ARM® CoreLink™ CCN-502 Cache Coherent Network

Revision: r0p1

Technical Reference Manual

ARM

ARM® CoreLink™ CCN-502 Cache Coherent Network

Technical Reference Manual

Copyright © 2014, 2015, 2017 ARM Limited or its affiliates. All rights reserved.

Release Information

Document History

Issue	Date	Confidentiality	Change
0000-00	01 October 2014	Confidential	First release for r0p0
0000-01	06 February 2015	Confidential	Second release for r0p0
0000-02	05 May 2015	Confidential	Third release for r0p0
0001-00	29 August 2017	Non-Confidential	First release for r0p1

Confidentiality Status

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

Unrestricted Access is an ARM internal classification.

Product Status

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

Web Address

http://www.arm.com

Contents
ARM® CoreLink™ CCN-502 Cache Coherent Network Technical Reference Manual

Appendix B   Revisions

Preface

This preface introduces the *ARM® CoreLink™ CCN-502 Cache Coherent Network Technical Reference Manual*.

It contains the following:

About this book

This book is for the ARM® CoreLink™ CCN-502 Cache Coherent Network.

Product revision status

The r*m*p*n* identifier indicates the revision status of the product described in this book, for example, r*1*p*2*, where:

r*m*   Identifies the major revision of the product, for example, r1.

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

Intended audience

This book is written for system designers, system integrators, and programmers who are designing or programming a *System-on-Chip* (SoC) that uses the CCN-502.

Using this book

This book is organized into the following chapters:

Chapter 1 Introduction
    This chapter describes the CCN-502.

Chapter 2 Functional Description
    This chapter describes the functionality of the CCN-502.

Chapter 3 Programmers Model
    This chapter describes the programmers model.

Chapter 4 L3 Memory System
    This chapter describes the Level 3 memory system.

Chapter 5 Debug
    This chapter describes the debug features.

Chapter 6 Performance Optimization and Monitoring
    This chapter describes performance optimization techniques for use by system integrators, and the *Performance Monitoring Unit* (PMU).

Appendix A Signal Descriptions
    This appendix describes the external signals of the CCN-502 for a system that includes all possible CCN-502 components.

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

Glossary

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

See the *ARM® Glossary* for more information.

Conventions

Typographic conventions

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, IMPLEMENTATION SPECIFIC, UNKNOWN, and UNPREDICTABLE.

Timing diagrams

The following figure 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.
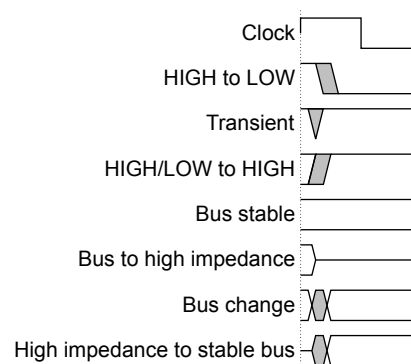
Figure 1 Key to timing diagram conventions

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.

Lowercase n

> At the start or end of a signal name denotes an active-LOW signal.

Additional reading

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

ARM publications

- *ARM® CoreLink™ CCN-502 Cache Coherent Network Configuration and Sign-off Guide* (ARM 100053).
- *ARM® CoreLink™ CCN-502 Cache Coherent Network Integration Manual* (ARM 100054).
- *ARM® AMBA® AXI and ACE Protocol Specification* (ARM IHI 0022).
- *ARM® AMBA® 5 CHI Architecture Specification* (ARM IHI 0050).
- *AMBA® Low Power Interface Specification, ARM® Q-Channel and P-Channel Interfaces* (ARM IHI 0068).
- *ARM® Architecture Reference Manual ARMv7-A and ARMv7-R Edition* (ARM DDI 0406).
- *ARM® Architecture Reference Manual ARMv8, for ARMv8-A architecture profile* (ARM DDI 0487).

Other publications

- *JEDEC Standard Manufacturer's Identification Code*, JEP106, *http://www.jedec.org*.

Feedback

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 *ARM CoreLink CCN-502 Cache Coherent Network Technical Reference Manual*.
• The number ARM 100052_0001_00_en.
• If applicable, the page number(s) to which your comments refer.
• A concise explanation of your comments.

ARM also welcomes general suggestions for additions and improvements.

———— **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 describes the CCN-502.

It contains the following sections:

1.1 About the CCN-502 Cache Coherent Network

The CCN-502 is a scalable coherent interconnect based on the AMBA 5 CHI architecture. It is designed for use in high-end networking and enterprise compute systems.

The CCN-502 combines interconnect and coherency functions into a single module. It provides the following external interfaces:

- Four fully coherent CHI ports for connection to CHI fully coherent devices.
- Two (6XP/2HNF configuration) or four (8XP/4HNF configuration) CHI slave-node ports or AXI4 master ports for connection to CHI or AXI4 memory controllers.
- One ACE-Lite/AXI4 master port for connection to the slave *Input/Output* (I/O) subsystem.
- Nine ACE-Lite/ACE-Lite+DVM/AXI4 slave ports for connection to the master I/O subsystems.

A system containing the CCN-502 contains the following protocol nodes:

Fully-coherent Requesting Node **(RN-F)**
> A fully-coherent master device.

I/O-coherent Requesting Node **(RN-I) bridge**
> An I/O-coherent master device. This is a native CHI bridge acting as an RN-I proxy for one or more non-native CHI devices located behind the RN-I bridge.

Fully-coherent Home Node **(HN-F)**
> A device that is a home node for a region of memory, accepting coherent requests from RN-Fs and RN-Is, and generating snoops to all applicable RN-Fs in the system as required to support the coherency protocol.

I/O Home Node **(HN-I)**
> A device that acts as a home-node for the slave I/O subsystem, mainly responsible for ensuring proper ordering of requests sent into the slave I/O subsystem.

Fully-coherent Slave Node **(SN-F)**
> A fully-coherent device that communicates with one or more HN-Fs that is solely a recipient of CHI commands, limited to fulfilling simple read and write commands.

Miscellaneous Node **(MN)**
> A device that is responsible for handling barriers, *Distributed Virtual Memory* (DVM) operations, configuration accesses, error reporting and signaling, interrupt generation, and debug-support features. The MN shares a single device port with the HN-I.

DVM Requesting Node **(RN-D)**
> An I/O-coherent master device that can accept DVM messages on the snoop channel.

Related concepts

2.1 About the functions on page 2-25.

1.2 Compliance

The CCN-502 implements the AMBA 5 CHI architecture and complies with the AMBA AXI4 and ACE protocol.

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

AMBA 5 CHI architecture

The CCN-502 implements the AMBA 5 CHI architecture. This architecture provides the following capabilities:

- High-performance coherence protocol.
- Packet-based communication.
- Four channels:
  — *Request* (REQ).
  — *Response* (RSP).
  — *Snoop* (SNP).
  — *Data* (DAT).
- Wire and buffer scalability to provide the required bandwidth and storage in coherent systems.
- Credited end-to-end protocol-layer flow-control with retry-once mechanism for flexible bandwidth and resource allocation.
- Integrated end-to-end *Quality-of-Service* (QoS) capabilities.

See the *ARM® AMBA® 5 CHI Architecture Specification* for more information.

AMBA AXI4 and ACE protocol

The CCN-502 complies with the AMBA AXI4 and ACE protocol. See the *ARM® AMBA® AXI and ACE Protocol Specification* for more information.

1.3   Features

This section describes key CCN-502 features.

The CCN-502 provides the following key features:
- Dual simplex ring-bus interconnect topology consisting of 6 or 8 crosspoints, with each crosspoint supporting up to two device ports.
- Support for up to 4 fully coherent processor compute clusters.
- Support for up to four memory controllers.
- Support for up to 9 ACE-Lite/ACE-Lite+DVM/AXI4 I/O master devices. Additional devices can be supported using an additional level of interconnect hierarchy, such as the CoreLink NIC-400 Network Interconnect.
- Byte-level odd parity protection on all datapaths.
- Byte-level odd parity generation for data produced by the coherent compute clusters and the L3 cache.
- Broadcast snoop channel.
- DVM message transport between masters.
- QoS regulation for shaping traffic profiles.
- A *Performance Monitoring Unit* (PMU) to count performance-related events.
- High-performance distributed system cache, 0KB, 256KB, 512KB, 1MB, 2MB, 4MB, or 8MB in capacity, consisting of two or four partitions, each 0KB, 128KB, 512KB, 1MB, or 2MB in capacity. The system cache includes an integrated *Point-of-Serialization* (PoS) and *Point-of-Coherency* (PoC) and can be used both for compute and I/O caching.
- *Snoop filter* (SF) capable of covering 1MB, 2MB, 4MB, 8MB, or 16MB of 64-byte cache-line tags for increased coherency scalability. The SF consists of two or four partitions, each covering either 512KB, 2MB, or 4MB of 64-byte cache-line tags.
- One I/O Home Node with an ACE-Lite or AXI4 master port.
- Error signal gathering using an error bus, with a single point of interrupt coordination on errors.
- 40-bit physical address support
- *On-Chip Memory* (OCM), which allows for systems without physical memory. The CCN-502 does not access the SN-F, under specific use cases.

1.4 Interfaces

The following figure shows the interfaces of the CCN-502.
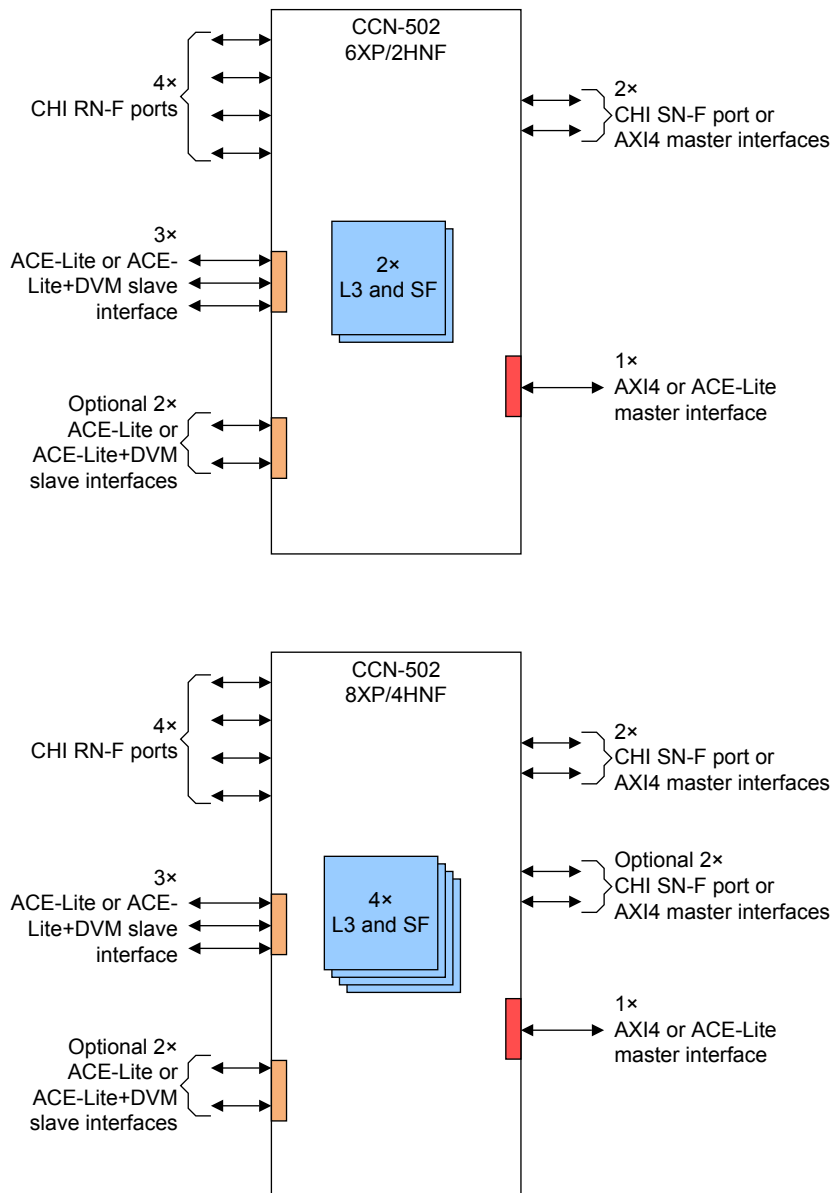
Figure 1-1 CCN-502 interfaces

1.5 Configurable options

This section describes CCN-502 configurable options.

This section contains the following subsections:

1.5.1 Configurable parameters

This section lists the parameters that are configurable at build time.

Table 1-1 Configurable parameters

Component	Feature	Options	Comments
External interfaces	Number of ACE-Lite/ACE-Lite+DVM slave ports for connection to the master I/O subsystems	3, 6, or 9.	Up to two RN-I bridges, that is, any two, can be depopulated. The number of AMBA interfaces per RN-I bridge is fixed at three.
	RN-I bridge interface type	ACE-Lite or ACE-Lite+DVM.	All interfaces for any particular RN-I bridge must be identical.
Build-time component population or depopulation	Number of RN-I bridges	1, 2, or 3.	Up to two RN-I bridges, that is, any two, can be depopulated.
	CHI to AXI protocol bridges (SBSX)	Present or not present.	Populated as a group, that is, all or none. SBSX population determines the SN-F interface type, either CHI or AXI4.
XP	End-to-end ring parity protection	Present or not present.	Parity protection from ingress to egress of the ring.
HN-F/L3	L3 cache capacity	0KB, 128KB, 512KB, 1MB, or 2MB per partition.	All L3s must be configured identically.
	Snoop filter capacity	512KB, 2MB, or 4MB per partition.	All snoop filters must be configured identically.
	L3 tag/data/SF RAM latency	2 or 3 cycles.	All tag, data, and snoop filter RAMs in all HN-Fs have identical latency.
SBSX	Data width on SBSX AMBA interface	128-bit or 256-bit data.	Data width is determined by **SBSX_128_n256** input pin.
MC/SN-F	Number of memory controllers	2 or 4.	Up to two SN-F interfaces can be depopulated. This parameter applies only to 8XP/4HNF configuration.
REQ Channel	RSVDC field width	4 or 8 bits.	On the REQ channel, the RSVDC field width can be set to 4 or 8 bits.

Table 1-1 Configurable parameters (continued)

Component	Feature	Options	Comments
Clocking and timing	XP to device clocking	Synchronous 1:1, Asynchronous.	*Device to XP asynchronous Bridge* (DSSB) is not supported in combination with protocol bridge devices or with HN-F/L3 components.
	DSSB population	Present or not present at group granularity, RN-F or SN.	DSSB or *Device Register Slice* (DRS) usage is mutually exclusive.
	DSSB FIFO depth	8, 10, or 12.	Allows for 1, 2, or 3 cycles of latency between the CCN-502 and the CCN502_RNF_DSSB and CCN502_SNF_DSSB blocks.
	AMBA interface clocking	N:1 synchronous, where N is 1-4.	AMBA interface runs at the same or a lower frequency than the CCN-502.
	STMHWEVENT interface clocking	N:1 synchronous, where N is 2-4.	The **STMHWEVENT** interface runs at a lower frequency than the CCN-502.
	Number of XP to DRS	0, 1, or 2.	DRS or DSSB usage is mutually exclusive. DRS can otherwise be present at any device interface.

Related concepts

1.5.2   Static parameters

This section lists the static parameters.

Table 1-2 Static parameters

Component	Feature	Value	Comments
External interfaces	AXI4/ACE-Lite master port for connection to the slave I/O subsystem.	1	-
XP	Ring data-channel width.	128 bits	-
	Number of ring data channels, each dual simplex.	1	-
	Device data-channel width.	128 bits	-
	Number of crosspoints.	6 or 8	6 for 6XP/2HNF configuration, or 8 for 8XP/4HNF configuration.
	Number of devices ports.	2	-
	Number of devices per device port.	1	-
HN-F/L3	Number of L3 partitions.	2 (6XP/2HNF) or 4 (8XP/4HNF)	2 for 6XP/2HNF configuration, or 4 for 8XP/4HNF configuration.
	L3 CHI interface data-channel width.	128 bits	-
	Number of data channels.	1	-
Bridges	Maximum number of RN-I bridges.	3	-
	Number of HN-I bridges.	1	-

Table 1-2 Static parameters (continued)

Component	Feature	Value	Comments
RN-I bridge	Number of outstanding reads.	32	-
	Number of outstanding writes.	8	-
	Number of ACE-Lite+DVM ports	3	-
	Device CHI data-channel width.	128 bits	-
	Device AXI data width.	128 bits	-
	AxID width.	11	-
	AxUser width.	4 or 8	Based on the setting of RSVDC field width, **AxUser** width is set to 4 bits or 8 bits.
SBSX	Number of outstanding transactions, either reads or writes.	32	-
	Device CHI data-channel width.	128 bits	-
	AxID width.	11	-
	AxUser width.	4 or 8	Based on the setting of RSVDC field width, **AxUser** width is set to 4 bits or 8 bits.
HN-I	Number of outstanding transactions, either reads or writes.	16	Maximum of 15 reads or 16 writes.
	Device CHI data-channel width.	128 bits	-
	Device AXI data width.	128 bits	-
	AxID width.	11	-
	AxUser width.	4 or 8	Based on the setting of RSVDC field width, **AxUser** width is set to 4 bits or 8 bits.

1.5.3   Tie-off signals

The CCN-502 provides input tie-off signals whose state at reset defines the behavior of the network.

The input tie-off signals must be stable when reset deasserts and they must remain stable through operation. These signals define the behavior of the CCN-502.

Related references

1.6 Test features

This section describes CCN-502 test features.

See the *ARM® CoreLink™ CCN-502 Cache Coherent Network Integration Manual* for information about the test features.

1.7   Product documentation and design flow

This section describes the CCN-502 books and how they relate to the design flow.

Documentation

The CCN-502 documentation is as follows:

Technical Reference Manual

The *Technical Reference Manual* (TRM) describes the functionality and the effects of functional options on the behavior of the CCN-502. It is required at all stages of the design flow. The choices you make in the design flow can mean that some behavior described in the TRM is not relevant. If you are programming the CCN-502 then contact:
- The implementer to determine:
  — The build configuration of the implementation.
  — What integration, if any, was performed before implementing the CCN-502.
- The integrator to determine the pin configuration of the device that you are using.

Configuration and Sign-off Guide

The *Configuration and Sign-off Guide* (CSG) describes:
- The available build configuration options and related issues in selecting them.
- How to configure the *Register Transfer Level* (RTL) with the build configuration options.
- How to integrate RAM arrays.
- How to run test patterns.
- The processes to sign off the configured design.

The ARM product deliverables include reference scripts and information about using them to implement your design. Reference methodology flows supplied by ARM are example reference implementations. Contact your EDA vendor for EDA tool support.

The CSG is a confidential book that is only available to licensees.

Integration Manual

The *Integration Manual* (IM) describes how to integrate the CCN-502 into an SoC. It includes a description of the pins that the integrator must tie off to configure the macrocell for the required integration. Some of the integration is affected by the configuration options used when implementing the CCN-502.

The IM is a confidential book that is only available to licensees.

Design flow

The CCN-502 is delivered as synthesizable RTL. Before it can be used in a product, it must go through the following processes:

Implementation

The implementer configures and synthesizes the RTL to produce a hard macrocell. This includes integrating RAMs into the design.

Integration

The integrator connects the implemented design into an SoC. This includes connecting it to a memory system and peripherals.

Programming

This is the last process. The system programmer develops the software required to configure and initialize the CCN-502, and tests the required application software.

Each process:

- Can be performed by a different party.
- Can include implementation and integration choices that affect the behavior and features of the CCN-502.

The operation of the final device depends on:

Build configuration

The implementer chooses the options that affect how the RTL source files are pre-processed. These options usually include or exclude logic that affects one or more of the area, maximum frequency, and features of the resulting macrocell.

Configuration inputs

The integrator configures some features of the CCN-502 by tying inputs to specific values. These configurations affect the start-up behavior before any software configuration is made. They can also limit the options available to the software.

Software configuration

The programmer configures the CCN-502 by programming particular values into registers. This affects the behavior of the CCN-502.

——————— Note ———————

This manual refers to implementation-defined features that are applicable to build configuration options. Reference to a feature that is included means that the appropriate build and pin configuration options are selected. Reference to an enabled feature means one that has also been configured by software.

—————————————

Related concepts

1.2 Compliance on page 1-14.

Related references

Additional reading on page 10.

1.8 Product revisions

This section describes the differences in functionality between successive product revisions of the CCN-502.

r0p0

First release.

r0p1

There are no functional changes in this release.

Chapter 2
Functional Description

This chapter describes the functionality of the CCN-502.

It contains the following sections:

2.1 About the functions

This section describes the functional blocks in the CCN-502.

The CCN-502 combines interconnect and coherency functions into a single module. It supports connectivity for up to 4 CHI masters, one AXI4/ACE-Lite slave, up to 9 AXI4/ACE-Lite/ACE-Lite +DVM masters, plus optional *Distributed Virtual Memory* (DVM) message support on these interfaces to manage distributed *Memory Management Unit*s (MMUs).

The following figure shows a block-level view of an example configuration of the CCN-502.
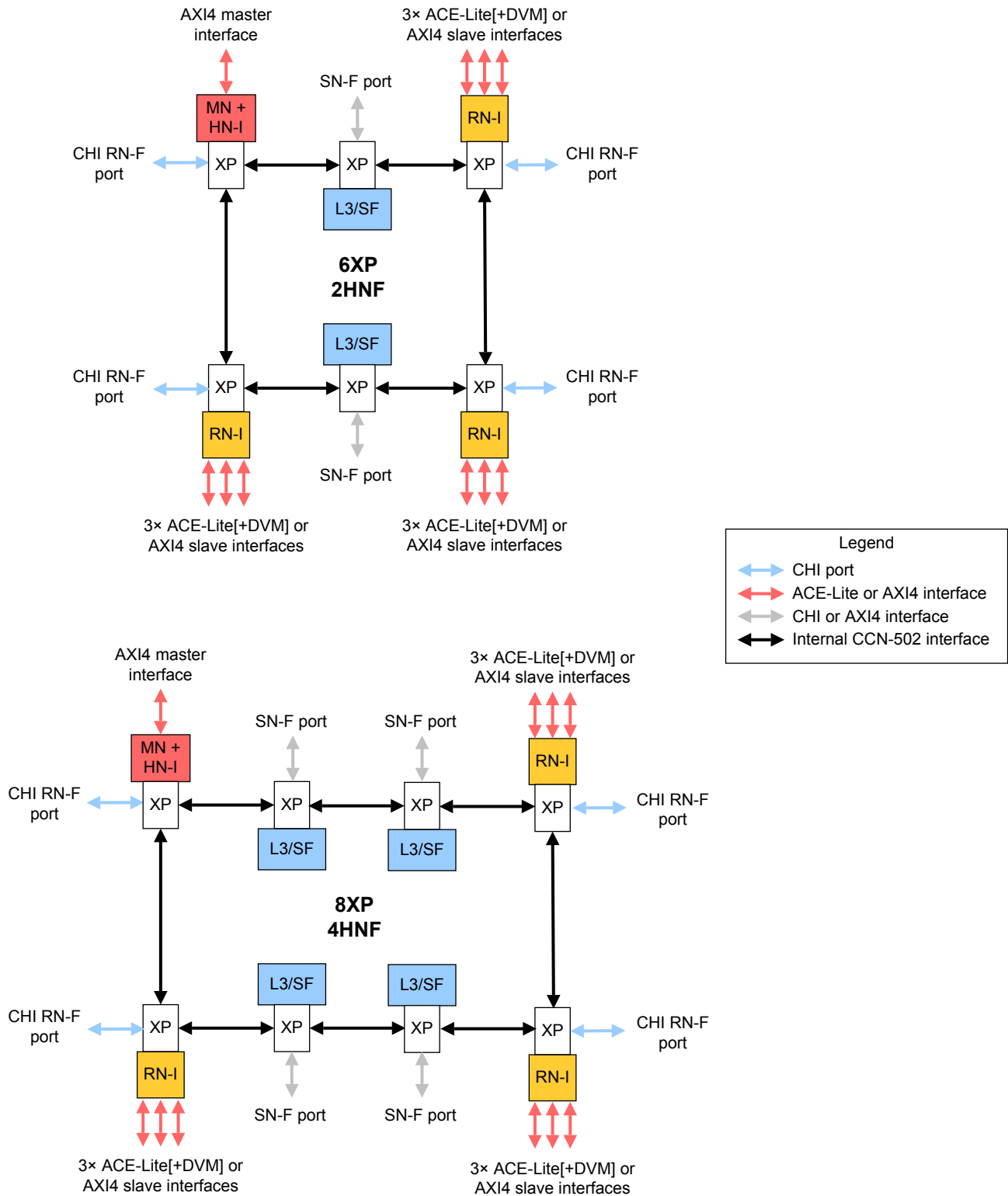
Figure 2-1 CCN-502 block diagram

A complete SoC system includes many devices. This section only describes the devices that are deliverables in the CCN-502 product.

This section contains the following subsections:

2.1.1 Crosspoint

A *crosspoint* (XP) is a switch and router logic unit that includes two interconnect ports and two device ports, and is the fundamental component of the CCN-502 transport mechanism.

A collection of XPs arranged in a dual-simplex ring topology provides all the packet transport capability for a CCN-502 system.

Each XP includes transport and routing capabilities for each of the four channels:
- *Request* (REQ).
- *Response* (RSP).
- *Snoop* (SNP).
- *Data* (DAT).

Because the XP is the entry and exit point for the CCN-502, it includes an optional DSSB for XP to device communication. The inclusion of DSSB in the XP to device communication path is optional depending on the clocking requirements of specific devices that are attached to an XP.

2.1.2 Fully-coherent Home Node

The *Fully-coherent Home Node* (HN-F) is responsible for managing part of the address space.

The HN-F consists of the following:

L3 cache   The L3 cache is a distributed, mostly exclusive last-level cache. The L3 cache-allocation policy is exclusive for data lines, except where sharing patterns are detected, and pseudo-inclusive for code lines, as controlled by the RN-Fs, meaning that all code lines can be allocated into the L3 on the initial request.

Combined PoS/PoC   The combined *Point-of-Serialization/Point-of-Coherency* (PoS/PoC) is responsible for the ordering of all memory requests sent to the HN-F. This includes coherency ordering, that is, serialization of multiple outstanding requests and actions to the same line, and request ordering as required by the RNs.

Snoop filter   The snoop filter reduces snoop coherency traffic in the system by tracking cache lines that are present in the RN-Fs in the system, and generally converting snoop broadcasts to directed snoops. This substantially reduces the quadratic growth in snoop response traffic that might otherwise be required without the snoop filter.

Each HN-F in the system is configured to manage a specific portion of the overall address space, and all three functionalities are included in this management responsibility:

- The L3 in an HN-F caches only data from the addresses that are assigned to that HN-F.
- The PoS/PoC manages ordering and coherency only for the addresses that are assigned to the HN-F.
- The snoop filter tracks RN-F caching for the addresses that are assigned to the HN-F.

Therefore, the entire memory address space is managed through the combination of all HN-Fs in the system, except for the memory-mapped I/O address space.

───────── **Note** ─────────

The HN-F is architecturally defined to manage only well-behaved memory, that is, memory without any possible side effects. The HN-F includes microarchitectural optimizations to exploit this architectural guarantee.

─────────────────────

2.1.3 I/O-coherent Requesting Node bridge

The *I/O-coherent Requesting Node* (RN-I) bridge connects I/O-coherent AMBA masters to the CCN-502 system.

An RN-I bridge includes:
- Up to three ACE-Lite/ACE-Lite+DVM slave ports.
- A single CHI RN-I interface.

The RN-I bridge can act as a proxy only for masters that do not contain hardware-coherent caches, because there is no capability to extend system coherency through the RN-I bridge.

2.1.4 I/O Home Node

The *I/O Home Nodes* (HN-I) are home-nodes for all CHI transactions that target AMBA slave devices. They act as proxies for all the RNs of the CCN-502, converting CHI transactions to ACE-Lite or AXI4 transactions.

They include support for the correct ordering of ARM device types and can optionally broadcast *Data Synchronization Barriers* (DSBs) and *Data Memory Barriers* (DMBs) into the slave I/O subsystem.

The HN-Is do not support coherent caching of any data read from or written to the downstream AMBA I/O slave subsystem. This means that any cacheable request sent to the HN-Is do not result in any snoops being sent to RN-Fs in the system, but are instead converted to the appropriate AXI read or write command and sent to the downstream AMBA subsystem. If an RN-F does cache data read from or written to the downstream AMBA I/O slave subsystem, coherency is not maintained, and any subsequent access to that data reads from or writes to the AMBA I/O slave subsystem directly, ignoring the cached data.

2.1.5 CHI to AXI bridge

The *CHI to AXI bridge* (SBSX) enables an AXI4 slave device such as a CoreLink DMC-400 Dynamic Memory Controller, to be used as an SN-F in a CCN-502 system.

2.1.6 Miscellaneous Node

The *Miscellaneous Node* (MN) is responsible for handling the following features:

- Barriers.
- DVM operations.
- Configuration accesses.
- Error reporting and signaling.
- Interrupt generation.
- Centralized debug and *Performance Monitoring Unit* (PMU) support features.

Barrier or DVM transactions that are handled by the MN do not target the memory address space or the I/O address space. Each of these must be multicast or broadcast to peer RNs or HNs, and responses must be aggregated.

Although the MN is responsible for broadcasting and gathering responses, the requesting node sends a single request to the MN and receives a single completion response from the MN.

The MN includes the following dedicated ports:
- A CHI port for inbound and outbound communication of the MN-supported CHI commands.
- Ports to collect error signals from CHI components within the CCN-502.
- A configuration bus that connects to all of the nodes, to transfer the reads and writes of the internal configuration registers.
- An interrupt request output, **INTREQ**, that is asserted on errors or performance monitor event counter overflow.

2.1.7 Power/Clock Control Block

The *Power/Clock Control Block* (PCCB) provides separate communication channels to pass information about the power and clock management between the SoC and the network.

The PCCB acts as an aggregator to convey information between the SoC and the other CCN-502 components, in the following manner:
- The PCCB receives transaction activity indicators from other relevant CCN-502 components and conveys that information to the external power and clock control units.
- When the PCCB receives a power or clock control management request from the external power or clock control units, it conveys that request to the relevant CCN-502 components, where applicable.
- The PCCB waits for the appropriate responses from the relevant CCN-502 components, and conveys an aggregated response to the external power and clock control units.

2.1.8 System Address Map overview

This section describes the System Address Map overview.

All CHI commands must include a fully resolved network address, that is, the address must include a source and target ID. For originating requests, this is achieved by passing a request address through a *System Address Map* (SAM), which effectively maps a memory or I/O address to the target device to satisfy that request. Because this conversion must be complete for a request to be valid, the SAM functionality is integrated in each requesting device, and cannot exist as a separate in-network device.

The SAM consists of two logical units:
- An RN-SAM in each RN to map from an address to HN-F, HN-I, and MN target IDs.
- An HN-F SAM in the HN-F, that maps from an address to a *Memory Controller* (MC) target ID.

The RN-SAM functionality is integrated into RN devices and is not a separate CHI-based unit.

Related concepts
2.12 System Address Map on page 2-53.

2.1.9 Debug Event Module overview

The *Debug Event Module* (DEM) provides debug, trace, and PMU capability.

The DEM is included in the MN and is responsible for the following operations:
- Aggregating watchpoint trigger events from all *Debug Watchpoint Module*s (DWMs) that are included in all crosspoints, and optionally asserting the **DBGWATCHTRIGREQ** signal for a trigger event.
- Time-aligning these trigger events.
- Communicating these trigger events to a CoreSight *System Trace Module* (STM).
- Counting PMU events that are communicated by other components in the CCN-502.
- Requesting assertion of the **INTREQ** interrupt signal on overflow of PMU counters.

Related references
5.4 Debug Event Module on page 5-236.

2.1.10 QoS regulator

This section describes the QoS regulator.

The CCN-502 supports end-to-end *Quality-of-Service* (QoS) guarantees using QoS mechanisms distributed throughout the system. The QoS provision uses the QoS field in each RN request packet to influence arbitration priority at every QoS decision point. The QoS field is then propagated through all secondary packets issued by a request packet. RNs must either self-modulate their QoS priority depending on how well their respective QoS requirements are being met, or make use of the integrated QoS regulators at ingress points to the CCN-502.

It is possible to include non-QoS-aware devices in the system, but still have these devices meet the QoS modulation requirement of the QoS architecture. To enable this, the CCN-502 includes inline regulators that perform the QoS functionality without the requesting device requiring any awareness of QoS. A *QoS Regulator* (QR) provides an interstitial layer between an RN and the interconnect. The QR monitors how the bandwidth and latency requirements of the RN are met, and does in-line replacement of the RN-provided QoS field, adjusting upwards to gain additional priority in the system, and downwards to reduce priority.

Related concepts
2.5 Quality of Service on page 2-36.

2.1.11 Optional components

Device-to-XP bridges (DSSBs) and register slice are optional components that can be added at XPs.

DSSB and register slice usage is mutually exclusive.

Device/XP Source-Synchronous Asynchronous Bridge

The DSSB is an optional component that provides an asynchronous bridge between the device and an XP, to allow different power, clock, or voltage domains.

The DSSB also allows the interfaces between the device and XP to be clocked source-synchronously.

Register slices

Register slices are optional components that you can insert at any given XP to device interface to assist in timing closure in a CCN-502 system. This enables synchronous but higher latency communication at any point in the system.

Related concepts
Timing closure with register slices on page 2-66.

Related references
1.5.1 Configurable parameters on page 1-17.

2.2 System configurations

This section shows examples of how you can configure a CCN-502 system.

shows interconnects that include only the components that are deliverables in the CCN-502 product. The following two figures show interconnects that include additional optional components that are not deliverables in the CCN-502 product.

shows a CCN-502 interconnect that includes an optional SBSX protocol bridge.
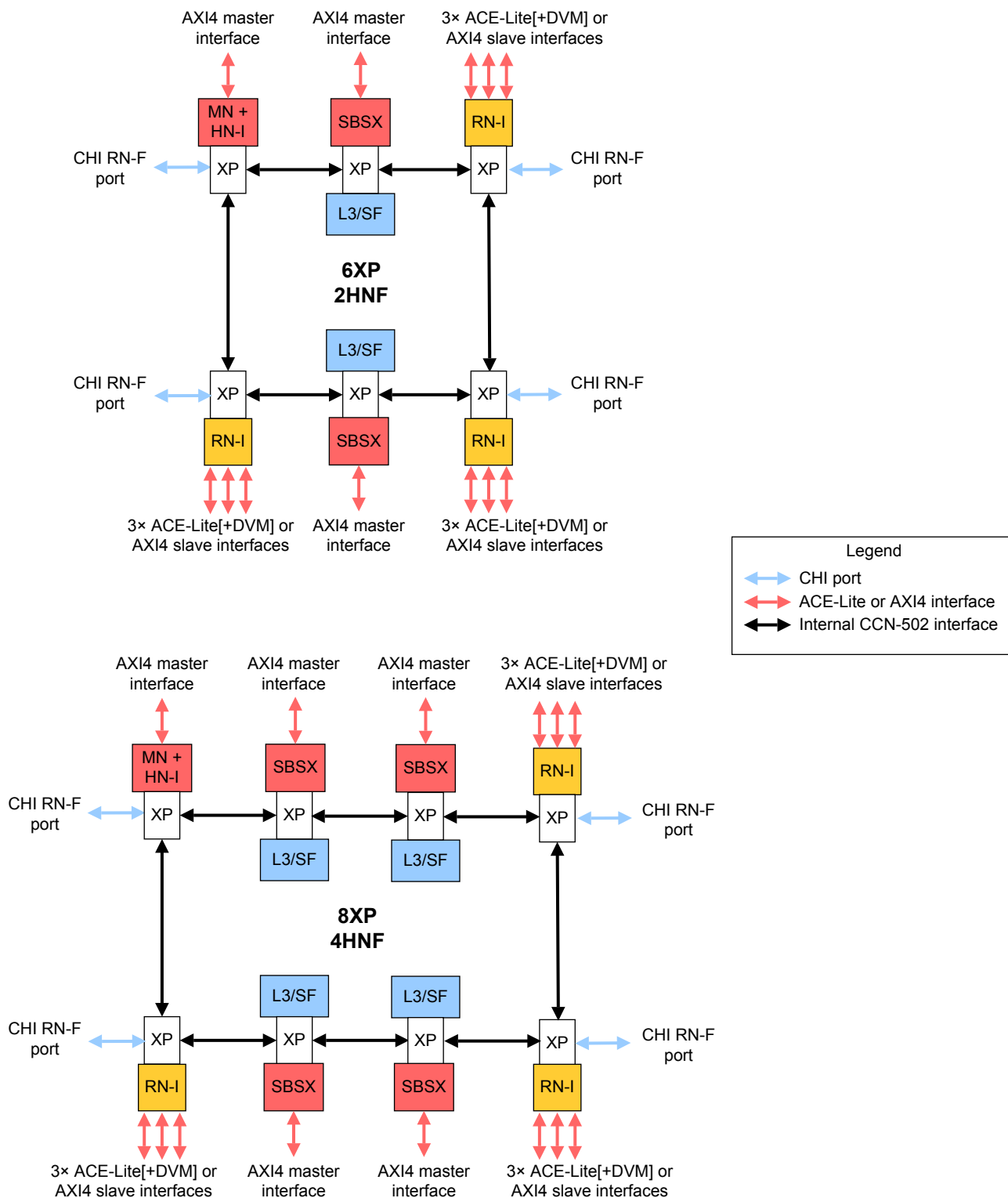
Figure 2-2 Interconnect with optional SBSX protocol bridges

Figure 2-3 Interconnect with memory controller and processor on page 2-33 shows an example of a CCN-502 system, including optional devices such as a processor and *Memory Controller* (MC), to create a complete coherent subsystem. The interconnect in this baseline system does not include the optional protocol bridges, and therefore requires processor clusters and *Dynamic Memory Controller*s (DMCs) that include native CHI interfaces, such as the Cortex®-A57 MPCore multiprocessor and the CoreLink DMC-520 memory controller, respectively.
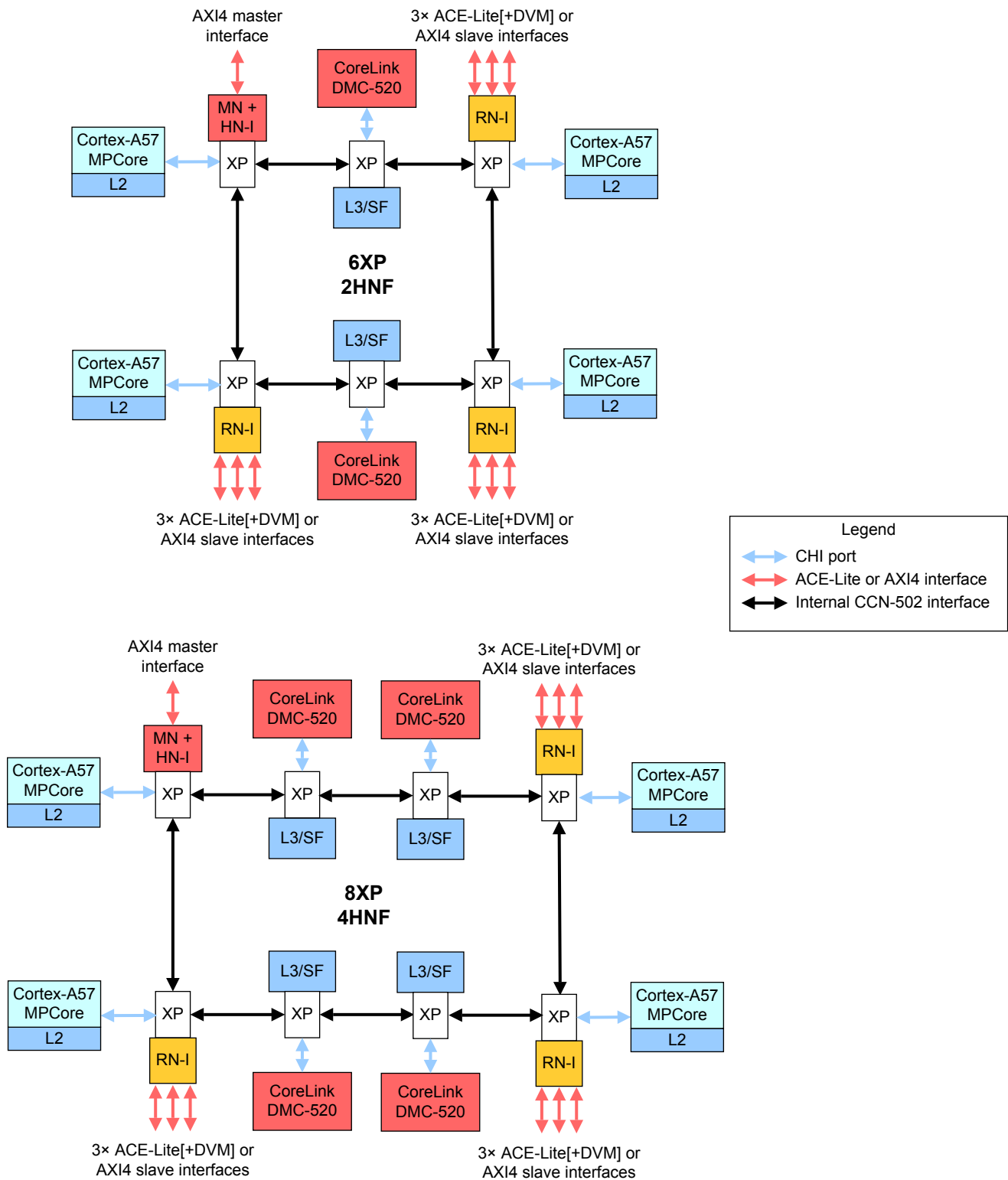
Figure 2-3 Interconnect with memory controller and processor

2.3 Addressing capabilities

This section describes CCN-502 addressing capabilities.

