



Arm® Neoverse™ V1 reference design

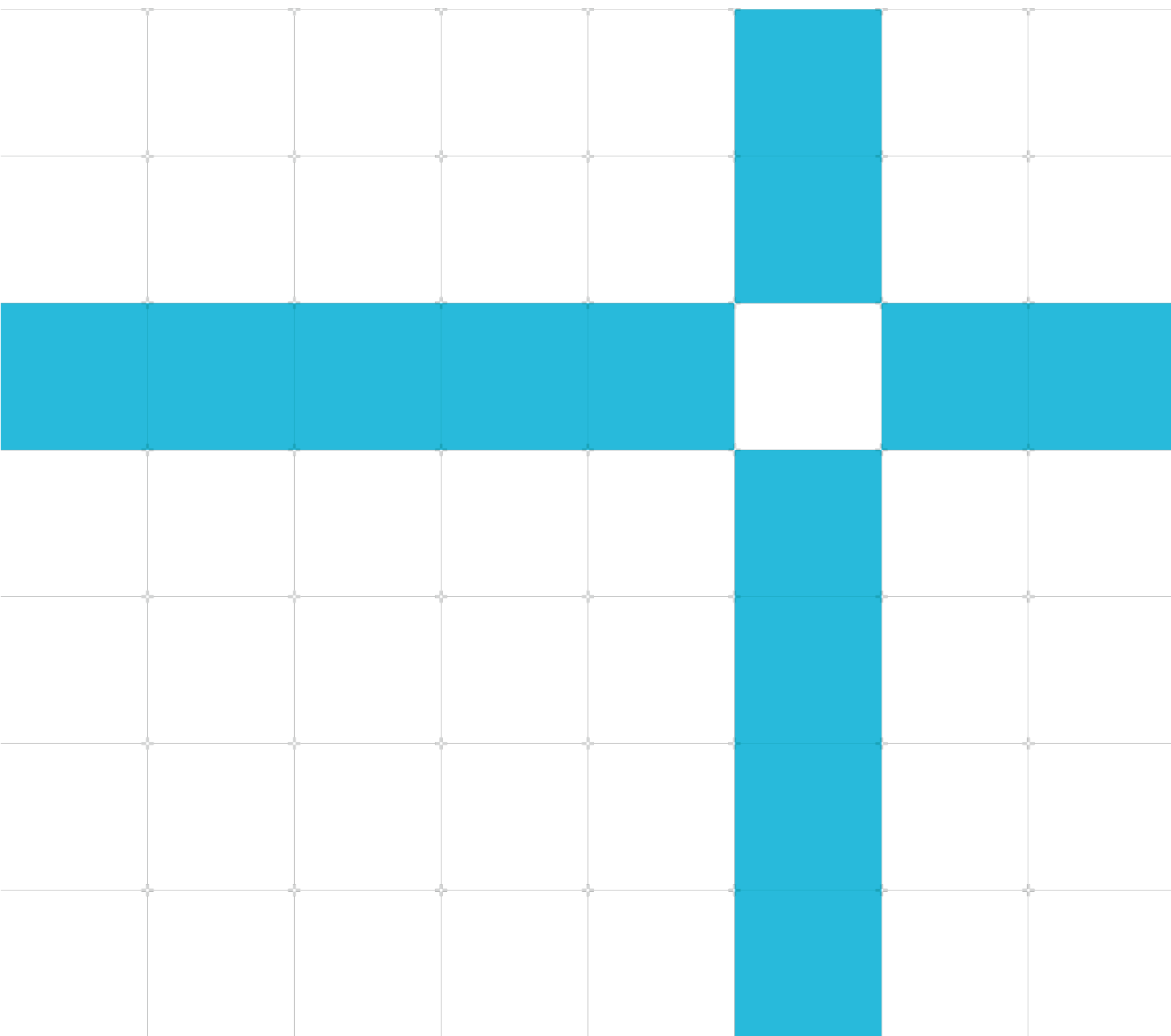
Software Developer Guide

Non-Confidential

Copyright © 2021 Arm Limited (or its affiliates).
All rights reserved.

Issue 01

PJDOC-1779577084-33214



Arm® Neoverse™ V1 reference design Software Developer Guide

Copyright © 2021 Arm Limited (or its affiliates). All rights reserved.

Release information

Document history

Issue	Date	Confidentiality	Change
01	April 2021	Non-Confidential	Initial release

Non-Confidential Proprietary Notice

This document is protected by copyright and other related rights and the practice or implementation of the information contained in this document may be protected by one or more patents or pending patent applications. No part of this document may be reproduced in any form by any means without the express prior written permission of Arm. No license, express or implied, by estoppel or otherwise to any intellectual property rights is granted by this document unless specifically stated.

Your access to the information in this document is conditional upon your acceptance that you will not use or permit others to use the information for the purposes of determining whether implementations infringe any third party patents.

THIS DOCUMENT IS PROVIDED "AS IS". ARM PROVIDES NO REPRESENTATIONS AND NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, INCLUDING, WITHOUT LIMITATION, THE IMPLIED WARRANTIES OF MERCHANTABILITY, SATISFACTORY QUALITY, NON-INFRINGEMENT OR FITNESS FOR A PARTICULAR PURPOSE WITH RESPECT TO THE DOCUMENT. For the avoidance of doubt, Arm makes no representation with respect to, and has undertaken no analysis to identify or understand the scope and content of, patents, copyrights, trade secrets, or other rights.

This document may include technical inaccuracies or typographical errors.

TO THE EXTENT NOT PROHIBITED BY LAW, IN NO EVENT WILL ARM BE LIABLE FOR ANY DAMAGES, INCLUDING WITHOUT LIMITATION ANY DIRECT, INDIRECT, SPECIAL, INCIDENTAL, PUNITIVE, OR CONSEQUENTIAL DAMAGES, HOWEVER CAUSED AND REGARDLESS OF THE THEORY OF LIABILITY, ARISING OUT OF ANY USE OF THIS DOCUMENT, EVEN IF ARM HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

This document consists solely of commercial items. You shall be responsible for ensuring that any use, duplication or disclosure of this document complies fully with any relevant export laws and regulations to assure that this document or any portion thereof is not exported, directly or indirectly, in violation of such export laws. Use of the word "partner" in reference to Arm's customers is not intended to create or refer to any partnership relationship with any other company. Arm may make changes to this document at any time and without notice.

If any of the provisions contained in these terms conflict with any of the provisions of any click through or signed written agreement covering this document with Arm, then the click through or signed written agreement prevails over and supersedes the conflicting provisions of these terms. This document may be translated into other languages for convenience, and you agree that if there is any conflict between the English version of this document and any translation, the terms of the English version of the Agreement shall prevail.

The Arm corporate logo and words marked with ® or ™ are registered trademarks or trademarks of Arm Limited (or its affiliates) in the US and/or elsewhere. All rights reserved. Other brands and names mentioned in this document may be the trademarks of their respective owners. Please follow Arm's trademark usage guidelines at <https://www.arm.com/company/policies/trademarks>.

Copyright © 2021 Arm Limited (or its affiliates). All rights reserved.

Arm Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

(LES-PRE-20349)

Confidentiality Status

This document is Non-Confidential. The right to use, copy and disclose this document may be subject to license restrictions in accordance with the terms of the agreement entered into by Arm and the party that Arm delivered this document to.

Unrestricted Access is an Arm internal classification.

Product Status

The information in this document is final, that is for a developed product.

Web Address

developer.arm.com

Progressive terminology commitment

Arm values inclusive communities. Arm recognizes that we and our industry have used terms that can be offensive. Arm strives to lead the industry and create change.

This document includes terms that can be offensive. We will replace these terms in a future issue of this document. If you find offensive terms in this document, please email terms@arm.com.

Contents

1 Introduction	8
1.1 Product revision status	8
1.2 Intended audience.....	8
1.3 Conventions.....	8
1.3.1 Glossary	8
1.3.2 Terms and abbreviations	8
1.3.3 Typographical conventions.....	10
1.4 Feedback.....	11
1.4.1 Feedback on this product	11
1.4.2 Feedback on content	11
2 Overview of Arm® Neoverse™ V1 Fixed Virtual Platform.....	12
2.1 About Arm® Neoverse™ V1 reference design.....	12
2.1.1 Deliverables	12
2.1.2 Features.....	12
2.1.3 Configurations	13
2.2 Compliance	13
3 Hardware	14
3.1 System architecture	14
3.2 Processor element	16
3.3 Interconnect element	18
3.3.1 Coherent mesh network	19
3.3.2 Generic interrupt controller.....	23
3.3.3 System memory management unit	24
3.3.4 Interconnect extension interfaces.....	25
3.4 MSCP element	26
3.4.1 SCP	26
3.4.2 MCP.....	28
3.4.3 Message Handling Unit.....	30
3.5 Memory element.....	30
3.6 Base element	31

3.7 System Boot	32
4 Functional description	33
4.1 Power management	33
4.1.1 Voltage and power domains	34
4.1.2 Domain hierarchy	34
4.1.3 Element-specific power management	35
4.1.4 Arm® Neoverse™ V1 core power and performance management	38
4.2 Clocks	39
4.2.1 Input clocks	39
4.2.2 Output clocks	40
4.2.3 System clocks	41
4.3 Counters	43
4.3.1 REFCLK counter	43
4.3.2 CoreSight timestamp counter	44
4.4 Timers	44
4.4.1 Time domains	44
4.4.2 Generic Timers	45
4.4.3 Watchdog timers	46
4.5 Power down considerations	47
4.5.1 AP Wakeup	47
4.5.2 SCP Wakeup	47
5 Fixed Virtual Platform	49
5.1 About the FVP	49
5.2 FVP peripherals	49
6 Software	53
6.1 About the software	53
6.2 SCP firmware	54
6.2.1 Power control	54
6.2.2 SCP boot ROM	54
6.3 MCP firmware	54
6.3.1 MCP boot ROM	55
6.4 AP firmware	55
6.4.1 Arm Trusted firmware BL1	55

6.4.2 Arm Trusted firmware BL2	55
6.4.3 Arm Trusted firmware BL31	55
6.5 Linux kernel.....	56
6.5.1 Multi-Processing (MP)	56
6.5.2 UEFI awareness	57
6.5.3 Device drivers	57
6.6 Multi-chiplet support	57
7 Memory map.....	58
7.1 AP memory map	58
7.1.1 Boot region	62
7.1.2 Peripherals region	63
7.1.3 CoreSight debug and trace.....	68
7.1.4 Expansion Space	70
7.1.5 DRAM.....	70
7.2 AP interrupt map	71
7.2.1 PPI interrupts	71
7.2.2 SPI interrupts.....	71
7.3 SCP memory map	76
7.3.1 ROM and RAM sizes	78
7.3.2 Peripherals region	78
7.4 SCP interrupt map.....	80
7.5 MCP memory map.....	84
7.5.1 ROM and RAM sizes	85
7.5.2 Peripherals region	86
7.6 MCP Interrupt map.....	88
7.7 Register set	90
7.7.1 System ID registers unit	90
7.7.2 REFCLK counter	94
7.7.3 Generic timers.....	95
7.7.4 CPU manager and power control registers	103
7.7.5 System PIK.....	117
7.7.6 Debug PIK	127
7.7.7 Debug chain power control logic	134
7.7.8 MSCP PIK.....	139

Appendix A Arm IP used in the Reference Design 157

Appendix B Revisions..... 158

1 Introduction

1.1 Product revision status

The rxy identifier indicates the revision status of the product described in this book, for example, r1p2, where:

rx

Identifies the major revision of the product, for example, r1.

py

Identifies the minor revision or modification status of the product, for example, p2.

1.2 Intended audience

This Software Developer Guide (SDG) is intended to assist software developers working with the related Fixed Virtual Platform (FVP) model and Open Source Software stack.

1.3 Conventions

The following subsections describe conventions used in Arm documents.

1.3.1 Glossary

The Arm Glossary is a list of terms used in Arm documentation, together with definitions for those terms. The Arm Glossary does not contain terms that are industry standard unless the Arm meaning differs from the generally accepted meaning.

See the Arm Glossary for more information: <https://developer.arm.com/glossary>.







1.3.2 Terms and abbreviations

This document uses the following terms and abbreviations.

Term	Meaning
ATB	AMBA® Trace Bus
AXI	AMBA eXtensible interface
BMC	Baseboard Management Controller
CCIX	Cache coherent Interconnect for Accelerators
CHI	Coherent Hub Interface
CHI-D	Coherent Hub Interface Issue D

Term	Meaning
CMN	Coherent Mesh Network
CPU	Central Processing Unit
CSS	Compute Subsystem
CTI	Cross Trigger Interface
DDR	Double Data Rate DRAM
Device memory	Memory type. Access rules for device memory are more restrictive than access rules for Normal memory, see the <i>Arm® Architecture Reference Manual ARMv7</i>
DFI	DDR PHY Interface
DMA	Direct Memory Access
DMC	Dynamic Memory Controller
DP	Debug Port
DRAM	Dynamic Random-Access Memory
DVFS	Dynamic Voltage and Frequency Scaling
ETF	Embedded Trace Funnel
ETR	Embedded Trace Router
GALS	Globally Asynchronous, Locally Synchronous
GIC	Generic Interrupt Controller
iBEP	implementation Base Enablement Package
MMU	Memory Management Unit
NIC	Network Interconnect
Normal memory	Memory type, used for program code and data storage, see the <i>Arm® Architecture Reference Manual ARMv7</i>
NVIC	Nested Vector Interrupt Controller
PDD	Platform Design Document
PHY	Physical layer
RAM	Random Access Memory
RAS	Reliability, Availability, and Serviceability
ROM	Read-only Memory
SCP	System Control Processor
SMMU	System Memory Management Unit
STM	System Trace Macrocell
SWD	Serial Wire Debug
TLB	Translation Lookaside Buffer

1.3.3 Typographical conventions

Convention	Use
<i>italic</i>	Introduces citations.
bold	Highlights interface elements, such as menu names. Denotes signal names. Also used for terms in descriptive lists, where appropriate.
monospace	Denotes text that you can enter at the keyboard, such as commands, file and program names, and source code.
monospace bold	Denotes language keywords when used outside example code.
monospace <u>underline</u>	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: <code>MRC p15, 0, <Rd>, <CRn>, <CRm>, <Opcode_2></code>
SMALL CAPITALS	Used in body text for a few terms that have specific technical meanings, that are defined in the Arm® Glossary. For example, IMPLEMENTATION DEFINED, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.
 Caution	This represents a recommendation which, if not followed, might lead to system failure or damage.
 Warning	This represents a requirement for the system that, if not followed, might result in system failure or damage.
 Danger	This represents a requirement for the system that, if not followed, will result in system failure or damage.
 Note	This represents an important piece of information that needs your attention.
 Tip	This represents a useful tip that might make it easier, better or faster to perform a task.
 Remember	This is a reminder of something important that relates to the information you are reading.

1.4 Feedback

Arm welcomes feedback on this product and its documentation.

1.4.1 Feedback on this product

If you have any comments or suggestions about this product, contact your supplier and give:

- The product name.
- The product revision or version.
- An explanation with as much information as you can provide. Include symptoms and diagnostic procedures if appropriate.

1.4.2 Feedback on content

If you have comments on content, send an email to errata@arm.com and give:

- The title Arm® Neoverse™ V1 reference design Software Developer Guide.
- The number PJDOC-1779577084-33214.
- If applicable, the page number(s) to which your comments refer.
- A concise explanation of your comments.

Arm also welcomes general suggestions for additions and improvements.



Arm tests the PDF only in Adobe Acrobat and Acrobat Reader and cannot guarantee the quality of the represented document when used with any other PDF reader.

2 Overview of Arm® Neoverse™ V1 Fixed Virtual Platform

Arm® Neoverse™ V1 reference design (RD-V1) describes and models the design choices for recommended configurations of typical Arm-based subsystems. The RD-V1 Fixed Virtual Platform (FVP) is a functional model intended for software development on the RD-V1 subsystem configurations.

2.1 About Arm® Neoverse™ V1 reference design

RD-V1 incorporates Arm® Neoverse™ V1 cores, CMN-650 interconnect, and other system IP, into designs with a high core count, targeted at the following markets:

- Edge - designs optimized for Network Edge and I/O Accelerators.
 - Small networking and smart I/O devices.
 - Storage array controllers.
- Hyperscale - designs optimized for Data Center and HPC Servers.
 - Large-scale multi-chiplet integration with optional HBM.
 - Cloud computing server systems.

See [Hardware on page 14](#) for details of the IP components used in RD-V1.

2.1.1 Deliverables

The following deliverables are provided to enable software development:

- The *Arm® Neoverse™ V1 reference design Software Developer Guide*, a document that provides a high-level overview of RD-V1, including the architecture from which the RD-V1 design has been derived, the software stack, and the FVP.
- A Fixed Virtual Platform (FVP), providing a software model of the RD-V1 design to explore from a software perspective.
- An integrated software stack, providing a starting point for custom modification, extension, and development.

2.1.2 Features

RD-V1 has the following main features:

- Built around Arm® Neoverse™ V1 cores and CMN-650.
- Extension capabilities by using external interfaces, enabling partners to add high-performance peripherals.

2.1.3 Configurations

Configurations supported by the FVP are detailed in *Fixed Virtual Platform on page 49*.

2.2 Compliance

The RD-V1 design complies with, or includes components that comply with, the following specifications:

- Arm® *Architecture Reference Manual Armv8, for Armv8-A architecture profile*
- Arm® *Generic Interrupt Controller Architecture Specification, GIC architecture version 3 and version 4*
- AMBA® *5 CHI Architecture Specification*
- Arm® *Server Base System Architecture, version 6.0 (Level 4)*
- AMBA® *AXI and ACE Protocol Specification*
- Arm® *System Memory Management Unit Architecture Specification, SMMU architecture versions 3.0, 3.1 and 3.2*

3 Hardware

Two configurations of RD-V1 are supported:

- RD-V1 FVP models Config-M, a single-chiplet system with 16 Arm® Neoverse™ V1 cores.
- RD-V1 quad-chiplet FVP models a reduced-size variant of Config-XL, consisting of four compute subsystems linked by CMN-650 CML. It provides a functional model of a quad-chiplet system. Each subsystem contains four Arm® Neoverse™ V1 cores, for a total of 16 cores in the FVP.

For more information, see [Fixed Virtual Platform on page 49](#).

3.1 System architecture

The RD-V1 design is partitioned into functional blocks, which are known as elements. Elements are a combination of major IP and the supporting logic around them. Some features of the design use multiple elements. The element-based design approach provides flexibility, scalability, and modularity. This section describes these elements, and features that are implemented across multiple elements.

Figure 3-1 shows the top-level architecture.

[illegible]

The following elements have been defined:

- Processor element – The Processor element contains high-performance Armv8.4-A Arm® Neoverse™ V1 cores, Direct DSU, debug, and power management components.
- Debug element – The Debug element provides real-time trace facilities for the Application Processors (APs), the Manageability Control Processor (MCP), and the System Control Processor (SCP).
- Interconnect element – The Interconnect element contains:
 - A coherent mesh network with coherent multichip support using the CCIX standard, which CMN-650 implements.
 - A GIC distributor (GIC-D) and Interrupt Translation Service (ITS) logic, which GIC-700 implements.
 - A System Memory Management Unit (SMMU) with Translation Control Unit (TCU) and Translation Buffer Unit (TBU) logic, which MMU-600 implements.

- Multiple instances of NIC-450 switches.
- Miscellaneous blocks such as clock domain bridges, power management, and other wrapper glue logics.
- The Interconnect element also has master and slave extension interfaces.
- MSCP element – The Manageability and System Control Processor (MSCP) element implements the Cortex-M7-based SCP and MCP. It connects to all system control and power control logic for the relevant elements.
- Memory element – The Memory element contains a memory controller that implements an AMBA® AXI data path to the cache-coherent interconnect. The Memory element wraps the memory controller with the necessary glue-logic to integrate with the rest of the subsystem. It contains the inline external Arm TrustZone controller TZC-400, an Address translation function, power management, clock and reset generation, and control logic.
- Base element – The Base element contains multiple system peripherals, such as generic and watchdog timers, scratch RAM, boot ROM, firmware ROM, Universal Asynchronous Receiver-Transmitter (UART), and STM-500. It also includes NIC-450, to connect the peripherals.
- Clock element – The Clock element contains the clock generation logic for system-level clocks, but not for clocks that are internally generated by the Processor element, Debug element, and Memory element. The clock element logics are developed for Arm internal use, but you can use these details as a reference.

Features implemented across multiple elements:

- Debug and trace, which SoC-600 implements, with full support for CoreSight debug and trace in all cores in the Processor element, SCP, MCP, and CMN-650.
- System boot, which is implemented using Cortex-M7 and Arm® Neoverse™ V1 cores to support the Secure Boot.
- Security Architecture support. RD-V1 supports multiple security features, which are implemented by GIC-700, CMN-650, NIC-450, MSCP element, and associated CoreSight components to enable system level security.
- Power and clock management functions.

3.2 Processor element

The Processor element contains high-performance Armv8.4-A Arm® Neoverse™ V1 cores, Direct DSU, debug, and power management components.

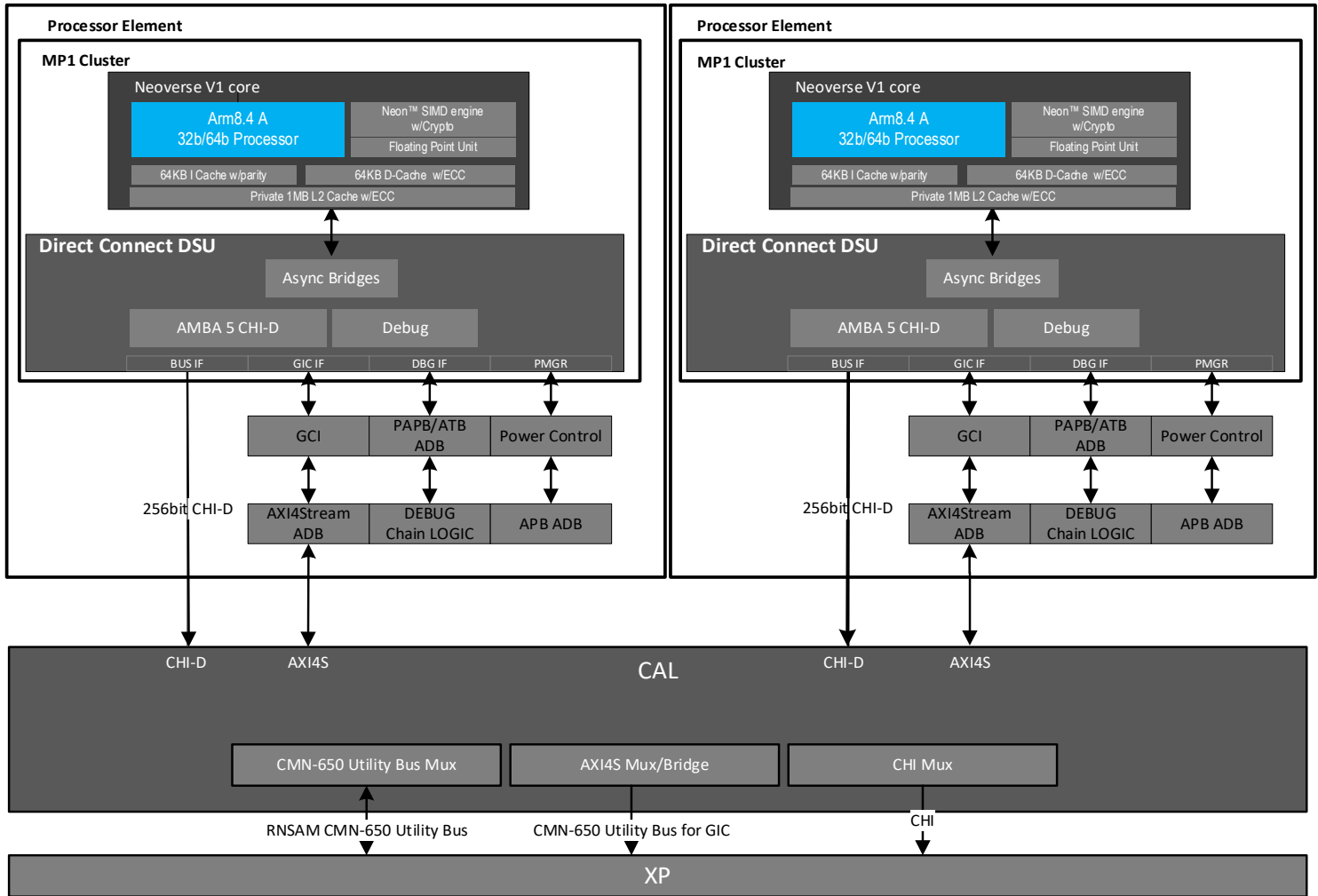
The Processor element has the following features:

- One MP1 Arm® Neoverse™ V1 core:
 - Arm v8.4 architecture support
 - 48-bit address space
 - Memory Partitioning and Monitoring (MPAM) for per-thread resource management
 - SVE extensions

- RAS, PMU Extensions
- Enhanced Cryptography
- Nested virtualization support for hypervisors
- Cache clean to persistent memory for high-availability services for Hyperscale
- JavaScript type conversion, enhanced cryptography, pointer authentication
- Arm® Neoverse™ V1 adaptive aggressiveness support (CBusy mechanism)
- Pin-based power control for Max Power Mitigation Mechanism (MPMM) and Dispatch Throttling (DT)
- Contains 1MB private L2 cache for each core and direct connect to CMN-650, which supports the next caching level
- Supports separate power domain per core, synchronous Dynamic Voltage and Frequency Scaling (DVFS) for each core
- Supports 256-bit Arm AMBA Coherent Hub Interface Issue D (CHI-D) interfaces to CMN-650
- Supports clock crossing bridge options that are configured as asynchronous between cores and Direct connect DSU, and synchronous between CMN-650 and Direct connect DSU
- Supports integrated Redistributor in the cluster power domain which maintains the Private Peripheral Interrupts (PPIs) and Software Generated Interrupts (SGIs) for cores within the cluster
- Supports ELA-500 support for each core
- Supports TrustZone technology

Figure 3-2 shows the block diagram of the Processor element. The two Processor elements are connected to the XP in CMN-650 through Component Aggregation Layer (CAL) connectivity.

Figure 3-2 Processor element block diagram



Related references

- [Processor element power states on page 35.](#)

3.3 Interconnect element

The Interconnect element contains:

- A coherent mesh network with coherent multichip support using the CCIX standard, which CMN-650 implements.
- A GIC distributor (GIC-D) and Interrupt Translation Service (ITS) logic, which GIC-700 implements.
- A System Memory Management Unit (SMMU) with Translation Control Unit (TCU) and Translation Buffer Unit (TBU) logic, which MMU-600 implements.
- Multiple instances of NIC-450 switches.

- Miscellaneous blocks, such as clock domain bridges, power management, and other wrapper glue logics.

The Interconnect element also has master and slave extension interfaces.

Related references

- [Interconnect element voltage and power domains on page 36.](#)

3.3.1 Coherent mesh network

CMN-650 implements the coherent mesh network. The coherent mesh network has the following features:



Configurations supported by the FVP are described in Section 5.1 on page 49.

- Fully compliant with CHI-D protocol.
- High-performance, distributed SLC support.
- Supports the CCIX protocol for coherent multichip or off-chip hardware accelerator communications. This protocol allows memory system masters, caches, and memory to remain coherent and consistent among multiple chips from the same or different vendors.
- Distributed Virtual Memory (DVM) message transport between masters.
- Quality of Service (QoS) regulation for shaping traffic profiles.
- Supports system cache group and striping capability on SLC partitions and memory interfaces.
- A Performance Monitoring Unit (PMU) to count performance-related events.

The system cache includes an integrated Point-of-Serialization (PoS) and Point-of-Coherency (PoC) and can be used as both to support compute and I/O caching.

- The Snoop filter configured as either 32MB (Config-M) or 64MB (Config-L).
- Dual DAT and RSP channels supported in mesh for Config-L to reduce congestion within the mesh.
- Supports CMN-650 Utility Bus interface, which is a generic transport layer in the mesh, leaving request and response fields to be defined and processed by requestor and responder. Used for internal communications such as configuring nodes, programming RNSAM, trace and performance events, error reporting, and Q- and P-channel controls.
- CMN-650 supports Memory Partitioning and Monitoring (MPAM) for per software thread shared resource management.
- Control of Request Node (RN) device inclusion or exclusion by a SYSCOREQ/SYSCOACK hardware interface.

- CMN-650 provides the option to connect either native CHI-based memory controllers or AXI-based memory controllers by using a bridge support called SBSX. The SBSX bridge and CMN-650 support Completer Busy (CBusy) signaling to Arm® Neoverse™ V1, which adaptively adjusts its aggressiveness to better manage overall large system performance.

The following selections determine the CMN-650 mesh structure:

- Requesting master selection:
 - The number of masters with coherent caches in the SoC determines the number of RN-F ports that are required.
 - The number of masters without internal coherent caches in the SoC determines the number of RN-D components that are required.
- Home Node (HN) selection:
 - The SLC and snoop filter size requirements determines the number of HN-F instances. The appropriate number of HN-F nodes are selected in the system according to the performance requirement and area trade-off.
 - The total bandwidth requirements of the AXI4 slave bandwidth and the physical placement of these slave peripherals determine the number of HN-D and HN-I instances.

Figure 3-3 shows the CMN-650 Interconnect element mesh network 3 x 5 used in Config-M.

Figure 3-3 Interconnect element mesh network 3 x 5 (Config-M)

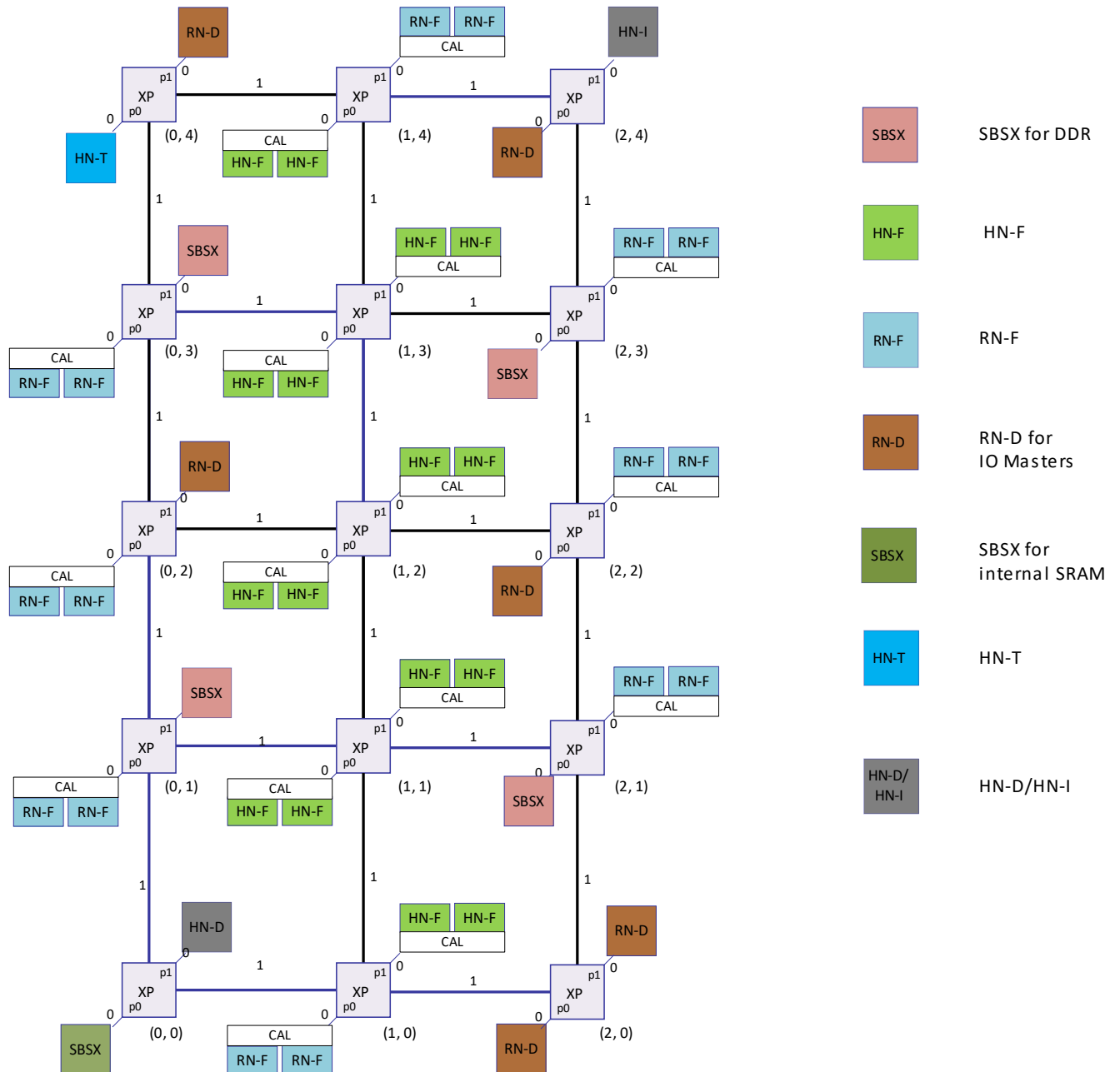


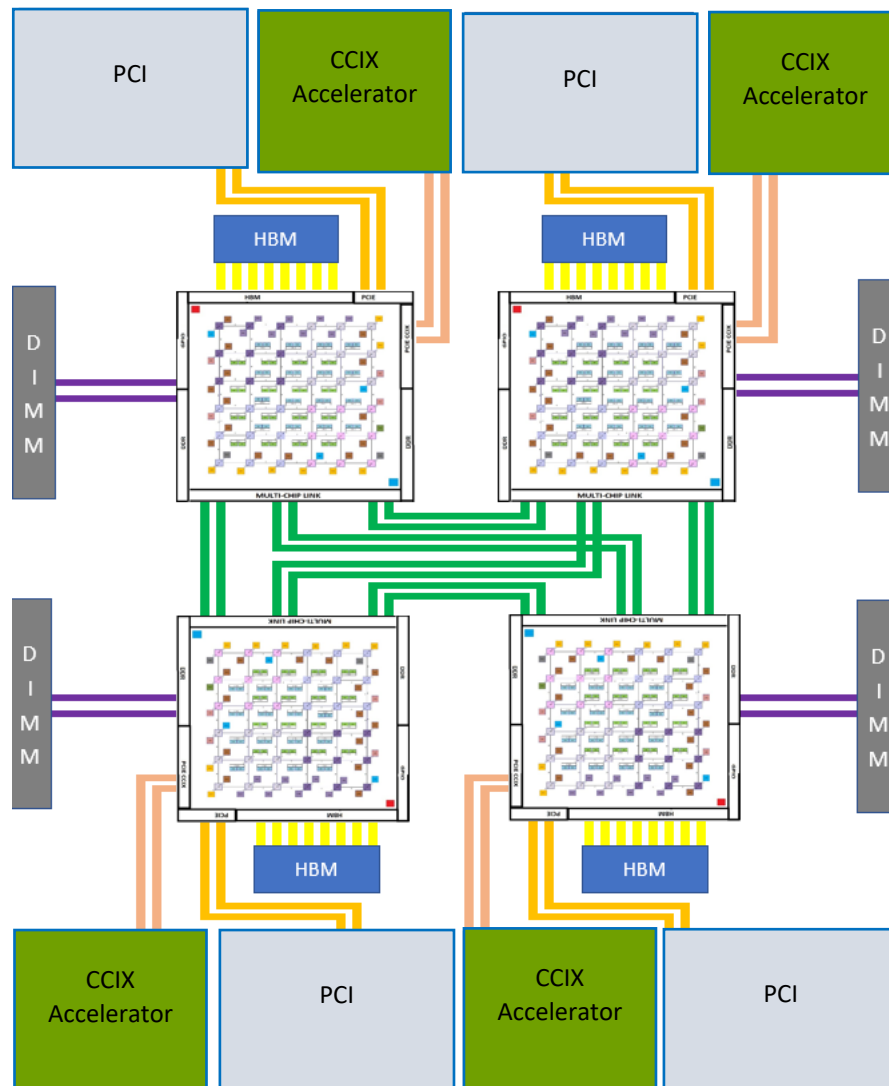
Figure 3-4 shows the CMN-650 interconnect element mesh network 6 x 6 used in Config-L.

