Juno r2 ARM® Development Platform SoC

Revision: r2p0

Technical Reference Manual

ARM®

ARM DDI 0515F (ID052716)

Juno r2 ARM Development Platform SoC
Technical Reference Manual

Release Information

The following changes have been made to this book.

Change history

Date	Issue	Confidentiality	Change
03 June 2014	A	Confidential	First issue of TRM that relates to r0p0
23 September 2014	B.a	Non-Confidential	Second issue of TRM that relates to r0p0
29 April 2016	B.b	Non-Confidential	Third issue of TRM that relates to r0p0\
31 March 2015	C	Non-Confidential	First issue of TRM that relates to r1p0
05 June 2015	D.a	Non-Confidential	Second issue of TRM that relates to r1p0
29 April 2016	D.b	Non-Confidential	Third issue of TRM that relates to r1p0
20 November 2015	E	Non-Confidential	First issue of TRM that relates to r2p0
31 May 2016	F	Non-Confidential	Second issue of TRM that relates to r2p0

Proprietary Notice

ARM Limited. Company 02557590 registered in England.

110 Fulbourn Road, Cambridge, England CB1 9NJ.

LES-PRE-20348

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.

Product Status

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

Web Address

`http://www.arm.com`

Contents
Juno r2 ARM Development Platform SoC Technical Reference Manual

Preface

About this book vii
Feedback xii

Chapter 1 **Introduction**

1.1 About the ARM Development Platform (ADP) SoC 1-2
1.2 Components 1-3
1.3 Compliance 1-4
1.4 Product documentation and architecture 1-7
1.5 ARM IP revisions 1-8
1.6 Product revisions 1-9

Chapter 2 **Functional Description**

2.1 ADP components 2-2
2.2 Application processors 2-3
2.3 Mali-T624 Graphics Processing Unit 2-6
2.4 Memory 2-7
2.5 Application processor peripherals 2-8
2.6 Interconnect and memory system 2-17
2.7 Security 2-20
2.8 Timers 2-29
2.9 Debug and Profiling 2-33
2.10 Boot 2-42
2.11 HDLCD controller 2-43
2.12 PCIe 2-44
2.13 Thin Links 2-46

Chapter 3 **Programmers Model**
3.1 About this programmers model 3-2
3.2 Application processor interrupt map 3-3
3.3 Application processor memory map 3-10
3.4 ADP System Security Control Registers 3-27
3.5 System override registers 3-40
3.6 MHU Registers 3-42
3.7 SoC Interconnect NIC-400 Registers 3-50
3.8 Compute subsystem NIC-400 Registers 3-63
3.9 HDLCD Registers 3-68
3.10 PCIe Control Registers 3-85
3.11 PCIe Root Port configuration registers 3-96
3.12 MSI Registers 3-132
3.13 Trusted Entropy Source Registers 3-139

Appendix A **Subcomponent Configurations**
A.1 Cortex-A72 processor cluster A-2
A.2 Cortex-A53 processor cluster A-3
A.3 Mali-T624 GPU A-4
A.4 CoreLink MMU-40x System Memory Management Unit (SMMU) components A-5
A.5 CoreLink DMC-400 Dynamic Memory Controller (DMC) A-6
A.6 CoreLink SMC-354 Static Memory Controller (SMC) A-7

Appendix B **Revisions**

Preface

This preface introduces the *Juno r2 ARM® Development Platform (ADP) SoC Technical Reference Manual*. It contains the following sections:

- *About this book* on page vii.
- *Feedback* on page xii.

About this book

This book is for the *Juno r2 ARM Development Platform* (ADP) SoC. It provides a high-level overview of the ADP. It describes all architectural information, and as such, facilitates the creation of ADP software.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r2p0, where:

r*m* Identifies the major revision of the product, for example, r2.

p*n* Identifies the minor revision or modification status of the product, for example, p0.

Intended audience

This book is written for software engineers who want to work with an ARM reference platform. The manual describes the functionality of the ADP.

Using this book

This book is organized into the following chapters:

Chapter 1 *Introduction*

Read this chapter for a high-level view of the *ARM Development Platform* (ADP) SoC and a description of its features.

Chapter 2 *Functional Description*

Read this chapter for a description of the major interfaces and components of the ADP. This chapter also describes how the components operate.

Chapter 3 *Programmers Model*

Read this chapter for a description of the address map and registers of the ADP.

Appendix A *Subcomponent Configurations*

Read this appendix for a description of the parameter configurations for each of the system IP components that are used in the ARM Development Platform.

Appendix B *Revisions*

Read this appendix for a description of the technical changes between released issues of this book.

Glossary

The *ARM Glossary* is a list of terms that are 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.

The *ARM Glossary* is available on the ARM Infocenter at http://infocenter.arm.com/help/topic/com.arm.doc.aeg0014-/index.html.

Conventions

Conventions that this book can use are described in:

- *Typographical conventions*.
- *Timing diagrams*.
- *Signals* on page ix.

Typographical conventions

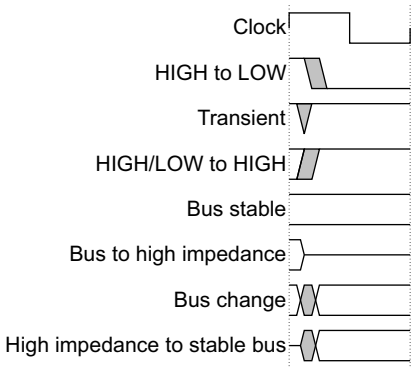
The following table describes the typographical conventions:

Style	Purpose
italic	Introduces special terminology, denotes cross-references, and 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`	Denotes a permitted abbreviation for a command or option. You can enter the underlined text instead of the full command or option name.
`monospace` *`italic`*	Denotes arguments to monospace text where the argument is to be replaced by a specific value.
`monospace bold`	Denotes language keywords when used outside example code.
<and>	Encloses replaceable terms for assembler syntax where they appear in code or code fragments. For example: `MRC p15, 0 <Rd>, <CRn>, <CRm>, <Opcode_2>`
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, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The figure named *Key to timing diagram conventions* explains the components used in timing diagrams. Variations, when they occur, have clear labels. You must not assume any timing information that is not explicit in the diagrams.

Shaded bus and signal areas are undefined, so the bus or signal can assume any value within the shaded area at that time. The actual level is unimportant and does not affect normal operation.



Key to timing diagram conventions

Timing diagrams sometimes show single-bit signals as HIGH and LOW at the same time and they look similar to the bus change shown in *Key to timing diagram conventions* on page viii. If a timing diagram shows a single-bit signal in this way then its value does not affect the accompanying description.

Signals

The signal conventions are:

Signal level The level of an asserted signal depends on whether the signal is active-HIGH or active-LOW. Asserted means:
- HIGH for active-HIGH signals.
- LOW for active-LOW signals.

Lower-case n At the start or end of a signal name denotes an active-LOW signal.

Additional reading

This section lists publications by ARM and by third parties.

See Infocenter, http://infocenter.arm.com for access to ARM documentation.

ARM publications

This book contains information that is specific to this product. See the following documents for other relevant information:

- *Juno r2 ARM® Development Platform SoC Technical Overview* (ARM DTO 0038). http://infocenter.arm.com/help/topic/com.arm.doc.dto0038c
- *ARM® Versatile™ Express Juno r2 Development Platform (V2M-Juno r2) Technical Reference Manual* (ARM 100114). http://infocenter.arm.com/help/topic/com.arm.doc.100114_0200_00_en.
- *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual* (ARM 100095_0002_03_en). http://infocenter.arm.com/help/topic/com.arm.doc.100095_0002_03_en/
- *ARM® Cortex®-A53 MPCore Processor Technical Reference Manual* (ARM DDI 0500). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0500f
- *ARM® Cortex®-M3 Technical Reference Manual* (ARM DDI 0337). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0337i
- *ARM® CoreLink™ CCI-400 Cache Coherent Interconnect Technical Reference Manual* (ARM DDI 0470). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0470i
- *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (ARM DDI 0475). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0475c and http://infocenter.arm.com/help/topic/com.arm.doc.ddi0475a
- *ARM® CoreLink™ TLX-400 Network Interconnect Thin Links Supplement to CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (ARM DSU 0028) http://infocenter.arm.com/help/topic/com.arm.doc.dsu0028a
- *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* (ARM DSU 0026). http://infocenter.arm.com/help/topic/com.arm.doc.dsu0026c

- *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual* (ARM DDI 0466). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0466e
- *ARM® CoreLink™ SMC-35x AXI Static Memory Controller Series Technical Reference Manual* (ARM DDI 0380). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0380h
- *ARM® CoreLink™ DMA-330 DMA Controller Technical Reference Manual* (ARM DDI 0424). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0424d
- *ARM® CoreLink™ MMU-400 System Memory Management Unit Technical Reference Manual* (ARM DDI 0472). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0472a
- *ARM® CoreLink™ MMU-401 System Memory Management Unit Technical Reference Manual* (ARM DDI 0521). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0521a
- *ARM® CoreLink™ GIC-400 Generic Interrupt Controller Technical Reference Manual* (ARM DDI 0471). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0471b
- *ARM® CoreLink™ TZC-400 TrustZone® Address Space Controller Technical Reference Manual* (ARM DDI 0424). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0504b
- *ARM® CoreSight™ Components Technical Reference Manual* (ARM DDI 0480). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0480b
- *ARM® CoreSight™ Trace Memory Controller Technical Reference Manual* (ARM DDI 0461). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0461b
- *ARM® CoreSight™ STM-500 System Trace Macrocell Technical Reference Manual* (ARM DDI 0528). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0528b
- *ARM® CoreSight™ System Trace Macrocell Technical Reference Manual* (ARM DDI 0444). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0444b
- *ARM® Watchdog Module (SP805) Technical Reference Manual* (ARM DDI 0270). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0270b
- *ARM® PrimeCell® UART (PL011) Technical Reference Manual* (ARM DDI 0270). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0183g
- *ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile* (ARM DDI 0487). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0487a.b
- *ARM®v7M Architecture Reference Manual* (ARM DUI 0403). http://infocenter.arm.com/help/topic/com.arm.doc.ddi0406c
- *ARM® Generic Interrupt Controller Architecture Specification, Architecture version 2.0* (ARM IHI 0048). http://infocenter.arm.com/help/topic/com.arm.doc.ihi0048b
- *AMBA® AXI™ and ACE™ Protocol Specification AXI3™, AXI4™, and AXI4-Lite™ ACE and ACE-Lite™* (ARM IHI 0022). http://infocenter.arm.com/help/topic/com.arm.doc.ihi0022e
- *AMBA® 3 ATB Protocol Specification* (ARM IHI 0032). http://infocenter.arm.com/help/topic/com.arm.doc.ihi0032b
- *AMBA® 3 AHB-Lite Protocol Specification* (ARM IHI 0033). http://infocenter.arm.com/help/topic/com.arm.doc.ihi0033a
- *AMBA® APB Protocol Specification* (ARM IHI 0024). http://infocenter.arm.com/help/topic/com.arm.doc.ihi0024c

- *ARM® Architecture Standard Configurations* (ARM DEN 0016).
  http://infocenter.arm.com/help/topic/com.arm.doc.den0016b
- *Principles of ARM® Memory Maps Platform Design Document* (ARM DEN 0001).
  http://infocenter.arm.com/help/topic/com.arm.doc.den0001c
- *Trusted Base System Architecture Platform Design Document* (ARM DEN 0007).
- *ARM® Server Base System Architecture Platform Design Document* (ARM-DEN-0029).
  http://infocenter.arm.com/help/topic/com.arm.doc.den0029

Other publications

This section lists relevant documents published by third parties:

- *PCI Express Base Specification, Revision 3.0*.

Feedback

ARM welcomes feedback on this product and its documentation.

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.

Feedback on content

If you have comments on content then send an e-mail to `errata@arm.com`. Give:

- The title.
- The number, ARM DDI 0515F.
- The page numbers to which your comments apply.
- A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

Note

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.

Chapter 1
Introduction

This chapter introduces the *Juno r2 ARM Development Platform* (ADP). It contains the following sections:

- *About the ARM Development Platform (ADP) SoC* on page 1-2.
- *Components* on page 1-3.
- *Compliance* on page 1-4.
- *Product documentation and architecture* on page 1-7.
- *ARM IP revisions* on page 1-8.
- *Product revisions* on page 1-9.

1.1 About the *ARM Development Platform* (ADP) SoC

The ADP SoC is a development chip that is implemented on the TSMC 28HPM process and delivers the following:

- A platform for ARMv8-A software and tool development to enable robust testing of software deliverables on Linaro-based kernels such as Linux and Android.
- A platform for optimized software and tool development of heterogeneous compute such as:
  - AArch64 kernel and tools.
  - big.LITTLE™.
  - *General-Purpose computing on Graphics Processing Units* (GPGPU) compute, for example, OpenCL.
- 3D graphics and *Graphics Processing Unit* (GPU) compute.
- Power and thermal management.
- 64-bit drivers, file system, and middleware.
- Third-party hypervisor integration.
- Third-party Secure OS integration.
- A platform for Secure software development that complies with the *Trusted Base System Architecture CLIENT1 Platform Design Document*.
- Platform tuning for power and performance.

1.2 Components

The ADP SoC contains the following components:

- A compute platform including:
  - A Cortex®-A72 processor cluster containing two cores and 2MB of L2 cache.
  - A Cortex-A53 processor cluster containing four cores and 1MB of L2 cache.
  - A Mali-T624 GPU containing four shader cores.
- *System Control Processor* (SCP) based on the Cortex-M3 processor.
- Dual HDCLCD Display Controllers with the resolution characteristics that Table 2-5 on page 2-11 shows.
- CoreLink™ CCI-400 Cache Coherent Interconnect.
- CoreLink NIC-400 Network Interconnect.
- Thin Links based chip-to-chip interface.
- System Memory Management Units.
  - CoreLink MMU-400 System Memory Management Units.
  - CoreLink MMU-401 System Memory Management Units.
- CoreLink DMC-400 *Dynamic Memory Controller* (DMC).
- CoreLink TZC-400 TrustZone Address Space Controller.
- System Profiler.
- ARM CoreSight™ technology.
- Dual 32-bit DDR3. 1600MTps.
- A *Peripheral Component Interconnect Express* (PCIe) Gen2.0 four lanes, Root Port and PHY, that has both I/O coherent and non-coherent modes.
- USB 2.0 *Enhanced Host Controller Interface* (EHCI), *Open Host Controller Interface* (OHCI), 480Mbps, *UTMI+ Low-Pin Interface* (ULPI) interface to off-chip PHY.[1]
- Static memory controller PL354, 64MB NOR flash and board peripherals.
- CoreLink DMA-330 *Direct Memory Access* (DMA) Controller.
- Two PL011 UARTs.
- Single I²S with four stereo channels.
- I²C.
- *Process, Voltage, and Temperature* (PVT) monitors.
- Security peripherals:
  - *Random Number Generator* (RNG).
  - Simulated *Non-Volatile* (NV) Counters.
  - Simulated fuses.

Figure 2-1 on page 2-2 shows a block diagram of the ADP system.

1. *USB 2.0 Transceiver Macrocell Interface* (UTMI).

1.3 Compliance

The ADP complies with, or includes components that comply with, the following specifications:

- *ARM Architecture*.
- *Generic Interrupt Controller architecture*.
- *Advanced Microcontroller Bus Architecture*.
- *Platform Design Documents and white papers* on page 1-5.
- *big.LITTLE* on page 1-5.
- *Virtualization* on page 1-5.

This TRM complements the TRMs for included components, architecture reference manuals, architecture specifications, protocol specifications, and relevant external standards. It does not duplicate information from these sources.

1.3.1 ARM Architecture

The ADP implements the ARMv8-A architecture and includes:

- Support for both AArch32 and AArch64 Execution states.
- Support for all exception levels, EL0, EL1, EL2, and EL3, in each Execution state.
- The A32 instruction set, previously called the ARM instruction set.
- The T32 instruction set, previously called the Thumb instruction set.
- The A64 instruction set.

The processor clusters support the following features:

- Advanced *Single Instruction Multiple Data* (SIMD) operations.
- Floating-point operations.
- Cryptography Extension.

The clusters do not support the T32EE, that is, ThumbEE, instruction set.

See the *ARM® Architecture Reference Manual, ARMv8, for ARMv8-A architecture profile* for more information.

1.3.2 Generic Interrupt Controller architecture

The ADP implements the ARM *Generic Interrupt Controller* (GIC) v2m architecture, that the *Server Base System Architecture* (SBSA) *Platform Design Document* (PDD) defines.

GICv2m provides an extension to the GICv2 Generic Interrupt Controller Architecture, that the *ARM Generic Interrupt Controller Architecture Specification, Architecture Version 2.0* defines. GICv2m enables PCI Express *Message Signaled Interrupts* (MSIs) to set GICv2 *Shared Peripheral Interrupts* (SPIs) to pending and provides a similar mechanism to the message-based interrupt features added in GICv3.

1.3.3 Advanced Microcontroller Bus Architecture

The ADP complies with the:

- AMBA® 4 *Advanced eXtensible Interface* (AXI) and ACE protocol. See the *AMBA® AXI™ and ACE™ Protocol Specification AXI3™, AXI4™, and AXI4-Lite™ ACE and ACE-Lite™*.
- AMBA *Advanced Peripheral Bus* (APB) protocol. See the *AMBA® APB Protocol Specification*.
- AMBA 3 *Advanced Trace Bus* (ATB) protocol. See the *AMBA® 3 ATB Protocol Specification*.

1.3.4 *Advanced High-performance Bus* (AHB)-Lite

The *System Control Processor* (SCP) Cortex-M3 processor in the ADP complies with the ARM *AMBA® 3 AHB-Lite Protocol (v1.0) Specification*.

AHB-Lite is a subset of the full AMBA AHB protocol specification. It provides all the basic functions that are required by most AMBA AHB slave and master designs, particularly when used with a multi-layer AMBA interconnect.

1.3.5 Platform Design Documents and white papers

The ADP architecture complies with the following *Platform Design Documents* (PDDs):

- Level 1 of *Server Base System Architecture PDD*.
  http://infocenter.arm.com/help/topic/com.arm.doc.den0029
- *Trusted Board Boot Requirements PDD.*
- *Trusted Base System Architecture CLIENT1 PDD.*
  There are some limitations to the TBSA. See *Trusted Base System Architecture (TBSA) compliance* on page 2-20.

The ADP architecture complies with the following white paper:

- *Principles of ARM® Memory Maps (v1.0)*.
  http://infocenter.arm.com/help/topic/com.arm.doc.den0001c

1.3.6 big.LITTLE

The ADP includes the following coherent clusters:

- Cortex-A72 cluster.
- Cortex-A53 cluster.

The inclusion of two coherent clusters enables you to use big.LITTLE power-management strategies. This enables a single system to handle both high-intensity and low-intensity tasks in the most energy-efficient manner.

1.3.7 Virtualization

The Cortex-A72 and Cortex-A53 clusters implement the ARMv8-A architecture that includes support for virtualization. The ADP provides extra support for virtualization as follows:

The CoreLink GIC-400 Generic Interrupt Controller supports virtualization of interrupts.

Stage 2 page table translation for masters, that the following provide:

- A CoreLink MMU-401 *System Memory Management Unit* (SMMU) for the following:
  - The *Embedded Trace Router* (ETR).
  - PCI Express.
  - Each HDLCD controller.
  - The CoreLink DMA-330.
- A CoreLink MMU-400 SMMU for the Mali-T624 GPU.

  Note

  The MMU-400 does not support the AArch64 page table format.

See also:

- *ARM® Architecture Reference Manual, ARMv8-A, for ARMv8-A architecture profile Architecture Reference Manual*.
- *ARM® CoreLink™ GIC-400 Generic Interrupt Controller Technical Reference Manual*.
- *ARM® CoreLink™ MMU-400 System Memory Management Unit Technical Reference Manual*.
- *ARM® CoreLink™ MMU-401 System Memory Management Unit Technical Reference Manual*.

1.4 Product documentation and architecture

For information on the relevant architectural standards and protocols, see *Compliance* on page 1-4.

1.5 ARM IP revisions

Table 1-1 shows the IP versions of the ARM components that the ADP uses.

Table 1-1 IP revisions

Product	Revision
Cortex-A72 processor cluster	r0p0-50
Cortex-A53 processor cluster	r0p3
Cortex-M3 processor	r2p1
Mali-T624 GPU	r1p0
CoreLink NIC-400 Network Interconnect	r0p0 and r0p2
CoreLink CCI-400 Cache Coherent Interconnect	r1p3
CoreLink DMC-400 Dynamic Memory Controller	r1p2
CoreLink DMA-330 DMA Controller	r1p2
CoreLink TLX-400 Network Interconnect Thin Links	r0p0
CoreLink MMU-400 System Memory Management Unit	r0p0
CoreLink MMU-401 System Memory Management Unit	r0p0
CoreLink GIC-400 Generic Interrupt Controller	r0p1
CoreLink TZC-400 TrustZone Address Space Controller	r0p1
CoreLink ADB-400 AMBA Domain Bridge	r1p0
BP136 AMBA 2 AHB to AMBA 3 AXI Bridges	r0p1
BP140 Internal Memory Interface	r0p0
BP210 Cortex-M System Design Kit	r0p0
PL354 Static Memory Controller	r2p2
PL011 UART	r1p5
PL111 HDLCD Controller	r0p2
Power, Voltage, Temperature (PVT) sensors	-
Random Number Generator (RNG)	-
CoreSight SoC	r1p0
CoreSight STM-500	r0p1
SP805 Watchdog	r2p0
28HPM GPIO	r0p0
28HPM DDR3 PHY	r0p0
PCIe write buffer	-

1.6 Product revisions

This section describes the differences in functionality between product revisions:

r0p0 First release.

r1p0 Functional changes are:

- New IP versions of the following:
  - Cortex-A57 processor.
  - Cortex-A53 processor.
  - CoreLink CCI-400 Cache Coherent Interconnect.
  - CoreLink NIC-400 Network Interconnect.
- Enhanced debug features:
  - CoreSight STM-500 64-bit trace.
  - Logic Analyzer.
  - Single-chain debug expanded to include CoreLink CCI-400 and CoreLink NIC-400.
  - Extra *Embedded Trace FIFO* (ETF) for debug profiling.
- *Quality of Service* (QoS) improvements on the off-chip IOFPGA interface.

r2p0 Functional changes are:

- Cortex-A72 processor replaces the Cortex-A57 processor.

Chapter 2
Functional Description

This chapter describes the functionality of the *Juno r2 ARM Development Platform* (ADP).

It contains the following sections:

- *ADP components* on page 2-2.
- *Application processors* on page 2-3.
- *Mali-T624 Graphics Processing Unit* on page 2-6.
- *Memory* on page 2-7.
- *Application processor peripherals* on page 2-8.
- *Interconnect and memory system* on page 2-17.
- *Security* on page 2-20.
- *Timers* on page 2-29.
- *Debug and Profiling* on page 2-33.
- *Boot* on page 2-42.
- *HDLCD controller* on page 2-43.
- *PCIe* on page 2-44.
- *Thin Links* on page 2-46.

2.1 ADP components

This chapter describes the functionality of the components that Figure 2-1 shows.

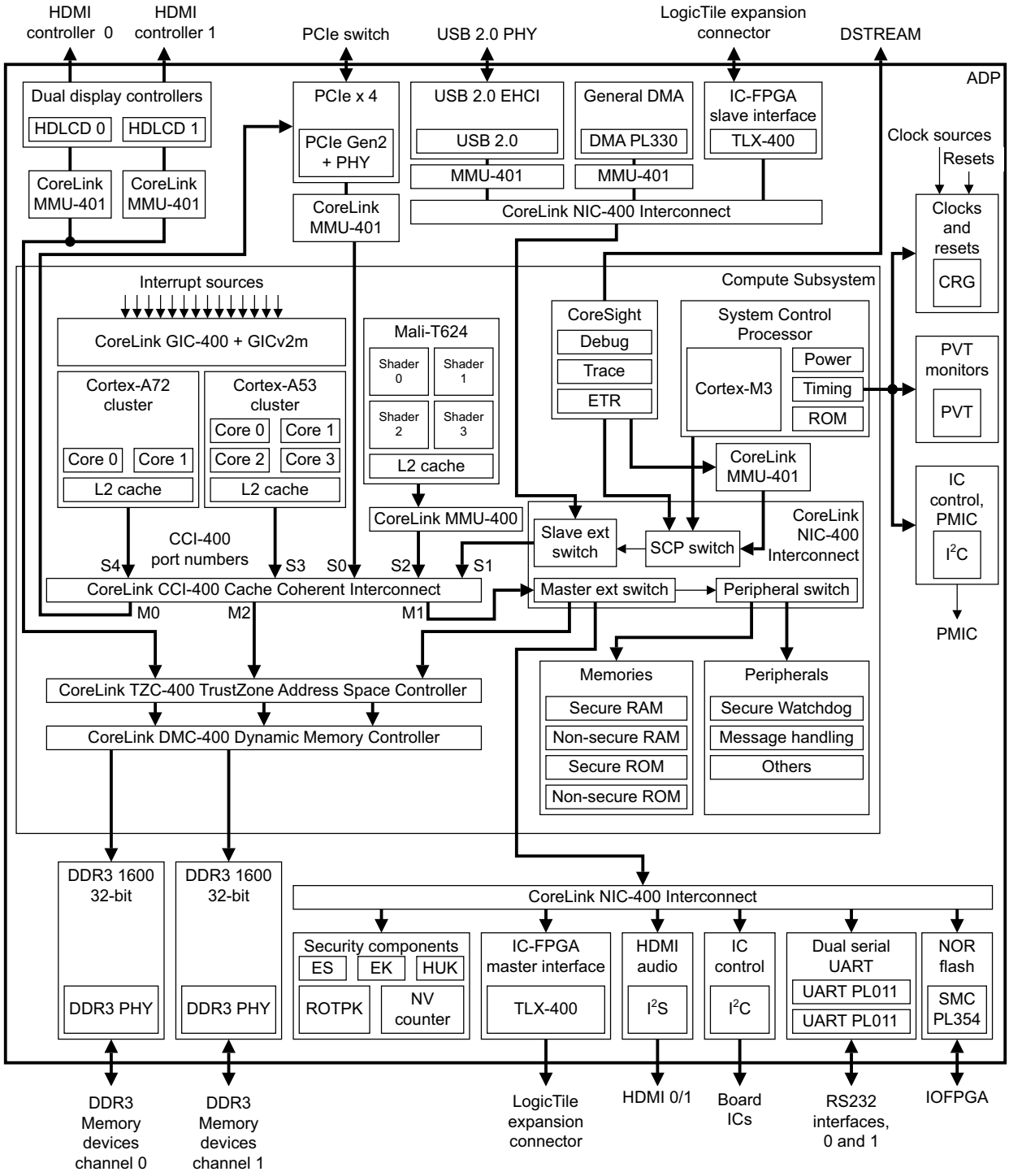


Figure 2-1 ADP block diagram

2.2 Application processors

This section describes:

- *Processor affinity*.
- *Cortex-A72 cluster*.
- *Cortex-A53 cluster*.
- *Interrupts*.
- *Interconnects* on page 2-4.
- *Events* on page 2-5.

2.2.1 Processor affinity

Table 2-1 shows the affinity values assigned to each application processor.

Table 2-1 Application processor affinity values

Processor	Affinity level 2 value	Affinity level 1 value	Affinity level 0 value
Cortex-A72 cluster	0	0	0-1
Cortex-A53 cluster	0	1	0-3

2.2.2 Cortex-A72 cluster

The Cortex-A72 cluster contains the following:

- Two cores and a *Snoop Control Unit* (SCU) that you can use to ensure coherency within the cluster, and with other devices, using the CoreLink CCI-400.
- An integrated Generic Timer for each core in the Cortex-A72 cluster.

See *Cortex-A72 processor cluster* on page A-2.

2.2.3 Cortex-A53 cluster

The Cortex-A53 cluster contains the following:

- Four cores and a *Snoop Control Unit* (SCU) that you can use to ensure coherency within the cluster, and with other devices, using the CoreLink CCI-400.
- An integrated Generic Timer for each core in the Cortex-A53 cluster.

See *Cortex-A53 processor cluster* on page A-3.

2.2.4 Interrupts

All Cortex-A72 and Cortex-A53 interrupts are routed through a shared *Generic Interrupt Controller* (GIC), the CoreLink GIC-400. The Cortex-M3 SCP has its own interrupt controller. Some interrupts from the SCP are routed to the shared GIC-400.

The GIC-400 provides registers that manage the following, for one or more:

- Interrupt sources.
- Interrupt behavior.
- Interrupt routing.

The CoreLink GIC-400 supports Security Extensions and can:

- Enable, disable, and generate interrupts from hardware or peripheral interrupt sources.
- Generate software interrupts.
- Mask and prioritize interrupts.

The CoreLink GIC-400 signals interrupts to each connected cluster as appropriate. It implements the following interrupt requests to a connected cluster:

- **IRQ**.
- **VIRQ**.
- **FIQ**.
- **VFIQ**.

Figure 2-2 shows the interrupt connections.

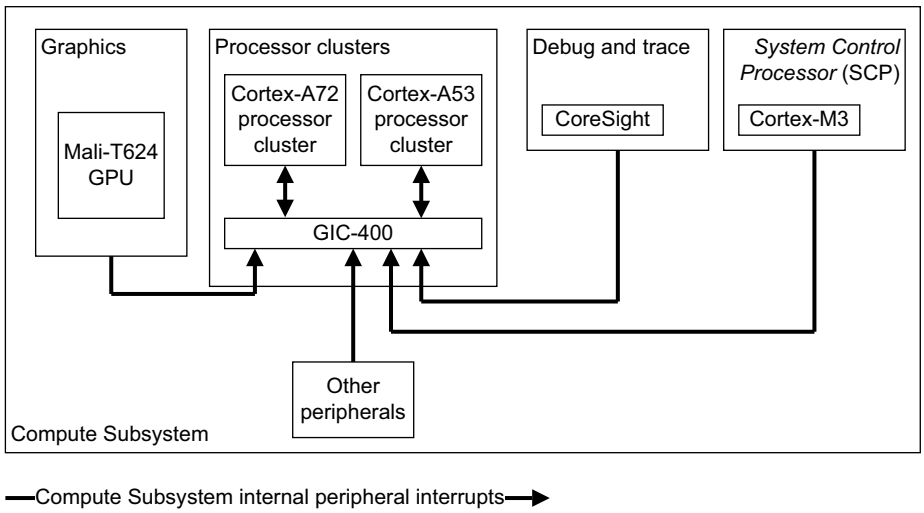


Figure 2-2 Application processor interrupt connections

2.2.5 Interconnects

This section describes the interconnects that the ADP uses. It contains the following subsections:

- *CoreLink CCI-400 Cache Coherent Interconnect*.
- *CoreLink NIC-400 Network Interconnect* on page 2-5.
- *CoreLink DMC-400 Dynamic Memory Controller* on page 2-5.

CoreLink CCI-400 Cache Coherent Interconnect

The CoreLink CCI-400 Cache Coherent Interconnect provides:

- Full cache coherency between the Cortex-A72 and Cortex-A53 clusters.
- I/O coherency with the Mali-T624 GPU.
- I/O coherency for *PCI Express* (PCIe).

The CCI-400 also supports *Distributed Virtual Memory* (DVM) messaging for managing caches, system MMUs, and *Translation Lookaside Buffers* (TLBs) in ACE-Lite components.

Note

- DVM messages do not maintain the MMU-401 for the *Embedded Trace Router* (ETR).

- DVM messages sent from the Cortex-A72 and Cortex-A53 processors are DVMv8 and are not required to maintain MMU-400 or MMU-401 because the processors are DVMv7 components. Therefore, you must use the registers for each component to maintain the system MMUs.

The Mali-T624 GPU connects to a single CCI-400 ACE-Lite slave port.

See the *ARM® CoreLink™ CCI-400 Cache Coherent Interconnect Technical Reference Manual*.

CoreLink NIC-400 Network Interconnect

The CoreLink NIC-400 Network Interconnect provides a complete high-performance, optimized, AMBA-compliant network infrastructure. CoreLink NIC-400 Network Interconnect components connect other system components into the CoreLink CCI-400 Cache Coherent Interconnect. For example, the CoreLink NIC-400 connects:

- Master components to a CCI-400 slave port.
- Slave components to a CCI-400 master port.

CoreLink DMC-400 Dynamic Memory Controller

The CoreLink DMC-400 Dynamic Memory Controller connects from two 128-bit CCI-400 ACE-Lite master ports, and the HDLCD controllers, through the TZC-400, to the two 32-bit *DDR PHY Interfaces* (DFIs). Transactions from any of the three ACE-Lite ports can access either of the DFI interfaces.

2.2.6 Events

The **EVENTO** and **EVENTI** signals are cross-wired between the application processor clusters. Cross-wiring ensures that *Send Event* (SEV) instructions that a core in one cluster executes causes cores in the other cluster to wake from *Wait For Event* (WFE).

The event signals are also cross-wired between the application processor clusters and the Cortex-M3 *System Control Processor* (SCP). SEV instructions that are executed on the SCP wake application processor clusters from a WFE. Similarly, SEV instructions that are executed on the application processor clusters wake the SCP from a WFE.

In the ARMv8-A Architecture, if the global exclusive monitor for a cluster changes from the *Exclusive* state to the *Open* state, an event for that cluster is generated. In the ADP, the DMC-400 implements the global exclusive monitor. The Cortex-A72 and Cortex-A53 clusters provide an interface to receive notification that this global exclusive monitor has made this state. To support this v8 behavior, an event signal from the DMC-400 connects to the Cortex-A72 and Cortex-A53 clusters.

2.3 Mali-T624 Graphics Processing Unit

The Mali-T624 GPU is a high-performance hardware accelerator for 2D and 3D graphics systems.

The GPU consists of:

- A job manager that controls the graphics processing that is sent to the GPU.
- Four *Shader cores* that perform all the rendering and computation operations.
- A hierarchical tiler that lists all the objects in a scene, so that the shader cores can process the objects efficiently.
- A *Memory-Management Unit* (MMU) that performs address translation of data reads and writes from components in the system.
- A *Power-Management Unit* (PMU).

The GPU uses multiple texture formats that you must enable.

The GPU and its associated software are compatible with the following standards:

- OpenCL 1.1.
- OpenGL ES 1.1, 2.0, and 3.0.
- OpenVG 1.1.
- EGL 1.4.

The Mali-T624 GPU contains a 256KB L2 RAM configuration. For more information about the Mali-T624 GPU, see the following:

- *Mali™-T600 Series GPU Technical Overview*.
- *Mali™-T600 Series GPU Technical Reference Manual*.

2.4 Memory

The following sections describe the memory organization in the ADP:

- *Application memory map*.
- *CoreLink DMC-400 Memory Controller*.
- *Memory bus architecture*.
- *TrustZone system design*.

2.4.1 Application memory map

See *Application memory map summary* on page 3-21.

2.4.2 CoreLink DMC-400 Memory Controller

The DMC-400 is a memory controller that is designed for use with high-performance systems based around ARM processor and GPU cores. It offers many fully programmable *Quality of Service* (QoS) settings and supports DDR3 JEDEC-compatible memory devices.

The DMC-400 supports dual 32-bit DDR3 memory. The memory controller contains two interfaces to the CCI-400, two asynchronous QVN AXI slave interfaces, that connect non-coherent masters, and two DFI 2.1-compliant interfaces for DRAM PHYs. For more information about the DMC-400, see the *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual*.

2.4.3 Memory bus architecture

The memory system in the ADP utilizes advanced QoS methods to manage latency and bandwidth for the components and interfaces.

The bus architecture is based on the AMBA4 protocol and provides a low latency access path to the external memory device from the processors in the system and also provides sufficient bandwidth to the high-bandwidth components. It is based on the CoreLink NIC-400 Network Interconnect and CCI-400 Cache Coherent Interconnect. See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual*.

2.4.4 TrustZone system design

The ADP design provides security features at the system level to facilitate the use of trusted software on the system. It provides a base set of security infrastructure within the ADP. The ADP memory system includes features that provide compliance with the *ARM Trusted Base System Architecture PDD*. See the following:

- *Security* on page 2-20.
- *CoreLink TZC-400 TrustZone Address Space Controller security* on page 2-22.
- *ARM® Trusted Base System Architecture CLIENT1 Platform Design Document*.

2.5 Application processor peripherals

The ADP contains the peripherals that are required to support a Rich OS. Other required peripherals are located outside the ADP. The peripherals encompass essential system-based timers.

This section contains the following subsections:

- *USB 2.0 Host controller* on page 2-9.
- *Direct Memory Access (DMA) controller* on page 2-10.
- *HDLCD controller* on page 2-11.
- *Coherency support for non-ACE-Lite masters* on page 2-12.
- *Static Memory Controller (SMC), PL354* on page 2-12.
- *I2S interface* on page 2-13.
- *I2C interface* on page 2-14.
- *UART* on page 2-15.
- *System override* on page 2-16.

Table 2-2 shows all peripherals that are accessible to all application processors using the Application memory, excluding debug components that are mapped to the CoreSight AXI and APB areas. See *Application processor memory map* on page 3-10.

Table 2-2 Application processor peripherals

Name	Description
NIC-400 GPV	The NIC-400 programmers model. See: • *SoC Interconnect NIC-400 Registers* on page 3-50. • *Compute subsystem NIC-400 Registers* on page 3-63.
ADP System Security Control Registers	-
AON_REF_CLK CNTControl	**AON_REF_CLK** Generic Counter, Control Registers.
AON_REF_CLK CNTRead	**AON_REF_CLK** Generic Counter, Status Registers.
Application processor **AON_REF_CLK** CNTCTL	Application processor **AON_REF_CLK** Generic Timers, Control Registers. See *Application processor AON_REF_CLK Generic Timer* on page 2-30.
Application processor **AON_REF_CLK** CNTBase0	Application processor **AON_REF_CLK** Generic Timers, Control Registers. See *Application processor AON_REF_CLK Generic Timer* on page 2-30.
Application processor **AON_REF_CLK** CNTBase1	
Trusted watchdog	Trusted Watchdog. See the *ARM® Watchdog Module (SP805) Technical Reference Manual*.
Generic Watchdog	Generic Watchdog. See the *ARM® Server Base System Architecture Platform Design Document*.
TZC-400	TZC-400 TrustZone Address Space Controller. See *CoreLink TZC-400 TrustZone Address Space Controller security* on page 2-22.
DMC_CFG	Configuration interface of the DMC-400. See the *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual*.
SCP_MHU	Message Handling Unit.
Juno System Profiler	System Profiler.

Table 2-2 Application processor peripherals (continued)

Name	Description
GPU0_SMMU	System MMU, MMU-400, components. See *CoreLink MMU-401 and MMU-400 System Memory Management (SMMU) components* on page 2-18.
PCIe SMMU	
ETR_SMMU	
GIC distributor	CoreLink GIC-400 Generic Interrupt Controller. See *Application processor interrupt map* on page 3-3 and the *ARM® GIC-400 Generic Interrupt Controller Technical Reference Manual*.
GIC physical core interface	
GIC virtual interface control	
GIC virtual core interface	
GICv2m MSI	Handles PCI Express *Message Signaled Interrupts* (MSIs).
CCI_PV	CoreLink CCI-400 Cache Coherent Interconnect. See the *ARM® CoreLink™ CCI-400 Cache Coherent Interconnect Technical Reference Manual*.
Mali-T624 GPU	GPU register interface. See the *Mali™-T600 Series GPU Technical Reference Manual*.
USB controller	*USB 2.0 Host controller*
DMA controller	*Direct Memory Access (DMA) controller* on page 2-10
HDLCD controller	*HDLCD controller* on page 2-11
Coherency	*Coherency support for non-ACE-Lite masters* on page 2-12
SMC	*Static Memory Controller (SMC), PL354* on page 2-12
I²S interface	*I2S interface* on page 2-13
I²C interface	*I2C interface* on page 2-14
UART	*UART* on page 2-15
System Override Registers	*System override registers* on page 3-40
System Configuration Controller	-

2.5.1 USB 2.0 Host controller

The ADP supports USB 2.0 with a data rate of 480Mbps, and can connect common system peripherals such as keyboard, mouse, and flash drive. The USB 2.0 Host Controller has the following features:

- Fully compliant with the *Enhanced Host Controller Interface* (EHCI) specification.
- Contains a single EHCI controller that supports USB high-speed data rates of 480Mbps.
- Contains a single companion *Open Host Controller Interface* (OHCI) controller that supports USB full and low speed device data rates.
- The USB 2.0 Host Controller is on the chip and connects to a USB PHY on the board using a 60MHz 12-pin *UTMI+ Low-Pin Interface* (ULPI) SDR interface.
- Debug is an optional feature in EHCI, and the USB 2.0 Host Controller does not support debug.

- 64-bit EHCI addressing capability is enabled on the ADP to be compatible with ARMv8-A.
- Legacy emulation interface and PCI power-management features of the USB Host Controller IP that the ADP does not use.

2.5.2 *Direct Memory Access* (DMA) controller

The ADP includes a system CoreLink DMA-330 controller as a master on the compute subsystem I/O coherent slave interface. The DMA controller can transfer data within memory, or between memory and peripherals.

The DMA-330 implements TrustZone. The **boot_manager_ns** input signal on the DMA-330 controls the security state of the device as follows:

`0` When the pin is 0, the DMA manager is in the Secure state and the Secure APB interface issues instructions. The ADP ignores any instructions that the Non-secure APB interface issues.

`1` When the pin is 1, the DMA manager is in the Non-secure state, and either the Secure or the Non-secure APB interface can control the DMA channel.

In the ADP, the DMA_BOOT_MANAGER_NS bit of the DMA Control Register 0 controls this pin.

Table 2-3 shows the DMA-330 configuration parameters.

Table 2-3 DMA-330 configuration

DMA-330 configuration	Value
AXI data width	128
Icache lines	16
Icache words	8
Number of channels	8
Write issuing capacity	8
Read issuing capacity	8
Read LSQ depth	16
Write LSQ depth	16
FIFO depth	1024
Number of interrupts	8
Number of peripheral request interfaces	8

The system DMA controller also includes peripheral request interfaces to support transfer between peripherals and memory, without the intervention of the processor. Table 2-4 shows the peripheral request IDs.

Table 2-4 DMA-330 peripheral request interface mapping

Peripheral request ID	ADP peripheral
0	STM
1	UART0 RX
2	UART0 TX
3	UART1 RX
4	UART1 TX
5	I^2S
6	External source, FPGA
7	Reserved

2.5.3 HDLCD controller

The ADP includes two independent HDLCD display controllers. Table 2-5 shows the display resolutions that are supported, and the pixel clock frequencies required.

Table 2-5 Display resolutions

Type	Resolution	Frame rate	Pixel clock frequency
VGA	640 × 480	60Hz	23.75MHz
UXGA	1600 × 1200	60Hz	63.5MHz
Full HD	1920 × 1080	60Hz	148.5MHz

HDLCD controller configuration

Table 2-6 shows the HDLCD controller configuration parameters.

Table 2-6 HDLCD controller configuration parameters

Parameter	Default	Juno setting	Description
`FIFO_ADDRESS_BITS`	8	9	Number of address bits to access the pixel data FIFO. FIFO size is 8 bytes × 2^9 = 4KB.
`COUNTER_BITS`	11	12	Number of bits used for all timing counters and configuration registers. The maximum resolution that is supported is $2^{12} \times 2^{12} = 4096 \times 4096$ and this resolution is sufficient for QFHD and 4K2K.

2.5.4 Coherency support for non-ACE-Lite masters

The DMA and USB masters are AXI3 masters and do not support ACE-Lite natively. To enable these masters to snoop the hardware coherent caches of the CPUs, Juno supports a limited form of ACE-Lite signaling. Signaling is performed by driving the **AxDOMAIN** signals appropriately from the MMU-401 associated with the master.

This scheme supports a restricted set of ACE-Lite transactions, that is, only the ACE-Lite transactions that are compatible with AXI4 and NIC-400. Table 2-7 and Table 2-8 show the supported types of ACE-Lite transactions for write and read transactions. For this scheme to work, the only extra signal an AXI3 master requires is the shareability domain, and the MMU-401 supports this scheme.

Table 2-7 shows the permitted ACE-Lite write transactions.

Table 2-7 Permitted ACE-Lite write transactions

Type	AWSNOOP[2:0]	AWDOMAIN[1:0]	Transaction
Non-coherent	0b000	0b00, 0b11	WriteNoSnoop
Coherent	0b000	0b01, 0b10	WriteUnique

Table 2-8 shows the permitted ACE-Lite read transactions.

Table 2-8 Permitted ACE-Lite read transactions

Type	ARSNOOP[3:0]	ARDOMAIN[1:0]	Transaction
Non-coherent	0b0000	0b00, 0b11	ReadNoSnoop
Coherent	0b0000	0b01, 0b10	ReadOnce

2.5.5 *Static Memory Controller* (SMC), PL354

The static memory interface in the ADP supports NOR memory natively and other peripherals through the FPGA on the board. The NOR flash is intended to be a simple boot mechanism during early board bring-up activities.

The PL354 supports two SRAM style memory interfaces. Each memory interface supports four chip selects, providing eight chip selects in total. Juno uses six of the eight available chip selects. To reduce the pin count on the package, the two memory interfaces on the PL354 share the set of test chip pins using the *External Bus Interface* (EBI), PL220, block. The EBI handles the required arbitration between the memory interfaces. The PL354 configuration space is memory-mapped at 0x00_7FFD_0000.

Table 2-9 on page 2-13 shows the default address ranges for the chip selects. However, you can change the address range by programming appropriate address mask and match values in the SCC registers using the serial interface when the chip is held in reset. Select the address mask and match values so that no address maps to more than one chip select.

CoreLink SMC-354 Static Memory Controller (SMC) on page A-7 describes the PL354 configuration parameters.

Table 2-9 shows the PL354 chip selects and address ranges.

Table 2-9 PL354 chip selects and address ranges

Chip select	Address range	Board peripherals
CS0	0x00_0800_0000-0x00_0BFF_FFFF	NOR
CS4	Reserved	-
CS5	Reserved	-
CS1	Reserved	-
	0x00_1400_0000-0x00_1403_FFFF	PSRAM, 256kB
CS2	0x00_1800_0000-0x00_1BFF_FFFF	SMC Ethernet
CS3	0x00_1C00_0000-0x00_1EFF_FFFF	IOFPGA system peripherals
CS6	Not used	-
CS7	Not used	-

2.5.6 I²S interface

The ADP includes an I²S interface that supports four-channel stereo audio output for high-quality audio to the external display. The I²S bus only handles audio data, and so the control signaling is handled through a separate I²C bus.

The ADP board includes two HDMI connectors for attaching displays. The DisplayPort supports 8-channel audio streams with sampling rates up to 24 bits at 192kHz. These values dictate the maximum audio resolution that is configured on the Juno I²S controller.

Stereo data pairs, left and right, that the processor writes using the APB interface, are shifted out serially in the appropriate data output, SD0, SD1, SD2, or SD3. The shifting is timed with respect to **I2S_CLK**. WS is the word select line and selects between left and right data.

The following signals gate the clock output from the ADP to the slave device:

SCLK_GATE Clock enable signal that disables the output clock when the controller has been disabled.

SCLK_EN Gating signal when data resolution of the transmit channel is less than the current word select size.

Figure 2-3 shows the I²S integration.

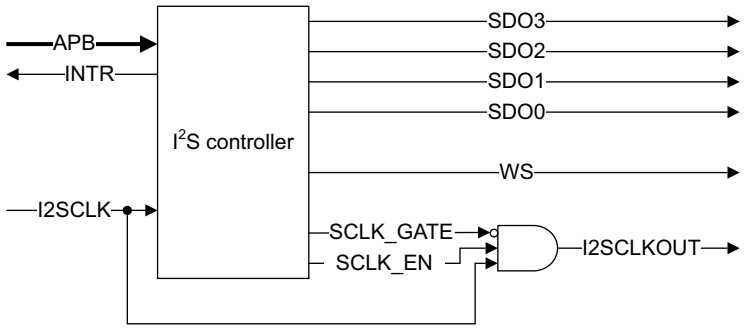


Figure 2-3 I²S integration

The I2S used in the ADP is APB mapped. The key features are as follows:

- Four stereo transmit channels that are configured with a maximum audio resolution of 24. If necessary, you can reprogram the transmit channels during operation to any supported resolution lesser than 24.
- Word select length is set to 24. You can reprogram the word select length during operation to supported values.
- **I2S_CLK** is asynchronous to APB clock.

2.5.7 I2C interface

The ADP includes the following I2C controllers:

1. An I2C that is mapped in the SCP AHB expansion area and exclusively controls the PMIC.
2. An I2C that is mapped in the application processor area and performs board functions such as HDMI controller configuration.
3. An I2C that is mapped in the application processor area and exclusively controls trusted user input from a keypad.

The I2C controller that is used in the ADP is an APB mapped peripheral.

The key features are as follows:

- Two wire, *Serial Data Line* (SDA) and *Serial CLock* (SCL).
- IP configured to support the following speed modes:

  Standard Up to 100Kbps.

  Fast Up to 400Kbps.

  High-speed High-speed mode, up to 3.4Mbps, is disabled.
- Interrupt and polled mode operation is supported.
- 7-bit addressing selected.
- The I2C IP can operate in either master or slave mode, but not both. For the ADP, all three I2C instances operate in master mode.
- The I2C controller requires that **ic_clk** is the same frequency or faster than APB interface clock. To guarantee that this requirement is always met, in the ADP, the primary input clock, **I2CCLK**, drives both **ic_clk** and the APB interface. An asynchronous APB bridge is then added to ensure that the APB interface can be operated asynchronously with rest of the system.