The CCN-502 includes a 44-bit physical address capability on several interfaces. However, this capability exists primarily to enable DVM messages compliant with ARMv8-A TLB maintenance operations and to support potential address-space expansion in other members of the Cache Coherent Network product line. An SoC designer using the CCN-502 is not expected to make use of more than 40 bits of the physical address space.

2.4 Exclusive accesses

The CCN-502 supports exclusive accesses to both Shareable and Non-shareable locations as the CHI architecture describes.

2.4.1 HN-F exclusive accesses

The HN-F supports exclusive access on ReadNoSnp, WriteNoSnp, ReadShared, ReadClean, and CleanUnique transactions to any address that maps to the HN-F.

————— **Note** —————

The RN-F generates:

*   ReadNoSnp and WriteNoSnp exclusives for memory locations that are marked Non-cacheable or Device.
*   ReadShared, ReadClean, and CleanUnique exclusives are used for Shareable and coherent memory locations.

—————————————

Each HN-F partition includes 32 exclusive monitors for tracking these transaction types, and each monitor can operate as both a PoC monitor and System monitor, as defined by the CHI architecture.

For each of the 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-F partitions, only 32 unique logical threads, either processor or device threads, designated by a unique SrcID or LPID, can access the exclusive monitors of that HN-F during one instance of the CCN-502 operation, from reset deassertion to reset assertion. This is because these monitors are permanently assigned to a logical thread on the first exclusive access by that thread, and are not then available to any other thread.

2.4.2 HN-I exclusive accesses

The HN-Is support exclusive access on ReadNoSnp and WriteNoSnp transactions to any address that maps to an HN-I.

Each HN-I includes 16 system monitors as defined in the CHI architecture for tracking of these transaction types. Only 16 unique logical threads, either processor or device threads, designated by a unique combination of SrcID and LPID, can concurrently access the HN-I system exclusive monitors.

If 16 unique threads are allocated a monitor with a ReadNoSnp exclusive to an HN-I, and those monitors have not yet been deallocated by a subsequent WriteNoSnp exclusive, then if another unique thread tries to allocate a monitor with a ReadNoSnp exclusive transaction, that transaction completes with an indication that the monitor was not successfully allocated. This prevents that thread from making forward progress in completing its exclusive access.

All exclusives targeting an HN-I are terminated at the HN-I and are not propagated downstream, regardless of the value of the HN-I PoS Control register and Auxiliary Control register.

Related references

PoS Control register on page 3-165.
SA Auxiliary Control register, HN-I on page 3-169.

2.5 Quality of Service

The CCN-502 includes end-to-end *Quality of Service* (QoS) capabilities that are designed to support the latency and bandwidth requirements of different types of devices.

The QoS device classes are:

Devices with bounded latency requirements

These are primarily real-time or isochronous devices that require some or all of their transactions be complete within a specific time period to meet overall system requirements. These devices are typically highly latency-tolerant within the bounds of their maximum latency requirement. Examples of this class of device include networking I/O devices and display devices.

Latency-sensitive devices

These are devices whose performance is highly impacted by the response latency incurred by their transactions. Processors are traditionally highly latency-sensitive devices, although a processor can also be a bandwidth-sensitive device depending on its workload.

Bandwidth-sensitive devices

These are devices that have a minimum bandwidth requirement to meet system requirements. An example of this class of device is a video codec engine, which requires a minimum bandwidth to sustain real-time video encode and decode throughput.

Bandwidth-hungry devices

These are devices that have significant bandwidth requirements and can use as much system bandwidth as is made available, to the limits of the system. These devices determine the overall scalability limits of a system, with the devices and system scaling until all available bandwidth is consumed.

——————— Note ———————

A device might fit into more than one of these classes, depending on its requirements at any time in its workload.

————————————————

Support for these different types of devices and their resulting traffic is included in the CHI architecture and in the entirety of the CCN-502 microarchitecture. Each component in the CCN-502 contributes to the overall QoS microarchitecture.

This section contains the following subsections:

- *2.5.1 Architectural QoS support* on page 2-36.
- *2.5.2 Microarchitectural QoS support* on page 2-36.

2.5.1 Architectural QoS support

The CHI architecture specifies that all message flits include a 4-bit *QoS Priority Value* (QPV).

The QPV of the originating message must propagate through all messages in a transaction. The QPV is defined as higher values being higher priority and lower values being lower priority. All CCN-502 components use the QPV to provide prioritized arbitration and to prevent head-of-line-blocking based on the QPV.

2.5.2 Microarchitectural QoS support

The CCN-502 provides QoS support.

The following subsections describe the QoS support that the CCN-502 components provide:

- *QoS regulators* on page 2-37.
- *QoS regulator operation* on page 2-37.
- *Ring/XP QoS support* on page 2-39.
- *HN-F QoS support* on page 2-39.

QoS regulators

This section describes QoS regulators.

The QPV of RN requests must be modulated depending on how the respective QoS requirements are met. Although the QoS-modulation capability can be integrated into the RN, the CCN-502 enables system designers to include non-QoS-aware devices in the CCN-502 system, but still have these devices meet the QoS-modulation requirements of the CCN-502 QoS microarchitecture.

The CCN-502 includes inline QoS regulators that perform QoS modulation without requiring any QoS-awareness by the requesting device. A QoS regulator introduces an interstitial layer between an RN and the interconnect that monitors whether the bandwidth and latency requirements of the RN are being met. It also performs in-line replacement of the RN-provided QPV field as required, adjusting upwards to increase priority or downwards to reduce priority in the system.

The QoS regulators are present at all entry points into the CCN-502:

- For CHI ports, the regulator is present in the XP.
- For AMBA slave interfaces, the regulator is present at the AMBA side of the protocol bridge.

Therefore, for AMBA interfaces there are two QoS regulators:

- One in the XP at the CHI side of the protocol bridge.
- One in the protocol bridge at the AMBA interface.

For AMBA interfaces, the XP QoS regulator must be configured to operate in a pass-through mode, so that only the AMBA-side regulator performs active regulation.

The CCN-502 QoS regulators have three operating modes, controlled through memory-mapped configuration registers:
1. Pass-through.
2. Programmed QoS value.
3. Regulation.

Related concepts

QoS regulator operation

The values of the base QPV, **AxQOS** for AMBA interfaces or REQ.QOS for CHI ports, are input to the QoS sub-block. When latency regulation or period regulation is enabled, these values are replaced by the **AxQOS** or REQ.QOS values generated by the regulators. For CHI RN-Fs, the one QoS regulator monitors read-type CHI transactions, and the resultant QPV is applied to all CHI requests. For RN-Is, separate QoS regulators exist for AR and AW channels.

The QoS regulators can operate in either latency regulation mode or period regulation mode. The registers to configure the QoS regulators exist in each RN-I and XP.

The following sections describe operating modes for slave interface S0 in the RN-I.

Latency regulation mode

When configured for latency regulation, the QoS regulator increases the QPV whenever actual latency is higher than the target, and decreases the QPV when it is lower:

- For every cycle that the latency of a transaction is more than the target latency, the QPV is increased by a fractional amount, the scale factor K_i.
- For every cycle that latency of a transaction is less than the target latency, the QPV is decreased by the same fractional amount, the scale factor K_i.

The Port 0 QoS Latency Target register specifies the target latency in cycles.

The Port 0 QoS Latency Scale register specifies the scale factor K_i. It is coded in powers of two, so that a programmed value of `0x0` = 2^{-12} and a programmed value of `0x7` = 2^{-5}.

You can program the QoS regulator to operate in latency regulation mode by programming the following bits in the Port 0 QoS Control register:

- Set the s0_ar_qos_override_en bit to 1.
- Set the s0_ar_lat_en bit to 1.
- Set the s0_ar_reg_mode bit to 0.
- Set the s0_ar_pqv_mode bit to 0.

Period regulation mode for bandwidth regulation

When configured for period regulation, the QoS regulator increases the QPV whenever the period between transactions is larger than the target, and decreases the QPV when it is lower:

- For every cycle that the period between transactions is more than the target period, the QPV is increased by a fractional amount, the scale factor K_i.
- For every cycle that the period between transactions is less than the target period, the QPV is decreased by the same fractional amount, the scale factor K_i.

The Port 0 QoS Latency Target register specifies the target period in cycles.

The Port 0 QoS Latency Scale register specifies the scale factor K_i. It is coded in powers of two, so that a programmed value of `0x0` = 2^{-12} and a programmed value of `0x7` = 2^{-5}.

You can program the QoS regulator to operate in period regulation mode by programming the following bits in the Port 0 QoS Control register:

- Set the s0_ar_qos_override_en bit to 1.
- Set the s0_ar_lat_en bit to 1.
- Set the s0_ar_reg_mode bit to 1.

There are two modes of period regulation:

- In normal mode, the QPV neither increases nor decreases when there are zero outstanding transactions.
- In quiesce high mode, the QPV increases by a fractional amount, the scale factor K_i, in every cycle where there are zero outstanding transactions.

Select the mode of period regulation by programming the s0_ar_pqv_mode bit in the Port 0 QoS Control register.

——————— Note ———————

The example shows the register names for the Port 0 RN-I bridge. The QoS register names for the XP are similar but use a dev0_ and dev1_ prefix.

—————————————————

Related references

Port S0 QoS Control register, RN-I on page 3-191.
Port S0 QoS Latency Target register, RN-I on page 3-192.

Ring/XP QoS support

This section describes Ring/XP QoS support.

In addition to the integrated QoS regulators, the XP includes support for prioritized arbitration.

The XP includes a general starvation prevention mechanism to ensure that all devices make forward progress. This includes an upload and a download starvation prevention mechanism.

Upload starvation mechanism

When sending a flit from a device onto the ring during an upload, and the flit cannot be uploaded for a number of cycles defined by the upload_starv_thresh value in the XP Auxiliary Control register, the mechanism assigns a ring-slot for use only by the starving device.

When that slot becomes free, that is, when its current flit has been downloaded, the slot is only used by the starving device, guaranteeing that the starving device makes forward progress. When the qpc_en bit of the Auxiliary Control register is set, for requests with the highest QPV value, that is, QPV==15, the slot is assigned immediately if the flit is not able to upload, without waiting for the flit age to reach the defined starvation threshold. This effectively prioritizes QoS-15 requests over other requests.

Download starvation mechanism

When sending a flit from the ring to the device during a download, and the flit cannot download for a number of cycles defined by the dnload_starv_thresh value in the XP Auxiliary Control register, the mechanism sets a bit in the download port of the relevant XP. This reserves a flit-buffer for use only by the starving device.

When that buffer becomes free, that is, when its current flit has been sent to the device, the buffer is only used by the starving flit, guaranteeing that the starving flit makes forward progress. When the qpc_en bit of the Auxiliary Control register is set, for requests with the highest QPV value, that is, QPV==15, this bit is set immediately if the flit is not able to download, without waiting for the flit age to reach the defined starvation threshold. This effectively prioritizes QoS-15 requests over other requests.

Related references

HN-F QoS support

The HN-F is a key shared system resource used for system caching and for communication with the memory controller for external memory access. It includes the following QoS support mechanisms:

QoS decoding in HN-F

The HN-F interprets the 4-bit QPV at a coarser granularity, as the following table shows.

Table 2-1 QoS classes in HN-F

QoS value range	QoS Class	Class mnemonic	Priority
15	HighHigh	HH	Highest
14-12	High	H	High
11-8	Med	M	Medium
7-0	Low	L	Low

QoS class and POCQ resource availability

The HN-F includes a 32-entry (6XP/2HNF) or 16/32-entry option (8XP/4HNF) structure, the *Point-of-Coherency Queue* (POCQ), from which all transaction ordering and scheduling is performed. The POCQ buffers are shared resources for all QoS classes, with one entry being reserved for internal use. The higher the QoS class, the higher the occupancy availability. As the following figure shows, the POCQ is partitioned so that higher priority requests are able to use a larger percentage of the POCQ buffering, ensuring bandwidth and latency requirements of higher priority transactions are met. The QoS bands change depending on which configuration option is selected. The configuration format is: `[32] Param HNF POCQ_NUM_ENTRIES_PARAM // NUM POCQ ENTRIES [16/32]`

The number of entries available for use by each QoS class is defined in the HN-F QoS Reservation register, and is software-programmable.

POCQ logical view
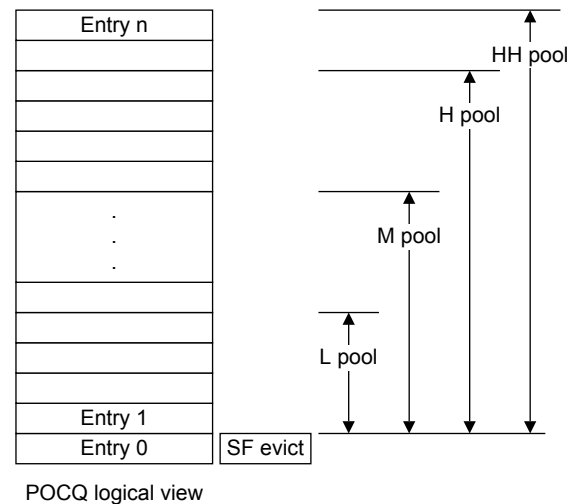
Figure 2-4 POCQ availability and QoS classes

The QoS pools are:

hh_pool   Available for HH class.

h_pool   Available for H class and HH class.

m_pool   Available for M class, H class, and HH class.

l_pool   Available for all classes.

seq   Snoop filter evictions only.

Related references

QoS Reservation register on page 3-147.

2.6 Barriers

The DMBs and DSBs barriers are based on the ARM architecture and AMBA 4 protocol. The EOBarriers and ECBarriers barriers are based on the CHI architecture.

DMBs and DSBs that are injected into the system must be converted to EOBarriers and ECBarriers respectively, at the interface to the CCN-502. All injected barriers are sent first to the MN, then to both HN-Is, to create the required sequence for previous device-type or NC accesses.

The CCN-502 supports barriers in the following manner:

- If the HN-I PoS Control and HN-I Auxiliary Control registers program the CCN-502 to terminate barriers at the HN-I, then:
  — For an EOBarrier, the HN-I:
    1. Marks all previous transactions, both reads and writes.
    2. Sends a response for the EOBarrier back to the MN, which then returns a response to the requester.
    3. Sends the previous read or write transactions downstream.
    4. Waits for completions for those previous reads and writes. Any subsequent reads or writes are not sent downstream until completions for all previous reads and writes have been received.
  — For an ECBarrier, the HN-I:
    1. Marks all previous transactions, both reads and writes.
    2. Sends the previous read and write transactions downstream.
    3. Waits for completions for those previous reads and writes. Any subsequent reads or writes received before the barrier completion, are not ordered by the barrier and can be sent downstream immediately.
    4. When completions for all previous reads and writes are received, the HN-I sends a response for the ECBarrier back to the MN, which then returns a response to the requester.
- If the HN-I PoS Control and HN-I Auxiliary Control registers program the CCN-502 to propagate barriers beyond the HN-I, then:
  — For an EOBarrier, the HN-I:
    1. Marks all prior transactions, both reads and writes.
    2. Sends a response for the EOBarrier back to the MN, which then returns a response to the requester.
    3. Sends the prior read and write transactions downstream.
    4. Sends the EOBarrier downstream as an ACE-Lite DMB.
    5. Waits for completions for the DMB. Any subsequent reads or writes are not sent downstream until completion for the DMB has been received.
  — For an ECBarrier, the HN-I:
    1. Marks all previous transactions, both reads and writes.
    2. Sends the previous read and write transactions downstream.
    3. Sends the ECBarrier downstream as an ACE-Lite DSB.
    4. Waits for completions for both the DSB and for those previous reads and writes. Any subsequent reads or writes received before the barrier completion, are not ordered by the barrier and can be sent downstream immediately.
    5. On receiving completions for the DSB and all previous reads and writes, the HN-I sends a response for the ECBarrier back to the MN, which then returns a response to the requester.

For more information about barriers, see the *ARM® AMBA® 5 CHI Architecture Specification* and the *ARM® AMBA® AXI and ACE Protocol Specification*.

Related references

PoS Control register on page 3-165.

SA Auxiliary Control register, HN-I on page 3-169.

2.7 DVM messages

The RN-Fs in the CCN-502 that support *Distributed Virtual Memory* (DVM) messages can send DVM requests and receive DVM snoops. In addition, RN-Is that include a *System Memory Management Unit* (SMMU) and connect to an RN-I bridge that supports DVMs, can receive DVM snoops.

A DVM message from an RN-F is sent to the MN and then the MN forwards it as a snoop to the participating RNs. The MN is also responsible for collecting the individual snoop responses and sending a single response back to the RN-F, that originated the DVM message transaction. The DVM Domain Control register in the MN includes the list of RNs that are destinations for DVM snoops.

——————— **Note** ———————

- An RN that issues DVM messages must also be able to receive DVM messages. If this requirement is violated, the system must not rely on the DVM message causing any DVM snoops.
- An RN-F can issue only one outstanding DVMOp(Sync).
- The CCN-502 can send a maximum of four outstanding DVM transactions to an RN-F.

—————————————————

For more information about DVM messages, see the *ARM® AMBA® 5 CHI Architecture Specification*.

Related references

DVM Domain Control register on page 3-102.

2.8 PCIe integration

The CCN-502 supports integration of a PCIe *Root Complex* (RC) or *EndPoint* (EP).

This section contains the following subsections:

2.8.1 PCIe master and slave restrictions and requirements

This section describes PCIe master and slave restrictions and requirements.

The restrictions and requirements are:
- Peer-to-peer PCIe traffic, that is, one PCIe EP talking to another PCIe EP, must not pass through the CCN-502. Requests from the PCIe master can target memory only through the HN-F, MN, or an I/O slave device downstream of the HN-I, and must not target any PCIe slave downstream of the HN-I.
- The PCIe master must not create same-**AWID** dependency between *Non-Posted Write* (NPR-Wr) and *Posted Write* (P-Wr) transactions that are sent on the RN-I AXI4/ACE-Lite slave port.
- The flow control requirements are:

  CCN-502 to PCIe slave
  The PCIe slave must be able to sink at least one NPR-Wr from the CCN-502, sent on the HN-I AXI4/ACE-Lite master port. This requirement guarantees that the HN-I AW channel remains unblocked, enabling P-Wrs from PCIe master targeting the I/O slave device to make forward progress, as required by the PCIe ordering rules.

  PCIe master to CCN-502
  If a *System Memory Management Unit* (SMMU) is in the path between the PCIe master interface and the RN-I slave interface, there are two possible options:
  — *Non-Posted Read*s (NPR-Rds) from the PCIe master must not target the HN-I.
  — Use a separate master interface port in the SMMU for page table walks, such as the *Translation Control Unit* (TCU) in a CoreLink MMU-500, and connect this port to a different RN-I that does not send any requests to the HN-I.

  ————— Note —————
  This option is only available with the MMU-500.
  —————————————

- The PCIe master can have a maximum of 256 outstanding barriers on the AW channel.

2.8.2 System requirements

This section describes system requirements.

The system requirements are as follows:
- All non-PCIe I/O slave devices must complete all writes without creating any dependency on a transaction in the PCIe subsystem.
- All non-PCIe I/O masters connected to the same RN-I as a PCIe master must not send any transactions that target or apply to I/O slave devices downstream of the HN-I.
- If an SMMU is placed in the path between the PCIe master interface and the RN-I slave interface, table-walk requests from the SMMU can only be sent to memory through the HN-F.
- ARM recommends that you set the wuo bit in the RN-I Auxiliary Control register of the RN-I that connects to the PCIe master. The wuo bit enables high bandwidth strongly-ordered coherent writes, that is, PCIe ordered coherent writes. If there are multiple RN-Is with PCIe masters attached, you can set this bit for only one of those RN-Is.

Related references

RN-I Auxiliary Control register on page 3-203.

2.8.3 HN-I programming sequence

Complete these programming steps before any non-configuration access to the HN-I.

Procedure

1. Program the PCIeRC RN-I Node ID List register to identify the RN-Is attached to PCIe masters. The PCIeRC RN-I Node ID List register is a 64-bit register where each bit signifies the NodeID of an RN. For example, if the NodeID of an RN-I attached to PCIe master is 0x2 then bit[2] of the register must be set.

2. Set the ser_devne_wr bit in the HN-I Auxiliary Control register. When this bit is set, the HN-I serializes the Device-nGnRnE writes and does not send any other write request with the same AWID as an outstanding Device-nGnRnE write.

3. Program the HN-I to not give early write completions. To do this, first clear the hni_pos_en bit in the HN-I PoS Control register, then clear the pos_early_wr_comp_en bit in the HN-I Auxiliary Control register.

Related tasks

3.4.2 Programming requirements for designs with an alternative path to the HN-I memory space

Related references

PCIeRC RN-I Node ID List register
SA Auxiliary Control register, HN-I
PoS Control register

2.9 Error handling

This section describes error handling.

This section contains the following subsections:

2.9.1 Error types

The CCN-502 supports a number of error types.

The supported errors are:

Correctable errors

These are errors that can be corrected using *Error Correction Code* (ECC) or other methods. They include:

- An L3 single-bit ECC error.

These errors are handled in the following manner. The system:

1. Counts the occurrence of these errors.
2. Masks signaling of the error to the MN.
3. Triggers error signaling to the MN using a threshold count.

Uncorrectable fatal errors

These are errors in the control logic at a node, where continuing operation might corrupt the system beyond recovery. They include:

- Double-bit ECC error in data being read from the L3 cache or snoop filter.
- A packet received with error in the target ID.
- An internal logic error.

These errors are handled in the following manner. The system:

1. Logs the error.
2. Sends an error signal to the MN.

2.9.2 Error detection, signaling, and reporting

Each CCN-502 block that connects to a configuration bus can be included in the local error reporting mechanism.

The error handling protocol is as follows:

- Each block is responsible for classifying the error into one of two predefined error types.
- The block is also responsible for logging the relevant error information, and maintaining a set of registers that are mapped into the configuration address space and accessed over the configuration bus.
- The component signals the error to the MN.
- Only the error-detecting component and the target of the error data are responsible for reporting the error. For data errors that continue with a data response, the first detector signals the error. For example:
  — If data read out of memory in response to a read has a double-bit ECC error, the memory controller sends a data response with the RespErr field set to *Data Error* (DERR). The HN-F node that receives this packet and forwards it to the requesting RN does not log or signal any error.
  — If an L3 eviction has a double-bit ECC error, the HN-F/L3 signals the error to the MN.
  — If an RN-F responding to a snoop has a data ECC error, the data is forwarded to HN-F with the RespErr field encoded as required. The RN-F might signal an error, although its responsibility is

outside the error signaling mechanism. The HN-F does not signal any error while forwarding the data to the requester and, if required, writing back the data to memory.

- The HN-F maintains a counter that is used to count the number of correctable errors encountered. The counter maintains a threshold value to enable an error to be signaled when the number of errors reaches the threshold.

Error signaling

When the MN captures an error signal, the signal is sticky and is only cleared by the error handler reading the Error Signal Valid registers in the MN.

Related references

Error Signal Valid [63:0] register on page 3-103.
Error Signal Valid [127:64] register on page 3-104.
Error Signal Valid [191:128] register on page 3-105.

Error logging

Each CCN-502 component records the details of the error in the Error Syndrome registers.

The number of Error Syndrome registers is either 1 or 2 depending on the amount of information the component must log:

- An XP uses one Error Syndrome register to log parity errors.
- The HN-I and HN-F use two Error Syndrome registers each.

The following fields in the Error Syndrome registers are used:

err_extnd	Extended. Set to 1 if the error log information extends into a second Error Syndrome register or beyond.
first_err_vld	First error valid. Set to 1 when an error is first logged.
err_class	First error classification. The error is classified into one of the three predefined error classes. See the following table.

Table 2-2 Error classification field encoding

Error class [1:0]	Field
00	Reserved
01	Correctable
10	Reserved
11	Fatal

mult_err	Multiple errors. More than one error is seen.
corrected_err_count	Corrected Error Count. A saturating counter with up to 16 bits to count corrected errors.
component_specific_reg0, component_specific_reg1	Component Specific. These fields are reserved for component-specific error logging. For packet errors, the complete control portion of the packet can be stored in these fields, extended over multiple registers.

Related references

Error Syndrome 0 register, XP on page 3-135.
Error Syndrome 0 register, L3 cache on page 3-159.
Error Syndrome 1 register, L3 cache on page 3-160.
Error Syndrome 0 register, HN-I on page 3-166.
Error Syndrome 1 register, HN-I on page 3-167.

Error log clearing

In addition to the Error Syndrome registers, each component has a write-only Error Syndrome Clear register.

Write the applicable mask bits to clear the first_err_vld and mult_err bits of the Error Syndrome 0 registers.

Related references

XP Error Syndrome Clear register on page 3-136.
L3 cache Error Syndrome Clear register on page 3-160.
HN-I Error Syndrome Clear register on page 3-168.

Related concepts

2.9.1 Error types on page 2-45.

2.9.3 Error handling requirements

The CCN-502 follows the CHI error handling methodology.

This section describes the specific behavior of the CCN-502.

Error reporting rules

The rules regarding error reporting in the CCN-502 are:
- Any error originating in the CCN-502 is reported.
- Any error originating outside the CCN-502 but corrupting the CCN-502 is reported.
- The HN-I can report an error in a response packet from outside the CCN-502 if it does not propagate the response any further, as controlled by the HN-I PoS Control register.
- All non-write errors, reported or otherwise, are propagated where possible.
- All non-posted write errors are propagated where possible.

Related references

PoS Control register on page 3-165.

Suggested interrupt handling flow

This section describes the suggested interrupt handling flow using the CCN-502 registers.

At the device

Complete the following steps to handle interrupts at the device.

Procedure

1. When an error occurs, and the first_err_vld bit is not asserted:
   a. Log the error information in the applicable Error Syndrome register and set the first_err_vld bit. The information to be logged is device-specific.
   b. Signal the error to the MN using the error signal bus.
2. If the first_err_vld bit is already asserted and the mult_err bit is not set, then set the mult_err bit.
3. If it is set, do nothing. You can set the mult_err bit multiple times, and ignore this step.

At the MN

The MN sets the **INTREQ** signal HIGH under certain conditions.

The conditions are:

- At least one bit in the applicable Error Type Valid register is set.
- The corresponding data_int_status, to mask the type of register, is not asserted in the Error Interrupt Status register.

For the error handling software on detection of assertion of INTREQ

Complete the following steps to handle errors on detection of **INTREQ**.

Procedure

1. Read the three Error Signal Valid registers.
   Error Signal valid registers are atomically cleared on a read.
2. Read the six Error Type registers.
3. For each device *x* that has its err_sig_val_*x* bits set, read the applicable Error Syndrome 0 register, except in the case of an error signaled by XP-0. If the error is signaled by XP-0, read the Error Syndrome registers of all XPs to determine which particular XP detected the parity error.

   Depending on the error device type, the error handler might not have to read any of the other Error Syndrome registers.
4. When the error handler has read all the required Error Syndrome registers:
   a. In the applicable Error Signal Valid register, write the following to each device *x* that has its err_sig_val_*x* bits set. If the error is signaled by XP-0, repeat the following write to either each XP that has its first_err_vld bit set or to all XPs. More than one XP might have detected errors.
   b. To the Error Syndrome Clear register, write 1 to bits[62,59]. Ignore the remaining bits.
      An example of data to be written for 64-bit and 32-bit registers is:
      - A 64-bit write of `0x4800000000000000` or `0xFFFFFFFFFFFFFFFF` to `0x480`.
      - A 32-bit write of `0x48000000` or `0xFFFFFFFF` to `0x484`.

      This clears the first_err_vld and mult_err bits.
5. Write to the Error Interrupt Status register to deassert the interrupt.
   Setting bit[0] = 1 enables writes to bit[4], and setting bit[4] = 1 disables the **INTREQ** interrupt. An example of data to be written for 64-bit and 32-bit registers is:
   - A 64-bit write of `0x0000000000000011`.
   - A 32-bit write of `0x00000011`.
6. Optional: Write to the global interrupt controller to enable a new interrupt capture.
7. Optional: Write to the Error Interrupt Status register to enable the interrupt.
   Setting bit[0] = 1 enables writes to bit[4], and setting bit[4] = 0 enables the **INTREQ** interrupt. An example of data to be written for 64-bit and 32-bit registers is:
   - A 64-bit write of `0x0000000000000001`.
   - A 32-bit write of `0x00000001`.

Related references

Error Interrupt Status register values

This section suggests values that you can write to the register depending on the state of the CCN-502.

The following table lists the values to write to the Error Interrupt Status register for various scenarios.

Table 2-3 Error Interrupt Status register values

Scenario	Value
To disable interrupt generation because of a PMU event overflow.	`0x88`
To disable interrupt generation because of corrected errors.	`0x44`
To disable interrupt generation because of any error.	`0x22`
To deassert an asserted **INTREQ** signal. This is not a sticky bit, that is, it always reads as zero.	`0x11`

Related references

Error reporting and signaling at the HN-I

Errors are reported at the HN-I for a number of different reasons.

The HN-I signals an error to the MN if any of the following conditions apply:

- It receives a Cacheable read, Cacheable write, or a *Cache maintenance Operation* CMO, or, it receives an MN-bound configuration read or write that does not meet the requirements specified in *3.1.3 Requirements of configuration register reads and writes* on page 3-86. These requests are steered to the HN-I. Signaling of errors to the MN can be enabled and disabled by using bit[2] (err_req_en) of the sa_aux_ctl register. By default this bit is set.
- It receives an error response on **BRESP** from downstream. This can be enabled and disabled by using bit[3] (err_rsp_en) of the sa_aux_ctl register. By default this bit is clear.

The HN-I sends a *Non-data Error* (NDERR) response to a requesting RN if any of the following applies:
- It receives an HN-I or MN-bound coherent read request.
- It receives an HN-I or MN-bound CMO, and bit[8] (rsperr_cmo_en) of the sa_aux_ctl register is set. By default this bit is clear.

Data Errors (DERR) or NDERRs from downstream read responses are passed on unmodified to the requesting RN.

Related references

2.10 Debug and PMU

The CCN-502 provides at-speed self-hosted debug and trace capabilities, and access to various performance events.

Related references

Chapter 5 Debug on page 5-230.

Chapter 6 Performance Optimization and Monitoring on page 6-246.

2.11 Node ID mapping

The following figure shows a CCN-502 interconnect with node ID and XP ID identifiers.

The identifiers are indicated as follows:
- Node ID identifiers in the corner of each component attached to an XP device port. The node ID identifier is important because all transactions to and from a device include this node ID as a *Source Identifier* (SrcID) or *Target Identifier* (TgtID) in the transactions flits. The node IDs are also included as an identifier in the CCN-502 interface signals.
- XP ID identifiers in the corner of each XP. The XP ID identifier is used to determine XP configuration register addresses and to configure debug watchpoints and performance monitors.
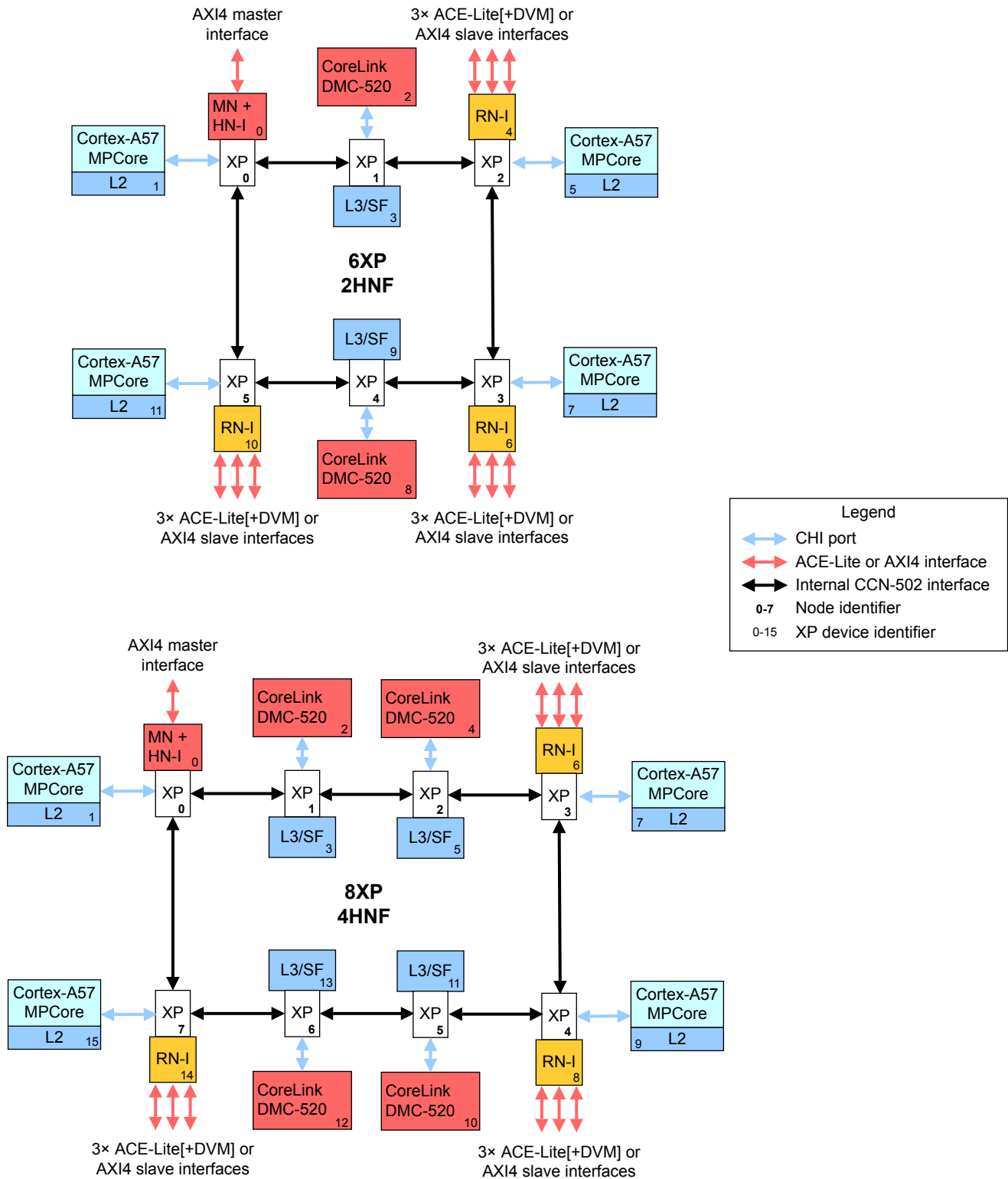
Figure 2-5 CCN-502 system with node and XP IDs

2.12 System Address Map

All CHI transactions require a TgtID.

