced plus) from the input bitstream. To do that, it is necessary to measure the frequency of the incoming signal in relationship with the system clock (BUS_CLK).

Associated with it are two registers, PhaseConfig and FreqMeas. The circuit will measure the frequency of the incoming clock as a function of the BUS_CLK. The circuit is a second-order filter. The output is a value represented by an unsigned number stored in the 24-bit FreqMeas register, giving the frequency of the source as a function of the BUS_CLK.

FreqMeas[23:0] = FreqMeas_CLK / BUS_CLK * 2^{10} * GAIN.

For example, if the GAIN is selected as 8*(2**10) (PhaseConfig[5:3] = 3'b011), the actual result

FreqMeas_CLK / BUS_CLK is equal to FreqMeas[23:0] / 2^{23}.

13.9.3.2 SPDIF Transmitter

Audio data for the SPDIF transmitter is provided by processor via the SPDIFTxLeft and SPDIFTxRight registers.

Clocking for SPDIF transmitter is selected through a multiplexer from several clock sources (see TxClk_Source for clock source inputs). The SPDIF transmitter clock source can be divided down as needed using Txclk_DF. The SPDIF transmitter output can be chosen from either the SPDIF transmitter block, directly from the SPDIF receiver (via the output multiplexer), or disabled.

The SPDIF transmitter generates a SPDIF output bitstream in IEC60958 biphase mark format, consisting of audio data, channel status.

13.9.3.2.1 Audio Data Transmission

Audio data for the SPDIF transmitter is provided by the processor via SPDIFTxLeft and SPDIFTxRight registers. They send audio data to Tx left and right FIFOs. The Tx left and right FIFOs are also 16-deep and 24-width (equal to the audio data width).

- **SPDIF transmitter data registers - Behavior on overrun, underrun**

The SPDIF Data Transmit registers (SPDIFTxLeft and SPDIFTxRight) have individual FIFOs for left and right channel. As a result, there is always the possibility that left and right FIFOs may go out of sync due to FIFO underruns and FIFO overruns that affect only one part (left or right) of any FIFO. To prevent this from happening, hardware has been added on the device. Two mechanisms to prevent mismatch between the FIFOs are available.

If SPDIF Tx FIFO underruns on the right half of the FIFO, no sample leaves that FIFO (because it was already empty). Special hardware will make sure that the next sample read from the left FIFO will not leave the FIFO (no read strobe is generated). If the underrun occurs on the left half of the FIFO, next read strobe to the right FIFO is blocked.

- **SPDIF transmitter data registers - Automatic resynchronization of FIFOs**

See Audio Data Reception.

- **SPDIFTxLeft, SPDIFTxRight details**

With SPDIF Tx FIFOs three exceptions are associated.

- empty
- under/overrun
- resync

When the empty condition is set for processor data output registers, the processor should write data to the FIFO, before underrun occurs. When empty is set and, for instance, 6 samples need to be written, it is acceptable for the software to write first 6 samples from the LEFT address, followed by 6 samples from the RIGHT address, or 1 sample LEFT,

followed by 1 sample RIGHT repeated 6 times. Left should be written before right. The implementation of all data out FIFOs is a double FIFO, one for left and one for right. Empty is set when both FIFOs are empty. Underrun, overrun are set when one of the FIFOs do underrun or do overrun. Resync is set when the hardware resynchronizes left and right FIFOs.

On receiving underrun, overrun interrupt, synchronization between Left and Right words in the FIFOs may be lost. Synchronization will not be lost when the underrun or overrun comes from the IEC60958 side of the FIFO. If the processor reads or writes more data from, for example, left than from right, synchronization will be lost. If automatic resynchronization is enabled, and if the software obeys the rules to let this work, resynchronization will be automatic.

13.9.3.2.2 Channel Status Transmission

A total of 48 Consumer channel status bits are transmitted from two registers. Channel Status Bits are ordered first bit left.

CS-channel MSB bit "0" is located in bit position 23 in the memory-mapped register SPDIFTxCChannelCons_h. CS-channel bit "23" is considered bit 0 in the register. C-channel bits 24-47 are seen as MSB-LSB bits of register SPDIFTxCChannelCons_l.

13.9.3.2.3 Validity Flag Transmission

The validity bit setting is performed via bit 5 of the SPDIF_SCR register.

13.9.4 SPDIF Memory Map/Register Definition

SPDIF memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3081_0000	SPDIF Configuration Register (SPDIF1_SCR)	32	R/W	0000_0400h	13.9.4.1/ 5043
3081_0004	CDText Control Register (SPDIF1_SRCD)	32	R/W	0000_0000h	13.9.4.2/ 5045
3081_0008	PhaseConfig Register (SPDIF1_SRPC)	32	R/W	0000_0000h	13.9.4.3/ 5046
3081_000C	InterruptEn Register (SPDIF1_SIE)	32	R/W	0000_0000h	13.9.4.4/ 5047
3081_0010	InterruptStat Register (SPDIF1_SIS)	32	R	0000_0002h	13.9.4.5/ 5049

Table continues on the next page...

SPDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3081_0010	InterruptClear Register (SPDIF1_SIC)	32	W	0000_0000h	13.9.4.6/ 5051
3081_0014	SPDIFRxLeft Register (SPDIF1_SRL)	32	R	0000_0000h	13.9.4.7/ 5052
3081_0018	SPDIFRxRight Register (SPDIF1_SRR)	32	R	0000_0000h	13.9.4.8/ 5053
3081_001C	SPDIFRxCChannel_h Register (SPDIF1_SRCSH)	32	R	0000_0000h	13.9.4.9/ 5053
3081_0020	SPDIFRxCChannel_l Register (SPDIF1_SRCSL)	32	R	0000_0000h	13.9.4.10/ 5054
3081_0024	UchannelRx Register (SPDIF1_SRU)	32	R	0000_0000h	13.9.4.11/ 5054
3081_0028	QchannelRx Register (SPDIF1_SRQ)	32	R	0000_0000h	13.9.4.12/ 5055
3081_002C	SPDIFTxLeft Register (SPDIF1_STL)	32	W	0000_0000h	13.9.4.13/ 5055
3081_0030	SPDIFTxRight Register (SPDIF1_STR)	32	W	0000_0000h	13.9.4.14/ 5056
3081_0034	SPDIFTxCChannelCons_h Register (SPDIF1_STCSCH)	32	R/W	0000_0000h	13.9.4.15/ 5056
3081_0038	SPDIFTxCChannelCons_l Register (SPDIF1_STCSCL)	32	R/W	0000_0000h	13.9.4.16/ 5057
3081_0044	FreqMeas Register (SPDIF1_SRFM)	32	R	0000_0000h	13.9.4.17/ 5057
3081_0050	SPDIFTxClk Register (SPDIF1_STC)	32	R/W	0002_0F00h	13.9.4.18/ 5058
308A_0000	SPDIF Configuration Register (SPDIF2_SCR)	32	R/W	0000_0400h	13.9.4.1/ 5043
308A_0004	CDText Control Register (SPDIF2_SRCD)	32	R/W	0000_0000h	13.9.4.2/ 5045
308A_0008	PhaseConfig Register (SPDIF2_SRPC)	32	R/W	0000_0000h	13.9.4.3/ 5046
308A_000C	InterruptEn Register (SPDIF2_SIE)	32	R/W	0000_0000h	13.9.4.4/ 5047
308A_0010	InterruptStat Register (SPDIF2_SIS)	32	R	0000_0002h	13.9.4.5/ 5049
308A_0010	InterruptClear Register (SPDIF2_SIC)	32	W	0000_0000h	13.9.4.6/ 5051
308A_0014	SPDIFRxLeft Register (SPDIF2_SRL)	32	R	0000_0000h	13.9.4.7/ 5052
308A_0018	SPDIFRxRight Register (SPDIF2_SRR)	32	R	0000_0000h	13.9.4.8/ 5053
308A_001C	SPDIFRxCChannel_h Register (SPDIF2_SRCSH)	32	R	0000_0000h	13.9.4.9/ 5053

Table continues on the next page...

SPDIF memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
308A_0020	SPDIFRxCChannel_l Register (SPDIF2_SRCSL)	32	R	0000_0000h	13.9.4.10/ 5054
308A_0024	UchannelRx Register (SPDIF2_SRU)	32	R	0000_0000h	13.9.4.11/ 5054
308A_0028	QchannelRx Register (SPDIF2_SRQ)	32	R	0000_0000h	13.9.4.12/ 5055
308A_002C	SPDIFTxLeft Register (SPDIF2_STL)	32	W	0000_0000h	13.9.4.13/ 5055
308A_0030	SPDIFTxRight Register (SPDIF2_STR)	32	W	0000_0000h	13.9.4.14/ 5056
308A_0034	SPDIFTxCChannelCons_h Register (SPDIF2_STCSCH)	32	R/W	0000_0000h	13.9.4.15/ 5056
308A_0038	SPDIFTxCChannelCons_l Register (SPDIF2_STCSCL)	32	R/W	0000_0000h	13.9.4.16/ 5057
308A_0044	FreqMeas Register (SPDIF2_SRFM)	32	R	0000_0000h	13.9.4.17/ 5057
308A_0050	SPDIFTxClk Register (SPDIF2_STC)	32	R/W	0002_0F00h	13.9.4.18/ 5058

13.9.4.1 SPDIF Configuration Register (SPDIFx_SCR)

Address: Base address + 0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								RxFIFO_Ctrl	RxFIFO_Off_On	RxFIFO_Rst	RxFIFOFull_Sel		RxAutoSync	TxAutoSync	TxFIFOEmpty_Sel
Reset									0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TxFIFOEmpty_Sel	Reserved	LOW_POWER	soft_reset	TxFIFO_Ctrl		DMA_Rx_En	DMA_TX_En	Reserved		ValCtrl	TxSel			USrc_Sel	
Reset	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0

SPDIFx_SCR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 RxFIFO_Ctrl	0 Normal operation 1 Always read zero from Rx data register
22 RxFIFO_Off_On	0 SPDIF Rx FIFO is on 1 SPDIF Rx FIFO is off. Does not accept data from interface
21 RxFIFO_Rst	0 Normal operation 1 Reset register to 1 sample remaining
20–19 RxFIFOFull_Sel	00 Full interrupt if at least 1 sample in Rx left and right FIFOs 01 Full interrupt if at least 4 sample in Rx left and right FIFOs 10 Full interrupt if at least 8 sample in Rx left and right FIFOs 11 Full interrupt if at least 16 sample in Rx left and right FIFO
18 RxAutoSync	0 Rx FIFO auto sync off 1 RxFIFO auto sync on
17 TxAutoSync	0 Tx FIFO auto sync off 1 Tx FIFO auto sync on
16–15 TxFIFOEmpty_Sel	00 Empty interrupt if 0 sample in Tx left and right FIFOs 01 Empty interrupt if at most 4 sample in Tx left and right FIFOs 10 Empty interrupt if at most 8 sample in Tx left and right FIFOs 11 Empty interrupt if at most 12 sample in Tx left and right FIFOs
14 -	This field is reserved. Reserved
13 LOW_POWER	When write 1 to this bit, it will cause SPDIF enter low-power mode. return 1 when SPDIF in Low-Power mode.
12 soft_reset	When write 1 to this bit, it will cause SPDIF software reset. The software reset will last 8 cycles. When in the reset process, return 1 when read. else return 0 when read.
11–10 TxFIFO_Ctrl	00 Send out digital zero on SPDIF Tx 01 Tx Normal operation 10 Reset to 1 sample remaining 11 Reserved
9 DMA_Rx_En	DMA Receive Request Enable (RX FIFO full)
8 DMA_TX_En	DMA Transmit Request Enable (Tx FIFO empty)
7–6 -	This field is reserved. Reserved
5 ValCtrl	0 Outgoing Validity always set 1 Outgoing Validity always clear
4–2 TxSel	000 Off and output 0 001 Feed-through SPDIFIN

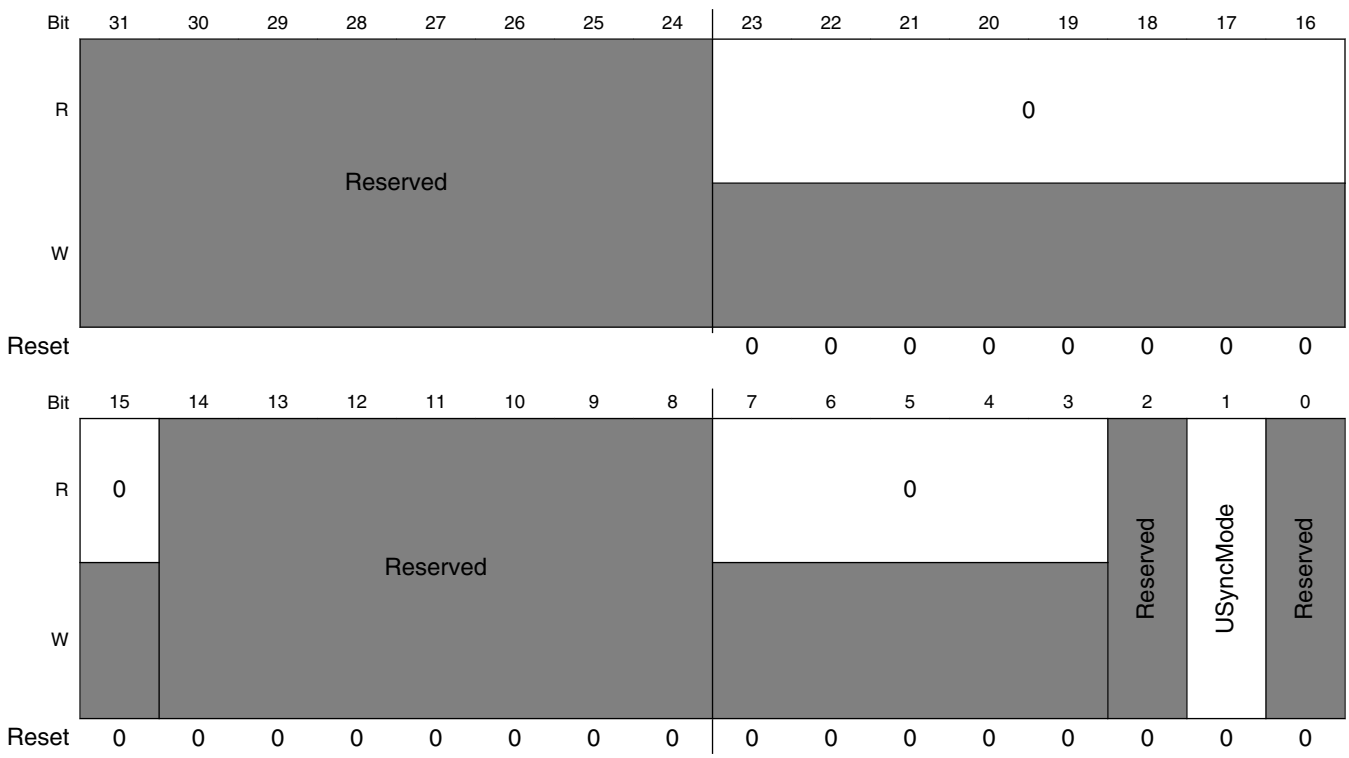
Table continues on the next page...

SPDIFx_SCR field descriptions (continued)

Field	Description
	101 Tx Normal operation Others Reserved
USrc_Sel	00 No embedded U channel 01 U channel from SPDIF receive block (CD mode) 10 Reserved 11 U channel from on chip transmitter

13.9.4.2 CDText Control Register (SPDIFx_SRCD)

Address: Base address + 4h offset



SPDIFx_SRCD field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–15 Reserved	This read-only field is reserved and always has the value 0.
14–8 -	This field is reserved. Reserved. set to zero.
7–3 Reserved	This read-only field is reserved and always has the value 0.

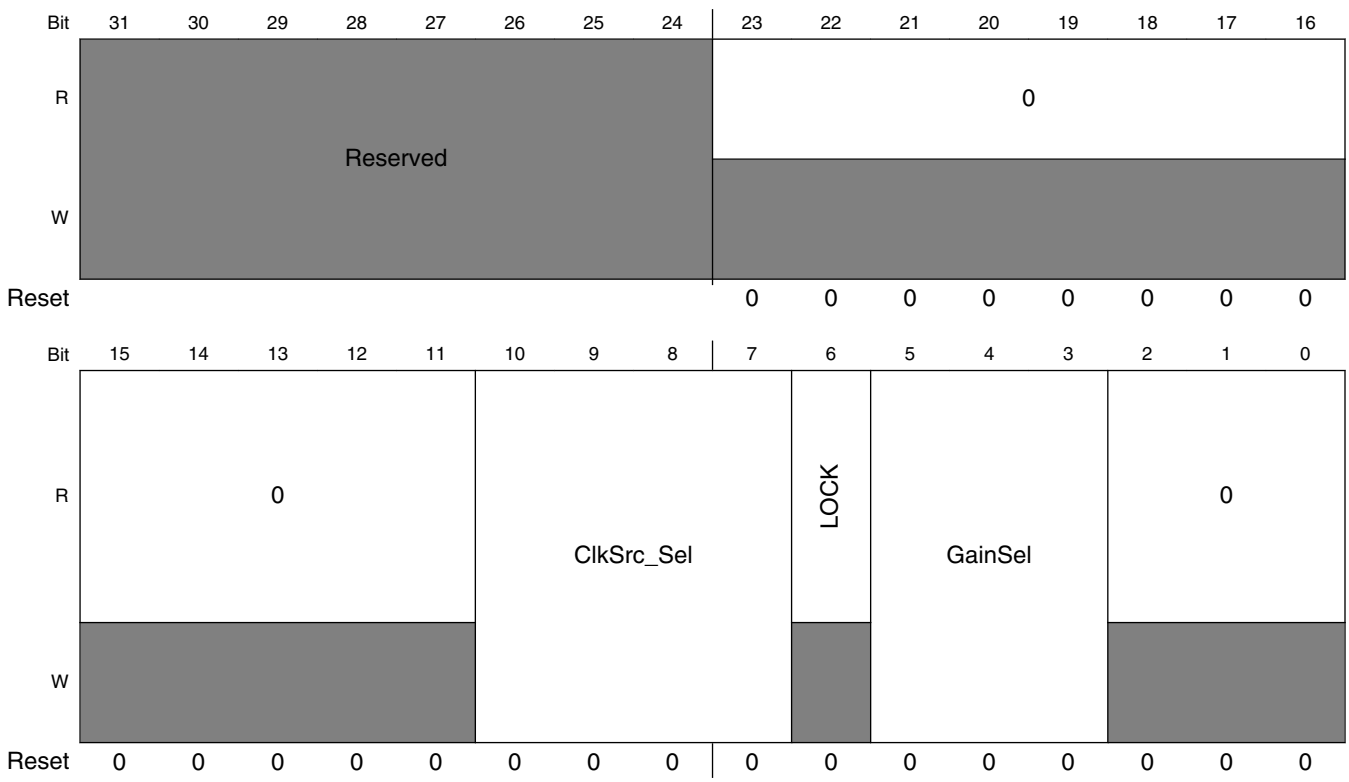
Table continues on the next page...

SPDIFx_SRCD field descriptions (continued)

Field	Description
2 -	This field is reserved. Reserved.
1 USyncMode	0 Non-CD data 1 CD user channel subcode
0 -	This field is reserved. Reserved.

13.9.4.3 PhaseConfig Register (SPDIFx_SRPC)

Address: Base address + 8h offset



SPDIFx_SRPC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–11 Reserved	This read-only field is reserved and always has the value 0.
10–7 ClkSrc_Sel	Clock source selection, all other settings not shown are reserved: 0000 if (DPLL Locked) SPDIF_RxClk else REF_CLK_32K (XTALOSC)

Table continues on the next page...

SPDIFx_SRPC field descriptions (continued)

Field	Description
	0001 if (DPLL Locked) SPDIF_RxClk else tx_clk (SPDIF0_CLK_ROOT) 0011 if (DPLL Locked) SPDIF_RxClk else SPDIF_EXT_CLK 0101 REF_CLK_32K (XTALOSC) 0110 tx_clk (SPDIF0_CLK_ROOT) 1000 SPDIF_EXT_CLK
6 LOCK	LOCK bit to show that the internal DPLL is locked, read only
5–3 GainSel	Gain selection: 000 24*(2**10) 001 16*(2**10) 010 12*(2**10) 011 8*(2**10) 100 6*(2**10) 101 4*(2**10) 110 3*(2**10)
Reserved	This read-only field is reserved and always has the value 0.

13.9.4.4 InterruptEn Register (SPDIFx_SIE)

The InterruptEn register (SPDIF_SIE) provides control over the enabling of interrupts.

Address: Base address + Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								0	Reserved		Lock	TxUnOv	TxResyn	CNew	ValNoGood
W																
Reset									0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SymErr	BitErr	Reserved			URxFul	URxOv	QRxFul	QRxOv	UQSync	UQErr	RxFIFOUnOv	RxFIFOResyn	LockLoss	TxEm	RxFIFOFul
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

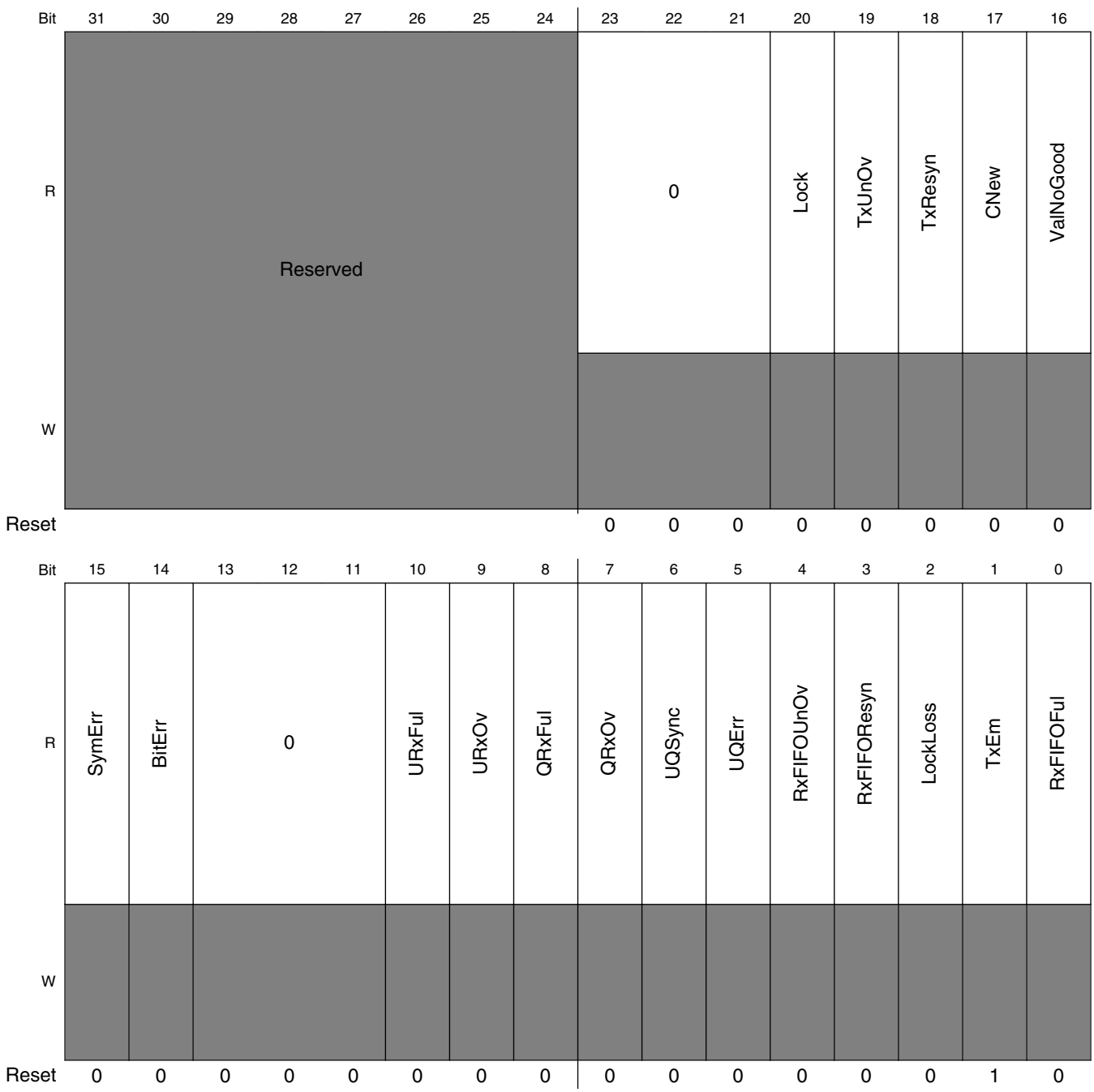
SPDIFx_SIE field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23 Reserved	This read-only field is reserved and always has the value 0.
22–21 -	This field is reserved. Reserved. set to zero.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 -	This field is reserved. Reserved. set to zero.
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

13.9.4.5 InterruptStat Register (SPDIFx_SIS)

The InterruptStat (SPDIF_SIS) register is a read only register that provides the status on interrupt operations.

Address: Base address + 10h offset



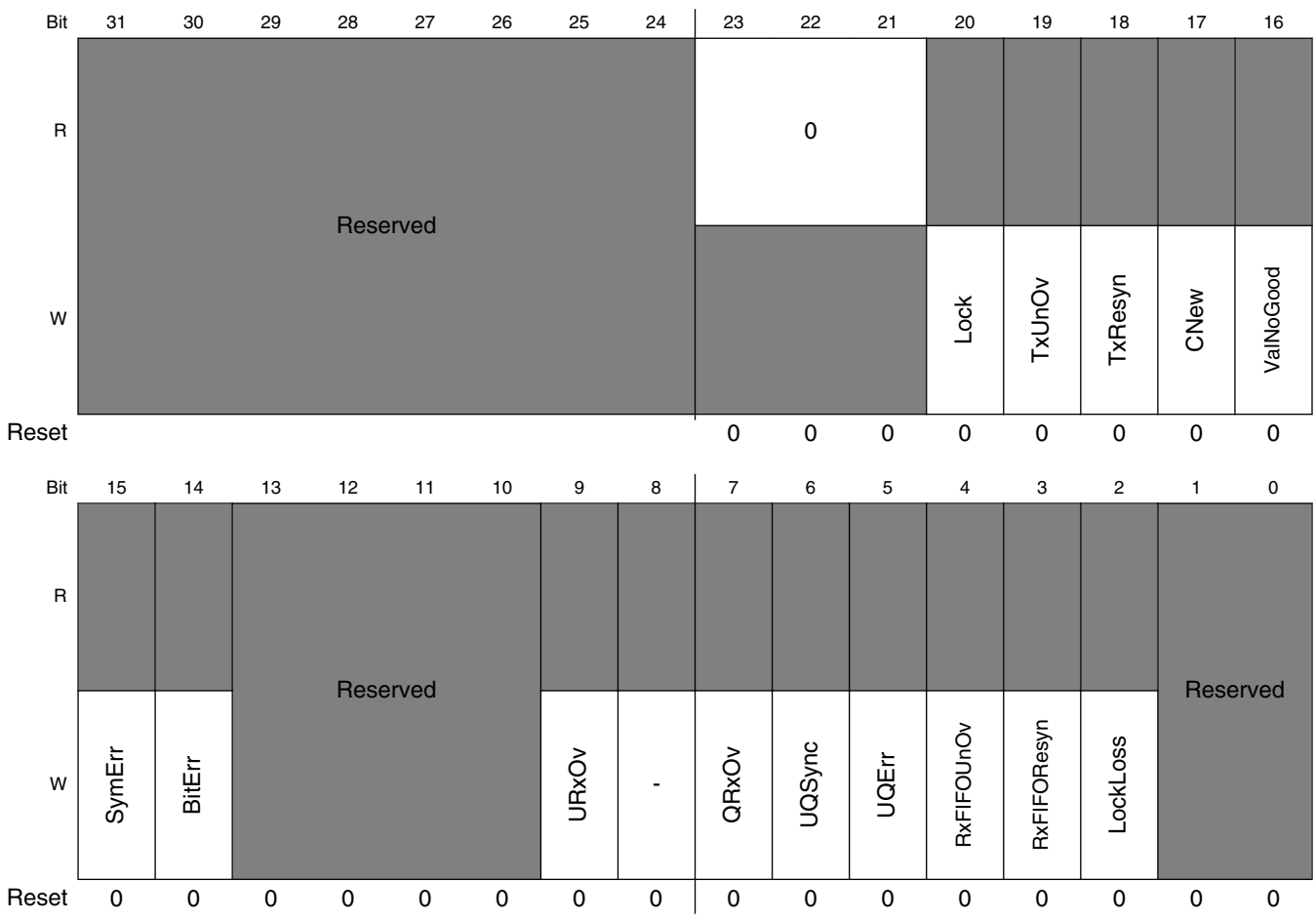
SPDIFx_SIS field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–21 Reserved	This read-only field is reserved and always has the value 0.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–11 Reserved	This read-only field is reserved and always has the value 0.
10 URxFul	U Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from U Rx reg.
9 URxOv	U Channel receive register overrun
8 QRxFul	Q Channel receive register full, can't be cleared with reg. IntClear. To clear it, read from Q Rx reg.
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
1 TxEm	SPDIF Tx FIFO empty, can't be cleared with reg. IntClear. To clear it, write toTx FIFO.
0 RxFIFOFul	SPDIF Rx FIFO full, can't be cleared with reg. IntClear. To clear it, read from Rx FIFO.

13.9.4.6 InterruptClear Register (SPDIFx_SIC)

The InterruptClear (SPDIF_SIC) register is a write only register and is used to clear interrupts.

Address: Base address + 10h offset



SPDIFx_SIC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–21 Reserved	This read-only field is reserved and always has the value 0.
20 Lock	SPDIF receiver's DPLL is locked
19 TxUnOv	SPDIF Tx FIFO under/overrun

Table continues on the next page...

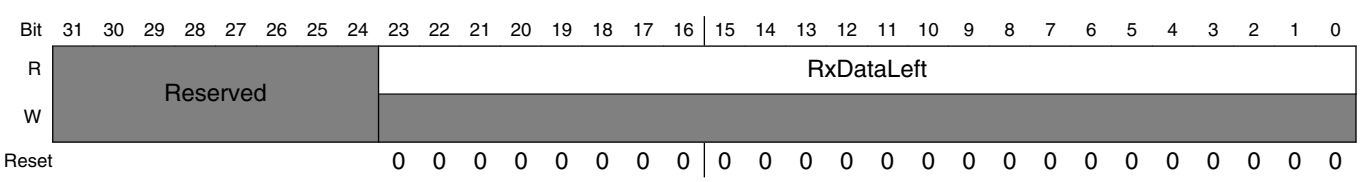
SPDIFx_SIC field descriptions (continued)

Field	Description
18 TxResyn	SPDIF Tx FIFO resync
17 CNew	SPDIF receive change in value of control channel
16 ValNoGood	SPDIF validity flag no good
15 SymErr	SPDIF receiver found illegal symbol
14 BitErr	SPDIF receiver found parity bit error
13–10 -	This field is reserved. Reserved.
9 URxOv	U Channel receive register overrun
8 -	Reserved
7 QRxOv	Q Channel receive register overrun
6 UQSync	U/Q Channel sync found
5 UQErr	U/Q Channel framing error
4 RxFIFOUnOv	Rx FIFO underrun/overrun
3 RxFIFOResyn	Rx FIFO resync
2 LockLoss	SPDIF receiver loss of lock
-	This field is reserved. Reserved.

13.9.4.7 SPDIFRxLeft Register (SPDIFx_SRL)

SPDIFRxLeft register is an audio data reception register.

Address: Base address + 14h offset



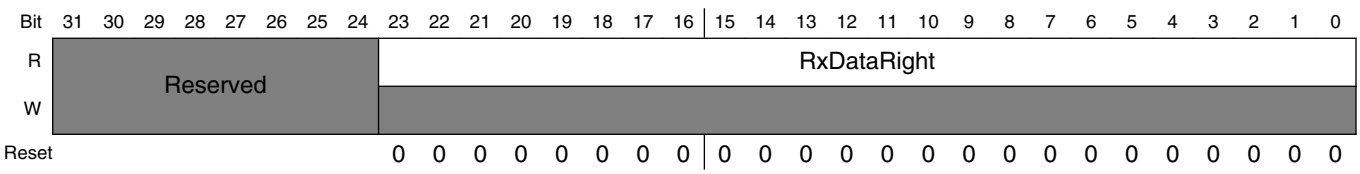
SPDIFx_SRL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxDataLeft	Processor receive SPDIF data left

13.9.4.8 SPDIFRxRight Register (SPDIFx_SRR)

SPDIFRxRight register is an audio data reception register.

Address: Base address + 18h offset



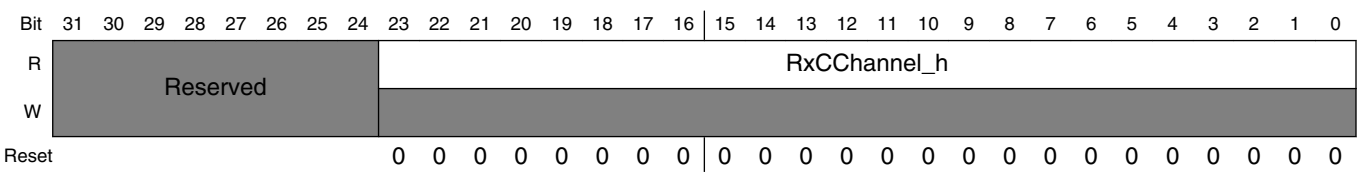
SPDIFx_SRR field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxDataRight	Processor receive SPDIF data right

13.9.4.9 SPDIFRxCChannel_h Register (SPDIFx_SRCSH)

SPDIFRxCChannel_h register is a channel status reception register.

Address: Base address + 1Ch offset



SPDIFx_SRCSH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented.

Table continues on the next page...

SPDIFx_SRCSH field descriptions (continued)

Field	Description
	This field is reserved.
RxCChannel_h	SPDIF receive C channel register, contains first 24 bits of C channel without interpretation

13.9.4.10 SPDIFRxCChannel_l Register (SPDIFx_SRCSL)

SPDIFRxCChannel_l register is a channel status reception register.

Address: Base address + 20h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxCChannel_l																							
W																																
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIFx_SRCSL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxCChannel_l	SPDIF receive C channel register, contains next 24 bits of C channel without interpretation

13.9.4.11 UchannelRx Register (SPDIFx_SRU)

UChannelRx register is a user bits reception register.

Address: Base address + 24h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								RxUChannel																							
W																																
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

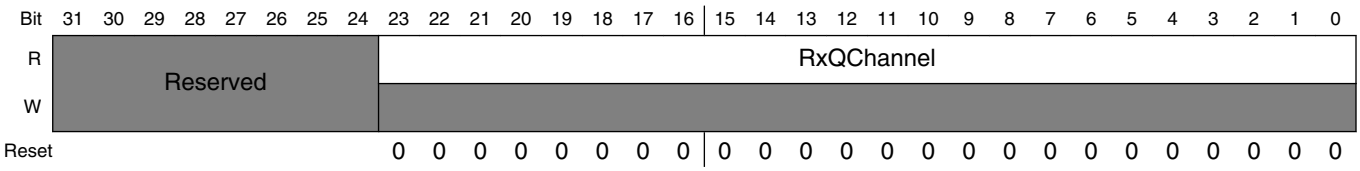
SPDIFx_SRU field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
RxUChannel	SPDIF receive U channel register, contains next 3 U channel bytes

13.9.4.12 QchannelRx Register (SPDIFx_SRQ)

QChannelRx register is a user bits reception register.

Address: Base address + 28h offset



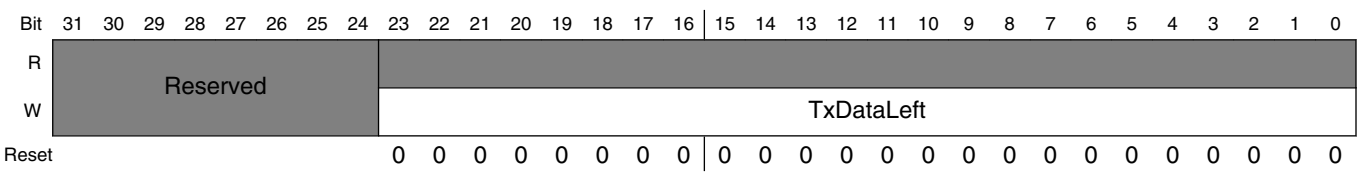
SPDIFx_SRQ field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
RxQChannel	SPDIF receive Q channel register, contains next 3 Q channel bytes

13.9.4.13 SPDIFTxLeft Register (SPDIFx_STL)

SPDIFTxLeft register is an audio data transmission register.

Address: Base address + 2Ch offset



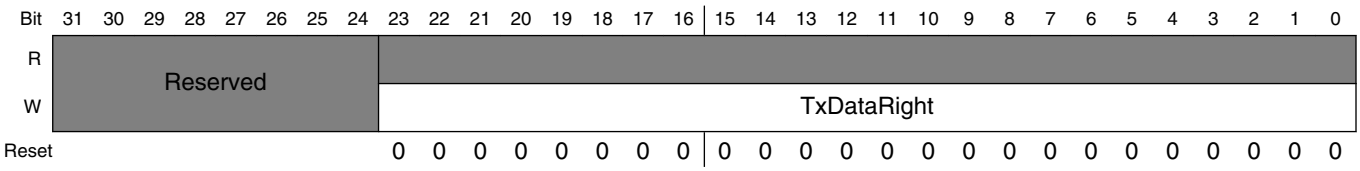
SPDIFx_STL field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
TxDataLeft	SPDIF transmit left channel data. It is write-only, and always returns zeros when read

13.9.4.14 SPDIFTxRight Register (SPDIFx_STR)

SPDIFTxRight register is an audio data transmission register.

Address: Base address + 30h offset



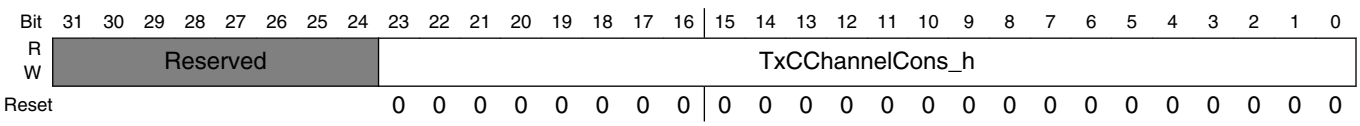
SPDIFx_STR field descriptions

Field	Description
31–24 [unimplemented]	This field is reserved. This is a 24-bit register the upper byte is unimplemented.
TxDataRight	SPDIF transmit right channel data. It is write-only, and always returns zeros when read

13.9.4.15 SPDIFTxCChannelCons_h Register (SPDIFx_STCSCH)

SPDIFTxCChannelCons_h register is a channel status transmission register.

Address: Base address + 34h offset



SPDIFx_STCSCH field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
TxCChannelCons_h	SPDIF transmit Cons. C channel data, contains first 24 bits without interpretation. When read, it returns the latest data written by the processor

13.9.4.16 SPDIFTxCChannelCons_l Register (SPDIFx_STCSCL)

SPDIFTxCChannelCons_l register is a channel status transmission register.

Address: Base address + 38h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								TxCChannelCons_l																							
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

SPDIFx_STCSCL field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
TxCChannelCons_l	SPDIF transmit Cons. C channel data, contains next 24 bits without interpretation. When read, it returns the latest data written by the processor

13.9.4.17 FreqMeas Register (SPDIFx_SRFM)

Address: Base address + 44h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								FreqMeas																							
W																																
Reset									0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

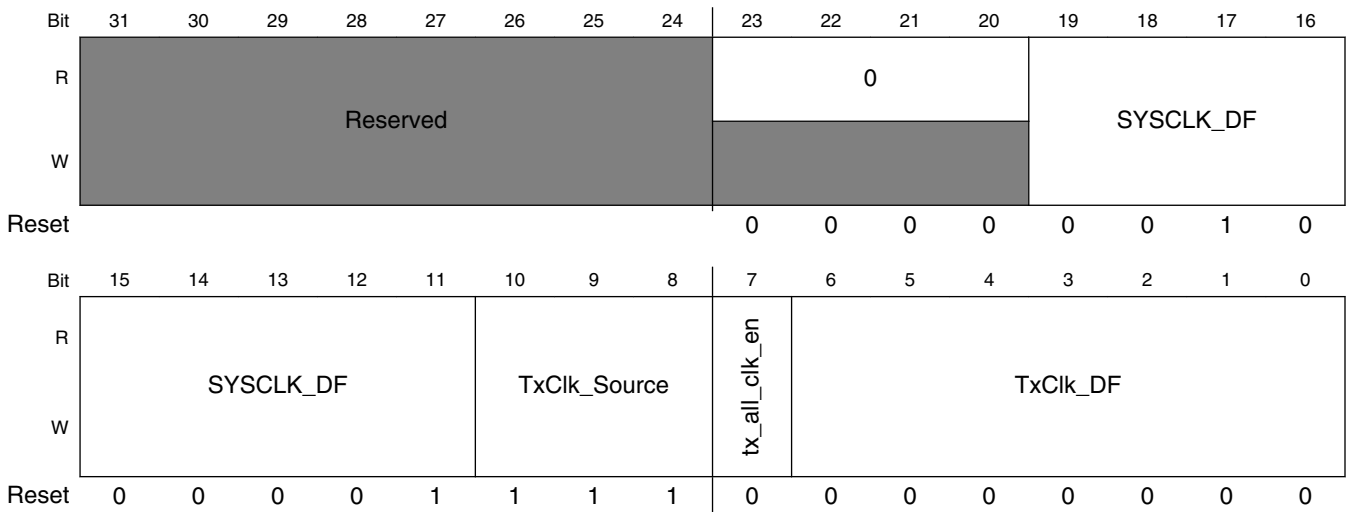
SPDIFx_SRFM field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
FreqMeas	Frequency measurement data

13.9.4.18 SPDIFTxClk Register (SPDIFx_STC)

The SPDIFTxClk Control register includes the means to select the transmit clock and frequency division.

Address: Base address + 50h offset



SPDIFx_STC field descriptions

Field	Description
31–24 [unimplemented]	This is a 24-bit register the upper byte is unimplemented. This field is reserved.
23–20 Reserved	This read-only field is reserved and always has the value 0.
19–11 SYSCLK_DF	system clock divider factor, 2~512. 0 no clock signal 1 divider factor is 2 ... ... 511 divider factor is 512
10–8 TxClk_Source	000 REF_CLK_32K input (XTALOSC 32 kHz clock) 001 tx_clk input (from SPDIF0_CLK_ROOT. See CCM.) 011 SPDIF_EXT_CLK, from pads 101 ipg_clk input (frequency divided)
7 tx_all_clk_en	Spdif transfer clock enable. When data is going to be transfered, this bit should be set to1. 0 disable transfer clock. 1 enable transfer clock.
TxClk_DF	Divider factor (1-128) 0 divider factor is 1

Table continues on the next page...

SPDIFx_STC field descriptions (continued)

Field	Description
	1 divider factor is 2 ... ... 127 divider factor is 128

13.10 Synchronous Audio Interface (SAI)

13.10.1 Introduction

The I^2S (or I2S) module provides a synchronous audio interface (SAI) that supports full-duplex serial interfaces with frame synchronization such as I^2S, AC97, TDM, and codec/DSP interfaces.

13.10.1.1 Features

Note that some of the features are not supported across all SAI instances; see the chip-specific information in the first section of this chapter.

- Transmitter with independent bit clock and frame sync supporting 8 data lines
- Receiver with independent bit clock and frame sync supporting 8 data lines
- Each data line can support a maximum Frame size of 32 words
- Word size of between 8-bits and 32-bits
- Word size configured separately for first word and remaining words in frame
- Asynchronous 128 x 32-bit FIFO for each transmit and receive data line
- Supports graceful restart after FIFO error
- Supports automatic restart after FIFO error without software intervention
- Supports packing of 8-bit and 16-bit data into each 32-bit FIFO word
- Supports combining multiple data line FIFOs into single data line FIFO

13.10.1.2 Block diagram

The following block diagram also shows the module clocks.

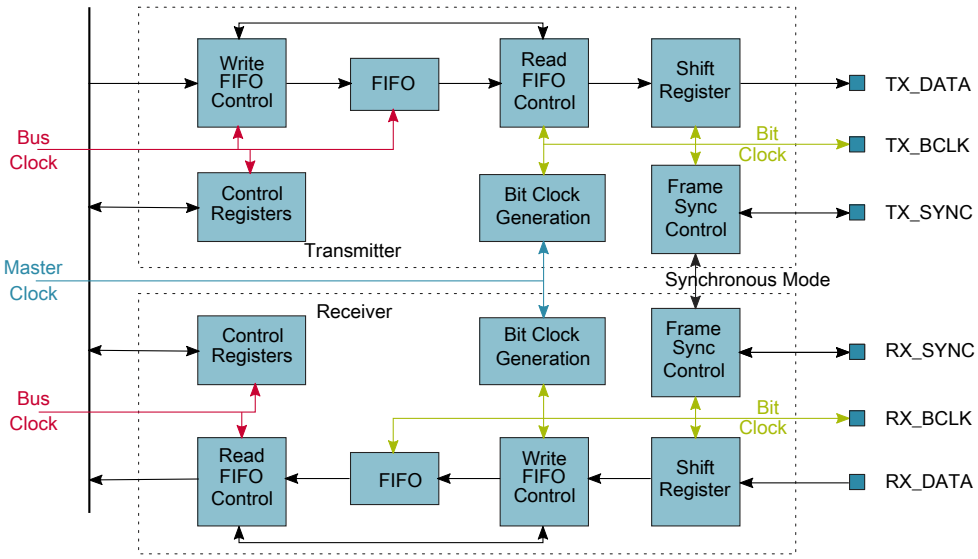


Figure 13-83. I²S/SAI block diagram

13.10.1.3 Modes of operation

Module power modes include Run mode, and Debug mode.

13.10.1.3.1 Run mode

In Run mode, the SAI transmitter and receiver operate normally.

13.10.1.3.2 Debug mode

In Debug mode, the SAI transmitter and/or receiver can continue operating provided the Debug Enable bit is set. When TCSR[DBGE] or RCSR[DBGE] bit is clear and Debug mode is entered, the SAI is disabled after completing the current transmit or receive frame. The transmitter and receiver bit clocks are not affected by Debug mode.

13.10.2 External signals

Name	Function	I/O
TX_BCLK	Transmit Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O

Table continues on the next page...

Name	Function	I/O
TX_SYNC	Transmit Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
TX_DATA[7:0]	Transmit Data. The transmit data is generated synchronously by the bit clock and is tristated whenever not transmitting a word.	O
RX_BCLK	Receive Bit Clock. The bit clock is an input when externally generated and an output when internally generated.	I/O
RX_SYNC	Receive Frame Sync. The frame sync is an input sampled synchronously by the bit clock when externally generated and an output generated synchronously by the bit clock when internally generated.	I/O
RX_DATA[7:0]	Receive Data. The receive data is sampled synchronously by the bit clock.	I

13.10.3 Functional description

This section provides a complete functional description of the block.

13.10.3.1 SAI clocking

The SAI clocks include:

- The audio master clock
- The bit clock
- The bus clock

13.10.3.1.1 Audio master clock

The audio master clock is used to generate the bit clock when the receiver or transmitter is configured for an internally generated bit clock. The transmitter and receiver can independently select between the bus clock and up to three audio master clocks to generate the bit clock.

The audio master clock generation and selection is chip-specific. Refer to chip-specific clocking information about how the audio master clocks are generated.

13.10.3.1.2 Bit clock

The SAI transmitter and receiver support asynchronous free-running bit clocks that can be generated internally from an audio master clock or supplied externally. There is also the option for synchronous bit clock and frame sync operation between the receiver and transmitter or between multiple SAI peripherals.

- If both transmitter and receiver are configured for asynchronous operation, then the transmitter and receiver will each use *their own* bit clock and frame sync.
- If the *transmitter* is configured for asynchronous mode and the receiver is configured for synchronous mode, then both transmitter and receiver will use the *transmitter* bit clock and frame sync.
- If the *receiver* is configured for asynchronous mode and the transmitter is configured for synchronous mode, then both transmitter and receiver will use the *receiver* bit clock and frame sync.

Note that the software configures synchronous or asynchronous mode, and that choice selects the bit clock/frame sync used.

Externally generated bit clocks must be:

- Enabled before the SAI transmitter or receiver is enabled
- Disabled after the SAI transmitter or receiver is disabled and completes its current frames

If the SAI transmitter or receiver is using an externally generated bit clock in asynchronous mode and that bit clock is generated by an SAI that is disabled in stop mode, then the transmitter or receiver should be disabled by software before entering stop mode. This issue does not apply when the transmitter or receiver is in a synchronous mode because all synchronous SAIs are enabled and disabled simultaneously.

13.10.3.1.3 Bus clock

The bus clock is used by the control and configuration registers and to generate synchronous interrupts and DMA requests.

NOTE

Although there is no specific minimum bus clock frequency specified, the bus clock frequency must be fast enough (relative to the bit clock frequency) to ensure that the FIFOs can be serviced, without generating either a transmitter FIFO underrun or receiver FIFO overflow condition.

13.10.3.2 SAI resets

The SAI is asynchronously reset on system reset. The SAI has a software reset and a FIFO reset.

13.10.3.2.1 Software reset

The SAI transmitter includes a software reset that resets all transmitter internal logic, including the bit clock generation, status flags, and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

The SAI receiver includes a software reset that resets all receiver internal logic, including the bit clock generation, status flags and FIFO pointers. It does not reset the configuration registers. The software reset remains asserted until cleared by software.

13.10.3.2.2 FIFO reset

The SAI transmitter includes a FIFO reset that synchronizes the FIFO write pointer to the same value as the FIFO read pointer. This empties the FIFO contents and is to be used after TCSR[FEF] is set, and before the FIFO is re-initialized and TCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI transmitter can also reset the FIFO of individual data channels by setting the appropriate TCR3[CFR] bit. This should only be done when the corresponding TCR3[TCE] bit is clear.

The SAI receiver includes a FIFO reset that synchronizes the FIFO read pointer to the same value as the FIFO write pointer. This empties the FIFO contents and is to be used after the RCSR[FEF] is set and any remaining data has been read from the FIFO, and before the RCSR[FEF] is cleared. The FIFO reset is asserted for one cycle only.

The SAI receiver can also reset the FIFO of individual data channels by setting the appropriate RCR3[CFR] bit. This should only be done when the corresponding RCR3[RCE] bit is clear.

13.10.3.3 Synchronous modes

The SAI transmitter and receiver can operate synchronously to each other.

13.10.3.3.1 Synchronous mode

The SAI transmitter and receiver can be configured to operate with synchronous bit clock and frame sync.

If the transmitter bit clock and frame sync are to be used by both the transmitter and receiver:

- The transmitter must be configured for asynchronous operation and the receiver for synchronous operation.
- In synchronous mode, the receiver is enabled only when both the transmitter and receiver are enabled.
- It is recommended that the transmitter is the last enabled and the first disabled.

If the receiver bit clock and frame sync are to be used by both the transmitter and receiver:

- The receiver must be configured for asynchronous operation and the transmitter for synchronous operation.
- In synchronous mode, the transmitter is enabled only when both the receiver and transmitter are both enabled.
- It is recommended that the receiver is the last enabled and the first disabled.

When operating in synchronous mode, only the bit clock, frame sync, and transmitter/receiver enable are shared. The transmitter and receiver otherwise operate independently, although configuration registers must be configured consistently across both the transmitter and receiver.

13.10.3.4 Frame sync configuration

When enabled, the SAI continuously transmits and/or receives frames of data. Each frame consists of a fixed number of words and each word consists of a fixed number of bits. Within each frame, any given word can be masked causing the receiver to ignore that word and the transmitter to tri-state for the duration of that word.

The frame sync signal is used to indicate the start of each frame. A valid frame sync requires a rising edge (if active high) or falling edge (if active low) to be detected and the transmitter or receiver cannot be busy with a previous frame. A valid frame sync is also ignored (slave mode) or not generated (master mode) for the first four bit clock cycles after enabling the transmitter or receiver.

The transmitter and receiver frame sync can be configured independently with any of the following options:

- Externally generated or internally generated
- Active high or active low
- Assert with the first bit in frame or asserts one bit early
- Assert for a duration between 1 bit clock and the first word length
- Frame length from 1 to 32 words per frame

- Word length to support 8 to 32 bits per word
  - First word length and remaining word lengths can be configured separately
- Words can be configured to transmit/receive MSB first or LSB first

These configuration options cannot be changed after the SAI transmitter or receiver is enabled.

13.10.3.5 Data FIFO

Each transmit and receive channel includes a FIFO of size 128 x 32-bit. The FIFO data is accessed using the SAI Transmit/Receive Data Registers.

13.10.3.5.1 Data alignment

Data in the FIFO can be aligned anywhere within the 32-bit wide register through the use of the First Bit Shifted configuration field, which selects the bit index (between 31 and 0) of the first bit shifted.

Examples of supported data alignment and the required First Bit Shifted configuration are illustrated in Figure 13-84 for LSB First configurations and Figure 13-85 for MSB First configurations.

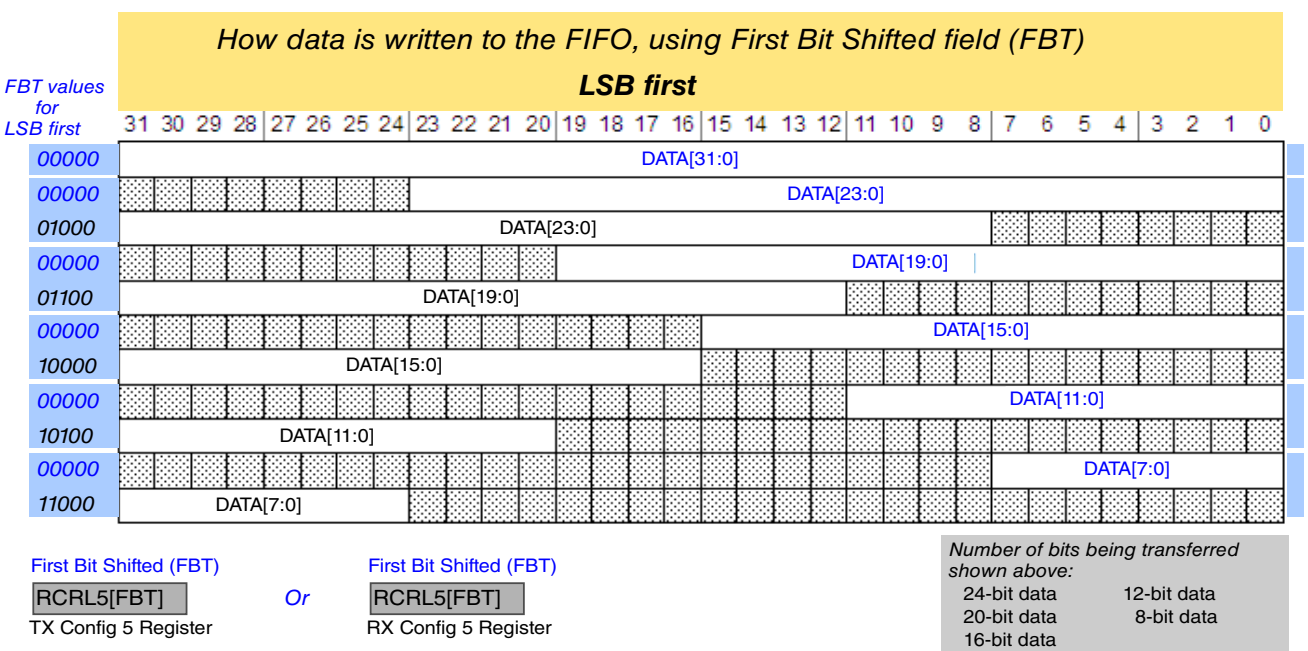


Figure 13-84. SAI first bit shifted, LSB first

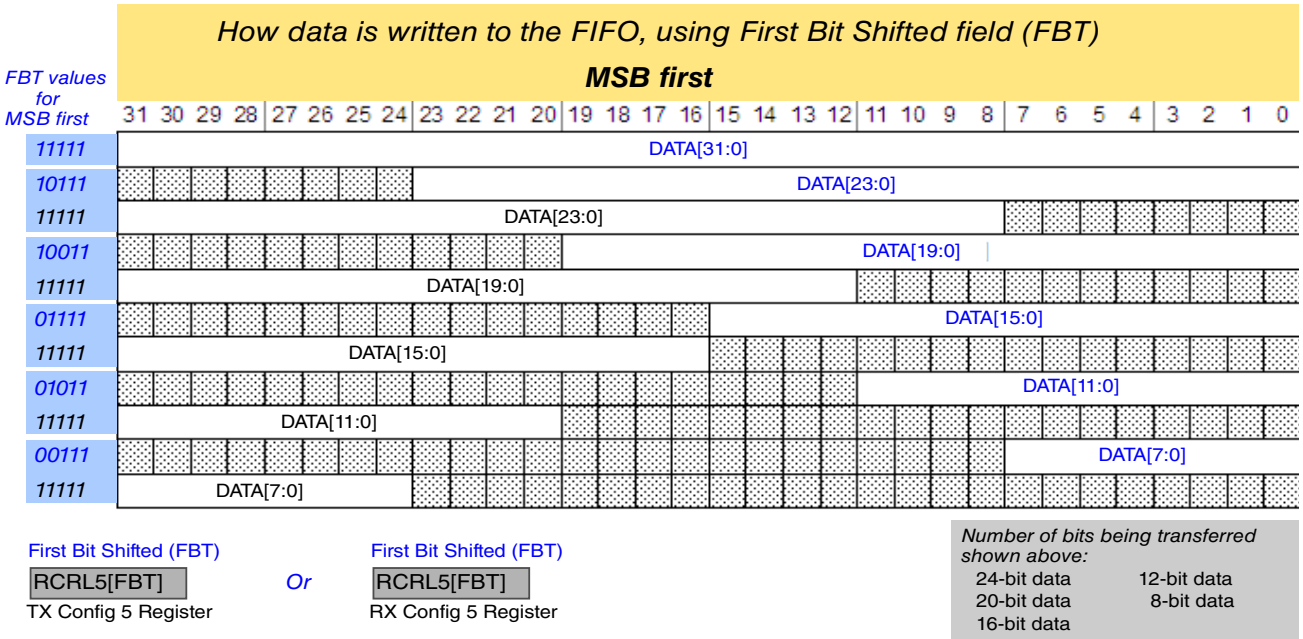


Figure 13-85. SAI first bit shifted, MSB first

13.10.3.5.2 FIFO pointers

When writing to a Transmit Data Register (TDR*n*), the Write FIFO Pointer (WFP) of the corresponding Transmit FIFO Register (TFR*n*) increments after each valid write. The SAI supports 8-bit, 16-bit and 32-bit writes to the Transmit Data Register and the FIFO pointer will increment after each individual write. Note that 8-bit writes should only be used when transmitting up to 8-bit data; 16-bit writes should only be used when transmitting up to 16-bit data.

- If the Transmit FIFO is full, then writes to a Transmit Data Register are ignored.
- If the Transmit FIFO is empty, then to avoid a FIFO underrun, the Transmit Data Register must be written at least 3 bit clocks before the start of the next unmasked word. Before enabling the transmitter, the Transmit FIFO should be initialized with data (since after the transmitter is enabled, the transmitter will start a new frame, and if no data is in the FIFO, then the transmitter will immediately give an error).

When reading a Receive Data Register (RDR*n*), the Read FIFO Pointer (RFP) of the corresponding Receive FIFO Register (RFR*n*) increments after each valid read. The SAI supports 8-bit, 16-bit and 32-bit reads from the RDR and the FIFO pointer will increment after each individual read. Note that 8-bit reads should only be used when receiving up to 8-bit data; 16-bit reads should only be used when receiving up to 16-bit data.

- If the Receive FIFO is empty, then reads from a Receive Data Register are ignored.
- If the Receive FIFO is full, then to avoid a FIFO overrun, the Receive Data Register must be read at least 3 bit clocks before the end of an unmasked word.

13.10.3.5.3 FIFO packing

FIFO packing supports storing multiple 8-bit or 16-bit data words in one 32-bit FIFO word for the transmitter and/or receiver. While this can be emulated by adjusting the number of bits per word and number of words per frame (for example, one 32-bit word per frame versus two 16-bit words per frame), FIFO packing does not require even multiples of words per frame and fully supports word masking. When FIFO packing is enabled, the FIFO pointers only increment when the full 32-bit FIFO word has been written (transmit) or read (receive) by software, supporting scenarios where different words within each frame are loaded/stored in different areas of memory.

When 16-bit FIFO packing is enabled for transmit, the transmit shift register is loaded at the start of each frame and after every second unmasked transmit word. The first word transmitted is taken from 16-bit word at byte offset $0 (first bit is selected by TCFG5[FBT] must be configured within this 16-bit word) and the second word transmitted is taken from the 16-bit word at byte offset $2 (first bit is selected by TCSR5[FBT][3:0]). The transmitter will transmit logic zero until the start of the next word once the 16-bit word has been transmitted.

When 16-bit FIFO packing is enabled for receive, the receive shift register is stored after every second unmasked received word, and at the end of each frame if there is an odd number of unmasked received words in each frame. The first word received is stored in the 16-bit word at byte offset $0 (first bit is selected by RCFG5[FBT] and must be configured within this 16-bit word) and the second word received is stored in the 16-bit word at byte offset $2 (first bit is selected by RCSR5[FBT][3:0]). The receiver will ignore received data until the start of the next word once the 16-bit word has been received.

The 8-bit FIFO packing is similar to 16-bit packing except four words are loaded or stored into each 32-bit FIFO word. The first word is loaded/stored in byte offset $0, second word in byte offset $1, third word in byte offset $2 and fourth word in byte offset $3. The TCFG5[FBT] and/or RCFG5[FBT] must be configured within byte offset $0.

13.10.3.5.4 FIFO Combine

FIFO combining mode allows the separate FIFOs for multiple data channels to be used as a single FIFO for either software accesses or a single data channel or both. Note that the enabled data channels must be contiguous and data channel 0 must be enabled when FIFO Combine mode is enabled.

Combining FIFOs for software access (writing transmit FIFO registers, reading receive FIFO registers) allows a DMA controller or software to read or write multiple FIFOs without incrementing the address that is accessed. Once enabled, the first software access

to a FIFO register will access the first enabled channel FIFO, while the second access to a FIFO register will access the second enabled channel FIFO. This continues until software accesses the last enabled channel FIFO and the pointer resets back to the first enabled channel FIFO. To reset the pointer manually, software can reset the FIFOs or disable the FIFO combining on software accesses.

Combining FIFOs for transmit data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with identical data output on each enabled data channel. The transmit shift registers for all enabled data channels are loaded at the start of each frame and every N unmasked words (where N is the number of enabled data channels). The first word transmitted is loaded from the first enabled channel FIFO, while the second word transmitted is loaded from the second enabled channel FIFO, and so on until the end of the frame. Since the first word in each frame is always loaded from the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Combining FIFOs for receive data channels allows one data channel to use the FIFOs of all enabled channel FIFOs, with received data from channel 0 stored into each enabled data channel. The receive shift register for all enabled data channels are stored after every N unmasked words (where N is the number of enabled data channels). The first word received is stored to the first enabled channel FIFO, while the second word received is stored to the second enabled channel FIFO, and so on until the end of the frame. Since the first word in each frame is always stored the first enabled data channel, it is recommended that the number of unmasked words in each frame is evenly divisible by the number of enabled data channels.

Note that combining FIFOs for data channels will load or store each channel FIFO at the same time. This means that FIFO error conditions are only checked every N words (where N is the number of enabled data channels) and that the FIFO warning and request flags will assert if any of the enabled data channel meets the warning flag or request flag conditions.

13.10.3.6 Word mask register

The SAI transmitter and receiver each contain a word mask register, namely TMR and RMR, that can be used to mask any word in the frame. Because the word mask register is double buffered, software can update it before the end of each frame to mask a particular word in the next frame.

The TMR causes the Transmit Data pin to be tri-stated for the length of each selected word and the transmit FIFO is not read for masked words.

The RMR causes the received data for each selected word to be discarded and not written to the receive FIFO.

13.10.3.7 Interrupts and DMA requests

The SAI transmitter and receiver generate separate interrupts and separate DMA requests, but support the same status flags.

13.10.3.7.1 FIFO request flag

The FIFO request flag is set based on the number of entries in the FIFO and the FIFO watermark configuration.

The transmit FIFO request flag is set when the number of entries in any of the enabled transmit FIFOs is less than or equal to the transmit FIFO watermark configuration and is cleared when the number of entries in each enabled transmit FIFO is greater than the transmit FIFO watermark configuration.

The receive FIFO request flag is set when the number of entries in any of the enabled receive FIFOs is greater than the receive FIFO watermark configuration and is cleared when the number of entries in each enabled receive FIFO is less than or equal to the receive FIFO watermark configuration.

The FIFO request flag can generate an interrupt or a DMA request.

13.10.3.7.2 FIFO warning flag

The FIFO warning flag is set based on the number of entries in the FIFO.

The transmit warning flag is set when the number of entries in any of the enabled transmit FIFOs is empty and is cleared when the number of entries in each enabled transmit FIFO is not empty.

The receive warning flag is set when the number of entries in any of the enabled receive FIFOs is full and is cleared when the number of entries in each enabled receive FIFO is not full.

The FIFO warning flag can generate an Interrupt or a DMA request.

13.10.3.7.3 FIFO error flag

The transmit FIFO error flag is set when the any of the enabled transmit FIFOs underflow. After it is set, all enabled transmit channels will transmit zero data before TCSR[FEF] is cleared.

When TCR4[FCONT] is set, the FIFO will continue transmitting data following an underflow without software intervention. To ensure that data is transmitted in the correct order, the transmitter will continue from the same word number in the frame that caused the FIFO to underflow, but only after new data has been written to the transmit FIFO. Software should still clear the TCSR[FEF] flag, but without reinitializing the transmit FIFOs.

RCSR[FEF] is set when the any of the enabled receive FIFOs overflow. After it is set, all enabled receive channels discard received data until RCSR[FEF] is cleared and the next next receive frame starts. All enabled receive FIFOs should be emptied before RCSR[FEF] is cleared.

When RCR4[FCONT] is set, the FIFO will continue receiving data following an overflow without software intervention. To ensure that data is received in the correct order, the receiver will continue from the same word number in the frame that caused the FIFO to overflow, but only after data has been read from the receive FIFO. Software should still clear the RCSR[FEF] flag, but without emptying the receive FIFOs.

The FIFO error flag can generate only an interrupt.

13.10.3.7.4 Sync error flag

The sync error flag, TCSR[SEF] or RCSR[SEF], is set when configured for an externally generated frame sync and the external frame sync asserts when the transmitter or receiver is busy with the previous frame. The external frame sync assertion is ignored and the sync error flag is set. When the sync error flag is set, the transmitter or receiver continues checking for frame sync assertion when idle or at the end of each frame.

The sync error flag can generate an interrupt only.

13.10.3.7.5 Word start flag

The word start flag is set at the start of the second bit clock for the selected word, as configured by the Word Flag register field.

The word start flag can generate an interrupt only.

13.10.4 Memory map and register definition

A read or write access to an address from offset 0x100 and above will result in a bus error.

13.10.4.1 I2S register descriptions

13.10.4.1.1 I2S Memory map

SAI1 base address: 3001_0000h

SAI2 base address: 308B_0000h

SAI3 base address: 308C_0000h

SAI4 base address: 3005_0000h

SAI5 base address: 3004_0000h

SAI6 base address: 3003_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Version ID Register (VERID)	32	RO	0300_0000h
4h	Parameter Register (PARAM)	32	RO	Table 13-227
8h	SAI Transmit Control Register (TCSR)	32	RW	0000_0000h
Ch	SAI Transmit Configuration 1 Register (TCR1)	32	RW	0000_0000h
10h	SAI Transmit Configuration 2 Register (TCR2)	32	RW	0000_0000h
14h	SAI Transmit Configuration 3 Register (TCR3)	32	RW	0000_0000h
18h	SAI Transmit Configuration 4 Register (TCR4)	32	RW	0000_0000h
1Ch	SAI Transmit Configuration 5 Register (TCR5)	32	RW	0000_0000h
20h - 3Ch	SAI Transmit Data Register (TDR0 - TDR7)	32	WORZ	See description
40h - 5Ch	SAI Transmit FIFO Register (TFR0 - TFR7)	32	RO	See description
60h	SAI Transmit Mask Register (TMR)	32	RW	0000_0000h
88h	SAI Receive Control Register (RCSR)	32	RW	0000_0000h
8Ch	SAI Receive Configuration 1 Register (RCR1)	32	RW	0000_0000h
90h	SAI Receive Configuration 2 Register (RCR2)	32	RW	0000_0000h
94h	SAI Receive Configuration 3 Register (RCR3)	32	RW	0000_0000h
98h	SAI Receive Configuration 4 Register (RCR4)	32	RW	0000_0000h
9Ch	SAI Receive Configuration 5 Register (RCR5)	32	RW	0000_0000h
A0h - BCh	SAI Receive Data Register (RDR0 - RDR7)	32	RO	See description
C0h - DCh	SAI Receive FIFO Register (RFR0 - RFR7)	32	RO	See description
E0h	SAI Receive Mask Register (RMR)	32	RW	0000_0000h

13.10.4.1.2 Version ID Register (VERID)

13.10.4.1.2.1 Offset

Register	Offset
VERID	0h

13.10.4.1.2.2 Function

Contains version numbers for the module design and feature set.

13.10.4.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	MAJOR								MINOR							
W																
Reset	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	FEATURE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.10.4.1.2.4 Fields

Field	Function
31-24 MAJOR	Major Version Number This read only field returns the major version number for the specification.
23-16 MINOR	Minor Version Number This read only field returns the minor version number for the specification.
15-0 FEATURE	Feature Specification Number This read only field returns the feature set number. 0000000000000000b - Standard feature set.

13.10.4.1.3 Parameter Register (PARAM)

13.10.4.1.3.1 Offset

Register	Offset
PARAM	4h

13.10.4.1.3.2 Function

Contains parameter values that were implemented in the module.

13.10.4.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0												FRAME			
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0				FIFO				0				DATALINE			
W																
Reset	See Register reset values.															

13.10.4.1.3.4 Register reset values

Register	Reset value
PARAM	: 0005_0708h –: 0005_0701h

13.10.4.1.3.5 Fields

Field	Function
31-20 —	Reserved
19-16 FRAME	Frame Size The maximum number of slots per frame is 2^FRAME.
15-12 —	Reserved
11-8 FIFO	FIFO Size The number of words in each FIFO is 2^FIFO.

Table continues on the next page...

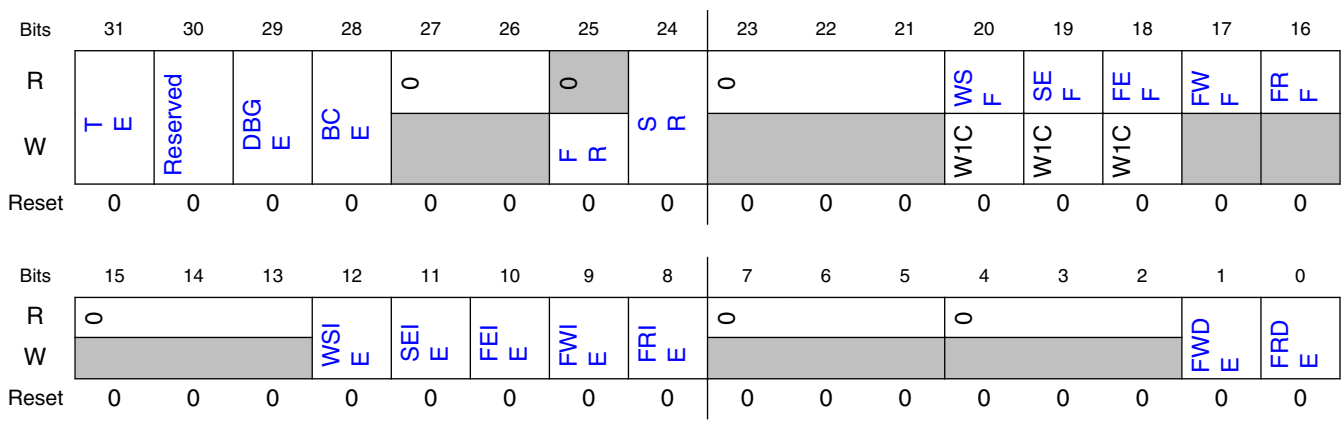
Field	Function
7-4 —	Reserved
3-0 DATALINE	Number of Datalines The number of datalines implemented.

13.10.4.1.4 SAI Transmit Control Register (TCSR)

13.10.4.1.4.1 Offset

Register	Offset
TCSR	8h

13.10.4.1.4.2 Diagram



13.10.4.1.4.3 Fields

Field	Function
31 TE	Transmitter Enable Enables/disables the transmitter. When software clears this field, the transmitter remains enabled, and this bit remains set, until the end of the current frame. 0b - Transmitter is disabled. 1b - Transmitter is enabled, or transmitter has been disabled and has not yet reached end of frame.
30 —	Reserved. Software should only write zero to this reserved bit.
29 DBGE	Debug Enable

Table continues on the next page...

Field	Function
	Enables/disables transmitter operation in Debug mode. The transmit bit clock is not affected by debug mode. 0b - Transmitter is disabled in Debug mode, after completing the current frame. 1b - Transmitter is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the transmit bit clock, separately from the TE. This field is automatically set whenever TE is set. When software clears this field, the transmit bit clock remains enabled, and this bit remains set, until the end of the current frame. 0b - Transmit bit clock is disabled. 1b - Transmit bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the transmitter is disabled or the FIFO error flag is set. 0b - No effect. 1b - FIFO reset.
24 SR	Software Reset When set, resets the internal transmitter logic including the FIFO read and write pointers. Software-visible registers are not affected, except for the status registers. 0b - No effect. 1b - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0b - Start of word not detected. 1b - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0b - Sync error not detected. 1b - Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled transmit FIFO has underrun. Write a logic 1 to this field to clear this flag. 0b - Transmit underrun not detected. 1b - Transmit underrun detected.
17 FWF	FIFO Warning Flag Indicates that an enabled transmit FIFO is empty. 0b - No enabled transmit FIFO is empty. 1b - Enabled transmit FIFO is empty.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled transmit channel FIFO is less than or equal to the transmit FIFO watermark. 0b - Transmit FIFO watermark has not been reached. 1b - Transmit FIFO watermark has been reached.

Table continues on the next page...

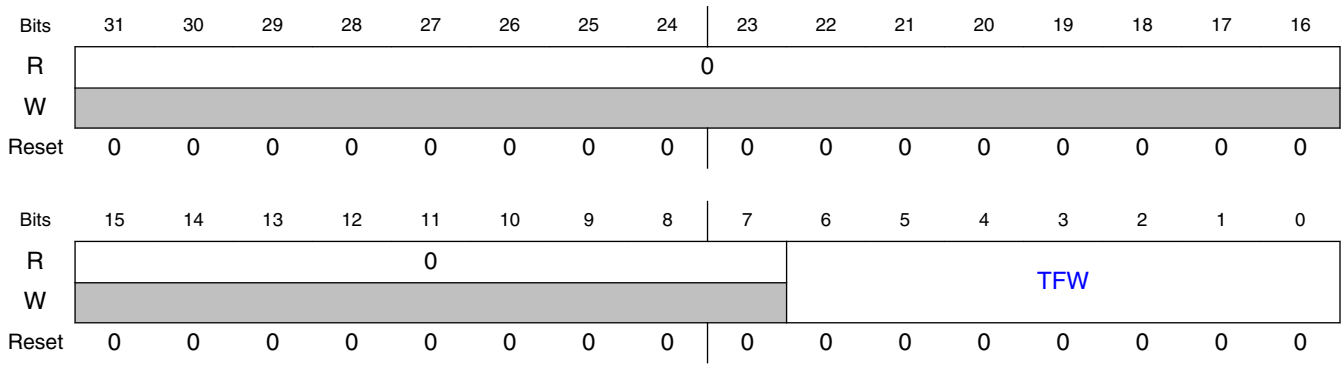
Field	Function
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0b - Disables interrupt. 1b - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0b - Disables interrupt. 1b - Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0b - Disables the interrupt. 1b - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0b - Disables the interrupt. 1b - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts. 0b - Disables the interrupt. 1b - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0b - Disables the DMA request. 1b - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0b - Disables the DMA request. 1b - Enables the DMA request.

13.10.4.1.5 SAI Transmit Configuration 1 Register (TCR1)

13.10.4.1.5.1 Offset

Register	Offset
TCR1	Ch

13.10.4.1.5.2 Diagram



13.10.4.1.5.3 Fields

Field	Function
31-7 —	Reserved
6-0 TFW	Transmit FIFO Watermark Configures the watermark level for all enabled transmit channels.

13.10.4.1.6 SAI Transmit Configuration 2 Register (TCR2)

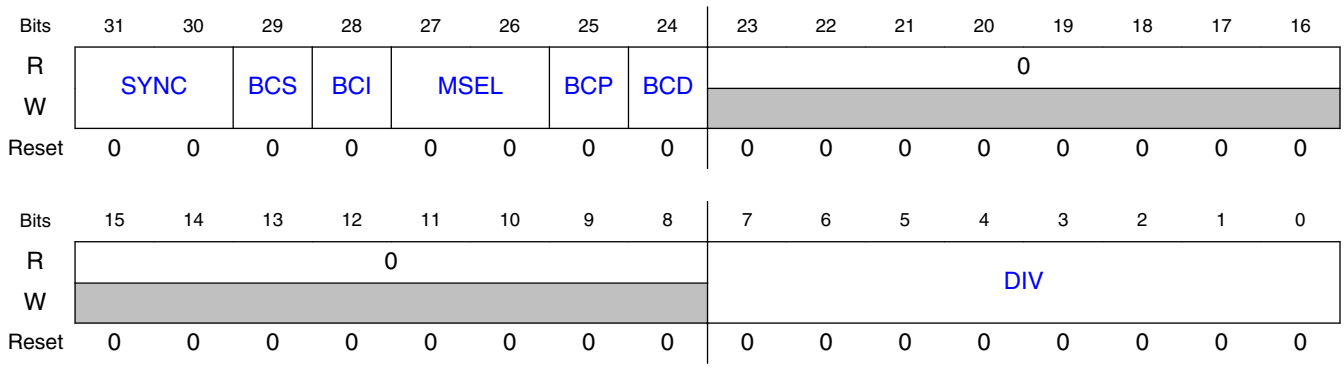
13.10.4.1.6.1 Offset

Register	Offset
TCR2	10h

13.10.4.1.6.2 Function

This register must not be altered when TCSR[TE] is set.

13.10.4.1.6.3 Diagram



13.10.4.1.6.4 Fields

Field	Function
31-30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the receiver must be configured for asynchronous operation. 00b - Asynchronous mode. 01b - Synchronous with receiver. 10b - Reserved. 11b - Reserved.
29 BCS	Bit Clock Swap This field swaps the bit clock used by the transmitter. When the transmitter is configured in asynchronous mode and this bit is set, the transmitter is clocked by the receiver bit clock (RX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the transmitter continues to use the transmit frame sync (TX_SYNC). When the transmitter is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the transmitter bit clock (TX_BCLK) but use the receiver frame sync (RX_SYNC). 0b - Use the normal bit clock source. 1b - Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the transmitter is delayed by the pad output delay (the transmitter is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the transmitter when this bit is set. In synchronous mode, this bit allows the transmitter to use the slave mode timing from the datasheet, while the receiver uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0b - No effect. 1b - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock.

Table continues on the next page...

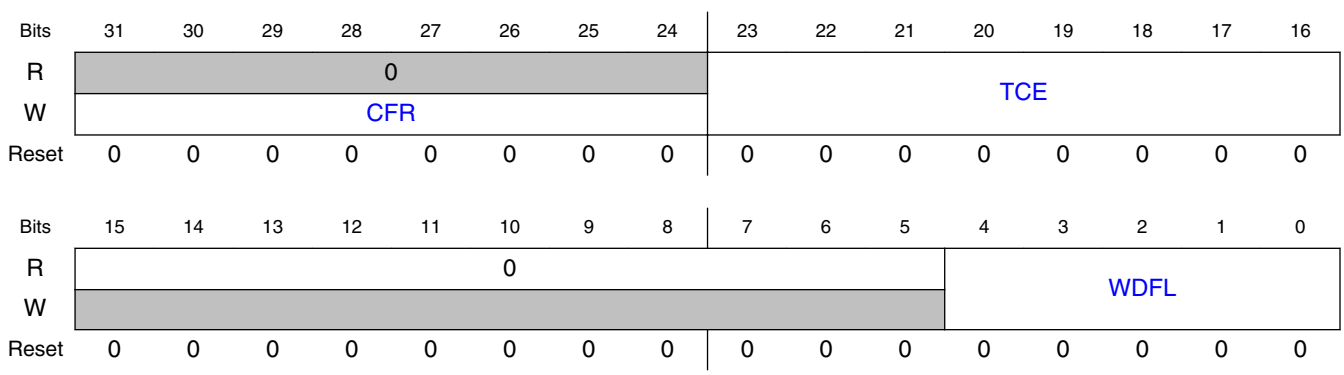
Field	Function
	NOTE: Depending on the device, some Master Clock options might not be available. See the chip-specific information for the meaning of each option. 00b - Bus Clock selected. 01b - Master Clock (MCLK) 1 option selected. 10b - Master Clock (MCLK) 2 option selected. 11b - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0b - Bit clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1b - Bit clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0b - Bit clock is generated externally in Slave mode. 1b - Bit clock is generated internally in Master mode.
23-8 —	Reserved
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

13.10.4.1.7 SAI Transmit Configuration 3 Register (TCR3)

13.10.4.1.7.1 Offset

Register	Offset
TCR3	14h

13.10.4.1.7.2 Diagram



13.10.4.1.7.3 Fields

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>31-24
CFR</td><td>Channel FIFO Reset

Resets the FIFO pointers for a specific channel. Reading this field will always return zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set.

The width of CFR field = the number of transmit channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to transmit channel 1 FIFO pointer and bit position 25 refers to transmit channel 2 FIFO pointer. Setting bit 24 resets transmit channel 1 FIFO pointer, and setting bit 25 enables transmit channel 2 FIFO pointer. Setting bit N will reset transmit channel N FIFO pointer.

0b - No effect.
1b - Transmit data channel N FIFO is reset.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1_TCR3</td><td>—</td></tr>
<tr><td>—</td><td>SAI2_TCR3</td></tr>
<tr><td>—</td><td>SAI3_TCR3</td></tr>
<tr><td>—</td><td>SAI4_TCR3</td></tr>
<tr><td>—</td><td>SAI5_TCR3</td></tr>
<tr><td>—</td><td>SAI6_TCR3</td></tr>
</table>
</td></tr>
<tr><td>23-16
TCE</td><td>Transmit Channel Enable

Enables the corresponding data channel for transmit operation. Changing TCE field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for transmit operation.

The width of TCE field = the number of transmit channels (call it N). For example, if TCE field is 2 bits wide, then bit position 16 refers to transmit channel 1 and bit position 17 refers to transmit channel 2. Setting bit 16 enables transmit channel 1, and setting bit 17 enables transmit channel 2. Setting bit N will enable transmit channel N.

0b - Transmit data channel N is disabled.
1b - Transmit data channel N is enabled.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1_TCR3</td><td>—</td></tr>
<tr><td>SAI2_TCR3[16]</td><td>SAI2_TCR3[23–17]</td></tr>
<tr><td>SAI3_TCR3[16]</td><td>SAI3_TCR3[23–17]</td></tr>
<tr><td>SAI4_TCR3[16]</td><td>SAI4_TCR3[23–17]</td></tr>
<tr><td>SAI5_TCR3[16]</td><td>SAI5_TCR3[23–17]</td></tr>
<tr><td>SAI6_TCR3[16]</td><td>SAI6_TCR3[23–17]</td></tr>
</table>
</td></tr>
<tr><td>15-5</td><td>Reserved</td></tr>
</table>

Table continues on the next page...

Field	Function
—	
4-0 WDFL	Word Flag Configuration Configures which word sets the start of word flag. The value written must be one less than the word number. For example, writing 0 configures the first word in the frame. When configured to a value greater than TCR4[FRSZ], then the start of word flag is never set.

13.10.4.1.8 SAI Transmit Configuration 4 Register (TCR4)

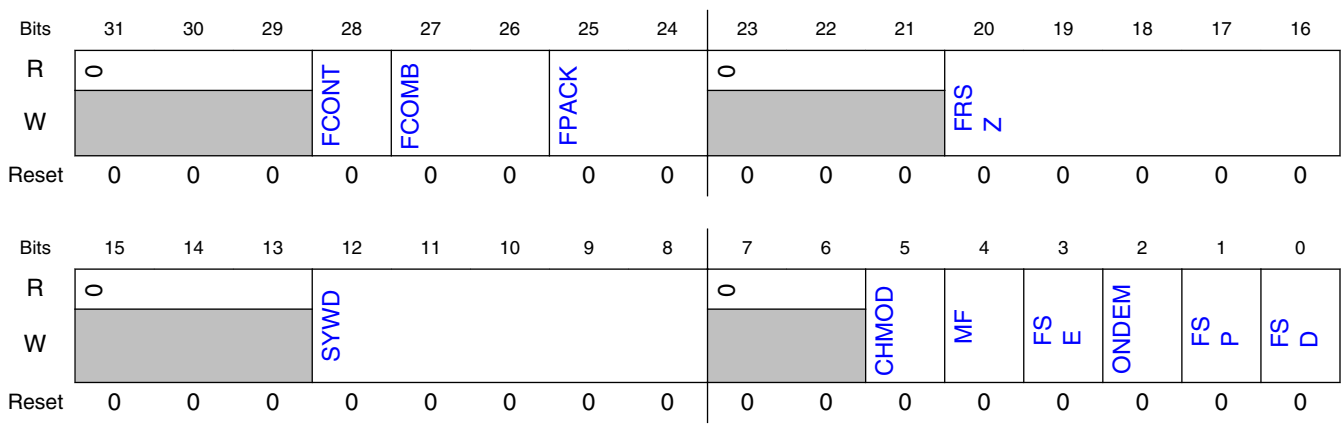
13.10.4.1.8.1 Offset

Register	Offset
TCR4	18h

13.10.4.1.8.2 Function

This register must not be altered when TCSR[TE] is set.

13.10.4.1.8.3 Diagram



13.10.4.1.8.4 Fields

Field	Function
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue transmitting after a FIFO error has been detected.

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td></td><td>0b - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared.
1b - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.</td></tr>
<tr><td>27-26
FCOMB</td><td>FIFO Combine Mode
When FIFO combine mode is enabled for FIFO writes, software writing to any FIFO data register will alternate the write among the enabled data channel FIFOs. For example, if two data channels are enabled then the first write will be performed to the first enabled data channel FIFO and the second write will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO writes will reset the pointer back to the first enabled data channel.
When FIFO combine mode is enabled for FIFO reads from the transmit shift registers, the transmit data channel output will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked word will be transmitted from the first enabled data channel FIFO and the second unmasked word will be transmitted from the second enabled data channel FIFO. Since the first word of the frame is always transmitted from the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels.
NOTE: This field is not supported in every instance. The following table includes only supported registers.
<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1_TCR4</td><td>—</td></tr><tr><td>—</td><td>SAI2_TCR4</td></tr><tr><td>—</td><td>SAI3_TCR4</td></tr><tr><td>—</td><td>SAI4_TCR4</td></tr><tr><td>—</td><td>SAI5_TCR4</td></tr><tr><td>—</td><td>SAI6_TCR4</td></tr></table>
00b - FIFO combine mode disabled.
01b - FIFO combine mode enabled on FIFO reads (from transmit shift registers).
10b - FIFO combine mode enabled on FIFO writes (by software).
11b - FIFO combine mode enabled on FIFO reads (from transmit shift registers) and writes (by software).</td></tr>
<tr><td>25-24
FPACK</td><td>FIFO Packing Mode
Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are loaded from the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO write pointer will only increment when the full 32-bit FIFO word has been written by software.
00b - FIFO packing is disabled
01b - Reserved
10b - 8-bit FIFO packing is enabled
11b - 16-bit FIFO packing is enabled</td></tr>
<tr><td>23-21
—</td><td>Reserved</td></tr>
<tr><td>20-16
FRSZ</td><td>Frame size
Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.</td></tr>
</table>

Table continues on the next page...

Field	Function
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 CHMOD	Channel Mode Configures if transmit data pins are configured for TDM mode or Output mode. 0b - TDM mode, transmit data pins are tri-stated when slots are masked or channels are disabled. 1b - Output mode, transmit data pins are never tri-stated and will output zero when slots are masked or channels are disabled.
4 MF	MSB First Configures whether the LSB or the MSB is transmitted first. 0b - LSB is transmitted first. 1b - MSB is transmitted first.
3 FSE	Frame Sync Early 0b - Frame sync asserts with the first bit of the frame. 1b - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear. 0b - Internal frame sync is generated continuously. 1b - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0b - Frame sync is active high. 1b - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0b - Frame sync is generated externally in Slave mode. 1b - Frame sync is generated internally in Master mode.

13.10.4.1.9 SAI Transmit Configuration 5 Register (TCR5)

13.10.4.1.9.1 Offset

Register	Offset
TCR5	1Ch

13.10.4.1.9.2 Function

This register must not be altered when TCSR[TE] is set.

13.10.4.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.10.4.1.9.4 Fields

Field	Function
31-29 —	Reserved
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit transmitted for each word in the frame. If configured for MSB First, the index of the next bit transmitted is one less than the current bit transmitted. If configured for LSB First, the index of the next bit transmitted is one more than the current bit transmitted. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

13.10.4.1.10 SAI Transmit Data Register (TDR0 - TDR7)

13.10.4.1.10.1 Offset

For a = 0 to 7:

Register	Offset
TDRa	20h + (a × 4h)

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_TDR0–TDR7	—
SAI2_TDR0	SAI2_TDR1–TDR7
SAI3_TDR0	SAI3_TDR1–TDR7
SAI4_TDR0	SAI4_TDR1–TDR7
SAI5_TDR0	SAI5_TDR1–TDR7
SAI6_TDR0	SAI6_TDR1–TDR7

13.10.4.1.10.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W	TDR															
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0															
W	TDR															
Reset	See Register reset values.															

13.10.4.1.10.3 Register reset values

Register	Reset value
TDR0	–: 0000_0000h
TDR1–TDR7	0000_0000h

13.10.4.1.10.4 Fields

Field	Function
31-0 TDR	Transmit Data Register Writes to this register when the transmit FIFO is not full will push the data written into the transmit data FIFO. Writes to this register when the transmit FIFO is full are ignored.

13.10.4.1.11 SAI Transmit FIFO Register (TFR0 - TFR7)

13.10.4.1.11.1 Offset

For a = 0 to 7:

Register	Offset
TFRa	40h + (a × 4h)

13.10.4.1.11.2 Function

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_T FR0– TFR7	—
SAI2_T FR0	SAI2_T FR1– TFR7
SAI3_T FR0	SAI3_T FR1– TFR7
SAI4_T FR0	SAI4_T FR1– TFR7
SAI5_T FR0	SAI5_T FR1– TFR7
SAI6_T FR0	SAI6_T FR1– TFR7

13.10.4.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	WCP	0							WFP							
W																
Reset	See Register reset values.															

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								RFP							
W																
Reset	See Register reset values.															

13.10.4.1.11.4 Register reset values

Register	Reset value
TFR0	–: 0000_0000h
TFR1–TFR7	0000_0000h

13.10.4.1.11.5 Fields

Field	Function
31 WCP	Write Channel Pointer When FIFO Combine mode is enabled for writes, indicates that this data channel is the next FIFO to be written. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. 0b - No effect. 1b - FIFO combine is enabled for FIFO writes and this FIFO will be written on the next FIFO write.
30-24 —	Reserved
23-16 WFP	Write FIFO Pointer FIFO write pointer for transmit data channel.
15-8 —	Reserved
7-0 RFP	Read FIFO Pointer FIFO read pointer for transmit data channel.

Field supported in	Field not supported in
SAI1_TFR0–TFR7	—
—	SAI2_TFR0
—	SAI3_TFR0
—	SAI4_TFR0
—	SAI5_TFR0
—	SAI6_TFR0

13.10.4.1.12 SAI Transmit Mask Register (TMR)

13.10.4.1.12.1 Offset

Register	Offset
TMR	60h

13.10.4.1.12.2 Function

This register is double-buffered and updates:

1. When TCSR[TE] is first set
2. At the end of each frame.

This allows the masked words in each frame to change from frame to frame.

13.10.4.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.10.4.1.12.4 Fields

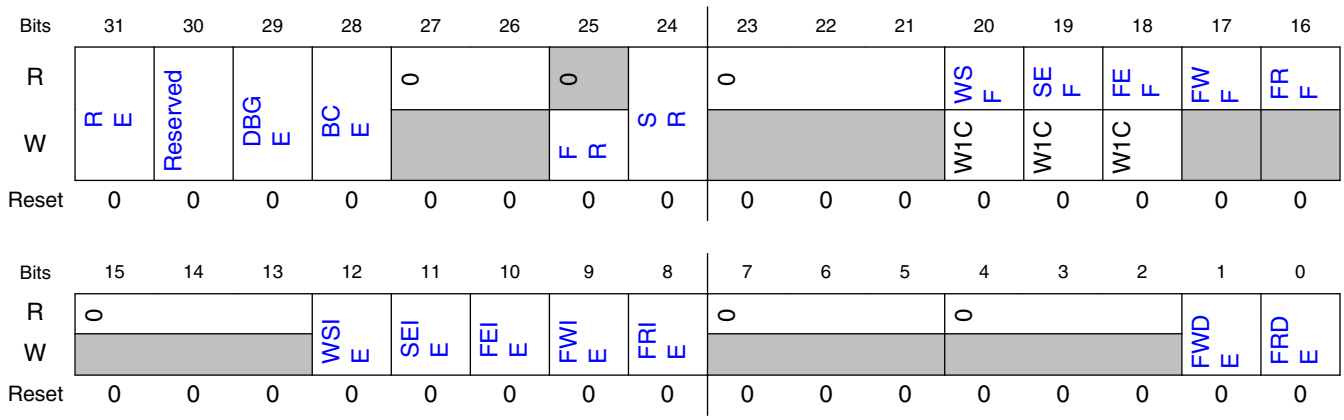
Field	Function
31-0 TWM	Transmit Word Mask Configures whether the transmit word is masked (transmit data pins are tri-stated or drive zero and transmit data not read from FIFO) for the corresponding word in the frame. 00000000000000000000000000000000b - Word N is enabled. 00000000000000000000000000000001b - Word N is masked. The transmit data pins are tri-stated or drive zero when masked.

13.10.4.1.13 SAI Receive Control Register (RCSR)

13.10.4.1.13.1 Offset

Register	Offset
RCSR	88h

13.10.4.1.13.2 Diagram



13.10.4.1.13.3 Fields

Field	Function
31 RE	Receiver Enable Enables/disables the receiver. When software clears this field, the receiver remains enabled, and this bit remains set, until the end of the current frame. 0b - Receiver is disabled. 1b - Receiver is enabled, or receiver has been disabled and has not yet reached end of frame.
30 —	Reserved. Software should only write zero to this reserved bit.
29 DBGE	Debug Enable Enables/disables receiver operation in Debug mode. The receive bit clock is not affected by Debug mode. 0b - Receiver is disabled in Debug mode, after completing the current frame. 1b - Receiver is enabled in Debug mode.
28 BCE	Bit Clock Enable Enables the receive bit clock, separately from RE. This field is automatically set whenever RE is set. When software clears this field, the receive bit clock remains enabled, and this field remains set, until the end of the current frame. 0b - Receive bit clock is disabled. 1b - Receive bit clock is enabled.
27-26 —	Reserved
25 FR	FIFO Reset Empties the FIFO, and sets the FIFO read and write pointers to the same value, which may or may not be zero. Reading this field will always return zero. FIFO pointers should only be reset when the receiver is disabled or the FIFO error flag is set. 0b - No effect. 1b - FIFO reset.
24 SR	Software Reset Resets the internal receiver logic including the FIFO pointers. Software-visible registers are not affected, except for the status registers. 0b - No effect.

Table continues on the next page...

Field	Function
	1b - Software reset.
23-21 —	Reserved
20 WSF	Word Start Flag Indicates that the start of the configured word has been detected. Write a logic 1 to this field to clear this flag. 0b - Start of word not detected. 1b - Start of word detected.
19 SEF	Sync Error Flag Indicates that an error in the externally-generated frame sync has been detected. Write a logic 1 to this field to clear this flag. 0b - Sync error not detected. 1b - Frame sync error detected.
18 FEF	FIFO Error Flag Indicates that an enabled receive FIFO has overflowed. Write a logic 1 to this field to clear this flag. 0b - Receive overflow not detected. 1b - Receive overflow detected.
17 FWF	FIFO Warning Flag Indicates that an enabled receive FIFO is full. 0b - No enabled receive FIFO is full. 1b - Enabled receive FIFO is full.
16 FRF	FIFO Request Flag Indicates that the number of words in an enabled receive channel FIFO is greater than the receive FIFO watermark. 0b - Receive FIFO watermark not reached. 1b - Receive FIFO watermark has been reached.
15-13 —	Reserved
12 WSIE	Word Start Interrupt Enable Enables/disables word start interrupts. 0b - Disables interrupt. 1b - Enables interrupt.
11 SEIE	Sync Error Interrupt Enable Enables/disables sync error interrupts. 0b - Disables interrupt. 1b - Enables interrupt.
10 FEIE	FIFO Error Interrupt Enable Enables/disables FIFO error interrupts. 0b - Disables the interrupt. 1b - Enables the interrupt.
9 FWIE	FIFO Warning Interrupt Enable Enables/disables FIFO warning interrupts. 0b - Disables the interrupt. 1b - Enables the interrupt.
8 FRIE	FIFO Request Interrupt Enable Enables/disables FIFO request interrupts.

Table continues on the next page...

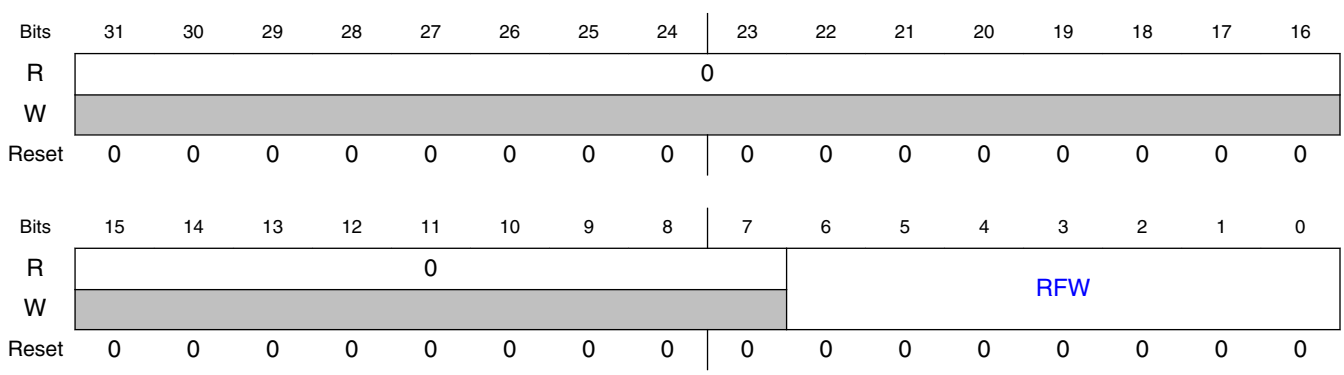
Field	Function
	0b - Disables the interrupt. 1b - Enables the interrupt.
7-5 —	Reserved
4-2 —	Reserved
1 FWDE	FIFO Warning DMA Enable Enables/disables DMA requests. 0b - Disables the DMA request. 1b - Enables the DMA request.
0 FRDE	FIFO Request DMA Enable Enables/disables DMA requests. 0b - Disables the DMA request. 1b - Enables the DMA request.

13.10.4.1.14 SAI Receive Configuration 1 Register (RCR1)

13.10.4.1.14.1 Offset

Register	Offset
RCR1	8Ch

13.10.4.1.14.2 Diagram



13.10.4.1.14.3 Fields

Field	Function
31-7	Reserved

Table continues on the next page...

Field	Function
—	
6-0 RFW	Receive FIFO Watermark Configures the watermark level for all enabled receiver channels.

13.10.4.1.15 SAI Receive Configuration 2 Register (RCR2)

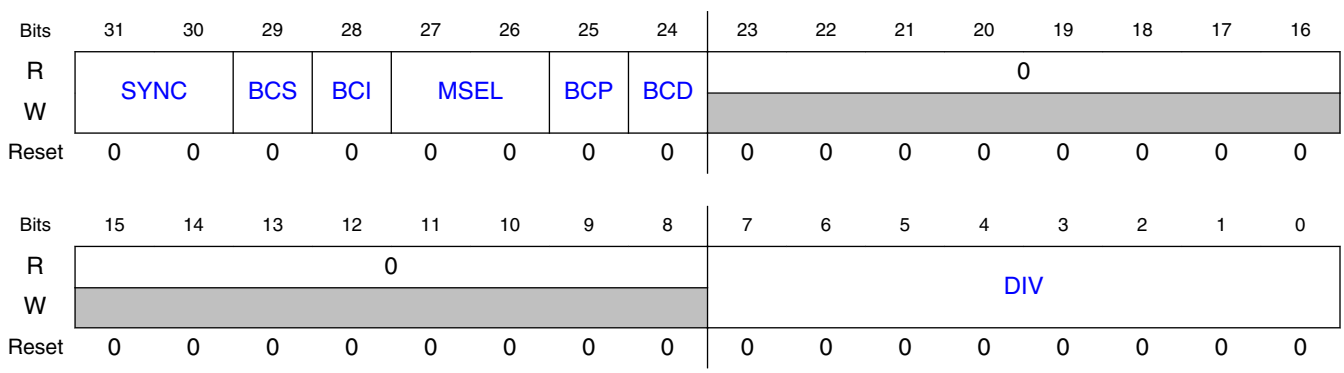
13.10.4.1.15.1 Offset

Register	Offset
RCR2	90h

13.10.4.1.15.2 Function

This register must not be altered when RCSR[RE] is set.

13.10.4.1.15.3 Diagram



13.10.4.1.15.4 Fields

Field	Function
31-30 SYNC	Synchronous Mode Configures between asynchronous and synchronous modes of operation. When configured for a synchronous mode of operation, the transmitter must be configured for asynchronous operation. 00b - Asynchronous mode. 01b - Synchronous with transmitter. 10b - Reserved. 11b - Reserved.
29	Bit Clock Swap

Table continues on the next page...

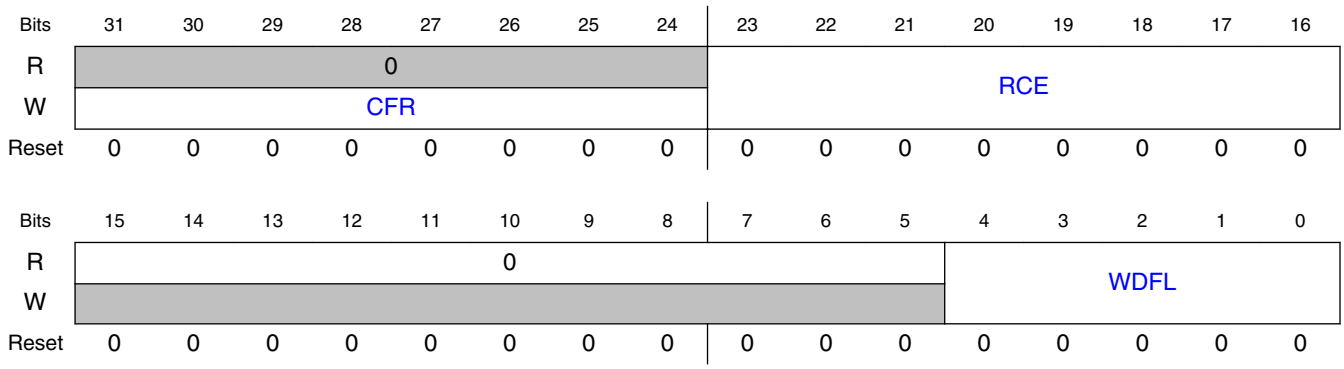
Field	Function
BCS	This field swaps the bit clock used by the receiver. When the receiver is configured in asynchronous mode and this bit is set, the receiver is clocked by the transmitter bit clock (TX_BCLK). This allows the transmitter and receiver to share the same bit clock, but the receiver continues to use the receiver frame sync (RX_SYNC). When the receiver is configured in synchronous mode, the transmitter BCS field and receiver BCS field must be set to the same value. When both are set, the transmitter and receiver are both clocked by the receiver bit clock (RX_BCLK) but use the transmitter frame sync (TX_SYNC). 0b - Use the normal bit clock source. 1b - Swap the bit clock source.
28 BCI	Bit Clock Input When this field is set and using an internally generated bit clock in either synchronous or asynchronous mode, the bit clock actually used by the receiver is delayed by the pad output delay (the receiver is clocked by the pad input as if the clock was externally generated). This has the effect of decreasing the data input setup time, but increasing the data output valid time. The slave mode timing from the datasheet should be used for the receiver when this bit is set. In synchronous mode, this bit allows the receiver to use the slave mode timing from the datasheet, while the transmitter uses the master mode timing. This field has no effect when configured for an externally generated bit clock . 0b - No effect. 1b - Internal logic is clocked as if bit clock was externally generated.
27-26 MSEL	MCLK Select Selects the audio Master Clock option used to generate an internally generated bit clock. This field has no effect when configured for an externally generated bit clock. **NOTE:** Depending on the device, some Master Clock options might not be available. See the chip-specific information for the availability and chip-specific meaning of each option. 00b - Bus Clock selected. 01b - Master Clock (MCLK) 1 option selected. 10b - Master Clock (MCLK) 2 option selected. 11b - Master Clock (MCLK) 3 option selected.
25 BCP	Bit Clock Polarity Configures the polarity of the bit clock. 0b - Bit Clock is active high with drive outputs on rising edge and sample inputs on falling edge. 1b - Bit Clock is active low with drive outputs on falling edge and sample inputs on rising edge.
24 BCD	Bit Clock Direction Configures the direction of the bit clock. 0b - Bit clock is generated externally in Slave mode. 1b - Bit clock is generated internally in Master mode.
23-8 —	Reserved
7-0 DIV	Bit Clock Divide Divides down the audio master clock to generate the bit clock when configured for an internal bit clock. The division value is (DIV + 1) * 2.

13.10.4.1.16 SAI Receive Configuration 3 Register (RCR3)

13.10.4.1.16.1 Offset

Register	Offset
RCR3	94h

13.10.4.1.16.2 Diagram



13.10.4.1.16.3 Fields

Field	Function
31-24 CFR	Channel FIFO Reset Resets the FIFO pointers for a specific channel. Reading this field will always return zero. FIFO pointers should only be reset when a channel is disabled or the FIFO error flag is set. The width of CFR field = the number of receive channels (call it N). For example, if CFR is 2 bits wide, then bit position 24 refers to receive channel 1 FIFO pointer and bit position 25 refers to receive channel 2 FIFO pointer. Setting bit 24 resets receive channel 1 FIFO pointer, and setting bit 25 enables receive channel 2 FIFO pointer. Setting bit N will reset receive channel N FIFO pointer. 0b - No effect. 1b - Receive data channel N FIFO is reset. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.
23-16	Receive Channel Enable

Field supported in	Field not supported in
SAI1_RCR3	—
—	SAI2_RCR3
—	SAI3_RCR3
—	SAI4_RCR3
—	SAI5_RCR3
—	SAI6_RCR3

Table continues on the next page...

<table>
<tr><th>Field</th><th>Function</th></tr>
<tr><td>RCE</td><td>Enables the corresponding data channel for receive operation. Changing this field will take effect immediately for generating the FIFO request and warning flags, but at the end of each frame for receive operation.

The width of RCE field = the number of receive channels (call it N). For example, if RCE field is 2 bits wide, then bit position 16 refers to receive channel 1 and bit position 17 refers to receive channel 2. Setting bit 16 enables receive channel 1, and setting bit 17 enables receive channel 2. Setting bit N will enable receive channel N.

0b - Receive data channel N is disabled.
1b - Receive data channel N is enabled.

NOTE: This field is not supported in every instance. The following table includes only supported registers.

<table>
<tr><th>Field supported in</th><th>Field not supported in</th></tr>
<tr><td>SAI1_RCR3</td><td>—</td></tr>
<tr><td>SAI2_RCR3[16]</td><td>SAI2_RCR3[23–17]</td></tr>
<tr><td>SAI3_RCR3[16]</td><td>SAI3_RCR3[23–17]</td></tr>
<tr><td>SAI4_RCR3[16]</td><td>SAI4_RCR3[23–17]</td></tr>
<tr><td>SAI5_RCR3[16]</td><td>SAI5_RCR3[23–17]</td></tr>
<tr><td>SAI6_RCR3[16]</td><td>SAI6_RCR3[23–17]</td></tr>
</table>
</td></tr>
<tr><td>15-5
—</td><td>Reserved</td></tr>
<tr><td>4-0
WDFL</td><td>Word Flag Configuration

Configures which word the start of word flag is set. The value written should be one less than the word number (for example, write zero to configure for the first word in the frame). When configured to a value greater than the Frame Size field, then the start of word flag is never set.</td></tr>
</table>

13.10.4.1.17 SAI Receive Configuration 4 Register (RCR4)

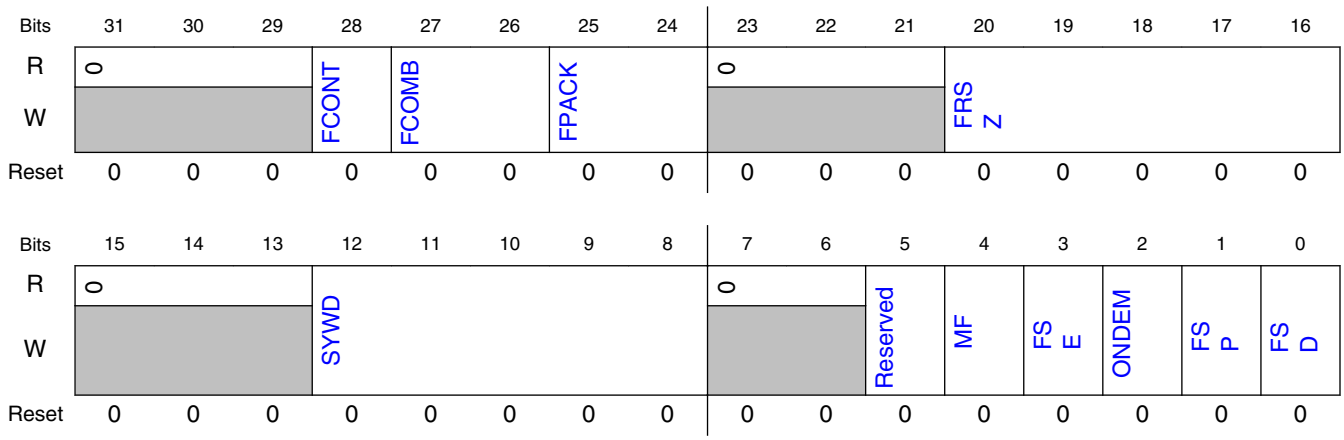
13.10.4.1.17.1 Offset

Register	Offset
RCR4	98h

13.10.4.1.17.2 Function

This register must not be altered when RCSR[RE] is set.

13.10.4.1.17.3 Diagram



13.10.4.1.17.4 Fields

Field	Function
31-29 —	Reserved
28 FCONT	FIFO Continue on Error Configures when the SAI will continue receiving after a FIFO error has been detected. 0b - On FIFO error, the SAI will continue from the start of the next frame after the FIFO error flag has been cleared. 1b - On FIFO error, the SAI will continue from the same word that caused the FIFO error to set after the FIFO warning flag has been cleared.
27-26 FCOMB	FIFO Combine Mode When FIFO combine mode is enabled for FIFO reads, software reading any FIFO data register will alternate the read among the enabled data channel FIFOs. For example, if two data channels are enabled then the first read will be performed to the first enabled data channel FIFO and the second read will be performed to the second enabled data channel FIFO. Reseting the FIFO or disabling FIFO combine mode for FIFO reads will reset the pointer back to the first enabled data channel. When FIFO combine mode is enabled for FIFO writes from the receive shift registers, the first enabled data channel input will alternate between the enabled data channel FIFOs. For example, if two data channels are enabled then the first unmasked received word will be stored in the first enabled data channel FIFO and the second unmasked received word will be stored in the second enabled data channel FIFO. Since the first word of the frame is always stored in the first enabled data channel FIFO, it is recommended that the number of unmasked words per frame is evenly divisible by the number of enabled data channels. **NOTE:** This field is not supported in every instance. The following table includes only supported registers. **Field supported in** / **Field not supported in** SAI1_RCR4 / — — / SAI2_RCR4 — / SAI3_RCR4

Table continues on the next page...

Field	Function
	<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>—</td><td>SAI4_RCR4</td></tr><tr><td>—</td><td>SAI5_RCR4</td></tr><tr><td>—</td><td>SAI6_RCR4</td></tr></table> 00b - FIFO combine mode disabled. 01b - FIFO combine mode enabled on FIFO writes (from receive shift registers). 10b - FIFO combine mode enabled on FIFO reads (by software). 11b - FIFO combine mode enabled on FIFO writes (from receive shift registers) and reads (by software).
25-24 FPACK	FIFO Packing Mode Enables packing of 8-bit data or 16-bit data into each 32-bit FIFO word. If the word size is greater than 8-bit or 16-bit then only the first 8-bit or 16-bits are stored to the FIFO. The first word in each frame always starts with a new 32-bit FIFO word and the first bit shifted must be configured within the first packed word. When FIFO packing is enabled, the FIFO read pointer will only increment when the full 32-bit FIFO word has been read by software. 00b - FIFO packing is disabled 01b - Reserved. 10b - 8-bit FIFO packing is enabled 11b - 16-bit FIFO packing is enabled
23-21 —	Reserved
20-16 FRSZ	Frame Size Configures the number of words in each frame. The value written must be one less than the number of words in the frame. For example, write 0 for one word per frame. The maximum supported frame size is 32 words.
15-13 —	Reserved
12-8 SYWD	Sync Width Configures the length of the frame sync in number of bit clocks. The value written must be one less than the number of bit clocks. For example, write 0 for the frame sync to assert for one bit clock only. The sync width cannot be configured longer than the first word of the frame.
7-6 —	Reserved
5 —	Reserved. Software should only write zero to this bit.
4 MF	MSB First Configures whether the LSB or the MSB is received first. 0b - LSB is received first. 1b - MSB is received first.
3 FSE	Frame Sync Early 0b - Frame sync asserts with the first bit of the frame. 1b - Frame sync asserts one bit before the first bit of the frame.
2 ONDEM	On Demand Mode When set, and the frame sync is generated internally, a frame sync is only generated when the FIFO warning flag is clear.

Table continues on the next page...

Field	Function
	0b - Internal frame sync is generated continuously. 1b - Internal frame sync is generated when the FIFO warning flag is clear.
1 FSP	Frame Sync Polarity Configures the polarity of the frame sync. 0b - Frame sync is active high. 1b - Frame sync is active low.
0 FSD	Frame Sync Direction Configures the direction of the frame sync. 0b - Frame Sync is generated externally in Slave mode. 1b - Frame Sync is generated internally in Master mode.

13.10.4.1.18 SAI Receive Configuration 5 Register (RCR5)

13.10.4.1.18.1 Offset

Register	Offset
RCR5	9Ch

13.10.4.1.18.2 Function

This register must not be altered when RCSR[RE] is set.

13.10.4.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0			WNW					0			W0W				
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0			FBT					0							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.10.4.1.18.4 Fields

Field	Function
31-29	Reserved

Table continues on the next page...

Field	Function
—	
28-24 WNW	Word N Width Configures the number of bits in each word, for each word except the first in the frame. The value written must be one less than the number of bits per word. Word width of less than 8 bits is not supported.
23-21 —	Reserved
20-16 W0W	Word 0 Width Configures the number of bits in the first word in each frame. The value written must be one less than the number of bits in the first word. Word width of less than 8 bits is not supported if there is only one word per frame.
15-13 —	Reserved
12-8 FBT	First Bit Shifted Configures the bit index for the first bit received for each word in the frame. If configured for MSB First, the index of the next bit received is one less than the current bit received. If configured for LSB First, the index of the next bit received is one more than the current bit received. The value written must be greater than or equal to the word width when configured for MSB First. The value written must be less than or equal to 31-word width when configured for LSB First.
7-0 —	Reserved

13.10.4.1.19 SAI Receive Data Register (RDR0 - RDR7)

13.10.4.1.19.1 Offset

For a = 0 to 7:

Register	Offset
RDRa	A0h + (a × 4h)

13.10.4.1.19.2 Function

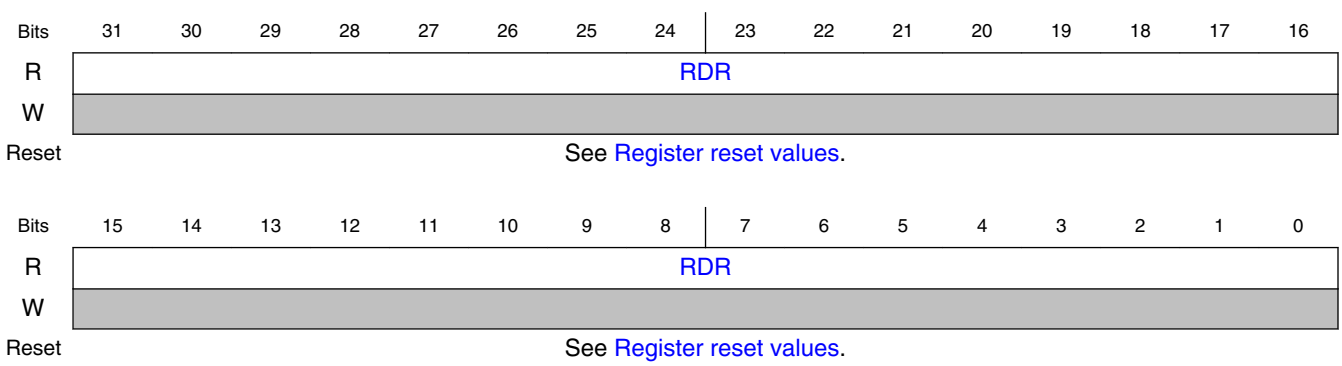
Reading this register introduces one additional peripheral clock wait state on each read.

NOTE

Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_R DR0– RDR7	—
SAI2_R DR0	SAI2_R DR1– RDR7
SAI3_R DR0	SAI3_R DR1– RDR7
SAI4_R DR0	SAI4_R DR1– RDR7
SAI5_R DR0	SAI5_R DR1– RDR7
SAI6_R DR0	SAI6_R DR1– RDR7

13.10.4.1.19.3 Diagram



13.10.4.1.19.4 Register reset values

Register	Reset value
RDR0	–: 0000_0000h
RDR1–RDR7	0000_0000h

13.10.4.1.19.5 Fields

Field	Function
31-0 RDR	Receive Data Register Reads from this register when the receive FIFO is not empty will return the data from the top of the receive FIFO. Reads from this register when the receive FIFO is empty are ignored.

13.10.4.1.20 SAI Receive FIFO Register (RFR0 - RFR7)

13.10.4.1.20.1 Offset

For a = 0 to 7:

Register	Offset
RFRa	C0h + (a × 4h)

13.10.4.1.20.2 Function

The MSB of the read and write pointers is used to distinguish between FIFO full and empty conditions. If the read and write pointers are identical, then the FIFO is empty. If the read and write pointers are identical except for the MSB, then the FIFO is full.

NOTE

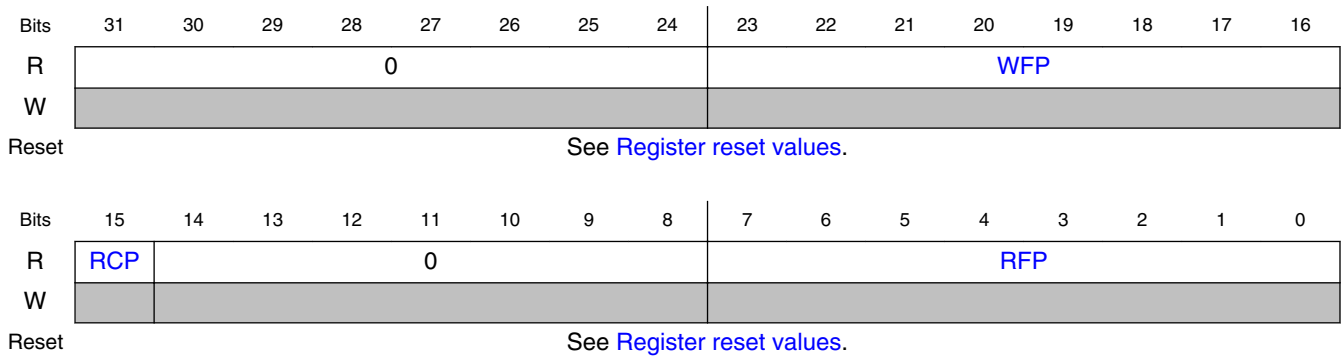
Each module instance supports a different number of registers.

Register supported	Register not supported
SAI1_RFR0–RFR7	—
SAI2_RFR0	SAI2_RFR1–RFR7
SAI3_RFR0	SAI3_RFR1–RFR7
SAI4_RFR0	SAI4_RFR1–RFR7
SAI5_RFR0	SAI5_RFR1–RFR7

Table continues on the next page...

Register supported	Register not supported
SAI6_R FR0	SAI6_R FR1– RFR7

13.10.4.1.20.3 Diagram



13.10.4.1.20.4 Register reset values

Register	Reset value
RFR0	–: 0000_0000h
RFR1–RFR7	0000_0000h

13.10.4.1.20.5 Fields

Field	Function
31-24 —	Reserved
23-16 WFP	Write FIFO Pointer FIFO write pointer for receive data channel.
15 RCP	Receive Channel Pointer When FIFO Combine mode is enabled for reads, indicates that this data channel is the next FIFO to be read. **NOTE:** This field is not supported in every instance. The following table includes only supported registers.

Table continues on the next page...

Field	Function
	<table><tr><th>Field supported in</th><th>Field not supported in</th></tr><tr><td>SAI1_RFR0–RFR7</td><td>—</td></tr><tr><td>—</td><td>SAI2_RFR0</td></tr><tr><td>—</td><td>SAI3_RFR0</td></tr><tr><td>—</td><td>SAI4_RFR0</td></tr><tr><td>—</td><td>SAI5_RFR0</td></tr><tr><td>—</td><td>SAI6_RFR0</td></tr></table> 0b - No effect. 1b - FIFO combine is enabled for FIFO reads and this FIFO will be read on the next FIFO read.
14-8 —	Reserved
7-0 RFP	Read FIFO Pointer FIFO read pointer for receive data channel.

13.10.4.1.21 SAI Receive Mask Register (RMR)

13.10.4.1.21.1 Offset

Register	Offset
RMR	E0h

13.10.4.1.21.2 Function

This register is double-buffered and updates:

1. When RCSR[RE] is first set
2. At the end of each frame

This allows the masked words in each frame to change from frame to frame.

13.10.4.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RWM															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

13.10.4.1.21.4 Fields

Field	Function
31-0 RWM	Receive Word Mask Configures whether the receive word is masked (received data ignored and not written to receive FIFO) for the corresponding word in the frame. 00000000000000000000000000000000b - Word N is enabled. 00000000000000000000000000000001b - Word N is masked.

Chapter 14
Video Processing Unit (VPU)

14.1 VPU G1 (VPU_G1)

14.1.1 Overview

This block details the VPU G1 hardware based decoders and API. The decoders are able to decode H.264, VP8. The decoders conform to the H.264 Baseline, Main and High profile, and VP8 Main Profile

NOTE

The overall performance and capabilities of the decoders is system dependent. Formats and features that are presented depend on the chip configuration.

14.1.2 Features (G1)

The API is implemented on top of the VPU G1 decoder H.264 and VP8. The features of the decoders are noted in the following sections.

14.1.2.1 Decoder Features

The VPU G1 decoder has the following features:

- H.264 baseline profile, levels 1-4.1 decoding
  - Byte stream input
  - NAL unit input
- VP8 decoding
- Video post-processing features
  - Frame rotation 90 degrees left/right
  - Frame mirroring horizontally/vertically
  - Frame cropping

- Frame conversion from YCbCr formats to 16-bit or 32-bit RGB formats
- Frame scaling with maximum up-scaling factor of 3
- Two rectangular or alpha blending masks for output frame

The post-processing features can be used in pipeline with the decoder. The post-processing features can also be used as stand-alone, without performing any decoding.

- H.264 main and high profile, levels 1-4.1 decoding
- VP8 decoding

14.1.3 Video Frame Storage Format

This chapter describes the different input and output picture storage formats supported by the decoder and post-processor. Notice that some of the formats are only used for input or output.

14.1.3.1 YCbCr 4:2:0 Planar Format

In the planar format each video sample component forms one memory plane. The luminance and both chrominance planes must be stored in a linear and contiguous memory block as shown in Figure 3. The luminance samples are stored in raster-scan order (Y0Y1Y2Y3Y4….). The chrominance samples are stored in two planes also in raster scan order (Cb0Cb1Cb2Cb3… and Cr0Cr1Cr2Cr3…).

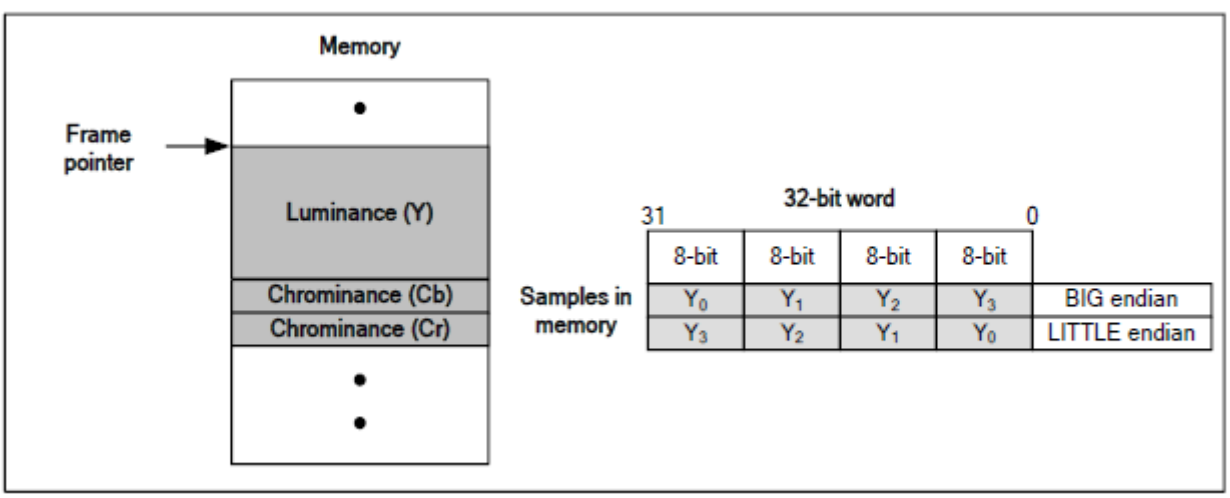


Figure 14-1. YCBCR 4:2:0 Planar Format External Memory Usage

The YCbCr 4:2:0 planar format is supported only as an input format for the postprocessor. In this format each pixel takes 12 bits of memory.

14.1.3.2 YCbCr 4:2:0 Semi-Planar Format

In semi-planar YCbCr 4:2:0 format the luminance samples form one plane in memory, and chrominance samples form another. The luminance and chrominance planes must be stored in a linear and contiguous memory block as presented in the following figure. The luminance pixels are stored in raster-scan order Y0Y1Y2Y3Y4…. The interleaved chrominance CbCr samples are stored in raster-scan order in memory as Cb0Cr0Cb1Cr1Cb2Cr2Cb3....

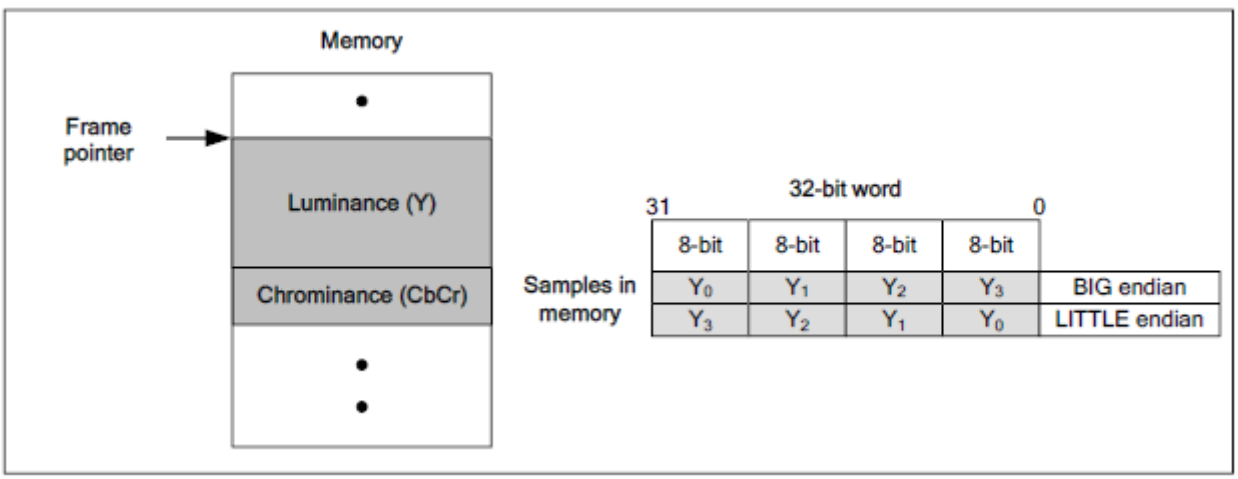


Figure 14-2. YCBCR 4:2:0 Planar Format External Memory Usage

The YCbCr 4:2:0 semi-planar format is supported both as an input and as an output format for the post-processor. In this format each pixel takes 12 bits of memory. As the chrominance components are interleaved, the bus load caused by this format can be slightly lower than with the planar format due to the reduced amount of non-sequential memory addressing.

14.1.3.3 YCbCr 4:2:2 Interleaved Format

In the interleaved YCbCr 4:2:2 format the pixel samples form a single plane in which the data has to be stored linearly and contiguously as shown in Figure 5. The pixel data is in raster scan order and the chrominance samples are interleaved between the luminance samples as Y0Cb0Y1Cr0Y2Cb1Y3Cr1….

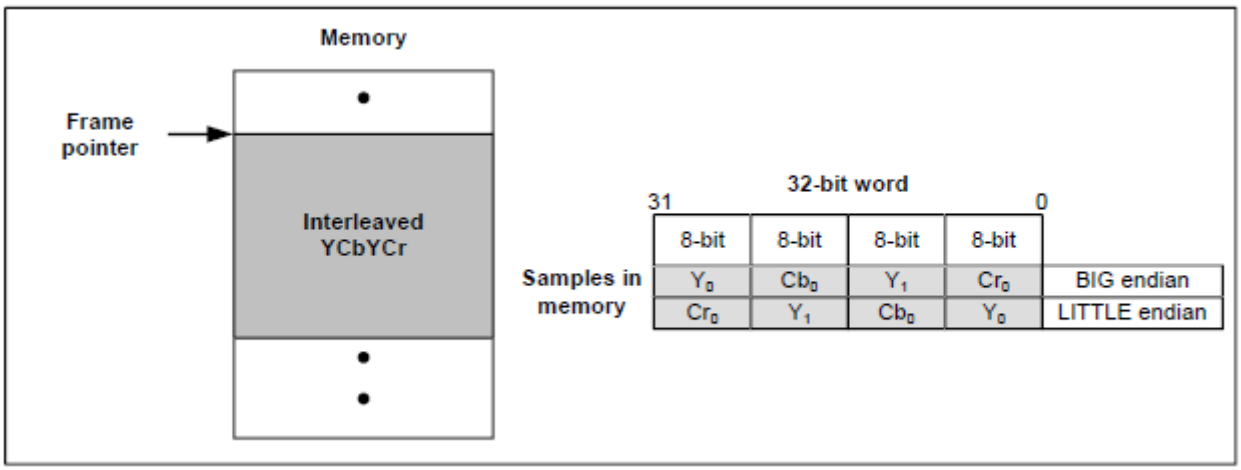


Figure 14-3. YCBCR 4:2:2 Interleaved Format External Memory Usage

The YCbCr 4:2:2 interleaved format is supported both as an input and as an output format for the post-processor. In this format each pixel takes 16 bits of memory. Although there is more data to be transferred than in the 4:2:0 formats, the interleaved format is bus effective as there are no non-sequential memory accesses caused by the plane changes.

14.1.3.4 G2 Decoder Tiled 4x4 Format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 4x4 tiled format, i.e. luminance data forms one plane in memory, and chrominance data forms another. The distinction from raster scan format is that the output picture is grouped into tiles, which are then stored linearly and contiguously in the memory. The chrominance tiles have to be stored right after the luminance tiles in external memory as shown in Figure 6.

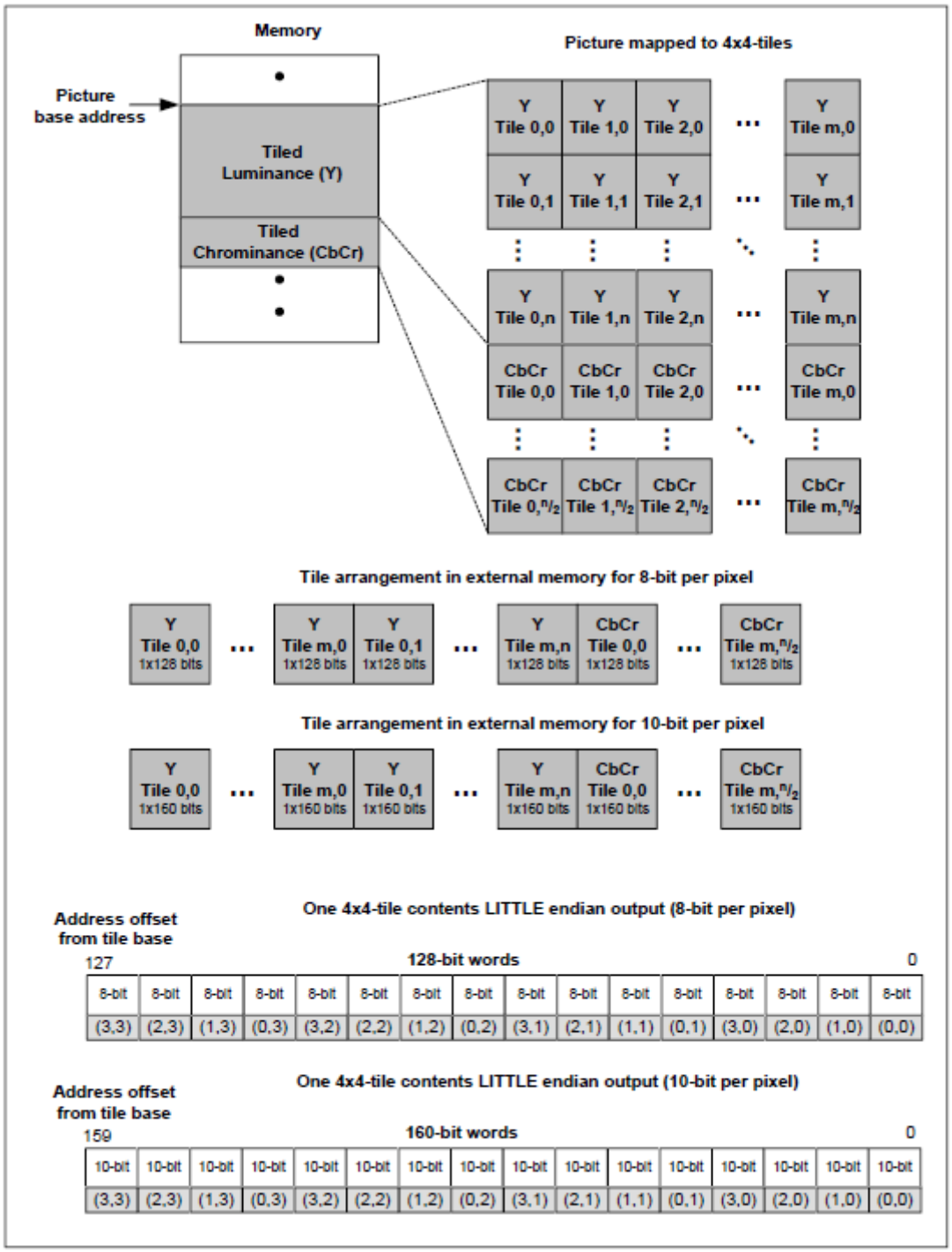


Figure 14-4. Tiled 4x4 Format External Memory Usage

14.1.3.5 RGB 16bpp Format

In this format each pixel is represented by 16 or less bits containing the red, blue and green samples. There are several 16bpp formats which use different number of bits for each sample. For example the RGB 5-5-5 format uses 5 bits for each sample and 1 bit is left unused or can represent a transparency flag, where RGB 5-6-5 uses 6 bits for the G sample and 5 bits for R and B samples. Common for all 16bpp types is that two pixels fit into one 32-bit space.

The data has to be stored linearly and contiguously in the memory as shown in Figure 7. The order of the sample bytes is always defined with a 16 bit word which will define the byte order in the memory. The pixel samples are stored in raster-scan order.

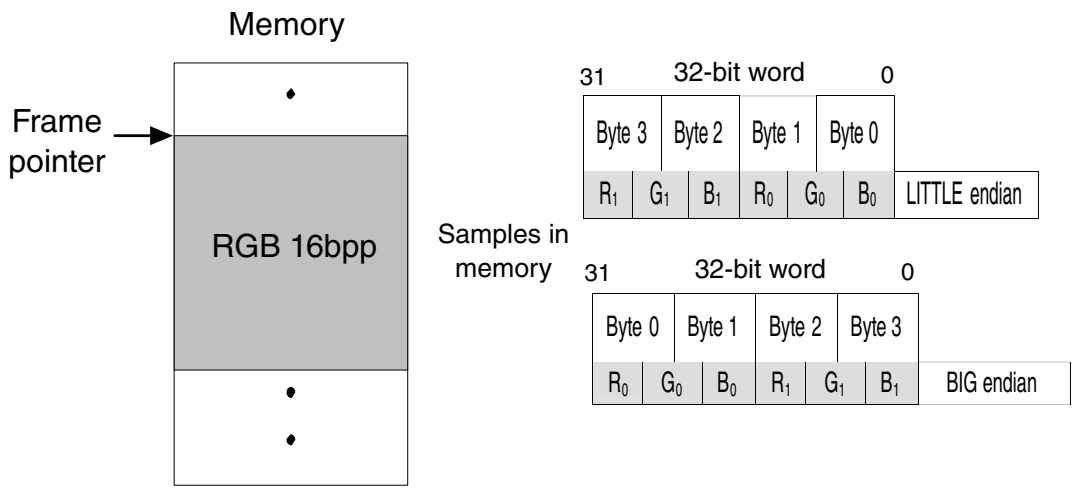


Figure 14-5. RGB 16bpp Format External Memory Usage

The RGB 16bpp formats are supported only as post-processor output formats.

14.1.3.6 RGB 32bpp Format

Any RGB format that has its pixels represented by more than 16 bits each is considered to be of 32bpp type. Typically in this format each pixel is represented by three bytes containing a red, blue and green sample and a 4th byte which can be empty or hold an alpha blending value. Common for all 32bpp types is that only one pixel fit into one 32-bit space.

The data has to be stored linearly and contiguously in the memory as shown in Figure 8. The order of the sample bytes is always defined on a 32 bit word and that will define the byte order in the memory (for example, ARGB order means that on a little endian system

the B sample will be stored in the lowest offset byte followed by the G, R and A samples. In a big endian system the A sample will be on the lowest offset byte). The pixel samples are stored in raster-scan order.

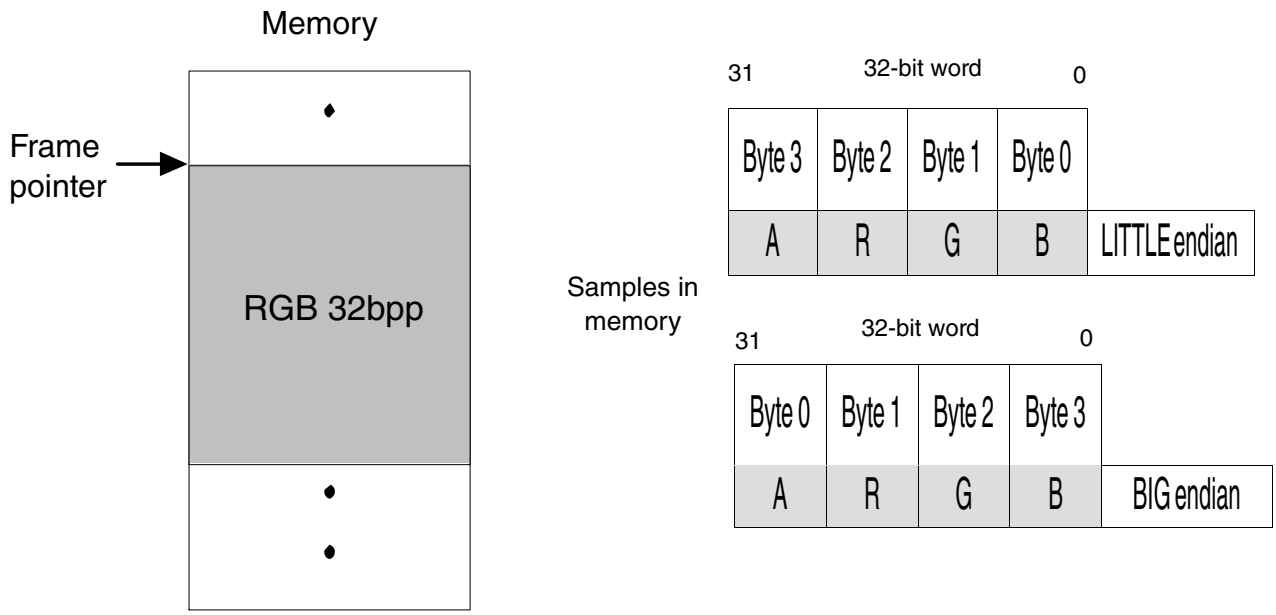


Figure 14-6. RGB 32bpp Format External Memory Usage

14.1.4 G1 Register Decoder

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
swreg0										
31:16	-	x	x	x	x	x	x	x	x	x
15:12	-	x	x	x	x	x	x	x	x	x
11:4	-	x	x	x	x	x	x	x	x	x
3	-	x	x	x	x	x	x	x	x	x
2:0	-	x	x	x	x	x	x	x	x	x
swreg1										
31:25	-									
24	sw_dec_pic_inf	x	x							
23:19	-									
18	sw_dec_timeout	x	x	x	x	x	x	x	x	x
17	sw_dec_slice_int			x						
16	sw_dec_error_int	x	x	x	x	x	x	x	x	x
15	sw_dec_aso_int	x							x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
14	sw_dec_buffer_int	x	x	x	x	x	x	x	x	x
13	sw_dec_bus_int	x	x	x	x	x	x	x	x	x
12	sw_dec_rdy_int	x	x	x	x	x	x	x	x	x
11	sw_dec_abort_int	x	x	x	x	x	x	x	x	x
10:9	-									
8	sw_dec_irq	x	x	x	x	x	x	x	x	x
7:6	-									
5	sw_dec_abort_e	x	x	x	x	x	x	x	x	x
4	sw_dec_irq_dis	x	x	x	x	x	x	x	x	x
3:1	-									
0	sw_dec_e	x	x	x	x	x	x	x	x	x
swreg2										
31:24	sw_dec_axi_rd_id	x	x	x	x	x	x	x	x	x
23	sw_dec_timeout_e	x	x	x	x	x	x	x	x	x
22	sw_dec_strswap32_e	x	x	x	x	x	x	x	x	x
21	sw_dec_strendian_e	x	x	x	x	x	x	x	x	x
20	sw_dec_inswap32_e	x	x	x	x	x	x	x	x	x
19	sw_dec_outswap32_e	x	x	x	x	x	x	x	x	x
18	sw_dec_data_disc_e	x	x	x	x	x	x	x	x	x
18	sw_dec_2chan_dis	x	x	x	x	x	x	x	x	x
17	sw_tiled_mode_msb	x	x		x	x	x	x	x	x
17	sw_dec_out_tiled_e	x	x	x	x	x				
16:11	sw_dec_latency	x	x	x	x	x	x	x	x	x
10	sw_dec_clk_gate_e	x	x	x	x	x	x	x	x	x
9	sw_dec_in_endian	x	x	x	x	x	x	x	x	x
8	sw_dec_out_endian	x	x	x	x	x	x	x	x	x
7:5	sw_priority_mode	x	x	x	x	x				
7	sw_tiled_mode_lsb	x	x	x	x	x	x	x	x	x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
6	sw_dec_adv_pre_dis	x	x		x	x	x	x	x	x
5	sw_dec_scmd_dis	x	x	x	x	x	x	x	x	x
4:0	sw_dec_max_burst	x	x	x	x	x	x	x	x	x
swreg3										
31:28	sw_dec_mode	x	x	x	x	x	x	x	x	x
27	sw_rlc_mode_e	x	x							
26	sw_skip_mode		x						x	x
25	sw_divx3_e		x							
24	sw_pjpeg_e			x						
23	sw_pic_interlace_e	x	x		x	x		x		x
22	sw_pic_fieldmode_e	x	x		x	x				x
21	sw_pic_b_e		x		x	x		x		x
20	sw_pic_inter_e		x		x	x	x		x	x
19	sw_pic_topfield_e	x	x		x	x				x
18	sw_fwd_interlace_e		x		x	x				
17	sw_sorenson_e		x							
16	sw_ref_topfield_e				x					
15	sw_dec_out_dis	x	x	x	x	x	x	x	x	x
14	sw_filtering_dis	x	x		x	x	x	x		x
13	sw_webp_e								x	
13	sw_mvc_e	x								
13	sw_pic_fixed_quant				x					x
12	sw_write_mvs_e	x	x		x	x	x	x	x	x
11	sw_reftopfirst_e				x					
10	sw_seq_mbaff_e	x								
9	sw_picord_count_e	x								
8	sw_dec_ahb_hlock_e	x	x	x	x	x	x	x	x	x
7:0	sw_dec_axi_wr_id	x	x	x	x	x	x	x	x	x
swreg4										
31:23	sw_pic_mb_width	x	x	x	x	x	x	x	x	x
22:19	sw_mb_width_off				x					

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
18:11	sw_pic_mb_height_p	x	x	x	x	x	x	x	x	x
10:7	sw_mb_height_off				x					
6	sw_alt_scan_e		x			x				
5	sw_topfieldfirst_e		x		x					
4:0	sw_ref_frames	x			x					
5:3	sw_pic_mb_w_ext			x					x	
2:0	sw_pic_mb_h_ext		x	x		x	x	x	x	
0	sw_pic_refer_flag									x
swreg5										
31:26	sw_strm_start_bit	x	x	x	x	x	x	x	x	x
25	sw_sync_marker_e		x	x	x					
24	sw_type1_quant_e	x	x							
23:19	sw_ch_qp_offset	x	x							
18:14	sw_ch_qp_offset2	x								
13:1	-	x								
0	sw_fieldpic_flag_e	x								
18:16	sw_intradc_vlc_thr		x							
15:0	sw_vop_time_incr		x							
24	sw_dq_profile				x					
23	sw_dqbi_level				x					
22	sw_range_red_frm_e				x					
21	-				x					
20	sw_fast_uvmc_e				x					
19:18	-				x					
17	sw_transdctab		x		x					
16:15	sw_transacfrm		x		x					
14:13	sw_transacfrm2		x		x					
12:10	sw_mb_mode_tab				x					
9:7	sw_mvtab				x					
6:4	sw_cbptab				x					

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
3:2	sw_2mv_blk_pat_tab				x					
1:0	sw_4mv_blk_pat_tab				x					
24	sw_qscale_type					x				
23:5	-					x				
4	sw_con_mv_e					x				
3:2	sw_intra_dc_prec					x				
1	sw_intra_vlc_tab					x				
0	sw_frame_pred_dct					x				
12:11	sw_jpeg_qtables			x						
10:8	sw_jpeg_mode			x						
7	sw_jpeg_filright_e			x						
6	sw_jpeg_stream_all			x						
5	sw_cr_ac_vlctable			x						
4	sw_cb_ac_vlctable			x						
3	sw_cr_dc_vlctable			x						
2	sw_cb_dc_vlctable			x						
1	sw_cr_dc_vlctable3			x						
0	sw_cb_dc_vlctable3			x						
23:18	sw_strm1_start_bit						x		x	
17	sw_huffman_e						x			
16	sw_multistream_e						x			
15:8	sw_boolean_value						x		x	
7:0	sw_boolean_range						x		x	
9:5	sw_alpha_offset									x
4:0	sw_beta_offset									x
swreg6										
31	sw_start_code_e	x			x					
30:25	sw_init_qp	x	x		x	x	x			x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
24	sw_ch_8pix_ileav_e	x								
31:24	sw_stream_len_ext								x	
23:0	sw_stream_len	x	x	x	x	x	x	x	x	x
swreg7										
31	sw_cabac_e	x								
30	sw_blackwhite_e	x								
29	sw_dir_8x8_infer_e	x								
28	sw_weight_pred_e	x								
27:26	sw_weight_bipr_idc	x								
25	sw_avs_h264_h_ext	x								x
24:21	-	x								
20:16	sw_framenum_len	x						x		
15:0	sw_framenum	x								
31	sw_bitplane0_e				x					
30	sw_bitplane1_e				x					
29	sw_bitplane2_e				x					
28:24	sw_alt_pquant				x					
23:20	sw_dq_edges				x					
19	sw_ttmbf				x					
18:14	sw_pqindex				x					
13	sw_vc1_height_ext				x					
12	sw_bilin_mc_e				x		x		x	
11	sw_uniqp_e				x					
10	sw_halfqp_e				x					
9:8	sw_ttfrm				x					
7	sw_2nd_byte_emul_e				x					
6	sw_dquant_e				x					
5	sw_vc1_adv_e				x					
31:26	sw_dct1_start_bit								x	
25:20	sw_dct2_start_bit								x	
13	sw_ch_mv_res								x	
11:9	sw_init_dc_match0								x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
8:6	sw_init_dc_match1								x	
5	sw_vp7_version								x	
swreg8										
31	sw_const_intra_e	x								
30	sw_filt_ctrl_pres	x								
29	sw_rdpic_cnt_pres	x								
28	sw_8x8trans_flag_e	x								
27:17	sw_refpic_mk_len	x								
16	sw_idr_pic_e	x								
15:0	sw_idr_pic_id	x								
31:24	sw_mv_scalefactor				x					
23:19	sw_ref_dist_fwd				x					
18:14	sw_ref_dist_bwd				x					
17:14	sw_loop_filt_limit						x			
13	sw_variance_test_e						x			
12:10	sw_mv_threshold						x			
9:0	sw_var_threshold						x			
8	sw_divx_idct_e		x							
7:0	sw_divx3_slice_size		x							
31:30	sw_rv_profile							x		
29:28	sw_rv_osv_quant							x		
27:14	sw_rv_fwd_scale							x		
13:0	sw_rv_bwd_scale							x		
31:16	sw_init_dc_comp0								x	
15:0	sw_init_dc_comp1								x	
swreg9										
31:24	sw_pps_id	x								
23:19	sw_refidx1_active	x								
18:14	sw_refidx0_active	x								
7:0	sw_poc_length	x								
24	sw_icomp0_e				x					

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
23:16	sw_iscale0				x					
15:0	sw_ishift0				x					
23:0	sw_stream1_len						x		x	
31:2	sw_mb_ctrl_base	x	x							
12:0	sw_pic_slice_am							x		
27:24	sw_coeffs_part_am								x	
swreg10										
31:2	sw_diff_mv_base	x	x							
29:25	sw_pinit_rlist_f9	x								
24:20	sw_pinit_rlist_f8	x								
19:15	sw_pinit_rlist_f7	x								
14:10	sw_pinit_rlist_f6	x								
9:5	sw_pinit_rlist_f5	x								
4:0	sw_pinit_rlist_f4	x								
24	sw_icomp1_e				x					
23:16	sw_iscale1				x					
15:0	sw_ishift1				x					
31:2	sw_segment_base								x	
1	sw_segment_upd_e								x	
0	sw_segment_e								x	
swreg11										
31:2	sw_i4x4_or_dc_base	x	x							
29:25	sw_pinit_rlist_f15	x								
24:20	sw_pinit_rlist_f14	x								
19:15	sw_pinit_rlist_f13	x								
14:10	sw_pinit_rlist_f12	x								
9:5	sw_pinit_rlist_f11	x								
4:0	sw_pinit_rlist_f10	x								
24	sw_icomp2_e				x					
23:16	sw_iscale2				x					
15:0	sw_ishift2				x					
29:24	sw_dct3_start_bit								x	
23:18	sw_dct4_start_bit								x	
17:12	sw_dct5_start_bit								x	
11:6	sw_dct6_start_bit								x	
5:0	sw_dct7_start_bit								x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
swreg12										
31:0	sw_rlc_vlc_base	x	x	x	x	x	x	x	x	x
swreg13										
31:2	sw_dec_out_base	x	x	x	x	x	x	x	x	x
1	sw_dpb_ilace_mode	x	x		x					x
swreg14										
31:2	sw_refer0_base	x	x		x	x	x	x	x	x
1	sw_refer0_field_e	x								x
0	sw_refer0_topc_e	x								x
31:2	sw_jpg_ch_out_base			x					x	
swreg15										
31:2	sw_refer1_base	x	x		x	x		x	x	x
1	sw_refer1_field_e	x								x
0	sw_refer1_topc_e	x								x
7:0	sw_jpeg_slice_h			x					x	
swreg16										
31:2	sw_refer2_base	x	x		x	x		x		x
1	sw_refer2_field_e	x								x
0	sw_refer2_topc_e	x								x
31	-									
30:24	sw_ac1_code6_cnt			x						
23:22	-									
21:16	sw_ac1_code5_cnt			x						
15:11	sw_ac1_code4_cnt			x						
10:7	sw_ac1_code3_cnt			x						
6	-									
5:3	sw_ac1_code2_cnt			x						
2	-									
1:0	sw_ac1_code1_cnt			x						
swreg17										

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
31:2	sw_refer3_base	x	x		x	x		x		x
1	sw_refer3_field_e	x								x
0	sw_refer3_topc_e	x								x
31:24	sw_ac1_code10_cnt			x						
23:16	sw_ac1_code9_cnt			x						
15:8	sw_ac1_code8_cnt			x						
7:0	sw_ac1_code7_cnt			x						
swreg18										
31:2	sw_refer4_base	x					x		x	
1	sw_refer4_field_e	x								
0	sw_refer4_topc_e	x								
31:16	sw_pic_header_len				x					
15:14	-				x					
13	sw_pic_4mv_e				x					
12	-				x					
11	sw_range_red_ref_e				x					
10:9	sw_vc1_difmv_range				x					
8	-				x					
7:6	sw_mv_range				x					
5	sw_overlap_e				x					
4:3	sw_overlap_method				x					
19	sw_alt_scan_flag_e		x			x				
18:15	sw_fcode_fwd_hor		x			x				
14:11	sw_fcode_fwd_ver					x				
10:7	sw_fcode_bwd_hor		x			x				
6:3	sw_fcode_bwd_ver					x				
2	sw_mv_accuracy_fwd		x		x	x				

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
1	sw_mv_accuracy_bwd					x				
1	sw_mpeg4_vc1_rc		x		x					
0	sw_prev_anc_type		x		x			x		x
31:24	sw_ac1_code14_cnt			x						
23:16	sw_ac1_code13_cnt			x						
15:8	sw_ac1_code12_cnt			x						
7:0	sw_ac1_code11_cnt			x						
0	sw_gref_sign_bias								x	
swreg19										
31:2	sw_refer5_base	x							x	
1	sw_refer5_field_e	x								
0	sw_refer5_topc_e	x								
26:0	sw_trb_per_trd_d0		x							
24	sw_icomp3_e				x					
23:16	sw_iscale3				x					
15:0	sw_ishift3				x					
31:27	sw_ac2_code4_cnt			x						
26:23	sw_ac2_code3_cnt			x						
22	-									
21:19	sw_ac2_code2_cnt			x						
18	-									
17:16	sw_ac2_code1_cnt			x						
15:8	sw_ac1_code16_cnt			x						
7:0	sw_ac1_code15_cnt			x						
29:24	sw_scan_map_1						x		x	
23:18	sw_scan_map_2						x		x	
17:12	sw_scan_map_3						x		x	
11:6	sw_scan_map_4						x		x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
5:0	sw_scan_map_5						x		x	
0	sw_aref_sign_bias								x	
swreg20										
31:2	sw_refer6_base	x								
31:2	sw_vp8_dec_ch_base	x								
1	sw_vp8_stride_e								x	
0	sw_vp8_ch_base_e								x	
1	sw_refer6_field_e	x								
0	sw_refer6_topc_e	x								
26:0	sw_trb_per_trd_dm1		x							
24	sw_icomp4_e				x					
23:16	sw_iscale4				x					
15:0	sw_ishift4				x					
31:24	sw_ac2_code8_cnt			x						
23:16	sw_ac2_code7_cnt			x						
15										
14:8	sw_ac2_code6_cnt			x						
7:6	-									
5:0	sw_ac2_code5_cnt			x						
29:24	sw_scan_map_6						x		x	
23:18	sw_scan_map_7						x		x	
17:12	sw_scan_map_8						x		x	
11:6	sw_scan_map_9						x		x	
5:0	sw_scan_map_10						x		x	
swreg21										
31:2	sw_refer7_base	x								
31:27	sw_y_stride_pow2								x	
26:22	sw_c_stride_pow2								x	
1	sw_refer7_field_e	x								
0	sw_refer7_topc_e	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
26:0	sw_trb_per_trd_d1		x							
31:24	sw_ac2_code12_cnt			x						
23:16	sw_ac2_code11_cnt			x						
15:8	sw_ac2_code10_cnt			x						
7:0	sw_ac2_code9_cnt			x						
29:24	sw_scan_map_11						x		x	
23:18	sw_scan_map_12						x		x	
17:12	sw_scan_map_13						x		x	
11:6	sw_scan_map_14						x		x	
5:0	sw_scan_map_15						x		x	
swreg22										
31:2	sw_refer8_base	x								
31:2	sw_dct_strm1_base								x	
1	sw_refer8_field_e	x								
0	sw_refer8_topc_e	x								
31:24	sw_ac2_code16_cnt			x						
23:16	sw_ac2_code15_cnt			x						
15:8	sw_ac2_code14_cnt			x						
7:0	sw_ac2_code13_cnt			x						
29:24	sw_scan_map_16						x			
23:18	sw_scan_map_17						x			
17:12	sw_scan_map_18						x			
11:6	sw_scan_map_19						x			
5:0	sw_scan_map_20						x			

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
swreg23										
31:2	sw_refer9_base	x								
31:2	sw_dct_strm2_base								x	
1	sw_refer9_field_e	x								
0	sw_refer9_topc_e	x								
31:28	sw_dc1_code8_cnt			x						
27:24	sw_dc1_code7_cnt			x						
23:20	sw_dc1_code6_cnt			x						
19:16	sw_dc1_code5_cnt			x						
15:12	sw_dc1_code4_cnt			x						
11:8	sw_dc1_code3_cnt			x						
7	-									
6:4	sw_dc1_code2_cnt			x						
3:2	-									
1:0	sw_dc1_code1_cnt			x						
29:24	sw_scan_map_21						x			
23:18	sw_scan_map_22						x			
17:12	sw_scan_map_23						x			
11:6	sw_scan_map_24						x			
5:0	sw_scan_map_25						x			
swreg24										
31:2	sw_refer10_base	x								
31:2	sw_dct_strm3_base								x	
1	sw_refer10_field_e	x								
0	sw_refer10_topc_e	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
31:28	sw_dc1_code16_cnt			x						
27:24	sw_dc1_code15_cnt			x						
23:20	sw_dc1_code14_cnt			x						
19:16	sw_dc1_code13_cnt			x						
15:12	sw_dc1_code12_cnt			x						
11:8	sw_dc1_code11_cnt			x						
7:4	sw_dc1_code10_cnt			x						
3:0	sw_dc1_code9_cnt			x						
29:24	sw_scan_map_26						x			
23:18	sw_scan_map_27						x			
17:12	sw_scan_map_28						x			
11:6	sw_scan_map_29						x			
5:0	sw_scan_map_30						x			
swreg25										
31:2	sw_refer11_base	x								
31:2	sw_dct_strm4_base								x	
1	sw_refer11_field_e	x								
0	sw_refer11_topc_e	x								
31:28	sw_dc2_code8_cnt			x						
27:24	sw_dc2_code7_cnt			x						
23:20	sw_dc2_code6_cnt			x						
19:16	sw_dc2_code5_cnt			x						
15:12	sw_dc2_code4_cnt			x						

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
11:8	sw_dc2_code3_cnt			x						
7	-									
6:4	sw_dc2_code2_cnt			x						
3:2	-									
1:0	sw_dc2_code1_cnt			x						
29:24	sw_scan_map_31						x			
23:18	sw_scan_map_32						x			
17:12	sw_scan_map_33						x			
11:6	sw_scan_map_34						x			
5:0	sw_scan_map_35						x			
swreg26										
31:2	sw_refer12_base	x								
31:2	sw_dct_strm5_base								x	
1	sw_refer12_field_e	x								
0	sw_refer12_topc_e	x								
31:28	sw_dc2_code16_cnt			x						
27:24	sw_dc2_code15_cnt			x						
23:20	sw_dc2_code14_cnt			x						
19:16	sw_dc2_code13_cnt			x						
15:12	sw_dc2_code12_cnt			x						
11:8	sw_dc2_code11_cnt			x						
7:4	sw_dc2_code10_cnt			x						
3:0	sw_dc2_code9_cnt			x						
29:24	sw_scan_map_36						x			

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
23:18	sw_scan_map_37						x			
17:12	sw_scan_map_38						x			
11:6	sw_scan_map_39						x			
5:0	sw_scan_map_40						x			
swreg27										
31:2	sw_refer13_base	x								
1	sw_refer13_field_e	x								
0	sw_refer13_topc_e	x								
31:28	sw_dc3_code8_cnt			x						
27:24	sw_dc3_code7_cnt			x						
23:20	sw_dc3_code6_cnt			x						
19:16	sw_dc3_code5_cnt			x						
15:12	sw_dc3_code4_cnt			x						
11:8	sw_dc3_code3_cnt			x						
7	-									
6:4	sw_dc3_code2_cnt			x						
3:2	-									
1:0	sw_dc3_code1_cnt			x						
31:2	sw_bitpl_ctrl_base				x		x		x	
swreg28										
31:2	sw_refer14_base	x								
31:2	sw_dct_strm6_base								x	
1	sw_refer14_field_e	x								
0	sw_refer14_topc_e	x								
31:16	sw_ref_invd_cur_1									x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
15:0	sw_ref_invd_cur_0									x
31:28	sw_dc3_code16_cnt			x						
27:24	sw_dc3_code15_cnt			x						
23:20	sw_dc3_code14_cnt			x						
19:16	sw_dc3_code13_cnt			x						
15:12	sw_dc3_code12_cnt			x						
11:8	sw_dc3_code11_cnt			x						
7:4	sw_dc3_code10_cnt			x						
3:0	sw_dc3_code9_cnt			x						
29:24	sw_scan_map_41						x			
23:18	sw_scan_map_42						x			
17:12	sw_scan_map_43						x			
11:6	sw_scan_map_44						x			
5:0	sw_scan_map_45						x			
swreg29										
31:2	sw_refer15_base	x								
31:2	sw_dct_strm7_base								x	
1	sw_refer15_field_e	x								
0	sw_refer15_topc_e	x								
31:16	sw_ref_invd_cur_3									x
15:0	sw_ref_invd_cur_2									x
29:24	sw_scan_map_46						x			
23:18	sw_scan_map_47						x			

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
17:12	sw_scan_map_48						x			
11:6	sw_scan_map_49						x			
5:0	sw_scan_map_50						x			
swreg30										
31:16	sw_refer1_nbr	x								
15:0	sw_refer0_nbr	x								
31:16	sw_ref_dist_cur_1									x
15:0	sw_ref_dist_cur_0									x
31	sw_filt_type								x	
30:28	sw_filt_sharpness								x	
27:21	sw_filt_mb_adj_0								x	
20:14	sw_filt_mb_adj_1								x	
13:7	sw_filt_mb_adj_2								x	
6:0	sw_filt_mb_adj_3								x	
swreg31										
31:16	sw_refer3_nbr	x								
15:0	sw_refer2_nbr	x								
29:24	sw_scan_map_51						x			
23:18	sw_scan_map_52						x			
17:12	sw_scan_map_53						x			
11:6	sw_scan_map_54						x			
5:0	sw_scan_map_55						x			
31:16	sw_ref_dist_cur_3									x
15:0	sw_ref_dist_cur_2									x
27:21	sw_filt_ref_adj_0								x	
20:14	sw_filt_ref_adj_1								x	
13:7	sw_filt_ref_adj_2								x	
6:0	sw_filt_ref_adj_3								x	
swreg32										
31:16	sw_refer5_nbr	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
15:0	sw_refer4_nbr	x								
29:24	sw_scan_map_56						x			
23:18	sw_scan_map_57						x			
17:12	sw_scan_map_58						x			
11:6	sw_scan_map_59						x			
5:0	sw_scan_map_60						x			
31:16	sw_ref_invd_col_1									x
15:0	sw_ref_invd_col_0									x
23:18	sw_filt_level_0								x	
17:12	sw_filt_level_1								x	
11:6	sw_filt_level_2								x	
5:0	sw_filt_level_3								x	
swreg33										
31:16	sw_refer7_nbr	x								
15:0	sw_refer6_nbr	x								
29:24	sw_scan_map_61						x			
23:18	sw_scan_map_62						x			
17:12	sw_scan_map_63						x			
31:16	sw_ref_invd_col_3									x
15:0	sw_ref_invd_col_2									x
31:27	sw_quant_delta_0								x	
26:22	sw_quant_delta_1								x	
21:11	sw_quant_0								x	
10:0	sw_quant_1								x	
swreg34										
31:16	sw_refer9_nbr	x								
15:0	sw_refer8_nbr	x								
31:22	sw_pred_bc_tap_0_3		x		x		x	x	x	x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
21:12	sw_pred_bc_tap_1_0				x		x	x	x	x
11:2	sw_pred_bc_tap_1_1				x		x	x	x	x
swreg35										
31:16	sw_refer11_nbr	x								
15:0	sw_refer10_nbr	x								
31:22	sw_pred_bc_tap_1_2				x		x	x	x	x
21:12	sw_pred_bc_tap_1_3				x		x	x	x	x
11:2	sw_pred_bc_tap_2_0				x		x		x	x
swreg36										
31:16	sw_refer13_nbr	x								
15:0	sw_refer12_nbr	x								
31:22	sw_pred_bc_tap_2_1				x		x		x	x
21:12	sw_pred_bc_tap_2_2				x		x		x	x
11:2	sw_pred_bc_tap_2_3				x		x		x	x
swreg37										
31:16	sw_refer15_nbr	x								
15:0	sw_refer14_nbr	x								
31:22	sw_pred_bc_tap_3_0						x		x	
21:12	sw_pred_bc_tap_3_1						x		x	
11:2	sw_pred_bc_tap_3_2						x		x	
swreg38										
31:0	sw_refer_lterm_e	x								
31:22	sw_pred_bc_tap_3_3						x		x	
21:12	sw_pred_bc_tap_4_0						x		x	
11:2	sw_pred_bc_tap_4_1						x		x	
swreg39										
31:0	sw_refer_valid_e	x								
31:22	sw_pred_bc_tap_4_2						x		x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
21:12	sw_pred_bc_tap_4_3						x		x	
11:2	sw_pred_bc_tap_5_0						x		x	
swreg40										
31:2	sw_qtable_base	x	x	x		x	x	x	x	
swreg41										
31:2	sw_dir_mv_base	x	x	x	x	x		x	x	x
swreg42										
29:25	sw_binit_rlist_b2	x								
24:20	sw_binit_rlist_f2	x								
19:15	sw_binit_rlist_b1	x								
14:10	sw_binit_rlist_f1	x								
9:5	sw_binit_rlist_b0	x								
4:0	sw_binit_rlist_f0	x								
31:22	sw_pred_bc_tap_5_1						x		x	
21:12	sw_pred_bc_tap_5_2						x		x	
11:2	sw_pred_bc_tap_5_3						x		x	
31:24	sw_weight_qp_1									x
23:21	sw_ref_delta_col_0									x
20:18	sw_ref_delta_col_1									x
17:15	sw_ref_delta_col_2									x
14:12	sw_ref_delta_col_3									x
11:9	sw_ref_delta_cur_0									x
8:6	sw_ref_delta_cur_1									x
5:3	sw_ref_delta_cur_2									x
2:0	sw_ref_delta_cur_3									x
swreg43										
29:25	sw_binit_rlist_b5	x								
24:20	sw_binit_rlist_f5	x								
19:15	sw_binit_rlist_b4	x								
14:10	sw_binit_rlist_f4	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
9:5	sw_binit_rlist_b3	x								
4:0	sw_binit_rlist_f3	x								
31:22	sw_pred_bc_tap_6_0						x		x	
21:12	sw_pred_bc_tap_6_1						x		x	
11:2	sw_pred_bc_tap_6_2						x		x	
31:24	sw_weight_qp_2									x
23:16	sw_weight_qp_3									x
15:8	sw_weight_qp_4									x
7:0	sw_weight_qp_5									x
swreg44										
29:25	sw_binit_rlist_b8	x								
24:20	sw_binit_rlist_f8	x								
19:15	sw_binit_rlist_b7	x								
14:10	sw_binit_rlist_f7	x								
9:5	sw_binit_rlist_b6	x								
4:0	sw_binit_rlist_f6	x								
31:22	sw_pred_bc_tap_6_3						x		x	
21:12	sw_pred_bc_tap_7_0						x		x	
11:2	sw_pred_bc_tap_7_1						x		x	
26	sw_dec_avsp_ena									x
25	sw_weight_qp_e									x
24:23	sw_weight_qp_model									x
22	sw_avs_aec_e									x
21	sw_no_fwd_ref_e									x
20	sw_pb_field_enhanced_e									x
19:14	sw_qp_delta_cb									x
13:8	sw_qp_delta_cr									x
7:0	sw_weight_qp_0									x
swreg45										
29:25	sw_binit_rlist_b11	x								
24:20	sw_binit_rlist_f11	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
19:15	sw_binit_rlist_b10	x								
14:10	sw_binit_rlist_f10	x								
9:5	sw_binit_rlist_b9	x								
4:0	sw_binit_rlist_f9	x								
31:22	sw_pred_bc_tap_7_2						x		x	
21:12	sw_pred_bc_tap_7_3						x		x	
11:10	sw_pred_tap_2_m1								x	
9:8	sw_pred_tap_2_4								x	
7:6	sw_pred_tap_4_m1								x	
5:4	sw_pred_tap_4_4								x	
3:2	sw_pred_tap_6_m1								x	
1:0	sw_pred_tap_6_4								x	
31:0	sw_dir_mv_base2									x
swreg46										
31:30										
29:25	sw_binit_rlist_b14	x								
24:20	sw_binit_rlist_f14	x								
19:15	sw_binit_rlist_b13	x								
14:10	sw_binit_rlist_f13	x								
9:5	sw_binit_rlist_b12	x								
4:0	sw_binit_rlist_f12	x								
31:27	sw_quant_delta_2								x	
26:22	sw_quant_delta_3								x	
21:11	sw_quant_2								x	
10:0	sw_quant_3								x	
swreg47										
31:30	-									
29:25	sw_pinit_rlist_f3	x								

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
24:20	sw_pinit_rlist_f2	x								
19:15	sw_pinit_rlist_f1	x								
14:10	sw_pinit_rlist_f0	x								
9:5	sw_binit_rlist_b15	x								
4:0	sw_binit_rlist_f15	x								
31:27	sw_quant_delta_4								x	
21:11	sw_quant_4								x	
10:0	sw_quant_5								x	
swreg48										
31:23	sw_startmb_x								x	
22:14	sw_startmb_y								x	
13:12	sw_error_conc_mode								x	
11:0	-									
swreg49										
31:22	sw_pred_bc_tap_0_0	x	x		x		x	x		x
21:12	sw_pred_bc_tap_0_1	x	x		x		x	x		x
11:2	sw_pred_bc_tap_0_2	x	x		x		x	x		x
1:0	-									
swreg50										
31	SW_DEC_MPEG2_PROF					x				
30:29	SW_DEC_VC1_PROF				x					
28	SW_DEC_JPEG_PROF			x						
27:26	SW_DEC_MPEG4_PROF		x							
25:24	SW_DEC_H264_PROF	x								
23	SW_DEC_VP6_PROF						x			
22	SW_DEC_PJPEG_EXIST			x						
21	SW_DEC_OBUF_LEVEL	x	x		x	x	x	x	x	x
20	SW_REF_BUFF_EXIST	x	x		x	x	x	x	x	x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
19:16	SW_DEC_BUS_STRD	x	x	x	x	x	x	x	x	x
15:14	SW_DEC_SYNTH_LAN	x	x	x	x	x	x	x	x	x
13:12	SW_DEC_BUS_WIDTH	x	x	x	x	x	x	x	x	x
11	SW_DEC_SOREN_PROF		x							
10:0	SW_DEC_MAX_OWIDTH	x	x	x	x	x	x	x	x	x
swreg51										
31	sw_refbu_e	x	x		x	x	x	x	x	x
30:19	sw_refbu_thr	x	x		x	x	x	x	x	x
18:14	sw_refbu_picid	x	x		x	x	x	x	x	x
13	sw_refbu_eval_e	x	x		x	x	x	x	x	x
12	sw_refbu_fparmod_e	x	x		x	x	x	x	x	x
11:9	-									
8:0	sw_refbu_y_offset	x	x		x	x	x	x	x	x
swreg52										
31:16	sw_refbu_hit_sum	x	x		x	x	x	x	x	x
15:0	sw_refbu_intra_sum	x	x		x	x	x	x	x	x
swreg53										
31:22	-									
21:0	sw_refbu_y_mv_sum	x	x		x	x	x	x	x	x
swreg54										
31	SW_DEC_JPEG_EXTENS			x						
30	SW_DEC_REFBU_ILACE	x	x		x	x				
29	SW_DEC_DIVX_PROF		x							
28:0	SW_REF_BUFF2_EXIST	x	x		x	x	x	x	x	x
27:26	SW_DEC_RV_PROF							x		
25	SW_DEC_RTL_ROM		x		x			x		
24	SW_DEC_VP7_PROF								x	

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
23	SW_DEC_VP8_PROF								x	
22	SW_DEC_AVS_PROF									x
21:20	SW_DEC_MVC_PROF	x								
19	SW_DEC_WEBP_E								x	
18:17	SW_DEC_TILED_L	x	x		x	x	x	x	x	x
16	SW_DEC_VP8S_ARCH								x	
15:14	SW_DEC_MAX_OW_EXT	x	x		x	x	x	x	x	x
13:12	SW_DEC_ERRCO_LEVEL								x	
11	SW_VP8_STRIDE_E								x	
10	SW_DPB_FIELD_E	x	x	x	x	x	x	x	x	x
9:7	SW_DEC_CORE_AM	x	x	x	x	x	x	x	x	x
6:0	-									
swreg55										
31	sw_refbu2_buf_e	x	x		x	x	x	x	x	x
30:19	sw_refbu2_thr	x	x		x	x	x	x	x	x
18:14	sw_refbu2_picid	x	x		x	x	x	x	x	x
13:0	sw_apf_threshold	x	x		x	x	x	x	x	x
swreg56										
31:16	sw_refbu_top_sum	x	x		x	x	x	x	x	x
15:0	sw_refbu_bot_sum	x	x		x	x	x	x	x	x
swreg57										
31	fuse_dec_h264	x								
30	fuse_dec_mpeg4		x							
29	fuse_dec_mpeg2					x				
28	fuse_dec_sorenson		x							
27	fuse_dec_jpeg			x						
26	fuse_dec_vp6						x			
25	fuse_dec_vc1				x					
24	fuse_dec_pjpeg			x						

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
23	fuse_dec_divx		x							
22	fuse_dec_rv							x		
21	fuse_dec_vp7								x	
20	fuse_dec_vp8								x	
19	fuse_dec_avs									x
18	fuse_dec_mvc	x								x
17	-									
16	fuse_dec_maxw_4k	x	x	x	x	x	x	x	x	x
15	fuse_dec_maxw_1920	x	x		x	x	x	x	x	x
14	fuse_dec_maxw_1280	x	x		x	x	x	x	x	x
13	fuse_dec_maxw_720	x	x		x	x	x	x	x	x
12	fuse_dec_maxw_352	x	x		x	x	x	x	x	x
11:8	-									
7	fuse_dec_refbuffer	x	x		x	x	x	x	x	x
6:0	-									
swreg58										
31	sw_serv_merge_dis	x	x	x	x	x	x	x	x	x
30	sw_dec_multicore_e	x							x	
29	sw_dec_writestat_e	x							x	
28:27	sw_dec_mc_poll mode	x							x	
26:17	sw_dec_mc_polltime	x							x	
16:0	-									
swreg59										
31:2	sw_dec_ch8pix_base	x								
1:0	-									
Post-processor (swreg60-100)										
swreg102										
31:2	sw_dec_ch_base	x								x
1	-									
0	sw_ch_base_e	x								x

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
swreg103										
31:2	sw_refer0_ch_base	x								x
1:0	-									
swreg104										
31:2	sw_refer1_ch_base	x								x
1:0	-									
swreg105										
31:2	sw_refer2_ch_base	x								x
1:0	-									
swreg106										
31:2	sw_refer3_ch_base	x								x
1:0	-									
swreg107										
31:2	sw_refer4_ch_base	x								x
1:0	-									
swreg108										
31:2	sw_refer5_ch_base	x								x
1:0	-									
swreg109										
31:2	sw_refer6_ch_base	x								x
1:0	-									
swreg110										
31:2	sw_refer7_ch_base	x								x
1:0	-									
swreg111										
31:2	sw_refer8_ch_base	x								x
1:0	-									
swreg112										
31:2	sw_refer9_ch_base	x								x
1:0	-									
swreg113										

Table continues on the next page...

Bit	Name	Dec Modes								
		H.264	H.263	JPEG	VC-1	MPEG2/ MPEG1	VP6	RV	VP7 / VP8	AVS
31:2	sw_refer10_ch_base	x								x
1:0	-									
swreg114										
31:2	sw_refer11_ch_base	x								x
1:0	-									
swreg115										
31:2	sw_refer12_ch_base	x								x
1:0	-									
swreg116										
31:2	sw_refer13_ch_base	x								x
1:0	-									
swreg117										
31:2	sw_refer14_ch_base	x								x
1:0	-									
swreg118										
31:2	sw_refer15_ch_base	x								x
1:0	-									

14.1.5 VPU G1 Memory Map/Register Definition

14.1.5.1 VPU_G1 common registers descriptions

14.1.5.1.1 VPU_G1 common registers memory map

VPU_G1 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
4h	Interrupt register decoder (SWREG1)	32	RW	0000_0000h
8h	Device configuration register decoder (SWREG2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
Ch	Decoder control register 0 (decmode,picture type etc) (SWREG3)	32	RW	0000_0000h
30h	Base address for RLC data (RLC) / stream start address/decoded end addr register (VLC) (SWREG12)	32	RW	0000_0000h
34h	Base address for decoded picture (SWREG13)	32	RW	0000_0000h
A0h	Base address for standard dependent tables (SWREG40)	32	RW	0000_0000h
A4h	Base address for direct mode motion vectors (SWREG41)	32	RW	0000_0000h
C0h	Error concealment register (SWREG48)	32	RW	0000_0000h
C4h	Prediction filter tap register for H264 (SWREG49)	32	RW	0000_0000h
C8h	Synthesis configuration register decoder 0 (SWREG50)	32	RO	Table 14-
CCh	Reference picture buffer control register (SWREG51)	32	RW	0000_0000h
D0h	Reference picture buffer information register 1 (SWREG52)	32	RO	0000_0000h
D4h	Reference picture buffer information register 2 (SWREG53)	32	RO	0000_0000h
D8h	Synthesis configuration register decoder 1 (SWREG54)	32	RO	0000_0000h
DCh	Reference picture buffer 2 / Advanced prefetch control register (SWREG55)	32	RW	0000_0000h
E0h	Reference buffer information register 3 (SWREG56)	32	RO	0000_0000h
E4h	Decoder fuse register (SWREG57)	32	RO	Table 14-
E8h	Device configuration register decoder 2 + Multi core control register (SWREG58)	32	RW	0000_0000h
ECh	H264 Chrominance 8 pixel interleaved data base (SWREG59)	32	RW	0000_0000h
F0h	Interrupt register post-processor (SWREG60)	32	RW	0000_0000h
F4h	Device configuration register post-processor (SWREG61)	32	RW	0000_0100h
F8h	Deinterlace control register (SWREG62)	32	RW	0000_0000h
FCh	Base address for reading post-processing input picture luminance (top field/frame) (SWREG63)	32	RW	0000_0000h
100h	Base address for reading post-processing input picture Cb/Ch (top field/frame) (SWREG64)	32	RW	0000_0000h
104h	Base address for reading post-processing input picture Cr (SWRE G65)	32	RW	0000_0000h
108h	Base address for writing post-processed picture luminance/RGB (SWREG66)	32	RW	0000_0000h
10Ch	Base address for writing post-processed picture Ch (SWREG67)	32	RW	0000_0000h
110h	Register for contrast adjusting (SWREG68)	32	RW	0000_0000h
114h	Register for colour conversion and contrast adjusting/YUYV 422 channel orders (SWREG69)	32	RW	0000_0000h
118h	Register for colour conversion 0 (SWREG70)	32	RW	0000_0000h
11Ch	Register for colour conversion 1 + rotation mode (SWREG71)	32	RW	0000_0000h
120h	PP input size and -cropping register (SWREG72)	32	RW	0000_0000h
124h	PP input picture base address for Y bottom field (SWREG73)	32	RW	0000_0000h
128h	PP input picture base for Ch bottom field (SWREG74)	32	RW	0000_0000h
13Ch	Scaling register 0 ratio and padding for R and G (SWREG79)	32	RW	0000_0000h
140h	Scaling ratio register 1 and padding for B (SWREG80)	32	RW	0000_0000h

Table continues on the next page...

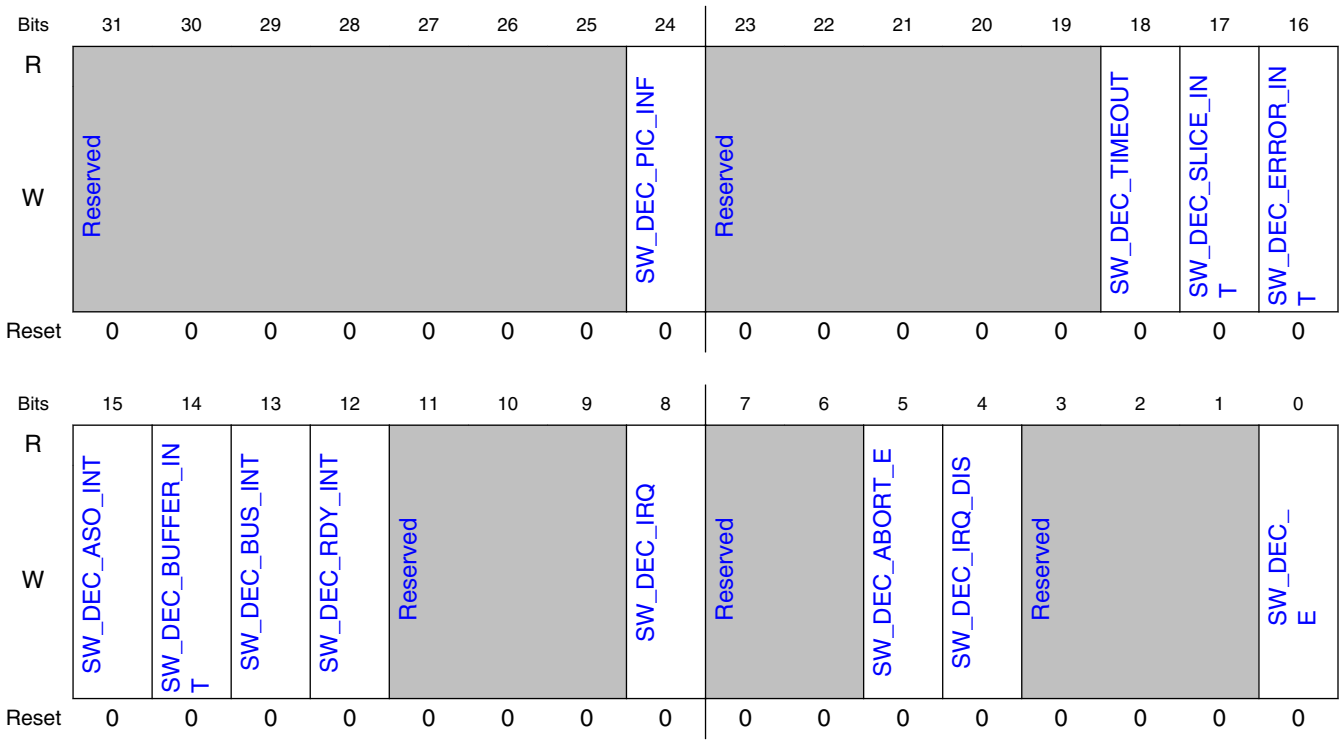
Offset	Register	Width (In bits)	Access	Reset value
144h	Scaling ratio register 2 (SWREG81)	32	RW	0000_0000h
148h	Rmask register (SWREG82)	32	RW	0000_0000h
14Ch	Gmask register (SWREG83)	32	RW	0000_0000h
150h	Bmask register (SWREG84)	32	RW	0000_0000h
154h	Post-processor control register (SWREG85)	32	RW	0000_0000h
158h	Mask 1 start coordinate register (SWREG86)	32	RW	0000_0000h
15Ch	Mask 2 start coordinate register + Mask extensions (SWREG87)	32	RW	0000_0000h
160h	Mask 1 size and PP original width register (SWREG88)	32	RW	0000_0000h
164h	Mask 2 size register + mask extensions (SWREG89)	32	RW	0000_0000h
168h	PiP register 0 (SWREG90)	32	RW	0000_0000h
16Ch	PiP register 1 and dithering control (SWREG91)	32	RW	0000_0000h
170h	Display width and PP input size extension register (SWREG92)	32	RW	0000_0000h
174h	Base address for alpha blend 1 gui component (SWREG93)	32	RW	0000_0000h
178h	Base address for alpha blend 2 gui component (SWREG94)	32	RW	0000_0000h
17Ch	Alpha blend input cropping register (scanline for cropping) (SWREG95)	32	RW	0000_0000h
18Ch	PP fuse register (SWREG99)	32	RO	Table 14-
190h	Synthesis configuration register post-processor (SWREG100)	32	RO	Table 14-
198h	Base address for H264 decoded chroma picture (SWREG102)	32	RW	0000_0000h
19Ch	Base address for reference chroma picture index 0 (SWREG103)	32	RW	0000_0000h
1A0h	Base address for reference chroma picture index 1 (SWREG104)	32	RW	0000_0000h
1A4h	Base address for reference chroma picture index 2 (SWREG105)	32	RW	0000_0000h
1A8h	Base address for reference chroma picture index 3 (SWREG106)	32	RW	0000_0000h
1ACh	Base address for reference chroma picture index 4 (SWREG107)	32	RW	0000_0000h
1B0h	Base address for reference chroma picture index 5 (SWREG108)	32	RW	0000_0000h
1B4h	Base address for reference chroma picture index 6 (SWREG109)	32	RW	0000_0000h
1B8h	Base address for reference chroma picture index 7 (SWREG110)	32	RW	0000_0000h
1BCh	Base address for reference chroma picture index 8 (SWREG111)	32	RW	0000_0000h
1C0h	Base address for reference chroma picture index 9 (SWREG112)	32	RW	0000_0000h
1C4h	Base address for reference chroma picture index 10 (SWREG113)	32	RW	0000_0000h
1C8h	Base address for reference chroma picture index 11 (SWREG114)	32	RW	0000_0000h
1CCh	Base address for reference chroma picture index 12 (SWREG115)	32	RW	0000_0000h
1D0h	Base address for reference chroma picture index 13 (SWREG116)	32	RW	0000_0000h
1D4h	Base address for reference chroma picture index 14 (SWREG117)	32	RW	0000_0000h
1D8h	Base address for reference chroma picture index 15 (SWREG118)	32	RW	0000_0000h

14.1.5.1.2 Interrupt register decoder (SWREG1)

14.1.5.1.2.1 Offset

Register	Offset
SWREG1	4h

14.1.5.1.2.2 Diagram



14.1.5.1.2.3 Fields

Field	Function
31-25 —	Reserved.
24 SW_DEC_PIC_INF	B slice detected. This signal is driven high during picture ready interrupt if B-type slice is found. This bit does not launch interrupt but is used to inform SW about h264 tools.
23-19 —	Reserved.
18 SW_DEC_TIMEOUT	Interrupt status bit decoder timeout. When high the decoder has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled

Table continues on the next page...

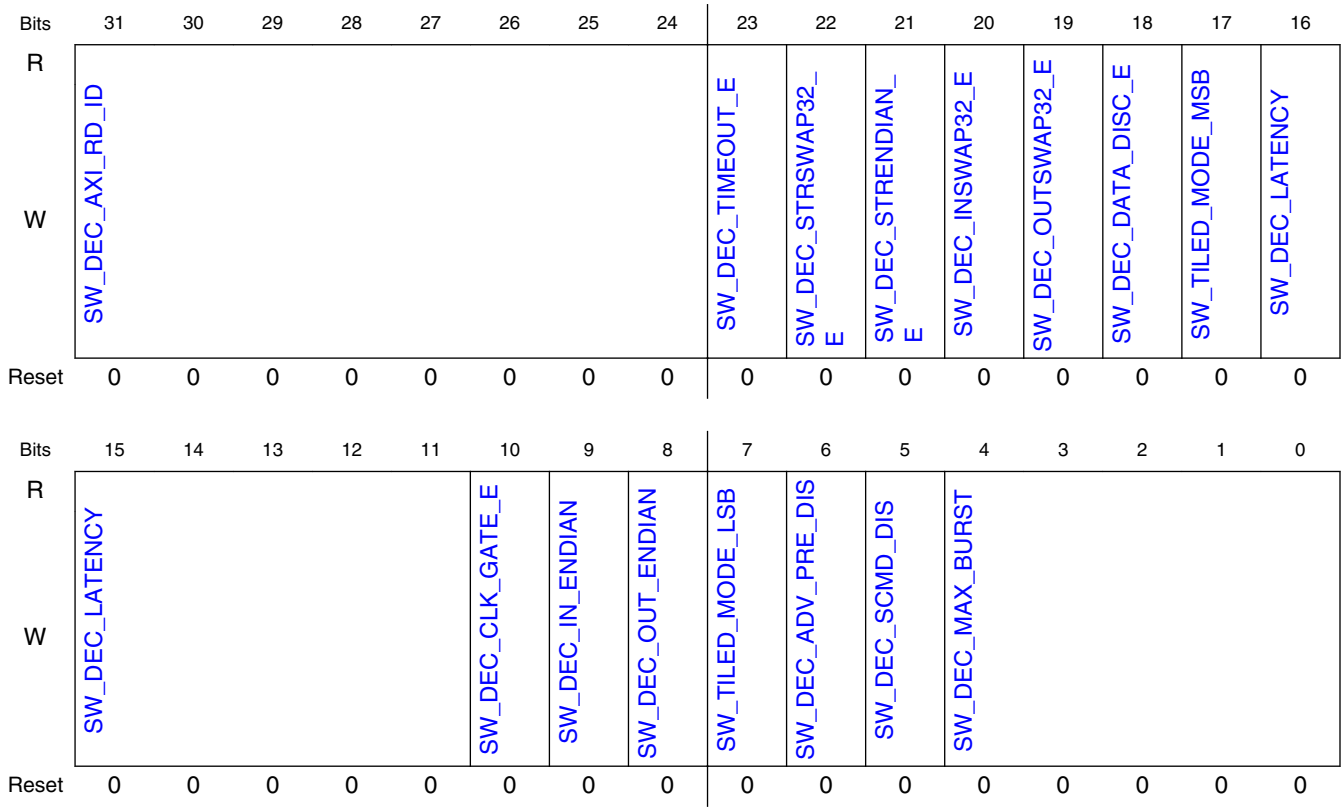
Field	Function
17 SW_DEC_SLICE_INT	Interrupt status bit dec_slice_decoded. When high SW must set new base addresses for sw_dec_out_base and sw_jpg_ch_out_base before resetting this status bit. Used for VP8 web-p modes
16 SW_DEC_ERROR_INT	Interrupt status bit input stream error. When high, an error is found in input data stream decoding. HW will self reset.
15 SW_DEC_ASO_INT	H264: Interrupt status bit ASO (Arbitrary Slice Ordering) detected. When high, ASO detected in input data stream decoding. HW will self reset. VP8: Error detected in Residual data. HW returns MB number in error concealment register for MB it detected it
14 SW_DEC_BUFFER_INT	Interrupt status bit input buffer empty. When high, input stream buffer is empty but picture is not ready. HW will not self reset.
13 SW_DEC_BUS_INT	Interrupt status bit bus. Error response from bus.
12 SW_DEC_RDY_INT	Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset.
11-9 —	Reserved.
8 SW_DEC_IRQ	Decoder IRQ. When high, decoder requests an interrupt. SW will reset this after interrupt is handled.
7-6 —	Reserved.
5 SW_DEC_ABORT_E	Abort decoding enable. Setting this bit high will cause HW to abort decoding and safely to reset itself down. After abort is complete the corresponding interrupt status is set and this bit is set low as well as the decoder enable.
4 SW_DEC_IRQ_DIS	Decoder IRQ disable. When high, there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses.
3-1 —	Reserved.
0 SW_DEC_E	Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.

14.1.5.1.3 Device configuration register decoder (SWREG2)

14.1.5.1.3.1 Offset

Register	Offset
SWREG2	8h

14.1.5.1.3.2 Diagram



14.1.5.1.3.3 Fields

Field	Function
31-24 SW_DEC_AXI_RD_ID	Read ID used for decoder reading services in AXI bus (if connected to AXI).
23 SW_DEC_TIMEOUT_E	Timeout interrupt enable. If enabled HW may return timeout interrupt in case HW gets stuck while decoding picture.
22 SW_DEC_STRSWAP32_E	Decoder input 32bit data swap for stream data (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
21 SW_DEC_STRENDIAN_E	Decoder input endian mode for stream data: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
20	Decoder input 32bit data swap for other than stream data (may be used for 64 bit environment): 0b - no swapping of 32 bit words

Table continues on the next page...

Field	Function
SW_DEC_INSWAP32_E	1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
19 SW_DEC_OUTSWAP32_E	Decoder output 32bit data swap (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
18 SW_DEC_DATA_DISC_E	Data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally. Note. If AHB maxburst 17 is used data discard cannot be enabled (causes conflict)
17 SW_TILED_MODE_MSB	Tiled mode msb. Concatenated to Tiled mode lsb which form 2 bit tiled mode. 0b - Tiled mode not enabled 1b - Tiled mode enabled for 8x4 tile size
16-11 SW_DEC_LATENCY	Decoder master interface additional latency. Can be used to slow down decoder HW between services in steps of 8 clock cycles: 0 = no latency 1 = minimum 8 cycles of IDLE between services 2 = minimum 16 cycles of IDLE between services . . . 63 = minimum latency of 504 cycles of IDLE between services
10 SW_DEC_CLK_GATE_E	Decoder dynamic clock gating enable: 0b - Clock is running for all structures 1b - Clock is gated for decoder structures that are not used. Note: Clock gating value can be changed only when decoder is disabled.
9 SW_DEC_IN_ENDIAN	Decoder input endian mode for other than stream data: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
8 SW_DEC_OUT_ENDIAN	Decoder output endian mode: 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
7 SW_TILED_MODE_LSB	Tiled mode lsb. Concatenated to Tiled mode msb which form 2 bit tiled mode. Defined in tiled_mode_msb
6 SW_DEC_ADV_PRE_DIS	Advanced PREFETCH mode disable (advanced reference picture reading mode for video)
5 SW_DEC_SCMD_DIS	9170 decoder and later->: AXI Single Command Multiple Data disable. 9170 axi wrapper supports this mode by default (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly)
4-0 SW_DEC_MAX_BURST	Maximum burst length for decoder bus transactions. Valid values: AHB: 1, 4, 8, 16 and 17. Other values will result in INCR type (undefined length) for all transfers. INCR used for lengths 2 and 3.

Field	Function
	17 = fixed bursts are used when possible. INCR bursts are used for lengths 2, 3, 5, 6, 7, 9, 10, 11, 12, 13, 14 and 15. OCP: 16-31 AXI: 1-16

14.1.5.1.4 Decoder control register 0 (decmode,picture type etc) (SWREG3)

14.1.5.1.4.1 Offset

Register	Offset
SWREG3	Ch

14.1.5.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_DEC_MODE				SW_RLC_MODE_E	SW_SKIP_MODE	Reserved	Reserved	SW_PIC_INTERLACE_E	SW_PIC_FIELDMODE_E	SW_PIC_B_E	SW_PIC_INTER_E	SW_PIC_TOPFIELD_E	SW_FWD_INTERLACE_E	Reserved	SW_REF_TOPFIELD_E
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_DEC_OUT_DIS	SW_FILTERING_DIS	SW_WEBP_E	SW_WRITE_MVS_E	SW_REFTOPFIRST_E	SW_SEQ_MBAFF_E	SW_PICORD_COUNT_E	SW_DEC_AHB_HLOCK_E	SW_DEC_AXI_WR_ID							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.4.3 Fields

Field	Function
31-28 SW_DEC_MODE	Decoding mode: 0000b - H.264 0001b - Reserved 0010b - Reserved 0011b - Reserved 0100b - Reserved 0101b - Reserved 0110b - Reserved 0111b - Reserved 1000b - Reserved 1001b - Reserved 1010b - VP8 1011b - Reserved 1100b - Reserved 1101b - Reserved 1110b - Reserved 1111b - Reserved
27 SW_RLC_MODE_E	RLC mode enable: 0b - HW decodes video from bit stream (VLC mode) + side information 1b - HW decodes video from RLC input data + side information (Differential MV's, separate DC coeffs, Intra 4x4 modes, MB control). Valid only for H.264 Baseline.
26 SW_SKIP_MODE	VP8: • 0 : HW decodes mb_coeff_skip -flag • 1 : HW does not decode mb_coeff_skip -flag
25 —	Reserved.
24 —	Reserved.
23 SW_PIC_INTERLACE_E	Coding mode of the current picture: 0b - progressive 1b - interlaced
22 SW_PIC_FIELDMODE_E	Structure of the current picture (residual structure) 0b - Frame structure. For H264, this means MBAFF structured picture for interlaced sequence 1b - Field structure
21 SW_PIC_B_E	B picture enable for current picture: 0b - picture type is I or P depending on sw_pic_inter_e 1b - picture type is B depending on sw_pic_inter_e (not valid for H264 since it is slice based information)
20 SW_PIC_INTER_E	Picture type. Please also see SW_PIC_B_E. 0b - Intra type (I) 1b - Inter type (P)
19 SW_PIC_TOPFIELD_E	If field structure is enabled, this bit informs which one of the fields is being decoded: 0b - bottom field 1b - top field
18 SW_FWD_INTERLACE_E	Coding mode of forward reference picture **NOTE:** For backward reference picture, the coding mode is always the same as for current picture. 0b - progressive

Table continues on the next page...

Field	Function
	1b - interlaced
17 —	Reserved.
16 SW_REF_TOPFIELD_E	Indicates which field should be used as reference if sw_ref_frames = '0': 0b - bottom field 1b - top field
15 SW_DEC_OUT_DIS	Disable decoder output picture writing: 0b - Decoder output picture is written to external memory 1b - Decoder output picture is not written to external memory
14 SW_FILTERING_DIS	De-block filtering disable: 0b - filtering is enabled for current picture 1b - filtering is disabled for current picture
13 SW_WEBP_E	SW_WEBP_E: Webp enable for VP8: • '0' = Normal VP8 • '1' = WEBP picture SW_MVC_E: Multi View Coding enable. Possible for H264 only.
12 SW_WRITE_MVS_E	Direct mode motion vector write enable for current picture / VPX motion vector write enable for error concealment purposes: 0b - Writing disabled for current picture 1b - The direct mode motion vectors are written to external memory. H264 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base.
11 SW_REFTOPFIRST_E	Indicates which FWD reference field has been decoded first. 0b - FWD reference bottom field 1b - FWD reference top field
10 SW_SEQ_MBAFF_E	Sequence includes MBAFF coded pictures
9 SW_PICORD_COUNT_E	h264_high config: Picture order count table read enable. If enabled HW will read picture order counts from memory in the beginning of picture
8 SW_DEC_AHB_HLOCK_E	AHB master HLOCK enable. When high the service is locked to decoder as long as it needs the bus (whenever decoder requests the bus it will be granted)
7-0 SW_DEC_AXI_WR_ID	Write ID used for decoder writing services in AXI bus (if connected to AXI)

14.1.5.1.5 Base address for RLC data (RLC) / stream start address/ decoded end addr register (VLC) (SWREG12)

14.1.5.1.5.1 Offset

Register	Offset
SWREG12	30h

14.1.5.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_RLC_VLC_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_RLC_VLC_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.5.3 Fields

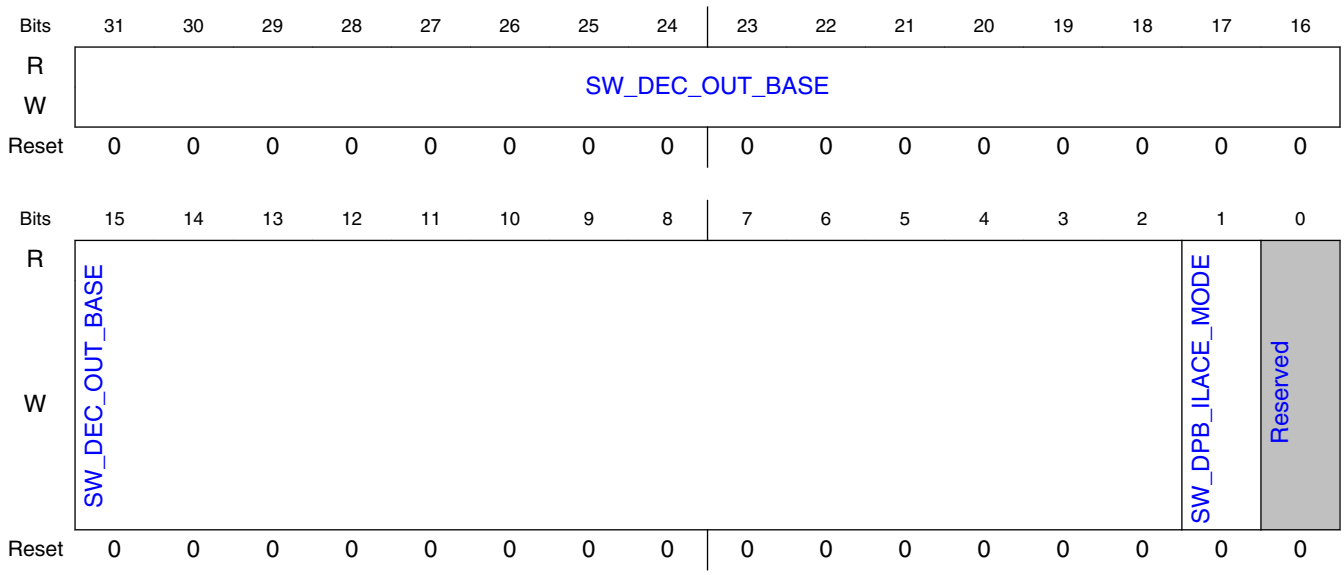
Field	Function
31-0 SW_RLC_VLC_BASE	RLC mode: Base address for RLC data (swreg3.sw_rlc_mode_e = 1). VLC mode: Stream start address / end addr+I288ess with byte precision (swreg4.rlc_mode_en = 0), start bit number in swreg5.stream_start_bit. When sw_dec_buffer_int is high or sw_dec_e is low this register contains HW return value of last_byte_address where stream has been read (and used) in accuracy of byte. For debug purposes the last_byte_address is also written when stream error/ASO is detected even though it may not be accurate. VP8: This base address is used as sw_dct_strm0_base including DCT stream for MB rows 0,2n.

14.1.5.1.6 Base address for decoded picture (SWREG13)

14.1.5.1.6.1 Offset

Register	Offset
SWREG13	34h

14.1.5.1.6.2 Diagram



14.1.5.1.6.3 Fields

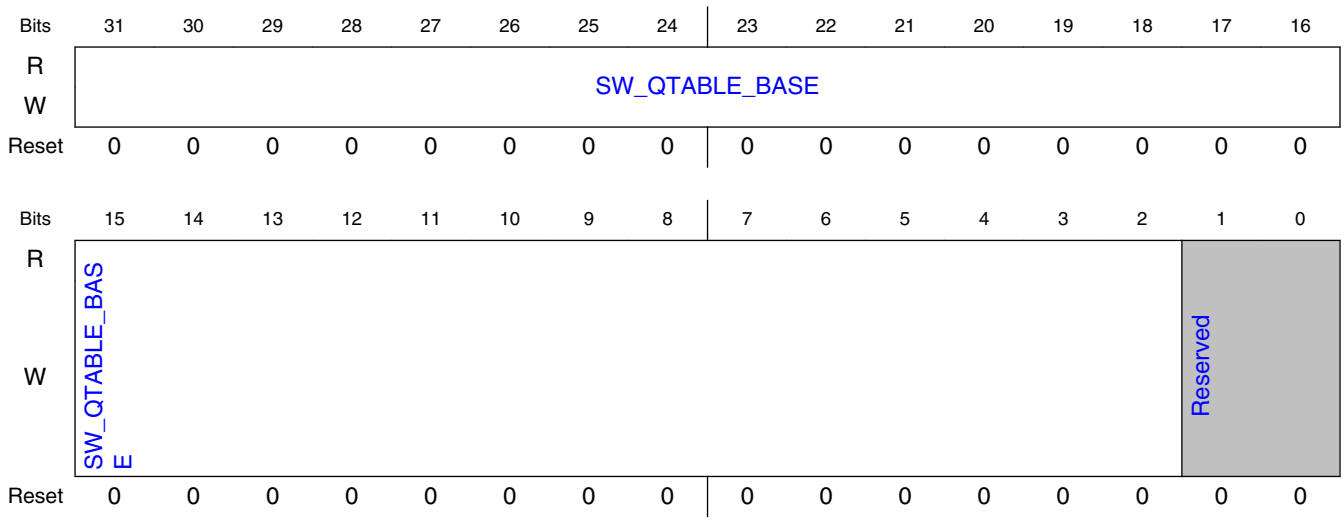
Field	Function
31-2 SW_DEC_OUT_BASE	Video: Base address for decoder output picture. Points directly to start of decoder output picture or field.
1 SW_DPB_ILACE_MODE	DPB ilaced mode: '0' : DPB consist of ilaced/progressive frames '1' : DPB consist of progressive frames / separate fields. This mode requires config support from HW
0 —	Reserved.

14.1.5.1.7 Base address for standard dependent tables (SWREG40)

14.1.5.1.7.1 Offset

Register	Offset
SWREG40	A0h

14.1.5.1.7.2 Diagram



14.1.5.1.7.3 Fields

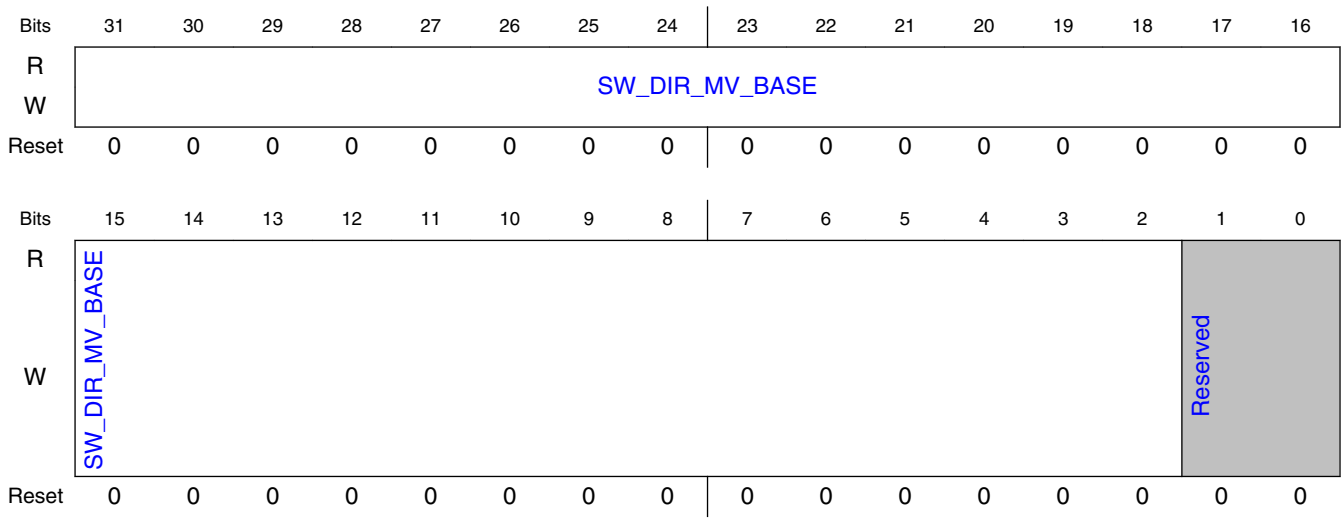
Field	Function
31-2 SW_QTABLE_BASE	Base address for standard dependent tables: H.264: base address for various tables VP8: base address for stream decoding tables
1-0 —	Reserved.

14.1.5.1.8 Base address for direct mode motion vectors (SWREG41)

14.1.5.1.8.1 Offset

Register	Offset
SWREG41	A4h

14.1.5.1.8.2 Diagram



14.1.5.1.8.3 Fields

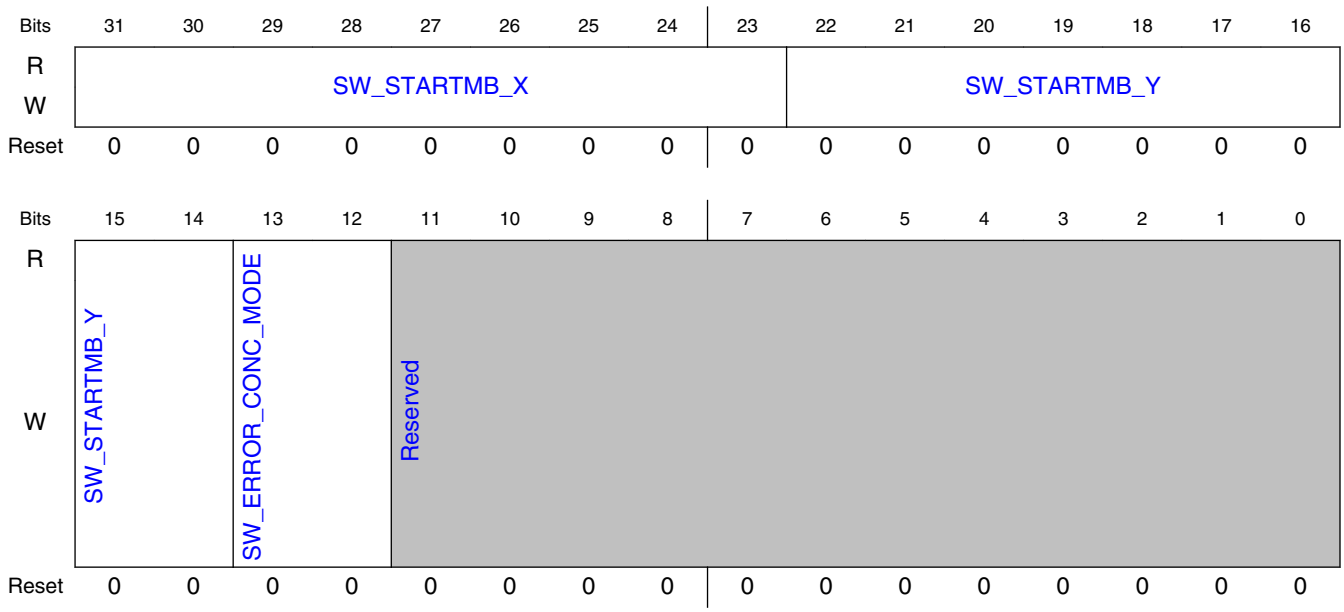
Field	Function
31-2 SW_DIR_MV_BASE	Direct mode motion vector write/read base address. For H264 this is used only for direct mode motion vector write base. VP8: Motion vectors are written for error concealment purposes if sw_write_mvs is high. In error concealment mode motion vectors are read from this base address
1-0 —	Reserved.

14.1.5.1.9 Error concealment register (SWREG48)

14.1.5.1.9.1 Offset

Register	Offset
SWREG48	C0h

14.1.5.1.9.2 Diagram



14.1.5.1.9.3 Fields

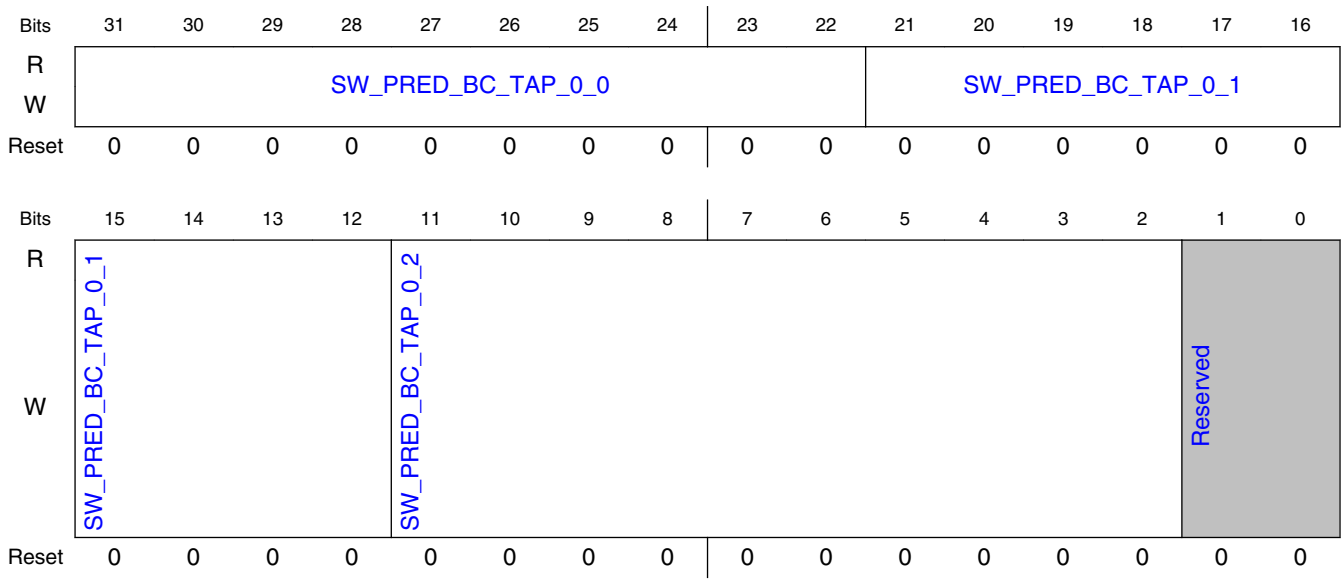
Field	Function
31-23 SW_STARTMB_X	Start MB from SW for X dimension. Used in error concealment case as HW return value if HW founds an error or in HW init mb for error concealment if SW enables error concealment
22-14 SW_STARTMB_Y	Start MB from SW for Y dimension. Used in error concealment case as HW return value if HW founds an error or in HW init mb for error concealment if SW enables error concealment
13-12 SW_ERROR_CONC_MODE	Error concealment mode: 00b - disabled (normal decoding mode) 01b - enabled for direct mode MV usage starting from MB defined by sw_startmb_x, sw_startmb_y
11-0 —	Reserved.

14.1.5.1.10 Prediction filter tap register for H264 (SWREG49)

14.1.5.1.10.1 Offset

Register	Offset
SWREG49	C4h

14.1.5.1.10.2 Diagram



14.1.5.1.10.3 Fields

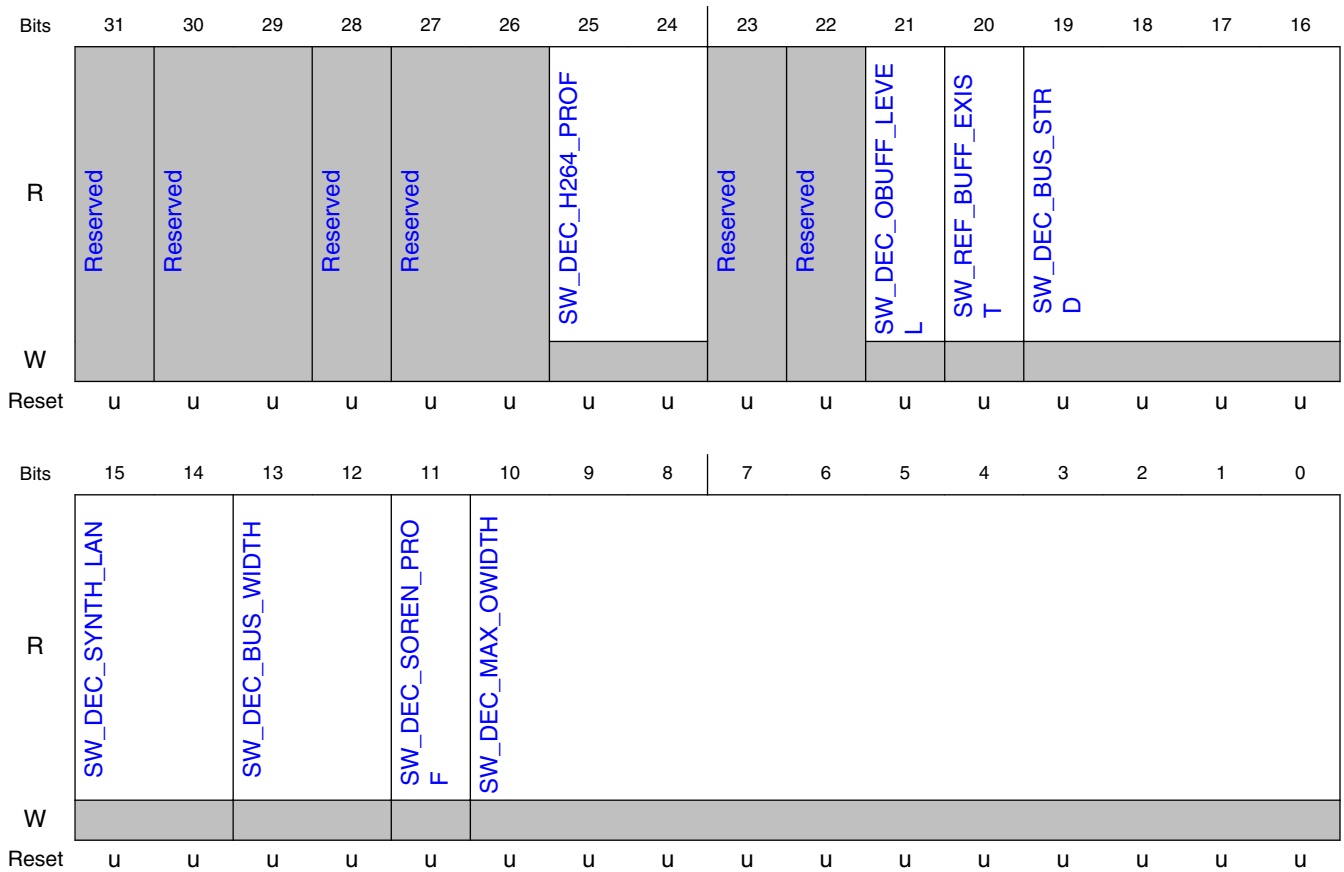
Field	Function
31-22 SW_PRED_BC_TAP_0_0	Prediction filter set 0, tap 0
21-12 SW_PRED_BC_TAP_0_1	Prediction filter set 0, tap 1
11-2 SW_PRED_BC_TAP_0_2	Prediction filter set 0, tap 2
1-0 —	Reserved.

14.1.5.1.11 Synthesis configuration register decoder 0 (SWREG50)

14.1.5.1.11.1 Offset

Register	Offset
SWREG50	C8h

14.1.5.1.11.2 Diagram



14.1.5.1.11.3 Fields

Field	Function
31 —	Reserved.
30-29 —	Reserved.
28 —	Reserved.
27-26 —	Reserved.
25-24 SW_DEC_H264_PROF	Decoding format support, H.264 00b - not supported 01b - supported up to baseline profile 10b - supported up to high profile labeled stream with restricted high profile tools (Tools that are used in Hantro 7280, 8270 encoder) 11b - supported up to high profile

Table continues on the next page...

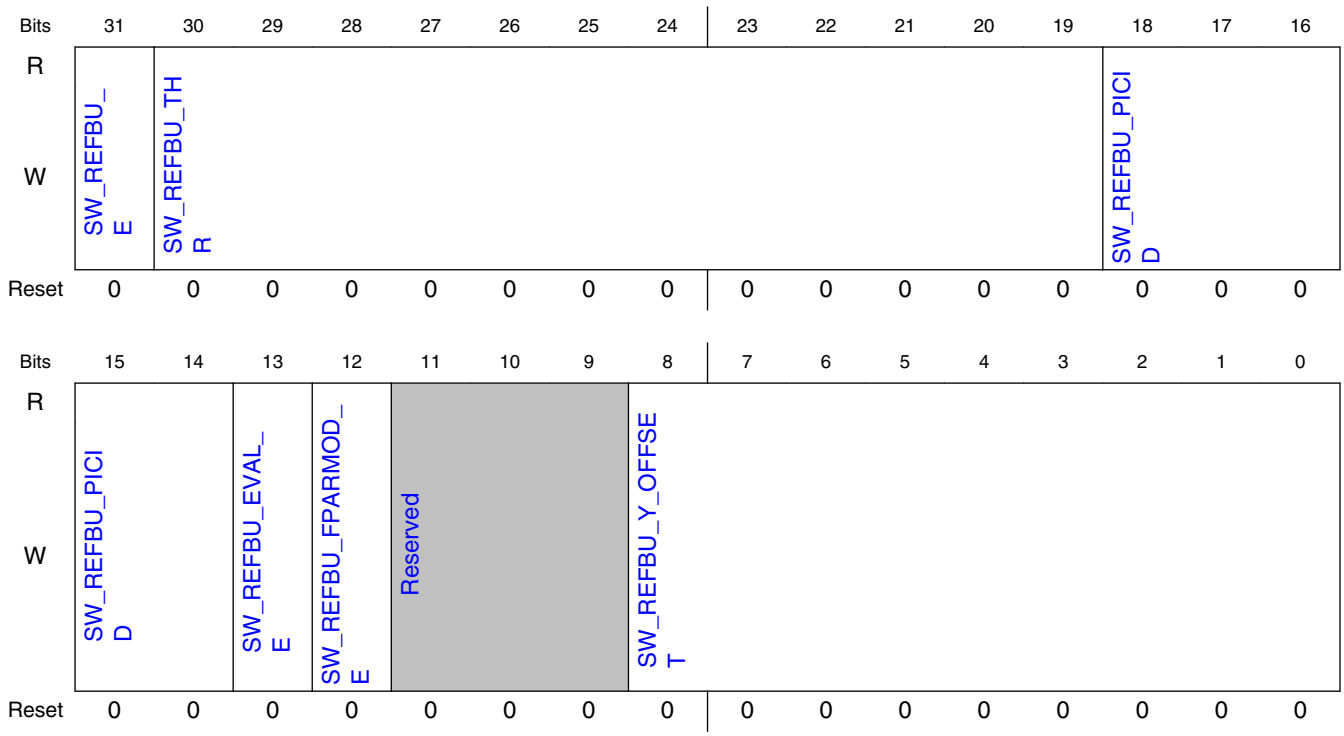
Field	Function
23 —	Reserved.
22 —	Reserved.
21 SW_DEC_OBUFF_LEVEL	Decoder output buffer level: 0b - 1 MB buffering is used 1b - 4 MB buffering is used
20 SW_REF_BUFF_EXIST	Reference picture buffer usage: 0b - not supported 1b - reference buffer is used
19-16 SW_DEC_BUS_STRD	Connected to standard bus: 0000b - error 0001b - AHB master, AHB slave 0010b - OCP master, OCP slave 0011b - AXI master, AXI slave 0100b - AXI master, APB slave 0101b - AXI master, AHB slave
15-14 SW_DEC_SYNTH_LAN	00b - error 01b - vhdl 10b - verilog
13-12 SW_DEC_BUS_WIDTH	00b - error 01b - 32 bit bus 10b - 64 bit bus 11b - 128 bit bus
11 SW_DEC_SOREN_PROF	Decoding format support, Sorenson 0b - not supported 1b - supported
10-0 SW_DEC_MAX_OWIDTH	Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels.

14.1.5.1.12 Reference picture buffer control register (SWREG51)

14.1.5.1.12.1 Offset

Register	Offset
SWREG51	CCh

14.1.5.1.12.2 Diagram



14.1.5.1.12.3 Fields

Field	Function
31 SW_REFBU_E	Refer picture buffer enable: 0b - refer picture buffer disabled 1b - refer picture buffer enabled. Valid if picture size is QVGA or more.
30-19 SW_REFBU_THR	Reference buffer disable threshold value (cache miss amount). Used to buffer shut down (if more misses than allowed)
18-14 SW_REFBU_PICID	The used reference picture ID for reference buffer usage
13 SW_REFBU_EVAL_E	Enable for HW internal reference ID calculation. If given threshold level is reached by any picture_id after first MB row, that picture_id is used for reference buffer fill for rest of the picture
12 SW_REFBU_FPARMOD_E	Field parity mode enable. Used in refbufferd evaluation mode. 0b - use the result field of the evaluation 1b - use the parity mode field
11-9 —	Reserved.
8-0 SW_REFBU_Y_OFFSET	Y offset for refbufferd. This coordinate is used to compensate the global motion of the video for better buffer hit rate

14.1.5.1.13 Reference picture buffer information register 1 (SWREG52)

14.1.5.1.13.1 Offset

Register	Offset
SWREG52	D0h

14.1.5.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFBU_HIT_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU_INTRA_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.13.3 Fields

Field	Function
31-16 SW_REFBU_HIT_SUM	The sum of the refbufferd hits of the picture. Determined for each 8x8 luminance partition of the picture. The proceeding of the HW calculation can be read during HW decoding.
15-0 SW_REFBU_INTRA_SUM	The sum of the luminance 8x8 intra partitions of the picture. The proceeding of the HW calculation can be read during HW decoding.

14.1.5.1.14 Reference picture buffer information register 2 (SWREG53)

14.1.5.1.14.1 Offset

Register	Offset
SWREG53	D4h

14.1.5.1.14.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved										SW_REFBU_Y_MV_SUM					
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU_Y_MV_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.14.3 Fields

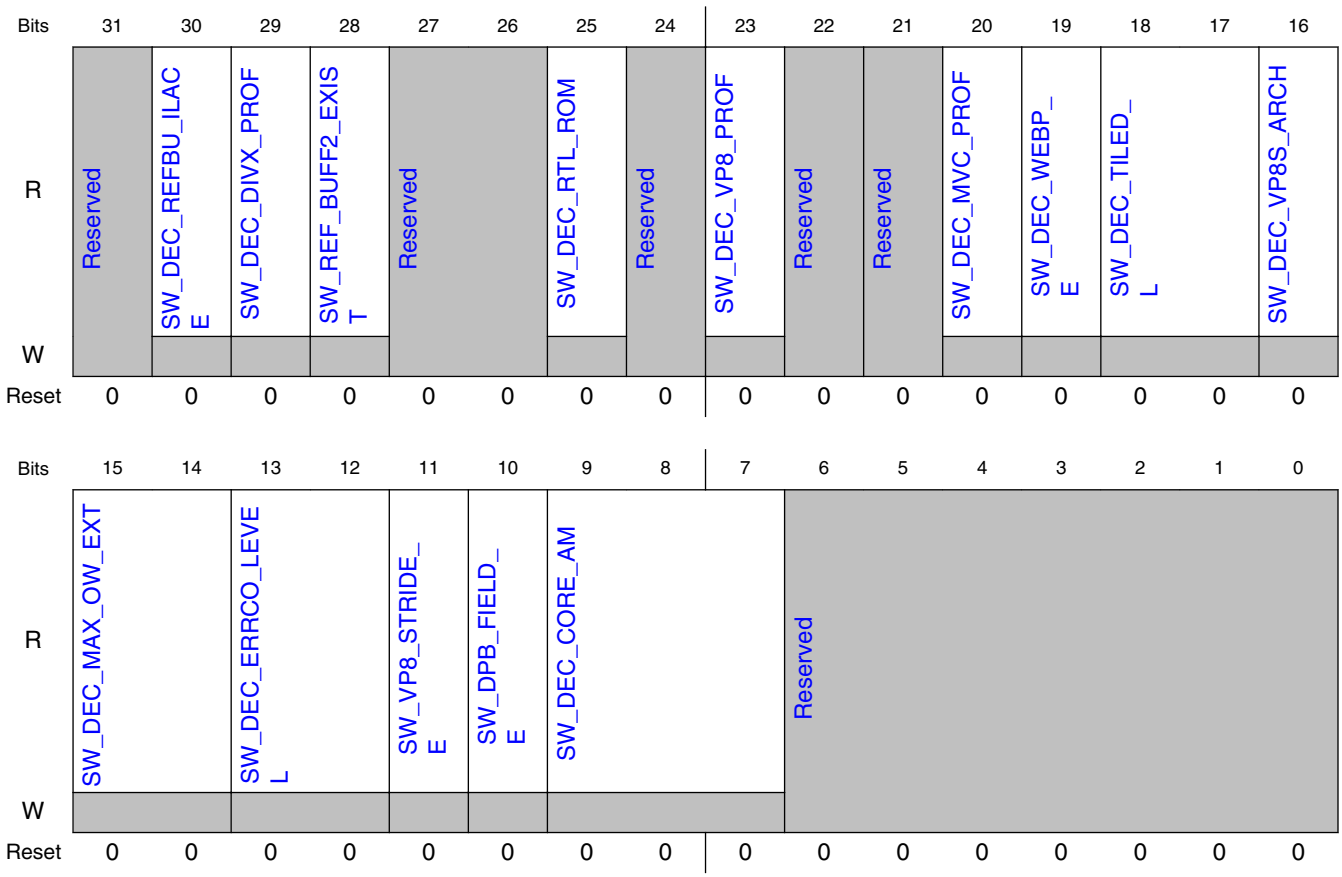
Field	Function
31-22 —	Reserved.
21-0 SW_REFBU_Y_MV_SUM	The sum of the decoded motion vector y-components of the picture. The first luminance motion vector of each MB is used in calculation. Other motion vectors of the MB are discarded. Each motion vector is saturated between -256 - 255 before calculation. The proceeding of the HW calculation can be read during HW decoding.

14.1.5.1.15 Synthesis configuration register decoder 1 (SWREG54)

14.1.5.1.15.1 Offset

Register	Offset
SWREG54	D8h

14.1.5.1.15.2 Diagram



14.1.5.1.15.3 Fields

Field	Function
31 —	Reserved.
30 SW_DEC_REFBU_ILACE	Refbufferd support for interlaced content: 0b - not supported 1b - supported
29 SW_DEC_DIVX_PROF	DIVX Support: 0b - not supported 1b - supported
28 SW_REF_BUFF2_EXIST	Reference picture buffer 2 usage: 0b - not supported 1b - reference buffer 2 is used
27-26 —	Reserved.
25	ROM implementation type (If design includes ROMs) 0b - ROMs are implemented from actual ROM units

Table continues on the next page...

Field	Function
SW_DEC_RTL_ROM	1b - ROMs are implemented from RTL
24 —	Reserved.
23 SW_DEC_VP8_PROF	Decoding format support, VP8 0b - not supported 1b - supported
22 —	Reserved.
21 —	Reserved.
20 SW_DEC_MVC_PROF	Decoding format support, MVC 0b - not supported 1b - supported
19 SW_DEC_WEBP_E	Decoding format support, Web-p 0b - not supported bigger than 1080p resolution 1b - supported upto 16kx16k pixel resolution (defined max)
18-17 SW_DEC_TILED_L	Tiled mode support level 00b - not supported 01b - supported with 8x4 tile size for progressive content 10b - supported with 8x4 tile size for progressive/ilaced content
16 SW_DEC_VP8S_ARCH	VP8 Architecture type (for prediction) 0b - Same prediction architecture as for other decoding formats 1b - Dedicated small architecture for VP8 (refbuffer cannot be used either)
15-14 SW_DEC_MAX_OW_EXT	Max configured decoder video resolution that can be decoded. This is the MSB part of the configuration signal
13-12 SW_DEC_ERRCO_LEVEL	Decoder error concealment support level: 00b - Error concealment not supported (only error detection) 01b - VP8 direct mode motion vector error concealment supported
11 SW_VP8_STRIDE_E	Decoder output stride support for VP8. Separate base addresses for Y/C data and possibility to set scanline bigger than picture width: 0b - not supported, Y and C tables attached. 1b - supported, Y and C tables can be set freely.
10 SW_DPB_FIELD_E	DPB field separate mode support for ilaced content: 0b - Not supported. For ilaced content, DPB is ilaced frame order. 1b - Supported. For ilaced content, DPB can consist of ilaced frames or separate fields (TOP/BOT).
9-7 SW_DEC_CORE_AM	Decoder core amount. If other than 0, the multicore can be used. Each individual cores can be identified from corresponding core ID register: 000b - single core decoder 001b - dual core decoder 010b - 3 core decoder 011b - 4 core decoder 100b - 5 core decoder 101b - 6 core decoder 110b - 7 core decoder 111b - 8 core decoder

Table continues on the next page...

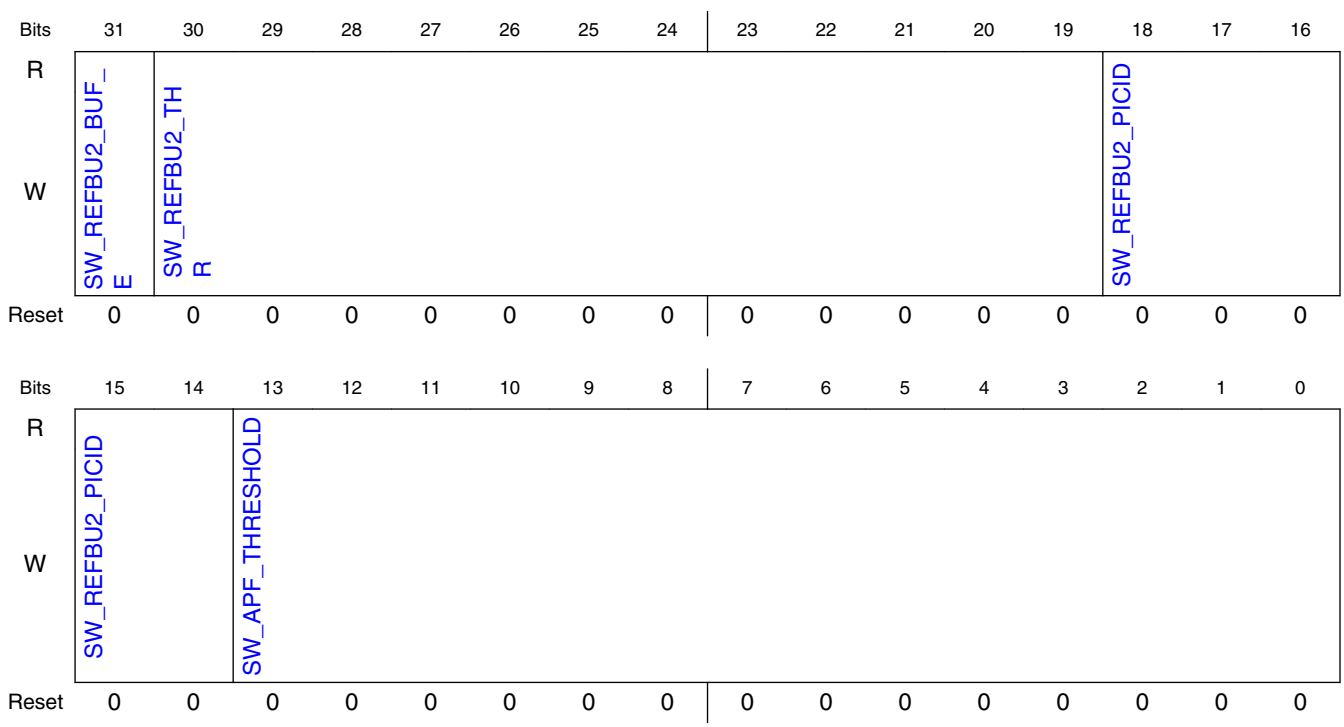
Field	Function
6-0 —	Reserved.

14.1.5.1.16 Reference picture buffer 2 / Advanced prefetch control register (SWREG55)

14.1.5.1.16.1 Offset

Register	Offset
SWREG55	DCh

14.1.5.1.16.2 Diagram



14.1.5.1.16.3 Fields

Field	Function
31 SW_REFBU2_BUF_E	Refer picture buffer 2 enable: 0b - refer picture buffer disabled

Table continues on the next page...

Field	Function
	1b - refer picture buffer enabled. Valid if picture size is QVGA or more (can be turned of by HW if threshold value reached).
30-19 SW_REFBU2_THR	Reference buffer disable threshold value (buffer miss amount). Used to buffer shut down (if more misses than allowed)
18-14 SW_REFBU2_PICID	The used reference picture ID for reference buffer usage
13-0 SW_APF_THRESHOLD	G1 decoder and later :Advanced prefetch threshold value. If current MB exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only

14.1.5.1.17 Reference buffer information register 3 (SWREG56)

14.1.5.1.17.1 Offset

Register	Offset
SWREG56	E0h

14.1.5.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_REFBU_TOP_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_REFBU_BOT_SUM															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.17.3 Fields

Field	Function
31-16 SW_REFBU_TOP_SUM	The sum of the top partitions of the picture
15-0	The sum of the bottom partitions of the picture

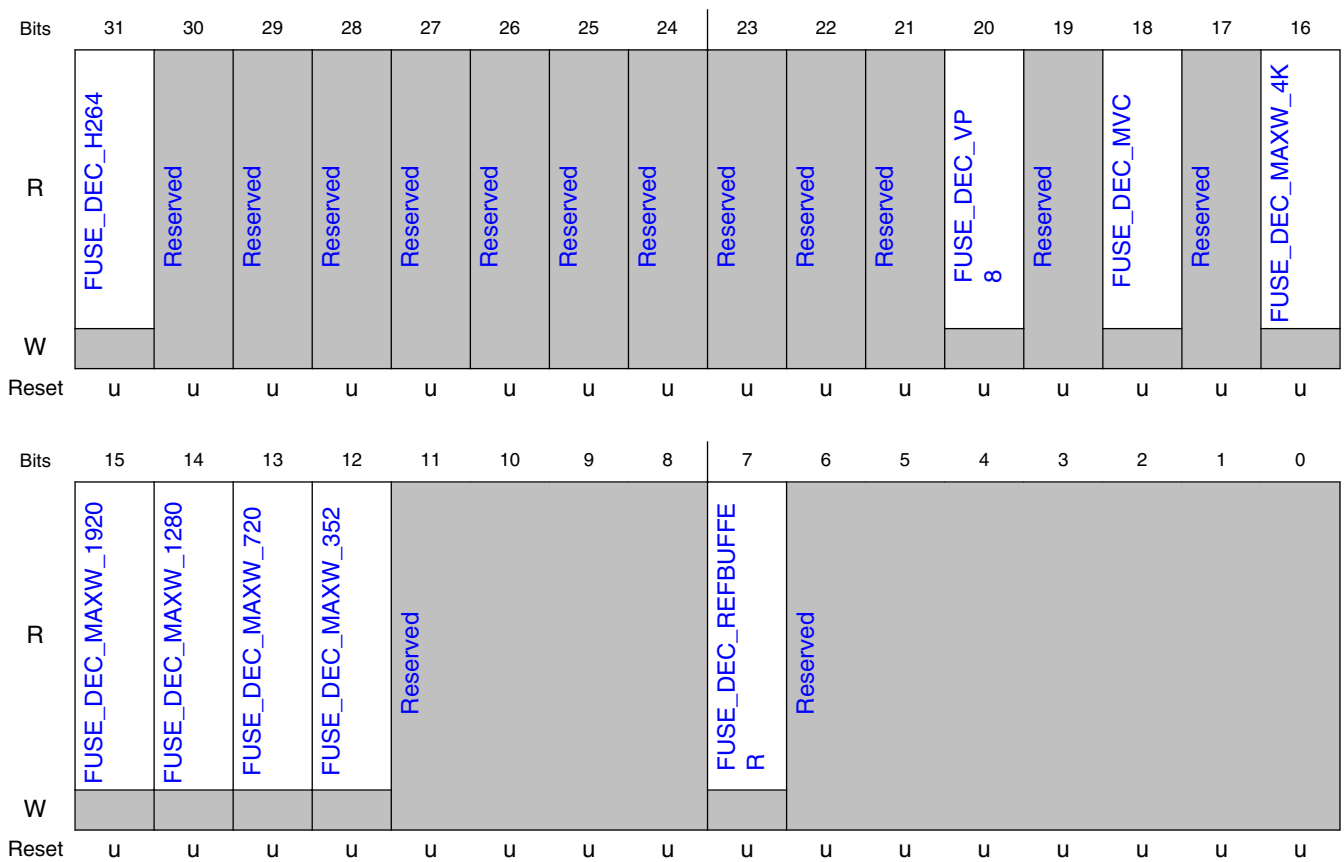
Field	Function
SW_REFBU_BOT_SUM	

14.1.5.1.18 Decoder fuse register (SWREG57)

14.1.5.1.18.1 Offset

Register	Offset
SWREG57	E4h

14.1.5.1.18.2 Diagram



14.1.5.1.18.3 Fields

Field	Function
31 FUSE_DEC_H264	1 = H.264 enabled
30 —	Reserved.
29 —	Reserved.
28 —	Reserved.
27 —	Reserved.
26 —	Reserved.
25 —	Reserved.
24 —	Reserved.
23 —	Reserved.
22 —	Reserved.
21 —	Reserved.
20 FUSE_DEC_VP8	1 = VP8 enabled
19 —	Reserved.
18 FUSE_DEC_MVC	1 = MVC enabled (requires also H264 to be enabled)
17 —	Reserved.
16 FUSE_DEC_MAXW_4K	1 = Max video width up to 4096 pixels enabled. Priority coded with priority 1.
15 FUSE_DEC_MAXW_1920	1 = Max video width up to 1920 pixels enabled. Priority coded with priority 2.

Table continues on the next page...

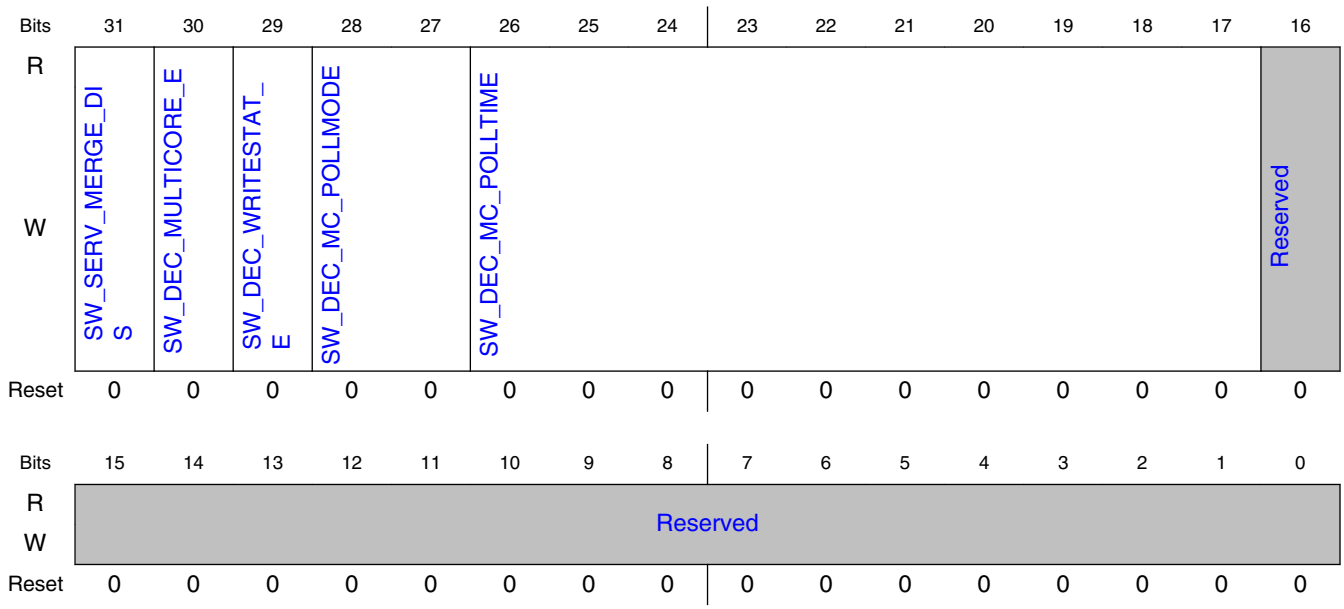
Field	Function
14 FUSE_DEC_MAXW_1280	1 = Max video width up to 1280 pixels enabled. Priority coded with priority 3.
13 FUSE_DEC_MAXW_720	1 = Max video width up to 720 pixels enabled. Priority coded with priority 4.
12 FUSE_DEC_MAXW_352	1 = Max video width up to 352 pixels enabled. Priority coded with priority 5.
11-8 —	Reserved.
7 FUSE_DEC_REFBUFFER	1 = reference buffer used
6-0 —	Reserved.

14.1.5.1.19 Device configuration register decoder 2 + Multi core control register (SWREG58)

14.1.5.1.19.1 Offset

Register	Offset
SWREG58	E8h

14.1.5.1.19.2 Diagram



14.1.5.1.19.3 Fields

Field	Function
31 SW_SERV_MERGE_DIS	Decoder service merge disable: 0b - HW merges simultaneous sub-block requests internally if they are same type (read or write). 1b - decoder serves one sub-block per service and merging is disabled.
30 SW_DEC_MULTICORE_E	Decoder multi core enable: 0b - Multi core disabled or only one core exists in design. 1b - Multi core enable. Each reference picture status must be verified from external memory status field before usage. 128 bits status word exists after each reference picture and include picture proceeding coordinates Y and X.
29 SW_DEC_WRITESTAT_E	Decoder write statusword enable. Must be high if multi core decoding enabled. HW writes output picture data proceeding to external memory after picture data (and after H264 direct mode MVS if they exist)
28-27 SW_DEC_MC_POLLMODE	Decoder multicore status reading mode: 00b - HW internal status polling mechanism is used. Status of reference picture is read only when required coordinate for the reference picture is not big enough. If the status is still not big enough after reading it the HW waits N clock cycles per pixel from the coordinate difference. The N is defined by the sw_dec_mc_polltime (range 0...4). 01b - Dummy status polling mechanism is used for all reference pictures. HW reads status of all reference pictures at frequency defined by sw_dec_mc_polltime.
26-17 SW_DEC_MC_POLLTIME	sw_dec_mc_polltime definition depends on sw_dec_mc_mode. if mode is 0 (HW internal mechanism) the sw_dec_mc_polltime: • 0 = 1/4 cycles per pixel • 1 = wait 1/2 cycles per pixel • 2 = wait one cycle per pixel • 3 = wait 2 cycles per pixel • 4=wait 4 cycles per pixel

Table continues on the next page...

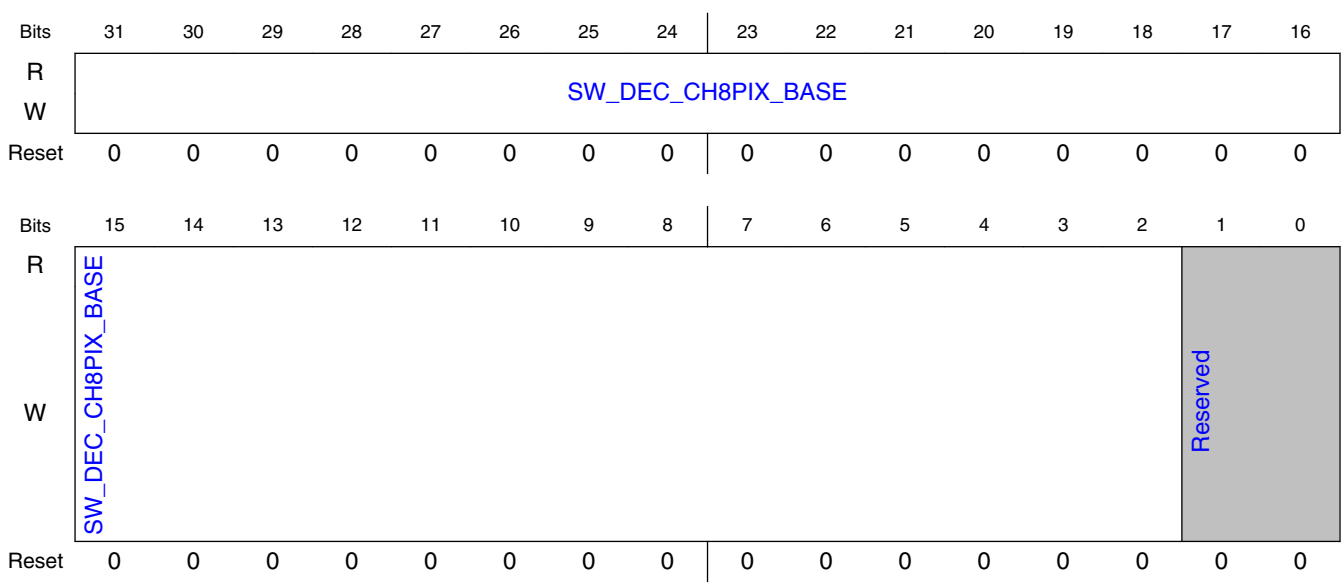
Field	Function
	For dummy polling mechanism the sw_dec_mc_polltime defines the amount cycles between status reads x1024: • 0 = wait 1024 cycles between status reading • 1 = wait 2048 cycles between status reading • 2 = wait 3072 cycles between status reading • N = wait (N+1)x1024 cycles
16-0 —	Reserved.

14.1.5.1.20 H264 Chrominance 8 pixel interleaved data base (SWREG59)

14.1.5.1.20.1 Offset

Register	Offset
SWREG59	ECh

14.1.5.1.20.2 Diagram



14.1.5.1.20.3 Fields

Field	Function
31-2	Base address for additional chrominance data format where chrominance is interleaved in group of 8 pixels. The usage is enabled by sw_ch_8pix_ileav_e

Table continues on the next page...

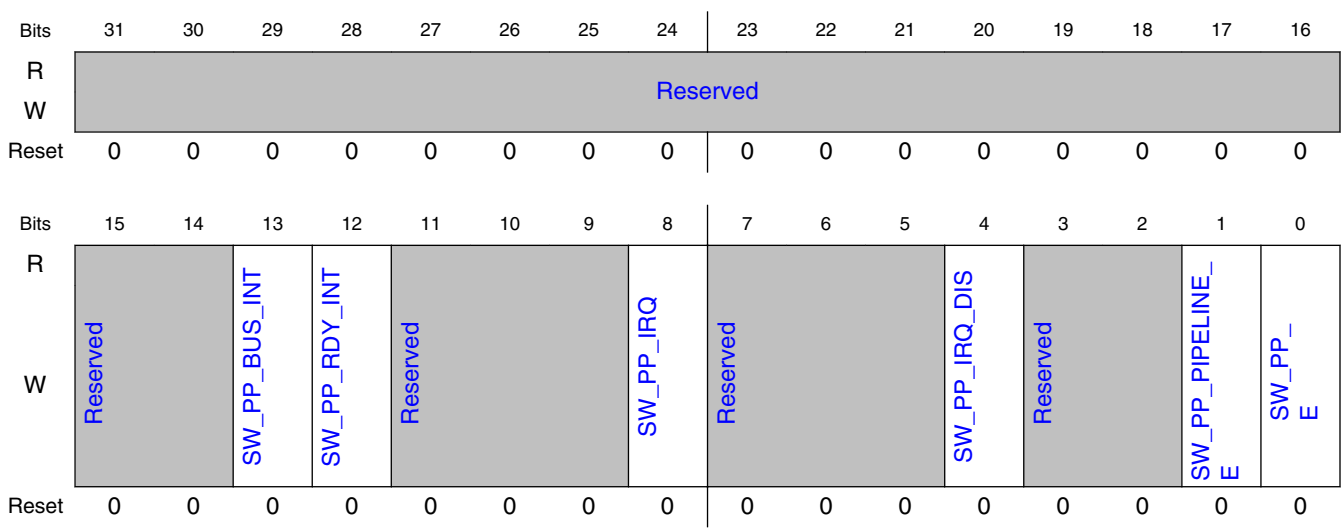
Field	Function
SW_DEC_CH8PIX_BASE	
1-0 —	Reserved.

14.1.5.1.21 Interrupt register post-processor (SWREG60)

14.1.5.1.21.1 Offset

Register	Offset
SWREG60	F0h

14.1.5.1.21.2 Diagram



14.1.5.1.21.3 Fields

Field	Function
31-14 —	Reserved.
13 SW_PP_BUS_INT	Interrupt status bit bus. Error response from bus. In pipeline mode this bit is not used

Table continues on the next page...

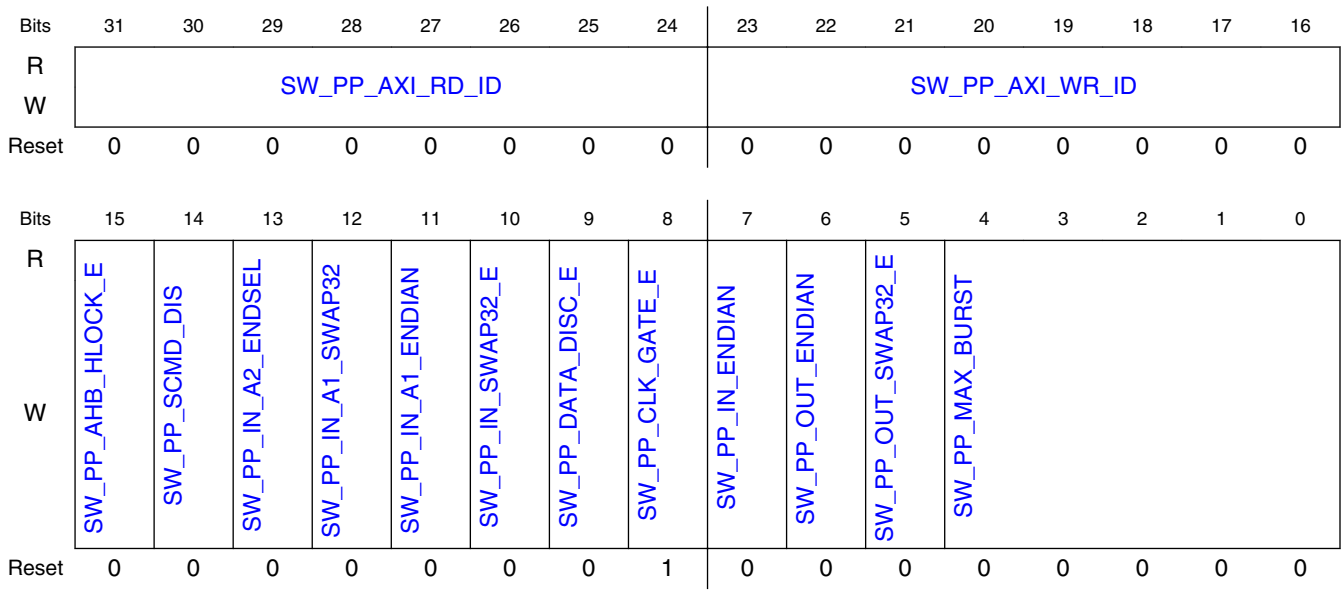
Field	Function
12 SW_PP_RDY_INT	Interrupt status bit pp. When this bit is high post processor has processed a picture in external mode. In pipeline mode this bit is not used.
11-9 —	Reserved.
8 SW_PP_IRQ	Post-processor IRQ. SW will reset this after interrupt is handled. HINTpp is not used for pp if IRQ disable pp is high (sw_pp_irq_n_e = 1). In pipeline mode this bit is not used
7-5 —	Reserved.
4 SW_PP_IRQ_DIS	Post-processor IRQ disable. When high, there are no interrupts from HW concerning post processing. Polling must be used to see the interrupt
3-2 —	Reserved.
1 SW_PP_PIPELINE_E	Decoder – post-processing pipeline enable: 0b - Post-processing is processing different picture than decoder or is disabled 1b - Post-processing is performed in pipeline with decoder
0 SW_PP_E	External mode post-processing enable. This bit will start the post-processing operation. Not to be used if PP is in pipeline with decoder (sw_pp_pipeline_e = 1). HW will reset this when picture is post-processed.

14.1.5.1.22 Device configuration register post-processor (SWREG61)

14.1.5.1.22.1 Offset

Register	Offset
SWREG61	F4h

14.1.5.1.22.2 Diagram



14.1.5.1.22.3 Fields

Field	Function
31-24 SW_PP_AXI_RD_ID	Read ID used for AXI PP read services (if connected to AXI)
23-16 SW_PP_AXI_WR_ID	Write ID used for AXI PP write services (if connected to AXI)
15 SW_PP_AHB_HLOCK_E	AHB master HLOCK enable. When high the service is locked to pp as long as it needs the bus (whenever pp requests the bus it will be granted)
14 SW_PP_SCMD_DIS	9170 decoder: AXI Single Command Multiple Data disable. 9170 axi wrapper supports this mode by default (where only the first addresses of the burst are given from address generator). This bit is used to disable the feature (possible SW workaround if something is not working correctly)
13 SW_PP_IN_A2_ENDSEL	Endian/swap select for Alpha blend input source 2: 0b - Use PP in endian/swap definitions (sw_pp_in_endian, sw_pp_in_swap) 1b - Use Ablend source 1 endian/swap definitions (sw_pp_in_a1_endian, sw_pp_in_a1_swap)
12 SW_PP_IN_A1_SWAP32	Alpha blend source 1 input 32bit data swap (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
11 SW_PP_IN_A1_ENDIAN	Alpha blend source 1 input data byte endian mode. 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
10	PP input 32bit data swap (may be used for 64 bit environment):

Table continues on the next page...

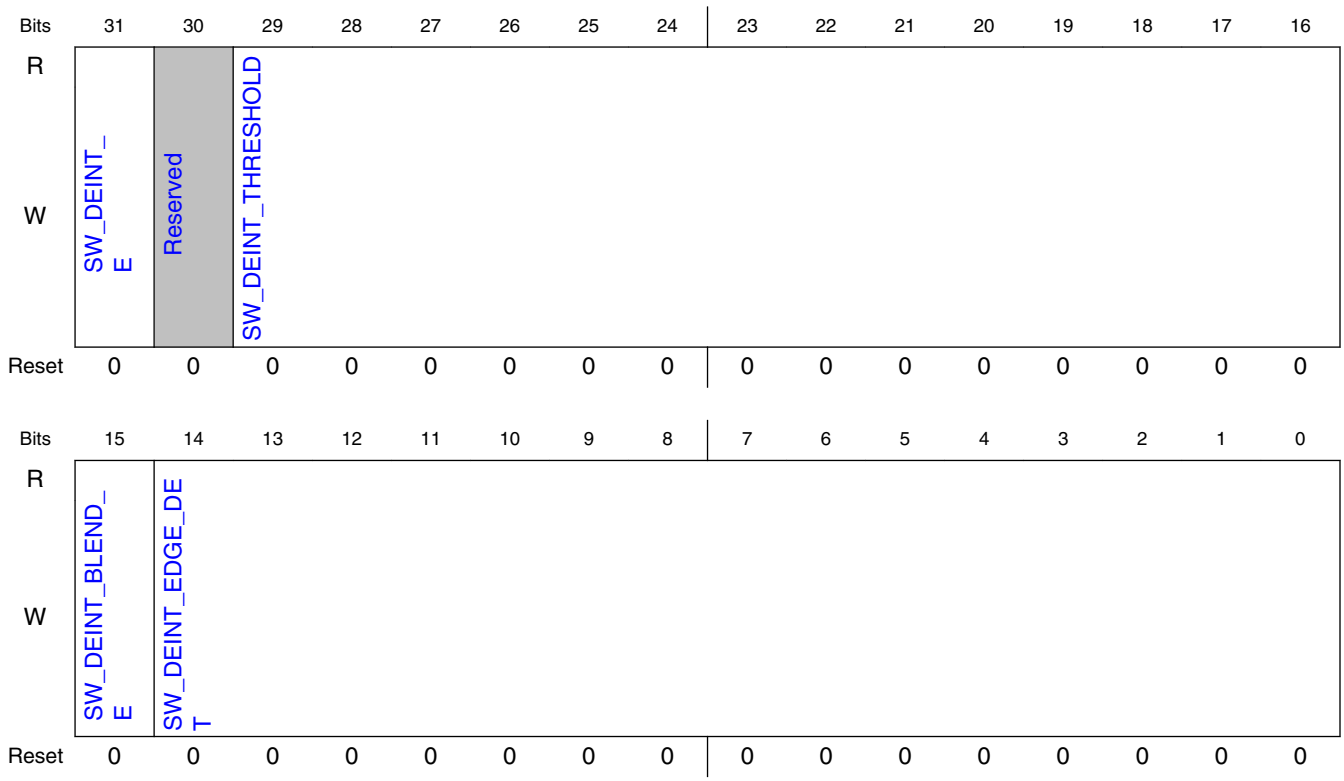
Field	Function
SW_PP_IN_SWAP32_E	0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order(also little endian should be enabled))
9 SW_PP_DATA_DISC_E	PP data discard enable. Precise burst lengths are used with reading services. Extra data is discarded internally. Note. If AHB maxburst 17 is used data discard cannot be enabled (causes conflict)
8 SW_PP_CLK_GATE_E	PP dynamic clock gating enable. **NOTE:** Clock gating value can be changed only when PP is not enabled. 0b - Clock is running for all PP structures 1b - Clock is gated from PP structures that are not used
7 SW_PP_IN_ENDIAN	PP input picture byte endian mode. Used only if PP is in standalone mode. If PP is running pipelined with the decoder, this bit has no effect. 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
6 SW_PP_OUT_ENDIAN	PP output picture endian mode for YCbCr data or for any data if config value SW_PP_OEN_VERSION = 1. **NOTE:** For SW_PP_OEN_VERSION=0, 16 bit RGB data, this bit works as pixel swapping bit. For 32 bit RGB, this bit has no meaning. 0b - Big endian (0-1-2-3 order) 1b - Little endian (3-2-1-0 order)
5 SW_PP_OUT_SWAP32_E	PP output data word swap (may be used for 64 bit environment): 0b - no swapping of 32 bit words 1b - 32 bit data words are swapped (needed in 64 bit environment to achieve 7-6-5-4-3-2-1-0 byte order (also little endian should be enabled))
4-0 SW_PP_MAX_BURST	Maximum burst length for PP bus transactions. Valid values: AHB: 1, 4, 8 , 16 and 17. Other values will result in INCR type (undefined length) transfers. 17 = fixed bursts are used when possible. INCR bursts for others. OCP: 16-31 AXI: 1-16

14.1.5.1.23 Deinterlace control register (SWREG62)

14.1.5.1.23.1 Offset

Register	Offset
SWREG62	F8h

14.1.5.1.23.2 Diagram



14.1.5.1.23.3 Fields

Field	Function
31 SW_DEINT_E	De-interlace enable. Input data is in interlaced format and deinterlacing needs to be performed
30 —	Reserved.
29-16 SW_DEINT_THRESHOLD	Threshold value used in deinterlacing
15 SW_DEINT_BLEND_E	Blend enable for de-interlacing
14-0 SW_DEINT_EDGE_DET	Edge detect value used for deinterlacing

14.1.5.1.24 Base address for reading post-processing input picture luminance (top field/frame) (SWREG63)

14.1.5.1.24.1 Offset

Register	Offset
SWREG63	FCh

14.1.5.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PP_IN_LU_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PP_IN_LU_BASE														Reserved	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.24.3 Fields

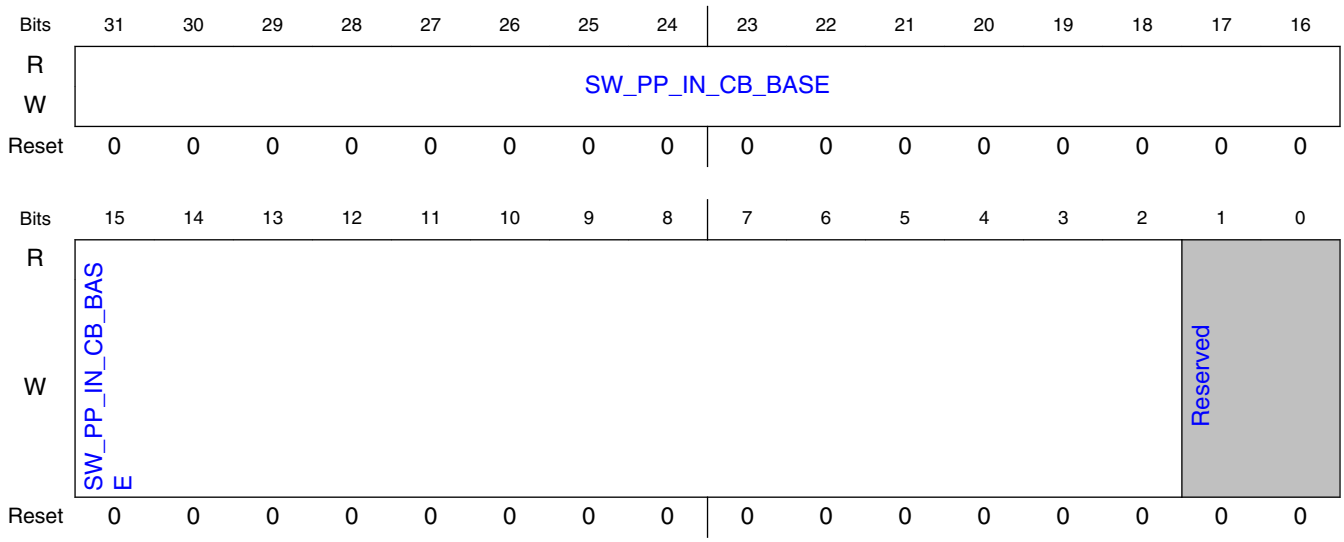
Field	Function
31-2 SW_PP_IN_LU_BASE	Base address for post-processing input luminance picture. If PP input picture is fetched from fields this base address is used to point to top field of the picture. Used in external mode only.
1-0 —	Reserved.

14.1.5.1.25 Base address for reading post-processing input picture Cb/Ch (top field/frame) (SWREG64)

14.1.5.1.25.1 Offset

Register	Offset
SWREG64	100h

14.1.5.1.25.2 Diagram



14.1.5.1.25.3 Fields

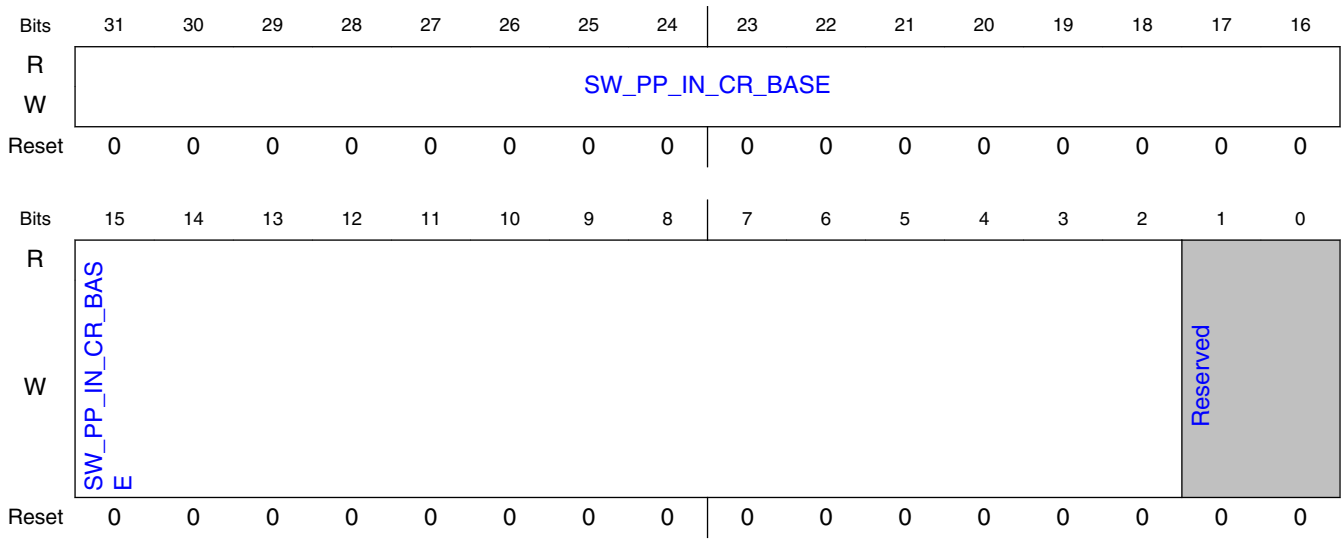
Field	Function
31-2 SW_PP_IN_CB_BASE	Base address for post-processing input Cb picture or for both chrominance pictures (if chrominances interleaved). If PP input picture is fetched from fields this base address is used to point to top field of the picture. Used in external mode only
1-0 —	Reserved.

14.1.5.1.26 Base address for reading post-processing input picture Cr (SWREG65)

14.1.5.1.26.1 Offset

Register	Offset
SWREG65	104h

14.1.5.1.26.2 Diagram



14.1.5.1.26.3 Fields

Field	Function
31-2 SW_PP_IN_CR_BASE	Base address for post-processing input cr picture. Used in external mode only
1-0 —	Reserved.

14.1.5.1.27 Base address for writing post-processed picture luminance/RGB (SWREG66)

14.1.5.1.27.1 Offset

Register	Offset
SWREG66	108h

14.1.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PP_OUT_LU_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PP_OUT_LU_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.27.3 Fields

Field	Function
31-0 SW_PP_OUT_LU_BASE	Base address for post-processing output picture (luminance/YUYV/RGB). NOTE: Bits 2:0 are used to adjust the post-processor output to start from zertain byte (1:0 for 32 bit bus). These bits can be other than zero only if Pixel Accurate PP output configuration is enabled

14.1.5.1.28 Base address for writing post-processed picture Ch (SWREG67)

14.1.5.1.28.1 Offset

Register	Offset
SWREG67	10Ch

14.1.5.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PP_OUT_CH_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PP_OUT_CH_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.28.3 Fields

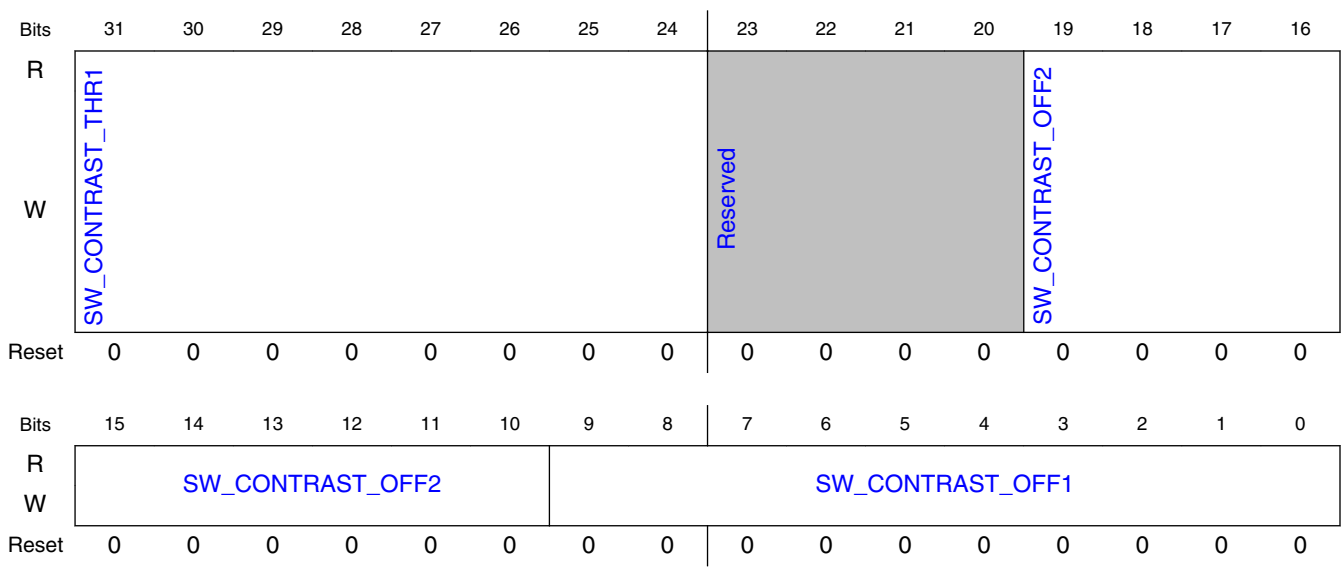
Field	Function
31-0 SW_PP_OUT_CH_BASE	Base address for post-processing output chrominance picture (interleaved chrominance). NOTE: Bits 2:0 are used to adjust the post-processor output to start from zertain byte (1:0 for 32 bit bus). These bits can be other than zero only if Pixel Accurate PP output configuration is enabled

14.1.5.1.29 Register for contrast adjusting (SWREG68)

14.1.5.1.29.1 Offset

Register	Offset
SWREG68	110h

14.1.5.1.29.2 Diagram



14.1.5.1.29.3 Fields

Field	Function
31-24 SW_CONTRAST_THR1	Threshold value 1, used with contrast adjusting

Table continues on the next page...