Table 2-10 shows the minimum **ic_clk** values for Standard and Fast modes, with high, and low count values for SCL generation. The maximum suppressed spike length registers **IC_FS_SPKLEN** and **IC_HS_SPKLEN** are programmed to 1. In the ADP, **ic_clk** is the same as the APB interface clock, that is, 100MHz, and considerably higher than the minimum **ic_clk** frequency permitted.

Table 2-10 Minimum ic_clk frequency

Speed mode	ic_clk (MHz)	SCL low count	SCL low program value	SCL low time	SCL high count	SCL high program value	SCL high time
SS	2.7	13	12	4.7µs	14	7	5.2µs
FS	12.0	16	15	1.33µs	14	7	1.16µs

Figure 2-4 shows the pad connections for the I2C interface. You can use the PBIDIR pad from the 28HPM I/O library for the I2C signals.

ic_clk_oe and **ic_data_oe** are the I2C clock and data outputs respectively. Both outputs are open drain signals and require external pull-up on the board.

ic_clk_in and **ic_data_in** are incoming I2C clock and data.

Figure 2-4 shows the I2C interface pad connections.

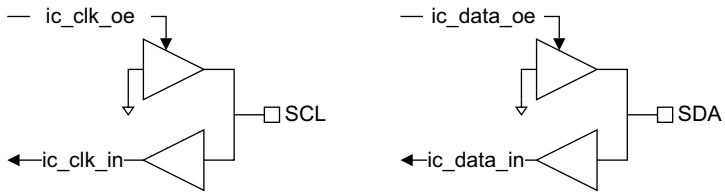


Figure 2-4 I2C interface pad connections

2.5.8 UART

The ADP includes UARTs, SoC UART 0, and SoC UART 1, for:

- Firmware messages.
- Debugging OS kernel.

The UART signals connect to RS232 connectors on the board. Figure 2-5 shows the PL011 UART integration.

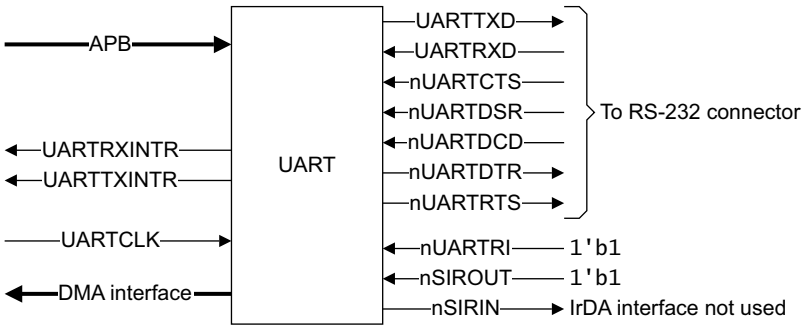


Figure 2-5 UART integration

The key features are as follows.

- The PL011 UART provides a DMA request interface that, in the ADP, is connected to the DMA-330 system DMA controller.
- IrDA transmit and receive signals are not used.
- **UART_CLK** is a primary input to the ADP and is asynchronous to APB clock.

Figure 2-5 on page 2-15 shows the UART integration.

The UART supports a wide range of baud rates depending on the frequency of the input **UART_CLK** in addition to the integer, UARTIBRD, and fractional, UARTFBRD, divisor registers in the PL011 UART.

Table 2-11 shows the typical Baud rates and the divisor values, with a **UART_CLK** clock frequency of 7.3728MHz from the crystal. The fractional divisor is not used in this specific case. If a non-friendly **UART_CLK** frequency is used, there can be slight error in the generated Baud rate when compared to the required Baud rate.

Table 2-11 Baud rates

Integer divisor	Baud rate
0x1	460800
0x2	230400
0x4	115200
0x6	76800
0x8	57600
0xC	38400
0x18	19200
0x20	14400
0x30	9600

2.5.9 System override

You can use system override registers to override aspects of the system outside the compute subsystem, such as the security state of peripherals and the shareability settings of masters that do not support ACE-Lite signaling. See *System override registers* on page 3-40. The System Override Registers Unit is an APB peripheral in the application processor memory map. The base address of the peripheral is 0x7FFF_0000.

It is a Secure access only peripheral. If a Non-secure access attempts to access any of these registers, the interconnect responds with a DECERR response. Any Secure access to unimplemented areas within the 4K region is RAZ, WI. The block also includes the following general purpose registers for software use:

- GPR_0.
- GPR_1.

2.6 Interconnect and memory system

This section describes the interconnect and memory system components:

- *CoreLink NIC-400 Network Interconnect*.
- *CoreLink MMU-401 and MMU-400 System Memory Management (SMMU) components* on page 2-18.
- *CoreLink DMC-400 Dynamic Memory Controller* on page 2-18.

2.6.1 CoreLink NIC-400 Network Interconnect

CoreLink NIC-400 Network Interconnect components connect other system components into the CCI-400. For example, NIC-400s connect master components to CCI-400 slave ports, and slave components to CCI-400 master ports.

Two separate NIC-400 components:

- Connect compute subsystem components to the CCI-400 and provide access for masters and slaves outside the compute subsystem.
- Connect SoC-level peripheral components, that is, components that are outside the compute subsystem, to the compute subsystem.

See the following:

- Figure 2-1 on page 2-2.
- *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual*.

2.6.2 CoreLink CCI-400 Cache Coherent Interconnect

CoreLink CCI-400 Cache Coherent Interconnect components connect other system components.

The CCI-400 has the following ports:

- Slave port S0 which connects to the master interface of the PCIe Root Complex.
- Slave port S1 which connects, through NIC-400 interconnect, to the SCP, CoreSight Debug system, USB 2.0 interface, DMA-330, and the external Thin Links IC-FPGA slave extension interface.
- Slave port S2 which connects to the Mali-T624 GPU.
- Slave port S3 which connects to the Cortex-A53 cluster.
- Slave port S4 which connects to the Cortex-A72 cluster.
- Master port M0 which connects to the slave interface of the PCIe Root Complex.
- Master port M1 which connects, through NIC-400 interconnect, to DMC-400 and on-chip memories, various Compute Subsystem peripherals, NOR flash interface, I^2S, I^2C, Dual UARTs, various security components and the external Thin Links IC-FPGA master extension interface.
- Master port M2 which connects to the DMC-400.

See the following:

- Figure 2-1 on page 2-2.
- *ARM® CoreLink™ CCI-400 Cache Coherent Interconnect Technical Reference Manual*.

2.6.3 CoreLink MMU-401 and MMU-400 *System Memory Management* (SMMU) components

The ADP includes system *Memory Management Units* (MMUs) to support system virtualization, where guest *Operating Systems* (OSs) run on a hypervisor and are only aware of *Intermediate Physical Addresses* (IPAs). The system MMU components handle stage 2 address translations, that is, IPA to *Physical Address* (PA).

CoreLink MMU-401 components are located in-front of the following:

- I/O coherent PCI Express Root Port.
- CoreLink CCI-400.
- SCP switch in the NIC-400.
- DMA controller, DMA-330.
- Two HDLCD controllers.
- USB host controller.

A CoreLink MMU-400 is located between the Mali-T624 GPU and the CCI-400.

Virtual memory maintenance does not use *Distributed Virtual Memory* (DVM) transactions. Instead, in the ADP, the translation tables are maintained in software by writing to the APB4 programming interface of the MMU blocks.

DMA virtualization information:

- The DMA-330 has one manager thread and eight channel threads. The DMA outputs the channel information using the AXI **ARID[3:0]** and **AWID[3:0]** signals. For example, an **AxID** value of `0x0` corresponds to channel 0, and an **AxID** value of `0x1` corresponds to channel 1, until the **AxID** value of `0x7` correspond to channel 7. Transactions belonging to the manager thread have an **AxID** value of `0x8`.
- The MMU-401 in-front of the DMA-330 includes a *Security State Determination* (SSD) table containing nine entries. The **AxID[3:0]** signal performs the SSD table indexing. An **AxID** value of 0 corresponds to SSD entry 0, an **AxID** value of 1 corresponds to SSD entry 1, until the **AxID** value of 8 correspond to SSD entry 8.
- The MMU-401 in-front of the DMA-330 supports four contexts. The **AxID** of incoming transaction is compared with the eight stream matching registers to map to one of the four contexts.

CoreLink MMU-40x System Memory Management Unit (SMMU) components on page A-5 shows the MMU-40x configuration options.

See the following:

- *ARM® CoreLink™ MMU-400 System Memory Management Unit Technical Reference Manual*.
- *ARM® CoreLink™ MMU-401 System Memory Management Unit Technical Reference Manual*.

2.6.4 CoreLink DMC-400 Dynamic Memory Controller

The CoreLink DMC-400 Dynamic Memory Controller connects the ADP memory system to the DDR3 PHY that provides access to the SDRAMs.

See:

- *CoreLink DMC-400 Memory Controller* on page 2-7

- The *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual*.

2.7 Security

This section describes:

- *About security*.
- *Trusted Base System Architecture (TBSA) compliance*.
- *ADP device security model*.
- *Trusted entropy source* on page 2-24.
- *Trusted root-key storage* on page 2-25.
- *Trusted non-volatile counters* on page 2-26.
- *Trusted HDLCD controller* on page 2-27.
- *Secure on-chip memories* on page 2-28.
- *Interconnect IP* on page 2-28.

2.7.1 About security

The section describes the support that the ADP platform provides for the Secure platform. Level 1 of the *Trusted Base System Architecture CLIENT1 Platform Design Document* (PDD) defines the Secure platform.

2.7.2 *Trusted Base System Architecture* (TBSA) compliance

Although the ADP intends to comply with TBSA v1, there are some differences to keep the platform design simple and to support more use cases.

- The HDLCD controller does not support the issuing of Secure accesses. You can force it to support Secure accesses by driving the **AxPROT[1]** bits from Secure memory mapped registers.
- Trusted root key storage requires fuses. The ADP implements key storage using registers with the key values tied-off in hardware.
- Trusted non-volatile counters are implemented as fuses using registers as follows:
  - A 31 state counter is hardwired with a value of 31.
  - A 223 state counter is hardwired with a value of 223.
  - A Secure storage counter is supported off-chip through the Secure *Inter-Integrated Circuit* (I^2C) interface.
- The ADP contains the following ROMs:
  - Secure SCP.
  - Secure AP.
  - Non-secure AP.

  Because the ADP is a development chip, it includes a mechanism to bypass the boot ROMs, and fetch the code directly from the external NOR flash memory. This alternative boot is enabled by loading special values in the alternate boot registers in the SCC.

2.7.3 ADP device security model

The following subsections provide additional information to the TBSA, and focus on key devices, modules, and interfaces in the system that contribute to security:

- *Application processor security* on page 2-21.
- *CoreLink GIC-400 Generic Interrupt Controller security* on page 2-21.
- *Interconnect security* on page 2-21.
- *CoreLink TZC-400 TrustZone Address Space Controller security* on page 2-22.

- *Trusted on-chip boot ROM* on page 2-23.
- *Trusted on-chip SRAM* on page 2-23.
- *DDR memory controller and DFI interfaces* on page 2-23.
- *Watchdog security* on page 2-23.

Application processor security

The application processors in the ADP support the ARMv8-A architecture including the Security Extensions.

Note

The **CP15SDISABLE** input signals to the Cortex-A72 and Cortex-A53 clusters are tied LOW.

See the following:

- *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual*.
- *ARM® Cortex®-A53 MPCore Processor Technical Reference Manual*.

CoreLink GIC-400 Generic Interrupt Controller security

A CoreLink GIC-400 Generic Interrupt Controller that implements the GICv2 architecture is present, and all application processors in the ADP share it. The GIC-400 supports Security Extensions.

Security Extensions enable the GIC-400 to support the following:

- Banking of key registers to provide the following views:
  - Trusted world view.
  - Non-trusted world view.
- Configuring each interrupt as either:
  - Secure, trusted world.
  - Non-secure, non-trusted world.
- Signaling Secure interrupts to the target processor using either the IRQ or the FIQ exception request.
- A unified scheme for handling the priority of Secure and Non-secure interrupts.
- Optional lockdown of the configuration of some Secure interrupts.

The CoreLink GIC-400 supports 16 *Software Generated Interrupts* (SGIs). You can program these interrupts to implement the following concurrently:

- Eight Non-secure software-generated interrupts.
- Eight Secure software-generated interrupts.

See the *ARM® Generic Interrupt Controller Architecture Specification, Architecture version 2.0*.

Interconnect security

Within the ADP, one of the following places peripherals into Secure or Non-secure world:

- The peripheral itself, interpreting the NS flag that is provided by the master that made the request.
- The NIC-400 either:
  - Permanently as Secure or Non-secure.

— Software-configurable using the programmers model registers of the NIC-400.

For information on how the peripherals are security-mapped, see *Application processor memory map* on page 3-10.

The NIC-400 registers that affect security are Secure access only.

For more information about the NIC-400 registers, see the following:

- *SoC Interconnect NIC-400 Registers* on page 3-50.
- *Compute subsystem NIC-400 Registers* on page 3-63.
- *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual*.

CoreLink TZC-400 TrustZone Address Space Controller security

TrustZone is the security architecture that ARM developed. The TZC-400 TrustZone Address Space Controller is an important component in systems that use the TrustZone architecture. The TZC-400 controls access to address regions and is programmed to enable specific masters to access particular regions of memory, and block access from other masters.

In the ADP, the TZC-400 is located between the CoreLink CCI-400 Cache Coherent Interconnect, and the CoreLink DMC-400 Dynamic Memory Controller. The TZC-400 enables the Trusted OS to define multiple regions within the DDR memory that have different security access permissions. The configuration of the TZC-400 in the ADP is as follows:

- Four filter units, one for each ACE-Lite interface of the DMC-400. These filter units share region configurations.
- Nine regions, that include one base region, region 0, and eight fully software-configurable regions, regions 1-8. These regions enable you to define up to eight independent regions with different security requirements.
- The ADP supports 13 *Non-Secure Access IDentifier* (NSAID) values, including the default ID, and enables up to 12 masters, or groups of masters, to be uniquely identified so that they can be given different Non-secure access permissions. NSAID values are mapped as follows:

  `0` Default ID that any masters use that are not necessary to be identified separately from the rest, including CCI-400. This value is used by default.

  Note

  The CCI-400 does not store the NSAID values of cache lines that it temporarily stores. As a result, if the CCI-400 performs line write backs, it does not reinstate the NSAID values of the original accesses. Therefore, you must not set Non-secure memory areas that are protected by NSAID as shareable. Otherwise, security violations can occur.

  `1` PCIe.

  `2` HDLCD0.

  `3` HDLCD1.

  `4` USB.

  `5` DMA.

  `6` Thin Links.

  `7-8` Reserved. Do not use.

  `9` Application processors.

10	Mali-T624 GPU.
11	SCP.
12	All CoreSight accesses.
13-15	Reserved. Do not use.

Note

When programming the TZC-400 regions, ensure that all regions are 64KB-aligned, and that each region has a size that is at least 64KB, or a multiple of 64KB, to comply with the TBSA. See the *ARM® Trusted Base System Architecture PDD*.

See the following:

- *ADP device security model* on page 2-20.
- *ARM® CoreLink™ TZC-400 TrustZone® Address Space Controller Technical Reference Manual*.

Trusted on-chip boot ROM

The following Secure access only on-die boot ROM is implemented in the ADP:

- *Trusted boot ROM for application processor boot code*.
- *Trusted boot ROM for the SCP*.

Trusted boot ROM for application processor boot code

The trusted boot ROM for application processor boot code is permanently mapped to address `0x00_0000_0000`, the reset vector for the application processors, and is on-chip.

Trusted boot ROM for the SCP

An on-chip trusted boot ROM for the SCP is also provided.

Trusted on-chip SRAM

Application processor trusted software can use a Secure access only on-die RAM area. The NIC-400 is configured to make this SRAM Secure access only.

DDR memory controller and DFI interfaces

The DDR Memory Controller, DMC-400, and the DDR PHY interface attached to the ADP do not directly support TrustZone technology-based security. However, the NIC-400 maps the configuration register spaces for the DMC-400 and the DDR3 PHY as software-configurable and defaults at reset to permit only Secure accesses.

Watchdog security

The ADP provides two Secure watchdog timers that enable you to force an automatic hardware reset of the whole system when they time out. The watchdog timers are:

- The watchdog timer in the SCP subsystem. Because it is in the SCP subsystem, only Secure software can access it.
- The trusted watchdog in the application processor system.

Both watchdogs are accessible by Secure accesses only.

Note

A Generic Watchdog also exists in the application processor system, that both Secure and Non-secure software can access. This watchdog cannot cause an automatic hardware reset.

See:

- *Watchdog timers* on page 2-31.
- *Cross triggers* on page 2-37.
- *Trusted entropy source*.
- *Trusted root-key storage* on page 2-25.
- *Trusted non-volatile counters* on page 2-26.
- *Trusted HDLCD controller* on page 2-27.

2.7.4 Trusted entropy source

The ADP includes a Trusted Entropy Source that is based on a design from ARM Research and Development. The Trusted Entropy Source generates one 128-bit random number at a time. Software then generates 512 different random numbers that are based on that seed.

See *Trusted Entropy Source Registers* on page 3-139.

Flow diagrams

The flow diagrams in this section provide a general idea on how to program the RNG peripheral and generate the 128-bit random number value.

Configuration for interrupt-based

Figure 2-6 shows the configuration flow chart.

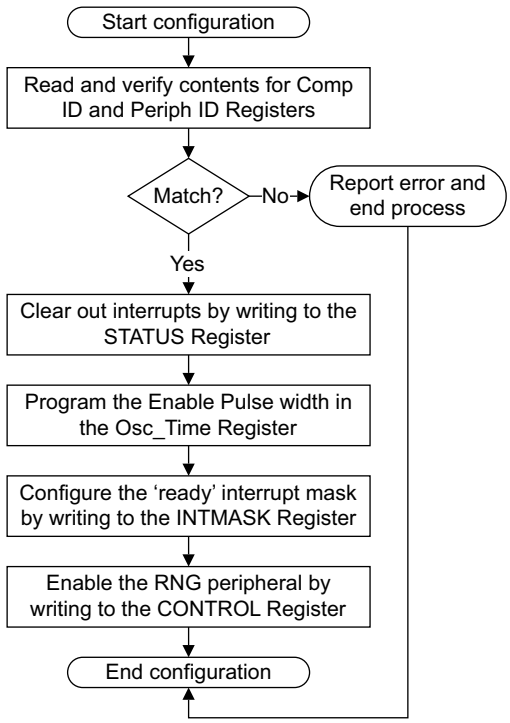


Figure 2-6 Configuration flow chart

Interrupt Service Routine (ISR)

Figure 2-7 shows the *Interrupt Service Routine* (ISR) flow chart.

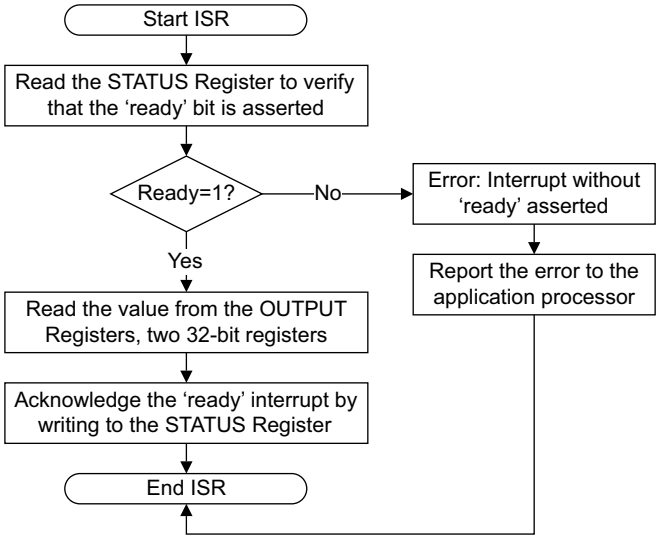


Figure 2-7 ISR flow chart

2.7.5 Trusted root-key storage

The Trusted Root-Key Storage is implemented as an APB mapped register bank with the programmers view that Table 2-12 shows. The Secure platform certification process requires that the reset value of these registers is always the same, and the certification requires that the values are hardcoded.

If a Non-secure access attempts to access any of these registers, the response is a DECERR. Any Secure access to unimplemented areas within the 4K region is RAZ, WI.

Table 2-12 Trusted root-key storage registers

Offset	R/W	Reset value	Name	Description
0x00	R	0xB2043562	TZ_PUB_KEY_HASH_0	Lowest 32 bits [31:0] of 256-bit *Trust Public Key* (ROTPK)
0x04	R	0x1B702EFD	TZ_PUB_KEY_HASH_1	Intermediate bits of 256-bit ROTPK
0x08	R	0x930AEA03	TZ_PUB_KEY_HASH_2	-
0x0C	R	0x648C4055	TZ_PUB_KEY_HASH_3	-
0x10	R	0x4365DFF1	TZ_PUB_KEY_HASH_4	-
0x14	R	0x0D6EEB27	TZ_PUB_KEY_HASH_5	-
0x18	R	0xE43AEDCB	TZ_PUB_KEY_HASH_6	-
0x1C	R	0xED61412A	TZ_PUB_KEY_HASH_7	Highest 32 bits [255:224] of 256-bit ROTPK
0x20	R	0xFC1C5C16	HU_KEY_0	Lowest 32 bits [31:0] of 128-bit *Hardware Unique Key* (HUK)
0x24	R	0x0D34E9D0	HU_KEY_1	Intermediate bits of 128-bit HUK
0x28	R	0xA573A314	HU_KEY_2	-
0x2C	R	0x666E5E53	HU_KEY_3	Highest 32-bits [127:96] of 128-bit HUK

Table 2-12 Trusted root-key storage registers (continued)

Offset	R/W	Reset value	Name	Description
0x44	R	0x57AA4A40	END_KEY_0	Lowest 32 bits [31:0] of 256-bit *Private Endorsement Key* (PEK)
0x48	R	0x2E0E0993	END_KEY_1	Intermediate bits of 256-bit PEK
0x4C	R	0x8606B765	END_KEY_2	-
0x50	R	0x08F0220F	END_KEY_3	-
0x54	R	0x28C80B76	END_KEY_4	-
0x58	R	0x58E55816	END_KEY_5	-
0x5C	R	0x2CF04CC5	END_KEY_6	-
0x60	R	0xE7D5DDE1	END_KEY_7	Highest 32 bits [255:224] of 256-bit PEK
0xFD0	RO	0x04	PID4	Peripheral ID 4
0xFE0	RO	0xA8	PID0	Peripheral ID 0
0xFE4	RO	0xB0	PID1	Peripheral ID 1
0xFE8	RO	0x0B	PID2	Peripheral ID 2
0xFEC	RO	0x00	PID3	Peripheral ID 3
0xFF0	RO	0x0D	COMPID0	Component ID 0
0xFF4	RO	0xF0	COMPID1	Component ID 1
0xFF8	RO	0x05	COMPID2	Component ID 2
0xFFC	RO	0xB1	COMPID3	Component ID 3

2.7.6 Trusted non-volatile counters

The trusted non-volatile counters are implemented as an APB-mapped register bank with the programmers view that Table 2-13 shows. The Secure platform certification process requires that the reset value of these registers is always the same, and requires the values to be hardcoded.

Any Non-secure accesses to these registers receive a DECERR response.

Any Secure accesses to unimplemented areas within the 4K region are RAZ, WI.

Table 2-13 Trusted non-volatile counters

Offset	R/W	Reset value	Register name	Description
0x00	R	0x1F	TZ_FW_CNT	Trusted Firmware Updates Counter, as required by R39.a from TBSA v1, tied to 31
0x04	R	0xDF	NON_TZ_FW_CNT	Non-trusted Firmware Updates Counter, as required by R74.b from TBSA v2, tied to 223
0xFD0	RO	0x04	PID4	Peripheral ID 4
0xFE0	RO	0xA9	PID0	Peripheral ID 0
0xFE4	RO	0xB0	PID1	Peripheral ID 1
0xFE8	RO	0x0B	PID2	Peripheral ID 2

Table 2-13 Trusted non-volatile counters (continued)

Offset	R/W	Reset value	Register name	Description
0xFEC	RO	0x00	PID3	Peripheral ID 3
0xFF0	RO	0x0D	COMPID0	Component ID 0
0xFF4	RO	0xF0	COMPID1	Component ID 1
0xFF8	RO	0x05	COMPID2	Component ID 2
0xFFC	RO	0xB1	COMPID3	Component ID 3

2.7.7 Trusted HDLCD controller

The HDLCD controller must be able to display trusted frame data and the controller must read trusted frame buffer using Secure access.

The ARM HDLCD controller does not support security. In the ADP, the SSD table present in the MMU-401 determines the security state of the HDLCD controller. Secure software can control the security state of the HDLCD Controller by causing it to transition between Secure and Non-secure. If the SSD table entry is marked as Secure, then the Non-secure transactions from the HDLCD controller that arrive at the MMU-401 are converted to Secure transactions before they are sent to the memory.

The ADP also permits the HDLCD transactions to be marked as Secure by writing to the SEC_HDLCD register in the System Override Registers Unit. Writing to the SEC_HDLCD register forces all HDLCD transactions to be Secure. That is, **ARPROT[1]** = 0, independent of the MMU-401 settings. Figure 2-8 on page 2-28 shows this scheme.

The APB interfaces of the HDLCD controllers have programmable security settings on the interconnect. The default state is Secure access only but you can update it by programming the interconnect.

By using this scheme, the *Trusted Base System Architecture Platform Design Document* recommends the following:

- Ensure that the security settings in the MMU-401 SSD table and APB configuration interface of the HDLCD controller are switched at the same time when the display is inactive.
- The mode must not be switched while there are valid AXI transactions on the bus.
- When the HDLCD controller is configured to be Secure, it must point to trusted frame buffer in trusted memory.
- When switching from Secure to Non-secure, any pixel data in the display must be wiped if it is accessible.

Figure 2-8 on page 2-28 shows the trusted display controller.

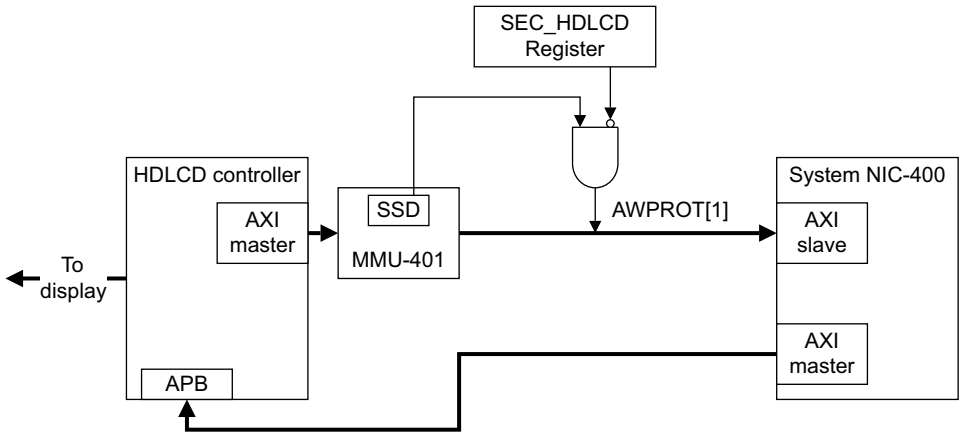


Figure 2-8 Trusted display controller

2.7.8 Secure on-chip memories

The system contains the following individual memory regions in the application memory map that are marked as always Secure access only:

- A memory region for the on-chip ROM.
- A memory region for the on-chip SRAM.

Note

The SCP also contains Secure ROM and SRAM.

2.7.9 Interconnect IP

Only a subset of masters within the system can generate Secure accesses. These masters include:

- The SCP.
- The application processors.
- Debug and trace logic, when it is operating in Secure mode.

Also, software can configure the GPU and displays to generate Secure accesses. Some peripherals are also mapped as Secure access only, and you can use software to configure others to be Secure access only. You can configure the DMA to generate Secure or Non-secure transactions.

2.8 Timers

This section describes:

- *Time domains*.
- *Counter and timer components* on page 2-30.
- *Processor power modes* on page 2-31.
- *Watchdog timers* on page 2-31.

2.8.1 Time domains

The ADP contains the following separate time domains:

AON_REF_CLK time

View of time that the application processors observe. Low-power modes can affect this time domain.

CoreSight timestamp time

View of time that is used exclusively for the generation of CoreSight timestamps.

This section describes:

- *AON_REF_CLK time domain*.
- *CoreSight timestamp time domain* on page 2-30.

AON_REF_CLK time domain

The application processors operate in a time domain called **AON_REF_CLK** time. This time domain is based on the main reference clock, **AON_REF_CLK**, and is also visible to the SCP.

A Generic Counter component, called the **AON_REF_CLK** counter, generates a time value for the **AON_REF_CLK** time domain. The component meets the requirements of the memory-mapped counter module that the *ARM v7AR Architecture Reference Manual* describes. It is in the VSYS.AON power domain.

You can halt the **AON_REF_CLK** time domain during debug.

The **AON_REF_CLK** time domain contains the following timers:

Generic Timer

All application processors in the ADP implement the ARM Generic Timer. Interrupts from these timers are mapped to *Private Peripheral Interrupts* (PPIs) in the CoreLink GIC-400 Generic Interrupt Controller. Accesses to these Generic Timers are through a low-latency system-mapped register.

SCP AON_REF_CLK Generic Timer

The SCP subsystem contains a memory-mapped Generic Timer.

Extra AON_REF_CLK Generic Timers

Two extra **AON_REF_CLK** Generic Timers, one Secure and one Non-secure that the application processors can use.

AON_REF_CLK also drives the Mali-T624 global timestamp counter.

See the following:

- *CoreSight timestamp time domain* on page 2-30.
- *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual*.

- *ARM® Cortex®-A53 MPCore Processor Technical Reference Manual*.
- *ARM® Architecture Reference Manual v7AR Edition*.

CoreSight timestamp time domain

An extra time domain exists that exclusively generates CoreSight timestamps. This time domain is also based on the main reference clock **AON_REF_CLK**, but differs from the **AON_REF_CLK** time domain because it cannot be halted during debug. The SCP can save and restore the value of this counter in the same way that it does for the **AON_REF_CLK** time domain. See *CoreSight timestamps* on page 2-40.

2.8.2 Counter and timer components

This section describes:

- *CoreSight timestamp counter*.
- *Application processor AON_REF_CLK Generic Timer*.
- *Memory map frames* on page 2-31.

CoreSight timestamp counter

The CoreSight timestamp counter is an implementation of the memory-mapped counter module that the *ARM® Architecture Reference Manual* defines. The counter is visible in the SCP memory map, labeled CS CNTControl. The CNTRead counter frame that the counter architecture defines is not made available for this counter. This counter is the basis for CoreSight timestamp generation, that *CoreSight timestamps* on page 2-40 describes.

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

Application processor AON_REF_CLK Generic Timer

The ADP includes two memory-mapped ARM Generic Timers, that the *ARM Architecture Reference Manual v7AR Edition* defines. The Generic Timers provide two timer frames, each without a second view, and each without a virtual timer capability.

The timer is called the application processor **AON_REF_CLK** Generic Timer, and the individual timers are named as follows in the applications memory map, and are in the VSYS.SYSTOP power domain:

- Application processor **AON_REF_CLK** CNTCTL.
- Application processor **AON_REF_CLK** CNTBase0.
- Application processor **AON_REF_CLK** CNTBase1.

One timer is Secure and the other is Non-secure.

Memory map frames

Table 2-14 shows the frames that the application processor and SCP memory map contain, that relate to the counter and timer components.

Table 2-14 Frames for counter and timer components

Frame	Description
AON_REF_CLK CNTControl	Frame that contains the control registers for the **AON_REF_CLK** counter
AON_REF_CLK CNTRead	Frame that contains the status registers for the **AON_REF_CLK** counter
AON_REF_CLK CNTCTL	CNTCTLBase frame for the SCP **AON_REF_CLK** timer
AON_REF_CLK CNTBase0	CNTBase0 frame for the SCP **AON_REF_CLK** timer
CS CNTControl	Frame that contains the control registers for the CoreSight timestamp counter
Application processor **AON_REF_CLK** CNTCTL	CNTCTLBase frame for the application processor **AON_REF_CLK** timers
Application processor **AON_REF_CLK** CNTBase0	CNTBase0 frame for the application processor **AON_REF_CLK** timer, and is only accessible using Secure accesses
Application processor **AON_REF_CLK** CNTBase1	CNTBase1 frame for the application processor **AON_REF_CLK** timer

2.8.3 Processor power modes

A processor can generate timer interrupts after entering into WFI or WFE mode. Processor 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.

2.8.4 Watchdog timers

This section describes:

- *Trusted watchdog*.
- *Generic Watchdog*.

Trusted watchdog

The application processor subsystem includes an ARM SP805 watchdog timer that protects the Secure boot process when it is necessary to run non-trusted device drivers.

The first time the trusted watchdog expires, an interrupt to the CoreLink GIC-400 Generic Interrupt Controller is generated. If Secure boot software fails to clear the watchdog, and it expires for a second time, a global reset is generated.

This watchdog increments at 32.768kHz. This watchdog can also be halted in debug using the cross trigger network, but only when the SYSWD_HOD_EN field in the SSC_SWDHOD register is set to HIGH.

See *Cross triggers* on page 2-37.

Generic Watchdog

The Server Base System Architecture defines and requires a Generic Watchdog for use by EL2 software. This watchdog generates the following interrupts.

1. This interrupt is configured as an EL2 interrupt that is routed as an SPI.

2. This interrupt must cause EL2 and higher levels to be reset. Because the application processors in the ADP both implement EL3, you can reset EL2 by routing this interrupt as an SPI that you can configure as an EL3 interrupt.

Two memory mapped register frames manage the Generic Watchdog. In the ADP, Secure and Non-secure accesses can access the frames.

For information about the programmers model of the Generic Watchdog, see the following:

- *Application processor interrupt map* on page 3-3.
- *Application processor memory map* on page 3-10.
- *ARM® Server Base System Architecture Platform Design Document*.

2.9 Debug and Profiling

This section describes:

- *ADP Debug architecture*.
- *Cortex-A72 and Cortex-A53 processor Debug architecture* on page 2-35.
- *Trace* on page 2-36.
- *Cross triggers* on page 2-37.
- *System Trace Macrocell* on page 2-38.
- *Debugger connectivity* on page 2-40.
- *CoreSight timestamps* on page 2-40.

2.9.1 ADP Debug architecture

Figure 2-9 on page 2-34 shows a high-level representation of the ADP debug architecture. It excludes timestamp distribution. See *CoreSight timestamps* on page 2-40. Most of the debug-related components are located in the CoreSight subsystem. Others are located in the Cortex-A72 and Cortex-A53 clusters, the SCP subsystem, and in the main system.

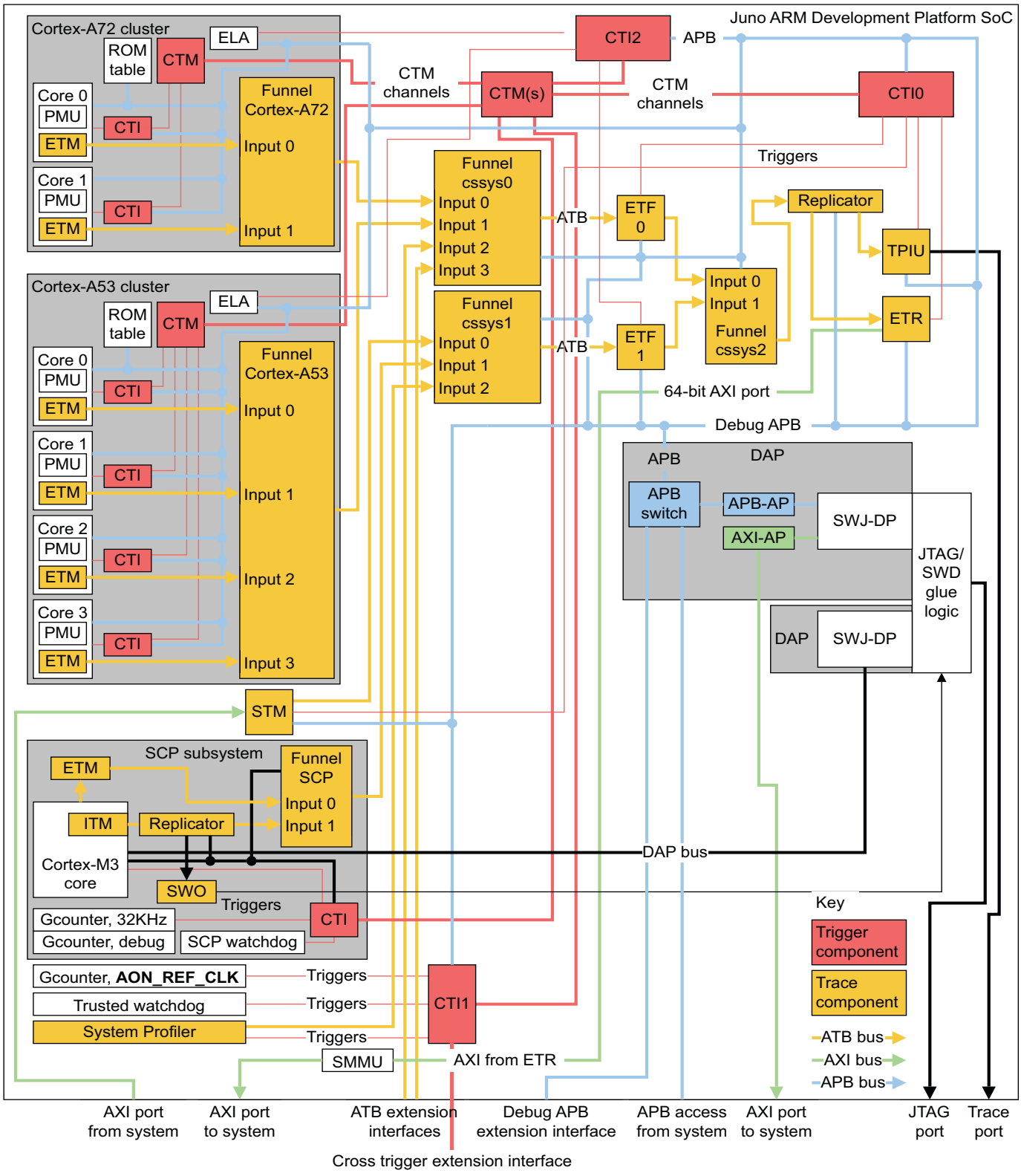


Figure 2-9 ADP debug architecture, excluding timestamp distribution

2.9.2 Cortex-A72 and Cortex-A53 processor Debug architecture

The ADP contains two application processor clusters. Each cluster, internally, contains the following debug components:

- A *Performance Monitor Unit* (PMU) for each core. This module implements Cortex-A72 embedded performance monitoring functionality.
- An *Embedded Trace Macrocell* (ETM) for each processor. This module generates real-time trace information that trace tools can use to reconstruct the execution of all or part of a program. Each ETM generates an ATB trace output that is sent to a funnel before going to the CoreSight subsystem.
- A trace funnel that arbitrates up to four trace sources down to one ATB trace output before it is sent to the CoreSight subsystem.
- A *Cross Trigger Interface* (CTI) for each processor, with a *Cross Trigger Matrix* (CTM) for the cluster. The CTI enables debug subsystems to interact, that is, cross trigger, with each other. The CTI connects to a CTM in the CoreSight subsystem.
- Debug control with an APB register interface to provide accessibility to debug registers in the cluster.
- A local debug ROM table that contains a list of components in the cluster, enabling the debugger to determine the components that are implemented locally in the cluster.
- An *Embedded Logic Analyzer* (ELA) that has a multi-state trigger capability to drive actions such as:
  - Stop clock and scan.
  - Start and stop trace.
  - Cause irritation.
  - Generate a trigger on an external pad.

Each core in the Cortex-A72 and Cortex-A53 clusters also generates and receives the following debug-related signaling:

- Debug communication signal outputs that drive interrupts at the GIC-400. See *Application processor interrupt map* on page 3-3.
- Power control requests from each cluster, and each drives an interrupt input of the SCP, **A72DBGPWRUPREQ** or **A53DBGPWRUPREQ**.

Each cluster also contains a debug authentication interface, with the signals **DBGEN**, **NIDEN**, **SPIDEN**, and **SPNIDEN**.

DBGEN and **NIDEN** are tied HIGH.

For more information about the Cortex-A72 and Cortex-A53 debug architecture, see the following:

- *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual*.
- *ARM® Cortex®-A53 MPCore Processor Technical Reference Manual*.

2.9.3 Trace

The ADP contains the following trace sources:

- An ETM for each Cortex-A72 and Cortex-A53 core.
- An ITM in the SCP subsystem.
- An ETM in the SCP subsystem.
- An STM.
- The System Profiler.
- Two 64-bit ATB extension interfaces.

Each cluster contains an ATB funnel that is associated with it. The funnels are the Cortex-A72 funnel, and Cortex-A53 funnel, that funnels all trace sources from each core down to one ATB interface each. Each funnel contains four ATB slave ports, with slave port 0 connected to core 0, slave port 1 connected to core 1. If any core is not implemented, the associated slave port is tied-off.

The trace output signals combine with two trace expansions using another funnel, Funnel cssys0, before they are fed into the *Embedded Trace FIFO* (ETF), ETF0. The funnel input connections are as follows:

ATB Slave Port 0 Connected to the ATB bus from the Cortex-A72 funnel.

ATB Slave Port 1 Connected to the ATB bus from the Cortex-A53 funnel.

ATB Slave Port 2 Connected using an asynchronous bridge and provided as an ATB extension Interface, ATB Extension 0.

ATB Slave Port 3 Connected using an asynchronous bridge and provided as an ATB extension Interface, ATB Extension 1.

The SCP subsystem also contains an ATB funnel that funnels the two trace sources from the ITM and the ETM down to one ATB interface.

This trace output is then combined with the STM trace source and System Profiler trace source using another funnel, Funnel cssys1, before it is fed into the ETF, ETF1. The funnel inputs are connected as follows:

ATB Slave Port 0 Connected to the STM.

ATB Slave Port 1 Connected to the SCP.

ATB Slave Port 2 Connected to the System Profiler.

The trace outputs of funnels cssys0 and cssys1 each feed into an *Embedded Trace FIFO* (ETF) and then combine using funnel cssys2. The funnel inputs connect as follows:

ATB Slave Port 0 Connected to ETF0.

ATB Slave Port 1 Connected to ETF1.

Both ETFs implement a 64KB buffer that enables buffering of trace data. The output trace data stream of the funnel, Funnel cssys2, is then replicated before it is sent to either the:

- *Trace Port Interface Unit* (TPIU), that sends it out using the trace port.
- ETR that can write the trace data to memory located in the application memory space.

The MMU-401 translates addresses of accesses from the ETR to memory to support the use of *Intermediate Physical Address* (IPA) at the ETR. The ETF can also operate as a trace buffer enabling either:

- The debugger to capture the trace data using the JTAG interface.
- The application processors to access it when performing self-hosted debug.

For more information about these components, see the following:

- *ARM® CoreSight™ Components Technical Reference Manual*.
- *ARM® CoreSight™ Trace Memory Controller Technical Reference Manual*.
- *ARM® CoreSight™ System Trace Macrocell Technical Reference Manual*.

2.9.4 Cross triggers

Cross triggers provide a way for cores and devices to trigger each other in a controlled manner. Triggers to and from a device or processor connect to a CTI module that maps them onto channels that are then connected to one or more CTMs to replicate the channels to all CTIs in the system.

Table 2-15 shows the components that have trigger sources and sinks in the ADP and how they are connected.

Table 2-15 Trigger sources, sinks, and their connectivity

Module name	Source	Sinks	CTI and CTM connectivity
Cortex-A72 cores	Y	Y	Triggers from each core connect to a CTI associated with the cluster, and all CTIs in the cluster then connect to a CTM within the cluster before connecting to a CTM in the CoreSight subsystem.
Cortex-A72 ELA	Y	N	The Cortex-A72 ELA generates the following triggers: • dbgreq. • start_trace. • stop_trace. These triggers connect to CTI2 in the CoreSight subsystem over event bridges.
Cortex-A53 cores	Y	Y	Triggers from each core connect to a CTI associated with the cluster, and all CTIs in the cluster then connect to a CTM within the cluster before connecting to a CTM in the CoreSight subsystem.
Cortex-A53 ELA	Y	N	The Cortex-A53 ELA generates the following triggers: • dbgreq. • start_trace. • stop_trace. These triggers connect to CTI2 in the CoreSight subsystem over event bridges.
Cortex-M3 SCP	Y	Y	Triggers from the Cortex-M3 processor connect to a CTI shared with the 32kHz Generic Counter in the SCP. This CTI then connects to the CTM within the CoreSight subsystem.
32kHz Generic Counter 32k CNTControl	N	Y	This Generic Counter takes two trigger sources from the CTI, shared with the Cortex-M3 SCP, to enable it to halt the counter.
STM	Y	Y	The STM provides trigger outputs **TRIGOUTSPTE**, **TRIGOUTSW**, **TRIGOUTHETE**, and **ASYNCOUT**. It is necessary to have two trigger inputs to drive four inputs on its hardware event interface, with two driving **HWEVENTS[0]** and **HWEVENTS[2]** directly, and the same signals, after inverting, driving **HWEVENTS[1]** and **HWEVENTS[3]**. These triggers connect to CTI0 in the CoreSight subsystem.

Table 2-15 Trigger sources, sinks, and their connectivity (continued)

Module name	Source	Sinks	CTI and CTM connectivity
ETF	Y	Y	The ETF generates two triggers, FULL and ACQCOMP, and takes trigger inputs to drive **FLUSHIN** and **TRIGIN**. These triggers connect to CTI0 in the CoreSight subsystem.
ETR	Y	Y	The ETR generates two triggers, FULL and ACQCOMP, and takes trigger inputs for **FLUSHIN** and **TRIGIN**. These triggers connect to CTI0 in the CoreSight subsystem.
TPIU	N	Y	The TPIU takes trigger inputs for **FLUSHIN** and **TRIGIN**. These triggers connect to CTI0 in the CoreSight subsystem.
AON_REF_CLK Generic Counter **AON_REF_CLK** CNTControl	N	Y	This Generic Counter takes two triggers to enable it to halt the counter. These sources are from CTI1 in the CoreSight subsystem.
Juno System Profiler	Y	Y	The System Profiler takes a single trigger input and provides a single trigger output. These triggers connect to CTI1 in the CoreSight subsystem.
SCP watchdog	N	Y	The SCP watchdog takes two trigger sources from the CTI, shared with the Cortex-M3 SCP, to enable it to halt the watchdog timer.
Trusted watchdog	N	Y	The trusted watchdog takes two trigger sources to enable it to halt the watchdog timer. These sources are from CTI1 in the CoreSight subsystem.

For more information about CTI and CTM, see the *ARM® CoreSight™ Components Technical Reference Manual*.

2.9.5 System Trace Macrocell

The *System Trace Macrocell* (STM) is a trace source that provides high-bandwidth trace of instrumentation that is embedded into software. This instrumentation is made up of memory-mapped writes to the STM using its AXI slave interface. In addition, the STM provides a hardware event interface, generating trace data on the rising edge of the signals on the hardware event interface. It also implements an APB interface that connects to the APB bus from the CoreSight subsystem.

The STM AXI interface occupies a 16MB address space in the application memory map. However, for each processor core in the ADP, and also each unique master interface, or group of master interfaces, in the system, the STM presents a separate view of the 16MB AXI space to each as if each core or master has its own private STM, therefore providing up to a full 65 536 extended stimulus port for each core or master interface.

Table 2-16 on page 2-39 shows the STM views and the master or processor cores with which each is associated.

Note

The STM provides at total of 64 views, and some are Reserved. Table 2-16 on page 2-39 also shows how these views are mapped to STPv2 master ID on the ATB trace port.

When accessing the STM using the AXI slave expansion interface, each external master is identified using the same **ARUSER[9:6]** and **AWUSER[8:5]** bits that the TZC-400 NSAID inputs also use. For each access to the STM from the AXI slave expansion interface, the value

on **ARUSER[9:6]** and **AWUSER[8:5]** identifies the view of the expansion master view that is required, with the value of x+1 selecting the expansion master x view, where x is in the range 0-7.

If the value of the **ARUSER[9:6]** and **AWUSER[8:5]** bits is not in the range 1-8, and the access does not belong to any logical system master that Table 2-16 shows, then the default master view, 62, is selected.

Table 2-16 STM views and associated STPv2 Master ID

View	Logical system master	STPv2 master ID for Secure accesses	STPv2 master ID for Non-secure accesses
[0]	Cortex-A72 core 0	0	64
[1]	Cortex-A72 core 1	1	65
[2:3]	Reserved	-	-
[4]	Cortex-A53 core 0	4	68
[5]	Cortex-A53 core 1	5	69
[6]	Cortex-A53 core 2	6	70
[7]	Cortex-A53 core 3	7	71
[8:31]	Reserved	-	-
[32]	Mali-T624 GPU	32	96
[33]	Cortex-M3 SCP	33	97
[34]	Expansion master 0	34	98
[35]	Expansion master 1	35	99
[36]	Expansion master 2	36	100
[37]	Expansion master 3	37	101
[38]	Expansion master 4	38	102
[39]	Expansion master 5	39	103
[40]	Expansion master 6	40	104
[41]	Expansion master 7	41	105
[42:61]	Reserved	-	-
[62]	Default master	62	126
[63]	DAP-AXI-AP	63	127

Table 2-17 shows the STM hardware events inputs.