Figure 3-4 Interconnect element mesh network 6 x 6 (Config-L)



Figure 3-5 shows the chiplet system topology, Config-XL, which is 4 x Config-L connected using CCIX links.

Figure 3-5 Chiplet system topology for Config-XL



3.3.2 Generic interrupt controller

GIC-700 implements the generic interrupt controller. The generic interrupt controller has the following features:

- Arm GICv3 and GICv4.1 architecture, for Armv8 and Armv8.4-A.
- Distributed structure – multiple ITS blocks can be distributed across the system.
- Security Extensions.
- Distributed, one-per-cluster PPI and SGI (PPI/SGI also known as GCI, GIC Cluster Interface).
- Shared Peripheral Interrupt (SPI) supported up to 1984 interrupts.
- P-Channel and Q-Channel support for low-power control.

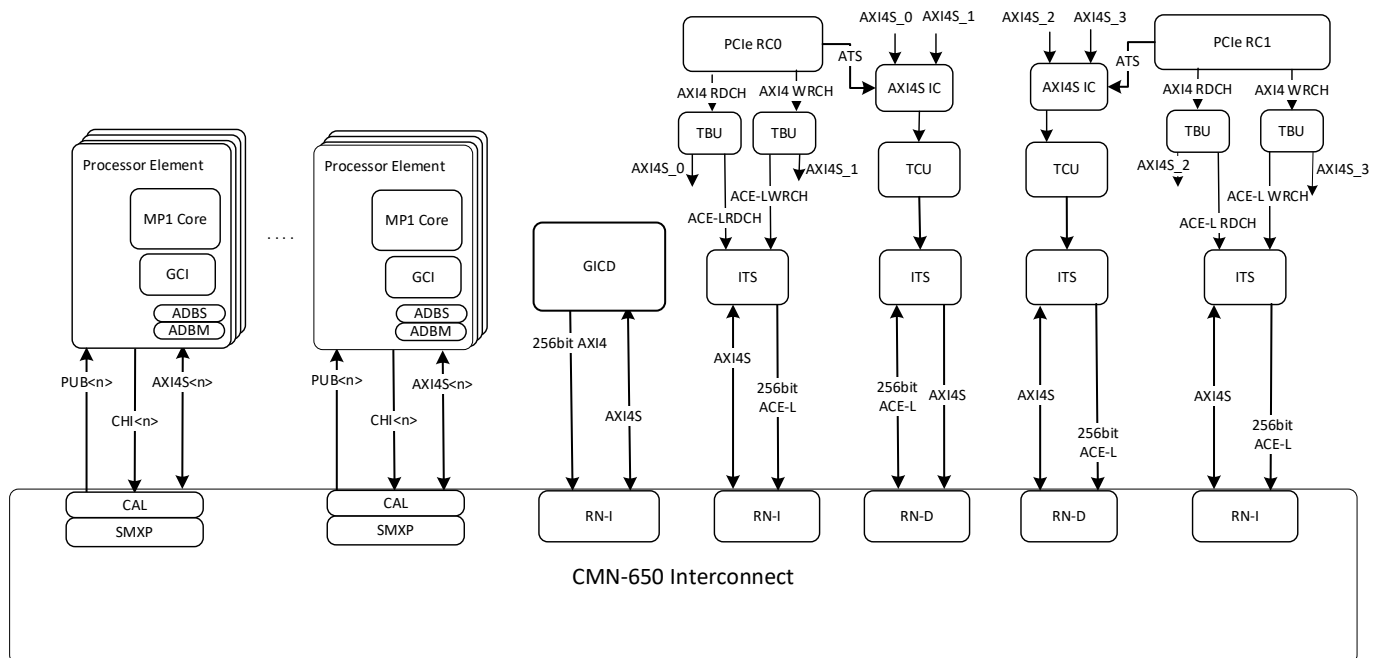
- AXI4-Stream interface supported on GIC-700 blocks.



Backwards compatibility with GICv2 programmer model is not supported.

Figure 3-6 shows the GIC-700 interrupt routing topology used in RD-V1. A standalone AXI4-Stream interconnect bus fabric is used for the GIC interrupt communications between the cores (that are on the same chip) and GIC-D. The CMN-650 Utility Bus interface connectivity is used for GIC interrupt communications, including chip-to-chip communication, using the AXI4-Stream to the bridge.

Figure 3-6 GIC topology used in RD-V1



3.3.3 System memory management unit

The MMU-600 implements the system memory management unit. The system memory management unit has the following features:

- Compliant with the SMMUv3.1 architecture.
- Provides address translation functions for memory accessing I/O masters, and external PCI master interface.
- Distributed design allows for multiple TBUs to one centralized page walker block, called the TCU.
- The number of TCUs and TBUs can be configured on the external I/O master ports.
- The TCU and TBU can be physically dispersed in a system.
- Single or two stage address translation for PCIe traffic:

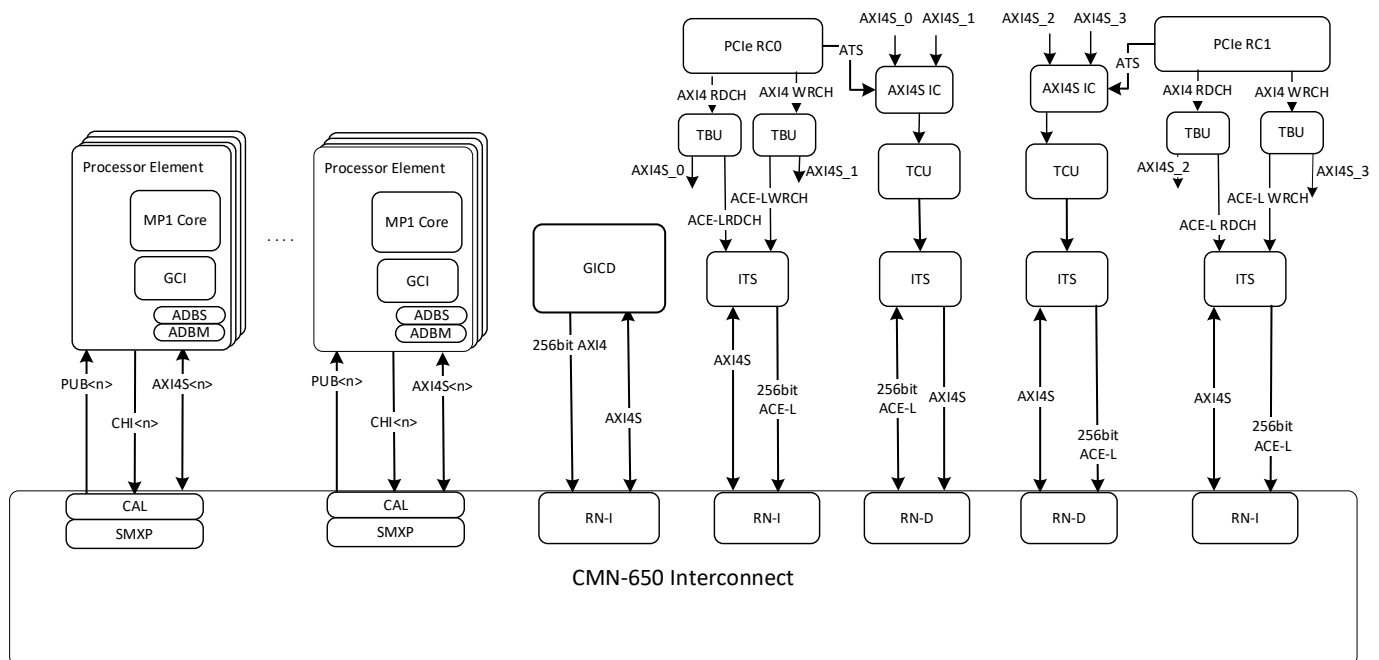
- Single stage: From Virtual Address (VA) to Physical Address (PA).
- Two stage: Stage 1, from VA to Intermediate Physical Address (IPA). Stage 2, from IPA to PA.
- Support for PCIe features, including ATS and PASIDs.
- Scalable to millions of active contexts.
- Provide virtualization support for SoC-600 debug ETR interface.

The number of TCUs and TBUs required in the system depends on the bandwidth and latency performance requirements for the I/O interfaces.

Figure 3-7 shows the MMU-600 topology in RD-V1. One TCU with two TBUs are supported in each PCIe-RC port, with dedicated TBUs on the Write and Read paths to meet PCIe bandwidth requirements, for example x 16 PCIe Gen4. Two PCIe ports are supported in RD-V1, as shown in the diagram.

In RD-V1, a dedicated external AXI4-Stream bus interconnect has been used to connect the TBUs with the TCU, as they are co-located.

Figure 3-7 System MMU topology used in RD-V1



3.3.4 Interconnect extension interfaces

The Interconnect element contains three NIC-450 Network Interconnects that provide master and slave extension interfaces. These interfaces enable you to connect your masters and slaves to the reference design.

The use of NIC-450 in the interconnect element enables the interfaces to be extended by grouping multiple interfaces under a CMN-650 device port. In particular, the NIC-450 can be used to extend

or group the non-performance critical interfaces to optimize the number of nodes and XPs required in CMN-650.

You can decide to use the NIC-450 based interface extension differently, depending on your performance and system requirements.

3.4 MSCP element

The Manageability and System Control Processor (MSCP) element implements the Cortex-M7-based SCP and MCP. It connects to all system control and power control logic for the relevant elements.

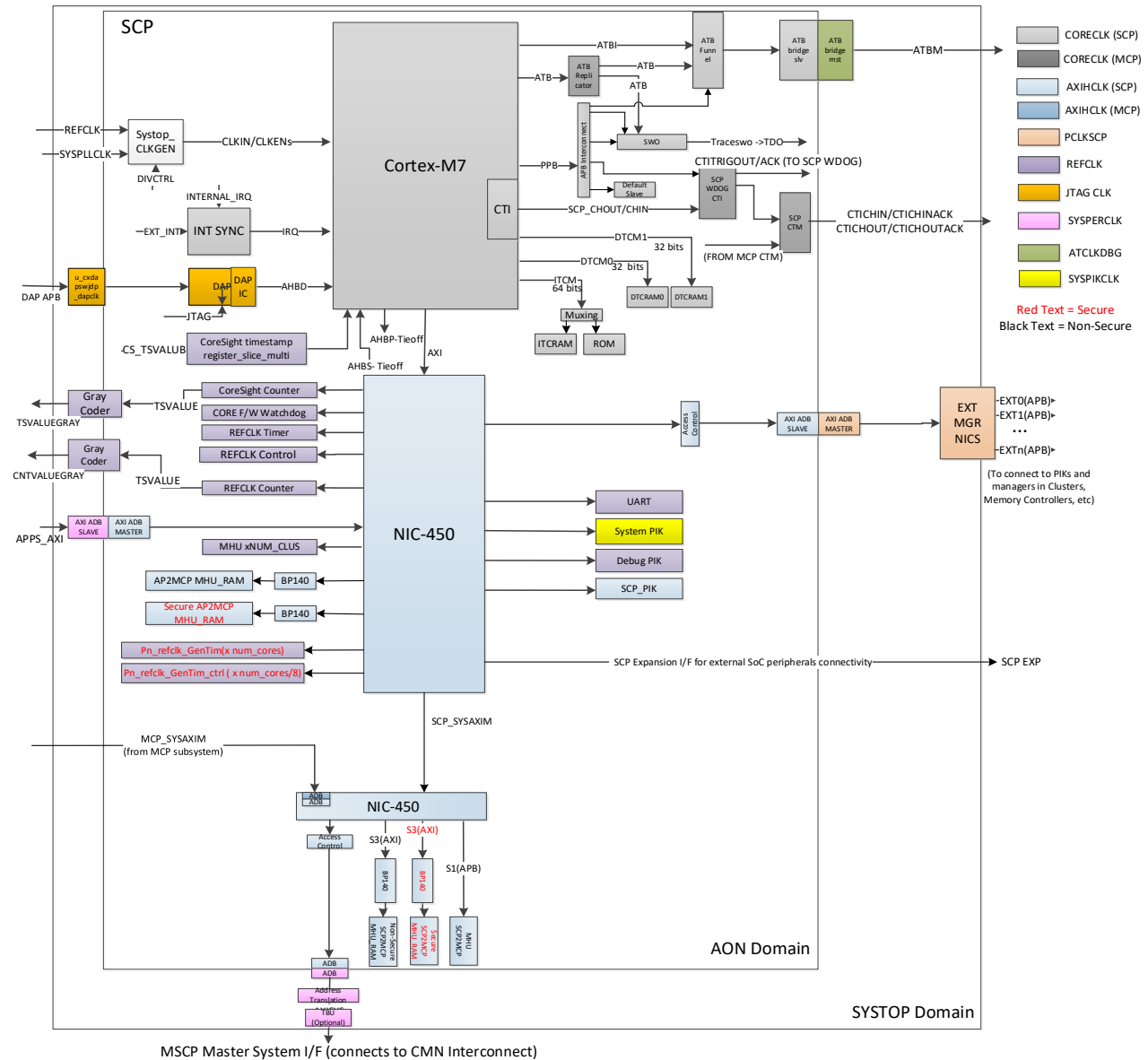
3.4.1 SCP

The SCP manages the overall power, clock, reset, and system control of RD-V1. It has the following features:

- A Cortex-M7 processor that operates in the Secure world only.
- Boot sequencing during system start-up.
- Communication with the MCP for establishing trust and security integrity of the system.
- Responsibility for all the clock and power controls.
- Initial configuration and subsequent reset.
- Managing transitions between operating points, that is, external voltage regulator and clock management to support DVFS.
- Handling hardware wake-up requests from components such as timers and interrupts.
- Saving and restoring all states in the interconnect when powering down or powering up.
- Responsibility for managing a consistent matrix of device states across the whole system.
- Control of components that are both internal and external to the SoC, for example, PLL, Clock and Reset controls, and Power Management Integrated Circuit (PMIC).

Figure 3-8 shows the SCP block diagram, a Cortex-M7 based system with NIC-450 to extend and support interfaces such as Power Integration Kit (PIK) interfaces, watchdogs, timers, and counters.

Figure 3-8 SCP block diagram



The SCP physically contains several peripherals. These peripherals are used for basic operation, such as timers, instructions, and data memories, and to directly drive power-control logic. The SCP peripherals are:

- NIC-450
- Message Handling Unit (MHU) to communicate with MCP and processor cores. See [3.5.3 Message Handling Unit](#).
- SRAM for private data and code
- On-chip (trusted) and Secure boot ROM
- PCK-600/PIK

- PL011-UART
- AXI Mem Interface (BP140)
- SCP timing peripherals
- SCP watchdog timer
- REFCLK counter
- SCP REFCLK Generic Timer
- AP wake-up REFCLK Generic Timers
- CoreSight Timestamp counter

3.4.2 MCP

The MCP manages the communications with the external Baseboard Management Controller (BMC). It has the following features:

- A Cortex-M7 processor with private peripherals such as timers, UART, instruction, and data memory.
- Boot sequencing during system start-up.
- Establishing security integrity with SCP through software.
- Communicating with external BMC controller.
- Responsibility for handling reset of the SoC when the BMC controller sends a message to reset, or when indicated through General-Purpose Input/Output (GPIO).
- Responsibility for logging events and communicate to the external BMC controller.
- Responsibility, along with processor cores, for all the RAS manageability.

Figure 3-9 shows the MCP block diagram.

The diagram illustrates the internal architecture of the MCP, showing its connection to various external and internal components. The system is divided into two main domains: the SYSTOP Domain (top) and the AON Domain (bottom).

SYSTOP Domain Components:

- Cortex-M7:** The central processing unit, connected to the APB Interconnect and CTI.
- CTI (Cortex-M7 to Interconnect Bridge):** Manages data flow between the Cortex-M7 and the APB Interconnect.
- APB Interconnect:** Connects the Cortex-M7 to various peripheral blocks.
- ATB Replicator & ATB Funnel:** Manage the ATB (Advanced Trace Bus) signals.
- MCP WDOG G CTI:** Watchdog and gate control logic.
- MCP CTM (Cortex-M7 to Interconnect Bridge):** Manages data flow between the Cortex-M7 and the APB Interconnect.

AON Domain Components:

- CoreSight Timestamp register_slice_multi:** Provides timestamping for CoreSight.
- Core F/W Watchdog:** Monitors the system for faults.
- REFCLK Timer & REFCLK Timer CNTCTL:** Manage the REFCLK (Reference Clock).
- AXI ADB MASTER:** Manages AXI (Advanced eXtensible Interface) data flow.
- MHU_AP2MCPx1:** Manages the Main Host User Interface (MHU) for the AP2MCP.
- AP2MCP MHU_RAM & Secure AP2MCP MHU_RAM:** Memory buffers for the AP2MCP MHU.
- BP140:** Boundary Protection Registers (BPRs) for memory protection.
- MCP PIK (Platform Integrity Key):** Manages the integrity of the system.
- UART:** Universal Asynchronous Receiver/Transmitter for serial communication.

Legend:

- CORECLK (SCP)
- CORECLK (MCP)
- AXIHLCK (SCP)
- AXIHLCK (MCP)
- PCLKSCP
- REFCLK
- JTAG CLK
- SYSPERCLK
- ATCLKDBG
- SYSPKCLK

- MHU to communicate with SCP and processor cores, see [3.5.3 Message Handling Unit](#).
- SRAM for private data and code.
- On-chip (trusted) Secure boot ROM.
- PIK to handle the reset and power states of MCP.
- MCP timing peripherals.
- MCP Watchdog.
- MCP REFCLK Generic Timer.
- MCP UART0/1.
- AP UART1.



REFCLK Generic Timer control, MCP PIK, MCP Watchdog, timer peripheral components, and core Warm reset are all the same as the SCP.

3.4.3 Message Handling Unit

Both the SCP and MCP have a memory-mapped MHU peripheral that provides a mechanism to assert interrupt signals to facilitate inter-processor message passing. The message payload can be deposited into MHU buffer memories that are dedicated to each MHU. Therefore, the MHU is used as a message signaling mechanism.

The MHU asserts the following interrupts to be mapped between processors for message passing:

- A Non-secure and Secure interrupt mapped to one processor.
- A Non-secure and Secure interrupt mapped to the other processor.

The preceding interrupts are generated by writing to a corresponding 32-bit register. The use of 32 bits for each interrupt line enables software to provide more information about the source of the interrupt. For example, each bit of the register can be associated with an event type that can contribute to raising the interrupt.

Each MHU also has dedicated Secure (512 bytes) and Non-secure (512 bytes) MHU buffers to write request and response messages.

3.4.3.1 Message communication between processor cores, SCP, and MCP

- One MHU for each cluster to support message passing between SCP and cores in each cluster.
- One MHU support message passing between SCP and MCP.
- One MHU support message passing between cores and MCP.

Related references

- [MSCP voltage and power domains on page 36.](#)

3.5 Memory element

The Memory element contains a memory controller that implements an AMBA® AXI data path to the cache-coherent interconnect. The Memory element wraps the memory controller with the necessary glue-logic to integrate with the rest of the subsystem. It contains the inline external TrustZone controller TZC-400, an Address translation function, power management, clock and reset generation, and control logic.

3.6 Base element

The Base element contains multiple system peripherals, such as generic and watchdog timers, scratch RAM, boot ROM, firmware ROM, Universal Asynchronous Receiver-Transmitter (UART), and STM-500. It also includes NIC-450, to connect the peripherals.

Figure 3-10 Base element block diagram

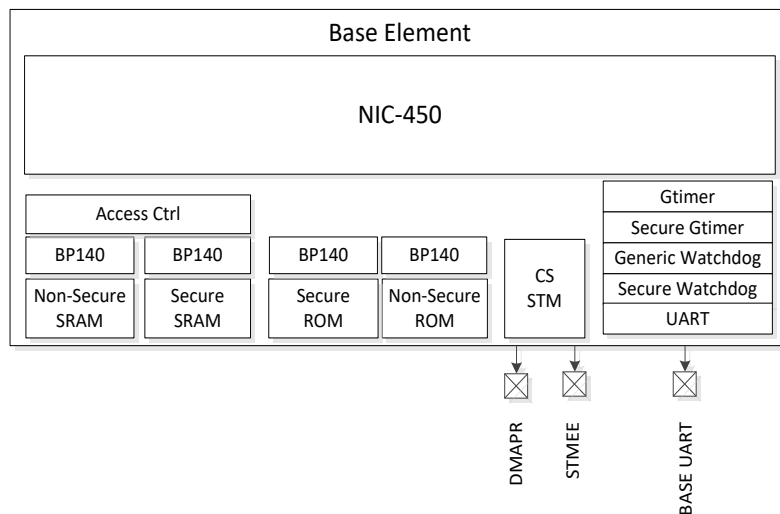


Table 3-2 lists the peripherals in the Base element.

Table 3-1 Peripherals in base element

Peripheral	Peripheral Description
AP_REFCLK_CNTCTL	AP Generic Timer control.
Secure AP Watchdog	Generic Watchdog Timer for Trusted applications.
Non-secure AP Watchdog	Generic Watchdog Timer for Non-trusted applications.
AP_REFCLK_S_CNTBase1	AP Generic Timer. This timer is Secure access only.
AP_REFCLK_NS_CNTBase0	AP Generic Timer. This timer permits Non-secure accesses.
Secure Boot ROM	Contains the code for initializing the AP boot process. This ROM is accessible in Secure mode only.
Non-secure Boot ROM	Contains code that is required during firmware update. This ROM is accessible in both Secure and Non-secure modes.
STM-500	STM-500.
PL011-UART	PrimeCell PL011 Universal Asynchronous Receiver Transmitter (UART).
BP140	PrimeCell Infrastructure AMBA 3 AXI Internal Memory Interface (BP140)

See [8.1.2 Peripherals Region](#) for information on the peripherals that are visible to the Processor element and the SCP element.

3.7 System Boot

System boot, which is implemented using Cortex-M7 and Arm® Neoverse™ V1 cores to support the Secure Boot. The subsystem provides:

- A Cortex-M7 based SCP designed to function as a Trusted subsystem.
 - A local, on-chip, Trusted boot ROM.
 - A local, on-chip, Secure SRAM to execute the main SCP firmware loaded in from external NVM.
- An Arm® Neoverse™ V1 based system with the following features:
 - On-chip, Secure Boot ROM intended for storing Trusted boot code.
 - On-chip, Non-secure ROM intended for storing Non-secure boot code.
 - On-chip, Secure SRAM intended for storing Trusted data.

In RD-V1, for Config-XL, each chiplet has its own dedicated MSCP (Cortex-M7 cores) and boot resources. The chip0 MSCP turns into a Master MSCP, as shown in Figure 3-9 on page 28. Each chiplet boots up independently, immediately after the power-on reset, and eventually communicates with the chip0 master MSCP using a side band serial interface.

4 Functional description

The major functionality of the RD-V1 design is described in this section.

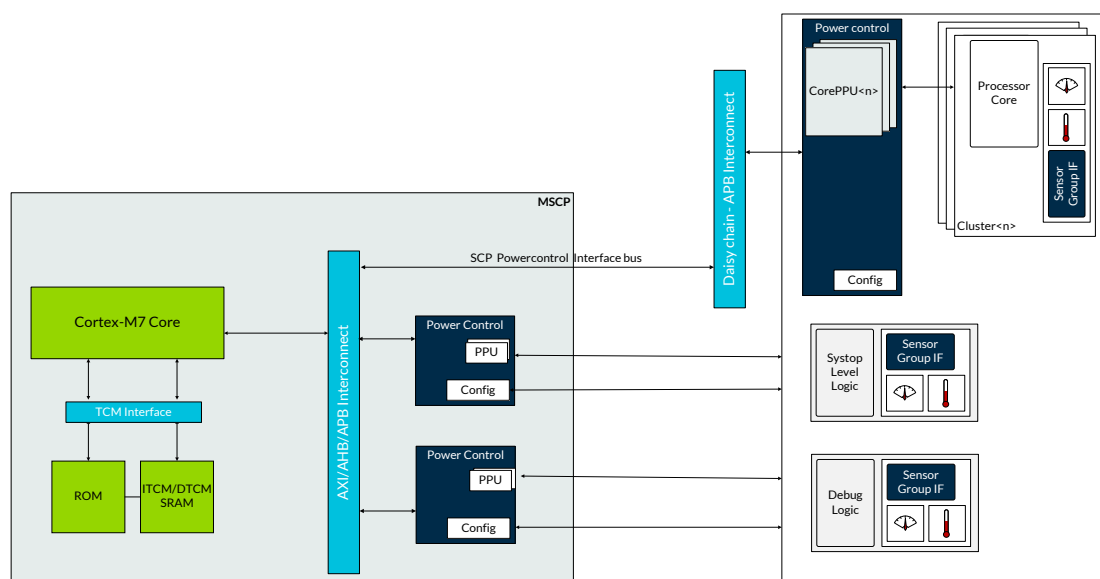
4.1 Power management

RD-V1 supports the following features for power management:

- The SCP, a Cortex-M7 core that controls power, clock, reset, and the static configuration of the system or subsystem.
- Multiple voltage domains to allow for DVFS on processor cores.
- Multiple power-gated regions providing comprehensive leakage management.
- Low-power, autonomous wake/sleep mode.
- Globally Asynchronous Locally Synchronous (GALS) clocking architecture:
 - All primary clock domains are independent from each other.
 - Simplifies frequency scaling and interfacing.
 - Enables an accelerated route to a higher-quality result for the physical design.
- Hierarchical clock gating across the system.

Figure 4-1 shows an overview of how the distributed power management system has been implemented in RD-V1. The power control blocks such as MSCP, PPU, and sensors are distributed across the system.

Figure 4-1 RD-V1 distributed power management overview



4.1.1 Voltage and power domains

RD-V1 supports the following voltage domains:

- VCPUn – Separate voltage domain for each cluster. For example, VCPU0, VCPU1, VCPU2.
- VSYS – The rest of the subsystem. Does not support DVFS.

A power domain is a collection of hardware modules within a voltage domain that share common power control. A voltage domain can have one or more power domains. A power-gated domain is a power domain, the power of which can be removed by on-chip power switches.

RD-V1 supports the following top-level power domains within the VCPUn and VSYS voltage domains:

- AONTOP – The power domain for the AON part of the logic. This domain is separate but not power-gated. This domain can be expanded to encompass other logic components in the rest-of-SoC.
- SYSTOP – System top power domain. The power domain for the rest of the system. This domain is power-gated. The SYSTOP power domain can be expanded to encompass other logic components in the rest-of-SoC.
 - SCRAM – Scratch RAMs power domain.
 - SLCRAM0/1 – System cache RAMs power domain.
 - SFRAM – Snoop filter cache RAM power domain.
- DBGTOP – Debug top power domain. The system-level debug supported with a separate power domain to optionally implement with a power gating to save power.
- CLUSnCPU0 – Power domains supported at individual cluster level, where “n” is the cluster number, 0 to 15 for Config-M, 0 to 31 for Config-L (in RD-V1 there is one core per cluster).

4.1.2 Domain hierarchy

The voltage and power domain hierarchies shown in Figure 4-2 and Figure 4-3 are illustrative of the logical relationships between the power-gated domains.

Figure 4-2 VSYS hierarchy, all configurations

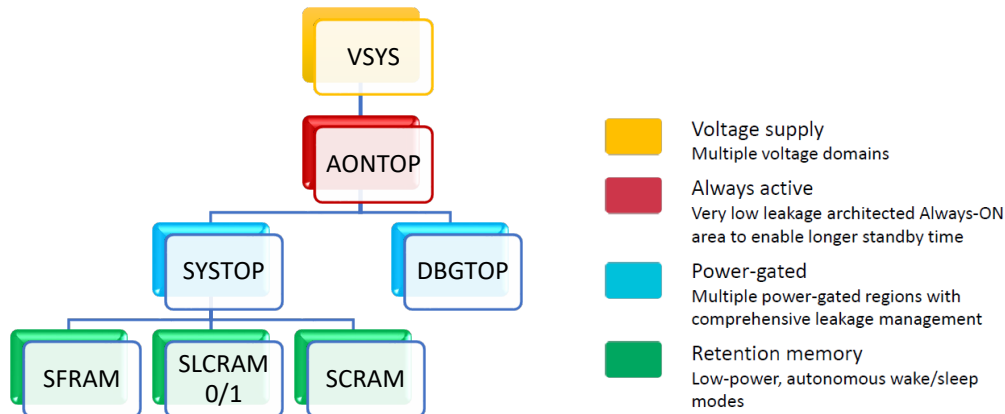
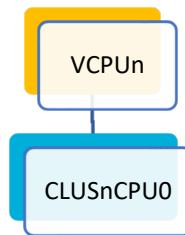


Figure 4-3 VCPUn hierarchy, all configurations



A PPU controls each power domain. PPUs contain a programmer-visible state that the SCP uses to set the power policy and control the power mode of the power domains.

4.1.3 Element-specific power management

This section gives information about the different power management techniques that are used by individual elements in the RD-V1 design.

4.1.3.1 Processor element power states

The Processor element supports all core power modes defined in the Arm® Neoverse™ V1 core documentation. In RD-V1, per core/cluster (CLUSnCPU0) power control is supported. It is associated with a dedicated PPU unit to control the power states of each individual core or cluster independently. In RD-V1, the per core power domain is the same as the per-cluster power domain as there is just one core for each cluster.

Table 4-2 describes legal power policy state combinations for the Processor element voltage domains. It also shows the dependencies to relevant power-gated regions under VSYS and VCPUn in respective configurations. The states of all other power-gated regions should be considered orthogonal to the core and cluster power states.

The ON power policy encompasses RUN and IDLE (STANDBYWFI) device states.

Table 4-1 Processor element power policy states

VSYS	CLUS<n>CPU0	SYSTOP
ON	ANY	ON
ON/OFF	OFF	ON/OFF

4.1.3.2 Interconnect element voltage and power domains

The Interconnect element lies in the VSYS voltage domain. It is encapsulated inside the SYSTOP power-gated domain.

CCIX power domain and clocking support

- The CCIX LA block is in a different power domain to the CMN-650 mesh. From the SoC perspective, the CCIX LA is in the CCIX transport (such as PCIe, USB, and D2D) power domain (CXS power Q-Channel). For reference, the CMN-650 mesh logic is part of the SYSTOP power domain.

GIC-700 power domains

- The distributor GIC-D is in the SYSTOP power domain.
- The ITS blocks are in the same power domain as the respective PCIe master power domain.
- The PPIs are in the respective core or cluster power domains.

MMU-600 power domains

- The TCUs are in the SYSTOP power domain.
- The TBUs are in the same power domain as the respective PCIe master power domain.

4.1.3.3 MSCP element voltage and power domains

The MSCP element is in AON domain. It connects to all power control logic for the relevant elements.

4.1.3.4 SCP power control

System PIK

This block has a set of registers to control the system PPU and manages the power domains and state transitions by the SCP firmware code.

SCP Power Control

The SCP PIK is designed to work with the SCP element. The SCP PIK is the first one to come out of reset and generates all the reset controls for the SCP.

Power Control Sequences - SYSTOP Power Domain

The SYSTOP region is a switchable domain that must be first on and last off. This is stipulated because no component, except for the SCP, can perform useful work if this region is not available.

The following tasks are required for transition to CSS.SLEEP1 state:

1. Power state pre-conditions:
 - All power domains in the Processor elements are OFF.
 - The AP wake-up latency requirements must also permit the exit latency from the SYSTOP OFF state.
2. State save:
 - The SCP saves any system IP states that are not already saved.
3. Memory controller shutdown and DDR Self-Refresh:
 - All memory controllers must be placed in an architectural low-power state.
4. Stop PLLs:
 - If a PLL shares external logic that remains active, then it can remain running.
5. Power down:
 - The SCP sets the POLICY field in the Power Policy Register to OFF for the SRAM PPU and then the Interconnect PPU. Before this occurs, any state in the SRAM must be saved to memory before the DMC shutdown. Any remaining input clocks to the SYSTOP region, for example REFCLK and any running PLLs, can be managed by the SCP at this point. Turn OFF if appropriate.

The following tasks are required to manage a SYSTOP OFF to ON transition:

6. Power up:
 - The SCP sets the POLICY in the Interconnect PPU and then the SRAM PPU Power Policy Register to ON. At completion of the sequence, indicated by PPU interrupt, REFCLK, and any already running PLL clocks are available.
7. Start PLLs:
 - Restart any required PLLs and wait for lock interrupts.
8. State restore:
 - The system IP, SRAM memory state, and memory controller states are restored.
9. Start memory controllers:
 - Perform the DDR PHY start-up sequence and then configure the memory controller to the active state.

Power Control Sequences – CLUSnCPU0 Power Domain

Usually, before powering up any power domain in a cluster, you must first power up and configure the core power domain. But, as there is only one core per cluster in RD-V1, there is no cluster power domain.

The following sequence gives a general overview of the tasks that are required to manage a CLUSnCPU0 OFF to ON transition:

- Set the POLICY bit in the CLUSnCPU0 PPU policy register to ON.

- CLUSnCPU0 powerup is complete when an interrupt is detected from the PPU.
- The SCP then powers a core by setting the POLICY bit in one of the core PPUs to ON.
- When one core is available, and the reset is released, it determines from a register the boot type and L2 cache retention status and takes appropriate action.

The following tasks are required to manage a CLUSnCPU0 ON to OFF transition:

- To power down a core, the OS sends a message through the MHU to the SCP. This message states that the CLUSnCPU0 is to be powered down and clarifies whether the L2 Cache requires flushing using the SCP-controlled L2 cache flush mechanism. The message indicates that the OS is ready to power down, with context saved, at the next STANDBYWFI.
- When saving context, the core must mask interrupts to the core in the GIC-700 to prevent exit from STANDBYWFI before the power down sequence has completed.
- The SCP programs the POLICY bit in the Policy Register of the core PPU to OFF.
- When the processor STANDBYWFI is asserted, the core PPU starts the power down sequence.
- Program the CMN-650 to remove the cluster from coherency.
- Program the CLUSnCPU0 PPU policy to OFF.
- The power sequence completes when the SCP receives an interrupt from the PPU.

4.1.4 Arm® Neoverse™ V1 core power and performance management

The Arm® Neoverse™ V1 core implements power management features that enable a system to regulate high-activity workloads with the aim of trading off peak performance for improved power efficiency.