Field	Function
23-20 —	Reserved.
19-10 SW_CONTRAST_OFF2	Offset value 2, used with contrast adjusting
9-0 SW_CONTRAST_OFF1	Offset value 1, used with contrast adjusting

14.1.5.1.30 Register for colour conversion and contrast adjusting/YUYV 422 channel orders (SWREG69)

14.1.5.1.30.1 Offset

Register	Offset
SWREG69	114h

14.1.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PP_IN_START_CH	SW_PP_IN_CR_FIRST	SW_PP_OUT_START_CH	SW_PP_OUT_CR_FIRST	SW_COLOR_COEFFA2										SW_COLOR_COEFFA1	
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_COLOR_COEFFA1								SW_CONTRAST_THR2							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.30.3 Fields

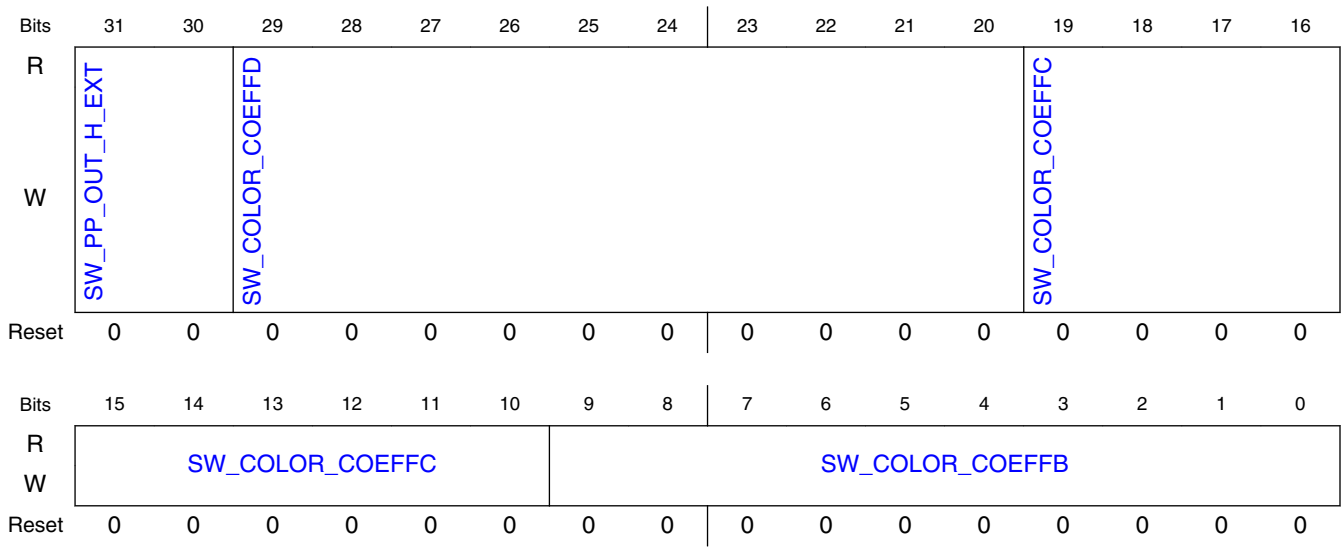
Field	Function
31 SW_PP_IN_START_CH	For YUYV 422 input format. Enable for start_with_chrominance. 0b - the order is Y0CbY0Cr or Y0CrY0Cb 1b - the order is CbY0CrY0 or CrY0CbY0
30 SW_PP_IN_CR_FIRST	For YUYV 422 input format. Enable for Cr first (before Cb). 0b - the order is Y0CbY0Cr or CbY0CrY0 (if 420 semiplanar chrominance: CbCrCbCr) 1b - the order is Y0CrY0Cb or CrY0CbY0 (if 420 semiplanar chrominance: CrCbCrCb)
29 SW_PP_OUT_START_CH	For YUYV 422 output format. Enable for start_with_chrominance. 0b - the order is Y0CbY0Cr or Y0CrY0Cb 1b - the order is CbY0CrY0 or CrY0CbY0
28 SW_PP_OUT_CR_FIRST	For YUYV 422 output format. Enable for Cr first (before Cb). 0b - the order is Y0CbY0Cr or CbY0CrY0 1b - the order is Y0CrY0Cb or CrY0CbY0
27-18 SW_COLOR_COEFFA2	Coefficient a2, used with Y pixel to calculate all color components
17-8 SW_COLOR_COEFFA1	Coefficient a1, used with Y pixel to calculate all color components
7-0 SW_CONTRAST_THR2	Threshold value 2, used with contrast adjusting

14.1.5.1.31 Register for colour conversion 0 (SWREG70)

14.1.5.1.31.1 Offset

Register	Offset
SWREG70	118h

14.1.5.1.31.2 Diagram



14.1.5.1.31.3 Fields

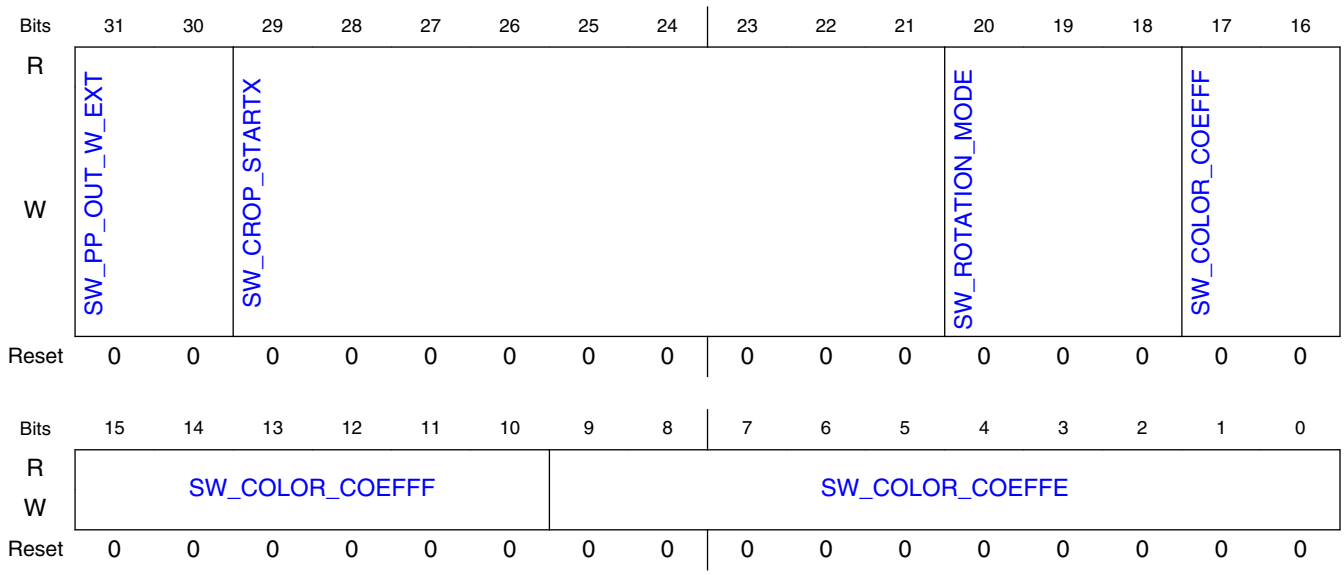
Field	Function
31-30 SW_PP_OUT_H_EXT	Extended output height for 4k resolution
29-20 SW_COLOR_COEFFD	Coefficient d, used with Cb to calculate green component value
19-10 SW_COLOR_COEFFC	Coefficient c, used with Cr to calculate green component value
9-0 SW_COLOR_COEFFB	Coefficient b, used with Cr to calculate red component value

14.1.5.1.32 Register for colour conversion 1 + rotation mode (SWREG71)

14.1.5.1.32.1 Offset

Register	Offset
SWREG71	11Ch

14.1.5.1.32.2 Diagram



14.1.5.1.32.3 Fields

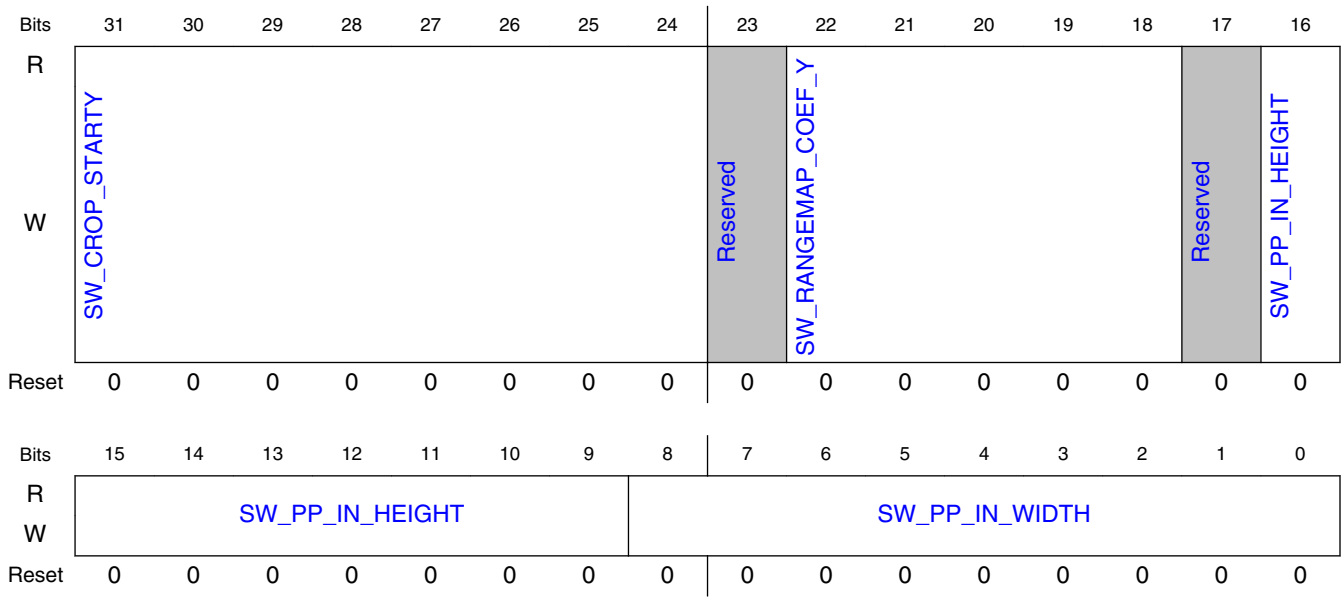
Field	Function
31-30 SW_PP_OUT_W_EXT	Extended output width for 4k resolution
29-21 SW_CROP_STARTX	Start coordinate x for the cropped area in macroblocks.
20-18 SW_ROTATION_MODE	Rotation mode: 000b - rotation disabled 001b - rotate + 90 010b - rotate – 90 011b - horizontal flip (mirror) 100b - vertical flip 101b - rotate 180
17-10 SW_COLOR_COEFFF	Coefficient f, used with Y to adjust brightness
9-0 SW_COLOR_COEFFE	Coefficient e, used with Cb to calculate blue component value

14.1.5.1.33 PP input size and -cropping register (SWREG72)

14.1.5.1.33.1 Offset

Register	Offset
SWREG72	120h

14.1.5.1.33.2 Diagram



14.1.5.1.33.3 Fields

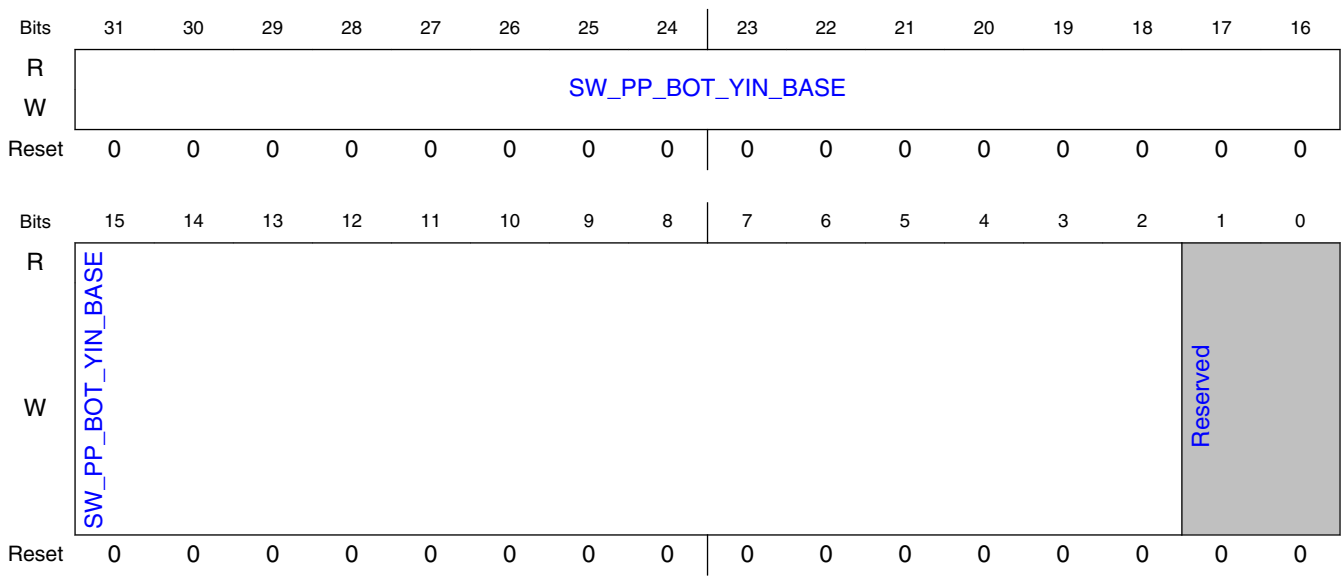
Field	Function
31-24 SW_CROP_STARTY	Start coordinate y for the cropped area in macroblocks.
23 —	Reserved.
22-18 SW_RANGEMAP_COEF_Y	Range map value for Y component
17 —	Reserved.
16-9 SW_PP_IN_HEIGHT	PP input picture height in MBs. Can be cropped from a bigger input picture in external mode
8-0 SW_PP_IN_WIDTH	PP input picture width in MBs. Can be cropped from a bigger input picture in external mode

14.1.5.1.34 PP input picture base address for Y bottom field (SWREG73)

14.1.5.1.34.1 Offset

Register	Offset
SWREG73	124h

14.1.5.1.34.2 Diagram



14.1.5.1.34.3 Fields

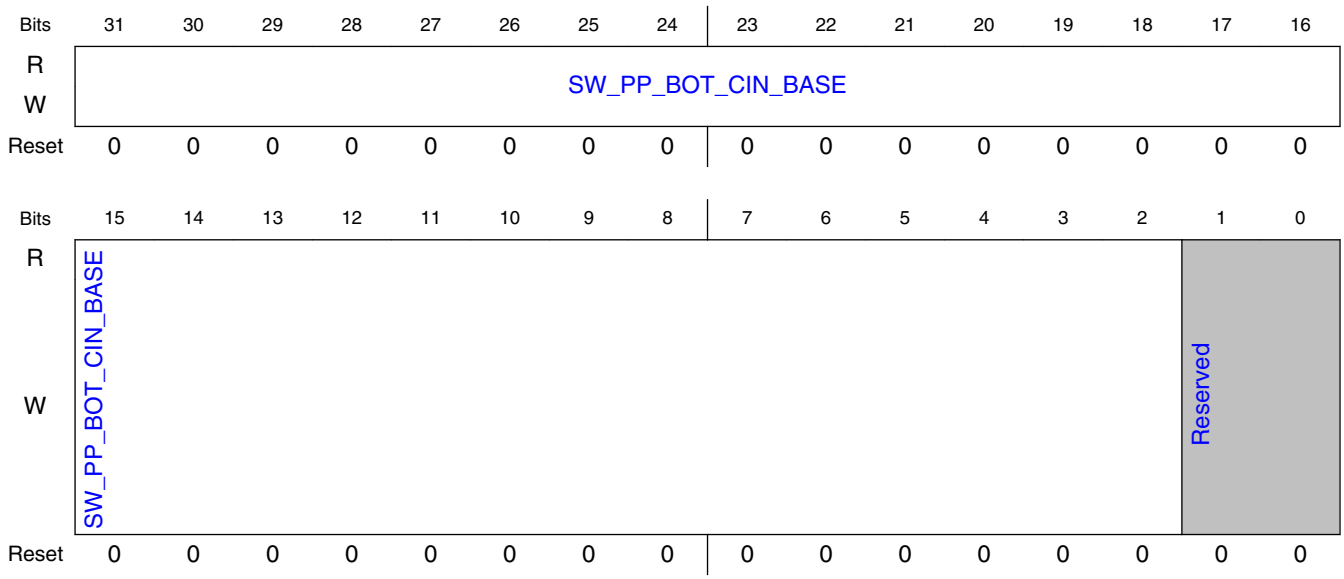
Field	Function
31-2 SW_PP_BOT_YIN_BASE	PP input Y base for bottom field
1-0 —	Reserved.

14.1.5.1.35 PP input picture base for Ch bottom field (SWREG74)

14.1.5.1.35.1 Offset

Register	Offset
SWREG74	128h

14.1.5.1.35.2 Diagram



14.1.5.1.35.3 Fields

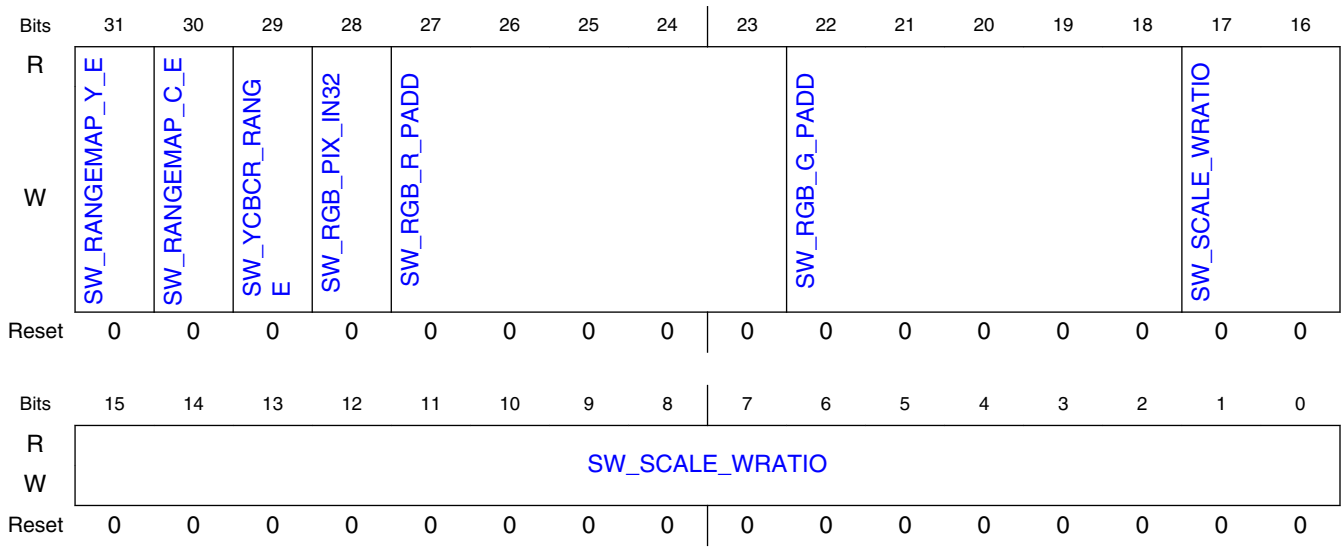
Field	Function
31-2 SW_PP_BOT_CIN_BASE	PP input C base for bottom field (mixed chrominance)
1-0 —	Reserved.

14.1.5.1.36 Scaling register 0 ratio and padding for R and G (SWREG79)

14.1.5.1.36.1 Offset

Register	Offset
SWREG79	13Ch

14.1.5.1.36.2 Diagram



14.1.5.1.36.3 Fields

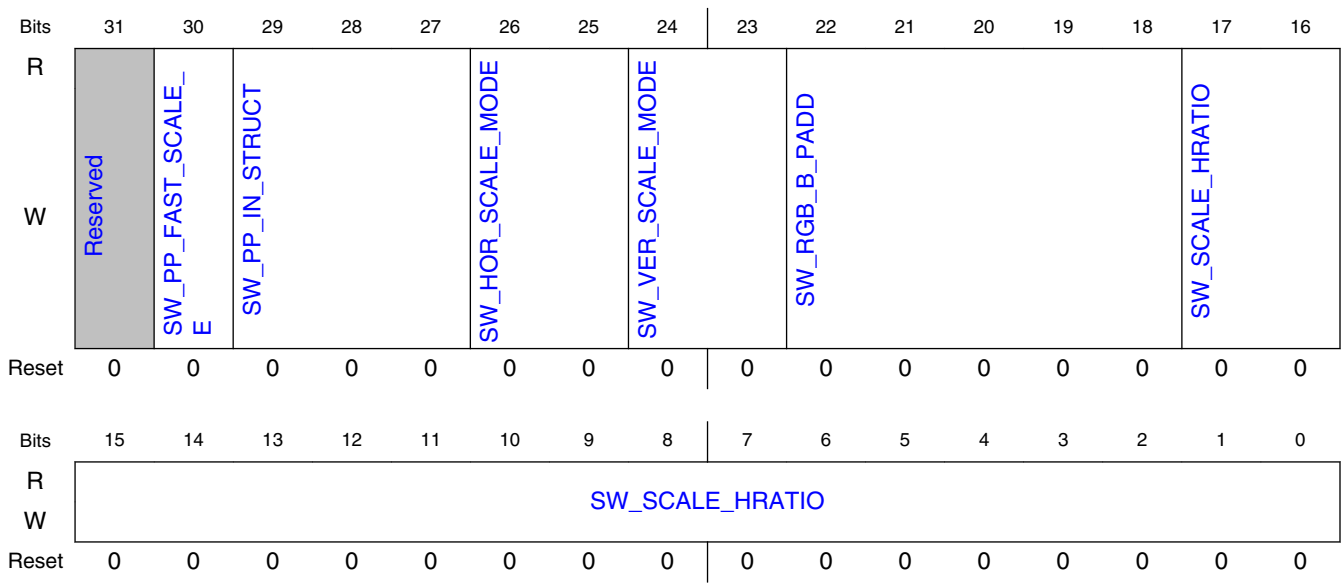
Field	Function
31 SW_RANGEMAP_Y_E	Range map enable for Y component
30 SW_RANGEMAP_C_E	Range map enable for chrominance component
29 SW_YCBCR_RANGE	Defines the YCbCr range in RGB conversion: 0b - 16…235 for Y, 16…240 for Chrominance. 1b - 0…255 for all components
28 SW_RGB_PIX_IN32	RGB pixel amount/ 32 bit word 0b - 1 RGB pixel/32 bit 1b - 2 RGB pixels/32 bit
27-23 SW_RGB_R_PADD	Amount of ones that will be padded in front of the R-component
22-18 SW_RGB_G_PADD	Amount of ones that will be padded in front of the G-component
17-0 SW_SCALE_WRATIO	Scaling ratio for width (outputw-1/inputw-1)

14.1.5.1.37 Scaling ratio register 1 and padding for B (SWREG80)

14.1.5.1.37.1 Offset

Register	Offset
SWREG80	140h

14.1.5.1.37.2 Diagram



14.1.5.1.37.3 Fields

Field	Function
31 —	Reserved.
30 SW_PP_FAST_SCALE_E	0b - fast downscaling is not enabled 1b - fast downscaling is enabled. The quality of the picture is decreased but performance is improved.
29-27 SW_PP_IN_STRUCT	PP input data picture structure: 000b - Top field / progressive frame structure: Read input data from top field base address /frame base address and read every line. 001b - Bottom field structure: Read input data from bottom field base address and read every line. 010b - Interlaced field structure: Read input data from both top and bottom field base address and take every line from each field. 011b - Interlaced frame structure: Read input data from both top and bottom field base address and take every second line from each field. 100b - Ripped top field structure: Read input data from top field base address and read every second line.

Table continues on the next page...

Field	Function
	101b - Ripped bottom field structure: Read input data from bottom field base address and read every second line.
26-25 SW_HOR_SCALE_MODE	Horizontal scaling mode: 00b - Off 01b - Upscale 10b - Downscale
24-23 SW_VER_SCALE_MODE	Vertical scaling mode: 00b - Off 01b - Upscale 10b - Downscale
22-18 SW_RGB_B_PADD	Amount of ones that will be padded in front of the B-component
17-0 SW_SCALE_HRATIO	Scaling ratio for height (outputh-1/inputh-1)

14.1.5.1.38 Scaling ratio register 2 (SWREG81)

14.1.5.1.38.1 Offset

Register	Offset
SWREG81	144h

14.1.5.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_WSCALE_INVRA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_HSCALE_INVRA															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.38.3 Fields

Field	Function
31-16	Inverse scaling ratio for width, or ch (inputw-1 / outputw-1)

Table continues on the next page...

Field	Function
SW_WSCALE_INVRA	
15-0 SW_HSCALE_INVRA	Inverse scaling ratio for height or cv (inputh-1 / outputh-1)

14.1.5.1.39 Rmask register (SWREG82)

14.1.5.1.39.1 Offset

Register	Offset
SWREG82	148h

14.1.5.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_R_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_R_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.39.3 Fields

Field	Function
31-0 SW_R_MASK	Bit mask for R component (and alpha channel)

14.1.5.1.40 Gmask register (SWREG83)

14.1.5.1.40.1 Offset

Register	Offset
SWREG83	14Ch

14.1.5.1.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_G_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_G_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.40.3 Fields

Field	Function
31-0 SW_G_MASK	Bit mask for G component (and alpha channel)

14.1.5.1.41 Bmask register (SWREG84)

14.1.5.1.41.1 Offset

Register	Offset
SWREG84	150h

14.1.5.1.41.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_B_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_B_MASK															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.41.3 Fields

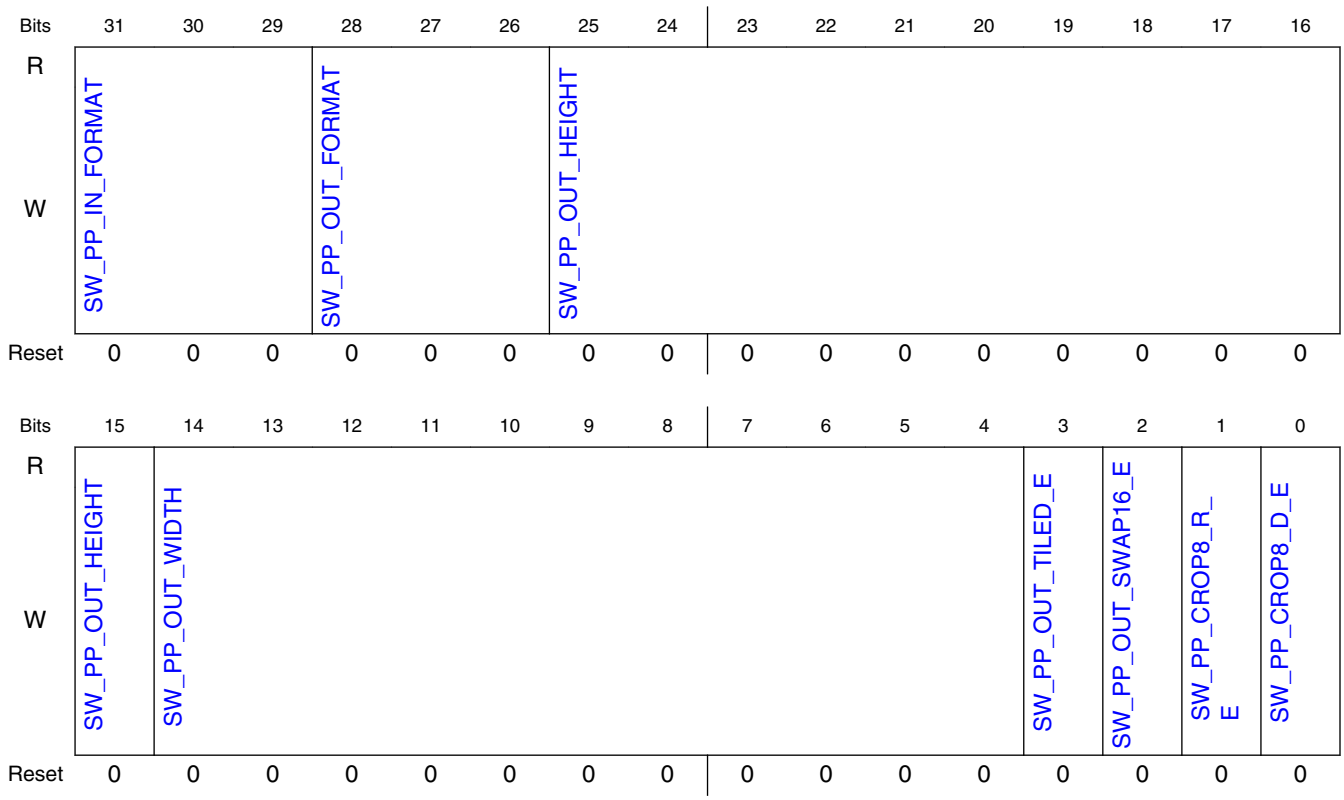
Field	Function
31-0 SW_B_MASK	Bit mask for B component (and alpha channel)

14.1.5.1.42 Post-processor control register (SWREG85)

14.1.5.1.42.1 Offset

Register	Offset
SWREG85	154h

14.1.5.1.42.2 Diagram



14.1.5.1.42.3 Fields

Field	Function
31-29 SW_PP_IN_FORMAT	PP input picture data format 000b - YUYV 4:2:2 interleaved (supported only in external mode) 001b - YCbCr 4:2:0 Semi-planar in linear raster-scan format 010b - YCbCr 4:2:0 planar (supported only in external mode) 011b - YCbCr 4:0:0 (supported only in pipelined mode) 100b - YCbCr 4:2:2 Semi-planar (supported only in pipelined mode) 101b - YCbCr 4:2:0 Semi-planar in tiled format (supported only in external mode (8170 decoder only) 110b - Reserved 111b - Escape pp input data format. Defined in swreg86.
28-26 SW_PP_OUT_FORMAT	PP output picture data format: 000b - RGB 001b - YCbCr 4:2:0 planar (Not supported) 010b - YCbCr 4:2:2 planar (Not supported) 011b - YUYV 4:2:2 interleaved 100b - YCbCr 4:4:4 planar (Not supported) 101b - YCh 4:2:0 chrominance interleaved 110b - YCh 4:2:2 (Not supported) 111b - YCh 4:4:4 (Not supported)

Table continues on the next page...

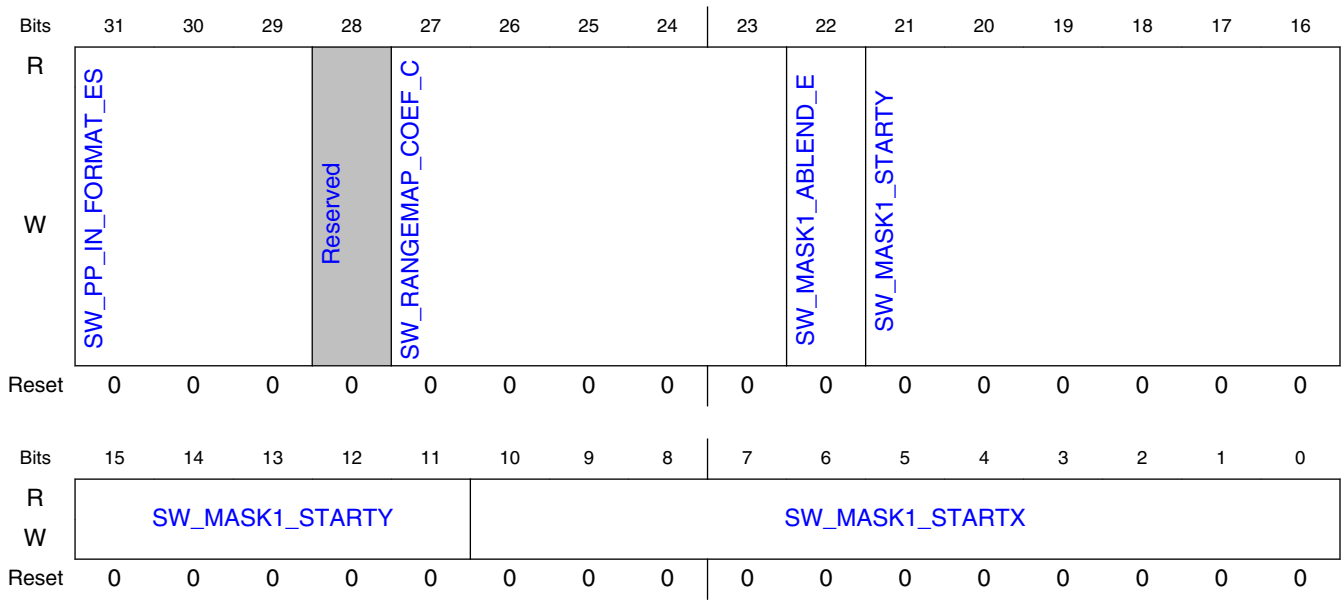
Field	Function
25-15 SW_PP_OUT_HEIGHT	Scaled picture height in pixels (Must be dividable by 2 or by any if Pixel Accurate PP output configuration is enabled) Max scaled picture height is 1920 pixels or maximum three times the input source height minus 8 pixels
14-4 SW_PP_OUT_WIDTH	Scaled picture width in pixels. Must be dividable by 8 or by any if Pixel Accurate PP output configuration is enabled. Max scaled picture width is 1920 pixels or maximum three times the input source width minus 8 pixels
3 SW_PP_OUT_TILED_E	Tiled mode enable for PP output. Can be used only for YCbYCr 422 output format. Can be used only if corresponding configuration supports this feature. Tile size is 4x4 pixels.
2 SW_PP_OUT_SWAP16_E	PP output swap 16, swaps 16 bit half inside of 32 bit word. Can be used for 16 bit RGB to change pixel orders but is valid also for any output format. NOTE: requires that configuration of SW_PPD_OEN_VERSION=1
1 SW_PP_CROP8_R_E	PP input picture width is not 16 pixels multiple. Only 8 pixels of the most right MB of the unrotated input picture is used for PP input.
0 SW_PP_CROP8_D_E	PP input picture height is not 16 pixels multiple. Only 8 pixel rows of the most down MB of the unrotated input picture is used for PP input.

14.1.5.1.43 Mask 1 start coordinate register (SWREG86)

14.1.5.1.43.1 Offset

Register	Offset
SWREG86	158h

14.1.5.1.43.2 Diagram



14.1.5.1.43.3 Fields

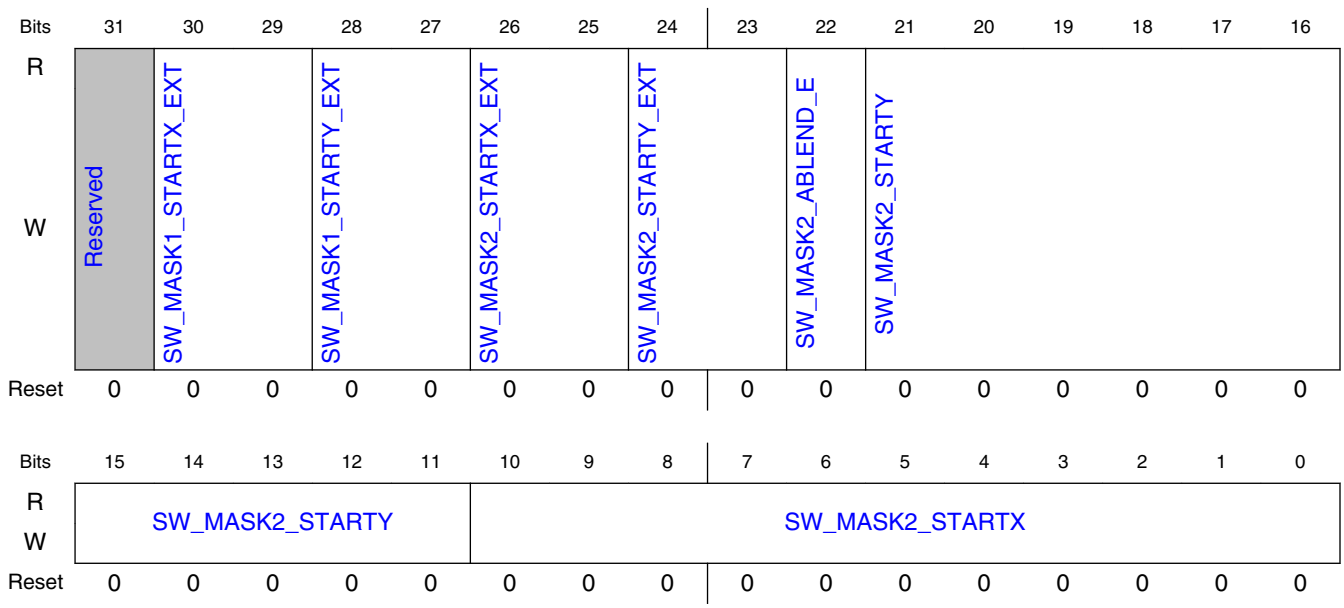
Field	Function
31-29 SW_PP_IN_FORMAT_ES	Escape PP in format. Used if sw_pp_in_format is defined to 7. 000b - YCbCr 4:4:4 001b - YCbCr 4:1:1
28 —	Reserved.
27-23 SW_RANGEMAP_COEF_C	Range map value for chrominance component
22 SW_MASK1_ABLEND_E	Mask 1 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 1 base address. Alpha blending can be enabled only for RGB/ YUYV 422 data.
21-11 SW_MASK1_STARTY	Vertical start pixel for mask area 1. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions
10-0 SW_MASK1_STARTX	Horizontal start pixel for mask area 1. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output luminance picture. See Table 47 for restrictions

14.1.5.1.44 Mask 2 start coordinate register + Mask extensions (SWREG87)

14.1.5.1.44.1 Offset

Register	Offset
SWREG87	15Ch

14.1.5.1.44.2 Diagram



14.1.5.1.44.3 Fields

Field	Function
31 —	Reserved.
30-29 SW_MASK1_STARTX_EXT	Extended coordinate upto 4k resolution
28-27 SW_MASK1_STARTY_EXT	Extended coordinate upto 4k resolution
26-25 SW_MASK2_STARTX_EXT	Extended coordinate upto 4k resolution

Table continues on the next page...

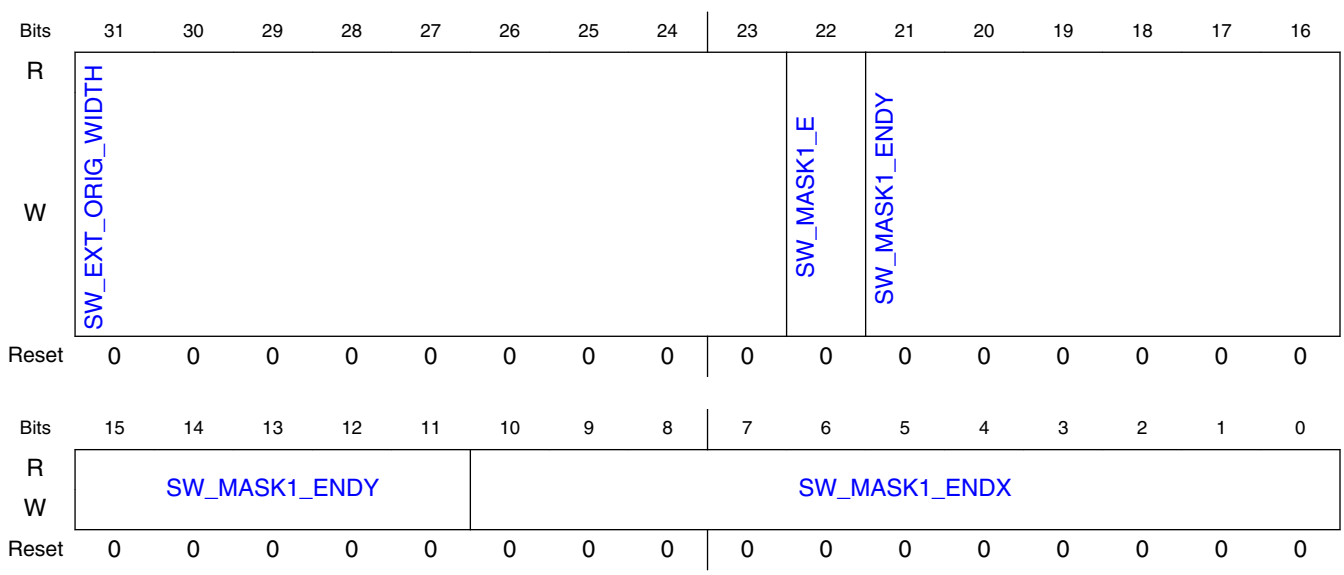
Field	Function
24-23 SW_MASK2_STARTY_EXT	Extended coordinate upto 4k resolution
22 SW_MASK2_ABLEND_E	Mask 2 alpha blending enable. Instead of masking the output picture the alpha blending is performed. Alpha blending source can be found from alpha blend 2 base address. Alpha blending can be enabled only for RGB/YUYV 422 data.
21-11 SW_MASK2_STARTY	Vertical start pixel for mask area 2. Defines the y coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions
10-0 SW_MASK2_STARTX	Horizontal start pixel for mask area 2. Defines the x coordinate. Coordinate 0,0 means the up-left corner in PP output Y picture. See Table 47 for restrictions

14.1.5.1.45 Mask 1 size and PP original width register (SWREG88)

14.1.5.1.45.1 Offset

Register	Offset
SWREG88	160h

14.1.5.1.45.2 Diagram



14.1.5.1.45.3 Fields

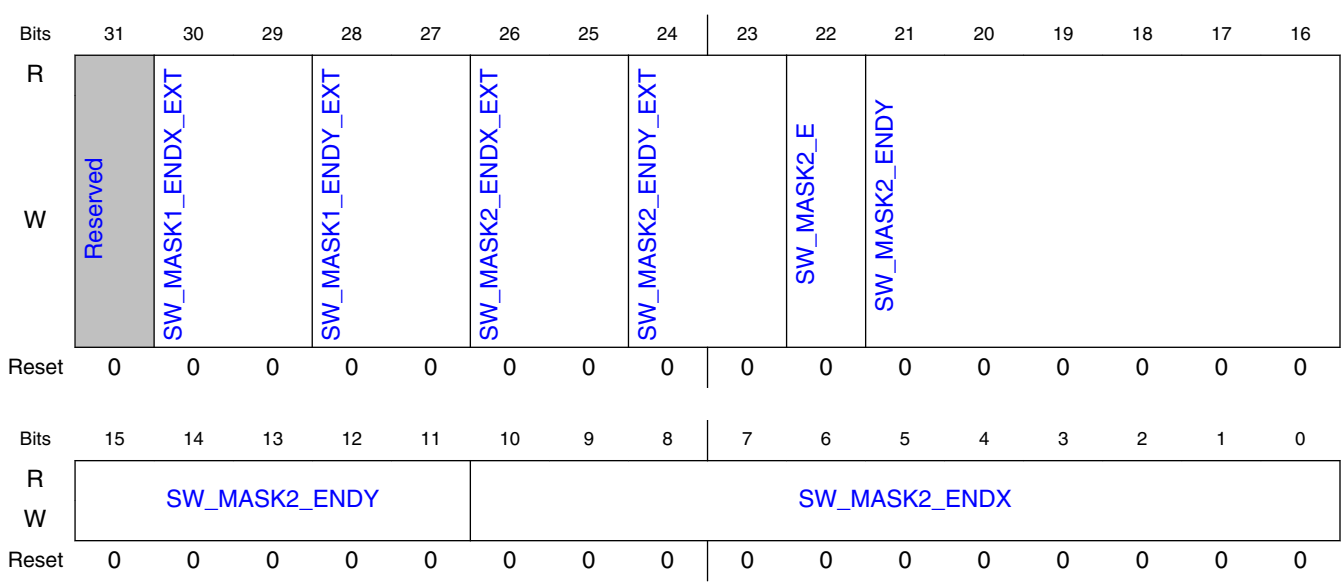
Field	Function
31-23 SW_EXT_ORIG_WIDTH	PP input picture original width in macro blocks.
22 SW_MASK1_E	Mask 1 enable. If mask 1 is used this bit is high
21-11 SW_MASK1_ENDY	Mask 1 end coordinate y in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateY, ScaledHeight].
10-0 SW_MASK1_ENDX	Mask 1 end coordinate x in pixels (inside of PPD output picture). Range must be between [Mask1StartCoordinateX, ScaledWidth]

14.1.5.1.46 Mask 2 size register + mask extensions (SWREG89)

14.1.5.1.46.1 Offset

Register	Offset
SWREG89	164h

14.1.5.1.46.2 Diagram



14.1.5.1.46.3 Fields

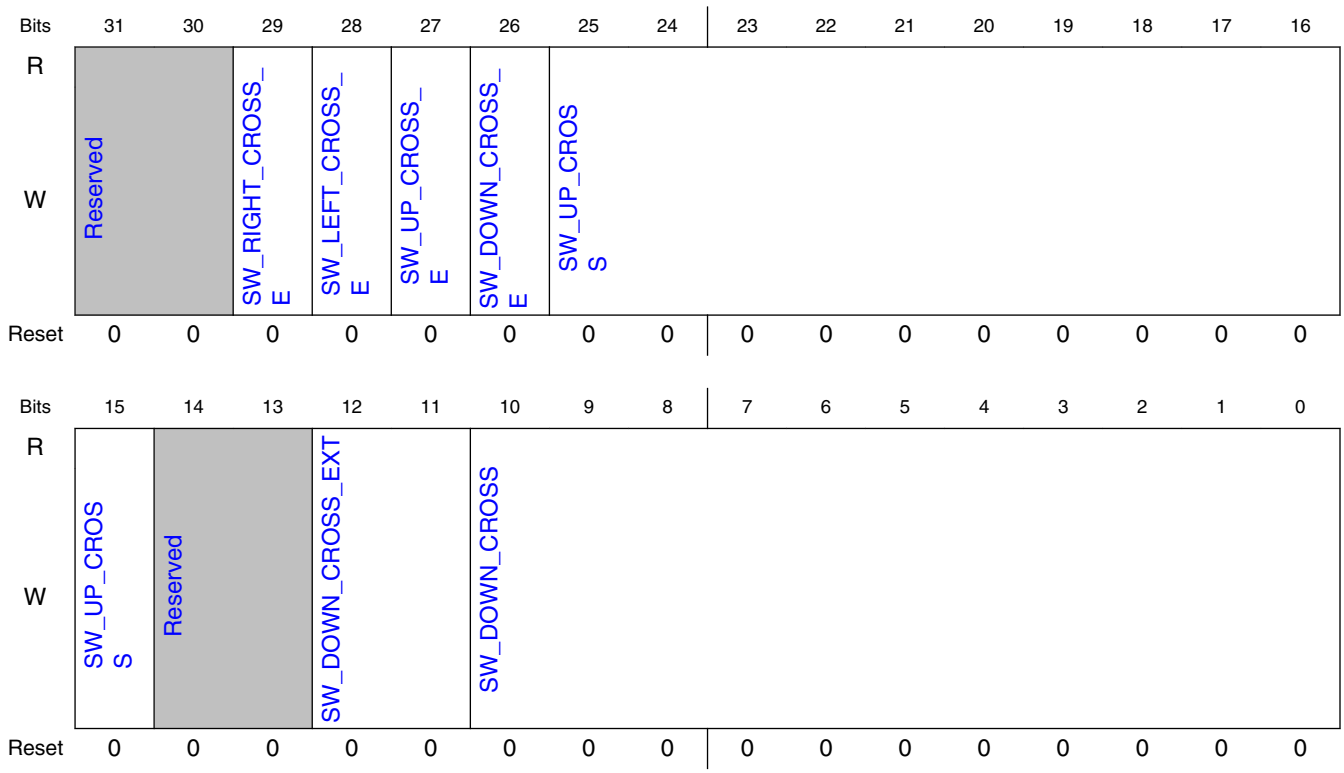
Field	Function
31 —	Reserved.
30-29 SW_MASK1_ENDX_EXT	Extended coordinate upto 4k resolution
28-27 SW_MASK1_ENDY_EXT	Extended coordinate upto 4k resolution
26-25 SW_MASK2_ENDX_EXT	Extended coordinate upto 4k resolution
24-23 SW_MASK2_ENDY_EXT	Extended coordinate upto 4k resolution
22 SW_MASK2_E	Mask 2 enable. If mask 1 is used this bit is high
21-11 SW_MASK2_ENDY	Mask 2 end coordinate y in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateY, ScaledHeight].
10-0 SW_MASK2_ENDX	Mask 2 end coordinate x in pixels (inside of PP output picture). Range must be between [Mask2StartCoordinateX, ScaledWidth].

14.1.5.1.47 PiP register 0 (SWREG90)

14.1.5.1.47.1 Offset

Register	Offset
SWREG90	168h

14.1.5.1.47.2 Diagram



14.1.5.1.47.3 Fields

Field	Function
31-30 —	Reserved.
29 SW_RIGHT_CROSS_E	Right side overcross enable. 0b - No right side overcross 1b - Right side overcross
28 SW_LEFT_CROSS_E	Left side overcross enable. 0b - No left side overcross 1b - Left side overcross
27 SW_UP_CROSS_E	Upward overcross enable. 0b - No upward overcross 1b - Upward overcross
26 SW_DOWN_CROSS_E	Downward overcross enable. 0b - No downward overcross 1b - Downward overcross
25-15 SW_UP_CROSS	Amount of upward overcross (vertical pixels outside of display from the upper side). Range must be between [0, ScaledHeight].

Table continues on the next page...

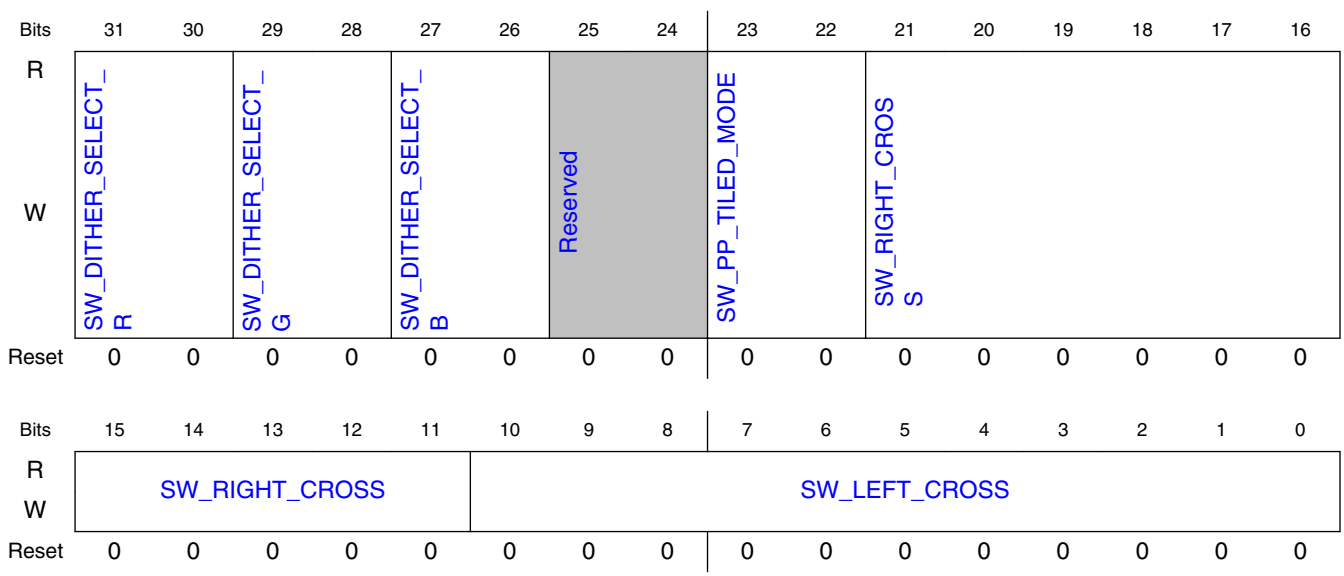
Field	Function
14-13 —	Reserved.
12-11 SW_DOWN_CROSS_EXT	Extended coordinate for 4k resolution
10-0 SW_DOWN_CROSS	Amount of downward overcross (vertical pixels outside of display from the down side). Range must be between [0, ScaledHeight].

14.1.5.1.48 PiP register 1 and dithering control (SWREG91)

14.1.5.1.48.1 Offset

Register	Offset
SWREG91	16Ch

14.1.5.1.48.2 Diagram



14.1.5.1.48.3 Fields

Field	Function
31-30	Dithering control for R channel: 00b - dithering disabled

Table continues on the next page...

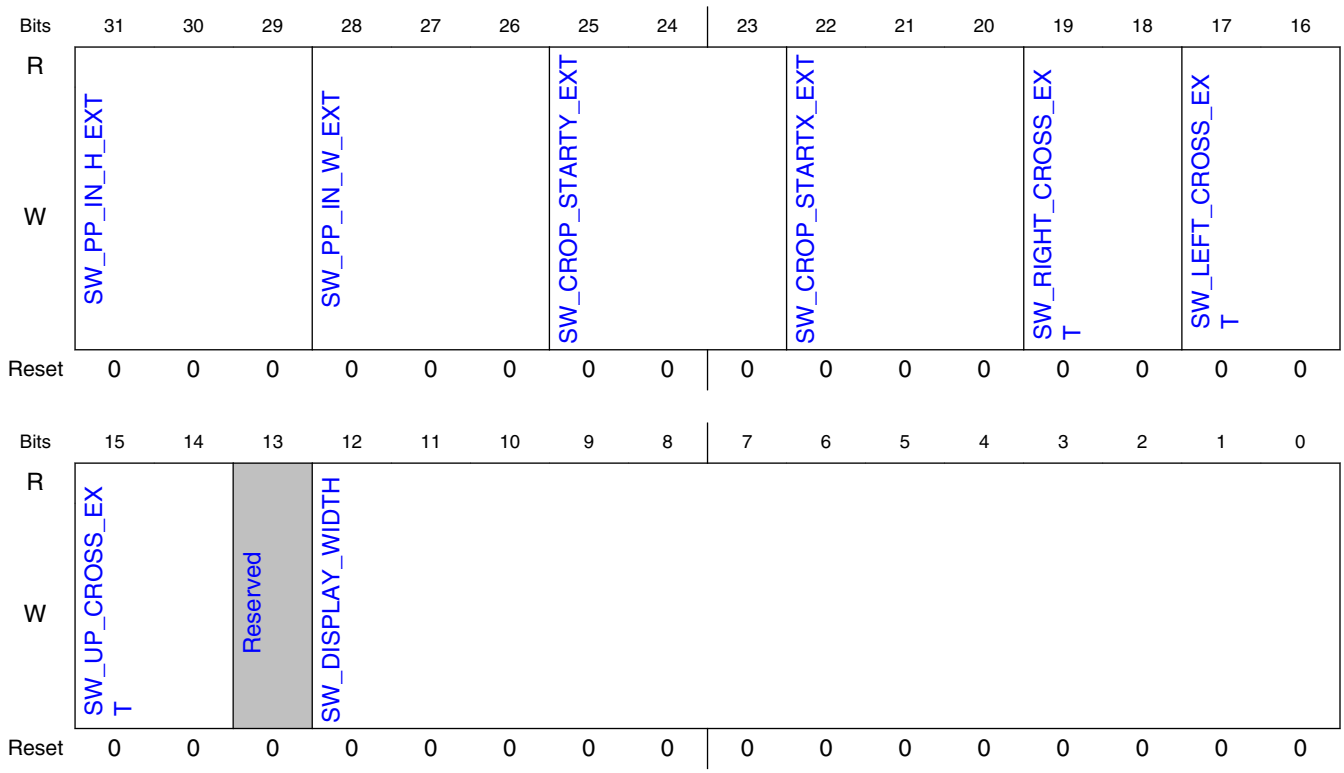
Field	Function
SW_DITHER_SELECT_R	01b - use four-bit dither matrix 10b - use five-bit dither matrix 11b - use six-bit dither matrix
29-28 SW_DITHER_SELECT_G	Dithering control for G channel: 00b - dithering disabled 01b - use four-bit dither matrix 10b - use five-bit dither matrix 11b - use six-bit dither matrix
27-26 SW_DITHER_SELECT_B	Dithering control for B channel: 00b - dithering disabled 01b - use four-bit dither matrix 10b - use five-bit dither matrix 11b - use six-bit dither matrix
25-24 —	Reserved.
23-22 SW_PP_TILED_MODE	Input data is in tiled mode (at the moment valid only for YCbCr 420 data, pipeline or external mode): 00b - Tiled mode not used 01b - Tiled mode enabled for 8x4 sized tiles
21-11 SW_RIGHT_CROSS	Amount of right side overcross (Horizontal pixels outside of display from the right side). Range must be between [0, ScaledWidth].
10-0 SW_LEFT_CROSS	Amount of left side overcross (Horizontal pixels outside of display from the left side). Range must be between [0, ScaledWidth].

14.1.5.1.49 Display width and PP input size extension register (SWREG92)

14.1.5.1.49.1 Offset

Register	Offset
SWREG92	170h

14.1.5.1.49.2 Diagram



14.1.5.1.49.3 Fields

Field	Function
31-29 SW_PP_IN_H_EXT	Extended PP input height. Used with WEBP
28-26 SW_PP_IN_W_EXT	Extended PP input width. Used with WEBP
25-23 SW_CROP_STARTY_EXT	Extended PP input crop start coordinate x. Used with WEBP
22-20 SW_CROP_STARTX_EXT	Extended PP input crop start coordinate y. Used with WEBP
19-18 SW_RIGHT_CROSS_EXT	Extended coordinate for 4k resolution
17-16 SW_LEFT_CROSS_EXT	Extended coordinate for 4k resolution

Table continues on the next page...

Field	Function
15-14 SW_UP_CROSS_EXT	Extended coordinate for 4k resolution
13 —	Reserved.
12-0 SW_DISPLAY_WIDTH	Width of the display in pixels. Max 4k (depends on HW config support)

14.1.5.1.50 Base address for alpha blend 1 gui component (SWREG93)

14.1.5.1.50.1 Offset

Register	Offset
SWREG93	174h

14.1.5.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ABLEND1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ABLEND1_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.50.3 Fields

Field	Function
31-0 SW_ABLEND1_BASE	Base address for alpha blending input 1 (if mask1 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 1 size or with ablend1_scanline if ablend cropping is supported in configuration.

14.1.5.1.51 Base address for alpha blend 2 gui component (SWREG94)

14.1.5.1.51.1 Offset

Register	Offset
SWREG94	178h

14.1.5.1.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_ABLEND2_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_ABLEND2_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.1.51.3 Fields

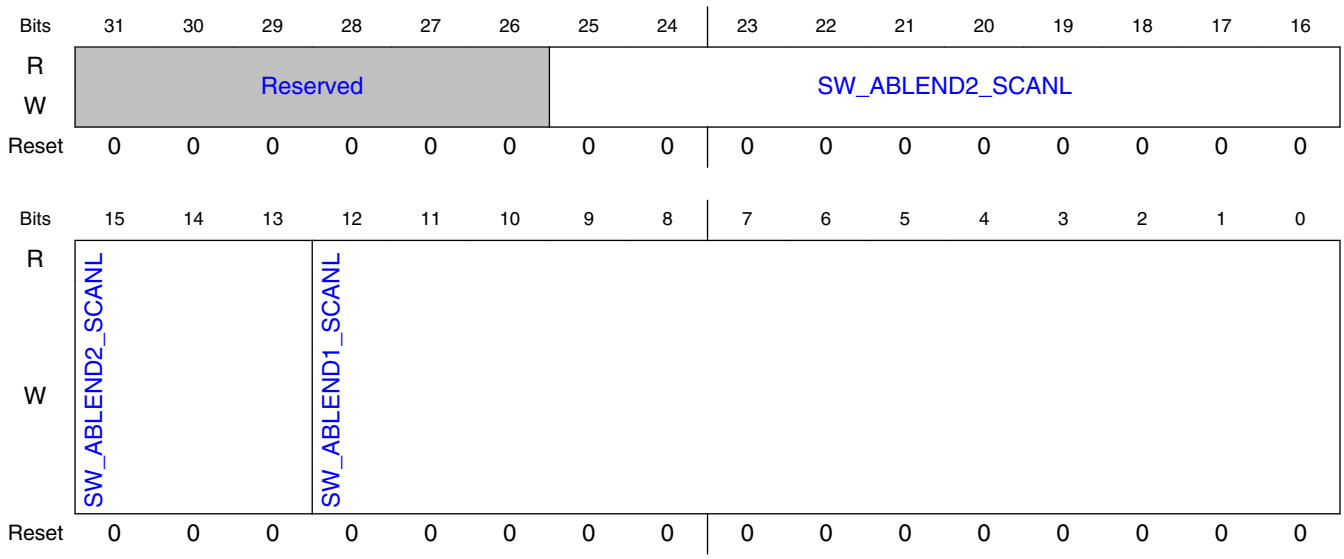
Field	Function
31-0 SW_ABLEND2_BASE	Base address for alpha blending input 2 (if mask2 is used in alpha blending mode). Format of data is 24 bit RGB/ YCbCr and endian/swap -mode is as in PP input. Amount of data is informed with mask 2 size or with ablend2_scanline if ablend cropping is supported in configuration.

14.1.5.1.52 Alpha blend input cropping register (scanline for cropping) (SWREG95)

14.1.5.1.52.1 Offset

Register	Offset
SWREG95	17Ch

14.1.5.1.52.2 Diagram



14.1.5.1.52.3 Fields

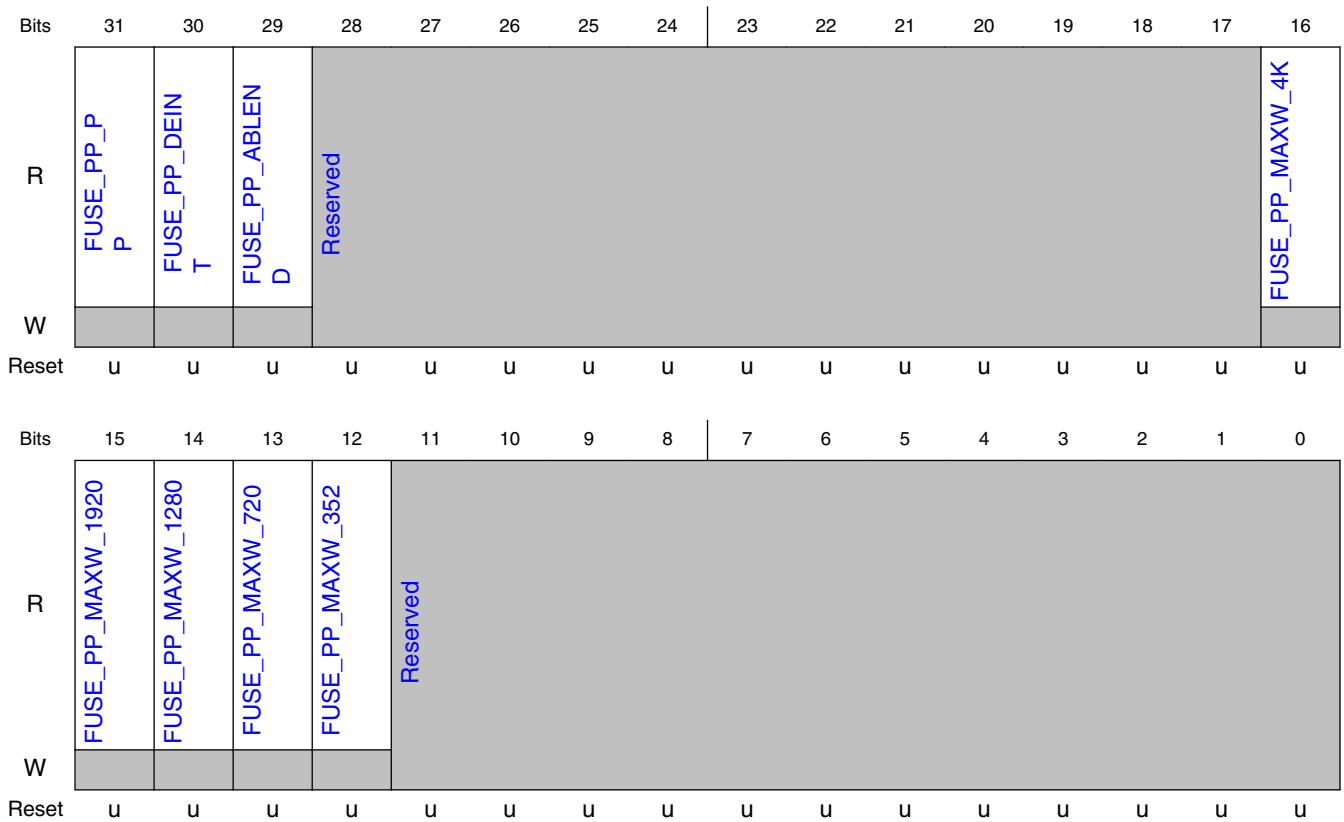
Field	Function
31-26 —	Reserved.
25-13 SW_ABLEND2_SCANL	Scanline width in pixels for Ablend 2. Usage enabled if corresponding configuration bit is enabled
12-0 SW_ABLEND1_SCANL	Scanline width in pixels for Ablend 1. Usage enabled if corresponding configuration bit is enabled

14.1.5.1.53 PP fuse register (SWREG99)

14.1.5.1.53.1 Offset

Register	Offset
SWREG99	18Ch

14.1.5.1.53.2 Diagram



14.1.5.1.53.3 Fields

Field	Function
31 FUSE_PP_PP	1 = PP enabled
30 FUSE_PP_DEINT	1 = Deinterlacing enabled
29 FUSE_PP_ABLEND	1 = Alpha Blending enabled
28-17 —	Reserved.
16 FUSE_PP_MAXW_4K	1 = Max PP output width up to 4096 pixels enabled. Priority coded with priority 1
15 FUSE_PP_MAXW_1920	1 = Max PP output width up to 1920 pixels enabled. Priority coded with priority 2

Table continues on the next page...

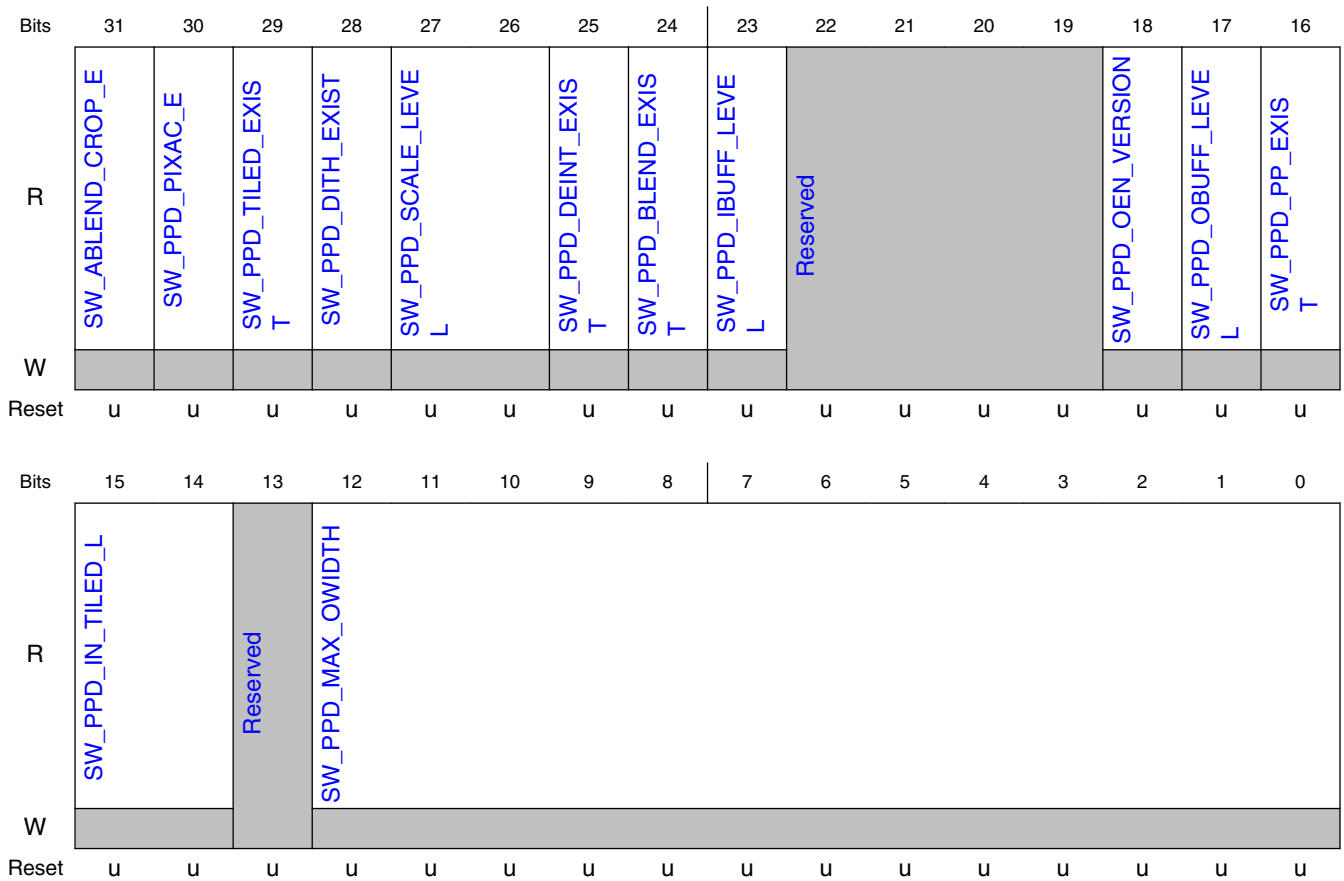
Field	Function
14 FUSE_PP_MAXW_1280	1 = Max PP output width up to 1280 pixels enabled. Priority coded with priority 3
13 FUSE_PP_MAXW_720	1 = Max PP output width up to 720 pixels enabled. Priority coded with priority 4
12 FUSE_PP_MAXW_352	1 = Max PP output width up to 352 pixels enabled. Priority coded with priority 5
11-0 —	Reserved.

14.1.5.1.54 Synthesis configuration register post-processor (SWREG100)

14.1.5.1.54.1 Offset

Register	Offset
SWREG100	190h

14.1.5.1.54.2 Diagram



14.1.5.1.54.3 Fields

Field	Function
31 SW_ABLEND_CROP_E	Alpha blending support for input cropping: 0b - Not supported. External memory must include the exact image of the area being alpha blended. 1b - Supported. External memory can include a picture from blended area can be cropped. Requires usage of swreg95.
30 SW_PPD_PIXAC_E	Pixel Accurate PP output mode exists: 0b - PIP, Scaling and masks can be adjusted by steps of 8 pixels (width) or 2 pixels (height) 1b - PIP, Scaling and masks can be adjusted by steps of 1 pixel for RGB and 2 pixels for subsampled chroma formats (by using bus specific write strobe functionality)
29 SW_PPD_TILED_EXIST	PP output YCbYCr 422 tiled support (4x4 pixel tiles) 0b - Not supported 1b - Supported
28 SW_PPD_DITH_EXIST	Dithering exists: 0b - No 1b - Yes
27-26	Scaling support:

Table continues on the next page...

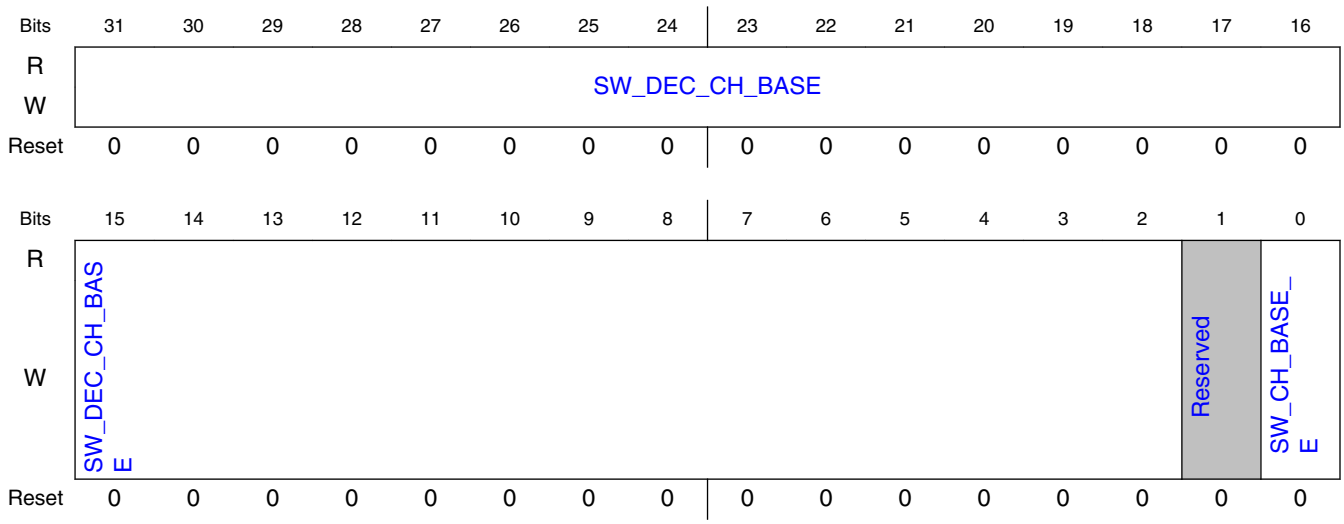
Field	Function
SW_PPD_SCALE_LEVEL	00b - No scaling 01b - Scaling with lo performance architecture 10b - Scaling with high performance architecture 11b - Scaling with high performance architecture + fast downscaling enabled
25 SW_PPD_DEINT_EXIST	De-interlacing exits: 0b - No 1b - Yes
24 SW_PPD_BLEND_EXIST	Alpha blending exists: 0b - No 1b - Yes
23 SW_PPD_IBUFF_LEVEL	PP input buffering level: 0b - 1 MB input buffering is used 1b - 4 MB input buffering is used
22-19 —	Reserved.
18 SW_PPD_OEN_VERSION	PP output endian version: 0b - Endian mode supported for other than RGB 1b - Endian mode supported for any output format
17 SW_PPD_OBUFF_LEVEL	PP output buffering level: 0b - 1 unit output buffering is used 1b - 4 unit output buffering is used
16 SW_PPD_PP_EXIST	PPD exists: 0b - No 1b - Yes
15-14 SW_PPD_IN_TILED_L	PPD input tiled mode support level 00b - not supported 01b - 8x4 tile size supported
13 —	Reserved.
12-0 SW_PPD_MAX_OWIDTH	Max supported PP output width in pixels

14.1.5.1.55 Base address for H264 decoded chroma picture (SWREG102)

14.1.5.1.55.1 Offset

Register	Offset
SWREG102	198h

14.1.5.1.55.2 Diagram



14.1.5.1.55.3 Fields

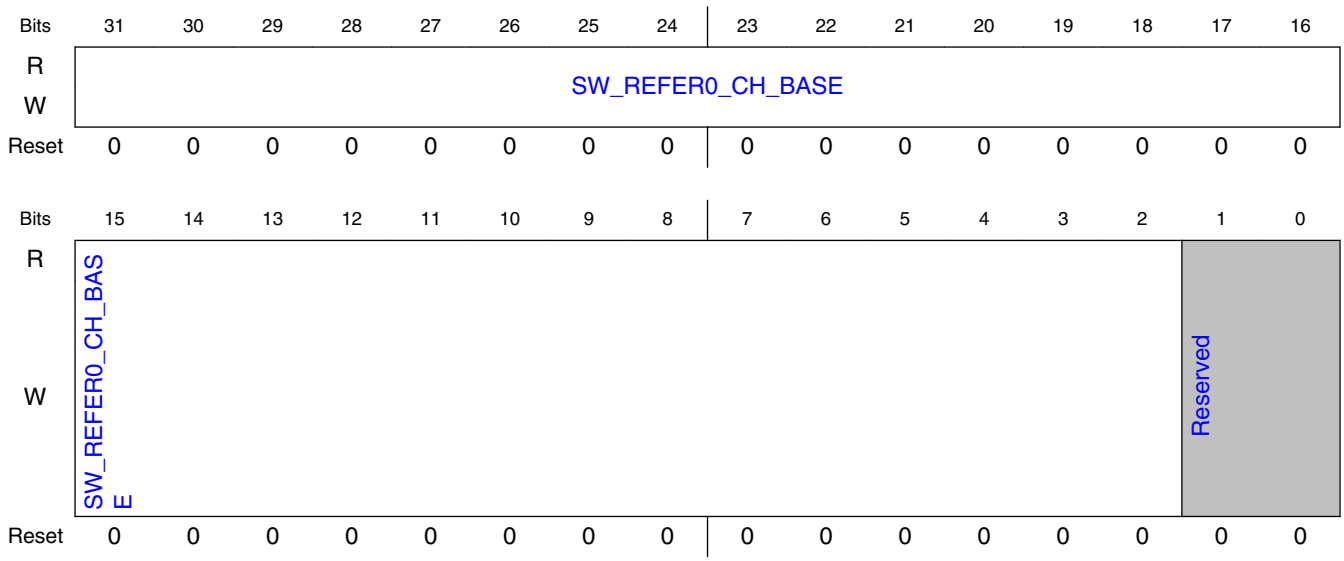
Field	Function
31-2 SW_DEC_CH_BASE	Valid only if chroma address separate mode is enabled. H264: Base address for decoder output chroma picture. Points directly to start of decoder output chroma picture or field.
1 —	Reserved.
0 SW_CH_BASE_E	chroma address separate mode enable: 0b - HW outputs decoded chroma picture to the end of decoded luma picture. HW calculates the chroma picture address according to sw_dec_base and luma data length. 1b - HW outputs decoded chroma picture to independent memory address

14.1.5.1.56 Base address for reference chroma picture index 0 (SWREG103)

14.1.5.1.56.1 Offset

Register	Offset
SWREG103	19Ch

14.1.5.1.56.2 Diagram



14.1.5.1.56.3 Fields

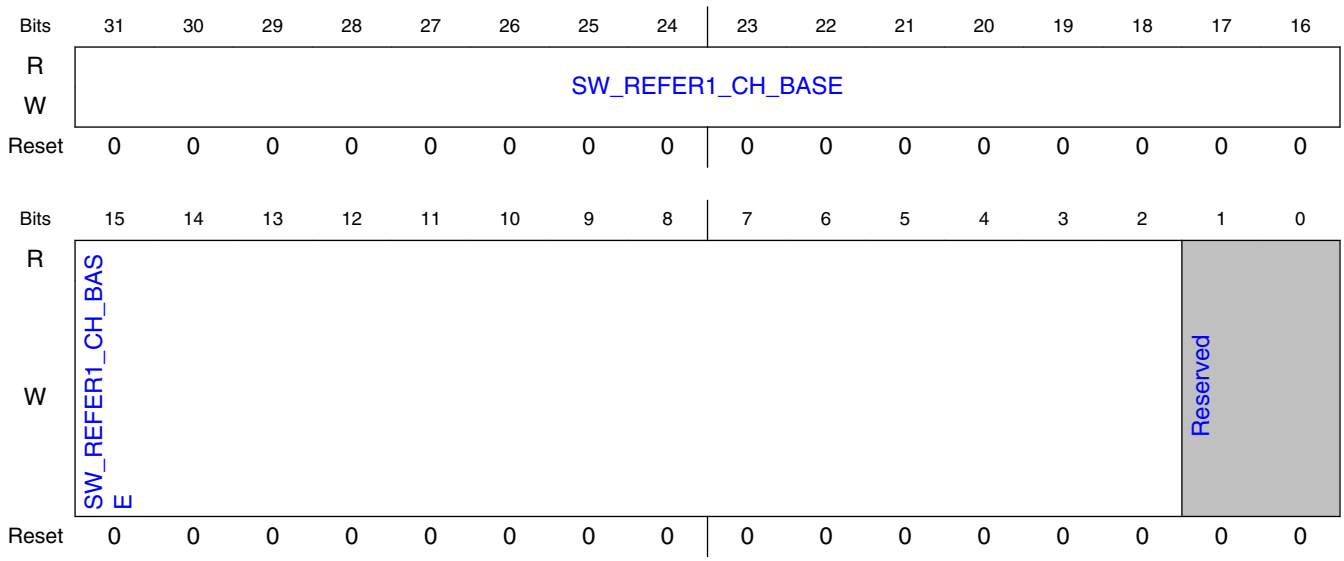
Field	Function
31-2 SW_REFER0_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 0.
1-0 —	Reserved.

14.1.5.1.57 Base address for reference chroma picture index 1 (SWREG104)

14.1.5.1.57.1 Offset

Register	Offset
SWREG104	1A0h

14.1.5.1.57.2 Diagram



14.1.5.1.57.3 Fields

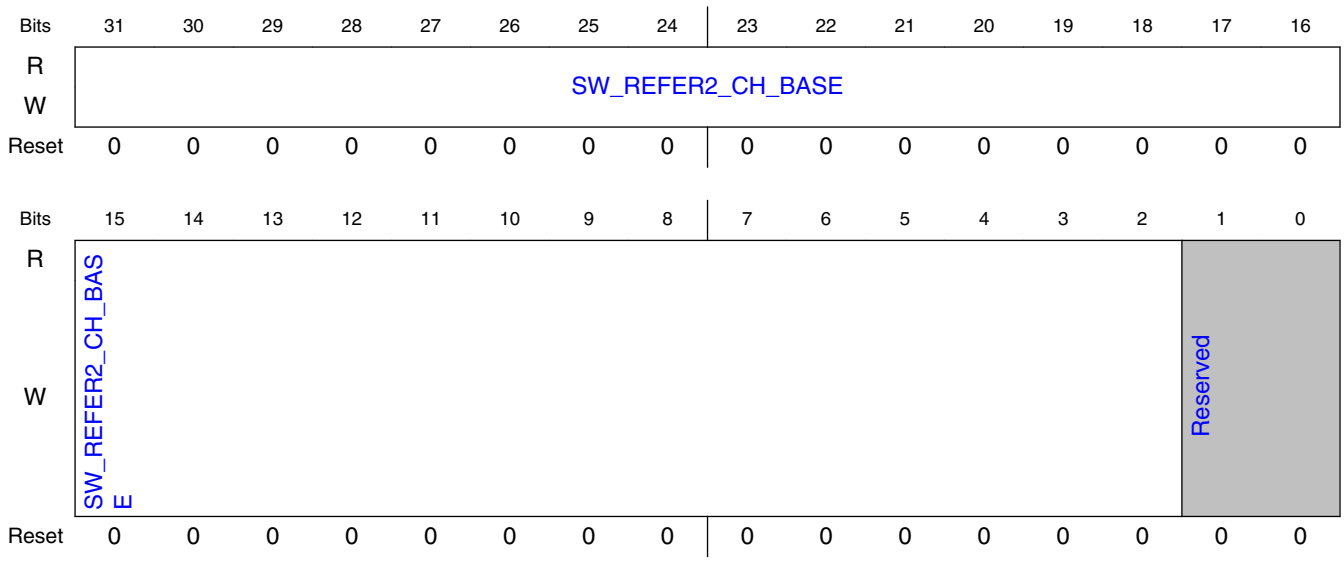
Field	Function
31-2 SW_REFER1_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 1.
1-0 —	Reserved.

14.1.5.1.58 Base address for reference chroma picture index 2 (SWREG105)

14.1.5.1.58.1 Offset

Register	Offset
SWREG105	1A4h

14.1.5.1.58.2 Diagram



14.1.5.1.58.3 Fields

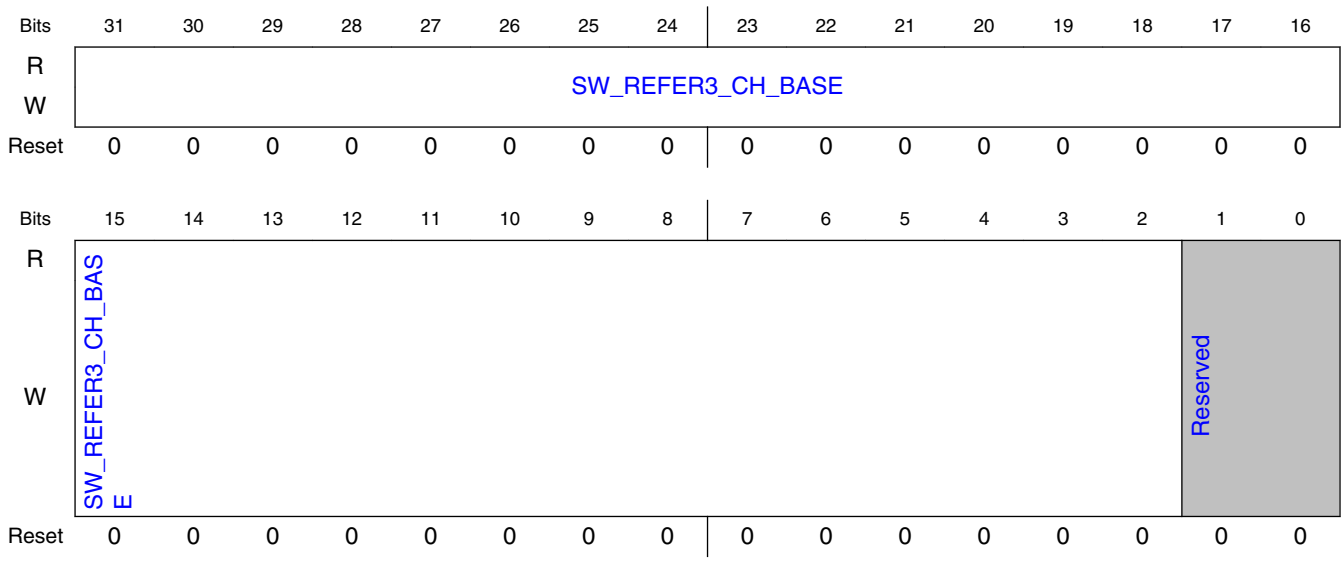
Field	Function
31-2 SW_REFER2_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 2.
1-0 —	Reserved.

14.1.5.1.59 Base address for reference chroma picture index 3 (SWREG106)

14.1.5.1.59.1 Offset

Register	Offset
SWREG106	1A8h

14.1.5.1.59.2 Diagram



14.1.5.1.59.3 Fields

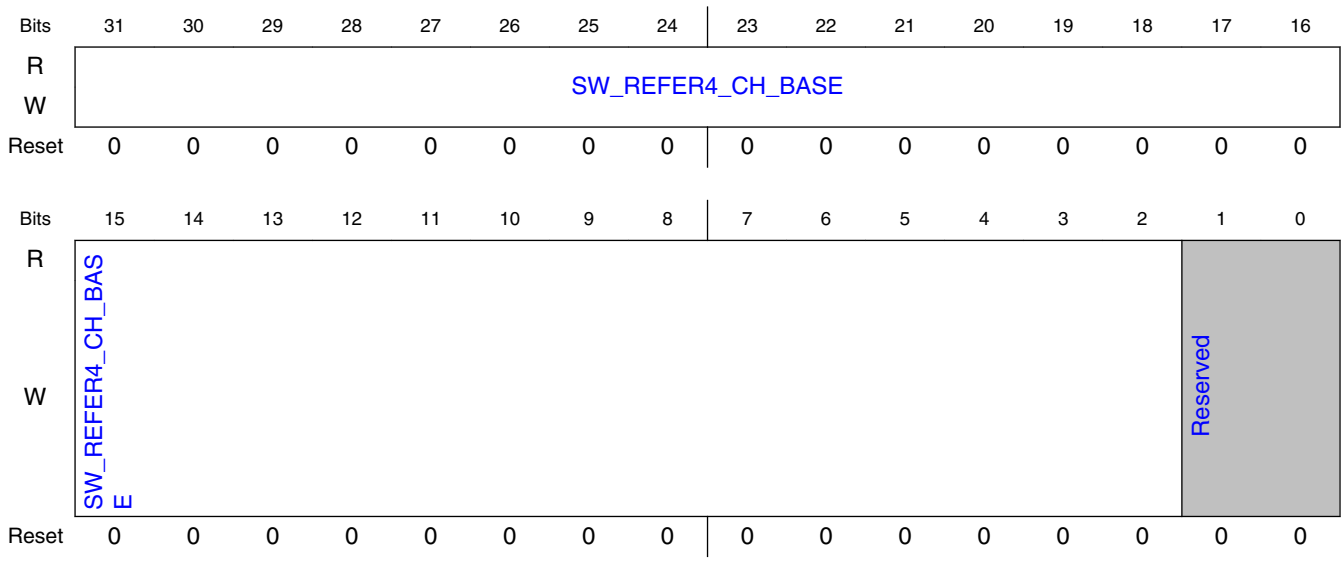
Field	Function
31-2 SW_REFER3_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 3.
1-0 —	Reserved.

14.1.5.1.60 Base address for reference chroma picture index 4 (SWREG107)

14.1.5.1.60.1 Offset

Register	Offset
SWREG107	1ACh

14.1.5.1.60.2 Diagram



14.1.5.1.60.3 Fields

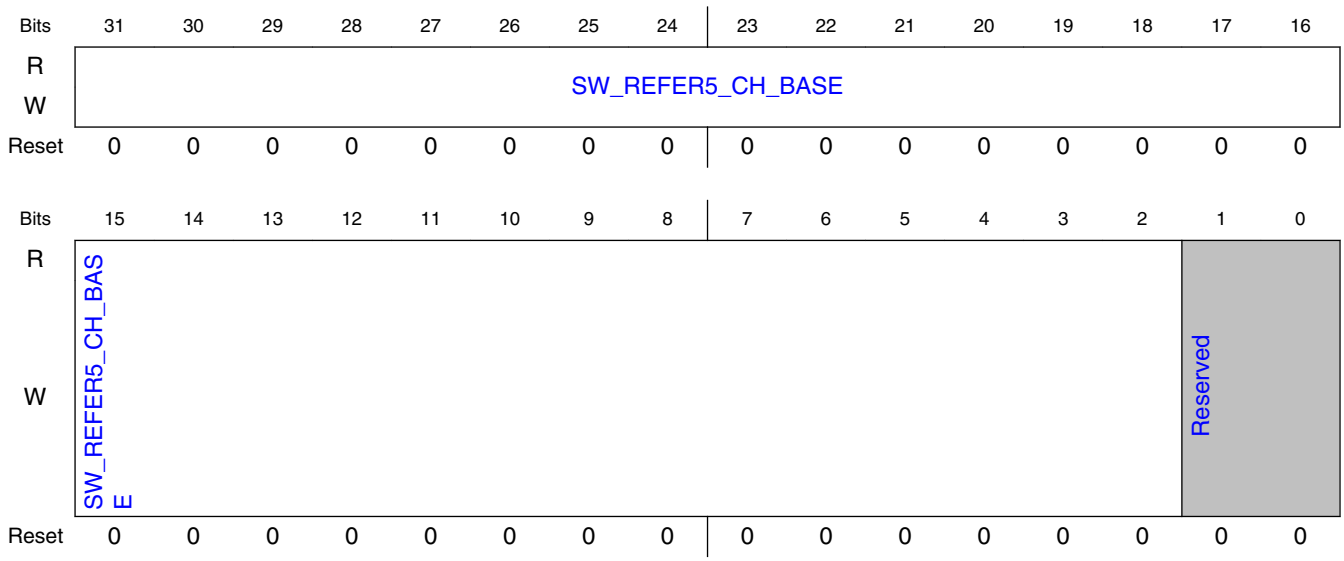
Field	Function
31-2 SW_REFER4_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 4.
1-0 —	Reserved.

14.1.5.1.61 Base address for reference chroma picture index 5 (SWREG108)

14.1.5.1.61.1 Offset

Register	Offset
SWREG108	1B0h

14.1.5.1.61.2 Diagram



14.1.5.1.61.3 Fields

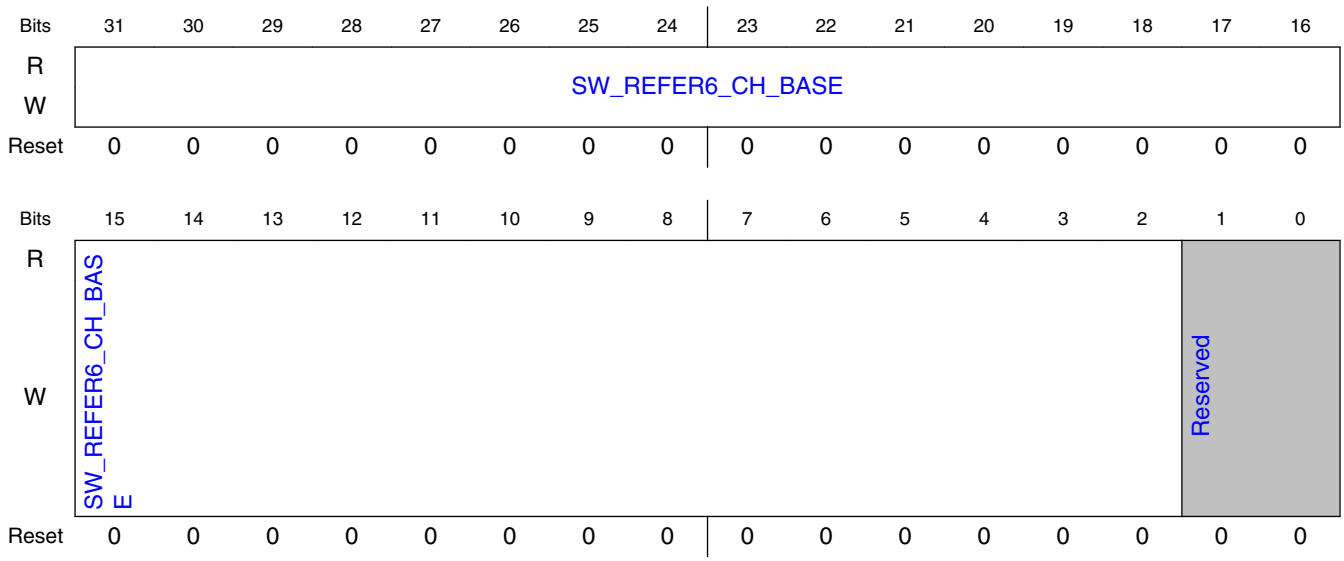
Field	Function
31-2 SW_REFER5_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 5.
1-0 —	Reserved.

14.1.5.1.62 Base address for reference chroma picture index 6 (SWREG109)

14.1.5.1.62.1 Offset

Register	Offset
SWREG109	1B4h

14.1.5.1.62.2 Diagram



14.1.5.1.62.3 Fields

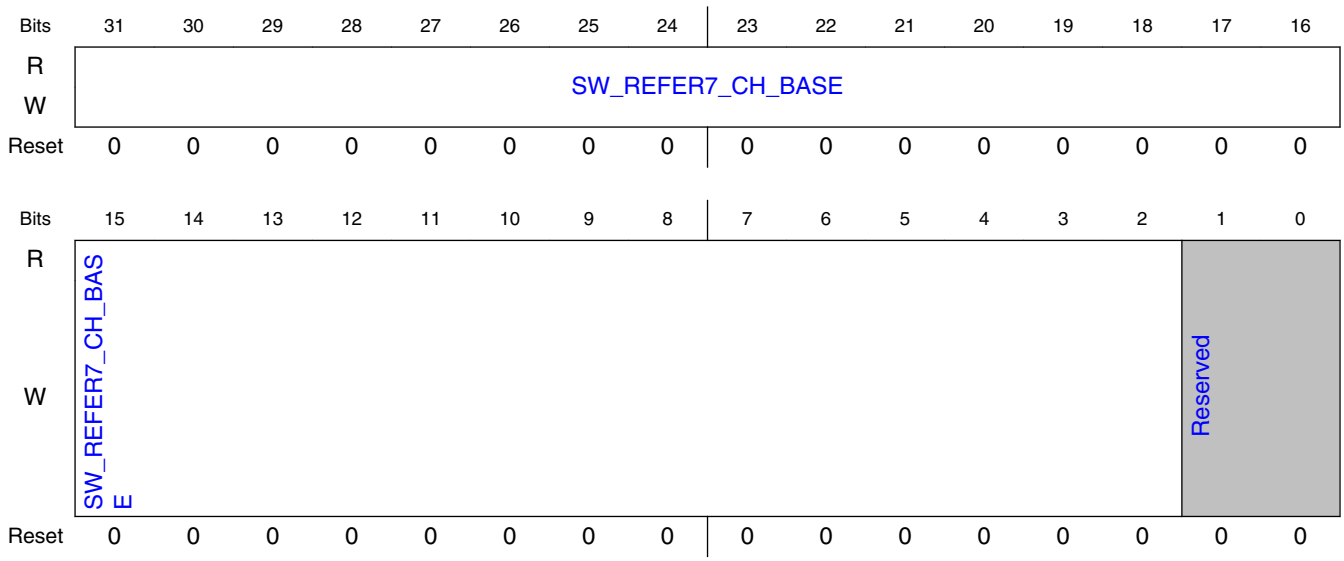
Field	Function
31-2 SW_REFER6_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 6.
1-0 —	Reserved.

14.1.5.1.63 Base address for reference chroma picture index 7 (SWREG110)

14.1.5.1.63.1 Offset

Register	Offset
SWREG110	1B8h

14.1.5.1.63.2 Diagram



14.1.5.1.63.3 Fields

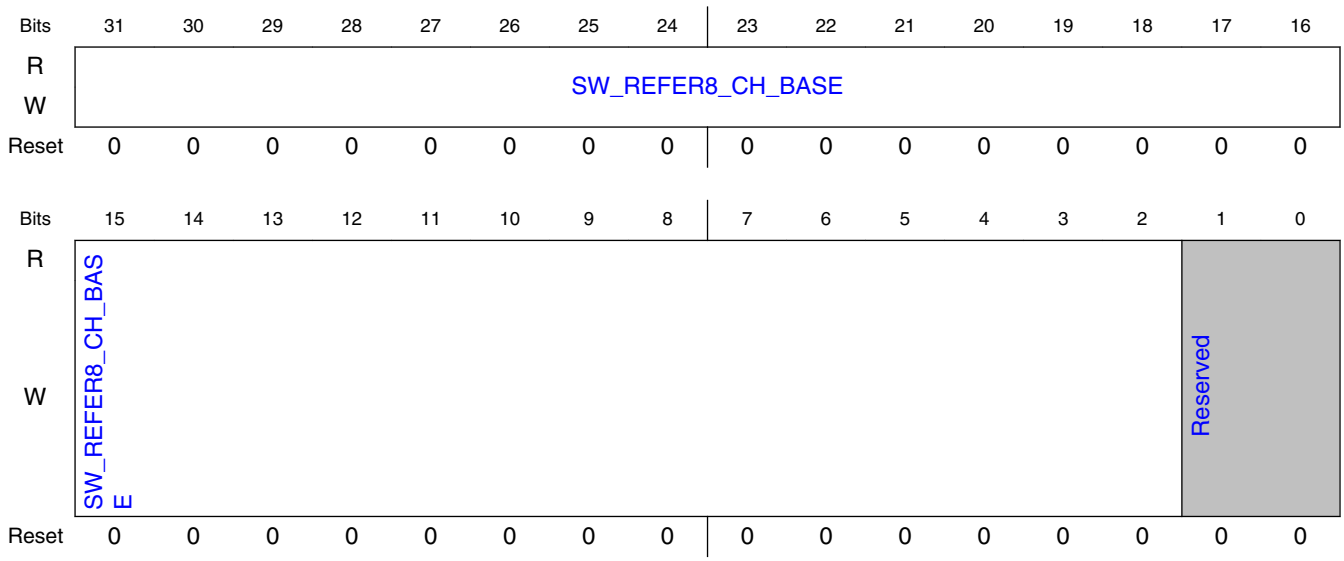
Field	Function
31-2 SW_REFER7_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 7.
1-0 —	Reserved.

14.1.5.1.64 Base address for reference chroma picture index 8 (SWREG111)

14.1.5.1.64.1 Offset

Register	Offset
SWREG111	1BCh

14.1.5.1.64.2 Diagram



14.1.5.1.64.3 Fields

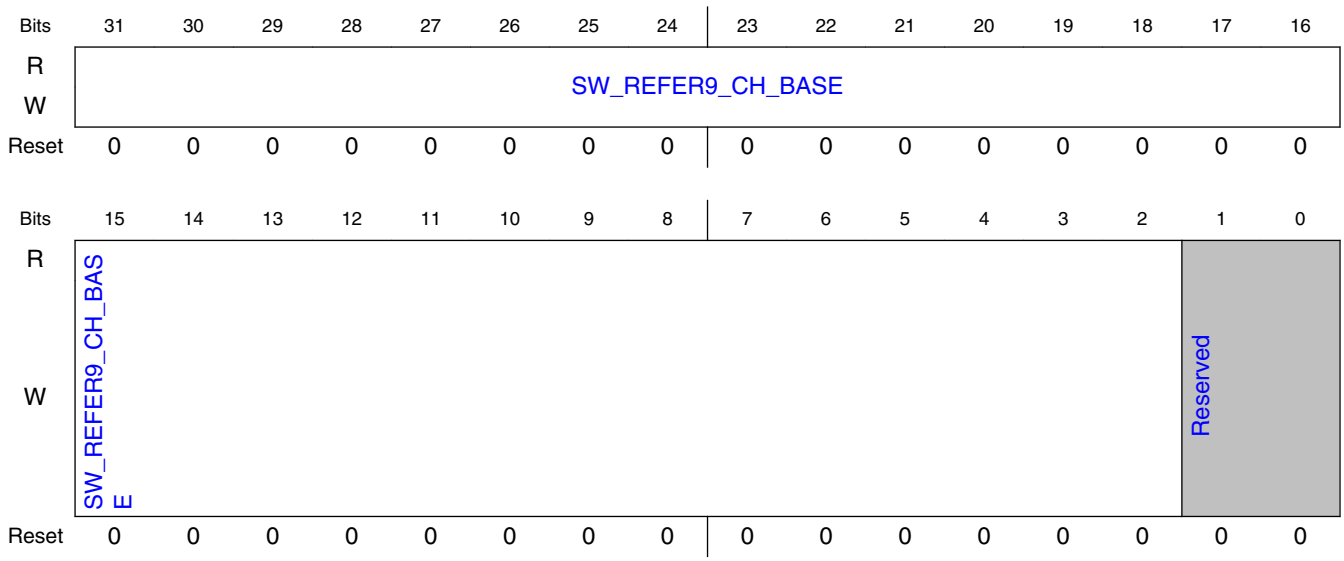
Field	Function
31-2 SW_REFER8_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 8.
1-0 —	Reserved.

14.1.5.1.65 Base address for reference chroma picture index 9 (SWREG112)

14.1.5.1.65.1 Offset

Register	Offset
SWREG112	1C0h

14.1.5.1.65.2 Diagram



14.1.5.1.65.3 Fields

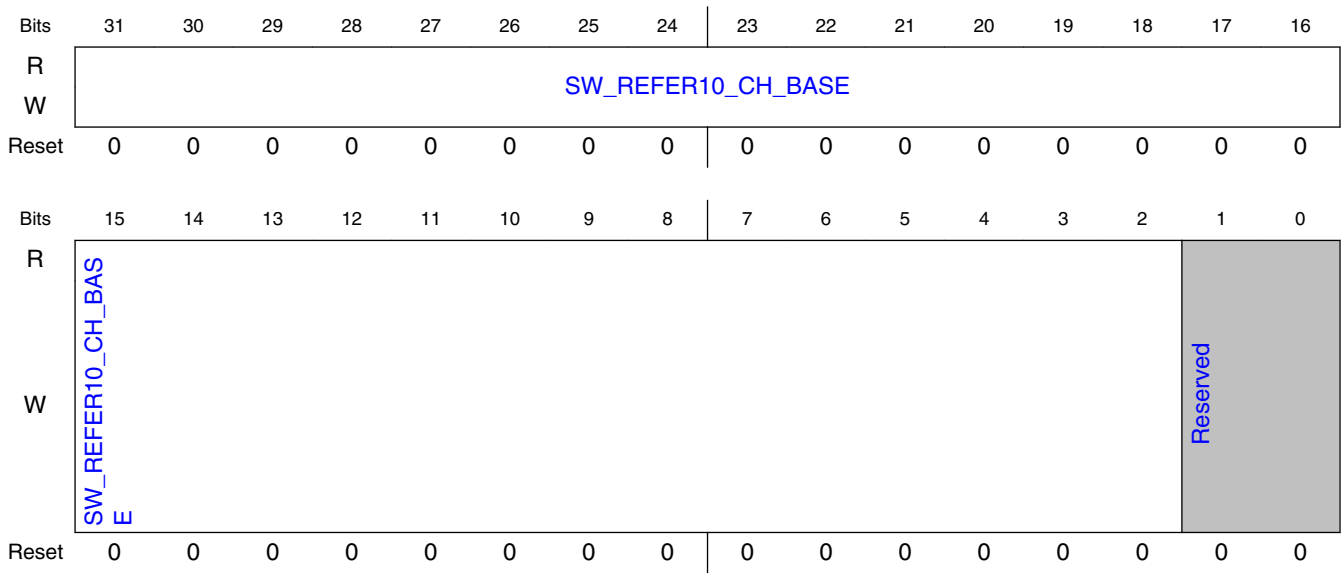
Field	Function
31-2 SW_REFER9_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 9.
1-0 —	Reserved.

14.1.5.1.66 Base address for reference chroma picture index 10 (SWREG113)

14.1.5.1.66.1 Offset

Register	Offset
SWREG113	1C4h

14.1.5.1.66.2 Diagram



14.1.5.1.66.3 Fields

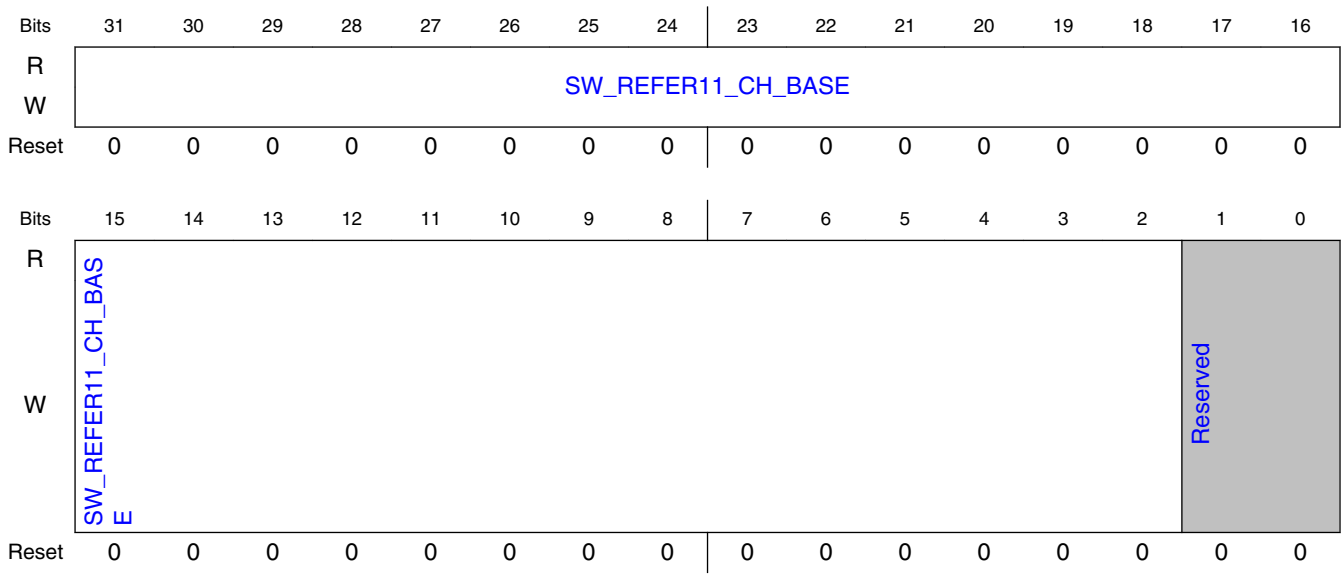
Field	Function
31-2 SW_REFER10_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 10.
1-0 —	Reserved.

14.1.5.1.67 Base address for reference chroma picture index 11 (SWREG114)

14.1.5.1.67.1 Offset

Register	Offset
SWREG114	1C8h

14.1.5.1.67.2 Diagram



14.1.5.1.67.3 Fields

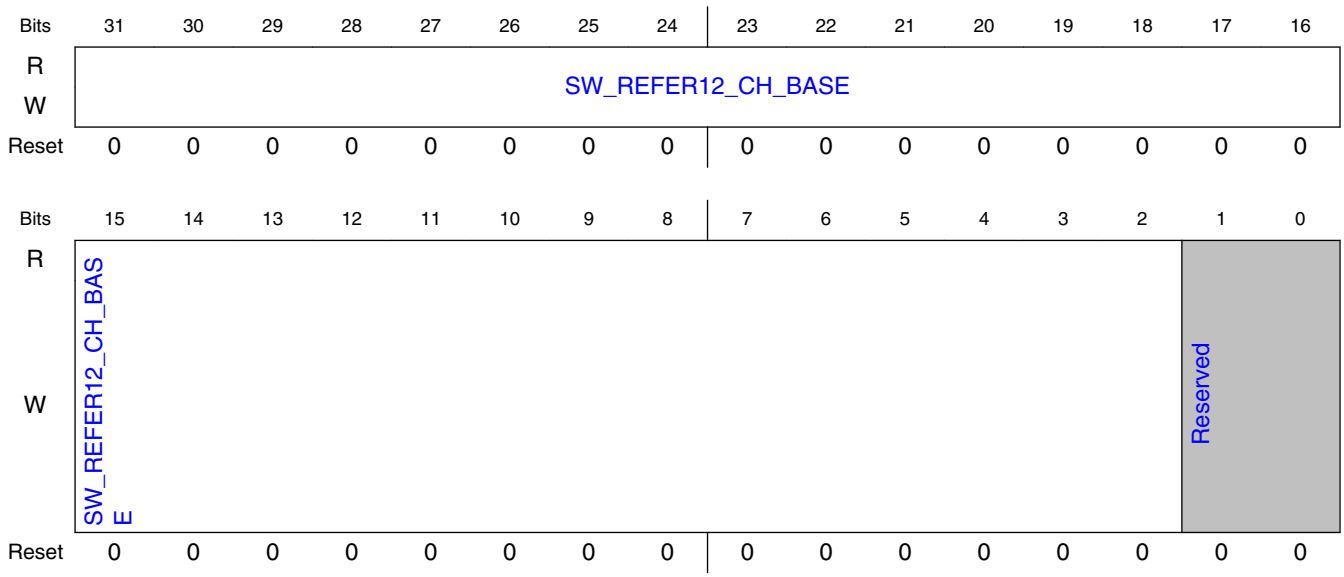
Field	Function
31-2 SW_REFER11_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 11.
1-0 —	Reserved.

14.1.5.1.68 Base address for reference chroma picture index 12 (SWREG115)

14.1.5.1.68.1 Offset

Register	Offset
SWREG115	1CCh

14.1.5.1.68.2 Diagram



14.1.5.1.68.3 Fields

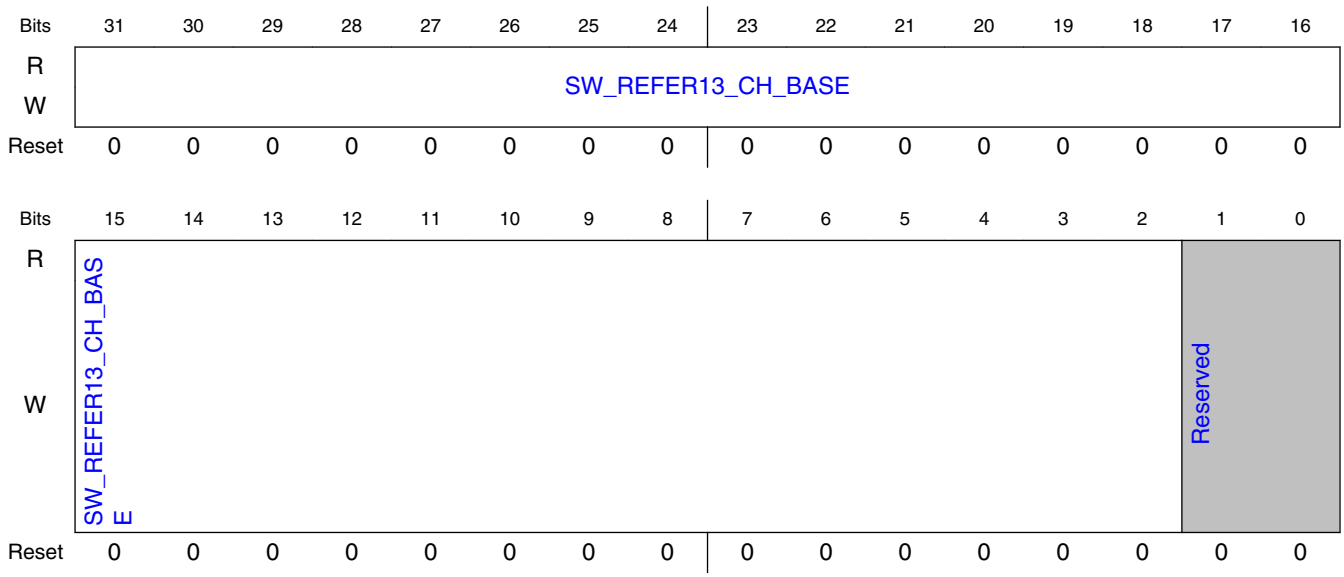
Field	Function
31-2 SW_REFER12_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 12.
1-0 —	Reserved.

14.1.5.1.69 Base address for reference chroma picture index 13 (SWREG116)

14.1.5.1.69.1 Offset

Register	Offset
SWREG116	1D0h

14.1.5.1.69.2 Diagram



14.1.5.1.69.3 Fields

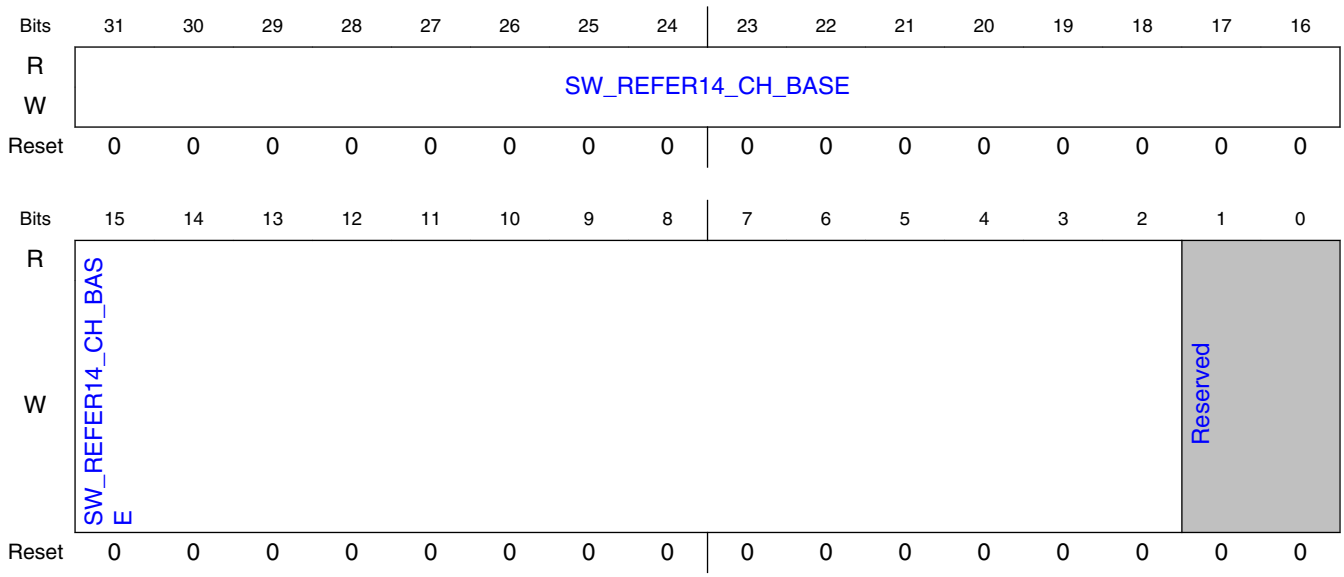
Field	Function
31-2 SW_REFER13_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 13.
1-0 —	Reserved.

14.1.5.1.70 Base address for reference chroma picture index 14 (SWREG117)

14.1.5.1.70.1 Offset

Register	Offset
SWREG117	1D4h

14.1.5.1.70.2 Diagram



14.1.5.1.70.3 Fields

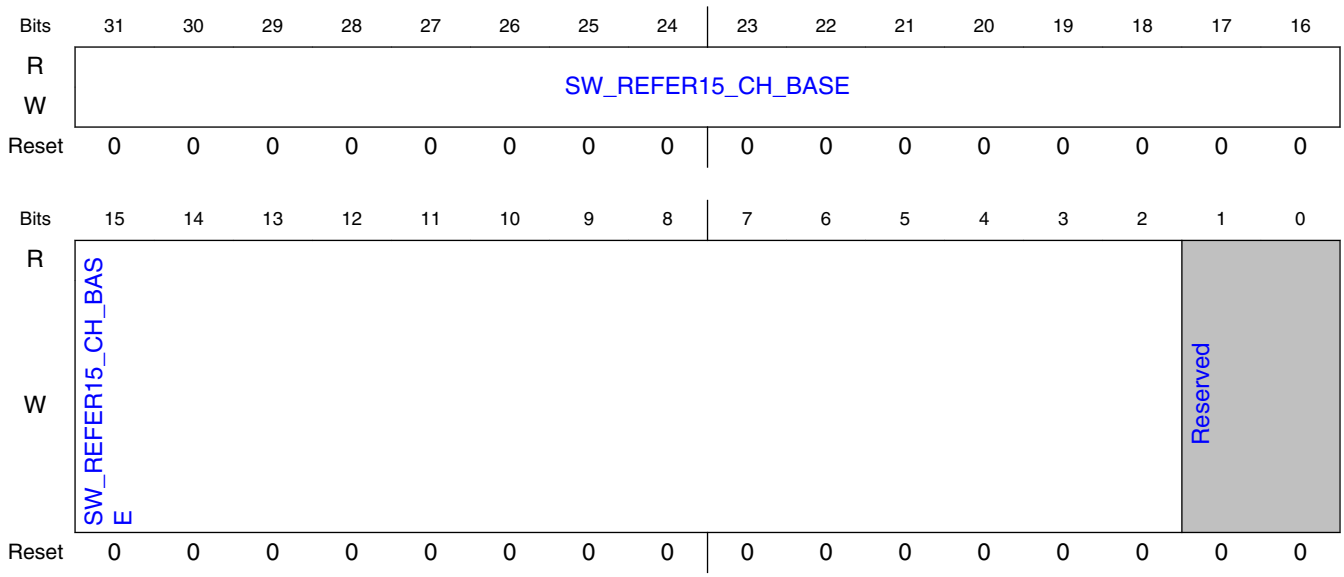
Field	Function
31-2 SW_REFER14_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 14.
1-0 —	Reserved.

14.1.5.1.71 Base address for reference chroma picture index 15 (SWREG118)

14.1.5.1.71.1 Offset

Register	Offset
SWREG118	1D8h

14.1.5.1.71.2 Diagram



14.1.5.1.71.3 Fields

Field	Function
31-2 SW_REFER15_CH_BASE	Valid only if chroma address separate mode is enabled. Base address for reference chroma picture index 15.
1-0 —	Reserved.

14.1.5.2 VPU_G1 H264 decoder register descriptions

14.1.5.2.1 VPU_G1 H264 decoder memory map

VPU_G1_H264 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Decoder control register 1 (picture parameters) (SWREG4)	32	RW	0000_0000h
14h	Decoder control register 2 (stream decoding table selects) (SWREG5)	32	RW	0000_0000h
18h	Decoder control register 3 (stream buffer information) (SWREG6)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1Ch	Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7)	32	RW	0000_0000h
20h	Decoder control register 5 (H264, VC-1, VP6, Progressive JPEG and RV control) (SWREG8)	32	RW	0000_0000h
24h	Decoder control register 6 / base address for MB-control (RLC) / VC-1 intensity control 0/ VP6,VP7,VP8 ctrl-stream length/ RV pic slice amount (SWREG9)	32	RW	0000_0000h
28h	Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10)	32	RW	0000_0000h
2Ch	Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11)	32	RW	0000_0000h
38h	Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14)	32	RW	0000_0000h
3Ch	Base address for reference picture index 1 / JPEG control (SWRE G15)	32	RW	0000_0000h
40h	Base address for reference picture index 2 / List of VLC code lengths in first JPEG AC table (SWREG16)	32	RW	0000_0000h
44h	Base address for reference picture index 3 / List of VLC code lengths in first JPEG AC table (SWREG17)	32	RW	0000_0000h
48h	Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWRE G18)	32	RW	0000_0000h
4Ch	Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19)	32	RW	0000_0000h
50h	Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20)	32	RW	0000_0000h
54h	Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21)	32	RW	0000_0000h
58h	Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22)	32	RW	0000_0000h
5Ch	Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23)	32	RW	0000_0000h
60h	Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24)	32	RW	0000_0000h
64h	Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25)	32	RW	0000_0000h

Table continues on the next page...

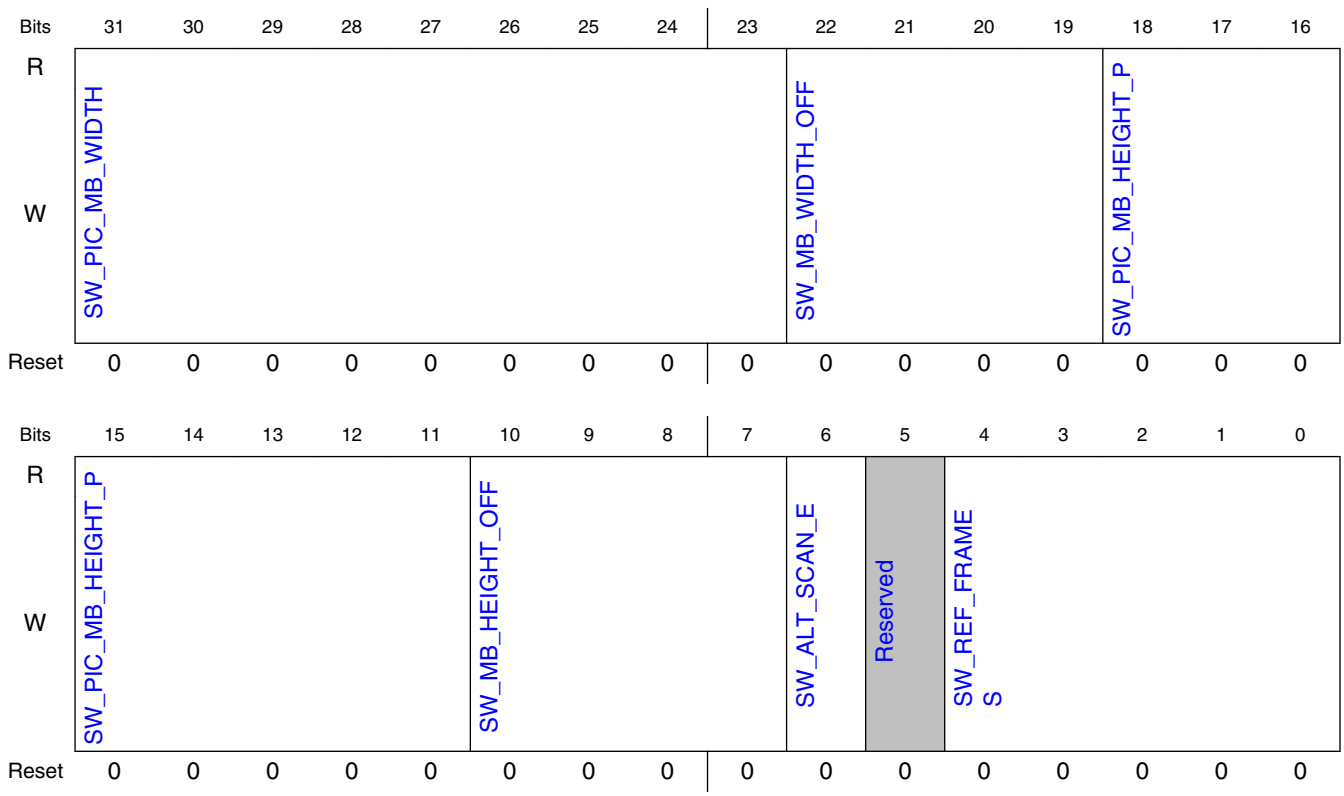
Offset	Register	Width (In bits)	Access	Reset value
68h	Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26)	32	RW	0000_0000h
6Ch	Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27)	32	RW	0000_0000h
70h	Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28)	32	RW	0000_0000h
74h	Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29)	32	RW	0000_0000h
78h	Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30)	32	RW	0000_0000h
7Ch	Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31)	32	RW	0000_0000h
80h	Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32)	32	RW	0000_0000h
84h	Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33)	32	RW	0000_0000h
88h	Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34)	32	RW	0000_0000h
8Ch	Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35)	32	RW	0000_0000h
90h	Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36)	32	RW	0000_0000h
94h	Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37)	32	RW	0000_0000h
98h	Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38)	32	RW	0000_0000h
9Ch	Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39)	32	RW	0000_0000h
A8h	bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_H264)	32	RW	0000_0000h
ACh	bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_H264)	32	RW	0000_0000h
B0h	bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_H264)	32	RW	0000_0000h
B4h	bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45)	32	RW	0000_0000h
B8h	bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46)	32	RW	0000_0000h
BCh	bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47)	32	RW	0000_0000h

14.1.5.2.2 Decoder control register 1 (picture parameters) (SWREG4)

14.1.5.2.2.1 Offset

Register	Offset
SWREG4	10h

14.1.5.2.2.2 Diagram



14.1.5.2.2.3 Fields

Field	Function
31-23 SW_PIC_MB_WIDTH	Picture width in macroblocks = ((width in pixels + 15) /16)
22-19 SW_MB_WIDTH_OFF	The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull
18-11	Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded

Table continues on the next page...

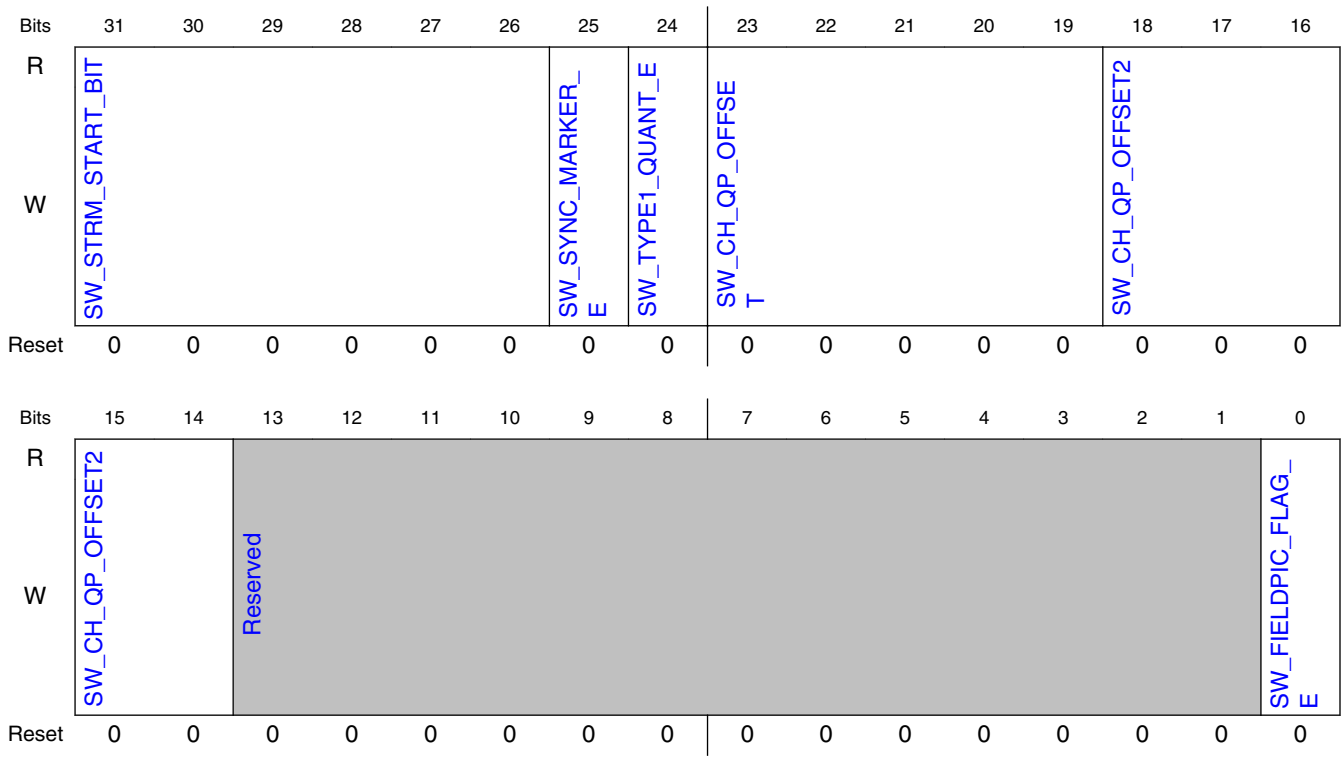
Field	Function
SW_PIC_MB_HEIGHT_P	
10-7 SW_MB_HEIGHT_OFF	The amount of meaningful vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull
6 SW_ALT_SCAN_E	indicates alternative vertical scan method used for interlaced frames
5 —	Reserved.
4-0 SW_REF_FRAMES	H.264: num_ref_frames, maximum number of short and long term reference frames in decoded picture buffer VC-1: num_ref semantics

14.1.5.2.3 Decoder control register 2 (stream decoding table selects) (SWREG5)

14.1.5.2.3.1 Offset

Register	Offset
SWREG5	14h

14.1.5.2.3.2 Diagram



14.1.5.2.3.3 Fields

Field	Function
31-26 SW_STRM_START_BIT	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
25 SW_SYNC_MARKER_E	Sync markers enable For progressive JPEG, this indicates that there are restart markers in the stream after restart interval steps. 0b - synch markers are not used 1b - synch markers are used.
24 SW_TYPE1_QUANT_E	MPEG4: Type 1 quantization enable • '0' = type 2 inverse Q method • '1' = type 1 inverse Q method (Q-tables used) H264 (h264_high config): scaling matrix enable • '0' = normal transform • '1' = use scaling matrix for transform (read from external memory)
23-19 SW_CH_QP_OFFSET	Chroma Qp filter offset. (For H.264 this offset concerns Cb only)
18-14	Chroma Qp filter offset for cr type

Table continues on the next page...

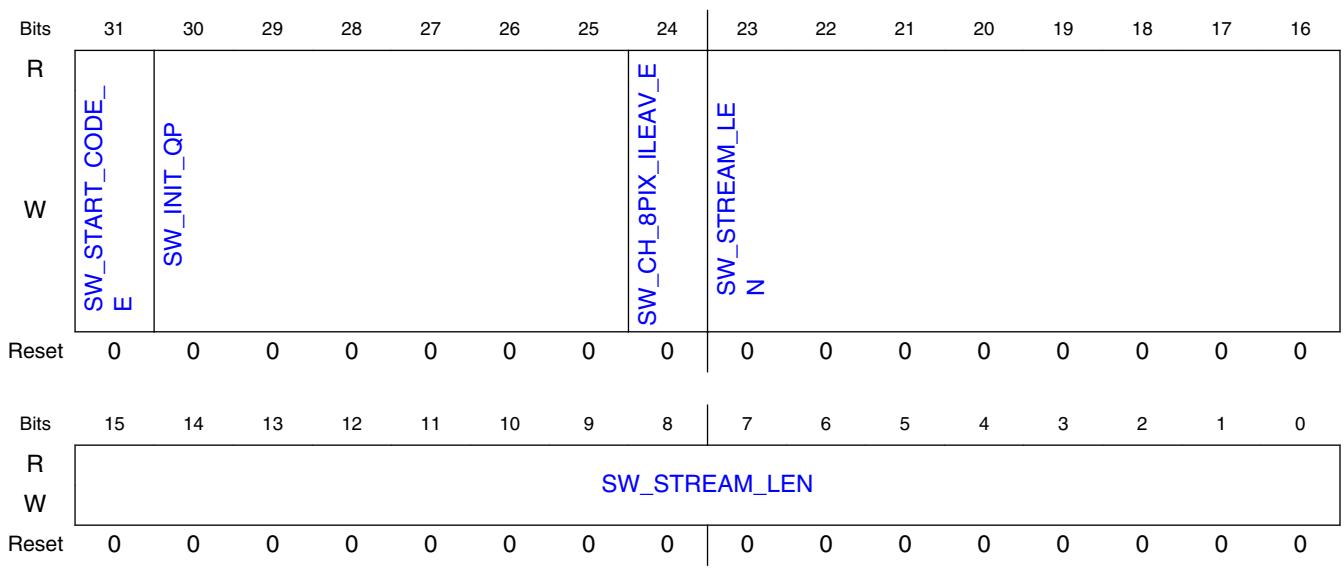
Field	Function
SW_CH_QP_OFFSET2	
13-1 —	Reserved.
0 SW_FIELDPIC_FLAG_E	Flag for streamd that field_pic_flag exists in stream

14.1.5.2.4 Decoder control register 3 (stream buffer information) (SWREG6)

14.1.5.2.4.1 Offset

Register	Offset
SWREG6	18h

14.1.5.2.4.2 Diagram



14.1.5.2.4.3 Fields

Field	Function
31	Bit for indicating stream start code existence: 0b - stream doesn't contain start codes

Table continues on the next page...

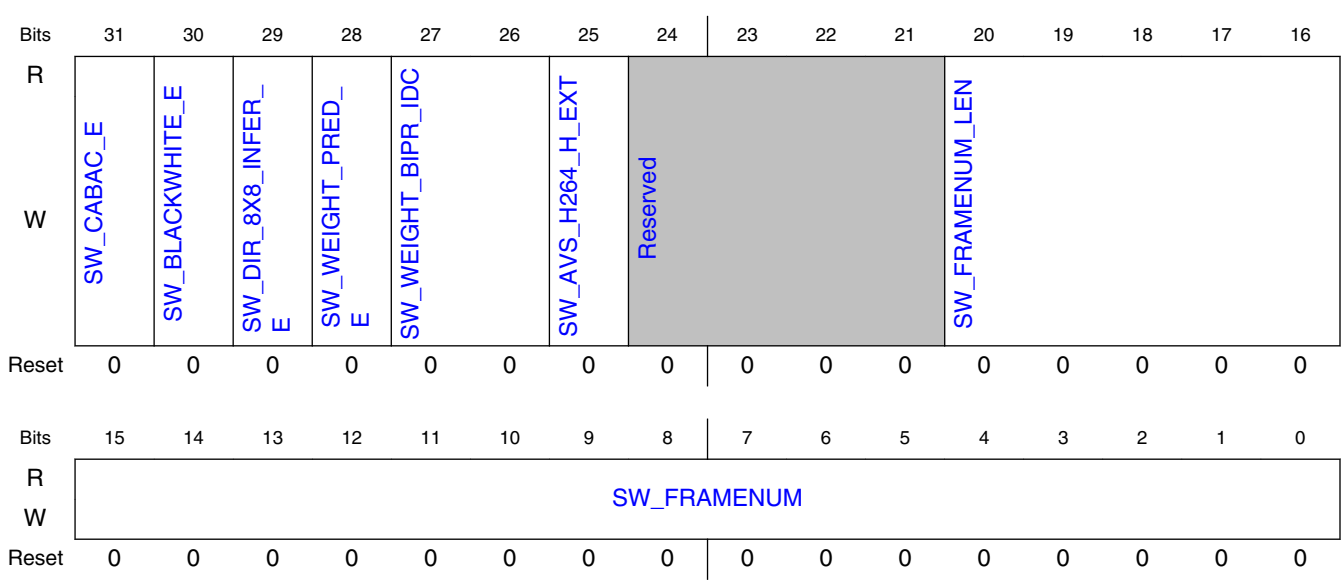
Field	Function
SW_START_CODE_E	1b - stream contains start codes
30-25 SW_INIT_QP	Initial value for quantization parameter (picture quantizer).
24 SW_CH_8PIX_ILEAV_E	Enable for additional chrominance data format writing where decoder writes chrominance in group of 8 pixels of Cb and then corresponding 8 pixels of Cr. Data is written to sw_dec_ch8pix_base. Cannot be used if tiled mode is enabled
23-0 SW_STREAM_LEN	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For VC-1/VP6 the buffer must include data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice/VP of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture.

14.1.5.2.5 Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7)

14.1.5.2.5.1 Offset

Register	Offset
SWREG7	1Ch

14.1.5.2.5.2 Diagram



14.1.5.2.5.3 Fields

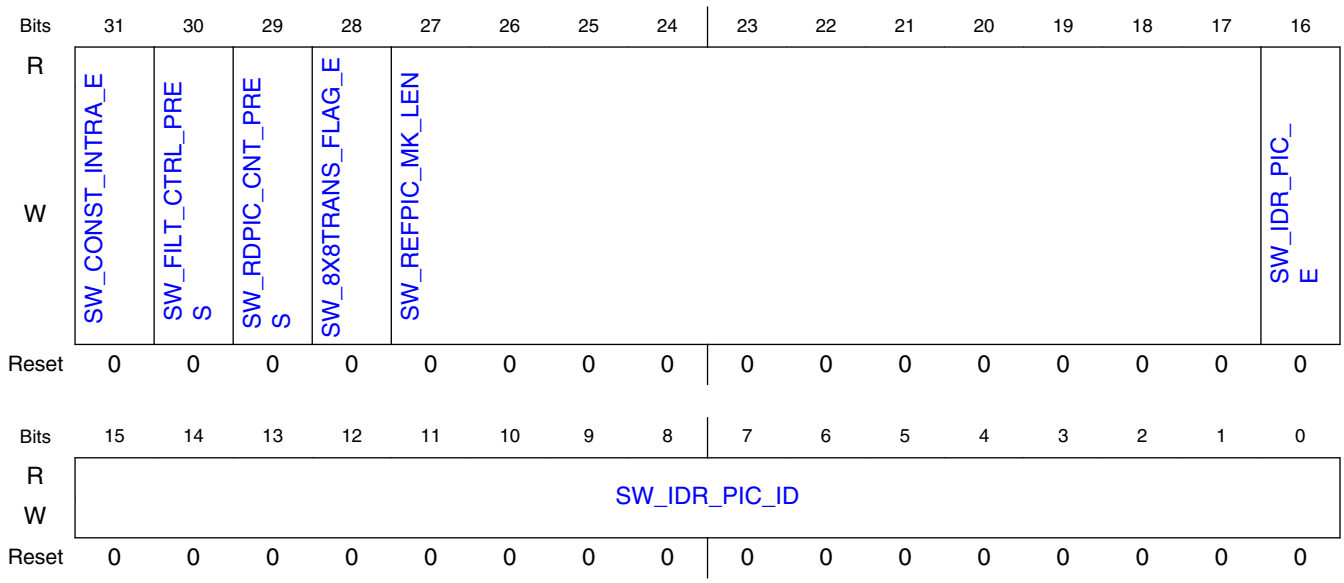
Field	Function
31 SW_CABAC_E	CABAC enable
30 SW_BLACKWHITE_E	0b - 4:2:0 sampling format 1b - 4:0:0 sampling format (H264 monochroma)
29 SW_DIR_8X8_INFER_E	Specifies the method to use to derive luma motion vectors in B_skip, B_Direct_16x16 and B_direct_8x8_inference_flag (see direct_8x8_inference flag)
28 SW_WEIGHT_PRED_E	Weighted prediction enable for P slices
27-26 SW_WEIGHT_BIPR_IDC	weighted prediction specification for B slices: 00b - default weighted prediction is applied to B slices 01b - explicit weighted prediction shall be applied to B slices 10b - implicit weighted prediction shall be applied to B slices
25 SW_AVS_H264_H_EXT	Resolution extension to support 4k resolution for AVS/H264. Used as MSB of sw_pic_mb_height
24-21 —	Reserved.
20-16 SW_FRAMENUM_LEN	H.264: Bit length of frame_num in data stream RV: frame size length. Informs how many bits in stream are used for frame size (HW discards these bits)
15-0 SW_FRAMENUM	current frame_num, used to identify short-term reference frames. Used in reference picture reordering

14.1.5.2.6 Decoder control register 5 (H264, VC-1, VP6, Progressive JPEG and RV control) (SWREG8)

14.1.5.2.6.1 Offset

Register	Offset
SWREG8	20h

14.1.5.2.6.2 Diagram



14.1.5.2.6.3 Fields

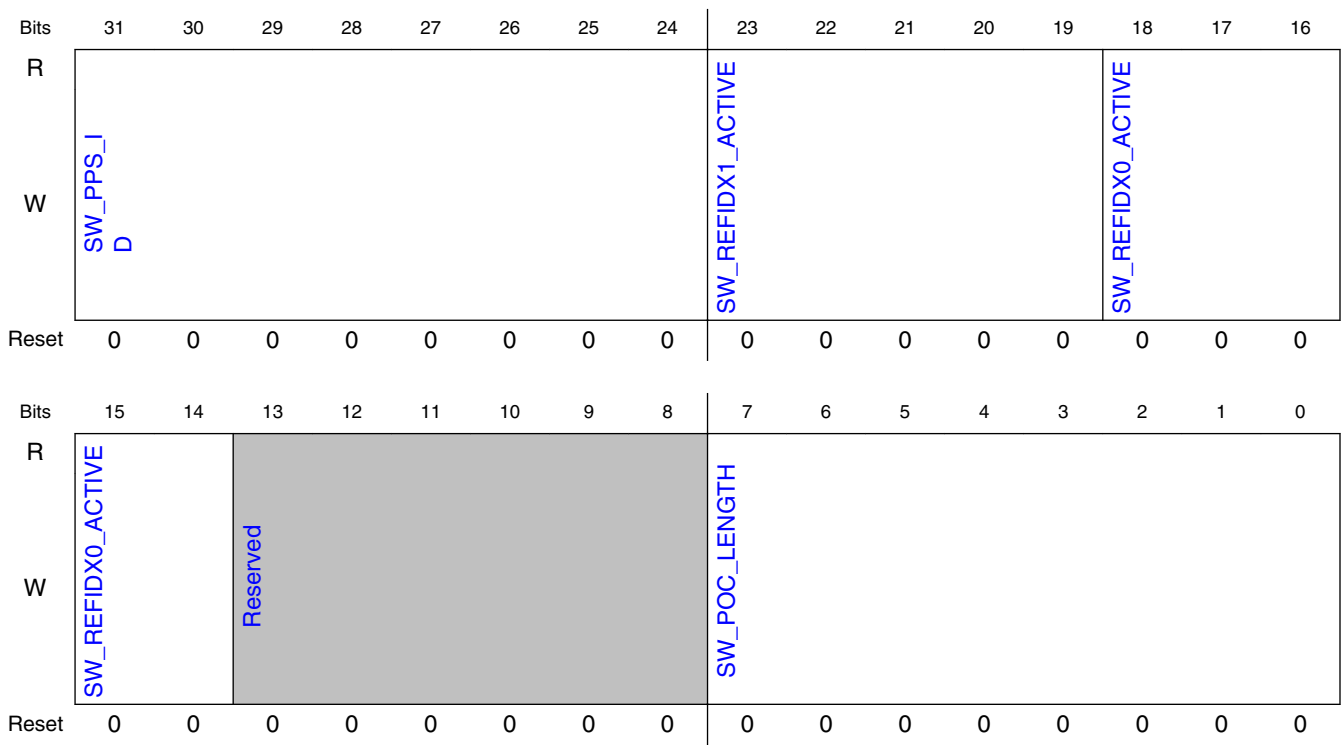
Field	Function
31 SW_CONST_INTRA_E	constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process.
30 SW_FILT_CTRL_PRES	deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.
29 SW_RDPIC_CNT_PRES	redundant_pic_cnt_present_flag specifies whether redundant_pic_cnt syntax elements are present in the slice header.
28 SW_8X8TRANS_FLAG_E	8x8 transform flag enable for stream decoding
27-17 SW_REFPIC_MK_LEN	Length of decoded reference picture marking bits
16 SW_IDR_PIC_E	IDR (instantaneous decoding refresh) picture flag.
15-0 SW_IDR_PIC_ID	idr_pic_id, identifies IDR (instantaneous decoding refresh) picture

14.1.5.2.7 Decoder control register 6 / base address for MB-control (RLC) / VC-1 intensity control 0/ VP6,VP7,VP8 ctrl-stream length/ RV pic slice amount (SWREG9)

14.1.5.2.7.1 Offset

Register	Offset
SWREG9	24h

14.1.5.2.7.2 Diagram



14.1.5.2.7.3 Fields

Field	Function
31-24 SW_PPS_ID	pic_parameter_set_id, identifies the picture parameter set that is referred to in the slice header.
23-19 SW_REFIDX1_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks.

Table continues on the next page...

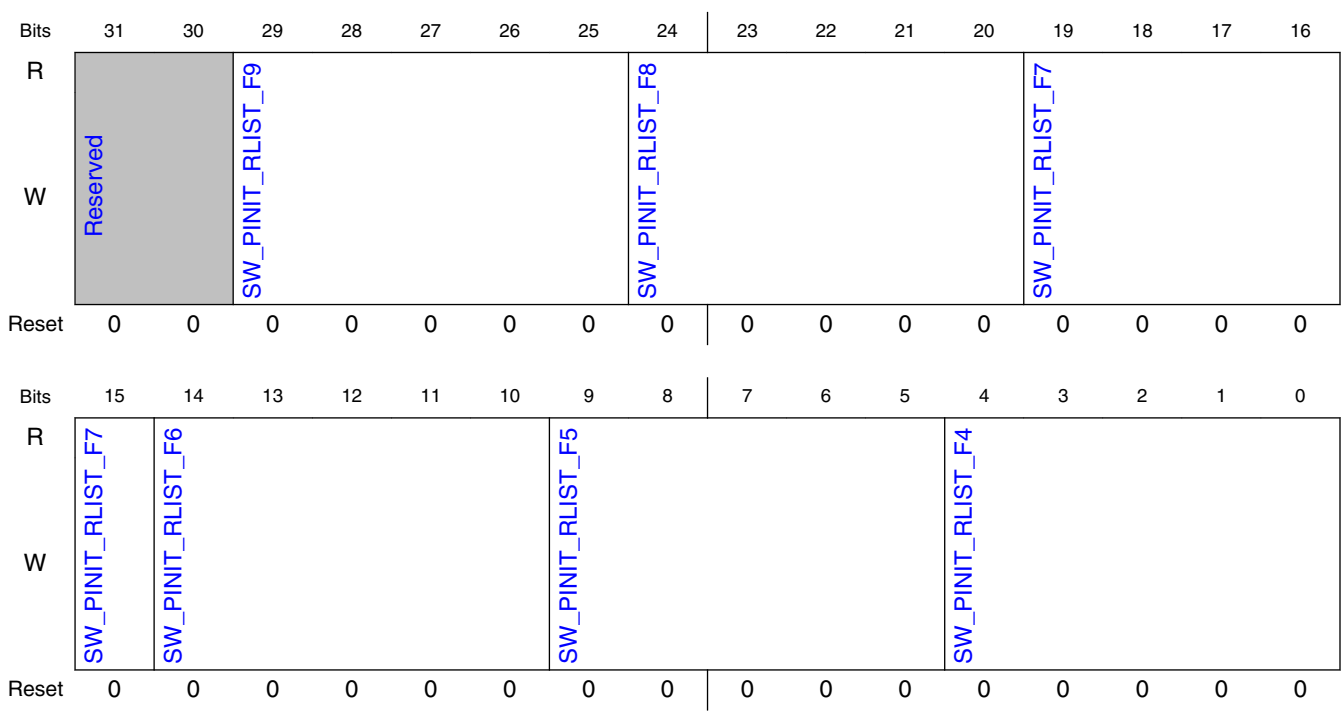
Field	Function
18-14 SW_REFIDX0_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit)
13-8 —	Reserved.
7-0 SW_POC_LENGTH	Length of picture order count field in stream

14.1.5.2.8 Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10)

14.1.5.2.8.1 Offset

Register	Offset
SWREG10	28h

14.1.5.2.8.2 Diagram



14.1.5.2.8.3 Fields

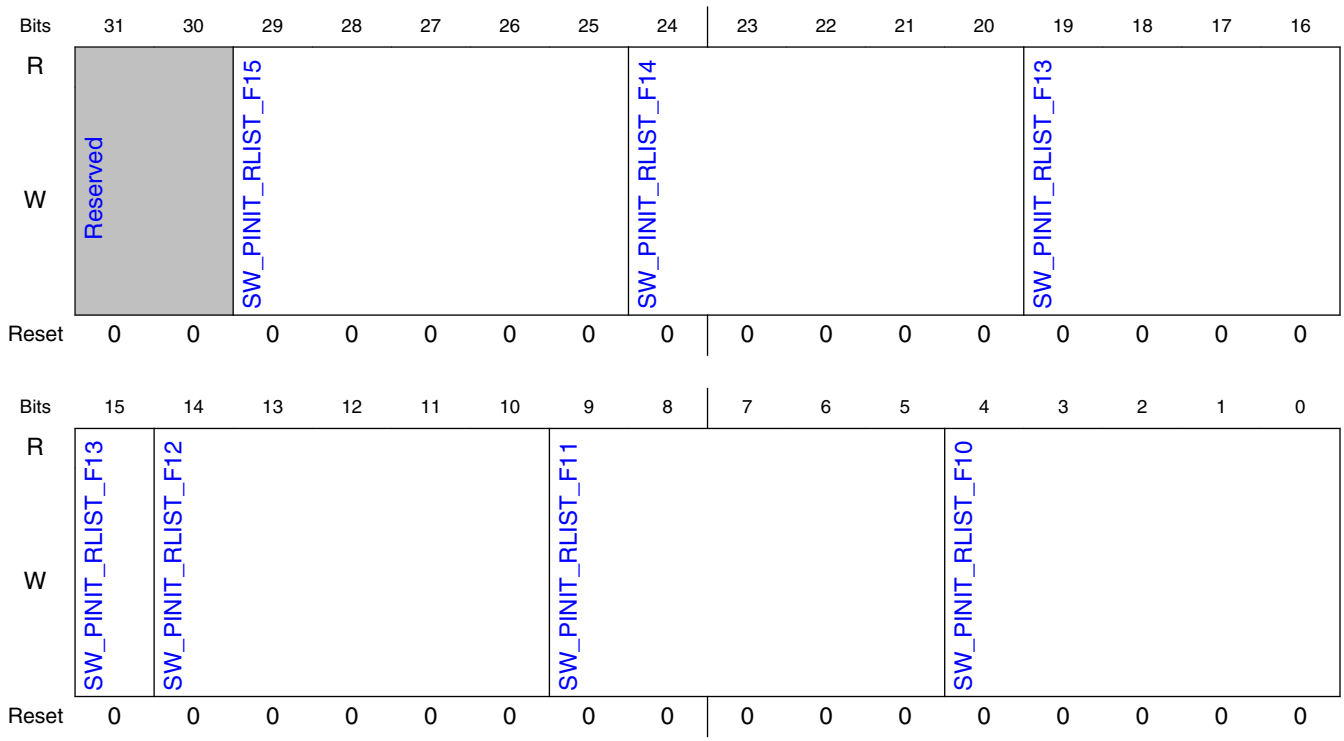
Field	Function
31-30 —	Reserved.
29-25 SW_PINIT_RLIST_F9	Initial reference picture list for P forward picid 9
24-20 SW_PINIT_RLIST_F8	Initial reference picture list for P forward picid 8
19-15 SW_PINIT_RLIST_F7	Initial reference picture list for P forward picid 7
14-10 SW_PINIT_RLIST_F6	Initial reference picture list for P forward picid 6
9-5 SW_PINIT_RLIST_F5	Initial reference picture list for P forward picid 5
4-0 SW_PINIT_RLIST_F4	Initial reference picture list for P forward picid 4

14.1.5.2.9 Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11)

14.1.5.2.9.1 Offset

Register	Offset
SWREG11	2Ch

14.1.5.2.9.2 Diagram



14.1.5.2.9.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_PINIT_RLIST_F15	Initial reference picture list for P forward picid 15
24-20 SW_PINIT_RLIST_F14	Initial reference picture list for P forward picid 14
19-15 SW_PINIT_RLIST_F13	Initial reference picture list for P forward picid 13
14-10 SW_PINIT_RLIST_F12	Initial reference picture list for P forward picid 12
9-5 SW_PINIT_RLIST_F11	Initial reference picture list for P forward picid 11
4-0	Initial reference picture list for P forward picid 10

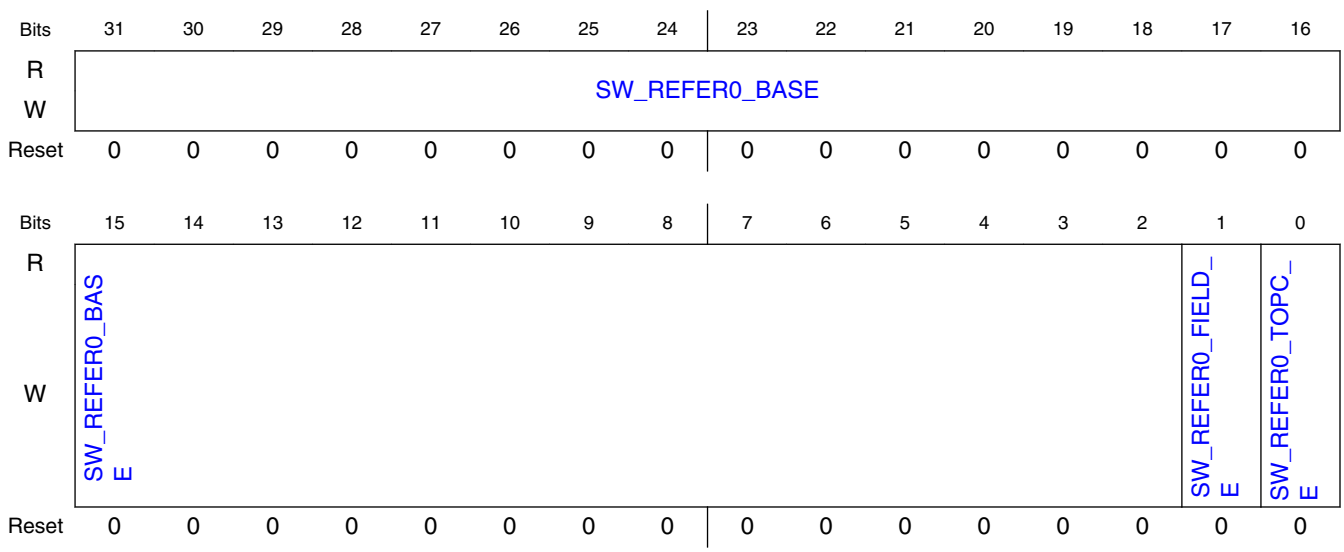
Field	Function
SW_PINIT_RLIST_F10	

14.1.5.2.10 Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14)

14.1.5.2.10.1 Offset

Register	Offset
SWREG14	38h

14.1.5.2.10.2 Diagram



14.1.5.2.10.3 Fields

Field	Function
31-2 SW_REFER0_BASE	Base address for reference picture index 0. See picture index definition from toplevel_sp
1 SW_REFER0_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0	Which field of reference picture is closer to current picture:

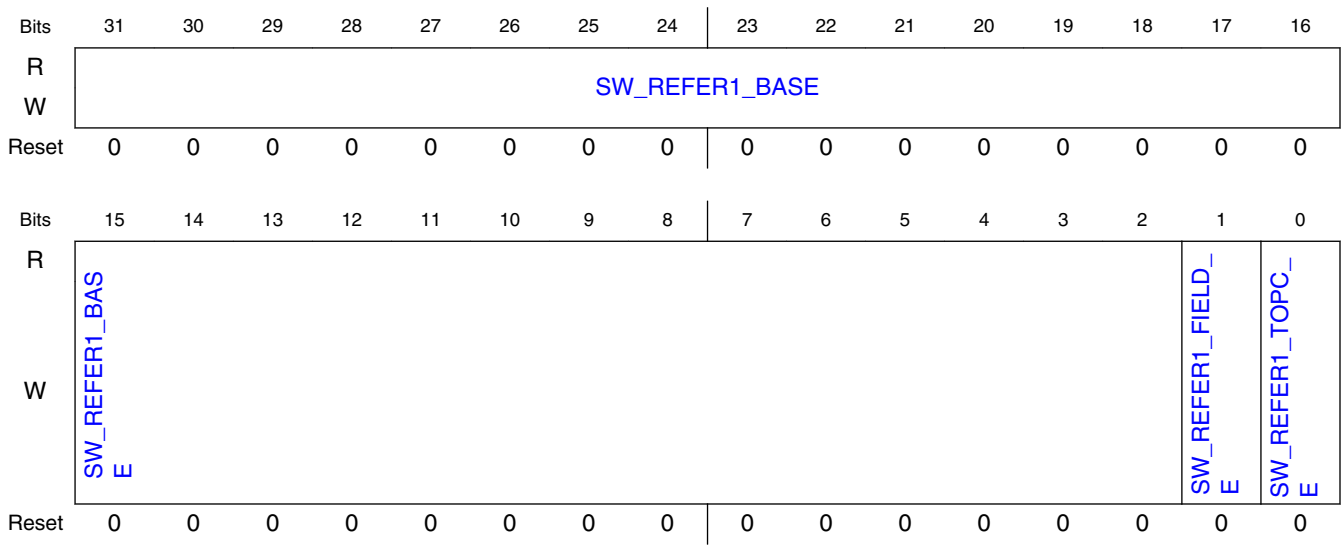
Field	Function
SW_REFER0_TOPC_E	0b - Bottom field is closer to current picture 1b - Top field is closer to current picture

14.1.5.2.11 Base address for reference picture index 1 / JPEG control (SWREG15)

14.1.5.2.11.1 Offset

Register	Offset
SWREG15	3Ch

14.1.5.2.11.2 Diagram



14.1.5.2.11.3 Fields

Field	Function
31-2 SW_REFER1_BASE	Base address for reference picture index 1. See picture index definition from toplevel_sp. For VP8 this base address is used as Chrominance base address for reference picture 0 (if vp8 stride configuration is enabled)
1 SW_REFER1_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0	Which field of reference picture is closer to current picture:

Field	Function
SW_REFER1_TOPC_E	0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.12 Base address for reference picture index 2 / List of VLC code lengths in first JPEG AC table (SWREG16)

14.1.5.2.12.1 Offset

Register	Offset
SWREG16	40h

14.1.5.2.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_BASE														SW_REFER2_FIELD_E	SW_REFER2_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.12.3 Fields

Field	Function
31-2 SW_REFER2_BASE	Base address for reference picture index 2. See picture index definition from toplevel_sp. For VP8 video this base address is used as Golden reference chrominance base address (if vp8 stride configuration is enabled)
1 SW_REFER2_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0	Which field of reference picture is closer to current picture:

Field	Function
SW_REFER2_TOPC_E	0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.13 Base address for reference picture index 3 / List of VLC code lengths in first JPEG AC table (SWREG17)

14.1.5.2.13.1 Offset

Register	Offset
SWREG17	44h

14.1.5.2.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_BASE														SW_REFER3_FIELD_E	SW_REFER3_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.13.3 Fields

Field	Function
31-2 SW_REFER3_BASE	Base address for reference picture index 3. See picture index definition from toplevel_sp. For VP8 video this base address is used as Alternate reference chrominance base address (if vp8 stride configuration is enabled)
1 SW_REFER3_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0	Which field of reference picture is closer to current picture:

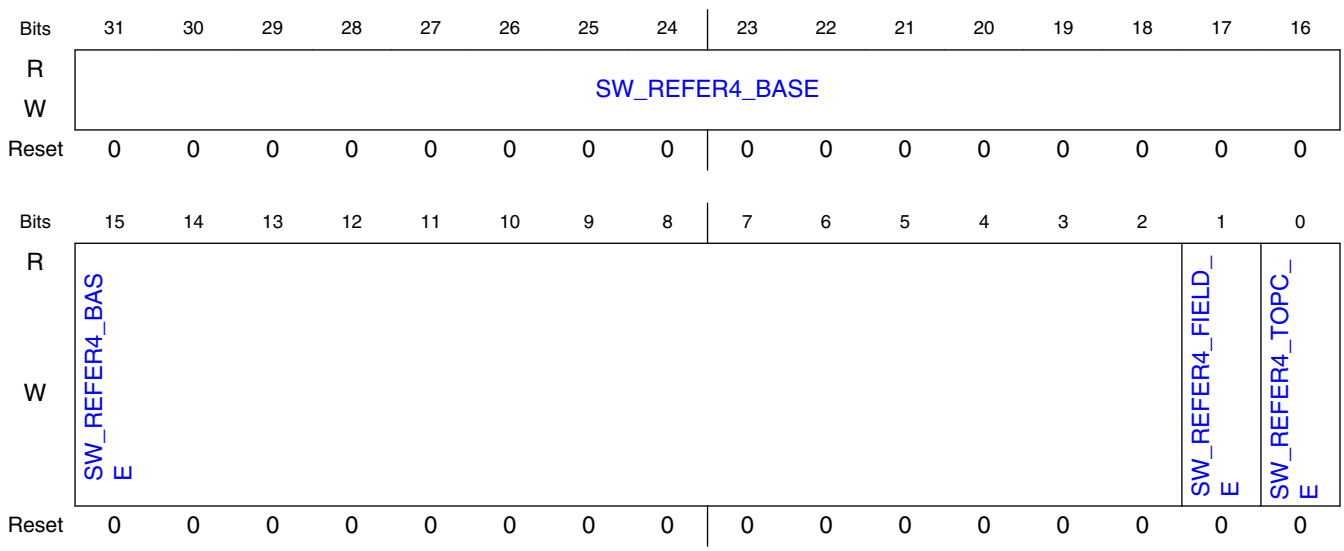
Field	Function
SW_REFER3_TOPC_E	0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.14 Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18)

14.1.5.2.14.1 Offset

Register	Offset
SWREG18	48h

14.1.5.2.14.2 Diagram



14.1.5.2.14.3 Fields

Field	Function
31-2 SW_REFER4_BASE	H264: Base address for reference picture index 4 VP6/VP7/VP8: Base address for Golden reference picture (corresponds picid 4)
1	Refer picture consist of single fields or frame: 0b - reference picture consists of frame

Table continues on the next page...

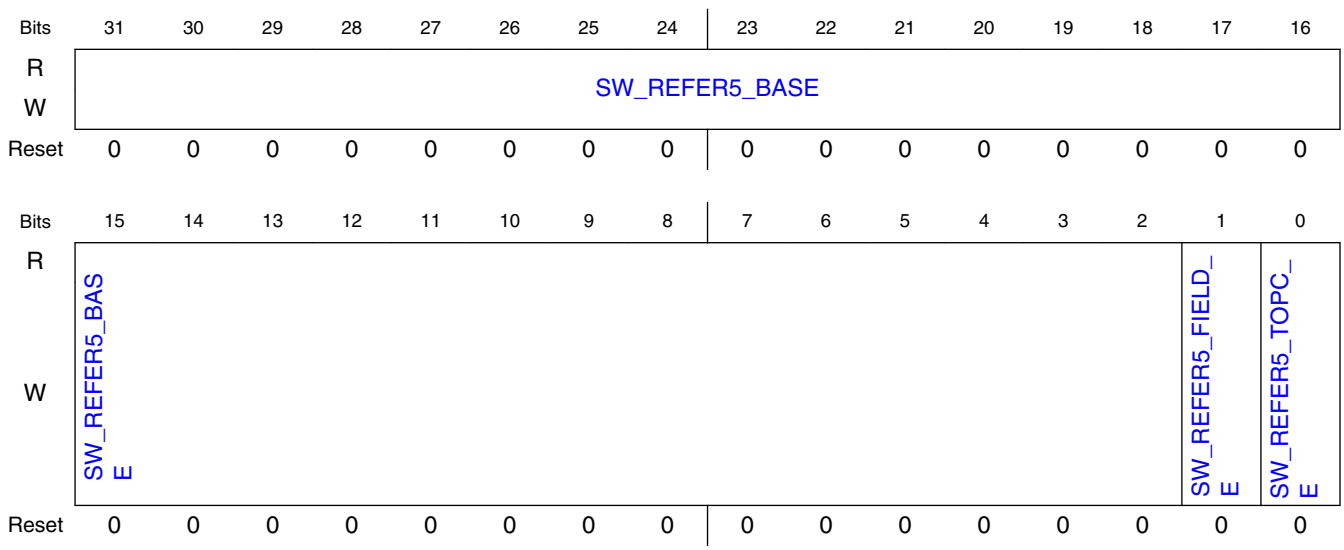
Field	Function
SW_REFER4_FIELD_E	1b - reference picture consists of fields
0 SW_REFER4_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.15 Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19)

14.1.5.2.15.1 Offset

Register	Offset
SWREG19	4Ch

14.1.5.2.15.2 Diagram



14.1.5.2.15.3 Fields

Field	Function
31-2 SW_REFER5_BASE	H.264: Base address for reference picture index 5 VP8: Base address for alternate reference picture (corresponds picid 5)

Table continues on the next page...

Field	Function
1 SW_REFER5_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER5_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.16 Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20)

14.1.5.2.16.1 Offset

Register	Offset
SWREG20	50h

14.1.5.2.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_BASE														SW_REFER6_FIELD_E	SW_REFER6_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.16.3 Fields

Field	Function
31-2	Base address for reference picture index 6. For VP8 video this base address is used as decoder output chrominance base address (if vp8 stride configuration is enabled)

Table continues on the next page...

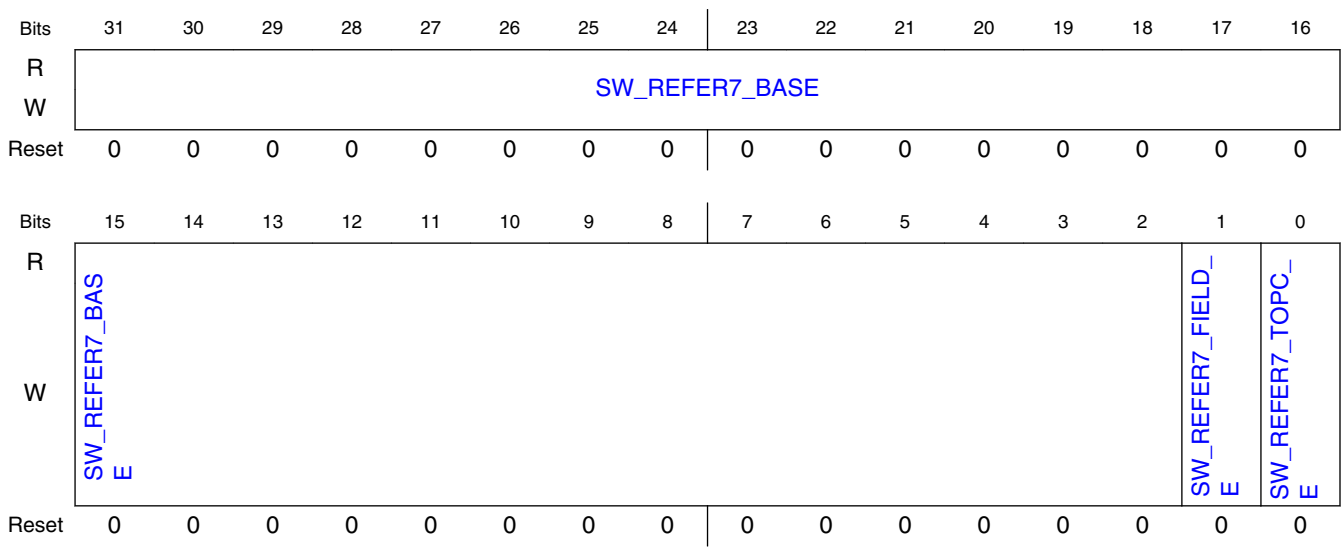
Field	Function
SW_REFER6_BASE	
1 SW_REFER6_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER6_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.17 Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21)

14.1.5.2.17.1 Offset

Register	Offset
SWREG21	54h

14.1.5.2.17.2 Diagram



14.1.5.2.17.3 Fields

Field	Function
31-2	Base address for reference picture index 7

Table continues on the next page...

Field	Function
SW_REFER7_BASE	
1 SW_REFER7_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER7_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.18 Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22)

14.1.5.2.18.1 Offset

Register	Offset
SWREG22	58h

14.1.5.2.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_BASE														SW_REFER8_FIELD_E	SW_REFER8_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.18.3 Fields

Field	Function
31-2	Base address for reference picture index 8

Table continues on the next page...

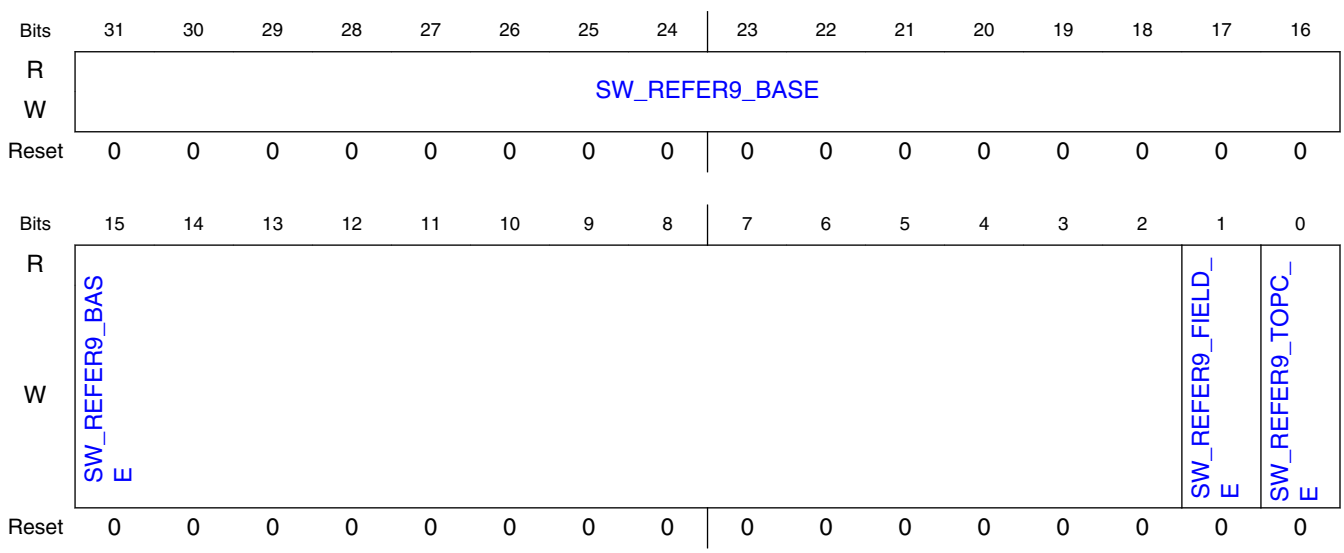
Field	Function
SW_REFER8_BASE	
1 SW_REFER8_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER8_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.19 Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23)

14.1.5.2.19.1 Offset

Register	Offset
SWREG23	5Ch

14.1.5.2.19.2 Diagram



14.1.5.2.19.3 Fields

Field	Function
31-2	Base address for reference picture index 9

Table continues on the next page...

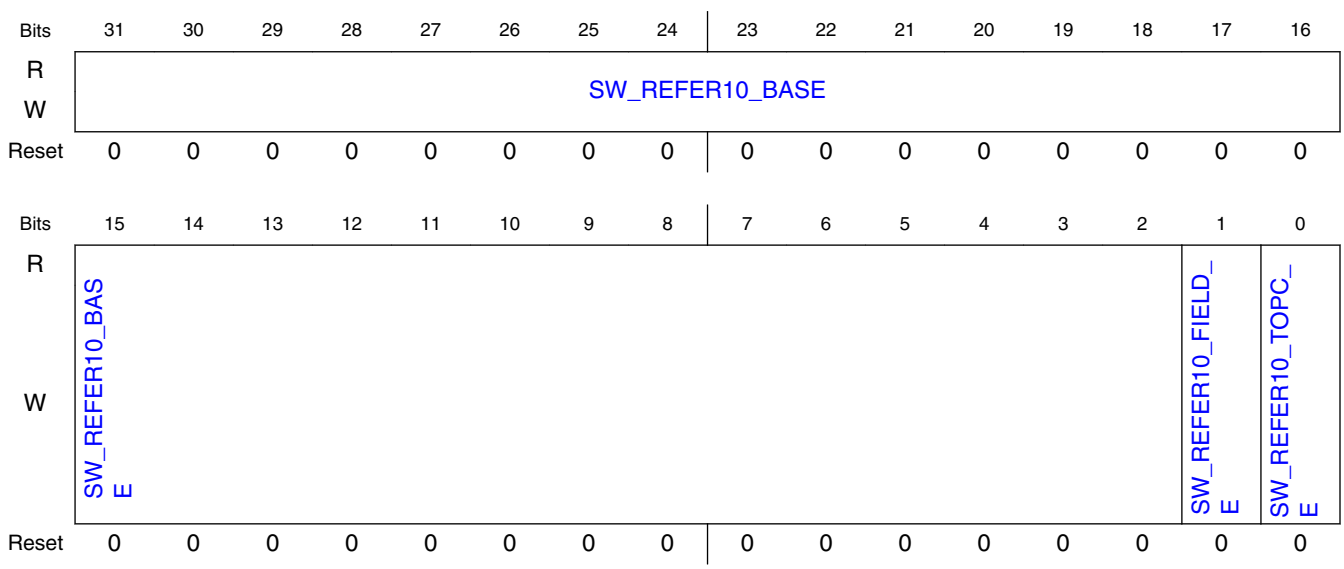
Field	Function
SW_REFER9_BASE	
1 SW_REFER9_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER9_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.20 Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24)

14.1.5.2.20.1 Offset

Register	Offset
SWREG24	60h

14.1.5.2.20.2 Diagram



14.1.5.2.20.3 Fields

Field	Function
31-2 SW_REFER10_BASE	Base address for reference picture index 10
1 SW_REFER10_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER10_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.21 Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25)

14.1.5.2.21.1 Offset

Register	Offset
SWREG25	64h

14.1.5.2.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_BASE														SW_REFER11_FIELD_E	SW_REFER11_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.21.3 Fields

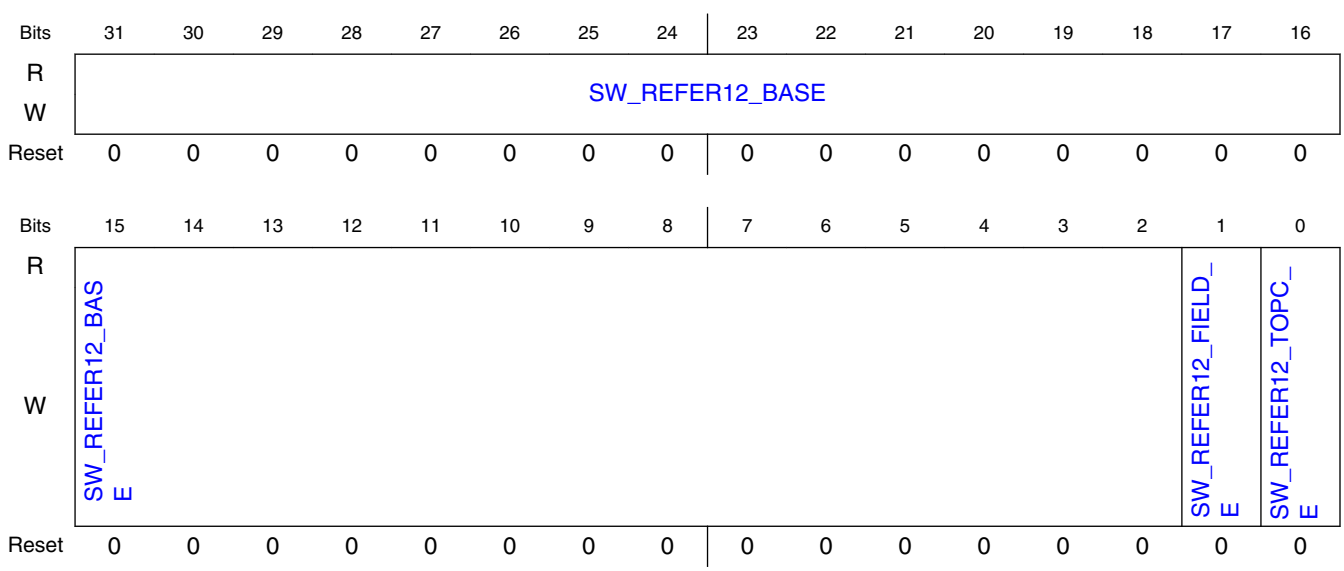
Field	Function
31-2 SW_REFER11_BASE	Base address for reference picture index 11
1 SW_REFER11_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER11_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.22 Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26)

14.1.5.2.22.1 Offset

Register	Offset
SWREG26	68h

14.1.5.2.22.2 Diagram



14.1.5.2.22.3 Fields

Field	Function
31-2 SW_REFER12_BASE	Base address for reference picture index 12
1 SW_REFER12_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER12_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.23 Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27)

14.1.5.2.23.1 Offset

Register	Offset
SWREG27	6Ch

14.1.5.2.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_BASE														SW_REFER13_FIELD_E	SW_REFER13_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.23.3 Fields

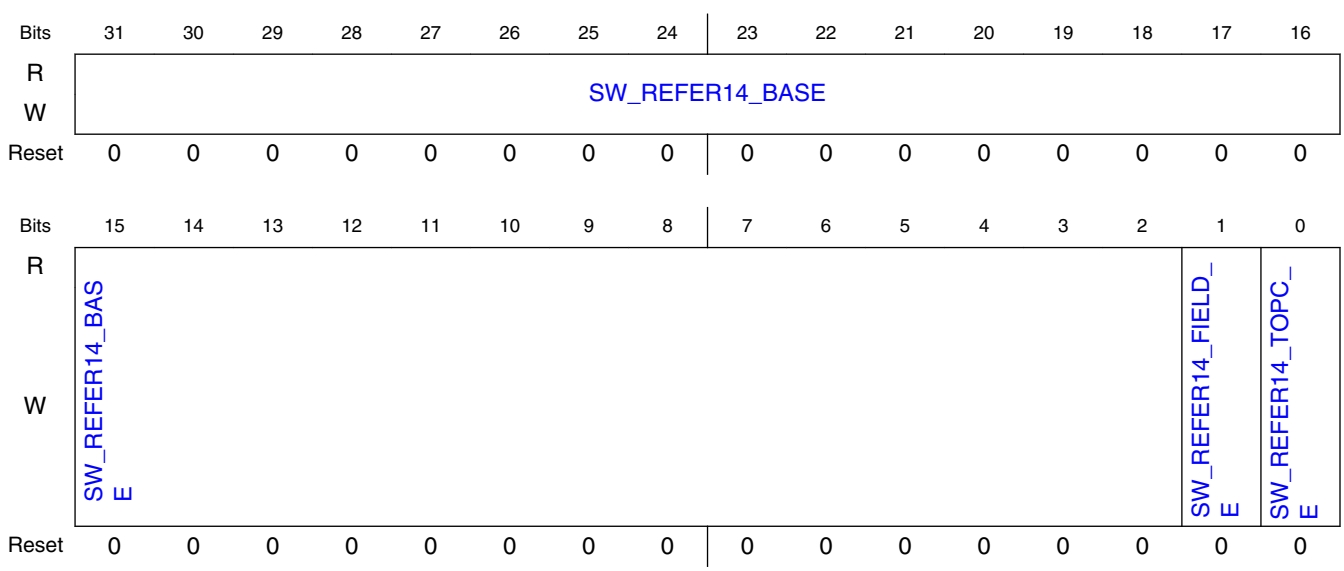
Field	Function
31-2 SW_REFER13_BASE	Base address for reference picture index 13
1 SW_REFER13_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER13_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.24 Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28)

14.1.5.2.24.1 Offset

Register	Offset
SWREG28	70h

14.1.5.2.24.2 Diagram



14.1.5.2.24.3 Fields

Field	Function
31-2 SW_REFER14_BASE	Base address for reference picture index 14
1 SW_REFER14_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER14_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.25 Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29)

14.1.5.2.25.1 Offset

Register	Offset
SWREG29	74h

14.1.5.2.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_BASE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_BASE														SW_REFER15_FIELD_E	SW_REFER15_TOPC_E
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.25.3 Fields

Field	Function
31-2 SW_REFER15_BASE	Base address for reference picture index 15. For Multi View Coding this base address refers to inter view base address
1 SW_REFER15_FIELD_E	Refer picture consist of single fields or frame: 0b - reference picture consists of frame 1b - reference picture consists of fields
0 SW_REFER15_TOPC_E	Which field of reference picture is closer to current picture: 0b - bottom field is closer to current picture 1b - top field is closer to current picture

14.1.5.2.26 Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30)

14.1.5.2.26.1 Offset

Register	Offset
SWREG30	78h

14.1.5.2.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.26.3 Fields

Field	Function
31-16 SW_REFER1_NBR	Number for reference picture index 1

Table continues on the next page...

Field	Function
15-0 SW_REFER0_NBR	Number for reference picture index 0

14.1.5.2.27 Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31)

14.1.5.2.27.1 Offset

Register	Offset
SWREG31	7Ch

14.1.5.2.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.27.3 Fields

Field	Function
31-16 SW_REFER3_NBR	Number for reference picture index 3
15-0 SW_REFER2_NBR	Number for reference picture index 2

14.1.5.2.28 Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32)

14.1.5.2.28.1 Offset

Register	Offset
SWREG32	80h

14.1.5.2.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.28.3 Fields

Field	Function
31-16 SW_REFER5_NBR	Number for reference picture index 5
15-0 SW_REFER4_NBR	Number for reference picture index 4

14.1.5.2.29 Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33)

14.1.5.2.29.1 Offset

Register	Offset
SWREG33	84h

14.1.5.2.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.29.3 Fields

Field	Function
31-16 SW_REFER7_NBR	Number for reference picture index 7
15-0 SW_REFER6_NBR	Number for reference picture index 6

14.1.5.2.30 Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34)

14.1.5.2.30.1 Offset

Register	Offset
SWREG34	88h

14.1.5.2.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.30.3 Fields

Field	Function
31-16 SW_REFER9_NBR	Number for reference picture index 9
15-0 SW_REFER8_NBR	Number for reference picture index 8

14.1.5.2.31 Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35)

14.1.5.2.31.1 Offset

Register	Offset
SWREG35	8Ch

14.1.5.2.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.31.3 Fields

Field	Function
31-16 SW_REFER11_NBR	Number for reference picture index 11
15-0 SW_REFER10_NBR	Number for reference picture index 10

14.1.5.2.32 Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36)

14.1.5.2.32.1 Offset

Register	Offset
SWREG36	90h

14.1.5.2.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.32.3 Fields

Field	Function
31-16 SW_REFER13_NBR	Number for reference picture index 13
15-0 SW_REFER12_NBR	Number for reference picture index 12

14.1.5.2.33 Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37)

14.1.5.2.33.1 Offset

Register	Offset
SWREG37	94h

14.1.5.2.33.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_NBR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.33.3 Fields

Field	Function
31-16 SW_REFER15_NBR	Number for reference picture index 15
15-0 SW_REFER14_NBR	Number for reference picture index 14

14.1.5.2.34 Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38)

14.1.5.2.34.1 Offset

Register	Offset
SWREG38	98h

14.1.5.2.34.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER_LTERM_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER_LTERM_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.34.3 Fields

Field	Function
31-0 SW_REFER_LTERM_E	Long term flag for reference picture index [31:0]. Definition: If frame is being decoded the bits 31:15 are used, Bit 31 for picture index 0, Bit 30 for picture index 1 etc… IF field is being decoded the bits 31:0 are used, Bit 31 for reference picture 0 top field, bit 30 for reference picture 0 bottom field etc...

14.1.5.2.35 Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39)

14.1.5.2.35.1 Offset

Register	Offset
SWREG39	9Ch

14.1.5.2.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER_VALID_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER_VALID_E															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.2.35.3 Fields

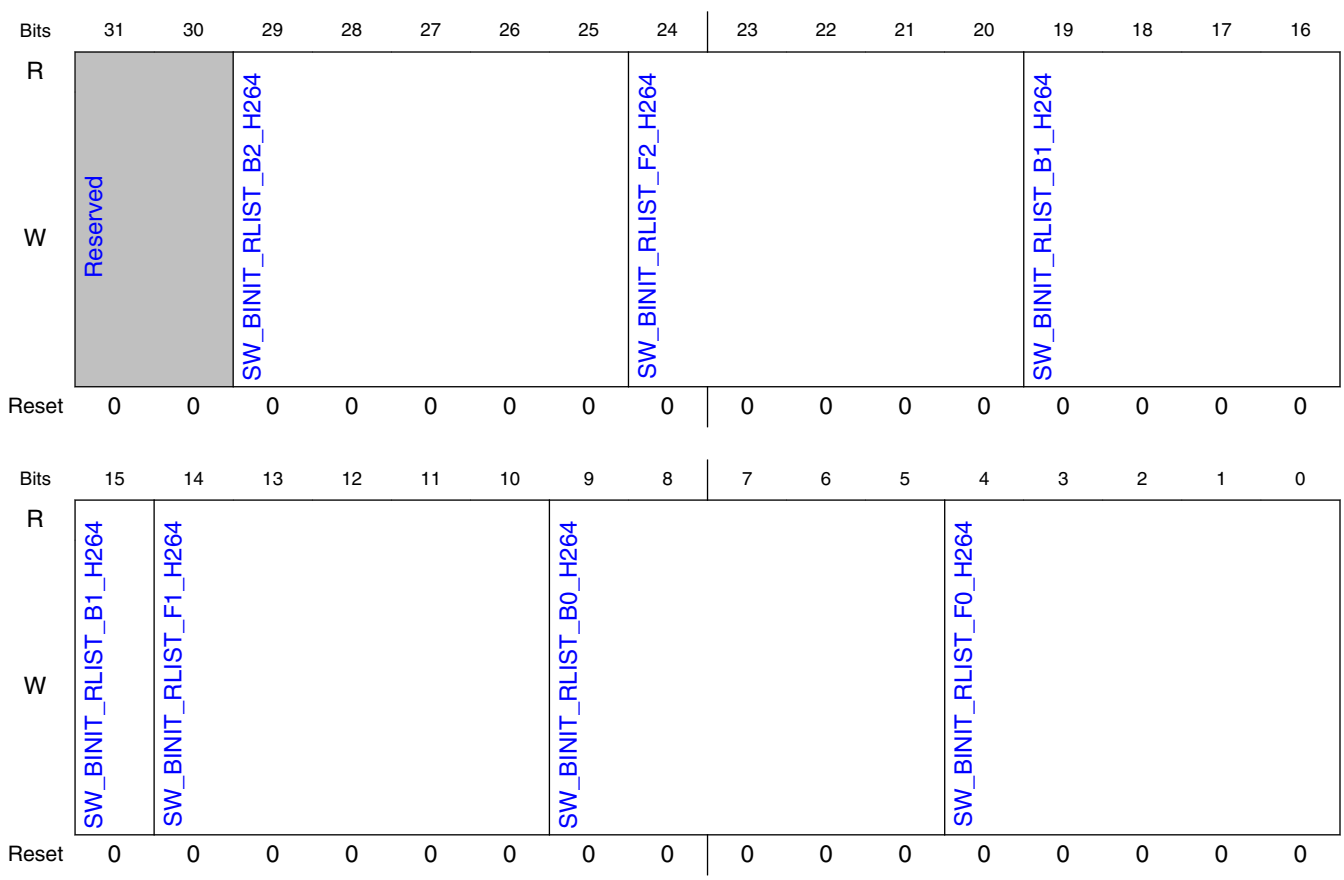
Field	Function
31-0 SW_REFER_VALID_E	Valid flag for reference picture index [31:0].Definition: If frame is being decoded the bits 31:15 are used, Bit 31 for picture index 0, Bit 30 for picture index 1 etc… IF field is being decoded the bits 31:0 are used, Bit 31 for reference picture 0 top field, bit 30 for reference picture 0 bottom field etc...

14.1.5.2.36 bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_H264)

14.1.5.2.36.1 Offset

Register	Offset
SWREG42_H264	A8h

14.1.5.2.36.2 Diagram



14.1.5.2.36.3 Fields

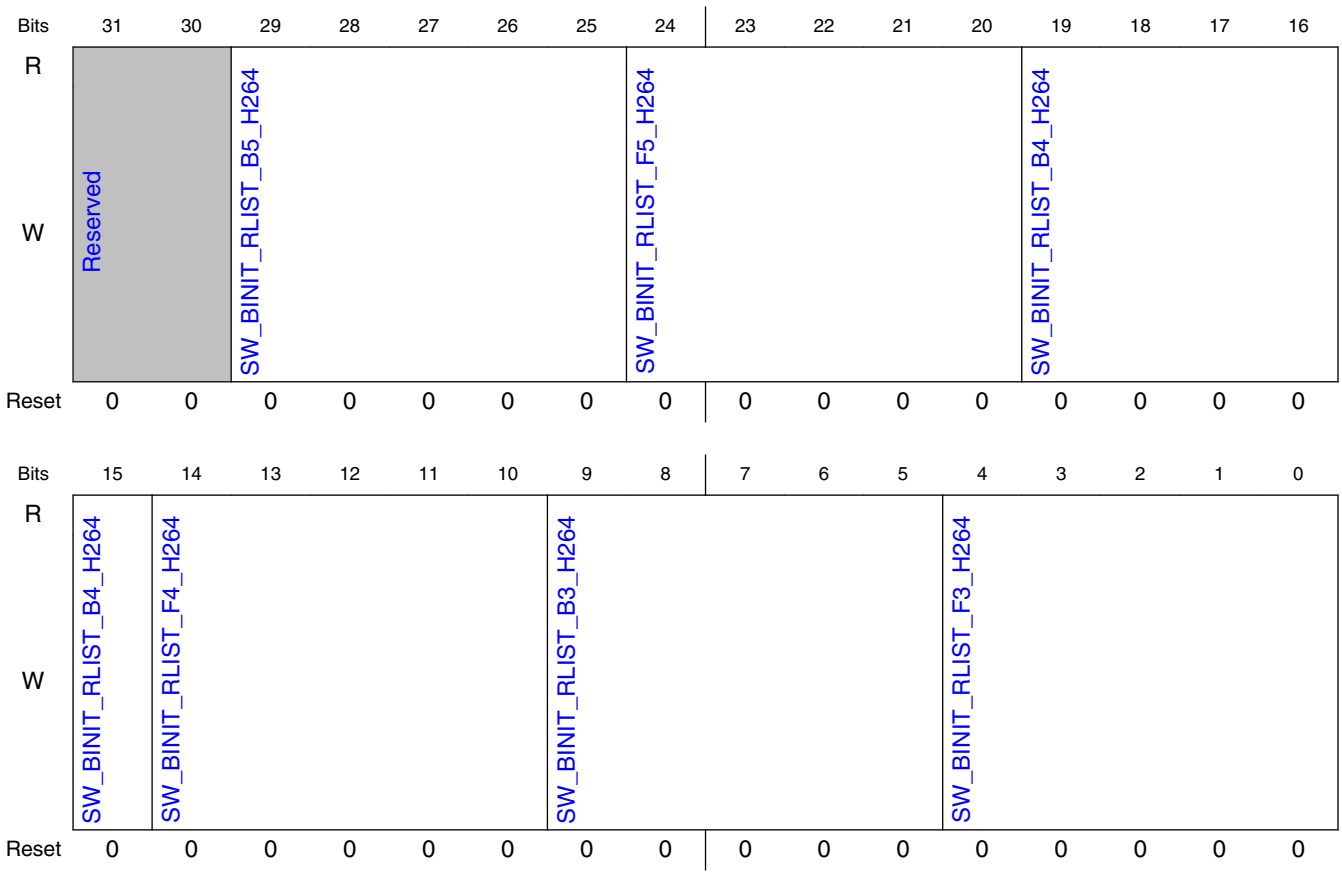
Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B2_H264	Initial reference picture list for bi-direct backward picid 2
24-20 SW_BINIT_RLIST_F2_H264	Initial reference picture list for bi-direct forward picid 2
19-15 SW_BINIT_RLIST_B1_H264	Initial reference picture list for bi-direct backward picid 1
14-10 SW_BINIT_RLIST_F1_H264	Initial reference picture list for bi-direct forward picid 1
9-5 SW_BINIT_RLIST_B0_H264	Initial reference picture list for bi-direct backward picid 0
4-0 SW_BINIT_RLIST_F0_H264	Initial reference picture list for bi-direct forward picid 0

14.1.5.2.37 bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_H264)

14.1.5.2.37.1 Offset

Register	Offset
SWREG43_H264	ACh

14.1.5.2.37.2 Diagram



14.1.5.2.37.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B5_H264	Initial reference picture list for bi-direct backward picid 5
24-20 SW_BINIT_RLIST_F5_H264	Initial reference picture list for bi-direct forward picid 5
19-15 SW_BINIT_RLIST_B4_H264	Initial reference picture list for bi-direct backward picid 4
14-10 SW_BINIT_RLIST_F4_H264	Initial reference picture list for bi-direct forward picid 4
9-5	Initial reference picture list for bi-direct backward picid 3

Table continues on the next page...

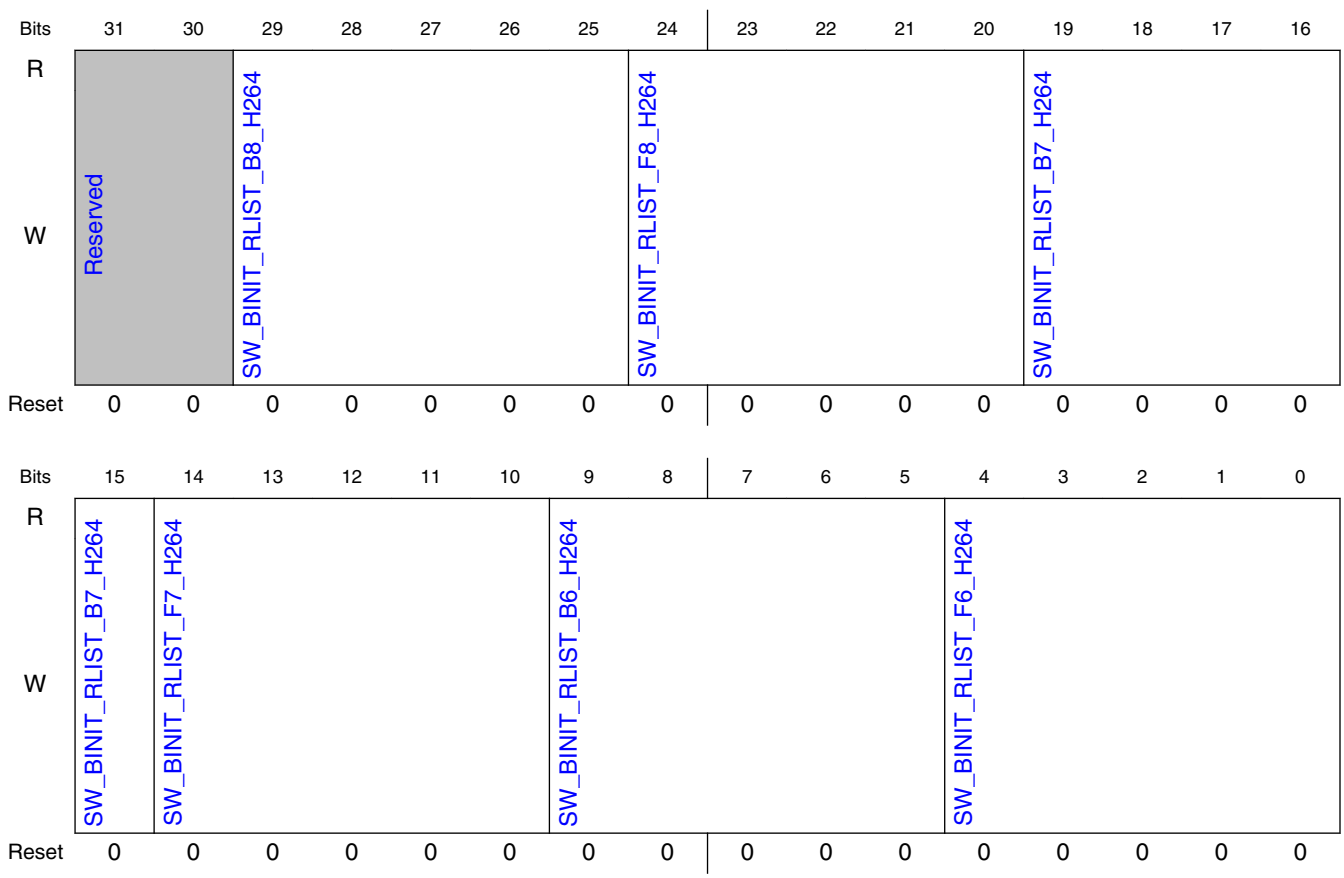
Field	Function
SW_BINIT_RLIST_B3_H264	
4-0 SW_BINIT_RLIST_F3_H264	Initial reference picture list for bi-direct forward picid 3

14.1.5.2.38 bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_H264)

14.1.5.2.38.1 Offset

Register	Offset
SWREG44_H264	B0h

14.1.5.2.38.2 Diagram



14.1.5.2.38.3 Fields

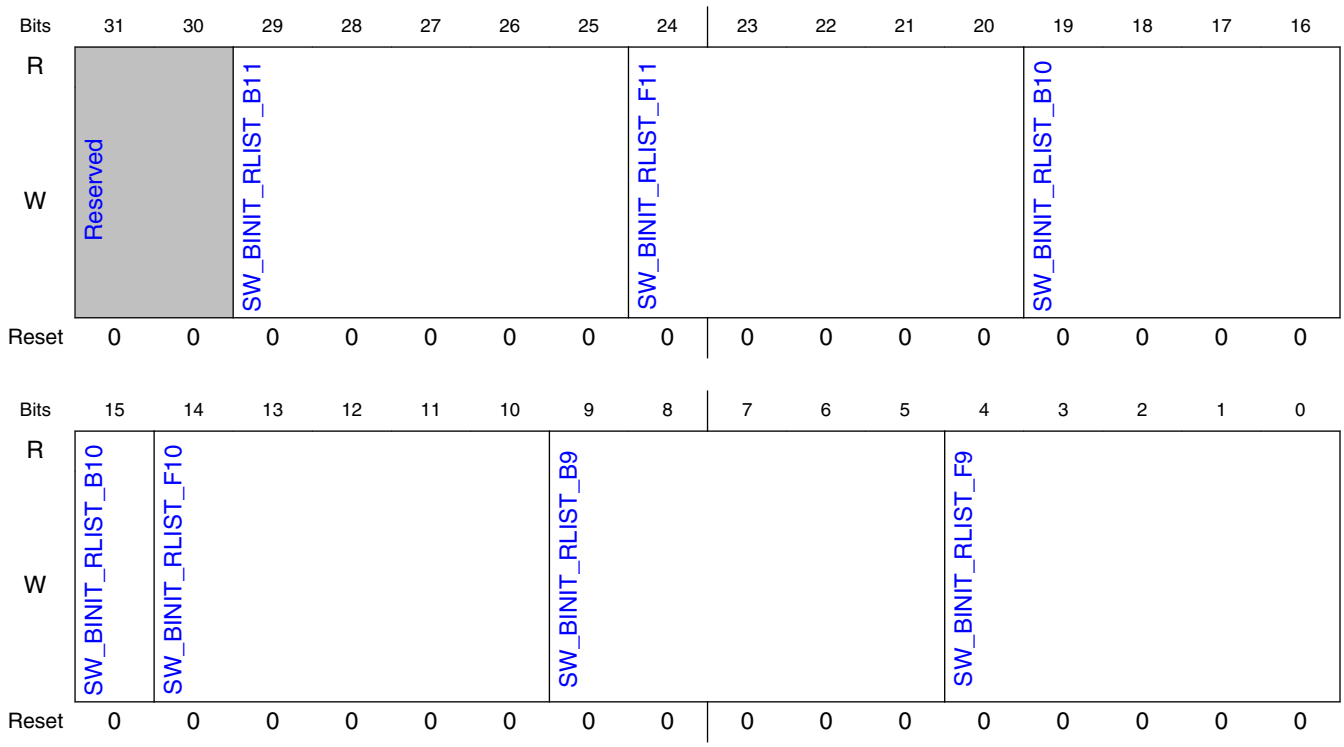
Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B8_H264	Initial reference picture list for bi-direct backward picid 8
24-20 SW_BINIT_RLIST_F8_H264	Initial reference picture list for bi-direct forward picid 8
19-15 SW_BINIT_RLIST_B7_H264	Initial reference picture list for bi-direct backward picid 7
14-10 SW_BINIT_RLIST_F7_H264	Initial reference picture list for bi-direct forward picid 7
9-5 SW_BINIT_RLIST_B6_H264	Initial reference picture list for bi-direct backward picid 6
4-0 SW_BINIT_RLIST_F6_H264	Initial reference picture list for bi-direct forward picid 6

14.1.5.2.39 bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45)

14.1.5.2.39.1 Offset

Register	Offset
SWREG45	B4h

14.1.5.2.39.2 Diagram



14.1.5.2.39.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_BINIT_RLIST_B11	Initial reference picture list for bi-direct backward picid 11
24-20 SW_BINIT_RLIST_F11	Initial reference picture list for bi-direct forward picid 11
19-15 SW_BINIT_RLIST_B10	Initial reference picture list for bi-direct backward picid 10
14-10 SW_BINIT_RLIST_F10	Initial reference picture list for bi-direct forward picid 10
9-5 SW_BINIT_RLIST_B9	Initial reference picture list for bi-direct backward picid 9
4-0	Initial reference picture list for bi-direct forward picid 9

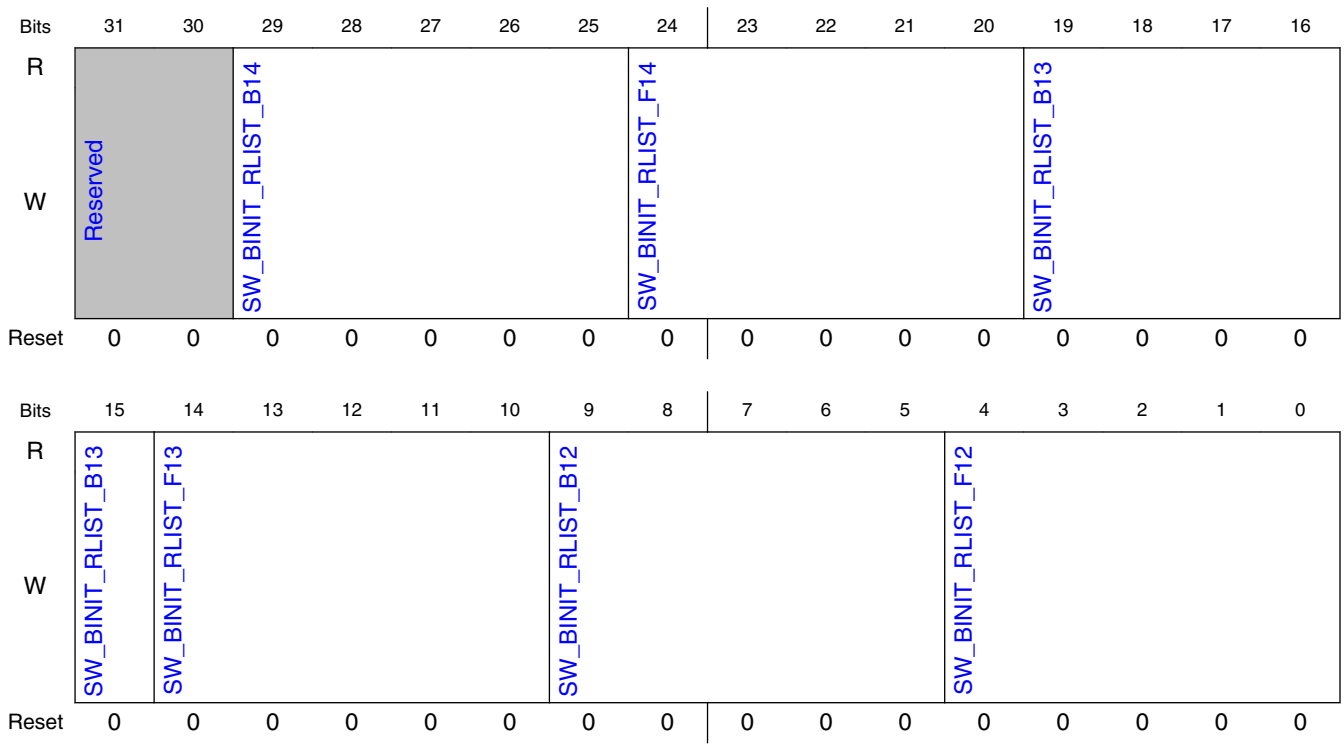
Field	Function
SW_BINIT_RLIST_F9	

14.1.5.2.40 bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46)

14.1.5.2.40.1 Offset

Register	Offset
SWREG46	B8h

14.1.5.2.40.2 Diagram



14.1.5.2.40.3 Fields

Field	Function
31-30 —	Reserved.

Table continues on the next page...

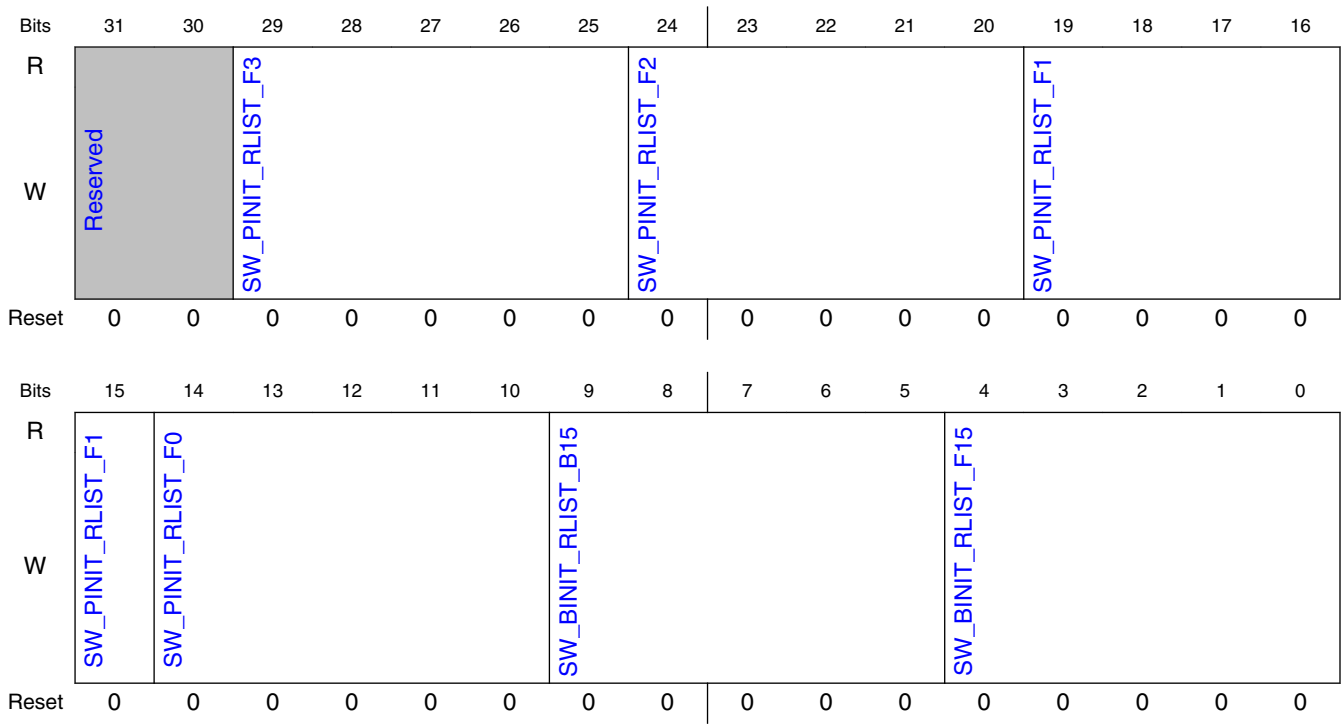
Field	Function
29-25 SW_BINIT_RLIST_B14	Initial reference picture list for bi-direct backward picid 14
24-20 SW_BINIT_RLIST_F14	Initial reference picture list for bi-direct forward picid 14
19-15 SW_BINIT_RLIST_B13	Initial reference picture list for bi-direct backward picid 13
14-10 SW_BINIT_RLIST_F13	Initial reference picture list for bi-direct forward picid 13
9-5 SW_BINIT_RLIST_B12	Initial reference picture list for bi-direct backward picid 12
4-0 SW_BINIT_RLIST_F12	Initial reference picture list for bi-direct forward picid 12

14.1.5.2.41 bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47)

14.1.5.2.41.1 Offset

Register	Offset
SWREG47	BCh

14.1.5.2.41.2 Diagram



14.1.5.2.41.3 Fields

Field	Function
31-30 —	Reserved.
29-25 SW_PINIT_RLIST_F3	Initial reference picture list for P forward picid 3
24-20 SW_PINIT_RLIST_F2	Initial reference picture list for P forward picid 2
19-15 SW_PINIT_RLIST_F1	Initial reference picture list for P forward picid 1
14-10 SW_PINIT_RLIST_F0	Initial reference picture list for P forward picid 0
9-5 SW_BINIT_RLIST_B15	Initial reference picture list for bi-direct backward picid 15
4-0	Initial reference picture list for bi-direct forward picid 15

Field	Function
SW_BINIT_RLIST_F15	

14.1.5.3 VPU_G1 VP8 decoder register descriptions

14.1.5.3.1 VPU_G1 VP8 decoder memory map

VPU_G1_VP8 base address: 3830_0000h

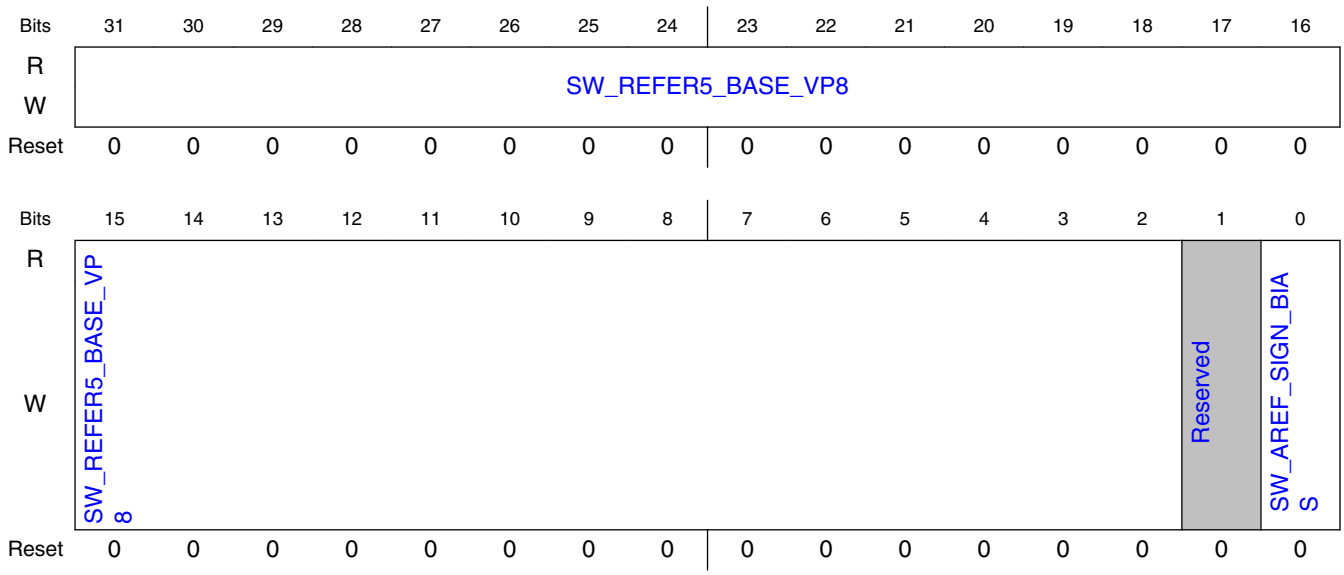
Offset	Register	Width (In bits)	Access	Reset value
4Ch	Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19_VP8)	32	RW	0000_0000h
50h	Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20_VP8)	32	RW	0000_0000h
54h	Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21_VP8)	32	RW	0000_0000h

14.1.5.3.2 Base address for reference picture index 5 / MPEG4 TRB/TRD delta 0 / VC-1 intensity control 3 List of VLC code lengths in first/second JPEG AC table / VP6/VP7 scan maps (SWREG19_VP8)

14.1.5.3.2.1 Offset

Register	Offset
SWREG19_VP8	4Ch

14.1.5.3.2.2 Diagram



14.1.5.3.2.3 Fields

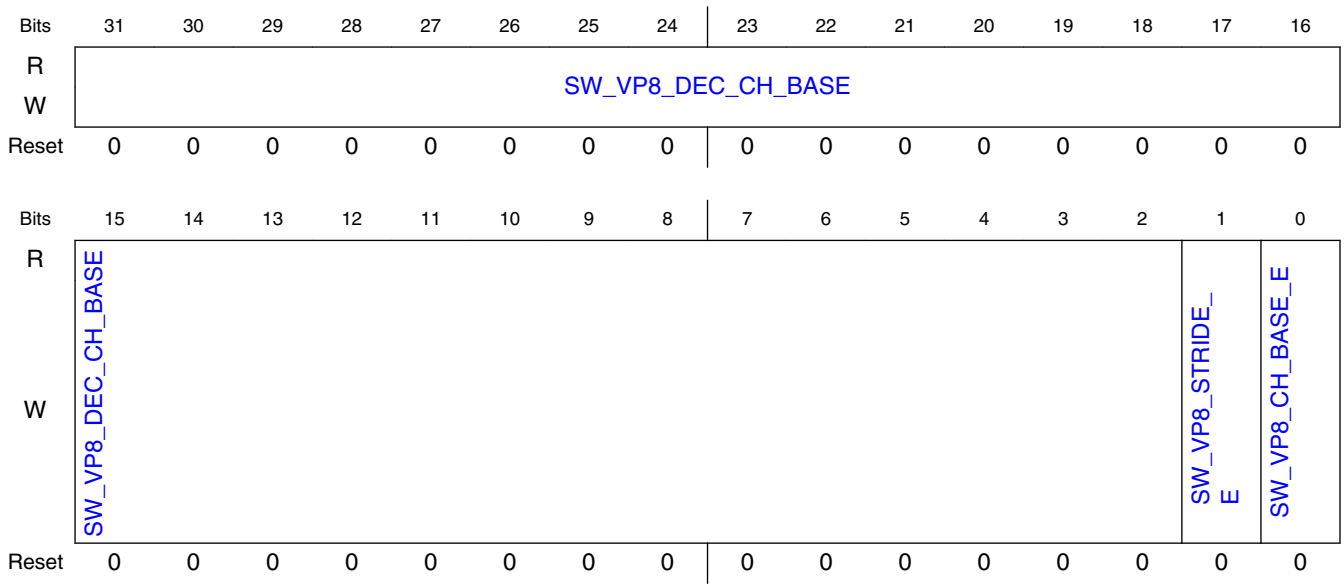
Field	Function
31-2 SW_REFER5_BASE_VP8	H.264: Base address for reference picture index 5 VP8: Base address for alternate reference picture (corresponds picid 5)
1 —	Reserved.
0 SW_AREF_SIGN_BIAS	VP8 only: Reference picture sign bias for Alternate reference frame

14.1.5.3.3 Base address for reference picture index 6 / / MPEG4 TRB/TRD delta -1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG20_VP8)

14.1.5.3.3.1 Offset

Register	Offset
SWREG20_VP8	50h

14.1.5.3.3.2 Diagram



14.1.5.3.3.3 Fields

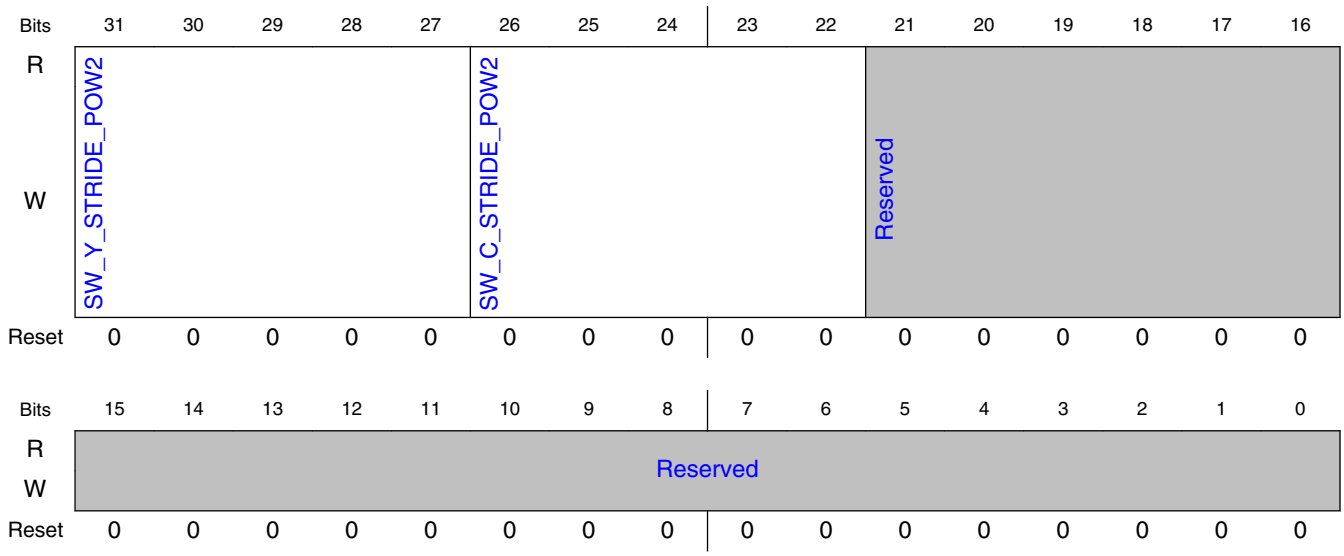
Field	Function
31-2 SW_VP8_DEC_CH_BASE	VP8 video base address for decoder output chrominance data (if vp8 stride configuration is enabled)
1 SW_VP8_STRIDE_E	VP8 stride enable. Can be set high only if HW configuration supports strides. Y and C strides are used instead of picture width. Separate chrominance base addresses are used instead of internal chrominance offsets. 0b - Not enabled 1b - Enabled
0 SW_VP8_CH_BASE_E	VP8 separate chrominance enable: 0b - Write/Read chrominance data from internal offset after the luminance data 1b - Write/Read chrominance data from separate base addresses given by SW

14.1.5.3.4 Base address for reference picture index 7 / MPEG4 TRB/TRD delta 1 / List of VLC code lengths in second JPEG AC table / VP6/VP7 scan maps (SWREG21_VP8)

14.1.5.3.4.1 Offset

Register	Offset
SWREG21_VP8	54h

14.1.5.3.4.2 Diagram



14.1.5.3.4.3 Fields

Field	Function
31-27 SW_Y_STRIDE_POW2	VP8 Y stride length informed by 2^n (n=sw_y_stride_pow2). Valid range 10-17 for 32 bit bus and 10-18 for 64 bit bus
26-22 SW_C_STRIDE_POW2	VP8 C stride length informed by 2^n (n=sw_c_stride_pow2). Valid range 10-17 for 32 bit bus and 10-18 for 64 bit bus
21-0 —	Reserved.

14.1.5.4 VPU_G1 VP7/VP8 decoder register descriptions

14.1.5.4.1 VPU_G1 VP7/VP8 decoder memory map

VPU_G1_VP7_VP8 base address: 3830_0000h

Offset	Register	Width (In bits)	Access	Reset value
10h	Decoder control register 1 (picture parameters) (SWREG4_JPEG_VP7_VP8)	32	RW	0000_0000h
14h	Decoder control register 2 (stream decoding table selects) (SWREG5_VP7_VP8)	32	RW	0000_0000h
18h	Decoder control register 3 (stream buffer information) (SWREG6_VP7_VP8)	32	RW	0000_0000h
1Ch	Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7_VP7_VP8)	32	RW	0000_0000h
28h	Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10_VP7_VP8)	32	RW	0000_0000h
2Ch	Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11_VP7_VP8)	32	RW	0000_0000h
38h	Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14_VP7_VP8)	32	RW	0000_0000h
3Ch	Base address for reference picture index 1 / JPEG control (SWREG15_VP7_VP8)	32	RW	0000_0000h
48h	Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18_VP7_VP8)	32	RW	0000_0000h
58h	Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22_VP7_VP8)	32	RW	0000_0000h
5Ch	Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23_VP7_VP8)	32	RW	0000_0000h
60h	Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24_VP7_VP8)	32	RW	0000_0000h
64h	Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25_VP7_VP8)	32	RW	0000_0000h
68h	Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26_VP7_VP8)	32	RW	0000_0000h
6Ch	Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27_VC1)	32	RW	0000_0000h
70h	Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28_VP7_VP8)	32	RW	0000_0000h
74h	Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29_VP7_VP8)	32	RW	0000_0000h
78h	Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30_VP7_VP8)	32	RW	0000_0000h

Table continues on the next page...

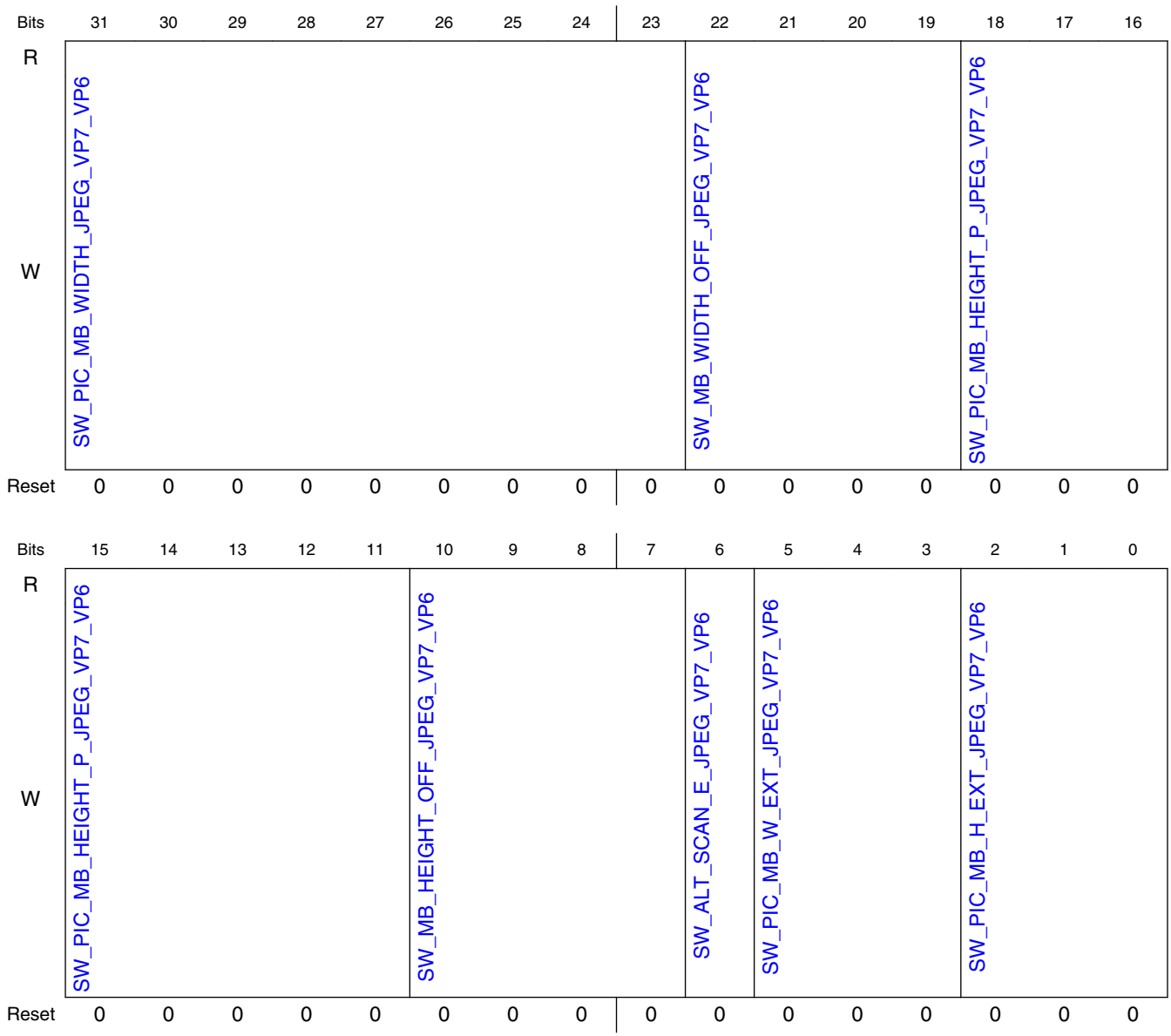
Offset	Register	Width (In bits)	Access	Reset value
7Ch	Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31_VP7_VP8)	32	RW	0000_0000h
80h	Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32_VP7_VP8)	32	RW	0000_0000h
84h	Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33_VP7_VP8)	32	RW	0000_0000h
88h	Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34_H263)	32	RW	0000_0000h
8Ch	Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35_VC1)	32	RW	0000_0000h
90h	Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36_VC1)	32	RW	0000_0000h
94h	Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37_VP6_VP7_VP8)	32	RW	0000_0000h
98h	Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38_VP6_VP7_VP8)	32	RW	0000_0000h
9Ch	Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39_VP6_VP7_VP8)	32	RW	0000_0000h
A8h	bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_VP6)	32	RW	0000_0000h
ACh	bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_VP7_VP8)	32	RW	0000_0000h
B0h	bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_VP7_VP8)	32	RW	0000_0000h
B4h	bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45_VP7_VP8)	32	RW	0000_0000h
B8h	bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46_VP7_VP8)	32	RW	0000_0000h
BCh	bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47_VP7_VP8)	32	RW	0000_0000h

14.1.5.4.2 Decoder control register 1 (picture parameters) (SWREG4_JPEG_VP7_VP8)

14.1.5.4.2.1 Offset

Register	Offset
SWREG4_JPEG_VP7_VP8	10h

14.1.5.4.2.2 Diagram



14.1.5.4.2.3 Fields

Field	Function
31-23 SW_PIC_MB_WIDTH_JPEG_VP7_VP6	Picture width in macroblocks = ((width in pixels + 15) /16)
22-19	The amount of meaningfull horizontal pixels in last MB (width offset) 0 if exactly 16 pixels multiple picture and all the horizontal pixels in last MB are meaningfull

Table continues on the next page...

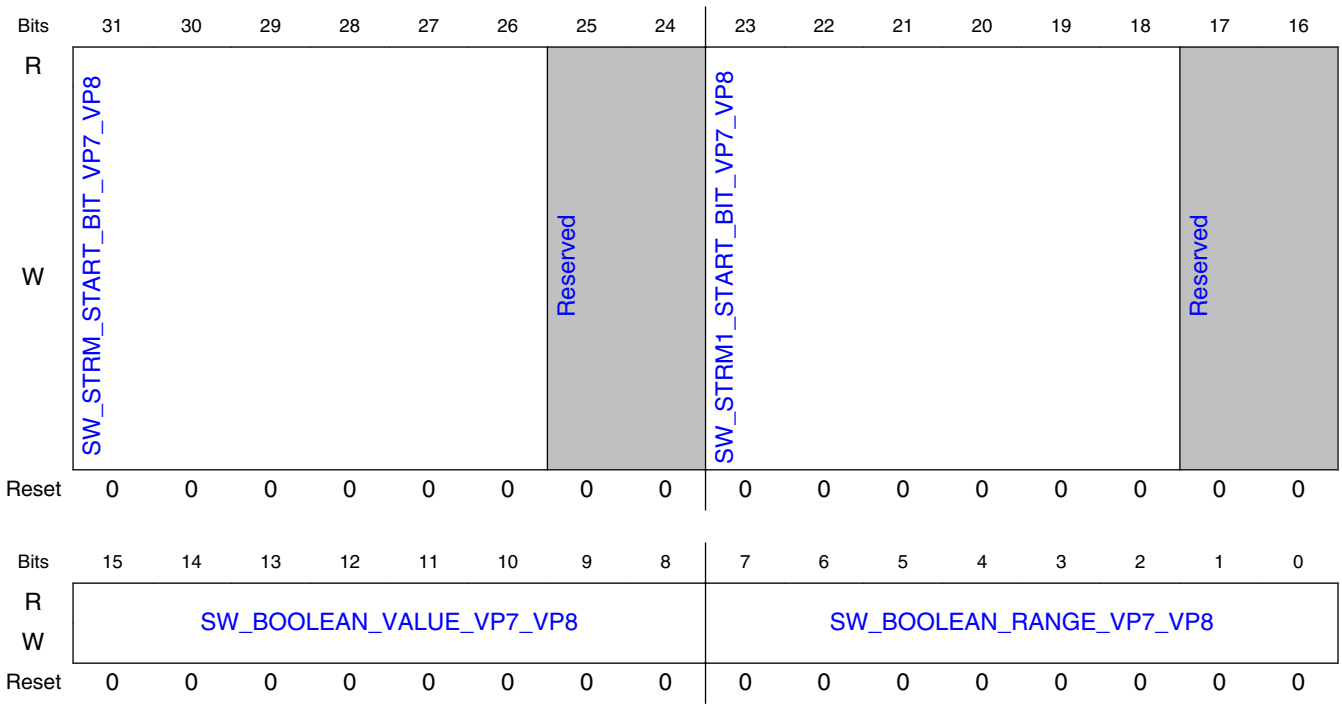
Field	Function
SW_MB_WIDTH_OFF_JPEG_VP7_VP6	
18-11 SW_PIC_MB_HEIGHT_P_JPEG_VP7_VP6	Picture height in macroblocks =((height in pixels+15)/16). Picture height is informed as size of the (progressive) frame also for single field (of interlaced content) is being decoded
10-7 SW_MB_HEIGHT_OFF_JPEG_VP7_VP6	The amount of meaningful vertical pixels in last MB (height offset 0 if exactly 16 pixels multiple picture and all the vertical pixels in last MB are meaningfull
6 SW_ALT_SCAN_E_JPEG_VP7_VP6	indicates alternative vertical scan method used for interlaced frames
5-3 SW_PIC_MB_W_EXT_JPEG_VP7_VP6	Picture mb width extension. If sw_pic_mb_width does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb)
2-0 SW_PIC_MB_H_EXT_JPEG_VP7_VP6	Picture mb height extension. If sw_pic_mb_height_p does not fit to 9 bits then these bits are used to increase the range upto 11 bits (used as 3 msb). For 4k video one bit is used for extension (bit 0)

14.1.5.4.3 Decoder control register 2 (stream decoding table selects) (SWREG5_VP7_VP8)

14.1.5.4.3.1 Offset

Register	Offset
SWREG5_VP7_VP8	14h

14.1.5.4.3.2 Diagram



14.1.5.4.3.3 Fields

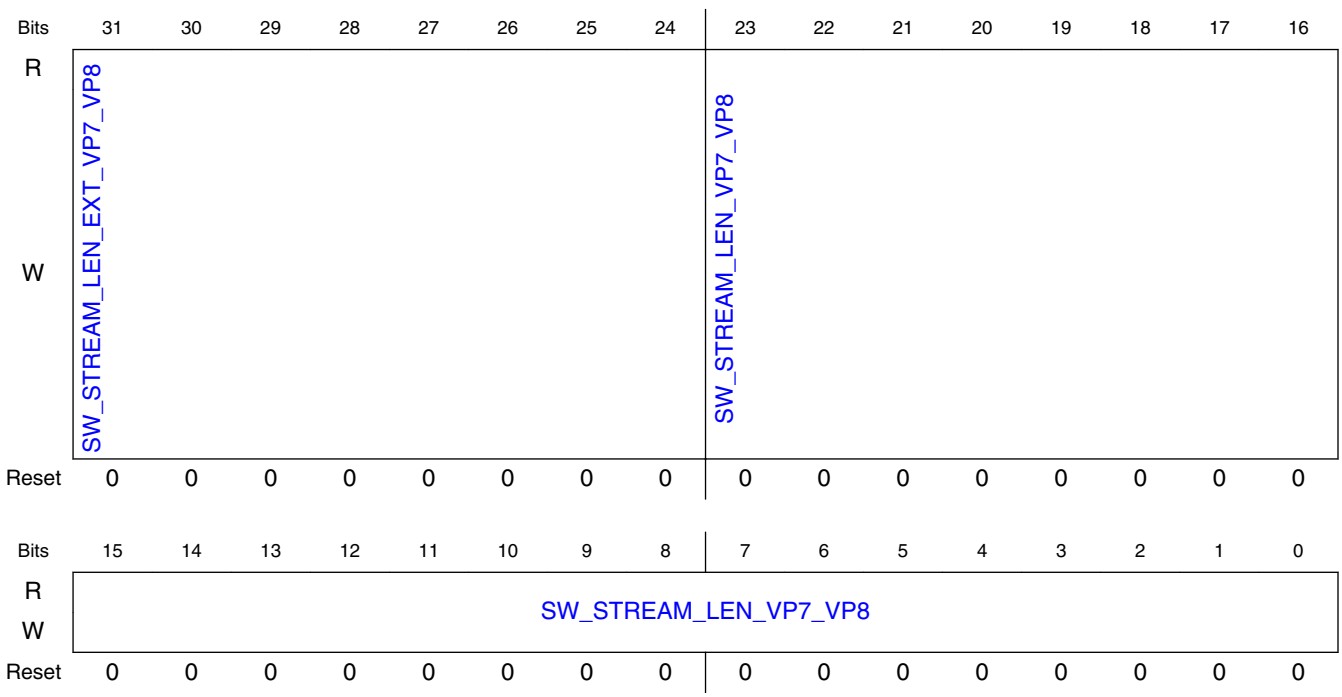
Field	Function
31-26 SW_STRM_START_BIT_VP7_VP8	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
25-24 —	Reserved.
23-18 SW_STRM1_START_BIT_VP7_VP8	Start bit for ctrl-stream (needed if multistream is enabled, assosiates with sw_bitpl_ctrl_base)
17-16 —	Reserved.
15-8 SW_BOOLEAN_VALUE_VP7_VP8	Initial value for boolean dec
7-0 SW_BOOLEAN_RANGE_VP7_VP8	Initial range for boolean dec

14.1.5.4.4 Decoder control register 3 (stream buffer information) (SWREG6_VP7_VP8)

14.1.5.4.4.1 Offset

Register	Offset
SWREG6_VP7_VP8	18h

14.1.5.4.4.2 Diagram



14.1.5.4.4.3 Fields

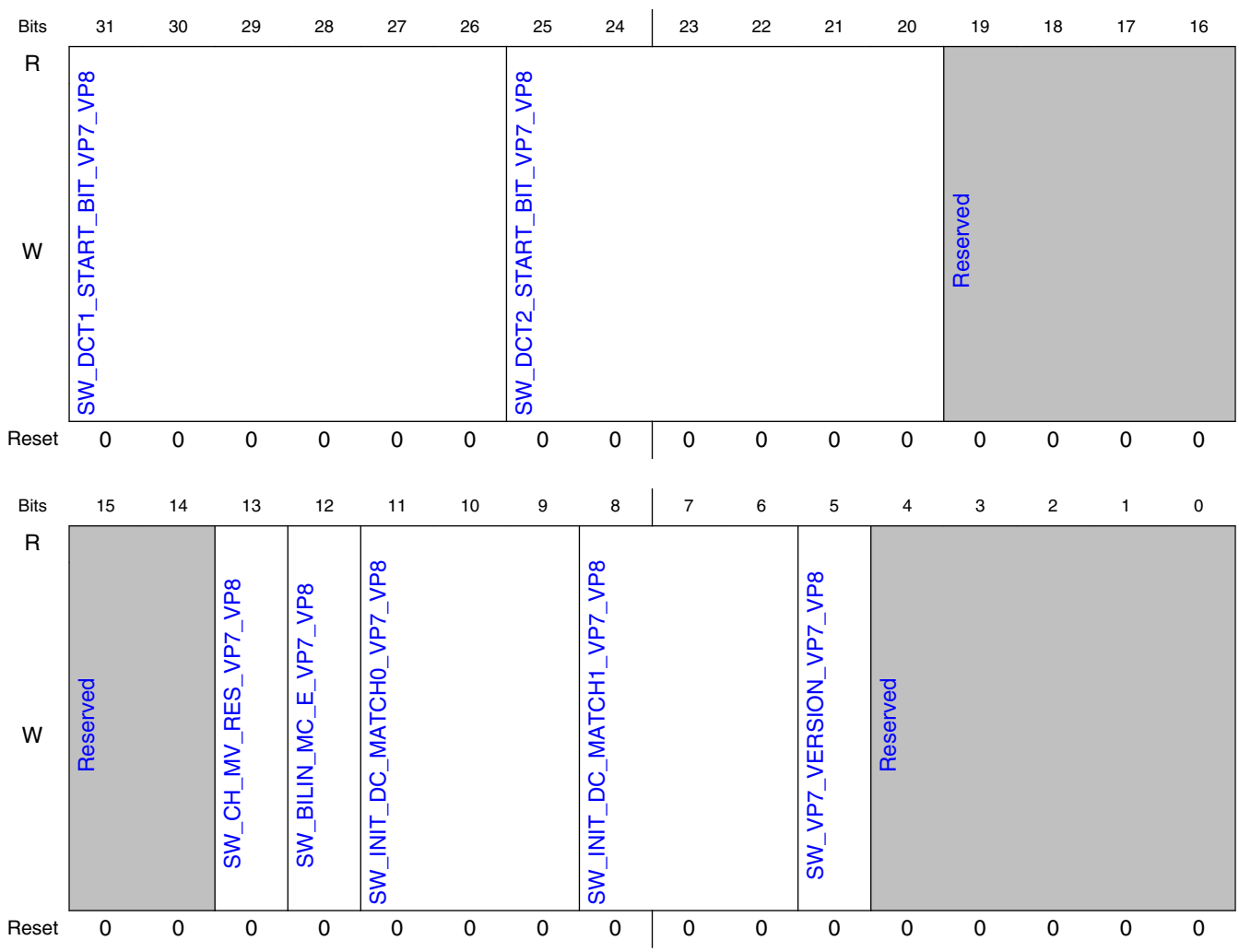
Field	Function
31-24 SW_STREAM_LEN_EXT_VP7_VP8	Extended stream length for WEBP/VP8
23-0 SW_STREAM_LEN_VP7_VP8	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For VC-1/VP6 the buffer must include data for one picture/slice of the picture For H264/MPEG4/H263/MPEG2/MPEG1 the buffer must include at least data for one slice/VP of the picture For JPEG the buffer size must be a multiple of 256 bytes or the amount of data for one picture.

14.1.5.4.5 Decoder control register 4 (H264, VC-1, VP6 and progressive JPEG control) (SWREG7_VP7_VP8)

14.1.5.4.5.1 Offset

Register	Offset
SWREG7_VP7_VP8	1Ch

14.1.5.4.5.2 Diagram



14.1.5.4.5.3 Fields

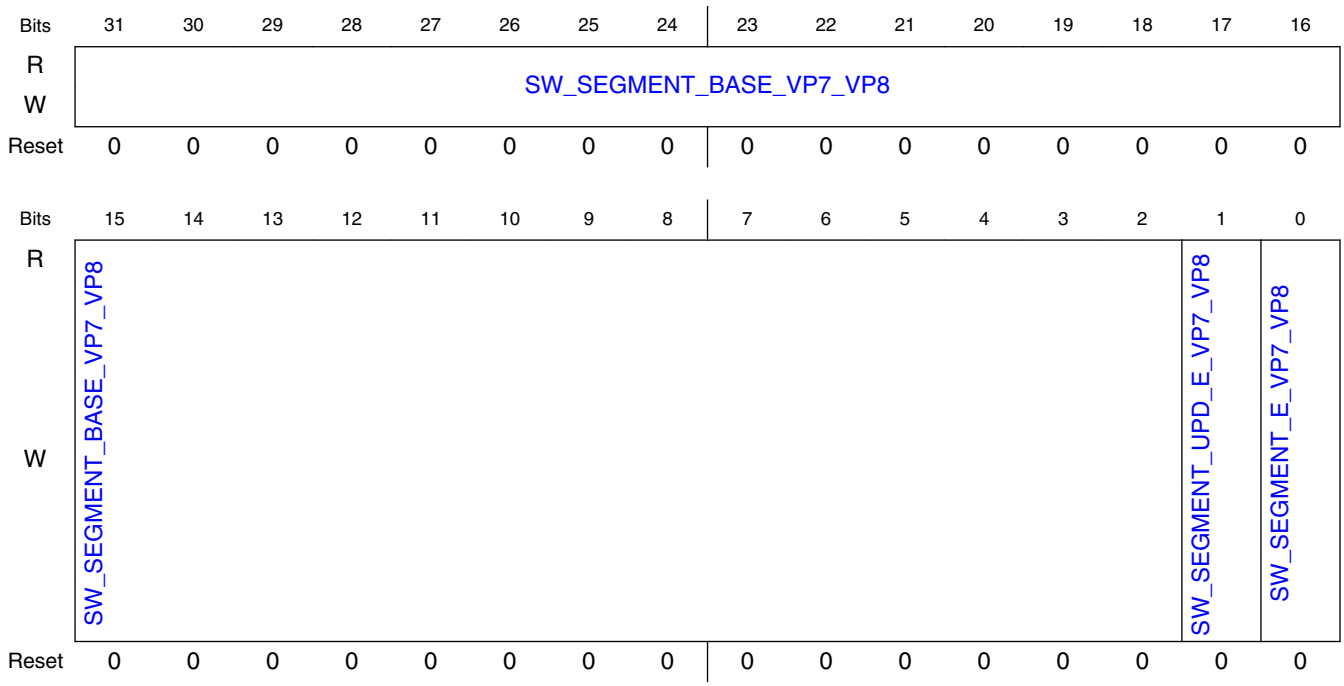
Field	Function
31-26 SW_DCT1_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 1
25-20 SW_DCT2_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 2
19-14 —	Reserved.
13 SW_CH_MV_RES_VP7_VP8	VP7/VP8 Chrominance motion vector resolution: 0b - Full pixel 1b - 1/8 pixel
12 SW_BILIN_MC_E_VP7_VP8	Bilinear motion compensation enable: 0b - Bicubic interpolation used 1b - Bilinear interpolation used
11-9 SW_INIT_DC_MATCH0_VP7_VP8	Initial DC prediction mach count 0. After HW has decoded a picture HW returns the final match count0 information which is read by SW
8-6 SW_INIT_DC_MATCH1_VP7_VP8	Initial DC prediction mach count 1. After HW has decoded a picture HW returns the final match count1 information which is read by SW
5 SW_VP7_VERSION_VP7_VP8	VP7 version information to streamd: 0b - VP7 version 7.0 1b - VP7 version 7.1 or better
4-0 —	Reserved.

14.1.5.4.6 Base address for differential motion vector base address (RLC-mode) /H264 P initial fwd ref pic list register (4-9)/ VC-1 intensity control 1/ VP7 and VP8 segmentation base register (SWREG10_VP7_VP8)

14.1.5.4.6.1 Offset

Register	Offset
SWREG10_VP7_VP8	28h

14.1.5.4.6.2 Diagram



14.1.5.4.6.3 Fields

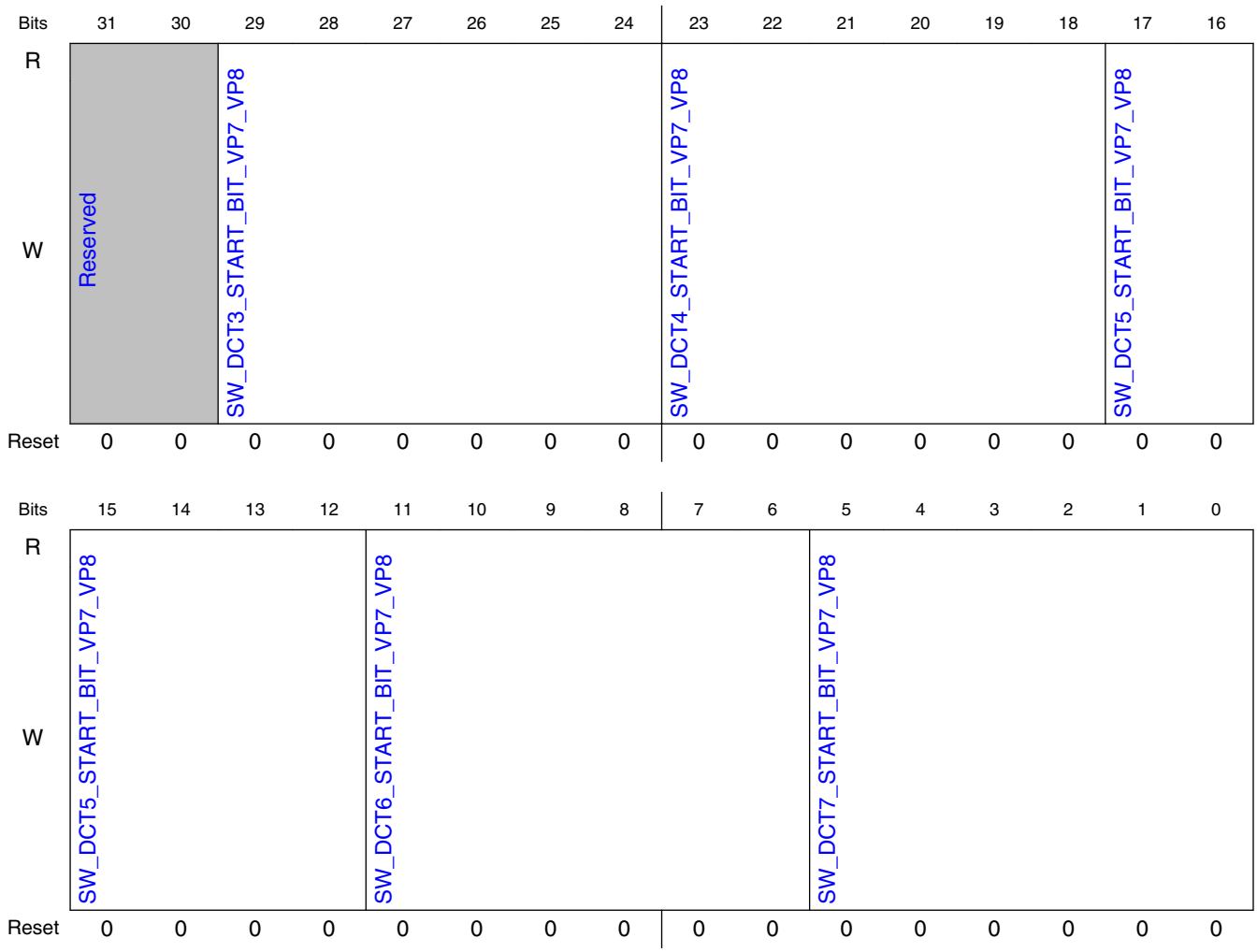
Field	Function
31-2 SW_SEGMENT_BASE_VP7_VP8	VP7/VP8: base address for segmentation map values
1 SW_SEGMENT_UPD_E_VP7_VP8	VP7/VP8 Segmentation map update enable: '0': segmentation values are read from external memory (from segment_base) '1': segmentation update is included in stream
0 SW_SEGMENT_E_VP7_VP8	Segmentation enable: '0': segmentation is not enabled '1': segmentation is enabled (sw_segment_upd_e value is used)

14.1.5.4.7 Decoder control register 7 (VLC) / base address for H.264 intra prediction 4x4 / base address for MPEG-4 DC component (RLC) / H264 P initial fwd ref pic list register (10-15) / VC-1 intensity control 2 (SWREG11_VP7_VP8)

14.1.5.4.7.1 Offset

Register	Offset
SWREG11_VP7_VP8	2Ch

14.1.5.4.7.2 Diagram



14.1.5.4.7.3 Fields

Field	Function
31-30 —	Reserved.
29-24	Start bit for VP7/VP8 DCT stream partition index 3

Table continues on the next page...

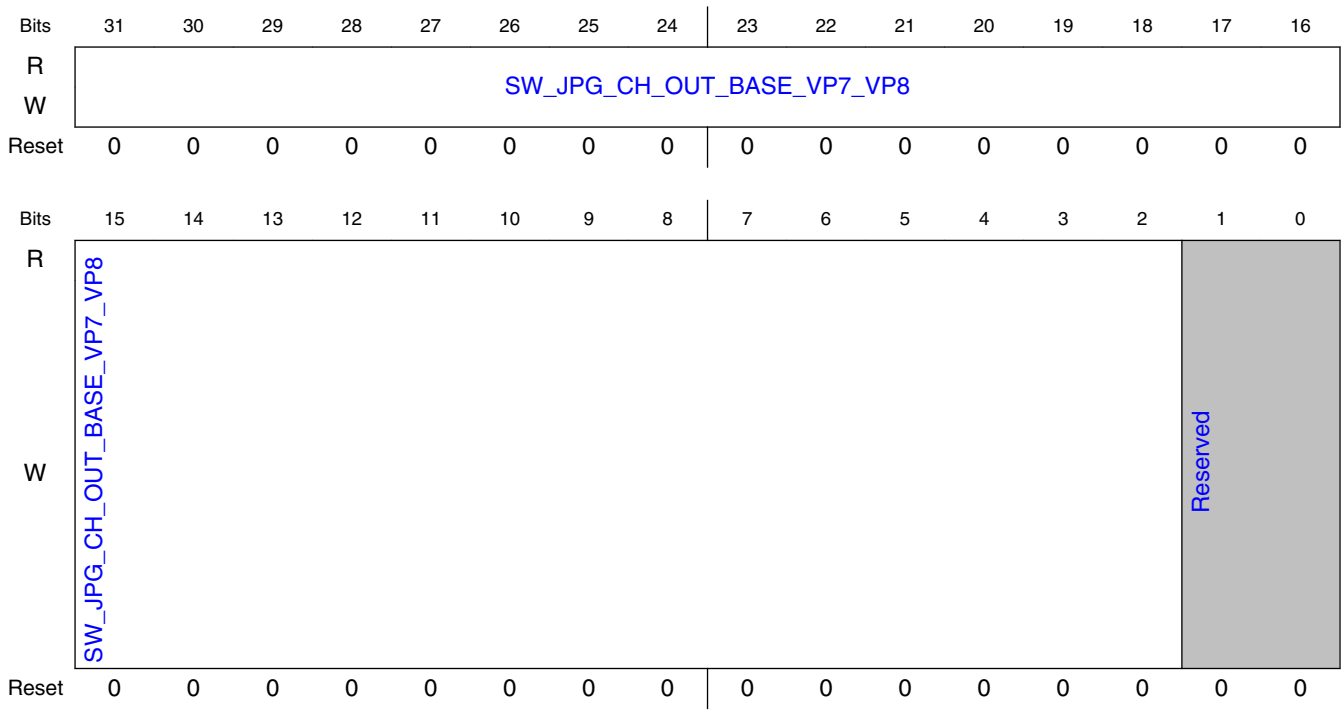
Field	Function
SW_DCT3_START_BIT_VP7_VP8	
23-18 SW_DCT4_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 4
17-12 SW_DCT5_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 5
11-6 SW_DCT6_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 6
5-0 SW_DCT7_START_BIT_VP7_VP8	Start bit for VP7/VP8 DCT stream partition index 7

14.1.5.4.8 Base address for reference picture index 0 / base address for JPEG decoder output chrominance picture (SWREG14_VP7_VP8)

14.1.5.4.8.1 Offset

Register	Offset
SWREG14_VP7_VP8	38h

14.1.5.4.8.2 Diagram



14.1.5.4.8.3 Fields

Field	Function
31-2 SW_JPG_CH_OUT_BASE_VP7_VP8	Base address for decoder output chrominance picture. Used in JPEG and web-p picture mode (not needed if decoder output is not written)
1-0 —	Reserved.

14.1.5.4.9 Base address for reference picture index 1 / JPEG control (SWREG15_VP7_VP8)

14.1.5.4.9.1 Offset

Register	Offset
SWREG15_VP7_VP8	3Ch

14.1.5.4.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved								SW_JPEG_SLICE_H_VP7_VP8							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.1.5.4.9.3 Fields

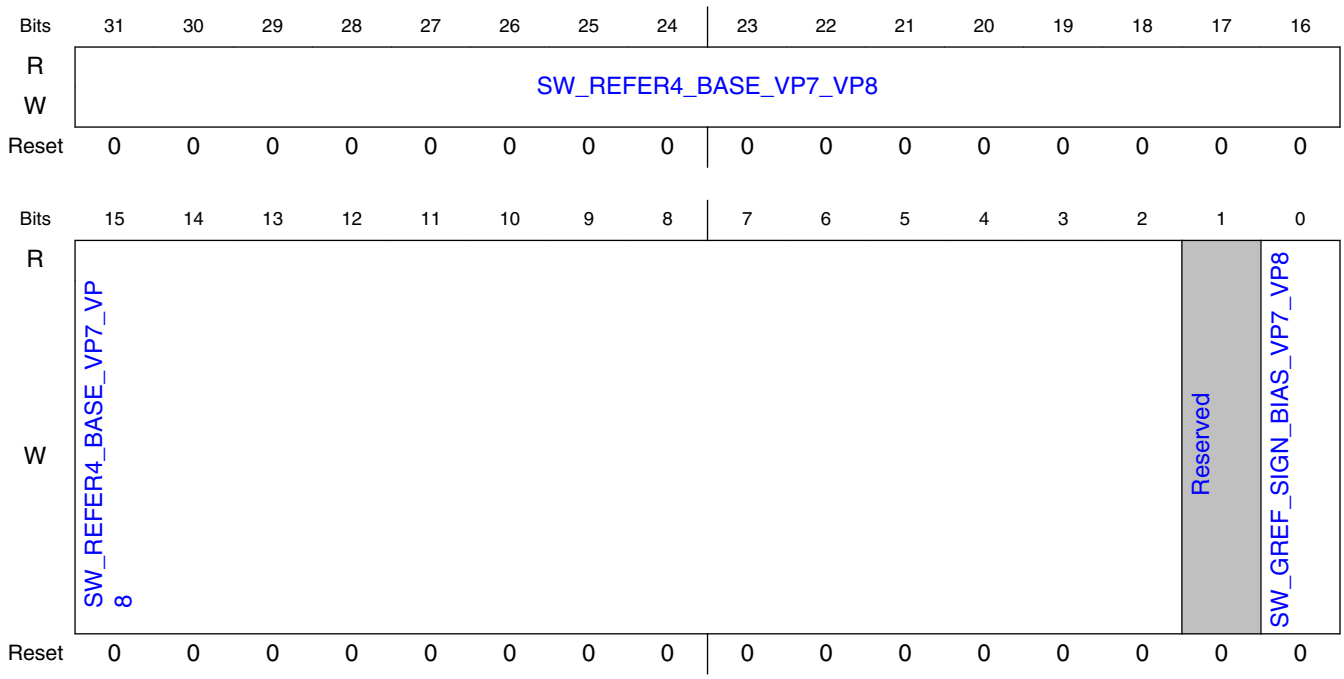
Field	Function
31-8 —	Reserved.
7-0 SW_JPEG_SLICE_H_VP7_VP8	JPEG/Web-p. Height of the slice (multiple of 16 pixels) that HW decodes before interrupt. When slice is decoded HW will rise an interrupt and reset external addresses back to base address. Note, value 0 disables slice mode. Slice mode must be used if picture size is more than 16 Mpixels. However for bigger than 4096 MBs the slice mode usage is recommended.

14.1.5.4.10 Base address for reference picture index 4 / VC1 control / MPEG4 MVD control/ List of VLC code lengths in first JPEG AC table / VC-1 intensity control 4 / VP6/VP7, VP8 Golden refer picture base (SWREG18_VP7_VP8)

14.1.5.4.10.1 Offset

Register	Offset
SWREG18_VP7_VP8	48h

14.1.5.4.10.2 Diagram



14.1.5.4.10.3 Fields

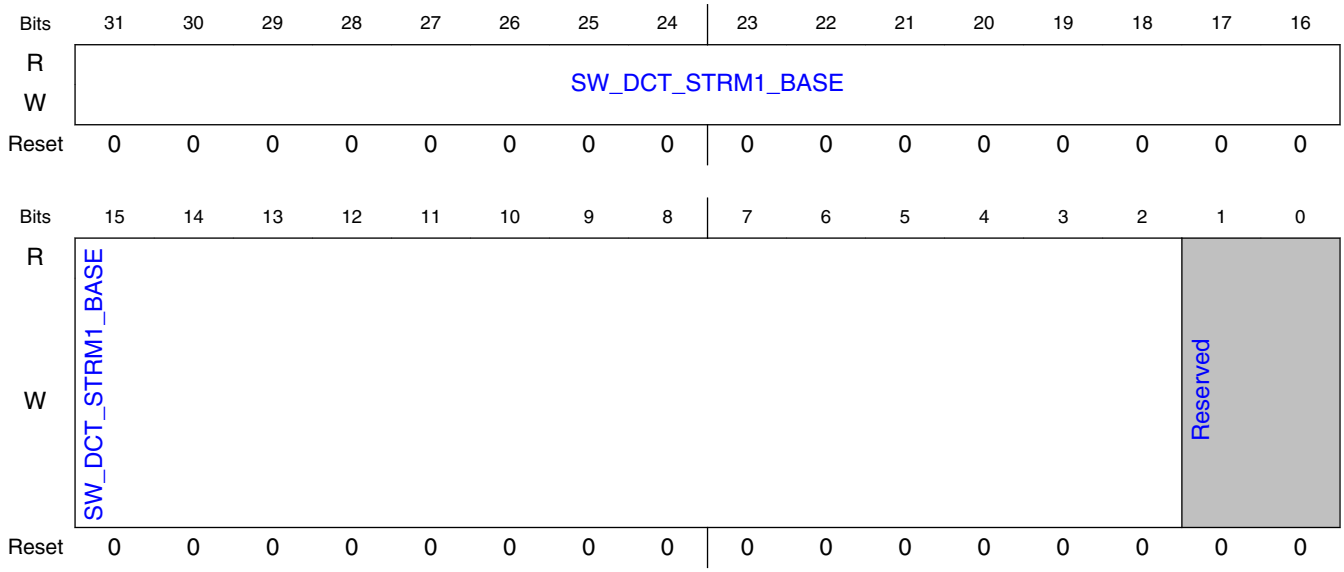
Field	Function
31-2 SW_REFER4_BASE_VP7_VP8	H264: Base address for reference picture index 4 VP6/VP7/VP8: Base address for Golden reference picture (corresponds picid 4)
1 —	Reserved.
0 SW_GREF_SIGN_BIAS_VP7_VP8	Reference picture sign bias for Golden reference frame

14.1.5.4.11 Base address for reference picture index 8 / List of VLC code lengths in second JPEG AC table / VP6 scan maps / VP7,VP8 DCT stream 1 base (SWREG22_VP7_VP8)

14.1.5.4.11.1 Offset

Register	Offset
SWREG22_VP7_VP8	58h

14.1.5.4.11.2 Diagram



14.1.5.4.11.3 Fields

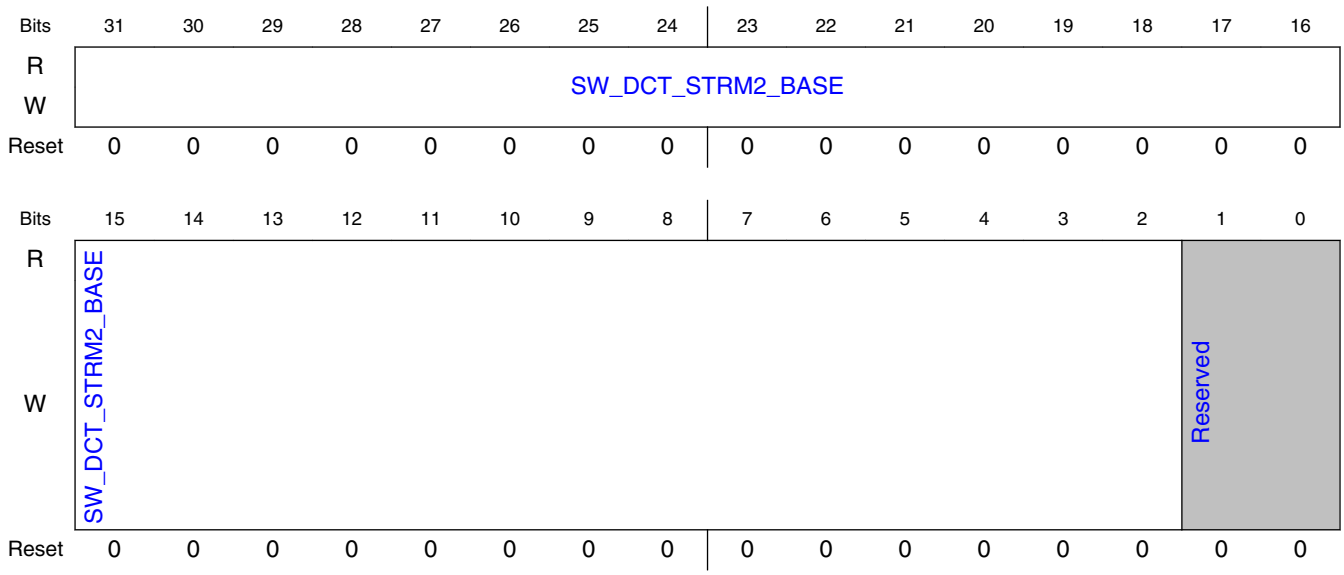
Field	Function
31-2 SW_DCT_STRM1_BASE	Base address for VP7/VP8 DCT stream MB row 1,2n+1
1-0 —	Reserved.

14.1.5.4.12 Base address for reference picture index 9 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 2 base (SWREG23_VP7_VP8)

14.1.5.4.12.1 Offset

Register	Offset
SWREG23_VP7_VP8	5Ch

14.1.5.4.12.2 Diagram



14.1.5.4.12.3 Fields

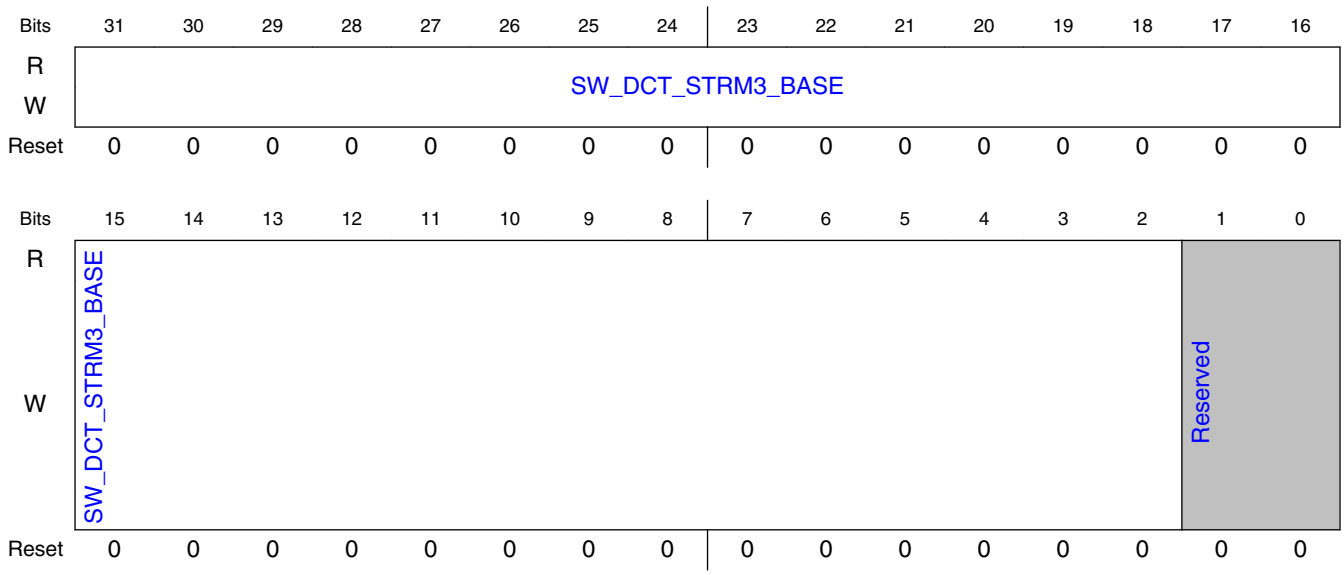
Field	Function
31-2 SW_DCT_STRM2_BASE	Base address for VP7/VP8 DCT stream MB row 2,2n+2
1-0 —	Reserved.

14.1.5.4.13 Base address for reference picture index 10 / List of VLC code lengths in first JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 3 base (SWREG24_VP7_VP8)

14.1.5.4.13.1 Offset

Register	Offset
SWREG24_VP7_VP8	60h

14.1.5.4.13.2 Diagram



14.1.5.4.13.3 Fields

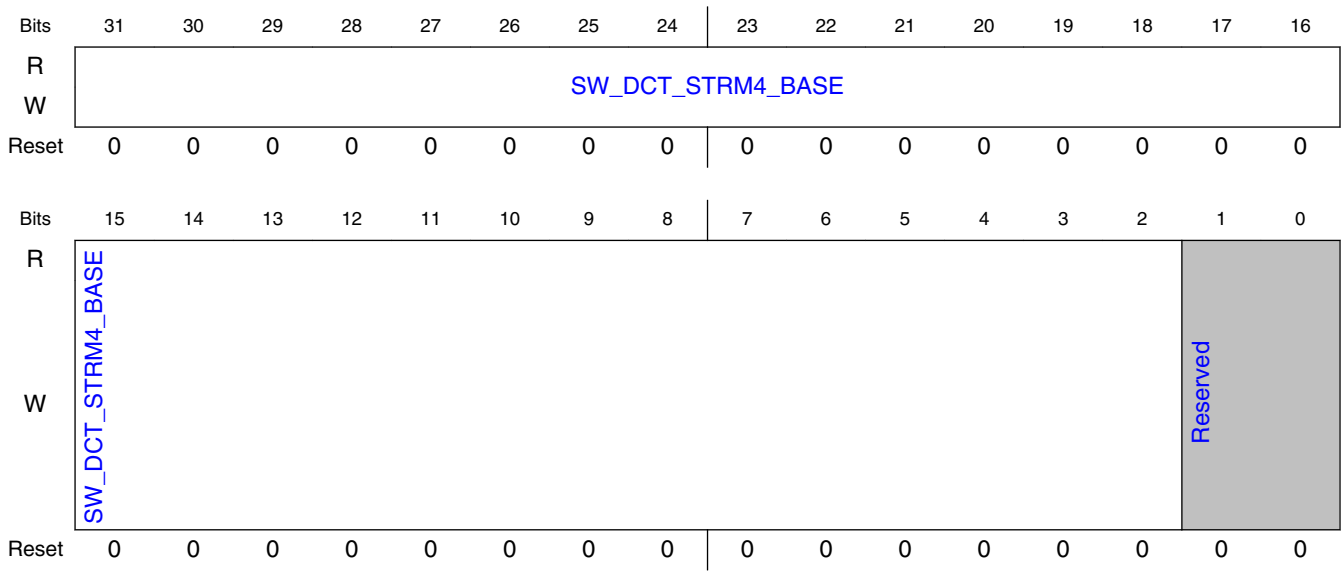
Field	Function
31-2 SW_DCT_STRM3_BASE	Base address for VP7/VP8 DCT stream MB row 3,2n+3
1-0 —	Reserved.

14.1.5.4.14 Base address for reference picture index 11 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 4 base (SWREG25_VP7_VP8)

14.1.5.4.14.1 Offset

Register	Offset
SWREG25_VP7_VP8	64h

14.1.5.4.14.2 Diagram



14.1.5.4.14.3 Fields

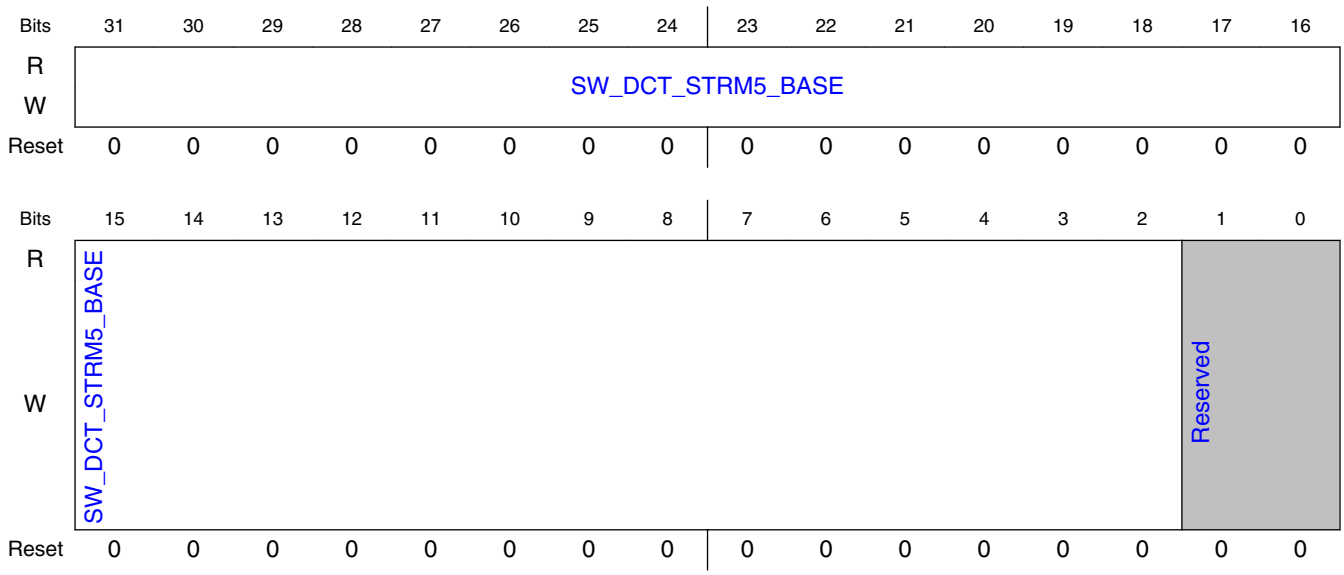
Field	Function
31-2 SW_DCT_STRM4_BASE	Base address for VP7/VP8 DCT stream MB row 4,2n+4
1-0 —	Reserved.

14.1.5.4.15 Base address for reference picture index 12 / List of VLC code lengths in second JPEG DC table / VP6 scan maps / VP7,VP8 DCT stream 5 base (SWREG26_VP7_VP8)

14.1.5.4.15.1 Offset

Register	Offset
SWREG26_VP7_VP8	68h

14.1.5.4.15.2 Diagram



14.1.5.4.15.3 Fields

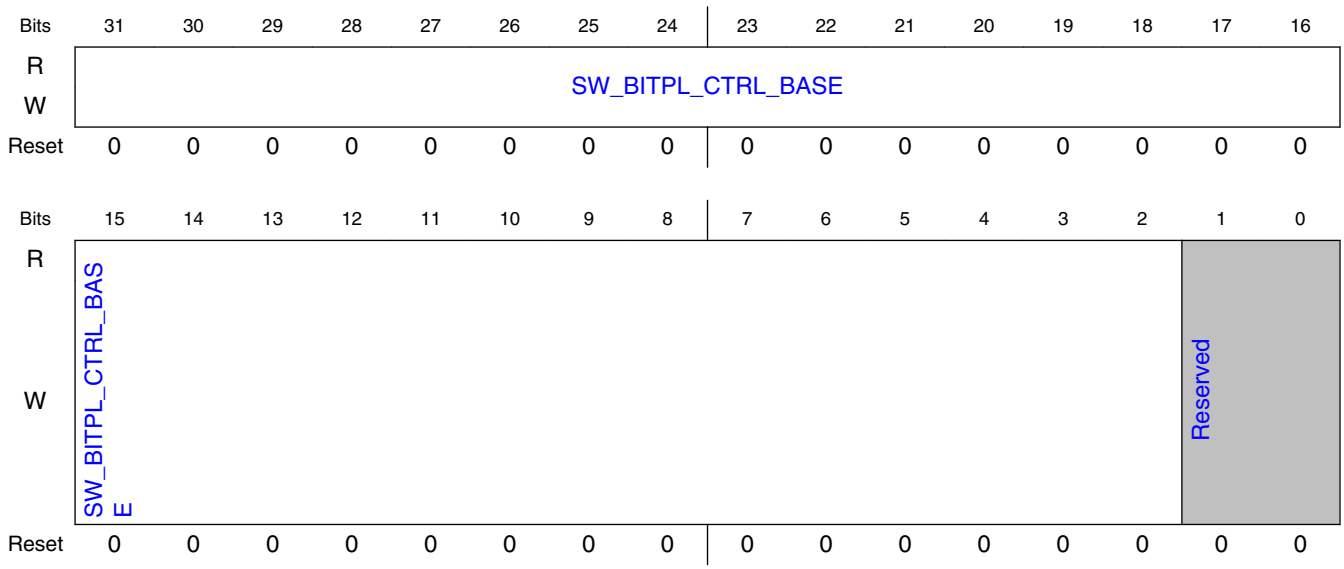
Field	Function
31-2 SW_DCT_STRM5_BASE	Base address for VP7/VP8 DCT stream MB row 5,2n+5
1-0 —	Reserved.