Table 2-17 STM hardware events inputs

STM event input	Event	Description
HWEVENTS[0]	CTI0TRIGOUT[4]	Trigger output **CTI0TRIGOUT[4]** from CTI0, rising edge detection
HWEVENTS[1]	CTI0TRIGOUTN[4]	Trigger output **CTI0TRIGOUT[4]** from CTI0, falling edge detection
HWEVENTS[2]	CTI0TRIGOUT[5]	Trigger output **CTI0TRIGOUT[5]** from CTI0, rising edge detection
HWEVENTS[3]	CTI0TRIGOUTN[5]	Trigger output **CTI0TRIGOUT[5]** from CTI0, falling edge detection
HWEVENTS[63:4]	-	Reserved

For more information about the STM, see the following:

- *ARM® CoreSight™ System Trace Macrocell Technical Reference Manual*.
- *ARM® CoreSight STM-500 System Trace Macrocell Technical Reference Manual*.

2.9.6 Debugger connectivity

You debug the entire ADP using a common debug interface.

Application CoreSight Debug subsystem

This debug subsystem includes the Cortex-A72 and Cortex-A53 clusters, and associated debug components that are accessible using the application memory map. External debug access to this subsystem uses a DAP that consists of the following:

- An SWJ - DP that is a combined JTAG-DP and SWDP, enabling debug access either using JTAG or SWD.
- An APB-AP, that is an APB Memory Access Port. In the ADP, this module enables the debugger to directly connect to the APB bus within the CoreSight subsystem that connects using an APB switch to all debug related components mapped to the CoreSight debug and trace APB region within the peripherals region of the application memory map.
- An APB switch, that enables APB accesses from the system and from the APB-AP to either access CoreSight subsystem components.
- An AXI-AP, that is an AXI Memory Access Port.

  In the ADP, this module enables the debugger to access the application memory map using an AXI interface.

2.9.7 CoreSight timestamps

All debug components in the ADP operate in a separate time domain to the time domains of the application processors and the system control processor that *Timers* on page 2-29 describes. A separate Generic Counter, CS CNTControl, is implemented that provides a timestamp only for use as the CoreSight timestamp. This Generic Counter operates on the main reference clock that is distributed to all CoreSight components.

The CoreSight system requires a higher resolution, higher clock frequency 64-bit time value that timestamps trace data from trace sources. The 56 least significant bits of the **AON_REF_CLK** time count value from CS CNTControl are the 56 most significant bits of the timestamp value. A CoreSight Interpolator component generates the least significant 8 bits of the timestamp, using a local high frequency clock and interpolation.

The following equation describes how to generate the high-resolution *Timestamp* from the original low-resolution *Timestamp Counter*:

Timestamp = (*Timestamp Counter* << 8) + interpolated bits

The timestamp value that the STM reports is natural binary encoded and 64 bits wide.

2.10 Boot

This section describes:

- *About boot*.
- *Initial power, clock, and Reset requirements for boot*.

2.10.1 About boot

The ADP supports the *Trusted Board Boot on ARM Reference Hardware Platform Design Document*. To support boot, the ADP provides:

- A Cortex-M3-based SCP designed to function as a trusted subsystem. This subsystem also includes:
  - A local trusted on-chip ROM.
  - Support for a local trusted on-chip SRAM to execute the main SCP firmware.
- A Cortex-A Series-based system with the following features:
  - Able to support trusted on-chip boot ROM that stores associated cryptography functionality.
  - Support for a non-trusted on-chip ROM that includes drivers for SoC interfaces from the manufacturer.
  - Able to support trusted on-chip SRAM.

For more information, see the following:

- *ARM® Trusted Board Boot Platform Design Document*.
- *ARM® Trusted Base System Architecture Platform Design Document*.

2.10.2 Initial power, clock, and Reset requirements for boot

For the ADP to successfully boot-up, the SCP must be supplied with power and clocks, and its reset must be deasserted, without any software support. The ADP must meet the following conditions:

- The ADP Board power-management IC supplies VAON, VSYS, VA72, VA53, and VGPU.
- The real-time clock, **S32K_CLK**, starts automatically.
- Powerup reset, **nPORESET**, is deasserted when **S32K_CLK** is running.
- **AON_REF_CLK** always exists.

2.11 HDLCD controller

The HDLCD controller, supporting *High Definition* (HD) resolutions, has the following features:

- Resolution:

  VGA 640 × 480 at 60fps and pixel clock frequency of 23.75MHz.

  UXGA 1600 × 1200 at 60fps and pixel clock frequency of 63.5MHz.

  Full HD 1920 × 1080 at 60fps and pixel clock frequency of 148.5MHz.

  QXGA, 40% blanking
  2048 × 1536 at 30fps and pixel clock frequency of 132MHz.

  WQXGA, 40% blanking
  2560 × 1600 at 30fps and pixel clock frequency of 172MHz.

  WQXGA, reduced blanking
  2560 × 1600 at 60fps and pixel clock frequency of 210MHz.

- Frame buffer:
  - Supports all common non-indexed RGB formats.
  - Frame buffer can be placed anywhere in memory.
  - Scan lines must be a multiple of 8 bytes long, and aligned to 8-byte boundaries. There are no other restrictions on size or placement. Line pitch configurable in multiples of 8 bytes.
- Management:
  - Frame buffer address can be updated at any time, and applies from the next full frame.
  - Frame buffer size, color depth, and timing can only be changed while the display is disabled.
- Maskable interrupts:
  - DMA-end, last part of frame that is read from bus.
  - VSYNC.
  - Underrun.
  - Bus error.
- Color depths:
  - Supports 8 bits per color. Frame buffers with other color depths are truncated or interpolated to 8 bits per component.
- Interfaces:
  - AMBA 3 APB interface for configuration.
  - Read-only AXI bus for frame buffer reads.
  - Standard LCD external interface. All timings and polarities are configurable.
  - APB, AXI, and pixel clock can run on separate asynchronous clocks.
- Buffering:
  - Internal 1KB buffer.
  - After underrun, it blanks the rest of the frame and resynchronizes from the next frame.

2.12 PCIe

The ADP includes a 4-lane PCIe Root Port capable of operating at up to 5GTps per lane. The Root Port supports high-bandwidth connectivity with external peripherals such as SATA disk controllers and Gigabit Ethernet NIC. The PCIe Root Port and PHY are integrated on the chip.

This section contains the following subsections:

- *Configuration space*.
- *Interrupts*.
- *MSI*.
- *Clock requirements*.
- *Limitations*.

2.12.1 Configuration space

The Root Port supports the *Enhanced Configuration Access Mechanism* (ECAM) for access to the PCIe configuration spaces of all devices in the PCIe subsystem. See the *PCI Express Base Specification Revision 3.0* for more information on ECAM.

2.12.2 Interrupts

The ADP includes a *Message Signaled Interrupt* (MSI) unit that complies with the GICv2m architecture. The *Server Base System Architecture* (SBSA) defines the GICv2m architecture. The MSI unit converts memory writes, that target GICv2m registers, into edge-triggered interrupt signals that are connected to the GIC-400 Generic Interrupt Controller. *Application processor interrupt map* on page 3-3 shows the SPIs allocated to the MSI unit. The GICv2m architecture enables an IMPLEMENTATION DEFINED number of Non-secure MSI frames to be implemented. The MSI unit implements four of these frames. *Application processor memory map* on page 3-10 shows these frames.

The PCIe macro provides interrupts that Table 3-3 on page 3-4 shows. All the interrupts are of type *level* except for the System Error Interrupt, that is of type *edge*.

2.12.3 MSI

The ADP can convert memory writes to an interrupt using an MSI component that implements the GICv2m architecture.

See the following:

- *ARM® Server Base System Architecture Platform Design Document*.
- *ARM® Generic Interrupt Controller Architecture Specification, Architecture version 2.0*.

In the ADP, the MSI component supports four Non-secure frames. Each frame occupies a 64KB region and has 32 SPIs.

2.12.4 Clock requirements

The **tl_clk** signal must be 133MHz for correct operation of the PCIe macro.

2.12.5 Limitations

The PCIe macro does not support the following features:

- Beacons.
- Wake-up on LAN support.

- Retention of PCIe configuration, sticky registers in the *PCI Express Base Specification, Revision 3.0*, when the SoC is powered down.
- Relaxed Ordering.
- ID based ordering.
- Virtual Channels.
- Address Translation Services.
- Slot power limit.
- Peer-to-peer.

2.13 Thin Links

This section contains the following subsections:

- *About Thin Links*.
- *TLX master and slave bridges*.
- *System latency* on page 2-49.

2.13.1 About Thin Links

The ADP supports the addition of peripherals such as GPUs and codecs to the external FPGA LogicTile for use cases such as prototyping and development of device drivers. Adding peripherals requires the ADP to provide both AXI slave and master interfaces to connect to external peripherals. However, exporting the full AXI interface from the chip is expensive in terms of pin count and die area.

Therefore, the ADP uses a simple low pin-count solution using the TLX-400 Thin Links. The chip-to-chip interface does not use any high-speed I/O, SerDes, or PHY, because these components are not available in the 28HPM process.

The CoreLink TLX-400 Network Interconnect Thin Links is an extension to the CoreLink NIC-400 Network Interconnect base product and provides a mechanism to reduce the number of signals in an AXI point-to-point connection and enable it to be routed over a longer distance.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Implementation Guide*.

2.13.2 TLX master and slave bridges

This section describes:

- *AXI interfaces*.
- *TLX-400 bandwidth* on page 2-48.

AXI interfaces

Two Thin Link bridges are required for off-chip communication using the AXI protocol.

This section describes:

- *AXI interface requirements*.
- *AXI slave interface* on page 2-47.
- *AXI master interface* on page 2-48.

AXI interface requirements

The chip-to-chip interface in the ADP consists of a Thin Links based AXI master and AXI slave interface to integrate off-chip logic. External masters on the FPGA, such as the GPU and video codecs, use the AXI slave interface to access the ADP memory. The cores in the ADP use the AXI master interface to configure the peripherals on the FPGA.

Figure 2-10 on page 2-47 shows the AXI interfaces between the ADP and the FPGA.

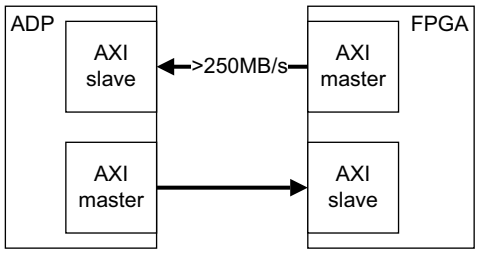


Figure 2-10 TLX-400 AXI interfaces

AXI slave interface

Location The AXI master port in the FPGA connects to the AXI slave port in the ADP.

Configurations This AXI4 interface can connect to an AXI master interface, or multiple AXI masters on the FPGA. The interface supports the following.

- 128-bit data.
- 40-bit address.
- 6 ID bits.
- The **ARUSERTLXS[5:0]** and **AWUSERTLXS[4:0]** signals support coherency signals on the ACE-Lite interface, excluding **ARBAR** and **AWBAR**:

  ARSNOOP[3:0] Indicates that the transaction type of a shareable access is mapped to **ARUSERTLXS[5:2]**.

  ARDOMAIN[1:0] Indicates that the shareability domain of the transaction is mapped to **ARUSERTLXS[1:0]**.

  AWSNOOP[2:0] Indicates that the transaction type of a shareable access is mapped to **AWUSERTLXS[4:2]**.

  AWDOMAIN[1:0] Indicates that the shareability domain of the transaction is mapped to **AWUSERTLXS[1:0]**.

Bandwidth Requirement

Table 2-18 shows the bandwidth that is required on the AXI slave interface.

Table 2-18 Bandwidth requirements of external masters

External master	Required bandwidth on Thin Links based AXI slave port
Mali-T624 GPU, assuming 50MHz in V7	250MBps
Mali-450 8PP, assuming 50MHz in V7	>250MBps
Video, assuming 75MHz	100MBps

AXI master interface

Location The AXI Master Port in the ADP is connected to the AXI Slave Port in the FPGA.

Configurations This AXI4 interface can connect to an AXI slave interface, or multiple AXI interfaces, on the FPGA. The interface supports the following.

- 64-bit data.
- 40-bit address.
- 14 ID bits.
- **ARUSERTLXM[3:0]** and **AWUSERTLXM[3:0]** signals identify masters. That is, the ID of the master that generated the transaction to the AXI master interface. The master could be either in the compute subsystem or outside at the SoC level.

Bandwidth Requirement

The AXI master interface bandwidth is not critical because the traffic is mostly APB configuration writes. However, it is necessary to occasionally clear the frame buffer in the FPGA tile.

TLX-400 bandwidth

This section describes:

- *Thin Links based AXI slave interface*.
- *Thin Links based AXI master interface*.

Thin Links based AXI slave interface

To support the bandwidth requirements on the AXI slave interface, the ADP uses the following link widths:

- 56 in the forward direction, AW, AR, W.
- 48 in the reverse direction, R, B.

The 128-bit AXI bus internal to the ADP runs at up to 533MHz that provides a bandwidth capacity of 8.5GBps. The Thin Links configuration in the ADP supports the following, assuming the same 533MHz clock:

- 25% of the maximum AXI bandwidth, 2.13GBps, in the forward direction.
- 31% of the maximum AXI bandwidth, 2.64GBps, in the reverse direction.

However, the ADP-FPGA interface only supports up to 61.5MHz clock. Therefore the bandwidth that is supported on the ADP slave interface is as follows:

- Forward direction, 2.13GBps x (61.5/533) = 246MBps.
- Reverse direction, 2.64GBps × (61.5/533) = 305MBps.

The Thin Links based slave interface can accept 16 outstanding reads and 16 outstanding writes from an external master.

Thin Links based AXI master interface

The AXI master interface has a data width of 64 bits. This width means that the AXI width is different to the AXI slave interface. The 64-bit AXI bus internal to the ADP runs at up to 400MHz and supports a maximum bandwidth of 4GBps.

The AXI master interface bandwidth is not critical at the slave interface. Therefore, a link width of 16, single direction, is selected for the ADP. This scheme supports bandwidths of:

- 68MBps, forward.

- 78MBps, reverse.

2.13.3 System latency

Table 2-19 shows the system latency.

Table 2-19 System latency

Block	Clock frequency	Period	Latency				
			AR	R	AW	W	B
TLX-400 slave interface	61.5MHz	16.26ns	8 cycles	8	8	11	6
TLX-400 master interface	61.5MHz	16.26ns	12 cycles	11	12	17	7

Chapter 3
Programmers Model

This chapter describes the *Juno r2 ARM Development Platform* (ADP) registers, and provides information on how to program the ADP.

It contains the following sections:

- *About this programmers model* on page 3-2.
- *Application processor interrupt map* on page 3-3.
- *Application processor memory map* on page 3-10.
- *ADP System Security Control Registers* on page 3-27.
- *System override registers* on page 3-40.
- *MHU Registers* on page 3-42.
- *SoC Interconnect NIC-400 Registers* on page 3-50.
- *Compute subsystem NIC-400 Registers* on page 3-63.
- *HDLCD Registers* on page 3-68.
- *PCIe Control Registers* on page 3-85.
- *PCIe Root Port configuration registers* on page 3-96.
- *MSI Registers* on page 3-132.
- *Trusted Entropy Source Registers* on page 3-139.

3.1 About this programmers model

The following information applies to all registers:

- Do not attempt to access Reserved or unused address locations. Attempting to access these locations can result in UNPREDICTABLE behavior.
- Unless otherwise stated in the accompanying text:
  - Do not modify UNDEFINED register bits.
  - Ignore UNDEFINED register bits on reads.
  - All register bits are reset to a logic 0 by a system or powerup reset.
- The following describes the access type:

  RW Read and write.

  RW1C Read. Write 1 to clear a bit.

  RO Read-only.

  WO Write-only.

3.2 Application processor interrupt map

This section describes:

- *CPUID definition*.
- *Application processors interrupt map*.
- *Board interrupts* on page 3-8.
- *Message Signaled Interrupt (MSI) unit* on page 3-9.
- *System error interrupts* on page 3-9.

3.2.1 CPUID definition

The *ARM Development Platform* (ADP) supports a configurable number of cores across two clusters.

Interrupts from processor cores connect to a GIC-400 generic interrupt controller that both clusters share. The *ARM Generic Interrupt Controller Architecture* associates each core with a core ID value.

Table 3-1 shows the core ID values that are assigned for each core.

Table 3-1 GIC core IDs for ADP configurations

Cortex-A72 cluster		Cortex-A53 cluster			
Core 0	**Core 1**	**Core 0**	**Core 1**	**Core 2**	**Core 3**
0	1	2	3	4	5

3.2.2 Application processors interrupt map

The *Generic Interrupt Controller Architecture* defines the following types of interrupt:

- *Private Peripheral Interrupts* (PPIs) separately exist for each processor.
- *Shared Peripheral Interrupts* (SPIs) are shared for all processors.

Table 3-2 shows the PPI map for the application processors in the ADP.

Table 3-2 Private peripheral interrupts

Interrupt ID	Source	Output	Description
25	Cortex-A72 or Cortex-A53 cluster	-	Virtual maintenance interrupt, PPI6.
26		**nCNTHPIRQ**	Non-secure PL2 timer event, PPI5.
27		**nCNTVIRQ**	Virtual timer event, PPI4.
28	Not implemented	-	Legacy **nFIQ** signal, PPI0.
29	Cortex-A72 or Cortex-A53 cluster	**nCNTPSIRQ**	Secure PL1 physical timer event, PPI1.
30		**nCNTPNSIRQ**	Non-secure PL1 physical timer event, PPI2.
31	Not implemented	-	Legacy **nIRQ** signal, PPI3.

Table 3-3 shows the SPI map for the application processor cores in the ADP. In the ADP, for configurations that include fewer than eight cores, interrupts for unimplemented cores are Reserved.

Table 3-3 Application processor cluster shared peripheral interrupt map

Interrupt ID	Source	Description
32	Cortex-A72 cluster	Cortex-A72 core 0 comms channel transmit.
33		Cortex-A72 core 0 comms channel receive.
34		Cortex-A72 core 0 performance monitor.
35		Cortex-A72 core 0 cross trigger.
36		Cortex-A72 core 1 comms channel transmit.
37		Cortex-A72 core 1 comms channel receive.
38		Cortex-A72 core 1 performance monitor.
39		Cortex-A72 core 1 cross trigger.
40		Reserved.
41		
42		
43		
44		
45		
46		
47		

Table 3-3 Application processor cluster shared peripheral interrupt map (continued)

Interrupt ID	Source	Description
48	Cortex-A53 cluster	Cortex-A53 core 0 comms channel transmit.
49		Cortex-A53 core 0 comms channel receive.
50		Cortex-A53 core 0 performance monitor.
51		Cortex-A53 core 0 cross trigger.
52		Cortex-A53 core 1 comms channel transmit.
53		Cortex-A53 core 1 comms channel receive.
54		Cortex-A53 core 1 performance monitor.
55		Cortex-A53 core 1 cross trigger.
56		Cortex-A53 core 2 comms channel transmit.
57		Cortex-A53 core 2 comms channel receive.
58		Cortex-A53 core 2 performance monitor.
59		Cortex-A53 core 2 cross trigger.
60		Cortex-A53 core 3 comms channel transmit.
61		Cortex-A53 core 3 comms channel receive.
62		Cortex-A53 core 3 performance monitor.
63		Cortex-A53 core 3 cross trigger.
64	Mali-T624	GPU interrupt request.
65		Job interrupt request.
66		MMU interrupt request.
67	MHU	MHU high-priority interrupt.
68		MHU low-priority interrupt.
69		MHU Secure interrupt.
70	GPU SMMU	Combined Non-secure interrupt.
71		Combined Secure interrupt.
72	PCI Express SMMU	Combined Non-secure interrupt.
73		Combined Secure interrupt.
74	ETR SMMU	Combined Non-secure interrupt.
75		Combined Secure interrupt.
76	CCI	Imprecise error response interrupt.
77		PMU overflow interrupt.
78	-	Reserved.
79		
80	TZC-400	TrustZone address space controller interrupt.

Table 3-3 Application processor cluster shared peripheral interrupt map (continued)

Interrupt ID	Source	Description
81	Juno System Profiler	System Profiler interrupt.
82	-	Reserved.
83	STM	STM synchronization interrupt.
84	CTI	CTI trigger output 0 from CSSYS.
85		CTI trigger output 1 from CSSYS.
86	Trusted watchdog	Trusted watchdog interrupt.
87	Cortex-A53 cluster	**nEXTERRIRQ** AXI write error condition.
88		**nINTERRIRQ** L2 RAM double-bit ECC error.
89	Cortex-A72 cluster	**nEXTERRIRQ** AXI write error condition.
90		**nINTERRIRQ** L2 RAM double-bit ECC error.
91	Application processor **AON_REF_CLK** Generic Timer, Secure	Application processor **AON_REF_CLK** Generic Timer Interrupt, Secure.
92	Application processor **AON_REF_CLK** Generic Timer, Non-secure	Application processor **AON_REF_CLK** Generic Timer Interrupt, Non-secure.
93	EL2 Generic Watchdog	Watchdog EL2 Interrupt.
94		Watchdog EL3 Interrupt.
95-99	-	Reserved.
100	Board	RTC.
101-114	-	Reserved.
115	On-chip UART 0	UART interrupt.
116	On-chip UART 1	UART interrupt.
117	HDLCD Controller 0	HDLCD Controller interrupt.
118	SMC PL354	Interface 0 interrupt.
119		Interface 1 interrupt.
120	DMA controller DMA-330	Interrupt 0.
121		Interrupt 1.
122		Interrupt 2.
123		Interrupt 3.
124		Abort interrupt.
125	HDLCD Controller 1	HDLCD Controller interrupt.
126	DMA controller SMMU	Combined Secure interrupt.
127		Combined Non-secure interrupt.

Table 3-3 Application processor cluster shared peripheral interrupt map (continued)

Interrupt ID	Source	Description
128	HDLCD controller 0 SMMU	Combined Secure interrupt.
129		Combined Non-secure interrupt.
130	HDLCD controller 1 SMMU	Combined Secure interrupt.
131		Combined Non-secure interrupt.
132	USB controller SMMU	Combined Secure interrupt.
133		Combined Non-secure interrupt.
134	Thin Links TLX-400 SMMU	Combined Secure interrupt.
135		Combined Non-secure interrupt.
136	ADP	I^2C interrupt.
137	ADP	Secure I^2C interrupt.
138	ADP	I^2S interrupt.
139	ADP	TRNG.
140	ADP	DMA Controller [4].
141	ADP	DMA Controller [5].
142	ADP	DMA Controller [6].
143	ADP	DMA Controller [7].
144	ADP	DDR3 PHY0.
145	ADP	DDR3 PHY1.
146	-	Reserved.
147		
148	ADP	USB *Open Host Controller Interface* (OHCI) Controller.
149	ADP	USB *Enhanced Host Controller Interface* (EHCI) Controller.
150-159	ADP	Reserved.
160	PCIe Root Port	AXI address translation post error interrupt.
161	PCIe Root Port	AXI address translation fetch error interrupt.
162	PCIe Root Port	AXI address translation discard error interrupt.
163	PCIe Root Port	AXI address translation doorbell interrupt.
164	PCIe Root Port	PCIe address translation post error interrupt.
165	PCIe Root Port	PCIe address translation fetch error interrupt.
166	PCIe Root Port	PCIe address translation discard error interrupt.
167	PCIe Root Port	PCIe address translation doorbell interrupt.
168	PCIe Root Port	PCI interrupt line A.

Table 3-3 Application processor cluster shared peripheral interrupt map (continued)

Interrupt ID	Source	Description
169	PCIe Root Port	PCI interrupt line B.
170	PCIe Root Port	PCI interrupt line C.
171	PCIe Root Port	PCI interrupt line D.
172	PCIe Root Port	*Message Signaled Interrupt* (MSI) received.
173	PCIe Root Port	*Advanced Error Reporting* (AER) event.
174	PCIe Root Port	PM, hotplug event.
175	PCIe Root Port	System error.
176-191	ADP	Reserved.
192-201	Board	Board Peripherals.
202-223	ADP	Reserved.
224-351	ADP	GICv2m PCI Express MSI.

See the following:

- *ARM® Generic Interrupt Controller Architecture Specification v2.0*.
- *ARM® CoreLink™ GIC-400 Generic Interrupt Controller Technical Reference Manual*.

3.2.3 Board interrupts

Table 3-4 shows the board interrupts.

Table 3-4 Board interrupts

Interrupt ID	ADP input pin name	Source
100	**EXT_IRQ[0]**	*Real Time Clock* (RTC), PL031.
101	**EXT_IRQ[1]**	Reserved.
102	**EXT_IRQ[2]**	
103-191	-	Juno ARM Development Platform SoC internal peripherals and systems.
192	**EXT_IRQ[3]**	SMC 10/100 Ethernet.
193	**EXT_IRQ[4]**	Reserved.
194	**EXT_IRQ[5]**	PL180 user microSD card.
195	**EXT_IRQ[6]**	GPIO (0) and GPIO (1) used for extra user key entry.
196	**EXT_IRQ[7]**	*Watch Dog Timer* (WDT), SP805.
197	**EXT_IRQ[8]**	*Keyboard and Mouse Interface* (KMI), PL050.
198	**EXT_IRQ[9]**	SP804 dual-timer.
199	**EXT_IRQ[10]**	System registers.
200	**EXT_IRQ[11]**	Interrupt from FPGA on LogicTile daughterboard.
201	**PB_IRQ**	Interrupt that is generated from board push button.

All board interrupts are level-sensitive. The IOFPGA conditions any edge-triggered interrupt before routing it to the ADP, and provides appropriate registers for software to clear the interrupt.

For more information, see the *ARM® Versatile™ Express Juno r2 Development Platform (V2M-Juno r2) Technical Reference Manual*.

3.2.4 *Message Signaled Interrupt* (MSI) unit

See *Interrupts* on page 2-44.

3.2.5 System error interrupts

ARMv8-A supports an SError exception, an interrupt mechanism for system error events. The GICv2 does not support these exceptions, so the **nREI** and **nSEI** inputs to the Cortex-A72 and Cortex-A53 clusters are unused and tied HIGH.

3.3 Application processor memory map

The application memory map is the memory map of the application processor subsystem. It contains mappings for most components in the system, such as the application processors, GPU, and peripherals. It also includes mapping for the AXI expansion interfaces around the ADP.

The following can view the application memory map:

- Application processors.
- Mali-T624.
- Embedded trace router.
- Main *Debug Access Port* (DAP).
- USB.
- DMA.
- PCIe Root Port.

Figure 3-1 on page 3-11 shows a top-level representation of the application memory map that complies with the *Principles of ARM Memory Maps*. It shows 1024GB of address space, accessible using 40-bit addressing, and is divided up into the following types of subregions:

DRAM memory areas

These areas are mapped to DRAM through the DMC-400 Dynamic Memory Controller. For more information about how the DMC-400 maps these areas to DRAM, see *DRAM* on page 3-20.

Expansion area These areas are mapped to the asynchronous AXI master expansion interface to enable partners to interface to the subsystem.

ADP private peripherals and memory area

This region is divided into the following regions:

- A 128MB boot region. See *Boot region* on page 3-12.
- A 256MB peripherals region. See *Peripherals region* on page 3-14.
- Reserved areas. Accesses to these areas results in a decode error response.

The memory map that this section describes also shows the overall security attributes associated with each area of memory. The security attributes 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 results in 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 called *securable*

Components or regions that are defined to be independently software-configurable. Trusted software can change them between the ‘always Secure access’ and ‘Secure and Non-secure access’ access. You can configure these attributes in the NIC-400, or in the component itself, and the default state is ‘Secure access only’ from reset.

User-defined

These areas are mapped to expansion interfaces and components outside the ADP and define 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. Accesses that fail any external security checks must result in a DECERR response.

Figure 3-1 shows the ADP top-level application memory map, but it does not show areas with multiple security attributes.

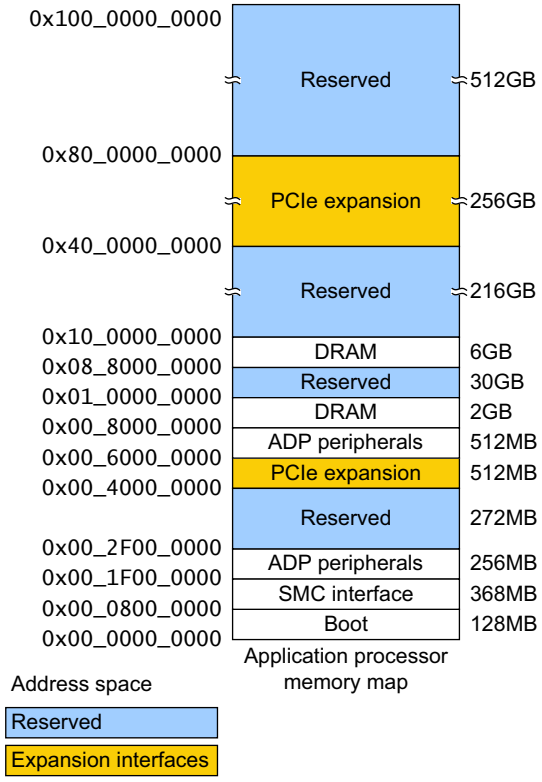


Figure 3-1 ADP top-level application memory map

In general, unless explicitly stated otherwise:

- Where a region maps a peripheral or device, if the peripheral or device occupies less than the region size used, for example, if a peripheral only occupies 4KB from the available 64KB of the region Reserved for it, access to the unmapped region results in a DECERR response.
- Accesses to Reserved areas within the memory map result in a DECERR response.
- When accessing areas that peripherals or devices occupy, these peripherals or devices determine the response to return. These areas can include unmapped or Reserved areas within the areas that the peripheral or device occupies.

The following sections describe the boot and ADP peripheral areas, and how the DMC area maps to DRAM memory using the memory controller. This section describes:

- *Boot region* on page 3-12.
- *Peripherals region* on page 3-14.
- *DRAM* on page 3-20.
- *Application memory map summary* on page 3-21.

3.3.1 Boot region

Figure 3-2 shows that the first 128MB of the address map is defined as a *Secure boot region* and only Secure accesses can access it.

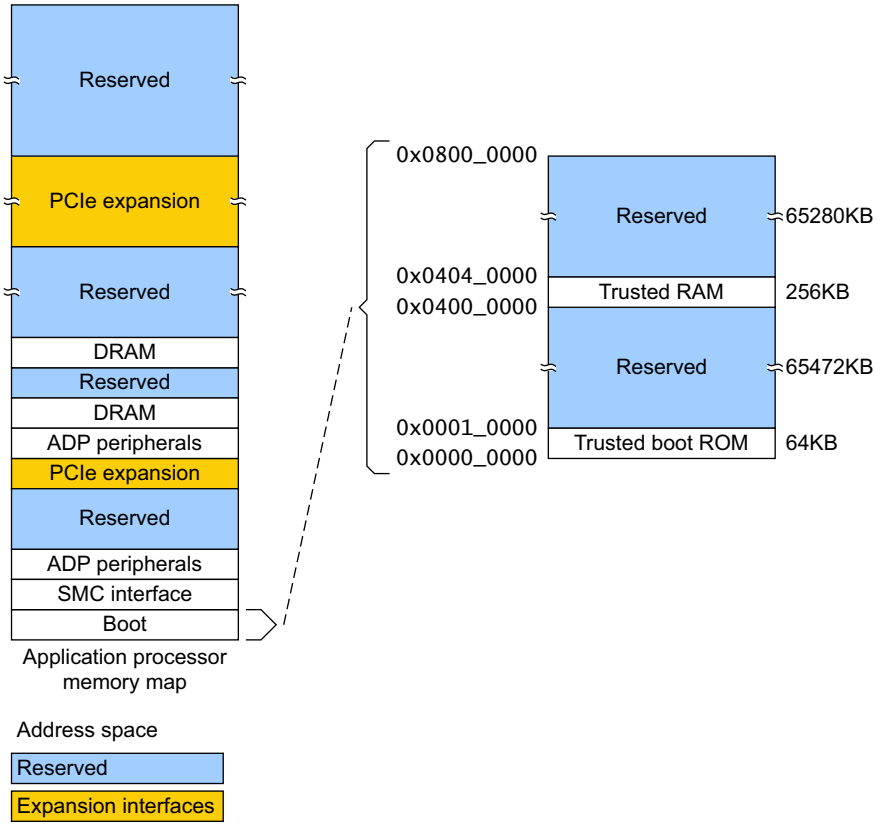


Figure 3-2 ADP boot area memory map

This region is split into ROM and RAM areas. It contains:

- 64KB of trusted boot ROM.
- 256KB of trusted RAM.

3.3.2 SMC interface region

Figure 3-3 shows that a 368MB region starting from address `0x00_0800_0000` is defined as the ADP SMC interface area.

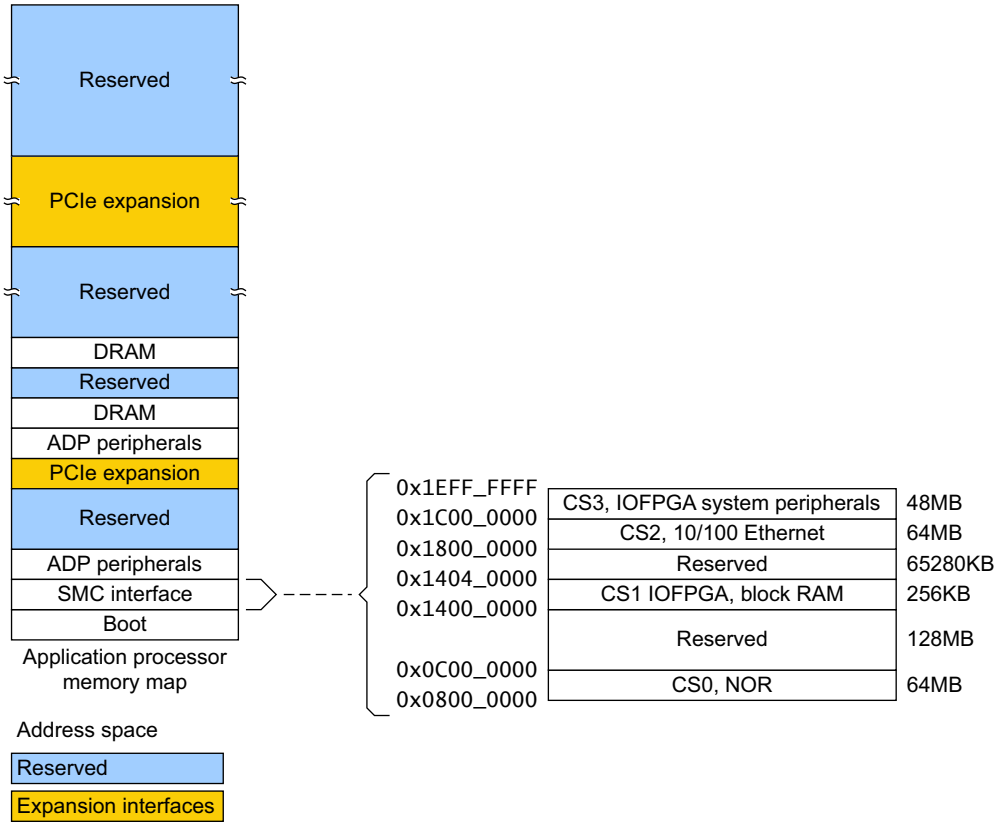


Figure 3-3 ADP SMC interface area memory map

3.3.3 Peripherals region

This region defines the memory map for peripherals and memories that are part of the ADP.

This section contains the following subsections:

- *ADP Peripherals region, 0x1F00_0000-0x2F00_0000*.
- *ADP Peripherals region, 0x6000_0000-0x8000_0000* on page 3-18.

ADP Peripherals region, `0x1F00_0000-0x2F00_0000`

Figure 3-4 shows that a 256MB region starting from address `0x00_1F00_0000` is defined as the ADP peripherals and memory area.

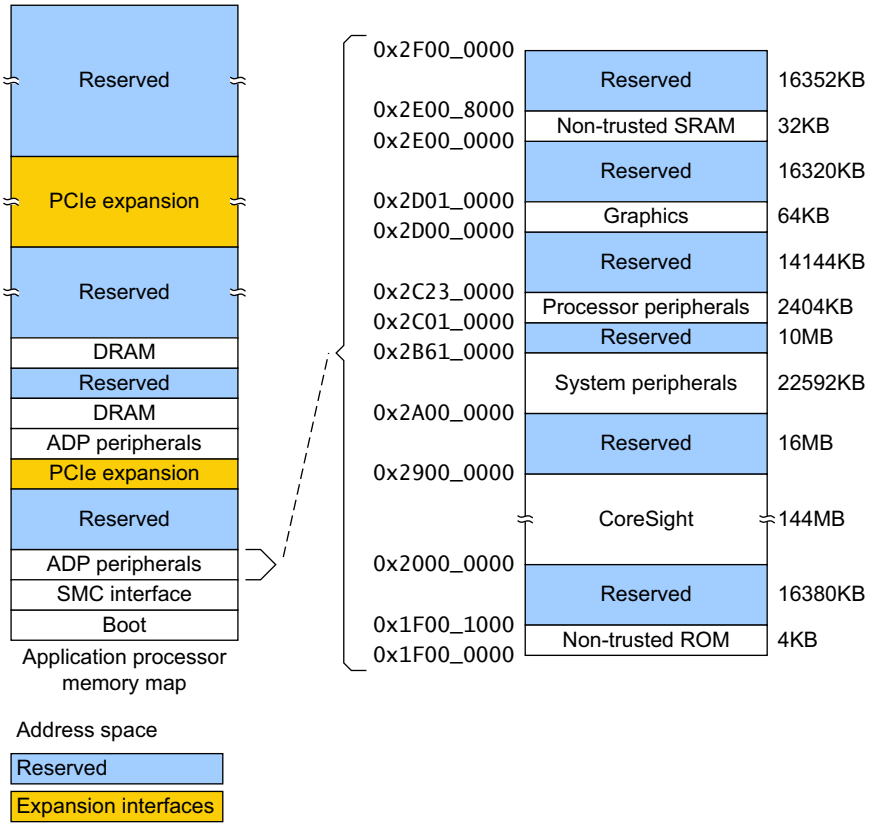


Figure 3-4 ADP peripherals region memory map, `0x1F00_0000-0x2F00_0000`

This section contains the following subsections:

- *Non-trusted ROM*.
- *CoreSight debug and trace* on page 3-15.
- *System peripherals regions* on page 3-16.
- *Processor peripherals region* on page 3-16.
- *Graphics peripherals region* on page 3-17.
- *Non-trusted SRAM* on page 3-18.

Non-trusted ROM

This non-trusted ROM area contains non-trusted boot code. See the *Trusted Base System Architecture Platform Design Document*.

CoreSight debug and trace

Figure 3-5 shows the CoreSight debug and trace region.

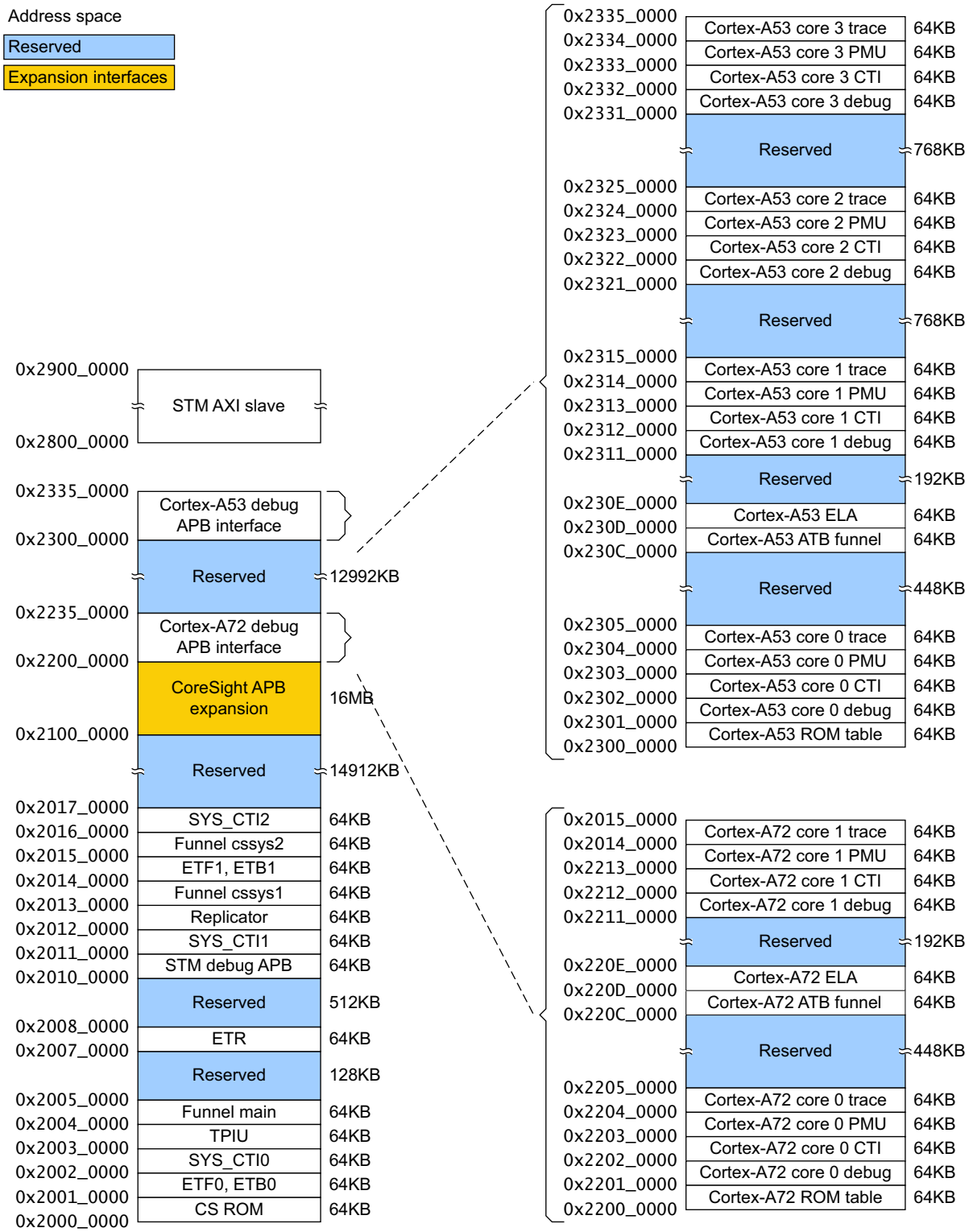


Figure 3-5 CoreSight debug and trace region memory map

System peripherals regions

Figure 3-6 shows the system peripherals region.

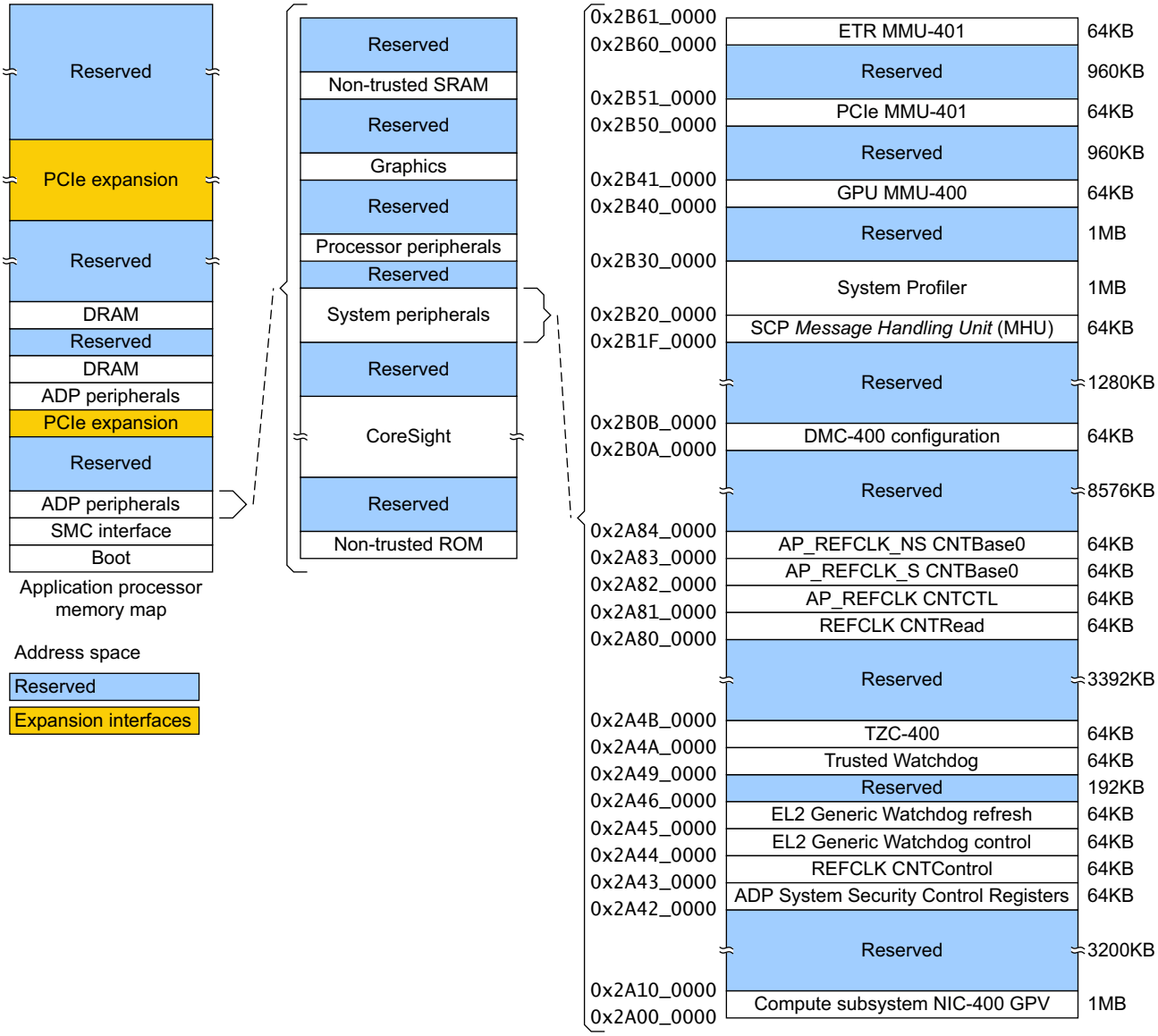


Figure 3-6 System peripherals memory map

Processor peripherals region

The processor peripherals region is Reserved for peripherals that the application processors require.

This area contains the following:

- GIC-400. See the *ARM® CoreLink™ GIC-400 Generic Interrupt Controller Technical Reference Manual*.
- CCI-400 Programmers View. See the *ARM® CoreLink™ CCI-400 Cache Coherent Interconnect Technical Reference Manual*.
- GICv2m MSI unit. See *Message Signaled Interrupt (MSI) unit* on page 3-9.

Figure 3-7 shows the processor peripherals region memory map.

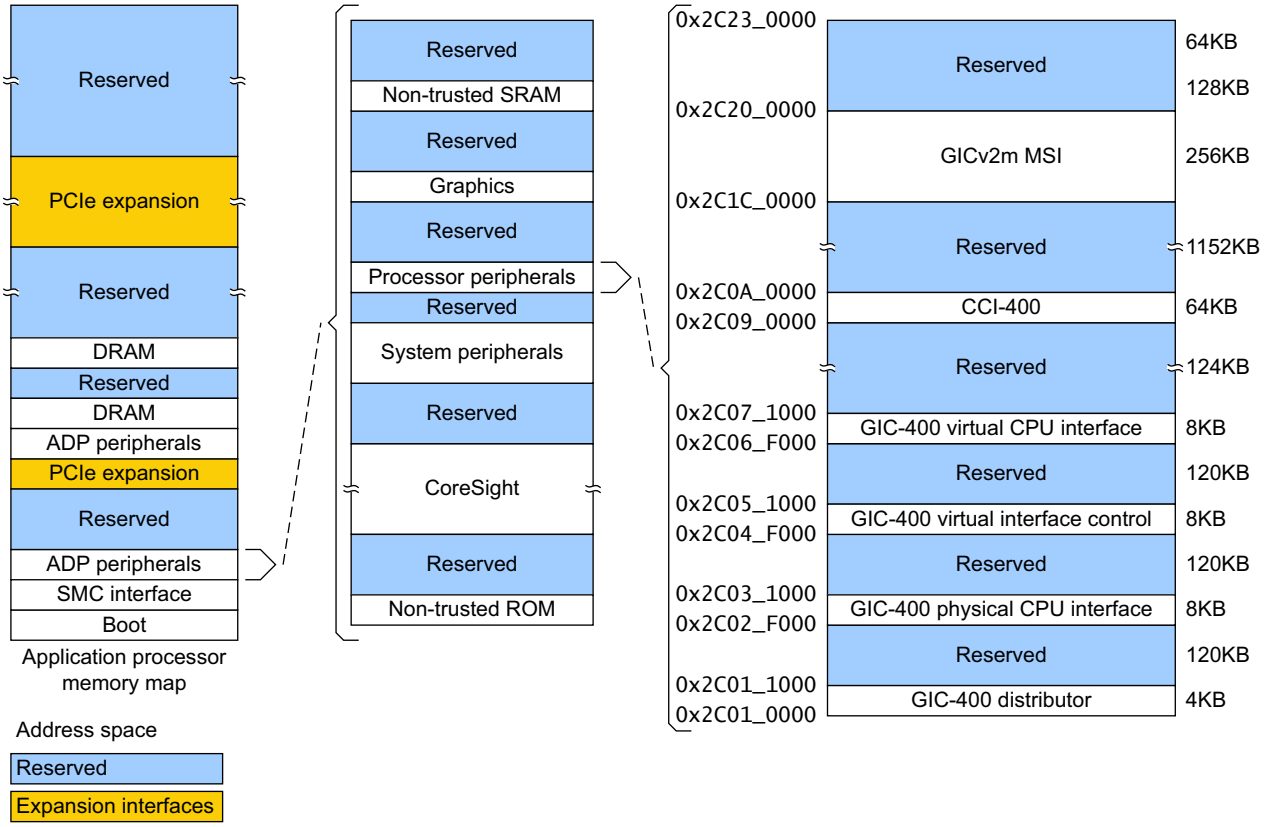


Figure 3-7 Processor peripherals region memory map

Graphics peripherals region

The graphics region contains graphics-related processing units. Figure 3-8 on page 3-18 shows the graphics region memory map.