The Arm® Neoverse™ V1 processor features include:

- **Max Power Mitigation Mechanism (MPMM)**
Aims to limit the maximum time-averaged power based on a set threshold below the maximum virus workload. Three MPMM parameter sets, called "Gears", enable the threshold to vary during runtime based on pin control settings. MPMM does not address short-term di/dt demands on the Power Delivery Network (PDN) but can limit the time-averaged power to an expected worst-case real workload.
- **Activity Monitoring Unit (AMU)**
A set of PMU counters that is dedicated to monitoring activity within the processor that are always accessible over the debug APB sideband bus even when the Arm® Neoverse™ V1 core is powered down. In addition, the AMU can be exposed at various levels of the EL# software stack based on a boot time configuration. The AMU can be used for software-based power management.
- **Performance Defined Power (PDP)**
A "smart" autonomous mechanism within the Arm® Neoverse™ V1 core that power gates aspects of the processor microarchitecture based on the running state.
- **Dispatch Throttling (DT)**
Limits the maximum number of instructions that can be dispatched within a given window of

processor cycles. While MPMM and PDP have limited capacity to reduce power consumption, DT can totally stop the processor execution under pin control.

In RD-V1, the Processor element implements PPUs for each Arm® Neoverse™ V1 core, to manage power per core. Pin control is also supported for the Arm® Neoverse™ V1 cores to control the power states of each individual core independently.

Table 7-61 shows the pin control register bits implemented in the "PE configuration" register that is part of the Processor element. This register, with the programming of the Arm® Neoverse™ V1 CPUPPMCR_EL3 register, drives configuration inputs to the core/cluster. SCP power management Firmware can program these values during boot and runtime.

In RD-V1, the following settings are used:

- For MPMM:
 - MPMMEN Enabled
 - MPMM Gear 0
 - MPMM Gear 2 (for vector workloads)
- For DT
 - DT Enabled
 - DT Threshold = 50%
- For PDP
 - PDP Enabled
 - PDP Threshold = Engage if empty 95% of cycles, disengage if stall at 5% of cycles, default recommended

4.2 Clocks

This section describes the clocks in the RD-V1 design.

4.2.1 Input clocks

Table 4-4 lists the input clocks in the RD-V1 design, some of which, as indicated, are driven by Phase Locked Loops (PLLs).

Table 4-2 Input clocks

Clock signal	PLL lock signal	Description
REFCLK	N/a	Main input clock that SCP/MCP boots coming out of reset.
CPULLCLK<n>	CPULLLOCK<n>	Core PLL, one per core, where n = 0-15 for Config-M and n = 0-31 for Config-L.
INTPLLCLK	INTPLLLOCK	PLL to generate interconnect clock.
SYSPLLCLK	SYSINPLLLOCK	Clock input for the main system. Used to generate the clock for PPUs.
DDRPLLCLK	DDRPLLLOCK	Clock input for the memory system.

Clock signal	PLL lock signal	Description
EMCLK0- EMCLK<n-1>	N/a	Clock input for external master expansion I/O ports with TBU or ITS present on the expansion I/O path. "n" represents number of external Master expansion I/O ports.
CXS clock <n>	N/a	External clock input for the CCIX Stream interface port. CXS clock per CML link. "n" represents number of CCIX/CML links. It applies to Config-L and Config-XL.
SWCLKTCK	N/a	Clock that drives the combined JTAG and Serial Wire Debug (SWD) interface.

4.2.2 Output clocks

Output clocks are generated and routed out of the subsystem for external SoC integration purposes. See Table 4-5.

Table 4-3 Output clocks

Clock	Description
INTCLK/INTCLKFREE	Output clock for Coherent Expansion ACE-Lite Slave Ports, CEACES0 to CEACESn.
	Output clock for Expansion ACE-Lite Slave ports, ESACELM0 to ESACELMn.
	Output clock for Expansion ACE-Lite master interface, EACELM0.
MC<n>_CLK	Memory controller output clock to DDR PHY. "n" represents No. of memory controller interfaces in a system. This is from the same clock source as MC<n>_DFICLK. It is optional, depending on the DFI/PHY and Clock ratio support requirements, and it must be used accordingly.
MC<n>_DFICLK	Memory controller DFI output clock for DFI Master Interfaces. "n" represents the number of memory controller interfaces in a system.
GICCLK GICCLK_FREE	GIC clock. All the incoming GIC_EXT_INT are assumed level sensitive. If the incoming interrupts are edge-triggered, then they must be synchronized to this clock domain. They must be as wide as one clock pulse of this clock. The GIGCLK_FREE is a free running clock, NOT a gated version of GICCLK. Both versions of the clocks are exported out of the subsystem.
SYSPERCLK SYSPERCLK_FREE	Output clock for expansion AXI4 master interface, EAXIM. Both versions of the clocks are exported out of the subsystem.
PCLKSCP PCLKSCP_FREE	Clocks the APB interfaces from the SCP element. Both versions of the clocks are exported out of the subsystem.
PCLKDBG PCLKDBG_FREE	Output clock for CoreSight APB Master Expansion Interface, CSAPBME.
TRACECLK	Output clock for Trace Out Port Interface.
SCPAXICLK	Output clock for SCP AXI Expansion Interface. Exporting this interface at the subsystem level is just a reference implementation.
MCPAXICLK	Output clock for MCP AXI Expansion Interface. Exporting this interface at the subsystem level is just a reference implementation.

4.2.3 System clocks

RD-V1 internally derives clocks that are used in the components of the subsystem. These clocks are only generated when the VSYS.SYSTOP power domain is powered. Each clock can be individually controlled.

Table 4-4 Subsystem internal derived clocks

Clock signal	Description	Source PLL	Comments
CORECLK<n>	Core clock	CPULLCLK<n>	Each core is clocked independently of each other. The clock of each core is generated out of processor PLLs.
INTCLK	Coherent Interconnect Clock	INTPLLCLK	Clocks the interconnect and other expansion master and slave ports. This clock is generated from INTPLLCLK.
SYSPERCLK	Interconnect clock	SYSPLLCLK	Clocks the NIC-450 in the Base element. This clock is generated out of SYSPLLCLK.
SYSPCLKDBG	Debug APB clock	SYSPLLCLK	Clocks the APB interfaces and bridge at the top level.
DBGCLK	Debug clock	SYSPLLCLK	Debug clock.
TRACECLK	Trace clock	SYSPLLCLK	Debug Trace clock.
DBGCLK_FREE	Free Debug clock	SYSPLLCLK	Free Debug clock.
PCLKDBG	Debug APB clock	SYSPLLCLK	Clocks the APB interfaces.
ATCLKnDBG	Debug ATB clock	SYSPLLCLK	Clocks the ATB interfaces from the Debug Element.
GICCLK	GIC clock	SYSPLLCLK	Clocks the GIC-700.
TCUCLK	SMMU clock	SYSPLLCLK	SMMU TCU clock. Each TCU would have its own clock.
SCPCORECLK	SCP core clock	SYSPLLCLK	Clocks the core of the SCP. This clock is sourced by REFCLK initially during a full subsystem reset. It is later switched over to a divided down version of SYSPLLCLK by the SCP firmware when the PLL is locked.
MCPCORECLK	MCP core clock	SYSPLLCLK	Clocks the core of the MCP. This clock is sourced by REFCLK initially during a full subsystem reset. It is later switched over to a divided down version of SYSPLLCLK by the MCP firmware when the PLL is locked.
PCLKSCP	SCP APB clock	SYSPLLCLK	Clocks the APB interfaces from the SCP element.
UARTCLK	UART clock	SYSPLLCLK	Clock for the UART module.

Table 4-5 shows system level clocks that are generated by the CGU.

Table 4-5 Top-level clocks generated by the CGU

Clock signal	Source clock	Divider	Gated	Comments
INTCLK	REFCLK	No	Yes	Interconnect clock.
	INTPLLCLK	Yes		

Clock signal	Source clock	Divider	Gated	Comments
MC<n>_CLK MC<n>_DFICLK	DDRPLLCLK	No	No	Per memory controller interfaces.
		Yes		
GICCLK	REFCLK	No	Yes	GIC clock.
	SYSPLLCLK	Yes		
TCUCLK<n>	REFCLK	No	Yes	System MMU TCU clock (where n is number of TCUs present in a system). There are two TCUCLK clocks in RD-V1, TCUCLK0, and TCUCLK1.
	SYSPLLCLK	Yes		
SYSPERCLK	REFCLK	No	Yes	System AXI clock.
	SYSPLLCLK	Yes		
SYSPDBGCLK	REFCLK	No	Yes	System PDBG clock.
	SYSPLLCLK	Yes		
ATCLK (ATCLKDBG / DBGCLK)	REFCLK	No	No	ATB trace clock / DEBUG clock.
	SYSPLLCLK	Yes		
PCLKSCP	REFCLK	No	No	System PIK interface clock.
	SYSPLLCLK	Yes		

Table 4-6 MSCP internal clocks generated by the CGU

Clock Signal	Source clock	Divider	Gated	Comments
SCPCORECLK	REFCLK	No	-	SCP core clock.
	SYSPLLCLK	Yes	Yes	
SCPAXICLK	REFCLK	No	-	SCP AXI clock.
	SYSPLLCLK	Yes	Yes	
MCPCORECLK	REFCLK	No	-	MCP core clock.
	SYSPLLCLK	Yes	Yes	
MCPAXICLK	REFCLK	No	-	MCP AXI clock.
	SYSPLLCLK	Yes	Yes	
SYSPIK_PPUCLK	REFCLK	No	-	System PIK PPU clock.
	SYSPLLCLK	Yes	Yes	

Table 4-7 Processor element internal clocks generated by the CGU

Clock Signal	Source clock	Divider	Gated	Comments
CORECLK	REFCLK	No	-	Processor core clock (for each core).
	CPULLCLK	Yes	Yes	
SCLK	INTCLK	Yes	Yes	Direct connect DSU Clock.
PCLK	INTCLK	Yes	Yes	Debug I/F Clock in processor cluster.

Clock Signal	Source clock	Divider	Gated	Comments
ATCLK	INTCLK	Yes	Yes	Internal processor Trace clock.
GICCLK	INTCLK	Yes	Yes	Internal processor GIC clock.

The Processor element supports the following clocking options:

- SYSPLLCLK clock as input to generate the clock for PPU.
- Support for multiple PLLs (PLL<0> to PLL<n>) to generate up to the number of cores used. These PLLs are used to generate core clocks.

CCIX clocking support:

- From the SoC perspective, the CCIX LA can be in the CCIX transport domain.
- For reference, the CMN-650 mesh, including CCIX logics, is running on the INTCLK clock domain.

GIC-700 clock domains:

- The distributor GIC-D is in the main GICCLK clock domain.
- The ITS is in the same clock domain as the PCIe master interface clock domain (External Master Clock, EMCLK<n>).
- The PPIs are in the respective cluster GIC clock domain.

MMU-600 clock domains:

- Each TCU is in the corresponding TCUCCLK<n> clock domain.
- The TBUs are in the same clock domain as the respective PCIe master interface clock domain (External Master Clock, EMCLK<n>).

4.3 Counters

RD-V1 has counters for REFCLK and CoreSight timestamp.

4.3.1 REFCLK counter

The REFCLK counter is an implementation of the memory-mapped counter module that is defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*. The counter is visible only in the AP memory map (REFCLK CNTControl and REFCLK CNTRead). The SCP must access these locations through the Interconnect. The counter is implemented in the VSYS.AONTOP power domain. Access to the CNTControl frame is Secure.

This counter has a single frequency mode, as shown in the following table.

Table 4-8 REFCLK counter frequency modes

Mode	Name
0	REFCLK frequency mode

The frequency mode table entry for the REFCLK frequency mode is writable. This entry is intended to be initialized by firmware during the boot sequence.

This counter can be halted during debug using the cross-trigger network.

See the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*.

4.3.2 CoreSight timestamp counter

The CoreSight timestamp counter is an implementation of the memory-mapped counter module that is defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*. The counter is visible in the SCP memory map (labeled CS CNTControl). The CNTRead counter frame that is defined by the counter architecture is not made available for this counter.

This counter has the same features as the REFCLK counter. However, there is no mechanism to stop this counter using debug.

4.4 Timers

This section lists the time domains existing in RD-V1.

4.4.1 Time domains

RD-V1 contains the following separate time domains:

- REFCLK time is the view of time observed by the AP cores.
- CoreSight timestamp time is a view of time that is used exclusively for the generation of CoreSight timestamps.

4.4.1.1 REFCLK time domain

The AP cores operate in a time domain referred to as REFCLK time. This time domain is based on the main reference clock, REFCLK.

A custom counter component, referred to as the REFCLK counter, generates a time value for the REFCLK time domain. The component meets the requirements of the memory-mapped counter module that is described by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*.

This time domain can be halted during debug.

The REFCLK time domain contains several timers:

- All application cores in RD-V1 implement the Arm Generic Timer, which is defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*. The interrupts from these timers are mapped to the PPI in the GIC-700. Accesses to these Generic Timers are through a low-latency CP15 register interface.
- SCP_REFCLK Generic Timer, see *SCP_REFCLK Generic Timer*, for private use by the SCP.

- MCP_REFCLK Generic Timer, see *MCP_REFCLK Generic Timer*, for private use by the MCP.
- Pn_REFCLK Per cluster Generic Timers for n = 0 to Number of clusters or cores (in RD-V1, the number of clusters is equal to the number of cores), see *Pn_REFCLK per cluster Generic Timer*, for Secure use by the APs and the SCP. These timers are used for waking up the cores when SYSTOP is powered off.
- Two more REFCLK Generic Timers, one Secure and one Non-secure for use solely by the APs for general purposes.

4.4.2 Generic Timers

This section contains the generic timers available in RD-V1.

4.4.2.1 SCP_REFCLK Generic Timer

The SCP element includes a private memory-mapped Arm Generic Timer, which is defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*. This timer provides a single timer frame, without a second view, and without a virtual timer capability. This timer is called the SCP REFCLK Generic Timer, REFCLK CNTCTL, and REFCLK CNTBase0 in the SCP memory map, and is in the VSYS.AONTOP power domain.

4.4.2.2 MCP_REFCLK Generic Timer

The MCP element includes a private memory-mapped Arm Generic Timer, which is defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*. This timer provides a single timer frame, without a second view, and without a virtual timer capability. This timer is called the MCP REFCLK Generic Timer, REFCLK CNTCTL, and REFCLK CNTBase0 in the MCP memory map, and is in the VSYS.AONTOP power domain.

4.4.2.3 Pn_REFCLK per cluster Generic Timer

The SCP Subsystem includes 'n' memory-mapped Arm Generic Timers, which are defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*, where n = number of clusters in the subsystem. Each timer provides a single timer frame, without a second view and without a virtual timer capability. These timers are used for waking up the respective cluster from power down mode. These timers are called Pn_REFCLK_GENTIM per Processor Generic Timers (for n = 0 to 1). There is a single Pn_GENTIMCTRL for every eight Pn_REFCLK_GENTIM, and Pn_REFCLK_GENTIM for n = 0 to 1 in the SCP, MCP, and AP memory maps. These timers are in the VSYS.AONTOP power domain and access to these timer frames is Secure.

4.4.2.4 AP_REFCLK Generic Timer

RD-V1 includes two memory-mapped Arm Generic Timers, which are defined by the *Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile*, for general-purpose functions. Each timer provides single frames, without a second view, and without a virtual timer capability. These timers are called the AP_REFCLK Generic Timer (AP_REFCLK CNTCTL, AP_REFCLK_S CNTBase1, and AP_REFCLK_NS CNTBase0 in the AP memory map) and are in the VSYS.SYSTOP

power domain. Access to the CNTBase0 frame is Non-secure, while the CNTCTL and CNTBase1 frames are Secure.

4.4.3 Watchdog timers

This section contains the watchdog timers available in RD-V1.

4.4.3.1 Generic Watchdog

The *Arm® Server Base System Architecture, version 6.0* defines and requires a Generic Watchdog for use by EL2 software.

This watchdog generates two interrupts. The first interrupt is configured as an EL2 interrupt and is routed as an SPI. It is also sent to SCP to support logging.

The second interrupt is routed as an SPI and can be configured as an EL2 or EL3 interrupt. It is also sent to SCP to support logging. As the APs in RD-V1 implement EL3, the Warm reset of all cores to EL2 on the second watchdog expiry can be supported by dropping down to EL2 after the Warm reset. This is a requirement of the *Arm® Server Base System Architecture, version 6.0*.

The Generic Watchdog is managed by two memory-mapped register frames. In the subsystem, these frames are accessible by Secure and Non-secure accesses from the APs.

This watchdog timer is clocked by REFCLK.

4.4.3.2 SCP Watchdog

The SCP subsystem includes a Cortex-M System Design Kit Watchdog Timer that protects against lockups in the firmware. This watchdog timer is clocked by REFCLK.

4.4.3.3 MCP Watchdog

The MCP subsystem includes a Cortex-M System Design Kit Watchdog Timer that protects against lockups in the firmware. This watchdog timer is clocked by REFCLK.

4.4.3.4 Secure Watchdog

A Secure Watchdog Timer is available to the APs when operating in Secure mode.

The first time the Secure Watchdog expires, interrupts to the GIC-700 (routed as SPI interrupt at EL3) and SCP (for logging) are generated. If the software fails to clear the watchdog and it expires for a second time, an interrupt is sent to the SCP. The SCP takes appropriate action, which is, at a minimum, a Warm reset of all cores.



If a global Subsystem reset is the response the SCP takes on second expiration, the Trusted Watchdog state is not preserved through reset (not a requirement of the *Arm® Server Base System Architecture, version 6.0*).

The Secure Watchdog Timer is clocked by REFCLK.

4.4.3.5 Watchdog security

Both SCP Watchdog and Secure Watchdog Timers are accessible by Secure accesses only. These watchdogs also support halt on debug functionality, enabling cross triggers to halt the watchdog.

4.5 Power down considerations

A core can generate timer interrupts after entering WFI or WFE mode. Cores can enter WFI and WFE modes without any side-effects relating to their timers.

When a cluster is powered down, Generic Timer state is lost, so extra steps must be taken.

There are two models for powering down a core, referred to here as AP Wakeup and SCP Wakeup.

4.5.1 AP Wakeup

In this model, software running on the APs must ensure that no timers are active on the core that is to be powered down. The core saves the state of its Generic Timer to the respective Pn_REFCLK timer in the SCP and execute WFI. Once the core is in WFI, it is powered down by the hardware. Any AP interrupts that are mapped to this core wake up the core.

4.5.2 SCP Wakeup

This model applies when powering down the last AP.

The timer state must be saved to the AP_REFCLK (legacy timer) or Pn_REFCLK (for n = 0 to Number of clusters) Generic Timers as part of the power down sequence. Interrupts are masked during this sequence. When the AP core is in WFI mode, power is removed by power gating. The AP_REFCLK or Pn_REFCLK (for n = 0 to Number of clusters) Generic Timers cause the processor to be woken when the timer expires.

The sequence for transition into CSS.SLEEP1 state is as follows.

- Before the SCP powers down VSYS.SYSTOP, it must:
 - Put the memory controllers in LP mode.
 - Save the state of the GIC and memory controller configuration state, before VSYS.SYSTOP is powered down.
 - If the AP_REFCLK Generic Timers are intended to be used for system wakeup, save the state of the AP_REFCLK Generic Timers before powering down VSYS.SYSTOP. Use the state from the AP_REFCLK Generic Timers to schedule the wakeup of VSYS.SYSTOP. The value that is programmed into the AP_REFCLK timer accounts for the wakeup time of the SYSTOP and restoration of the system state.
 - Enable the interrupts to SCP core from the Pn_REFCLK Generic Timers if these timers are intended to be used for system wakeup. Use the state from the Pn_REFCLK Generic Timers

to schedule the wakeup of VSYS.SYSTOP. The value that is programmed into the Pn_REFCLK timers accounts for the wakeup time of SYSTOP and restoration of the system state.

- Restore state to the GIC and memory controllers state.
- After waking VSYS.SYSTOP, the SCP restores system state, triggers an interrupt, and the AP_REFCLK Generic Timers or interrupts from Pn_REFCLK wake up the corresponding AP cores.

5 Fixed Virtual Platform

To develop ahead of hardware availability and to explore the design from a software perspective, the Fixed Virtual Platform (FVP) models many of the Arm IP in the RD-V1 design.

5.1 About the FVP

Two configurations of RD-V1 are supported:

- RD-V1 FVP models Config-M, a single-chiplet system with 16 Arm® Neoverse™ V1 cores.
- RD-V1 quad-chiplet FVP models a reduced-size variant of Config-XL, consisting of four compute subsystems linked by CMN-650 CML. It provides a functional model of a quad-chiplet system. Each subsystem contains four Arm® Neoverse™ V1 cores, for a total of 16 cores in the FVP (at full size, Config-XL has 4 x 32 cores).

The FVP models the following IP components:

- Arm® Neoverse™ V1 MP1
- GIC-700
- MMU-600
- SCP
- MCP
- CMN-650
- Multiple NIC-450 interconnects
- Memory access path towards DRAM which includes a TrustZone controller.



The FVP does not model every component that RD-V1 describes. For example, it does not model the CoreSight technology components.

The RD-V1 FVP drives system architecture and software standardization. The models provide software and binaries of proprietary firmware that reduce the amount of work that is required for SoC development.

5.2 FVP peripherals

The RD-V1 FVP includes peripherals that the software payload requires to run.

These peripherals are organized in two layers:

- SoC - The SoC peripherals represent peripherals that can be added to a compute subsystem in a SoC design. The Arm® Neoverse™ V1 reference design SoC model is based on the Juno Arm Development Platform (ADP).
- Board - The board peripherals represent peripherals that can be present on the board onto which the SoC is mounted. The RD-V1 board model is based on the Juno Arm Development Platform (ADP).

In the quad-chiplet system, each compute subsystem has SoC and Board layers dedicated to that subsystem. Addressing for each chip is defined in *7.1 AP memory map*.

- Chip 0: 0x000_0000_0000 – 0x3FF_FFFF_FFFF
- Chip 1: 0x400_0000_0000 – 0x7FF_FFFF_FFFF
- Chip 2: 0x800_0000_0000 – 0xBFF_FFFF_FFFF
- Chip 3: 0xC00_0000_0000 – 0xFFF_FFFF_FFFF

A sideband communication channel is required to coordinate multi-chiplet software boot over CMN-650. The FVP implements this using the MHU device, but Arm recommends using a solution such as I²C in hardware.

Table 5-1 SoC peripherals

Name	Base address	Size	Description
SMC interface	0x00_0800_0000	368MB	Routed to Board
SMMUv3	0x00_2B40_0000	1MB	-
PCIe Config	0x00_6000_0000	16MB	-
PCIe Memory	0x00_7000_0000	132MB	-
DMA MMU-400	0x00_7FB0_0000	64KB	-
HDLCD1 MMU-400	0x00_7FB1_0000	64KB	-
HDLCD0 MMU-400	0x00_7FB2_0000	64KB	-
SoC Interconnect NIC-400 GPV	0x00_7FD0_0000	1MB	-
Surge detector	0x00_7FE5_0000	4KB	Dummy APB
TRNG	0x00_7FE6_0000	4KB	-
Trusted Non-volatile counters	0x00_7FE7_0000	4KB	-
Trusted Root-Key storage	0x00_7FE8_0000	4KB	-
Secure I2C	0x00_7FE9_0000	256B	A Secure I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
DDR3 PHY 3	0x00_7FB9_0000	64KB	Dummy APB
DDR3 PHY 2	0x00_7FB8_0000	64KB	Dummy APB

Name	Base address	Size	Description
DDR3 PHY 1	0x00_7FB7_0000	64KB	Dummy APB
DDR3 PHY 0	0x00_7FB6_0000	64KB	Dummy APB
DMA Non-secure	0x00_7FF0_0000	4KB	-
DMA Secure	0x00_7FF1_0000	4KB	-
HDLCD1	0x00_7FF5_0000	4KB	-
HDLCD0	0x00_7FF6_0000	4KB	-
UART 1	0x00_7FF7_0000	4KB	-
UART 0	0x00_7FF8_0000	4KB	-
I2S	0x00_7FF9_0000	1KB	An I2S component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
I2C	0x00_7FFA_0000	256B	An I2C component does not exist in the FVP. Instead, a PL061_GPIO is mapped.
PL352	0x00_7FFD_0000	4KB	PL354
System override Registers	0x00_7FFF_0000	4KB	-
AP configuration	0x00_7FFE_0000	4KB	GPR
PCIe Memory	0x05_0000_0000	12GB	-



The following devices are discoverable on the PCIe bus:

- Virtio block device x 2.
- AHCI controller with attached SATA disk.

Table 5-2 Board peripherals

Name	Base address	Size	Description
NOR Flash 0	0x00_0800_0000	64MB	-
NOR Flash 1	0x00_0C00_0000	64MB	-
NOR Flash 2	0x00_1000_0000	64MB	-
Ethernet	0x00_1800_0000	64MB	SMSC 91C111
System registers	0x00_1C01_0000	64KB	-
MCI	0x00_1C05_0000	64KB	PL180
KMI 0	0x00_1C06_0000	64KB	PL050
KMI 1	0x00_1C07_0000	64KB	PL050
UART 0	0x00_1C09_0000	64KB	PL011
UART 1	0x00_1C0A_0000	64KB	PL011
Watchdog	0x00_1C0F_0000	64KB	SP805

Name	Base address	Size	Description
Dual Timer	0x00_1C11_0000	64KB	SP804
Virtio Block Device	0x00_1C13_0000	64KB	-
Virtio Net Device	0x00_1C15_0000	64KB	-
RTC	0x00_1C17_0000	64KB	PL031
GPIO 0	0x00_1C1D_0000	64KB	PL061_GPIO
GPIO 1	0x00_1C1E_0000	64KB	PL061_GPIO
DRAM	0x00_8000_0000	2GB	-
DRAM	0x80_8000_0000	6GB	-



The SoC peripherals area and board peripherals area in the RD-V1 memory map are mapped to an Expansion AXI region. Therefore, these mappings are not defined in the reference design.

Table 5-3 Interrupt map at the SoC layer

Interrupt ID	Source	Description
147	UART 0	-
148	UART 1	-
171	TRNG	-

Table 5-4 Interrupt map at the board layer

Interrupt ID	Source	Description
111	Ethernet	-
132	RTC	-
133	UART 0	-
134	UART 1	-
140	VFS2	-
202	Virtio	-
204	Virtio net device	-
228	Watchdog	-
229	KMI O	-
230	Dual Timer	Interrupts 0 and 1
231	System registers Ethernet IRQ	-

6 Software

The integrated software stack provides a starting point to modify, extend, and develop the software stack for a SoC based on the RD-V1 design.

6.1 About the software

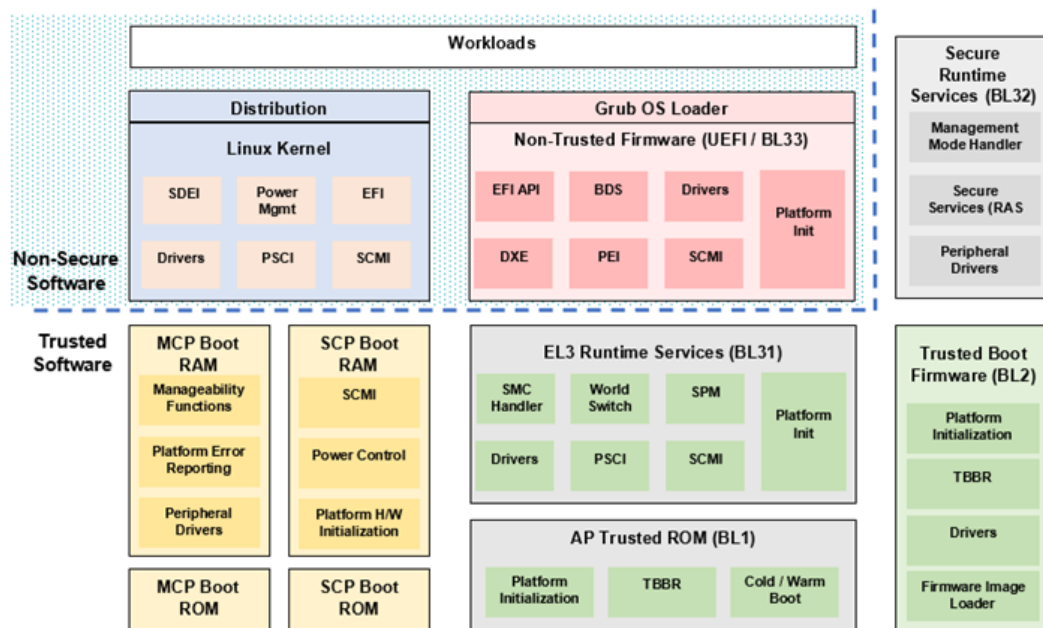
The Board Support Package (BSP) software contains firmware and kernel components that can be run on the associated FVP.

The BSP consists of the following:

- SCP firmware – Controls the system and manages the power.
- MCP firmware – Performs minimal MCP element initialization.
- AP firmware – Comprising the Arm Trusted Firmware and Unified Extensible Firmware Interface (UEFI), it loads the OS and provides platform services to the OS. The supported OS is a variant of Linux.
- Linux kernel – Contains the following kernel features, which are required to exploit and demonstrate RD-V1 hardware features:
 - Linux device drivers for all the I/O components.
 - Support for virtio disk.

Figure 6-1 shows a high-level overview of the software components and their features.

Figure 6-1 Software overview



6.2 SCP firmware

The SCP is a compute unit running in the always-ON power domain of RD-V1 and is responsible for low-level system management. The SCP is a Cortex-M7 processor with a set of dedicated peripherals and interfaces that you can extend.

The SCP firmware supports:

- Powerup sequence and system startup
- Initial hardware configuration
- Clock and regulator management
- Servicing power state requests from the OS Power Management (OSPM) software
- Multi-chiplet operation

The SCP firmware manages the overall power, clock, reset, and system control of RD-V1. The SCP firmware is an inherently trusted part of the software. All the memory that it uses for execution and private storage is internal RAM, to prevent tampering.

6.2.1 Power control

This block is the generic interface to the SCP from the AP. It performs the following functions:

- Inquires about the capabilities of the system and individual devices.
- Obtains or sets the state of the whole RD-V1 subsystem and individual devices under SCP control.
- Obtains or sets the performance level of the processors.

The SCP provides a well-defined API for extending the standard commands in the power control interface.

6.2.2 SCP boot ROM

The SCP boot ROM code executes after RD-V1 exits from a Cold reset. The SCP boot ROM configures the initial state of the hardware. For example, it can define:

- The processor cores that are released from reset.
- The clocks that are available.
- The state of the Power Management Integrated Circuit (PMIC).

6.3 MCP firmware

The MCP is a Cortex-M7 processor compute unit running in the AON power domain of RD-V1. The MCP firmware is intended to control all the manageability functions and RAS features. The MCP firmware can also be extended to communicate with an external BMC to provide event logging and communication of alerts.

6.3.1 MCP boot ROM

The MCP boot ROM code executes after RD-V1 exits from a Cold reset.

The MCP boot ROM configures the initial state of the hardware. For example, it can:

- Control the MCP subsystem clocks.
- Establish communication with the SCP.

6.4 AP firmware

The AP firmware consists of the code that is required to boot RD-V1 up to the point where the OS execution starts.

The firmware contains the code that is required to:

- Set up the initial security environment.
- Support runtime processor power state control using the Power State Coordination Interface (PSCI).
- Load Linux from boot media.

6.4.1 Arm Trusted firmware BL1

The AP contains an on-chip trusted ROM that executes the boot code on an RD-V1 SoC. The AP Boot ROM is fixed for the lifetime of the device and executes minimal code to:

- Maximize robustness.
- Reduce the risk of security vulnerabilities.

6.4.2 Arm Trusted firmware BL2

Arm AP trusted firmware BL2 executes in on-chip Trusted RAM, where content is loaded from non-volatile storage. Subsequent AP firmware images are stored in DRAM after the AP is initialized.

6.4.2.1 TrustZone initialization, TZ Init

The AP trusted RAM firmware initializes any Trusted world resources that the AP Trusted boot ROM either does not initialize, or any Trusted world resources that require more initialization, for example the memory controller.

6.4.3 Arm Trusted firmware BL31

The following constitute Arm Trusted firmware BL31.

6.4.3.1 Power State Coordination Interface (PSCI)

The AP Trusted RAM firmware defines a Secure Monitor Call (SMC) interface to support Rich OS Power Management in accordance with the Power State Coordination Interface (PSCI) System Software on Arm® Systems. Linux CPU idle framework uses PSCI to power up or down the AP cores.

6.4.3.2 Secure Monitor framework

The Secure Monitor framework handles all SMCs in the AP trusted RAM firmware. The framework handles the transition to Trusted world execution and distributes the SMCs to the correct SMC handler. The following might own these SMC handlers:

- Arm
- Silicon partner (SiP)
- Original Design Manufacturer (ODM)
- Original Equipment Manufacturer (OEM)

6.4.3.3 Secure partition manager

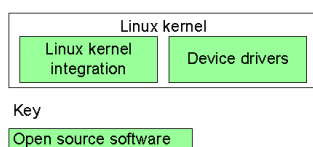
This framework can be extended to pass the RAS event that is generated in the Secure world to the Normal world.

6.5 Linux kernel

The RD-V1 Linux kernel contains the subsystem-specific features that demonstrate the capabilities of RD-V1.

Figure 6-5 shows the structure of the Linux kernel.

Figure 6-2 Linux kernel



The RD-V1 Linux kernel contains the following features:

6.5.1 Multi-Processing (MP)

Linux can view all Arm® Neoverse™ V1 processors at the same time, which enables the task scheduler to make the best use of them.

In addition to the MP mechanics, the Linux kernel contains optimizations to permit efficient operation on RD-V1. The optimizations include tuning, to ensure that threads are scheduled efficiently across all AP cores. These optimizations are validated against multiple use cases.

6.5.2 UEFI awareness

The kernel contains the functionality that is required to configure the system using UEFI, ACPI, and SMBIOS tables that the underlying firmware provides.

6.5.3 Device drivers

The RD-V1 Linux kernel contains the following device drivers:

- Generic Timer
- UART
- GIC (GICv3)
- Virtio network controller
- PCIe-RC and AHCI

See the *Arm® Neoverse™ V1 reference design Release Note* for instructions on how to set up and run the software stack and User Guide.

6.6 Multi-chiplet support

SCP firmware supports configuring the platform for multi-chiplet operation. SCP configures CMN-650 to set up a coherent link with the other chips on the platform. SCP also uses a low-speed sideband interface to communicate inter-chip control messages such as power on and off, and performance control. Using this coherent link across the chips, Linux kernel boots the Arm® Neoverse™ V1 cores on all of the chips.

7 Memory map

This chapter briefly describes the AP, SCP, and MCP memory and interrupt maps. The SCP and MCP have their own memory maps that differ from the AP memory map.