14.1.5.4.16 Base address for reference picture index 13 / VC-1 bitpl mbctrl or VP6,VP7,VP8 ctrl stream base /Progressive JPEG DC table (SWREG27_VC1)

14.1.5.4.16.1 Offset

Register	Offset
SWREG27_VC1	6Ch

14.1.5.4.16.2 Diagram



14.1.5.4.16.3 Fields

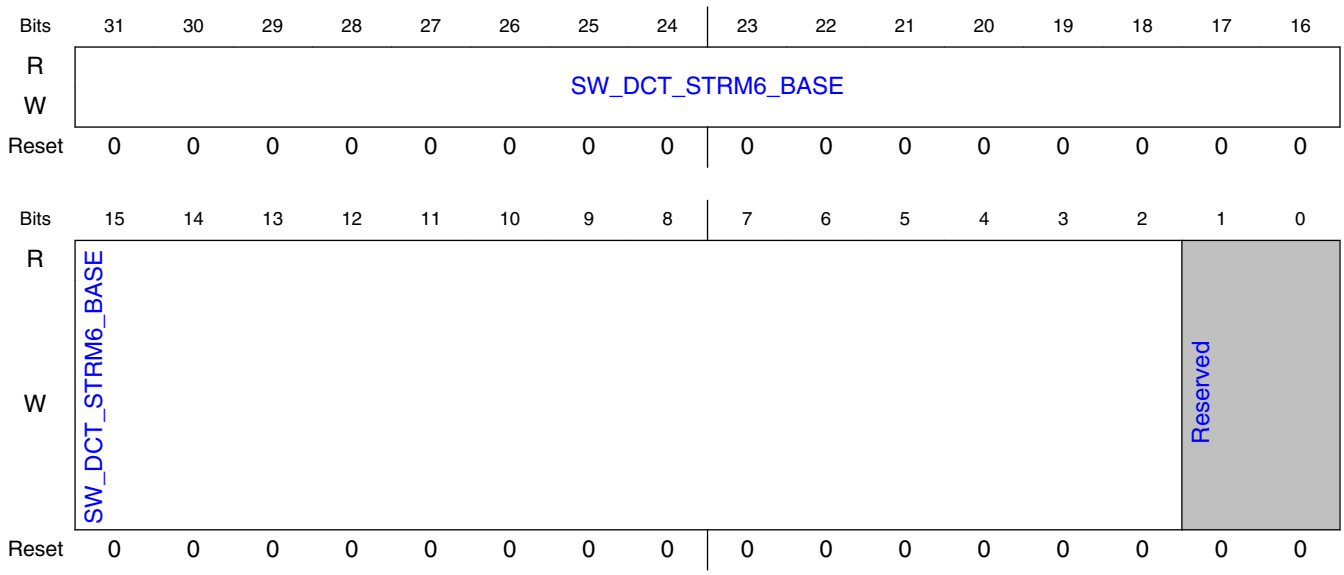
Field	Function
31-2 SW_BITPL_CTRL_BASE	VC-1: Base address for bitplane mb control VP6/VP7/VP8 : Base address for ctrl data stream. Used if multistream is enabled
1-0 —	Reserved.

14.1.5.4.17 Base address for reference picture index 14 / VP6 scan maps / Progressive JPEG DC table / VP7,VP8 DCT stream 6 base (SWREG28_VP7_VP8)

14.1.5.4.17.1 Offset

Register	Offset
SWREG28_VP7_VP8	70h

14.1.5.4.17.2 Diagram



14.1.5.4.17.3 Fields

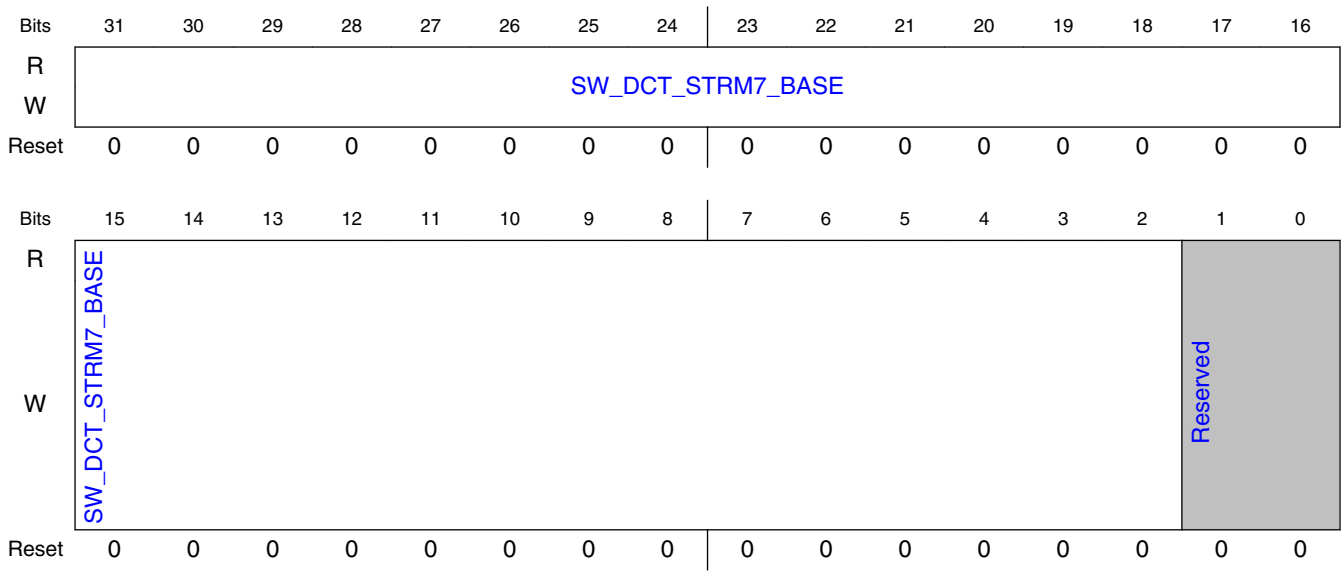
Field	Function
31-2 SW_DCT_STRM6_BASE	Base address for VP7/VP8 DCT stream MB row 6,2n+6
1-0 —	Reserved.

14.1.5.4.18 Base address for reference picture index 15 / VP6 scan maps / VP7,VP8 DCT stream 7 base (SWREG29_VP7_VP8)

14.1.5.4.18.1 Offset

Register	Offset
SWREG29_VP7_VP8	74h

14.1.5.4.18.2 Diagram



14.1.5.4.18.3 Fields

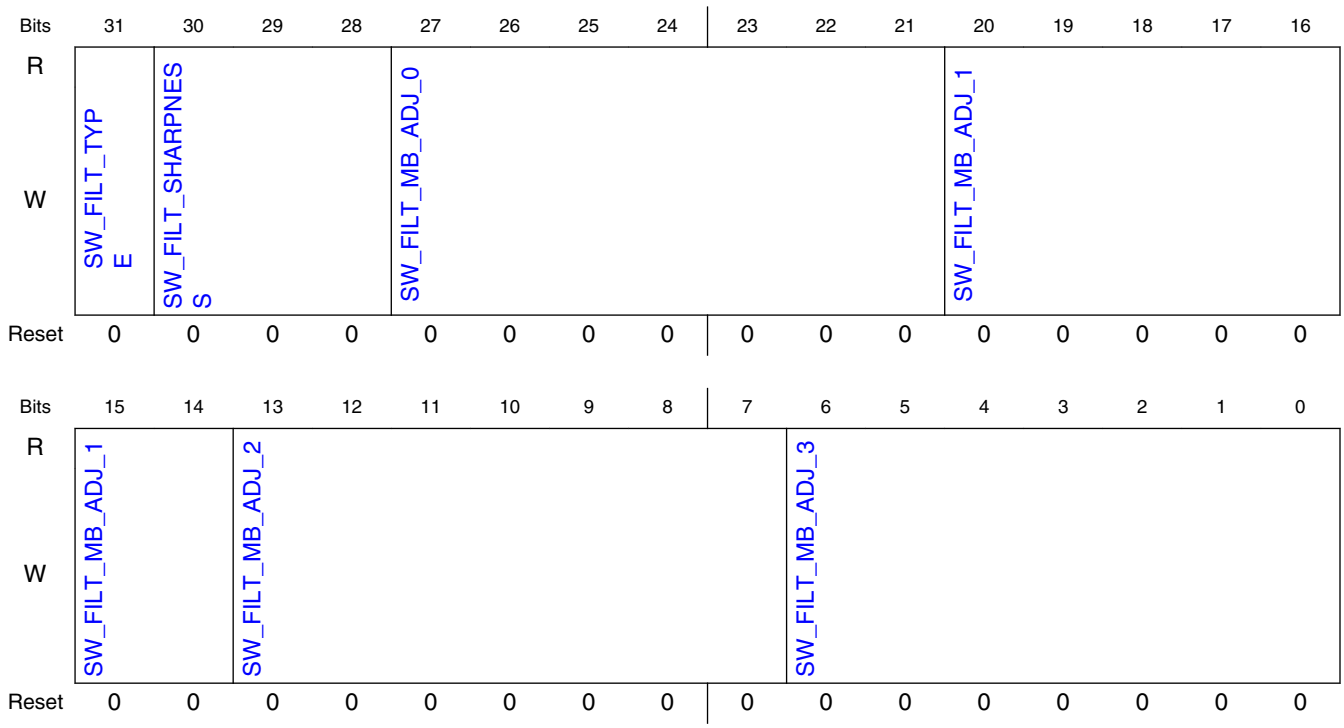
Field	Function
31-2 SW_DCT_STRM7_BASE	Base address for VP7/VP8 DCT stream MB row 7,2n+7
1-0 —	Reserved.

14.1.5.4.19 Reference picture numbers for index 0 and 1 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter mb level adjusts (SWREG30_VP7_VP8)

14.1.5.4.19.1 Offset

Register	Offset
SWREG30_VP7_VP8	78h

14.1.5.4.19.2 Diagram



14.1.5.4.19.3 Fields

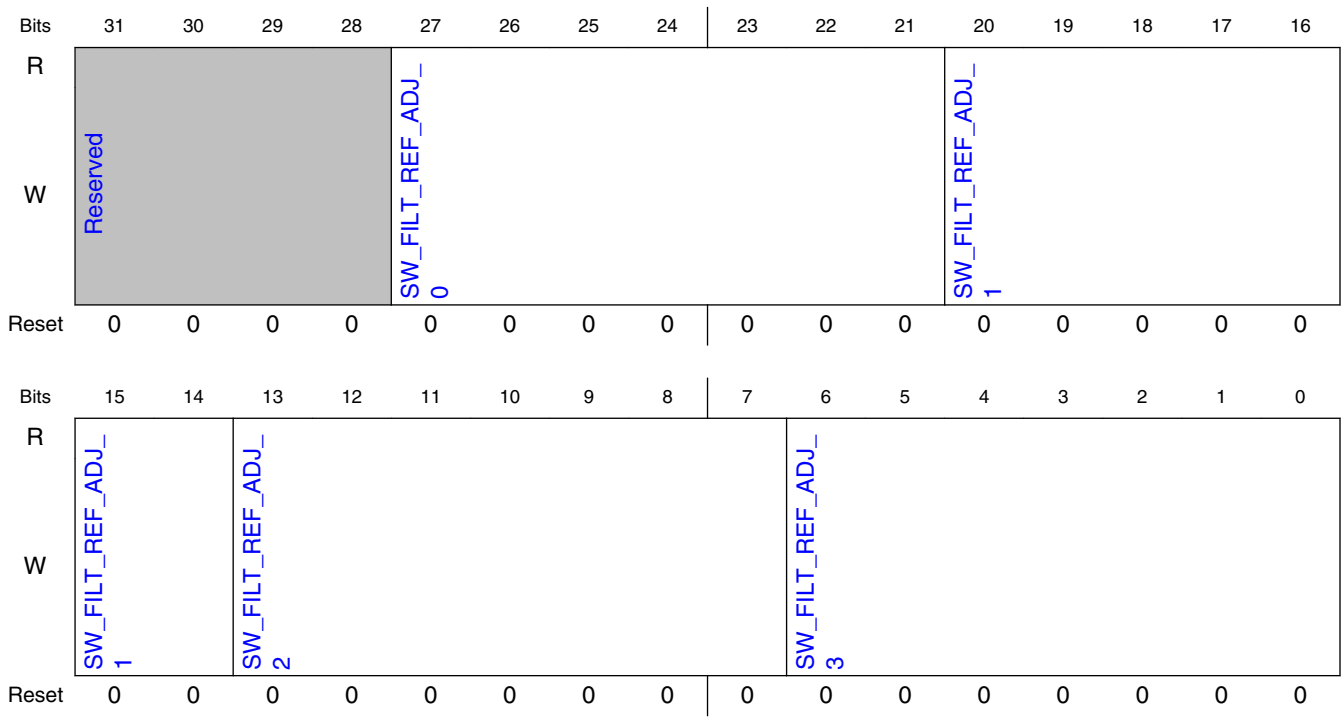
Field	Function
31 SW_FILT_TYPE	VP7/VP8 loop filter type
30-28 SW_FILT_SHARPNESS	VP7/VP8 loop filter sharpness
27-21 SW_FILT_MB_ADJ_0	VP7/VP8 filter level adjustment for MB type 0
20-14 SW_FILT_MB_ADJ_1	VP7/VP8 filter level adjustment for MB type 1
13-7 SW_FILT_MB_ADJ_2	VP7/VP8 filter level adjustment for MB type 2
6-0 SW_FILT_MB_ADJ_3	VP7/VP8 filter level adjustment for MB type 3

14.1.5.4.20 Reference picture numbers for index 2 and 3 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter ref pic level adjusts (SWREG31_VP7_VP8)

14.1.5.4.20.1 Offset

Register	Offset
SWREG31_VP7_VP8	7Ch

14.1.5.4.20.2 Diagram



14.1.5.4.20.3 Fields

Field	Function
31-28 —	Reserved.
27-21 SW_FILT_REF_ADJ_0	VP7/VP8 filter level adjustment for reference frame type 0
20-14	VP7/VP8 filter level adjustment for reference frame type 1

Table continues on the next page...

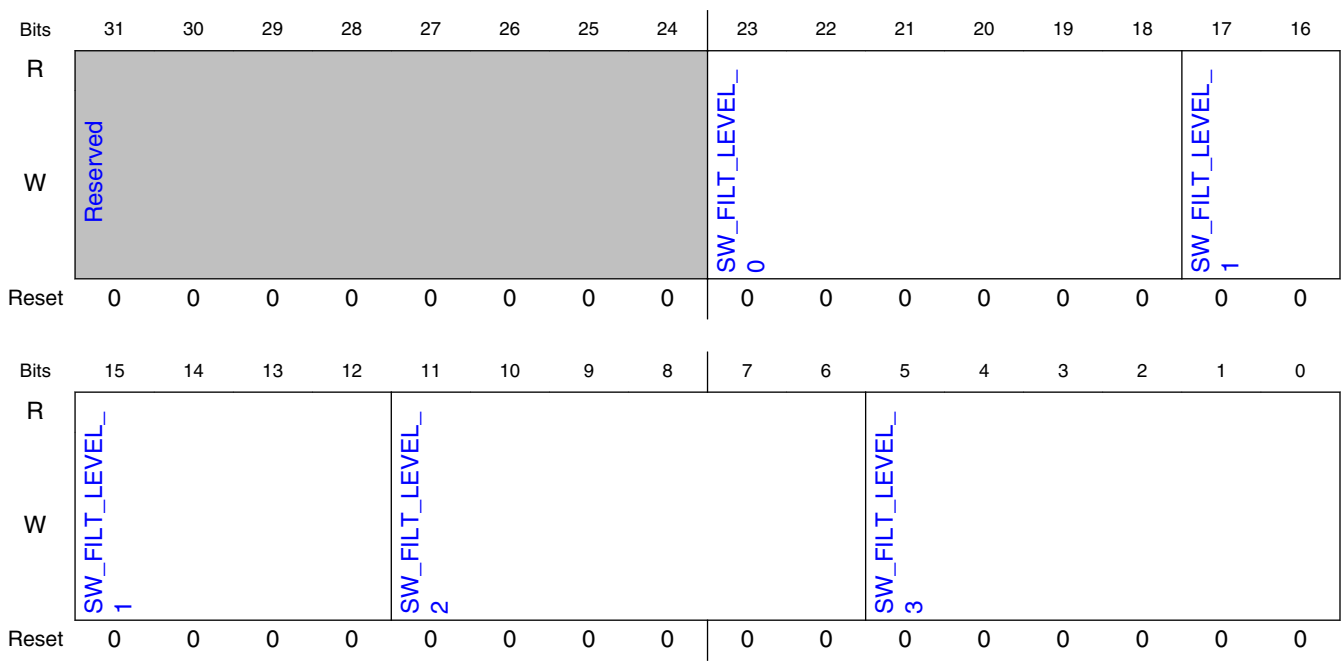
Field	Function
SW_FILT_REF_ADJ_1	
13-7 SW_FILT_REF_ADJ_2	VP7/VP8 filter level adjustment for reference frame type 2
6-0 SW_FILT_REF_ADJ_3	VP7/VP8 filter level adjustment for reference frame type 3

14.1.5.4.21 Reference picture numbers for index 4 and 5 (H264 VLC) / VP6 scan maps / VP7,VP8 loop filter levels (SWREG32_VP7_VP8)

14.1.5.4.21.1 Offset

Register	Offset
SWREG32_VP7_VP8	80h

14.1.5.4.21.2 Diagram



14.1.5.4.21.3 Fields

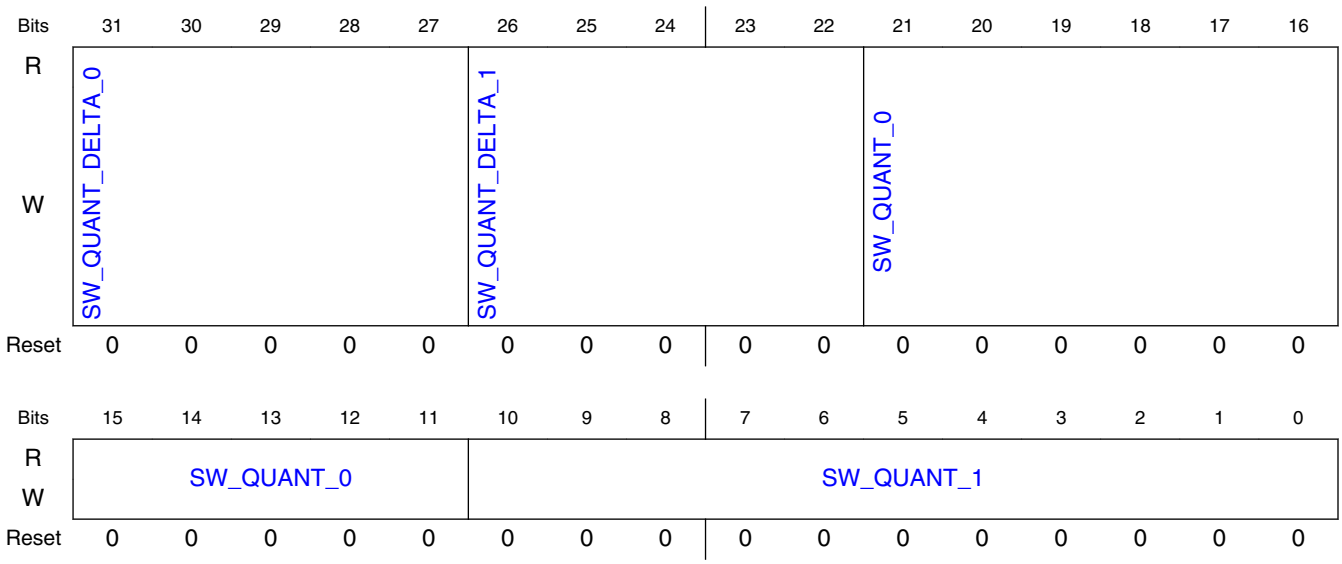
Field	Function
31-24 —	Reserved.
23-18 SW_FILT_LEVEL_0	VP7/VP8 filter level value for reference frame type 0
17-12 SW_FILT_LEVEL_1	VP7/VP8 filter level value for reference frame type 1
11-6 SW_FILT_LEVEL_2	VP7/VP8 filter level value for reference frame type 2
5-0 SW_FILT_LEVEL_3	VP7/VP8 filter level value for reference frame type 3

14.1.5.4.22 Reference picture numbers for index 6 and 7 (H264 VLC) / VP6 scan maps / VP7,VP8 quantization values (SWREG33_VP7_VP8)

14.1.5.4.22.1 Offset

Register	Offset
SWREG33_VP7_VP8	84h

14.1.5.4.22.2 Diagram



14.1.5.4.22.3 Fields

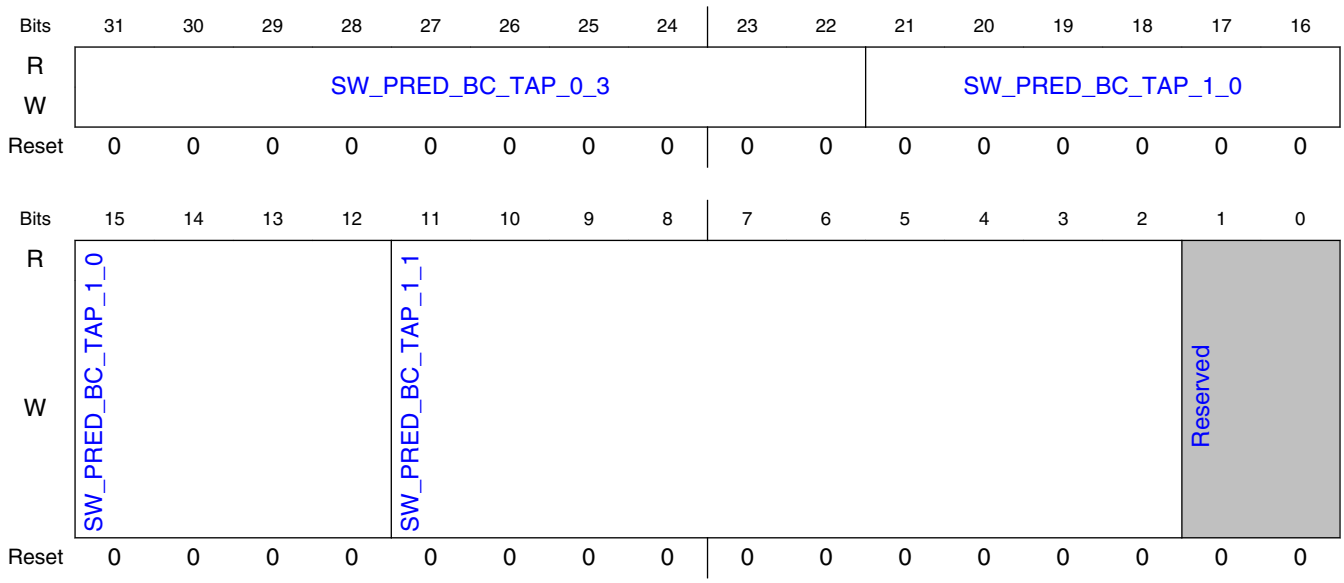
Field	Function
31-27 SW_QUANT_DELTA_0	VP8 quantisizer delta 0
26-22 SW_QUANT_DELTA_1	VP8 quantisizer delta 1
21-11 SW_QUANT_0	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)
10-0 SW_QUANT_1	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)

14.1.5.4.23 Reference picture numbers for index 8 and 9 (H264 VLC) / MPEG4, VC1, VPx prediction filter taps (SWREG34_H263)

14.1.5.4.23.1 Offset

Register	Offset
SWREG34_H263	88h

14.1.5.4.23.2 Diagram



14.1.5.4.23.3 Fields

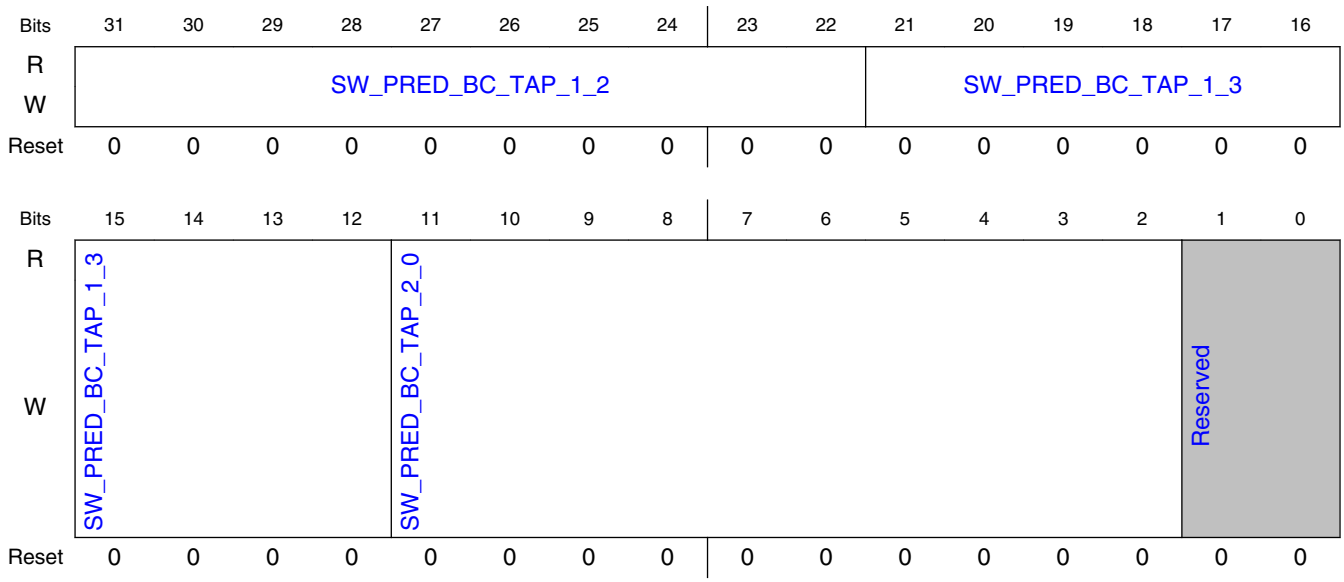
Field	Function
31-22 SW_PRED_BC_TAP_0_3	Prediction filter set 0, tap 3
21-12 SW_PRED_BC_TAP_1_0	Prediction filter set 1, tap 0
11-2 SW_PRED_BC_TAP_1_1	Prediction filter set 1, tap 1
1-0 —	Reserved.

14.1.5.4.24 Reference picture numbers for index 10 and 11 (H264 VLC) / VC1, VPx prediction filter taps (SWREG35_VC1)

14.1.5.4.24.1 Offset

Register	Offset
SWREG35_VC1	8Ch

14.1.5.4.24.2 Diagram



14.1.5.4.24.3 Fields

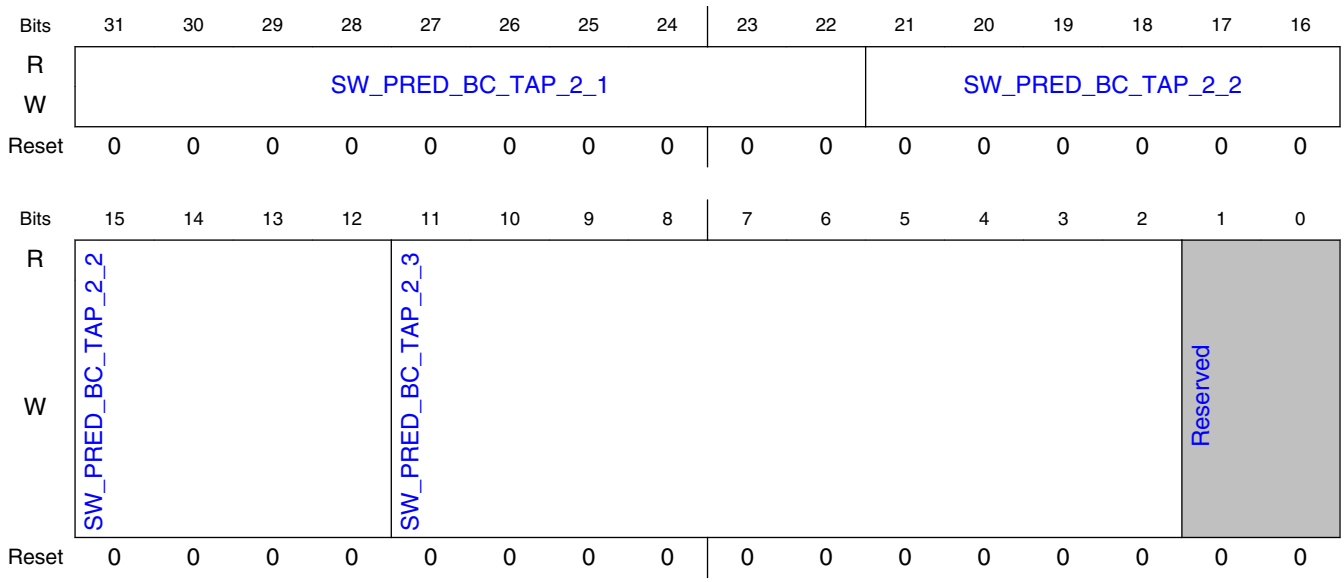
Field	Function
31-22 SW_PRED_BC_TAP_1_2	Prediction filter set 1, tap 2
21-12 SW_PRED_BC_TAP_1_3	Prediction filter set 1, tap 3
11-2 SW_PRED_BC_TAP_2_0	Prediction filter set 2, tap 0
1-0 —	Reserved.

14.1.5.4.25 Reference picture numbers for index 12 and 13 (H264 VLC) / VC1, VPx prediction filter taps (SWREG36_VC1)

14.1.5.4.25.1 Offset

Register	Offset
SWREG36_VC1	90h

14.1.5.4.25.2 Diagram



14.1.5.4.25.3 Fields

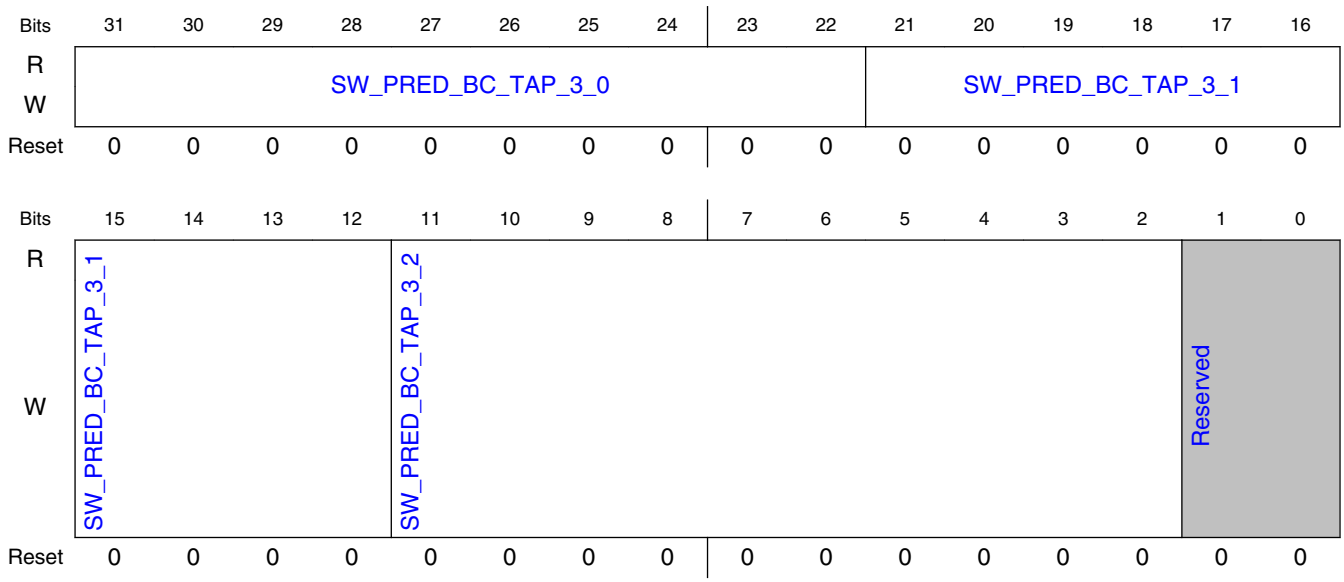
Field	Function
31-22 SW_PRED_BC_TAP_2_1	Prediction filter set 2, tap 1
21-12 SW_PRED_BC_TAP_2_2	Prediction filter set 2, tap 2
11-2 SW_PRED_BC_TAP_2_3	Prediction filter set 2, tap 3
1-0 —	Reserved.

14.1.5.4.26 Reference picture numbers for index 14 and 15 (H264 VLC) / VPx prediction filter taps (SWREG37_VP6_VP7_VP8)

14.1.5.4.26.1 Offset

Register	Offset
SWREG37_VP6_VP7_VP8	94h

14.1.5.4.26.2 Diagram



14.1.5.4.26.3 Fields

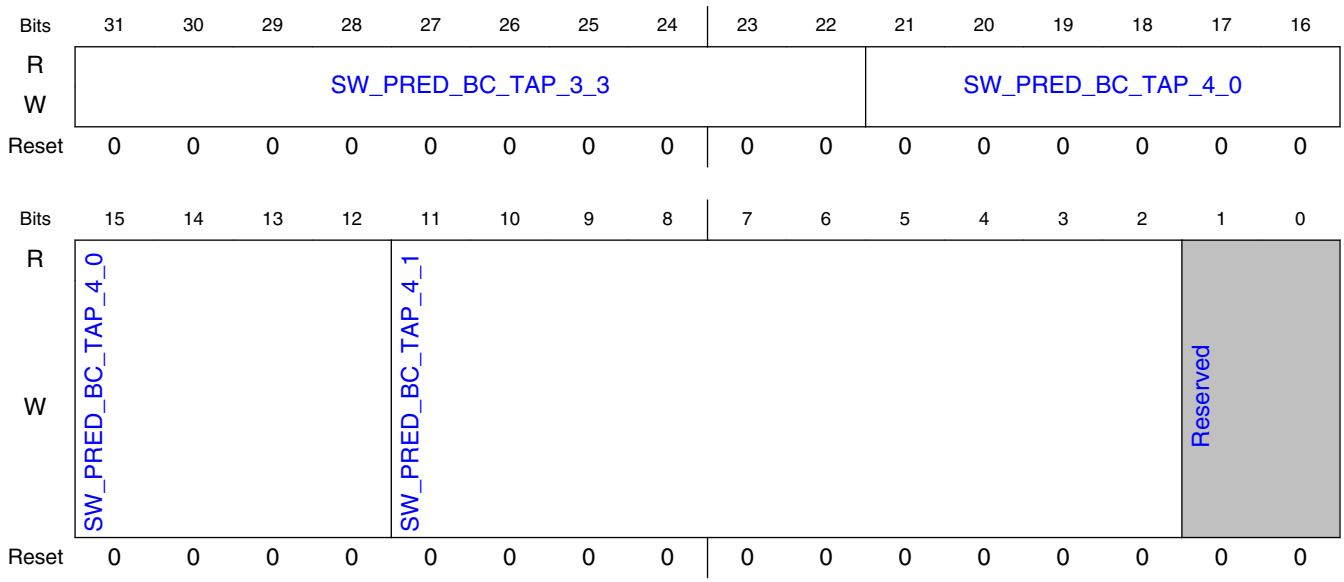
Field	Function
31-22 SW_PRED_BC_TAP_3_0	Prediction filter set 3, tap 0
21-12 SW_PRED_BC_TAP_3_1	Prediction filter set 3, tap 1
11-2 SW_PRED_BC_TAP_3_2	Prediction filter set 3, tap 2
1-0 —	Reserved.

14.1.5.4.27 Reference picture long term flags (H264 VLC) / VPx prediction filter taps (SWREG38_VP6_VP7_VP8)

14.1.5.4.27.1 Offset

Register	Offset
SWREG38_VP6_VP7_VP8	98h

14.1.5.4.27.2 Diagram



14.1.5.4.27.3 Fields

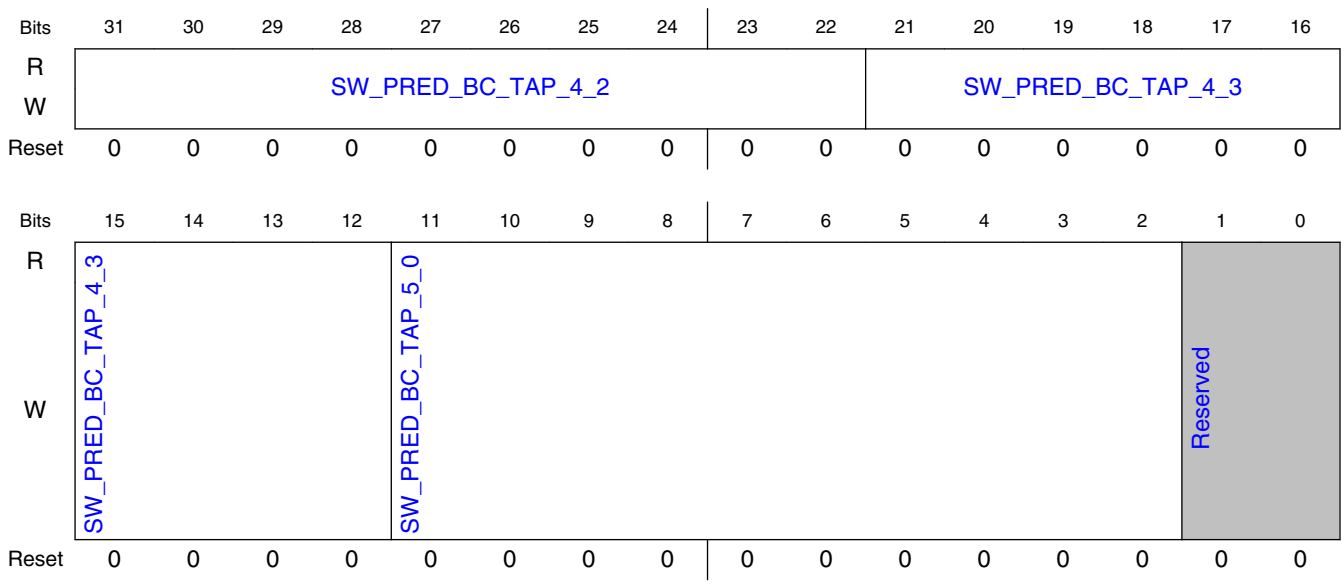
Field	Function
31-22 SW_PRED_BC_TAP_3_3	Prediction filter set 3, tap 3
21-12 SW_PRED_BC_TAP_4_0	Prediction filter set 4, tap 0
11-2 SW_PRED_BC_TAP_4_1	Prediction filter set 4, tap 1
1-0 —	Reserved.

14.1.5.4.28 Reference picture valid flags (H264 VLC) / VPx prediction filter taps (SWREG39_VP6_VP7_VP8)

14.1.5.4.28.1 Offset

Register	Offset
SWREG39_VP6_VP7_VP8	9Ch

14.1.5.4.28.2 Diagram



14.1.5.4.28.3 Fields

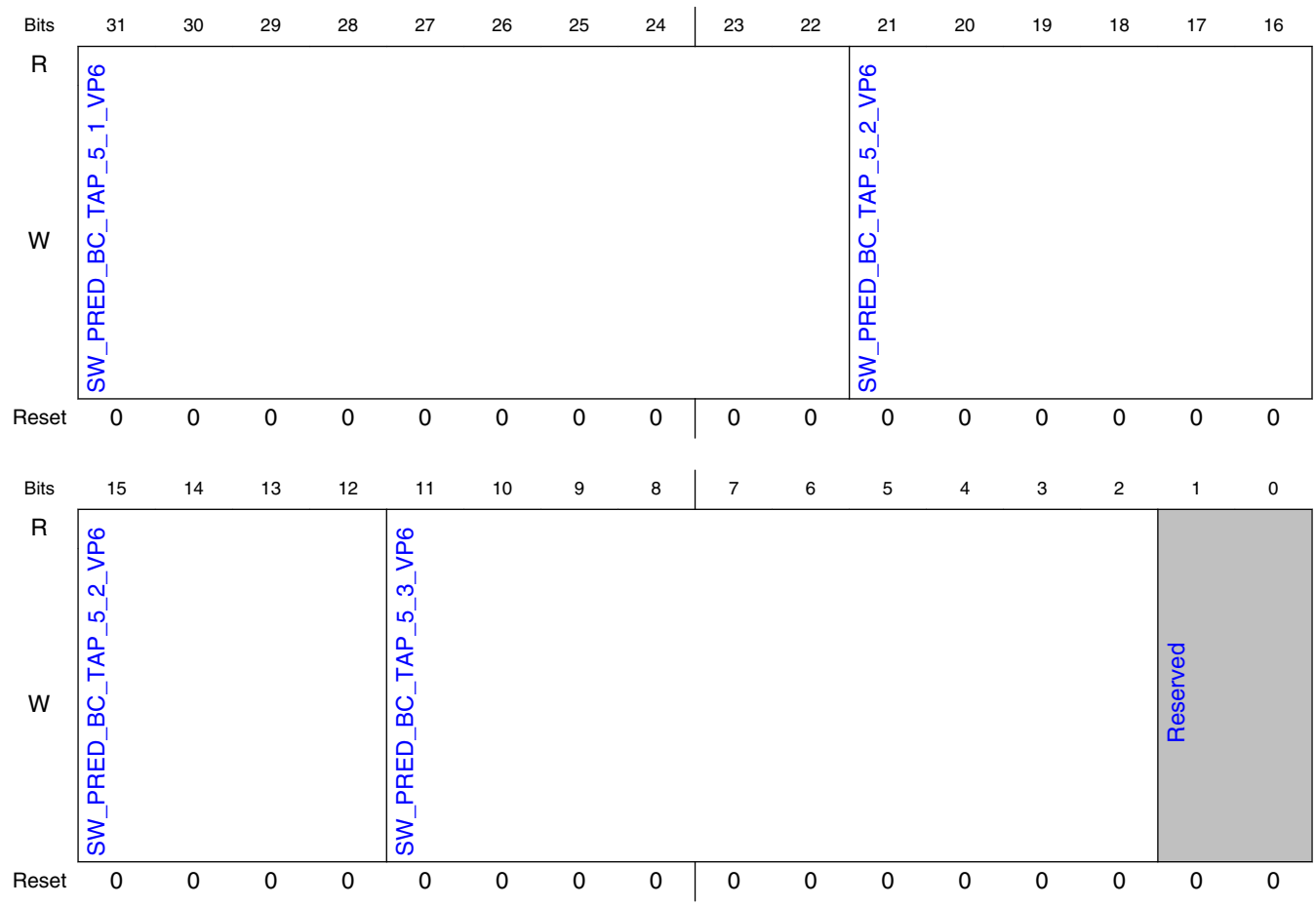
Field	Function
31-22 SW_PRED_BC_TAP_4_2	Prediction filter set 4, tap 2
21-12 SW_PRED_BC_TAP_4_3	Prediction filter set 4, tap 3
11-2 SW_PRED_BC_TAP_5_0	Prediction filter set 5, tap 0
1-0 —	Reserved.

14.1.5.4.29 bi_dir initial ref pic list register (0-2) / VP6 prediction filter taps / Progressive JPEG Cb ACDC coefficient base (SWREG42_VP6)

14.1.5.4.29.1 Offset

Register	Offset
SWREG42_VP6	A8h

14.1.5.4.29.2 Diagram



14.1.5.4.29.3 Fields

Field	Function
31-22	Prediction filter set 5, tap 1

Table continues on the next page...

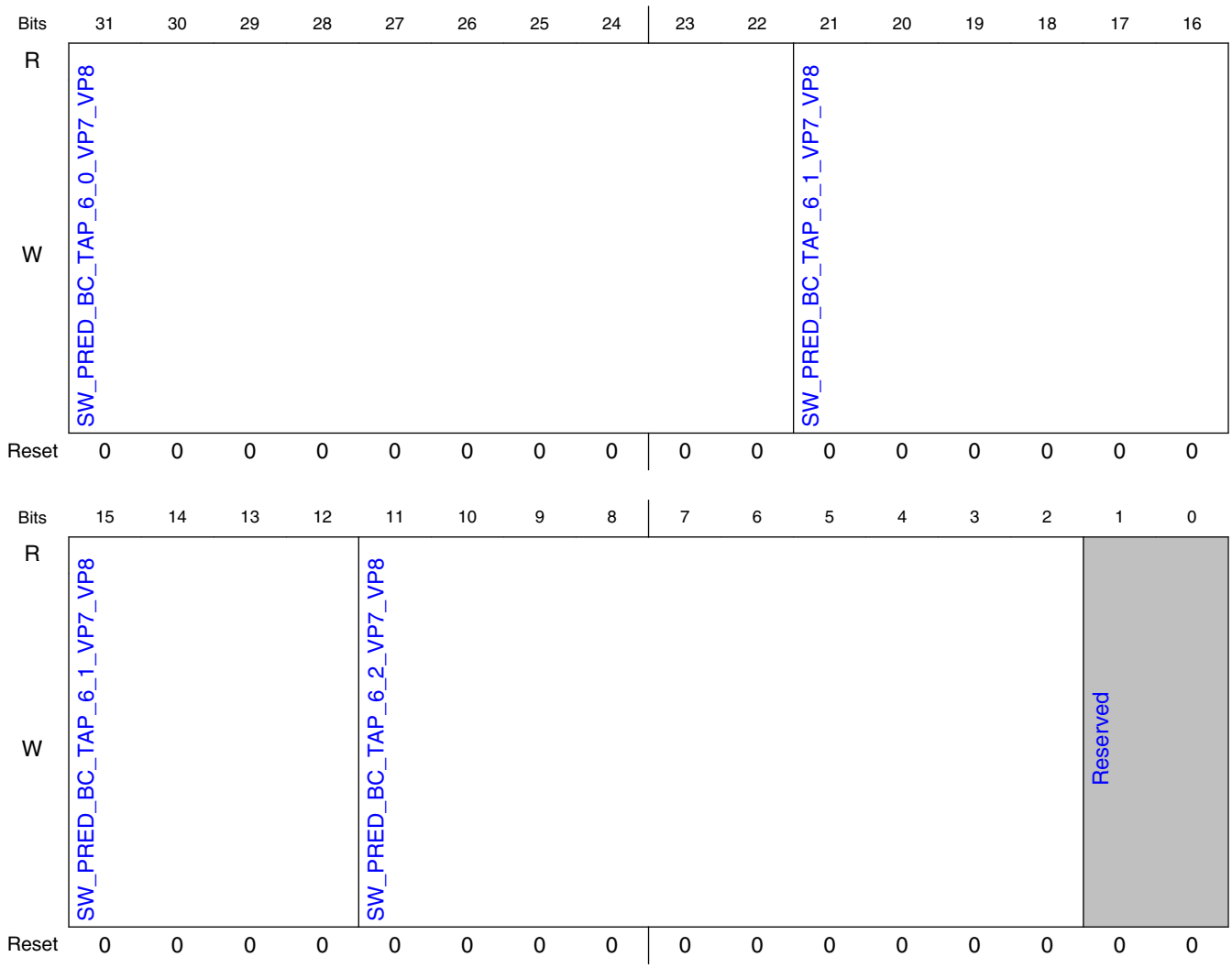
Field	Function
SW_PRED_BC_TAP_5_1_VP6	
21-12 SW_PRED_BC_TAP_5_2_VP6	Prediction filter set 5, tap 2
11-2 SW_PRED_BC_TAP_5_3_VP6	Prediction filter set 5, tap 3
1-0 —	Reserved.

14.1.5.4.30 bi-dir initial ref pic list register (3-5) / VP6 prediction filter taps / Progressive JPEG Cr ACDC coefficient base (SWREG43_VP7_VP8)

14.1.5.4.30.1 Offset

Register	Offset
SWREG43_VP7_VP8	ACh

14.1.5.4.30.2 Diagram



14.1.5.4.30.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_6_0_VP7_VP8	Prediction filter set 6, tap 0
21-12 SW_PRED_BC_TAP_6_1_VP7_VP8	Prediction filter set 6, tap 1
11-2	Prediction filter set 6, tap 2

Table continues on the next page...

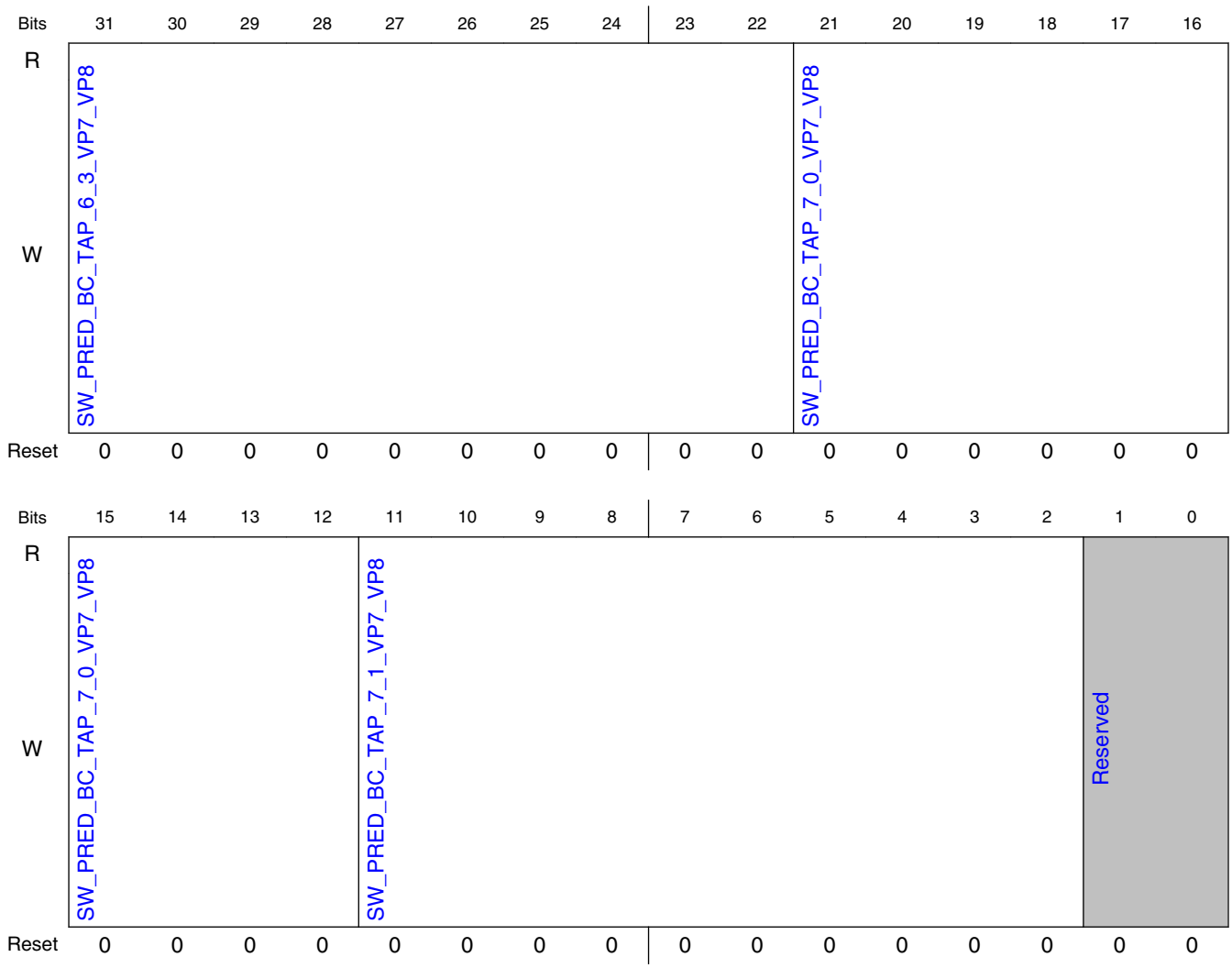
Field	Function
SW_PRED_BC_TAP_6_2_VP7_VP8	
1-0 —	Reserved.

14.1.5.4.31 bi-dir initial ref pic list register (6-8) / VP6 prediction filter taps (SWREG44_VP7_VP8)

14.1.5.4.31.1 Offset

Register	Offset
SWREG44_VP7_VP8	B0h

14.1.5.4.31.2 Diagram



14.1.5.4.31.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_6_3_VP7_VP8	Prediction filter set 6, tap 3
21-12 SW_PRED_BC_TAP_7_0_VP7_VP8	Prediction filter set 7, tap 0
11-2	Prediction filter set 7, tap 1

Table continues on the next page...

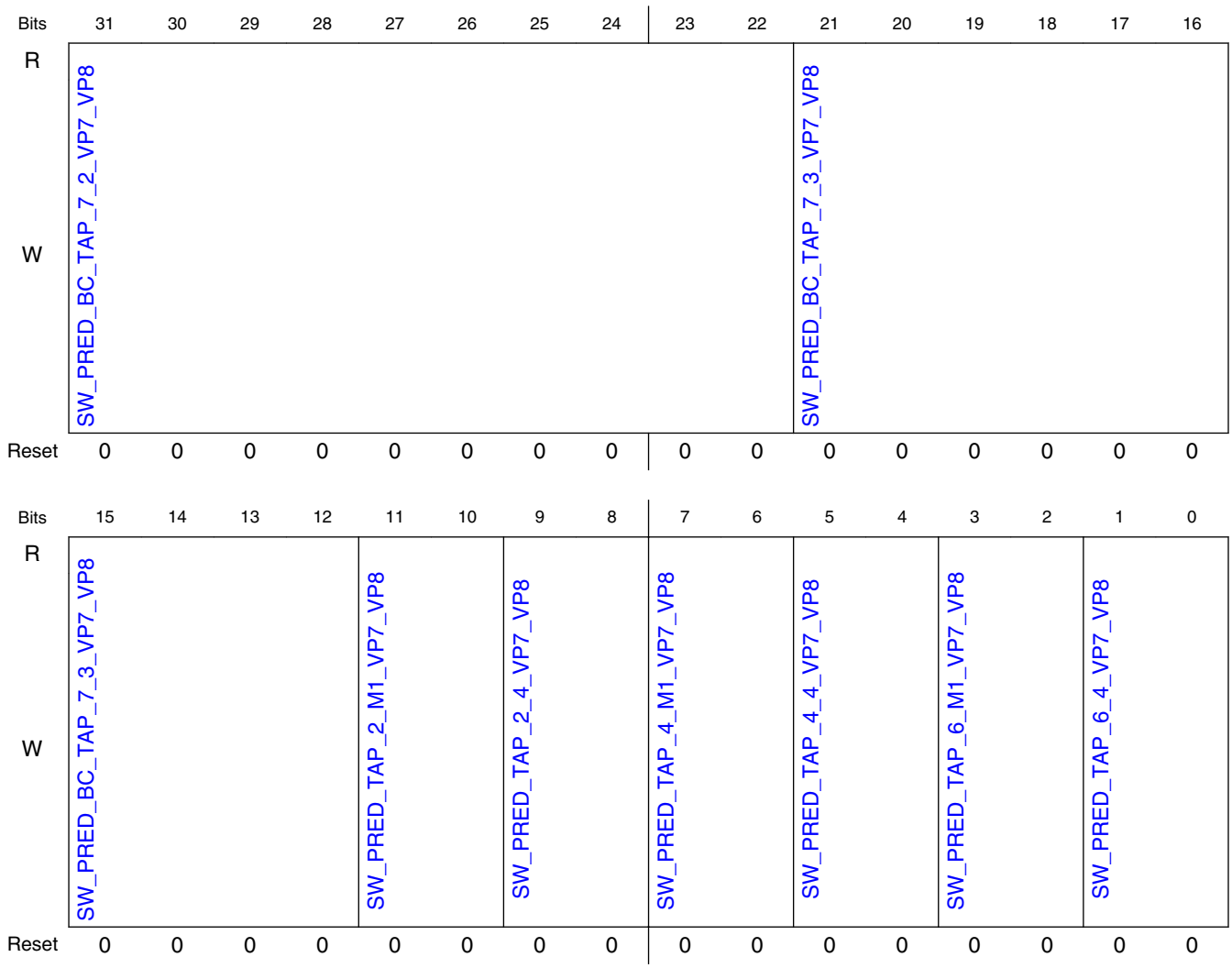
Field	Function
SW_PRED_BC_TAP_7_1_VP7_VP8	
1-0 —	Reserved.

14.1.5.4.32 bi-dir initial ref pic list register (9-11) / VP6 prediction filter taps (SWREG45_VP7_VP8)

14.1.5.4.32.1 Offset

Register	Offset
SWREG45_VP7_VP8	B4h

14.1.5.4.32.2 Diagram



14.1.5.4.32.3 Fields

Field	Function
31-22 SW_PRED_BC_TAP_7_2_VP7_VP8	Prediction filter set 7, tap 2
21-12 SW_PRED_BC_TAP_7_3_VP7_VP8	Prediction filter set 7, tap 3
11-10	Additional Prediction filter tap -1 for set 2

Table continues on the next page...

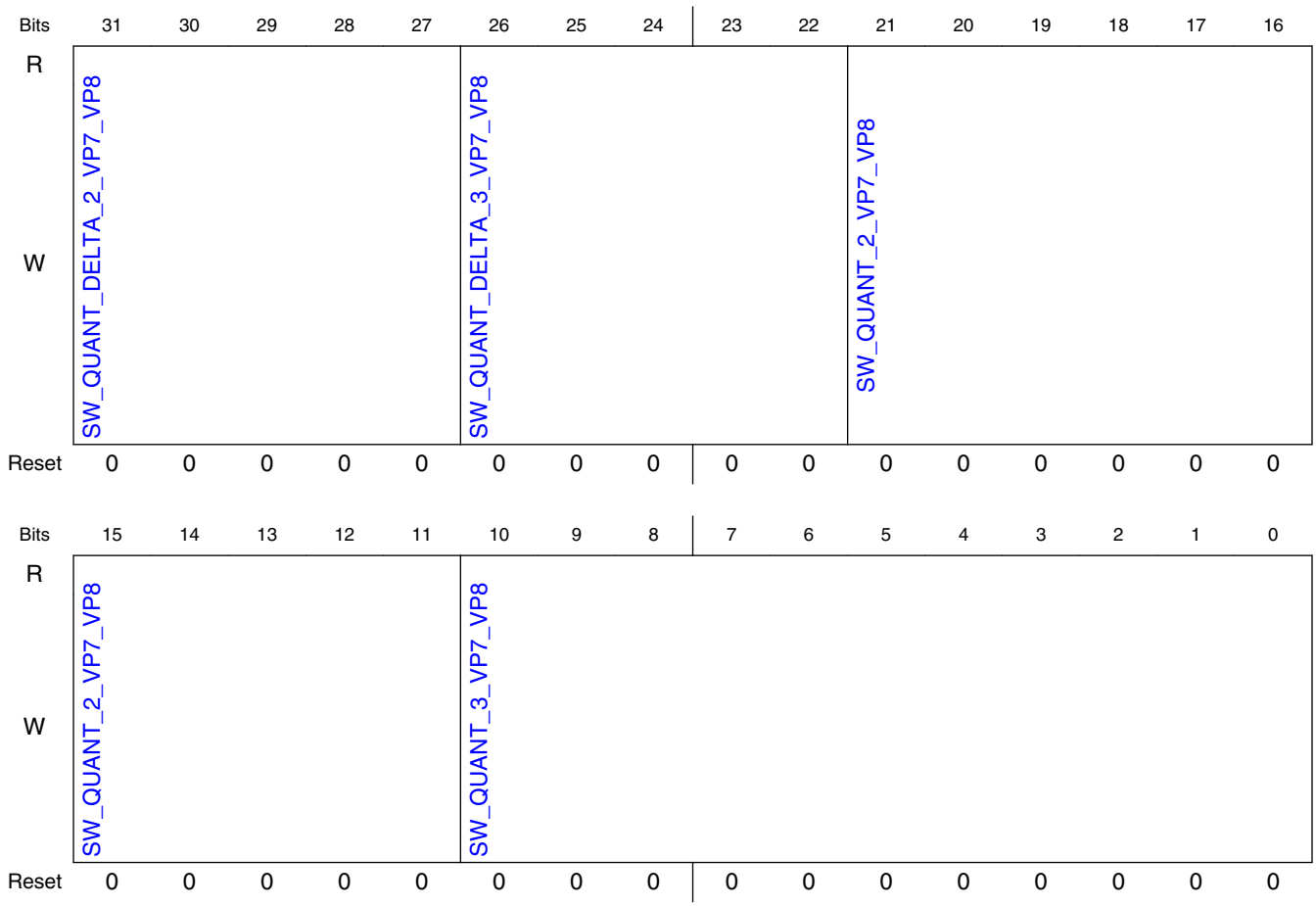
Field	Function
SW_PRED_TAP_2_M1_VP7_VP8	
9-8 SW_PRED_TAP_2_4_VP7_VP8	Additional Prediction filter tap 4 for set 2
7-6 SW_PRED_TAP_4_M1_VP7_VP8	Additional Prediction filter tap -1 for set 4
5-4 SW_PRED_TAP_4_4_VP7_VP8	Additional Prediction filter tap 4 for set 4
3-2 SW_PRED_TAP_6_M1_VP7_VP8	Additional Prediction filter tap -1 for set 6
1-0 SW_PRED_TAP_6_4_VP7_VP8	Additional Prediction filter tap 4 for set 6

14.1.5.4.33 bi-dir initial ref pic list register (12-14) / VP7,VP8 quantization values (SWREG46_VP7_VP8)

14.1.5.4.33.1 Offset

Register	Offset
SWREG46_VP7_VP8	B8h

14.1.5.4.33.2 Diagram



14.1.5.4.33.3 Fields

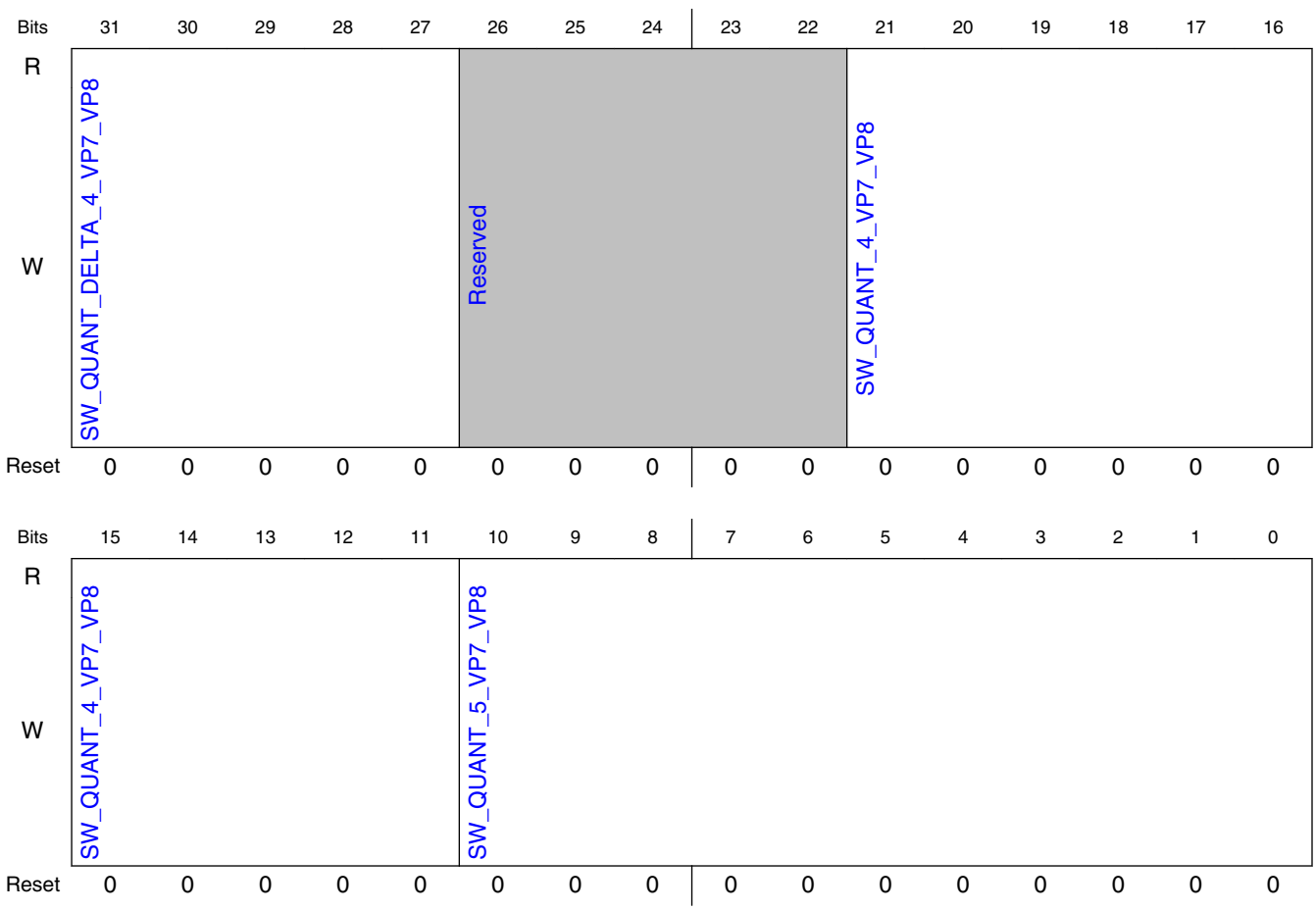
Field	Function
31-27 SW_QUANT_DELTA_2_VP7_VP8	VP8 quantisizer delta 2
26-22 SW_QUANT_DELTA_3_VP7_VP8	VP8 quantisizer delta 3
21-11 SW_QUANT_2_VP7_VP8	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)
10-0 SW_QUANT_3_VP7_VP8	VP7: QP (11 bit) VP8: quantisizer value for LUT (7 bit)

14.1.5.4.34 bi-dir and P fwd initial ref pic list register (15 and P 0-3) / VP7,VP8 quantization values (SWREG47_VP7_VP8)

14.1.5.4.34.1 Offset

Register	Offset
SWREG47_VP7_VP8	BCh

14.1.5.4.34.2 Diagram



14.1.5.4.34.3 Fields

Field	Function
31-27	VP8 quantisizer delta 4

Table continues on the next page...

Field	Function
SW_QUANT_DELTA_4_VP7_VP8	
26-22 —	Reserved.
21-11 SW_QUANT_4_VP7_VP8	VP7 QP (11 bit)
10-0 SW_QUANT_5_VP7_VP8	VP7 QP (11 bit)

14.2 VPU G2 (VPU_G2)

14.2.1 Overview

This block details the G2 hardware-based decoder and API. The decoder is able to decode HEVC standard Main/Main 10 profile compatible video streams, and Google's VP9 Profile 0 compatible video streams. The G2 decoder conforms to the HEVC Main/ Main 10 profiles and can decode streams up to level 5.1.

14.2.2 Video Frame Storage Format

The HW decoder may support two storage formats for frames, raster-scan, tiled without being compressed and compressed output. The application using the decoder may specify upon initialization which format is preferred, depending on HW support. The tiled format generally speaking offers better bus utilization and performance, however the decoder output must be converted back to raster-scan format prior to displaying it, except that GPU supports tiled format displaying. This postprocessing is beyond the scope of G2 decoder.

14.2.2.1 Raster-scan format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 format, i.e. luminance data forms one plane in memory, and chrominance data forms the other. The output picture has to be stored linearly and contiguously in the memory. The interleaved

chrominance block has to be stored right after the luminance block in external memory as shown in Figure . The number of bytes in one luminance row must be divisible by 16, which means there will be padded 0 in the right edge when necessary.

The interleaved chrominance is stored raster-scanned in memory in Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

The decoder output picture for interlaced sequence is always stored as an interlaced frame where even lines belongs to top field and odd lines belongs to bottom field of the frame.

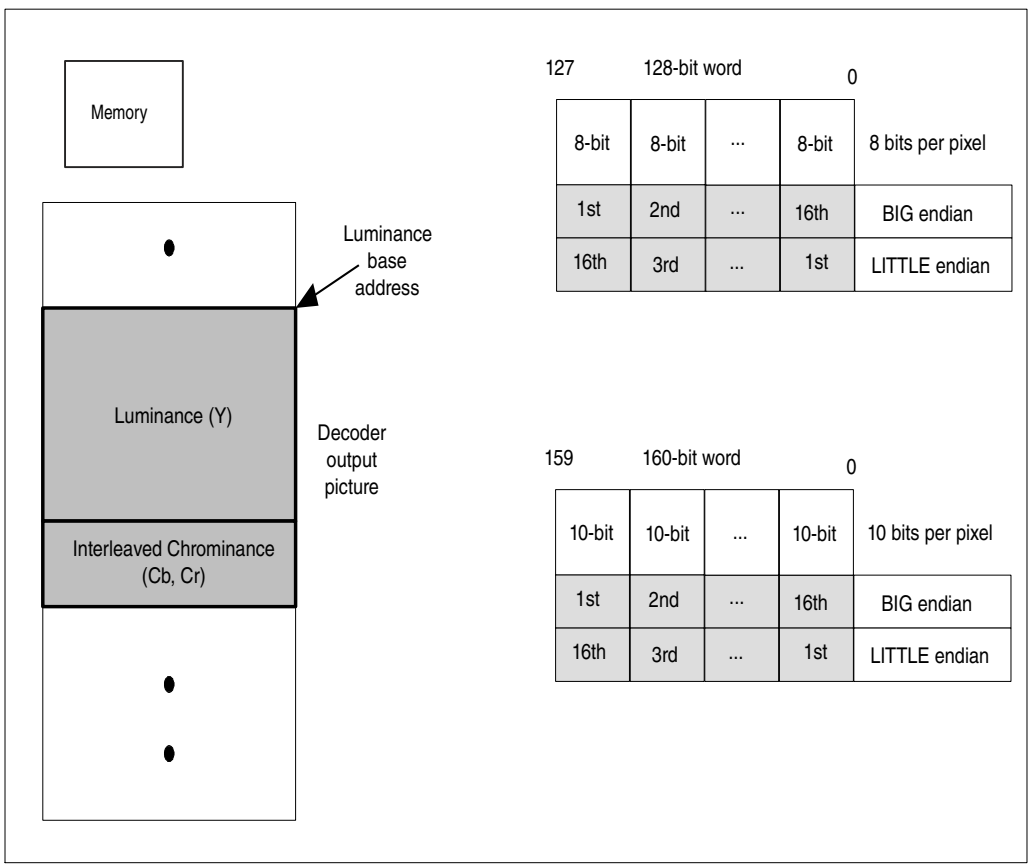


Figure 14-7. External Memory Usage

As an example the luminance pixels of a QCIF sized (176 x 144) frame with each pixel in 8 bits are numbered as presented in Table 1. Table 2 shows the storage of the luminance pixel data of the QCIF sized video frame in raster-scan order (in little endian mode).

Table 14-1. Pixel numbers of the luminance data of the QCIF video frame

0	1	2	3	4	5	6	7	…	172	173	174	175
176	177	178	179	180	181	182	183	…	348	349	350	351
::												::
25200	25201	25202	25203	25204	25205	25206	25207	…	25340	25341	25342	25343

The storage of the luminance pixel data of a QCIF video frame in raster-scan order. All pixels of a pixel row are stored in consecutive memory locations.

Table 14-2. Pixel Memory Locations

Address offset (decimal)	128-bit words(Pixel data of the luminance component. Each byte contains data for one pixel. Number in table are pixel numbers)			
0	15	...	1	0
16	31	...	17	16
32	47	...	33	32
48	63	...	49	48
64	79	...	65	64
:	:			
25296	25311	...	25297	25296
25312	25327	...	25313	25312
25328	25343	...	25329	25328

14.2.2.2 Tiled format

The output picture of the decoder is in semi-planar YCbCr 4:2:0 4x4 tiled format, i.e. luminance data forms one plane in memory, and chrominance data forms another. The distinction from raster scan format is that the output picture is grouped into tiles, which are then stored linearly and contiguously in the memory. The chrominance tiles have to be stored right after the luminance tiles in external memory as shown in Figure . The number of luminance pixels in one row must be divisible by 16.

The chrominance is stored into tiles in interleaved Cb0Cr0Cb1Cr1Cb2Cr2Cb3… format.

Figure contains example external memory usage when using 4x4 tiles, which is used in G2 decoder.

Below is an example code of converting tiled YCbCr 4:2:0 output picture back to raster scan format. The code is in generalized form, i.e. can function on arbitrary tile sizes as long as the picture dimensions are multiple of tile sizes. In the example, chrominance data is interleaved into 4x4 tile so that one 4x4 tile contains one 2x4 block of Cb and Cr each. Output is YCbCr 4:2:0 semiplanar.

```
void ConvertTiledToRaster( DecPicture decPicture, u32 tileWidth,
            u32 tileHeight, u8 *pOutput )
            {
            u32 i, j;
            u32 k, l;
            u32 s, t;
            u8 *pIn;
            pIn = decPicture.pOutputPicture;
            /* loop all tile rows for luminance */
            for( i = 0 ; i < decPicture.frameHeight ; i += tileHeight )
```

```
{
t = 0;
s = 0;
/* loop all tiles in one row */
for( j = 0 ; j < decPicture.frameWidth ; j += tileWidth )
{
/* copy one tile */
for( k = 0 ; k < tileHeight ; ++k )
{
for( l = 0 ; l < tileWidth ; ++l )
{
pOutput[ k*decPicture.frameWidth + l + s ] = pIn[t++];
}
}
/* move to next horizontal tile */
s += tileWidth;
}
/* move to next tile row */
pOutput += decPicture.frameWidth * tileHeight;
pIn += decPicture.frameWidth * tileHeight;
}
/* setup pointers to chrominance data */
pIn = decPicture.pOutputPicture +
decPicture.frameWidth * decPicture.frameHeight;
/* loop all tile rows for chrominance */
for( i = 0 ; i < decPicture.frameHeight/2 ; i += tileHeight )
{
t = 0;
s = 0;
/* loop all tiles in one row */
for( j = 0 ; j < decPicture.frameWidth ; j += tileWidth )
{
/* copy one tile */
for( k = 0 ; k < tileHeight ; ++k )
{
for( l = 0 ; l < tileWidth ; ++l )
{
pOutput[ k*decPicture.frameWidth + l + s ] = pIn[t++];
}
}
/* move to next horizontal tile */
s += tileWidth;
}
/* move to next tile row */
pOutput += decPicture.frameWidth * tileHeight;
pIn += decPicture.frameWidth * tileHeight;
}
}
```

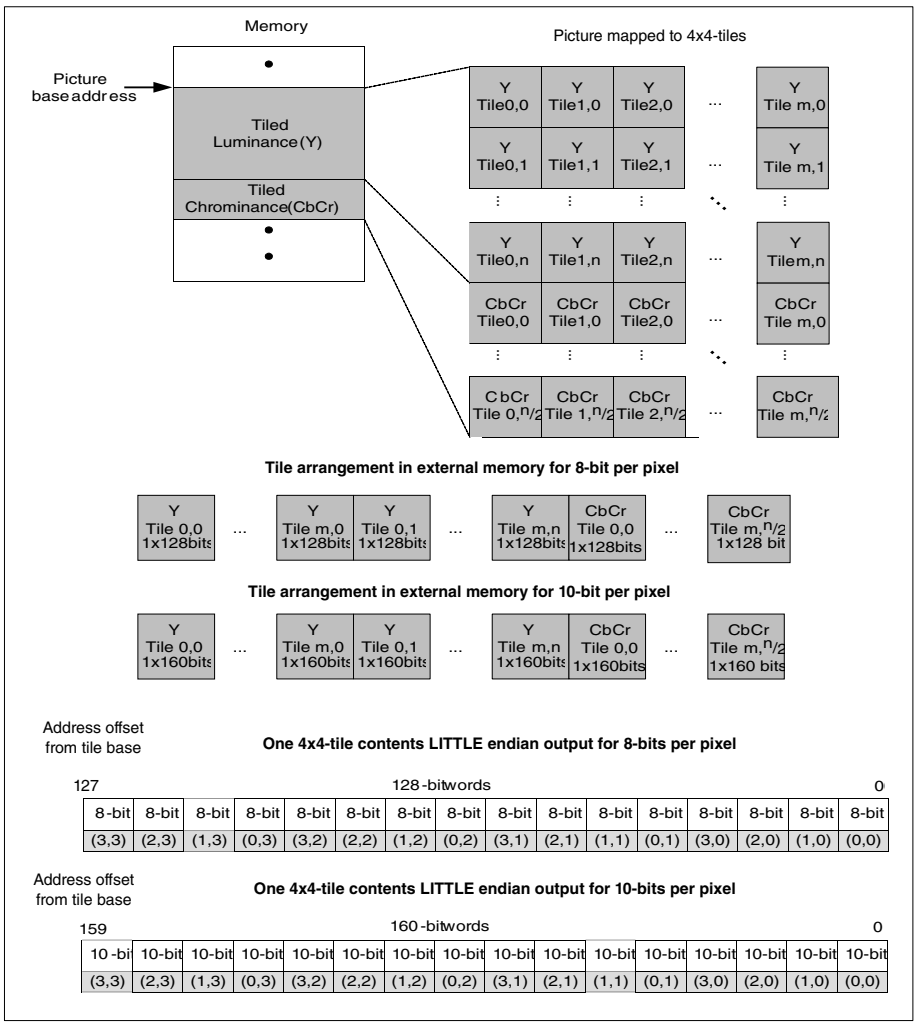


Figure 14-8. External Memory Usage in 4x4 Tiled Reference Picture Mode

14.2.2.3 Compressed format

G2 decoder supports compressed tiled output, when reference buffer compression (RFC) is enabled when synthesis, and user allows compressed reference frame output directly.

Reference frame compression is a feature added in G2 decoder to compress frame buffer so that the bandwidth of storing/loading reference frame can be reduced, especially when the resolution of decoded stream is of high definition.

When compression is enabled, the picture is divided into CBS rows, which is further divided into continuous CBS groups. Each CBS group is composed of 16 CBS. The luminance CBS is composed of 1 8x8 coded block (CB), which one chrominance CBS is composed of two 8x4 coded blocks, with cb CB first then cr CB following. The compression is performed to each CB in the CBS. The first CB's compressed data is

saved from the same offset of that CBS in raster scan buffer. And the compressed output of following CBs in the CBS row is saved continuously. That means there may be gaps between the compressed data of each CBS row.

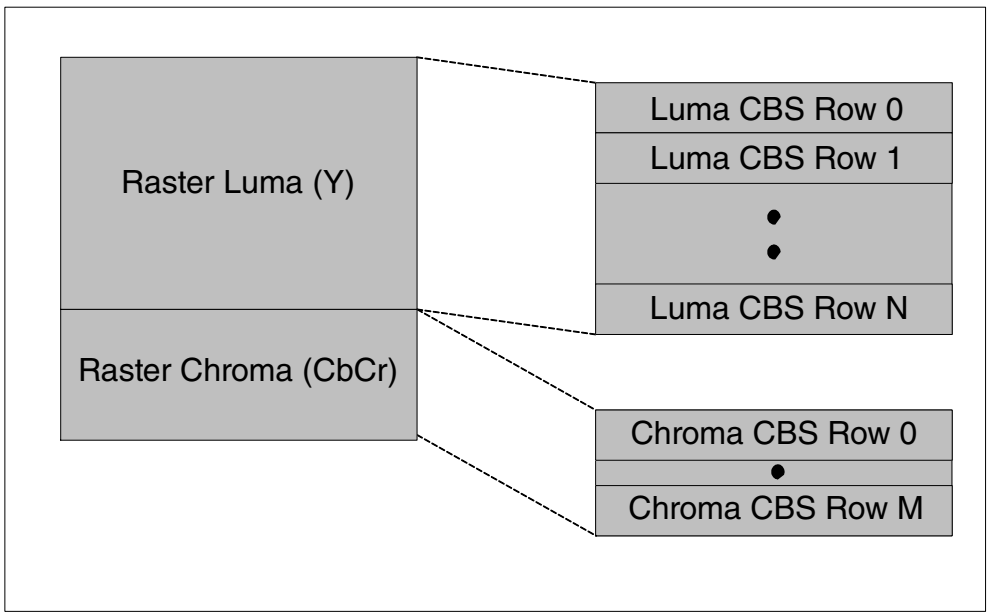


Figure 14-9. Raster scan picture mapped to CBS rows

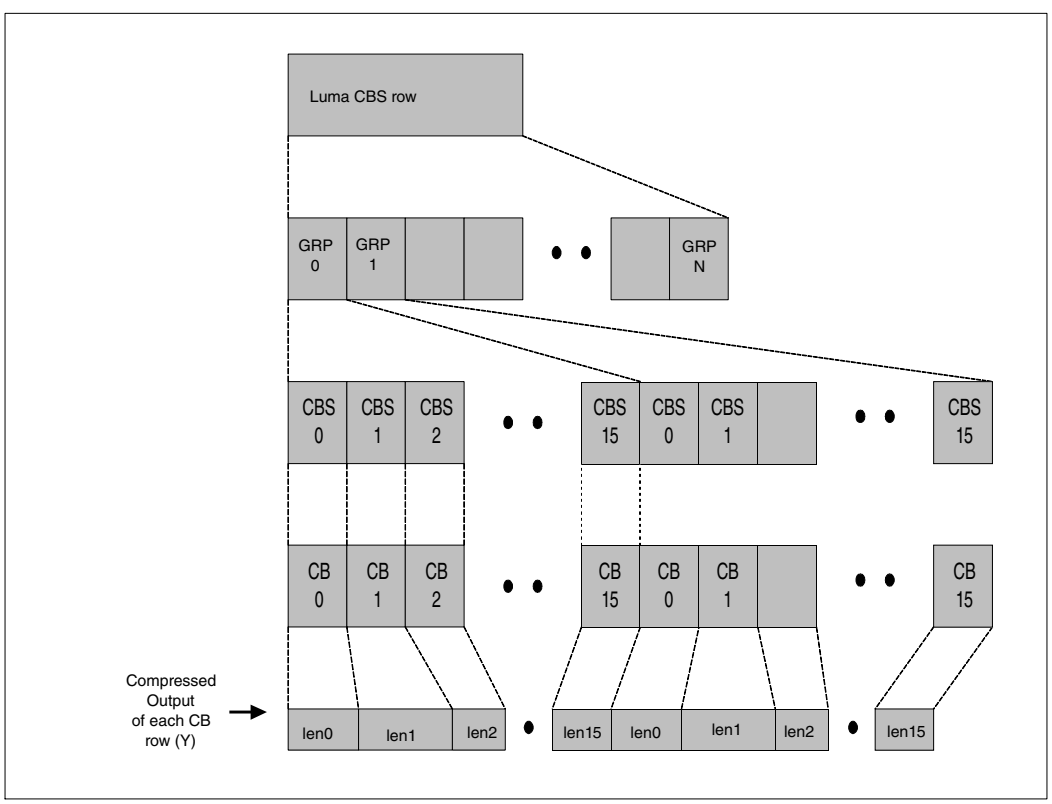


Figure 14-10. Compression performed to each Luma CBS row

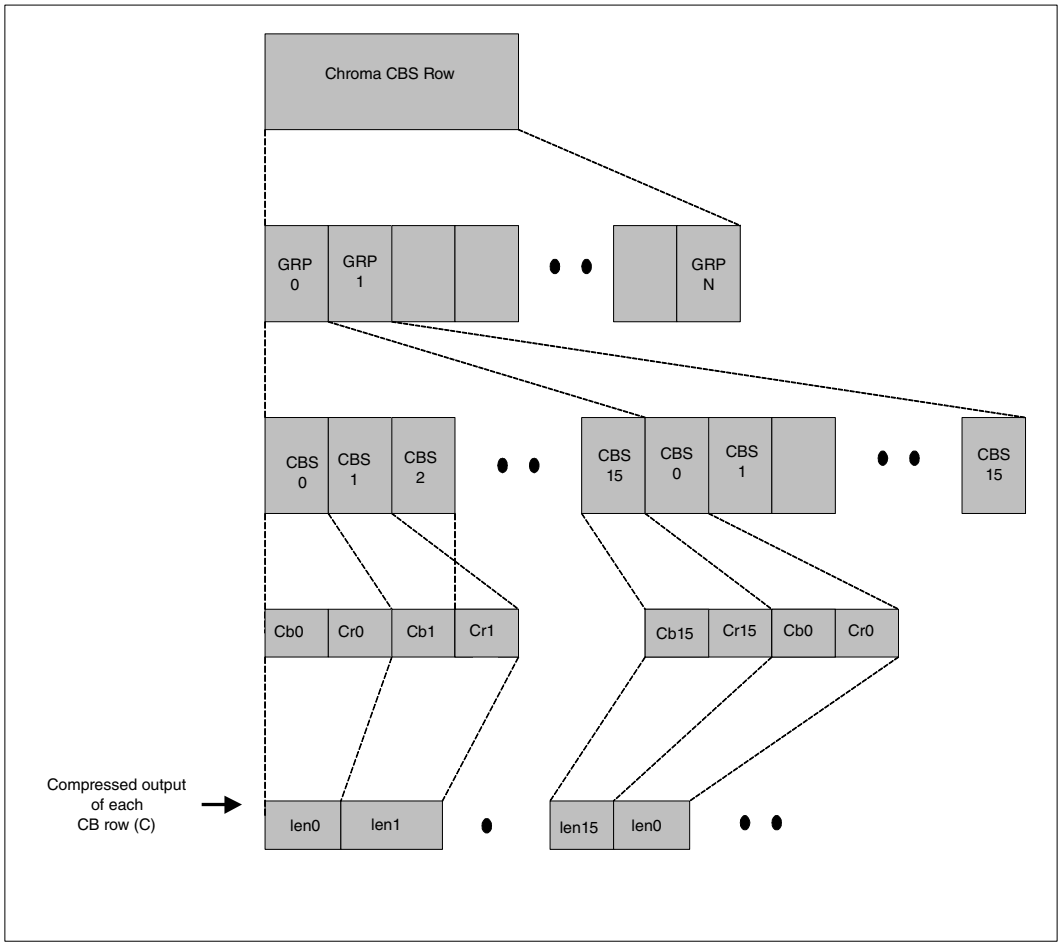


Figure 14-11. Compression performed to each Chroma CBS row

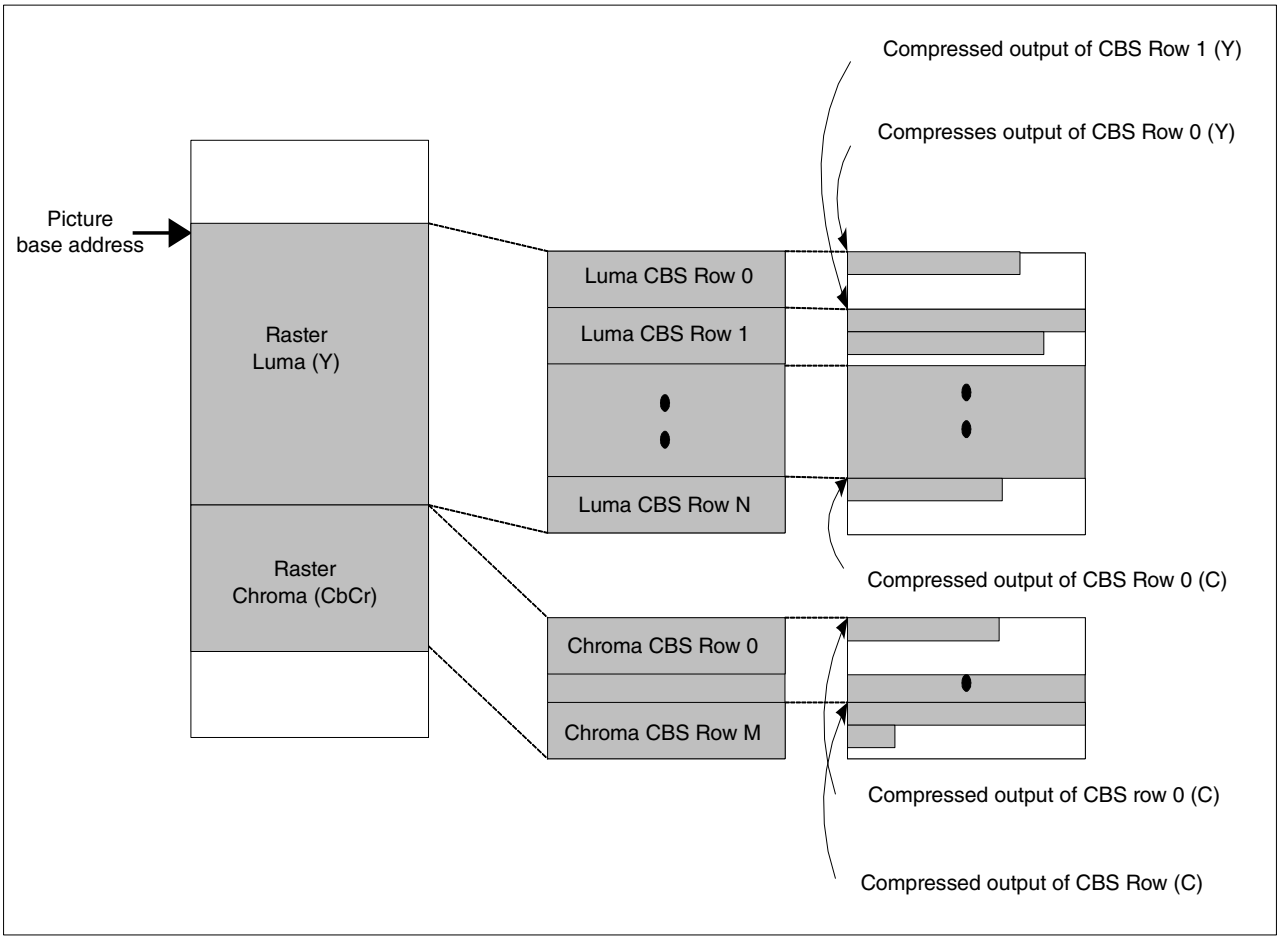


Figure 14-12. Map of external memory in RFC output buffer

Corresponding to each CBS group the starting offset of first compressed CBS output is saved in two bytes in CBS table, followed by the length of each CBS compress output, which is saved in 7 bits. So the table length of each CBS group information is 2+7*16/8=16 bytes.

This 16 bytes for each CBS group in compression table is mapped as shown in the following table. Here, X[n] means the bit-n of the length of CBS X in CBS group.

Table 14-3. Storage of the compression cache table for each CBS group

Byte	Bit7	Bit6	Bit5	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
0	Offset[7:0]							
1	Offset[15:8]							
2	1[0]	0[6]	0[5]	0[4]	0[3]	0[2]	0[1]	0[0]
3	2[1]	2[0]	1[6]	1[5]	1[4]	1[3]	1[2]	1[1]
4	3[2]	3[1]	3[0]	2[6]	2[5]	2[4]	2[3]	2[2]
5	4[3]	4[2]	4[1]	4[0]	3[6]	3[5]	3[4]	3[3]
6	5[4]	5[3]	5[2]	5[1]	5[0]	4[6]	4[5]	4[4]

Table continues on the next page...

Table 14-3. Storage of the compression cache table for each CBS group (continued)

Byte	Bit7	Bit6	Bit5	Bit4	Bit 3	Bit 2	Bit 1	Bit 0
7	6[5]	6[4]	6[3]	6[2]	6[1]	6[0]	5[6]	5[5]
8	7[6]	7[5]	7[4]	7[3]	7[2]	7[1]	7[0]	6[6]
9	9[0]	8[6]	8[5]	8[4]	8[3]	8[2]	8[1]	8[0]
10	10[1]	10[0]	9[6]	9[5]	9[4]	9[3]	9[2]	9[1]
11	11[2]	11[1]	11[0]	10[6]	10[5]	10[4]	10[3]	10[2]
12	12[3]	12[2]	12[1]	12[0]	11[6]	11[5]	11[4]	11[3]
13	13[4]	13[3]	13[2]	13[1]	13[0]	12[6]	12[5]	12[4]
14	14[5]	14[4]	14[3]	14[2]	14[1]	14[0]	13[6]	13[5]
15	15[6]	15[5]	15[4]	15[3]	15[2]	15[1]	15[0]	14[6]

14.2.3 Interface Functions

This chapter describes the API declared in decapi.h, the definition of API in “software \source\common\decapi.c” of the release package.

Common numeric data types are declared in basetype.h as follows:

- **u8** – unsigned 8 bits integer value
- **i8** – signed 8 bits integer value
- **u16** – unsigned 16 bits integer value
- **i16** – signed 16 bits integer value
- **u32** – unsigned 32 bits value
- **i32** – signed 32 bits value

14.2.3.1 DecGetBuild

Syntax

```
struct DecSwHwBuild DecGetBuild(void)
```

Purpose

Returns the hardware and software build information of the decoder. Does not require the creation of a decoder instance.

Parameters

None.

Return value

DecSwHwBuild -- A structure containing the build information of the decoder.

```
struct DecSwHwBuild {
        u32 sw_build;                                   /* Software build ID */
        u32 hw_build;                                   /* Hardware build ID */
        struct DecHwConfig hw_config;          /* Hardware configuration */
};
```

Function	Description
sw_build	The internal release version of the Hantro HW HEVC software.
hw_build	The internal release version of the Hantro HW hardware.

DecHwConfig -- A structure containing the hardware-supported configuration.

```
struct DecHwConfig {
     u32 mpeg4_support;
     u32 custom_mpeg4_support;
     u32 h264_support;
     u32 vc1_support;
     u32 mpeg2_support;
     u32 jpeg_support;
     u32 jpeg_prog_support;
     u32 max_dec_pic_width;
     u32 max_dec_pic_height;
     u32 pp_support;
     u32 pp_config;
     u32 max_pp_out_pic_width;
     u32 sorenson_spark_support;
     u32 ref_buf_support;
     u32 tiled_mode_support;
     u32 vp6_support;
     u32 vp7_support;
     u32 vp8_support;
     u32 vp9_support;
     u32 avs_support;
     u32 jpeg_esupport;
     u32 rv_support;
     u32 mvc_support;
     u32 webp_support;
     u32 ec_support;
     u32 stride_support;
     u32 field_dpb_support;
     u32 hevc_support;
     u32 double_buffer_support;
     u32 hevc_main10_support;
     u32 vp9_10bit_support;
     u32 ds_support;
     u32 rfc_support;
     u32 ring_buffer_support;
};
```

Function	Description
mpeg4_support	Hardware support for MPEG-4 decoding. Possible values: • MPEG4_NOT_SUPPORTED – decoding not supported • MPEG4_SIMPLE_PROFILE – simple profile decoder • MPEG4_ADVANCED_SIMPLE_PROFILE – advanced simple profile decoder (includes • simple profile)
custom_mpeg4_support	Hardware support for DivX®decoding. Possible values:

Table continues on the next page...

Function	Description
	• MPEG4_CUSTOM_NOT_SUPPORTED – decoding not supported • MPEG4_CUSTOM - DivX Home Theater Profile QualifiedTM decoder available
h264_support	Hardware support for H.264 decoding. Possible values: • H264_NOT_SUPPORTED – decoding not supported • H264_BASELINE_PROFILE – baseline profile decoder • H264_MAIN_PROFILE – main profile decoder (includes baseline profile) • H264_HIGH_PROFILE – high profile decoder (includes baseline and main profile)
vc1_support	Hardware support for VC-1 decoding. Possible values: • VC1_NOT_SUPPORTED – decoding not supported. • VC1_SIMPLE_PROFILE – simple profile decoder • VC1_MAIN_PROFILE – main profile decoder (includes simple profile) • VC1_ADVANCED_PROFILE – advanced profile decoder (includes main profile)
mpeg2_support	Hardware support for MPEG-2 (includes MPEG-1) decoding. Possible values: • MPEG2_NOT_SUPPORTED – decoding not supported • MPEG2_MAIN_PROFILE – main profile decoder
jpeg_support	Hardware support for JPEG decoding. Possible values: • JPEG_NOT_SUPPORTED – decoding not supported • JPEG_BASELINE – baseline decoder
jpeg_prog_support	Hardware support for progressive JPEG decoding. Possible values: • JPEG_NOT_SUPPORTED – decoding not supported • JPEG_PROGRESSIVE – progressive decoder available
max_dec_pic_width	Maximum decoded picture's width for video decoders. The value is given in pixel units.
max_dec_pic_height	Maximum decoded picture's height for video decoders. The value is given in pixel units.
pp_support	Hardware support for video post-processing. Possible values: • PP_NOT_SUPPORTED – post-processing not supported. • PP_SUPPORTED – post-processing supported in hardware.
pp_config	Post-processing functions available. This is a bitwise list of the following values: • PP_DITHERING, PP_SCALING, PP_DEINTERLACING, and • PP_ALPHA_BLENDING. The availability of all these post-processor functions is • dependent on the hardware build.
max_pp_out_pic_width	Maximum output picture's width for the post-processor. The value is given in pixel units.

Table continues on the next page...

Function	Description
sorenson_spark_support	Hardware support for Sorenson Spark decoding. Possible values: • SORENSON_SPARK_NOT_SUPPORTED – decoding not supported • SORENSON_SPARK_SUPPORTED – decoder available
ref_buf_support	Hardware includes reference buffer handling speedup logic. Possible values: • REF_BUF_NOT_SUPPORTED – logic not available • REF_BUF_SUPPORTED – logic available for progressive pictures (flag) • REF_BUF_INTERLACED – logic available for interlaced pictures (flag) • REF_BUF_DOUBLE – logic available for buffering two reference pictures (flag) • Multiple values are possible using bitwise OR operation of valid flags.
tiled_mode_support	Hardware includes support for handling reference pictures in tiled mode to optimize bus utilization. Possible values: • TILED_NOT_SUPPORTED – tiled reference picture support not available • TILED_8x4_SUPPORTED – tiled reference picture support for 8x4 tiles available • TILED_8x4_ILACED_SUPPORTED – tiled reference picture support for 8x4 tiles • available also for interlaced pictures.
vp6_support	Hardware support for VP6 decoding. Possible values: • VP6_NOT_SUPPORTED – decoding not supported • VP6_SUPPORTED – decoder available
vp7_support	Hardware support for VP7 decoding. Possible values: • VP7_NOT_SUPPORTED – decoding not supported • VP7_SUPPORTED – decoder available
vp8_support	Hardware support for VP8 decoding. Possible values: • VP8_NOT_SUPPORTED – decoding not supported • VP8_SUPPORTED – decoder available
vp9_support	Hardware support for VP9 decoding. Possible values: • VP9_NOT_SUPPORTED – decoding not supported • VP9_SUPPORTED – decoder available
avs_support	Hardware support for AVS decoding. Possible values: • AVS_NOT_SUPPORTED – decoding not supported • AVS_SUPPORTED – decoder available
jpeg_esupport	Hardware support for baseline JPEG decoding up to 67 Mpixel images, and 4:1:1 and 4:4:4 sampling. Possible values: • JPEG_EXT_NOT_SUPPORTED – decoding not supported • JPEG_EXT_SUPPORTED – decoder available
rv_support	Hardware support for RV decoding. Possible values:

Table continues on the next page...

Function	Description
	• RV_NOT_SUPPORTED – decoding not supported • RV_SUPPORTED – decoder available
mvc_support	Hardware support for MVC decoding. Possible values: • MVC_NOT_SUPPORTED – decoding not supported • MVC_SUPPORTED – decoder available
webp_support	Hardware support for WebP decoding. Possible values: • WEBP_NOT_SUPPORTED – decoding not supported • WEBP_SUPPORTED – decoder available
ec_support	Hardware support for error concealment. Possible values: • EC_NOT_SUPPORTED – error concealment not supported • EC_SUPPORTED – error concealment available
stride_support	Hardware support for separate Y and C strides. Possible values: • STRIDE_NOT_SUPPORTED – strides not supported • STRIDE_SUPPORTED – strides available
field_dpb_support	Hardware support for field-mode DPB. Possible values: • FIELD_DPB_NOT_SUPPORTED – field-mode DPB not supported • FIELD_DPB_SUPPORTED –field-mode DPB available
hevc_support	Hardware support for HEVC decoding. Possible values: • HEVC_NOT_SUPPORTED – decoding not supported • HEVC_SUPPORTED – decoder available
double_buffer_support	Hardware support for decoder internal reference double buffering. Possible values: • DOUBLE_BUFFER_NOT_SUPPORTED – internal reference double buffering not • supported • DOUBLE_BUFFER_SUPPORTED – internal reference double buffering available
hevc_main10_support	Hardware support for HEVC main10 profile decoding. Possible values: • 0 – decoding not supported • 1 – decoder available
vp9_10bit_support	Hardware support for VP9 10 bits decoding. Possible values: • 0 – decoding not supported • 1 – decoder available
ds_support	Hardware support for decoder down scale outputting. Possible values: • 0 – decoding not supported • 1 – decoder available
rfc_support	Hardware support for reference frame compression. Possible values: • 0 – decoding not supported • 1 – decoder available
ring_buffer_support	Hardware support for ring buffer. Possible values:

Function	Description
	• 0 – decoding not supported • 1 – decoder available

14.2.3.2 DecInit

Syntax

```
enum DecRet DecInit(enum DecCodec codec,
                                  DecInst* decoder,
                                  struct DecConfig config,
                                  struct DecClientHandle callbacks)
```

Purpose

DecInit initializes a new decoder instance based on the given codec type. If parameter noOutputReordering is set to a non-zero value the decoder will not provide output pictures in display order but in decoding order (unless decoding order and display order is the same). Outputting pictures in decode order will lower the memory consumption of the decoder in case the number of reference frames used by the stream is less than the maximum number allowed by the profile and level of the stream.

Parameters

```
enum DecCodec codec
                  Codec type to be initialized.
                  enum DecCodec {
                  DEC_VP9,
                  DEC_HEVC
                  };
             DecInst * decoder
                  Pointer to the location where DecInit returns a decoder instance. This
instance is later
                  passed to other decoder API functions.
             struct DecConfig config
                  Decoder initialization parameters.
                    struct DecConfig {
                    u32 disable_picture_reordering;
                    enum DecPictureFormat output_format;
                    struct DWL dwl;
                    const void* dwl_inst;
                    u32 max_num_pics_to_decode;
                    enum DecErrorConcealment concealment_mode;
                    struct DecDownscaleCfg dscale_cfg;
                    u32 use_video_compressor;
                    u32 use_ringbuffer;
                    u32 use_8bits_output;
             }
             u32 disable_picture_reordering
                 Flag to disable the reordering of output frames. When set to a non-zero
value the
                 decoder will not provide output pictures in display order but in decoding
order (unless
                 decoding order and display order is the same).
             enum DecPictureFormat output_format
                  Format of the output picture.
```

```
                    enum DecPictureFormat {
                     DEC_OUT_FRM_TILED_4X4 = 0,
                     DEC_OUT_FRM_RASTER_SCAN = 1,
                     DEC_OUT_FRM_PLANAR_420 = 2
                     };
                    DEC_OUT_FRM_TILED_4X4 - Tiled 4x4 format
                    DEC_OUT_FRM_RASTER_SCAN - raster scan format (a.k.a semi-planar)
                    DEC_OUT_FRM_PLANAR_420 - planar format
               struct DWL dwl
                    Decoder wrapper layer functionality.
               const void* dwl_inst
                    Pointer to a DWL instance, which should be instantiated by calling DWLInit
in
                    advance.
               u32 max_num_pics_to_decode
                    Limits the maximum pictures number to be decoded, 0 for unlimited.
               enum DecErrorConcealment concealment_mode
                    Flag to determine which error concealment method to use. When set to
                    DEC_INTRA_FREEZE the decoder will conceal every frame after an error has
been
                    detected in the bitstream, until the next IDR (key frame) frame is
decoded. When set to
                    DEC_PICTURE_FREEZE, the decoder will conceal only the frames having errors
in the
                    bitstream.

                    enum DecErrorConcealment {
                    DEC_PICTURE_FREEZE = 0,
                    DEC_INTRA_FREEZE = 1
                    };
               struct DecDownscaleCfg dscale_cfg
                    Down scale parameters.
                    struct DecDownscaleCfg {
                      u32 down_scale_x;
                      u32 down_scale_y;
                    };
                    u32 down_scale_x
                        Down scale ratio in horizontal direction. Possible value: 2, 4, 8.
Available
                        value:2, 4, 8, which means down scaled to 1/2, 1/4 or 1/8
respectively. 1
                        means no scaling.
                    u32 down_scale_y
                        Down scale ratio in vertical direction. Possible value: 2, 4, 8.
Available value:2,
                        4, 8, which means down scaled to 1/2, 1/4 or 1/8 respectively. 1 means
no
                        scaling.
```

NOTE

If the down scale ratio in one direction is set to 1, no matter what the value is in the other direction, no down scale will be performed in both directions.

```
               u32 use_video_compressor
                   Flag to enable reference buffer compression. Set 0 to bypass reference
buffer
                   compression, or when reference buffer compression is not supported. When
set to 1,
                   the decoded reference buffer will be saved in compressed format.
               u32 use_ringbuffer
                   Flag to indicate decoder use ring buffer or not. Set 1 to enable ring
buffer. When set to
                   0, ring buffer will be disabled.
                   Stream ring buffer mode is an alternative storage for stream. In this mode,
frame
```

stream data usually doesn’t start from buffer head but follow a previous frame stream
data. When stream data goes to stream buffer end, it must turnaround to buffer head to
save left stream data. It requires a whole frame stream data saved in stream buffer and
stream data start address is byte-aligned.
Below figure shows how stream data is stored in stream buffer if ring buffer mode.

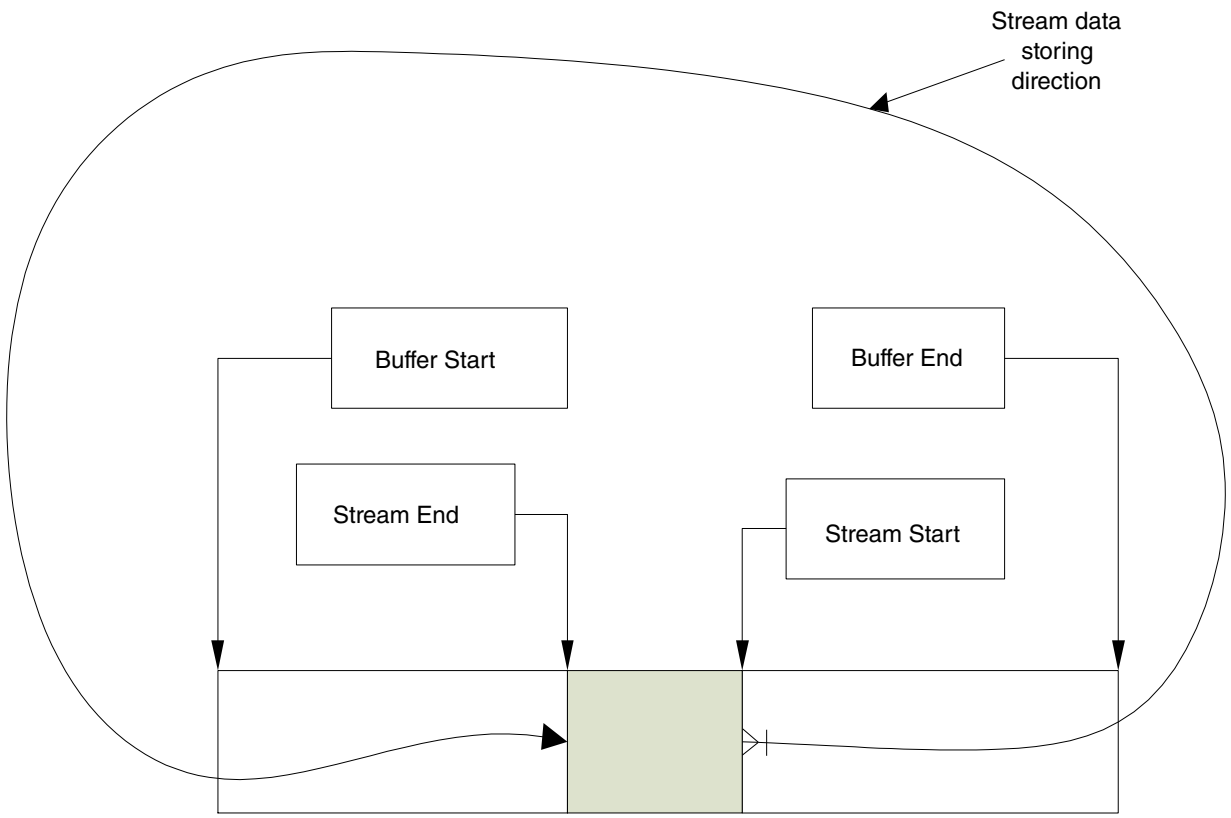


Figure 14-13. Data storage in Stream buffer in Ring Buffer Mode

u32 use_8bits_output
Flag to force 8 bit per pixel output when it's a stream with higher bit depth, e.g., HEVC
Main10 profile stream. The conversion is based on following rule:

Table 14-4. Rule for conversion

Bit depth of original pixel	Original pixel	8bit output
9	XXXXXXXX0	XXXXXXXX
	XXXXXXXX1	XXXXXXXX+1
10	XXXXXXXX00	XXXXXXXX
	XXXXXXXX01	XXXXXXXX
	XXXXXXXX10	XXXXXXXX+1
	XXXXXXXX11	XXXXXXXX+1

struct DecClientHandle callbacks
Callback interfaces that client should provide to facilitate the decoder

```
to inform the client that
                    there is any events or state changing in decoder.
                    struct DecClientHandle {
                           ClientInst client;
                           ClientInitialized* Initialized;
                           ClientHeadersDecoded* HeadersDecoded;
                           ClientBufferDecoded* BufferDecoded;
                           ClientPictureReady* PictureReady;
                           ClientEndOfStream* EndOfStream;
                           ClientReleased* Released;
                           ClientNotifyError* NotifyError;
                    };
                    typedef const void* ClientInst;
                           Opaque pointer to the client.
                    typedef void ClientInitialized(ClientInst inst);
                           Function to notify the client that decoder has successfully
initialized.
                    typedef void ClientHeadersDecoded(ClientInst inst,
                                struct DecSequenceInfo sequence_info);
                           Function to notify about successful decoding of the stream
parameters. Decoder
                           expects client to provide needed buffers through
DecSetPictureBuffers function to
                           continue decoding the actual stream.
                    typedef void ClientBufferDecoded(ClientInst inst,
                                                     struct DecInput* input);
                           Function to notify client that a buffer has been consumed by the
decoder and it can be
                           handled freely by the client.
                    typedef void ClientPictureReady(ClientInst inst,
                                                    struct DecPicture picture);
                           Function to notify about picture that is ready to be outputted.
Client is expected to notify
                           the decoder when it has finished processing the picture, so
decoder can reuse the
                           picture buffer for another picture.
                    typedef void ClientEndOfStream(ClientInst inst);
                            Function to notify the client that all the pending pictures
have been outputted and
                            decoder can be safely shut down.
                    typedef void ClientReleased(ClientInst inst);
                           Function to notify the client that decoder has shut down.
                    typedef void ClientNotifyError(ClientInst inst,
                                                   u32 pic_id,
                                                   enum DecRet rv);
                           Function to notify client about error in the decoding process.
```

Return

```
DEC_OK
                 API function returned successfully.
             DEC_PARAM_ERROR
                 Bad input parameters.
             DEC_MEMFAIL
                 The decoder was not able to allocate memory.
             DEC_FORMAT_NOT_SUPPORTED
                 Format is not supported in hardware. See DecGetBuild.
```

14.2.3.3 DecDecode

Syntax

```
enum DecRet DecDecode(DecInst decInst,
        DecInput *pInput)
```

Purpose

This function will push a COMMAND_DECODE command to the decoding thread, along with the input stream parameters, which will inform the decoding thread to decode one or more NAL units from the current stream for HEVC or element stream for VP9. The input buffer should contain one of the following:

1. Exactly one NAL unit and nothing else.
2. One or more NAL units in byte stream format, as defined in Annex B of the standard [1].
3. Or for VP9 at least one whole frame bit stream data.

The decoder automatically detects the format of the stream data and decodes NAL units until the whole buffer is processed or decoding of a picture is finished for HEVC. For VP9 if there is not enough bit stream data in input buffer, HW decoder will return TIMEOUT error.

Parameters

DecInst*decInst* - A decoder instance created earlier with an DecInit call.

DecInput **pInput* - Pointer to the decoder input structure.

```
struct DecInput {
               struct DWLLinearMem buffer; /* Pointer to the input buffer. */
               u32 data_len;
        };
```

struct DWLLinearMem buffer -- Linear buffer descriptor.

```
struct DWLLinearMem {
                  u32 *virtual_address;
                  u32 bus_address;
                  u32 size;
                  u32 logical_size;
          };
```

Function	Description
*virtual_address	Pointer to the virtual address of stream linear buffer.
bus_address	Bus address of the linear buffer. NOTE: The start of the stream buffer must be in a 128-bit aligned position or accessible for reading from the previous 128-bit aligned position.
size	Physical size of the linear buffer, which is rounded to page multiple.
logical_size	Physical size of the linear buffer, which is rounded to page multiple.

Table continues on the next page...

Function	Description
datalen	Number of bytes contained in the buffer stream buffer. The length can be represented on maximum 32 bits. Valid range: [0, (2^32-1)]

Return value

DEC_OK - The COMMAND_DECODE is successfully pushed to decoder thread's input queue.

DEC_PARAM_ERROR - Error in calling parameters.

DEC_NOT_INITIALIZED - Decoder instance is not initialized. Stream decoding is not started. DecInit must be called before decoding can be started.

14.2.3.4 DecSetPictureBuffers

Syntax

```
enum DecRet DecSetPictureBuffers(DecInst dec_inst,
               const struct DWLLinearMem* buffers,
               u32 num_of_buffers)
```

Purpose

DecSetPictureBuffers assigns picture buffers for the decoder. When decoder has finished decoding the stream headers and knows which types of buffers and how many of them it will need, it will inform that through the HeadersDecoded callback. Buffers must not be written into until client has successfully called DecRelease or decoder has requested new set of buffers through HeadersDecoded callback.

Parameters

DecInst*decInst* - A decoder instance created earlier with an DecInit call.

const struct DWLLinearMem* buffers - Linear buffer to be assigned to the decoder.

u32 num_of_buffers - Number of buffers to be assigned to the decoder.

Return value

DEC_OK - Successfully set picture buffers.

DEC_PARAM_ERROR - Error in calling parameters.

14.2.3.5 DecPictureConsumed

Syntax

```
enum DecRet DecPictureConsumed(DecInst decInst,
        struct DecPicture picture)
```

Purpose

DecPictureConsumed informs the decoder that the client has finished processing a specific picture, which was previously sent to client through the PictureReady callback.

Parameters

DecInst*decInst* - A decoder instance.

DecPicture *pOutput - Pointer to the location used to return the picture parameters. The picture parameters are valid only when the return value indicates that an output picture is available.

```
struct DecPicture {
               struct DecSequenceInfo sequence_info;
               struct DWLLinearMem luma;
               struct DWLLinearMem chroma;
               struct DecPictureInfo picture_info;
               struct DWLLinearMem dscale_luma;
               struct DWLLinearMem dscale_chroma;
               u32 dscale_width;
               u32 dscale_height;
               u32 dscale_stride;
          };
          struct DecSequenceInfo sequence_info
                struct DecSequenceInfo {
                   u32 pic_width;
                   u32 pic_height;
                   u32 sar_width;
                   u32 sar_height;
                   struct DecCropParams crop_params;
                   enum DecVideoRange video_range;
                   u32 matrix_coefficients;
                   u32 is_mono_chrome;
                   u32 is_interlaced;
                   u32 num_of_ref_frames;
                   u32 bit_depth_luma;
                   u32 bit_depth_chroma;
                   u32 pic_stride;
               };
```

Function	Description
picture_width	Decoded picture width in pixels.
picture_height	Decoded picture height in pixels.
sar_width	Horizontal size of the sample aspect ratio.
sar_height	Vertical size of the sample aspect ratio.
struct DecCropParams crop_param	Cropping parameters for the picture.

Table continues on the next page...

Function	Description

	struct DecCropParams {
	u32 crop_left_offset;
	u32 crop_out_width;
	u32 crop_top_offset;
	u32 crop_out_height;
	};

crop_left_offset	Left offset for cropping process, i.e. number of luminance samples removed from the left edge of the picture (always a multiple of 2).
crop_out_width	Width of the picture in luminance samples after cropping is performed.
crop_top_offset	Top offset for cropping process, i.e. number of luminance samples removed from the top edge of the picture (always a multiple of 2).
crop_out_height	Height of the picture in luminance samples after cropping is performed.
Enum DecVideoRange video_range	YUV sample video range. Possible value: • DEC_VIDEO_RANGE_NORMAL - sample range [16, 235] • DEC_VIDEO_RANGE_FULL - sample range [0, 255]
matrix_coefficients	Matrix coefficients RGB->YUV conversion.
is_mono_chrome	Flag to indicate whether sequence is of monochrome.
is_interlaced	Flag to indicate whether sequence is interlaced.
num_of_ref_frames	Maximum number of reference frames.
bit_depth_luma	Bit depth of stored pixels for luma plane.
bit_depth_chroma	Bit depth of stored pixels for chroma plane.
pic_stride	Stride of the picture as stored in memory in bytes.
struct DWLLinearMem luma	Linear buffer for luma pixels.
struct DWLLinearMem chroma	Linear buffer for chroma pixels.

struct DecPictureInfo picture_info	Picture specific information.
	struct DecPictureInfo {
	enum DecPicCodingType pic_coding_type;
	u32 is_corrupted;
	enum DecPictureFormat format;
	u32 cycles_per_mb;
	u32 pic_id;
	};

enum DecPicCodingType pic_coding_type	Picture coding type. Possible value: • DEC_PIC_TYPE_I - key frame • DEC_PIC_TYPE_P - P frame • DEC_PIC_TYPE_B - B frame
is_corrupted	Flag indicates whether picture is corrupted.
enum DecPictureFormat format	Color format of the picture. Refer to DecPictureFormat in 4.2.
cycles_per_mb	Average decoding time in cycles per mb.

Table continues on the next page...

Function	Description
pic_id	Identifier for the picture to be decoded.
struct DWLLinearMem dscale_luma	Linear buffer for luma pixels in down scaled output.
struct DWLLinearMem dscale_chroma	Linear buffer for chroma pixels in down scaled output.
dscale_width	Width of down scaled output in pixels.
dscale_height	Height of down scaled output in pixels.
dscale_stride	Stride of a pixel line in down scaled output in bytes.

Return value

DEC_OK - No pictures available for display.

DEC_PARAM_ERROR - Error in calling parameters.

DEC_NOT_INITIALIZED - Decoder instance is not initialized. DecInit must be called before decoding can be started.

14.2.3.6 DecEndOfStream

Syntax

```
enum DecRet DecEndOfStream (DecInst decInst)
```

Purpose

DecEndOfStream sends a COMMAND_END_OF_STREAM command to the decoder thread to inform the decoder that it should not be expecting any more input stream and finish decoding and outputting all the buffers that are currently pending in the component. Once decoder has finished outputting the pending pictures it will notify the client about it by calling the EndOfStream callback.

Parameters

DecInst*decInst* - A decoder instance.

Return value

DEC_OK - API function returned successfully

DEC_PARAM_ERROR - Error in calling parameters.

DEC_INITFAIL - Decoder instance is not initialized.

14.2.3.7 DecRelease

Syntax

```
void DecRelease(DecInst decInst)
```

Purpose

DecRelease sends a COMMAND_RELEASE command to decoder thread, which will finally close the decoder instance decInst and releases all internally allocated resources. This function must not be called twice for the same instance.

Parameters

DecInst*decInst* - A decoder instance to be released.

Return value

None.

14.2.4 Decoder Application Examples

Purpose

The purpose of the following examples is to show how to implement client's callback functions, as long as using G2 decoder API to implement a simple decoder with demuxer to decode both HEVC and VP9 streams. InitializedCb, HeadersDecodedCb, BufferDecodedCb, PictureReadyCb, EndOfStreamCb, ReleasedCb, and NotifyErrorCb are the system/application dependent callback functions, which are provided by client to the decoder wrapper.

The examples are simplified, e.g. they do not take errors into consideration.

NOTE

Program code in the examples should be considered as pseudo code and may not compile as such.

```
            void SetupDefaultParams(struct TestParams* params);
            int ParseParams(int argc, char* argv[], struct TestParams* params);
            void* CreateDemuxer(struct Client* client);
            void ReleaseDemuxer(struct Client* client);
            void PostProcessPicture(struct Client* client, struct DecPicture* picture);

            static void DispatchBufferForDecoding(struct Client* client, struct DecInput*
buffer)
            {
             enum DecRet rv;
             i32 size = buffer->buffer.size;
             memset(buffer->buffer.virtual_address, 0, size);
             i32 len = client->demuxer.ReadPacket(client->demuxer.inst, (u8*)buffer-
>buffer.virtual_address, &size);
             if (len <= 0) {/* EOS or error. */
```

```
              /* If reading was due to insufficient buffer size, try to realloc with
sufficient size. */
              if (size > buffer->buffer.size) {
               i32 i;
               for (i = 0; i < GetStreamBufferCount(client); i++) {
                if (client->buffers[i].buffer.virtual_address == buffer-
>buffer.virtual_address) {
                  DWLFreeLinear(client->dwl, &client->buffers[i].buffer);
                  if (DWLMallocLinear(client->dwl, size, &client->buffers[i].buffer))
                  {
                    DispatchEndOfStream(client);
                    return;
                  }
                  DispatchBufferForDecoding(client, &client->buffers[i]);
                  return;
                 }
                }
               }
               else {
                DispatchEndOfStream(client);
                return;
               }
              }
              buffer->data_len = len;
              if (client->eos) return; /* Don't dispatch new buffers if EOS already done. */
              /* Decode the contents of the input stream buffer. */
              switch (rv = DecDecode(client->decoder, buffer)) {
               case DEC_OK:
                /* Everything is good, keep on going. */
                break;
               default:
                DispatchEndOfStream(client);
                break;
              }
             }

             static void InitializedCb(ClientInst inst)
             {
              struct Client* client = (struct Client*)inst;
              /* Internal testing feature: Override HW configuration parameters */
              HwconfigOverride(client->decoder,  &tb_cfg);
              /* Start the output handling thread, if needed. */
              if (client->test_params.extra_output_thread)
               pthread_create(&client->parallel_output_thread, NULL, ParallelOutput, client);
               for (int i = 0; i < GetStreamBufferCount(client); i++) {
                if (DWLMallocLinear(client->dwl, DEFAULT_STREAM_BUFFER_SIZE,&client-
>buffers[i].buffer)) {
                 DecRelease(client->decoder);
                 return;
                }
                /* Dispatch the first buffers for decoding. When decoder finished
                * decoding each buffer it will be refilled within the callback. */
                DispatchBufferForDecoding(client, &client->buffers[i]);
               }
             }

             static void HeadersDecodedCb(ClientInst inst, struct DecSequenceInfo info)
             {
              struct Client* client = (struct Client*)inst;
              if (client->yuvsink.inst == NULL) {
               if (client->test_params.out_file_name == NULL) {
                client->yuvsink.inst = CreateSink(client);
               }
               DecSetPictureBuffers(client->decoder, NULL, 0);
              }
              static void BufferDecodedCb(ClientInst inst, struct DecInput* buffer) {
               struct Client* client = (struct Client*)inst;
                if (!client->eos) {
                 DispatchBufferForDecoding(client, buffer);
                }
```

```
            }
           }

           static void PictureReadyCb(ClientInst inst, struct DecPicture picture)
           {
            static char* pic_types[] = {" IDR", "Non-IDR (P)", "Non-IDR (B)"};
            struct Client* client = (struct Client*)inst;
            client->num_of_output_pics++;
            DEBUG_PRINT(("PIC %2d/%2d, type %s,", client->num_of_output_pics,
picture.picture_info.pic_id, pic_types[picture.picture_info.pic_coding_type]));
            if (picture.picture_info.cycles_per_mb) {
             client->cycle_count += picture.picture_info.cycles_per_mb;
             DEBUG_PRINT((" %4d cycles / mb,", picture.picture_info.cycles_per_mb));
            }
            DEBUG_PRINT((" %d x %d, Crop: (%d, %d), %d x %d %s\n",
            picture.sequence_info.pic_width,
            picture.sequence_info.pic_height,
            picture.sequence_info.crop_params.crop_left_offset,
            picture.sequence_info.crop_params.crop_top_offset,
            picture.sequence_info.crop_params.crop_out_width,
            picture.sequence_info.crop_params.crop_out_height,
            picture.picture_info.is_corrupted ? "CORRUPT" : ""));
            if (client->test_params.extra_output_thread) {
             struct DecPicture* copy = malloc(sizeof(struct DecPicture));
             *copy = picture;
             FifoPush(client->pic_fifo, copy, FIFO_EXCEPTION_DISABLE);
            } else {
             PostProcessPicture(client, &picture);
            }
           }

           static void EndOfStreamCb(ClientInst inst)
           {
            struct Client* client = (struct Client*)inst;
            client->eos = 1;
            if (client->test_params.extra_output_thread) {
             /* We're done, wait for the output to Finish it's job. */
             FifoPush(client->pic_fifo, NULL, FIFO_EXCEPTION_DISABLE);
             pthread_join(client->parallel_output_thread, NULL);
            }
            DecRelease(client->decoder);
           }

           static void ReleasedCb(ClientInst inst)
           {
            struct Client* client = (struct Client*)inst;
            for (int i = 0; i < NUM_OF_STREAM_BUFFERS; i++) {
             if (client->buffers[i].buffer.virtual_address) {
              DWLFreeLinear(client->dwl, &client->buffers[i].buffer);
             }
            }
            ReleaseSink(client);
            sem_post(&client->dec_done);
           }
           static void DispatchEndOfStream(struct Client* client) {
            if (!client->eos) {
             client->eos = 1;
             DecEndOfStream(client->decoder);
            }
           }

           static void NotifyErrorCb(ClientInst inst, u32 pic_id, enum DecRet rv)
           {
            struct Client* client = (struct Client*)inst;
            /* There's serious decoding error, so we'll consider it as end of stream to
            get the pending pictures out of the decoder. */
            DispatchEndOfStream(client);
           }
```

```
            int main(int argc, char* argv[])
            {
             struct Client client;
             memset(&client, 0, sizeof(struct Client));
             struct DecClientHandle client_if = { &client, InitializedCb, HeadersDecodedCb,
BufferDecodedCb,
             PictureReadyCb, EndOfStreamCb, ReleasedCb, NotifyErrorCb, };
             SetupDefaultParams(&client.test_params);
             ParseParams(argc, argv, &client.test_params);
             struct DecSwHwBuild build = DecGetBuild();
             /* Check whether the decoder feature meets our requirements here. */
             ...
             client.demuxer.inst = CreateDemuxer(&client);
             /* Create struct DWL. */
             struct DWLInitParam dwl_params = {DWL_CLIENT_TYPE_HEVC_DEC};
             client.dwl = DWLInit(&dwl_params);
             if (client.test_params.extra_output_thread)
             /* Create the fifo to enable parallel output processing */
             FifoInit(2, &client.pic_fifo);
             sem_init(&client.dec_done, 0, 0);
             enum DecCodec codec;
             switch (client.demuxer.GetVideoFormat(client.demuxer.inst)) {
              case BITSTREAM_HEVC:
               codec = DEC_HEVC;
               break;
              case BITSTREAM_VP9:
               codec = DEC_VP9;
               break;
              default:
               return -1;
             }
             struct DecConfig config;
             config.disable_picture_reordering= client.test_params.disable_display_order;
             config.concealment_mode = client.test_params.concealment_mode;
             switch (client.test_params.hw_format) {
              case DEC_OUT_FRM_TILED_4X4:
               config.output_format = DEC_OUT_FRM_TILED_4X4;
               break;
              case DEC_OUT_FRM_RASTER_SCAN: /* fallthrough */
              case DEC_OUT_FRM_PLANAR_420:
               if (!build.hw_config.pp_support) {
                 fprintf(stderr, "Cannot do raster output; No PP support.\n");
                 return -1;
               }
               config.output_format = DEC_OUT_FRM_RASTER_SCAN;
               break;
              default:
               return -1;
             }
             config.dwl = dwl;
             config.dwl_inst = client.dwl;
             config.max_num_pics_to_decode = client.test_params.num_of_decoded_pics;
             config.dscale_cfg = client.test_params.dscale_cfg;
             config.use_video_compressor = client.test_params.compress_bypass ? 0 : 1;
             /* Initialize the decoder. */
             DecInit(codec, &client.decoder, config, client_if) != DEC_OK;
             /* The rest is driven by the callbacks and this thread just has to Wait
             * until decoder has finished its job. */
             sem_wait(&client.dec_done);
             ReleaseDemuxer(&client);
             if (client.pic_fifo != NULL)
               FifoRelease(client.pic_fifo);
             if (client.dwl != NULL) DWLRelease(client.dwl);
               return 0;
            }
```

14.2.5 VPU G2 Memory Map/Register Definition

14.2.5.1 VPU_G2 register descriptions

14.2.5.1.1 VPU Memory map

VPU_G2 base address: 3831_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	ID register (read only) (SWREG0)	32	RO	Table 14-4
4h	Interrupt register decoder (SWREG1)	32	RW	0000_0000h
8h	Data configuration register decoder (SWREG2)	32	RW	0000_0000h
Ch	Decoder control register 0 (SWREG3)	32	RW	0000_0000h
10h	Decoder control register 1 (SWREG4)	32	RW	0000_0000h
14h	Decoder control register 2 (SWREG5)	32	RW	0000_0000h
18h	Decoder control register 3 (SWREG6)	32	RW	0000_0000h
1Ch	Decoder control register 4 (SWREG7)	32	RW	0000_0000h
20h	Decoder control register 5 (SWREG8)	32	RW	0000_0000h
24h	Decoder control register 6 (SWREG9)	32	RW	0000_0000h
28h	Decoder control register 7 (SWREG10)	32	RW	0000_0000h
2Ch	Decoder control register 8 (SWREG11)	32	RW	0000_0000h
30h	Decoder control register 9 (SWREG12)	32	RW	0000_0000h
34h	Decoder control register 10 (SWREG13)	32	RW	0000_0000h
38h	Initial ref pic list register (0-2) (SWREG14)	32	RW	0000_0000h
3Ch	Initial ref pic list register (3-5) (SWREG15)	32	RW	0000_0000h
40h	Initial ref pic list register (6-8) (SWREG16)	32	RW	0000_0000h
44h	Initial ref pic list register (9-11) (SWREG17)	32	RW	0000_0000h
48h	Initial ref pic list register (12-14) (SWREG18)	32	RW	0000_0000h
4Ch	Initial ref pic list register (15 and P 0-3) (SWREG19)	32	RW	0000_0000h
50h	Decoder control register 11 (SWREG20)	32	RW	0000_0000h
54h	Not used (SWREG21)	32	RU	0000_0000h
58h	Not used (SWREG22)	32	RU	0000_0000h
5Ch	Decoder configure status register (SWREG23)	32	RO	0000_0000h
60h	Not used (SWREG24)	32	RU	0000_0000h
64h	Not used (SWREG25)	32	RU	0000_0000h
68h	Not used (SWREG26)	32	RU	0000_0000h
6Ch	Not used (SWREG27)	32	RU	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
70h	Not used (SWREG28)	32	RU	0000_0000h
74h	Not used (SWREG29)	32	RU	0000_0000h
78h	Not used (SWREG30)	32	RU	0000_0000h
7Ch	VP9 segmentation values (SWREG31)	32	RW	0000_0000h
80h	VP9 segmentation values (SWREG32)	32	RW	0000_0000h
84h	VP9 reference picture scaling register 0 (SWREG33)	32	RW	0000_0000h
88h	VP9 reference picture scaling register 1 (SWREG34)	32	RW	0000_0000h
8Ch	VP9 reference picture scaling register 2 (SWREG35)	32	RW	0000_0000h
90h	VP9 reference picture scaling register 3 (SWREG36)	32	RW	0000_0000h
94h	VP9 reference picture scaling register 4 (SWREG37)	32	RW	0000_0000h
98h	VP9 reference picture scaling register 5 (SWREG38)	32	RW	0000_0000h
9Ch	Not used (SWREG39)	32	RU	0000_0000h
A0h	Not used (SWREG40)	32	RU	0000_0000h
A4h	Not used (SWREG41)	32	RU	0000_0000h
A8h	Not used (SWREG42)	32	RU	0000_0000h
ACh	Not used (SWREG43)	32	RU	0000_0000h
B0h	Not used (SWREG44)	32	RU	0000_0000h
B4h	Timeout control register (SWREG45)	32	RW	0000_0000h
B8h	Picture order count from current pictures for index 0-3 (SWREG46)	32	RW	0000_0000h
BCh	Picture order count from current pictures for index 4-7 (SWREG47)	32	RW	0000_0000h
C0h	Picture order count from current pictures for index 8-11 (SWREG48)	32	RW	0000_0000h
C4h	Picture order count from current pictures for index 12-15 (SWREG49)	32	RW	0000_0000h
C8h	Synthesis configuration register decoder 0 (read only) (SWREG50)	32	RO	Table 14-4
CCh	Reference picture buffer control register (SWREG51)	32	RU	0000_0000h
D0h	Reference picture buffer information register 1 (read only) (SWREG52)	32	RU	0000_0000h
D4h	Reference picture buffer information register 2 (read only) (SWREG53)	32	RU	0000_0000h
D8h	Synthesis configuration register decoder 1 (read only) (SWREG54)	32	RO	0000_0000h
DCh	Advanced prefetch control register (SWREG55)	32	RW	0000_0000h
E0h	Synthesis configuration register decoder 2 (read only) (SWREG56)	32	RO	0000_0000h
E4h	Decoder fuse register (read only) (SWREG57)	32	RU	0000_0000h
E8h	Device configuration register decoder 2 + Multi core control register (SWREG58)	32	RW	0000_0000h
ECh	Device configuration register AXI ID (SWREG59)	32	RW	0000_0000h
F0h	Synthesis configuration register decoder 3 for PP (read only) (SWREG60)	32	RO	0000_0000h
F4h	Not used (SWREG61)	32	RU	0000_0000h
F8h	HW proceed register (CU location) (SWREG62)	32	RW	0000_0000h
FCh	HW performance register (cycles running) (SWREG63)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
100h	Base address MSB (bits 63:32) for decoded luminance picture (SWREG64)	32	RW	0000_0000h
104h	Base address LSB (bits 31:0) for decoded luminance picture (SWREG65)	32	RW	0000_0000h
108h	Base address MSB (bits 63:32) for reference luminance picture index 0 (SWREG66)	32	RW	0000_0000h
10Ch	Base address LSB (bits 31:0) for reference luminance picture index 0 (SWREG67)	32	RW	0000_0000h
110h	Base address MSB (bits 63:32) for reference luminance picture index 1 (SWREG68)	32	RW	0000_0000h
114h	Base address LSB (bits 31:0) for reference luminance picture index 1 (SWREG69)	32	RW	0000_0000h
118h	Base address MSB (bits 63:32) for reference luminance picture index 2 (SWREG70)	32	RW	0000_0000h
11Ch	Base address LSB (bits 31:0) for reference luminance picture index 2 (SWREG71)	32	RW	0000_0000h
120h	Base address MSB (bits 63:32) for reference luminance picture index 3 (SWREG72)	32	RW	0000_0000h
124h	Base address LSB (bits 31:0) for reference luminance picture index 3 (SWREG73)	32	RW	0000_0000h
128h	Base address MSB (bits 63:32) for reference luminance picture index 4 (SWREG74)	32	RW	0000_0000h
12Ch	Base address LSB (bits 31:0) for reference luminance picture index 4 (SWREG75)	32	RW	0000_0000h
130h	Base address MSB (bits 63:32) for reference luminance picture index 5 (SWREG76)	32	RW	0000_0000h
134h	Base address LSB (bits 31:0) for reference luminance picture index 5 (SWREG77)	32	RW	0000_0000h
138h	Base address MSB (bits 63:32) for reference luminance picture index 6 /VP9 segment write base MSB (SWREG78)	32	RW	0000_0000h
13Ch	Base address LSB (bits 31:0) for reference luminance picture index 6 /VP9 segment write base LSB (SWREG79)	32	RW	0000_0000h
140h	Base address MSB (bits 63:32) for reference luminance picture index 7 /VP9 segment read base MSB (SWREG80)	32	RW	0000_0000h
144h	Base address LSB (bits 31:0) for reference luminance picture index 7 /VP9 segment read base LSB (SWREG81)	32	RW	0000_0000h
148h	Base address MSB (bits 63:32) for reference luminance picture index 8 (SWREG82)	32	RW	0000_0000h
14Ch	Base address LSB (bits 31:0) for reference luminance picture index 8 (SWREG83)	32	RW	0000_0000h
150h	Base address MSB (bits 63:32) for reference luminance picture index 9 (SWREG84)	32	RW	0000_0000h
154h	Base address LSB (bits 31:0) for reference luminance picture index 9 (SWREG85)	32	RW	0000_0000h
158h	Base address MSB (bits 63:32) for reference luminance picture index 10 (SWREG86)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
15Ch	Base address LSB (bits 31:0) for reference luminance picture index 10 (SWREG87)	32	RW	0000_0000h
160h	Base address MSB (bits 63:32) for reference luminance picture index 11 (SWREG88)	32	RW	0000_0000h
164h	Base address LSB (bits 31:0) for reference luminance picture index 11 (SWREG89)	32	RW	0000_0000h
168h	Base address MSB (bits 63:32) for reference luminance picture index 12 (SWREG90)	32	RW	0000_0000h
16Ch	Base address LSB (bits 31:0) for reference luminance picture index 12 (SWREG91)	32	RW	0000_0000h
170h	Base address MSB (bits 63:32) for reference luminance picture index 13 (SWREG92)	32	RW	0000_0000h
174h	Base address LSB (bits 31:0) for reference luminance picture index 13 (SWREG93)	32	RW	0000_0000h
178h	Base address MSB (bits 63:32) for reference luminance picture index 14 (SWREG94)	32	RW	0000_0000h
17Ch	Base address LSB (bits 31:0) for reference luminance picture index 14 (SWREG95)	32	RW	0000_0000h
180h	Base address MSB (bits 63:32) for reference luminance picture index 15 (SWREG96)	32	RW	0000_0000h
184h	Base address LSB (bits 31:0) for reference luminance picture index 15 (SWREG97)	32	RW	0000_0000h
188h	Base address MSB (bits 63:32) for decoded chrominance picture (SWREG98)	32	RW	0000_0000h
18Ch	Base address LSB (bits 31:0) for decoded chrominance picture (SWREG99)	32	RW	0000_0000h
190h	Base address MSB (bits 63:32) for reference chrominance picture index 0 (SWREG100)	32	RW	0000_0000h
194h	Base address LSB (bits 31:0) for reference chrominance picture index 0 (SWREG101)	32	RW	0000_0000h
198h	Base address MSB (bits 63:32) for reference chrominance picture index 1 (SWREG102)	32	RW	0000_0000h
19Ch	Base address LSB (bits 31:0) for reference chrominance picture index 1 (SWREG103)	32	RW	0000_0000h
1A0h	Base address MSB (bits 63:32) for reference chrominance picture index 2 (SWREG104)	32	RW	0000_0000h
1A4h	Base address LSB (bits 31:0) for reference chrominance picture index 2 (SWREG105)	32	RW	0000_0000h
1A8h	Base address MSB (bits 63:32) for reference chrominance picture index 3 (SWREG106)	32	RW	0000_0000h
1ACh	Base address LSB (bits 31:0) for reference chrominance picture index 3 (SWREG107)	32	RW	0000_0000h
1B0h	Base address MSB (bits 63:32) for reference chrominance picture index 4 (SWREG108)	32	RW	0000_0000h
1B4h	Base address LSB (bits 31:0) for reference chrominance picture index 4 (SWREG109)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1B8h	Base address MSB (bits 63:32) for reference chrominance picture index 5 (SWREG110)	32	RW	0000_0000h
1BCh	Base address LSB (bits 31:0) for reference chrominance picture index 5 (SWREG111)	32	RW	0000_0000h
1C0h	Base address MSB (bits 63:32) for reference chrominance picture index 6 (SWREG112)	32	RW	0000_0000h
1C4h	Base address LSB (bits 31:0) for reference chrominance picture index 6 (SWREG113)	32	RW	0000_0000h
1C8h	Base address MSB (bits 63:32) for reference chrominance picture index 7 (SWREG114)	32	RW	0000_0000h
1CCh	Base address LSB (bits 31:0) for reference chrominance picture index 7 (SWREG115)	32	RW	0000_0000h
1D0h	Base address MSB (bits 63:32) for reference chrominance picture index 8 (SWREG116)	32	RW	0000_0000h
1D4h	Base address LSB (bits 31:0) for reference chrominance picture index 8 (SWREG117)	32	RW	0000_0000h
1D8h	Base address MSB (bits 63:32) for reference chrominance picture index 9 (SWREG118)	32	RW	0000_0000h
1DCh	Base address LSB (bits 31:0) for reference chrominance picture index 9 (SWREG119)	32	RW	0000_0000h
1E0h	Base address MSB (bits 63:32) for reference chrominance picture index 10 (SWREG120)	32	RW	0000_0000h
1E4h	Base address LSB (bits 31:0) for reference chrominance picture index 10 (SWREG121)	32	RW	0000_0000h
1E8h	Base address MSB (bits 63:32) for reference chrominance picture index 11 (SWREG122)	32	RW	0000_0000h
1ECh	Base address LSB (bits 31:0) for reference chrominance picture index 11 (SWREG123)	32	RW	0000_0000h
1F0h	Base address MSB (bits 63:32) for reference chrominance picture index 12 (SWREG124)	32	RW	0000_0000h
1F4h	Base address LSB (bits 31:0) for reference chrominance picture index 12 (SWREG125)	32	RW	0000_0000h
1F8h	Base address MSB (bits 63:32) for reference chrominance picture index 13 (SWREG126)	32	RW	0000_0000h
1FCh	Base address LSB (bits 31:0) for reference chrominance picture index 13 (SWREG127)	32	RW	0000_0000h
200h	Base address MSB (bits 63:32) for reference chrominance picture index 14 (SWREG128)	32	RW	0000_0000h
204h	Base address LSB (bits 31:0) for reference chrominance picture index 14 (SWREG129)	32	RW	0000_0000h
208h	Base address MSB (bits 63:32) for reference chrominance picture index 15 (SWREG130)	32	RW	0000_0000h
20Ch	Base address LSB (bits 31:0) for reference chrominance picture index 15 (SWREG131)	32	RW	0000_0000h
210h	Base address MSB (bits 63:32) for decoded direct mode MVS (SWREG132)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
214h	Base address LSB (bits 31:0) for decoded direct mode MVS (SWREG133)	32	RW	0000_0000h
218h	Base address MSB (bits 63:32) for reference direct mode MVS index 0 (SWREG134)	32	RW	0000_0000h
21Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 0 (SWREG135)	32	RW	0000_0000h
220h	Base address MSB (bits 63:32) for reference direct mode MVS index 1 (SWREG136)	32	RW	0000_0000h
224h	Base address LSB (bits 31:0) for reference direct mode MVS index 1 (SWREG137)	32	RW	0000_0000h
228h	Base address MSB (bits 63:32) for reference direct mode MVS index 2 (SWREG138)	32	RW	0000_0000h
22Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 2 (SWREG139)	32	RW	0000_0000h
230h	Base address MSB (bits 63:32) for reference direct mode MVS index 3 (SWREG140)	32	RW	0000_0000h
234h	Base address LSB (bits 31:0) for reference direct mode MVS index 3 (SWREG141)	32	RW	0000_0000h
238h	Base address MSB (bits 63:32) for reference direct mode MVS index 4 (SWREG142)	32	RW	0000_0000h
23Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 4 (SWREG143)	32	RW	0000_0000h
240h	Base address MSB (bits 63:32) for reference direct mode MVS index 5 (SWREG144)	32	RW	0000_0000h
244h	Base address LSB (bits 31:0) for reference direct mode MVS index 5 (SWREG145)	32	RW	0000_0000h
248h	Base address MSB (bits 63:32) for reference direct mode MVS index 6 (SWREG146)	32	RW	0000_0000h
24Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 6 (SWREG147)	32	RW	0000_0000h
250h	Base address MSB (bits 63:32) for reference direct mode MVS index 7 (SWREG148)	32	RW	0000_0000h
254h	Base address LSB (bits 31:0) for reference direct mode MVS index 7 (SWREG149)	32	RW	0000_0000h
258h	Base address MSB (bits 63:32) for reference direct mode MVS index 8 (SWREG150)	32	RW	0000_0000h
25Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 8 (SWREG151)	32	RW	0000_0000h
260h	Base address MSB (bits 63:32) for reference direct mode mode MVS index 9 (SWREG152)	32	RW	0000_0000h
264h	Base address LSB (bits 31:0) for reference direct mode mode MVS index 9 (SWREG153)	32	RW	0000_0000h
268h	Base address MSB (bits 63:32) for reference direct mode MVS index 10 (SWREG154)	32	RW	0000_0000h
26Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 10 (SWREG155)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
270h	Base address MSB (bits 63:32) for reference direct mode MVS index 11 (SWREG156)	32	RW	0000_0000h
274h	Base address LSB (bits 31:0) for reference direct mode MVS index 11 (SWREG157)	32	RW	0000_0000h
278h	Base address MSB (bits 63:32) for reference direct mode MVS index 12 (SWREG158)	32	RW	0000_0000h
27Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 12 (SWREG159)	32	RW	0000_0000h
280h	Base address MSB (bits 63:32) for reference direct mode MVS index 13 (SWREG160)	32	RW	0000_0000h
284h	Base address LSB (bits 31:0) for reference direct mode MVS index 13 (SWREG161)	32	RW	0000_0000h
288h	Base address MSB (bits 63:32) for reference direct mode MVS index 14 (SWREG162)	32	RW	0000_0000h
28Ch	Base address LSB (bits 31:0) for reference direct mode MVS index 14 (SWREG163)	32	RW	0000_0000h
290h	Base address MSB (bits 63:32) for reference direct mode MVS index 15 (SWREG164)	32	RW	0000_0000h
294h	Base address LSB (bits 31:0) for reference direct mode MVS index 15 (SWREG165)	32	RW	0000_0000h
298h	Base address MSB (bits 63:32) for tile sizes (SWREG166)	32	RW	0000_0000h
29Ch	Base address LSB (bits 31:0) for tile sizes (SWREG167)	32	RW	0000_0000h
2A0h	Base address MSB (bits 63:32) for / stream start address/decoded end addr register (SWREG168)	32	RW	0000_0000h
2A4h	Base address LSB (bits 31:0) for / stream start address/decoded end addr register (SWREG169)	32	RW	0000_0000h
2A8h	Base address MSB (bits 63:32) for scaling lists / VP9 CTX counter values (SWREG170)	32	RW	0000_0000h
2ACh	Base address LSB (bits 31:0) for scaling lists / VP9 CTX counter values (SWREG171)	32	RW	0000_0000h
2B0h	Base address MSB (bits 63:32) for stream propability tables (SWREG172)	32	RW	0000_0000h
2B4h	Base address LSB (bits 31:0) for stream propability tables (SWREG173)	32	RW	0000_0000h
2B8h	Base address MSB (bits 63:32) for decoder output raster scan Y picture (SWREG174)	32	RW	0000_0000h
2BCh	Base address LSB (bits 31:0) for decoder output raster scan Y picture (SWREG175)	32	RW	0000_0000h
2C0h	Base address MSB (bits 63:32) for decoder output raster scan C picture (SWREG176)	32	RW	0000_0000h
2C4h	Base address LSB (bits 31:0) for decoder output raster scan C picture (SWREG177)	32	RW	0000_0000h
2C8h	Base address MSB (bits 63:32) for tile border coeffients of filter (SWREG178)	32	RW	0000_0000h
2CCh	Base address LSB (bits 31:0) for tile border coeffients of filter (SWREG179)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
2D0h	Base address MSB (bits 63:32) for tile border coeffients of sao (SWREG180)	32	RW	0000_0000h
2D4h	Base address LSB (bits 31:0) for tile border coeffients of sao (SWREG181)	32	RW	0000_0000h
2D8h	Base address MSB (bits 63:32) for tile border bsd control data (SWREG182)	32	RW	0000_0000h
2DCh	Base address LSB (bits 31:0) for tile border bsd control data (SWREG183)	32	RW	0000_0000h
2E0h	Raster scan down scale control register MSM (SWREG184)	32	RW	0000_0000h
2E4h	Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture (SWREG185)	32	RW	0000_0000h
2E8h	Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture (SWREG186)	32	RW	0000_0000h
2ECh	Base address MSB (bits 63:32) for decoder output raster scan down scale C picture (SWREG187)	32	RW	0000_0000h
2F0h	Base address LSB (bits 31:0) for decoder output raster scan down scale C picture (SWREG188)	32	RW	0000_0000h
2F4h	Base address MSB (bits 63:32) for decoder output compress luminance table (SWREG189)	32	RW	0000_0000h
2F8h	Base address LSB (bits 31:0) for decoder output compress luminance table (SWREG190)	32	RW	0000_0000h
2FCh	Base address MSB (bits 63:32) for reference compress luminance table index 0 (SWREG191)	32	RW	0000_0000h
300h	Base address LSB (bits 31:0) for reference compress luminance table index 0 (SWREG192)	32	RW	0000_0000h
304h	Base address MSB (bits 63:32) for reference compress luminance table index 1 (SWREG193)	32	RW	0000_0000h
308h	Base address LSB (bits 31:0) for reference compress luminance table index 1 (SWREG194)	32	RW	0000_0000h
30Ch	Base address MSB (bits 63:32) for reference compress luminance table index 2 (SWREG195)	32	RW	0000_0000h
310h	Base address LSB (bits 31:0) for reference compress luminance table index 2 (SWREG196)	32	RW	0000_0000h
314h	Base address MSB (bits 63:32) for reference compress luminance table index 3 (SWREG197)	32	RW	0000_0000h
318h	Base address LSB (bits 31:0) for reference compress luminance table index 3 (SWREG198)	32	RW	0000_0000h
31Ch	Base address MSB (bits 63:32) for reference compress luminance table index 4 (SWREG199)	32	RW	0000_0000h
320h	Base address LSB (bits 31:0) for reference compress luminance table index 4 (SWREG200)	32	RW	0000_0000h
324h	Base address MSB (bits 63:32) for reference compress luminance table index 5 (SWREG201)	32	RW	0000_0000h
328h	Base address LSB (bits 31:0) for reference compress luminance table index 5 (SWREG202)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
32Ch	Base address MSB (bits 63:32) for reference compress luminance table index 6 (SWREG203)	32	RW	0000_0000h
330h	Base address LSB (bits 31:0) for reference compress luminance table index 6 (SWREG204)	32	RW	0000_0000h
334h	Base address MSB (bits 63:32) for reference compress luminance table index 7 (SWREG205)	32	RW	0000_0000h
338h	Base address LSB (bits 31:0) for reference compress luminance table index 7 (SWREG206)	32	RW	0000_0000h
33Ch	Base address MSB (bits 63:32) for reference compress luminance table index 8 (SWREG207)	32	RW	0000_0000h
340h	Base address LSB (bits 31:0) for reference compress luminance table index 8 (SWREG208)	32	RW	0000_0000h
344h	Base address MSB (bits 63:32) for reference compress luminance table index 9 (SWREG209)	32	RW	0000_0000h
348h	Base address LSB (bits 31:0) for reference compress luminance table index 9 (SWREG210)	32	RW	0000_0000h
34Ch	Base address MSB (bits 63:32) for reference compress luminance table index 10 (SWREG211)	32	RW	0000_0000h
350h	Base address LSB (bits 31:0) for reference compress luminance table index 10 (SWREG212)	32	RW	0000_0000h
354h	Base address MSB (bits 63:32) for reference compress luminance table index 11 (SWREG213)	32	RW	0000_0000h
358h	Base address LSB (bits 31:0) for reference compress luminance table index 11 (SWREG214)	32	RW	0000_0000h
35Ch	Base address MSB (bits 63:32) for reference compress luminance table index 12 (SWREG215)	32	RW	0000_0000h
360h	Base address LSB (bits 31:0) for reference compress luminance table index 12 (SWREG216)	32	RW	0000_0000h
364h	Base address MSB (bits 63:32) for reference compress luminance table index 13 (SWREG217)	32	RW	0000_0000h
368h	Base address LSB (bits 31:0) for reference compress luminance table index 13 (SWREG218)	32	RW	0000_0000h
36Ch	Base address MSB (bits 63:32) for reference compress luminance table index 14 (SWREG219)	32	RW	0000_0000h
370h	Base address LSB (bits 31:0) for reference compress luminance table index 14 (SWREG220)	32	RW	0000_0000h
374h	Base address MSB (bits 63:32) for reference compress luminance table index 15 (SWREG221)	32	RW	0000_0000h
378h	Base address LSB (bits 31:0) for reference compress luminance table index 15 (SWREG222)	32	RW	0000_0000h
37Ch	Base address MSB (bits 63:32) for decoder output compress chrominance table (SWREG223)	32	RW	0000_0000h
380h	Base address LSB (bits 31:0) for decoder output compress chrominance table (SWREG224)	32	RW	0000_0000h
384h	Base address MSB (bits 63:32) for reference compress chrominance table index 0 (SWREG225)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
388h	Base address LSB (bits 31:0) for reference compress chrominance table index 0 (SWREG226)	32	RW	0000_0000h
38Ch	Base address MSB (bits 63:32) for reference compress chrominance table index 1 (SWREG227)	32	RW	0000_0000h
390h	Base address LSB (bits 31:0) for reference compress chrominance table index 1 (SWREG228)	32	RW	0000_0000h
394h	Base address MSB (bits 63:32) for reference compress chrominance table index 2 (SWREG229)	32	RW	0000_0000h
398h	Base address LSB (bits 31:0) for reference compress chrominance table index 2 (SWREG230)	32	RW	0000_0000h
39Ch	Base address MSB (bits 63:32) for reference compress chrominance table index 3 (SWREG231)	32	RW	0000_0000h
3A0h	Base address LSB (bits 31:0) for reference compress chrominance table index 3 (SWREG232)	32	RW	0000_0000h
3A4h	Base address MSB (bits 63:32) for reference compress chrominance table index 4 (SWREG233)	32	RW	0000_0000h
3A8h	Base address LSB (bits 31:0) for reference compress chrominance table index 4 (SWREG234)	32	RW	0000_0000h
3ACh	Base address MSB (bits 63:32) for reference compress chrominance table index 5 (SWREG235)	32	RW	0000_0000h
3B0h	Base address LSB (bits 31:0) for reference compress chrominance table index 5 (SWREG236)	32	RW	0000_0000h
3B4h	Base address MSB (bits 63:32) for reference compress chrominance table index 6 (SWREG237)	32	RW	0000_0000h
3B8h	Base address LSB (bits 31:0) for reference compress chrominance table index 6 (SWREG238)	32	RW	0000_0000h
3BCh	Base address MSB (bits 63:32) for reference compress chrominance table index 7 (SWREG239)	32	RW	0000_0000h
3C0h	Base address LSB (bits 31:0) for reference compress chrominance table index 7 (SWREG240)	32	RW	0000_0000h
3C4h	Base address MSB (bits 63:32) for reference compress chrominance table index 8 (SWREG241)	32	RW	0000_0000h
3C8h	Base address LSB (bits 31:0) for reference compress chrominance table index 8 (SWREG242)	32	RW	0000_0000h
3CCh	Base address MSB (bits 63:32) for reference compress chrominance table index 9 (SWREG243)	32	RW	0000_0000h
3D0h	Base address LSB (bits 31:0) for reference compress chrominance table index 9 (SWREG244)	32	RW	0000_0000h
3D4h	Base address MSB (bits 63:32) for reference compress chrominance table index 10 (SWREG245)	32	RW	0000_0000h
3D8h	Base address LSB (bits 31:0) for reference compress chrominance table index 10 (SWREG246)	32	RW	0000_0000h
3DCh	Base address MSB (bits 63:32) for reference compress chrominance table index 11 (SWREG247)	32	RW	0000_0000h
3E0h	Base address LSB (bits 31:0) for reference compress chrominance table index 11 (SWREG248)	32	RW	0000_0000h

Table continues on the next page...

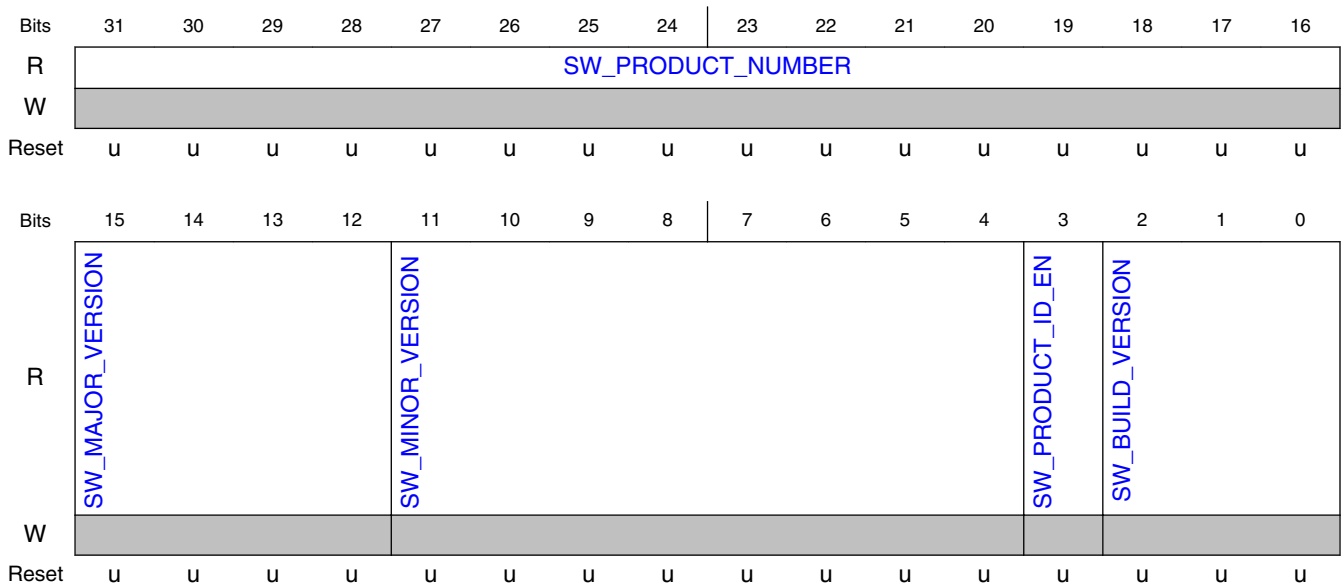
Offset	Register	Width (In bits)	Access	Reset value
3E4h	Base address MSB (bits 63:32) for reference compress chrominance table index 12 (SWREG249)	32	RW	0000_0000h
3E8h	Base address LSB (bits 31:0) for reference compress chrominance table index 12 (SWREG250)	32	RW	0000_0000h
3ECh	Base address MSB (bits 63:32) for reference compress chrominance table index 13 (SWREG251)	32	RW	0000_0000h
3F0h	Base address LSB (bits 31:0) for reference compress chrominance table index 13 (SWREG252)	32	RW	0000_0000h
3F4h	Base address MSB (bits 63:32) for reference compress chrominance table index 14 (SWREG253)	32	RW	0000_0000h
3F8h	Base address LSB (bits 31:0) for reference compress chrominance table index 14 (SWREG254)	32	RW	0000_0000h
3FCh	Base address MSB (bits 63:32) for reference compress chrominance table index 15 (SWREG255)	32	RW	0000_0000h
400h	Base address LSB (bits 31:0) for reference compress chrominance table index 15 (SWREG256)	32	RW	0000_0000h
404h	Not used (SWREG257)	32	RU	0000_0000h
408h	input stream buffer length (SWREG258)	32	RW	0000_0000h
40Ch	input stream buffer start offset (SWREG259)	32	RW	0000_0000h

14.2.5.1.2 ID register (read only) (SWREG0)

14.2.5.1.2.1 Offset

Register	Offset
SWREG0	0h

14.2.5.1.2.2 Diagram



14.2.5.1.2.3 Fields

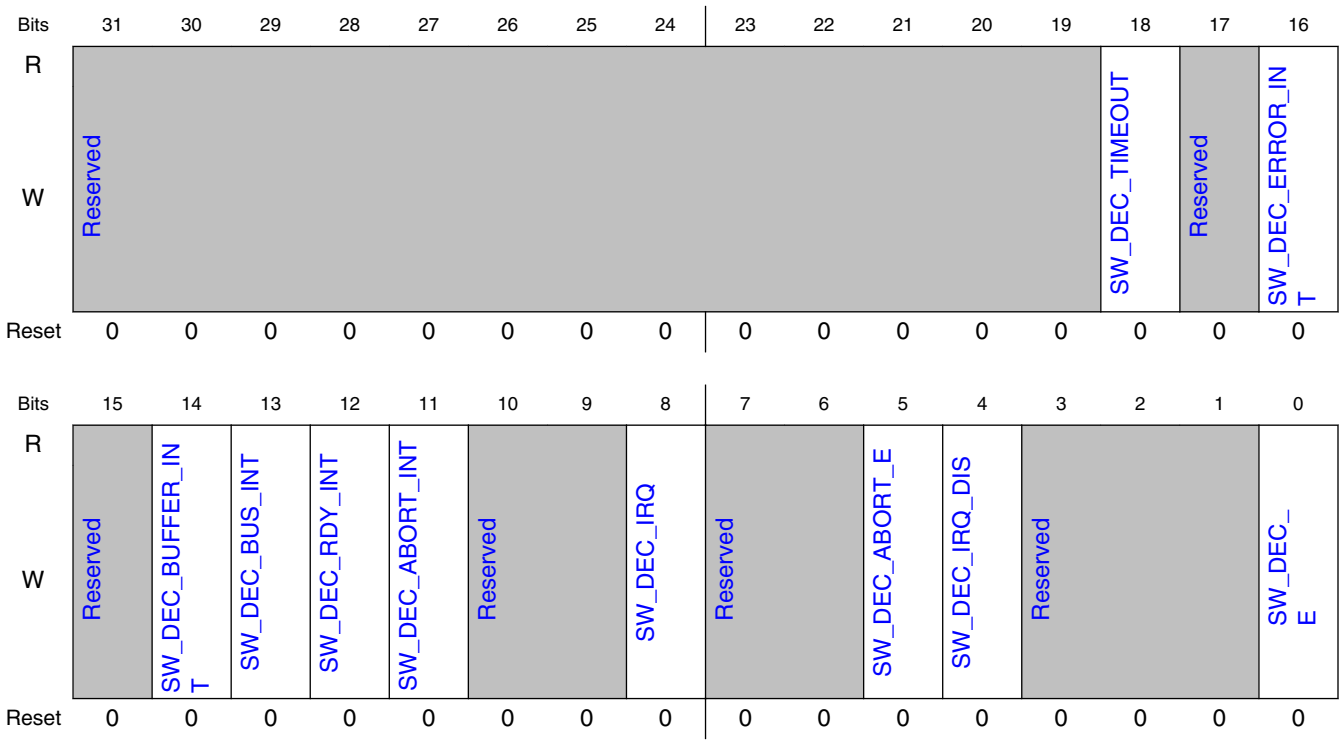
Field	Function
31-16 SW_PRODUCT_NUMBER	Product number (g2)
15-12 SW_MAJOR_VERSION	Major version
11-4 SW_MINOR_VERSION	Minor version
3 SW_PRODUCT_ID_EN	ASCII type product ID enable
2-0 SW_BUILD_VERSION	Build version (core number)

14.2.5.1.3 Interrupt register decoder (SWREG1)

14.2.5.1.3.1 Offset

Register	Offset
SWREG1	4h

14.2.5.1.3.2 Diagram



14.2.5.1.3.3 Fields

Field	Function
31-19 —	Reserved.
18 SW_DEC_TIMEOUT	Interrupt status bit decoder timeout. When high the decoder has been idling for too long. HW will self reset. Possible only if timeout interrupt is enabled
17 —	Reserved.
16 SW_DEC_ERROR_INT	Interrupt status bit input stream error. When high an error is found in input data stream decoding. HW will self reset.
15	Reserved.

Table continues on the next page...

Field	Function
—	
14 SW_DEC_BUFFER_INT	Interrupt status bit input buffer empty. When high input stream buffer is empty but picture is not ready. HW will not self reset.
13 SW_DEC_BUS_INT	Interrupt status bit bus. Error response from bus. HW will self reset.
12 SW_DEC_RDY_INT	Interrupt status bit decoder. When this bit is high decoder has decoded a picture. HW will self reset.
11 SW_DEC_ABORT_INT	Interrupt status bit decoding aborted. When this bit is high decoder has aborted the current picture decoding as SW requested (sw_dec_abort_e). Decoder self reset and sw_dec_abort_e written low
10-9 —	Reserved.
8 SW_DEC_IRQ	Decoder IRQ. When high decoder requests an interrupt. SW will reset this after interrupt is handled.
7-6 —	Reserved.
5 SW_DEC_ABORT_E	Abort decoding enable. Setting this bit high will cause HW to abort decoding and safely to reset itself down. After abort is complete the corresponding interrupt status is set and this bit is set low as well as the decoder enable.
4 SW_DEC_IRQ_DIS	Decoder IRQ disable. When high there are no interrupts concerning decoder from HW. Polling must be used to see the interrupt statuses.
3-1 —	Reserved.
0 SW_DEC_E	Decoder enable. Setting this bit high will start the decoding operation. HW will reset this when picture is processed or ASO or stream error is detected or bus error or timeout interrupt is given.

14.2.5.1.4 Data configuration register decoder (SWREG2)

14.2.5.1.4.1 Offset

Register	Offset
SWREG2	8h

14.2.5.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_DEC_STRM_SWAP				SW_DEC_PIC_SWAP				SW_DEC_DIRMV_SWAP				SW_DEC_TAB0_SWAP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_DEC_TAB1_SWAP				SW_DEC_TAB2_SWAP				SW_DEC_TAB3_SWAP				SW_DEC_RSCAN_SWAP			
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.4.3 Fields

Field	Function
31-28 SW_DEC_STRM_SWAP	Byte swap configuration for stream data 4 Bit byte order vector to control byte locations inside HW internal 128 bit data vector. For 64 and 32 bit external bus widths, the data is first gathered to 128 bit width and then bytes swapped accordingly: Bit 0: 8 bit swap Bit 1: 16 bit swap Bit 2: 32 bit swap Bit 3: 64 bit swap '0000' = 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 '0001' = 1-0-3-2-5-4-7-6... '0010' = 2-3-0-1-6-7-4-5... '0011' = 3-2-1-0-7-6-5-4... '0100' = 4-5-6-7-0-1-2-3... ... '0111' = 7-6-5-4-3-2-1-0 '1000' = 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7... ...

Table continues on the next page...

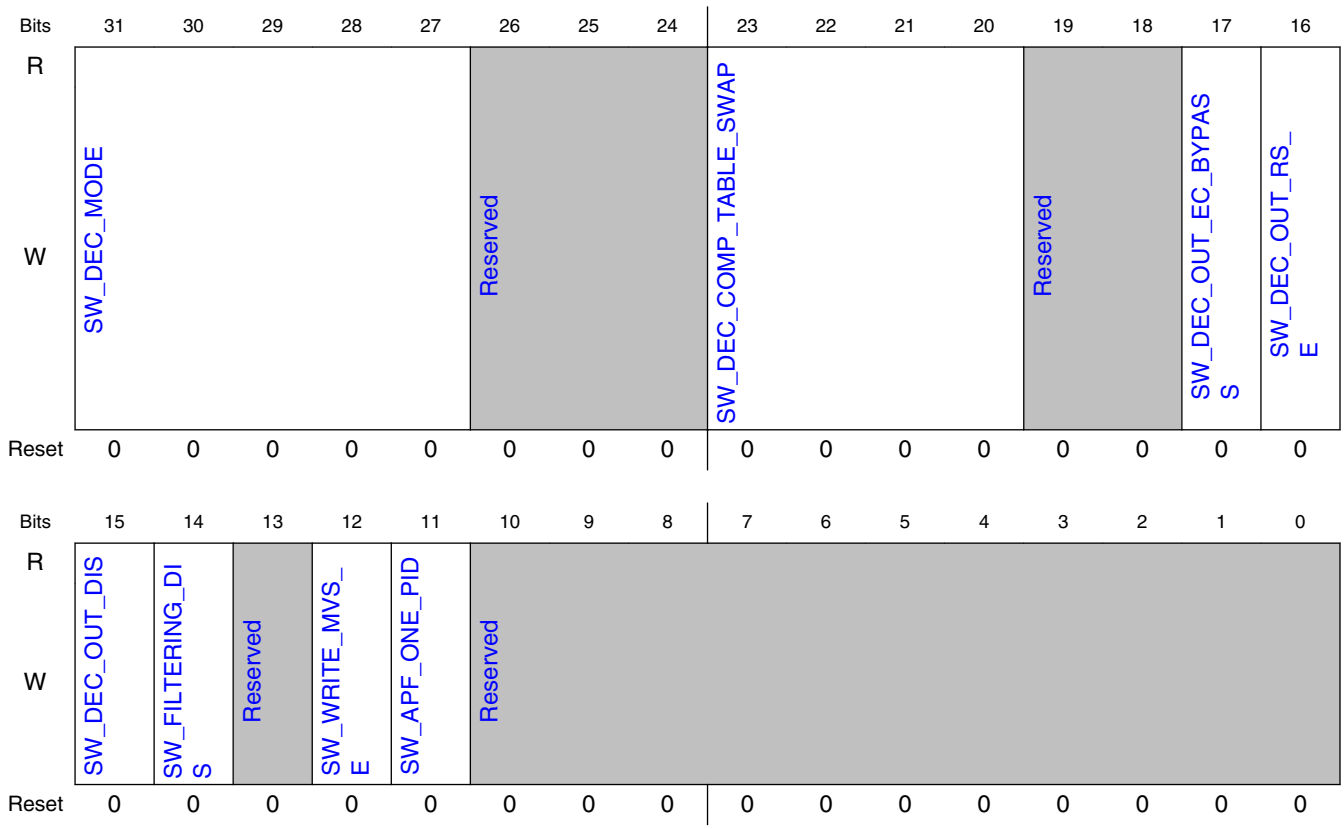
Field	Function
	'1111'=15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
27-24 SW_DEC_PIC_SWAP	Byte swap configuration for decoder reference output picture data
23-20 SW_DEC_DIRMV_SWAP	Byte swap configuration for direct mode MV data (read/write)
19-16 SW_DEC_TAB0_SWAP	Byte swap configuration for VP9 stream propability tables
15-12 SW_DEC_TAB1_SWAP	Byte swap configuration for HEVC scaling lists / VP9 segmentation map read/write
11-8 SW_DEC_TAB2_SWAP	Byte swap configuration for VP9 CTX counter values
7-4 SW_DEC_TAB3_SWAP	Byte swap configuration for tile sizes
3-0 SW_DEC_RSCAN_SWAP	Byte swap for raster scan output picture data

14.2.5.1.5 Decoder control register 0 (SWREG3)

14.2.5.1.5.1 Offset

Register	Offset
SWREG3	Ch

14.2.5.1.5.2 Diagram



14.2.5.1.5.3 Fields

Field	Function
31-27 SW_DEC_MODE	Decoding mode: 00000-01011b - Reserved 01100b - HEVC 01101b - VP9 01110-11111b - Reserved
26-24 —	Reserved.
23-20 SW_DEC_COMP_TABLE_SWAP	Byte swap configuration for compress table data
19-18 —	Reserved.
17 SW_DEC_OUT_EC_BYPASS	Compress bypass

Table continues on the next page...

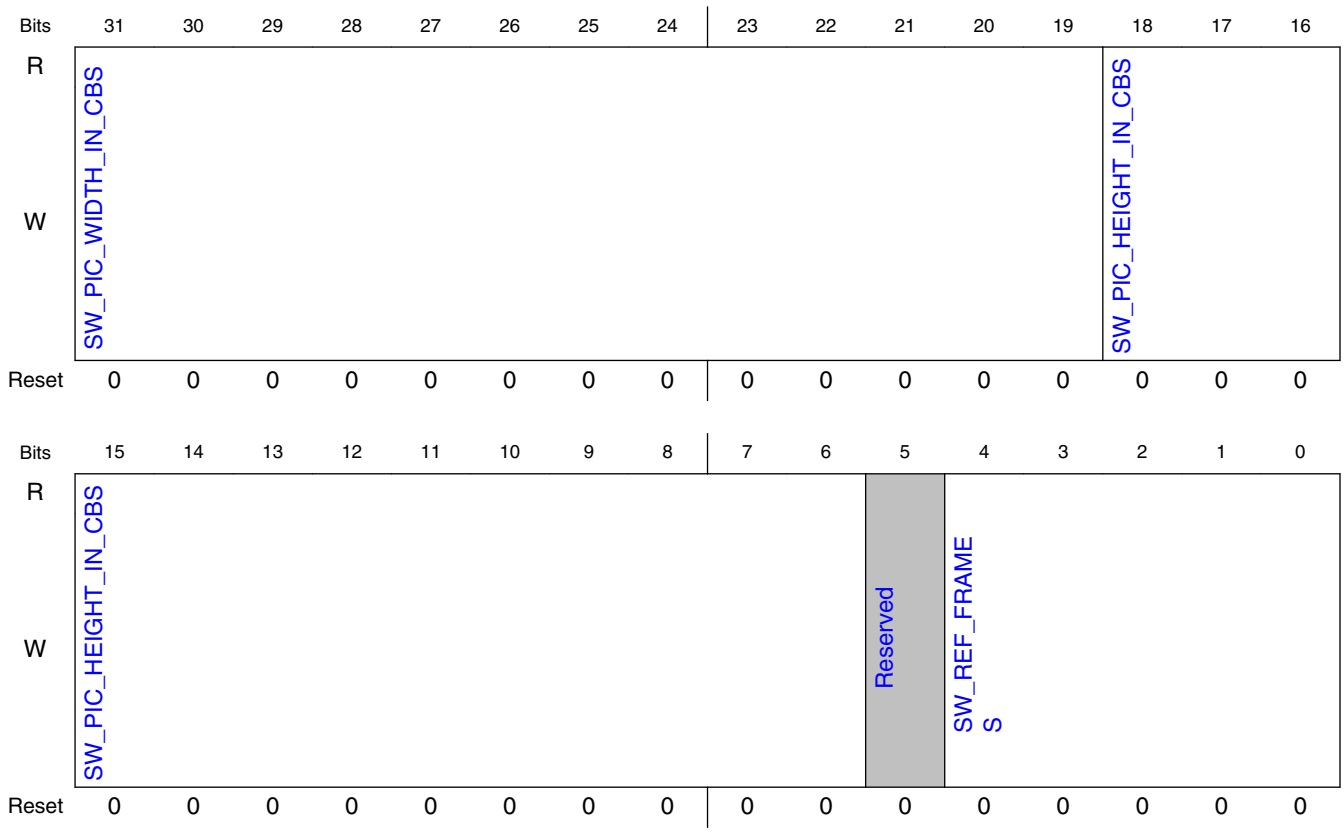
Field	Function
16 SW_DEC_OUT_RS_E	Raster scan output enable. If high decoder writes the raster scan output if the configuration of Decoder includes PP raster scan output
15 SW_DEC_OUT_DIS	Disable decoder output picture writing 0b - Decoder output picture is written to external memory 1b - Decoder output picture is not written to external memory
14 SW_FILTERING_DIS	De-block filtering disable 0b - Filtering is enabled for current picture 1b - Filtering is disabled for current picture
13 —	Reserved.
12 SW_WRITE_MVS_E	Direct mode motion vector write enable for current picture 0b - Writing disabled for current picture. 1b - The direct mode motion vectors are written to external memory. HEVC/VP9 direct mode motion vectors are written to DPB aside with the corresponding reference picture. Other decoding mode dir mode mvs are written to external memory starting from sw_dir_mv_base.
11 SW_APF_ONE_PID	Prefetch partitions that have the same pic_id together
10-0 —	Reserved.

14.2.5.1.6 Decoder control register 1 (SWREG4)

14.2.5.1.6.1 Offset

Register	Offset
SWREG4	10h

14.2.5.1.6.2 Diagram



14.2.5.1.6.3 Fields

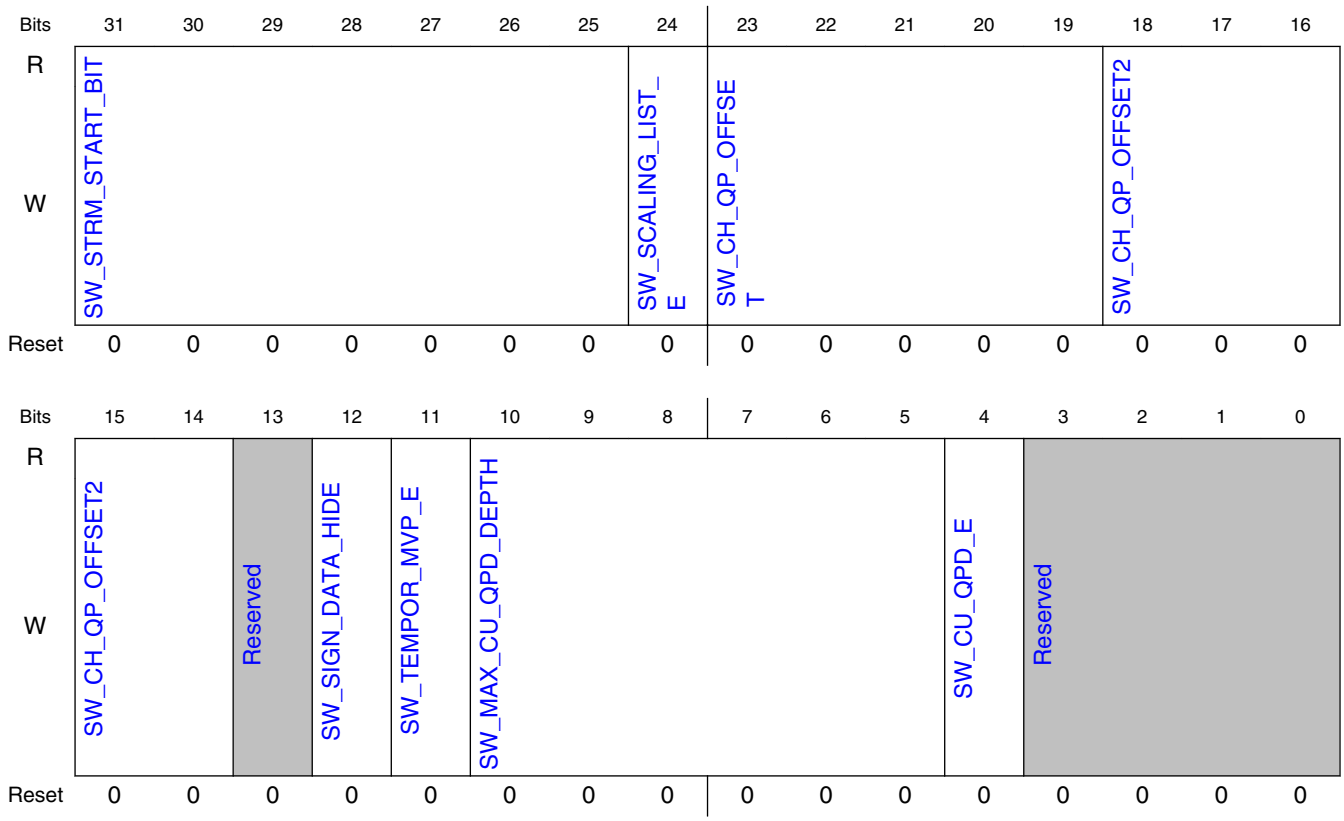
Field	Function
31-19 SW_PIC_WIDTH_IN_CBS	Picture width in min coded blocks (min = 8pix)
18-6 SW_PIC_HEIGHT_IN_CBS	Picture height in min coded blocks (min = 8pix)
5 —	Reserved.
4-0 SW_REF_FRAMES	HEVC: num_ref_frames maximum number of short and long term reference frames in decoded picture buffer

14.2.5.1.7 Decoder control register 2 (SWREG5)

14.2.5.1.7.1 Offset

Register	Offset
SWREG5	14h

14.2.5.1.7.2 Diagram



14.2.5.1.7.3 Fields

Field	Function
31-25 SW_STRM_START_BIT	Exact bit of stream start word where decoding can be started (assosiates with sw_rlc_vlc_base)
24 SW_SCALING_LIST_E	Scaling matrix enable 0b - Normal transform 1b - Use scaling matrix for transform (read from external memory)
23-19 SW_CH_QP_OFFSET	Chroma Qp filter offset. (For HEVC this offset concerns Cb only)

Table continues on the next page...

Field	Function
18-14 SW_CH_QP_OFFSET2	Chroma Qp filter offset for cr type
13 —	Reserved.
12 SW_SIGN_DATA_HIDE	Flag for stream decoding
11 SW_TEMPOR_MVP_E	Temporal mvp enable
10-5 SW_MAX_CU_QPD_DEPTH	Max CU qp delta depth
4 SW_CU_QPD_E	CU qp delta enable
3-0 —	Reserved.

14.2.5.1.8 Decoder control register 3 (SWREG6)

14.2.5.1.8.1 Offset

Register	Offset
SWREG6	18h

14.2.5.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.8.3 Fields

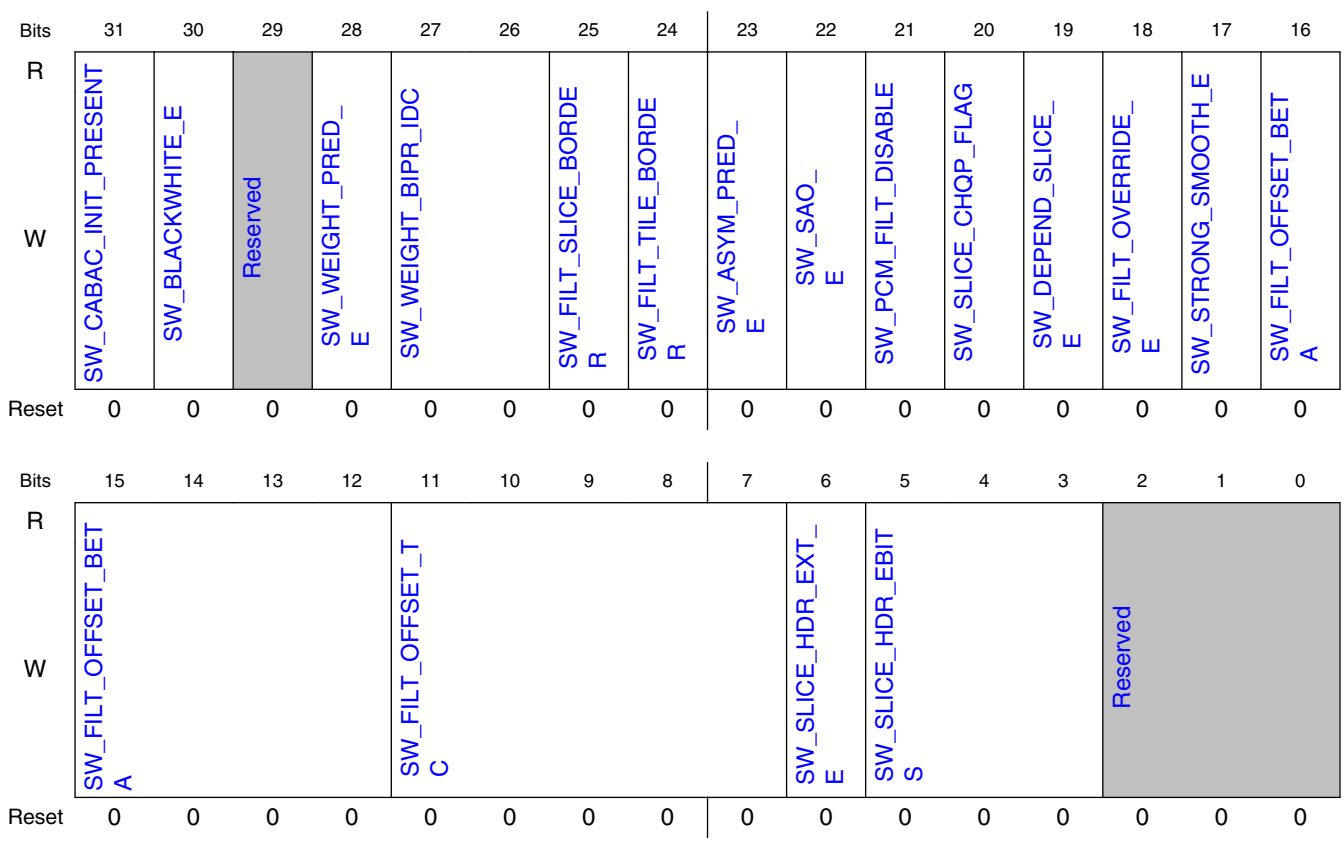
Field	Function
31-0 SW_STREAM_LEN	Amount of stream data bytes in input buffer. If the given buffer size is not enough for finishing the picture the corresponding interrupt is given and new stream buffer base address and stream buffer size information should be given (assosiates with sw_rlc_vlc_base). For HEVC the buffer must include at least data for one slice/VP of the picture

14.2.5.1.9 Decoder control register 4 (SWREG7)

14.2.5.1.9.1 Offset

Register	Offset
SWREG7	1Ch

14.2.5.1.9.2 Diagram



14.2.5.1.9.3 Fields

Field	Function
31 SW_CABAC_INIT_PRESENT	CABAC init present enable for stream decoding
30 SW_BLACKWHITE_E	Sampling 0b - 4:2:0 sampling format 1b - 4:0:0 sampling format (H264 monochroma)
29 —	Reserved.
28 SW_WEIGHT_PRED_E	Weighted prediction enable for P slices
27-26 SW_WEIGHT_BIPR_IDC	Weighted prediction specification 00b - Default weighted prediction is applied to B slices 01b - Explicit weighted prediction shall be applied to B slices 10b - NA 11b - NA
25 SW_FILT_SLICE_BORDER	Filter enable over slice border
24 SW_FILT_TILE_BORDER	Filter enable over tile border
23 SW_ASYM_PRED_E	Asymmetric prediction flag for stream decoding
22 SW_SAO_E	Sample Adaptive Offset enable for stream decoding
21 SW_PCM_FILT_DISABLE	Disable for PCM loop filtering
20 SW_SLICE_CHQP_FLAG	Slice header flag for chroma QP present (if it is included in slice header)
19 SW_DEPEND_SLICE_E	Dependent slice enable
18 SW_FILT_OVERRIDE_E	Filter override enable
17 SW_STRONG_SMOOTH_E	Strong smoothing enable
16-12	Filter beta offset (declared as div2)

Table continues on the next page...

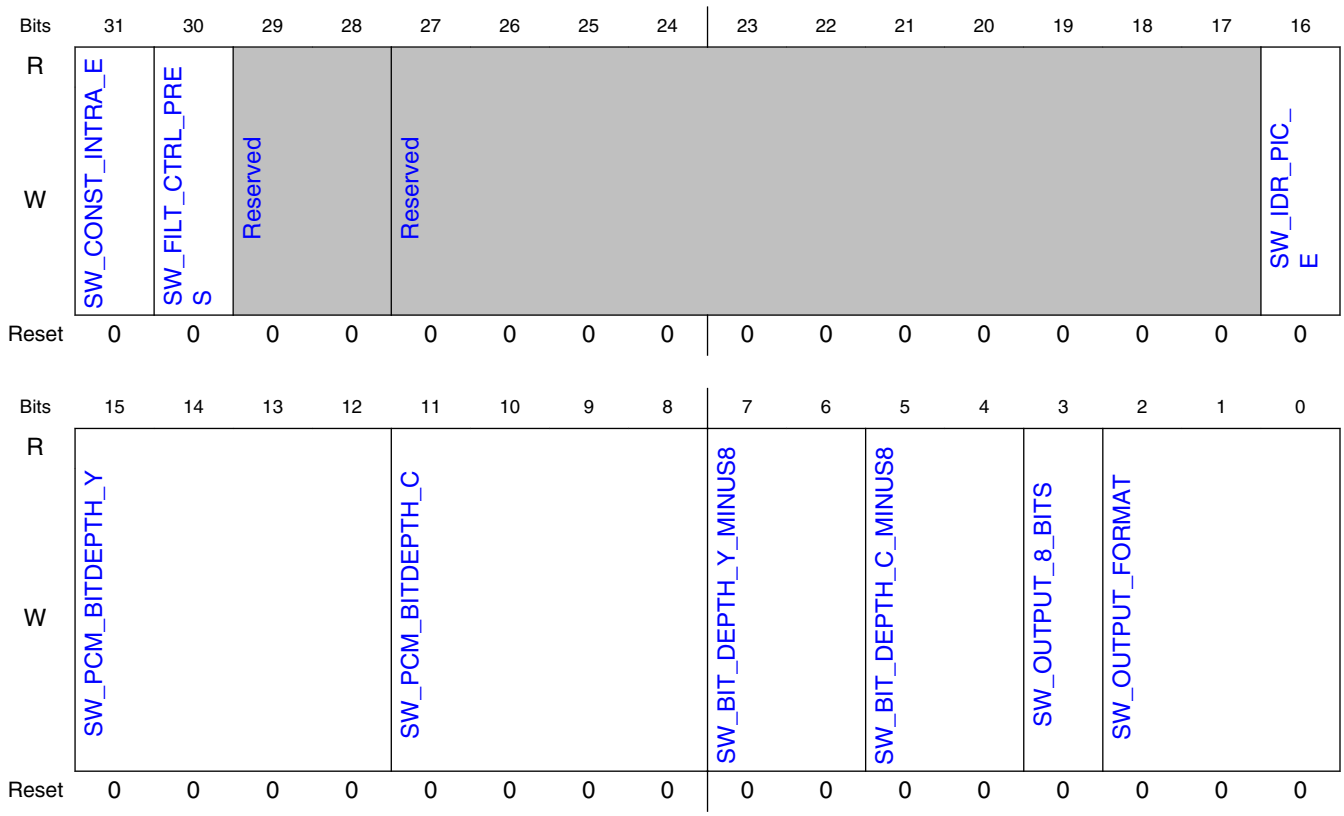
Field	Function
SW_FILT_OFFSET_BETA	
11-7 SW_FILT_OFFSET_TC	Filter tc offset (declared as div2)
6 SW_SLICE_HDR_EXT_E	Slice header extension enable. Reserved for future use
5-3 SW_SLICE_HDR_EBITS	Number of extra slice header bits (if enabled slice header extension)
2-0 —	Reserved.

14.2.5.1.10 Decoder control register 5 (SWREG8)

14.2.5.1.10.1 Offset

Register	Offset
SWREG8	20h

14.2.5.1.10.2 Diagram



14.2.5.1.10.3 Fields

Field	Function
31 SW_CONST_INTRA_E	constrained_intra_pred_flag equal to 1 specifies that intra prediction uses only neighbouring intra macroblocks in prediction. When equal to 0 also neighbouring inter macroblocks are used in intra prediction process.
30 SW_FILT_CTRL_PRES	deblocking_filter_control_present_flag indicates whether extra variables controlling characteristics of the deblocking filter are present in the slice header.
29-28 —	Reserved.
27-17 —	Reserved.
16 SW_IDR_PIC_E	IDR (instantaneous decoding refresh) picture flag.
15-12 SW_PCM_BITDEPTH_Y	Bit depth for PCM Y data
11-8	Bit depth for PCM C data

Table continues on the next page...

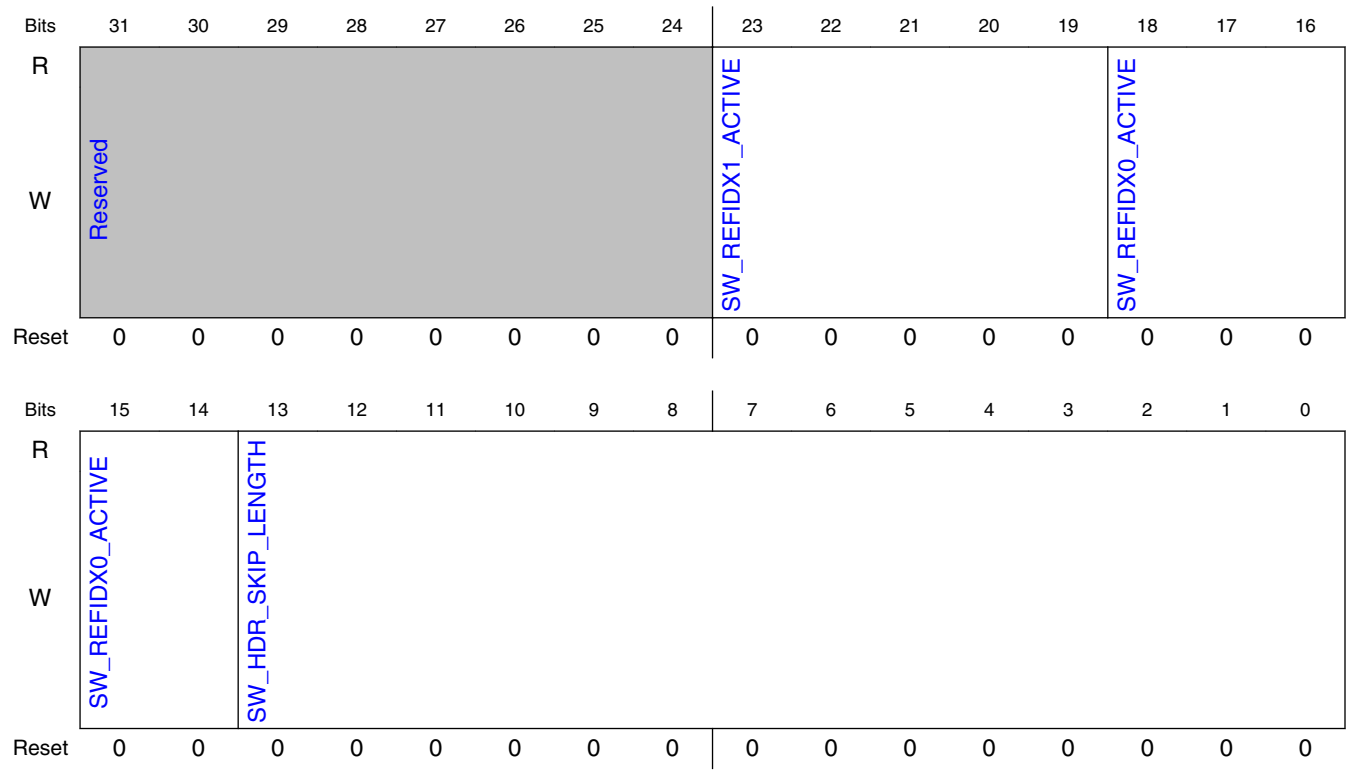
Field	Function
SW_PCM_BITDEPTH_C	
7-6 SW_BIT_DEPTH_Y_MINUS8	Bit depth of luma samples minus 8
5-4 SW_BIT_DEPTH_C_MINUS8	Bit depth of chroma samples minus 8
3 SW_OUTPUT_8_BITS	enable rasterscan output force to 8 bit(only for hevc main10 and vp9 10bit)
2-0 SW_OUTPUT_FORMAT	Raster scan and down scale output data format 000b - Each pixel in 10 bits when luma or chroma pixel bit depth is larger than 8; or 8 bits when both luma and chroma pixel bit depth are 8 bits. (default) 001b - Store in P010 format when luma or chroma pixel bit depth is larger than 8. 010b - A customized format: please refer to register SWREG23[6].

14.2.5.1.11 Decoder control register 6 (SWREG9)

14.2.5.1.11.1 Offset

Register	Offset
SWREG9	24h

14.2.5.1.11.2 Diagram



14.2.5.1.11.3 Fields

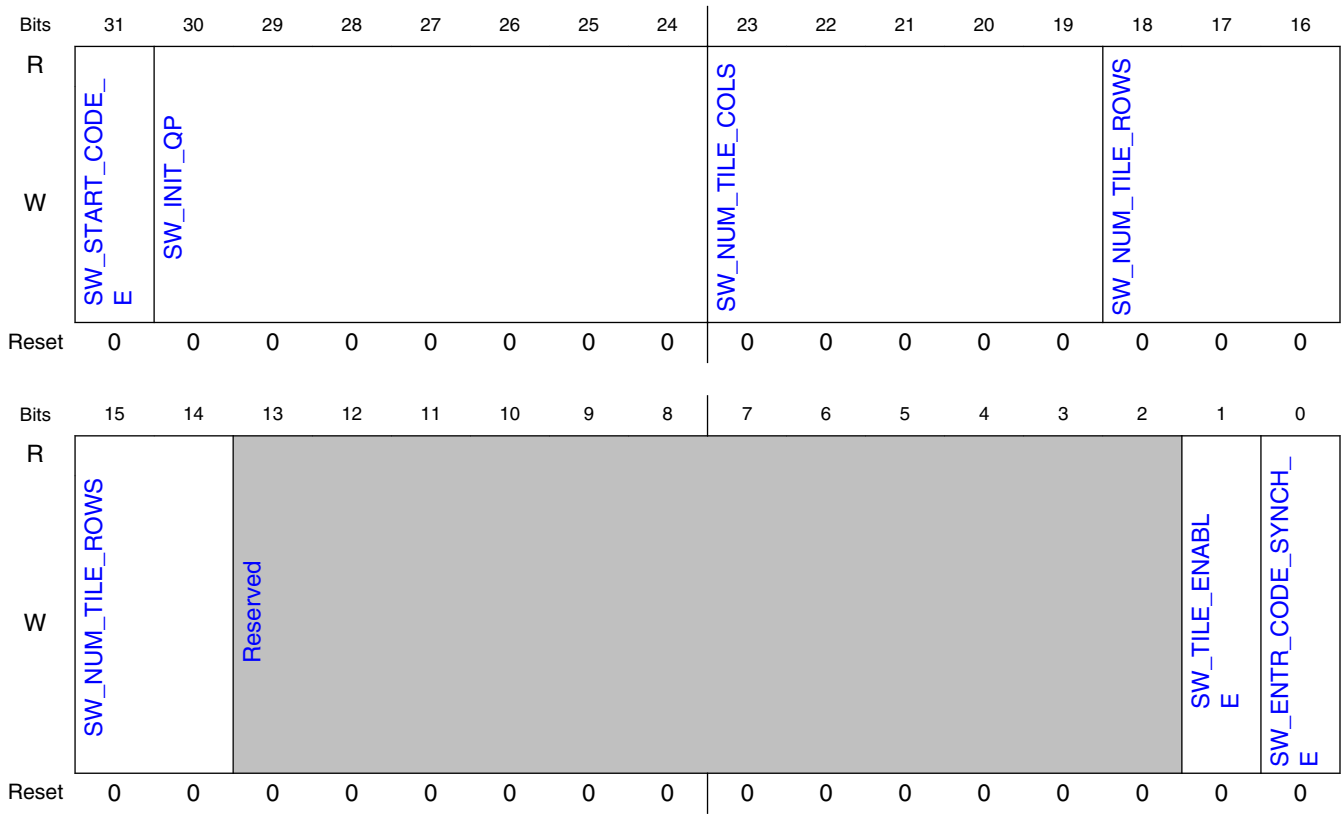
Field	Function
31-24 —	Reserved.
23-19 SW_REFIDX1_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks.
18-14 SW_REFIDX0_ACTIVE	Specifies the maximum reference index that can be used while decoding inter predicted macro blocks. This is same as in previous decoders (width increased with q bit)
13-0 SW_HDR_SKIP_LENGTH	Length of slice header skip length (bytes used by sw)

14.2.5.1.12 Decoder control register 7 (SWREG10)

14.2.5.1.12.1 Offset

Register	Offset
SWREG10	28h

14.2.5.1.12.2 Diagram



14.2.5.1.12.3 Fields

Field	Function
31 SW_START_CODE_E	Bit for indicating stream start code existence 0b - Stream does not contain start codes 1b - Stream contains start codes
30-24 SW_INIT_QP	Initial value for quantization parameter (picture quantizer).
23-19 SW_NUM_TILE_COLS	Number of tile columns in picture
18-14	Number of tile rows in picture

Table continues on the next page...

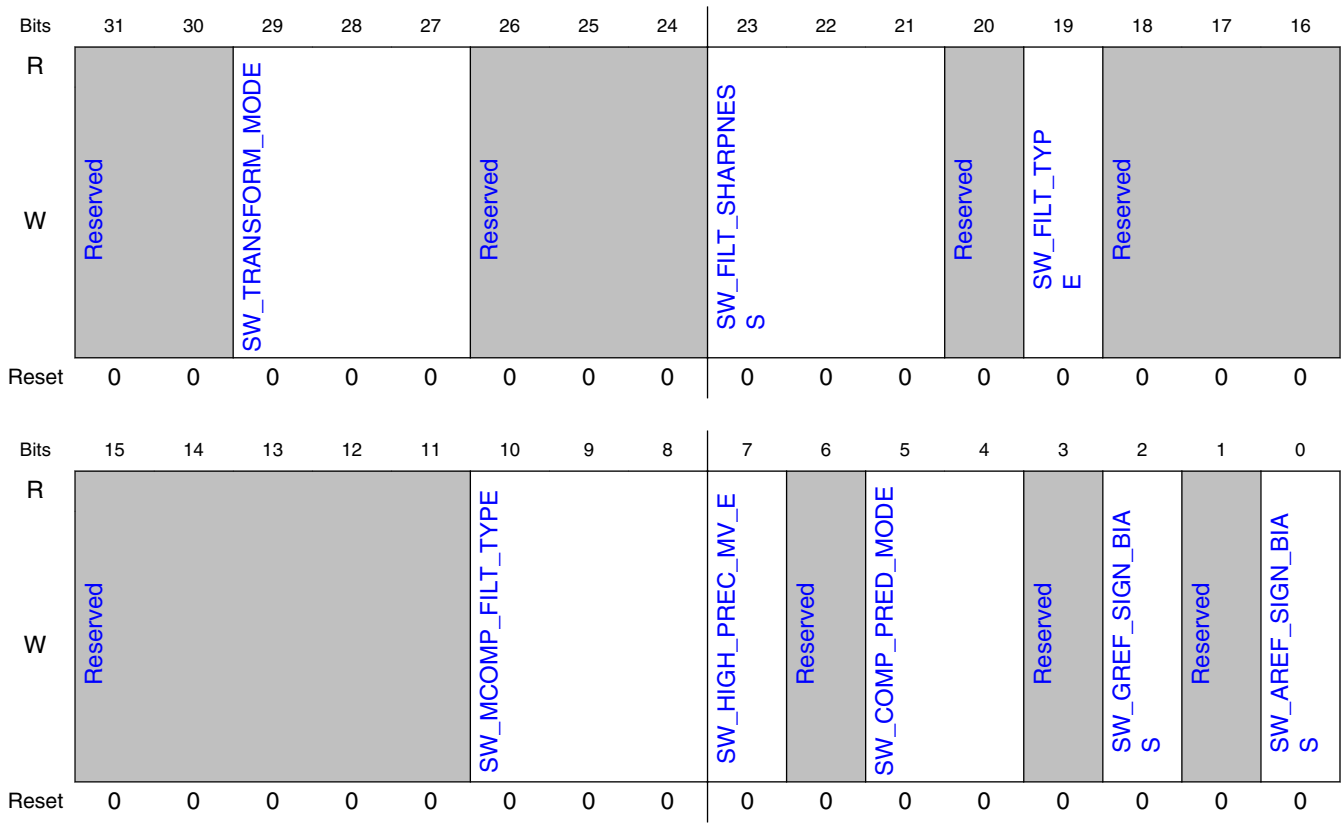
Field	Function
SW_NUM_TILE_ROWS	
13-2 —	Reserved.
1 SW_TILE_ENABLE	Tile enable
0 SW_ENTR_CODE_SYNCH_E	Entropy coding synchronization enable (Possible parallel cabac decoding)

14.2.5.1.13 Decoder control register 8 (SWREG11)

14.2.5.1.13.1 Offset

Register	Offset
SWREG11	2Ch

14.2.5.1.13.2 Diagram



14.2.5.1.13.3 Fields

Field	Function
31-30 —	Reserved.
29-27 SW_TRANSFORM_MODE	Transform modes 000b - 4x4 only 001b - Allow 8x8 010b - Allow 16x16 011b - Allow 32x32 100b - TX mode select
26-24 —	Reserved.
23-21 SW_FILT_SHARPNESS	Filter sharpness value
20 —	Reserved.
19 SW_FILT_TYPE	Filter Type

Table continues on the next page...

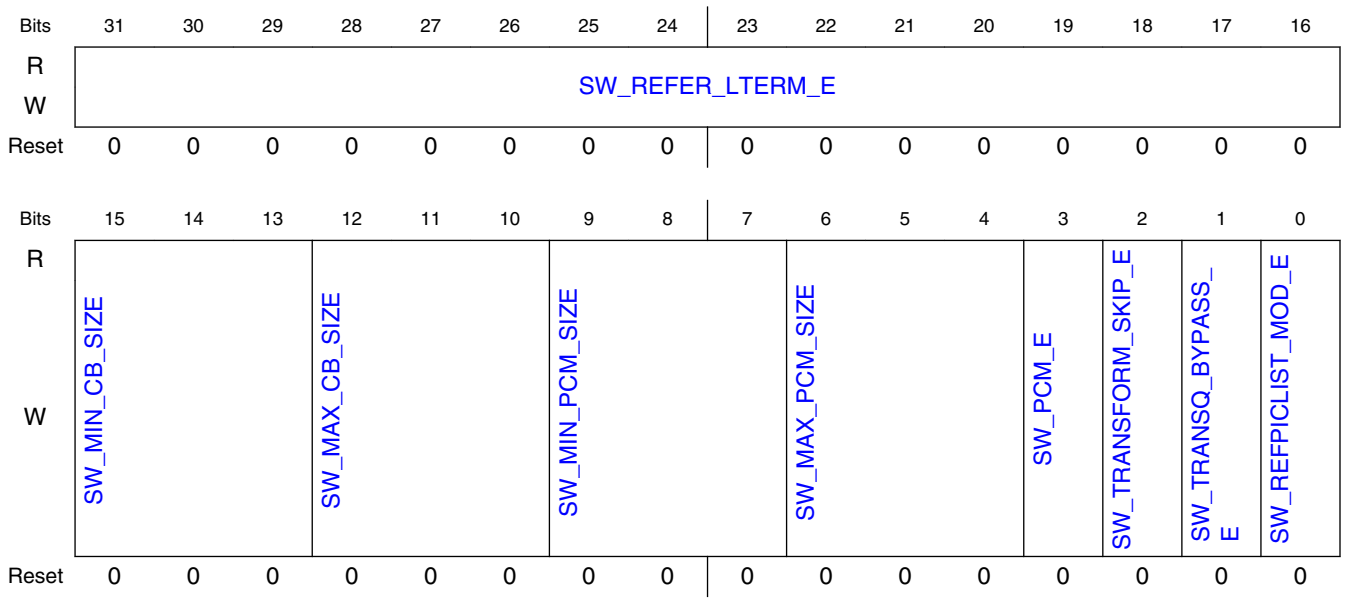
Field	Function
18-11 —	Reserved.
10-8 SW_MCOMP_FILT_TYPE	Inter prediction filter type to stream decoder 000b - Eight tap smooth 001b - Eight tap 010b - Eight tap sharp 011b - Bilinear 100b - Switchable
7 SW_HIGH_PREC_MV_E	High precision MV prediction enable
6 —	Reserved.
5-4 SW_COMP_PRED_MODE	Prediction Comp Type 00b - Single prediction only 01b - COMP prediction only 10b - Hybrid prediction
3 —	Reserved.
2 SW_GREF_SIGN_BIAS	Golden reference picture sign bias used for motion vector decoding
1 —	Reserved.
0 SW_AREF_SIGN_BIAS	Alternate reference picture sign bias used for motion vector decoding

14.2.5.1.14 Decoder control register 9 (SWREG12)

14.2.5.1.14.1 Offset

Register	Offset
SWREG12	30h

14.2.5.1.14.2 Diagram



14.2.5.1.14.3 Fields

Field	Function
31-16 SW_REFER_LTERM_E	Long term flag for reference picture index Definition: Bit 31 for picture index 0 Bit 30 for picture index 1 etc.
15-13 SW_MIN_CB_SIZE	CodedBlock min size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
12-10 SW_MAX_CB_SIZE	CodedBlock max size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
9-7 SW_MIN_PCM_SIZE	PCM min size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
6-4 SW_MAX_PCM_SIZE	PCM max size (2^N): 011b - 8 pix 100b - 16 pix 101b - 32 pix 110b - 64 pix
3 SW_PCM_E	IPCM MBs flag

Table continues on the next page...

Field	Function
2 SW_TRANSFORM_SKIP_E	Transform skipping flag
1 SW_TRANSQ_BYPASS_E	Transform bypass flag (lossless mode)
0 SW_REFPICLIST_MOD_E	Refpic list reordering flag

14.2.5.1.15 Decoder control register 10 (SWREG13)

14.2.5.1.15.1 Offset

Register	Offset
SWREG13	34h

14.2.5.1.15.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			DEC_CTRL_REG10_BF												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DEC_CTRL_REG10_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.15.3 Fields

Field	Function
31-29 —	Reserved.
28-0 DEC_CTRL_REG10_BF	For HEVC: [31:16] - Reserved - Not used [15:13] - sw_min_trb_size - Transform Block min size (2^N): 3 = 8 pix, 4 = 16 pix, 5 = 32 pix, 6 = 64 pix.

Field	Function
	[12:10] - sw_max_trb_size - Transform Block max size (2^N): 3 = 8 pix, 4 = 16 pix, 5 = 32 pix, 6 = 64 pix. [9:7] - sw_max_intra_hierdepth - Intra max hierarchy dept [6:4] - sw_max_inter_hierdepth - Inter max hierarchy depth [3:0] - sw_parallel_merge - Information about differential MV exist inside of coded block. For VP9: [31:29] - Reserved - Not used [28:23] - sw_qp_delta_y_dc - QP delta value for Y DC [22:17] - sw_qp_delta_ch_dc - QP delta value for C DC [16:11] - sw_qp_delta_ch_ac - QP delta value for C AC [10] - sw_last_sign_bias - Previous reference picture sign bias for motion vector decoding [9] - sw_lossless_e - Lossless transform enable [8:7] - sw_comp_pred_var_ref1 - Compaund prediction parameter to select correct sign bias from bin decoding [6:5] - sw_comp_pred_var_ref0 - Compaund prediction parameter to select correct sign bias from bin decoding [4:3] - sw_comp_pred_fixed_ref - Compaund prediction parameter to select last picture sign bias [2] - sw_segment_temp_upd_e - temporal segmentation update enable [1] - sw_segment_upd_e - segmentation update enable. If high new segmentation values are delivered in stream (and will be written out). If 0 the segmentation values are read from external memory [0] - sw_segment_e - segmentation enable

14.2.5.1.16 Initial ref pic list register (0-2) (SWREG14)

14.2.5.1.16.1 Offset

Register	Offset
SWREG14	38h

14.2.5.1.16.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_0_2_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_0_2_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.16.3 Fields

Field	Function
31-0 INIT_REF_PIC_0_2_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b2 - Initial reference picture list for backward picid 2 [24:20] - sw_init_rlist_f2 - Initial reference picture list for forward picid 2 [19:15] - sw_init_rlist_b1 - Initial reference picture list for backward picid 1 [14:10] - sw_init_rlist_f1 - Initial reference picture list for forward picid 1 [9:5] - sw_init_rlist_b0 - Initial reference picture list for backward picid 0 [4:0] - sw_init_rlist_f0 - Initial reference picture list for forward picid 0 For VP9: [31:24] - Reserved - Not used [23:18] - sw_filt_level - Frame filtering level [17:15] - sw_refpic_seg0 - Segment refer picture [14] - sw_skip_seg0 - Segment skip enable [13:8] - sw_filt_level_seg0 - Segment filter level [7:0] - sw_quant_seg0 - Segment quantization parameter

14.2.5.1.17 Initial ref pic list register (3-5) (SWREG15)

14.2.5.1.17.1 Offset

Register	Offset
SWREG15	3Ch

14.2.5.1.17.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_3_5_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_3_5_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.17.3 Fields

Field	Function
31-0	For HEVC:

Field	Function
INIT_REF_PIC_3_5_BF	[31:30] - Reserved - Not used [29:25] - sw_init_rlist_b5 - Initial reference picture list for backward picid 5 [24:20] - sw_init_rlist_f5 - Initial reference picture list for forward picid 5 [19:15] - sw_init_rlist_b4 - Initial reference picture list for backward picid 4 [14:10] - sw_init_rlist_f4 - Initial reference picture list for forward picid 4 [9:5] - sw_init_rlist_b3 - Initial reference picture list for backward picid 3 [4:0] - sw_init_rlist_f3 - Initial reference picture list for forward picid 3 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg1 - Segment refer picture [14] - sw_skip_seg1 - Segment skip enable [13:8] - sw_filt_level_seg1 - Segment filter level [7:0] - sw_quant_seg1 - Segment quantization parameter

14.2.5.1.18 Initial ref pic list register (6-8) (SWREG16)

14.2.5.1.18.1 Offset

Register	Offset
SWREG16	40h

14.2.5.1.18.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_6_8_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_6_8_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.18.3 Fields

Field	Function
31-0 INIT_REF_PIC_6_8_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b8 - Initial reference picture list for backward picid 8 [24:20] - sw_init_rlist_f8 - Initial reference picture list for forward picid 8 [19:15] - sw_init_rlist_b7 - Initial reference picture list for backward picid 7 [14:10] - sw_init_rlist_f7 - Initial reference picture list for forward picid 7

Field	Function
	[9:5] - sw_init_rlist_b6 - Initial reference picture list for backward picid 6 [4:0] - sw_init_rlist_f6 - Initial reference picture list for forward picid 6 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg2 - Segment refer picture [14] - sw_skip_seg2 - Segment skip enable [13:8] - sw_filt_level_seg2 - Segment filter level [7:0] - sw_quant_seg2 - Segment quantization parameter

14.2.5.1.19 Initial ref pic list register (9-11) (SWREG17)

14.2.5.1.19.1 Offset

Register	Offset
SWREG17	44h

14.2.5.1.19.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_9_11_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_9_11_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.19.3 Fields

Field	Function
31-0 INIT_REF_PIC_9_11_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b11 - Initial reference picture list for backward picid 11 [24:20] - sw_init_rlist_f11 - Initial reference picture list for forward picid 11 [19:15] - sw_init_rlist_b10 - Initial reference picture list for backward picid 10 [14:10] - sw_init_rlist_f10 - Initial reference picture list for forward picid 10 [9:5] - sw_init_rlist_b9 - Initial reference picture list for backward picid 9 [4:0] - sw_init_rlist_f9 - Initial reference picture list for forward picid 9 For VP9: [31:18] - Reserved - Not used

Field	Function
	[17:15] - sw_refpic_seg3 - Segment refer picture [14] - sw_skip_seg3 - Segment skip enable [13:8] - sw_filt_level_seg3 - Segment filter level [7:0] - sw_quant_seg3 - Segment quantization parameter

14.2.5.1.20 Initial ref pic list register (12-14) (SWREG18)

14.2.5.1.20.1 Offset

Register	Offset
SWREG18	48h

14.2.5.1.20.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_12_14_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_12_14_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.20.3 Fields

Field	Function
31-0 INIT_REF_PIC_12_14_BF	For HEVC: [31:30] - Reserved - Not used [29:25] - sw_init_rlist_b14 - Initial reference picture list for backward picid 14 [24:20] - sw_init_rlist_f14 - Initial reference picture list for forward picid 14 [19:15] - sw_init_rlist_b13 - Initial reference picture list for backward picid 13 [14:10] - sw_init_rlist_f13 - Initial reference picture list for forward picid 13 [9:5] - sw_init_rlist_b12 - Initial reference picture list for backward picid 12 [4:0] - sw_init_rlist_f12 - Initial reference picture list for forward picid 12 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg4 - Segment refer picture [14] - sw_skip_seg4 - Segment skip enable [13:8] - sw_filt_level_seg4 - Segment filter level [7:0] - sw_quant_seg4 - Segment quantization parameter

14.2.5.1.21 Initial ref pic list register (15 and P 0-3) (SWREG19)

14.2.5.1.21.1 Offset

Register	Offset
SWREG19	4Ch

14.2.5.1.21.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INIT_REF_PIC_15_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INIT_REF_PIC_15_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.21.3 Fields

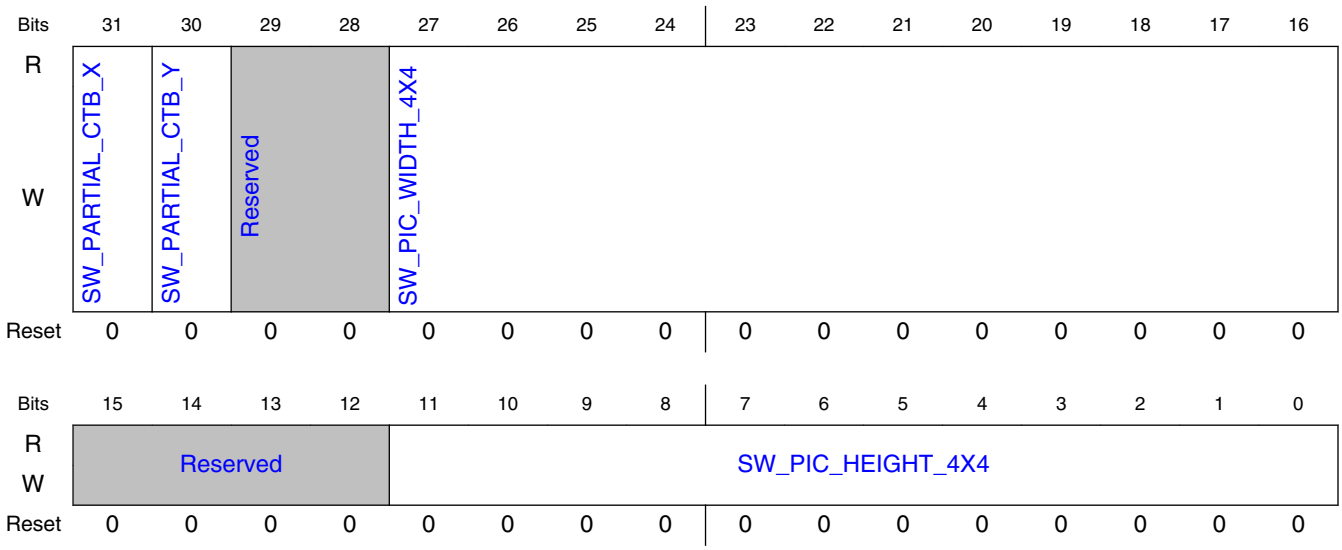
Field	Function
31-0 INIT_REF_PIC_15_BF	For HEVC: [31:10] - Reserved - Not used [9:5] - sw_init_rlist_b15 - Initial reference picture list for backward picid 15 [4:0] - sw_init_rlist_f15 - Initial reference picture list for forward picid 15 For VP9: [31:18] - Reserved - Not used [17:15] - sw_refpic_seg5 - Segment refer picture [14] - sw_skip_seg5 - Segment skip enable [13:8] - sw_filt_level_seg5 - Segment filter level [7:0] - sw_quant_seg5 - Segment quantization parameter

14.2.5.1.22 Decoder control register 11 (SWREG20)

14.2.5.1.22.1 Offset

Register	Offset
SWREG20	50h

14.2.5.1.22.2 Diagram



14.2.5.1.22.3 Fields

Field	Function
31 SW_PARTIAL_CTB_X	Picture width not multiple of CTB size
30 SW_PARTIAL_CTB_Y	Picture height not multiple of CTB size
29-28 —	Reserved.
27-16 SW_PIC_WIDTH_4X4	Current picture width in 4x4 blocks (Needed to reduce overlapping HW conditions in various blocks)
15-12 —	Reserved.
11-0 SW_PIC_HEIGHT_4X4	Current picture height in 4x4 blocks (Needed to reduce overlapping HW conditions in various blocks)

14.2.5.1.23 Not used (SWREG21)

14.2.5.1.23.1 Offset

Register	Offset
SWREG21	54h

14.2.5.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.23.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.24 Not used (SWREG22)

14.2.5.1.24.1 Offset

Register	Offset
SWREG22	58h

14.2.5.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.24.3 Fields

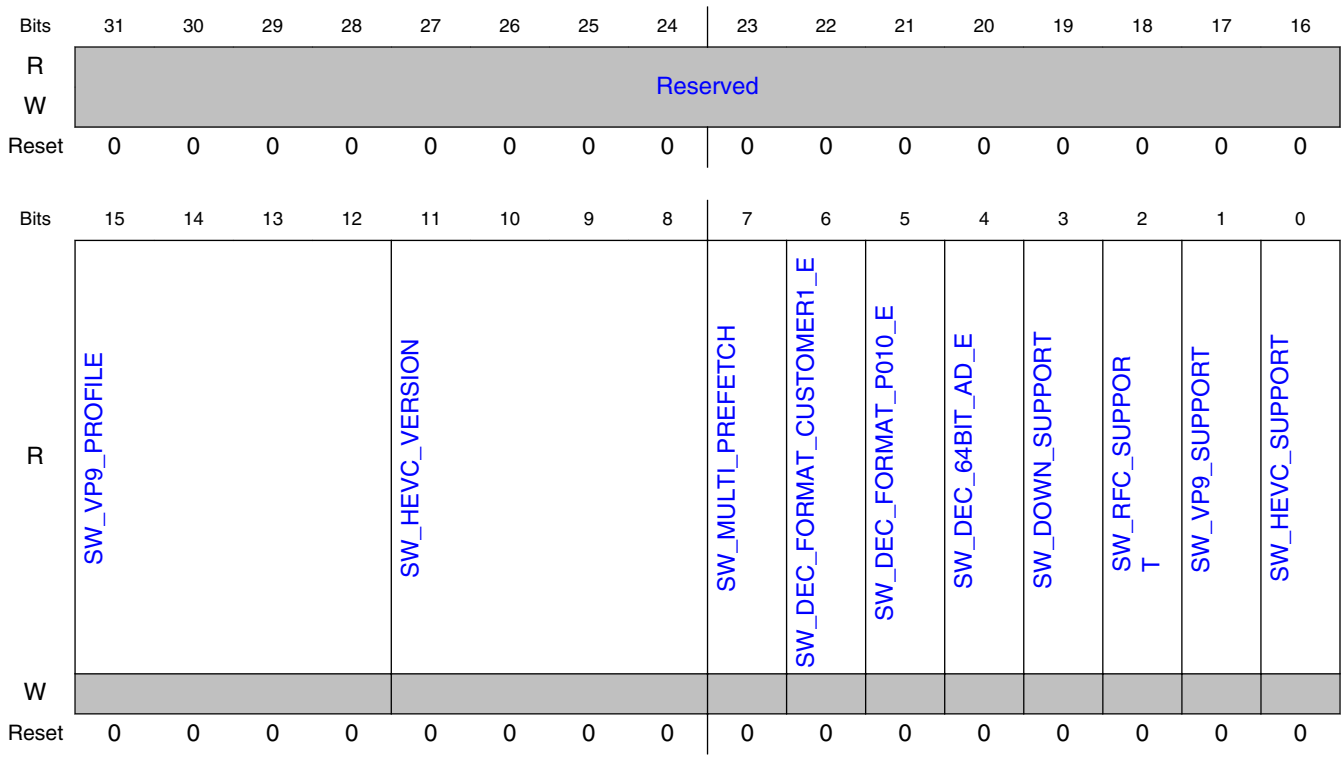
Field	Function
31-0 —	Reserved.

14.2.5.1.25 Decoder configure status register (SWREG23)

14.2.5.1.25.1 Offset

Register	Offset
SWREG23	5Ch

14.2.5.1.25.2 Diagram



14.2.5.1.25.3 Fields

Field	Function
31-16 —	Reserved.
15-12 SW_VP9_PROFILE	VP9 version 0000b - vp9 profile 0 0001b - vp9 profile 2 - 10bits
11-8 SW_HEVC_VERSION	HEVC version 0000b - main8 0001b - main10
7 SW_MULTI_PREFETCH	Multi-Reference Blocks Prefetch 0b - Not supported 1b - Supported
6 SW_DEC_FORMAT_CUSTOMER1_E	Customized output format support Demonstrated as following - from MSB->LSB in a 128-bit burst. • 10-bit pixel • 128-bit burst0: Y0|Y1|Y2|Y3|...|Y11|Y12[9:2] • 128-bit burst1: Y12[1:0]|Y13|Y14|Y15|...|Y24|Y25[9:4] • 128-bit burst2: Y25[3:0]|Y26|Y27|Y28|...|Y37|Y38[9:6]

Table continues on the next page...

Field	Function
	• 128-bit burst3: Y38[5:0]|Y39|Y40|Y41|...|Y50|Y51[9:8] • 128-bit burst4: Y51[7:0]|Y52|Y53|Y54|...|Y62|Y63 • 8-bit pixel • 128-bit burst: Y0|Y1|Y2|...|Y15 • 128-bit burst: Y16|Y17|Y18|...|Y31 0b - Not supported 1b - Supported
5 SW_DEC_FORMAT_P010_E	P010 output format support 0b - Not supported 1b - Supported
4 SW_DEC_64BIT_AD_E	64 bit addressing of master interface support 0b - Not supported (32 bit addressing) 1b - Supported
3 SW_DOWN_SUPPORT	Downscale support 0b - Do not support downscale 1b - Support downscale
2 SW_RFC_SUPPORT	RFC support 0b - Do not support RFC 1b - Support RFC
1 SW_VP9_SUPPORT	VP9 support 0b - Do not support VP9 1b - Support VP9
0 SW_HEVC_SUPPORT	HEVC support 0b - Do not support HEVC 1b - Support HEVC

14.2.5.1.26 Not used (SWREG24)

14.2.5.1.26.1 Offset

Register	Offset
SWREG24	60h

14.2.5.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.26.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.27 Not used (SWREG25)

14.2.5.1.27.1 Offset

Register	Offset
SWREG25	64h

14.2.5.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.27.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.28 Not used (SWREG26)

14.2.5.1.28.1 Offset

Register	Offset
SWREG26	68h

14.2.5.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.28.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.29 Not used (SWREG27)

14.2.5.1.29.1 Offset

Register	Offset
SWREG27	6Ch

14.2.5.1.29.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.29.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.30 Not used (SWREG28)

14.2.5.1.30.1 Offset

Register	Offset
SWREG28	70h

14.2.5.1.30.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.30.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.31 Not used (SWREG29)

14.2.5.1.31.1 Offset

Register	Offset
SWREG29	74h

14.2.5.1.31.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.31.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.32 Not used (SWREG30)

14.2.5.1.32.1 Offset

Register	Offset
SWREG30	78h

14.2.5.1.32.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.32.3 Fields

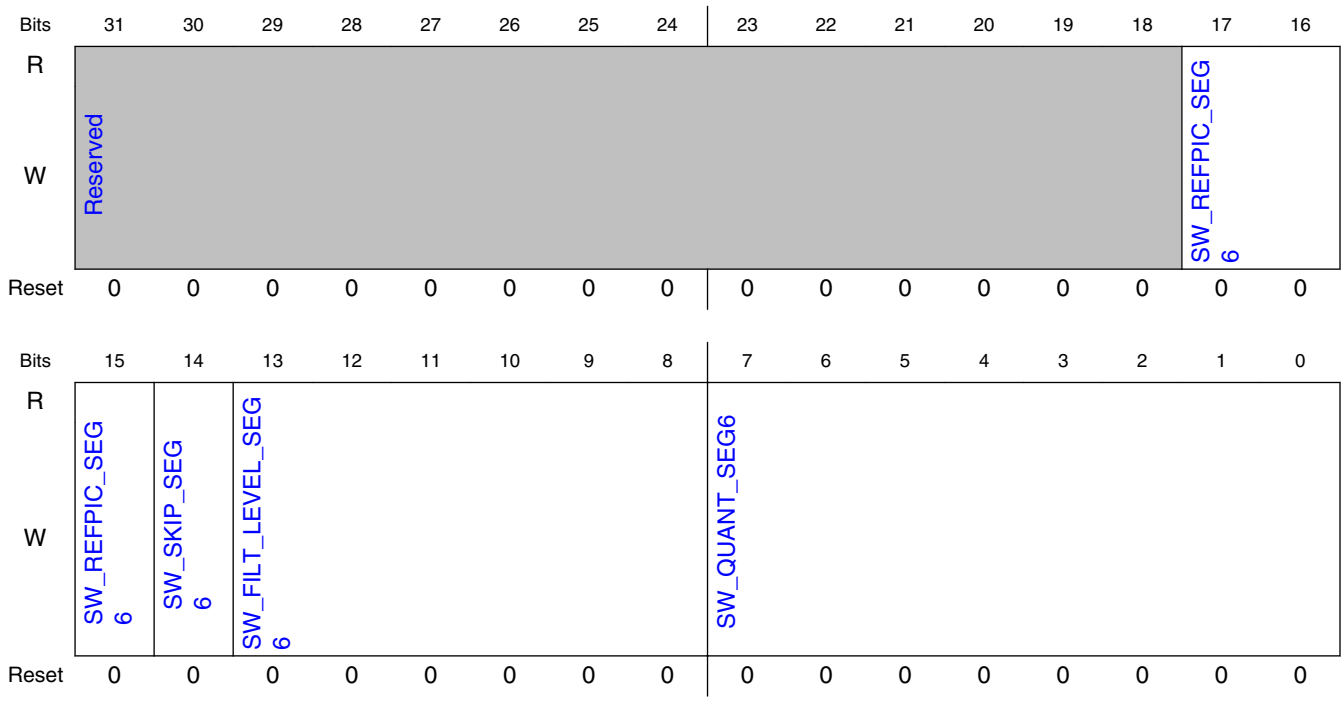
Field	Function
31-0 —	Reserved.

14.2.5.1.33 VP9 segmentation values (SWREG31)

14.2.5.1.33.1 Offset

Register	Offset
SWREG31	7Ch

14.2.5.1.33.2 Diagram



14.2.5.1.33.3 Fields

Field	Function
31-18 —	Reserved.
17-15 SW_REFPIC_SEG6	Segment refer picture
14 SW_SKIP_SEG6	Segment skip enable
13-8 SW_FILT_LEVEL_SEG6	Segment filter level
7-0	Segment quantization parameter

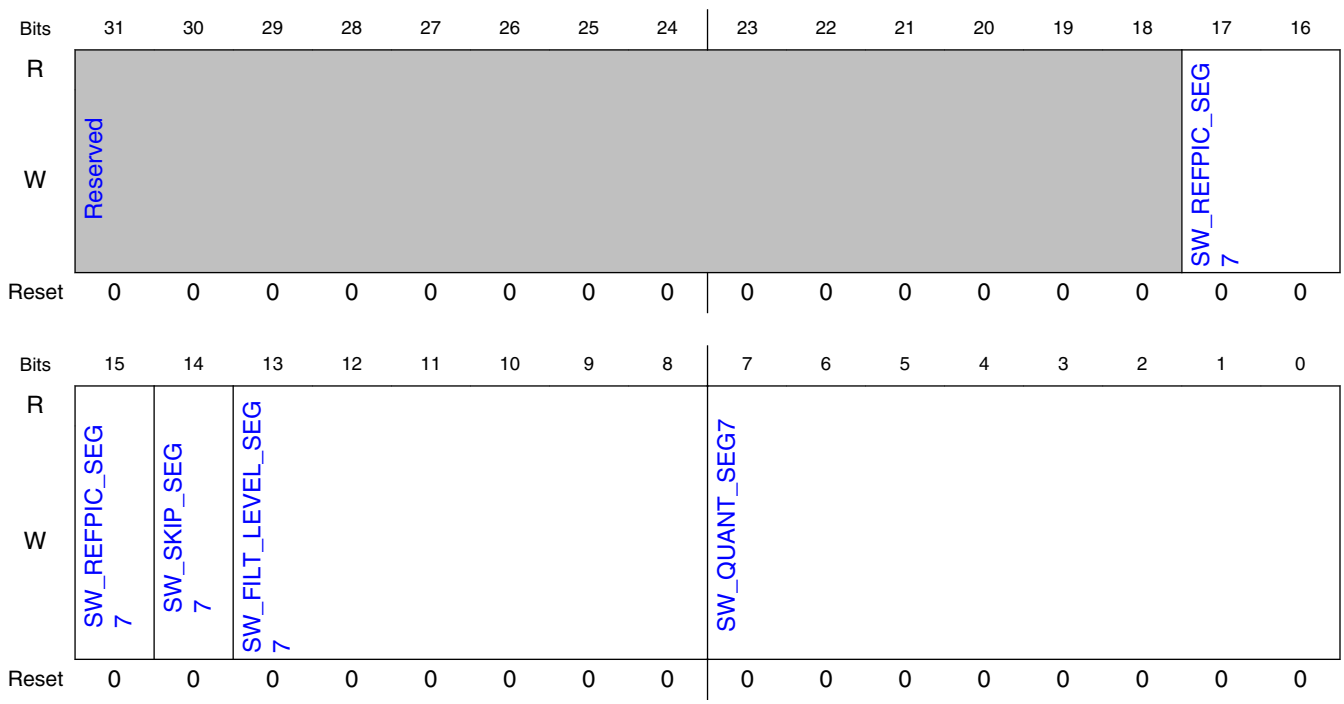
Field	Function
SW_QUANT_SEG6	

14.2.5.1.34 VP9 segmentation values (SWREG32)

14.2.5.1.34.1 Offset

Register	Offset
SWREG32	80h

14.2.5.1.34.2 Diagram



14.2.5.1.34.3 Fields

Field	Function
31-18 —	Reserved.
17-15	Segment refer picture

Table continues on the next page...

Field	Function
SW_REFPIC_SEG7	
14 SW_SKIP_SEG7	Segment skip enable
13-8 SW_FILT_LEVEL_SEG7	Segment filter level
7-0 SW_QUANT_SEG7	Segment quantization parameter

14.2.5.1.35 VP9 reference picture scaling register 0 (SWREG33)

14.2.5.1.35.1 Offset

Register	Offset
SWREG33	84h

14.2.5.1.35.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_LREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_LREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.35.3 Fields

Field	Function
31-16 SW_LREF_WIDTH	Accurate width of last (previous) reference picture in pixels
15-0	Accurate height of last (previous) reference picture in pixels

Field	Function
SW_LREF_HEIGHT	

14.2.5.1.36 VP9 reference picture scaling register 1 (SWREG34)

14.2.5.1.36.1 Offset

Register	Offset
SWREG34	88h

14.2.5.1.36.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_GREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_GREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.36.3 Fields

Field	Function
31-16 SW_GREF_WIDTH	Accurate width of golden reference picture in pixels
15-0 SW_GREF_HEIGHT	Accurate height of golden reference picture in pixels

14.2.5.1.37 VP9 reference picture scaling register 2 (SWREG35)

14.2.5.1.37.1 Offset

Register	Offset
SWREG35	8Ch

14.2.5.1.37.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_AREF_WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_AREF_HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.37.3 Fields

Field	Function
31-16 SW_AREF_WIDTH	Accurate width of alternate reference picture in pixels
15-0 SW_AREF_HEIGHT	Accurate height of alternate reference picture in pixels

14.2.5.1.38 VP9 reference picture scaling register 3 (SWREG36)

14.2.5.1.38.1 Offset

Register	Offset
SWREG36	90h

14.2.5.1.38.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_LREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_LREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.38.3 Fields

Field	Function
31-16 SW_LREF_HOR_SCALE	Horizontal scaling factor for last (previous) reference picture
15-0 SW_LREF_VER_SCALE	Vertical scaling factor for last (previous) reference picture

14.2.5.1.39 VP9 reference picture scaling register 4 (SWREG37)

14.2.5.1.39.1 Offset

Register	Offset
SWREG37	94h

14.2.5.1.39.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_GREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_GREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.39.3 Fields

Field	Function
31-16 SW_GREF_HOR_SCALE	Horizontal scaling factor for golden reference picture
15-0 SW_GREF_VER_SCALE	Vertical scaling factor for golden reference picture

14.2.5.1.40 VP9 reference picture scaling register 5 (SWREG38)

14.2.5.1.40.1 Offset

Register	Offset
SWREG38	98h

14.2.5.1.40.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_AREF_HOR_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_AREF_VER_SCALE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.40.3 Fields

Field	Function
31-16 SW_AREF_HOR_SCALE	Horizontal scaling factor for alternate reference picture
15-0 SW_AREF_VER_SCALE	Vertical scaling factor for alternate reference picture

14.2.5.1.41 Not used (SWREG39)

14.2.5.1.41.1 Offset

Register	Offset
SWREG39	9Ch

14.2.5.1.41.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.41.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.42 Not used (SWREG40)

14.2.5.1.42.1 Offset

Register	Offset
SWREG40	A0h

14.2.5.1.42.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.42.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.43 Not used (SWREG41)

14.2.5.1.43.1 Offset

Register	Offset
SWREG41	A4h

14.2.5.1.43.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.43.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.44 Not used (SWREG42)

14.2.5.1.44.1 Offset

Register	Offset
SWREG42	A8h

14.2.5.1.44.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.44.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.45 Not used (SWREG43)

14.2.5.1.45.1 Offset

Register	Offset
SWREG43	ACh

14.2.5.1.45.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.45.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.46 Not used (SWREG44)

14.2.5.1.46.1 Offset

Register	Offset
SWREG44	B0h

14.2.5.1.46.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.46.3 Fields

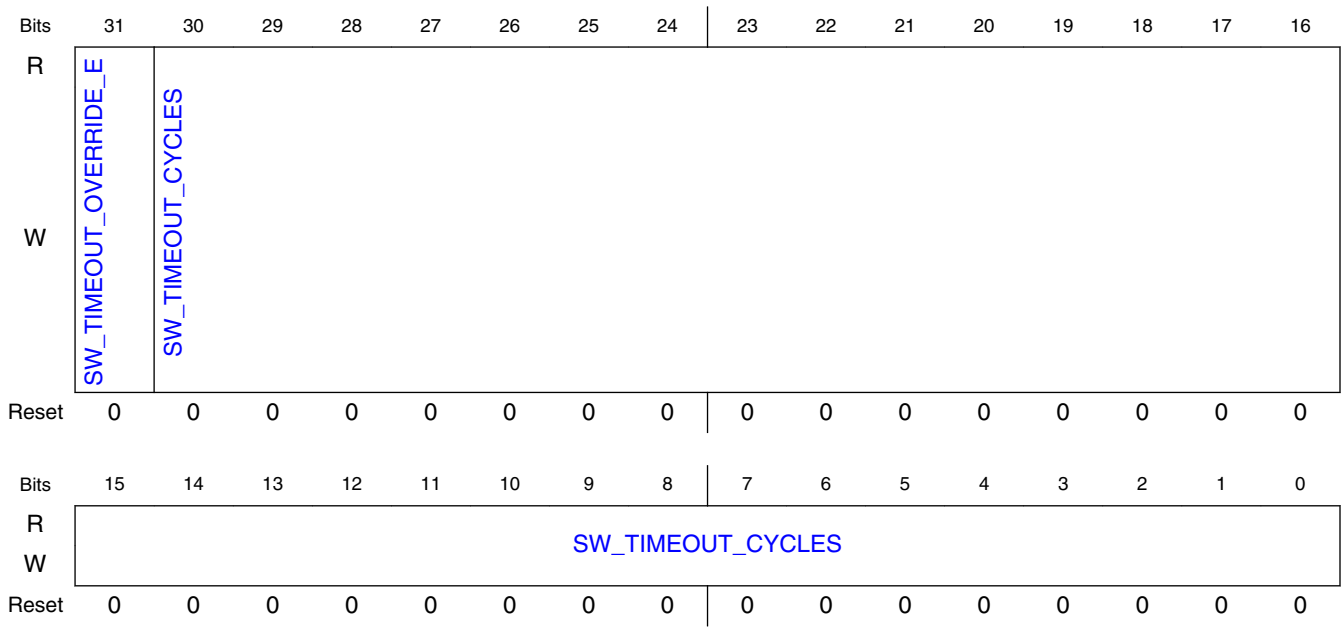
Field	Function
31-0 —	Reserved.

14.2.5.1.47 Timeout control register (SWREG45)

14.2.5.1.47.1 Offset

Register	Offset
SWREG45	B4h

14.2.5.1.47.2 Diagram



14.2.5.1.47.3 Fields

Field	Function
31 SW_TIMEOUT_OVERRIDE_E	Enable for SW controlled timeout. If enabled the sw_timeout_cycles is used to detect HW timeout instead of hard coded HW value
30-0 SW_TIMEOUT_CYCLES	Amount of clock cycles to trigger timeout interrupt if no external master activity acknowledged. Used if sw_timeout_override_e is set

14.2.5.1.48 Picture order count from current pictures for index 0-3 (SWREG46)

14.2.5.1.48.1 Offset

Register	Offset
SWREG46	B8h

14.2.5.1.48.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PIC_ORD_0_3_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIC_ORD_0_3_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.48.3 Fields

Field	Function
31-0 PIC_ORD_0_3_BF	For HEVC: [31:24] - sw_cur_poc_00 - Picture order count from current picture 0 [23:16] - sw_cur_poc_01 - Picture order count from current picture 1 [15:8] - sw_cur_poc_02 - Picture order count from current picture 2 [7:0] - sw_cur_poc_03 - Picture order count from current picture 3 For VP9: [31] - Reserved - Not used [30:24] - sw_filt_ref_adj_0 - Filter level intra adjustment [23] - Reserved - Not used [22:16] - sw_filt_ref_adj_1 - Filter level last ref pic adjustment [15] - Reserved - Not used [14:8] - sw_filt_ref_adj_2 - Filter level golden pic adjustment [7] - Reserved - Not used [6:0] - sw_filt_ref_adj_3 - Filter level alt ref pic adjustment

14.2.5.1.49 Picture order count from current pictures for index 4-7 (SWREG47)

14.2.5.1.49.1 Offset

Register	Offset
SWREG47	BCh

14.2.5.1.49.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PIC_ORD_4_7_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PIC_ORD_4_7_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.49.3 Fields

Field	Function
31-0 PIC_ORD_4_7_BF	For HEVC: [31:24] - sw_cur_poc_04 - Picture order count from current picture 4 [23:16] - sw_cur_poc_05 - Picture order count from current picture 5 [15:8] - sw_cur_poc_06 - Picture order count from current picture 6 [7:0] - sw_cur_poc_07 - Picture order count from current picture 7 For VP9: [31] - Reserved - Not used [30:24] - sw_filt_mb_adj_0 - Filter level ZERO mv adjustment [23] - Reserved - Not used [22:16] - sw_filt_mb_adj_1 - Filter level adjustment [15] - Reserved - Not used [14:8] - sw_filt_mb_adj_2 - Filter level adjustment (Not used) [7] - Reserved - Not used [6:0] - sw_filt_mb_adj_3 - Filter level adjustment (Not used)

14.2.5.1.50 Picture order count from current pictures for index 8-11 (SWREG48)

14.2.5.1.50.1 Offset

Register	Offset
SWREG48	C0h

14.2.5.1.50.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CUR_POC_08								SW_CUR_POC_09							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CUR_POC_10								SW_CUR_POC_11							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.50.3 Fields

Field	Function
31-24 SW_CUR_POC_08	Picture order count from current picture 8
23-16 SW_CUR_POC_09	Picture order count from current picture 9
15-8 SW_CUR_POC_10	Picture order count from current picture 10
7-0 SW_CUR_POC_11	Picture order count from current picture 11

14.2.5.1.51 Picture order count from current pictures for index 12-15 (SWREG49)

14.2.5.1.51.1 Offset

Register	Offset
SWREG49	C4h

14.2.5.1.51.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CUR_POC_12								SW_CUR_POC_13							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CUR_POC_14								SW_CUR_POC_15							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.51.3 Fields

Field	Function
31-24 SW_CUR_POC_12	Picture order count from current picture 12
23-16 SW_CUR_POC_13	Picture order count from current picture 13
15-8 SW_CUR_POC_14	Picture order count from current picture 14
7-0 SW_CUR_POC_15	Picture order count from current picture 15

14.2.5.1.52 Synthesis configuration register decoder 0 (read only) (SWREG50)

14.2.5.1.52.1 Offset

Register	Offset
SWREG50	C8h

14.2.5.1.52.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved					SW_DEC_MAX_OWIDTH										
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

14.2.5.1.52.3 Fields

Field	Function
31-11 —	Reserved.
10-0 SW_DEC_MAX_OWIDTH	Max configured decoder video resolution that can be decoded. Informed as width of the picture in pixels

14.2.5.1.53 Reference picture buffer control register (SWREG51)

14.2.5.1.53.1 Offset

Register	Offset
SWREG51	CCh

14.2.5.1.53.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.53.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.54 Reference picture buffer information register 1 (read only) (SWREG52)

14.2.5.1.54.1 Offset

Register	Offset
SWREG52	D0h

14.2.5.1.54.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.54.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.55 Reference picture buffer information register 2 (read only) (SWREG53)

14.2.5.1.55.1 Offset

Register	Offset
SWREG53	D4h

14.2.5.1.55.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.55.3 Fields

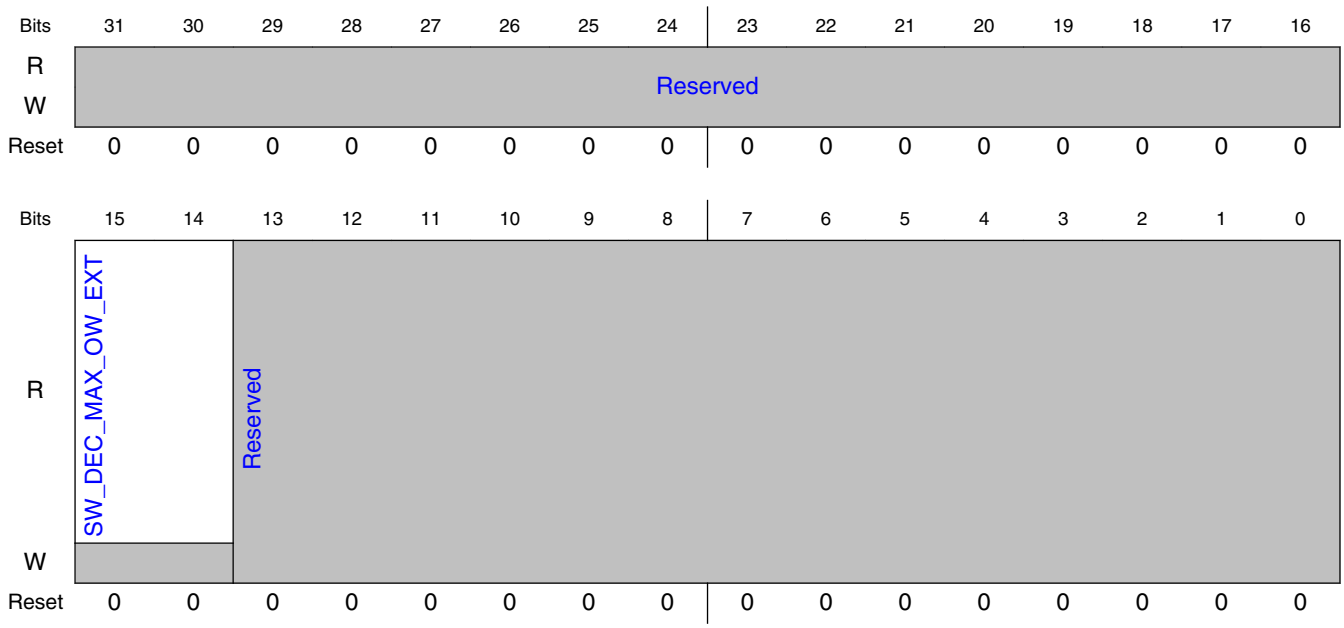
Field	Function
31-0 —	Reserved.

14.2.5.1.56 Synthesis configuration register decoder 1 (read only) (SWREG54)

14.2.5.1.56.1 Offset

Register	Offset
SWREG54	D8h

14.2.5.1.56.2 Diagram



14.2.5.1.56.3 Fields

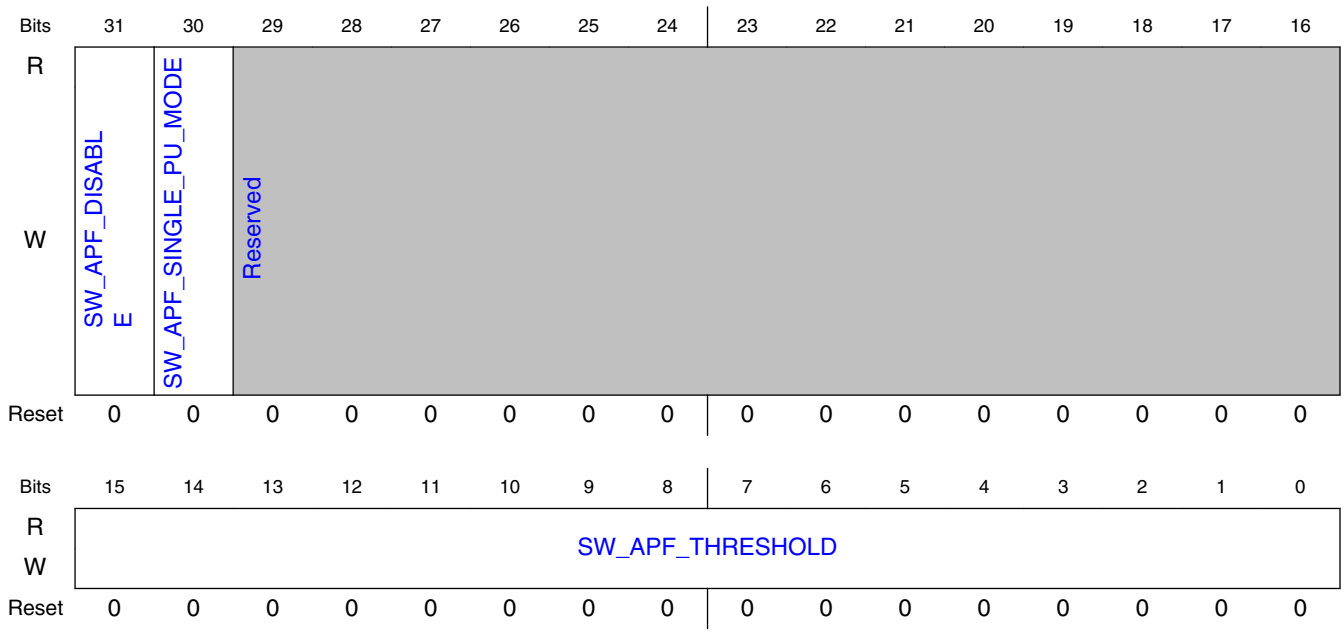
Field	Function
31-16 —	Reserved.
15-14 SW_DEC_MAX_OW_EXT	Max configured decoder video resolution that can be decoded. This is the MSB part of the configuration signal
13-0 —	Reserved.

14.2.5.1.57 Advanced prefetch control register (SWREG55)

14.2.5.1.57.1 Offset

Register	Offset
SWREG55	DCh

14.2.5.1.57.2 Diagram



14.2.5.1.57.3 Fields

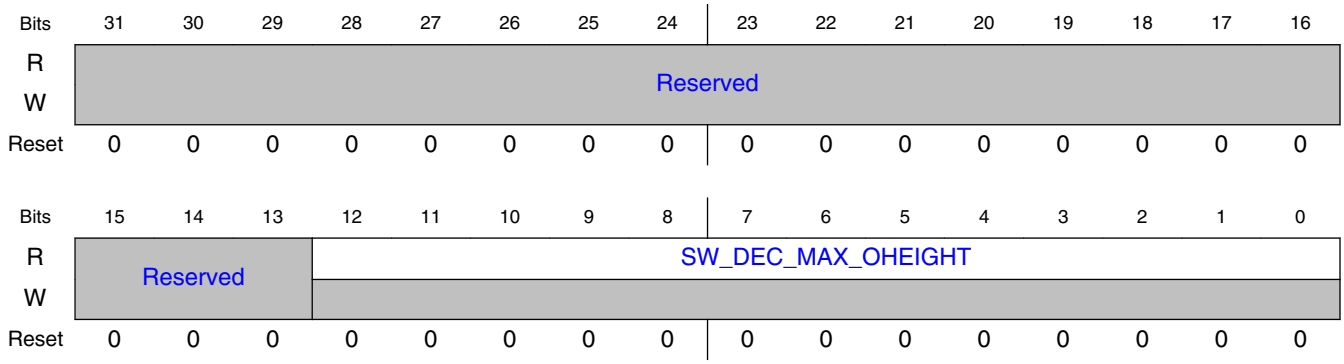
Field	Function
31 SW_APF_DISABLE	Advanced prefetch disable. If hight each partition is read separately
30 SW_APF_SINGLE_PU_MODE	APF amount of buffered Pus: can be restricted to buffer one PU at a time
29-16 —	Reserved.
15-0 SW_APF_THRESHOLD	Advanced prefetch threshold. If current buffered unit exceeds the threshold the advanced mode is not used. Value 0 disables threshold usage and advanced prefetch usage is restricted by internal memory limitation only

14.2.5.1.58 Synthesis configuration register decoder 2 (read only) (SWREG56)

14.2.5.1.58.1 Offset

Register	Offset
SWREG56	E0h

14.2.5.1.58.2 Diagram



14.2.5.1.58.3 Fields

Field	Function
31-13 —	Reserved.
12-0 SW_DEC_MAX_OHEIGHT	Max supported picture height in pixels

14.2.5.1.59 Decoder fuse register (read only) (SWREG57)

14.2.5.1.59.1 Offset

Register	Offset
SWREG57	E4h

14.2.5.1.59.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.59.3 Fields

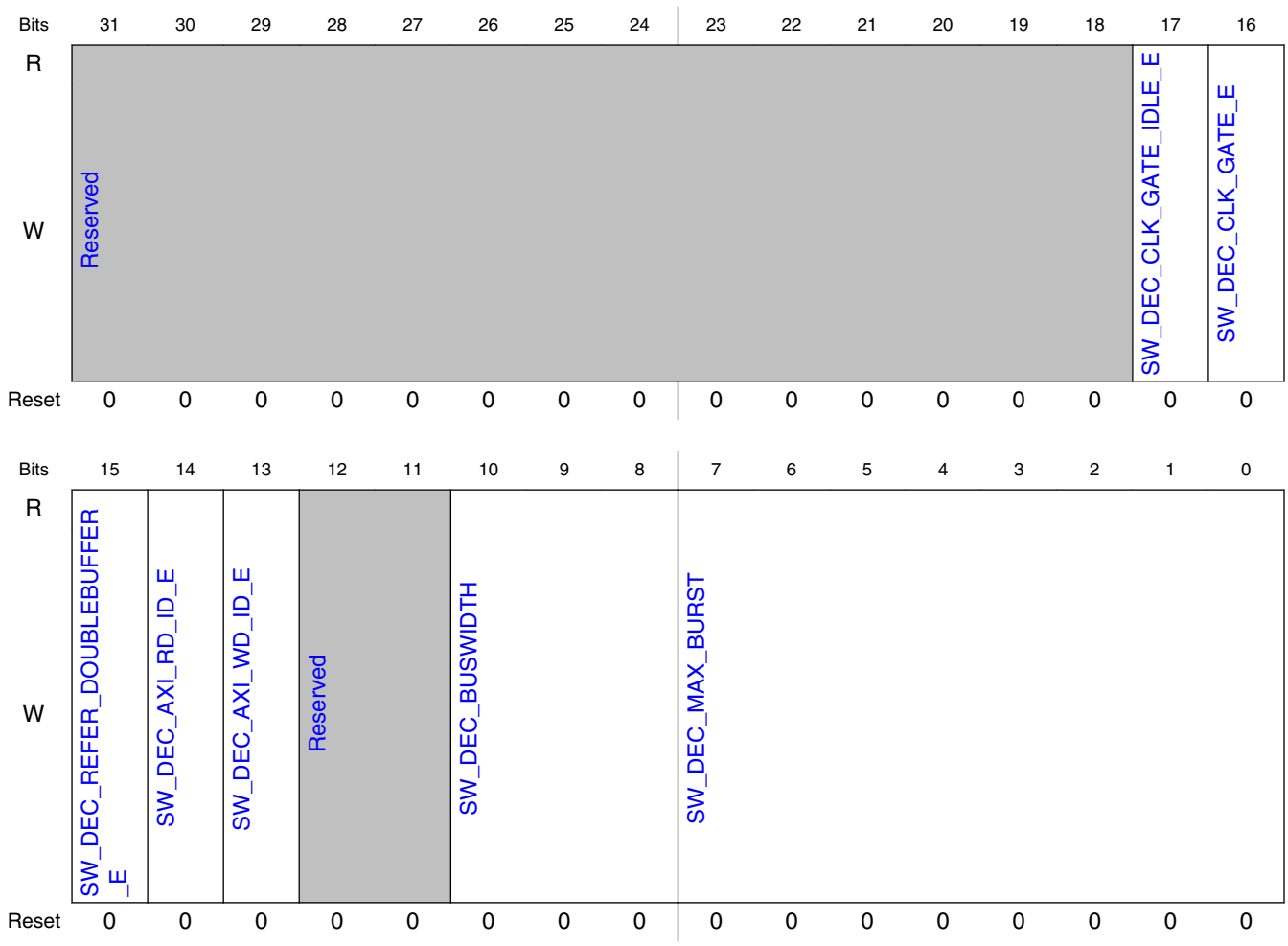
Field	Function
31-0 —	Reserved.

14.2.5.1.60 Device configuration register decoder 2 + Multi core control register (SWREG58)

14.2.5.1.60.1 Offset

Register	Offset
SWREG58	E8h

14.2.5.1.60.2 Diagram



14.2.5.1.60.3 Fields

Field	Function
31-18 —	Reserved.
17 SW_DEC_CLK_GATE_IDLE_E	Clock gating enable for decoder run-time. Generated separate clocks for each block by its own IDLE signal.
16 SW_DEC_CLK_GATE_E	Clock gating enable for picture-wise/decoding format clock gating. Between each picture the clock is gated from HW if this bit is high
15 SW_DEC_REFER_DOUBLEBUFFER_E	HW internal double buffering enable for reference data. This enable requires that there are two buffers available at the configured decoder (see configuration register values)

Table continues on the next page...

Field	Function
14 SW_DEC_AXI_RD_ID_E	SW axi ID enable. When enabled the given sw_dec_axi_rd_id is used as ID base and each sub-block will use offsets 0...max
13 SW_DEC_AXI_WD_ID_E	SW axi ID enable. When enabled the given sw_dec_axi_wd_id is used as ID base and each sub-block will use offsets 0...max
12-11 —	Reserved.
10-8 SW_DEC_BUS WIDTH	Decoder master interface buswidth 000b - 32 bit bus 001b - 64 bit bus 010b - 128 bit bus
7-0 SW_DEC_MAX_BURST	Maximum burst length for decoder bus transactions. Valid values: AXI: 1-256

14.2.5.1.61 Device configuration register AXI ID (SWREG59)

14.2.5.1.61.1 Offset

Register	Offset
SWREG59	ECh

14.2.5.1.61.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_AXI_WR_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_AXI_RD_ID															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.61.3 Fields

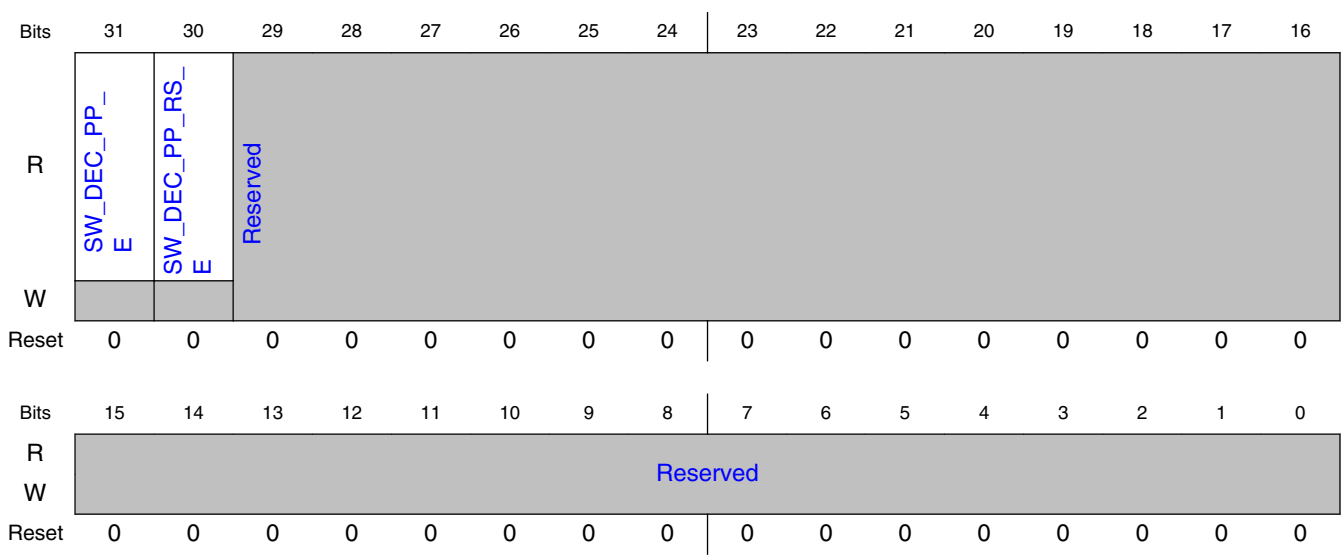
Field	Function
31-16 SW_DEC_AXI_WR_ID	Read ID base for HW write accesses. Each writing device use AXI ID of base+deviceoffset (where device offset is 0 1 2 3...Number of reading sub-blocks)
15-0 SW_DEC_AXI_RD_ID	Write ID base for HW write accesses. Each writing device use AXI ID of base+deviceoffset (where device offset is 0 1 2 3...Number of writing sub-blocks)

14.2.5.1.62 Synthesis configuration register decoder 3 for PP (read only) (SWREG60)

14.2.5.1.62.1 Offset

Register	Offset
SWREG60	F0h

14.2.5.1.62.2 Diagram



14.2.5.1.62.3 Fields

Field	Function
31	Decoder include PP 0b - PP does not exist. None of the PP features can be enabled.

Table continues on the next page...

Field	Function
SW_DEC_PP_E	1b - PP exists
30 SW_DEC_PP_RS_E	Decoder PP raster scan output support 0b - Raster scan output not supported 1b - Raster scan output supported
29-0 —	Reserved.