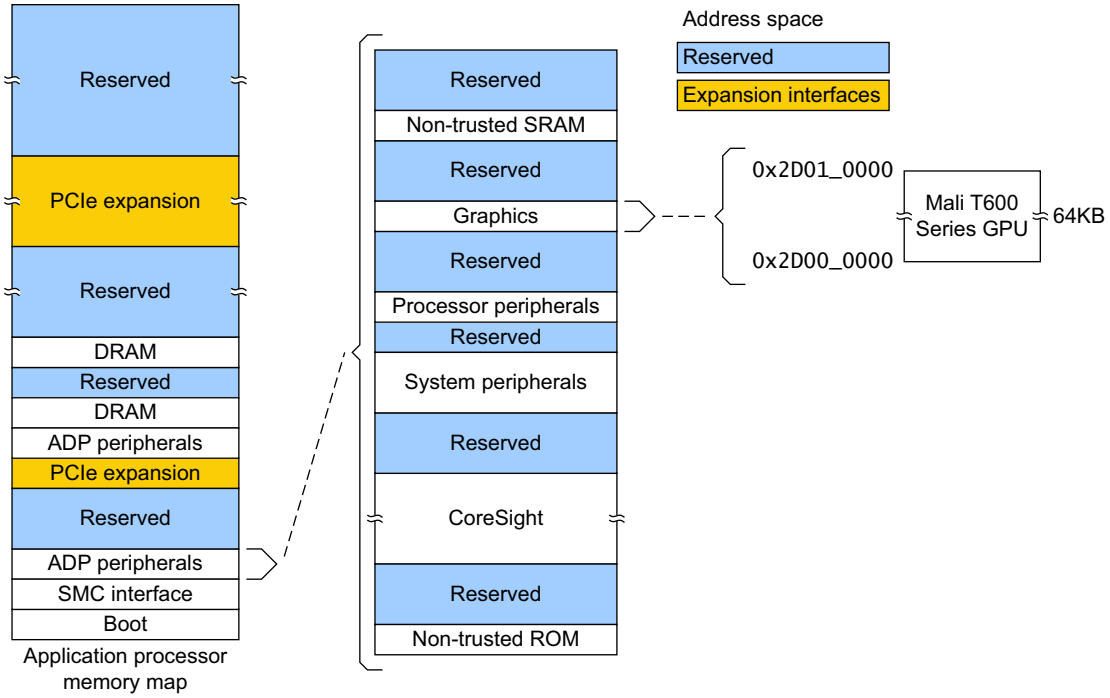


Figure 3-8 Graphics region memory map

Non-trusted SRAM

The Non-trusted SRAM is a 32KB scratch RAM that application processor software uses.

ADP Peripherals region, `0x6000_0000-0x8000_0000`

Figure 3-9 on page 3-19 shows that a 256MB region starting from address `0x00_1F00_0000` is defined as the ADP peripherals and memory area.

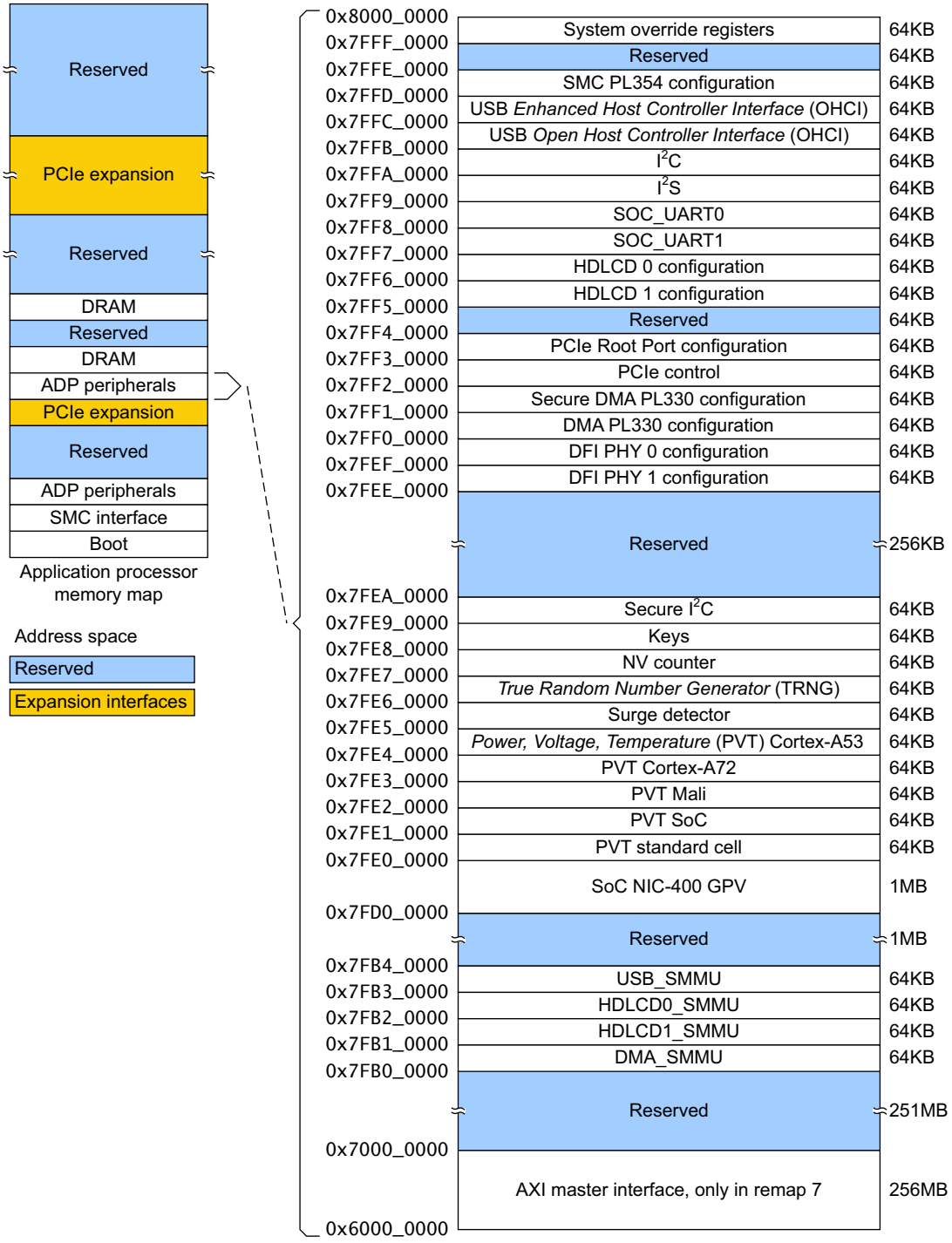


Figure 3-9 ADP peripherals region memory map, `0x6000_0000-0x8000_0000`

3.3.4 DRAM

The ADP can support up to a total of 512GB of DRAM, although the board supplies 8GB of DRAM. The DRAM regions of the memory map are provided in accordance with the *Principle of ARM Memory Maps* white paper. The ADP includes a DMC-400 Dynamic Memory Controller and a DDR3 PHY that enable it to access DRAM.

The CoreLink DMC-400 can protect areas of memory from unwanted memory map aliasing using address mask and address match registers. These registers determine whether a transaction to a location is to result in a DECERR response. For more information, see the *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual Supplement*.

Figure 3-10 shows the DRAM memory map.

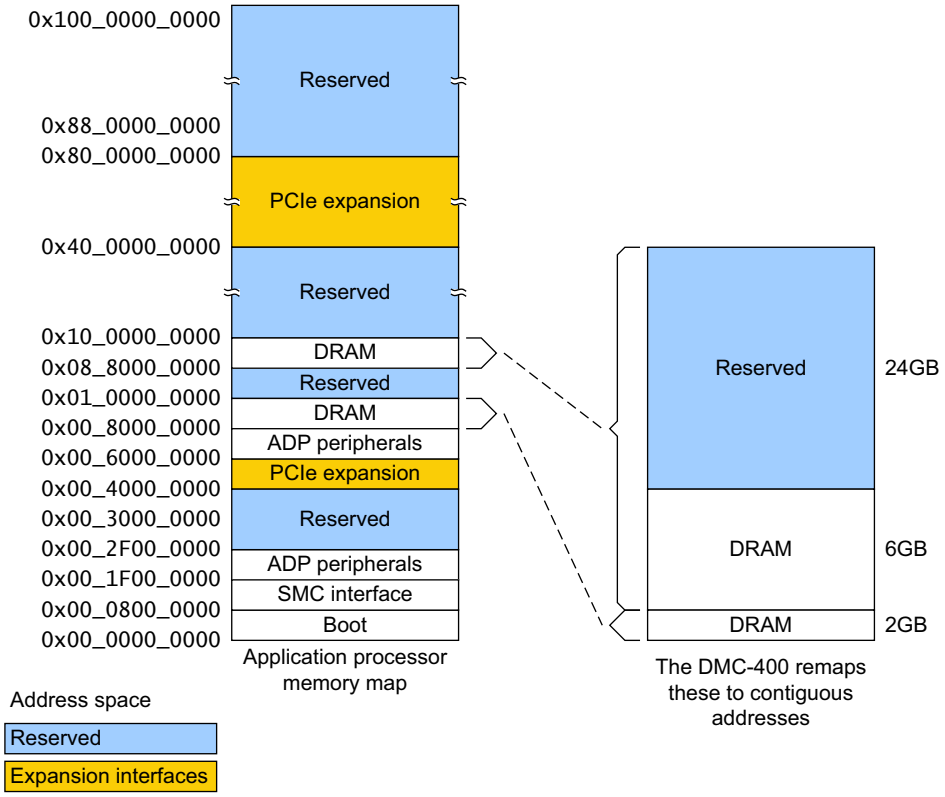


Figure 3-10 DRAM memory map

You must configure the DMC-400 to remap the distributed areas so that they form a contiguous area before accessing DRAM as Figure 3-10 shows. You must configure the DMC-400 configuration registers during Secure boot so that when access targets any unpopulated areas of memory in the DMC-400, it returns a DECERR, including any access that is beyond 512GB.

See the *ARM® CoreLink™ DMC-400 Dynamic Memory Controller Technical Reference Manual*.

An option exists to program up to eight Secure or Non-secure access regions, excluding the base region, in the external memory using the TZC-400 Address Space Controller. See *CoreLink TZC-400 TrustZone Address Space Controller security* on page 2-22.

3.3.5 Application memory map summary

Table 3-5 shows a summary of the application memory map.

Table 3-5 Application memory map

Name	Base address	Top address	Size	Security
Trusted boot ROM	0x00_0000_0000	0x00_0000_FFFF	64KB	Secure access only
Reserved	0x00_0001_0000	0x00_03FF_FFFF	65472KB	-
Trusted RAM	0x00_0400_0000	0x00_0403_FFFF	256KB	Secure access only
Reserved	0x00_0404_0000	0x00_07FF_FFFF	65280KB	-
SMC interface	0x00_0800_0000	0x00_1EFF_FFFF	368MB	
Non-trusted ROM	0x00_1F00_0000	0x00_1F00_0FFF	4KB	Secure and Non-secure access
Reserved	0x00_1F00_1000	0x00_1FFF_FFFF	16380KB	-
CS ROM	0x00_2000_0000	0x00_2000_FFFF	64KB	Programmable access security
ETF0, ETB0	0x00_2001_0000	0x00_2001_FFFF	64KB	
SYS_CTI0	0x00_2002_0000	0x00_2002_FFFF	64KB	
TPIU	0x00_2003_0000	0x00_2003_FFFF	64KB	
Funnel main cssys0	0x00_2004_0000	0x00_2004_FFFF	64KB	
Reserved	0x00_2005_0000	0x00_2006_FFFF	128KB	-
ETR	0x00_2007_0000	0x00_2007_FFFF	64KB	Programmable access security
Reserved	0x00_2008_0000	0x00_200F_FFFF	512KB	-
STM Debug APB	0x00_2010_0000	0x00_2010_FFFF	64KB	Programmable access security
SYS_CTI1	0x00_2011_0000	0x00_2011_FFFF	64KB	
Replicator	0x00_2012_0000	0x00_2012_FFFF	64KB	
Funnel cssys1	0x00_2013_0000	0x00_2013_FFFF	64KB	
ETF1, ETB1	0x00_2014_0000	0x00_2014_FFFF	64KB	
Funnel cssys2	0x00_2015_0000	0x00_2015_FFFF	64KB	
SYS_CTI2	0x00_2016_0000	0x00_2016_FFFF	64KB	
Reserved	0x00_2017_0000	0x00_20FF_FFFF	14912KB	-
CoreSight APB Expansion	0x00_2100_0000	0x00_21FF_FFFF	16MB	Programmable access security
Cortex-A72 ROM Table	0x00_2200_0000	0x00_2200_FFFF	64KB	
Cortex-A72 core 0 Debug	0x00_2201_0000	0x00_2201_FFFF	64KB	
Cortex-A72 core 0 CTI	0x00_2202_0000	0x00_2202_FFFF	64KB	
Cortex-A72 core 0 PMU	0x00_2203_0000	0x00_2203_FFFF	64KB	
Cortex-A72 core 0 Trace	0x00_2204_0000	0x00_2204_FFFF	64KB	
Reserved	0x00_2205_0000	0x00_220B_FFFF	448KB	-

Table 3-5 Application memory map (continued)

Name	Base address	Top address	Size	Security
Cortex-A72 ATB Funnel	0x00_220C_0000	0x00_220C_FFFF	64KB	Programmable access security
Cortex-A72 ATB ELA	0x00_220D_0000	0x00_220D_FFFF	64KB	
Reserved	0x00_220E_0000	0x00_2210_FFFF	192KB	-
Cortex-A72 core 1 Debug	0x00_2211_0000	0x00_2211_FFFF	64KB	Programmable access security
Cortex-A72 core 1 CTI	0x00_2212_0000	0x00_2212_FFFF	64KB	
Cortex-A72 core 1 PMU	0x00_2213_0000	0x00_2213_FFFF	64KB	
Cortex-A72 core 1 Trace	0x00_2214_0000	0x00_2214_FFFF	64KB	
Reserved	0x00_2215_0000	0x00_22FF_FFFF	15040KB	-
Cortex-A53 ROM Table	0x00_2300_0000	0x00_2300_FFFF	64KB	Programmable access security
Cortex-A53 core 0 Debug	0x00_2301_0000	0x00_2301_FFFF	64KB	
Cortex-A53 core 0 CTI	0x00_2302_0000	0x00_2302_FFFF	64KB	
Cortex-A53 core 0 PMU	0x00_2303_0000	0x00_2303_FFFF	64KB	
Cortex-A53 core 0 Trace	0x00_2304_0000	0x00_2304_FFFF	64KB	
Reserved	0x00_2305_0000	0x00_230B_FFFF	448KB	-
Cortex-A53 ATB Funnel	0x00_230C_0000	0x00_230C_FFFF	64KB	Programmable access security
Cortex-A53 ATB ELA	0x00_230D_0000	0x00_230D_FFFF	64KB	
Reserved	0x00_230E_0000	0x00_230E_FFFF	256KB	-
Cortex-A53 core 1 Debug	0x00_2311_0000	0x00_2311_FFFF	64KB	Programmable access security
Cortex-A53 core 1 CTI	0x00_2312_0000	0x00_2312_FFFF	64KB	
Cortex-A53 core 1 PMU	0x00_2313_0000	0x00_2313_FFFF	64KB	
Cortex-A53 core 1 Trace	0x00_2314_0000	0x00_2314_FFFF	64KB	
Reserved	0x00_2315_0000	0x00_2320_FFFF	768KB	-
Cortex-A53 core 2 Debug	0x00_2321_0000	0x00_2321_FFFF	64KB	Programmable access security
Cortex-A53 core 2 CTI	0x00_2322_0000	0x00_2322_FFFF	64KB	
Cortex-A53 core 2 PMU	0x00_2323_0000	0x00_2323_FFFF	64KB	
Cortex-A53 core 2 Trace	0x00_2324_0000	0x00_2324_FFFF	64KB	
Reserved	0x00_2325_0000	0x00_2330_FFFF	768KB	-
Cortex-A53 core 3 Debug	0x00_2331_0000	0x00_2331_FFFF	64KB	Programmable access security
Cortex-A53 core 3 CTI	0x00_2332_0000	0x00_2332_FFFF	64KB	
Cortex-A53 core 3 PMU	0x00_2333_0000	0x00_2333_FFFF	64KB	
Cortex-A53 core 3 Trace	0x00_2334_0000	0x00_2334_FFFF	64KB	
Reserved	0x00_2335_0000	0x00_27FF_FFFF	78528KB	-
STM AXI Slave	0x00_2800_0000	0x00_28FF_FFFF	16MB	Secure and Non-secure access

Table 3-5 Application memory map (continued)

Name	Base address	Top address	Size	Security
Reserved	0x00_2900_0000	0x00_29FF_FFFF	16MB	-
Compute subsystem NIC-400 GPV	0x00_2A00_0000	0x00_2A0F_FFFF	1MB	Secure access only
Reserved	0x00_2A10_0000	0x00_2A41_FFFF	3200KB	-
ADP System Security Control Registers	0x00_2A42_0000	0x00_2A42_FFFF	64KB	Secure access only
AON_REF_CLK CNTControl	0x00_2A43_0000	0x00_2A43_FFFF	64KB	
EL2 Generic Watchdog Control	0x00_2A44_0000	0x00_2A44_FFFF	64KB	Secure and Non-secure access
EL2 Generic Watchdog Refresh	0x00_2A45_0000	0x00_2A45_FFFF	64KB	
Reserved	0x00_2A46_0000	0x00_2A48_FFFF	192KB	-
Trusted Watchdog	0x00_2A49_0000	0x00_2A49_FFFF	64KB	Secure access only
TZC-400	0x00_2A4A_0000	0x00_2A4A_FFFF	64KB	
Reserved	0x00_2A4B_0000	0x00_2A7F_FFFF	3392KB	-
AON_REF_CLK CNTRead	0x00_2A80_0000	0x00_2A80_FFFF	64KB	Secure and Non-secure access
Application processor **AON_REF_CLK** CNTCTL	0x00_2A81_0000	0x00_2A81_FFFF	64KB	Device, peripheral, defined security
Application processor **AON_REF_CLK** CNTBase0	0x00_2A82_0000	0x00_2A82_FFFF	64KB	Secure access only
Application processor **AON_REF_CLK** CNTBase1	0x00_2A83_0000	0x00_2A83_FFFF	64KB	Secure and Non-secure access
Reserved	0x00_2A84_0000	0x00_2B09_FFFF	8576KB	-
DMC-400 CFG	0x00_2B0A_0000	0x00_2B0A_FFFF	64KB	Programmable access security
Reserved	0x00_2B0B_0000	0x00_2B1E_FFFF	1280KB	-
SCP_MHU	0x00_2B1F_0000	0x00_2B1F_FFFF	64KB	Device, peripheral, defined security
System Profiler	0x00_2B20_0000	0x00_2B2F_FFFF	1MB	
Reserved	0x00_2B30_0000	0x00_2B3F_FFFF	1MB	-
GPU MMU-400	0x00_2B40_0000	0x00_2B40_FFFF	64KB	Device, peripheral, defined security
Reserved	0x00_2B41_0000	0x00_2B4F_FFFF	960KB	-
PCIe MMU-401	0x00_2B50_0000	0x00_2B50_FFFF	64KB	Device, peripheral, defined security
Reserved	0x00_2B51_0000	0x00_2B5F_FFFF	960KB	-
ETR MMU-401	0x00_2B60_0000	0x00_2B60_FFFF	64KB	Device, peripheral, defined security
Reserved	0x00_2B61_0000	0x00_2C0F_FFFF	10240KB	-
GIC Distributor	0x00_2C01_0000	0x00_2C01_0FFF	4KB	Device, peripheral, defined security
Reserved	0x00_2C01_1000	0x00_2C02_EFFF	120KB	-
GIC physical CPU interface	0x00_2C02_F000	0x00_2C03_0FFF	8KB	Device, peripheral, defined security
Reserved	0x00_2C03_1000	0x00_2C04_EFFF	120KB	-

Table 3-5 Application memory map (continued)

Name	Base address	Top address	Size	Security
GIC virtual interface control	0x00_2C04_F000	0x00_2C05_0FFF	8KB	Device, peripheral, defined security
Reserved	0x00_2C05_1000	0x00_2C06_EFFF	120KB	-
GIC virtual CPU interface	0x00_2C06_F000	0x00_2C07_0FFF	8KB	Device, peripheral, defined security
Reserved	0x00_2C07_1000	0x00_2C08_FFFF	124KB	-
CCI_PV	0x00_2C09_0000	0x00_2C09_FFFF	64KB	Device, peripheral, defined security
Reserved	0x00_2C0A_0000	0x00_2C1B_FFFF	1152KB	-
GICv2m MSI Frame 0	0x00_2C1C_0000	0x00_2C1C_FFFF	64KB	Secure and Non-secure access
GICv2m MSI Frame 1	0x00_2C1D_0000	0x00_2C1D_FFFF	64KB	Secure and Non-secure access
GICv2m MSI Frame 2	0x00_2C1E_0000	0x00_2C1E_FFFF	64KB	Secure and Non-secure access
GICv2m MSI Frame 3	0x00_2C1F_0000	0x00_2C1F_FFFF	64KB	Secure and Non-secure access
Reserved	0x00_2C20_0000	0x00_2C22_FFFF	192KB	-
Reserved	0x00_2C23_0000	0x00_2CFF_FFFF	14144KB	
Mali-T624 GPU	0x00_2D00_0000	0x00_2D00_FFFF	64KB	Programmable access security
Reserved	0x00_2D01_0000	0x00_2DFF_FFFF	16320KB	-
Non-Trusted SRAM	0x00_2E00_0000	0x00_2E00_7FFF	32KB	Secure and Non-secure access
Reserved	0x00_2E00_8000	0x00_2EFF_FFFF	16352KB	-
Reserved	0x00_2F00_0000	0x00_2FFF_FFFF	16MB	User-defined, exported security
Reserved	0x00_3000_0000	0x00_3FFF_FFFF	256MB	
PCIe Enhanced Configuration Access Mechanism	0x00_4000_0000	0x00_4FFF_FFFF	256MB	
PCIe memory address space	0x00_5000_0000	0x00_5FFF_FFFF	256MB	
Thin-Links master interface	0x00_6000_0000	0x00_6FFF_FFFF	256MB	
Reserved	0x00_7000_0000	0x00_7FAF_FFFF	251MB	
DMA MMU-401	0x00_7FB0_0000	0x00_7FB0_FFFF	64KB	Device, peripheral, defined security
HDLCD1 MMU-401	0x00_7FB1_0000	0x00_7FB1_FFFF	64KB	
HDLCD0 MMU-401	0x00_7FB2_0000	0x00_7FB2_FFFF	64KB	
USB MMU-401	0x00_7FB3_0000	0x00_7FB3_FFFF	64KB	
SoC Interconnect NIC-400 GPV	0x00_7FD0_0000	0x00_7FDF_FFFF	1MB	Secure access only

Table 3-5 Application memory map (continued)

Name	Base address	Top address	Size	Security
PVT Monitor, standard cell	0x00_7FE0_0000	0x00_7FE0_0FFF	4KB	Programmable access security
PVT Monitor, SoC	0x00_7FE1_0000	0x00_7FE1_0FFF	4KB	
PVT Monitor, Mali-T624 GPU	0x00_7FE2_0000	0x00_7FE2_0FFF	4KB	
PVT Monitor, Cortex-A72	0x00_7FE3_0000	0x00_7FE3_0FFF	4KB	
PVT Monitor, Cortex-A53	0x00_7FE4_0000	0x00_7FE4_0FFF	4KB	
Surge detector	0x00_7FE5_0000	0x00_7FE5_0FFF	4KB	
TRNG	0x00_7FE6_0000	0x00_7FE6_0FFF	4KB	Secure access only
Trusted Non-Volatile Counters	0x00_7FE7_0000	0x00_7FE7_0FFF	4KB	
Trusted Root-Key Storage	0x00_7FE8_0000	0x00_7FE8_0FFF	4KB	
Secure I^2C	0x00_7FE9_0000	0x00_7FE9_00FF	256B	
DDR3 PHY 1	0x00_7FEE_0000	0x00_7FEE_FFFF	64KB	Programmable access security
DDR3 PHY 0	0x00_7FEF_0000	0x00_7FEF_FFFF	64KB	
DMA Non-secure	0x00_7FF0_0000	0x00_7FFF_0FFF	4KB	
DMA Secure	0x00_7FF1_0000	0x00_7FF1_0FFF	4KB	Secure access only
PCIe Control Registers	0x00_7FF2_0000	0x00_7FF2_FFFF	64KB	Device, peripheral, defined security
PCIe Root Port Internal Configuration Registers	0x00_7FF3_0000	0x00_7FF3_0FFF	4KB	
HDLCD 1	0x00_7FF5_0000	0x00_7FF5_0FFF	4KB	Programmable access security
HDLCD 0	0x00_7FF6_0000	0x00_7FF6_0FFF	4KB	
SoC UART 1	0x00_7FF7_0000	0x00_7FF7_0FFF	4KB	
SoC UART 0	0x00_7FF8_0000	0x00_7FF8_0FFF	4KB	
I^2S	0x00_7FF9_0000	0x00_7FF9_03FF	1KB	
I^2C	0x00_7FFA_0000	0x00_7FFA_00FF	256B	
USB OHCI	0x00_7FFB_0000	0x00_7FFB_0FFF	4KB	
USB EHCI	0x00_7FFC_0000	0x00_7FFC_0FFF	4KB	
PL354	0x00_7FFD_0000	0x00_7FFD_0FFF	4KB	
System Override Registers	0x00_7FFF_0000	0x00_7FFF_0FFF	4KB	Secure access only
DRAM	0x00_8000_0000	0x00_FFFF_FFFF	2GB	Programmable access security
Reserved	0x01_0000_0000	0x08_7FFF_FFFF	30GB	-
DRAM	0x08_8000_0000	0x0F_FFFF_FFFF	30GB	Programmable access security
Reserved	0x10_0000_0001	0x3F_FFFF_FFFF	192GB	-

Table 3-5 Application memory map (continued)

Name	Base address	Top address	Size	Security
PCIe memory address space	0x40_0000_0000	0x7F_FFFF_FFFF	256GB	User-defined, exported security
Reserved	0x80_0000_0000	0x87_FFFF_FFFF	32GB	-
DRAM	0x88_0000_0000	0xFF_FFFF_FFFF	480GB	Programmable access security

3.4 ADP System Security Control Registers

The ADP *System Security Control* (SSC) Registers Unit provides an interface for controlling the system-wide security features of the ADP. The features include the following:

- Providing the selection and internal sources for debug authentication signals.
- Controls to drive all Secure lock down type signals.
- Controls to enable or disable Secure watchdog halt on debug functionality.
- General purpose register for Secure state storage.

These registers are located in the AON power domain and their states are maintained even if all the system has been powered down, except for the removal of VSYS.

UNDEFINED addresses are Reserved.

3.4.1 About the ADP programmers model

The programmer-visible state in the ADP consists of the following domains:

- Private register map of the application processor.
- ADP peripherals register map that is equally visible to both the application processor and the SCP.
- Private register map of the SCP.
- Debug and trace programmer visible state.

Most of the programmer-visible register state is composed of standard ARM IP configurations and control registers that integral to the components that constitute ADP.

The following sections are not comprehensive descriptions, or even a full summary of all the visible register states. They summarize the register states of peripherals that are not part of the standard ARM IP distributions, and have been developed specifically for the ADP with the Technical Reference Manuals for all the standard ARM IP register states.

3.4.2 Register summary

Table 3-6 shows the SSC registers in offset order from the base memory address.

The SSC_BASE is the base address of the SSC. All registers in this module are Secure access only. If a Non-secure access attempts to access these Secure registers, or any unmapped unimplemented registers within the 4KB region starting from SSC_BASE, they receive a DECERR response. Any Secure accesses to unimplemented areas within the 4KB region starting from SSC_BASE RAZ, WI.

Table 3-6 System Security Control Register summary

Offset	Name	Type	Reset	Width	Description
0x000	SSC_ICCFG_STAT	RO	0x00000000	32	*SSC_ICCFG_STAT Register* on page 3-28
0x004	SSC_ICCFG_SET	WO	-	32	*SSC_ICCFG_SET Register* on page 3-28
0x008	SSC_ICCFG_CLR	WO	-	32	*SSC_ICCFG_CLR Register* on page 3-29
0x010	SSC_DBGCFG_STAT	RO	0x00000000	32	*SSC_DBGCFG_STAT Register* on page 3-29
0x014	SSC_DBGCFG _SET	WO	-	32	*SSC_DBGCFG_SET Register* on page 3-30
0x018	SSC_DBGCFG _CLR	WO	-	32	*SSC_DBGCFG_CLR Register* on page 3-31

Table 3-6 System Security Control Register summary (continued)

Offset	Name	Type	Reset	Width	Description
0x01C	SSC_AUXDBCFG	RW	-	32	*SSC_AUXDBGCFG Register* on page 3-32
0x024	SSC_SWDHOD	RW	0x00000000	32	*SSC_SWDHOD Register* on page 3-33
0x030	SSC_GPRETN	RW	0x00000000	32	*SSC_GPRETN Register* on page 3-34
0x040	SSC_VERSION	RO	0x00041030	32	*SSC_VERSION Register* on page 3-34
0xFD0	SSC_PID_4	RO	0x00000004	32	*SSC_PID_4 Register* on page 3-35
0xFE0	SSC_PID_0	RO	0x00000044	32	*SSC_PID_0 Register* on page 3-36
0xFE4	SSC_PID_1	RO	0x000000B8	32	*SSC_PID_1 Register* on page 3-36
0xFE8	SSC_PID_2	RO	0x0000000B	32	*SSC_PID_2 Register* on page 3-36
0xFEC	SSC_PID_3	RO	0x00000000	32	*SSC_PID_3 Register* on page 3-37
0xFF0	COMP_ID0	RO	0x0000000D	32	*COMP_ID0 Register* on page 3-37
0xFF4	COMP_ID1	RO	0x000000F0	32	*COMP_ID1 Register* on page 3-38
0xFF8	COMP_ID2	RO	0x00000005	32	*COMP_ID2 Register* on page 3-38
0xFFC	COMP_ID3	RO	0x000000B1	32	*COMP_ID3 Register* on page 3-38

3.4.3 Register descriptions

This section describes the System Security Control registers. Table 3-6 on page 3-27 provides cross references to individual registers.

SSC_ICCFG_STAT Register

The SSC_ICCFG_STAT Register characteristics are:

Purpose A Secure access only memory-mapped register. It configures the security level of accesses that masters issue into the NIC-400 interconnect. This register is empty and all fields are Reserved, and RAZ.

To set or clear any fields within the register, use *SSC_ICCFG_SET Register* and *SSC_ICCFG_CLR Register* on page 3-29.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

SSC_ICCFG_SET Register

The SSC_ICCFG_SET Register characteristics are:

Purpose A Secure access only memory-mapped register. It sets bits in the SSC_ICCFG_STAT register. This register is empty and all fields are Reserved, and RAZ.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

SSC_ICCFG_CLR Register

The SSC_ICCFG_CLR Register characteristics are:

Purpose A Secure access only memory-mapped register. It clears bits in the SSC_ICCFG_STAT register. This register is empty and all fields are Reserved, and RAZ.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

SSC_DBGCFG_STAT Register

The SSC_DBGCFG_STAT Register characteristics are:

Purpose The SSC_DBGCFG_STAT register is a Secure access only memory-mapped register that:

- Controls how to drive the debug authentication signals either:
  - From an external source to the subsystem.
  - Internally using build-in register bits, also implemented in this register.
- Defines the values of the debug authentication signals when you configure them to be internally driven.

To set or clear any fields within the register, use *SSC_DBGCFG_SET Register* on page 3-30 and *SSC_DBGCFG_CLR Register* on page 3-31.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-11 shows the bit assignments.

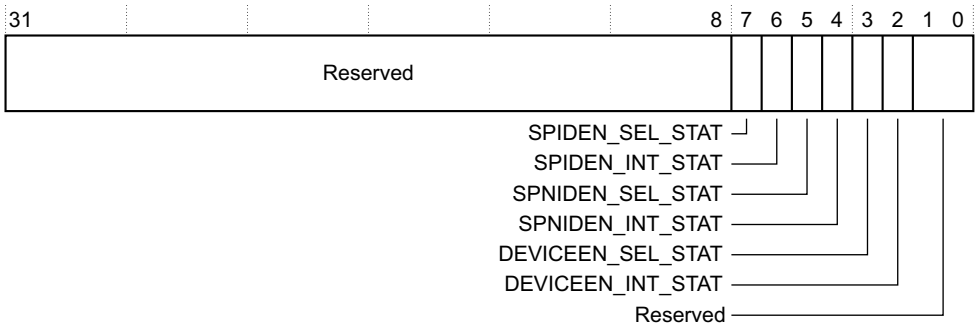


Figure 3-11 SSC_DBGCFG_STAT Register bit assignments

Table 3-7 shows the bit assignments.

Table 3-7 SSC_DBGCFG_STAT Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7]	SPIDEN_SEL_STAT	**SPIDEN** drive selection: 0 Debug Authentication Register in SCC. 1 SPIDEN_INT_STAT in this register. If you select the Debug Authentication Register in SCC mode, an ADP top-level configuration input, **SPIDEN_CFG SPIDEN** drives **SPIDEN_CFG**.
[6]	SPIDEN_INT_STAT	**SPIDEN** internal drive value.
[5]	SPNIDEN_SEL_STAT	**SPNIDEN** external or internal drive selection: 0 Debug Authentication Register in SCC. 1 SPIDEN_INT_STAT in this register. If you select the Debug Authentication Register in SCC mode, an ADP top-level configuration input drives **SPNIDEN**.
[4]	SPNIDEN_INT_STAT	**SPNIDEN** internal drive value.
[3]	DEVICEEN_SEL_STAT	**DEVICEEN** external or internal drive selection: 0 Debug Authentication Register in SCC. 1 DEVICEEN_INT_STAT in this register. If you select the Debug Authentication Register in SCC mode, an ADP top-level configuration input drives **DEVICEEN**.
[2]	DEVICEEN_INT_STAT	**DEVICEEN** internal drive value.
[1:0]	-	Reserved. RAZ.

SSC_DBGCFG_SET Register

The SSC_DBGCFG_SET Register characteristics are:

Purpose A Secure access only memory-mapped register. It is associated with the *SSC_DBGCFG_STAT Register* on page 3-29, and when any field is written with a HIGH, the corresponding bit in the *SSC_DBGCFG_SET Register* is set to HIGH. This register always reads as zero.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-12 on page 3-31 shows the bit assignments.

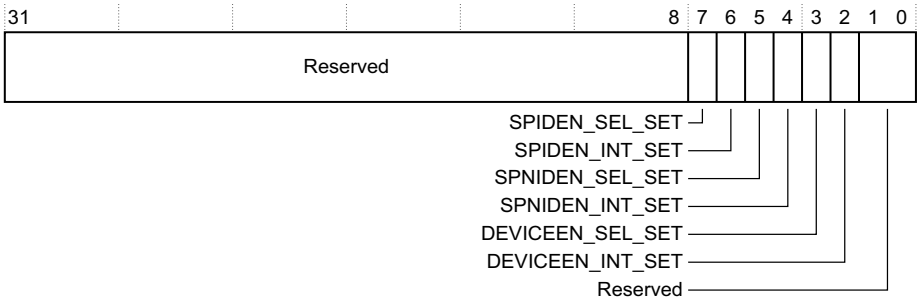


Figure 3-12 SSC_DBGCFG_SET Register bit assignments

Table 3-8 shows the bit assignments.

Table 3-8 SSC_DBGCFG_SET Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7]	SPIDEN_SEL_SET	Sets **SPIDEN** external or internal drive selection, **SPIDEN_SEL_STAT**, to HIGH.
[6]	SPIDEN_INT_SET	Sets **SPIDEN** internal drive value, **SPIDEN_INT_STAT**, to HIGH.
[5]	SPNIDEN_SEL_SET	Sets **SPNIDEN** external or internal drive selection, **SPNIDEN_SEL_STAT**, to HIGH.
[4]	SPNIDEN_INT_SET	Sets **SPNIDEN** internal drive value, **SPNIDEN_INT_STAT**, to HIGH.
[3]	DEVICEEN_SEL_SET	Sets **DEVICEEN** external or internal drive selection, **DEVICEEN_SEL_STAT**, to HIGH.
[2]	DEVICEN_INT_SET	Sets **DEVICEEN** internal drive value, **DEVICEEN_INT_STAT**, to HIGH.
[1:0]	-	Reserved. WI.

SSC_DBGCFG_CLR Register

The SSC_DBGCFG_CLR Register characteristics are:

Purpose A Secure access only memory-mapped register. This register is associated with the SSC_DBGCFG_STAT register, and when any field in this register is set to HIGH, the corresponding bit in the *SSC_DBGCFG_STAT Register* on page 3-29 is set to LOW. This register always reads as zero.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-13 on page 3-32 shows the bit assignments.

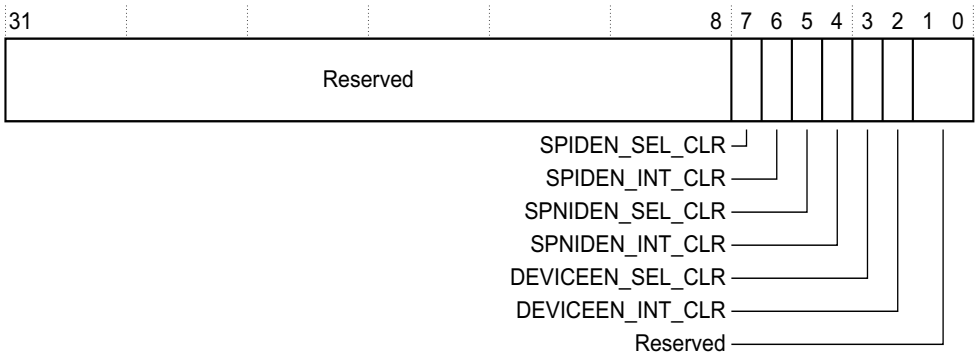


Figure 3-13 SSC_DBGCFG_CLR Register bit assignments

Table 3-9 shows the bit assignments.

Table 3-9 SSC_DBGCFG_CLR Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7]	SPIDEN_SEL_CLR	Clears **SPIDEN** external or internal drive selection, **SPIDEN_SEL_STAT**, to LOW.
[6]	SPIDEN_INT_CLR	Clears **SPIDEN** internal drive value, **SPIDEN_INT_STAT**, to LOW.
[5]	SPNIDEN_SEL_CLR	Clears **SPNIDEN** external or internal drive selection, **SPNIDEN_SEL_STAT**, to LOW.
[4]	SPNIDEN_INT_CLR	Clears **SPNIDEN** internal drive value, **SPNIDEN_INT_STAT**, to LOW.
[3]	DEVICEEN_SEL_CLR	Clears **DEVICEEN** external or internal drive selection, **DEVICEEN_SEL_STAT**, to LOW.
[2]	DEVICEEN_INT_CLR	Clears **DEVICEEN** internal drive value, **DEVICEEN_INT_STAT**, to LOW.
[1:0]	-	Reserved. RAZ.

SSC_AUXDBGCFG Register

The SSC_AUXDBGCFG Register characteristics are:

Purpose Auxiliary Debug Configuration Register. A Secure access only read and write register. This register provides override control of the **DBGEN** and **NIDEN** debug authentication signals.

Usage constraints Setting any of the bits in this register violates compliance with the *ARM Architecture Standard Configurations Platform Design Document*, and requires self-hosted debug to always be present. To disable the use of external debuggers, drive **DEVICEEN** LOW using the SSC_DBGCFG registers instead.

Note

ARM strongly recommends not to use this register, and to leave both bits at their reset values.

Attributes See Table 3-6 on page 3-27.

Figure 3-14 on page 3-33 shows the bit assignments.

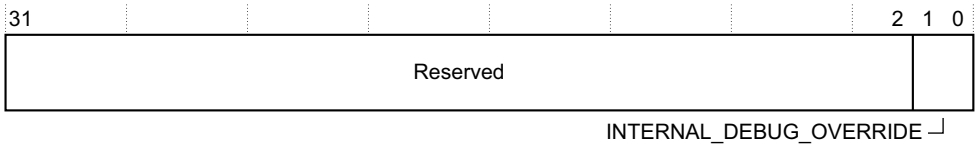


Figure 3-14 SSC_AUXDBGCFG Register bit assignments

Table 3-10 shows the bit assignments.

Table 3-10 SSC_AUXDBGCFG Register bit assignments

Bits	Name	Description
[31:2]	-	Reserved. RAZ, WI.
[1:0]	INTERNAL_DEBUG_OVERRIDE	The settings are as follows: `b00` Non-secure self-hosted debug is enabled. This value is the reset value. `b11` **or** `b10` Both invasive and non-invasive Non-secure self-hosted debug are disabled. `b01` Invasive Non-secure self-hosted debug is disabled, but non-invasive Non-secure self-hosted debug remains enabled. The **DBGEN** inputs to the application processors are driven LOW, and the **NIDEN** inputs to the application processors are driven HIGH.

SSC_SWDHOD Register

The SSC_SWDHOD Register characteristics are:

Purpose A Secure access only read and write register. It drives control signals that enable the Halt On Debug functionality of both the SCP watchdog timer, and the system Secure watchdog.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-15 shows the bit assignments.

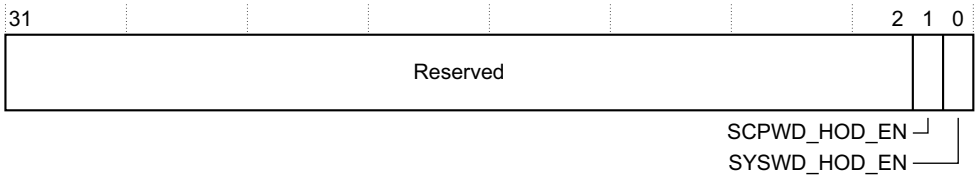


Figure 3-15 SSC_SWDHOD Register bit assignments

Table 3-11 shows the bit assignments.

Table 3-11 SSC_SWDHOD Register bit assignments

Bits	Name	Description
[31:2]	-	Reserved. RAZ, WI.
[1]	SCPWD_HOD_EN	Setting this bit to HIGH enables the Halt On Debug functionality of the watchdog timer in the SCP subsystem. When enabled, you can halt the SCP Watchdog using the cross trigger network. Setting this field to LOW disables Halt on Debug for the SCP watchdog.
[0]	SYSWD_HOD_EN	Setting this bit to HIGH enables the Halt On Debug functionality of the trusted watchdog in the main subsystem. When enabled, you can halt the trusted watchdog using the cross trigger network. Setting this bit to LOW disables Halt on Debug for the Secure watchdog in the main system.

SSC_GPRETN Register

The SSC_GPRETN Register characteristics are:

Purpose A Secure access only read and write memory-mapped register that provides 16 bits of general storage space for security purposes. The SSC_GPRETN Register resets only on a system powerup reset.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-16 shows the bit assignments.

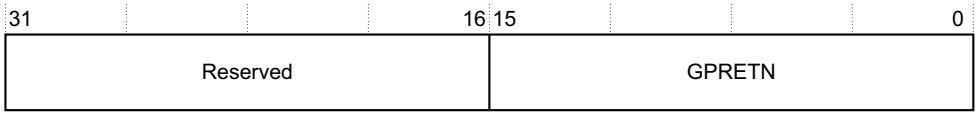


Figure 3-16 SSC_GPRETN Register bit assignments

Table 3-12 shows the bit assignments.

Table 3-12 SSC_GPRETN Register bit assignments

Bits	Name	Description
[31:16]	-	Reserved. RAZ.
[15:0]	GPRETN	General purpose register for Secure state storage.

SSC_VERSION Register

The SSC_VERSION Register characteristics are:

Purpose A Secure access only memory-mapped register that specifies the version ID for the ADP security feature.

Usage constraints Secure access only.

Attributes See Table 3-6 on page 3-27.

Figure 3-17 on page 3-35 shows the bit assignments.

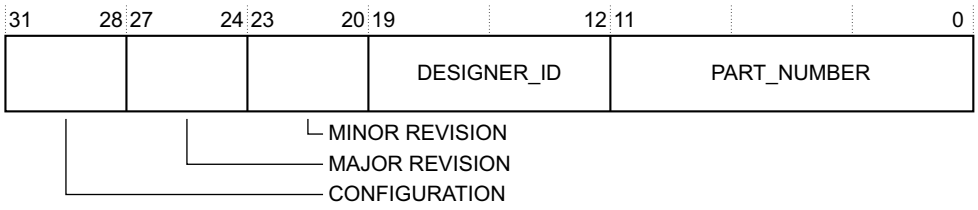


Figure 3-17 SSC_VERSION Register bit assignments

Table 3-13 shows the bit assignments.

Table 3-13 SSC_VERSION Register bit assignments

Bits	Name	Description
[31:28]	CONFIGURATION	Set to `0x0` because the ADP is not configurable.
[27:24]	MAJOR REVISION	
[23:20]	MINOR REVISION	
[19:12]	DESIGNER_ID	ARM product with designer code `0x41`
[11:0]	PART_NUMBER	`0x030` for the Juno ADP

Peripheral ID Registers

The Peripheral ID Register characteristics are:

Purpose Provides information to identify the component.

Usage constraints There are no usage constraints.

Attributes See Table 3-6 on page 3-27.

SSC_PID_4 Register

Figure 3-18 shows the bit assignments.

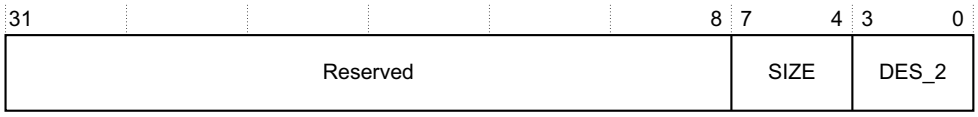


Figure 3-18 SSC_PID_4 Register bit assignments

Table 3-14 shows the bit assignments.

Table 3-14 SSC_PID_4 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	SIZE	Indicates the $\log_2$ of the number of 4KB blocks that the interface occupies. This field is set to `0x0`.
[3:0]	DES_2	JEP106 continuation code that identifies the designer. This field is set to `0x4` for ARM.

SSC_PID_0 Register

Figure 3-19 shows the bit assignments.

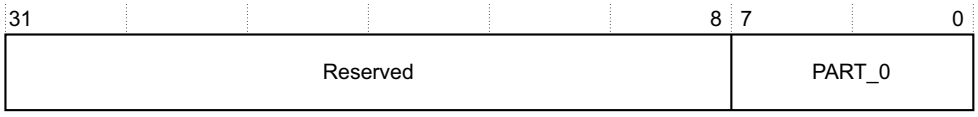


Figure 3-19 SSC_PID_0 Register bit assignments

Table 3-15 shows the bit assignments.

Table 3-15 SSC PID_0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	PART_0	Bits [7:0] of the part number. This field is set to `0x41`.

SSC_PID_1 Register

Figure 3-20 shows the bit assignments.

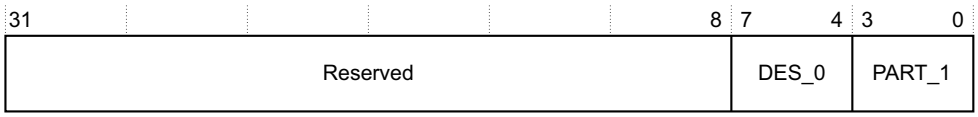


Figure 3-20 SSC_PID_1 Register bit assignments

Table 3-16 shows the bit assignments.

Table 3-16 SSC PID_1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	DES_0	Bits [3:0] of the JEP Identity. This field is set to `0xB` for ARM.
[3:0]	PART_1	Bits [11:8] of the part number. This field is set to `0x8`.

SSC_PID_2 Register

Figure 3-21 shows the bit assignments.

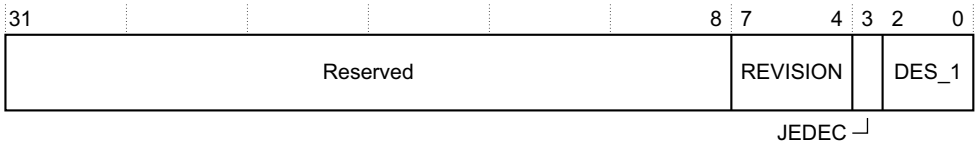


Figure 3-21 SSC_PID_2 Register bit assignments

Table 3-17 shows the bit assignments.

Table 3-17 SSC PID_2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	REVISION	This field is set to 0x1 for r0p1.
[3]	JEDEC	This bit is set to 0x1.
[2:0]	DES_1	Bits [6:4] of Designer field. This field is set to 0x3 for ARM.

SSC_PID_3 Register

Figure 3-22 shows the bit assignments.

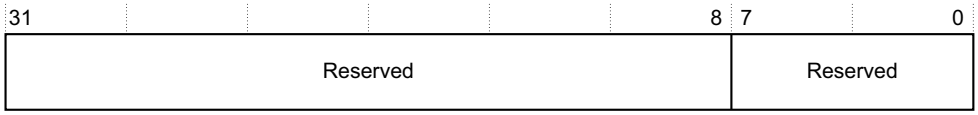


Figure 3-22 SSC_PID_3 Register bit assignments

Table 3-18 shows the bit assignments.

Table 3-18 SSC PID_3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	-	Reserved. RAZ.

COMP_ID Registers

The COMP_ID Registers characteristics are:

Purpose Provides information to identify the component.

Usage constraints There are no usage constraints.