For addressable requests, the TgtID is determined by the *System Address Map* (SAM). For responses, the TgtID is determined based on the *Node Identifier* (NodeID) of the requester, which is sent as a SrcID on a request. The coherent interconnect uses this TgtID to route packets from source to destination.

For the CCN-502, all RNs must generate the TgtID and present it on the CHI interface. An address map therefore is located in every node capable of generating a CHI addressable request. This includes the following:

• RNs because addressable requests are sent to HNs and MN.
• HNs because addressable requests are sent to the SNs, for example, memory controllers.

ACE and AXI masters are excluded from this requirement, because the RN SAM is implemented in the RN-I bridge.

This section contains the following subsections:

2.12.1 CCN-502 address map

This section describes how the address map is split into 20 distinct regions.

The CCN-502 has a global address map, that is, every master has the same view of memory. The map is split into 20 distinct memory regions across the 44-bit address space. The decode for each region is determined using a number of CCN-502 inputs that are expected to be static at and beyond the de-assertion of reset. The inputs are **SAMADDRMAPx[1:0]**, where x is an integer from 0-19.

The following figure shows the CCN-502 address map.

16 TB	ADDRMAP19[1:0]
8 TB	ADDRMAP18[1:0]
4 TB	ADDRMAP17[1:0]
2 TB	ADDRMAP16[1:0]
1 TB	ADDRMAP15[1:0]
512 GB	ADDRMAP14[1:0]
256 GB	ADDRMAP13[1:0]
128 GB	ADDRMAP12[1:0]
64 GB	ADDRMAP11[1:0]
32 GB	ADDRMAP10[1:0]
16 GB	ADDRMAP9[1:0]
8 GB	ADDRMAP8[1:0]
4 GB	ADDRMAP7[1:0]
3.5 GB	ADDRMAP6[1:0]
3 GB	ADDRMAP5[1:0]
2.5 GB	ADDRMAP4[1:0]
2 GB	ADDRMAP3[1:0]
1.5 GB	ADDRMAP2[1:0]
1 GB	ADDRMAP1[1:0]
0.5 GB	ADDRMAP0[1:0]
0 GB	

Figure 2-6 Address map

Related references

A.6 Configuration input signals on page Appx-A-271.

2.12.2 SAM configuration

This section describes the SAM configuration.

The **SAMADDRMAP***, **SAMMNNODEID**, **SAMHNINODEID***, **SAMHNF*NODEID**, **SAMHNFMODE**, and **PERIPHBASE** signals configure the SAM in the following way:

1. Associate each memory region with HN-Fs or HN-Is.
2. Identify the node IDs of HNs and MN.
3. Define the number of HN-Fs.
4. Specify the base address of the CCN-502 configuration registers:

```
SAMADDRMAP0[1:0]    //      0 - 512MB   Region Mapping
SAMADDRMAP1[1:0]    // 512MB - 1GB      Region Mapping
SAMADDRMAP2[1:0]    //    1GB - 1.5GB   Region Mapping
SAMADDRMAP3[1:0]    // 1.5GB - 2GB      Region Mapping
SAMADDRMAP4[1:0]    //    2GB - 2.5GB   Region Mapping
SAMADDRMAP5[1:0]    // 2.5GB - 3GB      Region Mapping
SAMADDRMAP6[1:0]    //    3GB - 3.5GB   Region Mapping
SAMADDRMAP7[1:0]    // 3.5GB - 4GB      Region Mapping
SAMADDRMAP8[1:0]    //    4GB - 8GB     Region Mapping
SAMADDRMAP9[1:0]    //    8GB - 16GB    Region Mapping
SAMADDRMAP10[1:0]   //   16GB - 32GB    Region Mapping
SAMADDRMAP11[1:0]   //   32GB - 64GB    Region Mapping
SAMADDRMAP12[1:0]   //   64GB - 128GB   Region Mapping
SAMADDRMAP13[1:0]   //  128GB - 256GB   Region Mapping
```

```
SAMADDRMAP14[1:0]  // 256GB - 512GB  Region Mapping
SAMADDRMAP15[1:0]  // 512GB - 1TB    Region Mapping
SAMADDRMAP16[1:0]  //   1TB - 2TB    Region Mapping
SAMADDRMAP17[1:0]  //   2TB - 4TB    Region Mapping
SAMADDRMAP18[1:0]  //   4TB - 8TB    Region Mapping
SAMADDRMAP19[1:0]  //   8TB - 16TB   Region Mapping
SAMMNNODEID[5:0]   // NodeID of MN
SAMHNINODEID0[5:0] // NodeID of HN-I 0
SAMHNINODEID1[5:0] // NodeID of HN-I 1
SAMHNF0NODEID[5:0] // NodeID of HN-F 0
SAMHNF1NODEID[5:0] // NodeID of HN-F 1
SAMHNF2NODEID[5:0] // NodeID of HN-F 2
SAMHNF3NODEID[5:0] // NodeID of HN-F 3
SAMHNF4NODEID[5:0] // NodeID of HN-F 4
SAMHNF5NODEID[5:0] // NodeID of HN-F 5
SAMHNF6NODEID[5:0] // NodeID of HN-F 6
SAMHNF7NODEID[5:0] // NodeID of HN-F 7
SAMHNFMODE[2:0]    // Indication of number of HN-Fs
PERIPHBASE[43:24]  // Address offset of configuration registers in MN
```

The decoded destination for each region must be static. The following table shows the valid values for the **SAMADDRMAPx[1:0]** inputs.

Table 2-4 Decoder mapping

SAMADDRMAPx[1:0]	Decode
0b00	HN-F(s)
0b01	HN-I
0b10	Reserved
0b11	

Although **SAMADDRMAPx[1:0]** only decodes to two destinations, there are three possible destinations for addressable requests:

- MN.
- HN-I
- One of the HN-F partitions.

The MN is responsible for all accesses to the CCN-502 *Control and Status register*s (CSRs), and for distribution and handling of barrier and *Distributed Virtual Message operations* (DVMops). Identification of barrier and DVMop transactions targeted to the MN occurs as a function of the transaction type, not as a function of the address.

Access to the CSRs is through a 16MB memory-mapped region. This region must be mapped to the HN-I where address decoding is performed to identify CSR accesses. These accesses are subsequently managed by the MN. Therefore, although the MN manages accesses to the CSRs, these accesses must initially be mapped to and sent to HN-I.

The base address for the CSRs is defined using the static input **PERIPHBASE[43:24]**. **PERIPHBASE** must reside in a **SAMADDRMAPx[1:0]** that corresponds to HN-I. ARM recommends that **PERIPHBASE** resides in the bottom 4GB of address space so that 32-bit devices can access the CCN-502 configuration registers.

For the **SAMHNFxNODEID** inputs, the HN-Fs are numbered in ascending order from the smallest HN-F NodeID to the largest HN-F NodeID.

Related references

A.6 Configuration input signals on page Appx-A-271.

2.12.3 RN SAM address hash function

For the RN SAM, the HN-F partitions are treated as a single destination. To determine the HN-F partition responsible for a particular address, the following hash function is applied:

```
------------------ 6XP/2HNF ------------------
HN-F identifier: Which HN[1:0] = ^ addr[43:6];

------------------------------ 8XP/4HNF ------------------------------
HN-F identifier: Which HN[1:0] = addr[43:42] ^ … ^ addr[9:8] ^ addr[7:6];
```

This hash identifies which HN-F is responsible for an address, but the identifier must be passed through the RN SAM to obtain the node ID of the identified HN-F.

2.12.4 HN-F SAM

The HN-F SAM is similar to the RN SAM. It is present in all HN-F partitions to route transactions to the SN-Fs, which are generally memory controllers. The HN-F has mechanisms to identify RNs for snoops that are not part of the HN-F SAM and so are not described in this section. The following figure shows the HN-F SAM.
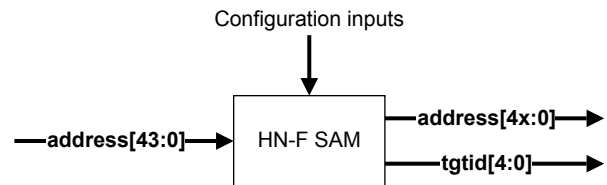
Figure 2-7 HN-F SAM

The HN-F SAM supports one, two, three, or four SN-Fs. To determine the particular SN-F for a particular address, you must use a hash function. The hash function depends on:

*   The number of HN-F partitions.
*   The number of SN-Fs.

The following table shows the function that generates the SN-F TgtID and the address bits that must be used by the SN-F to guarantee contiguity of addresses presented to the SN-F. ARM recommends that the SoC designer carefully analyze the address mapping functions between the request source and MC to understand the resulting address map for each MC. SN-F and HN-F identifiers refer to NodeIDs.

————— Note —————

The CCN-502 does not modify the address sent to the SN-F. The system designer is responsible for ensuring that the appropriate address bits are used within the memory subsystem to enable security, contiguity of memory regions, and other issues as required.

Table 2-5 HN-F SAM map

HN-Fs	SN-Fs	SN-Fs at (NodeID)	HN-F partitions	Which SN-F? (NodeID)	SN-F address
6XP/2HNF					
2	2	2 and 8	3 and 9	2 and 8	{Address[43:9], Address[7:0]}
8XP/4HNF					

Table 2-5 HN-F SAM map (continued)

HN-Fs	SN-Fs	SN-Fs at (NodeID)	HN-F partitions	Which SN-F? (NodeID)	SN-F address
4	2	2 and 10	3, 5	2	{Address[43:8], Address[6:0]}
			11, 13	10	
		4 and 12	3, 5	4	
			11, 13	12	
	3	-	-	Use 3 SN striping	Use 3 SN striping
	4	2, 4, 10, 12	3	2	{Address[43:9], Address[6:0]}
			5	4	
			11	10	
			13	12	

3 SN-F memory striping

In the 3 SN-F hashed mode, addresses are striped at 256-byte granularity between the 3 SN-Fs.

Each HN-F uses the hn_cfg_three_sn_en bit in its hnf_sam_control register to enable routing to 3 SNs.

In the hnf_sam_control register, the hn_cfg_sam_top_address_bit0 and hn_cfg_sam_top_address_bit1 fields must be configured at boot time. These fields must be set to the top address bits of addressable DRAM. These two address bits are decoded, and are then used with a hashing function to determine the target SN-F. For example, if 3GB of DRAM are used, that is, 1GB at each SN, then the hn_cfg_sam_top_address_bit1 field must be set to 31, and the hn_cfg_sam_top_address_bit0 field must be set to 30.

——————— **Note** ———————

Memory aliasing or holes can occur if the top two address bits cannot be used to decode between the three DRAM regions.

————————————————

The full physical address is sent to the SNs, but the memory controller must ignore the top 2 bits of the addressable DRAM. Continuing the example of 3GB, the memory controller must ignore bits[31:30] of the address, using only bits[29:0].

Each HN-F uses the hn_cfg_sn<N>_nodeid fields, in its hnf_sam_control register, to map each target index to a slave node. For example, if the target index is:
*   0, the hn_cfg_sn0_nodeid field defines the target SN.
*   1, the hn_cfg_sn1_nodeid field defines the target SN.
*   2, the hn_cfg_sn2_nodeid field defines the target SN.

If the CCN-502 is configured at build time to have three SNs, then the default values are shown in the following table.

Table 2-6 3 SN striping values

Chosen SNs	SN 0	SN 1	SN 2
2, 4, and 10	2	4	10
4, 10, and 12	4	10	12
10, 12, and 2	2	10	12
12, 2, and 4	2	4	12

Related references

HN-F SAM Control register on page 3-143.

2.13 Clocking and resets

This section contains the following subsections:

2.13.1 Clocking

The following sections describe the CCN-502 clocking microarchitecture:

Asynchronous communication

To close timing in a CCN-502 system, there are various classes of timing paths to consider.

The timing paths are:

- Paths within a CCN-502 logic block.
- Paths from a device to and from a CCN-502 XP upload and download port.
- Paths between XPs.

The CCN-502 provides straightforward convergence of timing paths within a logic block, to avoid timing issues.

For the device/XP communication path, each XP optionally includes a *device/XP source-synchronous asynchronous bridge* (DSSB). This enables the device to run asynchronously to the CCN-502. The DSSBs can only exist in the XPs as the following figure shows, and the DSSBs exist in two distinct groups:

- Those in XPs connected to memory controllers.
- Those in XPs connected to processor compute clusters.

The inclusion or exclusion of each of these groups is independently optional. However, inclusion or exclusion occurs only at group granularity, that is, if a DSSB is present in a group, all DSSBs in that group must be present.

Each DSSB is implemented as two blocks:
- A block in the XP that contains the CCN-502 power and clock domain functionality.
- The other block contains the device domain clock and power functionality, which can be implemented in the device hierarchy.

The DSSB blocks connected to the processor compute clusters are the CCN502_RNF_DSSB, and the DSSB blocks connected to the memory controllers are the CCN502_SNF_DSSB.

The following figure shows a CCN-502 system with optional DSSBs.
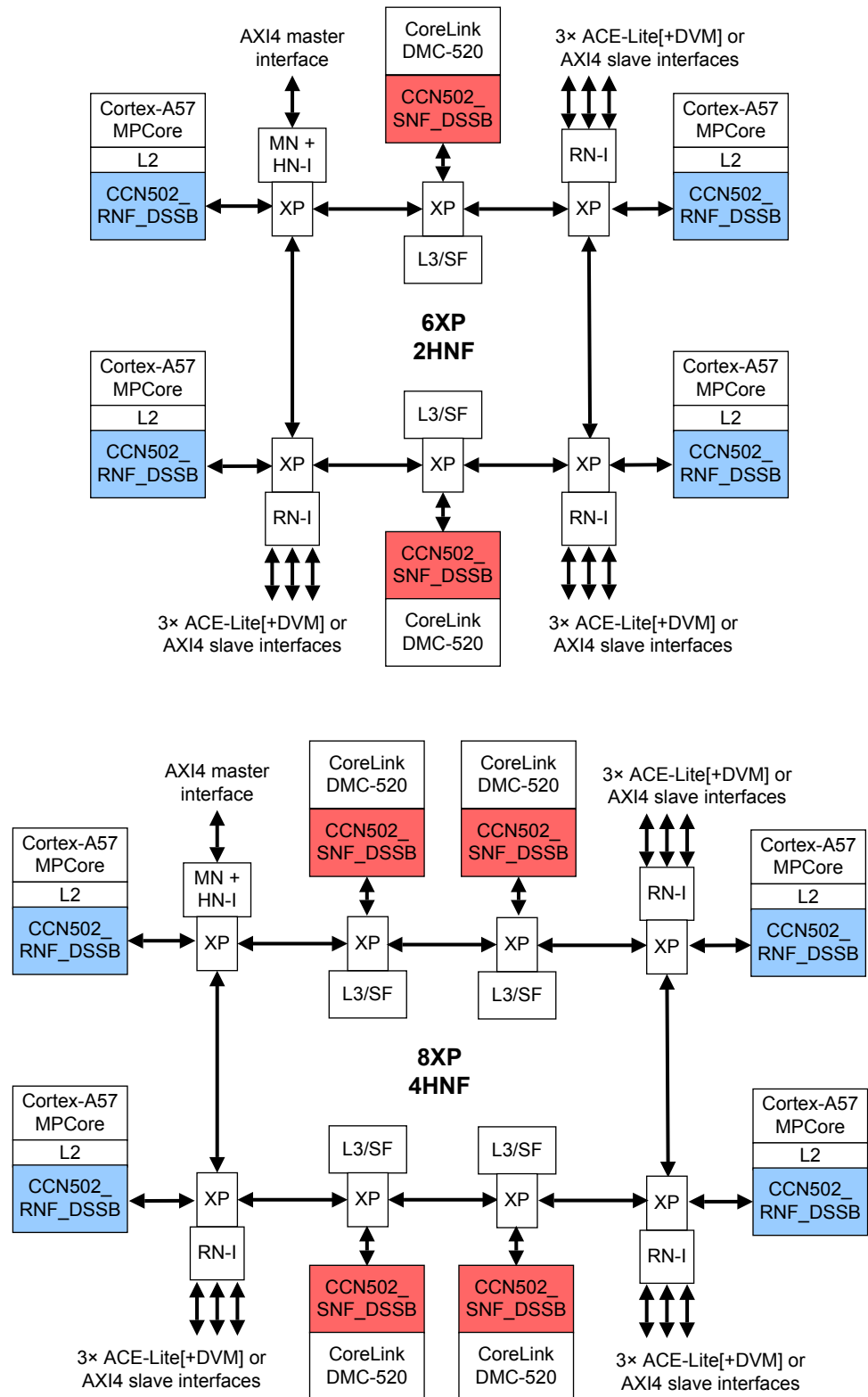
Figure 2-8 CCN-502 system with optional DSSBs

All asynchronous communication between CCN-502 and AMBA devices, for example, asynchronous communication between the CCN-502 interconnect and master I/O subsystem, are supported by existing AMBA asynchronous domain bridge products, such as the CoreLink ADB-400 AMBA Domain Bridge.

Clock domains

The CCN-502 has different clock domains depending on the options included in a particular implementation.

Figure 2-9 CCN-502 clock domain, fully synchronous on page 2-61 shows the single clock domain in a CCN-502 interconnect without asynchronous capabilities or register slices.
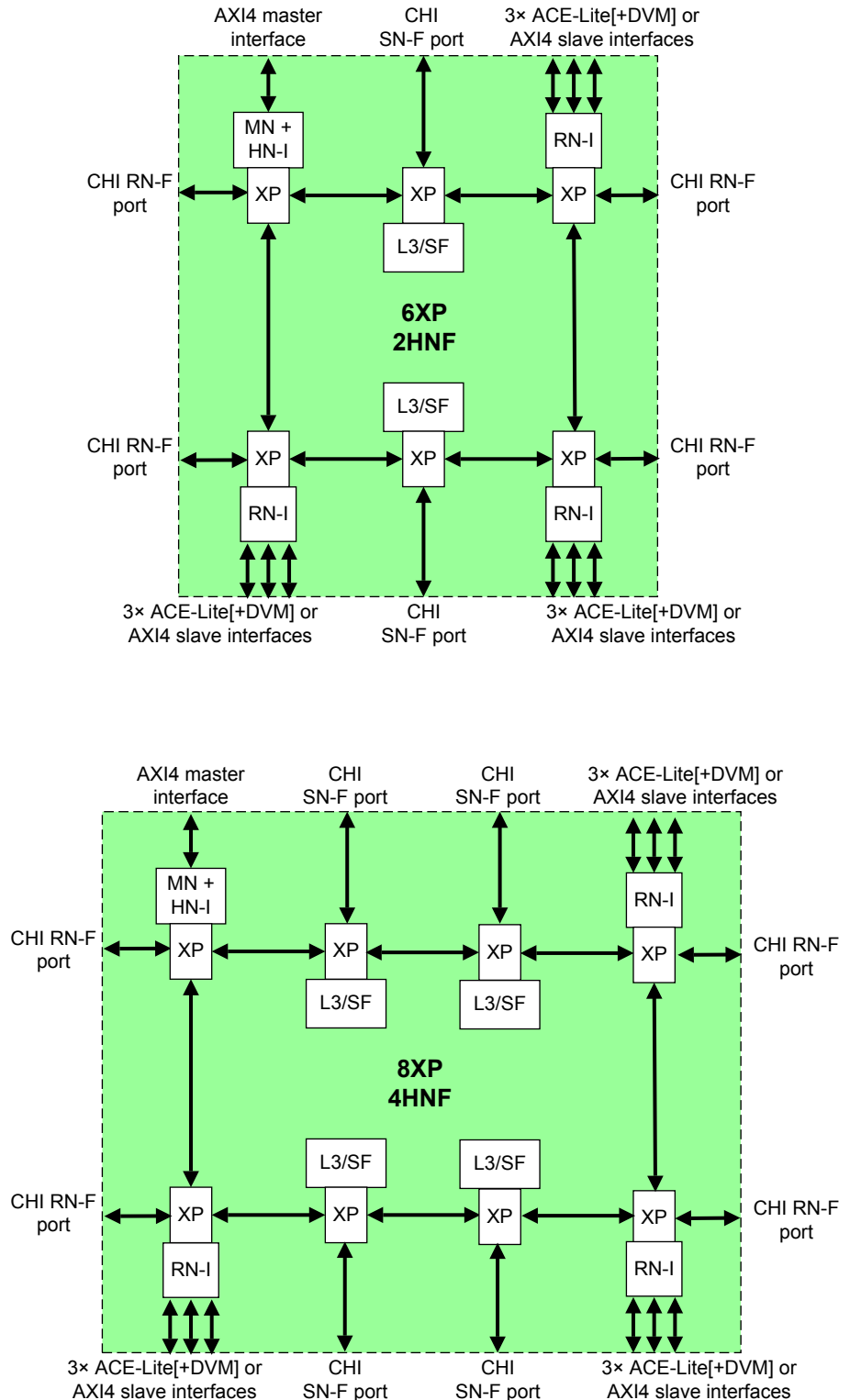
Figure 2-9 CCN-502 clock domain, fully synchronous

Figure 2-10 CCN-502 clock domains with optional DSSBs on page 2-62 shows the multiple clock domains in a CCN-502 interconnect that includes the optional DSSBs for asynchronous communication with processor compute clusters or DMCs. Although most of the CCN-502 is clocked with a shared synchronous clock, the receive-FIFO logic in the respective DSSBs is clocked with the clock sent by the transmitting device, as required for source-synchronous communication. The CCN-502 does not place any requirements or constraints on the various processor or DMC clocks.
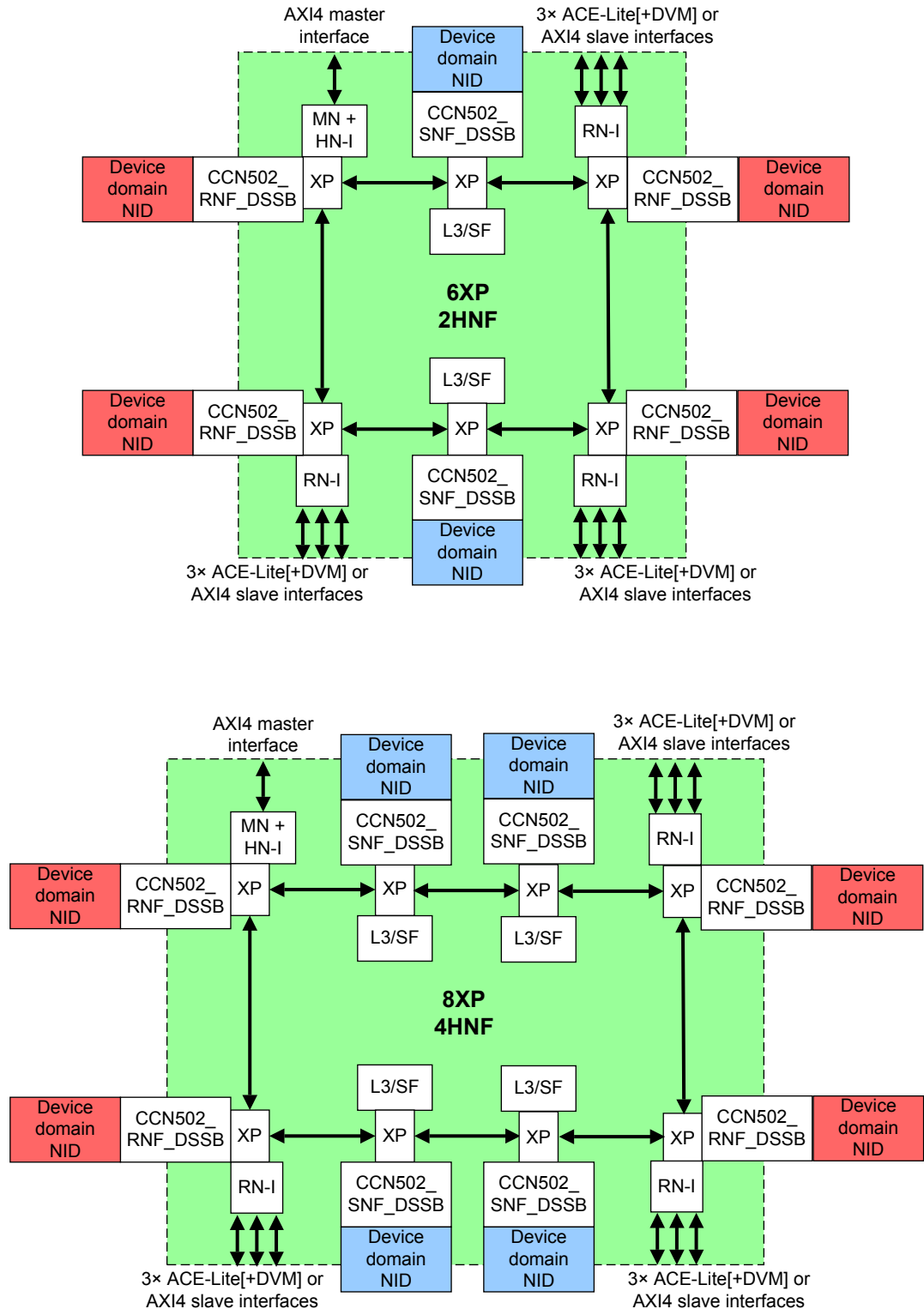
Figure 2-10 CCN-502 clock domains with optional DSSBs

Because of the group-level granularity, that is, processor or DMC groups, at which the DSSBs are optionally included, there are three possible configurations when DSSBs are included:

- Both processor and DMC DSSBs are included.
- Only processor DSSBs are included.
- Only DMC DSSBs are included.

The respective clock domain requirements of these three configurations change as required.

The clocking configurations in a CCN-502 interconnect that includes some or all of the optional protocol bridges are different from those described for the CCN-502 baseline configuration. The main difference between the two is that the inclusion of a protocol bridge on an XP device port prevents the inclusion of a DSSB on that device port. In other words, any asynchronous communication between an AMBA device and the CCN-502 interconnect through an optional protocol bridge must be provided by an existing AMBA asynchronous domain bridge, such as the CoreLink ADB-400 AMBA Domain Bridge.

Clock hierarchy

The clocking delivery and clock gating architecture is hierarchical.

Within the clock gating hierarchy, three levels of clocks are defined:

Global clocks
These are the clock inputs to the CCN-502 system. The global clocks that the SoC provides are likely to be controlled by an additional level of clock gating or clock control outside of the system. Although this is not a system requirement, the CCN-502 includes support for external clock control.

Regional clocks
Regional clocks are created as an output of regional clock gaters that include a coarse enable for coarse-grained clock gating under idle or mostly idle conditions. This enables a higher level of power reduction than is possible using local clock gating, because the clock network between the regional and local gaters can be shut down using the regional gaters. The regional clock gaters are instantiated in and controlled by the CCN-502 RTL. The exact set of regional clocks is internal to the CCN-502 and is not described in this book.

Local clocks
Local clocks are created as an output of the local clock gaters that are controlled by fine-grained enables that the CCN-502 RTL creates. Local clocks are used to directly clock sequential elements in the CCN-502. The exact set of local clocks is internal to the CCN-502 and is not described in this book.

The following figure shows the clocking hierarchy.
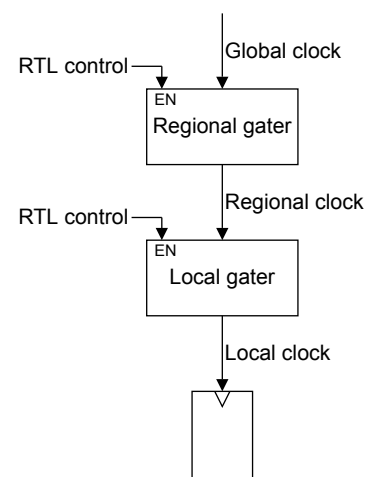
Figure 2-11 Clocking hierarchy

Related concepts

2.14 Power and clock management on page 2-68.

Global clocks

Global clock inputs for a specific configuration are a combination of the global clocks for the baseline configuration and those included for all applicable configuration options.

The number and types of global clock inputs depend on the configuration of the network, as follows:

Baseline

The baseline configuration contains a single synchronous clock domain for the entire CCN-502 system, and includes the following global clock input:

GCLK0 Clock input for Domain 0, which, in this configuration, is defined as the entire CCN-502.

Optional DMC DSSB

When the optional DMC DSSBs are included, additional clock inputs and outputs are provided. In this configuration, the definition of Domain 0 is unchanged. In the following clock names, *<x>* represents the node ID of the *Dynamic Memory Controller* (DMC):

RXRSPGCLKCD_NID<x> Source-synchronous input clock that is used to receive the RSP flit from the DMC.

RXDATGCLKCD_NID<x> Source-synchronous input clock that is used to receive the DAT flit from the DMC.

TXREQGCLK_NID<x> Source-synchronous output clock that is used to send the REQ flit from the CCN-502 to the DMC.

TXDATGCLK_NID<x> Source-synchronous output clock that is used to send the DAT flit from the CCN-502 to the DMC.

Optional processor DSSB

When the optional processor DSSBs are included, additional clock inputs and outputs are provided. In this configuration, the definition of Domain 0 is unchanged. In the following clock names, *<x>* represents the node ID of the processor cluster:

RXREQGCLKCD_NID<x> Source-synchronous input clock that is used to receive the REQ flit from the processor cluster.

RXRSPGCLKCD_NID<x> Source-synchronous input clock that is used to receive the RSP flit from the processor cluster.

RXDATGCLKCD_NID<x> Source-synchronous input clock that is used to receive the DAT flit from the processor cluster.

TXRSPGCLK_NID<x> Source-synchronous output clock that is used to send the RSP flit from the CCN-502 to the processor cluster.

TXDATGCLK_NID<x> Source-synchronous output clock that is used to send the DAT flit from the CCN-502 to the processor cluster.

TXSNPGCLK_NID<x> Source-synchronous output clock that is used to send the SNP flit from the CCN-502 to the processor cluster.

The following table shows the possible clocking combinations in the CCN-502 system. In the clock names, *<x>* represents the node ID of the processor cluster or DMC.

Table 2-7 Clock domain options

Processor-DSSB	DMC-DSSB	Number of domains	Global clock inputs
No	No	1 (6XP/2HNF). 1 (8XP/4HNF).	**GCLK0**
No	Yes	3 (6XP/2HNF). 5 (8XP/4HNF).	**GCLK0** **RXRSPGCLKCD_NID<x>, RXDATGCLKCD_NID<x>, TXREQGCLK_NID<x>, TXDATGCLK_NID<x>**
Yes	No	5 (6XP/2HNF). 5 (8XP/4HNF).	**GCLK0** **RXREQGCLKCD_NID<x>, RXRSPGCLKCD_NID<x>, RXDATGCLKCD_NID<x>, TXRSPGCLK_NID<x>, TXDATGCLK_NID<x>, TXSNPGCLK_NID<x>**
Yes	Yes	7 (6XP/2HNF). 9 (8XP/4HNF).	**GCLK0** **RXRSPGCLKCD_NID<x>, RXDATGCLKCD_NID<x>, TXREQGCLK_NID<x>, TXDATGCLK_NID<x>** **RXREQGCLKCD_NID<x>, RXRSPGCLKCD_NID<x>, RXDATGCLKCD_NID<x>, TXRSPGCLK_NID<x>, TXDATGCLK_NID<x>, TXSNPGCLK_NID<x>**

Related concepts

Clock enable inputs

The CCN-502 includes several clock enable inputs.

The clock enable input signals are:

ACLKEN_S   This input is present on each AMBA slave interface.

ACLKEN_M   This input is present on each AMBA master interface.

DCLKEN   This input is present on the debug and trace **STMHWEVENT** interface.

All the clock enables, shown here as ***CLKEN***, have identical functionality, enabling the respective interfaces with which they are included to run at integer fractions of **GCLK0**, that is, slower than **GCLK0**, ranging from 1:1 to 4:1. **DCLKEN** is limited to 2:1 to 4:1 integer fractions. This enables synchronous communication with slower SoC logic.

CLKEN asserts one **GCLK0** cycle before the rising edge of **SoC-CLK**. SoC control logic can change the ratio of **GCLK0** frequency to the SoC clock, **SoC-CLK**, frequency dynamically using ***CLKEN***.

The following figure shows a timing example of ***CLKEN*** that changes the ratio of the frequency at which the relevant interface operates respective to **GCLK0** from 3:1 to 1:1.
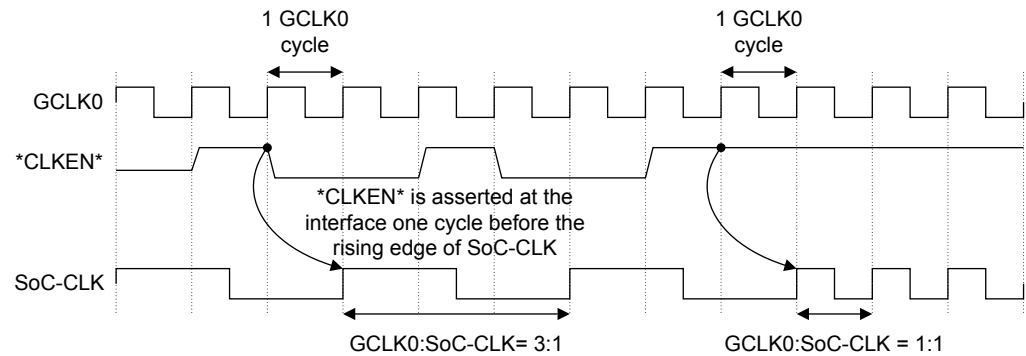
Figure 2-12 *CLKEN* with GCLK0:SoC-CLK ratio changing from 3:1 to 1:1

Timing closure with register slices

The network provides register slices to assist during timing closure.

The CCN-502 includes the following optional register slices:

• *Device Register Slice* (DRS), at the device/XP boundaries.

The slices are simple repeater-flop structures applied across the entire communication boundary. Register slices can only be used at a synchronously-clocked communication boundary, and a DRS cannot be used in conjunction with a DSSB. Any device/XP boundary can contain up to two back-to-back DRSs. Link-layer buffering for devices connected to the DRSs is automatically adjusted to handle the additional credit/response latency.

2.13.2   Reset

The CCN-502 has a single global reset input signal, **nSRESET**. If the network includes the optional DMC or DSSBs, then each CCN502_RNF_DSSB or CCN502_SNF_DSSB also has an **nDEVRESET** reset input.

nSRESET is an active-LOW signal, and is used as an input for each CCN-502 component. All CCN-502 components locally synchronize their **nSRESET** input, so that **nSRESET** at the component boundary can be asynchronously or synchronously asserted and deasserted.

There are no specific requirements for the relative timing of **nSRESET** assertion or deassertion as received by the respective components in a CCN-502 system. That is, all versions of **nSRESET** in a CCN-502 can assert or deassert asynchronously and at different times as required by the implementation of what is expected to be a multicycle path to each component. However, all components must see at least 24 concurrent cycles of an asserted **nSRESET** before **nSRESET** is deasserted. This 24-cycle time period is measured from the time **nSRESET** is asserted at the boundary of the last CCN-502 component to receive **nSRESET** and is measured using the period of the slowest global clock input.

All CCN-502 clock inputs must be active during the required 24-cycle period of **nSRESET** assertion, and must remain active for at least 10 cycles following deassertion of **nSRESET**.

Optional DSSB

When the optional DMC or processor DSSBs are included, the CCN502_RNF_DSSB and CCN502_SNF_DSSB blocks have a reset input signal, **nDEVRESET**. This signal is active-LOW, and must connect to the primary reset input of the device.

————— **Note** —————

For a protocol device to communicate with other protocol devices after reset, that device:

• Must have completed all necessary protocol layer initialization.
• Must be in the correct state as required by the devices with which it communicates.

This means that:

- For a device that originates requests, that device has no outstanding requests after reset.
- For a device that receives requests, the system expects no outstanding responses from that device.

2.14 Power and clock management

The CCN-502 includes several power management and clock management capabilities, that are either externally controllable or are assisted by the *System-on-Chip* (SoC).

The power management and clock management capabilities are:

- High-level clock gating that indicates inactivity in the system, enabling an external clock controller to disable global clock inputs during periods of inactivity. This significantly reduces dynamic power consumption.
- A number of distinct predefined power states, including states in which all, half, or none of the L3 tag/data RAMs can be powered up, powered down, or in retention:
  — A state in which only the HN-F snoop filter is active.
  — A state in which neither the L3 RAMs, nor snoop filter RAMs, are active.

  These power states reduce static and dynamic power consumption.
- Support for static retention in HN-F in which the SoC places L3 and SF RAMs in a retention state. This reduces static power consumption.
- Support for in-pipeline low-latency data RAM retention control, in which a 4-cycle wakeup signal provided by the L3 can be used to put the L3 data RAMs in retention for very short periods of time, relative to P-Channel-controlled retention states.

This section contains the following subsections:

2.14.1 High-level clock gating

High-level Clock Gating (HCG) is a mechanism supported by the PCCB that notifies the SoC when the CCN-502 is inactive. HCG enables an external SoC clock control unit, the *External Clock Controller* (ExtCC), to stop the CCN-502 **GCLK0** clock inputs.

The CCN-502 includes a Q-Channel interface that enables the CCN-502 and the SoC to communicate to achieve HCG functionality through the PCCB. See the *AMBA® Low Power Interface Specification, ARM® Q-Channel and P-Channel Interfaces* for more information.

External clock controller

This section describes the external clock controller.