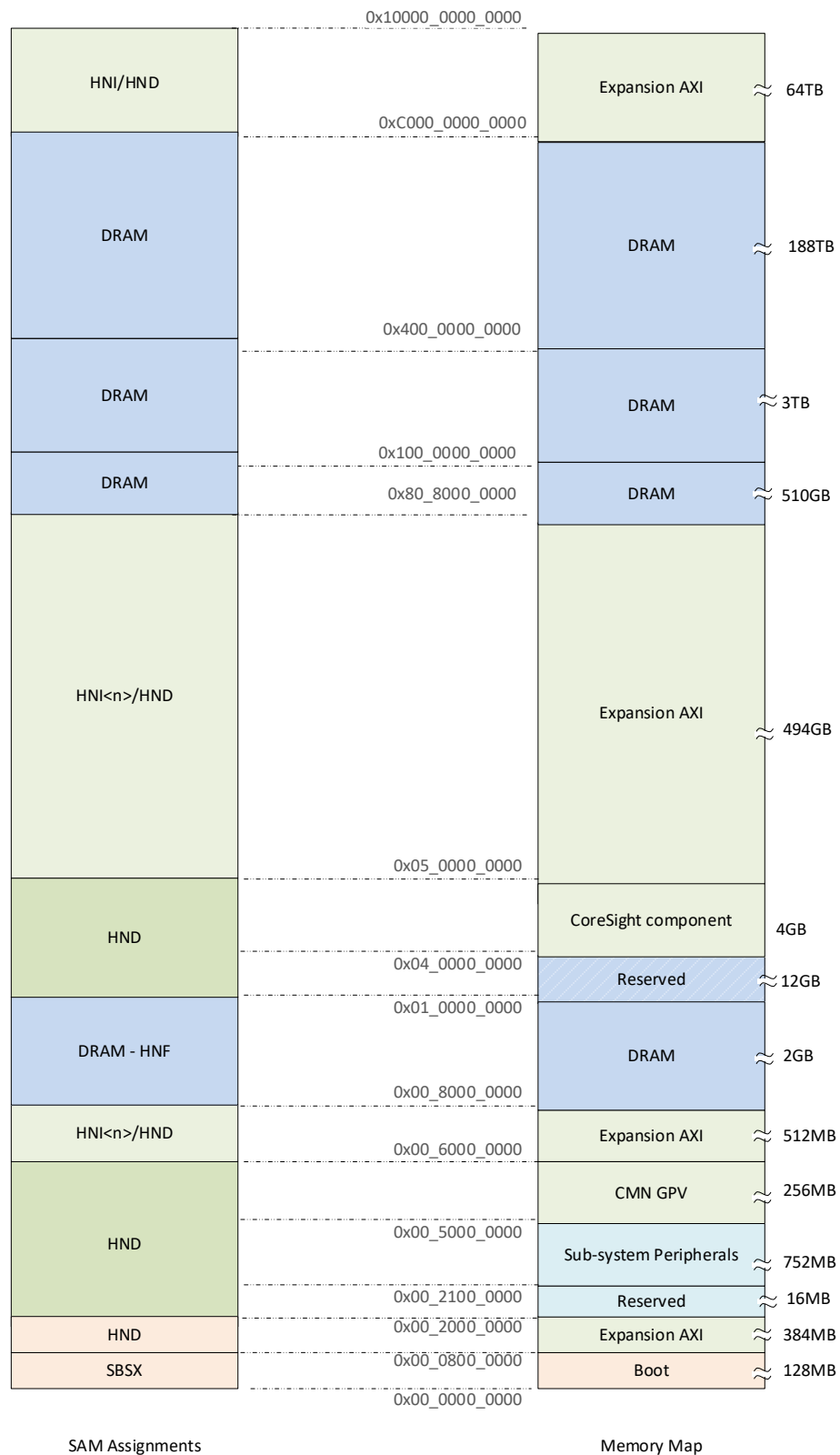
7.1 AP memory map

The AP memory map is visible to the following:

- APs
- SCP
- MCP
- Embedded Trace Router
- Coherent Expansion AXI slave interfaces
- AP DAP

Figure 7-1 on page 59 shows a top-level representation of the AP memory map.

Figure 7-1 AP memory map





The memory map does not adhere to the *Principles of ARM® Memory Maps White Paper* standard.

Unless explicitly stated otherwise:

- Where a region is used to map a peripheral or device, it is possible that the peripheral or device occupies less than the region size used. For example, a peripheral might only occupy 4KB out of the 64KB of the region that is reserved for it. Access to an unmapped region causes a Decode Error (DECERR) response.
- Accesses to reserved areas within the memory map also results in a DECERR response.
- When accessing areas occupied by peripherals or devices, it is the peripherals or devices themselves that determine the response to return. These accesses can include unmapped or reserved area within the areas occupied by the peripheral or device.
- The MCP or SCP might require access to external peripherals that reside outside the subsystem. If such an access is necessary, those peripherals must be mapped to the lower 2GB of address space.

The following sections describe in more detail the RD-V1 Boot area, Peripheral areas, and how the memory controller area maps to DRAM memory using the memory controller.

The memory map in this and the following sections also shows the overall security attributes associated to each area of memory. These are split into the following groups:

- Always Secure Access: A component or region that is only accessible to Secure transactions. Any Non-secure access targeting these components causes a DECERR response.
- Secure and Non-secure Access: A component or region that is accessible to both Secure and Non-secure transactions.
- Programmable Access security (also referred to as “securable”): Components or regions that are defined to be independently software configurable can be changed between the above two states by trusted software. These attributes can be configured in the network interconnect, NIC-450 or in the component itself, and the default state is Secure access only from reset.
- User Defined: These attributes are areas which are mapped to expansion interfaces and the components outside of the RD-V1 subsystem defines their access security. These components must use the **ARPROT[1]** or **AWPROT[1]** bits provided on the expansion interfaces to determine the security permission of each access. Any accesses that fail any external security checks must result in a DECERR response.

For a multichip case, each chip is assigned an equal amount of memory space size of 4TB. Each chip has the same memory map with a different base address. The `SID_CHIP_ID` determines the base address.

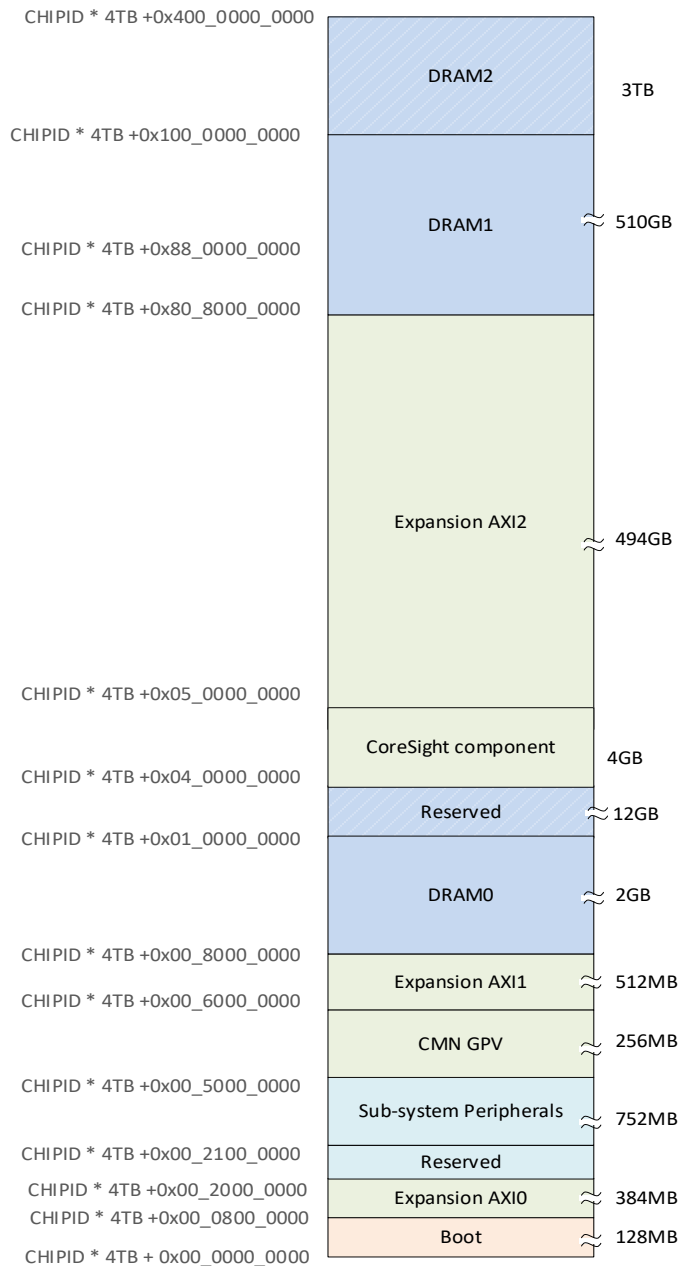
The application memory base address of each chip is shown in the following calculation:

$$AP_BASE_ADDRESS = SID_CHIP_ID * 4TB + 0x000_0000_0000$$

For example, for quad-chiplet design, each chip is assigned the following regions:

- Chip 0: 0x000_0000_0000 - 0x3FF_FFFF_FFFF
- Chip 1: 0x400_0000_0000 - 0x7FF_FFFF_FFFF
- Chip 2: 0x800_0000_0000 - 0xBFF_FFFF_FFFF
- Chip 3: 0xC00_0000_0000 - 0xFFF_FFFF_FFFF

Figure 7-2 AP memory map for a multi-chiplet use case



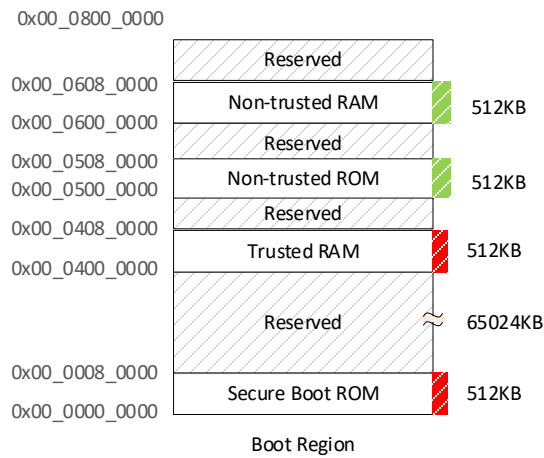
7.1.1 Boot region

As shown in Figure 7-3, the first 128MB of the address map is the boot region maximum allocated sizes for Secure Boot ROM and RAM. 512KB of the Secure Boot ROM and RAM are accessed only by Secure accesses. The actual ROM and RAM sizes can be configurable and implementation dependent on the use case.

For the ROMs, the hardware decodes the whole 512KB ROM space but ignores the writes and returns zeros for the read data for read accesses outside the configured ROM size address region. OK responses are always returned.

For the RAMs, any read or write accesses outside the configured RAM size causes a DECERR.

Figure 7-3 RD-V1 boot area memory map



7.1.1.1 Non-secure SRAM

The Non-secure SRAM is defined to be 512KB scratch RAM for use by Non-secure AP software. It accepts both Secure and Non-secure accesses. The actual RAM size is implementation-specific.

7.1.1.2 Non-trusted ROM

This Non-trusted ROM area is expected to contain code that is required during firmware update. This area is accessible in both Secure and Non-secure mode. 512KB region sets the maximum size of the Non-trusted ROM. The actual ROM size is implementation-specific.

7.1.2 Peripherals region

As shown in Table 7-1, a region starting from address 0x00_2100_0000 is defined as the subsystem peripherals memory area. This region defines the memory map for peripherals and memories that are part of RD-V1.



Note

In Table 7-1, the Peripherals region address map is shown with configurable option variables such as number of clusters (NUM_CLUSTERS), number of cores (NUM_CORES), and number of TCUs and TBUs. For RD-V1, the configuration values are to be applied accordingly to derive the absolute physical addresses.

Table 7-1 Peripheral area memory map overview

Start	End	State	Peripheral
0x00_2100_0000	0x00_29FF_FFFF	Secure	Memory Region allocated for 8 DMC controllers. See Memory Controller region.
0x00_2A00_0000	0x00_2A0F_FFFF	Non-secure	Reserved
0x00_2A10_0000	0x00_2A1F_FFFF	Secure	Interconnect Coherent NIC GPV
0x00_2A20_0000	0x00_2A2F_FFFF	Non-secure	Reserved
0x00_2A30_0000	0x00_2A3F_FFFF	Secure	Base NIC GPV
0x00_2A40_0000	0x00_2A40_0FFF	Non-secure	UART0
0x00_2A40_1000	0x00_2A41_FFFF	Non-secure	UART1
0x00_2A42_0000	0x00_2A42_FFFF	Secure	Reserved
0x00_2A43_0000	0x00_2A43_FFFF	Secure	REFCLK CNTControl
0x00_2A44_0000	0x00_2A44_FFFF	Non-secure	Generic Wdog Control
0x00_2A45_0000	0x00_2A45_FFFF	Non-secure	Generic Wdog Refresh
0x00_2A46_0000	0x00_2A47_FFFF	Non-secure	Reserved
0x00_2A48_0000	0x00_2A48_FFFF	Secure	Trusted Watchdog Control
0x00_2A49_0000	0x00_2A49_FFFF	Secure	Trusted Watchdog Refresh
0x00_2A4A_0000	0x00_2A4A_FFFF	Non-secure	SID Registers
0x00_2A4A_0000	0x00_2A7F_FFFF	Non-secure	Reserved
0x00_2A80_0000	0x00_2A80_FFFF	Non-secure	REFCLK CNTRead
0x00_2A81_0000	0x00_2A81_FFFF	Non-secure	AP_REFCLK_CNTCTL
0x00_2A82_0000	0x00_2A82_FFFF	Secure	AP_REFCLK_S_CNTBase0
0x00_2A83_0000	0x00_2A83_FFFF	Non-secure	AP_REFCLK_NS_CNTBase1
0x00_2A84_0000	0x00_2AFF_FFFF	Non-secure	Reserved
0x00_2B00_0000	0x00_2B0F_FFFF	Non-secure	Reserved for SCP/MCP to access DDR space using 1MB window address translation.
0x00_2B10_0000	0x00_2EFF_FFFF	Non-secure	Reserved
0x00_2F00_0000	0x00_2FFF_FFFF	Non-secure	Reserved (reserved for future GIC)
0x00_3000_0000	0x00_31FF_FFFF	Non-secure	GIC
0x00_3200_0000	0x00_42FF_FFFF	Non-secure	Reserved
0x00_4410_0000	0x00_44FF_FFFF	Secure	Generic Pn Wakeup timers*(NUM_CLUSTERS + NUM_CLUSTERS/8) (0x1_0000 each)
0x00_4500_0000	0x00_453F_FFFF	Non-secure	AP_SCP_MHUV2 Non-secure
0x00_4540_0000	0x00_457F_FFFF	Secure	AP_SCP_MHUV2 Secure
0x00_4600_0000	0x00_46FF_FFFF	-	Reserved
0x00_4700_0000	0x00_4C3F_FFFF	Non-secure	Reserved
0x00_4C40_0000	0x00_4C40_FFFF	Non-secure	AP2MCP MHU / MHUV2 AP_MCP_SEND_NS

Start	End	State	Peripheral
0x00_4C41_0000	0x00_4C41_FFFF	Non-secure	AP2MCP MHU Non-secure RAM / MHUV2 AP_MCP_RCV_NS
0x00_4C42_0000	0x00_4C42_FFFF	Secure	AP2MCP MHU Secure RAM / MHUV2 AP_MCP_SEND_S
0x00_4C43_0000	0x00_4C43_FFFF	Secure	Reserved / MHUV2 AP_MCP_RCV_S
0x00_4C44_0000	0x00_4CFF_FFFF	Non-secure	Reserved
0x00_4D00_0000	0x00_4DFF_FFFF	Non-secure	BASE STM
0x00_4E00_0000	0x00_4EFF_FFFF	Non-secure	Reserved
0x00_4F00_0000	0x00_4FFF_FFFF	Non-secure	SMMU

Figure 7-4 Peripheral area memory map part 1

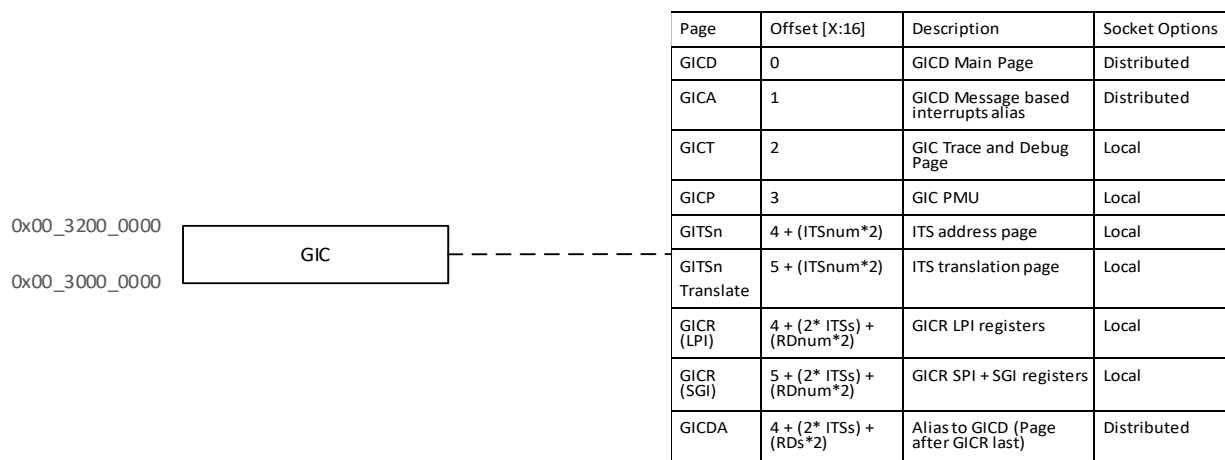
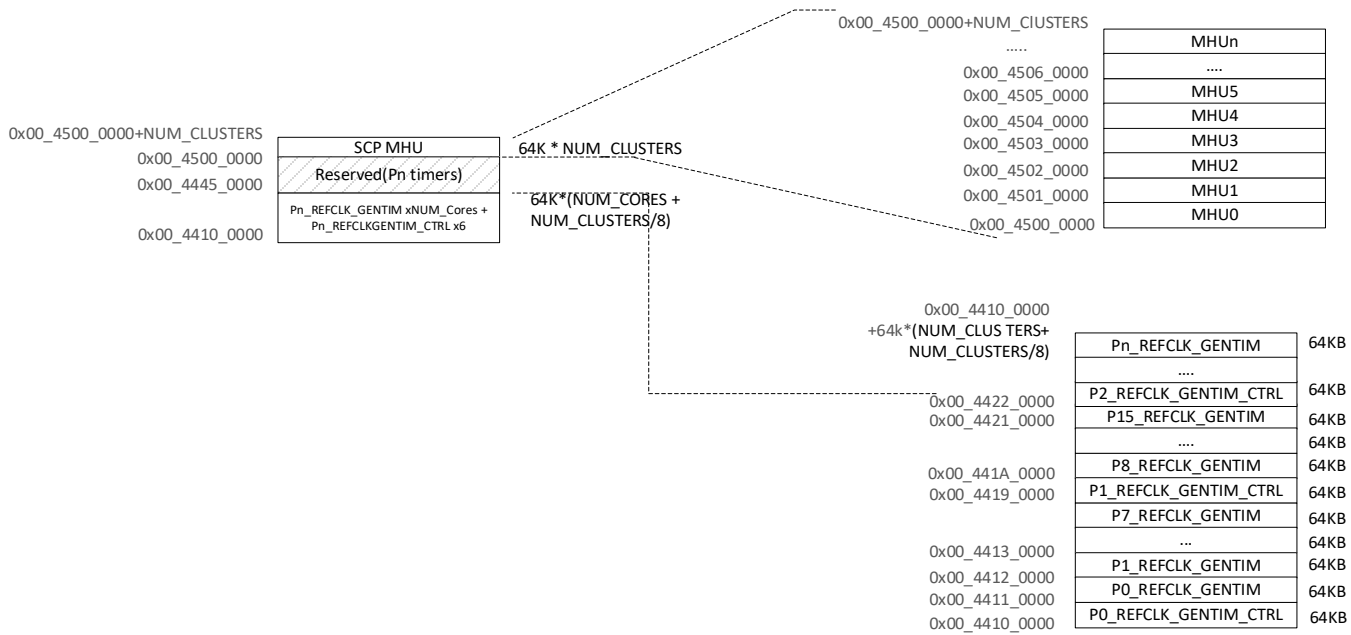
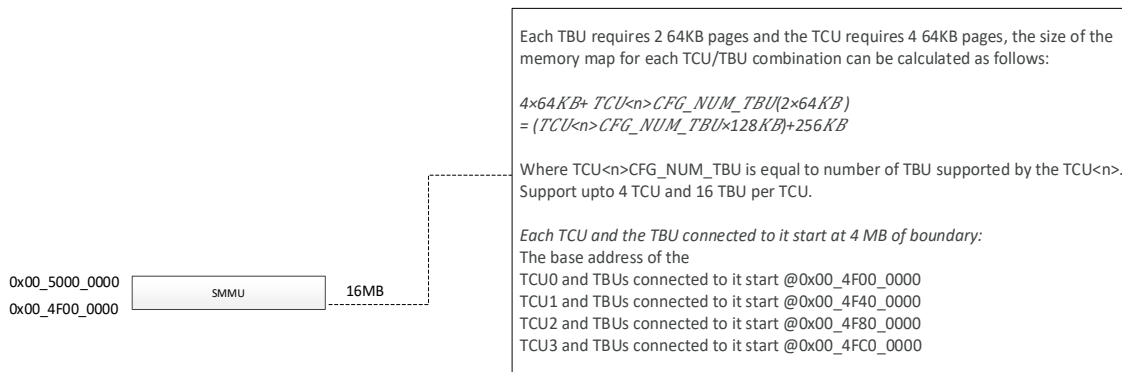


Figure 7-5 Peripheral area memory map part 2



See [3.4.3 Message Handling Unit](#) for guidance to size the number of MHU. See [4.4.2 Generic Timers](#) for guidance on how to size the number of Pn_REFCLK timers.

Figure 7-6 Peripheral area memory map part 3



See [Figure 3-7](#) for the number of TCUs and TBUs used in the RD-V1.

Table 7-2 shows the Memory element address region from 0x2100_0000 to 0x2A00_0000, allocated for all the Memory element-related configurations such as memory controller, TZC-400, Manager, and other functions.

Table 7-2 Peripheral area memory map for Memory element

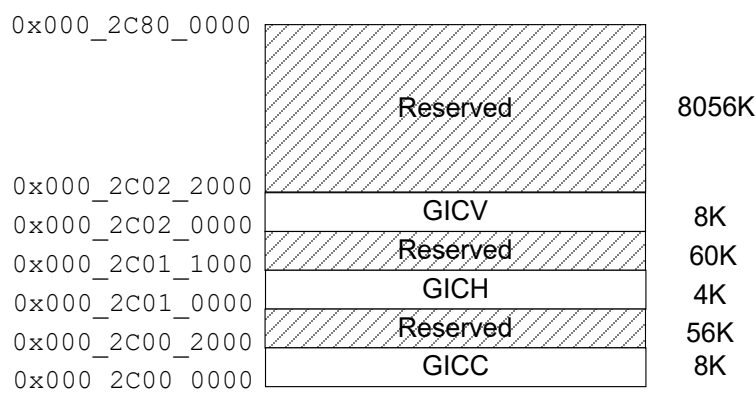
Start	End	Peripheral
0x2100_0000	0x217F_FFFF	Memory Element0 Configuration
0x2180_0000	0x2181_FFFF	Memory Element0 Manager
0x2182_0000	0x2182_FFFF	Memory Element0 Sensor Group
0x2183_0000	0x2183_FFFF	Memory Element0 TZC
0x2184_0000	0x21FF_FFFF	Reserved
0x2200_0000	0x227F_FFFF	Memory Element1 Configuration
0x2280_0000	0x2281_FFFF	Memory Element1 Manager
0x2282_0000	0x2282_FFFF	Memory Element1 Sensor Group
0x2283_0000	0x2283_FFFF	Memory Element1 TZC
0x2284_0000	0x22FF_FFFF	Reserved
0x2300_0000	0x237F_FFFF	Memory Element2 Configuration
0x2380_0000	0x2381_FFFF	Memory Element2 Manager
0x2382_0000	0x2382_FFFF	Memory Element2 Sensor Group
0x2383_0000	0x2383_FFFF	Memory Element2 TZC
0x2384_0000	0x23FF_FFFF	Reserved
0x2400_0000	0x247F_FFFF	Memory Element3 Configuration
0x2480_0000	0x2481_FFFF	Memory Element3 Manager
0x2482_0000	0x2482_FFFF	Memory Element3 Sensor Group
0x2483_0000	0x2483_FFFF	Memory Element3 TZC
0x2484_0000	0x24FF_FFFF	Reserved
0x2500_0000	0x257F_FFFF	Memory Element4 Configuration
0x2580_0000	0x2581_FFFF	Memory Element4 Manager
0x2582_0000	0x2582_FFFF	Memory Element4 Sensor Group
0x2583_0000	0x2583_FFFF	Memory Element4 TZC
0x2584_0000	0x25FF_FFFF	Reserved
0x2600_0000	0x267F_FFFF	Memory Element5 Configuration
0x2680_0000	0x2681_FFFF	Memory Element5 Manager
0x2682_0000	0x2682_FFFF	Memory Element5 Sensor Group
0x2683_0000	0x2683_FFFF	Memory Element5 TZC
0x2684_0000	0x26FF_FFFF	Reserved
0x2700_0000	0x277F_FFFF	Memory Element6 Configuration
0x2780_0000	0x2781_FFFF	Memory Element6 Manager
0x2782_0000	0x2782_FFFF	Memory Element6 Sensor Group
0x2783_0000	0x2783_FFFF	Memory Element6 TZC
0x2784_0000	0x27FF_FFFF	Reserved

Start	End	Peripheral
0x2800_0000	0x287F_FFFF	Memory Element7 Configuration
0x2880_0000	0x2881_FFFF	Memory Element7 Manager
0x2882_0000	0x2882_FFFF	Memory Element7 Sensor Group
0x2883_0000	0x2883_FFFF	Memory Element7 TZC
0x2884_0000	0x28FF_FFFF	Reserved
0x2900_0000	0x29FF_FFFF	Reserved

7.1.2.1 Processor peripherals region

The processor peripherals region is used for peripherals that are required by the APs. This region is dedicated to the GIC. GIC-700 defines the security of the region internally.

Figure 7-7 Processor peripheral region



7.1.3 CoreSight debug and trace

Table 7-3 shows the SoC-600 debug and trace region of the subsystem memory map with multichain debug support that is used in RD-V1.

Table 7-3 SoC-600 debug and trace address map

Base Address	End Address	Size	Peripheral
<debug base> + 0x00000000	<debug base> + 0x00000FFF	4KB	Main APP Debug ROM
<debug base> + 0x00001000	<debug base> + 0x0000FFFF	60KB	Reserved
<debug base> + 0x00010000	<debug base> + 0x00010FFF	4KB	STM ETF
<debug base> + 0x00011000	<debug base> + 0x0001FFFF	60KB	Reserved

Base Address	End Address	Size	Peripheral
<debug base> + 0x00020000	<debug base> + 0x00020FFF	4KB	System ETF
<debug base> + 0x00021000	<debug base> + 0x0008FFFF	444KB	Reserved
<debug base> + 0x00090000	<debug base> + 0x00090FFF	4KB	If (NUM_TRACE_SLV≥2) then: System Funnel Else: Reserved
<debug base> + 0x00091000	<debug base> + 0x0009FFFF	60KB	Reserved
<debug base> + 0x000A0000	<debug base> + 0x000A0FFF	4KB	Master Funnel
<debug base> + 0x000A1000	<debug base> + 0x000AFFFF	60KB	Reserved
<debug base> + 0x000B0000	<debug base> + 0x000B0FFF	4KB	If (NUM_DEBUG_CHAINS>1) then: DBGCH Funnel Else: Reserved
<debug base> + 0x000B1000	<debug base> + 0x0010FFFF	380KB	Reserved
<debug base> + 0x00110000	<debug base> + 0x00110FFF	4KB	Replicator
<debug base> + 0x00111000	<debug base> + 0x0011FFFF	60KB	Reserved
<debug base> + 0x00120000	<debug base> + 0x00120FFF	4KB	ETR
<debug base> + 0x00121000	<debug base> + 0x0012FFFF	60KB	Reserved
<debug base> + 0x00130000	<debug base> + 0x00130FFF	4KB	TPIU
<debug base> + 0x00131000	<debug base> + 0x0013FFFF	60KB	Reserved
<debug base> + 0x00140000	<debug base> + 0x00140FFF	4KB	CTI
<debug base> + 0x00141000	<debug base> + 0x0015FFFF	124KB	Reserved
<debug base> + 0x00160000	<debug base> + 0x00160FFF	4KB	If (CATU_INCLUDED=1) then: CATU Else: Reserved
<debug base> + 0x00161000	<debug base> + 0x0016FFFF	60KB	Reserved
<debug base> + 0x00170000	<debug base> + 0x00170FFF	4KB	Debug Element STM Debug APB
<debug base> + 0x00171000	<debug base> + 0x003FFFFF	2620KB	Reserved
<debug base> + 0x00400000	<debug base> + 0x01FFFFFF	28MB	If (NUM_DEBUG_APB_MST>0) then: APB Expansion Regions Else: Reserved
<debug base> + 0x02000000	<debug base> + 0x41FFFFFF	1GB	Total number of CPU cores or clusters supported 0-63. Debug APB Size supported per core or cluster is 16MB.

The Debug components within each cluster are specified in Table 7-4.

Table 7-4 Processor cluster debug component address map

Address	Component
0x000000	ROM Table
0x010000	PE0 Debug
0x020000	PE0 CTI
0x030000	PE0 PMU
0x040000	PE0 ETM
0x050000	Reserved
0x060000	Reserved
0x070000	Reserved
0x080000	Reserved
0x090000	Reserved
0x0A0000	Reserved
0x0B0000	Reserved
0x0C0000	PE0 ELA
0x0D0000	Cluster ELA
0x0E0000	Cluster CTI
0x0F0000	PE0 ETF

7.1.4 Expansion Space

In RD-V1, the AP memory map provides the following types of expansion regions to enable peripherals to be added to the base memory map:

- AXI Expansion
- CoreSight APB Expansion

These regions implement “user-defined” security. Components that are integrated in these regions must use the **ARPROT[1]** or **AWPROT[1]** bits provided on the expansion interfaces to determine the security permission of each access. Any accesses that fail any external security checks must result in a DECERR response.

7.1.5 DRAM

DRAM areas are mapped to each memory access port. RD-V1 utilizes multiple memory channels. Transactions are distributed across these channels using address striping.

7.2 AP interrupt map

The Generic Interrupt Controller Architecture defines two different types of interrupts. PPIs exist separately for every microprocessor. SPIs are shared for all microprocessors. PPI and SPI interrupts have configurable options, including number of interrupts, Edge or Level triggered, and Polarity, that is, active-LOW, active-HIGH, or Rising edge.

7.2.1 PPI interrupts

Table 7-5 shows the PPI map for the application processors. The interrupt map gets repeated for each of the cores in the system.

Table 7-5 AP PPI interrupt map table

ID	Name	Description	Edge/Level	Polarity
16	nERRIRQ	Core error interrupt	Level	Active-LOW
17	nFaultIRQ	Core fault interrupt	Level	Active-LOW
18-20	Reserved	-	-	-
21	PMBIRQn	SPE interrupt request. Only exists for cores that support the architectural Statistical Profiling Extension (SPE). Reserved otherwise.	Level	Active-LOW
22	COMMIRQn	Debug Communications Channel receive or transmit request	Level	Active-LOW
23	PMUIRQn	PMU interrupt	Level	Active-LOW
24	CTIIRQ	CTI Interrupt	Level	Rising edge
25	VCPUMNTIRQn	Virtual Maintenance Interrupt (PPI6)	Level	Active-LOW
26	CNTHPIRQn	Non-secure PL2 Timer event (PPI5)	Level	Active-LOW
27	CNTVIRQn	Virtual Timer event (PPI4)	Level	Active-LOW
28	CNTHVIRQn		Level	Active-LOW
29	CNTPSIRQn	Secure PL1 Physical Timer event (PPI1)	Level	Active-LOW
30	CNTPNSIRQn	Non-secure PL1 Physical Timer event (PPI2)	Level	Active-LOW
31	Reserved	-	-	-

7.2.2 SPI interrupts

Table 7-6 shows the SPI interrupt map for the applications processors in the system.