Attributes See Table 3-6 on page 3-27.

COMP_ID0 Register

Table 3-19 shows the bit assignments.

Table 3-19 SSC COMP_ID0 Register

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID0	Reads as 0x0D.

COMP_ID1 Register

Table 3-20 shows the bit assignments.

Table 3-20 SSC COMP_ID1 Register

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID1	Reads as `0xF0`.

COMP_ID2 Register

Table 3-21 shows the bit assignments.

Table 3-21 SSC COMP_ID2 Register

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID2	Reads as `0x05`.

COMP_ID3 Register

Table 3-22 shows the bit assignments.

Table 3-22 SSC COMP_ID3 Register

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID3	Reads as `0xB1`.

3.4.4 Message Handling Unit (MHU)

The *Message Handling Unit* (MHU) is a memory-mapped peripheral that provides a mechanism to assert interrupt signals to facilitate inter-processor message passing between the SCP and the application processor. The message payload can be deposited into main memory or on-chip memories, and therefore, the MHU is used as a messaging signaling mechanism.

The MHU asserts the following interrupt signals:

- A high-priority Non-secure interrupt and a low-priority Non-secure interrupt in the application processor interrupt map.
- A Secure interrupt in the application processor interrupt map.
- A high-priority Non-secure interrupt and a low priority Non-secure interrupt in the SCP interrupt map.
- A Secure interrupt in the SCP interrupt map.

See *Application processor interrupt map* on page 3-3.

For each of the six interrupt signals, with a slight difference for the Secure interrupt to the application processor, the MHU drives the signal using a 32-bit register, with all 32 bits logically ORed together. The MHU provides a set of registers to enable software to set, clear, and check

the status of each of the bits of this register independently. 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 a type of event that can contribute to raising the interrupt.

From these memory-mapped registers, all registers that are associated with Secure interrupts are mapped as Secure access only, and the rest are accessible to Secure and Non-secure accesses. If Non-secure accesses attempt to access Secure registers, these accesses are RAZ, WI. If such a violation occurs, you can use software to configure the MHU to raise an interrupt to the application processor.

This interrupt is merged with the Secure application processor interrupt, and when enabled using the MHU_SCFG register, bit 31 of the SCP_INTR_S_STAT register is also used as the status for this interrupt.

All unmapped, unused areas within the 4KB region that the MHU occupies is Reserved and accesses targeting them are RAZ, WI.

Note

Software cannot set bit 31 of the MHU SCP_INTR_S register directly regardless of the settings of the *MHU_SCFG Register* on page 3-45. It is Reserved for reporting an access violation. Therefore, writes to bit 31 of the SCP_INTR_S_SET register are ignored.

Because the MHU registers all reside in the same 64Kbyte region, to avoid conflicts with Normal world software, avoid Secure access to any Non-secure MHU interrupt set, clear, and status registers.

Table 3-27 on page 3-42 shows a summary of the register map of the MHU.

3.5 System override registers

See *System override* on page 2-16.

This section describes:

- *Register summary*.
- *Register descriptions*.

3.5.1 Register summary

Table 3-23 shows the System Override registers in offset order from the base memory address.

UNDEFINED addresses are Reserved.

Table 3-23 System Override Registers summary

Offset	Name	Type	Width	Reset	Description
0x000	SEC_HDLCD	RW	32	0x0	*SEC_HDLCD Register*
0x004	GPR_0	RW	32	0x0	*GPR_0 Register* on page 3-41
0x008	GPR_1	RW	32	0x0	*GPR_1 Register* on page 3-41
0xFD0	Peripheral ID 4	RO	32	0x04	Peripheral ID 4
0xFE0	Peripheral ID 0	RO	32	0xAB	Peripheral ID 0
0xFE4	Peripheral ID 1	RO	32	0xB0	Peripheral ID 1
0xFE8	Peripheral ID 2	RO	32	0x0B	Peripheral ID 2
0xFEC	Peripheral ID 3	RO	32	0x00	Peripheral ID 3
0xFF0	Component ID 0	RO	32	0x0D	Component ID 0
0xFF4	Component ID 1	RO	32	0xF0	Component ID 1
0xFF8	Component ID 2	RO	32	0x05	Component ID 2
0xFFC	Component ID 3	RO	32	0xB1	Component ID 3

3.5.2 Register descriptions

This section describes the System Override registers. Table 3-23 provides cross references to individual registers.

SEC_HDLCD Register

The SEC_HDLCD Register characteristics are:

Purpose Sets the HDLCD Controller security states.

Usage constraints There are no usage constraints.

Attributes See Table 3-23.

Table 3-24 shows the bit assignments.

Table 3-24 SEC_HDLCD Register

Bits	Name	Description
[31:2]	-	Reserved.
[1]	HDLCD1_SEC	Selects the security state of the HDLCD controller 1: 0 Secure. 1 Non-secure.
[0]	HDLCD0_SEC	Selects the security state of the HDLCD controller 0: 0 Secure. 1 Non-secure.

GPR_0 Register

The GPR_0 Register characteristics are:

Purpose General purpose register.

Usage constraints There are no usage constraints.

Attributes See Table 3-23 on page 3-40.

Table 3-25 shows the bit assignments.

Table 3-25 GPR_0 Register

Bits	Name	Description
[31:0]	**GPR_0**	General Purpose Register 0.

GPR_1 Register

The GPR_1 Register characteristics are:

Purpose General purpose register.

Usage constraints There are no usage constraints.

Attributes See Table 3-23 on page 3-40.

Table 3-26 shows the bit assignments.

Table 3-26 GPR_1 Register

Bits	Name	Description
[31:0]	**GPR_1**	General Purpose Register 1.

3.6 MHU Registers

This section describes:

- *Register summary*.
- *Register descriptions* on page 3-43.

3.6.1 Register summary

Addresses are relative to the base address of the MHU that the ADP memory map defines.

Table 3-27 shows the MHU registers in offset order from the base memory address.

UNDEFINED addresses are Reserved.

Table 3-27 MHU Register summary

Offset	Name	Secure only	Type	Reset	Description
0x000	SCP_INTR_L_STAT	N	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x008	SCP_INTR_L_SET	N	WO	-	*<n>_SET Registers* on page 3-44
0x010	SCP_INTR_L_CLEAR	N	WO	-	*<n>_CLEAR Registers* on page 3-44
0x020	SCP_INTR_H_STAT	N	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x028	SCP_INTR_H_SET	N	WO	-	*<n>_SET Registers* on page 3-44
0x030	SCP_INTR_H_CLEAR	N	WO	-	*<n>_CLEAR Registers* on page 3-44
0x100	CPU_INTR_L_STAT	N	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x108	CPU_INTR_L_SET	N	WO	-	*<n>_SET Registers* on page 3-44
0x110	CPU_INTR_L_CLEAR	N	WO	-	*<n>_CLEAR Registers* on page 3-44
0x120	CPU_INTR_H_STAT	N	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x128	CPU_INTR_H_SET	N	WO	-	*<n>_SET Registers* on page 3-44
0x130	CPU_INTR_H_CLEAR	N	WO	-	*<n>_CLEAR Registers* on page 3-44
0x200	SCP_INTR_S_STAT	Y	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x208	SCP_INTR_S_SET	Y	WO	-	*<n>_SET Registers* on page 3-44
0x210	SCP_INTR_S_CLEAR	Y	WO	-	*<n>_CLEAR Registers* on page 3-44
0x300	CPU_INTR_S_STAT	Y	RO	0x0000_0000	*<n>_STAT Registers* on page 3-43
0x308	CPU_INTR_S_SET	Y	WO	-	*<n>_SET Registers* on page 3-44.
0x310	CPU_INTR_S_CLEAR	Y	WO	-	*<n>_CLEAR Registers* on page 3-44
0x400	MHU_SCFG	Y	RW	0x0000_0000	*MHU_SCFG Register* on page 3-45
0xFD0	MHU_PID_4	N	RO	0x00000004	*MHU_PID_4 Register* on page 3-46
0xFE0	MHU_PID_0	N	RO	0x00000098	*MHU_PID_0 Register* on page 3-47
0xFE4	MHU_PID_1	N	RO	0x000000B0	*MHU_PID_1 Register* on page 3-46
0xFE8	MHU_PID_2	N	RO	0x0000001B	*MHU_PID_2 Register* on page 3-46
0xFEC	MHU_PID_3	N	RO	0x00000000	*MHU_PID_3 Register* on page 3-47

Table 3-27 MHU Register summary (continued)

Offset	Name	Secure only	Type	Reset	Description
0xFF0	MHU_COMP_ID0	N	RO	0x0000000D	*MHU_COMP_ID0 Register* on page 3-48
0xFF4	MHU_COMP_ID1	N	RO	0x000000F0	*MHU_COMP_ID1 Register* on page 3-48
0xFF8	MHU_COMP_ID2	N	RO	0x00000005	*MHU_COMP_ID2 Register* on page 3-49
0xFFC	MHU_COMP_ID3	N	RO	0x000000B1	*MHU_COMP_ID3 Register* on page 3-49

For more information on the MHU, see *Message Handling Unit (MHU)* on page 3-38.

Because each interrupt line contains three similar control and status registers that are associated with it, the following sections describe these three registers together, where one of the following replaces <n> for its associated interrupt line:

SCP_INTR_L Low priority Non-secure interrupt from the SCP to the application processor.

SCP_INTR_H High priority Non-secure interrupt from the SCP to the application processor.

CPU_INTR_L Low priority Non-secure interrupt from the application processor to the SCP.

CPU_INTR_H High priority Non-secure interrupt from the application processor to the SCP.

CPU_INTR_S Secure interrupt from the application processor to the SCP.

SCP_INTR_S Secure interrupt from the SCP to the application processor.

3.6.2 Register descriptions

This section describes the MHU registers. Table 3-27 on page 3-42 provides cross references to individual registers.

<n>_STAT Registers

The <n>_STAT Register characteristics are:

Purpose Low priority, Non-secure interrupt status registers, from the SCP to the application processor. Each one shows the status of all 32 bits of the register that drives the associated interrupt line. If any of the bits are set in the register, then the associated interrupt line is asserted.

Note

For the SCP_INTR_S_STAT register that controls the Secure interrupt to the application processor, bit 31 also denotes an access violation when the SVIEN field in the MHU_SCFG register is set HIGH.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

Figure 3-23 on page 3-44 shows the bit assignments.

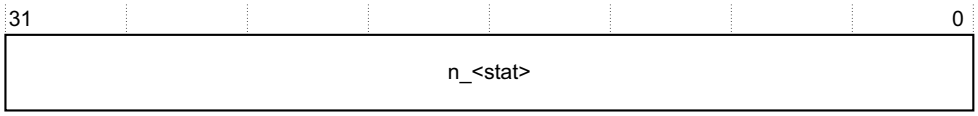


Figure 3-23 <n>_STAT Register bit assignments

Table 3-28 shows the bit assignments.

Table 3-28 <n>_STAT Register bit assignments

Bits	Name	Description
[31:0]	<n>_STAT	If any of these bits are set, then the associated interrupt line is asserted. Bits are set by writing a 1 to the associated SET register, and are cleared by writing a 0 to the associated CLEAR register.

<n>_SET Registers

The <n>_SET Register characteristics are:

Purpose Low priority Non-secure interrupt set register, from the SCP to the application processor. It sets bits in the associated <n>_STAT register. When read, it always returns `0x0000_0000`.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

Figure 3-24 shows the bit assignments.

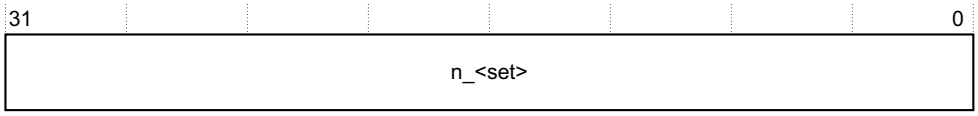


Figure 3-24 <n>_SET Register bit assignments

Table 3-29 shows the bit assignments.

Table 3-29 <n>_SET Register bit assignments

Bits	Name	Description
[31:0]	<n>_SET	Setting any bit in this 32-bit field HIGH sets the corresponding bit in the associated <n>_STAT register. You can set this field as follows: `0` No effect. 1 Writing a 1 to any bit in this 32-bit field clears the corresponding bit in the associated <n>_STAT register.

<n>_CLEAR Registers

The <n>_CLEAR Register characteristics are:

Purpose Low priority Non-secure interrupt clear register, from the SCP to the application processor. It clears bits in the associated <n>_STAT register. When read, it always returns `0x0000_0000`.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

Figure 3-25 shows the bit assignments.

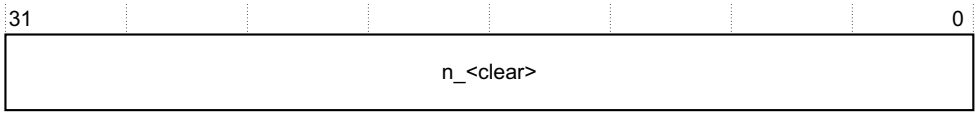


Figure 3-25 <n>_CLEAR Register bit assignments

Table 3-30 shows the bit assignments.

Table 3-30 <n>_CLEAR Register bit assignments

Bits	Name	Description
[31:0]	<n>_CLEAR	You can set this field as follows: 0 No effect. 1 Writing a 1 to any bit in this 32-bit field clears the corresponding bit in the associated <n>_STAT register.

MHU_SCFG Register

The MHU_SCFG Register characteristics are:

Purpose MHU Secure configuration register that enables or disables the raising of interrupts when access security violations occur. An access security violation is when a Non-secure access attempts to access a Secure only access register. When enabled, and an access security violation occurs, bit 31 of the INTR_S_STAT register, that controls the MHU_SCP_ INTR_S Secure interrupt to the application processor, is set HIGH.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

Figure 3-26 shows the bit assignments.

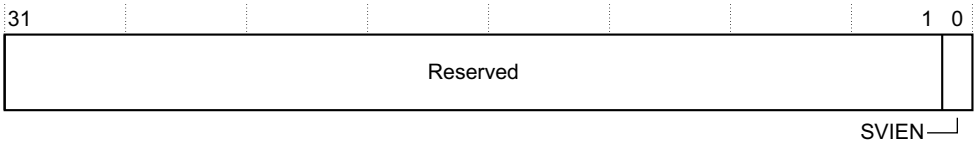


Figure 3-26 MHU_SCFG Register bit assignments

Table 3-31 shows the bit assignments.

Table 3-31 MHU_SCFG register bit assignments

Bits	Name	Description
[31:1]	-	Reserved. RAZ.
[0]	SVIEN	Secure Violation Interrupt Enable. You can set this bit as follows: 0 Disables interrupts. 1 Enables an interrupt to be raised by setting bit 31 of the INTR_S_STAT register when a security violation occurs.

PID Registers

The PID Registers characteristics are:

Purpose Provides information to identify the component.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

MHU_PID_4 Register

Figure 3-27 shows the bit assignments.

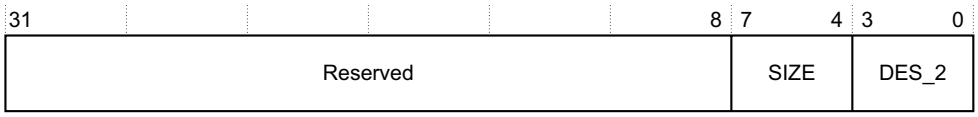


Figure 3-27 MHU_PID_4 Register bit assignments

Table 3-32 shows the bit assignments.

Table 3-32 MHU_PID_4 register

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	SIZE	Indicates the $\log_2$ of the number of 4KB blocks that the interface occupies. This field is set to `0x0`.
[3:0]	DES_2	JEP106 continuation code that identifies the designer. This field is set to `0x4` for ARM.

MHU_PID_1 Register

Figure 3-28 shows the bit assignments.

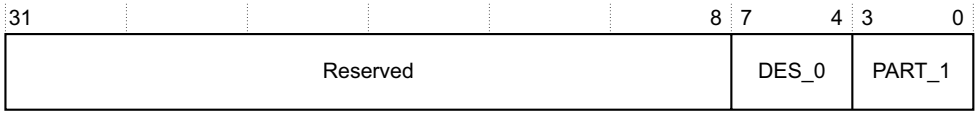


Figure 3-28 MHU_PID_1 Register bit assignments

Table 3-33 shows the bit assignments.

Table 3-33 MHU_PID_1 register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	DES_0	Bits [3:0] of the JEP Identity. This field is set to `0xB` for ARM.
[3:0]	PART_1	Bits [11:8] of the part number. This field is set to `0x0`.

MHU_PID_2 Register

Figure 3-29 on page 3-47 shows the bit assignments.

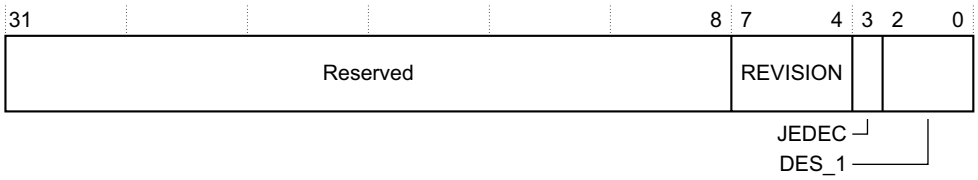


Figure 3-29 MHU_PID_2 Register bit assignments

Table 3-34 shows the bit assignments.

Table 3-34 MHU_PID_2 register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:4]	REVISION	This field is set to `0x2` for r2p0.
[3]	JEDEC	This bit is set to `0x1`
[2:0]	DES_1	Bits [6:4] of the Designer field. This field is set to `0x3` for ARM.

MHU_PID_3 Register

Figure 3-30 shows the bit assignments.

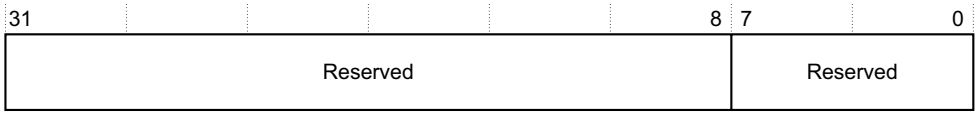


Figure 3-30 MHU_PID_3 Register bit assignments

Table 3-35 shows the bit assignments.

Table 3-35 MHU_PID_3 register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	-	Reserved. RAZ.

MHU_PID_0 Register

Figure 3-31 shows the bit assignments.

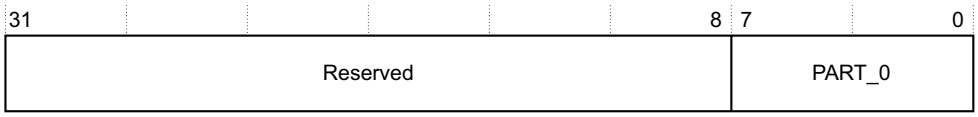


Figure 3-31 MHU_PID_0 Register bit assignments

Table 3-36 shows the bit assignments.

Table 3-36 MHU_PID_0 register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	PART_0	Bits [7:0] of the part number. This field is set to `0x98`.

COMP_ID Registers

The COMP_ID Registers characteristics are:

Purpose Provides information to identify the component.

Usage constraints There are no usage constraints.

Attributes See Table 3-27 on page 3-42.

MHU_COMP_ID0 Register

Figure 3-32 shows the bit assignments.

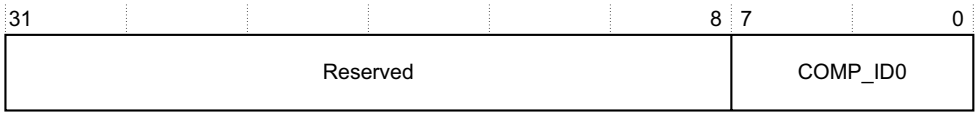


Figure 3-32 MHU_COMP_ID0 Register bit assignments

Table 3-37 shows the bit assignments.

Table 3-37 MHU_COMP_ID0 register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID0	This field reads as `0x0D`.

MHU_COMP_ID1 Register

Figure 3-33 shows the bit assignments.

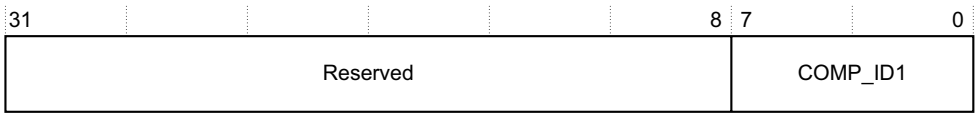


Figure 3-33 MHU_COMP_ID1 Register bit assignments

Table 3-38 shows the bit assignments.

Table 3-38 MHU_COMP_ID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID1	This field reads as `0xF0`.

MHU_COMP_ID2 Register

Figure 3-34 shows the bit assignments.

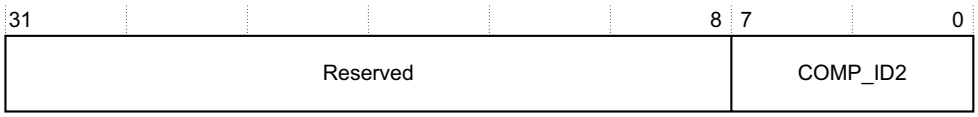


Figure 3-34 MHU_COMP_ID2 Register bit assignments

Table 3-39 shows the bit assignments.

Table 3-39 MHU_COMP_ID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID2	This field reads as `0x05`.

MHU_COMP_ID3 Register

Figure 3-35 shows the bit assignments.

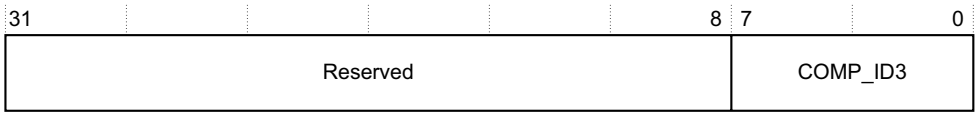


Figure 3-35 MHU_COMP_ID3 Register bit assignments

Table 3-40 shows the bit assignments.

Table 3-40 MHU_COMP_ID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ.
[7:0]	COMP_ID3	This field reads as `0xB1`

3.7 SoC Interconnect NIC-400 Registers

In the SoC Interconnect NIC-400, each interface that contains registers has its own sub block within the SoC Interconnect NIC-400 GPV block that *Application memory map summary* on page 3-21 defines. See Figure 2-1 on page 2-2 and Figure 3-9 on page 3-19.

This section provides a summary of the registers present in the SoC Interconnect NIC-400 *Global Programmers View* (GPV) block.

This section describes:

- *SoC Interconnect NIC-400 sub block memory map*.
- *Register summary* on page 3-51.

3.7.1 SoC Interconnect NIC-400 sub block memory map

Table 3-41 shows the SoC Interconnect NIC-400 sub blocks in offset order from the block base memory address.

UNDEFINED addresses are Reserved.

Table 3-41 SoC Interconnect NIC-400 sub block memory map

Address range		Description
0x0	0xFFF	*Address Region Control Registers* on page 3-51
0x1000	0x1FFF	*ID Registers* on page 3-51
Master interfaces		
0x2000	0x2FFF	*USB_EHCI sub block registers* on page 3-52
0x3000	0x3FFF	*TLX_MST sub block registers* on page 3-52
0x4000	0x4FFF	*USB_OHCI sub block registers* on page 3-53
0x5000	0x5FFF	*PL354_SMC sub block registers* on page 3-53
0x6000	0x6FFF	*APB4_BRIDGE sub block registers* on page 3-53
0x7000	0x7FFF	*BOOTSEC_BRIDGE sub block registers* on page 3-54
0x8000	0x8FFF	*SECURE_BRIDGE sub block registers* on page 3-54
0xB000	0xBFFF	*CSS_SC sub block registers* on page 3-55
0x42000	0x42FFF	*CSS_M sub block registers* on page 3-55
Slave interfaces		
0x44000	0x44FFF	*HDLCD_0 sub block registers* on page 3-55
0x45000	0x45FFF	*HDLCD_1 sub block registers* on page 3-57
0x46000	0x46FFF	*DMA330_DMA sub block registers* on page 3-58
0x47000	0x47FFF	*TLX_SLV sub block registers* on page 3-59
0x48000	0x48FFF	*USB_MST sub block registers* on page 3-60
0xC2000	0xC2FFF	*IB0 sub block registers* on page 3-61

3.7.2 Register summary

This section provides a summary of the registers that are located in each sub block of the SoC interconnect NIC-400. Table 3-41 on page 3-50 provides cross references to the sub block base addresses.

Address Region Control Registers

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-42 shows the Address Region Control Registers.

Table 3-42 Address Region Control Registers

Offset	Name	Type	Width	Description
0x008	security0	WO	32	Security setting for transactions through usb_ehci master interface.
0x00C	security1	WO	32	Security setting for transactions through tlx_mst master interface.
0x010	security2	WO	32	Security setting for transactions through usb_ohci master interface.
0x01C	security5	WO	32	Security settings for transactions passing through the identified boot-Secure APB master interfaces: **[0]** surge_det. **[1]** pvt_a53. **[2]** pvt_soc. **[3]** pvt_mali. **[4]** pvt_a72. **[5]** pvt_std_cell. **[6]** dfi_phy0_cfg. **[7]** dfi_phy1_cfg. **[8]** i2s. **[9]** hdlcd0_cfg. **[10]** hdlcd1_cfg. **[11]** uart0. **[12]** uart1. **[13]** i2c. **[14]** pl354_smc_cfg. **[15]** pl330_dma_nsec. Bit values: 0 Secure. 1 Non-secure.

ID Registers

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-43 shows the ID Registers.

Table 3-43 ID Registers

Offset	Name	Type	Width	Description
0xFD0	Peripheral ID4	RO	32	4KB count, JEP106 continuation code.
0xFD4	Peripheral ID5	RO	32	Reserved.
0xFD8	Peripheral ID6	RO	32	Reserved.
0xFDC	Peripheral ID7	RO	32	Reserved.
0xFE0	Peripheral ID0	RO	32	Part number[7:0].
0xFE4	Peripheral ID1	RO	32	JEP106[3:0], part number[11:8].
0xFE8	Peripheral ID2	RO	32	Revision, JEP106 code flag, JEP106[6:4].
0xFEC	Peripheral ID3	RO	32	User-configurable in AMBA Designer, set to 0x0.
0xFF0	Component ID0	RO	32	Preamble.
0xFF4	Component ID1	RO	32	Generic IP component class, preamble.
0xFF8	Component ID2	RO	32	Preamble.
0xFFC	Component ID3	RO	32	Preamble.

USB_EHCI sub block registers

Registers for NIC-400 master interface to the USB EHCI slave interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-44 shows the USB_EHCI master interface Registers.

Table 3-44 USB_EHCI sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.
0x044	ahb_cntl	RW	32	AHB interface control bits.

TLX_MST sub block registers

Registers for NIC-400 master interface to the slave interface on the *Thin Links* (TLX) master port.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-45 shows the TLX_MST master interface Registers.

Table 3-45 TLX_MST sub block registers

Offset	Name	Type	Width	Description
0x108	fn_mod	RW	32	Issuing functionality modification register.

USB_OHCI sub block registers

Registers for NIC-400 master interface to the USB OHCI slave interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-46 shows the USB_OHCI master interface Registers.

Table 3-46 USB_OHCI sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.
0x044	ahb_cntl	RW	32	AHB interface control bits.

PL354_SMC sub block registers

Registers for NIC-400 master interface to the PL354 SMC slave interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-47 shows the PL354_SMC sub block registers.

Table 3-47 PL354_SMC sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.
0x108	fn_mod	RW	32	Issuing functionality modification register.

APB4_BRIDGE sub block registers

Registers for the internal NIC-400 bridge between the AXI backplane and the master interfaces to the following APB4 slave interfaces:

- HDLCD 0 SMMU configuration.
- HDLCD 1 SMMU configuration.
- USB SMMU configuration.
- DMA-330 configuration.
- PCIe configuration.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-48 shows the APB4_BRIDGE sub block registers.

Table 3-48 APB4_BRIDGE sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.

BOOTSEC_BRIDGE sub block registers

Registers for the internal NIC-400 bridge between the AXI backplane and the master interfaces to the following slave interfaces:

- DMA-330 DMA Non-secure interface.
- PL354 SMC configuration.
- Surge Detector.
- *Process, Voltage, and Temperature* (PVT) monitors.
- DFI PHY 0 configuration.
- DFI PHY 1 configuration.
- I^2S.
- HDLCD 0 configuration.
- HDLCD 1 configuration.
- UART 0.
- UART 1.
- I^2C.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-49 shows the BOOTSEC_BRIDGE sub block registers.

Table 3-49 BOOTSEC_BRIDGE sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.

SECURE_BRIDGE sub block registers

Registers for the internal NIC-400 bridge between the AXI backplane and the master interfaces to the following slave interfaces:

- Random Number Generator.
- Non-volatile counters.
- Keys and fuses.
- System Override Registers.

- Secure I2C interface.
- Secure DMA-330 DMA interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-50 shows the SECURE_BRIDGE sub block registers.

Table 3-50 SECURE_BRIDGE sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.

CSS_SC sub block registers

Registers for the NIC-400 master interface to the I/O coherent slave port on the compute subsystem.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-51 shows the CSS_SC sub block registers.

Table 3-51 CSS_SC sub block registers

Offset	Name	Type	Width	Description
0x108	fn_mod	RW	32	Issuing functionality modification register.

CSS_M sub block registers

Registers for the NIC-400 slave interface to the master extension port of the compute subsystem.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-52 shows the CSS_M sub block registers.

Table 3-52 CSS_M sub block registers

Offset	Name	Type	Width	Description
0x108	fn_mod	RW	32	Issuing functionality modification register.

HDLCD_0 sub block registers

Registers for the NIC-400 slave interface to the HDLCD Controller 0 master.

Registers not specified are Reserved. RAZ, WI.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers, except where indicated.

Table 3-53 shows the HDLCD_0 sub block registers.

Table 3-53 HDLCD_0 sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** **mode_ar_fc**. **[19:17]** Reserved. RAZ, do not modify. **[16]** **mode_aw_fc**. **[15:8]** Reserved. RAZ, do not modify. **[7]** **en_awar_ot**. **[6]** **en_ar_ot**. **[5]** **en_aw_ot**. **[4]** **en_ar_fc**. **[3]** **en_aw_fc**. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register: **[31:30]** Reserved. RAZ, do not modify. **[29:24]** **ar_max_oti**. **[23:16]** **ar_max_otf**. **[15:14]** Reserved. RAZ, do not modify. **[13:8]** **aw_max_oti**. **[7:0]** **aw_max_otf**.
0x114	max_comb_ot[a]	RW	32	Maximum number of combined transactions register.
0x118	aw_p[a]	RW	32	AW channel peak rate Register.
0x11C	aw_b[a]	RW	32	AW channel burstiness allowance Register.
0x120	aw_r[a]	RW	32	AW channel average rate Register.
0x124	ar_p[a]	RW	32	AR channel peak rate Register.
0x128	ar_b[a]	RW	32	AR channel burstiness allowance Register.
0x12C	ar_r[a]	RW	32	AR channel average rate Register.
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for a description of this register.

HDLCD_1 sub block registers

Registers for the NIC-400 slave interface to the HDLCD Controller 1 master.

Registers not specified are Reserved. RAZ, WI.

Except where stated otherwise, see the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-54 shows the HDLCD_1 sub block registers.

Table 3-54 HDLCD_1 sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** **mode_ar_fc**. **[19:17]** Reserved. RAZ, do not modify. **[16]** **mode_aw_fc**. **[15:8]** Reserved. RAZ, do not modify. **[7]** **en_awar_ot**. **[6]** **en_ar_ot**. **[5]** **en_aw_ot**. **[4]** **en_ar_fc**. **[3]** **en_aw_fc**. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register: **[31:30]** Reserved. RAZ, do not modify. **[29:24]** **ar_max_oti**. **[23:16]** **ar_max_otf**. **[15:14]** Reserved. RAZ, do not modify. **[13:8]** **aw_max_oti**. **[7:0]** **aw_max_otf**.
0x114	max_comb_ot	RW	32	Maximum number of combined transactions register.
0x118	aw_p[a]	RW	32	AW channel peak rate Register.
0x11C	aw_b[a]	RW	32	AW channel burstiness allowance Register.
0x120	aw_r[a]	RW	32	AW channel average rate Register.
0x124	ar_p[a]	RW	32	AR channel peak rate Register.
0x128	ar_b[a]	RW	32	AR channel burstiness allowance Register.
0x12C	ar_r[a]	RW	32	AR channel average rate Register.

Table 3-54 HDLCD_1 sub block registers (continued)

Offset	Name	Type	Width	Description
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

DMA330_DMA sub block registers

Registers for the NIC-400 slave interface to the DMA-330 DMA master.

Registers not specified are Reserved. RAZ, WI.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-55 shows the DMA330_DMA sub block registers.

Table 3-55 DMA330_DMA sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** **mode_ar_fc**. **[19:17]** Reserved. RAZ, do not modify. **[16]** **mode_aw_fc**. **[15:8]** Reserved. RAZ, do not modify. **[7]** **en_awar_ot**. **[6]** **en_ar_ot**. **[5]** **en_aw_ot**. **[4]** **en_ar_fc**. **[3]** **en_aw_fc**. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register: **[31:30]** Reserved. RAZ, do not modify. **[29:24]** **ar_max_oti**. **[23:16]** **ar_max_otf**. **[15:14]** Reserved. RAZ, do not modify. **[13:8]** **aw_max_oti**. **[7:0]** **aw_max_otf**.
0x114	max_comb_ot[a]	RW	32	Maximum number of combined transactions register.

Table 3-55 DMA330_DMA sub block registers (continued)

Offset	Name	Type	Width	Description
0x118	aw_p[a]	RW	32	AW channel peak rate Register.
0x11C	aw_b[a]	RW	32	AW channel burstiness allowance Register.
0x120	aw_r[a]	RW	32	AW channel average rate Register.
0x124	ar_p[a]	RW	32	AR channel peak rate Register.
0x128	ar_b[a]	RW	32	AR channel burstiness allowance Register.
0x12C	ar_r[a]	RW	32	AR channel average rate Register.
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

TLX_SLV sub block registers

Registers for the NIC-400 slave interface to the master interface of the Thin Links slave port.

Except where stated otherwise, see the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-56 shows the TLX_SLV sub block registers.

Table 3-56 TLX_SLV sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** **mode_ar_fc**. **[19:17]** Reserved. RAZ, do not modify. **[16]** **mode_aw_fc**. **[15:8]** Reserved. RAZ, do not modify. **[7]** **en_awar_ot**. **[6]** **en_ar_ot**. **[5]** **en_aw_ot**. **[4]** **en_ar_fc**. **[3]** **en_aw_fc**. **[2:0]** Reserved. RAZ, do not modify.

Table 3-56 TLX_SLV sub block registers (continued)

Offset	Name	Type	Width	Description
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register: **[31:30]** Reserved. RAZ, do not modify. **[29:24]** **ar_max_oti**. **[23:16]** **ar_max_otf**. **[15:14]** Reserved. RAZ, do not modify. **[13:8]** **aw_max_oti**. **[7:0]** **aw_max_otf**.
0x114	max_comb_ot	RW	32	Maximum number of combined transactions register.
0x118	aw_p[a]	RW	32	AW channel peak rate Register.
0x11C	aw_b[a]	RW	32	AW channel burstiness allowance Register.
0x120	aw_r[a]	RW	32	AW channel average rate Register.
0x124	ar_p[a]	RW	32	AR channel peak rate Register.
0x128	ar_b[a]	RW	32	AR channel burstiness allowance Register.
0x12C	ar_r[a]	RW	32	AR channel average rate Register.
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

USB_MST sub block registers

Registers for the NIC-400 slave interface to the USB master interface.

Registers not specified are Reserved. RAZ, WI.

Except where stated otherwise, see the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-57 shows the USB_MST sub block registers.

Table 3-57 USB_MST sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.

Table 3-57 USB_MST sub block registers (continued)

Offset	Name	Type	Width	Description
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** **mode_ar_fc**. **[19:17]** Reserved. RAZ, do not modify. **[16]** **mode_aw_fc**. **[15:8]** Reserved. RAZ, do not modify. **[7]** **en_awar_ot**. **[6]** **en_ar_ot**. **[5]** **en_aw_ot**. **[4]** **en_ar_fc**. **[3]** **en_aw_fc**. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register: **[31:30]** Reserved. RAZ, do not modify. **[29:24]** **ar_max_oti**. **[23:16]** **ar_max_otf**. **[15:14]** Reserved. RAZ, do not modify. **[13:8]** **aw_max_oti**. **[7:0]** **aw_max_otf**.
0x114	max_comb_ot[a]	RW	32	Maximum number of combined transactions register.
0x118	aw_p[a]	RW	32	AW channel peak rate Register.
0x11C	aw_b[a]	RW	32	AW channel burstiness allowance Register.
0x120	aw_r[a]	RW	32	AW channel average rate Register.
0x124	ar_p[a]	RW	32	AR channel peak rate Register.
0x128	ar_b[a]	RW	32	AR channel burstiness allowance Register.
0x12C	ar_r[a]	RW	32	AR channel average rate Register.
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

IB0 sub block registers

Registers for the NIC-400 internal downsizing bridge between the compute subsystem master port slave interface and the main switch for the master interfaces for the attached SoC level peripherals.

Registers not specified are Reserved. RAZ, WI.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-58 shows the IB0 Registers.

Table 3-58 IB0 sub block registers

Offset	Name	Type	Width	Description
0x008	fn_mod_bm_iss	RW	32	Bus matrix issuing functionality modification register.
0x024	fn_mod2	RW	32	Bypass merge.
0x02C	fn_mod_lb	RW	32	Long burst functionality modification register.
0x108	fn_mod	RW	32	Issuing functionality modification register.

3.8 Compute subsystem NIC-400 Registers

In the compute subsystem NIC-400, each interface that contains registers has its own sub block within the compute subsystem NIC-400 GPV block that *Application memory map summary* on page 3-21 defines. See Figure 2-1 on page 2-2 and Figure 3-6 on page 3-16.

This section provides a summary of the registers present in the compute subsystem NIC-400 GPV block.

This section describes:

- *Compute subsystem NIC-400 Sub block memory map*.
- *Register summary*.

3.8.1 Compute subsystem NIC-400 Sub block memory map

Table 3-59 shows the compute subsystem NIC-400 sub blocks in offset order from the block base memory address.

Table 3-59 Compute subsystem NIC-400 sub block memory map

Address range		Description
0x0	0xFFF	*Address Region Control Registers*
0x1000	0x1FFF	*ID Registers* on page 3-64
0x2000	0x41FFF	Reserved
0x42000	0x42FFF	*CCISLAVE sub block registers* on page 3-64
0x43000	0x43FFF	*CUST_EXT_SLV sub block registers* on page 3-65
0x44000	0x44FFF	*SCP sub block registers* on page 3-65
0x45000	0x45FFF	*CS_DAP sub block registers* on page 3-66
0x46000	0x46FFF	*CS_ETR sub block registers* on page 3-66
0x47000	0xFFFFF	Reserved

3.8.2 Register summary

This section provides a summary of the registers in each sub block of the Compute subsystem NIC-400 registers. Table 3-59 provides cross references to sub block base addresses.

Address Region Control Registers

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-60 shows the address regions control registers.

Table 3-60 Address Region Control Registers

Offset	Name	Type	Width	Description
0x028	security8	WO	32	slave_bootsecure APB group security settings. Bit to APB master mapping: **[2]** GPU_CFG. **[1]** DMC_CFG. **[0]** CS_CFG. Bit values: 0 Secure. 1 Non-secure.

ID Registers

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-61 shows the ID registers.

Table 3-61 ID Registers

Offset	Name	Type	Width	Description
0xFD0	Peripheral ID4	RO	32	4KB count, JEP106 continuation code.
0xFD4	Peripheral ID5	RO	32	Reserved.
0xFD8	Peripheral ID6	RO	32	Reserved.
0xFDC	Peripheral ID7	RO	32	Reserved.
0xFE0	Peripheral ID0	RO	32	Part number[7:0].
0xFE4	Peripheral ID1	RO	32	JEP106[3:0], part number[11:8].
0xFE8	Peripheral ID2	RO	32	Revision, JEP106 code flag, JEP106[6:4].
0xFEC	Peripheral ID3	RO	32	Configurable in AMBA Designer. Set to 0x0.
0xFF0	Component ID0	RO	32	Preamble.
0xFF4	Component ID1	RO	32	Generic IP component class, preamble.
0xFF8	Component ID2	RO	32	Preamble.
0xFFC	Component ID3	RO	32	Preamble.

CCISLAVE sub block registers

Registers for the NIC-400 slave interface that is connected to CCI-400 Master Interface 1.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-62 shows the CCISLAVE sub block registers.

Table 3-62 CCISLAVE sub block registers

Offset	Name	Type	Width	Description
0x108	fn_mod	RW	32	Issuing functionality modification register.

CUST_EXT_SLV sub block registers

Registers for the NIC-400 slave interface that connects to the SoC Interconnect NIC-400 master interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-63 shows the CUST_EXT_SLV sub block registers.

Table 3-63 CUST_EXT_SLV sub block registers

Offset	Name	Type	Width	Description
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** mode_ar_fc. **[19:17]** Reserved. RAZ, do not modify. **[16]** mode_aw_fc. **[15:8]** Reserved. RAZ, do not modify. **[7]** en_awar_ot. **[6]** en_ar_ot. **[5]** en_aw_ot. **[4]** en_ar_fc. **[3]** en_aw_fc. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register.
0x114	max_comb_ot[a]	RW	32	Maximum number of combined transactions register.
0x118 – 0x12C	-	-	-	Reserved. RAZ, do not modify
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

SCP sub block registers

Registers for the NIC-400 slave interface that connects to the SCP master interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-64 shows the SCP sub block registers.

Table 3-64 SCP sub block registers

Offset	Name	Type	Width	Description
0x024	fn_mod2	RW	32	Bypass merge.
0x028 – 0x0FC	-	-	-	Reserved. RAZ, do not modify.
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.

CS_DAP sub block registers

Registers for the NIC-400 slave interface that connects to the CoreSight Subsystem DAP master interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-65 shows the CS_DAP sub block registers.

Table 3-65 CS_DAP sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.

CS_ETR sub block registers

Registers for the NIC-400 slave interface that connects to the CoreSight Subsystem ETR master interface.

UNDEFINED addresses are Reserved.

See the *ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual* for descriptions of these registers.

Table 3-66 on page 3-67 shows the CS_ETR sub block registers.

Table 3-66 CS_ETR sub block registers

Offset	Name	Type	Width	Description
0x100	read_qos	RW	32	Read channel QoS value.
0x104	write_qos	RW	32	Write channel QoS value.
0x108	fn_mod	RW	32	Issuing functionality modification register.
0x10C	qos_cntl[a]	RW	32	QoS control register. Enable bits for all the QoS regulators: **[31:21]** Reserved. RAZ, do not modify. **[20]** mode_ar_fc. **[19:17]** Reserved. RAZ, do not modify. **[16]** mode_aw_fc. **[15:8]** Reserved. RAZ, do not modify. **[7]** en_awar_ot. **[6]** en_ar_ot. **[5]** en_aw_ot. **[4]** en_ar_fc. **[3]** en_aw_fc. **[2:0]** Reserved. RAZ, do not modify.
0x110	max_ot[a]	RW	32	Maximum number of outstanding transactions register:
0x114	max_comb_ot[a]	RW	32	Maximum number of combined transactions register. **[31:15]** Reserved. RAZ, do not modify. **[14:8]** awar_max_oti. **[7:0]** awar_max_otf.
0x118 – 0x12C	-	-	32	Reserved. RAZ, do not modify.
0x130	target_fc[a]	RW	32	Feedback controlled target register.
0x134	ki_fc[a]	RW	32	Feedback controlled scale register.
0x138	qos_range[a]	RW	32	QoS range register.
0x13C – 0xFFC	-	-	-	Reserved. RAZ, WI.

a. See the *ARM® CoreLink™ QoS-400 Network Interconnect Advanced Quality of Service Supplement to ARM® CoreLink™ NIC-400 Network Interconnect Technical Reference Manual for a description of this register.*

3.9 HDLCD Registers

This section describes the programmers model. It contains the following subsections:

- *Register summary*.
- *Register descriptions* on page 3-69.

3.9.1 Register summary

Table 3-67 shows the registers in offset order from the base memory address.

Table 3-67 Register summary

Offset	Name	Type	Reset	Width	Description
0x0000	VERSION	RO	VERSION	32	*Version Register* on page 3-69
0x0010	INT_RAWSTAT	RW	0x0	32	*Interrupt Raw Status Register* on page 3-69
0x0014	INT_CLEAR	WO	N/A	32	*Interrupt Clear Register* on page 3-70
0x0018	INT_MASK	RW	0x0	32	*Interrupt Mask Register* on page 3-71
0x001C	INT_STATUS	RO	0x0	32	*Interrupt Status Register* on page 3-72
0x0100	FB_BASE	RW	0x0	32	*Frame Buffer Base Address Register* on page 3-73
0x0104	FB_LINE_LENGTH	RW	0x0	32	*Frame Buffer Line Length Register* on page 3-73
0x0108	FB_LINE_COUNT	RW	0x0	32	*Frame Buffer Line Count Register* on page 3-74
0x010C	FB_LINE_PITCH	RW	0x0	32	*Frame Buffer Line Pitch Register* on page 3-75
0x0110	BUS_OPTIONS	RW	0x408	32	*Bus Options Register* on page 3-75
0x0200	V_SYNC	RW	0x0	32	*Vertical Synch Width Register* on page 3-76
0x0204	V_BACK_PORCH	RW	0x0	32	*Vertical Back Porch Width Register* on page 3-77
0x0208	V_DATA	RW	0x0	32	*Vertical Data Width Register* on page 3-77
0x020C	V_FRONT_PORCH	RW	0x0	32	*Vertical Front Porch Width Register* on page 3-78
0x0210	H_SYNC	RW	0x0	32	*Horizontal Synch Width Register* on page 3-79
0x0214	H_BACK_PORCH	RW	0x0	32	*Horizontal Back Porch Width Register* on page 3-79
0x0218	H_DATA	RW	0x0	32	*Horizontal Data Width Register* on page 3-80
0x021C	H_FRONT_PORCH	RW	0x0	32	*Horizontal Front Porch Width Register* on page 3-80
0x0220	POLARITIES	RW	0x0	32	*Polarities Register* on page 3-81
0x0230	COMMAND	RW	0x0	32	*Command Register* on page 3-82
0x0240	PIXEL_FORMAT	RW	0x0	32	*Pixel Format Register* on page 3-83
0x0244	RED_SELECT	RW	0x0	32	*Color Select Registers* on page 3-84
0x0248	GREEN_SELECT	RW	0x0	32	*Color Select Registers* on page 3-84
0x024C	BLUE_SELECT	RW	0x0	32	*Color Select Registers* on page 3-84

3.9.2 Register descriptions

This section describes the HDLCD controller registers. Table 3-67 on page 3-68 provides cross references to individual registers.

Version Register

The VERSION Register characteristics are:

Purpose Contains a static version number for the LCD controller. Changes to the processor that affect registers and data structures increment the VERSION_MAJOR value and reset the VERSION_MINOR value.

Other changes that do not affect the binary compatibility only increment the VERSION_MINOR number.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-36 shows the bit assignments.

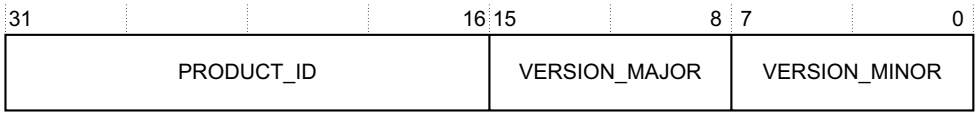


Figure 3-36 Version Register bit assignments

Table 3-68 shows the bit assignments.

Table 3-68 Version Register bit assignments

Bits	Name	Description
[31:16]	PRODUCT_ID	Product ID number `0x1CDC`.
[15:8]	VERSION_MAJOR	These bits provide the major product version information. For release r0p0, the value is `0x00`.
[7:0]	VERSION_MINOR	These bits provide the minor product version information. For release r0p0, the value is `0x00`.

Interrupt Raw Status Register

The INT_RAWSTAT Register characteristics are:

Purpose Shows the unmasked status of the interrupt sources.

Writing a 1 to the bit of an interrupt source forces this bit to be set and generates an interrupt if the corresponding bit in the *Interrupt Mask Register* on page 3-71 does not mask it.

Writing a 0 to the bit of an interrupt source has no effect.

Use the *Interrupt Clear Register* on page 3-70 to clear interrupts.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-37 shows the bit assignments.

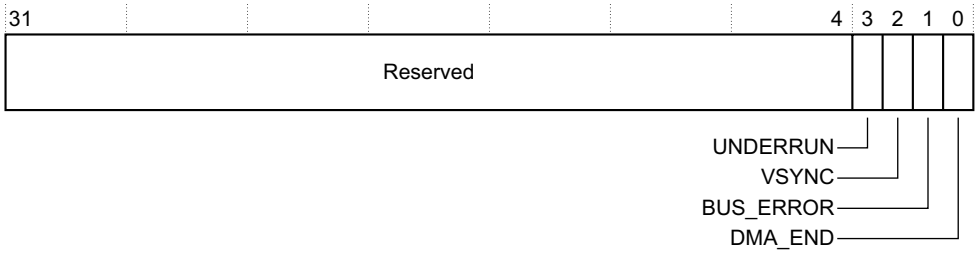


Figure 3-37 Interrupt Raw Status Register bit assignments

Table 3-69 shows the bit assignments.

Table 3-69 Interrupt Raw Status Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved. Write as zero, read UNDEFINED.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this trigger occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Clear Register

The INT_CLEAR Register characteristics are:

Purpose Clears interrupt sources. Writing a 1 to the bit of an asserted source clears the interrupt in the *Interrupt Raw Status Register* on page 3-69, and in the *Interrupt Status Register* on page 3-72 if it is not masked.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-38 on page 3-71 shows the bit assignments.