The following figure shows an example of how the ExtCC controls the clock gating flow. This example clock gating sequence begins and ends with the Q-Channel in either of the following states:

- Quiescent state (Q_STOPPED), where **QREQn** and **QACCEPTn** are asserted.
- Active state (Q_RUN), where **QREQn** and **QACCEPTn** are deasserted.
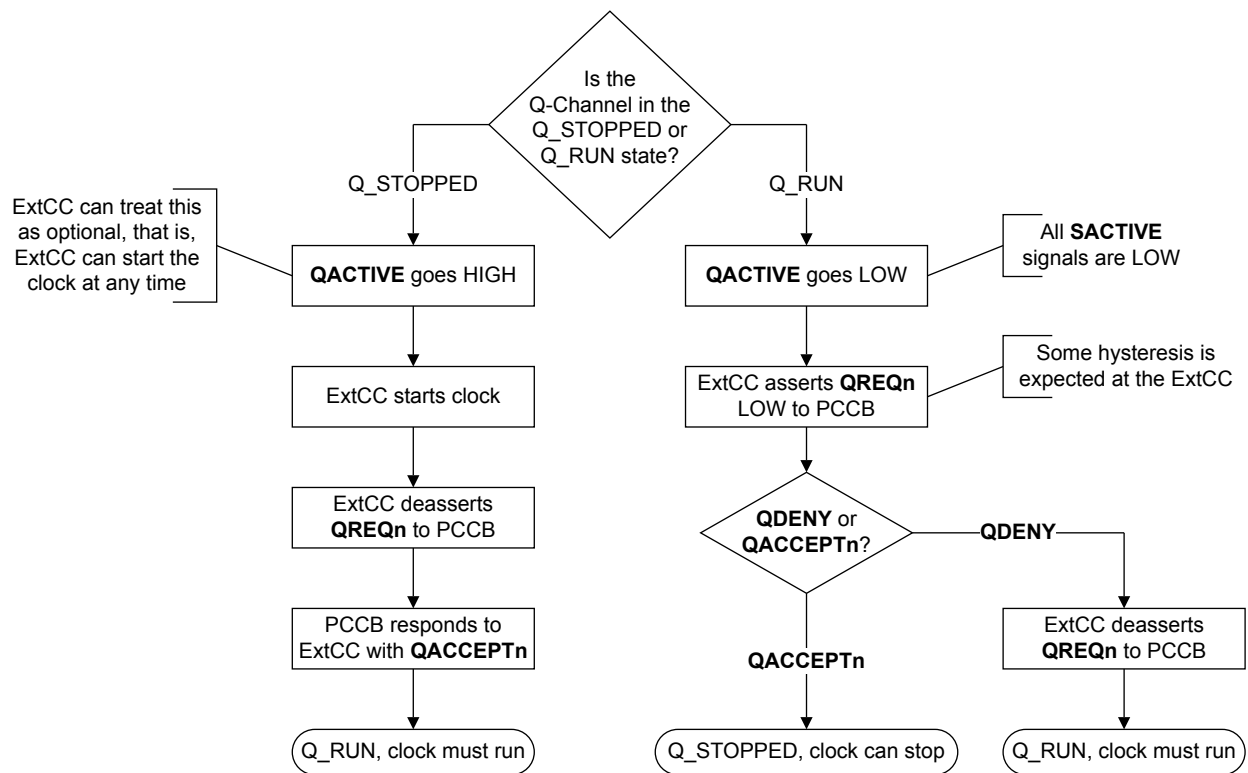
Figure 2-13 Clock gating control using ExtCC

The requirements of the ExtCC are as follows:

- It must supply a clock to the CCN-502 when the Q-Channel is in any state other than Q_STOPPED.
- The ExtCC can either choose to gate the clock to the CCN-502 when the Q-Channel is in the Q_STOPPED state, or it can choose to run the clock at any time.
- Although this manual does not describe the exact behavior of the ExtCC and its usage of **QREQn** in response to **QACTIVE** deassertion, the design of the ExtCC is likely to include a control loop with some hysteresis so that HCG is enabled when the system is inactive for long periods, but is not enabled for very short periods of inactivity. If the clocks are stopped in response to short periods of inactivity, performance of the CCN-502 can be negatively affected.
- It is the responsibility of the SoC designer to fully control the clock management Q-Channel. If there is a requirement for a control or configuration bit to completely enable or disable HCG functionality, that register or bit must exist outside of the CCN-502. More specifically, the CCN-502 has no internal means of disabling HCG.

Related references

2.14.2   Power domains

This section describes the CCN-502 power domains.

The following figure shows the CCN-502 power domains.
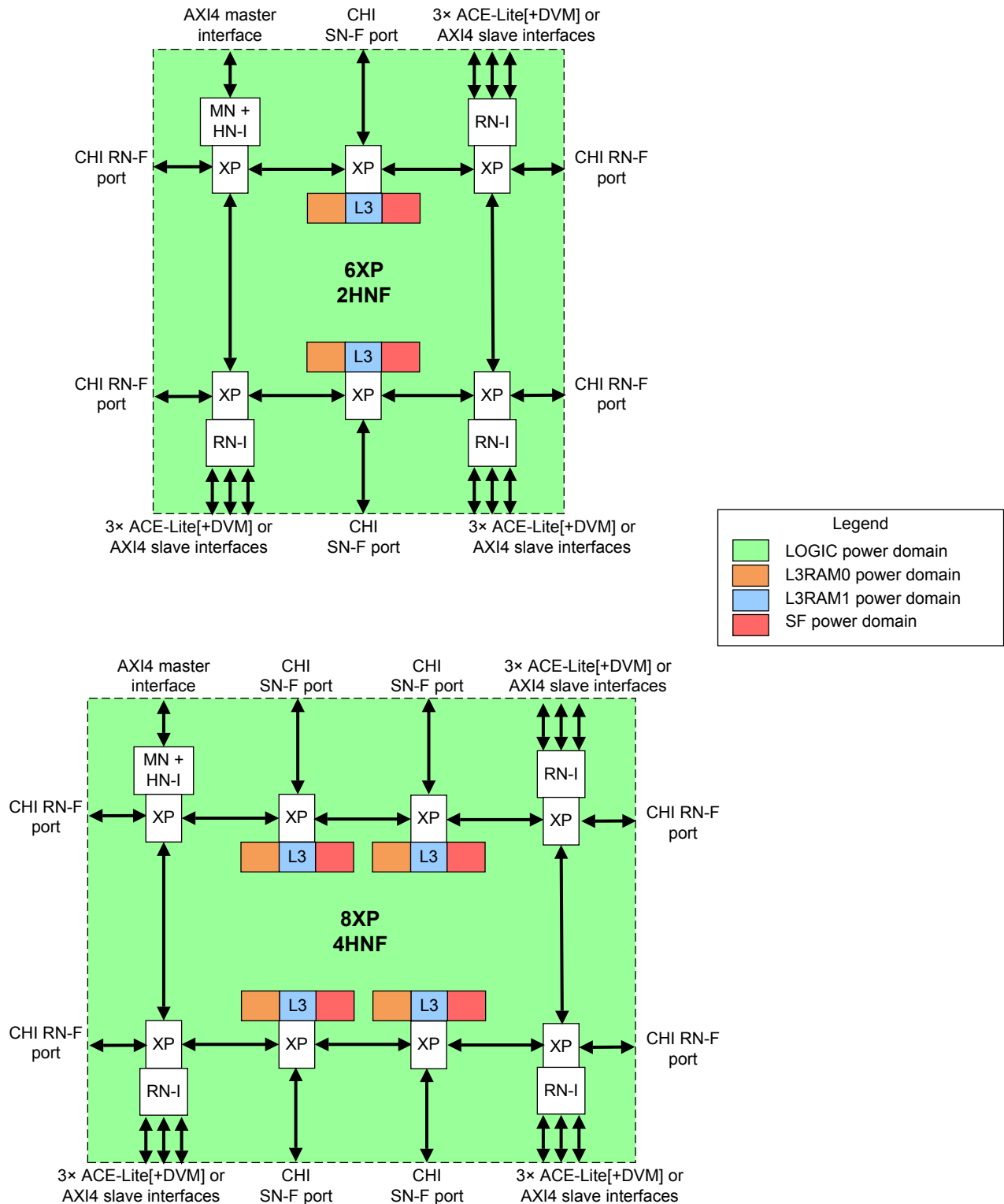
Figure 2-14 CCN-502 power domains

─────── **Note** ───────

For configurations with CHI interfaces and optional DSSBs, *Figure 2-10 CCN-502 clock domains with optional DSSBs* on page 2-62 shows the applicable device domain node IDs.

───────────────

The power domains are used as follows:

LOGIC power domain
> All logic except HN-F L3 tag and data RAMs and HN-F snoop filter RAMs.

L3RAM0 power domain
> L3 tag/data RAMs way[7:0] for all 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-F partitions. This is a single logical power domain across all partitions.

L3RAM1 power domain
> L3 tag/data RAMs way[15:8] for all 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-F partitions. This is a single logical power domain across all partitions.

SF power domain
> Snoop filter RAMs for all 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-F partitions. This is a single logical power domain across all partitions.

Optional DMC DSSB power domains
> CCN502_SNF_DSSB logical power domain. There is a unique domain for each DSSB instance.

Optional processor DSSB power domains
> CCN502_RNF DSSB logical power domain. There is a unique domain for each DSSB instance.

2.14.3 Power states

This section lists the valid CCN-502 power states and shows the power state transition diagram.

The following table shows the valid CCN-502 power states and their requirements.

Table 2-8 CCN-502 power states

State	Description	Control logic	Snoop filter power state	L3 way[7:0] power state	L3 way[15:8] power state
FAM	Full run mode	On	On	On	On
HAM	Run mode with L3H1 (L3 upper ways) disabled.	On	On	On	Off
SF	Run mode with L3H1 and L3H2 disabled.	On	On	Off	Off
NOL3	Run mode with L3H1, L3H2, and SF disabled.	On	Off	Off	Off
FAM Dyn. Ret.	Run mode with L3H1, L3H2, and SF in retention.	On	Retention	Retention	Retention
HAM Dyn. Ret.	Run mode with L3H1 and SF in retention, and L3H2 in powerdown.	On	Retention	Retention	Off
SF Dyn. Ret.	Run mode with SF in retention, and L3H1 and L3H2 in powerdown.	On	Retention	Off	Off
FAM Static Ret.	Shutdown mode with L3H1, L3H2, and SF in retention.	Off	Retention	Retention	Retention
HAM Static Ret.	Shutdown mode with L3H1 and SF in retention, and L3H2 in powerdown.	Off	Retention	Retention	Off
SF Static Ret.	Shutdown mode with SF in retention, and L3H1 and L3H2 in powerdown.	Off	Retention	Off	Off
OFF	Shutdown.	Off	Off	Off	Off

The L3 cache operates in four main operational modes:

FAM   Full-Associativity Mode, where the snoop filter and the entire L3 cache are used.

HAM   Half-Associativity Mode, where the snoop filter is enabled but the upper half of the L3 ways are disabled and powered off.

SFONLY Snoop-Filter-Only mode, where the snoop filter is enabled but all the L3 cache is powered off.

NOL3   No-L3 mode, where the snoop filter and L3 cache are disabled and powered off.

The following figure shows the valid power states and transitions for a CCN-502 system.
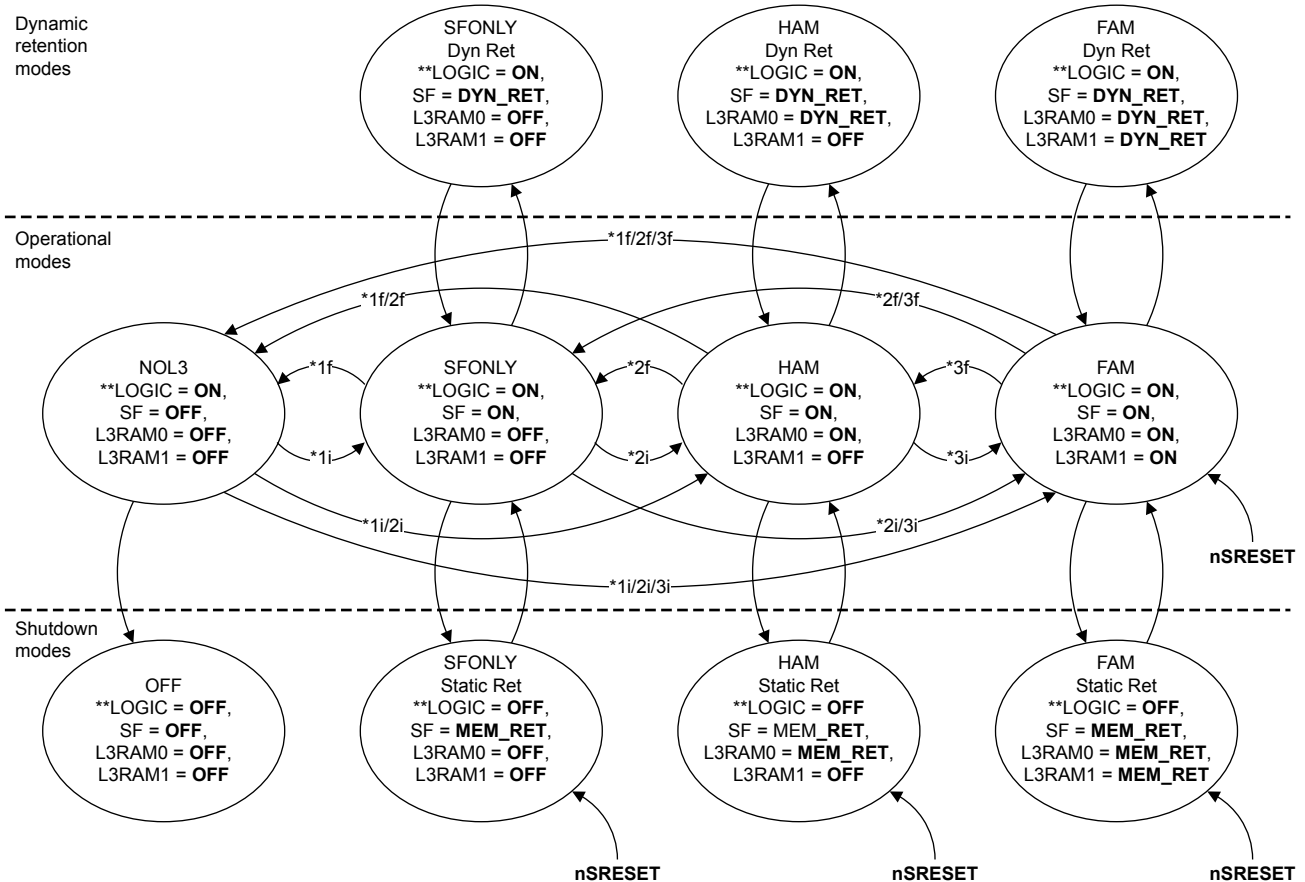
Note: **BOLD** text shows the required power state.

* Automatic initialization and flushing actions:
    1i: Initialize snoop filter RAMs.
    2i: Initialize lower ways of tag RAMs.
    3i: Initialize upper ways of tag RAMs.
    1f: Flush (force back-invalidations as necessary and invalidate) snoop filter RAMs.
    2f: Flush (clean/invalidate) lower ways of tag/data RAMs.
    3f: Flush (clean/invalidate) upper ways of tag/data RAMs.

** All designations refer to P-state values required to enter the respective state.

Any transition between power states which does not include an **nSRESET** designation on that arc is achieved through P-channel **PREQ** commands.

Figure 2-15 Power state transitions

From FAM, HAM, or SFONLY, the L3 cache can enter a dynamic retention mode, where:

- The logic power is on.
- The voltage to the RAMs is on, but is reduced to a level that is sufficient for bitcell retention but insufficient for normal operation.

From these states, the L3 cache can also enter a static retention mode, where:

- The logic power is turned off.
- The voltage to the RAMs is on, but is reduced to a level that is sufficient for bitcell retention but insufficient for normal operation.

The difference between the dynamic and static retention modes is that dynamic retention is entered because of a dynamic activity or inactivity indicator from the HN-F to the SoC. This is an output of the HN-F that is used to determine periods of inactivity long enough to warrant entering retention mode, but not long enough or not the type of inactivity to make the SoC place the L3 and SF in static retention. In addition to the static retention modes, the control logic can be powered down from the NOL3 state, at which point the CCN-502 is fully off.

All activity that is required to enable safe transition between the respective power states is performed automatically by the HN-Fs in response to input P-Channel P-state transitions. No additional activity is required of the SoC logic to enable transitions between power states. For example, the HN-F performs clean and invalidation of half of the ways of the L3 and clean and invalidation of all ways of the L3, as required by the respective power state transitions.

——————— Note ———————

The power controller cannot make any power transitions while the control logic is powered off. For example, if the power controller wants to transition from FAM static retention to OFF, it must transition through the FAM and NOL3 power states. This is because the RAMs must be flushed before they are powered down.

———————————————————

2.14.4 P-Channel

Each power domain in the CCN-502 includes a separate P-Channel for control of that domain. The P-Channel is a simple power-controller-to-device interface that manages device power states.

The P-Channel interface has the following features:

- Power state transitions are managed by the power controller.
- The device can optionally indicate a hint for an opportunistic state transition.
- The device can optionally deny a power state transition.
- Robust clock domain crossing semantics enable safe asynchronous interfacing.

This protocol is a generic way to request a transition to a particular state using a request-acknowledge 4-phase handshake. The specific state transitions of the device under management are not restricted by the P-Channel protocol, but might be restricted by the capabilities of the device, as they are in the CCN-502.

The P-Channel contains the following signals, where * is an identifier for a power domain:

PREQ_*   Indicates a request for a power state transition.

PSTATE_*[n−1:0]   The power state to which a transition is requested.

PACCEPT_*   Indicates acknowledgment of the power state transition and completion of the power state transition in the device.

PDENY_*   Indicates denial of the power state transition.

PACTIVE_*   A hint signal that indicates opportunistic power state transitions such as dynamic retention modes. The signal name and state transition hint are defined by the device implementation.

The following figure shows the signals and their connections.
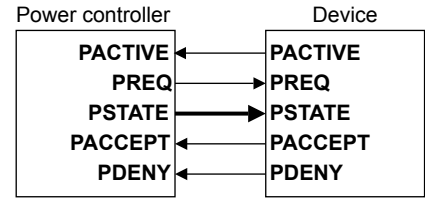
Figure 2-16 P-Channel interface with ACTIVE hint

The following sections describe the P-Channel:

P-Channel protocol

P-Channel protocol rules control P-Channel state transition.

The P-Channel protocol is as follows:

- **PREQ** can only transition from LOW to HIGH when **PACCEPT** and **PDENY** are both LOW.
- **PREQ** can only transition from HIGH to LOW when either:
  — **PACCEPT** is HIGH and **PDENY** is LOW.
  — **PACCEPT** is LOW and **PDENY** is HIGH.
- **PSTATE** can only transition when **PREQ**, **PACCEPT**, and **PDENY** are LOW. The signal transition must be guaranteed to be complete, and metastability resolved, when **PREQ** is asserted or **RESETn** is deasserted.
- **PACCEPT** can only transition from LOW to HIGH when **PREQ** is HIGH and **PDENY** is LOW.
- **PACCEPT** can only transition from HIGH to LOW when **PREQ** is LOW and **PDENY** is LOW.
- **PDENY** can only transition from LOW to HIGH when **PREQ** is HIGH and **PACCEPT** is LOW.
- **PDENY** can only transition from HIGH to LOW when **PREQ** is LOW and **PACCEPT** is LOW.

P-Channel state transition

This section describes the 4-phase handshake of the P-Channel.

The following figure shows a basic state transition timing diagram.
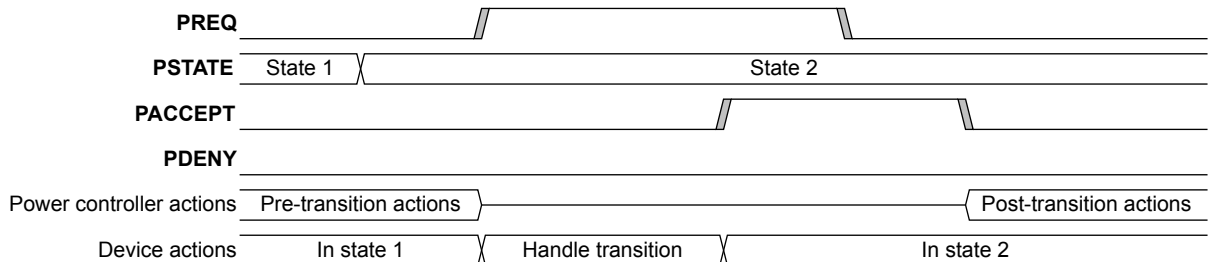
Figure 2-17 State transition timing diagram

The state transition uses the following 4-phase handshake:

1. The power controller drives the required state on **PSTATE**.
2. When it is guaranteed that this signal is stable, the power controller asserts **PREQ**.
3. The device asserts **PACCEPT**. If the state transition requires any actions from the device, such as cache initialization, the device must complete the action before it asserts **PACCEPT**.
4. The power controller responds by deasserting **PREQ**, and the device finishes by deasserting **PACCEPT**.

P-Channel on device reset

This section shows how to initialize the power state of a power domain.

The following figure shows the state initialization on reset. Certain device power states might power down the control logic. When powering this control logic back on, the power controller must indicate the state that the device must power up. The device detects the required state by sampling **PSTATE** when **RESETn** deasserts. The **PSTATE** inputs must be asserted before the deassertion of reset and remain after the deassertion of **RESETn** to allow reset propagation. The power controller must ensure that the reset sequence has completed before transitioning **PSTATE**, otherwise the device might sample an undetermined value.
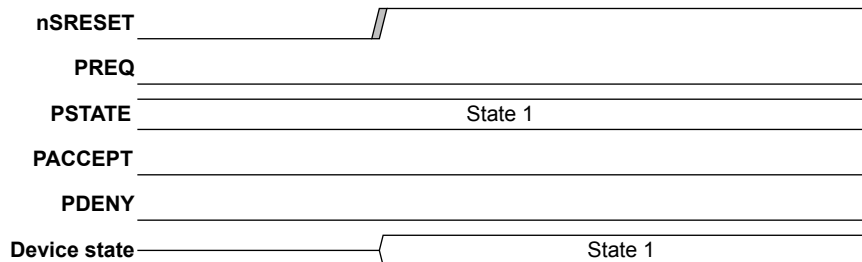
Figure 2-18 Reset state initialization

P-Channel interfaces

This section describes the various P-Channel interfaces in the CCN-502.

The CCN-502 power states are managed by the following P-Channel interfaces:

LOGIC   Controls the power state of the logic domain.

SF   Controls the power state of the snoop filter.

L3RAM0 Controls the power state of the L3 tag/data RAMs way[7:0].

L3RAM1 Controls the power state of the L3 tag/data RAMs way[15:0].

DEV   An optional interface that controls the power state of the optional DMC and processor DSSBs. These interfaces can be controlled uniquely, and are used to control the ON/OFF power states of the CCN502_SNF_DSSB and CCN502_RNF_DSSB blocks, after the respective DMC and processor cluster devices are powered down.

The P-Channel for the snoop filter, L3-RAM-Hi, and L3-RAM-Lo power domains is communicated to the 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-Fs by the PCCB. The PCCB acts as a distributor of P-Channel requests to the 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-Fs and aggregator of P-Channel responses from the 2 (6XP/2HNF) or 4 (8XP/4HNF) HN-Fs.

Related references

A.4 Power management signals on page Appx-A-266.

Power state transitions that require control of multiple P-Channels

The power controller must control multiple P-Channels for some power state transitions.

As *Figure 2-15 Power state transitions* on page 2-72 shows, some state transition arcs require P-state changes in multiple P-Channels, for example, to transition from NOL3 to FAM. There is no requirement for all P-Channels to simultaneously request the P-state change as indicated for these transitions, because the CCN-502 has internal control interlocks to ensure that it only allows transitions between valid power states. Therefore, the P-Channels can be independently controlled and might not be concurrently asserted, and the CCN-502 ensures that a required power state transition occurs only after receiving the last P-state request that leads to a valid power state.

The control mechanism ensures that transitions only occur between valid states. However, the control mechanism might enable transitions from one valid state to another by taking an unintended arc through another valid state, for example, in the transition from NOL3 to FAM. It is possible to transition between these two states in any of the following four sequences, depending on the perceived ordering of the P-Channel P-state transitions:

NOL3 to FAM
   The sequence is L3RAM1PSTATE=ON to L3RAM0PSTATE=ON to SFPSTATE=ON.
NOL3 to SFONLY to HAM to FAM
   The sequence is SFPSTATE=ON to L3RAM0PSTATE=ON to L3RAM1PSTATE=ON.
NOL3 to SFONLY to FAM
   The sequence is SFPSTATE=ON to L3RAM1PSTATE=ON to L3RAM0PSTATE=ON.
NOL3 to HAM to FAM
   The sequence is L3RAM0PSTATE=ON to SFPSTATE=ON to L3RAM1PSTATE=ON.

Although these are all valid state transition sequences and the CCN-502 guarantees correct functionality for any of these sequences, the most efficient sequence is the first. All other sequences result in multiple initialization sequences of some of the L3/SF RAMs. For the most efficient sequence, the P-state transition requests must be seen at the receiver in the order that guarantees exactly the required state transition. Although the P-Channel does not include a specific way of determining the order of arrival, the CCN-502 timing requirements are such that, if the P-Channel **PREQ** assertions on different P-Channels are separated by 15 or more clock cycles, those requests are guaranteed to be observed in that order at the receiver. For this reason, it is possible, and recommended, that you construct the P-Channel control mechanisms to ensure the most efficient transition between power states in the CCN-502.

Transitions to and from shutdown states

The power controller must satisfy certain conditions to enable the various shutdown state transitions.

There are two types of shutdown state transitions:

Transitions to a shutdown state
   - NOL3 to OFF.
   - SFONLY to SFONLY static retention.
   - HAM to HAM static retention.
   - FAM to FAM static retention.

Transitions from a shutdown state
   - SFONLY static retention to SFONLY.
   - HAM static retention to HAM.
   - FAM static retention to FAM.

The power controller must not perform any power transitions when the control logic is powered off. This means that:

- When transitioning from a shutdown state to a functional state, which includes a transition of LOGICPSTATE from OFF to ON, that transition must have been initiated and must be complete, that is, it must have received **PACCEPT**, before a transition of any of the other P-Channels can be initiated. This ensures that the logic required to complete transition of the other P-Channels is powered up to enable the transition.
- When transitioning to a shutdown state, which includes a transition of LOGICPSTATE from ON to OFF, transitions of all other P-Channels, as applicable, must have been initiated and must be complete, that is, must have received **PACCEPT**, before the transition of the LOGIC P-Channel can be initiated. This ensures that the logic required to complete transition of other P-Channels is powered up to enable the transition.

   In addition, when transitioning to a shutdown state, the control logic issues a **PDENY** to a **PREQ** on the LOGIC P-Channel if there is any outstanding traffic in the CCN-502. After receiving the **PDENY**, it is the responsibility of the power controller to undo the P-Channel transitions that have already been accepted. For example, in response to a **PDENY** on the LOGIC P-Channel during a

HAM to HAM static retention transition, the power controller must then issue P-Channel transitions to SFPSTATE=ON and L3RAM0PSTATE=ON.

The **PDENY** response only occurs if there is ongoing activity in the CCN-502, therefore it is possible to avoid the **PDENY** response if the CCN-502 is fully quiesced, that is, contains no in-flight transactions of any kind, before the power controller initiates a LOGIC P-Channel request to enter a shutdown state.

PSTATE on reset

Not all PSTATEs are available for the power controller to use when the control logic is reset.

The CCN-502 enables entry into four power states when the control logic is reset, with the following restrictions on the power controller:

SFONLY static retention
- LOGICPSTATE = OFF.
- SFPSTATE = MEM_RET.
- L3RAM0PSTATE = OFF.
- L3RAM1PSTATE = OFF.

HAM static retention
- LOGICPSTATE = OFF.
- SFPSTATE = MEM_RET.
- L3RAM0PSTATE = MEM_RET.
- L3RAM1PSTATE = OFF.

FAM static retention
- LOGICPSTATE = OFF.
- SFPSTATE = MEM_RET.
- L3RAM0PSTATE = MEM_RET.
- L3RAM1PSTATE = MEM_RET.

FAM
- LOGICPSTATE = ON.
- SFPSTATE = ON.
- L3RAM0PSTATE = ON.
- L3RAM1PSTATE = ON.

All **PSTATE** signals must be asserted at deassertion of reset. All **PREQ** signals must be LOW at the deassertion of reset. Any P-state values other than those described here are invalid and can result in unpredictable behavior.

Related references

P-Channel on device reset on page 2-75.

2.14.5 L3 data RAM retention control

This section describes how to use the **l3_reten_hx** signal when the RAM is in retention mode.

The HN-F L3 data RAM quadwords have an input, **l3_reten_hx**, that can be used for dynamic retention control. This signal asserts four cycles before the data RAMs are accessed, either read or write, and is held for the duration of the access, accounting for RAM latency. This enables the RAMs to be put in a retention mode, provided the 4-cycle wakeup is sufficient to exit retention mode and allow a read or write.

The following figure shows the **l3_reten_hx** signal behavior for a single L3 data RAM read. It asserts four cycles before the RAM read enable, and is held for the duration of the RAM read, three cycles after the RAM read enable in this case, showing the behavior of 3-cycle data RAMs.
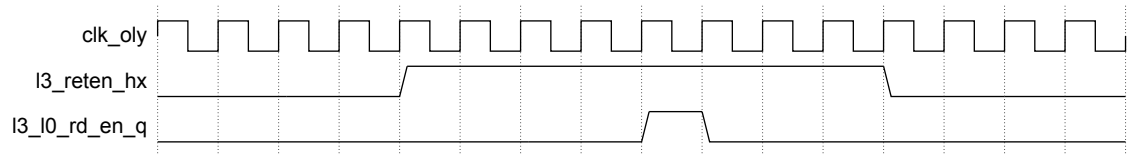
Figure 2-19 l3_reten_hx timing for single L3 data RAM read

2.15   Link layer

The CCN-502 provides link initialization, operation, and deactivation functionality for the link layer.

This functionality consists of the following mechanisms:
- A receiving device that communicates link layer credits to a transmitting device immediately following reset.
- A flow-control mechanism, where a receiving device can transmit credits and a transmitting device can consume credits during functional operation.
- A mechanism, where a transmitting device can return all unused link layer credits to the adjacent receiving device, enabling clock stop or power down of either device using that link.

See the *ARM® AMBA® 5 CHI Architecture Specification* for a description of the functional requirements of the CHI link layer.

2.16 Data integrity

The CCN-502 implements byte parity protection on all data buses and data storage, as well as parity generation and checking for sources and endpoints of data within CCN-502.

The sources of data include L3 cache read data and all RN-F DAT VC inputs, that is, write data and snoop response data.

The data endpoints include L3 cache write data and RN-F DAT VC outputs, that is, read data.

In addition, byte parity information is propagated to all the RN-I, HN-I, and SBSX RDATA and WDATA interfaces, using the **RUSER** and **WUSER** signals.

2.16.1 Parity error reporting, poisoning, and logging

Byte parity checking is performed at the CCN-502 interface for all DAT flits that target RN-Fs, and the CCN-502 generates a DAT flit Data Error response when it detects a parity error. In addition, error information is logged in the XP Error Syndrome 0 register.

Byte parity is also checked on L3 fill data. If the CCN-502 detects a parity error, it modifies the ECC data for the fill so that a double-bit ECC error occurs on the next read of that cacheline. In addition, error information is logged in the HN-F Error Syndrome 1 register.

Related references

Error Syndrome 0 register, XP on page 3-135.
Error Syndrome 1 register, L3 cache on page 3-160.

2.16.2 Byte parity error injection

The CCN-502 has error injection logic in the XPs and HN-Fs. For test purposes, software can use that logic to inject byte parity errors in the data stream and test that the software error routine can correctly process the parity errors.

Software can inject byte parity errors when the CCN-502 generates byte parity. The XP can inject byte parity errors in the RN-F DAT VC input data and the HN-F can inject byte parity errors in the L3 cache read data.

After the XP Byte Parity Error Injection register is written, the XP injects a parity error on the specified byte lane of the next valid DAT flit sent from the RN-F.

After the HN-F Byte Parity Error Injection register is written, the HN-F injects a parity error on the specified byte lane for the 4 DAT flits that are associated with the next L3 hit.

Related references

Byte Parity Error Injection register, XP on page 3-138.
HN-F Byte Parity Error Injection register on page 3-153.

Chapter 3
Programmers Model

This chapter describes the programmers model.

It contains the following sections:

3.1 About the programmers model

The following information applies to the CCN-502 registers:
- The base address is not fixed, and can be different for any particular system implementation. The offset of each register from the base address is fixed.
- All CCN-502 registers are 64-bit.
- 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 tables in *3.2 Register summary* on page 3-87 describe access types as follows:

RW   Read and write.

RO   Read only.

WO   Write only.

RAZ   Read as zero.

WI   Write ignored.

3.3 Register descriptions on page 3-93 describe the configuration registers included in each type of component in a CCN-502 system. The overall configuration register space is determined by the specific product implementation, with inclusion or exclusion of protocol bridges being the main distinction.

Each of the register descriptions is identical for all instances of that component, except for the identification registers. Each component contains all the registers included in its type, but the register space for each component is contained within a 64KB region specific to that component.

The exception to the identical register space across all instances of a specific component is in the node_id field in the identification registers (*_oly_id) included in all CCN-502 components. This field is different for each component instance in a CCN-502 system. The value for the node_id field in the component registers is set to `0x0`, although the actual value for the specific component instances can differ, depending on the configuration and topology.

This section contains the following subsections:
- *3.1.1 Node configuration register address mapping* on page 3-82.
- *3.1.2 Node type IDs* on page 3-85.
- *3.1.3 Requirements of configuration register reads and writes* on page 3-86.

3.1.1 Node configuration register address mapping

The CCN-502 requires 16MB of address space, split into 256 subregions of 64KB each.

The subregions have the following characteristics:
- Each of these subregions corresponds to a specific CCN-502 component, for example, MN, HN-F, or RN-I.
- The subregions are organized by the type of block, with the 64K offsets shown in the following table.

Table 3-1 Node region mapping

Region	Owner node
6XP/2HNF	
0	MN
1	Debug
8	HN-I

Table 3-1 Node region mapping (continued)

Region	Owner node
16-17	SBSX
32-33	HN-F
64-69	XP
128-130	RN-I
8XP/4HNF	
0	MN
1	Debug
8	HN-I
16-19	SBSX
32-35	HN-F
64-71	XP
128-130	RN-I

———— **Note** ————

Not all subregions that are listed in the following table are necessarily populated in a CCN-502 instantiation.

There are only as many valid subregions as there are components, and the region for the HN-F, HN-I, SBSX, and XP component types is calculated using the region base that the following table shows. Each successive valid component of that type, in ascending NodeID order, increments the region number. The subregions for SBSX are in ascending order, starting with PERIPHBASE + `0x10000`, in configurations where SBSXs are depopulated.

The region offset for the RN-I components is calculated as (128 + NodeID of RN-I).

The following table shows the valid regions for the CCN-502.

Table 3-2 Node register regions

NodeID or XP ID	Component	Region	Region base address
6XP/2HNF			
0	MN	0	PERIPHBASE
0	DT	1	PERIPHBASE + `0x10000`
0	HN-I	8	PERIPHBASE + `0x80000`
2	SBSX	16	PERIPHBASE + `0x100000`
8		17	PERIPHBASE + `0x110000`
3	HN-F	32	PERIPHBASE + `0x200000`
9		33	PERIPHBASE + `0x210000`

Table 3-2 Node register regions (continued)

NodeID or XP ID	Component	Region	Region base address
0	XP	64	PERIPHBASE + 0x400000
1		65	PERIPHBASE + 0x410000
2		66	PERIPHBASE + 0x420000
3		67	PERIPHBASE + 0x430000
4		68	PERIPHBASE + 0x440000
5		69	PERIPHBASE + 0x450000
4	RN-I	132	PERIPHBASE + 0x840000
6		134	PERIPHBASE + 0x860000
10		138	PERIPHBASE + 0x8A0000
8XP/4HNF			
0	MN	0	PERIPHBASE
0	DT	1	PERIPHBASE + 0x10000
0	HN-I	8	PERIPHBASE + 0x80000
2	SBSX	16	PERIPHBASE + 0x100000
4		17	PERIPHBASE + 0x110000
10		18	PERIPHBASE + 0x120000
12		19	PERIPHBASE + 0x130000
3	HN-F	32	PERIPHBASE + 0x200000
5		33	PERIPHBASE + 0x210000
11		34	PERIPHBASE + 0x220000
13		35	PERIPHBASE + 0x230000
0	XP	64	PERIPHBASE + 0x400000
1		65	PERIPHBASE + 0x410000
2		66	PERIPHBASE + 0x420000
3		67	PERIPHBASE + 0x430000
4		68	PERIPHBASE + 0x440000
5		69	PERIPHBASE + 0x450000
6		70	PERIPHBASE + 0x460000
7		71	PERIPHBASE + 0x470000
6	RN-I	134	PERIPHBASE + 0x860000
8		136	PERIPHBASE + 0x880000
14		142	PERIPHBASE + 0x8E0000

Related references

XP Identification register on page 3-141.

3.1.2   Node type IDs

Each 64K subregion in the CCN-502 system register map includes an oly_id field in the applicable component Identification register that identifies the node type of the owner for that specific 64K subregion.

The following table shows how the IDs are mapped to node types.