14.2.5.1.63 Not used (SWREG61)

14.2.5.1.63.1 Offset

Register	Offset
SWREG61	F4h

14.2.5.1.63.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.63.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.64 HW proceed register (CU location) (SWREG62)

14.2.5.1.64.1 Offset

Register	Offset
SWREG62	F8h

14.2.5.1.64.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_CU_LOCATION_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_CU_LOCATION_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.64.3 Fields

Field	Function
31-16 SW_CU_LOCATION_X	Cu horizontal start location X in pixels (returned HW internal position during interrupt)
15-0 SW_CU_LOCATION_Y	Cu vertical start location Y in pixels (returned HW internal position during interrupt)

14.2.5.1.65 HW performance register (cycles running) (SWREG63)

14.2.5.1.65.1 Offset

Register	Offset
SWREG63	FCh

14.2.5.1.65.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_PERF_CYCLE_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_PERF_CYCLE_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.65.3 Fields

Field	Function
31-0 SW_PERF_CYCLE_COUNT	HW clock cycle counter return value. Amount of consumed clock cycles returned to this register when interrupt is being made (any kind of interrupt)

14.2.5.1.66 Base address MSB (bits 63:32) for decoded luminance picture (SWREG64)

14.2.5.1.66.1 Offset

Register	Offset
SWREG64	100h

14.2.5.1.66.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SW_DEC_OUT_YBASE_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SW_DEC_OUT_YBASE_MSB															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.66.3 Fields

Field	Function
31-0 SW_DEC_OUT_YBASE_MSB	Base address MSB (bits 63:32) for decoded luminance picture

14.2.5.1.67 Base address LSB (bits 31:0) for decoded luminance picture (SWREG65)

14.2.5.1.67.1 Offset

Register	Offset
SWREG65	104h

14.2.5.1.67.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.67.3 Fields

Field	Function
31-0 SW_DEC_OUT_YBASE_LSB	Base address LSB (bits 31:0) for decoded luminance picture

14.2.5.1.68 Base address MSB (bits 63:32) for reference luminance picture index 0 (SWREG66)

14.2.5.1.68.1 Offset

Register	Offset
SWREG66	108h

14.2.5.1.68.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.68.3 Fields

Field	Function
31-0 SW_REFER0_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 0

14.2.5.1.69 Base address LSB (bits 31:0) for reference luminance picture index 0 (SWREG67)

14.2.5.1.69.1 Offset

Register	Offset
SWREG67	10Ch

14.2.5.1.69.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.69.3 Fields

Field	Function
31-0 SW_REFER0_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 0

14.2.5.1.70 Base address MSB (bits 63:32) for reference luminance picture index 1 (SWREG68)

14.2.5.1.70.1 Offset

Register	Offset
SWREG68	110h

14.2.5.1.70.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.70.3 Fields

Field	Function
31-0 SW_REFER1_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 1

14.2.5.1.71 Base address LSB (bits 31:0) for reference luminance picture index 1 (SWREG69)

14.2.5.1.71.1 Offset

Register	Offset
SWREG69	114h

14.2.5.1.71.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.71.3 Fields

Field	Function
31-0 SW_REFER1_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 1

14.2.5.1.72 Base address MSB (bits 63:32) for reference luminance picture index 2 (SWREG70)

14.2.5.1.72.1 Offset

Register	Offset
SWREG70	118h

14.2.5.1.72.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.72.3 Fields

Field	Function
31-0 SW_REFER2_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 2

14.2.5.1.73 Base address LSB (bits 31:0) for reference luminance picture index 2 (SWREG71)

14.2.5.1.73.1 Offset

Register	Offset
SWREG71	11Ch

14.2.5.1.73.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.73.3 Fields

Field	Function
31-0 SW_REFER2_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 2

14.2.5.1.74 Base address MSB (bits 63:32) for reference luminance picture index 3 (SWREG72)

14.2.5.1.74.1 Offset

Register	Offset
SWREG72	120h

14.2.5.1.74.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.74.3 Fields

Field	Function
31-0 SW_REFER3_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 3

14.2.5.1.75 Base address LSB (bits 31:0) for reference luminance picture index 3 (SWREG73)

14.2.5.1.75.1 Offset

Register	Offset
SWREG73	124h

14.2.5.1.75.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.75.3 Fields

Field	Function
31-0 SW_REFER3_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 3

14.2.5.1.76 Base address MSB (bits 63:32) for reference luminance picture index 4 (SWREG74)

14.2.5.1.76.1 Offset

Register	Offset
SWREG74	128h

14.2.5.1.76.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.76.3 Fields

Field	Function
31-0 SW_REFER4_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 4

14.2.5.1.77 Base address LSB (bits 31:0) for reference luminance picture index 4 (SWREG75)

14.2.5.1.77.1 Offset

Register	Offset
SWREG75	12Ch

14.2.5.1.77.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.77.3 Fields

Field	Function
31-0 SW_REFER4_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 4

14.2.5.1.78 Base address MSB (bits 63:32) for reference luminance picture index 5 (SWREG76)

14.2.5.1.78.1 Offset

Register	Offset
SWREG76	130h

14.2.5.1.78.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.78.3 Fields

Field	Function
31-0 SW_REFER5_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 5

14.2.5.1.79 Base address LSB (bits 31:0) for reference luminance picture index 5 (SWREG77)

14.2.5.1.79.1 Offset

Register	Offset
SWREG77	134h

14.2.5.1.79.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.79.3 Fields

Field	Function
31-0 SW_REFER5_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 5

14.2.5.1.80 Base address MSB (bits 63:32) for reference luminance picture index 6 /VP9 segment write base MSB (SWREG78)

14.2.5.1.80.1 Offset

Register	Offset
SWREG78	138h

14.2.5.1.80.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_6_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_6_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.80.3 Fields

Field	Function
31-0 BASE_ADDR_6_MSB_BF	For HEVC: [31:0] - sw_refer6_ybase_msb - Base address MSB (bits 63:32) for reference luminance picture index 6 For VP9: [31:0] - sw_segment_write_base_msb - VP9 segment write base MSB

14.2.5.1.81 Base address LSB (bits 31:0) for reference luminance picture index 6 /VP9 segment write base LSB (SWREG79)

14.2.5.1.81.1 Offset

Register	Offset
SWREG79	13Ch

14.2.5.1.81.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_6_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_6_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.81.3 Fields

Field	Function
31-0 BASE_ADDR_6_LSB_BF	For HEVC: [31:0] - sw_refer6_ybase_lsb - Base address LSB (bits 31:0) for reference luminance picture index 6 For VP9: [31:0] - sw_segment_write_base_lsb - VP9 segment write base LSB

14.2.5.1.82 Base address MSB (bits 63:32) for reference luminance picture index 7 /VP9 segment read base MSB (SWREG80)

14.2.5.1.82.1 Offset

Register	Offset
SWREG80	140h

14.2.5.1.82.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_7_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_7_MSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.82.3 Fields

Field	Function
31-0 BASE_ADDR_7_MSB_BF	For HEVC: [31:0] - sw_refer7_ybase_msb - Base address MSB (bits 63:32) for reference luminance picture index 7 For VP9: [31:0] - sw_segment_read_base_msb - VP9 segment read base MSB

14.2.5.1.83 Base address LSB (bits 31:0) for reference luminance picture index 7 /VP9 segment read base LSB (SWREG81)

14.2.5.1.83.1 Offset

Register	Offset
SWREG81	144h

14.2.5.1.83.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR_7_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR_7_LSB_BF															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.83.3 Fields

Field	Function
31-0 BASE_ADDR_7_LSB_BF	For HEVC: [31:0] - sw_refer7_ybase_lsb - Base address LSB (bits 31:0) for reference luminance picture index 7 For VP9: [31:0] - sw_segment_read_base_lsb - VP9 segment read base LSB

14.2.5.1.84 Base address MSB (bits 63:32) for reference luminance picture index 8 (SWREG82)

14.2.5.1.84.1 Offset

Register	Offset
SWREG82	148h

14.2.5.1.84.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.84.3 Fields

Field	Function
31-0 SW_REFER8_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 8

14.2.5.1.85 Base address LSB (bits 31:0) for reference luminance picture index 8 (SWREG83)

14.2.5.1.85.1 Offset

Register	Offset
SWREG83	14Ch

14.2.5.1.85.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.85.3 Fields

Field	Function
31-0 SW_REFER8_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 8

14.2.5.1.86 Base address MSB (bits 63:32) for reference luminance picture index 9 (SWREG84)

14.2.5.1.86.1 Offset

Register	Offset
SWREG84	150h

14.2.5.1.86.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.86.3 Fields

Field	Function
31-0 SW_REFER9_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 9

14.2.5.1.87 Base address LSB (bits 31:0) for reference luminance picture index 9 (SWREG85)

14.2.5.1.87.1 Offset

Register	Offset
SWREG85	154h

14.2.5.1.87.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.87.3 Fields

Field	Function
31-0 SW_REFER9_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 9

14.2.5.1.88 Base address MSB (bits 63:32) for reference luminance picture index 10 (SWREG86)

14.2.5.1.88.1 Offset

Register	Offset
SWREG86	158h

14.2.5.1.88.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.88.3 Fields

Field	Function
31-0 SW_REFER10_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 10

14.2.5.1.89 Base address LSB (bits 31:0) for reference luminance picture index 10 (SWREG87)

14.2.5.1.89.1 Offset

Register	Offset
SWREG87	15Ch

14.2.5.1.89.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.89.3 Fields

Field	Function
31-0 SW_REFER10_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 10

14.2.5.1.90 Base address MSB (bits 63:32) for reference luminance picture index 11 (SWREG88)

14.2.5.1.90.1 Offset

Register	Offset
SWREG88	160h

14.2.5.1.90.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.90.3 Fields

Field	Function
31-0 SW_REFER11_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 11

14.2.5.1.91 Base address LSB (bits 31:0) for reference luminance picture index 11 (SWREG89)

14.2.5.1.91.1 Offset

Register	Offset
SWREG89	164h

14.2.5.1.91.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.91.3 Fields

Field	Function
31-0 SW_REFER11_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 11

14.2.5.1.92 Base address MSB (bits 63:32) for reference luminance picture index 12 (SWREG90)

14.2.5.1.92.1 Offset

Register	Offset
SWREG90	168h

14.2.5.1.92.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.92.3 Fields

Field	Function
31-0 SW_REFER12_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 12

14.2.5.1.93 Base address LSB (bits 31:0) for reference luminance picture index 12 (SWREG91)

14.2.5.1.93.1 Offset

Register	Offset
SWREG91	16Ch

14.2.5.1.93.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.93.3 Fields

Field	Function
31-0 SW_REFER12_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 12

14.2.5.1.94 Base address MSB (bits 63:32) for reference luminance picture index 13 (SWREG92)

14.2.5.1.94.1 Offset

Register	Offset
SWREG92	170h

14.2.5.1.94.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.94.3 Fields

Field	Function
31-0 SW_REFER13_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 13

14.2.5.1.95 Base address LSB (bits 31:0) for reference luminance picture index 13 (SWREG93)

14.2.5.1.95.1 Offset

Register	Offset
SWREG93	174h

14.2.5.1.95.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.95.3 Fields

Field	Function
31-0 SW_REFER13_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 13

14.2.5.1.96 Base address MSB (bits 63:32) for reference luminance picture index 14 (SWREG94)

14.2.5.1.96.1 Offset

Register	Offset
SWREG94	178h

14.2.5.1.96.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.96.3 Fields

Field	Function
31-0 SW_REFER14_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 14

14.2.5.1.97 Base address LSB (bits 31:0) for reference luminance picture index 14 (SWREG95)

14.2.5.1.97.1 Offset

Register	Offset
SWREG95	17Ch

14.2.5.1.97.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.97.3 Fields

Field	Function
31-0 SW_REFER14_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 14

14.2.5.1.98 Base address MSB (bits 63:32) for reference luminance picture index 15 (SWREG96)

14.2.5.1.98.1 Offset

Register	Offset
SWREG96	180h

14.2.5.1.98.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_YBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.98.3 Fields

Field	Function
31-0 SW_REFER15_YBASE_MSB	Base address MSB (bits 63:32) for reference luminance picture index 15

14.2.5.1.99 Base address LSB (bits 31:0) for reference luminance picture index 15 (SWREG97)

14.2.5.1.99.1 Offset

Register	Offset
SWREG97	184h

14.2.5.1.99.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_YBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.99.3 Fields

Field	Function
31-0 SW_REFER15_YBASE_LSB	Base address LSB (bits 31:0) for reference luminance picture index 15

14.2.5.1.100 Base address MSB (bits 63:32) for decoded chrominance picture (SWREG98)

14.2.5.1.100.1 Offset

Register	Offset
SWREG98	188h

14.2.5.1.100.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.100.3 Fields

Field	Function
31-0 SW_DEC_OUT_CBASE_MSB	Base address MSB (bits 64:32) for decoded chrominance picture

14.2.5.1.101 Base address LSB (bits 31:0) for decoded chrominance picture (SWREG99)

14.2.5.1.101.1 Offset

Register	Offset
SWREG99	18Ch

14.2.5.1.101.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.101.3 Fields

Field	Function
31-0 SW_DEC_OUT_CBASE_LSB	Base address LSB (bits 31:0) for decoded chrominance picture

14.2.5.1.102 Base address MSB (bits 63:32) for reference chrominance picture index 0 (SWREG100)

14.2.5.1.102.1 Offset

Register	Offset
SWREG100	190h

14.2.5.1.102.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.102.3 Fields

Field	Function
31-0 SW_REFER0_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 0

14.2.5.1.103 Base address LSB (bits 31:0) for reference chrominance picture index 0 (SWREG101)

14.2.5.1.103.1 Offset

Register	Offset
SWREG101	194h

14.2.5.1.103.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.103.3 Fields

Field	Function
31-0 SW_REFER0_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 0

14.2.5.1.104 Base address MSB (bits 63:32) for reference chrominance picture index 1 (SWREG102)

14.2.5.1.104.1 Offset

Register	Offset
SWREG102	198h

14.2.5.1.104.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.104.3 Fields

Field	Function
31-0 SW_REFER1_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 1

14.2.5.1.105 Base address LSB (bits 31:0) for reference chrominance picture index 1 (SWREG103)

14.2.5.1.105.1 Offset

Register	Offset
SWREG103	19Ch

14.2.5.1.105.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.105.3 Fields

Field	Function
31-0 SW_REFER1_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 1

14.2.5.1.106 Base address MSB (bits 63:32) for reference chrominance picture index 2 (SWREG104)

14.2.5.1.106.1 Offset

Register	Offset
SWREG104	1A0h

14.2.5.1.106.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.106.3 Fields

Field	Function
31-0 SW_REFER2_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 2

14.2.5.1.107 Base address LSB (bits 31:0) for reference chrominance picture index 2 (SWREG105)

14.2.5.1.107.1 Offset

Register	Offset
SWREG105	1A4h

14.2.5.1.107.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.107.3 Fields

Field	Function
31-0 SW_REFER2_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 2

14.2.5.1.108 Base address MSB (bits 63:32) for reference chrominance picture index 3 (SWREG106)

14.2.5.1.108.1 Offset

Register	Offset
SWREG106	1A8h

14.2.5.1.108.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.108.3 Fields

Field	Function
31-0 SW_REFER3_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 3

14.2.5.1.109 Base address LSB (bits 31:0) for reference chrominance picture index 3 (SWREG107)

14.2.5.1.109.1 Offset

Register	Offset
SWREG107	1ACh

14.2.5.1.109.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.109.3 Fields

Field	Function
31-0 SW_REFER3_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 3

14.2.5.1.110 Base address MSB (bits 63:32) for reference chrominance picture index 4 (SWREG108)

14.2.5.1.110.1 Offset

Register	Offset
SWREG108	1B0h

14.2.5.1.110.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.110.3 Fields

Field	Function
31-0 SW_REFER4_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 4

14.2.5.1.111 Base address LSB (bits 31:0) for reference chrominance picture index 4 (SWREG109)

14.2.5.1.111.1 Offset

Register	Offset
SWREG109	1B4h

14.2.5.1.111.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.111.3 Fields

Field	Function
31-0 SW_REFER4_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 4

14.2.5.1.112 Base address MSB (bits 63:32) for reference chrominance picture index 5 (SWREG110)

14.2.5.1.112.1 Offset

Register	Offset
SWREG110	1B8h

14.2.5.1.112.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.112.3 Fields

Field	Function
31-0 SW_REFER5_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 5

14.2.5.1.113 Base address LSB (bits 31:0) for reference chrominance picture index 5 (SWREG111)

14.2.5.1.113.1 Offset

Register	Offset
SWREG111	1BCh

14.2.5.1.113.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.113.3 Fields

Field	Function
31-0 SW_REFER5_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 5

14.2.5.1.114 Base address MSB (bits 63:32) for reference chrominance picture index 6 (SWREG112)

14.2.5.1.114.1 Offset

Register	Offset
SWREG112	1C0h

14.2.5.1.114.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.114.3 Fields

Field	Function
31-0 SW_REFER6_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 6

14.2.5.1.115 Base address LSB (bits 31:0) for reference chrominance picture index 6 (SWREG113)

14.2.5.1.115.1 Offset

Register	Offset
SWREG113	1C4h

14.2.5.1.115.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.115.3 Fields

Field	Function
31-0 SW_REFER6_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 6

14.2.5.1.116 Base address MSB (bits 63:32) for reference chrominance picture index 7 (SWREG114)

14.2.5.1.116.1 Offset

Register	Offset
SWREG114	1C8h

14.2.5.1.116.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.116.3 Fields

Field	Function
31-0 SW_REFER7_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 7

14.2.5.1.117 Base address LSB (bits 31:0) for reference chrominance picture index 7 (SWREG115)

14.2.5.1.117.1 Offset

Register	Offset
SWREG115	1CCh

14.2.5.1.117.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.117.3 Fields

Field	Function
31-0 SW_REFER7_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 7

14.2.5.1.118 Base address MSB (bits 63:32) for reference chrominance picture index 8 (SWREG116)

14.2.5.1.118.1 Offset

Register	Offset
SWREG116	1D0h

14.2.5.1.118.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.118.3 Fields

Field	Function
31-0 SW_REFER8_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 8

14.2.5.1.119 Base address LSB (bits 31:0) for reference chrominance picture index 8 (SWREG117)

14.2.5.1.119.1 Offset

Register	Offset
SWREG117	1D4h

14.2.5.1.119.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.119.3 Fields

Field	Function
31-0 SW_REFER8_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 8

14.2.5.1.120 Base address MSB (bits 63:32) for reference chrominance picture index 9 (SWREG118)

14.2.5.1.120.1 Offset

Register	Offset
SWREG118	1D8h

14.2.5.1.120.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.120.3 Fields

Field	Function
31-0 SW_REFER9_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 9

14.2.5.1.121 Base address LSB (bits 31:0) for reference chrominance picture index 9 (SWREG119)

14.2.5.1.121.1 Offset

Register	Offset
SWREG119	1DCh

14.2.5.1.121.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.121.3 Fields

Field	Function
31-0 SW_REFER9_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 9

14.2.5.1.122 Base address MSB (bits 63:32) for reference chrominance picture index 10 (SWREG120)

14.2.5.1.122.1 Offset

Register	Offset
SWREG120	1E0h

14.2.5.1.122.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.122.3 Fields

Field	Function
31-0 SW_REFER10_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 10

14.2.5.1.123 Base address LSB (bits 31:0) for reference chrominance picture index 10 (SWREG121)

14.2.5.1.123.1 Offset

Register	Offset
SWREG121	1E4h

14.2.5.1.123.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.123.3 Fields

Field	Function
31-0 SW_REFER10_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 10

14.2.5.1.124 Base address MSB (bits 63:32) for reference chrominance picture index 11 (SWREG122)

14.2.5.1.124.1 Offset

Register	Offset
SWREG122	1E8h

14.2.5.1.124.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.124.3 Fields

Field	Function
31-0 SW_REFER11_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 11

14.2.5.1.125 Base address LSB (bits 31:0) for reference chrominance picture index 11 (SWREG123)

14.2.5.1.125.1 Offset

Register	Offset
SWREG123	1ECh

14.2.5.1.125.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.125.3 Fields

Field	Function
31-0 SW_REFER11_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 11

14.2.5.1.126 Base address MSB (bits 63:32) for reference chrominance picture index 12 (SWREG124)

14.2.5.1.126.1 Offset

Register	Offset
SWREG124	1F0h

14.2.5.1.126.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.126.3 Fields

Field	Function
31-0 SW_REFER12_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 12

14.2.5.1.127 Base address LSB (bits 31:0) for reference chrominance picture index 12 (SWREG125)

14.2.5.1.127.1 Offset

Register	Offset
SWREG125	1F4h

14.2.5.1.127.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.127.3 Fields

Field	Function
31-0 SW_REFER12_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 12

14.2.5.1.128 Base address MSB (bits 63:32) for reference chrominance picture index 13 (SWREG126)

14.2.5.1.128.1 Offset

Register	Offset
SWREG126	1F8h

14.2.5.1.128.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.128.3 Fields

Field	Function
31-0 SW_REFER13_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 13

14.2.5.1.129 Base address LSB (bits 31:0) for reference chrominance picture index 13 (SWREG127)

14.2.5.1.129.1 Offset

Register	Offset
SWREG127	1FCh

14.2.5.1.129.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.129.3 Fields

Field	Function
31-0 SW_REFER13_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 13

14.2.5.1.130 Base address MSB (bits 63:32) for reference chrominance picture index 14 (SWREG128)

14.2.5.1.130.1 Offset

Register	Offset
SWREG128	200h

14.2.5.1.130.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.130.3 Fields

Field	Function
31-0 SW_REFER14_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 14

14.2.5.1.131 Base address LSB (bits 31:0) for reference chrominance picture index 14 (SWREG129)

14.2.5.1.131.1 Offset

Register	Offset
SWREG129	204h

14.2.5.1.131.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.131.3 Fields

Field	Function
31-0 SW_REFER14_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 14

14.2.5.1.132 Base address MSB (bits 63:32) for reference chrominance picture index 15 (SWREG130)

14.2.5.1.132.1 Offset

Register	Offset
SWREG130	208h

14.2.5.1.132.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_CBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.132.3 Fields

Field	Function
31-0 SW_REFER15_CBASE_MSB	Base address MSB (bits 63:32) for reference chrominance picture index 15

14.2.5.1.133 Base address LSB (bits 31:0) for reference chrominance picture index 15 (SWREG131)

14.2.5.1.133.1 Offset

Register	Offset
SWREG131	20Ch

14.2.5.1.133.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_CBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.133.3 Fields

Field	Function
31-0 SW_REFER15_CBASE_LSB	Base address LSB (bits 31:0) for reference chrominance picture index 15

14.2.5.1.134 Base address MSB (bits 63:32) for decoded direct mode MVS (SWREG132)

14.2.5.1.134.1 Offset

Register	Offset
SWREG132	210h

14.2.5.1.134.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.134.3 Fields

Field	Function
31-0 SW_DEC_OUT_DBASE_MSB	Base address MSB (bits 63:32) for decoded direct mode MVS

14.2.5.1.135 Base address LSB (bits 31:0) for decoded direct mode MVS (SWREG133)

14.2.5.1.135.1 Offset

Register	Offset
SWREG133	214h

14.2.5.1.135.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.135.3 Fields

Field	Function
31-0 SW_DEC_OUT_DBASE_LSB	Base address LSB (bits 31:0) for decoded direct mode MVS

14.2.5.1.136 Base address MSB (bits 63:32) for reference direct mode MVS index 0 (SWREG134)

14.2.5.1.136.1 Offset

Register	Offset
SWREG134	218h

14.2.5.1.136.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.136.3 Fields

Field	Function
31-0 SW_REFER0_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 0

14.2.5.1.137 Base address LSB (bits 31:0) for reference direct mode MVS index 0 (SWREG135)

14.2.5.1.137.1 Offset

Register	Offset
SWREG135	21Ch

14.2.5.1.137.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.137.3 Fields

Field	Function
31-0 SW_REFER0_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 0

14.2.5.1.138 Base address MSB (bits 63:32) for reference direct mode MVS index 1 (SWREG136)

14.2.5.1.138.1 Offset

Register	Offset
SWREG136	220h

14.2.5.1.138.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.138.3 Fields

Field	Function
31-0 SW_REFER1_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 1

14.2.5.1.139 Base address LSB (bits 31:0) for reference direct mode MVS index 1 (SWREG137)

14.2.5.1.139.1 Offset

Register	Offset
SWREG137	224h

14.2.5.1.139.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.139.3 Fields

Field	Function
31-0 SW_REFER1_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 1

14.2.5.1.140 Base address MSB (bits 63:32) for reference direct mode MVS index 2 (SWREG138)

14.2.5.1.140.1 Offset

Register	Offset
SWREG138	228h

14.2.5.1.140.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.140.3 Fields

Field	Function
31-0 SW_REFER2_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 2

14.2.5.1.141 Base address LSB (bits 31:0) for reference direct mode MVS index 2 (SWREG139)

14.2.5.1.141.1 Offset

Register	Offset
SWREG139	22Ch

14.2.5.1.141.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.141.3 Fields

Field	Function
31-0 SW_REFER2_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 2

14.2.5.1.142 Base address MSB (bits 63:32) for reference direct mode MVS index 3 (SWREG140)

14.2.5.1.142.1 Offset

Register	Offset
SWREG140	230h

14.2.5.1.142.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.142.3 Fields

Field	Function
31-0 SW_REFER3_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 3

14.2.5.1.143 Base address LSB (bits 31:0) for reference direct mode MVS index 3 (SWREG141)

14.2.5.1.143.1 Offset

Register	Offset
SWREG141	234h

14.2.5.1.143.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.143.3 Fields

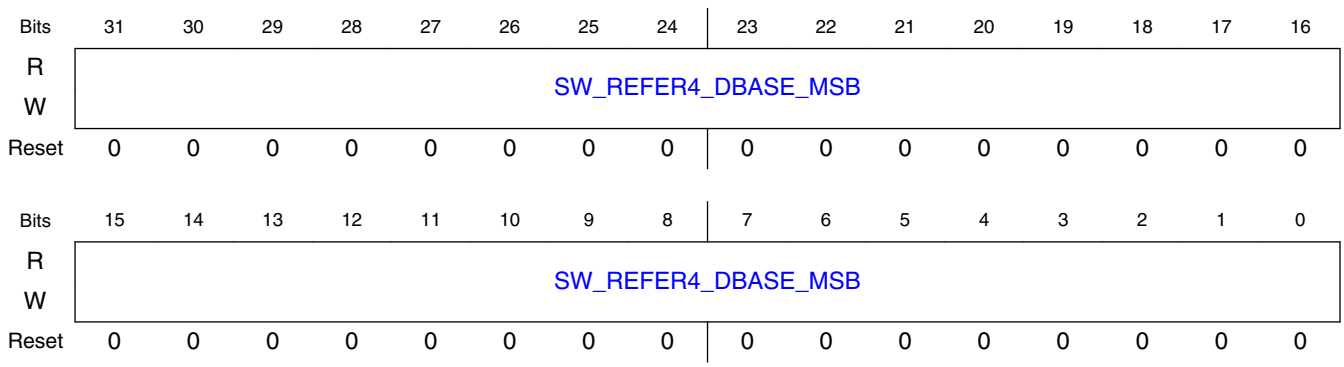
Field	Function
31-0 SW_REFER3_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 3

14.2.5.1.144 Base address MSB (bits 63:32) for reference direct mode MVS index 4 (SWREG142)

14.2.5.1.144.1 Offset

Register	Offset
SWREG142	238h

14.2.5.1.144.2 Diagram



14.2.5.1.144.3 Fields

Field	Function
31-0 SW_REFER4_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 4

14.2.5.1.145 Base address LSB (bits 31:0) for reference direct mode MVS index 4 (SWREG143)

14.2.5.1.145.1 Offset

Register	Offset
SWREG143	23Ch

14.2.5.1.145.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.145.3 Fields

Field	Function
31-0 SW_REFER4_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 4

14.2.5.1.146 Base address MSB (bits 63:32) for reference direct mode MVS index 5 (SWREG144)

14.2.5.1.146.1 Offset

Register	Offset
SWREG144	240h

14.2.5.1.146.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.146.3 Fields

Field	Function
31-0 SW_REFER5_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 5

14.2.5.1.147 Base address LSB (bits 31:0) for reference direct mode MVS index 5 (SWREG145)

14.2.5.1.147.1 Offset

Register	Offset
SWREG145	244h

14.2.5.1.147.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.147.3 Fields

Field	Function
31-0 SW_REFER5_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 5

14.2.5.1.148 Base address MSB (bits 63:32) for reference direct mode MVS index 6 (SWREG146)

14.2.5.1.148.1 Offset

Register	Offset
SWREG146	248h

14.2.5.1.148.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.148.3 Fields

Field	Function
31-0 SW_REFER6_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 6

14.2.5.1.149 Base address LSB (bits 31:0) for reference direct mode MVS index 6 (SWREG147)

14.2.5.1.149.1 Offset

Register	Offset
SWREG147	24Ch

14.2.5.1.149.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.149.3 Fields

Field	Function
31-0 SW_REFER6_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 6

14.2.5.1.150 Base address MSB (bits 63:32) for reference direct mode MVS index 7 (SWREG148)

14.2.5.1.150.1 Offset

Register	Offset
SWREG148	250h

14.2.5.1.150.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.150.3 Fields

Field	Function
31-0 SW_REFER7_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 7

14.2.5.1.151 Base address LSB (bits 31:0) for reference direct mode MVS index 7 (SWREG149)

14.2.5.1.151.1 Offset

Register	Offset
SWREG149	254h

14.2.5.1.151.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.151.3 Fields

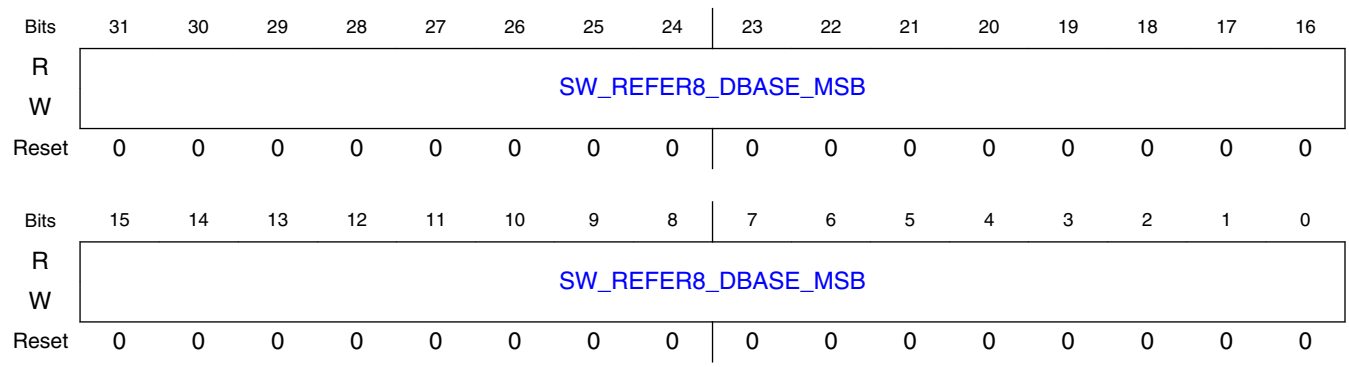
Field	Function
31-0 SW_REFER7_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 7

14.2.5.1.152 Base address MSB (bits 63:32) for reference direct mode MVS index 8 (SWREG150)

14.2.5.1.152.1 Offset

Register	Offset
SWREG150	258h

14.2.5.1.152.2 Diagram



14.2.5.1.152.3 Fields

Field	Function
31-0 SW_REFER8_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 8

14.2.5.1.153 Base address LSB (bits 31:0) for reference direct mode MVS index 8 (SWREG151)

14.2.5.1.153.1 Offset

Register	Offset
SWREG151	25Ch

14.2.5.1.153.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.153.3 Fields

Field	Function
31-0 SW_REFER8_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode mode MVS index 8

14.2.5.1.154 Base address MSB (bits 63:32) for reference direct mode mode MVS index 9 (SWREG152)

14.2.5.1.154.1 Offset

Register	Offset
SWREG152	260h

14.2.5.1.154.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.154.3 Fields

Field	Function
31-0 SW_REFER9_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 9

14.2.5.1.155 Base address LSB (bits 31:0) for reference direct mode mode MVS index 9 (SWREG153)

14.2.5.1.155.1 Offset

Register	Offset
SWREG153	264h

14.2.5.1.155.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.155.3 Fields

Field	Function
31-0 SW_REFER9_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode mode MVS index 9

14.2.5.1.156 Base address MSB (bits 63:32) for reference direct mode MVS index 10 (SWREG154)

14.2.5.1.156.1 Offset

Register	Offset
SWREG154	268h

14.2.5.1.156.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.156.3 Fields

Field	Function
31-0 SW_REFER10_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 10

14.2.5.1.157 Base address LSB (bits 31:0) for reference direct mode MVS index 10 (SWREG155)

14.2.5.1.157.1 Offset

Register	Offset
SWREG155	26Ch

14.2.5.1.157.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.157.3 Fields

Field	Function
31-0 SW_REFER10_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 10

14.2.5.1.158 Base address MSB (bits 63:32) for reference direct mode MVS index 11 (SWREG156)

14.2.5.1.158.1 Offset

Register	Offset
SWREG156	270h

14.2.5.1.158.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.158.3 Fields

Field	Function
31-0 SW_REFER11_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 11

14.2.5.1.159 Base address LSB (bits 31:0) for reference direct mode MVS index 11 (SWREG157)

14.2.5.1.159.1 Offset

Register	Offset
SWREG157	274h

14.2.5.1.159.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.159.3 Fields

Field	Function
31-0 SW_REFER11_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 11

14.2.5.1.160 Base address MSB (bits 63:32) for reference direct mode MVS index 12 (SWREG158)

14.2.5.1.160.1 Offset

Register	Offset
SWREG158	278h

14.2.5.1.160.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.160.3 Fields

Field	Function
31-0 SW_REFER12_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 12

14.2.5.1.161 Base address LSB (bits 31:0) for reference direct mode MVS index 12 (SWREG159)

14.2.5.1.161.1 Offset

Register	Offset
SWREG159	27Ch

14.2.5.1.161.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.161.3 Fields

Field	Function
31-0 SW_REFER12_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 12

14.2.5.1.162 Base address MSB (bits 63:32) for reference direct mode MVS index 13 (SWREG160)

14.2.5.1.162.1 Offset

Register	Offset
SWREG160	280h

14.2.5.1.162.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.162.3 Fields

Field	Function
31-0 SW_REFER13_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 13

14.2.5.1.163 Base address LSB (bits 31:0) for reference direct mode MVS index 13 (SWREG161)

14.2.5.1.163.1 Offset

Register	Offset
SWREG161	284h

14.2.5.1.163.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.163.3 Fields

Field	Function
31-0 SW_REFER13_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 13

14.2.5.1.164 Base address MSB (bits 63:32) for reference direct mode MVS index 14 (SWREG162)

14.2.5.1.164.1 Offset

Register	Offset
SWREG162	288h

14.2.5.1.164.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.164.3 Fields

Field	Function
31-0 SW_REFER14_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 14

14.2.5.1.165 Base address LSB (bits 31:0) for reference direct mode MVS index 14 (SWREG163)

14.2.5.1.165.1 Offset

Register	Offset
SWREG163	28Ch

14.2.5.1.165.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.165.3 Fields

Field	Function
31-0 SW_REFER14_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 14

14.2.5.1.166 Base address MSB (bits 63:32) for reference direct mode MVS index 15 (SWREG164)

14.2.5.1.166.1 Offset

Register	Offset
SWREG164	290h

14.2.5.1.166.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_DBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.166.3 Fields

Field	Function
31-0 SW_REFER15_DBASE_MSB	Base address MSB (bits 63:32) for reference direct mode MVS index 15

14.2.5.1.167 Base address LSB (bits 31:0) for reference direct mode MVS index 15 (SWREG165)

14.2.5.1.167.1 Offset

Register	Offset
SWREG165	294h

14.2.5.1.167.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_DBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.167.3 Fields

Field	Function
31-0 SW_REFER15_DBASE_LSB	Base address LSB (bits 31:0) for reference direct mode MVS index 15

14.2.5.1.168 Base address MSB (bits 63:32) for tile sizes (SWREG166)

14.2.5.1.168.1 Offset

Register	Offset
SWREG166	298h

14.2.5.1.168.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_TILE_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_TILE_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.168.3 Fields

Field	Function
31-0 SW_TILE_BASE_MSB	Base address MSB (bits 63:32) for tile sizes

14.2.5.1.169 Base address LSB (bits 31:0) for tile sizes (SWREG167)

14.2.5.1.169.1 Offset

Register	Offset
SWREG167	29Ch

14.2.5.1.169.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_TILE_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_TILE_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.169.3 Fields

Field	Function
31-0 SW_TILE_BASE_LSB	Base address LSB (bits 31:0) for tile sizes

14.2.5.1.170 Base address MSB (bits 63:32) for / stream start address/ decoded end addr register (SWREG168)

14.2.5.1.170.1 Offset

Register	Offset
SWREG168	2A0h

14.2.5.1.170.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.170.3 Fields

Field	Function
31-0 SW_STREAM_BASE_MSB	Base address MSB (bits 63:32) for / stream start address/decoded end addr register

14.2.5.1.171 Base address LSB (bits 31:0) for / stream start address/ decoded end addr register (SWREG169)

14.2.5.1.171.1 Offset

Register	Offset
SWREG169	2A4h

14.2.5.1.171.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STREAM_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STREAM_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.171.3 Fields

Field	Function
31-0 SW_STREAM_BASE_LSB	Base address LSB (bits 31:0) for / stream start address/decoded end addr register

14.2.5.1.172 Base address MSB (bits 63:32) for scaling lists / VP9 CTX counter values (SWREG170)

14.2.5.1.172.1 Offset

Register	Offset
SWREG170	2A8h

14.2.5.1.172.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_SCALE_LIST_CTX_COUNTER_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_SCALE_LIST_CTX_COUNTER_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.172.3 Fields

Field	Function
31-0 SW_SCALE_LIST_CTX_COUNTER_BASE_MSB	HEVC: Base address MSB (bits 63:32) for scaling lists VP9: CTX counter values

14.2.5.1.173 Base address LSB (bits 31:0) for scaling lists / VP9 CTX counter values (SWREG171)

14.2.5.1.173.1 Offset

Register	Offset
SWREG171	2ACh

14.2.5.1.173.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.173.3 Fields

Field	Function
31-0 SW_SCALE_LISTT_CTX_COUNTER_BASE_LSB	HEVC: Base address LSB (bits 31:0) for scaling lists VP9: CTX counter values

14.2.5.1.174 Base address MSB (bits 63:32) for stream propability tables (SWREG172)

14.2.5.1.174.1 Offset

Register	Offset
SWREG172	2B0h

14.2.5.1.174.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PROB_TAB_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PROB_TAB_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.174.3 Fields

Field	Function
31-0 SW_PROB_TAB_BASE_MSB	Base address MSB (bits 63:32) for stream propability tables

14.2.5.1.175 Base address LSB (bits 31:0) for stream propability tables (SWREG173)

14.2.5.1.175.1 Offset

Register	Offset
SWREG173	2B4h

14.2.5.1.175.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_PROB_TAB_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_PROB_TAB_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.175.3 Fields

Field	Function
31-0 SW_PROB_TAB_BASE_LSB	Base address LSB (bits 31:0) for stream propability tables

14.2.5.1.176 Base address MSB (bits 63:32) for decoder output raster scan Y picture (SWREG174)

14.2.5.1.176.1 Offset

Register	Offset
SWREG174	2B8h

14.2.5.1.176.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.176.3 Fields

Field	Function
31-0 SW_DEC_RSY_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan Y picture

14.2.5.1.177 Base address LSB (bits 31:0) for decoder output raster scan Y picture (SWREG175)

14.2.5.1.177.1 Offset

Register	Offset
SWREG175	2BCh

14.2.5.1.177.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.177.3 Fields

Field	Function
31-0 SW_DEC_RSY_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan Y picture

14.2.5.1.178 Base address MSB (bits 63:32) for decoder output raster scan C picture (SWREG176)

14.2.5.1.178.1 Offset

Register	Offset
SWREG176	2C0h

14.2.5.1.178.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.178.3 Fields

Field	Function
31-0 SW_DEC_RSC_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan C picture

14.2.5.1.179 Base address LSB (bits 31:0) for decoder output raster scan C picture (SWREG177)

14.2.5.1.179.1 Offset

Register	Offset
SWREG177	2C4h

14.2.5.1.179.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_RSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_RSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.179.3 Fields

Field	Function
31-0 SW_DEC_RSC_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan C picture

14.2.5.1.180 Base address MSB (bits 63:32) for tile border coeffients of filter (SWREG178)

14.2.5.1.180.1 Offset

Register	Offset
SWREG178	2C8h

14.2.5.1.180.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_FILT_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_FILT_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.180.3 Fields

Field	Function
31-0 SW_DEC_VERT_FILT_BASE_MSB	Base address MSB to store/read filtering coeffients of current picture at tile border.

14.2.5.1.181 Base address LSB (bits 31:0) for tile border coeffients of filter (SWREG179)

14.2.5.1.181.1 Offset

Register	Offset
SWREG179	2CCh

14.2.5.1.181.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_FILT_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_FILT_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.181.3 Fields

Field	Function
31-0 SW_DEC_VERT_FILT_BASE_LSB	Base address LSB to store/read filtering coeffients of current picture at tile border.

14.2.5.1.182 Base address MSB (bits 63:32) for tile border coeffients of sao (SWREG180)

14.2.5.1.182.1 Offset

Register	Offset
SWREG180	2D0h

14.2.5.1.182.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_SAO_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_SAO_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.182.3 Fields

Field	Function
31-0 SW_DEC_VERT_SAO_BASE_MSB	Base address MSB to store/read sao coeffients of current picture at tile border.

14.2.5.1.183 Base address LSB (bits 31:0) for tile border coeffients of sao (SWREG181)

14.2.5.1.183.1 Offset

Register	Offset
SWREG181	2D4h

14.2.5.1.183.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_VERT_SAO_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_VERT_SAO_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.183.3 Fields

Field	Function
31-0 SW_DEC_VERT_SAO_BASE_LSB	Base address LSB to store/read sao coeffients of current picture at tile border.

14.2.5.1.184 Base address MSB (bits 63:32) for tile border bsd control data (SWREG182)

14.2.5.1.184.1 Offset

Register	Offset
SWREG182	2D8h

14.2.5.1.184.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_BSD_CTRL_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_BSD_CTRL_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.184.3 Fields

Field	Function
31-0 SW_DEC_BSD_CTRL_BASE_MSB	Base address MSB to store/read BSD control data of current picture at tile border.

14.2.5.1.185 Base address LSB (bits 31:0) for tile border bsd control data (SWREG183)

14.2.5.1.185.1 Offset

Register	Offset
SWREG183	2DCh

14.2.5.1.185.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_BSD_CTRL_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_BSD_CTRL_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.185.3 Fields

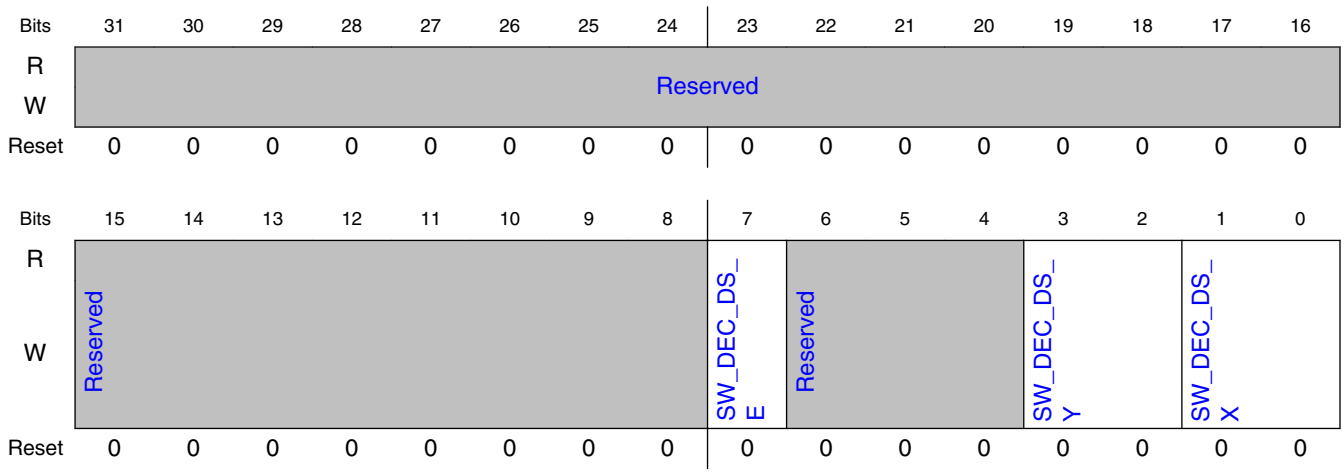
Field	Function
31-0 SW_DEC_BSD_CTRL_BASE_LSB	Base address LSB to store/read BSD control data of current picture at tile border.

14.2.5.1.186 Raster scan down scale control register MSM (SWREG184)

14.2.5.1.186.1 Offset

Register	Offset
SWREG184	2E0h

14.2.5.1.186.2 Diagram



14.2.5.1.186.3 Fields

Field	Function
31-8 —	Reserved.
7 SW_DEC_DS_E	Raster scan down scale enable 0b - Disable 1b - Enable
6-4 —	Reserved.
3-2 SW_DEC_DS_Y	Y coordinate down scale times for raster scan output picture data 00b - 1/2 01b - 1/4 10b - 1/8
1-0 SW_DEC_DS_X	X coordinate down scale times for raster scan output picture data 00b - 1/2 01b - 1/4 10b - 1/8

14.2.5.1.187 Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture (SWREG185)

14.2.5.1.187.1 Offset

Register	Offset
SWREG185	2E4h

14.2.5.1.187.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSY_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.187.3 Fields

Field	Function
31-0 SW_DEC_DSY_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan down scale Y picture

14.2.5.1.188 Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture (SWREG186)

14.2.5.1.188.1 Offset

Register	Offset
SWREG186	2E8h

14.2.5.1.188.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSY_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.188.3 Fields

Field	Function
31-0 SW_DEC_DSY_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan down scale Y picture

14.2.5.1.189 Base address MSB (bits 63:32) for decoder output raster scan down scale C picture (SWREG187)

14.2.5.1.189.1 Offset

Register	Offset
SWREG187	2ECh

14.2.5.1.189.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSC_BASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.189.3 Fields

Field	Function
31-0 SW_DEC_DSC_BASE_MSB	Base address MSB (bits 63:32) for decoder output raster scan down scale C picture

14.2.5.1.190 Base address LSB (bits 31:0) for decoder output raster scan down scale C picture (SWREG188)

14.2.5.1.190.1 Offset

Register	Offset
SWREG188	2F0h

14.2.5.1.190.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_DSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_DSC_BASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.190.3 Fields

Field	Function
31-0 SW_DEC_DSC_BASE_LSB	Base address LSB (bits 31:0) for decoder output raster scan down scale C picture

14.2.5.1.191 Base address MSB (bits 63:32) for decoder output compress luminance table (SWREG189)

14.2.5.1.191.1 Offset

Register	Offset
SWREG189	2F4h

14.2.5.1.191.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.191.3 Fields

Field	Function
31-0 SW_DEC_OUT_TYBASE_MSB	Base address MSB (bits 63:32) for decoder output compress luminance table

14.2.5.1.192 Base address LSB (bits 31:0) for decoder output compress luminance table (SWREG190)

14.2.5.1.192.1 Offset

Register	Offset
SWREG190	2F8h

14.2.5.1.192.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.192.3 Fields

Field	Function
31-0 SW_DEC_OUT_TYBASE_LSB	Base address LSB (bits 31:0) for decoder output compress luminance table

14.2.5.1.193 Base address MSB (bits 63:32) for reference compress luminance table index 0 (SWREG191)

14.2.5.1.193.1 Offset

Register	Offset
SWREG191	2FCh

14.2.5.1.193.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.193.3 Fields

Field	Function
31-0 SW_REFER0_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 0

14.2.5.1.194 Base address LSB (bits 31:0) for reference compress luminance table index 0 (SWREG192)

14.2.5.1.194.1 Offset

Register	Offset
SWREG192	300h

14.2.5.1.194.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.194.3 Fields

Field	Function
31-0 SW_REFER0_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 0

14.2.5.1.195 Base address MSB (bits 63:32) for reference compress luminance table index 1 (SWREG193)

14.2.5.1.195.1 Offset

Register	Offset
SWREG193	304h

14.2.5.1.195.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.195.3 Fields

Field	Function
31-0 SW_REFER1_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 1

14.2.5.1.196 Base address LSB (bits 31:0) for reference compress luminance table index 1 (SWREG194)

14.2.5.1.196.1 Offset

Register	Offset
SWREG194	308h

14.2.5.1.196.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.196.3 Fields

Field	Function
31-0 SW_REFER1_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 1

14.2.5.1.197 Base address MSB (bits 63:32) for reference compress luminance table index 2 (SWREG195)

14.2.5.1.197.1 Offset

Register	Offset
SWREG195	30Ch

14.2.5.1.197.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.197.3 Fields

Field	Function
31-0 SW_REFER2_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 2

14.2.5.1.198 Base address LSB (bits 31:0) for reference compress luminance table index 2 (SWREG196)

14.2.5.1.198.1 Offset

Register	Offset
SWREG196	310h

14.2.5.1.198.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.198.3 Fields

Field	Function
31-0 SW_REFER2_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 2

14.2.5.1.199 Base address MSB (bits 63:32) for reference compress luminance table index 3 (SWREG197)

14.2.5.1.199.1 Offset

Register	Offset
SWREG197	314h

14.2.5.1.199.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.199.3 Fields

Field	Function
31-0 SW_REFER3_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 3

14.2.5.1.200 Base address LSB (bits 31:0) for reference compress luminance table index 3 (SWREG198)

14.2.5.1.200.1 Offset

Register	Offset
SWREG198	318h

14.2.5.1.200.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.200.3 Fields

Field	Function
31-0 SW_REFER3_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 3

14.2.5.1.201 Base address MSB (bits 63:32) for reference compress luminance table index 4 (SWREG199)

14.2.5.1.201.1 Offset

Register	Offset
SWREG199	31Ch

14.2.5.1.201.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.201.3 Fields

Field	Function
31-0 SW_REFER4_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 4

14.2.5.1.202 Base address LSB (bits 31:0) for reference compress luminance table index 4 (SWREG200)

14.2.5.1.202.1 Offset

Register	Offset
SWREG200	320h

14.2.5.1.202.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.202.3 Fields

Field	Function
31-0 SW_REFER4_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 4

14.2.5.1.203 Base address MSB (bits 63:32) for reference compress luminance table index 5 (SWREG201)

14.2.5.1.203.1 Offset

Register	Offset
SWREG201	324h

14.2.5.1.203.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.203.3 Fields

Field	Function
31-0 SW_REFER5_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 5

14.2.5.1.204 Base address LSB (bits 31:0) for reference compress luminance table index 5 (SWREG202)

14.2.5.1.204.1 Offset

Register	Offset
SWREG202	328h

14.2.5.1.204.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.204.3 Fields

Field	Function
31-0 SW_REFER5_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 5

14.2.5.1.205 Base address MSB (bits 63:32) for reference compress luminance table index 6 (SWREG203)

14.2.5.1.205.1 Offset

Register	Offset
SWREG203	32Ch

14.2.5.1.205.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.205.3 Fields

Field	Function
31-0 SW_REFER6_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 6

14.2.5.1.206 Base address LSB (bits 31:0) for reference compress luminance table index 6 (SWREG204)

14.2.5.1.206.1 Offset

Register	Offset
SWREG204	330h

14.2.5.1.206.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.206.3 Fields

Field	Function
31-0 SW_REFER6_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 6

14.2.5.1.207 Base address MSB (bits 63:32) for reference compress luminance table index 7 (SWREG205)

14.2.5.1.207.1 Offset

Register	Offset
SWREG205	334h

14.2.5.1.207.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.207.3 Fields

Field	Function
31-0 SW_REFER7_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 7

14.2.5.1.208 Base address LSB (bits 31:0) for reference compress luminance table index 7 (SWREG206)

14.2.5.1.208.1 Offset

Register	Offset
SWREG206	338h

14.2.5.1.208.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.208.3 Fields

Field	Function
31-0 SW_REFER7_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 7

14.2.5.1.209 Base address MSB (bits 63:32) for reference compress luminance table index 8 (SWREG207)

14.2.5.1.209.1 Offset

Register	Offset
SWREG207	33Ch

14.2.5.1.209.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.209.3 Fields

Field	Function
31-0 SW_REFER8_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 8

14.2.5.1.210 Base address LSB (bits 31:0) for reference compress luminance table index 8 (SWREG208)

14.2.5.1.210.1 Offset

Register	Offset
SWREG208	340h

14.2.5.1.210.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.210.3 Fields

Field	Function
31-0 SW_REFER8_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 8

14.2.5.1.211 Base address MSB (bits 63:32) for reference compress luminance table index 9 (SWREG209)

14.2.5.1.211.1 Offset

Register	Offset
SWREG209	344h

14.2.5.1.211.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.211.3 Fields

Field	Function
31-0 SW_REFER9_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 9

14.2.5.1.212 Base address LSB (bits 31:0) for reference compress luminance table index 9 (SWREG210)

14.2.5.1.212.1 Offset

Register	Offset
SWREG210	348h

14.2.5.1.212.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.212.3 Fields

Field	Function
31-0 SW_REFER9_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 9

14.2.5.1.213 Base address MSB (bits 63:32) for reference compress luminance table index 10 (SWREG211)

14.2.5.1.213.1 Offset

Register	Offset
SWREG211	34Ch

14.2.5.1.213.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.213.3 Fields

Field	Function
31-0 SW_REFER10_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 10

14.2.5.1.214 Base address LSB (bits 31:0) for reference compress luminance table index 10 (SWREG212)

14.2.5.1.214.1 Offset

Register	Offset
SWREG212	350h

14.2.5.1.214.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.214.3 Fields

Field	Function
31-0 SW_REFER10_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 10

14.2.5.1.215 Base address MSB (bits 63:32) for reference compress luminance table index 11 (SWREG213)

14.2.5.1.215.1 Offset

Register	Offset
SWREG213	354h

14.2.5.1.215.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.215.3 Fields

Field	Function
31-0 SW_REFER11_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 11

14.2.5.1.216 Base address LSB (bits 31:0) for reference compress luminance table index 11 (SWREG214)

14.2.5.1.216.1 Offset

Register	Offset
SWREG214	358h

14.2.5.1.216.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.216.3 Fields

Field	Function
31-0 SW_REFER11_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 11

14.2.5.1.217 Base address MSB (bits 63:32) for reference compress luminance table index 12 (SWREG215)

14.2.5.1.217.1 Offset

Register	Offset
SWREG215	35Ch

14.2.5.1.217.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.217.3 Fields

Field	Function
31-0 SW_REFER12_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 12

14.2.5.1.218 Base address LSB (bits 31:0) for reference compress luminance table index 12 (SWREG216)

14.2.5.1.218.1 Offset

Register	Offset
SWREG216	360h

14.2.5.1.218.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.218.3 Fields

Field	Function
31-0 SW_REFER12_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 12

14.2.5.1.219 Base address MSB (bits 63:32) for reference compress luminance table index 13 (SWREG217)

14.2.5.1.219.1 Offset

Register	Offset
SWREG217	364h

14.2.5.1.219.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.219.3 Fields

Field	Function
31-0 SW_REFER13_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 13

14.2.5.1.220 Base address LSB (bits 31:0) for reference compress luminance table index 13 (SWREG218)

14.2.5.1.220.1 Offset

Register	Offset
SWREG218	368h

14.2.5.1.220.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.220.3 Fields

Field	Function
31-0 SW_REFER13_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 13

14.2.5.1.221 Base address MSB (bits 63:32) for reference compress luminance table index 14 (SWREG219)

14.2.5.1.221.1 Offset

Register	Offset
SWREG219	36Ch

14.2.5.1.221.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.221.3 Fields

Field	Function
31-0 SW_REFER14_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 14

14.2.5.1.222 Base address LSB (bits 31:0) for reference compress luminance table index 14 (SWREG220)

14.2.5.1.222.1 Offset

Register	Offset
SWREG220	370h

14.2.5.1.222.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.222.3 Fields

Field	Function
31-0 SW_REFER14_TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 14

14.2.5.1.223 Base address MSB (bits 63:32) for reference compress luminance table index 15 (SWREG221)

14.2.5.1.223.1 Offset

Register	Offset
SWREG221	374h

14.2.5.1.223.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TYBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.223.3 Fields

Field	Function
31-0 SW_REFER15_TYBASE_MSB	Base address MSB (bits 63:32) for reference compress luminance table index 15

14.2.5.1.224 Base address LSB (bits 31:0) for reference compress luminance table index 15 (SWREG222)

14.2.5.1.224.1 Offset

Register	Offset
SWREG222	378h

14.2.5.1.224.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TYBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.224.3 Fields

Field	Function
31-0 SW_REFER15_ TYBASE_LSB	Base address LSB (bits 31:0) for reference compress luminance table index 15

14.2.5.1.225 Base address MSB (bits 63:32) for decoder output compress chrominance table (SWREG223)

14.2.5.1.225.1 Offset

Register	Offset
SWREG223	37Ch

14.2.5.1.225.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.225.3 Fields

Field	Function
31-0 SW_DEC_OUT _TCBASE_MSB	Base address MSB (bits 63:32) for decoder output compress chrominance table

14.2.5.1.226 Base address LSB (bits 31:0) for decoder output compress chrominance table (SWREG224)

14.2.5.1.226.1 Offset

Register	Offset
SWREG224	380h

14.2.5.1.226.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_DEC_OUT_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_DEC_OUT_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.226.3 Fields

Field	Function
31-0 SW_DEC_OUT_TCBASE_LSB	Base address LSB (bits 31:0) for decoder output compress chrominance table

14.2.5.1.227 Base address MSB (bits 63:32) for reference compress chrominance table index 0 (SWREG225)

14.2.5.1.227.1 Offset

Register	Offset
SWREG225	384h

14.2.5.1.227.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.227.3 Fields

Field	Function
31-0 SW_REFER0_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 0

14.2.5.1.228 Base address LSB (bits 31:0) for reference compress chrominance table index 0 (SWREG226)

14.2.5.1.228.1 Offset

Register	Offset
SWREG226	388h

14.2.5.1.228.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER0_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER0_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.228.3 Fields

Field	Function
31-0 SW_REFER0_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 0

14.2.5.1.229 Base address MSB (bits 63:32) for reference compress chrominance table index 1 (SWREG227)

14.2.5.1.229.1 Offset

Register	Offset
SWREG227	38Ch

14.2.5.1.229.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.229.3 Fields

Field	Function
31-0 SW_REFER1_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 1

14.2.5.1.230 Base address LSB (bits 31:0) for reference compress chrominance table index 1 (SWREG228)

14.2.5.1.230.1 Offset

Register	Offset
SWREG228	390h

14.2.5.1.230.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER1_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER1_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.230.3 Fields

Field	Function
31-0 SW_REFER1_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 1

14.2.5.1.231 Base address MSB (bits 63:32) for reference compress chrominance table index 2 (SWREG229)

14.2.5.1.231.1 Offset

Register	Offset
SWREG229	394h

14.2.5.1.231.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.231.3 Fields

Field	Function
31-0 SW_REFER2_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 2

14.2.5.1.232 Base address LSB (bits 31:0) for reference compress chrominance table index 2 (SWREG230)

14.2.5.1.232.1 Offset

Register	Offset
SWREG230	398h

14.2.5.1.232.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER2_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER2_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.232.3 Fields

Field	Function
31-0 SW_REFER2_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 2

14.2.5.1.233 Base address MSB (bits 63:32) for reference compress chrominance table index 3 (SWREG231)

14.2.5.1.233.1 Offset

Register	Offset
SWREG231	39Ch

14.2.5.1.233.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.233.3 Fields

Field	Function
31-0 SW_REFER3_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 3

14.2.5.1.234 Base address LSB (bits 31:0) for reference compress chrominance table index 3 (SWREG232)

14.2.5.1.234.1 Offset

Register	Offset
SWREG232	3A0h

14.2.5.1.234.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER3_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER3_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.234.3 Fields

Field	Function
31-0 SW_REFER3_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 3

14.2.5.1.235 Base address MSB (bits 63:32) for reference compress chrominance table index 4 (SWREG233)

14.2.5.1.235.1 Offset

Register	Offset
SWREG233	3A4h

14.2.5.1.235.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.235.3 Fields

Field	Function
31-0 SW_REFER4_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 4

14.2.5.1.236 Base address LSB (bits 31:0) for reference compress chrominance table index 4 (SWREG234)

14.2.5.1.236.1 Offset

Register	Offset
SWREG234	3A8h

14.2.5.1.236.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER4_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER4_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.236.3 Fields

Field	Function
31-0 SW_REFER4_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 4

14.2.5.1.237 Base address MSB (bits 63:32) for reference compress chrominance table index 5 (SWREG235)

14.2.5.1.237.1 Offset

Register	Offset
SWREG235	3ACh

14.2.5.1.237.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.237.3 Fields

Field	Function
31-0 SW_REFER5_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 5

14.2.5.1.238 Base address LSB (bits 31:0) for reference compress chrominance table index 5 (SWREG236)

14.2.5.1.238.1 Offset

Register	Offset
SWREG236	3B0h

14.2.5.1.238.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER5_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER5_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.238.3 Fields

Field	Function
31-0 SW_REFER5_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 5

14.2.5.1.239 Base address MSB (bits 63:32) for reference compress chrominance table index 6 (SWREG237)

14.2.5.1.239.1 Offset

Register	Offset
SWREG237	3B4h

14.2.5.1.239.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.239.3 Fields

Field	Function
31-0 SW_REFER6_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 6

14.2.5.1.240 Base address LSB (bits 31:0) for reference compress chrominance table index 6 (SWREG238)

14.2.5.1.240.1 Offset

Register	Offset
SWREG238	3B8h

14.2.5.1.240.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER6_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER6_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.240.3 Fields

Field	Function
31-0 SW_REFER6_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 6

14.2.5.1.241 Base address MSB (bits 63:32) for reference compress chrominance table index 7 (SWREG239)

14.2.5.1.241.1 Offset

Register	Offset
SWREG239	3BCh

14.2.5.1.241.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.241.3 Fields

Field	Function
31-0 SW_REFER7_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 7

14.2.5.1.242 Base address LSB (bits 31:0) for reference compress chrominance table index 7 (SWREG240)

14.2.5.1.242.1 Offset

Register	Offset
SWREG240	3C0h

14.2.5.1.242.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER7_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER7_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.242.3 Fields

Field	Function
31-0 SW_REFER7_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 7

14.2.5.1.243 Base address MSB (bits 63:32) for reference compress chrominance table index 8 (SWREG241)

14.2.5.1.243.1 Offset

Register	Offset
SWREG241	3C4h

14.2.5.1.243.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.243.3 Fields

Field	Function
31-0 SW_REFER8_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 8

14.2.5.1.244 Base address LSB (bits 31:0) for reference compress chrominance table index 8 (SWREG242)

14.2.5.1.244.1 Offset

Register	Offset
SWREG242	3C8h

14.2.5.1.244.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER8_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER8_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.244.3 Fields

Field	Function
31-0 SW_REFER8_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 8

14.2.5.1.245 Base address MSB (bits 63:32) for reference compress chrominance table index 9 (SWREG243)

14.2.5.1.245.1 Offset

Register	Offset
SWREG243	3CCh

14.2.5.1.245.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.245.3 Fields

Field	Function
31-0 SW_REFER9_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 9

14.2.5.1.246 Base address LSB (bits 31:0) for reference compress chrominance table index 9 (SWREG244)

14.2.5.1.246.1 Offset

Register	Offset
SWREG244	3D0h

14.2.5.1.246.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER9_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER9_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.246.3 Fields

Field	Function
31-0 SW_REFER9_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 9

14.2.5.1.247 Base address MSB (bits 63:32) for reference compress chrominance table index 10 (SWREG245)

14.2.5.1.247.1 Offset

Register	Offset
SWREG245	3D4h

14.2.5.1.247.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.247.3 Fields

Field	Function
31-0 SW_REFER10_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 10

14.2.5.1.248 Base address LSB (bits 31:0) for reference compress chrominance table index 10 (SWREG246)

14.2.5.1.248.1 Offset

Register	Offset
SWREG246	3D8h

14.2.5.1.248.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER10_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER10_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.248.3 Fields

Field	Function
31-0 SW_REFER10_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 10

14.2.5.1.249 Base address MSB (bits 63:32) for reference compress chrominance table index 11 (SWREG247)

14.2.5.1.249.1 Offset

Register	Offset
SWREG247	3DCh

14.2.5.1.249.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.249.3 Fields

Field	Function
31-0 SW_REFER11_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 11

14.2.5.1.250 Base address LSB (bits 31:0) for reference compress chrominance table index 11 (SWREG248)

14.2.5.1.250.1 Offset

Register	Offset
SWREG248	3E0h

14.2.5.1.250.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER11_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER11_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.250.3 Fields

Field	Function
31-0 SW_REFER11_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 11

14.2.5.1.251 Base address MSB (bits 63:32) for reference compress chrominance table index 12 (SWREG249)

14.2.5.1.251.1 Offset

Register	Offset
SWREG249	3E4h

14.2.5.1.251.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.251.3 Fields

Field	Function
31-0 SW_REFER12_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 12

14.2.5.1.252 Base address LSB (bits 31:0) for reference compress chrominance table index 12 (SWREG250)

14.2.5.1.252.1 Offset

Register	Offset
SWREG250	3E8h

14.2.5.1.252.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER12_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER12_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.252.3 Fields

Field	Function
31-0 SW_REFER12_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 12

14.2.5.1.253 Base address MSB (bits 63:32) for reference compress chrominance table index 13 (SWREG251)

14.2.5.1.253.1 Offset

Register	Offset
SWREG251	3ECh

14.2.5.1.253.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.253.3 Fields

Field	Function
31-0 SW_REFER13_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 13

14.2.5.1.254 Base address LSB (bits 31:0) for reference compress chrominance table index 13 (SWREG252)

14.2.5.1.254.1 Offset

Register	Offset
SWREG252	3F0h

14.2.5.1.254.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER13_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER13_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.254.3 Fields

Field	Function
31-0 SW_REFER13_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 13

14.2.5.1.255 Base address MSB (bits 63:32) for reference compress chrominance table index 14 (SWREG253)

14.2.5.1.255.1 Offset

Register	Offset
SWREG253	3F4h

14.2.5.1.255.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.255.3 Fields

Field	Function
31-0 SW_REFER14_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 14

14.2.5.1.256 Base address LSB (bits 31:0) for reference compress chrominance table index 14 (SWREG254)

14.2.5.1.256.1 Offset

Register	Offset
SWREG254	3F8h

14.2.5.1.256.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER14_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER14_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.256.3 Fields

Field	Function
31-0 SW_REFER14_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 14

14.2.5.1.257 Base address MSB (bits 63:32) for reference compress chrominance table index 15 (SWREG255)

14.2.5.1.257.1 Offset

Register	Offset
SWREG255	3FCh

14.2.5.1.257.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TCBASE_MSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.257.3 Fields

Field	Function
31-0 SW_REFER15_TCBASE_MSB	Base address MSB (bits 63:32) for reference compress chrominance table index 15

14.2.5.1.258 Base address LSB (bits 31:0) for reference compress chrominance table index 15 (SWREG256)

14.2.5.1.258.1 Offset

Register	Offset
SWREG256	400h

14.2.5.1.258.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_REFER15_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_REFER15_TCBASE_LSB															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.258.3 Fields

Field	Function
31-0 SW_REFER15_TCBASE_LSB	Base address LSB (bits 31:0) for reference compress chrominance table index 15

14.2.5.1.259 Not used (SWREG257)

14.2.5.1.259.1 Offset

Register	Offset
SWREG257	404h

14.2.5.1.259.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.259.3 Fields

Field	Function
31-0 —	Reserved.

14.2.5.1.260 input stream buffer length (SWREG258)

14.2.5.1.260.1 Offset

Register	Offset
SWREG258	408h

14.2.5.1.260.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STRM_BUFFER_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STRM_BUFFER_LEN															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.260.3 Fields

Field	Function
31-0 SW_STRM_BUFFER_LEN	input stream buffer length

14.2.5.1.261 input stream buffer start offset (SWREG259)

14.2.5.1.261.1 Offset

Register	Offset
SWREG259	40Ch

14.2.5.1.261.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SW_STRM_START_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SW_STRM_START_OFFSET															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

14.2.5.1.261.3 Fields

Field	Function
31-0 SW_STRM_START_OFFSET	input stream buffer start offset

Chapter 15
Display Controller Subsystem (DCSS)

15.1 Display Controller Subsystem

15.1.1 Overview

The Display Controller Subsystem (DCSS) is used to source up to three display buffers, compose them, and drive a display using HDMI 2.0a with HDCP 2.2. The DCSS is intended to support up to 4kp60 displays. HDR10 image processing capabilities are included to provide a solution capable of driving next generation high dynamic range displays.

15.1.2 Features

The DCSS supports two major modes: HDR10 and Dolby Vision. These modes are mutually exclusive modes of operation. In HDR10 mode, there are 3 identical pipelines for video or graphics. In Dolby Vision mode, there are two pipelines, the Video pipeline (consists of 2 streams, base and enhanced layers) and the Overlay pipeline. The features of the DCSS include support for the following capabilities.

- Up to 4Kp60 video/graphics display over HDMI 2.0a with HDCP 2.2 encryption and audio formats including Dolby Digital, DTS, TrueHD, LPCM
- On-the-fly image composition of up to 3 source display buffers.
  - PIG (picture in GFX) overlay on video
  - PIP (picture in picture) scaled video on video with graphics
- On-the-fly decompression of compressed video or graphics frame buffers to optimize DRAM bandwidth.
- On-the-fly conversion of tile frame buffers to raster for display refresh.
- Integrated memory to store pre-fetched pixel data to manage DRAM access latencies.
- 3 independent 7x7 tap scaling to process 3 display buffers.

    - Supports x7 up and x1/7 down scaling ratios, UHD/HD/SD/PIP
    - PIP (picture in picture) scaled video on video with graphics
- Dolby or HDR10 image processing (see below for detailed feature list). These modes are mutually exclusive.
- Scaled output to DDR write back for 1 of three scaler engines (programmable). Used for large ratio downscaled images to manage dataflow from DDR and real time display processing.
- Read bus master to source raster frames from memory with direct input into Dolby and HDR10 pixel processing data pipes (bypass buffer conversion and scaling HW).
- CEA861 compatible display timing generator.
- Display state controller to manage display context changes.
- HDMI supporting HDCP output interface.
- YUV420/422/444 and ARGB with 8/10-bit per component source buffer formats

15.1.2.1 HDR10 Pixel Processing

In HDR10 mode, all three pixel processing pipelines are identical.

Each HDR10 pixel processing input pipeline supports
- 10-bit per component input
- Color space conversion
- LUT gamma correction per component

HDR10 Output Pipeline The HDR10 pixel processing output pipeline supports
- Linear to non-linear conversion per component
- Color space conversion
- 10-bit per component RGB/YUV444

15.1.2.2 Dolby Vision Pixel Processing

When the DCSS is configured to operate in Dolby Vision Mode, the follow capabilities are supported.

Dolby Vision Video pixel processing features supported.

- Algorithms
    - Dolby Vision base and enhanced layer processing and composition
    - Dolby Vision color space conversion
    - Dolby Vision color volume mapping
- Metadata and Tone Curve information processing.

- 422 10-bit pixel input
- Bypass supported for non-Dolby content

Dolby Overlay(Graphics) pixel processing features supported.

- Algorithms
  - Dolby Vision Input color volume conversion for overlay
  - Color space conversion
- Bypass supported for non-Dolby content
- Support for RGB and YUV422/444 8-bit pixel formats
- Tone Curve information processing

The overlay pipeline supports graphics and/or video overlay pixel processing. The video and graphics overlay pixels are composited at the output of these two Dolby pixel processing pipes. The overlay buffer can contain any content preprocessed by the SoC graphics accelerator or general purpose microprocessor. This can include preprocessed video and graphics composited buffers.

Dolby Output pixel processing features supported.

- Algorithms
  - Dolby Vision output color space mapping
  - Video+metadata scrambling
  - Dithering for 8/10 bit YUV output
- Output formats
  - Bypassed 8-bit RGB
  - 8/10-bit UV444

15.1.3 Use Cases

Some of the use cases the DCSS supports are decribed here:

- Video only – This can be optimized for power by turning off the CPUs and graphics and unused VPU decoders
- Graphics only - Supports GUI or gaming. This can be optimized for power by turning off the VPU
- Video with graphics overlay – Per pixel alpha blending supported
- PIG (picture in graphics) - Graphics with PIP video in corner
- PIP (picture in picture) – Full screen video with PIP video in corner
- Video with PIP and graphics – Full screen video with PIP video in corner and graphics overlay

Since the DCSS is a programmable display engine, the use cases are not limited to the cases listed above.

15.1.4 DCSS Block Diagram

The first diagram below contains the front end of the DCSS datapath, including all the busmaster components, decompression blocks, resolve memory, and scaler. All components feed the pixel processing components later in the DCSS pipeline.

The second diagram below contains the display pipeline, including the mutually exclusive Dolby and HDR10 processing pixel pipes, alpha blender, color subsampler, and HDMI controller and PHY.

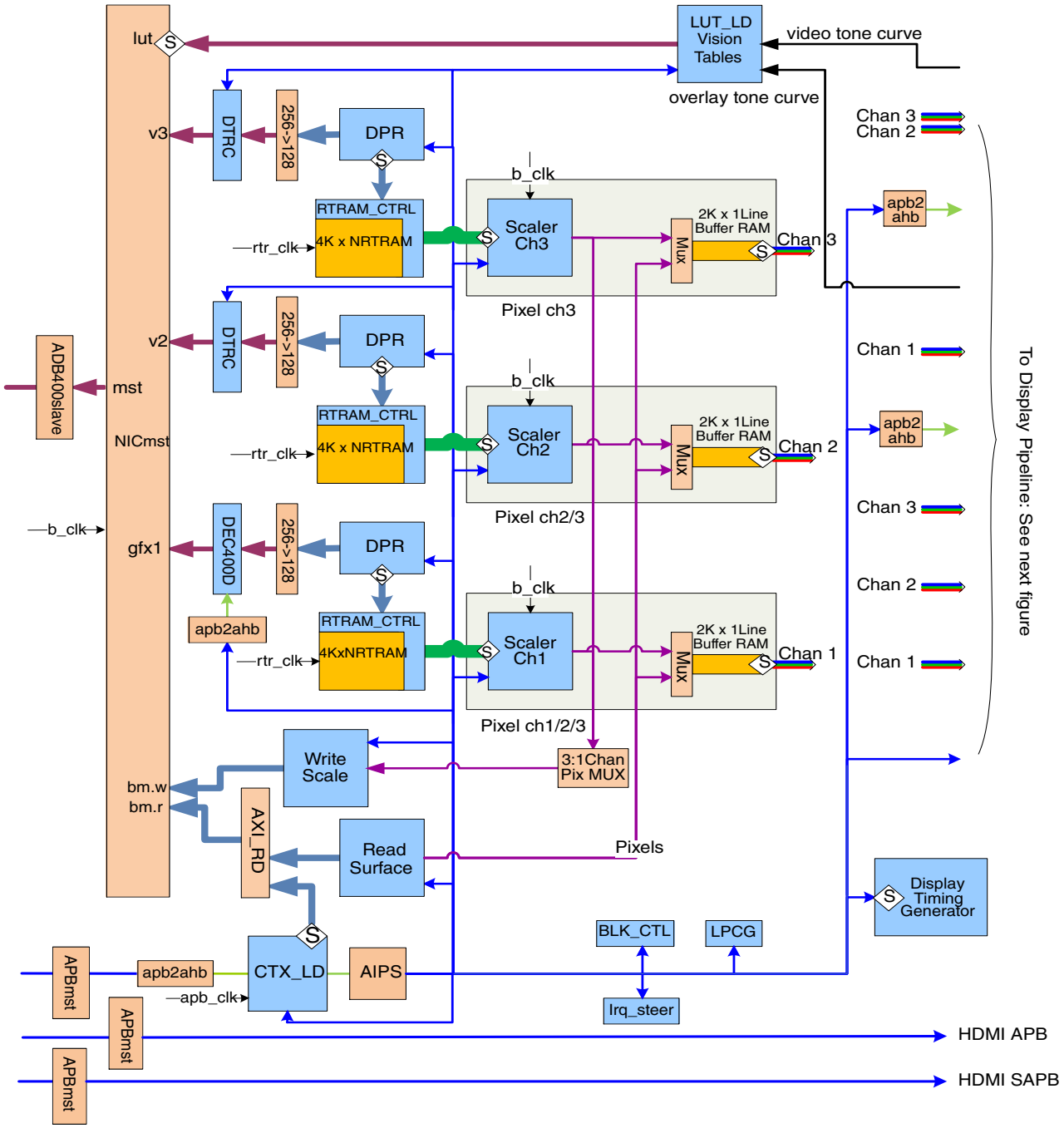


Figure 15-1. DCSS Front End Block Diagram

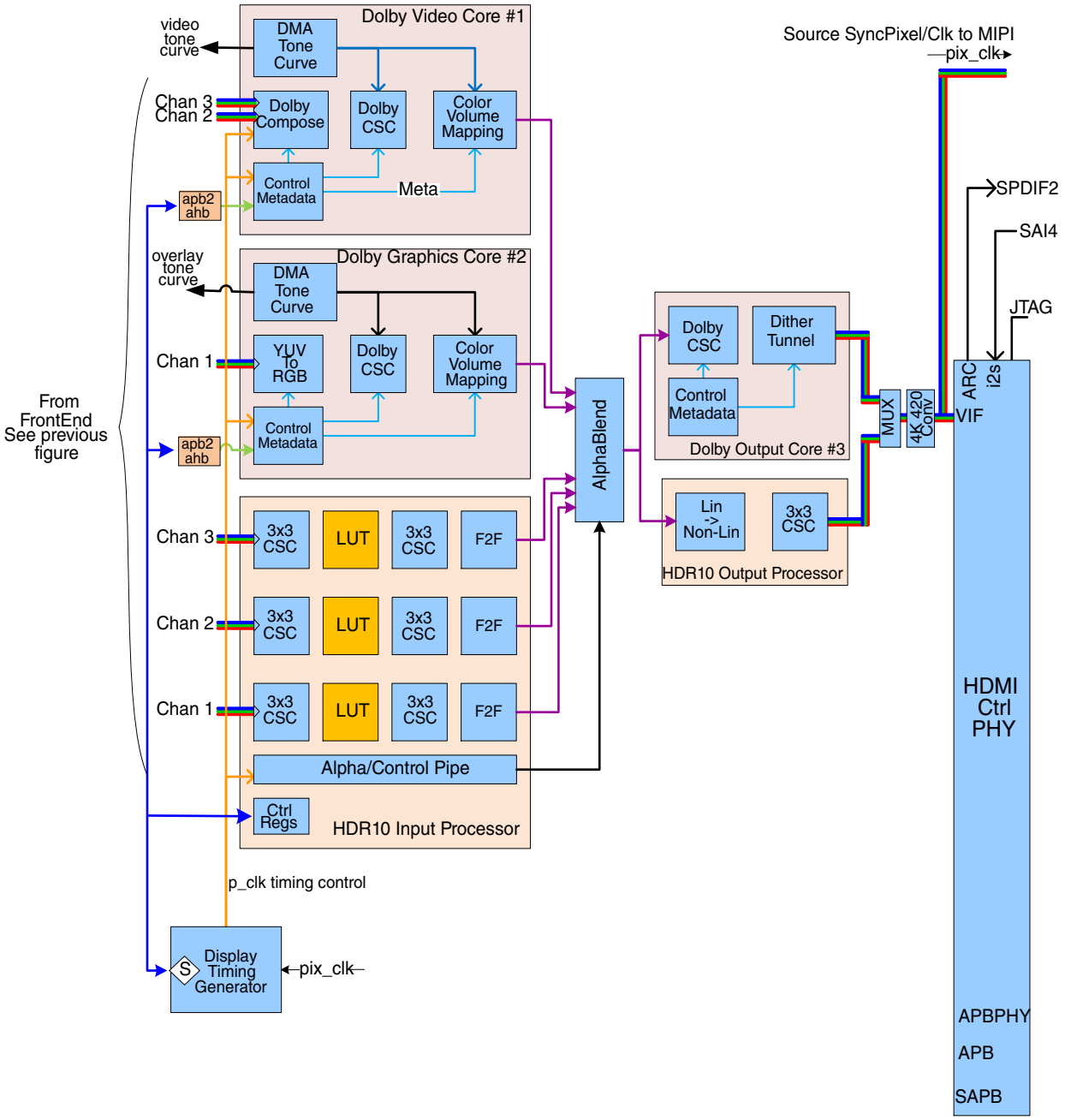


Figure 15-2. DCSS Pixel Processing Block Diagram

15.1.5 Memory Map and register definition

The HDMI controller has independent APB interfaces that are separate from the DCSS APB input bus. There is one HDMI APB for general purpose APB access, and another for secure SAPB access. Since they are separate DCSS input busses, their base address in the system is dependent on the address decoding outside of the DCSS. Refer to the top level memory map definition for the system.

The HDMI.APB interface uses 256KB of address space.

The HDMI.SAPB interface uses 64KB of address space.

The DCSS address space is configured as shown. All addresses are relative to the base address of the DCSS in the system. This memory map defines the offsets for all components from the base address which is defined outside of the DCSS.

Table 15-1. DCSS Memory Map

Start Address	End Address	Size	Allocation
0002_F000	0002_FFFF	4KB	BLK_CTL
0002_E000	0002_EFFF	4KB	LPCG See Note 1.
0002_D000	0002_DFFF	4KB	IRQ_STEER
0002_C000	0002_CFFF	4KB	HDMI_PHY registers
		28KB	Reserved
0002_4000	0002_4FFF	4KB	LUT_LD
0002_3000	0002_3FFF	4KB	CTX_LD
0002_2000	0002_2FFF	4KB	RD_SRC
0002_1000	0002_1FFF	4KB	WR_SCL
0002_0000	0002_0FFF	4KB	DTG
		12KB	Reserved
0001_C000	0001_CFFF	4KB	Scaler (All Channels 3 in 1)
0001_B000	0001_BFFF	4KB	SUBSAM
0001_A000	0001_AFFF	4KB	DPR-chan3
0001_9000	0001_9FFF	4KB	DPR-chan2
0001_8000	0001_8FFF	4KB	DPR-chan1
0001_7000	0001_7FFF	4KB	DTRC-chan3
0001_6000	0001_7FFF	4KB	DTRC-chan2
0001_5000	0001_7FFF	4KB	DEC400D-chan1
		8KB	Reserved
0001_2000	0001_2FFF	4KB	Dolby Vision Output (Core3)
0001_1000	0001_1FFF	4KB	Dolby Vision Graphics (Core2)
0001_0000	0001_0FFF	4KB	Dolby Vision Video (Core1)
0000_C000	0000_FFFF	16KB	HDR10 Tables and Registers
0000_8000	0000_BFFF	16KB	HDR10 Chan3 LUT
0000_4000	0000_7FFF	16KB	HDR10 Chan2 LUT
0000_0000	0000_3FFF	16KB	HDR10 Chan1 LUT

Note 1: The LPCG registers are at the base address in the table above. Each LPCG(i) register is located at an offset of i*4 bytes from the LPCG base address.

15.1.6 Functional description

At a high level, the DCSS consists of either three separate pipelines in HDR10 mode or two separate pipelines in Dolby Vision mode. These sources are composed to drive a single output display image.

The front end of the DCSS connects to DDR memory through an ADB400 slave interface with three decompression, DPR DMA channels, and three scalers. These componenent groups complete three separate channels for sourcing independent display buffers for display. Accesses for frame buffer data and control information is arbitrated by the bus interconnect component. There is also one write back channel and one read back channel, all clocked with the 800 MHz system bus clock. The display timing generator and the context loader work congruently to provide the overall DCSS control for updating the display between frame sequences. The pixel processing components (Dolby or HDR10) of the DCSS runs with a single display resolution and a single pixel clock. An HDMI 2.0 compliant controller and PHY complete the source solution for driving next generation HDR UHD displays with compatibility for current non-HDR HD displays.

The two operation modes, Dolby and HDR10, are mutually exclusive.

15.1.6.1 Operating Modes

15.1.6.1.1 Dolby Vision Mode Operation

In Dolby Vision mode, the video pipeline and the overlay pipeline get blended together, post processed and sent to the HDMI protocol controller. The HDMI output consists of a display window containing a video window and an overlay window. Both the video and overlay windows can be driven by the read back channel. Both the video and overlay windows can be full screen or smaller and offset inside the display window with the rest of the display window filled with the background color. The compositing of the video and overlay windows is controlled by the per pixel alpha channel. The overlay path creates an alpha based on the window priority and the graphics per pixel alpha. The diagram below shows a possible window configuration.

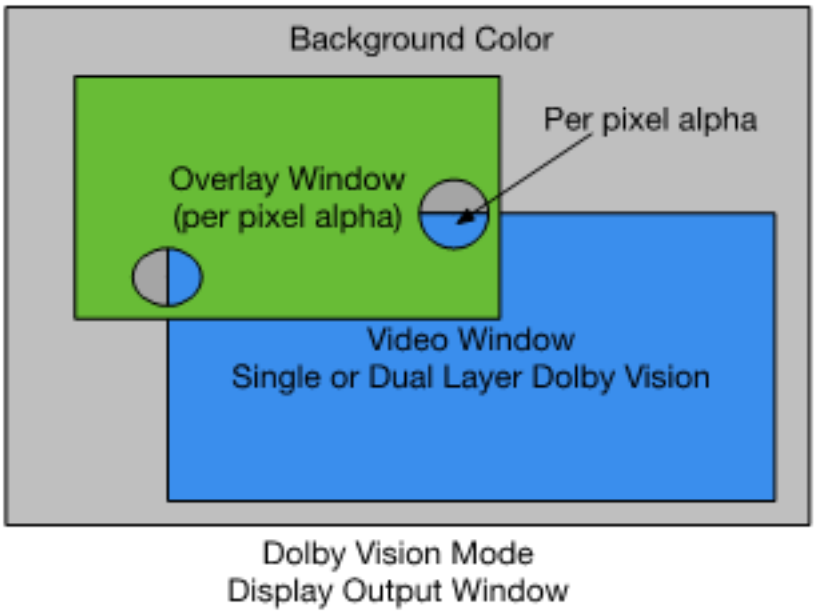


Figure 15-3. Dolby Processed Display

15.1.6.1.2 HDR10 Mode Operation

In HDR10 mode, the three inputs get merged or blended together, post processed and sent to the HDMI protocol controller. The HDMI output consists of a display window containing up to three input windows. Any of the three windows can by driven by the Readback channel. All three windows can be full screen or smaller and offset inside the display window with the rest of the display window filled with the background color. The compositing of the three windows is controlled by the per pixel alpha channel from channel 1. The alpha logic creates an alpha based on the window priority and the graphics per pixel alpha. The diagram below shows a possible window configuration. Note that window priority is fixed with channel 1 on top of channel 2 on top of channel 3. Only channel 1 can have a per pixel alpha.

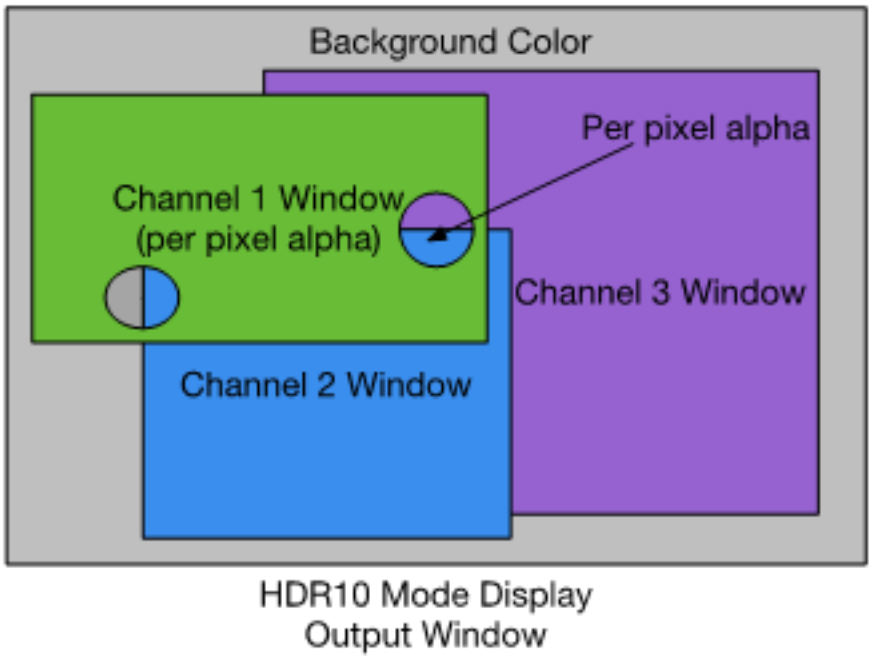


Figure 15-4. HDR10 Processed Display

15.1.6.2 Clocks

Clocks used in the DCSS are either sourced from the top level (see CCM) or the integrated HDMI PHY. The fundamental clock frequencies required to clock the DCSS are listed in the table below.

Table 15-2. DCSS Interrupt Mapping Table

Clock Domain	DCSS Clocked Components	Max Frequency
b_clk	NIC, DPR, RTRAM_CTRL, Scaler, etc	800MHz
rtr_clk	RTRAM_CTRL, DPR, Scaler. NOTE: the following condition must be met: b_clk/2 .lt. rtr_clk	400MHz
apb_clk	All peripheral PIO access interfaces. All APB and core clocks for the HDMI controller.	133MHz
ss_refclock	Reference clock for HDMI PLL. This reference source clock is optional since the refclk_pln pins are the primary source for the reference clock for the PHY.	27MHz
p_clk	This is driven by the phy_pma_pixel_clk_out port of the PHY	max 594MHz

Table 15-2. DCSS Interrupt Mapping Table

Clock Domain	DCSS Clocked Components	Max Frequency
	(in HDMI mode), or the top level pixel clock input (in MIPI mode). Dolby Vision, HDR10, HDMI_CTRL, SUBSAM	This clock is programmed to the correct frequency to drive the desired display resolution and refresh rate. Refer to the CEA861 display timing specification for HDMI compliant pixel clock frequencies.

15.1.6.3 Interrupts

There are many sources for interrupts within the DCSS. The IRQ_STEER module is used to collect interrupts in the DC. SW can read the IRQ_STEER status bits to determine which DC interrupt sources are active. The IRQ_STEER provides the first level resource to determine the interrupt source. Refer to the chapter related to the IRQ_STEER module for detailed operation related to interrupt steering.

15.1.6.3.1 DCSS Interrupt Outputs

The DCSS dcss_int_out[0] is driven by internal_irq logically ANDed with IRQSTEER_CH0_MINTDIS[0] register. The DCSS dcss_int_out[1] is driven by the internal_irq directly. This bit is logically equal to dcss_int_out[0], with the exception that it is not gated with the IRQSTEER_CH0_MINTDIS[0] register. The DCSS dcss_int_out[2] is not routed through the IRQ_STEER module, but is driven directly from the HDMI PHYs source_hpd_out signal. This is the HDMI hot plug detect signal.

15.1.6.3.2 DTG General Purpose Interrupt Allocation

There are 8 general purpose interrupt collected by the Display Timing Generator (DTG). Refer to the DTG Interrupt Generation section for a description of the general purpose interrupt generator. The table below indicates the DCSS level allocation of the 8 interrupts implemented in the DTG.

Table 15-3. DTG Interrupt Map

DTG IRQ Input	Signal Source to generate interrupt
0	Panic Channel 0 – This is the panic signal output from the RTRAM_CTRL for channel 0.
1	Panic Channel 1 – This is the panic signal output from the RTRAM_CTRL for channel 1.
2	Panic Channel 2 – This is the panic signal output from the RTRAM_CTRL for channel 2.
3-7	reserved

15.1.6.3.3 DCSS Interrupt Mapping

The mapping of interrupts into the IRQ_STEER are defined by the following table. Details for the interrupt functionality including status and mask can be obtained from description of the respective interrupt registers within the source modules.

Table 15-4. DCSS Interrupt Mapping Table

IRQ Steer Interrupt Number	Interrupt Source
0	dolby_video_core
1	dolby_output_core
2	dolby_graphics_core
3	dpr_dc_ch1
4	dpr_dc_ch2
5	dpr_dc_ch3
6	ctx_ld
7	rd_src
8	dtg_programmable_1
9	dtg_programmable_2
10	dtg_programmable_3
11	dtg_programmable_4
12	dtg_dcss_misc_8to1
13	hdmi_sirq
14	hdmi_pirq
15	dec400d_ch1
16	dtrc_ch2
17	dtrc_ch3
18	lut_ld
19	wr_scl
20	scaler_fifo_ch1_high_threshold_detect
21	scaler_fifo_ch2_high_threshold_detect
22	scaler_fifo_ch3_high_threshold_detect
23	scaler_fifo_ch1_low_threshold_detect
24	scaler_fifo_ch2_low_threshold_detect
25	scaler_fifo_ch3_low_threshold_detect
26	scaler_fifo_ch1_fifo_underrun_detect
27	scaler_fifo_ch2_fifo_underrun_detect
28	scaler_fifo_ch3_fifo_underrun_detect

15.2 DCSS Block Control (BLK_CTL)

15.2.1 Overview

The DC Block Control is the Display Controller Subsystem's general purpose control register block. The control registers contain various functionality including but not limited to:

- Async reset per clock domain
- Clock muxing control
- Async reset per clock domain
- HDMI source secure

15.2.2 Memory Map and Registers

15.2.2.1 register descriptions

15.2.2.1.1 MED_DCSS_BLK_CTL Memory map

MED_DCSS_BLK_CTL base address: 2_F000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Reset Control (RESET_CTRL)	32	RW	0000_0000h
4h	Reset Control (RESET_CTRL_SET)	32	RW	0000_0000h
8h	Reset Control (RESET_CTRL_CLR)	32	RW	0000_0000h
Ch	Reset Control (RESET_CTRL_TOG)	32	RW	0000_0000h
10h	Control (CONTROL0)	32	RW	0000_0000h
14h	Control (CONTROL0_SET)	32	RW	0000_0000h
18h	Control (CONTROL0_CLR)	32	RW	0000_0000h
1Ch	Control (CONTROL0_TOG)	32	RW	0000_0000h
20h	Spare Control0 (SPARE_CTRL0)	32	RW	0000_0000h
24h	Spare Control0 (SPARE_CTRL0_SET)	32	RW	0000_0000h
28h	Spare Control0 (SPARE_CTRL0_CLR)	32	RW	0000_0000h
2Ch	Spare Control0 (SPARE_CTRL0_TOG)	32	RW	0000_0000h
30h	Spare Control1 (SPARE_CTRL1)	32	RW	0000_0000h
34h	Spare Control1 (SPARE_CTRL1_SET)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
38h	Spare Control1 (SPARE_CTRL1_CLR)	32	RW	0000_0000h
3Ch	Spare Control1 (SPARE_CTRL1_TOG)	32	RW	0000_0000h
40h	Spare Status0 (SPARE_STATUS0)	32	RO	0000_0000h
44h	Spare Status0 (SPARE_STATUS0_SET)	32	RO	0000_0000h
48h	Spare Status0 (SPARE_STATUS0_CLR)	32	RO	0000_0000h
4Ch	Spare Status0 (SPARE_STATUS0_TOG)	32	RO	0000_0000h

15.2.2.1.2 Reset Control (RESET_CTRL)

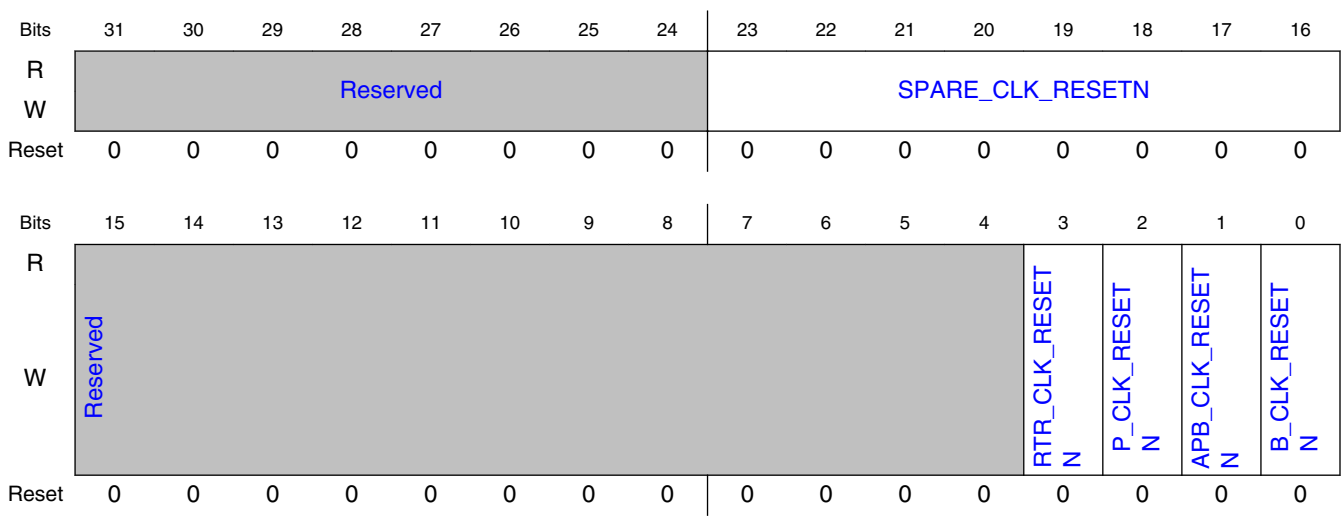
15.2.2.1.2.1 Offset

Register	Offset
RESET_CTRL	0h

15.2.2.1.2.2 Function

Asynchronous reset register control. There is one reset bit per clock domain. Each reset will be asynchronously asserted, and synchronized to synchronously deassert.

15.2.2.1.2.3 Diagram



15.2.2.1.2.4 Fields

Field	Function
31-24 —	Reserved.
23-16 SPARE_CLK_RESETN	spare clk domain resetn Spare clock domain resets, implemented for potential ECO purposes. 0x0 = assert reset 0x1 = deassert reset
15-4 —	Reserved.
3 RTR_CLK_RESETN	rtr_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
2 P_CLK_RESETN	p_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
1 APB_CLK_RESETN	apb_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
0 B_CLK_RESETN	b_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset

15.2.2.1.3 Reset Control (RESET_CTRL_SET)

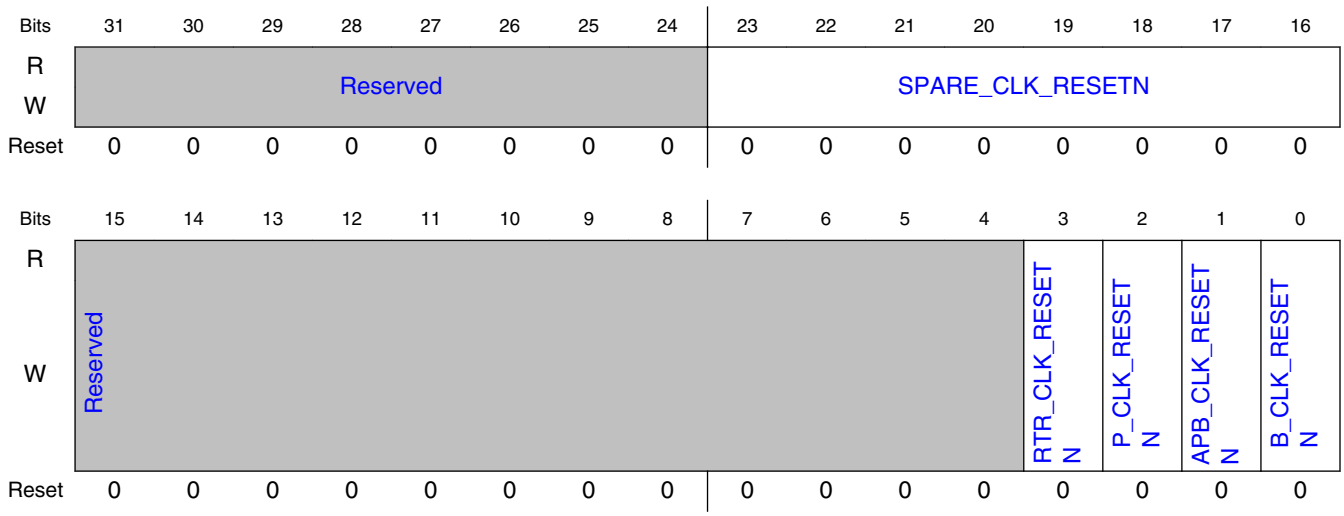
15.2.2.1.3.1 Offset

Register	Offset
RESET_CTRL_SET	4h

15.2.2.1.3.2 Function

Asynchronous reset register control. There is one reset bit per clock domain. Each reset will be asynchronously asserted, and synchronized to synchronously deassert.

15.2.2.1.3.3 Diagram



15.2.2.1.3.4 Fields

Field	Function
31-24 —	Reserved.
23-16 SPARE_CLK_RESETN	spare clk domain resetn Spare clock domain resets, implemented for potential ECO purposes. 0x0 = assert reset 0x1 = deassert reset
15-4 —	Reserved.
3 RTR_CLK_RESETN	rtr_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
2 P_CLK_RESETN	p_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
1 APB_CLK_RESETN	apb_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
0 B_CLK_RESETN	b_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset

15.2.2.1.4 Reset Control (RESET_CTRL_CLR)

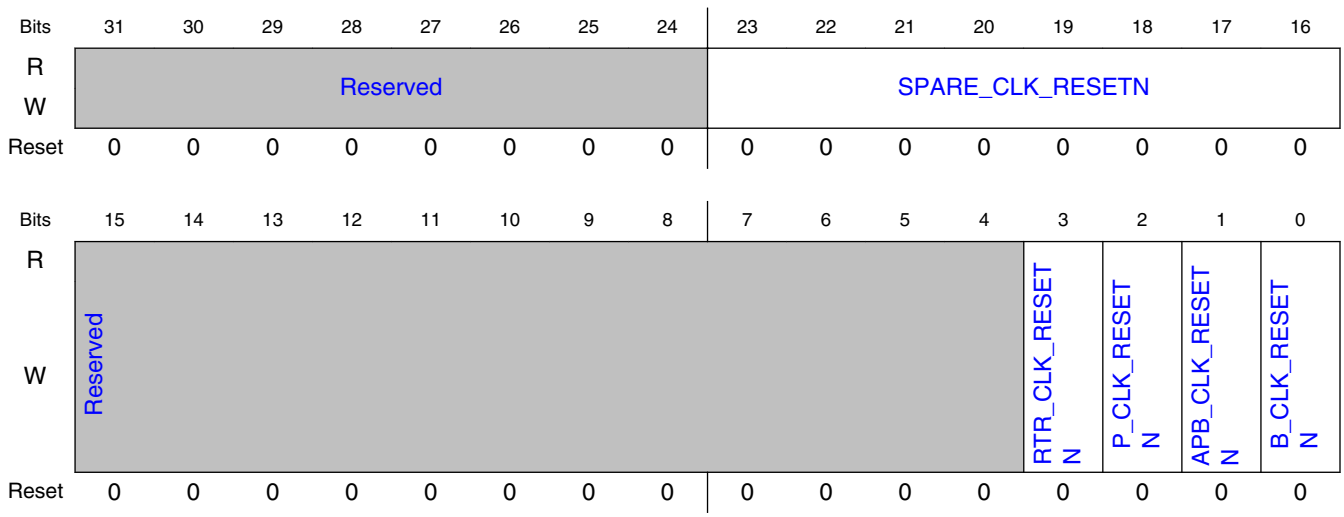
15.2.2.1.4.1 Offset

Register	Offset
RESET_CTRL_CLR	8h

15.2.2.1.4.2 Function

Asynchronous reset register control. There is one reset bit per clock domain. Each reset will be asynchronously asserted, and synchronized to synchronously deassert.

15.2.2.1.4.3 Diagram



15.2.2.1.4.4 Fields

Field	Function
31-24 —	Reserved.
23-16 SPARE_CLK_RESETN	spare clk domain resetn Spare clock domain resets, implemented for potential ECO purposes. 0x0 = assert reset 0x1 = deassert reset
15-4 —	Reserved.
3 RTR_CLK_RESETN	rtr_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
2	p_clk domain modules resetn

Table continues on the next page...

Field	Function
P_CLK_RESETN	0x0 = assert reset 0x1 = deassert reset
1 APB_CLK_RESETN	apb_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
0 B_CLK_RESETN	b_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset

15.2.2.1.5 Reset Control (RESET_CTRL_TOG)

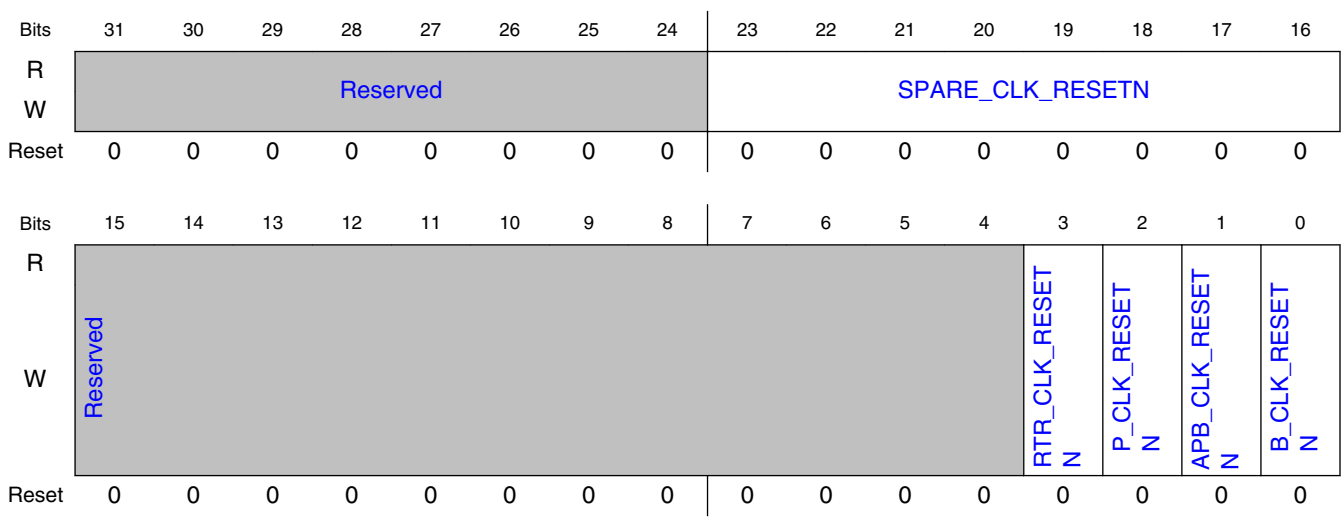
15.2.2.1.5.1 Offset

Register	Offset
RESET_CTRL_TOG	Ch

15.2.2.1.5.2 Function

Asynchronous reset register control. There is one reset bit per clock domain. Each reset will be asynchronously asserted, and synchronized to synchronously deassert.

15.2.2.1.5.3 Diagram



15.2.2.1.5.4 Fields

Field	Function
31-24 —	Reserved.
23-16 SPARE_CLK_RESETN	spare clk domain resetn Spare clock domain resets, implemented for potential ECO purposes. 0x0 = assert reset 0x1 = deassert reset
15-4 —	Reserved.
3 RTR_CLK_RESETN	rtr_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
2 P_CLK_RESETN	p_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
1 APB_CLK_RESETN	apb_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset
0 B_CLK_RESETN	b_clk domain modules resetn 0x0 = assert reset 0x1 = deassert reset

15.2.2.1.6 Control (CONTROL0)

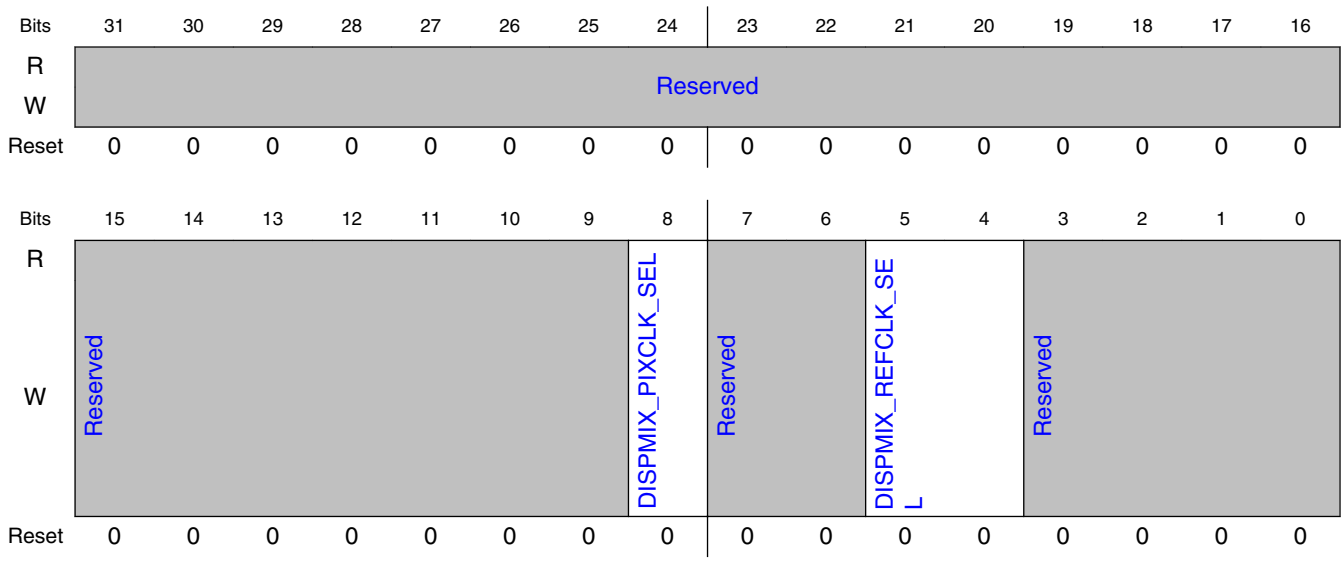
15.2.2.1.6.1 Offset

Register	Offset
CONTROL0	10h

15.2.2.1.6.2 Function

Misc. Control registers

15.2.2.1.6.3 Diagram



15.2.2.1.6.4 Fields

Field	Function
31-9 —	Reserved.
8 DISPMIX_PIXCLK_SEL	Display Subsystem Pixel Clock Select Pixel Clock source selection. 0x0 = Video PLL2 Clock 0x1 = CCM DC Pixel Clock
7-6 —	Reserved.
5-4 DISPMIX_REFCLK_SEL	Display Subsystem Reference Clock Select Reference clock source selection. 0x0 = 27 MHz Oscillator Reference Clock 0x1 = Video PLL2 Clock 0x2 = CCM DC Pixel Clock
3-0 —	Reserved.

15.2.2.1.7 Control (CONTROL0_SET)

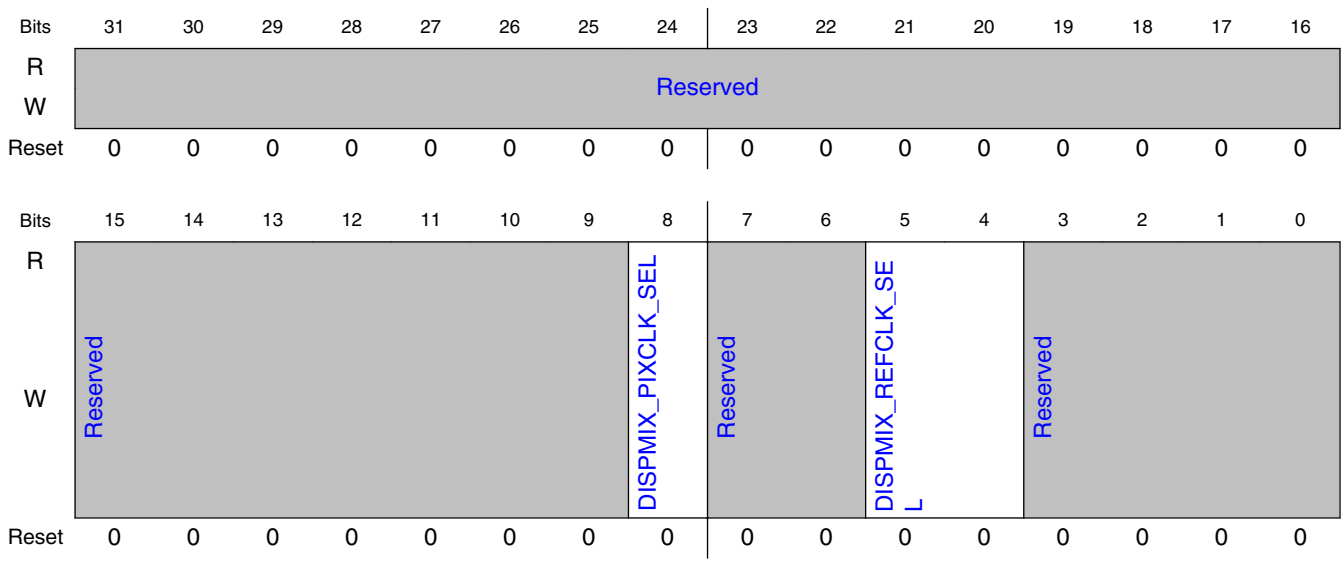
15.2.2.1.7.1 Offset

Register	Offset
CONTROL0_SET	14h

15.2.2.1.7.2 Function

Misc. Control registers

15.2.2.1.7.3 Diagram



15.2.2.1.7.4 Fields

Field	Function
31-9 —	Reserved.
8 DISPMIX_PIXCLK_SEL	Display Subsystem Pixel Clock Select Pixel Clock source selection. 0x0 = Video PLL2 Clock 0x1 = CCM DC Pixel Clock
7-6 —	Reserved.
5-4 DISPMIX_REFCLK_SEL	Display Subsystem Reference Clock Select Reference clock source selection. 0x0 = 27 MHz Oscillator Reference Clock 0x1 = Video PLL2 Clock 0x2 = CCM DC Pixel Clock

Table continues on the next page...

Field	Function
3-0 —	Reserved.

15.2.2.1.8 Control (CONTROL0_CLR)

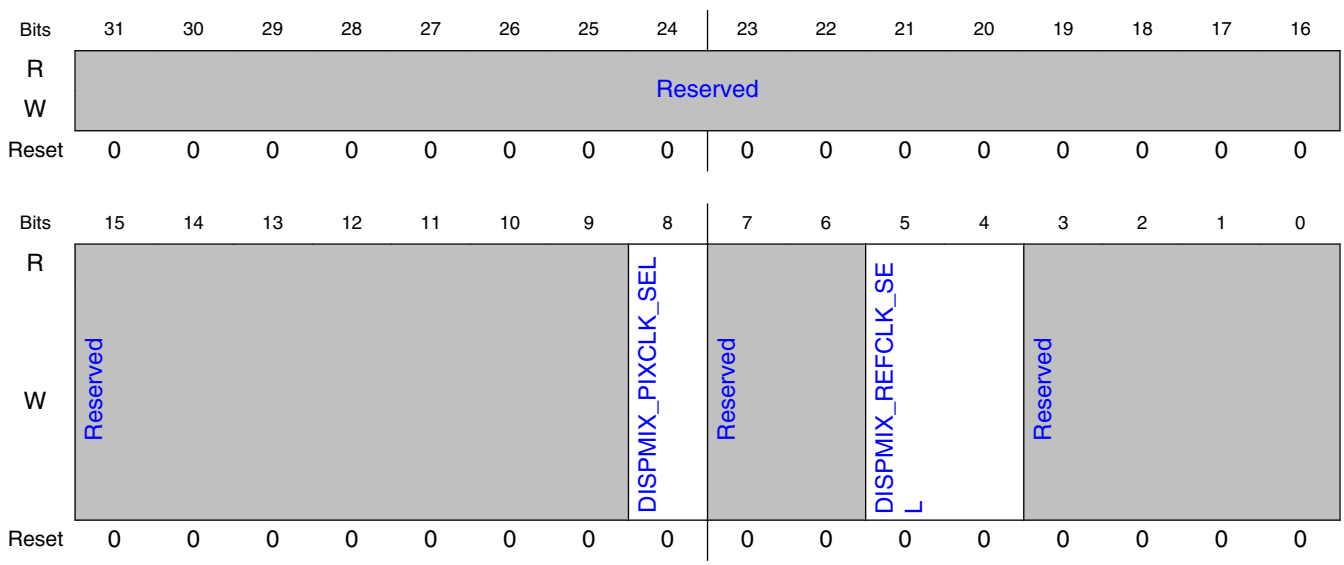
15.2.2.1.8.1 Offset

Register	Offset
CONTROL0_CLR	18h

15.2.2.1.8.2 Function

Misc. Control registers

15.2.2.1.8.3 Diagram



15.2.2.1.8.4 Fields

Field	Function
31-9 —	Reserved.
8	Display Subsystem Pixel Clock Select

Table continues on the next page...

Field	Function
DISPMIX_PIXCLK_SEL	Pixel Clock source selection. 0x0 = Video PLL2 Clock 0x1 = CCM DC Pixel Clock
7-6 —	Reserved.
5-4 DISPMIX_REFCLK_SEL	Display Subsystem Reference Clock Select Reference clock source selection. 0x0 = 27 MHz Oscillator Reference Clock 0x1 = Video PLL2 Clock 0x2 = CCM DC Pixel Clock
3-0 —	Reserved.

15.2.2.1.9 Control (CONTROL0_TOG)

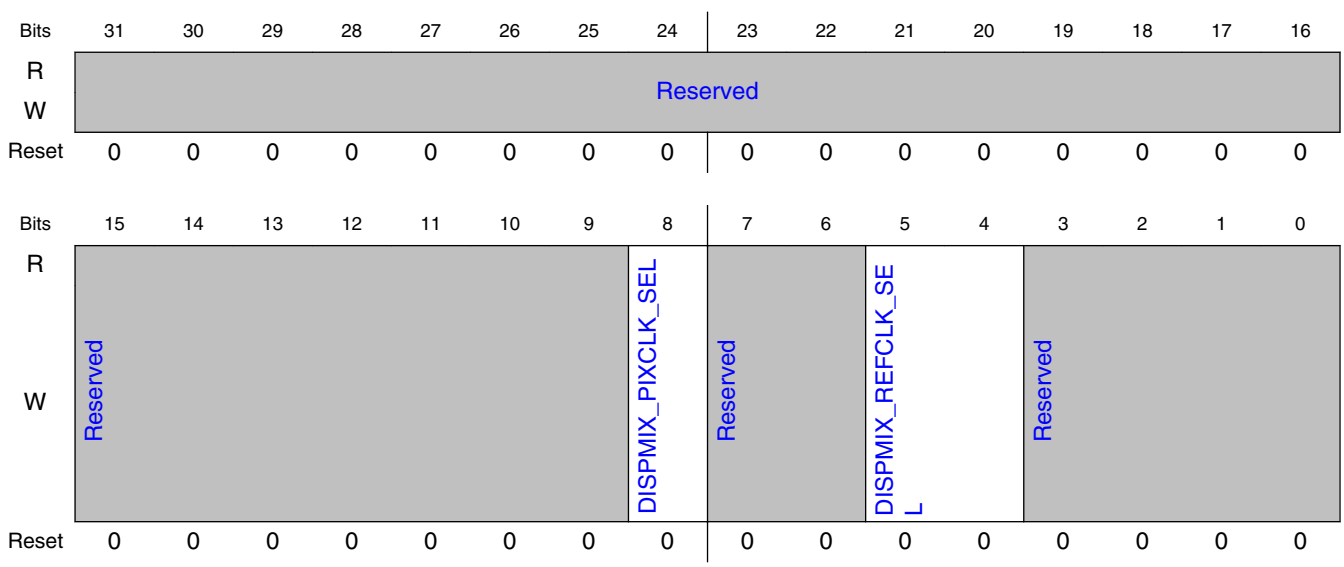
15.2.2.1.9.1 Offset

Register	Offset
CONTROL0_TOG	1Ch

15.2.2.1.9.2 Function

Misc. Control registers

15.2.2.1.9.3 Diagram



15.2.2.1.9.4 Fields

Field	Function
31-9 —	Reserved.
8 DISPMIX_PIXCLK_SEL	Display Subsystem Pixel Clock Select Pixel Clock source selection. 0x0 = Video PLL2 Clock 0x1 = CCM DC Pixel Clock
7-6 —	Reserved.
5-4 DISPMIX_REFCLK_SEL	Display Subsystem Reference Clock Select Reference clock source selection. 0x0 = 27 MHz Oscillator Reference Clock 0x1 = Video PLL2 Clock 0x2 = CCM DC Pixel Clock
3-0 —	Reserved.

15.2.2.1.10 Spare Control0 (SPARE_CTRL0)

15.2.2.1.10.1 Offset

Register	Offset
SPARE_CTRL0	20h

15.2.2.1.10.2 Function

Spare ECO Control Registers

15.2.2.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.10.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.11 Spare Control0 (SPARE_CTRL0_SET)

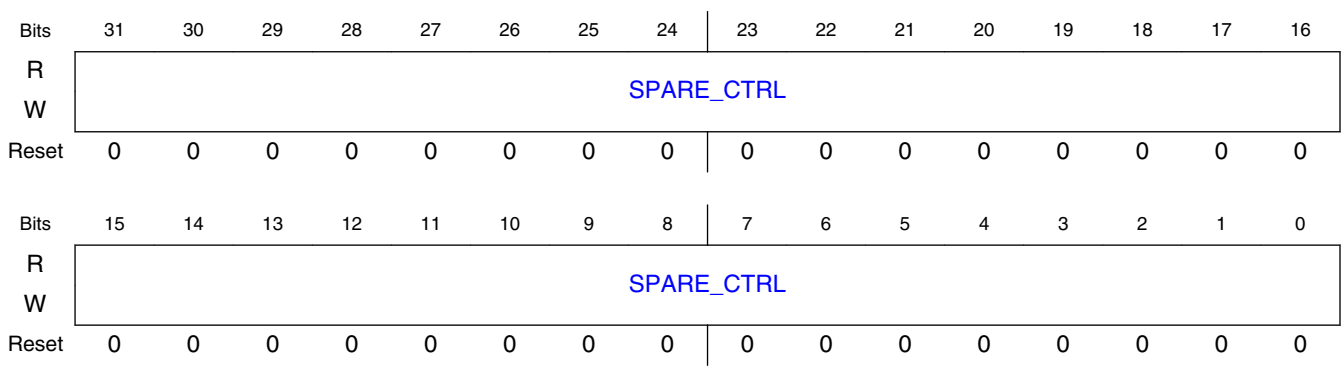
15.2.2.1.11.1 Offset

Register	Offset
SPARE_CTRL0_SET	24h

15.2.2.1.11.2 Function

Spare ECO Control Registers

15.2.2.1.11.3 Diagram



15.2.2.1.11.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.12 Spare Control0 (SPARE_CTRL0_CLR)

15.2.2.1.12.1 Offset

Register	Offset
SPARE_CTRL0_CLR	28h

15.2.2.1.12.2 Function

Spare ECO Control Registers

15.2.2.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.12.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.13 Spare Control0 (SPARE_CTRL0_TOG)

15.2.2.1.13.1 Offset

Register	Offset
SPARE_CTRL0_TOG	2Ch

15.2.2.1.13.2 Function

Spare ECO Control Registers

15.2.2.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.13.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.14 Spare Control1 (SPARE_CTRL1)

15.2.2.1.14.1 Offset

Register	Offset
SPARE_CTRL1	30h

15.2.2.1.14.2 Function

Spare ECO Control Registers

15.2.2.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.14.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.15 Spare Control1 (SPARE_CTRL1_SET)

15.2.2.1.15.1 Offset

Register	Offset
SPARE_CTRL1_SET	34h

15.2.2.1.15.2 Function

Spare ECO Control Registers

15.2.2.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.15.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.16 Spare Control1 (SPARE_CTRL1_CLR)

15.2.2.1.16.1 Offset

Register	Offset
SPARE_CTRL1_CLR	38h

15.2.2.1.16.2 Function

Spare ECO Control Registers

15.2.2.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.16.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.17 Spare Control1 (SPARE_CTRL1_TOG)

15.2.2.1.17.1 Offset

Register	Offset
SPARE_CTRL1_TOG	3Ch

15.2.2.1.17.2 Function

Spare ECO Control Registers

15.2.2.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SPARE_CTRL															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.17.4 Fields

Field	Function
31-0 SPARE_CTRL	SPARE Control Spare control registers intended for ECO purposes.

15.2.2.1.18 Spare Status0 (SPARE_STATUS0)

15.2.2.1.18.1 Offset

Register	Offset
SPARE_STATUS0	40h

15.2.2.1.18.2 Function

Spare ECO Control Status Registers

15.2.2.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.18.4 Fields

Field	Function
31-0 SPARE_STATUS	SPARE Control Spare control status registers intended for ECO purposes.

15.2.2.1.19 Spare Status0 (SPARE_STATUS0_SET)

15.2.2.1.19.1 Offset

Register	Offset
SPARE_STATUS0_SET	44h

15.2.2.1.19.2 Function

Spare ECO Control Status Registers

15.2.2.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.19.4 Fields

Field	Function
31-0 SPARE_STATUS	SPARE Control Spare control status registers intended for ECO purposes.

15.2.2.1.20 Spare Status0 (SPARE_STATUS0_CLR)

15.2.2.1.20.1 Offset

Register	Offset
SPARE_STATUS0_CLR	48h

15.2.2.1.20.2 Function

Spare ECO Control Status Registers

15.2.2.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.20.4 Fields

Field	Function
31-0 SPARE_STATUS	SPARE Control Spare control status registers intended for ECO purposes.

15.2.2.1.21 Spare Status0 (SPARE_STATUS0_TOG)

15.2.2.1.21.1 Offset

Register	Offset
SPARE_STATUS0_TOG	4Ch

15.2.2.1.21.2 Function

Spare ECO Control Status Registers

15.2.2.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	SPARE_STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.2.2.1.21.4 Fields

Field	Function
31-0 SPARE_STATUS	SPARE Control Spare control status registers intended for ECO purposes.

15.3 Display Timing Generator (DTG)

15.3.1 Overview

The purpose of the Display Timing Generator (DTG) is to generate the display timing control signals for the display and the pixel data path in the pixel clk domain. The outputs from the DTG controls the pixel data path and front end components (scaler, DPR, CTX_LD).

15.3.1.1 Block Diagram

The following diagram gives a high-level overview of the configuration of the DTG.

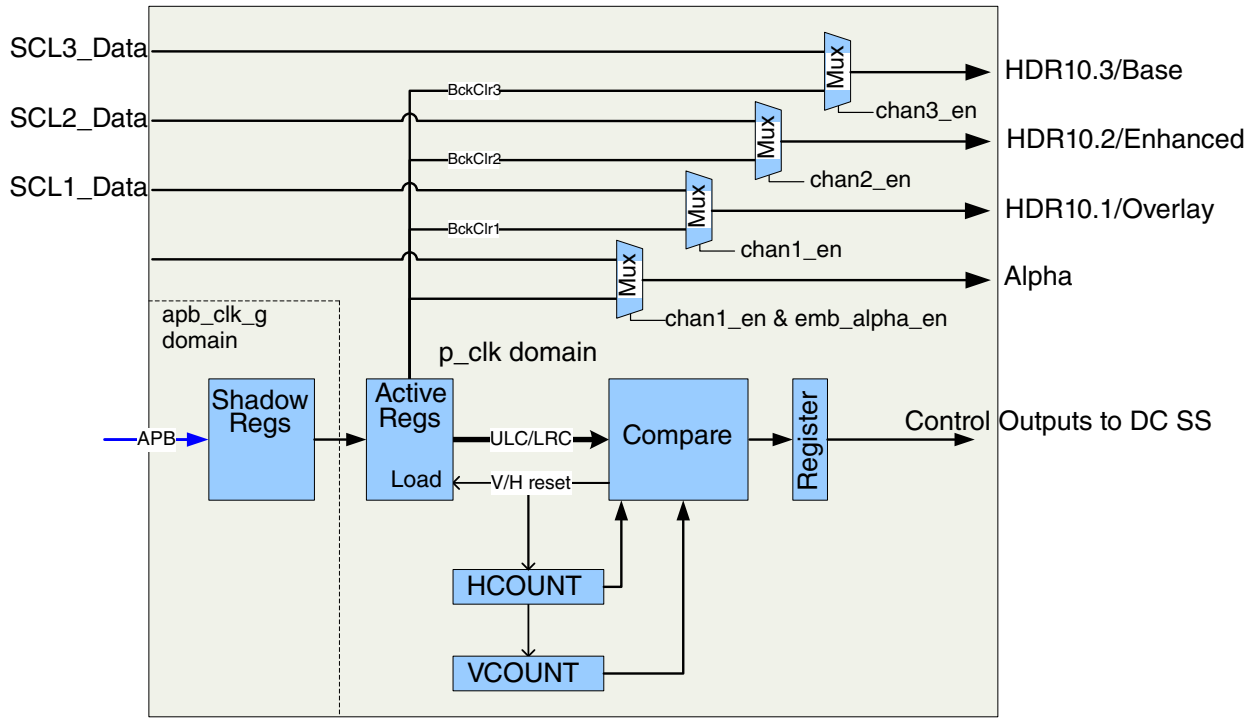


Figure 15-5. DTG block diagram

15.3.1.2 Features

The DTG contains the following components:

- Shadow Registers – These registers are loaded in advance of the next vertical retrace by system SW and used to load the active registers to setup for the next display trace at the vertical retrace time.
- Active Registers – These registers are loaded from the shadow registers in one clock in parallel. This occurs at the single p_clk edge where hcount = csr_dtg_lrc_x and vcount = csr_dtg_lrc_y.
- Hcount, vcount – Counters that determine the position in the display trace.
- Compare – Comparators that create active control signals for all DCSS components.
- Register – All control outputs are registered.
- Datapath – Muxes that select the scaler output data, or programmed background color based on whether the current H/Vcount are in the respective active regions.

Examples of control outputs from the DTG are:

- VActive_n, HActive, HSYNC – Note, these signals map to Dolby and HDR10 specific timing inputs. They are NOT equal to the traditional definition of display VSYNC/HSYNC/DE.

```
VActive_n = dolby_vsync_n
HActive = dolby_hsync
DE = dolby_de
```

- PS_EN[3,2,1] – Pixel Stream Enable, also referred to as VIDEO_ENABLE and OVERLAY_ENABLE. These active times are defined by the parameters *_ULC[X,Y]/*_LRC[X,Y], or the upper left and lower right display coordinates for the three pixel streams. These signals drive the bl_de, el_de, and de_in of the Dolby video and graphics pixel cores and the read_enable into the three scaler output FIFOs.
- CTX_LD__[DB|SB]_EN – Issue a control signal that indicates when a new display state can be loaded by the display context loader. There is one event for components with doubled buffered shadow registers, another with single buffered CSRs, since the timing events occur at different times in a single display trace.
- CONTROL[31:0] – Additional control signals that are used as the design evolves. These can be reduced at the final design release to match that actual implementation. Individual CSRs are used to determine when these signals are active within the display trace, and if they are horizontal or vertical control signals. Vertical means they are qualified by the horizontal count resetting to zero.

15.3.2 Functional description

15.3.2.1 DTG Timing Generator

The purpose of the DTG is to generate the display timing control signals for the display and the pixel data path in the p_clk domain. The DTG is fully synchronous to the p_clk domain except for the first set of double buffered CSR registers, also referred to as “shadow” registers. The shadow registers are synchronous to the apb_clk_s domain. The shadow registers are loaded into the active registers at the p_clk that defines the “origin” of the display. The active registers are clocked on p_clk and loaded when the shadow registers are not actively being updated. The outputs from the DTG control the pixel data path and front end components (scaler, DPR, CTX_LD). The “origin” of the display is defined as the p_clk cycle that causes the hcount and vcount to be set to zero. This occurs when the respective counters, hcount/vcount, reach their maximum programmed values and the p_clk results in hcount = vcount = 0x0. The origin of the display control is not arbitrarily defined, but must be the point at which the display enters the vertical front porch. This is the p_clk cycle that is just after the last pixel is sent to the display, or the lower right most pixel on the display. This origin is required to meet the proper

initialization of the display IP (Dolby and HDR10 components) AND to achieve synchronization with the output timing generator, or SUBSAM, that produces CEA861 HDMI compliant timing.

The Display Controller Subsystem (DCSS) does not support on-the-fly display resolution changes. Any display timing registers that effect the display resolution or refresh rate require the DCSS (and all components integrated) to be stopped, reset, and restarted with the correct pixel clock frequency and display parameters to drive the new resolution. One reason is that the SUBSAM (see subsequent descriptions) module doesn't implement double buffered display timing registers. Also, for a resolution update, the PLL, PHY, and clocking needs to be reinitialized. The double buffered registers that could be modified in the case of a display context change, and reloaded at retrace back to the origin, would be those that locate the three channels of display data within the active display region. In this case, the display resolution and timing is not changing, but the timing of the channels 1, 2, or 3, within the current display configuration, can change on-the-fly. This is one example of DTG registers that could change on-the-fly. Considering the transfer of the registers from shadows to active registers at retrace back to the origin, it is up to the application to determine which registers can be loaded using the double buffered scheme on-the-fly.

The CEA861 timing is generated at the end of the pixel data path in the SUBSAM module. The CEA861 timing is required to comply with the HDMI interface specification and provide the correct interface timing to the HDMI controller. The DTG timing is slightly different than the CEA861 timing. DTG produces the timing needed by Dolby and HDR10 pixel data path components, and the SUBSAM produces the CEA861 timing for HDMI controller and spec compliance. Refer to section Display Timing and Window Overlay for more information on display timing. A timing diagram that indicates the origin of the display is provided below. ALL timing parameters need to be programmed with the origin defined as the start of the vertical front porch AND the start of the horizontal front porch. The pixel data path IP (Dolby and HDR10) doesn't consider porches. There is only a blanking region, and a vertical active region (VActive). The vertical active region, VActive, is initiated when the VActive signal goes low. The vertical blanking region is initiated when the VActive signal goes high. The DTG and pixel data path logic needs to see the rising edge of the VActive signal as the first transition when the DTG is enabled. The reset state of the VActive signal is low. The low to high transition on VActive is the transition from the active display into the vertical front porch, and this event marks the origin (start) of the display trace.

In addition to timing and control, the DTG also implements the data path muxes for background color and alpha values.

NOTE

The timing of the VActive signal needs to be precise. No additional timing pulses during start up can exist since other components depend on a clean startup sequence.

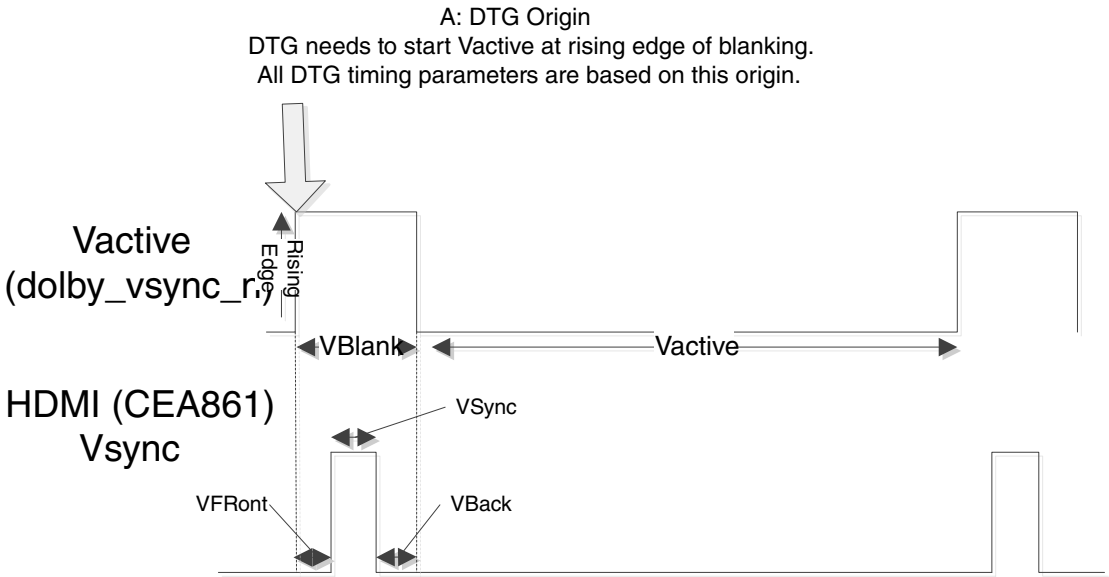


Figure 15-6. DTG Timing Diagram

15.3.2.2 DTG Window Overlay

The DTG generates the timing and control for the display and the pixel processing pipelines (Dolby/HDR10). Also, it generates timing for the CTX_LD to control when next state information is loaded to all DCSS components. An example of a display configuration is indicated as in the below diagram.

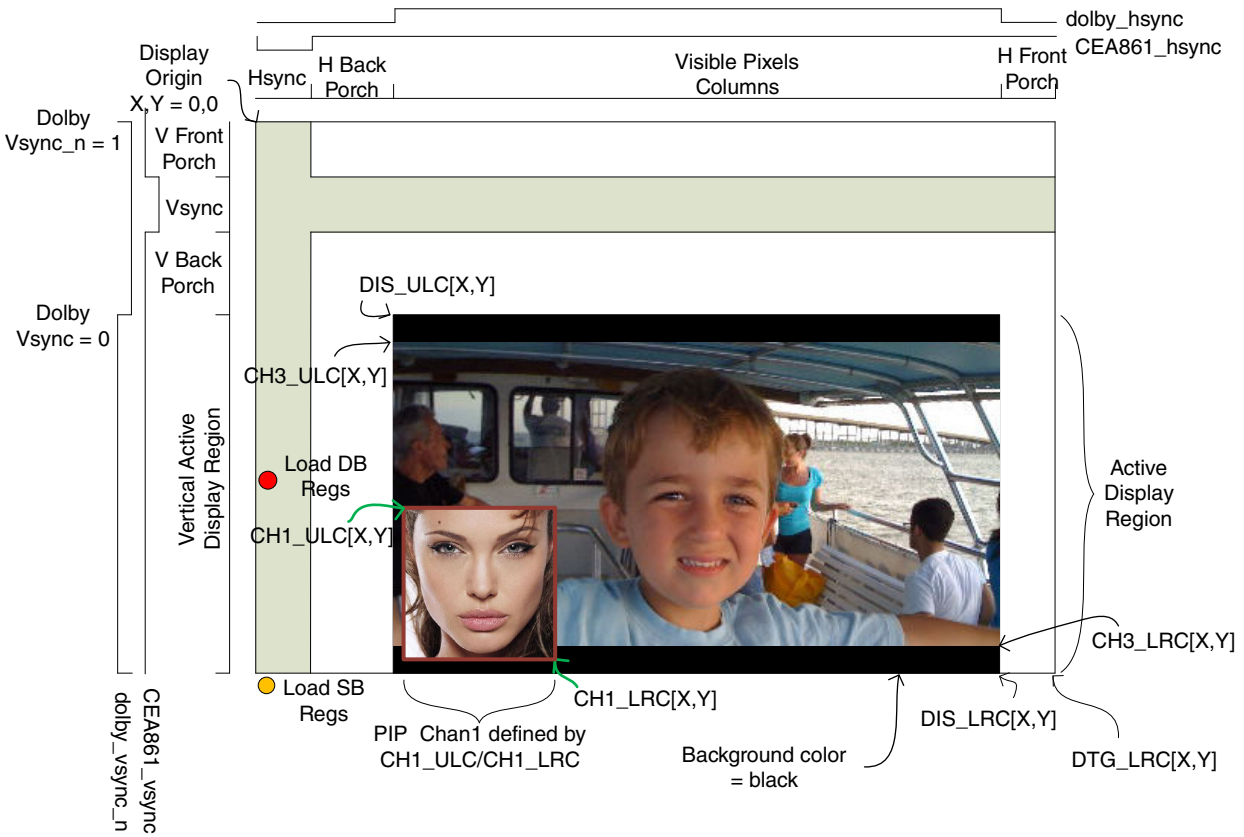


Figure 15-7. DTG Display Configuration

The DTG determines timing for the following events:

- When hcount and vcount counters are reset to zero, defined by =TC_DTG_LOWER_RIGHT_[X,Y], respectfully.
- When the active display region starts (Upper Left Coordinate or TC_DISPLAY_UPPER_LEFT_[X,Y]) and ends (Lower Right Coordinate or TC_DISPLAY_LOWER_RIGHT_[X,Y]).
- When Ch3/BL, starts/ends within the active display region defined by TC_CHANNEL_3_UPPER_LEFT_[X,Y]/ TC_CHANNEL_3_LOWER_RIGHT_[X,Y]. It is the same as the active display region in this example.
- When CH2/EL starts/ends indicated by TC_CHANNEL_2_UPPER_LEFT_[X,Y]/ TC_CHANNEL_2_LOWER_RIGHT_[X,Y]. This is the PIP window in this example.
- When the CH1/Overlay start/ends indicated by TC_CHANNEL_1_UPPER_LEFT_[X,Y]/ TC_CHANNEL_1_LOWER_RIGHT_[X,Y].
- When the context loader fetches register contents for double buffered (Dolby/DTG), indicated by the red dot, and single buffered (DPR/bus masters) components, indicated by the orange dot, in the DCSS.

The DTG output signals drive the Dolby Vision or HDR10 pixel processing IP and also drive the pixel_enables of the scaler output buffers. Also, the DTG drives the context loader enable with several event triggers during a single frame trace. This is required since the Dolby IP and DTG have double buffered registers that need to be preloaded during the active display time. Other processing components may not have double buffered registers (NXP IP), so these parameters need to be loaded during the vertical blanking time (not during active display time). The context loader bus masters the display state for the next display trace for the Dolby IP and NXP IP at programmable times determined by the DTG.

The display state sequence controlled by the DTG is:

1. At the Red Dot, Context Load DB Regs – Context load engine bus masters the double buffered registers in the Dolby IP and DTG. Must follow the sequence using the Metadata Copy, Start Bit, and Finish Bit (see Dolby IP specs).
2. VSYNC Active – Active CSR registers automatically loaded from shadow registers.
3. At the Orange Dot, Context Load SB Regs - Context load engine bus masters the register contents from DRAM to the single buffered IP control registers.
4. Prefetch/Process – the DPRs prefetch data and store rows of blocks into the RTRAM. The RTRAM_CTRL indicates when data is available to the scalers to begin processing. The rd_src engine also prefetches a buffer if enabled.
5. Display – When in the active display region, the PS_EN signals modulate when pixel data is transferred from the scaler output buffers and consumed by the Dolby pixel data path cores. HSYNC indicates an increment in each line.
6. Display Front Porch – goto #1.

15.3.2.2.1 HCount

The DTG uses a horizontal and vertical counter and the necessary parameters to maintain the sequence of the display. The horizontal counter (hcount) increments once every p_clk and reset back to zero when the hcount counter matches the TC_DTG_LOWER_RIGHT_[X] value in the active register set. The follow pseudo code is an example of when hcount resets to zero.

```
if (hcount == dtg_lrc_x) hcount <= 0;
else                     hcount <= hcount + 1;
```

15.3.2.2.2 VCount

The vertical counter (vcount) increments when the hcount counter resets to zero (hcount == TC_DTG_LOWER_RIGHT_[X]). The vertical counter resets to zero in the p_clk domain when hcount resets to zero and vcount matches the TC_DTG_LOWER_RIGHT_[Y] value in the active register set. The following pseudo code is an example of when the vcount resets to zero.

```
if ((hcount == TC_DTG_LOWER_RIGHT_X) &&
    (vcount == TC_DTG_LOWER_RIGHT_Y))    vcount <= 0;
else if (hcount == TC_DTG_LOWER_RIGHT_X) vcount <= vcount + 1;
```

A region in the display sequence that would actively process pixels for a channel is defined by the upper left and lower right coordinates, i.e. ULC/LRC. As an example, the following code would determine when to process pixels in the overlay channel.

```
if ((TC_CHANNEL_1_UPPER_LEFT_X <= hcount) &&
    (TC_CHANNEL_1_UPPER_LEFT_Y <= vcount) &&
    (TC_CHANNEL_1_LOWER_RIGHT_X >= hcount) &&
    (TC_CHANNEL_1_LOWER_RIGHT_Y >= vcount)) chan1_active <= 1'b1;
else                                chan1_active <= 1'b0;
```

The chan1_active (overlay is active in Dolby mode) signal would be used to read signals from scaler #1 output FIFO. It is typically active within the region defined by TC_DISPLAY_UPPER_LEFT_/TC_DISPLAY_LOWER_RIGHT_. Similar logic is used to create the timing for active processing of the other channels #2/3 , or video channels.

15.3.2.3 DTG Datapath

The DTG also implement the three pixel data path muxes for pixel data and back ground color. With the respective regions defined by ULC/LRC, pixel data is used to drive the pixel data path. Outside of these regions, the background color is forwarded to the pixel data path. The alpha MUX is also implemented in the DTG. The per pixel alpha value embedded in channel 1 is forwarded to the data path when channel 1 is active (or the TC_DEFAULT_OVERLAY_ALPHA is used when TC_CH1_PER_PEL_ALPHA_SEL = 0). The blender alpha is based on the per pixel graphics alpha:

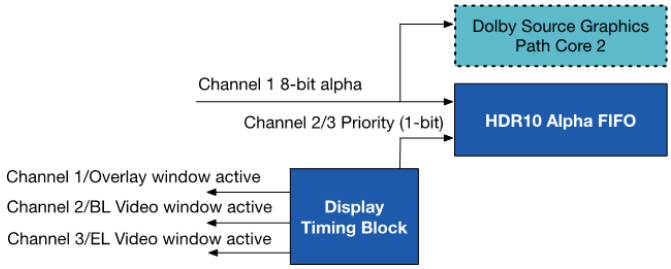


Figure 15-8. DTG Datapath Diagram

The Channel 2/3 Priority is 1 when Channel 2 is active and 0 when Channel 2 is inactive.

15.3.2.4 DTG Programming Example

The table below indicates the timing parameters for the Video Information Code (97, 107) defined for a progressive 4K TV at 60Hz.

Table 15-5. 4k TV at 60Hz Timing Parameters

Hactive	Vactive	Htotal	Hblank	Vtotal	Vblank	Pixel Freq
3840	2160	4400	560	2250	90	594
Hfront	**Hsync**	**Hback**	**Vfront**	**Vsync**	**Vback**	
176	88	296	8	10	72	

The DTG parameters for this display format are:

- TC_DTG_LOWER_RIGHT_X = 4400 – 1 = 4399; this is the Htotal from the table. Note the MINUS 1 value.
- TC_DTG_LOWER_RIGHT_Y = 2250 – 1 = 2249; this is the Vtotal from the table.
- TC_DISPLAY_UPPER_LEFT_X = 4400 – 3840 – 176 – 1 = 383; this is when the horizontal blanking ends, and active display region begins. dolby_hsync goes high at the p_clk when hcount == TC_DISPLAY_UPPER_LEFT_X
- TC_DISPLAY_UPPER_LEFT_Y = 89; this is when dolby_vsync_n goes low, and enters the active display region
- TC_DISPLAY_LOWER_RIGHT_X = 4400 – 176 – 1 = 4223; dolby_hsync goes low
- TC_DISPLAY_LOWER_RIGHT_Y = 2249; this is when dolby_vsync_n goes high, and enters vertical blanking

To extend the example to add a 64x64 pixel CH1 overlay with a 10 pixel offset into the active display area in both X and Y direction, the following CH1 parameters would be set:

- TC_CHANNEL_1_UPPER_LEFT_X = 4400 – 3840 – 176 – 1 + 10 = 559 + 10 = 393, or start of active display region (TC_CHANNEL_1_UPPER_LEFT_X) + 10

- TC_CHANNEL_1_UPPER_LEFT_Y = 90 – 1 + 10 = 89 + 10 = 99
- TC_CHANNEL_1_LOWER_RIGHT_X = 4400 – 3840 – 176 – 1 + 10 + 64 = 393 + 64 + 10 = 457 (TC_DISPLAY_LOWER_RIGHT_X + 64)
- TC_CHANNEL_1_LOWER_RIGHT_Y = 89 + 64 + 10 = 163

The timing to load SB registers starting with the second line (line #0, then line #1) in the vertical blanking region:

- TC_CNTXT_SB_Y = 0

And, to load double buffered registers in the active display region.

- TC_CNTXT_DB_Y = 199 //110 lines after the active display area begins

NOTE

The ATG in the SUBSAM must be programmed with compatible values that are consistent with the DTG values.

15.3.2.5 DTG Interlaced Video Timing Generator

The DTG supports interlaced timing generation. Interlaced timings comply to the definition of “General Interlaced” and “Special Interlaced” timing in the CEA861 specification. The timing diagram for the General Interlaced timing from the CEA861 spec is shown below.

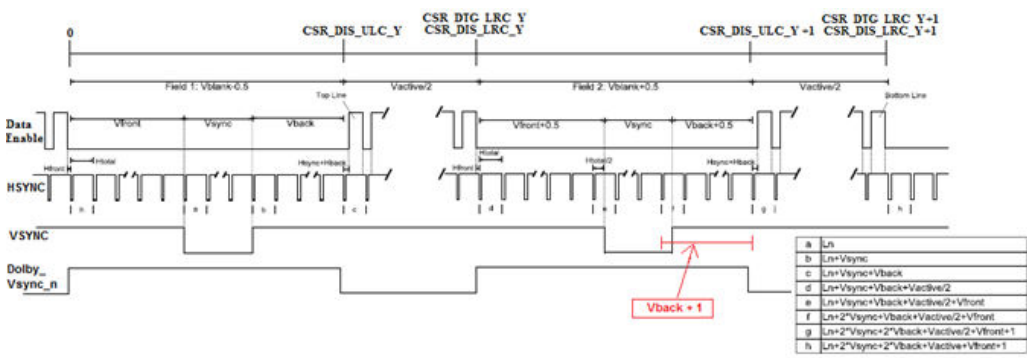


Figure 15-9. General Interlaced CEA861 Timing

The Dolby and HDR10 pixel processing engines do not use vertical front/back porches. The entire vertical blanking region is consider as one region.

```
Vblank = Vfront + Vsync + Vback
```

In interlaced mode, “Field 1” is the first field displayed, and “Field 2” is the second displayed, when the DTG is enabled. The DTG alternates between these fields indefinitely after the GO bit is set. The DTG doesn’t have separate timing parameters for the two fields, only one set of programming parameters is provided. There is an additional control bit to enable interlaced timing.

For Special Interlaced mode, there is no difference for the DTG between the fields.

For General Interlaced mode, Field 2 has an additional line added to the vertical blanking region (see Vback + 1 in the diagram above). The addition of the extra line is done automatically when the DTG interlaced mode is selected. The ATG in the SUBSAM module also needs to be programmed with the same General Interlaced mode when the extra line is needed. This is required to be sure the two timing generators are synced for both the odd and even fields in interlaced mode.

NOTE

Only the General Interlaced mode requires the DTG interlaced enable bit to the set. Special Interlaced mode does NOT need the interlaced mode to be set. Simply, the timing parameters need to be programmed considering there are half as many lines in the active display region, i.e. 540 lines for 1080i.

15.3.2.6 DTG Interrupt Generation

The DTG module can generate interrupts under the following conditions:

- There are four interrupts that are programmable which can generate an interrupt at a programmable XY coordinate
- Each of these are identical and independently programmable
- When the programmed coordinate (XY) is reached within the sweep of the display, the interrupt goes active. The X value is compared to the HCOUNT, and the Y value is compared to the VCOUNT. When both compares are equal, the interrupt status bit is set
- These timing based interrupts can be used for any purpose. Examples of interrupts include:
  - Notify when the display enters/exits the active display region
  - Notify when the Vsync goes active

The DTG also collects events, or signals, from other components in the system. The DTG supports up to 8 asynchronous event inputs that can be used as general DCSS level interrupts. The status and mask bits associated with these event sources are implemented within the DTG to generate DCSS interrupts. These 8 events, and using the status and mask registers, are collected and multiplexed down to a single interrupt output from the DTG module.

15.3.2.7 DTG Timing Standards

Refer to the CEA861-F for timing standards to drive displays at different resolutions. A summary of timings is in the following table.

Table 15-6. CEA861-F Timing Summary

Name	Resolution	Refresh	Pixel Clock	Hblank	Vblank	Aspect Ratio
2160p60	3840x2160	60.00	594.00	560	90	16:09
2160p59.94	3840x2160	59.94	593.41	560	90	16:09
2160p50	3840x2160	50.00	594.00	1440	90	16:09
2160p30	3840x2160	30.00	297.00	560	90	16:09
2160p29.97	3840x2160	29.97	296.70	560	90	16:09
2160p25	3840x2160	25.00	297.00	1440	90	16:09
2160p24	3840x2160	24.00	297.00	1660	90	16:09
2160p23.98	3840x2160	23.98	296.70	1660	90	16:09
1080p60	1920x1080	60.00	148.50	280	45	16:09
1080p59.94	1920x1080	59.94	148.35	280	45	16:09
1080p50	1920x1080	50.00	148.50	720	45	16:09
1080p30	1920x1080	30.00	74.25	280	45	16:09
1080p29.97	1920x1080	29.97	74.18	280	45	16:09
1080p25	1920x1080	25.00	74.25	720	45	16:09
1080p24	1920x1080	24.00	74.25	830	45	16:09
1080p23.98	1920x1080	23.98	74.18	830	45	16:09
720p60	1280x720	60.00	74.25	370	30	16:09
720p59.94	1280x720	59.94	74.18	370	30	16:09
720p50	1280x720	50.00	74.25	700	30	16:09
720p30	1280x720	30.00	74.25	2020	30	16:09
720p29.97	1280x720	29.97	74.18	2020	30	16:09
720p25	1280x720	25.00	74.25	2680	30	16:09
720p24	1280x720	24.00	74.25	2845	30	16:09
720p23.98	1280x720	23.98	74.18	2845	30	16:09
576p50	720x576	50.00	27.00	144	49	4:03
576p25	720x576	25.00	13.50	144	49	4:03
480p59.94	720x480	59.94	27.00	138	45	4:03
480p29.97	720x480	29.97	13.50	138	45	4:03

The display timing creates all control signals for the entire pixel data path and the output FIFOs for the scalers and the readback FIFOs. Some of the control signals that are used by the pixel data path are listed below:

- CHAN3_ACTIVE, CHAN2_ACTIVE – Indicates when pixels are fetched from the video scaler output FIFO. This signal, for either the base layer (bl_*) or enhancement layer (el_*), is logically equal to (~[bl|el]_vsync_n && [bl|el]_hsync && [bl|el]_de).
- CHAN1_ACTIVE – Indicates when pixels are fetched from the overlay scaler output FIFO. This signal is logically equal to (~vsync_n_in && hsync_in && de_in).

15.3.2.8 DTG Background Color

The YUV 3x10 bit and RGB 3x8 bit background colors are inserted at the inputs to the channel 1/2/3, or video and overlay, pipelines when CHAN1/2/3_ACTIVE are not asserted.

15.3.3 Memory Map and Registers

15.3.3.1 register descriptions

15.3.3.1.1 TIMING_CONTROL_AND_BLENDER Memory map

DTG base address: 2_0000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Timing Controller Control Register (TC_CONTROL_STATUS)	32	RW	0000_0000h
4h	DTG lower right corner locations (TC_DTG_REG1)	32	RW	0000_0000h
8h	Display Register: TOP Window Coordinates for Active display area (TC_DISPLAY_REG2)	32	RW	0000_0000h
Ch	Display Register: BOTTOM Window Coordinates for Active display area (TC_DISPLAY_REG3)	32	RW	0000_0000h
10h	Channel 1 window Register: TOP Window Coordinates for channel1 (TC_CH1_REG4)	32	RW	0000_0000h
14h	Channel_1 window Register: BOTTOM Window Coordinates for channel_1 window (TC_CH1_REG5)	32	RW	0000_0000h
18h	Channel 2 window Register: TOP Window Coordinates for channel_2 (TC_CH2_REG6)	32	RW	0000_0000h
1Ch	Channel_2 window Register: BOTTOM Window Coordinates for channel_2 pixel window (TC_CH2_REG7)	32	RW	0000_0000h
20h	Channel 3 window Register: TOP Window Coordinates for channel_3 (TC_CH3_REG8)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
24h	Channel_3 window Register: BOTTOM Window Coordinates for channel_3 pixel window (TC_CH3_REG9)	32	RW	0000_0000h
28h	Context Loader Register: Coordinates in the raster table where the context loader is started. (TC_CTX_LD_REG10)	32	RW	0000_0000h
2Ch	Channel_1 background pixel color. (TC_CH1_BKRND_REG11)	32	RW	0000_0000h
30h	Channel_2 background pixel color. (TC_CH2_BKRND_REG12)	32	RW	0000_0000h
38h	DBY MODE Blender control. (BLENDER_DBY_EOTF_RANGEINV)	32	RW	0000_0000h
3Ch	DBY MODE Blender control. (BLENDER_DBY_EOTF_RANGEMIN)	32	RW	0000_0000h
40h	DBY MODE blender control. (BLENDER_DBY_BDP)	32	RW	0000_0000h
44h	Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_I_GRAPHICS)	32	RW	0000_0000h
48h	Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_P_GRAPHICS)	32	RW	0000_0000h
4Ch	Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_T_GRAPHICS)	32	RW	0000_0000h
50h	LINE1 interrupt control: Coordinate where line1 interrupt is asserted (TC_LINE1_INT_REG13)	32	RW	0000_0000h
54h	LINE2 interrupt control: Coordinate where line2 interrupt is asserted (TC_LINE2_INT_REG14)	32	RW	0000_0000h
58h	default alpha (TC_ALPHA_DEFAULT_REG15)	32	RW	0000_0000h
5Ch	Timing Controller interrupt status (TC_INTERRUPT_STATUS)	32	RO	0000_0000h
60h	Timing Controller interrupt control. (TC_INTRERRUPT_CONTROL_REG17)	32	RW	0000_0000h
64h	Channel_3 background pixel color. (TC_CH3_BKRND_REG18)	32	RW	0000_0000h
68h	Timing Controller interrupt masks (TC_INTRERRUPT_MASK)	32	RW	0000_0000h
6Ch	LINE3 interrupt control: Coordinate where line3 interrupt is asserted (TC_LINE3_INT_REG)	32	RW	0000_0000h
70h	LINE4 interrupt control: Coordinate where line4 interrupt is asserted (TC_LINE4_INT_REG)	32	RW	0000_0000h
74h	For DBY Mode: Controls the assertion and de-assertion DE signal (Overlay channel). (TC_OL_DE_CONTROL_REG)	32	RW	0000_0000h
78h	For DBY Mode: Controls the assertion and de-assertion DE signal (Base layer (BL) channel). (TC_BL_DE_CONTROL_REG)	32	RW	0000_0000h
7Ch	For DBY Mode: Controls the assertion and de-assertion DE signal (Enhancement layer (EL) channel). (TC_EL_DE_CONTROL_REG)	32	RW	0000_0000h

15.3.3.1.2 Timing Controller Control Register (TC_CONTROL_STATUS)

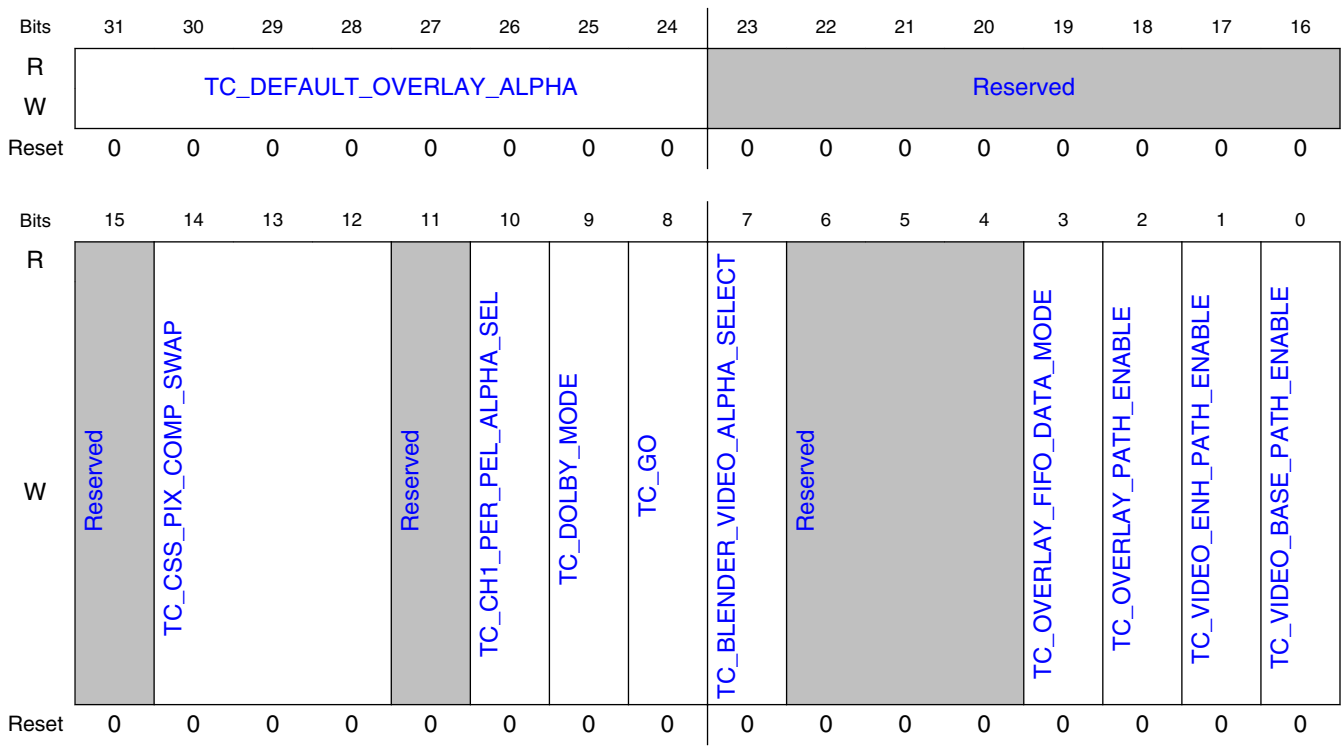
15.3.3.1.2.1 Offset

Register	Offset
TC_CONTROL_STATUS	0h

15.3.3.1.2.2 Function

Enables and controls the behaviour of timing controller and the three pixel channels in the display controller

15.3.3.1.2.3 Diagram



15.3.3.1.2.4 Fields

Field	Function
31-24 TC_DEFAULT_OVERLAY_ALPHA	Default alpha foreground used for the active regions where graphics channel does not provide an alpha value This is used when Graphics channel is ONLY RGB
23-15 —	Reserved.
14-12 TC_CSS_PIX_COMP_SWAP	Permutes the pixel conponent ordering into the chroma subsampler (CSS) block. 0: I P T 1: I T P 2: T I P 3: T P I 4: P I T 5: P T I

Table continues on the next page...

Field	Function
11 —	Reserved.
10 TC_CH1_PER_PEL_ALPHA_SEL	Enables per pixel alpha for channel1 (Overlay or Graphics) 0: use alpha from TC_DEFAULT_OVERLAY_ALPHA or foreground alpha 1: use alpha from channel1 Data stream and apply blending per pixel
9 TC_DOLBY_MODE	Enables Dolby mode. 0: HDR10 pixel processing is performed. Dolby processing disabled 1: Dolby Processing Enabled. Dolby timing control features enabled. Dolby Blending is enabled
8 TC_GO	Used to start and stop timing controller 0: timing controller inactive. No display control signal assertions (HSYNC = 0) 1: timing controller enabled. Display control processing is per the programming of the timing controller. HSYNC and VSYNC DE are running
7 TC_BLENDER_VIDEO_ALPHA_SELECT	alpha_valid 0: No alpha in the ch1 pixel so use default foreground alpha TC_DEFAULT_OVERLAY_ALPHA 1: there is an alpha in ch1 pixel
6-4 —	Reserved.
3 TC_OVERLAY_FIFO_DATA_MODE	Overlay fifo output data in yuv or RGB mode 0: YUV 1: RGB
2 TC_OVERLAY_PATH_ENABLE	Enable channel_1 (Dolby_mode:Overlay /HDR10: (GFX)) processing. 0: channel_1 disabled. Pixels cannot pass thru channel_1 1: channel_1 enabled
1 TC_VIDEO_ENH_PATH_ENABLE	Enable channel_2 (Dolby_mode:Enhancement channel) processing. 0: channel_2 disabled. Pixels cannot pass thru channel_2 1: channel_2 enabled
0 TC_VIDEO_BASE_PATH_ENABLE	Enable channel3 (Dolby_mode:Base Layer channel) processing. 0: channel_3 disabled. Pixels cannot pass thru channel_3 1: channel_3 enabled

15.3.3.1.3 DTG lower right corner locations (TC_DTG_REG1)

15.3.3.1.3.1 Offset

Register	Offset
TC_DTG_REG1	4h

15.3.3.1.3.2 Function

The X,Y cooridnates in this register specify the size of the raster table including blanking regions The upper left coordinate of this is (0,0)

15.3.3.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_DTG_LOWER_RIGHT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_DTG_LOWER_RIGHT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.3.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_DTG_LOWER_RIGHT_Y	lower right corner Y (vertical) coordinate of the raster table
15-13 —	Reserved.
12-0 TC_DTG_LOWER_RIGHT_X	lower right corner X (horizontal) coordinate of the raster table

15.3.3.1.4 Display Register: TOP Window Coordinates for Active display area (TC_DISPLAY_REG2)

15.3.3.1.4.1 Offset

Register	Offset
TC_DISPLAY_REG2	8h

15.3.3.1.4.2 Function

The X,Y cooridnates in this register specify the size of the active display. The upper left coordinate of the display window to specify the horizontal and vertical blankin region as an offset from (0,0). hactive and vactive are represented by the two values Any area in the raster table which is not covered by the active display area has a background color inserted.

15.3.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_DISPLAY_UPPER_LEFT_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_DISPLAY_UPPER_LEFT_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.4.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_DISPLAY_UPPER_LEFT_Y	upper left corner Y (vertical) coordinate of the active display region
15-13 —	Reserved.
12-0 TC_DISPLAY_UPPER_LEFT_X	upper left corner X (horizontal) coordinate of the active dispplay region

15.3.3.1.5 Display Register: BOTTOM Window Coordinates for Active display area (TC_DISPLAY_REG3)

15.3.3.1.5.1 Offset

Register	Offset
TC_DISPLAY_REG3	Ch

15.3.3.1.5.2 Function

The X,Y cooridnates in this register specify the size of the active display, Bottom Right Corner with respect to the Upper Left coordinates. Any area in the raster table which is not covered by the active display area has a background color inserted.

15.3.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_DISPLAY_LOWER_RIGHT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_DISPLAY_LOWER_RIGHT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.5.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_DISPLAY_LOWER_RIGHT_Y	lower right corner Y (vertical) coordinate of the active display region
15-13 —	Reserved.
12-0 TC_DISPLAY_LOWER_RIGHT_X	lower rignt corner X (horizontal) coordinate of the active dispplay region

15.3.3.1.6 Channel 1 window Register: TOP Window Coordinates for channel1 (TC_CH1_REG4)

15.3.3.1.6.1 Offset

Register	Offset
TC_CH1_REG4	10h

15.3.3.1.6.2 Function

The X,Y cooridnates in this register specify the upper left coordinate of channel_1.
In HDR10 mode this may be the graphics channel or video channel.
In DBY mode this is the Overlay Channel

15.3.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_CHANNEL_1_UPPER_LEFT_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_CHANNEL_1_UPPER_LEFT_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.6.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CHANNEL_1_UPPER_LEFT_Y	upper left corner Y (vertical) coordinate of the channel_1 window
15-13 —	Reserved.
12-0 TC_CHANNEL_1_UPPER_LEFT_X	upper left corner X (horizontal) coordinate of the channel_1 window

15.3.3.1.7 Channel_1 window Register: BOTTOM Window Coordinates for channel_1 window (TC_CH1_REG5)

15.3.3.1.7.1 Offset

Register	Offset
TC_CH1_REG5	14h

15.3.3.1.7.2 Function

The X,Y cooridnates in this register specify the Bottom Right coordinate of the channel_1 pixel window

15.3.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_CHANNEL_1_LOWER_RIGHT_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_CHANNEL_1_LOWER_RIGHT_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.7.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CHANNEL_1_LOWER_RIGHT_Y	lower right corner Y (vertical) coordinate of the channel_1 window
15-13 —	Reserved.
12-0 TC_CHANNEL_1_LOWER_RIGHT_X	lower rignt corner X (horizontal) coordinate of the channel_1 window

15.3.3.1.8 Channel 2 window Register: TOP Window Coordinates for channel_2 (TC_CH2_REG6)

15.3.3.1.8.1 Offset

Register	Offset
TC_CH2_REG6	18h

15.3.3.1.8.2 Function

The X,Y cooridnates in this register specify the size of the channel_2 window. The upper left coordinate of channel_2.

In HDR10 mode this is a video channel.

In DBY mode this is the Enhancement Layer Channel

15.3.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_CHANNEL_2_UPPER_LEFT_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_CHANNEL_2_UPPER_LEFT_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.8.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CHANNEL_2_UPPER_LEFT_Y	upper left corner Y (vertical) coordinate of the channel_2 window
15-13 —	Reserved.
12-0	upper left corner X (horizontal) coordinate of the channel_2 window

Field	Function
TC_CHANNEL_2_UPPER_LEFT_X	

15.3.3.1.9 Channel_2 window Register: BOTTOM Window Coordinates for channel_2 pixel window (TC_CH2_REG7)

15.3.3.1.9.1 Offset

Register	Offset
TC_CH2_REG7	1Ch

15.3.3.1.9.2 Function

The X,Y cooridnates in this register specify the Bottom Right coordinate of the channel_2 pixel window.

15.3.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_CHANNEL_2_LOWER_RIGHT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_CHANNEL_2_LOWER_RIGHT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.9.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CHANNEL_2_LOWER_RIGHT_Y	lower right corner Y (vertical) coordinate of the channel_2 window

Table continues on the next page...

Field	Function
15-13 —	Reserved.
12-0 TC_CHANNEL_2_LOWER_RIGHT_X	lower rignt corner X (horizontal) coordinate of the channel_2 window

15.3.3.1.10 Channel 3 window Register: TOP Window Coordinates for channel_3 (TC_CH3_REG8)

15.3.3.1.10.1 Offset

Register	Offset
TC_CH3_REG8	20h

15.3.3.1.10.2 Function

The X,Y cooridnates in this register specify the size of the channel_3 window. The upper left coordinate of channel_3.

In HDR10 mode this is a video channel.

In DBY mode this is the Base Layer (BL)

15.3.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_CHANNEL_3_UPPER_LEFT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_CHANNEL_3_UPPER_LEFT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.10.4 Fields

Field	Function
31-29	Reserved.

Table continues on the next page...

Field	Function
—	
28-16 TC_CHANNEL_3_UPPER_LEFT_Y	upper left corner Y (vertical) coordinate of the channel_3 window
15-13 —	Reserved.
12-0 TC_CHANNEL_3_UPPER_LEFT_X	upper left corner X (horizontal) coordinate of the channel_3 window

15.3.3.1.11 Channel_3 window Register: BOTTOM Window Coordinates for channel_3 pixel window (TC_CH3_REG9)

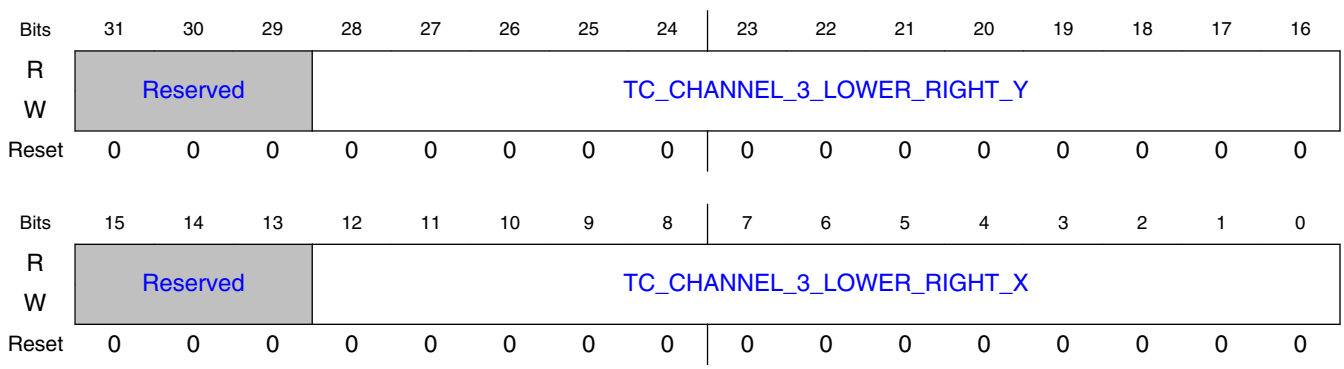
15.3.3.1.11.1 Offset

Register	Offset
TC_CH3_REG9	24h

15.3.3.1.11.2 Function

The X,Y cooridnates in this register specify the size of the channel_3 pixel window. Bottom Right

15.3.3.1.11.3 Diagram



15.3.3.1.11.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CHANNEL_3_LOWER_RIGHT_Y	lower right corner Y (vertical) coordinate of the channel_3 window
15-13 —	Reserved.
12-0 TC_CHANNEL_3_LOWER_RIGHT_X	lower rignt corner X (horizontal) coordinate of the channel_3 window

15.3.3.1.12 Context Loader Register: Coordinates in the raster table where the context loader is started. (TC_CTX_LD_REG10)

15.3.3.1.12.1 Offset

Register	Offset
TC_CTX_LD_REG10	28h

15.3.3.1.12.2 Function

Line Count in the raster table where the CTX_LD is triggered for loading context (some double buffered). The register field provides the line count that is compared with the VCOUNT counter value to generate the event for loading of Single Buffered or Double Buffered Context Timing Controller and Dolby context are double buffered. Rest of the blocks are single buffered

15.3.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_CNTXT_SB_LINE_COUNT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_CNTXT_DB_LINE_COUNT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.12.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_CNTXT_SB_LINE_COUNT	Line Count for Sngle Bufferend (SB) Context Loading
15-13 —	Reserved.
12-0 TC_CNTXT_DB_LINE_COUNT	Line Count for Double Buffered (DB) Context Loading

15.3.3.1.13 Channel_1 background pixel color. (TC_CH1_BKRND_REG11)

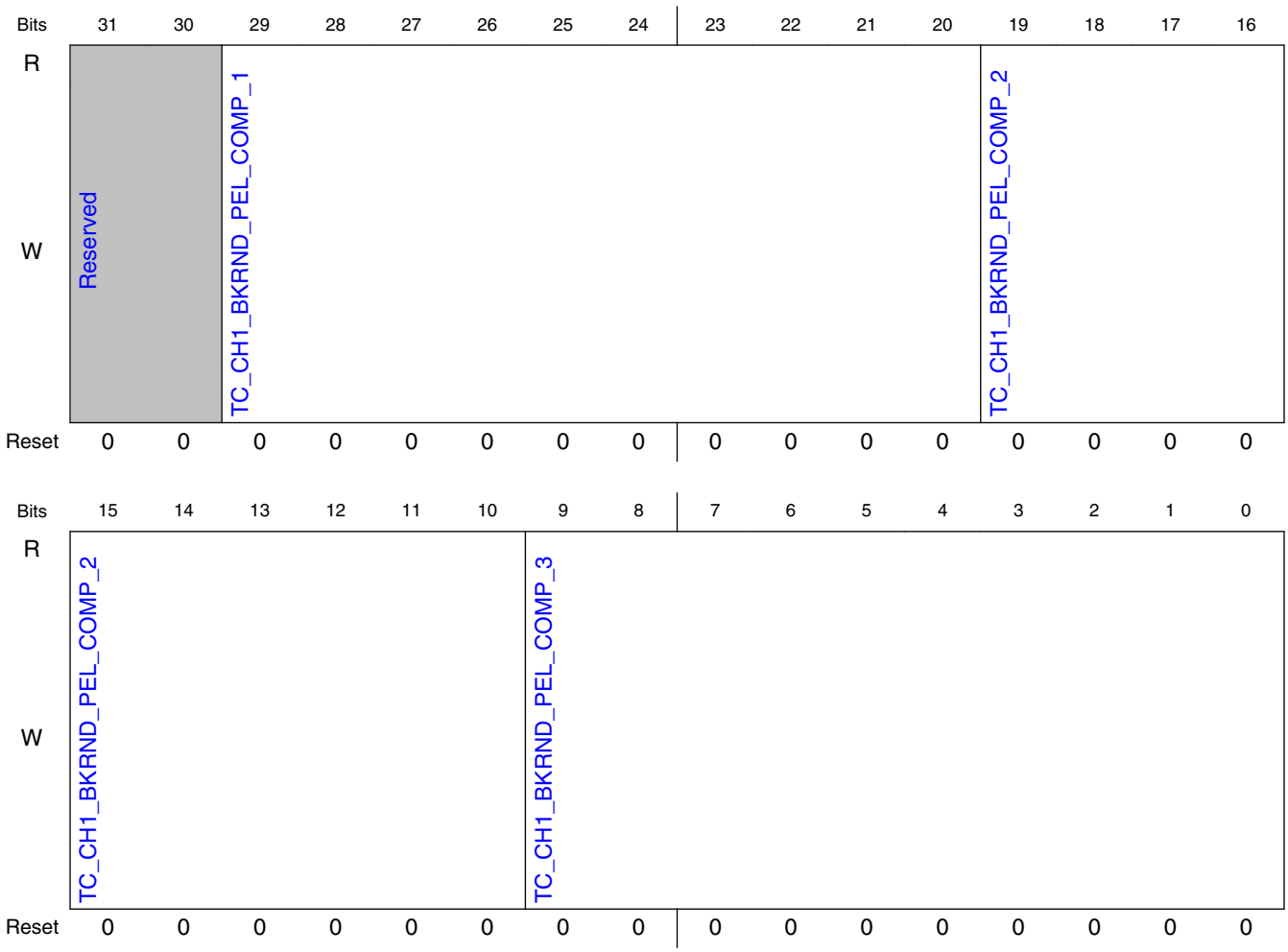
15.3.3.1.13.1 Offset

Register	Offset
TC_CH1_BKRND_REG11	2Ch

15.3.3.1.13.2 Function

Outside of the Channel_1 pixel window, This background color is put on channel_1. 10 bits per component. Color space depending on programming

15.3.3.1.13.3 Diagram



15.3.3.1.13.4 Fields

Field	Function
31-30 —	Reserved.
29-20 TC_CH1_BKRND_PEL_COMP_1	10-bit component (R or Y)
19-10 TC_CH1_BKRND_PEL_COMP_2	10-bit component (G or Cb)
9-0	10-bit component (B or Cr)

Field	Function
TC_CH1_BKRND_PEL_COMP_3	

15.3.3.1.14 Channel_2 background pixel color. (TC_CH2_BKRND_REG12)

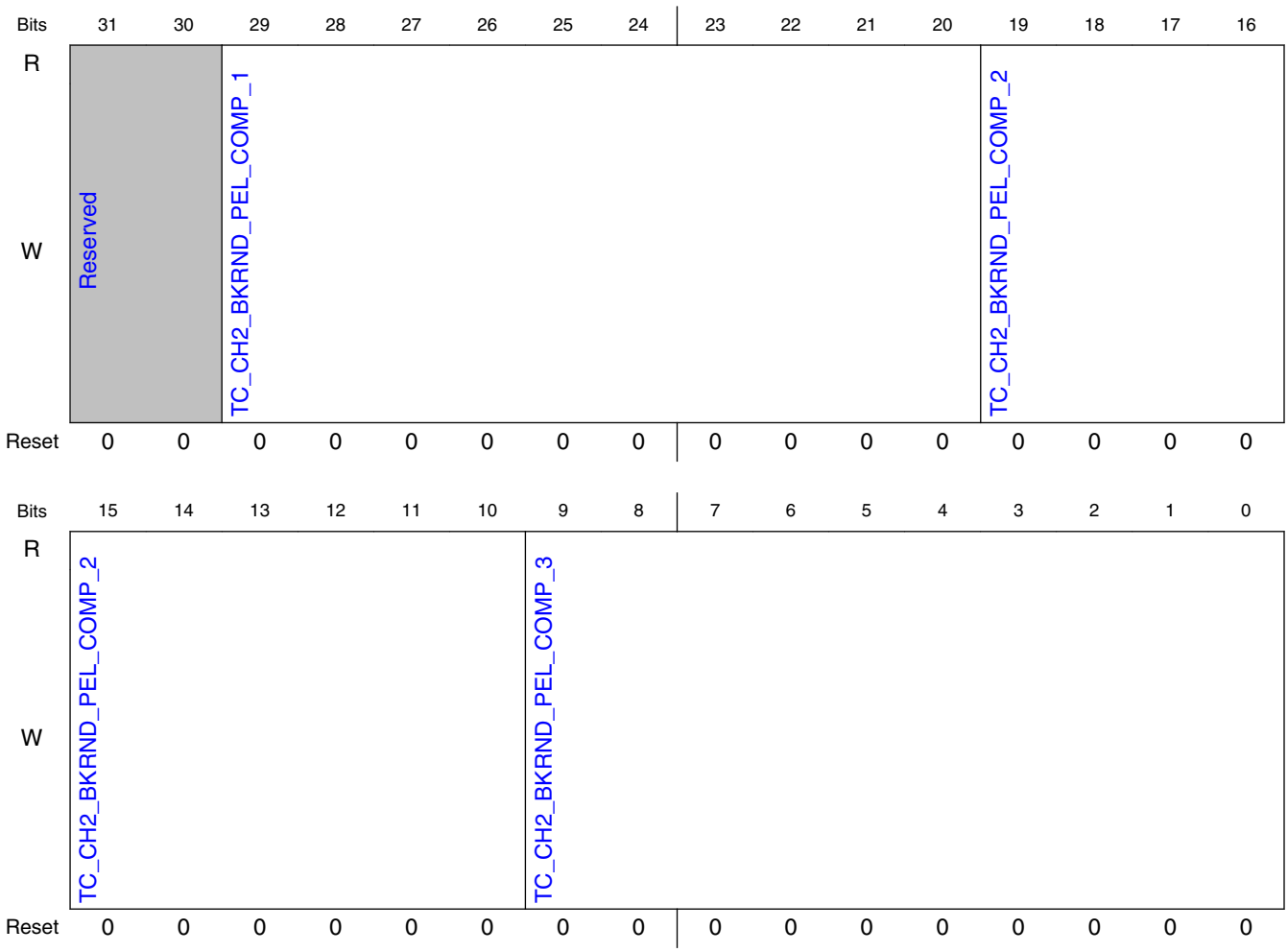
15.3.3.1.14.1 Offset

Register	Offset
TC_CH2_BKRND_REG12	30h

15.3.3.1.14.2 Function

Outside of the Channel_2 pixel window, This background color is put on channel_2. 10 bits per component. Color space depending on programming

15.3.3.1.14.3 Diagram



15.3.3.1.14.4 Fields

Field	Function
31-30 —	Reserved.
29-20 TC_CH2_BKRND_PEL_COMP_1	10-bit component (R or Y)
19-10 TC_CH2_BKRND_PEL_COMP_2	10-bit component (G or Cb)
9-0	10-bit component (B or Cr)

Field	Function
TC_CH2_BKRND_PEL_COMP_3	

15.3.3.1.15 DBY MODE Blender control. (BLENDER_DBY_EOTF_RANGEINV)

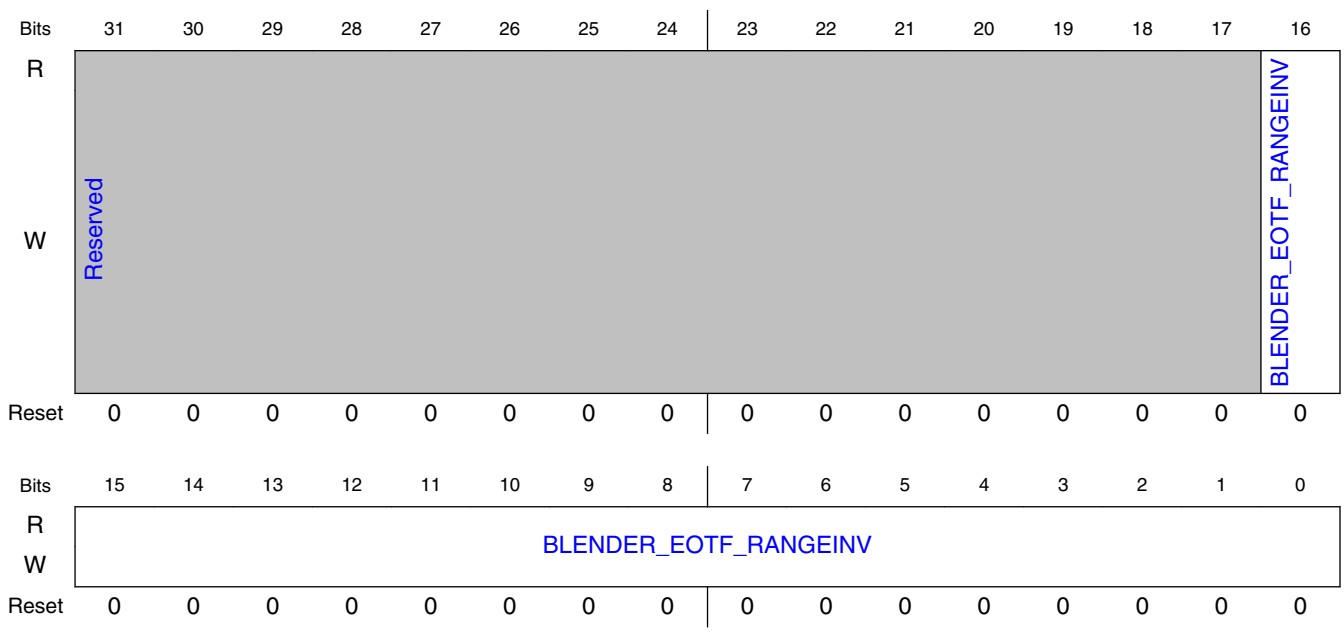
15.3.3.1.15.1 Offset

Register	Offset
BLENDER_DBY_EOTF_RANGEINV	38h

15.3.3.1.15.2 Function

eotf_rangeInv in DBY Blender EQN.

15.3.3.1.15.3 Diagram



15.3.3.1.15.4 Fields

Field	Function
31-17 —	Reserved.
16-0 BLENDER_EOTF_RANGEINV	eotf_rangeInv parameter for DBY blender

15.3.3.1.16 DBY MODE Blender control. (BLENDER_DBY_EOTF_RANGEMIN)

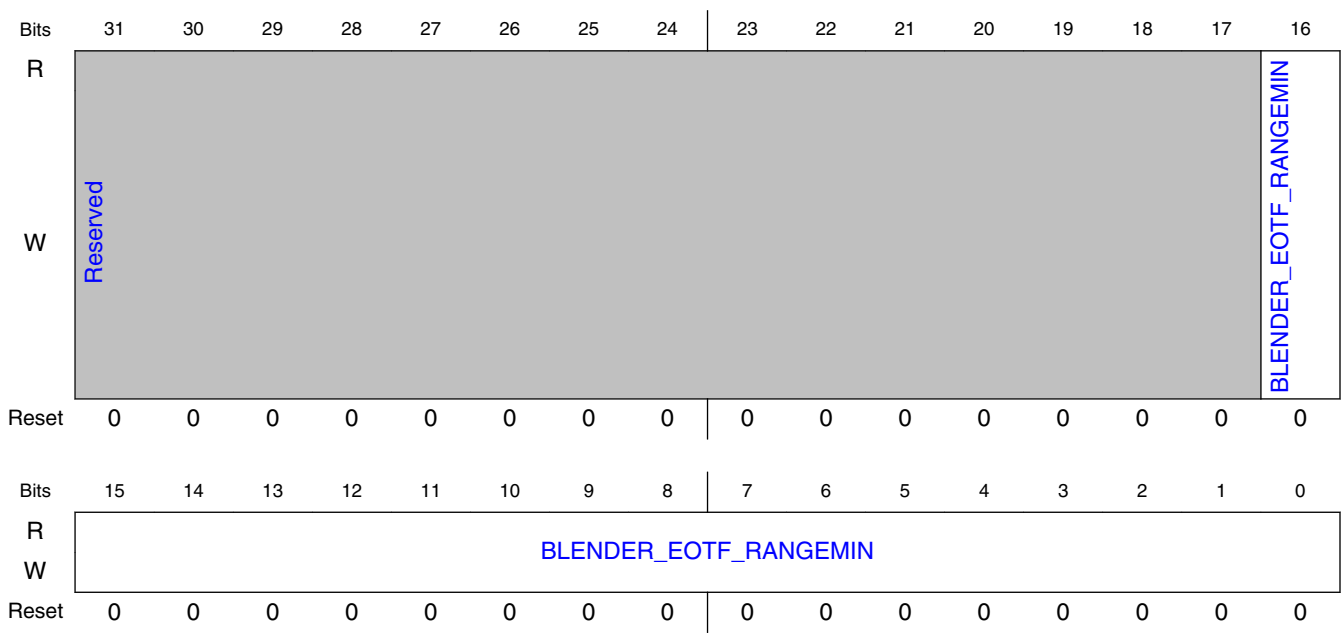
15.3.3.1.16.1 Offset

Register	Offset
BLENDER_DBY_EOTF_RANGEMIN	3Ch

15.3.3.1.16.2 Function

eotf_rangeMin in DBY blender EQN.

15.3.3.1.16.3 Diagram



15.3.3.1.16.4 Fields

Field	Function
31-17 —	Reserved.
16-0 BLENDER_EOTF_RANGEMIN	eotf_rangeMin parameter for DBY blender

15.3.3.1.17 DBY MODE blender control. (BLENDER_DBY_BDP)

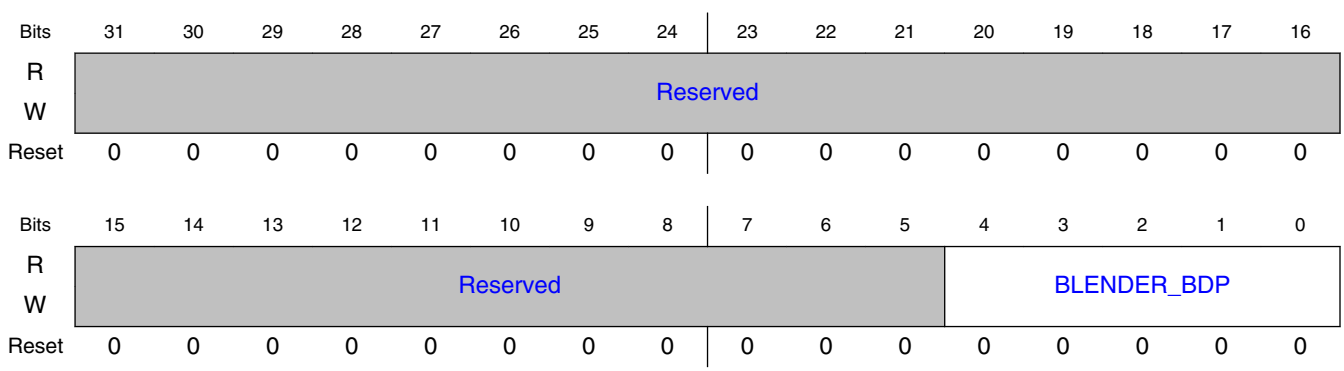
15.3.3.1.17.1 Offset

Register	Offset
BLENDER_DBY_BDP	40h

15.3.3.1.17.2 Function

bitDepth parameter for DBY blender equation

15.3.3.1.17.3 Diagram



15.3.3.1.17.4 Fields

Field	Function
31-5 —	Reserved.

Table continues on the next page...

Field	Function
4-0 BLENDER_BDP	bitDepth parameter in DBY blender

15.3.3.1.18 Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_I_GRAPHICS)

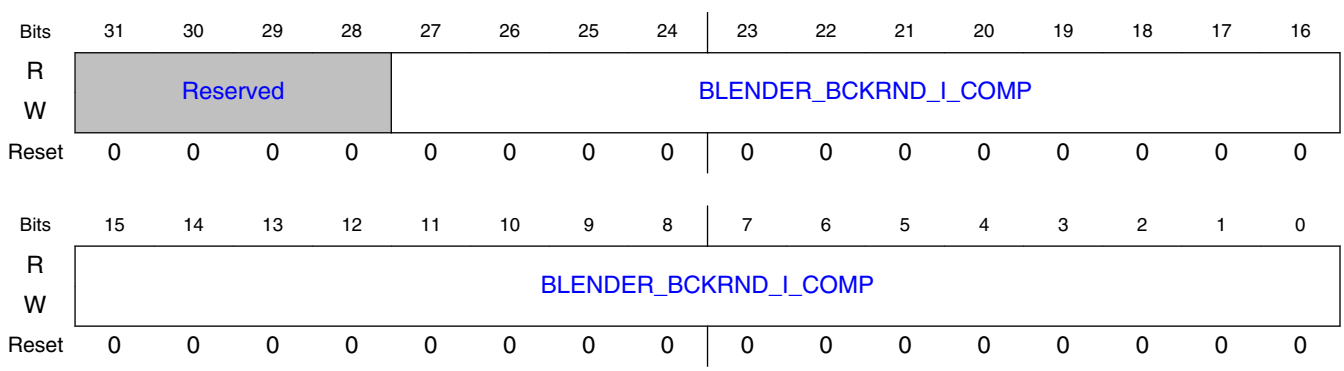
15.3.3.1.18.1 Offset

Register	Offset
BLENDER_BKRND_I_ GRAPHICS	44h

15.3.3.1.18.2 Function

Backgound pixel color sent to blender. Used when no valid pixels 28 bits per component. Color space depending on programming

15.3.3.1.18.3 Diagram



15.3.3.1.18.4 Fields

Field	Function
31-28 —	Reserved.
27-0 BLENDER_BCK RND_I_COMP	28-bit component I component in DBY mode R/Y component in HDR10 MODE

15.3.3.1.19 Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_P_GRAPHICS)

15.3.3.1.19.1 Offset

Register	Offset
BLENDER_BKRND_P_GRAPHICS	48h

15.3.3.1.19.2 Function

Backgound pixel color sent to blender. Used when no valid pixels 28 bits per component. Color space depending on programming

15.3.3.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				BLENDER_BCKRND_P_COMP											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BLENDER_BCKRND_P_COMP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.19.4 Fields

Field	Function
31-28 —	Reserved.
27-0 BLENDER_BCKRND_P_COMP	28-bit component P component in DBY mode G/Cb component in HDR10 MODE

15.3.3.1.20 Backgound pixel color component sent to blender. Used when no valid pixels (BLENDER_BKRND_T_GRAPHICS)

15.3.3.1.20.1 Offset

Register	Offset
BLENDER_BKRND_T_GRAPHICS	4Ch

15.3.3.1.20.2 Function

Backgound pixel color sent to blender. Used when no valid pixels 28 bits per component. Color space depending on programming

15.3.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				BLENDER_BCKRND_T_COMP											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BLENDER_BCKRND_T_COMP															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.20.4 Fields

Field	Function
31-28 —	Reserved.
27-0 BLENDER_BCKRND_T_COMP	28-bit component T component in DBY mode B/Cr component in HDR10 MODE

15.3.3.1.21 LINE1 interrupt control: Coordinate where line1 interrupt is asserted (TC_LINE1_INT_REG13)

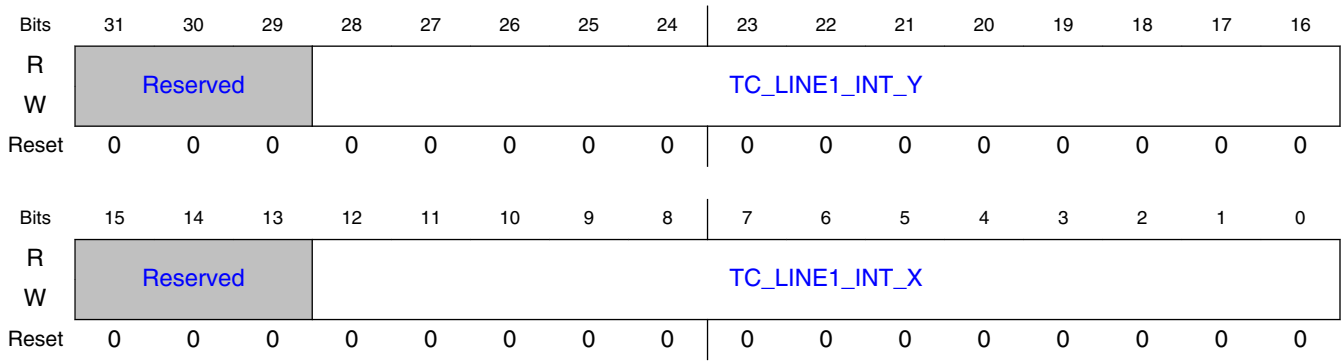
15.3.3.1.21.1 Offset

Register	Offset
TC_LINE1_INT_REG13	50h

15.3.3.1.21.2 Function

The X,Y cooridnates in this register specify where line1 interrupt is asserted as timing controller encounters these coordinates. Can be anywhere on raster table. For historical reasons this is called LINE1 interrupt in RTL

15.3.3.1.21.3 Diagram



15.3.3.1.21.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_LINE1_INT_Y	Y (vertical) coordinate for line1 interrupt
15-13 —	Reserved.
12-0 TC_LINE1_INT_X	X (horizontal) coordinate for line1 interrupt

15.3.3.1.22 LINE2 interrupt control: Coordinate where line2 interrupt is asserted (TC_LINE2_INT_REG14)

15.3.3.1.22.1 Offset

Register	Offset
TC_LINE2_INT_REG14	54h

15.3.3.1.22.2 Function

The X,Y cooridnates in this register specify where line2 interrupt is asserted as timing controller encounters these coordinates. Can be anywhere on raster table. For historical reasons this is called LINE2 interrupt in RTL

15.3.3.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_LINE2_INT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_LINE2_INT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.22.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_LINE2_INT_Y	Y (vertical) coordinate for line2 interrupt
15-13 —	Reserved.
12-0 TC_LINE2_INT_X	X (horizontal) coordinate for line2 interrupt

15.3.3.1.23 default alpha (TC_ALPHA_DEFAULT_REG15)

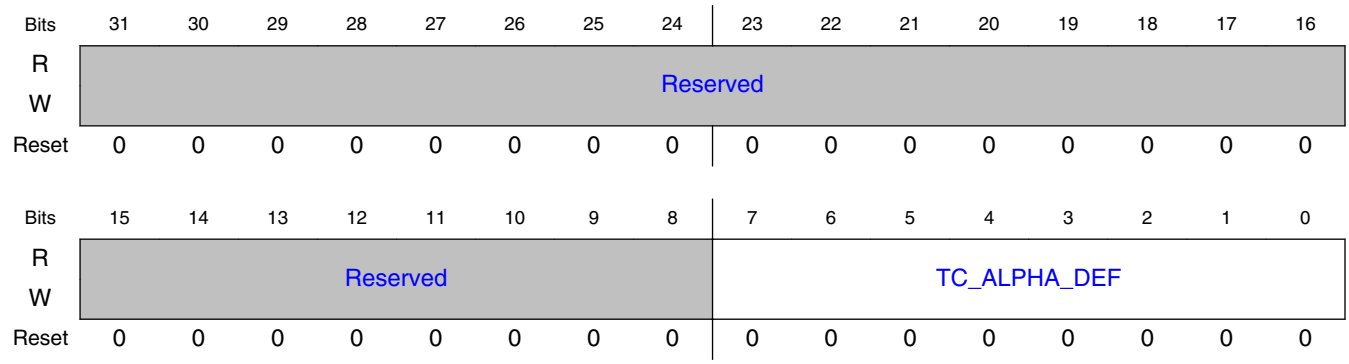
15.3.3.1.23.1 Offset

Register	Offset
TC_ALPHA_DEFAULT_REG15	58h

15.3.3.1.23.2 Function

This background alpha is applied when the per pixel alpha is not enabled

15.3.3.1.23.3 Diagram



15.3.3.1.23.4 Fields

Field	Function
31-8 —	Reserved.
7-0 TC_ALPHA_DEF	default background alpha value

15.3.3.1.24 Timing Controller interrupt status (TC_INTERRUPT_STATUS)

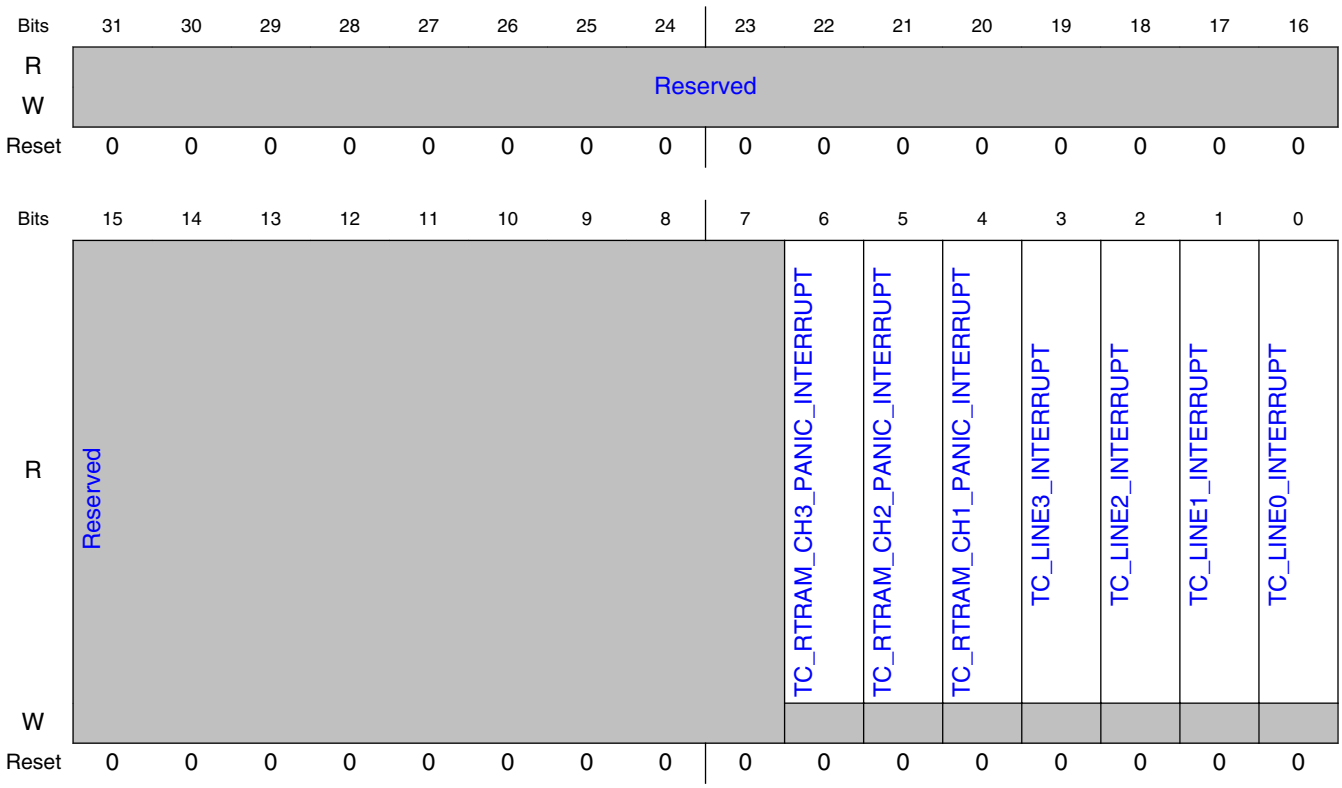
15.3.3.1.24.1 Offset

Register	Offset
TC_INTERRUPT_STATUS	5Ch

15.3.3.1.24.2 Function

This register has per interrupt status bits to indicate which interrupt(s) has occurred

15.3.3.1.24.3 Diagram



15.3.3.1.24.4 Fields

Field	Function
31-7 —	Reserved.
6 TC_RTRAM_CH3_PANIC_INTERRUPT	Panic interrupt is asserted by the scaler for channel3. Note: scaler doc may call this channel2 0: no interrupt 1: panic interrput has been asserted
5 TC_RTRAM_CH2_PANIC_INTERRUPT	Panic interrupt is asserted by the scaler for channel2. Note: scaler doc may call this channel1 0: no interrupt 1: panic interrput has been asserted
4 TC_RTRAM_CH1_PANIC_INTERRUPT	Panic interrupt is asserted by the scaler for channel1. Note: scaler doc may call this channel0 0: no interrupt 1: panic interrput has been asserted
3 TC_LINE3_INTERRUPT	LINE3 interrupt status. LINE3 interrupt can be programmed to occur at any coordinate. 0: no interrupt 1: LINE3 interrupt has been asserted
2	LINE2 interrupt status. LINE2 interrupt can be programmed to occur at any coordinate.

Table continues on the next page...

Field	Function
TC_LINE2_INTERRUPT	0: no interrupt 1: LINE2 interrupt has been asserted
1 TC_LINE1_INTERRUPT	LINE1 interrupt status. LINE1 interrupt can be programmed to occur at any coordinate. 0: no interrupt 1: LINE1 interrupt has been asserted
0 TC_LINE0_INTERRUPT	LINE0 interrupt status. LINE0 interrupt can be programmed to occur at any coordinate. 0: no interrupt 1: LINE0 interrupt has been asserted

15.3.3.1.25 Timing Controller interrupt control. (TC_INTRERRUPT_CONTROL_REG17)

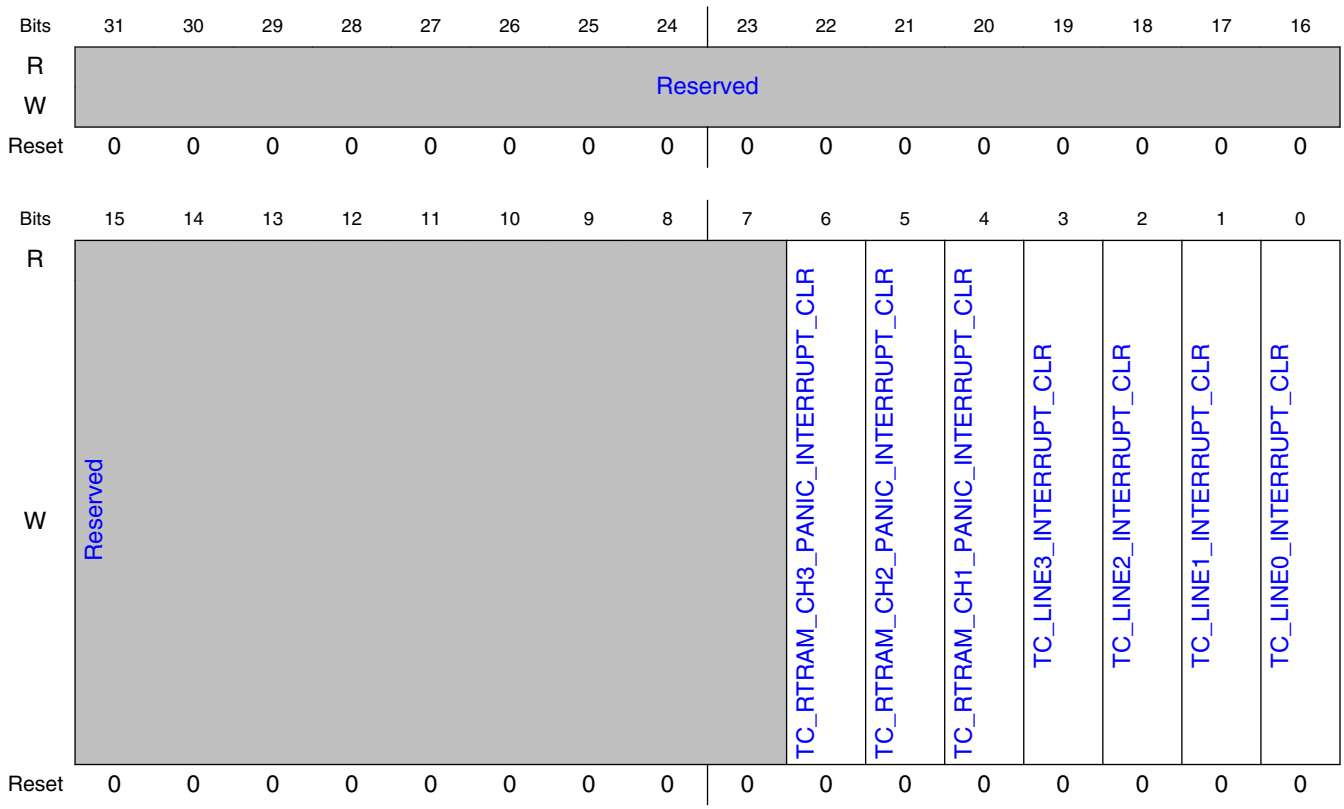
15.3.3.1.25.1 Offset

Register	Offset
TC_INTRERRUPT_CONTROL_REG17	60h

15.3.3.1.25.2 Function

This register has per interrupt control bits to clear interrupts in the TC_INTERRUPT_STATUS register

15.3.3.1.25.3 Diagram



15.3.3.1.25.4 Fields

Field	Function
31-7 —	Reserved.
6 TC_RTRAM_CH3_PANIC_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register. Channel 3 panic interrupt
5 TC_RTRAM_CH2_PANIC_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register. Channel 2 panic interrupt
4 TC_RTRAM_CH1_PANIC_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register. Channel 1 panic interrupt
3 TC_LINE3_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register.

Table continues on the next page...

Field	Function
2 TC_LINE2_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register.
1 TC_LINE1_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register.
0 TC_LINE0_INTERRUPT_CLR	Writing a 1 to this bit clears the respective interrupt in the TC_INTERRUPT_STATUS register.

15.3.3.1.26 Channel_3 background pixel color. (TC_CH3_BKRND_REG18)

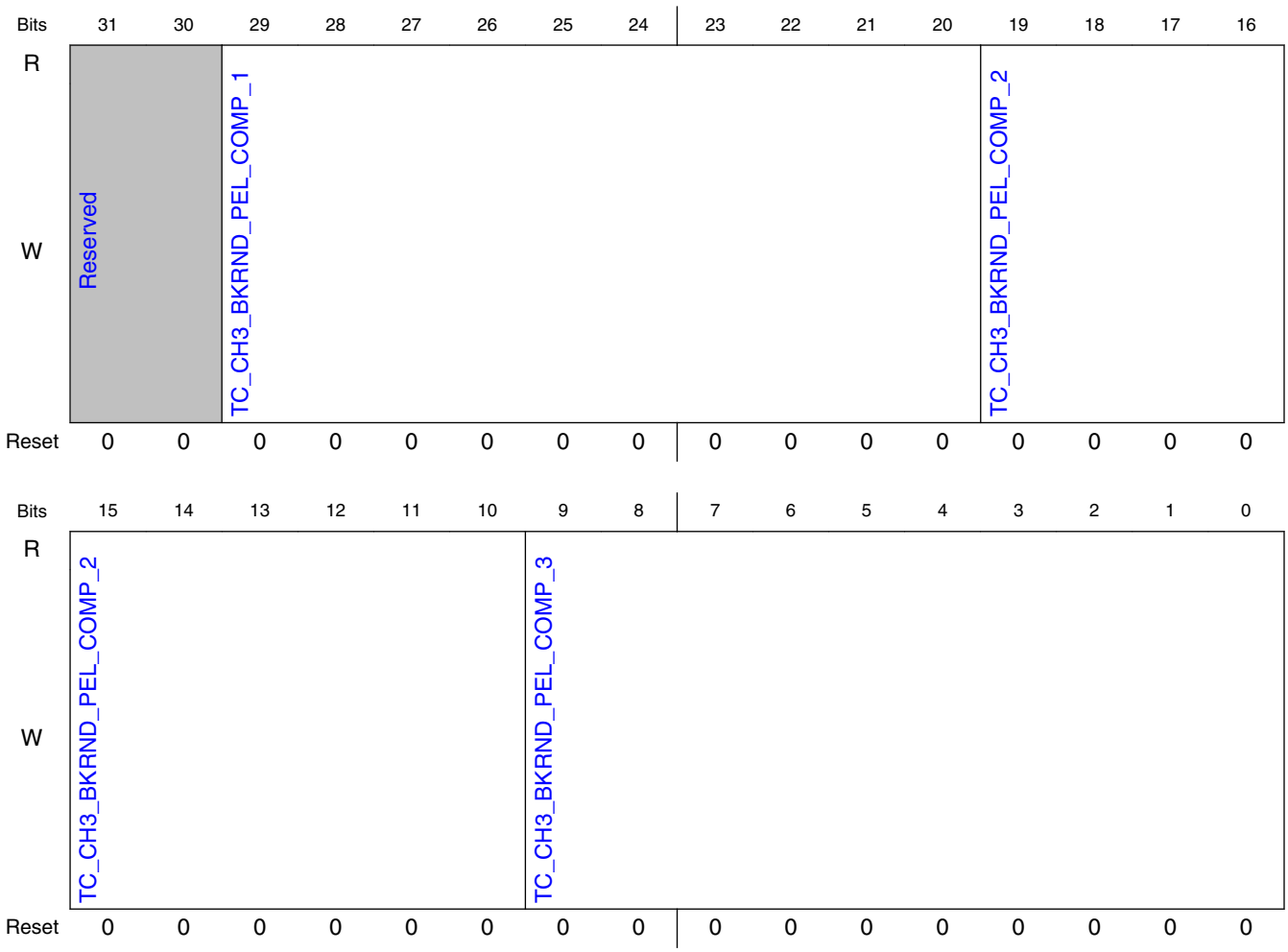
15.3.3.1.26.1 Offset

Register	Offset
TC_CH3_BKRND_REG18	64h

15.3.3.1.26.2 Function

Outside of the Channel_3 pixel window, This background color is put on channel_3. 10 bits per component. Color space depending on programming

15.3.3.1.26.3 Diagram



15.3.3.1.26.4 Fields

Field	Function
31-30 —	Reserved.
29-20 TC_CH3_BKRND_PEL_COMP_1	10-bit component (R or Y)
19-10 TC_CH3_BKRND_PEL_COMP_2	10-bit component (G or Cb)
9-0	10-bit component (B or Cr)

Field	Function
TC_CH3_BKRND_PEL_COMP_3	

15.3.3.1.27 Timing Controller interrupt masks (TC_INTRERRUPT_MASK)

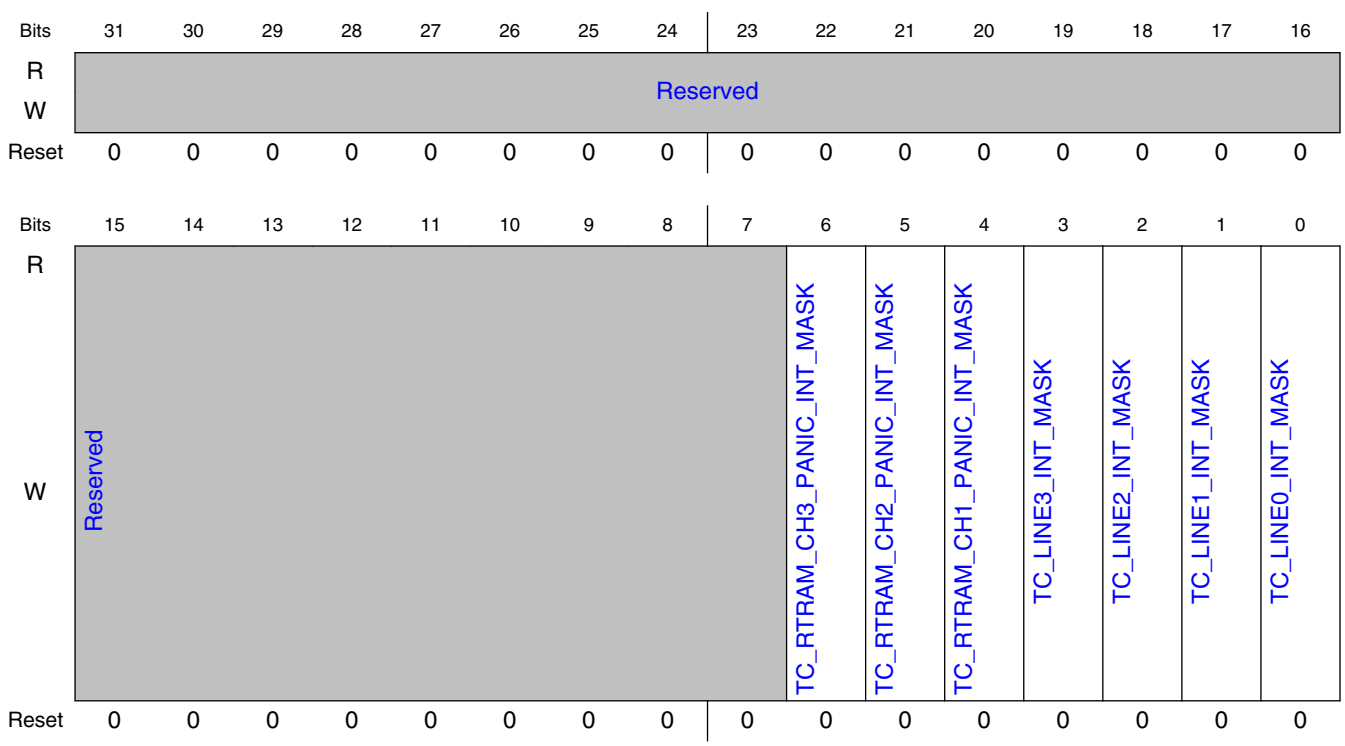
15.3.3.1.27.1 Offset

Register	Offset
TC_INTRERRUPT_MASK	68h

15.3.3.1.27.2 Function

This register has per interrupt mask bits to allow interrupts to be disabled

15.3.3.1.27.3 Diagram



15.3.3.1.27.4 Fields

Field	Function
31-7 —	Reserved.
6 TC_RTRAM_CH3_PANIC_INT_MASK	Mask bit for Channel 3 Panic interrupt asserted by the scaler 0: interrupt is disabled (masked) 1: panic interrupt is enabled
5 TC_RTRAM_CH2_PANIC_INT_MASK	Mask bit for Channel 2 Panic interrupt asserted by the scaler 0: interrupt is disabled (masked) 1: panic interrupt is enabled
4 TC_RTRAM_CH1_PANIC_INT_MASK	Mask bit for Channel 1 Panic interrupt asserted by the scaler 0: interrupt is disabled (masked) 1: panic interrupt is enabled
3 TC_LINE3_INT_MASK	LINE3 interrupt status. LINE3 interrupt can be programmed to occur at any coordinate. 0: interrupt is disabled (masked) 1: LINE3 interrupt has been enabled
2 TC_LINE2_INT_MASK	LINE2 interrupt status. LINE2 interrupt can be programmed to occur at any coordinate. 0: interrupt is disabled (masked) 1: LINE2 interrupt has been enabled
1 TC_LINE1_INT_MASK	LINE1 interrupt status. LINE1 interrupt can be programmed to occur at any coordinate. 0: interrupt is disabled (masked) 1: LINE1 interrupt has been enabled
0 TC_LINE0_INT_MASK	LINE0 interrupt status. LINE0 interrupt can be programmed to occur at any coordinate. 0: interrupt is disabled (masked) 1: LINE0 interrupt has been enabled

15.3.3.1.28 LINE3 interrupt control: Coordinate where line3 interrupt is asserted (TC_LINE3_INT_REG)

15.3.3.1.28.1 Offset

Register	Offset
TC_LINE3_INT_REG	6Ch

15.3.3.1.28.2 Function

The X,Y cooridnates in this register specify where line3 interrupt is asserted as timing controller encounters these coordinates. Can be anywhere on raster table. For historical reasons this is called LINE3 interrupt in RTL

15.3.3.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			TC_LINE3_INT_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			TC_LINE3_INT_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.28.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_LINE3_INT_Y	Y (vertical) coordinate for line3 interrupt
15-13 —	Reserved.
12-0 TC_LINE3_INT_X	X (horizontal) coordinate for line3 interrupt

15.3.3.1.29 LINE4 interrupt control: Coordinate where line4 interrupt is asserted (TC_LINE4_INT_REG)

15.3.3.1.29.1 Offset

Register	Offset
TC_LINE4_INT_REG	70h

15.3.3.1.29.2 Function

The X,Y cooridnates in this register specify where line4 interrupt is asserted as timing controller encounters these coordinates. Can be anywhere on raster table. For historical reasons this is called LINE4 interrupt in RTL.

15.3.3.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			TC_LINE4_INT_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			TC_LINE4_INT_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.3.3.1.29.4 Fields

Field	Function
31-29 —	Reserved.
28-16 TC_LINE4_INT_Y	Y (vertical) coordinate for line4 interrupt
15-13 —	Reserved.
12-0 TC_LINE4_INT_X	X (horizontal) coordinate for line4 interrupt

15.3.3.1.30 For DBY Mode: Controls the assertion and de-assertion DE signal (Overlay channel). (TC_OL_DE_CONTROL_REG)

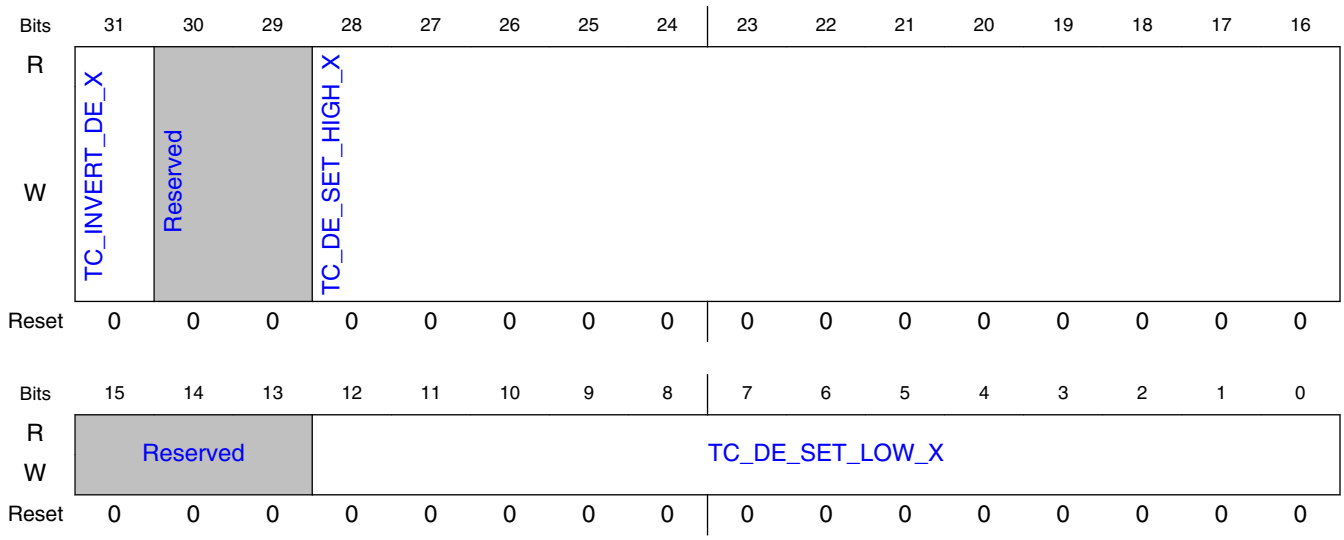
15.3.3.1.30.1 Offset

Register	Offset
TC_OL_DE_CONTROL_REG	74h

15.3.3.1.30.2 Function

This control register pertains to the DBY Overlay (OL) channel. The X cooridnates in this register specify where the DE signal is made 0 or 1 in a line.

15.3.3.1.30.3 Diagram



15.3.3.1.30.4 Fields

Field	Function
31 TC_INVERT_DE_X	0: DE signal is not inverted 1: DE signal is INVERTED
30-29 —	Reserved.
28-16 TC_DE_SET_HIGH_X	X (horizontal) coordinate Where DE control signal is set to 1 in a line.
15-13 —	Reserved.
12-0 TC_DE_SET_LOW_X	X (horizontal) coordinate Where DE control signal is set to 0 in a line.

15.3.3.1.31 For DBY Mode: Controls the assertion and de-assertion DE signal (Base layer (BL) channel). (TC_BL_DE_CONTROL_REG)

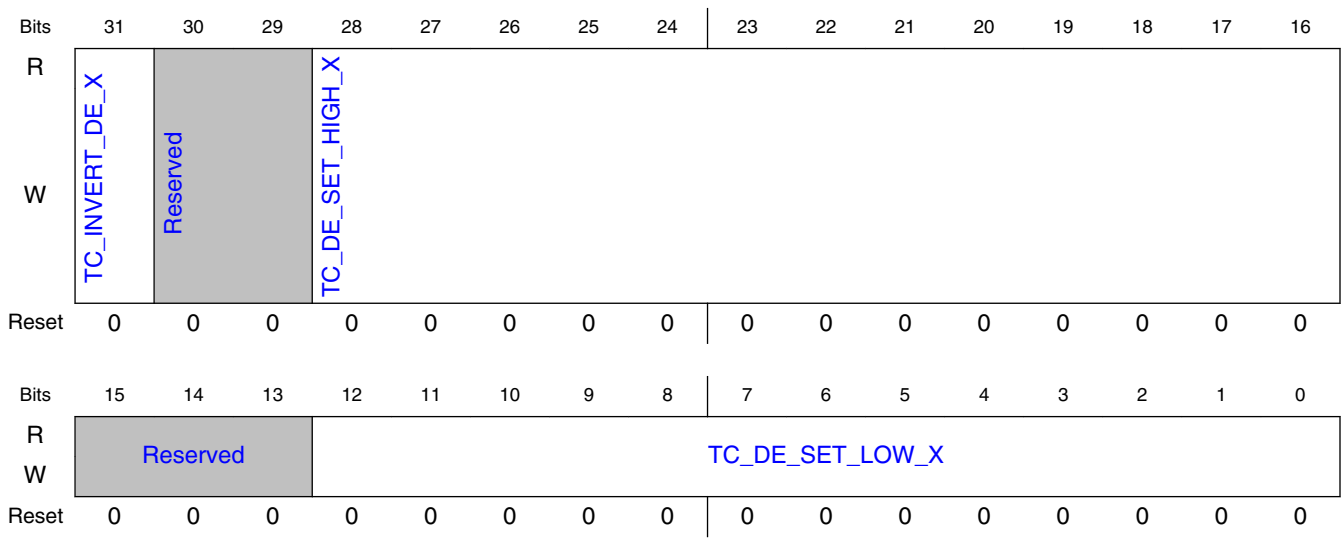
15.3.3.1.31.1 Offset

Register	Offset
TC_BL_DE_CONTROL_REG	78h

15.3.3.1.31.2 Function

This control register pertains to the DBY Base layer (BL) channel. The X cooridnates in this register specify where the DE signal is made 0 or 1 in a line.

15.3.3.1.31.3 Diagram



15.3.3.1.31.4 Fields

Field	Function
31 TC_INVERT_DE_X	0: DE signal is not inverted 1: DE signal is INVERTED
30-29 —	Reserved.
28-16 TC_DE_SET_HIGH_X	X (horizontal) coordinate Where DE control signal is set to 1 in a line.
15-13 —	Reserved.
12-0	X (horizontal) coordinate Where DE control signal is set to 0 in a line.

Field	Function
TC_DE_SET_LOW_X	

15.3.3.1.32 For DBY Mode: Controls the assertion and de-assertion DE signal (Enhancement layer (EL) channel). (TC_EL_DE_CONTROL_REG)

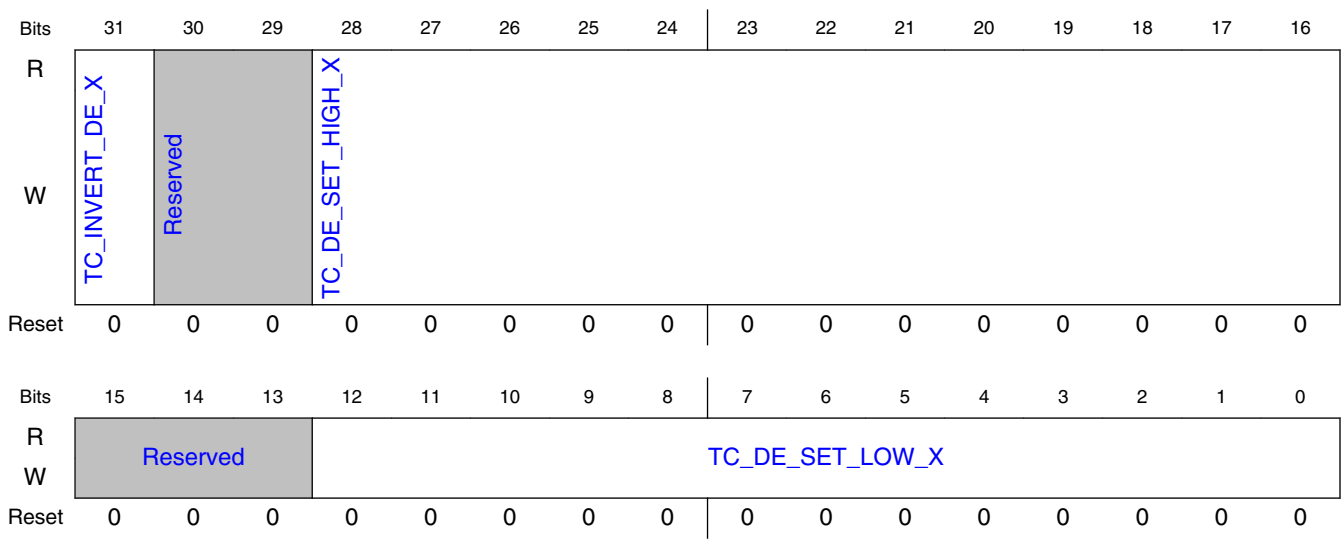
15.3.3.1.32.1 Offset

Register	Offset
TC_EL_DE_CONTROL_REG	7Ch

15.3.3.1.32.2 Function

This control register pertains to the DBY Enhancement layer (EL) channel. The X cooridnates in this register specify where the DE signal is made 0 or 1 in a line.

15.3.3.1.32.3 Diagram



15.3.3.1.32.4 Fields

Field	Function
31	0: DE signal is not inverted

Table continues on the next page...

Field	Function
TC_INVERT_DE_X	1: DE signal is INVERTED
30-29 —	Reserved.
28-16 TC_DE_SET_HIGH_X	X (horizontal) coordinate Where DE control signal is set to 1 in a line.
15-13 —	Reserved.
12-0 TC_DE_SET_LOW_X	X (horizontal) coordinate Where DE control signal is set to 0 in a line.

15.4 Context Load (CTX_LD)

15.4.1 Overview

The Context Loader (CTX_LD) module fetches and loads the next state of the display from system memory.

15.4.1.1 Block Diagram

The diagram below indicates the functional components and data flow in the CTX_LD module.

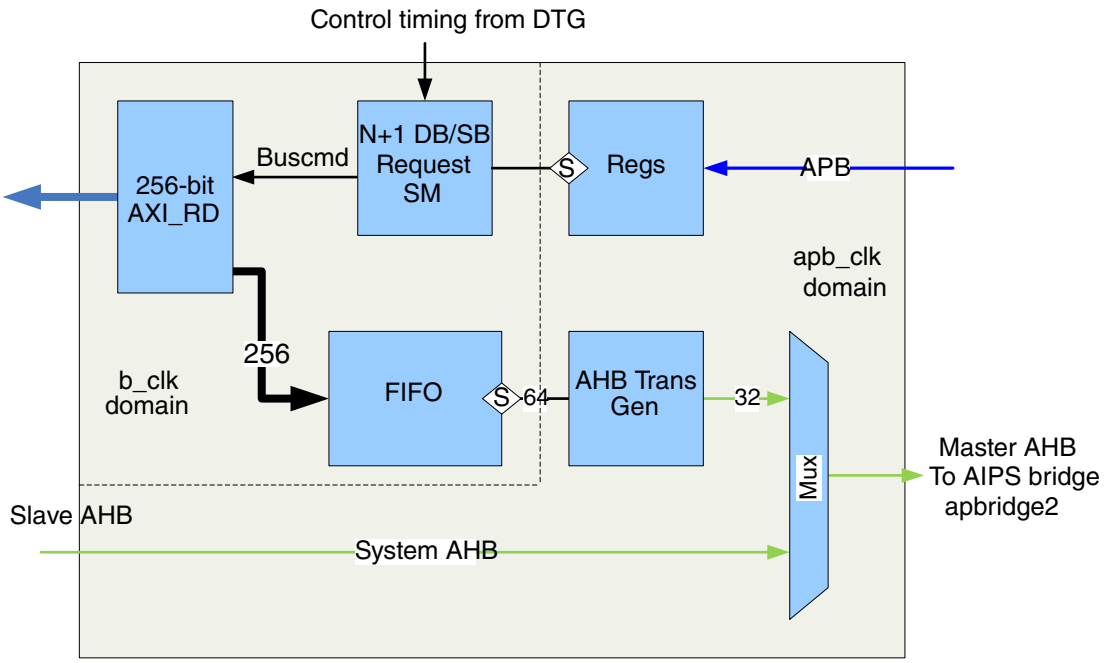


Figure 15-10. CTX_LD Block Diagram

- The request state machine issues transactions to fetch the register addr/data pairs from system memory (see System Memory Display State Format). It manages the FIFO to keep the register loading operation advancing until all the DB and SB registers are loaded.
- The loading sequence is described below in section Display State Loading Sequence. A likely size for the FIFO is 512 bytes. Each request to system memory by the state machine is half the size of the FIFO. When the FIFO is half full/empty, the SM requests another half FIFO size of addr/data pairs from system memory. This process proceeds until all the DB and SB registers are loaded.
- The AHB transaction generator creates the transactions needed to empty the FIFO and load DC registers contents.

15.4.2 Functional Description

Two events indicate when the next display context is loaded. The timing of the CTX_LD is controlled by the Display Timing Generator (DTG).

The first event, marked by the red dot in the DTG chapter's timing example (Display Timing and Window Overlay section), and also in the timing diagram in the section Timing Sequence, is for components that implement double buffered registers and require their next state to be loaded into shadow registers during the active display time. This interval is defined as the db_region, or double buffered region.

The second event, marked by the orange dot, occurs after the active display region for components that don't support double buffered registers and require the next state to be loaded during the vertical blanking time. This interval is defined as the sb_region, or single buffered region. The second region is subdivided into two regions. The first is high priority and critical from a timing perspective. The second includes all single buffered components that aren't critical from a timing perspective. These two sub regions are defined as sb_region.high_prioity and sb_region.low_priority.

The critical region should include loading component parameters that need to be issued first. As an example, the pixel prefetch components (DPR) would need to be programmed first and kicked to start their prefetch operation before the first scan line in the active display region. The DPR would not be enabled to start after the HDR10 LUTs are loaded, as an example. The idea is that low priority registers are being loaded in parallel with the DPR's prefetch operation. This provides the best opportunity to prefetch data BEFORE the first pixel is needed to drive the display when the active region begins

As stated before, the loading sequence has the following regions.

```
1. db_region; Region where all double buffered registers are loaded.
2. sb_region; Region where all single buffered registers are loaded.
   a. sb_region.high_priority - critical time to load single buffered
         components.
   b. sb_region.low_priority - time to load components not critical
         on timing.
```

The timing of db/sb regions are controlled and timed by the DTG, or respective display timing generator. These event trigger times are programmable based on vertical timing parameters in the DTG that go high for one scan line interval. The rising edge of these events cause the CTX_LD engine to fetch the next state parameters from DRAM.

The components that implement double buffered registers should transfer the shadow registers into the active registers when ctx_ld_sb_en goes active. This is the trigger used to start loading single buffered registers, and is used to allow the double buffered register components to start processing for the next frame. The idea is that the double buffered components should be enabled shortly after the active display region is complete, and the display enters the vertical front porch region. When the trigger is received by the double buffered component, then the component begins processing for trace N+1. The earliest possible event to enable the DB components is necessary to allow the most time to complete processing for trace N+1 before the active display area becomes active.

Components that implement single buffer registers receives their new values from the ctx_ld module after the ctx_ld_sb_en signal goes high. This is the event that indicates the CTX_LD fetches and loads the single buffered components in the display SS. Single buffered components starts processing using their last register write, or can also use the ctx_ld_sb_hp_kick output signal to "kick" their processing for trace N+1.

NOTE

It could be valid that the CTX_LD is NOT enabled to fetch the next state for N+1, i.e. there is no state change and the display system repeats the frame N+1 with the existing parameters already programmed. In this case, the CTX_LD still receives the ctx_ld_[sb|db]_en signal inputs and the CTX_LD does not issue DRAM accesses to load the parameters. The ctx_ld_sb_hp_kick_en and the ctx_ld_sb_lp_kick_en output strobes cycle low/high/low to indicate all modules should start processing using their existing parameters. Or, 0 registers were fetched in 0 time, and the process completes by issuing both ctx_ld_sb_hp_kick and ctx_ld_sb_lp_kick to “kick” all the SB components so processing can begin. The ctx_ld_sb_[hp|lp]_kick signal is asserted just after it receives the ctx_ld_sb_en input strobe if the CTX_LD register loading sequence was not enabled during trace N. This allows to restart the components with the previous state used in N.

Double buffered components use either vsync or the ctx_ld_sb_hp_kick signal to start processing for trace N+1.

Once the display state is completely loaded, both those that are doubled buffered and those that use a single buffering scheme, all components commence operation to refresh the display. Some of the next state parameters that are loaded by the CTX_LD engine are defined as, but not limited to, the following:

- The source buffer addresses, formats, and rotation parameters fetched by each DPR.
- The scale width, height, X/Y scaling factors and initial offsets, and coeffs for the scaling operation
- X,Y min/max extents for the three surfaces in the output display.
- WR_SCL and RD_SRC buffer addresses, pitch, and formats, if needed.
- Data flow configuration options (which scaler goes to DRAM if any).

The Context Loader Engine is also capable of loading the Dolby Vision tables or the HDR10 LUT entries from DDR. When the CTX_LD is done with the next state loading sequence, it returns to its idle state by auto clearing the ENABLE bit in the CTRL_STATUS register.

15.4.2.1 Timing Sequence

The timing diagram below indicates the sequence over the trace time N, in preparation for trace N+1. The DB load strobe starts the CTX_LD sequence. The first ctx_ld_db_en event that occurs AFTER the PIO write to set CTX_LD_CTRL[ENABLE] commences the register loading sequence for display trace N+1.

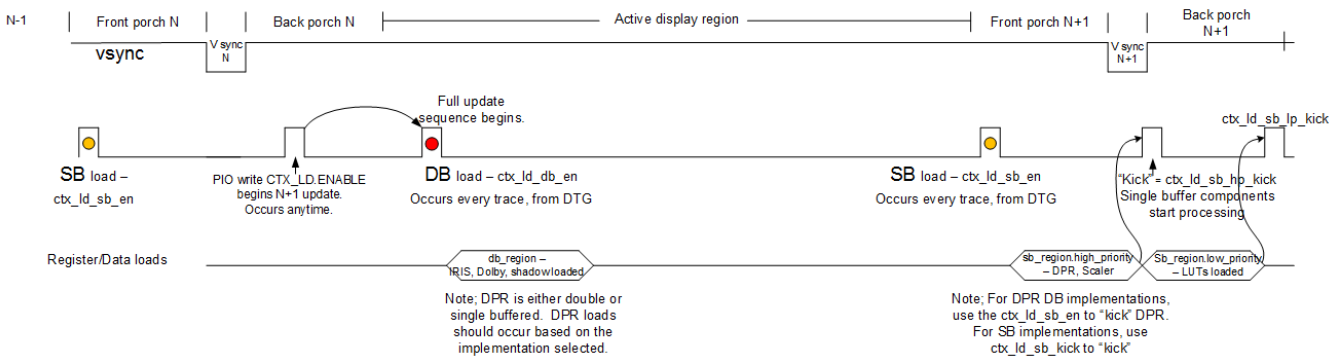


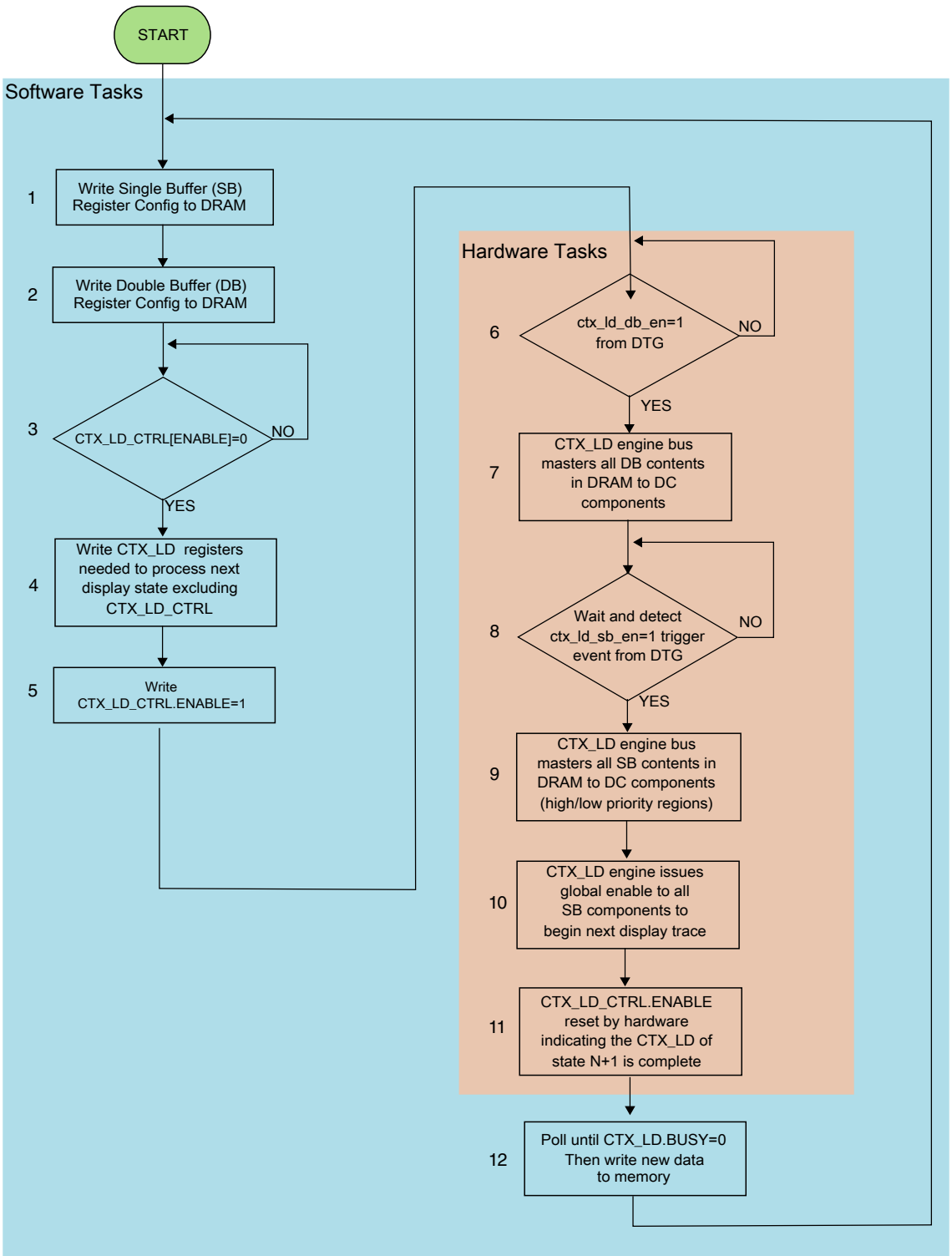
Figure 15-11. CTX_LD Timing Sequence

NOTE

If the PIO write to set the CTX_LD_CTRL[ENABLE] comes after the ctx_ld_db_en event AND before the ctx_ld_sb_en event, nothing occurs. The process to load DB and SB registers always starts at the ctx_ld_db_en event. So, DB registers are loaded first, and then the SB registers are loaded second to complete the full sequence.

15.4.2.2 Display State Loading Sequence

The sequence to properly load all registers for the next display state N+1 is shown below.



See detailed descriptions of these steps below for more information.

Figure 15-12. Display State Loading Sequence

NOTE: “SW” indicates an action taken by the software, and “HW” indicates action on behalf of the hardware.

1. SW: Write all single buffered register configurations in DRAM at address SB_BASE_ADDR. This should be a new address not used in the previous CTX_LD sequence.
2. SW: Write all double buffered register configurations in DRAM at address DB_BASE_ADDR. This should be a new address not used in the previous CTX_LD sequence.
3. SW: Read the CTX_LD_CTRL[ENABLE] = 0. This indicates the CTX_LD is idle and ready for the next state update.
   a. CTX_LD_CTRL[ENABLE] = 1, wait, goto #3
   b. CTX_LD_CTRL[ENABLE] = 0, goto #4
4. SW: Write all CTX_LD registers needed to process the next display state excluding CTX_LD_CTRL. CTX_LD_SB_HP_CNT cannot be 0 if CTX_LD_SB_LP_CNT != 0.
5. SW: Write the CTX_LD_CTRL[ENABLE] = 1. This bit can only be set by SW. Note, it is reset by HW when the CTX_LD engine is complete loading the entire next state. When the ENABLE bit is high, it indicates that
   a. The CTX_LD is ready to start the loading sequence when the ctx_ld_db_en input goes high.
   b. or CTX_LD is currently busy loading the state N+1.
6. HW: Wait and detect the ctx_ld_db_en trigger input event from the DTG. This occurs around the timing indicated by the red dot in the display timing diagram.
7. HW: The CTX_LD engine bus masters all double buffered register contents from DRAM to DC components.
8. HW: Wait and detect the ctx_ld_sb_en trigger event input from the DTG. Generally, this timing is indicated by the orange dot and is just after the completion of the active display region (start of vertical front porch).
9. HW: The ctx_ld engine bus masters all single buffered register contents from DRAM to DC components. Waits for the last transaction response via AHB. a. NOTE: During this interval, there are two regions to load registers divided into the high/low priority regions.
10. HW: The ctx_ld engine issues a global enable control signal to all single buffered components to begin processing for the next display trace after each of the high/low priority regions are complete. These control signals are the ctx_ld_sb_hp_kick and ctx_ld_sb_lp_kick. DC components that are considered high priority should use ctx_ld_sb_hp_kick signal to indicate when they should start processing. Likewise for low priority DC components.
11. The CTX_LD_CTRL[ENABLE] bit is reset by HW indicating the CTX_LD of state N+1 is complete.

12. Goto #1. When CTX_LD_CTRL[ENABLE] set by SW, it should not update the DRAM contents the CTX_LD uses to set the state for display trace N+1. SW should also refrain from writing CTX_LD registers that control the next context load for display trace N+1. If alternate buffers in DRAM are used for another display state, then they can be preloaded at a different address since the CTX_LD engine only uses the buffers pointed to by the base address registers. When the CTX_LD_CTRL[ENABLE] bit is low, SW can program CTX_LD registers and set CTX_LD_CTRL[ENABLE] to initiate the process again.

15.4.2.3 System Memory Display State Format

The format of the entries in DRAM used to load a register is indicated in the table below:

Table 15-7. 64-bit Address Offset and Data Pair in Memory

Address in DRAM	63:32 DC register address offset	31:0 DC register value
ctx_[db|sb]_base_addr	First reg offset to be loaded in DC	Respective first reg data value.
ctx_[db|sb]_base_addr + 8	Second reg offset to loaded in DC	Respective second reg value.
...	...	...
ctx_[db|sb]_base_addr + 8*count	CTX_[DB|SB]_COUNT reg offset in DC	Respective last reg value.

15.4.2.4 Arbitration

The MUX arbitrates between transactions from the AHB transaction generator and the system AHB bus. When the CTX_LD state machine is actively fetching register contents from system memory and loading registers in the DC, accesses from the slave AHB bus (the host) are arbitrated. This occurs during the time when the CTX_LD is loading DB or SB registers. So, if concurrent access is required by the CTX_LD engine AND the host (via AHB), then the accesses are arbitrated in an alternating sequence. If the host is attempting a burst, a single burst is consider as a single access. This access completes, and then the CTX_LD engine wins the right to initiate a single burst.

As a programmable option (NOT the default), the host AHB bus can be blocked while the CTX_LD engine is actively processing transactions. This is intended to allow the CTX_LD to completely load the register state without interruption. It is expected that blocking the host is not permitted for long intervals. This option is an intent to avoid programming issues while the CTX_LD and host are concurrently accessing registers. Another way to view this is that the CTX_LD always has priority over the slave AHB accesses via the host. The following CTX_LD operational states would stall the slave AHB bus.

- CTX_LD request state machine is active and generating requests.
- CTX_LD request state machine is done, but the transaction FIFO is not empty and still has transactions to complete to components within the DC.

NOTE

System software is responsible for guaranteeing the system AHB transactions and the CTX_LD transactions do not violate the programming and operation model of the DC SS. This is required regardless if the arbitration scheme, or the round robin scheme, is used for processing transactions from the two sources.

15.4.3 Memory Map and Registers

15.4.3.1 register descriptions

15.4.3.1.1 CTX_LD Memory map

ctx_ld base address: 2_3000h

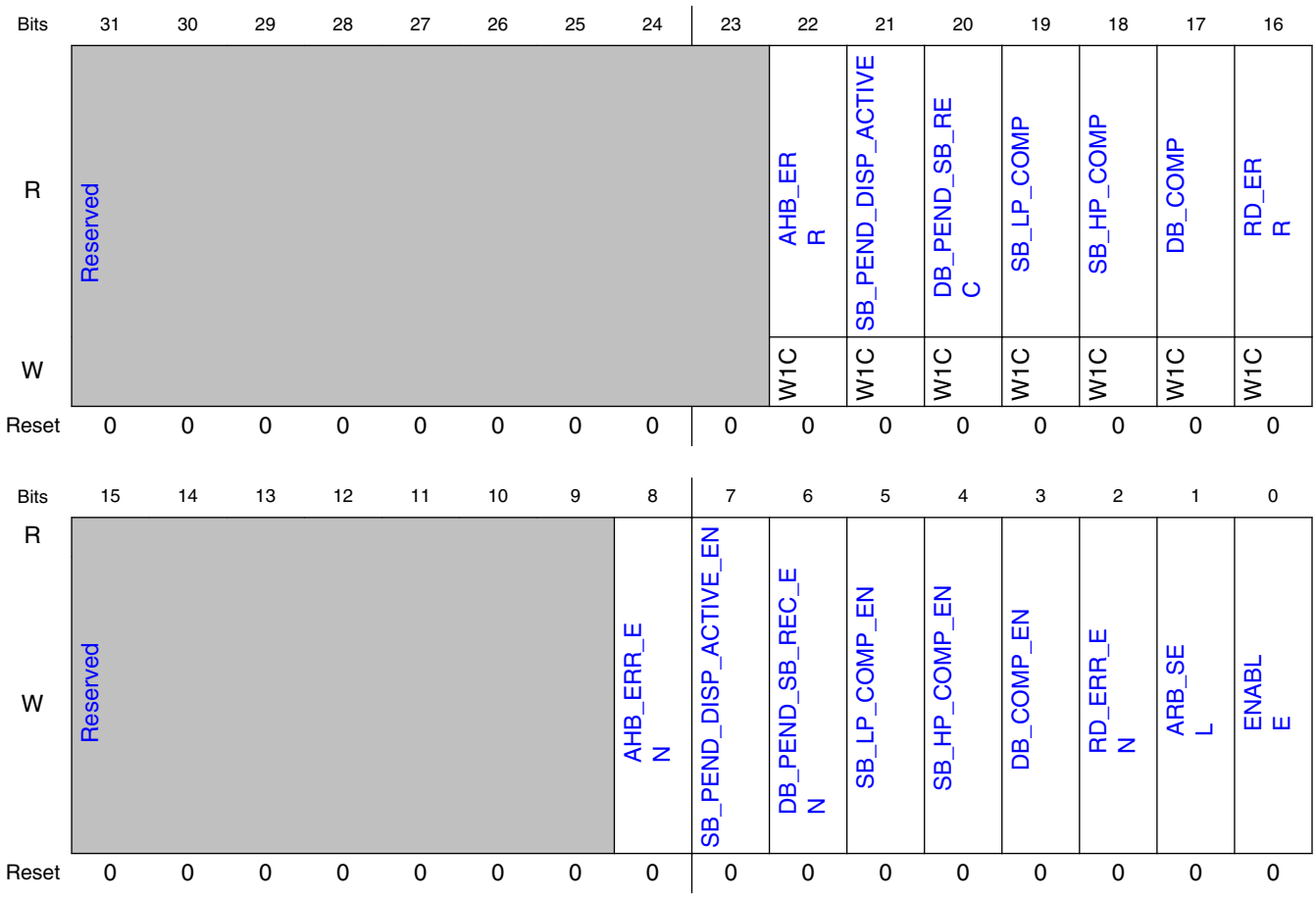
Offset	Register	Width (In bits)	Access	Reset value
0h	Control status register for Context Loader. (CTRL_STATUS)	32	RW	0000_0000h
4h	Control status register for Context Loader. (CTRL_STATUS_SET)	32	RW	0000_0000h
8h	Control status register for Context Loader. (CTRL_STATUS_CLR)	32	RW	0000_0000h
Ch	Control status register for Context Loader. (CTRL_STATUS_TOG)	32	RW	0000_0000h
10h	DRAM addr for double buffered register fetch. (DB_BASE_ADDR)	32	RW	0000_0000h
14h	Double buffer register count (DB_COUNT)	32	RW	0000_0000h
18h	DRAM addr for single buffered registers. (SB_BASE_ADDR)	32	RW	0000_0000h
1Ch	Single buffer register count (SB_COUNT)	32	RW	0000_0000h
20h	AHB address with error response. (AHB_ERR_ADDR)	32	RO	0000_0000h

15.4.3.1.2 Control status register for Context Loader. (CTRL_STATUS)

15.4.3.1.2.1 Offset

Register	Offset
CTRL_STATUS	0h

15.4.3.1.2.2 Diagram



15.4.3.1.2.3 Fields

Field	Function
31-23 —	Reserved
22 AHB_ERR	AHB error Status bit to indicate when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped. This status bit can be set ONLY for CTX_LD initiated transactions

Table continues on the next page...

Field	Function
	that receive an AHB error. System transactions that respond with an error are passed through, and the initiating master manages the error response at the source.
21 SB_PEND_DISP_ACTIVE	Single/active region overlap Status bit to indicate when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
20 DB_PEND_SB_REC	Double/single region overlap Status bit to indicate when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
19 SB_LP_COMP	Single buffer low priority region loading complete IRQ enable Status bit to indicate when the low priority SB region loading sequence is complete.
18 SB_HP_COMP	Single buffer high priority region loading complete Status bit to indicate when the high priority SB region loading sequence is complete.
17 DB_COMP	Double buffer region loading complete Status bit to indicate when the DB (double buffer) region loading sequence is complete.
16 RD_ERR	AXI read error Status bit to indicate when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
15-9 —	Reserved
8 AHB_ERR_EN	AHB error IRQ enable Enable and interrupt when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped.
7 SB_PEND_DISP_ACTIVE_EN	Single/active region overlap interrupt enable. Enable an interrupt when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
6 DB_PEND_SB_REC_EN	Double/single region overlap interrupt enable Enable an interrupt when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
5 SB_LP_COMP_EN	Single buffer low priority region loading complete IRQ enable Enable an interrupt when the low priority SB region loading sequence is complete.
4 SB_HP_COMP_EN	Single buffer high priority region loading complete IRQ enable Enable an interrupt when the high priority SB region loading sequence is complete.
3 DB_COMP_EN	Double buffer region loading complete IRQ enable Enable an interrupt when the DB (double buffer) region loading sequence is complete.
2 RD_ERR_EN	AXI read error IRQ enable

Table continues on the next page...

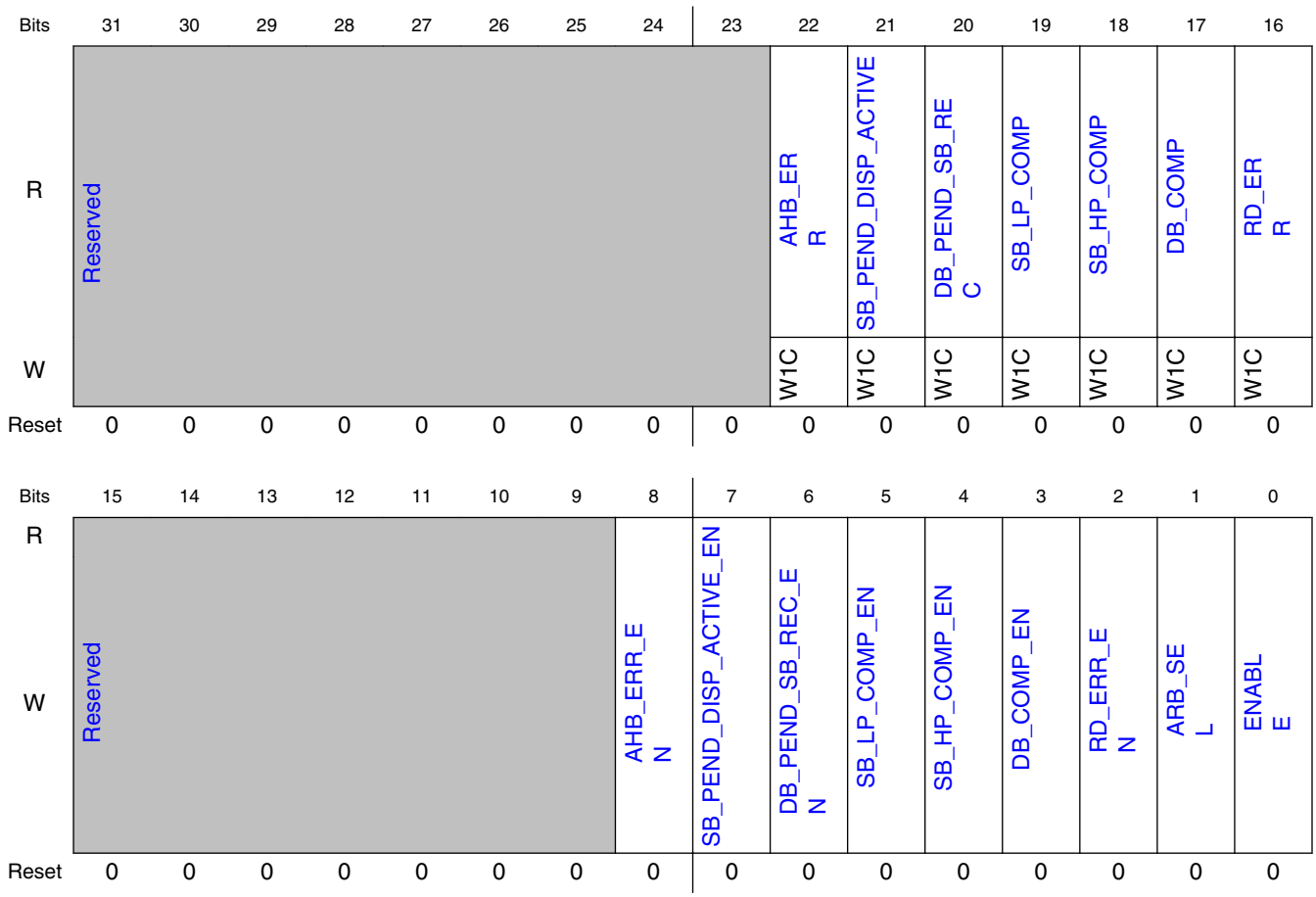
Field	Function
	Enable an interrupt when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
1 ARB_SEL	Arbitration select This selects between always choosing the CTX_LD while it's active verses a round robin arbitration scheme of equal weight between the CTX_LD and the system register access bus. • 0 - Round robin arbitration at all times. • 1 - CTX_LD gets priority while it's active.
0 ENABLE	Enable/Busy Enable bit for the context loader. The context loader does not start the register loading sequence if this bit is not set. SW can set this bit only when it is logic 0 (never set this bit if it is already logic 1). The CTX_LD hardware resets this bit when the entire register loading sequence is complete. It is complete after the DB and SB regions are loaded and the CTX_LD state machine returns to the idle state to allow the next context loading sequence to be enabled.

15.4.3.1.3 Control status register for Context Loader. (CTRL_STATUS_SET)

15.4.3.1.3.1 Offset

Register	Offset
CTRL_STATUS_SET	4h

15.4.3.1.3.2 Diagram



15.4.3.1.3.3 Fields

Field	Function
31-23 —	Reserved
22 AHB_ERR	AHB error Status bit to indicate when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped. This status bit can be set ONLY for CTX_LD initiated transactions that receive an AHB error. System transactions that respond with an error are passed through, and the initiating master manages the error response at the source.
21 SB_PEND_DISP_ACTIVE	Single/active region overlap Status bit to indicate when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
20 DB_PEND_SB_REC	Double/single region overlap Status bit to indicate when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region

Table continues on the next page...

Field	Function
	in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
19 SB_LP_COMP	Single buffer low priority region loading complete IRQ enable Status bit to indicate when the low priority SB region loading sequence is complete.
18 SB_HP_COMP	Single buffer high priority region loading complete Status bit to indicate when the high priority SB region loading sequence is complete.
17 DB_COMP	Double buffer region loading complete Status bit to indicate when the DB (double buffer) region loading sequence is complete.
16 RD_ERR	AXI read error Status bit to indicate when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
15-9 —	Reserved
8 AHB_ERR_EN	AHB error IRQ enable Enable and interrupt when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped.
7 SB_PEND_DISP_ACTIVE_EN	Single/active region overlap interrupt enable. Enable an interrupt when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
6 DB_PEND_SB_REC_EN	Double/single region overlap interrupt enable Enable an interrupt when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
5 SB_LP_COMP_EN	Single buffer low priority region loading complete IRQ enable Enable an interrupt when the low priority SB region loading sequence is complete.
4 SB_HP_COMP_EN	Single buffer high priority region loading complete IRQ enable Enable an interrupt when the high priority SB region loading sequence is complete.
3 DB_COMP_EN	Double buffer region loading complete IRQ enable Enable an interrupt when the DB (double buffer) region loading sequence is complete.
2 RD_ERR_EN	AXI read error IRQ enable Enable an interrupt when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
1 ARB_SEL	Arbitration select This selects between always choosing the CTX_LD while it's active verses a round robin arbitration scheme of equal weight between the CTX_LD and the system register access bus. • 0 - Round robin arbitration at all times. • 1 - CTX_LD gets priority while it's active.
0 ENABLE	Enable/Busy Enable bit for the context loader. The context loader does not start the register loading sequence if this bit is not set. SW can set this bit only when it is logic 0 (never set this bit if it is already logic 1). The CTX_LD

Field	Function
	hardware resets this bit when the entire register loading sequence is complete. It is complete after the DB and SB regions are loaded and the CTX_LD state machine returns to the idle state to allow the next context loading sequence to be enabled.

15.4.3.1.4 Control status register for Context Loader. (CTRL_STATUS_CLR)

15.4.3.1.4.1 Offset

Register	Offset
CTRL_STATUS_CLR	8h

15.4.3.1.4.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved									AHB_ERR	SB_PEND_DISP_ACTIVE	DB_PEND_SB_REC	SB_LP_COMP	SB_HP_COMP	DB_COMP	RD_ERR
W										W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved							AHB_ERR_EN	SB_PEND_DISP_ACTIVE_EN	DB_PEND_SB_REC_EN	SB_LP_COMP_EN	SB_HP_COMP_EN	DB_COMP_EN	RD_ERR_EN	ARB_SEL	ENABLE
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.3.1.4.3 Fields

Field	Function
31-23 —	Reserved
22 AHB_ERR	AHB error Status bit to indicate when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped. This status bit can be set ONLY for CTX_LD initiated transactions that receive an AHB error. System transactions that respond with an error are passed through, and the initiating master manages the error response at the source.
21 SB_PEND_DISP_ACTIVE	Single/active region overlap Status bit to indicate when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
20 DB_PEND_SB_REC	Double/single region overlap Status bit to indicate when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
19 SB_LP_COMP	Single buffer low priority region loading complete IRQ enable Status bit to indicate when the low priority SB region loading sequence is complete.
18 SB_HP_COMP	Single buffer high priority region loading complete Status bit to indicate when the high priority SB region loading sequence is complete.
17 DB_COMP	Double buffer region loading complete Status bit to indicate when the DB (double buffer) region loading sequence is complete.
16 RD_ERR	AXI read error Status bit to indicate when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
15-9 —	Reserved
8 AHB_ERR_EN	AHB error IRQ enable Enable and interrupt when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped.
7 SB_PEND_DISP_ACTIVE_EN	Single/active region overlap interrupt enable. Enable an interrupt when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
6 DB_PEND_SB_REC_EN	Double/single region overlap interrupt enable Enable an interrupt when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
5	Single buffer low priority region loading complete IRQ enable Enable an interrupt when the low priority SB region loading sequence is complete.

Table continues on the next page...

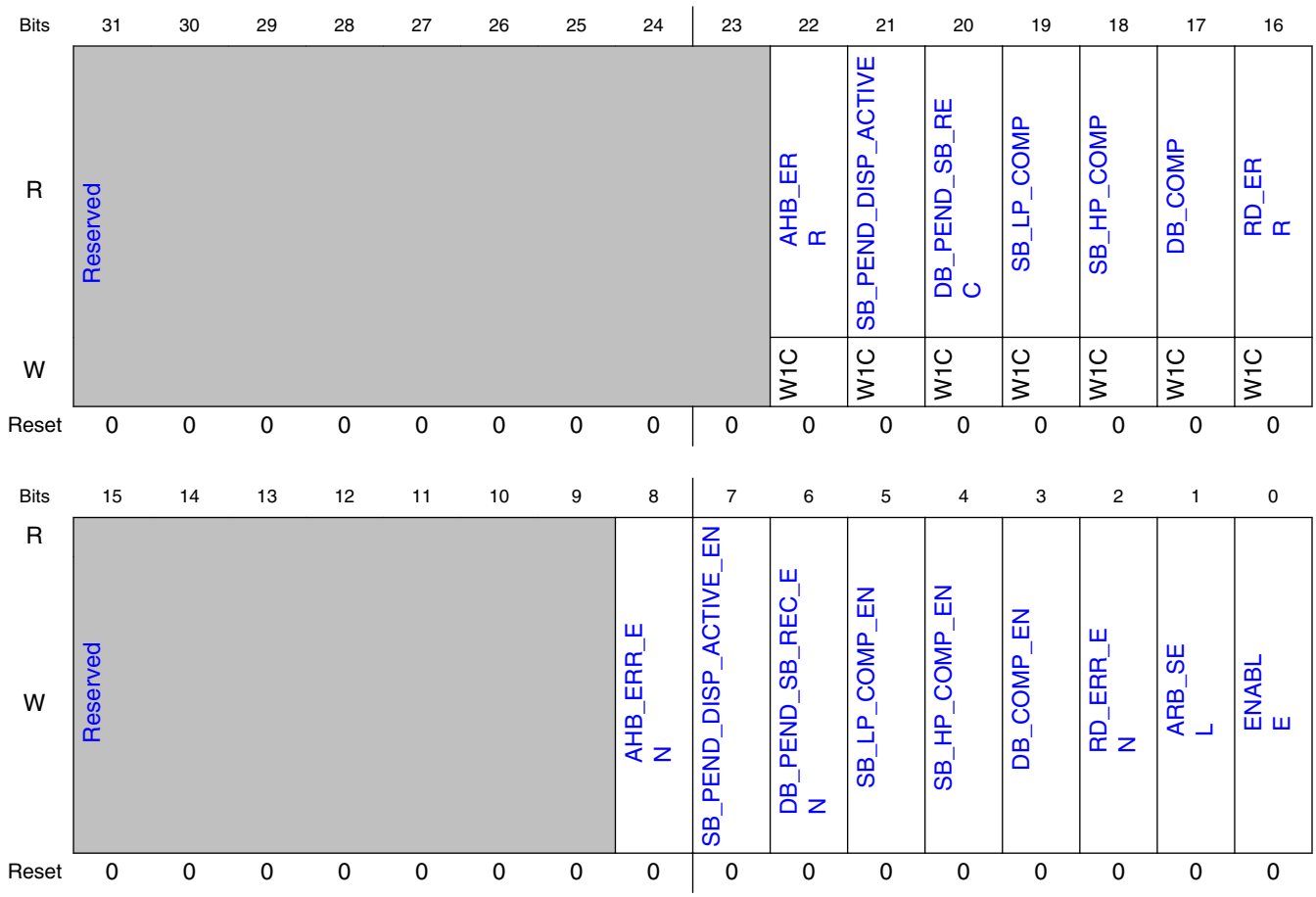
Field	Function
SB_LP_COMP_EN	
4 SB_HP_COMP_EN	Single buffer high priority region loading complete IRQ enable Enable an interrupt when the high priority SB region loading sequence is complete.
3 DB_COMP_EN	Double buffer region loading complete IRQ enable Enable an interrupt when the DB (double buffer) region loading sequence is complete.
2 RD_ERR_EN	AXI read error IRQ enable Enable an interrupt when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
1 ARB_SEL	Arbitration select This selects between always choosing the CTX_LD while it's active verses a round robin arbitration scheme of equal weight between the CTX_LD and the system register access bus. • 0 - Round robin arbitration at all times. • 1 - CTX_LD gets priority while it's active.
0 ENABLE	Enable/Busy Enable bit for the context loader. The context loader does not start the register loading sequence if this bit is not set. SW can set this bit only when it is logic 0 (never set this bit if it is already logic 1). The CTX_LD hardware resets this bit when the entire register loading sequence is complete. It is complete after the DB and SB regions are loaded and the CTX_LD state machine returns to the idle state to allow the next context loading sequence to be enabled.

15.4.3.1.5 Control status register for Context Loader. (CTRL_STATUS_TOG)

15.4.3.1.5.1 Offset

Register	Offset
CTRL_STATUS_TOG	Ch

15.4.3.1.5.2 Diagram



15.4.3.1.5.3 Fields

Field	Function
31-23 —	Reserved
22 AHB_ERR	AHB error Status bit to indicate when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped. This status bit can be set ONLY for CTX_LD initiated transactions that receive an AHB error. System transactions that respond with an error are passed through, and the initiating master manages the error response at the source.
21 SB_PEND_DISP_ACTIVE	Single/active region overlap Status bit to indicate when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
20 DB_PEND_SB_REC	Double/single region overlap Status bit to indicate when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region

Table continues on the next page...

Field	Function
	in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
19 SB_LP_COMP	Single buffer low priority region loading complete IRQ enable Status bit to indicate when the low priority SB region loading sequence is complete.
18 SB_HP_COMP	Single buffer high priority region loading complete Status bit to indicate when the high priority SB region loading sequence is complete.
17 DB_COMP	Double buffer region loading complete Status bit to indicate when the DB (double buffer) region loading sequence is complete.
16 RD_ERR	AXI read error Status bit to indicate when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
15-9 —	Reserved
8 AHB_ERR_EN	AHB error IRQ enable Enable and interrupt when the internal AHB bus returns an error response. This reflects an error on the AHB bus output from the CTX_LD to the DCSS memory mapped components. This could occur when accessing a region that is not mapped.
7 SB_PEND_DISP_ACTIVE_EN	Single/active region overlap interrupt enable. Enable an interrupt when the SB region is still actively loading registers to the display subsystem AND the display enters the active display time. This would indicate the SB registers are not loaded in time and the display trace could be corrupt. Unpredicatable behaviour could result if this occurs.
6 DB_PEND_SB_REC_EN	Double/single region overlap interrupt enable Enable an interrupt when the DB region is still actively loading registers to the display subsystem AND a kick to load the SB region is received. This could indicate a timing or BW issue with loading the DB region in time before the SB region register loading should begin. Unpredicatable behaviour could result if this occurs.
5 SB_LP_COMP_EN	Single buffer low priority region loading complete IRQ enable Enable an interrupt when the low priority SB region loading sequence is complete.
4 SB_HP_COMP_EN	Single buffer high priority region loading complete IRQ enable Enable an interrupt when the high priority SB region loading sequence is complete.
3 DB_COMP_EN	Double buffer region loading complete IRQ enable Enable an interrupt when the DB (double buffer) region loading sequence is complete.
2 RD_ERR_EN	AXI read error IRQ enable Enable an interrupt when the AXI read bus returns a SLV_ERR or DEC_ERR. This could result in an illegal or unmapped address programmed into the base address registers.
1 ARB_SEL	Arbitration select This selects between always choosing the CTX_LD while it's active verses a round robin arbitration scheme of equal weight between the CTX_LD and the system register access bus. • 0 - Round robin arbitration at all times. • 1 - CTX_LD gets priority while it's active.
0 ENABLE	Enable/Busy Enable bit for the context loader. The context loader does not start the register loading sequence if this bit is not set. SW can set this bit only when it is logic 0 (never set this bit if it is already logic 1). The CTX_LD

Field	Function
	hardware resets this bit when the entire register loading sequence is complete. It is complete after the DB and SB regions are loaded and the CTX_LD state machine returns to the idle state to allow the next context loading sequence to be enabled.

15.4.3.1.6 DRAM addr for double buffered register fetch. (DB_BASE_ADDR)

15.4.3.1.6.1 Offset

Register	Offset
DB_BASE_ADDR	10h

15.4.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	DB_BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DB_BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.3.1.6.3 Fields

Field	Function
31-0 DB_BASE_ADDR	Double buffered registers base address. Stores the system start address(DRAM) of double buffered registers.

15.4.3.1.7 Double buffer register count (DB_COUNT)

15.4.3.1.7.1 Offset

Register	Offset
DB_COUNT	14h

15.4.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	DB_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.3.1.7.3 Fields

Field	Function
31-16 —	Reserved
15-0 DB_COUNT	Double buffered region fetch count Number of 64 bit entries of double buffered registers to fetch and load.

15.4.3.1.8 DRAM addr for single buffered registers. (SB_BASE_ADDR)

15.4.3.1.8.1 Offset

Register	Offset
SB_BASE_ADDR	18h

15.4.3.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	SB_BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	SB_BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.3.1.8.3 Fields

Field	Function
31-0 SB_BASE_ADDR	Single buffered register base address. Stores the system start address(DRAM) of single buffered registers.

15.4.3.1.9 Single buffer register count (SB_COUNT)

15.4.3.1.9.1 Offset

Register	Offset
SB_COUNT	1Ch

15.4.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	LP_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HP_COUNT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.4.3.1.9.3 Fields

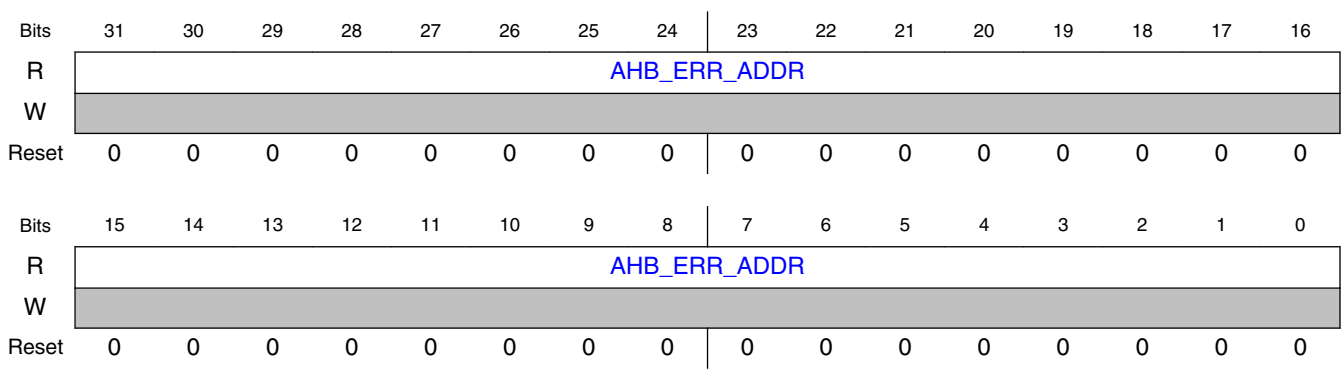
Field	Function
31-16 LP_COUNT	Single buffered low priority region fetch count Number of 64 bit low priority entries of single buffered registers to fetch and load.
15-0 HP_COUNT	Single buffered high priority region fetch count Number of 64 bit high priority entries of single buffer registers to fetch and load.

15.4.3.1.10 AHB address with error response. (AHB_ERR_ADDR)

15.4.3.1.10.1 Offset

Register	Offset
AHB_ERR_ADDR	20h

15.4.3.1.10.2 Diagram



15.4.3.1.10.3 Fields

Field	Function
31-0 AHB_ERR_ADDR	AHB error address. Stores the address of the CTX_LD transaction that responded with an AHB error. A new address is stored only if ahb_error bit in ctrl_status is 0. If AHB_ERR is set, new address is not logged until the bit is cleared. This address is trapped only for CTX_LD initiated transactions when it is fetching the display state from memory. Error response for system transactions are not trapped and does NOT effect this register.

15.5 Graphics Decompression (DEC400D)

15.5.1 Overview

The DEC400D decompresses graphics data. The DEC400D decompresses the data on the fly, and the compressed format in DRAM is transparent to the Display Prefetch Resolve (DPR) block. The DPR is programmed as if the frame buffer is not compressed.

15.5.1.1 Block diagram

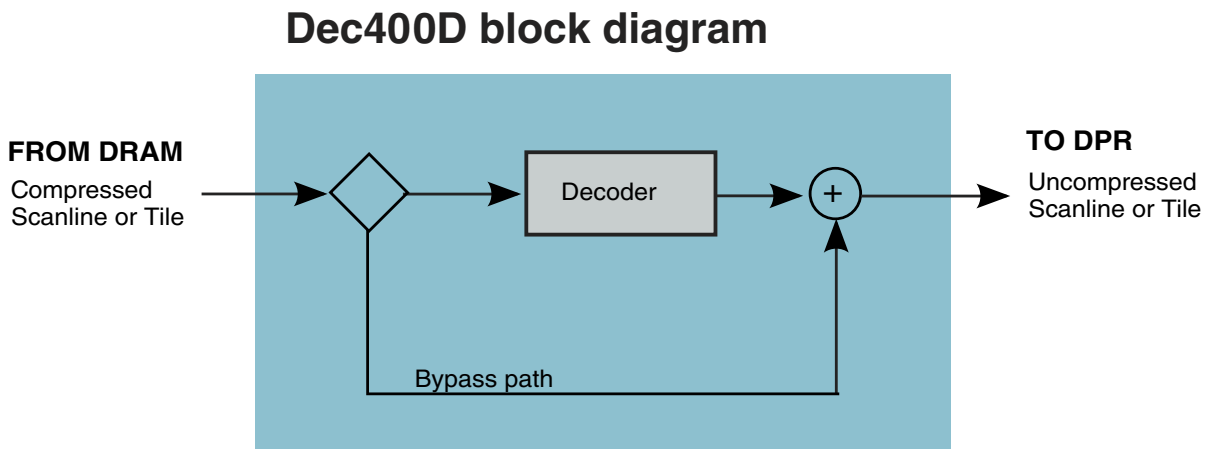


Figure 15-13. DEC400D block diagram

15.5.1.2 Features

The DEC400D includes the following host and memory features:

- 32-bit AHB Interface
- 128-bit width AXI data channel with dedicated pins for Pixel Engine and Memory
- 8-bit AXI ID, up to 16 read transactions supported for decompression
- 16-256 Byte AXI burst data range
- Supports interleaved return data
- Output arbitration, required to insure data output matches request
- Resource locks with CPU (Semaphore lock)

The DEC400D includes the following decode features:

- Up to 8/16 pixels per cycle decode for 32bpp with 128 bit AXI

- 2:1 to 3:1 average decompression ratio
- Pixel data organization support for Linear (scanline) or Tiled (4x4, 4x8, 8x4 and 8x8-XMajor)
- Bypass option controlled via software
- Full frame or partial screen update supported

15.5.2 Functional description

The following sections describe functional details of the DEC400D module.

15.5.2.1 Graphics compression/decompression

The DEC400D decompresses graphics data. The DPR issues AXI requests at a pixel size of 8x4 at 32bpp, or 128B per request, as an uncompressed AXI transaction. The DEC400D decompresses on the fly, and the compressed format in DRAM is transparent to the DPR. The DPR is programmed as if the frame buffer is not compressed.

Graphics buffer formats use the Z order super tile frame buffer format. A diagram depicting the tile format is shown below.

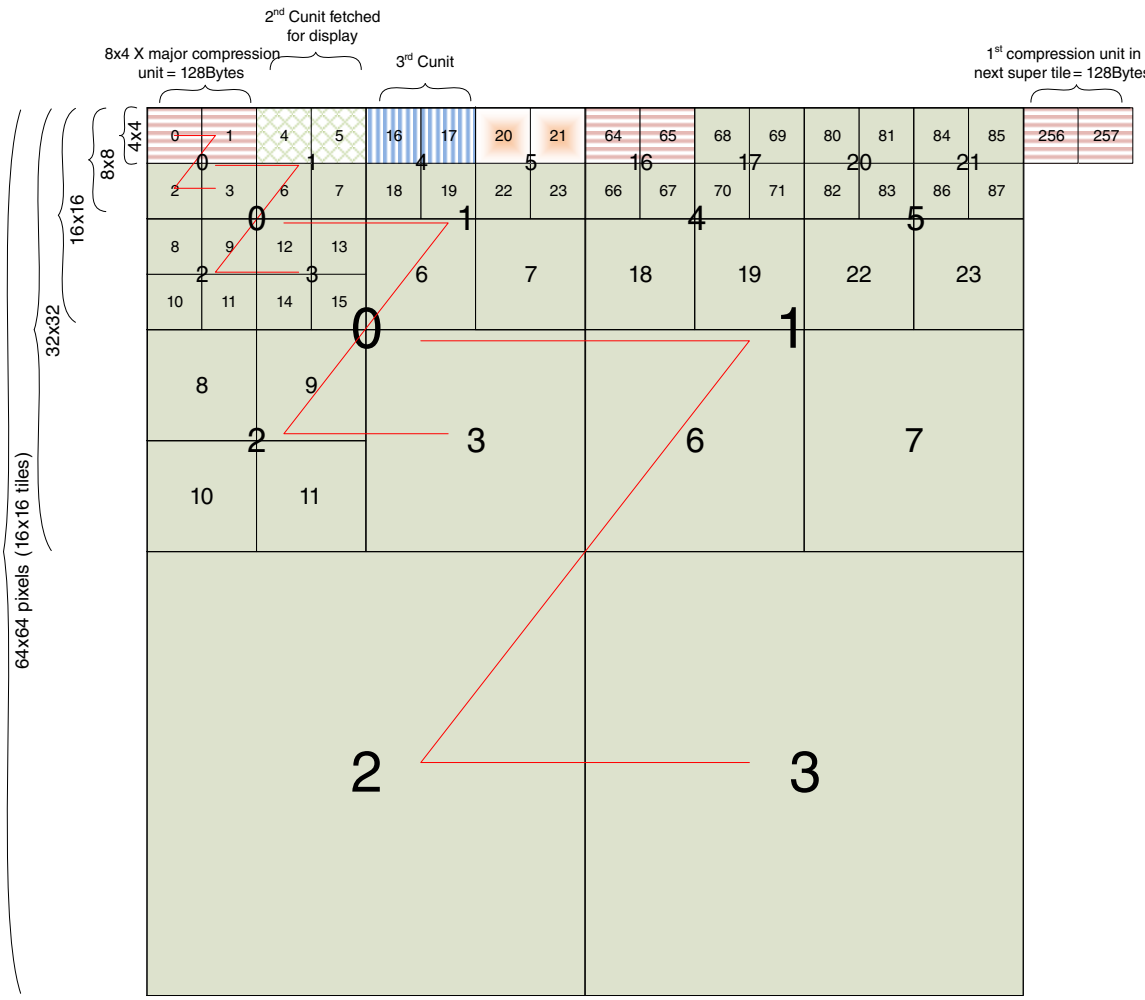


Figure 15-14. Graphics Tile Format

The DPR, or bus master that fetches data from DRAM uncompressed, is programmed to request 128B for each request. Each compression unit is 128B in the X major organization. The first compression unit in the diagram above is two 4x4 tiles labeled as 0,1, the second compression unit the DPR fetches contains the tiles labeled as 4,5. This fetch pattern continues left to right across the supertiled source buffer. The GPU compression scheme must compliment the DC's decompression scheme in that compressed units must be 8x4 pixel groups, which is 128B for two 4x4 tiles. The complete order of compression units the DPR fetches for GPU frames is:

- Bank 1 = cu(0,1), cu(4,5), cu(16,17), cu(20,21), cu(64,65), cu(68,69), cu(80,81), cu(84,85), cu(256,257), … continues to the programmed width of the buffer.
- Bank 2 = cu(2,3), cu(6,7), cu(18,19), …
- Bank 3 = cu(8,9), cu(12,13), cu(24,25), …

The equivalent address fetch pattern for the first row of CUs is:

- 0x0, 0x100, 0x400, 0x500, 0x1000, 0x1100, 0x1400, 0x1500, 0x4000, …

Each bank above is 4 lines high across the display. This data is stored locally in the RTRAM memory for subsequent use by the display controller for screen refresh (see DPR/RTRAM sections).