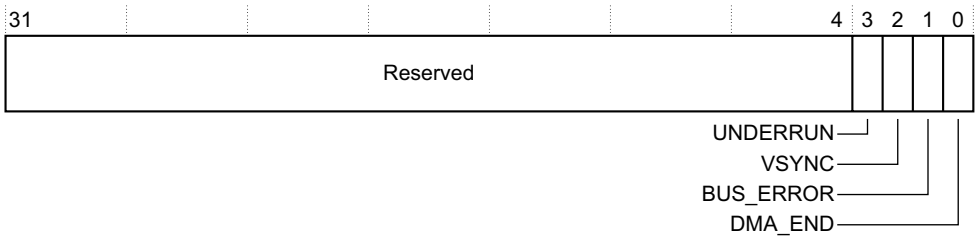


Figure 3-38 Interrupt Clear Register bit assignments

Table 3-70 shows the bit assignments.

Table 3-70 Interrupt Clear Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved. Write as zero.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this trigger occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Mask Register

The INT_MASK Register characteristics are:

Purpose If the corresponding mask bit is set to 1, this register contains the bit mask that enables an interrupt source.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-39 shows the bit assignments.

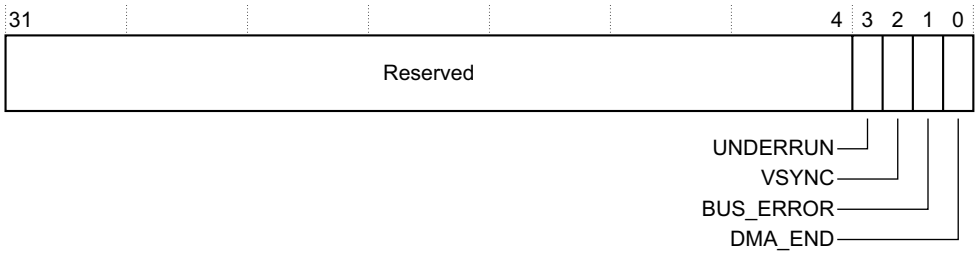


Figure 3-39 Interrupt Mask Register bit assignments

Table 3-71 shows the bit assignments.

Table 3-71 Interrupt Mask Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved. Write as zero, read UNDEFINED.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this trigger occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Interrupt Status Register

The INT_STATUS Register characteristics are:

Purpose This register is the *Interrupt Raw Status Register* on page 3-69 ANDed with the *Interrupt Mask Register* on page 3-71 and shows the active and masked interrupt sources. Bits selected by the *Interrupt Mask Register* on page 3-71 are active in the *Interrupt Status Register*. These bits show the status of the interrupt sources. Bits not selected by the Interrupt Mask are inactive. If any of the sources are asserted in the *Interrupt Status Register*, then the external **IRQ** line is asserted.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-40 shows the bit assignments.

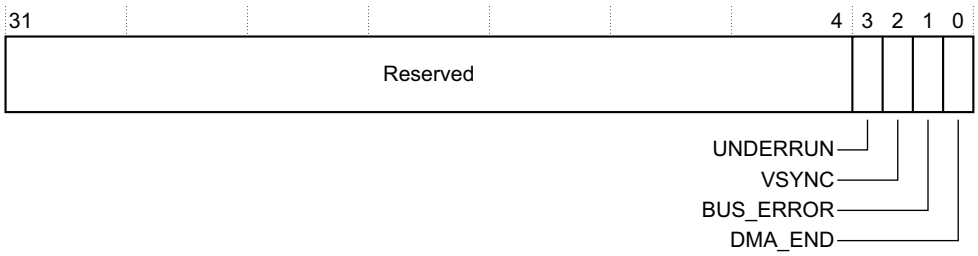


Figure 3-40 Interrupt Status Register bit assignments

Table 3-72 shows the bit assignments.

Table 3-72 Interrupt Status Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved, read UNDEFINED.
[3]	UNDERRUN	No data was available to display while **DATAEN** was active. This interrupt triggers if the controller does not have pixel data available to drive when **DATAEN** is active. When this trigger occurs, the controller drives the default color for the rest of the screen and attempts to display the next frame correctly.
[2]	VSYNC	Vertical sync is active. This interrupt triggers at the moment the **VSYNC** output goes active.
[1]	BUS_ERROR	The DMA module received a bus error while reading data. This interrupt triggers if any frame buffer read operation ever reports an error.
[0]	DMA_END	The DMA module has finished reading a frame. This interrupt triggers when the last piece of data for a frame has been read. The DMA immediately continues on the next frame, so this interrupt only ensures that the frame buffer for the previous frame is no longer required.

Frame Buffer Base Address Register

The FB_BASE Register characteristics are:

Purpose Contains the address of the first pixel of the first line in the frame buffer.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-41 shows the bit assignments.

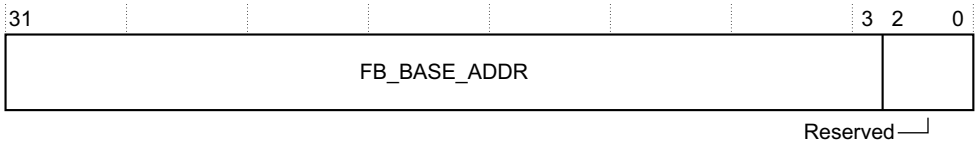


Figure 3-41 Frame Buffer Base Address Register bit assignments

Table 3-73 shows the bit assignments.

Table 3-73 Frame Buffer Base Address Register bit assignments

Bits	Name	Description
[31:3]	FB_BASE_ADDR	Frame buffer base address.
[2:0]	-	Reserved. Write as zero, read UNDEFINED.

Frame Buffer Line Length Register

The FB_LINE_LENGTH Register characteristics are:

Purpose Contains the length of each frame buffer line in bytes.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-42 shows the bit assignments.

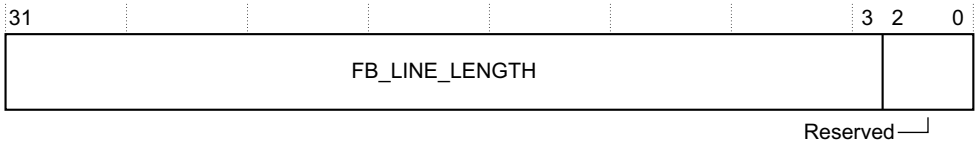


Figure 3-42 Frame Buffer Line Length Register bit assignments

Table 3-74 shows the bit assignments.

Table 3-74 Frame Buffer Line Length Register bit assignments

Bits	Name	Description
[31:3]	FB_LINE_LENGTH	Frame buffer line length.
[2:0]	-	Reserved. Write as zero, read UNDEFINED.

Frame Buffer Line Count Register

The FB_LINE_COUNT Register characteristics are:

Purpose Contains the number of lines to read from the frame buffer.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-43 shows the bit assignments.

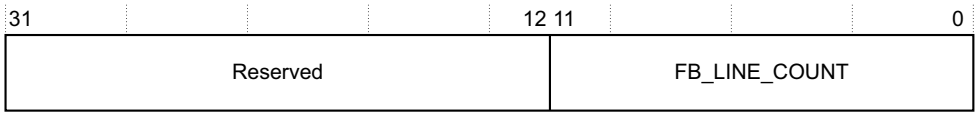


Figure 3-43 Frame Buffer Line Count Register bit assignments

Table 3-75 shows the bit assignments.

Table 3-75 Frame Buffer Line Count Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	FB_LINE_COUNT	Frame buffer line count.

Frame Buffer Line Pitch Register

The FB_LINE_PITCH Register characteristics are:

Purpose Contains the number of bytes between the start of one line in the frame buffer, and the start of the next line. This value is treated as a signed 2's complement number, enabling negative pitch if necessary.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-44 shows the bit assignments.

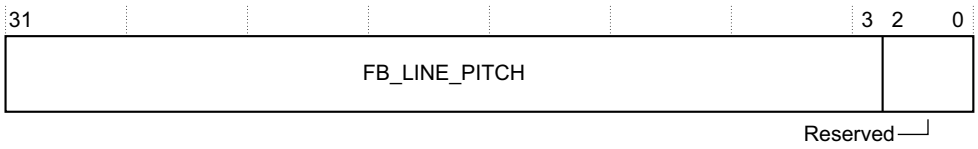


Figure 3-44 Frame Buffer Line Count Pitch bit assignments

Table 3-76 shows the bit assignments.

Table 3-76 Frame Buffer Line Pitch Register bit assignments

Bits	Name	Description
[31:3]	FB_LINE_PITCH	Frame buffer line pitch.
[2:0]	-	Reserved. Write as zero, read UNDEFINED.

Bus Options Register

The BUS_OPTIONS Register characteristics are:

Purpose Controls aspects of how the LCD controller accesses the bus. This value can be tuned to better match the characteristics of the memory controller and other units in the system.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-45 on page 3-76 shows the bit assignments.

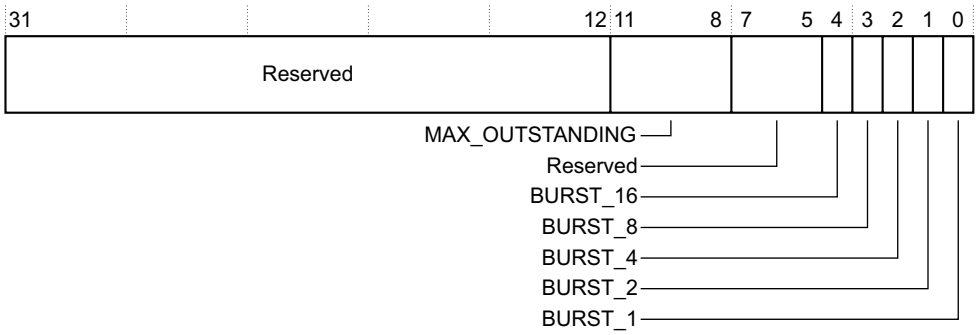


Figure 3-45 Bus Options Register bit assignments

Table 3-77 shows the bit assignments.

Table 3-77 Bus Options Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:8]	MAX_OUTSTANDING	Maximum number of outstanding requests the LCD controller is permitted to have on the bus at any time. **Caution** A value of zero disables all bus transfers
[7:5]	-	Reserved. Write as zero, read UNDEFINED.
[4]	BURST_16	Permit the use of 16-beat bursts.
[3]	BURST_8	Permit the use of 8-beat bursts.
[2]	BURST_4	Permit the use of 4-beat bursts.
[1]	BURST_2	Permit the use of 2-beat bursts.
[0]	BURST_1	Permit the use of 1-beat bursts.

Note

- If the scan line length does not end up at a multiple of the permitted burst lengths, the controller uses smaller bursts to read the remaining few pixels in each scan line. If no bursts are permitted, this mechanism also triggers, and has the same effect as permitting all bursts.
- Incorrectly configuring this register can degrade the performance of both the LCD controller and the rest of the system.

Vertical Synch Width Register

The V_SYNC Register characteristics are:

Purpose Contains the width of the vertical synch signal, which is counted in number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-46 shows the bit assignments.

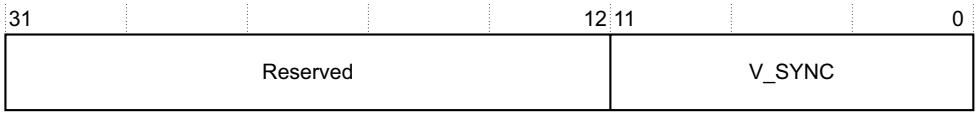


Figure 3-46 Vertical Synch Width Register bit assignments

Table 3-78 shows the bit assignments.

Table 3-78 Vertical Synch Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	V_SYNC	Vertical synch width -1.

Vertical Back Porch Width Register

The V_BACK_PORCH Register characteristics are:

Purpose Contains the width of the interval between the vertical sync and the first visible line, which is counted in number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-47 shows the bit assignments.

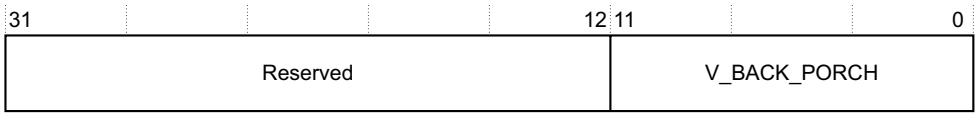


Figure 3-47 Vertical Back Porch Width Register bit assignments

Table 3-79 shows the bit assignments.

Table 3-79 Vertical Back Porch Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	V_BACK_PORCH	Vertical back porch width -1.

Vertical Data Width Register

The V_DATA Register characteristics are:

Purpose Contains the width of the vertical data area, that is, the number of visible lines, which are counted in the number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-48 shows the bit assignments.

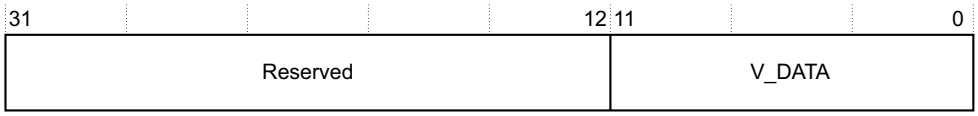


Figure 3-48 Vertical Data Width Register bit assignments

Table 3-80 shows the bit assignments.

Table 3-80 Vertical Data Width Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	V_DATA	Vertical data width -1.

Vertical Front Porch Width Register

The V_FRONT_PORCH Register characteristics are:

Purpose Contains the width of the interval between the last visible line and the next vertical synchronization, which is counted in number of horizontal scan lines.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-49 shows the bit assignments.

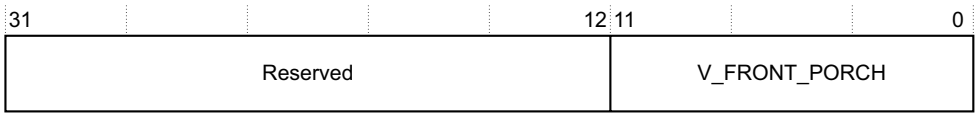


Figure 3-49 Vertical Front Porch Width Register bit assignments

Table 3-81 shows the bit assignments.

Table 3-81 Vertical Front Porch Width Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	V_FRONT_PORCH	Vertical front porch width -1.

Horizontal Synch Width Register

The H_SYNCH Register characteristics are:

Purpose Contains the width of the horizontal synch signal, which is counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-50 shows the bit assignments.

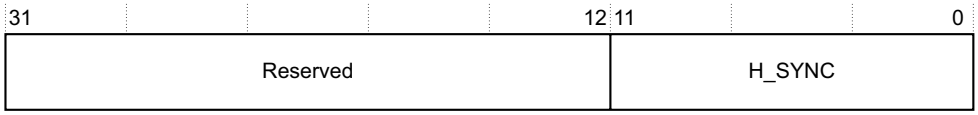


Figure 3-50 Horizontal Synch Width Register bit assignments

Table 3-82 shows the bit assignments.

Table 3-82 Horizontal Synch Width Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	H_SYNC	Horizontal synch width -1.

Horizontal Back Porch Width Register

The H_BACK_PORCH Register characteristics are:

Purpose Contains the width of the interval between the horizontal sync and the first visible column, which is counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-51 shows the bit assignments.

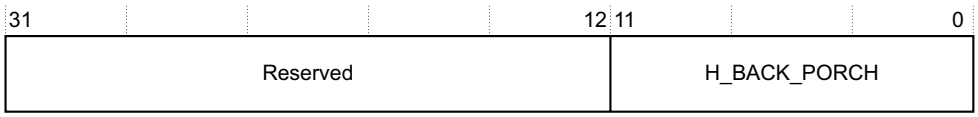


Figure 3-51 Horizontal Back Porch Width Register bit assignments

Table 3-83 shows the bit assignments.

Table 3-83 Horizontal Back Porch Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	H_BACK_PORCH	Horizontal back porch width -1.

Horizontal Data Width Register

The H_DATA Register characteristics are:

Purpose Contains the width of the horizontal data area, that is, the number of visible columns that are counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-52 shows the bit assignments.

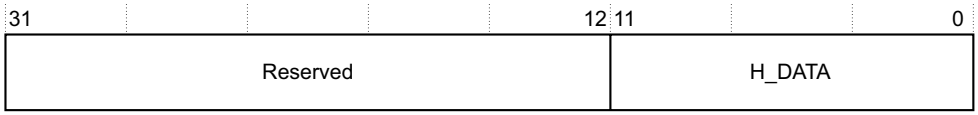


Figure 3-52 Horizontal Data Width Register bit assignments

Table 3-84 shows the bit assignments.

Table 3-84 Horizontal Data Width Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	H_DATA	Horizontal data width -1.

Horizontal Front Porch Width Register

The H_FRONT_PORCH Register characteristics are:

Purpose Contains the width of the interval between the last visible column and the next horizontal synchronization, which is counted in pixel clocks.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-53 on page 3-81 shows the bit assignments.

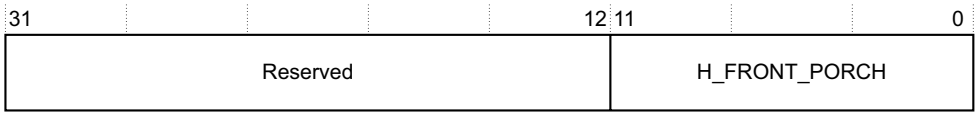


Figure 3-53 Horizontal Front Porch Width Register bit assignments

Table 3-85 shows the bit assignments.

Table 3-85 Horizontal Front Porch Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:0]	H_FRONT_PORCH	Horizontal front porch width -1.

Polarities Register

The POLARITIES Register characteristics are:

Purpose Controls the polarities of the synchronization signals and **PXLCLK** that is exported.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-54 shows the bit assignments.

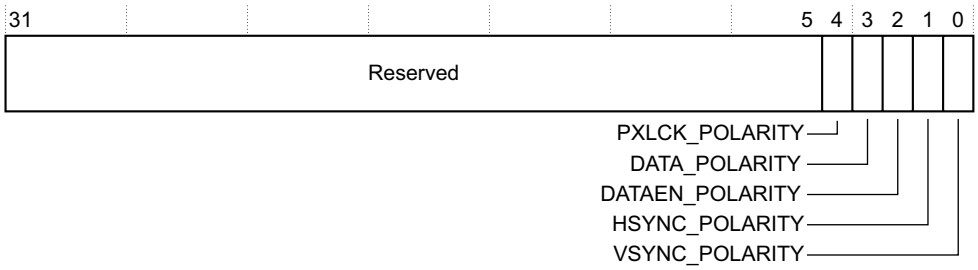


Figure 3-54 Polarities Register bit assignments

Table 3-86 shows the bit assignments.

Table 3-86 Polarities Register bit assignments

Bits	Name	Description
[31:5]	-	Reserved. Write as zero, read UNDEFINED.
[4]	PXLCLK_POLARITY	Contains value of the PXLCLKPOL output. This value is intended to be used for controlling the polarity of the pixel clock that is exported. **Note** • PXLCLK_POLARITY=b1; — **PXLCLK**, HDLCD, and **MMB_IDCLK**, export, are the same polarity. • PXLCLK_POLARITY=b0; — **PXLCLK**, HDLCD, and **MMB_IDCLK**, export, are the opposite polarity.
[3]	DATA_POLARITY	Contains the active level of the DATA output.
[2]	DATAEN_POLARITY	Contains the active level of the DATAEN output.
[1]	HSYNC_POLARITY	Contains the active level of the HSYNC output.
[0]	VSYNC_POLARITY	Contains the active level of the VSYNC output.

Command Register

The COMMAND Register characteristics are:

Purpose Starts and stops the LCD controller.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-55 shows the bit assignments.

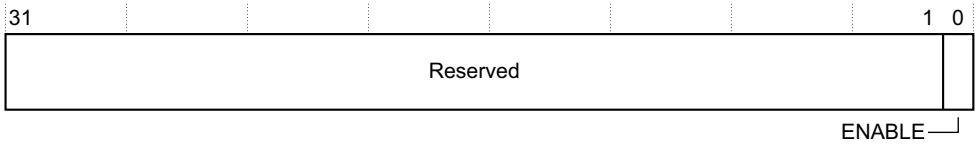


Figure 3-55 Command Register bit assignments

Table 3-87 shows the bit assignments.

Table 3-87 Command Register bit assignments

Bits	Name	Description
[31:1]	-	Reserved. Write as zero, read UNDEFINED.
[0]	ENABLE	Enable the LCD controller.

Pixel Format Register

The PIXEL_FORMAT Register characteristics are:

Purpose BYTES_PER_PIXEL plus 1 bytes are extracted from the internal buffer. The extracted bytes form a 32-bit value. If BIG_ENDIAN is set before the color components are extracted, the individual bytes are then optionally reordered.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-56 shows the little endian byte layout.

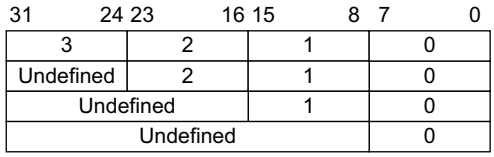


Figure 3-56 Little endian byte layout

Figure 3-57 shows the big endian byte layout.

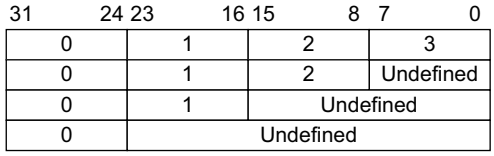


Figure 3-57 Big endian byte layout

Figure 3-58 shows the bit assignments for the big endian format.

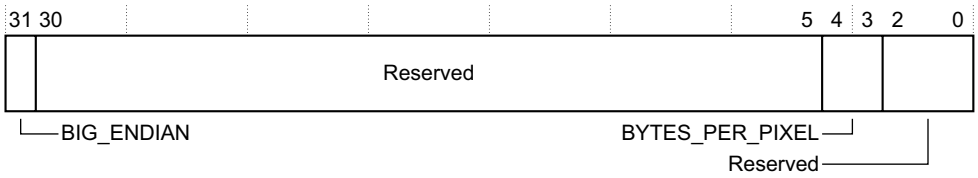


Figure 3-58 Pixel Format Register bit assignments

Table 3-88 shows the bit assignments for the big endian format.

Table 3-88 Pixel Format Register bit assignments

Bits	Name	Description
[31]	BIG_ENDIAN	Use big endian byte order.
[30:5]	-	Reserved. Write as zero, read UNDEFINED.
[4:3]	BYTES_PER_PIXEL	Number of bytes to extract from the buffer for each pixel to display, minus one.
[2:0]	-	Reserved. Write as zero, read UNDEFINED.

Color Select Registers

The RED_SELECT, GREEN_SELECT and BLUE_SELECT Registers characteristics are:

Purpose The bytes extracted from the internal buffer are presented as a 32-bit value. These registers select how many bits, and at which position, extract and use as the red, green, and blue color components. If no bits are extracted or no data is available, the default color is used.

Usage constraints There are no usage constraints.

Configurations Available in all HDLCD controller configurations.

Attributes See Table 3-67 on page 3-68.

Figure 3-59 shows the bit assignments.

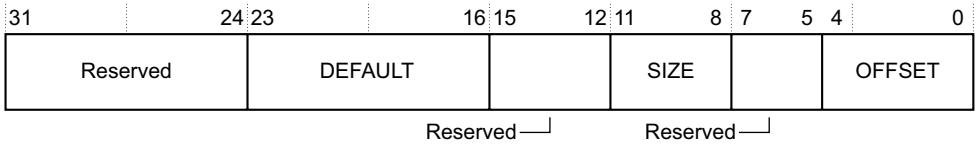


Figure 3-59 Color Select Register bit assignments

Table 3-89 shows the bit assignments.

Table 3-89 Color Select Register bit assignments

Bits	Name	Description
[31:24]	-	Reserved. Write as zero, read UNDEFINED.
[23:16]	DEFAULT	Default color. This color is used if any of the following occur: • SIZE is zero. • The buffer underruns. • While outside the visible frame area.
[15:12]	-	Reserved. Write as zero, read UNDEFINED.
[11:8]	SIZE	Number of bits to extract. If this value is zero, the default color is used. If this value is in the range 1-7, the extracted MSBs are repeated for the LSBs until 8 bits are reached. If this value is larger than 8, the behavior is UNDEFINED.
[7:5]	-	Reserved. Write as zero, read UNDEFINED.
[4:0]	OFFSET	Index of the lowest bit to extract. If OFFSET + SIZE >= 8 + 8 x BYTES_PER_PIXEL, the behavior is UNDEFINED.

3.10 PCIe Control Registers

This section describes the PCIe Control Registers. It contains the following subsections:

- *Register summary*.
- *Register descriptions* on page 3-86.

3.10.1 Register summary

Table 3-90 shows the PCIe Control Registers in offset order from the base memory address.

Table 3-90 PCIe control Registers summary

Address	Register name	Type	Reset	Width	Description
0x0000-0x00FF	PHY internal registers	RW	-	32	*PHY internal registers* on page 3-86.
0x0100-0x0FFF	-	RO	0x0	32	Reserved. RAZ/WI.
0x1000	PLL Config	RW	0x0	32	*PLL Config Register* on page 3-86.
0x1004	PCIe reset control	WO	0x0	32	*PCIe Reset Control Register* on page 3-86.
0x1008	PCIe reset status	RO	0x0	32	*PCIe Reset Status Register* on page 3-87.
0x100C	PCIe clock control	RW	0x0	32	*PCIe Clock Control Register* on page 3-88.
0x2004	PMAD_DEBUG	RO	-	32	*PMAD_DEBUG Register* on page 3-89.
0x2008	PCS_DEBUG_SEL	RW	0x0	32	*PCS_DEBUG_SEL Register* on page 3-89.
0x200C	PCS_DEBUG_OUT	RO	-	32	*PCS_DEBUG_OUT Register* on page 3-89.
0x2010	Root Port Test In Lower	RW	0x0	32	*Root Port Test In Register* on page 3-89.
0x2014	Root Port Test In Upper	RW	0x0	32	
0x2018	Root Port Test Out Lower	RO	0x0	32	*Root Port Test Out Register* on page 3-89.
0x201C	Root Port Test Out Upper	RO	0x0	32	
0x2020	Root Port LTSSM	RO	0x0	32	*Root Port LTSSM Register* on page 3-89.
0x3000	Secure	RW	0x2	32	*Secure Register* on page 3-90.
0xFFD0	PID4	RO	0x64	32	*PID4 Register* on page 3-90.
0xFFE0	PID0	RO	0xAC	32	*PID0 Register* on page 3-91.
0xFFE4	PID1	RO	0xB0	32	*PID1 Register* on page 3-91.
0xFFE8	PID2	RO	0x0B	32	*PID2 Register* on page 3-92.
0xFFEC	PID3	RO	0x00	32	*PID3 Register* on page 3-92.
0xFFF0	ID0	RO	0x0D	32	*ID0 Register* on page 3-93.
0xFFF4	ID1	RO	0xF0	32	*ID1 Register* on page 3-93.
0xFFF8	ID2	RO	0x05	32	*ID2 Register* on page 3-94.
0xFFFC	ID3	RO	0xB1	32	*ID3 Register* on page 3-94.

Note

Any memory that is not allocated is treated as RAZ/WI and does not generate an error.

3.10.2 Register descriptions

This section describes the PCIe Control registers. Table 3-90 on page 3-85 provides cross references to individual registers.

PHY internal registers

Caution

These registers are only for debug and DFT purposes. Software must not access these registers.

PLL Config Register

The PLL Config Register characteristics are:

Purpose Configures the settings of the PLL within the PHY.

Usage constraints Do not modify the values in this register after the PHY_REL bit has been set.

Attributes See Table 3-90 on page 3-85.

Figure 3-60 shows the bit assignments.

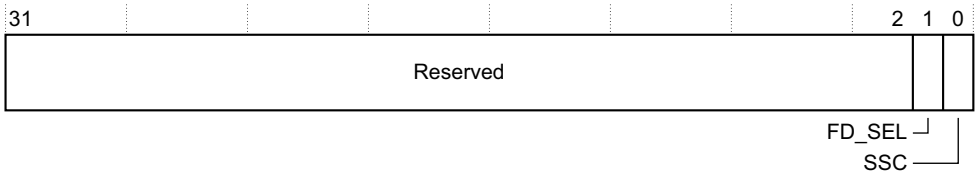


Figure 3-60 PLL Config Register bit assignments

Table 3-91 shows the bit assignments.

Table 3-91 PLL Config Register bit assignments

Bits	Name	Description
[31:2]	-	Reserved. RAZ/WI.
[1]	FD_SEL	Feedback divider input selection for the PHY PLL: 0 Integer divider. 1 Fractional divider.
[0]	SSC	PLL Spectrum Spreader Control enable: 0 No SSC. 1 Enable SSC.

PCIe Reset Control Register

The PCIe Reset Control Register characteristics are:

Purpose Reset control register for the PCIe subsystem.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-61 on page 3-87 shows the bit assignments.

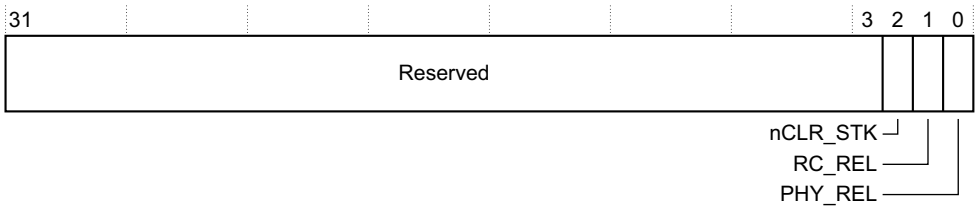


Figure 3-61 PCIe Reset Control Register bit assignments

Table 3-92 shows the bit assignments.

Table 3-92 PCIe Reset Control Register bit assignments

Bits	Name	Description
[31:3]	-	Reserved. RAZ/WI.
[2]	nCLR_STK	Select whether on the next reset of the PCIe subsystem, the sticky bits in the PCIe configuration registers inside the Root Port are reset. 0 Sticky bits are reset at next reset of PCIe subsystem. 1 Sticky bits are not reset at next reset of PCIe subsystem. **Note** In the ADP, this feature is not implemented and software is advised to always set this bit to 0.
[1]	RC_REL	Releases the transaction and PCIe layers of the Root Port. 0 Writing a 0 has no effect. 1 Writing a 1 to this register releases the transaction and PCIe layers from reset when the PCIe subsystem is released from reset. This register changes to 0 when the PCIe subsystem reset occurs.
[0]	PHY_REL	Releases the pipe interface of the PHY. 0 Writing a 0 has no effect. 1 Writing a 1 to this register releases the PHY pipe interface when the PCIe subsystem is released from reset. This register changes to 0 when the PCIe subsystem reset occurs.

Table 3-93 shows the valid combinations of RC_REL and PHY_REL.

Table 3-93 Valid combinations of RC_REL and PHY_REL

RC_REL	PHY_REL	Result
x	0	SoC reset condition. The PCIe link is held in reset.
0	1	PHY released from reset.
1	1	Root Port and PHY released from reset.

PCIe Reset Status Register

The PCIe Reset Status Register characteristics are:

Purpose Reset status register for the PCIe subsystem.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-62 shows the bit assignments.

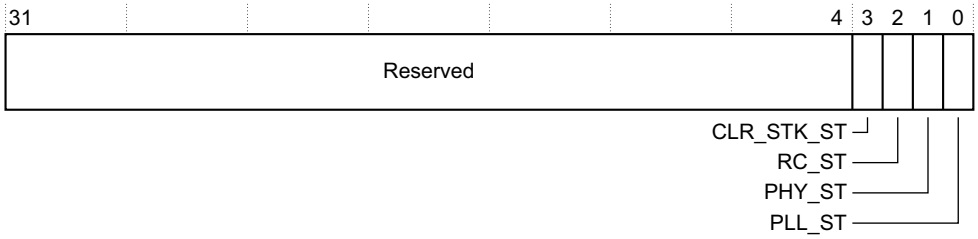


Figure 3-62 PCIe Reset Status Register bit assignments

Table 3-94 shows the bit assignments.

Table 3-94 PCIe Reset Status Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved. RAZ/WI.
[3]	CLR_STK_ST	Indicates whether on the next reset of the PCIe subsystem, the sticky bits in the PCIe configuration registers are reset. 0 Sticky bits are reset at next reset of PCIe subsystem. 1 Sticky bits are not reset at next reset of PCIe subsystem.
[2]	RC_ST	Indicates when the RC is out of reset: 0 RC is in reset. 1 RC is out of reset.
[1]	PHY_ST	Indicates when the PHY is out of reset: 0 PHY is in reset. 1 PHY out of reset.
[0]	PLL_ST	Indicates when the PLL of the PHY is stable: 0 PLL not stable. 1 PLL stable.

PCIe Clock Control Register

The PCIe Clock Control Register characteristics are:

Purpose Clock control register for the PCIe subsystem.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-63 shows the bit assignments.

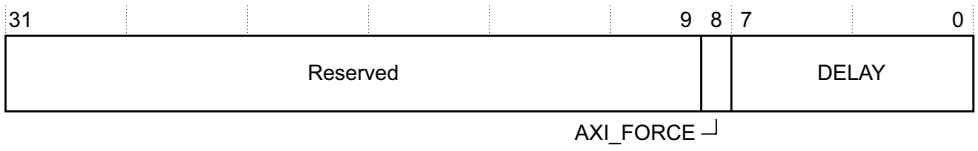


Figure 3-63 PCIe Clock Control Register bit assignments

Table 3-95 shows the bit assignments.

Table 3-95 PCIe Clock Control Register bit assignments

Bits	Name	Description
[31:9]	-	Reserved. RAZ/WI.
[8]	AXI_FORCE	AXI clock force: 0 The AXI clock clock-gates when no transactions are active. 1 The AXI clock does not clock-gate when no transactions are active.
[7:0]	DELAY	Selects the number of AXI clock cycles of bus inactivity that is required to pass before the AXI clock is gated. 0 The AXI clock is gated when all transactions have completed. 1 The AXI clock is gated one clock cycle after all transaction have completed.

PMAD_DEBUG Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

PCS_DEBUG_SEL Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

PCS_DEBUG_OUT Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

Root Port Test In Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

Root Port Test Out Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

Root Port LTSSM Register

Note

This register is for debug and DFT purposes only. Software must not access this register.

Secure Register

The Secure Register characteristics are:

Purpose Control of security.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-64 shows the bit assignments.

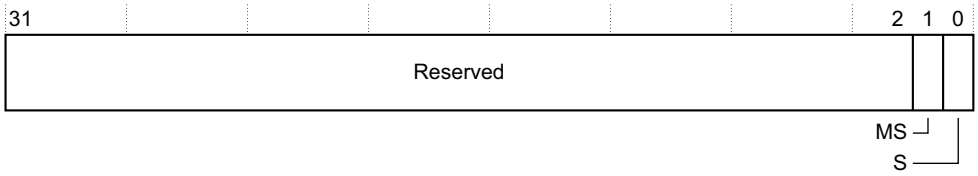


Figure 3-64 Secure Register bit assignments

Table 3-96 shows the bit assignments.

Table 3-96 Secure Register bit assignments

Bits	Name	Description
[31:2]	-	Reserved. RAZ/WI.
[1]	MS	Controls the security of transactions that the PCIe endpoints generate: 0 Secure transaction. 1 Non-secure transaction.
[0]	S	Controls whether Non-secure transactions can access the PCIe registers: 0 Only Secure. 1 Non-secure and Secure.

Note

This register is RAZ/WI for Non-secure accesses.

PID4 Register

The PID4 Register characteristics are:

Purpose Peripheral identification 4 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-65 shows the bit assignments.

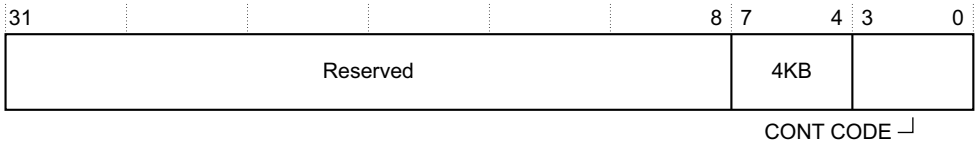


Figure 3-65 PID4 Register bit assignments

Table 3-97 shows the bit assignments.

Table 3-97 PID4 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	4KB	Number of 4KB blocks that are used by this component in a power of 2 format. For the PCIe registers, the value is `0x6`.
[3:0]	CONT CODE	The JEP106 continuation code. For ARM Limited, the value is `0x4`.

PID0 Register

The PID0 Register characteristics are:

Purpose Peripheral identification 0 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-66 shows the bit assignments.

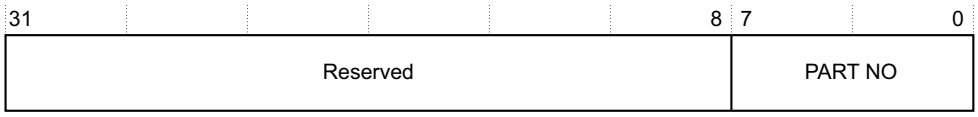


Figure 3-66 PID0 Register bit assignments

Table 3-98 shows the bit assignments.

Table 3-98 PID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PART NO	Part Number[7:0]. Reads as `0xAC`.

PID1 Register

The PID1 Register characteristics are:

Purpose Peripheral identification 4 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-67 shows the bit assignments.

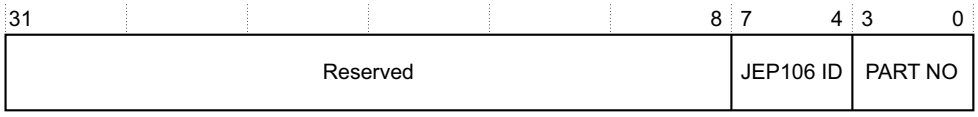


Figure 3-67 PID1 Register bit assignments

Table 3-99 shows the bit assignments.

Table 3-99 PID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	JEP106 ID	JEP106 Identity code [3:0]. For ARM Limited, the value is `0xB`.
[3:0]	PART NO	Part Number [11:8]. Reads as `0x0`.

PID2 Register

The PID2 Register characteristics are:

Purpose Peripheral identification 2 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-68 shows the bit assignments.

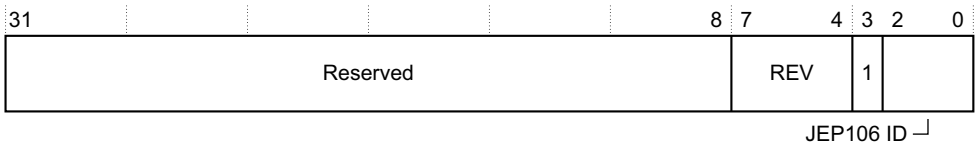


Figure 3-68 PID2 Register bit assignments

Table 3-100 shows the bit assignments.

Table 3-100 PID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	REV	Revision number. Incremental value starting at `0x0`. For the PCIe registers, the value is `0x0`.
[3]	1	Treat as RAO.
[2:0]	JEP106 ID	JEP106 identity code [6:4]. For ARM Limited, the value is `0x3`.

PID3 Register

The PID3 Register characteristics are:

Purpose Peripheral identification 3 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-69 on page 3-93 shows the bit assignments.

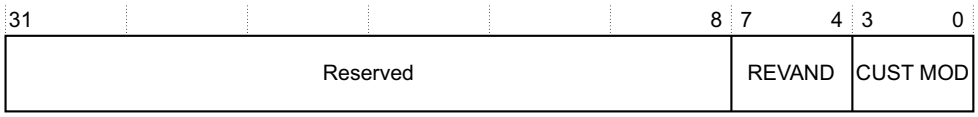


Figure 3-69 PID3 Register bit assignments

Table 3-101 shows the bit assignments.

Table 3-101 PID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	REVAND	Indicates minor errata fixes specific to this design, for example, metal fixes. For the PCIe registers, the value is `0x0`.
[3:0]	CUST MOD	Indicates customer modification to reusable IP. For the PCIe registers, the value is `0x0`.

ID0 Register

The ID0 Register characteristics are:

Purpose Identification 0 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-70 shows the bit assignments.

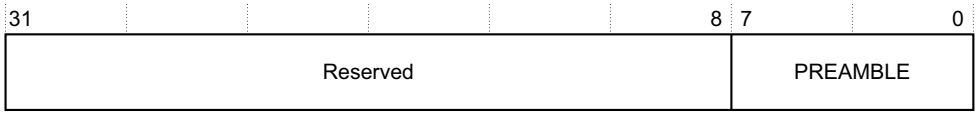


Figure 3-70 ID0 Register bit assignments

Table 3-102 shows the bit assignments.

Table 3-102 ID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMBLE	The value is `0x0D`.

ID1 Register

The ID1 Register characteristics are:

Purpose Identification 1 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-71 on page 3-94 shows the bit assignments.

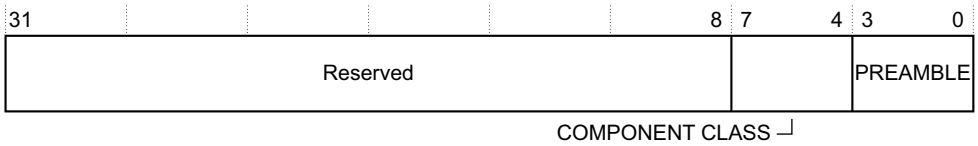


Figure 3-71 ID1 Register bit assignments

Table 3-103 shows the bit assignments.

Table 3-103 ID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	COMPONENT CLASS	The class of the component. For the PCIe registers, the value is `0xF`.
[3:0]	PREAMBLE	Value is `0x0`.

ID2 Register

The ID1 Register characteristics are:

Purpose Identification 2 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-72 shows the bit assignments.

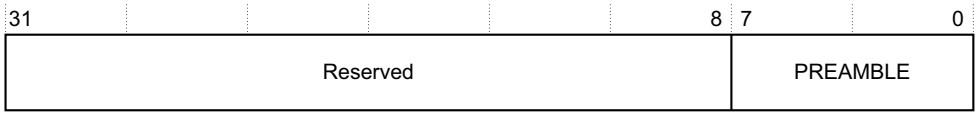


Figure 3-72 ID2 Register bit assignments

Table 3-104 shows the bit assignments.

Table 3-104 ID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMBLE	Value is `0x05`.

ID3 Register

The ID3 Register characteristics are:

Purpose Identification 3 Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-90 on page 3-85.

Figure 3-73 on page 3-95 shows the bit assignments.

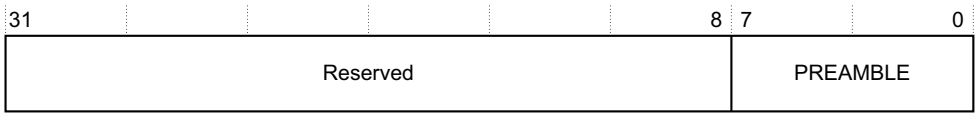


Figure 3-73 ID3 Register bit assignments

Table 3-105 shows the bit assignments.

Table 3-105 ID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMBLE	The value is `0xB1`.

3.11 PCIe Root Port configuration registers

This section describes the PCIe Root Port configuration Registers. It contains the following subsections:

- *Register summary*.
- *Control and Status Registers descriptions* on page 3-98.
- *Interrupt and event Registers descriptions* on page 3-120.
- *Address translation Registers descriptions* on page 3-126.

3.11.1 Register summary

Table 3-106 shows the PCIe Root Port configuration Registers in address order from the base memory address.

Table 3-106 PCIe Root Port configuration Registers summary

Address	Register name	Type	Width	Description
Control and Status Registers descriptions on page 3-98				
0x0000	BRIDGE_VER	RO	32	*BRIDGE_VER Register* on page 3-98
0x0004	BRIDGE_BUS	RO	32	*BRIDGE_BUS Register* on page 3-99
0x0008-0x000F	-	-	-	Reserved.
0x0010	PCIE_IF_CONF	RO	32	*PCIE_IF_CONF Register* on page 3-100
0x0014	PCIE_BASIC_CONF	RO	32	*PCIE_BASIC_CONF Register* on page 3-101
0x0018	PCIE_BASIC_STATUS	RO	32	*PCIE_BASIC_STATUS Register* on page 3-102
0x001C-0x007F	-	-	-	Reserved.
0x0080	GEN_SETTINGS	RO	32	*GEN_SETTINGS Register* on page 3-103
0x0084-0x008F	-	-	-	Reserved.
0x0090	PCIE_VC_CRED_0	RW	32	*PCIE_VC_CRED_0 Register* on page 3-103
0x0094	PCIE_VC_CRED_1			*PCIE_VC_CRED_1 Register* on page 3-104
0x0098-	PCIE_PCI_IDS_0	RW	32	*PCIE_PCI_IDS_0* on page 3-105
0x009C	PCIE_PCI_IDS_1			*PCIE_PCI_IDS_1* on page 3-105
0x00A0	PCIE_PCI_IDS_2			*PCIE_PCI_IDS_2* on page 3-106
0x00A4	PCIE_PCI_LPM	RW	32	*PCIE_PCI_LPM Register* on page 3-106
0x00A8	PCIE_PCI_IRQ_0	RW	32	*PCIE_PCI_IRQ_0 Register* on page 3-107
0x00AC	PCIE_PCI_IRQ_1			*PCIE_PCI_IRQ_1 Register* on page 3-108
0x00B0	PCIE_PCI_IRQ_2			*PCIE_PCI_IRQ_2 Register* on page 3-109
0x00B4-0x00BF	-	-	-	Reserved.
0x00C0	PCIE_PEX_DEV	RW	32	*PCIE_PEX_DEV Register* on page 3-109
0x00C4	-	-	-	Reserved.
0x00C8	PCIE_PEX_LINK	RW	32	*PCIE_PEX_LINK Register* on page 3-110

Table 3-106 PCIe Root Port configuration Registers summary (continued)

Address	Register name	Type	Width	Description
0x00CC-0x00CF	-	-	-	Reserved.
0x00D0-0x00D3	-	-	-	Reserved.
0x00D4	PCIE_PEX_SPC	RW	32	*PCIE_PEX_SPC Register* on page 3-111
0x00D8	PCIE_PEX_SPC2	RW	32	*PCIE_PEX_SPC2 Register* on page 3-112
0x00DC	PCIE_PEX_NFTS	RW	32	*PCIE_PEX_NFTS Register* on page 3-113
0x00E0-0x00FB	-	-	-	Reserved.
0x00FC	PCIE_BAR_WIN	RW	32	*PCIE_BAR_WIN Register* on page 3-113
0x0100	PCIE_EQ_PRESET_LANE_0_1	RW	32	*PCIE_EQ_PRESET_LANE_0_1 Register* on page 3-114
0x0104	PCIE_EQ_PRESET_LANE_2_3			*PCIE_EQ_PRESET_LANE_2_3 Register* on page 3-115
0x0108-0x013F	-	-	-	Reserved.
0x0140	PCIE_CFGNUM	RO	32	*PCIE_CFGNUM Register* on page 3-116
0x0144-0x0173	-	-	-	Reserved.
0x0174	PM_CONF_0	RW	32	*PM_CONF_0 Register* on page 3-117
0x0178	PM_CONF_1			*PM_CONF_1 Register* on page 3-118
0x017C	PM_CONF_2			*PM_CONF_2 Register* on page 3-119
Interrupt and event Registers descriptions on page 3-120				
0x0180	IMASK_LOCAL	RW	32	*IMASK_LOCAL Register* on page 3-120
0x0184	ISTATUS_LOCAL	RW1C	32	*ISTATUS_LOCAL Register* on page 3-120
0x0188-0x018F	-	-	-	Reserved.
0x0190	IMSI_ADDR	RO	32	*IMSI_ADDR Register* on page 3-121
0x0194	ISTATUS_MSI	RW1C	32	*ISTATUS_MSI Register* on page 3-122
0x0198	ICMD_PM	RW	32	*ICMD_PM Register* on page 3-122
0x019C-0x01D7	-	-	-	Reserved.
0x01D8	ISTATUS_P_ADT_WIN0	RO	32	*ISTATUS_P_ADT_WIN0 Register* on page 3-123
0x01DC	ISTATUS_P_ADT_WIN1	RO	32	*ISTATUS_P_ADT_WIN1 Register* on page 3-124
0x01E0	ISTATUS_A_ADT_SLV0	RO	32	*ISTATUS_A_ADT_SLV0 Register* on page 3-125
0x01E4-0x3F03	-	-	-	Reserved.
Address translation Registers descriptions on page 3-126				

Table 3-106 PCIe Root Port configuration Registers summary (continued)

Address	Register name	Type	Width	Description
0x0600-0x06FF	ATR_PCIE_WIN0	RW	32	PCIe Window 0 address translation tables 0-7.
0x0700-0x07FF	ATR_PCIE_WIN1	RW	32	PCIe Window 1 address translation tables 0-7.
0x0800-0x08FF	ATR_AXI4_SLV0	RW	32	AXI4 Slave 0 address translation tables 0-7.

Note

Program these registers before the Root Port is released from reset. See the RC_REL bit in *PCIe Reset Control Register* on page 3-86. Any changes to these registers after reset is released can lead to UNPREDICTABLE results. The exceptions are the following registers:

- *IMASK_LOCAL Register* on page 3-120.
- *ISTATUS_LOCAL Register* on page 3-120.
- *ISTATUS_MSI Register* on page 3-122.
- *ICMD_PM Register* on page 3-122.

3.11.2 Control and Status Registers descriptions

This section describes the control and status registers. Table 3-106 on page 3-96 provides cross references to individual registers.

BRIDGE_VER Register

The BRIDGE_VER Register characteristics are:

Purpose Bridge IP version and revision.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-74 shows the bit assignments.

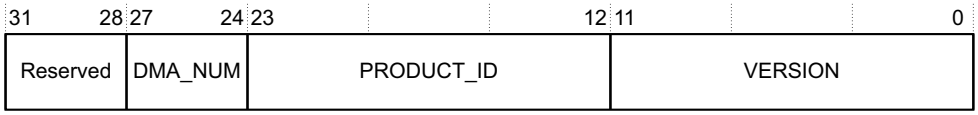


Figure 3-74 BRIDGE_VER Register bit assignments

Table 3-107 shows the bit assignments.