Table 3-3 Mapping of ID to node type

ID	Node type
0x00	Invalid node
0x01	MN
0x02	DT
0x03	Reserved
0x04	HN-F
0x05	HN-I
0x06-0x07	Reserved
0x08	XP
0x09-0x0B	Reserved
0x0C	SBSX
0x0D-0x13	Reserved
0x14	RN-I with 1 ACE-Lite interface
0x15	RN-I with 2 ACE-Lite interfaces
0x16	RN-I with 3 ACE-Lite interfaces
0x17	Reserved
0x18	RN-I with 1 ACE-Lite+DVM interface
0x19	RN-I with 2 ACE-Lite+DVM interfaces
0x1A	RN-I with 3 ACE-Lite+DVM interfaces
0x1B-0x1F	Reserved

Related references

3.1.3 Requirements of configuration register reads and writes

Reads and writes to the CCN-502 configuration registers must meet certain requirements.

If the following requirements are not met then this can result in unpredictable behavior.

- All accesses must be of device type, either:
  — Device, Strongly Ordered.
  — nGnRE, nGnRnE.
- All accesses must have a data size of 32 bits or 64 bits.
- All accesses must be natively aligned, that is:
  — 32-bit accesses must be aligned to a 32-bit boundary.
  — 64-bit accesses must be aligned to a 64-bit boundary.
- For configuration register writes, all bits, 32 or 64, must be written, that is, all byte lanes must be valid:
  — **WRSTB** must indicate that all bytes lanes are valid if the write transaction is from an AMBA interface.
  — **BE** must indicate that all byte lanes are valid if the write transaction is sent from a CHI interface.
- Secure registers can only be accessed by a Secure access, that is, NS = 0b0. Non-secure registers can be accessed by either a Secure or Non-secure access.

3.2 Register summary

The register summary tables list the registers in the CCN-502.

MN register summary

The following table shows the *Miscellaneous Node* (MN) registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-4 MN register summary

Offset	Name	Type	Description
0x0000	secure_access	RW	*Secure Access register* on page 3-94
0x0008	errint_status	RW	*Error Interrupt Status register* on page 3-95
0x0180	oly_rnf_nodeid_list	RO	*RN-F Node ID register* on page 3-96
0x0190	oly_rni_nodeid_list	RO	*RN-I Node ID register* on page 3-97
0x01A0	oly_rnidvm_nodeid_list	RO	*RN-D Node ID register* on page 3-97
0x01B0	oly_hnf_nodeid_list	RO	*HN-F Node ID register* on page 3-98
0x01C0	oly_hni_nodeid_list	RO	*HN-I Node ID register* on page 3-98
0x01D0	oly_sn_nodeid_list	RO	*SN Node ID register* on page 3-99
0x01E0	oly_comp_list_63_0	RO	*Component List [63:0] register* on page 3-100
0x01E8	oly_comp_list_127_64	RO	*Component List [127:64] register* on page 3-100
0x01F0	oly_comp_list_191_128	RO	*Component List [191:128] register* on page 3-101
0x01F8	oly_comp_list_255_192	RO	*Component List [255:192] register* on page 3-101
0x0200	dvm_domain_ctl	RO	*DVM Domain Control register* on page 3-102
0x0210	dvm_domain_ctl_set	WO	*DVM Domain Control Set register* on page 3-102
0x0220	dvm_domain_ctl_clr	WO	*DVM Domain Control Clear register* on page 3-103
0x0300	err_sig_val_63_0	RO	*Error Signal Valid [63:0] register* on page 3-103
0x0308	err_sig_val_127_64	RO	*Error Signal Valid [127:64] register* on page 3-104
0x0310	err_sig_val_191_128	RO	*Error Signal Valid [191:128] register* on page 3-105
0x0320	err_type_31_0	RO	*Error Type Value [31:0] register* on page 3-105
0x0328	err_type_63_32	RO	*Error Type Value [63:32] register* on page 3-106
0x0330	err_type_95_64	RO	*Error Type Value [95:64] register* on page 3-107
0x0340	err_type_159_128	RO	*Error Type Value [159:128] register* on page 3-108
0x0FD0	periph_id_4_periph_id_5	RO	*Peripheral ID 4 and Peripheral ID 5 register* on page 3-108
0x0FD8	periph_id_6_periph_id_7	RO	*Peripheral ID 6 and Peripheral ID 7 register* on page 3-109
0x0FE0	periph_id_0_periph_id_1	RO	*Peripheral ID 0 and Peripheral ID 1 register* on page 3-109
0x0FE8	periph_id_2_periph_id_3	RO	*Peripheral ID 2 and Peripheral ID 3 register* on page 3-110
0x0FF0	component_id_0_component_id_1	RO	*Component ID 0 and Component ID 1 register* on page 3-111

Table 3-4 MN register summary (continued)

Offset	Name	Type	Description
`0x0FF8`	component_id_2_component_id_3	RO	*Component ID 2 and Component ID 3 register* on page 3-111
`0xFF00`	oly_mn_oly_id	RO	*MN Identification register* on page 3-112

XP register summary

The following table shows the *crosspoint* (XP) registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-5 XP register summary

Offset	Name	Type	Description
`0x0000`	xp_routing_control	RW	*XP Routing Control register* on page 3-113
`0x0008`	dev0_nsm_routing_vector	RW	*XP Device 0 Port NSM Routing register* on page 3-114
`0x0010`	dev1_nsm_routing_vector	RW	*XP Device 1 Port NSM Routing register* on page 3-114
`0x0110`	dev0_qos_control	RW	*Device 0 Port QoS Control register* on page 3-115
`0x0118`	dev0_qos_lat_tgt	RW	*Device 0 Port QoS Latency Target register* on page 3-116
`0x0120`	dev0_qos_lat_scale	RW	*Device 0 Port QoS Latency Scale register* on page 3-117
`0x0128`	dev0_qos_lat_range	RW	*Device 0 Port QoS Latency Range register* on page 3-117
`0x0210`	dev1_qos_control	RW	*Device 1 Port QoS Control register* on page 3-118
`0x0218`	dev1_qos_lat_tgt	RW	*Device 1 Port QoS Target Latency register* on page 3-119
`0x0220`	dev1_qos_lat_scale	RW	*Device 1 Port QoS Latency Scale register* on page 3-119
`0x0228`	dev1_qos_lat_range	RW	*Device 1 Port QoS Latency Range register* on page 3-120
`0x0300`	dt_config	RW	*Debug and Trace Configuration register* on page 3-121
`0x0308`	dt_interface_sel	RW	*Debug and Trace Interface Select register* on page 3-122
`0x0310`	dt_cmp_val0_l	RW	*Debug and Trace Comparison Low Value 0 register* on page 3-123
`0x0318`	dt_cmp_val0_h	RW	*Debug and Trace Comparison High Value 0 register* on page 3-124
`0x0320`	dt_cmp_mask0_l	RW	*Debug and Trace Comparison Low Mask 0 register* on page 3-125
`0x0328`	dt_cmp_mask0_h	RW	*Debug and Trace Comparison High Mask 0 register* on page 3-126
`0x0350`	dt_cmp_val1_l	RW	*Debug and Trace Comparison Low Value 1 register* on page 3-127
`0x0358`	dt_cmp_val1_h	RW	*Debug and Trace Comparison High Value 1 register* on page 3-128
`0x0360`	dt_cmp_mask1_l	RW	*Debug and Trace Comparison Low Mask 1 register* on page 3-129
`0x0368`	dt_cmp_mask1_h	RW	*Debug and Trace Comparison High Mask 1 register* on page 3-130
`0x0370`	dt_control	RW	*Debug and Trace Control register, dt_control* on page 3-131
`0x0378`	dt_status	RW	*Debug and Trace Status register* on page 3-134
`0x0380`	dt_status_clr	WO	*Debug and Trace Status Clear register* on page 3-134
`0x0400`	err_syndrome_reg0	RO	*Error Syndrome 0 register, XP* on page 3-135
`0x0480`	err_syndrome_clr	WO	*XP Error Syndrome Clear register* on page 3-136

Table 3-5 XP register summary (continued)

Offset	Name	Type	Description
0x0500	aux_ctl	RW	*Auxiliary Control register, XP* on page 3-137
0x0508	byte_par_err_inj	WO	*Byte Parity Error Injection register, XP* on page 3-138
0x0600	pmu_event_sel	RW	*PMU Event Select register, XP* on page 3-139
0xFF00	oly_xp_oly_id	RO	*XP Identification register* on page 3-141

HN-F register summary

The following table shows the *Fully-coherent Home Node* (HN-F) registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping on page 3-82* for information about individual region base addresses.

Table 3-6 HN-F register summary

Offset	Name	Type	Description
0x0000	hnf_cfg_ctrl	RW	*HN-F Configuration Control register* on page 3-142
0x0008	hnf_sam_control	RW	*HN-F SAM Control register* on page 3-143
0x0010	hn_cfg_pstate_req	WO	*HN-F P-state Request register* on page 3-144
0x0018	hn_cfg_pstate_status	RO	*HN-F P-state Status register* on page 3-145
0x0020	qos_band	RO	*QoS Band register* on page 3-146
0x0028	qos_reservation	RW	*QoS Reservation register* on page 3-147
0x0030	rn_starvation	RW	*RN Starvation register* on page 3-148
0x0038	hnf_err_inj	RW	*HN-F Error Injection Enable and Setup register* on page 3-149
0x0040	hnf_l3_lock_ways	RW	*HN-F L3 Lock Ways register* on page 3-150
0x0048	hnf_l3_lock_base0	RW	*HN-F L3 Lock Base 0 register* on page 3-151
0x0050	hnf_l3_lock_base1	RW	*HN-F L3 Lock Base 1 register* on page 3-151
0x0058	hnf_l3_lock_base2	RW	*HN-F L3 Lock Base 2 register* on page 3-152
0x0060	hnf_l3_lock_base3	RW	*HN-F L3 Lock Base 3 register* on page 3-152
0x0068	hnf_byte_par_err_inj	WO	*HN-F Byte Parity Error Injection register* on page 3-153
0x0108	hn_cfg_rni_vec	RW	*HN Configuration RN-I Vector register* on page 3-154
0x0200	snoop_domain_ctl	RO	*Snoop Domain Control register* on page 3-154
0x0210	snoop_domain_ctl_set	WO	*Snoop Domain Control Set register* on page 3-155
0x0220	snoop_domain_ctl_clr	WO	*Snoop Domain Control Clear register* on page 3-156
0x0300	hn_cfg_l3sf_dbgrd	WO	*HN Debug Read Configuration register* on page 3-156
0x0308	l3_cache_access_l3_tag	RO	*L3 Cache Access Tag register* on page 3-157
0x0310	l3_cache_access_l3_data	RO	*L3 Cache Access Data register* on page 3-158
0x0318	l3_cache_access_sf_tag	RO	*L3 Cache Access SF Tag register* on page 3-158
0x0400	err_syndrome_reg0	RO	*Error Syndrome 0 register, L3 cache* on page 3-159
0x0408	err_syndrome_reg1	RO	*Error Syndrome 1 register, L3 cache* on page 3-160

Table 3-6 HN-F register summary (continued)

Offset	Name	Type	Description
0x0480	err_syndrome_clr	WO	*L3 cache Error Syndrome Clear register* on page 3-160
0x0500	hnf_aux_ctl	RW	*HN-F Auxiliary Control register* on page 3-161
0x0600	pmu_event_sel	RW	*PMU Event Select register, L3 cache* on page 3-162
0xFF00	oly_hnf_misc_oly_id	RO	*HN-F Identification register* on page 3-164

HN-I register summary

The following table shows the *I/O Home Node* (HN-I) registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-7 HN-I register summary

Offset	Name	Type	Description
0x0000	pos_control	RW	*PoS Control register* on page 3-165
0x0008	pcierc_rni_nodeid_list	RW	*PCIeRC RN-I Node ID List register* on page 3-166
0x0400	err_syndrome_reg0	RO	*Error Syndrome 0 register, HN-I* on page 3-166
0x0408	err_syndrome_reg1	RO	*Error Syndrome 1 register, HN-I* on page 3-167
0x0480	err_syndrome_clr	WO	*HN-I Error Syndrome Clear register* on page 3-168
0x0500	sa_aux_ctl	RW	*SA Auxiliary Control register, HN-I* on page 3-169
0xFF00	oly_hni_oly_id	RO	*HN-I Identification register* on page 3-170

Debug event module register summary

The following table shows the debug event module registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-8 Debug event module register summary

Offset	Name	Type	Description
0x0000	active_dsm	RW	*Active DSM register* on page 3-172
0x0008	trigger_ctl	RW	*Trigger Control register* on page 3-173
0x0010	trigger_status	RW	*Trigger Status register* on page 3-173
0x0018	trigger_status_clr	WO	*Trigger Status Clear register* on page 3-174
0x0020	timer_val	RW	*Timer Value register* on page 3-174
0x0028	dt_ctl	RW	*Debug and Trace Control register, dt_ctl* on page 3-175
0x0080	dbg_id	RW	*Debug Identification register* on page 3-176
0x0100	pmevcnt0	RW	*PMU Event Counter 0 register* on page 3-176
0x0108	pmevcnt1	RW	*PMU Event Counter 1 register* on page 3-177
0x0110	pmevcnt2	RW	*PMU Event Counter 2 register* on page 3-177
0x0118	pmevcnt3	RW	*PMU Event Counter 3 register* on page 3-178

Table 3-8 Debug event module register summary (continued)

Offset	Name	Type	Description
0x0120	pmevcnt4	RW	*PMU Event Counter 4 register* on page 3-178
0x0128	pmevcnt5	RW	*PMU Event Counter 5 register* on page 3-179
0x0130	pmevcnt6	RW	*PMU Event Counter 6 register* on page 3-179
0x0138	pmevcnt7	RW	*PMU Event Counter 7 register* on page 3-180
0x0140	pmccntr	RW	*PMU Cycle Counter register* on page 3-180
0x0150	pmevcntsr0	RW	*PMU Event Counter Shadow 0 register* on page 3-181
0x0158	pmevcntsr1	RW	*PMU Event Counter Shadow 1 register* on page 3-181
0x0160	pmevcntsr2	RW	*PMU Event Counter Shadow 2 register* on page 3-182
0x0168	pmevcntsr3	RW	*PMU Event Counter Shadow 3 register* on page 3-182
0x0170	pmevcntsr4	RW	*PMU Event Counter Shadow 4 register* on page 3-183
0x0178	pmevcntsr5	RW	*PMU Event Counter Shadow 5 register* on page 3-183
0x0180	pmevcntsr6	RW	*PMU Event Counter Shadow 6 register* on page 3-184
0x0188	pmevcntsr7	RW	*PMU Event Counter Shadow 7 register* on page 3-184
0x0190	pmccntrsr	RW	*PMU Cycle Counter Shadow register* on page 3-185
0x0198	pmovsr	RO	*PMU Overflow Status register* on page 3-185
0x01A0	pmovsr_clr	RW	*PMU Overflow Status Clear register* on page 3-186
0x01A8	pmcr	RW	*PMU Control register* on page 3-186
0x01B0	pmsr	RO	*PMU Status register* on page 3-187
0x01B8	pmsr_req	WO	*PMU Snapshot Request register* on page 3-188
0x01C0	pmsr_clr	WO	*PMU Snapshot Status Clear register* on page 3-188
0xFF00	oly_mn_dt_oly_id	RO	*Debug and Trace Identification register* on page 3-189

RN-I register summary

The following table shows the *I/O-coherent Requesting Node* (RN-I) bridge registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-9 RN-I bridge register summary

Offset	Name	Type	Description
0x0008	s0_port_control	RW	*Port S0 Control register, RN-I* on page 3-190
0x0010	s0_qos_control	RW	*Port S0 QoS Control register, RN-I* on page 3-191
0x0018	s0_qos_lat_tgt	RW	*Port S0 QoS Latency Target register, RN-I* on page 3-192
0x0020	s0_qos_lat_scale	RW	*Port S0 QoS Latency Scale register, RN-I* on page 3-193
0x0028	s0_qos_lat_range	RW	*Port S0 QoS Latency Range register, RN-I* on page 3-194
0x0108	s1_port_control	RW	*Port S1 Control register, RN-I* on page 3-194
0x0110	s1_qos_control	RW	*Port S1 QoS Control register, RN-I* on page 3-195

Table 3-9 RN-I bridge register summary (continued)

Offset	Name	Type	Description
0x0118	s1_qos_lat_tgt	RW	*Port S1 QoS Latency Target register, RN-I* on page 3-196
0x0120	s1_qos_lat_scale	RW	*Port S1 QoS Latency Scale register, RN-I* on page 3-197
0x0128	s1_qos_lat_range	RW	*Port S1 QoS Latency Range register, RN-I* on page 3-198
0x0208	s2_port_control	RW	*Port S2 Control register, RN-I* on page 3-198
0x0210	s2_qos_control	RW	*Port S2 QoS Control register, RN-I* on page 3-199
0x0218	s2_qos_lat_tgt	RW	*Port S2 QoS Latency Target register, RN-I* on page 3-201
0x0220	s2_qos_lat_scale	RW	*Port S2 QoS Latency Scale register, RN-I* on page 3-201
0x0228	s2_qos_lat_range	RW	*Port S2 QoS Latency Range register, RN-I* on page 3-202
0x0500	aux_ctl	RW	*RN-I Auxiliary Control register* on page 3-203
0x0600	pmu_event_sel	RW	*PMU Event Select register, RN-I* on page 3-204
0xFF00	oly_rni_oly_id	RO	*RN-I Identification register* on page 3-205

SBSX register summary

The following table shows the *CHI to AXI bridge* (SBSX) registers in offset order from the base memory address, **PERIPHBASE[43:24]**. See *3.1.1 Node configuration register address mapping* on page 3-82 for information about individual region base addresses.

Table 3-10 SBSX register summary

Offset	Name	Type	Description
0x0500	sa_aux_ctl	RW	*SA Auxiliary Control register, SBSX* on page 3-206
0xFF00	oly_sbsx_oly_id	RO	*SBSX Identification register* on page 3-206

3.3 Register descriptions

This section contains the following subsections:

3.3.1 MN register descriptions

This section lists the MN registers.

Secure Access register

The secure_access register is at offset `0x0000`. Its characteristics are:

Purpose	Permits a Non-secure access request to access Secure registers.
Usage constraints	Only accessible by Secure accesses.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the secure_access register bit assignments.
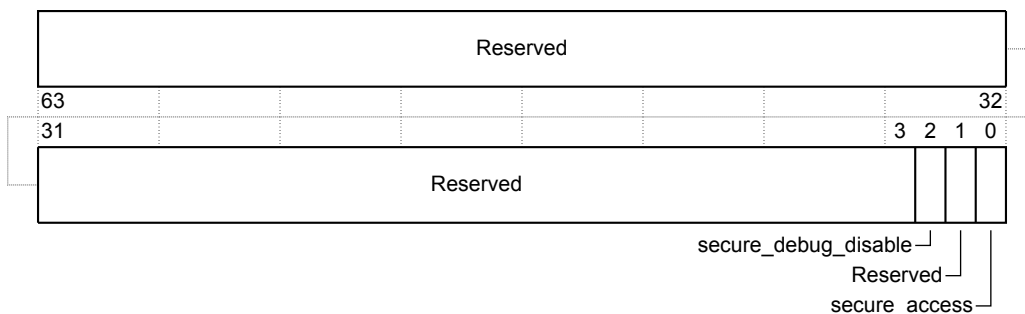
Figure 3-1 secure_access register bit assignments

The following table shows the secure_access register bit assignments.

Table 3-11 secure_access register bit assignments

Bits	Name	Access	Reset value	Description
[63:3]	-	RAZ/WI	0x0	Reserved
[2]	secure_debug_disable	RW	0	Secure debug disable: **1** If **SPNIDEN** is HIGH then Secure events are monitored by the PMU. **0** Secure events are monitored by the PMU.
[1]	-	RAZ/WI	0	Reserved
[0]	secure_access	RW	0	Secure access: **1** Enables Non-secure access to Secure registers. **0** Precludes Non-secure access to Secure registers.

Error Interrupt Status register

The errint_status register is at offset `0x0008`. Its characteristics are:

Purpose Disables interrupts and disables corrected error logging.

Usage constraints Only accessible by Secure accesses. Bits[3:0] control whether writes to bits[7:4] are successful.

Configurations Available in all configurations.

Attributes See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the errint_status register bit assignments.
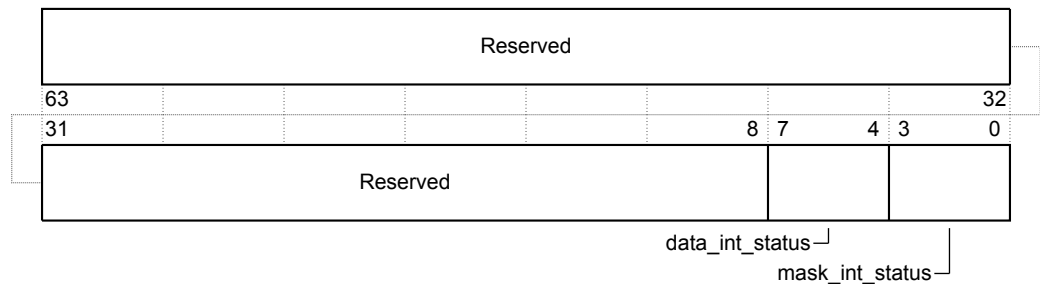
Figure 3-2 errint_status register bit assignments

The following table shows the errint_status register bit assignments.

Table 3-12 errint_status register bit assignments

Bits	Name	Access	Reset value	Description
[63:8]	-	RAZ/WI	0x0	Reserved
[7:4]	data_int_status	RW	0x0	A read returns the interrupt disable status: • 0 = Interrupt type is enabled. • 1 = Interrupt type is disabled. A write enables or disables the interrupt, provided the corresponding write enable (bits[3:0]) is set: **Bit[7]** 0 = Enable interrupt for PMU event. 1 = Disable interrupt for PMU event. **Bit[6]** 0 = Enable interrupt for corrected error. 1 = Disable interrupt for corrected error. **Bit[5]** 0 = Enable interrupt for all errors, including corrected errors. 1 = Disable interrupt for all errors, including corrected errors. **Bit[4]** 0 = Enable the **INTREQ** interrupt. 1 = Disable the **INTREQ** interrupt. If the interrupt is asserted, then it also deasserts the **INTREQ** interrupt signal.
[3:0]	mask_int_status	RW	0x0	These bits are write enables for the data_int_status bits, [7:4]. Always Read-As-Zero. **Bit[3]** Set to 1, to enable writes to data_int_status[7], the PMU event interrupt mask. **Bit[2]** Set to 1, to enable writes to data_int_status[6], the Corrected error mask. **Bit[1]** Set to 1, to enable writes to data_int_status[5], the All error mask. **Bit[0]** Set to 1, to enable writes to data_int_status[4], the **INTREQ** interrupt enable.

Related references

Error Interrupt Status register values on page 2-49.

RN-F Node ID register

The oly_rnf_nodeid_list register is at offset `0x0180`. Its characteristics are:

Purpose   A bit vector that indicates the RN-Fs in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an RN-F is present at that nodeID.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_rnf_nodeid_list register bit assignments.

Figure 3-3 oly_rnf_nodeid_list register bit assignments

The following table shows the oly_rnf_nodeid_list register bit assignments.

Table 3-13 oly_rnf_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_rnf_nodeid_list	RO	`0x8A2` (6XP/ 2HNF)	Bit vector of NodeIDs for RN-Fs. The value that is returned depends on the setting of the **RNFEN_NID<x>**. The reset value that is shown is for a CCN with a fully populated configuration of RN-Fs.
			`0x8282` (8XP/ 4HNF)	

RN-I Node ID register

The oly_rni_nodeid_list register is at offset `0x0190`. Its characteristics are:

Purpose   A bit vector that indicates the RN-Is in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an RN-I is present at that nodeID.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_rni_nodeid_list register bit assignments.
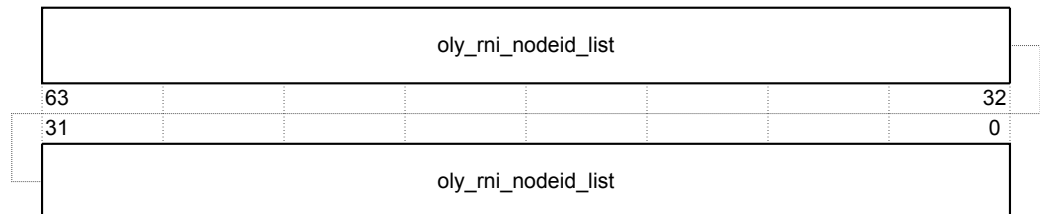
Figure 3-4 oly_rni_nodeid_list register bit assignments

The following table shows the oly_rni_nodeid_list register bit assignments.

Table 3-14 oly_rni_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_rni_nodeid_list	RO	Value depends on customer configuration	Bit vector of NodeIDs for RN-Is

RN-D Node ID register

The oly_rnidvm_nodeid_list register is at offset `0x01A0`. Its characteristics are:

Purpose   A bit vector that indicates the RN-Ds in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an RN-D is present at that nodeID.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_rnidvm_nodeid_list register bit assignments.

oly_rnidvm_nodeid_list
63   32
31   0
oly_rnidvm_nodeid_list
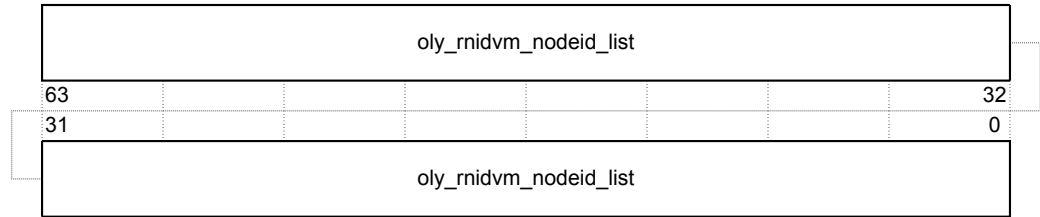
Figure 3-5 oly_rnidvm_nodeid_list register bit assignments

The following table shows the oly_rnidvm_nodeid_list register bit assignments.

Table 3-15 oly_rnidvm_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_rnidvm_nodeid_list	RO	Value depends on customer configuration	Bit vector of NodeIDs for RN-D bridges

HN-F Node ID register

The oly_hnf_nodeid_list register is at offset `0x01B0`. Its characteristics are:

Purpose   A bit vector that indicates the HN-Fs in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an HN-F is present at that nodeID.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_hnf_nodeid_list register bit assignments.

oly_hnf_nodeid_list
63   32
31   0
oly_hnf_nodeid_list
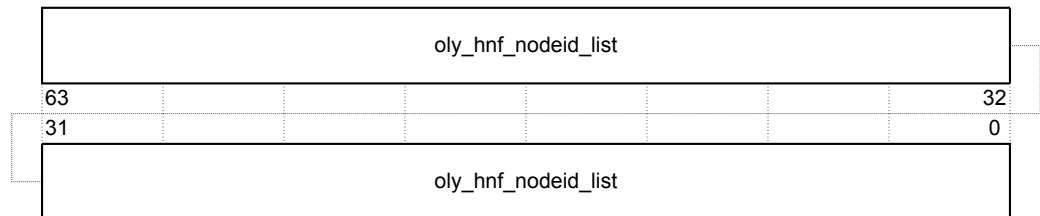
Figure 3-6 oly_hnf_nodeid_list register bit assignments

The following table shows the oly_hnf_nodeid_list register bit assignments.

Table 3-16 oly_hnf_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_hnf_nodeid_list	RO	0x208 (6XP/2HNF)	Bit vector of NodeIDs for HN-Fs.
			0x2828 (8XP/4HNF)	

HN-I Node ID register

The oly_hni_nodeid_list register is at offset `0x01C0`. Its characteristics are:

Purpose	A bit vector that indicates the HN-Is in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an HN-I is present at that nodeID.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_hni_nodeid_list register bit assignments.
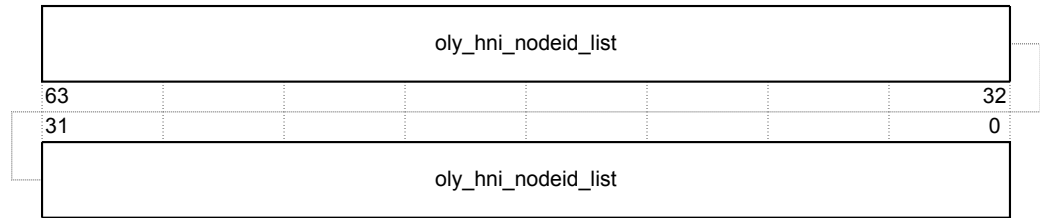
Figure 3-7 oly_hni_nodeid_list register bit assignments

The following table shows the oly_hni_nodeid_list register bit assignments.

Table 3-17 oly_hni_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_hni_nodeid_list	RO	0x1 (6XP/2HNF)	Bit vector of NodeIDs for HN-Is.
			0x1 (8XP/4HNF)	

SN Node ID register

The oly_sn_nodeid_list register is at offset `0x01D0`. Its characteristics are:

Purpose	A bit vector that indicates the SNs in the system. Each bit in the vector corresponds to a nodeID, with a 1'b1 indicating an SN is present at that nodeID.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_sn_nodeid_list register bit assignments.
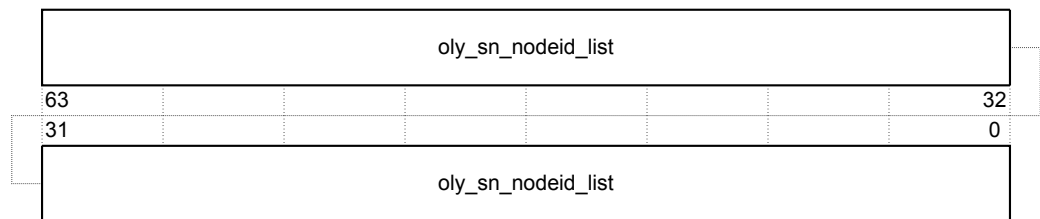
Figure 3-8 oly_sn_nodeid_list register bit assignments

The following table shows the oly_sn_nodeid_list register bit assignments.

Table 3-18 oly_sn_nodeid_list register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_sn_nodeid_list	RO	0x104 (6XP/2HNF)	Bit vector of NodeIDs for SN-F ports
			0x1414 (8XP/4HNF)	

Component List [63:0] register

The oly_cfg_comp_list_63_0 register is at offset `0x01E0`. Its characteristics are:

Purpose   Indicates the presence of valid components corresponding to configuration register regions 0-63.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_cfg_comp_list_63_0 register bit assignments.

Figure 3-9 oly_cfg_comp_list_63_0 register bit assignments

The following table shows the oly_cfg_comp_list_63_0 register bit assignments.

Table 3-19 oly_cfg_comp_list_63_0 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_cfg_comp_list_63_0	RO	Value depends on customer configuration	Indicates the presence of valid components corresponding to configuration register regions 0-63

Component List [127:64] register

The oly_cfg_comp_list_127_64 register is at offset `0x01E8`. Its characteristics are:

Purpose   Indicates the presence of valid components corresponding to configuration register regions 64-127.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_cfg_comp_list_127_64 register bit assignments.

Figure 3-10 oly_cfg_comp_list_127_64 register bit assignments

The following table shows the oly_cfg_comp_list_127_64 register bit assignments.

Table 3-20 oly_cfg_comp_list_127_64 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_cfg_comp_list_127_64	RO	Value depends on customer configuration	Indicates the presence of valid components corresponding to configuration register regions 64-127

Component List [191:128] register

The oly_cfg_comp_list_191_128 register is at offset `0x01F0`. Its characteristics are:

Purpose   Indicates the presence of valid components corresponding to configuration register regions 128-191.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_cfg_comp_list_191_128 register bit assignments.

```
┌─────────────────────────────────────────────────────────┐
│                 oly_cfg_comp_list_191_128                │
│ 63                                                    32 │
│ 31                                                     0 │
│                 oly_cfg_comp_list_191_128                │
└─────────────────────────────────────────────────────────┘
```

Figure 3-11 oly_cfg_comp_list_191_128 register bit assignments

The following table shows the oly_cfg_comp_list_191_128 register bit assignments.

Table 3-21 oly_cfg_comp_list_191_128 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_cfg_comp_list_191_128	RO	Value depends on customer configuration	Indicates the presence of valid components corresponding to configuration register regions 128-191

Component List [255:192] register

The oly_cfg_comp_list_255_192 register is at offset `0x01F8`. Its characteristics are:

Purpose   Indicates the presence of valid components corresponding to configuration register regions 192-255.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the oly_cfg_comp_list_255_192 register bit assignments.
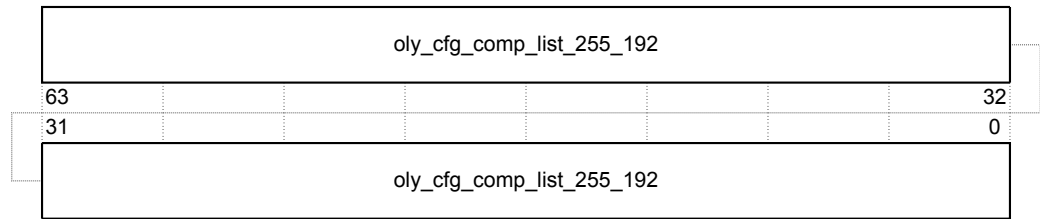
Figure 3-12 oly_cfg_comp_list_255_192 register bit assignments

The following table shows the oly_cfg_comp_list_255_192 register bit assignments.

Table 3-22 oly_cfg_comp_list_255_192 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	oly_cfg_comp_list_255_192	RO	Value depends on customer configuration	Indicates the presence of valid components corresponding to regions configuration register 192-255

DVM Domain Control register

The dvm_domain_ctl register is at offset `0x0200`. Its characteristics are:

Purpose	A bit vector that defines the RNs that must be sent and must respond to a DVMOp snoop from the MN. Each bit in the vector corresponds to a nodeID, and when a bit is set to 1 it indicates that an RN in the DVM domain is present at that nodeID.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

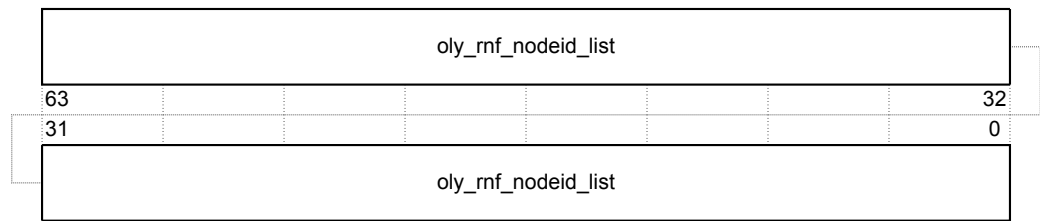The following figure shows the dvm_domain_ctl register bit assignments.

Figure 3-13 dvm_domain_ctl register bit assignments

The following table shows the dvm_domain_ctl register bit assignments.

Table 3-23 dvm_domain_ctl register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	dvm_domain_ctl	RO	0x0	Bit vector of NodeIDs for all RN-Fs and RN-Is that are active in the DVM domain. These RNs are devices that receive and must respond to DVMOps.

DVM Domain Control Set register

The dvm_domain_ctl_set register is at offset `0x0210`. Its characteristics are:

Purpose	A bit vector that controls which nodeIDs of RNs to insert into the active DVM domain. Each bit in the vector corresponds to a nodeID.
Usage constraints	Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the dvm_domain_ctl_set register bit assignments.

dvm_domain_ctl_set
63   32

31   0
dvm_domain_ctl_set

Figure 3-14 dvm_domain_ctl_set register bit assignments

The following table shows the dvm_domain_ctl_set register bit assignments.

Table 3-24 dvm_domain_ctl_set register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	dvm_domain_ctl_set	WO	0x0	Bit vector of NodeIDs of RNs to insert into the active DVM domain. Completion of insertion, results in the indicated RNs receiving and being required to respond to DVMOps.

DVM Domain Control Clear register

The dvm_domain_ctl_clr register is at offset `0x0220`. Its characteristics are:

Purpose   A bit vector that controls which nodeIDs of RNs to remove from the active DVM domain. Each bit in the vector corresponds to a nodeID.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the dvm_domain_ctl_clr register bit assignments.

dvm_domain_ctl_clr
63   32

31   0
dvm_domain_ctl_clr
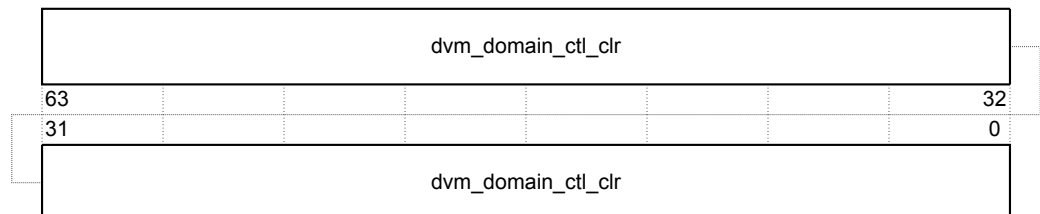
Figure 3-15 dvm_domain_ctl_clr register bit assignments

The following table shows the dvm_domain_ctl_clr register bit assignments.

Table 3-25 dvm_domain_ctl_clr register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	dvm_domain_ctl_clr	WO	0x0	Bit vector of NodeIDs of RNs to remove from the active DVM domain. Completion of removal, results in the indicated RNs no longer receiving nor being required to respond to DVMOps.

Error Signal Valid [63:0] register

The err_sig_val_63_0 register is at offset `0x0300`. Its characteristics are:

Purpose	Indicates an error in nodes [63:0].
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the err_sig_val_63_0 register bit assignments.
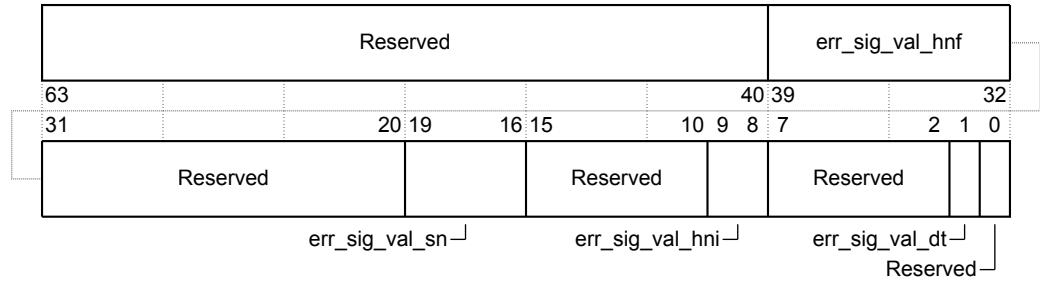
Figure 3-16 err_sig_val_63_0 register bit assignments

The following table shows the err_sig_val_63_0 register bit assignments.