Table 7-6 AP SPI interrupt map table

ID	Name	Description	Edge/Level	Polarity
32	DMC0_pmuirq	PMU event interrupt from DMC0	Level	Active-HIGH
33	DMC0_comb_err_owflow	Combined Error interrupt overflow from DMC0	Level	Active-HIGH
34	DMC0_failed_access_int	TZ access error interrupt from DMC0	Level	Active-HIGH
35	DMC0_ecc_err	ECC Error from DMC0	Level	Active-HIGH
36	DMC1_pmuirq	PMU event interrupt from DMC1	Level	Active-HIGH
37	DMC1_comb_err_owflow	Combined Error interrupt overflow from DMC1	Level	Active-HIGH
38	DMC1_failed_access_int	TZ access error interrupt from DMC1	Level	Active-HIGH
39	DMC1_ecc_err	ECC Error from DMC1	Level	Active-HIGH
40	DMC2_pmuirq	PMU event interrupt from DMC2	Level	Active-HIGH
41	DMC2_comb_err_owflow	Combined Error interrupt overflow from DMC2	Level	Active-HIGH
42	DMC2_failed_access_int	TZ access error interrupt from DMC2	Level	Active-HIGH
43	DMC2_ecc_err	ECC Error from DMC2	Level	Active-HIGH
44	DMC3_pmuirq	PMU event interrupt from DMC3	Level	Active-HIGH
45	DMC3_comb_err_owflow	Combined Error interrupt overflow from DMC3	Level	Active-HIGH
46	DMC3_failed_access_int	TZ access error interrupt from DMC3	Level	Active-HIGH
47	DMC3_ecc_err	PMU event interrupt from DMC3	Level	Active-HIGH
48	DMC4_pmuirq	PMU event interrupt from DMC4	Level	Active-HIGH
49	DMC4_comb_err_owflow	Combined Error interrupt overflow from DMC4	Level	Active-HIGH
50	DMC4_failed_access_int	TZ access error interrupt from DMC4	Level	Active-HIGH
51	DMC4_ecc_err	PMU event interrupt from DMC4	Level	Active-HIGH
52	DMC5_pmuirq	PMU event interrupt from DMC5	Level	Active-HIGH
53	DMC5_comb_err_owflow	Combined Error interrupt overflow from DMC5	Level	Active-HIGH
54	DMC5_failed_access_int	TZ access error interrupt from DMC5	Level	Active-HIGH
55	DMC5_ecc_err	PMU event interrupt from DMC6	Level	Active-HIGH
56	DMC6_pmuirq	PMU event interrupt from DMC6	Level	Active-HIGH
57	DMC6_comb_err_owflow	Combined Error interrupt overflow from DMC6	Level	Active-HIGH
58	DMC6_failed_access_int	TZ access error interrupt from DMC6	Level	Active-HIGH
59	DMC6_ecc_err	PMU event interrupt from DMC7	Level	Active-HIGH
60	DMC7_pmuirq	PMU event interrupt from DMC7	Level	Active-HIGH
61	DMC7_comb_err_owflow	Combined Error interrupt overflow from DMC7	Level	Active-HIGH

ID	Name	Description	Edge/Level	Polarity
62	DMC7_failed_access_int	TZ access error interrupt from DMC7	Level	Active-HIGH
63	DMC7_ecc_err	PMU event interrupt from DMC7	Level	Active-HIGH
64-66	Reserved	-	-	-
67	MCP2APMHU_NS	MHU Non-secure interrupt	Level	Active-HIGH
68	Reserved	-	-	-
69	MCP2APMHU_S	MHU secure interrupt	Level	Active-HIGH
70-77	Reserved	-	-	-
78	CMN_INTREQPMU_DTC0	PMU Count Overflow Interrupt	Level	Active-HIGH
79	CMN_INTREQPMU_DTC1	PMU Count Overflow Interrupt if #DTC >1	Level	Active-HIGH
80	CMN_INTREQPMU_DTC2	PMU Count Overflow Interrupt if #DTC >2	Level	Active-HIGH
81	CMN_INTREQPMU_DTC3	PMU Count Overflow Interrupt if #DTC >3	Level	Active-HIGH
82	Reserved	-	-	-
83	STM-500	STM-500 Synchronization Interrupt	Edge	Rising edge
84	CTI	CTI trigger output 6 from CTI2	Edge	Rising edge
85	CTI	CTI trigger output 7 from CTI2	Edge	Rising edge
86	Trusted Watchdog	Trusted Watchdog interrupt(WSO)	Level	Active-HIGH
87-90	Reserved	-	-	-
91	AP_REFCLK Generic Timer (Secure)	AP_REFCLK Generic Timer Interrupt (Secure)	Level	Active-HIGH
92	AP_REFCLK Generic Timer (Non-secure)	AP_REFCLK Generic Timer Interrupt (Non-secure)	Level	Active-HIGH
93	Generic Watchdog	Watchdog WSO Interrupt	Level	Active-HIGH
94	Generic Watchdog	Watchdog WS1 Interrupt	Level	Active-HIGH
95	AP_UART_INT	AP UART interrupt	Level	Active-HIGH
96	AP_UART_INT1	AP UART1 interrupt	Level	Active-HIGH
97-127	Reserved	-	-	-
128-255	Expansion	<p>Expansion interrupts</p> <p>SoC can add interrupts from Alert logic, RTC, Thermal events, Power sensors events, GPIO events, I2C, SGI, PCI, SATA, Ethernet, Accelerators, DMAs.</p> <p>Note: Only level interrupts can be used to wake up when the SYSTOP is powered down.</p>	SoC IMPLEMENTATION DEFINED	Implementation dependent
256	MMUTCU1_PMU_IRPT(Reserved if NUM_TCU = 0)	PMU interrupt	Edge	Rising edge

ID	Name	Description	Edge/Level	Polarity
257	MMUTCU1_EVENT_Q_IRPT_S(Reserved if NUM_TCU = 0)	Event queue Secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
258	MMUTCU1_CMD_SYNC_IRPT_S(Reserved if NUM_TCU = 0)	SYNC Complete Secure interrupt	Edge	Rising edge
259	MMUTCU1_GLOBAL_IRPT_S(Reserved if NUM_TCU = 0)	Global Secure interrupt	Edge	Rising edge
260	MMUTCU1_EVENT_Q_IRPT_NS(Reserved if NUM_TCU = 0)	Event queue Non-secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
261	MMUTCU1_CMD_SYNC_IRPT_NS(Reserved if NUM_TCU = 0)	SYNC Complete Non-secure interrupt	Edge	Rising edge
262	MMUTCU1_GLOBAL_IRPT_NS(Reserved if NUM_TCU = 0)	Global Non-secure interrupt	Edge	Rising edge
263	MMUTCU2_PMU_IRPT(Reserved if NUM_TCU < 2)	PMU interrupt	Edge	Rising edge
264	MMUTCU2_EVENT_Q_IRPT_S(Reserved if NUM_TCU < 2)	Event queue Secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
265	MMUTCU2_CMD_SYNC_IRPT_S(Reserved if NUM_TCU < 2)	SYNC Complete Secure interrupt	Edge	Rising edge
266	MMUTCU2_GLOBAL_IRPT_S(Reserved if NUM_TCU < 2)	Global Secure interrupt	Edge	Rising edge
267	MMUTCU2_EVENT_Q_IRPT_NS(Reserved if NUM_TCU < 2)	Event queue Non-secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
268	MMUTCU2_CMD_SYNC_IRPT_NS(Reserved if NUM_TCU < 2)	SYNC Complete Non-secure interrupt	Edge	Rising edge
269	MMUTCU2_GLOBAL_IRPT_NS(Reserved if NUM_TCU < 2)	Global Non-secure interrupt	Edge	Rising edge
270	MMUTCU3_PMU_IRPT(Reserved if NUM_TCU < 2)	PMU interrupt	Edge	Rising edge
271	MMUTCU3_EVENT_Q_IRPT_S(Reserved if NUM_TCU < 3)	Event queue Secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
272	MMUTCU3_CMD_SYNC_IRPT_S(Reserved if NUM_TCU < 3)	SYNC Complete Secure interrupt	Edge	Rising edge
273	MMUTCU3_GLOBAL_IRPT_S(Reserved if NUM_TCU < 3)	Global Secure interrupt	Edge	Rising edge
274	MMUTCU3_EVENT_Q_IRPT_NS(Reserved if NUM_TCU < 3)	Event queue Non-secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
275	MMUTCU3_CMD_SYNC_IRPT_NS(Reserved if NUM_TCU < 3)	SYNC Complete Non-secure interrupt	Edge	Rising edge

ID	Name	Description	Edge/Level	Polarity
276	MMUTCU3_GLOBAL_IRPT_NS (Reserved if NUM_TCU < 3)	Global Non-secure interrupt	Edge	Rising edge
277	MMUTCU4_PMU_IRPT (Reserved if NUM_TCU < 4)	PMU interrupt	Edge	Rising edge
278	MMUTCU4_EVENT_Q_IRPT_S (Reserved if NUM_TCU < 4)	Event queue Secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
279	MMUTCU4_CMD_SYNC_IRPT_S (Reserved if NUM_TCU < 4)	SYNC Complete Secure interrupt	Edge	Rising edge
280	MMUTCU4_GLOBAL_IRPT_S (Reserved if NUM_TCU < 4)	Global Secure interrupt	Edge	Rising edge
281	MMUTCU4_EVENT_Q_IRPT_NS (Reserved if NUM_TCU < 4)	Event queue Non-secure interrupt, indicating Event queue Non-Empty or Overflow	Edge	Rising edge
282	MMUTCU4_CMD_SYNC_IRPT_N S (Reserved if NUM_TCU < 4)	SYNC Complete Non-secure interrupt	Edge	Rising edge
283	MMUTCU4_GLOBAL_IRPT_NS (Reserved if NUM_TCU < 4)	Global Non-secure interrupt	Edge	Rising edge
284- 319	Reserved for interrupts from multiple TCU	Reserved for up to support up to 4 TCUs total	Edge	Rising edge
320- 383	MMUTBU_PMU_IRPT[NUM_TBU S-1:0]	TBU PMU Interrupt. Allocated to support up to 16 TBU.	Edge	Rising edge
384- 511	Reserved	-	-	-
512- 512 + NU M_C LUS TERS * 2 - 1	{Cluster<n> SCP ->AP MHU Secure, CLUSTER<n> SCP ->AP MHU Non-secure }	CLUSTER<n>_SCP2APMHU_NS	Level	Active-HIGH
576- 576 + NU M_C LUS TER - 1	PO_REFCLK_GENTIM - P{NUM_CLUSTERS- 1:0}_REFCLK_GENTIM	Pn_REFCLK Generic Secure Timer interrupts	Level	Active-HIGH
609- 1919	Reserved	-	-	-

7.3 SCP memory map

The System Control Processor is a Cortex-M7 based subsystem which implements a 32-bit address space. The Cortex-M7 uses a fixed high level memory map as specified in the Arm® *Architecture Reference Manual* ARMv7 and is shown in Table 7-7 SCP memory map table. The SCP core is always treated as a trusted core. See the Arm® *Architecture Reference Manual* ARMv7 for recommendations regarding the use of these memory areas.

Table 7-7 SCP memory map table

Start Address	End Address	Memory Region
0x0000_0000	0x007F_FFFF	BOOT ROM
0x0080_0000	0x00FF_FFFF	ITCMRAM
0x0100_0000	0x0FFF_FFFF	SCP_SOC_EXPANSION
0x1000_0000	0x1FFF_FFFF	SCP_SOC_EXPANSION
0x2000_0000	0x20FF_FFFF	DTCRAM
0x2100_0000	0x29FF_FFFF	Memory Region allocated for eight DMC controllers. See Memory Controller region.
0x2A00_0000	0x3FFF_FFFF	Reserved
0x4000_0000	0x43FF_FFFF	SCP SOC EXPANSION
0x4400_0000	0x4400_0FFF	REFCLK_CNTCTL
0x4400_1000	0x4400_1FFF	REFCLK_CNTBASE0
0x4400_2000	0x4400_2FFF	SCP UART
0x4400_3000	0x4400_5FFF	Reserved
0x4400_6000	0x4400_6FFF	Watchdog
0x4400_7000	0x4400_7FFF	Reserved
0x4400_8000	0x4400_8FFF	Reserved
0x4400_9000	0x4400_9FFF	Reserved
0x4400_A000	0x4400_AFFF	CS CNTCONTROL
0x4400_B000	0x4400_BFFF	Reserved
0x4400_C000	0x440F_FFFF	Reserved
0x4410_0000	0x44FF_FFFF	Generic Pn wakeup timers
0x4500_0000	0x453F_FFFF	SCP_AP_MHUV2 Non-secure
0x4540_0000	0x457F_FFFF	SCP_AP_MHUV2 Secure
0x4580_0000	0x4583_FFFF	SCP_MCP_MHUV2
0x4584_0000	0x46FF_FFFF	Reserved
0x4700_0000	0x4700_FFFF	SYSCNT0_MSTSYNC_CTRL
0x4701_0000	0x4701_FFFF	CCNT0_MSTSYNC_CTRL
0x4702_0000	0x4702_FFFF	SYSCNT1_MSTSYNC_CTRL
0x4703_0000	0x4703_FFFF	CCNT1_MSTSYNC_CTRL

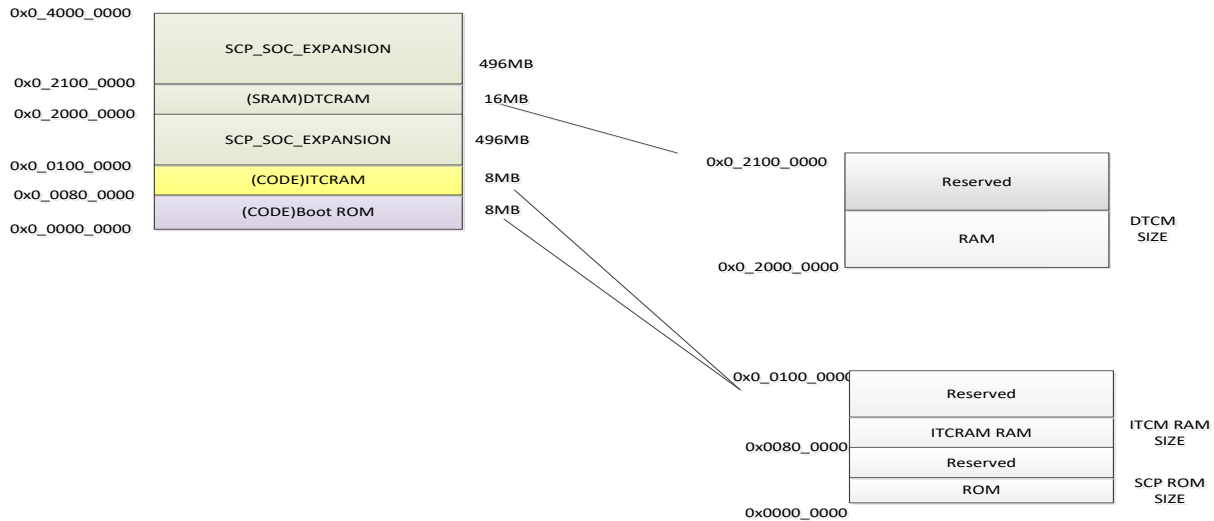
Start Address	End Address	Memory Region
0x4704_0000	0x4704_FFFF	SYS_CNT2_MSTSYNC_CTRL
0x4705_0000	0x4705_FFFF	CCNT2_MSTSYNC_CTRL
0x4706_0000	0x4706_FFFF	SYS_CNT3_MSTSYNC_CTRL
0x4707_0000	0x4707_FFFF	CCNT3_MSTSYNC_CTRL
0x4708_0000	0x48FF_FFFF	Reserved
0x4709_0000	0x47FF_FFFF	Reserved
0x4800_0000	0x4BFF_FFFF	Reserved
0x4C00_0000	0x4CFF_FFFF	Reserved
0x4D00_0000	0x4DFF_FFFF	Reserved
0x4E00_0000	0x4FFF_FFFF	Reserved
0x5000_0000	0x5000_FFFF	SCP PIK
0x5001_0000	0x5001_FFFF	Reserved
0x5002_0000	0x5002_FFFF	Debug PIK
0x5003_0000	0x5003_FFFF	Debug Sensor Group
0x5004_0000	0x5004_FFFF	System PIK
0x5005_0000	0x5005_FFFF	System Sensor Group
0x5006_0000	0x5085_FFFF	Reserved This region is reserved for CPU cluster PIKs if present in a system to support up to 64 clusters. CPU cluster MGR+Sensor NUM_CLUS * (64KB CPU cluster<n>MGR + 64KB CPU cluster<n> Sensor Group Example: 0x5006_0000 - 0x5006_FFFF - CPU_CLUSTER0_MGR 0x5007_0000 - 0x5007_FFFF - CPU_CLUSTER0_Sensor Group 0x5008_0000 - 0x5008_FFFF - CPU_CLUSTER1_MGR 0x5009_0000 - 0x5009_FFFF - CPU_CLUSTER1_Sensor Group)
0x5086_0000	0x5086_FFFF	Reserved memory space for DBGCHx PPU 0x5050_0000 - 0x5050_0FFF - DBGCH0 PPU 0x5050_1000 - 0x5050_1FFF - DBGCH1 PPU
0x5087_0000	0x50EF_FFFF	Reserved
0x50F0_0000	0x50FF_FFFF	Expansion PIK ports
0x5100_0000	0x5FFF_FFFF	Reserved
0x6000_0000	0x9FFF_FFFF	System Access Port
0xA000_0000	0xDFFF_FFFF	System Access Port
0xE000_0000	0xE003_FFFF	PPB (Internal)
0xE004_0000	0xE00F_FFFF	PPB (External)
0xE010_0000	0xFFFF_FFFF	Reserved

7.3.1 ROM and RAM sizes

The bottom 16MB of the address contains a SIZE=64KB boot ROM (starting at BASE=0x0_0000_0000), an ITCRAM=256KB (starting at BASE=0x0_0080_0000), and DTCRAM=256KB (starting at ASE=0x0_2000_0000).

Actual ROM and RAM sizes are configurable and implementation dependent, as shown in Figure 7-8.

Figure 7-8 SCP on-chip memory map



Any Read or Write accesses outside the configured RAM or ROM size causes respective TCM errors to the SCP core.

7.3.2 Peripherals region

The first 1MB of the top 512MB of address space is reserved for the Private Peripheral Bus (PPB). This region is further divided into internal PPB and external PPB.

The internal PPB (see Figure 7-9) is the bottom 256KB of the PPB space and is accessed using an AHB-Lite bus with the SCP subsystem. This space includes Cortex-M7 system components such as System Control Space (SCS), FPB, DWT, and ITM.

Figure 7-9 SCP PPB internal memory map

0x0_E004_0000	Reserved	196KB
0x0_E000_F000 0x0_E000_E000	SCS	4KB
	Reserved	44KB
0x0_E000_3000	FPB	4KB
0x0_E000_2000	DWT	4KB
0x0_E000_1000	ITM	4KB
0x0_E000_0000		

- The external PPB space (see Figure 7-10) contains more Cortex-M7 system components. They are generally debug-related components (SWO, ETM, CTI, SCP Funnel, and ROM Table). The rest of the address space is reserved. The ROM table follows the format specified in the *Arm® Cortex®-M7 Processor Technical Reference Manual* with the part number set to 0x4D0.

Figure 7-10 SCP PPB external memory map

0x0_E010_0000	PPB ROM Table	
0x0_E00F_F000 0x0_E00F_E000	Processor ROM Table	4KB
	Reserved	4KB
0x0_E004_6000	ATB Replicator	4KB
0x0_E004_5000	Funnel SCP	4KB
0x0_E004_4000	SWO	4KB
0x0_E004_3000	CTI	4KB
0x0_E004_2000	ETM	4KB
0x0_E004_1000	Reserved	8KB
0x0_E004_0000		

All other address space is accessed using the Cortex-M7 System bus and is divided into the following regions:

- The SCP Peripherals region contains the Generic Timers, Generic Counters, Watchdog Timer, Configuration registers and Power registers, as shown in Table 7-7.
- The Element Management Peripherals (8MB) region contains interfaces to management peripherals that either exist as part of the SCP Element, or through APB interfaces to other elements.
- The External RAM (1GB) region is assumed to behave as Normal memory. This region is by default mapped to a single contiguous 1GB region of the applications memory map starting at 0x00_4000_0000. Any SCP memory accesses in the External RAM region (0x6000_0000 to 0x9FFF_FFFF) actually target memory locations in the region 0x00_4000_0000 to

0x00_7FFF_FFFF of the applications memory map. This region of the applications memory map contains the expansion AXI space.

- The External Device (1GB) region is normally intended for off-chip Device memory. By default, this region is mapped to a single contiguous 1GB region of the applications memory map starting at 0x00_0000_0000.



Note

By default, if DBG_ADDR_TRANS[EN] is cleared, any SCP memory accesses in the External Device region (0xA000_0000 to 0xDFFF_FFFF) target memory locations in the region 0x00_0000_0000 to 0x00_3FFF_FFFF of the AP memory map. This region of the base memory map contains the Boot area, subsystem peripherals, and an expansion AXI area.



Note

If DBG_ADDR_TRANS[EN] is set to 1'b1, then any SCP memory accesses in the External Device region (0xA000_0000 to 0xDFFF_FFFF) target memory locations in the region 0x04_0000_0000 to 0x04_3FFF_FFFF of the memory map. This region of the memory map contains CoreSight debug components.

- All vendor-specific peripherals can be put into the SCP SoC Expansion space (shown in Table 7-7). These include the register state for controlling the Power Management Integrated Circuit (PMIC), the PLLs, the process, voltage and temperature (PVT) sensors, and the scratch RAMs. This area is accessible through the SCP external AXI Expansion interface.

7.4 SCP interrupt map

The SCP receives interrupts from the following sources:

- Applications processor system wakeup interrupts
- CoreSight power and reset request interrupts
- Internal SCP subsystem interrupts
- Expansion SCP interrupts

These interrupts are routed to the NVIC that is included in the Cortex-M7 processor, where they can be managed by software.

Table 7-8 SCP interrupt map table

ID	Source	Description	Edge/Level	Polarity
NMI	SCP Generic Watchdog	SCP Watchdog(WSO)	Level	Active-HIGH
0	CoreSight	CDBGPWRREQ_INT	Level	Active-HIGH
1	CoreSight	CSYSPWRREQ_INT	Level	Active-HIGH
2	CoreSight	CDBGIRSTREQ_INT	Level	Active-HIGH
3	CoreSight	CSYSIRSTREQ_INT	Level	Active-HIGH

ID	Source	Description	Edge/Level	Polarity
4	GIC expansion interrupt	External GIC wakeup interrupt. This is generated by the logical OR of all the GIC Expansion Interrupts	Level	Active-HIGH
5	Reserved	Reserved for CryptoCell™ TEE interrupt – expected to be handled by Trusted world SW.	-	-
6	Reserved	Reserved for CryptoCell REE interrupt – expected to be handled by Normal world SW.	-	-
7-13	Reserved	-	-	-
14-15	Reserved	-	-	-
16-31	External	SoC AON Expansion interrupts	Level or Edge. SoC IMPLEMENTATION DEFINED	Active-HIGH
32	Reserved	-	-	-
33	SCP REFCLK Generic Timer	REFCLK Physical Timer interrupt	Level	Active-HIGH
34	GENTIM_SYNC	System generic timer synchronization interrupt	Level	Active-HIGH
35	CSTS_SYNC	CoreSight Time stamp synchronization interrupt	Level	Active-HIGH
36	Reserved	-	-	-
37	CTI	CTI trigger 0	Edge	Active-HIGH
38	CTI	CTI trigger 1	Edge	Active-HIGH
39	GICECCFATAL	GIC Fatal ECC failure	Level	Active-HIGH
40	GICAXIMERR	GIC Fatal AXI Master error	Level	Active-HIGH
41	Reserved	-	-	-
42	AON_UART_INT	Always-on UART interrupt	Level	Active-HIGH
43	Reserved	-	-	-
44	Generic Watchdog	Generic Watchdog timer interrupt WS0	Level	Active-HIGH
45	Generic Watchdog	Generic Watchdog timer interrupt WS1	Level	Active-HIGH
46	Trusted Watchdog	Trusted Watchdog timer interrupt WS0	Level	Active-HIGH
47	Trusted Watchdog	Trusted Watchdog timer interrupt WS1	Level	Active-HIGH
48	APPS_UART_INT	Applications UART interrupt	Level	Active-HIGH
49	Reserved	-	-	-
50	Consolidated CPU Core Power Policy Units	Consolidated CPU PPU Interrupt for 1-32 cores	Level	Active-HIGH
51-53	Reserved	-	-	-
54	Reserved	-	-	-

ID	Source	Description	Edge/Level	Polarity
55	Consolidated CPU Core PLL Lock	Consolidated CPU PLL Lock for PLLs 1-32	Level	Active-HIGH
56	Reserved	-	-	-
57	Consolidated CPU PLL UNLock	Consolidated CPU PLL UNLock for PLLs 1-32	Level	Active-HIGH
58	Reserved	-	-	-
59	Consolidated CPU Core Fault Indicator interrupts	Consolidated nFaultIRQ for cores 1-32	Level	Active-LOW
60-63	Reserved	-	-	-
64	Consolidated CPU ECC error interrupts	Consolidated nERRIRQ for cores 1-32	Level	Active-LOW
65-83	Reserved	-	-	-
84	MCP2SCP MHU Non-secure Int	MCP2SCP MHU High Priority Int	Level	Active-HIGH
85	MCP2SCP MHU Secure Int	MCP2SCP MHU High Priority Int	Level	Active-HIGH
86-89	Reserved	-	-	-
90	Pn_REFCLK_GENTIM_1_32	Consolidated Pn REFCLK Timer Interrupt (1-32 clusters)	Level	Active-HIGH
91-93	Reserved, for wake-up Timers	-	-	-
94	CONS_MMU_TCU_RASIRPT	Consolidated MMU RAS for the interrupt coming from multiple TCUs	Level	Active-HIGH
95	CONS_MMU_TBU_RASIRPT [NUM_TBUS-1:0]	Consolidated TBU for the interrupts coming from various TBUs	Level	Active-HIGH
96	INTREQPPU	PPU interrupt from CMN-650	Level	Active-HIGH
97	INTREQERRNS	Non-secure error handling interrupt from CMN	Level	Active-HIGH
98	INTREQERRS	Secure error handling interrupt from CMN	Level	Active-HIGH
99	INTREQFAULTS	Secure Fault handling interrupt from CMN	Level	Active-HIGH
100	INTREQFAULTNS	Non-secure Fault handling interrupt from CMN	Level	Active-HIGH
101	INTREQPMU	PMU count overflow interrupt	Level	Active-HIGH
102-119	Reserved	-	-	-
120-127	DBGCH[0-7]_PPU_INT	Debug channel PPU interrupts	-	Active-HIGH
128-129	Reserved	-	-	-
130	Power Integration Kit	Debug PIK Interrupt	Level	Active-HIGH

ID	Source	Description	Edge/Level	Polarity
131	LOGIC_PPU_INT	LOGIC PPU Interrupt	Level	Active-HIGH
132-134	Reserved	-	-	-
135	SRAM_PPU_INT	SRAM PPU Interrupt	Level	Active-HIGH
136-138	Reserved	-	-	-
139	MCP_WS1	MCP Watch dog reset	Level	Active-HIGH
140	SYSPLL_LOCK	Sys PLL Lock	Level	Active-HIGH
141	SYSPLL_UNLOCK	Sys PLL Unlock	Level	Active-HIGH
142	INTPLL_LOCK	Interconnect PLL Lock	Level	Active-HIGH
143	INTPLL_UNLOCK	Interconnect PLL UnLock	Level	Active-HIGH
144-173	Reserved	-	-	-
174	DMC_PLL_LOCK	DMC PLL Lock	Level	Active-HIGH
175	DMC_PLL_UNLOCK	DMC PLL Unlock	Level	Active-HIGH
176-179	-	Reserved	Level	Active-HIGH
180	DMC0/DMC4 Interrupts	DMC0/DMC4_combined_misc oflow	Level	Active-HIGH
181	-	DMC0/DMC4_combined_err_oflow	Level	Active-HIGH
182	-	DMC0/DMC4_combined_ecc_err_int	Level	Active-HIGH
183	-	DMC0/DMC4_combined_misc_access_int	Level	Active-HIGH
184	-	DMC0/DMC4_temperature_event_int	Level	Active-HIGH
185	-	DMC0/DMC4_failed_access_int	Level	Active-HIGH
186	-	DMC0/DMC4_combined_mgr_int	Level	Active-HIGH
187	DMC1/DMC5 Interrupts	DMC1/DMC5_combined_misc oflow	Level	Active-HIGH
188	-	DMC1/DMC5_combined_err_oflow	Level	Active-HIGH
189	-	DMC1/DMC5_combined_ecc_err_int	Level	Active-HIGH
190	-	DMC1/DMC5_combined_misc_access_int	Level	Active-HIGH
191	-	DMC1/DMC5_temperature_event_int	Level	Active-HIGH
192	-	DMC1/DMC5_failed_access_int	Level	Active-HIGH
193	-	DMC1/DMC5_combined_mgr_int	Level	Active-HIGH
194	DMC2/DMC6 Interrupts	DMC2/DMC6_combined_misc oflow	Level	Active-HIGH
195	-	DMC2/DMC6_combined_err_oflow	Level	Active-HIGH
196	-	DMC2/DMC6_combined_ecc_err_int	Level	Active-HIGH
197	-	DMC2/DMC6_combined_misc_access_int	Level	Active-HIGH
198	-	DMC2/DMC6_temperature_event_int	Level	Active-HIGH
199	-	DMC2/DMC6_failed_access_int	Level	Active-HIGH

ID	Source	Description	Edge/Level	Polarity
200	-	DMC2/DMC6_combined_mgr_int	Level	Active-HIGH
201	DMC3/DMC7 Interrupts	DMC3/DMC7_combined_misc oflow	Level	Active-HIGH
202	-	DMC3/DMC7_combined_err_oflow	Level	Active-HIGH
203	-	DMC3/DMC7_combined_ecc_err_int	Level	Active-HIGH
204	-	DMC3/DMC7_combined_misc_access_int	Level	Active-HIGH
205	-	DMC3/DMC7_temperature_event_int	Level	Active-HIGH
206	-	DMC3/DMC7_failed_access_int	Level	Active-HIGH
207	-	DMC3/DMC7_combined_mgr_int	Level	Active-HIGH
208-239	External Interrupts	32 Interrupts for SCP SoC Expansion	Level	Active-HIGH

7.5 MCP memory map

The MCP is a Cortex-M7-based subsystem that implements a 32-bit address space. The Cortex-M7 uses a fixed high-level memory map as specified in the *Arm® Architecture Reference Manual ARMv7*, that also contains recommendations regarding the use of these memory areas.

Table 7-9 MCP memory map table

Start Address	End Address	Memory Region
0x0000_0000	0x007F_FFFF	BOOT ROM
0x0080_0000	0x00FF_FFFF	ITCMRAM
0x0100_0000	0x0FFF_FFFF	MCP_SOC_EXPANSION
0x1000_0000	0x1FFF_FFFF	MCP_SOC_EXPANSION
0x2000_0000	0x20FF_FFFF	DTCRAM
0x2100_0000	0x3FFF_FFFF	MCP_SOC_EXPANSION
0x4000_0000	0x43FF_FFFF	MCP SOC EXPANSION
0x4400_0000	0x454F_FFFF	Reserved
0x4550_0000	0x455F_FFFF	Reserved
0x4560_0000	0x4560_FFFF	SCP2MCP MHU
0x4561_0000	0x4561_FFFF	SCP2MCP MHU Non-secure RAM
0x4562_0000	0x4562_FFFF	SCP2MCP MHU Secure RAM
0x4563_0000	0x47FF_FFFF	Reserved
0x4800_0000	0x4BFF_FFFF	MCP_SOC_Expansion
0x4C00_0000	0x4C00_0FFF	REFCLK_CNTCTL
0x4C00_1000	0x4C00_1FFF	REFCLK_CNTBASE0
0x4C00_2000	0x4C00_2FFF	MCP_UART1

Start Address	End Address	Memory Region
0x4C00_3000	0x4C00_3FFF	MCP_UART2
0x4C00_4000	0x4C00_5FFF	Reserved
0x4C00_6000	0x4C00_6FFF	Watchdog
0x4C00_7000	0x4C00_7FFF	Reserved
0x4C00_8000	0x4C00_8FFF	Reserved
0x4C00_9000	0x4C00_9FFF	Reserved
0x4C00_A000	0x4C00_AFFF	Reserved
0x4C00_B000	0x4C00_BFFF	Reserved
0x4C00_C000	0x4C0F_AFFF	Reserved
0x4C10_0000	0x4C3F_FFFF	Reserved
0x4C40_0000	0x4C40_FFFF	MCP_AP_SND_NS
0x4C41_0000	0x4C41_FFFF	MCP_AP_RCV_NS
0x4C42_0000	0x4C42_FFFF	MCP_AP_SND_S
0x4C43_0000	0x4C43_FFFF	MCP_AP_RCV_S
0x4C44_0000	0x4DFF_FFFF	Reserved
0x4E00_0000	0x4FFF_FFFF	Reserved (for Memory Controller)
0x5000_0000	0x5000_FFFF	MCP PIK
0x5001_0000	0x507F_FFFF	Reserved
0x5080_0000	0x5FFF_FFFF	Reserved
0x6000_0000	0x9FFF_FFFF	System Access Port
0xA000_0000	0xDFFF_FFFF	System Access Port
0xE000_0000	0xE003_FFFF	PPB (Internal)
0xE004_0000	0xE00F_FFFF	PPB (External)
0xE010_0000	0xFFFF_FFFF	Reserved

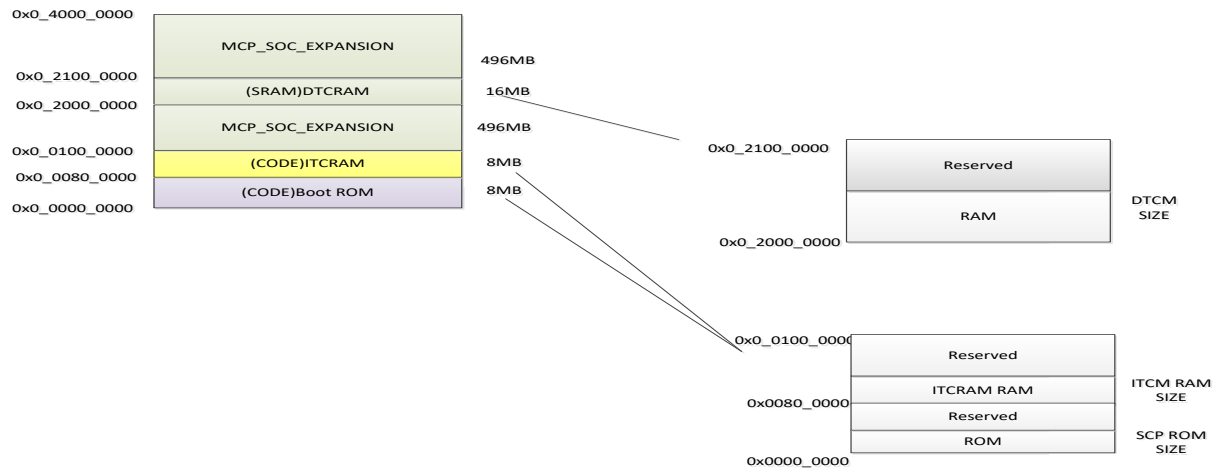
7.5.1 ROM and RAM sizes

The bottom 512MB of the address contains a SIZE=64KB boot ROM (starting at BASE=0x0_0000_0000), an ITCRAM=64KB (starting at BASE=0x0_1000_0000) and DTCRAM=64KB (starting at BASE=0x0_2000_0000).

The actual ROM/RAM sizes are configurable and implementation dependent, as shown in Figure 7-11.

Any Read or Write Accesses outside the configured RAM/ROM size causes respective TCM errors to the SCP core.

Figure 7-11 MCP on-chip memory map

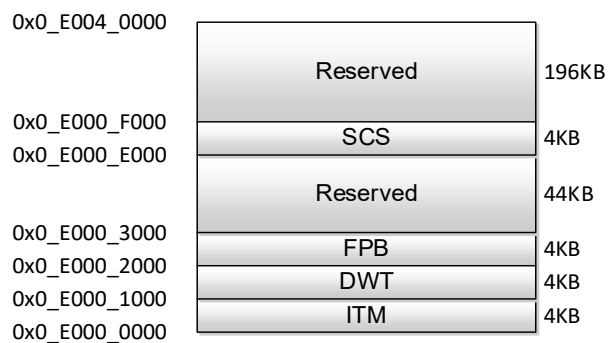


7.5.2 Peripherals region

The first 1MB of the top 512MB of address space is reserved for the Private Peripheral Bus (PPB). This region is further divided into internal PPB and external PPB.

The Internal PPB space (see Figure 7-12) is the bottom 256KB of the PPB space and is accessible using an AHB-Lite bus with the MCP subsystem. This space includes Cortex-M7 system components such as System Control Space (SCS), FPB, DWT, and ITM.

Figure 7-12 MCP PPB internal memory map



The External PPB space (see Figure 7-13) contains more Cortex-M7 system components, which are generally debug-related components (SWO, ETM, CTI, MCP Funnel, and ROM Table). The rest of the address space is reserved. The ROM table follows the format specified in Cortex-M7 TRM with the part number set to 0x4D0.

Figure 7-13 MCP PPB external memory map

Private Peripheral Bus – External Region		
0x0_E010_0000	PPB ROM Table	4KB
0x0_E00F_F000	Processor ROM Table	
0x0_E00F_E000	Reserved	
0x0_E004_6000	Reserved	4KB
0x0_E004_5000	Reserved	4KB
0x0_E004_4000	Funnel MCP	4KB
0x0_E004_3000	Reserved	4KB
0x0_E004_2000	CTI	4KB
0x0_E004_1000	ETM	4KB
0x0_E004_0000	Reserved	8KB

All other address space is accessed using the Cortex-M7 System bus and is divided into the following regions:

- MCP peripherals, containing Generic Timers, Watchdog Timer, Configuration registers and Power registers.
- Element Management Peripherals (8MB), containing interfaces to management peripherals that either exist as part of the MCP Element, or through APB interfaces to other elements.
- External RAM (1GB) region, assumed to behave as Normal memory. By default, this region is mapped to a single contiguous 1GB region of the applications memory map starting at 0x00_4000_0000. Any MCP memory accesses in the External RAM region (0x6000_0000 to 0x9FFF_FFFF) actually target memory locations in the region 0x00_4000_0000 to 0x00_7FFF_FFFF of the applications memory map. This region of the applications memory map contains the expansion AXI space.
- External Device (1GB) region, normally intended for off-chip Device memory. By default, this region is mapped to a single contiguous 1GB region of the applications memory map starting at 0x00_0000_0000.