Table 3-107 BRIDGE_VER Register bit assignments

Bits	Name	Description
[31:28]	-	Reserved.
[27:24]	DMA_NUM	Indicates the number of DMA engines that are implemented in the core. Supported values are in the range 0x0-0x8.
[23:12]	PRODUCT_ID	Provides the bridge IP product ID, equal to 0x511.
[11:0]	VERSION	Provides the bridge IP core version. For example, 0x123 indicates version 1.2.3 of the core.

BRIDGE_BUS Register

The BRIDGE_BUS Register characteristics are:

Purpose Bridge internal bus.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-75 shows the bit assignments.

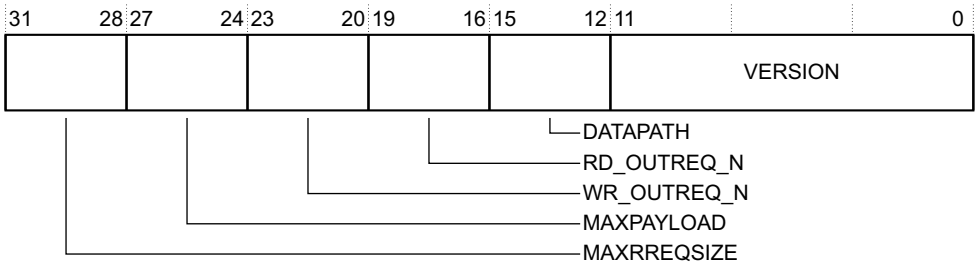


Figure 3-75 BRIDGE_BUS Register bit assignments

Table 3-108 shows the bit assignments.

Table 3-108 BRIDGE_BUS Register bit assignments

Bits	Name	Description
[31:28]	MAXRREQSIZE	Provides the maximum read request size of the bridge internal bus. Supported values are: `0` 128 bytes. `1` 256 bytes. `2` 512 bytes. `3` 1024 bytes. `4` 2048 bytes. `5` 4096 bytes.
[27:24]	MAXPAYLOAD	Provides the maximum payload size of the bridge internal bus. Supported values are: `0` 128 bytes. `1` 256 bytes. `2` 512 bytes. `3` 1024 bytes. `4` 2048 bytes. `5` 4096 bytes.
[23:20]	WR_OUTREQ_N	Number of outstanding write requests. Supported values are: `8` 256 outstanding requests.
[19:16]	RD_OUTREQ_N	Number of outstanding read requests. Supported values are: `8` 256 outstanding requests.
[15:12]	DATAPATH	Indicates the bridge internal bus data path width: `5` 256-bits
[11:0]	VERSION	Provides the bridge internal bus version. For example, `0x123` indicates version 1.2.3 of the Bus.

PCIE_IF_CONF Register

The PCIE_IF_CONF Register characteristics are:

Purpose PCIe interface configuration.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-76 shows the bit assignments.

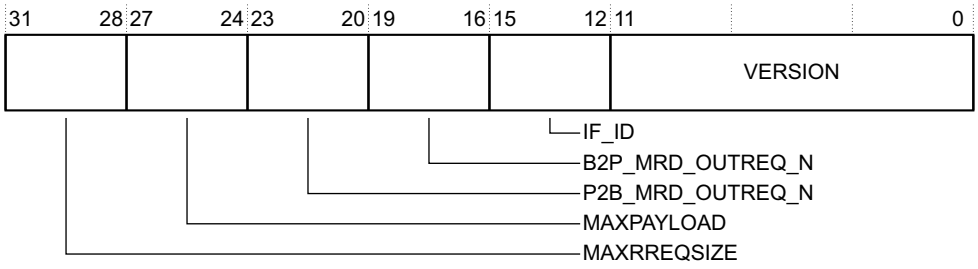


Figure 3-76 PCIE_IF_CONF Register bit assignments

Table 3-109 shows the bit assignments.

Table 3-109 PCIE_IF_CONF Register bit assignments

Bits	Name	Description
[31:28]	MAXRREQSIZE	Provides the maximum supported read request size for the PCIe interface. Supported values are: 0 128 Bytes. 1 256 Bytes. 2 512 Bytes. 3 1024 Bytes. 4 2048 Bytes. 5 4096 Bytes.
[27:24]	MAXPAYLOAD	Provides the maximum supported payload size for the PCIe interface. Supported values are: 0 128 Bytes. 1 256 Bytes. 2 512 Bytes. 3 1024 Bytes. 4 2048 Bytes. 5 4096 Bytes.
[23:20]	P2B_MRD_OUTREQ_N	Number of outstanding read requests from the PCIe domain that the bridge can handle simultaneously. Supported values are: 0 1 outstanding request. 1 2 outstanding requests. … 7 128 outstanding requests.

Table 3-109 PCIE_IF_CONF Register bit assignments (continued)

Bits	Name	Description
[19:16]	B2P_MRD_OUTREQ_N	Number of outstanding read requests that the bridge can issue to the PCIe domain. Supported values are: `0` One outstanding request. `1` Two outstanding requests. … `7` 128 outstanding requests.
[15:12]	IF_ID	Provides the ID to target this interface, that is, 0. This ID specifies the TRSL_ID fields of the *Address translation Registers descriptions* on page 3-126.
[11:0]	VERSION	Provides the PCI Express Controller core version. For example, `0x123` indicates version 1.2.3 of the core.

PCIE_BASIC_CONF Register

The PCIE_BASIC_CONF Register characteristics are:

Purpose Provides information on the implementation of the PCIe Root Port core.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-77 shows the bit assignments.

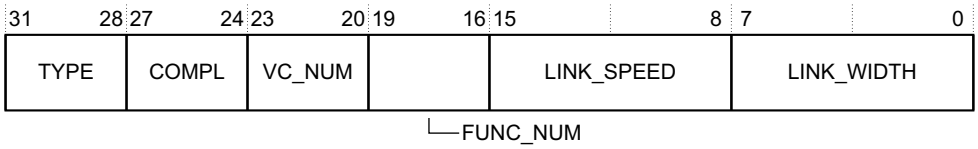


Figure 3-77 PCIE_BASIC_CONF Register bit assignments

Table 3-110 shows the bit assignments.

Table 3-110 PCIE_BASIC_CONF Register bit assignments

Bits	Name	Description
[31:28]	TYPE	Advertises the PCI Express core type. Supported values are: `0x0` Native Endpoint. `0x1` Root Port. Other values are Reserved.
[27:24]	COMPL	Advertises the core compliance to PCI Express 3.0 specification. The only supported value is `0x3`. Other values are Reserved.
[23:20]	VC_NUM	Advertises the number of virtual channels that are implemented in the core. The only supported value is `0x1`. Other values are Reserved.

Table 3-110 PCIE_BASIC_CONF Register bit assignments (continued)

Bits	Name	Description
[19:16]	FUNC_NUM	Advertises the number of functions that are implemented in the core. Supported values are between `4'h1` and `4'h8`. Other values are Reserved.
[15:8]	LINK_SPEED	Advertises the supported link speed: **Bit 0** Supports a 2.5Gbps link speed. **Bit 1** Supports a 5.0Gbps link speed. **Bit 2** Supports a 8.0Gbps link speed. **Bits 7-3** Reserved. **Note** Several bits of this field can be asserted at the same time.
[7:0]	LINK_WIDTH	Advertises the supported link width: **Bit 0** Supports x1 configuration. This bit is always asserted. **Bit 1** Supports x2 configuration. **Bit 2** Supports x4 configuration. **Bit 3** Supports x8 configuration. **Bit 4** Supports x16 configuration. **Bit 5** Supports x32 configuration. **Bits 7-6** Reserved. **Note** Several bits of this field can be asserted at the same time.

PCIE_BASIC_STATUS Register

The PCIE_BASIC_STATUS Register characteristics are:

Purpose PCIe IP Basic Status.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-78 shows the bit assignments.

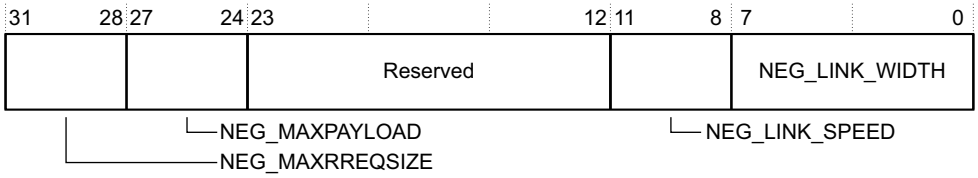


Figure 3-78 PCIE_BASIC_STATUS Register bit assignments

Table 3-111 shows the bit assignments.

Table 3-111 PCIE_BASIC_STATUS Register bit assignments

Bits	Name	Description
[31:28]	NEG_MAXRREQSIZE	Reports the negotiated maximum read request size of the PCIe link. Supported values are: 0 128 bytes. 1 256 bytes. 2 512 bytes. 3 1024 bytes. 4 2048 bytes. 5 4096 bytes.
[27:24]	NEG_MAXPAYLOAD	Reports the negotiated maximum Payload of the PCIe link. Supported values are: 0 128 bytes. 1 256 bytes. 2 512 bytes. 3 1024 bytes. 4 2048 bytes. 5 4096 bytes.
[23:12]	-	Reserved. RAZ/WI.
[11:8]	NEG_LINK_SPEED	Reports the negotiated link speed of the PCIe link. Supported values are: 1 2.5Gbps link speed. 2 5.0Gbps link speed. 3 8.0Gbps link speed.
[7:0]	NEG_LINK_WIDTH	Reports the negotiated link width of the PCIe link. Supported values are: 01 x1 link width. 02 x2 link width. 04 x4 link width. 08 x8 link width. 10 x16 link width.

GEN_SETTINGS Register

This PCIE_CFGCTRL Register is deprecated. Software must use the PCIe configuration register instead.

PCIE_VC_CRED_0 Register

The PCIE_VC_CRED_0 Register characteristics are:

Purpose Enables system firmware to set the initial flow control credit values for Virtual Channel 0.

Usage constraints All fields in this register are set to their maximum legal values at reset. An initial value of `0x0` means infinite flow control credits. A Root Port must advertise infinite credits for completion data and completion headers, so the only legal value for completion credits is `0x0`. The minimum legal value for the posted and non-posted credits is `0x1`.

Attributes See Table 3-106 on page 3-96.

Figure 3-79 on page 3-104 shows the bit assignments.

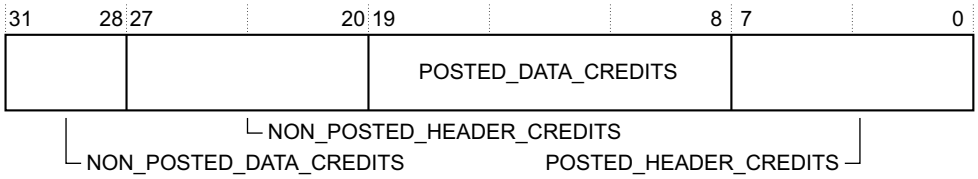


Figure 3-79 PCIE_VC_CRED_0 Register bit assignments

Table 3-112 shows the bit assignments.

Table 3-112 PCIE_VC_CRED_0 Register bit assignments

Bits	Name	Description
[31:28]	NON_POSTED_DATA_CREDITS	Bits [3:0] of Non-Posted data credits. Legal values are `0x01-0x10`.
[27:20]	NON_POSTED_HEADER_CREDITS	Number of Non-Posted Header credits. Legal values are `0x01-0x0F`.
[19:8]	POSTED_DATA_CREDITS	Number of Posted Data credits. Legal values are `0x01-0x0B8`.
[7:0]	POSTED_HEADER_CREDITS	Number of Posted Header credits. Legal values are `0x01-0x18`.

Note

Software must not rely on the values of this register after reset and is therefore required to program this register with legal values before releasing the Root Port from reset. See the RC_REL bit in the *PCIe Reset Control Register* on page 3-86.

PCIE_VC_CRED_1 Register

The PCIE_VC_CRED_1 Register characteristics are:

Purpose Enables system firmware to set the initial flow control credit values for Virtual Channel 0.

Usage constraints All fields in this register are set to their maximum legal values at reset. An initial value of `0x0` means infinite flow control credits. A Root Port must advertise infinite credits for completion data and completion headers, so the only legal value for completion credits is `0x0`. The minimum legal value for the posted and non-posted credits is `0x1`.

Attributes See Table 3-106 on page 3-96.

Figure 3-80 shows the bit assignments.

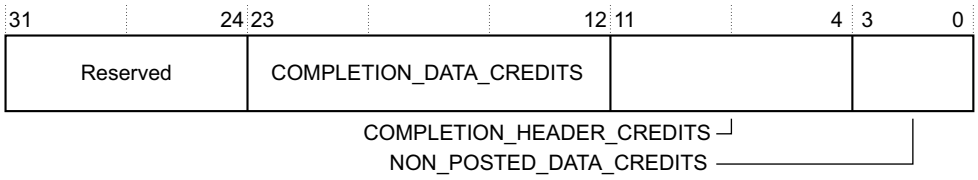


Figure 3-80 PCIE_VC_CRED_1 Register bit assignments

Table 3-113 shows the bit assignments.

Table 3-113 PCIE_VC_CRED_1 Register bit assignments

Bits	Name	Description
[31:24]	-	Reserved.
[23:12]	COMPLETION_DATA_CREDITS	Number of Completion Data credits. Legal value is `0x0`.
[11:4]	COMPLETION_HEADER_CREDITS	Number of Completion Header credits. Legal value is `0x0`.
[3:0]	NON_POSTED_DATA_CREDITS	Bits [7:4] of Non-Posted Data credits. Legal values are `0x01-0x10`.

Note

Software must not rely on the values of this register after reset and is therefore required to program this register with legal values before releasing the Root Port from reset. See the RC_REL bit in the *PCIe Reset Control Register* on page 3-86.

PCIE_PCI_IDS_0

The PCIE_PCI_IDS_0 Register characteristics are:

Purpose PCIe PCI Standard Configuration Identification Settings.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-81 shows the bit assignments.

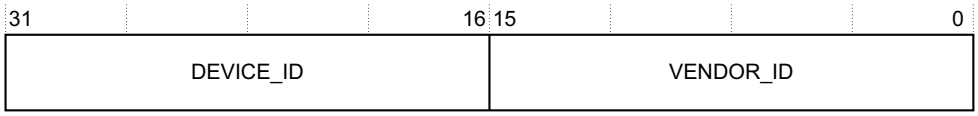


Figure 3-81 PCIE_PCI_IDS_0 Register bit assignments

Table 3-114 shows the bit assignments.

Table 3-114 PCIE_PCI_IDS_0 Register bit assignments

Bits	Name	Description
[31:16]	DEVICE_ID	Sets the value of the PCIe Device ID for the Root Port.
[15:0]	VENDOR_ID	Sets the value of the PCIe Vendor ID for the Root Port.

PCIE_PCI_IDS_1

The PCIE_PCI_IDS_1 Register characteristics are:

Purpose PCIe PCI Standard Configuration Identification Settings.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-82 on page 3-106 shows the bit assignments.

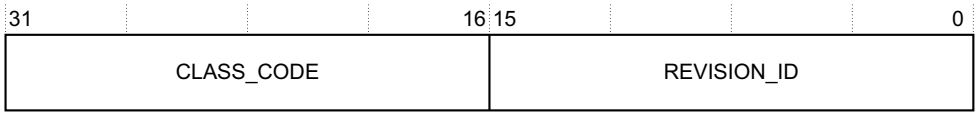


Figure 3-82 PCIE_PCI_IDS_1 Register bit assignments

Table 3-115 shows the bit assignments.

Table 3-115 PCIE_PCI_IDS_1 Register bit assignments

Bits	Name	Description
[31:8]	CLASS_CODE	Sets the value of the PCIe Class code for the Root Port.
[7:0]	REVISION_ID	Sets the value of the PCIe Revision ID for the Root Port.

PCIE_PCI_IDS_2

The PCIE_PCI_IDS_2 Register characteristics are:

Purpose PCIe PCI Standard Configuration Identification Settings.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-83 shows the bit assignments.

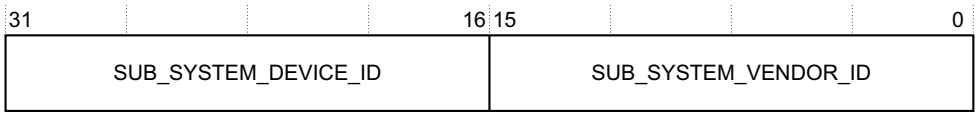


Figure 3-83 PCIE_PCI_IDS_2 Register bit assignments

Table 3-116 shows the bit assignments.

Table 3-116 PCIE_PCI_IDS_2 Register bit assignments

Bits	Name	Description
[31:16]	SUB_SYSTEM_DEVICE_ID	Sets the value of the PCIe Subsystem device ID for the Root Port.
[15:0]	SUB_SYSTEM_VENDOR_ID	Sets the value of the PCIe Subsystem vendor ID for the Root Port.

PCIE_PCI_LPM Register

The PCIE_PCI_LPM Register characteristics are:

Purpose Enables system firmware to set the values of the capability fields in the PCI Power Management Capabilities Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-84 on page 3-107 shows the bit assignments.

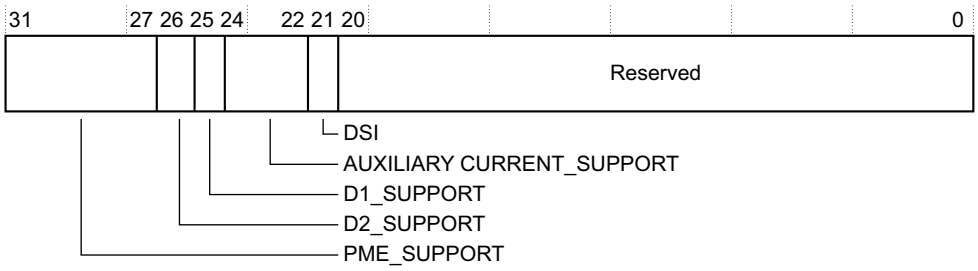


Figure 3-84 PCIE_PCI_LPM Register bit assignments

Table 3-117 shows the bit assignments.

Table 3-117 PCIE_PCI_LPM Register bit assignments

Bits	Name	Description
[31:27]	PME_SUPPORT	Enables PME support.
[26]	D2_SUPPORT	Enables D2 support.
[25]	D1_SUPPORT	Enables D1 support.
[24:22]	AUXILIARY CURRENT_SUPPORT	Sets Auxiliary current support.
[21]	DSI	Enables Device-specific initialization.
[20:0]	-	Reserved.

PCIE_PCI_IRQ_0 Register

The PCIE_PCI_IRQ_0 Register characteristics are:

Purpose PCI Interrupt, MSI, and MSI-X Settings, per function number.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-85 shows the bit assignments.

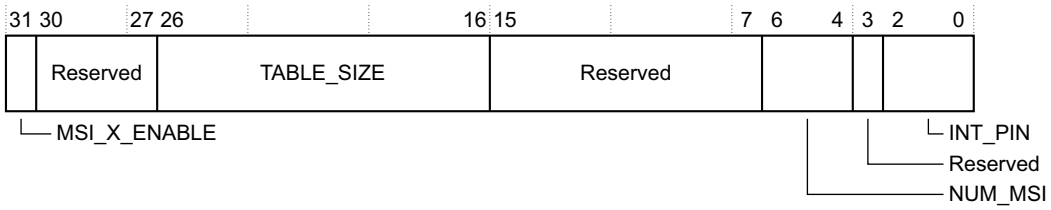


Figure 3-85 PCIE_PCI_IRQ_0 Register bit assignments

Table 3-118 shows the bit assignments.

Table 3-118 PCIE_PCI_IRQ _0 Register bit assignments

Bits	Name	Description
[31]	MSI_X_ENABLE	Enables MSI-X capability.
[30:27]	-	Reserved.
[26:16]	TABLE_SIZE	Size of MSI-X table.
[15:7]	-	Reserved.
[6:4]	NUM_MSI	Sets the number of MSI messages: 000 1. 001 2. ... ... 101 32.
[3]	-	Reserved.
[2:0]	INT_PIN	Interrupt Pin: 000 None. 001 INTA. 010 INTB. 011 INTC. 100 INTD.

Note

The use of MSI and MSI-X generation by the Root Port is deprecated. Software must instead use the GICv2m to generate MSI or MSI-X interrupts.

PCIE_PCI_IRQ_1 Register

The PCIE_PCI_IRQ_1 Register characteristics are:

Purpose PCI Interrupt, MSI, and MSI-X Settings, per function number.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-86 shows the bit assignments.

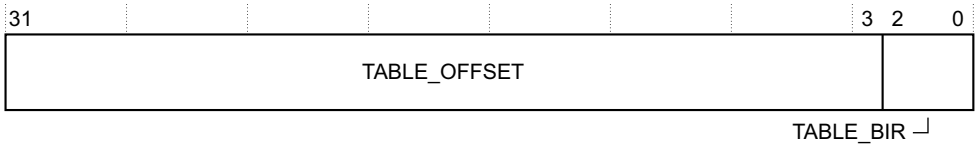


Figure 3-86 PCIE_PCI_IRQ_1 Register bit assignments

Table 3-119 shows the bit assignments.

Table 3-119 PCIE_PCI_IRQ_1 Register bit assignments

Bits	Name	Description
[31:3]	TABLE_OFFSET	Sets the PCIe MSI-X table offset field.
[2:0]	TABLE_BIR	Sets the PCIe MSI-X table BIR field.

Note

The use of MSI and MSI-X generation by the Root Port is deprecated. Software must instead use the GICv2m to generate MSI or MSI-X interrupts.

PCIE_PCI_IRQ_2 Register

The PCIE_PCI_IRQ_2 Register characteristics are:

Purpose PCI Interrupt, MSI, and MSI-X Settings, per function number.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-87 shows the bit assignments.

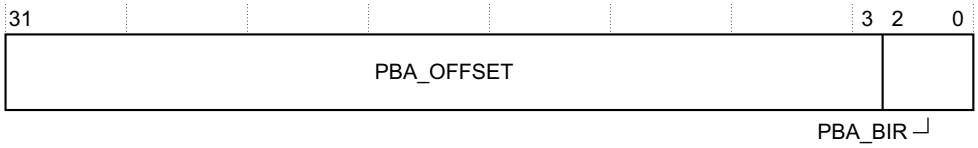


Figure 3-87 PCIE_PCI_IRQ_2 Register bit assignments

Table 3-120 shows the bit assignments.

Table 3-120 PCIE_PCI_IRQ_2 Register bit assignments

Bits	Name	Description
[31:3]	PBA_OFFSET	Sets the PCIe MSI-X PBA offset field.
[2:0]	PBA_BIR	Sets the PCIe MSI-X PBA BIR field.

Note

The use of MSI and MSI-X generation by the Root Port is deprecated. Software must instead use the GICv2m to generate MSI or MSI-X interrupts.

PCIE_PEX_DEV Register

The PCIE_PEX_DEV Register characteristics are:

Purpose Enables system firmware to set the values of the capability fields in the PCI Express Device Capabilities Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-88 shows the bit assignments.

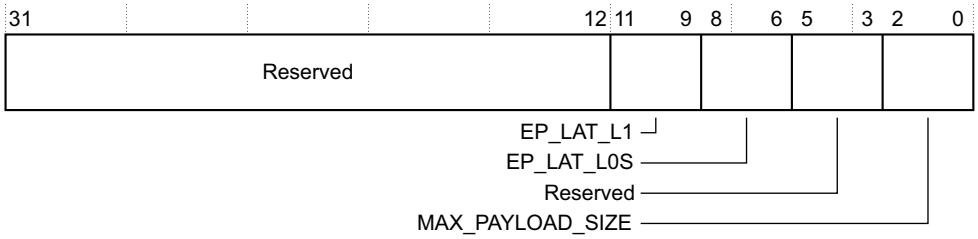


Figure 3-88 PCIE_PEX_DEV Register bit assignments

Table 3-121 shows the bit assignments.

Table 3-121 PCIE_PEX_DEV Register bit assignments

Bits	Name	Description
[31:12]	-	Reserved.
[11:9]	EP_LAT_L1	Endpoint L1 acceptable latency.
[8:6]	EP_LAT_L0S	Endpoint L0s acceptable latency.
[5:3]	-	Reserved.
[2:0]	MAX_PAYLOAD_SIZE	Maximum payload size.

PCIE_PEX_LINK Register

The PCIE_PEX_LINK Register characteristics are:

Purpose Enables system firmware to set the values of the capability fields in the PCI Express Link Capabilities Register.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-89 shows the bit assignments.

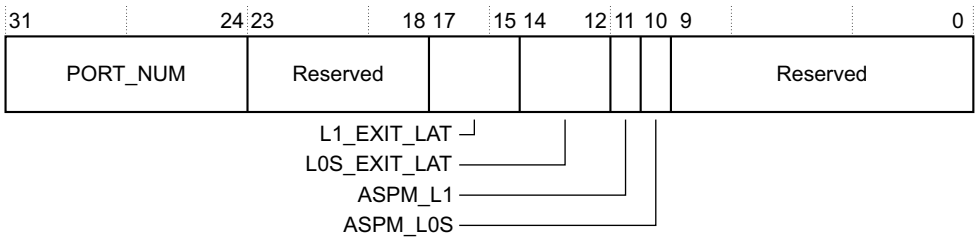


Figure 3-89 PCIE_PEX_LINK Register bit assignments

Table 3-122 shows the bit assignments.

Table 3-122 PCIE_PEX_LINK Register bit assignments

Bits	Name	Description
[31:24]	PORT_NUM	Sets the port number.
[23:18]	-	Reserved.
[17:15]	L1_EXIT_LAT	Sets the L1 exit latency.
[14:12]	L0S_EXIT_LAT	Sets the L0s exit latency.
[11]	ASPM_L1	Enables ASPM L1 support.
[10]	ASPM_L0S	Enables ASPM L0s support.
[9:0]	-	Reserved.

PCIE_PEX_SPC Register

The PCIE_PEX_SPC Register characteristics are:

Purpose Enables system firmware to set the values of miscellaneous capability fields in the PCI Configuration Space.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-90 shows the bit assignments.

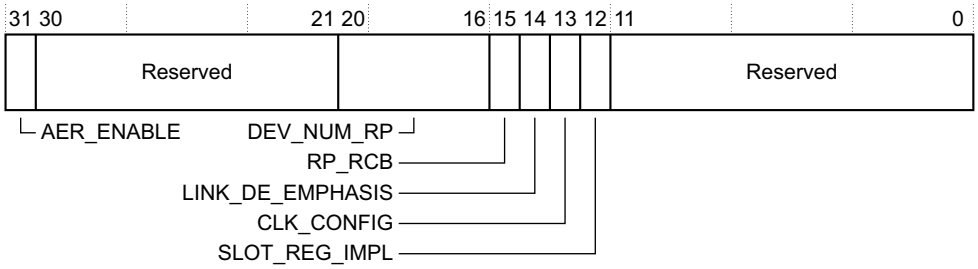


Figure 3-90 PCIE_PEX_SPC Register bit assignments

Table 3-123 shows the bit assignments.

Table 3-123 PCIE_PEX_SPC Register bit assignments

Bits	Name	Description
[31]	AER_ENABLE	Enable Advanced Error Reporting Capability.
[30:21]	-	Reserved.
[20:16]	DEV_NUM_RP	Sets the device number of the Root Port.
[15]	RP_RCB	Sets the Root Port Read Completion Boundary.
[14]	LINK_DE_EMPHASIS	Sets the initial value of the Selectable de-emphasis field in the Link Control 2 registers.

Table 3-123 PCIE_PEX_SPC Register bit assignments (continued)

Bits	Name	Description
[13]	CLK_CONFIG	Sets the Slot Clock Configuration field in the PCIe Link Status Register: 0 Independent. 1 **SYS_REF_CLK**.
[12]	SLOT_REG_IMPL	Enables PCIe Slot registers.
[11:0]	-	Reserved.

PCIE_PEX_SPC2 Register

The PCIE_PEX_SPC2 Register characteristics are:

Purpose Enables system firmware to set the values of miscellaneous capability fields in the PCI Configuration Space.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-91 shows the bit assignments.

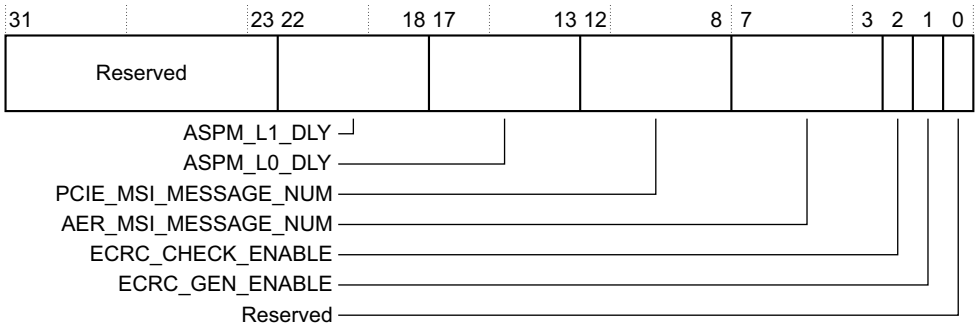


Figure 3-91 PCIE_PEX_SPC2 Register bit assignments

Table 3-124 shows the bit assignments.

Table 3-124 PCIE_PEX_SPC2 Register bit assignments

Bits	Name	Description
[31:23]	-	Reserved.
[22:18]	ASPM_L1_DLY	ASPM L1 entry delay, in steps of 256ns.
[17:13]	ASPM_L0_DLY	ASPM L0s entry delay, in steps of 256ns.
[12:8]	PCIE_MSI_MESSAGE_NUM	PCI Express MSI message number.
[7:3]	AER_MSI_MESSAGE_NUM	AER MSI message number.
[2]	ECRC_CHECK_ENABLE	Enables the ECRC Check Capable bit in the PCI Express Advanced Error Capabilities and Control Register.
[1]	ECRC_GEN_ENABLE	Enables the ECRC Generation Capable bit in the PCI Express Advanced Error Capabilities and Control Register.
[0]	-	Reserved.

PCIE_PEX_NFTS Register

The PCIE_PEX_NFTS Register characteristics are:

Purpose Number of Fast Training Sequences Settings.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-92 shows the bit assignments.

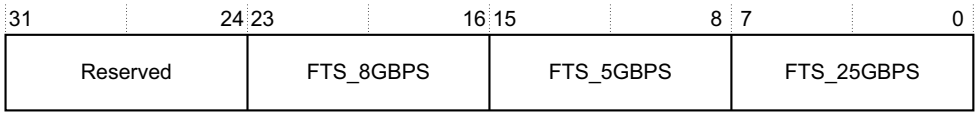


Figure 3-92 PCIE_PEX_NFTS Register bit assignments

Table 3-125 shows the bit assignments.

Table 3-125 PCIE_PEX_NFTS Register bit assignments

Bits	Name	Description
[31:24]	-	Reserved.
[23:16]	FTS_8GBPS	Number of fast training sequences at 8.0Gbps.
[15:8]	FTS_5GBPS	Number of fast training sequences at 5.0Gbps.
[7:0]	FTS_25GBPS	Number of fast training sequences at 2.5Gbps.

PCIE_BAR_WIN Register

The PCIE_PEX_NFTS Register characteristics are:

Purpose PCIe Windows Settings, per function number.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-93 shows the bit assignments.

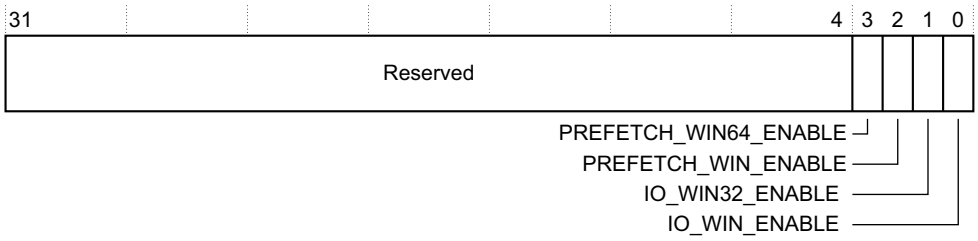


Figure 3-93 PCIE_BAR_WIN Register bit assignments

Table 3-126 shows the bit assignments.

Table 3-126 PCIE_BAR_WIN Register bit assignments

Bits	Name	Description
[31:4]	-	Reserved.
[3]	PREFETCH_WIN64_ENABLE	Prefetchable memory window 64-bit addressing support.
[2]	PREFETCH_WIN_ENABLE	Prefetchable memory window implemented.
[1]	IO_WIN32_ENABLE	I/O window 32-bit addressing support.
[0]	IO_WIN_ENABLE	I/O window is implemented.

PCIE_EQ_PRESET_LANE_0_1 Register

The PCIE_EQ_PRESET_LANE_0_1 Register characteristics are:

Purpose Sets the values of the Lane Equalization Control Register of the Secondary PCI Express Extend Capability.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-94 shows the bit assignments.

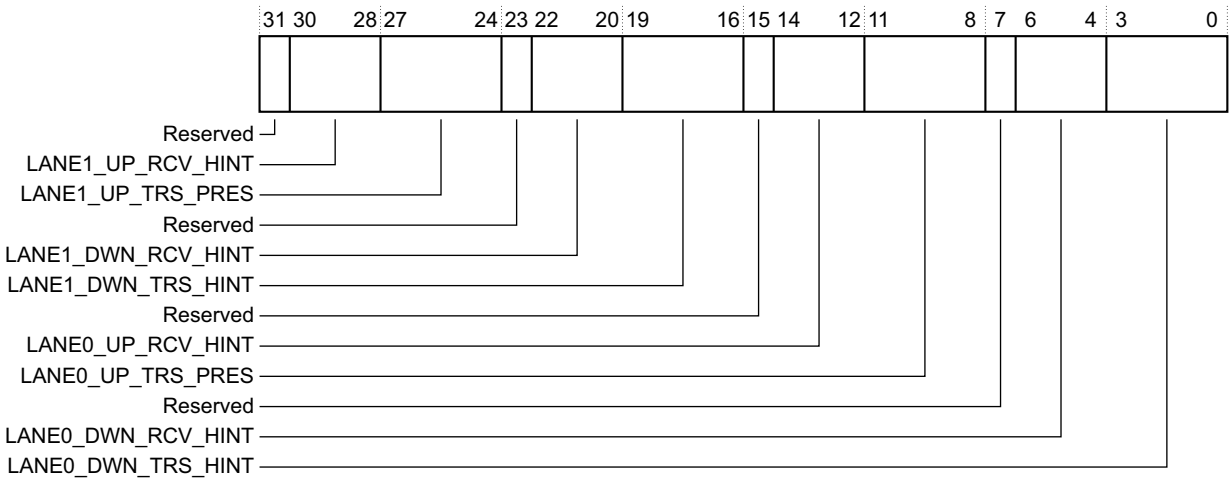


Figure 3-94 PCIE_EQ_PRESET_LANE_0_1 Register bit assignments

Table 3-127 shows the bit assignments.

Table 3-127 PCIE_EQ_PRESET_LANE_0_1 Register bit assignments

Bits	Name	Description
[31]	-	Reserved.
[30:28]	LANE1_UP_RCV_HINT	Sets the value of the upstream port receiver preset hint for lane 1.
[27:24]	LANE1_UP_TRS_PRES	Sets the value of the upstream port transmitter preset for lane 1.
[23]	-	Reserved.

Table 3-127 PCIE_EQ_PRESET_LANE_0_1 Register bit assignments (continued)

Bits	Name	Description
[22:20]	LANE1_DWN_RCV_HINT	Sets the value of the downstream port receiver preset hint for lane 1.
[19:16]	LANE1_DWN_TRS_HINT	Sets the value of the downstream port transmitter preset for lane 1.
[15]	-	Reserved.
[14:12]	LANE0_UP_RCV_HINT	Sets the value of the upstream port receiver preset hint for lane 0.
[11:8]	LANE0_UP_TRS_PRES	Sets the value of the upstream port transmitter preset hint for lane 0.
[7]	-	Reserved.
[6:4]	LANE0_DWN_RCV_HINT	Sets the value of the downstream port receiver preset hint or lane 0.
[3:0]	LANE0_DWN_TRS_HINT	Sets the value of the downstream port transmitter preset hint for lane 0.

Note

Transmitter preset and receiver preset values must be set to legal values, as specified in the PCI Express Specification.

PCIE_EQ_PRESET_LANE_2_3 Register

The PCIE_EQ_PRESET_LANE_2_3 Register characteristics are:

Purpose Sets the values of the Lane Equalization Control Register of the Secondary PCI Express Extend Capability.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-95 shows the bit assignments.

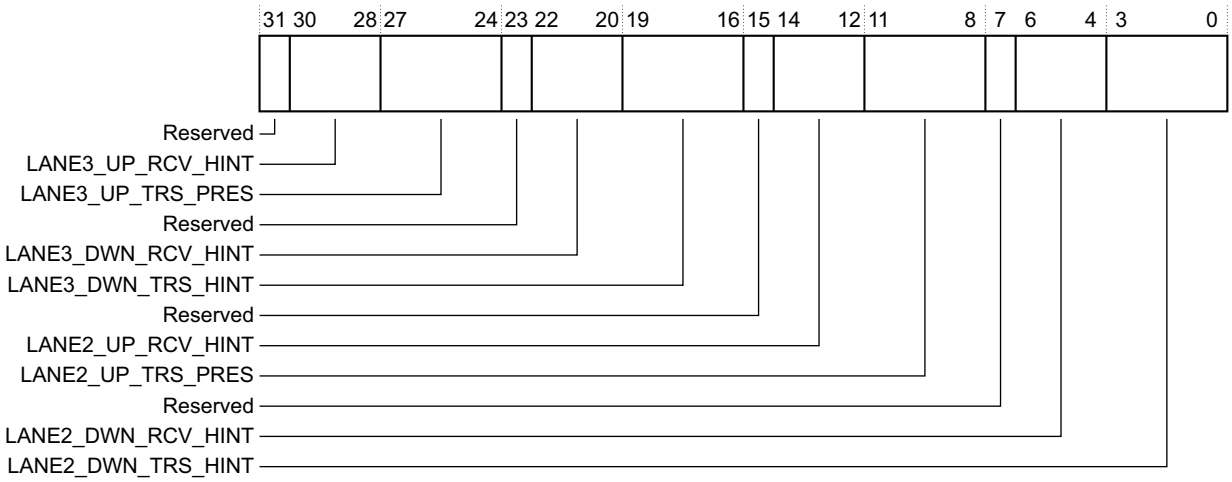


Figure 3-95 PCIE_EQ_PRESET_LANE_2_3 Register bit assignments

Table 3-128 shows the bit assignments.

Table 3-128 PCIE_EQ_PRESET_LANE_2_3 register bit assignments

Bits	Name	Description
[31]	-	Reserved.
[30:28]	LANE3_UP_RCV_HINT	Sets the value of the upstream port receiver preset hint for lane 3.
[27:24]	LANE3_UP_TRS_PRES	Sets the value of the upstream port transmitter preset for lane 3.
[23]	-	Reserved.
[22:20]	LANE3_DWN_RCV_HINT	Sets the value of the downstream port receiver preset hint for lane 3.
[19:16]	LANE3_DWN_TRS_HINT	Sets the value of the downstream port transmitter preset for lane 3.
[15]	-	Reserved.
[14:12]	LANE2_UP_RCV_HINT	Sets the value of the upstream port receiver preset hint for lane 2.
[11:8]	LANE2_UP_TRS_PRES	Sets the value of the upstream port transmitter preset hint for lane 2.
[7]	-	Reserved.
[6:4]	LANE2_DWN_RCV_HINT	Sets the value of the downstream port receiver preset hint or lane 2.
[3:0]	LANE2_DWN_TRS_HINT	Sets the value of the downstream port transmitter preset hint for lane 2.

PCIE_CFGNUM Register

The PCIE_CFGNUM Register characteristics are:

Purpose Selects the configuration space that the bridge configuration space accesses.

Usage constraints None.

Attributes See Table 3-106 on page 3-96.

Figure 3-96 shows the bit assignments.

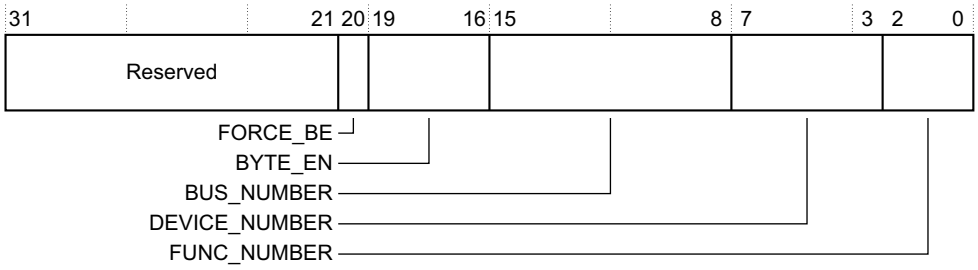


Figure 3-96 PCIE_CFGNUM Register bit assignments

Table 3-129 shows the bit assignments.

Table 3-129 PCIE_CFGNUM Register bit assignments

Bits	Name	Description
[20]	FORCE_BE	When asserted, the byte enable of the CFG read or write request is forced to the BYTE_EN field value, regardless of AXI strobes. Asserting this bit might be required, for example, when targeting the R1C register, because there is no read strobe in the AXI protocol.
[19:16]	BYTE_EN	CFG byte enable.
[15:8]	BUS_NUMBER	Bus Number.
[7:3]	DEVICE_NUMBER	Device Number.
[2:0]	FUNC_NUMBER	Function Number.

Note

This register is deprecated. Software must use the ECAM method of accessing the PCI Configuration Space. See the *PCI Express Base Specification Revision 3.0*.

PM_CONF_0 Register

The PM_CONF_0 Register characteristics are:

Purpose PCI Power Management Data Register. PCI Power Management Data Register provides the scaling factor and state dependant data that are related to the power state selected by the Data Select Register in the Power Budgeting Capability.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-97 shows the bit assignments.

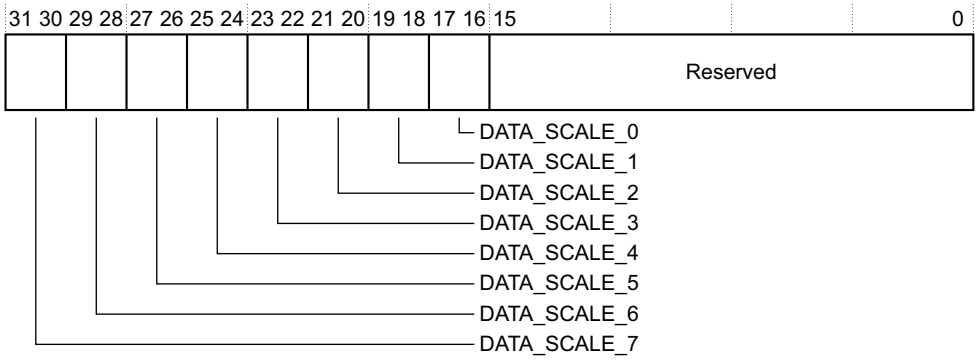


Figure 3-97 PCIE_CONF_0 Register bit assignments

Table 3-130 shows the bit assignments.

Table 3-130 PM_CONF_0 Register bit assignments

Bits	Name	Description
[31:30]	DATA_SCALE_7	Data scale value that is returned when the Data Select Register is set to 0x7: 0x0 Reserved. 0x1 0.1 multiplier. 0x2 0.01 multiplier. 0x3 0.001 multiplier.
[29:28]	DATA_SCALE_6	Data scale value that is returned when the Data Select Register is set to 0x6.
[27:26]	DATA_SCALE_5	Data scale value that is returned when the Data Select Register is set to 0x5.
[25:24]	DATA_SCALE_4	Data scale value that is returned when the Data Select Register is set to 0x4.
[23:22]	DATA_SCALE_3	Data scale value that is returned when the Data Select Register is set to 0x3.
[21:20]	DATA_SCALE_2	Data scale value that is returned when the Data Select Register is set to 0x2.
[19:18]	DATA_SCALE_1	Data scale value that is returned when the Data Select Register is set to 0x1.
[17:16]	DATA_SCALE_0	Data scale value that is returned when the Data Select Register is set to 0x0.
[15:0]	-	Reserved.

Note

See the PCI Power Management Specification, v1.2 for a full description of this register.

PM_CONF_1 Register

The PM_CONF_1 Register characteristics are:

Purpose PCI Power Management Data Register. PCI Power Management Data Register provides the scaling factor and state dependant data that are related to the power state selected by the Data Select Register in the Power Budgeting Capability.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-98 shows the bit assignments.

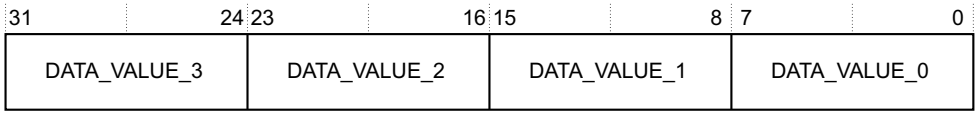


Figure 3-98 PM_CONF_1 Register bit assignments

Table 3-131 shows the bit assignments.

Table 3-131 PM_CONF_1 Register bit assignments

Bit	Name	Description
[31:24]	DATA_VALUE_3	Sets the value that is returned in the Base Power field when the Data Select Register is 0x3.
[23:16]	DATA_VALUE_2	Sets the value that is returned in the Base Power field when the Data Select Register is 0x2.
[15:8]	DATA_VALUE_1	Sets the value that is returned in the Base Power field when the Data Select Register is 0x1.
[7:0]	DATA_VALUE_0	Sets the value that is returned in the Base Power field when the Data Select Register is 0x0.

Note

See the PCI Power Management Specification, v1.2 for a full description of this register.

PM_CONF_2 Register

The PM_CONF_2 Register characteristics are:

Purpose PCI Power Management Data Register. PCI Power Management Data Register provides the scaling factor and state dependant data that are related to the power state selected by the Data Select Register in the Power Budgeting Capability.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-99 shows the bit assignments.

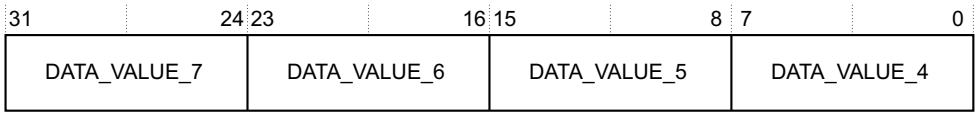


Figure 3-99 PM_CONF_2 Register bit assignments

Table 3-132 shows the bit assignments.

Table 3-132 PM_CONF_2 Register bit assignments

Bit	Name	Description
[31:24]	DATA_VALUE_7	Sets the value that is returned in the Base Power field when the Data Select Register is 0x7.
[23:16]	DATA_VALUE_6	Sets the value that is returned in the Base Power field when the Data Select Register is 0x6.
[15:8]	DATA_VALUE_5	Sets the value that is returned in the Base Power field when the Data Select Register is 0x5.
[7:0]	DATA_VALUE_4	Sets the value that is returned in the Base Power field when the Data Select Register is 0x4.

Note

See the PCI Power Management Specification, v1.2 for a full description of this register.

3.11.3 Interrupt and event Registers descriptions

This section describes the interrupt and event registers that enable, disable, monitor, and clear interrupt sources. Table 3-106 on page 3-96 provides cross references to individual registers.

IMASK_LOCAL Register

The IMASK_LOCAL Register characteristics are:

Purpose Root Port Interrupt Mask. Setting a bit enables the associated interrupt source and clearing a bit masks the interrupt source. See *ISTATUS_LOCAL Register* for information about the bits of this register.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

ISTATUS_LOCAL Register

The ISTATUS_LOCAL Register characteristics are:

Purpose Root Port Interrupt Status.

The bits of this register are automatically set when the corresponding interrupt source is activated. Each source is independent and therefore multiple sources might be active simultaneously. Software can monitor and clear status bits:

- Writing 1 clears a bit.
- Writing 0 has no effect.

If one or more ISTATUS_LOCAL interrupt sources are active and not masked by IMASK_LOCAL, an interrupt is asserted as Table 3-3 on page 3-4 shows.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-100 shows the bit assignments.

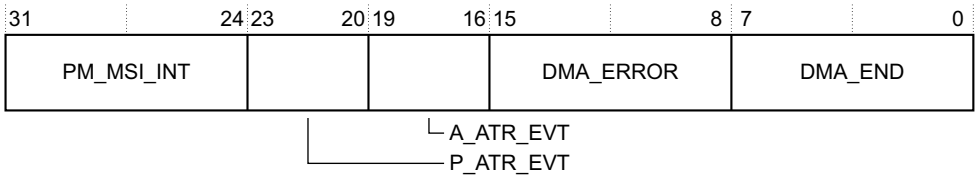


Figure 3-100 ISTATUS_LOCAL Register bit assignments

Table 3-133 shows the bit assignments.