Table 3-26 err_sig_val_63_0 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:32]	err_sig_val_hnf	RO	0x0	Indicates an HN-F error
[31:20]	-	RAZ/WI	0x0	Reserved
[19:16]	err_sig_val_sn	RO	0x0	Indicates an SN error
[15:10]	-	RAZ/WI	0x0	Reserved
[9:8]	err_sig_val_hni	RO	0x0	Indicates an HN-I error
[7:2]	-	RAZ/WI	0x0	Reserved
[1]	err_sig_val_dt	RO	0	Indicates a DT error
[0]	-	RO	0	Reserved

Error Signal Valid [127:64] register

The err_sig_val_127_64 register is at offset `0x0308`. Its characteristics are:

Purpose	Indicates an error in nodes [127:64].
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the err_sig_val_127_64 register bit assignments.
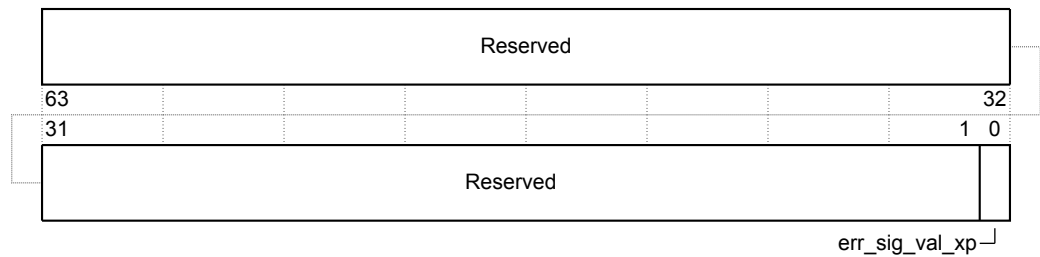
Figure 3-17 err_sig_val_127_64 register bit assignments

The following table shows the err_sig_val_127_64 register bit assignments.

Table 3-27 err_sig_val_127_64 register bit assignments

Bits	Name	Access	Reset value	Description
[63:1]	-	RAZ/WI	0x0	Reserved
[0]	err_sig_val_xp	RO	0	Indicates an XP error

Error Signal Valid [191:128] register

The err_sig_val_191_128 register is at offset `0x0310`. Its characteristics are:

Purpose   Indicates an error in nodes [191:128].

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the err_sig_val_191_128 register bit assignments.
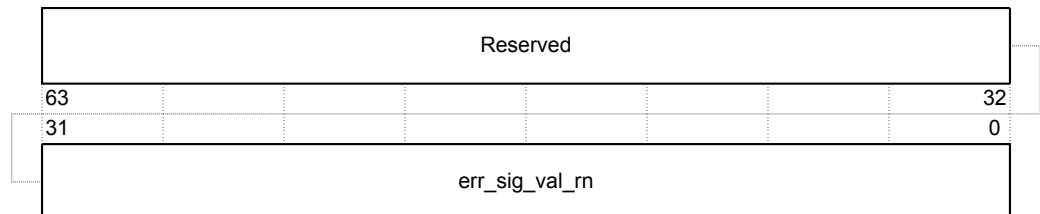
Figure 3-18 err_sig_val_191_128 register bit assignments

The following table shows the err_sig_val_191_128 register bit assignments.

Table 3-28 err_sig_val_191_128 register bit assignments

Bits	Name	Access	Reset value	Description
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	err_sig_val_rn	RO	0x0	Indicates an RN interface error

Error Type Value [31:0] register

The err_type_31_0 register is at offset `0x0320`. Its characteristics are:

Purpose   Indicates the type of error in nodes [31:0].

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

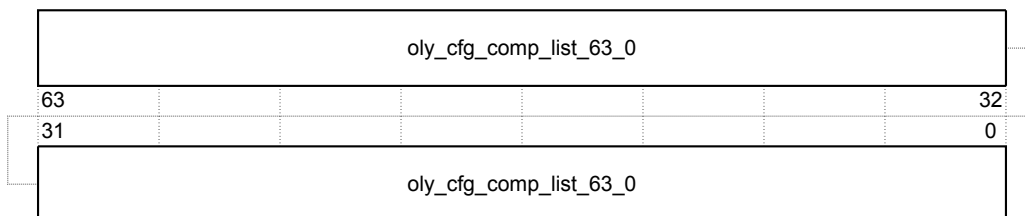The following figure shows the err_type_31_0 register bit assignments.
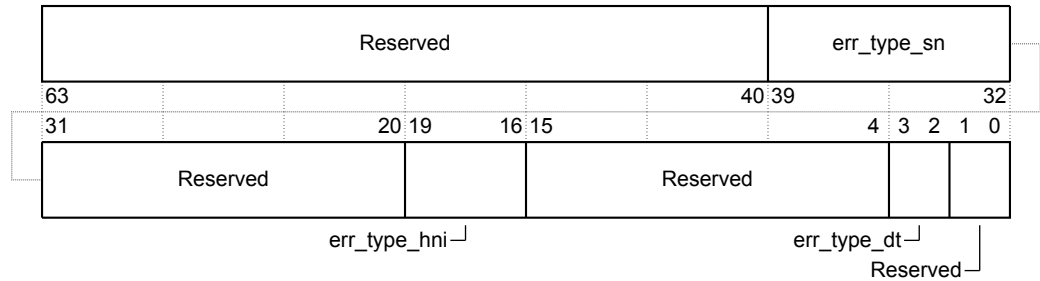
Figure 3-19 err_type_31_0 register bit assignments

The following table shows the err_type_31_0 register bit assignments.

Table 3-29 err_type_31_0 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:32]	err_type_sn	RO	0x0	Indicates the type of SN error: 0b01 = Correctable error. 0b11 = Fatal error. Within this field, the slave nodes are: • SN0 = bits[33:32]. • SN1 = bits[35:34]. • SN2 = bits[37:36], for 8XP/4HNF only. • SN3 = bits[39:38], for 8XP/4HNF only.
[31:20]	-	RAZ/WI	0x0	Reserved
[19:16]	err_type_hni	RO	0x0	Indicates the type of HN-I error: 0b01 = Correctable error. 0b11 = Fatal error. Within this field, the HN-I nodes are: • HN-I0 = bits[17:16]. • Reserved = bits[19:18].
[15:4]	-	RAZ/WI	0x0	Reserved
[3:2]	err_type_dt	RO	0x0	Indicates the type of DT error: 0b01 = Correctable error. 0b11 = Fatal error.
[1:0]	-	RO	0x0	Reserved

Error Type Value [63:32] register

The err_type_63_32 register is at offset 0x0328. Its characteristics are:

Purpose   Indicates the type of error in nodes [63:32].

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

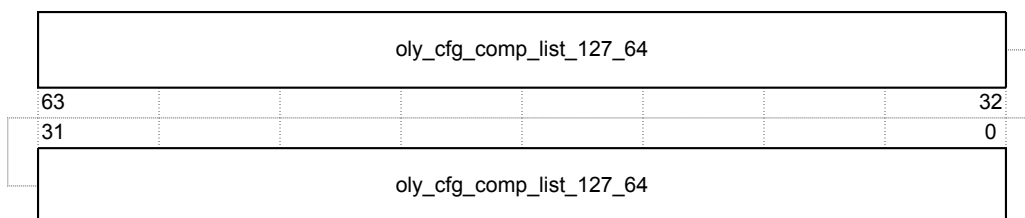The following figure shows the err_type_63_32 register bit assignments.

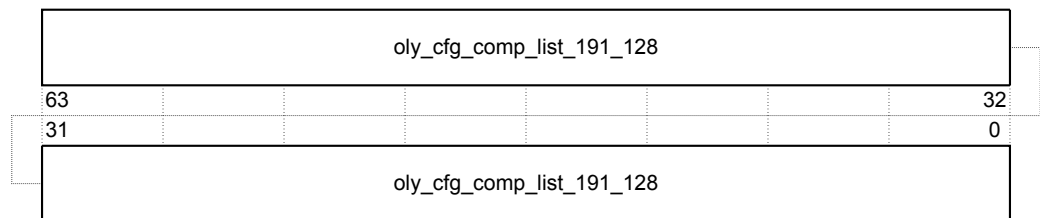

Figure 3-20 err_type_63_32 register bit assignments

The following table shows the err_type_63_32 register bit assignments.

Table 3-30 err_type_63_32 register bit assignments

Bits	Name	Access	Reset value	Description
[63:16]	-	RAZ/WI	0x0	Reserved
[15:0]	err_type_hnf	RO	0x0	Indicates the type of HN-F error: `0b01` = Correctable error. `0b11` = Fatal error. Within this field, the HN-F nodes are: • HN-F0 = bits[1:0]. • HN-F1 = bits[3:2]. • HN-F2 = bits[5:4], for 8XP/4HNF only. • HN-F3 = bits[7:6], for 8XP/4HNF only. Bits[15:8] are Reserved.

Error Type Value [95:64] register

The err_type_95_64 register is at offset `0x0330`. Its characteristics are:

Purpose	Indicates the type of error in nodes [95:64].
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the err_type_95_64 register bit assignments.
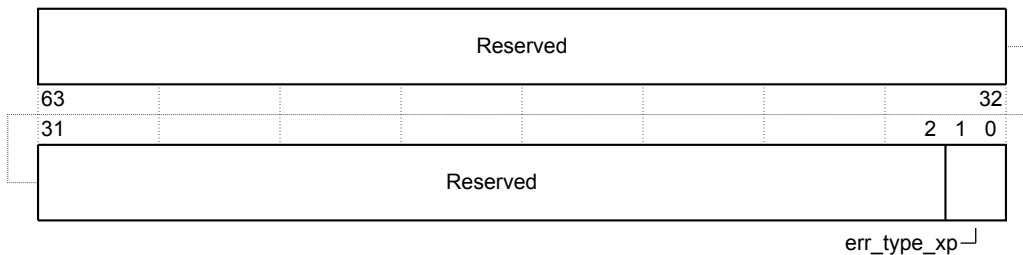
Figure 3-21 err_type_95_64 register bit assignments

The following table shows the err_type_95_64 register bit assignments.

Table 3-31 err_type_95_64 register bit assignments

Bits	Name	Access	Reset value	Description
[63:2]	-	RAZ/WI	0x0	Reserved
[1:0]	err_type_xp	RO	0b00	Indicates the type of XP error: 0b01 = Correctable error. 0b11 = Fatal error.

Error Type Value [159:128] register

The err_type_159_128 register is at offset 0x0340. Its characteristics are:

Purpose	Indicates the type of error in nodes [159:128].
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the err_type_159_128 register bit assignments.
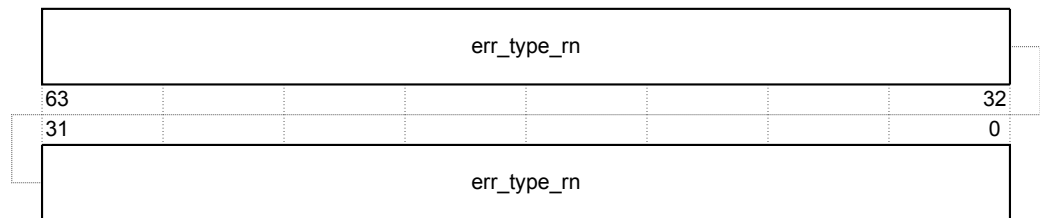
Figure 3-22 err_type_159_128 register bit assignments

The following table shows the err_type_159_128 register bit assignments.

Table 3-32 err_type_159_128 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	err_type_rn	RO	0x0	Indicates the type of RN-I interface error: 0b01 = Correctable error. 0b11 = Fatal error. Within this field, the RN-I nodes are: • RN-I0 = bits[1:0]. • RN-I1 = bits[3:2]. • RN-I2 = bits[5:4]. Bits[63:6] are Reserved.

Peripheral ID 4 and Peripheral ID 5 register

The periph_id_4_periph_id_5 register is at offset 0x0FD0. Its characteristics are:

Purpose	Contains Peripheral ID 4 in bits[31:0] and Peripheral ID 5 in bits[63:32].
Usage constraints	Only accessible by Secure accesses.
Configurations	Available in all configurations.
Attributes	See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the periph_id_4_periph_id_5 register bit assignments.
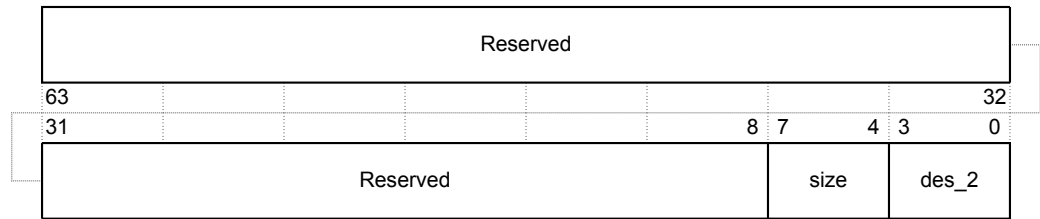
Figure 3-23 periph_id_4_periph_id_5 register bit assignments

The following table shows the periph_id_4_periph_id_5 register bit assignments.

Table 3-33 periph_id_4_periph_id_5 register bit assignments

Bits	Name	Access	Reset value	Description
[63:8]	-	RAZ/WI	0x0	Reserved
[7:4]	size	RO	0xC	Log_2 of the number of 4KB blocks occupied by the interface
[3:0]	des_2	RO	0x4	JEP106 continuation code [3:0]

Peripheral ID 6 and Peripheral ID 7 register

The periph_id_6_periph_id_7 register is at offset 0x0FD8. Its characteristics are:

Purpose   Contains Peripheral ID 6 in bits[31:0] and Peripheral ID 7 in bits[63:32].

Usage constraints Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the periph_id_6_periph_id_7 register bit assignments.
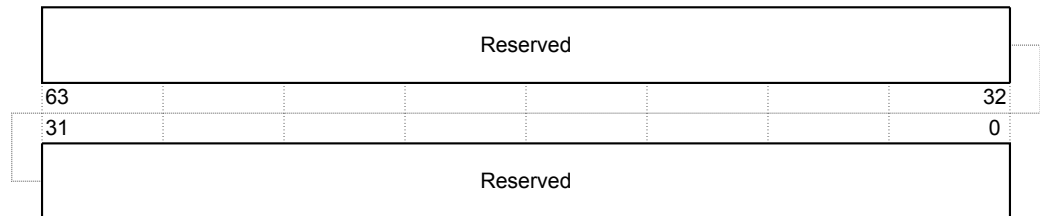
Figure 3-24 periph_id_6_periph_id_7 register bit assignments

The following table shows the periph_id_6_periph_id_7 register bit assignments.

Table 3-34 periph_id_6_periph_id_7 register bit assignments

Bits	Name	Access	Reset value	Description
[63:0]	-	RAZ/WI	0x0	Reserved

Peripheral ID 0 and Peripheral ID 1 register

The periph_id_0_periph_id_1 register is at offset 0x0FE0. Its characteristics are:

Purpose   Contains Peripheral ID 0 in bits[31:0] and Peripheral ID 1 in bits[63:32].

Usage constraints Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the periph_id_0_periph_id_1 register bit assignments.

Reserved		des_0	part_1
63	40	39 36	35 32
31	8	7	0
Reserved		part_0	
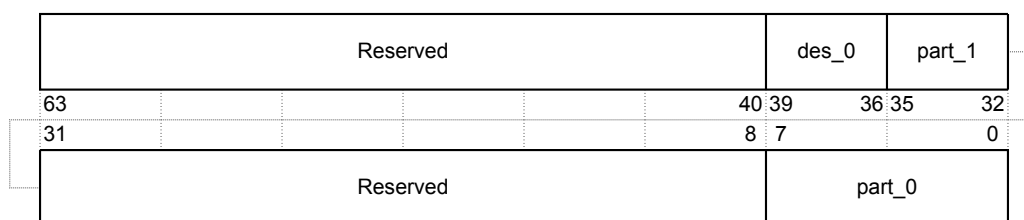
Figure 3-25 periph_id_0_periph_id_1 register bit assignments

The following table shows the periph_id_0_periph_id_1 register bit assignments.

Table 3-35 periph_id_0_periph_id_1 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:36]	des_0	RO	0xB	JEP106 identity code [3:0]
[35:32]	part_1	RO	0x4	Part number [11:8]
[31:8]	-	RAZ/WI	0x0	Reserved
[7:0]	part_0	RO	0x30	Part number [7:0]

Peripheral ID 2 and Peripheral ID 3 register

The periph_id_2_periph_id_3 register is at offset 0x0FE8. Its characteristics are:

Purpose   Contains Peripheral ID 2 in bits[31:0] and Peripheral ID 3 in bits[63:32].

Usage constraints Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the periph_id_2_periph_id_3 register bit assignments.

Reserved		cmod		
63	40	39		32
31	8	7 4	3 2	0
Reserved		revision		des_1

jedec ⌐
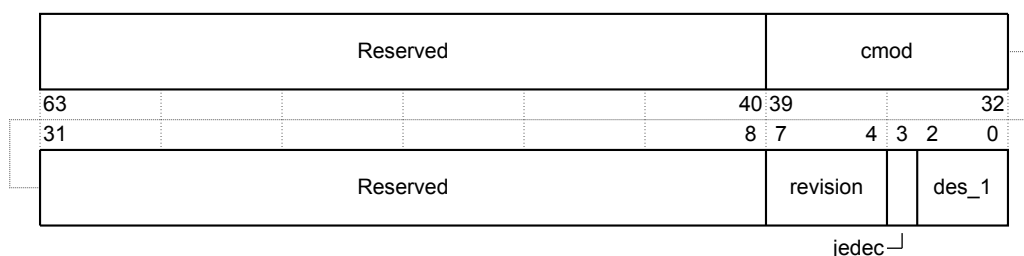
Figure 3-26 periph_id_2_periph_id_3 register bit assignments

The following table shows the periph_id_2_periph_id_3 register bit assignments.

Table 3-36 periph_id_2_periph_id_3 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:32]	cmod	RO	0x0	Customer and manufacturer revision.
[31:8]	-	RAZ/WI	0x0	Reserved
[7:4]	revision	RO	0x0	Revision: 0x0 r0p0.

Table 3-36 periph_id_2_periph_id_3 register bit assignments (continued)

Bits	Name	Access	Reset value	Description
[3]	jedec	RO	1	JEDEC JEP106 identity code is used.
[2:0]	des_1	RO	0b011	JEP106 identity code [6:4].

Component ID 0 and Component ID 1 register

The component_id_0_component_id_1 register is at offset 0x0FF0. Its characteristics are:

Purpose   Contains Component ID 0 in bits[31:0] and Component ID 1 in bits[63:32].

Usage constraints Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the component_id_0_component_id_1 register bit assignments.

Reserved	class	prmbl_1
63 40	39 36	35 32
31 8	7	0
Reserved		prmbl_0
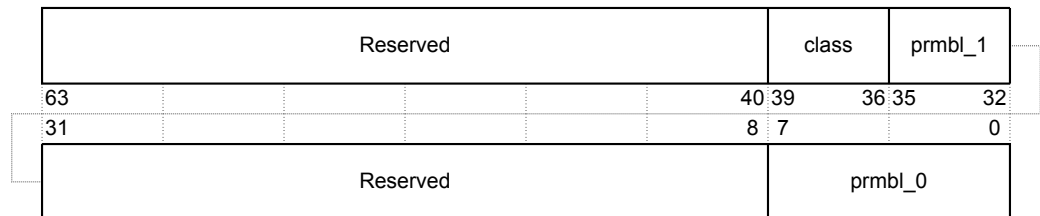
Figure 3-27 component_id_0_component_id_1 register bit assignments

The following table shows the component_id_0_component_id_1 register bit assignments.

Table 3-37 component_id_0_component_id_1 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:36]	class	RO	0xF	Component class
[35:32]	prmbl_1	RO	0x0	Component ID 1
[31:8]	-	RAZ/WI	0x0	Reserved
[7:0]	prmbl_0	RO	0x0D	Component ID 0

Component ID 2 and Component ID 3 register

The component_id_2_component_id_3 register is at offset 0x0FF8. Its characteristics are:

Purpose   Contains Component ID 2 in bits[31:0] and Component ID 3 in bits[63:32].

Usage constraints Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

The following figure shows the component_id_2_component_id_3 register bit assignments.
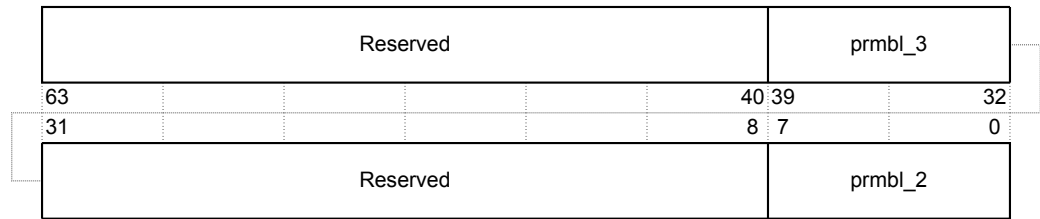
Figure 3-28 component_id_2_component_id_3 register bit assignments

The following table shows the component_id_2_component_id_3 register bit assignments.

Table 3-38 component_id_2_component_id_3 register bit assignments

Bits	Name	Access	Reset value	Description
[63:40]	-	RAZ/WI	0x0	Reserved
[39:32]	prmbl_3	RO	0xB1	Component ID 3
[31:8]	-	RAZ/WI	0x0	Reserved
[7:0]	prmbl_2	RO	0x05	Component ID 2

MN Identification register

The oly_mn_oly_id register is at offset 0xFF00. Its characteristics are:

Purpose   Contains the component identification information.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-4 MN register summary* on page 3-87.

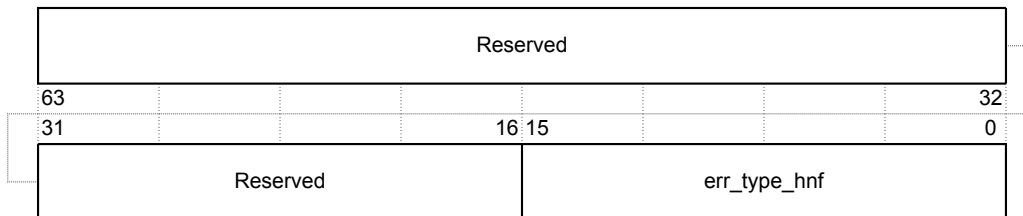The following figure shows the oly_mn_oly_id register bit assignments.
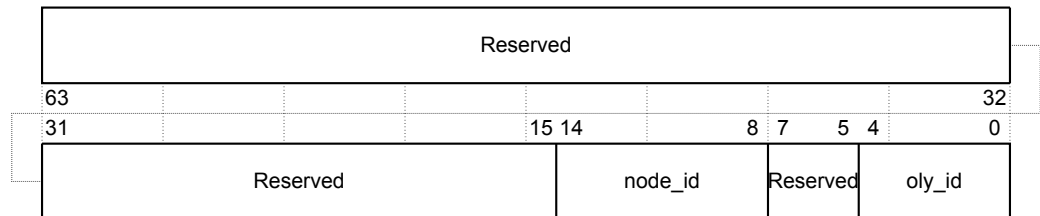
Figure 3-29 oly_mn_oly_id register bit assignments

The following table shows the oly_mn_oly_id register bit assignments.

Table 3-39 oly_mn_oly_id register bit assignments

Bits	Name	Access	Reset value	Description
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	0x0	The node ID of the MN
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x1	Node-type identifier

Related references

3.1.2 Node type IDs on page 3-85.

3.3.2 XP register descriptions

This section lists the XP registers.

XP Routing Control register

The xp_routing_control register is at offset `0x0000`. Its characteristics are:

Purpose	Controls the XP routing.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the xp_routing_control register bit assignments.
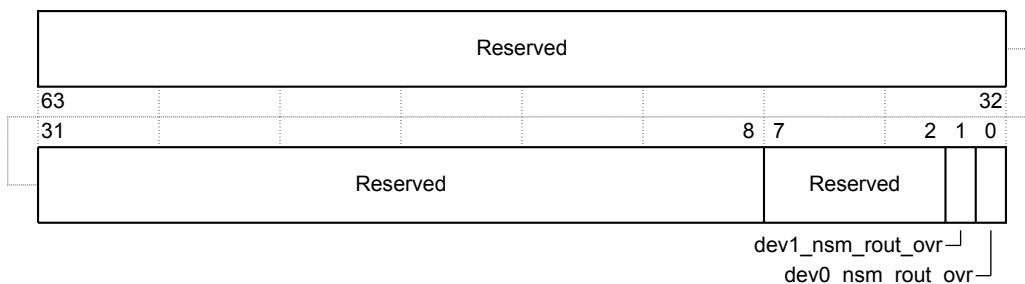
Figure 3-30 xp_routing_control register bit assignments

The following table shows the xp_routing_control register bit assignments.

Table 3-40 xp_routing_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:8]	-	RAZ/WI	0x0	Reserved
[7:2]	-	RW	0x0	Reserved
[1]	dev1_nsm_rout_ovr	RW	0	Device 1 port non-broadcast routing vector override enable
[0]	dev0_nsm_rout_ovr	RW	0	Device 0 port non-broadcast routing vector override enable

XP Device 0 Port NSM Routing register

The dev0_nsm_routing_vector register is at offset `0x0008`. Its characteristics are:

Purpose   Specifies the NSM routing information for an XP device 0 port.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev0_nsm_routing_vector register bit assignments.
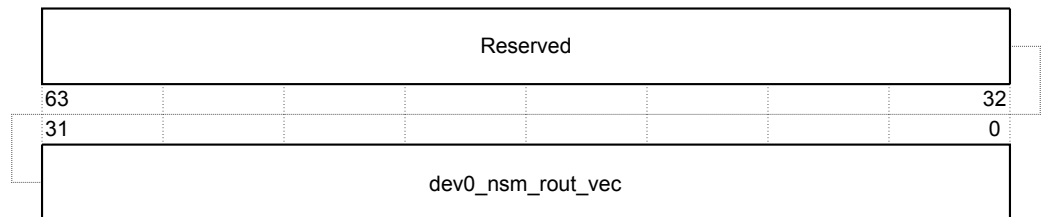
Figure 3-31 dev0_nsm_routing_vector register bit assignments

The following table shows the dev0_nsm_routing_vector register bit assignments.

Table 3-41 dev0_nsm_routing_vector register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	dev0_nsm_rout_vec	RW	0x0	Device 0 non-broadcast routing vector

XP Device 1 Port NSM Routing register

The dev1_nsm_routing_vector register is at offset `0x0010`. Its characteristics are:

Purpose   Specifies the NSM routing information for an XP device 1 port.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev1_nsm_routing_vector register bit assignments.
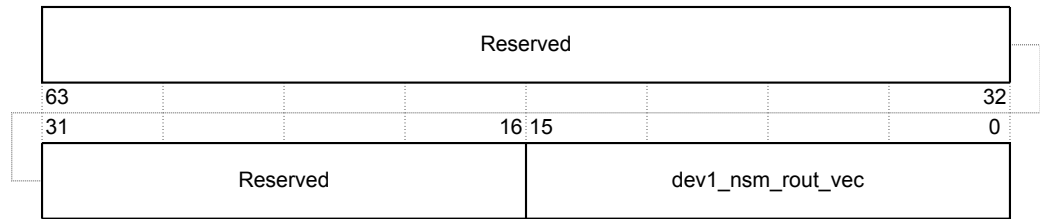
Figure 3-32 dev1_nsm_routing_vector register bit assignments

The following table shows the dev1_nsm_routing_vector register bit assignments.

Table 3-42 dev1_nsm_routing_vector register bit assignments

Bits	Name	Access	Reset value	Function
[63:16]	-	RAZ/WI	0x0	Reserved
[15:0]	dev1_nsm_rout_vec	RW	0x0	Device 1 non-broadcast routing vector

Device 0 Port QoS Control register

The dev0_qos_control register is at offset `0x0110`. Its characteristics are:

Purpose Controls the QoS settings for the device 0 port.

Usage constraints Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations Available in all configurations.

Attributes See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev0_qos_control register bit assignments.

Figure 3-33 dev0_qos_control register bit assignments

The following table shows the dev0_qos_control register bit assignments.

Table 3-43 dev0_qos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:20]	-	RAZ/WI	0x0	Reserved
[19:16]	dev0_qos_override	RW	0x0	Port 0 QoS override value.
[15:7]	-	RAZ/WI	0x0	Reserved

Table 3-43 dev0_qos_control register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[6]	dev0_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[5]	-	RAZ/WI	0	Reserved
[4]	dev0_reg_mode	RW	0	Configures the mode of the QoS regulator: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[3]	-	RAZ/WI	0	Reserved
[2]	dev0_qos_override_en	RW	0	Port 0 QoS override enable. When set, this bit enables the QoS value on inbound transactions to be overridden. When this device port is connected to a protocol bridge, this bit must be set to 0.
[1]	-	RAZ/WI	0	Reserved
[0]	dev0_lat_en	RW	0	Port 0 QoS regulation enable. When set, this bit enables regulation.

Device 0 Port QoS Latency Target register

The dev0_qos_lat_tgt register is at offset `0x0118`. Its characteristics are:

Purpose	Controls the QoS target latency, in cycles, for the regulation of the device 0 port. A value of 0 corresponds to no regulation.
Usage constraints	Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev0_qos_lat_tgt register bit assignments.
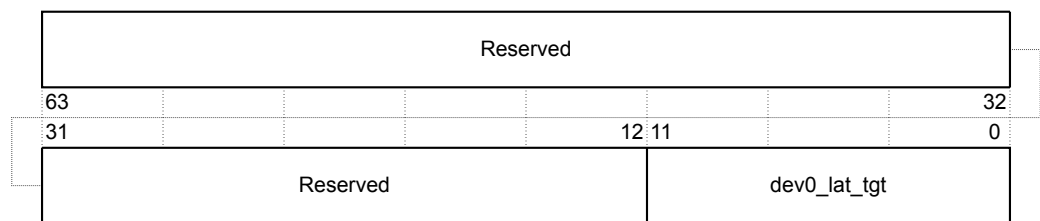
Figure 3-34 dev0_qos_lat_tgt register bit assignments

The following table shows the dev0_qos_lat_tgt register bit assignments.

Table 3-44 dev0_qos_lat_tgt register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11:0]	dev0_lat_tgt	RW	0x0	Port 0 target latency

Device 0 Port QoS Latency Scale register

The dev0_qos_lat_scale register is at offset `0x0120`. Its characteristics are:

Purpose   Controls the QoS target latency scale factor for the device 0 port. It is coded for powers of 2 in the range 2^{-5} to 2^{-12}.

Usage constraints   Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev0_qos_lat_scale register bit assignments.

Figure 3-35 dev0_qos_lat_scale register bit assignments

The following table shows the dev0_qos_lat_scale register bit assignments.

Table 3-45 dev0_qos_lat_scale register bit assignments

Bits	Name	Access	Reset value	Function
[63:3]	-	RAZ/WI	0x0	Reserved
[2:0]	dev0_lat_scale	RW	0x0	Port 0 QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Device 0 Port QoS Latency Range register

The dev0_qos_lat_range register is at offset `0x0128`. Its characteristics are:

Purpose   Controls the QoS minimum and maximum values generated by the QoS latency regulator for the device 0 port.

Usage constraints   Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev0_qos_lat_range register bit assignments.

Figure 3-36 dev0_qos_lat_range register bit assignments

The following table shows the dev0_qos_lat_range register bit assignments.

Table 3-46 dev0_qos_lat_range register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11:8]	dev0_lat_max_qos	RW	0x0	Port 0 QoS maximum value
[7:4]	-	RAZ/WI	0x0	Reserved
[3:0]	dev0_lat_min_qos	RW	0x0	Port 0 QoS minimum value

Device 1 Port QoS Control register

The dev1_qos_control register is at offset `0x0210`. Its characteristics are:

Purpose	Controls the QoS settings for the device 1 port.
Usage constraints	Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev1_qos_control register bit assignments.

Figure 3-37 dev1_qos_control register bit assignments

The following table shows the dev1_qos_control register bit assignments.

Table 3-47 dev1_qos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:20]	-	RAZ/WI	0x0	Reserved
[19:16]	dev1_qos_override	RW	0x0	Port 1 QoS override value.
[15:7]	-	RAZ/WI	0x0	Reserved
[6]	dev1_pqv_mode		0	Configures the mode of the QoS regulator during period mode for bandwidth regulation: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[5]	-	RAZ/WI	0	Reserved

Table 3-47 dev1_qos_control register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[4]	dev1_reg_mode		0	Configures the mode of the QoS regulator: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[3]	-	RAZ/WI	0	Reserved
[2]	dev1_qos_override_en	RW	0	Port 1 QoS override enable. When set, this bit enables the QoS value on inbound transactions to be overridden. When this device port is connected to a protocol bridge, this bit must be set to 0.
[1]	-	RAZ/WI	0	Reserved
[0]	dev1_lat_en	RW	0	Port 1 QoS regulation enable. When set, this bit enables regulation.

Device 1 Port QoS Target Latency register

The dev1_qos_lat_tgt register is at offset `0x0218`. Its characteristics are:

Purpose	Controls the QoS target latency, in cycles, for the regulation of the device 1 port. A value of 0 corresponds to no regulation.
Usage constraints	Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev1_qos_lat_tgt register bit assignments.

Reserved		
63		32
31	12	11 0
Reserved		dev1_lat_tgt
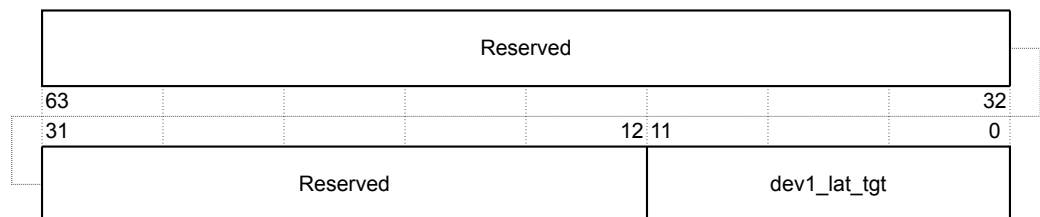
Figure 3-38 dev1_qos_lat_tgt register bit assignments

The following table shows the dev1_qos_lat_tgt register bit assignments.

Table 3-48 dev1_qos_lat_tgt register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11:0]	dev1_lat_tgt	RW	0x0	Port 1 target latency

Device 1 Port QoS Latency Scale register

The dev1_qos_lat_scale register is at offset `0x0220`. Its characteristics are:

Purpose	Controls the QoS target latency scale factor for the device 1 port. It is coded for powers of 2 in the range 2^{-5} to 2^{-12}.
Usage constraints	Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev1_qos_lat_scale register bit assignments.
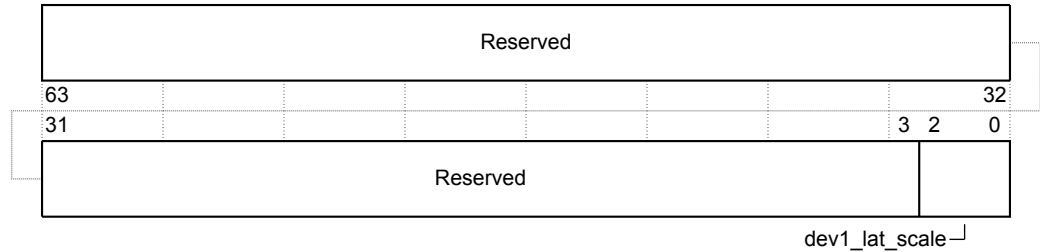
Figure 3-39 dev1_qos_lat_scale register bit assignments

The following table shows the dev1_qos_lat_scale register bit assignments.

Table 3-49 dev1_qos_lat_scale register bit assignments

Bits	Name	Access	Reset value	Function
[63:3]	-	RAZ/WI	0x0	Reserved
[2:0]	dev1_lat_scale	RW	0x0	Port 1 QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Device 1 Port QoS Latency Range register

The dev1_qos_lat_range register is at offset 0x0228. Its characteristics are:

Purpose   Controls the QoS minimum and maximum values generated by the QoS latency regulator for the device 1 port.

Usage constraints   Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dev1_qos_lat_range register bit assignments.

Figure 3-40 dev1_qos_lat_range register bit assignments

The following table shows the dev1_qos_lat_range register bit assignments.