Note

By default, if DBG_ADDR_TRANS[EN] is cleared, any SCP memory accesses in the External Device region (0xA000_0000 to 0xDFFF_FFFF) target memory locations in the region 0x00_0000_0000 to 0x00_3FFF_FFFF of the base memory map. This region of the base memory map contains the Boot area, subsystem peripherals, and an expansion AXI area.



Note

If DBG_ADDR_TRANS[EN] is set to 1'b1, then any SCP memory accesses in the External Device region (0xA000_0000 to 0xDFFF_FFFF) target memory locations in the region 0x04_0000_0000 to 0x04_3FFF_FFFF of the memory map. This region of the memory map contains CoreSight debug components.

- MCP SoC Expansion (see Table 7-9) includes all vendor-specific peripherals such as the memory space for logging events, power and reset control register. This area is accessible through the MCP external AHB Expansion interface.

7.6 MCP Interrupt map

The MCP receives interrupts from the following sources:

- CoreSight power and reset request interrupts
- Internal MCP subsystem interrupts
- Expansion MCP interrupts

These interrupts are routed to the NVIC that is included in the Cortex-M7 processor, where they can be managed by software.

Table 7-10 MCP interrupt map table

ID	Source	Description	Edge / Level	Polarity
NMI	MCP Generic Watchdog	MCP Watchdog(WSO)	Level	Active-HIGH
1	CoreSight Debug Power up Request, Reserved	CoreSight debug power up request (if there is a separate debug power domain) Note: SCP Firmware must support optional debug power up rail.	Internal	Edge
2	CoreSight System Power up request	CoreSight system power up request	Internal	Edge
3	CoreSight Debug Reset Request	CoreSight debug reset request	Internal	Edge
4	GIC expansion interrupt	External GIC wakeup interrupt. This is generated by the logical OR of all the GIC Expansion Interrupts (see the Applications Processors Interrupt Map).	Level	Active-HIGH
5-15	Reserved	-	-	-
16-31	External	SoC AON Expansion interrupts	Level or Edge. SoC IMPLEMENTATION DEFINED	Active-HIGH
32	Reserved	-	-	-
33	MCP REFCLK Generic Timer	REFCLK Physical Timer interrupt	Level	Active-HIGH
34	Non-secure AP2MCP MHU	MHU Non-secure interrupt	Level	Active-HIGH
35	Reserved	-	Level	Active-HIGH
36	AP2MCP Secure MHU	MHU Secure interrupt	Level	Active-HIGH (tie off 1'b0)

ID	Source	Description	Edge / Level	Polarity
37	CTI	CTI trigger 0	Edge	Rising and Falling edge
38	CTI	CTI trigger 1	Edge	Rising and Falling edge
39-41	Reserved	-	-	-
42	MCP_UART0_INT	Always-on UART interrupt	Level	Active-HIGH
43	MCP_UART1_INT	Always-on UART interrupt	Level	Active-HIGH
44-83	Reserved	-	-	-
84	MCP2SCP MHU Non-secure Int	MCP2SCP MHU High Priority Int	Internal	Level
85	MCP2SCP MHU Secure Int	MCP2SCP MHU High Priority Int	Internal	Level
86-93	Reserved	-	-	-
94	MMU_TCU_RASIRPT	Consolidated MMU RAS interrupt from TCU	Level	Active-HIGH
95	MMU_TBU_RASIRPT [NUM_TBUS-1:0]	Consolidated TBU interrupts from various TBUs	Level	Active-HIGH
96	INTREQPPU	PPU interrupt from CMN-650	Level	Active-HIGH
97	INTREQERRNS	Non-secure error handling interrupt from CMN-650	Level	Active-HIGH
98	INTREQERRS	Secure error handling interrupt from CMN-650	Level	Active-HIGH
99	INTREQFAULTS	Secure Fault handling interrupt from CMN-650	Level	Active-HIGH
100	INTREQFAULTNS	Non-secure Fault handling interrupt from CMN-650	Level	Active-HIGH
101	INTREQPMU	PMU count overflow interrupt	Level	Active-HIGH
102-138	Reserved	-	-	-
139	MCP_WS1	MCP Watch dog reset	Level	Active-HIGH
140-179	Reserved	-	-	-
180	DMC0/DMC4 Interrupts	DMC0/DMC4_combined_misc oflow	Level	Active-HIGH
181		DMC0/DMC4_combined_err_oflow	Level	Active-HIGH
182		DMC0/DMC4_combined_ecc_err_int	Level	Active-HIGH
183		DMC0/DMC4_combined_misc_access_int	Level	Active-HIGH
184		DMC0/DMC4_temperature_event_int	Level	Active-HIGH
185		DMC0/DMC4_failed_access_int	Level	Active-HIGH
186		DMC0/DMC4_combined_mgr_int	Level	Active-HIGH
187	DMC1/DMC5 Interrupts	DMC1/DMC5_combined_misc oflow	Level	Active-HIGH
188		DMC1/DMC5_combined_err_oflow	Level	Active-HIGH
189		DMC1/DMC5_combined_ecc_err_int	Level	Active-HIGH

ID	Source	Description	Edge / Level	Polarity
190		DMC1/DMC5_combined_misc_access_int	Level	Active-HIGH
191		DMC1/DMC5_temperature_event_int	Level	Active-HIGH
192		DMC1/DMC5_failed_access_int	Level	Active-HIGH
193		DMC1/DMC5_combined_mgr_int	Level	Active-HIGH
194	DMC2/DMC6 Interrupts	DMC2/DMC6_combined_misc oflow	Level	Active-HIGH
195		DMC2/DMC6_combined_err_oflow	Level	Active-HIGH
196		DMC2/DMC6_combined_ecc_err_int	Level	Active-HIGH
197		DMC2/DMC6_combined_misc_access_int	Level	Active-HIGH
198		DMC2/DMC6_temperature_event_int	Level	Active-HIGH
199		DMC2/DMC6_failed_access_int	Level	Active-HIGH
200		DMC2/DMC6_combined_mgr_int	Level	Active-HIGH
201	DMC3/DMC7 Interrupts	DMC3/DMC7_combined_misc oflow	Level	Active-HIGH
202		DMC3/DMC7_combined_err_oflow	Level	Active-HIGH
203		DMC3/DMC7_combined_ecc_err_int	Level	Active-HIGH
204		DMC3/DMC7_combined_misc_access_int	Level	Active-HIGH
205		DMC3/DMC7_temperature_event_int	Level	Active-HIGH
206		DMC3/DMC7_failed_access_int	Level	Active-HIGH
207		DMC3/DMC7_combined_mgr_int	Level	Active-HIGH
208-239	External Interrupts	32 Interrupts for MCP SoC Expansion	Level	Active-HIGH

7.7 Register set

The system control registers are described in the following sections.

7.7.1 System ID registers unit

System ID Registers Unit (SID) provides an interface for giving an ID to the system.

Table 7-11 summarizes the registers and their corresponding address offsets for the SID. The SID_BASE is the base address of the SID. All registers in this module allow Secure and Non-secure accesses.

Table 7-11 System ID register summary

Name	Address offset	Type	Default	Description
SID_SYSTEM_ID	SID_BASE + 0x0040	RO	0x10041030	ID register
SID_SOC_ID	SID_BASE + 0x0050	RO	System specific	ID register
SID_NODE_ID	SID_BASE + 0x0060	RO	System specific	ID register

Name	Address offset	Type	Default	Description
SID_SYSTEM_CFG	SID_BASE + 0x0070	RO	System specific	Configuration register
PID4	SID_BASE + 0x0FD0	RO	0x00000004	Peripheral ID 4
PID0	SID_BASE + 0x0FE0	RO	0x000000D2	Peripheral ID 0
PID1	SID_BASE + 0x0FE4	RO	0x000000B0	Peripheral ID 1
PID2	SID_BASE + 0x0FE8	RO	0x0000000B	Peripheral ID 2
PID3	SID_BASE + 0x0FEC	RO	0x00000000	Peripheral ID 3
COMPID0	SID_BASE + 0x0FF0	RO	0x0000000D	Component ID 0
COMPID1	SID_BASE + 0x0FF4	RO	0x000000F0	Component ID 1
COMPID2	SID_BASE + 0x0FF8	RO	0x00000005	Component ID 2
COMPID3	SID_BASE + 0x0FFC	RO	0x000000B1	Component ID 3

7.7.1.1 SID_SYSTEM_ID register

This register specifies the version ID for the subsystem.

Table 7-12 SID_SYSTEM_ID register

Bits	Name	Description
[31:28]	Variant Number	0x1 - Config-M 0x2 - Config-L 0x3 - Config-XL
[27:24]	MAJOR REVISION	Set to 0x0.
[23:20]	MINOR REVISION	Set to 0x0.
[19:12]	DESIGNER_ID	Arm product with designer code 0x41.
[11:0]	PART_NUMBER	0x785 to 0x78E – Used for the Reference Design (RD) products. 0x78F – Reserved RD-V1 PART_NUMBER defined as 0x78A

7.7.1.2 SID_SOC_ID register

This register specifies the ID for the SoC that integrates the subsystem.

Table 7-13 SID_SOC_ID register

Bits	Name	Description
[31:28]	-	Reserved
[27:24]	MAJOR REVISION	IMPLEMENTATION DEFINED
[23:20]	MINOR REVISION	IMPLEMENTATION DEFINED
[19:12]	DESIGNER_ID	IMPLEMENTATION DEFINED
[11:0]	PART_NUMBER	IMPLEMENTATION DEFINED

7.7.1.3 SID_NODE_ID register

This register specifies the ID for the node when there are multiple sockets.

Table 7-14 SID_NODE_ID register

Bits	Name	Description
[31:9]	-	Reserved
[8]	MULTI_CHIP_MODE	Read-only register bits to indicate the Multi-chip mode tie off value. 0 – Single-chip 1 – Multi-chip
[7:0]	NODE_NUMBER	Read-only register bits to indicate the NODE_NUMBER or CHIP ID tie off value in the multi-chip mode. For single-chip this is 0.

7.7.1.4 SID_SYSTEM_CFG register

This register specifies the configuration details of the subsystem.

Table 7-15 SID_SYSTEM_CFG register

Bits	Name	Description
[31:0]	CONFIG_NUMBER	System specific

7.7.1.5 SID_PID registers

The SID Peripheral ID registers set are listed in the following tables.

Table 7-16 SID_PID_0 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	PART_0	Bits [7:0] of part number. Set to 0x44.

Table 7-17 SID_PID_1 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:4]	DES_0	Bits [3:0] of JEP Identity. Set to 0xB for Arm.
[3:0]	PART_1	Bits [11:8] of part number. Set to 0x8.

Table 7-18 SID_PID_2 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:4]	REVISION	Set to 0x0 for r0p0.

Bits	Name	Description
[3]	JEDEC	Set to 0x1.
[2:0]	DES_1	Bits [6:4] of Designer field. Set to 0x3 for Arm.

Table 7-19 SID PID_3 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	-	Reserved, Read-As-Zero.

Table 7-20 SID PID_4 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:4]	SIZE	Indicates the log2 of the number of 4KB blocks occupied by the interface. Set to 0x0.
[3:0]	DES_2	JEP106 continuation code identifies the designer. Set to 0x4 for Arm.

7.7.1.6 SID COMP_ID registers

The SID component ID registers set are listed in the following tables.

Table 7-21 SID COMP_ID0 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID0	Reads as 0x0D.

Table 7-22 SID COMP_ID1 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID0	Reads as 0xF0.

Table 7-23 SID COMP_ID2 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID0	Reads as 0x05.

Table 7-24 SID COMP_ID3 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID0	Reads as 0xB1.

7.7.2 REFCLK counter

The REFCLK counter is an implementation of the memory mapped counter defined by the Arm® *Architecture Reference Manual Armv8, for Armv8-A architecture profile*. It implements both the standard REFCLK CNTControl frame and the REFCLK CNTRead frame in the AP memory map. Their base address and accessibility is defined in 7.1.2 *Peripherals region*.

This counter implements three additional registers in the CNTControl frame that are not defined in the Arm® *Architecture Reference Manual Armv8, for Armv8-A architecture profile*. Table 7-25 list these three registers and their offset address in the CNT Control frame.

Table 7-25 Additional REFCLK CNT control registers

Offset	Name	Type	Default	Description
0xC0	CNTSCR	R/W	0x00	Counter Synchronization Control Register.
0xC4	CNTSVL	RO	0x00	Counter Synchronized Counter Lower Value Register.
0xC8	CNTSVU	RO	0x00	Counter Synchronized Counter Upper Value Register.

7.7.2.1 CNTSCR register

Counter synchronization control register (CNTSCR) controls how the enable/disable of the REFCLK generic counter is performed.

Table 7-26 CNTSCR register

Bits	Name	Description
[31:1]	-	Reserved, RAZ/WI
[0]	ENSYNC	Controls the way the Counter Control register EN bit operates: 0 – The counter is enabled or disabled immediately. 1 – The enabling of the counter is delayed until just after the next rising edge at the REFCLK. Disabling the counter is not delayed.

7.7.2.2 CNTSVL register

CNTControl synchronized counter value LOW (CNTSVL) register is used to read back value of the counter sampled on the rising edge of REFCLK. This register returns the lower word CNTSV[31:0].

Table 7-27 CNTSVL register

Bits	Name	Description
[31:0]	CNTSVL	REFCLK-sampled value of the counter, lower word CNTSV[31:0]

7.7.2.3 CNTSVU register

CNTControl synchronized counter value HIGH (CNTSVU) register is used to read back value of the counter sampled on the rising of edge of. This register returns the upper word CNTSV[63:32].

Table 7-28 CNTSVU register

Bits	Name	Description
[31:0]	CNTSVU	REFCLK-sampled value of the counter, upper word CNTSV[63:32]

7.7.3 Generic timers

The RD-V1 design supports various timer frames. Refer to the AP/SCP/MCP memory maps for their respective timer base addresses.

- SCP_REFCLK Generic timer is mapped in the SCP memory map.
- SCP_REFCLK Generic timer is mapped in the MCP memory map.
- Pn_REFCLK per Processor Generic wake up Timers for n=0 to NUM_CLUSTERS, described in 4.4.2.3 *Pn_REFCLK per cluster Generic Timer* for Secure use by the applications processors and the SCP used for waking up the cores when SYSTOP is powered off. These are mapped in the SCP and AP memory maps. Refer to the SCP peripheral memory map and the AP memory map.
- Two additional REFCLK Generic Timers, 1 Secure and 1 Non-Secure for use solely by the applications processors for general purposes. Refer to the AP memory map.
- Global timer synchronization timer is mapped in the SCP memory map.

7.7.3.1 Generic timer register

Table 7-29 shows the generic timer register summary.

Table 7-29 Generic timer register summary table

Offset	Name	Type	Default	Description
0x00	CNTPCT[31:0]	RO	N/A	Physical Count Register
0x04	CNTPCT[63:32]	RO	N/A	Physical Count Register
0x08	CNTVCT[31:0]	RO	N/A	Virtual Count Register
0x0C	CNTVCT[63:32]	RO	N/A	Virtual Count Register
0x10	CNTRFQ	RW	N/A	Counter Frequency Register

Offset	Name	Type	Default	Description
0x14	CNTPLOACR	RW	0x0	Counter PLO Access Control Register
0x18	CNTVOFF[31:0]	RO	N/A	Virtual Offset Register
0x1C	CNTVOFF[63:32]	RO	N/A	Virtual Offset Register
0x20	CNTP_CVAL[31:0]	RW	0x0	Physical Timer CompareValue Register
0x24	CNTP_CVAL[63:32]	RW	0x0	Physical Timer CompareValue Register
0x28	CNTP_TVAL	RW	0x0	Physical Timer Value Register
0x2C	CNTP_CTL	RW	0x0	Physical Timer Control Register
0x30	CNTV_CVAL[31:0]	RW	0x0	Virtual Timer Compare Value Register
0x34	CNTV_CVAL[63:32]	RW	0x0	Virtual Timer Compare Value Register
0x38	CNTV_TVAL	RW	0x0	Virtual Timer Value Register
0x3C	CNTV_CTL	RW	0x0	Virtual Timer Control Register
0x40 – 0xFCC	Reserved, UNK/SBZP	-	-	-
0xFD0	CNTPIDR4	RO	0x04	Peripheral Identification Register
0xFD4 – 0xFDC	Reserved UNK/SBZP	-	-	-
0xFE0	CNTPIDR0	RO	0x0E	Peripheral Identification Register
0xFE4	CNTPIDR1	RO	0xB0	Peripheral Identification Register
0xFE8	CNTPIDR2	RO	0x0B	Peripheral Identification Register
0xFEC	CNTPIDR3	RO	0x00	Peripheral Identification Register
0xFF0	CNTCIDR0	RO	0x0D	Component Identification Register
0xFF4	CNTCIDR1	RO	0xF0	Component Identification Register
0xFF8	CNTCIDR2	RO	0x05	Component Identification Register
0xFFC	CNTCIDR3	RO	0xB1	Component Identification Register

7.7.3.1.1 Peripheral ID registers (CNTPIDRn)

The following tables show the various peripheral ID registers of RD-V1.

Table 7-30 Peripheral ID register 4 (CNTPIDR4)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:4]	SIZE	4KB Count
[3:0]	DES_2	JEP106 continuation code

Table 7-31 Peripheral ID register 3 (CNTPIDR3)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:4]	REVAND	Manufacturer revision number
[3:0]	CMOD	Incremented on authorized customer modification

Table 7-32 Peripheral ID register 2 (CNTPIDR2)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:4]	REVISION	Revision (set to 0 for rOp0)
[3]	JEDEC	'1' required
[2:0]	DES_1	JEP106 identity code [6:4]

Table 7-33 Peripheral ID register 1 (CNTPIDR1)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:4]	DES_0	JEP106 identity code [3:0]
[3:0]	PART_1	Part number [11:8]

Table 7-34 Peripheral ID register 0 (CNTPIDR0)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:0]	PART_0	Part number [7:0]

7.7.3.1.2 Component ID registers (CNTCIDRn)

The following tables show the various component ID registers of RD-V1.

Table 7-35 Component ID Register 0 (CNTCIDR0)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:0]	PRMBL_0	CNTCIDR[7:0], default: 0x0D

Table 7-36 Component ID Register 1 (CNTCIDR1)

Bits	Name	Description
[31:8]	RAZ	Reserved
[7:4]	CLASS	CNTCIDR[15:12]
[3:0]	PRMBL_1	CNTCIDR[11:8]

Table 7-37 Component ID Register 2 (CNTCIDR2)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:0]	PRMBL_2	CNTCIDR[23:16], default: 0x05

Table 7-38 Component ID Register 3 (CNTCIDR3)

Bits	Name	Description
[31:8]	Reserved	Reserved, RAZ/WI
[7:0]	PRMBL_3	CNTCIDR[31:24], default: 0xB1

7.7.3.2 Global timer synchronization register

These registers are defined for global timer synchronization across multiple chips.

Table 7-39 SYSCNT_SYNC_CTRL<x> and CSCNT_SYNC_CTRL<x> register summary

Offset	Name	Type	Default	Description
0x0000	MST_GCNT_SYNC_CTRL	RW	0x0	Master Generic counter synchronization control register
0x0004	SLVCHIP_GCNT_SYNC_CTRL	RW	0x0	Slave Generic counter synchronization control register
0x0008	SLVCHIP_GCNT_OFFSET_THRESHOLD	RW	0x0	Offset threshold registers for each slave chip
0x000C	SLVCHIP_GCNT_INT_STATUS	WC	0x0	Interrupt status register
0x0010	GCNT_TIMEOUT	RW	0x0	Time out register – Used for phase of the synchronization process
0x0014	SLVCHIP_GCNT_SYNC_INTERVAL	RW	0x0	Time interval to perform synchronization process
0x0018	SLVCHIP_GCNT_UPDT_TVALUE_I	R	0x0	Time value sent from the Master Counter
0x001C	SLVCHIP_GCNT_UPDT_TVALUE_U	R	0x0	Time value sent from the Master Counter
0x0020	GCNT_SYNC_STATUS	RW	0x0	Status of Current Sync process
0x0024	SLVCHIP_GCNT_NW_DLY	RW	0x0	Network delay through CCIX from master to slave
0x0FD0	PID4	PID4	0x00000004	Peripheral ID 4
0x0FE0	PID0	PID0	0x00000098	Peripheral ID 0
0x0FE4	PID1	PID1	0x000000B0	Peripheral ID 1
0x0FE8	PID2	PID2	0x0000001B	Peripheral ID 2
0x0FEC	PID3	PID3	0x00000000	Peripheral ID 3
0x0FF0	COMPID0	COMPID0	0x0000000D	Component ID 0
0x0FF4	COMPID1	COMPID1	0x000000F0	Component ID 1

Offset	Name	Type	Default	Description
0x0FF8	COMPID2	COMPID2	0x00000005	Component ID 2
0x0FFC	COMPID3	COMPID3	0x000000B1	Component ID 3

7.7.3.2.1 MST_GCNT_SYNC_CTRL register

Table 7-40 MST_GCNT_SYNC_CTRL register

Bits	Name	Description
[31:2]	-	Reserved.
[1]	EN_SYNC_IMM	Enables immediate synchronization process.
[0]	EN	Enable sync process.

7.7.3.2.2 SLVCHIP_GCNT_SYNC_CTRL register

Table 7-41 SLVCHIP_GCNT_SYNC_CTRL register

Bits	Name	Description
[31:2]	-	Reserved.
[1]	EN_SYNC_IMM	Enables immediate synchronization process.
[0]	EN	Enable sync process.

7.7.3.2.3 SLVCHIP_GCNT_OFFSE_THRESHOLD register

Table 7-42 SLVCHIP_GCNT_OFFSE_THRESHOLD register

Bits	Name	Description
[31:0]	THRSHLD_VAL	Threshold value to be compared.

7.7.3.2.4 SLVCHIP_GCNT_INT_STATUS register

Table 7-43 SLVCHIP_GCNT_INT_STATUS register

Bits	Name	Description
[31:1]	RSVD	Reserved
[0]	Interrupt	Interrupt status of sync process

7.7.3.2.5 GCNT_TIMEOUT register

Table 7-44 GCNT_TIMEOUT register

Bits	Name	Description
[31:0]	TIMEOUT_VAL	Synchronization timeout value

7.7.3.2.6 SLVCHIP_GCNT_SYNC_INTERVAL register

Table 7-45 SLVCHIP_GCNT_SYNC_INTERVAL register

Bits	Name	Description
[31:0]	SYNC_INTRVL	Interval time to start the next sync process

7.7.3.2.7 SLVCHIP_GCNT_UPDT_VAL_L register

Table 7-46 SLVCHIP_GCNT_UPDT_VAL_L register

Bits	Name	Description
[31:0]	TUPDT_L	Lower 32 bits of T update value

7.7.3.2.8 SLVCHIP_GCNT_UPDT_VAL_U register

Table 7-47 SLVCHIP_GCNT_UPDT_VAL_U register

Bits	Name	Description
[31:0]	TUPDT_U	Upper 32 bits of T update value

7.7.3.2.9 SLVCHIP_GCNT_NW_DLY register

Table 7-48 SLVCHIP_GCNT_NW_DLY register

Bits	Name	Description
[31:0]	NW_DLY	Network delay value between master and slave using CCIX

7.7.3.2.10 GCNT_SYNC_STATUS register

Table 7-49 GCNT_SYNC_STATUS register

Bits	Name	Description
[31:2]	RSVD	Reserved
[1:0]	Status	Indicates the status of current synchronization process: 00: never run 01: sync in progress 10: error in sync process 11: successful sync

7.7.3.2.11 PID_0 register

Table 7-50 PID_0 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	PART_0	Bits [7:0] of part number. Set to 0x98.

7.7.3.2.12 PID_1 register

Table 7-51 PID_1 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:4]	DES_0	Bits [3:0] of JEP Identity. Set to 0xB for Arm.
[3:0]	PART_1	Bits [11:8] of part number. Set to 0x0.

7.7.3.2.13 PID_2 register

Table 7-52 PID_2 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:4]	REVISION	Set to 0x0 for r0p0.
[3]	JEDEC	Set to 0x1.
[2:0]	DES_1	Bits [6:4] of Designer field. Set to 0x3 for Arm.

7.7.3.2.14 PID_3 register

Table 7-53 PID_3 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	-	Reserved, Read-As-Zero.

7.7.3.2.15 COMP_ID0 register

Table 7-54 COMP_ID0 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID0	Reads as 0x0D.

7.7.3.2.16 COMP_ID1 register

Table 7-55 COMP_ID1 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID1	Reads as 0xF0.

7.7.3.2.17 COMP_ID2 register

Table 7-56 COMP_ID2 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID2	Reads as 0x05.

7.7.3.2.18 COMP_ID3 register

Table 7-57 COMP_ID3 register

Bits	Name	Description
[31:8]	-	Reserved, Read-As-Zero.
[7:0]	COMP_ID3	Reads as 0xB1.

7.7.4 CPU manager and power control registers

The following section describes the CPU manager and power control registers for each core.

Table 7-58 shows the CPU manager and power control registers summary.

Table 7-58 CPU manager and power control register summary

Bits	Name	Type	Reset	Width	Description
0x0000	CLUSTER_CONFIG	RW	0x0	32	Static Config for Global features
0x0004 - 0x000C	-	RO	0x0	-	RAZ/WI
0x0010 - 0x010C	PE Configuration	RW	-	32	Processing element Configuration
0x0110 - 0x01FC	-	RO	0x0	-	RAZ/WI
0x0200 - 0x03FC	-	RO	0x0	-	RAZ/WI
0x0400 - 0x07FC	-	RO	0x0	-	RAZ/WI
0x0800	PPUCLK_CTRL	RW	0x00000001	32	PPU Clock Control register
0x0804	PPUCLK_DIV1	RW	0x0000001F	32	PPU Clock Divider Control register
0x0808 - 0x080C	-	RO	-	32	RAZ/WI
0x0810	PCLK_CTRL	RW	0x000000F0	32	PCLK Control
0x0814 - 0x081C	-	RO	-	32	RAZ/WI
0x0820	DBGCLK_CTRL	RW	0x00000001	32	DBGCLK Control
0x0824	DBGCLK_DIV1	RW	0x0000001F	32	DBGCLK Divider Control register
0x0828 - 0x082C	-	RO	-	32	RAZ/WI
0x0830	GICCLK_CTRL	RW	0x000000F0	32	GICCLK Control
0x0834 - 0x083C	-	RO	-	32	RAZ/WI
0x0840	AMBACLK_CTRL	RW	0x00000000	32	AMBACLK Control
0x0844 - 0x084C	-	RO	-	32	RAZ/WI
0x0850	CLUSCLK_CTRL	RW	0x00000001	32	Cluster Clock Control register
0x0854	CLUSCLK_DIV1	RW	0x0000001F	32	Cluster Clock Divider Control register
0x0858 - 0x085C	-	RO	-	32	RAZ/WI
0x0860	CORE0CLK_CTRL	RW	0x0000_0101	32	CORE 0 Clock Control
0x0864	CORE0CLK_DIV1	RW	0x001F_001F	32	CORE 0 Clock Divider 1

Bits	Name	Type	Reset	Width	Description
0x0868	-	RO	-	32	RAZ/WI
0x086C	CORE0CLK_MOD1	RW	0x0101_0101	32	CORE 0 Clock Modulator 1
0x0870 - 0x09FC	-	RO	-	32	RAZ/WI
0x0A00	CLKFORCE_STATUS	RO	-	32	Clock Force Status register
0x0A04	CLKFORCE_SET	WO	-	32	Clock Force Set register
0x0A08	CLKFORCE_CLR	WO	-	32	Clock Force Clear register
0x0A0C - 0x0FB0	-	RO	0x0	-	RAZ/WI
0x0B00	nERRIQ_INT_STATUS	RO	-	32	nERRIQ interrupt status register
0x0B04	nFAULTIRQ_INT_STATUS	RO	-	32	nFAULT interrupt status register
0x0B08 - 0x0FB0	-	RO	0x0	-	RAZ/WI
0x0FB4	CAP3	RO	-	32	Capabilities 3 register
0x0FB8	CAP2	RO	-	32	Capabilities 2 register
0x0FBC	CAP	RO	-	32	Capabilities register
0x0FC0	CONFIG	RO	-	32	Configuration register. Value dependent upon chosen configuration.
0x0FC4 - 0x0FCC	-	RO	0x0	-	RAZ/WI
0x0FD0	PID4	RO	0x44	32	Peripheral ID 4 register
0x0FD4	PID5	RO	0x00	32	Peripheral ID 5 register
0x0FD8	PID6	RO	0x00	32	Peripheral ID 6 register
0x0FDC	PID7	RO	0x00	32	Peripheral ID 7 register
0x0FE0	PID0	RO	0xB8	32	Peripheral ID 0 register
0x0FE4	PID1	RO	0xB0	32	Peripheral ID 1 register
0x0FE8	PID2	RO	0x0B	32	Peripheral ID 2 register
0x0FEC	PID3	RO	0x00	32	Peripheral ID 3 register
0x0FF0	ID0	RO	0x0D	32	Component ID 0 register
0x0FF4	ID1	RO	0xF0	32	Component ID 1 register
0x0FF8	ID2	RO	0x05	32	Component ID 2 register
0x0FFC	ID3	RO	0xB1	32	Component ID 3 register
0x1000 - 0x1FFC	CLUS-PPU0	-	-	-	Cluster PPU Configuration dependent registers. Always implemented.

7.7.4.1.1 CLUSTER_CONFIG register

Table 7-59 CLUSTER_CONFIG register

Bits	Name	Description
[31:2]	-	Reserved
[1]	ELADISABLE	Disables Embedded Logic Analyzer hardware inside the Corinth debug block. This pin is only sampled during reset of the cluster debug logic.
[0]	CRYPTODISABLE	Disables Cryptographic Extensions. This pin is only sampled during reset of the processor.

7.7.4.1.2 PE configuration registers

The PE Config region is split into one section for each Processing Element implemented within the cluster. Each section is made up of four 32-bit registers arranged, as shown in Table 7-60.

Table 7-60 PE configuration register sets

Offset	Name	Type	Reset	Description
0x0	PE Configuration X	RW	0x8	"X" is between 0 to Max number of PE
0x4	RVBARADDRx_LW	RW	0x0	-
0x8	RVBARADDRx_UP	RW	0x0	-
0xC	Reserved	-	-	-

7.7.4.1.3 PE configuration core<0-7> registers

This register, and the CPUPPMCR_EL3 register programming in the Arm® Neoverse™ V1 core, define the core/cluster power configuration. It has both static and dynamic programming register fields in it.

The register bit fields [3:0] are static under a normal circumstance, SCP firmware programs these values at boot and during runtime.

The PPM power control register bit fields [10:4] are programmed during the run time dynamically.

Table 7-61 PE configuration core<0-7> registers

Bits	Name	Description
[31:11]	-	Reserved
[10]	PPMCTLn [3]	PDP_EN - Performance Defined Power (PDP) enable bit PPMCTLn[3] = 1'b1, PDP Enabled
[9:7]	PPMCTLn [2:0]	DT_THR [2:0], Dispatch Throttling Setpoint Default: PPMCTLn[2:0] = 3'b110, DT Threshold = 50%

Bits	Name	Description
[6:5]	MPMMSTATEn [1:0]	MPMM_SEL[1:0], MPMM "Gear" select This is an external control signal that selects the current "Gear" – the MXP_ATHR and MXP_TP configuration values that are used to set the current threshold and throttle percentage. These are controlled by the SCP core during run time. Default: MPMMSTATEn[1:0] = 2'b00, MPMM Gear 0 Active
[4]	MPMMEN[n]	MPMM_EN, Master MPMM enable. Enable bit for max-power throttling mechanism of each processor. Default: MPMMEN[n] = 1'b1, MPMM Enabled
[3]	AA64nAA32	Enable 32-bit cold boot for each PE. This pin is only sampled during a power on reset of the processor.
[2]	VINITHI	Enable high exception vectors when booting in 32-bit state for each PE. This pin is only sampled during reset of the processor.
[1]	CFGTE	Individual processor control of the default exception handling state. This control is only sampled by the processor during reset.
[0]	CFGEND	Individual processor control of the endianness configuration at reset. This control is only sampled by the processor during reset.

7.7.4.1.4 RVBARADDR<0-7>_LW registers

Table 7-62 RVBARADDR<0-7>_LW registers

Bits	Name	Description
[31:2]	RVBAR	Provides bits [31:2] of the reset value of the RVBAR_EL3 register in the PE. This value is used as the address that execution starts from when the processor is in 64-bit state. All other bits of RVBAR_EL3 are zero. This value is used to drive bits [31:2] of all the RVBARADDRx inputs to the PE. x = 0, 1, 2, 3... These pins are only sampled during reset of the processor.
[1:0]	-	Reserved

7.7.4.1.5 RVBARADDR<0-7>_UP registers

Table 7-63 RVBARADDR<0-7>_UP registers

Bits	Name	Description
[31:2]	-	Reserved

Bits	Name	Description
[11:0]	RVBAR	<p>Provides bits [43:32] of the reset value of the RVBAR_EL3 register in the PE. This value is used as the address that execution starts from when the processor is in 64-bit state. All other bits of RVBAR_EL3 are zero.</p> <p>This value is used to drive bits [43:32] of all the RVBARADDR_x inputs to the PE.</p> <p>x = 0, 1, 2, 3...</p> <p>These pins are only sampled during reset of the processor.</p>

7.7.4.1.6 PPUCLK_CTRL, DBGCLK_CTRL registers

Table 7-64 PPUCLK_CTL, DBGCLK_CTRL registers

Bits	Name	Description
[31:24]	ENTRY_DLY	<p>Number of clock cycles between the clock not being required and the request to dynamically clock gate it.</p> <p>0x0 – No cycles</p> <p>0x1 – 1 cycle</p> <p>...</p> <p>0xFF – 255 cycles</p>
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	<p>Acknowledges the currently selected clock source.</p> <p>0000_0000 – Clock Gated</p> <p>0000_0001 – REFCLK</p> <p>0000_0010 – SYSPLLCLK</p> <p>Other values are reserved.</p>
[7:0]	CLKSELECT	<p>Selects the clock source.</p> <p>0000_0000 – Clock Gated</p> <p>0000_0001 – REFCLK</p> <p>0000_0010 – SYSPLLCLK</p> <p>Other values are Reserved. The result of writing one of the Reserved values into this field is UNPREDICTABLE.</p>

7.7.4.1.7 PPUCLK_DIV1, DBGCLK_DIV1 registers

Table 7-65 PPUCLK_DIV1, DBGCLK_DIV1 registers

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	<p>Acknowledges the currently selected clock divider value for SYSPLL.</p> <p>The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1.</p>
[15:5]	-	Reserved

Bits	Name	Description
[4:0]	CLKDIV	Requests a new clock divider value for the SYSPLL. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1

7.7.4.1.8 PCLK_CTRL, GICCLK_CTRL registers

Table 7-66 PCLK_CTRL, GICCLK_CTRL registers

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles
[23:16]	-	Reserved
[15:12]	CLKDIV_CUR	Acknowledges the currently active clock divider value. Divider value is the value + 1. E.g. Setting of 0 indicates divider value of 1.
[11:8]	-	Reserved
[7:4]	CLKDIV	Requests new clock divider value. Divider value is the value + 1. For example Setting of 0 indicates divider value of 1.
[3:0]	-	Reserved

7.7.4.1.9 AMBACLK_CTRL register

Table 7-67 AMBACLK_CTRL register

Bits	Name	Description
[31:16]	-	Reserved
[15:12]	CLKDIV_CUR	Acknowledges the currently active clock divider value. Divider value is the value + 1. For example Setting of 0 indicates divider value of 1.
[11:8]	-	Reserved
[7:4]	CLKDIV	Requests new clock divider value. Divider value is the value + 1. For example, setting of 0 indicates divider value of 1.
[3:0]	-	Reserved

7.7.4.1.10 CLUSCLK_CTRL register

Table 7-68 CLUSCLK_CTRL register

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – PLL0 0000_0011 – PLL1 0000_0100 – PLL2 0000_0101 – PLL3 0000_0110 – PLL4 0000_0111 – PLL5 0000_1000 – PLL6 0000_1001 – PLL7 0000_1010 – SYSPLLCLK Other values are reserved.
[7:0]	CLKSELECT	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – PLL0 0000_0011 – PLL1 0000_0100 – PLL2 0000_0101 – PLL3 0000_0110 – PLL4 0000_0111 – PLL5 0000_1000 – PLL6 0000_1001 – PLL7 0000_1010 – SYSPLLCLK Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.