Note the Z order of 4x4 tiles, which extends to the next size of 8x8, 16x16, etc., to the super tile size that is 64x64 pixels. At a resolution above 64x64, the ordering of super tiles is raster ordered which extends for the entire frame size.

15.5.2.2 CU tile status

Each compression unit (CU) is two consecutive 4x4 tiles. The memory use before/after compression is depicted in the diagram below.

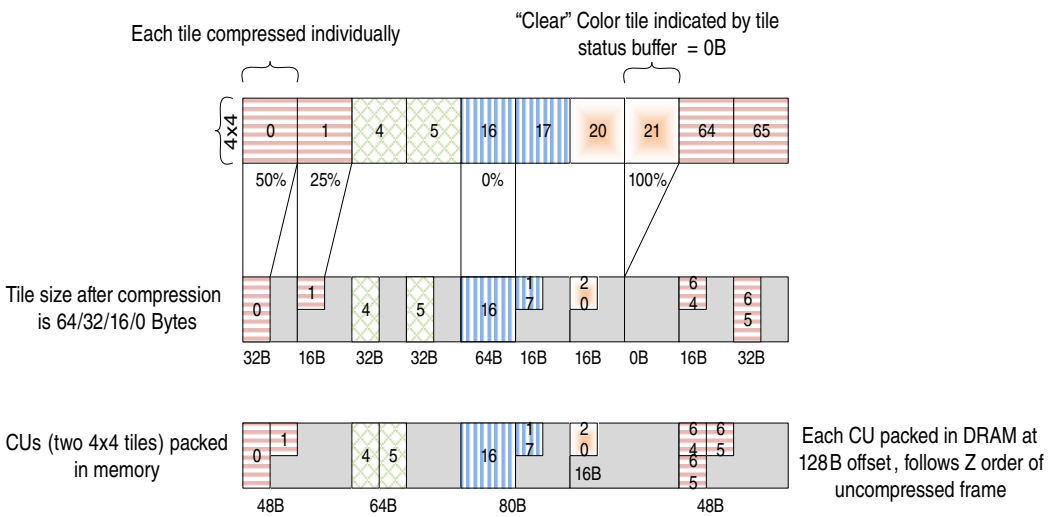


Figure 15-15. Memory use before/after compression

The uncompressed frame buffer is on the first row of tiles above, with each tile arranged as 4x4 pixels. The tiles are compressed to a reduced number of bytes, as in the second row. The compressed tiles are then packed into memory on a per CU basis (note the address order of CUs is still Z which is not depicted in the diagram). DRAM access to compressed tiles is on a per CU basis, and the AXI transaction size is determined by the tile status buffer entry for each CU. The CU data request could be from 16B – 128B on 16B increments. From an LPDDR4 32-bit implementation, 16/32/48/64B accesses are effectively 64B accesses. 80/96/112/128B accesses are effectively 128B accesses from the DRAM perspective.

15.5.3 Memory Map and register definition

This section includes the DEC400D module memory map and detailed descriptions of all registers.

15.5.3.1 DEC400D register descriptions

15.5.3.1.1 DEC400D Memory map

DEC400D_CHAN1 base address: 1_5000h

Offset	Register	Width (In bits)	Access	Reset value
24h	Revision ID (GCCHIPREV)	32	RO	See description.
28h	Release Date (GCCHIPDATE)	32	RO	See description.
98h	Patch Revision (GCREGHICHIPPATCHREV)	32	RO	See description.
A8h	Product ID (GCPRODUCTID)	32	RO	See description.
800h - 87Ch	Decode Read Configuration (GCREGAHBDECREADCONFIG0 - GCREGAHBDECREADCONFIG31)	32	RW	0000_0000h
900h - 97Ch	Decode Read Buffer Base (GCREGAHBDECREADBUFFERBASE0 - GCREGAHBDECREADBUFFERBASE31)	32	RW	0000_0000h
980h - 9FCh	Decode Read Cache Base (GCREGAHBDECREADCACHEBASE0 - GCREGAHBDECREADCACHEBASE31)	32	RW	0000_0000h
B00h	Dec400D Control (GCREGAHBDECCONTROL)	32	RW	0000_0000h
B04h	Interrupt Acknowledge (GCREGAHBDECINTRACKNOWLEDGE)	32	RO	0000_0000h
B08h	Interrupt Enable (GCREGAHBDECINTRENBL)	32	RW	0000_0000h
B0Ch	Tile Status Module Debug (GCREGAHBDECTILESTATUSDEBUG)	32	RO	0000_0000h
B14h	Decompression Module Debug (GCREGAHBDECDECODERDEBUG)	32	RO	0000_0000h
B18h	Total Reads In (GCREGAHBDECTOTALREADSIN)	32	RO	0000_0000h
B20h	Total Read Data Count (GCREGAHBDECTOTALREADBURSTSIN)	32	RO	0000_0000h
B28h	Total Read Request In (GCREGAHBDECTOTALREADREQIN)	32	RO	0000_0000h
B30h	Total Input Read Last Number (GCREGAHBDECTOTALREADLASTSIN)	32	RO	0000_0000h
B38h	Total Reads Out (GCREGAHBDECTOTALREADSOUT)	32	RO	0000_0000h
B40h	Total Read Bursts Out (GCREGAHBDECTOTALREADBURSTSOUT)	32	RO	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B48h	Total Read Request Out (GCREGAHBDECTOTALREADREQOUT)	32	RO	0000_0000h
B50h	Total Read Last Out (GCREGAHBDECTOTALREADLASTSOUT)	32	RO	0000_0000h
B58h	Debug Register 0 (GCREGAHBDECDEBUG0)	32	RO	0000_0000h
B5Ch	Debug Register 1 (GCREGAHBDECDEBUG1)	32	RO	0000_0000h
B60h	Debug register 2 (GCREGAHBDECDEBUG2)	32	RO	0000_0000h
B64h	Debug Register 3 (GCREGAHBDECDEBUG3)	32	RO	0000_0000h
B68h	GCREGAHBDECCONTROLEX (GCREGAHBDECCONTROLEX)	32	RW	0000_0000h
B6Ch	GCREGAHBDECSTATECOMMIT (GCREGAHBDECSTATECOMMIT)	32	RW	0000_0000h
B70h	GCREGAHBDECSTATELOCK (GCREGAHBDECSTATELOCK)	32	RO	0000_0000h
C00h - C7Ch	Decode Read Extra Configuration (GCREGAHBDECREADEXCONFIG0 - GCREGAHBDECREADEXCONFIG31)	32	RW	0000_0000h
C80h - CFCh	Decoder Read Stride (GCREGAHBDECREADSTRIDE0 - GCREGAHBDECREADSTRIDE31)	32	RW	0000_0000h
E00h - E7Ch	Decoder Read Buffer End (GCREGAHBDECREADBUFFEREND0 - GCREGAHBDECREADBUFFEREND31)	32	RW	0000_0000h

15.5.3.1.2 Revision ID (GCCHIPREV)

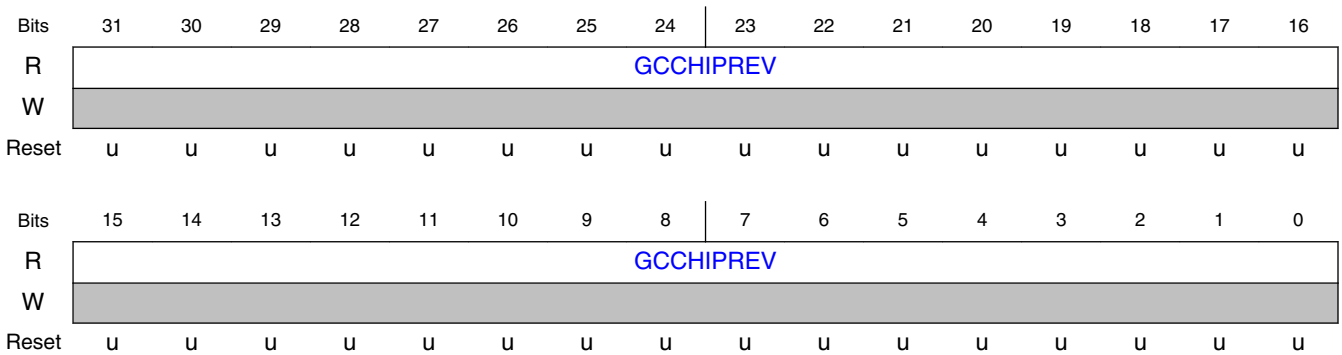
15.5.3.1.2.1 Offset

Register	Offset
GCCHIPREV	24h

15.5.3.1.2.2 Function

This read-only register shows the revision for the chip in BCD. This register has no set reset value. It varies with the implementation.

15.5.3.1.2.3 Diagram



15.5.3.1.2.4 Fields

Field	Function
31-0 GCCHIPREV	Revision ID This read only field returns the revision number.

15.5.3.1.3 Release Date (GCCHIPDATE)

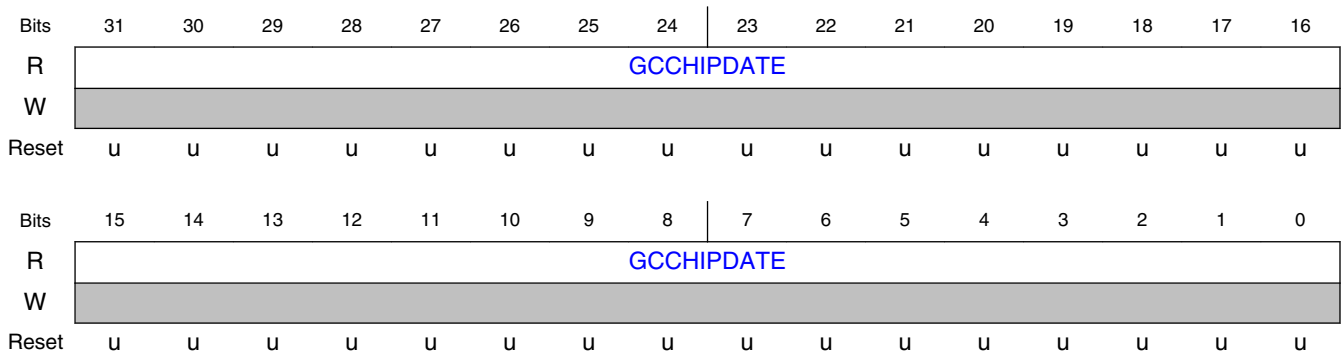
15.5.3.1.3.1 Offset

Register	Offset
GCCHIPDATE	28h

15.5.3.1.3.2 Function

Shows the release date for the IP in YYYYMMDD (year/month/day) format. This register has no set reset value. It varies with the implementation.

15.5.3.1.3.3 Diagram



15.5.3.1.3.4 Fields

Field	Function
31-0 GCCHIPDATE	Date This read only field returns the date of design release.

15.5.3.1.4 Patch Revision (GCREGHICHIPPATCHREV)

15.5.3.1.4.1 Offset

Register	Offset
GCREGHICHIPPATCHREV	98h

15.5.3.1.4.2 Function

Patch revision level for the chip.

15.5.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GCREGHICHIPPATCHREV															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GCREGHICHIPPATCHREV															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

15.5.3.1.4.4 Fields

Field	Function
31-0 GCREGHICHIPPATCHREV	Product ID This read only field returns patch revision level.

15.5.3.1.5 Product ID (GCPRODUCTID)

15.5.3.1.5.1 Offset

Register	Offset
GCPRODUCTID	A8h

15.5.3.1.5.2 Function

This read only register shows the Dec400D Product ID

15.5.3.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GCPRODUCTID															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GCPRODUCTID															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

15.5.3.1.5.4 Fields

Field	Function
31-0 GCPRODUCTID	Product ID This read only field returns the Dec400D Product ID for this implementation.

15.5.3.1.6 Decode Read Configuration (GCREGAHBDECREADCONFIG0 - GCREGAHBDECREADCONFIG31)

15.5.3.1.6.1 Offset

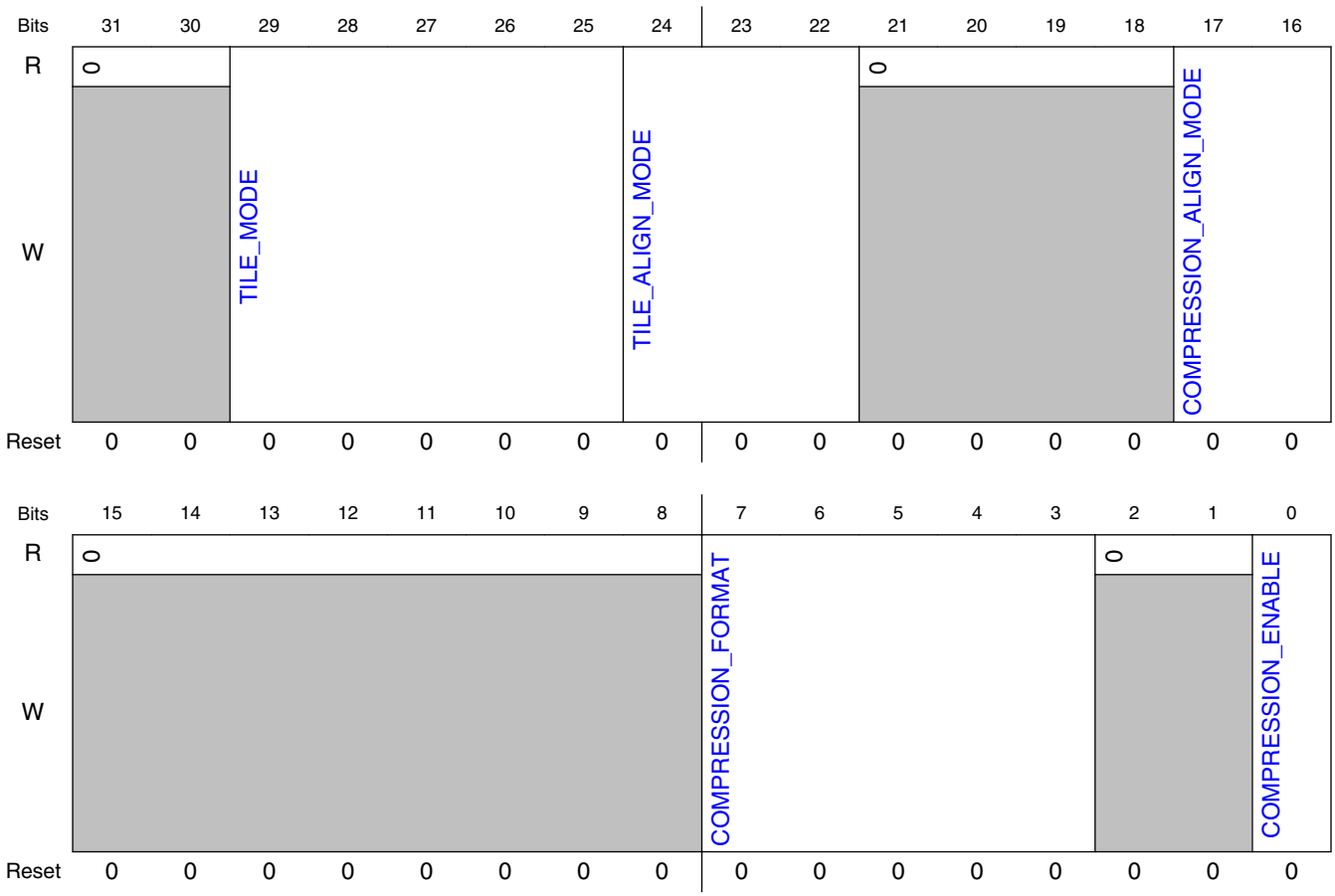
For n = 0 to 31:

Register	Offset
GCREGAHBDECREADCONFIGn	800h + (n × 4h)

15.5.3.1.6.2 Function

Dec400D configuration register for read ID.

15.5.3.1.6.3 Diagram



15.5.3.1.6.4 Fields

Field	Function
31-30 —	Reserved
29-25 TILE_MODE	Tile Mode How many pixels in the tile and the walking direction in the tile. 00000b - TILE8X8_XMAJOR 00001b - TILE8X8_YMAJOR 00010b - TILE16X4 00011b - TILE8X4 00100b - TILE4X8 00101b - TILE4X4 00110b - RASTER16X4 00111b - TILE64X4 01000b - TILE32X4 01001b - RASTER256X1 01010b - RASTER128X1 01011b - RASTER64X4 01100b - RASTER256X2

Table continues on the next page...

Field	Function
	01101b - RASTER128X2 01110b - RASTER128X4 01111b - RASTER64X1
24-22 TILE_ALIGN_MODE	Tile Align Mode The tile alignment mode. 000b - TILE1_ALIGN 001b - TILE2_ALIGN 010b - TILE4_ALIGN 011b - CBSR_ALIGN
21-18 —	Reserved
17-16 COMPRESSION_ALIGN_MODE	Compression Align Mode Compression result size alignment mode. 00b - ALIGN1_BYTE 01b - ALIGN16_BYTE 10b - ALIGN32_BYTE 11b - ALIGN64_BYTE
15-8 —	Reserved
7-3 COMPRESSION_FORMAT	Compression Format Compression color format. 00000b - ARGB8
2-1 —	Reserved
0 COMPRESSION_ENABLE	Compression Enable Compression enable for individual streams. 0b - Disable 1b - Enable

15.5.3.1.7 Decode Read Buffer Base (GCREGAHBDECREADBUFFERBASE0 - GCREGAHBDECREADBUFFERBASE31)

15.5.3.1.7.1 Offset

For n = 0 to 31:

Register	Offset
GCREGAHBDECREADBUFFERBASEn	900h + (n × 4h)

15.5.3.1.7.2 Function

Base address for pixel buffer for read ID.

15.5.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BUFFER_ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BUFFER_ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.7.4 Fields

Field	Function
31-0 BUFFER_ADDRESS	Base address for pixel buffer for read ID.

15.5.3.1.8 Decode Read Cache Base (GCREGAHBDECREADCACHEBASE 0 - GCREGAHBDECREADCACHEBASE31)

15.5.3.1.8.1 Offset

For n = 0 to 31:

Register	Offset
GCREGAHBDECREADCACHEBASEn	980h + (n × 4h)

15.5.3.1.8.2 Function

Base address for tile status buffer for read ID.

15.5.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CACHE_ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CACHE_ADDRESS															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.8.4 Fields

Field	Function
31-0 CACHE_ADDRESS	Base address for tile status buffer for read ID.

15.5.3.1.9 Dec400D Control (GCREGAHBDECCONTROL)

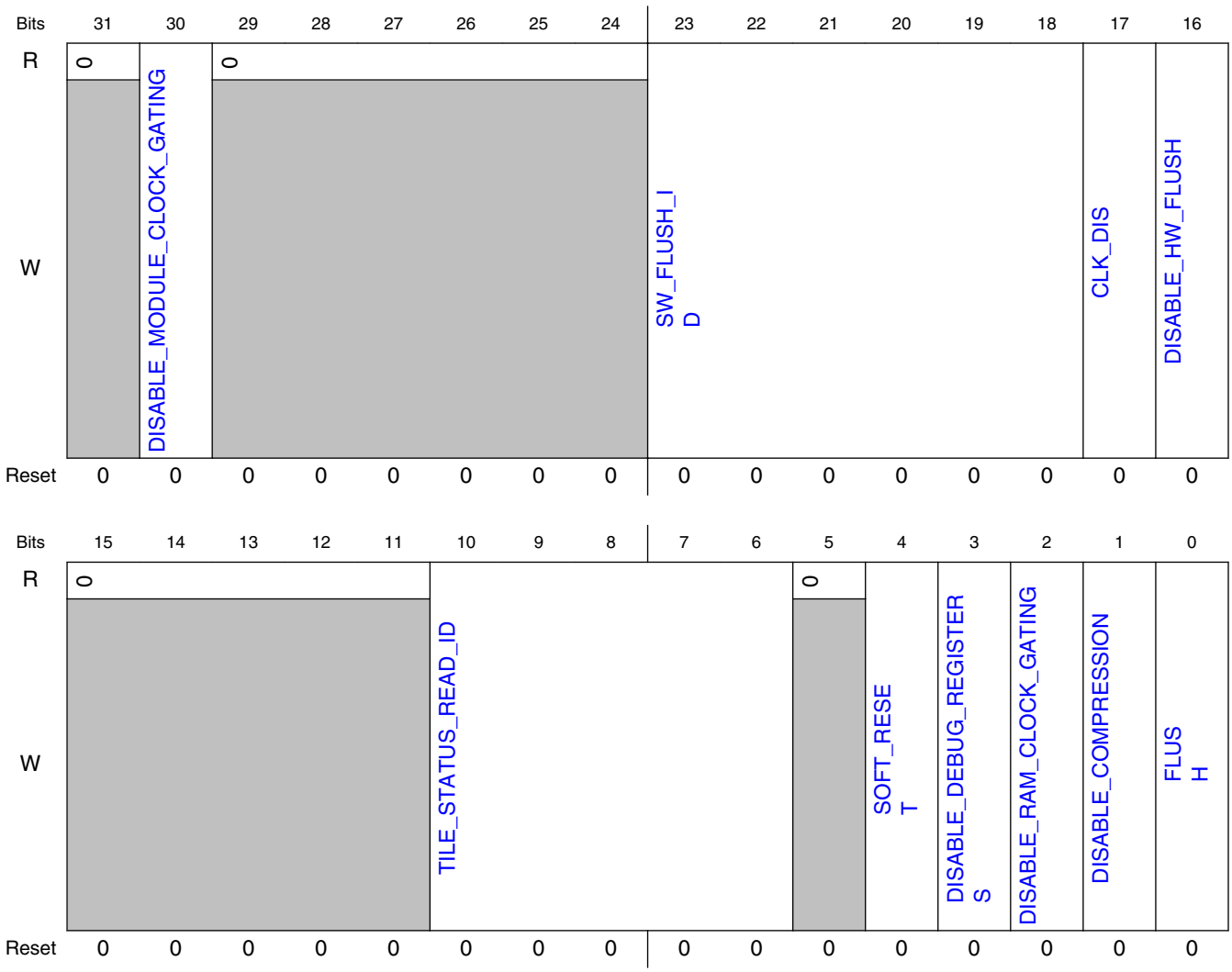
15.5.3.1.9.1 Offset

Register	Offset
GCREGAHBDECCONTROL	B00h

15.5.3.1.9.2 Function

Global Dec400D control register.

15.5.3.1.9.3 Diagram



15.5.3.1.9.4 Fields

Field	Function
31 —	Reserved
30 DISABLE_MODULE_CLOCK_GATING	Disable clock gating for sub modules 0b - Disable 1b - Enable
29-24 —	Reserved
23-18 SW_FLUSH_ID	ID of tile status flush. Bit[23] represents the request type, 0 is read, 1 is write. Bit[22:18] is the stream id

Table continues on the next page...

Field	Function
17 CLK_DIS	Disable clock. 0b - Disable 1b - Enable
16 DISABLE_HW_FLUSH	Tile status cache flush through frame end pin is disabled. 0b - Disable 1b - Enable
15-11 —	Reserved
10-6 TILE_STATUS_READ_ID	Tile status cache's AXI read ID.
5 —	Reserved
4 SOFT_RESET	Soft reset the Dec400D. 0b - Disable 1b - Enable
3 DISABLE_DEBUG_REGISTERS	Disable debug registers. If this bit is 1, debug registers are clock gated. 0b - Disable 1b - Enable
2 DISABLE_RAM_CLOCK_GATING	Disable clock gating for RAMs. 0b - Disable 1b - Enable
1 DISABLE_COMPRESSION	Bypass compression for all streams. 0b - Disable 1b - Enable
0 FLUSH	Flush tile status cache. 0b - Disable 1b - Enable

15.5.3.1.10 Interrupt Acknowledge (GCREGAHBDECINTRACKNOWLEDGE)

15.5.3.1.10.1 Offset

Register	Offset
GCREGAHBDECINTRACKNOWLEDGE	B04h

15.5.3.1.10.2 Function

Each bit represents a corresponding event being triggered. Reading from this register clears the outstanding interrupt

15.5.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	INTR_VEC															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.10.4 Fields

Field	Function
31-0 INTR_VEC	Interrupt vector For each interrupt event, 0=Clear, 1=Interrupt Active Bit 31 is AXI bus error interrupt Bits 30:0 is tile status flush done interrupt for write stream

15.5.3.1.11 Interrupt Enable (GCREGAHBDECINTRENBL)

15.5.3.1.11.1 Offset

Register	Offset
GCREGAHBDECINTRENBL	B08h

15.5.3.1.11.2 Function

Each bit enables a corresponding event. Bit 31 is AXI bus error interrupt Bit 30:0 is tile status flush done interrupt for write stream.

15.5.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	INTR_ENBL_VEC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.11.4 Fields

Field	Function
31-0 INTR_ENBL_VEC	Interrupt enable vector For each bit, 0: Disable interrupt, 1: Enable interrupt. Bit 31 is AXI bus error interrupt. Bit 30:0 is tile status flush done interrupt for write stream.

15.5.3.1.12 Tile Status Module Debug (GCREGAHBDECTILESTATUSDEBUG)

15.5.3.1.12.1 Offset

Register	Offset
GCREGAHBDECTILESTATUSDEBUG	B0Ch

15.5.3.1.12.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	TILE_DEBUG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	TILE_DEBUG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.12.3 Fields

Field	Function
31-0 TILE_DEBUG	Debug

15.5.3.1.13 Decompression Module Debug (GCREGAHBDECDECODERDEBUG)

15.5.3.1.13.1 Offset

Register	Offset
GCREGAHBDECDECODERDEBUG	B14h

15.5.3.1.13.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEC_DEBUG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEC_DEBUG															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.13.3 Fields

Field	Function
31-0 DEC_DEBUG	Debug

15.5.3.1.14 Total Reads In (GCREGAHBDECTOTALREADSIN)

15.5.3.1.14.1 Offset

Register	Offset
GCREGAHBDECTOTALREADSIN	B18h

15.5.3.1.14.2 Function

Total reads in terms of dataWidth in input (pixel engine) side. Sum of ARLEN+1 of all requests sent to DEC.

15.5.3.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDIN_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDIN_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.14.4 Fields

Field	Function
31-0 RDIN_COUNT	Count

15.5.3.1.15 Total Read Data Count (GCREGAHBDECTOTALREADBURSTSIN)

15.5.3.1.15.1 Offset

Register	Offset
GCREGAHBDECTOTALREADBURSTSIN	B20h

15.5.3.1.15.2 Function

Total read data count in terms of dataWidth in input (pixel engine) side. Sum of returned data beats count sent back to pixel engine. Equal to gcregAHBDECTotalReadsIn.

15.5.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDIN_BURST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDIN_BURST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.15.4 Fields

Field	Function
31-0 RDIN_BURST_COUNT	Count

15.5.3.1.16 Total Read Request In (GCREGAHBDECTOTALREADREQIN)

15.5.3.1.16.1 Offset

Register	Offset
GCREGAHBDECTOTALREADREQIN	B28h

15.5.3.1.16.2 Function

Total reads request number in input (pixel engine) side.

15.5.3.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDIN_REQ_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDIN_REQ_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.16.4 Fields

Field	Function
31-0 RDIN_REQ_COUNT	Count

15.5.3.1.17 Total Input Read Last Number (GCREGAHBDECTOTALREADLASTSIN)

15.5.3.1.17.1 Offset

Register	Offset
GCREGAHBDECTOTALREADLASTSIN	B30h

15.5.3.1.17.2 Function

Total input read last number in input (pixel engine) side. Equal to gcregAHBDECTotalReadsReqIn.

15.5.3.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDIN_LAST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDIN_LAST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.17.4 Fields

Field	Function
31-0 RDIN_LAST_COUNT	Count

15.5.3.1.18 Total Reads Out (GCREGAHBDECTOTALREADSOUT)

15.5.3.1.18.1 Offset

Register	Offset
GCREGAHBDECTOTALREADSOUT	B38h

15.5.3.1.18.2 Function

Total reads request in terms of dataWidth in output (DDR) side. Sum of ARLEN+1 of all the requests the Dec400D sends out.

15.5.3.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDOUT_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDOUT_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.18.4 Fields

Field	Function
31-0 RDOUT_COUNT	Count

15.5.3.1.19 Total Read Bursts Out (GCREGAHBDECTOTALREADBURSTSOUT)

15.5.3.1.19.1 Offset

Register	Offset
GCREGAHBDECTOTALREADBURSTSOUT	B40h

15.5.3.1.19.2 Function

Total read data count in terms of dataWidth in output (DDR) side. Should be equal to gcregAHBDECTotalReadsOUT.

15.5.3.1.19.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDOUT_BURST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDOUT_BURST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.19.4 Fields

Field	Function
31-0 RDOUT_BURST_COUNT	Count

15.5.3.1.20 Total Read Request Out (GCREGAHBDECTOTALREADREQOUT)

15.5.3.1.20.1 Offset

Register	Offset
GCREGAHBDECTOTALREADREQOUT	B48h

15.5.3.1.20.2 Function

Total reads request number in output (DDR) side..

15.5.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDOUT_REQ_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDOUT_REQ_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.20.4 Fields

Field	Function
31-0 RDOUT_REQ_COUNT	Count

15.5.3.1.21 Total Read Last Out (GCREGAHBDECTOTALREADLASTSOUT)

15.5.3.1.21.1 Offset

Register	Offset
GCREGAHBDECTOTALREADLASTSOUT	B50h

15.5.3.1.21.2 Function

Total read last number in output (DDR) side. Equal to gcregAHBDECTotalReadsReqOUT.

15.5.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	RDOUT_LAST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	RDOUT_LAST_COUNT															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.21.4 Fields

Field	Function
31-0 RDOUT_LAST_COUNT	Count

15.5.3.1.22 Debug Register 0 (GCREGAHBDECDEBUG0)

15.5.3.1.22.1 Offset

Register	Offset
GCREGAHBDECDEBUG0	B58h

15.5.3.1.22.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEBUG0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEBUG0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.22.3 Fields

Field	Function
31-0 DEBUG0	Debug register 0

15.5.3.1.23 Debug Register 1 (GCREGAHBDECDEBUG1)

15.5.3.1.23.1 Offset

Register	Offset
GCREGAHBDECDEBUG1	B5Ch

15.5.3.1.23.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEBUG1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEBUG1															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.23.3 Fields

Field	Function
31-0 DEBUG1	Debug 1

15.5.3.1.24 Debug register 2 (GCREGAHBDECDEBUG2)

15.5.3.1.24.1 Offset

Register	Offset
GCREGAHBDECDEBUG2	B60h

15.5.3.1.24.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEBUG2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEBUG2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.24.3 Fields

Field	Function
31-0 DEBUG2	Debug 2

15.5.3.1.25 Debug Register 3 (GCREGAHBDECDEBUG3)

15.5.3.1.25.1 Offset

Register	Offset
GCREGAHBDECDEBUG3	B64h

15.5.3.1.25.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	DEBUG3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DEBUG3															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.25.3 Fields

Field	Function
31-0 DEBUG3	Debug 3

15.5.3.1.26 GCREGAHBDECCONTROLEX (GCREGAHBDECCONTROLEX)

15.5.3.1.26.1 Offset

Register	Offset
GCREGAHBDECCONTROLEX	B68h

15.5.3.1.26.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GCREGAHBDECCONTROLEX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GCREGAHBDECCONTROLEX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.26.3 Fields

Field	Function
31-0 GCREGAHBDECCONTROLEX	GCREGAHBDECCONTROLEX

15.5.3.1.27 GCREGAHBDECSTATECOMMIT (GCREGAHBDECSTATECOMMIT)

15.5.3.1.27.1 Offset

Register	Offset
GCREGAHBDECSTATECOMMIT	B6Ch

15.5.3.1.27.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	GCREGAHBDECSTATECOMMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	GCREGAHBDECSTATECOMMIT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.27.3 Fields

Field	Function
31-0 GCREGAHBDECSTATECOMMIT	GCREGAHBDECSTATECOMMIT

15.5.3.1.28 GCREGAHBDECSTATELOCK (GCREGAHBDECSTATELOCK)

15.5.3.1.28.1 Offset

Register	Offset
GCREGAHBDECSTATELOCK	B70h

15.5.3.1.28.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	GCREGAHBDECSTATELOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	GCREGAHBDECSTATELOCK															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.28.3 Fields

Field	Function
31-0 GCREGAHBDECSTATELOCK	GCREGAHBDECSTATELOCK

15.5.3.1.29 Decode Read Extra Configuration (GCREGAHBDECREADEXCONFIG0 - GCREGAHBDECREADEXCONFIG31)

15.5.3.1.29.1 Offset

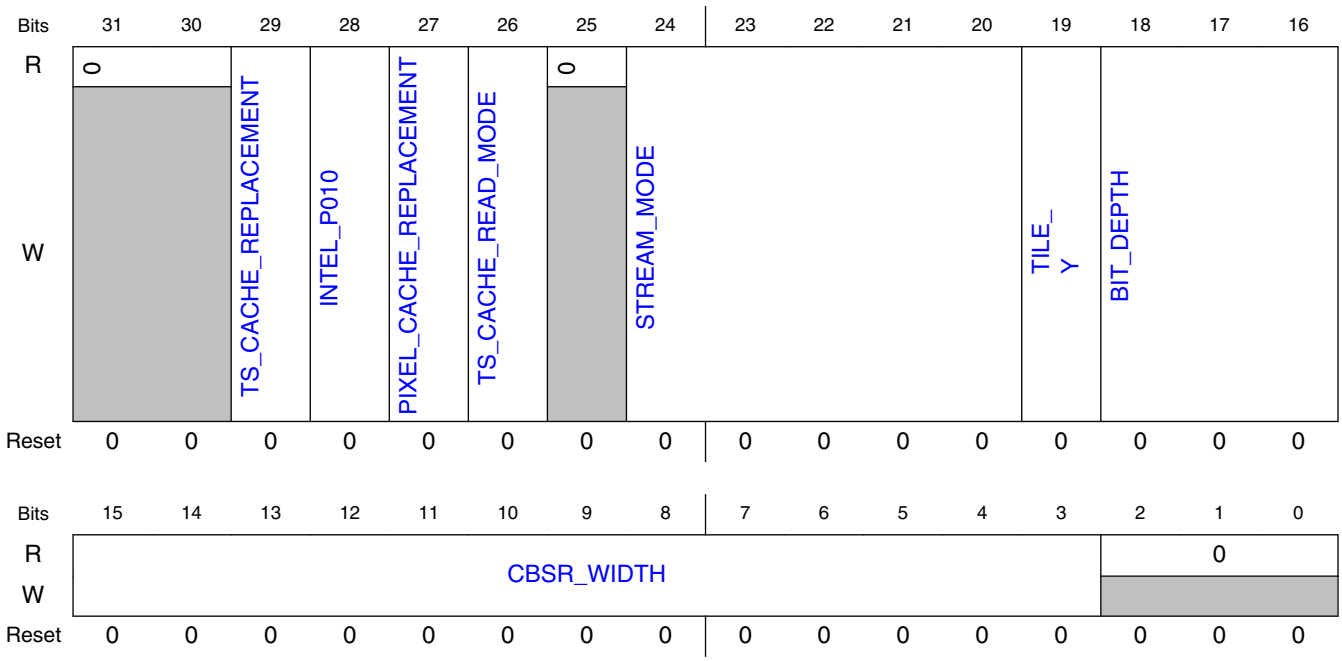
For n = 0 to 31:

Register	Offset
GCREGAHBDECREADEXCONFIGn	C00h + (n × 4h)

15.5.3.1.29.2 Function

Dec400D extra configuration register for read ID.

15.5.3.1.29.3 Diagram



15.5.3.1.29.4 Fields

Field	Function
31-30 —	Reserved
29 TS_CACHE_REPLACEMENT	TS cache replacement Replacement method support for tile status cache. 0b - LRU 1b - FIFO
28 INTEL_P010	Intel's P010 format Intel's P010 format for 10 bit. 0b - Disable 1b - Enable
27 PIXEL_CACHE_REPLACEMENT	Pixel cache replacement Replacement method support for pixel overfetch cache. 0b - LRU 1b - FIFO
26 TS_CACHE_READ_MODE	TS cache read mode By default, the TS cache for write client is write only cache. If this bit is set, the cache reads back the cache line first, then modify it not valid for the read ID. 0b - Disable 1b - Enable
25	Reserved

Table continues on the next page...

Field	Function
—	
24-20 STREAM_MODE	Stream mode • Mode 1-9 only valid for DEC_IPU • Mode 10-13 only valid for DEC_VPU • Mode 14-19 only valid for DEC_XYZ Stream mode enumerates the possible streams sent by clients. In addition to the basic format configured for compression, streams have additional characteristics, such as reading or writing walking pattern, which distinguish one type from another 00000b - Default 00001b - ISA_STREAM0 00010b - ISA_STREAM1 00011b - ISA_STREAM2 00100b - ISA_STREAM3 00101b - TNR_STREAM_Y 00110b - TNR_STREAM_UV 00111b - GDC_STREAM_Y 01000b - GDC_STREAM_U 01001b - GDC_STREAM_V 01010b - VPU_SRC_Y 01011b - VPR_SRC_UV 01100b - VPU_REF_Y 01101b - VPU_REF_UV 01110b - XYZ_STREAM_AY 01111b - XYZ_STREAM_AU 10000b - XYZ_STREAM_AV 10001b - XYZ_STREAM_BY 10010b - XYZ_STREAM_BU 10011b - XYZ_STREAM_BV
19 TILE_Y	Tile Y Memory layout TileY 0b - Disable 1b - Enable
18-16 BIT_DEPTH	Bit depth Bit depth for Y/UV/Bayer format. 000b - 8 bit 001b - 10 bit 010b - 12 bit 011b - 16 bit
15-3 CBSR_WIDTH	CBSR width For CBSR Align mode, this bitfield indicates Y CBS number included in one CBSR since CBSG_SADDR field in CBSG table entry is based from each CBSR start address. Also used to judge whether CbCr needs decompression if Y is in odd CBS number.
2-0 —	Reserved

15.5.3.1.30 Decoder Read Stride (GCREGAHBDECREADSTRIDE0 - GCREGAHBDECREADSTRIDE31)

15.5.3.1.30.1 Offset

For n = 0 to 31:

Register	Offset
GCREGAHBDECREADSTRIDEn	C80h + (n × 4h)

15.5.3.1.30.2 Function

Dec400D surface stride in bytes for read ID.

15.5.3.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0														STRIDE	
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STRIDE															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.30.4 Fields

Field	Function
31-18 —	Reserved
17-0 STRIDE	Surface stride

15.5.3.1.31 Decoder Read Buffer End (GCREGAHBDECREADBUFFEREND0 - GCREGAHBDECREADBUFFEREND31)

15.5.3.1.31.1 Offset

For n = 0 to 31:

Register	Offset
GCREGAHBDECREADBUFFERENDn	E00h + (n × 4h)

15.5.3.1.31.2 Function

End address for pixel buffer for read ID. Pixel and tile status cache invalidation is triggered automatically when address of a request matches this address.

15.5.3.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	RD_BUFF_END															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	RD_BUFF_END															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.5.3.1.31.4 Fields

Field	Function
31-0 RD_BUFF_END	Address

15.6 Decompression and Tile to Raster Conversion (DTRC)

15.6.1 Overview

The Decompression and Tile to Raster Conversion (DTRC) module decompresses content for video frames. It decompresses and resolves the tile data into linear scan lines.

The DTRC buffers a full 4K of raster lines in its local memory. This is required since the DPR fetches scan lines from the resolve memory attached to the DTRC.

15.6.2 Features

The DTRC includes the following functionality features:

- Input data format can be one of the following:
  - Compressed Hantro G2 reference frame buffer format
  - Uncompressed tile Hantro G2 reference frame buffer format
  - Uncompressed tile Hantro G1 reference frame buffer format
- Output data format is uncompressed YCbCr pixels on AXI bus
- Decompression for G2:
  - DTRC decompresses data in units of Coding Block Set Groups (CBSG)
  - DTRC decompression can be bypassed
- The DTRC converts G2/G1 tile format pixels into raster scan order pixels
- Picture size:
  - Minimum width of 144
  - Minimum height of 144
  - Maximum width of 4096
  - Maximum width of 4096
  - The width and height must be integer multiples of 8
- Maximum pixel sample rate is 4096 x 2304 at 60fps
- Output pixel scan order is normal whole picture raster scan
- Output picture crop:
  - Cropped picture origin (x,y) points to the top left corner of the cropped picture
  - Both x and y are even, and the unit is pixel
  - Cropped width is even and unit is pixel
  - Cropped height is even and unit is pixel
  - Minimum cropped picture size is 144 x 2
- Non-G2/G1 data read transactions directly pass through the DTRC
  - The transactions are distinguished by ARID or address
- Run-time hot reset triggered by software
  - G2/G1 data processing path is reset
  - Non-G2/G1 data path continues working
- Data prefetch and pre-decompression supported:
  - Prefetch between frames
  - Prefetch along with pixel scan order

15.6.3 Block diagram

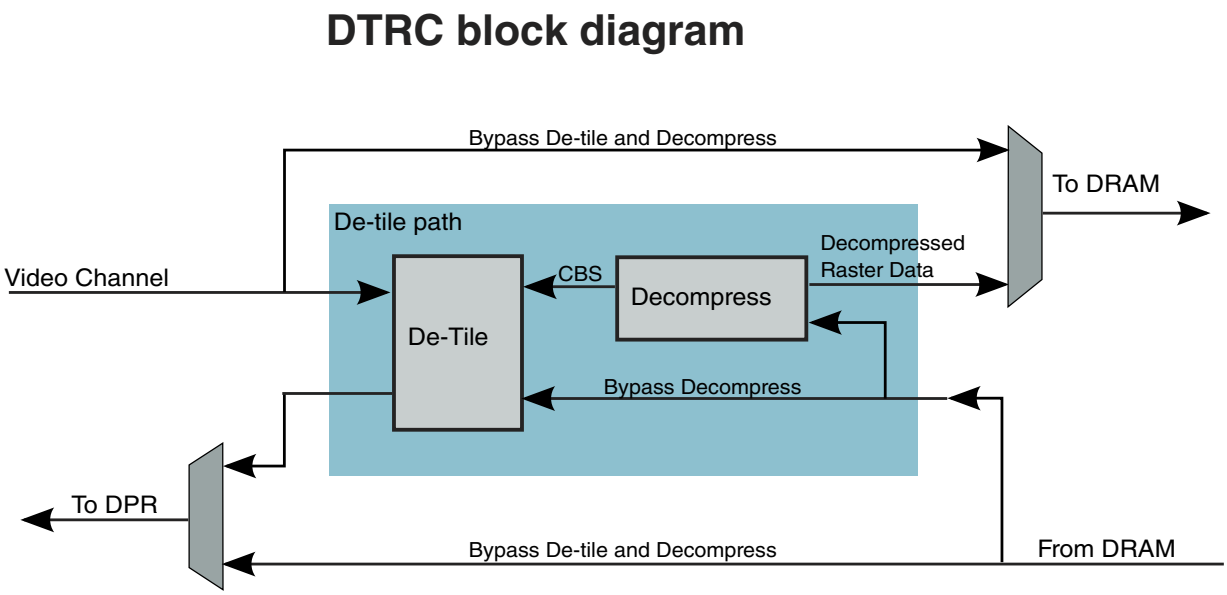


Figure 15-16. DTRC block diagram

15.6.4 Functional description

The following sections describe functional details of the DTRC module.

15.6.4.1 Video compression/decompression

Video decompression uses the DTRC module. This module decompresses CBSGs (Coding Block Set Groups) and store the resolved data in raster order in a local buffer exclusive to each DTRC instance. The DPR is programmed to fetch raster data, and the DTRC returns the scan line data to the DPRs.

The compression scheme used by the video engine is not Z ordered like in GPU frames. Instead, video frames are compressed using 8x8 pixel tiles (for Luma) as each compressed unit, and the compressed results are packed in memory. The 8x8 tiles are compressed in raster scan order from left to right, then top to bottom. The tile order of video frames is depicted in the following diagram.

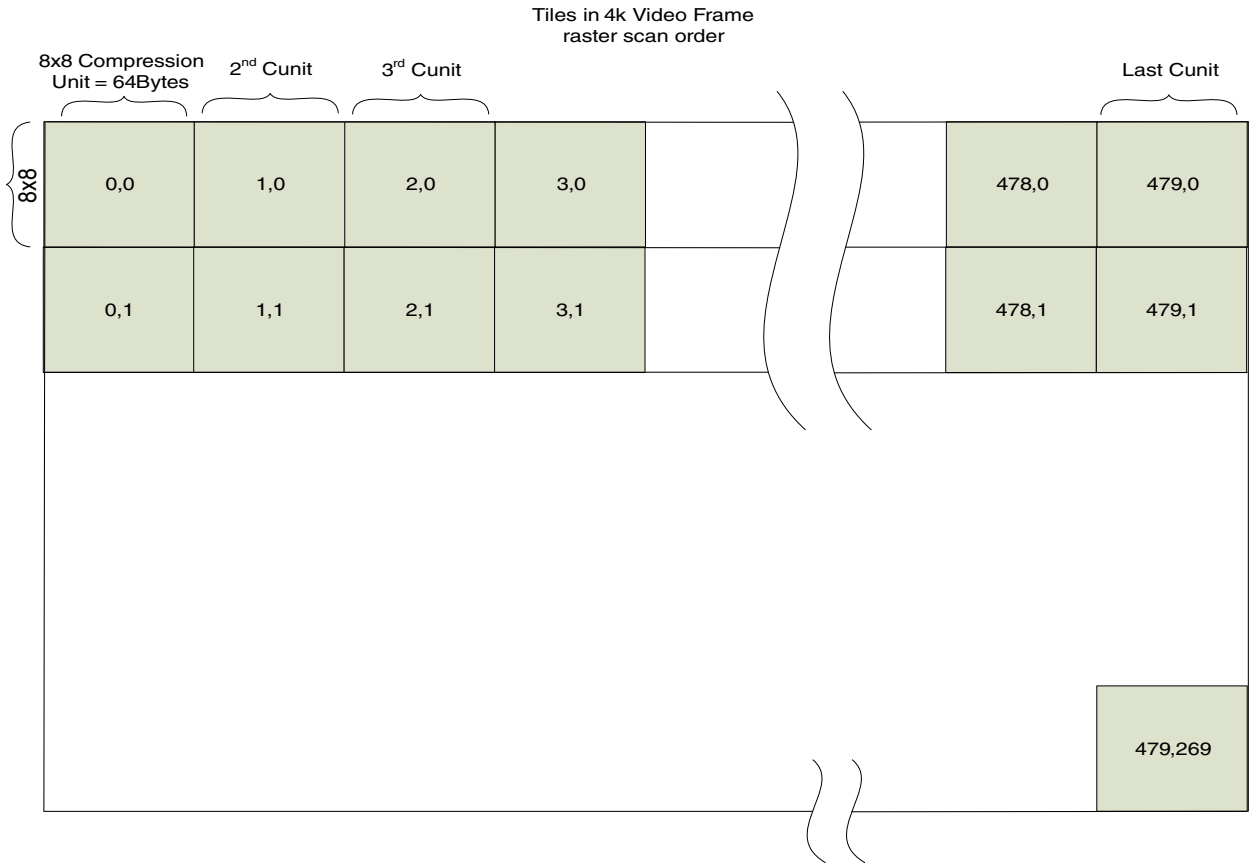


Figure 15-17. Video Frame Tile Order

NOTE

CbCr frame buffers are 8x4 compressed units, which are also 64B per compressed unit.

The video compression scheme is depicted in the diagram below.

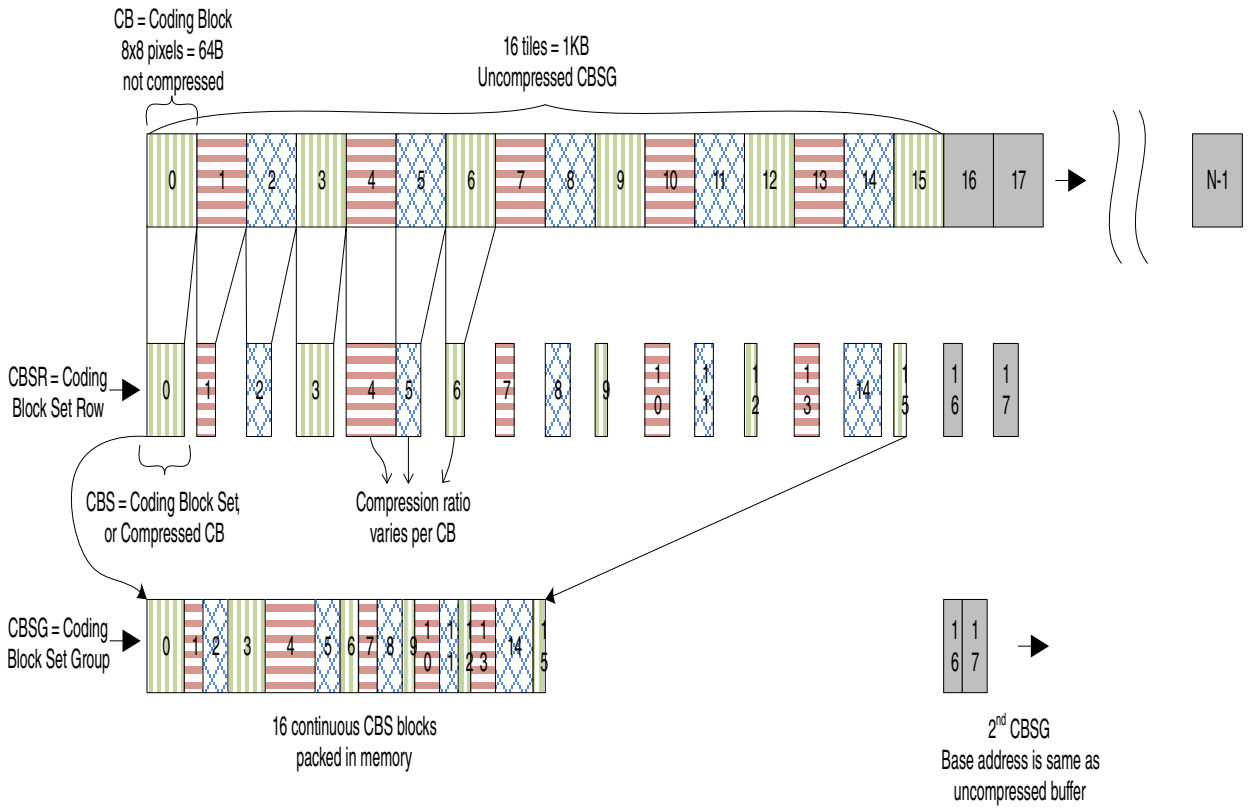


Figure 15-18. Video Compression Scheme

Each 8x8 tile (for Luma) is compressed separately as a single compressed unit, called a CBS unit. The compression ratio can be different for each unit. Then, 16 CBSs are packed in memory to form a CBSG. These 16 uncompressed tiles are 1KB, and when compressed and packed requires less than 1KB. In the example above, the compressed result for the 1st CBSG is about 50%, so the CBSG requires 512B in memory.

NOTE

The DTRC fetches compressed CBSGs, uncompress the units, and store them as raster scan lines in local memory attached to the DTRC. The DTRC then returns the data to the DPR in raster scan order. The DPR is programmed in raster scan mode, with a burst length of 4KB. This forces the DPR to fetch an entire scan line, and then move to the next scan line in the vertical direction. In this case, the DPR does not resolve tiles, but just process raster ordered scan lines.

15.6.4.2 DTRC scan line memory size

The maximum display is 3840 pixels. The scan line structure is 8 lines high for luma, and 4 lines high for chroma. Each pixel is 10-bit per component. The memory size is:

- Luma buffer = 3840 * 8 * 10/8 = 38400B
- Chroma buffer = 3840 * 4 * 10/8 = 19200B

This implementation supports 40960B for the luma buffer and 20480B for the chroma buffer to support 4096 horizontal displays. The total memory size is 61440B.

15.6.5 Programming considerations

The following sections describe software usage for the DTRC module.

15.6.5.1 Initial configuration

At the beginning to start the decompression of a video sequence, software needs to configure the following:

- Set DTRCINTEN register to enable/disable interrupt sources
- Set DTID2DDR register to configure AXI ID of DTRC generated transaction to fetch G2/G1 video data. If the default, AXI ID won't be used.
- Configure the mechanism to determine between G2/G1 and non-G2/G1 data
  - If ARID used to determine:
    - Set DTCTRL[30] to 0
    - Set DTCTRL[1:0] to select a mode of how to use IDs in ARIDR register
    - According to DTCTRL[1:0], set all of the four IDs in ARIDR. Duplicate if ID to be set is less than 4
  - If address used to determine:
    - Set DTCTRL[30] to 1
    - If all transactions are non-G2/G1 transactions, so that no meaningful address segment corresponding to G2/G1 video data, set all bit 0 of F0SYSSA, F0SYSEA, F0SUVSSA, F0SUVSEA, F1SYSSA, F1SYSEA, F1SUVSSA, F1SUVSEA to 1.

15.6.5.2 Picture-level configuration

There are two sets of picture level configuration registers corresponding to two back-to-back pictures, which allow seamless display between two pictures.

Software first looks at DTCTRL[31] to know which set of registers to program. After that, it sets addresses, picture size, vertical scan direction etc. into corresponding registers.

If ARADDR_S is used to tell G2/G1 and non-G2/G1 transactions, set luma and chroma video data address segment into F0SYSSA, F0SYSEA, F0SUVSSA, F0SUVSEA, or F1SYSSA, F1SYSEA, F1SUVSSA, F1SUVSEA.

After the whole set of configuration is done, set the F0DCTL[0] or F1DCTL[0] to 1.

In scenario that there is no G2/G1 transactions, set F0DCTL[0] or F1DCTL[0] to 0 and do not need to do any picture level configuration.

The DTRC internal state machine periodically performs the following action sequence:

1. (After an asynchronous reset) wait frame 0 bit 0 == 1’b1
2. Decompress frame
3. Wait frame 1 bit 0 == 1’b1
4. Decompress frame
5. Wait frame 0 bit 0 == 1’b1
6. Decompress frame
7. Wait frame 1 bit 0 == 1’b1
8. Decompress frame
9. ...

Although the switch of the two sets of picture configuration is predictable, it is recommended software always check the DTCTRL[31] to program picture settings.

15.6.5.3 Interrupts

If interrupt is enabled, DTRC raises the interrupt signal to high and sets the interrupt cause bit in FDINTR. Software should clean the interrupt after it finished the handling (such as program the next picture configurations).

15.6.5.4 Time out

To enable time out interrupt, software not only enables the interrupt but also enables the performance counter. Software calculates a reasonable maximum cycle count according to DTRC frequency and picture frame rate to set TOCR.

15.6.5.5 Hot reset

The video path of decompression and detile can be hot reset and do not influence the bypass non-G2/G1 transactions. To do the reset, the user needs to guarantee the following requirements:

NOTE

Requirement: The hot reset must be triggered when there is no outstanding G2/G1 video transactions remained at the DTRC AXI slave port.

The hot reset details are described below:

1. Software can write DTCTRL[2] by 1 to start a hot reset
2. A hot reset can start at any time in case the requirement described above is met
3. The reset action is hold a while after DTRC received the reset
4. When G2/G1 data related BUS transactions are clean, DTRC performs the reset
    a. The following external visible register bits are reset, and others are not reset
        1. F0DCTL[0]
        2. F1DCTL[1]
        3. DTCTRL[31]
        4. PFCR[31:0]
    b. Decompression and detile status is reset so that DTRC restarts from a new frame configured by the frame setting register group 0
    c. If DTRCINTEN[4] is 1, DTRC raises hot reset finish interrupt as well as set the interrupt bit DTRCINTR[4]
    d. DTRC clears DTCTRL[2] to 0
5. During the hot reset procedure, only bus error interrupt can be asserted, except for hot reset finish interrupt
6. During the hot reset procedure, the whole non-G2/G1 data path keeps working

15.6.6 Memory Map and register definition

This section includes the DTRC module memory map and detailed descriptions of all registers.

15.6.6.1 DTRC register descriptions

15.6.6.1.1 DTRC Memory map

DTRC_CHAN2 base address: 1_6000h

DTRC_CHAN3 base address: 1_7000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Luma video data start address (F0DYDSADDR)	32	RW	0000_0000h
4h	Chroma video data start address (F0DCDSADDR)	32	RW	0000_0000h
8h	Luma table data start address (F0DYTSADDR)	32	RW	0000_0000h
Ch	Chroma table data start address (F0DCTSADDR)	32	RW	0000_0000h
10h	Frame size (F0SIZE)	32	RW	0000_0000h
14h	Luma data slave start address (F0SYSSA)	32	RW	0000_0000h
18h	Luma data slave end address (F0SYSEA)	32	RW	0000_0000h
1Ch	Chroma data slave start address (F0SUVSSA)	32	RW	0000_0000h
20h	Chroma data slave end address (F0SUVSEA)	32	RW	0000_0000h
24h	Cropped picture origin (F0CROPORIG)	32	RW	0000_0000h
28h	Cropped picture size (F0CROPSIZE)	32	RW	0000_0000h
2Ch	Frame data control (F0DCTL)	32	RW	0000_0000h
60h	Luma video data start address (F1DYDSADDR)	32	RW	0000_0000h
64h	Chroma video data start address (F1DCDSADDR)	32	RW	0000_0000h
68h	Luma table data start address (F1DYTSADDR)	32	RW	0000_0000h
6Ch	Chroma table data start address (F1DCTSADDR)	32	RW	0000_0000h
70h	Frame size (F1SIZE)	32	RW	0000_0000h
74h	Luma data slave start address (F1SYSSA)	32	RW	0000_0000h
78h	Luma data slave end address (F1SYSEA)	32	RW	0000_0000h
7Ch	Chroma data slave start address (F1SUVSSA)	32	RW	0000_0000h
80h	Chroma data slave end address (F1SUVSEA)	32	RW	0000_0000h
84h	Cropped picture origin (F1CROPORIG)	32	RW	0000_0000h
88h	Cropped picture size (F1CROPSIZE)	32	RW	0000_0000h
8Ch	Frame data control (F1DCTL)	32	RW	0000_0000h
C0h	DTRC Interrupt enables (DTRCINTEN)	32	RW	0000_0000h
C4h	DTRC Interrupt Requests (FDINTR)	32	RW	0000_0000h
C8h	DTRC Control (DTCTRL)	32	RW	0000_0000h
CCh	ARIDR (ARIDR)	32	RW	0000_0000h
D0h	DTID2DDR (DTID2DDR)	32	RW	0000_0F0Eh
D4h	DTRCCONFIG (DTRCCONFIG)	32	RO	See description.
D8h	DTRC Version (DTRCVERSION)	32	RO	See description.
F0h	Performance counter control (PFCTRL)	32	RW	0000_0000h
F4h	Performance counter (PFCR)	32	RW	0000_0000h
F8h	Time Out Cycles (TOCR)	32	RW	0000_0000h

15.6.6.1.2 Luma video data start address (F0DYDSADDR)

15.6.6.1.2.1 Offset

Register	Offset
F0DYDSADDR	0h

15.6.6.1.2.2 Function

This register specifies the start address for luma video data saved in external memory. This data is compressed or decompressed data.

15.6.6.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F0DYDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F0DYDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.2.4 Fields

Field	Function
31-0 F0DYDSADDR	Luma video data start address Start address for luma data saved in external memory. Unit is byte, but should be 16-byte aligned.

15.6.6.1.3 Chroma video data start address (F0DCDSADDR)

15.6.6.1.3.1 Offset

Register	Offset
F0DCDSADDR	4h

15.6.6.1.3.2 Function

This register specifies the start address for chroma video data saved in external memory.

15.6.6.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F0DCDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F0DCDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.3.4 Fields

Field	Function
31-0 F0DCDSADDR	Chroma video data start address Start address for chroma decompression data saved in external memory. Unit is byte, but should be 16-byte aligned.

15.6.6.1.4 Luma table data start address (F0DYTSADDR)

15.6.6.1.4.1 Offset

Register	Offset
F0DYTSADDR	8h

15.6.6.1.4.2 Function

This register specifies the start address for luma table data saved in external memory. Not used when compression is disabled in G2.

15.6.6.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F0DYTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F0DYTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.4.4 Fields

Field	Function
31-0 F0DYTSADDR	Luma table data start address Start address for luma table data saved in external memory. Unit is byte and 16-byte aligned.

15.6.6.1.5 Chroma table data start address (F0DCTSADDR)

15.6.6.1.5.1 Offset

Register	Offset
F0DCTSADDR	Ch

15.6.6.1.5.2 Function

This register specifies the start address for chroma table data saved in external memory. Not used when compression is disabled in G2.

15.6.6.1.5.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F0DCTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F0DCTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.5.4 Fields

Field	Function
31-0 F0DCTSADDR	Chroma table data start address Start address for chroma table data saved in external memory. Unit is byte and 16-byte aligned.

15.6.6.1.6 Frame size (F0SIZE)

15.6.6.1.6.1 Offset

Register	Offset
F0SIZE	10h

15.6.6.1.6.2 Function

This register specifies frame width and frame height in units of 8.

15.6.6.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved						F0HEIGHT									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved						F0WIDTH									
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.6.4 Fields

Field	Function
31-26 —	Reserved.
25-16 F0HEIGHT	Frame height Indicates frame height in units of 8 lines.
15-10 —	Reserved.
9-0 F0WIDTH	Frame width Indicates frame width in units of 8 pixels.

15.6.6.1.7 Luma data slave start address (F0SYSSA)

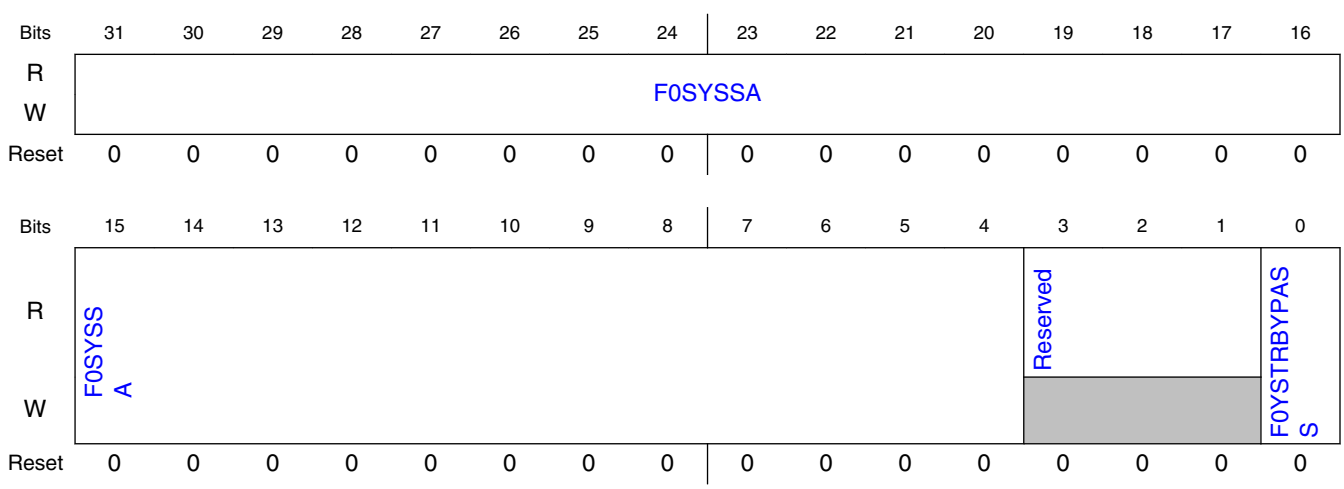
15.6.6.1.7.1 Offset

Register	Offset
F0SYSSA	14h

15.6.6.1.7.2 Function

This register specifies the start address range of the luma data associated to frame 0 and appears at ARADDR_S. Only used when G1/G2 transactions are told by ARADDR_S. It has nothing to do with F0DYDSADDR.

15.6.6.1.7.3 Diagram



15.6.6.1.7.4 Fields

Field	Function
31-4 F0SYSSA	Luma data slave start address Start of address space of slave interface luma data. This address should be 16-byte aligned and unit is 16 bytes, inclusive.
3-1 —	Reserved.
0 F0YSTRBYPASS	Luma Start Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.8 Luma data slave end address (F0SYSEA)

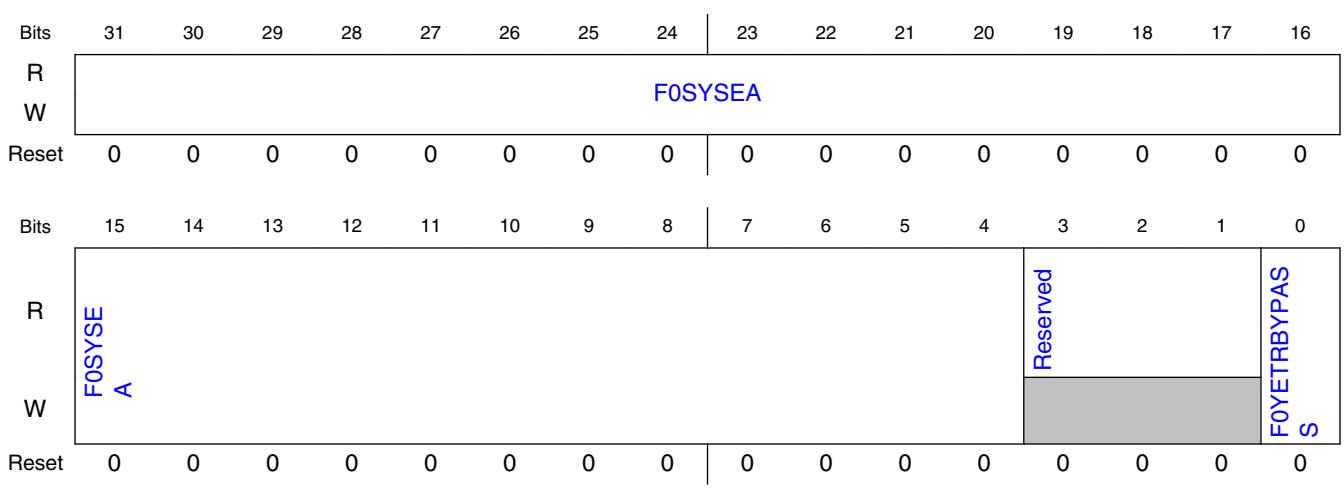
15.6.6.1.8.1 Offset

Register	Offset
F0SYSEA	18h

15.6.6.1.8.2 Function

This register specifies the end address space of luma data.

15.6.6.1.8.3 Diagram



15.6.6.1.8.4 Fields

Field	Function
31-4 F0SYSEA	Luma data slave end address End of address space of slave interface luma data. This address should be 16-byte aligned and unit is 16 bytes, exclusive.
3-1 —	Reserved.
0 F0YETRBYPASS	End Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.9 Chroma data slave start address (F0SUVSSA)

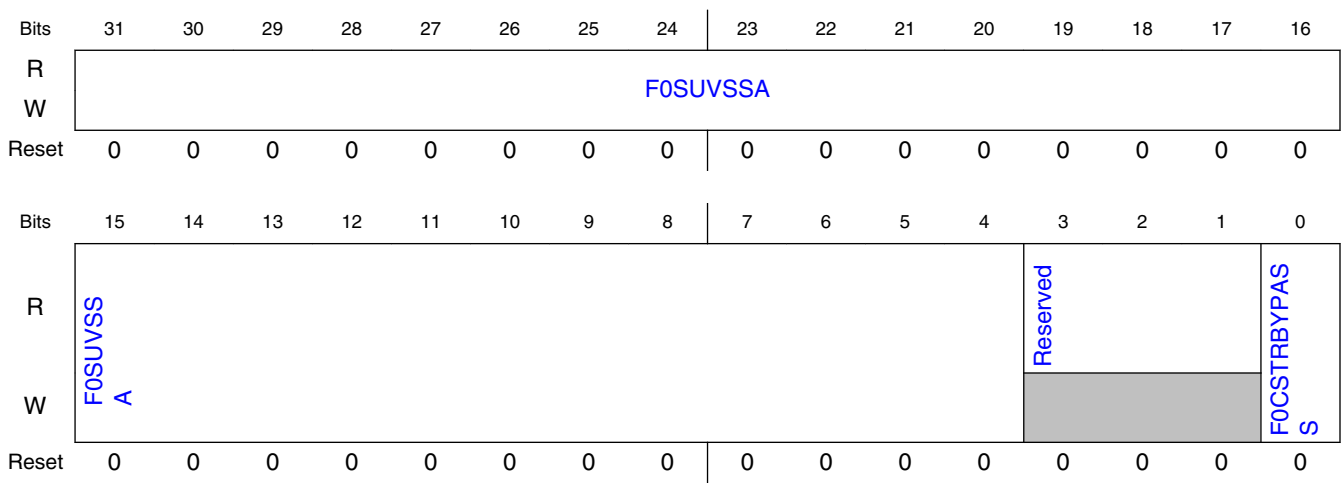
15.6.6.1.9.1 Offset

Register	Offset
F0SUVSSA	1Ch

15.6.6.1.9.2 Function

This register specifies the start address of the address range of the chroma data associated to frame 0 and appears at ARADDR_S. Only used when G2/G1 and non G2/G1 transactions are told by ARADDR_S. It has nothing to do with F0DCDSADDR.

15.6.6.1.9.3 Diagram



15.6.6.1.9.4 Fields

Field	Function
31-4 F0SUVSSA	Chroma data slave start address Start of address space of slave interface chroma data. This address should be 16-byte aligned and unit is 16 bytes, inclusive.
3-1 —	Reserved.
0 F0CSTRBYPASS	Chroma Start Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.10 Chroma data slave end address (F0SUVSEA)

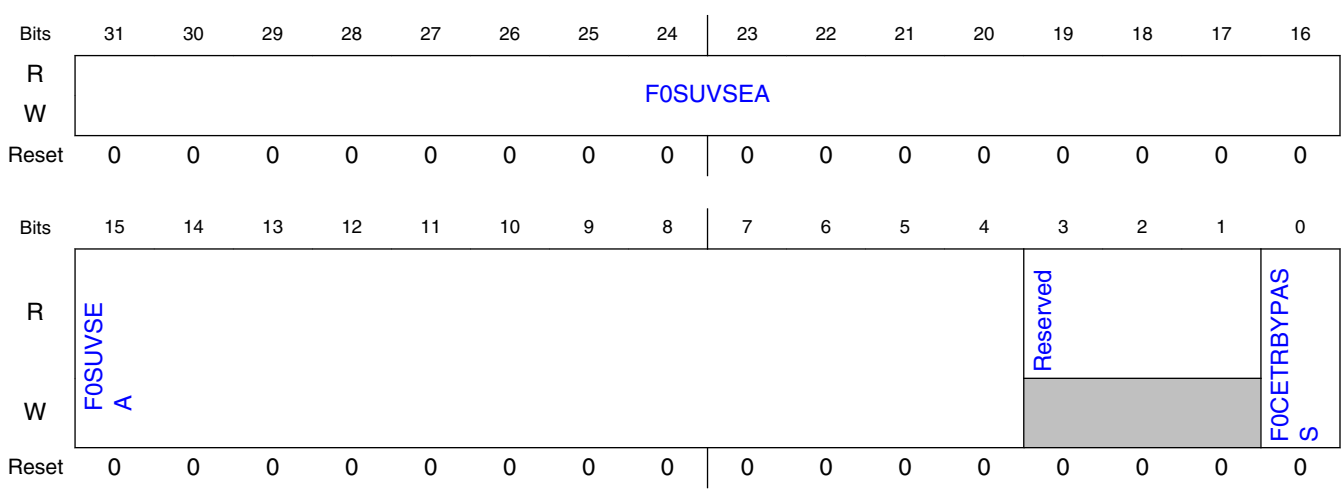
15.6.6.1.10.1 Offset

Register	Offset
F0SUVSEA	20h

15.6.6.1.10.2 Function

This register specifies the end address space of slave interface chroma data.

15.6.6.1.10.3 Diagram



15.6.6.1.10.4 Fields

Field	Function
31-4 F0SUVSEA	Chroma data slave end address End of address space of slave interface chroma data. This address should be 16-byte aligned and unit is 16 bytes, exclusive.
3-1 —	Reserved.
0 F0CETRBYPASS	End Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.11 Cropped picture origin (F0CROPORIG)

15.6.6.1.11.1 Offset

Register	Offset
F0CROPORIG	24h

15.6.6.1.11.2 Function

This register specifies the origin of the cropped picture.

15.6.6.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			F0CROPORIGY												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			F0CROPORIGX												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.11.4 Fields

Field	Function
31-29	Reserved.

Table continues on the next page...

Field	Function
—	
28-16 F0CROPORIGY	Cropped picture y origin Indicates the y origin of the cropped picture in units of pixels. This number must be even. 0 points to the top left corner.
15-13 —	Reserved.
12-0 F0CROPORIGX	Cropped picture x origin Indicates the x origin of the cropped picture in units of pixels. This number must be even. 0 points to the top left corner.

15.6.6.1.12 Cropped picture size (F0CROPSIZE)

15.6.6.1.12.1 Offset

Register	Offset
F0CROPSIZE	28h

15.6.6.1.12.2 Function

This register specifies the size of the cropped picture.

15.6.6.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			F0CROPHEIGHT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			F0CROPWIDTH												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.12.4 Fields

Field	Function
31-29	Reserved.

Table continues on the next page...

Field	Function
—	
28-16 F0CROPHEIGHT	Cropped picture height Indicates the height of the cropped picture in units of pixels. This number must be even.
15-13 —	Reserved.
12-0 F0CROPWIDTH	Cropped picture width Indicates the width of the cropped picture in units of pixels. This number must be even.

15.6.6.1.13 Frame data control (F0DCTL)

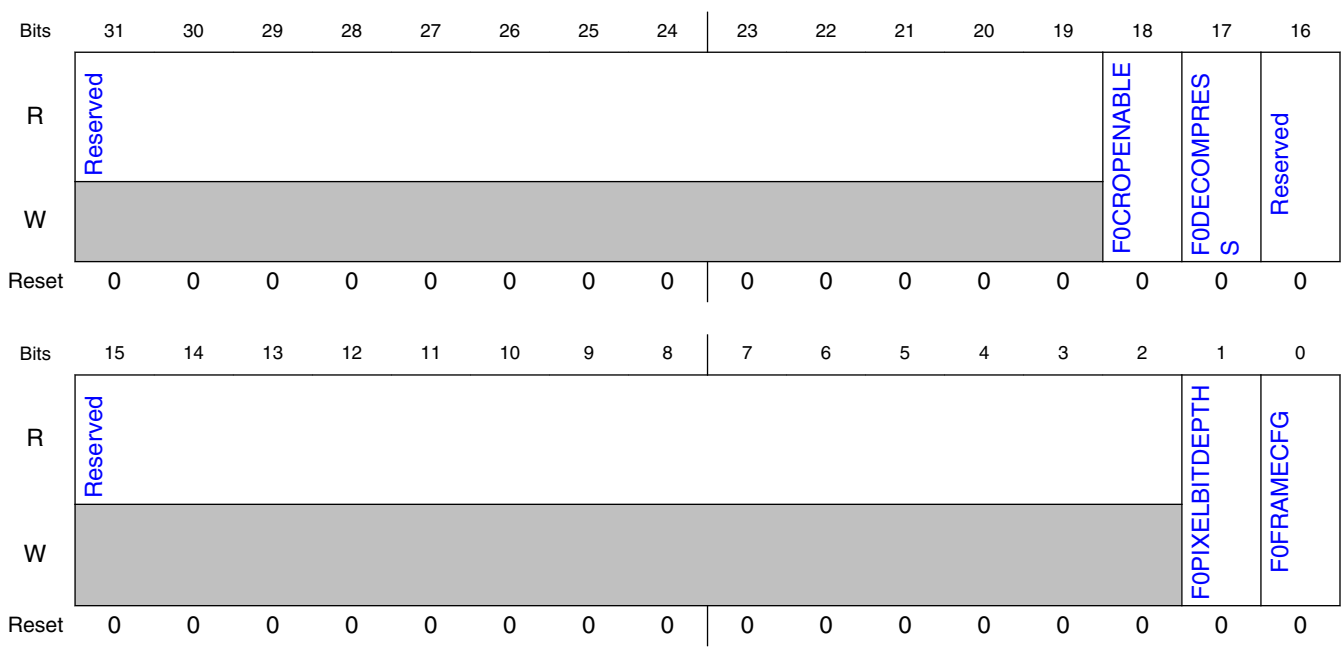
15.6.6.1.13.1 Offset

Register	Offset
F0DCTL	2Ch

15.6.6.1.13.2 Function

Hardware fetches data of Frame - first after reset.

15.6.6.1.13.3 Diagram



15.6.6.1.13.4 Fields

Field	Function
31-19 —	Reserved.
18 F0CROPENABLE	Cropped enable DTRC outputs the cropped picture data according to the settings of Frame0 Cropping Origin Register (F0CROPORIG) and Frame0 Cropping Size Register (F0CROPSIZE).
17 F0DECOMPRESS	Decompress bypass Indicates whether or not decompress is bypassed. For G1, this should always be set to 1. 0b - G2 reference frame is compressed. 1b - G2/G1 reference frame is not compressed.
16 —	Reserved.
15-2 —	Reserved.
1 F0PIXELBITDEPTH	Pixel bit depth Indicates pixel bit depth 0b - 10-bit pixel depth 1b - 8-bit pixel depth
0 F0FRAMECFG	Frame configuration ready Indicates whether or not the frame configuration is ready 0b - Frame 0 configuration is not ready. 1b - Frame 0 configuration is ready and decompress can start for the frame. All other configuration, such as main8/main10 are updated before setting this bit to 1. If there is no G1/G2 video transaction, set this bit to 0.

15.6.6.1.14 Luma video data start address (F1DYDSADDR)

15.6.6.1.14.1 Offset

Register	Offset
F1DYDSADDR	60h

15.6.6.1.14.2 Function

This register specifies the start address for luma video data saved in external memory. This data is compressed or decompressed data.

15.6.6.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F1DYDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F1DYDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.14.4 Fields

Field	Function
31-0 F1DYDSADDR	Luma video data start address Start address for luma data saved in external memory. Unit is byte, but should be 16-byte aligned.

15.6.6.1.15 Chroma video data start address (F1DCDSADDR)

15.6.6.1.15.1 Offset

Register	Offset
F1DCDSADDR	64h

15.6.6.1.15.2 Function

This register specifies the start address for chroma video data saved in external memory.

15.6.6.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F1DCDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F1DCDSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.15.4 Fields

Field	Function
31-0 F1DCDSADDR	Chroma video data start address Start address for chroma decompression data saved in external memory. Unit is byte, but should be 16-byte aligned.

15.6.6.1.16 Luma table data start address (F1DYTSADDR)

15.6.6.1.16.1 Offset

Register	Offset
F1DYTSADDR	68h

15.6.6.1.16.2 Function

This register specifies the start address for luma table data saved in external memory. Not used when compression is disabled in G2.

15.6.6.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F1DYTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F1DYTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.16.4 Fields

Field	Function
31-0 F1DYTSADDR	Luma table data start address Start address for luma table data saved in external memory. Unit is byte and 16-byte aligned.

15.6.6.1.17 Chroma table data start address (F1DCTSADDR)

15.6.6.1.17.1 Offset

Register	Offset
F1DCTSADDR	6Ch

15.6.6.1.17.2 Function

This register specifies the start address for chroma table data saved in external memory. Not used when compression is disabled in G2.

15.6.6.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	F1DCTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	F1DCTSADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.17.4 Fields

Field	Function
31-0 F1DCTSADDR	Chroma table data start address Start address for chroma table data saved in external memory. Unit is byte and 16-byte aligned.

15.6.6.1.18 Frame size (F1SIZE)

15.6.6.1.18.1 Offset

Register	Offset
F1SIZE	70h

15.6.6.1.18.2 Function

This register specifies frame width and frame height in units of 8.

15.6.6.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								F1HEIGHT							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								F1WIDTH							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.18.4 Fields

Field	Function
31-26 —	Reserved.
25-16 F1HEIGHT	Frame height Indicates frame height in units of 8 lines.
15-10 —	Reserved.
9-0 F1WIDTH	Frame width Indicates frame width in units of 8 pixels.

15.6.6.1.19 Luma data slave start address (F1SYSSA)

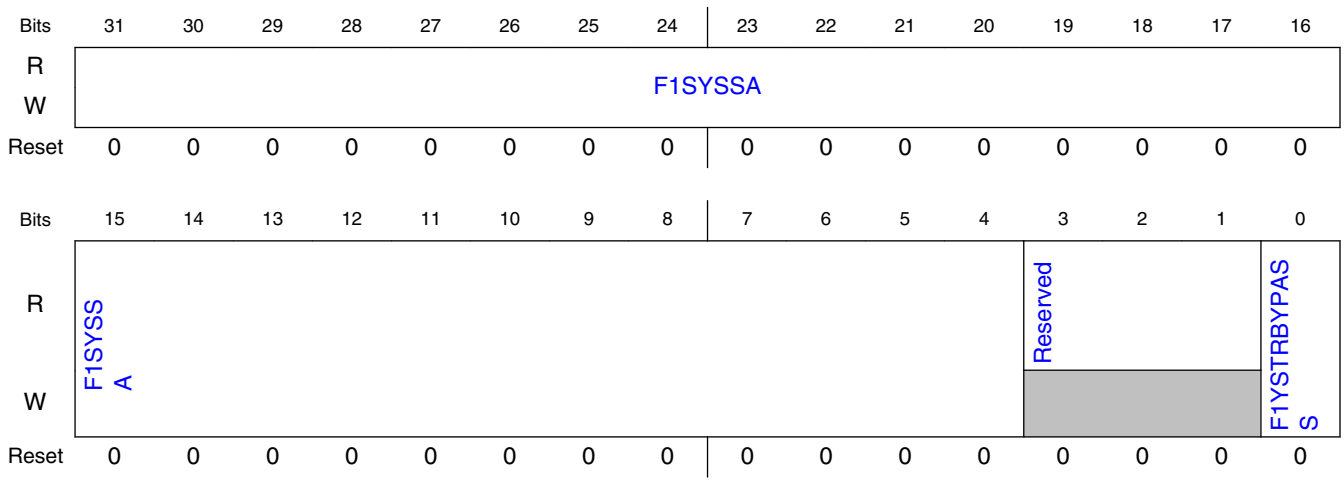
15.6.6.1.19.1 Offset

Register	Offset
F1SYSSA	74h

15.6.6.1.19.2 Function

This register specifies the start address range of the luma data associated to frame 0 and appears at ARADDR_S. Only used when G1/G2 transactions are told by ARADDR_S. It has nothing to do with F1DYDSADDR.

15.6.6.1.19.3 Diagram



15.6.6.1.19.4 Fields

Field	Function
31-4 F1SYSSA	Luma data slave start address Start of address space of slave interface luma data. This address should be 16-byte aligned and unit is 16 bytes, inclusive.
3-1 —	Reserved.
0 F1YSTRBYPASS	Luma Start Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.20 Luma data slave end address (F1SYSEA)

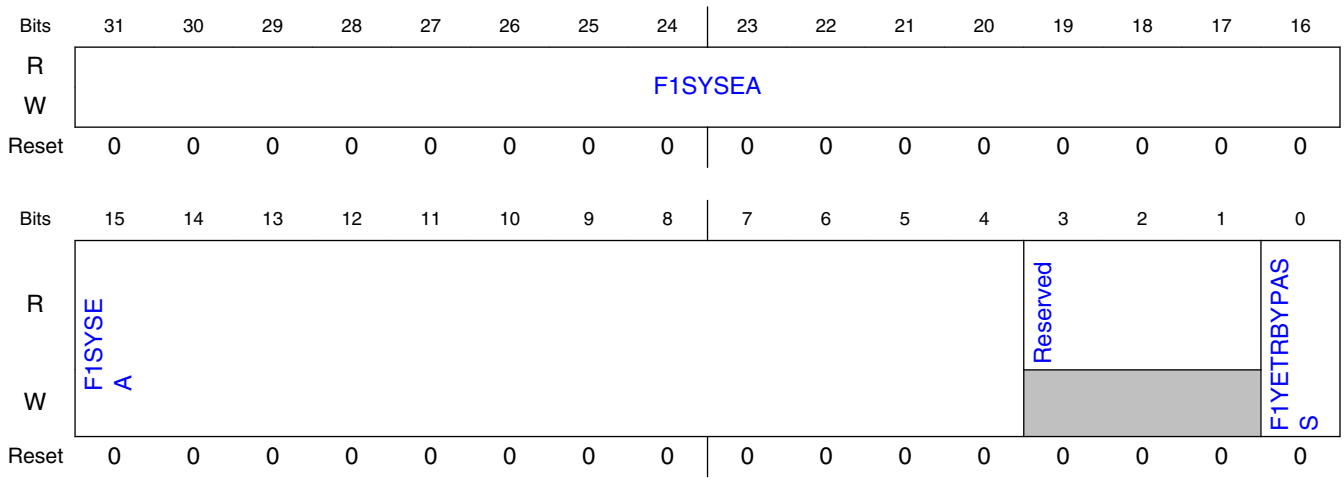
15.6.6.1.20.1 Offset

Register	Offset
F1SYSEA	78h

15.6.6.1.20.2 Function

This register specifies the end address space of luma data.

15.6.6.1.20.3 Diagram



15.6.6.1.20.4 Fields

Field	Function
31-4 F1SYSEA	Luma data slave end address End of address space of slave interface luma data. This address should be 16-byte aligned and unit is 16 bytes, exclusive.
3-1 —	Reserved.
0 F1YETRBYPASS	End Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.21 Chroma data slave start address (F1SUVSSA)

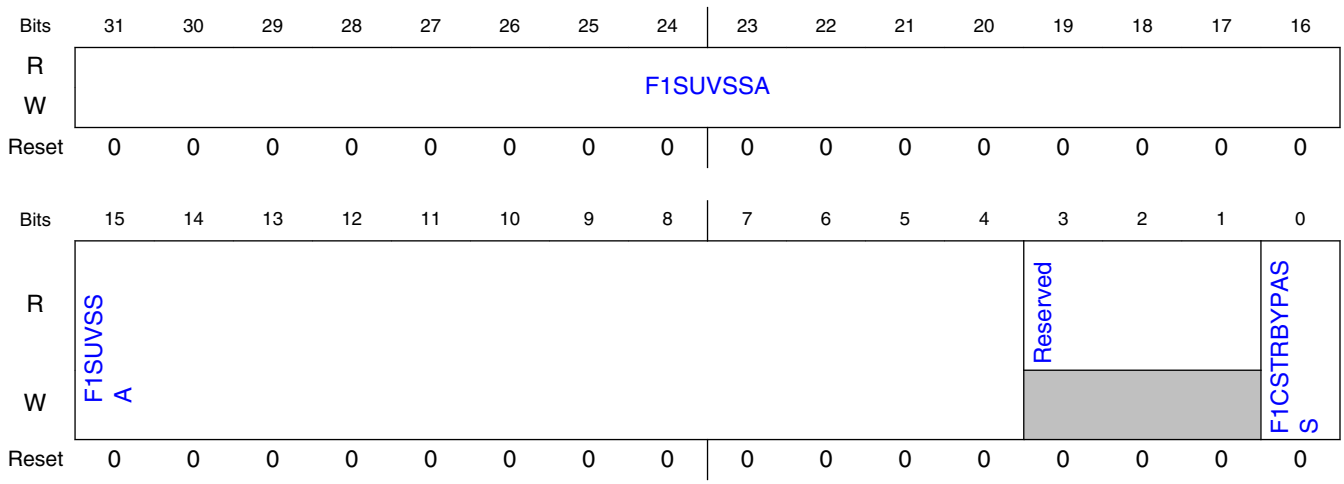
15.6.6.1.21.1 Offset

Register	Offset
F1SUVSSA	7Ch

15.6.6.1.21.2 Function

This register specifies the start address of the address range of the chroma data associated to frame 0 and appears at ARADDR_S. Only used when G2/G1 and non G2/G1 transactions are told by ARADDR_S. It has nothing to do with F1DCDSADDR.

15.6.6.1.21.3 Diagram



15.6.6.1.21.4 Fields

Field	Function
31-4 F1SUVSSA	Chroma data slave start address Start of address space of slave interface chroma data. This address should be 16-byte aligned and unit is 16 bytes, inclusive.
3-1 —	Reserved.
0 F1CSTRBYPASS	Chroma Start Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.22 Chroma data slave end address (F1SUVSEA)

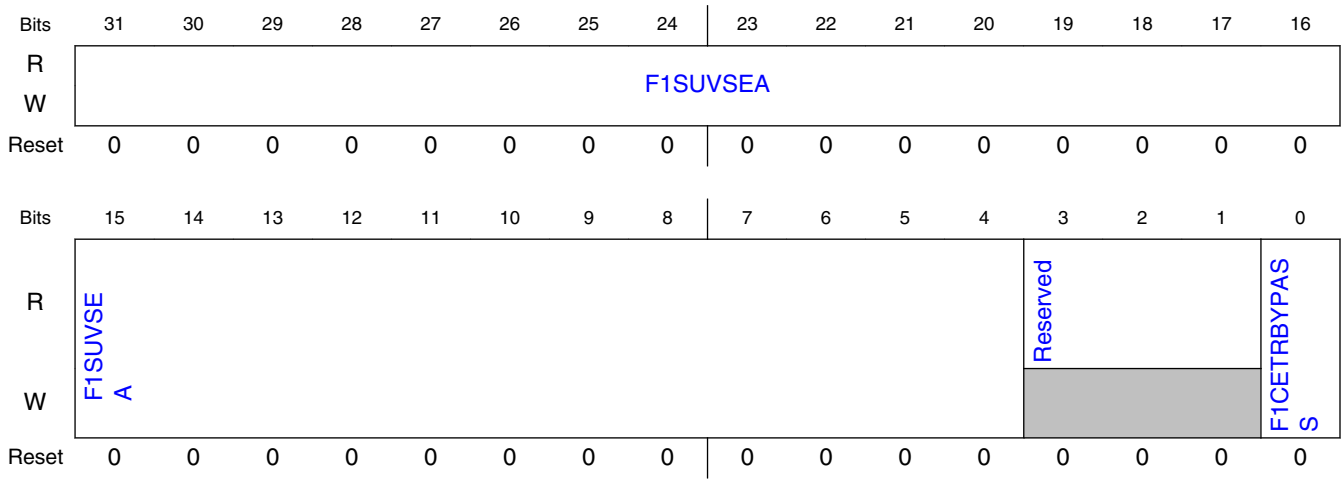
15.6.6.1.22.1 Offset

Register	Offset
F1SUVSEA	80h

15.6.6.1.22.2 Function

This register specifies the end address space of slave interface chroma data.

15.6.6.1.22.3 Diagram



15.6.6.1.22.4 Fields

Field	Function
31-4 F1SUVSEA	Chroma data slave end address End of address space of slave interface chroma data. This address should be 16-byte aligned and unit is 16 bytes, exclusive.
3-1 —	Reserved.
0 F1CETRBYPASS	End Tile to Raster scan Bypass 0b - All ARADDR does NOT bypass the tile-to-rasterscan logic. 1b - All ARADDR bypasses the tile-to-rasterscan logic.

15.6.6.1.23 Cropped picture origin (F1CROPORIG)

15.6.6.1.23.1 Offset

Register	Offset
F1CROPORIG	84h

15.6.6.1.23.2 Function

This register specifies the origin of the cropped picture.

15.6.6.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			F1CROPORIGY												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			F1CROPORIGX												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.23.4 Fields

Field	Function
31-29 —	Reserved.
28-16 F1CROPORIGY	Cropped picture y origin Indicates the y origin of the cropped picture in units of pixels. This number must be even. 0 points to the top left corner.
15-13 —	Reserved.
12-0 F1CROPORIGX	Cropped picture x origin Indicates the x origin of the cropped picture in units of pixels. This number must be even. 0 points to the top left corner.

15.6.6.1.24 Cropped picture size (F1CROPSIZE)

15.6.6.1.24.1 Offset

Register	Offset
F1CROPSIZE	88h

15.6.6.1.24.2 Function

This register specifies the size of the cropped picture.

15.6.6.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			F1CROPHEIGHT												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			F1CROPWIDTH												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.24.4 Fields

Field	Function
31-29 —	Reserved.
28-16 F1CROPHEIGHT	Cropped picture height Indicates the height of the cropped picture in units of pixels. This number must be even.
15-13 —	Reserved.
12-0 F1CROPWIDTH	Cropped picture width Indicates the width of the cropped picture in units of pixels. This number must be even.

15.6.6.1.25 Frame data control (F1DCTL)

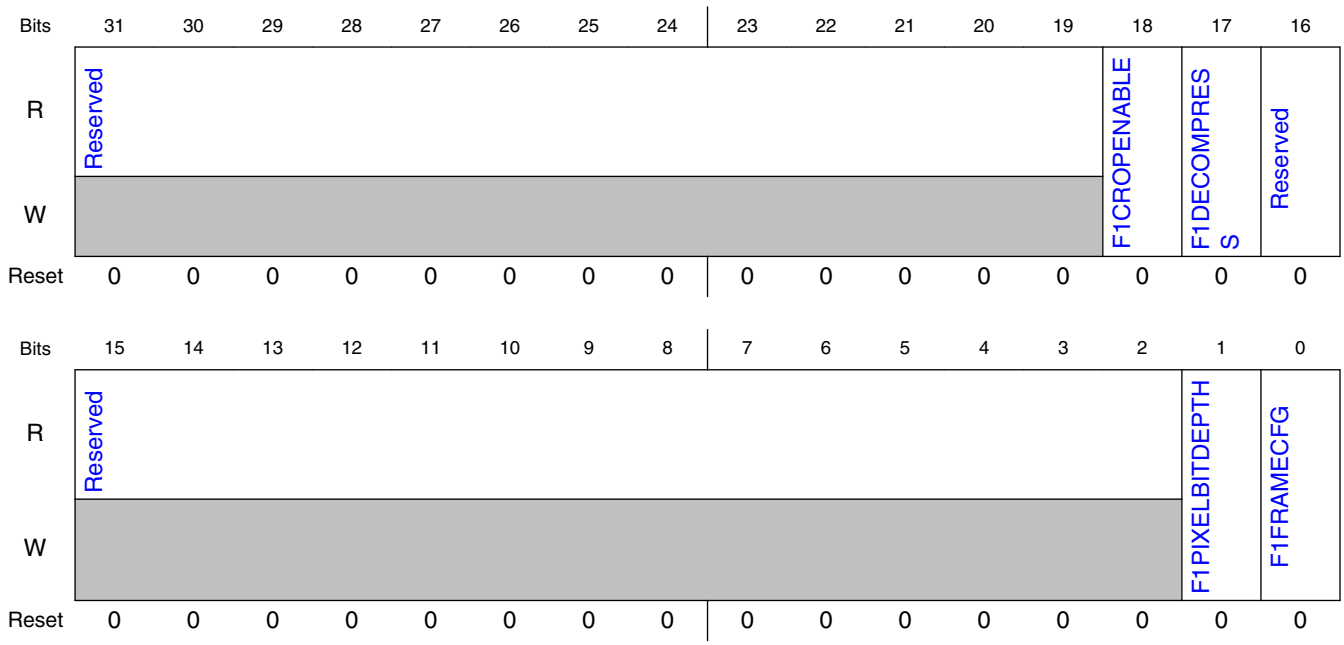
15.6.6.1.25.1 Offset

Register	Offset
F1DCTL	8Ch

15.6.6.1.25.2 Function

Hardware fetches data of Frame - first after reset.

15.6.6.1.25.3 Diagram



15.6.6.1.25.4 Fields

Field	Function
31-19 —	Reserved.
18 F1CROPENABLE	Cropped enable DTRC outputs the cropped picture data according to the settings of Frame0 Cropping Origin Register (F1CROPORIG) and Frame0 Cropping Size Register (F1CROPSIZE).
17 F1DECOMPRESS	Decompress bypass Indicates whether or not decompress is bypassed. For G1, this should always be set to 1. 0b - G2 reference frame is compressed. 1b - G2/G1 reference frame is not compressed.
16 —	Reserved.
15-2 —	Reserved.
1 F1PIXELBITDEPTH	Pixel bit depth Indicates pixel bit depth 0b - 10-bit pixel depth 1b - 8-bit pixel depth
0 F1FRAMECFG	Frame configuration ready Indicates whether or not the frame configuration is ready 0b - Frame 0 configuration is not ready.

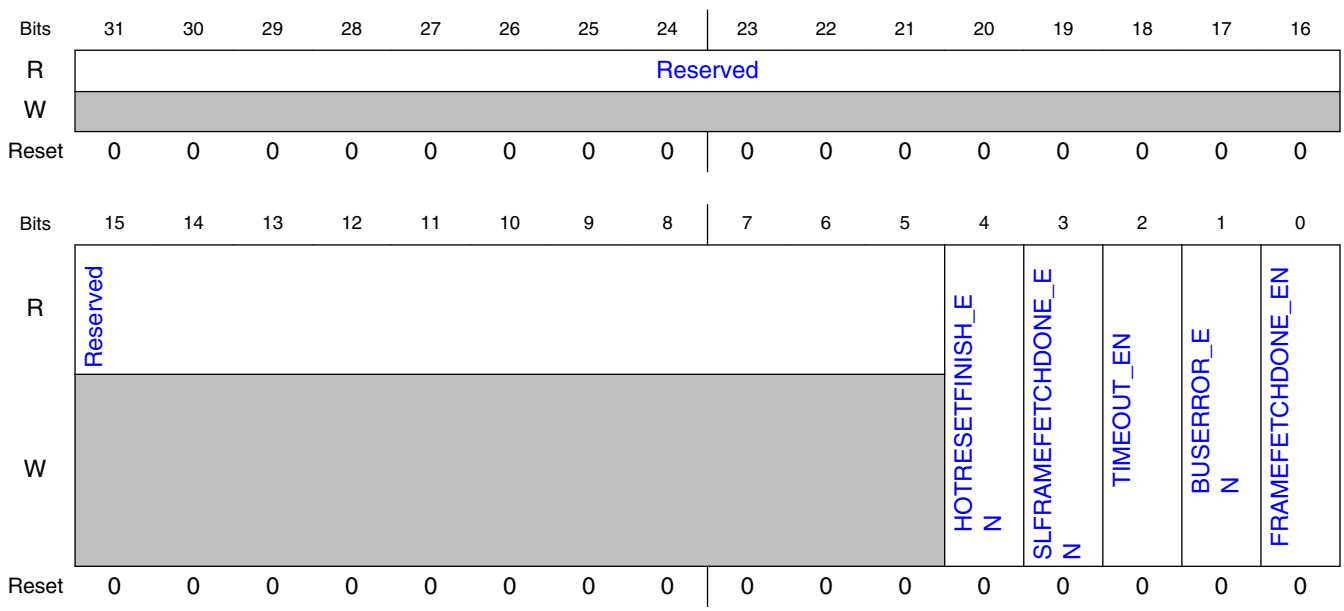
Field	Function
	1b - Frame 0 configuration is ready and decompress can start for the frame. All other configuration, such as main8/main10 are updated before setting this bit to 1. If there is no G1/G2 video transaction, set this bit to 0.

15.6.6.1.26 DTRC Interrupt enables (DTRCINTEN)

15.6.6.1.26.1 Offset

Register	Offset
DTRCINTEN	C0h

15.6.6.1.26.2 Diagram



15.6.6.1.26.3 Fields

Field	Function
31-5 —	Reserved.
4 HOTRESETFINISH_EN	Hot reset finish 0b - Hot reset finish disabled. 1b - Hot reset finish enabled.

Table continues on the next page...

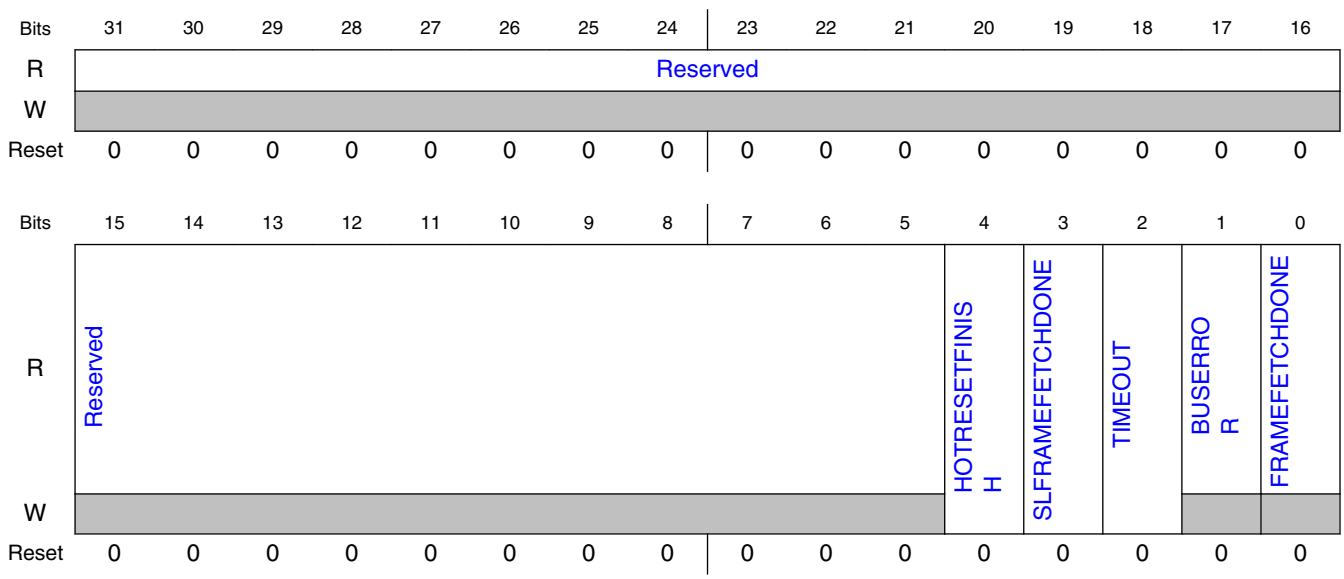
Field	Function
3 SLFRAMEFETCHDONE_EN	Slave frame fetch done 0b - Slave frame fetch done disabled. 1b - Slave frame fetch done enabled.
2 TIMEOUT_EN	Time out enable 0b - Time out disabled. 1b - Time out enabled.
1 BUSERROR_EN	Bus error interrupt enable 0b - Bus error interrupt disabled. 1b - Bus error interrupt enabled.
0 FRAMEFETCHDONE_EN	Frame fetch done interrupt enable 0b - Frame fetch done interrupt disabled. 1b - Frame fetch done interrupt enabled.

15.6.6.1.27 DTRC Interrupt Requests (FDINTR)

15.6.6.1.27.1 Offset

Register	Offset
FDINTR	C4h

15.6.6.1.27.2 Diagram



15.6.6.1.27.3 Fields

Field	Function
31-5 —	Reserved.
4 HOTRESETFINISH	Hot reset finish interrupt Write 1 clear.
3 SLFRAMEFETCHDONE	Slave frame fetch done interrupt Set by hardware when the down-stream IP finishes fetching the data of one frame from DTRC. Software usually ignore this interrupt. Write 1 clear.
2 TIMEOUT	Time out interrupt Set by hardware if the value of PFCR is larger than TOCR. It is cleared after software writes 1'b1.
1 BUSERROR	Bus error interrupt Set by hardware, and cleared after software writes 1'b1
0 FRAMEFETCHDONE	Frame fetch done interrupt Set by hardware and cleared after software writes 1'b1. Software watches this interrupt to configure frame parameters to DTRC.

15.6.6.1.28 DTRC Control (DTCTRL)

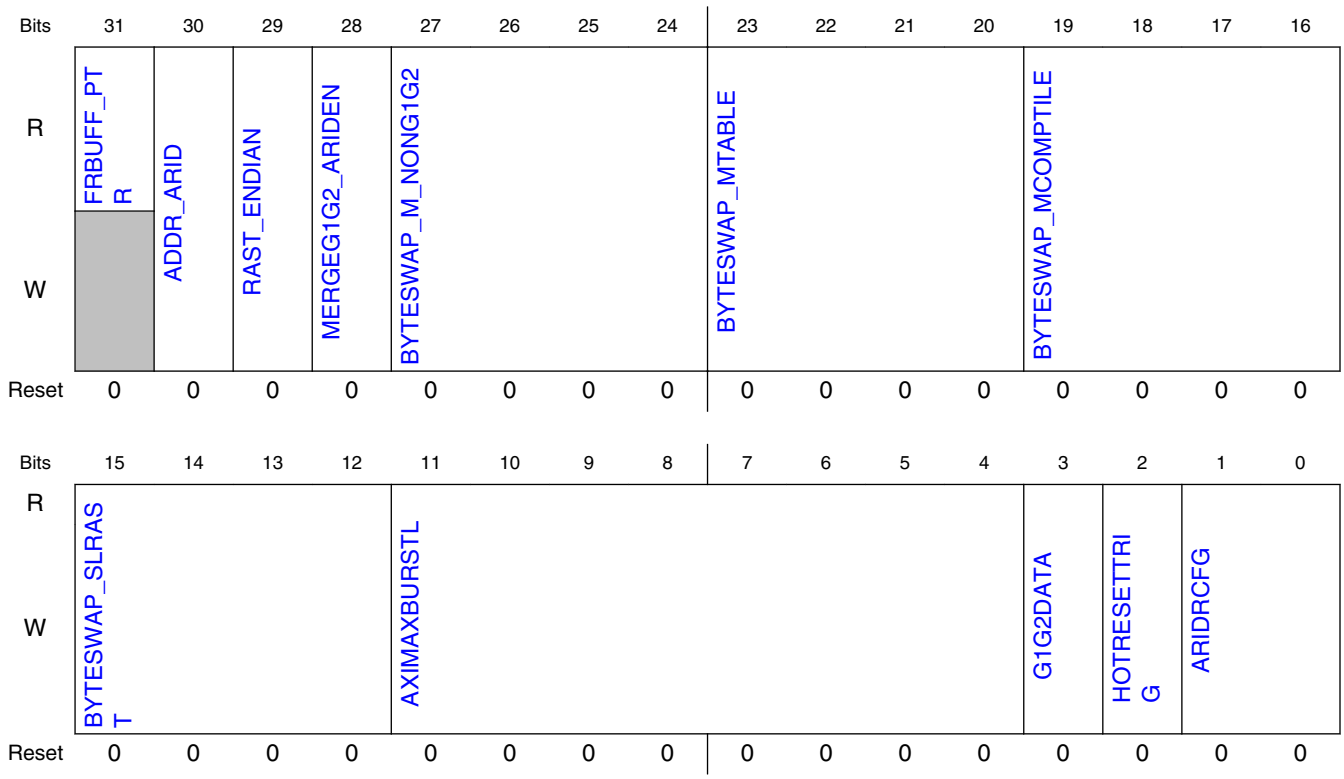
15.6.6.1.28.1 Offset

Register	Offset
DTCTRL	C8h

15.6.6.1.28.2 Function

This register specifies some global control information.

15.6.6.1.28.3 Diagram



15.6.6.1.28.4 Fields

Field	Function
31 FRBUFF_PTR	FRBUFF_PTR Frame buffer configure pointer, to indicate current pointer (Frme0 or Frame1) need to configure. 0b - Configure frame 0 registers. 1b - Configure frame 1 registers.
30 ADDR_ARID	ADDR_ARID Bypass tile-to-rasterscan by Address or AXI ID 0b - By ARID (See bit[1:0] of this register). 1b - By ARADDR
29 RAST_ENDIAN	Raster endian mode This bit indicates the endian mode of DTRC 10-bit raster scan output data. 0b - 10-bit output format is little-endian. Byte swap setting of DTCTRL[15:12] is used. 1b - 10-bit output format is big-endian. Byte swap setting of DTCTRL[15:12] is ignored.
28 MERGEG1G2_ARIDEN	Merge G2/G1 ARID enable 0b - G2/G1 transactions at AXI master interface use different id for table/chroma and data/luma according to DTID2DDR definition. 1b - All G2/G1 transactions at AXI master interface use the same id configured in DTID2DDR[15:8] Please note that DTID2DDR[15:8] and DTID2DDR[7:0] still need to be set the same way as when DTCTRL[28] is 0.
27-24	Byte swap mode for master interface non-G1/G2 data

Table continues on the next page...

Field	Function
BYTESWAP_M_NONG1G2	• bit 4: 8 bit swap • bit 5: 16 bit swap • bit 6: 32 bit swap • bit 7: 64 bit swap • 4'b0000: 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 (the original bit sequence) • 4'b0001: 1-0-3-2-5-4-7-6… • 4'b0010: 2-3-0-1-6-7-4-5… • 4'b0011:3-2-1-0-7-6-5-4… • 4'b0100: 4-5-6-7-0-1-2-3… • ... • 4'b0111: 7-6-5-4-3-2-1-0… • 4'b1000: 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7… • ... • 4'b1111: 15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
23-20 BYTESWAP_MTABLE	Byte swap mode for master interface table data • bit 4: 8 bit swap • bit 5: 16 bit swap • bit 6: 32 bit swap • bit 7: 64 bit swap • 4'b0000: 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 (the original bit sequence) • 4'b0001: 1-0-3-2-5-4-7-6… • 4'b0010: 2-3-0-1-6-7-4-5… • 4'b0011:3-2-1-0-7-6-5-4… • 4'b0100: 4-5-6-7-0-1-2-3… • ... • 4'b0111: 7-6-5-4-3-2-1-0… • 4'b1000: 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7… • ... • 4'b1111: 15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
19-16 BYTESWAP_MCOMPTILE	Byte swap mode for master interface compressed data and tiled data • bit 4: 8 bit swap • bit 5: 16 bit swap • bit 6: 32 bit swap • bit 7: 64 bit swap • 4'b0000: 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 (the original bit sequence) • 4'b0001: 1-0-3-2-5-4-7-6… • 4'b0010: 2-3-0-1-6-7-4-5… • 4'b0011:3-2-1-0-7-6-5-4… • 4'b0100: 4-5-6-7-0-1-2-3… • ... • 4'b0111: 7-6-5-4-3-2-1-0… • 4'b1000: 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7… • ... • 4'b1111: 15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
15-12 BYTESWAP_SLRAST	Byte swap mode for slave interface raster scan data • bit 4: 8 bit swap • bit 5: 16 bit swap • bit 6: 32 bit swap • bit 7: 64 bit swap • 4'b0000: 0-1-2-3-4-5-6-7-8-9-10-11-12-13-14-15 (the original bit sequence) • 4'b0001: 1-0-3-2-5-4-7-6…

Table continues on the next page...

Field	Function
	• 4'b0010: 2-3-0-1-6-7-4-5… • 4'b0011:3-2-1-0-7-6-5-4… • 4'b0100: 4-5-6-7-0-1-2-3… • ... • 4'b0111: 7-6-5-4-3-2-1-0… • 4'b1000: 8-9-10-11-12-13-14-15-0-1-2-3-4-5-6-7… • ... • 4'b1111: 15-14-13-12-11-10-9-8-7-6-5-4-3-2-1-0
11-4 AXIMAXBURSTL	Maximum burst length of AXI master port If maximum burst length feature is supported (see DTRCCONFIG[0]), DTRC sends the G1/G2 relating requests with ARLEN less than or equal to (the maximum burst length - 1).
3 G1G2DATA	G2 or G1 source data When this bit is set, F0DCTL[17] and F1DCTL[17] must be 1'b1. DTRC converts G1 tile data to raster scan data. 0b - The source data is G2 data. 1b - The source data is G1 tile data
2 HOTRESETTRIG	Hot reset trigger Software writes 1 to initiate a hot reset of video decompression and detile path. Hardware clears the bit to 0 after the reset finished.
1-0 ARIDRCFG	ARIDR configuration Specify ARIDR mode. 00b - All ARID is de-tiled. 01b - ARID in ARIDR is de-tiled, and others are bypass. NOTE: ARID[0] specify decode luma or chroma, so 4 ARIDs in ARIDR should include both of ARID[0] ==1 and ARID[0] ==0. 10b - ARID in ARIDR is bypass_de-tile, and others are de-tiled. 11b - Same as 2'b01.

15.6.6.1.29 ARIDR (ARIDR)

15.6.6.1.29.1 Offset

Register	Offset
ARIDR	CCh

15.6.6.1.29.2 Function

This register specifies ARID to use to fetch de-tile data or bypass de-tile data.

15.6.6.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	ARIDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ARIDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.29.4 Fields

Field	Function
31-0 ARIDR	ARIDR 4 ARID, each is 8 bits.

15.6.6.1.30 DTID2DDR (DTID2DDR)

15.6.6.1.30.1 Offset

Register	Offset
DTID2DDR	D0h

15.6.6.1.30.2 Function

This register specifies ARID to use to fetch video data from DTRC to external memory.

15.6.6.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ARID_TABLE								ARID_COMPR							
Reset	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	0

15.6.6.1.30.4 Fields

Field	Function
31-16 —	Reserved.
15-8 ARID_TABLE	ARID_TABLE If G2 frame is compressed, this is ARID of table fetching, from DTRC to external memory. If G2 frame is not compressed or G1 tile is the case, this is ARID of chroma data The value should be different from non-G2/G1 transaction ARID received from DTRC AXI slave port, and different from bits 7:0 of this register.
7-0 ARID_COMPR	ARID_COMPR If G2 frame is compressed, this is ARID of compressed data fetching, from DTRC to external memory. If G2 frame is not compressed or G1 tile is the case, this is ARID of luma data. The value should be different from non-G2/G1 transaction ARID received from DTRC AXI slave port, and different from bits 15:8 of this register.

15.6.6.1.31 DTRCCONFIG (DTRCCONFIG)

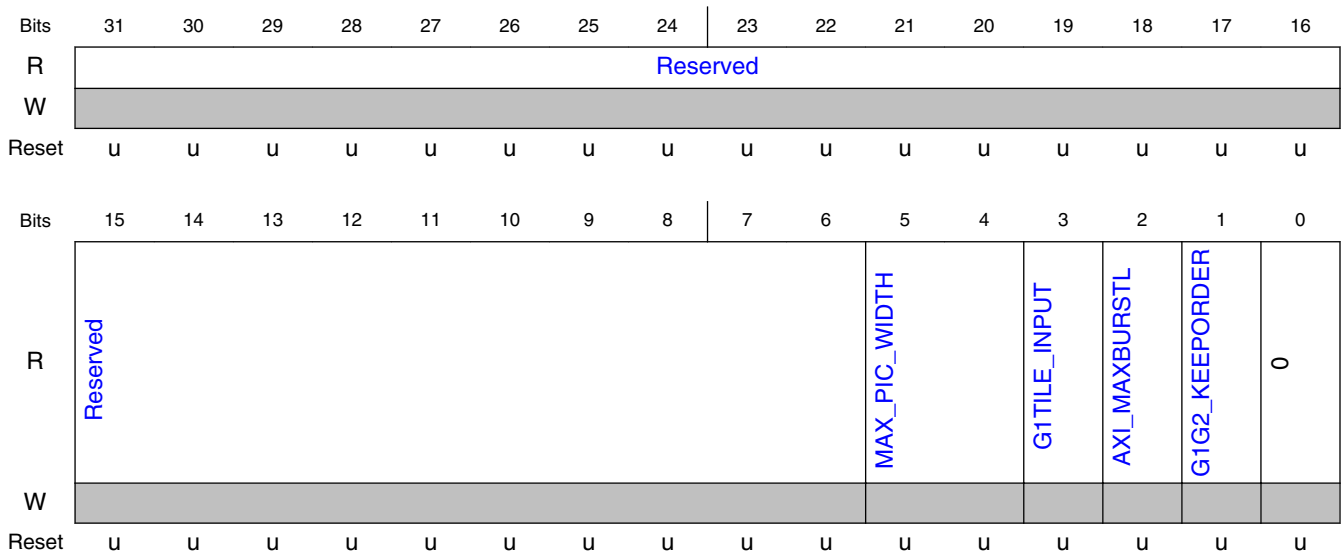
15.6.6.1.31.1 Offset

Register	Offset
DTRCCONFIG	D4h

15.6.6.1.31.2 Function

This register specifies ARID to use to fetch video data from DTRC to external memory.

15.6.6.1.31.3 Diagram



15.6.6.1.31.4 Fields

Field	Function
31-6 —	Reserved.
5-4 MAX_PIC_WIDTH	MAX_PIC_WIDTH Maximum supported picture width 00b - 4096 01b - 1920 10b - Reserved. 11b - Reserved.
3 G1TILE_INPUT	G1TILE_INPUT G1 tile input 0b - Not supported. 1b - Supported.
2 AXI_MAXBURSTL	AXI_MAXBURSTL Maximum burst length of AXI master port. 0b - Not supported. 1b - Supported. DTRC sends the G1/G2 relating requests with ARLEN <= (the maximum burst length - 1). The maximum burst length is set by DTCTRL[16:8].
1 G1G2_KEEPORDER	G1G2_KEEPORDER Keep the order between g1/g2 transactions and non-g1/g2 transactions at AXI slave interface. 0b - Not supported. DTRC sends out data transactions as soon as they are ready regardless of the address transactions order, The master connected to AXI slave interface must recognize the data transactions by the RID. 1b - Supported, DTRC ensure return read data from a sequence of read transactions in the same order in which it received the address.
0 —	Reserved.

15.6.6.1.32 DTRC Version (DTRCVERSION)

15.6.6.1.32.1 Offset

Register	Offset
DTRCVERSION	D8h

15.6.6.1.32.2 Function

This is a constant read-only register that contains encoded information about the hardware version.

15.6.6.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	MAJOR						MINOR						CUST_VERSION			
W																
Reset	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u	u

15.6.6.1.32.4 Fields

Field	Function
31-16 —	Reserved.
15-10 MAJOR	MAJOR Major version. • 0x00: Version before 0831, including 0831. Support G2 tile + G2 compressed + raster scan output • 0x01: Support G1 tile + G2 tile + G2 compressed + raster scan output. • All other values: Reserved.
9-4 MINOR	MINOR Minor version. Simply an update count (update at a release time) of a certain Major version. Count from 0.
3-0 CUST_VERSION	CUST_VERSION Customer Version.

15.6.6.1.33 Performance counter control (PFCTRL)

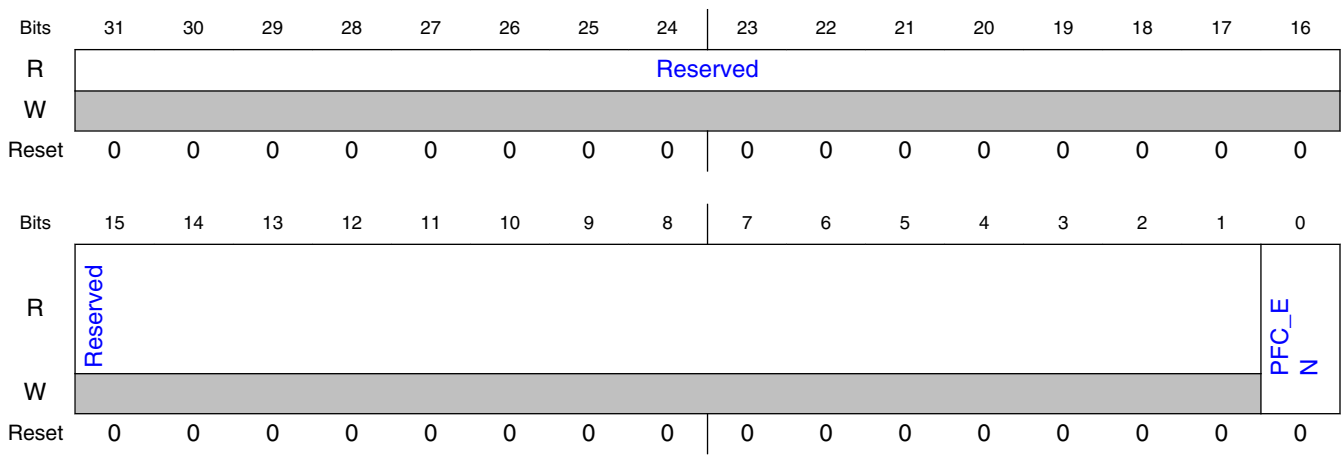
15.6.6.1.33.1 Offset

Register	Offset
PFCTRL	F0h

15.6.6.1.33.2 Function

This register enables or disables performance counter.

15.6.6.1.33.3 Diagram



15.6.6.1.33.4 Fields

Field	Function
31-1 —	Reserved.
0 PFC_EN	PFC_EN Performance Counter Enable. 0b - Performance Counter disabled. 1b - Performance Counter enabled.

15.6.6.1.34 Performance counter (PFCR)

15.6.6.1.34.1 Offset

Register	Offset
PFCR	F4h

15.6.6.1.34.2 Function

This is performance counter to count total clock cycle from DTRC begins to send out a frame and finishes sending out the frame.

This counter belongs to decompression clock domain, not APB clock domain. The software can read this counter after received slave frame fetch done interrupt.

15.6.6.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	PFCR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PFCR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.34.4 Fields

Field	Function
31-0 PFCR	Performance Counter

15.6.6.1.35 Time Out Cycles (TOCR)

15.6.6.1.35.1 Offset

Register	Offset
TOCR	F8h

15.6.6.1.35.2 Function

This register sets the clock cycles (decompression clock domain) to wait before launching the time-out interrupt. When the value of performance counter register is larger than the value of this register, the time-out interrupt is set.

15.6.6.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	TOCR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	TOCR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.6.6.1.35.4 Fields

Field	Function
31-0 TOCR	Time Out Cycles

15.7 Display, Prefetch and Resolve (DPR)

15.7.1 Overview

Display prefetch resolve, or DPR, is the process of fetching display data before the display pipeline needs the data to drive pixels in the active display region. This data is transformed, or resolved, from a variety of tiled buffer formats into linear format.

In a system with many bus masters, arbitration and transaction reordering, significant delays in the response time of returned read data can occurr. Therefore, the prefetch process is required to issue transaction sequences which are processed with a high level of DRAM efficiency.

The resolve process is required to transform tile formatted frame buffers into linear buffers. Displays process data in linear scan line order. In the resolve process, local memory is used to store a small number of scan lines which the local display controller will source to refresh the panel. The DPR works with a double/triple bank memory structure. This memory structure is implemented in the RTRAM_CTRL and the banks are referred to as A, B, and optionally C. The DPR is configurable to work with either two or three banks, and each bank is either 4 or 8 lines high depending on the source frame buffer format. Zenverge VPU 2-Plane 420 source buffers use RTRAM banks that are 8 lines high. All other GPU, VP9 and VPU 1-Plane buffers use RTRAM banks that are 4 lines high.

15.7.1.1 Block Diagram

The following diagram gives a high-level overview of the configuration of the DPR.

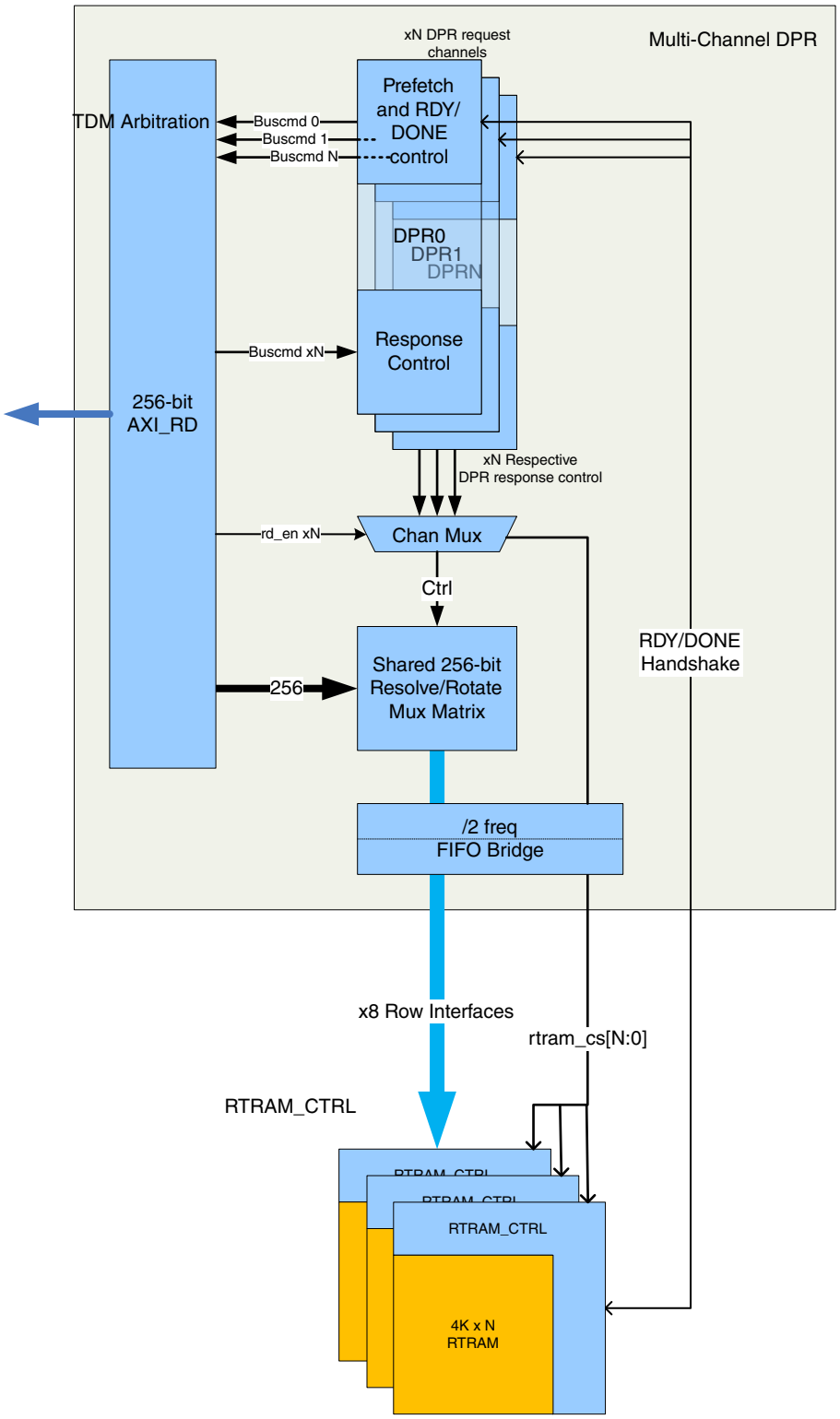


Figure 15-19. DPR block diagram

15.7.1.2 Features

The DPR module is capable of supporting the following features:

- Double/Triple (A,B and opt C) RTRAM buffers.
- Resolves and rotates at the bus width, 256-bit, without stalling the input bus.
- 32bpp ARGB pixel swapping
- 1-Plane YUV422 YUV pixel swapping
- 2-Plane 8bpp YUV420 UV pixel swapping
- RTRAM panic and stall underrun detection.

The supported prefetch and resolve modes are listed in the table below:

Mode	Linear or Tile	HFLIP	VFLIP	Rotation (0, 90, 180 or 270)	Lines per RTRAM Bank
GPU 32bpp xRGB	Linear	Yes	Yes	0 and 180 only	4
GPU 32bpp xRGB	X-Major Supertile and Standard tile	Yes	Yes	0, 90, 180 or 270	4
GPU 16bpp xRGB, 1-Plane YUV422	Linear	Yes	Yes	0 and 180 only	4
GPU 16bpp xRGB, 1-Plane YUV422	X-Major Supertile and Standard tile	Yes	Yes	0 and 180 only	4
VPU 8bpp 2-Plane YUV420	Linear	Yes	Yes	0 and 180 only	8
VPU 8bpp 2-Plane YUV420	Zenverge tile	Yes	Yes	0, 90, 180 or 270	8
VPU 10bpp 2-Plane YUV420	Linear	No	Yes	0 only	8
VPU 10bpp 2-Plane YUV420	Zenverge tile	No	Yes	0 only	8
VP9 8bpp 2-Plane YUV420	Linear	Yes	Yes	0 and 180 only	4
VP9 8bpp 2-Plane YUV420	Google tile	Yes	Yes	0 and 180 only	4
VP9 10bpp 2-Plane YUV420	Linear	No	Yes	0 only	4
VP9 10bpp 2-Plane YUV420	Google tile	No	Yes	0 only	4

For detailed GPU and VPU tile specifications, refer to the following application documents:

- Vivante.ArchNote.GPU.TileSpec-v2.0-20141003.pdf
- VPU_memory_tiling_v0.4.09092015.docx
- G-Series_2_v3_HW_Integration_Guide-2015-06-29.pdf

The following features are not supported:

- Progressive to interlaced conversion by dropping lines
- 3-plane YUV source buffers
- 2-plane to 1-plane conversion
- 90/270 rotation for all linear, all 16bpp and 10bpp 2-Plane YUV422 sources
- 10-bit packed input will be maintained as 10-bit packed output. The output will not unpack the compnents to 32/24/16-bit alignment.

15.7.1.3 DPR Signal Descriptions

Signal	Description	I/O
rtr_m_clk	RTMEM memory clock	I
rtr_aresetn	RTMEM async reset	I
dpr_apb_clk	APB Control register clock	I
dpr_b_clkg	APB Control register gated clock	I
dpr_apb_aresetn	APB async reset	I
dpr_b_clk	DPR system clock	I
dpr_b_clkg	DPR system gated clock	I
dpr_chip_resetn	DPR system async reset	I
dpr_b_clkg_en	DPR system gated clock enable	O
dpr_irq[3-1:0]	DPR_IRQ per dpr_ctrl block	O
tg2dpr_start_en[3-1:0]	Logic high enable from timing generator (during VSYNC), to restart DPR with current CSR settings	I
tg2dpr_shadow_load_en[3-1:0]	Logic high strobe from timing generator indicating load shadow registers to active registers on next start.	I
dpr_apb_sel[3-1:0]	APB Select	I
dpr_apb_enable	APB Enable	I
dpr_apb_addr[16-1:0]	APB Address	I
dpr_apb_write	APB Write=1, Read=0	I
dpr_apb_wdata[31:0]	APB Write Data	I
dpr_apb_rdata[31:0]	APB Read Data	O
dpr_apb_ready	APB Ready	O
dpr_apb_slverr	APB Slave Error	O
dpr_axi_ar	AXI Read Command Channel Bus	O
dpr_axi_arready	AXI Read Command Channel Ready	I
dpr_axi_r	AXI Read Data Channel Bus	I
dpr_axi_rready	AXI Read Data Channel Ready	O
rtr2dpr_abort_yrgb[(3*3)-1:0]	1=abort YRGB buffer being written to RTMEM bank	I
rtr2dpr_abort_uv[(3*3)-1:0]	1=abort UV buffer being written to RTMEM bank	I

Table continues on the next page...

Signal	Description	I/O
rtr2dpr_buf_done_yrgb[(3*3)-1:0]	YRGB buffer becomes available to prefetch	I
rtr2dpr_buf_done_uv[(3*3)-1:0]	UV buffer becomes available to prefetch	I
dpr2rtr_ch_done_yrgb[3-1:0]	DPR completed prefetch/resolve of the YRGB channel	O
dpr2rtr_ch_done_uv[3-1:0]	DPR completed prefetch/resolve of the UV channel	O
dpr2rtr_buf_rdy_yrgb[(3*3)-1:0]	DPR has completed filling current YRGB row of blocks in the RTRAM buffer	O
dpr2rtr_buf_rdy_uv[(3*3)-1:0]	DPR has completed filling current UV row of blocks in the RTRAM buffer	O
dpr2rtr_cs[3-1:0]	Chip select which RTMEM is being written	O
dpr2rtr_row_data[(64*8)-1:0]	RTMEM row data	O
dpr2rtr_row_uv_sel	2-Plane UV memory select	O
dpr2rtr_bank_addr[1:0]	memory bank address	O
dpr2rtr_row_addr[13:3]	memory row address	O
dpr2rtr_row_sel[2:0]	memory row select	0
dpr2rtr_csr_mode[(3*3)-1:0]	RTRAM Mode Control	O
dpr2rtr_csr_num_rows[3-1:0]	Number of RTRAM lines/buffer: 0=8 lines, 1=4 lines	O
dpr2rtr_csr_num_planes[3-1:0]	0=Enable 1-Plane, 1=Enable 2-Plane	O
dpr2rtr_csr_num_rows_active[3-1:0]	0=Row0-4 enabled. 1=Row0-6, all rows, enabled	O
dpr2rtr_csr_thres_high[(3*3)-1:0]	panic high water mark threshold	O
dpr2rtr_csr_thres_low[(3*3)-1:0]	panic low water mark threshold	O
dpr2rtr_csr_abort_sel[3-1:0]	0=Panic Select. 1=Abort Select	O
dpr2rtr_csr_vflip_en[3-1:0]	1=vflip enable, 0=not vertical flipped	O
ipt_scan_mode	scan mode enable	I
ipt_se_gatedclk	ICG gated clock test enable	I

15.7.2 Functional description

15.7.2.1 Clocking and Resets

15.7.2.1.1 Functional clocks

The DPR b_clk clock, RTRAM rtr_m_clk and APB apb_clk clock domains are all asynchronous. The following clocking restriction exists in the DPR: m_clk >= b_clk/2

15.7.2.1.2 Chip resets

There is a seperate asynchronous chip reset for each clock domain. The DPR implements reset synchronizers for each asynchronous reset, to deassert each reset synchronously.

15.7.2.2 DPR Sub-blocks

15.7.2.2.1 DPR Prefetch Control

This control engine is responsible for prefetching the source frame data. Request sizes are software programmable in a range from 64 bytes to 4k bytes. The prefetch engine is designed so the RTRAM buffer will always fill from left to right, top to bottom across the line regardless of the output image orientation.

15.7.2.2.2 DPR Response Control

The response control block is responsible for managing the DPR datapath resolve mux, horizontal flip, vertical flip and image rotation.

15.7.2.2.3 DPR Datapath

The DPR datapath block is a shared across all DPR channel response blocks because on any given clock cycle, pixels for only 1 DPR response channel will be returned by the system. The datapath implements pixel cross bar logic for tile resolve, horizontal flip, vertical flip and image rotation by 0, 90, 180 or 270 degrees.

15.7.3 Memory Map and Registers

15.7.3.1 register descriptions

15.7.3.1.1 DPR Memory map

DPR1 base address: 1_8000h

DPR2 base address: 1_9000h

DPR3 base address: 1_A000h

Offset	Register	Width (In bits)	Access	Reset value
0h	System Control 0 (SYSTEM_CTRL0)	32	RW	0000_0010h
4h	System Control 0 (SYSTEM_CTRL0_SET)	32	RW	0000_0010h
8h	System Control 0 (SYSTEM_CTRL0_CLR)	32	RW	0000_0010h
Ch	System Control 0 (SYSTEM_CTRL0_TOG)	32	RW	0000_0010h
20h	Interrupt Mask (IRQ_MASK)	32	RW	0000_0000h
24h	Interrupt Mask (IRQ_MASK_SET)	32	RW	0000_0000h
28h	Interrupt Mask (IRQ_MASK_CLR)	32	RW	0000_0000h
2Ch	Interrupt Mask (IRQ_MASK_TOG)	32	RW	0000_0000h
30h	Status Register of Masked IRQ (IRQ_MASK_STATUS)	32	RO	0000_0000h
34h	Status Register of Masked IRQ (IRQ_MASK_STATUS_SET)	32	RO	0000_0000h
38h	Status Register of Masked IRQ (IRQ_MASK_STATUS_CLR)	32	RO	0000_0000h
3Ch	Status Register of Masked IRQ (IRQ_MASK_STATUS_TOG)	32	RO	0000_0000h
40h	Status of Non-Masked IRQ (IRQ_NONMASK_STATUS)	32	W1C	0000_0000h
44h	Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_SET)	32	W1C	0000_0000h
48h	Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_CLR)	32	W1C	0000_0000h
4Ch	Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_TOG)	32	W1C	0000_0000h
50h	Mode Control 0 (MODE_CTRL0)	32	RW	0000_0000h
54h	Mode Control 0 (MODE_CTRL0_SET)	32	RW	0000_0000h
58h	Mode Control 0 (MODE_CTRL0_CLR)	32	RW	0000_0000h
5Ch	Mode Control 0 (MODE_CTRL0_TOG)	32	RW	0000_0000h
70h	Frame Control 0 (FRAME_CTRL0)	32	RW	0000_0000h
74h	Frame Control 0 (FRAME_CTRL0_SET)	32	RW	0000_0000h
78h	Frame Control 0 (FRAME_CTRL0_CLR)	32	RW	0000_0000h
7Ch	Frame Control 0 (FRAME_CTRL0_TOG)	32	RW	0000_0000h
90h	Frame 1-Plane Control 0 (FRAME_1P_CTRL0)	32	RW	0000_0000h
94h	Frame 1-Plane Control 0 (FRAME_1P_CTRL0_SET)	32	RW	0000_0000h
98h	Frame 1-Plane Control 0 (FRAME_1P_CTRL0_CLR)	32	RW	0000_0000h
9Ch	Frame 1-Plane Control 0 (FRAME_1P_CTRL0_TOG)	32	RW	0000_0000h
A0h	Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL)	32	RW	0000_0000h
A4h	Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_SET)	32	RW	0000_0000h
A8h	Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_CLR)	32	RW	0000_0000h
ACh	Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_TOG)	32	RW	0000_0000h
B0h	Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL)	32	RW	0000_0000h
B4h	Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_SET)	32	RW	0000_0000h
B8h	Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_CLR)	32	RW	0000_0000h
BCh	Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_TOG)	32	RW	0000_0000h
C0h	Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0)	32	RW	0000_0000h
C4h	Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_SET)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
C8h	Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_CLR)	32	RW	0000_0000h
CCh	Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_TOG)	32	RW	0000_0000h
E0h	Frame 2-Plane Control 0 (FRAME_2P_CTRL0)	32	RW	0000_0000h
E4h	Frame 2-Plane Control 0 (FRAME_2P_CTRL0_SET)	32	RW	0000_0000h
E8h	Frame 2-Plane Control 0 (FRAME_2P_CTRL0_CLR)	32	RW	0000_0000h
ECh	Frame 2-Plane Control 0 (FRAME_2P_CTRL0_TOG)	32	RW	0000_0000h
F0h	Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL)	32	RW	0000_0000h
F4h	Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_SET)	32	RW	0000_0000h
F8h	Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_CLR)	32	RW	0000_0000h
FCh	Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_TOG)	32	RW	0000_0000h
100h	Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL)	32	RW	0000_0000h
104h	Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_SET)	32	RW	0000_0000h
108h	Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_CLR)	32	RW	0000_0000h
10Ch	Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_TOG)	32	RW	0000_0000h
110h	Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0)	32	RW	0000_0000h
114h	Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_SET)	32	RW	0000_0000h
118h	Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_CLR)	32	RW	0000_0000h
11Ch	Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_TOG)	32	RW	0000_0000h
200h	RTRAM Control 0 (RTRAM_CTRL0)	32	RW	0000_0000h
204h	RTRAM Control 0 (RTRAM_CTRL0_SET)	32	RW	0000_0000h
208h	RTRAM Control 0 (RTRAM_CTRL0_CLR)	32	RW	0000_0000h
20Ch	RTRAM Control 0 (RTRAM_CTRL0_TOG)	32	RW	0000_0000h

15.7.3.1.2 System Control 0 (SYSTEM_CTRL0)

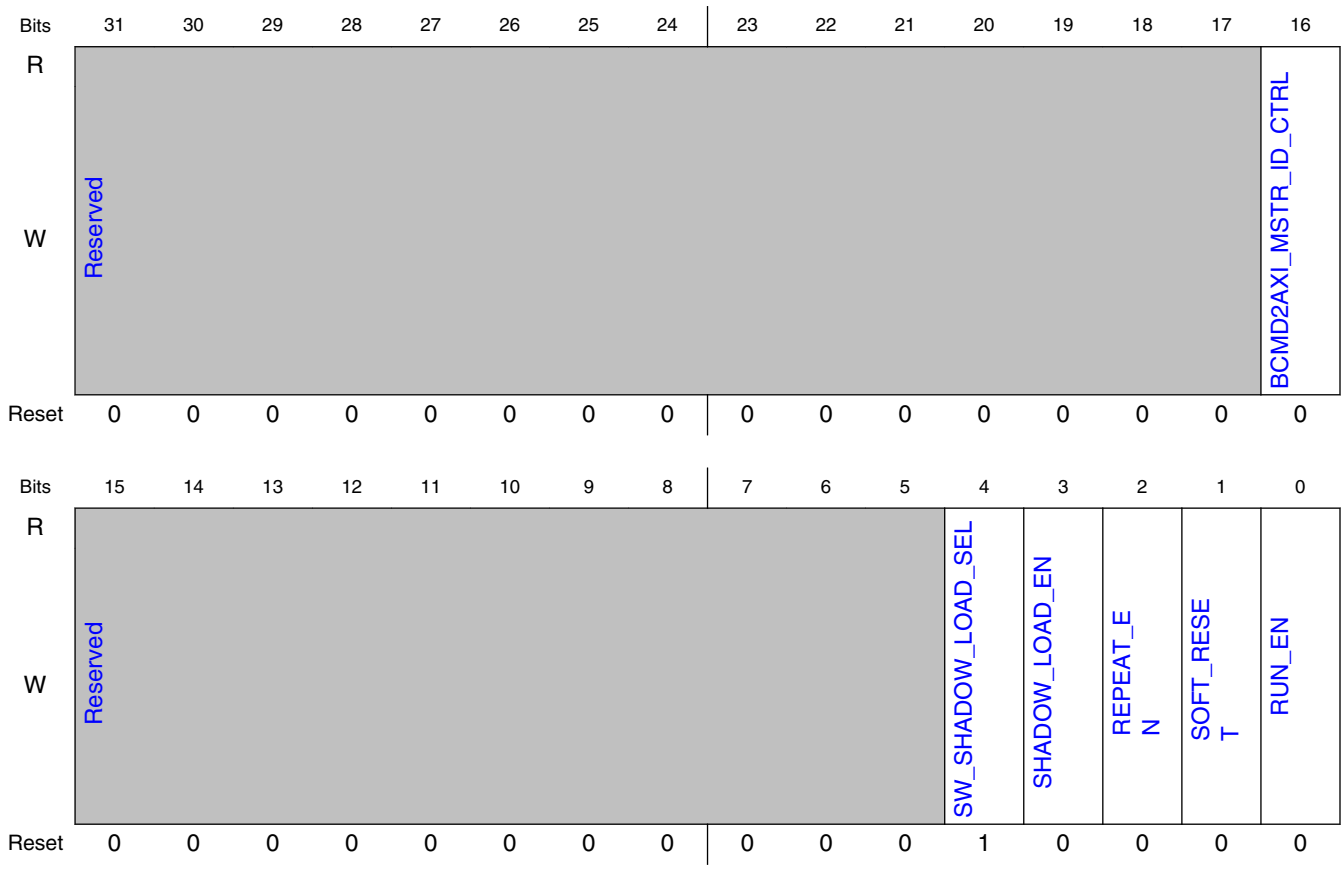
15.7.3.1.2.1 Offset

Register	Offset
SYSTEM_CTRL0	0h

15.7.3.1.2.2 Function

The System Control0 register contains register for DPR block level control functions.

15.7.3.1.2.3 Diagram



15.7.3.1.2.4 Fields

Field	Function
31-17 —	Reserved.
16 BCMD2AXI_MSTR_ID_CTRL	Buscmd To AXI Master ID Control 0x0 = All DPR buscmd master channels assigned a unique AXI ID 0x1 = All DPR buscmd master channels assigned the same AXI ID Note: If the DPR is configured to have > 1 DPR channel, then DPR channel 0, will control the BCMD2AXI_MSTR_ID_CTRL for all DPR Channels. DPR Channel 1-N's BCMD2AXI_MSTR_ID_CTRL will be reserved.
15-5 —	Reserved.
4 SW_SHADOW_LOAD_SEL	Software Shadow Load Select The SW_SHADOW_LOAD_SEL is used to select the shadow_load_en source.

Table continues on the next page...

Field	Function
	If set to 1, SYSTEM_CTRL0: SHADOW_LOAD_EN control register is used to allow update of the shadow control registers to the active control registers. If set to 0, Subystem hardware signal is used to allow update of the shadow control registers to the active control registers. SW_SHADOW_LOAD_SEL should be set to 1, for initial DPR control register load and RUN_EN enablement. SW_SHADOW_LOAD_SEL should only be set to 0, after SYSTEM_CTRL0: REPEAT_EN is enabled.
3 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all control registers are written. If set to 1, shadowed control registers are updated to the active control registers on next REPEAT_EN or RUN_EN to start DPR processing of the next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.
2 REPEAT_EN	Repeat Enable Writing a 1 to this register will enable the timing generator to write the RUN_EN register on subsequent restarts using the current control register settings. The REPEAT_EN will not take effect until after the 1st frame has been processed with RUN_EN issued by software. The REPEAT_EN may remain enabled for as many concurrent frames as desired to be automatically restarted. When automatic restart should be disabled, software must clear the REPEAT_EN before the next available hardware restart command.
1 SOFT_RESET	Soft Reset This bit is not self-clearing software will need to write a: 0x0 = deassert soft reset 0x1 = assert soft reset Note: Channel 0 soft reset will reset the SYSTEM_CTRL0:BCMD2AXI_MSTR_ID_CTRL. This should only be done if all DPR Channels are done processing.
0 RUN_EN	Run Enable Writing a 1 to this register will start processing the next frame image. This register is self-clearing.

15.7.3.1.3 System Control 0 (SYSTEM_CTRL0_SET)

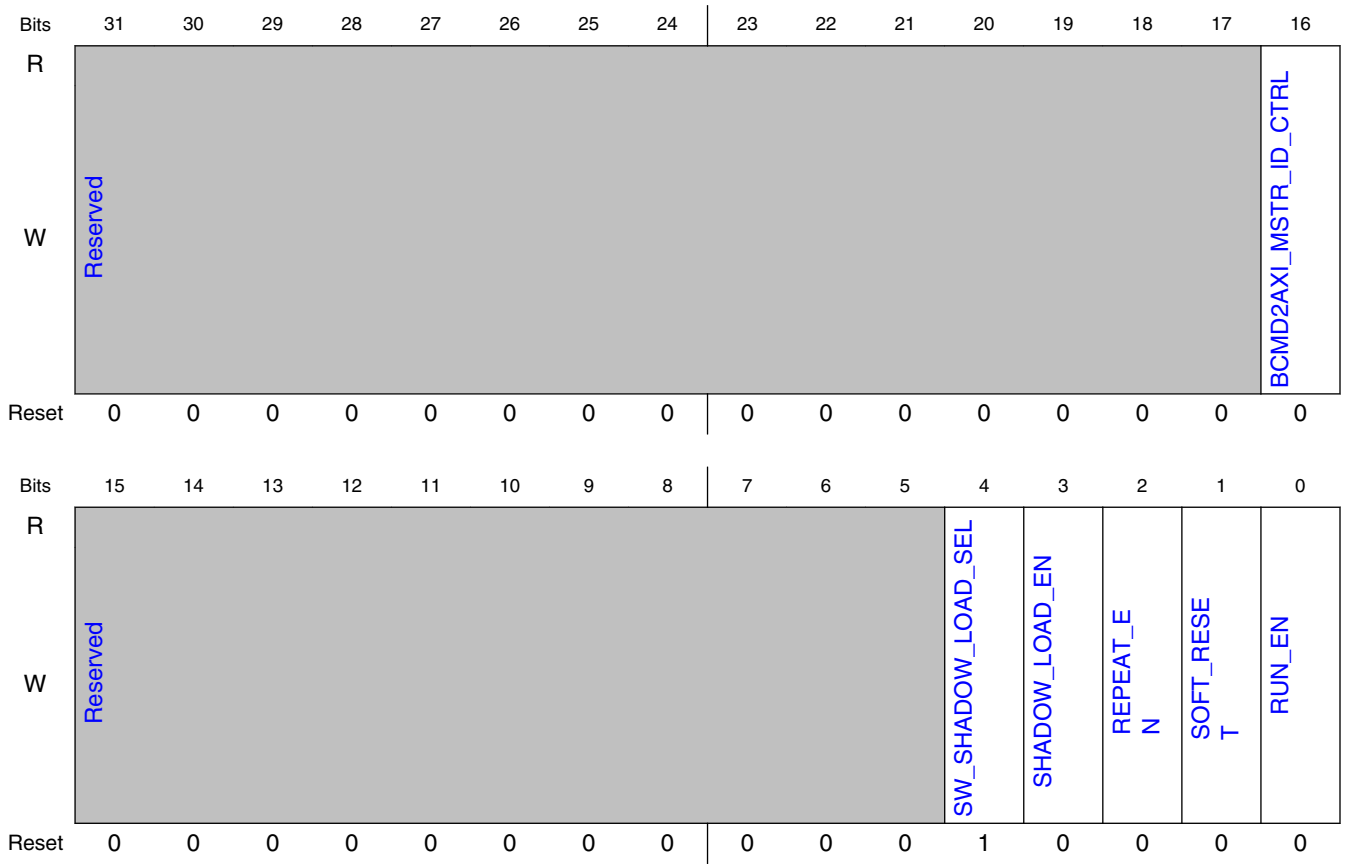
15.7.3.1.3.1 Offset

Register	Offset
SYSTEM_CTRL0_SET	4h

15.7.3.1.3.2 Function

The System Control0 register contains register for DPR block level control functions.

15.7.3.1.3.3 Diagram



15.7.3.1.3.4 Fields

Field	Function
31-17 —	Reserved.
16 BCMD2AXI_MSTR_ID_CTRL	Buscmd To AXI Master ID Control 0x0 = All DPR buscmd master channels assigned a unique AXI ID 0x1 = All DPR buscmd master channels assigned the same AXI ID Note: If the DPR is configured to have > 1 DPR channel, then DPR channel 0, will control the BCMD2AXI_MSTR_ID_CTRL for all DPR Channels. DPR Channel 1-N's BCMD2AXI_MSTR_ID_CTRL will be reserved.
15-5	Reserved.

Table continues on the next page...

Field	Function
—	
4 SW_SHADOW_LOAD_SEL	Software Shadow Load Select The SW_SHADOW_LOAD_SEL is used to select the shadow_load_en source. If set to 1, SYSTEM_CTRL0: SHADOW_LOAD_EN control register is used to allow update of the shadow control registers to the active control registers. If set to 0, Subystem hardware signal is used to allow update of the shadow control registers to the active control registers. SW_SHADOW_LOAD_SEL should be set to 1, for initial DPR control register load and RUN_EN enablement. SW_SHADOW_LOAD_SEL should only be set to 0, after SYSTEM_CTRL0: REPEAT_EN is enabled.
3 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all control registers are written. If set to 1, shadowed control registers are updated to the active control registers on next REPEAT_EN or RUN_EN to start DPR processing of the next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.
2 REPEAT_EN	Repeat Enable Writing a 1 to this register will enable the timing generator to write the RUN_EN register on subsequent restarts using the current control register settings. The REPEAT_EN will not take effect until after the 1st frame has been processed with RUN_EN issued by software. The REPEAT_EN may remain enabled for as many concurrent frames as desired to be automatically restarted. When automatic restart should be disabled, software must clear the REPEAT_EN before the next available hardware restart command.
1 SOFT_RESET	Soft Reset This bit is not self-clearing software will need to write a: 0x0 = deassert soft reset 0x1 = assert soft reset Note: Channel 0 soft reset will reset the SYSTEM_CTRL0:BCMD2AXI_MSTR_ID_CTRL. This should only be done if all DPR Channels are done processing.
0 RUN_EN	Run Enable Writing a 1 to this register will start processing the next frame image. This register is self-clearing.

15.7.3.1.4 System Control 0 (SYSTEM_CTRL0_CLR)

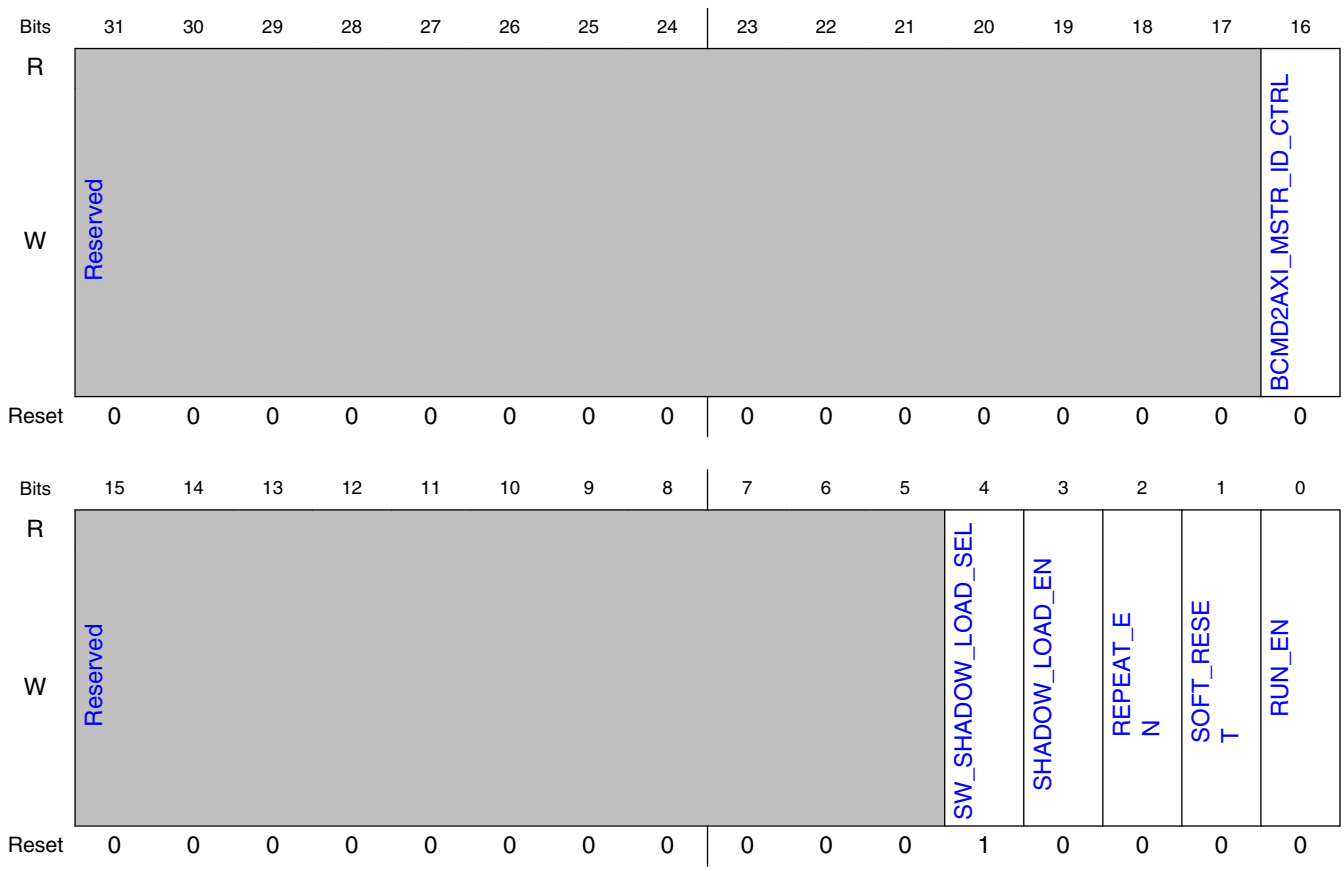
15.7.3.1.4.1 Offset

Register	Offset
SYSTEM_CTRL0_CLR	8h

15.7.3.1.4.2 Function

The System Control0 register contains register for DPR block level control functions.

15.7.3.1.4.3 Diagram



15.7.3.1.4.4 Fields

Field	Function
31-17 —	Reserved.
16	Buscmd To AXI Master ID Control 0x0 = All DPR buscmd master channels assigned a unique AXI ID

Table continues on the next page...

Field	Function
BCMD2AXI_MSTR_ID_CTRL	0x1 = All DPR buscmd master channels assigned the same AXI ID Note: If the DPR is configured to have > 1 DPR channel, then DPR channel 0, will control the BCMD2AXI_MSTR_ID_CTRL for all DPR Channels. DPR Channel 1-N's BCMD2AXI_MSTR_ID_CTRL will be reserved.
15-5 —	Reserved.
4 SW_SHADOW_LOAD_SEL	Software Shadow Load Select The SW_SHADOW_LOAD_SEL is used to select the shadow_load_en source. If set to 1, SYSTEM_CTRL0: SHADOW_LOAD_EN control register is used to allow update of the shadow control registers to the active control registers. If set to 0, Subystem hardware signal is used to allow update of the shadow control registers to the active control registers. SW_SHADOW_LOAD_SEL should be set to 1, for initial DPR control register load and RUN_EN enablement. SW_SHADOW_LOAD_SEL should only be set to 0, after SYSTEM_CTRL0: REPEAT_EN is enabled.
3 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all control registers are written. If set to 1, shadowed control registers are updated to the active control registers on next REPEAT_EN or RUN_EN to start DPR processing of the next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.
2 REPEAT_EN	Repeat Enable Writing a 1 to this register will enable the timing generator to write the RUN_EN register on subsequent restarts using the current control register settings. The REPEAT_EN will not take effect until after the 1st frame has been processed with RUN_EN issued by software. The REPEAT_EN may remain enabled for as many concurrent frames as desired to be automatically restarted. When automatic restart should be disabled, software must clear the REPEAT_EN before the next available hardware restart command.
1 SOFT_RESET	Soft Reset This bit is not self-clearing software will need to write a: 0x0 = deassert soft reset 0x1 = assert soft reset Note: Channel 0 soft reset will reset the SYSTEM_CTRL0:BCMD2AXI_MSTR_ID_CTRL. This should only be done if all DPR Channels are done processing.
0 RUN_EN	Run Enable Writing a 1 to this register will start processing the next frame image. This register is self-clearing.

15.7.3.1.5 System Control 0 (SYSTEM_CTRL0_TOG)

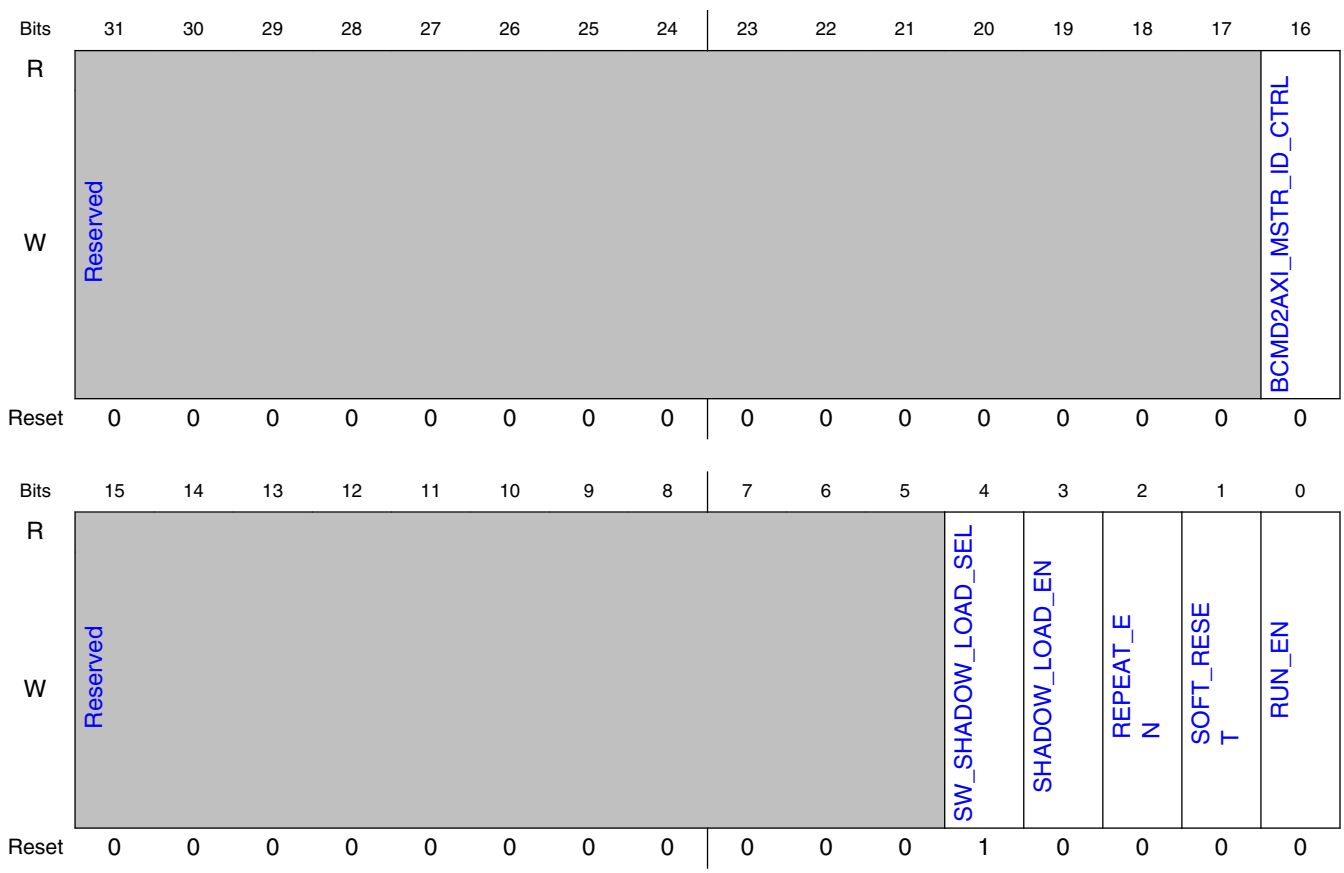
15.7.3.1.5.1 Offset

Register	Offset
SYSTEM_CTRL0_TOG	Ch

15.7.3.1.5.2 Function

The System Control0 register contains register for DPR block level control functions.

15.7.3.1.5.3 Diagram



15.7.3.1.5.4 Fields

Field	Function
31-17	Reserved.

Table continues on the next page...

Field	Function
—	
16 BCMD2AXI_MSTR_ID_CTRL	Buscmd To AXI Master ID Control 0x0 = All DPR buscmd master channels assigned a unique AXI ID 0x1 = All DPR buscmd master channels assigned the same AXI ID Note: If the DPR is configured to have > 1 DPR channel, then DPR channel 0, will control the BCMD2AXI_MSTR_ID_CTRL for all DPR Channels. DPR Channel 1-N's BCMD2AXI_MSTR_ID_CTRL will be reserved.
15-5 —	Reserved.
4 SW_SHADOW_LOAD_SEL	Software Shadow Load Select The SW_SHADOW_LOAD_SEL is used to select the shadow_load_en source. If set to 1, SYSTEM_CTRL0: SHADOW_LOAD_EN control register is used to allow update of the shadow control registers to the active control registers. If set to 0, Subystem hardware signal is used to allow update of the shadow control registers to the active control registers. SW_SHADOW_LOAD_SEL should be set to 1, for initial DPR control register load and RUN_EN enablement. SW_SHADOW_LOAD_SEL should only be set to 0, after SYSTEM_CTRL0: REPEAT_EN is enabled.
3 SHADOW_LOAD_EN	Shadow Load Enable The SHADOW_LOAD_EN bit is written to 1 by software after all control registers are written. If set to 1, shadowed control registers are updated to the active control registers on next REPEAT_EN or RUN_EN to start DPR processing of the next frame. If set to 0, shadowed control registers are not loaded into the active control registers. The previous active control register settings will be used to process the next frame. Hardware will automatically clear this bit, when the shadow registers are loaded to the active control regsisters.
2 REPEAT_EN	Repeat Enable Writing a 1 to this register will enable the timing generator to write the RUN_EN register on subsequent restarts using the current control register settings. The REPEAT_EN will not take effect until after the 1st frame has been processed with RUN_EN issued by software. The REPEAT_EN may remain enabled for as many concurrent frames as desired to be automatically restarted. When automatic restart should be disabled, software must clear the REPEAT_EN before the next available hardware restart command.
1 SOFT_RESET	Soft Reset This bit is not self-clearing software will need to write a: 0x0 = deassert soft reset 0x1 = assert soft reset Note: Channel 0 soft reset will reset the SYSTEM_CTRL0:BCMD2AXI_MSTR_ID_CTRL. This should only be done if all DPR Channels are done processing.
0 RUN_EN	Run Enable

Field	Function
	Writing a 1 to this register will start processing the next frame image. This register is self-clearing.

15.7.3.1.6 Interrupt Mask (IRQ_MASK)

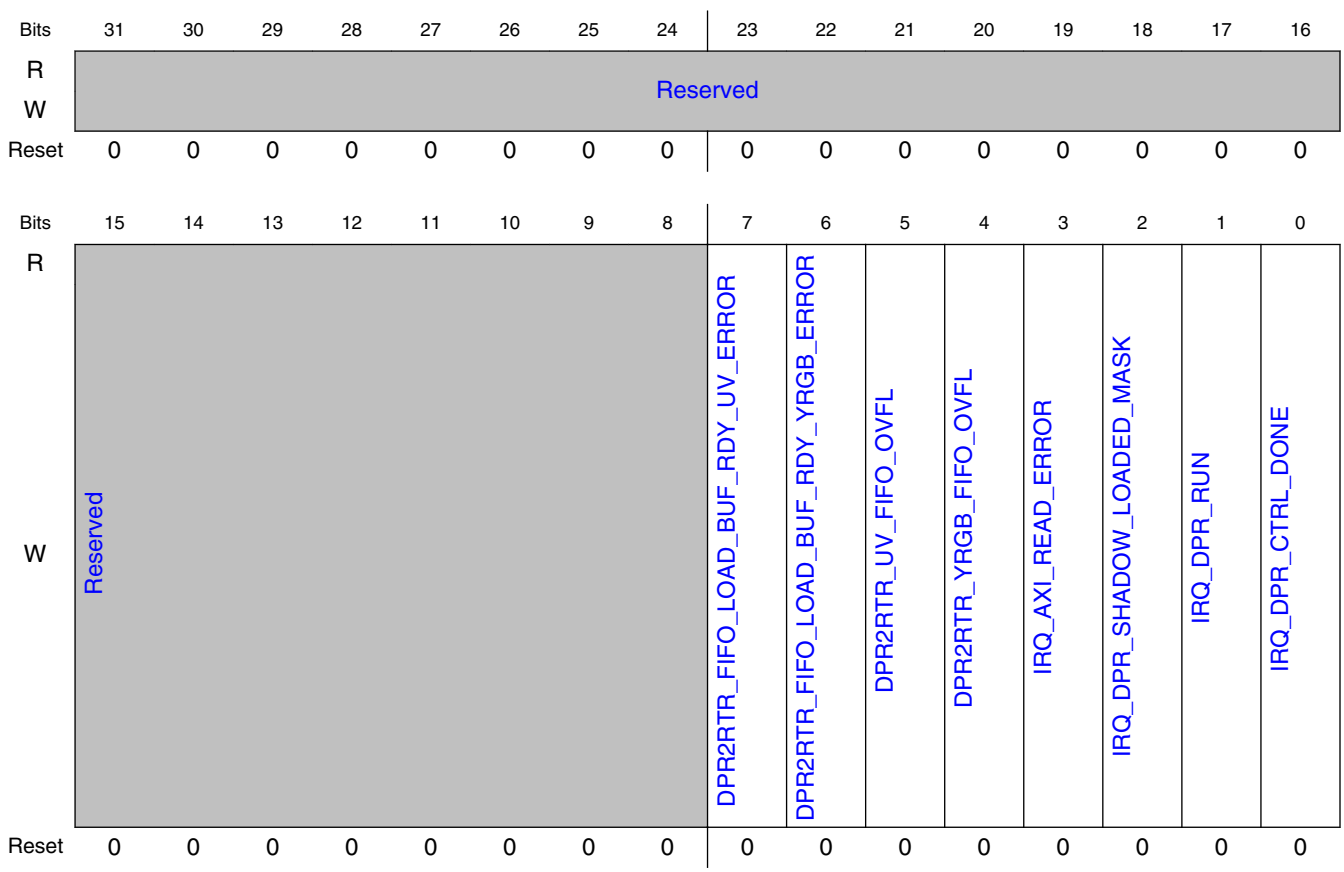
15.7.3.1.6.1 Offset

Register	Offset
IRQ_MASK	20h

15.7.3.1.6.2 Function

The IRQ_MASK control register contains interrupt mask control register bits.
Writing a 1 to the IRQ_MASK control register will mask off the interrupt.

15.7.3.1.6.3 Diagram



15.7.3.1.6.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow IRQ Mask Mask off the DPR2RTR_UV Fifo Overflow interrupt.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow IRQ Mask Mask off the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error IRQ Mask Mask off the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED_MASK	DPR Shadow Loaded IRQ Mask Mask off the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run IRQ Mask Mask off the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done IRQ Mask Mask off the DPR_CTRL_DONE interrupt.

15.7.3.1.7 Interrupt Mask (IRQ_MASK_SET)

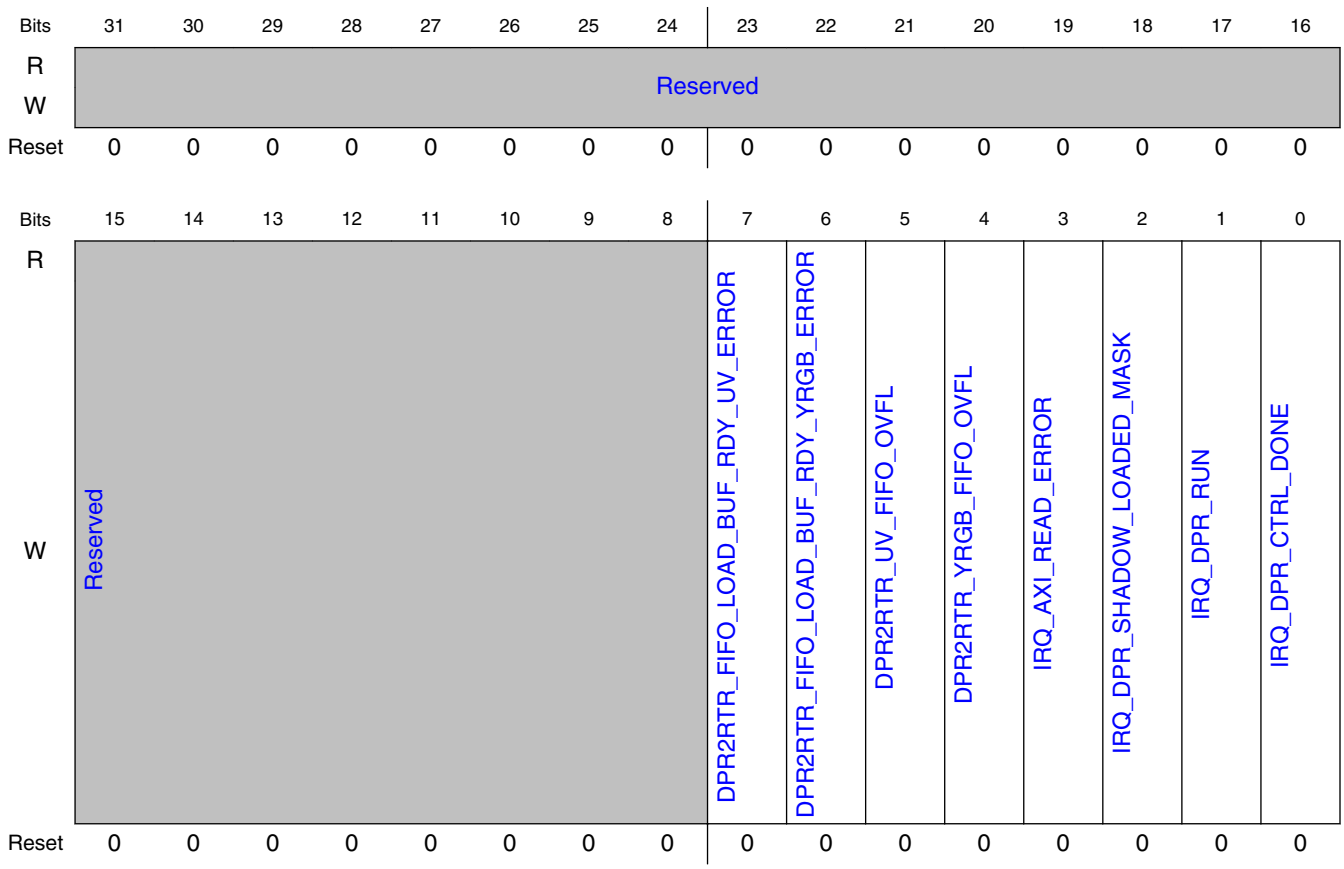
15.7.3.1.7.1 Offset

Register	Offset
IRQ_MASK_SET	24h

15.7.3.1.7.2 Function

```
The IRQ_MASK control register contains interrupt mask control register bits.
Writing a 1 to the IRQ_MASK control register will mask off the interrupt.
```

15.7.3.1.7.3 Diagram



15.7.3.1.7.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6	DPR to RTRAM Fifo load YRGB buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.

Table continues on the next page...

Field	Function
DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow IRQ Mask Mask off the DPR2RTR_UV Fifo Overflow interrupt.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow IRQ Mask Mask off the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error IRQ Mask Mask off the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED_MASK	DPR Shadow Loaded IRQ Mask Mask off the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run IRQ Mask Mask off the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done IRQ Mask Mask off the DPR_CTRL_DONE interrupt.

15.7.3.1.8 Interrupt Mask (IRQ_MASK_CLR)

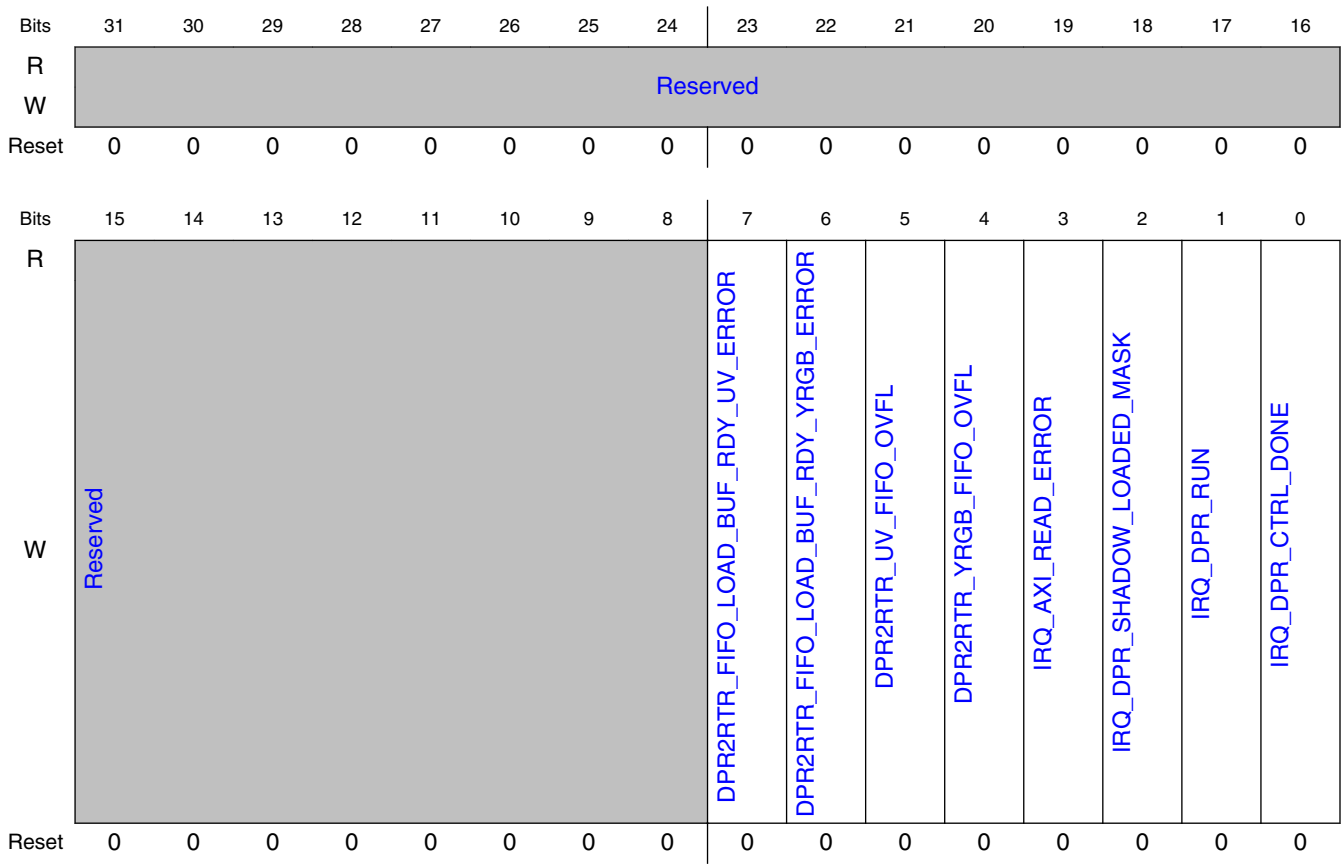
15.7.3.1.8.1 Offset

Register	Offset
IRQ_MASK_CLR	28h

15.7.3.1.8.2 Function

```
The IRQ_MASK control register contains interrupt mask control register bits.
Writing a 1 to the IRQ_MASK control register will mask off the interrupt.
```

15.7.3.1.8.3 Diagram



15.7.3.1.8.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow IRQ Mask Mask off the DPR2RTR_UV Fifo Overflow interrupt.
4	DPR to RTRAM YRGB Fifo Overflow IRQ Mask

Table continues on the next page...

Field	Function
DPR2RTR_YRGB_FIFO_OVFL	Mask off the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error IRQ Mask Mask off the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED_MASK	DPR Shadow Loaded IRQ Mask Mask off the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run IRQ Mask Mask off the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done IRQ Mask Mask off the DPR_CTRL_DONE interrupt.

15.7.3.1.9 Interrupt Mask (IRQ_MASK_TOG)

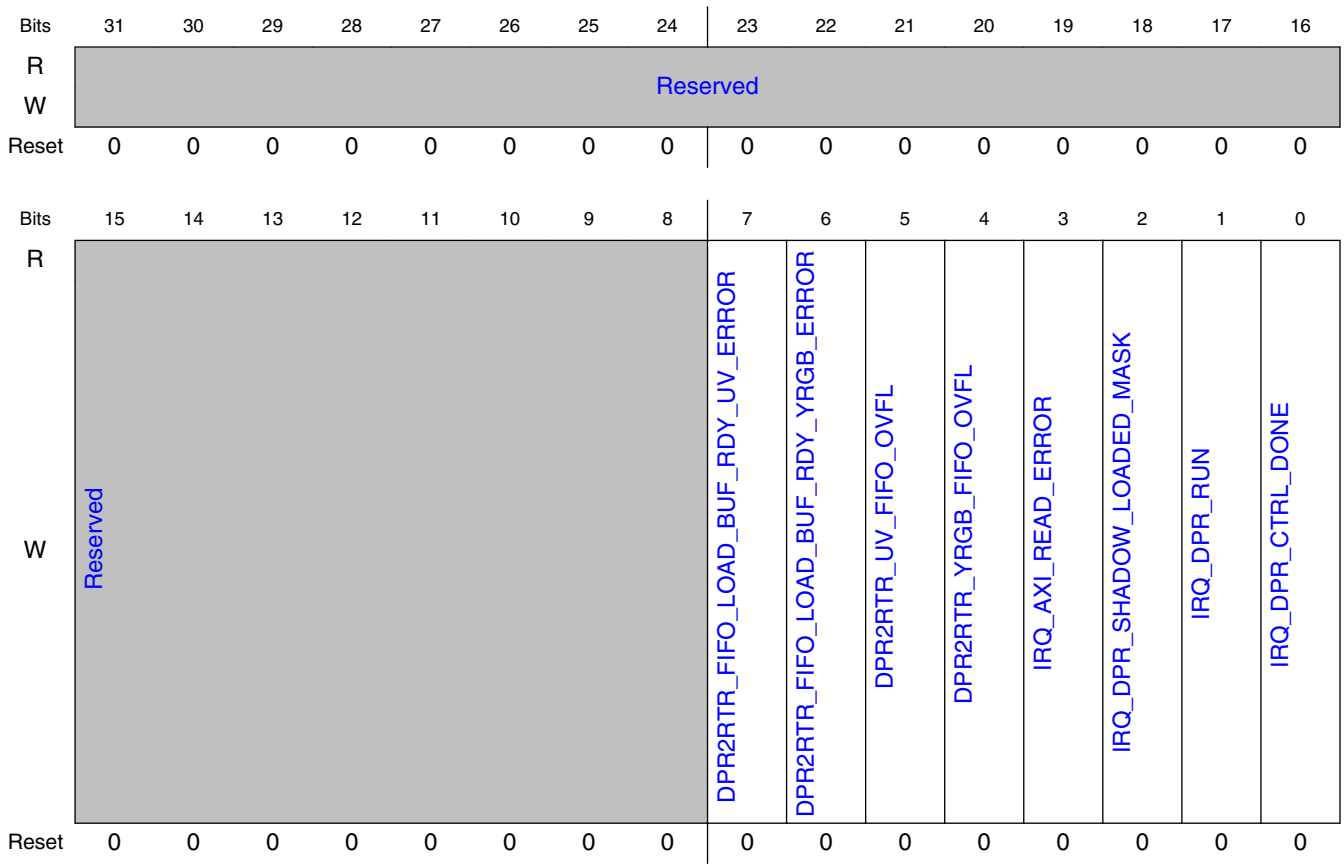
15.7.3.1.9.1 Offset

Register	Offset
IRQ_MASK_TOG	2Ch

15.7.3.1.9.2 Function

```
The IRQ_MASK control register contains interrupt mask control register bits.
Writing a 1 to the IRQ_MASK control register will mask off the interrupt.
```

15.7.3.1.9.3 Diagram



15.7.3.1.9.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer ready IRQ error Mask Mask off the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow IRQ Mask Mask off the DPR2RTR_UV Fifo Overflow interrupt.
4	DPR to RTRAM YRGB Fifo Overflow IRQ Mask

Table continues on the next page...

Field	Function
DPR2RTR_YRGB_FIFO_OVFL	Mask off the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error IRQ Mask Mask off the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED_MASK	DPR Shadow Loaded IRQ Mask Mask off the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run IRQ Mask Mask off the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done IRQ Mask Mask off the DPR_CTRL_DONE interrupt.

15.7.3.1.10 Status Register of Masked IRQ (IRQ_MASK_STATUS)

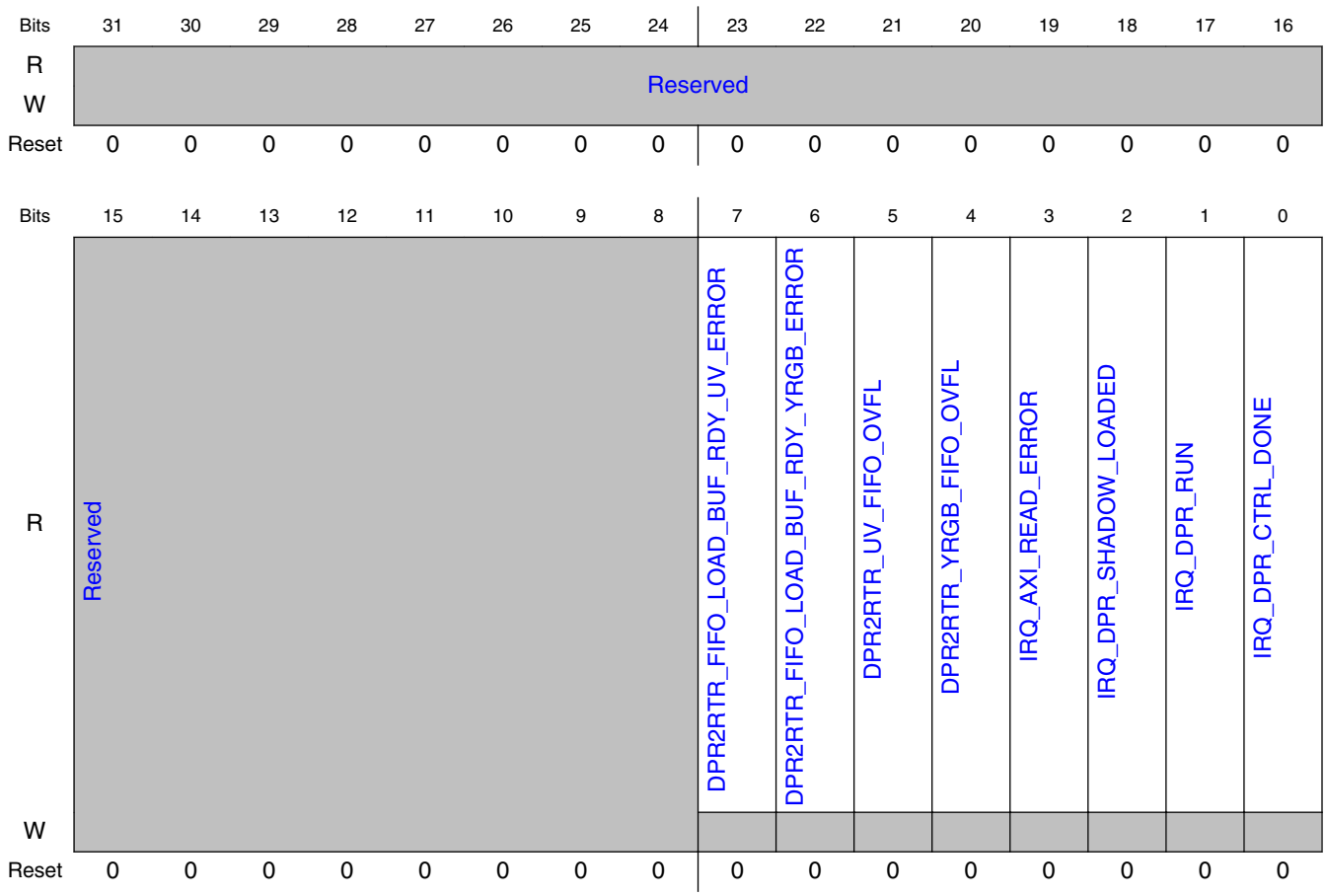
15.7.3.1.10.1 Offset

Register	Offset
IRQ_MASK_STATUS	30h

15.7.3.1.10.2 Function

The IRQ_MASK_STATUS control register contains the masked interrupt status.

15.7.3.1.10.3 Diagram



15.7.3.1.10.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt.

Table continues on the next page...

Field	Function
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error Masked IRQ Masked interrupt status of the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED	DPR Shadow Loaded Masked IRQ Masked interrupt status of the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run Masked IRQ Masked interrupt status of the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Masked IRQ Masked interrupt status of the DPR_CTRL_DONE interrupt.

15.7.3.1.11 Status Register of Masked IRQ (IRQ_MASK_STATUS_SET)

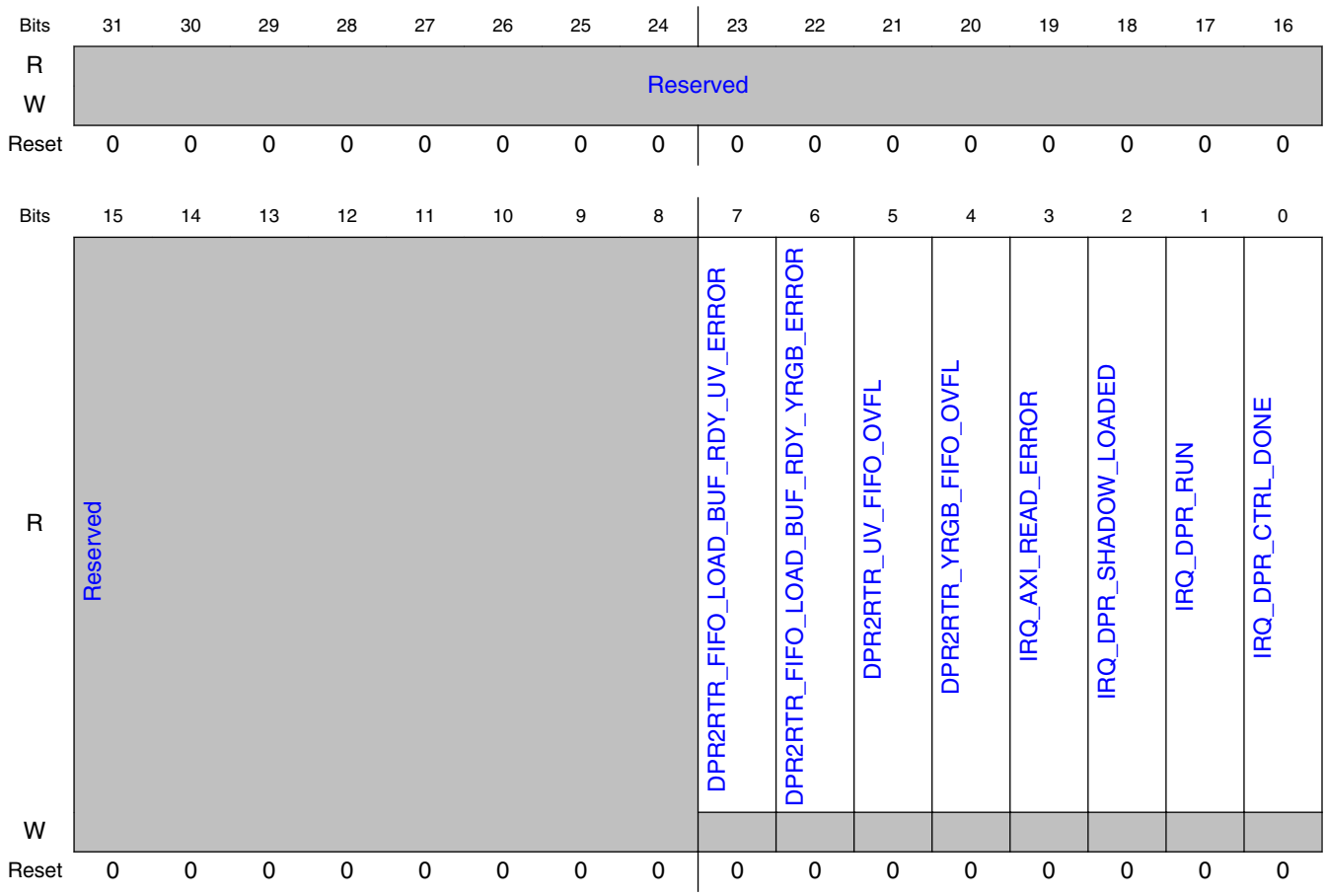
15.7.3.1.11.1 Offset

Register	Offset
IRQ_MASK_STATUS_SET	34h

15.7.3.1.11.2 Function

The IRQ_MASK_STATUS control register contains the masked interrupt status.

15.7.3.1.11.3 Diagram



15.7.3.1.11.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt.

Table continues on the next page...

Field	Function
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error Masked IRQ Masked interrupt status of the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED	DPR Shadow Loaded Masked IRQ Masked interrupt status of the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run Masked IRQ Masked interrupt status of the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Masked IRQ Masked interrupt status of the DPR_CTRL_DONE interrupt.

15.7.3.1.12 Status Register of Masked IRQ (IRQ_MASK_STATUS_CLR)

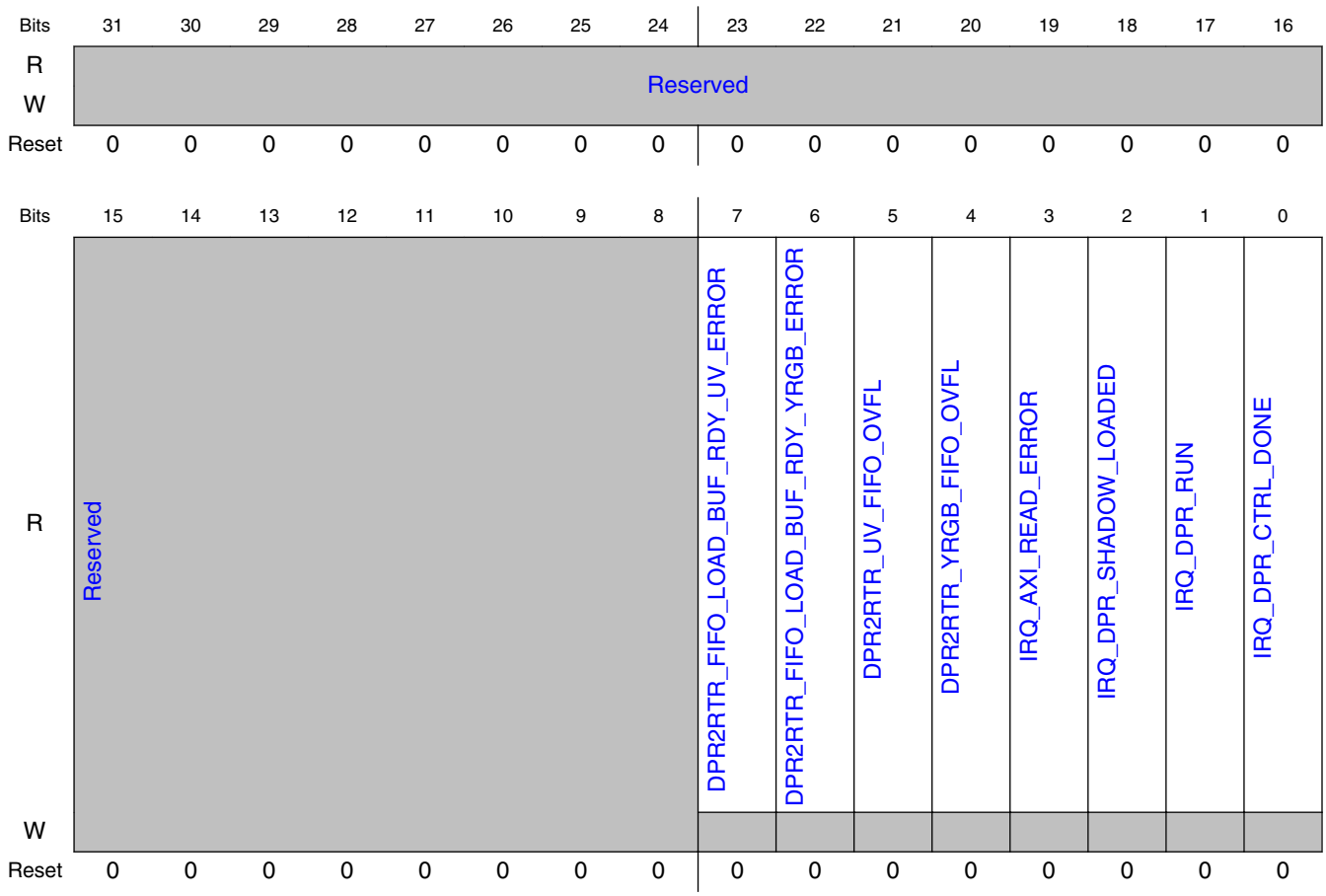
15.7.3.1.12.1 Offset

Register	Offset
IRQ_MASK_STATUS_CLR	38h

15.7.3.1.12.2 Function

The IRQ_MASK_STATUS control register contains the masked interrupt status.

15.7.3.1.12.3 Diagram



15.7.3.1.12.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt.

Table continues on the next page...

Field	Function
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error Masked IRQ Masked interrupt status of the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED	DPR Shadow Loaded Masked IRQ Masked interrupt status of the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run Masked IRQ Masked interrupt status of the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Masked IRQ Masked interrupt status of the DPR_CTRL_DONE interrupt.

15.7.3.1.13 Status Register of Masked IRQ (IRQ_MASK_STATUS_TOG)

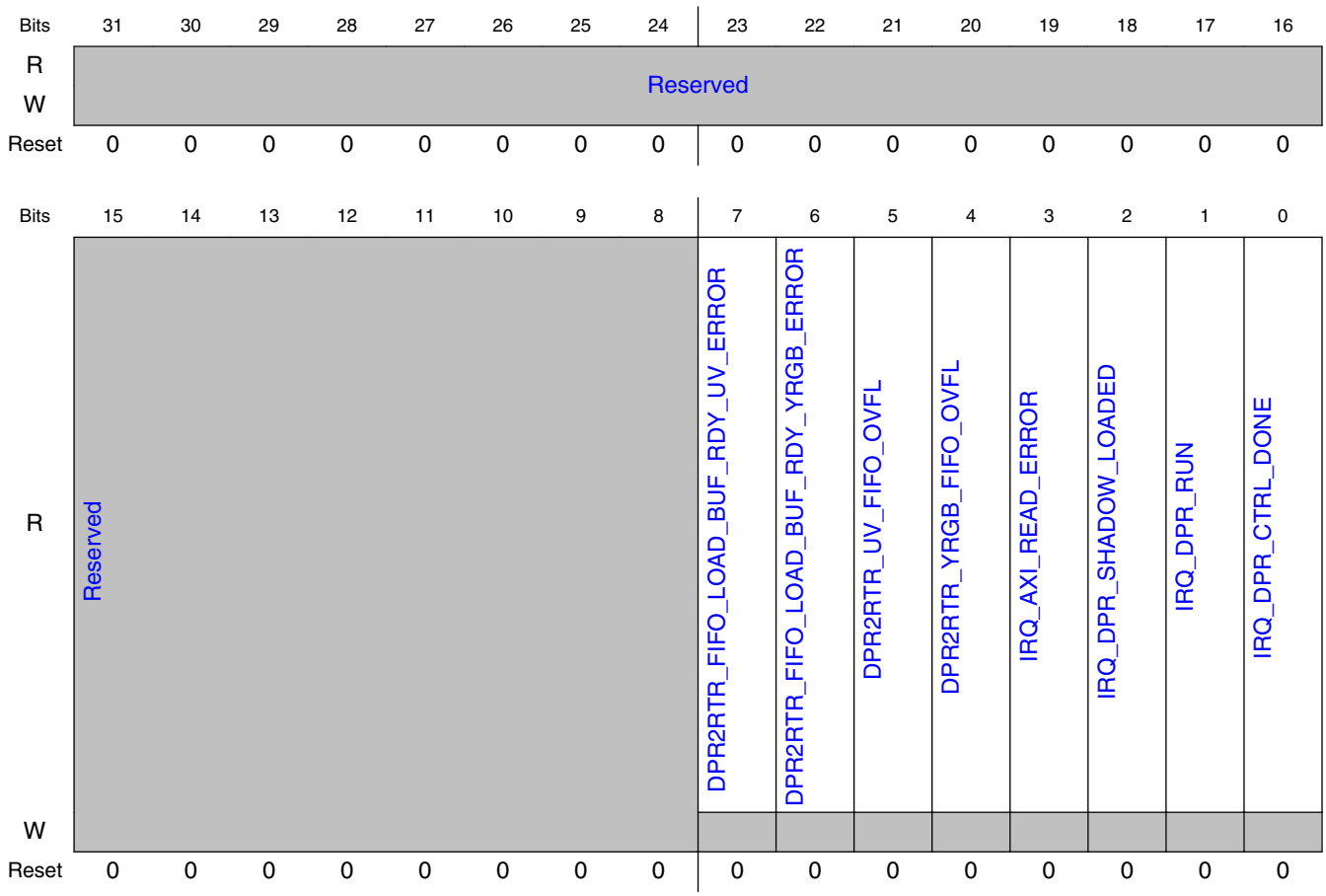
15.7.3.1.13.1 Offset

Register	Offset
IRQ_MASK_STATUS_TOG	3Ch

15.7.3.1.13.2 Function

The IRQ_MASK_STATUS control register contains the masked interrupt status.

15.7.3.1.13.3 Diagram



15.7.3.1.13.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer error Masked IRQ Masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt.

Table continues on the next page...

Field	Function
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Masked IRQ Masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt.
3 IRQ_AXI_READ_ERROR	AXI Read Error Masked IRQ Masked interrupt status of the AXI Read Error interrupt.
2 IRQ_DPR_SHADOW_LOADED	DPR Shadow Loaded Masked IRQ Masked interrupt status of the DPR_SHADOW_LOADED interrupt.
1 IRQ_DPR_RUN	DPR Run Masked IRQ Masked interrupt status of the DPR_RUN interrupt.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Masked IRQ Masked interrupt status of the DPR_CTRL_DONE interrupt.

15.7.3.1.14 Status of Non-Masked IRQ (IRQ_NONMASK_STATUS)

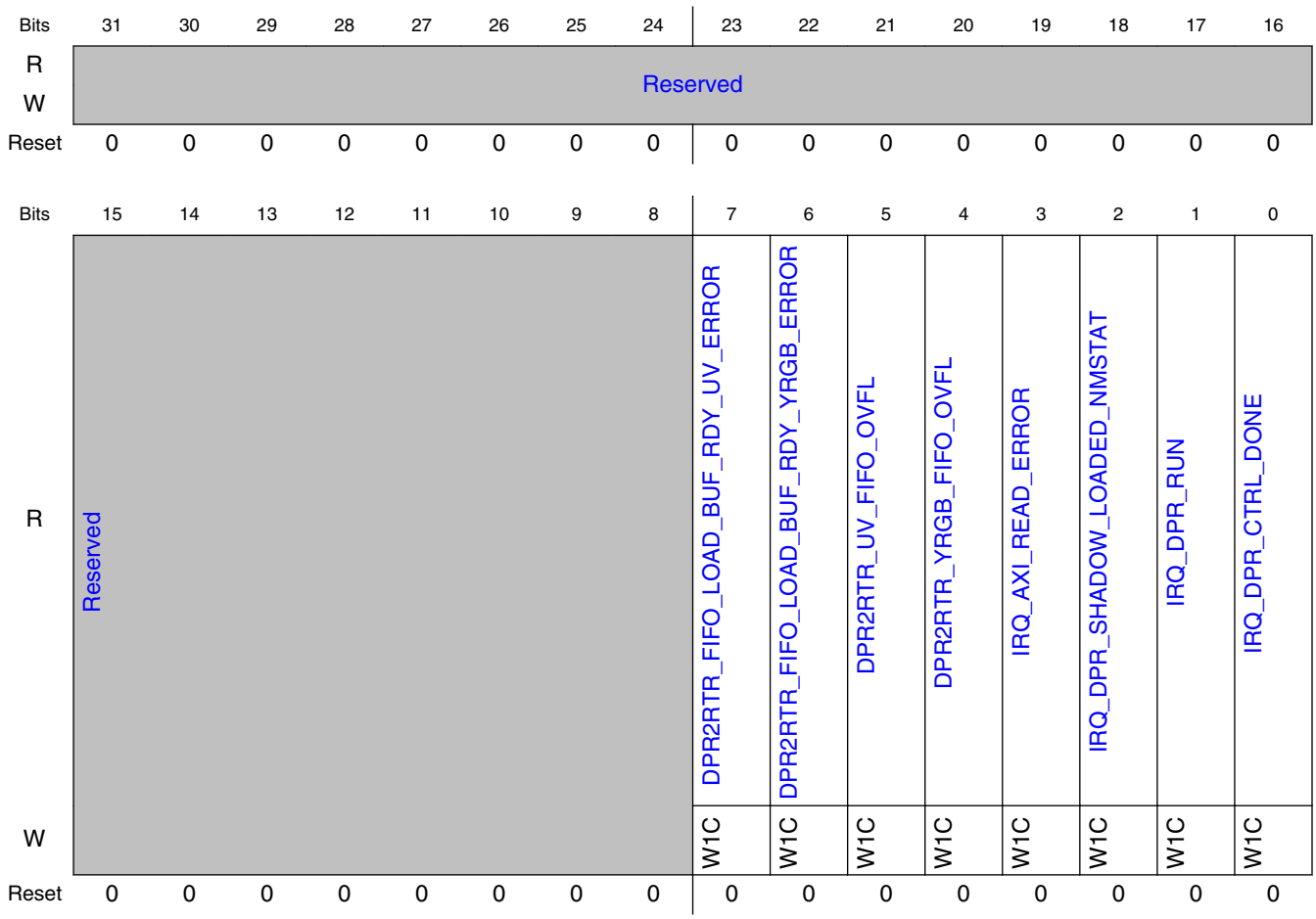
15.7.3.1.14.1 Offset

Register	Offset
IRQ_NONMASK_STATUS	40h

15.7.3.1.14.2 Function

```
The IRQ_NONMASK_STATUS control register contains the raw interrupt status before applying
the mask.
Writing a 1 to the SCT's CLEAR mask bit, will clear the associated interrupt.
```

15.7.3.1.14.3 Diagram



15.7.3.1.14.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt. When this IRQ is set, the DPR UV buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
6	DPR to RTRAM Fifo load YRGB buffer ready error Non-Masked IRQ

Table continues on the next page...

Field	Function
DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt. When this IRQ is set, the DPR YRGB buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt. When this IRQ is set, the DPR2RTR Fifo Overflowed writing 2PYUV420 UV data.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt. When this is set, the DPR2RTR Fifo Overflowed writing YRGB data.
3 IRQ_AXI_READ_ERROR	AXI Read Error Non-Masked IRQ Non-masked interrupt status of the AXI Read Error interrupt. When this IRQ is set, the AXI Read Master received a bus transaction error code.
2 IRQ_DPR_SHADOW_LOADED_NMSTAT	DPR Shadow Loaded Non-Masked IRQ Non-masked interrupt status of the DPR_SHADOW_LOADED interrupt. When this IRQ IS set, the dpr_ctrl shadow control registers have been loaded into the active control registers.
1 IRQ_DPR_RUN	DPR Run Non-Masked IRQ Non-masked interrupt status of the DPR_RUN interrupt. When this IRQ IS set, the dpr_ctrl RUN_EN has been asserted in the control registrer clock domain. All control registers are considered active at this time.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Non-Masked IRQ Non-masked interrupt status of the DPR_CTRL_DONE interrupt. When this IRQ is set, the dpr_ctrl_prefetch and dpr_ctrl_response blocks are done processing the current frame.

15.7.3.1.15 Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_SET)

15.7.3.1.15.1 Offset

Register	Offset
IRQ_NONMASK_STATUS_SET	44h

15.7.3.1.15.2 Function

The IRQ_NONMASK_STATUS control register contains the raw interrupt status before applying the mask.
Writing a 1 to the SCT's CLEAR mask bit, will clear the associated interrupt.

15.7.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved								DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR2RTR_UV_FIFO_OVFL	DPR2RTR_YRGB_FIFO_OVFL	IRQ_AXI_READ_ERROR	IRQ_DPR_SHADOW_LOADED_NMSTAT	IRQ_DPR_RUN	IRQ_DPR_CTRL_DONE
W									W1C	W1C	W1C	W1C	W1C	W1C	W1C	W1C
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.15.4 Fields

Field	Function
31-8 —	Reserved.
7	DPR to RTRAM Fifo load UV buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt.

Table continues on the next page...

Field	Function
DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	When this IRQ is set, the DPR UV buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt. When this IRQ is set, the DPR YRGB buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt. When this IRQ is set, the DPR2RTR Fifo Overflowed writing 2PYUV420 UV data.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt. When this is set, the DPR2RTR Fifo Overflowed writing YRGB data.
3 IRQ_AXI_READ_ERROR	AXI Read Error Non-Masked IRQ Non-masked interrupt status of the AXI Read Error interrupt. When this IRQ is set, the AXI Read Master received a bus transaction error code.
2 IRQ_DPR_SHADOW_LOADED_NMSTAT	DPR Shadow Loaded Non-Masked IRQ Non-masked interrupt status of the DPR_SHADOW_LOADED interrupt. When this IRQ IS set, the dpr_ctrl shadow control registers have been loaded into the active control registers.
1 IRQ_DPR_RUN	DPR Run Non-Masked IRQ Non-masked interrupt status of the DPR_RUN interrupt. When this IRQ IS set, the dpr_ctrl RUN_EN has been asserted in the control registrer clock domain. All control registers are considered active at this time.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Non-Masked IRQ Non-masked interrupt status of the DPR_CTRL_DONE interrupt. When this IRQ is set, the dpr_ctrl_prefetch and dpr_ctrl_response blocks are done processing the current frame.

15.7.3.1.16 Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_CLR)

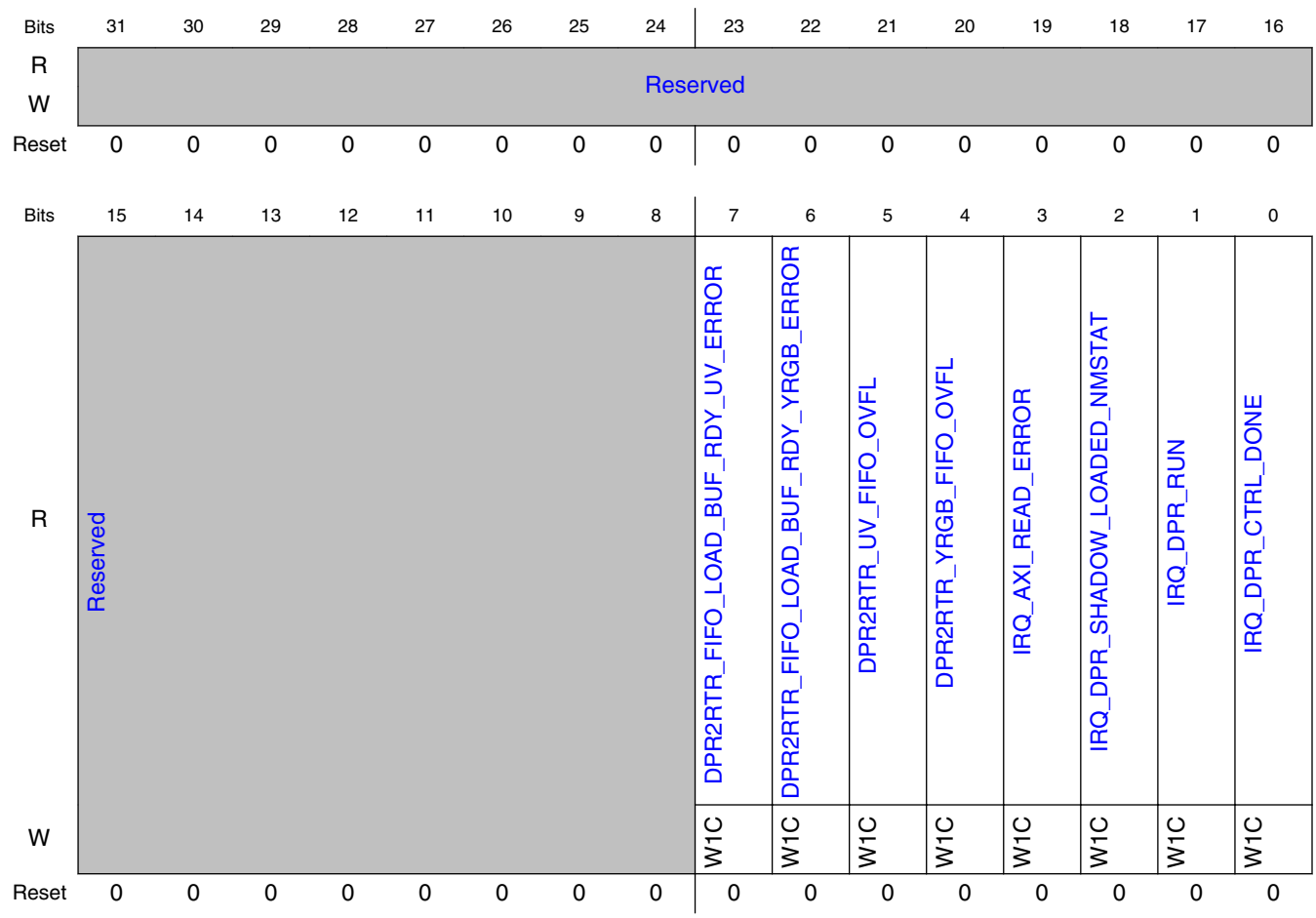
15.7.3.1.16.1 Offset

Register	Offset
IRQ_NONMASK_STATUS_CLR	48h

15.7.3.1.16.2 Function

The IRQ_NONMASK_STATUS control register contains the raw interrupt status before applying the mask.
Writing a 1 to the SCT's CLEAR mask bit, will clear the associated interrupt.

15.7.3.1.16.3 Diagram



15.7.3.1.16.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt. When this IRQ is set, the DPR UV buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
6 DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	DPR to RTRAM Fifo load YRGB buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt. When this IRQ is set, the DPR YRGB buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt. When this IRQ is set, the DPR2RTR Fifo Overflowed writing 2PYUV420 UV data.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt. When this is set, the DPR2RTR Fifo Overflowed writing YRGB data.
3 IRQ_AXI_READ_ERROR	AXI Read Error Non-Masked IRQ Non-masked interrupt status of the AXI Read Error interrupt. When this IRQ is set, the AXI Read Master received a bus transaction error code.
2 IRQ_DPR_SHADOW_LOADED_NMSTAT	DPR Shadow Loaded Non-Masked IRQ Non-masked interrupt status of the DPR_SHADOW_LOADED interrupt. When this IRQ IS set, the dpr_ctrl shadow control registers have been loaded into the active control registers.
1 IRQ_DPR_RUN	DPR Run Non-Masked IRQ Non-masked interrupt status of the DPR_RUN interrupt. When this IRQ IS set, the dpr_ctrl RUN_EN has been asserted in the control registrer clock domain. All control registers are considered active at this time.
0	DPR Control Done Non-Masked IRQ

Field	Function
IRQ_DPR_CTRL_DONE	Non-masked interrupt status of the DPR_CTRL_DONE interrupt. When this IRQ is set, the dpr_ctrl_prefetch and dpr_ctrl_response blocks are done processing the current frame.

15.7.3.1.17 Status of Non-Masked IRQ (IRQ_NONMASK_STATUS_TOG)

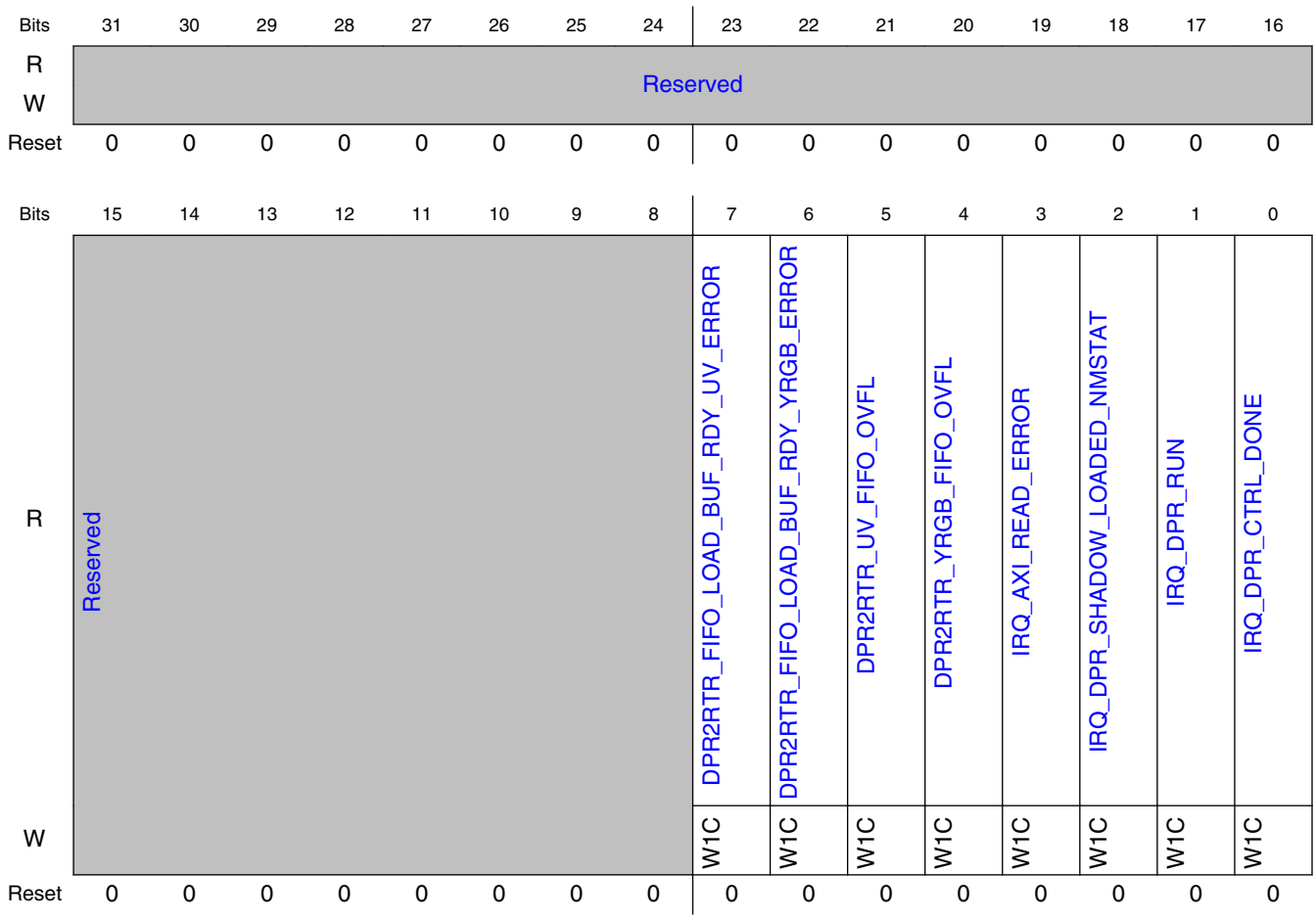
15.7.3.1.17.1 Offset

Register	Offset
IRQ_NONMASK_STATUS_TOG	4Ch

15.7.3.1.17.2 Function

The IRQ_NONMASK_STATUS control register contains the raw interrupt status before applying the mask.
Writing a 1 to the SCT's CLEAR mask bit, will clear the associated interrupt.

15.7.3.1.17.3 Diagram



15.7.3.1.17.4 Fields

Field	Function
31-8 —	Reserved.
7 DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR	DPR to RTRAM Fifo load UV buffer ready error Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_UV_ERROR interrupt. When this IRQ is set, the DPR UV buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
6	DPR to RTRAM Fifo load YRGB buffer ready error Non-Masked IRQ

Table continues on the next page...

Field	Function
DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR	Non-masked interrupt status of the DPR2RTR_FIFO_LOAD_BUF_RDY_YRGB_ERROR interrupt. When this IRQ is set, the DPR YRGB buffer ready was set during the 1st 32-bytes instead of the 2nd 32-bytes of the 64-byte payload. This error indicates the following: • pixel data will be loaded incorrectly into the RTRAM • The error is (most likely) due to incorrect programming of the NUM_X_PIX_WIDE value not being evenly divisible by 64-bytes.
5 DPR2RTR_UV_FIFO_OVFL	DPR to RTRAM UV Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_UV Fifo Overflow interrupt. When this IRQ is set, the DPR2RTR Fifo Overflowed writing 2PYUV420 UV data.
4 DPR2RTR_YRGB_FIFO_OVFL	DPR to RTRAM YRGB Fifo Overflow Non-Masked IRQ Non-masked interrupt status of the DPR2RTR_YRGB Fifo Overflow interrupt. When this is set, the DPR2RTR Fifo Overflowed writing YRGB data.
3 IRQ_AXI_READ_ERROR	AXI Read Error Non-Masked IRQ Non-masked interrupt status of the AXI Read Error interrupt. When this IRQ is set, the AXI Read Master received a bus transaction error code.
2 IRQ_DPR_SHADOW_LOADED_NMSTAT	DPR Shadow Loaded Non-Masked IRQ Non-masked interrupt status of the DPR_SHADOW_LOADED interrupt. When this IRQ IS set, the dpr_ctrl shadow control registers have been loaded into the active control registers.
1 IRQ_DPR_RUN	DPR Run Non-Masked IRQ Non-masked interrupt status of the DPR_RUN interrupt. When this IRQ IS set, the dpr_ctrl RUN_EN has been asserted in the control registrer clock domain. All control registers are considered active at this time.
0 IRQ_DPR_CTRL_DONE	DPR Control Done Non-Masked IRQ Non-masked interrupt status of the DPR_CTRL_DONE interrupt. When this IRQ is set, the dpr_ctrl_prefetch and dpr_ctrl_response blocks are done processing the current frame.

15.7.3.1.18 Mode Control 0 (MODE_CTRL0)

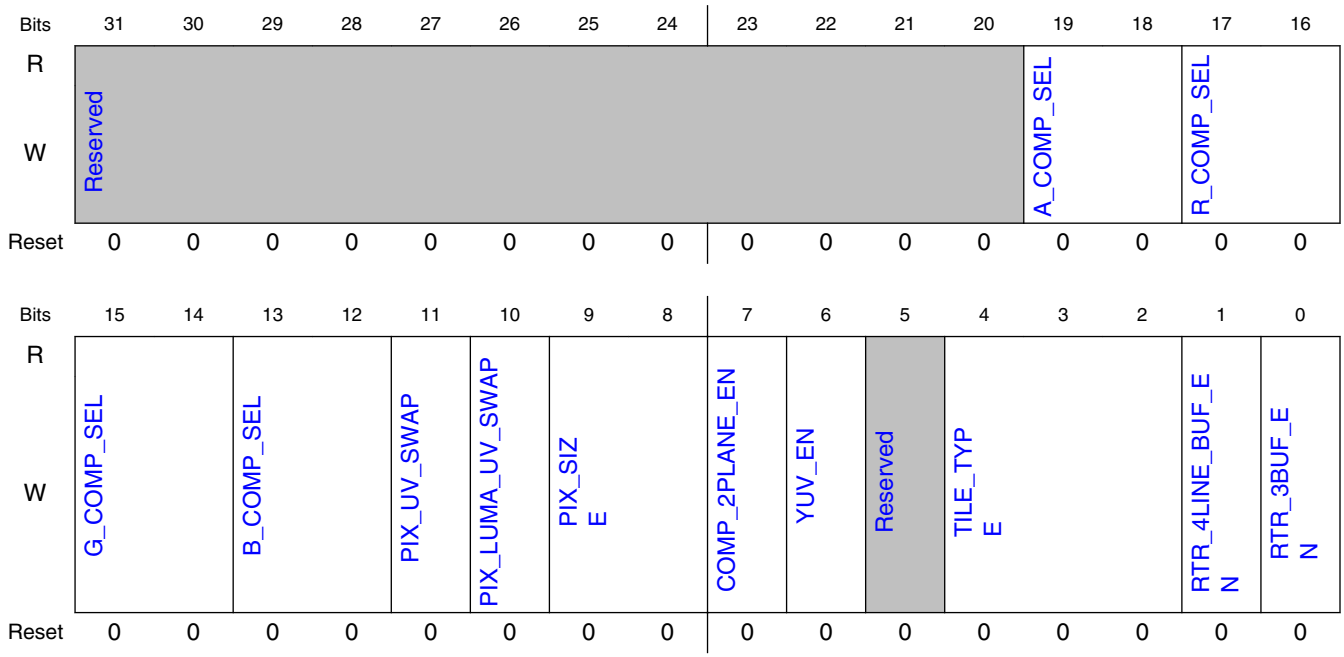
15.7.3.1.18.1 Offset

Register	Offset
MODE_CTRL0	50h

15.7.3.1.18.2 Function

The Mode Control0 register controls mode of operation configuration.

15.7.3.1.18.3 Diagram



15.7.3.1.18.4 Fields

Field	Function
31-20 —	Reserved.
19-18 A_COMP_SEL	A Component Select 32 bit per pixel, such as ARGB888, Alpha component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
17-16 R_COMP_SEL	R Component Select 32 bit per pixel, such as ARGB888, Red component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
15-14 G_COMP_SEL	G Component Select 32 bit per pixel, such as ARGB888, Green component byte select 0x0 = byte 0, bits[7:0]

Table continues on the next page...

Field	Function
	0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
13-12 B_COMP_SEL	B Component Select When processing 32 bit per pixel, such as ARGB888, Blue component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
11 PIX_UV_SWAP	Pixel UV Swap If processing 1PYUV422 or 2PYUV420 pixels, swap position of Chroma pixel component position. 0x0 = UV 0x1 = VU If processing 1PYUV422, the PIX_UV_SWAP and PIX_LUMA_UV_SWAP combinations will decode over 2 pixels. Default pixel orientation: Pixel_1 = Y0_U0, and Pixel_2 = Y1_V0: (PIX_LUMA_UV_SWAP, PIX_UV_SWAP) 0,0 = Y0_U0 Y1_V0 0,1 = Y0_V0 Y1_U0 1,0 = U0_Y0 V0_Y1 1,1 = V0_Y0 U0_Y1
10 PIX_LUMA_UV_SWAP	Pixel luma/UV position Swap If processing 1PYUV422 pixels, swap position of Luma and Chroma pixel component position. 0x0 = YUYV 0x1 = UYVY
9-8 PIX_SIZE	Pixel Size 0x0 = 8 bits per pixel 0x1 = 16 bits per pixel 0x2 = 32 bits per pixel 0x3 = reserved Note: The source frame buffer for 2PYUV420 10 bit pixels are packed horizontally for every row of pixels. The DPR should be programmed in 8 bits per pixel mode. The source frame buffer for VP9-10 10 bit pixels are unpacked horizontally, 16bpp for every 10 bit pixel. The DPR should be programmed in 16 bits per pixel mode.
7 COMP_2PLANE_EN	Component 2-Plane Enable 0x0 = 1-Plane operation. i.e. YUV422 (1PYUV422) or RGB operation. 0x1 = 2-Plane operation. i.e. YUV420 (2PYUV420), 2PVP9-8 or 2pVP9-10 operation.
6 YUV_EN	YUV Enable 0x0 = xRGB modes of operation. 0x1 = 2-Plane VP9-8 (2PVP9-8), 2-Plane VP9-10 (2PVP9-10), 2-Plane YUV420 (2PYUV420) or 1-Plane YUV422 (1PYUV422) mode of operation.
5 —	Reserved.
4-2 TILE_TYPE	Tile Type 0x0 = Linear (raster) tile. 0x1 = GPU Standard Tile.

Table continues on the next page...

Field	Function
	0x2 = GPU Super Tile. 0x3 = VPU 2PYUV420 Tile. 0x4 = VPU 2-Plane VP9 Tile.
1 RTR_4LINE_BUF_EN	RTRAM Lines Per Buffer 0x0 = 8 RTRAM lines per buffer. This setting would be applied for non-linear VPU tiles. 0x1 = 4 RTRAM lines per buffer. This setting would be applied for non-linear GPU tiles. For linear tile_type, the number of RTRAM lines per buffer depends on the Display Controller Scaler or PRG configuration.
0 RTR_3BUF_EN	RTRAM Buffer Implementation 0x0 = Process 2 RTRAM buffers. 0x1 = Process 3 RTRAM buffers.

15.7.3.1.19 Mode Control 0 (MODE_CTRL0_SET)

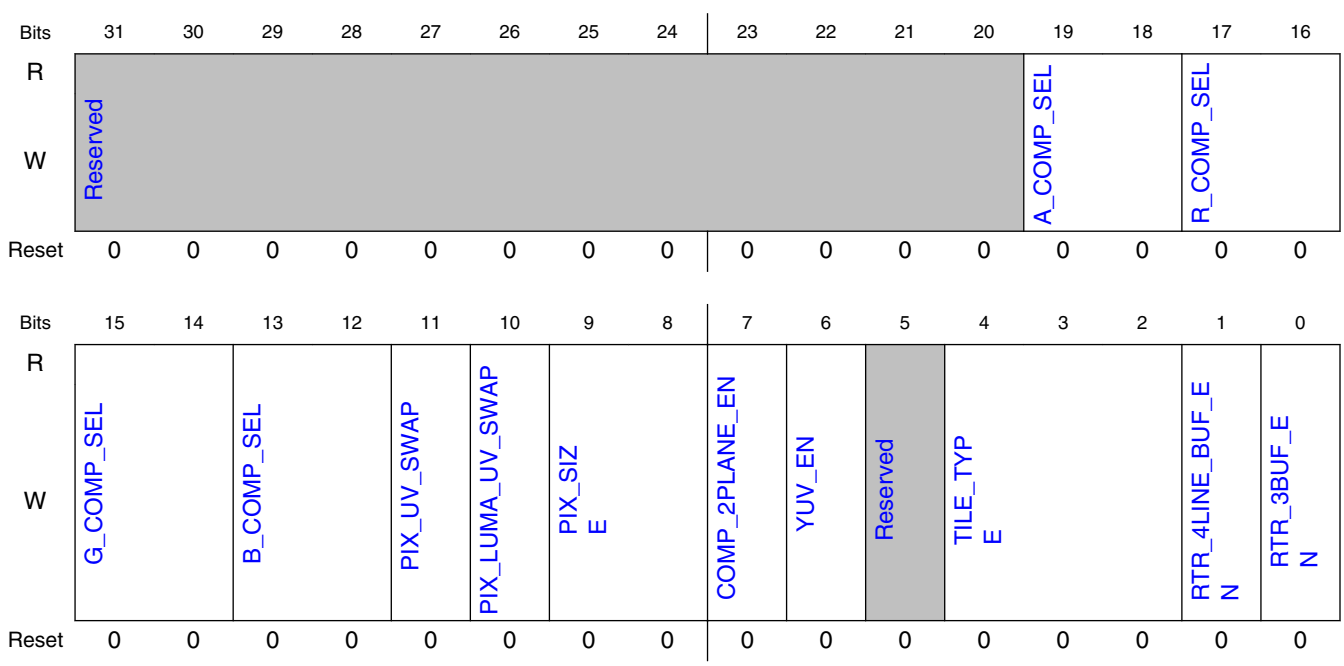
15.7.3.1.19.1 Offset

Register	Offset
MODE_CTRL0_SET	54h

15.7.3.1.19.2 Function

The Mode Control0 register controls mode of operation configuration.

15.7.3.1.19.3 Diagram



15.7.3.1.19.4 Fields

Field	Function
31-20 —	Reserved.
19-18 A_COMP_SEL	A Component Select 32 bit per pixel, such as ARGB888, Alpha component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
17-16 R_COMP_SEL	R Component Select 32 bit per pixel, such as ARGB888, Red component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
15-14 G_COMP_SEL	G Component Select 32 bit per pixel, such as ARGB888, Green component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
13-12 B_COMP_SEL	B Component Select When processing 32 bit per pixel, such as ARGB888, Blue component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
11 PIX_UV_SWAP	Pixel UV Swap If processing 1PYUV422 or 2PYUV420 pixels, swap position of Chroma pixel component position. 0x0 = UV 0x1 = VU If processing 1PYUV422, the PIX_UV_SWAP and PIX_LUMA_UV_SWAP combinations will decode over 2 pixels. Default pixel orientation: Pixel_1 = Y0_U0, and Pixel_2 = Y1_V0: (PIX_LUMA_UV_SWAP, PIX_UV_SWAP) 0,0 = Y0_U0 Y1_V0 0,1 = Y0_V0 Y1_U0 1,0 = U0_Y0 V0_Y1 1,1 = V0_Y0 U0_Y1
10 PIX_LUMA_UV_SWAP	Pixel luma/UV position Swap If processing 1PYUV422 pixels, swap position of Luma and Chroma pixel component position. 0x0 = YUYV 0x1 = UYVY
9-8	Pixel Size

Table continues on the next page...

Field	Function
PIX_SIZE	0x0 = 8 bits per pixel 0x1 = 16 bits per pixel 0x2 = 32 bits per pixel 0x3 = reserved Note: The source frame buffer for 2PYUV420 10 bit pixels are packed horizontally for every row of pixels. The DPR should be programmed in 8 bits per pixel mode. The source frame buffer for VP9-10 10 bit pixels are unpacked horizontally, 16bpp for every 10 bit pixel. The DPR should be programmed in 16 bits per pixel mode.
7 COMP_2PLANE_EN	Component 2-Plane Enable 0x0 = 1-Plane operation. i.e. YUV422 (1PYUV422) or RGB operation. 0x1 = 2-Plane operation. i.e. YUV420 (2PYUV420), 2PVP9-8 or 2pVP9-10 operation.
6 YUV_EN	YUV Enable 0x0 = xRGB modes of operation. 0x1 = 2-Plane VP9-8 (2PVP9-8), 2-Plane VP9-10 (2PVP9-10), 2-Plane YUV420 (2PYUV420) or 1-Plane YUV422 (1PYUV422) mode of operation.
5 —	Reserved.
4-2 TILE_TYPE	Tile Type 0x0 = Linear (raster) tile. 0x1 = GPU Standard Tile. 0x2 = GPU Super Tile. 0x3 = VPU 2PYUV420 Tile. 0x4 = VPU 2-Plane VP9 Tile.
1 RTR_4LINE_BUF_EN	RTRAM Lines Per Buffer 0x0 = 8 RTRAM lines per buffer. This setting would be applied for non-linear VPU tiles. 0x1 = 4 RTRAM lines per buffer. This setting would be applied for non-linear GPU tiles. For linear tile_type, the number of RTRAM lines per buffer depends on the Display Controller Scaler or PRG configuration.
0 RTR_3BUF_EN	RTRAM Buffer Implementation 0x0 = Process 2 RTRAM buffers. 0x1 = Process 3 RTRAM buffers.

15.7.3.1.20 Mode Control 0 (MODE_CTRL0_CLR)

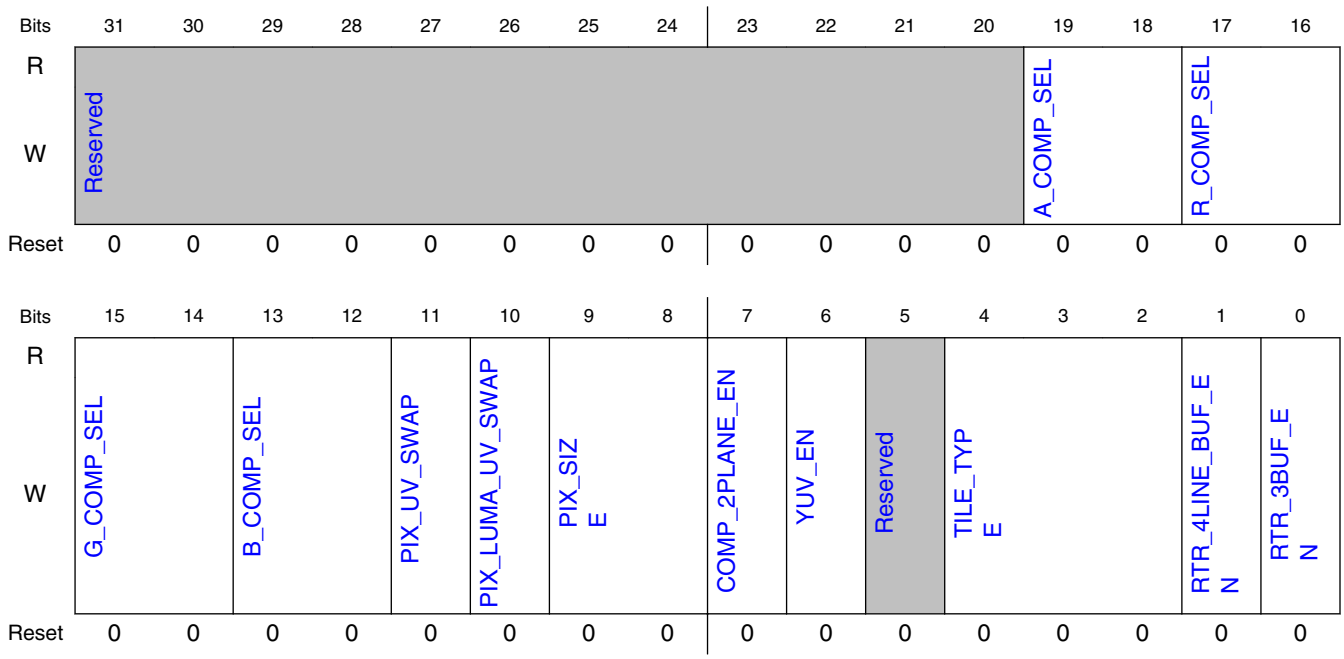
15.7.3.1.20.1 Offset

Register	Offset
MODE_CTRL0_CLR	58h

15.7.3.1.20.2 Function

The Mode Control0 register controls mode of operation configuration.

15.7.3.1.20.3 Diagram



15.7.3.1.20.4 Fields

Field	Function
31-20 —	Reserved.
19-18 A_COMP_SEL	A Component Select 32 bit per pixel, such as ARGB888, Alpha component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
17-16 R_COMP_SEL	R Component Select 32 bit per pixel, such as ARGB888, Red component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
15-14 G_COMP_SEL	G Component Select 32 bit per pixel, such as ARGB888, Green component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
13-12 B_COMP_SEL	B Component Select When processing 32 bit per pixel, such as ARGB888, Blue component byte select 0x0 = byte 0, bits[7:0]

Table continues on the next page...

Field	Function
	0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
11 PIX_UV_SWAP	Pixel UV Swap If processing 1PYUV422 or 2PYUV420 pixels, swap position of Chroma pixel component position. 0x0 = UV 0x1 = VU If processing 1PYUV422, the PIX_UV_SWAP and PIX_LUMA_UV_SWAP combinations will decode over 2 pixels. Default pixel orientation: Pixel_1 = Y0_U0, and Pixel_2 = Y1_V0: (PIX_LUMA_UV_SWAP, PIX_UV_SWAP) 0,0 = Y0_U0 Y1_V0 0,1 = Y0_V0 Y1_U0 1,0 = U0_Y0 V0_Y1 1,1 = V0_Y0 U0_Y1
10 PIX_LUMA_UV_SWAP	Pixel luma/UV position Swap If processing 1PYUV422 pixels, swap position of Luma and Chroma pixel component position. 0x0 = YUYV 0x1 = UYVY
9-8 PIX_SIZE	Pixel Size 0x0 = 8 bits per pixel 0x1 = 16 bits per pixel 0x2 = 32 bits per pixel 0x3 = reserved Note: The source frame buffer for 2PYUV420 10 bit pixels are packed horizontally for every row of pixels. The DPR should be programmed in 8 bits per pixel mode. The source frame buffer for VP9-10 10 bit pixels are unpacked horizontally, 16bpp for every 10 bit pixel. The DPR should be programmed in 16 bits per pixel mode.
7 COMP_2PLANE_EN	Component 2-Plane Enable 0x0 = 1-Plane operation. i.e. YUV422 (1PYUV422) or RGB operation. 0x1 = 2-Plane operation. i.e. YUV420 (2PYUV420), 2PVP9-8 or 2pVP9-10 operation.
6 YUV_EN	YUV Enable 0x0 = xRGB modes of operation. 0x1 = 2-Plane VP9-8 (2PVP9-8), 2-Plane VP9-10 (2PVP9-10), 2-Plane YUV420 (2PYUV420) or 1-Plane YUV422 (1PYUV422) mode of operation.
5 —	Reserved.
4-2 TILE_TYPE	Tile Type 0x0 = Linear (raster) tile. 0x1 = GPU Standard Tile. 0x2 = GPU Super Tile. 0x3 = VPU 2PYUV420 Tile. 0x4 = VPU 2-Plane VP9 Tile.
1 RTR_4LINE_BUF_EN	RTRAM Lines Per Buffer 0x0 = 8 RTRAM lines per buffer. This setting would be applied for non-linear VPU tiles. 0x1 = 4 RTRAM lines per buffer. This setting would be applied for non-linear GPU tiles.

Table continues on the next page...

Field	Function
	For linear tile_type, the number of RTRAM lines per buffer depends on the Display Controller Scaler or PRG configuration.
0 RTR_3BUF_EN	RTRAM Buffer Implementation 0x0 = Process 2 RTRAM buffers. 0x1 = Process 3 RTRAM buffers.

15.7.3.1.21 Mode Control 0 (MODE_CTRL0_TOG)

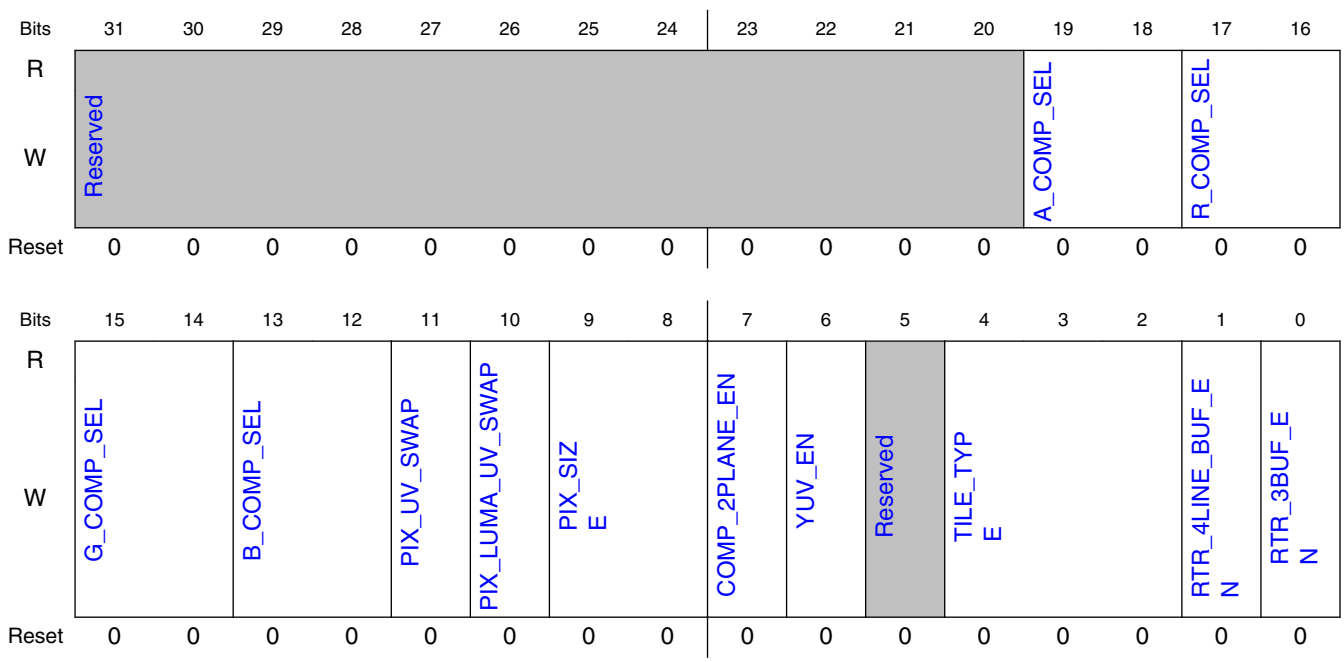
15.7.3.1.21.1 Offset

Register	Offset
MODE_CTRL0_TOG	5Ch

15.7.3.1.21.2 Function

The Mode Control0 register controls mode of operation configuration.

15.7.3.1.21.3 Diagram



15.7.3.1.21.4 Fields

Field	Function
31-20 —	Reserved.
19-18 A_COMP_SEL	A Component Select 32 bit per pixel, such as ARGB888, Alpha component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
17-16 R_COMP_SEL	R Component Select 32 bit per pixel, such as ARGB888, Red component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
15-14 G_COMP_SEL	G Component Select 32 bit per pixel, such as ARGB888, Green component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
13-12 B_COMP_SEL	B Component Select When processing 32 bit per pixel, such as ARGB888, Blue component byte select 0x0 = byte 0, bits[7:0] 0x1 = byte 1, bits[15:8] 0x2 = byte 2, bits[23:16] 0x3 = byte 3, bits[31:24]
11 PIX_UV_SWAP	Pixel UV Swap If processing 1PYUV422 or 2PYUV420 pixels, swap position of Chroma pixel component position. 0x0 = UV 0x1 = VU If processing 1PYUV422, the PIX_UV_SWAP and PIX_LUMA_UV_SWAP combinations will decode over 2 pixels. Default pixel orientation: Pixel_1 = Y0_U0, and Pixel_2 = Y1_V0: (PIX_LUMA_UV_SWAP, PIX_UV_SWAP) 0,0 = Y0_U0 Y1_V0 0,1 = Y0_V0 Y1_U0 1,0 = U0_Y0 V0_Y1 1,1 = V0_Y0 U0_Y1
10 PIX_LUMA_UV_SWAP	Pixel luma/UV position Swap If processing 1PYUV422 pixels, swap position of Luma and Chroma pixel component position. 0x0 = YUYV 0x1 = UYVY
9-8 PIX_SIZE	Pixel Size 0x0 = 8 bits per pixel 0x1 = 16 bits per pixel

Table continues on the next page...

Field	Function
	0x2 = 32 bits per pixel 0x3 = reserved Note: The source frame buffer for 2PYUV420 10 bit pixels are packed horizontally for every row of pixels. The DPR should be programmed in 8 bits per pixel mode. The source frame buffer for VP9-10 10 bit pixels are unpacked horizontally, 16bpp for every 10 bit pixel. The DPR should be programmed in 16 bits per pixel mode.
7 COMP_2PLANE_EN	Component 2-Plane Enable 0x0 = 1-Plane operation. i.e. YUV422 (1PYUV422) or RGB operation. 0x1 = 2-Plane operation. i.e. YUV420 (2PYUV420), 2PVP9-8 or 2pVP9-10 operation.
6 YUV_EN	YUV Enable 0x0 = xRGB modes of operation. 0x1 = 2-Plane VP9-8 (2PVP9-8), 2-Plane VP9-10 (2PVP9-10), 2-Plane YUV420 (2PYUV420) or 1-Plane YUV422 (1PYUV422) mode of operation.
5 —	Reserved.
4-2 TILE_TYPE	Tile Type 0x0 = Linear (raster) tile. 0x1 = GPU Standard Tile. 0x2 = GPU Super Tile. 0x3 = VPU 2PYUV420 Tile. 0x4 = VPU 2-Plane VP9 Tile.
1 RTR_4LINE_BUF_EN	RTRAM Lines Per Buffer 0x0 = 8 RTRAM lines per buffer. This setting would be applied for non-linear VPU tiles. 0x1 = 4 RTRAM lines per buffer. This setting would be applied for non-linear GPU tiles. For linear tile_type, the number of RTRAM lines per buffer depends on the Display Controller Scaler or PRG configuration.
0 RTR_3BUF_EN	RTRAM Buffer Implementation 0x0 = Process 2 RTRAM buffers. 0x1 = Process 3 RTRAM buffers.

15.7.3.1.22 Frame Control 0 (FRAME_CTRL0)

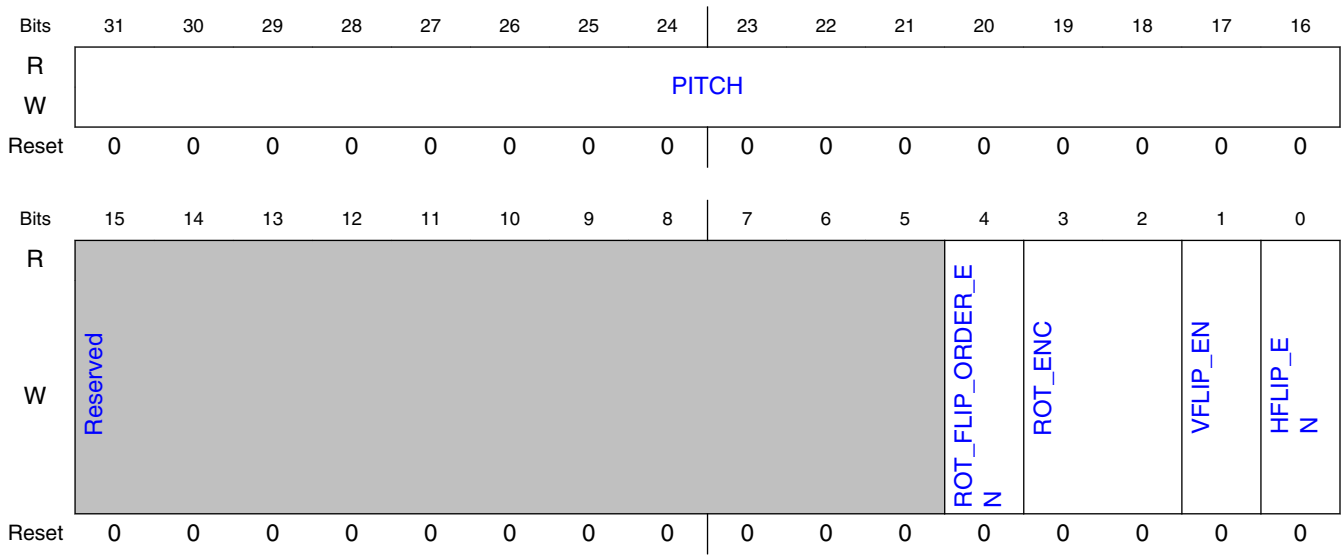
15.7.3.1.22.1 Offset

Register	Offset
FRAME_CTRL0	70h

15.7.3.1.22.2 Function

The Frame Control0 register controls fields that are common to processing both 1-Plane and 2-Plane image processing.

15.7.3.1.22.3 Diagram



15.7.3.1.22.4 Fields

Field	Function

31-16
PITCH
	Image Pitch
	
	The number of bytes to next horizontal line.
	
	First calculate pitch as usual for a raster image
	bpp = (PIX_SIZE==2) ? 4 // 32bpp
	(PIX_SIZE==1) ? 2 // 16bpp
	? 1; // 8bpp
	image_pitch = x_wide * bpp;
	
	Second calculae offset for GPU tile types
	// Calculate pixel offset for GPU Super Tile to prevent address alising of pixels
	// The Super Tile is 64x64 pixel block, so width and height must be divisible by 64 to prevent alising.
	// decode number of pixels in a 64byte boundary
	num_pix_x_dir = ((PIX_SIZE==2) && (TILE_TYPE==2)) ? 64 : // GPU 32bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	((PIX_SIZE==1) && (TILE_TYPE==2)) ? 64 : // GPU 16bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	
	image_pitch; //
	
	// modulus of pix_x_width by number of pixels in a 64byte boundary.
	pix_x_mod = image_width % num_pix_x_dir;
	// number of pixels to offset in x-direction.
	pix_x_offset = (pix_x_mod == 0) ? 0 : (num_pix_x_dir - pix_x_mod);
	
	Third calculate final pitch value
	PITCH =
	(TILE_TYPE==2) ?
	pix_x_offset * bpp :

Table continues on the next page...

Field	Function
	(TILE_TYPE==1) && (PIX_SIZE==2) && ((x_wide%4)>0) ? ({x_wide[15:2],2'h0} + 4) * bpp : (TILE_TYPE==1) && (PIX_SIZE==1) && ((x_wide%8)>0) ? ({x_wide[15:3],3'h0} + 8) * bpp : ((TILE_TYPE==3) || (TILE_TYPE==4)) && ((x_wide%8)>0) ? ({x_wide[15:3],3'h0} + 8) * bpp : image_pitch;
15-5 —	Reserved.
4 ROT_FLIP_ORDER_EN	Rotation Flip Order The order of operation for rotate and/or horizontal/vertical flip of the output image 0x0 = Rotate followed by horizontal/vertical flip 0x1 = horizonalt/vertical flip followed by rotate
3-2 ROT_ENC	Encoded Rotation Rotate the output image 0x0 = Rotate by 0 degrees 0x1 = Rotate by 90 degrees counter clock wise 0x2 = Rotate by 180 degrees counter clock wise 0x3 = Rotate by 270 degrees counter clock wise Rotation by 90 and 270 degrees is not supported for the following: • All linear (raster) tile modes of operation • 16bpp GPU Standard Tile mode of operation • 16bpp GPU Super Tile mode of operation • VPU 2PYUV420 10bpp mode • VPU 2PVP9 8bpp mode • VPU 2PVP9 10bpp mode
1 VFLIP_EN	Vertical Flip Enable Vertical Flip the output image. 0x0 = Disable vertical flip 0x1 = Enable Vertical flip
0 HFLIP_EN	Horizontal Flip Enable Horizontal Flip the output image. 0x0 = Disable horizontal flip 0x1 = Enable Horizontal flip Note: HFLIP is not supported for the following: • VPU 2PYUV420 10bpp mode • VPU 2-Plane VP9 10bpp mode

15.7.3.1.23 Frame Control 0 (FRAME_CTRL0_SET)

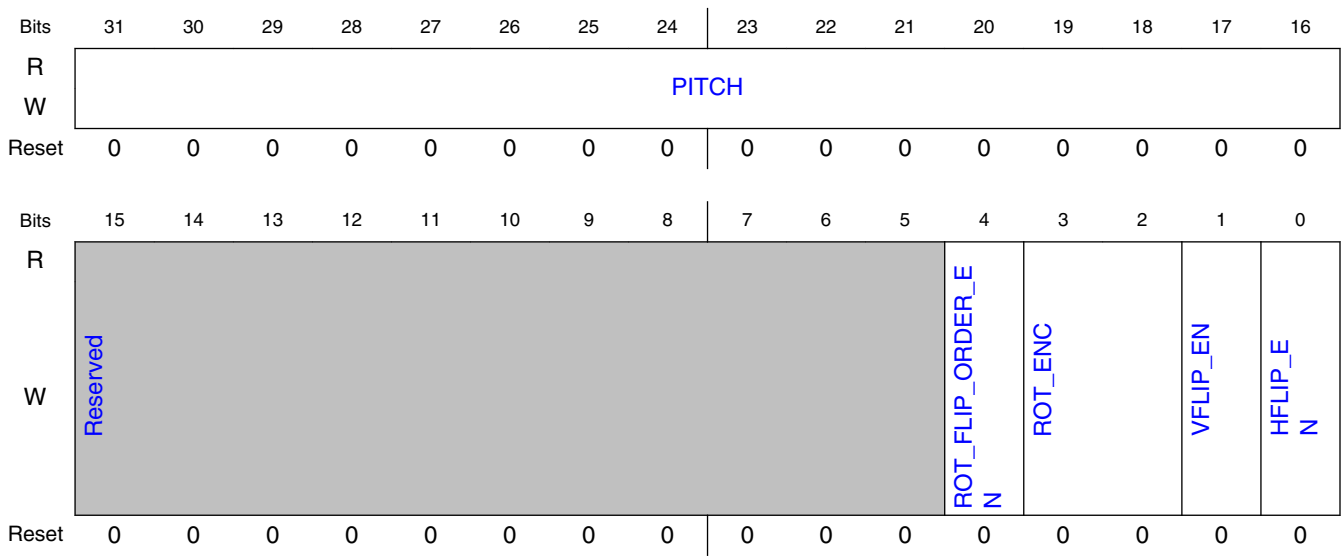
15.7.3.1.23.1 Offset

Register	Offset
FRAME_CTRL0_SET	74h

15.7.3.1.23.2 Function

The Frame Control0 register controls fields that are common to processing both 1-Plane and 2-Plane image processing.

15.7.3.1.23.3 Diagram



15.7.3.1.23.4 Fields

Field	Function

31-16
PITCH
	Image Pitch
	
	The number of bytes to next horizontal line.
	
	First calculate pitch as usual for a raster image
	bpp = (PIX_SIZE==2) ? 4 // 32bpp
	(PIX_SIZE==1) ? 2 // 16bpp
	? 1; // 8bpp
	image_pitch = x_wide * bpp;
	
	Second calculae offset for GPU tile types
	// Calculate pixel offset for GPU Super Tile to prevent address alising of pixels
	// The Super Tile is 64x64 pixel block, so width and height must be divisible by 64 to prevent alising.
	// decode number of pixels in a 64byte boundary
	num_pix_x_dir = ((PIX_SIZE==2) && (TILE_TYPE==2)) ? 64 : // GPU 32bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	((PIX_SIZE==1) && (TILE_TYPE==2)) ? 64 : // GPU 16bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	
	image_pitch; //

Table continues on the next page...

Field	Function

	`// modulus of pix_x_width by number of pixels in a 64byte boundary.`
	`pix_x_mod = image_width % num_pix_x_dir;`
	`// number of pixels to offset in x-direction.`
	`pix_x_offset = (pix_x_mod == 0) ? 0 : (num_pix_x_dir - pix_x_mod);`
	
	`Third calculate final pitch value`
	`PITCH = (TILE_TYPE==2) ? pix_x_offset * bpp :`
	`(TILE_TYPE==1) && (PIX_SIZE==2) && ((x_wide%4)>0) ? ({x_wide[15:2],2'h0} + 4) * bpp :`
	`(TILE_TYPE==1) && (PIX_SIZE==1) && ((x_wide%8)>0) ? ({x_wide[15:3],3'h0} + 8) * bpp :`
	`((TILE_TYPE==3) || (TILE_TYPE==4)) && ((x_wide%8)>0) ? ({x_wide[15:3],3'h0} + 8) * bpp :`
	
	`image_pitch;`

15-5 —	Reserved.
4 ROT_FLIP_ORDER_EN	Rotation Flip Order The order of operation for rotate and/or horizontal/vertical flip of the output image 0x0 = Rotate followed by horizontal/vertical flip 0x1 = horizonalt/vertical flip followed by rotate
3-2 ROT_ENC	Encoded Rotation Rotate the output image 0x0 = Rotate by 0 degrees 0x1 = Rotate by 90 degrees counter clock wise 0x2 = Rotate by 180 degrees counter clock wise 0x3 = Rotate by 270 degrees counter clock wise Rotation by 90 and 270 degrees is not supported for the following: • All linear (raster) tile modes of operation • 16bpp GPU Standard Tile mode of operation • 16bpp GPU Super Tile mode of operation • VPU 2PYUV420 10bpp mode • VPU 2PVP9 8bpp mode • VPU 2PVP9 10bpp mode
1 VFLIP_EN	Vertical Flip Enable Vertical Flip the output image. 0x0 = Disable vertical flip 0x1 = Enable Vertical flip
0 HFLIP_EN	Horizontal Flip Enable Horizontal Flip the output image. 0x0 = Disable horizontal flip 0x1 = Enable Horizontal flip Note: HFLIP is not supported for the following: • VPU 2PYUV420 10bpp mode • VPU 2-Plane VP9 10bpp mode

15.7.3.1.24 Frame Control 0 (FRAME_CTRL0_CLR)

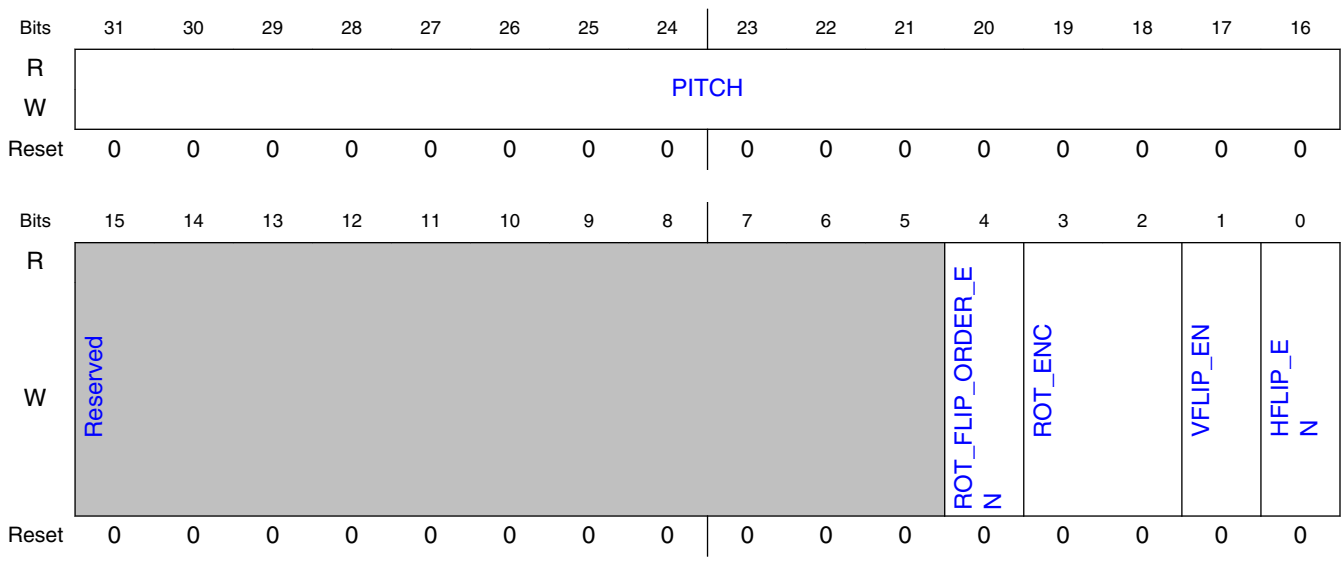
15.7.3.1.24.1 Offset

Register	Offset
FRAME_CTRL0_CLR	78h

15.7.3.1.24.2 Function

The Frame Control0 register controls fields that are common to processing both 1-Plane and 2-Plane image processing.

15.7.3.1.24.3 Diagram



15.7.3.1.24.4 Fields

Field	Function

31-16
PITCH
	Image Pitch
	
	The number of bytes to next horizontal line.
	
	First calculate pitch as usual for a raster image
	bpp = (PIX_SIZE==2) ? 4 // 32bpp
	(PIX_SIZE==1) ? 2 // 16bpp
	? 1; // 8bpp
	image_pitch = x_wide * bpp;
	
	Second calculae offset for GPU tile types
	// Calculate pixel offset for GPU Super Tile to prevent address alising of pixels

Table continues on the next page...

Field	Function

	// The Super Tile is 64x64 pixel block, so width and height must be divisible by 64 to prevent alising.
	// decode number of pixels in a 64byte boundary
	num_pix_x_dir = ((PIX_SIZE==2) && (TILE_TYPE==2)) ? 64 : // GPU 32bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	((PIX_SIZE==1) && (TILE_TYPE==2)) ? 64 : // GPU 16bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	
	image_pitch; //
	
	// modulus of pix_x_width by number of pixels in a 64byte boundary.
	pix_x_mod = image_width % num_pix_x_dir;
	// number of pixels to offset in x-direction.
	pix_x_offset = (pix_x_mod == 0) ? 0 : (num_pix_x_dir - pix_x_mod);
	
	Third calculate final pitch value
	PITCH =
	(TILE_TYPE==2) ?
	pix_x_offset * bpp :
	(TILE_TYPE==1) && (PIX_SIZE==2) && ((x_wide%4)>0) ?
	({x_wide[15:2],2'h0} + 4) * bpp :
	(TILE_TYPE==1) && (PIX_SIZE==1) && ((x_wide%8)>0) ?
	({x_wide[15:3],3'h0} + 8) * bpp :
	((TILE_TYPE==3) || (TILE_TYPE==4)) && ((x_wide%8)>0) ?
	({x_wide[15:3],3'h0} + 8) * bpp :
	
	image_pitch;

15-5 —	Reserved.
4 ROT_FLIP_ORDER_EN	Rotation Flip Order The order of operation for rotate and/or horizontal/vertical flip of the output image 0x0 = Rotate followed by horizontal/vertical flip 0x1 = horizonalt/vertical flip followed by rotate
3-2 ROT_ENC	Encoded Rotation Rotate the output image 0x0 = Rotate by 0 degrees 0x1 = Rotate by 90 degrees counter clock wise 0x2 = Rotate by 180 degrees counter clock wise 0x3 = Rotate by 270 degrees counter clock wise Rotation by 90 and 270 degrees is not supported for the following: • All linear (raster) tile modes of operation • 16bpp GPU Standard Tile mode of operation • 16bpp GPU Super Tile mode of operation • VPU 2PYUV420 10bpp mode • VPU 2PVP9 8bpp mode • VPU 2PVP9 10bpp mode
1 VFLIP_EN	Vertical Flip Enable Vertical Flip the output image. 0x0 = Disable vertical flip 0x1 = Enable Vertical flip
0	Horizontal Flip Enable

Field	Function
HFLIP_EN	Horizontal Flip the output image. 0x0 = Disable horizontal flip 0x1 = Enable Horizontal flip Note: HFLIP is not supported for the following: • VPU 2PYUV420 10bpp mode • VPU 2-Plane VP9 10bpp mode

15.7.3.1.25 Frame Control 0 (FRAME_CTRL0_TOG)

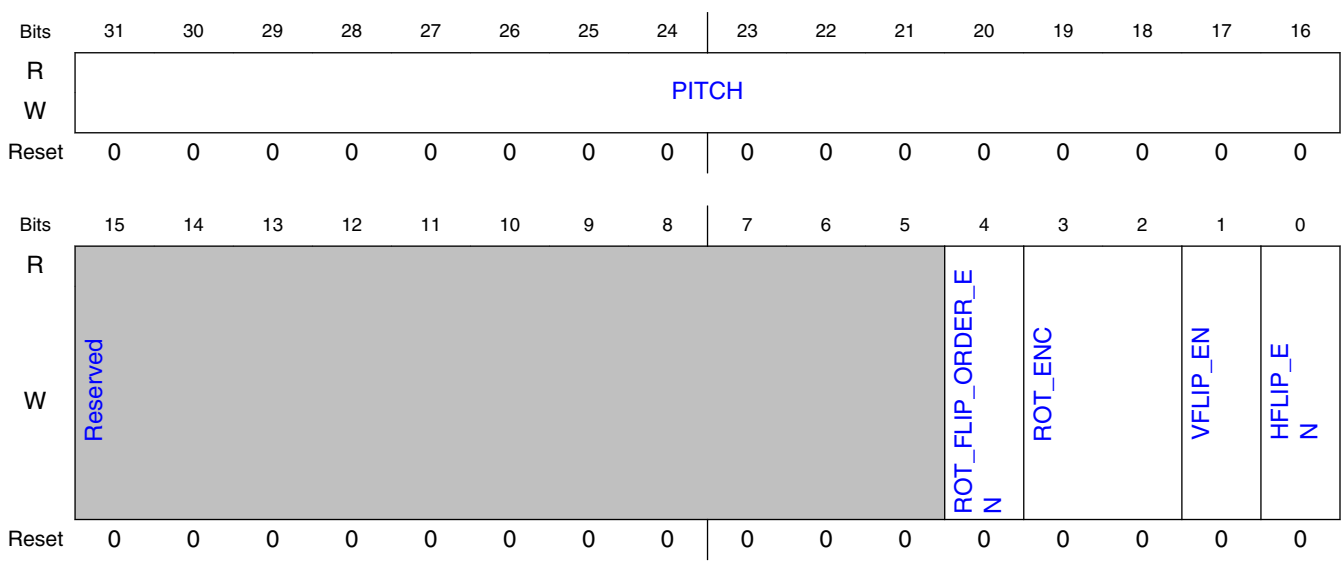
15.7.3.1.25.1 Offset

Register	Offset
FRAME_CTRL0_TOG	7Ch

15.7.3.1.25.2 Function

The Frame Control0 register controls fields that are common to processing both 1-Plane and 2-Plane image processing.

15.7.3.1.25.3 Diagram



15.7.3.1.25.4 Fields

Field	Function

31-16
PITCH
	Image Pitch
	
	The number of bytes to next horizontal line.
	
	First calculate pitch as usual for a raster image
	bpp = (PIX_SIZE==2) ? 4 // 32bpp
	(PIX_SIZE==1) ? 2 // 16bpp
	? 1; // 8bpp
	image_pitch = x_wide * bpp;
	
	Second calculae offset for GPU tile types
	// Calculate pixel offset for GPU Super Tile to prevent address alising of pixels
	// The Super Tile is 64x64 pixel block, so width and height must be divisible by 64 to prevent alising.
	// decode number of pixels in a 64byte boundary
	num_pix_x_dir = ((PIX_SIZE==2) && (TILE_TYPE==2)) ? 64 : // GPU 32bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	((PIX_SIZE==1) && (TILE_TYPE==2)) ? 64 : // GPU 16bpp super tile must be divisible by 64 pixels in x-direction to prevent aliasing
	
	image_pitch; //
	
	// modulus of pix_x_width by number of pixels in a 64byte boundary.
	pix_x_mod = image_width % num_pix_x_dir;
	// number of pixels to offset in x-direction.
	pix_x_offset = (pix_x_mod == 0) ? 0 : (num_pix_x_dir - pix_x_mod);
	
	Third calculate final pitch value
	PITCH =
	(TILE_TYPE==2) ?
	pix_x_offset * bpp :
	(TILE_TYPE==1) && (PIX_SIZE==2) && ((x_wide%4)>0) ?
	({x_wide[15:2],2'h0} + 4) * bpp :
	(TILE_TYPE==1) && (PIX_SIZE==1) && ((x_wide%8)>0) ?
	({x_wide[15:3],3'h0} + 8) * bpp :
	((TILE_TYPE==3) || (TILE_TYPE==4)) && ((x_wide%8)>0) ?
	({x_wide[15:3],3'h0} + 8) * bpp :
	
	image_pitch;

15-5 —	Reserved.
4 ROT_FLIP_ORDER_EN	Rotation Flip Order The order of operation for rotate and/or horizontal/vertical flip of the output image 0x0 = Rotate followed by horizontal/vertical flip 0x1 = horizonalt/vertical flip followed by rotate
3-2 ROT_ENC	Encoded Rotation Rotate the output image 0x0 = Rotate by 0 degrees 0x1 = Rotate by 90 degrees counter clock wise 0x2 = Rotate by 180 degrees counter clock wise 0x3 = Rotate by 270 degrees counter clock wise

Table continues on the next page...

Field	Function
	Rotation by 90 and 270 degrees is not supported for the following: • All linear (raster) tile modes of operation • 16bpp GPU Standard Tile mode of operation • 16bpp GPU Super Tile mode of operation • VPU 2PYUV420 10bpp mode • VPU 2PVP9 8bpp mode • VPU 2PVP9 10bpp mode
1 VFLIP_EN	Vertical Flip Enable Vertical Flip the output image. 0x0 = Disable vertical flip 0x1 = Enable Vertical flip
0 HFLIP_EN	Horizontal Flip Enable Horizontal Flip the output image. 0x0 = Disable horizontal flip 0x1 = Enable Horizontal flip Note: HFLIP is not supported for the following: • VPU 2PYUV420 10bpp mode • VPU 2-Plane VP9 10bpp mode

15.7.3.1.26 Frame 1-Plane Control 0 (FRAME_1P_CTRL0)

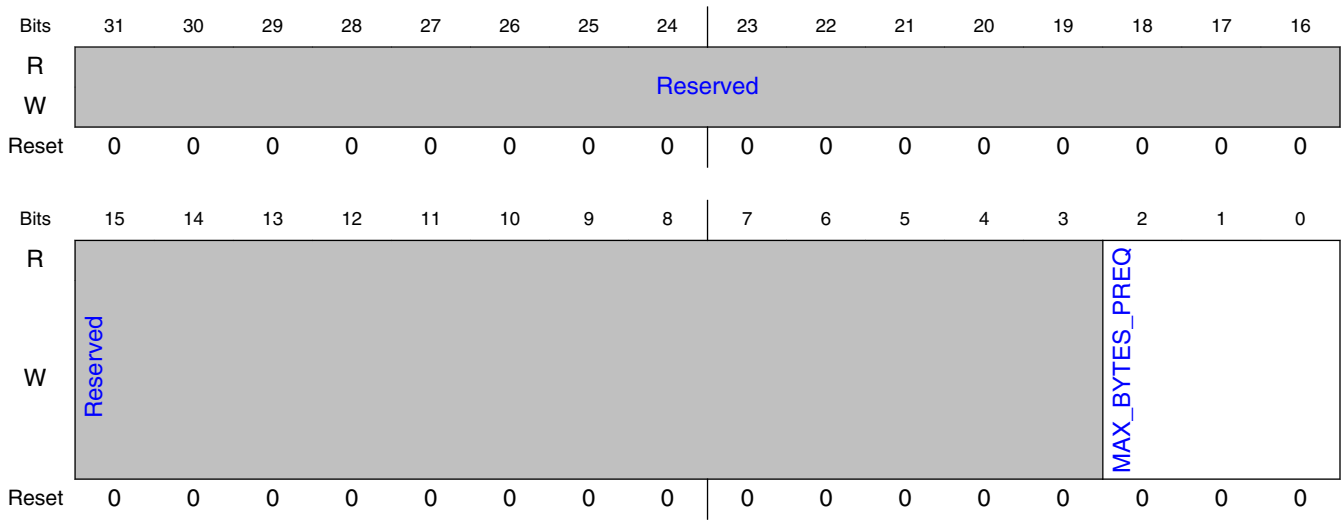
15.7.3.1.26.1 Offset

Register	Offset
FRAME_1P_CTRL0	90h

15.7.3.1.26.2 Function

The Frame 1P Control0 register controls fields only for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.26.3 Diagram



15.7.3.1.26.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.27 Frame 1-Plane Control 0 (FRAME_1P_CTRL0_SET)

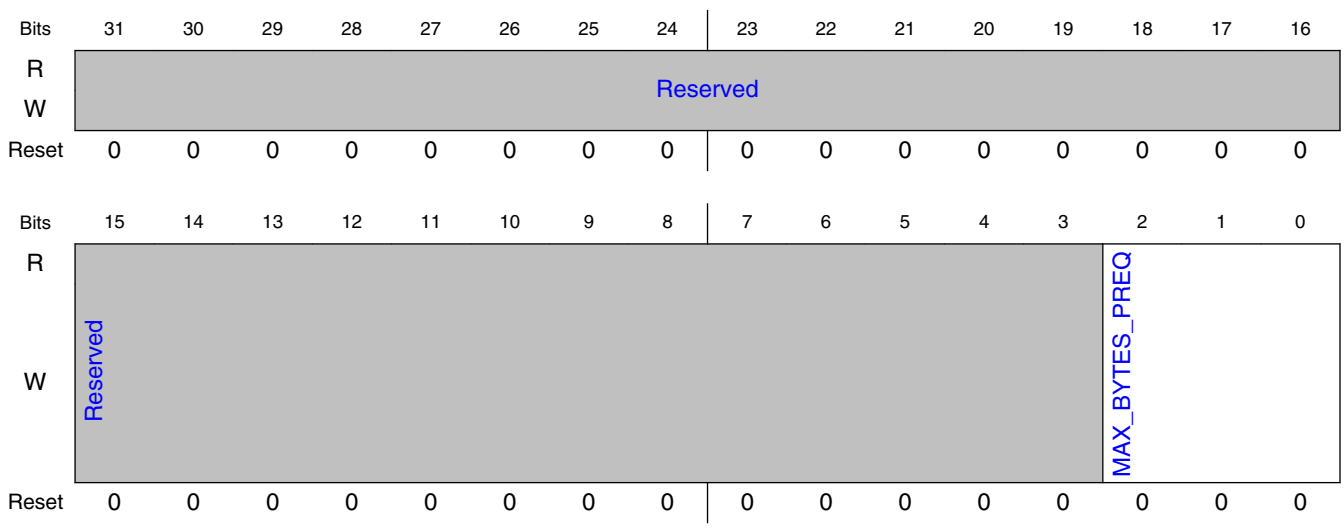
15.7.3.1.27.1 Offset

Register	Offset
FRAME_1P_CTRL0_SET	94h

15.7.3.1.27.2 Function

The Frame 1P Control0 register controls fields only for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.27.3 Diagram



15.7.3.1.27.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes

Field	Function
	Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.28 Frame 1-Plane Control 0 (FRAME_1P_CTRL0_CLR)

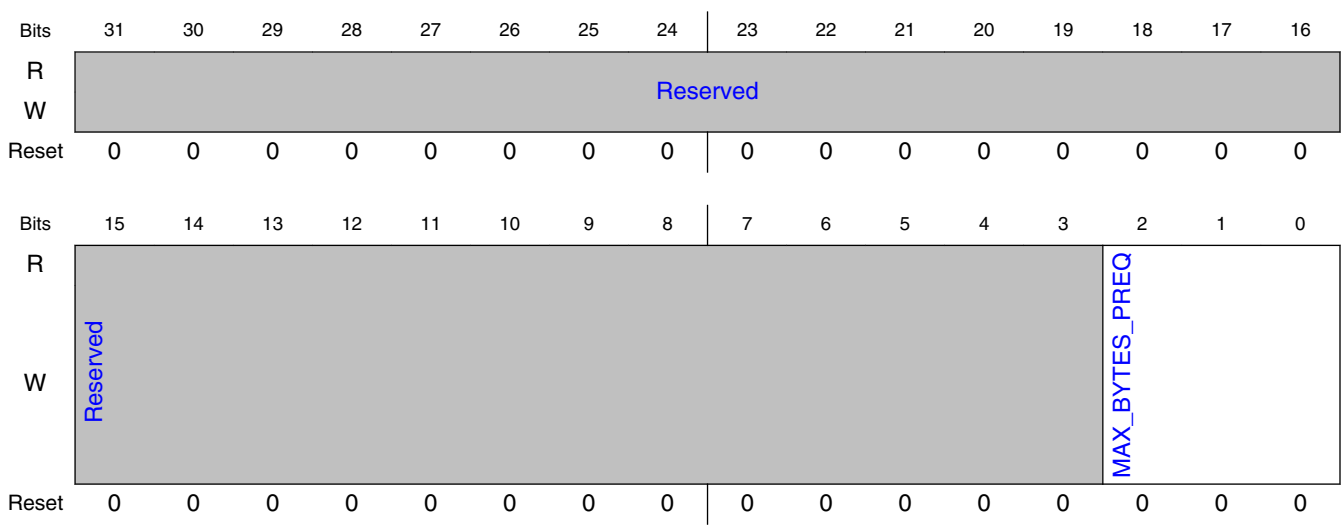
15.7.3.1.28.1 Offset

Register	Offset
FRAME_1P_CTRL0_CLR	98h

15.7.3.1.28.2 Function

The Frame 1P Control0 register controls fields only for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.28.3 Diagram



15.7.3.1.28.4 Fields

Field	Function
31-3	Reserved.