Table 3-50 dev1_qos_lat_range register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11:8]	dev1_lat_max_qos	RW	0x0	Port 1 QoS maximum value

Table 3-50 dev1_qos_lat_range register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[7:4]	-	RAZ/WI	0x0	Reserved
[3:0]	dev1_lat_min_qos	RW	0x0	Port 1 QoS minimum value

Debug and Trace Configuration register

The dt_config register is at offset 0x0300. Its characteristics are:

Purpose   Configures the debug and trace logic.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_config register bit assignments.
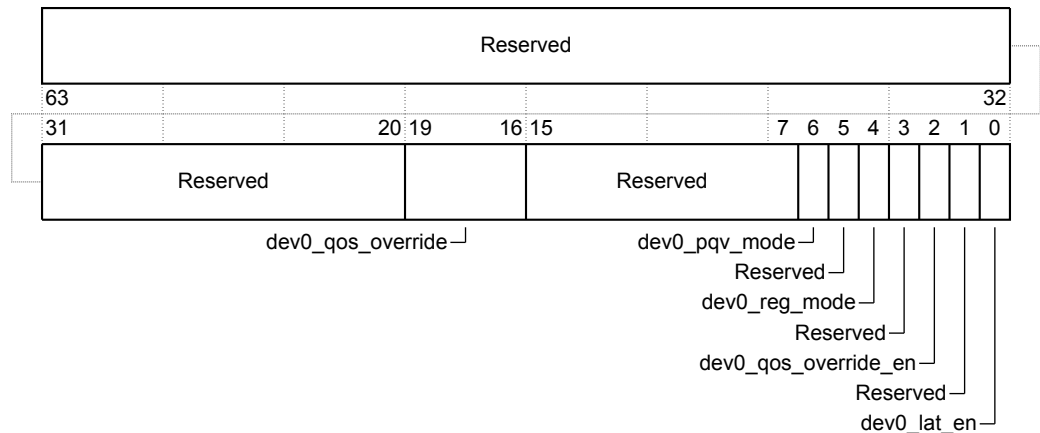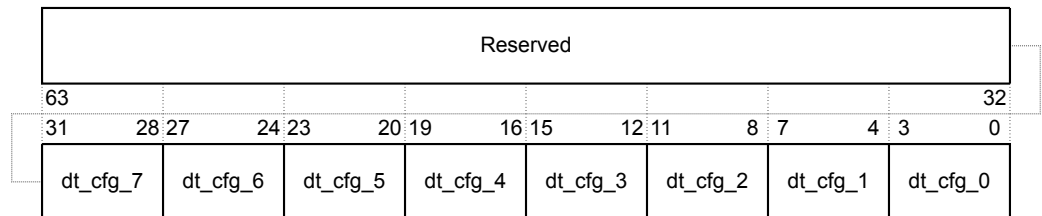

Figure 3-41 dt_config register bit assignments

The following table shows the dt_config register bit assignments.

Table 3-51 dt_config register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:28]	dt_cfg_7	RW	0x0	Select source to be transmitted on DTBus[7][a]
[27:24]	dt_cfg_6	RW	0x0	Select source to be transmitted on DTBus[6][a]
[23:20]	dt_cfg_5	RW	0x0	Select source to be transmitted on DTBus[5][a]
[19:16]	dt_cfg_4	RW	0x0	Select source to be transmitted on DTBus[4][a]
[15:12]	dt_cfg_3	RW	0x0	Select source to be transmitted on DTBus[3][a]
[11:8]	dt_cfg_2	RW	0x0	Select source to be transmitted on DTBus[2][a]
[7:4]	dt_cfg_1	RW	0x0	Select source to be transmitted on DTBus[1][a]
[3:0]	dt_cfg_0	RW	0x0	Select source to be transmitted on DTBus[0][a]

Table 3-52 dt_cfg field values

Value	Description
0x0	DT bus input from previous XP (pass-through)
0x1	OR of watchpoint 0 and 1
0x2	Watchpoint 0

---

[a] See *Table 3-52 dt_cfg field values* on page 3-121 for the dt_cfg field values.

Table 3-52 dt_cfg field values (continued)

Value	Description
`0x3`	Watchpoint 1
`0x4`	XP PMU event 0
`0x5`	XP PMU event 1
`0x6`	XP PMU event 2
`0x7`	XP PMU event 3
`0x8`	Device 0 PMU event 0
`0x9`	Device 0 PMU event 1
`0xA`	Device 0 PMU event 2
`0xB`	Device 0 PMU event 3
`0xC`	Device 1 PMU event 0
`0xD`	Device 1 PMU event 1
`0xE`	Device 1 PMU event 2
`0xF`	Device 1 PMU event 3

Debug and Trace Interface Select register

The dt_interface_sel register is at offset `0x0308`. Its characteristics are:

Purpose	Selects the interface to watch during debug.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_interface_sel register bit assignments.
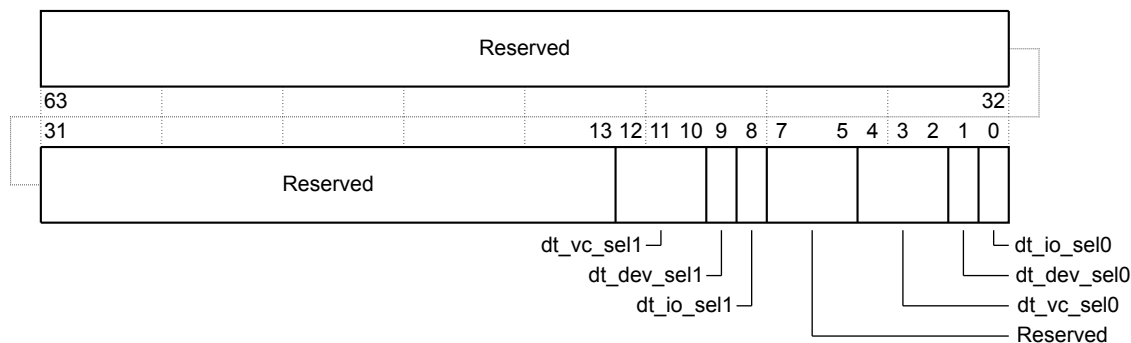
Figure 3-42 dt_interface_sel register bit assignments

The following table shows the dt_interface_sel register bit assignments.

Table 3-53 dt_interface_sel register bit assignments

Bits	Name	Access	Reset value	Function
[63:13]	-	RAZ/WI	0x0	Reserved
[12:10]	dt_vc_sel1	RW	0b000	Selection of channel type:
				0b000 Select REQ channel.
				0b001 Select RESP channel.
				0b010 Select SNP channel.
				0b011 Select DATA channel.
				0b100 Reserved.
				0b101 Reserved.
				0b110 Reserved.
				0b111 Select DATB channel.
[9]	dt_dev_sel1	RW	0	Selection of device 0 or device 1 port in specified XP:
				0 Select device port 0.
				1 Select device port 1.
[8]	dt_io_sel1	RW	0	Selection of TX or RX type for specified channel:
				0 Select RX channel.
				1 Select TX channel.
[7:5]	-	RAZ/WI	0x0	Reserved
[4:2]	dt_vc_sel0	RW	0b000	Selection of channel type:
				0b000 Select REQ channel.
				0b001 Select RESP channel.
				0b010 Select SNP channel.
				0b011 Select DATA channel.
				0b100 Reserved.
				0b101 Reserved.
				0b110 Reserved.
				0b111 Select DATB channel.
[1]	dt_dev_sel0	RW	0	Selection of device 0 or device 1 port in specified XP:
				0 Select device port 0.
				1 Select device port 1.
[0]	dt_io_sel0	RW	0	Selection of TX or RX type for specified channel:
				0 Select RX channel.
				1 Select TX channel.

Debug and Trace Comparison Low Value 0 register

The dt_cmp_val0_l register is at offset 0x0310. Its characteristics are:

Purpose   Value used for least-significant bits of watchpoint comparison.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes See *Table 3-5 XP register summary* on page 3-88.

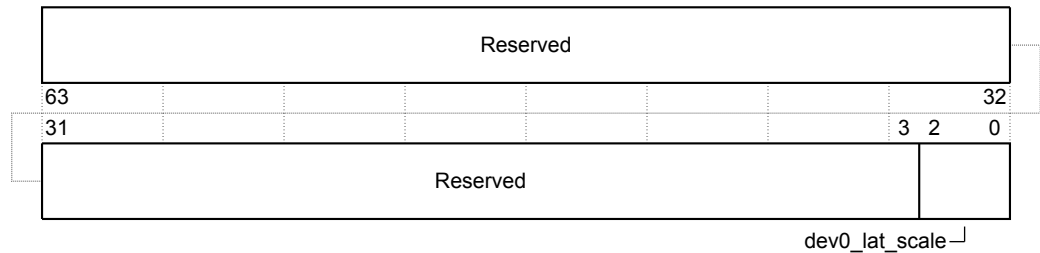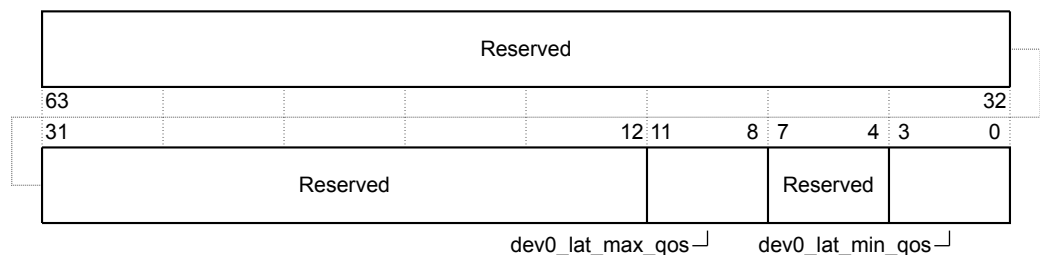The following figure shows the dt_cmp_val0_l register bit assignments.

Figure 3-43 dt_cmp_val0_l register bit assignments

The following table shows the dt_cmp_val0_l register bit assignments.

Table 3-54 dt_cmp_val0_l register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	`0x0`	Reserved
[62:0]	dt_cmp_val0_l	RW	`0x0`	Flit mapping: **val/mask[43:0]** ADDR **val/mask[45:44]** CCID **val/mask[47:46]** DATAID **val/mask[55:48]** DBID **val/mask[56:56]** DYNPCRD **val/mask[57:57]** EXCL **val/mask[58:58]** EXPCOMPACK **val/mask[59:59]** LIKELYSHARED **val/mask[62:60]** LPID

Debug and Trace Comparison High Value 0 register

The dt_cmp_val0_h register is at offset `0x0318`. Its characteristics are:

Purpose Value used for most-significant bits of watchpoint comparison.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *Table 3-5 XP register summary* on page 3-88.

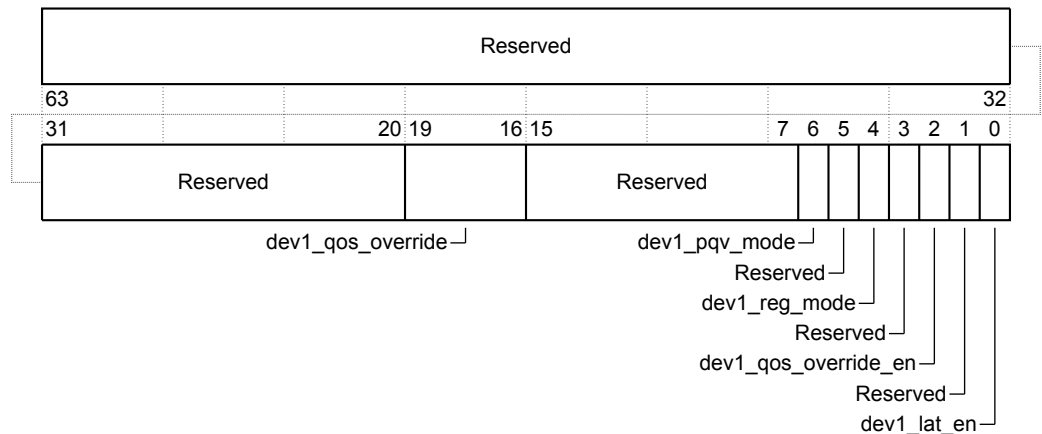The following figure shows the dt_cmp_val0_h register bit assignments.
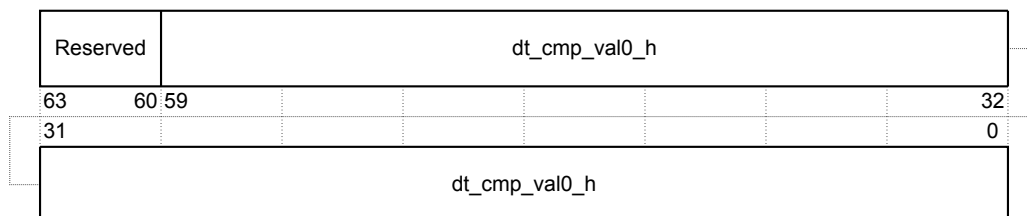
Figure 3-44 dt_cmp_val0_h register bit assignments

The following table shows the dt_cmp_val0_h register bit assignments.

Table 3-55 dt_cmp_val0_h register bit assignments

Bits	Name	Access	Reset value	Function
[63:60]	-	RAZ/WI	0x0	Reserved
[59:0]	dt_cmp_val0_h	RW	0x0	Flit mapping:
				val/mask[3:0]　　MEMATTR
				val/mask[4:4]　　MEMATTR_ALLOCATE
				val/mask[5:5]　　MEMATTR_CACHEABLE
				val/mask[6:6]　　MEMATTR_DEVICE
				val/mask[7:7]　　MEMATTR_EARLYWRACK
				val/mask[8:8]　　NS
				val/mask[13:9]　　OPCODE
				val/mask[15:14]　　ORDER
				val/mask[17:16]　　PCRDTYPE
				val/mask[21:18]　　QOS
				val/mask[24:22]　　RESP
				val/mask[26:25]　　RESPERR
				val/mask[30:27]　　RSVDC
				val/mask[33:31]　　SIZE
				val/mask[35:34]　　SNPATTR
				val/mask[36:36]　　SNPATTR_SNOOPABLE
				val/mask[37:37]　　SNPATTR_SNPDOMAIN
				val/mask[44:38]　　SRCID
				val/mask[51:45]　　TGTID
				val/mask[59:52]　　TXNID

Debug and Trace Comparison Low Mask 0 register

The dt_cmp_mask0_l register is at offset 0x0320. Its characteristics are:

Purpose　　　　　　Mask used for qualification of least-significant bits of watchpoint comparison:

0b0　The corresponding bit in the dt_cmp_val0_l register is compared to determine flit-match.

0b1　The corresponding bit in the dt_cmp_val0_l register is not compared to determine flit-match.

Usage constraints　　There are no usage constraints.

Configurations　　　Available in all configurations.

Attributes　　　　　See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_cmp_mask0_l register bit assignments.

Reserved

	dt_cmp_mask0_l

63 62   32

31   0

dt_cmp_mask0_l

Figure 3-45 dt_cmp_mask0_l register bit assignments

The following table shows the dt_cmp_mask0_l register bit assignments.

Table 3-56 dt_cmp_mask0_l register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0x0	Reserved
[62:0]	dt_cmp_mask0_l	RW	0x0	Flit mapping: **val/mask[43:0]** ADDR **val/mask[45:44]** CCID **val/mask[47:46]** DATAID **val/mask[55:48]** DBID **val/mask[56:56]** DYNPCRD **val/mask[57:57]** EXCL **val/mask[58:58]** EXPCOMPACK **val/mask[59:59]** LIKELYSHARED **val/mask[62:60]** LPID

Debug and Trace Comparison High Mask 0 register

The dt_cmp_mask0_h register is at offset `0x0328`. Its characteristics are:

Purpose   Mask used for qualification of most-significant bits of watchpoint comparison:

      `0b0` The corresponding bit in the dt_cmp_val0_h register is compared to determine flit-match.

      `0b1` The corresponding bit in the dt_cmp_val0_h register is not compared to determine flit-match.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_cmp_mask0_h register bit assignments.

Reserved	dt_cmp_mask0_h

63   60 59   32

31   0

dt_cmp_mask0_h

Figure 3-46 dt_cmp_mask0_h register bit assignments

The following table shows the dt_cmp_mask0_h register bit assignments.

Table 3-57 dt_cmp_mask0_h register bit assignments

Bits	Name	Access	Reset value	Function
[63:60]	-	RAZ/WI	0x0	Reserved
[59:0]	dt_cmp_mask0_h	RW	0x0	Flit mapping:
				val/mask[3:0] MEMATTR
				val/mask[4:4] MEMATTR_ALLOCATE
				val/mask[5:5] MEMATTR_CACHEABLE
				val/mask[6:6] MEMATTR_DEVICE
				val/mask[7:7] MEMATTR_EARLYWRACK
				val/mask[8:8] NS
				val/mask[13:9] OPCODE
				val/mask[15:14] ORDER
				val/mask[17:16] PCRDTYPE
				val/mask[21:18] QOS
				val/mask[24:22] RESP
				val/mask[26:25] RESPERR
				val/mask[30:27] RSVDC
				val/mask[33:31] SIZE
				val/mask[35:34] SNPATTR
				val/mask[36:36] SNPATTR_SNOOPABLE
				val/mask[37:37] SNPATTR_SNPDOMAIN
				val/mask[44:38] SRCID
				val/mask[51:45] TGTID
				val/mask[59:52] TXNID

Debug and Trace Comparison Low Value 1 register

The dt_cmp_val1_l register is at offset `0x0350`. Its characteristics are:

Purpose   Value used for least-significant bits of watchpoint comparison.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

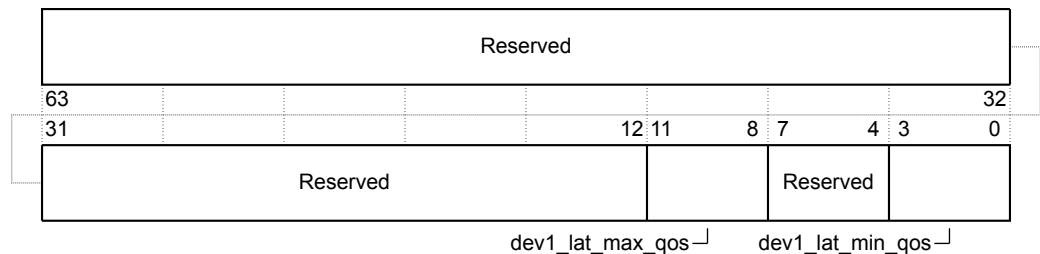The following figure shows the dt_cmp_val1_l register bit assignments.

Figure 3-47 dt_cmp_val1_l register bit assignments

The following table shows the dt_cmp_val1_l register bit assignments.

Table 3-58 dt_cmp_val1_l register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0x0	Reserved
[62:0]	dt_cmp_val1_l	RW	0x0	Flit mapping: **val/mask[43:0]** ADDR **val/mask[45:44]** CCID **val/mask[47:46]** DATAID **val/mask[55:48]** DBID **val/mask[56:56]** DYNPCRD **val/mask[57:57]** EXCL **val/mask[58:58]** EXPCOMPACK **val/mask[59:59]** LIKELYSHARED **val/mask[62:60]** LPID

Debug and Trace Comparison High Value 1 register

The dt_cmp_val1_h register is at offset 0x0358. Its characteristics are:

Purpose   Value used for most-significant bits of watchpoint comparison.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_cmp_val1_h register bit assignments.

Figure 3-48 dt_cmp_val1_h register bit assignments

The following table shows the dt_cmp_val1_h register bit assignments.

Table 3-59 dt_cmp_val1_h register bit assignments

Bits	Name	Access	Reset value	Function
[63:60]	-	RAZ/WI	`0x0`	Reserved
[59:0]	dt_cmp_val1_h	RW	`0x0`	Flit mapping:

For the [59:0] row, the Function column contains:

Flit mapping:

val/mask[3:0]	MEMATTR
val/mask[4:4]	MEMATTR_ALLOCATE
val/mask[5:5]	MEMATTR_CACHEABLE
val/mask[6:6]	MEMATTR_DEVICE
val/mask[7:7]	MEMATTR_EARLYWRACK
val/mask[8:8]	NS
val/mask[13:9]	OPCODE
val/mask[15:14]	ORDER
val/mask[17:16]	PCRDTYPE
val/mask[21:18]	QOS
val/mask[24:22]	RESP
val/mask[26:25]	RESPERR
val/mask[30:27]	RSVDC
val/mask[33:31]	SIZE
val/mask[35:34]	SNPATTR
val/mask[36:36]	SNPATTR_SNOOPABLE
val/mask[37:37]	SNPATTR_SNPDOMAIN
val/mask[44:38]	SRCID
val/mask[51:45]	TGTID
val/mask[59:52]	TXNID

Debug and Trace Comparison Low Mask 1 register

The dt_cmp_mask1_l register is at offset `0x0360`. Its characteristics are:

Purpose Mask used for qualification of least-significant bits of watchpoint comparison:

`0b0` The corresponding bit in the dt_cmp_val1_l register is compared to determine flit-match.

`0b1` The corresponding bit in the dt_cmp_val1_l register is not compared to determine flit-match.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_cmp_mask1_l register bit assignments.

Reserved

| dt_cmp_mask1_l |
| 63 62 ... 32 |
| 31 ... 0 |
| dt_cmp_mask1_l |
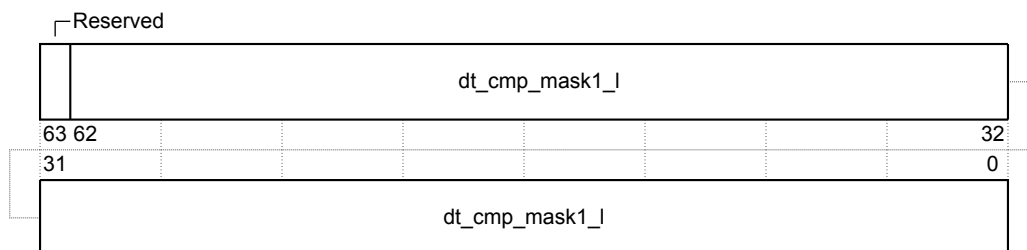
Figure 3-49 dt_cmp_mask1_l register bit assignments

The following table shows the dt_cmp_mask1_l register bit assignments.

Table 3-60 dt_cmp_mask1_l register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0x0	Reserved
[62:0]	dt_cmp_mask1_l	RW	0x0	Flit mapping: **val/mask[43:0]**　　ADDR **val/mask[45:44]**　CCID **val/mask[47:46]**　DATAID **val/mask[55:48]**　DBID **val/mask[56:56]**　DYNPCRD **val/mask[57:57]**　EXCL **val/mask[58:58]**　EXPCOMPACK **val/mask[59:59]**　LIKELYSHARED **val/mask[62:60]**　LPID

Debug and Trace Comparison High Mask 1 register

The dt_cmp_mask1_h register is at offset `0x0368`. Its characteristics are:

Purpose　　　　　Mask used for qualification of most-significant bits of watchpoint comparison:

　`0b0`　The corresponding bit in the dt_cmp_val1_h register is compared to determine flit-match.

　`0b1`　The corresponding bit in the dt_cmp_val1_h register is not compared to determine flit-match.

Usage constraints　There are no usage constraints.

Configurations　　Available in all configurations.

Attributes　　　See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the dt_cmp_mask1_h register bit assignments.
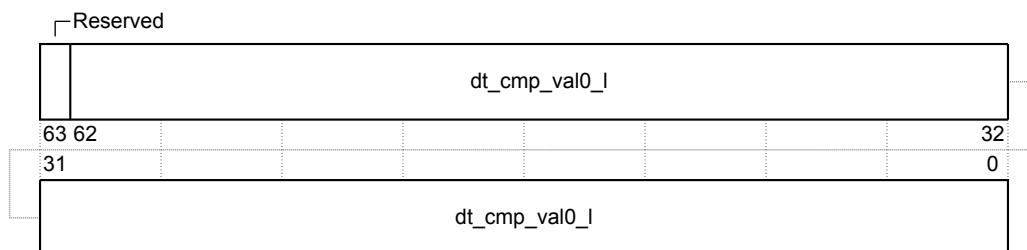
| Reserved | dt_cmp_mask1_h |
| 63　　61 60 ... 32 |
| 31 ... 0 |
| dt_cmp_mask1_h |
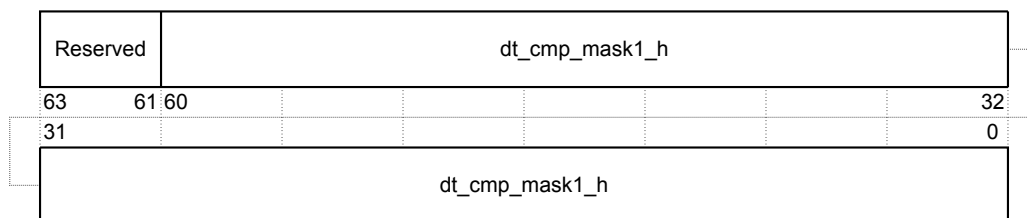
Figure 3-50 dt_cmp_mask1_h register bit assignments

The following table shows the dt_cmp_mask1_h register bit assignments.

Table 3-61 dt_cmp_mask1_h register bit assignments

Bits	Name	Access	Reset value	Function
[63:60]	-	RAZ/WI	0x0	Reserved
[59:0]	dt_cmp_mask1_h	RW	0x0	Flit mapping:
				val/mask[3:0] MEMATTR
				val/mask[4:4] MEMATTR_ALLOCATE
				val/mask[5:5] MEMATTR_CACHEABLE
				val/mask[6:6] MEMATTR_DEVICE
				val/mask[7:7] MEMATTR_EARLYWRACK
				val/mask[8:8] NS
				val/mask[13:9] OPCODE
				val/mask[15:14] ORDER
				val/mask[17:16] PCRDTYPE
				val/mask[21:18] QOS
				val/mask[24:22] RESP
				val/mask[26:25] RESPERR
				val/mask[30:27] RSVDC
				val/mask[33:31] SIZE
				val/mask[35:34] SNPATTR
				val/mask[36:36] SNPATTR_SNOOPABLE
				val/mask[37:37] SNPATTR_SNPDOMAIN
				val/mask[44:38] SRCID
				val/mask[51:45] TGTID
				val/mask[59:52] TXNID

Debug and Trace Control register, dt_control

The dt_control register is at offset 0x0370. Its characteristics are:

Purpose	Controls the debug and trace settings.
Usage constraints	Before writing bit[0], all other debug and trace configuration registers must be programmed. After debug and trace is enabled by writing bit[0], no other debug and trace configuration registers must be modified.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

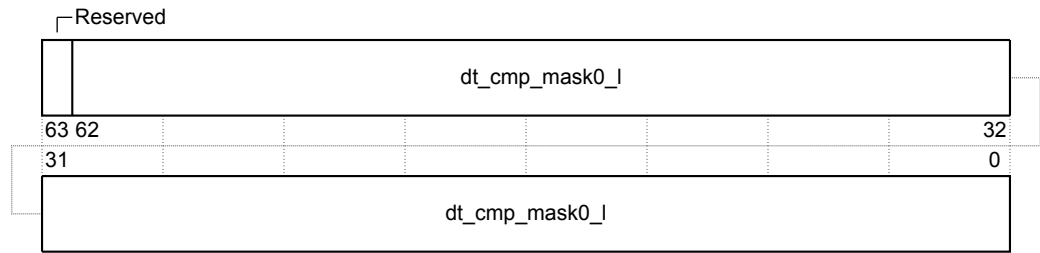The following figure shows the dt_control register bit assignments.

Figure 3-51 dt_control register bit assignments

The following table shows the dt_control register bit assignments.

Table 3-62 dt_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:24]	wp1_event_count	RW	0x0	The number of events that watchpoint 1 must observe before the trigger can be generated. The cumulative count is reset when the watchpoint is disabled by writing 0 to the dt_enable bit of this register.
[23:20]	wp0_event_count	RW	0x0	The number of events that watchpoint 0 must observe before the trigger can be generated. The cumulative count is reset when the watchpoint is disabled by writing 0 to the dt_enable bit of this register.
[19:16]	wp1_arm_sel	RW	0xF	Selects the event source that is used to arm the watchpoint 1 trigger. Any active event from the source activates the watchpoint 1 trigger logic. Arming is deactivated after reset or when the watchpoint is disabled by writing 0 to the dt_enable bit of this register: 0x0 DTBus[0]. 0x1 DTBus[1]. 0x2 DTBus[2]. 0x3 DTBus[3]. 0x4 DTBus[4]. 0x5 DTBus[5]. 0x6 DTBus[6]. 0x7 DTBus[7]. 0x8 Watchpoint 0 trigger. 0x9-0xE Reserved. 0xF Always armed.

Table 3-62 dt_control register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[15:12]	wp0_arm_sel	RW	0xF	Selects the event source that is used to arm the watchpoint 0 trigger. Any active event from the source activates the watchpoint 0 trigger logic. Arming is deactivated after reset or when the watchpoint is disabled by writing 0 to the dt_enable bit of this register: `0x0` DTBus[0]. `0x1` DTBus[1]. `0x2` DTBus[2]. `0x3` DTBus[3]. `0x4` DTBus[4]. `0x5` DTBus[5]. `0x6` DTBus[6]. `0x7` DTBus[7]. `0x8` Watchpoint 1 trigger. `0x9-0xE` Reserved. `0xF` Always armed.
[11]	txnid_copyover	RW	0	Controls whether the TXNID field from the watchpoint 0 input flit must be copied over to watchpoint 1. The copy happens the first time when watchpoint 0 is triggered: **1** Enabled. **0** Disabled.
[10:3]	dt_bus_or_mode	RW	0x0	Controls whether the bit on the DT bus must OR the input from the previous XP, instead of muxing in the current result: `0b0` OR mode disabled. `0b1` OR mode enabled.
[2:1]	dt_ss_capture_en	RW	0x0	Control snapshotting of flit on first watchpoint match. See the following table for field values. Any field not defined for the flit is written as 0.
[0]	dt_enable	RW	0	Enable debug watchpoint and PMU capability: **0** Disabled. **1** Enabled. See Usage constraints in register characteristics description.

Table 3-63 Snapshot capture enable values

Value	DWM 1	DWM 0
`0b00`	Disabled	Disabled
`0b01`	Disabled	Enabled
`0b10`	Enabled	Disabled
`0b11`	Enabled	Enabled

Debug and Trace Status register

The dt_status register is at offset `0x0378`. Its characteristics are:

Purpose	Indicates the debug and trace status.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

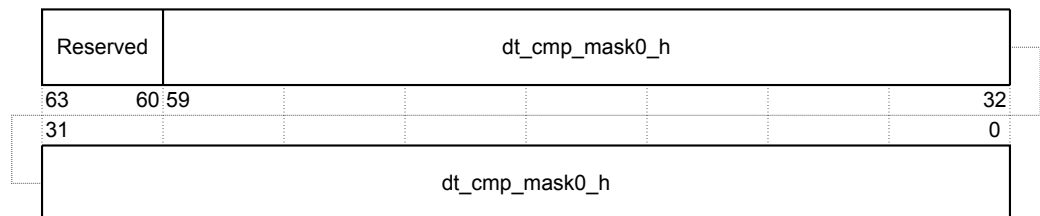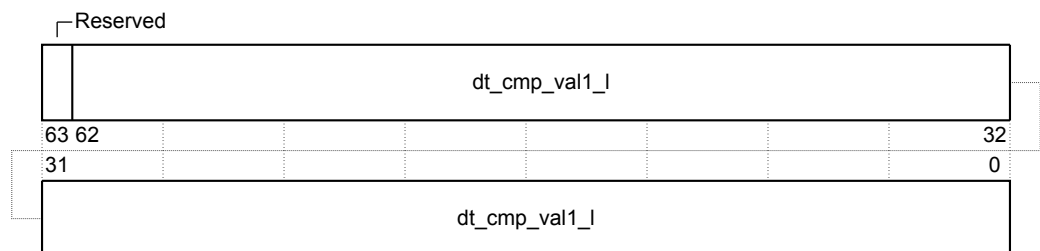The following figure shows the dt_status register bit assignments.

```
       Reserved
  63                                          32
  31                                      2  1  0
       Reserved
```
sscapture_status ┘
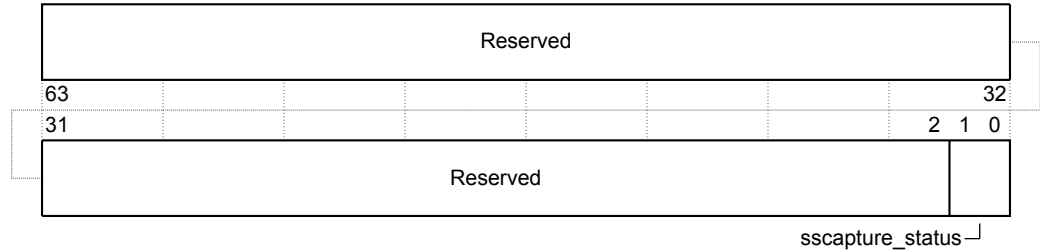
Figure 3-52 dt_status register bit assignments

The following table shows the dt_status register bit assignments.

Table 3-64 dt_status register bit assignments

Bits	Name	Access	Reset value	Function
[63:2]	-	RAZ/WI	`0x0`	Reserved
[1:0]	sscapture_status	RO	`0b00`	Indication that a flit has been snapshotted because of watchpoint match. See the following table for field values.

Table 3-65 Snapshot capture status values

Value	DWM 1	DWM 0
`0b00`	Not captured	Not captured
`0b01`	Not captured	Captured
`0b10`	Captured	Not captured
`0b11`	Captured	Captured

Debug and Trace Status Clear register

The dt_status_clr register is at offset `0x0380`. Its characteristics are:

Purpose	Clears the debug and trace status.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

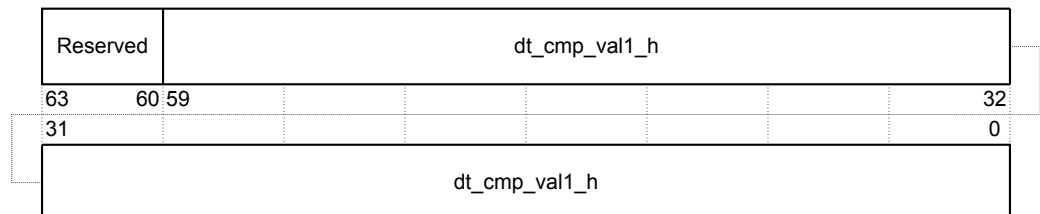The following figure shows the dt_status_clr register bit assignments.

Figure 3-53 dt_status_clr register bit assignments

The following table shows the dt_status_clr register bit assignments.

Table 3-66 dt_status_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63:2]	-	RAZ/WI	0x0	Reserved
[1:0]	dt_status_clr	WO	0b00	Write 1 to clear the DT status bit. See the following table for field values.

Table 3-67 Snapshot capture status values

Value	DWM 1	DWM 0
0b00	Not cleared	Not cleared
0b01	Not cleared	Cleared
0b10	Cleared	Not cleared
0b11	Cleared	Cleared

Error Syndrome 0 register, XP

The err_syndrome_reg0 register is at offset 0x0400. Its characteristics are:

Purpose　　　　　Indicates the XP parity error log information.

Usage constraints　　There are no usage constraints.

Configurations　　Available in all configurations.

Attributes　　　　See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the err_syndrome_reg0 register bit assignments.

Figure 3-54 err_syndrome_reg0 register bit assignments

The following table shows the err_syndrome_reg0 register bit assignments.

Table 3-68 err_syndrome_reg0 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	err_extnd	RO	0	Error extended.
[62]	first_err_vld	RO	0	First error valid.
[61:60]	err_class	RO	`0b00`	Error classification.
[59]	mult_err	RO	0	Multiple errors.
[58:43]	corrected_err_count	RO	`0x0`	Corrected error count.
[42:22]	-	RAZ/WI	`0x0`	Reserved
[21:8]	-	RAZ/WI	`0x0`	Reserved
[7:6]	-	RAZ/WI	`0b00`	Reserved
[5:0]	err_id	RO	`0x0`	Error identifier:

Error identifier (bits [5:0]):

Bit[0] Download device port number.

Bits[2:1] Download source:

00	Bus 0.
01	Bus 1.
10	Bypass.

Bits[5:3] Channel type:

000	REQ.
001	RSP.
010	SNP.
011	DATA.
111	DATB.

Related concepts

2.16.1 Parity error reporting, poisoning, and logging on page 2-80.

Error logging on page 2-46.

XP Error Syndrome Clear register

The err_syndrome_clr register is at offset `0x0480`. Its characteristics are:

Purpose Clears the error log in the Error Syndrome 0 register.

Usage constraints There are no usage constraints.

Configurations Available in all configurations.

Attributes See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the err_syndrome_clr register bit assignments.

Figure 3-55 err_syndrome_clr register bit assignments

The following table shows the err_syndrome_clr register bit assignments.

Table 3-69 err_syndrome_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0	Reserved
[62]	first_err_vld_clr	WO	0	Clears the first_err_vld bit in the Error Syndrome 0 register
[61:60]	-	RAZ/WI	0b00	Reserved
[59]	mult_err_clr	WO	0	Clears the mult_err bit in the Error Syndrome 0 register
[58:0]	-	RAZ/WI	0x0	Reserved

Related references

Error log clearing on page 2-47.
Error Syndrome 0 register, XP on page 3-135.

Auxiliary Control register, XP

The aux_ctl register is at offset `0x0500`. Its characteristics are:

Purpose	Controls various modes of operation.
Usage constraints	This register can be modified only with prior written permission from ARM.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

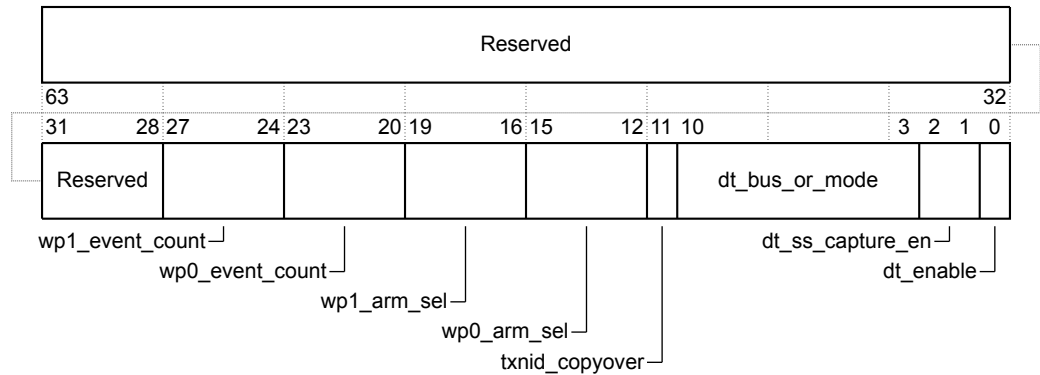The following figure shows the aux_ctl register bit assignments.

Figure 3-56 aux_ctl register bit assignments

The following table shows the aux_ctl register bit assignments.