When CAP register CLUSSYNC is 1, the CLKSELECT field becomes Read-Only and is read back as 0x01. This indicates that the selected clock is the clock of the interconnect.

7.7.4.1.11 CLUSCLK_DIV1 register

Table 7-69 CLUSCLK_DIV1 register

Bits	Name	Description
[31:16]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for SYSPLLCLK. The divider value is the value of CLKDIV_CUR + 1. For example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	Requests a new clock divider value for the SYSPLL. The divider value is the value of CLKDIV + 1. For example, setting a value of 0 indicates a divider value of 1.

When CAP register CLUSSYNC is 1, this register is read-only and reads as 0x0000_0000.

7.7.4.1.12 CORE0CLK_CTRL register

Table 7-70 CORE0CLK_CTRL register

Bits	Name	Description
[31:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – PLL0 0000_0011 – PLL1 0000_0100 – PLL2 0000_0101 – PLL3 0000_0110 – PLL4 0000_0111 – PLL5 0000_1000 – PLL6 0000_1001 – PLL7 Other values reserved.

Bits	Name	Description
[7:0]	CLKSELECT	<p>Selects the clock source.</p> <p>0000_0000 – Clock Gated</p> <p>0000_0001 – REFCLK</p> <p>0000_0010 – PLL0</p> <p>0000_0011 – PLL1</p> <p>0000_0100 – PLL2</p> <p>0000_0101 – PLL3</p> <p>0000_0110 – PLL4</p> <p>0000_0111 – PLL5</p> <p>0000_1000 – PLL6</p> <p>0000_1001 – PLL7</p> <p>Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.</p>

When CAP register CORE0SYNC is 1, the CLKSELECT field is read-only and reads as 0x01. This indicates that the CORECLK is driven by CLUSCLK.

7.7.4.1.13 CORE0CLK_DIV1 register

Table 7-71 CORE0CLK_DIV1 register

Bits	Name	Description
[31:16]	-	Reserved
[20:16]	CLKDIV_CUR	<p>Current value of the integer divider applied to clock selected by CORExCLK.CLKSELECT.</p> <p>0 – Divide by 1</p> <p>1 – Divide by 2</p> <p>....</p> <p>1F – Divide by 32</p>
[15:5]	-	Reserved
[4:0]	CLKDIV	<p>Select the value of the integer divider applied to clock selected by CORExCLK.CLKSELECT.</p> <p>0 – Divide by 1</p> <p>1 – Divide by 2</p> <p>....</p> <p>1F – Divide by 32</p>

When CAP register CORE0SYNC is 1, this register is read-only and reads as 0x0000_0000.

7.7.4.1.14 CORE0CLK_MOD1 register

Table 7-72 CORE0CLK_MOD1 register

Bits	Name	Description
[31:24]	CLKMOD_NUM_CUR	Current value of the clock modulator numerator.
[23:16]	CLKMOD_DEN_CUR	Current value of the clock modulator denominator.

Bits	Name	Description
[15:8]	CLKMOD_NUM	Clock modulator numerator. A value of 0 will cause the clock to be disabled for all cycles of the input clock.
[7:0]	CLKMOD_DEN	Clock modulator denominator. A value of 0 is reserved, and the result of writing a 0 to this field is UNPREDICTABLE. If the numerical value of CLKMOD_DEN is smaller than the numerical value of CLKMOD_NUM, then that results in the clock being enabled for all clock cycles.

7.7.4.1.15 CLKFORCE_STATUS register

If a bit reads back as 1, then the associated dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-73 CLKFORCE_STATUS register

Bits	Name	Description
[31:13]	-	Reserved
[9:2]	-	Reserved
[5]	CLUSCLKFORCE	Clock force for DSU cluster clock
[4]	-	Reserved
[3]	GICCLKFORCE	Clock force for GICCLK
[2]	DBGCLKFORCE	Clock force for DBGCLK
[1]	PCLKFORCE	Clock force for PCLK
[0]	PPUCLKFORCE	Clock force for PPUCLK

7.7.4.1.16 CLKFORCE_SET register

The bit allocation is the same as the CLKFORCE_STATUS register. Writing a 1 to a bit within the CLKFORCE_STATUS register disables any dynamic hardware clock gating, whilst writing 0 to a bit is ignored.

7.7.4.1.17 nERRIRQ_INT_STATUS register

Table 7-74 nERRIRQ_INT_STATUS register

Bits	Name	Description
[31:N+1]	-	Reserved
[N:0]	nERRIRQ	Bit [0] – Cluster nERRIRQ Bit [1] to Bit [N] – Core nERRIRQ Each bit indicates the status of the nERRIRQ output 0 – Interrupt asserted 1 – Interrupt not asserted Where "N" is Number of CPU cores

7.7.4.1.18 nFAULTIRQ_INT_STATUS register

Table 7-75 nFAULTIRQ_INT_STATUS register

Bits	Name	Description
[31:N+1]	-	Reserved
[N:0]	nFAULTIRQ	Bit [0] – Cluster nFAULTIRQ Bit [1] to Bit [N] – Core nFAULTIRQ Each bit indicates the status of the nFAULTIRQ output 0 – Interrupt asserted 1 – Interrupt not asserted Where "N" is Number of CPU cores

7.7.4.1.19 CAP3 register

Table 7-76 CAP3 register

Bits	Name	Description
[31:8]	-	Reserved
[7]	PLL7_IMPLEMENTED	Shows whether PLL7 clock input is implemented in the design. 0 – PLL 7 present 1 – PLL7 not present
[6]	PLL6_IMPLEMENTED	Shows whether PLL6 clock input is implemented in the design. 0 – PLL6 present 1 – PLL6 not present
[5]	PLL5_IMPLEMENTED	Shows whether PLL5 clock input is implemented in the design. 0 – PLL5 present 1 – PLL5 not present
[4]	PLL4_IMPLEMENTED	Shows whether PLL4 clock input is implemented in the design. 0 – PLL4 present 1 – PLL4 not present
[3]	PLL3_IMPLEMENTED	Shows whether PLL3 clock input is implemented in the design. 0 – PLL3 present 1 – PLL3 not present
[2]	PLL2_IMPLEMENTED	Shows whether PLL2 clock input is implemented in the design. 0 – PLL2 present 1 – PLL2 not present
[1]	PLL1_IMPLEMENTED	Shows whether PLL1 clock input is implemented in the design. 0 – PLL1 present 1 – PLL1 not present
[0]	PLLO_IMPLEMENTED	Shows whether PLLO clock input is implemented in the design. 0 – PLLO present 1 – PLLO not present

7.7.4.1.20 CAP2 register

Table 7-77 CAP2 register

Bits	Name	Description
[31:16]	-	Reserved
[15:14]	THREADS_CORE7	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[13:12]	THREADS_CORE6	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[11:10]	THREADS_CORE5	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[9:8]	THREADS_CORE4	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[7:6]	THREADS_CORE3	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[5:4]	THREADS_CORE2	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[3:2]	THREADS_CORE1	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.
[1:0]	THREADS_CORE0	Number of threads in each CPU core. 0 for 1 thread on CPU 1 for 2 threads on CPU Other values are reserved.

7.7.4.1.21 CAP register

Table 7-78 CAP register

Bits	Name	Description
[31:28]	NUM_PE	Number of CPU Processing Elements. 0 for 1 PE 1 for 2 PEs ... 15 for 16 PEs
[27:9]	-	Reserved
[2:8]	CORE<1to7>SYNC	Indicates if the CPU core is synchronous to the cluster Clock. 0 – Core is ASYNC to the cluster 1 – Core is SYNC to the cluster Note: NOT applicable for Direct-connect
[1]	CORE0SYNC	Indicates if the CPU core 0 is synchronous to the Cluster Clock. 0 – Core is ASYNC to the cluster 1 – Core is SYNC to the cluster
[0]	CLUSSYNC	Indicates if the cluster is synchronous to the Interconnect or not. 0 – Cluster is ASYNC to the Interconnect 1 – Cluster is SYNC to the Interconnect

7.7.4.1.22 Config register

Table 7-79 Config register

Bits	Name	Description
[31:16]	ID	ID. This field is set to 0x0013.
[15:4]	-	Reserved
[3:0]	NO_OF_PPU	Defines the number of PPUs in the power control logic. This value is set to Number of cores in a cluster + 1. For Direct connect, it is set to 0x2.

7.7.4.1.23 PID4 register

Table 7-80 PID4 register

Bits	Name	Description
[31:8]	-	Reserved
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as 0x4. This means that the power control logic occupies 64KB address block.

Bits	Name	Description
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer's identity code. These bits read back as 0x4.

7.7.4.1.24 PID5, PID6, PID7 registers

Unused. Register is RAZ/WI.

7.7.4.1.25 PID0 register

Table 7-81 PID0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	part_number_0	These bits read back as 0xB8.

7.7.4.1.26 PID1 register

Table 7-82 PID1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	jep106_id_3_0	JEP106 identity code [3:0].
[3:0]	part_number_1	Part Number. These bits read back as 0x0.

7.7.4.1.27 PID2 register

Table 7-83 PID2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Revision	For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1.
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.7.4.1.28 PID3 register

Table 7-84 PID3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Rev	This is set to 0x0.

Bits	Name	Description
[3:0]	mod_number	This is set to 0x0.

7.7.4.1.29 Component ID 0 register

Table 7-85 Component ID 0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_0	These bits read back as 0x0D.

7.7.4.1.30 Component ID 1 register

Table 7-86 Component ID 1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_1	These bits read back as 0xF0.

7.7.4.1.31 Component ID 2 register

Table 7-87 Component ID 2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_2	These bits read back as 0x05.

7.7.4.1.32 Component ID 3 register

Table 7-88 Component ID 3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_3	These bits read back as 0xB1.

7.7.4.1.33 CLUSx-PPU0 Configuration Dependent Registers

The CLUSx PPU0s are based on the *ARM PPU architecture specification v1.1*.

7.7.5 System PIK

The System PIK occupies a 64KB which is split into 4KB blocks as shown in the following tables:

Table 7-89 System PIK address space table

Address Offset	Name	Description
0x0000-0x0FFF	PIK Control Registers	Clock and pseudo static control signals for SYSTOP Clocks.
0x1000-0x1FFF	SYS-PPU0	Responsible for CMN's Logic P-Channel.
0x2000-0x4FFF	-	Reserved
0x5000-0x5FFF	SYS-PPU1	Responsible for SRAM Q-Channel.
0x6000-0xFFFF	-	Reserved

Table 7-90 System PIK control registers summary

Address Offset	Name	Type	Reset	Width	Description
0x000-0x7FC	-	-	-	-	RAZ/WI
0x800	PPUCLK_CTRL	RW	0x0000_0001	32	System PPU Clock Control register
0x804	PPUCLK_DIV1	RW	0x0000_001F	32	System PPU Clock Divider Control register
0x808-0x81C	-	-	-	-	RAZ/WI
0x820	INTCLK_CTRL	RW	0x0000_0001	32	CMN Interconnect Clock Control register
0x824	INTCLK_DIV1	RW	0x0000_001F	32	CMN Interconnect Clock Divider Control register
0x828-0x82C	-	-	-	-	RAZ/WI
0x830	TCU1CLK_CTRL	RW	0x0000_0001	32	TCU1 Clock Control register
0x834	TCU1CLK_DIV1	RW	0x0000_001F	32	TCU1 Clock Divider Control register
0x838	TCU2CLK_CTRL	RW	0x0000_0001	32	TCU2 Clock Control register
0x83C	TCU2CLK_DIV1	RW	0x0000_001F	32	TCU2 Clock Divider Control register
0x840	TCU3CLK_CTRL	RW	0x0000_0001	32	TCU3 Clock Control register
0x844	TCU3CLK_DIV1	RW	0x0000_001F	32	TCU3 Clock Divider Control register
0x848	TCU4CLK_CTRL	RW	0x0000_0001	32	TCU4 Clock Control register
0x84C	TCU4CLK_DIV1	RW	0x0000_001F	32	TCU4 Clock Divider Control register
0x850	GICCLK_CTRL	RW	0x0000_0001	32	GIC Clock Control register
0x854	GICCLK_DIV1	RW	0x0000_001F	32	GIC Clock Divider Control register
0x858-0x85C	-	-	-	-	RAZ/WI
0x860	PCLKSCP_CTRL	RW	0x0000_0001	32	SCP APB Clock Control register
0x864	PCLKSCP_DIV1	RW	0x0000_001F	32	SCP APB Clock Divider Control register
0x868-0x86C	-	-	-	-	RAZ/WI
0x870	SYSPERCLK_CTRL	RW	0x0000_0001	32	System Peripheral Clock Control register
0x874	SYSPERCLK_DIV1	RW	0x0000_001F	32	System Peripheral Clock Divider Control register
0x878-0x87C	-	-	-	-	RAZ/WI

Address Offset	Name	Type	Reset	Width	Description
0x880	DMCCLK_CTRL	RW	0x0000_0001	32	DMC Clock Control register
0x884	DMCCLK_DIV1	RW	0x0000_001F	32	DMC Clock Divider Control register
0x888-0x88C	-	-	-	-	RAZ/WI
0x890	SYSPCLKDBG_CTRL	RW	0x0000_0001	32	CPU Debug APB Slave port Clock Control register
0x894	SYSPCLKDBG_DIV1	RW	0x0000_001F	32	CPU Debug APB Slave port Clock Divider Control register
0x898-0x89C	-	-	-	-	RAZ/WI
0x8A0	UARTCLK_CTRL	RW	0x0000_0001	32	UART Clock Control register
0x8A4	UARTCLK_DIV1	RW	0x0000_001F	32	UART Clock Divider Control register
0x8A8-0x9FC	-	-	-	-	RAZ/WI
0x8B8-0x9FC	-	-	-	-	RAZ/WI
0xA00	CLKFORCE_STATUS	RO	-	32	Clock Force Status register
0xA04	CLKFORCE_SET	WO	-	32	Clock Force Set register
0xA08	CLKFORCE_CLR	WO	0x0	32	Clock Force Clear register
0xA0C-0xAFC	-	-	-	32	RAZ/WI
0xB08	-	RW	-	-	RAZ/WI
0xB0C	PD_SYSTOP_RST_DLY	RW	0x0000_0018	32	Delay value for SYSTOPRESETn
0xB10-0xFBC	-	-	-	32	RAZ/WI
0xFC0	PIK_CONFIG	RO	0x0021_0002	32	Configuration register
0xFD0	PID4	RO	0x44	8	Peripheral ID 4 register
0xFD4	PID5	RO	0x00	8	Peripheral ID 5 register
0xFD8	PID6	RO	0x00	8	Peripheral ID 6 register
0xFDC	PID7	RO	0x00	8	Peripheral ID 7 register
0xFE0	PID0	RO	0x00	8	Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	8	Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	8	Peripheral ID 2 register
0xFEC	PID3	RO	0x00	8	Peripheral ID 3 register
0xFF0	ID0	RO	0x0D	8	Component ID 0 register
0xFF4	ID1	RO	0xF0	8	Component ID 1 register
0xFF8	ID2	RO	0x05	8	Component ID 2 register
0xFFC	ID3	RO	0xB1	8	Component ID 3 register

7.7.5.1 PPUCLK_CTRL, PCLKSCP_CTRL, TCU<n>CLK_CTRL, SYSPERCLK_CTRL, SYSPCLKDBG_CTRL1, GICCLK_CTRL registers

All these clock control registers share the same register field semantics as shown in the Table 7-91.

Table 7-91 Clock control register

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved.
[7:0]	CLKSELECT	Selects the clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.

7.7.5.2 PPUCLK_DIV1, PCLKSCP_DIV1, TCU<n>CLK_DIV1, SYSPERCLK_DIV1, SYSPCLKDBG_DIV1, GICCLK_DIV1 registers

All these clock divider control registers share the same register field semantics as Table 7-92 shows.

Table 7-92 Clock divider control register

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, <ul style="list-style-type: none"> CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved

Bits	Name	Description
[4:0]	CLKDIV	<p>CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example,</p> <ul style="list-style-type: none"> CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. <p>The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.</p>

7.7.5.3 INTCLK_CTRL register

Table 7-93 INTCLK_CTRL clock control register

Bits	Name	Description
[31:24]	ENTRY_DLY	<p>Number of clock cycles between the clock not being required and the request to dynamically clock gate it.</p> <p>0x0 – No cycles</p> <p>0x1 – 1 cycle</p> <p>...</p> <p>0xFF – 255 cycles</p>
[23:16]	-	Reserved
[15:8]	CLKSELECT_CUR	<p>Acknowledges the currently selected clock source.</p> <p>0000_0000 – Clock Gated</p> <p>0000_0001 – REFCLK</p> <p>0000_0010 – INTPLLCLK</p> <p>Other values are reserved.</p>
[7:0]	CLKSELECT	<p>Selects the clock source.</p> <p>0000_0000 – Clock Gated</p> <p>0000_0001 – REFCLK</p> <p>0000_0010 – INTPLLCLK</p> <p>Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.</p>

7.7.5.4 INTCLK_DIV1 register

Table 7-94 INTCLK_DIV1 register

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	<p>Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example,</p> <ul style="list-style-type: none"> CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. <p>The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.</p>

Bits	Name	Description
[15:5]	-	Reserved
[4:0]	CLKDIV	<p>CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example,</p> <ul style="list-style-type: none"> CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. <p>The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.</p>

7.7.5.5 UARTCLK_CTRL register

Table 7-95 UARTCLK_CTRL clock control register

Bits	Name	Description
[31:16]	-	Reserved
[15:8]	CLKSELECT_CUR	<p>Acknowledges the currently selected clock source.</p> <p>0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved.</p>
[7:0]	CLKSELECT	<p>Selects the clock source.</p> <p>0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.</p>

7.7.5.6 UARTCLK_DIV1 register

Table 7-96 UARTCLK_DIV1 register

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	<p>Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example:</p> <ul style="list-style-type: none"> CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. <p>The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.</p>
[15:5]	-	Reserved
[4:0]	CLKDIV	<p>CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example,</p> <ul style="list-style-type: none"> CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. <p>The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.</p>

7.7.5.7 DMCCLK_CTRL register

Table 7-97 DMCCLK_CTRL register

Bits	Name	Description
[31:24]	ENTRY_DLY	Number of clock cycles between the clock not being required and the request to dynamically clock gate it. 0x0 – No cycles 0x1 – 1 cycle ... 0xFF – 255 cycles
[23:17]	-	Reserved
[16]	DMCCLK_1XCLKBYPASSDIV2	CLKCTRL_DMCCLK_1XCLKBYPASSDIV2: control bit for ClkMux on 1x and 2x clock after the divider to bypass the divider, in the DMC clock selection. Default set to 1'b0. 0 – div2 applied. DMCCLK1x is ½ the frequency of DMCCLK2x. (Default) 1 – div2 bypassed. DMCCLK1x is the same frequency as DMCCLK2x.
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – DDRPLLCLK Other values are reserved.
[7:0]	CLKSELECT	Selects the clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – DDRPLLCLK Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.

7.7.5.8 DMCCLK_DIV1 register

Table 7-98 DMCCLK_DIV1 register

Bits	Name	Description
[31:21]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example: <ul style="list-style-type: none"> CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved

Bits	Name	Description
[4:0]	CLKDIV	<p>The CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example,</p> <ul style="list-style-type: none"> CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010 and so on. <p>The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.</p>

7.7.5.9 CLKFORCE_SET register

This is a Write Only (WO) register. Writing 1 to a bit within the CLKFORCE_SET register disables any dynamic hardware clock gating, while writing 0 to a bit is ignored. The bit allocation is the same as the CLKFORCE_STATUS register.

7.7.5.10 CLKFORCE_CLR register

This is a Write Only (WO) register. Writing 1 to a bit within the CLKFORCE_CLR register enables the dynamic hardware clocking gating, while writing 0 to a bit is ignored. The bit allocation is the same as the CLKFORCE_STATUS register.

7.7.5.11 CLKFORCE_STATUS register

This is a Read Only (RO) register. If a bit reads back as 1 then the associated dynamic clock gating associated with the clock is disabled, otherwise it is enabled.

Table 7-99 CLKFORCE_STATUS register

Bits	Name	Description
[31:13]	-	Reserved
[13]	GICCLKFORCE	Clock force status for GICCLK
[12:9]	TCUCLKFORCE[3:0]	Clock force status for TCUCLK1-TCUCLK4
[8]	SYSPCLKDBGFORCE	Clock force status for SYSPCLKDBG
[7]	DMCCLKFORCE	Clock force status for DMCCLK
[6]	SYSPERCLKFORCE	Clock force status for SYSPERCLKFORCE
[5]	PCLKSCPFORCE	Clock force status for PCLKSCP
[4]	-	Reserved
[3]	-	Reserved
[2]	INTCLKFORCE	Clock force status for INTCLK
[1]	-	Reserved
[0]	PPUCLKFORCE	Clock force status for PPUCLK

7.7.5.12 SYSTOP_RST_DLY register

Table 7-100 SYSTOP_RST_DLY register

Bits	Name	Description
[31:6]	-	Reserved
[5:0]	RST_DLY	Delay value for SYSTOPRESETn. Reset is 6' h18.

7.7.5.13 Config register

Table 7-101 Config register

Bits	Name	Description
[31:16]	-	ID. This field is set to 0x0024.
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of PPUs in the power control logic. This value is set to indicate number of PPUs. The values are dependent on the number PPUs implemented in the subsystem. This reads back as 2.

7.7.5.14 PID4 register

Table 7-102 PID4 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as 0x4. This means that the PIK occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer's identity code. These bits read back as 0x4.

7.7.5.15 PID5 register

Register is RAZ/WI.

7.7.5.16 PID6 register

Register is RAZ/WI.

7.7.5.17 PID7 register

Register is RAZ/WI.

7.7.5.18 PID0 register

Table 7-103 PID0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	part_number_0	These bits read back as 0xB8.

7.7.5.19 PID1 register

Table 7-104 PID1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as 0xB.
[3:0]	part_number_1	Part Number. These bits read back as 0x0.

7.7.5.20 PID2 register

Table 7-105 PID2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Revision	Identifies the revision of the base PIK. For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates that the PIK uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1.
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.7.5.21 PID3 register

Table 7-106 PID3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Rev	This is set to 0x0.
[3:0]	mod_number	This is set to 0x0.

7.7.5.22 Component ID 0 register

Table 7-107 Component ID 0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_0	These bits read back as 0x0D.

7.7.5.23 Component ID 1 register

Table 7-108 Component ID 1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_1	These bits read back as 0xF0.

7.7.5.24 Component ID 2 register

Table 7-109 Component ID 2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_2	These bits read back as 0x05.

7.7.5.25 Component ID 3 register

Table 7-110 Component ID 3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_3	These bits read back as 0xB1.

7.7.5.26 SYS_PPU0, SYS_PPU1 Configuration Dependent Registers

The SYS_PPU0 and SYS_PPU1 are based on the PPU architecture specification v1.1.

7.7.6 Debug PIK

The Debug PIK occupies a 64KB which is split into 4KB blocks as shown in Table 7-111.

Table 7-111 Debug PIK address space

Bits	Name	Description
0x0000-0x0FFF	Debug PIK Control Registers	Clocks and debug control registers
0x1000-0x1FFF	DBGPIK-PPU	Responsible for Debug logic
0x2000-0x4FFF	-	Reserved

Table 7-112 Debug PIK control registers

Address Offset	Name	Type	Reset	Width	Description
0x000	DEBUG_CTRL	RW	0x0	32	Debug Control register
0x004	DEBUG_STATUS	RO	0x0	32	Debug Status register
0x008	DEBUG_CONFIG	RW	0x0	32	Debug Configuration register
0x00C	-	RO	0x0	-	RAZ/WI
0x010	APP_DAP_TARGET_ID	RO	IMPL_DEF	32	Application DAP Target ID register
0x014	SCP_DAP_TARGET_ID	RO	IMPL_DEF	32	SCP DAP Target ID register
0x018	DAP_INSTANCE_ID	RO	IMPL_DEF	32	DAP Instance ID register
0x01C-0x7FC	-	RO	0x0	32	RAZ/WI
0x800-0x80C	-	RO	-	-	RAZ/WI
0x810	TRACECLK_CTRL	RW	0x0000_0001	32	Trace Clock Control register
0x814	TRACECLK_DIV1	RW	0x0000_001F	32	Trace Clock Divider Control register
0x818-0x81C	-	RO	-	-	RAZ/WI
0x820	SYSPCLKDBG_CTRL	RW	0x0000_00F0	32	Debug APB Clock Control register
0x824-0x82C	-	RO	-	-	RAZ/WI
0x830	ATCLKDBG_CTRL	RW	0x0000_0001	32	Debug ATB Clock Control register
0x834	ATCLKDBG_DIV1	RW	0x0000_001F	32	Debug ATB Clock Divider Control register
0x838-0x9FC	-	RO	-	-	RAZ/WI
0xA00-0xFBC	-	RO	0x0	32	RAZ/WI
0xFC0	POWER CONTROL LOGIC_CONFIG	RO	0x0030_0001	32	POWER CONTROL LOGIC configuration register
0xFC4-0xFCC	-	RO	0x0	32	RAZ/WI
0xFD0	PID4	RO	0x44	8	Peripheral ID 4 register
0xFD4	PID5	RO	0x00	8	Peripheral ID 5 register
0xFD8	PID6	RO	0x00	8	Peripheral ID 6 register
0xFDC	PID7	RO	0x00	8	Peripheral ID 7 register

Address Offset	Name	Type	Reset	Width	Description
0xFE0	PID0	RO	0x00	8	Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	8	Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	8	Peripheral ID 2 register
0xFEC	PID3	RO	0x00	8	Peripheral ID 3 register
0xFF0	ID0	RO	0x0D	8	Component ID 0 register
0xFF4	ID1	RO	0xF0	8	Component ID 1 register
0xFF8	ID2	RO	0x05	8	Component ID 2 register
0xFFC	ID3	RO	0xB1	8	Component ID 3 register
0x1000-0x1FFC	DBG-PPU	-	-	-	PPU Configuration dependent registers

7.7.6.1 DEBUG_CTRL register

Table 7-113 DEBUG_CTRL register

Bits	Name	Description
[31:3]	-	Reserved
[2]	CSYSPWRUPACK	Sets the acknowledge (CSYSPWRUPACK) to a system power up request (CSYSPWRUPREQ) from the applications processor DAP.
[1]	CDBGPWRUPACK	Sets the acknowledge (CDBGPWRUPACK) to a debug power up request (CDBGPWRUPREQ) from the applications processor DAP.
[0]	CDBGRSTACK	Sets the acknowledge (CDBGRSTACK) to a debug system reset request (CDBGRSTREQ) from the applications processor DAP.

7.7.6.2 DEBUG_STATUS register

Table 7-114 DEBUG_STATUS register

Bits	Name	Description
[31:3]	-	Reserved
[2]	CSYSPWRUPREQ	Status of CSYSPWRUPREQ signal from the applications processor DAP
[1]	CDBGPWRUPREQ	Status of CDBGPWRUPREQ signal from the applications processor DAP
[0]	CDBGRSTREQ	Status of CDBGRSTREQ signal from the applications processor DAP

7.7.6.3 DEBUG_CONFIG register

Table 7-115 DEBUG_CONFIG register

Bits	Name	Description
[31:1]	-	Reserved
[0]	DBGCONNECTED	Drives the DBGCONNECTED signal inputs into all processor cores/clusters.

7.7.6.4 APP_DAP_TARGET_ID register

Table 7-116 DEBUG_CONFIG register

Bits	Name	Description
[31:0]	TARGETID	Controls the DAP target ID tie-off for multi drop support for the Applications Processor DAP. This value is IMPLEMENTATION DEFINED, but it must conform to the <i>ARM Debug Interface Architecture Specification ADIv5.x to ADIv6</i> .

7.7.6.5 SCP_DAP_TARGET_ID register

Table 7-117 DEBUG_CONFIG register

Bits	Name	Description
[31:0]	TARGETID	Controls the DAP target ID tie-off for multi drop support for the SCP DAP. This value is IMPLEMENTATION DEFINED, but it must conform to the <i>ARM Debug Interface Architecture Specification ADIv5.x to ADIv6</i> .

7.7.6.6 DAP_INSTANCE_ID register

Table 7-118 DAP_INSTANCE_ID register

Bits	Name	Description
[31:4]	-	Reserved
[3:0]	INSTANCEID	Controls the DAP instance ID tie-off for multi drop support. This value is IMPLEMENTATION DEFINED, but it must conform to the <i>ARM Debug Interface Architecture Specification ADIv5.0 to ADIv5.2</i> .

7.7.6.7 ATCLKDBG_CTRL, TRACECLK_CTRL registers

Table 7-119 ATCLKDBG_CTRL, TRACECLK_CTRL registers

Bits	Name	Description
[31:16]	-	Reserved

Bits	Name	Description
[15:8]	CLKSELECT_CUR	Acknowledges the currently selected clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved.
[7:0]	CLKSELECT	Selects the clock source. 0000_0000 – Clock Gated 0000_0001 – REFCLK 0000_0010 – SYSPLLCLK Other values are reserved. The result of writing one of the reserved values into this field is UNPREDICTABLE.

7.7.6.8 ATCLKDBG_DIV1, TRACECLK_DIV1 registers

Table 7-120 ATCLKDBG_DIV1, TRACECLK_DIV1 registers

Bits	Name	Description
[31:16]	-	Reserved
[20:16]	CLKDIV_CUR	Acknowledges the currently selected clock divider value for the clock source selected by bit n. For example, CLK_DIV1 acknowledges the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV_CUR + 1, for example, setting a value of 0 indicates a divider value of 1.
[15:5]	-	Reserved
[4:0]	CLKDIV	The CLK_DIVn register requests a new clock divider value for the clock source selected by bit n. For example: CLK_DIV1 selects the divider value for the clock selected when CLKSELECT is 0000_0010. The divider value is the value of CLKDIV + 1, for example, setting a value of 0 indicates a divider value of 1.

7.7.6.9 SYSPCLKDBG_CTRL register

This register selects the divider ratio for SYSPCLKDBG. The input clock is ATCLKDBG.

Table 7-121 SYSPCLKDBG_CTRL register

Bits	Name	Description
[31:16]	-	Reserved
[15:12]	CLKDIV_CUR	Acknowledges the currently active clock divider value. Divider value is the value + 1. For example, setting of 0 indicates divider value of 1.
[11:8]	-	Reserved

Bits	Name	Description
[7:4]	CLKDIV	Requests new clock divider value. Divider value is the value + 1. For example, setting of 0 indicates divider value of 1.
[3:0]	-	Reserved

7.7.6.10 POWER CONTROL LOGIC_CONFIG register

Table 7-122 POWER CONTROL LOGIC_CONFIG register

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. This field is set to 0x0032.
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of PPU's in the POWER CONTROL LOGIC.

7.7.6.11 PID4 register

Table 7-123 PID4 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as 0x4. This means that the PIK occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer's identity code. These bits read back as 0x4.

7.7.6.12 PID5 register

Register is RAZ/WI.

7.7.6.13 PID6 register

Register is RAZ/WI.

7.7.6.14 PID7 register

Register is RAZ/WI.

7.7.6.15 PID0 register

Table 7-124 PID0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	part_number_0	These bits read back as 0xB8 .

7.7.6.16 PID1 register

Table 7-125 PID1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as 0xB .
[3:0]	part_number_1	Part Number. These bits read back as 0x0 .

7.7.6.17 PID2 register

Table 7-126 PID2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Revision	Identifies the revision of the base PIK. For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates that the POWER CONTROL LOGIC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1 .
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.7.6.18 PID3 register

Table 7-127 PID3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Rev	The top-level RTL provides a 4-bit input, ECOREVNUM, that is normally tied LOW and provides a read value of 0x0. When silicon is available, and if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	mod_number	This is set to 0x0 .

7.7.6.19 Component ID 0 register

Table 7-128 Component ID 0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_0	These bits read back as 0x0D.

7.7.6.20 Component ID 1 register

Table 7-129 Component ID 1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_1	These bits read back as 0xF0.

7.7.6.21 Component ID 2 register

Table 7-130 Component ID 2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_2	These bits read back as 0x05.

7.7.6.22 Component ID 3 register

Table 7-131 Component ID 3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_3	These bits read back as 0xB1.

7.7.7 Debug chain power control logic

Table 7-132 Debug chain Power Control Logic address space

Address Offset	Name	Description
0x0000-0x0FFF	Debug Daisy-chain PIK registers	Debug Daisy-chain PIK registers
0x1000-0x1FFF	DBG Chain0 PPU Control Registers	PPU registers in the DBGCH0 PPU
0x2000-0x2FFF	DBG Chain1 PPU Control Registers	PPU registers in the DBGCH1 PPU

Address Offset	Name	Description
0x3000-0x3FFF	DBG Chain2 PPU Control Registers	PPU registers in the DBGCH2 PPU
0x4000-0x4FFF	DBG Chain3 PPU Control Registers	PPU registers in the DBGCH3 PPU
0x5000-0x5FFF	DBG Chain4 PPU Control Registers	PPU registers in the DBGCH3 PPU
0x6000-0x6FFF	DBG Chain5 PPU Control Registers	PPU registers in the DBGCH1 PPU
0x7000-0x7FFF	DBG Chain6 PPU Control Registers	PPU registers in the DBGCH6 PPU
0x8000-0x8FFF	DBG Chain7 PPU Control Registers	PPU registers in the DBGCH7 PPU

Table 7-133 Debug Daisy-chain PIK registers

Address Offset	Name	Type	Reset	Width	Description
0x000	DBG_CHN_CTRL	RW	0x0	32	Debug Chain Control register
0x004	DBG_CHN_STAT	RO	0x0	32	Debug Chain Status register
0x008-0xAFC	-	RO	0x0	32	RAZ/WI
0xB00	DBG_CHN_PPU_INT_STAT	RO	0x0	32	Debug Chain PPU interrupt status register
0xB04	DBG_CHN_PWRUP_INT_STAT	R/WTC	0x0	32	Debug Chain Power interrupt status register
0xB08-0xFBC	-	RO	-	-	RAZ/WI
0xFC0	POWER CONTROL LOGIC_CONFIG	RO	0x0030_0001	32	POWER CONTROL LOGIC configuration register
0xFD0	PID4	RO	0x44	8	Peripheral ID 4 register
0xFD4	PID5	RO	0x00	8	Peripheral ID 5 register
0xFD8	PID6	RO	0x00	8	Peripheral ID 6 register
0xFDC	PID7	RO	0x00	8	Peripheral ID 7 register
0xFE0	PID0	RO	0x00	8	Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	8	Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	8	Peripheral ID 2 register
0xFEC	PID3	RO	0x00	8	Peripheral ID 3 register
0xFF0	ID0	RO	0x0D	8	Component ID 0 register
0xFF4	ID1	RO	0xF0	8	Component ID 1 register
0xFF8	ID2	RO	0x05	8	Component ID 2 register
0xFFC	ID3	RO	0xB1	8	Component ID 3 register

0x1000-0x1FFC	DBG-PPU	-	-	-	PPU Configuration dependent registers
---------------	---------	---	---	---	---------------------------------------

7.7.7.1 DBG_CHN_CTRL register

Table 7-134 DBG_CHN_CTRL register

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	CDBGPWRUPACK	Sets the acknowledge (CDBGPWRUPACK[x]) to a debug chain power up request (CDBGPWRUPREQ[x]) from the GPR in the applications processor debug logic. N is the number of debug chains implemented.