Table continues on the next page...

Field	Function
—	
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.29 Frame 1-Plane Control 0 (FRAME_1P_CTRL0_TOG)

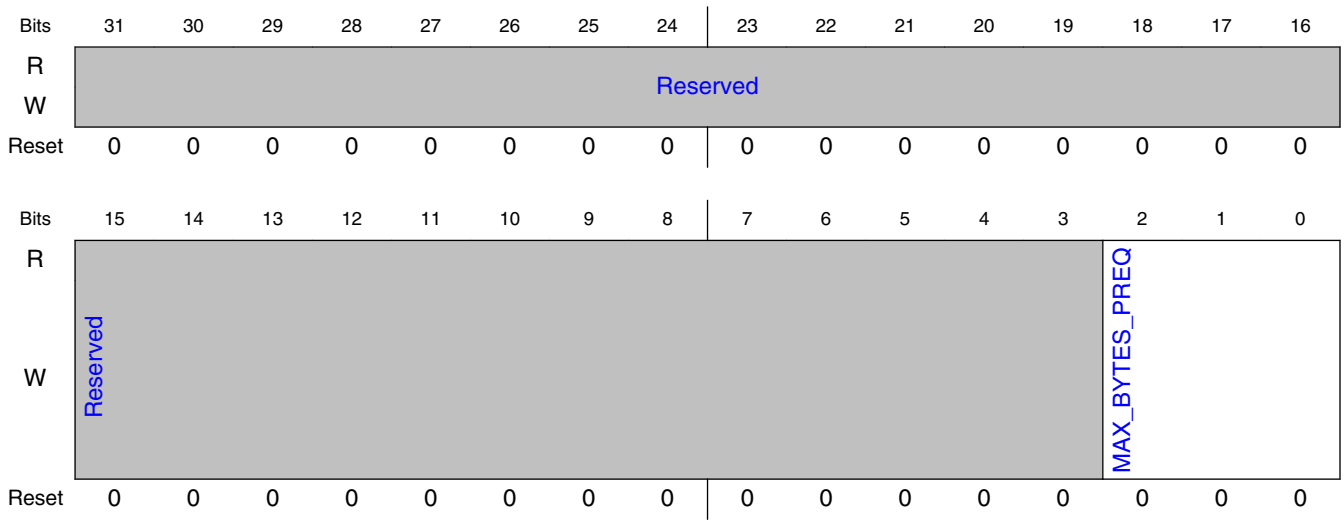
15.7.3.1.29.1 Offset

Register	Offset
FRAME_1P_CTRL0_TOG	9Ch

15.7.3.1.29.2 Function

The Frame 1P Control0 register controls fields only for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.29.3 Diagram



15.7.3.1.29.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.30 Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL)

15.7.3.1.30.1 Offset

Register	Offset
FRAME_1P_PIX_X_CTRL	A0h

15.7.3.1.30.2 Function

The Frame 1P PIX_X Control register controls x-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.30.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 |

Table continues on the next page...

Field	Function
	|------------------------------|--------------------|--------------------|

15-0
NUM_X_PIX_WIDE
	Number of Pixels Wide in X-direction
	
	The number of input frame pixels to fetch in the x-axis direction.
	In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed.
	
	The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied.
	To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes:
	
	First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings:
	num_pix_x_in_64byte = (((PIX_SIZE==2) && (TILE_TYPE==0)) ? 16 : // GPU 32bpp linear tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==2) && ((TILE_TYPE==1) || (TILE_TYPE==2))) ? 4 : // GPU 32bpp standard and/or super tile must be divisible by 4 pixels in x-direction
	((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // GPU 16bpp linear tile must be divisible by 32 pixels in x-direction. VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction
	((PIX_SIZE==1) && ((TILE_TYPE==1) || (TILE_TYPE==2) || (TILE_TYPE==4))) ? 8 : // GPU 16bpp standard and/or super tile must be divisible by 8 pixels in x-direction. VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp Luma zenverge tile must be divisible by 8 pixels in x-direction
	
	8); // default case shouldn't be hit
	
	Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte
	Note: for 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation.
	pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte
	
	Third calculate the number of offset pixels
	pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod)
	
	Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers.
	NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset
	
	Note1: If FRAME_1P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.31 Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_SET)

15.7.3.1.31.1 Offset

Register	Offset
FRAME_1P_PIX_X_CTRL_SET	A4h

15.7.3.1.31.2 Function

The Frame 1P PIX_X Control register controls x-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.31.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 |

Table continues on the next page...

Field	Function
	| VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_X_PIX_WIDE
	Number of Pixels Wide in X-direction
	
	The number of input frame pixels to fetch in the x-axis direction.
	In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed.
	
	The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied.
	To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes:
	
	First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings:
	num_pix_x_in_64byte = (((PIX_SIZE==2) && (TILE_TYPE==0)) ? 16 : // GPU 32bpp linear tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==2) && ((TILE_TYPE==1) || (TILE_TYPE==2))) ? 4 : // GPU 32bpp standard and/or super tile must be divisible by 4 pixels in x-direction
	((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // GPU 16bpp linear tile must be divisible by 32 pixels in x-direction. VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction
	((PIX_SIZE==1) && ((TILE_TYPE==1) || (TILE_TYPE==2) || (TILE_TYPE==4))) ? 8 : // GPU 16bpp standard and/or super tile must be divisible by 8 pixels in x-direction. VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp Luma zenverge tile must be divisible by 8 pixels in x-direction
	
	8); // default case shouldn't be hit
	
	Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte
	Note: for 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation.
	pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte
	
	Third calculate the number of offset pixels
	pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod)
	
	Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers.
	NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset
	
	Note1: If FRAME_1P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.32 Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_CLR)

15.7.3.1.32.1 Offset

Register	Offset
FRAME_1P_PIX_X_CTRL_CLR	A8h

15.7.3.1.32.2 Function

The Frame 1P PIX_X Control register controls x-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.32.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 |

Table continues on the next page...

Field	Function
	| GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_X_PIX_WIDE
	Number of Pixels Wide in X-direction
	
	The number of input frame pixels to fetch in the x-axis direction.
	In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed.
	
	The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied.
	To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes:
	
	First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings:
	num_pix_x_in_64byte = (((PIX_SIZE==2) && (TILE_TYPE==0)) ? 16 : // GPU 32bpp linear tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==2) && ((TILE_TYPE==1) || (TILE_TYPE==2))) ? 4 : // GPU 32bpp standard and/or super tile must be divisible by 4 pixels in x-direction
	((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // GPU 16bpp linear tile must be divisible by 32 pixels in x-direction. VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction
	((PIX_SIZE==1) && ((TILE_TYPE==1) || (TILE_TYPE==2) || (TILE_TYPE==4))) ? 8 : // GPU 16bpp standard and/or super tile must be divisible by 8 pixels in x-direction. VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction
	((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp Luma zenverge tile must be divisible by 8 pixels in x-direction
	
	8); // default case shouldn't be hit
	
	Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte
	Note: for 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation.
	pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte
	
	Third calculate the number of offset pixels
	pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod)
	
	Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers.
	NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset
	
	Note1: If FRAME_1P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND

Field	Function
	FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.33 Frame 1-Plane Pix X Control (FRAME_1P_PIX_X_CTRL_TOG)

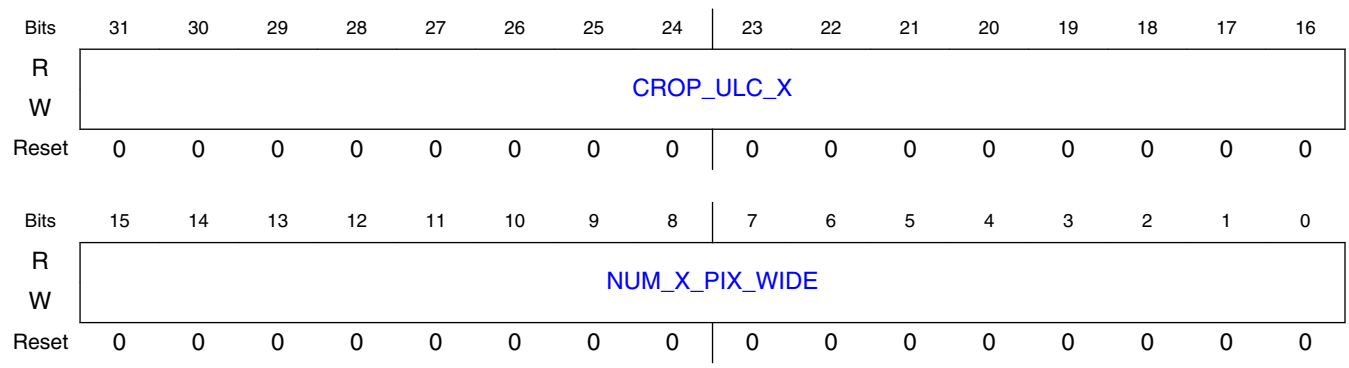
15.7.3.1.33.1 Offset

Register	Offset
FRAME_1P_PIX_X_CTRL_TOG	ACh

15.7.3.1.33.2 Function

The Frame 1P PIX_X Control register controls x-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.33.3 Diagram



15.7.3.1.33.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions

Table continues on the next page...

Field	Function
	apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|
15-0 NUM_X_PIX_WIDE	Number of Pixels Wide in X-direction The number of input frame pixels to fetch in the x-axis direction. In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed. The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied. To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes: First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings: num_pix_x_in_64byte = (((PIX_SIZE==2) && (TILE_TYPE==0)) ? 16 : // GPU 32bpp linear tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==2) && ((TILE_TYPE==1) || (TILE_TYPE==2))) ? 4 : // GPU 32bpp standard and/or super tile must be divisible by 4 pixels in x-direction ((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // GPU 16bpp linear tile must be divisible by 32 pixels in x-direction. VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction ((PIX_SIZE==1) && ((TILE_TYPE==1) || (TILE_TYPE==2) || (TILE_TYPE==4))) ? 8 : // GPU 16bpp standard and/or super tile must be divisible by 8 pixels in x-direction. VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp Luma zenverge tile must be divisible by 8 pixels in x-direction 8); // default case shouldn't be hit Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte Note: for 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation. pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte Third calculate the number of offset pixels pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod)

Field	Function
	Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers. NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset Note1: If FRAME_1P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.34 Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL)

15.7.3.1.34.1 Offset

Register	Offset
FRAME_1P_PIX_Y_CTRL	B0h

15.7.3.1.34.2 Function

The Frame 1P PIX_Y Control register controls y-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.34.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the

Table continues on the next page...

Field	Function
	NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);
	
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers.
	NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.35 Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_SET)

15.7.3.1.35.1 Offset

Register	Offset
FRAME_1P_PIX_Y_CTRL_SET	B4h

15.7.3.1.35.2 Function

The Frame 1P PIX_Y Control register controls y-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.35.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.35.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);

Field	Function
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers. NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.36 Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_CLR)

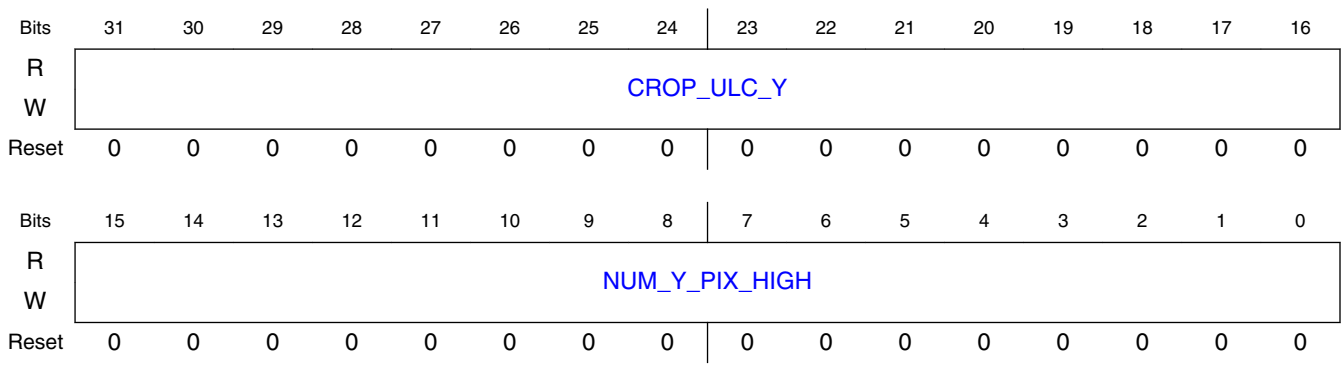
15.7.3.1.36.1 Offset

Register	Offset
FRAME_1P_PIX_Y_CTRL_CLR	B8h

15.7.3.1.36.2 Function

The Frame 1P PIX_Y Control register controls y-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.36.3 Diagram



15.7.3.1.36.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for

Table continues on the next page...

Field	Function
	proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction. In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);
	
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers.
	NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.37 Frame 1-Plane Pix Y Control (FRAME_1P_PIX_Y_CTRL_TOG)

15.7.3.1.37.1 Offset

Register	Offset
FRAME_1P_PIX_Y_CTRL_TOG	BCh

15.7.3.1.37.2 Function

The Frame 1P PIX_Y Control register controls y-axis pixel control for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.37.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (1-Plane or 2-Plane Luma) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);

Field	Function
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers. NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.38 Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0)

15.7.3.1.38.1 Offset

Register	Offset
FRAME_1P_BASE_ADDR_CTRL0	C0h

15.7.3.1.38.2 Function

The Frame 1P Base Address Control0 register is the input frame base address for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.38.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 1-Plane and 2-Plane Luma. The base address is a source frame buffer offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0, pix_y=0 position. If any non-linear GPU tile or VPU tile is being processed, then the base

Field	Function

	address must start at pixel 0 of the tile boundary.
	
	Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position.
	
	1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE.
	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the BASE_ADDR. This is the value which should be written to BASE_ADDR.
	BASE_ADDR = (BASE_ADDR - pix_y_byte_offset) - pix_x_byte_offset;

15.7.3.1.39 Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_SET)

15.7.3.1.39.1 Offset

Register	Offset
FRAME_1P_BASE_ADDR_CTRL0_SET	C4h

15.7.3.1.39.2 Function

The Frame 1P Base Address Control0 register is the input frame base address for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.39.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.39.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 1-Plane and 2-Plane Luma. The base address is a source frame buffer offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0, pix_y=0 position. If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary. Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position. 1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE.

Field	Function

	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the BASE_ADDR. This is the value which should be written to BASE_ADDR.
	BASE_ADDR = (BASE_ADDR - pix_y_byte_offset) - pix_x_byte_offset;

15.7.3.1.40 Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_CLR)

15.7.3.1.40.1 Offset

Register	Offset
FRAME_1P_BASE_ADDR_CTRL0_CLR	C8h

15.7.3.1.40.2 Function

The Frame 1P Base Address Control0 register is the input frame base address for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.40.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 1-Plane and 2-Plane Luma. The base address is a source frame buffer offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0, pix_y=0 position. If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary. Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position. 1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE. 2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.

Field	Function

	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1))); // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the BASE_ADDR. This is the value which should be written to BASE_ADDR.
	BASE_ADDR = (BASE_ADDR - pix_y_byte_offset) - pix_x_byte_offset;

15.7.3.1.41 Frame 1-Plane Base Address Control 0 (FRAME_1P_BASE_ADDR_CTRL0_TOG)

15.7.3.1.41.1 Offset

Register	Offset
FRAME_1P_BASE_ADDR_CTRL0_TOG	CCh

15.7.3.1.41.2 Function

The Frame 1P Base Address Control0 register is the input frame base address for 1-Plane or 2-Plane Luma image processing.

15.7.3.1.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.41.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 1-Plane and 2-Plane Luma. The base address is a source frame buffer offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0, pix_y=0 position. If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary. Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position. 1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE. 2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH. Steps to apply an offset to the BASE_ADDR: First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN. int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN

Field	Function

	&& HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_1P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the BASE_ADDR. This is the value which should be written to BASE_ADDR.
	BASE_ADDR = (BASE_ADDR - pix_y_byte_offset) - pix_x_byte_offset;

15.7.3.1.42 Frame 2-Plane Control 0 (FRAME_2P_CTRL0)

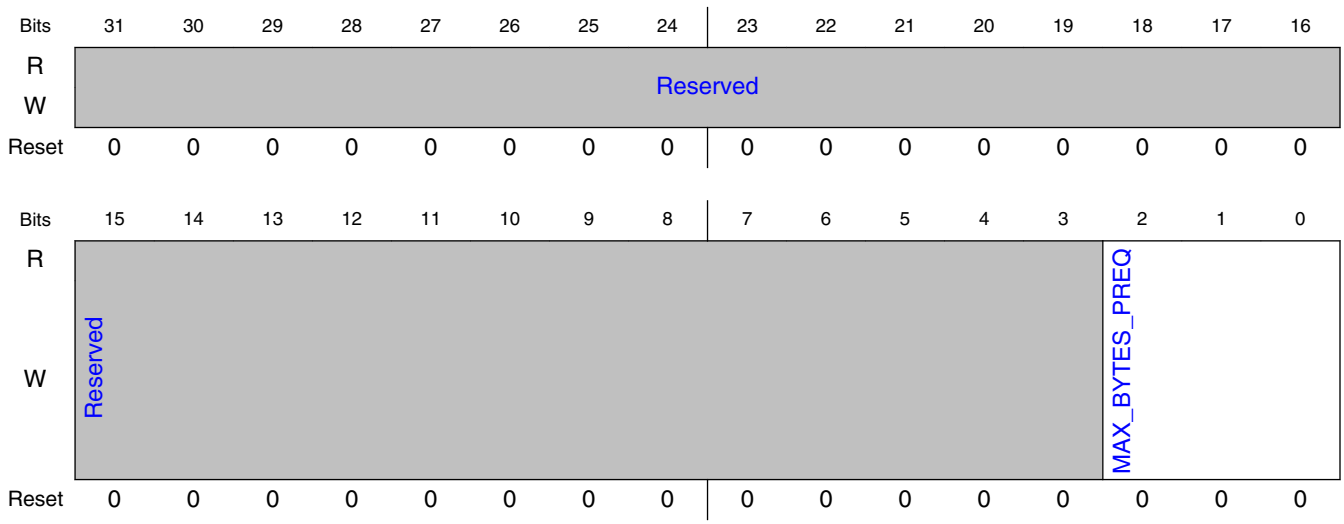
15.7.3.1.42.1 Offset

Register	Offset
FRAME_2P_CTRL0	E0h

15.7.3.1.42.2 Function

The Frame 2P Control0 register controls fields only for 2-Plane Chroma image processing.

15.7.3.1.42.3 Diagram



15.7.3.1.42.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.43 Frame 2-Plane Control 0 (FRAME_2P_CTRL0_SET)

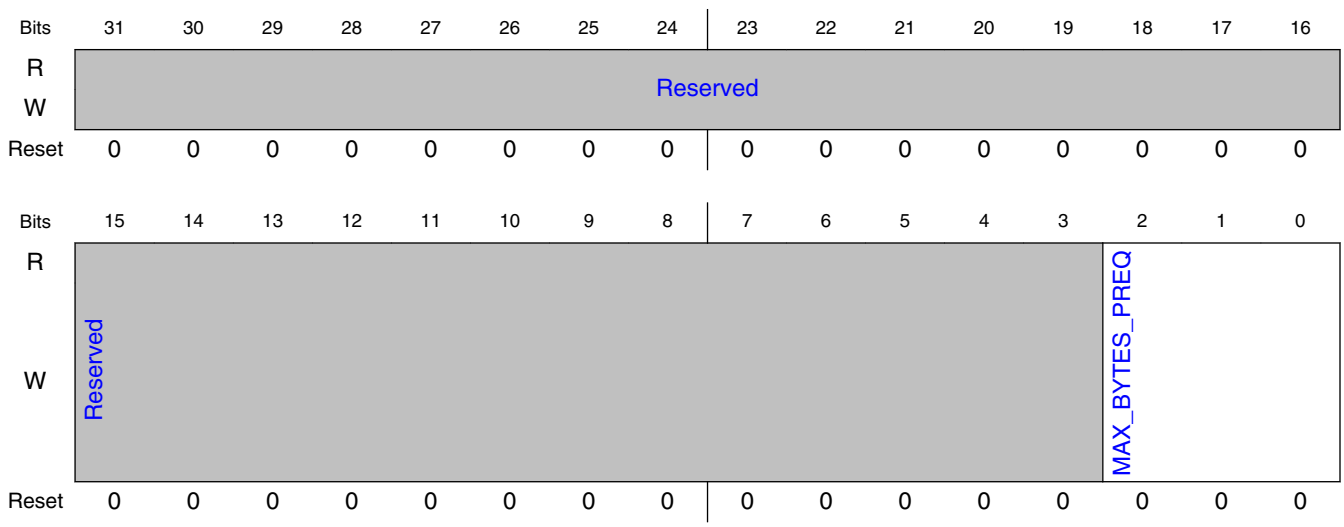
15.7.3.1.43.1 Offset

Register	Offset
FRAME_2P_CTRL0_SET	E4h

15.7.3.1.43.2 Function

The Frame 2P Control0 register controls fields only for 2-Plane Chroma image processing.

15.7.3.1.43.3 Diagram



15.7.3.1.43.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes

Field	Function
	0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.44 Frame 2-Plane Control 0 (FRAME_2P_CTRL0_CLR)

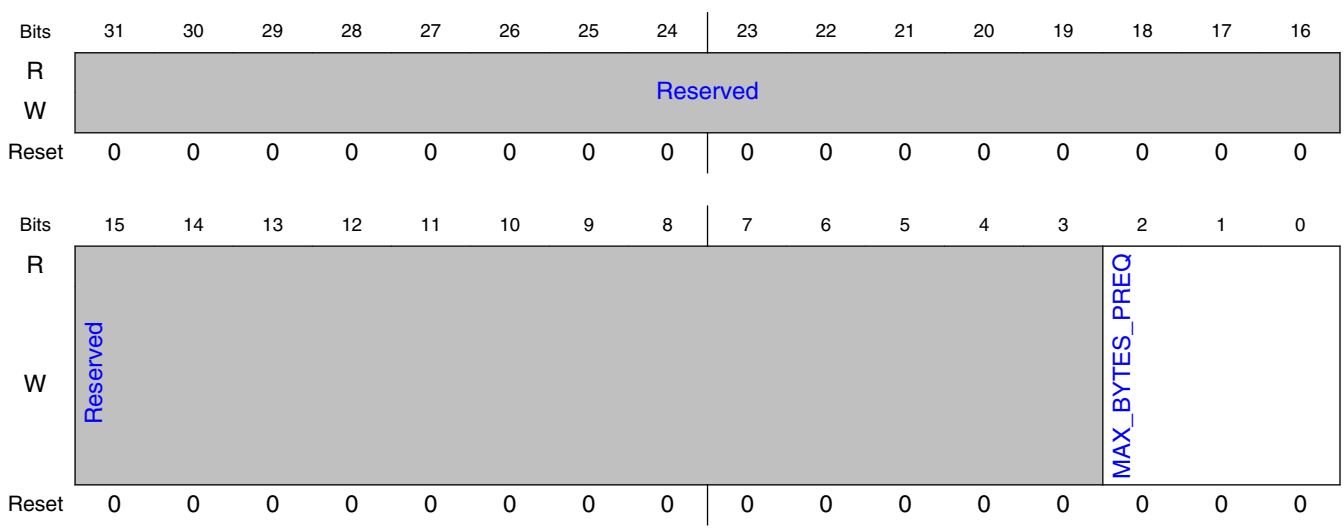
15.7.3.1.44.1 Offset

Register	Offset
FRAME_2P_CTRL0_CLR	E8h

15.7.3.1.44.2 Function

The Frame 2P Control0 register controls fields only for 2-Plane Chroma image processing.

15.7.3.1.44.3 Diagram



15.7.3.1.44.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.45 Frame 2-Plane Control 0 (FRAME_2P_CTRL0_TOG)

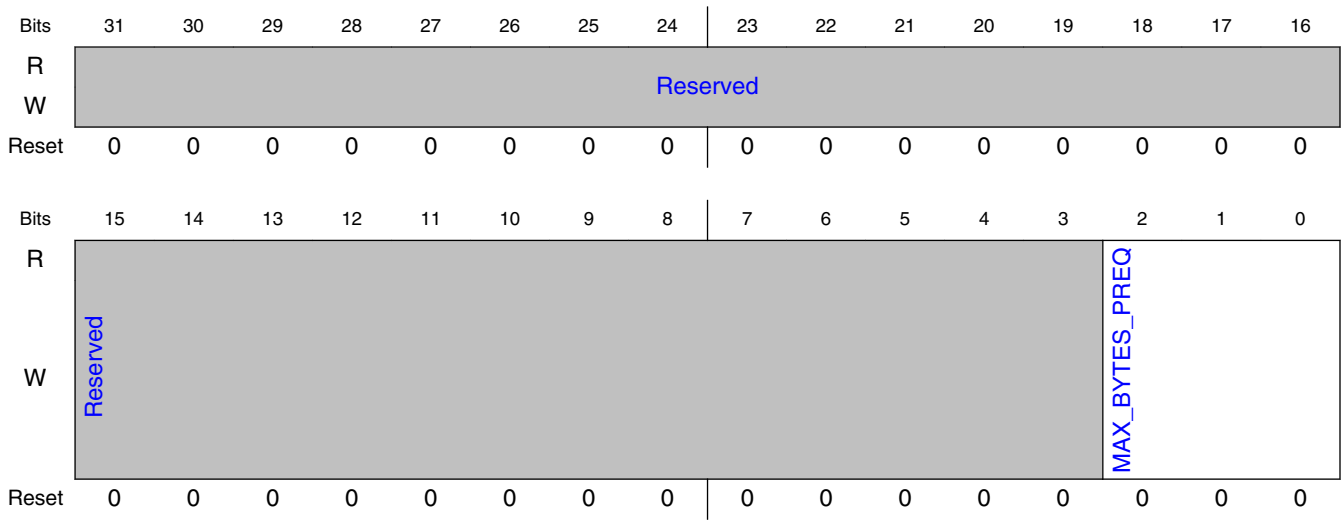
15.7.3.1.45.1 Offset

Register	Offset
FRAME_2P_CTRL0_TOG	ECh

15.7.3.1.45.2 Function

The Frame 2P Control0 register controls fields only for 2-Plane Chroma image processing.

15.7.3.1.45.3 Diagram



15.7.3.1.45.4 Fields

Field	Function
31-3 —	Reserved.
2-0 MAX_BYTES_PREQ	Max Bytes Per Request Maximum number of bytes to prefetch per request. 0x0 = 64 bytes 0x1 = 128 bytes 0x2 = 256 bytes 0x3 = 512 bytes 0x4 = 1K bytes 0x5 = 2K bytes 0x6 = 4K bytes 0x7 = 8K bytes Restrictions based on MODE_CTRL0:TILE_TYPE or FRAME_CTRL0:ROT_ENC • Rotation by 90 or 270: limit to 64 bytes • VPU Tile: limit to 64 bytes • GPU 16bpp super tile: limit to 64 bytes • GPU 32bpp super tile: limit to 128 bytes

15.7.3.1.46 Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL)

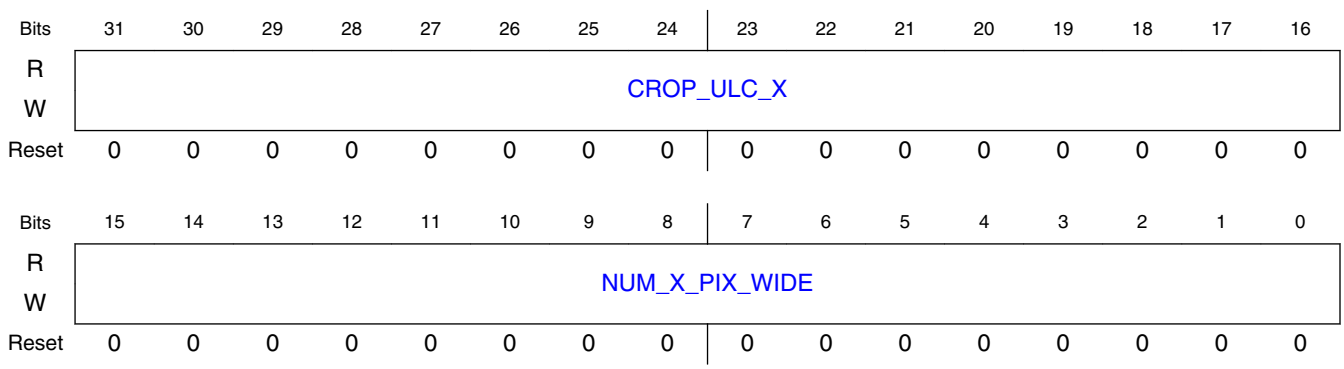
15.7.3.1.46.1 Offset

Register	Offset
FRAME_2P_PIX_X_CTRL	F0h

15.7.3.1.46.2 Function

The Frame 2P PIX_X Control register controls x-axis pixel control for 2-Plane UV image processing.

15.7.3.1.46.3 Diagram



15.7.3.1.46.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 |

Table continues on the next page...

Field	Function
	|------------------------------|--------------------|--------------------|
15-0 NUM_X_PIX_WIDE	Number of Pixels Wide in X-direction The number of input frame pixels to fetch in the x-axis direction. Note: To simplify the logic conversion from pixels to bytes, the U and V components will be calculated as seperate pixels instead of UV as a combined pixel. As and example, in 2PYUV420, the U component will be treated as a 8-bit pixel and the V component as a 2nd 8-bit pixel instead of a single 16-bit pixel. In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed. The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied. To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes: First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings: num_pix_x_in_64byte = (((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction ((PIX_SIZE==1) && (TILE_TYPE==4)) ? 8 : // VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp 2PYUV420 UV zenverge tile must be divisible by 8 pixels in x-direction 8); // default case shouldn't be hit Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte Note: for 2PYUV420 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation. pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte Third calculate the number of offset pixels pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod) Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers. NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset Note1: If FRAME_2P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.47 Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_SET)

15.7.3.1.47.1 Offset

Register	Offset
FRAME_2P_PIX_X_CTRL_SET	F4h

15.7.3.1.47.2 Function

The Frame 2P PIX_X Control register controls x-axis pixel control for 2-Plane UV image processing.

15.7.3.1.47.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.47.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 |

Table continues on the next page...

Field	Function
	| VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|
15-0 NUM_X_PIX_WIDE	Number of Pixels Wide in X-direction The number of input frame pixels to fetch in the x-axis direction. Note: To simplify the logic conversion from pixels to bytes, the U and V components will be calculated as seperate pixels instead of UV as a combined pixel. As and example, in 2PYUV420, the U component will be treated as a 8-bit pixel and the V component as a 2nd 8-bit pixel instead of a single 16-bit pixel. In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed. The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied. To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes: First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings: num_pix_x_in_64byte = (((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction ((PIX_SIZE==1) && (TILE_TYPE==4)) ? 8 : // VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp 2PYUV420 UV zenverge tile must be divisible by 8 pixels in x-direction 8); // default case shouldn't be hit Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte Note: for 2PYUV420 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation. pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte Third calculate the number of offset pixels pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod) Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers. NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset Note1: If FRAME_2P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.48 Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_CLR)

15.7.3.1.48.1 Offset

Register	Offset
FRAME_2P_PIX_X_CTRL_CLR	F8h

15.7.3.1.48.2 Function

The Frame 2P PIX_X Control register controls x-axis pixel control for 2-Plane UV image processing.

15.7.3.1.48.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.48.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 |

Table continues on the next page...

Field	Function
	| VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|
15-0 NUM_X_PIX_WIDE	Number of Pixels Wide in X-direction The number of input frame pixels to fetch in the x-axis direction. Note: To simplify the logic conversion from pixels to bytes, the U and V components will be calculated as seperate pixels instead of UV as a combined pixel. As and example, in 2PYUV420, the U component will be treated as a 8-bit pixel and the V component as a 2nd 8-bit pixel instead of a single 16-bit pixel. In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed. The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied. To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes: First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings: num_pix_x_in_64byte = (((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction ((PIX_SIZE==1) && (TILE_TYPE==4)) ? 8 : // VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp 2PYUV420 UV zenverge tile must be divisible by 8 pixels in x-direction 8); // default case shouldn't be hit Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte Note: for 2PYUV420 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation. pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte Third calculate the number of offset pixels pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod) Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers. NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset Note1: If FRAME_2P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.49 Frame 2-Plane Pix X Control (FRAME_2P_PIX_X_CTRL_TOG)

15.7.3.1.49.1 Offset

Register	Offset
FRAME_2P_PIX_X_CTRL_TOG	FCh

15.7.3.1.49.2 Function

The Frame 2P PIX_X Control register controls x-axis pixel control for 2-Plane UV image processing.

15.7.3.1.49.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_X															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_X_PIX_WIDE															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.49.4 Fields

Field	Function
31-16 CROP_ULC_X	Starting Coordinate of Cropped Image X (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 |

Table continues on the next page...

Field	Function
	| VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|
15-0 NUM_X_PIX_WIDE	Number of Pixels Wide in X-direction The number of input frame pixels to fetch in the x-axis direction. Note: To simplify the logic conversion from pixels to bytes, the U and V components will be calculated as seperate pixels instead of UV as a combined pixel. As and example, in 2PYUV420, the U component will be treated as a 8-bit pixel and the V component as a 2nd 8-bit pixel instead of a single 16-bit pixel. In non-linear (raster) tile mode, the starting pixel must be at the start of a tile boundary. Partial tile fetch is not allowed. The NUM_X_PIX_WIDE must be sized so the prefetch is evenly divisible by 64 bytes. If the NUM_X_PIX_WIDE is not evenly divisible by 64 bytes, then an offset must be applied. To calculate if NUM_X_PIX_WIDE is divisible by 64 bytes: First determine the number of pix_x pixels divisible in 64 bytes, based on MODE_CTRL0: PIX_SIZE and TILE_TYPE settings: num_pix_x_in_64byte = (((PIX_SIZE==1) && (TILE_TYPE==0)) ? 32 : // VPU 10bpp VP9-10 UV linear tile must be divisible by 32 pixels in x-direction ((PIX_SIZE==1) && (TILE_TYPE==4)) ? 8 : // VPU 10bpp VP9-10 UV tile must be divisible by 8 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==0)) ? 64 : // VPU 8bpp VP9 UV linear tile must be divisible by 64 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==4)) ? 16 : // VPU 8bpp VP9 UV tile must be divisible by 16 pixels in x-direction ((PIX_SIZE==0) && (TILE_TYPE==3)) ? 8 : // VPU 8bpp 2PYUV420 UV zenverge tile must be divisible by 8 pixels in x-direction 8); // default case shouldn't be hit Second calculate the modulus of the NUM_X_PIX_WIDE by dpr_num_pix_x_in_64byte Note: for 2PYUV420 10bpp source frame, NUM_X_PIX_WIDE must first be multiplied by 1.25 to account for MODE_CTRL0: PIX_SIZE=8bpp operation. pix_x_div_64byte_mod = NUM_X_PIX_WIDE % num_pix_x_in_64byte Third calculate the number of offset pixels pix_x_offset = (pix_x_div_64byte_mod == 0) ? 0 : (num_pix_x_in_64byte - pix_x_div_64byte_mod) Fourth add pix_x_offset to NUM_X_PIX_WIDE. This is the value written to the NUM_X_PIX_WIDE control registers. NUM_X_PIX_WIDE = NUM_X_PIX_WIDE + pix_x_offset Note1: If FRAME_2P_CTRL0:MAX_BYTES_PREQ > 64 bytes AND FRAME_CTRL0:HFLIP_EN=1, then the prefetch calculation above must be evenly divisible by 128 bytes instead of 64 bytes.

15.7.3.1.50 Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL)

15.7.3.1.50.1 Offset

Register	Offset
FRAME_2P_PIX_Y_CTRL	100h

15.7.3.1.50.2 Function

The Frame 2P PIX_Y Control register controls y-axis pixel control for 2-Plane UV image processing.

15.7.3.1.50.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.50.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 |

Table continues on the next page...

Field	Function
	| VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	Note: The NUM_Y_PIX_HIGH UV height is 1/2 height of Luma from FRAME_1P_PIX_Y_CTRL:NUM_Y_PIX_HIGH before luma height offset was calculated
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);
	
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers.
	NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.51 Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_SET)

15.7.3.1.51.1 Offset

Register	Offset
FRAME_2P_PIX_Y_CTRL_SET	104h

15.7.3.1.51.2 Function

The Frame 2P PIX_Y Control register controls y-axis pixel control for 2-Plane UV image processing.

15.7.3.1.51.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.51.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	Note: The NUM_Y_PIX_HIGH UV height is 1/2 height of Luma from FRAME_1P_PIX_Y_CTRL:NUM_Y_PIX_HIGH before luma height offset was calculated
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;

Field	Function
	Third calculate the number of rows to offset pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod); Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers. NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.52 Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_CLR)

15.7.3.1.52.1 Offset

Register	Offset
FRAME_2P_PIX_Y_CTRL_CLR	108h

15.7.3.1.52.2 Function

The Frame 2P PIX_Y Control register controls y-axis pixel control for 2-Plane UV image processing.

15.7.3.1.52.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.52.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching

Table continues on the next page...

Field	Function
	pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile)|CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	Note: The NUM_Y_PIX_HIGH UV height is 1/2 height of Luma from FRAME_1P_PIX_Y_CTRL:NUM_Y_PIX_HIGH before luma height offset was calculated
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;
	
	Third calculate the number of rows to offset
	pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod);
	
	Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers.
	NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.53 Frame 2-Plane Pix Y Control (FRAME_2P_PIX_Y_CTRL_TOG)

15.7.3.1.53.1 Offset

Register	Offset
FRAME_2P_PIX_Y_CTRL_TOG	10Ch

15.7.3.1.53.2 Function

The Frame 2P PIX_Y Control register controls y-axis pixel control for 2-Plane UV image processing.

15.7.3.1.53.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	CROP_ULC_Y															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	NUM_Y_PIX_HIGH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.53.4 Fields

Field	Function
31-16 CROP_ULC_Y	Starting Coordinate of Cropped Image Y (2-Plane UV) Due to the tile geometry in VPU and Super-Tile formats, a simple address offset cannot be used to crop an image (in all cases). In these modes, the CROP_ULC_X and CROP_ULC_Y coordinate pair will be used to specify an offset to begin fetching pixels within an image. In addition, the NUM_X_PIX_WIDE and NUM_Y_PIX_HIGH will indicate when to stop fetching pixels. Note, these coordinate pairs (CROP_ULC_X,CROP_ULC_Y) and (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH) must be aligned to block boundaries for proper functionality. The following table illustrates coordinate restrictions for the tile types of interest (Note, the same restrictions apply to (NUM_X_PIX_WIDE,NUM_Y_PIX_HIGH)): |---|---|---| | Tile Type |X Alignment |Y Alignment | |---|---|---| | GPU 32bpp (X-Major Super-tile) |CROP_ULC_X[1:0] = 0 |CROP_ULC_Y[1:0] = 0 | | GPU 16bpp xRGB,1-Plane YUV422 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | | VPU 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 8bpp 2-Plane YUV420 |CROP_ULC_X[3:0] = 0 |CROP_ULC_Y[3:0] = 0 | | VP9 10bpp 2-Plane YUV420 |CROP_ULC_X[2:0] = 0 |CROP_ULC_Y[2:0] = 0 | |---|---|---|

15-0
NUM_Y_PIX_HIGH
	Number of Pixels High in Y-direction
	
	The number of input frame pixels to fetch in the y-axis direction.
	In non-linear (raster) t mode, the number of pixels must be sized for a full tile block boundary. Partial tile fetch is not allowed.
	
	The NUM_Y_PIX_HIGH value must be evenly divisible by the number of rows programmed in MODE_CTRL0: RTR_4LINE_BUF_EN.
	First determine the number of rows in the buffer
	num_rows_buf = (RTR_4LINE_BUF_EN==0) ? 8 : 4;
	
	Second calculate the modulus of NUM_Y_PIX_HIGH by the num_rows_buf
	Note: The NUM_Y_PIX_HIGH UV height is 1/2 height of Luma from FRAME_1P_PIX_Y_CTRL:NUM_Y_PIX_HIGH before luma height offset was calculated
	pix_y_mod = NUM_Y_PIX_HIGH % num_rows_buf;

Field	Function
	Third calculate the number of rows to offset pix_y_offset = (pix_y_mod==0) ? 0 : (num_rows_buf - pix_y_mod); Fourth add pix_y_offset to NUM_Y_PIX_HIGH. This is the value written to the NUM_Y_PIX_HIGH control registers. NUM_Y_PIX_HIGH = NUM_Y_PIX_HIGH + pix_y_offset;

15.7.3.1.54 Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0)

15.7.3.1.54.1 Offset

Register	Offset
FRAME_2P_BASE_ADDR_CTRL0	110h

15.7.3.1.54.2 Function

The Frame 2P Base Address Control0 register is the input frame base address for 2-Plane Chroma image processing.

15.7.3.1.54.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.54.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 2-Plane Chroma. The base address is an input frame offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0,

Field	Function

	pix_y=0 position.
	If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary.
	
	Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position.
	
	1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE.
	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the

Field	Function
	BASE_ADDR. This is the value which should be written to BASE_ADDR. Note: the pix_y_byte_offset is divided by 2 because the 2-Plane YUV420 UV height is 1/2 the pix_y_byte_offset calculated for the 2-Plane YUV420 Luma in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH. BASE_ADDR = (BASE_ADDR - (pix_y_byte_offset >> 1)) - pix_x_byte_offset;

15.7.3.1.55 Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_SET)

15.7.3.1.55.1 Offset

Register	Offset
FRAME_2P_BASE_ADDR_CTRL0_SET	114h

15.7.3.1.55.2 Function

The Frame 2P Base Address Control0 register is the input frame base address for 2-Plane Chroma image processing.

15.7.3.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.55.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 2-Plane Chroma. The base address is an input frame offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0,

Field	Function

	pix_y=0 position.
	If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary.
	
	Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position.
	
	1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE.
	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the

Field	Function
	BASE_ADDR. This is the value which should be written to BASE_ADDR. Note: the pix_y_byte_offset is divided by 2 because the 2-Plane YUV420 UV height is 1/2 the pix_y_byte_offset calculated for the 2-Plane YUV420 Luma in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH. BASE_ADDR = (BASE_ADDR - (pix_y_byte_offset >> 1)) - pix_x_byte_offset;

15.7.3.1.56 Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_CLR)

15.7.3.1.56.1 Offset

Register	Offset
FRAME_2P_BASE_ADDR_CTRL0_CLR	118h

15.7.3.1.56.2 Function

The Frame 2P Base Address Control0 register is the input frame base address for 2-Plane Chroma image processing.

15.7.3.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.56.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 2-Plane Chroma. The base address is an input frame offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0,

Field	Function

	pix_y=0 position.
	If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary.
	
	Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position.
	
	1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE.
	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the

Field	Function
	BASE_ADDR. This is the value which should be written to BASE_ADDR. Note: the pix_y_byte_offset is divided by 2 because the 2-Plane YUV420 UV height is 1/2 the pix_y_byte_offset calculated for the 2-Plane YUV420 Luma in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH. BASE_ADDR = (BASE_ADDR - (pix_y_byte_offset >> 1)) - pix_x_byte_offset;

15.7.3.1.57 Frame 2-Plane Base Address Control 0 (FRAME_2P_BASE_ADDR_CTRL0_TOG)

15.7.3.1.57.1 Offset

Register	Offset
FRAME_2P_BASE_ADDR_CTRL0_TOG	11Ch

15.7.3.1.57.2 Function

The Frame 2P Base Address Control0 register is the input frame base address for 2-Plane Chroma image processing.

15.7.3.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.7.3.1.57.4 Fields

Field	Function
31-0 BASE_ADDR	Base Address The base address for 2-Plane Chroma. The base address is an input frame offset to the non-rotated, non-hflip, non-vflip output image. This will correspond to the output image's pix_x=0,

Field	Function

	pix_y=0 position.
	If any non-linear GPU tile or VPU tile is being processed, then the base address must start at pixel 0 of the tile boundary.
	
	Under certain conditions, a offset must be applied to the BASE_ADDR pix_x=0, pix_y=0 position.
	
	1. The DPR will perform an internal hflip AND the pix_x_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE.
	2. The DPR will perform an internal vflip AND the pix_y_offset calculation doesn't equal 0 as described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH.
	
	Steps to apply an offset to the BASE_ADDR:
	
	First determine if the DPR will perform an internal hflip or vflip base on FRAME_CTRL0: ROT_FLIP_ORDER_EN, ROT_ENC, VFLIP_EN and HFLIP_EN.
	int_hflip = (((ROT_ENC==0) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	int_vflip = (((ROT_ENC==0) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN))) || // rot_enc = 0 deg cases.
	((ROT_ENC==2) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN))) || // rot_enc = 180 deg cases.
	((ROT_ENC==1) && ((VFLIP_EN && ~HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 90 deg cases if rotation before v/hflip_en
	((ROT_ENC==1) && ((~VFLIP_EN && HFLIP_EN) || (VFLIP_EN && HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) || // rot_enc = 90 deg cases if rotation after v/hflip_en
	((ROT_ENC==3) && ((~VFLIP_EN && HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==0)) || // rot_enc = 270 deg cases if rotation before v/hflip_en
	((ROT_ENC==3) && ((VFLIP_EN && ~HFLIP_EN) || (~VFLIP_EN && ~HFLIP_EN)) && (ROT_FLIP_ORDER_EN==1)) ; // rot_enc = 270 deg cases if rotation after v/hflip_en
	
	Second calculate the number of pix_x_byte_offset based on the pix_x_offset calculation described in FRAME_2P_PIX_X_CTRL: NUM_X_PIX_WIDE, and the MODE_CTRL0: PIX_SIZE.
	pix_x_byte_offset = int_hflip_en ? (pix_x_offset << PIX_SIZE) : 0;
	
	Third calculate the number of pix_y_byte_offset based on the pix_y_offset calculation described in FRAME_2P_PIX_Y_CTRL: NUM_Y_PIX_HIGH, and the FRAME_CTRL0: PITCH.
	pix_y_byte_offset = int_vflip_en ? (PITCH * pix_y_offset) : 0;
	
	Fourth apply the following pix_x_byte_offset and pix_y_byte_offset to the

Field	Function
	BASE_ADDR. This is the value which should be written to BASE_ADDR. Note: the pix_y_byte_offset is divided by 2 because the 2-Plane YUV420 UV height is 1/2 the pix_y_byte_offset calculated for the 2-Plane YUV420 Luma in FRAME_1P_PIX_Y_CTRL: NUM_Y_PIX_HIGH. BASE_ADDR = (BASE_ADDR - (pix_y_byte_offset >> 1)) - pix_x_byte_offset;

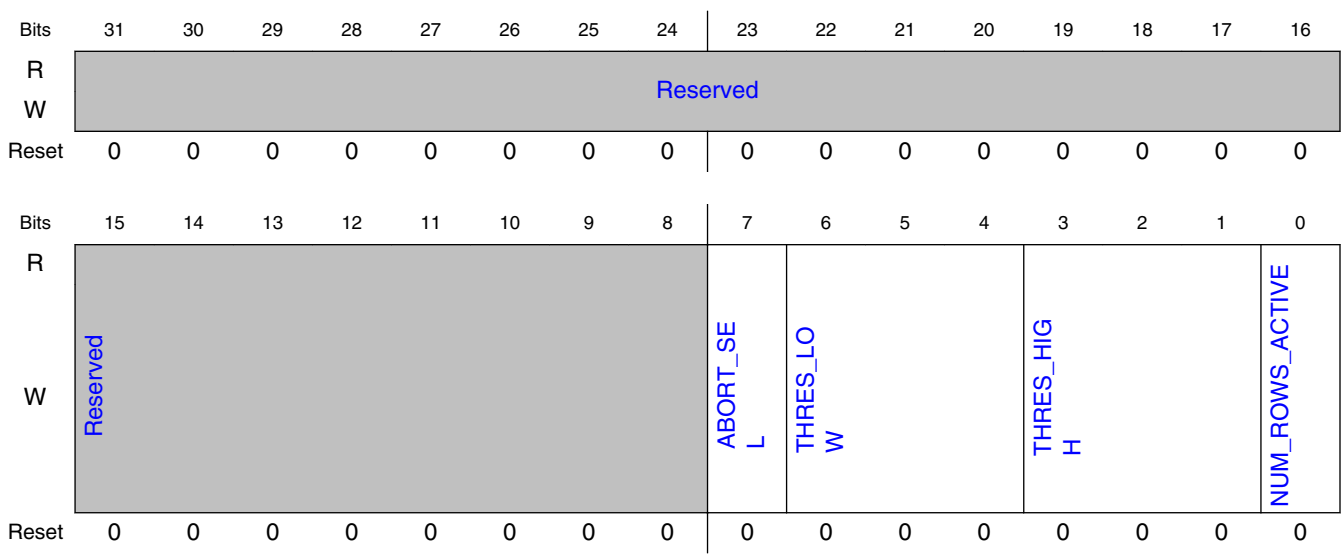
15.7.3.1.58 RTRAM Control 0 (RTRAM_CTRL0)

15.7.3.1.58.1 Offset

Register	Offset
RTRAM_CTRL0	200h

15.7.3.1.58.2 Function

15.7.3.1.58.3 Diagram



15.7.3.1.58.4 Fields

Field	Function
31-8 —	Reserved.

Table continues on the next page...

Field	Function
7 ABORT_SEL	Abort Select Configuration to manage the bank abort or stall logic. 0x0 = STALL Select 0x1 = ABORT Select
6-4 THRES_LOW	Threshold Low Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_LOW is the low water mark.
3-1 THRES_HIGH	Threshold High Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_HIGH is the high water mark.
0 NUM_ROWS_ACTIVE	Number of Rows Active Configuration to manage the number of rows that are enabled to read. 0x0 = Row 0-4 are enabled. 0x1 = Row 0-6, all rows, enabled. NOTE: Configuring MODE_CTRL0: COMP_2PLANE_EN, will enable the UV rows when in 2-plane modes.

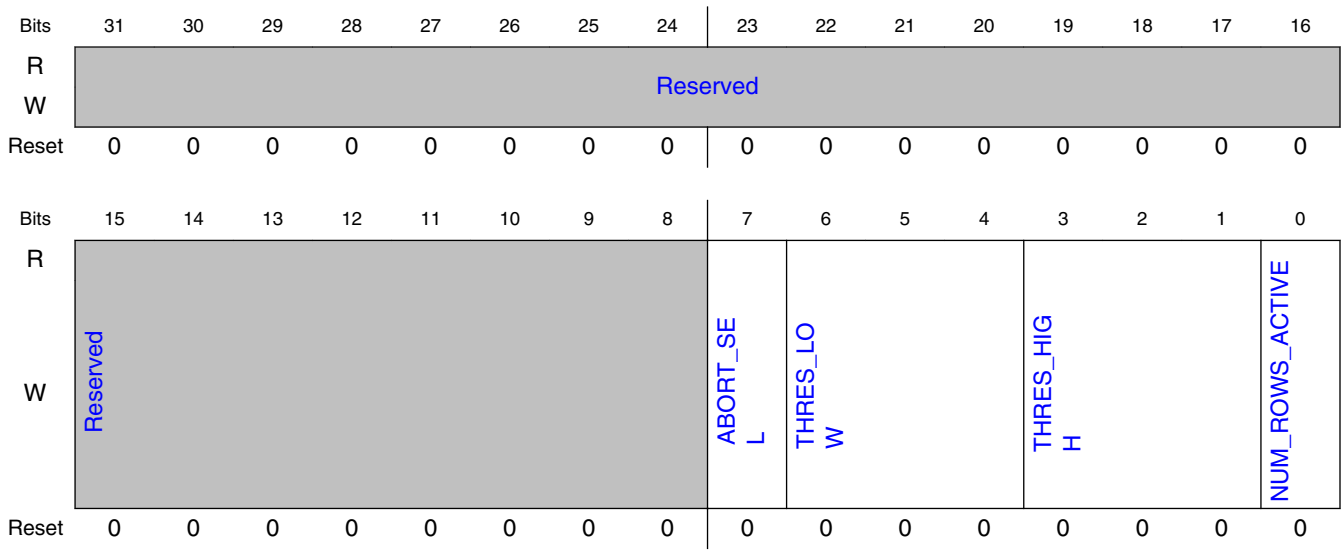
15.7.3.1.59 RTRAM Control 0 (RTRAM_CTRL0_SET)

15.7.3.1.59.1 Offset

Register	Offset
RTRAM_CTRL0_SET	204h

15.7.3.1.59.2 Function

15.7.3.1.59.3 Diagram



15.7.3.1.59.4 Fields

Field	Function
31-8 —	Reserved.
7 ABORT_SEL	Abort Select Configuration to manage the bank abort or stall logic. 0x0 = STALL Select 0x1 = ABORT Select
6-4 THRES_LOW	Threshold Low Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_LOW is the low water mark.
3-1 THRES_HIGH	Threshold High Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_HIGH is the high water mark.
0 NUM_ROWS_ACTIVE	Number of Rows Active Configuration to manage the number of rows that are enabled to read. 0x0 = Row 0-4 are enabled. 0x1 = Row 0-6, all rows, enabled. NOTE: Configuring MODE_CTRL0: COMP_2PLANE_EN, will enable the UV rows when in 2-plane modes.

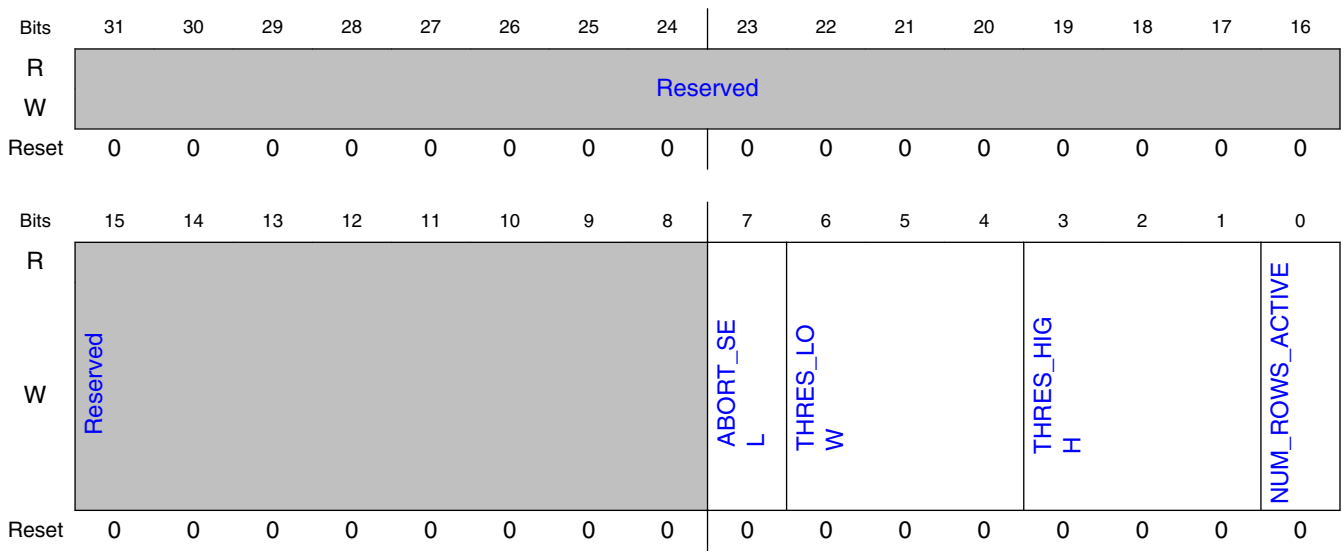
15.7.3.1.60 RTRAM Control 0 (RTRAM_CTRL0_CLR)

15.7.3.1.60.1 Offset

Register	Offset
RTRAM_CTRL0_CLR	208h

15.7.3.1.60.2 Function

15.7.3.1.60.3 Diagram



15.7.3.1.60.4 Fields

Field	Function
31-8 —	Reserved.
7 ABORT_SEL	Abort Select Configuration to manage the bank abort or stall logic. 0x0 = STALL Select 0x1 = ABORT Select
6-4 THRES_LOW	Threshold Low Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_LOW is the low water mark.

Table continues on the next page...

Field	Function
3-1 THRES_HIGH	Threshold High Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_HIGH is the high water mark.
0 NUM_ROWS_ACTIVE	Number of Rows Active Configuration to manage the number of rows that are enabled to read. 0x0 = Row 0-4 are enabled. 0x1 = Row 0-6, all rows, enabled. NOTE: Configuring MODE_CTRL0: COMP_2PLANE_EN, will enable the UV rows when in 2-plane modes.

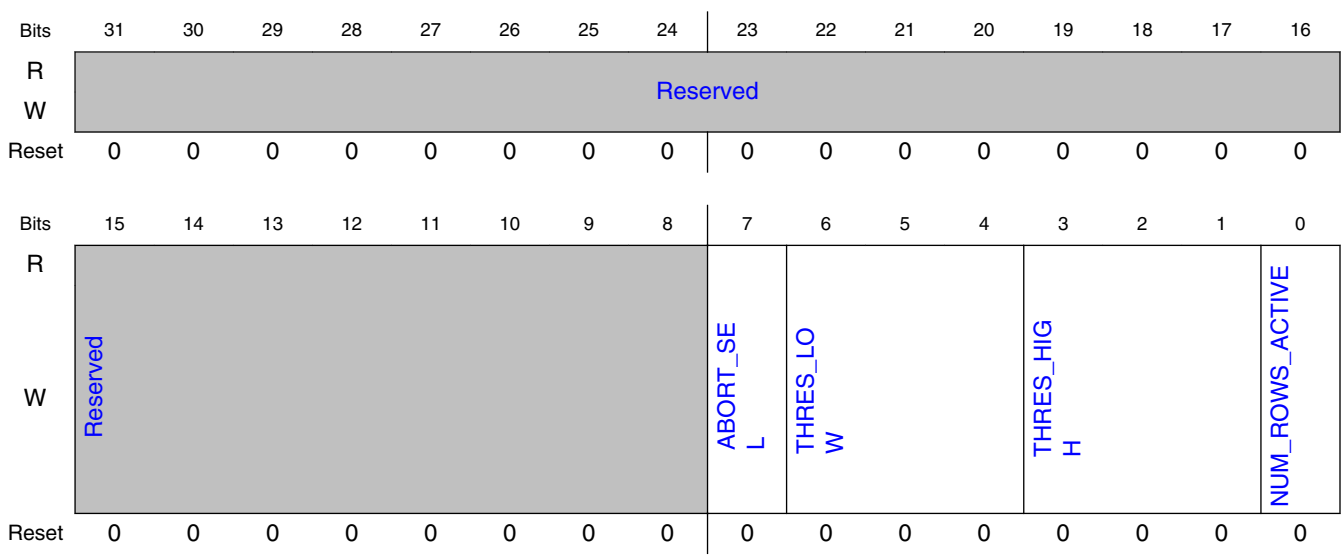
15.7.3.1.61 RTRAM Control 0 (RTRAM_CTRL0_TOG)

15.7.3.1.61.1 Offset

Register	Offset
RTRAM_CTRL0_TOG	20Ch

15.7.3.1.61.2 Function

15.7.3.1.61.3 Diagram



15.7.3.1.61.4 Fields

Field	Function
31-8 —	Reserved.
7 ABORT_SEL	Abort Select Configuration to manage the bank abort or stall logic. 0x0 = STALL Select 0x1 = ABORT Select
6-4 THRES_LOW	Threshold Low Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_LOW is the low water mark.
3-1 THRES_HIGH	Threshold High Configuration to manage the panic state threshold, or hysteresis, detection logic for each channel. The THRES_HIGH is the high water mark.
0 NUM_ROWS_ACTIVE	Number of Rows Active Configuration to manage the number of rows that are enabled to read. 0x0 = Row 0-4 are enabled. 0x1 = Row 0-6, all rows, enabled. NOTE: Configuring MODE_CTRL0: COMP_2PLANE_EN, will enable the UV rows when in 2-plane modes.

15.8 Scaler (MED_DC_SCALE)

15.8.1 Overview

The Scaler's purpose is to resize images for display. Content can be delivered to the application at a resolution that different than the display resolution. Or, images may be reduced in size for PIP viewing. The scaler can resize the images to the required resolution for display. There are three scalers within the DCSS design, one dedicated to each of the three display channels.

15.8.1.1 Block Diagram

The diagram below illustrates the micro architecture of the RTRAM_CTRL, Scaling engine, and scaler output buffer. The example depicts an RTRAM_CTRL 12 row GPU configuration. The VPU configuration uses the same RAMs logically configured as 24 rows with 8 rows per buffer. It is intended that the Scaler, in conjunction with the RTRAM_CTRL, is a cohesive functional unit to resize images on the fly to the display.

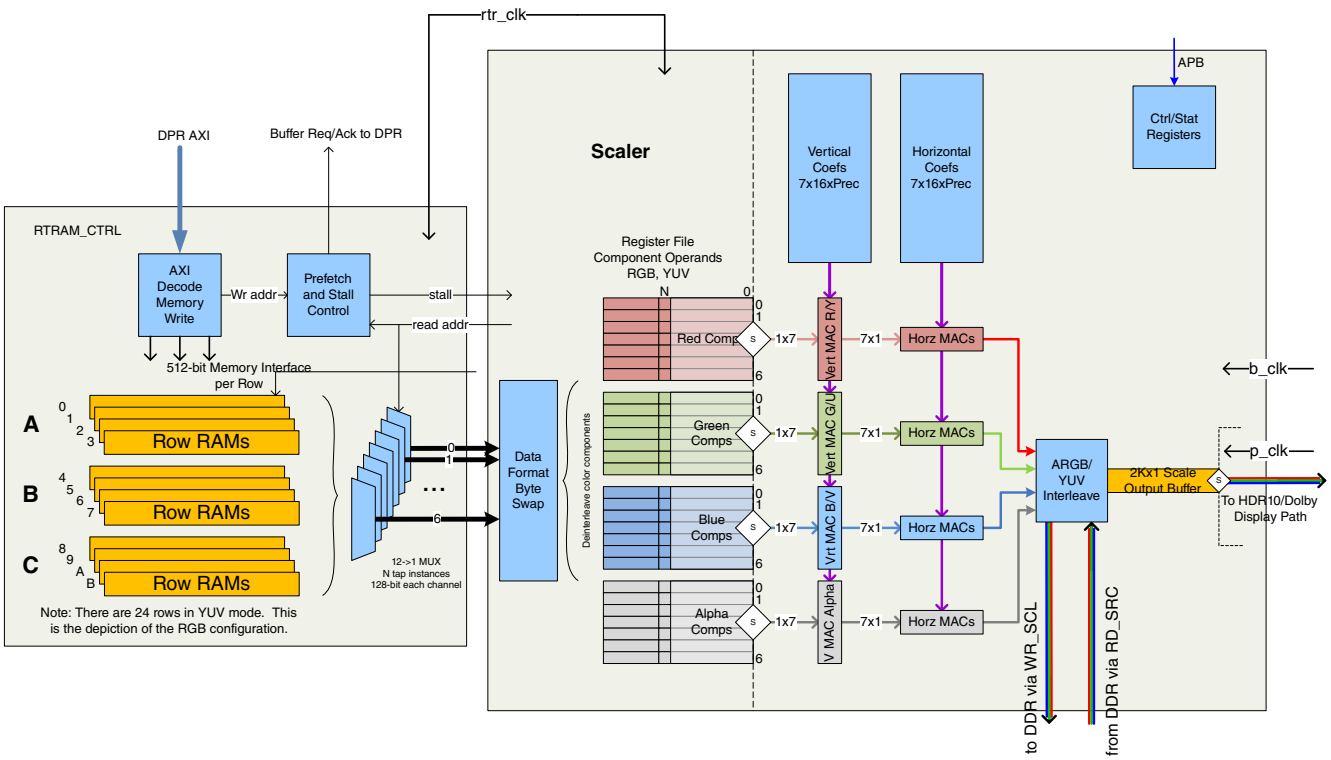


Figure 15-20. Block Diagram

The fundamental components within the Scaler design are as follows:

- Separate vertical and horizontal 7x1 MACs for independent image resizing.
- Separate vertical and horizontal coefficient memories for independent luma/chroma scaling.
- Data formatting and byte swapping to support pixel formats and data endianness.
- Color component FIFOS that contain pixel data used in the resizing of images. These FIFOs bridge between the rtr_clk and b_clk domains.
- Configuration registers to control the scaling configuration.
- Scaler output pixel FIFO to provide elasticity to meet the peak consumption rates of the pixel data path and display. This FIFO bridges between the b_clk and p_clk domains.

The Scaler takes in a block of 7 rows for video and 5 rows for graphics pipe and 128-bit wide data to process the current row. The input pixels go through vertical filtering first and horizontal filtering next to arrive at 1 output pixel. The input is fetched based on the size of the tap filter used. For the edges where the neighbor pixels aren't present, the boundary pixels are replicated. The scaling operation takes place in the raster format i.e. top-left to bottom right till the desired output size is obtained.

Based on the tap filter used (7 or 5), the input rows are filtered vertically giving out 1 pixel per cycle. These vertically filtered pixels are passed on to the horizontal pixels. Once the first 4 vertically filtered pixels are available, the horizontal filter kicks in and generates a scaled pixel output. Subsequently the horizontal tap filter is applied in sliding window manner as the vertical filter keeps giving out vertical filtered pixels till the end of the row. This gives out a row of scaled data and the same is continued till the end of the picture (based on the destination height).

For upscaling >= 1, this results in the output performance of one pixel per clock since there are fewer pixels used at the source to scale images up. For downscaling, the scaling ratio limits the output pixel rate. With a down scale ratio = 1/R, the scaler produces an output pixel per R clock cycles.

The output pixels generated are either sent to the pixel pipe directly or to the DDR through Write Scale interface. If the channel is programmed to send out the data to the memory using the write scale path, it should be programmed to use the data from the DDR via the RD_SRC block to send the data to the pixel pipe. The output format can be either YUV444, YUV422 or ARGB based on the configuration. Only the first channel has the capacity to send out 38 bit output (including 8 bit Alpha component). The output of the scaler is always 10bit. In case of YUV422 each cycle contains only 2 components, i.e. Y0U0, Y1V0 etc.

15.8.1.2 Features

- Upscaling (up to 1:8).
- Downscaling (up to 7:1 for video channels, up to 5:1 for graphics channels). The down scale ratio, 1/R, determines the rate at which the scaler produces output pixels. One pixel for every R cycles determines the down scaler performance.
- Alternate option for the Scaler to write data directly to the DRAM via the WR_SCL bus master interface.
- Alternate option for the Scaler Output FIFO to receive data directly from the DRAM via the RD_SRC bus master interface.
- 13-bit fractional accumulator to approximate fractional scaling ratios.
- Support 8/10-bits per component.
- Inputs formats supported, 8-bit components converted to 10-bit for processing.

  - 1; A8R8G8B8
  - 2; A8Y8U8V8
  - 3; A2R10G10B10
  - 4; A2Y10U10V10
  - 5; Y8U8Y8V8
  - 6; U8Y8V8Y8
  - 7; Y8_U8V8, 2-plane
  - 8; Y10_U10V10, 2-plane
- Output pixel format is 10-bit YUV422/444 (video) and ARGB-8:10:10:10 (graphics)
- 7-tap(video) and 5-tap(graphics) 16-phase FIR filtering
- Software programmable 12 bit filter coefficients in 2s complement
- Independent X/Y cropping

15.8.1.3 Scaler Use Cases

The horizontal and vertical scalers are used to resize source buffers for the following cases:

1. Downscaling for a lower resolution display or for PIP video with a maximum downscale of 7 to 1. The largest allowed 4K PIP window width is 2048 (determined by the scaler output FIFO size in pixels).
2. Upscaling to match the resolution of the display including upscaling HD graphics to 4K graphics with a maximum upscale of 1 to 8.

When the output resolution is UHD, and a UHD source is scaled down to cover less than the entire display, the largest sized PIP within the UHD output window is a HD PIP within the UHD display. This is limited by two factors:

- The Scaler Output FIFO is 2K pixels
- The rate of input into the scaling engine is 1 pixel per clock. So, it takes two clocks to derive one output pixel when scaling down by 2.

The FIFO allows the scaler to pre-compute pixels and store them for when the PIP window is active. This removes any issues with the scaler's performance to produce 1/R (R=scale ratio) pixels per clock. Or, the 2K pixel FIFO has a 2K PIP line worth of data for immediate consumption by the pixel pipeline for PIP display.

Cropping on the Left, Top, Right or Bottom boundaries to project only the area of interest in the incoming video

15.8.2 Functional Description

15.8.2.1 RTRAM Scaler Interface

RTRAM Interface

Scaler gets pixel data from RTRAM in raster format. The request is based on row number and address and RTRAM returns 7 rows of 128bits of data from the requested address and requested data. The row address is always a modulus of the total rows in the bank. If RTRAM operates in 4 lines per bank mode, the total rows is 12 (3 banks in RTRAM), if RTRAM operates in 8 lines per bank, the total rows is 24. For example, a request from ROW 38 and address 0 translates to row_address of 14 in 8 lines per bank mode and row_address of 2 in 4 lines per bank mode. RTRAM responds to the request from Scaler through ack signal. Scaler extends the request till the ack is received. The data from RTRAM is valid 1 cycle after the ack is received. The interface operates at 400MHz clock. In case of 2 plane mode operation i.e YUV420 semi planar, RTRAM has separate request bus to request for Y plane and UV plane. But the data bus is shared. There can be only 1 request at a time.

RTRAM Pixel Data

The data from the RTRAM is arranged in the following format:

1. 32 bpp A8R8G8B8/A8Y8U8V8 - 8bit input
2. 32 bpp A2R10G10B10/A2Y10U10V10 - 10 bit input
3. 16 bpp YUV 422 8 bit
4. 8 bpp Y/UV (2 planes) 420 - 8 bit input
5. 10 bpp Y/UV (2 planes) 420 - 10 bit input

Scaler supports swapping of the components based on the input pixel arrangement. Incase of 32bpp, the pixels can be in the format of ARGB, ARBG, AGBR, RGBA, BRGA etc. The input data is modified to match the scaler processing to form ARGB as the pixel arrangement. Incase of 16bpp, the data can be of the format Y0U0,Y1V0 or U0Y0,V0Y1 etc. Scaler swaps the data internally. All the pixels are in the little endian format from the RTRAM i.e. Pixel15..Pixel0. Scaler swaps it internally to convert to big endian format (pixel0..pixel15)

Incase of 32bpp (ARGB or AYUV), a 5 tap filter is used. RTRAM returns only 5 rows of valid data per request from Scaler. The rows6 and rows7 are not used by Scaler.

Once Scaler is done processing the rows of a bank, bank_done signal is asserted so that RTRAM can fill the next rows in the bank. rtram_vsync_reset is asserted after all the rows of the picture are processed.

15.8.2.2 Scaler Horizontal and Vertical Operation

The scaler, based on the current X,Y position within the source image, operates on the 7x7 matrix of source pixels to produce a single output pixel. Note that for ARGB scaling, a 5x5 matrix of source pixels is used to produce a single output pixel.

For example, for 1:1 scaling, each source pixel is mapped to a destination pixel. For 7:1 downscale, only 1 out of 7 source pixels is mapped to a destination pixel. For 1:4 upscale, the same source group of 7 source pixels is used 4 times at the output but with different filter coefficients. Note that an output buffer is required to support the case of downscaling to manage the rate of BW at the source. In effect, the output buffer allows for the scaling process to spread the input peak BW over the width of the entire display, not just the region that we are displaying, which may not be at the resolution of the display. Of concern are the downscaling cases, fetching large amounts of data, and displaying that data in small regions of the display.

The scaling hardware relies on software to configure the following parameters for the horizontal and vertical scalers:

- 26 bit (13 int, 13 frac) Fractional pixel start
- 17 bit (4 int, 13 frac) Fractional pixel increment (max value is 8)
- 12 bit (12 int) Filter coefficients

The fractional pixel start corresponds to the starting bias, or the first source pixel position to be selected. The start location acts as a top and left crop value since all output pixels before the start location are dropped. The fractional pixel increment gives the scale ratio, or the number of source pixels to ‘skip’ for each output pixel. The fractional pixel start location consists of an integer and fractional part. The integer component is used to skip a selected number of pixels, usually the desired offset into a row of tiles in the X or Y direction. For a 4K image and a 13-bit fractional component, a maximum error of .5 pixel is achieved with a ratio of 4095/4094. The pixel increment is 17-bit in width made up of 4 integer and 13 fractional bits. This is limited to a maximum downscale value of 7 to 1 for video, or 5 to 1 for graphics that matches the filter width so that no pixels are dropped (except pixels outside the crop region). To limit the size of the polyphase filter table, 16 phases are used between source pixels. The phase accumulator keeps accumulating the pixel position based on the pixel_increment. Based on the phase index which is bit [12:9] of the phase accumulated, the pixel position to apply the 7 tap filter is arrived at.

The scaling method is output centric. The input pixels are filtered vertically first and then horizontally. A running output pixel location is maintained using the phase accumulator. Based on the location of the pixel, the corresponding pixels are chosen from the vertically filtered samples. The filter window slides across the vertically filtered samples to generate the row of pixels, and then the process is repeated until the end of the picture.

15.8.2.3 Scaler Filter

For each color component (R/Y, G/U, B/V, A), the scaler computes an output pixel value as a weighted average of 7 (5 for RGBA) source pixel values. The weights, or coefficients, are programmed by software. To produce output pixels of acceptable quality, the selected source pixel's neighbors must figure in the output pixel generation, instead of simply dropping them. It is the job of the filter to include the neighbors in generating the final output pixel value. For each component (R/Y, G/U, B/V, A), the design implements a 7-tap filter (5-tap filter for RGBA), i.e. the selected source pixel, plus its 3 neighboring pixels to either side. Each of the 7 pixels is given a different weight with the coefficients programmed by software. The output pixel is the sum-of-products of the 7 pixels multiplied by the corresponding coefficients. Scaling is performed in the vertical orientation first, followed by the horizontal orientation to determine each final output pixel value.

Polyphase Filter

The source pixel is selected by the accumulated phase from the scaler, rounded up to the next integer. In this way, there is no difference between 2 'real' source pixels with the same integer but different fractional components. To account for the fractional phase difference, a polyphase filter is implemented. The 4-bit fraction allows for 16 phases between 2 integer positions. Instead of using the same set of coefficients for all, each phase is configured with its own coefficient set of 7 coefficients for each component (YUV or 5-tap RGBA), one for each tap. While the current phase's integer component selects the source pixel (and its neighbors), the fractional component selects the coefficient set to operate over the source pixels. As shown in the diagram above, the index into the coefficient table is the accumulated phase's fractional component. The index into the source pixel array is the accumulated phase's integer component.

Boundary Conditions

If a source pixel is less than 3 pixels from the edge, there are an insufficient number of neighboring pixels. In this case, the value of the edge pixel is simply repeated, i.e. the 'missing' neighbors assume the value of the edge pixels.

Filter Operation

The output buffer is sized to handle the worst case downscale case allowed which is the largest width PIP or graphics window of 2048 from 4096 (2 to 1) requiring a buffer that is 2Kx32 bits to support the following output pixel formats:

1. 38 bit RGBA 8 bit Alpha, 10bit pixels per component for RGB.
2. 20 bit YUV 422 10 bit
3. 30 bit YUV 444 10 bit

The Alpha output from the scaler is always 8-bit. For the color components its always 10bit output.

Note that for downscaling 3 to 1, the peak bandwidth goes up by a factor of 3 (but not 9 due to the output buffer).

The scaler output buffer is the data structure used to cross the asynchronous boundary between the b_clk and p_clk domains. This allows the scaler to run at the rate of the system bus interface with no asynchronous boundary crossing.

Scaler Output Buffer with Status Interrupts

The scaler output buffer has a mechanism to detect when underflow could occur. There are three levels of state that determine how full or empty the fifo is at any time. These states are listed and ordered in levels of severity if these states are reached.

- Underflow - a buffer underflow has occurred. This is the most severe state, as the pointers into the write and read interfaces of the buffer may not recover.
- Near Empty - Also known as low threshold. The number of valid pixels in the buffer is less than or equal to this programmable threshold.
- Almost Full – Also known as high threshold. The number of valid pixels in the buffer is less than or equal to this programmable threshold.

The states are described here.

Underflow – The buffer was read while empty. An interrupt is generated and SW restarts the display.

Low Threshold – This is a SW programmable threshold, between 1 and 2048, of pixels that are valid in the FIFO. This value of valid pixels can be reached when the scaler is not able to produce pixels at the rate of consumption by the pixel data path. The SW programmable register that sets this value should be less than or equal to the high threshold.

High Threshold – This is a SW programmable threshold, between 1 and 2048, of pixels that are valid in the FIFO. This is a high threshold of pixels in the buffer and should be set to a value equal or greater than the low threshold.

All states reached can generate an interrupt so that SW can respond by changing system priorities and restarting the display if needed.

15.8.3 Memory Map and Registers

15.8.3.1 register descriptions

15.8.3.1.1 SCALE Memory map

med_dc_scaler base address: 1_C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	Scale Control Register (SCALE_CTRL)	32	RW	0000_0000h
4h	Scale Output FIFO Control Register (SCALE_OFIFO_CTRL)	32	RW	0000_0000h
8h	Scale Source Data Control Register (SCALE_SRC_DATA_CTRL)	32	RW	0000_0000h
Ch	Scale Bit Depth Control Register (SCALE_BIT_DEPTH)	32	RW	0000_0000h
10h	Scale Source Format Control Register (SCALE_SRC_FORMAT)	32	RW	0000_0000h
14h	Scale Destination Format Control Register (SCALE_DST_FORMAT)	32	RW	0000_0000h
18h	Scale Source Luma Resolution Register (SCALE_SRC_LUMA_RES)	32	RW	0000_0000h
1Ch	Scale Source Chroma Resolution Register (SCALE_SRC_CHROMA_RES)	32	RW	0000_0000h
20h	Scale Destination Luma Resolution Register (SCALE_DST_LUMA_RES)	32	RW	0000_0000h
24h	Scale Destination Chroma Resolution Register (SCALE_DST_CHROMA_RES)	32	RW	0000_0000h
48h	Scale Vertical Luma Start Register (SCALE_V_LUMA_START)	32	RW	0000_0000h
4Ch	Scale Vertical Luma Increment Register (SCALE_V_LUMA_INC)	32	RW	0000_0000h
50h	Scale Horizontal Luma Start Register (SCALE_H_LUMA_START)	32	RW	0000_0000h
54h	Scale Horizontal Luma Increment Register (SCALE_H_LUMA_INC)	32	RW	0000_0000h
58h	Scale Vertical Chroma Start Register (SCALE_V_CHROMA_START)	32	RW	0000_0000h
5Ch	Scale Vertical Chroma Increment Register (SCALE_V_CHROMA_INC)	32	RW	0000_0000h
60h	Scale Horizontal Chroma Start Register (SCALE_H_CHROMA_START)	32	RW	0000_0000h
64h	Scale Horizontal Chroma Increment Register (SCALE_H_CHROMA_INC)	32	RW	0000_0000h
80h	Scale Coefficient Memory Array (SCALE_COEF_ARRAY)	32	RW	0000_0000h

15.8.3.1.2 Scale Control Register (SCALE_CTRL)

15.8.3.1.2.1 Offset

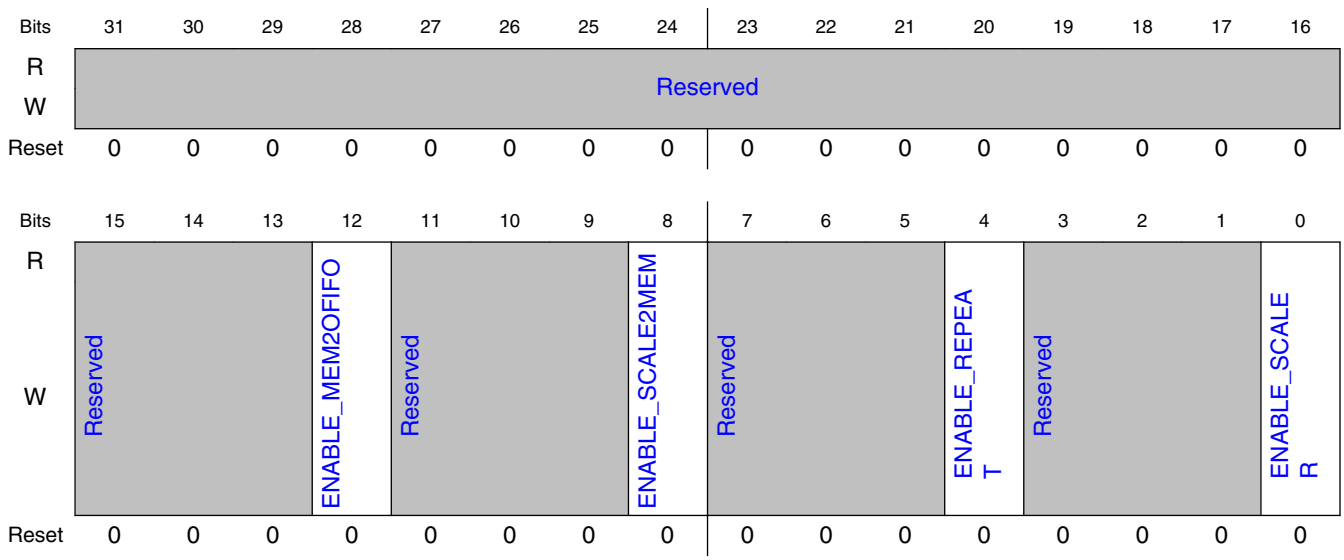
Register	Offset
SCALE_CTRL	0h

15.8.3.1.2.2 Function

This register is the main control register for the scaler.

Note: the bit fields ENABLE_MEM2OFIFO and ENABLE_SCALE2MEM MUST be programmed to the same state, both 1'b0 or both 1'b1.

15.8.3.1.2.3 Diagram



15.8.3.1.2.4 Fields

Field	Function
31-13 —	Reserved.
12 ENABLE_MEM2OFIFO	This bit enables the path from external memory to drive the data into the scale output FIFO. This option is used when the RD_SRC module is enabled to fetch data directly from the memory and bypassing decompression, DPR, and scaling is desired. 0: disabled, scaler drives the data directly into the scale output FIFO. 1: enabled, external pixel interface input bus is used to drive data into the scale output FIFO.
11-9 —	Reserved.

Table continues on the next page...

Field	Function
8 ENABLE_SCALE2MEM	This bit enables the path from the scaler output back to the system memory. This option is used when the WR_SCL module is enabled to write data out to memory. It is intended this path is used when significant down scaling is required to manage peak bandwidth requirements. 0: disabled, the scaler drives data directly into the scale output FIFO. 1: enabled, the scaler drives data out on the pixel data bus to the WR_SCL module.
7-5 —	Reserved.
4 ENABLE_REPEAT	This bit enables the scaler to restart processing a frame buffer without receiving a SW "kick" event using the ENABLE_SCALER control bit. There is a HW event that can start the scaler to process a frame buffer when the ENABLE_REPEAT is high. This event usually occurs after the current active display region is complete. 0: disabled, the scaler will not repeat the processing when the HW repeat event occurs. 1: enabled, the scaler will repeat with the existing programming when the HW "kick" occurs.
3-1 —	Reserved.
0 ENABLE_SCALER	This bit, when set to logic 1, will cause the scaler to begin processing a frame buffer based on all current programmable settings. This bit is cleared after a few cycles, once the start is latched by the HW. If ENABLE_REPEAT is not set to 1, after every frame, this bit has to be programmed to restart the scaler for the next frame. SW can set this bit during the processing of the current frame, but the other parameters shouldn't be changed during frame processing. For predictable results, set ENABLE_SCALER (along with all other programmable parameters) after the scaler is done with processing the current frame. The CTX_LD engine is the best way to determine when to load and enable the scaler considering the tight timing relationship with display retrace. 0: disabled, 1: enabled, the scaler will start processing the frame based on the programmable settings.

15.8.3.1.3 Scale Output FIFO Control Register (SCALE_OFIFO_CTRL)

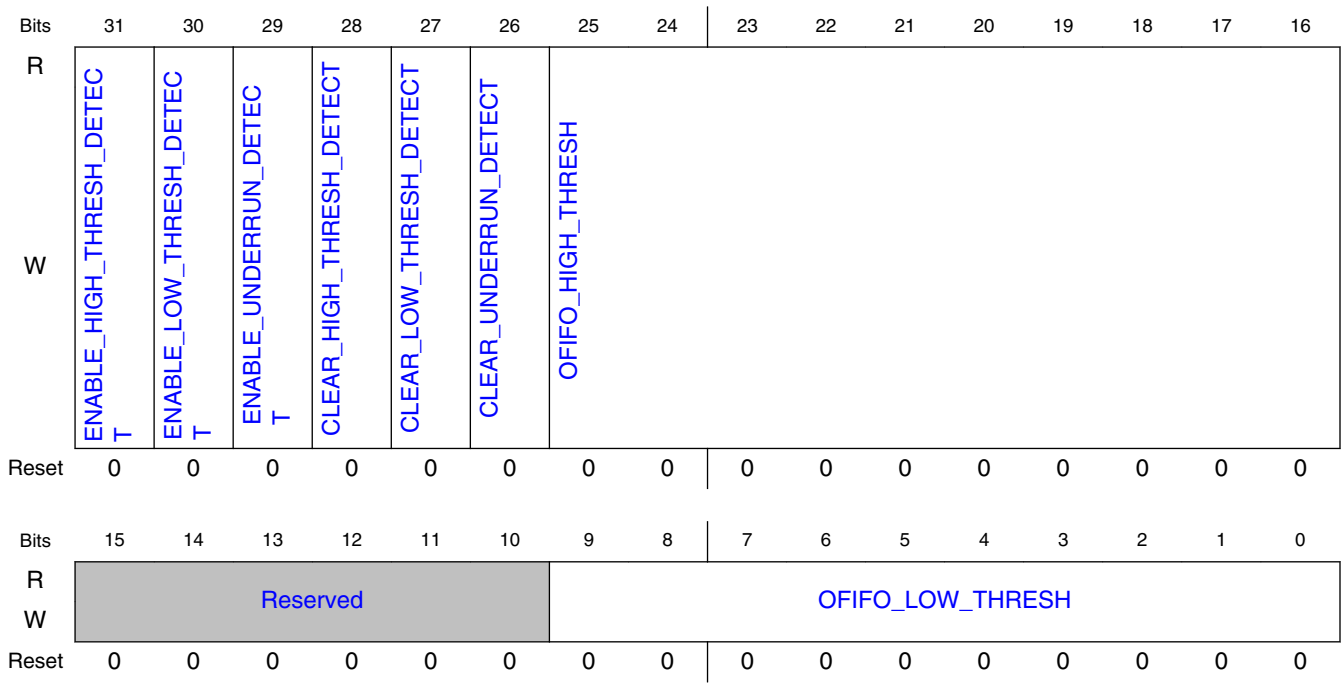
15.8.3.1.3.1 Offset

Register	Offset
SCALE_OFIFO_CTRL	4h

15.8.3.1.3.2 Function

This register controls when the output FIFO will generate an interrupt based on how many pixels are valid within the FIFO and the programmable thresholds.

15.8.3.1.3.3 Diagram



15.8.3.1.3.4 Fields

Field	Function
31 ENABLE_HIGH_THRESH_DETECT	This bit will control if the high threshold detection of the output FIFO can generate an interrupt. 0: disabled, no interrupt will be generated. 1: enabled, interrupt will occur if the high threshold is reached.
30 ENABLE_LOW_THRESH_DETECT	This bit will control if the low threshold detection of the output FIFO can generate an interrupt. 0: disabled, no interrupt will be generated. 1: enabled, interrupt will occur if the low threshold is reached.
29 ENABLE_UNDERRUN_DETECT	This bit will control if the underrun on the output FIFO can generate an interrupt. 0: disabled, no interrupt will be generated. 1: enabled, interrupt will occur if the low threshold is reached.
28 CLEAR_HIGH_THRESH_DETECT	This bit clears the high threshold detect interrupt. 0: disabled, no action. 1: enabled, clear the high threshold reached interrupt.
27 CLEAR_LOW_THRESH_DETECT	This bit clears the low threshold detect interrupt 0: disabled, no action. 1: enabled, clear the low threshold detect interrupt.
26	This bit clears the underrun detected interrupt. 0: disabled, no action.

Table continues on the next page...

Field	Function
CLEAR_UNDERRUN_DETECT	1: enabled, underrun detected interrupt is cleared.
25-16 OFIFO_HIGH_THRESH	This value is used to detect when the output FIFO has a number of pixels that are valid that first exceeds this value, and then decrements to a value equal to the value programmed in this register. Note: This value should be greater than the value programmed into OFIFO_LOW_THRESH.
15-10 —	Reserved.
9-0 OFIFO_LOW_THRESH	This value is used to detect when the output FIFO has a number of pixels that are valid that first exceeds this value, and then decrements to a value equal to the value programmed in this register. Note: This value should be less than the value programmed into OFIFO_HIGH_THRESH.

15.8.3.1.4 Scale Source Data Control Register (SCALE_SRC_DATA_CTRL)

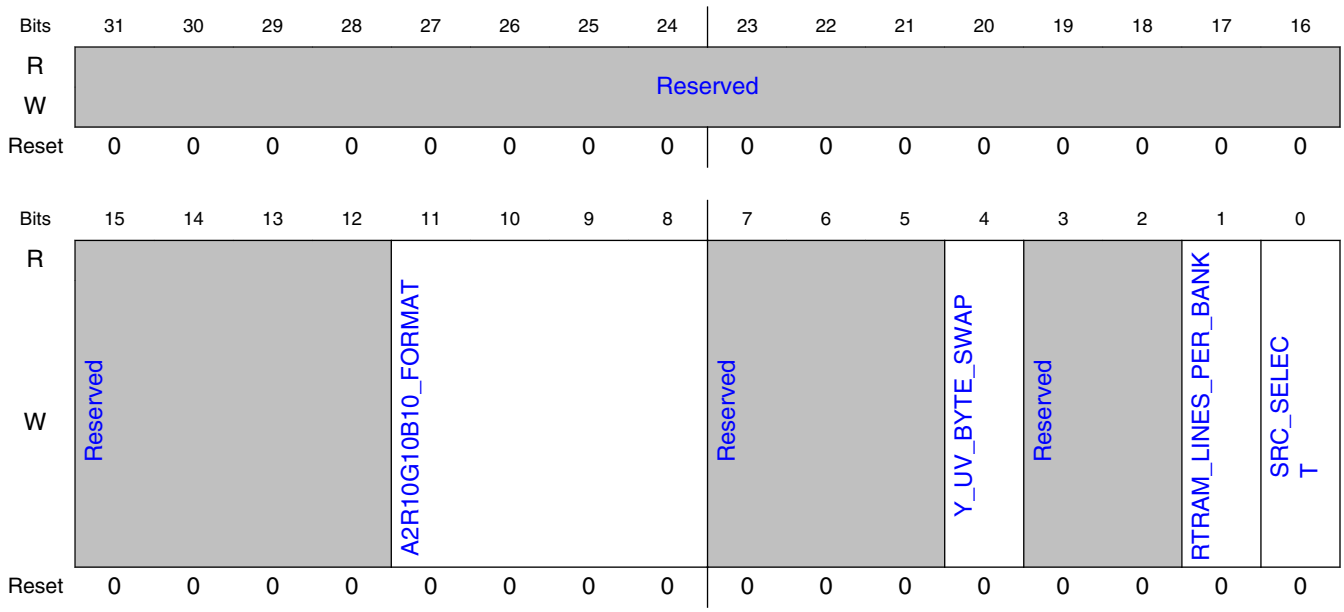
15.8.3.1.4.1 Offset

Register	Offset
SCALE_SRC_DATA_CTRL	8h

15.8.3.1.4.2 Function

This register control aspects related to data formats in the scaler. It also controls HW resources that are used based on the different modes that are programmed.

15.8.3.1.4.3 Diagram



15.8.3.1.4.4 Fields

Field	Function
31-12 —	Reserved.
11-8 A2R10G10B10_FORMAT	This field represents the data arrangement in the memory in case 10 RGB mode. 0: ARGB is the format in 32bit data. A[31:30], R[29:20], G[19:10], B[9:0] 1: ARBG is the format in 32bit data. A[31:30], R[29:20], B[19:10], G[9:0] 2: AGRB is the format in 32bit data. A[31:30], G[29:20], R[19:10], B[9:0] 3: AGBR is the format in 32bit data. A[31:30], R[29:20], B[19:10], R[9:0] 4: ABRG is the format in 32bit data. A[31:30], B[29:20], R[19:10], G[9:0] 5: ABGR is the format in 32bit data. A[31:30], B[29:20], G[19:10], R[9:0] 6: RGBA is the format in 32bit data. R[31:22], G[21:12], B[11: 2], A[1:0] 7: RBGA is the format in 32bit data. R[31:22], B[21:12], G[11: 2], A[1:0] 8: GRBA is the format in 32bit data. G[31:22], R[21:12], B[11: 2], A[1:0] 9: GBRA is the format in 32bit data. G[31:22], B[21:12], R[11: 2], A[1:0] 10: BRGA is the format in 32bit data. B[31:22], R[21:12], G[11: 2], A[1:0] 11: BGRA is the format in 32bit data. B[31:22], G[21:12], R[11: 2], A[1:0]
7-5 —	Reserved.
4 Y_UV_BYTE_SWAP	This bit will control the swapping of alternate bytes in the incoming data word. 0: disabled, no byte steering. 1: enabled, each pair of bytes is swapped. This option can be used to achieve UYVY -> YUYV swapping of bytes.
3-2 —	Reserved.

Table continues on the next page...

Field	Function
1 RTRAM_LINES_PER_BANK	This field determines the number of lines that are used in each bank of the RTRAM_CTRL module. This should match the respective setting of lines per bank in the DPR. 0: RTRAM has 4 lines per bank. 1: RTRAM has 8 lines per bank.
0 SRC_SELECT	This bit controls whether the source is Video format(YUV) or Graphics(RGB). AYUV444 operates similar to ARGB, hence the SRC_SELECT bit has to be set to 0. However to work around a HW bug in YUV444 to YUV422, the bit has to be programmed to 0. The below are the scenarios and the corresponding programming ARGB input: SRC_SELECT should be set to 0 AYUV444 input to YUV444 output: SRC_SELECT should be set to 1 AYUV444 input to YUV422 output: SRC_SELECT should be set to 0 AYUV422 input to Any output: SRC_SELECT should be set to 1 AYUV420 input to Any output: SRC_SELECT should be set to 1 In particular, it determines if the chroma coefficients, start value, and increment value in HW are used. If the input source is an RGB format, the Luma coefficients, start, and increment values are used, and the chroma parameters are ignored (programming not needed). If the input source is any YUV format, both the Luma and Chroma parameters need to be programmed. Programming of the Luma/Chroma parameters start/inc need to be congruent and consistent with the input format. For example: YUV444: requires Luma/Chroma start/inc values to be identical. YUV422/YUV420: horizontal/vertical Chroma start/inc values are independent of Luma values The values in this register define the source format as: 0: Input source has ARGB or AYUV444 components. 1: Input source has YUV components.

15.8.3.1.5 Scale Bit Depth Control Register (SCALE_BIT_DEPTH)

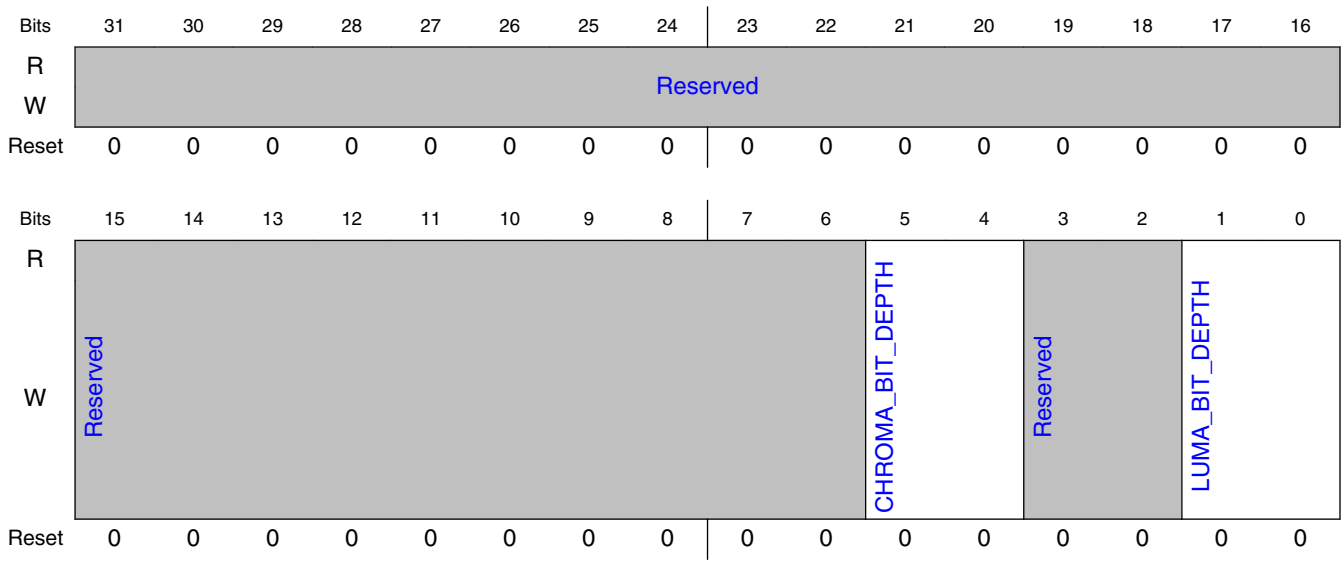
15.8.3.1.5.1 Offset

Register	Offset
SCALE_BIT_DEPTH	Ch

15.8.3.1.5.2 Function

This register determines the component bit depth of the source frame buffer.

15.8.3.1.5.3 Diagram



15.8.3.1.5.4 Fields

Field	Function
31-6 —	Reserved.
5-4 CHROMA_BIT_DEPTH	These bits determine the bit depth of the alternate component (UV 2-plane) processing. 0: 8-bit. 2: 10-bit. other: reserved Both Chroma and Luma bit depths MUST be programmed with the same value.
3-2 —	Reserved.
1-0 LUMA_BIT_DEPTH	These bits determine the bit depth of the primary component (Y, RGB) processing. 0: 8-bit. 2: 10-bit. other: reserved Both Chroma and Luma bit depths MUST be programmed with the same value.

15.8.3.1.6 Scale Source Format Control Register (SCALE_SRC_FORMAT)

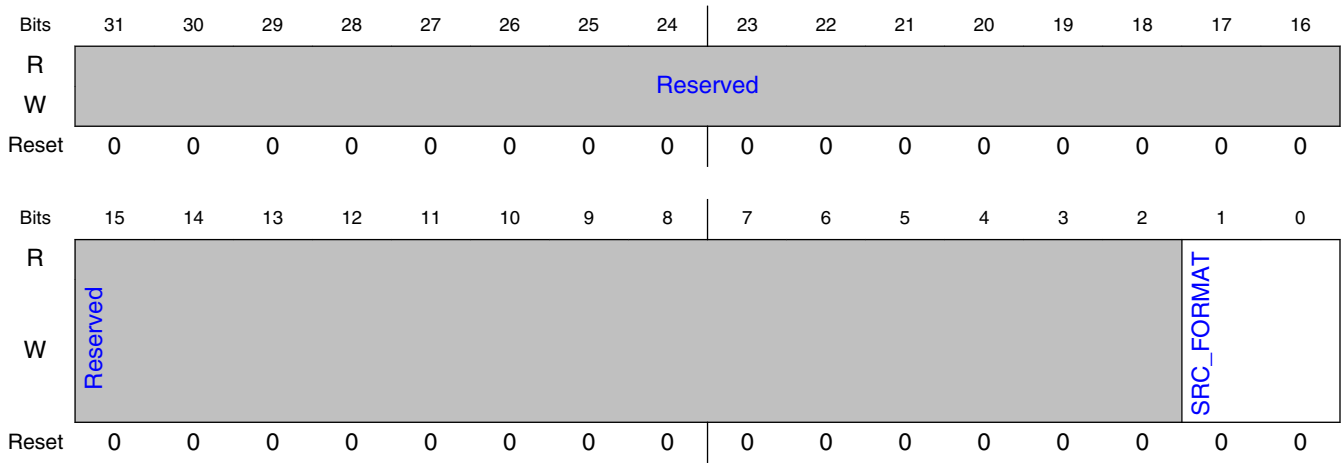
15.8.3.1.6.1 Offset

Register	Offset
SCALE_SRC_FORMAT	10h

15.8.3.1.6.2 Function

This register controls the format of the source frame buffer.

15.8.3.1.6.3 Diagram



15.8.3.1.6.4 Fields

Field	Function
31-2 —	Reserved.
1-0 SRC_FORMAT	These bits define the input buffer format. 0: YUV420 (2-plane chroma sub-sampled V/H). 1: YUV422 (1-plane chroma sub-sampled H). 2: ARGB8888/YUV444 (1-plane full color). Other: Reserved. This parameter needs to be congruent with the SRC_SELECT programming.

15.8.3.1.7 Scale Destination Format Control Register (SCALE_DST_FORMAT)

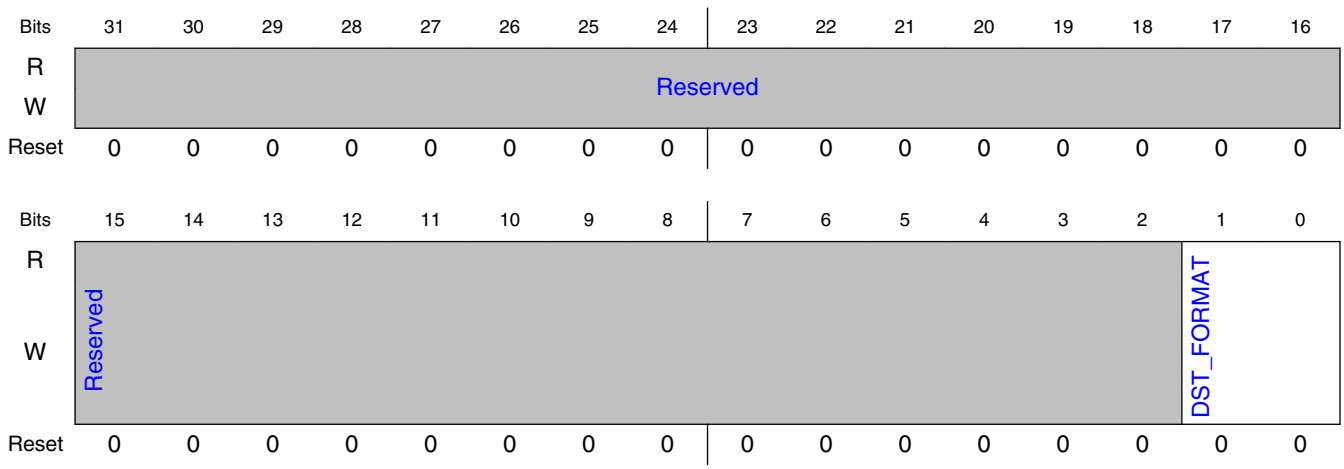
15.8.3.1.7.1 Offset

Register	Offset
SCALE_DST_FORMAT	14h

15.8.3.1.7.2 Function

This register controls the format of the scaler output.

15.8.3.1.7.3 Diagram



15.8.3.1.7.4 Fields

Field	Function
31-2 —	Reserved.
1-0 DST_FORMAT	These bits define the scaler output format. 1: YUV422. 2: RGB888/YUV444. Other: Reserved.

15.8.3.1.8 Scale Source Luma Resolution Register (SCALE_SRC_LUMA_RES)

15.8.3.1.8.1 Offset

Register	Offset
SCALE_SRC_LUMA_RES	18h

15.8.3.1.8.2 Function

This register defines the resolution of the source image. The source image resolution needs to be congruent with the resolution programmed in the DPR to guarantee that all data prefetched by the DPR is consumed by the scaler engine. If the resolution programmed into the DPR is not congruent, scaler and DCSS operation will be unpredictable. This restriction pertains to all source buffers formats, either raster or tiled, and with any RTRAM bank size configuration, either 4 or 8 lines per bank.

15.8.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				HEIGHT											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				WIDTH											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.8.4 Fields

Field	Function
31-28 —	Reserved.
27-16 HEIGHT	This field defines the height of the source image in pixels minus 1. Incase of cropping, the height should be programmed removing the bottom lines to be cropped. The top lines cropped will be conveyed through vs_luma_start programming. Programmed height = Actual_height - bottom_crop - 1.
15-12 —	Reserved.
11-0 WIDTH	This field defines the width of the source image in pixels minus 1. Incase of cropping, the width should be programmed removing the right pixels to be cropped. The left pixels to be cropped will be conveyed through hs_luma_start programming. programmed width = actual_width - right_crop - 1.

15.8.3.1.9 Scale Source Chroma Resolution Register (SCALE_SRC_CHROMA_RES)

15.8.3.1.9.1 Offset

Register	Offset
SCALE_SRC_CHROMA_RES	1Ch

15.8.3.1.9.2 Function

This register defines the resolution of the source image. The value of the chroma resolution is dependent on the resolution of the source image programmed in the SCALE_SRC_LUMA_RES register, since the luma and chroma scaling channels are separate. The programming restrictions are described below.

ARGB8888/YUV444: SCALE_SRC_CHROMA_RES = SCALE_SRC_LUMA_RES, SCALE_SRC_FORMAT.SRC_FORMAT = 444
YUV422: SCALE_SRC_CHROMA_RES.BUFFER_WIDTH = SCALE_SRC_LUMA_RES.BUFFER_WIDTH/2, SCALE_SRC_CHROMA_RES.BUFFER_HEIGHT = SCALE_SRC_LUMA_RES.BUFFER_HEIGHT, SCALE_SRC_FORMAT.SRC_FORMAT = 422
YUV420: SCALE_SRC_CHROMA_RES.BUFFER_WIDTH = SCALE_SRC_LUMA_RES.BUFFER_WIDTH/2, SCALE_SRC_CHROMA_RES.BUFFER_HEIGHT = SCALE_SRC_LUMA_RES.BUFFER_HEIGHT/2, SCALE_SRC_FORMAT.SRC_FORMAT = 420

15.8.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				HEIGHT											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				WIDTH											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.9.4 Fields

Field	Function
31-28 —	Reserved.
27-16 HEIGHT	This field defines the height of the source image in pixels minus 1. Incase of cropping, the height should be programmed removing the bottom lines to be cropped. The top lines cropped will be conveyed through vs_chroma_start programming. Programmed height = Actual_height - bottom_crop - 1.
15-12 —	Reserved.
11-0 WIDTH	This field defines the width of the source image in pixels minus 1. Incase of cropping, the width should be programmed removing the right pixels to be cropped. The left pixels to be cropped will be conveyed through hs_chroma_start programming. programmed width = actual_width - right_crop - 1.

15.8.3.1.10 Scale Destination Luma Resolution Register (SCALE_DST_LUMA_RES)

15.8.3.1.10.1 Offset

Register	Offset
SCALE_DST_LUMA_RES	20h

15.8.3.1.10.2 Function

This register defines the resolution of the scaled result, or destination image. The destination image resolution needs to be congruent with the source image resolution and the scaling factors applied. For example, if the source image is scaled down by a factor of 2, the destination resolution needs to be set to exactly half of the resolution of the source image. Fractional scaling ratios, cropping, and initial offsets into the source image all need to be congruent with the source and destination resolution values.

15.8.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				HEIGHT											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				WIDTH											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.10.4 Fields

Field	Function
31-28 —	Reserved.
27-16 HEIGHT	This field defines the height of the destination image in pixels minus 1.
15-12 —	Reserved.
11-0 WIDTH	This field defines the width of the destination image in pixels minus 1.

15.8.3.1.11 Scale Destination Chroma Resolution Register (SCALE_DST_CHROMA_RES)

15.8.3.1.11.1 Offset

Register	Offset
SCALE_DST_CHROMA_RES	24h

15.8.3.1.11.2 Function

This register defines the resolution of the scaled result, or destination image. The height field is not required for this register. Similarly to the destination luma register, this register needs to be congruent with all other scaler register settings.

15.8.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				WIDTH											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.11.4 Fields

Field	Function
31-12 —	Reserved.
11-0 WIDTH	This field defines the width of the destination image in pixels minus 1.

15.8.3.1.12 Scale Vertical Luma Start Register (SCALE_V_LUMA_START)

15.8.3.1.12.1 Offset

Register	Offset
SCALE_V_LUMA_START	48h

15.8.3.1.12.2 Function

This register contains the initial vertical pixel start, or offset, into the image that is fetched by the prefetch engine. Essentially it is used to discard pixels (integer component) and provide and initial phase (fractional component) into the image in RTRAM. Fundamentally, the images loaded into the RTRAM from the DPR are aligned to 64 byte addresses. This register is used for cropping the top rows from the source image. To achieve better DRAM efficiency, DPR should be used to not fetch the unwanted lines in the top of the picture. The remaining rows i.e. the rows that cannot be dropped by DPR can be cropped in Scaler. There is no limit on the number of lines that can be cropped in

the Scaler. If it crosses the bank boundary, Scaler gives out bank done without reading the bank data and this continues till the actual row is needed. If a few lines from the top are cropped, the source image pixels beyond the crop boundary aren't considered for scaling. Its treated as if they are outside the picture

15.8.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved								V_START							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.12.4 Fields

Field	Function
31-26 —	Reserved.
25-0 V_START	This register contains 13 integer and 13 fractional bits to define the vertical offset into the start of the prefetched image. It is a continuous 16 bit value with the following format ####_####_####.#_####_####_####. The start can be negative but greater than -1

15.8.3.1.13 Scale Vertical Luma Increment Register (SCALE_V_LUMA_INC)

15.8.3.1.13.1 Offset

Register	Offset
SCALE_V_LUMA_INC	4Ch

15.8.3.1.13.2 Function

This register contains the increment which is used to define the scaling ratio. The upper 4 bits is the integer component of the step value. The lowest 13 bits is the fractional component.

15.8.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved												V_INC			
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	V_INC															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.13.4 Fields

Field	Function
31-20 —	Reserved.
19-0 V_INC	Vertical increment value used for scaling the image. This register is programmed with the following format: ####.#_####_####_####. The maximum value that can be programmed into this register is 0xE000, for a maximum scale down ratio of 1/7. Other programming examples could be: 0x1000, scale up by 2. 0x0800, scale up by 4. 0x6000, scale down by 3. 0x0AAB, scale up by 3, digital approximation of 1/3.

15.8.3.1.14 Scale Horizontal Luma Start Register (SCALE_H_LUMA_STARTRT)

15.8.3.1.14.1 Offset

Register	Offset
SCALE_H_LUMA_STARTRT	50h

15.8.3.1.14.2 Function

This register contains the initial horizontal pixel start, or offset, into the image that is fetched by the prefetch engine. Essentially it is used to disgard pixels (integer component) and provide and initial phase (fractional component) into the image in RTRAM. Fundamentally, the images loaded into the RTRAM from the DPR are aligned to 64 byte addresses. This is essential to achieve DRAM efficiency.

The integer component of this register is used to skip input pixels in the horizontal direction.

This register may use a fractional component to establish an initial horizontal phase offset.

15.8.3.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved								H_START							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	H_START															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.14.4 Fields

Field	Function
31-26 —	Reserved.
25-0 H_START	This register contains 13 integer and 13 fractional bits to define the horizontal offset into the start of the prefetched image. It is a continuous 19 bit value with the following format ####_####_####.#_####_####_####. The start can be negative but greater than -1

15.8.3.1.15 Scale Horizontal Luma Increment Register (SCALE_H_LUMA_INC)

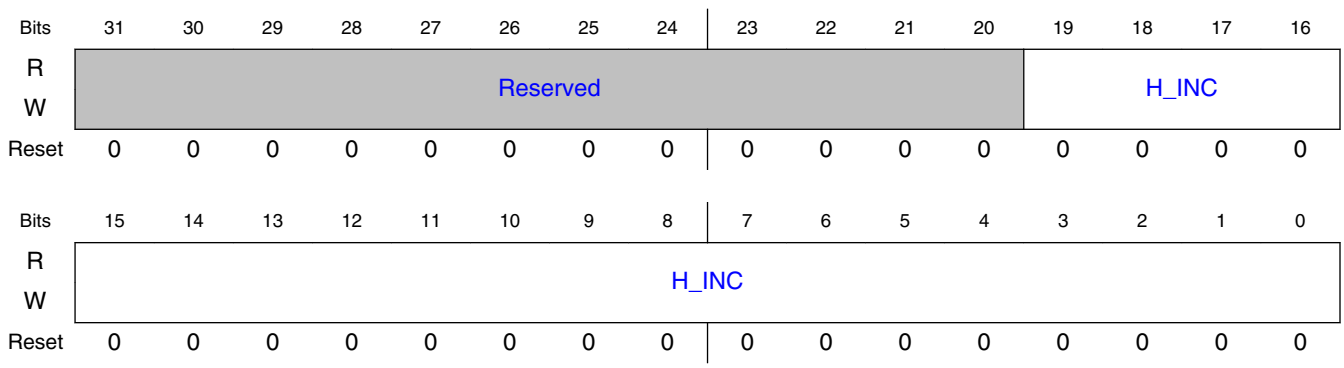
15.8.3.1.15.1 Offset

Register	Offset
SCALE_H_LUMA_INC	54h

15.8.3.1.15.2 Function

This register contains the increment which is used to define the scaling ratio. The upper 4 bits is the integer component of the step value. The lowest 13 bits is the fractional component.

15.8.3.1.15.3 Diagram



15.8.3.1.15.4 Fields

Field	Function
31-20 —	Reserved.
19-0 H_INC	Horizontal increment value used for scaling the image. This register is programmed with the following format: ###.#_####_####_####. The maximum value that can be programmed into this register is 0xE000, for a maximum scale down ratio of 1/7. Other programming examples could be: 0x1000, scale up by 2. 0x0800, scale up by 4. 0x6000, scale down by 3. 0x0AAB, scale up by 3, digital approximation of 1/3.

15.8.3.1.16 Scale Vertical Chroma Start Register (SCALE_V_CHROMA_START)

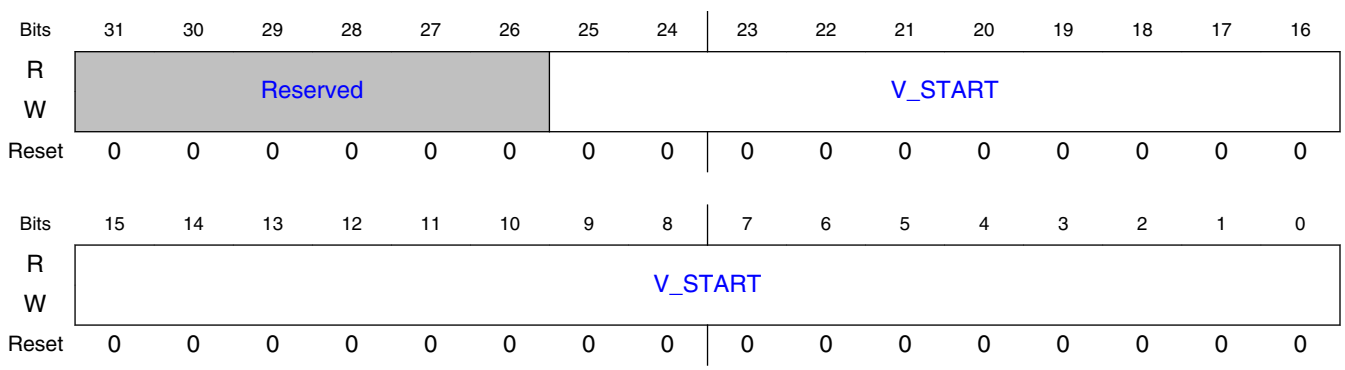
15.8.3.1.16.1 Offset

Register	Offset
SCALE_V_CHROMA_START	58h

15.8.3.1.16.2 Function

This register contains the initial vertical pixel start, or offset, into the image that is fetched by the prefetch engine. Essentially it is used to discard pixels (integer component) and provide and initial phase (fractional component) into the image in RTRAM. Fundamentally, the images loaded into the RTRAM from the DPR are aligned to 64 byte addresses. This register is used for cropping the top rows from the source image. To achieve better DRAM efficiency, DPR should be used to not fetch the unwanted lines in the top of the picture. The remaining rows i.e. the rows that cannot be dropped by DPR can be cropped in Scaler. There is no limit on the number of lines that can be cropped in the Scaler. If it crosses the bank boundary, Scaler gives out bank done without reading the bank data and this continues till the actual row is needed. If a few lines from the top are cropped, the source image pixels beyond the crop boundary aren't considered for scaling. Its treated as if they are outside the picture

15.8.3.1.16.3 Diagram



15.8.3.1.16.4 Fields

Field	Function
31-26 —	Reserved.
25-0 V_START	This register contains 13 integer and 13 fractional bits to define the vertical offset into the start of the prefetched image. It is a continuous 16 bit value with the following format ###.#_####_####_####.

15.8.3.1.17 Scale Vertical Chroma Increment Register (SCALE_V_CHROMA_INC)

15.8.3.1.17.1 Offset

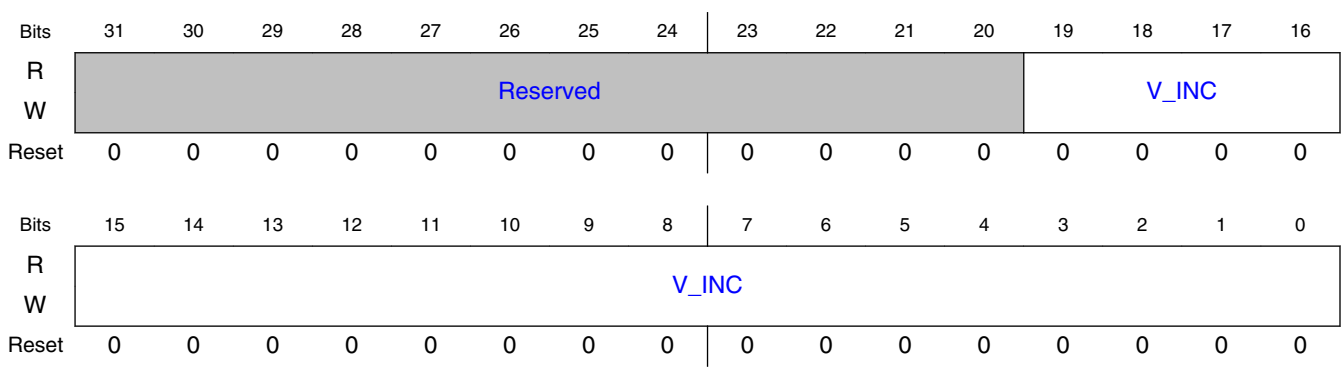
Register	Offset
SCALE_V_CHROMA_INC	5Ch

15.8.3.1.17.2 Function

This register contains the increment which is used to define the scaling ratio. The upper 4 bits is the integer component of the step value. The lowest 13 bits is the fractional component. The value programmed in this register is dependent on the value in the SCALE_V_LUMA_INC register AND the buffer format.

ARGB8888, YUV 1-plane: SCALE_V_CHROMA_INC = SCALE_V_LUMA_INC

15.8.3.1.17.3 Diagram



15.8.3.1.17.4 Fields

Field	Function
31-20 —	Reserved.
19-0 V_INC	Vertical increment value used for scaling the image. This register is programmed with the following format: ###.#_####_####_####. The maximum value that can be programmed into this register is 0xE000, for a maximum scale down ratio of 1/7. Other programming examples could be:

Field	Function
	0x1000, scale up by 2. 0x0800, scale up by 4. 0x6000, scale down by 3. 0x0AAB, scale up by 3, digital approximation of 1/3.

15.8.3.1.18 Scale Horizontal Chroma Start Register (SCALE_H_CHROMA_START)

15.8.3.1.18.1 Offset

Register	Offset
SCALE_H_CHROMA_START	60h

15.8.3.1.18.2 Function

This register contains the initial horizontal pixel start, or offset, into the image that is fetched by the prefetch engine. Essentially it is used to disgard pixels (integer component) and provide and initial phase (fractional component) into the image in RTRAM. Fundamentally, the images loaded into the RTRAM from the DPR are aligned to 64 byte addresses. This is essential to achieve DRAM efficiency.

The integer component of this register is used to skip input pixels in the horizontal direction.

If SW requires a high integer start value, change the base address of the source image in the DPR so that the undesired data is NOT prefetched.

This register may use a fractional component to establish an initial horizontal phase offset.

15.8.3.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						H_START									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H_START															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.18.4 Fields

Field	Function
31-26 —	Reserved.
25-0 H_START	This register contains 13 integer and 13 fractional bits to define the horizontal offset into the start of the prefetched image. It is a continuous 19 bit value with the following format ##_####.#_####_####_####. The start can be negative but greater than -1

15.8.3.1.19 Scale Horizontal Chroma Increment Register (SCALE_H_CHROMA_INC)

15.8.3.1.19.1 Offset

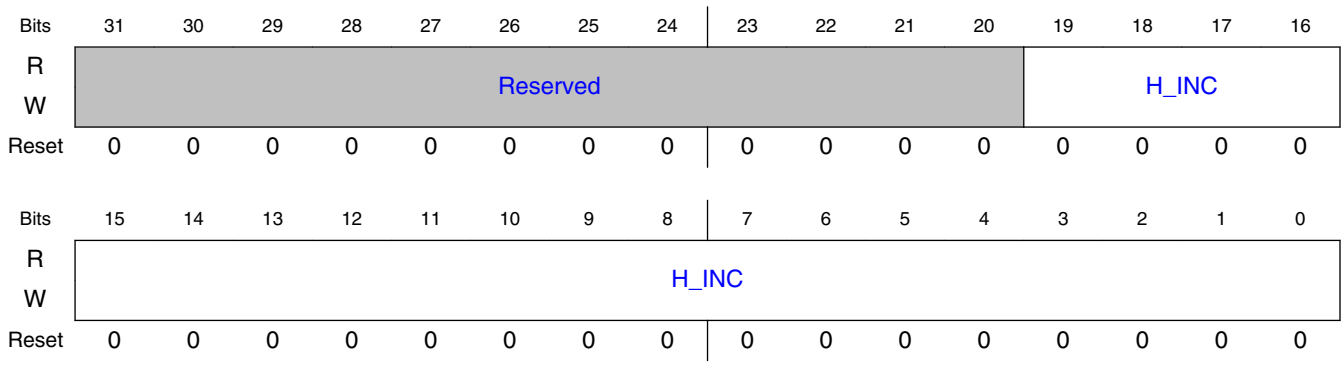
Register	Offset
SCALE_H_CHROMA_INC	64h

15.8.3.1.19.2 Function

This register contains the increment which is used to define the scaling ratio. The upper 4 bits is the integer component of the step value. The lowest 13 bits is the fractional component. The value programmed in this register is dependent on the value in the SCALE_H_LUMA_INC register AND the buffer format.

ARGB8888, YUV444: SCALE_H_CHROMA_INC = SCALE_H_LUMA_INC

15.8.3.1.19.3 Diagram



15.8.3.1.19.4 Fields

Field	Function
31-20 —	Reserved.
19-0 H_INC	Horizontal increment value used for scaling the image. This register is programmed with the following format: ###.#_####_####_####. The maximum value that can be programmed into this register is 0xE000, for a maximum scale down ratio of 1/7. Other programming examples could be: 0x1000, scale up by 2. 0x0800, scale up by 4. 0x6000, scale down by 3. 0x0AAB, scale up by 3, digital approximation of 1/3.

15.8.3.1.20 Scale Coefficient Memory Array (SCALE_COEF_ARRAY)

15.8.3.1.20.1 Offset

Register	Offset
SCALE_COEF_ARRAY	80h

15.8.3.1.20.2 Function

This register is not implemented in HW. This is place holder to document the address range used to program the scaling coefficients. The coefficients are defined as c(P,T) where P(0-15) are the possible 16 phases, and T(0-6) are the possible 7 coefficient taps per phase. The base address of all coefficient RAMs starts at 0x80. The entire range of addresses for coefficient access ends with address 0x3BF. Each coefficient is 12-bits.

Note, there are 12-bits per coefficent, and 7 taps, for a total of 12*7=84-bits. There are 3 memories to store the coefficients, and these memories have a word width of 28-bits. So, the coefficients are loaded 28-bits per PIO write. Since the coefficient bit width and memory word width are not evenly divisable, some coefficients will cross a memory word boundary. This is described below.

Vertical/Horizontal coefficients are accessed according to the following memory mapped layout.

ADDRESS	DESCRIPTION
0x80-0xBC	Vert/Luma, increment through 16-phases, tap 0, 1, and the high 4-bits of tap 2
0xC0-0xFF	Vert/Luma, increment through 16-phases, tap 2 low 8-bits, tap 3, and the high 8-bits of tap 4
0x100-0x13F	Vert/Luma, increment through 16-phases, tap 4 low 4-bits, tap 5, 6
0x140-0x17F	Horz/Luma, increment through 16-phases, taps 0, 1, and the high 4-bits of tap 2
0x180-0x1BF	Horz/Luma, increment through 16-phases, taps 2, 3, 4, as above
0x1C0-0x1FF	Horz/Luma, increment through 16-phases, taps 4, 5, 6, as above
0x200-0x23F	Vert/Chroma, increment through 16-phases, taps 0, 1, 2, as above
0x240-0x27F	Vert/Chroma, increment through 16-phases, taps 2, 3, 4, as above
0x280-0x2BF	Vert/Chroma, increment through 16-phases, taps 4, 5, 6, as above
0x300-0x33F	Horz/Chroma, increment through 16-phases, taps 0, 1, 2, as above
0x340-0x37F	Horz/Chroma, increment through 16-phases, taps 2, 3, 4, as above
0x380-0x3BF	Horz/Chroma, increment through 16-phases, taps 4, 5, 6, as above

```
As an example, the first few writes starting at address 0x80, 0x84, ..., would be:
# Adr = 0x80; Write phase 0, taps 0, 1, and the high 4-bits of tap 2.
# Adr = 0x84; Write phase 1, taps 0, 1, and the high 4-bits of tap 2.
...
# Adr = 0xBC; Write phase 15, taps 0, 1, and the high 4-bits of tap 2.
# Adr = 0xC0; Write phase 0, the low 8-bits of tap 2, tap 3, and the high 8-bits of tap 4.
WRITE(0x80, {4'b0, c(0,0),      c(0,1),  c(0,2)[11:8]})
WRITE(0x84, {4'b0, c(1,0),      c(1,1),  c(1,2)[11:8]})
...
WRITE(0xBC, {4'b0, c(15,0),     c(15,1), c(15,2)[11:8]})
WRITE(0xC0, {4'b0, c(0,2)[7:0], c(0,3),  c(0,4)[11:4]})
...
```

Using two dimensional arrays for vertical/horizontal and luma/chroma "c[v|h][l|c][P,T]" to store 12-bit coefficients, defined as c[15:0][6:0][11:0], the following code could be used to load the coefficient memories with the desired coefficient values.

```
for (i=0, i.lt.16, i=i+1) (
   WRT(0x80+(4*i) ,{cvl(i,0)[11:0], cvl(i,1)[11:0], cvl(i,2)[11:8]});
   WRT(0xC0+(4*i) ,{cvl(i,2)[7:0],  cvl(i,3)[11:0], cvl(i,4)[11:4]});
   WRT(0x100+(4*i),{cvl(i,4)[3:0],  cvl(i,5)[11:0], cvl(i,6)[11:0]});
   WRT(0x140+(4*i),{chl(i,0)[11:0], chl(i,1)[11:0], chl(i,2)[11:8]});
   WRT(0x180+(4*i),{chl(i,2)[7:0],  chl(i,3)[11:0], chl(i,4)[11:4]});
   WRT(0x1C0+(4*i),{chl(i,4)[3:0],  chl(i,5)[11:0], chl(i,6)[11:0]});
   WRT(0x200+(4*i),{cvc(i,0)[11:0], cvc(i,1)[11:0], cvc(i,2)[11:8]});
   WRT(0x240+(4*i),{cvc(i,2)[7:0],  cvc(i,3)[11:0], cvc(i,4)[11:4]});
   WRT(0x280+(4*i),{cvc(i,4)[3:0],  cvc(i,5)[11:0], cvc(i,6)[11:0]});
```

```
WRT(0x300+(4*i),{chc(i,0)[11:0], chc(i,1)[11:0], chc(i,2)[11:8]});
WRT(0x340+(4*i),{chc(i,2)[7:0],  chc(i,3)[11:0], chc(i,4)[11:4]});
WRT(0x380+(4*i),{chc(i,4)[3:0],  chc(i,5)[11:0], chc(i,6)[11:0]});
)
```

15.8.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved				COEF											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.8.3.1.20.4 Fields

Field	Function
31-12 —	Reserved.
11-0 COEF	The coefficients have 1-bit sign, 1-bit integer, and 10-bits fraction. A total of bits [11:0], or {sign,integer,frac[9:0]}

15.9 Look Up Table Load (LUT_LD)

15.9.1 Overview

The Look Up Table Load module (LUT_LD) provides the interface between the DMA interface of the Dolby IP and the AXI bus transactions to system RAM. This interface orchestrates the fetch of the tone curve LUT and the G2L LUT data from system RAM. Both the Dolby video and graphics pixel data pipelines require this LUT data. The LUT_LD engine will serve as the interface for both the video and graphics IP. Essentially, the LUT_LD will fetch the LUT data for the video interface first, followed by the LUT data for the graphics pipeline second. This module will not provide an asynchronous domain crossing between clock domains.

NOTE: More information on the DMA interface and LUT data processing can be obtained from the Dolby IP specs.

15.9.1.1 Block Diagram

The following diagram indicates the microarchitecture of the LUT_LD module.

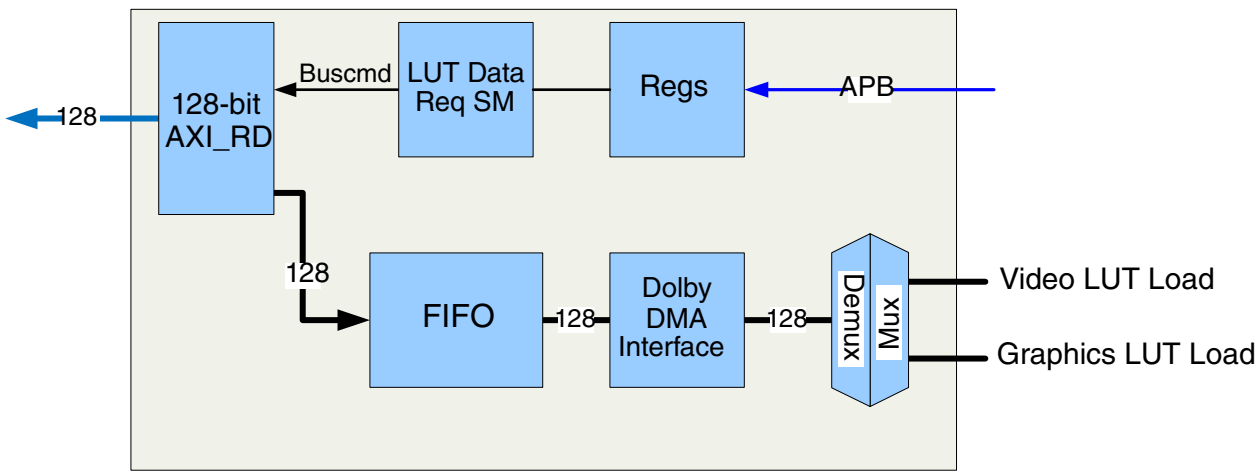


Figure 15-21. LUT_LD Block Diagram

15.9.1.2 Features

The LUT_LD integrates the following functional components:

- AXI_RD buscmd/AXI interface module at 128-bit bus width.
- LUT Data Request State Machine
- CSR registers to define the base address of the 10KB of LUT data.
- 512B FIFO.
- Dolby DMA interface SM

The amount of data to be fetched is defined in the Dolby IP specs. The video tone curve = 4KB and the G2L = 1KB.

Since both the video and graphics pixel pipelines require the same amount of LUT data (5KB), the LUT_LD engine serves the video pipeline first (5KB), followed by the graphics pipeline (5KB). Each Dolby pixel processing component needs 5KB of data to complete the LUT loading process. In all, 10KB is transferred over the two Dolby IP DMA interfaces (5KB each).

The FIFO is implemented to manage system memory latency and throughput.

The LUT_LD engine begins fetching LUT data from system memory when the CNTL_STATUS.ENABLE bit is set. The load sequence runs once and does not repeat every display trace. If the previous data needs to be fetched in subsequent display traces, the LUT_LD engine needs to be enabled for the loading sequence to commence.

15.9.2 Memory Map and Registers

15.9.2.1 register descriptions

15.9.2.1.1 LUT_LD Memory map

lut_ld base address: 2_4000h

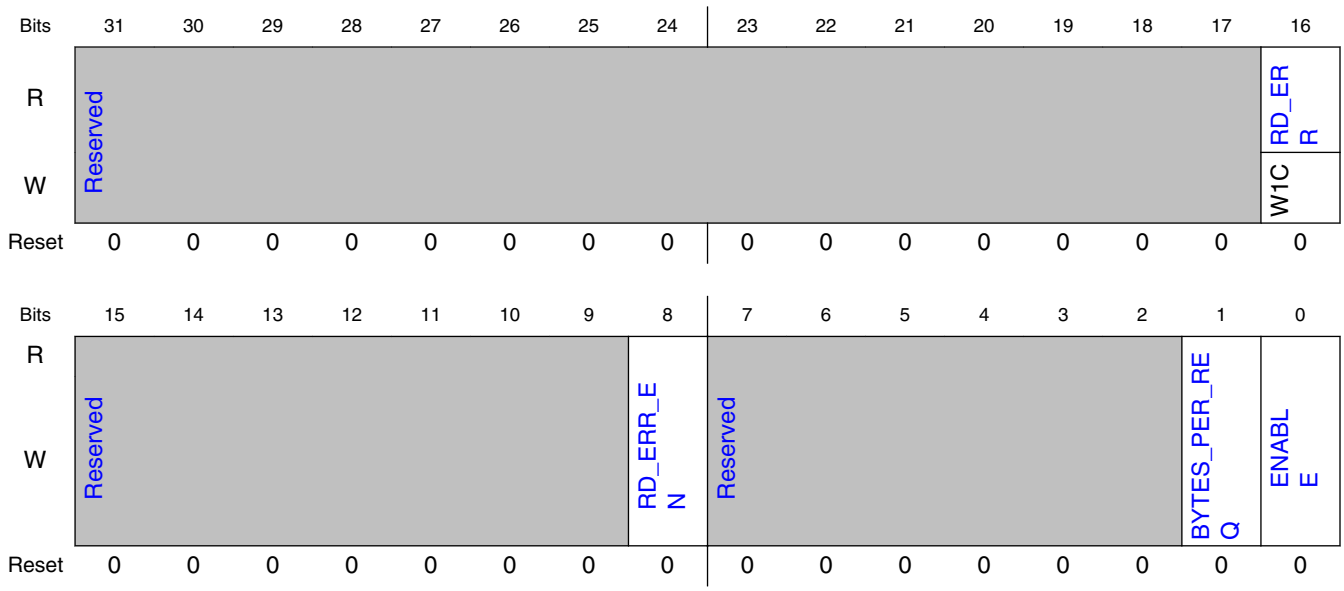
Offset	Register	Width (In bits)	Access	Reset value
0h	Control/Status register for LUT Loader. (CTRL_STATUS)	32	RW	0000_0000h
4h	Control/Status register for LUT Loader. (CTRL_STATUS_SET)	32	RW	0000_0000h
8h	Control/Status register for LUT Loader. (CTRL_STATUS_CLR)	32	RW	0000_0000h
Ch	Control/Status register for LUT Loader. (CTRL_STATUS_TOG)	32	RW	0000_0000h
10h	Address for data fetch. (BASE_ADDR)	32	RW	0000_0000h

15.9.2.1.2 Control/Status register for LUT Loader. (CTRL_STATUS)

15.9.2.1.2.1 Offset

Register	Offset
CTRL_STATUS	0h

15.9.2.1.2.2 Diagram



15.9.2.1.2.3 Fields

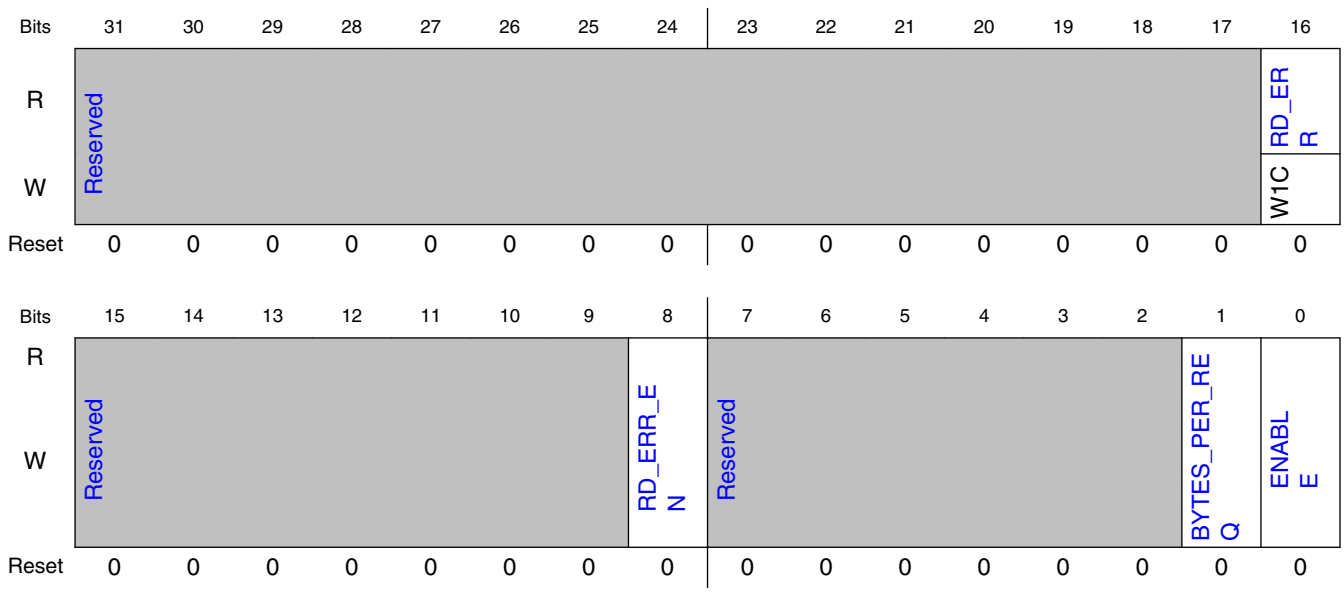
Field	Function
31-17 —	Reserved
16 RD_ERR	AXI Read Error. Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
15-9 —	Reserved
8 RD_ERR_EN	AXI Read Error IRQ enable. Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
7-2 —	Reserved
1 BYTES_PER_REQ	bytes per request control This field defines the number of bytes fetched for each AXI system memory request. • 0 - 256 bytes of data for each AXI transaction. • 1 - 128 bytes of data for each AXI transaction
0 ENABLE	Enable LUT_LD Enable bit for the lut loader. When set, the LUT_LD begins to fetch data from system memory. The data (Tone Curve, G2L) is transferred to the Dolby pixel engine via the DMA interface. A total of 10KB is fetched from system memory. 5 KB for video and 5 KB for graphics. Hardware automatically clears this bit when all 10KB of data is transferred to the Dolby pixel engine..

15.9.2.1.3 Control/Status register for LUT Loader. (CTRL_STATUS_SET)

15.9.2.1.3.1 Offset

Register	Offset
CTRL_STATUS_SET	4h

15.9.2.1.3.2 Diagram



15.9.2.1.3.3 Fields

Field	Function
31-17 —	Reserved
16 RD_ERR	AXI Read Error. Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
15-9 —	Reserved
8 RD_ERR_EN	AXI Read Error IRQ enable. Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
7-2 —	Reserved
1	bytes per request control

Table continues on the next page...

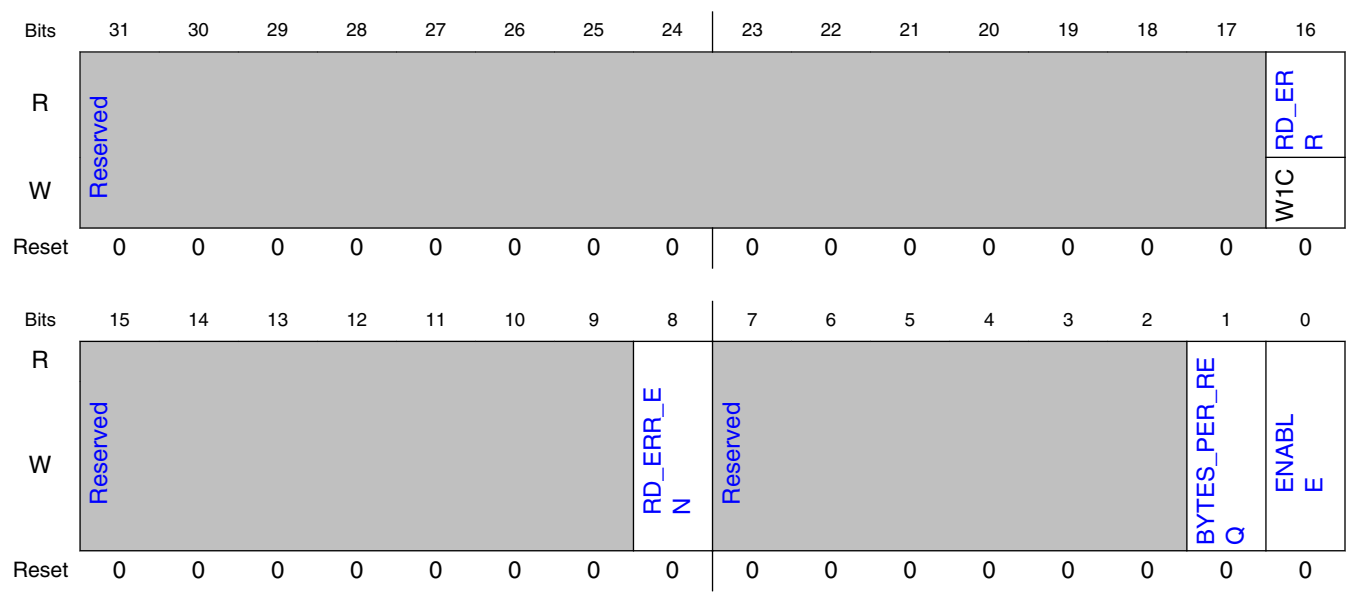
Field	Function
BYTES_PER_REQ	This field defines the number of bytes fetched for each AXI system memory request. • 0 - 256 bytes of data for each AXI transaction. • 1 - 128 bytes of data for each AXI transaction
0 ENABLE	Enable LUT_LD Enable bit for the lut loader. When set, the LUT_LD begins to fetch data from system memory. The data (Tone Curve, G2L) is transferred to the Dolby pixel engine via the DMA interface. A total of 10KB is fetched from system memory. 5 KB for video and 5 KB for graphics. Hardware automatically clears this bit when all 10KB of data is transferred to the Dolby pixel engine..

15.9.2.1.4 Control/Status register for LUT Loader. (CTRL_STATUS_CLR)

15.9.2.1.4.1 Offset

Register	Offset
CTRL_STATUS_CLR	8h

15.9.2.1.4.2 Diagram



15.9.2.1.4.3 Fields

Field	Function
31-17	Reserved

Table continues on the next page...

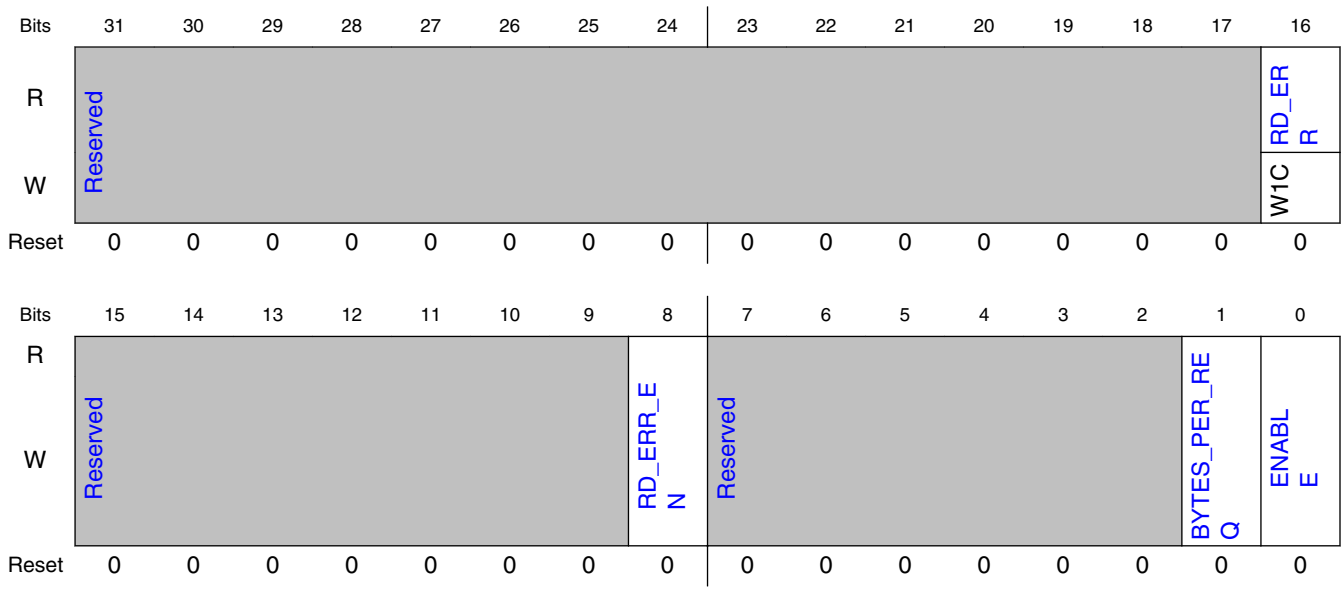
Field	Function
—	
16 RD_ERR	AXI Read Error. Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
15-9 —	Reserved
8 RD_ERR_EN	AXI Read Error IRQ enable. Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
7-2 —	Reserved
1 BYTES_PER_REQ	bytes per request control This field defines the number of bytes fetched for each AXI system memory request. • 0 - 256 bytes of data for each AXI transaction. • 1 - 128 bytes of data for each AXI transaction
0 ENABLE	Enable LUT_LD Enable bit for the lut loader. When set, the LUT_LD begins to fetch data from system memory. The data (Tone Curve, G2L) is transferred to the Dolby pixel engine via the DMA interface. A total of 10KB is fetched from system memory. 5 KB for video and 5 KB for graphics. Hardware automatically clears this bit when all 10KB of data is transferred to the Dolby pixel engine..

15.9.2.1.5 Control/Status register for LUT Loader. (CTRL_STATUS_TOG)

15.9.2.1.5.1 Offset

Register	Offset
CTRL_STATUS_TOG	Ch

15.9.2.1.5.2 Diagram



15.9.2.1.5.3 Fields

Field	Function
31-17 —	Reserved
16 RD_ERR	AXI Read Error. Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
15-9 —	Reserved
8 RD_ERR_EN	AXI Read Error IRQ enable. Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
7-2 —	Reserved
1 BYTES_PER_REQ	bytes per request control This field defines the number of bytes fetched for each AXI system memory request. • 0 - 256 bytes of data for each AXI transaction. • 1 - 128 bytes of data for each AXI transaction
0 ENABLE	Enable LUT_LD Enable bit for the lut loader. When set, the LUT_LD begins to fetch data from system memory. The data (Tone Curve, G2L) is transferred to the Dolby pixel engine via the DMA interface. A total of 10KB is fetched from system memory. 5 KB for video and 5 KB for graphics. Hardware automatically clears this bit when all 10KB of data is transferred to the Dolby pixel engine..

15.9.2.1.6 Address for data fetch. (BASE_ADDR)

15.9.2.1.6.1 Offset

Register	Offset
BASE_ADDR	10h

15.9.2.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.9.2.1.6.3 Fields

Field	Function
31-0 BASE_ADDR	Base Address Base address of continuous 5KB video and 5KB graphics LUT data. This address should be aligned to a 64B boundary in system memory.

15.10 HDR10 Image Processing (MED_HDR10)

15.10.1 Overview

In HDR10 mode, the three inputs get merged or blended together, post processed and sent to the HDMI protocol controller. The HDMI output consists of a display window containing up to three input windows. Any of the three windows can by driven by the Readback channel. All three windows can be full screen or smaller and offset inside the display window with the rest of the display window filled with the background color. The compositing of the three windows is controlled by the per pixel alpha channel from channel 1. The alpha logic creates an alpha based on the window priority and the graphics

per pixel alpha. The diagram below shows a possible window configuration. Note that window priority is fixed with channel 1 on top of channel 2 on top of channel 3. Only channel 1 can have a per pixel alpha.

Figure 15-22. HDR10 Processed Display

15.10.1.1 Features

In HDR10 mode, all three pixel processing pipelines are identical.

Each HDR10 pixel processing input pipeline supports:
- 10-bit per component input
- Color space conversion
- LUT gamma correction per component

HDR10 Output Pipeline The HDR10 pixel processing output pipeline supports:
- Linear to non-linear conversion per component
- Color space conversion
- 10-bit per component RGB/YUV444

15.10.2 Functional description

15.10.2.1 HDR10 Input Path

Below is a table showing the input to output flow depending on the source of HDR10 video, SDR video, or graphics and the output TV being HDR10 or SDR:

INPUT	Pre LUT	YCbCr CSC	EOTF LUT	2020 CSC	Output	TV
Graphics ARGB 8888 8b Rec. 709	Bypass	Bypass	Gamma EOTF (Non-linear RGB to Linear RGB) 3x1024 14b	RGB Rec.709 to RGB Rec.2020 Scale NIT 3x3 CSC 14b to 28b (1.14 coeff)	ARGB 8888 28b	HDR10
				Bypass		SDR
SDR Video YUV 444 8/10b Rec. 709	Remove offset 3x1024 10b	YCbCr to RGB 3x3 CSC 10b to 10b (1.14 coeff)		RGB Rec.709 to RGB Rec.2020 Scale NIT 3x3 CSC 14b to 28b (1.14 coeff)		HDR10
				Bypass		SDR

Table continues on the next page...

INPUT	Pre LUT	YCbCr CSC	EOTF LUT	2020 CSC	Output	TV
HDR10 Video YUV 444 10b Rec.2020			ST2084 EOTF (Non-linear RGB to Linear RGB) 3x1024 28b			HDR10

15.10.2.2 HDR10 Output Path

Below is a chart showing the input to output flow depending on the HDMI output format. The HDMI types include:

- HDMI 8-bit or 10-bit RGB
- HDMI 8-bit or 10-bit YCbCr 4:4:4
- HDMI 8-bit or 10-bit YCbCr 4:2:2
- HDMI 8-bit or 10-bit YCbCr 4:2:0

Output Path						
INPUT	**Piece-wise linear interpolation**	**CSC**	**Post LUT**	**Chroma Sample Decimation**	**Output**	**HDMI Format**
Linear RGB 444 28b	Reverse EOTF (Convert Linear RGB to Non-linear RBG OR Gamma) Piece-wise linear conversion 28b to 10b	Bypass	Add offset for limited range 3x1024 10b	Bypass	RGB 8/10b	HDMI RGB
		RGB to YCbCr 3x3 CSC 10b to 10b			HDMI YCbCr 444	HDMI YCbCr 444
				Horizontal		HDMI YCbCr 422
				Horizontal and Vertical		HDMI YCbCr 420

15.10.3 Memory Map and Registers

15.10.3.1 register descriptions

15.10.3.1.1 MED_HDR10 Memory map

med_hdr10 base address: C000h

Offset	Register	Width (In bits)	Access	Reset value
0h	A0 component Look-Up-Table. (LUT) (PIPE1_A0_LUT)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
1000h	A1 component Look-Up-Table. (LUT) (PIPE1_A1_LUT)	32	RW	0000_0000h
2000h	A2 component Look-Up-Table. (LUT) (PIPE1_A2_LUT)	32	RW	0000_0000h
3000h	Pipe1 Colorspace Converter A control. (HDR_PIPE1_CSCA_CONTROL_REG)	32	RW	0000_0000h
3004h	Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE1_CSCA_H00)	32	RW	0000_0000h
3008h	Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE1_CSCA_H10)	32	RW	0000_0000h
300Ch	Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE1_CSCA_H20)	32	RW	0000_0000h
3010h	Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE1_CSCA_H01)	32	RW	0000_0000h
3014h	Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE1_CSCA_H11)	32	RW	0000_0000h
3018h	Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE1_CSCA_H21)	32	RW	0000_0000h
301Ch	Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE1_CSCA_H02)	32	RW	0000_0000h
3020h	Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE1_CSCA_H12)	32	RW	0000_0000h
3024h	Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE1_CSCA_H22)	32	RW	0000_0000h
3028h	Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE1_CSCA_IO_0)	32	RW	0000_0000h
302Ch	Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE1_CSCA_IO_1)	32	RW	0000_0000h
3030h	Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE1_CSCA_IO_2)	32	RW	0000_0000h
3034h	Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE1_CSCA_IO_MIN_0)	32	RW	0000_0000h
3038h	Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE1_CSCA_IO_MIN_1)	32	RW	0000_0000h
303Ch	Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE1_CSCA_IO_MIN_2)	32	RW	0000_0000h
3040h	Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE1_CSCA_IO_MAX_0)	32	RW	0000_0000h
3044h	Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE1_CSCA_IO_MAX_1)	32	RW	0000_0000h
3048h	Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE1_CSCA_IO_MAX_2)	32	RW	0000_0000h
304Ch	Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE1_CSCA_NORM)	32	RW	0000_0000h
3050h	Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE1_CSCA_OO_0)	32	RW	0000_0000h
3054h	Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE1_CSCA_OO_1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
3058h	Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE1_CSCA_OO_2)	32	RW	0000_0000h
305Ch	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE1_CSCA_OMIN_0)	32	RW	0000_0000h
3060h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE1_CSCA_OMIN_1)	32	RW	0000_0000h
3064h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE1_CSCA_OMIN_2)	32	RW	0000_0000h
3068h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE1_CSCA_OMAX_0)	32	RW	0000_0000h
306Ch	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE1_CSCA_OMAX_1)	32	RW	0000_0000h
3070h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE1_CSCA_OMAX_2)	32	RW	0000_0000h
3074h	PIPE1: NOT USED (HDR_PIPE1_ENTRY_29)	32	RU	0000_0000h
3080h	Pipe1 LUT control register (HDR_PIPE1_LUT_CONTROL_REG)	32	RW	0000_0000h
3800h	Pipe1 Colorspace Converter B control. (HDR_PIPE1_CSCB_CONTROL_REG)	32	RW	0000_0000h
3804h	Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE1_CSCB_H00)	32	RW	0000_0000h
3808h	Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE1_CSCB_H10)	32	RW	0000_0000h
380Ch	Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE1_CSCB_H20)	32	RW	0000_0000h
3810h	Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE1_CSCB_H01)	32	RW	0000_0000h
3814h	Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE1_CSCB_H11)	32	RW	0000_0000h
3818h	Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE1_CSCB_H21)	32	RW	0000_0000h
381Ch	Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE1_CSCB_H02)	32	RW	0000_0000h
3820h	Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE1_CSCB_H12)	32	RW	0000_0000h
3824h	Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE1_CSCB_H22)	32	RW	0000_0000h
3828h	Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE1_CSCB_IO_0)	32	RW	0000_0000h
382Ch	Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE1_CSCB_IO_1)	32	RW	0000_0000h
3830h	Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE1_CSCB_IO_2)	32	RW	0000_0000h
3834h	Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE1_CSCB_IO_MIN_0)	32	RW	0000_0000h
3838h	Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE1_CSCB_IO_MIN_1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
383Ch	Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE1_CSCB_IO_MIN_2)	32	RW	0000_0000h
3840h	Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE1_CSCB_IO_MAX_0)	32	RW	0000_0000h
3844h	Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE1_CSCB_IO_MAX_1)	32	RW	0000_0000h
3848h	Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE1_CSCB_IO_MAX_2)	32	RW	0000_0000h
384Ch	Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE1_CSCB_NORM)	32	RW	0000_0000h
3850h	Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE1_CSCB_OO_0)	32	RW	0000_0000h
3854h	Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE1_CSCB_OO_1)	32	RW	0000_0000h
3858h	Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE1_CSCB_OO_2)	32	RW	0000_0000h
385Ch	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE1_CSCB_OMIN_0)	32	RW	0000_0000h
3860h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE1_CSCB_OMIN_1)	32	RW	0000_0000h
3864h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE1_CSCB_OMIN_2)	32	RW	0000_0000h
3868h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE1_CSCB_OMAX_0)	32	RW	0000_0000h
386Ch	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE1_CSCB_OMAX_1)	32	RW	0000_0000h
3870h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE1_CSCB_OMAX_2)	32	RW	0000_0000h
3874h	Pipe1 floating point to fixed point control (HDR_PIPE1_FL2FX)	32	RW	0000_0000h
3878h	PIPE1: NOT USED (HDR_PIPE1_ENTRY_30)	32	RU	0000_0000h
4000h	A0 component Look-Up-Table. (LUT) (PIPE2_A0_LUT)	32	RW	0000_0000h
5000h	A1 component Look-Up-Table. (LUT) (PIPE2_A1_LUT)	32	RW	0000_0000h
6000h	A2 component Look-Up-Table. (LUT) (PIPE2_A2_LUT)	32	RW	0000_0000h
7000h	Pipe1 Colorspace Converter A control. (HDR_PIPE2_CSCA_CONTROL_REG)	32	RW	0000_0000h
7004h	Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE2_CSCA_H00)	32	RW	0000_0000h
7008h	Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE2_CSCA_H10)	32	RW	0000_0000h
700Ch	Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE2_CSCA_H20)	32	RW	0000_0000h
7010h	Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE2_CSCA_H01)	32	RW	0000_0000h
7014h	Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE2_CSCA_H11)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7018h	Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE2_CSCA_H21)	32	RW	0000_0000h
701Ch	Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE2_CSCA_H02)	32	RW	0000_0000h
7020h	Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE2_CSCA_H12)	32	RW	0000_0000h
7024h	Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE2_CSCA_H22)	32	RW	0000_0000h
7028h	Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE2_CSCA_IO_0)	32	RW	0000_0000h
702Ch	Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE2_CSCA_IO_1)	32	RW	0000_0000h
7030h	Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE2_CSCA_IO_2)	32	RW	0000_0000h
7034h	Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE2_CSCA_IO_MIN_0)	32	RW	0000_0000h
7038h	Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE2_CSCA_IO_MIN_1)	32	RW	0000_0000h
703Ch	Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE2_CSCA_IO_MIN_2)	32	RW	0000_0000h
7040h	Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE2_CSCA_IO_MAX_0)	32	RW	0000_0000h
7044h	Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE2_CSCA_IO_MAX_1)	32	RW	0000_0000h
7048h	Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE2_CSCA_IO_MAX_2)	32	RW	0000_0000h
704Ch	Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE2_CSCA_NORM)	32	RW	0000_0000h
7050h	Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE2_CSCA_OO_0)	32	RW	0000_0000h
7054h	Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE2_CSCA_OO_1)	32	RW	0000_0000h
7058h	Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE2_CSCA_OO_2)	32	RW	0000_0000h
705Ch	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE2_CSCA_OMIN_0)	32	RW	0000_0000h
7060h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE2_CSCA_OMIN_1)	32	RW	0000_0000h
7064h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE2_CSCA_OMIN_2)	32	RW	0000_0000h
7068h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE2_CSCA_OMAX_0)	32	RW	0000_0000h
706Ch	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE2_CSCA_OMAX_1)	32	RW	0000_0000h
7070h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE2_CSCA_OMAX_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7074h	PIPE2: NOT USED (HDR_PIPE2_ENTRY_29)	32	RU	0000_0000h
7080h	Pipe1 LUT control register (HDR_PIPE2_LUT_CONTROL_REG)	32	RW	0000_0000h
7800h	Pipe1 Colorspace Converter B control. (HDR_PIPE2_CSCB_CONTROL_REG)	32	RW	0000_0000h
7804h	Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE2_CSCB_H00)	32	RW	0000_0000h
7808h	Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE2_CSCB_H10)	32	RW	0000_0000h
780Ch	Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE2_CSCB_H20)	32	RW	0000_0000h
7810h	Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE2_CSCB_H01)	32	RW	0000_0000h
7814h	Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE2_CSCB_H11)	32	RW	0000_0000h
7818h	Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE2_CSCB_H21)	32	RW	0000_0000h
781Ch	Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE2_CSCB_H02)	32	RW	0000_0000h
7820h	Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE2_CSCB_H12)	32	RW	0000_0000h
7824h	Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE2_CSCB_H22)	32	RW	0000_0000h
7828h	Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE2_CSCB_IO_0)	32	RW	0000_0000h
782Ch	Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE2_CSCB_IO_1)	32	RW	0000_0000h
7830h	Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE2_CSCB_IO_2)	32	RW	0000_0000h
7834h	Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE2_CSCB_IO_MIN_0)	32	RW	0000_0000h
7838h	Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE2_CSCB_IO_MIN_1)	32	RW	0000_0000h
783Ch	Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE2_CSCB_IO_MIN_2)	32	RW	0000_0000h
7840h	Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE2_CSCB_IO_MAX_0)	32	RW	0000_0000h
7844h	Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE2_CSCB_IO_MAX_1)	32	RW	0000_0000h
7848h	Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE2_CSCB_IO_MAX_2)	32	RW	0000_0000h
784Ch	Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE2_CSCB_NORM)	32	RW	0000_0000h
7850h	Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE2_CSCB_OO_0)	32	RW	0000_0000h
7854h	Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE2_CSCB_OO_1)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
7858h	Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE2_CSCB_OO_2)	32	RW	0000_0000h
785Ch	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE2_CSCB_OMIN_0)	32	RW	0000_0000h
7860h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE2_CSCB_OMIN_1)	32	RW	0000_0000h
7864h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE2_CSCB_OMIN_2)	32	RW	0000_0000h
7868h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE2_CSCB_OMAX_0)	32	RW	0000_0000h
786Ch	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE2_CSCB_OMAX_1)	32	RW	0000_0000h
7870h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE2_CSCB_OMAX_2)	32	RW	0000_0000h
7874h	Pipe1 floating point to fixed point control (HDR_PIPE2_FL2FX)	32	RW	0000_0000h
7878h	PIPE2: NOT USED (HDR_PIPE2_ENTRY_30)	32	RU	0000_0000h
8000h	A0 component Look-Up-Table. (LUT) (PIPE3_A0_LUT)	32	RW	0000_0000h
9000h	A1 component Look-Up-Table. (LUT) (PIPE3_A1_LUT)	32	RW	0000_0000h
A000h	A2 component Look-Up-Table. (LUT) (PIPE3_A2_LUT)	32	RW	0000_0000h
B000h	Pipe1 Colorspace Converter A control. (HDR_PIPE3_CSCA_CONTROL_REG)	32	RW	0000_0000h
B004h	Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE3_CSCA_H00)	32	RW	0000_0000h
B008h	Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE3_CSCA_H10)	32	RW	0000_0000h
B00Ch	Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE3_CSCA_H20)	32	RW	0000_0000h
B010h	Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE3_CSCA_H01)	32	RW	0000_0000h
B014h	Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE3_CSCA_H11)	32	RW	0000_0000h
B018h	Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE3_CSCA_H21)	32	RW	0000_0000h
B01Ch	Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE3_CSCA_H02)	32	RW	0000_0000h
B020h	Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE3_CSCA_H12)	32	RW	0000_0000h
B024h	Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE3_CSCA_H22)	32	RW	0000_0000h
B028h	Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE3_CSCA_IO_0)	32	RW	0000_0000h
B02Ch	Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE3_CSCA_IO_1)	32	RW	0000_0000h
B030h	Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE3_CSCA_IO_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B034h	Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE3_CSCA_IO_MIN_0)	32	RW	0000_0000h
B038h	Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE3_CSCA_IO_MIN_1)	32	RW	0000_0000h
B03Ch	Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE3_CSCA_IO_MIN_2)	32	RW	0000_0000h
B040h	Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE3_CSCA_IO_MAX_0)	32	RW	0000_0000h
B044h	Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE3_CSCA_IO_MAX_1)	32	RW	0000_0000h
B048h	Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE3_CSCA_IO_MAX_2)	32	RW	0000_0000h
B04Ch	Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE3_CSCA_NORM)	32	RW	0000_0000h
B050h	Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE3_CSCA_OO_0)	32	RW	0000_0000h
B054h	Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE3_CSCA_OO_1)	32	RW	0000_0000h
B058h	Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE3_CSCA_OO_2)	32	RW	0000_0000h
B05Ch	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE3_CSCA_OMIN_0)	32	RW	0000_0000h
B060h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE3_CSCA_OMIN_1)	32	RW	0000_0000h
B064h	Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE3_CSCA_OMIN_2)	32	RW	0000_0000h
B068h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE3_CSCA_OMAX_0)	32	RW	0000_0000h
B06Ch	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE3_CSCA_OMAX_1)	32	RW	0000_0000h
B070h	Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE3_CSCA_OMAX_2)	32	RW	0000_0000h
B074h	PIPE3: NOT USED (HDR_PIPE3_ENTRY_29)	32	RU	0000_0000h
B080h	Pipe1 LUT control register (HDR_PIPE3_LUT_CONTROL_REG)	32	RW	0000_0000h
B800h	Pipe1 Colorspace Converter B control. (HDR_PIPE3_CSCB_CONTROL_REG)	32	RW	0000_0000h
B804h	Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE3_CSCB_H00)	32	RW	0000_0000h
B808h	Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE3_CSCB_H10)	32	RW	0000_0000h
B80Ch	Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE3_CSCB_H20)	32	RW	0000_0000h
B810h	Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE3_CSCB_H01)	32	RW	0000_0000h
B814h	Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE3_CSCB_H11)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B818h	Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE3_CSCB_H21)	32	RW	0000_0000h
B81Ch	Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE3_CSCB_H02)	32	RW	0000_0000h
B820h	Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE3_CSCB_H12)	32	RW	0000_0000h
B824h	Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE3_CSCB_H22)	32	RW	0000_0000h
B828h	Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE3_CSCB_IO_0)	32	RW	0000_0000h
B82Ch	Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE3_CSCB_IO_1)	32	RW	0000_0000h
B830h	Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE3_CSCB_IO_2)	32	RW	0000_0000h
B834h	Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE3_CSCB_IO_MIN_0)	32	RW	0000_0000h
B838h	Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE3_CSCB_IO_MIN_1)	32	RW	0000_0000h
B83Ch	Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE3_CSCB_IO_MIN_2)	32	RW	0000_0000h
B840h	Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE3_CSCB_IO_MAX_0)	32	RW	0000_0000h
B844h	Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE3_CSCB_IO_MAX_1)	32	RW	0000_0000h
B848h	Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE3_CSCB_IO_MAX_2)	32	RW	0000_0000h
B84Ch	Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE3_CSCB_NORM)	32	RW	0000_0000h
B850h	Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE3_CSCB_OO_0)	32	RW	0000_0000h
B854h	Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE3_CSCB_OO_1)	32	RW	0000_0000h
B858h	Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE3_CSCB_OO_2)	32	RW	0000_0000h
B85Ch	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE3_CSCB_OMIN_0)	32	RW	0000_0000h
B860h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE3_CSCB_OMIN_1)	32	RW	0000_0000h
B864h	Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE3_CSCB_OMIN_2)	32	RW	0000_0000h
B868h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE3_CSCB_OMAX_0)	32	RW	0000_0000h
B86Ch	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE3_CSCB_OMAX_1)	32	RW	0000_0000h
B870h	Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE3_CSCB_OMAX_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
B874h	Pipe1 floating point to fixed point control (HDR_PIPE3_FL2FX)	32	RW	0000_0000h
B878h	PIPE3: NOT USED (HDR_PIPE3_ENTRY_30)	32	RU	0000_0000h
C000h	A0 component Linear-to-Non-linear conversion table (OPIPE_A0_TABLE)	32	RW	0000_0000h
D000h	A1 component Linear-to-Non-linear conversion table (OPIPE_A1_TABLE)	32	RW	0000_0000h
E000h	A2 component Linear-to-Non-linear conversion table (OPIPE_A2_TABLE)	32	RW	0000_0000h
F000h	HDR output stage Colorspace Converter (CSCO) control. (HDR_OPIPE_CSC_CONTROL_REG)	32	RW	0000_0000h
F004h	Pipe1 Colorspace Converter (CSC) h(0,0) matrix coefficient (HDR_OPIPE_CSC_H00)	32	RW	0000_0000h
F008h	Pipe1 Colorspace Converter (CSC) h(1,0) matrix coefficient (HDR_OPIPE_CSC_H10)	32	RW	0000_0000h
F00Ch	HDR OUTPUT Colorspace Converter (CSCO) h(2,0) matrix coefficient (HDR_OPIPE_CSC_H20)	32	RW	0000_0000h
F010h	HDR OUTPUT pipe Colorspace Converter (CSCO) h(0,1) matrix coefficient (HDR_OPIPE_CSC_H01)	32	RW	0000_0000h
F014h	HDR OUTPUT pipe Colorspace Converter (CSCO) h(1,1) matrix coefficient (HDR_OPIPE_CSC_H11)	32	RW	0000_0000h
F018h	HDR_output pipe Colorspace Converter (CSCO) h(2,1) matrix coefficient (HDR_OPIPE_CSC_H21)	32	RW	0000_0000h
F01Ch	HDR OUTPUT pipe Colorspace Converter (CSCO) h(0,2) matrix coefficient (HDR_OPIPE_CSC_H02)	32	RW	0000_0000h
F020h	HDR OUPUT pipe Colorspace Converter (CSCO) h(1,2) matrix coefficient (HDR_OPIPE_CSC_H12)	32	RW	0000_0000h
F024h	HDR OUPUT pipe Colorspace Converter (CSCO) h(2,2) matrix coefficient (HDR_)	32	RW	0000_0000h
F028h	HDR OUTPUT pipe Colorspace Converter (CSCO) component 0 pre-offset (HDR_OPIPE_CSC_IO_0)	32	RW	0000_0000h
F02Ch	HDR OUPTUT pipe Colorspace Converter (CSCO) component 1 pre-offset (HDR_OPIPE_CSC_IO_1)	32	RW	0000_0000h
F030h	HDR OUPUT pipe: Colorspace Converter (CSCO) component 2 pre-offset (HDR_OPIPE_CSC_IO_2)	32	RW	0000_0000h
F034h	HDR OUPTU pipe Colorspace Converter (CSCO) component 0 clip min. (HDR_OPIPE_CSC_MIN_0)	32	RW	0000_0000h
F038h	HDR OUPUT pipe Colorspace Converter (CSCO) component 1 clip min. (HDR_OPIPE_CSC_MIN_1)	32	RW	0000_0000h
F03Ch	HDR OUPTU pipe Colorspace Converter (CSCO) component 2 clip min. (HDR_OPIPE_CSC_MIN_2)	32	RW	0000_0000h
F040h	HDR OUPTUT pipe Colorspace Converter O (CSC) component 0 clip max value. (HDR_OPIPE_CSC_MAX_0)	32	RW	0000_0000h
F044h	HDR OUTPUT pipe Colorspace Converter (CSCO) component 1 clip max value. (HDR_OPIPE_CSC_MAX_1)	32	RW	0000_0000h
F048h	HDR OUTPUT pipe Colorspace Converter (CSCO) component 2 clip max value. (HDR_OPIPE_CSC_MAX_2)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
F04Ch	HDR OUPUT pipe Colorspace Converter (CSCO) normalization factor (HDR_OPIPE_CSC_NORM)	32	RW	0000_0000h
F050h	HDR OUPTUT pipe Colorspace Converter (CSC): Post offset component 0 (HDR_OPIPE_CSC_OO_0)	32	RW	0000_0000h
F054h	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset component 1 (HDR_OPIPE_CSC_OO_1)	32	RW	0000_0000h
F058h	HDR OUPTUT pipe Colorspace Converter (CSC): Post offset component 2 (HDR_OPIPE_CSC_OO_2)	32	RW	0000_0000h
F05Ch	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 0 (HDR_OPIPE_CSC_OMIN_0)	32	RW	0000_0000h
F060h	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 1 (HDR_OPIPE_CSC_OMIN_1)	32	RW	0000_0000h
F064h	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 2 (HDR_OPIPE_CSC_OMIN_2)	32	RW	0000_0000h
F068h	HDR OUPTUT pipe Colorspace Converter (CSC): Post offset max clip value for component 0 (HDR_OPIPE_CSC_OMAX_0)	32	RW	0000_0000h
F06Ch	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset max clip value for component 1 (HDR_OPIPE_CSC_OMAX_1)	32	RW	0000_0000h
F070h	HDR OUTPUT pipe Colorspace Converter (CSC): Post offset max clip value for component 2 (HDR_OPIPE_CSC_OMAX_2)	32	RW	0000_0000h
F874h	HDR OUTPUT -TO NON LINEAR pipeline control (HDR_OPIPE_2NL_CONTROL_REG)	32	RW	0000_0000h

15.10.3.1.2 A0 component Look-Up-Table. (LUT) (PIPE1_A0_LUT)

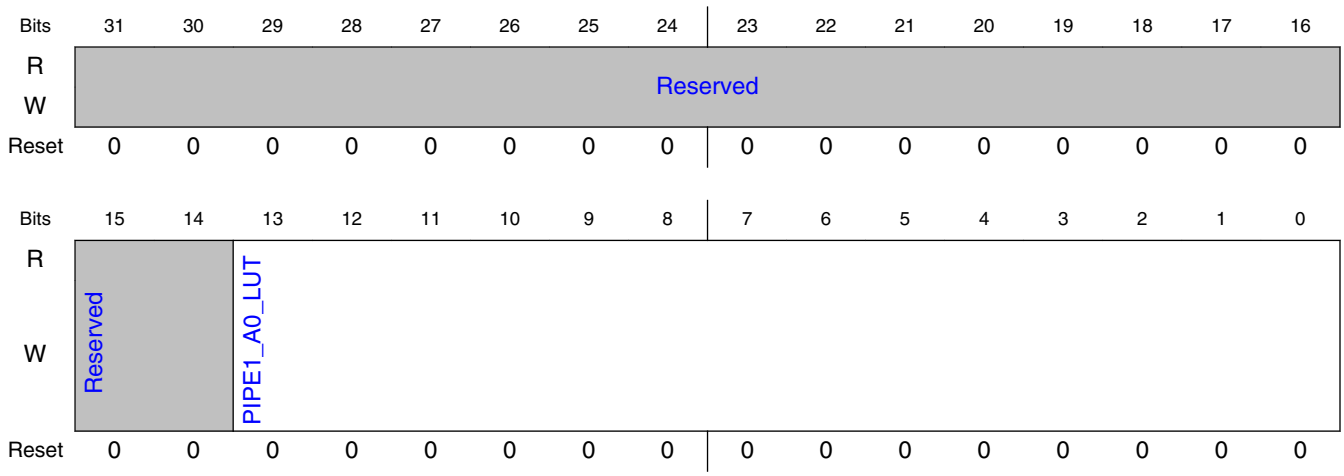
15.10.3.1.2.1 Offset

Register	Offset
PIPE1_A0_LUT	0h

15.10.3.1.2.2 Function

The LUT table has 1024 entries. A0 component may be R or Y or other

15.10.3.1.2.3 Diagram



15.10.3.1.2.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE1_A0_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.3 A1 component Look-Up-Table. (LUT) (PIPE1_A1_LUT)

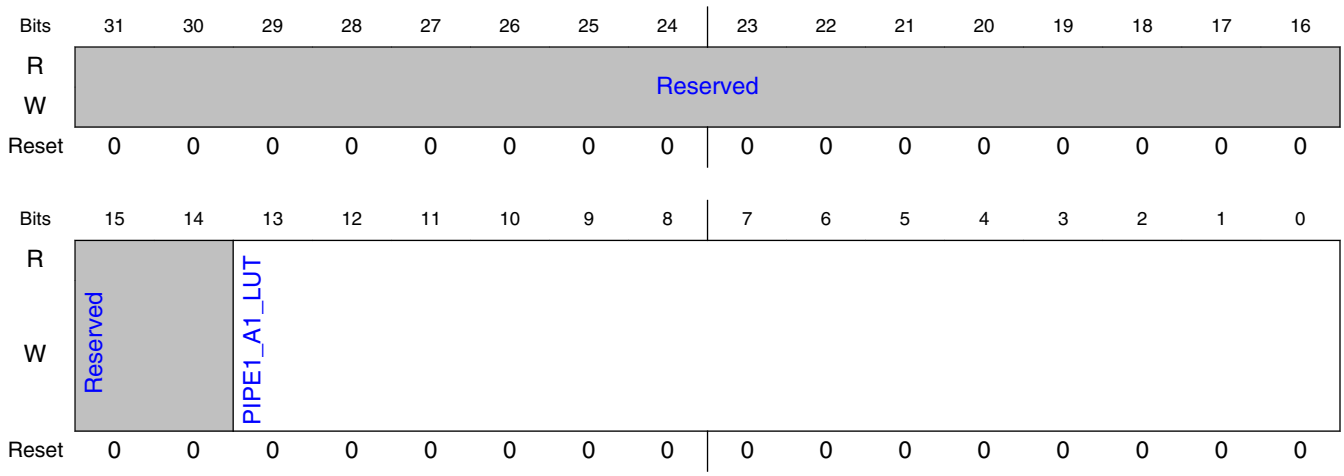
15.10.3.1.3.1 Offset

Register	Offset
PIPE1_A1_LUT	1000h

15.10.3.1.3.2 Function

The LUT table has 1024 entries. A1 component may be G or Cb or other

15.10.3.1.3.3 Diagram



15.10.3.1.3.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE1_A1_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.4 A2 component Look-Up-Table. (LUT) (PIPE1_A2_LUT)

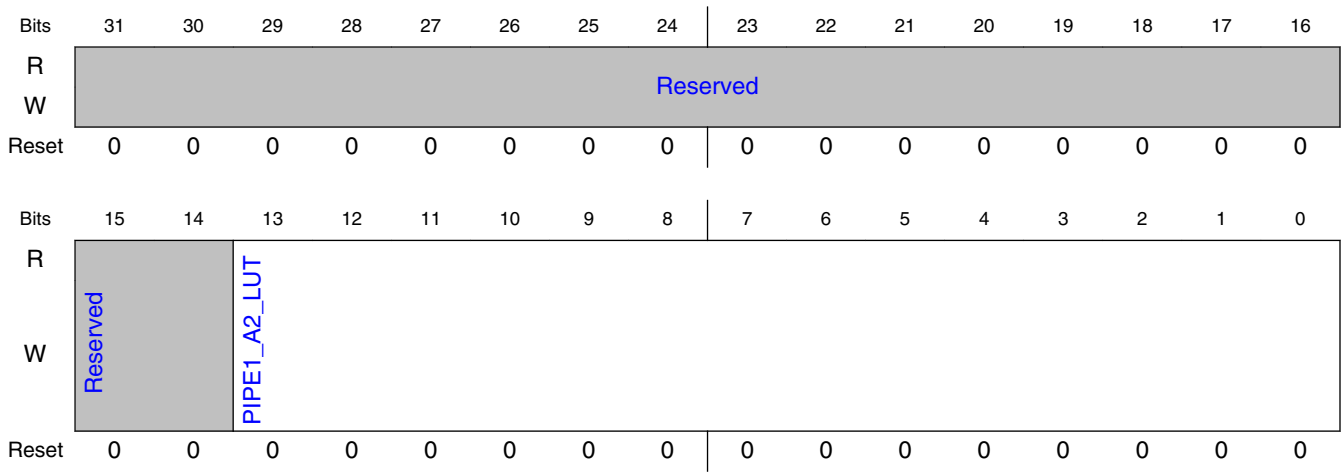
15.10.3.1.4.1 Offset

Register	Offset
PIPE1_A2_LUT	2000h

15.10.3.1.4.2 Function

The LUT table has 1024 entries. A2 component may be B or Cr or other

15.10.3.1.4.3 Diagram



15.10.3.1.4.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE1_A2_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.5 Pipe1 Colorspace Converter A control. (HDR_PIPE1_CSCA_CONTROL_REG)

15.10.3.1.5.1 Offset

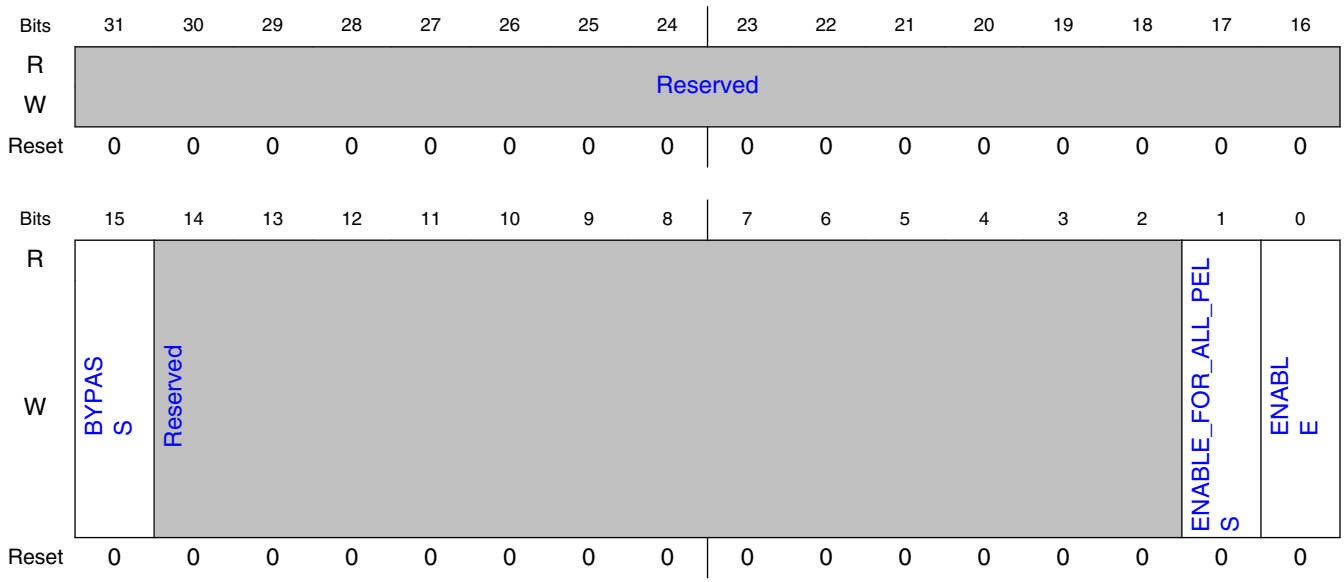
Register	Offset
HDR_PIPE1_CSCA_CONTROL_REG	3000h

15.10.3.1.5.2 Function

Controls the Color-Space-Converter-A (CSCA) in pipe1. This CSCA takes in 10-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of

the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 10 bit per component pixels. After CSCA the pixels are fed into a LUT. Pipe1 is also called channel1.

15.10.3.1.5.3 Diagram



15.10.3.1.5.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.6 Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE1_CSCA_H00)

15.10.3.1.6.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H00	3004h

15.10.3.1.6.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.6.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.7 Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE1_CSCA_H10)

15.10.3.1.7.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H10	3008h

15.10.3.1.7.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.7.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.8 Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE1_CSCA_H20)

15.10.3.1.8.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H20	300Ch

15.10.3.1.8.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.8.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.9 Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE1_CSCA_H01)

15.10.3.1.9.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H01	3010h

15.10.3.1.9.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.9.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.10 Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE1_CSCA_H11)

15.10.3.1.10.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H11	3014h

15.10.3.1.10.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.10.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.10.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.11 Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE1_CSCA_H21)

15.10.3.1.11.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H21	3018h

15.10.3.1.11.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.11.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.11.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.12 Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE1_CSCA_H02)

15.10.3.1.12.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H02	301Ch

15.10.3.1.12.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.12.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.12.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.13 Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE1_CSCA_H12)

15.10.3.1.13.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H12	3020h

15.10.3.1.13.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.13.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.13.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.14 Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE1_CSCA_H22)

15.10.3.1.14.1 Offset

Register	Offset
HDR_PIPE1_CSCA_H22	3024h

15.10.3.1.14.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.14.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.14.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.15 Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE1_CSCA_IO_0)

15.10.3.1.15.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_0	3028h

15.10.3.1.15.2 Function

An signed 10-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.15.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMPO_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.15.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.16 Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE1_CSCA_IO_1)

15.10.3.1.16.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_1	302Ch

15.10.3.1.16.2 Function

An signed 10-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.16.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.16.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.17 Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE1_CSCA_IO_2)

15.10.3.1.17.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_2	3030h

15.10.3.1.17.2 Function

An signed 10-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.17.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.17.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.18 Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE1_CSCA_IO_MIN_0)

15.10.3.1.18.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MIN_0	3034h

15.10.3.1.18.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.18.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP0_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.18.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.19 Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE1_CSCA_IO_MIN_1)

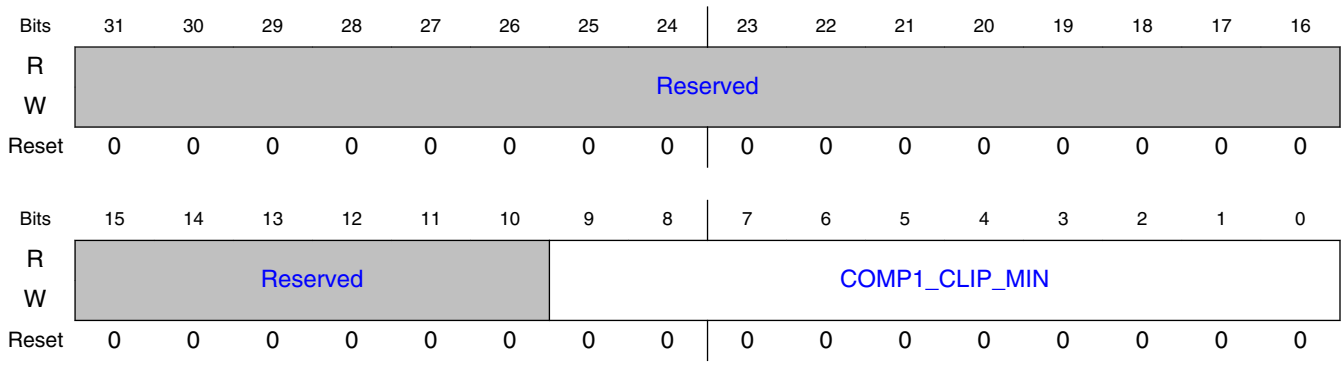
15.10.3.1.19.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MIN_1	3038h

15.10.3.1.19.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.19.3 Diagram



15.10.3.1.19.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.20 Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE1_CSCA_IO_MIN_2)

15.10.3.1.20.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MIN_2	303Ch

15.10.3.1.20.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.20.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.20.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.21 Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE1_CSCA_IO_MAX_0)

15.10.3.1.21.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MAX_0	3040h

15.10.3.1.21.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.21.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP0_CLIP_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.21.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.22 Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE1_CSCA_IO_MAX_1)

15.10.3.1.22.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MAX_1	3044h

15.10.3.1.22.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.22.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_CLIP_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.22.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.23 Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE1_CSCA_IO_MAX_2)

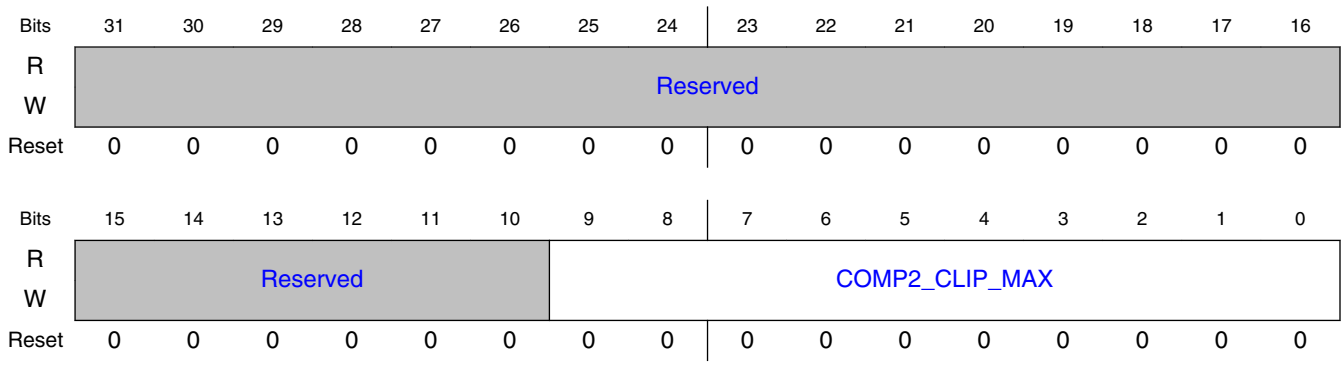
15.10.3.1.23.1 Offset

Register	Offset
HDR_PIPE1_CSCA_IO_MAX_2	3048h

15.10.3.1.23.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.23.3 Diagram



15.10.3.1.23.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.24 Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE1_CSCA_NORM)

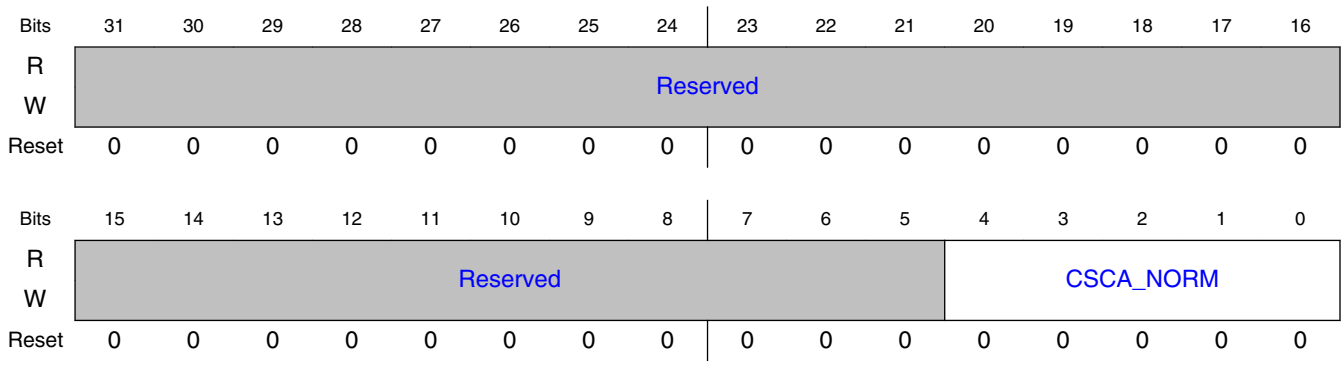
15.10.3.1.24.1 Offset

Register	Offset
HDR_PIPE1_CSCA_NORM	304Ch

15.10.3.1.24.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.24.3 Diagram



15.10.3.1.24.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCA_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.25 Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE1_CSCA_OO_0)

15.10.3.1.25.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OO_0	3050h

15.10.3.1.25.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.25.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_0	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.26 Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE1_CSCA_OO_1)

15.10.3.1.26.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OO_1	3054h

15.10.3.1.26.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.26.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.26.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_1	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.27 Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE1_CSCA_OO_2)

15.10.3.1.27.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OO_2	3058h

15.10.3.1.27.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.27.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_2											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.27.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_2	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.28 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE1_CSCA_OMIN_0)

15.10.3.1.28.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMIN_0	305Ch

15.10.3.1.28.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.28.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.28.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.29 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE1_CSCA_OMIN_1)

15.10.3.1.29.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMIN_1	3060h

15.10.3.1.29.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.29.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.29.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.30 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE1_CSCA_OMIN_2)

15.10.3.1.30.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMIN_2	3064h

15.10.3.1.30.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.30.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.30.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.31 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE1_CSCA_OMAX_0)

15.10.3.1.31.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMAX_0	3068h

15.10.3.1.31.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.31.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.31.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.32 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE1_CSCA_OMAX_1)

15.10.3.1.32.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMAX_1	306Ch

15.10.3.1.32.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.32.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.32.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.33 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE1_CSCA_OMAX_2)

15.10.3.1.33.1 Offset

Register	Offset
HDR_PIPE1_CSCA_OMAX_2	3070h

15.10.3.1.33.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.33.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.33.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.34 PIPE1: NOT USED (HDR_PIPE1_ENTRY_29)

15.10.3.1.34.1 Offset

Register	Offset
HDR_PIPE1_ENTRY_29	3074h

15.10.3.1.34.2 Function

PIPE1: NOT USED

15.10.3.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.34.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.35 Pipe1 LUT control register (HDR_PIPE1_LUT_CONTROL_REG)

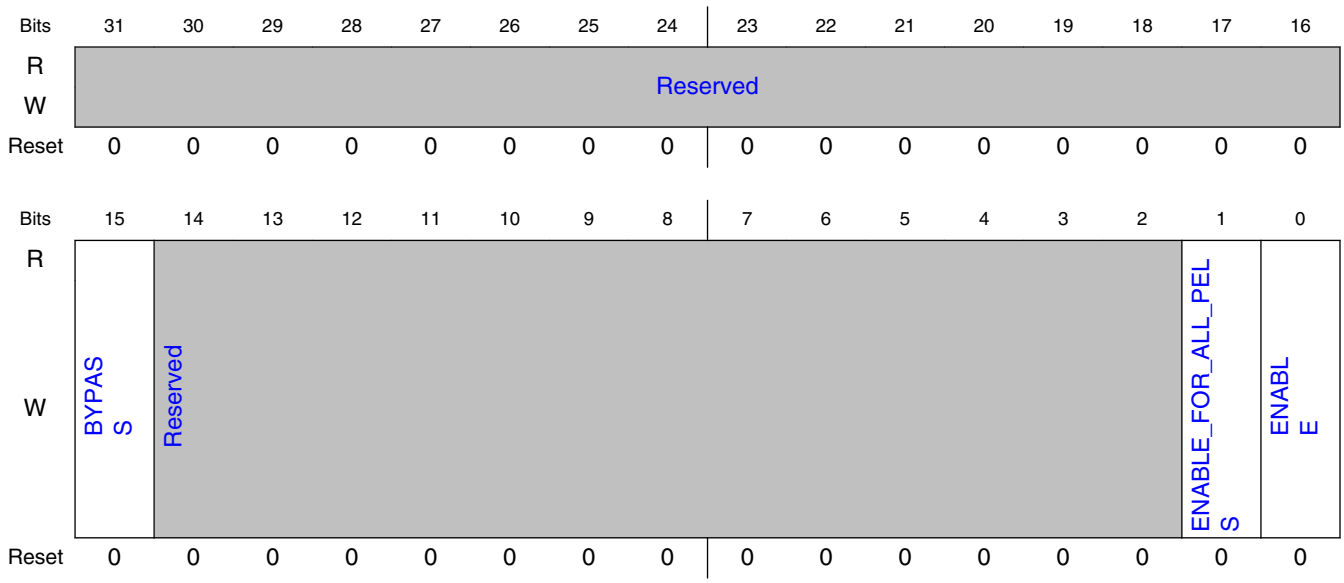
15.10.3.1.35.1 Offset

Register	Offset
HDR_PIPE1_LUT_CONTROL_REG	3080h

15.10.3.1.35.2 Function

Controls the Look-Up-Table in pipe1. Control of LUT is PER PIXEL not per component. LUT has 1024 14-bit, per component, entries. LUT maybe used for conversion from non-linear (gamma corrected) to linear pixels. LUT maybe used for conversion from linear to non linear pixels (gamma corrected). LUT can contain 14-bit fixed point or 14-bit floating point pixel value. Floating point format has 9 bit unsigned mantissa with hidden bit and 5 bit, unsigned, biased exponent. {exp[4:0], mantissa[8:0]} => 1.mantissa[8:0] * 2^(exp-1) When using floating piont numbers, in this LUT, these are used in HDR OUTPUT PIPE for HDR non-linear operation and must bypass Color Space Converter B (CSCB). Pipe1 is also called channel1.

15.10.3.1.35.3 Diagram



15.10.3.1.35.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this LUT 1: Pixels pass thru this LUT unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This LUT is enabled only for blened pixels 1: This LUT is enabled all pixels
0 ENABLE	0: Don't enable this LUT: Pixels pass thru the LUT unmodiifed 1: This LUT is enabled for current picture

15.10.3.1.36 Pipe1 Colorspace Converter B control. (HDR_PIPE1_CSCB_CONTROL_REG)

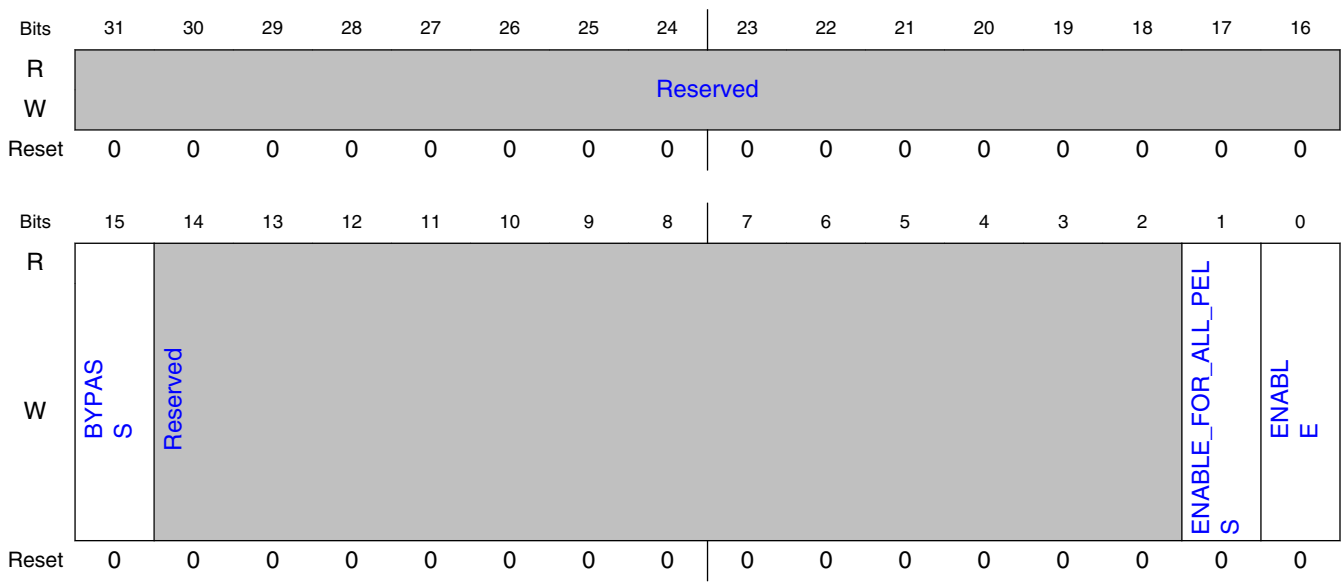
15.10.3.1.36.1 Offset

Register	Offset
HDR_PIPE1_CSCB_CONTROL_REG	3800h

15.10.3.1.36.2 Function

Controls the Color-Space-Converter-B (CSCB) in pipe1. This CSCB takes in 14-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 28 bit per component pixels. After CSCB the pixels are fed into the Blender. Pipe1 is also called channel1.

15.10.3.1.36.3 Diagram



15.10.3.1.36.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.

Table continues on the next page...

Field	Function
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.37 Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE1_CSCB_H00)

15.10.3.1.37.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H00	3804h

15.10.3.1.37.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.37.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.37.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.38 Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE1_CSCB_H10)

15.10.3.1.38.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H10	3808h

15.10.3.1.38.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.38.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.38.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.39 Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE1_CSCB_H20)

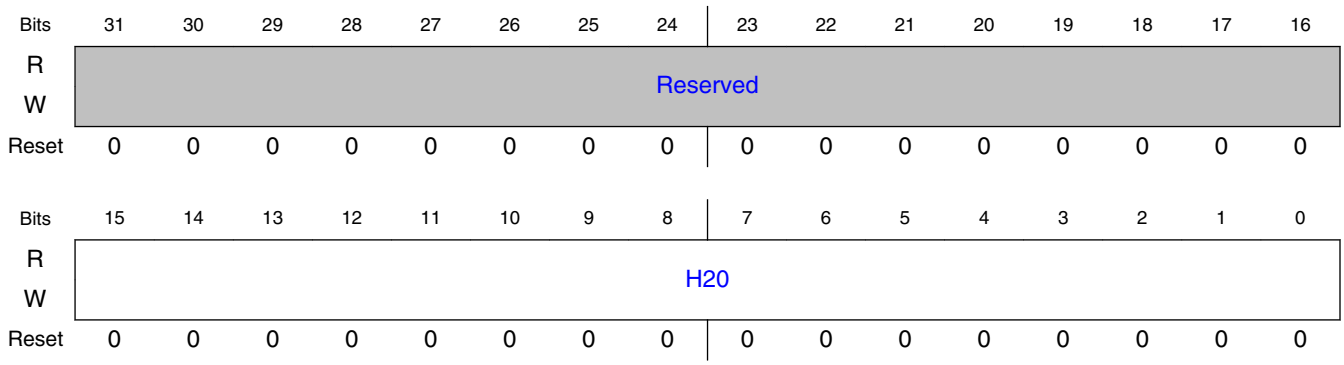
15.10.3.1.39.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H20	380Ch

15.10.3.1.39.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.39.3 Diagram



15.10.3.1.39.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.40 Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE1_CSCB_H01)

15.10.3.1.40.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H01	3810h

15.10.3.1.40.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.40.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.40.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.41 Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE1_CSCB_H11)

15.10.3.1.41.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H11	3814h

15.10.3.1.41.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.41.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.41.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.42 Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE1_CSCB_H21)

15.10.3.1.42.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H21	3818h

15.10.3.1.42.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.42.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.42.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.43 Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE1_CSCB_H02)

15.10.3.1.43.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H02	381Ch

15.10.3.1.43.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.43.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.43.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.44 Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE1_CSCB_H12)

15.10.3.1.44.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H12	3820h

15.10.3.1.44.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.44.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.44.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.45 Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE1_CSCB_H22)

15.10.3.1.45.1 Offset

Register	Offset
HDR_PIPE1_CSCB_H22	3824h

15.10.3.1.45.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.45.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.45.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.46 Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE1_CSCB_IO_0)

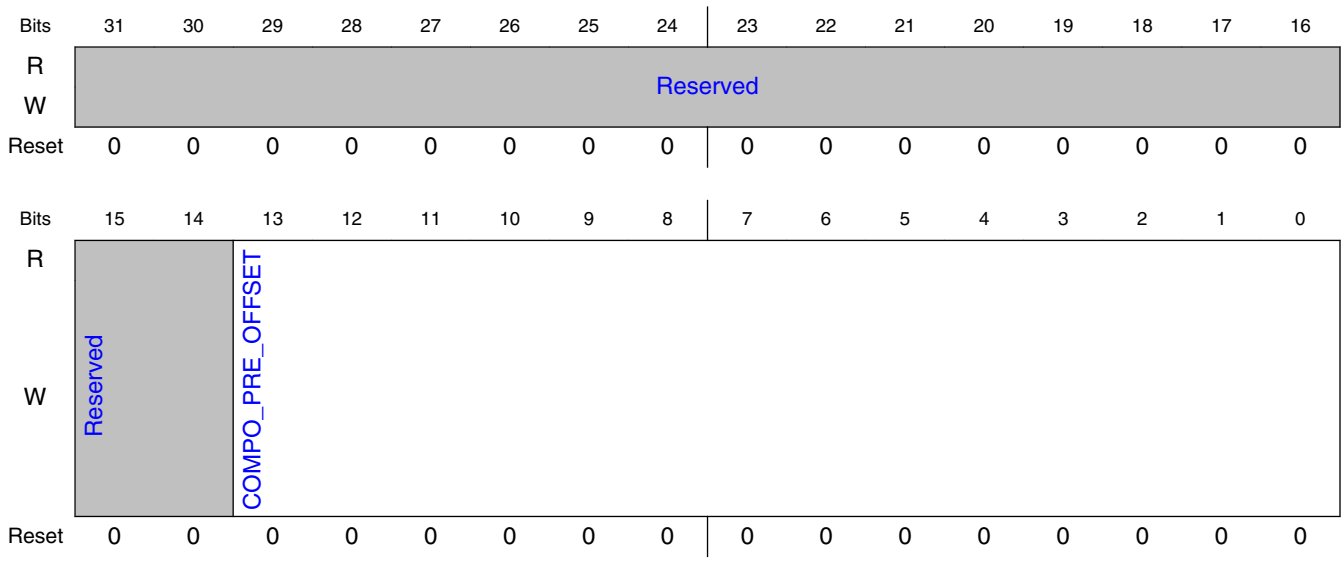
15.10.3.1.46.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_0	3828h

15.10.3.1.46.2 Function

A signed 14-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.46.3 Diagram



15.10.3.1.46.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.47 Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE1_CSCB_IO_1)

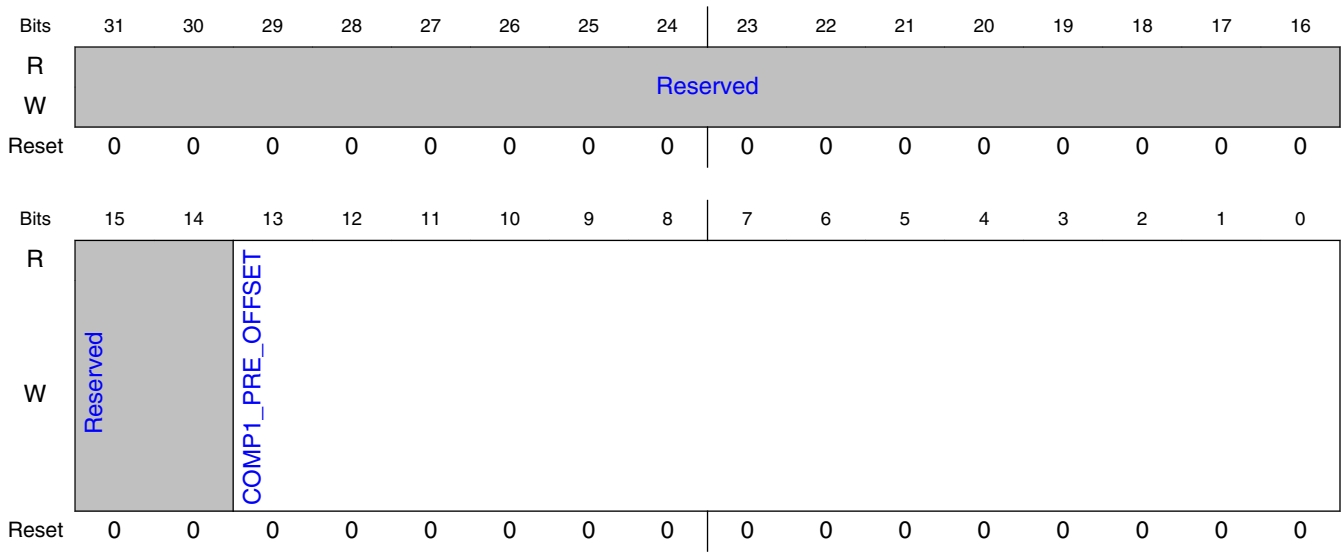
15.10.3.1.47.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_1	382Ch

15.10.3.1.47.2 Function

A signed 14-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.47.3 Diagram



15.10.3.1.47.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.48 Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE1_CSCB_IO_2)

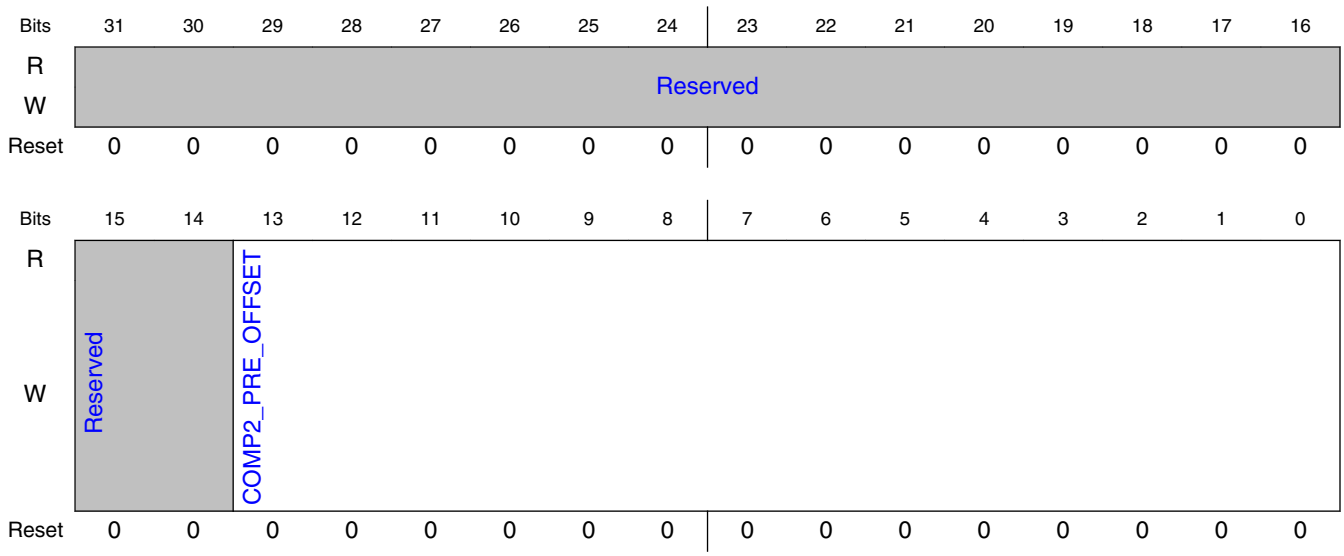
15.10.3.1.48.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_2	3830h

15.10.3.1.48.2 Function

A signed 14-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.48.3 Diagram



15.10.3.1.48.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.49 Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE1_CSCB_IO_MIN_0)

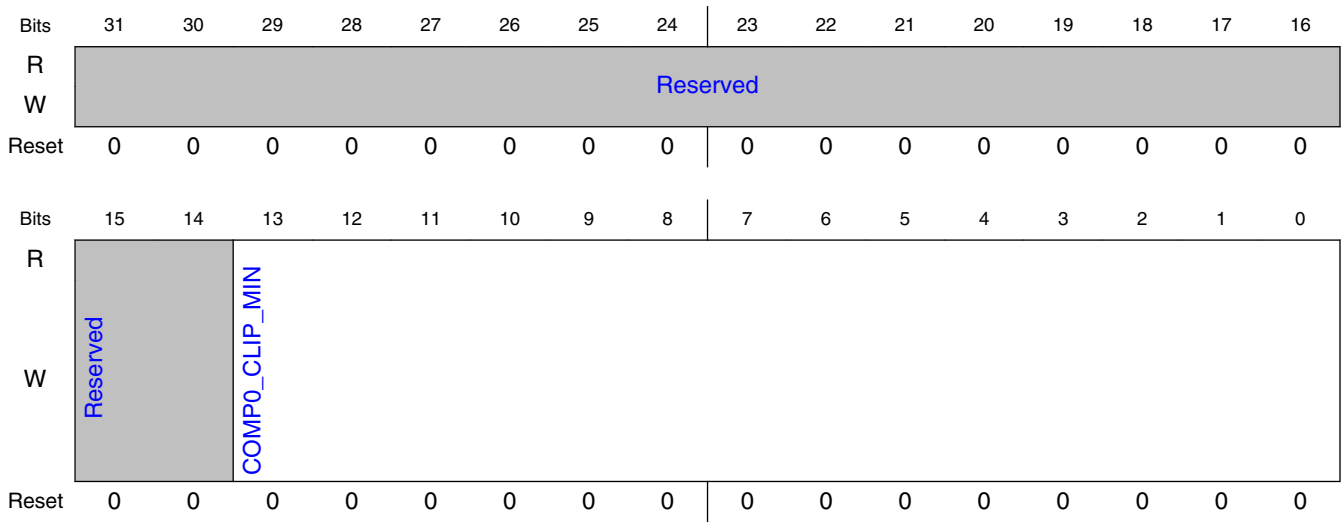
15.10.3.1.49.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MIN_0	3834h

15.10.3.1.49.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.49.3 Diagram



15.10.3.1.49.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.50 Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE1_CSCB_IO_MIN_1)

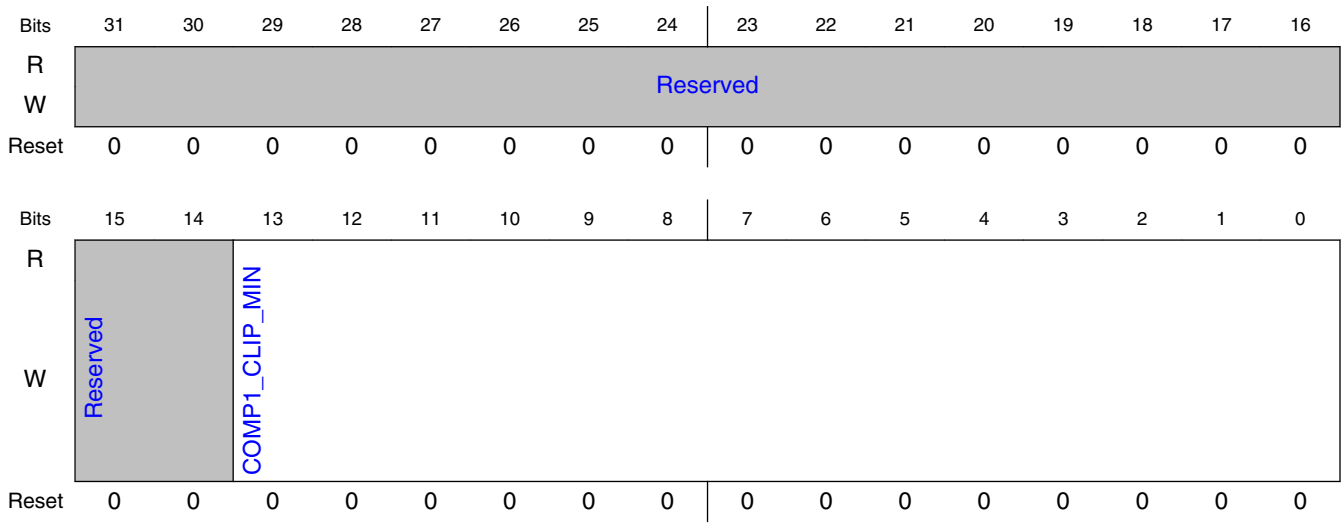
15.10.3.1.50.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MIN_1	3838h

15.10.3.1.50.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.50.3 Diagram



15.10.3.1.50.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.51 Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE1_CSCB_IO_MIN_2)

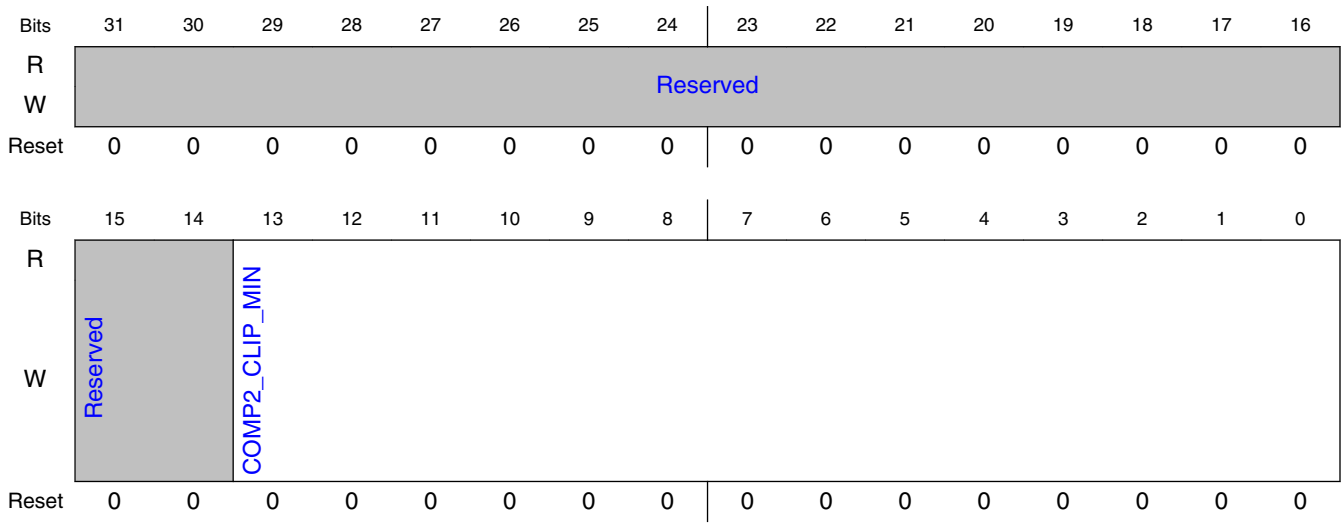
15.10.3.1.51.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MIN_2	383Ch

15.10.3.1.51.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 1891]. Pipe1 is also called channel1.

15.10.3.1.51.3 Diagram



15.10.3.1.51.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.52 Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE1_CSCB_IO_MAX_0)

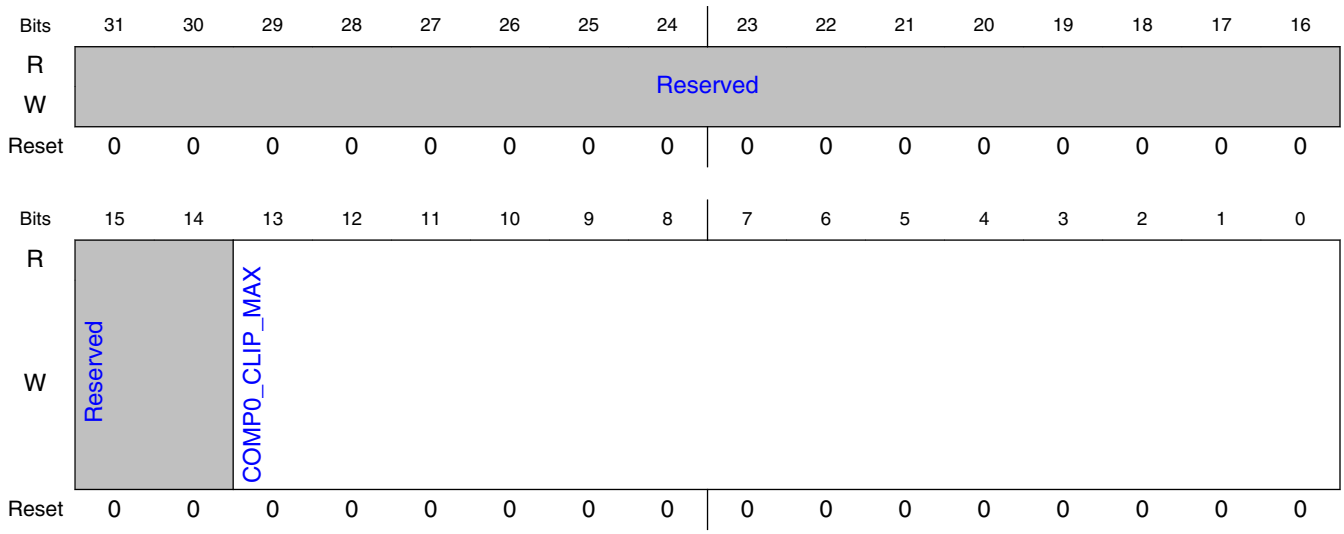
15.10.3.1.52.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MAX_0	3840h

15.10.3.1.52.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0, 16383]. Pipe1 is also called channel1.

15.10.3.1.52.3 Diagram



15.10.3.1.52.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.53 Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE1_CSCB_IO_MAX_1)

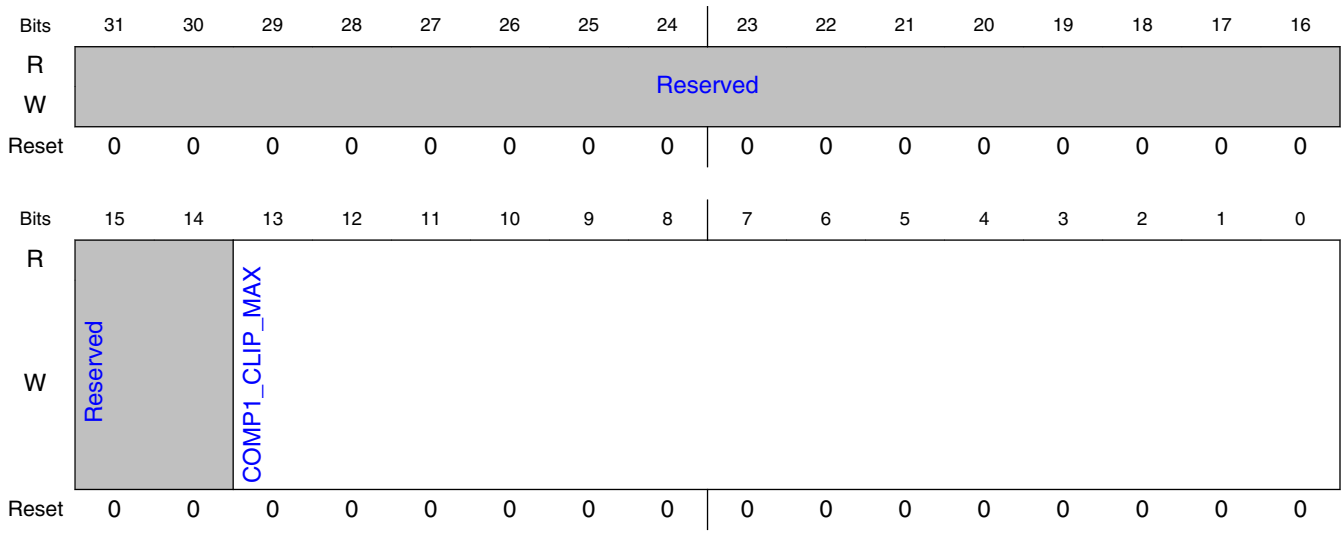
15.10.3.1.53.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MAX_1	3844h

15.10.3.1.53.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.53.3 Diagram



15.10.3.1.53.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.54 Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE1_CSCB_IO_MAX_2)

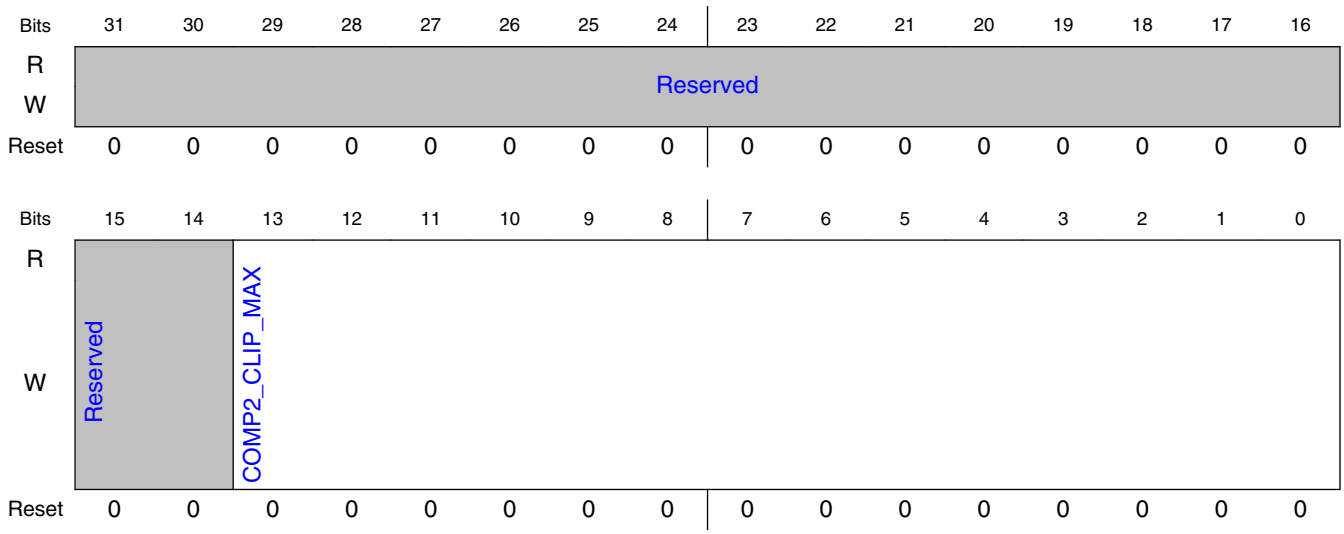
15.10.3.1.54.1 Offset

Register	Offset
HDR_PIPE1_CSCB_IO_MAX_2	3848h

15.10.3.1.54.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.54.3 Diagram



15.10.3.1.54.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.55 Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE1_CSCB_NORM)

15.10.3.1.55.1 Offset

Register	Offset
HDR_PIPE1_CSCB_NORM	384Ch

15.10.3.1.55.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.55.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved											CSCB_NORM				
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.55.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCB_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.56 Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE1_CSCB_OO_0)

15.10.3.1.56.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OO_0	3850h

15.10.3.1.56.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.56.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_0												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.56.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_0	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.57 Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE1_CSCB_OO_1)

15.10.3.1.57.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OO_1	3854h

15.10.3.1.57.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.57.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_1												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.57.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_1	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.58 Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE1_CSCB_OO_2)

15.10.3.1.58.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OO_2	3858h

15.10.3.1.58.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.58.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_2												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.58.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_2	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.59 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE1_CSCB_OMIN_0)

15.10.3.1.59.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMIN_0	385Ch

15.10.3.1.59.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.59.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.59.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.60 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE1_CSCB_OMIN_1)

15.10.3.1.60.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMIN_1	3860h

15.10.3.1.60.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.60.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.60.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.61 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE1_CSCB_OMIN_2)

15.10.3.1.61.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMIN_2	3864h

15.10.3.1.61.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.61.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.61.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minimum clipped pixel component.

15.10.3.1.62 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE1_CSCB_OMAX_0)

15.10.3.1.62.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMAX_0	3868h

15.10.3.1.62.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.62.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				POST_OFF_MAX											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	POST_OFF_MAX															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.62.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximum clipped pixel component.

15.10.3.1.63 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE1_CSCB_OMAX_1)

15.10.3.1.63.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMAX_1	386Ch

15.10.3.1.63.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.63.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.63.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.64 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE1_CSCB_OMAX_2)

15.10.3.1.64.1 Offset

Register	Offset
HDR_PIPE1_CSCB_OMAX_2	3870h

15.10.3.1.64.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.64.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MAX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.64.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.65 Pipe1 floating point to fixed point control (HDR_PIPE1_FL2FX)

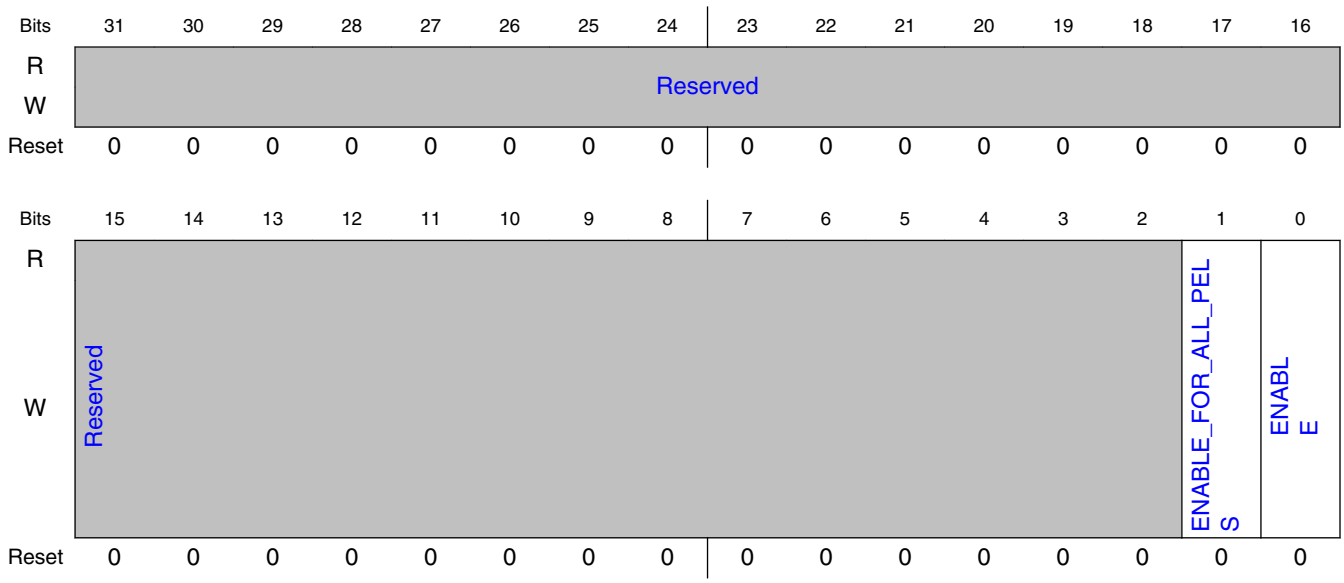
15.10.3.1.65.1 Offset

Register	Offset
HDR_PIPE1_FL2FX	3874h

15.10.3.1.65.2 Function

Pipe1 floating point to fixed point control Pipe1 is also called channel1.

15.10.3.1.65.3 Diagram



15.10.3.1.65.4 Fields

Field	Function
31-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This Float to Fixed operation is enabled only for blended pixels 1: This Float to Fixed operation is enabled all pixels
0 ENABLE	0: Don't enable this Float-to-Fixed converter: Pixels pass thru the Float-to-Fixed unmodiifed 1: This Float-to-Fixed converter is enabled for current picture

15.10.3.1.66 PIPE1: NOT USED (HDR_PIPE1_ENTRY_30)

15.10.3.1.66.1 Offset

Register	Offset
HDR_PIPE1_ENTRY_30	3878h

15.10.3.1.66.2 Function

PIPE1: NOT USED

15.10.3.1.66.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.66.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.67 A0 component Look-Up-Table. (LUT) (PIPE2_A0_LUT)

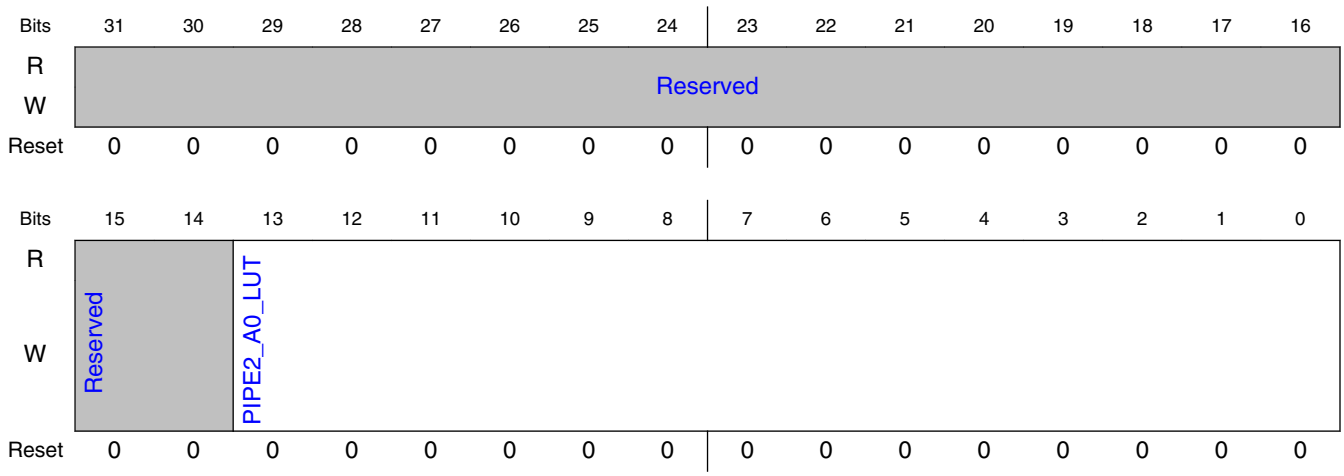
15.10.3.1.67.1 Offset

Register	Offset
PIPE2_A0_LUT	4000h

15.10.3.1.67.2 Function

The LUT table has 1024 entries. A0 component may be R or Y or other

15.10.3.1.67.3 Diagram



15.10.3.1.67.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE2_A0_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.68 A1 component Look-Up-Table. (LUT) (PIPE2_A1_LUT)

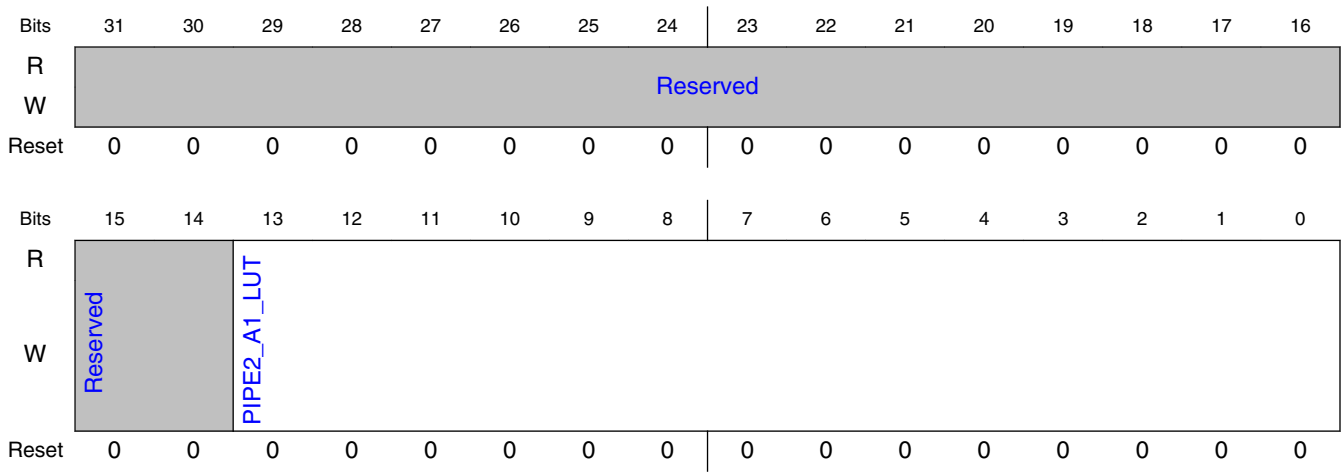
15.10.3.1.68.1 Offset

Register	Offset
PIPE2_A1_LUT	5000h

15.10.3.1.68.2 Function

The LUT table has 1024 entries. A1 component may be G or Cb or other

15.10.3.1.68.3 Diagram



15.10.3.1.68.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE2_A1_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.69 A2 component Look-Up-Table. (LUT) (PIPE2_A2_LUT)

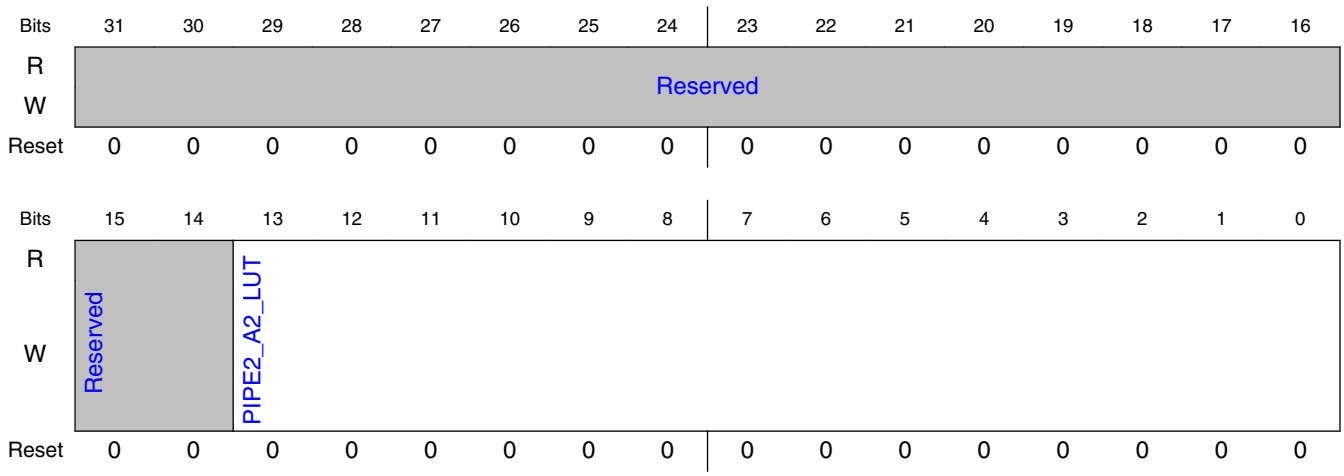
15.10.3.1.69.1 Offset

Register	Offset
PIPE2_A2_LUT	6000h

15.10.3.1.69.2 Function

The LUT table has 1024 entries. A2 component may be B or Cr or other

15.10.3.1.69.3 Diagram



15.10.3.1.69.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE2_A2_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.70 Pipe1 Colorspace Converter A control. (HDR_PIPE2_CSCA_CONTROL_REG)

15.10.3.1.70.1 Offset

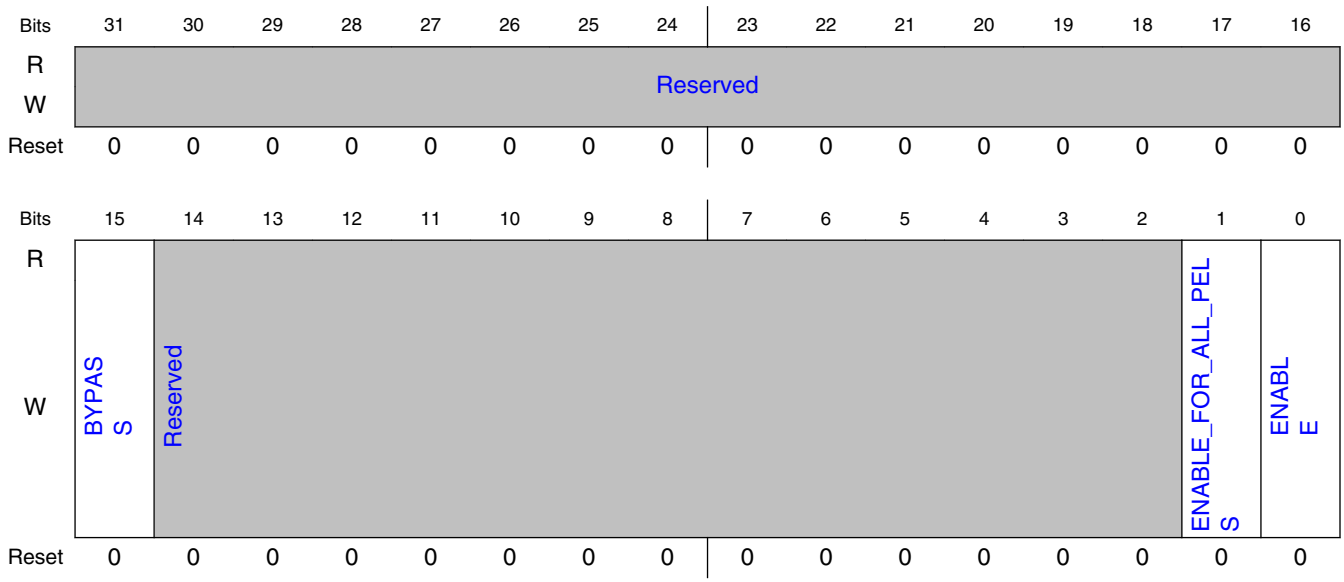
Register	Offset
HDR_PIPE2_CSCA_CONTROL_REG	7000h

15.10.3.1.70.2 Function

Controls the Color-Space-Converter-A (CSCA) in pipe1. This CSCA takes in 10-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of

the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 10 bit per component pixels. After CSCA the pixels are fed into a LUT. Pipe1 is also called channel1.

15.10.3.1.70.3 Diagram



15.10.3.1.70.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.71 Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE2_CSCA_H00)

15.10.3.1.71.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H00	7004h

15.10.3.1.71.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.71.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.71.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.72 Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE2_CSCA_H10)

15.10.3.1.72.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H10	7008h

15.10.3.1.72.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.72.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.72.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.73 Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE2_CSCA_H20)

15.10.3.1.73.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H20	700Ch

15.10.3.1.73.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.73.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.73.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.74 Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE2_CSCA_H01)

15.10.3.1.74.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H01	7010h

15.10.3.1.74.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.74.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.74.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.75 Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE2_CSCA_H11)

15.10.3.1.75.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H11	7014h

15.10.3.1.75.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.75.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.75.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.76 Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE2_CSCA_H21)

15.10.3.1.76.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H21	7018h

15.10.3.1.76.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.76.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.76.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.77 Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE2_CSCA_H02)

15.10.3.1.77.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H02	701Ch

15.10.3.1.77.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.77.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.77.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.78 Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE2_CSCA_H12)

15.10.3.1.78.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H12	7020h

15.10.3.1.78.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.78.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.78.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.79 Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE2_CSCA_H22)

15.10.3.1.79.1 Offset

Register	Offset
HDR_PIPE2_CSCA_H22	7024h

15.10.3.1.79.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.79.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.79.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.80 Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE2_CSCA_IO_0)

15.10.3.1.80.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_0	7028h

15.10.3.1.80.2 Function

An signed 10-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.80.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMPO_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.80.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.81 Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE2_CSCA_IO_1)

15.10.3.1.81.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_1	702Ch

15.10.3.1.81.2 Function

An signed 10-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.81.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.81.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.82 Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE2_CSCA_IO_2)

15.10.3.1.82.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_2	7030h

15.10.3.1.82.2 Function

An signed 10-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.82.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.82.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.83 Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE2_CSCA_IO_MIN_0)

15.10.3.1.83.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MIN_0	7034h

15.10.3.1.83.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.83.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP0_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.83.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.84 Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE2_CSCA_IO_MIN_1)

15.10.3.1.84.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MIN_1	7038h

15.10.3.1.84.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.84.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.84.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.85 Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE2_CSCA_IO_MIN_2)

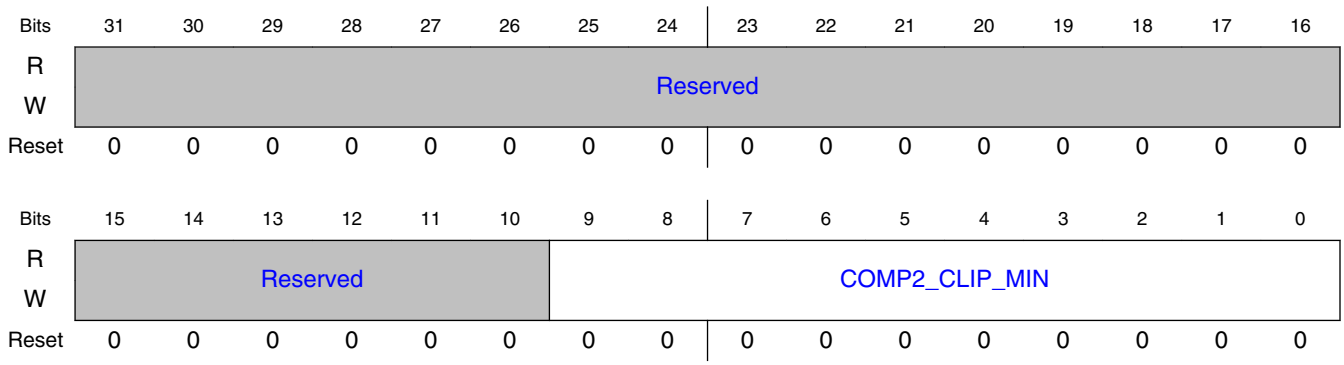
15.10.3.1.85.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MIN_2	703Ch

15.10.3.1.85.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.85.3 Diagram



15.10.3.1.85.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.86 Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE2_CSCA_IO_MAX_0)

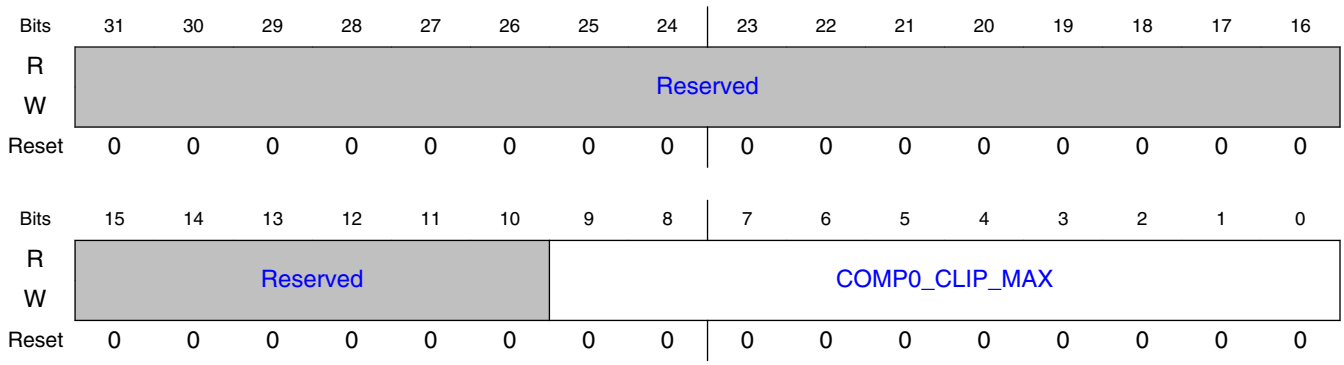
15.10.3.1.86.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MAX_0	7040h

15.10.3.1.86.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.86.3 Diagram



15.10.3.1.86.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.87 Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE2_CSCA_IO_MAX_1)

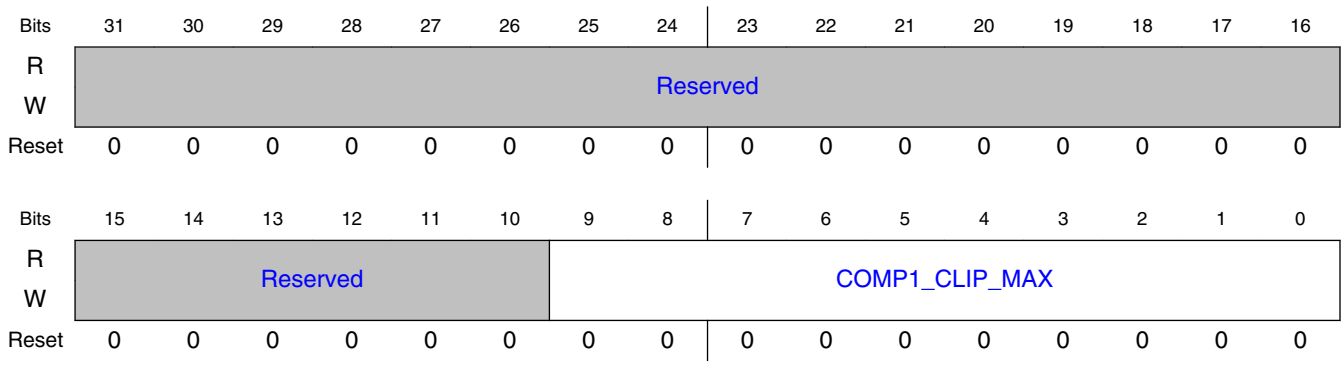
15.10.3.1.87.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MAX_1	7044h

15.10.3.1.87.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.87.3 Diagram



15.10.3.1.87.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.88 Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE2_CSCA_IO_MAX_2)

15.10.3.1.88.1 Offset

Register	Offset
HDR_PIPE2_CSCA_IO_MAX_2	7048h

15.10.3.1.88.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.88.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_CLIP_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.88.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.89 Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE2_CSCA_NORM)

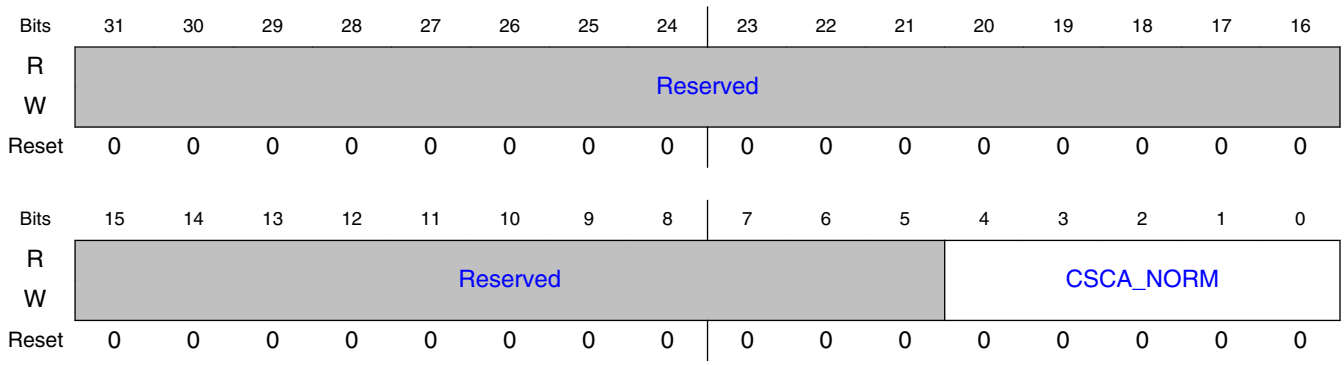
15.10.3.1.89.1 Offset

Register	Offset
HDR_PIPE2_CSCA_NORM	704Ch

15.10.3.1.89.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.89.3 Diagram



15.10.3.1.89.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCA_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.90 Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE2_CSCA_OO_0)

15.10.3.1.90.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OO_0	7050h

15.10.3.1.90.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.90.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.90.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_0	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.91 Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE2_CSCA_OO_1)

15.10.3.1.91.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OO_1	7054h

15.10.3.1.91.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.91.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.91.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_1	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.92 Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE2_CSCA_OO_2)

15.10.3.1.92.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OO_2	7058h

15.10.3.1.92.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.92.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_2											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.92.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_2	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.93 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE2_CSCA_OMIN_0)

15.10.3.1.93.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMIN_0	705Ch

15.10.3.1.93.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.93.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.93.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.94 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE2_CSCA_OMIN_1)

15.10.3.1.94.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMIN_1	7060h

15.10.3.1.94.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.94.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.94.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.95 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE2_CSCA_OMIN_2)

15.10.3.1.95.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMIN_2	7064h

15.10.3.1.95.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.95.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.95.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.96 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE2_CSCA_OMAX_0)

15.10.3.1.96.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMAX_0	7068h

15.10.3.1.96.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.96.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.96.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.97 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE2_CSCA_OMAX_1)

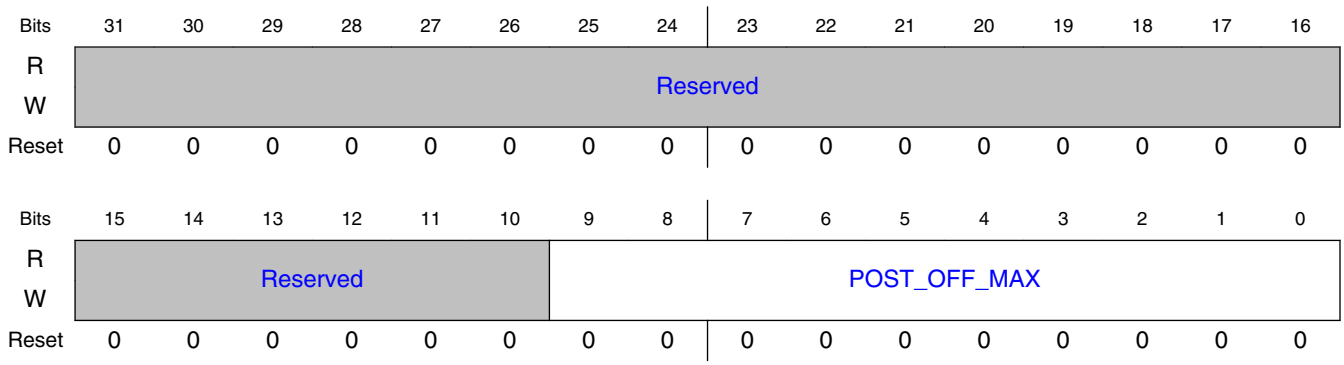
15.10.3.1.97.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMAX_1	706Ch

15.10.3.1.97.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.97.3 Diagram



15.10.3.1.97.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.98 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE2_CSCA_OMAX_2)

15.10.3.1.98.1 Offset

Register	Offset
HDR_PIPE2_CSCA_OMAX_2	7070h

15.10.3.1.98.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.98.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.98.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.99 PIPE2: NOT USED (HDR_PIPE2_ENTRY_29)

15.10.3.1.99.1 Offset

Register	Offset
HDR_PIPE2_ENTRY_29	7074h

15.10.3.1.99.2 Function

PIPE2: NOT USED

15.10.3.1.99.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.99.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.100 Pipe1 LUT control register (HDR_PIPE2_LUT_CONTROL_REG)

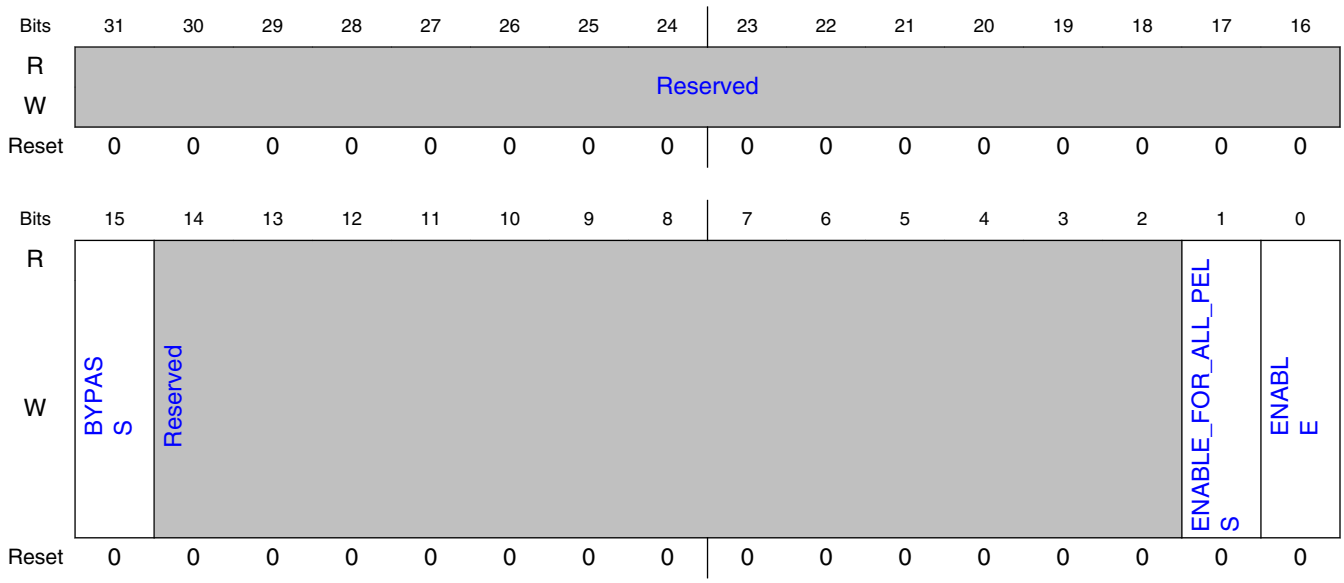
15.10.3.1.100.1 Offset

Register	Offset
HDR_PIPE2_LUT_CONTROL_REG	7080h

15.10.3.1.100.2 Function

Controls the Look-Up-Table in pipe1. Control of LUT is PER PIXEL not per component. LUT has 1024 14-bit, per component, entries. LUT maybe used for conversion from non-linear (gamma corrected) to linear pixels. LUT maybe used for conversion from linear to non linear pixels (gamma corrected). LUT can contain 14-bit fixed point or 14-bit floating point pixel value. Floating point format has 9 bit unsigned mantissa with hidden bit and 5 bit, unsigned, biased exponent. {exp[4:0], mantissa[8:0]} => 1.mantissa[8:0] * 2^(exp-1) When using floating piont numbers, in this LUT, these are used in HDR OUTPUT PIPE for HDR non-linear operation and must bypass Color Space Converter B (CSCB). Pipe1 is also called channel1.

15.10.3.1.100.3 Diagram



15.10.3.1.100.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this LUT 1: Pixels pass thru this LUT unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This LUT is enabled only for blened pixels 1: This LUT is enabled all pixels
0 ENABLE	0: Don't enable this LUT: Pixels pass thru the LUT unmodiifed 1: This LUT is enabled for current picture

15.10.3.1.101 Pipe1 Colorspace Converter B control. (HDR_PIPE2_CSCB_CONTROL_REG)

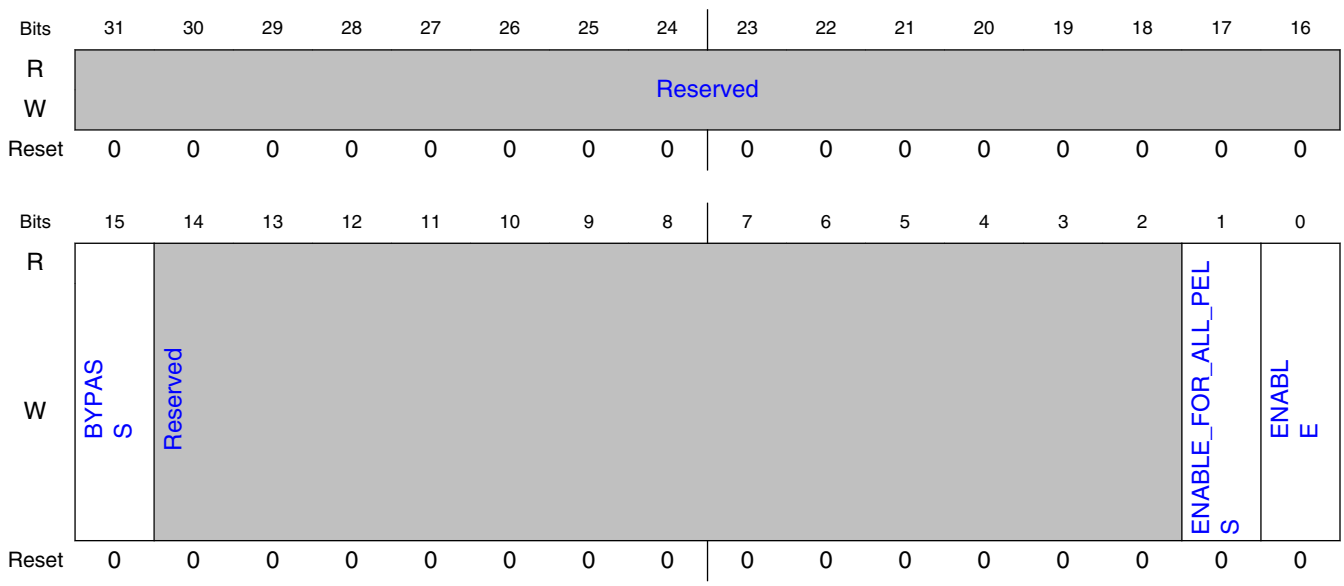
15.10.3.1.101.1 Offset

Register	Offset
HDR_PIPE2_CSCB_CONTROL_REG	7800h

15.10.3.1.101.2 Function

Controls the Color-Space-Converter-B (CSCB) in pipe1. This CSCB takes in 14-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 28 bit per component pixels. After CSCB the pixels are fed into the Blender. Pipe1 is also called channel1.

15.10.3.1.101.3 Diagram



15.10.3.1.101.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.

Table continues on the next page...

Field	Function
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.102 Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE2_CSCB_H00)

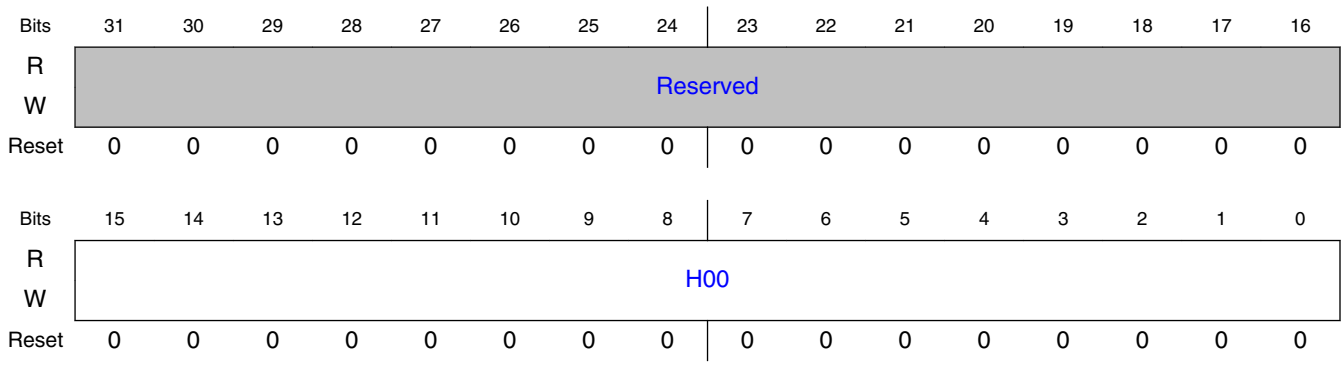
15.10.3.1.102.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H00	7804h

15.10.3.1.102.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.102.3 Diagram



15.10.3.1.102.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.103 Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE2_CSCB_H10)

15.10.3.1.103.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H10	7808h

15.10.3.1.103.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.103.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.103.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.104 Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE2_CSCB_H20)

15.10.3.1.104.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H20	780Ch

15.10.3.1.104.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.104.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.104.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.105 Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE2_CSCB_H01)

15.10.3.1.105.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H01	7810h

15.10.3.1.105.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.105.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.105.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.106 Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE2_CSCB_H11)

15.10.3.1.106.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H11	7814h

15.10.3.1.106.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.106.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.106.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.107 Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE2_CSCB_H21)

15.10.3.1.107.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H21	7818h

15.10.3.1.107.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.107.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.107.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.108 Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE2_CSCB_H02)

15.10.3.1.108.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H02	781Ch

15.10.3.1.108.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.108.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.108.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.109 Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE2_CSCB_H12)

15.10.3.1.109.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H12	7820h

15.10.3.1.109.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.109.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.109.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.110 Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE2_CSCB_H22)

15.10.3.1.110.1 Offset

Register	Offset
HDR_PIPE2_CSCB_H22	7824h

15.10.3.1.110.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.110.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.110.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.111 Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE2_CSCB_IO_0)

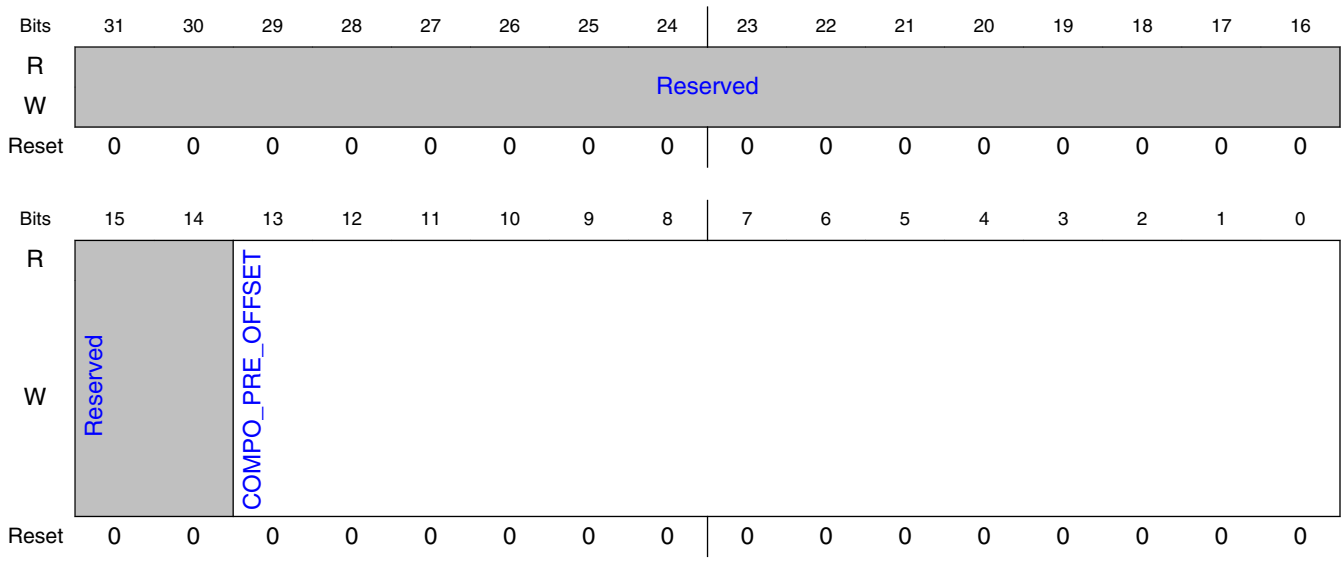
15.10.3.1.111.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_0	7828h

15.10.3.1.111.2 Function

A signed 14-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.111.3 Diagram



15.10.3.1.111.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.112 Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE2_CSCB_IO_1)

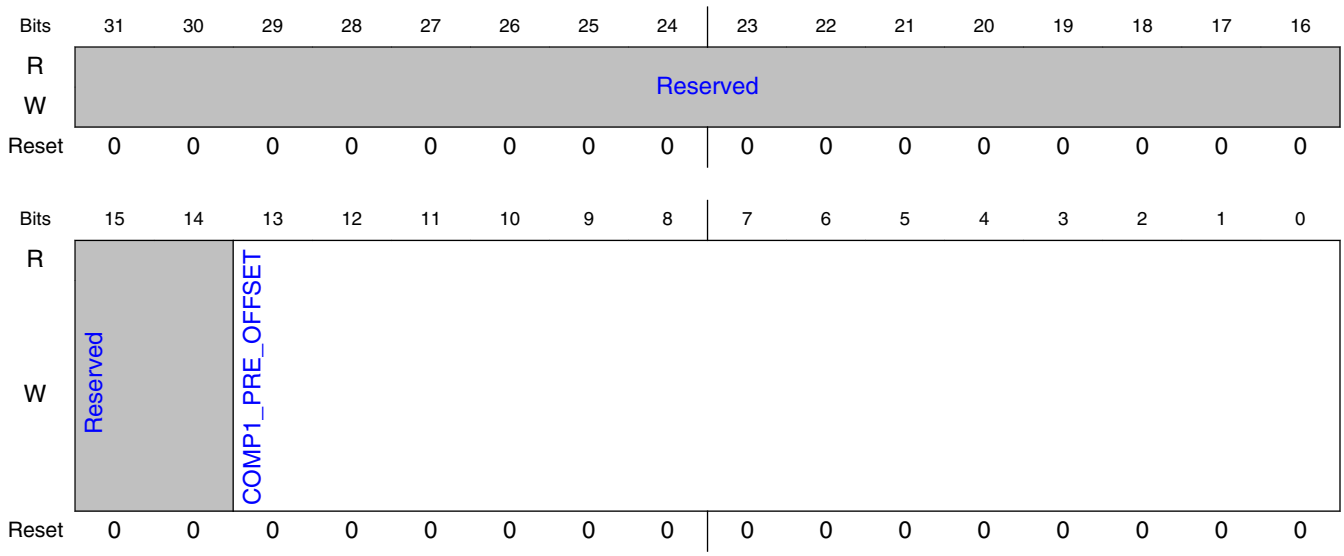
15.10.3.1.112.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_1	782Ch

15.10.3.1.112.2 Function

A signed 14-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.112.3 Diagram



15.10.3.1.112.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.113 Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE2_CSCB_IO_2)

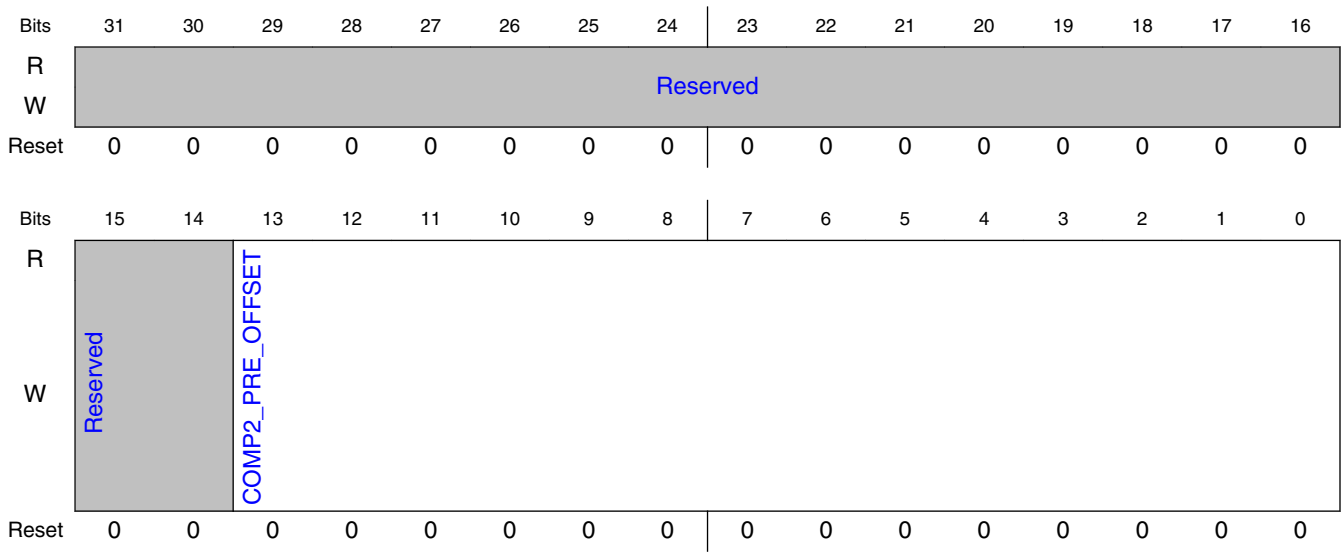
15.10.3.1.113.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_2	7830h

15.10.3.1.113.2 Function

A signed 14-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.113.3 Diagram



15.10.3.1.113.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.114 Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE2_CSCB_IO_MIN_0)

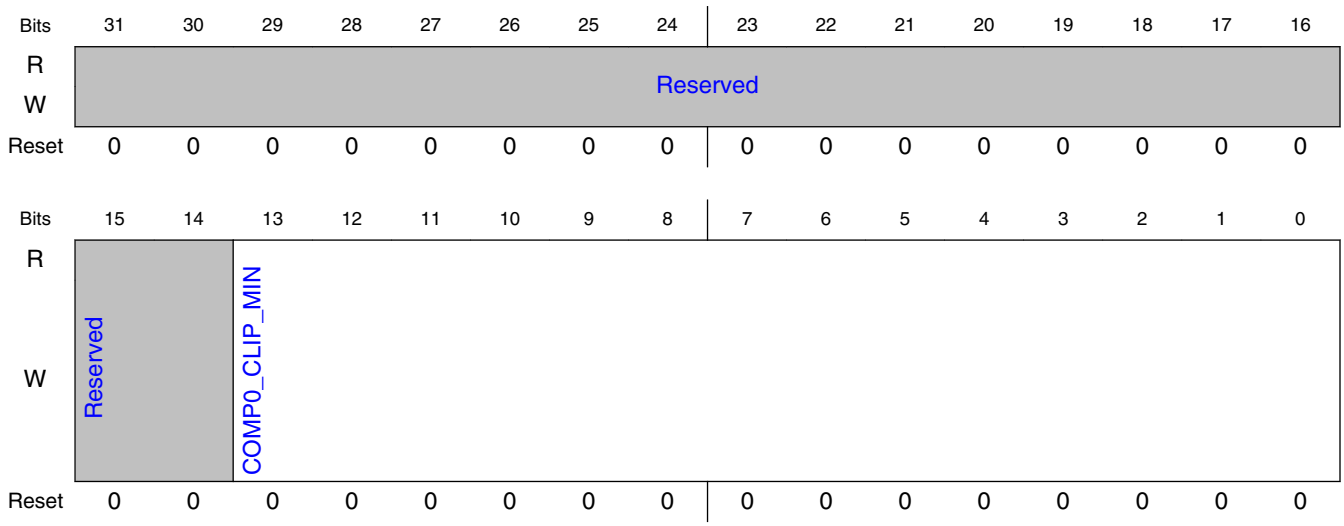
15.10.3.1.114.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MIN_0	7834h

15.10.3.1.114.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.114.3 Diagram



15.10.3.1.114.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.115 Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE2_CSCB_IO_MIN_1)

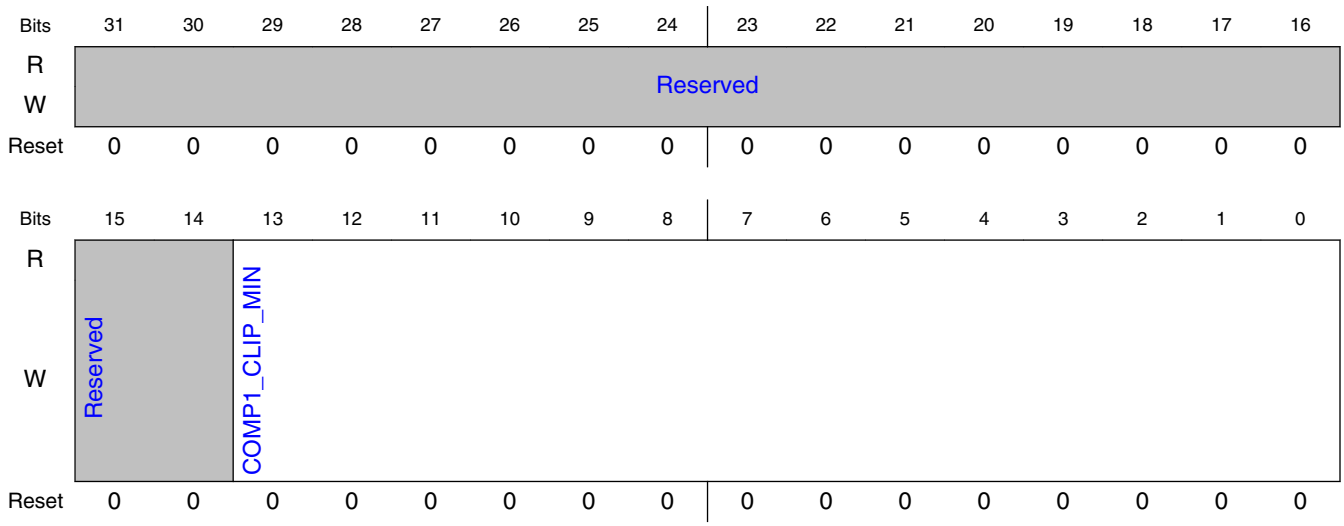
15.10.3.1.115.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MIN_1	7838h

15.10.3.1.115.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.115.3 Diagram



15.10.3.1.115.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.116 Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE2_CSCB_IO_MIN_2)

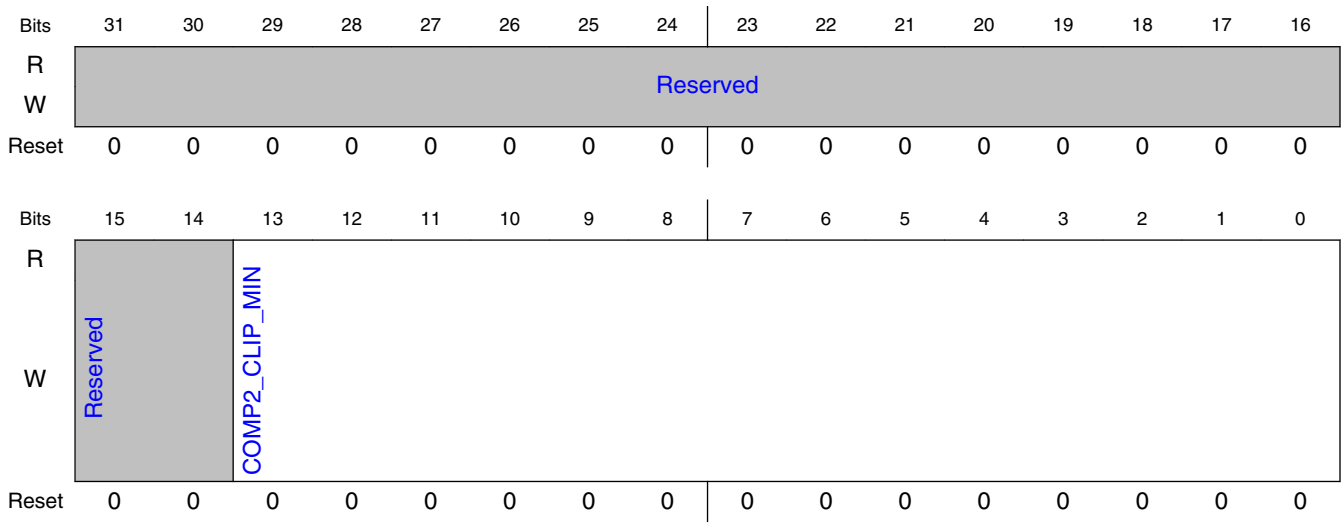
15.10.3.1.116.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MIN_2	783Ch

15.10.3.1.116.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 1891]. Pipe1 is also called channel1.