Table 3-133 ISTATUS_LOCAL Register bit assignments

Bits	Name	Description
[31:24]	PM_MSI_INT	Reports Power Management, MSI, and Interrupts events to the application processor: 7 System error signaled, Root Port only, Reserved for Endpoint. 6 PM/Hotplug event for Root Port, Legacy power management state change for Endpoint. 5 AER event, RP only, Reserved for EP. 4 MSI received, RP only, Reserved for EP. 3 Asserted when PCI interrupt line D is asserted, RP only, Reserved for EP. 2 Asserted when PCI interrupt line C is asserted, RP only, Reserved for EP. 1 Asserted when PCI interrupt line B is asserted, RP only, Reserved for EP. 0 Asserted when PCI interrupt line A is asserted, RP only, Reserved for EP. When one of the sources of these interrupts is activated, an interrupt asserted as Table 3-3 on page 3-4 shows.
[23:20]	P_ATR_EVT	Reports PCIe address translation events: 3 PCIe Doorbell. Asserted when a PCIe request has successfully targeted an address translation table. 2 PCIe Discard Error. Asserted to signal a completion timeout on a PCIe read request. 1 PCIe Fetch Error. Asserted to indicate that an error occurred on a PCIe read request. 0 PCIe Post Error. Asserted to indicate that an error occurred on a PCIe write request.
[19:16]	A_ATR_EVT	Reports AXI address translation events: 3 AXI Doorbell. Asserted when an AXI request has successfully targeted an address translation table. 2 AXI Discard Error. Asserted to signal a completion timeout on an AXI read request. 1 AXI Fetch Error. Asserted to indicate that an error occurred on an AXI read request. 0 AXI Post Error. Asserted to indicate that an error occurred on an AXI write request.
[15:8]	DMA_ERROR	Reports that an error occurred during a DMA transfer. The bit number i corresponds to DMA Engine number i.
[7:0]	DMA_END	Reports that a DMA transfer is ended. The bit number i corresponds to DMA Engine number i.

Note

The use of ISTATUS_LOCAL and IMASK_LOCAL for the following interrupts is deprecated:

- System error.
- PM/Hotplug.
- AER.
- MSI.
- INTx.

Software must use the GIC to control and clear interrupts.

IMSI_ADDR Register

The IMSI_ADDR Register characteristics are:

Purpose Specifies the address that the Root Port uses to trigger an MSI interrupt to the application processor.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Note

- This address is 64-bit aligned. Bits [2:0] are `000`.
- This register is deprecated. Software must use the GICv2m for MSI.

ISTATUS_MSI Register

The ISTATUS_MSI Register characteristics are:

Purpose MSI Message, that is a read, write, clear register. Bits [31:0] are asserted when an MSI with message number 31-0 is received. The application processor must monitor and clear these bits:

- Writing 1 clears a bit.
- Writing 0 has no effect.

Note

MSI messages with numbers greater than 31 are ignored and discarded.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Note

This register is deprecated. Software must use the GICv2m for MSI.

ICMD_PM Register

The ICMD_PM Register characteristics are:

Purpose Event Command. Enables the application processor to activate and send events to the PCIe bus.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-101 shows the bit assignments.

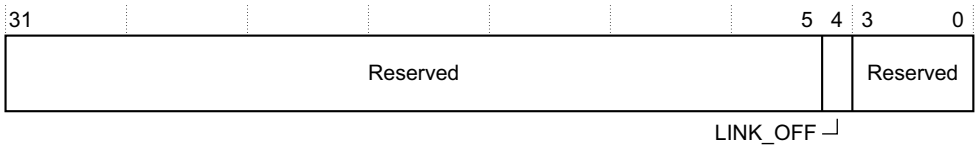


Figure 3-101 ICMD_PM Register bit assignments

Table 3-134 shows the bit assignments.

Table 3-134 ICMD_PM Register bit assignments

Bits	Name	Description
[31:5]	-	Reserved.
[4]	LINK_OFF	The application processor can send a Turn Off Link command to start L2 state entry negotiation. If the Endpoint device is also ready to enter this state, then both devices enter L2 state and this link is turned off. Deasserting this signal forces the core to exit L2 state and wakes the link.
[3:0]	-	Reserved.

ISTATUS_PM Register

The ISTATUS_PM Register characteristics are:

Purpose PCI Legacy Power Management State. Reports the PCIe power state.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-102 shows the bit assignments.

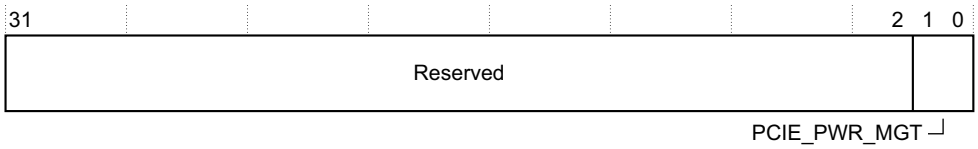


Figure 3-102 ISTATUS_PM Register bit assignments

Table 3-135 shows the bit assignments.

Table 3-135 ISTATUS_PM Register bit assignments

Bits	Name	Description
[31:2]	Reserved.	Reserved.
[1:0]	PCIE_PWR_MGT	Specifies the PCI Legacy Power Management state: 00 D0. 01 D1. 10 D2. 11 D3hot or D3cold. It is used when the PCIe is Endpoint. **Note** An interrupt in the ISTATUS_LOCAL register, bit 6 of PM_MSI_INT Field reports a change in the power management state.

ISTATUS_P_ADT_WIN0 Register

The ISTATUS_P_ADT_WIN0 Register characteristics are:

Purpose Address translation table WIN0 interrupt status.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-103 shows the bit assignments.

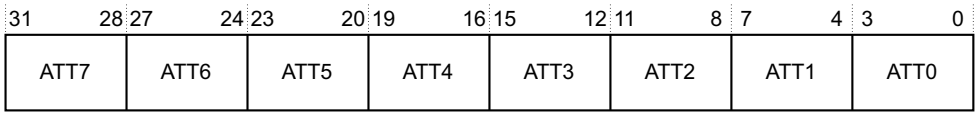


Figure 3-103 ISTATUS_P_ADT_WIN0 Register bit assignments

Table 3-136 shows the bit assignments.

Table 3-136 ISTATUS_P_ADT_WIN0 Register bit assignments

Bits	Name	Description
[31:28]	ATT7	Interrupt status for address translation table 7.
[27:24]	ATT6	Interrupt status for address translation table 6.
[23:20]	ATT5	Interrupt status for address translation table 5.
[19:16]	ATT4	Interrupt status for address translation table 4.
[15:12]	ATT3	Interrupt status for address translation table 3.
[11:8]	ATT2	Interrupt status for address translation table 2.
[7:4]	ATT1	Interrupt status for address translation table 1.
[3:0]	ATT0	Interrupt status for address translation table 0.

Table 3-137 shows the form of each nibble.

Table 3-137 ISTATUS_P_ADT_WIN0 Register nibble form

Bits	Name	Description
[3]	DOORBELL	PCIe Doorbell interrupt status. Indicates that a successful PCIe request has been translated.
[2]	DIS_ERR	PCIe Discard Error interrupt status. Indicates a completion timeout on a PCIe read request.
[1]	FETCH_ERR	PCIe Fetch Error interrupt status. Indicates that an error occurred with a PCIe read request.
[0]	POST_ERR	PCIe Post Error interrupt status. Indicates that an error occurred with a PCIe write request.

ISTATUS_P_ADT_WIN1 Register

The ISTATUS_P_ADT_WIN1 Register characteristics are:

Purpose Address translation table WIN1 interrupt status.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-104 on page 3-125 shows the bit assignments.

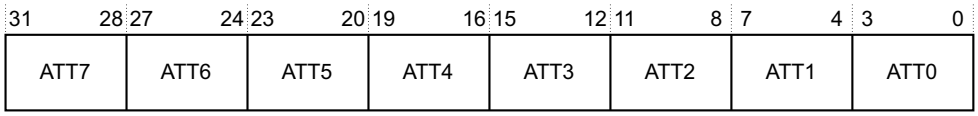


Figure 3-104 ISTATUS_P_ADT_WIN1 Register bit assignments

Table 3-138 shows the bit assignments.

Table 3-138 ISTATUS_P_ADT_WIN1 Register bit assignments

Bits	Name	Description
[31:28]	ATT7	Interrupt status for address translation table 7
[27:24]	ATT6	Interrupt status for address translation table 6
[23:20]	ATT5	Interrupt status for address translation table 5
[19:16]	ATT4	Interrupt status for address translation table 4
[15:12]	ATT3	Interrupt status for address translation table 3
[11:8]	ATT2	Interrupt status for address translation table 2
[7:4]	ATT1	Interrupt status for address translation table 1
[3:0]	ATT0	Interrupt status for address translation table 0

Table 3-139 shows the form of each nibble.

Table 3-139 ISTATUS_P_ADT_WIN1 Register nibble form

Bits	Name	Description
[3]	DOORBELL	PCIe Doorbell interrupt status. Indicates that a successful PCIe request has been translated.
[2]	DIS_ERR	PCIe Discard Error interrupt status. Indicates a completion timeout on a PCIe read request.
[1]	FETCH_ERR	PCIe Fetch Error interrupt status. Indicates that an error occurred with a PCIe read request.
[0]	POST_ERR	PCIe Post Error interrupt status. Indicates that an error occurred with a PCIe write request.

ISTATUS_A_ADT_SLV0 Register

The ISTATUS_A_ADT_SLV0 Register characteristics are:

Purpose Address translation table AXI Slave0 interrupt status.

Usage constraints There are no usage constraints.

Attributes See Table 3-106 on page 3-96.

Figure 3-105 shows the bit assignments.

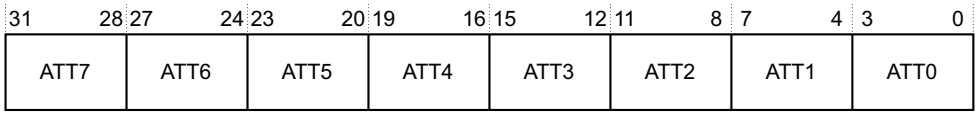


Figure 3-105 ISTATUS_A_ADT_SLV0 Register bit assignments

Table 3-140 shows the bit assignments.

Table 3-140 ISTATUS_A_ADT_SLV0 Register bit assignments

Bits	Name	Description
[31:28]	ATT7	Interrupt status for address translation table 7
[27:24]	ATT6	Interrupt status for address translation table 6
[23:20]	ATT5	Interrupt status for address translation table 5
[19:16]	ATT4	Interrupt status for address translation table 4
[15:12]	ATT3	Interrupt status for address translation table 3
[11:8]	ATT2	Interrupt status for address translation table 2
[7:4]	ATT1	Interrupt status for address translation table 1
[3:0]	ATT0	Interrupt status for address translation table 0

Table 3-141 shows the form of each nibble.

Table 3-141 ISTATUS_A_ADT_SLV0 Register nibble form

Bits	Name	Description
[3]	DOORBELL	AXI Doorbell interrupt status. Indicates that a successful AXI request has been translated.
[2]	DIS_ERR	AXI Discard Error interrupt status. Indicates a completion timeout on an AXI read request.
[1]	FETCH_ERR	AXI Fetch Error interrupt status. Indicates that an error has occurred with an AXI read request.
[0]	POST_ERR	AXI Post Error interrupt status. Indicates that an error has occurred with an AXI write request.

Note

An AXI transaction is a transaction that the application processor issues to the PCIe system.

3.11.4 Address translation Registers descriptions

This section describes the address translation registers. Table 3-106 on page 3-96 provides cross references to individual registers.

Three sets of Address translation tables exist, and each set contains up to eight individual programmable tables.

Two sets of Address translation tables translate from the PCIe address space to the application processor address space, one for each BAR implemented by the Root Port.

The other table performs address translation from the application processor address space to the PCIe address space.

Table 3-142 shows the layout of the individual address translation tables within the set.

Table 3-142 Address translation table register blocks

Address range	Name	Description
0x00-0x1F	TABLE0	Registers for address translation table 0.
0x20-0x3F	TABLE1	Registers for address translation table 1.
0x40-0x5F	TABLE2	Registers for address translation table 2.
0x60-0x7F	TABLE3	Registers for address translation table 3.
0x80-0x9F	TABLE4	Registers for address translation table 4.
0xA0-0xBF	TABLE5	Registers for address translation table 5.
0xC0-0xDF	TABLE6	Registers for address translation table 6.
0xE0-0xFF	TABLE7	Registers for address translation table 7.

Table 3-143 shows the registers within an address translation table.

Table 3-143 Address translation table Registers

Address	Register name	Type	Width	Description
0x00-0x03	SRC_ADDR_LO	RW	32	*SRC_ADDR_LO Register* on page 3-128
0x04-0x07	SRC_ADDR_UP	RW	32	*SRC_ADDR_UP Register* on page 3-128
0x08-0x0B	TRSL_ADDR_LO	RW	32	*TRSL_ADDR_LO Register* on page 3-128
0x0C-0x0F	TRSL_ADDR_UP	RW	32	*TRSL_ADDR_UP register* on page 3-129
0x10-0x13	TRSL_PARAM	RW	32	*TRSL_PARAM Register* on page 3-129
0x14-0x17	-	RO	-	Reserved
0x18-0x1F	TRSL_MASK	RO	32	*TRSL_MASK Register* on page 3-130

Note

- Table 0 of AXI4 Slave 0 Address translation tables is RO and is configured to map the PCIe configuration space to 0x4000_0000 in the application processor memory map.
- Tables 1-7 are RW and software must use them to map the address translation from the application processor address space to the PCIe address space.
- SRC_ADDR and TRSL_ADDR are aligned to the size of the translation table.

Note

If an address translation event occurs, it is reported to the ISTATUS_X_ADT_X registers and *ISTATUS_LOCAL Register* on page 3-120.

You can enable interrupts by using the *IMASK_LOCAL Register* on page 3-120 and clear interrupts by using the *ISTATUS_LOCAL Register* on page 3-120.

SRC_ADDR_LO Register

The SRC_ADDR_LO Register characteristics are:

Purpose Sets the source address and size of the region of memory that incoming transactions must address for the translation table to be applied.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

Figure 3-106 shows the bit assignments.

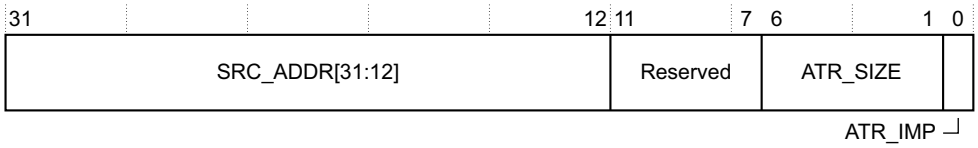


Figure 3-106 SRC_ADDR_LO Register bit assignments

Table 3-144 shows the bit assignments.

Table 3-144 SRC_ADDR_LO Register bit assignments

Bits	Name	Description
[31:12]	SRC_ADDR [31:12]	Bits [31:12] of the source address.
[11:7]	-	Reserved.
[6:1]	ATR_SIZE	Defines the Address Translation Space Size. This space size in bytes is equal to $2^{(ATR\_SIZE+1)}$. Permitted values for this field are from `6'd11` (2^{12} =4 KBytes) to `6'd63` (2^{64} = 16 Exabytes) only.
[0]	ATR_IMP	When set to 1, indicates that the Translation Address Table is enabled.

SRC_ADDR_UP Register

The SRC_ADDR_UP Register characteristics are:

Purpose Sets the upper 32 bits of the source address.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

Table 3-145 shows the bit assignments.

Table 3-145 SRC_ADDR_UP Register bit assignments

Bits	Name	Description
[31:0]	SRC_ADDR[63:32]	Bits [63:32] of the source address.

TRSL_ADDR_LO Register

The TRSL_ADDR_LO Register characteristics are:

Purpose Sets bits [31:12] of the translated address.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

Figure 3-107 shows the bit assignments.

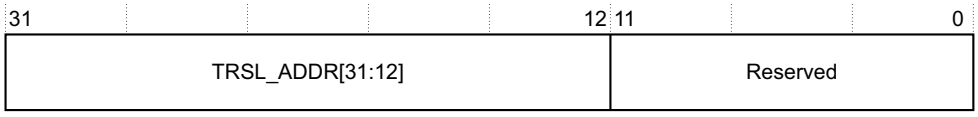


Figure 3-107 TRSL_ADDR_LO Register bit assignments

Table 3-146 shows the bit assignments.

Table 3-146 TRSL_ADDR_LO Register bit assignments

Bits	Name	Description
[31:12]	TRSL_ADDR[31:12]	Bits [31:12] of the translated address.
[11:0]	-	Reserved.

TRSL_ADDR_UP register

The TRSL_ADDR_UP Register characteristics are:

Purpose Sets bits [63:32] of the translated address.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

Table 3-147 shows the bit assignments.

Table 3-147 TRSL_ADDR[63:32] Register bit assignments

Bits	Name	Description
[31:0]	TRSL_ADDR[63:32]	Bits [63:32] of the translated address.

TRSL_PARAM Register

The TRSL_PARAM Register characteristics are:

Purpose Sets the transaction properties and destination of the transactions that this table translates.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

Figure 3-108 shows the bit assignments.

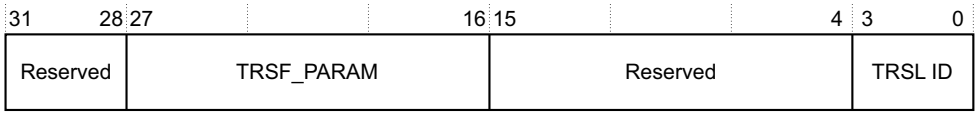


Figure 3-108 TRSL_PARAM Register bit assignments

Table 3-148 shows the bit assignments.

Table 3-148 TRSL_PARAM Register bit assignments

Bits	Name	Description
[31:28]	-	Reserved.
[27:16]	TRSF_PARAM	Defines the parameters of the translated transactions: If TRSL_ID is PCIe: [11] Reserved. [10:8] Traffic class. [7] ECRC Forward. [6] EP. [5:4] Attr. [3] Reserved. [2:0] TLP Type, as follows: 3'b000 Memory. 3'b001 Memory Locked. 3'b010 I/O. 3'b100 Message. If TRSL_ID is AXI4 Master: [11:8] AxQOS. [7:5] The MS bit in the *Secure Register* on page 3-90 controls AxPROT. AxPROT[1]. [4] AxLOCK. [3:0] AxCACHE.
[15:4]	-	Reserved.
[3:0]	TRSL_ID	Sets the target for the translated transaction. The permitted values are: 4'd0 PCIe Tx/Rx interface. 4'd1 PCIe config interface. **Note** Table 0 of ATR_AXI_SLV0 is configured to map 0x4000_0000 to the PCIe configuration space. ARM recommends that software does not map another region of memory to the configuration space. 4'd4 AXI4 Master 0. **Note** All other values are Reserved.

TRSL_MASK Register

The TRSL_MASK Register characteristics are:

Purpose Defines the translation table mask address. It is equal to:

2^{64} – Table Size

Where the table size is equal to $2^{(ATR\_SIZE+1)}$. For example, if the table size is fixed to 256 KBytes, TRSL_MASK is equal to:

0x16EB – 256KBytes = 0xFFFFFFFFFFFC0000.

This register is automatically updated when ATR_SIZE in the *SRC_ADDR_LO Register* on page 3-128 is updated.

Usage constraints There are no usage constraints.

Attributes See Table 3-143 on page 3-127.

3.12 MSI Registers

This section describes the *Message Signaled Interrupt* (MSI) Registers. It contains the following subsections:

- *Register summary*.
- *Register descriptions*.

3.12.1 Register summary

Table 3-149 shows the MSI Registers in offset order from the base memory address.

Table 3-149 MSI Registers summary

Address	Name	Type	Reset	Description
0x0000-0x0004	-	-	0x0	Reserved. RAZ/WI
0x0008	MSI_TYPER	RO	IMPLEMENTATION DEFINED	*MSI_TYPER Register*
0x000C-0x003C	-	-	0x0	Reserved. RAZ/WI
0x0040	MSI_SETSPI_NSR	WO	-	*MSI_SETSPI_NSR Register* on page 3-133
0x0044 – 0x0FC8	-	-	0x0	Reserved. RAZ/WI
0x0FCC	MSI_IIDR	RO	0x0AE0043B	*MSI_IIDR Register* on page 3-133
0x0FD0	Peripheral ID4	RO	0x4	*Peripheral ID4 Register* on page 3-134
0x0FE0	Peripheral ID0	RO	0xAE	*Peripheral ID0 Register* on page 3-134
0x0FE4	Peripheral ID1	RO	0xB0	*Peripheral ID1 Register* on page 3-135
0x0FE8	Peripheral ID2	RO	0x1B	*Peripheral ID2 Register* on page 3-135
0x0FEC	Peripheral ID3	RO	0x0	*Peripheral ID3 Register* on page 3-136
0x0FF0	Component ID0	RO	0xD	*Component ID0 Register* on page 3-136
0x0FF4	Component ID1	RO	0xF0	*Component ID1 Register* on page 3-137
0x0FF8	Component ID2	RO	0x5	*Component ID2 Register* on page 3-137
0x0FFC	Component ID3	RO	0xB1	*Component ID3 Register* on page 3-138
0x1000-0xFFFF	-	-	0x0	Reserved. RAZ/WI

3.12.2 Register descriptions

This section describes the MSI registers. Table 3-149 provides cross references to individual registers.

MSI_TYPER Register

Figure 3-109 shows the bit assignments.

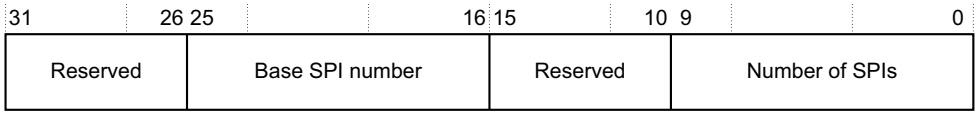


Figure 3-109 MSI_TYPER Register bit assignments

Table 3-150 shows the bit assignments.

Table 3-150 MSI_TYPER Register bit assignments

Bits	Name	Description
[31:26]	-	Reserved.
[25:16]	Base SPI number	Returns the lowest SPI assigned to the frame. For: **Frame 0** Value reads as 224. **Frame 1** Value reads as 256. **Frame 2** Value reads as 288. **Frame 3** Value reads as 320.
[15:10]	-	Reserved.
[9:0]	Number of SPIs	Number of contiguous SPIs assigned to the frame: `0x20` 32 contiguous SPIs. **All other values** Reserved.

MSI_SETSPI_NSR Register

See the *ARM® Server Base System Architecture Platform Design Document*, http://infocenter.arm.com/help/topic/com.arm.doc.den0029/index.html.

MSI_IIDR Register

Figure 3-110 shows the bit assignments.

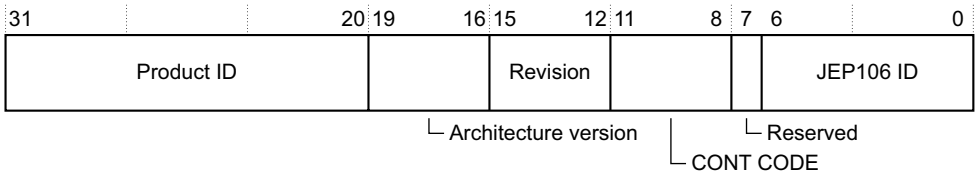


Figure 3-110 MSI_IIDR Register bit assignments

Table 3-151 shows the bit assignments.

Table 3-151 MSI_IIDR Register bit assignments

Bits	Name	Description
[31:20]	Product ID	-
[19:16]	Architecture version	This field is `0x0` for GICv2m v0.
[15:12]	Revision	Revision.
[11:8]	CONT CODE	JEP106 continuation code.
[7]	-	Reserved. RAZ
[6:0]	JEP106 ID	JEP106 identity code.

Peripheral ID4 Register

The PID4 Register characteristics are:

Purpose PID4.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-111 shows the bit assignments.

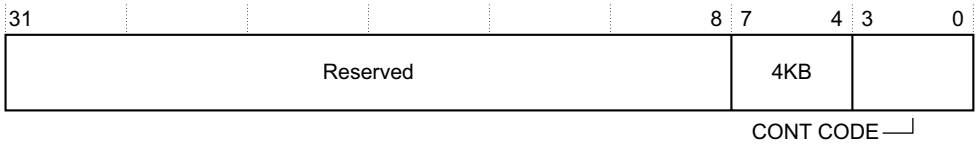


Figure 3-111 PID4 Register bit assignments

Table 3-152 shows the bit assignments.

Table 3-152 PID4 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	4KB	Number of 4KB blocks that are used by this component in a power of 2 format. For the ADP MSI, the value is `0x6`.
[3:0]	CONT CODE	The JEP106 continuation code. For ARM Limited, this field is `0x4`.

Peripheral ID0 Register

The PID0 Register characteristics are:

Purpose Peripheral identification 0.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-112 shows the bit assignments.

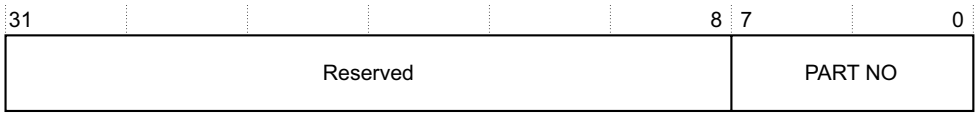


Figure 3-112 PID0 Register bit assignments

Table 3-153 shows the bit assignments.

Table 3-153 PID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PART NO	Part Number[7:0]. Reads as `0xAE`.

Peripheral ID1 Register

The PID1 Register characteristics are:

Purpose Peripheral identification 1.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-113 shows the bit assignments.

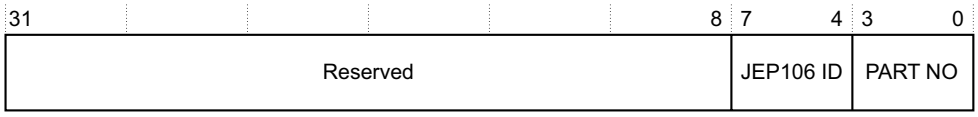


Figure 3-113 PID1 Register bit assignments

Table 3-154 shows the bit assignments.

Table 3-154 PID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	JEP106 ID	JEP106 Identity code [3:0]. For ARM Limited, the value is `0xB`.
[3:0]	PART NO	Part Number [11:8]. Reads as `0x0`.

Peripheral ID2 Register

The PID2 Register characteristics are:

Purpose Peripheral identification 2.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-114 shows the bit assignments.

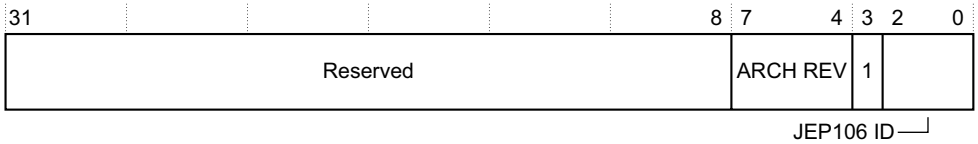


Figure 3-114 PID2 Register bit assignments

Table 3-155 shows the bit assignments.

Table 3-155 PID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.

Table 3-155 PID2 Register bit assignments (continued)

Bits	Name	Description
[7:4]	ARCHREV	Revision Number. Incremental value starting at `0x0`. For the ADP MSI, the value is `0x0`.
[3]	1	Treat as RAO.
[2:0]	JEP106 ID	JEP106 Identity code [6:4]. For ARM Limited, the value is `0x3`.

Peripheral ID3 Register

The PID3 Register characteristics are:

Purpose Peripheral identification 3.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-115 shows the bit assignments.

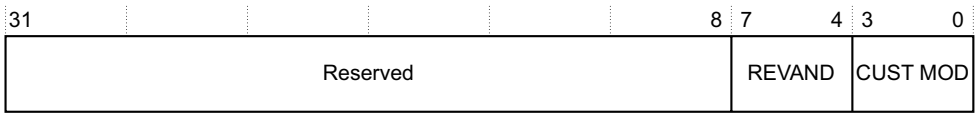


Figure 3-115 PID3 Register bit assignments

Table 3-156 shows the bit assignments.

Table 3-156 PID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	REVAND	Indicates minor errata fixes specific to this design, for example, metal fixes. For the ADP MSI, the value is `0x0`.
[3:0]	CUST MOD	Indicates customer modification to reusable IP. For the ADP MSI, the value is `0x0`.

Component ID0 Register

The ID0 Register characteristics are:

Purpose Identification 0.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-116 shows the bit assignments.

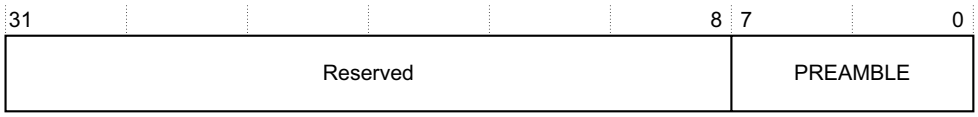


Figure 3-116 ID0 Register bit assignments

Table 3-157 shows the bit assignments.

Table 3-157 ID0 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMBLE	0x0D.

Component ID1 Register

The ID1 Register characteristics are:

Purpose Identification 1.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-117 shows the bit assignments.

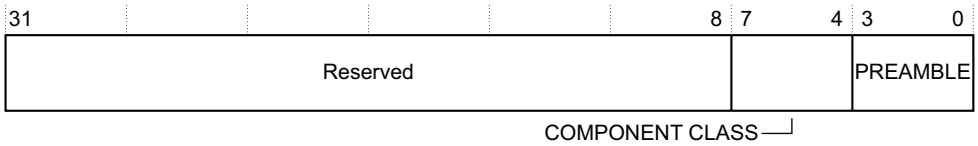


Figure 3-117 ID1 Register bit assignments

Table 3-158 shows the bit assignments.

Table 3-158 ID1 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:4]	COMPONENT CLASS	The class of the component. For the CSS MSI, the value is 0xF.
[3:0]	PREAMBLE	The value is 0x0.

Component ID2 Register

The ID2 Register characteristics are:

Purpose Identification 2.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-118 shows the bit assignments.

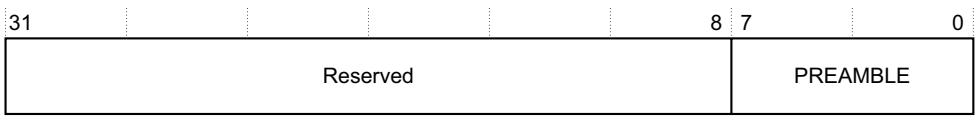


Figure 3-118 ID2 Register bit assignments

Table 3-159 shows the bit assignments.

Table 3-159 ID2 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMABLE	The value is `0x05`.

Component ID3 Register

The ID3 Register characteristics are:

Purpose Identification 3.

Usage constraints There are no usage constraints.

Attributes See Table 3-149 on page 3-132.

Figure 3-119 shows the bit assignments.

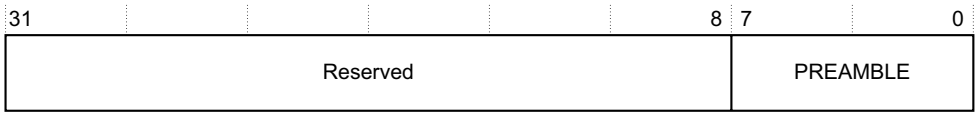


Figure 3-119 ID3 Register bit assignments

Table 3-160 shows the bit assignments.

Table 3-160 ID3 Register bit assignments

Bits	Name	Description
[31:8]	-	Reserved. RAZ/WI.
[7:0]	PREAMBLE	The value is `0xB1`.

3.13 Trusted Entropy Source Registers

This section describes the Trusted Entropy Source Registers. It describes:

- *Register summary*.
- *Register descriptions*.

3.13.1 Register summary

Table 3-161 shows the registers in offset order from the base memory address.

Table 3-161 Trusted Entropy Source Registers summary

Offset	Name	Type	Reset value	Description
0x00	OUTPUT_0	RO	0x0	*RNG Output, OUTPUT_0, Register*
0x04	OUTPUT_1	RO	0x0	*RNG Output, OUTPUT_1, Register* on page 3-140
0x08	OUTPUT_2	RO	0x0	*RNG Output, OUTPUT_2, Register* on page 3-140
0x0C	OUTPUT_3	RO	0x0	*RNG Output, OUTPUT_3, Register* on page 3-140
0X10	STATUS	RW	0x0	*STATUS Register* on page 3-141
0x14	INTMASK	RW	0x0	*Interrupt Mask, INTMASK, Register* on page 3-141
0x18	CONFIG	RW	0x0	*Configuration, CONFIG, Register* on page 3-142
0X1C	CONTROL	WO	0x0	*CONTROL Register* on page 3-142
0xFD0	Peripheral ID4	RO	0x04	*Peripheral ID4 Register* on page 3-134
0xFE0	Peripheral ID0	RO	0xAA	*Peripheral ID0 Register* on page 3-134
0xFE4	Peripheral ID1	RO	0xB0	*Peripheral ID1 Register* on page 3-135
0xFE8	Peripheral ID2	RO	0x2B	*Peripheral ID2 Register* on page 3-135
0xFEC	Peripheral ID3	RO	0x00	*Peripheral ID3 Register* on page 3-136
0xFF0	Component ID0	RO	0x0D	*Component ID0 Register* on page 3-136
0xFF4	Component ID1	RO	0xF0	*Component ID1 Register* on page 3-137
0xFF8	Component ID2	RO	0x05	*Component ID2 Register* on page 3-137
0xFFC	Component ID3	RO	0xB1	*Component ID3 Register* on page 3-138

3.13.2 Register descriptions

This section describes the trusted entropy source registers. Table 3-161 provides cross references to individual registers.

RNG Output, OUTPUT_0, Register

The RNG Output, OUTPUT_0, Register characteristics are:

Purpose Least significant word of the 128-bit random number.

Usage constraints There are no usage constraints.

Attributes See Table 3-161.

Table 3-162 shows the bit assignments.

Table 3-162 RNG Output Register (OUTPUT_0)

Bit	Name	Description
[31:0]	Output [31:0]	Least significant word of 128-bit word of random data. This field is only valid when the *Ready* bit in the *STATUS Register* on page 3-141 is set HIGH.

RNG Output, OUTPUT_1, Register

The RNG Output, OUTPUT_1, Register characteristics are:

Purpose Second word of the 128-bit random number.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Table 3-163 shows the bit assignments.

Table 3-163 RNG Output, OUTPUT_1, Register bit assignments

Bit	Name	Description
[31:0]	Output [63:32]	Second word of 128-bit word of random data. This field is only valid when the *Ready* bit in the *STATUS Register* on page 3-141 is set HIGH.

RNG Output, OUTPUT_2, Register

The RNG Output, OUTPUT_2, Register characteristics are:

Purpose Third word of the 128-bit random number.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Table 3-164 shows the bit assignments.

Table 3-164 RNG Output, OUTPUT_2, Register bit assignments

Bit	Name	Description
[31:0]	Output [95:64]	Third word of 128-bit word of random data. This field is only valid when the *Ready* bit in the *STATUS Register* on page 3-141 is set HIGH.

RNG Output, OUTPUT_3, Register

The RNG Output, OUTPUT_3, Register characteristics are:

Purpose Most significant word of the 128-bit random number.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Table 3-165 shows the bit assignments.

Table 3-165 RNG Output, OUTPUT_3, Register bit assignments

Bit	Name	Description
[31:0]	Output [127:96]	Final word of 128-bit word of random data. This field is only valid when the *Ready* bit in the *STATUS Register* is set HIGH.

STATUS Register

The Status, STATUS, Register characteristics are:

Purpose Indicates that a new 128-bit random number is available.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Figure 3-120 shows the bit assignments.

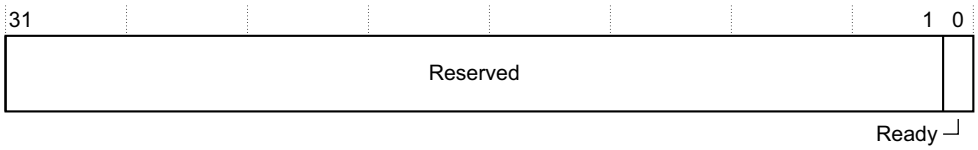


Figure 3-120 Status, STATUS, Register bit assignments

Table 3-166 shows the bit assignments.

Table 3-166 Status, STATUS, Register bit assignments

Bit	Name	Description
[31:1]	-	Reserved bits.
[0]	Ready	If set HIGH, data is available in the OUTPUT_0:3 registers. Acknowledging this state by setting bit [0] of this register HIGH resets the output value, and acknowledges the interrupt, if enabled through the *Interrupt Mask, INTMASK, Register*.

Interrupt Mask, INTMASK, Register

The Interrupt Mask, INTMASK, Register characteristics are:

Purpose If the mask register is set, the RNG generates an interrupt when the new random number is ready.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Figure 3-121 shows the bit assignments.

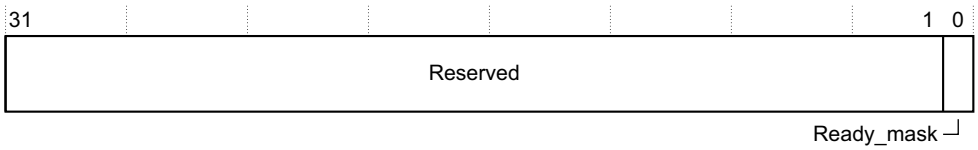


Figure 3-121 Interrupt Mask, INTMASK, Register bit assignments

Table 3-167 shows the bit assignments.

Table 3-167 Interrupt Mask, INTMASK, Register bit assignments

Bit	Name	Description
[31:1]	-	Reserved bits.
[0]	Ready_mask	If this bit is set HIGH, the interrupt line is asserted when the *ready* condition is true.

Configuration, CONFIG, Register

The Configuration, CONFIG, Register characteristics are:

Purpose Configures the RNG.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Table 3-168 shows the bit assignments.

Table 3-168 Configuration, CONFIG, Register bit assignments

Bit	Name	Description
[31:0]	Sample_clocks	This field programs the enable pulse width for the Entropy Source. The pulse width for the *Enable* control is equal to *sample_clocks* clock cycles. If you program it LOW, the enable is held asserted for one cycle. The pulse width limits are as follows: **Minimum** 1 clock cycle. **Maximum** (232 – 1) = 231 clock cycles. This field can only be written when the enable_trng bit in *CONTROL Register* is deasserted.

CONTROL Register

The Control, CONTROL, Register characteristics are:

Purpose Configures the RNG.

Usage constraints There are no usage constraints.

Attributes See Table 3-161 on page 3-139.

Figure 3-122 shows the bit assignments.

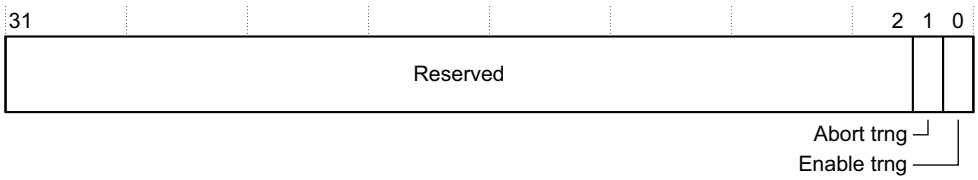


Figure 3-122 Control, CONTROL, Register bit assignments

Table 3-169 shows the bit assignments.

Table 3-169 Control, CONTROL, Register bit assignments

Bit	Name	Description
[31:2]	-	Reserved.
[1]	Abort_trng	Setting the bit to 1 aborts the entropy gathering tasks by deasserting the “Enable” control and resetting the output value registers. This bit automatically clears to 0.
[0]	Enable_trng	Setting this bit to 1 starts the TRNG gathering entropy from the entropy source. This bit automatically clears after deasserting the “Enable” control to the entropy source.

Note

This register is self-clearing.

Appendix A
Subcomponent Configurations

This chapter describes the IP configurations for the subcomponents of the *Juno r2 ARM Development Platform* (ADP). It contains the following sections:

- *Cortex-A72 processor cluster* on page A-2.
- *Cortex-A53 processor cluster* on page A-3.
- *Mali-T624 GPU* on page A-4.
- *CoreLink MMU-40x System Memory Management Unit (SMMU) components* on page A-5.
- *CoreLink DMC-400 Dynamic Memory Controller (DMC)* on page A-6.
- *CoreLink SMC-354 Static Memory Controller (SMC)* on page A-7.

A.1 Cortex-A72 processor cluster

Table A-1 shows the Cortex-A72 processor cluster configuration options.

Table A-1 Cortex-A72 cluster configuration options

Cortex-A72 cluster feature	Value
Number of cores	2
Cryptography engine	Included
L2 cache size	2MB
L2 logic idle gated clock	Included
ECC and parity support	ECC in L1 and L2

See the *ARM® Cortex®-A72 MPCore Processor Technical Reference Manual*.

A.2 Cortex-A53 processor cluster

Table A-2 shows the Cortex-A53 processor cluster configuration options.

Table A-2 Cortex-A53 cluster configuration options

Cortex-A53 cluster feature	Value
Number of cores	4
Cryptography engine	Included
Advanced SIMD and floating-point Extension	Included
L1 instruction cache size	32KB
L1 data cache size	32KB
L2 cache	Included
L2 cache size	1MB

See the *ARM® Cortex®-A53 MPCore Processor Technical Reference Manual*.

A.3 Mali-T624 GPU

Table A-3 shows the Mali-T624 GPU configuration options.

Table A-3 Mali-T624 GPU configuration options

Mali-T624 feature	Value
Number of shader cores	Four shader cores in one cluster
L2 cache size	256KB
L2 bus width	128
Supported texture formats	Programmable

See the *ARM® Mali™-T600 Series Implementation Guide*.

A.4 CoreLink MMU-40x *System Memory Management Unit* (SMMU) components

The ADP can contain the SMMUs that Figure 2-1 on page 2-2 shows.

Table A-4 shows the MMU-40x configuration options.

Table A-4 MMU-40x configuration options

Configuration options	Value						
	GPU	PCIe	ETR	DMA	USB	HDLCD0	HDLCD1
Component	MMU-400	MMU-401	MMU-401	MMU-401	MMU-401	MMU-401	MMU-401
APB type	APB4	APB4	APB4	APB4	APB4	APB4	APB4
AXI type	ACE-Lite + DVM	ACE-Lite + DVM	AXI3	AXI3	AXI3	AXI3	AXI3
AXI data width	128 bits	128 bits	64 bits	128 bits	128 bits	64 bits	64 bits
Number of contexts	1	4	1	4	1	1	1
Separate interface for PTW	No	No	No	No	No	No	No
Security determination, SSD table entries	1	1	1	9	1	1	1
Stream matching registers	2	32	2	8	2	2	2
Stream ID	1[a]	15[a]	1[a]	4[b]	1[a]	1[a]	1[a]
TLB depth	32	32	2	64	64	36	36
Write buffer depth	16	16	0	16	16	0	0

a. Stream ID value is 0.

b. Stream ID value is the channel number. In the DMA component, channel threads have a stream ID that is equal to the channel number. The manager thread has a stream ID of `0x8`.

In Table A-4, the stream ID value represents the width of the stream ID and not the value of the stream ID.

For the GPU, ETR, USB, HDLCD0, and HDLCD1 components, all transactions have a stream ID of 0.

For the DMA component, the value of the stream ID is the channel number. In the DMA component, channel threads have a stream ID that is equal to the channel number. The manager thread has a stream ID of `0x8`.

See *CoreLink MMU-401 and MMU-400 System Memory Management (SMMU) components* on page 2-18.

A.5 CoreLink DMC-400 *Dynamic Memory Controller* (DMC)

Table A-5 shows the DMC-400 configuration options for DDR3-1600.

Table A-5 DMC-400 configuration options for DDR3-1600

Configuration option	Value	Purpose
SYSTEM_INTERFACES	4	Number of system ACE4-Lite interfaces.
SYSTEM_ID_WIDTH	14	ACE interface ID signal width, for each system interface.
SYSTEM_ADDR_WIDTH	40	ACE address signal width, for each system interface.
SYSTEM_DATA_WIDTH	128	ACE data signal width, for each system interface.
SYSTEM_READ_ACCEPTANCE	64[a]	Outstanding read acceptance capability, for each system interface.
SYSTEM_READ_HAZARD_DEPTH	16[a]	Read hazard acceptance capability, for each system interface.
SYSTEM_READ_HAZARD_RAM	1	Read hazard buffer RAM implementation: 0 RTL models that you can synthesize. 1 Custom RAM models.
MEMORY_INTERFACES	2	Number of memory interfaces or channels.
MEMORY_DATA_WIDTH	128	Memory interface width, width of the DFI data buses.
MEMORY_CHIP_SELECTS	2	Number of chip-selects on a memory interface.
WRITE_BUFFER_DEPTH	32[a]	Depth of the write buffer, for each memory interface.
WRITE_BUFFER_RAM	1	Write buffer RAM implementation: 0 RTL models that you can synthesize. 1 Custom RAM models.
READ_QUEUE_DEPTH	32	Depth of the read queue, for each memory interface.
MAX_BURST_LENGTH	8	Maximum memory burst length, in beats.
ECC	0	*Single-Error-Correction, Double-Error-Detection, Error-Correcting-Code* (SECDED ECC) support: 0 Disabled. 1 Enabled.
QVN	1	Virtual networks.

a. Buffer and queue depths are specified in terms of the number of memory bursts.

A.6 CoreLink SMC-354 *Static Memory Controller* (SMC)

Table A-6 shows the PL354 configuration options.

Table A-6 PL354 configuration

Configuration option	Value
AXI width	64
MEMIF width	32
MEMIF CS	8
Exclusive monitors	4
CFIFO Depth	8
WFIFO Depth	16
RFIFO Depth	16
AID Width	14
Pipeline	True
ECC	False

Appendix B
Revisions

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

Table B-1 Issue A

Change	Location	Affects
First release.	-	-

Table B-2 Differences between issue A and issue B.a

Change	Location	Affects
Removed the *SCP controlled Level 2 cache maintenance* subsection.	*Application processors* on page 2-3.	r0p0
Removed the *System Control Processor (SCP) memory map* subsection.	*Memory* on page 2-7.	r0p0
Removed the following sections: • *Clocks and resets*. • *ADP SoC power-management*. • *ADP SoC power states* section. • *System Control Processor (SCP)*.	Chapter 2 *Functional Description*.	r0p0
Removed some information from *Time domains* section.	*Time domains* on page 2-29.	r0p0

Table B-2 Differences between issue A and issue B.a (continued)

Change	Location	Affects
Removed the following subsections: • *32kHz time domain*. • *Time domain relationship*. • *Restoring the AON_REF_CLK time domain*. • *Restoring the CoreSight timestamp time domain*. • *32kHz counter*. • *SCP AON_REF_CLK Generic Timer*. • *32kHz Generic Timer*.	*Time domains* on page 2-29.	r0p0
Removed some information from *Memory map frames* section.	*Memory map frames* on page 2-31.	r0p0
Removed some information from *Processor power modes* section.	*Processor power modes* on page 2-31.	r0p0
Removed the *SCP firmware watchdog* subsection.	*Watchdog timers* on page 2-31.	r0p0
Removed the following subsections: • *SCP debug architecture*. • *SCP CoreSight debug subsystem*. • *Debug power control*. • *Debug from reset*. • *System Profiler*. • *SCP AON_REF_CLK Generic Timer*. • *32kHz Generic Timer*.	*Debug and Profiling* on page 2-33.	r0p0
Removed the following sections: • *System Control Processor (SCP) interrupt map*. • *System Control Processor (SCP) memory map*. • *System Configuration Controller (SCC) registers*. • *ADP System Control Registers*. • *Power Policy Unit Registers*.	Chapter 3 *Programmers Model*.	r0p0
Removed the *Power and Access Control Sequences* appendix	-	r0p0

Table B-3 Differences between issue B.a and issue B.b

Change	Location	Affects
Added CCI-400 information to text and to ADP figure.	*CoreLink CCI-400 Cache Coherent Interconnect* on page 2-17. Figure 2-1 on page 2-2.	r0p0
Added RW1C to list of register access types.	*About this programmers model* on page 3-2.	r0p0

Table B-4 Differences between issue B.b and issue C

Change	Location	Affects
Updated the IP revisions.	*ARM IP revisions* on page 1-8.	r1p0
Updated the *ADP block diagram*.	Figure 2-1 on page 2-2.	r1p0
Updated the *ADP debug architecture, excluding timestamp distribution* diagram.	*ADP debug architecture, excluding timestamp distribution* on page 2-34.	r1p0

Table B-4 Differences between issue B.b and issue C (continued)

Change	Location	Affects
Added information to the *Cortex-A57 and Cortex-A53 processor debug architecture* section.	*Cortex-A72 and Cortex-A53 processor Debug architecture* on page 2-35.	r1p0
Added information to the *Trace* section.	*Trace* on page 2-36.	r1p0
Updated the *CoreSight debug and trace region memory map* diagram.	*CoreSight debug and trace region memory map* on page 3-15.	r1p0
Updated the *Application memory map* table.	*Application memory map summary* on page 3-21.	r1p0

Table B-5 Differences between issue C and issue D.a

Change	Location	Affects
Added information about the stream ID values for all components. In particular, the stream ID values for the PCIe and DMA components that have non-zero stream ID values.	*CoreLink MMU-40x System Memory Management Unit (SMMU) components* on page A-5.	r1p0

Table B-6 Differences between issue D.a and issue D.b

Change	Location	Affects
Removed incorrect footnotes, underneath Memory Management Unit configuration options table, that relate to non-zero stream ID values.	*CoreLink MMU-40x System Memory Management Unit (SMMU) components* on page A-5.	r1p0
Added CCI-400 information to text and to ADP figure.	*CoreLink CCI-400 Cache Coherent Interconnect* on page 2-17. Figure 2-1 on page 2-2.	r1p0
Added RW1C to list of register access types.	*About this programmers model* on page 3-2.	r1p0

Table B-7 Differences between issue D.b and issue E

Change	Location	Affects
Changed the big processor from a Cortex-A57 to a Cortex-A72.	Throughout the document.	r2p0

Table B-8 Differences between issue E and issue F

Change	Location	Affects
Removed incorrect footnotes, underneath Memory Management Unit configuration options table, that relate to non-zero stream ID values.	*CoreLink MMU-40x System Memory Management Unit (SMMU) components* on page A-5.	r2p0
Added CCI-400 information to text and to ADP figure.	*CoreLink CCI-400 Cache Coherent Interconnect* on page 2-17. Figure 2-1 on page 2-2.	r2p0
Added RW1C to list of register access types.	*About this programmers model* on page 3-2.	r2p0
Corrected interrupt table.	Table 3-3 on page 3-4.	r2p0