7.7.7.2 DBG_CHN_STAT register

Table 7-135 DBG_CHN_STAT register

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	CDBGPWRUPREQ	Status of CDBGPWRUPREQ[x] signal from the GPR in the applications processor debug logic. N is the number of debug chains implemented.

7.7.7.3 DBG_CHN_PPU_INT_STAT register

Table 7-136 DBG_CHN_PPU_INT_STAT register

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	PPU_INT	Status of interrupt from PPU for Debug chain N. N is the number of debug chains implemented.

7.7.7.4 DBG_CHN_PWRUP_INT_STAT register

Table 7-137 DBG_CHN_PWRUP_INT_STAT register

Bits	Name	Description
[31:N]	-	Reserved
[N-1:0]	CDBGPWRUPREQ_INT	This bit is set on any edge of CDBGPWRUPREQ signal for any debug chain. Writing a 1 to this bit clears it. N is the number of debug chains implemented.

7.7.7.5 POWER CONTROL LOGIC configuration register

Table 7-138 POWER CONTROL LOGIC configuration register

Bits	Name	Description
[31:16]	-	POWER CONTROL LOGIC_ID. This field is set to 0x0031.
[15:4]	-	Reserved
[3:0]	no_of_ppu	Defines the number of PPU's in the POWER CONTROL LOGIC. This value is set to the number of debug chains implemented.

7.7.7.6 PID4 register

Table 7-139 PID4 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as 0x6. This means that the POWER CONTROL LOGIC occupies 256KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the manufacturer's identity code. These bits read back as 0x4.

7.7.7.7 PID5 register

Register is RAZ/WI.

7.7.7.8 PID6 register

Register is RAZ/WI.

7.7.7.9 PID7 register

Register is RAZ/WI.

7.7.7.10 PID0 register

Table 7-140 PID0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	part_number_0	These bits read back as 0xB8.

7.7.7.11 PID1 register

Table 7-141 PID1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as 0xB.
[3:0]	part_number_1	Part Number. These bits read back as 0x0.

7.7.7.12 PID2 register

Table 7-142 PID2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Revision	Identifies the revision of the POWER CONTROL LOGIC. For revision r0p0, this field is set to 0x0.
[3]	jedec_used	This indicates that the POWER CONTROL LOGIC uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1.
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code.

7.7.7.13 PID3 register

Table 7-143 PID3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Rev	The top-level RTL provides a 4-bit input, ECOREVNUM, that is normally tied LOW and provides a read value of 0x0. When silicon is available, and if metal fixes are necessary, the manufacturer can modify the tie-offs to indicate a revision of the silicon.
[3:0]	mod_number	This is set to 0x0.

7.7.7.14 Component ID 0 register

Table 7-144 Component ID 0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_0	These bits read back as 0x0D.

7.7.7.15 Component ID 1 register

Table 7-145 Component ID 1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_1	These bits read back as 0xF0 .

7.7.7.16 Component ID 2 register

Table 7-146 Component ID 2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_2	These bits read back as 0x05 .

7.7.7.17 Component ID 3 register

Table 7-147 Component ID 3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_3	These bits read back as 0xB1 .

7.7.8 MSCP PIK

The MSCP PIK can be configured as either an instance for the SCP or MCP element. Depending on the configuration the programmers' model presents different registers. MCP programmers' model is a subset of the SCP programmers' model.



Registers that are between the address offset ranges 0xA60 to 0xFBC are used in SCP and NOT used in MCP

Table 7-148 SCP/MCP PIK registers summary

Address Offset	Name	Type	Reset	Width	Description
0x000-0x00C	-	-	-	-	RAZ/WI
0x010	RESET_SYNDROME	RW	0x1	32	Reset Syndrome Register
0x014-0x01C	-	-	-	-	RAZ/WI

Address Offset	Name	Type	Reset	Width	Description
0x020	SURVIVAL_RESET_STATUS	RW	0x0	32	Survival reset status register
0x024-0x02C	-	-	-	-	RAZ/WI
0x030	-	-	-	-	RAZ/WI
0x034	ADDR_TRANS	RW	0x0	32	Address Translation for SCP/MCP for accessing all the application memory space
0x038	DBG_ADDR_TRANS	RW	0x0	32	Address Translation for SCP/MCP for accessing AP debug components
0x3C	-	-	-	-	RAZ/WI
0x040	WS1_TIMER_MATCH	RW	0x0	32	WS1 Timer value
0x044	WS1_TIMER_EN	RW	0x1	32	WS1 Timer enable
0x048-0x1FC	-	-	-	-	RAZ/WI
0x200	SS_RESET_STATUS	RO	0x0	32	Subsystem reset request status
0x204	SS_RESET_SET	WO	0x0	32	Subsystem reset request set
0x208	SS_RESET_CLR	WO	0x0	32	Subsystem reset request clear
0x20C-0x7FC	-	-	-	-	RAZ/WI
0x800-0x80C	-	-	-	-	RAZ/WI
0x810	CORECLK_CTRL	RW	0x1	32	Core Clock Control
0x814	CORECLK_DIV1	RW	0x1	32	Core Clock Divider
0x818-0x81C	-	-	-	-	RAZ/WI
0x820	ACLK_CTRL	RW	0x1	32	AXI Clock Control
0x824	ACLK_DIV1	RW	0x1	32	AXI Clock Divider
0x828-0x9FC	-	-	-	-	RAZ/WI
0xA00-0xA0C	-	-	-	-	RAZ/WI
0xA10-0xA50	PLL_STATUS	RO	0x0	32	PLL Status
0xA54-0xA5F	-	-	-	-	RAZ/WI
0xA60	CONS_MMUTCU_INT_STATUS	-	-	-	Consolidated MMU TCU Interrupt status
0xA64	CONS_MMUTBU_INT_STATUS0	-	-	-	Consolidated MMU TBUs Interrupt status connected to TCU0 and TCU1
0xA68	CONS_MMUTBU_INT_STATUS1	-	-	-	Consolidated MMU TBUs Interrupt status connected to TCU 2 and TCU3

Address Offset	Name	Type	Reset	Width	Description
0xA6C	CONS_MMUTCU_INT_CLR	RO	0x0	32	Clear register to clear the Interrupt status of the interrupts captured in CONS_MMUTCU_INT_STATUS
0xA70	CONS_MMUTBU_INT_CLR0	RO	0x0	32	Clear register to clear the Interrupt status of the interrupts captured in
0xA74	CONS_MMUTBU_INT_CLR1	RO	0x0	32	CONS_MMUTBU_INT_STATUS0
0xA6C-0xAFF	-	-	-	-	Clear register to clear the Interrupt status of the interrupts captured in
0xB00	MHU_NS_INT_STATUS	RO	0x0	32	CONS_MMUTBU_INT_STATUS1
0xB04	MHU_S_INT_STATUS	RO	0x0	32	RAZ/WI
0xB08-0xB1C	-	-	-	-	MHU NS Interrupt Status
0xB20-0xB3C	CPU_PPU_INT_STATUS	RO	0x0	32	MHU S Interrupt Status
0xB40	CLUS_PPU_INT_STATUS	RO	0x0	32	RAZ/WI
0xB44-0xB5C	-	-	-	-	CPU PPU Interrupt Status
0xB60-0xB7C	TIMER_INT_STATUS	RO	0x0	32	Cluster PPU Interrupt Status
0xB80-0xB9C	CPU_PLL_LOCK_STATUS<x>	RW	0x0	32	RAZ/WI
0xBA0-0xBBC	-	RW	0x0	32	Timer Interrupt Status
0xBC0-0xBDC	CPU_PLL_UNLOCK_STATUS<x>	RW	0x0	32	CPU PLL Lock Status
0xBE0-0xBEC	-	RW	0x0	32	RAZ/WI
0xBF0	CLUSTER_PLL_LOCK_STATUS<x>	RW	0x0	32	Cluster PLL Lock Status
0xBF4	CLUSTER_PLL_UNLOCK_STATUS<x>	RW	0x0	32	Cluster PLL UnLock Status
0xBE8-0xBEC	Reserved	RW	0x0	32	RAZ/WI
0xC00	CLUS_FAULT_INT_STATUS	RO	0x0	32	Cluster Fault Interrupt Status
0xC04-0xC20	Reserved	-	-	-	RAZ/WI
0xC30	CLUS_ECCERR_INT_STATUS	RO	0x0	32	Cluster ECC error Interrupt Status
0xC34-0xC50	Reserved	-	-	-	RAZ/WI
0xC54-0xCFC	Reserved	-	-	-	RAZ/WI
0xD00	DMC0_4_INT_STATUS	RO	0x0	32	Status of DMC0 and DMC4 interrupts
0xD04	DMC1_5_INT_STATUS	RO	0x0	32	Status of DMC1 and DMC5 interrupts

Address Offset	Name	Type	Reset	Width	Description
0xD08	DMC2_6_INT_STATUS	RO	0x0	32	Status of DMC2 and DMC6 interrupts
0xD0C	DMC3_7_INT_STATUS	RO	0x0	32	Status of DMC3 and DMC7 interrupts
0xD10 - 0xFBC	Reserved	-	-	-	RAZ/WI
0xFC0	PWR_CTRL_CONFIG	RO	0x0	32	Power Control Config
0xFC4 - 0xFCC	Reserved	-	-	-	RAZ/WI
0xFD0	PID4	RO	0x44	8	Peripheral ID 4 register
0xFD4	PID5	RO	0x00	8	Peripheral ID 5 register
0xFD8	PID6	RO	0x00	8	Peripheral ID 6 register
0xFDC	PID7	RO	0x00	8	Peripheral ID 7 register
0xFE0	PID0	RO	0x00	8	Peripheral ID 0 register
0xFE4	PID1	RO	0xB0	8	Peripheral ID 1 register
0xFE8	PID2	RO	0x0B	8	Peripheral ID 2 register
0xFEC	PID3	RO	0x00	8	Peripheral ID 3 register
0xFF0	ID0	RO	0x0D	8	Component ID 0 register
0xFF4	ID1	RO	0xF0	8	Component ID 1 register
0xFF8	ID2	RO	0x05	8	Component ID 2 register
0xFFC	ID3	RO	0xB1	8	Component ID 3 register

7.7.8.1 SCP / MCP common PIK registers

This section covers ONLY the common registers that are used for both the SCP and MCP PIKs.

7.7.8.1.1 RESET_SYNDROME register

Table 7-149 RESET_SYNDROME register

Bits	Name	Default	Description
[31:6]	-	-	Reserved
[5]	CORE_RST	0x0	Last reset caused by a software request for a Core only reset.
[4]	LOCKUP	0x0	Indicates if core was locked up at last reset.
[3:1]	-	-	Reserved
[0]	PORESETn	0x1	Last reset caused by the PORESETn input.

To clear bit, SW should write 0 to the field.

7.7.8.1.2 SURVIVAL_RESET_STATUS register

Table 7-150 SURVIVAL_RESET_STATUS register

Bits	Name	Default	Description
[31:0]	SURVIVAL	UNK	This register has no defined reset value. The value is retained over a full SoC reset. When a watchdog reset occurs, this register is set to DEADBEEF.

7.7.8.1.3 ADDR_TRANS register

Table 7-151 ADDR_TRANS register

Bits	Name	Default	Description
[31:29]	-	-	Reserved
[28:1]	ADDR_47_20	0x0	Set the value of address bits 47 to 20 for SCP/MCP to access all of the Application memory map.
[0]	ADDR_TRANS_EN	0x0	Selects whether the address translation is enabled. 0 – Address translation is disabled 1 – Address translation is enabled

7.7.8.1.4 DBG_ADDR_TRANS register

Table 7-152 SURVIVAL_RESET_STATUS register

Bits	Name	Default	Description
[31:1]	-	-	Reserved
[0]	EN	0x0	Selects whether the address translation is enabled. 0 – Debug Address translation is disabled 1 – Debug Address translation is enabled to higher address space Translate 0x0_A000_0000 - 0x0_E000_0000

7.7.8.1.5 WS1_TIMER_MATCH register

Table 7-153 WS1_TIMER_MATCH register

Bits	Name	Default	Description
[31:0]	WS1_TIMER_MATCH_VALUE	0x0	Programs the delay applied to the WS1 for the SCP watchdog. When the delay expires, a subsystem reset is requested, and the SURVIVAL register is set to DEADBEEF.

7.7.8.1.6 WS1_TIMER_EN register

Table 7-154 WS1_TIMER_EN register

Bits	Name	Default	Description
[31:1]	-	-	Reserved
[0]	WS1_TIMER_EN		Enables the WS1 Timer. 0 – WS1 Timer disable. The counter will not count down the WS1 request. 1 – WS1 Timer enable. The counter will count down the WS1 request.

The WS1 Timer is loaded when the WS1_TIMER_MATCH register is written to. If the write to the register occurs when the timer is counting down, the count will start counting from the new value. It counts down when both WS1_TIMER_EN and WDOGRESETREQ are high. If either goes low, the count stops counting and resets to the value in WS1_TIMER_MATCH.

7.7.8.1.7 SS_RESET_STATUS/SS_RESET_SET/SS_RESET_CLR register

Table 7-155 SS_RESET_STATUS/SS_RESET_SET/SS_RESET_CLR register

Bits	Name	Default	Description
[31:1]	-	-	Reserved
[0]	SS_RST	0x0	Request a reset of the Subsystem.

In the status register, the current value of the SS_RST can be read back.

To set the bit, software must write a 1 to the SS_RST field in the SET register. To clear the bit, software must write a 1 to the SS_RST field in the CLR register.

7.7.8.1.8 CORECLK_CTRL register

Table 7-156 CORECLK_CTRL register

Bits	Name	Default	Description
[31:16]	-	-	Reserved
[15:8]	CLKSELECT_CUR	-	Acknowledges the currently selected clock source: 0x0 – Reserved 0x1 – REFCLK 0x2 – SYSPLL Other values are reserved.
[7:0]	CLKSELECT	0x1	Select current clock source: 0x0 – Reserved 0x1 – REFCLK 0x2 – SYSPLL Other values are reserved.

7.7.8.1.9 CORECLK_DIV1 register

Table 7-157 CORECLK_DIV1 register

Bits	Name	Default	Description
[31:21]	-	-	Reserved
[20:16]	CLKDIV_CUR	-	The current clock divider value for the SYSPLL
[15:5]	-	-	Reserved
[4:0]	CLKDIV	0x1	Select the clock divider value for the SYSPLL

7.7.8.1.10 ACLK_CTRL register

Table 7-158 ACLK_CTRL register

Bits	Name	Default	Description
[31:16]	-	-	Reserved
[15:8]	CLKSELECT_CUR	-	Acknowledges the currently selected clock source: 0x0 – Reserved 0x1 – REFCLK 0x2 – SYSPLL Other values are reserved.
[7:0]	CLKSELECT	0x1	Select current clock source: 0x0 – Reserved 0x1 – REFCLK 0x2 – SYSPLL Other values are reserved.

7.7.8.1.11 ACLK_DIV1 register

Table 7-159 ACLK_DIV1 register

Bits	Name	Default	Description
[31:21]	-	-	Reserved
[20:16]	CLKDIV_CUR	-	The current clock divider value for the SYSPLL.
[15:5]	-	-	Reserved
[4:0]	CLKDIV	0x1	Select the clock divider value for the SYSPLL.

7.7.8.1.12 PWR_CTRL_CONFIG register

Table 7-160 PWR_CTRL_CONFIG register

Bits	Name	Default	Description
[31:16]	PWR_CTRL_ID	CFG	Indicates the type of PIK: 0x0071 – SCP PIK 0x0072 – MCP PIK
[15:4]	-	-	Reserved
[3:0]	NO_OF_PPU	0x0	Defines the number of PPUs in the Power Control Registers. This value is set to 0 to indicate no PPU.

7.7.8.1.13 PLL_STATUS0 register

Table 7-161 PLL_STATUS0 register

Bits	Name	Default	Description
[31:0]	PLL_STATUS0	0x0	Each bit shows the current status of the PLL Lock signal. PLL_STATUS0 contains the REFCLK_ST in bit 0, SYSPLL in bit 1 and DDRPLL in bit 2. The other bits are IMPLEMENTATION DEFINED.

7.7.8.2 PID4 register

Table 7-162 PID4 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	4KB_count	The number of 4KB address blocks required to access the registers, expressed in powers of 2. These bits read back as 0x4. This means that the POWER CONTROL register occupies 64KB address block.
[3:0]	jep106_c_code	The JEP106 continuation code value represents how many 0x7F continuation characters occur in the identity code of the manufacturer. These bits read back as 0x4.

7.7.8.3 PID5 register

Register is RAZ/WI.

7.7.8.4 PID6 register

Register is RAZ/WI.

7.7.8.5 PID7 register

Register is RAZ/WI.

7.7.8.6 PID0 register

Table 7-163 PID0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	part_number_0	These bits read back as 0xB8 .

7.7.8.7 PID1 register

Table 7-164 PID1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	jep106_id_3_0	JEP106 identity code [3:0]. These bits read back as 0xB .
[3:0]	part_number_1	Part Number. These bits read back as 0x0 .

7.7.8.8 PID2 register

Table 7-165 PID2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Revision	Identifies the revision of the POWER CONTROL registers. For revision rOp0, this field is set to 0x0.
[3]	jedec_used	This indicates that the POWER CONTROL registers uses a manufacturer's identity code that was allocated by JEDEC according to JEP106. This bit always reads back as 0x1 .
[2:0]	jep106_id_6_4	JEP106 identity code [6:4]. See the JEP106, Standard Manufacturer's Identification Code. Default 0x3.

7.7.8.9 PID3 register

Table 7-166 PID3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:4]	Rev	Revision And field.
[3:0]	mod_number	This is set to 0x0 .

7.7.8.10 Component ID 0 register

Table 7-167 Component ID 0 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_0	PREAMBLE_0. These bits read back as 0x0D.

7.7.8.11 Component ID 1 register

Table 7-168 Component ID 1 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_1	PREAMBLE_1. These bits read back as 0xF0.

7.7.8.12 Component ID 2 register

Table 7-169 Component ID 2 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_2	PREAMBLE_2. These bits read back as 0x05.

7.7.8.13 Component ID 3 register

Table 7-170 Component ID 3 register

Bits	Name	Description
[31:8]	-	Reserved, SBZ
[7:0]	comp_id_3	PREAMBLE_3. These bits read back as 0xB1.

7.7.8.14 SCP ONLY Registers

This section covers the registers that are used ONLY in SCP PIK and not MCP PIK.

7.7.8.15 PLL_STATUS<x> register

Table 7-171 PLL_STATUS<x> register

Bits	Name	Default	Description
[31:0]	PLL_STATUS	0x0	Each bit shows the current status of the PLL Lock signal.

PLL_STATUS<0> contains:

- REFCLK_ST in bit 0
- SYSPLL in bit 1
- DDRPPLL in bit 2
- INTPLL in bit 3

The other bits are IMPL_DEF.

PLL_STATUS<1-9> contains the status information for core PLLs for Cores 0-256. For example:

- PLL_STATUS<1> shows the PLL status for Cores 0-31.
- PLL_STATUS<2> shows the PLL status for Cores 32-63.

7.7.8.16 CONS_MMUTCU_INT_STATUS register

Table 7-172 CONS_MMUTCU_INT_STATUS register

Bits	Name	Default	Description
[3:0]	CONS_TCU_INT_STATUS	0x0	Each bit shows the current status of the RAS MMU_TCU_RASIRPT interrupt from the associated TCUs. Bit 0 – TCU 0 MMU_TCU_RASIRPT Bit 1 – TCU 1 MMU_TCU_RASIRPT ... Bit 3 – TCU 3 MMU_TCU_RASIRPT

7.7.8.17 CONS_MMUTBU_INT_STATUS0 register

Table 7-173 CONS_MMUTBU_INT_STATUS0 register

Bits	Name	Default	Description
[31:16]	CONS_TBU_INT_STATUS	0x0	Each bit shows the current status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU1. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 7 MMU_TBU_RASIRPT
[15:0]	CONS_TBU_INT_STATUS	0x0	Each bit shows the current status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU0. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT

7.7.8.18 CONS_MMUTBU_INT_STATUS1 register

Table 7-174 CONS_MMUTBU_INT_STATUS1 register

Bits	Name	Default	Description
[31:16]	CONS_TBU_INT_STATUS	0x0	Each bit shows the current status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU3. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT
[15:0]	CONS_TBU_INT_STATUS	0x0	Each bit shows the current status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU2. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT

7.7.8.19 CONS_MMUTCU_INT_CLR register

Table 7-175 CONS_MMUTCU_INT_CLR register

Bits	Name	Type	Description
[3:0]	CONS_TCU_INT_CLR	WO	Each bit clears the status of the RAS MMU_TCU_RASIRPT interrupt from the associated TCUs. Bit 0 – TCU 0 MMU_TCU_RASIRPT Bit 1 – TCU 1 MMU_TCU_RASIRPT ... Bit 3 – TCU 3 MMU_TCU_RASIRPT

7.7.8.20 CONS_MMUTBU_INT_CLR0 register

Table 7-176 CONS_MMUTBU_INT_CLR0 register

Bits	Name	Type	Description
[31:16]	CONS_TBU_INT_CLR0	WO	Each bit clears the status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU1. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT

Bits	Name	Type	Description
[15:0]	CONS_TBU_INT_CLR0	WO	Each bit clears the status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU0. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT

7.7.8.21 CONS_MMUTBU_INT_CLR1 register

Table 7-177 CONS_MMUTBU_INT_CLR1 register

Bits	Name	Type	Description
[31:16]	CONS_TBU_INT_CLR1	WO	Each bit clears the status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU3. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT
[15:0]	CONS_TBU_INT_CLR0	WO	Each bit clears the status of the RAS MMU_TBU_RASIRPT interrupt from the associated TBUs connected to TCU2. Bit 0 – TBU 0 MMU_TBU_RASIRPT Bit 1 – TBU 1 MMU_TBU_RASIRPT ... Bit 15 – TBU 15 MMU_TBU_RASIRPT

7.7.8.22 MHU_NS_INT_STATUS register

Table 7-178 MHU_NS_INT_STATUS register

Bits	Name	Default	Description
[31:0]	MHU_NS_INT_STATUS	0x0	Each bit shows the current status of the Non-Secure interrupt from the associated Cluster MHU. Bit 0 – Cluster 0 MHU NS Interrupt Bit 1 – Cluster 1 MHU NS Interrupt ... Bit 31 – Cluster 31 MHU NS Interrupt

7.7.8.23 MHU_S_INT_STATUS register

Table 7-179 MHU_S_INT_STATUS register

Bits	Name	Default	Description
[31:0]	MHU_S_INT_STATUS	0x0	Each bit shows the current status of the Secure interrupt from the associated Cluster MHU. Bit 0 – Cluster 0 MHU NS Interrupt Bit 1 – Cluster 1 MHU NS Interrupt ... Bit 31 – Cluster 31 MHU NS Interrupt

7.7.8.24 CPU_PPU_INT_STATUS<x> register

Table 7-180 CPU_PPU_INT_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CPU_PPU_INT_STATUS	0x0	Each bit shows the current status of the associated PPU Interrupt.

There can be up to eight CPU_PPU_INT_STATUS<x> registers to support up to 256 cores. For example:

- CPU_PPU_INT_STATUS0 shows the interrupt status for PPU's for Cores 0-31.
- CPU_PPU_INT_STATUS1 shows the interrupt status for PPU's for Cores 32-63.

7.7.8.25 CLUS_PPU_INT_STATUS<x> register

Table 7-181 CLUS_PPU_INT_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CLUS_PPU_INT_STATUS	0x0	Each bit shows the current status of the associated PPU Interrupt. Bit 0 – Cluster 0 PPU Interrupt Bit 1 – Cluster 1 PPU Interrupt Bit 31 – Cluster 31 PPU Interrupts

There can be up to eight CLUS_PPU_INT_STATUS<x> registers to support up to 256 clusters. For example:

- CLUS_PPU_INT_STATUS0 shows the interrupt status for PPU's for Clusters 0-31.
- CLUS_PPU_INT_STATUS1 shows the interrupt status for PPU's for Clusters 32-63.

7.7.8.26 TIMER_INT_STATUS<x> register

Table 7-182 TIMER_INT_STATUS<x> register

Bits	Name	Default	Description
[31:0]	TIMER_INT_STATUS	0x0	Each bit shows the current status of the associated Generic Timer.

There can be up to NUM_Clusters/32 TIMER_INT_STATUS<x> registers. For example:

- TIMER_INT_STATUS0 shows the interrupt status for Generic Timers for clusters 0-31.
- TIMER_INT_STATUS1 shows the interrupt status for Generic Timers for clusters 32-63.

7.7.8.27 CPU_PLL_LOCK_STATUS<x> register

Table 7-183 CPU_PLL_LOCK_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CPU_PLL_LOCK_STATUS	0x0	Each bit is set to 1 when the associated PLL lock status goes HIGH. Software must write 1 to clear the bit.

There can be up to eight CPU_PLL_LOCK_STATUS<x> registers. For example:

- CPU_PLL_LOCK_STATUS0 shows the interrupt status for PLLs Cores 0-32.
- CPU_PLL_LOCK_STATUS1 shows the interrupt status for PLLs for Cores 32-63.

7.7.8.28 CPU_PLL_UNLOCK_STATUS<x> register

Table 7-184 CPU_PLL_UNLOCK_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CPU_PLL_UNLOCK_STATUS	0x0	Each bit is set to 1 when the associated PLL lock status goes LOW. Software must write 1 to clear the bit.

There can be up to eight CPU_PLL_UNLOCK_STATUS<x> registers. For example:

- CPU_PLL_UNLOCK_STATUS0 shows the interrupt status for PLLs for Cores 0-31.
- CPU_PLL_UNLOCK_STATUS1 shows the interrupt status for PLLs for Cores 32-63.

7.7.8.29 CLUSTER_PLL_LOCK_STATUS<x> register

Table 7-185 CLUSTER_PLL_LOCK_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CLUSTER_PLL_LOCK_STATUS<x>	0x0	Each bit is set to 1 when the associated PLL lock status goes HIGH. Software must write 1 to clear the bit.

7.7.8.30 CLUSTER_PLL_UNLOCK_STATUS<x> register

Table 7-186 CLUSTER_PLL_UNLOCK_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CLUSTER_PLL_UNLOCK_STATUS<x>	0x0	Each bit is set to 1 when the associated PLL lock status goes LOW. Software must write 1 to clear the bit.

7.7.8.31 CLUS_FAULT_INT_STATUS<x> register

Table 7-187 CLUS_FAULT_INT_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CLUS_FAULT_INT_STATUS<x>	0x0	Each bit shows the current status of the associated Cluster Fault Interrupt.

7.7.8.32 CLUS_ECCERR_INT_STATUS<x> register

Table 7-188 CLUS_ECCERR_INT_STATUS<x> register

Bits	Name	Default	Description
[31:0]	CLUS_ECCERR_INT_STATUS<x>	0x0	Each bit shows the current status of the associated CPU ECCERR Interrupt.

There is one CLUS_ECCERR_INT_STATUS<x> register to support up to 32 clusters. For example, CPU_ECCERR_INT_STATUS0 captures the ECC Error interrupts status for Cluster0-Cluster31.

7.7.8.33 DMC0_4_INT_STATUS register

Consolidated interrupt status registers for DMC0 and DMC4 interrupts.

Table 7-189 DMC0_4_INT_STATUS register

Bits	Name	Default	Description
[14]	DMC4_combine_misx_oflow_st	0x0	DMC4_combine_misx_oflow_st
[13]	DMC4_combined_err_ovlow	0x0	DMC4_combined_err_ovlow
[12]	DMC4_combined_ecc_err_int	0x0	DMC4_combined_ecc_err_int
[11]	DMC4_combined_misc_access_int	0x0	DMC4_combined_misc_access_int
[10]	DMC4_temperature_event_int	0x0	DMC4_temperature_event_int
[9]	DMC4_failed_access_int	0x0	DMC4_failed_access_int
[8]	DMC4_combined_mgr_int	0x0	DMC4_combined_mgr_int
[6]	DMC0_combine_misx_oflow_st	0x0	DMC0_combine_misx_oflow_st
[5]	DMC0_combined_err_ovlow	0x0	DMC0_combined_err_ovlow

Bits	Name	Default	Description
[4]	DMC0_combined_ecc_err_int	0x0	DMC0_combined_ecc_err_int
[3]	DMC0_combined_misc_access_int	0x0	DMC0_combined_misc_access_int
[2]	DMC0_temperature_event_int	0x0	DMC0_temperature_event_int
[1]	DMC0_failed_access_int	0x0	DMC0_failed_access_int
[0]	DMC0_combined_mgr_int	0x0	DMC0_combined_mgr_int

7.7.8.34 DMC1_5_INT_STATUS register

Consolidated interrupt status registers for DMC1 and DMC5 interrupts.

Table 7-190 DMC1_5_INT_STATUS register

Bits	Name	Default	Description
[14]	DMC5_combine_misx_oflow_st	0x0	DMC5_combine_misx_oflow_st
[13]	DMC5_combined_err_ovlow	0x0	DMC5_combined_err_ovlow
[12]	DMC5_combined_ecc_err_int	0x0	DMC5_combined_ecc_err_int
[11]	DMC5_combined_misc_access_int	0x0	DMC5_combined_misc_access_int
[10]	DMC5_temperature_event_int	0x0	DMC5_temperature_event_int
[9]	DMC5_failed_access_int	0x0	DMC5_failed_access_int
[8]	DMC5_combined_mgr_int	0x0	DMC5_combined_mgr_int
[6]	DMC1_combine_misx_oflow_st	0x0	DMC1_combine_misx_oflow_st
[5]	DMC1_combined_err_ovlow	0x0	DMC1_combined_err_ovlow
[4]	DMC1_combined_ecc_err_int	0x0	DMC1_combined_ecc_err_int
[3]	DMC1_combined_misc_access_int	0x0	DMC1_combined_misc_access_int
[2]	DMC1_temperature_event_int	0x0	DMC1_temperature_event_int
[1]	DMC1_failed_access_int	0x0	DMC1_failed_access_int
[0]	DMC1_combined_mgr_int	0x0	DMC1_combined_mgr_int

7.7.8.35 DMC2_6_INT_STATUS register

Consolidated interrupt status registers for DMC2 and DMC6 interrupts.

Table 7-191 DMC2_6_INT_STATUS register

Bits	Name	Default	Description
[14]	DMC6_combine_misx_oflow_st	0x0	DMC6_combine_misx_oflow_st
[13]	DMC6_combined_err_ovlow	0x0	DMC6_combined_err_ovlow
[12]	DMC6_combined_ecc_err_int	0x0	DMC6_combined_ecc_err_int
[11]	DMC6_combined_misc_access_int	0x0	DMC6_combined_misc_access_int

Bits	Name	Default	Description
[10]	DMC6_temperature_event_int	0x0	DMC6_temperature_event_int
[9]	DMC6_failed_access_int	0x0	DMC6_failed_access_int
[8]	DMC6_combined_mgr_int	0x0	DMC6_combined_mgr_int
[6]	DMC2_combine_misx_oflow_st	0x0	DMC2_combine_misx_oflow_st
[5]	DMC2_combined_err_ovlow	0x0	DMC2_combined_err_ovlow
[4]	DMC2_combined_ecc_err_int	0x0	DMC2_combined_ecc_err_int
[3]	DMC2_combined_misc_access_int	0x0	DMC2_combined_misc_access_int
[2]	DMC2_temperature_event_int	0x0	DMC2_temperature_event_int
[1]	DMC2_failed_access_int	0x0	DMC2_failed_access_int
[0]	DMC2_combined_mgr_int	0x0	DMC2_combined_mgr_int

7.7.8.36 DMC3_7_INT_STATUS register

Consolidated interrupt status registers for DMC3 and DMC7 interrupts.

Table 7-192 DMC3_7_INT_STATUS register

Bits	Name	Default	Description
[14]	DMC7_combine_misx_oflow_st	0x0	DMC7_combine_misx_oflow_st
[13]	DMC7_combined_err_ovlow	0x0	DMC7_combined_err_ovlow
[12]	DMC7_combined_ecc_err_int	0x0	DMC7_combined_ecc_err_int
[11]	DMC7_combined_misc_access_int	0x0	DMC7_combined_misc_access_int
[10]	DMC7_temperature_event_int	0x0	DMC7_temperature_event_int
[9]	DMC7_failed_access_int	0x0	DMC7_failed_access_int
[8]	DMC7_combined_mgr_int	0x0	DMC7_combined_mgr_int
[6]	DMC3_combine_misx_oflow_st	0x0	DMC3_combine_misx_oflow_st
[5]	DMC3_combined_err_ovlow	0x0	DMC3_combined_err_ovlow
[4]	DMC3_combined_ecc_err_int	0x0	DMC3_combined_ecc_err_int
[3]	DMC3_combined_misc_access_int	0x0	DMC3_combined_misc_access_int
[2]	DMC3_temperature_event_int	0x0	DMC3_temperature_event_int
[1]	DMC3_failed_access_int	0x0	DMC3_failed_access_int
[0]	DMC3_combined_mgr_int	0x0	DMC3_combined_mgr_int

Appendix A Arm IP used in the Reference Design

The following Arm IP products were used to build the RD-V1 system:

- Arm® Neoverse™ V1
- CMN-650
- CoreSight ELA-500
- CoreSight SoC-600
- CoreSight STM-500
- CoreLink NIC-450, includes NIC-400 and ADB-400
- CoreLink GIC-700
- CoreLink MMU-600
- Cortex-M7
- Corstone-100, includes Cortex-M SDK
- CoreLink PCK-600
- CoreLink TZC-400
- PL011 PrimeCell UART
- BP140 AXI Mem IF

The following additional logic blocks were developed to build the system. These blocks are developed for Arm internal use only and not included as part of the Arm standard IP products:

- Clock Generators - Dividers + Dynamic Clock Gating - using PCK-600
- Power Management Network - Configuration/network of PCK-600
- CoreSight/Debug Element - Configuration/network of CS SoC-600 and PCK-600
- Asynchronous Bridges for CDC - using ADB-400 and NIC-450 IP components
- Access Control Gates (ACG) – to gracefully error access to unpowered regions
- Message Handling Unit (MHU) – for inter-processor communications
- NIC-450 switches - for fine-grain decoding
- Generic Timers and Watchdogs system implementation
- Configurable AXI register slices - using NIC-450
- Multiple APB Register modules - for control/status from firmware
- Address Decoders - to enable SCP access to the AP memory map
- AHB to APB bridge, APB Watchdog, AHB Default Slave - from Cortex-M SDK

Appendix B Revisions

This appendix describes the technical changes between released issues of this book.

Table B-1 Issue 01

Change	Location
First release	-