15.10.3.1.116.3 Diagram



15.10.3.1.116.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.117 Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE2_CSCB_IO_MAX_0)

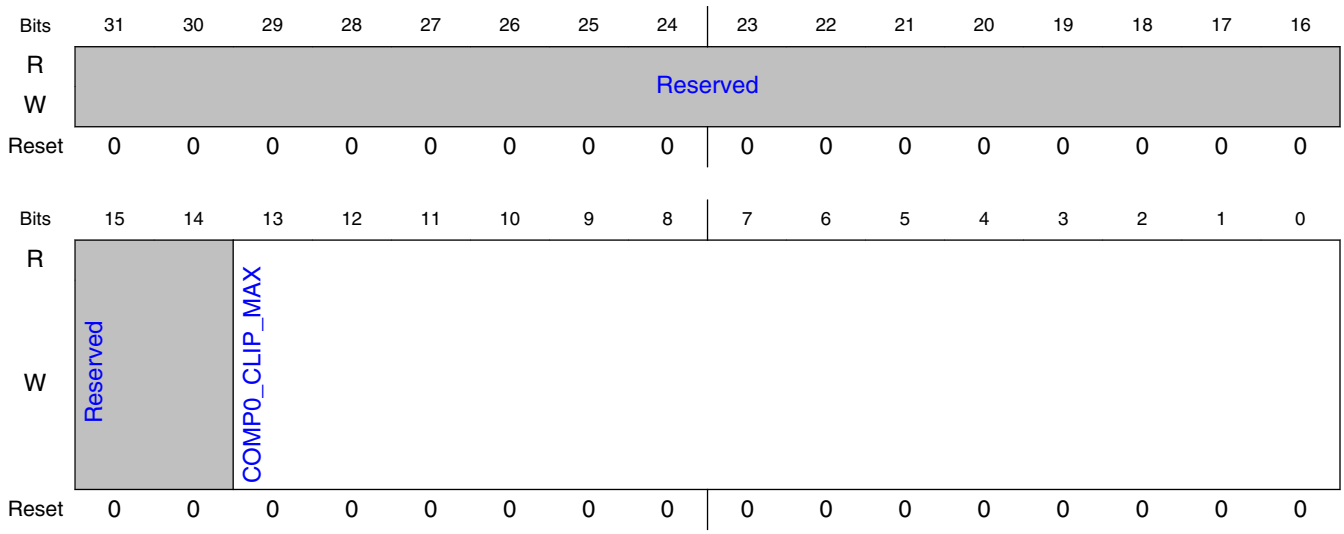
15.10.3.1.117.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MAX_0	7840h

15.10.3.1.117.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0, 16383]. Pipe1 is also called channel1.

15.10.3.1.117.3 Diagram



15.10.3.1.117.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.118 Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE2_CSCB_IO_MAX_1)

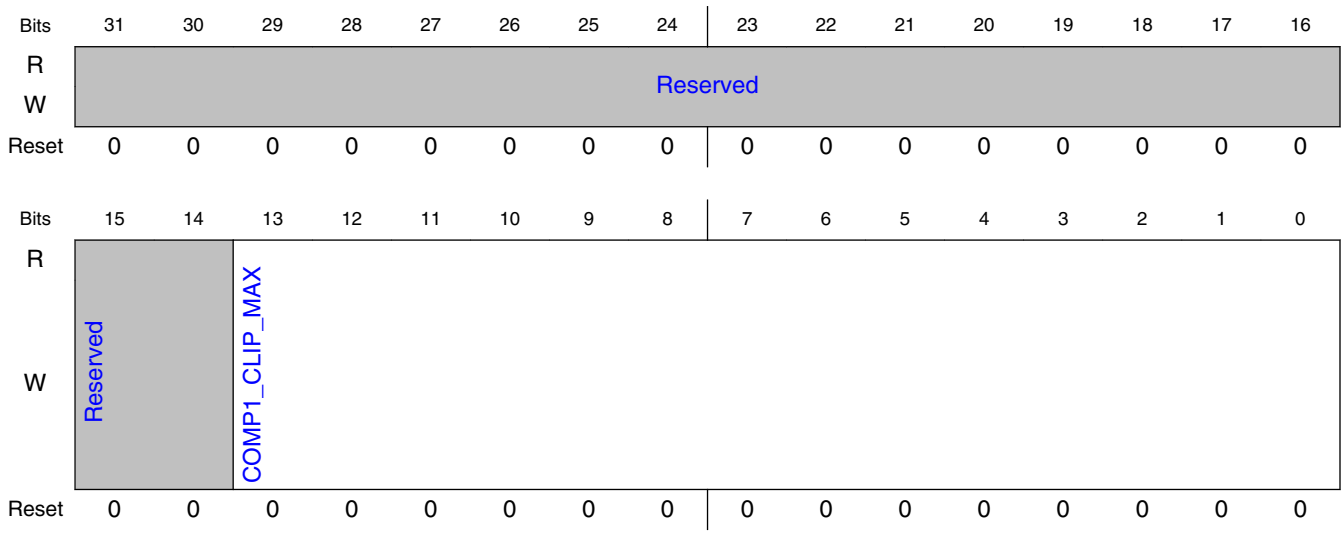
15.10.3.1.118.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MAX_1	7844h

15.10.3.1.118.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.118.3 Diagram



15.10.3.1.118.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.119 Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE2_CSCB_IO_MAX_2)

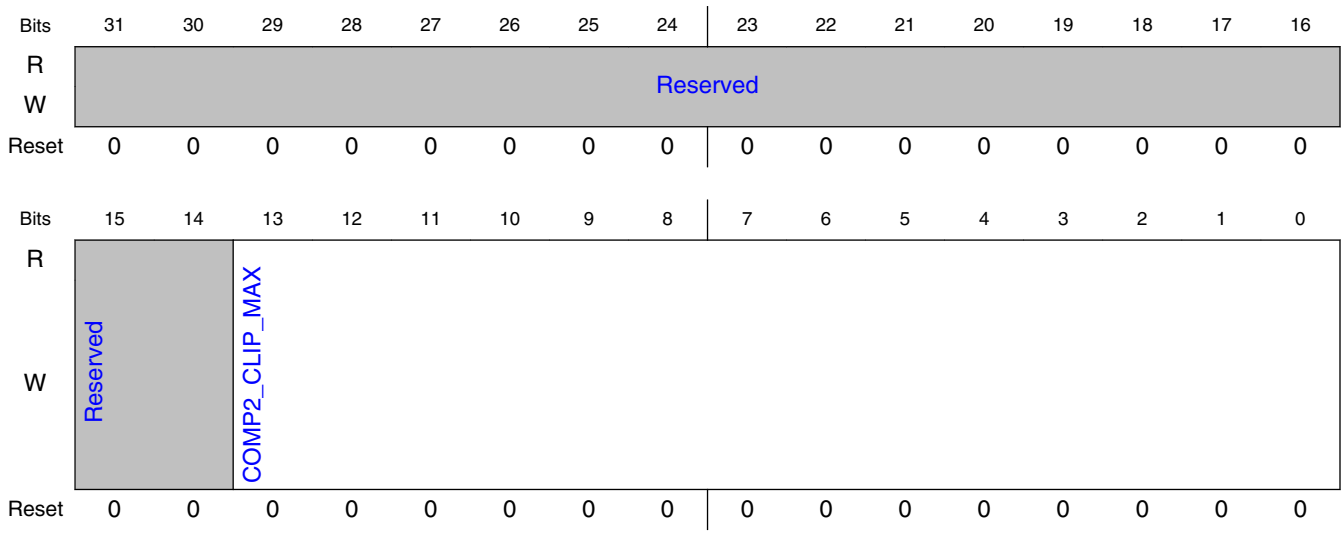
15.10.3.1.119.1 Offset

Register	Offset
HDR_PIPE2_CSCB_IO_MAX_2	7848h

15.10.3.1.119.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.119.3 Diagram



15.10.3.1.119.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.120 Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE2_CSCB_NORM)

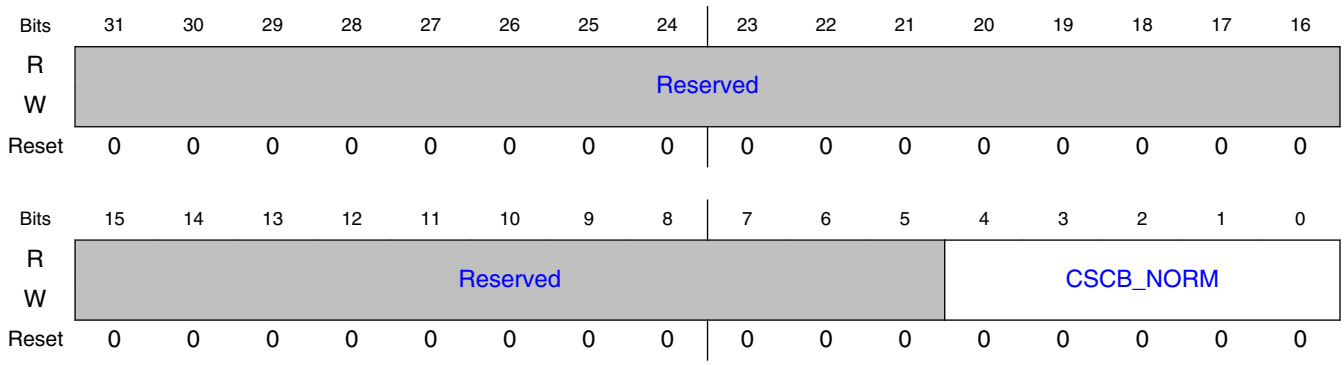
15.10.3.1.120.1 Offset

Register	Offset
HDR_PIPE2_CSCB_NORM	784Ch

15.10.3.1.120.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.120.3 Diagram



15.10.3.1.120.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCB_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.121 Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE2_CSCB_OO_0)

15.10.3.1.121.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OO_0	7850h

15.10.3.1.121.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.121.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_0												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.121.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_0	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.122 Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE2_CSCB_OO_1)

15.10.3.1.122.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OO_1	7854h

15.10.3.1.122.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.122.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_1												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.122.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_1	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.123 Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE2_CSCB_OO_2)

15.10.3.1.123.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OO_2	7858h

15.10.3.1.123.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.123.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_2												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.123.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_2	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.124 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE2_CSCB_OMIN_0)

15.10.3.1.124.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMIN_0	785Ch

15.10.3.1.124.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.124.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.124.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.125 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE2_CSCB_OMIN_1)

15.10.3.1.125.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMIN_1	7860h

15.10.3.1.125.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.125.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.125.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.126 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE2_CSCB_OMIN_2)

15.10.3.1.126.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMIN_2	7864h

15.10.3.1.126.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.126.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.126.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minimum clipped pixel component.

15.10.3.1.127 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE2_CSCB_OMAX_0)

15.10.3.1.127.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMAX_0	7868h

15.10.3.1.127.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.127.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MAX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.127.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximum clipped pixel component.

15.10.3.1.128 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE2_CSCB_OMAX_1)

15.10.3.1.128.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMAX_1	786Ch

15.10.3.1.128.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.128.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.128.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.129 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE2_CSCB_OMAX_2)

15.10.3.1.129.1 Offset

Register	Offset
HDR_PIPE2_CSCB_OMAX_2	7870h

15.10.3.1.129.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.129.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MAX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.129.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.130 Pipe1 floating point to fixed point control (HDR_PIPE2_FL2FX)

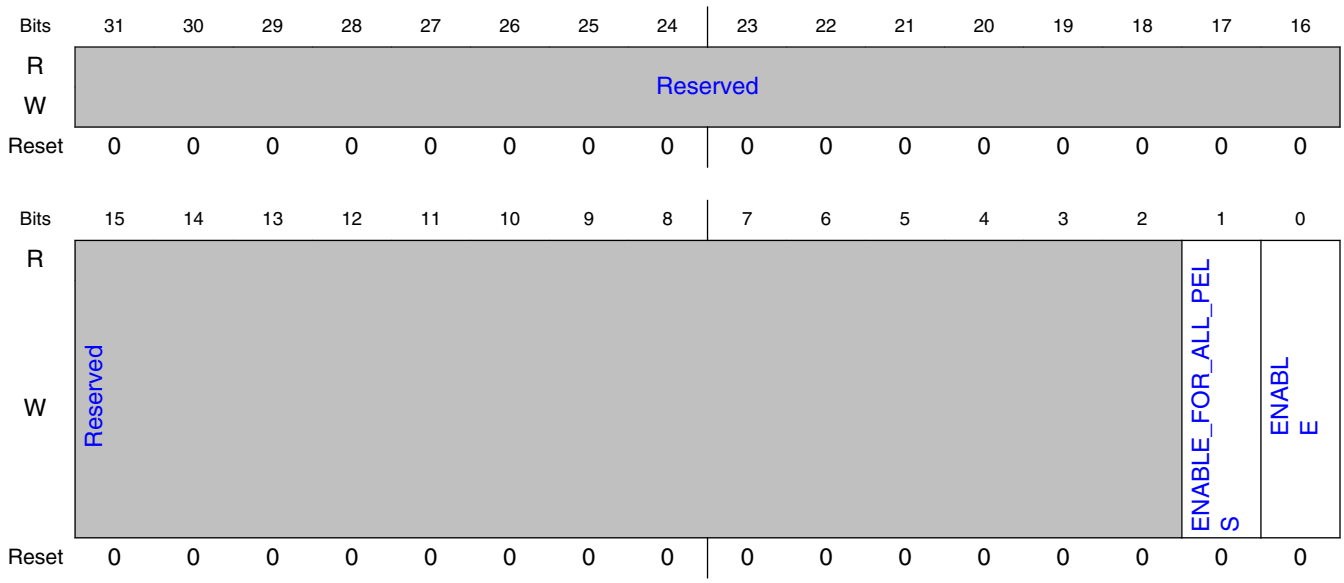
15.10.3.1.130.1 Offset

Register	Offset
HDR_PIPE2_FL2FX	7874h

15.10.3.1.130.2 Function

Pipe1 floating point to fixed point control Pipe1 is also called channel1.

15.10.3.1.130.3 Diagram



15.10.3.1.130.4 Fields

Field	Function
31-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This Float to Fixed operation is enabled only for blended pixels 1: This Float to Fixed operation is enabled all pixels
0 ENABLE	0: Don't enable this Float-to-Fixed converter: Pixels pass thru the Float-to-Fixed unmodiifed 1: This Float-to-Fixed converter is enabled for current picture

15.10.3.1.131 PIPE2: NOT USED (HDR_PIPE2_ENTRY_30)

15.10.3.1.131.1 Offset

Register	Offset
HDR_PIPE2_ENTRY_30	7878h

15.10.3.1.131.2 Function

PIPE2: NOT USED

15.10.3.1.131.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.131.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.132 A0 component Look-Up-Table. (LUT) (PIPE3_A0_LUT)

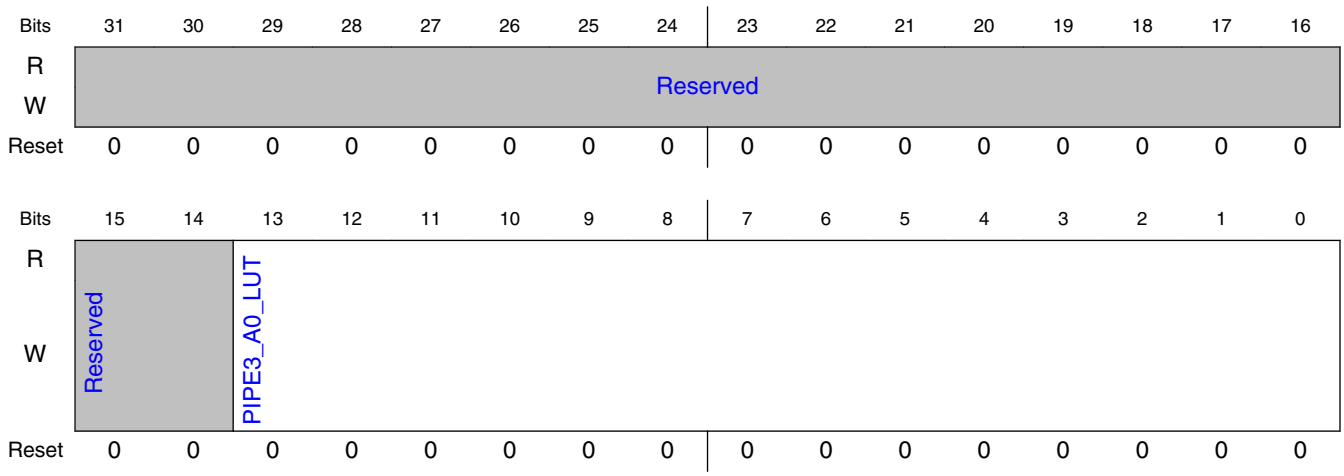
15.10.3.1.132.1 Offset

Register	Offset
PIPE3_A0_LUT	8000h

15.10.3.1.132.2 Function

The LUT table has 1024 entries. A0 component may be R or Y or other

15.10.3.1.132.3 Diagram



15.10.3.1.132.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE3_A0_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.133 A1 component Look-Up-Table. (LUT) (PIPE3_A1_LUT)

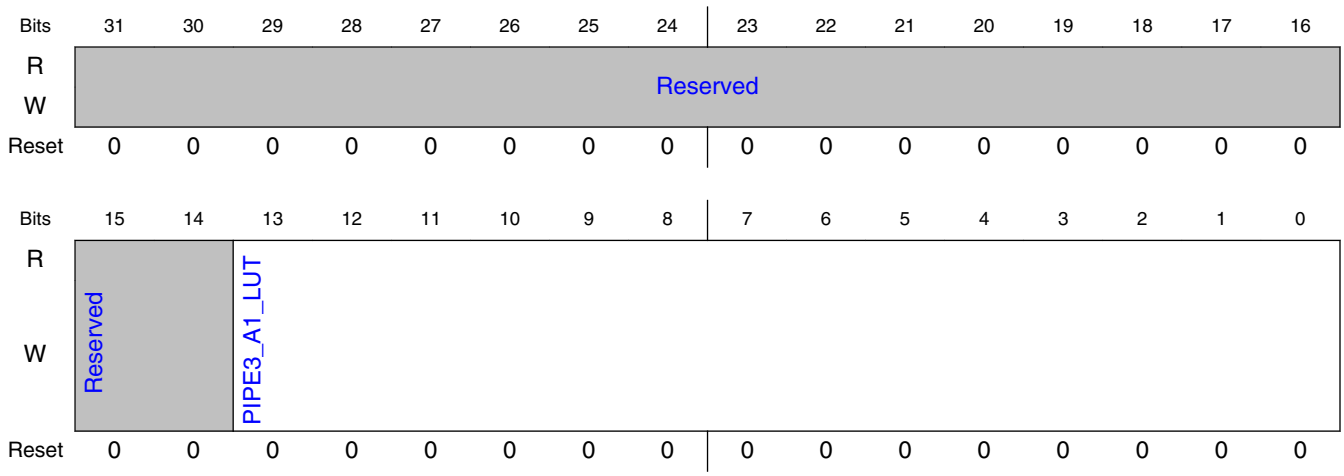
15.10.3.1.133.1 Offset

Register	Offset
PIPE3_A1_LUT	9000h

15.10.3.1.133.2 Function

The LUT table has 1024 entries. A1 component may be G or Cb or other

15.10.3.1.133.3 Diagram



15.10.3.1.133.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE3_A1_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.134 A2 component Look-Up-Table. (LUT) (PIPE3_A2_LUT)

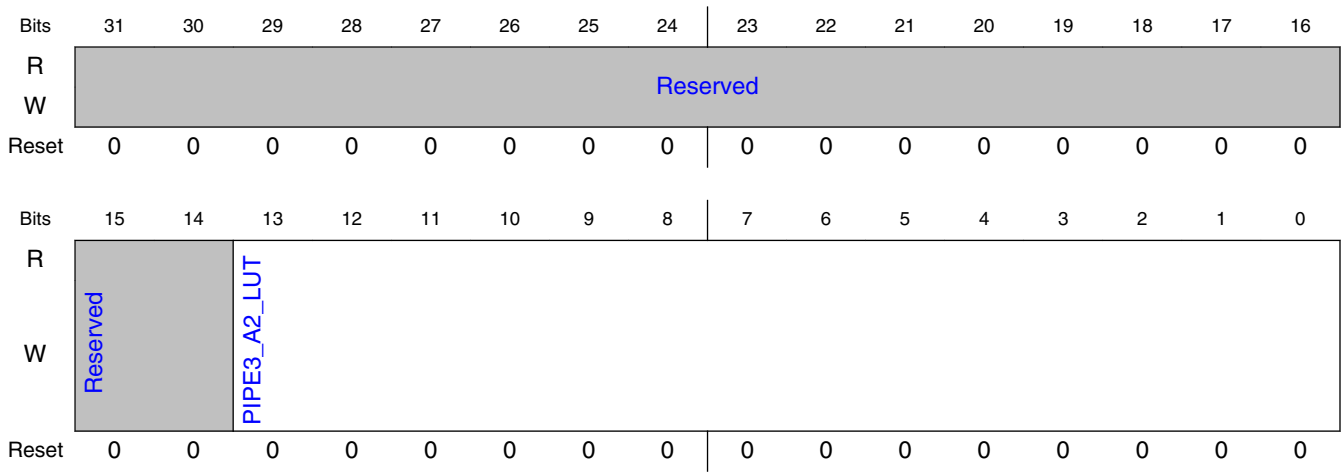
15.10.3.1.134.1 Offset

Register	Offset
PIPE3_A2_LUT	A000h

15.10.3.1.134.2 Function

The LUT table has 1024 entries. A2 component may be B or Cr or other

15.10.3.1.134.3 Diagram



15.10.3.1.134.4 Fields

Field	Function
31-14 —	Reserved.
13-0 PIPE3_A2_LUT	The LUT entries are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.135 Pipe1 Colorspace Converter A control. (HDR_PIPE3_CSCA_CONTROL_REG)

15.10.3.1.135.1 Offset

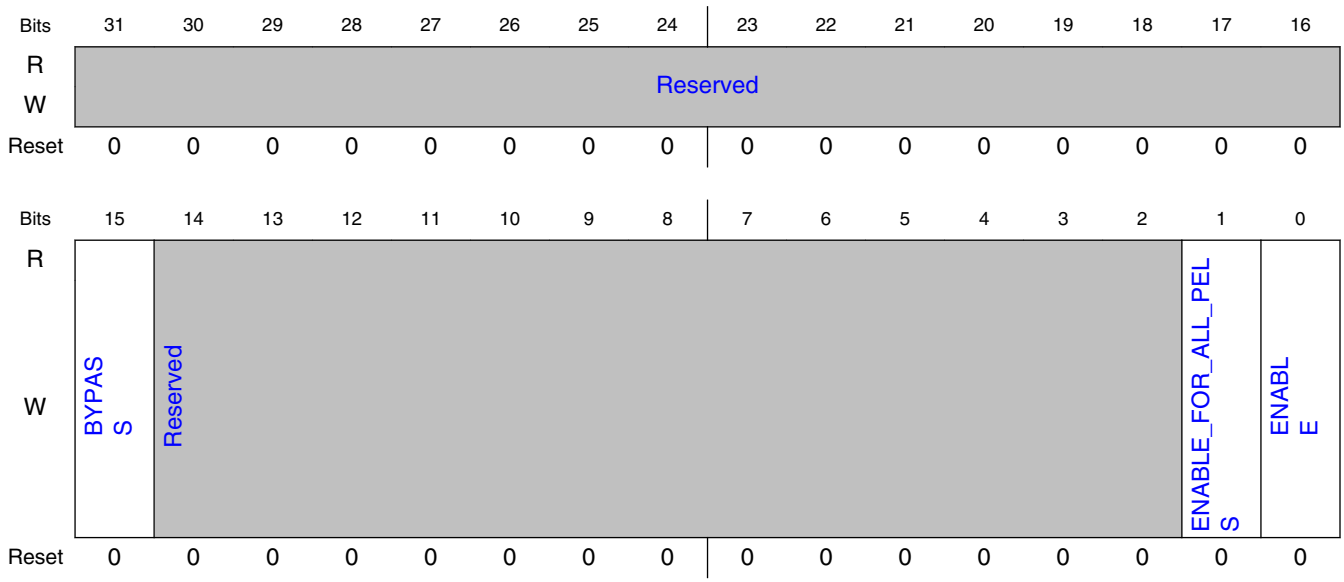
Register	Offset
HDR_PIPE3_CSCA_CONTROL_REG	B000h

15.10.3.1.135.2 Function

Controls the Color-Space-Converter-A (CSCA) in pipe1. This CSCA takes in 10-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of

the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 10 bit per component pixels. After CSCA the pixels are fed into a LUT. Pipe1 is also called channel1.

15.10.3.1.135.3 Diagram



15.10.3.1.135.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.136 Pipe1 Colorspace Converter A (CSCA) h(0,0) matrix coefficient (HDR_PIPE3_CSCA_H00)

15.10.3.1.136.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H00	B004h

15.10.3.1.136.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.136.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.136.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.137 Pipe1 Colorspace Converter A (CSCA) h(1,0) matrix coefficient (HDR_PIPE3_CSCA_H10)

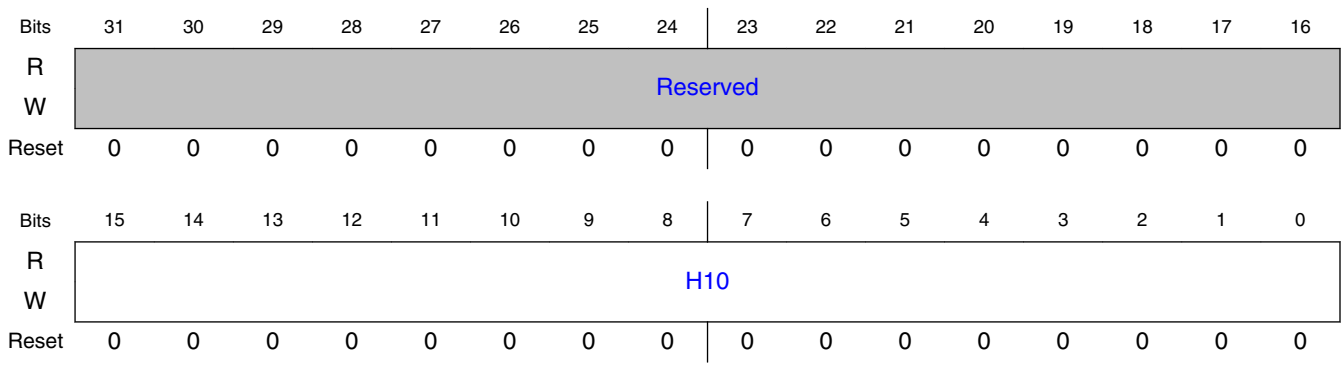
15.10.3.1.137.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H10	B008h

15.10.3.1.137.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.137.3 Diagram



15.10.3.1.137.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.138 Pipe1 Colorspace Converter A (CSCA) h(2,0) matrix coefficient (HDR_PIPE3_CSCA_H20)

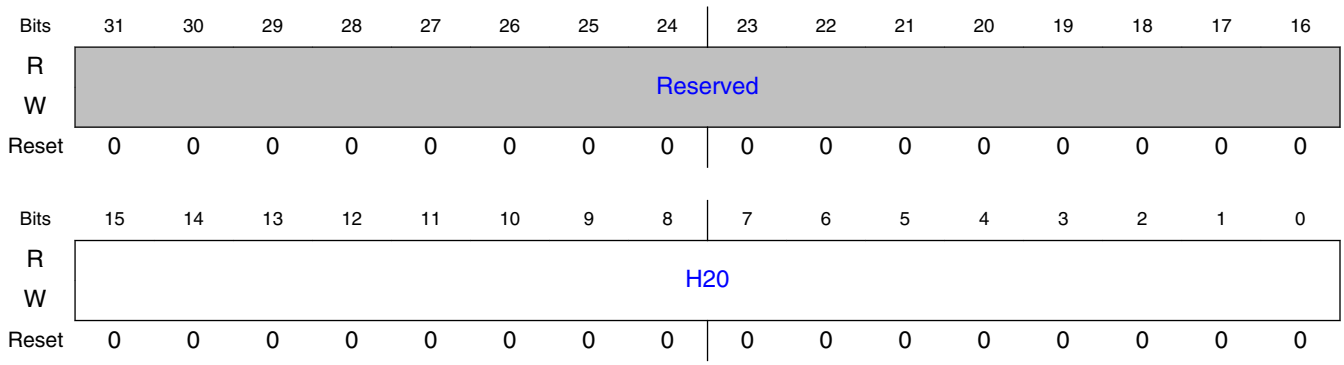
15.10.3.1.138.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H20	B00Ch

15.10.3.1.138.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.138.3 Diagram



15.10.3.1.138.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.139 Pipe1 Colorspace Converter A (CSCA) h(0,1) matrix coefficient (HDR_PIPE3_CSCA_H01)

15.10.3.1.139.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H01	B010h

15.10.3.1.139.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.139.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.139.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.140 Pipe1 Colorspace Converter A (CSCA) h(1,1) matrix coefficient (HDR_PIPE3_CSCA_H11)

15.10.3.1.140.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H11	B014h

15.10.3.1.140.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.140.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.140.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.141 Pipe1 Colorspace Converter A (CSCA) h(2,1) matrix coefficient (HDR_PIPE3_CSCA_H21)

15.10.3.1.141.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H21	B018h

15.10.3.1.141.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.141.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.141.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.142 Pipe1 Colorspace Converter A (CSCA) h(0,2) matrix coefficient (HDR_PIPE3_CSCA_H02)

15.10.3.1.142.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H02	B01Ch

15.10.3.1.142.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.142.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.142.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.143 Pipe1 Colorspace Converter A (CSCA) h(1,2) matrix coefficient (HDR_PIPE3_CSCA_H12)

15.10.3.1.143.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H12	B020h

15.10.3.1.143.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.143.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.143.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.144 Pipe1 Colorspace Converter A (CSCA) h(2,2) matrix coefficient (HDR_PIPE3_CSCA_H22)

15.10.3.1.144.1 Offset

Register	Offset
HDR_PIPE3_CSCA_H22	B024h

15.10.3.1.144.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.144.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.144.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.145 Pipe1 Colorspace Converter A (CSCA) component 0 pre-offset (HDR_PIPE3_CSCA_IO_0)

15.10.3.1.145.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_0	B028h

15.10.3.1.145.2 Function

An signed 10-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.145.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMPO_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.145.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.146 Pipe1 Colorspace Converter A (CSCA) component 1 pre-offset (HDR_PIPE3_CSCA_IO_1)

15.10.3.1.146.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_1	B02Ch

15.10.3.1.146.2 Function

An signed 10-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.146.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.146.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.147 Pipe1 Colorspace Converter A (CSCA) component 2 pre-offset (HDR_PIPE3_CSCA_IO_2)

15.10.3.1.147.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_2	B030h

15.10.3.1.147.2 Function

An signed 10-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.147.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.147.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.148 Pipe1 Colorspace Converter A (CSCA) component 0 clip min. (HDR_PIPE3_CSCA_IO_MIN_0)

15.10.3.1.148.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MIN_0	B034h

15.10.3.1.148.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.148.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP0_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.148.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.149 Pipe1 Colorspace Converter A (CSCA) component 1 clip min. (HDR_PIPE3_CSCA_IO_MIN_1)

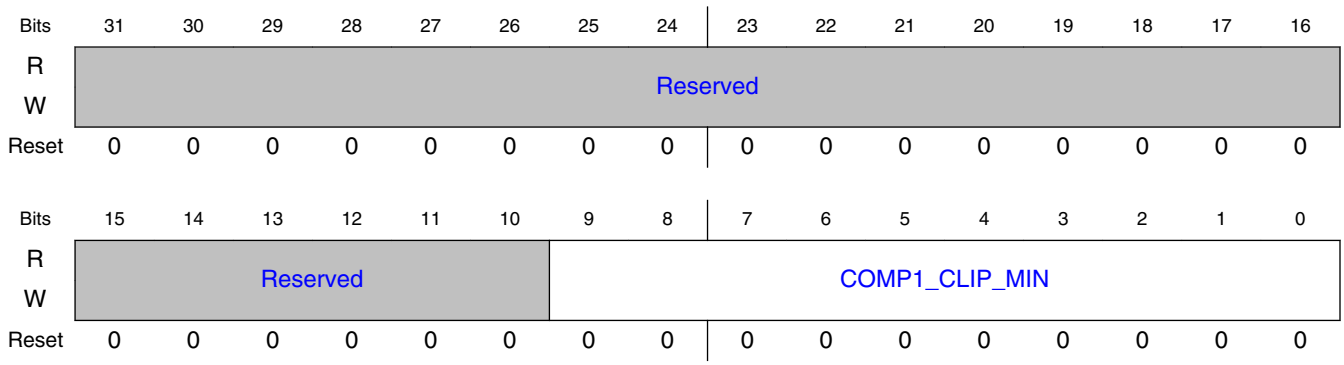
15.10.3.1.149.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MIN_1	B038h

15.10.3.1.149.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.149.3 Diagram



15.10.3.1.149.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.150 Pipe1 Colorspace Converter A (CSCA) component 2 clip min. (HDR_PIPE3_CSCA_IO_MIN_2)

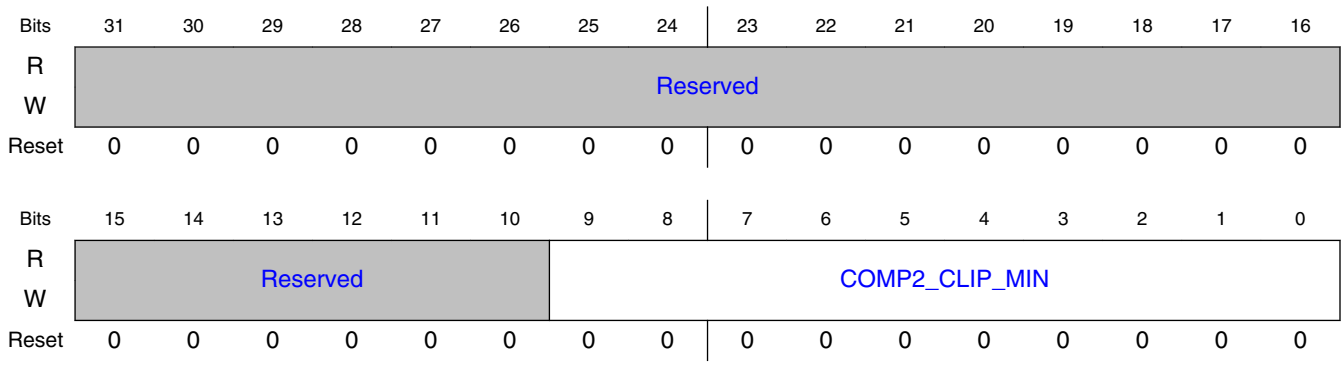
15.10.3.1.150.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MIN_2	B03Ch

15.10.3.1.150.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511]. Pipe1 is also called channel1.

15.10.3.1.150.3 Diagram



15.10.3.1.150.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.151 Pipe1 Colorspace Converter A (CSCA) component 0 clip max value. (HDR_PIPE3_CSCA_IO_MAX_0)

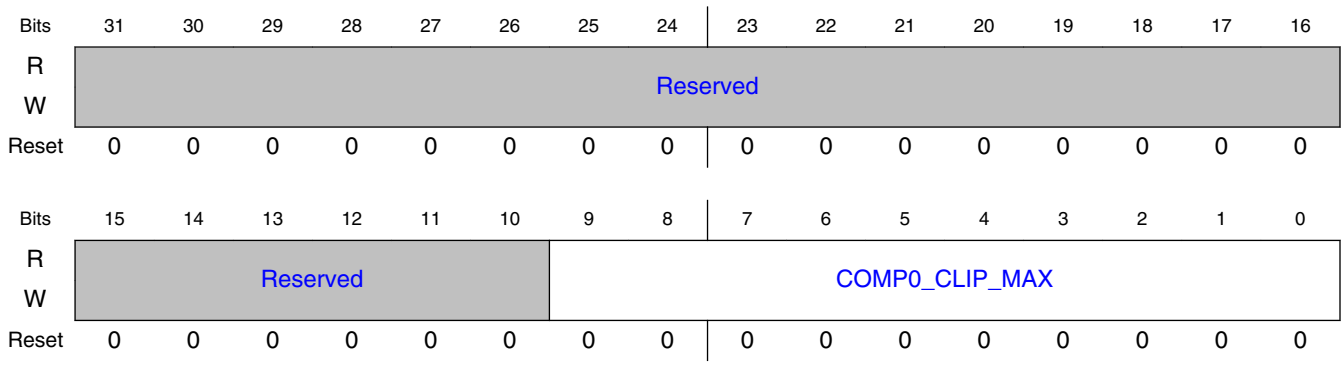
15.10.3.1.151.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MAX_0	B040h

15.10.3.1.151.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.151.3 Diagram



15.10.3.1.151.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.152 Pipe1 Colorspace Converter A (CSCA) component 1 clip max value. (HDR_PIPE3_CSCA_IO_MAX_1)

15.10.3.1.152.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MAX_1	B044h

15.10.3.1.152.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.152.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_CLIP_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.152.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.153 Pipe1 Colorspace Converter A (CSCA) component 2 clip max value. (HDR_PIPE3_CSCA_IO_MAX_2)

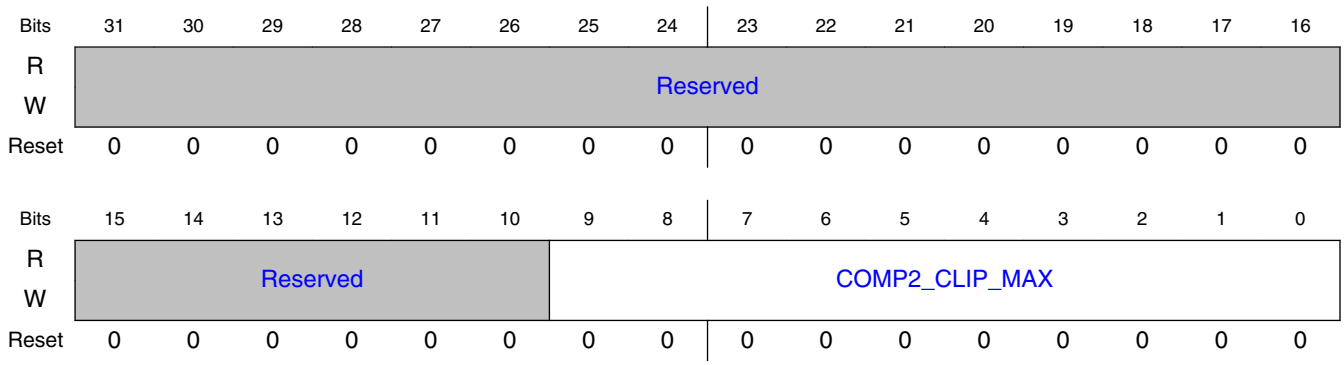
15.10.3.1.153.1 Offset

Register	Offset
HDR_PIPE3_CSCA_IO_MAX_2	B048h

15.10.3.1.153.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023]. Pipe1 is also called channel1.

15.10.3.1.153.3 Diagram



15.10.3.1.153.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.154 Pipe1 Colorspace Converter A (CSCA) normalization factor (HDR_PIPE3_CSCA_NORM)

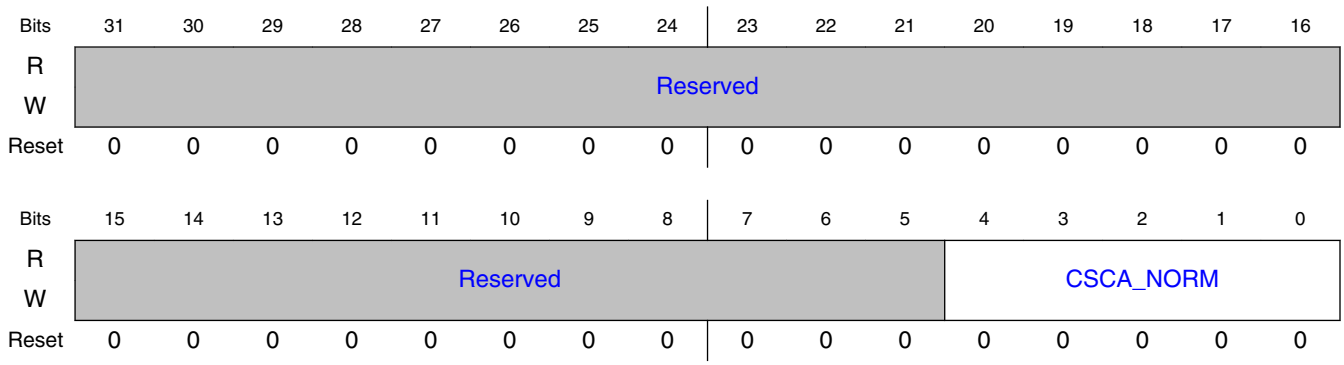
15.10.3.1.154.1 Offset

Register	Offset
HDR_PIPE3_CSCA_NORM	B04Ch

15.10.3.1.154.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.154.3 Diagram



15.10.3.1.154.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCA_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.155 Pipe1 Colorspace Converter A (CSCA): Post offset component 0 (HDR_PIPE3_CSCA_OO_0)

15.10.3.1.155.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OO_0	B050h

15.10.3.1.155.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.155.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_0											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.155.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_0	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.156 Pipe1 Colorspace Converter A (CSCA): Post offset component 1 (HDR_PIPE3_CSCA_OO_1)

15.10.3.1.156.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OO_1	B054h

15.10.3.1.156.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.156.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.156.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_1	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.157 Pipe1 Colorspace Converter A (CSCA): Post offset component 2 (HDR_PIPE3_CSCA_OO_2)

15.10.3.1.157.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OO_2	B058h

15.10.3.1.157.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.157.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSCA_OO_2											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCA_OO_2															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.157.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSCA_OO_2	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.158 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 0 (HDR_PIPE3_CSCA_OMIN_0)

15.10.3.1.158.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMIN_0	B05Ch

15.10.3.1.158.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.158.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.158.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.159 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 1 (HDR_PIPE3_CSCA_OMIN_1)

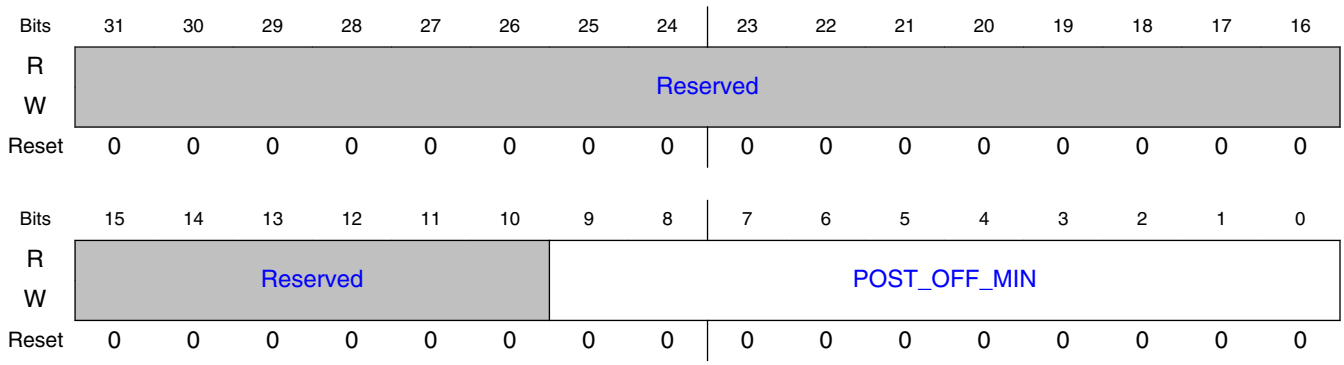
15.10.3.1.159.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMIN_1	B060h

15.10.3.1.159.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.159.3 Diagram



15.10.3.1.159.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.160 Pipe1 Colorspace Converter A (CSCA): Post offset min clip value for component 2 (HDR_PIPE3_CSCA_OMIN_2)

15.10.3.1.160.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMIN_2	B064h

15.10.3.1.160.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.160.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.160.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.161 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 0 (HDR_PIPE3_CSCA_OMAX_0)

15.10.3.1.161.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMAX_0	B068h

15.10.3.1.161.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.161.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.161.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.162 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 1 (HDR_PIPE3_CSCA_OMAX_1)

15.10.3.1.162.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMAX_1	B06Ch

15.10.3.1.162.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.162.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.162.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.163 Pipe1 Colorspace Converter A (CSCA): Post offset max clip value for component 2 (HDR_PIPE3_CSCA_OMAX_2)

15.10.3.1.163.1 Offset

Register	Offset
HDR_PIPE3_CSCA_OMAX_2	B070h

15.10.3.1.163.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.163.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.163.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.164 PIPE3: NOT USED (HDR_PIPE3_ENTRY_29)

15.10.3.1.164.1 Offset

Register	Offset
HDR_PIPE3_ENTRY_29	B074h

15.10.3.1.164.2 Function

PIPE3: NOT USED

15.10.3.1.164.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.164.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.165 Pipe1 LUT control register (HDR_PIPE3_LUT_CONTROL_REG)

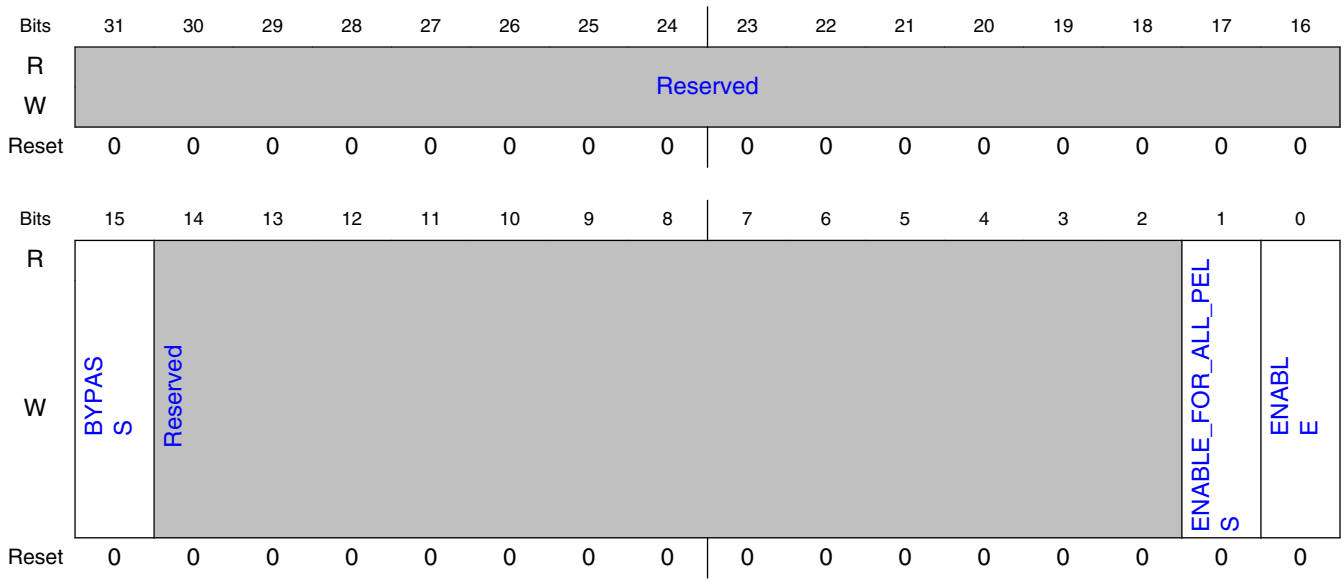
15.10.3.1.165.1 Offset

Register	Offset
HDR_PIPE3_LUT_CONTROL_REG	B080h

15.10.3.1.165.2 Function

Controls the Look-Up-Table in pipe1. Control of LUT is PER PIXEL not per component. LUT has 1024 14-bit, per component, entries. LUT maybe used for conversion from non-linear (gamma corrected) to linear pixels. LUT maybe used for conversion from linear to non linear pixels (gamma corrected). LUT can contain 14-bit fixed point or 14-bit floating point pixel value. Floating point format has 9 bit unsigned mantissa with hidden bit and 5 bit, unsigned, biased exponent. {exp[4:0], mantissa[8:0]} => 1.mantissa[8:0] * 2^(exp-1) When using floating piont numbers, in this LUT, these are used in HDR OUTPUT PIPE for HDR non-linear operation and must bypass Color Space Converter B (CSCB). Pipe1 is also called channel1.

15.10.3.1.165.3 Diagram



15.10.3.1.165.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this LUT 1: Pixels pass thru this LUT unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This LUT is enabled only for blened pixels 1: This LUT is enabled all pixels
0 ENABLE	0: Don't enable this LUT: Pixels pass thru the LUT unmodiifed 1: This LUT is enabled for current picture

15.10.3.1.166 Pipe1 Colorspace Converter B control. (HDR_PIPE3_CSCB_CONTROL_REG)

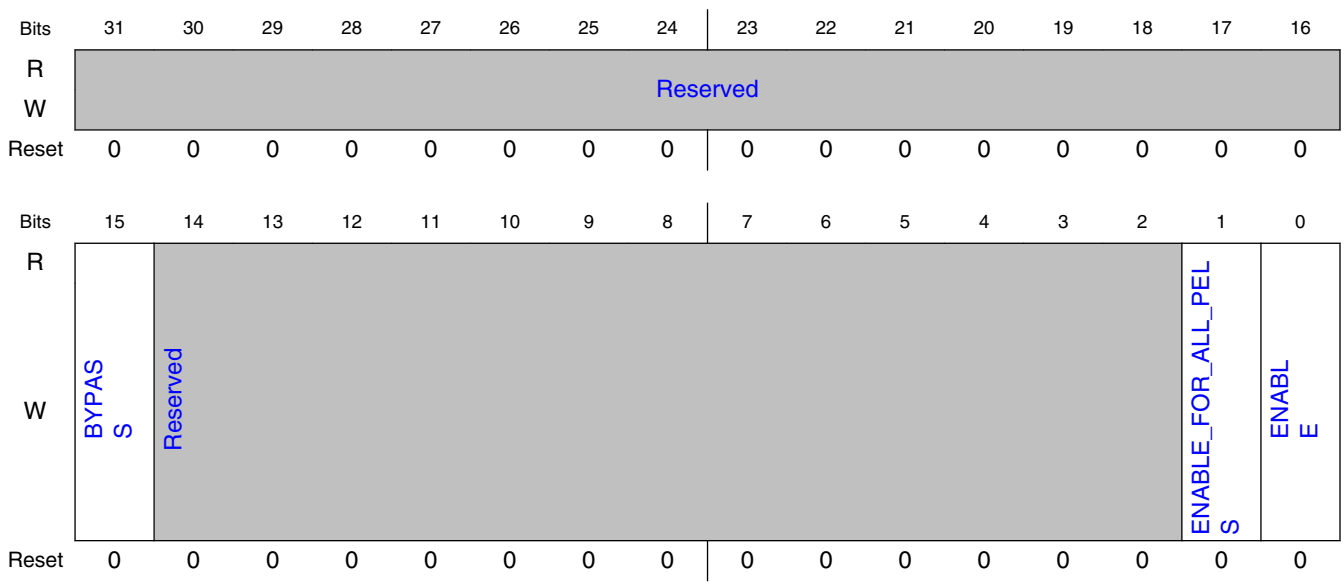
15.10.3.1.166.1 Offset

Register	Offset
HDR_PIPE3_CSCB_CONTROL_REG	B800h

15.10.3.1.166.2 Function

Controls the Color-Space-Converter-B (CSCB) in pipe1. This CSCB takes in 14-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCA are 28 bit per component pixels. After CSCB the pixels are fed into the Blender. Pipe1 is also called channel1.

15.10.3.1.166.3 Diagram



15.10.3.1.166.4 Fields

Field	Function
31-16 —	Reserved.
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.

Table continues on the next page...

Field	Function
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.167 Pipe1 Colorspace Converter A (CSCB) h(0,0) matrix coefficient (HDR_PIPE3_CSCB_H00)

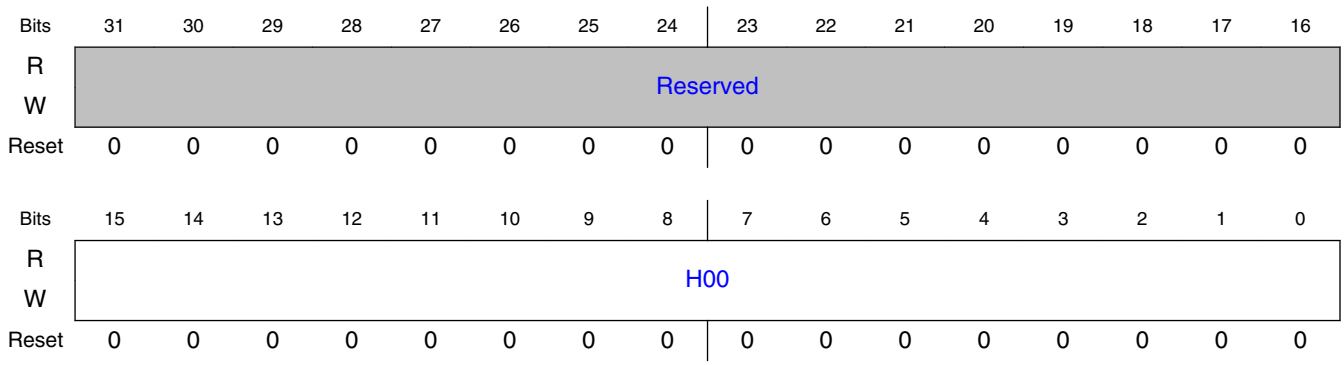
15.10.3.1.167.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H00	B804h

15.10.3.1.167.2 Function

h(0,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.167.3 Diagram



15.10.3.1.167.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.168 Pipe1 Colorspace Converter B (CSCB) h(1,0) matrix coefficient (HDR_PIPE3_CSCB_H10)

15.10.3.1.168.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H10	B808h

15.10.3.1.168.2 Function

h(1,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.168.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.168.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.169 Pipe1 Colorspace Converter B (CSCB) h(2,0) matrix coefficient (HDR_PIPE3_CSCB_H20)

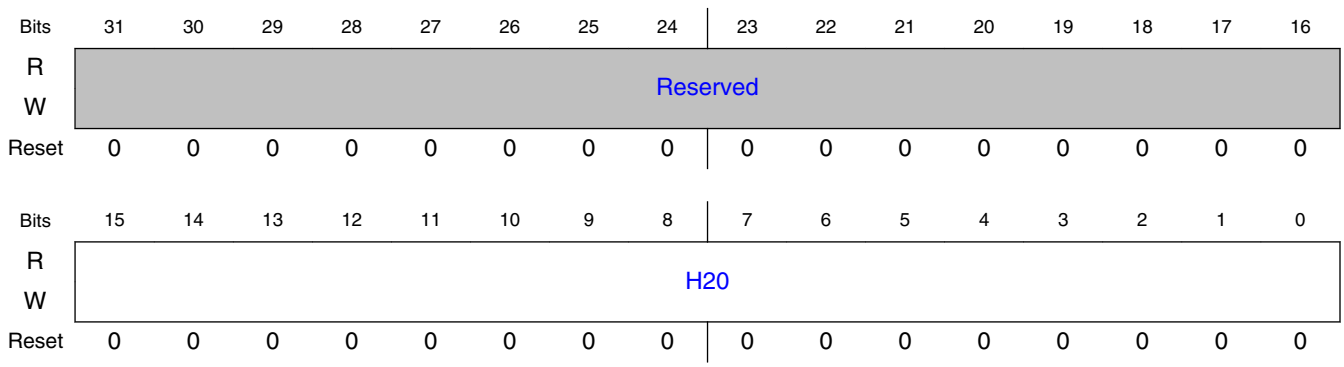
15.10.3.1.169.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H20	B80Ch

15.10.3.1.169.2 Function

h(2,0) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.169.3 Diagram



15.10.3.1.169.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.170 Pipe1 Colorspace Converter B (CSCB) h(0,1) matrix coefficient (HDR_PIPE3_CSCB_H01)

15.10.3.1.170.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H01	B810h

15.10.3.1.170.2 Function

h(0,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.170.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.170.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.171 Pipe1 Colorspace Converter B (CSCB) h(1,1) matrix coefficient (HDR_PIPE3_CSCB_H11)

15.10.3.1.171.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H11	B814h

15.10.3.1.171.2 Function

h(1,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.171.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.171.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.172 Pipe1 Colorspace Converter B (CSCB) h(2,1) matrix coefficient (HDR_PIPE3_CSCB_H21)

15.10.3.1.172.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H21	B818h

15.10.3.1.172.2 Function

h(2,1) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.172.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.172.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.173 Pipe1 Colorspace Converter B (CSCB) h(0,2) matrix coefficient (HDR_PIPE3_CSCB_H02)

15.10.3.1.173.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H02	B81Ch

15.10.3.1.173.2 Function

h(0,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.173.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.173.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.174 Pipe1 Colorspace Converter B (CSCB) h(1,2) matrix coefficient (HDR_PIPE3_CSCB_H12)

15.10.3.1.174.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H12	B820h

15.10.3.1.174.2 Function

h(1,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.174.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.174.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.175 Pipe1 Colorspace Converter B (CSCB) h(2,2) matrix coefficient (HDR_PIPE3_CSCB_H22)

15.10.3.1.175.1 Offset

Register	Offset
HDR_PIPE3_CSCB_H22	B824h

15.10.3.1.175.2 Function

h(2,2) coefficient in color space matrix multiply Pipe1 is also called channel1.

15.10.3.1.175.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.175.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.176 Pipe1 Colorspace Converter B (CSCB) component 0 pre-offset (HDR_PIPE3_CSCB_IO_0)

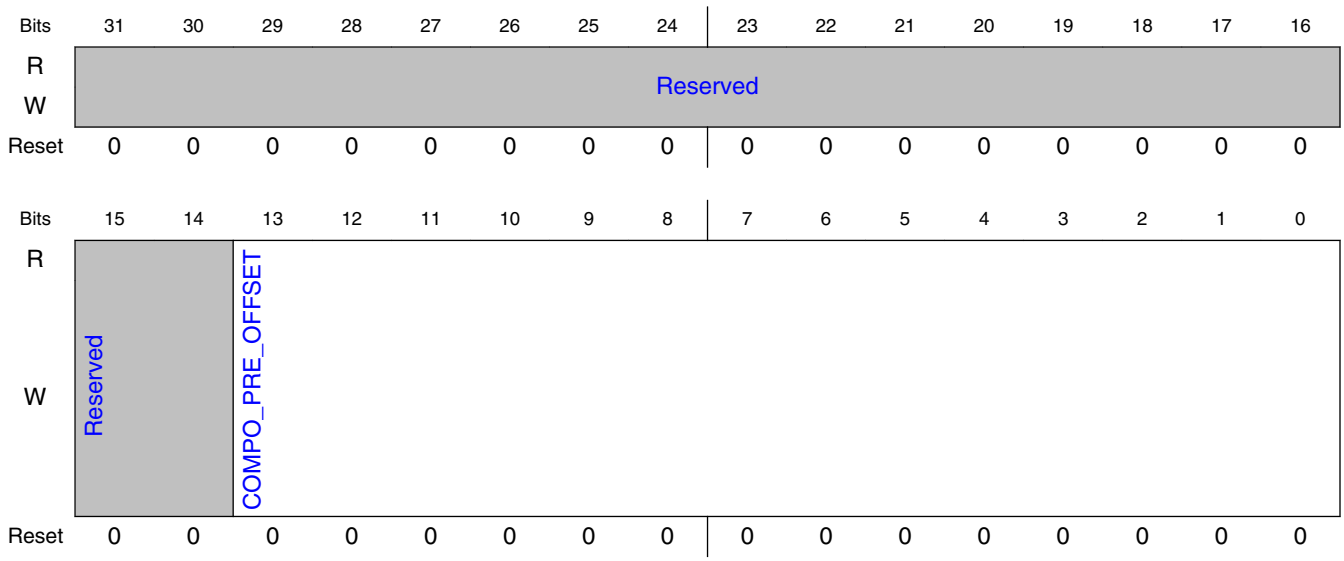
15.10.3.1.176.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_0	B828h

15.10.3.1.176.2 Function

A signed 14-bit offset is added to component 0 (R,Y) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.176.3 Diagram



15.10.3.1.176.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.177 Pipe1 Colorspace Converter B (CSCB) component 1 pre-offset (HDR_PIPE3_CSCB_IO_1)

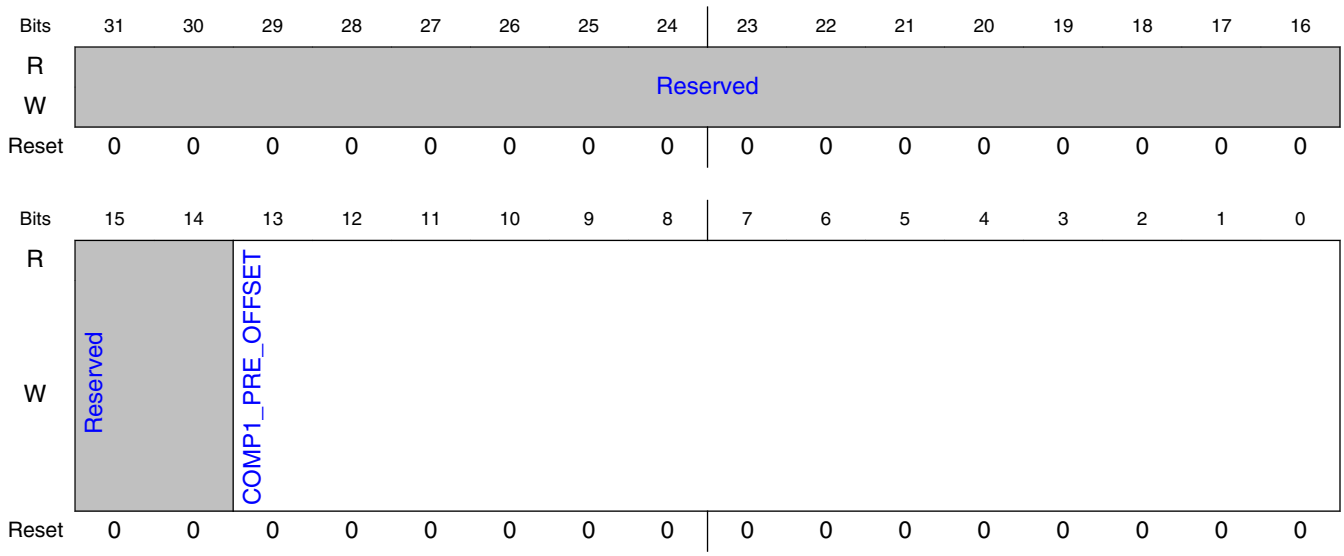
15.10.3.1.177.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_1	B82Ch

15.10.3.1.177.2 Function

A signed 14-bit offset is added to component 1 (G,Cb) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.177.3 Diagram



15.10.3.1.177.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.178 Pipe1 Colorspace Converter B (CSCB) component 2 pre-offset (HDR_PIPE3_CSCB_IO_2)

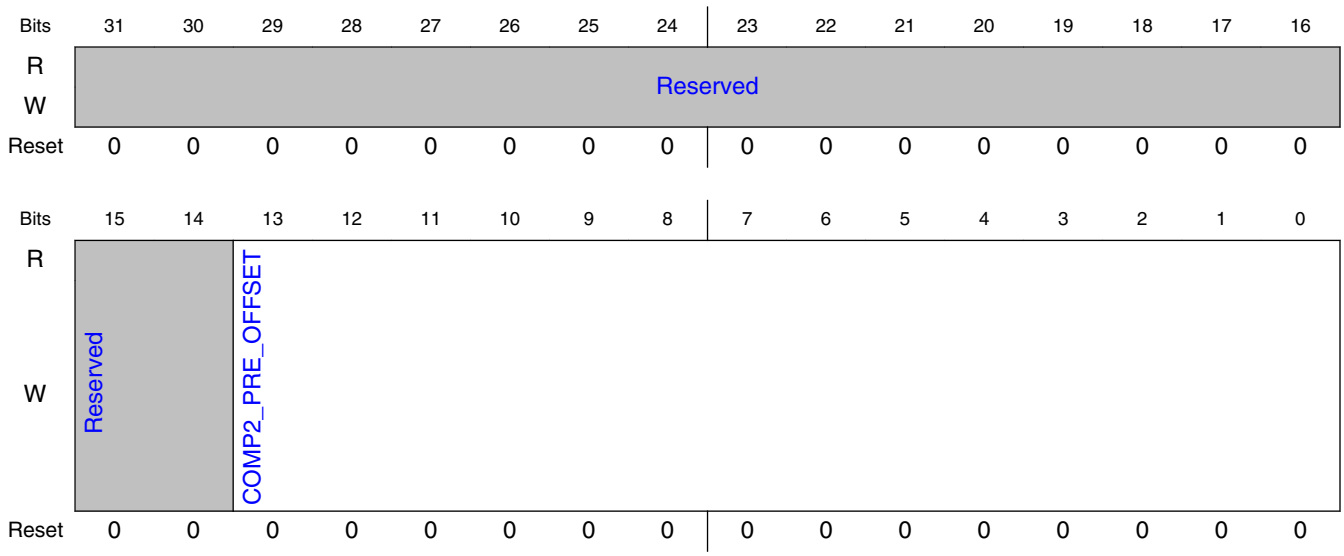
15.10.3.1.178.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_2	B830h

15.10.3.1.178.2 Function

A signed 14-bit offset is added to component 2 (B,Cr) before matrix multiply Pipe1 is also called channel1.

15.10.3.1.178.3 Diagram



15.10.3.1.178.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.179 Pipe1 Colorspace Converter B (CSCB) component 0 clip min. (HDR_PIPE3_CSCB_IO_MIN_0)

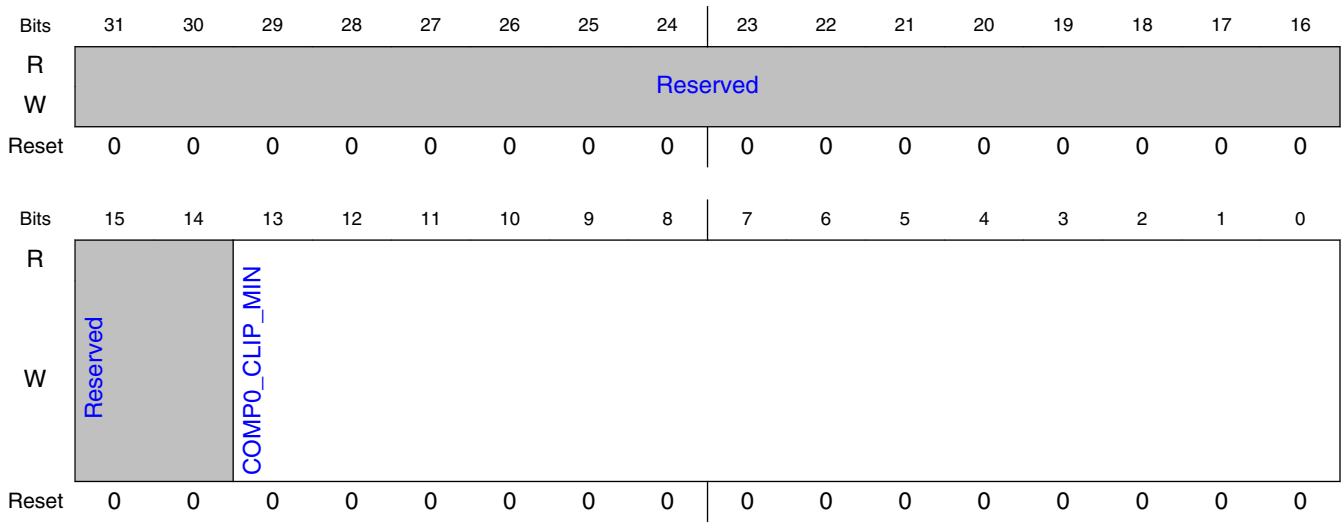
15.10.3.1.179.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MIN_0	B834h

15.10.3.1.179.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.179.3 Diagram



15.10.3.1.179.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.180 Pipe1 Colorspace Converter B (CSCB) component 1 clip min. (HDR_PIPE3_CSCB_IO_MIN_1)

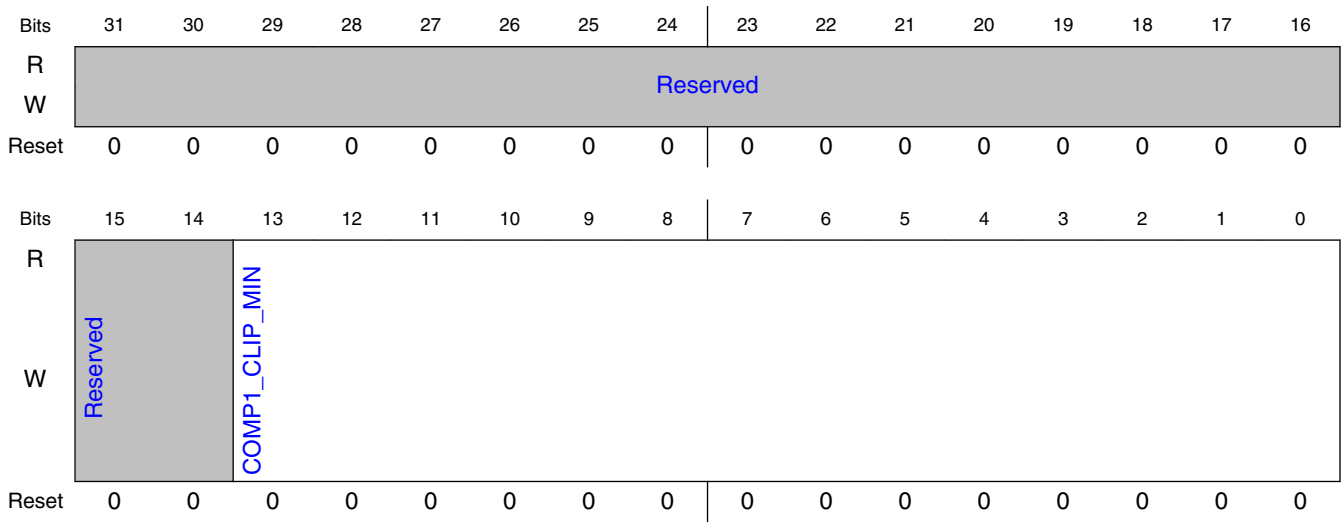
15.10.3.1.180.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MIN_1	B838h

15.10.3.1.180.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 8191]. Pipe1 is also called channel1.

15.10.3.1.180.3 Diagram



15.10.3.1.180.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MIN	This 14-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.181 Pipe1 Colorspace Converter B (CSCB) component 2 clip min. (HDR_PIPE3_CSCB_IO_MIN_2)

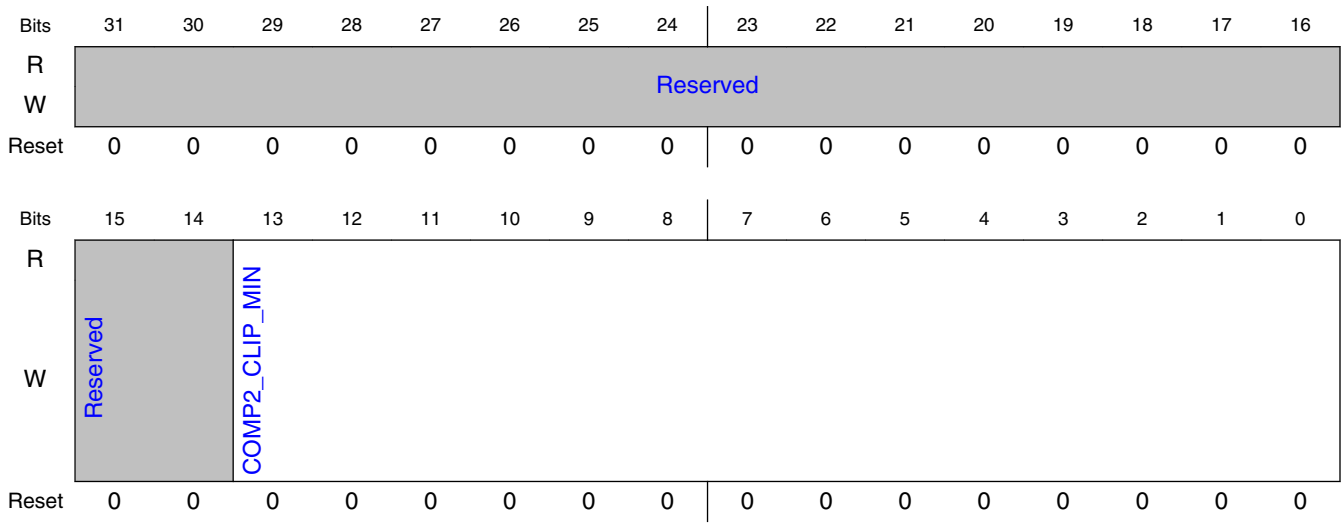
15.10.3.1.181.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MIN_2	B83Ch

15.10.3.1.181.2 Function

After the pre-increment, The result is clipped. This 14-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-8192 , 1891]. Pipe1 is also called channel1.

15.10.3.1.181.3 Diagram



15.10.3.1.181.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.182 Pipe1 Colorspace Converter B (CSCB) component 0 clip max value. (HDR_PIPE3_CSCB_IO_MAX_0)

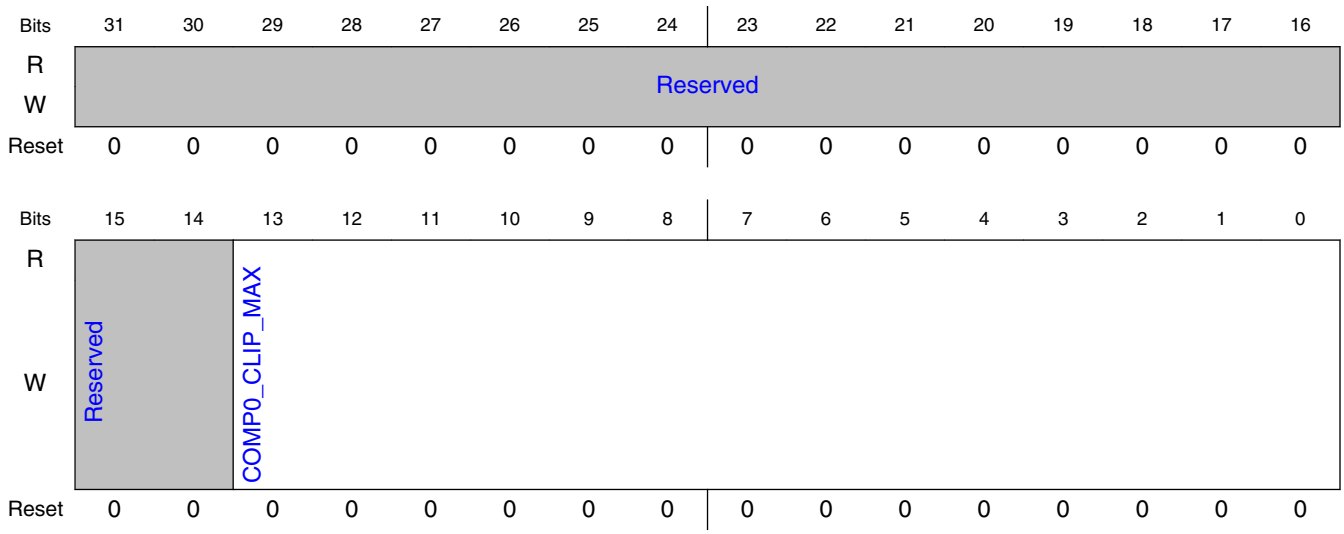
15.10.3.1.182.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MAX_0	B840h

15.10.3.1.182.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0, 16383]. Pipe1 is also called channel1.

15.10.3.1.182.3 Diagram



15.10.3.1.182.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP0_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.183 Pipe1 Colorspace Converter B (CSCB) component 1 clip max value. (HDR_PIPE3_CSCB_IO_MAX_1)

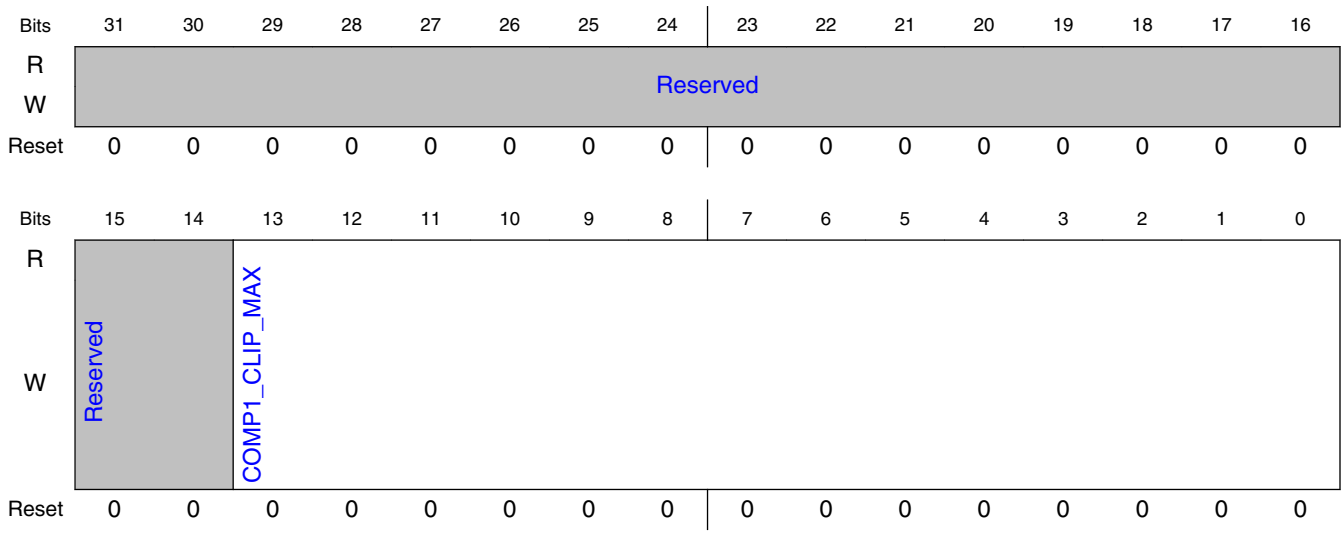
15.10.3.1.183.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MAX_1	B844h

15.10.3.1.183.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.183.3 Diagram



15.10.3.1.183.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP1_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.184 Pipe1 Colorspace Converter B (CSCB) component 2 clip max value. (HDR_PIPE3_CSCB_IO_MAX_2)

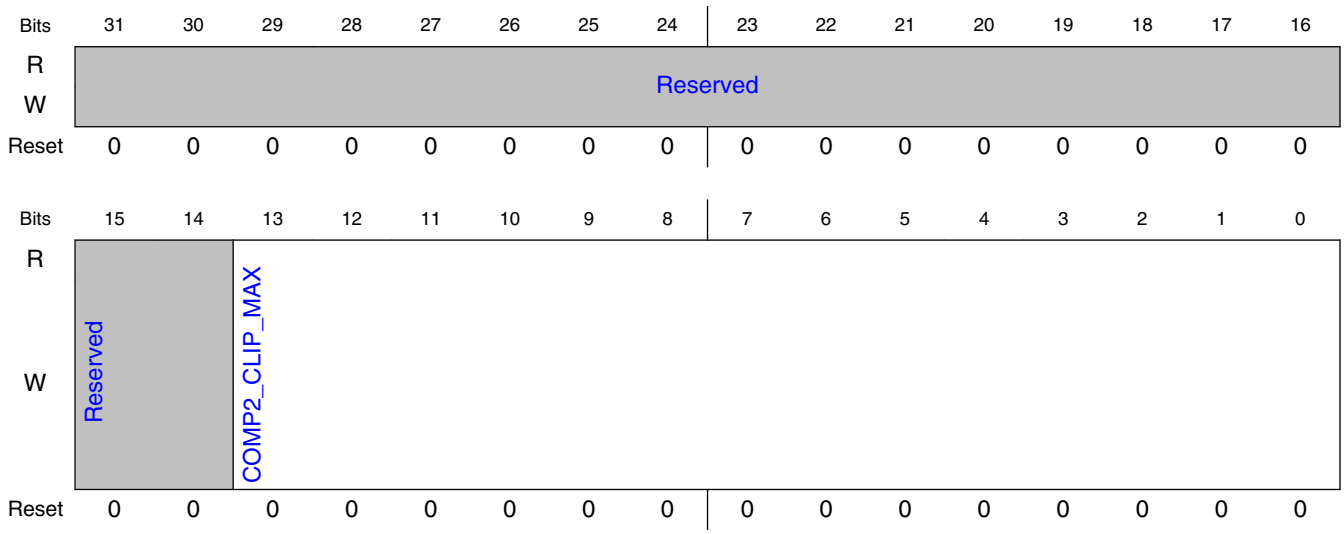
15.10.3.1.184.1 Offset

Register	Offset
HDR_PIPE3_CSCB_IO_MAX_2	B848h

15.10.3.1.184.2 Function

After the pre-increment, The result is clipped. This 14-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,16383]. Pipe1 is also called channel1.

15.10.3.1.184.3 Diagram



15.10.3.1.184.4 Fields

Field	Function
31-14 —	Reserved.
13-0 COMP2_CLIP_MAX	This 14-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.185 Pipe1 Colorspace Converter B (CSCB) normalization factor (HDR_PIPE3_CSCB_NORM)

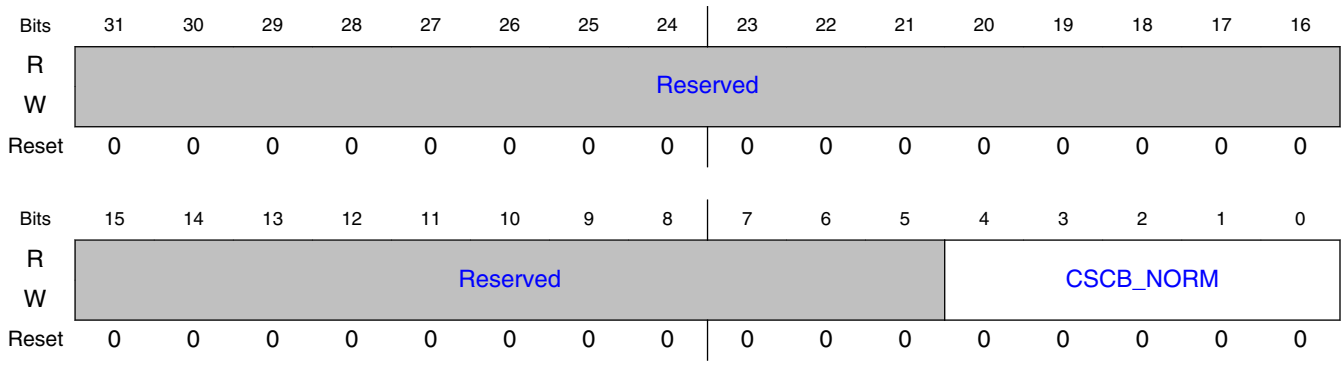
15.10.3.1.185.1 Offset

Register	Offset
HDR_PIPE3_CSCB_NORM	B84Ch

15.10.3.1.185.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift. Pipe1 is also called channel1.

15.10.3.1.185.3 Diagram



15.10.3.1.185.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSCB_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.186 Pipe1 Colorspace Converter B (CSCB): Post offset component 0 (HDR_PIPE3_CSCB_OO_0)

15.10.3.1.186.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OO_0	B850h

15.10.3.1.186.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.186.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_0												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_0															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.186.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_0	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.187 Pipe1 Colorspace Converter B (CSCB): Post offset component 1 (HDR_PIPE3_CSCB_OO_1)

15.10.3.1.187.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OO_1	B854h

15.10.3.1.187.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number. Pipe1 is also called channel1.

15.10.3.1.187.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			CSCB_OO_1												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSCB_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.187.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_1	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.188 Pipe1 Colorspace Converter B (CSCB): Post offset component 2 (HDR_PIPE3_CSCB_OO_2)

15.10.3.1.188.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OO_2	B858h

15.10.3.1.188.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 29-bit number. Pipe1 is also called channel1.

15.10.3.1.188.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			CSCB_OO_2												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSCB_OO_2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.188.4 Fields

Field	Function
31-29 —	Reserved.
28-0 CSCB_OO_2	Ouput Offset (OO) This is a signed 29-bit number. Per component

15.10.3.1.189 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 0 (HDR_PIPE3_CSCB_OMIN_0)

15.10.3.1.189.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMIN_0	B85Ch

15.10.3.1.189.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.189.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.189.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.190 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 1 (HDR_PIPE3_CSCB_OMIN_1)

15.10.3.1.190.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMIN_1	B860h

15.10.3.1.190.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.190.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.190.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minum clipped pixel component.

15.10.3.1.191 Pipe1 Colorspace Converter B (CSCB): Post offset min clip value for component 2 (HDR_PIPE3_CSCB_OMIN_2)

15.10.3.1.191.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMIN_2	B864h

15.10.3.1.191.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.191.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MIN											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MIN															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.191.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MIN	Minimum clipped pixel component.

15.10.3.1.192 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 0 (HDR_PIPE3_CSCB_OMAX_0)

15.10.3.1.192.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMAX_0	B868h

15.10.3.1.192.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.192.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MAX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.192.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximum clipped pixel component.

15.10.3.1.193 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 1 (HDR_PIPE3_CSCB_OMAX_1)

15.10.3.1.193.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMAX_1	B86Ch

15.10.3.1.193.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number. Pipe1 is also called channel1.

15.10.3.1.193.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.193.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.194 Pipe1 Colorspace Converter B (CSCB): Post offset max clip value for component 2 (HDR_PIPE3_CSCB_OMAX_2)

15.10.3.1.194.1 Offset

Register	Offset
HDR_PIPE3_CSCB_OMAX_2	B870h

15.10.3.1.194.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 28-bit number. Pipe1 is also called channel1.

15.10.3.1.194.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				POST_OFF_MAX											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	POST_OFF_MAX															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.194.4 Fields

Field	Function
31-28 —	Reserved.
27-0 POST_OFF_MAX	Maximun clipped pixel component.

15.10.3.1.195 Pipe1 floating point to fixed point control (HDR_PIPE3_FL2FX)

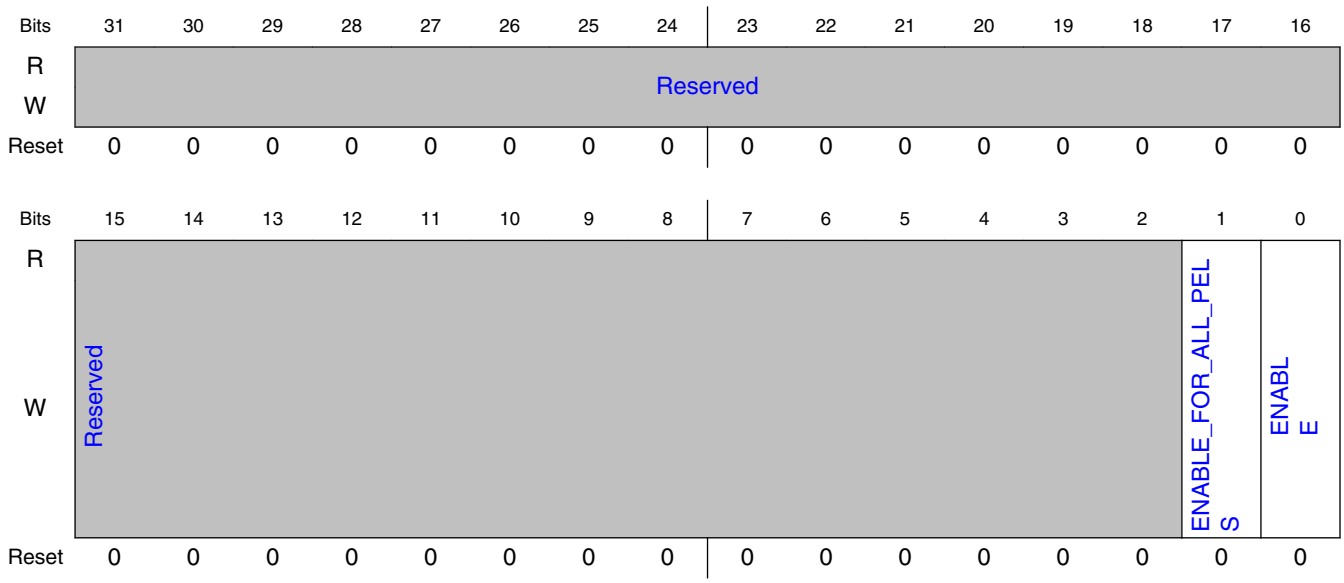
15.10.3.1.195.1 Offset

Register	Offset
HDR_PIPE3_FL2FX	B874h

15.10.3.1.195.2 Function

Pipe1 floating point to fixed point control Pipe1 is also called channel1.

15.10.3.1.195.3 Diagram



15.10.3.1.195.4 Fields

Field	Function
31-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This Float to Fixed operation is enabled only for blended pixels 1: This Float to Fixed operation is enabled all pixels
0 ENABLE	0: Don't enable this Float-to-Fixed converter: Pixels pass thru the Float-to-Fixed unmodiifed 1: This Float-to-Fixed converter is enabled for current picture

15.10.3.1.196 PIPE3: NOT USED (HDR_PIPE3_ENTRY_30)

15.10.3.1.196.1 Offset

Register	Offset
HDR_PIPE3_ENTRY_30	B878h

15.10.3.1.196.2 Function

PIPE3: NOT USED

15.10.3.1.196.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.196.4 Fields

Field	Function
31-0 —	Reserved.

15.10.3.1.197 A0 component Linear-to-Non-linear conversion table (OPIPE_A0_TABLE)

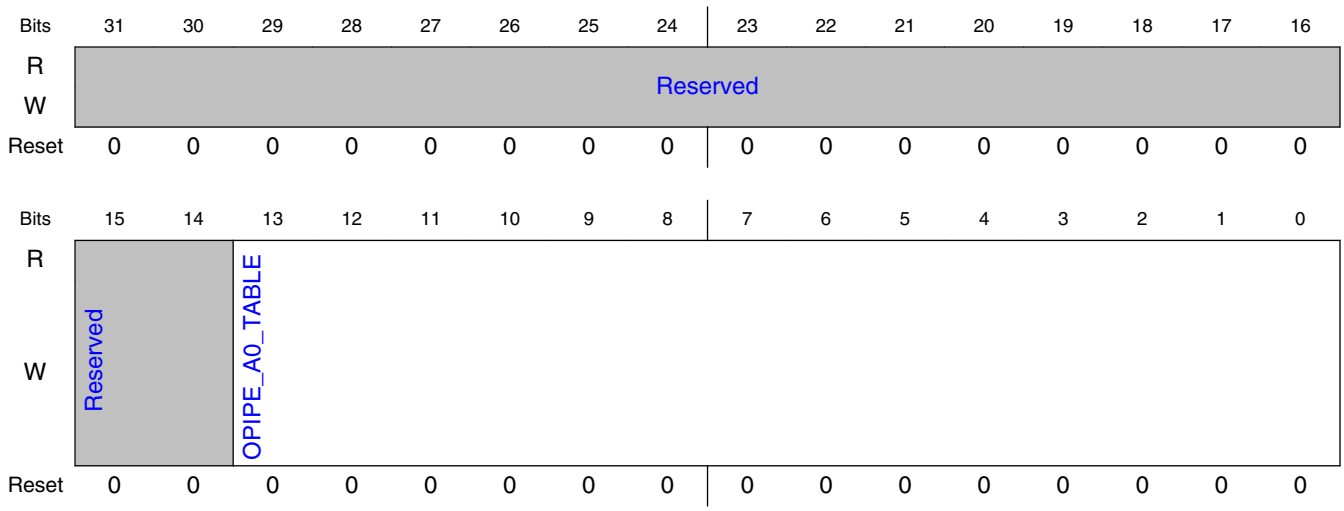
15.10.3.1.197.1 Offset

Register	Offset
OPIPE_A0_TABLE	C000h

15.10.3.1.197.2 Function

The Linear to Non Linear Conversion is done using a binary search algorithm. Oth stage Node is at: address 0 or 1; 1st stage Nodesare at: addresses 2-3; 2nd stage Nodes are at addresses 4-7; 3rd stage Nodes are at addresses 8-15; 4th stage Nodes are at addresses 16-31; 5th stage Nodes are at addresses 32-63; 6th stage Nodes are at addresses 64-127; 7th stage Nodes are at addresses 128-255; 8th stage Nodes are at addresses 256-511; The binary search algorithm traces the Nodes starting at 0th stage, then proceeding to the next stage. Inside a stage, the search comparisons procede "left-to-right" toward higher node address.

15.10.3.1.197.3 Diagram



15.10.3.1.197.4 Fields

Field	Function
31-14 —	Reserved.
13-0 OPIPE_A0_TABLE	The TABLE Nodes are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.198 A1 component Linear-to-Non-linear conversion table (OPIPE_A1_TABLE)

15.10.3.1.198.1 Offset

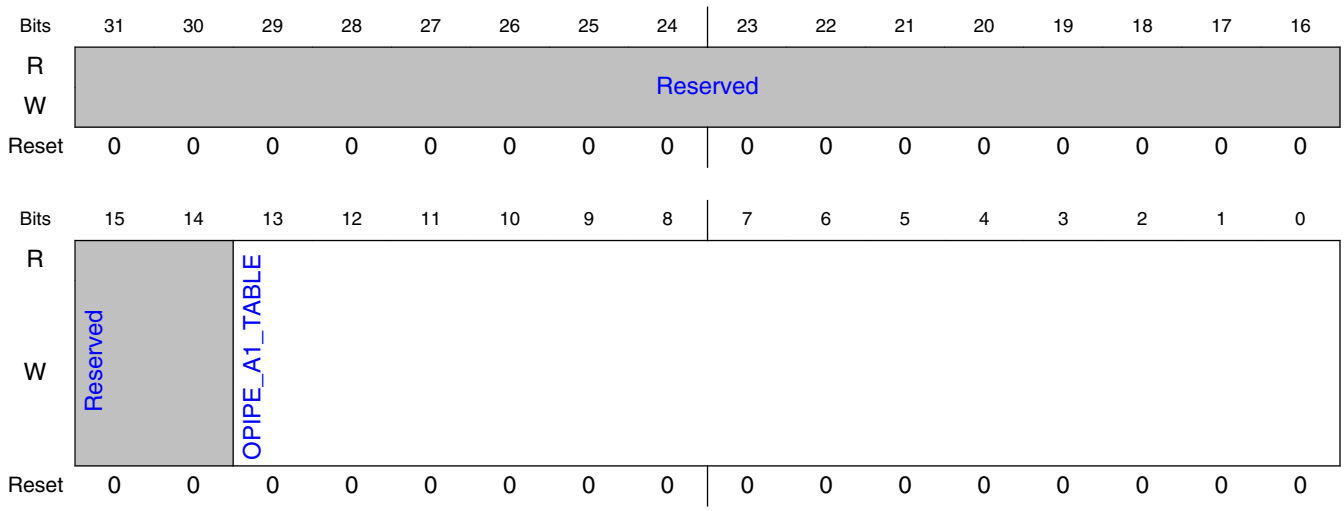
Register	Offset
OPIPE_A1_TABLE	D000h

15.10.3.1.198.2 Function

The Linear to Non Linear Conversion is done using a binary search algorithm. Oth stage Node is at: address 0 or 1; 1st stage Nodesare at: addresses 2-3; 2nd stage Nodes are at addresses 4-7; 3rd stage Nodes are at addresses 8-15; 4th stage Nodes are at addresses 16-31; 5th stage Nodes are at addresses 32-63; 6th stage Nodes are at addresses 64-127; 7th stage Nodes are at addresses 128-255; 8th stage Nodes are at addresses 256-511; The

binary search algorithm traces the Nodes starting at 0th stage, then proceeding to the next stage. Inside a stage, the search comparisons procede "left-to-right" toward higher node address.

15.10.3.1.198.3 Diagram



15.10.3.1.198.4 Fields

Field	Function
31-14 —	Reserved.
13-0 OPIPE_A1_TABLE	The TABLE Nodes are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.199 A2 component Linear-to-Non-linear conversion table (OPIPE_A2_TABLE)

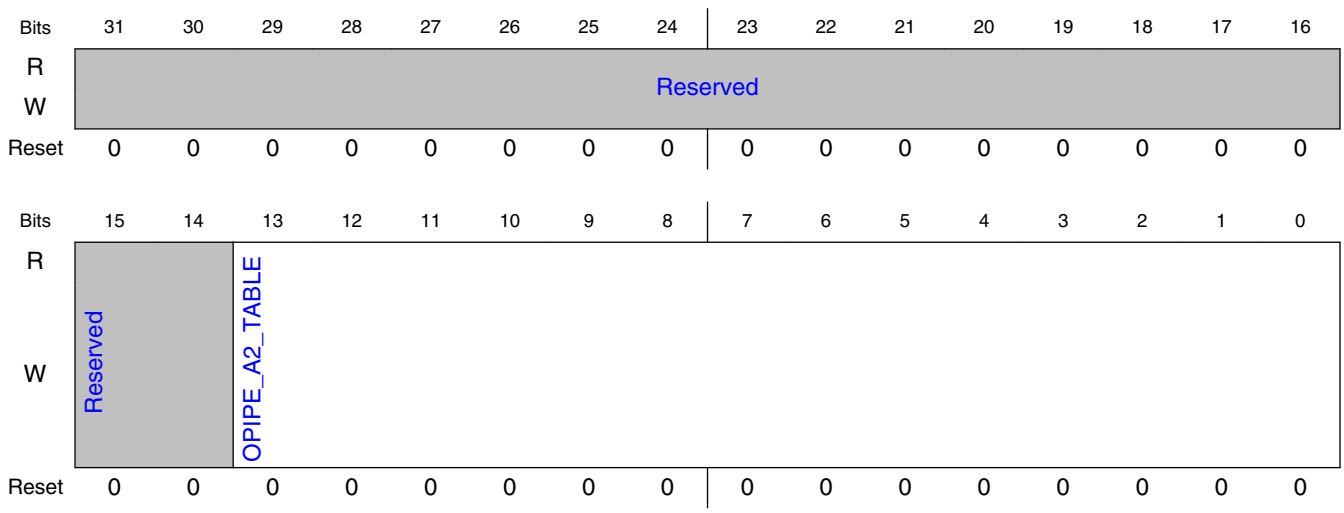
15.10.3.1.199.1 Offset

Register	Offset
OPIPE_A2_TABLE	E000h

15.10.3.1.199.2 Function

The Linear to Non Linear Conversion is done using a binary search algorithm. Oth stage Node is at: address 0 or 1; 1st stage Nodesare at: addresses 2-3; 2nd stage Nodes are at addresses 4-7; 3rd stage Nodes are at addresses 8-15; 4th stage Nodes are at addresses 16-31; 5th stage Nodes are at addresses 32-63; 6th stage Nodes are at addresses 64-127; 7th stage Nodes are at addresses 128-255; 8th stage Nodes are at addresses 256-511; The binary search algorithm traces the Nodes starting at 0th stage, then proceeding to the next stage. Inside a stage, the search comparisons procede "left-to-right" toward higher node address.

15.10.3.1.199.3 Diagram



15.10.3.1.199.4 Fields

Field	Function
31-14 —	Reserved.
13-0 OPIPE_A2_TABLE	The TABLE Nodes are 14 bits wide. They my be unsigned integers OR 14-bit floating point numbers

15.10.3.1.200 HDR output stage Colorspace Converter (CSCO) control. (HDR_OPIPE_CSC_CONTROL_REG)

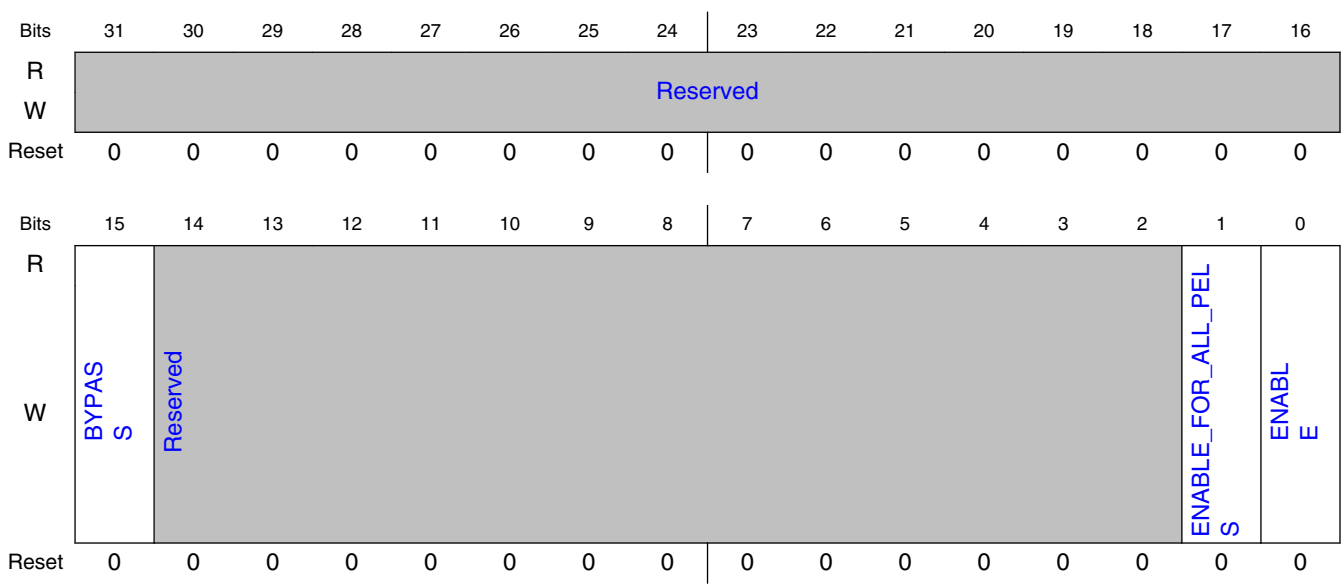
15.10.3.1.200.1 Offset

Register	Offset
HDR_OPIPE_CSC_CONTROL_REG	F000h

15.10.3.1.200.2 Function

Controls the Color-Space-Converter (CSC) in HDR output stage. This CSCO takes in 10-bit pixel components (component 0, component 1, component 2). A per-component offset (called pre-offset) is added to the pixel. After the pre-offset operation the pixel components are clipped. MAX_RANGE of PRE_CLIP is [-512, 1023]. A 3x3 matrix multiply (constant coefficiens H(x,y)) is performed on the pixel to take to a different color space. After this matrix multiply, a 28 bit signed offset (post-offset) is added to the normalized result of the matrix multiply. After the post-offset operation the results are clipped. MAX_RANGE of POST_CLIP is [0,1023] Output of this CSCO are 10 bit per component pixels. After CSCO the pixels are fed to the Chroma Subsampler.

15.10.3.1.200.3 Diagram



15.10.3.1.200.4 Fields

Field	Function
31-16 —	Reserved.

Table continues on the next page...

Field	Function
15 BYPASS	0: Don't bypass this CSC 1: Pixels pass thru this CSC unmodiifed
14-2 —	Reserved.
1 ENABLE_FOR_ALL_PELS	0: This CSC is enabled only for blened pixels 1: This CSC is enabled all pixels
0 ENABLE	0: Don't enable this CSC: Pixels pass thru the CSC unmodiifed 1: This CSC is enabled for current picture

15.10.3.1.201 Pipe1 Colorspace Converter (CSC) h(0,0) matrix coefficient (HDR_OPIPE_CSC_H00)

15.10.3.1.201.1 Offset

Register	Offset
HDR_OPIPE_CSC_H00	F004h

15.10.3.1.201.2 Function

h(0,0) coefficient in color space matrix multiply

15.10.3.1.201.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H00															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.201.4 Fields

Field	Function
31-16	Reserved.

Table continues on the next page...

Field	Function
—	
15-0 H00	h(0,0) 16 bit signed coefficient

15.10.3.1.202 Pipe1 Colorspace Converter (CSC) h(1,0) matrix coefficient (HDR_OPIPE_CSC_H10)

15.10.3.1.202.1 Offset

Register	Offset
HDR_OPIPE_CSC_H10	F008h

15.10.3.1.202.2 Function

h(1,0) coefficient in color space matrix multiply

15.10.3.1.202.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H10															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.202.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H10	h(1,0) 16 bit signed coefficient

15.10.3.1.203 HDR OUTPUT Colorspace Converter (CSCO) h(2,0) matrix coefficient (HDR_OPIPE_CSC_H20)

15.10.3.1.203.1 Offset

Register	Offset
HDR_OPIPE_CSC_H20	F00Ch

15.10.3.1.203.2 Function

h(2,0) coefficient in color space matrix multiply

15.10.3.1.203.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H20															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.203.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H20	h(2,0) 16 bit signed coefficient

15.10.3.1.204 HDR OUTPUT pipe Colorspace Converter (CSCO) h(0,1) matrix coefficient (HDR_OPIPE_CSC_H01)

15.10.3.1.204.1 Offset

Register	Offset
HDR_OPIPE_CSC_H01	F010h

15.10.3.1.204.2 Function

h(0,1) coefficient in color space matrix multiply

15.10.3.1.204.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H01															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.204.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H01	h(0,1) 16 bit signed coefficient

15.10.3.1.205 HDR OUTPUT pipe Colorspace Converter (CSCO) h(1,1) matrix coefficient (HDR_OPIPE_CSC_H11)

15.10.3.1.205.1 Offset

Register	Offset
HDR_OPIPE_CSC_H11	F014h

15.10.3.1.205.2 Function

h(1,1) coefficient in color space matrix multiply

15.10.3.1.205.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H11															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.205.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H11	h(1,1) 16 bit signed coefficient

15.10.3.1.206 HDR_output pipe Colorspace Converter (CSCO) h(2,1) matrix coefficient (HDR_OPIPE_CSC_H21)

15.10.3.1.206.1 Offset

Register	Offset
HDR_OPIPE_CSC_H21	F018h

15.10.3.1.206.2 Function

h(2,1) coefficient in color space matrix multiply

15.10.3.1.206.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H21															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.206.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H21	h(2,1) 16 bit signed coefficient

15.10.3.1.207 HDR OUTPUT pipe Colorspace Converter (CSCO) h(0,2) matrix coefficient (HDR_OPIPE_CSC_H02)

15.10.3.1.207.1 Offset

Register	Offset
HDR_OPIPE_CSC_H02	F01Ch

15.10.3.1.207.2 Function

h(0,2) coefficient in color space matrix multiply

15.10.3.1.207.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H02															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.207.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H02	h(0,2) 16 bit signed coefficient

15.10.3.1.208 HDR OUPUT pipe Colorspace Converter (CSCO) h(1,2) matrix coefficient (HDR_OPIPE_CSC_H12)

15.10.3.1.208.1 Offset

Register	Offset
HDR_OPIPE_CSC_H12	F020h

15.10.3.1.208.2 Function

h(1,2) coefficient in color space matrix multiply

15.10.3.1.208.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H12															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.208.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H12	h(1,2) 16 bit signed coefficient

15.10.3.1.209 HDR OUPUT pipe Colorspace Converter (CSCO) h(2,2) matrix coefficient (HDR_)

15.10.3.1.209.1 Offset

Register	Offset
HDR_	F024h

15.10.3.1.209.2 Function

h(2,2) coefficient in color space matrix multiply

15.10.3.1.209.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	H22															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.209.4 Fields

Field	Function
31-16 —	Reserved.
15-0 H22	h(2,2) 16 bit signed coefficient

15.10.3.1.210 HDR OUTPUT pipe Colorspace Converter (CSCO) component 0 pre-offset (HDR_OPIPE_CSC_IO_0)

15.10.3.1.210.1 Offset

Register	Offset
HDR_OPIPE_CSC_IO_0	F028h

15.10.3.1.210.2 Function

An signed 10-bit offset is added to component 0 (R,Y) before matrix multiply

15.10.3.1.210.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMPO_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.210.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMPO_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 0 of the pixel

15.10.3.1.211 HDR OUPTUT pipe Colorspace Converter (CSCO) component 1 pre-offset (HDR_OPIPE_CSC_IO_1)

15.10.3.1.211.1 Offset

Register	Offset
HDR_OPIPE_CSC_IO_1	F02Ch

15.10.3.1.211.2 Function

An signed 10-bit offset is added to component 1 (G,Cb) before matrix multiply

15.10.3.1.211.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.211.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 1 of the pixel

15.10.3.1.212 HDR OUPUT pipe: Colorspace Converter (CSCO) component 2 pre-offset (HDR_OPIPE_CSC_IO_2)

15.10.3.1.212.1 Offset

Register	Offset
HDR_OPIPE_CSC_IO_2	F030h

15.10.3.1.212.2 Function

An signed 10-bit offset is added to component 2 (B,Cr) before matrix multiply

15.10.3.1.212.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_PRE_OFFSET									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.212.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_PRE_OFFSET	Before the color space conversion matrix multiply, This offset is added to component 2 of the pixel

15.10.3.1.213 HDR OUPTU pipe Colorspace Converter (CSCO) component 0 clip min. (HDR_OPIPE_CSC_MIN_0)

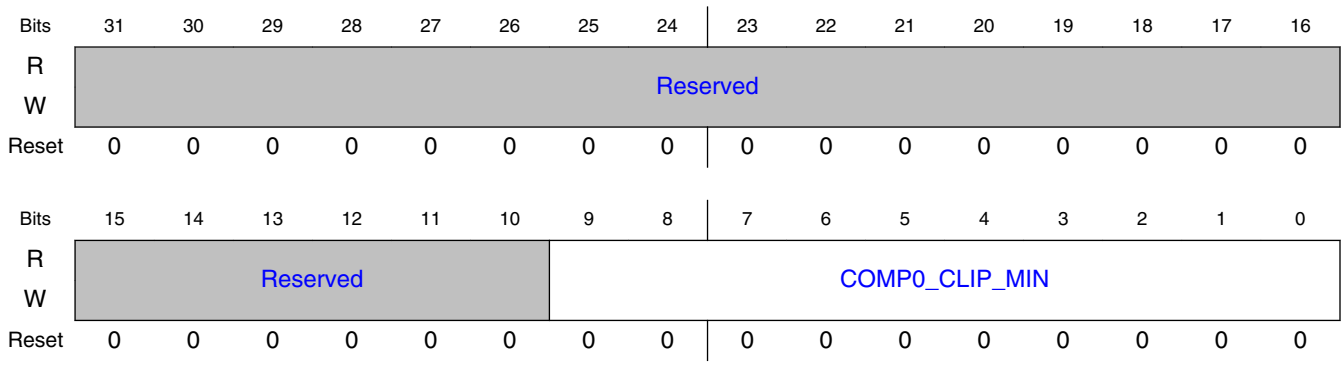
15.10.3.1.213.1 Offset

Register	Offset
HDR_OPIPE_CSC_MIN_0	F034h

15.10.3.1.213.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511].

15.10.3.1.213.3 Diagram



15.10.3.1.213.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.214 HDR OUPUT pipe Colorspace Converter (CSCO) component 1 clip min. (HDR_OPIPE_CSC_MIN_1)

15.10.3.1.214.1 Offset

Register	Offset
HDR_OPIPE_CSC_MIN_1	F038h

15.10.3.1.214.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511].

15.10.3.1.214.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP1_CLIP_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.214.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.215 HDR OUPTU pipe Colorspace Converter (CSCO) component 2 clip min. (HDR_OPIPE_CSC_MIN_2)

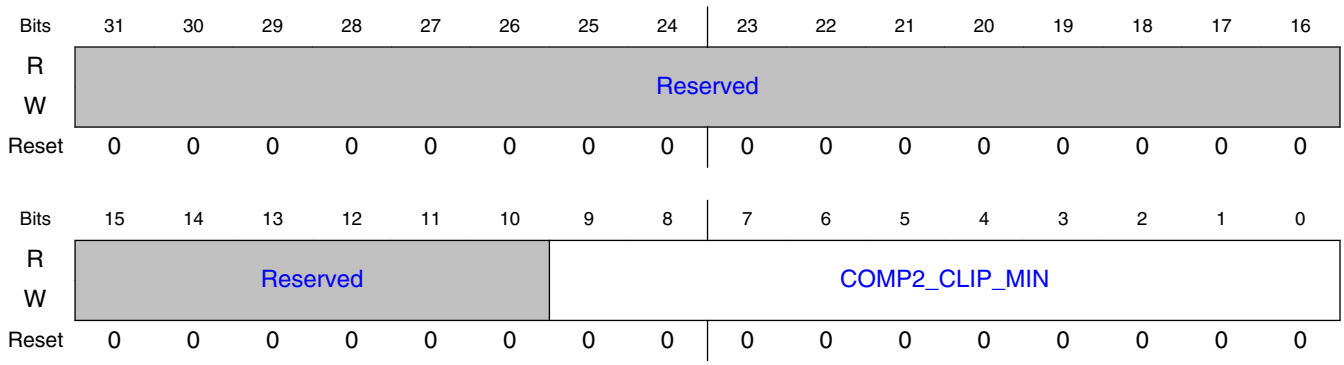
15.10.3.1.215.1 Offset

Register	Offset
HDR_OPIPE_CSC_MIN_2	F03Ch

15.10.3.1.215.2 Function

After the pre-increment, The result is clipped. This 10-bit signed value is the minimum value of pixel component after the pre-increment. So the minimun range is [-512,511].

15.10.3.1.215.3 Diagram



15.10.3.1.215.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MIN	This 10-bit signed value is the minimum value of pixel component after the pre-increment.

15.10.3.1.216 HDR OUPTUT pipe Colorspace Converter O (CSC) component 0 clip max value. (HDR_OPIPE_CSC_MAX_0)

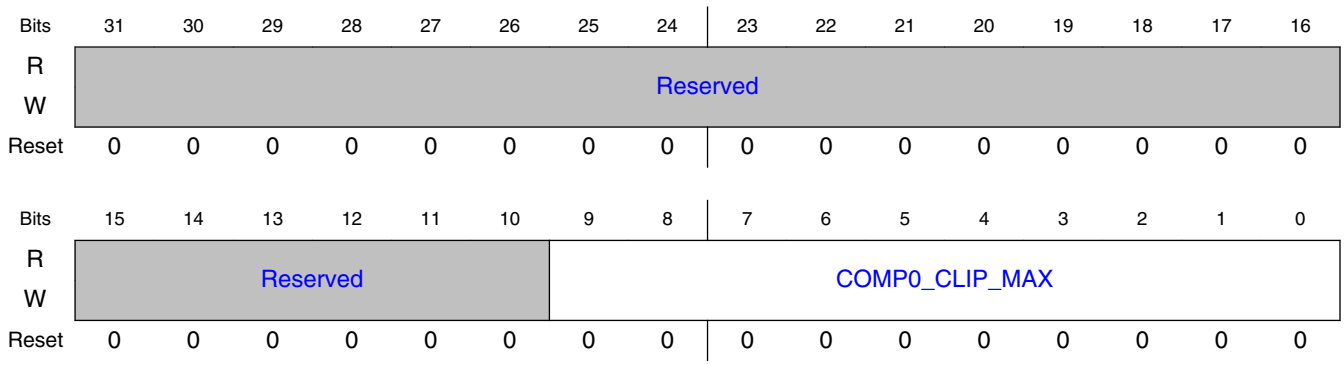
15.10.3.1.216.1 Offset

Register	Offset
HDR_OPIPE_CSC_MAX_0	F040h

15.10.3.1.216.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023].

15.10.3.1.216.3 Diagram



15.10.3.1.216.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP0_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.217 HDR OUTPUT pipe Colorspace Converter (CSCO) component 1 clip max value. (HDR_OPIPE_CSC_MAX_1)

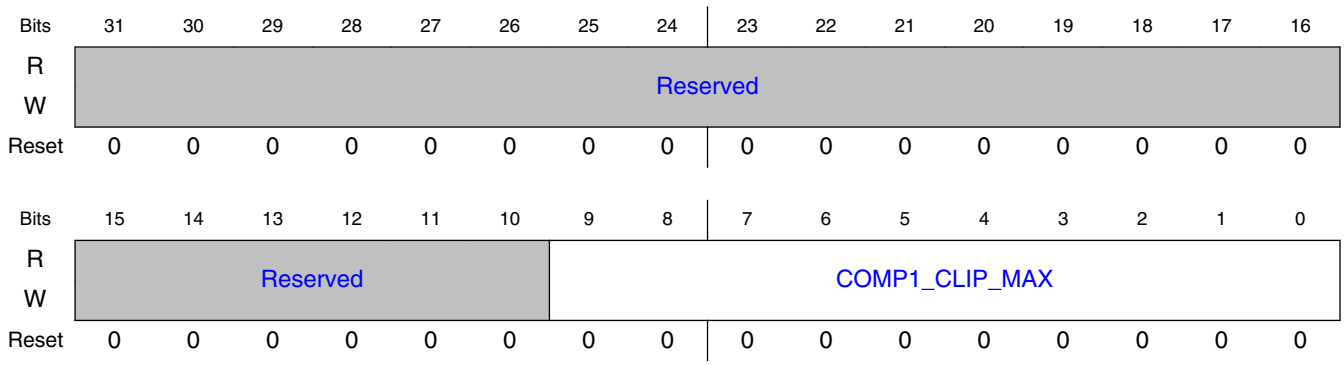
15.10.3.1.217.1 Offset

Register	Offset
HDR_OPIPE_CSC_MAX_1	F044h

15.10.3.1.217.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023].

15.10.3.1.217.3 Diagram



15.10.3.1.217.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP1_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.218 HDR OUTPUT pipe Colorspace Converter (CSCO) component 2 clip max value. (HDR_OPIPE_CSC_MAX_2)

15.10.3.1.218.1 Offset

Register	Offset
HDR_OPIPE_CSC_MAX_2	F048h

15.10.3.1.218.2 Function

After the pre-increment, The result is clipped. This 10-bit unsigned value is the maximum value of pixel component after the pre-increment. So the maximum range is [0,1023].

15.10.3.1.218.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						COMP2_CLIP_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.218.4 Fields

Field	Function
31-10 —	Reserved.
9-0 COMP2_CLIP_MAX	This 10-bit unsigned value is the maximun value of pixel component after the pre-increment.

15.10.3.1.219 HDR OUPUT pipe Colorspace Converter (CSCO) normalization factor (HDR_OPIPE_CSC_NORM)

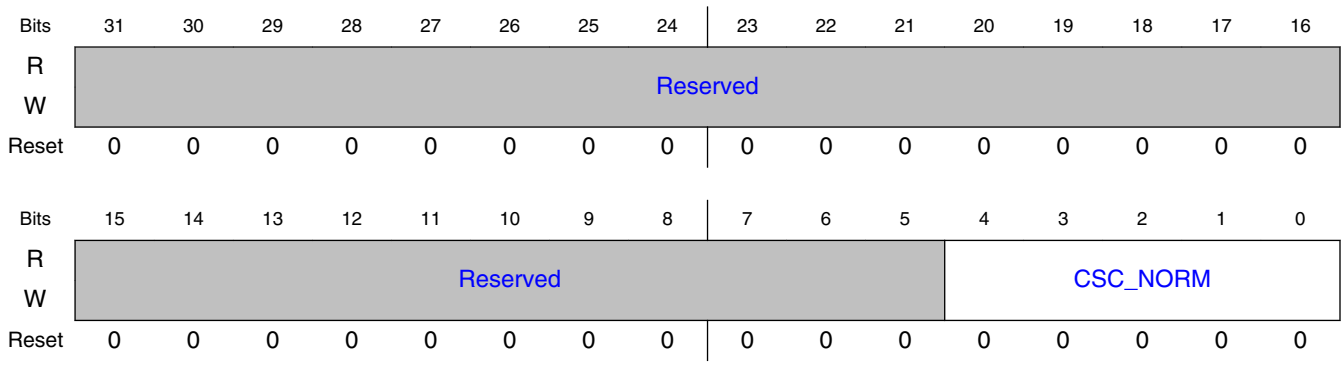
15.10.3.1.219.1 Offset

Register	Offset
HDR_OPIPE_CSC_NORM	F04Ch

15.10.3.1.219.2 Function

After the CSC matrix multiply, all components of the result are shifted right by the amount in this register. This is a singed right shift.

15.10.3.1.219.3 Diagram



15.10.3.1.219.4 Fields

Field	Function
31-5 —	Reserved.
4-0 CSC_NORM	This 5-bit unsigned value is size if arithmetic shift after matrix multiply.

15.10.3.1.220 HDR OUPTUT pipe Colorspace Converter (CSC): Post offset component 0 (HDR_OPIPE_CSC_OO_0)

15.10.3.1.220.1 Offset

Register	Offset
HDR_OPIPE_CSC_OO_0	F050h

15.10.3.1.220.2 Function

After the CSC normalization, this offset value is added to component 0. This is a signed 28-bit number.

15.10.3.1.220.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				CSC_OO_0											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSC_OO_0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.220.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSC_OO_0	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.221 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset component 1 (HDR_OPIPE_CSC_OO_1)

15.10.3.1.221.1 Offset

Register	Offset
HDR_OPIPE_CSC_OO_1	F054h

15.10.3.1.221.2 Function

After the CSC normalization, this offset value is added to component 1. This is a signed 28-bit number.

15.10.3.1.221.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved				CSC_OO_1											
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	CSC_OO_1															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.221.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSC_OO_1	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.222 HDR OUPTUT pipe Colorspace Converter (CSC): Post offset component 2 (HDR_OPIPE_CSC_OO_2)

15.10.3.1.222.1 Offset

Register	Offset
HDR_OPIPE_CSC_OO_2	F058h

15.10.3.1.222.2 Function

After the CSC normalization, this offset value is added to component 2. This is a signed 28-bit number.

15.10.3.1.222.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved				CSC_OO_2											
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CSC_OO_2															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.222.4 Fields

Field	Function
31-28 —	Reserved.
27-0 CSC_OO_2	Ouput Offset (OO) This is a signed 28-bit number. Per component

15.10.3.1.223 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 0 (HDR_OPIPE_CSC_OMIN_0)

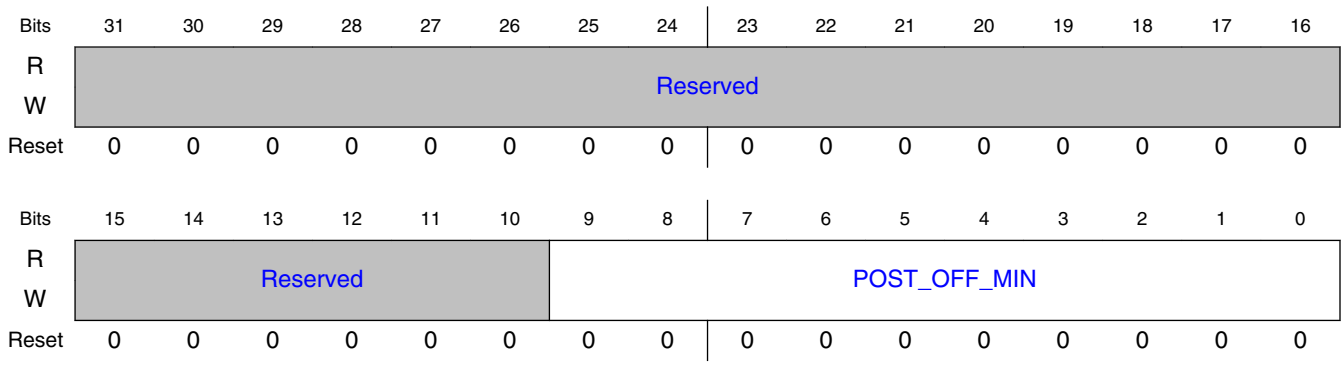
15.10.3.1.223.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMIN_0	F05Ch

15.10.3.1.223.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number.

15.10.3.1.223.3 Diagram



15.10.3.1.223.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.224 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 1 (HDR_OPIPE_CSC_OMIN_1)

15.10.3.1.224.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMIN_1	F060h

15.10.3.1.224.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number.

15.10.3.1.224.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.224.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.225 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset min clip value for component 2 (HDR_OPIPE_CSC_OMIN_2)

15.10.3.1.225.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMIN_2	F064h

15.10.3.1.225.2 Function

After the post offset is added thie component is clipped. This is the minimum clip value This is an unsigned 10-bit number.

15.10.3.1.225.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.225.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MIN	Minum clipped pixel component. [0-1023]

15.10.3.1.226 HDR OUPTUT pipe Colorspace Converter (CSC): Post offset max clip value for component 0 (HDR_OPIPE_CSC_OMAX_0)

15.10.3.1.226.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMAX_0	F068h

15.10.3.1.226.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number.

15.10.3.1.226.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						POST_OFF_MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.10.3.1.226.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.227 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset max clip value for component 1 (HDR_OPIPE_CSC_OMAX_1)

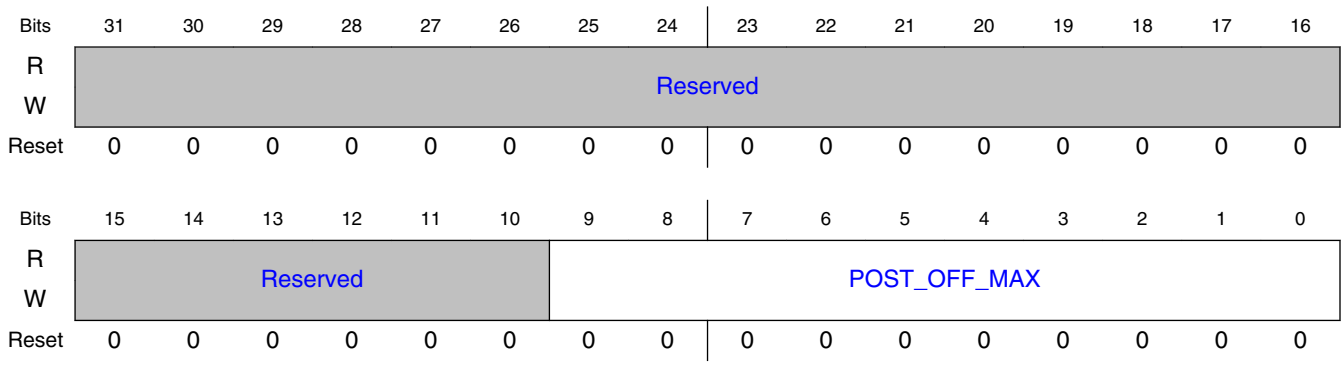
15.10.3.1.227.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMAX_1	F06Ch

15.10.3.1.227.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number.

15.10.3.1.227.3 Diagram



15.10.3.1.227.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.228 HDR OUTPUT pipe Colorspace Converter (CSC): Post offset max clip value for component 2 (HDR_OPIPE_CSC_OMAX_2)

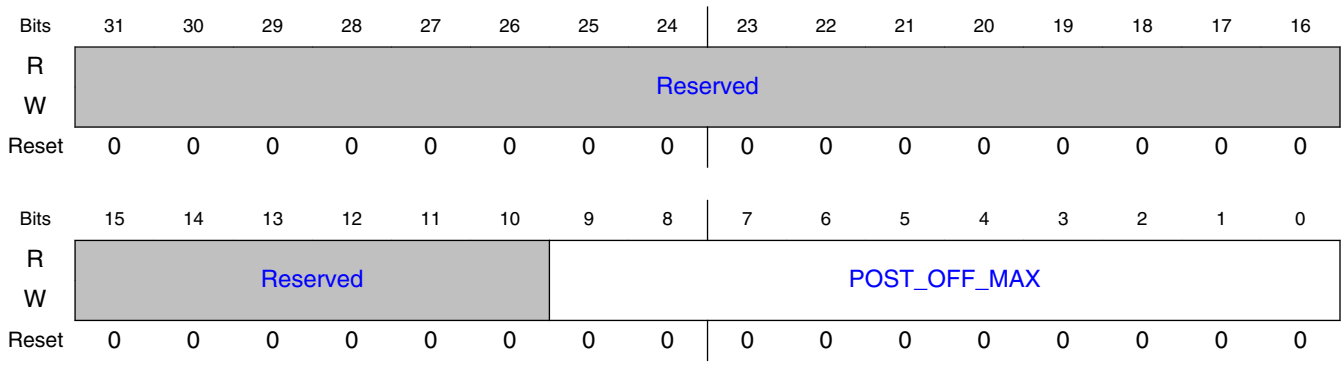
15.10.3.1.228.1 Offset

Register	Offset
HDR_OPIPE_CSC_OMAX_2	F070h

15.10.3.1.228.2 Function

After the post offset is added thie component is clipped. This is the maximum clip value This is an unsigned 10-bit number.

15.10.3.1.228.3 Diagram



15.10.3.1.228.4 Fields

Field	Function
31-10 —	Reserved.
9-0 POST_OFF_MAX	Maximum clipped pixel component. [0-1023]

15.10.3.1.229 HDR OUTPUT -TO NON LINEAR pipeline control (HDR_OPIPE_2NL_CONTROL_REG)

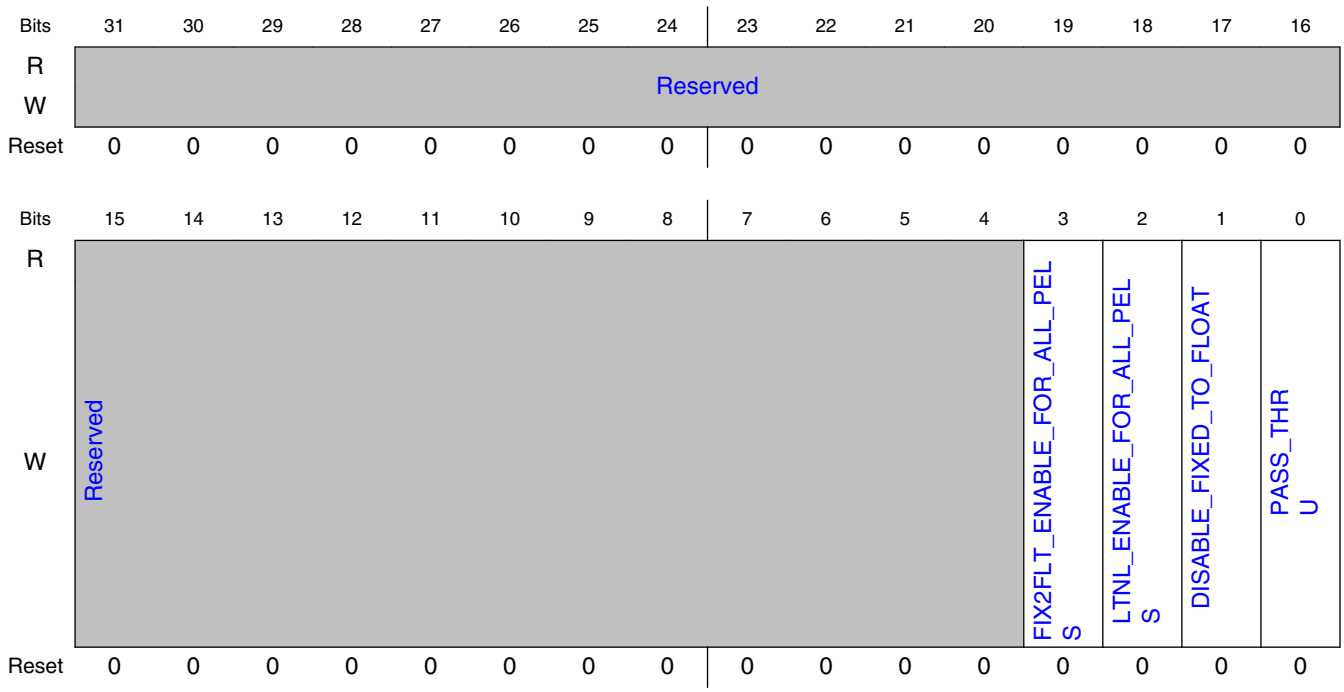
15.10.3.1.229.1 Offset

Register	Offset
HDR_OPIPE_2NL_CONTROL_REG	F874h

15.10.3.1.229.2 Function

HDR OUTPUT -TO NON LINEAR pipeline control and fixed_to_float control

15.10.3.1.229.3 Diagram



15.10.3.1.229.4 Fields

Field	Function
31-4 —	Reserved.
3 FIX2FLT_ENABLE_FOR_ALL_PELS	0: fixed-to_float is enabled for blended pels 1: fixed-to_float is enabled for all pels
2 LTNL_ENABLE_FOR_ALL_PELS	0: When 0 linear_to_non linear conversion is enabled for blended pels 1: When 1 linear_to_non linear conversion is enabled for all pels
1 DISABLE_FIXED_TO_FLOAT	0: The 28 bit input component is converted to floating point format to be processed by the linear to non-linear converter 1: The 28 bit fixed point number is truncated to 14 bits and passed the passed to the linear to non linear pipe
0 PASS_THRU	0: Process the pixels with linear to non-linear pipeline 1: Pass the data through the linear to non-linear pipeline unmodified

15.11 Color Sub-Sampler (SUBSAM)

15.11.1 Overview

The Color Sub-Sampler, or SUBSAM, is used to reduce the resolution of chroma samples for display that require YCbCr422/420 formats. Based on the HDMI2.0 specification, YCbCr420 is required for 4K displays at a 60Hz rate. There are three primary modes of operation: YCbCr444, the SUBSAM is in bypass. YCbCr422 conversions, a 3-tap horizontal linear filter is applied to the incoming pixel stream. YCbCr420, a 3x3 tap filter is used for sub-sampling. One Y line buffer and two of each Cb and Cr line buffers are used to facilitate the vertical scaling function. The chroma line buffers store the horizontally scaled result.

15.11.1.1 Block Diagram

The following diagram gives a high-level overview of the configuration of the SUBSAM.

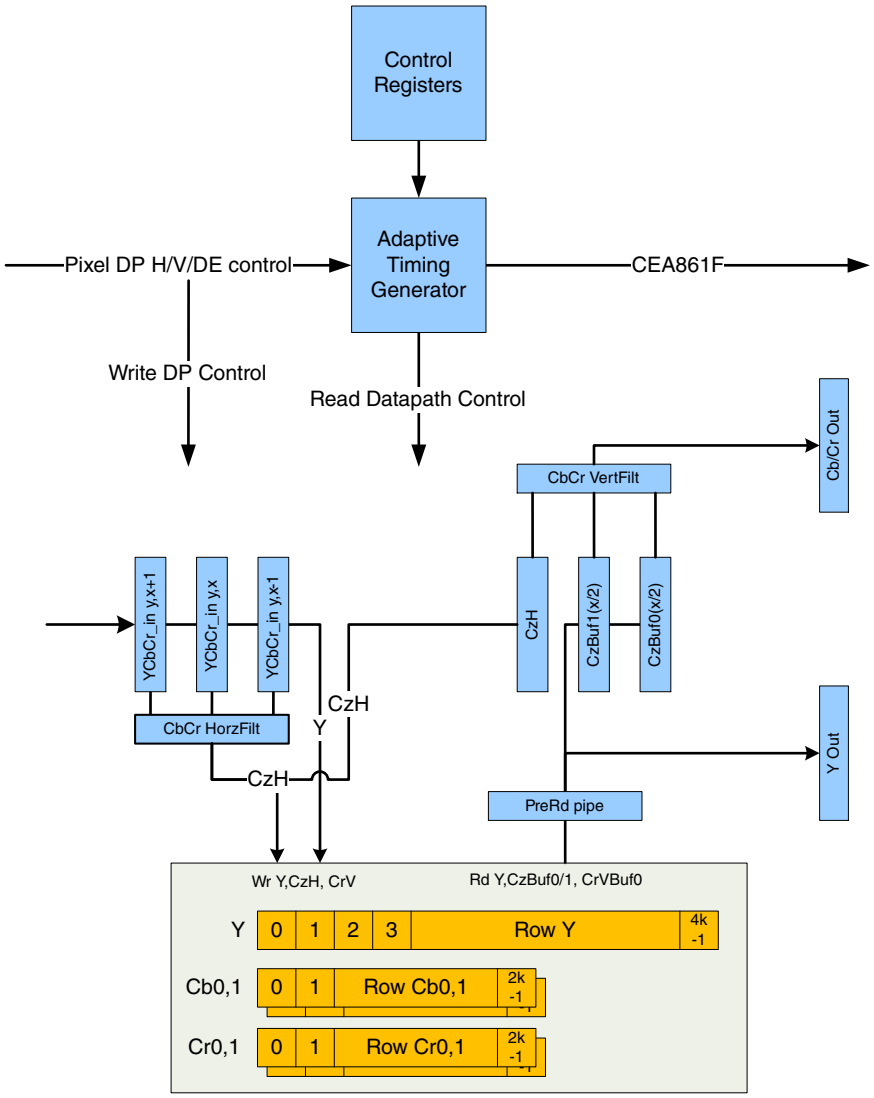


Figure 15-23. Color Subsampler Block Diagram

15.11.1.2 Features

From the block diagram above, the operands for the horizontal chroma subsampler, or CzH, are stored in shift registers. These registers are CbCr_in[x+1,x,x-1]. The results of the CzH are used to feed the vertical 3 tap filter and stored in line memories for subsequent vertical scaling. Line storage is used to store horizontally scaled chroma values and to store luma components. These components are sourced with one line of delay to feed the vertical scaling function. Luma values are stored unscaled, since luma is not subsampled.

These memories are single ported. The design pipelines the write and read data so that the same location is not accessed on the same clock. The preread function reads data out of the row rams so that reads lead locations that are written. Only 420 subsample mode uses

the memories due to the vertical scale requirement. 422 does not use the memories and the CzH is fed directly out of the subsample module aligned with the respective Y components.

15.11.2 Functional description

15.11.2.1 Sub-Sample Adaptive Timing Generator

The adaptive timing generator (ATG) uses the timing of the pixel data path signals to generate the timing of the CEA861F timing that is output to the HDMI controller. The timing is regenerated in the SUBSAM module, and the timing is completely programmable. Only the input pixdp_vsync is used to initiate the timing sequence at the SUBSAM output. When pixdp_vsync goes active (low to high transition), the SUBSAM resets its own H/V counters (h_count and v_count). The SUBSAM timing generator then walks across the entire display based on the programmable timing parameters defined in the SUBSAM PIO registers. It is expected the SUBSAM ATG timing parameters are consistent with the DTG timing parameters. As an example, the same LRC_X and LRC_Y values are programmed so that the display cycle time generated in the DTG is equivalent to the timing in the SUBSAM.

The display control is sequenced using the h_count/v_count counters. h_count is incremented on every clock, and is reset when it is equal to CSR_SS_DISPLAY.LRC_X. v_count is incremented, or reset, on the clock when h_count is reset to zero. The h_count value and respective comparators walks the display from left to right. The v_count value and vertical comparators walks the display timing from top to bottom. When the lower right corner (LRC) is reach, the timing generator resets back to the upper left corner (ULC) and begins to refresh the display again. This cycle continues indefinitely until the SUBSAM is disabled.

15.11.2.2 ATG Timing Details

Out of reset, the ATG timing sequence begins after receiving an initial positive edge of pixdp_vsync. This initial positive edge indicates the DTG is entering the vertical front porch. In order to synchronize the ATG HDMI output timing with the DTG pixel data path input timing, the ATG must also begin timing to enter the vertical front porch. This concept can be illustrated in the image below, and the timing begins when pixdp_vsync (also named dolby_vsync_n) transitions from low to high. Note the timing of pixdp_vsync is not the same as hdmi_vsync.

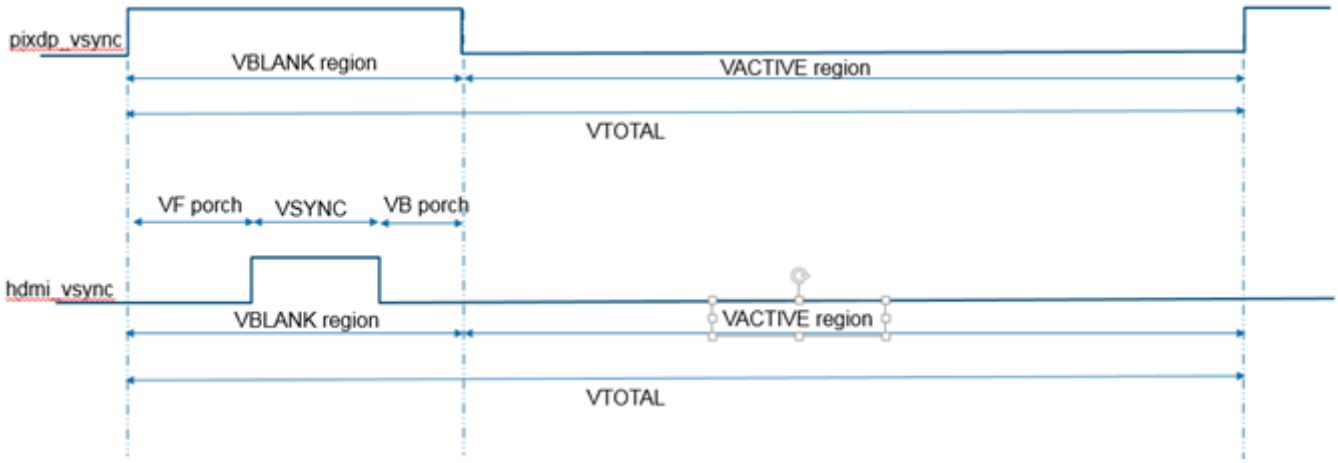


Figure 15-24. Initial VSYNC Timing

The following image illustrates the addition of pixdp_hsync and hdmi_de timing. Note the HDMI DE timing is based on the pixdp_hsync timing. This is based on how the pixel data path defines the timing of HSYNC and DE, which is not standard, and different then HDMI CEA861 timing standard.

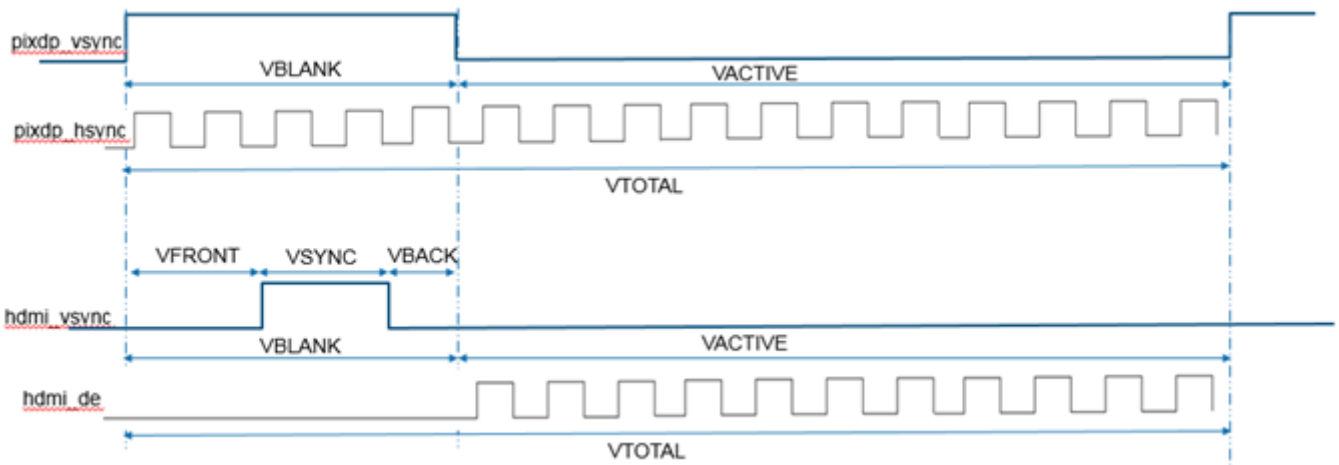


Figure 15-25. Initial VSYNC Timing with HSYNC and DE

The following provides an equivalence between standard HDMI timing and the ATG programming model. A waveform representation of CEA-861F timing is provided to illustrate which waveform sections correspond to which programming parameters. Typical HDMI timing parameters can be extracted from the following image:

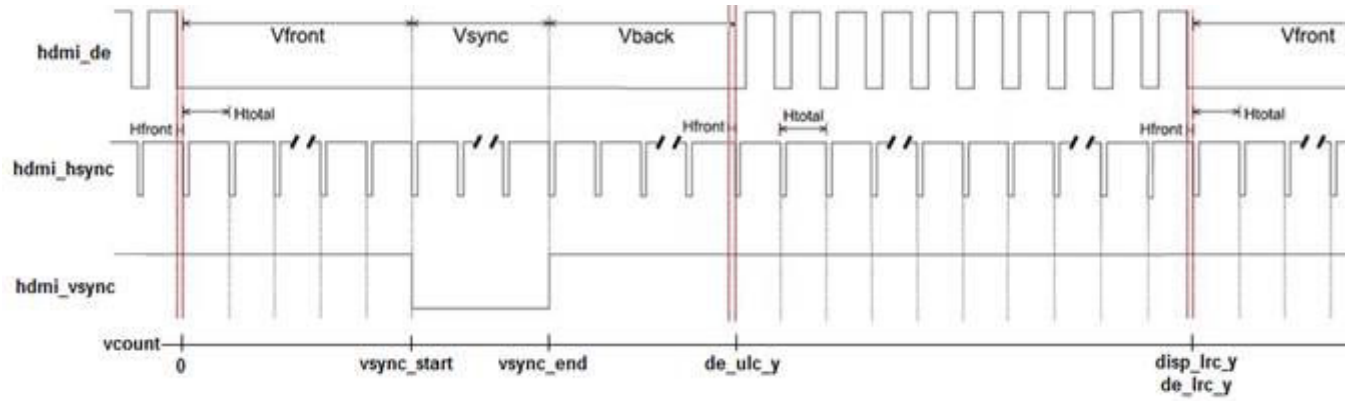


Figure 15-26. Typical HDMI Timing Parameters

- VFront = vsync_start + 1 (duration of VFront porch)
- VSync = vsync_end - vsync_start
- VBack = de_ulc_y - 1 - vsync_end
- VActive = disp_lrc_y - de_ulc_y - 1
- VTotal = disp_lrc_y + 1 = de_lrc_y + 1
- disp_lrc_y = de_lrc_y

In addition to the vertical programming guidelines shown in the image above, a set of horizontal programming guidelines should also be followed.

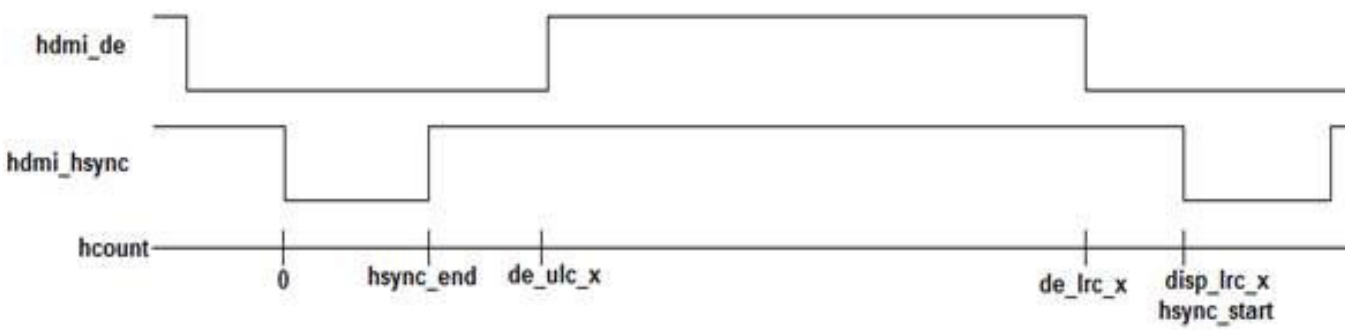


Figure 15-27. Typical HDMI horizontal timing parameters

Typical HDMI timing parameters can be extracted from the horizontal image above:

- HSync = hsync_end + 1
- HBack = de_ulc_x - hsync_end
- HActive = de_lrc_x - de_ulc_x
- HFront = disp_lrc_x - de_lrc_x
- HTotal = disp_lrc_x + 1 = hsync_start + 1
- disp_lrc_x = hsync_start

A table represetation of the programming example above is provided here:

Table 15-8. Subsampler Timing Registers

Subsampler Register	Register Setting
SS_DISPLAY[LRC_X]	Htotal - 1
SS_DISPLAY[LRC_Y]	Vtotal - 1
SS_HSYNC[START]	SS_DISPLAY[LRC_X]
SS_HSYNC[END]	Hsync - 1
SS_VSYNC[START]	Vfront - 1
SS_VSYNC[END]	Vsync + SS_VSYNC[START]
SS_DE_ULC[ULC_X]	Hback + SS_HSYNC[END]
SS_DE_ULC[ULC_Y]	Vback + SS_VSYNC[END] + 1
SS_DE_LRC[LRC_X]	Hactive + SS_DE_ULC[ULC_X]
SS_DE_LRC[LRC_Y]	SS_DISPLAY[LRC_Y]

15.11.2.3 SUBSAM Programming Example

The table below indicates the timing parameters for the Video Information Code (97, 107) defined for a progressive 4K TV at 60Hz.

Table 15-9. Video Information Code (97,107)

Hactive	Vactive	Htotal	Hblank	Vtotal	Vblank
3840	2160	4400	560	2250	90
Hfront	Hsync	Hback	Vfront	Vsync	Vback
176	88	296	8	10	72

The following programming example illustrates the practical use of the programming parameters derived in the ATG Timing Details section above. The SUBSAM parameters for this display format are:

- SS_DISPLAY[LRC_X] = Htotal - 1 = 4400 - 1 = 4399
- SS_DISPLAY[LRC_Y] = Vtotal - 1 = 2250 - 1 = 2249
- SS_HSYNC[START] = SS_DISPLAY[LRC_X] = 4399
- SS_HSYNC[END] = Hsync - 1 = 88 - 1 = 87
- SS_DE_ULC[ULC_X] = Hback + SS_HSYNC[END] = 296 + 87 = 383
- SS_DE_LRC[LRC_X] = Hactive + SS_DE_ULC[ULC_X] = 3840 + 383 = 4223
- SS_VSYNC[START] = Vfront - 1 = 8 - 1 = 7
- SS_VSYNC[END] = Vsync + SS_VSYNC[START] = 10 + 7 = 17
- SS_DE_ULC[ULC_Y] = Vback + SS_VSYNC[END] + 1 = 72 + 17 + 1 = 90
- SS_DE_LRC[LRC_Y] = SS_DISPLAY[LRC_Y] = 2249

Note that this example works for all subsampling modes (444, 422, and 420). To select between modes, the register field SS_MODE[PIPE_MODE] should be programmed. Each mode requires a different amount of processing latency. This latency is abstracted from the programming model and is accounted for in hardware based on the selection of SS_MODE[PIPE_MODE].

15.11.2.4 Sub-Sample Frame Sequence

The following table indicates the sequence of 420 subsampling over the first 6 lines of output. There is a nested looping structure iterating over Y and X pixel positions. The inner loop is iterating over the possible X values of the display.

The table represents what is done on each pixel clock cycle. Basically, x=0,1,…4K-1, then x resets to zero and the line increments. The first 7 lines of input processing is display, not the entire display of course. For 420, the Y loop repeats every two lines. The sequence reads as follows:

First horizontal iteration:

1. Pixel 0,0 is received, x=0
2. Hflit, CzH = H((x-1,x,x_1), or H(-1,0,1) on the first line. Both CbH and CrH are computed.
3. Vflit, ignored, since there is no vertical scaling when the first scan line is received.
4. No output, there is no output when the first input line is received, as we don’t have enough chroma data for the vertical filtering operation.
5. Store the Luma value in Ybuf(0) at address 0.
6. Store the CbH value in CbBuff[0](0)
7. No storage to CbBuf[1]
8. Store CrBuf[0](0)
9. No storage to CrBuff[1]

Second horizontal iteration. Note for entires where x/2 are not integers, then the operation is not performed.

1. At this point, x in incremented to 1 and the operation in the rist row repeat. The pixel position is now (1,0). Note all operations from 1-9 occur in the same clock cycle in a pipelined fashion in HW.
2. Hfilt is ignored, since x is odd, and the horizontal filter is valid only when x is even.
3. Again, ignored for the first input line.
4. ….

This loop continues until we reach the end of the scan line. This is determined when the logical DE goes low. The logical DE is a logic combination of Vsync, Hsync, and DE inputs from the pixel data path. Logical DE is active as follows:

- logical_de = ~pixdp_vsync && pixdp_hsync && pixdp_de

When all of the pixel are processed in the horizontal direction, processing increments to the next line, or all possible values of x for (x,1) for line 1.

Input Pixel YCbCr444(x,y) x = 0,1,… 3839	Hfilt (CzH) z = {b,r} x = even	Vfilt (CzV) z = {b,r}	Output Pixel 420	Ybuf(x)	CbBuf(0) (x/2)	CbBuf(1) (x/2)	CrBuf(0) (x/2)	CrBuf(1) (x/2)
1	2	3	4	5	6	7	8	9
(x,0)	CzH(x,0) = H(Cz(x-1, 0), Cz(x, 0), Cz(x +1,0))		none	Yin(x,0)	CbH(x,0)		CrH(x,0)	
(x,1)	CzH(x,1) = H(Cz(x-1, 1), Cz(x, 1), Cz(x +1,1))	CzV = V(bg_z, CzBuf(0) (x/2), CzH(x,1))	Ybuf(x), CbV	Yin(x,1)		CbH(x,1)	CrV(x,0)	CrH(x,1)
(x,2)	CzH(x,2) = H(Cz(x-1, 2), Cz(x, 2), Cz(x +1,2))	none	Ybuf(x), CrBuf(0) (x/2)	Yin(x,2)	CbH(x,2)		CrH(x,2)	
(x,3)	CzH(x,3) = H(Cz(x-1, 3),	CzV = V(CzBuf(1)(x/2), CzBuf(0)	Ybuf(x), CbV	Yin(x,3)		CbH(x,3)	CrV(x,2)	CrH(x,3)

Table continues on the next page...

Input Pixel YCbCr444(x,y) x = 0,1,… 3839	Hfilt (CzH) z = {b,r} x = even	Vfilt (CzV) z = {b,r}	Output Pixel 420	Ybuf(x)	CbBuf(0) (x/2)	CbBuf(1) (x/2)	CrBuf(0) (x/2)	CrBuf(1) (x/2)
	Cz(x, 3), Cz(x +1,3))	(x/2), CzH(x,3))						
(x,4)	CzH(x,4) = H(Cz(x-1, 4), Cz(x, 4), Cz(x +1,4))	none	Ybuf(x), CrBuf(0) (x/2)	Yin(x,4)	CbH(x,4)		CrH(x,4)	
(x,5)	CzH(x,5) = H(Cz(x-1, 5), Cz(x, 5), Cz(x +1,5))	CzV = V(CzBuf(1)(x/2), CzBuf(0) (x/2), CzH(x,5))	Ybuf(x), CbV	Yin(x,5)		CbH(x,5)	CrV(x,4)	CrH(x,5)
(x,6)	CzH(x,6) = H(Cz(x-1, 6), Cz(x, 6), Cz(x +1,6))	none	Ybuf(x), CrBuf(0) (x/2)	Yin(x,6)	CbH(x,6)		CrH(x,6)	

15.11.2.5 Sub-Sample Filter

For 422 conversions, the memory is bypassed, since the scaled result is a horizontal linear filter. Line storage is not needed in this case. For 420, line memory is needed to perform the 3 tap vertical filter.

The conversion for 422 is as follows:

```
        Cx = ( Hcoef0*Cx(x-1) + Hcoef1*Cx(x) + Hcoef2*Cx(x+1) + RoundX ) / NormX  // Where
Cx = Cb|Cr
```

Note: Output chroma values are computed at pixel positions where x is even. So, 0, 2, 4, etc are positions where pixel values are generated and output to the pixel data path.

The conversion for 420 is as follows:

```
      Cz = (
             Vcoef0 * (Hcoef0*Cz(x-1)(y-1) + Hcoef1*Cz(x)(y-1) + Hcoef2*Cz(x+1)(y-1) ) +
             Vcoef1 * (Hcoef0*Cz(x-1)(y) + Hcoef1*Cz(x)(y) + Hcoef2*Cz(x+1)(y) )       +
             Vcoef2 * (Hcoef0*Cz(x-1)(y+1) + Hcoef1*Cz(x)(y+1) + Hcoef2*Cz(x+1)(y+1) ) +
             RoundY
           ) / NormY
```

Note: x,y => can only take even values in the calculation.

There is also a programmable clipping function applied in the sub_sample module. The final Cz output value is:

```
         Cz_out = clip(Cz, minCzvalue, maxCzvalue)
```

The filter equations above use variable substitutions above for readability. To equate the variables to register settings, the following conversions should be used:

Hcoef0 = SS_COEFF[HORIZ_C]

Hcoef1 = SS_COEFF[HORIZ_B]

Hcoef2 = SS_COEFF[HORIZ_A]

If (SS_COEFF[HORIZ_NORM] = 0) RoundX = 0; NormX = 1;

else if (SS_COEFF[HORIZ_NORM] < 5) RoundX = 2^(NORM); NormX = 2^(NORM +1);

else RoundX = 2^(5); NormX = 2^(5+1);

Vcoef0 = SS_COEFF[VERT_C]

Vcoef1 = SS_COEFF[VERT_B]

Vcoef2 = SS_COEFF[VERT_A]

If (SS_COEFF[VERT_NORM] = 0) RoundY = 0; NormY = 1;

else if (SS_COEFF[VERT_NORM] < 5) RoundY = 2^(SS_COEFF[VERT_NORM]); NormY = 2^(SS_COEFF[VERT_NORM]+1);

else RoundY = 2^(5); NormY = 2^(5+1);

15.11.3 Memory Map and Registers

15.11.3.1 register descriptions

15.11.3.1.1 SUBSAM Memory map

SUBSAM base address: 1_B000h

Offset	Register	Width (In bits)	Access	Reset value
0h	(SS_SYS_CTRL)	32	RW	0000_0000h
4h	(SS_SYS_CTRL_SET)	32	RW	0000_0000h
8h	(SS_SYS_CTRL_CLR)	32	RW	0000_0000h
Ch	(SS_SYS_CTRL_TOG)	32	RW	0000_0000h
10h	(SS_DISPLAY)	32	RW	0000_0000h
14h	(SS_DISPLAY_SET)	32	RW	0000_0000h
18h	(SS_DISPLAY_CLR)	32	RW	0000_0000h
1Ch	(SS_DISPLAY_TOG)	32	RW	0000_0000h
20h	(SS_HSYNC)	32	RW	0000_0000h
24h	(SS_HSYNC_SET)	32	RW	0000_0000h
28h	(SS_HSYNC_CLR)	32	RW	0000_0000h
2Ch	(SS_HSYNC_TOG)	32	RW	0000_0000h
30h	(SS_VSYNC)	32	RW	0000_0000h
34h	(SS_VSYNC_SET)	32	RW	0000_0000h
38h	(SS_VSYNC_CLR)	32	RW	0000_0000h
3Ch	(SS_VSYNC_TOG)	32	RW	0000_0000h
40h	(SS_DE_ULC)	32	RW	0000_0000h
44h	(SS_DE_ULC_SET)	32	RW	0000_0000h
48h	(SS_DE_ULC_CLR)	32	RW	0000_0000h
4Ch	(SS_DE_ULC_TOG)	32	RW	0000_0000h

Table continues on the next page...

Offset	Register	Width (In bits)	Access	Reset value
50h	(SS_DE_LRC)	32	RW	0000_0000h
54h	(SS_DE_LRC_SET)	32	RW	0000_0000h
58h	(SS_DE_LRC_CLR)	32	RW	0000_0000h
5Ch	(SS_DE_LRC_TOG)	32	RW	0000_0000h
60h	(SS_MODE)	32	RW	0000_0000h
64h	(SS_MODE_SET)	32	RW	0000_0000h
68h	(SS_MODE_CLR)	32	RW	0000_0000h
6Ch	(SS_MODE_TOG)	32	RW	0000_0000h
70h	(SS_COEFF)	32	RW	0000_0000h
74h	(SS_COEFF_SET)	32	RW	0000_0000h
78h	(SS_COEFF_CLR)	32	RW	0000_0000h
7Ch	(SS_COEFF_TOG)	32	RW	0000_0000h
80h	(SS_CLIP_CB)	32	RW	0000_0000h
84h	(SS_CLIP_CB_SET)	32	RW	0000_0000h
88h	(SS_CLIP_CB_CLR)	32	RW	0000_0000h
8Ch	(SS_CLIP_CB_TOG)	32	RW	0000_0000h
90h	(SS_CLIP_CR)	32	RW	0000_0000h
94h	(SS_CLIP_CR_SET)	32	RW	0000_0000h
98h	(SS_CLIP_CR_CLR)	32	RW	0000_0000h
9Ch	(SS_CLIP_CR_TOG)	32	RW	0000_0000h
A0h	(SS_INTER_MODE)	32	RW	0000_0000h
A4h	(SS_INTER_MODE_SET)	32	RW	0000_0000h
A8h	(SS_INTER_MODE_CLR)	32	RW	0000_0000h
ACh	(SS_INTER_MODE_TOG)	32	RW	0000_0000h

15.11.3.1.2 (SS_SYS_CTRL)

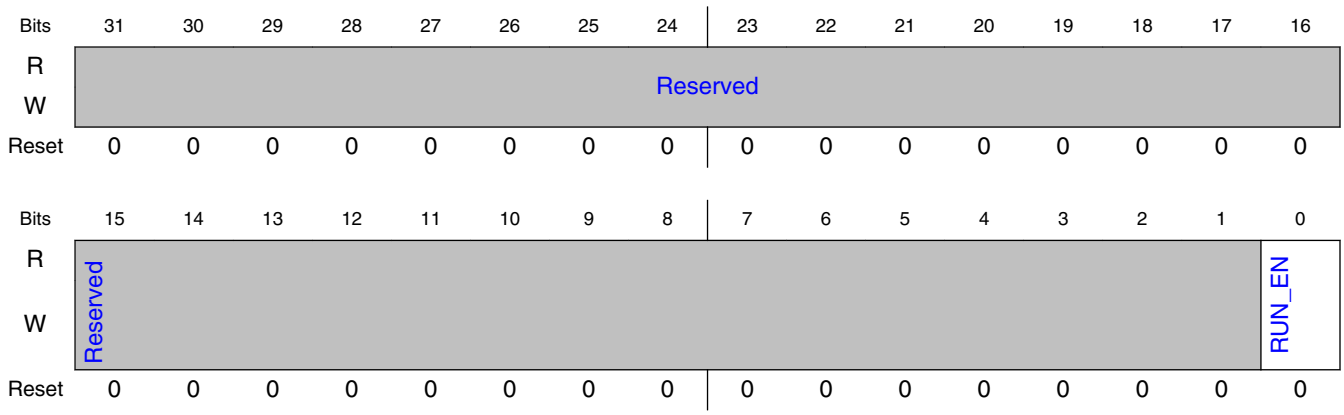
15.11.3.1.2.1 Offset

Register	Offset
SS_SYS_CTRL	0h

15.11.3.1.2.2 Function

The ss_sys_ctrl register contains all top level system control fields.

15.11.3.1.2.3 Diagram



15.11.3.1.2.4 Fields

Field	Function
31-1 —	Reserved
0 RUN_EN	Software sets this bit to indicate when the subsam module is ready to be enabled. Setting this bit from 1 to 0 causes a software reset.

15.11.3.1.3 (SS_SYS_CTRL_SET)

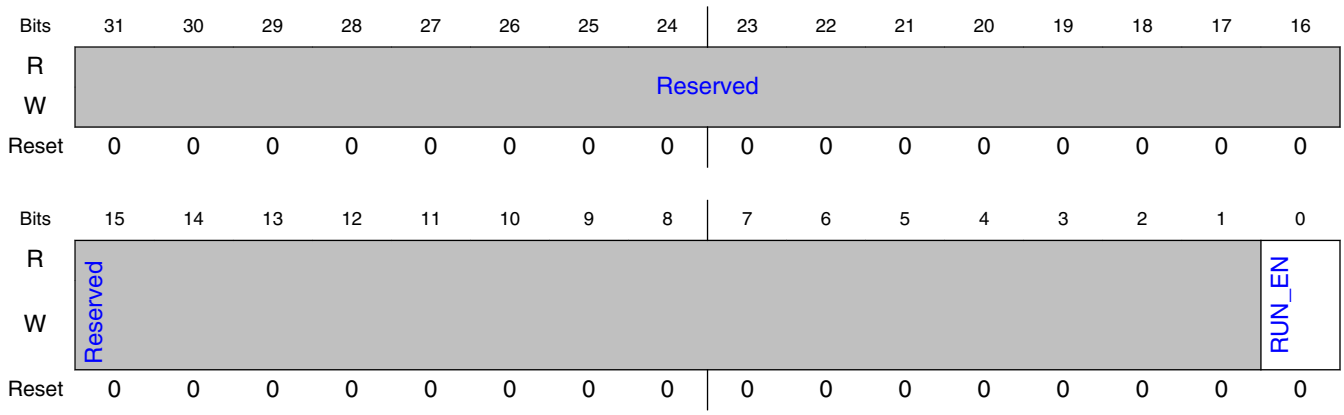
15.11.3.1.3.1 Offset

Register	Offset
SS_SYS_CTRL_SET	4h

15.11.3.1.3.2 Function

The ss_sys_ctrl register contains all top level system control fields.

15.11.3.1.3.3 Diagram



15.11.3.1.3.4 Fields

Field	Function
31-1 —	Reserved
0 RUN_EN	Software sets this bit to indicate when the subsam module is ready to be enabled. Setting this bit from 1 to 0 causes a software reset.

15.11.3.1.4 (SS_SYS_CTRL_CLR)

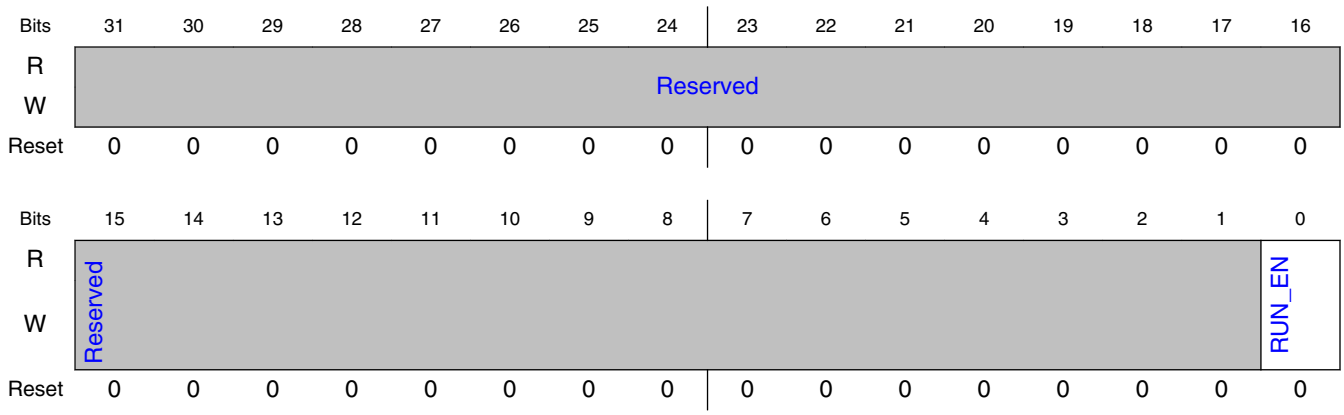
15.11.3.1.4.1 Offset

Register	Offset
SS_SYS_CTRL_CLR	8h

15.11.3.1.4.2 Function

The ss_sys_ctrl register contains all top level system control fields.

15.11.3.1.4.3 Diagram



15.11.3.1.4.4 Fields

Field	Function
31-1 —	Reserved
0 RUN_EN	Software sets this bit to indicate when the subsam module is ready to be enabled. Setting this bit from 1 to 0 causes a software reset.

15.11.3.1.5 (SS_SYS_CTRL_TOG)

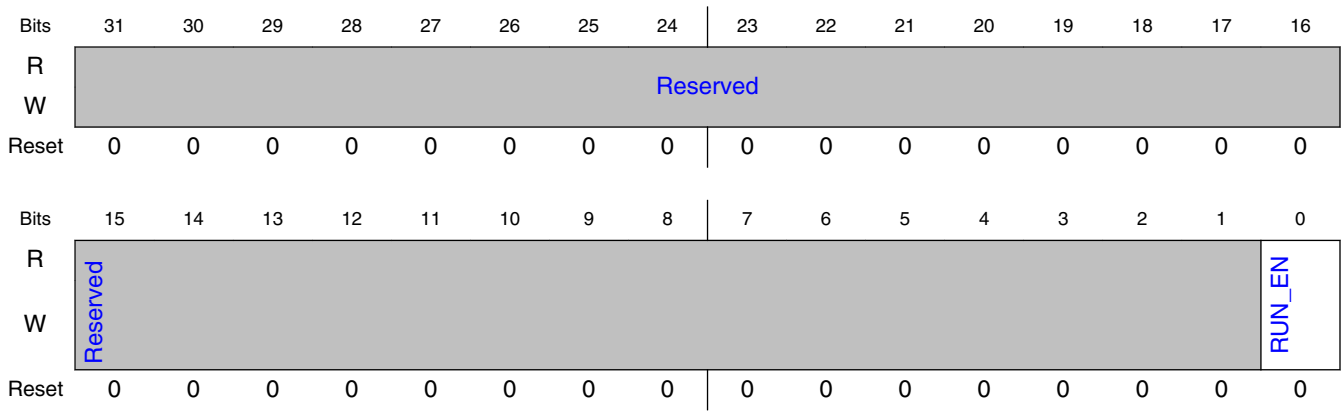
15.11.3.1.5.1 Offset

Register	Offset
SS_SYS_CTRL_TOG	Ch

15.11.3.1.5.2 Function

The ss_sys_ctrl register contains all top level system control fields.

15.11.3.1.5.3 Diagram



15.11.3.1.5.4 Fields

Field	Function
31-1 —	Reserved
0 RUN_EN	Software sets this bit to indicate when the subsam module is ready to be enabled. Setting this bit from 1 to 0 causes a software reset.

15.11.3.1.6 (SS_DISPLAY)

15.11.3.1.6.1 Offset

Register	Offset
SS_DISPLAY	10h

15.11.3.1.6.2 Function

```
The csr_ss_display register defines the entire size of the display timing window.
  SS_DISPLAY_LRX_Y = VACTIVE+VBLANK-1
  SS_DISPLAY_LRC_X = HACTIVE+HBLANK-1
```

15.11.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			LRC_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			LRC_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.6.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Holds the lower right corner y coordinate. The vertical count reg should conditionally reset to zero when it reaches this value and v_count = DTG_LRC_Y
15-13 —	Reserved
12-0 LRC_X	Holds the lower right corner x coordinate. The horizontal count reg should reset to zero when it reaches this value.

15.11.3.1.7 (SS_DISPLAY_SET)

15.11.3.1.7.1 Offset

Register	Offset
SS_DISPLAY_SET	14h

15.11.3.1.7.2 Function

```
The csr_ss_display register defines the entire size of the display timing window.
  SS_DISPLAY_LRX_Y = VACTIVE+VBLANK-1
  SS_DISPLAY_LRC_X = HACTIVE+HBLANK-1
```

15.11.3.1.7.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	Reserved			LRC_Y												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	Reserved			LRC_X												
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.7.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Holds the lower right corner y coordinate. The vertical count reg should conditionally reset to zero when it reaches this value and v_count = DTG_LRC_Y
15-13 —	Reserved
12-0 LRC_X	Holds the lower right corner x coordinate. The horizontal count reg should reset to zero when it reaches this value.

15.11.3.1.8 (SS_DISPLAY_CLR)

15.11.3.1.8.1 Offset

Register	Offset
SS_DISPLAY_CLR	18h

15.11.3.1.8.2 Function

```
The csr_ss_display register defines the entire size of the display timing window.
  SS_DISPLAY_LRX_Y = VACTIVE+VBLANK-1
  SS_DISPLAY_LRC_X = HACTIVE+HBLANK-1
```

15.11.3.1.8.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LRC_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			LRC_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.8.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Holds the lower right corner y coordinate. The vertical count reg should conditionally reset to zero when it reaches this value and v_count = DTG_LRC_Y
15-13 —	Reserved
12-0 LRC_X	Holds the lower right corner x coordinate. The horizontal count reg should reset to zero when it reaches this value.

15.11.3.1.9 (SS_DISPLAY_TOG)

15.11.3.1.9.1 Offset

Register	Offset
SS_DISPLAY_TOG	1Ch

15.11.3.1.9.2 Function

```
The csr_ss_display register defines the entire size of the display timing window.
  SS_DISPLAY_LRX_Y = VACTIVE+VBLANK-1
  SS_DISPLAY_LRC_X = HACTIVE+HBLANK-1
```

15.11.3.1.9.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LRC_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			LRC_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.9.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Holds the lower right corner y coordinate. The vertical count reg should conditionally reset to zero when it reaches this value and v_count = DTG_LRC_Y
15-13 —	Reserved
12-0 LRC_X	Holds the lower right corner x coordinate. The horizontal count reg should reset to zero when it reaches this value.

15.11.3.1.10 (SS_HSYNC)

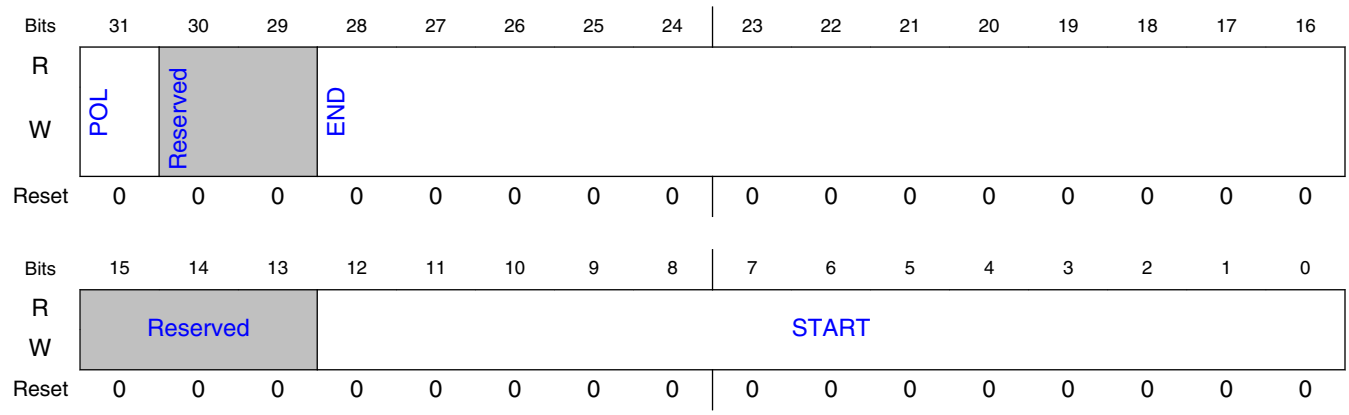
15.11.3.1.10.1 Offset

Register	Offset
SS_HSYNC	20h

15.11.3.1.10.2 Function

Defines when horizontal sync signal goes high/low.
There are two X coordinates defined in this register.
Horizontal sync width determined by END - START.

15.11.3.1.10.3 Diagram



15.11.3.1.10.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	horizontal sync is de-asserted when horizontal count == END
15-13 —	Reserved
12-0 START	horizontal sync is asserted when horizontal count == START

15.11.3.1.11 (SS_HSYNC_SET)

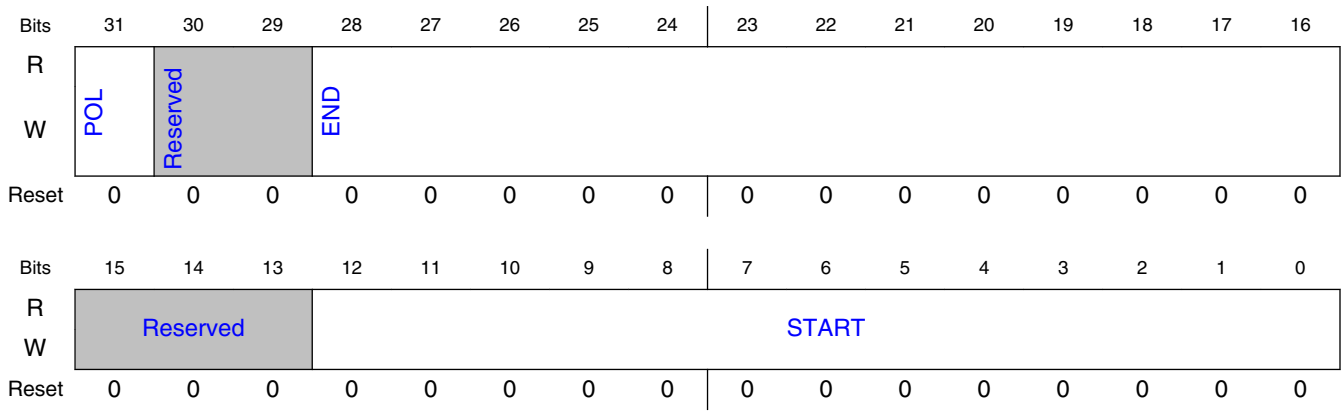
15.11.3.1.11.1 Offset

Register	Offset
SS_HSYNC_SET	24h

15.11.3.1.11.2 Function

```
Defines when horizontal sync signal goes high/low.
There are two X coordinates defined in this register.
Horizontal sync width determined by END - START.
```

15.11.3.1.11.3 Diagram



15.11.3.1.11.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	horizontal sync is de-asserted when horizontal count == END
15-13 —	Reserved
12-0 START	horizontal sync is asserted when horizontal count == START

15.11.3.1.12 (SS_HSYNC_CLR)

15.11.3.1.12.1 Offset

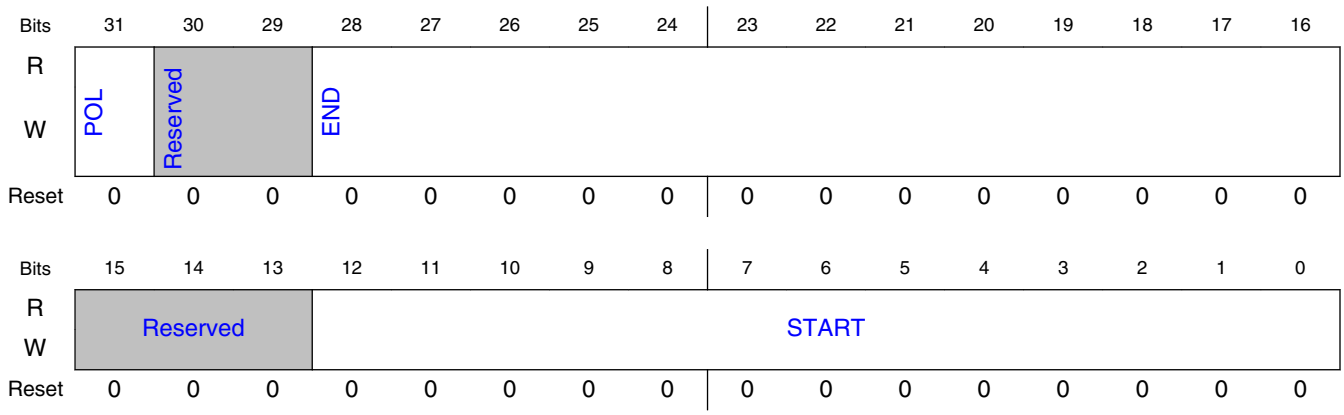
Register	Offset
SS_HSYNC_CLR	28h

15.11.3.1.12.2 Function

Defines when horizontal sync signal goes high/low.

```
There are two X coordinates defined in this register.
Horizontal sync width determined by END - START.
```

15.11.3.1.12.3 Diagram



15.11.3.1.12.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	horizontal sync is de-asserted when horizontal count == END
15-13 —	Reserved
12-0 START	horizontal sync is asserted when horizontal count == START

15.11.3.1.13 (SS_HSYNC_TOG)

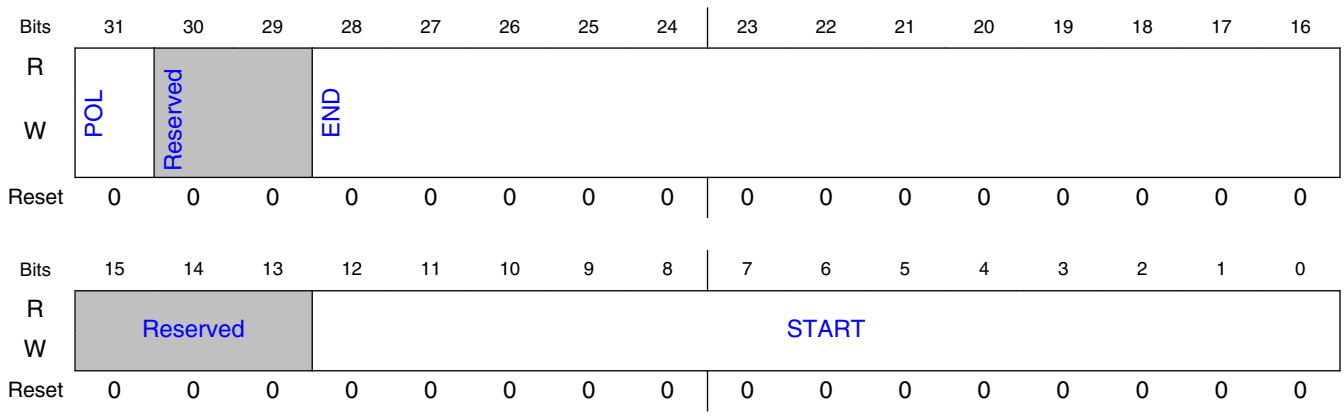
15.11.3.1.13.1 Offset

Register	Offset
SS_HSYNC_TOG	2Ch

15.11.3.1.13.2 Function

```
Defines when horizontal sync signal goes high/low.
There are two X coordinates defined in this register.
Horizontal sync width determined by END - START.
```

15.11.3.1.13.3 Diagram



15.11.3.1.13.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	horizontal sync is de-asserted when horizontal count == END
15-13 —	Reserved
12-0 START	horizontal sync is asserted when horizontal count == START

15.11.3.1.14 (SS_VSYNC)

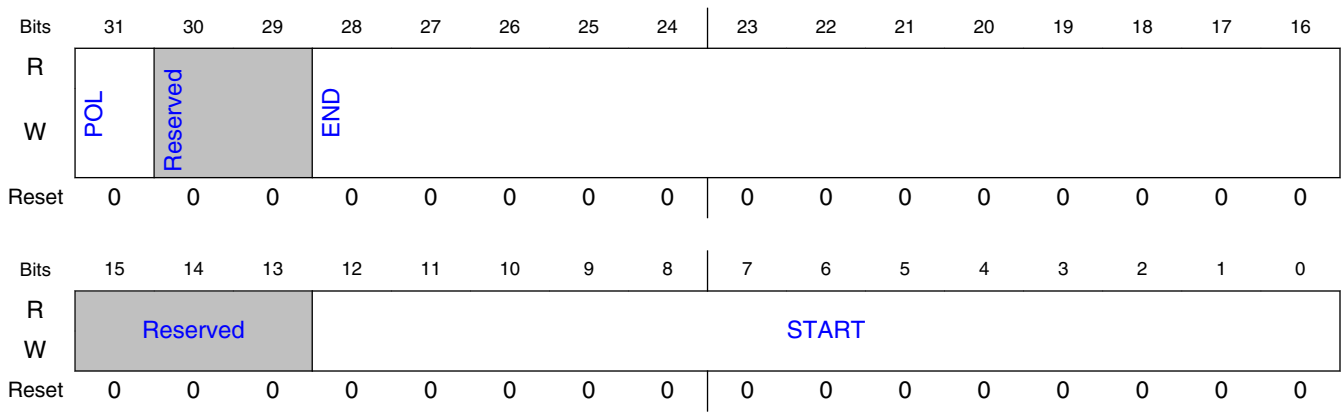
15.11.3.1.14.1 Offset

Register	Offset
SS_VSYNC	30h

15.11.3.1.14.2 Function

```
Defines Y coordinate when vertical sync signal goes high/low.
Horizontal sync width determined by END - START.
```

15.11.3.1.14.3 Diagram



15.11.3.1.14.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	When v_count = END h_count = CSR_SS_LRC_X vsync goes inactive
15-13 —	Reserved
12-0 START	When v_count = START and h_count = CSR_SS_LRC_X then vsync goes active

15.11.3.1.15 (SS_VSYNC_SET)

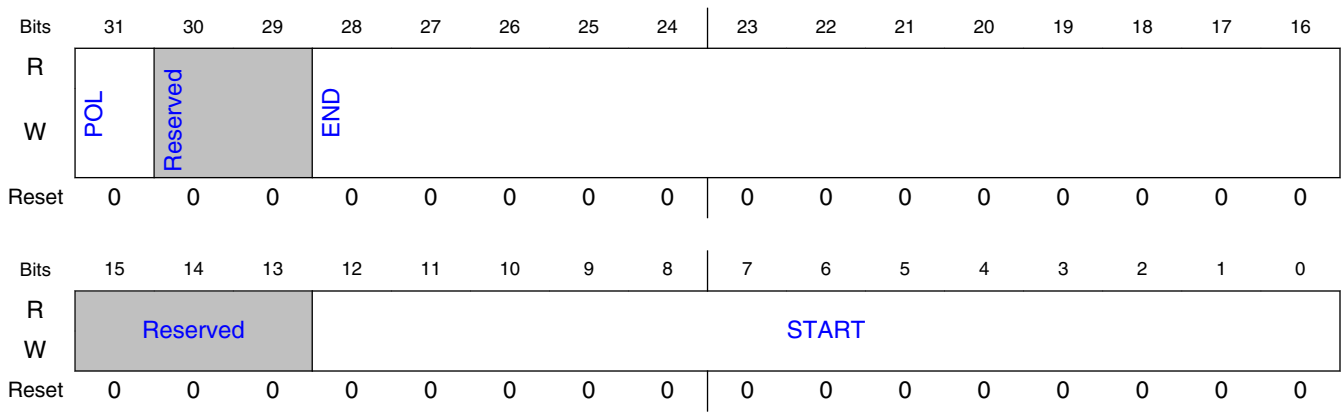
15.11.3.1.15.1 Offset

Register	Offset
SS_VSYNC_SET	34h

15.11.3.1.15.2 Function

```
Defines Y coordinate when vertical sync signal goes high/low.
Horizontal sync width determined by END - START.
```

15.11.3.1.15.3 Diagram



15.11.3.1.15.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	When v_count = END h_count = CSR_SS_LRC_X vsync goes inactive
15-13 —	Reserved
12-0 START	When v_count = START and h_count = CSR_SS_LRC_X then vsync goes active

15.11.3.1.16 (SS_VSYNC_CLR)

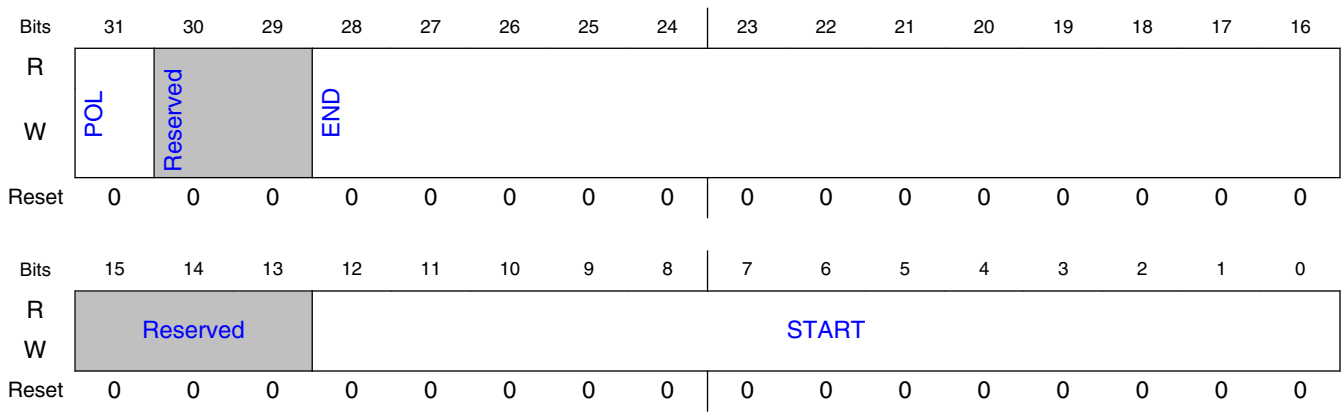
15.11.3.1.16.1 Offset

Register	Offset
SS_VSYNC_CLR	38h

15.11.3.1.16.2 Function

```
Defines Y coordinate when vertical sync signal goes high/low.
Horizontal sync width determined by END - START.
```

15.11.3.1.16.3 Diagram



15.11.3.1.16.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	When v_count = END h_count = CSR_SS_LRC_X vsync goes inactive
15-13 —	Reserved
12-0 START	When v_count = START and h_count = CSR_SS_LRC_X then vsync goes active

15.11.3.1.17 (SS_VSYNC_TOG)

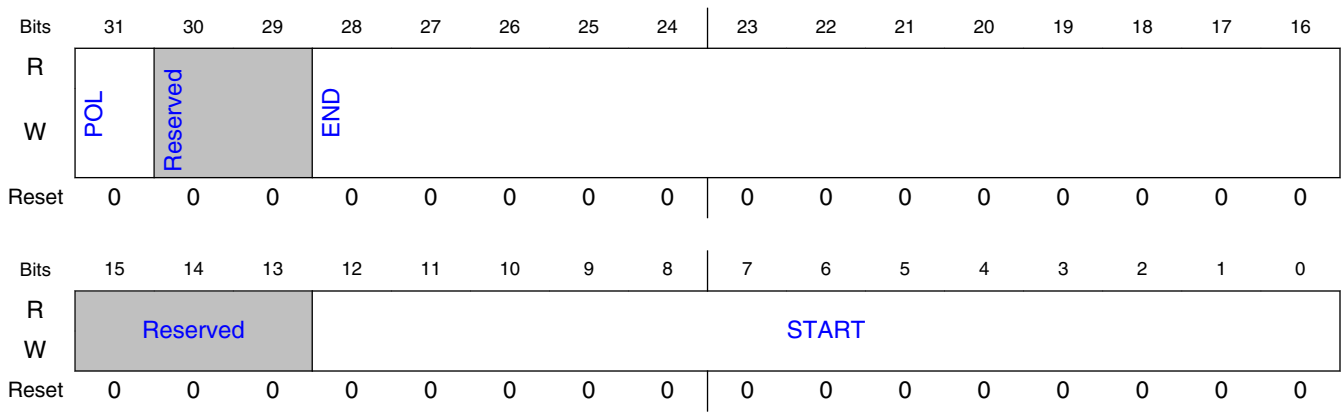
15.11.3.1.17.1 Offset

Register	Offset
SS_VSYNC_TOG	3Ch

15.11.3.1.17.2 Function

```
Defines Y coordinate when vertical sync signal goes high/low.
Horizontal sync width determined by END - START.
```

15.11.3.1.17.3 Diagram



15.11.3.1.17.4 Fields

Field	Function
31 POL	Polarity of sync signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 END	When v_count = END h_count = CSR_SS_LRC_X vsync goes inactive
15-13 —	Reserved
12-0 START	When v_count = START and h_count = CSR_SS_LRC_X then vsync goes active

15.11.3.1.18 (SS_DE_ULC)

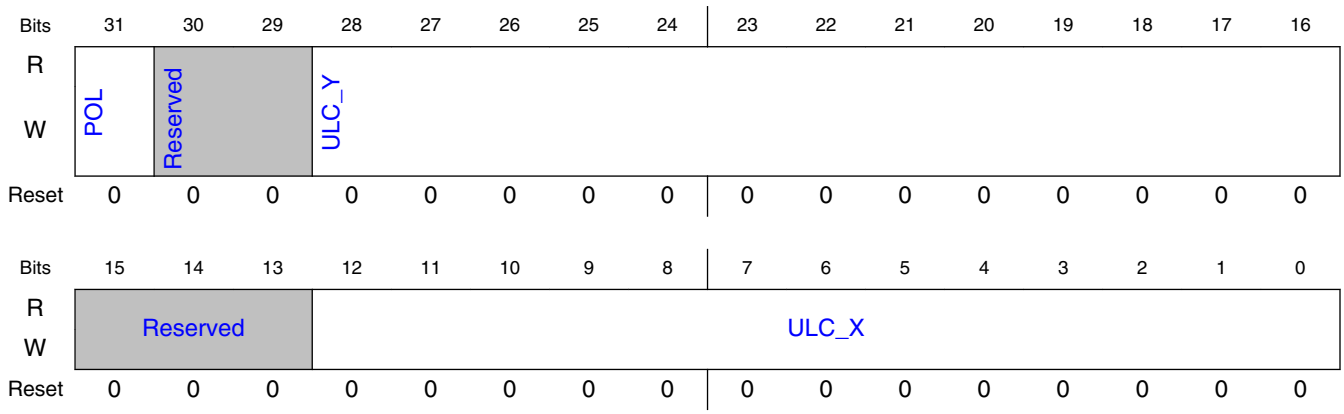
15.11.3.1.18.1 Offset

Register	Offset
SS_DE_ULC	40h

15.11.3.1.18.2 Function

```
Upper left coordinate of frame that defines when DE goes active.
Both X and Y coordinates are programmable as ULC_X and ULC_Y respectively.
Signal transitions occur when count == boundary.
```

15.11.3.1.18.3 Diagram



15.11.3.1.18.4 Fields

Field	Function
31 POL	Polarity of DE signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 ULC_Y	Vertical position in the display that determines when the DE goes active.
15-13 —	Reserved
12-0 ULC_X	Horizontal position in the scan line that determines when DE goes active.

15.11.3.1.19 (SS_DE_ULC_SET)

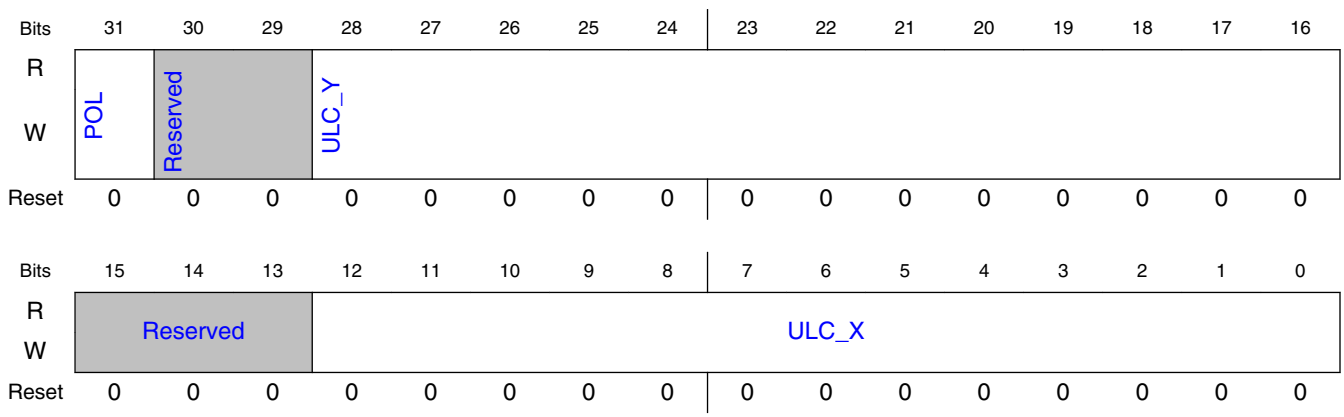
15.11.3.1.19.1 Offset

Register	Offset
SS_DE_ULC_SET	44h

15.11.3.1.19.2 Function

```
Upper left coordinate of frame that defines when DE goes active.
Both X and Y coordinates are programmable as ULC_X and ULC_Y respectively.
Signal transitions occur when count == boundary.
```

15.11.3.1.19.3 Diagram



15.11.3.1.19.4 Fields

Field	Function
31 POL	Polarity of DE signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 ULC_Y	Vertical position in the display that determines when the DE goes active.
15-13 —	Reserved
12-0 ULC_X	Horizontal position in the scan line that determines when DE goes active.

15.11.3.1.20 (SS_DE_ULC_CLR)

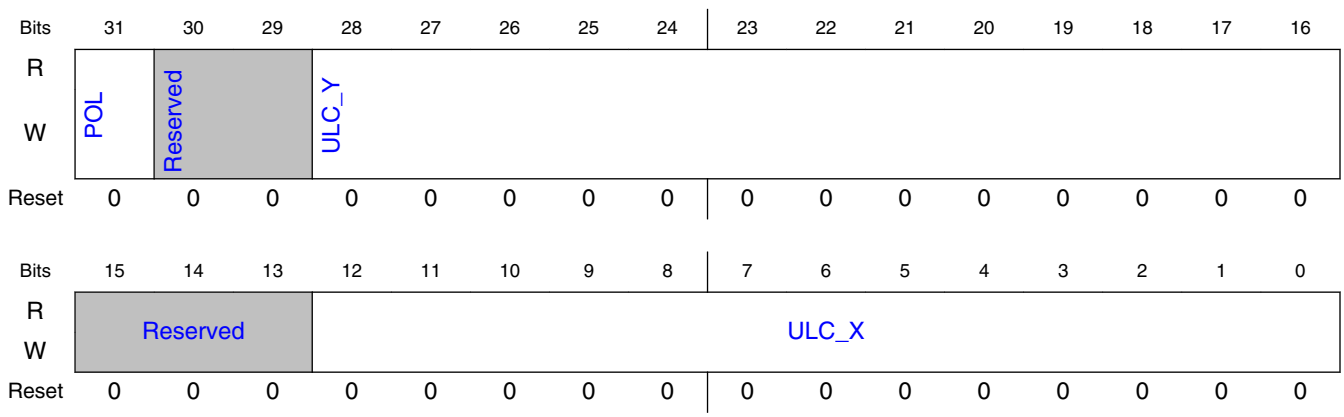
15.11.3.1.20.1 Offset

Register	Offset
SS_DE_ULC_CLR	48h

15.11.3.1.20.2 Function

```
Upper left coordinate of frame that defines when DE goes active.
Both X and Y coordinates are programmable as ULC_X and ULC_Y respectively.
Signal transitions occur when count == boundary.
```

15.11.3.1.20.3 Diagram



15.11.3.1.20.4 Fields

Field	Function
31 POL	Polarity of DE signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 ULC_Y	Vertical position in the display that determines when the DE goes active.
15-13 —	Reserved
12-0 ULC_X	Horizontal position in the scan line that determines when DE goes active.

15.11.3.1.21 (SS_DE_ULC_TOG)

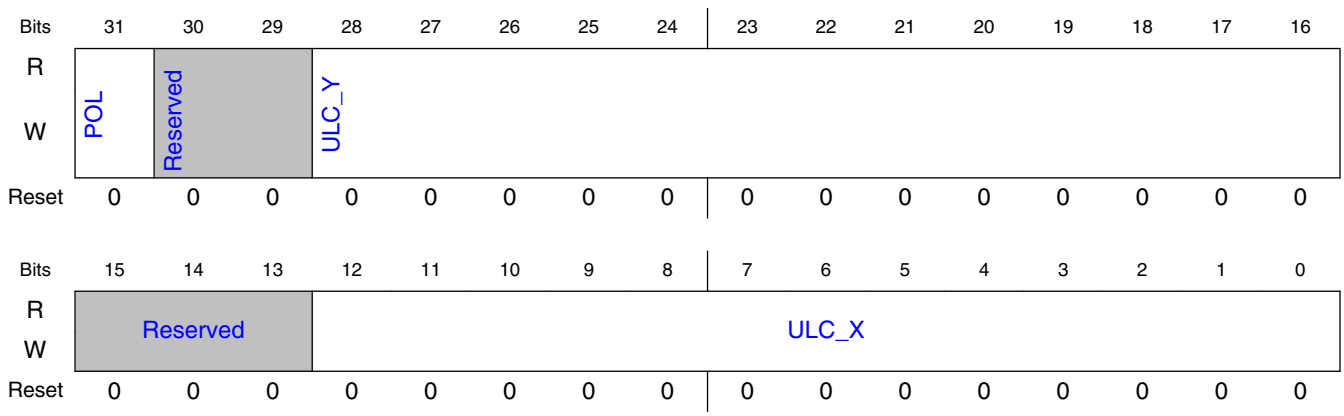
15.11.3.1.21.1 Offset

Register	Offset
SS_DE_ULC_TOG	4Ch

15.11.3.1.21.2 Function

```
Upper left coordinate of frame that defines when DE goes active.
Both X and Y coordinates are programmable as ULC_X and ULC_Y respectively.
Signal transitions occur when count == boundary.
```

15.11.3.1.21.3 Diagram



15.11.3.1.21.4 Fields

Field	Function
31 POL	Polarity of DE signal. 0 == active low ; 1 == active_high
30-29 —	Reserved
28-16 ULC_Y	Vertical position in the display that determines when the DE goes active.
15-13 —	Reserved
12-0 ULC_X	Horizontal position in the scan line that determines when DE goes active.

15.11.3.1.22 (SS_DE_LRC)

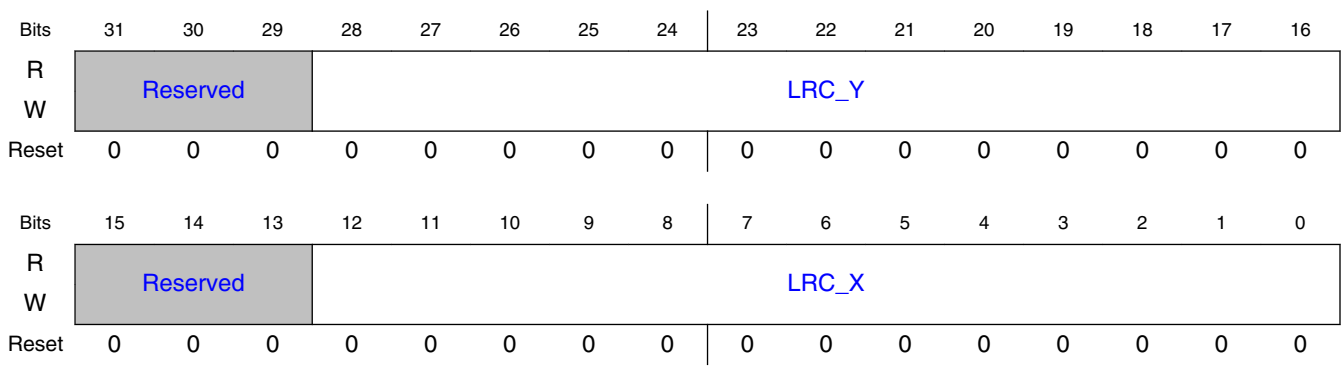
15.11.3.1.22.1 Offset

Register	Offset
SS_DE_LRC	50h

15.11.3.1.22.2 Function

Lower right coordinate that defines when DE goes NOT active.
Both X and Y coordinates are programmable as LRC_X and LRC_Y respectively.

15.11.3.1.22.3 Diagram



15.11.3.1.22.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Vertical position in the display that determines when the DE goes inactive.
15-13 —	Reserved
12-0 LRC_X	Horizontal position in the scan line that determines when DE goes inactive.

15.11.3.1.23 (SS_DE_LRC_SET)

15.11.3.1.23.1 Offset

Register	Offset
SS_DE_LRC_SET	54h

15.11.3.1.23.2 Function

```
Lower right coordinate that defines when DE goes NOT active.
Both X and Y coordinates are programmable as LRC_X and LRC_Y respectively.
```

15.11.3.1.23.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LRC_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			LRC_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.23.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Vertical position in the display that determines when the DE goes inactive.
15-13 —	Reserved
12-0 LRC_X	Horizontal position in the scan line that determines when DE goes inactive.

15.11.3.1.24 (SS_DE_LRC_CLR)

15.11.3.1.24.1 Offset

Register	Offset
SS_DE_LRC_CLR	58h

15.11.3.1.24.2 Function

```
Lower right coordinate that defines when DE goes NOT active.
Both X and Y coordinates are programmable as LRC_X and LRC_Y respectively.
```

15.11.3.1.24.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LRC_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			LRC_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.24.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Vertical position in the display that determines when the DE goes inactive.
15-13 —	Reserved
12-0 LRC_X	Horizontal position in the scan line that determines when DE goes inactive.

15.11.3.1.25 (SS_DE_LRC_TOG)

15.11.3.1.25.1 Offset

Register	Offset
SS_DE_LRC_TOG	5Ch

15.11.3.1.25.2 Function

```
Lower right coordinate that defines when DE goes NOT active.
Both X and Y coordinates are programmable as LRC_X and LRC_Y respectively.
```

15.11.3.1.25.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved			LRC_Y												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved			LRC_X												
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.25.4 Fields

Field	Function
31-29 —	Reserved
28-16 LRC_Y	Vertical position in the display that determines when the DE goes inactive.
15-13 —	Reserved
12-0 LRC_X	Horizontal position in the scan line that determines when DE goes inactive.

15.11.3.1.26 (SS_MODE)

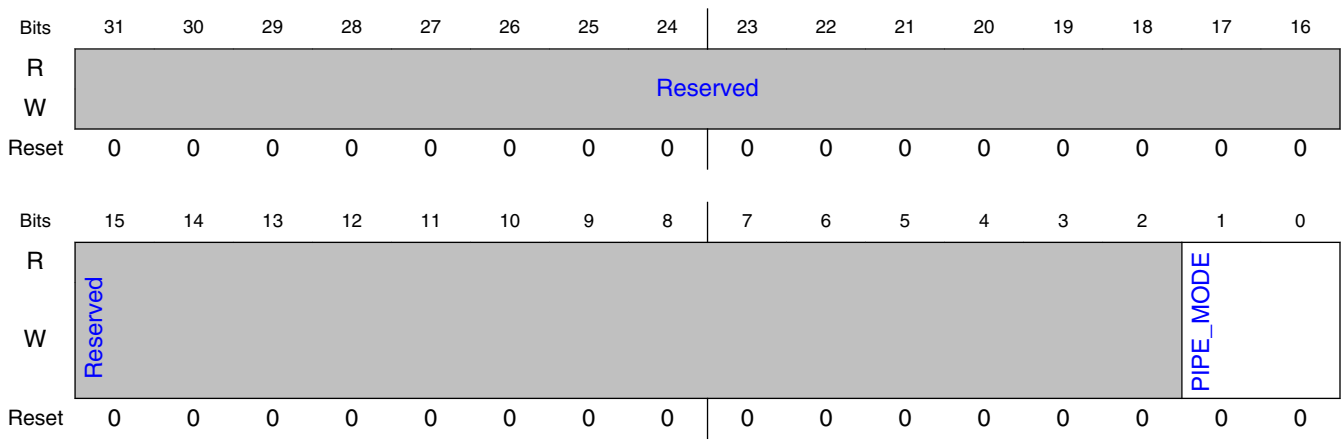
15.11.3.1.26.1 Offset

Register	Offset
SS_MODE	60h

15.11.3.1.26.2 Function

```
This register is used to select the type of chroma sub-sampling used.
Three options are available: bypass(444), 422, and 420.
```

15.11.3.1.26.3 Diagram



15.11.3.1.26.4 Fields

Field	Function
31-2 —	Reserved
1-0 PIPE_MODE	Sets the sub-sampler mode. 00: bypass 01: 422 subsample 10: 420 subsample 11: bypass

15.11.3.1.27 (SS_MODE_SET)

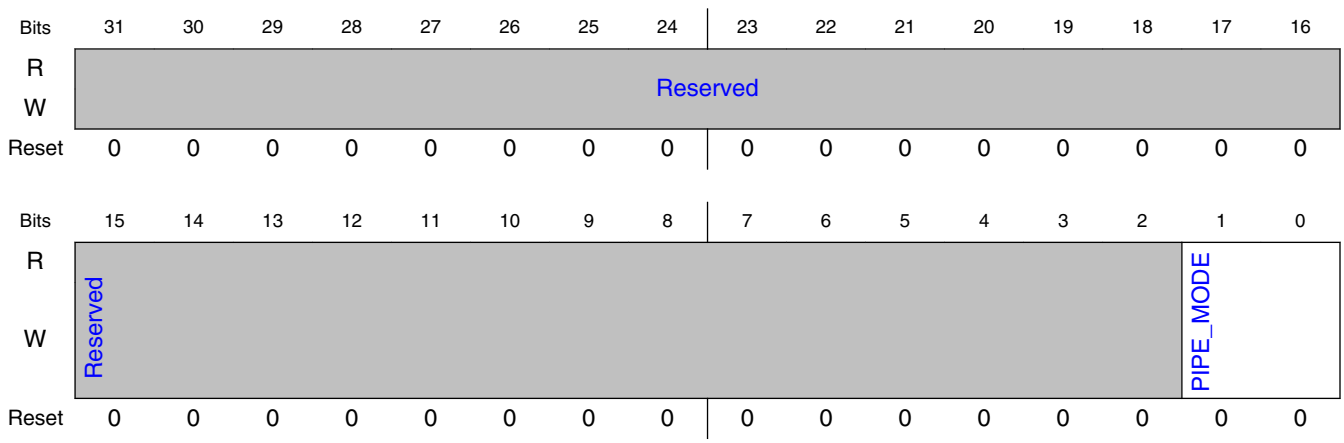
15.11.3.1.27.1 Offset

Register	Offset
SS_MODE_SET	64h

15.11.3.1.27.2 Function

```
This register is used to select the type of chroma sub-sampling used.
Three options are available: bypass(444), 422, and 420.
```

15.11.3.1.27.3 Diagram



15.11.3.1.27.4 Fields

Field	Function
31-2 —	Reserved
1-0 PIPE_MODE	Sets the sub-sampler mode. 00: bypass 01: 422 subsample 10: 420 subsample 11: bypass

15.11.3.1.28 (SS_MODE_CLR)

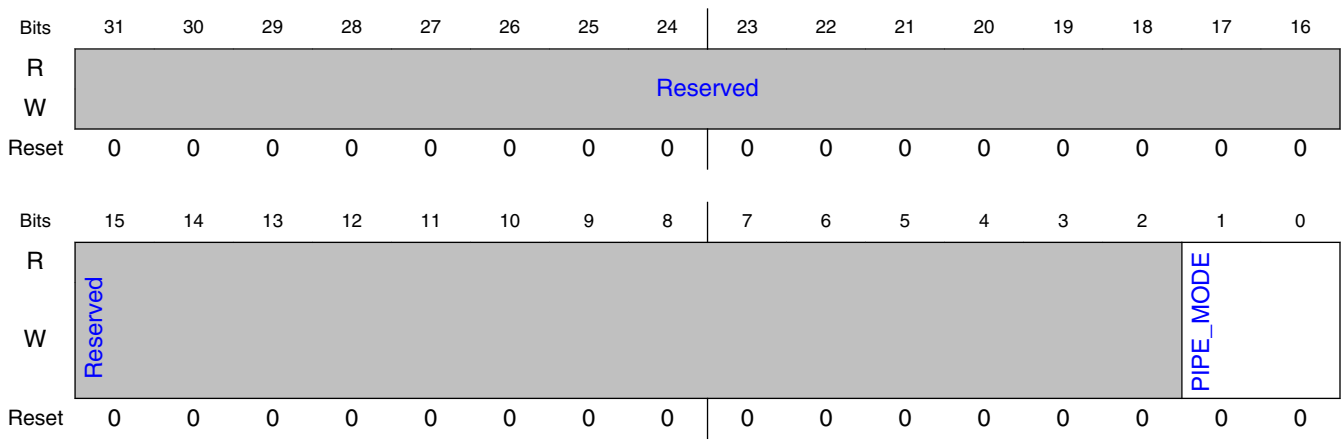
15.11.3.1.28.1 Offset

Register	Offset
SS_MODE_CLR	68h

15.11.3.1.28.2 Function

```
This register is used to select the type of chroma sub-sampling used.
Three options are available: bypass(444), 422, and 420.
```

15.11.3.1.28.3 Diagram



15.11.3.1.28.4 Fields

Field	Function
31-2 —	Reserved
1-0 PIPE_MODE	Sets the sub-sampler mode. 00: bypass 01: 422 subsample 10: 420 subsample 11: bypass

15.11.3.1.29 (SS_MODE_TOG)

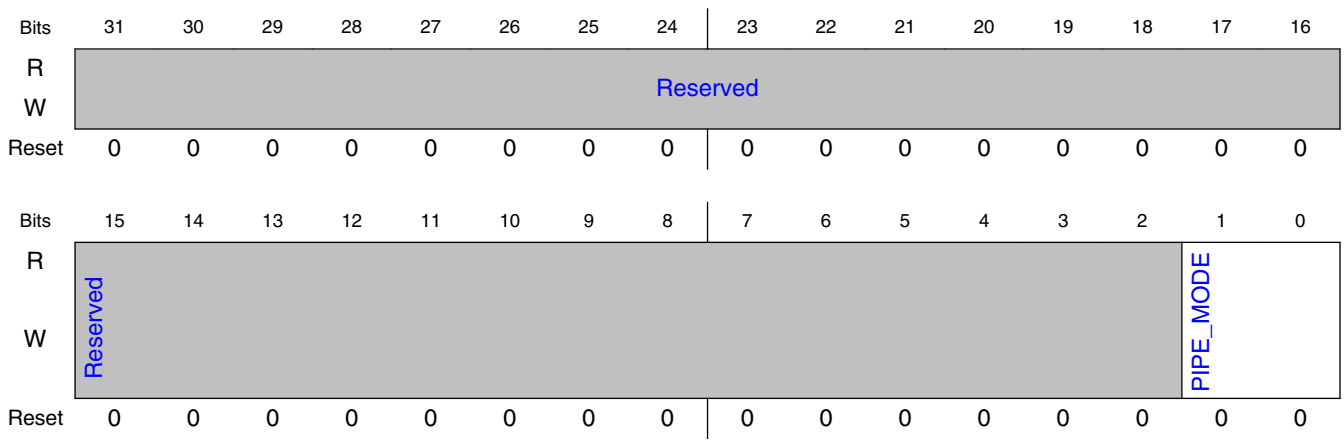
15.11.3.1.29.1 Offset

Register	Offset
SS_MODE_TOG	6Ch

15.11.3.1.29.2 Function

```
This register is used to select the type of chroma sub-sampling used.
Three options are available: bypass(444), 422, and 420.
```

15.11.3.1.29.3 Diagram



15.11.3.1.29.4 Fields

Field	Function
31-2 —	Reserved
1-0 PIPE_MODE	Sets the sub-sampler mode. 00: bypass 01: 422 subsample 10: 420 subsample 11: bypass

15.11.3.1.30 (SS_COEFF)

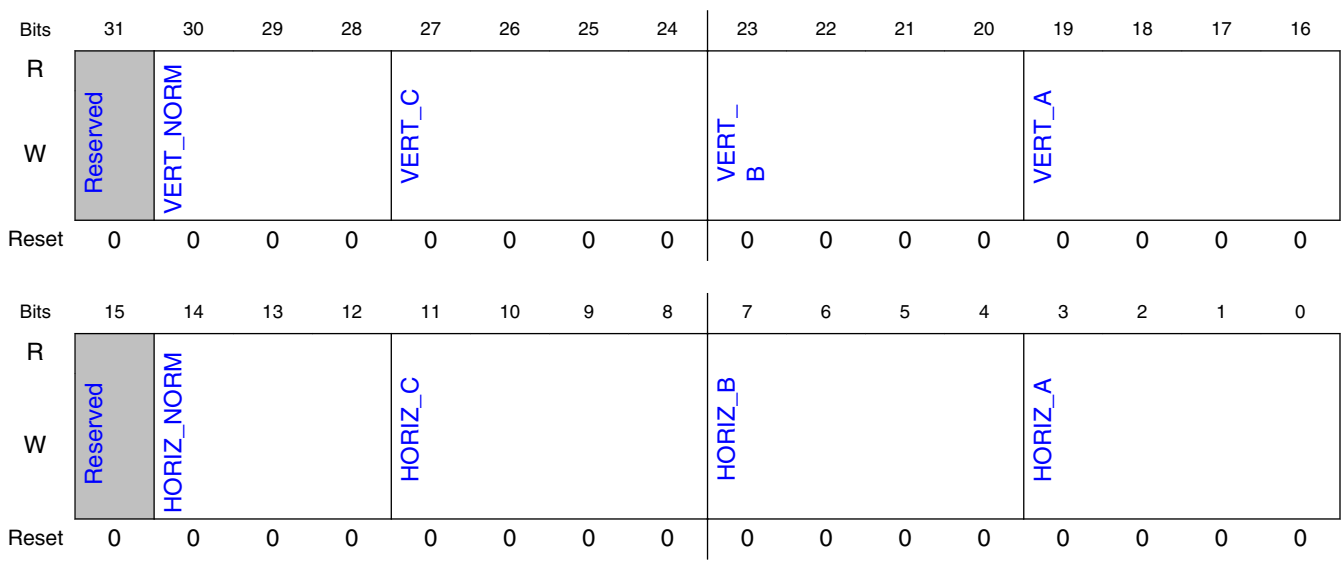
15.11.3.1.30.1 Offset

Register	Offset
SS_COEFF	70h

15.11.3.1.30.2 Function

```
Holds coefficients for both vertical + horizontal filters.
Three coefficients used for 3 tap fir filter.
HORIZ_A, HORIZ_B, HORIZ_C is used for horizontal filter coefficients.
VERT_A, VERT_B, VERT_C is used for vertical filter coefficients.
In addition, the horizontal and vertical filters have a normalization factor called
HORIZ_NORM and VERT_NORM respectively.
A general formula for these filters is:
   pixel_out = (A*pixel[x+1]+B*pixel[x]+C*pixel[x-1]+Round)/Normalize
The values Round and Normalize are selected based on the values of HORIZ_NORM and VERT_NORM.
The mapping can be described by the following equations:
   if (NORM = 0)       Round = 0; Normalize = 1;
   else if (NORM < 5)  Round = 2^(NORM); Normalize = 2^(NORM+1);
   else                Round = 2^(5); Normalize = 2^(5+1);
```

15.11.3.1.30.3 Diagram



15.11.3.1.30.4 Fields

Field	Function
31 —	Reserved
30-28	Sets the 3 bit vertical filter normalization factor.

Table continues on the next page...

Field	Function
VERT_NORM	
27-24 VERT_C	Sets the 4 bit vertical C coefficient.
23-20 VERT_B	Sets the 4 bit vertical B coefficient.
19-16 VERT_A	Sets the 4 bit vertical A coefficient.
15 —	Reserved
14-12 HORIZ_NORM	Sets the 3 bit horizontal filter normalization factor.
11-8 HORIZ_C	Sets the 4 bit horizontal C coefficient.
7-4 HORIZ_B	Sets the 4 bit horizontal B coefficient.
3-0 HORIZ_A	Sets the 4 bit horizontal A coefficient.

15.11.3.1.31 (SS_COEFF_SET)

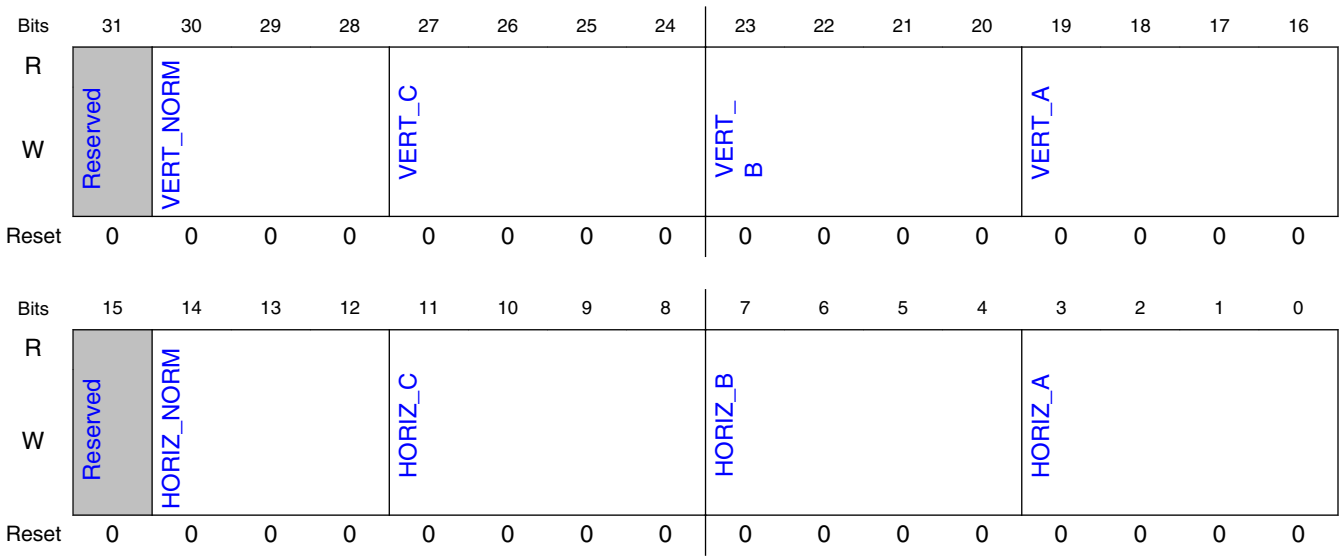
15.11.3.1.31.1 Offset

Register	Offset
SS_COEFF_SET	74h

15.11.3.1.31.2 Function

```
Holds coefficients for both vertical + horizontal filters.
Three coefficients used for 3 tap fir filter.
HORIZ_A, HORIZ_B, HORIZ_C is used for horizontal filter coefficients.
VERT_A, VERT_B, VERT_C is used for vertical filter coefficients.
In addition, the horizontal and vertical filters have a normalization factor called
HORIZ_NORM and VERT_NORM respectively.
A general formula for these filters is:
   pixel_out = (A*pixel[x+1]+B*pixel[x]+C*pixel[x-1]+Round)/Normalize
The values Round and Normalize are selected based on the values of HORIZ_NORM and VERT_NORM.
The mapping can be described by the following equations:
   if (NORM = 0)      Round = 0; Normalize = 1;
   else if (NORM < 5) Round = 2^(NORM); Normalize = 2^(NORM+1);
   else               Round = 2^(5); Normalize = 2^(5+1);
```

15.11.3.1.31.3 Diagram



15.11.3.1.31.4 Fields

Field	Function
31 —	Reserved
30-28 VERT_NORM	Sets the 3 bit vertical filter normalization factor.
27-24 VERT_C	Sets the 4 bit vertical C coefficient.
23-20 VERT_B	Sets the 4 bit vertical B coefficient.
19-16 VERT_A	Sets the 4 bit vertical A coefficient.
15 —	Reserved
14-12 HORIZ_NORM	Sets the 3 bit horizontal filter normalization factor.
11-8 HORIZ_C	Sets the 4 bit horizontal C coefficient.
7-4 HORIZ_B	Sets the 4 bit horizontal B coefficient.
3-0 HORIZ_A	Sets the 4 bit horizontal A coefficient.

15.11.3.1.32 (SS_COEFF_CLR)

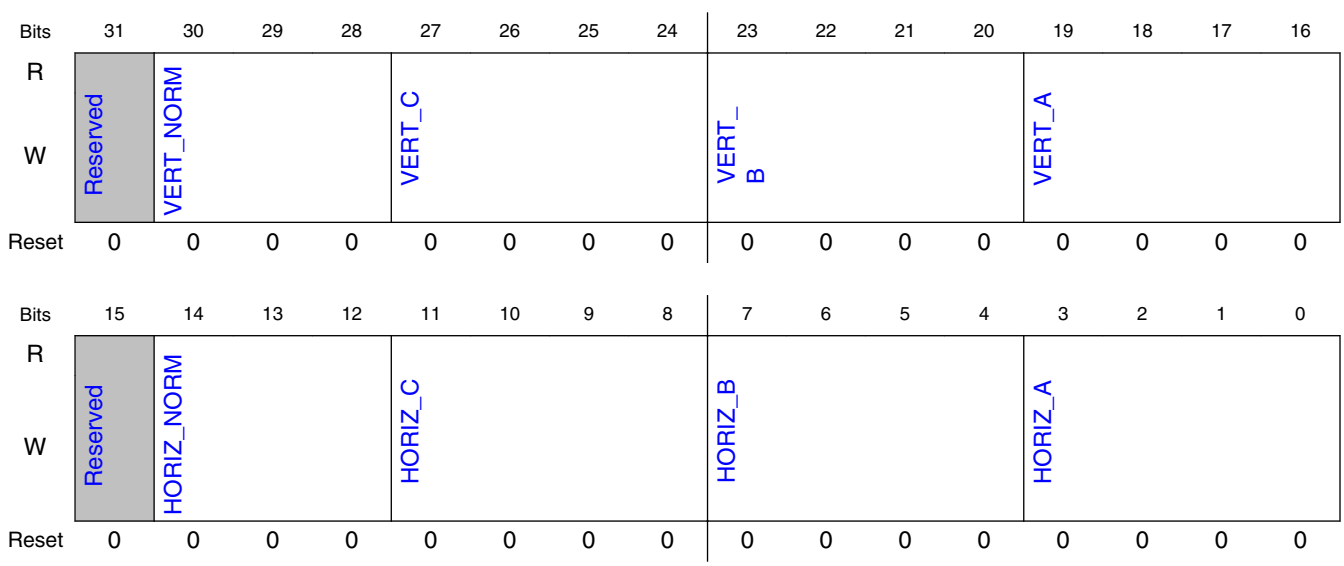
15.11.3.1.32.1 Offset

Register	Offset
SS_COEFF_CLR	78h

15.11.3.1.32.2 Function

```
Holds coefficients for both vertical + horizontal filters.
Three coefficients used for 3 tap fir filter.
HORIZ_A, HORIZ_B, HORIZ_C is used for horizontal filter coefficients.
VERT_A, VERT_B, VERT_C is used for vertical filter coefficients.
In addition, the horizontal and vertical filters have a normalization factor called
HORIZ_NORM and VERT_NORM respectively.
A general formula for these filters is:
   pixel_out = (A*pixel[x+1]+B*pixel[x]+C*pixel[x-1]+Round)/Normalize
The values Round and Normalize are selected based on the values of HORIZ_NORM and VERT_NORM.
The mapping can be described by the following equations:
   if (NORM = 0)       Round = 0; Normalize = 1;
   else if (NORM < 5)  Round = 2^(NORM); Normalize = 2^(NORM+1);
   else                Round = 2^(5); Normalize = 2^(5+1);
```

15.11.3.1.32.3 Diagram



15.11.3.1.32.4 Fields

Field	Function
31 —	Reserved
30-28 VERT_NORM	Sets the 3 bit vertical filter normalization factor.
27-24 VERT_C	Sets the 4 bit vertical C coefficient.
23-20 VERT_B	Sets the 4 bit vertical B coefficient.
19-16 VERT_A	Sets the 4 bit vertical A coefficient.
15 —	Reserved
14-12 HORIZ_NORM	Sets the 3 bit horizontal filter normalization factor.
11-8 HORIZ_C	Sets the 4 bit horizontal C coefficient.
7-4 HORIZ_B	Sets the 4 bit horizontal B coefficient.
3-0 HORIZ_A	Sets the 4 bit horizontal A coefficient.

15.11.3.1.33 (SS_COEFF_TOG)

15.11.3.1.33.1 Offset

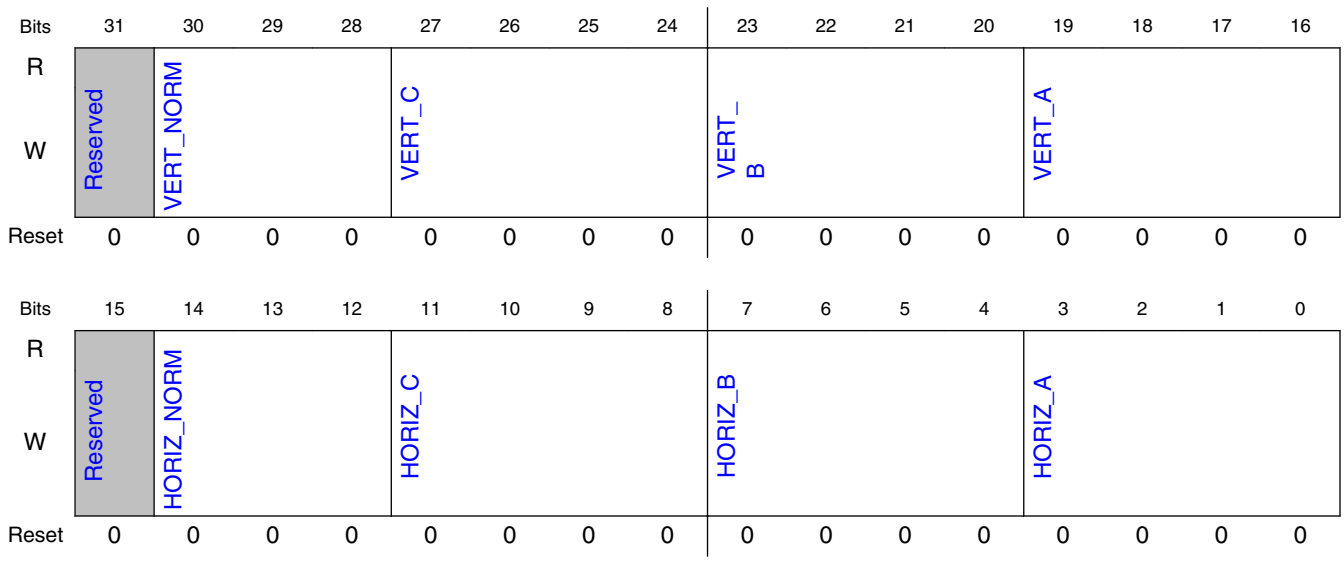
Register	Offset
SS_COEFF_TOG	7Ch

15.11.3.1.33.2 Function

```
Holds coefficients for both vertical + horizontal filters.
Three coefficients used for 3 tap fir filter.
HORIZ_A, HORIZ_B, HORIZ_C is used for horizontal filter coefficients.
VERT_A, VERT_B, VERT_C is used for vertical filter coefficients.
In addition, the horizontal and vertical filters have a normalization factor called
HORIZ_NORM and VERT_NORM respectively.
A general formula for these filters is:
   pixel_out = (A*pixel[x+1]+B*pixel[x]+C*pixel[x-1]+Round)/Normalize
The values Round and Normalize are selected based on the values of HORIZ_NORM and VERT_NORM.
```

```
The mapping can be described by the following equations:
   if (NORM = 0)       Round = 0; Normalize = 1;
   else if (NORM < 5)  Round = 2^(NORM); Normalize = 2^(NORM+1);
   else                Round = 2^(5); Normalize = 2^(5+1);
```

15.11.3.1.33.3 Diagram



15.11.3.1.33.4 Fields

Field	Function
31 —	Reserved
30-28 VERT_NORM	Sets the 3 bit vertical filter normalization factor.
27-24 VERT_C	Sets the 4 bit vertical C coefficient.
23-20 VERT_B	Sets the 4 bit vertical B coefficient.
19-16 VERT_A	Sets the 4 bit vertical A coefficient.
15 —	Reserved
14-12 HORIZ_NORM	Sets the 3 bit horizontal filter normalization factor.
11-8 HORIZ_C	Sets the 4 bit horizontal C coefficient.
7-4	Sets the 4 bit horizontal B coefficient.

Table continues on the next page...

Field	Function
HORIZ_B	
3-0 HORIZ_A	Sets the 4 bit horizontal A coefficient.

15.11.3.1.34 (SS_CLIP_CB)

15.11.3.1.34.1 Offset

Register	Offset
SS_CLIP_CB	80h

15.11.3.1.34.2 Function

```
Contains clipping fields for Cb chroma components.
For data bypass, set MIN to 0x0 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits
only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.
```

15.11.3.1.34.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved						MAX									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	Reserved						MIN									
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.11.3.1.34.4 Fields

Field	Function
31-26 —	Reserved
25-16	Set maximum value for Cb clipping function.

Table continues on the next page...

Field	Function
MAX	
15-10 —	Reserved
9-0 MIN	Set minimum value for Cb clipping function.

15.11.3.1.35 (SS_CLIP_CB_SET)

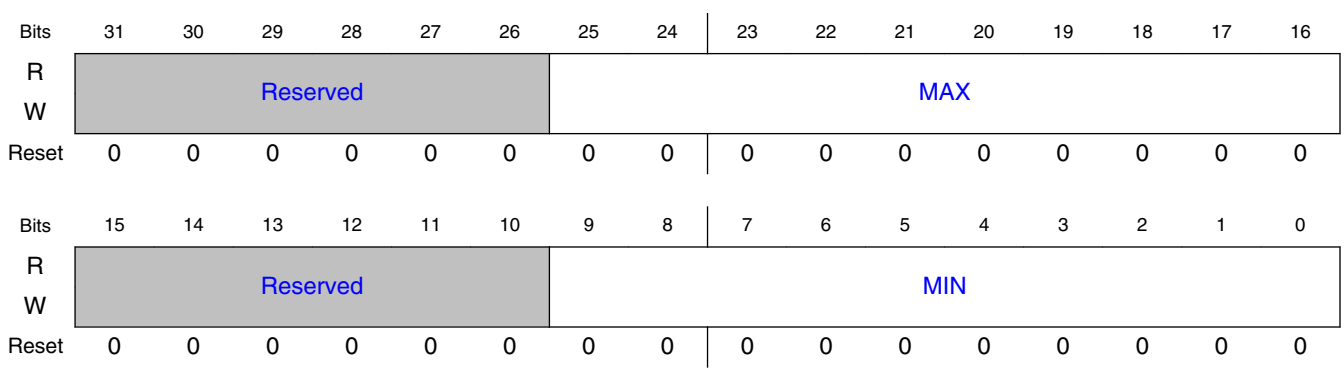
15.11.3.1.35.1 Offset

Register	Offset
SS_CLIP_CB_SET	84h

15.11.3.1.35.2 Function

```
Contains clipping fields for Cb chroma components.
For data bypass, set MIN to 0x0 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits
only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.
```

15.11.3.1.35.3 Diagram



15.11.3.1.35.4 Fields

Field	Function
31-26	Reserved

Table continues on the next page...

Field	Function
—	
25-16 MAX	Set maximum value for Cb clipping function.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cb clipping function.

15.11.3.1.36 (SS_CLIP_CB_CLR)

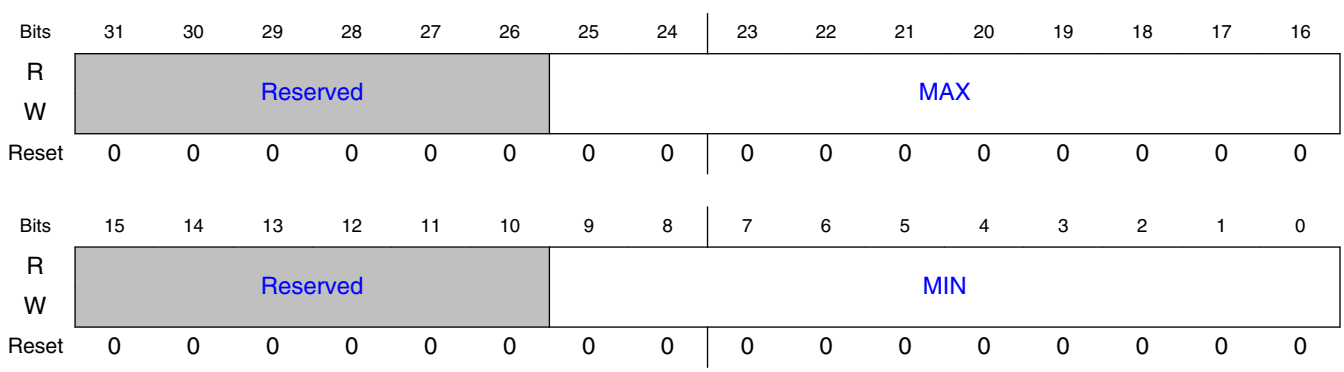
15.11.3.1.36.1 Offset

Register	Offset
SS_CLIP_CB_CLR	88h

15.11.3.1.36.2 Function

```
Contains clipping fields for Cb chroma components.
For data bypass, set MIN to 0x0 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits
only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.
```

15.11.3.1.36.3 Diagram



15.11.3.1.36.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cb clipping function.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cb clipping function.

15.11.3.1.37 (SS_CLIP_CB_TOG)

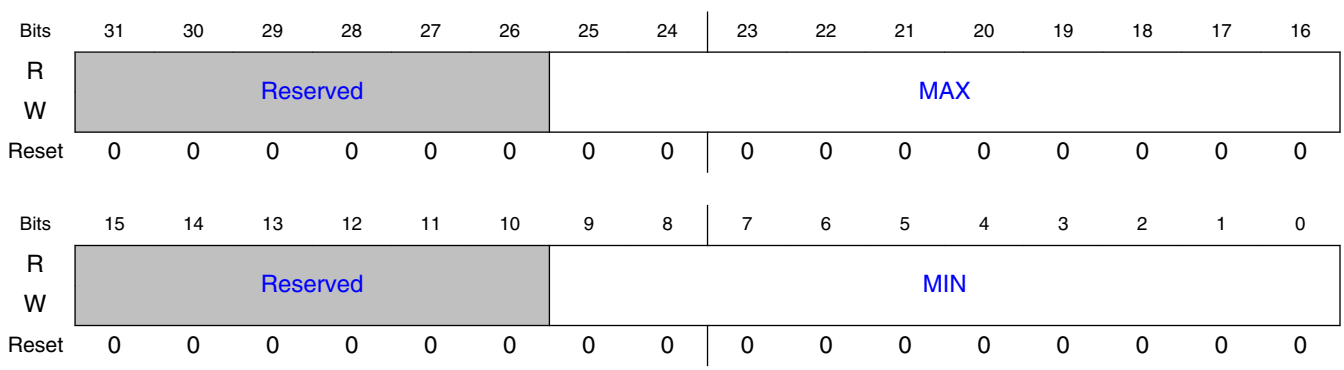
15.11.3.1.37.1 Offset

Register	Offset
SS_CLIP_CB_TOG	8Ch

15.11.3.1.37.2 Function

```
Contains clipping fields for Cb chroma components.
For data bypass, set MIN to 0x0 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits
only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.
```

15.11.3.1.37.3 Diagram



15.11.3.1.37.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cb clipping function.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cb clipping function.

15.11.3.1.38 (SS_CLIP_CR)

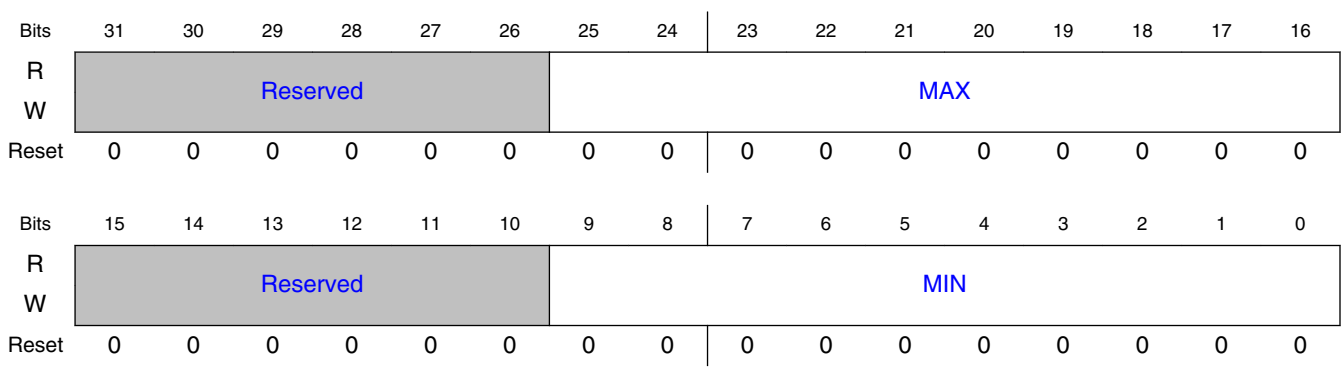
15.11.3.1.38.1 Offset

Register	Offset
SS_CLIP_CR	90h

15.11.3.1.38.2 Function

```
Contains clipping fields for Cr chroma components.
For data bypass, set MIN to 0x000 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits
only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.
```

15.11.3.1.38.3 Diagram



15.11.3.1.38.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cr clipping function. If the Cr output data is greater than this value it is clipped to MAX.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cr clipping function. If the Cr data is less than this value it is clipped to MIN.

15.11.3.1.39 (SS_CLIP_CR_SET)

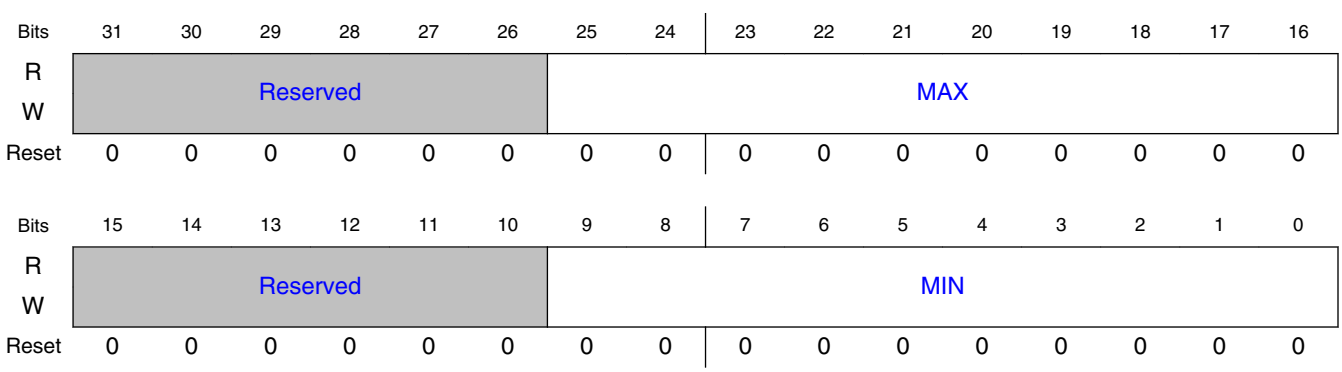
15.11.3.1.39.1 Offset

Register	Offset
SS_CLIP_CR_SET	94h

15.11.3.1.39.2 Function

Contains clipping fields for Cr chroma components.
For data bypass, set MIN to 0x000 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.

15.11.3.1.39.3 Diagram



15.11.3.1.39.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cr clipping function. If the Cr output data is greater than this value it is clipped to MAX.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cr clipping function. If the Cr data is less than this value it is clipped to MIN.

15.11.3.1.40 (SS_CLIP_CR_CLR)

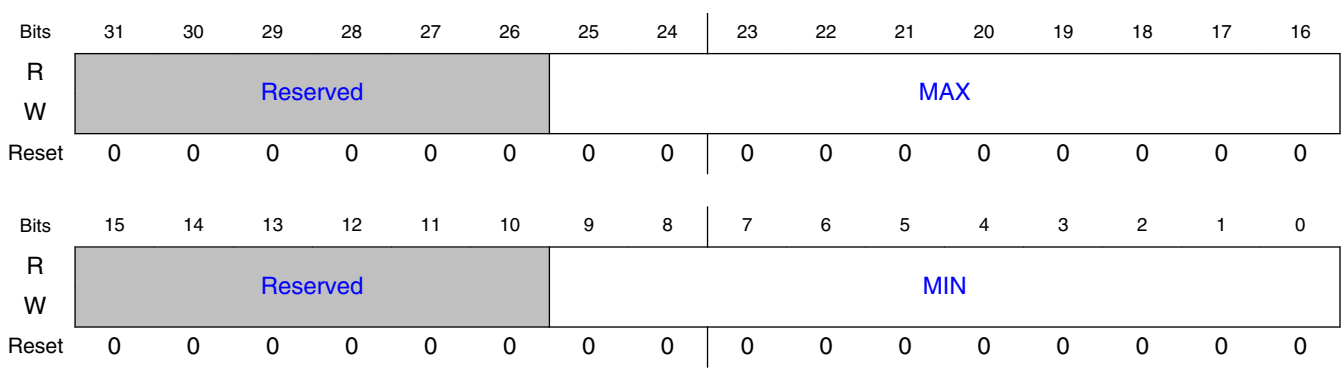
15.11.3.1.40.1 Offset

Register	Offset
SS_CLIP_CR_CLR	98h

15.11.3.1.40.2 Function

Contains clipping fields for Cr chroma components.
For data bypass, set MIN to 0x000 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.

15.11.3.1.40.3 Diagram



15.11.3.1.40.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cr clipping function. If the Cr output data is greater than this value it is clipped to MAX.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cr clipping function. If the Cr data is less than this value it is clipped to MIN.

15.11.3.1.41 (SS_CLIP_CR_TOG)

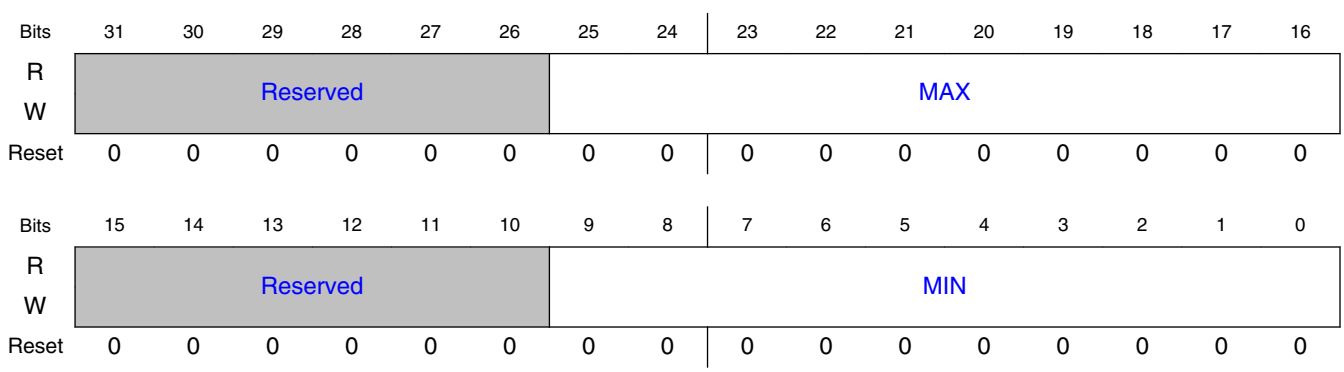
15.11.3.1.41.1 Offset

Register	Offset
SS_CLIP_CR_TOG	9Ch

15.11.3.1.41.2 Function

Contains clipping fields for Cr chroma components.
For data bypass, set MIN to 0x000 and MAX to 0x3FF.
Note, in 444 mode with 12 bit wide chroma components, clipping is applied to upper 10 bits only.
The lower 2 bits of 12 bit data in 444 mode passes through the subsampler untouched.

15.11.3.1.41.3 Diagram



15.11.3.1.41.4 Fields

Field	Function
31-26 —	Reserved
25-16 MAX	Set maximum value for Cr clipping function. If the Cr output data is greater than this value it is clipped to MAX.
15-10 —	Reserved
9-0 MIN	Set minimum value for Cr clipping function. If the Cr data is less than this value it is clipped to MIN.

15.11.3.1.42 (SS_INTER_MODE)

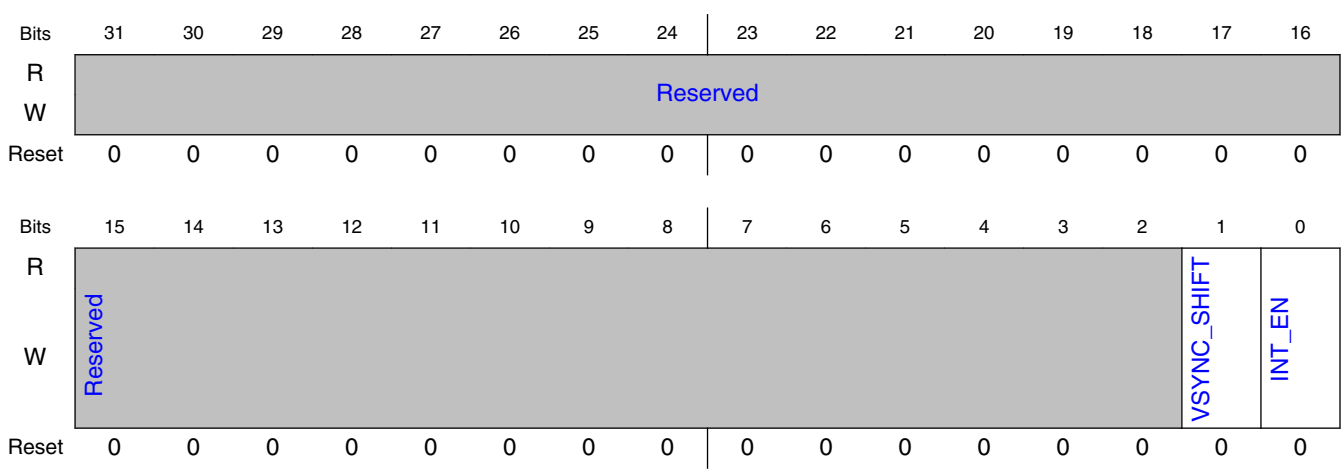
15.11.3.1.42.1 Offset

Register	Offset
SS_INTER_MODE	A0h

15.11.3.1.42.2 Function

Contains bits to enable HDMI timing in interlaced mode.
There are two interlaced formats supported in this module.
General + Special interlaced formats. General occurs when Vtotal is odd.

15.11.3.1.42.3 Diagram



15.11.3.1.42.4 Fields

Field	Function
31-2 —	Reserved
1 VSYNC_SHIFT	Set 0 to run in general interlaced mode. Every other field the vsync is shifted forward 1/2 scan lines. In addition, the horizontal back porch is extended 1 scan line. Set 1 run in special interlaced format. This shifts vsync backwards 1/2 scan line every other field. Remaining timing won't change.
0 INT_EN	Set this bit to 1 to enable interlaced HDMI timing.

15.11.3.1.43 (SS_INTER_MODE_SET)

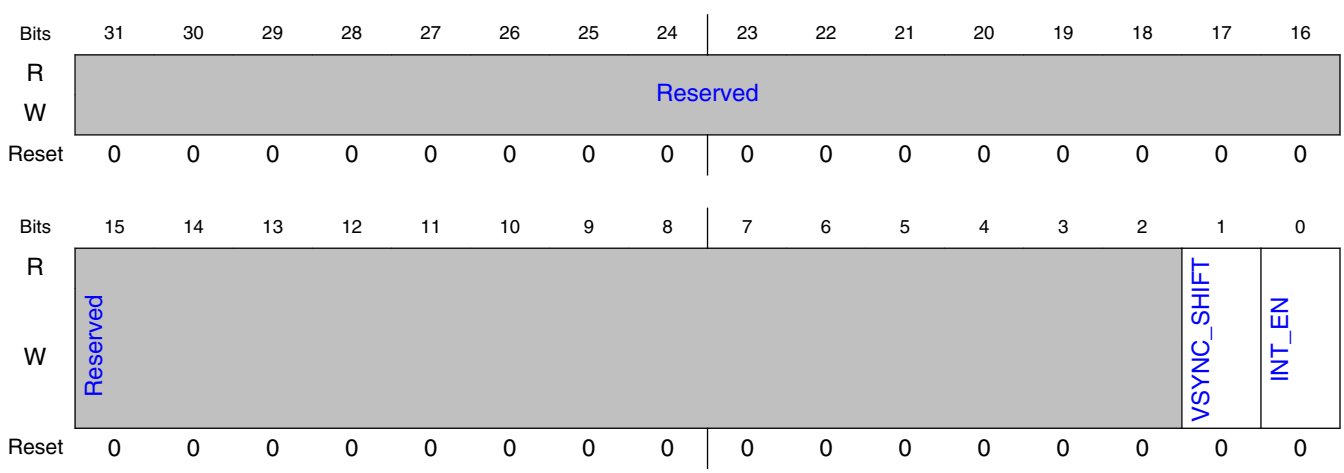
15.11.3.1.43.1 Offset

Register	Offset
SS_INTER_MODE_SET	A4h

15.11.3.1.43.2 Function

Contains bits to enable HDMI timing in interlaced mode.
There are two interlaced formats supported in this module.
General + Special interlaced formats. General occurs when Vtotal is odd.

15.11.3.1.43.3 Diagram



15.11.3.1.43.4 Fields

Field	Function
31-2 —	Reserved
1 VSYNC_SHIFT	Set 0 to run in general interlaced mode. Every other field the vsync is shifted forward 1/2 scan lines. In addition, the horizontal back porch is extended 1 scan line. Set 1 run in special interlaced format. This shifts vsync backwards 1/2 scan line every other field. Remaining timing won't change.
0 INT_EN	Set this bit to 1 to enable interlaced HDMI timing.

15.11.3.1.44 (SS_INTER_MODE_CLR)

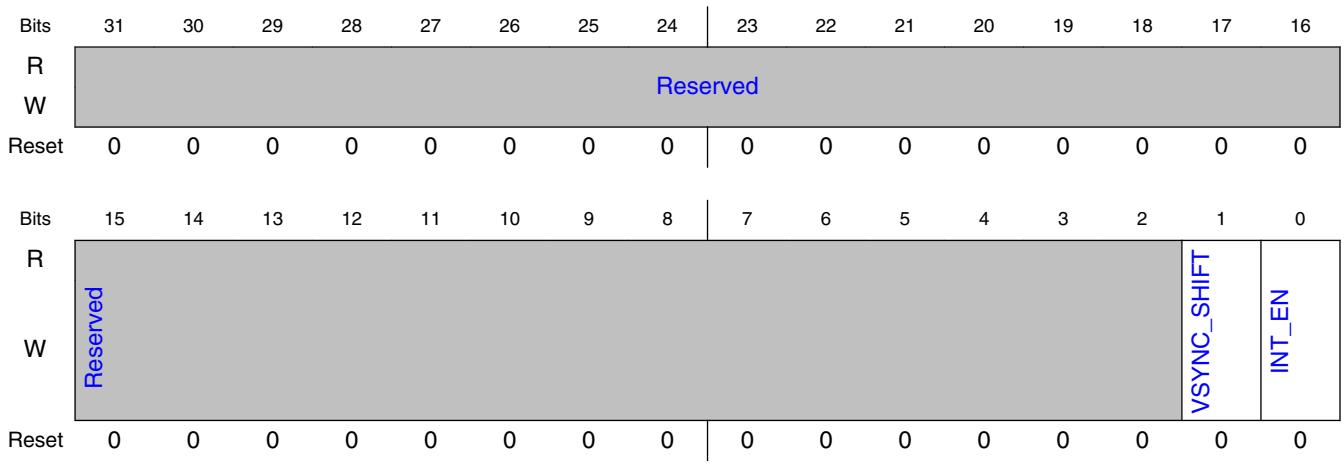
15.11.3.1.44.1 Offset

Register	Offset
SS_INTER_MODE_CLR	A8h

15.11.3.1.44.2 Function

Contains bits to enable HDMI timing in interlaced mode.
There are two interlaced formats supported in this module.
General + Special interlaced formats. General occurs when Vtotal is odd.

15.11.3.1.44.3 Diagram



15.11.3.1.44.4 Fields

Field	Function
31-2 —	Reserved
1 VSYNC_SHIFT	Set 0 to run in general interlaced mode. Every other field the vsync is shifted forward 1/2 scan lines. In addition, the horizontal back porch is extended 1 scan line. Set 1 run in special interlaced format. This shifts vsync backwards 1/2 scan line every other field. Remaining timing won't change.
0 INT_EN	Set this bit to 1 to enable interlaced HDMI timing.

15.11.3.1.45 (SS_INTER_MODE_TOG)

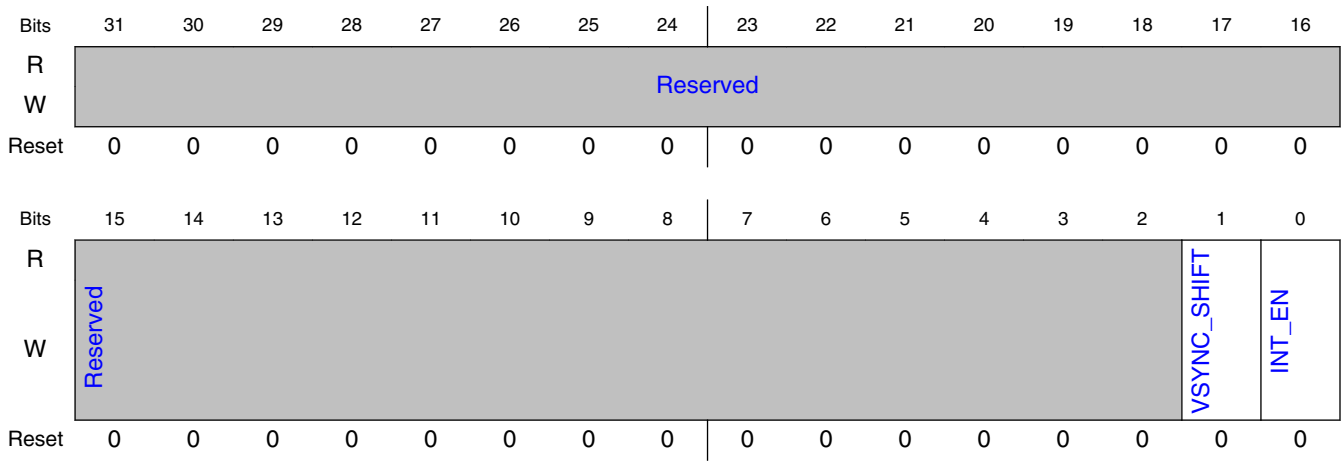
15.11.3.1.45.1 Offset

Register	Offset
SS_INTER_MODE_TOG	ACh

15.11.3.1.45.2 Function

Contains bits to enable HDMI timing in interlaced mode.
There are two interlaced formats supported in this module.
General + Special interlaced formats. General occurs when Vtotal is odd.

15.11.3.1.45.3 Diagram



15.11.3.1.45.4 Fields

Field	Function
31-2 —	Reserved
1 VSYNC_SHIFT	Set 0 to run in general interlaced mode. Every other field the vsync is shifted forward 1/2 scan lines. In addition, the horizontal back porch is extended 1 scan line. Set 1 run in special interlaced format. This shifts vsync backwards 1/2 scan line every other field. Remaining timing won't change.
0 INT_EN	Set this bit to 1 to enable interlaced HDMI timing.

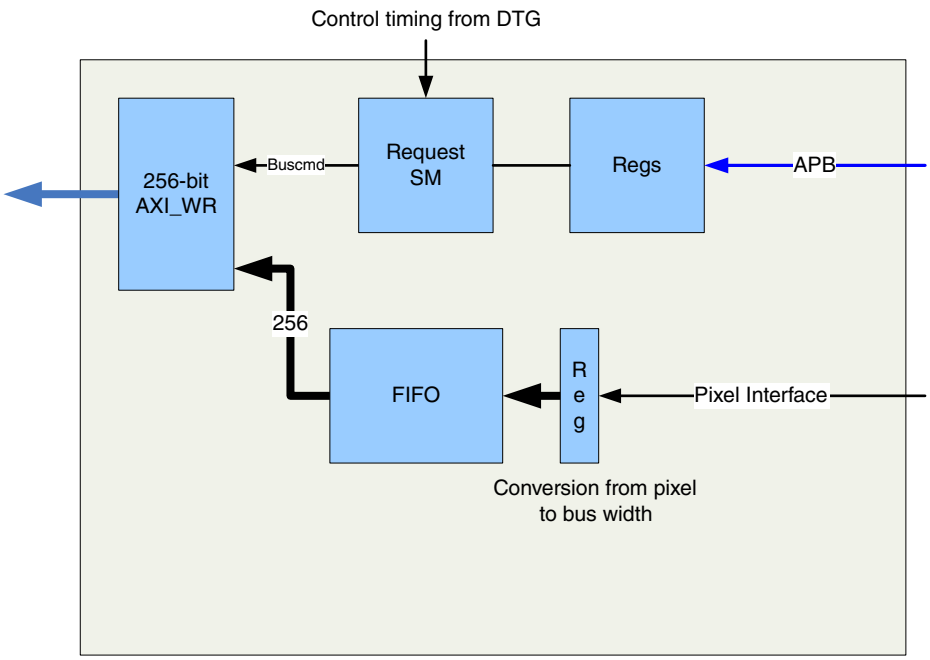
15.12 Write Scale (WR_SCL)

15.12.1 Overview

The Write Scale engine (WR_SCL) can be used to send pixel data from the scaler to DDR. This path to external memory would be used when significant downscaling factors are selected due to large bandwidth requirements on the source image within smaller regions of the display output. The WR_SCL module accepts valid pixels from the scaling engine and packs them into an output FIFO to manage bursts of data out onto the AXI interface to external memory.

15.12.1.1 Block Diagram

The diagram indicating the functional components in the WR_SCL module is below.



15.12.1.2 Features

The WR_SCL module is a bus master engine that accepts pixels from the scaler and generates packets of requests over the AXI interface.

The components in the WR_SCL are highlighted here:

- Req_SM: State machine to generate the sequence of transactions to write a buffer to system memory.
- Regs: All configuration registers used to store the WR_SCL configuration.
- AXI_WR: The buscmd to AXI interface convertor used in many AXI based designs.
- FIFO: used to store a sample set of data to be written to system memory.
- Up sample Reg: Register used to convert from pixel bus width to bus width.

15.12.2 Functional Description

It is expected that several parameters provides the configuration options to achieve buffer sizes and access patterns that maximizes DDR efficiency and design size. There are hardware and software parameters for the design as defined below.

WR_SCL SW parameters are:

- AXI transaction size (64B up to 512B per burst) is programmable.
- Buffer threshold: defines how much data needs to be in the FIFO before issuing a burst of system memory requests. When the number of bytes is greater than this value, the necessary AXI requests to transfer the data are issued
- Burst frequency: how often the wr_scale engine issues a sequence of AXI requests
- Buffer base address
- Line stride in bytes

15.12.2.1 WR_SCL Output Pixel Formatting

There are a variety of input to output format conversions that are available in the WR_SCL. The input is always 10-bit per component with an 8-bit alpha. Consumers of the data could be 8-bit or 10-bit, so the option to convert the output to 8-bit can be selected. Also, the consumer may not use an alpha value, so it can be truncated. The output pixel format should be influenced by the final consumer of pixels. It is up to SW to determine which to use. The idea is that data that is not needed at the consumer can be removed to save memory bandwidth.

The input pixel_data[37:0] to output conversions are programmed in the BPP field of the CSR_WR_SCL_CTRL register. The input to output conversion of the pixels is defined for each mode as below. Note that P0 is the first pixel received on the pixel_data[37:0] input bus, P1 is the second, etc.

BPP = 0x0: 40bpp, Full pass through of all 38-bits appended with 2’b00 for each pixel. Appending 2’b00 is intended to created byte alignment. The pixels in bold wrap on the 128-bit boundary.

This pattern repeats.

```
1st pixel_out[127:0] = {P3[7:0],
                        2'b00,P2[37:0],
                        2'b00,P1[37:0],
                        2'b00,P0[37:0]};

2nd pixel_out[127:0] = {P6[15:0],
                        2'b00,P5[37:0],
```

```
                          2'b00,P4[37:0],
                          2'b00,P3[37:8]};

3rd pixel_out[127:0] = {P9[23:0],
                          2'b00,P8[37:0],
                          2'b00,P7[37:0],
                          2'b00,P6[37:16]};

4th pixel_out[127:0] = {P12[31:0],
                          2'b00,P11[37:0],
                          2'b00,P10[37:0],
                          2'b00,P9[37:24]};
```

BPP = 0x1: 32bpp, ARGB8888/AYUV8888

The lowest two bits for each color component are dropped. Alpha is on the high order byte.

```
Pixel_out[127:0] = { P3[37:30,29:22,19:12,9:2],
                     P2[37:30,29:22,19:12,9:2],
                     P1[37:30,29:22,19:12,9:2],
                     P0[37:30,29:22,19:12,9:2]}
```

BPP = 0x2: 32bpp, RGB/YUV 10-bit per component

All 10-bits of each component are used. The 2 highest bits of alpha are passed.

```
Pixel_out[127:0] = { P3[37:36,29:20,19:10,9:0],
                     P2[37:36,29:20,19:10,9:0],
                     P1[37:36,29:20,19:10,9:0],
                     P0[37:36,29:20,19:10,9:0]}
```

BPP = 0x3: 24bpp, YUV422 10-bit per component

Y (pixel_data[29:20]) with alternate U/V (pixel_data[19:10]). The 20-bit pixel is concatenated with 4-bit Alpha.

```
1st pixel_out[127:0] = { P5[17:10],
                         P4[37:34],P4[29:10],
                         P3[37:34],P3[29:10],
                         P2[37:34],P2[29:10],
                         P1[37:34],P1[29:10],
                         P0[37:34],P0[29:10] }

2nd pixel_out[127:0] = { P10[25:10],
                         P9[37:34],P9[29:10],
                         P8[37:34],P8[29:10],
                         P7[37:34],P7[29:10],
                         P6[37:34],P6[29:10],
                         P5[37:34],P5[29:18] }

3rd pixel_out[127:0] = { P15[37:34],P15[27:10],
                         P14[37:34],P14[29:10],
                         P13[37:34],P13[29:10],
                         P12[37:34],P12[29:10],
                         P11[37:34],P11[29:10],
                         P10[37:34],P10[29:26] }
```

This pattern repeats.

BPP = 0x4: 16bpp, YUV422 8-bit per component

The lowest two bits for each color component are dropped. Pixel_data[9:0] is dropped and not used in YUV422 mode. No alpha is used.

Pixel_out[127:0] = { P7[29:22,19:12], P6, …, P3, P2[29:22,19:12], P1[29:22,19:12], P0[29:22,19:12] }

15.12.2.2 WR_SCL Interrupt

The WR_SCL module can generate interrupts under the following conditions.

- AXI SLVERR or DECERR is received.
- Frame processing complete. The last pixel coincident with pixel_v_sync is written to system memory.

15.12.3 Memory Map and Registers

15.12.3.1 register descriptions

15.12.3.1.1 WR_SCL Memory map

wr_scl base address: 2_1000h

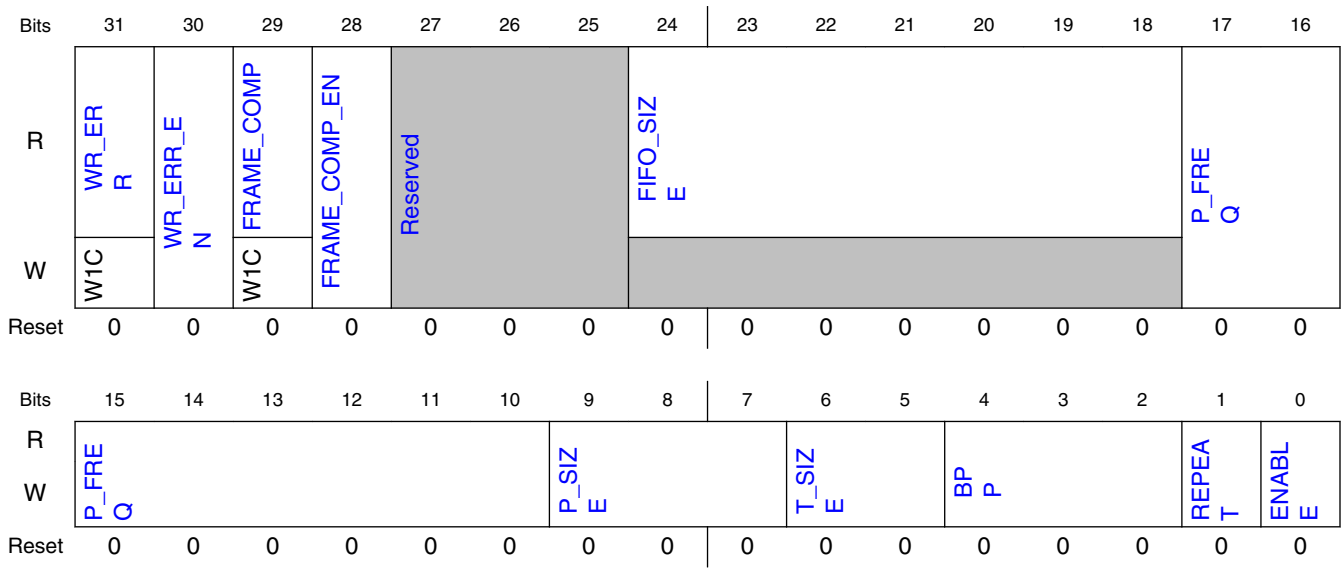
Offset	Register	Width (In bits)	Access	Reset value
0h	Control register for Context Loader. (CTRL_STATUS)	32	RW	0000_0000h
4h	Control register for Context Loader. (CTRL_STATUS_SET)	32	RW	0000_0000h
8h	Control register for Context Loader. (CTRL_STATUS_CLR)	32	RW	0000_0000h
Ch	Control register for Context Loader. (CTRL_STATUS_TOG)	32	RW	0000_0000h
10h	Holds the base address (BASE_ADDR)	32	RW	0000_0000h
14h	Pitch (PITCH)	32	RW	0000_0000h

15.12.3.1.2 Control register for Context Loader. (CTRL_STATUS)

15.12.3.1.2.1 Offset

Register	Offset
CTRL_STATUS	0h

15.12.3.1.2.2 Diagram



15.12.3.1.2.3 Fields

Field	Function
31 WR_ERR	Indicates a write error on the axi interface.
30 WR_ERR_EN	Write error IRQ enable If set, it allows WR_ERR to create an irq whenever there is an error responsde to a write request.
29 FRAME_COMP	Indicates the current frame being processed has finished.
28 FRAME_COMP_EN	Write error IRQ enable If set, it allows FRAME_COMP to create an irq whenever processing a frame finishes.
27-25 —	Reserved
24-18 FIFO_SIZE	Size of FIFO in design This is the fifo size in the HW. This can be used to program the appropriate P_SIZE. P_SIZE should ideally be 1/2 the FIFO size. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
17-10 P_FREQ	Payload frequency P_FREQ*8 is the number of clocks between payload requests. A value of 0 does not limit the frequency of the payload requests. Instead, the payload is written as frequently as data is available.
9-7	Payload Size

Table continues on the next page...

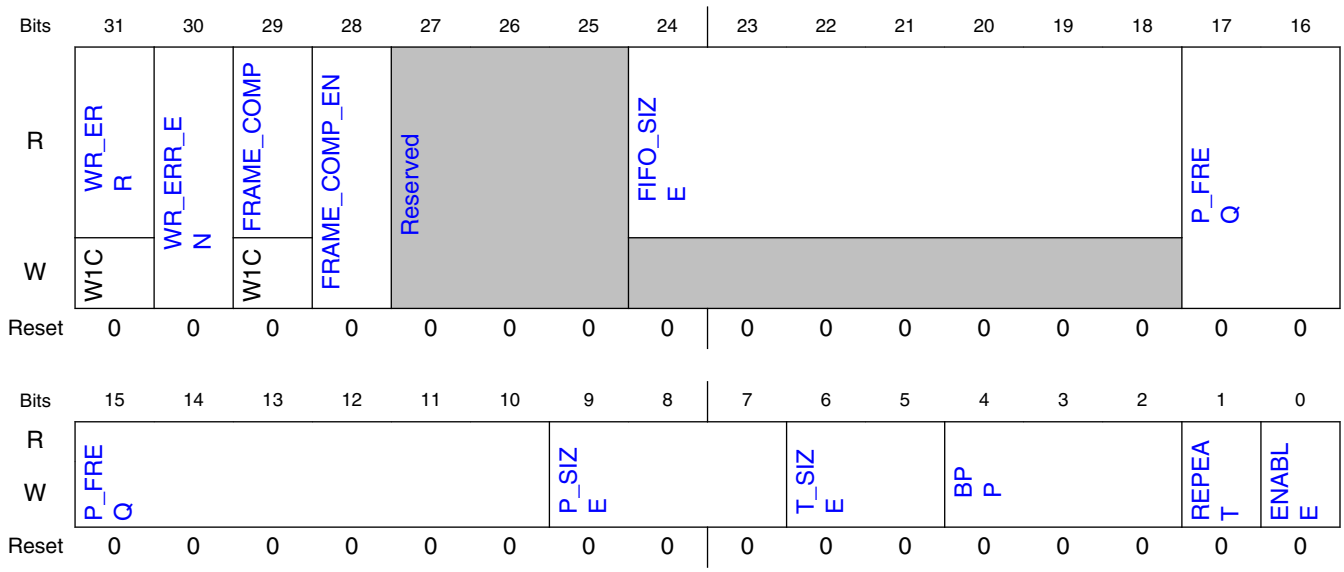
Field	Function
P_SIZE	Holds the payload size. This is used with the T_SIZE to calculate the number of transactions required per request. PSIZE and TSIZE are co-related and should be programmed as PSIZE >= TSIZE. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • 100 -> 1KB • 101 -> 2KB • 110 -> 4KB • 111 -> Reserved
6-5 T_SIZE	Transaction Size Holds the transaction size / request. Transaction size determines how many AXI transactions a payload should be divided into.T_SIZE is related to P_SIZE and needs to be programmed as PSIZE >= TSIZE • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits / pixel format interface on the output. • 0 -> 38 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 20 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 REPEAT	Repeat feature Repeat bit for the write_scalar. When set, write scalar does not lower the BUSY bit after each frame and processes the new frame of data with the same address, bpp, p_size and t_size values as the previous frame. If you do not expect a new frame, reset this signal before the current frame finishes. Once the frame finishes, and the design sees this bit set, it resets address to the programmed value and wait for new pixels. If you do not expect any more pixels, the only way to recover from this is through a reset.
0 ENABLE	Enable / Busy Enable bit for the write_scalar. When set, the wr_scalar waits for pixels and write them to memory as per the address and transfer sizes programmed in the control register. This bit is reset by HW when it has completed the frame write. If the REPEAT bit is set, this bit is not cleared as the WR_SCL resets it's pointers and waits for new pixels to be written to the same address in memory.

15.12.3.1.3 Control register for Context Loader. (CTRL_STATUS_SET)

15.12.3.1.3.1 Offset

Register	Offset
CTRL_STATUS_SET	4h

15.12.3.1.3.2 Diagram



15.12.3.1.3.3 Fields

Field	Function
31 WR_ERR	Indicates a write error on the axi interface.
30 WR_ERR_EN	Write error IRQ enable If set, it allows WR_ERR to create an irq whenever there is an error responsde to a write request.
29 FRAME_COMP	Indicates the current frame being processed has finished.
28 FRAME_COMP_EN	Write error IRQ enable If set, it allows FRAME_COMP to create an irq whenever processing a frame finishes.
27-25 —	Reserved
24-18 FIFO_SIZE	Size of FIFO in design This is the fifo size in the HW. This can be used to program the appropriate P_SIZE. P_SIZE should ideally be 1/2 the FIFO size. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
17-10 P_FREQ	Payload frequency P_FREQ*8 is the number of clocks between payload requests. A value of 0 does not limit the frequency of the payload requests. Instead, the payload is written as frequently as data is available.
9-7 P_SIZE	Payload Size

Table continues on the next page...

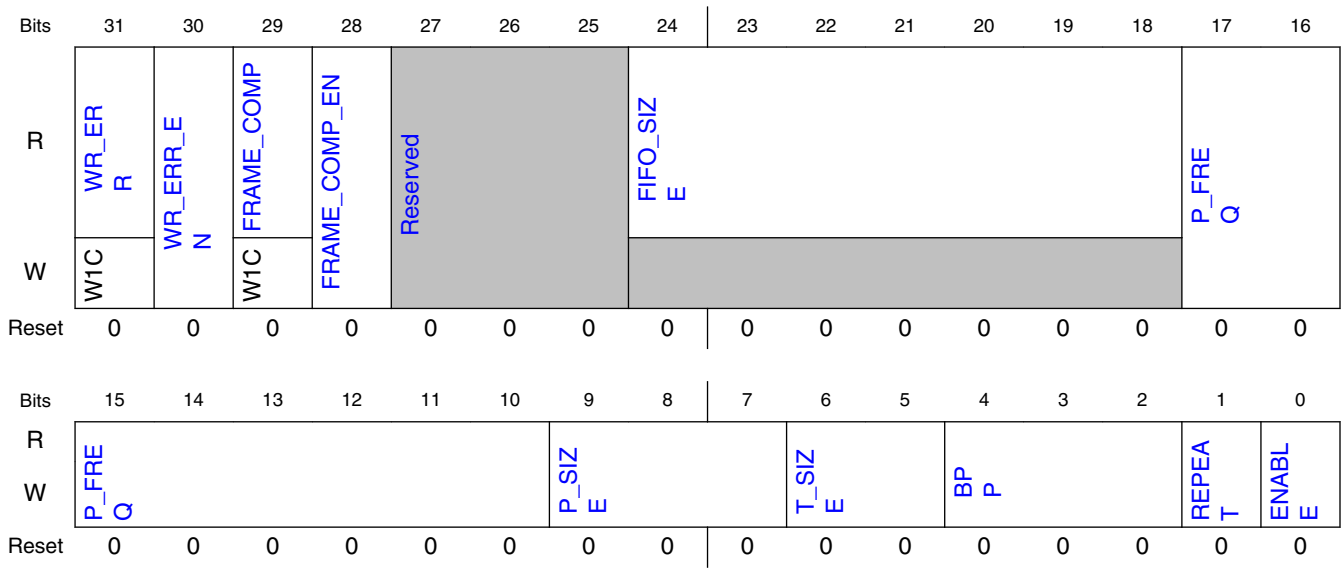
Field	Function
	Holds the payload size. This is used with the T_SIZE to calculate the number of transactions required per request. PSIZE and TSIZE are co-related and should be programmed as PSIZE >= TSIZE. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • 100 -> 1KB • 101 -> 2KB • 110 -> 4KB • 111 -> Reserved
6-5 T_SIZE	Transaction Size Holds the transaction size / request. Transaction size determines how many AXI transactions a payload should be divided into.T_SIZE is related to P_SIZE and needs to be programmed as PSIZE >= TSIZE • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits / pixel format interface on the output. • 0 -> 38 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 20 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 REPEAT	Repeat feature Repeat bit for the write_scalar. When set, write scalar does not lower the BUSY bit after each frame and processes the new frame of data with the same address, bpp, p_size and t_size values as the previous frame. If you do not expect a new frame, reset this signal before the current frame finishes. Once the frame finishes, and the design sees this bit set, it resets address to the programmed value and wait for new pixels. If you do not expect any more pixels, the only way to recover from this is through a reset.
0 ENABLE	Enable / Busy Enable bit for the write_scalar. When set, the wr_scalar waits for pixels and write them to memory as per the address and transfer sizes programmed in the control register. This bit is reset by HW when it has completed the frame write. If the REPEAT bit is set, this bit is not cleared as the WR_SCL resets it's pointers and waits for new pixels to be written to the same address in memory.

15.12.3.1.4 Control register for Context Loader. (CTRL_STATUS_CLR)

15.12.3.1.4.1 Offset

Register	Offset
CTRL_STATUS_CLR	8h

15.12.3.1.4.2 Diagram



15.12.3.1.4.3 Fields

Field	Function
31 WR_ERR	Indicates a write error on the axi interface.
30 WR_ERR_EN	Write error IRQ enable If set, it allows WR_ERR to create an irq whenever there is an error responsde to a write request.
29 FRAME_COMP	Indicates the current frame being processed has finished.
28 FRAME_COMP_EN	Write error IRQ enable If set, it allows FRAME_COMP to create an irq whenever processing a frame finishes.
27-25 —	Reserved
24-18 FIFO_SIZE	Size of FIFO in design This is the fifo size in the HW. This can be used to program the appropriate P_SIZE. P_SIZE should ideally be 1/2 the FIFO size. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
17-10 P_FREQ	Payload frequency P_FREQ*8 is the number of clocks between payload requests. A value of 0 does not limit the frequency of the payload requests. Instead, the payload is written as frequently as data is available.
9-7 P_SIZE	Payload Size

Table continues on the next page...

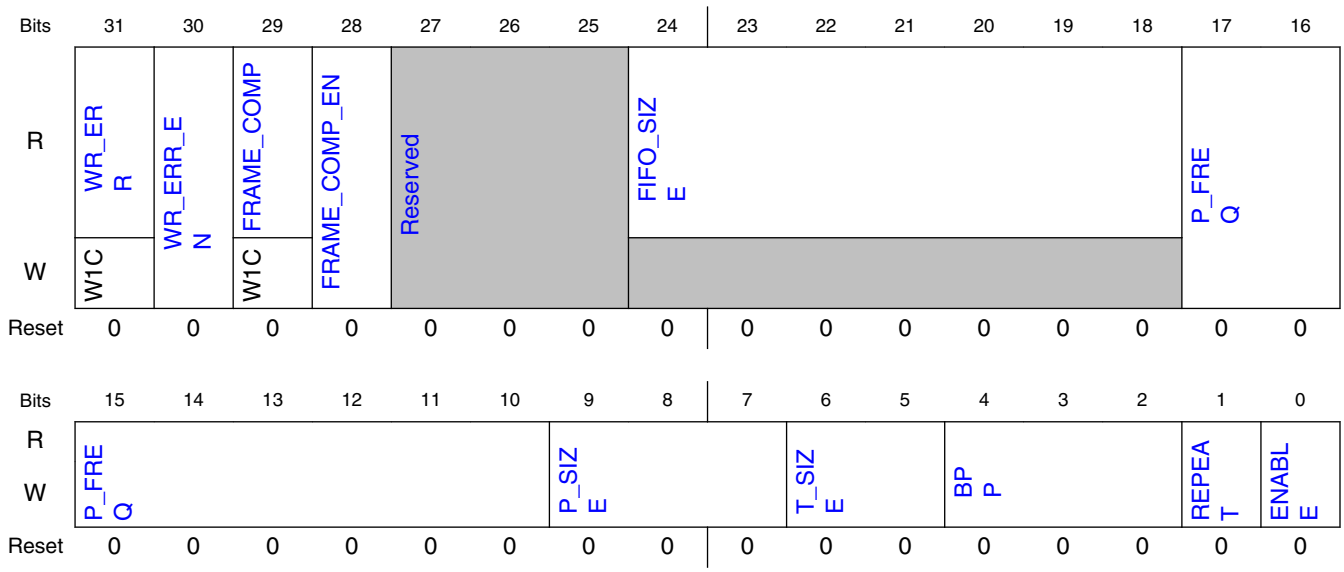
Field	Function
	Holds the payload size. This is used with the T_SIZE to calculate the number of transactions required per request. PSIZE and TSIZE are co-related and should be programmed as PSIZE >= TSIZE. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • 100 -> 1KB • 101 -> 2KB • 110 -> 4KB • 111 -> Reserved
6-5 T_SIZE	Transaction Size Holds the transaction size / request. Transaction size determines how many AXI transactions a payload should be divided into.T_SIZE is related to P_SIZE and needs to be programmed as PSIZE >= TSIZE • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits / pixel format interface on the output. • 0 -> 38 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 20 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 REPEAT	Repeat feature Repeat bit for the write_scalar. When set, write scalar does not lower the BUSY bit after each frame and processes the new frame of data with the same address, bpp, p_size and t_size values as the previous frame. If you do not expect a new frame, reset this signal before the current frame finishes. Once the frame finishes, and the design sees this bit set, it resets address to the programmed value and wait for new pixels. If you do not expect any more pixels, the only way to recover from this is through a reset.
0 ENABLE	Enable / Busy Enable bit for the write_scalar. When set, the wr_scalar waits for pixels and write them to memory as per the address and transfer sizes programmed in the control register. This bit is reset by HW when it has completed the frame write. If the REPEAT bit is set, this bit is not cleared as the WR_SCL resets it's pointers and waits for new pixels to be written to the same address in memory.

15.12.3.1.5 Control register for Context Loader. (CTRL_STATUS_TOG)

15.12.3.1.5.1 Offset

Register	Offset
CTRL_STATUS_TOG	Ch

15.12.3.1.5.2 Diagram



15.12.3.1.5.3 Fields

Field	Function
31 WR_ERR	Indicates a write error on the axi interface.
30 WR_ERR_EN	Write error IRQ enable If set, it allows WR_ERR to create an irq whenever there is an error responsde to a write request.
29 FRAME_COMP	Indicates the current frame being processed has finished.
28 FRAME_COMP_EN	Write error IRQ enable If set, it allows FRAME_COMP to create an irq whenever processing a frame finishes.
27-25 —	Reserved
24-18 FIFO_SIZE	Size of FIFO in design This is the fifo size in the HW. This can be used to program the appropriate P_SIZE. P_SIZE should ideally be 1/2 the FIFO size. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
17-10 P_FREQ	Payload frequency P_FREQ*8 is the number of clocks between payload requests. A value of 0 does not limit the frequency of the payload requests. Instead, the payload is written as frequently as data is available.
9-7 P_SIZE	Payload Size

Table continues on the next page...

Field	Function
	Holds the payload size. This is used with the T_SIZE to calculate the number of transactions required per request. PSIZE and TSIZE are co-related and should be programmed as PSIZE >= TSIZE. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • 100 -> 1KB • 101 -> 2KB • 110 -> 4KB • 111 -> Reserved
6-5 T_SIZE	Transaction Size Holds the transaction size / request. Transaction size determines how many AXI transactions a payload should be divided into.T_SIZE is related to P_SIZE and needs to be programmed as PSIZE >= TSIZE • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits / pixel format interface on the output. • 0 -> 38 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 20 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 REPEAT	Repeat feature Repeat bit for the write_scalar. When set, write scalar does not lower the BUSY bit after each frame and processes the new frame of data with the same address, bpp, p_size and t_size values as the previous frame. If you do not expect a new frame, reset this signal before the current frame finishes. Once the frame finishes, and the design sees this bit set, it resets address to the programmed value and wait for new pixels. If you do not expect any more pixels, the only way to recover from this is through a reset.
0 ENABLE	Enable / Busy Enable bit for the write_scalar. When set, the wr_scalar waits for pixels and write them to memory as per the address and transfer sizes programmed in the control register. This bit is reset by HW when it has completed the frame write. If the REPEAT bit is set, this bit is not cleared as the WR_SCL resets it's pointers and waits for new pixels to be written to the same address in memory.

15.12.3.1.6 Holds the base address (BASE_ADDR)

15.12.3.1.6.1 Offset

Register	Offset
BASE_ADDR	10h

15.12.3.1.6.2 Function

First comment.

secone third

15.12.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.12.3.1.6.4 Fields

Field	Function
31-0 BASE_ADDR	Base address System address of source buffer. Although byte granularity is supported, SW should align it to 64B boundry

15.12.3.1.7 Pitch (PITCH)

15.12.3.1.7.1 Offset

Register	Offset
PITCH	14h

15.12.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.12.3.1.7.3 Fields

Field	Function
31-16 —	Reserved
15-0 PITCH	Vertical pitch for memory address calculation. Number of bytes between 2 vertically adjacent pixels in system memory. Byte granularity is supported, but SW should align to 64B boundry.

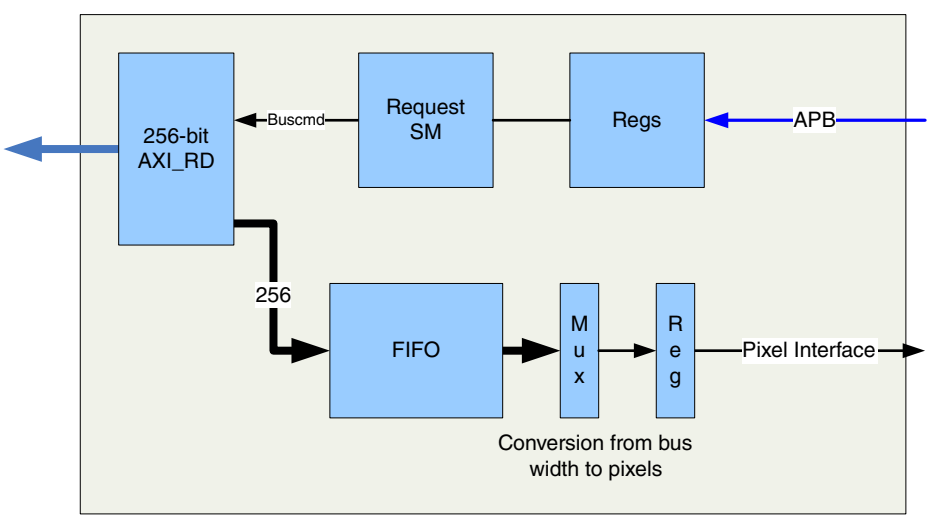
15.13 Read Surface (RD_SRC)

15.13.1 Overview

The Read Surface (RD_SRC) engine is a bus master that reads pixels from system memory and drive pixels out to the scale output buffer. The scale output buffer (1 of 3 not used for storing the scaled result) is reused as a read latency elasticity buffer. Minimal storage is needed in the RD_SRC module to store returned read data since the scale output buffer is repurposed as a pixel elasticity buffer. The RD_SRC engine supports raster scan source buffers. There is no support for tiled based buffers.

15.13.1.1 Block Diagram

The diagram indicating the functional components in the RD_SRC module is below.



The components are listed here:

- Req_SM: State machine to generate the sequence of transactions to read a buffer from system memory.
- Regs: All configuration registers used to store the RD_SRC configuration.
- AXI_RD: The buscmd to AXI interface convertor used in many AXI based designs. This instance is shared with the CTX_LD module.
- FIFO: used to store a sample set of data to be read from system memory.
- Down Sample Mux/Reg: Multiplexor used to down size from the bus width to a pixel interface to the scale output buffer.

The software parameters are:

- AXI transaction size (64B up to 512B per request) is programmable.
- Buffer threshold determines how empty the input data FIFO should be to trigger a sequence of read transactions from the system. When the number of available bytes is greater than this value, the necessary AXI requests to fill this buffer are issued.
- Buffer base address.
- Line stride in bytes.
- Source buffer format, either ARGB8888 (32bpp), or YUV422 1-plane (16bpp).
- Width, or number of pixels to fetch and write to the scale output buffer in a horizontal scan line.
- Height, number of lines in the source buffer.

15.13.2 Functional Description

The RD_SRC engine fetches and forwards only the pixels needed to drive the display. The granularity of the source buffer, in X and Y, forwarded to the display pipeline has no limits. Any width and height can be fetched and forwarded to the scale output buffer. Also, in YUV422 mode, the programmed width should be divisible by 2.

The RD_SRC has two events that can enable it to issue requests to fetch data for the next display trace.

1. SW can write the BUSY bit to 1’b1.
2. The rd_src_kick_enable signal input can go low to high.

The RD_SRC is disabled when the csr_rd_src_width AND csr_rd_src_height = 0x0. When these two registers are 0x0, both events that enable the RD_SRC are ignored. Also, if the RD_SRC is already enabled and processing a frame, any of the two events that are used to start processing would be ignored.

15.13.2.1 Input to Output Pixel Formatting

The pixel_data[37:0] output bus to the Scaler engine is normalized as 10-bit per color component, with an 8-bit alpha. Refer to section 4.21.1 Pixel Component Order for the standard on how to convert 8-bit to 10-bit color components when required depending on the format selected.

Input pixels from the system can be in 8/10-bit format, may contain an alpha value, or may have sub-sampled chroma components. The BPP field of the CSR_RD_SRC_CTRL register defines what the input format is. The definition of this field matches the respective control field in the WR_SCL module. Essentially, the RD_SRC reverses the pixel formatting process and send a full 38-bits back to the Scaler engine. The definition of this data format is defined as

```
Pixel_data[37:30] = Alpha
Pixel_data[29:20] = R/Y
Pixel_data[19:10] = G/U, or UV alternating in YUV422 modes
Pixel_data[9:0] = B/V
```

The output pixel modes are defined as:

- **BPP = 0x0: 40bpp**, Full pass through of all 38-bits with the appended 2'b00 bits dropped.
  - 1st pixel_data = [37:0],
  - 2nd pixel_data = [77:40], etc.
- **BPP = 0x1: 32bpp**, ARGB8888/AYUV8888
- 32bpp expanded to the full 38-bit vector. Refer to the standards section 4.21.2 Pixel Component Width Conversion to determine how to expand 8-bit components up to 10-bit.
- 1st pixel_data = {[31:24],[23:16],[23:22],[15:8],[15:14],[7:0],[7:6]} //ARGB: 8:10:10:10
- **BPP = 0x2: 32bpp**, RGB/YUV 10-bit per component
- All 10-bits of each component are passed to the output. The alpha is expanded from 2-bits to 8-bit.
- 1st pixel_data = {{4{[31:30]}},[29:20],[19:10],[9:0]} //Alpha expanded, component pass-through
- **BPP = 0x3: 24bpp**, YUV422 10-bit per component
- The full Y and alternating UV components are passed through as 10-bit. Bits 9:0 are driven as 10'b0. The 4-bit alpha is converted based on the standard conversion described in section 4.21.2 Pixel Component Width Conversion.
- 1st pixel_data = {{2{[23:20]}},[19:10],[9:0],10'b0} //Alpha expanded 4 to 8, Y and UV pass-through
- **BPP = 0x4: 16bpp**, YUV422 8-bit per component
- The Y and alternating UV components are expanded from 8 to 10 bits. Bits [9:0] and [37:30] are driven as 0x0.
- 1st pixel_data = {8'b0,[15:7],[15:14],[7:0],[7:6],10'b0} //Alpha = 0,

15.13.2.2 Interrupt

The RD_SRC module can generate interrupts under the following conditions.

- AXI SLVERR or DECERR is received.
- Frame processing complete. The last pixel coincident with pixel_v_sync is written to the scaler output FIFO.

15.13.3 Memory Map and Registers

15.13.3.1 register descriptions

15.13.3.1.1 RD_SRC Memory map

rd_src base address: 2_2000h

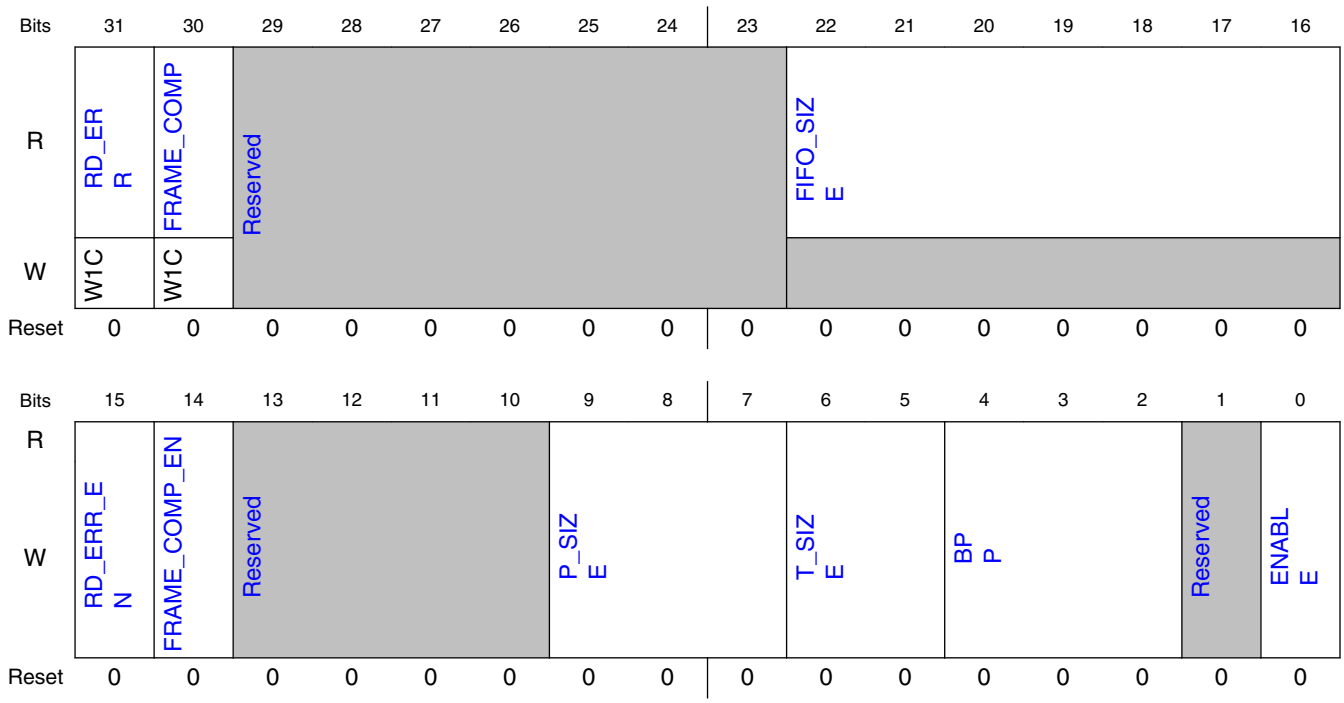
Offset	Register	Width (In bits)	Access	Reset value
0h	Control register for Read surface. (CTRL_STATUS)	32	RW	0000_0000h
4h	Control register for Read surface. (CTRL_STATUS_SET)	32	RW	0000_0000h
8h	Control register for Read surface. (CTRL_STATUS_CLR)	32	RW	0000_0000h
Ch	Control register for Read surface. (CTRL_STATUS_TOG)	32	RW	0000_0000h
10h	Read Surface Base address (BASE_ADDR)	32	RW	0000_0000h
14h	Read surface vertical pitch (PITCH)	32	RW	0000_0000h
18h	Source frame buffer width (WIDTH)	32	RW	0000_0000h
1Ch	Height of frame to be read (HEIGHT)	32	RW	0000_0000h

15.13.3.1.2 Control register for Read surface. (CTRL_STATUS)

15.13.3.1.2.1 Offset

Register	Offset
CTRL_STATUS	0h

15.13.3.1.2.2 Diagram



15.13.3.1.2.3 Fields

Field	Function
31 RD_ERR	AXI Read Error Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
30 FRAME_COMP	Frame processing complete Status bit that indicates the current frame processing is complete. This is set even if the height and width are 0. If the respective interrupt enable bit is set, an interrupt is generated.
29-23 —	Reserved
22-16 FIFO_SIZE	FIFO size This is the fifo size implemented in the HW. The P_SIZE should never be set to a value higher than this value in bytes. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
15 RD_ERR_EN	AXI Read Error IRQ enable Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
14 FRAME_COMP _EN	Frame complete IRQ enable Enable an interrupt when the read surface engine is finished processing a frame.
13-10	Reserved

Table continues on the next page...

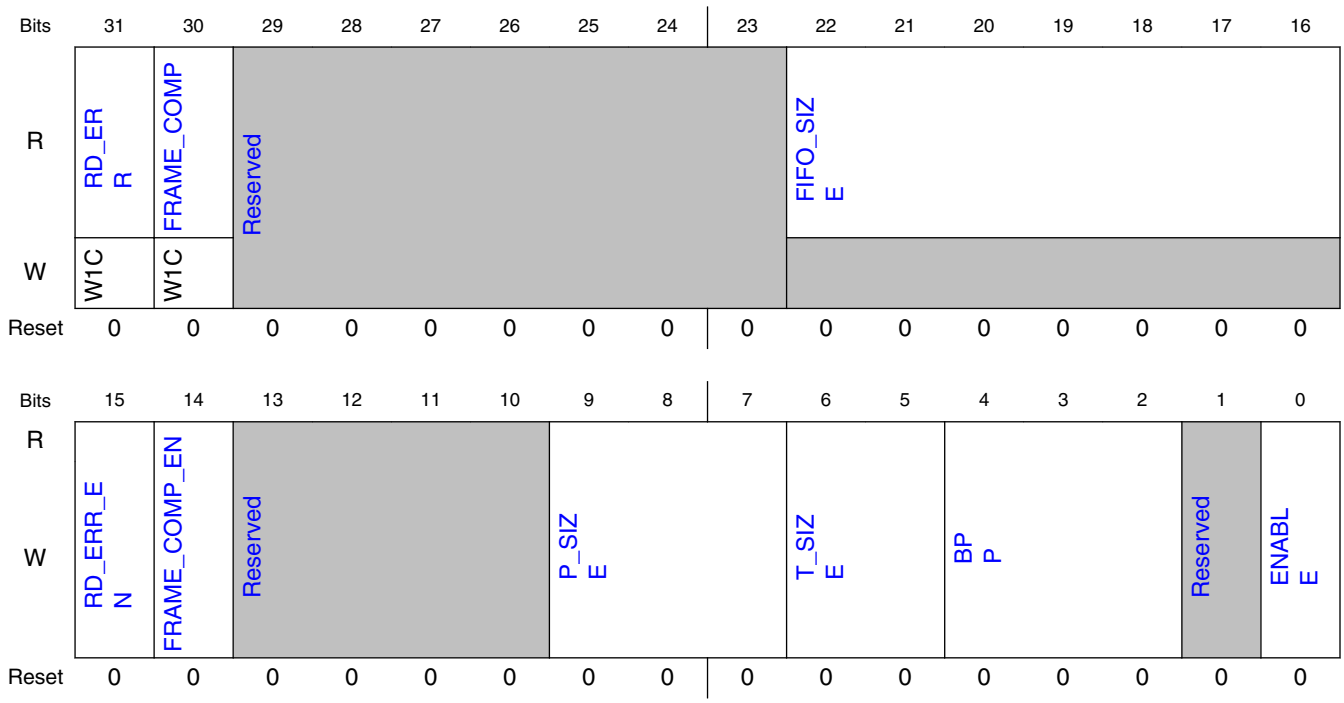
Field	Function
—	
9-7 P_SIZE	Payload size (P_SIZE) Holds the payload size per AXI burst sequence. The P_SIZE should never be programmed to be less than T_SIZE. If the T_SIZE and P_SIZE are both set to 64B, there is a high frequency of 64B requests issued to DRAM. If the P_SIZE is 256B, the requests are issued when there is 256B of space empty in the RD_SRC FIFO. The requests are issued 1/4 as frequently as when the P_SIZE is 64B. With a P_SIZE=256 and T_SIZE=128, two 128B transactions are issued to fullfill the 256B P_SIZE selection. The P_SIZE and T_SIZE should be set to optimize for DRAM efficiency. P_SIZE = T_SIZE = 256B is a good choice to start, but other settings could provide more optimal performance. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • others -> Reserved FIXME this should be based on the BUFF_SZ parameter setting.
6-5 T_SIZE	Transaction Size (T_SIZE) Holds the AXI system bus transaction size per request in bytes. This should always be programmed in conjunction with P_SIZE where P_SIZE >= T_SIZE. • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits/pixel format interface on the input. These settings are intended to reverse the operation of the respective settings in the WR_SCL engine. • 0 -> 40 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 24 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 —	Reserved
0 ENABLE	Read surface enable. Enable bit for read surface bus master. When this bit is set, AND the width and height are both not zero, the RD_SRC engine begins to fetch buffer based on existing programming.

15.13.3.1.3 Control register for Read surface. (CTRL_STATUS_SET)

15.13.3.1.3.1 Offset

Register	Offset
CTRL_STATUS_SET	4h

15.13.3.1.3.2 Diagram



15.13.3.1.3.3 Fields

Field	Function
31 RD_ERR	AXI Read Error Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
30 FRAME_COMP	Frame processing complete Status bit that indicates the current frame processing is complete. This is set even if the height and width are 0. If the respective interrupt enable bit is set, an interrupt is generated.
29-23 —	Reserved
22-16 FIFO_SIZE	FIFO size This is the fifo size implemented in the HW. The P_SIZE should never be set to a value higher than this value in bytes. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
15 RD_ERR_EN	AXI Read Error IRQ enable Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
14 FRAME_COMP_EN	Frame complete IRQ enable Enable an interrupt when the read surface engine is finished processing a frame.
13-10	Reserved

Table continues on the next page...

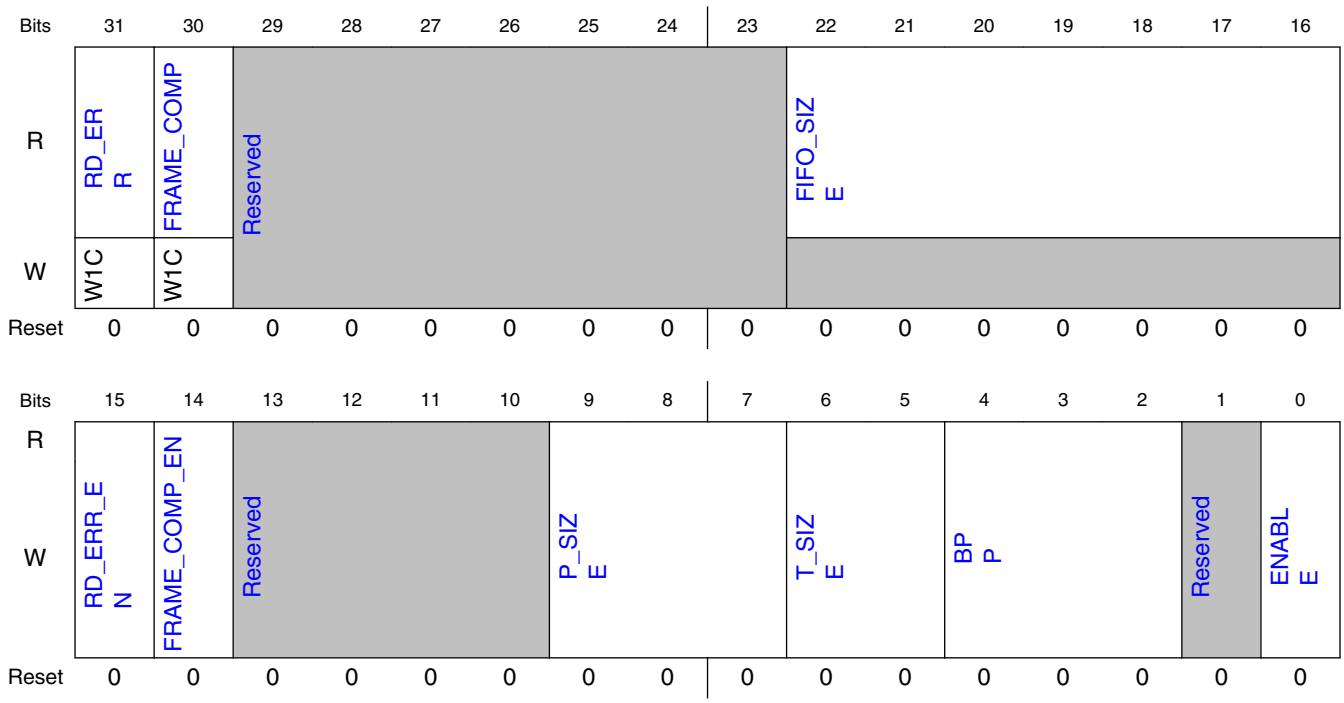
Field	Function
—	
9-7 P_SIZE	Payload size (P_SIZE) Holds the payload size per AXI burst sequence. The P_SIZE should never be programmed to be less than T_SIZE. If the T_SIZE and P_SIZE are both set to 64B, there is a high frequency of 64B requests issued to DRAM. If the P_SIZE is 256B, the requests are issued when there is 256B of space empty in the RD_SRC FIFO. The requests are issued 1/4 as frequently as when the P_SIZE is 64B. With a P_SIZE=256 and T_SIZE=128, two 128B transactions are issued to fullfill the 256B P_SIZE selection. The P_SIZE and T_SIZE should be set to optimize for DRAM efficiency. P_SIZE = T_SIZE = 256B is a good choice to start, but other settings could provide more optimal performance. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • others -> Reserved FIXME this should be based on the BUFF_SZ parameter setting.
6-5 T_SIZE	Transaction Size (T_SIZE) Holds the AXI system bus transaction size per request in bytes. This should always be programmed in conjunction with P_SIZE where P_SIZE >= T_SIZE. • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits/pixel format interface on the input. These settings are intended to reverse the operation of the respective settings in the WR_SCL engine. • 0 -> 40 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 24 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 —	Reserved
0 ENABLE	Read surface enable. Enable bit for read surface bus master. When this bit is set, AND the width and height are both not zero, the RD_SRC engine begins to fetch buffer based on existing programming.

15.13.3.1.4 Control register for Read surface. (CTRL_STATUS_CLR)

15.13.3.1.4.1 Offset

Register	Offset
CTRL_STATUS_CLR	8h

15.13.3.1.4.2 Diagram



15.13.3.1.4.3 Fields

Field	Function
31 RD_ERR	AXI Read Error Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
30 FRAME_COMP	Frame processing complete Status bit that indicates the current frame processing is complete. This is set even if the height and width are 0. If the respective interrupt enable bit is set, an interrupt is generated.
29-23 —	Reserved
22-16 FIFO_SIZE	FIFO size This is the fifo size implemented in the HW. The P_SIZE should never be set to a value higher than this value in bytes. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
15 RD_ERR_EN	AXI Read Error IRQ enable Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
14 FRAME_COMP_EN	Frame complete IRQ enable Enable an interrupt when the read surface engine is finished processing a frame.
13-10	Reserved

Table continues on the next page...

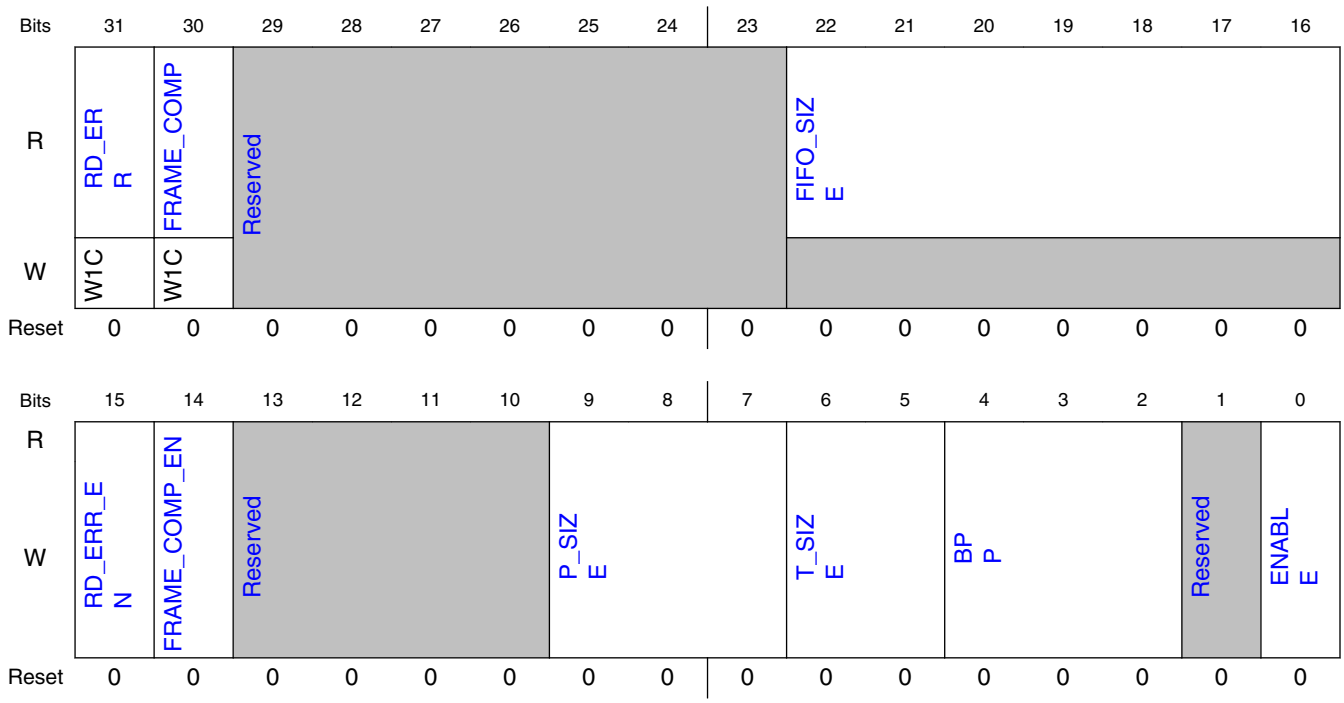
Field	Function
—	
9-7 P_SIZE	Payload size (P_SIZE) Holds the payload size per AXI burst sequence. The P_SIZE should never be programmed to be less than T_SIZE. If the T_SIZE and P_SIZE are both set to 64B, there is a high frequency of 64B requests issued to DRAM. If the P_SIZE is 256B, the requests are issued when there is 256B of space empty in the RD_SRC FIFO. The requests are issued 1/4 as frequently as when the P_SIZE is 64B. With a P_SIZE=256 and T_SIZE=128, two 128B transactions are issued to fullfill the 256B P_SIZE selection. The P_SIZE and T_SIZE should be set to optimize for DRAM efficiency. P_SIZE = T_SIZE = 256B is a good choice to start, but other settings could provide more optimal performance. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • others -> Reserved FIXME this should be based on the BUFF_SZ parameter setting.
6-5 T_SIZE	Transaction Size (T_SIZE) Holds the AXI system bus transaction size per request in bytes. This should always be programmed in conjunction with P_SIZE where P_SIZE >= T_SIZE. • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits/pixel format interface on the input. These settings are intended to reverse the operation of the respective settings in the WR_SCL engine. • 0 -> 40 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 24 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 —	Reserved
0 ENABLE	Read surface enable. Enable bit for read surface bus master. When this bit is set, AND the width and height are both not zero, the RD_SRC engine begins to fetch buffer based on existing programming.

15.13.3.1.5 Control register for Read surface. (CTRL_STATUS_TOG)

15.13.3.1.5.1 Offset

Register	Offset
CTRL_STATUS_TOG	Ch

15.13.3.1.5.2 Diagram



15.13.3.1.5.3 Fields

Field	Function
31 RD_ERR	AXI Read Error Status bit that indicates an AXI read response terminates with a SLV_ERR or DEC_ERR.
30 FRAME_COMP	Frame processing complete Status bit that indicates the current frame processing is complete. This is set even if the height and width are 0. If the respective interrupt enable bit is set, an interrupt is generated.
29-23 —	Reserved
22-16 FIFO_SIZE	FIFO size This is the fifo size implemented in the HW. The P_SIZE should never be set to a value higher than this value in bytes. • 7'h0F - 512B • 7'h1F - 1KB • 7'h3F - 2KB • 7'h7F - 4KB
15 RD_ERR_EN	AXI Read Error IRQ enable Enable an interrupt when the AXI read response completes with a SLV_ERR or DEC_ERR.
14 FRAME_COMP _EN	Frame complete IRQ enable Enable an interrupt when the read surface engine is finished processing a frame.
13-10	Reserved

Table continues on the next page...

Field	Function
—	
9-7 P_SIZE	Payload size (P_SIZE) Holds the payload size per AXI burst sequence. The P_SIZE should never be programmed to be less than T_SIZE. If the T_SIZE and P_SIZE are both set to 64B, there is a high frequency of 64B requests issued to DRAM. If the P_SIZE is 256B, the requests are issued when there is 256B of space empty in the RD_SRC FIFO. The requests are issued 1/4 as frequently as when the P_SIZE is 64B. With a P_SIZE=256 and T_SIZE=128, two 128B transactions are issued to fullfill the 256B P_SIZE selection. The P_SIZE and T_SIZE should be set to optimize for DRAM efficiency. P_SIZE = T_SIZE = 256B is a good choice to start, but other settings could provide more optimal performance. • 000 -> 64B • 001 -> 128B • 010 -> 256B • 011 -> 512B • others -> Reserved FIXME this should be based on the BUFF_SZ parameter setting.
6-5 T_SIZE	Transaction Size (T_SIZE) Holds the AXI system bus transaction size per request in bytes. This should always be programmed in conjunction with P_SIZE where P_SIZE >= T_SIZE. • 00 -> 64B • 01 -> 128B • 10 -> 256B • 11 -> 512B
4-2 BPP	Bits per pixel Holds the number of bits/pixel format interface on the input. These settings are intended to reverse the operation of the respective settings in the WR_SCL engine. • 0 -> 40 bpp, 38 bit components are unpacked • 1 -> 32 bpp, 8-bit to 10-bit upconversion • 2 -> 32 bpp, 10-bit output, alpha(37:30) tied to 0x0 • 3 -> 24 bpp, 10-bit YUV422. pixel_data[37:30]/pixel_data[9:0] tied to 0x0 • 4 -> 16 bpp, 8-bit component, YUV422
1 —	Reserved
0 ENABLE	Read surface enable. Enable bit for read surface bus master. When this bit is set, AND the width and height are both not zero, the RD_SRC engine begins to fetch buffer based on existing programming.

15.13.3.1.6 Read Surface Base address (BASE_ADDR)

15.13.3.1.6.1 Offset

Register	Offset
BASE_ADDR	10h

15.13.3.1.6.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	BASE_ADDR															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.13.3.1.6.3 Fields

Field	Function
31-0 BASE_ADDR	Base address System address of source buffer. Although byte granularity is supported, SW should align it to a 64B boundry to optimize system memory performance.

15.13.3.1.7 Read surface vertical pitch (PITCH)

15.13.3.1.7.1 Offset

Register	Offset
PITCH	14h

15.13.3.1.7.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	PITCH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.13.3.1.7.3 Fields

Field	Function
31-16 —	Reserved
15-0 PITCH	Pitch Number of bytes between 2 vertically adjacent pixels in system memory. Byte granularity is supported, but SW should align to 64B boundry.

15.13.3.1.8 Source frame buffer width (WIDTH)

15.13.3.1.8.1 Offset

Register	Offset
WIDTH	18h

15.13.3.1.8.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	WIDTH															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.13.3.1.8.3 Fields

Field	Function
31-16 —	Reserved
15-0 WIDTH	Width Width, in pixels, of the source frame buffer. This is the number of pixels written to the output buffer pixel interface per scan line.

15.13.3.1.9 Height of frame to be read (HEIGHT)

15.13.3.1.9.1 Offset

Register	Offset
HEIGHT	1Ch

15.13.3.1.9.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	Reserved															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	HEIGHT															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.13.3.1.9.3 Fields

Field	Function
31-16 —	Reserved
15-0 HEIGHT	Height Height, in pixels, of the source frame buffer. This is the number of scan lines written to the output buffer pixel interface per frame.

15.14 Interrupt Request Steering (IRQ_STEER)

15.14.1 Introduction

The irq_steer module redirects/steers the incoming interrupts to output interrupts of a selected/designated channel as specified by a set of configuration registers.

15.14.1.1 Block diagram

The figure below shows a high-level diagram of the irq_steer module.

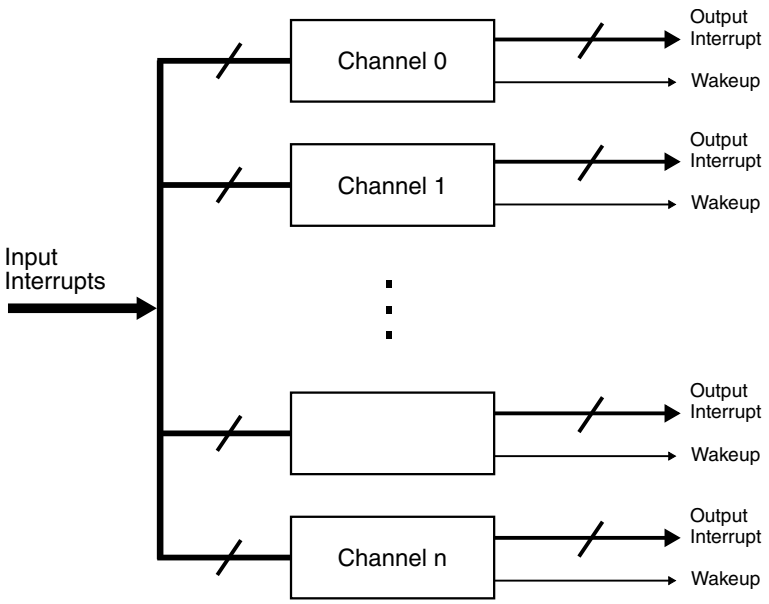


Figure 15-28. irq_steer block diagram

The figure below shows the high-level operation of one channel of the irq_steer module.

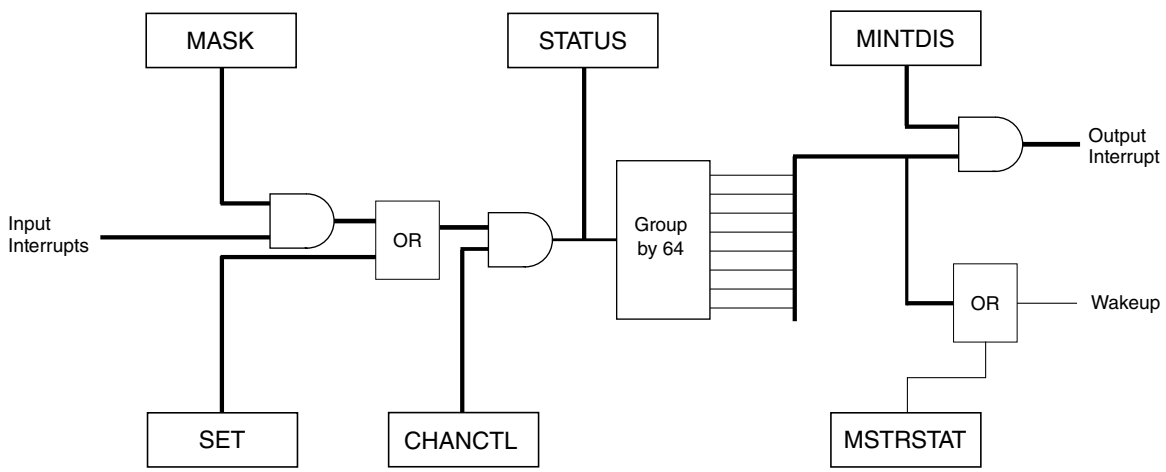


Figure 15-29. irq_steer channel block diagram

15.14.1.2 Features

This irq_steer module supports:

- 5 IRQ channels
- 512 interrupts per channel

15.14.2 Functional description

The irq_steer module allows the user to steer 512 interrupts into one of 5 channels.

15.14.3 Memory Map / Register Definition

This section describes the registers that are implemented for one channel. One channel in this module supports up to 512 interrupts per channel. Each channel has its own set of registers consisting of:

- One CHANnCTL register
- 16 CHn_MASK registers
- 16 CHn_SET registers
- 16 CHn_STATUS registers
- One CHn_MINTDIS register
- One CHn_MSTRSTAT register

This module supports 5 channels. Therefore, there are sets of 5 these registers. Each set of registers will have its own base offset, which must be added to the relative offset shown here.

The following table shows the correlation between an interrupt and its register and bit offset for the CHn_MASK, CHn_SET, and CHn_STATUS registers.

Table 15-10. Interrupt Mapping (incomplete)

n	Bit offset																															
	31	**30**	**29**	**28**	**27**	**26**	**25**	**24**	**23**	**22**	**21**	**20**	**19**	**18**	**17**	**16**	**15**	**14**	**13**	**12**	**11**	**10**	**9**	**8**	**7**	**6**	**5**	**4**	**3**	**2**	**1**	**0**
0	511	510	509	508	507	506	505	504	503	502	501	500	499	498	497	496	495	494	493	492	491	490	489	488	487	486	485	484	483	482	481	480
1	479	478	477	476	475	474	473	472	471	470	469	468	467	466	465	464	463	462	461	460	459	458	457	456	455	454	453	452	451	450	449	448
2	447																															416
3	415																															384
4	383																															384
5	351																															320
6	319																															288
7	287																															256
8	255																															224
9	223																															192
10	191																															160
11	159																															128
12	127																															96
13	95																															64
14	63																															32
15	31																															0

15.14.3.1 IRQSTEER register descriptions

15.14.3.1.1 IRQSTEER Memory map

irq_steer base address: 0h

Offset	Register	Width (In bits)	Access	Reset value
0h	Channel n Control Register (CHANnCTL)	32	RW	0000_0000h
4h - 40h	Channel n Interrupt Mask Register (CHn_MASK0 - CHn_MASK15)	32	RW	0000_0000h
44h - 80h	Channel n Interrupt Set Register (CHn_SET0 - CHn_SET15)	32	RW	0000_0000h
84h - C0h	Channel n Interrupt Status Register (CHn_STATUS0 - CHn_STATUS15)	32	RO	0000_0000h
C4h	Channel n Master Interrupt Disable Register (CHn_MINTDIS)	32	RW	0000_0000h
C8h	Channel n Master Status Register (CHn_MSTRSTAT)	32	RO	0000_0000h

15.14.3.1.2 Channel n Control Register (CHANnCTL)

15.14.3.1.2.1 Offset

Register	Offset
CHANnCTL	0h

15.14.3.1.2.2 Function

This register is used to disable or "zero-out" outputs on a per-channel basis.

NOTE

Only one channel (CH*n* field) may be set at any given time.

15.14.3.1.2.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0											CH4	CH3	CH2	CH1	CH0
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.14.3.1.2.4 Fields

Field	Function
31-5 —	Reserved
4 CH4	Channel 4 control 0b - Disable channel 4 1b - Enable channel 4
3 CH3	Channel 3 control 0b - Disable channel 3 1b - Enable channel 3
2 CH2	Channel 2 control 0b - Disable channel 2 1b - Enable channel 2
1 CH1	Channel 1 control 0b - Disable channel 1 1b - Enable channel 1
0 CH0	Channel 0 control 0b - Disable channel 0 1b - Enable channel 0

15.14.3.1.3 Channel n Interrupt Mask Register (CHn_MASK0 - CHn_MASK 15)

15.14.3.1.3.1 Offset

For a = 0 to 15:

Register	Offset
CHn_MASKa	4h + (a × 4h)

15.14.3.1.3.2 Function

The MASK registers are used to mask any of the 512 individual interrupts.

15.14.3.1.3.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	MASKFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	MASKFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.14.3.1.3.4 Fields

Field	Function
31-0 MASKFLD	Mask bits See Table 15-10 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Mask interrupt 00000000000000000000000000000001b - Do not mask interrupt

15.14.3.1.4 Channel n Interrupt Set Register (CHn_SET0 - CHn_SET15)

15.14.3.1.4.1 Offset

For a = 0 to 15:

Register	Offset
CHn_SETa	44h + (a × 4h)

15.14.3.1.4.2 Function

This register is used to force an interrupt.

15.14.3.1.4.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R W	FORCEFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	FORCEFLD															
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.14.3.1.4.4 Fields

Field	Function
31-0 FORCEFLD	Brief bitfield description. See Table 15-10 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Normal operation 00000000000000000000000000000001b - Force interrupt

15.14.3.1.5 Channel n Interrupt Status Register (CHn_STATUS0 - CHn_STATUS15)

15.14.3.1.5.1 Offset

For a = 0 to 15:

Register	Offset
CHn_STATUSa	84h + (a × 4h)

15.14.3.1.5.2 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	STATUS															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.14.3.1.5.3 Fields

Field	Function
31-0 STATUS	Status of an interrupt See Table 15-10 for the correlation between an interrupt and its register and bit offset. 00000000000000000000000000000000b - Interrupt is not set. 00000000000000000000000000000001b - Interrupt is set.

15.14.3.1.6 Channel n Master Interrupt Disable Register (CHn_MINTDIS)

15.14.3.1.6.1 Offset

Register	Offset
CHn_MINTDIS	C4h

15.14.3.1.6.2 Function

Table 15-11. Interrupt disable mapping

STATUS bit	Interrupts disabled
7	511 - 448
6	447 - 384
5	383 - 320
4	319 - 256
3	255 - 192
2	191 - 128
1	127 - 64
0	63 - 0

15.14.3.1.6.3 Diagram

Bits	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bits	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								DISABLE							
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

15.14.3.1.6.4 Fields

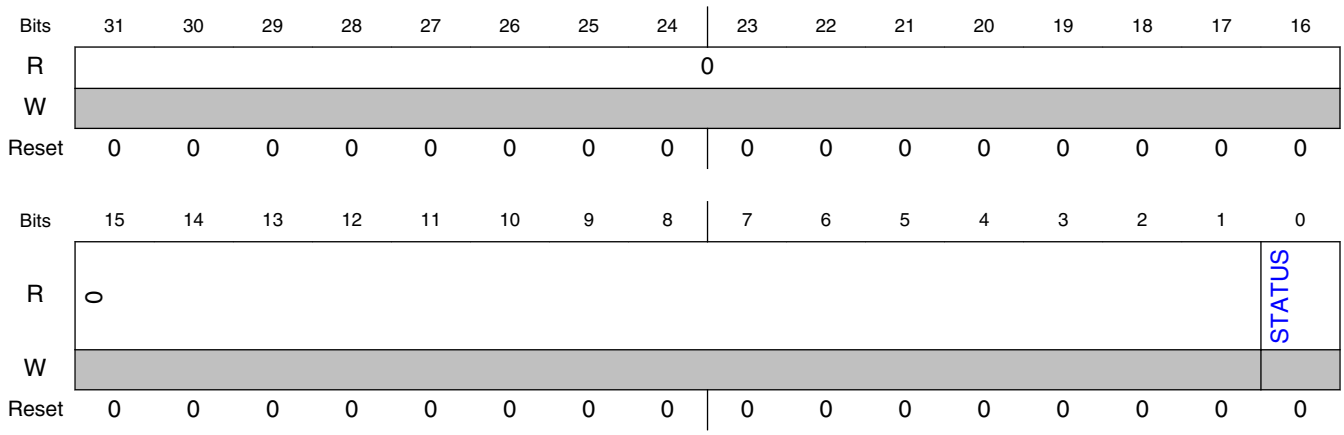
Field	Function
31-8 —	Reserved
7-0 DISABLE	Each bit of this field disables the corresponding interrupts in table above. 00000000b - Enable interrupts 00000001b - Disable interrupts

15.14.3.1.7 Channel n Master Status Register (CHn_MSTRSTAT)

15.14.3.1.7.1 Offset

Register	Offset
CHn_MSTRSTAT	C8h

15.14.3.1.7.2 Diagram



15.14.3.1.7.3 Fields

Field	Function
31-1 —	Reserved
0 STATUS	Status of all interrupts 0b - No interrupts are asserted. 1b - At least one interrupt is asserted.

Chapter 16
Low Speed Communication and Interconnects

16.1 I2C Controller (I2C)

16.1.1 Overview

This chapter describes block-level operation and programming of I2C. The chapter is intended for a block-driver software developer. To understand how the block is integrated at the SoC level, a system software developer should see discussions of the block in the appropriate SoC-level chapter(s).

References: This document assumes an understanding of the following document:

- *The I2C Bus Specification*, Version 2.1, by Philips Semiconductor

The Inter IC (I2C) provides functionality of a standard I2C slave and master. The I2C is designed to be compatible with the standard NXP I2C bus protocol.

NOTE

Four independent I2C channels are available.

I2C is a two-wire, bidirectional serial bus that provides a simple, efficient method of data exchange, minimizing the interconnection between devices. This bus is suitable for applications requiring occasional communications over a short distance between many devices. The flexible I2C standard allows additional devices to be connected to the bus for expansion and system development. See the connection diagram in the figure below.

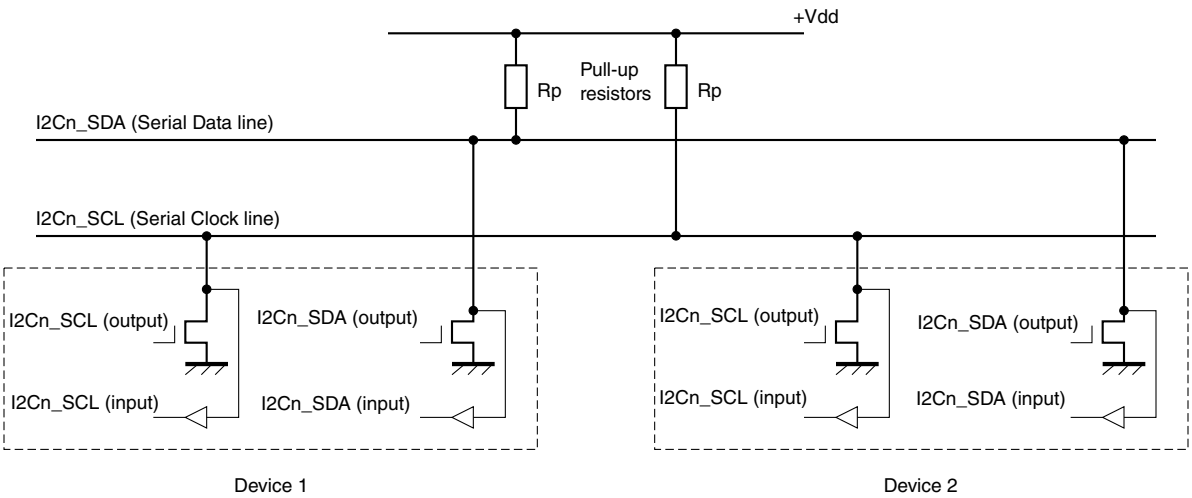


Figure 16-1. Connection of devices to I2C bus

The I2C interface speed is dependent on the I2C bus loading and timing characteristics. For pin requirement details, see *The I2C Bus Specification*. The I2C system is a true multimaster bus including arbitration and collision detection that prevents data corruption if multiple devices attempt to control the bus simultaneously. This feature supports complex applications with multiprocessor control and can be used for rapid testing and alignment of end products through external connections to an assembly-line computer. The figure below shows the block diagram of I2C.

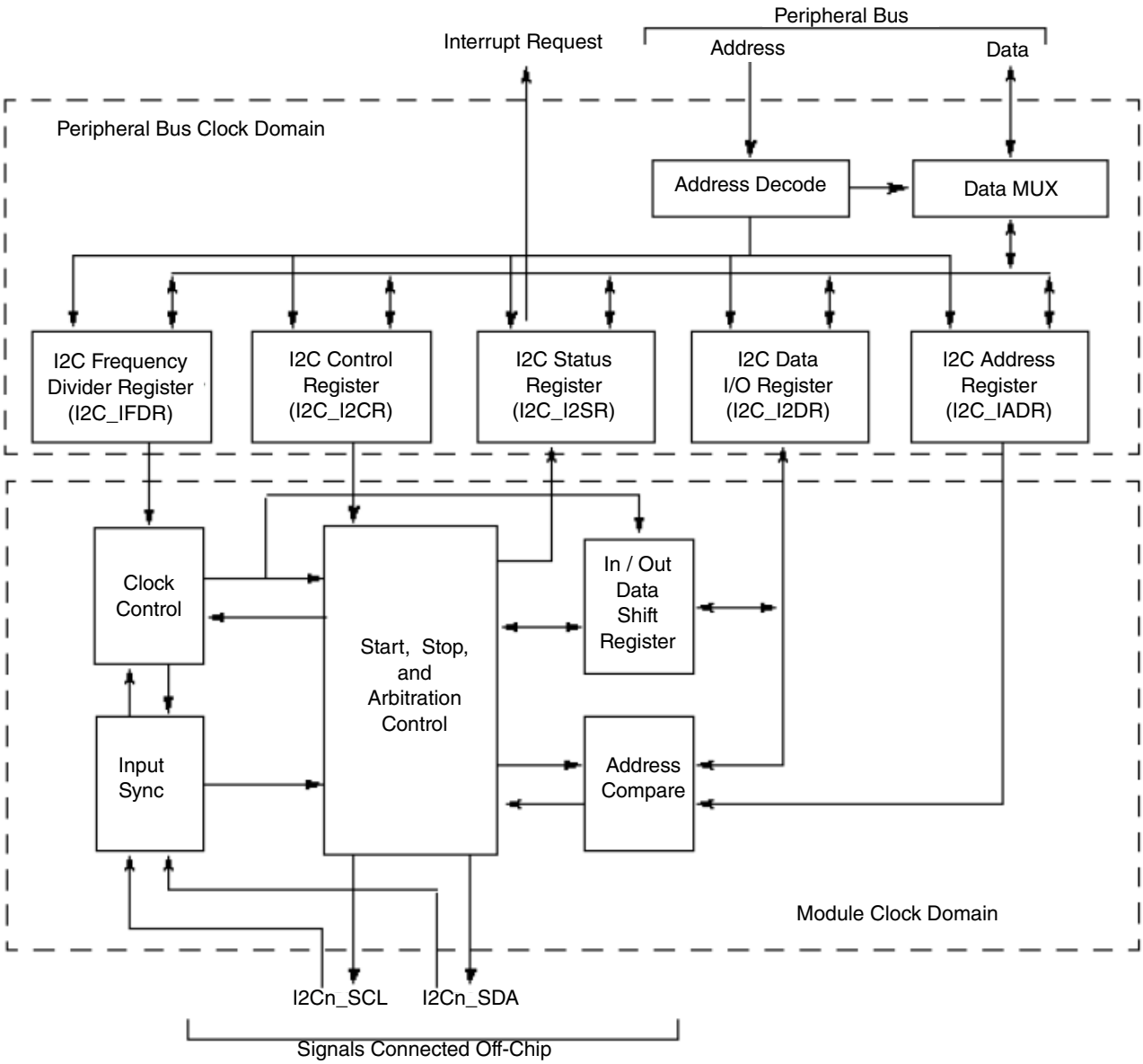


Figure 16-2. I2C block diagram

16.1.1.1 Features

The I2C has the following key features:

- Compatibility with I2C bus standard
- Multimaster operation
- Software programmability for one of 64 different serial clock frequencies
- Software-selectable acknowledge bit
- Interrupt-driven, byte-by-byte data transfer
- Arbitration-lost interrupt with automatic mode switching from master to slave

- Calling address identification interrupt
- Start and stop signal generation/detection
- Repeated Start signal generation
- Acknowledge bit generation/detection
- Bus-busy detection

16.1.1.2 Modes and operations

The I2C operates primarily in two functional modes: Standard mode and Fast mode.

- In Standard mode, I2C supports the data transfer rates up to 100 kbits/s.
- In Fast mode, data transfer rates up to 400 kbits/s can be achieved. Per block operation, there is no special configuration required for Fast or Standard mode. It is the data transfer rate that distinguishes Standard and Fast mode.

16.1.2 External Signals

This section discusses I2C signals that connect off-chip.

For I2C compliance, all devices connected to the I2Cn_SCL and I2Cn_SDA signals must have open-drain or open-collector outputs. The logic AND function is implemented on both lines with external pull-up resistors.

Inputs of I2Cn_SCL and I2Cn_SDA also need to be manually enabled by setting the SION bit in the IOMUX after the corresponding PADs are selected as I2C function.

16.1.3 Clocks

There are two input clocks for I2C.

The following table describes the clock sources for I2C. Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-1. I2C Clocks

Clock name	Clock Root	Description
ipg_clk_patref		Module clock
ipg_clk_s		Peripheral access clock

- Peripheral clock: This clock is used for peripheral bus register read/writes.
- Module clock: This is the functional clock of the I2C. The serial bit clock frequency is derived from the module clock. The module clock and peripheral clocks are synchronous with each other. The minimum frequency of the module clock should be 12.8 MHz for Fast mode to achieve 400-kbps operation.

16.1.4 Functional description

This section provides a complete functional description of the block.

16.1.4.1 I2C system configuration

After a reset, the I2C defaults to Slave Receive operations. Thus, when not operating as a master or responding to a slave transmit address, the I2C defaults to the Slave Receive state.

For exceptions, see Initialization sequence.

NOTE

The I2C is designed to be compatible with the Philips™ I2C bus protocol. For information on system configuration, protocol, and restrictions, see the *I2C Bus Specification*, version 2.1, by Philips Semiconductors. The I2C supports Standard and Fast modes only.

16.1.4.2 Arbitration procedure

If multiple devices simultaneously request the bus, the bus clock is determined by a synchronization procedure in which the low period equals the longest clock-low period among the devices, and the high period equals the shortest. A data arbitration procedure determines the relative priority of competing devices.

A device loses arbitration if it sends logic high while another sends logic low; it immediately switches to Slave Receive mode and stops driving I2Cn_SDA. In this case, the transition from master to Slave mode does not generate a Stop condition. Meanwhile, hardware sets the arbitration lost bit in the I2C Status register (I2C_I2SR[IAL] to indicate loss of arbitration).

16.1.4.3 Clock synchronization

Because wire-AND logic is used, a high-to-low transition on SCL affects devices connected to the bus. Devices start counting their low period when the master drives SCL low. When a device clock goes low, it holds SCL low until the Clock High state is reached. However, the low-to-high change in this device clock may not change the state of SCL if another device clock is still in its low period. Therefore, the device with the longest low period holds the synchronized clock SCL low.

Devices with shorter low periods enter a High Wait state during this time (see Figure 16-3). When all devices involved have counted off their low periods, the synchronized clock SCL is released and pulled high. There is then no difference between device clocks and the state of SCL, so all of the devices start counting their high periods. The first device to complete its high period pulls SCL low again.

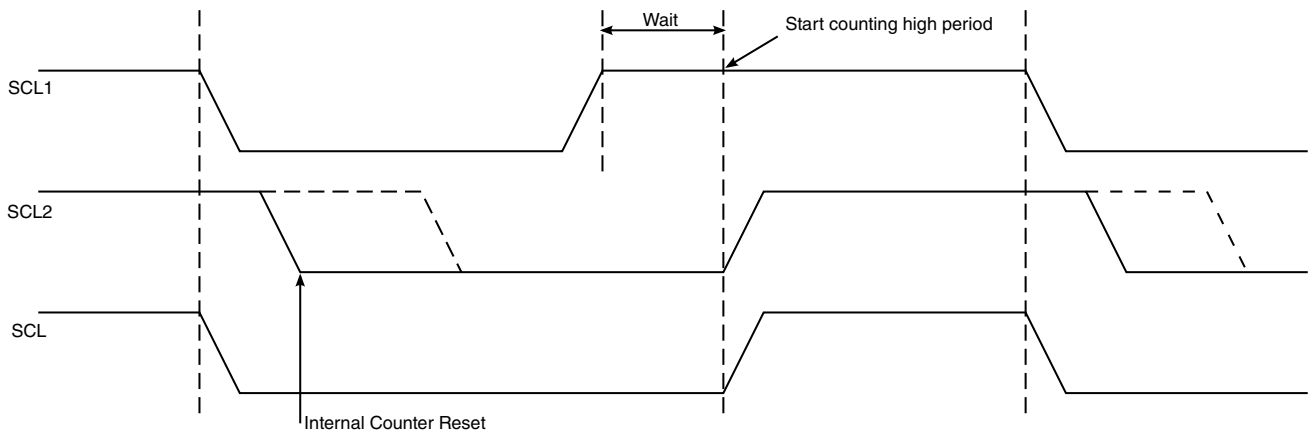


Figure 16-3. Synchronized clock SCL

16.1.4.4 Handshaking

The clock synchronization mechanism can be used as a handshake in data transfers. Slave devices can hold SCL low after completing one byte transfer (9 bits). In such a case, the clock mechanism halts the bus clock and forces the master clock into a Wait state until the slave releases SCL.

16.1.4.5 Clock stretching

Slaves can use the clock synchronization mechanism to slow down the transfer bit rate. After the master has driven SCL low, the slave can drive SCL low for the required period and then release it. If the slave SCL low period is longer than the master SCL low period, the resulting SCL bus signal low period is stretched.

16.1.4.6 Peripheral bus accesses

I2C is a 16-bit block. Only half-word accesses should be performed to the block.

16.1.4.7 Generation of transfer error on IP bus

If an address is received on the peripheral slave bus interface but it is not implemented, an access error is generated.

16.1.4.8 Reset

The I2C can be reset in the following ways:

- Global reset: A hard asynchronous reset of the whole I2C
- Software reset: An internal reset for the whole I2C (except for I2C_IADR and I2C_IFDR registers) initiated by deasserting the I2C_I2CR[IEN] bit

16.1.4.9 Interrupts

There is only one interrupt from the block, which is enabled by setting the I2C_I2CR[IIEN] bit.

The interrupt is generated in any one of the following conditions:

- One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock).
- An address is received that matches its own specific address in Slave Receive mode.
- Arbitration is lost.

16.1.4.10 Byte order

The block only supports the Little-Endian mode.

16.1.5 Initialization

NOTE

Ensure the input select pins for IOMUXC are configured correctly for I2C.

16.1.5.1 Initialization sequence

Before the interface can transfer serial data, registers must be initialized, as listed here.

1. Set the data sampling rate (I2C_IFDR[IC] to obtain SCL frequency from the system bus clock.
2. Update the address in the (I2C_IADR) to define its slave address (address can range from 0 to 0x7f).
3. Set the I2C enable bit (I2C_I2CR[IEN]) to enable the I2C bus interface system.
4. Modify the bits in the I2C_I2CR to select Master/Slave mode, Transmit/Receive mode, and Interrupt-Enable or not.

16.1.5.2 Generation of Start

After completion of the initialization procedure, serial data can be transmitted by selecting the Master Transmit mode. On a multimaster bus system, the busy bus (I2C_I2SR[IBB]) must be tested to determine whether the serial bus is free. If the bus is free (IBB = 0), the Start signal and the first byte (the slave address) can be sent. The data written to the data register comprises the address of the desired slave and the LSB indicates the transfer direction.

The free time between a Stop and the next Start condition is built into the hardware that generates the Start cycle. Depending on the relative frequencies of the system clock and the SCL period, it may be necessary to wait until the I2C is not busy after writing the calling address to the data register (I2C_I2DR), before proceeding to load data into the data register (I2C_I2DR).

16.1.5.3 Post-transfer software response

Sending or receiving a byte sets the data transferring bit (I2C_I2SR[ICF]), which indicates one byte of communication is finished. Upon completion, the interrupt status (I2C_I2SR[IIF]) is also set. An external interrupt is generated if the interrupt enable (I2C_I2CR[IIEN]) is set. The software must first clear the interrupt status (I2C_I2SR[IIF]) in the interrupt routine.

See the flow chart in Figure 16-5.

The data transferring bit (I2C_I2SR[ICF]) is cleared either by reading from I2C_I2DR in Receive mode or by writing to this register in Transmit mode.

The software can service the I2C I/O in the main program by monitoring the interrupt status (I2C_I2SR[IIF]) if the interrupt enable is deasserted. In this case, the interrupt status should be polled in the data transferring bit (I2C_I2SR[ICF]) because the operation is different when arbitration is lost.

When an interrupt occurs at the end of the address cycle, the master is always in Transmit mode; that is, the address is sent. If Master Receive mode is required, then I2C_I2CR[MTX] should be toggled and a dummy read of the I2C_I2DR register must be executed to trigger receive data.

During Slave-mode address cycles (I2C_I2SR[IAAS] = 1), the slave read/write bit I2C_I2SR[SRW] is read to determine the direction of the next transfer. The transmit/receive bit (I2C_I2CR[MTX]) should also be programmed accordingly. For Slave-mode data cycles (IAAS = 0), SRW is invalid. MTX should be read to determine the current transfer direction.

16.1.5.4 Generation of Stop

A data transfer ends when the master signals a Stop, which can occur after all data is sent.

For a master receiver to terminate a data transfer, it must inform the slave transmitter by not acknowledging the last data byte. This is done by setting the transmit acknowledge bit (I2C_I2CR[TXAK]) before reading the next-to-last byte. Before the last byte is read, a Stop signal must be generated.

16.1.5.5 Generation of Repeated Start

After the data transfer, if the master still requires the bus, it can signal another Start followed by another slave address without signaling a Stop.

16.1.5.6 Slave mode

In the slave interrupt service routine (see Figure 16-5), the block addressed as slave bit (IAAS) should be tested to check if a calling of its own address has just been received. If IAAS is set, software should set the Transmit/Receive mode select bit (I2C_I2CR[MTX]) according to the I2C_I2SR[SRW]. Writing to the I2C_I2CR clears the IAAS automatically. The only time IAAS is read as set is from the interrupt at the end of the address cycle where an address match occurred; interrupts resulting from subsequent data transfers will have IAAS cleared. A data transfer can now be initiated by writing information to I2C_I2DR for slave transmits, or read from I2C_I2DR in Slave Receive mode. A dummy read of I2C_I2DR in Slave Receive mode releases SCL, allowing the master to send data.

In the slave transmitter routine, the receive acknowledge bit (I2C_I2SR[RXAK]) must be tested before sending the next byte of data. Setting RXAK means an end-of-data signal from the master receiver, after which the software must switch it from Transmit to Receiver mode. Reading the data register (I2C_I2DR) then releases SCL so the master can generate a Stop signal.

16.1.5.7 Arbitration lost

If several devices try to engage the bus at the same time, one becomes master. Hardware immediately switches devices that lose arbitration to Slave Receive mode. Data output to 12Cn_SDA stops, but 12Cn_SCL is still generated until the end of the byte during which arbitration is lost. An interrupt occurs at the falling edge of the ninth clock of this transfer if the arbitration is lost (I2C_I2SR[IAL] = 1), and the Slave mode is selected (I2C_I2CR[MSTA] = 0).

See the flow chart in Figure 16-5.

If a device that is not a master tries to transmit or do a Start, hardware inhibits the transmission, clears MSTA without signaling a Stop, generates an interrupt to the Arm platform, and sets I2C_I2SR[IAL] to indicate a failed attempt to engage the bus. When considering these cases, the slave service routine should first test I2C_I2SR[IAL], and the software should clear it if it is set.

For Multimaster mode, when an I2C is enabled when the bus is busy and asserts Start, the I2C_I2SR[IAL] bit gets set only for 12Cn_SDA=0, 12Cn_SCL=0/1, 12Cn_SDA=1, and 12Cn_SCL=0; but not for 12Cn_SDA=1 and I2Cn_SCL=1, which is the equivalent of Bus Idle state.

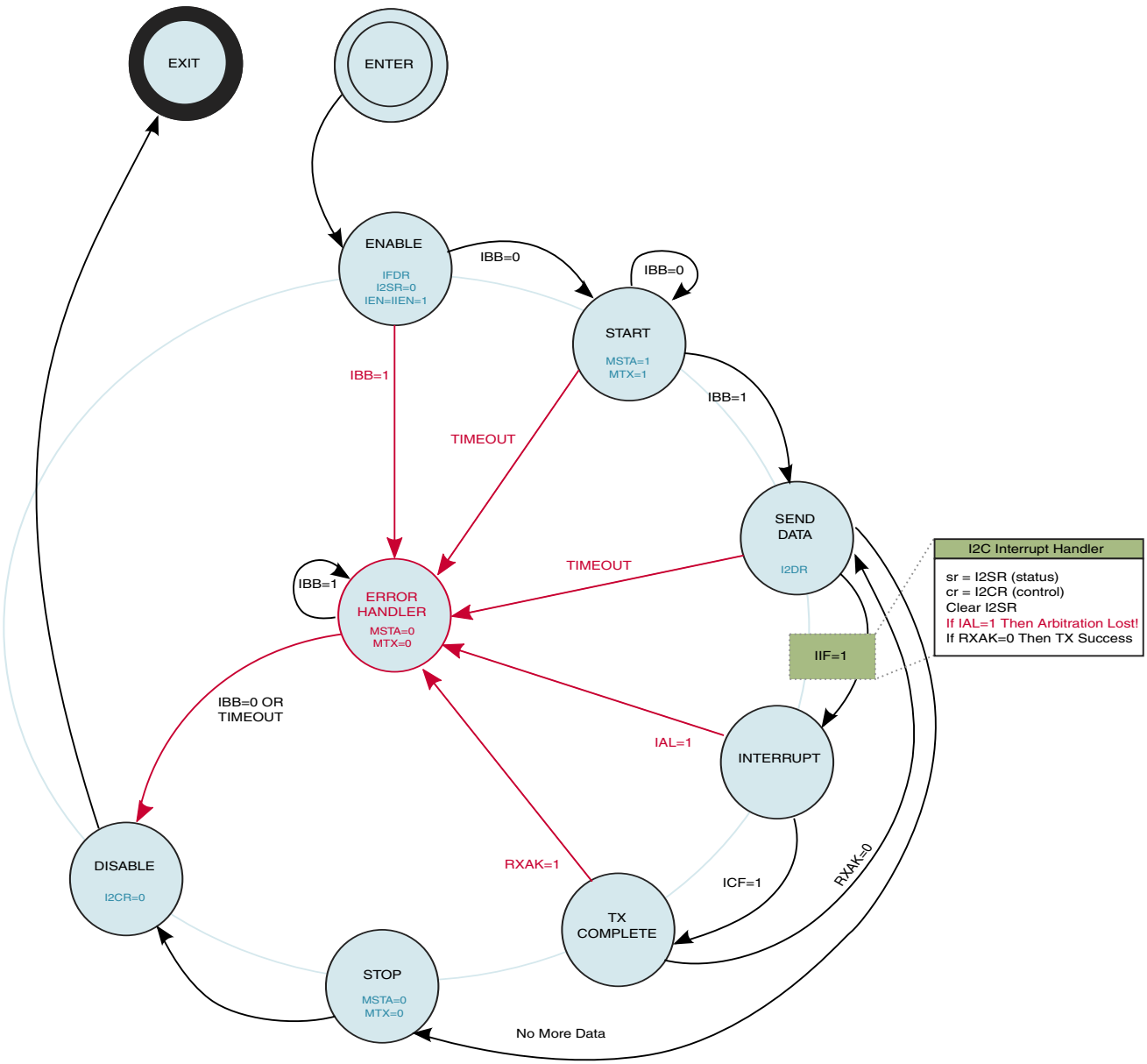


Figure 16-4. I2C Programming state diagram

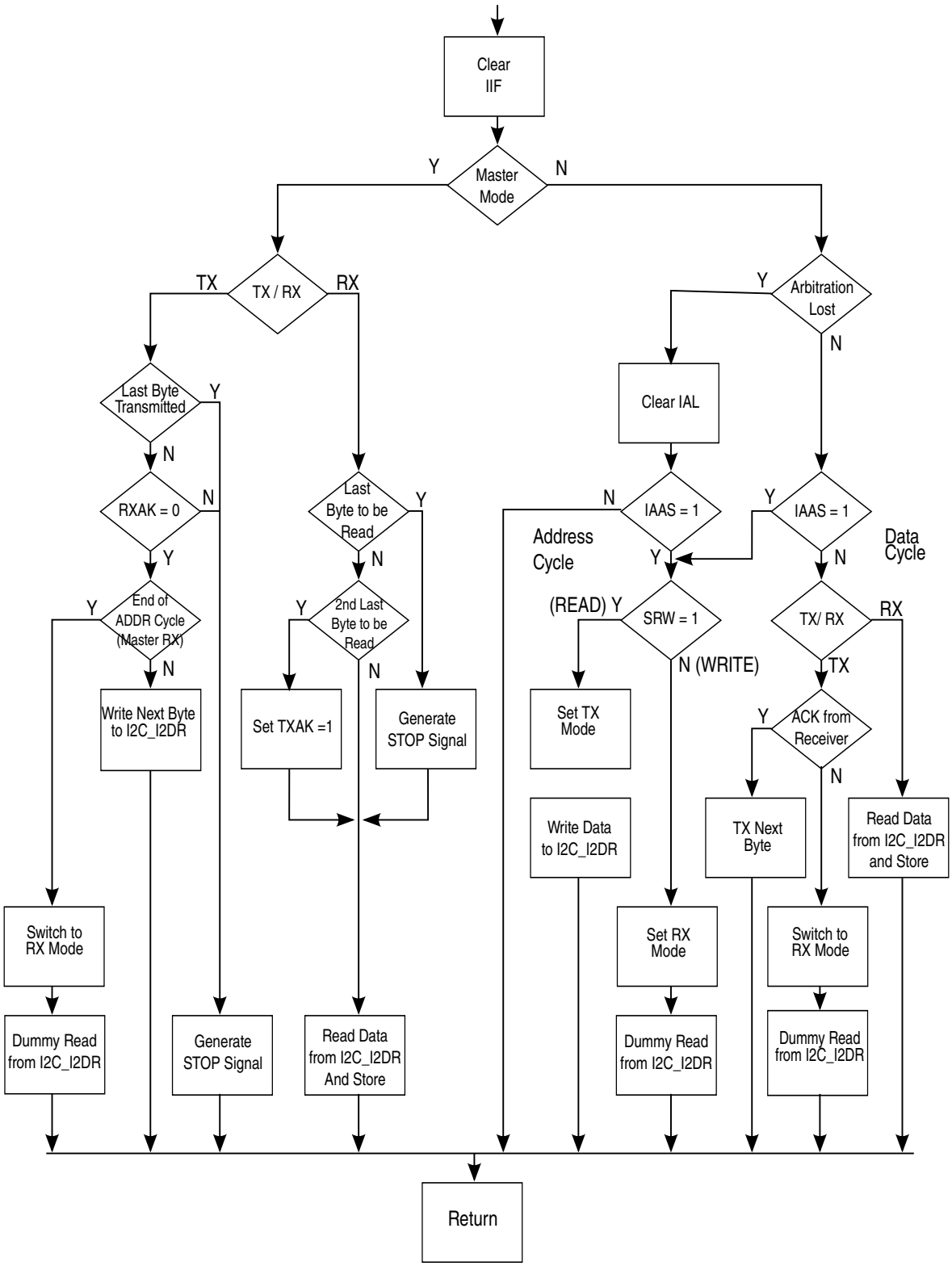


Figure 16-5. Flowchart of typical I2C interrupt routine

NOTE

For a Repeated Start only, the Stop-generation stage does not occur in Master mode. A loop repeats itself without stopping for the next start.

For Master Receive mode, I2C is programmed as Master Transmit during Address mode and after slave address transfer; the MTX bit should be cleared and a dummy read on the I2C_I2DR register should be performed so I2C can read the next receive data.

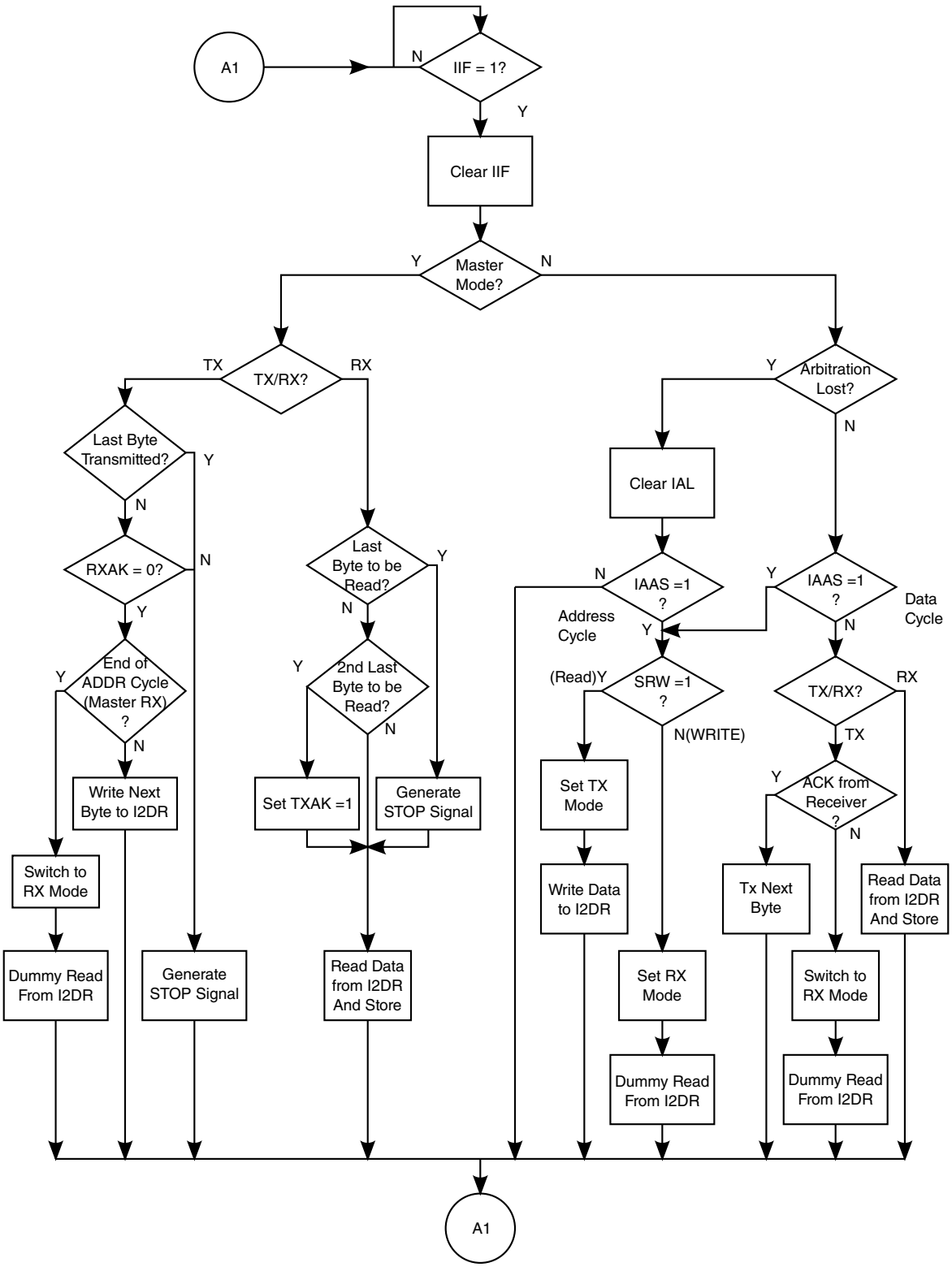


Figure 16-6. Flowchart for typical I2C polling routine

NOTE

The timeout value depends on the bus frequency at which I2C is operating. The minimum timeout for polling the IIF bit at a maximum I2C bus frequency of 400 kHz is T_{min} = 25 µs (=2.5 x 10 µs). This value can be calculated for any bus frequency. The formula is $T_{min} = 10/F_{SCL}$, where F_{SCL} is the frequency of the I2C clock (SCL).

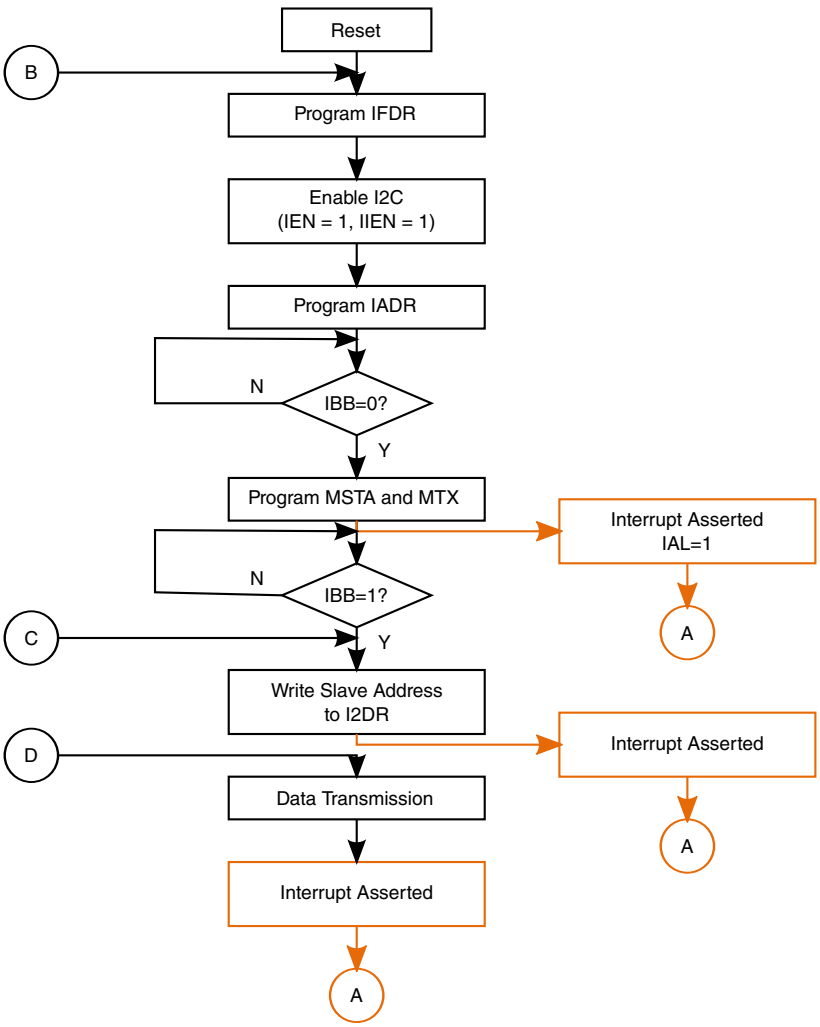


Figure 16-7. Detailed flowchart of a typical I2C Master Transmit mode, part 1

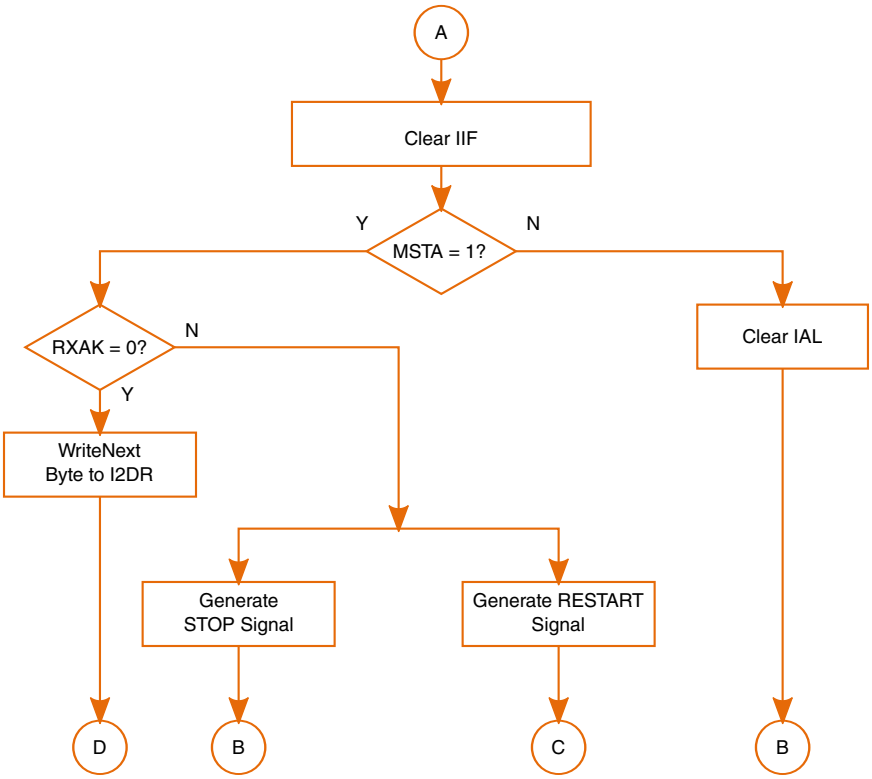


Figure 16-8. Detailed flowchart of a typical I2C Master Transmit mode, part 2

Figure 16-7 and Figure 16-8 show the Master Transmit mode operation with interrupt subroutine. If an interrupt is generated and the MSTA bit is 0, then bus arbitration is lost and IAL is set. Software can clear the IAL bit and reprogram I2C. If the MSTA bit is 1, then it is a transfer-generated interrupt. In this case, software can check the RXAK bit for a data receive acknowledgement by the slave and, accordingly, decide to do one of the following:

- Generate a STOP
- Generate a REPEATED START by writing to the I2C_I2CR register
- Perform the next data transfer by writing to the I2C_I2DR register

NOTE

The IBB bit is asserted by a Start condition on the bus, and it is deasserted by a Stop condition on the bus. Therefore, if arbitration is lost due to an unexpected Stop condition during transfer, then IBB is cleared. If arbitration is lost due to a data mismatch, then it is not cleared. Software should always clear the IEN bit and then set it if arbitration is lost.

16.1.6 Software restriction

Software should ensure that there is a delay of at least two module clock cycles after it sets the I2C_I2CR[RSTA] bit and before writing to the I2C_I2DR register. The maximum possible clock period of the module clock is 78 ns.

16.1.7 I2C Memory Map/Register Definition

The I2C contains five 16-bit registers.

NOTE

Registers at offsets 0x0002, 0x0006, 0x000A, and 0x000E are reserved for future additions.

I2C memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A2_0000	I2C Address Register (I2C1_IADR)	16	R/W	0000h	16.1.7.1/ 6280
30A2_0004	I2C Frequency Divider Register (I2C1_IFDR)	16	R/W	0000h	16.1.7.2/ 6280
30A2_0008	I2C Control Register (I2C1_I2CR)	16	R/W	0000h	16.1.7.3/ 6282
30A2_000C	I2C Status Register (I2C1_I2SR)	16	R/W	0081h	16.1.7.4/ 6283
30A2_0010	I2C Data I/O Register (I2C1_I2DR)	16	R/W	0000h	16.1.7.5/ 6285
30A3_0000	I2C Address Register (I2C2_IADR)	16	R/W	0000h	16.1.7.1/ 6280
30A3_0004	I2C Frequency Divider Register (I2C2_IFDR)	16	R/W	0000h	16.1.7.2/ 6280
30A3_0008	I2C Control Register (I2C2_I2CR)	16	R/W	0000h	16.1.7.3/ 6282
30A3_000C	I2C Status Register (I2C2_I2SR)	16	R/W	0081h	16.1.7.4/ 6283
30A3_0010	I2C Data I/O Register (I2C2_I2DR)	16	R/W	0000h	16.1.7.5/ 6285
30A4_0000	I2C Address Register (I2C3_IADR)	16	R/W	0000h	16.1.7.1/ 6280
30A4_0004	I2C Frequency Divider Register (I2C3_IFDR)	16	R/W	0000h	16.1.7.2/ 6280

Table continues on the next page...

I2C memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A4_0008	I2C Control Register (I2C3_I2CR)	16	R/W	0000h	16.1.7.3/ 6282
30A4_000C	I2C Status Register (I2C3_I2SR)	16	R/W	0081h	16.1.7.4/ 6283
30A4_0010	I2C Data I/O Register (I2C3_I2DR)	16	R/W	0000h	16.1.7.5/ 6285
30A5_0000	I2C Address Register (I2C4_IADR)	16	R/W	0000h	16.1.7.1/ 6280
30A5_0004	I2C Frequency Divider Register (I2C4_IFDR)	16	R/W	0000h	16.1.7.2/ 6280
30A5_0008	I2C Control Register (I2C4_I2CR)	16	R/W	0000h	16.1.7.3/ 6282
30A5_000C	I2C Status Register (I2C4_I2SR)	16	R/W	0081h	16.1.7.4/ 6283
30A5_0010	I2C Data I/O Register (I2C4_I2DR)	16	R/W	0000h	16.1.7.5/ 6285

16.1.7.1 I2C Address Register (I2Cx_IADR)

Address: Base address + 0h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								ADR							0
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_IADR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7–1 ADR	Slave address. Contains the specific slave address to be used by the I2C. Slave mode is the default I2C mode for an address match on the bus. **NOTE:** The I2C_IADR holds the address to which the I2C responds when addressed as a slave. The slave address is not the address sent on the bus during the address transfer. The register is not reset by a software reset.
0 Reserved	This read-only field is reserved and always has the value 0.

16.1.7.2 I2C Frequency Divider Register (I2Cx_IFDR)

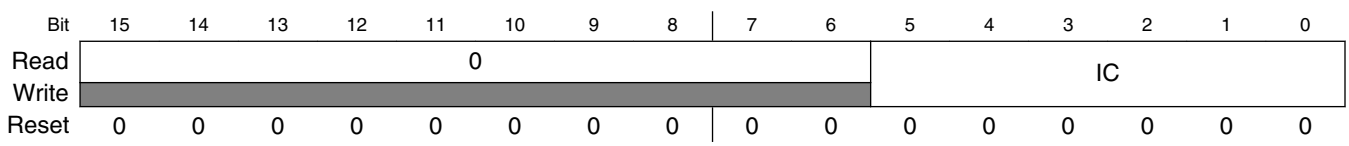
The I2C_IFDR provides a programmable prescaler to configure the clock for bit-rate selection. The register does not get reset by a software reset.

The following table describes the divider and register values for the register field "IC."

Table 16-2. I2C_IFDR Register Field Values

IC	Divider		IC	Divider		IC	Divider		IC	Divider
0x00	30		0x10	288		0x20	22		0x30	160
0x01	32		0x11	320		0x21	24		0x31	192
0x02	36		0x12	384		0x22	26		0x32	224
0x03	42		0x13	480		0x23	28		0x33	256
0x04	48		0x14	576		0x24	32		0x34	320
0x05	52		0x15	640		0x25	36		0x35	384
0x06	60		0x16	768		0x26	40		0x36	448
0x07	72		0x17	960		0x27	44		0x37	512
0x08	80		0x18	1152		0x28	48		0x38	640
0x09	88		0x19	1280		0x29	56		0x39	768
0x0A	104		0x1A	1536		0x2A	64		0x3A	896
0x0B	128		0x1B	1920		0x2B	72		0x3B	1024
0x0C	144		0x1C	2304		0x2C	80		0x3C	1280
0x0D	160		0x1D	2560		0x2D	96		0x3D	1536
0x0E	192		0x1E	3072		0x2E	112		0x3E	1792
0x0F	240		0x1F	3840		0x2F	128		0x3F	2048

Address: Base address + 4h offset



I2Cx_IFDR field descriptions

Field	Description
15–6 Reserved	This read-only field is reserved and always has the value 0.
IC	I2C clock rate. Prescales the clock for bit-rate selection. Due to potentially slow I2Cn_SCL and I2Cn_SDA rise and fall times, bus signals are sampled at the prescaler frequency. The serial bit clock frequency may be lower than IPG_CLK_ROOT divided by the divider shown in the I2C Data I/O Register. **NOTE:** The IC value should not be changed during the data transfer, however, it can be changed before a Repeat Start or Start programming sequence in I2C. The I2C protocol supports bit rates of up to 400 kbps. The IC bits need to be programmed in accordance with this constraint.

16.1.7.3 I2C Control Register (I2Cx_I2CR)

The I2C_I2CR is used to enable the I2C and the I2C interrupt. It also contains bits that govern operation as a slave or a master.

Address: Base address + 8h offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	IEN	IIEN	MSTA	MTX	TXAK	0	0	
Write						RSTA		
Reset	0	0	0	0	0	0	0	0

I2Cx_I2CR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 IEN	I2C enable. Also controls the software reset of the entire I2C. Resetting the bit generates an internal reset to the block. If the block is enabled in the middle of a byte transfer, Slave mode ignores the current bus transfer and starts operating when the next Start condition is detected. Master mode is not aware that the bus is busy, so initiating a start cycle may corrupt the current bus cycle, ultimately causing either the current master or the I2C to lose arbitration. Subsequently, bus operation returns to normal. 0 The block is disabled, but registers can still be accessed. 1 The I2C is enabled. This bit must be set before any other I2C_I2CR bits have an effect.
6 IIEN	I2C interrupt enable. **NOTE:** If data is written during the Start condition, that is, just after setting the I2C_I2CR[MSTA] and I2C_I2CR[MTX] bits, then the ICF bit is cleared at the falling edge of SCLK after Start. If data is written after the Start condition and falling edge of SCLK, then the ICF bit is cleared as soon as data is written. 0 I2C interrupts are disabled, but the status flag I2C_I2SR[IIF] continues to be set when an Interrupt condition occurs. 1 I2C interrupts are enabled. An I2C interrupt occurs if I2C_I2SR[IIF] is also set.
5 MSTA	Master/Slave mode select bit. If the master loses arbitration, MSTA is cleared without generating a Stop signal. **NOTE:** The module clock should be on for writing to the MSTA bit. **NOTE:** The MSTA bit is cleared by software to generate a Stop condition; it can also be cleared by hardware when the I2C loses the bus arbitration.

Table continues on the next page...

I2Cx_I2CR field descriptions (continued)

Field	Description
	0 Slave mode. Changing MSTA from 1 to 0 generates a Stop and selects Slave mode. 1 Master mode. Changing MSTA from 0 to 1 signals a Start on the bus and selects Master mode.
4 MTX	Transmit/Receive mode select bit. Selects the direction of master and slave transfers. 0 Receive. When a slave is addressed, the software should set MTX according to the slave read/write bit in the I2C status register (I2C_I2SR[SRW]). 1 Transmit. In Master mode, MTX should be set according to the type of transfer required. Therefore, for address cycles, MTX is always 1.
3 TXAK	Transmit acknowledge enable. Specifies the value driven onto I2Cn_SDA during acknowledge cycles for both master and slave receivers. **NOTE:** Writing TXAK applies only when the I2C bus is a receiver. 0 An acknowledge signal is sent to the bus at the ninth clock bit after receiving one byte of data. 1 No acknowledge signal response is sent (that is, the acknowledge bit = 1).
2 RSTA	Repeat start. Always reads as 0. Attempting a repeat start without bus mastership causes loss of arbitration. 0 No repeat start 1 Generates a Repeated Start condition
Reserved	This read-only field is reserved and always has the value 0.

16.1.7.4 I2C Status Register (I2Cx_I2SR)

The I2C_I2SR contains bits that indicate transaction direction and status.

Address: Base address + Ch offset

Bit	15	14	13	12	11	10	9	8
Read	0							
Write								
Reset	0	0	0	0	0	0	0	0

Bit	7	6	5	4	3	2	1	0
Read	ICF	IAAS	IBB	IAL	0	SRW	IIF	RXAK
Write								
Reset	1	0	0	0	0	0	0	1

I2Cx_I2SR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
7 ICF	Data transferring bit. While one byte of data is transferred, ICF is cleared. 0 Transfer is in progress. 1 Transfer is complete. This bit is set by the falling edge of the ninth clock of the last byte transfer.
6 IAAS	I2C addressed as a slave bit. The Arm platform is interrupted if the interrupt enable (I2C_I2CR[IIEN]) is set. The Arm platform must check the slave read/write bit (SRW) and set its Transfer/Receive mode accordingly. Writing to I2C_I2CR clears this bit. 0 Not addressed 1 Addressed as a slave. Set when its own address (I2C_IADR) matches the calling address.
5 IBB	I2C bus busy bit. Indicates the status of the bus. **NOTE:** When I2C is enabled (I2C_I2CR[IEN] = 1), it continuously polls the bus data (SDA) and clock (SCL) signals to determine a Start or Stop condition. 0 Bus is idle. If a Stop signal is detected, IBB is cleared. 1 Bus is busy. When Start is detected, IBB is set.
4 IAL	Arbitration lost. Set by hardware in the following circumstances (IAL must be cleared by software by writing a "0" to it at the start of the interrupt service routine): • I2Cn_SDA input samples low when the master drives high during an address or data-transmit cycle. • I2Cn_SDA input samples low when the master drives high during the acknowledge bit of a data-receive cycle. For the above two cases, the bit is set at the falling edge of the ninth I2Cn_SCL clock during the ACK cycle. • A Start cycle is attempted when the bus is busy. • A Repeated Start cycle is requested in Slave mode. • A Stop condition is detected when the master did not request it. **NOTE:** Software cannot set the bit. 0 No arbitration lost. 1 Arbitration is lost.
3 Reserved	This read-only field is reserved and always has the value 0.
2 SRW	Slave read/write. When the I2C is addressed as a slave, IAAS is set, and the slave read/write bit (SRW) indicates the value of the R/W command bit of the calling address sent from the master. SRW is valid only when a complete transfer has occurred, no other transfers have been initiated, and the I2C is a slave and has an address match. 0 Slave receive, master writing to slave 1 Slave transmit, master reading from slave
1 IIF	I2C interrupt. Must be cleared by the software by writing a "0" to it in the interrupt routine. **NOTE:** The software cannot set the bit. 0 No I2C interrupt pending. 1 An interrupt is pending.

Table continues on the next page...

I2Cx_I2SR field descriptions (continued)

Field	Description
	This causes a processor interrupt request (if the interrupt enable is asserted [IIEN = 1]). The interrupt is set when one of the following occurs: • One byte transfer is completed (the interrupt is set at the falling edge of the ninth clock). • An address is received that matches its own specific address in Slave Receive mode. • Arbitration is lost.
0 RXAK	Received acknowledge. This is the value received from the I2Cn_SDA input for the acknowledge bit during a bus cycle. 0 An "acknowledge" signal was received after the completion of an 8-bit data transmission on the bus. 1 A "No acknowledge" signal was detected at the ninth clock.

16.1.7.5 I2C Data I/O Register (I2Cx_I2DR)

In Master Receive mode, reading the data register allows a read to occur and initiates the next byte to be received. In Slave mode, the same function is available after it is addressed.

Address: Base address + 10h offset

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Read	0								DATA							
Write																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

I2Cx_I2DR field descriptions

Field	Description
15–8 Reserved	This read-only field is reserved and always has the value 0.
DATA	Data Byte. Holds the last data byte received or the next data byte to be transferred. Software writes the next data byte to be transmitted or reads the data byte received. **NOTE:** The core-written value in I2C_I2DR cannot be read back by the core. Only data written by the I2C bus side can be read.

16.2 Universal Asynchronous Receiver/Transmitter(UART)

16.2.1 Overview

Universal Asynchronous Receiver/Transmitter (UART) provides serial communication capability with external devices through a level converter and an RS-232 cable or through use of external circuitry that converts infrared signals to electrical signals (for reception) or transforms electrical signals to signals that drive an infrared LED (for transmission) to provide low speed IrDA compatibility.

UART supports NRZ encoding format , RS485 compatible 9 bit data format and IrDA-compatible infrared slow data rate (SIR) format.

Figure 16-9 is the UART block diagram.

The "Module Clock" is the UART_CLK which comes from CCM. The "Peripheral Clock" is the IPG_CLK which comes from CCM.

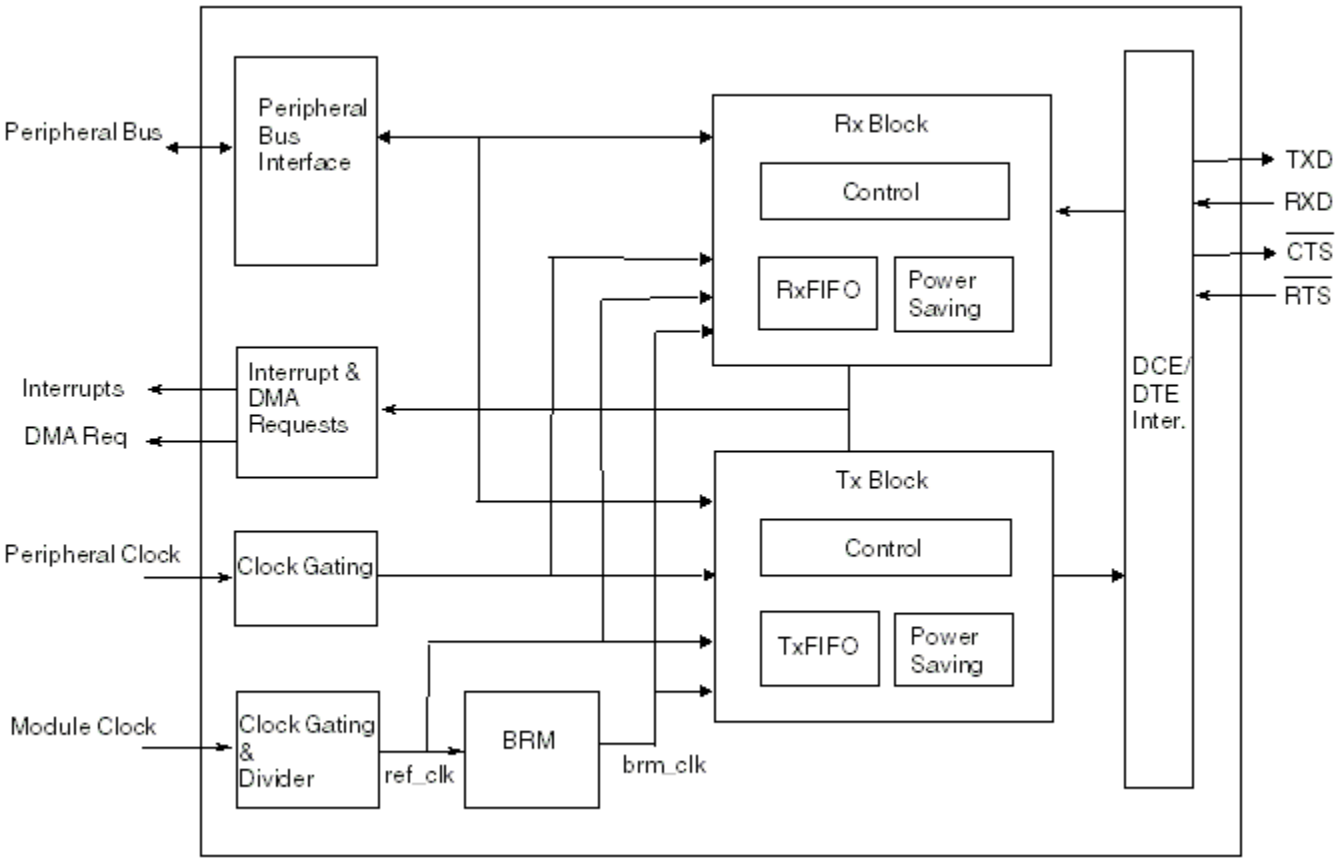


Figure 16-9. UART Block Diagram

16.2.1.1 Features

The UART includes the following features:

- High-speed TIA/EIA-232-F compatible, up to Mbit/s
- Serial IR interface low-speed, IrDA-compatible (up to 115.2 Kbit/s)
- 9-bit or Multidrop mode (RS-485) support (automatic slave address detection)
- 7 or 8 data bits for RS-232 characters, or 9 bit RS-485 format
- 1 or 2 stop bits
- Programmable parity (even, odd, and no parity)
- Hardware flow control support for request to send (RTS_B) and clear to send (CTS_B) signals
- RS-485 driver direction control via CTS_B signal
- Edge-selectable RTS_B and edge-detect interrupts
- Status flags for various flow control and FIFO states
- Voting logic for improved noise immunity (16x oversampling)
- Transmitter FIFO empty interrupt suppression

- UART internal clocks enable/disable
- Auto baud rate detection (up to 115.2 Kbit/s)
- Receiver and transmitter enable/disable for power saving
- RX_DATA input and TX_DATA output can be inverted respectively in RS-232/ RS-485 mode
- DCE/DTE capability
- RTS_B, IrDA asynchronous wake (AIRINT), receive asynchronous wake (AWAKE) interrupts wake the processor from STOP mode
- Maskable interrupts
- Two DMA Requests (TxFIFO DMA Request and RxFIFO DMA Request)
- Escape character sequence detection
- Software reset (SRST_B)
- Two independent, 32-entry FIFOs for transmit and receive
- The peripheral clock can be totally asynchronous with the module clock. The module clock determines baud rate. This allows frequency scaling on peripheral clock (such as during DVFS mode) while remaining the module clock frequency and baud rate.

16.2.1.2 Modes of operation

- Serial RS-232NRZ mode
- 9-bit RS-485 mode
- IrDA mode

To set UART in different modes, see the table below.

Table 16-3. UART mode definition

MDEN (UMCR[0])	IREN (UCR1[7])	UART Mode	Descriptioin
0	0	RS-232	RXD/TXD data is serial RS-232 NRZ format
0	1	IrDA (Interface)	RXD/TXD data is IrDA-compatible infrared slow data rate (SIR) format
1	0	RS-485	RXD/TXD data is RS485 compatible 9 bit data format
1	1	Undefined	Undefined

16.2.2 External Signals

The chip-level IOMUX modifies the direction and routing of the UART signals based on whether the UART is operating in DCE mode (UARTn_UFCR[DCEDTE]=0) or DTE mode (UARTn_UFCR[DCEDTE]=1). The routing of the external signals to the UART module is shown in the figure below.

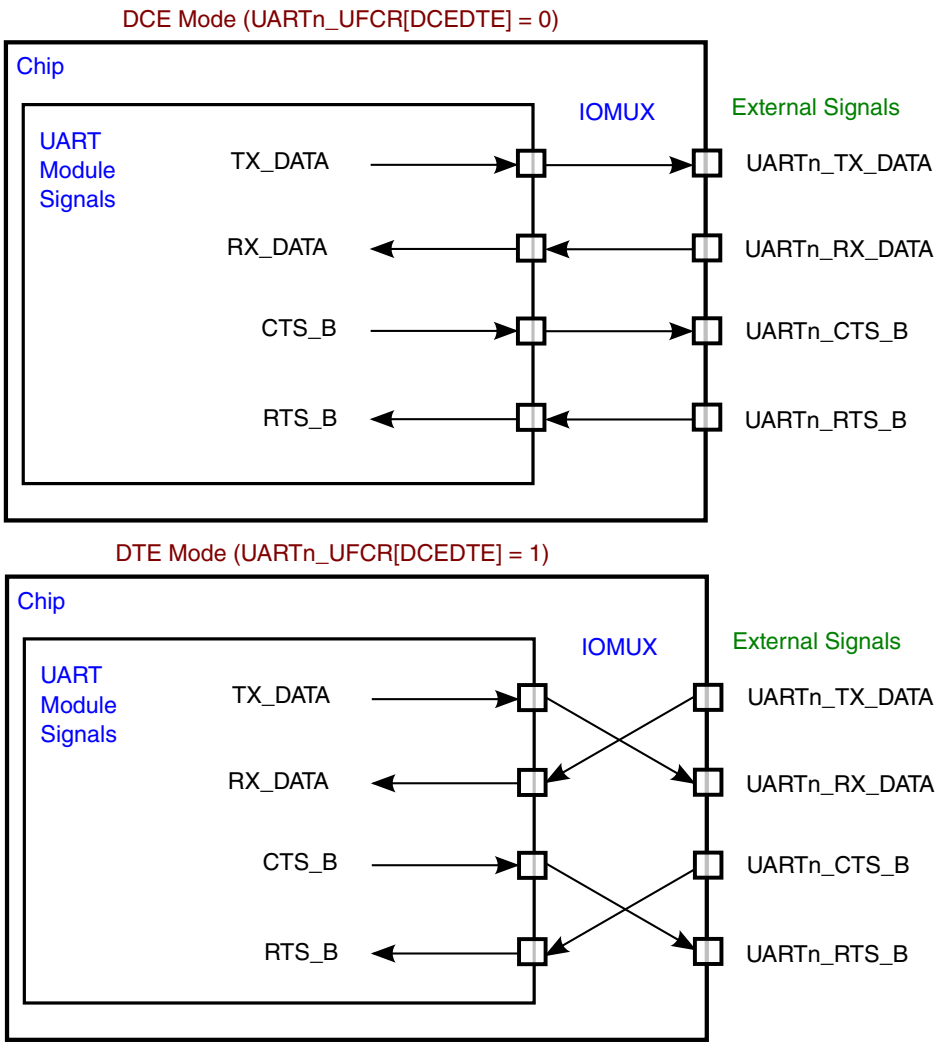


Figure 16-10. UART external signals to module signals routing with respect to DCE/DTE mode

The user must configure the input path to the UART by properly configuring the DAISY bits in the IOMUXC_UARTn_RX_DATA_INPUT and the IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT registers.

For IOMUXC_UARTn_UART_RTS_B_SELECT_INPUT[DAISY]:

- Configurations that select UARTn_RTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_CTS_B for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

For IOMUXC_UARTn_UART_RX_DATA_B_SELECT_INPUT[DAISY]:

- Configurations that select UARTn_RX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=0 (DCE mode)
- Configurations that select UARTn_TX_DATA for the pad are only valid when UARTn_UFCR[DCEDTE]=1 (DTE mode)

16.2.2.1 Detailed Signal Descriptions

16.2.2.1.1 Interrupt Signals

16.2.2.1.1.1 *interrupt_uart* - UART Interrupt

Output interrupt request.

16.2.2.1.2 DMA Request Signals

16.2.2.1.2.1 *dma_req_rx* - Receiver DMA Request

Output DMA Request signal for receiver interface.

16.2.2.1.2.2 *dma_req_tx* - Transmitter DMA Request

Output DMA Request signal for transmitter interface. Set at 0 when TXDMAEN (UCR1[3]) is at 1 and TRDY (USR1[13]) is also at 1.

16.2.2.1.3 Special Signals

16.2.2.1.3.1 *stop_req* - Stop Mode

Input stop mode. Indicates to UART that Arm platform is going to enter in Stop Mode and clocks are going to stop running.

See Low Power Modes for more information about Stop Mode.

16.2.2.1.3.2 *doze_req* - Doze Mode

Input doze mode. Arm platform requests UART to switch in doze mode (power saving mode).

See Low Power Modes for more information about Doze Mode.

16.2.2.1.3.3 *debug_req* - Debug Mode

Input debug mode. Indicates UART it has to enter in debug mode.

See UART Operation in System Debug State, for more information about Debug Mode.

16.2.3 Clocks

The table found here describes the clock sources for UART.

Please see Clock Controller Module (CCM) for clock setting, configuration and gating information.

Table 16-4. UART Clocks

Clock name	Clock Root	Description
ipg_clk	ipg_clk_root	Peripheral clock
ipg_clk_s	ipg_clk_root	Peripheral access clock
ipg_perclk	uart_clk_root	Module clock

16.2.4 Functional Description

This section provides a complete functional description of the block.

16.2.4.1 Interrupts and DMA Requests

See the following table for the lists of all interrupt and DMA signals and associated interrupt and DMA sources of the UART. See register description section for explanation of interrupt/DMA enable and status.

Table 16-5. Interrupts and DMA

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
interrupt_uart	RRDYEN	UCR1 (bit 9)	RRDY	USR1 (bit 9)
	IDEN	UCR1 (bit 12)	IDLE	USR2 (bit 12)
	DREN	UCR4 (bit 0)	RDR	USR2 (bit 0)
	RXDSEN	UCR3 (bit 6)	RXDS	USR1 (bit 6)
	ATEN	UCR2 (bit 3)	AGTIM	USR1 (bit 8)
interrupt_uart	TXMPTYEN	UCR1 (bit 6)	TXFE	USR2 (bit 14)

Table continues on the next page...

Table 16-5. Interrupts and DMA (continued)

Interrupt/DMA Output	Interrupt/DMA Enable	Enable Register Location	Interrupt/DMA Flag	Flag Register Location
	TRDYEN	UCR1 (bit 13)	TRDY	USR1 (bit 13)
	TCEN	UCR4 (bit 3)	TXDC	USR2 (bit 3)
interrupt_uart	OREN	UCR4 (bit 1)	ORE	USR2 (bit 1)
	BKEN	UCR4 (bit 2)	BRCD	USR2 (bit 2)
	WKEN	UCR4 (bit 7)	WAKE	USR2 (bit 7)
	ADEN	UCR1 (bit 15)	ADET	USR2 (bit 15)
	ACIEN	UCR3 (bit 0)	ACST	USR2 (bit 11)
	ESCI	UCR2 (bit 15)	ESCF	USR1 (bit 11)
	ENIRI	UCR4 (bit 8)	IRINT	USR2 (bit 8)
	AIRINTEN	UCR3 (bit 5)	AIRINT	USR1 (bit 5)
	AWAKEN	UCR3 (bit 4)	AWAKE	USR1 (bit 4)
	FRAERREN	UCR3 (bit 11)	FRAERR	USR1 (bit 10)
	PARERREN	UCR3 (bit 12)	PARITYERR	USR1 (bit 15)
	RTSDEN	UCR1 (bit 5)	RTSD	USR1 (bit 12)
	RTSEN	UCR2 (bit 4)	RTSF	USR2 (bit 4)
	DTREN (DCE)	UCR3 (bit 13)	DTRF	USR2 (bit 13)
	RI (DTE)	UCR3 (bit 8)	RIDELT	USR2 (bit 10)
	DCD (DTE)	UCR3 (bit 9)	DCDDELT	USR2 (bit 6)
	DTRDEN	UCR3 (bit 3)	DTRD	USR1 (bit 7)
	SADEN	UMCR (bit 3)	SAD	USR1 (bit 3)
dma_req_rx	RXDMAEN	UCR1 (bit 8)	RRDY	USR1 (bit 9)
	ATDMAEN	UCR1 (bit 2)	AGTIM	USR1 (bit 8)
	IDDMAEN	UCR4 (bit 6)	IDLE	USR2 (bit 12)
dma_req_tx	TXDMAEN	UCR1 (bit 3)	TRDY	USR1 (bit 13)

16.2.4.2 Clocks

This section describes clocks and special clocking requirements of the UART.

16.2.4.2.1 Clock requirements

UART module receives 2 clocks, *peripheral_clock* and *module_clock*. The *peripheral_clock* is used as write clock of the TxFIFO, read clock of the RxFIFO and synchronization of the modem control input pins. It must always be running when UART is enabled. There is an exception in stop mode (see Clocking in Low-Power Modes).

The *module_clock* is for all the state machines, writing RxFIFO, reading TxFIFO, etc. It must always be running when UART is sending or receiving characters.This clock is used in order to allow frequency scaling on *peripheral_clock* without changing configuration of baud rate (*module_clock* staying at a fixed frequency).

The constraints on *peripheral_clock* and *module_clock* are as follows:

- *peripheral_clock* and *module_clock* can totally be asynchronous. They can also be synchronous.
- Due to the 16x oversampling of the incoming characters, *module_clock* frequency must always be greater or equal to 16x the maximum baud rate. For example, if max baud rate is 4 Mbit/s, *module_clock* must be greater or equal to 4 M x 16 = 64 MHz.

NOTE

The restriction that *peripheral_clock* frequency must be higher or equal to 16x baud rate has been removed. There is no limitation on *peripheral_clock* frequency to baud rate.

16.2.4.2.2 Maximum Baud Rate

The max baud rate the UART can support is determined by the max frequency of the module_clock.

For example, if the SoC can provide the fastest module_clock 66.5 MHz, the UART can transmit and receive serial data with the maximum baud rate 66.5M/16 = 4.15 Mbit/s.

The UART supports serial IR interface low speed. In the low speed IrDA mode, the max baud rate is 115.2 Kbit/s. To support the 115.2 Kbit/s, module_clock frequency must be higher or equal to 1.8432 MHz.

16.2.4.2.3 Clocking in Low-Power Modes

The UART supports 2 low-power modes: DOZE and STOP.

In STOP mode (input pin *stop_req* is at '1'), the UART doesn't need any clock. In this mode the UART can wake-up the Arm platform with the asynchronous interrupts (see Low Power Modes).

- If before entering in STOP mode the software has enabled RTSDEN interrupt, when RTS will change state (put at '0' by external device started to send), the asynchronous interrupt will wake-up the system, *peripheral_clock* and *module_clock* will be provided to the UART before first start bit, so that no data will be lost.
- If RTS doesn't change state (already at '0' before entering in STOP mode), then wake-up interrupt (AWAKE) will be sent at the arrival of first Start bit (on falling

edge). In this case, the UART must receive the *peripheral_clock* and *module_clock* during the first half of start bit to correctly receive this character (for example, at 115.2 Kbit/s, UART must receive *peripheral_clock* and *module_clock* at maximum 4.3 microseconds after falling edge of Start bit). If the UART receives *peripheral_clock* and *module_clock* too late, first character will be lost, and so should be dropped. Also, if autobaud detection is enabled, the first character won't be correctly received and another autobaud detection will need to be initiated.

In Doze mode, UART behavior is programmable through DOZE bit (UCR1[1]). If DOZE bit is set to '1', then UART is disabled in Doze mode, and in consequence, UART clocks can be switched-off (after being sure UART is not transmitting nor receiving). On the contrary, if DOZE bit is set to '0', UART is enabled and it must receive *peripheral_clock* and *module_clock* .

16.2.4.3 General UART Definitions

Definitions of terms that occurs the following discussions are given in this section.

- Bit Time-The period of time required to serially transmit or receive 1 bit of data (1 cycle of the baud rate frequency).
- Start bit-The bit time of a logic 0 that indicates the beginning of a data frame. A start bit begins with a 1-to-0 transition, and is preceded by at least 1 bit time of logic 1.
- Stop bit-1 bit time of logic 1 that indicates the end of a data frame.
- BREAK-A frame in which all of the data bits, including the stop bit, are logic 0. This type of frame is usually sent to signal the end of a message or the beginning of a new message.
- Mark - When no data is being sent, the serial port's transmit pin's voltage is 1 and is said to be in a MARK state.
- Space - The serial port can also be forced to keep the transmit pin at a 0 and is said to be the SPACE or BREAK state.
- Frame-A start bit followed by a specified number of data or information bits and terminated by a stop bit. The number of data or information bits depends on the format specified and must be the same for the transmitting device and the receiving device. The most common frame format is 1 start bit followed by 8 data bits (least significant bit first) and terminated by 1 stop bit. An additional stop bit and a parity bit also can be included.
- Framing Error-An error condition that occurs when the stop bit of a received frame is missing, usually when the frame boundaries in the received bit stream are not synchronized with the receiver bit counter. Framing errors can go undetected if a data bit in the expected stop bit time happens to be a logic 1. A framing error is always present on the receiver side when the transmitter is sending BREAKs. However,

when the UART is programmed to expect 2 stop bits and only the first stop bit is received, this is not a framing error by definition.
- Parity Error-An error condition that occurs when the calculated parity of the received data bits in a frame does not match the parity bit received on the RX_DATA input. Parity error is calculated only after an entire frame is received.
- Idle-One in NRZ encoding format and selectable polarity in IrDA mode.
- Overrun Error-An error condition that occurs when the latest character received is ignored to prevent overwriting a character already present in the UART receive buffer (RxFIFO). An overrun error indicates that the software reading the buffer (RxFIFO) is not keeping up with the actual reception of characters on the RX_DATA input.

16.2.4.3.1 RTS_B - UART Request To Send

The UART Request To Send input controls the transmitter. The modem or other terminal equipment signals the UART when it is ready to receive by setting '0' on the RTS_B pin.

Normally, the transmitter waits until this signal is active (low) before transmitting a character, however when the Ignore RTS (IRTS) bit is set, the transmitter sends a character as soon as it is ready to transmit. An interrupt (RTSD) can be posted on any transition of this pin and can wake the Arm platform from STOP mode on its assertion. When RTS_B is set to '1' during a transmission, the UART transmitter finishes transmitting the current character and shuts off. The contents of the TxFIFO (characters to be transmitted) remain undisturbed. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode.

16.2.4.3.2 RTS Edge Triggered Interrupt

The input to the RTS_B pin can be programmed to generate an interrupt on a selectable edge.

See the table below for summary of the operation of the RTS edge triggered interrupt (RTSF).

To enable the RTS_B pin to generate an interrupt, set the request to send interrupt enable (RTSEN) bit (UCR2[4]) to 1. Writing 1 to the RTS_B edge triggered interrupt flag (RTSF) bit (USR2[4]) clears the interrupt flag. The interrupt can occur on the rising edge, falling edge, or either edge of the RTS_B input. The request to send edge control (RTEC) field (UCR2[10:9]) programs the edge that generates the interrupt. When RTEC is set to 0x00 and RTSEN = 1, the interrupt occurs on the rising edge (default). When RTEC is set to 0x01 and RTSEN = 1, the interrupt occurs on the falling edge. When RTEC is set to 0x1X and RTSEN = 1, the interrupt occurs on either edge. This is a synchronous interrupt. The RTSF bit is cleared by writing 1 to it. Writing 0 to RTSF has no effect.

Table 16-6. RTS_B Edge Triggered Interrupt Truth Table

RTS_B	RTSEN	RTEC [1]	RTEC [0]	RTSF	Interrupt Occurs On…	interrupt_uart
X	0	X	X	0	Interrupt disabled	1
1->0	1	0	0	0	Rising edge	1
0->1	1	0	0	1	Rising edge	0
1->0	1	0	1	1	Falling edge	0
0->1	1	0	1	0	Falling edge	1
1->0	1	1	X	1	Either edge	0
0->1	1	1	X	1	Either edge	0

There is another RTS_B interrupt that is not programmable. The status bit RTSD asserts the *interrupt_uart* interrupt when the RTS_B delta interrupt enable = 1. This is an asynchronous interrupt. The RTSD bit is cleared by writing 1 to it. Writing 0 to the RTSD bit has no effect.

16.2.4.3.3 CTS_B - Clear To Send

This output pin serves two purposes. Normally, the receiver indicates that it is ready to receive data by asserting this pin (low). When the CTS_B trigger level is programmed to trigger at 32 characters received and the receiver detects the valid start bit of the 33 character, it de-asserts this pin. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode.

16.2.4.3.4 Programmable CTS_B Deassertion

The CTS_B output can also be programmed to deassert when the RxFIFO reaches a certain level. Setting the CTS trigger level (UCR4[15:10]) at any value less than 32 deasserts the CTS_B pin on detection of the valid start bit of the N + 1 character (where N is the trigger level setting). However, the receiver continues to receive characters until the RxFIFO is full.

16.2.4.3.5 TX_DATA - UART Transmit

This is the transmitter serial output. When operating in RS-232/RS-485 mode, NRZ encoded data is transmitted, and the data can be inverted (controlled by INVT (UCR3[1])) before transmitted. When operating in infrared mode, a 3/16 bit-period pulse is output for each 0 bit transmitted, and no pulse is output for each 1 bit transmitted.

For RS-232/RS-485 applications, this pin must be connected to an RS-232/RS-485 transmitter. The operation of this output is the same regardless of whether the UART is in DTE or DCE mode. See Figure 16-11.

16.2.4.3.6 RX_DATA - UART Receive

This is the receiver serial input. When operating in RS-232/RS-485 mode, NRZ encoded data is expected, and the data can be inverted (controlled by INVR (UCR4[9])) before sampled. When operating in infrared mode, a narrow pulse is expected for each 0 bit received and no pulse is expected for each 1 bit received.

External circuitry must convert the IR signal to an electrical signal. RS-232/RS-485 applications require an external RS-232/RS-485 receiver to convert voltage levels. The operation of this input is the same regardless of whether the UART is in DTE or DCE mode. See the figure below.

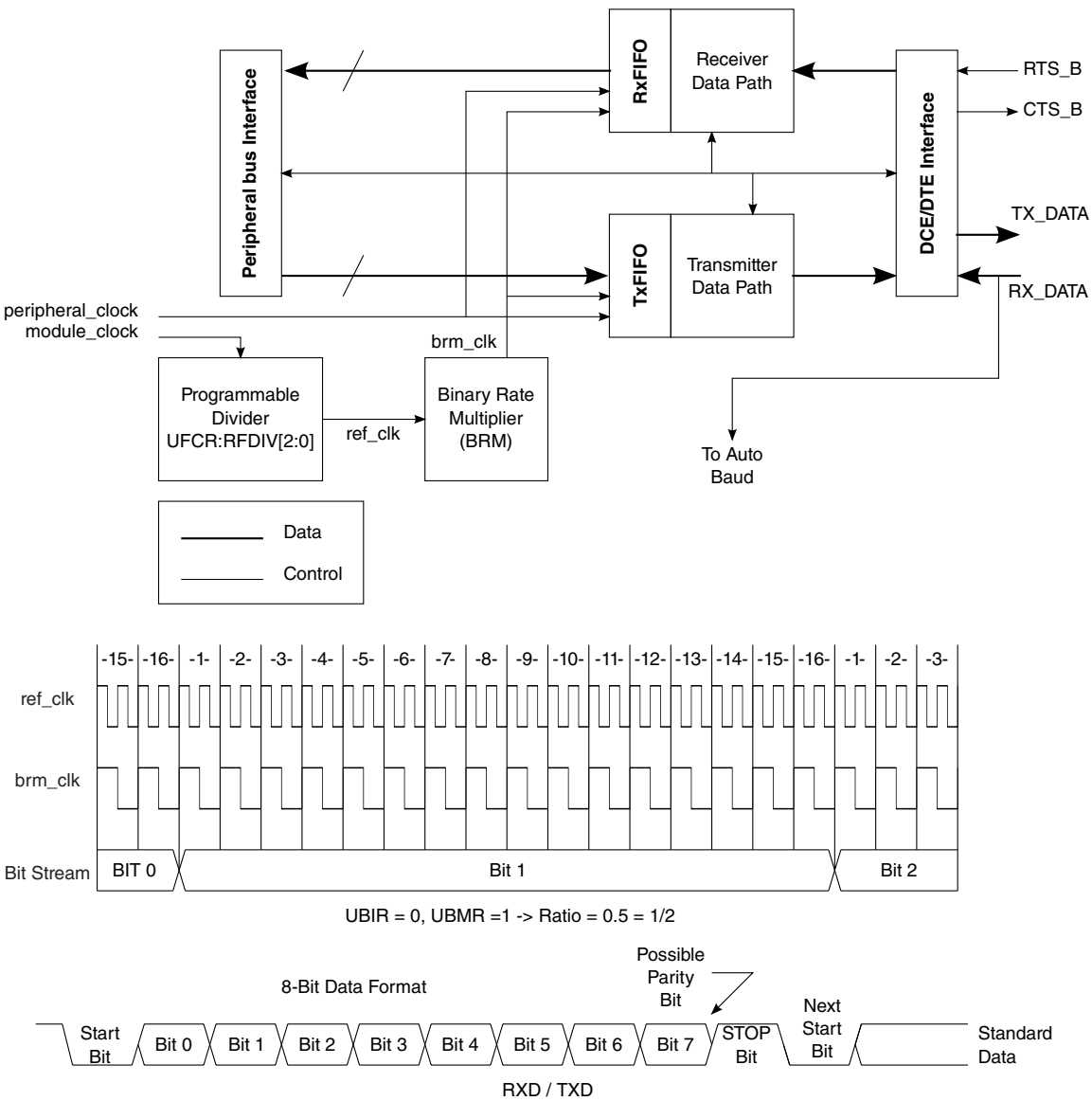


Figure 16-11. UART Simplified Block and Clock Generation Diagrams

16.2.4.4 Transmitter

The transmitter accepts a parallel character from the Arm platform and transmits it serially. The start, stop, and parity (when enabled) bits are added to the character.

When the ignore RTS bit (IRTS) is set, the transmitter sends a character as soon as it is ready to transmit. RTS_B can be used to provide flow-control of the serial data. When RTS_B is set to '1', the transmitter finishes sending the character in progress (if any), stops, and waits for RTS_B to be set to '0' again. Generation of BREAK characters and parity errors (for debugging purposes) is supported. The transmitter operates from the clock provided by the Binary Rate Multiplier(BRM). Normal NRZ encoded data is transmitted when the IR interface is disabled.

The transmitter FIFO (TxFIFO) contains 32 bytes. The data is written to TxFIFO by writing to the UTXD register with the byte data to the [7:0] bits. The data is written consecutively if the TxFIFO is not full. It is read (internally) consecutively if the TxFIFO is not empty. TXFULL bit (UTS[4]) can be used to control whether TXFIFO is full or not. The TxFIFO can be written regardless of the transmitter is disabled or enabled. If the UART is disabled, user can still write data into the TxFIFO correctly. But in this case the write access will yield to a transfer error.

16.2.4.4.1 Transmitter FIFO Empty Interrupt Suppression

The transmitter FIFO empty interrupt suppression logic suppresses the TXFE interrupt between writes to the TxFIFO.

When TxFIFO is empty, the software can either send one or several characters. If the software sends one character, it would write the character into the UTXD register, then that character is immediately transferred to the transmitter shift register, assuming the transmitter is already enabled. Without interrupt suppression logic, the TXFE interrupt flag would be set immediately. But, with this logic, the interrupt flag is set when the last bit of the character has been transmitted, for example, before the transmission of the parity bit (if exists) and the stop bit(s).

So, the suppression logic doesn't immediately send the TXFE interrupt flag. It allows the software to write another character to the TxFIFO before the interrupt flag is asserted.

When the transmitter shift register empties before another character is written to the TxFIFO, the interrupt flag is asserted. Writing data to the TxFIFO would release the interrupt flag. The interrupt flag is asserted on the following conditions:

- System Reset
- UART software reset

- When a single character has been written to Transmitter FIFO and then the Transmitter FIFO and the Transmitter Shift Register become empty until another character is written to the Transmitter FIFO
- The last character in the TxFIFO is transferred to the shift register, when TxFIFO contains two or more characters. See the figure below.

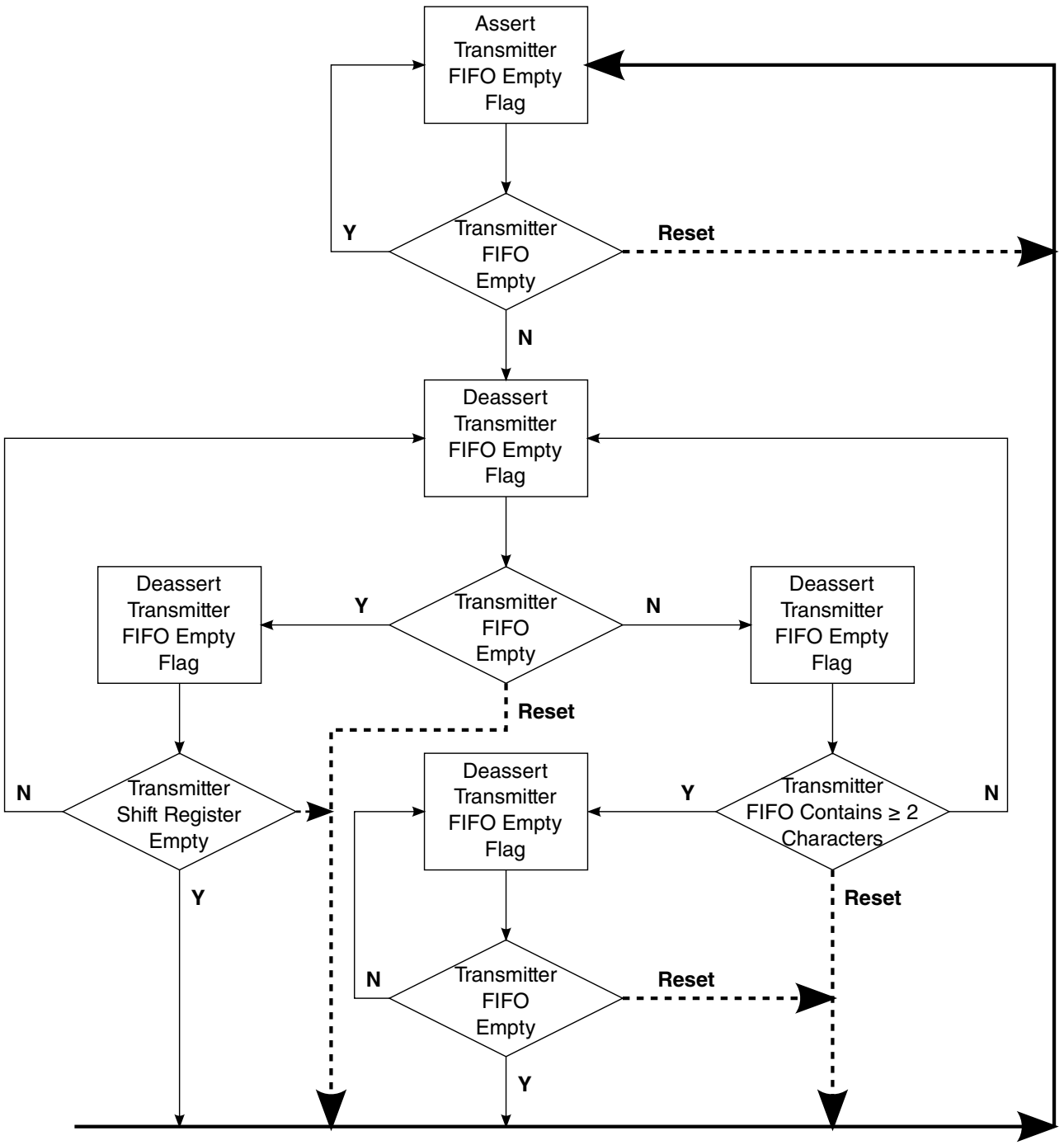


Figure 16-12. Transmitter FIFO Empty Interrupt Suppression Flow Chart

16.2.4.4.2 Transmitting a Break Condition

Asserting SNDBRK bit of the UCR1 Register forces the transmitter to send a break character (continuous zeros). The transmitter will finish sending the character in progress (if any) before sending break until this bit is reset.

The user is responsible to ensure that this bit is high for long enough to generate a valid BREAK. The transmitter samples SNDBRK after every bit is transmitted. Following completion of the BREAK transmission, the UART will transmit two mark bits. The user can continue to fill the FIFO and any character remaining will be transmitted when the break is terminated.

16.2.4.5 Receiver

See the figure below for the receiver flow chart.

The receiver accepts a serial data stream and converts it into parallel characters. When enabled, it searches for a start bit, qualifies it, and samples the following data bits at the bit-center.

Jitter tolerance and noise immunity are provided by sampling at a 16x rate and using voting techniques to clean up the samples. Once the start bit is found, the data bits, parity bit (if enabled), and stop bits (either 1 or 2 depending on user selection) are shifted in. Parity is checked and its status reported in the URXD register when parity is enabled. Frame errors and BREAKs are also checked and reported. When a new character is ready to be read by the Arm platform from the RxFIFO, the receive data ready (RDR = USR2[0]) bit is asserted and an interrupt is posted (if DREN = UCR4[0] = 1). If the receiver trigger level is set to 2 (RXTL[5:0] = UFCR[5:0] = 2), and 2 chars have been received into RxFIFO, the receiver ready interrupt flag (RRDY = USR1[9]) is asserted and an interrupt is posted if the receiver ready interrupt enable bit is set (RRDYEN = UCR1[9] = 1). If the UART Receiver Register (URXD) is read once, and in consequence there is only 1 character in the RxFIFO, the interrupt generated by the RDR bit is automatically cleared. The RRDY bit is cleared when the data in the RxFIFO falls below the programmed trigger level.

Normal NRZ encoded data is expected when the IR interface is disabled. The RxFIFO contains 32 half-word entries. Characters received are written consecutively into this FIFO. If the FIFO is full and a 33rd characters is received, this character will be ignored and the USR2[ORE] bit will be set.

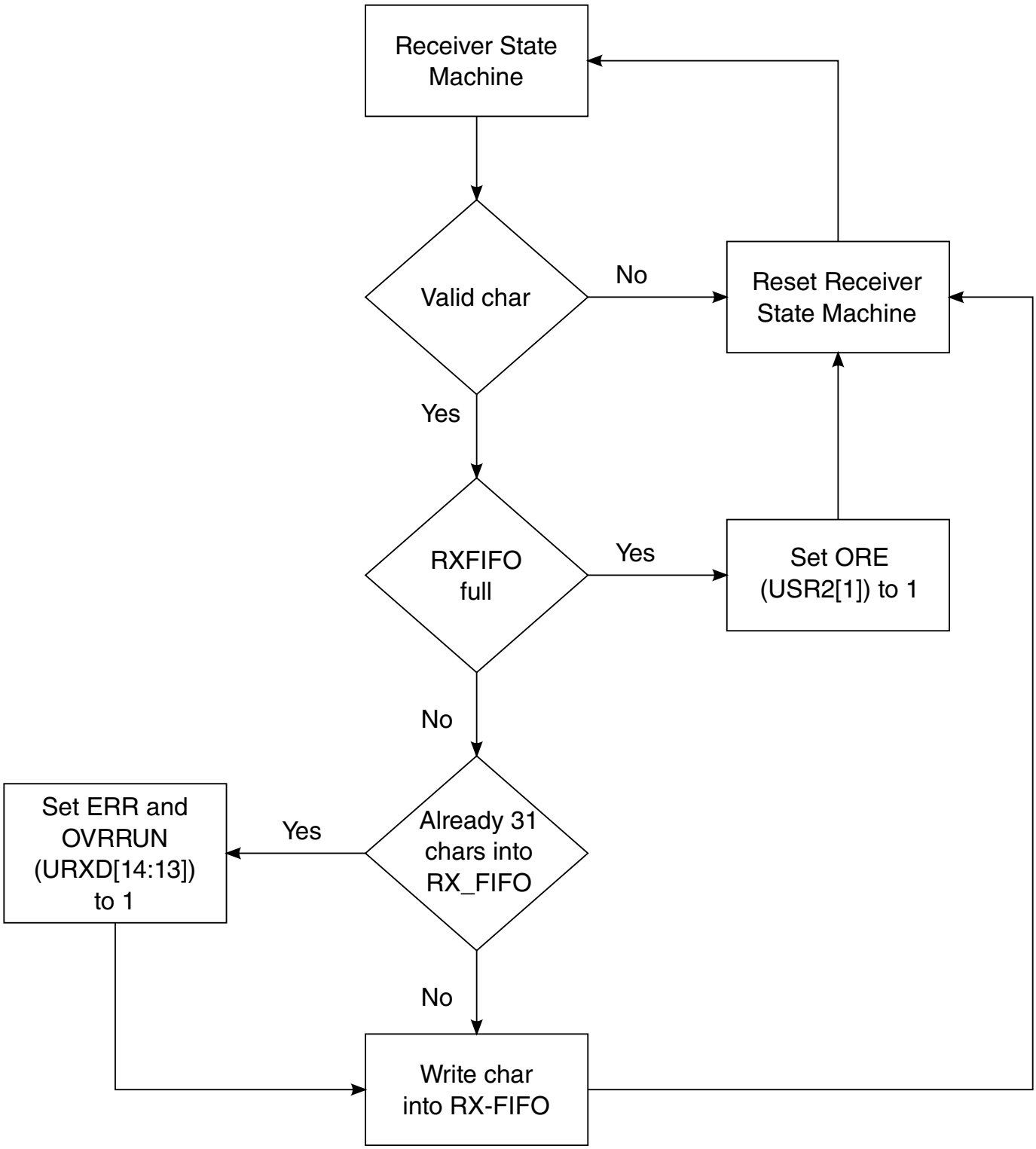


Figure 16-13. Receiver Flow Chart

16.2.4.5.1 Idle Line Detect

The receiver logic block includes the ability to detect an idle line. Idle lines indicate the end or the beginning of a message.

For an idle condition to occur:

- RxFIFO must be empty and
- RX_DATA pin must be idle for more than a configured number of frames (ICD[1:0] = UCR1[11:10]).

When the idle condition detected interrupt enable (IDEN = UCR1[12]) is set and the line is idle for 4 (default), 8, 16, or 32 (maximum) frames, the detection of an idle condition flags an interrupt (see the table below). When an idle condition is detected, the IDLE (USR2[12]) bit is set. Clear the IDLE bit by writing 1 to it. Writing 0 to the IDLE bit has no effect.

Table 16-7. Detection Truth Table

IDEN	ICD [1]	ICD [0]	IDLE	*interrupt_uart*
0	X	X	0	1
1	0	0	asserted after 4 idle frames	asserted after 4 idle frames
1	0	1	asserted after 8 idle frames	asserted after 8 idle frames
1	1	0	asserted after 16 idle frames	asserted after 16 idle frames
1	1	1	asserted after 32 idle frames	asserted after 32 idle frames
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal. This table shows how this interrupt affects the *interrupt_uart* signal.				

During a normal message there is no idle time between frames. When all of the information bits in a frame are logic 1s, the start bit ensures that at least one logic 0 bit time occurs for each frame so that the IDLE bit is not asserted.

16.2.4.5.2 Aging Character Detect

The receiver block also includes the possibility to detect when at least one character has been sitting into the RxFIFO for a time corresponding to 8 characters. This aging character capability allows the UART to inform the Arm platform that there is less character into the RxFIFO than the Rx trigger and, no new character has been detected on the RXD line.

The aging capability is a timer which starts to count as soon as the RxFIFO is not empty and its trigger level is not reached (RRDY=0). This counter is reset when either a RxFIFO read is performed or another character starts to present on the RXD line. If none of those two events occurs, the bit AGTIM (USR1[8]) is set when the counter has

measured a time corresponding to 8 characters. AGTIM is cleared by writing a 1 to it. AGTIM can flag an interrupt to Arm platform on *interrupt_uart* if ATEN (UCR2[3]) has been set.

To summarize, AGTIM is set when:

- There is at least one character into RxFIFO.
- No read has occurred on RxFIFO and RXD line has stayed high, for a time corresponding to 8 characters.
- The RxFIFO trigger is not reached (RRDY=0)

16.2.4.5.3 Receiver Wake

The WAKE bit (USR2[7]) is set when the receiver detects a qualified Start bit. For this, two conditions must be fulfilled, firstly a falling edge on RX_DATA line must be detected and secondly the RX_DATA line must stays at low level for more than a half-bit duration.

When the wake interrupt enable WKEN (UCR4[7]) bit is enabled, the receiver flags an interrupt (*interrupt_uart*) if the WAKE status bit is set. The WAKE bit is cleared by writing 1 to it. Writing 0 to the WAKE bit has no effect. The WAKE status bit can be asserted in either serial RS-232 mode or IR mode. The generation of the WAKE interrupt needs the clock *module_clock* .

When the asynchronous wake interrupt (AWAKE) is enabled (AWAKEN = UCR3[4] = 1), and the Arm platform is in STOP mode, and UART clocks have been shut-off, then a falling edge detected on the receive pin (RX_DATA) asserts the AWAKE bit (USR1[4]) and the *interrupt_uart* interrupt to wake the Arm platform from STOP mode. Re-enable UART clocks and clear the AWAKE bit by writing 1 to it. Writing 0 to the AWAKE bit has no effect. When IR interface is enabled (UCR1[7]=1), the AWAKE bit is always not asserted. The generation of the asynchronous AWAKE interrupt does not need any clocks.

In IR mode, if the asynchronous IR WAKE interrupt is enabled (AIRINTEN = UCR3[5] = 1), and if the Arm platform is in STOP mode (UART clocks are off when Arm platform in STOP mode), then the detection of a falling edge on the receive pin (RXD_IR), asserts the AIRINT bit (USR1[5]), and the *interrupt_uart* interrupt. This interrupt wakes the Arm platform from STOP mode. Software re-enables UART clocks and clear the AIRINT bit by writing 1 to it. Writing 0 to the AIRINT bit has no effect. When IR interface is disabled (UCR1[7]=0), the AIRINT bit is always not asserted. The generation of the asynchronous AIRINT interrupt does not need any clocks.

Recommended procedure for programming the asynchronous interrupts is to first clear them by writing 1 to the appropriate bit in the UART Status Register 1 (USR1). Poll or enable the interrupt for the Receiver IDLE Interrupt Flag (RXDS) in the USR1. When asserted, the RXDS bit indicates to the software that the receiver state machine is in the idle state, the next state is idle, and the RX_DATA pin is idle (high). After following this procedure, enable the asynchronous interrupt and enter STOP mode.

16.2.4.5.4 Receiving a BREAK Condition

A BREAK condition is received when the receiver detects all 0s (including a 0 during the bit time of the stop bit) in a frame. The BREAK condition asserts the BRCD bit (USR2[2]) and writes only the first BREAK character to the RxFIFO. Clear the BRCD bit by writing 1 to it. Writing 0 to the BRCD bit has no effect.

Asserting BRCD would generate an interrupt on *interrupt_uart*. The interrupt generation can be masked using the control bit BKEN (UCR4[2]). Receiving a break condition will also effect the following bits in the receiver register URXD:

URXD(11) = BRK. While high this bit indicates that the current char was detected as a break.

URXD(12) = FRMERR. The frame error bit will always be set when BRK is set.

URXD(10) = PRERR. If odd parity was selected the parity error bit will also be set when BRK is set.

URXD(14) = ERR. The error detect bit indicates that the character present in the rx data field has an error status. This can be asserted by a break.

16.2.4.5.5 Vote Logic

The vote logic block provides jitter tolerance and noise immunity by sampling with respect to a 16x clock (*brm_clk*) and using voting techniques to clean up the samples. The voting is implemented by sampling the incoming signal constantly on the rising edge of the *brm_clk*.

See Figure 16-14. The receiver is provided with the majority vote value, which is 2 out of the 3 samples. For examples of the majority vote results of the vote logic, see the following table.

Table 16-8. Majority Vote Results

Samples	Vote
000	0

Table continues on the next page...

Table 16-8. Majority Vote Results (continued)

Samples	Vote
101	1
001	0
111	1

The vote logic captures a sample on every rising edge of *brm_clk*, however the receiver uses 16x oversampling to take its value in the middle of the sample character.

The receiver starts to count when the Start bit is set however it does not capture the contents of the RxFIFO at the time the Start bit is set. The start bit is validated when 0s are received for 7 consecutive 1/16 of bit times following the 1-to-0 transition. Once the counter reaches 0xF, it starts counting on the next bit and captures it in the middle of the sampling frame (see Table 16-8). All data bits are captured in the same manner. Once the stop bit is detected, the receiver shift register (SIPO_OUT) data is parallel shifted to the RxFIFO.

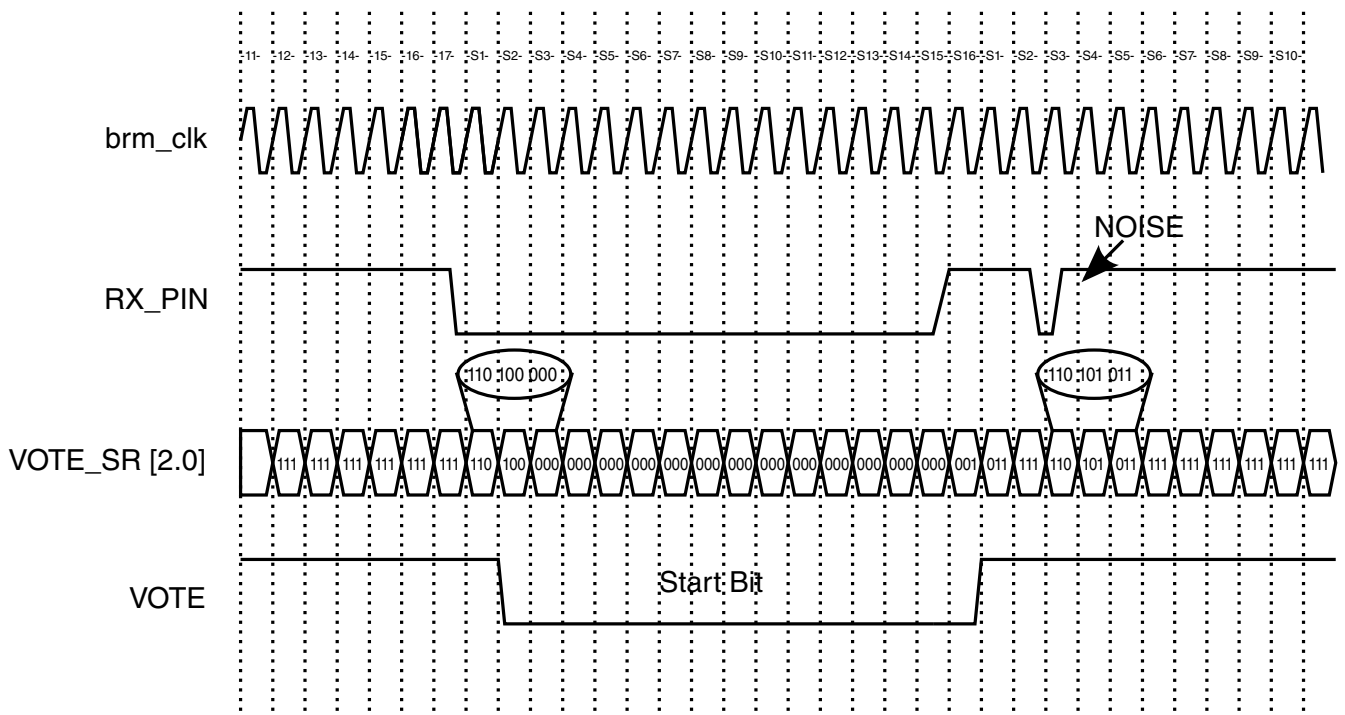


Figure 16-14. Majority Vote Results

A new feature has been recently implemented, it allows to re-synchronize the counter on each edge of RX_DATA line. This is automatic and allows to improve the immunity of UART against signal distortion.

There is a special case when the *brm_clk* frequency is too low and is unable to capture a 0 pulse in IrDA. In this case, the software must set the IRSC (UCR4[5]) bit so that the reference clock (after internal divider) is used for the voting logic. The pulse is validated by counting the length of the pulse.

Refer to Infrared Interface for more details.

16.2.4.5.6 Baud Rate Automatic Detection Logic

When the baud rate automatic detection logic is enabled, the UART locks onto the incoming baud rate. To enable this feature, set the automatic detection of baud rate bit (ADBR = UCR1[14] = 1) and write 1 to the ADET bit (USR2[15]) to clear it.

When ADET=0 and ADBR =1, the detection starts. Then, once the beginning of start bit (transition from 1-to-0 of RX_DATA) has been detected, UART starts a counter (UBRC) working at reference frequency. Once the end of start bit is detected (transition from 0-to-1 of RX_DATA), the value of UBRC - 1 is directly copied into UBMR register. UBIR register is filled with 0x000F.

So, at the end of start bit, registers gets following values:

```
UBRC = number of reference clock periods (after divider) during Start bit.
UBIR = 0x000F
UBMR = UBRC - 1
```

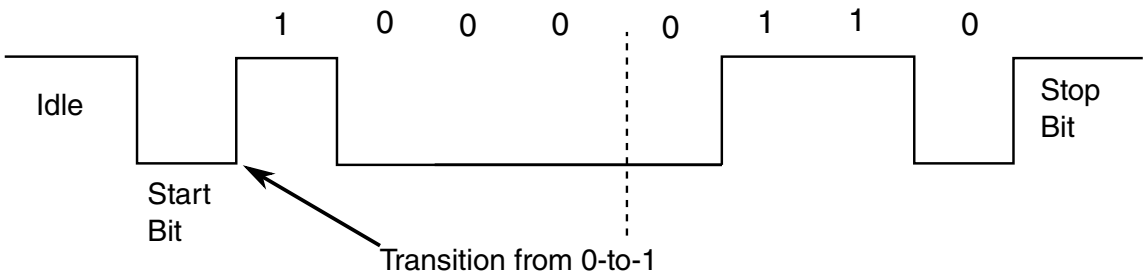
The updated values of the 3 registers can be read.

See Table 16-9 for list of parameters for baud rate detection and Figure 16-15 for baud rate detection protocol diagram.

If any of the UART BRM registers are simultaneously written by the baud rate automatic detection logic and by the peripheral data bus, the peripheral data bus would have lower priority.

Table 16-9. Baud Rate Automatic Detection

ADBR	ADET	Baud Rate Detection	*interrupt_uart*
0	X	Manual Configuration	1
1	0	Auto Detection Started	1
1	1	Auto Detection Complete	0
NOTE: This table assumes that no other interrupt is set at the same time this interrupt is set for the *interrupt_uart* signal.			



Note: LSB Transmitted first.

Figure 16-15. Baud Rate Detection Protocol Diagram

16.2.4.5.6.1 Baud Rate Automatic Detection Protocol

The receiver must receive an ASCII character "A" or "a" to verify proper detection of the incoming baud rate. When an ASCII character "A" (0x41) or "a" (0x61) is received and no error occurs, the Automatic Detect baud rate bit is set (ADET=1) and if the interrupt is enabled (ADEN=UCR1[15]=1), an interrupt *interrupt_uart* is generated.

When an ASCII character "A" or "a" is not received (because of a bit error or the reception of another character), the auto detection sequence restarts and waits for another 1-to-0 transition.

As long as ADET = 0 and ADBR = 1, the UART continues to try to lock onto the incoming baud rate. Once the ASCII character "A" or "a" is detected and the ADET bit is set, the receiver ignores the ADBR bit and continues normal operation with the calculated baud rate.

The UART interrupt is active (*interrupt_uart* = 0) as long as ADET = 1 and ADBR = 1. This can be disabled by clearing the automatic baud rate detection interrupt enable bit (ADEN = 0). Before starting an automatic baud rate detection sequence, set ADET = 0 and ADBR = 1.

The RxFIFO must contain the ASCII character "A" or "a" following the automatic baud rate detection interrupt.

The 16-bit UART Baud Rate Count Register (UBRC) is reset to 4 and stays at 0xFFFF when an overflow occurs. The UBRC register counts (measures) the duration of start bit. When the start bit is detected and counted, the UART Baud Rate Count Register retains its value until the next automatic baud rate detection sequence is initiated.

The Baud Rate Count Register counts only when auto detection is enabled.

16.2.4.5.6.2 New Baud Rate Determination

In order to fight against the problems caused by the distortion and the noise on the RX_DATA line, the duration of the baud rate measurement has been extended.

Previously, as described above, this determination was based on the measurement of the START bit duration. Now, this measurement is based on the duration of START bit + bit0. Bit0 is the first bit following the START bit. In fact, the counter which is started at the falling edge of START bit is no longer stopped at next rising edge (end of START bit), but it is stopped at the next falling edge (end of bit0). As the character sent is always a "A" (41h) or a "a" (61h), this second falling edge will always be present and it will indicate the end of bit0. Once this counter is stopped, the result is divided by 2 and used by the BRM to determine the incoming baud rate.

NOTE

UBRC register contains the result of this division by two, in consequence it reflects the measurement of the duration of one bit.

16.2.4.5.6.2.1 New Autobaud Counter Stopped bit and Interrupt

A new bit has been added in USR2 register: ACST (USR2[11]). This bit is set immediately after the determination of the baud rate.

So,

- if ADNIMP is not set (default), ACST is set to 1 after the end of bit0,
- If ADNIMP is set to 1, ACST is set to 1 at the end of START bit.

If ACIEN (UCR3[0]) is set to 1, ACST will flag an interrupt on *interrupt_uart* signal. This interrupt informs the Arm platform that the BRM has just been set with the result of the bit length measurement. If needed, the Arm platform can perform a read of UBMR (or UBRC) register and determine by itself the baud rate measured. Then the Arm platform has the possibility to correct the BRM registers with the nearest standardized baud rate.

NOTE

ACST is set only if ADBR is set to 1, for example, the UART is autobauding.

Clear the ACST bit by writing 1 to it. Writing 0 to the ACST bit has no effect.

16.2.4.6 Escape Sequence Detection

An escape sequence typically consists of 3 characters entered in rapid succession (such as +++). Because these are valid characters by themselves, the time between characters determines if it is a valid escape sequence.

Too much time between two of the "+" characters is interpreted as two "+" characters, and not part of an escape sequence.

The software chooses the escape character and writes its value to the UART Escape Character Register (UESC). The software must also enable escape detection feature by setting ESCEN (UCR2[11]) to 1. The hardware compares this value to incoming characters in the RxFIFO. When an escape character is detected, the internal escape timer starts to count. The software specifies a time-out value for the maximum allowable time between 2 successive escape characters (see the table below). The escape timer is programmable in intervals of 2 ms to a maximum interval of 8.192 seconds.

Table 16-10. Escape Timer Scaling

UTIM Register	Maximum Time Between Specified Escape Characters
0x000	2 ms
0x001	4 ms
0x002	6 ms
0x003	8 ms
0x004	10 ms
...	...
0F8	498 ms
0F9	500 ms
...	...
9C3	5 s
...	...
FFD	8.188 s
FFE	8.190 s
FFF	8.192 s
NOTE: To calculate the time interval: (UTIM_Value + 1) x 0.002 = Time_Interval Example: (09C3 + 1) x 0.002 = 5 s.	

The escape sequence detection feature is available for all the reference frequencies. Before using Escape Sequence Detection, the user must fill the ONEMS register. This 24-bit register must contain the value of the UART internal frequency divided by 1000. The internal frequency is obtained after the UART internal divider which is applied on *module_clock* clock.

Example I:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 2 with the internal divider: UFCR[9:7] = 3'b100

$$ONEMS = \frac{66.5 \times 10^6}{2 \times 1000} = 33250 = 81E2h$$

Figure 16-16. Calculation of Frequency for ONEMS Register

Example II:

- If the input clock *module_clock* frequency is 66.5 MHz.
- And if the input clock *module_clock* is divided by 1with the internal divider: UFCR[9:7] = 3'b101

$$ONEMS = \frac{66.5 \times 10^6}{1000} = 66500 = 103C4h$$

Figure 16-17. Calculation of Frequency for ONEMS Register

The escape sequence detection interrupt is asserted when the escape sequence interrupt enable (ESCI) bit is set and an escape sequence is detected (ESCF set). Clear the ESCF bit by writing 1 to it. Writing 0 to the ESCF bit has no effect.

16.2.5 Binary Rate Multiplier (BRM)

The BRM sub-block receives *ref_clk* (*module_clock* clock after divider). From this clock, and with integer and non-integer division, BRM generates a 16x baud rate clock .

The UART transmitte will shift data out based on this 16x baud rate clock. The UART receiver will sample the serial data line based on this 16x baud rate clock. The input and output frequency ratio is programmed in the UART BRM Incremental Register (UBIR) and UART BRM MOD Register (UBMR). The output frequency is divided by the input frequency to produce this ratio. For integer division, set the UBIR = 0x000F and write the divisor to the UBMR register. All values written to these registers must be one less than the actual value to eliminate division by 0 (undefined), and to increase the maximum range of the registers.

Updating the BRM registers requires writing to both registers. The UBIR register must be written before writing to the UBMR register. If only one register is written to by the software, the BRM continues to use the previous values.

The following examples show how to determine what values are to be programmed into UBIR and UBMR for a given reference frequency and desired baud rate. The following equation can be used to help determine these values:

$$\text{BaudRate} = \frac{\text{Ref Freq}}{\left(16 \times \frac{\text{UBMR} + 1}{\text{UBIR} + 1} \right)}$$

Figure 16-18. Frequency and Baud Rate for UBIR and UBMR

With:

Reference Frequency (Hz): UART Reference Frequency (*module_clock* after RFDIV divider)

Baud Rate (bit/s): Desired baud rate.

Integer Division ÷ 21

Reference Frequency = 19.44 MHz

UBIR = 0x000F

UBMR = 0x0014

Baud Rate = 925.7 kbit/s

NOTE

Observe that each value written to the registers is one less than the actual value.

Non-Integer Division

```
Reference Frequency = 16 MHz
Desired Baud Rate = 920 Kbits/s
```

$$\frac{UBMR + 1}{UBIR + 1} = \frac{RefFreq}{16 \times BaudRate} = \frac{16 \times 10^6}{16 \times 920 \times 10^3} = 1.087$$

```
Ratio = 1.087 = 1087 / 1000
UBIR = 999 (decimal)= 0x3E7
UBMR = 1086 (decimal)= 0x43E
Non-Integer Division
Reference Frequency = 25 MHz
Desired Baud Rate = 920 kbit/s
Ratio = 1.69837 = 625 / 368
UBIR = 367 (decimal)= 0x16F
UBMR = 624 (decimal)= 0x270
```

Non-Integer Division

```
Reference Frequency: 30 MHz
Desired Baud Rate = 115.2 kbit/s
Ratio = 16.276043 = 65153 / 4003
UBIR = 4002 (decimal) = 0x0FA2
UBMR = 65152 (decimal) = 0xFE80
```

16.2.6 Infrared Interface

16.2.6.1 Generalities-Infrared

The Infrared interface is selected when IREN (UCR1[7]) is set to 1.

The Infrared Interface is compatible with IrDA Serial Infrared Physical Layer Specification. In this specification, a "zero" is represented by a positive pulse, and a "one" is represented by no pulse (line remains low).

In the UART:

In TX: For each "zero" to be transmitted, a narrow positive pulse which is 3/16 of a bit time is generated. For each "one" to be transmitted no pulse is generated (output is low). External circuitry has to be provided to drive an Infrared LED.

In RX: When receiving, a narrow negative pulse is expected for each "zero" transmitted while no pulse is expected for each "one" transmitted (input is high).

NOTE

Rx part of IR block expects to receive an inverted signal compared to IrDA specification. Circuitry external to the IC transforms the Infrared signal to an electrical signal.

The IR interface has an edge triggered interrupt (IRINT). This interrupt validates a zero bit being received. This interrupt is enabled by writing a "one" to ENIRI bit.

The behavior of Infrared Interface is determined by 3 bits INVT (UCR3[1]), INVR (UCR4[9]) and IRSC (UCR4[5]).

16.2.6.2 Inverted Transmission and Reception bits (INVT & INVR)

The values of INVT and INVR depend of the IrDA transceiver connected on the TXD_IR and RXD_IR pins of the UART. If this transceiver is not inverting on both paths Tx and Rx, a Zero is represented by a positive pulse and a One is represented by no pulse (line remains low). In this case, the bit INVT must be set to 0 and the bit INVR must be set to 1 (because Rx IR block expects an inverted signal).

On the contrary user must set INVT=1 and INVR=0 if both paths of the transceiver are inverting, that is, a Zero is represented as a negative pulse and a One is represented by no pulse (line remains high). The transceiver can also be inverting on only one path (Tx or Rx), in this case INVT and INVR must be together equal to 1 or to 0, depending on which path is inverted.

16.2.6.3 InfraRed Special Case (IRSC) Bit

The value to apply to IRSC bit is based on 2 parameters: the baud rate and the Minimum Pulse Duration (MPD) of the transceiver.

According to IrDA Standard Specification, for SIR (Serial IR) baud rates from 2.4 Kbit/s to 115.2 Kbit/s this nominal pulse duration is equal to 3/16 of a bit duration (at the selected baud rate). But, for all the baud rates a Minimum Pulse Duration is also specified. According to IrDA Standard, a Zero is represented by a light pulse, so the IrDA transceiver can't emit a light pulse shorter than the MPD. For SIR, the MPD is constant and equal to 1.41 µs.

But user must take into account the electrical MPD associated with the transceiver on the receiver path. Typically this value is 2.0 µs, but for some manufacturers MPD can go down to 1.0 µs.

In order to understand the meaning of IRSC bit, one must understand how the RX path works in IrDA mode.

When the UART is in IrDA mode, a Zero is not only detected by the state of the RXD_IR line, but also with the duration of the pulse. This pulse duration can be measured with 2 different clocks. In this case, clock is selected with the IRSC bit.

- If IRSC = 0, the clock used is the BRM clock.
- If IRSC = 1, the clock used is the UART internal clock (UART clock after the divider (RFDIV)).

In normal operation, IRSC=0. This means that at any time, the user must ensure that the frequency of BRM_clock is high enough to measure the pulse. The pulse must last at least 2 BRM clock cycles. If this condition is not fulfilled, IRSC must be set to 1.

Let's examine two examples, for a Minimum Pulse Duration equal to the MPD from the IrDA SIR specification (i.e., 1.41 µs).

1: Calculation of BRM Clock Period (Clock Period < 1.41 µs)

The user wants to receive IrDA data at 115.2 Kbit/s. The UBIR and UBMR registers are set in order to create the BRM_clock with a frequency of 16*baud rate = 16 * 115.2K = 1.843 MHz. But at the same time, in order to correctly detect the pulse, the user must be sure that 2* BRM_clock period is lower than 1.41 µs. Lets check:

BRM_clock period = 1/1843000 = 542 ns

So 2*BRM_clock period = 1.09 µs < 1.41 µs. It is fine.

2: Calculation of BRM Clock Period (Clock Period > 1.41 µs)

This time the user wants to receive at 19.2 Kbit/s. So, the BRM_clock is set to 16*19200 = 307.2 kHz. Let's check if 2* BRM_clock period < 1.41 µs:

1. BRM_clock period =1/307200 = 3.25 µs

   So 2*BRM_clock period = 6.50 µs >> 1.41 µs. It doesn't work.

   So, in this case, the BRM clock can't be used to measure the pulse duration and the user must select the UART internal clock by setting IRSC =1.

NOTE

Like for Escape character detection, when IR Special Case is enabled (IRSC=1), the UART must measure a duration. In order to do that, the user must fill the ONEMS register. Refer to Escape Sequence Detection.

16.2.6.4 IrDA interrupt

Serial infrared mode (SIR) uses an edge triggered interrupt flag IRINT (USR2[8]). When INVR =0, detection of a falling edge on the RXD pin asserts the IRINT bit. When INVR=1, detection of a rising edge on the RXD pin asserts the IRINT bit. When IRINT and ENIRI bits are both asserted, the *interrupt_uart* interrupt is asserted. Clear the IRINT bit by writing 1 to it. Writing 0 to the IRINT bit has no effect.

16.2.6.5 Conclusion about IrDA

Before using the UART in IrDA, the baud rate limit must be calculated. This baud rate limit will inform the user if IRSC bit has to be set or not.

Let's determine this limit:

As already described, if IRSC = 0, the following condition must always be fulfilled

$$2 \times BRMClockPeriod < MinPulseDuration$$

Figure 16-19. Calculation of Baud Rate

So,

$$BRMClockFrequency > \frac{2}{MPD}$$

So, knowing BRM_clock frequency = 16 * Baud Rate, we get:

$$BaudRate > \frac{1}{8 \times MinPulseDuration}$$

So, the user needs to set IRSC = 0 when:

- If Minimum Pulse Duration = 2.5 us and Baud Rate > 50 Kbit/s.
- If Minimum Pulse Duration = 2.0 us and Baud Rate > 62.5 Kbit/s.
- If Minimum Pulse Duration = 1.41 us and Baud Rate > 88.6 Kbit/s.

NOTE

For baud rates lower than the limit, IRSC must be set to1.

16.2.6.6 Programming IrDA Interface

16.2.6.6.1 High Speed

As an example, the following sequence can be used to program the IrDA interface in order to send and receive characters at 115.2 Kbit/s.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 115.2 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR)

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UTS = 0x0000
UFCR = 0x0981
TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
RXTL[5:0] = 0x01: Default value
UBIR = 0x0202
UBMR = 0x20BE Baud rate = 115.2 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000

         UCR4 = 0x8201
CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is ready to send a character as soon as there is a write into UTXD register. And an interrupt will be sent to Arm platform when a character is received.

16.2.6.6.2 Low Speed

This time, we keep the same assumptions but the speed is now 9.6 Kbit/s. So, this baud rate is below the limit (even with a Min. Pulse Duration of 2.5 us) and thus IRSC must be set to 1.

Assumptions:

- Input UART clock = 90 MHz
- Internal clock divider = 3 (divide Input UART clock by 3)
- Baud rate = 9.6 Kbit/s
- IrDA transceiver is not inverting on both channels: for Tx and Rx, a Zero is represented by a positive pulse, and a One is represented by no pulse (line stays low).
- Interrupt: Sent to Arm platform when 1 char is received into the Rx FIFO (RDR).

Registers values and Programming orders:

```
UCR1 = 0x0085
UCR1[7] = IREN = 1: Enable IR interface
UCR1[0] = UARTEN = 1: Enable UART
UFCR = 0x0981
UFCR[15:10] = TXTL[5:0] = 0x02: Default value
RFDIV[2:0] = 0x3: Divide Input UART clock by 3 (resulting internal clock  is 30 MHz)
UFCR[5:0] = RXTL[5:0] = 0x01: Default value
UBIR = 0x00FF
UBMR = 0xC354 Baud rate = 9.6 kbit/s with internal clock = 30 MHz
UCR2 = 0x4027
UCR2[14] = IRTS = 1: Ignore level of RTS input signal
UCR2[5] = WS = 1: Characters are 8-bit length
UCR2[2] = TXEN = 1: Enable Rx path
UCR2 [1] = RXEN = 1: Enable Tx path
UCR2[0] = SRST_B = 1: No software reset
UCR3 = 0x0000
UCR3[1] = INVT = 0: Positive pulse represents 0.
UCR4 = 0x8221
UCR4[15:10] = CTSTL[5:0] = 0x20: Default value
UCR4[9] = INVR = 1: Inverted Infrared Reception (because IrDA transceiver is not inverting)
UCR4[5] = IRSC = 1: Because data rate is below the limit and thus the UART internal clock is
used to measure the pulse duration.
UCR4[1] = DREN = 1: To enable RDR interrupt (sent when one char is received)
```

The UART is now ready to send a character as soon as there is a write into UTXD register. An interrupt will be sent to Arm platform when a character is received.

16.2.7 9-bit RS-485 Mode

16.2.7.1 Generalities

The UART provides a 9-bit mode to facilitate multidrop (RS-485) network communication. To enable this mode, set MDEN bit in the UMCR register to 1. When 9-bit RS-485 mode is enabled, UART transmitter can transmit the ninth bit (9^{th} bit) set by TXB8, and UART receiver can differentiate between data frames (9^{th} bit = 0) and address frames (9^{th} bit = 1).

The CTS_B pin can be used to control RS-485 output driver outside the chip.

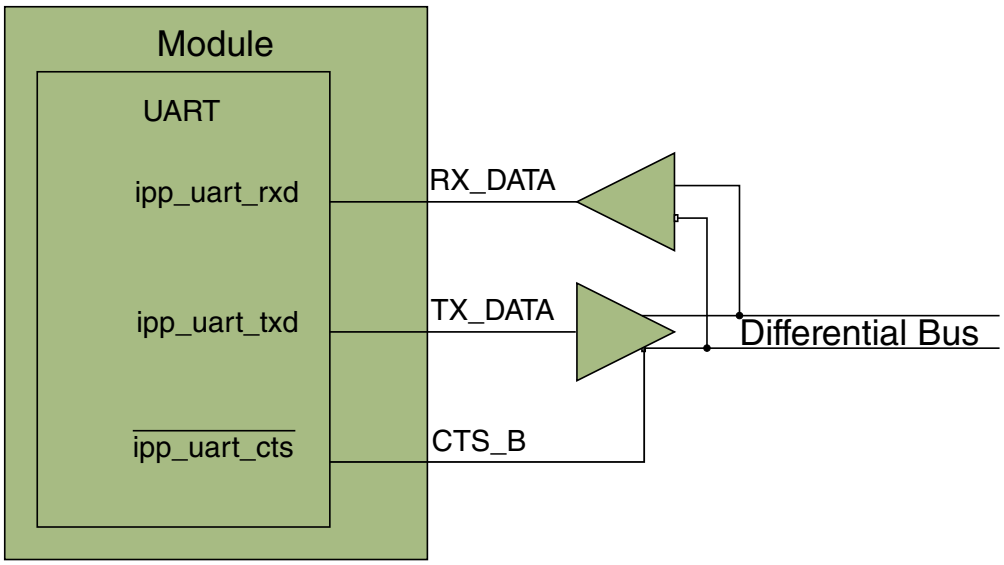


Figure 16-20. RS-485 driver connection (UART in DCE mode)

16.2.7.2 Transmit 9-bit RS-485 frames

To transmit 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable trasmitting the ninth data bit, set 8-bit data word size (WS=1), and write TXB8 (UMCR[2]) as the 9th bit (bit [8]) to be transmitted (write '0' to TXB8 to transmit a data frame, write '1' to transmit a address frame). The other data bit [7:0] is written to TxFIFO by writing to the UTXD same as normal RS-232 operation.

16.2.7.3 Receive 9-bit RS-485 frames

To receive 9-bit RS-485 frames, user need to enable parity (PREN=1) to enable receiving the ninth data bit, set 8-bit data word size (WS=1). The receiver will save the 9-bit data to RxFIFO, and user should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

There are two slave address detect modes, normal detect mode and automatic detect mode, and can be selected by SLAM (UMCR[1]).

16.2.7.3.1 RS-485 Slave Address Normal Detect Mode

To enable Normal Detect mode, clear SLAM (UMCR[1] to 0). The receiver ignores all data frames (9th bit = 0) until an address frame is received (9th bit = 1). At that time, the slave address detected (SAD = USR1[3]) bit is asserted and the *interrupt_uart* interrupt is

generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9th bit. The UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL) if receive ready DMA (RXDMAEN = UCR1[8]) request is enabled.

User should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

In this mode, once the UART has detected a 9th bit is equal to '1', it will always save the subsequent frames to RxFIFO. So the software must decide whether the address and data in RxFIFO are needed or not.

16.2.7.3.2 RS-485 Slave Address Automatic Detect Mode

To enable Automatic Detect Mode, set SLAM (UMCR[1]) to 1. The receiver tries to detect an address byte (frame 9th bit = 1) that matches the programmed SLADDR (UMCR[15:8]) character. If the received byte is a data or an address byte that does not match the programmed SLADDR character, the receiver will discard these data.

Once the UART receives a matching address byte, it will assert the slave address detected (SAD = USR1[3]) bit and the *interrupt_uart* interrupt will be generated (if SADEN = UMCR[3] = 1). The address byte and sebsequent bytes are all put into RxFIFO along with their 9th bit. If receive ready DMA(RXDMAEN = UCR1[8]) request is enabled, the UART will also generate DMA request *dma_req_rx* when the RxFIFO reaches the selected threshold (controlled by RXTL).

If another address byte is received and this address byte does not match SLADDR character, the receiver will discard the address byte and subsequent data byte. If the address byte again matches SLADDR character, the receiver will put this address byte and subsequent data byte in the RxFIFO along with their 9th bit.

User should read the 9th databit (bit [8]) by reading the PRERR (URXD[10]) bit, and read data bit [7:0] by reading the RX_DATA (URXD[7:0]).

See Initialization for 9-bit RS-485 programming guide.

16.2.8 Low Power Modes

These modes are controlled by the signals *doze_req* and *stop_req*. The control/status/data registers won't change when getting in/out of low power modes.

Table 16-11. UART Low Power State Operation

	Normal State (*doze_req* = 1'b0 & *stop_req* = 1'b0)	Doze State (*doze_req* = 1'b1)		Stop State (*stop_req* = 1'b1)
		DOZE bit = 0	DOZE bit = 1	
UART-Clock	ON	ON	ON	OFF
UART Serial / IrDA	ON	ON	OFF	OFF

16.2.8.1 UART Operation in System Doze Mode

While in Doze State (when *doze_req* input pin is set to 1'b1), the UART behavior depends on the DOZE (UCR1[1]) control bit.

While the DOZE bit is negated, the UART serial interface is enabled. While the system is in the Doze State, and the DOZE bit is asserted, the UART is disabled. If the Doze State is entered with the DOZE bit asserted while the UART serial interface was receiving or transmitting data, it will complete the receive/transmit of the current character and signal to the far-end transmitter/receiver to stop sending/receiving.

16.2.8.2 UART Operation in System Stop Mode

The internal baud rate clocks of the transmitter and receiver are gated off if the *stop_req* signal to UART is asserted. Even though the clocks at the input of the UART continue to run during system Stop mode, the UART will not do any transmission or reception.

The following UART interrupts wake the Arm platform processor from STOP mode:

- RTS (RTSD)
- IrDA Asynchronous WAKE (AIRINT)
- Asynchronous WAKE (AWAKE)
- RI (RIDELT in DTE mode only)
- DCD (DCDDELT in DTE mode only)
- DTR (DTRD in DCE mode only)
- DSR (DTRD in DTE mode only)

When an asynchronous WAKE (awake) interrupt exits the Arm platform from STOP mode, make sure that a dummy character is sent first because the first character may not be received correctly.

16.2.8.3 Power Saving Method in UART

The RXEN (UCR2[1]), TXEN (UCR2[2]) and UARTEN (UCR1[0]) bits are set by the user and provide software control of low-power modes.

Setting the UARTEN (UCR1[0]) bit to 0 shuts off the receiver and transmitter logic and the associated clocks.

If the UART is used only in transmit mode, UARTEN and TXEN must be set to 1. If the UART is used only in receive mode, UARTEN and RXEN must be set to 1. Setting TXEN or RXEN to 0 allows to save a lot of power.

16.2.9 UART Operation in System Debug State

The bit UTS [11] controls whether the UART will respond to the input signal *debug_req*, or whether it will continue to run as normal.

If the UART is programmed to respond to *debug_req*:

1. The UART will halt all operations upon detecting the *debug_req* input.
2. A transfer in progress, either to/from a core (using the IP Bus interface) or to/from an external device, will be completed before halting. This means a single byte/word transfer, not an entire FIFO. Reception of any further data from an external device will be disabled.
3. Internal registers will continue to be writable and readable using the IP Bus interface. A read will leave the contents unaffected.
4. The RX FIFO is affected in debug mode in the following way:
   - All writes into the RX FIFO are prevented.
   - The bit RXDBG (UTS[9]) is used to select the readability of the RX FIFO during debug mode:

     RXDBG = 0: hold the read pointer at the location it had upon entering debug mode, and URXD register returns only the data value at that location, no matter how many reads attempted.

RXDBG = 1, selectable at any time: Allow to read the characters received in Rx FIFO. It will not be possible to re-read previously read locations, nor will it be possible to readjust the read pointer to the value it had prior to entering debug mode.

16.2.10 Reset

This section describes how to reset the block and explains special requirements related to reset.

16.2.10.1 Hardware reset

All of registers, FIFOs, state machines and sequential elements can be reset to their initial values by hardware reset or power on reset.

16.2.10.2 Software reset

The status registers USR1 and USR2, BRM registers UBIR and UBMR, TxFIFO and RxFIFO, and transmitter and receiver state machines can be reset by software reset. Internal logic will keep the software reset asserted for about 4 *module_clock* cycles.

Programmer can follow the following software reset sequence:

1. Clear the SRST_B bit (UCR2[0])
2. Wait for software reset complete: poll SOFTRST bit (UTS[0]) until it is 0.
3. Re-program baud rate registers: Re-write UBIR and UBMR.

16.2.11 Transfer Error

The UART can generate a transfer error on the peripheral bus in the following cases:

- Core is writing into a read-only register.
- Core is accessing (read or write) an unused location within the assigned address space reserved to UART.
- Core is writing into UTXD register with transmit interface disabled (TXEN=0 or UARTEN=0)
- Core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0)

16.2.12 Functional Timing

This section includes timing diagrams for functional signaling.

16.2.12.1 IrDA Mode

According to IrDA specification, the low speed (115.2Kbit/s and below) IR frame format is compatible with UART frame.

In this figure, an example data 0x65 is used.

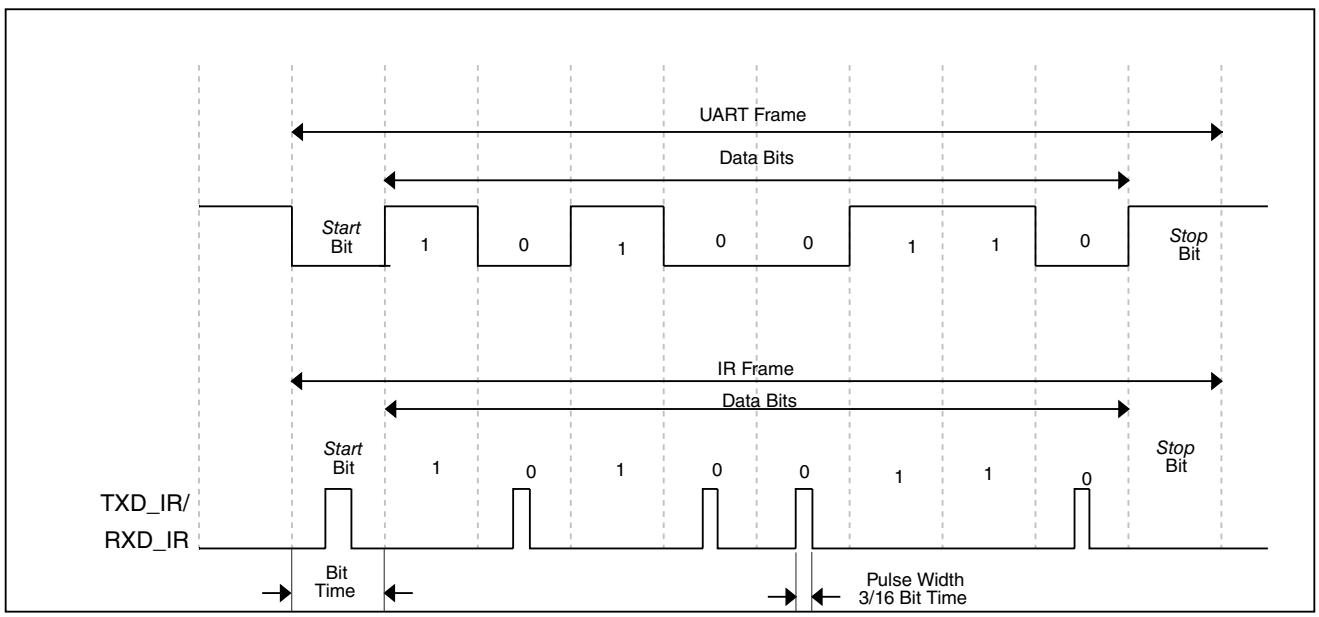


Figure 16-21. Timing diagram of Low Speed IR (<=115.2 Kbit/s) Data Line

16.2.13 Initialization

16.2.13.1 Programming the UART in RS-232 mode

As an example, the following sequence can be used to program the UART in order to send and receive characters in RS-232 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 921.6 Kbps

- Data bits = 8 bits
- Parity = Even
- Stop bits = 1 bit
- Flow control = Hardware

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x2127

   Set hardware flow control, data format and enable transmitter and receiver.

3. UCR3 = 0x0704

   Set UCR3[RXDMUXSEL] = 1.

4. UCR4 = 0x7C00

   Set CTS trigger level to 31,

5. UFCR = 0x089E

   Set internal clock divider = 5 (divide input uart clock by 5). So the reference clock is 100 MHz/5 = 20 MHz.

   Set TXTL = 2 and RXTL = 30.

6. UBIR = 0x08FF
7. UBMR = 0x0C34

   In the above two steps, set baud rate to 921.6Kbps based on the 20MHz reference clock.

8. UCR1 = 0x2201

   Enable the TRDY and RRDY interrupts.

9. UMCR = 0x0000

   UMCR stay at default value 0x0000

Interrupt service routine for the transmitter:

- Write characters into UTXD

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2. Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Read characters from URXD

The RRDY interrupt will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

16.2.13.2 Programming the UART in 9-bit RS-485 mode

As an example, the following sequence can be used to program the UART in order to send and receive frames in RS-485 mode.

Assumptions:

- Input uart clock = 100 MHz
- Baud rate = 5 Mbps

Main program:

1. UCR1 = 0x0001

   Enable the UART.

2. UCR2 = 0x4127

   Set software flow control ($\overline{\text{CTS}}$ pin is controlled by UCR2[12]), enable parity(enable 9^{th} bit rxd/txd), 8-bit word size , and enable transmitter and receiver.

3. UCR4 = 0x7C00

   Set CTS trigger level to 31,

4. UFCR = 0x0A9E

   Set RFDIV = 5 (divide input uart clock by 1), so the reference clock is 100 MHz. Set UART in DCE mode (RS-485 driver connection outside the chip is the same as Figure 16-20)

   Set TXTL = 2 and RXTL = 30.

5. UBIR = 0x0003
6. UBMR = 0x0004

   In the above two steps, set baud rate to 5 Mbps based on the 100 MHz reference clock.

7. UCR1 = 0x2001 when UART as a master ,

   or UCR1 = 0x0201 (or 0x0101) when UART as a slave.

Enable TRDY interrupt when UART as a master, enable RRDY interrupt or DMA request when UART as a slave.

8. UMCR = 0xA50B

   Enable 9-bit RS-485 mode, enable SAD interrupt, set automatic slave address detect mode, set slave address is 0xA5.

Interrupt service routine for the transmitter:

- Transmit data: write its ninth bit (bit[8]) to UMCR[2], write its bit [7:0] into UTXD[7:0]

The TRDY interrupt will be automatically de-asserted when the data level of the TxFIFO exceeds the TXTL=2.

Note: For the first time the interrupt may be de-asserted after 4 characters are written into the TxFIFO because of the shift register.

Interrupt service routine for the receiver:

- Receive data: read its ninth bit (bit[8]) from URXD[10] , read its bit [7:0] from URXD[7:0].

Note: in RS-485 mode, URXD[10] bit is not the parity error, instead it holds the ninth bit (bit[8]) of the received data.

The SAD interrupt can not de-assert automatically, it needs MCU write 1 to USR1[3] to clear it . The RRDY interrupt or DMA request will be automatically de-asserted when the data level of the RxFIFO is below the RXTL=30.

16.2.14 References

- EIA/TIA-232-F Interface Standard

  http://www.eia.org, http://www.tiaonline.org/standards

- IrDA Standard

  http://www.irda.org

16.2.15 UART Memory Map/Register Definition

UART supports 8-bit, 16-bit and 32-bit accesses to 32-bit memory-mapped addresses. Any access to unmapped memory location will yield a transfer error.

All registers except the ONEMS described in this section are 16-bit registers. The ONEMS register is a 24-bit register.

- For 32-bit write accesses, the upper two bytes will not be taken into account.
- For 32-bit read accesses the upper two bytes will return 0.

The ONEMS register is expanded from 16 bits to 24 bits in order to support the high frequency of the BRM internal clock *ref_clk* (*module_clock* after divider). The ONEMS register can be accessed as 8 bits, 16 bits or 32 bits.

- For 32-bit write accesses, the most significant byte of the ONEMS will be discarded.
- For 32-bit read accesses, the most significant byte of the ONEMS will be read as 0.

UART memory map

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3086_0000	UART Receiver Register (UART1_URXD)	32	R	0000_0000h	16.2.15.1/ 6331
3086_0040	UART Transmitter Register (UART1_UTXD)	32	W	0000_0000h	16.2.15.2/ 6333
3086_0080	UART Control Register 1 (UART1_UCR1)	32	R/W	0000_0000h	16.2.15.3/ 6334
3086_0084	UART Control Register 2 (UART1_UCR2)	32	R/W	0000_0001h	16.2.15.4/ 6336
3086_0088	UART Control Register 3 (UART1_UCR3)	32	R/W	0000_0700h	16.2.15.5/ 6339
3086_008C	UART Control Register 4 (UART1_UCR4)	32	R/W	0000_8000h	16.2.15.6/ 6341
3086_0090	UART FIFO Control Register (UART1_UFCR)	32	R/W	0000_0801h	16.2.15.7/ 6343
3086_0094	UART Status Register 1 (UART1_USR1)	32	R/W	0000_2040h	16.2.15.8/ 6345
3086_0098	UART Status Register 2 (UART1_USR2)	32	R/W	0000_4028h	16.2.15.9/ 6348
3086_009C	UART Escape Character Register (UART1_UESC)	32	R/W	0000_002Bh	16.2.15.10/ 6350
3086_00A0	UART Escape Timer Register (UART1_UTIM)	32	R/W	0000_0000h	16.2.15.11/ 6350
3086_00A4	UART BRM Incremental Register (UART1_UBIR)	32	R/W	0000_0000h	16.2.15.12/ 6351
3086_00A8	UART BRM Modulator Register (UART1_UBMR)	32	R/W	0000_0000h	16.2.15.13/ 6351
3086_00AC	UART Baud Rate Count Register (UART1_UBRC)	32	R	0000_0004h	16.2.15.14/ 6352

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3086_00B0	UART One Millisecond Register (UART1_ONEMS)	32	R/W	0000_0000h	16.2.15.15/ 6353
3086_00B4	UART Test Register (UART1_UTS)	32	R/W	0000_0060h	16.2.15.16/ 6354
3086_00B8	UART RS-485 Mode Control Register (UART1_UMCR)	32	R/W	0000_0000h	16.2.15.17/ 6355
3088_0000	UART Receiver Register (UART3_URXD)	32	R	0000_0000h	16.2.15.1/ 6331
3088_0040	UART Transmitter Register (UART3_UTXD)	32	W	0000_0000h	16.2.15.2/ 6333
3088_0080	UART Control Register 1 (UART3_UCR1)	32	R/W	0000_0000h	16.2.15.3/ 6334
3088_0084	UART Control Register 2 (UART3_UCR2)	32	R/W	0000_0001h	16.2.15.4/ 6336
3088_0088	UART Control Register 3 (UART3_UCR3)	32	R/W	0000_0700h	16.2.15.5/ 6339
3088_008C	UART Control Register 4 (UART3_UCR4)	32	R/W	0000_8000h	16.2.15.6/ 6341
3088_0090	UART FIFO Control Register (UART3_UFCR)	32	R/W	0000_0801h	16.2.15.7/ 6343
3088_0094	UART Status Register 1 (UART3_USR1)	32	R/W	0000_2040h	16.2.15.8/ 6345
3088_0098	UART Status Register 2 (UART3_USR2)	32	R/W	0000_4028h	16.2.15.9/ 6348
3088_009C	UART Escape Character Register (UART3_UESC)	32	R/W	0000_002Bh	16.2.15.10/ 6350
3088_00A0	UART Escape Timer Register (UART3_UTIM)	32	R/W	0000_0000h	16.2.15.11/ 6350
3088_00A4	UART BRM Incremental Register (UART3_UBIR)	32	R/W	0000_0000h	16.2.15.12/ 6351
3088_00A8	UART BRM Modulator Register (UART3_UBMR)	32	R/W	0000_0000h	16.2.15.13/ 6351
3088_00AC	UART Baud Rate Count Register (UART3_UBRC)	32	R	0000_0004h	16.2.15.14/ 6352
3088_00B0	UART One Millisecond Register (UART3_ONEMS)	32	R/W	0000_0000h	16.2.15.15/ 6353
3088_00B4	UART Test Register (UART3_UTS)	32	R/W	0000_0060h	16.2.15.16/ 6354
3088_00B8	UART RS-485 Mode Control Register (UART3_UMCR)	32	R/W	0000_0000h	16.2.15.17/ 6355
3089_0000	UART Receiver Register (UART2_URXD)	32	R	0000_0000h	16.2.15.1/ 6331
3089_0040	UART Transmitter Register (UART2_UTXD)	32	W	0000_0000h	16.2.15.2/ 6333

Table continues on the next page...

UART memory map (continued)

Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
3089_0080	UART Control Register 1 (UART2_UCR1)	32	R/W	0000_0000h	16.2.15.3/ 6334
3089_0084	UART Control Register 2 (UART2_UCR2)	32	R/W	0000_0001h	16.2.15.4/ 6336
3089_0088	UART Control Register 3 (UART2_UCR3)	32	R/W	0000_0700h	16.2.15.5/ 6339
3089_008C	UART Control Register 4 (UART2_UCR4)	32	R/W	0000_8000h	16.2.15.6/ 6341
3089_0090	UART FIFO Control Register (UART2_UFCR)	32	R/W	0000_0801h	16.2.15.7/ 6343
3089_0094	UART Status Register 1 (UART2_USR1)	32	R/W	0000_2040h	16.2.15.8/ 6345
3089_0098	UART Status Register 2 (UART2_USR2)	32	R/W	0000_4028h	16.2.15.9/ 6348
3089_009C	UART Escape Character Register (UART2_UESC)	32	R/W	0000_002Bh	16.2.15.10/ 6350
3089_00A0	UART Escape Timer Register (UART2_UTIM)	32	R/W	0000_0000h	16.2.15.11/ 6350
3089_00A4	UART BRM Incremental Register (UART2_UBIR)	32	R/W	0000_0000h	16.2.15.12/ 6351
3089_00A8	UART BRM Modulator Register (UART2_UBMR)	32	R/W	0000_0000h	16.2.15.13/ 6351
3089_00AC	UART Baud Rate Count Register (UART2_UBRC)	32	R	0000_0004h	16.2.15.14/ 6352
3089_00B0	UART One Millisecond Register (UART2_ONEMS)	32	R/W	0000_0000h	16.2.15.15/ 6353
3089_00B4	UART Test Register (UART2_UTS)	32	R/W	0000_0060h	16.2.15.16/ 6354
3089_00B8	UART RS-485 Mode Control Register (UART2_UMCR)	32	R/W	0000_0000h	16.2.15.17/ 6355
30A6_0000	UART Receiver Register (UART4_URXD)	32	R	0000_0000h	16.2.15.1/ 6331
30A6_0040	UART Transmitter Register (UART4_UTXD)	32	W	0000_0000h	16.2.15.2/ 6333
30A6_0080	UART Control Register 1 (UART4_UCR1)	32	R/W	0000_0000h	16.2.15.3/ 6334
30A6_0084	UART Control Register 2 (UART4_UCR2)	32	R/W	0000_0001h	16.2.15.4/ 6336
30A6_0088	UART Control Register 3 (UART4_UCR3)	32	R/W	0000_0700h	16.2.15.5/ 6339
30A6_008C	UART Control Register 4 (UART4_UCR4)	32	R/W	0000_8000h	16.2.15.6/ 6341
30A6_0090	UART FIFO Control Register (UART4_UFCR)	32	R/W	0000_0801h	16.2.15.7/ 6343

Table continues on the next page...

UART memory map (continued)

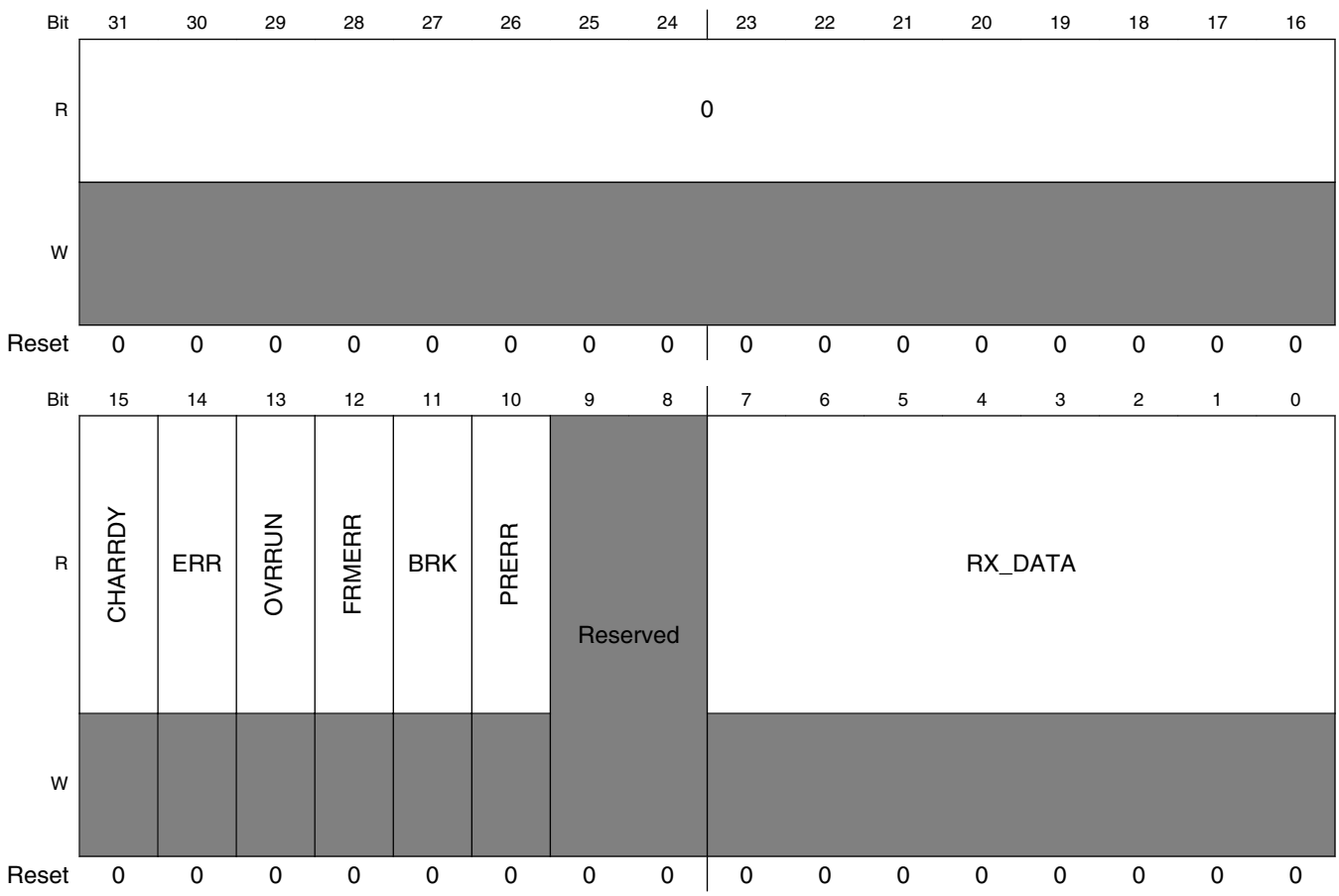
Absolute address (hex)	Register name	Width (in bits)	Access	Reset value	Section/ page
30A6_0094	UART Status Register 1 (UART4_USR1)	32	R/W	0000_2040h	16.2.15.8/ 6345
30A6_0098	UART Status Register 2 (UART4_USR2)	32	R/W	0000_4028h	16.2.15.9/ 6348
30A6_009C	UART Escape Character Register (UART4_UESC)	32	R/W	0000_002Bh	16.2.15.10/ 6350
30A6_00A0	UART Escape Timer Register (UART4_UTIM)	32	R/W	0000_0000h	16.2.15.11/ 6350
30A6_00A4	UART BRM Incremental Register (UART4_UBIR)	32	R/W	0000_0000h	16.2.15.12/ 6351
30A6_00A8	UART BRM Modulator Register (UART4_UBMR)	32	R/W	0000_0000h	16.2.15.13/ 6351
30A6_00AC	UART Baud Rate Count Register (UART4_UBRC)	32	R	0000_0004h	16.2.15.14/ 6352
30A6_00B0	UART One Millisecond Register (UART4_ONEMS)	32	R/W	0000_0000h	16.2.15.15/ 6353
30A6_00B4	UART Test Register (UART4_UTS)	32	R/W	0000_0060h	16.2.15.16/ 6354
30A6_00B8	UART RS-485 Mode Control Register (UART4_UMCR)	32	R/W	0000_0000h	16.2.15.17/ 6355

16.2.15.1 UART Receiver Register (UARTx_URXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is reading URXD register with receive interface disabled (RXEN=0 or UARTEN=0).

Address: Base address + 0h offset



UARTx_URXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 CHARRDY	Character Ready. This read-only bit indicates an invalid read when the FIFO becomes empty and software tries to read the same old data. This bit should not be used for polling for data written to the RX FIFO. 0 Character in RX_DATA field and associated flags are invalid. 1 Character in RX_DATA field and associated flags valid and ready for reading.

Table continues on the next page...

UARTx_URXD field descriptions (continued)

Field	Description
14 ERR	**Error Detect.** Indicates whether the character present in the RX_DATA field has an error (OVRRUN, FRMERR, BRK or PRERR) status. The ERR bit is updated and valid for each received character. 0 No error status was detected 1 An error status was detected
13 OVRRUN	**Receiver Overrun.** This read-only bit, when HIGH, indicates that the corresponding character was stored in the last position (32nd) of the Rx FIFO. Even if a 33rd character has not been detected, this bit will be set to '1' for the 32nd character. 0 No RxFIFO overrun was detected 1 A RxFIFO overrun was detected
12 FRMERR	**Frame Error.** Indicates whether the current character had a framing error (a missing stop bit) and is possibly corrupted. FRMERR is updated for each character read from the RxFIFO. 0 The current character has no framing error 1 The current character has a framing error
11 BRK	**BREAK Detect.** Indicates whether the current character was detected as a BREAK character. The data bits and the stop bit are all 0. The FRMERR bit is set when BRK is set. When odd parity is selected, PRERR is also set when BRK is set. BRK is valid for each character read from the RxFIFO. 0 The current character is not a BREAK character 1 The current character is a BREAK character
10 PRERR	**In RS-485 mode, it holds the ninth data bit (bit [8]) of received 9-bit RS-485 data** **In RS232/IrDA mode, it is the Parity Error flag**. Indicates whether the current character was detected with a parity error and is possibly corrupted. PRERR is updated for each character read from the RxFIFO. When parity is disabled, PRERR always reads as 0. 0 = No parity error was detected for data in the RX_DATA field 1 = A parity error was detected for data in the RX_DATA field
9–8 -	This field is reserved. **Reserved**
RX_DATA	**Received Data**. Holds the received character. In 7-bit mode, the most significant bit (MSB) is forced to 0. In 8-bit mode, all bits are active.

16.2.15.2 UART Transmitter Register (UARTx_UTXD)

NOTE

The UART will yield a transfer error on the peripheral bus when core is writing into UART_URXD register with transmit interface disabled (TXEN=0 or UARTEN=0).

Memory space between UART_URXD and UART_UTXD registers is reserved. Any read or write access to this space will be considered as an invalid access and yield a transfer error.

Address: Base address + 40h offset

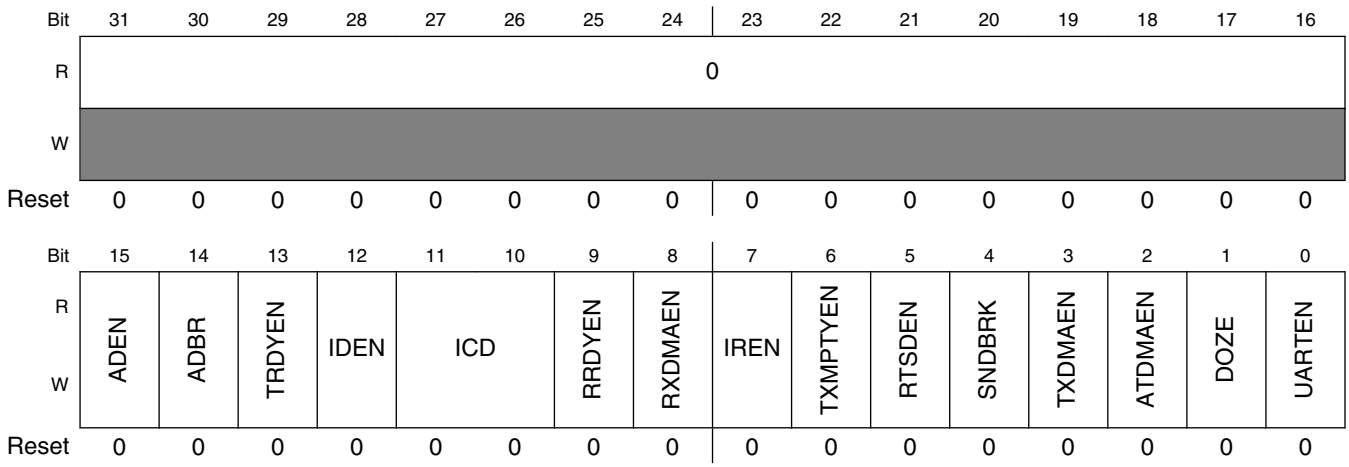
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																0															
W																									TX_DATA							
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UTXD field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 Reserved	This read-only field is reserved and always has the value 0.
TX_DATA	**Transmit Data**. Holds the parallel transmit data inputs. In 7-bit mode, D7 is ignored. In 8-bit mode, all bits are used. Data is transmitted least significant bit (LSB) first. A new character is transmitted when the TX_DATA field is written. The TX_DATA field must be written only when the TRDY bit is high to ensure that corrupted data is not sent.

16.2.15.3 UART Control Register 1 (UARTx_UCR1)

Address: Base address + 80h offset



UARTx_UCR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADEN	**Automatic Baud Rate Detection Interrupt Enable**. Enables/Disables the automatic baud rate detect complete (ADET) bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the automatic baud rate detection interrupt 1 Enable the automatic baud rate detection interrupt
14 ADBR	**Automatic Detection of Baud Rate**. Enables/Disables automatic baud rate detection. When the ADBR bit is set and the ADET bit is cleared, the receiver detects the incoming baud rate automatically. The ADET flag is set when the receiver verifies that the incoming baud rate is detected properly by detecting an ASCII character "A" or "a" (0x41 or 0x61). 0 Disable automatic detection of baud rate 1 Enable automatic detection of baud rate
13 TRDYEN	**Transmitter Ready Interrupt Enable**. Enables/Disables the transmitter Ready Interrupt (TRDY) when the transmitter has one or more slots available in the TxFIFO. The fill level in the TXFIFO at which an interrupt is generated is controlled by TxTL bits. When TRDYEN is negated, the transmitter ready interrupt is disabled. **NOTE:** An interrupt will be issued as long as TRDYEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TRDY interrupt. 0 Disable the transmitter ready interrupt 1 Enable the transmitter ready interrupt
12 IDEN	**Idle Condition Detected Interrupt Enable**. Enables/Disables the IDLE bit to generate an interrupt (*interrupt_uart* = 0). 0 Disable the IDLE interrupt 1 Enable the IDLE interrupt

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
11–10 ICD	**Idle Condition Detect**. Controls the number of frames RXD is allowed to be idle before an idle condition is reported. 00 Idle for more than 4 frames 01 Idle for more than 8 frames 10 Idle for more than 16 frames 11 Idle for more than 32 frames
9 RRDYEN	**Receiver Ready Interrupt Enable**. Enables/Disables the RRDY interrupt when the RxFIFO contains data. The fill level in the RxFIFO at which an interrupt is generated is controlled by the RXTL bits. When RRDYEN is negated, the receiver ready interrupt is disabled. 0 Disables the RRDY interrupt 1 Enables the RRDY interrupt
8 RXDMAEN	**Receive Ready DMA Enable**. Enables/Disables the receive DMA request *dma_req_rx* when the receiver has data in the RxFIFO. The fill level in the RxFIFO at which a DMA request is generated is controlled by the RXTL bits. When negated, the receive DMA request is disabled. 0 Disable DMA request 1 Enable DMA request
7 IREN	**Infrared Interface Enable**. Enables/Disables the IR interface. See the IR interface description in Infrared Interface, for more information. Note: MDEN(UMCR[0]) must be cleared to 0 when using IrDA interface. See Table 16-3 0 Disable the IR interface 1 Enable the IR interface
6 TXMPTYEN	**Transmitter Empty Interrupt Enable**. Enables/Disables the transmitter FIFO empty (TXFE) interrupt. *interrupt_uart*. When negated, the TXFE interrupt is disabled. **NOTE:** An interrupt will be issued as long as TXMPTYEN and TXFE are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXFE interrupt. 0 Disable the transmitter FIFO empty interrupt 1 Enable the transmitter FIFO empty interrupt
5 RTSDEN	**RTS Delta Interrupt Enable**. Enables/Disables the RTSD interrupt. The current status of the RTS_B pin is read in the RTSS bit. 0 Disable RTSD interrupt 1 Enable RTSD interrupt
4 SNDBRK	**Send BREAK**. Forces the transmitter to send a BREAK character. The transmitter finishes sending the character in progress (if any) and sends BREAK characters until SNDBRK is reset. Because the transmitter samples SNDBRK after every bit is transmitted, it is important that SNDBRK is asserted high for a sufficient period of time to generate a valid BREAK. After the BREAK transmission completes, the UART transmits 2 mark bits. The user can continue to fill the TxFIFO and any characters remaining are transmitted when the BREAK is terminated. 0 Do not send a BREAK character 1 Send a BREAK character (continuous 0s)
3 TXDMAEN	**Transmitter Ready DMA Enable**. Enables/Disables the transmit DMA request *dma_req_tx* when the transmitter has one or more slots available in the TxFIFO. The fill level in the TxFIFO that generates the *dma_req_tx* is controlled by the TXTL bits.

Table continues on the next page...

UARTx_UCR1 field descriptions (continued)

Field	Description
	NOTE: A DMA request will be issued as long as TXDMAEN and TRDY are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the transmit DMA request. 0 Disable transmit DMA request 1 Enable transmit DMA request
2 ATDMAEN	**Aging DMA Timer Enable**. Enables/Disables the receive DMA request *dma_req_rx* for the aging timer interrupt (triggered with AGTIM flag in USR1[8]). 0 Disable AGTIM DMA request 1 Enable AGTIM DMA request
1 DOZE	**DOZE**. Determines the UART enable condition in the DOZE state. When *doze_req* input pin is at '1', (the Arm Platform executes a doze instruction and the system is placed in the Doze State), the DOZE bit affects operation of the UART. While in the Doze State, if this bit is asserted, the UART is disabled. See the description in Low Power Modes. 0 The UART is enabled when in DOZE state 1 The UART is disabled when in DOZE state
0 UARTEN	**UART Enable**. Enables/Disables the UART. If UARTEN is negated in the middle of a transmission, the transmitter stops and pulls the TXD line to a logic 1. UARTEN must be set to 1 before any access to UTXD and URXD registers, otherwise a transfer error is returned. This bit can be set to 1 along with other bits in this register. There is no restriction to the sequence of programing this bit and other control registers. 0 Disable the UART 1 Enable the UART

16.2.15.4 UART Control Register 2 (UARTx_UCR2)

Address: Base address + 84h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R W	ESCI	IRTS	CTSC	CTS	ESCEN	RTEC		PREN	PROE	STPB	WS	RTSEN	ATEN	TXEN	RXEN	SRST
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

UARTx_UCR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ESCI	**Escape Sequence Interrupt Enable**. Enables/Disables the ESCF bit to generate an interrupt. 0 Disable the escape sequence interrupt 1 Enable the escape sequence interrupt
14 IRTS	**Ignore RTS Pin**. Forces the RTS input signal presented to the transmitter to always be asserted (set to low), effectively ignoring the external pin. When in this mode, the RTS pin serves as a general purpose input. 0 Transmit only when the RTS pin is asserted 1 Ignore the RTS pin
13 CTSC	CTS **Pin Control**. Controls the operation of the CTS_B module output. When CTSC is asserted, the CTS_B module output is controlled by the receiver. When the RxFIFO is filled to the level of the programmed trigger level and the start bit of the overflowing character (TRIGGER LEVEL + 1) is validated, the CTS_B module output is negated to indicate to the far-end transmitter to stop transmitting. When the trigger level is programmed for less than 32, the receiver continues to receive data until the RxFIFO is full. When the CTSC bit is negated, the CTS_B module output is controlled by the CTS bit. On reset, because CTSC is cleared to 0, the CTS_B pin is controlled by the CTS bit, which again is cleared to 0 on reset. This means that on reset the CTS_B signal is negated. 0 The CTS_B pin is controlled by the CTS bit 1 The CTS_B pin is controlled by the receiver
12 CTS	**Clear to Send**. Controls the CTS_B pin when the CTSC bit is negated. CTS has no function when CTSC is asserted. 0 The CTS_B pin is high (inactive) 1 The CTS_B pin is low (active)
11 ESCEN	**Escape Enable**. Enables/Disables the escape sequence detection logic. 0 Disable escape sequence detection 1 Enable escape sequence detection
10–9 RTEC	**Request to Send Edge Control**. Selects the edge that triggers the RTS interrupt. This has no effect on the RTS delta interrupt. RTEC has an effect only when RTSEN = 1 (see Table 16-6). 00 Trigger interrupt on a rising edge 01 Trigger interrupt on a falling edge 1X Trigger interrupt on any edge
8 PREN	**Parity Enable**. Enables/Disables the parity generator in the transmitter and parity checker in the receiver. When PREN is asserted, the parity generator and checker are enabled, and disabled when PREN is negated. 0 Disable parity generator and checker 1 Enable parity generator and checker
7 PROE	**Parity Odd/Even**. Controls the sense of the parity generator and checker. When PROE is high, odd parity is generated and expected. When PROE is low, even parity is generated and expected. PROE has no function if PREN is low. 0 Even parity 1 Odd parity

Table continues on the next page...

UARTx_UCR2 field descriptions (continued)

Field	Description
6 STPB	**Stop**. Controls the number of stop bits after a character. When STPB is low, 1 stop bit is sent. When STPB is high, 2 stop bits are sent. STPB also affects the receiver. 0 The transmitter sends 1 stop bit. The receiver expects 1 or more stop bits. 1 The transmitter sends 2 stop bits. The receiver expects 2 or more stop bits.
5 WS	**Word Size**. Controls the character length. When WS is high, the transmitter and receiver are in 8-bit mode. When WS is low, they are in 7-bit mode. The transmitter ignores bit 7 and the receiver sets bit 7 to 0. WS can be changed in-between transmission (reception) of characters, however not when a transmission (reception) is in progress, in which case the length of the current character being transmitted (received) is unpredictable. 0 7-bit transmit and receive character length (not including START, STOP or PARITY bits) 1 8-bit transmit and receive character length (not including START, STOP or PARITY bits)
4 RTSEN	**Request to Send Interrupt Enable.** Controls the RTS edge sensitive interrupt. When RTSEN is asserted and the programmed edge is detected on the RTS_B pin (the RTSF bit is asserted), an interrupt will be generated on the *interrupt_uart* pin. (See Table 16-6.) 0 Disable request to send interrupt 1 Enable request to send interrupt
3 ATEN	**Aging Timer Enable**. This bit is used to enable the aging timer interrupt (triggered with AGTIM) 0 AGTIM interrupt disabled 1 AGTIM interrupt enabled
2 TXEN	**Transmitter Enable**. Enables/Disables the transmitter. When TXEN is negated the transmitter is disabled and idle. When the UARTEN and TXEN bits are set the transmitter is enabled. If TXEN is negated in the middle of a transmission, the UART disables the transmitter immediately, and starts marking 1s. The transmitter FIFO cannot be written when this bit is cleared. 0 Disable the transmitter 1 Enable the transmitter
1 RXEN	**Receiver Enable**. Enables/Disables the receiver. When the receiver is enabled, if the RXD input is already low, the receiver does not recognize BREAK characters, because it requires a valid 1-to-0 transition before it can accept any character. 0 Disable the receiver 1 Enable the receiver
0 SRST	**Software Reset**. Once the software writes 0 to SRST_B, the software reset remains active for 4 *module_clock* cycles before the hardware deasserts SRST_B. The software can only write 0 to SRST_B. Writing 1 to SRST_B is ignored. 0 Reset the transmit and receive state machines, all FIFOs and register USR1, USR2, UBIR, UBMR, UBRC , URXD, UTXD and UTS[6-3]. 1 No reset

16.2.15.5 UART Control Register 3 (UARTx_UCR3)

Address: Base address + 88h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	DPEC		DTREN	PARERREN	FRAERREN	DSR	DCD	RI	ADNIMP	RXDSEN	AIRINTEN	AWAKEN	DTRDEN	RXDMUXSEL	INVT	ACIEN
W																
Reset	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0

UARTx_UCR3 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–14 DPEC	This bit is not used in this chip.
13 DTREN	This bit is not used in this chip.
12 PARERREN	**Parity Error Interrupt Enable. Enables/Disables the interrupt. When asserted, PARERREN causes the PARITYERR bit to generate an interrupt.** 0 Disable the parity error interrupt 1 Enable the parity error interrupt
11 FRAERREN	**Frame Error Interrupt Enable. Enables/Disables the interrupt. When asserted, FRAERREN causes the FRAMERR bit to generate an interrupt.** 0 Disable the frame error interrupt 1 Enable the frame error interrupt
10 DSR	This bit is not used in this chip.
9 DCD	This bit is not used in this chip.
8 RI	This bit is not used in this chip.
7 ADNIMP	**Autobaud Detection Not Improved-. Disables new features of autobaud detection (See Baud Rate Automatic Detection Protocol, for more details).** 0 Autobaud detection new features selected 1 Keep old autobaud detection mechanism

Table continues on the next page...

UARTx_UCR3 field descriptions (continued)

Field	Description
6 RXDSEN	Receive Status Interrupt Enable. Controls the receive status interrupt (*interrupt_uart*). When this bit is enabled and RXDS status bit is set, the interrupt *interrupt_uart* will be generated. 0 Disable the RXDS interrupt 1 Enable the RXDS interrupt
5 AIRINTEN	Asynchronous IR WAKE Interrupt Enable. Controls the asynchronous IR WAKE interrupt. An interrupt is generated when AIRINTEN is asserted and a pulse is detected on the RXD pin. 0 Disable the AIRINT interrupt 1 Enable the AIRINT interrupt
4 AWAKEN	Asynchronous WAKE Interrupt Enable. Controls the asynchronous WAKE interrupt. An interrupt is generated when AWAKEN is asserted and a falling edge is detected on the RXD pin. 0 Disable the AWAKE interrupt 1 Enable the AWAKE interrupt
3 DTRDEN	This bit is not used in this chip.
2 RXDMUXSEL	RXD Muxed Input Selected. Selects proper input pins for serial and Infrared input signal. **NOTE:** In this chip, UARTs are used in MUXED mode, so that this bit should always be set.
1 INVT	Invert TXD output in RS-232/RS-485 mode, set TXD active level in IrDA mode. In RS232/RS-485 mode(UMCR[0] = 1), if this bit is set to 1, the TXD output is inverted before transmitted. In **IrDA mode**, when INVT is cleared, the infrared logic block transmits a positive IR 3/16 pulse for all 0s and 0s are transmitted for 1s. When INVT is set (INVT = 1), the infrared logic block transmits an active low or negative infrared 3/16 pulse for all 0s and 1s are transmitted for 1s. 0 **TXD is not inverted** 1 **TXD is inverted** 0 **TXD** Active low transmission 1 **TXD** Active high transmission
0 ACIEN	**Autobaud Counter Interrupt Enable.** This bit is used to enable the autobaud counter stopped interrupt (triggered with ACST (USR2[11]). 0 ACST interrupt disabled 1 ACST interrupt enabled

16.2.15.6 UART Control Register 4 (UARTx_UCR4)

Address: Base address + 8Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
R	0															
W																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	CTSTL						INVR	ENIRI	WKEN	IDDMAEN	IRSC	LPBYP	TCEN	BKEN	OREN	DREN
W																
Reset	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UCR4 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 CTSTL	**CTS Trigger Level**. Controls the threshold at which the CTS_B pin is deasserted by the RxFIFO. After the trigger level is reached and the CTS_B pin is deasserted, the RxFIFO continues to receive data until it is full. The CTSTL bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 1 characters in the RxFIFO ... — ... — 100000 32 characters in the RxFIFO (maximum)
9 INVR	**Invert RXD input in RS-232/RS-485 Mode, d**etermine **RXD input** logic level being sampled in **In IrDA mode.** **In RS232/RS-485 Mode(UMCR[0] = 1), if this bit is set to 1, the RXD input is inverted before sampled.** **In IrDA mode**,when cleared, the infrared logic block expects an active low or negative IR 3/16 pulse for 0s and 1s are expected for 1s. When INVR is set (INVR 1), the infrared logic block expects an active high or positive IR 3/16 pulse for 0s and 0s are expected for 1s. 0 **RXD input is not inverted** 1 **RXD input is inverted** 0 **RXD** active low detection 1 **RXD** active high detection
8 ENIRI	**Serial Infrared Interrupt Enable**. Enables/Disables the serial infrared interrupt. 0 Serial infrared Interrupt disabled 1 Serial infrared Interrupt enabled

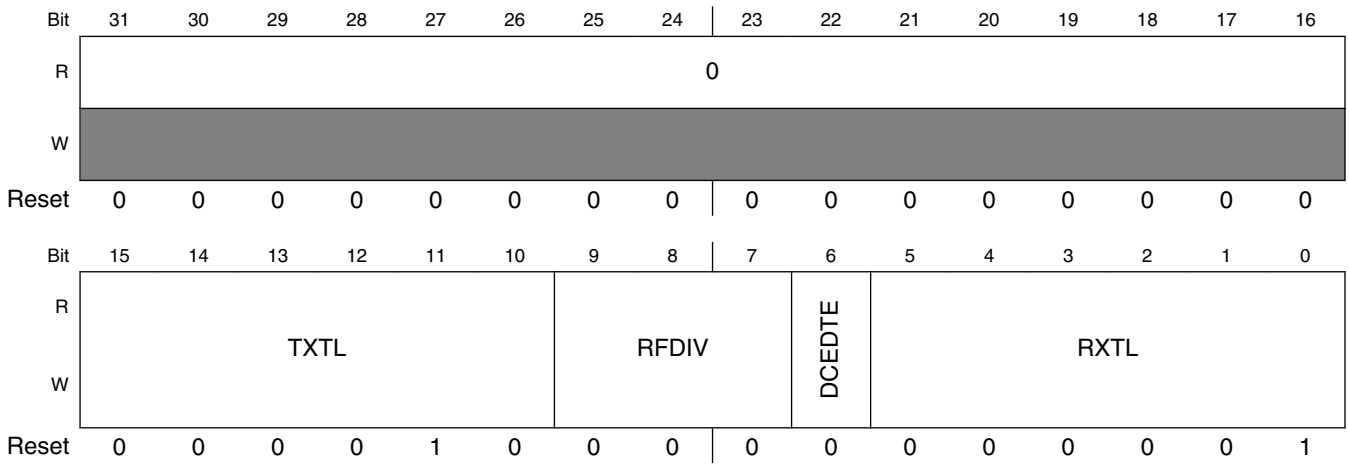
Table continues on the next page...

UARTx_UCR4 field descriptions (continued)

Field	Description
7 WKEN	**WAKE Interrupt Enable**. Enables/Disables the WAKE bit to generate an interrupt. The WAKE bit is set at the detection of a start bit by the receiver. 0 Disable the WAKE interrupt 1 Enable the WAKE interrupt
6 IDDMAEN	**DMA IDLE Condition Detected Interrupt Enable** Enables/Disables the receive DMA request *dma_req_rx* for the IDLE interrupt (triggered with IDLE flag in USR2[12]). 0 DMA IDLE interrupt disabled 1 DMA IDLE interrupt enabled
5 IRSC	**IR Special Case**. Selects the clock for the vote logic. When set, IRSC switches the vote logic clock from the sampling clock to the UART reference clock. The IR pulses are counted a predetermined amount of time depending on the reference frequency. See InfraRed Special Case (IRSC) Bit. 0 The vote logic uses the sampling clock (16x baud rate) for normal operation 1 The vote logic uses the UART reference clock
4 LPBYP	**Low Power Bypass**. Allows to bypass the low power new features in UART. To use during debug phase. 0 Low power features enabled 1 Low power features disabled
3 TCEN	**TransmitComplete Interrupt Enable**. Enables/Disables the TXDC bit to generate an interrupt (*interrupt_uart* = 0) **NOTE:** An interrupt will be issued as long as TCEN and TXDC are high even if the transmitter is not enabled. In general, user should enable the transmitter before enabling the TXDC interrupt. 0 Disable TXDC interrupt 1 Enable TXDC interrupt
2 BKEN	**BREAK Condition Detected Interrupt Enable**. Enables/Disables the BRCD bit to generate an interrupt. 0 Disable the BRCD interrupt 1 Enable the BRCD interrupt
1 OREN	**Receiver Overrun Interrupt Enable**. Enables/Disables the ORE bit to generate an interrupt. 0 Disable ORE interrupt 1 Enable ORE interrupt
0 DREN	**Receive Data Ready Interrupt Enable**. Enables/Disables the RDR bit to generate an interrupt. 0 Disable RDR interrupt 1 Enable RDR interrupt

16.2.15.7 UART FIFO Control Register (UARTx_UFCR)

Address: Base address + 90h offset



UARTx_UFCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–10 TXTL	**Transmitter Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the TxFIFO. A maskable interrupt is generated whenever the data level in the TxFIFO falls below the selected threshold. The bits are encoded as shown in the Settings column. Settings 0 to 32 are in use. All other settings are Reserved. 000000 Reserved 000001 Reserved 000010 TxFIFO has 2 or fewer characters ... — ... — 011111 TxFIFO has 31 or fewer characters 100000 TxFIFO has 32 characters (maximum)
9–7 RFDIV	Reference Frequency Divider. Controls the divide ratio for the reference clock. The input clock is *module_clock*. The output from the divider is *ref_clk* which is used by BRM to create the 16x baud rate oversampling clock (*brm_clk*). 000 Divide input clock by 6 001 Divide input clock by 5 010 Divide input clock by 4 011 Divide input clock by 3 100 Divide input clock by 2 101 Divide input clock by 1 110 Divide input clock by 7 111 Reserved
6 DCEDTE	**DCE/DTE mode select**. Select UART as data communication equipment (DCE mode) or as data terminal equipment (DTE mode).

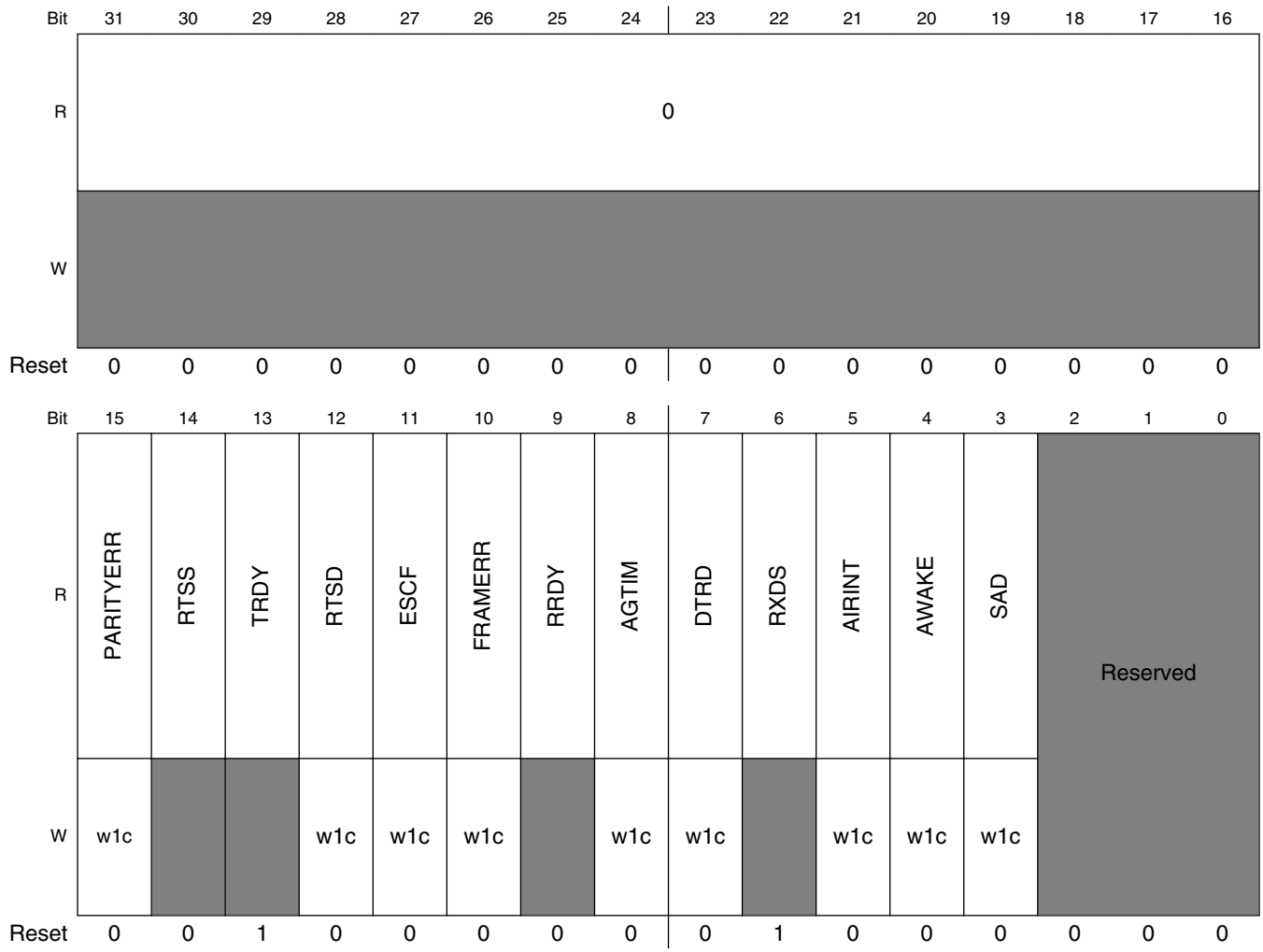
Table continues on the next page...

UARTx_UFCR field descriptions (continued)

Field	Description
	0 DCE mode selected 1 DTE mode selected
RXTL	**Receiver Trigger Level**. Controls the threshold at which a maskable interrupt is generated by the RxFIFO. A maskable interrupt is generated whenever the data level in the RxFIFO reaches the selected threshold. The RXTL bits are encoded as shown in the Settings column. Setting 0 to 32 are in use. All other settings are Reserved. 000000 0 characters received 000001 RxFIFO has 1 character ... — ... — 011111 RxFIFO has 31 characters 100000 RxFIFO has 32 characters (maximum)

16.2.15.8 UART Status Register 1 (UARTx_USR1)

Address: Base address + 94h offset



UARTx_USR1 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 PARITYERR	**Parity Error Interrupt Flag**. Indicates a parity error is detected. PARITYERR is cleared by writing 1 to it. Writing 0 to PARITYERR has no effect. When parity is disabled, PARITYERR always reads 0. At reset, PARITYERR is set to 0. 0 No parity error detected 1 Parity error detected (write 1 to clear)
14 RTSS	RTS_B **Pin Status**. Indicates the current status of the RTS_B pin. A "snapshot" of RTS_B is taken immediately before RTSS is presented to the data bus. RTSS cannot be cleared because all writes to RTSS are ignored. At reset, RTSS is set to 0.

Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
	0 The RTS_B module input is high (inactive) 1 The RTS_B module input is low (active)
13 TRDY	**Transmitter Ready Interrupt / DMA Flag**. Indicates that the TxFIFO emptied below its target threshold and requires data. TRDY is automatically cleared when the data level in the TxFIFO exceeds the threshold set by TXTL bits. At reset, TRDY is set to 1. 0 The transmitter does not require data 1 The transmitter requires data (interrupt posted)
12 RTSD	**RTS Delta.** Indicates whether the RTS_B pin changed state. It (RTSD) generates a maskable interrupt. When in STOP mode, RTS assertion sets RTSD and can be used to wake the processor. The current state of the RTS_B pin is available on the RTSS bit. Clear RTSD by writing 1 to it. Writing 0 to RTSD has no effect. At reset, RTSD is set to 0. 0 RTS_B pin did not change state since last cleared 1 RTS_B pin changed state (write 1 to clear)
11 ESCF	**Escape Sequence Interrupt Flag**. Indicates if an escape sequence was detected. ESCF is asserted when the ESCEN bit is set and an escape sequence is detected in the RxFIFO. Clear ESCF by writing 1 to it. Writing 0 to ESCF has no effect. 0 No escape sequence detected 1 Escape sequence detected (write 1 to clear).
10 FRAMERR	**Frame Error Interrupt Flag**. Indicates that a frame error is detected. The *interrupt_uart* interrupt will be generated if a frame error is detected and the interrupt is enabled. Clear FRAMERR by writing 1 to it. Writing 0 to FRAMERR has no effect. 0 No frame error detected 1 Frame error detected (write 1 to clear)
9 RRDY	**Receiver Ready Interrupt / DMA Flag**. Indicates that the RxFIFO data level is above the threshold set by the RXTL bits. (See the RXTL bits description in UART FIFO Control Register (UART_UFCR) for setting the interrupt threshold.) When asserted, RRDY generates a maskable interrupt or DMA request. RRDY is automatically cleared when data level in the RxFIFO goes below the set threshold level. At reset, RRDY is set to 0. 0 No character ready 1 Character(s) ready (interrupt posted)
8 AGTIM	Ageing Timer Interrupt Flag. Indicates that data in the RxFIFO has been idle for a time of 8 character lengths (where a character length consists of 7 or 8 bits, depending on the setting of the WS bit in UCR2, with the bit time corresponding to the baud rate setting) and FIFO data level is less than RxFIFO threshold level (RXTL in the UFCR). Clear by writing a 1 to it. 0 AGTIM is not active 1 AGTIM is active (write 1 to clear)
7 DTRD	This bit is not used in this chip.
6 RXDS	Receiver IDLE Interrupt Flag. Indicates that the receiver state machine is in an IDLE state, the next state is IDLE, and the receive pin is high. RXDS is automatically cleared when a character is received. RXDS is active only when the receiver is enabled. 0 Receive in progress 1 Receiver is IDLE

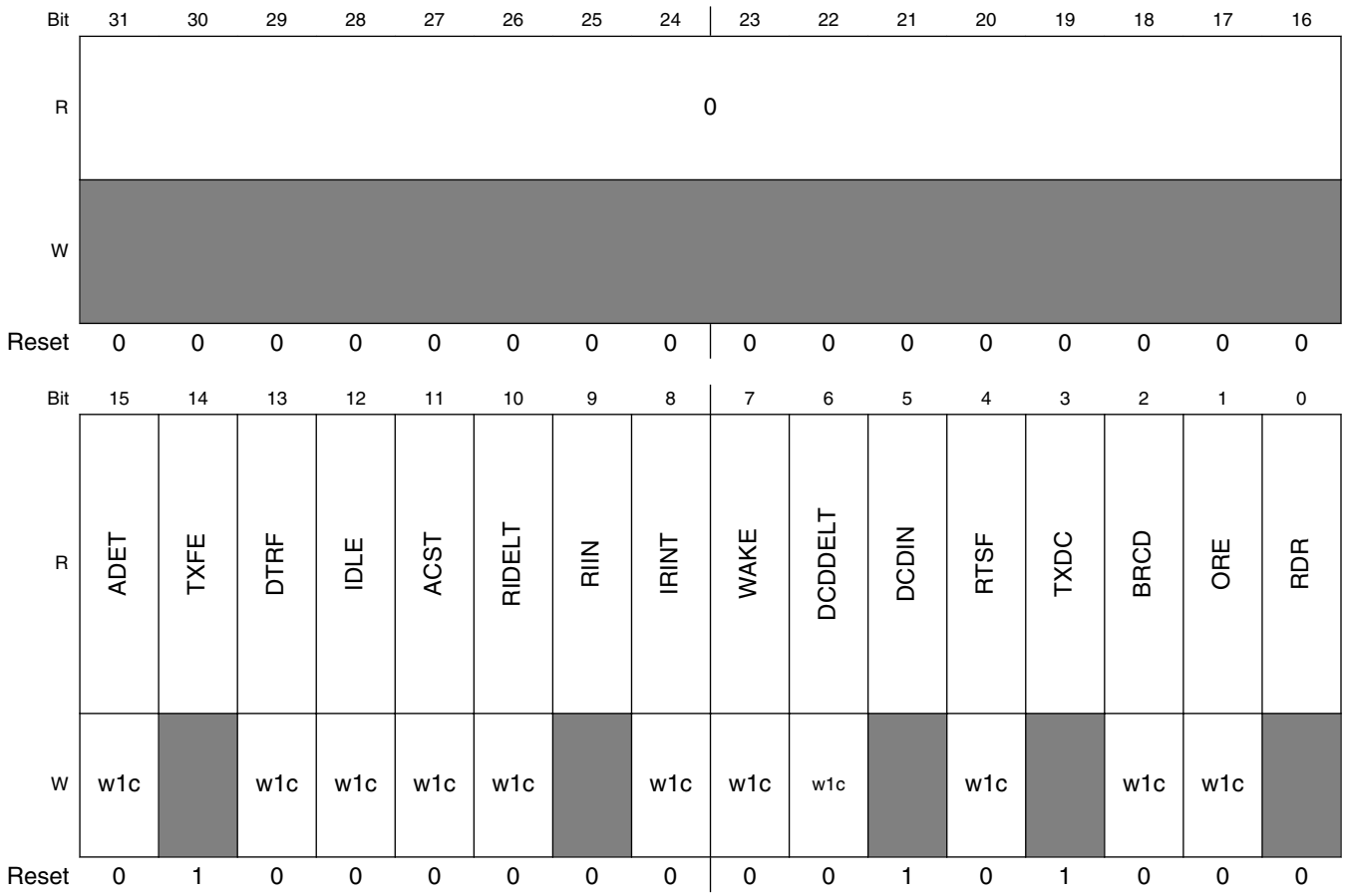
Table continues on the next page...

UARTx_USR1 field descriptions (continued)

Field	Description
5 AIRINT	Asynchronous IR WAKE Interrupt Flag. Indicates that the IR WAKE pulse was detected on the RXD pin. Clear AIRINT by writing 1 to it. Writing 0 to AIRINT has no effect. 0 No pulse was detected on the RXD IrDA pin 1 A pulse was detected on the RXD IrDA pin
4 AWAKE	Asynchronous WAKE Interrupt Flag. Indicates that a falling edge was detected on the RXD pin. Clear AWAKE by writing 1 to it. Writing 0 to AWAKE has no effect. 0 No falling edge was detected on the RXD Serial pin 1 A falling edge was detected on the RXD Serial pin
3 SAD	RS-485 Slave Address Detected Interrupt Flag. Indicates if RS-485 Slave Address was detected . SAD was asserted in RS-485 mode when the SADEN bit is set and Slave Address is detected in RxFIFO (in Nomal Address Detect Mode, the 9th data bit = 1; in Automatic Address Detect Mode, the received charater matches the programmed SLADDR). 0 No slave address detected 1 Slave address detected
-	This field is reserved. Reserved

16.2.15.9 UART Status Register 2 (UARTx_USR2)

Address: Base address + 98h offset



UARTx_USR2 field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15 ADET	**Automatic Baud Rate Detect Complete**. Indicates that an "A" or "a" was received and that the receiver detected and verified the incoming baud rate. Clear ADET by writing 1 to it. Writing 0 to ADET has no effect. 0 ASCII "A" or "a" was not received 1 ASCII "A" or "a" was received (write 1 to clear)
14 TXFE	**Transmit Buffer FIFO Empty**. Indicates that the transmit buffer (TxFIFO) is empty. TXFE is cleared automatically when data is written to the TxFIFO. Even though TXFE is high, the transmission might still be in progress. 0 The transmit buffer (TxFIFO) is not empty 1 The transmit buffer (TxFIFO) is empty
13 DTRF	This bit is not used in this chip.

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
12 IDLE	**Idle Condition**. Indicates that an idle condition has existed for more than a programmed amount frame (see Idle Line Detect. An interrupt can be generated by this IDLE bit if IDEN (UCR1[12]) is enabled. IDLE is cleared by writing 1 to it. Writing 0 to IDLE has no effect. 0 No idle condition detected 1 Idle condition detected (write 1 to clear)
11 ACST	**Autobaud Counter Stopped**. In autobaud detection (ADBR=1), indicates the counter which determines the baud rate was running and is now stopped. This means either START bit is finished (if ADNIMP=1), or Bit 0 is finished (if ADNIMP=0). See New Autobaud Counter Stopped bit and Interrupt, for more details. An interrupt can be flagged on *interrupt_uart* if ACIEN=1. 0 Measurement of bit length not finished (in autobaud) 1 Measurement of bit length finished (in autobaud). (write 1 to clear)
10 RIDELT	This bit is not used in this chip.
9 RIIN	This bit is not used in this chip.
8 IRINT	**Serial Infrared Interrupt Flag.** When an edge is detected on the RXD pin during SIR Mode, this flag will be asserted. This flag can cause an interrupt which can be masked using the control bit ENIRI: UCR4 [8]. 0 no edge detected 1 valid edge detected (write 1 to clear)
7 WAKE	**Wake**. Indicates the start bit is detected. WAKE can generate an interrupt that can be masked using the WKEN bit. Clear WAKE by writing 1 to it. Writing 0 to WAKE has no effect. 0 start bit not detected 1 start bit detected (write 1 to clear)
6 DCDDELT	This bit is not used in this chip.
5 DCDIN	This bit is not used in this chip.
4 RTSF	**RTS Edge Triggered Interrupt Flag. Indicates if** a programmed edge is detected on the RTS_B pin. The RTEC bits select the edge that generates an interrupt (see Table 16-6). RTSF can generate an interrupt that can be masked using the RTSEN bit. Clear RTSF by writing 1 to it. Writing 0 to RTSF has no effect. 0 Programmed edge not detected on RTS_B 1 Programmed edge detected on RTS_B (write 1 to clear)
3 TXDC	**Transmitter Complete**. Indicates that the transmit buffer (TxFIFO) and Shift Register is empty; therefore the transmission is complete. TXDC is cleared automatically when data is written to the TxFIFO. 0 Transmit is incomplete 1 Transmit is complete
2 BRCD	**BREAK Condition Detected**. Indicates that a BREAK condition was detected by the receiver. Clear BRCD by writing 1 to it. Writing 0 to BRCD has no effect. 0 No BREAK condition was detected 1 A BREAK condition was detected (write 1 to clear)
1 ORE	**Overrun Error**. When set to 1, ORE indicates that the receive buffer (RxFIFO) was full (32 chars inside), and a 33rd character has been fully received. This 33rd character has been discarded. Clear ORE by writing 1 to it. Writing 0 to ORE has no effect.

Table continues on the next page...

UARTx_USR2 field descriptions (continued)

Field	Description
	0 No overrun error 1 Overrun error (write 1 to clear)
0 RDR	**Receive Data Ready**-Indicates that at least 1 character is received and written to the RxFIFO. If the URXD register is read and there is only 1 character in the RxFIFO, RDR is automatically cleared. 0 No receive data ready 1 Receive data ready

16.2.15.10 UART Escape Character Register (UARTx_UESC)

Address: Base address + 9Ch offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																ESC_CHAR															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	1

UARTx_UESC field descriptions

Field	Description
31–8 Reserved	This read-only field is reserved and always has the value 0.
ESC_CHAR	**UART Escape Character**. Holds the selected escape character that all received characters are compared against to detect an escape sequence.

16.2.15.11 UART Escape Timer Register (UARTx_UTIM)

Address: Base address + A0h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																				TIM											
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UTIM field descriptions

Field	Description
31–12 Reserved	This read-only field is reserved and always has the value 0.
TIM	UART Escape Timer. Holds the maximum time interval (in ms) allowed between escape characters. The escape timer register is programmable in intervals of 2 ms. See Escape Sequence Detection and Table 16-10 for more information on the UART escape sequence detection. Reset value 0x000 = 2 ms up to 0xFFF = 8.192 s.

16.2.15.12 UART BRM Incremental Register (UARTx_UBIR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write 0x000F value into the UBIR after finishing detecting baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[3].

Please note software reset will reset the register to its reset value.

Address: Base address + A4h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																INC															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_UBIR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
INC	Incremental Numerator. Holds the numerator value minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this field using byte accesses is not recommended and is undefined.

16.2.15.13 UART BRM Modulator Register (UARTx_UBMR)

This register can be written by both software and hardware. When enabling the automatic baud rate detection feature hardware can write a proper value into the UBMR based on detected baud rate. Hardware has higher priority when both software and hardware try to write it at the same cycle[4].

Please note software reset will reset the register to its reset value.

Address: Base address + A8h offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																MOD															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

3. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.
4. Note: The write priority in the new design is not same as the original UART. In the orginal design, software has higher priority than hardware when writing this register at the same time.

UARTx_UBMR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
MOD	**Modulator Denominator.** Holds the value of the denominator minus one of the BRM ratio (see Binary Rate Multiplier (BRM)). The UBIR register MUST be updated before the UBMR register for the baud rate to be updated correctly. If only one register is written to by software, the BRM will ignore this data until the other register is written to by software. Updating this register using byte accesses is not recommended and undefined.

16.2.15.14 UART Baud Rate Count Register (UARTx_UBRC)

Address: Base address + ACh offset

Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0																BCNT															
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0

UARTx_UBRC field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
BCNT	**Baud Rate Count Register.** This read only register is used to count the start bit of the incoming baud rate (if ADNIMP=1), or start bit + bit0 (if ADNIMP=0). When the measurement is done, the Baud Rate Count Register contains the number of UART internal clock cycles (clock after divider) present in an incoming bit. BCNT retains its value until the next Automatic Baud Rate Detection sequence has been initiated. The 16 bit Baud Rate Count register is reset to 4 and stays at hex FFFF in the case of an overflow.

16.2.15.15 UART One Millisecond Register (UARTx_ONEMS)

NOTE

This register has been expanded from 16 bits to 24 bits. In previous versions, the 16-bit ONEMS can only support the maximum 65.535MHz (0xFFFFx1000) *ref_clk*. To support 4Mbps Bluetooth application with 66.5MHz *module_clock*, the value 0x103C4 (66.5M/1000) should be written into this register. In this case, the 16 bits are not enough to contain the 0x103C4. So this register was expanded to 24 bits to support high frequency of the *ref_clk*.

Address: Base address + B0h offset

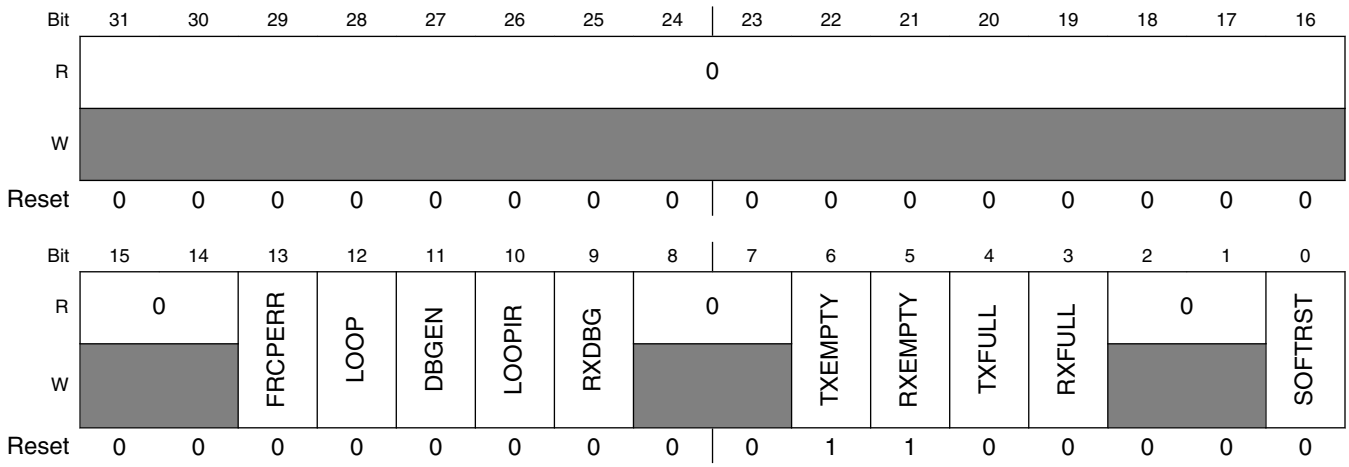
Bit	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
R	0								ONEMS																							
W																																
Reset	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

UARTx_ONEMS field descriptions

Field	Description
31–24 Reserved	This read-only field is reserved and always has the value 0.
ONEMS	**One Millisecond Register.** This 24-bit register must contain the value of the UART internal frequency (*ref_clk* in Figure 16-9) divided by 1000. The internal frequency is obtained after the UART BRM internal divider (F (*ref_clk*) = F(*module_clock*) / RFDIV). In fact this register contains the value corresponding to the number of UART BRM internal clock cycles present in one millisecond. The ONEMS (and UTIM) registers value are used in the escape character detection feature (Escape Sequence Detection) to count the number of clock cycles left between two escape characters. The ONEMS register is also used in infrared special case mode (IRSC = UCR4[5] = 1'b1), see InfraRed Special Case (IRSC) Bit.

16.2.15.16 UART Test Register (UARTx_UTS)

Address: Base address + B4h offset



UARTx_UTS field descriptions

Field	Description
31–14 Reserved	This read-only field is reserved and always has the value 0.
13 FRCPERR	Force Parity Error. Forces the transmitter to generate a parity error if parity is enabled. FRCPERR is provided for system debugging. 0 Generate normal parity 1 Generate inverted parity (error)
12 LOOP	Loop TX and RX for Test. Controls loopback for test purposes. When LOOP is high, the receiver input is internally connected to the transmitter and ignores the RXD pin. The transmitter is unaffected by LOOP. If RXDMUXSEL (UCR3[2]) is set to 1, the loopback is applied on serial and IrDA signals. If RXDMUXSEL is set to 0, the loopback is only applied on serial signals. 0 Normal receiver operation 1 Internally connect the transmitter output to the receiver input
11 DBGEN	This bit is not used in this chip. debug_enable_B**.** This bit controls whether to respond to the *debug_req* input signal. 0 UART will go into debug mode when *debug_req* is HIGH 1 UART will not go into debug mode even if *debug_req* is HIGH
10 LOOPIR	**Loop TX and RX for IR Test (LOOPIR)**. This bit controls loopback from transmitter to receiver in the InfraRed interface. 0 No IR loop 1 Connect IR transmitter to IR receiver
9 RXDBG	This bit is not used in this chip. **RX_fifo_debug_mode.** This bit controls the operation of the RX fifo read counter when in debug mode. 0 rx fifo read pointer does not increment 1 rx_fifo read pointer increments as normal

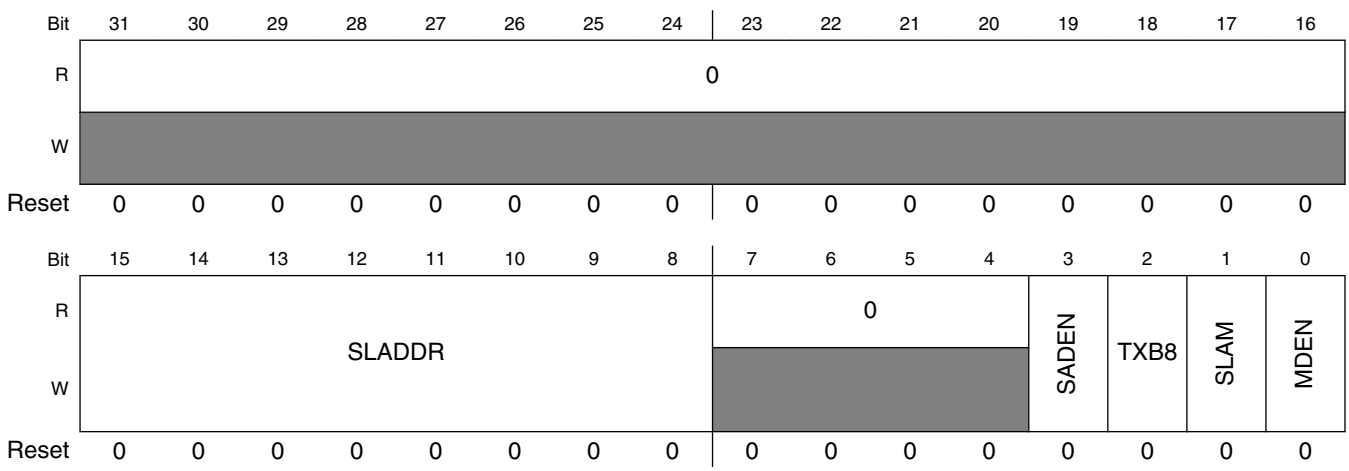
Table continues on the next page...

UARTx_UTS field descriptions (continued)

Field	Description
8–7 Reserved	This read-only field is reserved and always has the value 0.
6 TXEMPTY	TxFIFO Empty. Indicates that the TxFIFO is empty. 0 The TxFIFO is not empty 1 The TxFIFO is empty
5 RXEMPTY	RxFIFO Empty. Indicates the RxFIFO is empty. 0 The RxFIFO is not empty 1 The RxFIFO is empty
4 TXFULL	TxFIFO FULL. Indicates the TxFIFO is full. 0 The TxFIFO is not full 1 The TxFIFO is full
3 RXFULL	RxFIFO FULL. Indicates the RxFIFO is full. 0 The RxFIFO is not full 1 The RxFIFO is full
2–1 Reserved	This read-only field is reserved and always has the value 0.
0 SOFTRST	Software Reset. Indicates the status of the software reset (SRST_B bit of UCR2). 0 Software reset inactive 1 Software reset active

16.2.15.17 UART RS-485 Mode Control Register (UARTx_UMCR)

Address: Base address + B8h offset



UARTx_UMCR field descriptions

Field	Description
31–16 Reserved	This read-only field is reserved and always has the value 0.
15–8 SLADDR	RS-485 Slave Address Character. Holds the selected slave adress character that the receiver wil try to detect.
7–4 Reserved	This read-only field is reserved and always has the value 0.
3 SADEN	RS-485 Slave Address Detected Interrupt Enable. 0 Disable RS-485 Slave Address Detected Interrupt 1 Enable RS-485 Slave Address Detected Interrupt
2 TXB8	Transmit RS-485 bit 8 (the ninth bit or 9th bit). In RS-485 mode, software writes TXB8 bit as the 9th data bit to be transmitted. 0 0 will be transmitted as the RS485 9th data bit 1 1 will be transmitted as the RS485 9th data bit
1 SLAM	RS-485 Slave Address Detect Mode Selection. 0 Select Normal Address Detect mode 1 Select Automatic Address Detect mode
0 MDEN	9-bit data or Multidrop Mode (RS-485) Enable. 0 Normal RS-232 or IrDA mode, see Table 16-3 for detail. 1 Enable RS-485 mode, see Table 16-3 for detail

Appendix A
Revision History

A.1 Substantive changes from revision 1 to revision 2

Substantive changes from revision 1 to revision 2 are as follows:

A.1.1 Reference Manual Revision History

Topic Cross-Reference	Change Description
-	Updated SJC chapter

A.1.2 Introduction Revision History

No substantive changes

A.1.3 Memory Maps Revision History

No substantive changes

A.1.4 System Security Revision History

No substantive changes

A.1.5 RDC Revision History

No substantive changes

A.1.6 A53 Revision History

No substantive changes

A.1.7 CM4 Revision History

No substantive changes

A.1.8 MU Revision History

No substantive changes

A.1.9 SEMA4 Revision History

No substantive changes

A.1.10 OCRAM Revision History

No substantive changes

A.1.11 NIC Revision History

No substantive changes

A.1.12 AIPSTZ Revision History

No substantive changes

A.1.13 SPBA Revision History

No substantive changes

A.1.14 ROMCP Revision History

No substantive changes

A.1.15 ROMCP Revision History

No substantive changes

A.1.16 TZASC Revision History

No substantive changes

A.1.17 System Debug Revision History

No substantive changes

A.1.18 SJC Revision History

Topic Cross-Reference	Change Description
CLAMP Instruction	Added DFT_TAP_IDCODE description table
SJC Instruction Register (SJIR)	Updated SJC Instruction Register (SJIR) table
Features	Added DFT_TAP JTAG Instruction Register (DJIR)
Modes of Operation	Added CLAMP instruction topic
DFT_TAP JTAG Instruction Register (DJIR)	
Overview	
ID_CODE Instruction (SJC IDCODE / DFT_TAP IDCODE)	

A.1.19 CCM Revision History

No substantive changes

A.1.20 GPC Revision History

No substantive changes

A.1.21 XTALOSC Revision History

No substantive changes

A.1.22 TMU Revision History

No substantive changes

A.1.23 System Boot Revision History

Topic Cross-Reference	Change Description
USB Low Power Boot	Renamed BT_MMU_DISABLE to BT_DCACHE_DISABLE
Clocks at boot time	Updated ARM frequency for various low power boot modes
Boot eFUSE descriptions	Updated ARM_PLL and ARM_A53_ROOT frequencies

A.1.24 Fusemap Revision History

No substantive changes

A.1.25 OCOTP Revision History

Topic Cross-Reference	Change Description
OCOTP Memory Map/Register Definition	Corrected fuse bank/word for HW_OCOTP_FIELD_RETURN

A.1.26 SNVS Revision History

No substantive changes

A.1.27 SRC Revision History

No substantive changes

A.1.28 Interrupts Revision History

No substantive changes

A.1.29 SDMA Revision History

No substantive changes

A.1.30 External Signals Revision History

No substantive changes

A.1.31 IOMUXC Revision History

No substantive changes

A.1.32 GPIO Revision History

No substantive changes

A.1.33 External Memory Overview Revision History

No substantive changes

A.1.34 DDR Performance Monitor Revision History

No substantive changes

A.1.35 DDR Controller Revision History

No substantive changes

A.1.36 DDR PHY Revision History

No substantive changes

A.1.37 APBH Revision History

No substantive changes

A.1.38 BCH Revision History

No substantive changes

A.1.39 GPMI Revision History

No substantive changes

A.1.40 ECSPI Revision History

No substantive changes

A.1.41 QSPI Revision History

No substantive changes

A.1.42 USDHC Revision History

No substantive changes

A.1.43 USB 3.0 Revision History

No substantive changes

A.1.44 Peripheral Component Interconnect express (PCIe) Revision History

No substantive changes

A.1.45 PCIe PHY Revision History

No substantive changes

A.1.46 ENET Revision History

No substantive changes

A.1.47 GPT Revision History

No substantive changes

A.1.48 PWM Revision History

No substantive changes

A.1.49 Multimedia Overview Revision History

No substantive changes

A.1.50 eLCDIF Revision History

No substantive changes

A.1.51 Graphics Processing Unit (GPU) Revision History

No substantive changes

A.1.52 HDMI TX Revision History

No substantive changes

A.1.53 HDMI TX PHY Revision History

No substantive changes

A.1.54 MIPI DSI Revision History

No substantive changes

A.1.55 CSI Revision History

No substantive changes

A.1.56 MIPI CSI2 Revision History

No substantive changes

A.1.57 SPDIF Revision History

No substantive changes

A.1.58 SAI Revision History

No substantive changes

A.1.59 VPU G1 Revision History

No substantive changes

A.1.60 VPU G2 Revision History

No substantive changes

A.1.61 Display Controller Subsystem (DCSS) Revision History

No substantive changes

A.1.62 DCSS Block Control (BLK_CTL) Revision History

No substantive changes

A.1.63 Display Timing Generator (DTG) Revision History

No substantive changes

A.1.64 Context Load (CTX_LD) Revision History

No substantive changes

A.1.65 Graphics Decompression (DEC400D) Revision History

No substantive changes

A.1.66 Decompression and Tile to Raster Conversion (DTRC) Revision History

No substantive changes

A.1.67 Display, Prefetch and Resolve (DPR) Revision History

No substantive changes

A.1.68 Scaler (MED_DC_SCALER) Revision History

No substantive changes

A.1.69 Look Up Table Load (LUT_LD) Revision History

No substantive changes

A.1.70 HDR10 Image Processing (MED_HDR10) Revision History

No substantive changes

A.1.71 Color Sub-Sampler (SUBSAM) Revision History

No substantive changes

A.1.72 Write Scale (WR_SCL) Revision History

No substantive changes

A.1.73 Read Surface (RD_SRC) Revision History

No substantive changes

A.1.74 Interrupt Request Steering (IrqSteer) Revision History

No substantive changes

A.1.75 I2C Revision History

No substantive changes

A.1.76 UART Revision History

No substantive changes

How to Reach Us:

Home Page:
nxp.com

Web Support:
nxp.com/support

Information in this document is provided solely to enable system and software implementers to use NXP products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document. NXP reserves the right to make changes without further notice to any products herein.

NXP makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does NXP assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in NXP data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. NXP does not convey any license under its patent rights nor the rights of others. NXP sells products pursuant to standard terms and conditions of sale, which can be found at the following address: nxp.com/SalesTermsandConditions.

While NXP has implemented advanced security features, all products may be subject to unidentified vulnerabilities. Customers are responsible for the design and operation of their applications and products to reduce the effect of these vulnerabilities on customer's applications and products, and NXP accepts no liability for any vulnerability that is discovered. Customers should implement appropriate design and operating safeguards to minimize the risks associated with their applications and products.

Document Number IMX8MDQLQRM
Revision 2, 08/2019