Table 3-70 aux_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:24]	byp_prio_weight	RW	0x10	The number of cycles that a stalled bypass request waits until being prioritized over ring downloads. ──────── **Note** ──────── The priority weight value must be set greater than the respin latency, that is, the number of clocks it takes to traverse the ring once. This is to avoid starvation of a bus message when it contends with a port bypass message. ────────────────────
[23:16]	dnload_starv_thresh	RW	0x04	The number of cycles a flit, that is unable to download, waits until reserving a download flit-buffer in the target XP.
[15:8]	upload_starv_thresh	RW	0x20	The number of cycles a flit, that is unable to upload, waits until reserving a ring-slot.
[7:5]	Reserved	RAZ/WI	0b000	-
[4]	dat_parity_resperr_disable	RW	0	DAT parity signaling RespErr disable.
[3]	parity_irq_disable	RW	0	Disable parity interrupt. This bit is applicable only in configurations that include ring parity.
[2]	qpc_en	RW	0	Enable QoS priority class based upload arbitration.
[1]	dnload_starv_en	RW	1	Enable download starvation prevention mechanism.
[0]	upload_starv_en	RW	1	Enable upload starvation prevention mechanism.

Byte Parity Error Injection register, XP

The byte_par_err_inj register is at offset 0x0508. Its characteristics are:

Purpose   Selects a byte lane, within the 128-bit data bus, and injects a byte parity error on the next DAT flit.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the byte_par_err_inj register bit assignments.
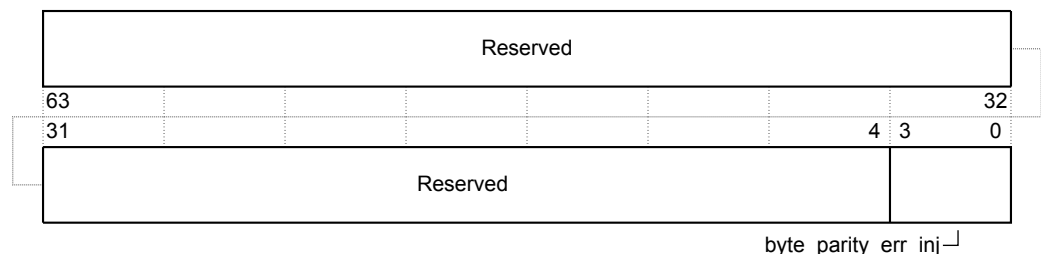
Figure 3-57 byte_par_err_inj register bit assignments

The following table shows the byte_par_err_inj register bit assignments.

Table 3-71 byte_par_err_inj register bit assignments

Bits	Name	Access	Reset value	Function
[63:4]	-	RAZ/WI	`0x0`	Reserved.
[3:0]	byte_parity_err_inj	WO	-	Selects the byte lane, within the 128-bit data bus, where the XP injects a byte parity error. The XP injects a byte parity error on the next DAT flit that it passes.
				The bit values are:
				`0b0000` Inserts a parity error in bits[7:0].
				`0b0001` Inserts a parity error in bits[15:8].
				`0b0010` Inserts a parity error in bits[23:16].
				... …
				`0b1111` Inserts a parity error in bits[127:120].
				If multiple writes occur to this field before the XP passes a DAT flit, then the XP uses the initial value that is written and ignores the subsequent writes.

Related concepts

2.16.2 Byte parity error injection on page 2-80.

PMU Event Select register, XP

The pmu_event_sel register is at offset `0x0600`. Its characteristics are:

Purpose	Selects the *Performance Monitoring Unit* (PMU) events to be counted.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the pmu_event_sel register bit assignments.

Reserved			
63			32

31 28	27 21	20 14	13 7	6 0
Reserved	pmu_event3_id	pmu_event2_id	pmu_event1_id	pmu_event0_id
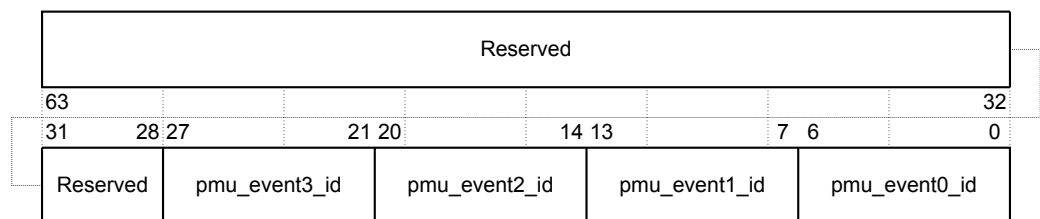
Figure 3-58 pmu_event_sel register bit assignments

The following table shows the pmu_event_sel register bit assignments.

Table 3-72 pmu_event_sel register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:21]	pmu_event3_id	RW	0x0	PMU Event 3 ID. The event is specified as a 7-bit ID with the following encodings:

Event_ID[27:25] Channel type:

0b000	REQ.
0b001	RSP.
0b010	SNP.
0b011	DAT or DATA.
0b111	DATB.

Event_ID[24] Bus number:

0 Bus 0.

1 Bus 1. This is not applicable for the SNP channel.

Event_ID[23:21] Event specifier:

0b000	Null (no event).
0b001	Set H-bit, signaled when this XP sets the H-bit.
0b010	Set S-bit, signaled when this XP sets the S-bit.
0b011	Set P-Cnt, signaled when this XP sets the P-Cnt. This is not applicable for the SNP channel.
0b100	No TknV, signaled when this XP transmits a valid packet.
0b101-0b111	Reserved.

Bits	Name	Access	Reset value	Function
[20:14]	pmu_event2_id	RW	0x0	PMU Event 2 ID. The event is specified as a 7-bit ID with the following encodings: **Event_ID[20:18]** Channel type. **Event_ID[17]** Bus number. **Event_ID[16:14]** Event specifier. See pmu_event3_id in this table for more information.
[13:7]	pmu_event1_id	RW	0x0	PMU Event 1 ID. The event is specified as a 7-bit ID with the following encodings: **Event_ID[13:11]** Channel type. **Event_ID[10]** Bus number. **Event_ID[9:7]** Event specifier. See pmu_event3_id in this table for more information.
[6:0]	pmu_event0_id	RW	0x0	PMU Event 0 ID. The event is specified as a 7-bit ID with the following encodings: **Event_ID[6:4]** Channel type. **Event_ID[3]** Bus number. **Event_ID[2:0]** Event specifier. See pmu_event3_id in this table for more information.

XP Identification register

The oly_xp_oly_id register is at offset `0xFF00`. Its characteristics are:

Purpose   Contains the component identification information.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-5 XP register summary* on page 3-88.

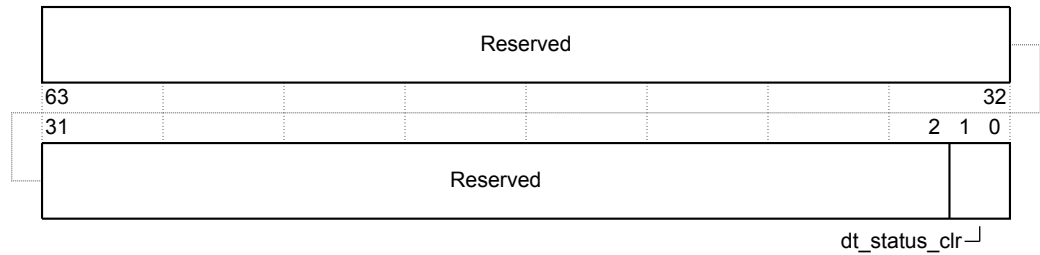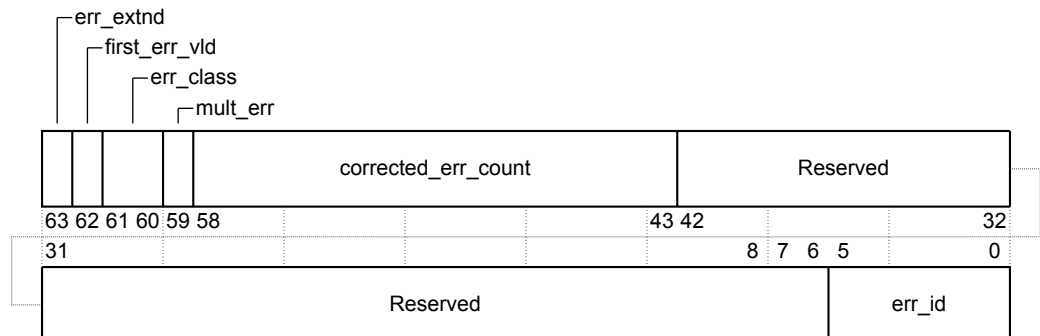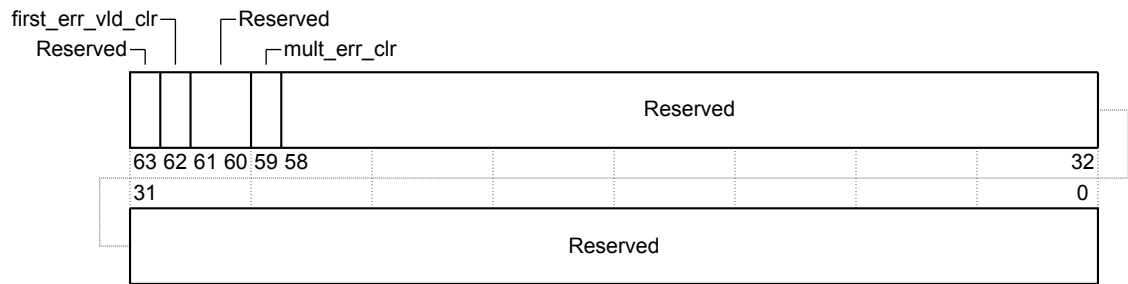The following figure shows the oly_xp_oly_id register bit assignments.

Figure 3-59 oly_xp_oly_id register bit assignments

The following table shows the oly_xp_oly_id register bit assignments.

Table 3-73 oly_xp_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	Reset value is specific to each XP	The node ID of the XP
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x8	Indicates that this node is an XP

Related references

3.1.1 Node configuration register address mapping on page 3-82.
3.1.2 Node type IDs on page 3-85.

3.3.3 HN-F register descriptions

Lists the HN-F registers.

HN-F Configuration Control register

The hnf_cfg_ctrl register is at offset `0x0000`. Its characteristics are:

Purpose	Controls the HN-F configuration.
Usage constraints	Only accessible by Secure accesses. Writes to this register must be complete before the first non-configuration access targeting the HN-F.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the hnf_cfg_ctrl register bit assignments.
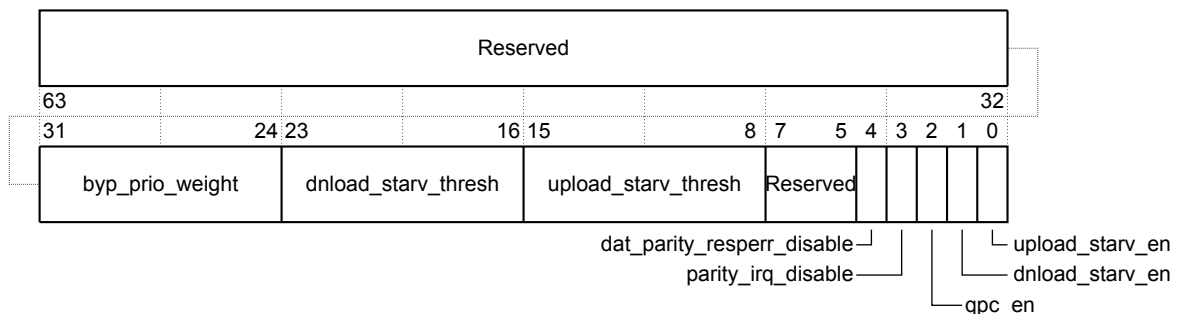

Figure 3-60 hnf_cfg_ctrl register bit assignments

The following table shows the hnf_cfg_ctrl register bit assignments.

Table 3-74 hnf_cfg_ctrl register bit assignments

Bits	Name	Access	Reset value	Function
[63:21]	-	RAZ/WI	0x0	Reserved
[20]	ncdevcmo_mc_comp	RW	0	Disable HN-F completion. The HN-F sends completion for the following transactions after receiving completion from SN-F: • Non-cacheable WriteNoSnp. • Device WriteNoSnp. • *Cache Maintenance Operation*s (CMOs).
[19]	-	RAZ/WI	0	Reserved
[18]	sf_ecc_scrub_disable	RW	0	Disable SF tag single-bit ECC error scrubbing.
[17]	l3_dat_ecc_scrub_disable	RW	0	Disable L3 data single-bit ECC error scrubbing.
[16]	l3_tag_ecc_scrub_disable	RW	0	Disable L3 tag single-bit ECC error scrubbing.
[15]	-	RAZ/WI	0b0	Reserved
[14]	pois_dis	RW	0	Disable parity error data poisoning.
[13]	par_err_dis	RW	0	Disable parity error interrupt signaling.
[12:9]	Reserved	RAZ/WI	0x0	-
[8]	cg_disable	RW	0	Disable HN-F architectural clock gates.
[7:5]	-	RAZ/WI	0x0	Reserved
[4]	ecc_disable	RW	0	Disable L3 and SF ECC generation and detection.
[3:0]	-	RAZ/WI	0x0	Reserved

HN-F SAM Control register

The hnf_sam_control register is at offset 0x0008. Its characteristics are:

Purpose   Controls the HN-F *System Address Map* (SAM).

Usage constraints   Only accessible by Secure accesses. Writes to this register must be complete before any non-configuration access targets the HN-F.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the hnf_sam_control register bit assignments.

Figure 3-61 hnf_sam_control register bit assignments

The following table shows the hnf_sam_control register bit assignments.

Table 3-75 hnf_sam_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:62]	-	RAZ/WI	0x0	Reserved
[61:56]	hn_cfg_sam_top_address_bit1	RW	0x0	Number for the bit position of the top[1] physical address bit of DRAM, which is used by the 3 SN routing mechanism. Permitted values are 28-43 inclusive.
[55:54]	-	RAZ/WI	0b00	Reserved
[53:48]	hn_cfg_sam_top_address_bit0	RW	0x0	Number for the bit position of the top[0] physical address bit of DRAM, which is used by the 3 SN routing mechanism. Permitted values are 28-43 inclusive.
[47:33]	-	RAZ/WI	0x0	Reserved
[32]	hn_cfg_three_sn_en	RW	0b0	Enable for 3 SN mode. Set to 1 to enable routing to three SNs.
[31:23]	-	RAZ/WI	0x0	Reserved
[22:16]	hn_cfg_sn2_nodeid	RW	Value depends on HN-F	Node ID for slave node 2. This field is only valid when hn_cfg_three_sn_en=1.
[15]	-	RAZ/WI	0	Reserved
[14:8]	hn_cfg_sn1_nodeid	RW	0x0	Node ID for slave node 1. This field is only valid when hn_cfg_three_sn_en=1.
[7]	-	RAZ/WI	0	Reserved
[6:0]	hn_cfg_sn0_nodeid	RW	Value depends on HN-F	Node ID for slave node 0.

Related concepts

2.12.4 HN-F SAM on page 2-56.

3 SN-F memory striping on page 2-57.

HN-F P-state Request register

The hn_cfg_pstate_req register is at offset 0x0010. Its characteristics are:

Purpose   Controls the HN-F P-state requests.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the hn_cfg_pstate_req register bit assignments.

Figure 3-62 hn_cfg_pstate_req register bit assignments

The following table shows the hn_cfg_pstate_req register bit assignments.

Table 3-76 hn_cfg_pstate_req register bit assignments

Bits	Name	Access	Reset value	Function
[63:2]	-	RAZ/WI	0x0	Reserved
[1:0]	pstate	WO	0b00	P-state request: 0b00 HNF_PM_NOL3. 0b01 HNF_PM_SFONLY. 0b10 HNF_PM_HALF. 0b11 HNF_PM_FULL.

HN-F P-state Status register

The hn_cfg_pstate_status register is at offset 0x0018. Its characteristics are:

Purpose   Indicates the HN-F P-state status.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the hn_cfg_pstate_status register bit assignments.

Figure 3-63 hn_cfg_pstate_status register bit assignments

The following table shows the hn_cfg_pstate_status register bit assignments.

Table 3-77 hn_cfg_pstate_status register bit assignments

Bits	Name	Access	Reset value	Function
[63:6]	-	RAZ/WI	`0x0`	Reserved
[5:4]	retention	RO	`0b00`	P-state retention status: `0b00` HNF_PM_RET_IDLE. `0b01` HNF_PM_RET_IDLE_2_RET. `0b10` HNF_PM_RET_RET. `0b11` HNF_PM_RET_RET_2_IDLE.
[3:0]	pstate	RO	`0x0`	P-state status: `0b0000` HNF_PM_NOL3. `0b0001` HNF_PM_NOL3_2_SFONLY. `0b0010` HNF_PM_NOL3_2_HALF. `0b0011` HNF_PM_NOL3_2_FULL. `0b0100` HNF_PM_SFONLY. `0b0101` HNF_PM_SFONLY_2_NOL3. `0b0110` HNF_PM_SFONLY_2_HALF. `0b0111` HNF_PM_SFONLY_2_FULL. `0b1000` HNF_PM_HALF. `0b1001` HNF_PM_HALF_2_NOL3. `0b1010` HNF_PM_HALF_2_SFONLY. `0b1011` HNF_PM_HALF_2_FULL. `0b1100` HNF_PM_FULL. `0b1101` HNF_PM_FULL_2_NOL3. `0b1110` HNF_PM_FULL_2_SFONLY. `0b1111` HNF_PM_FULL_2_HALF.

QoS Band register

The qos_band register indicates the QoS classifications based on the QoS value ranges.

The qos_band register is at offset `0x0020`. Its characteristics are:

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the qos_band register bit assignments.
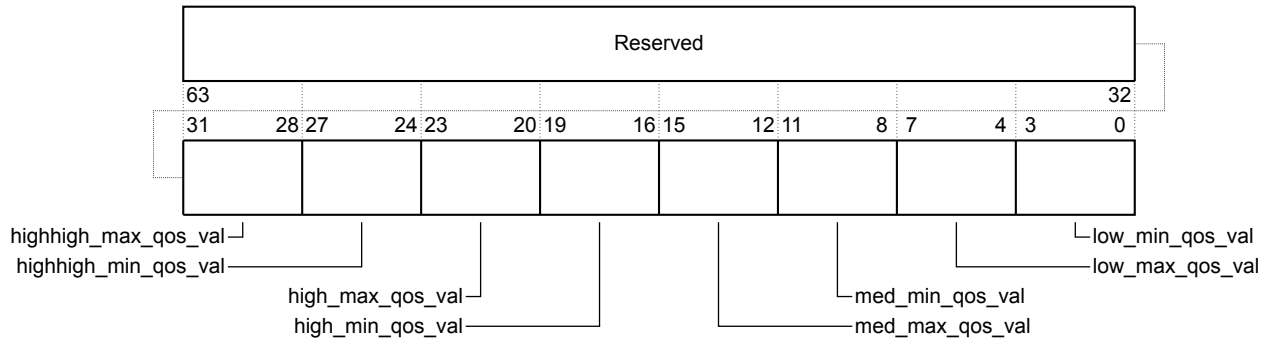
Figure 3-64 qos_band register bit assignments

The following table shows the qos_band register bit assignments.

Table 3-78 qos_band register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:28]	highhigh_max_qos_val	RO	0xF	Highest QoS class: Maximum value
[27:24]	highhigh_min_qos_val	RO	0xF	Highest QoS class: Minimum value
[23:20]	high_max_qos_val	RO	0xE	High QoS class: Maximum value
[19:16]	high_min_qos_val	RO	0xC	High QoS class: Minimum value
[15:12]	med_max_qos_val	RO	0xB	Medium QoS class: Maximum value
[11:8]	med_min_qos_val	RO	0x8	Medium QoS class: Minimum value
[7:4]	low_max_qos_val	RO	0x7	Low QoS class: Maximum value
[3:0]	low_min_qos_val	RO	0x0	Low QoS class: Minimum value

QoS Reservation register

The qos_reservation register is at offset `0x0028`. Its characteristics are:

Purpose Selects the POCQ maximum occupancy counts for each QoS class, that is, highest, high, medium, and low.

Usage constraints Writes to this register must be complete before the first non-configuration access to the HN-F.

Configurations Available in all configurations.

Attributes See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the qos_reservation register bit assignments.

Figure 3-65 qos_reservation register bit assignments

The following table shows the qos_reservation register bit assignments.

Table 3-79 qos_reservation register bit assignments

Bits	Name	Access	Reset value	Function
[63:37]	-	RAZ/WI	0x0	Reserved
[36:32]	seq_qos_max_cnt	RW	0x01	Number of entries that are reserved for snoop filter evictions in POCQ. Must be 1.
[31:29]	-	RAZ/WI	0x0	Reserved
[28:24]	highhigh_qos_max_cnt	RW	0x1F	Maximum number of highest QoS class occupancy. Allowed range is 5-31.
[23:21]	-	RAZ/WI	0x0	Reserved
[20:16]	high_qos_max_cnt	RW	0x1E	Maximum number of high QoS class occupancy. Allowed range is 4 - (highhigh_qos_max_cnt–1).
[15:13]	-	RAZ/WI	0x0	Reserved
[12:8]	med_qos_max_cnt	RW	0x0F	Maximum number of medium QoS class occupancy. Allowed range is 3 - (high_qos_max_cnt–1).
[7:5]	-	RAZ/WI	0x0	Reserved
[4:0]	low_qos_max_cnt	RW	0x05	Maximum number of low QoS class occupancy. Allowed range is 2 - (med_qos_max_cnt–1).

RN Starvation register

The rn_starvation register is at offset 0x0030. Its characteristics are:

Purpose　　　　　Selects the starvation counts for various QoS classes for static credit grantee selection.

Usage constraints　Writes to this register must be complete before the first non-configuration access to the HN-F.

Configurations　　Available in all configurations.

Attributes　　　　See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the rn_starvation register bit assignments.
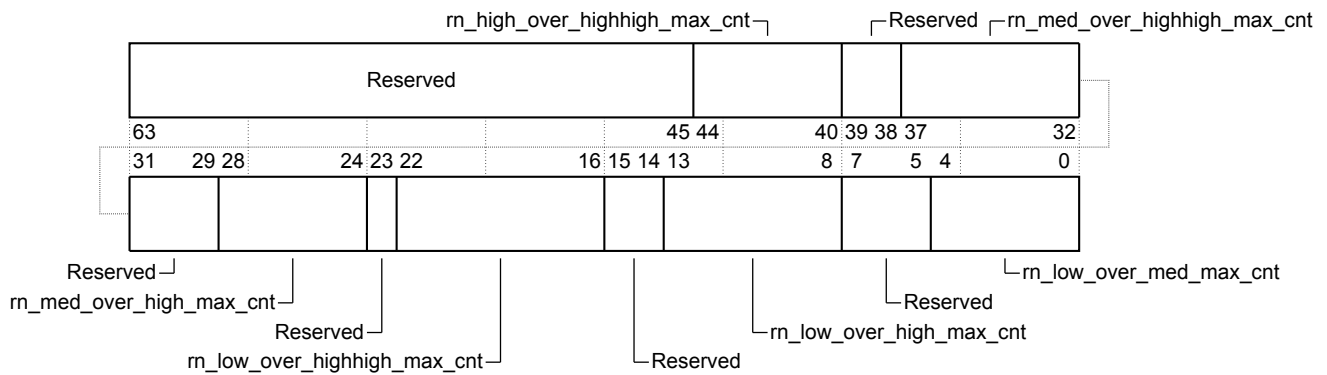
Figure 3-66 rn_starvation register bit assignments

The following table shows the rn_starvation register bit assignments.

Table 3-80 rn_starvation register bit assignments

Bits	Name	Access	Reset value	Function
[63:45]	-	RAZ/WI	`0x0`	Reserved
[44:40]	rn_high_over_high_high_max_cnt	RW	`0x1F`	Maximum number of consecutive times highest QoS class win over high QoS class
[39:38]	-	RAZ/WI	`0x0`	Reserved
[37:32]	rn_med_over_highhigh_max_cnt	RW	`0x3F`	Maximum number of consecutive times highest QoS class win over medium QoS class
[31:29]	-	RAZ/WI	`0x0`	Reserved
[28:24]	rn_med_over_high_max_cnt	RW	`0x1F`	Maximum number of consecutive times high QoS class win over medium QoS class
[23]	-	RAZ/WI	0	Reserved
[22:16]	rn_low_over_highhigh_max_cnt	RW	`0x3F`	Maximum number of consecutive times highest QoS class win over low QoS class
[15:14]	-	RAZ/WI	`0x0`	Reserved
[13:8]	rn_low_over_high_max_cnt	RW	`0x3F`	Maximum number of consecutive times high QoS class win over low QoS class
[7:5]	-	RAZ/WI	`0x0`	Reserved
[4:0]	rn_low_over_med_max_cnt	RW	`0x1F`	Maximum number of consecutive times medium QoS class win over low QoS class

HN-F Error Injection Enable and Setup register

The hnf_err_inj register is at offset `0x0038`. Its characteristics are:

Purpose   Error injection enable and setup register. When enabled for a specific SrcID and LPID, the HN-F returns a slave error and reports an error interrupt through the MN to emulate an L3 double bit data ECC error. This feature enables software to test the error handler. A slave error is reported for a cacheable read access when an L3 hit is the source of the data. For a cacheable read access that results in an L3 miss, no slave error or error interrupt is reported.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

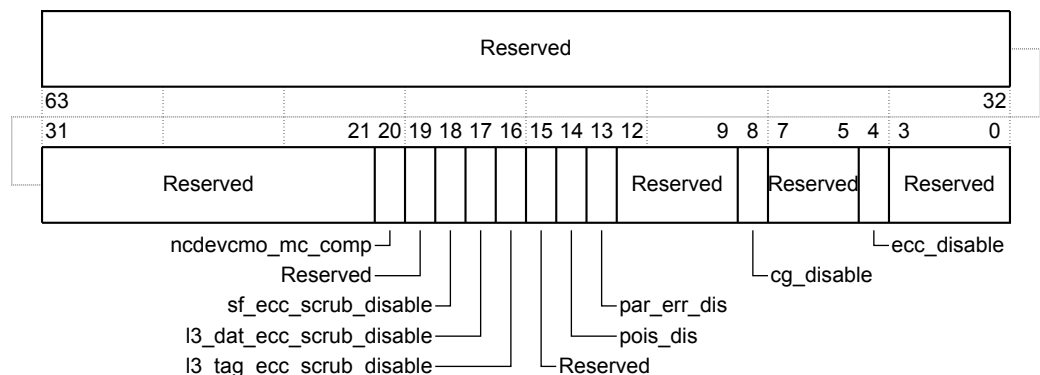The following figure shows the hnf_err_inj register bit assignments.
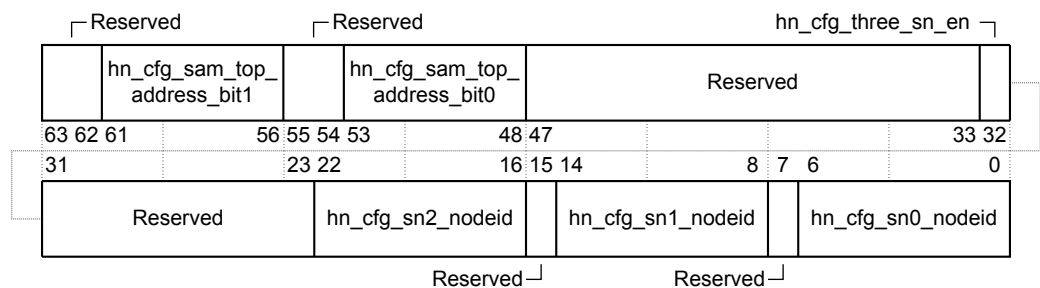

Figure 3-67 hnf_err_inj register bit assignments

The following table shows the hnf_err_inj register bit assignments.

Table 3-81 hnf_err_inj register bit assignments

Bits	Name	Access	Reset value	Function
[63:23]	-	RAZ/WI	0x0	Reserved
[22:16]	hnf_err_inj_srcid	RW	0x0	SrcID read access that results in an L3 miss, with no slave error or error to match for HN-F error injection
[15:7]	-	RAZ/WI	0x0	Reserved
[6:4]	hnf_err_inj_lpid	RW	0x0	LPID to match for HN-F error injection
[3:1]	-	RAZ/WI	0x0	Reserved
[0]	hnf_err_inj_en	RW	0	HN-F error injection and report enable

HN-F L3 Lock Ways register

The hnf_l3_lock_ways register is at offset `0x0040`. Its characteristics are:

Purpose   Controls the number of locked HN-F L3 ways. This can be a value of 1, 2, 4, 8, or 12.

Usage constraints   Only accessible by Secure accesses. The L3 must be flushed before writing this register, and no non-configuration accesses to the HN-F can be in-flight while the write to this register is occurring.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the bit assignments.

Figure 3-68 hnf_l3_lock_ways register bit assignments

The following table shows the bit assignments.

Table 3-82 hnf_l3_lock_ways register bit assignments

Bits	Name	Access	Reset value	Function
[63:4]	-	RAZ/WI	0x0	Reserved
[3:0]	ways	RW	0x0	Number of ways locked

HN-F L3 Lock Base 0 register

The hnf_l3_lock_base0 register is at offset `0x0048`. Its characteristics are:

Purpose   Base register for lock range 0 [43:0].

Usage constraints Only accessible by Secure accesses. The L3 must be flushed before writing this register, and no non-configuration accesses to the HN-F can be in-flight while the write to this register is occurring.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the bit assignments.
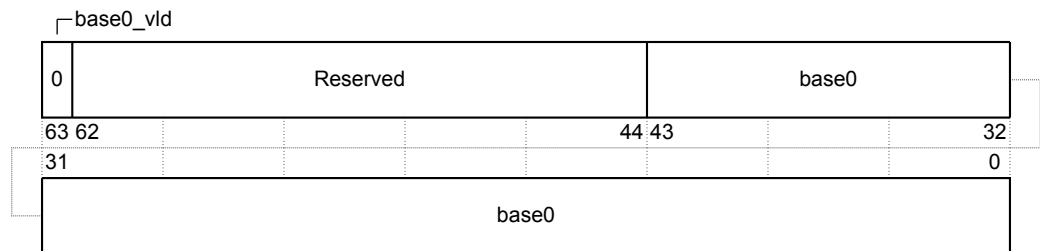
Figure 3-69 hnf_l3_lock_base0 register bit assignments

The following table shows the bit assignments.

Table 3-83 hnf_l3_lock_base0 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	base0_vld	RW	0	Lock base 0 valid
[62:44]	-	RAZ/WI	0x0	Reserved
[43:0]	base0	RW	0x0	Lock base 0

HN-F L3 Lock Base 1 register

The hnf_l3_lock_base1 register is at offset `0x0050`. Its characteristics are:

Purpose   Base register for lock range 1 [43:0].

Usage constraints Only accessible by Secure accesses. The L3 must be flushed before writing this register, and no non-configuration accesses to the HN-F can be in-flight while the write to this register is occurring.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.
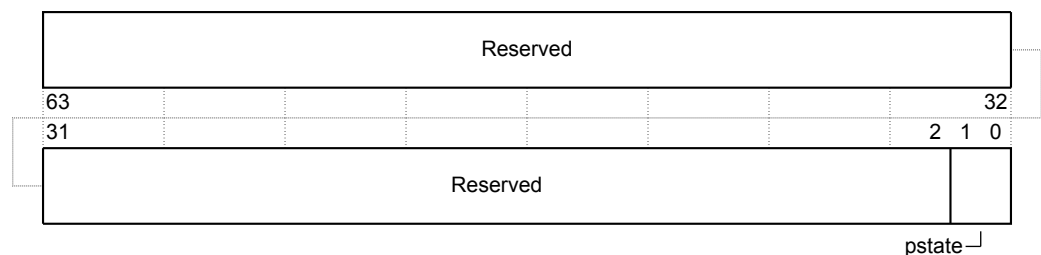
The following figure shows the bit assignments.
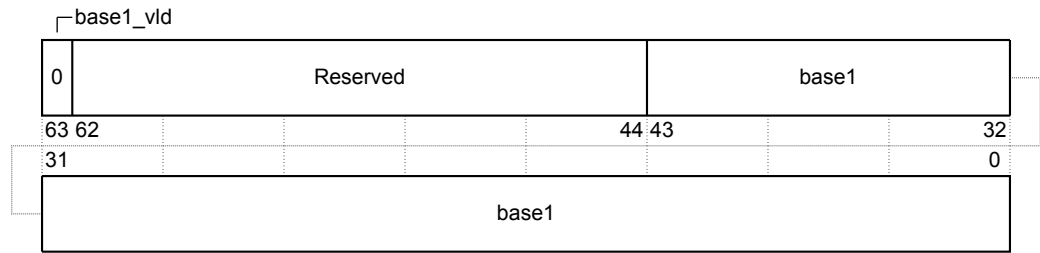
Figure 3-70 hnf_l3_lock_base1 register bit assignments

The following table shows the bit assignments.

Table 3-84 hnf_l3_lock_base1 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	base1_vld	RW	0	Lock base 1 valid
[62:44]	-	RAZ/WI	0x0	Reserved
[43:0]	base1	RW	0x0	Lock base 1

HN-F L3 Lock Base 2 register

The hnf_l3_lock_base2 register is at offset `0x0058`. Its characteristics are:

Purpose Base register for lock range 2 [43:0].

Usage constraints Only accessible by Secure accesses. The L3 must be flushed before writing this register, and no non-configuration accesses to the HN-F can be in-flight while the write to this register is occurring.

Configurations Available in all configurations.

Attributes See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the bit assignments.

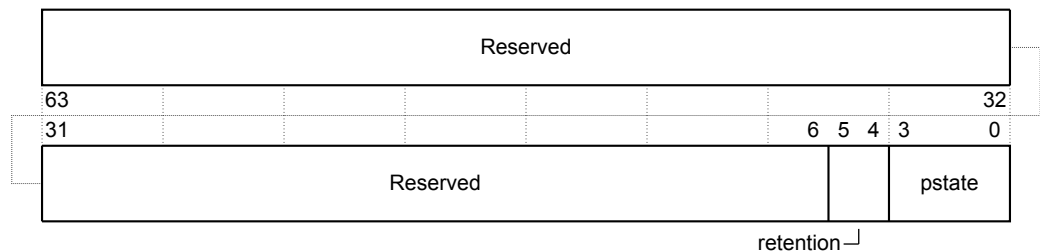**Figure 3-71 hnf_l3_lock_base2 register bit assignments**

The following table shows the bit assignments.

Table 3-85 hnf_l3_lock_base2 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	base2_vld	RW	0	Lock base 2 valid
[62:44]	-	RAZ/WI	0x0	Reserved
[43:0]	base2	RW	0x0	Lock base 2

HN-F L3 Lock Base 3 register

The hnf_l3_lock_base3 register is at offset `0x0060`. Its characteristics are:

|	
Purpose	Base register for lock range 3 [43:0].
Usage constraints	Only accessible by Secure accesses. The L3 must be flushed before writing this register, and no non-configuration accesses to the HN-F can be in-flight while the write to this register is occurring.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the bit assignments.

Figure 3-72 hnf_l3_lock_base3 register bit assignments

The following table shows the bit assignments.

Table 3-86 hnf_l3_lock_base3 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	base3_vld	RW	0	Lock base 3 valid
[62:44]	-	RAZ/WI	0x0	Reserved
[43:0]	base3	RW	0x0	Lock base 3

HN-F Byte Parity Error Injection register

The hnf_byte_par_err_inj register is at offset `0x0068`. Its characteristics are:

Purpose	Selects a byte lane, within the 128-bit data bus, and injects a byte parity error on the DAT flits when the next L3 hit occurs.
Usage constraints	Only accessible by Secure accesses.
Configurations	Available in all configurations.
Attributes	See *Table 3-5 XP register summary* on page 3-88.

The following figure shows the hnf_byte_par_err_inj register bit assignments.
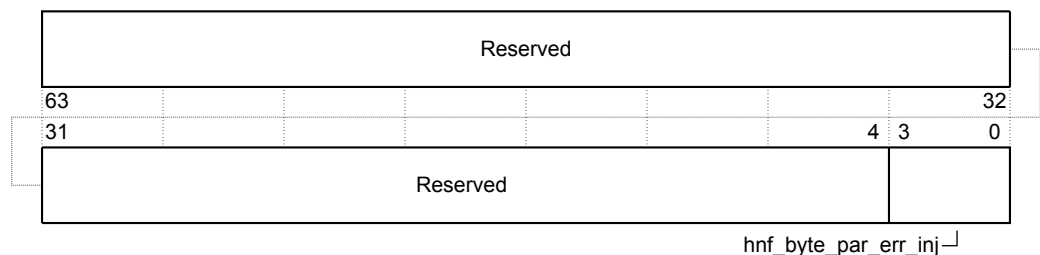
Figure 3-73 hnf_byte_par_err_inj register bit assignments

The following table shows the hnf_byte_par_err_inj register bit assignments.

Table 3-87 byte_par_err_inj register bit assignments

Bits	Name	Access	Reset value	Function
[63:4]	-	RAZ/WI	`0x0`	Reserved.
[3:0]	hnf_byte_par_err_inj	WO	-	The value selects a byte lane within the 128-bit data bus. The CCN injects a byte parity error on the chosen byte lane in 4 DAT flits, when the next L3 hit occurs. The bit values are: `0b0000` Inserts a parity error in bits[7:0]. `0b0001` Inserts a parity error in bits[15:8]. `0b0010` Inserts a parity error in bits[23:16]. **...** … `0b1111` Inserts a parity error in bits[127:120]. If multiple writes occur to this field before the HN-F generates the 4 DAT flits, then the HN-F uses the initial value that is written and ignores the subsequent writes.

Related concepts

2.16.2 Byte parity error injection on page 2-80.

HN Configuration RN-I Vector register

The hn_cfg_rni_vec register is at offset `0x0108`. Its characteristics are:

Purpose	Indicates which SrcIDs are RN-I protocol nodes.
Usage constraints	Writes to this register must be complete before the first coherent access to the HN-F.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

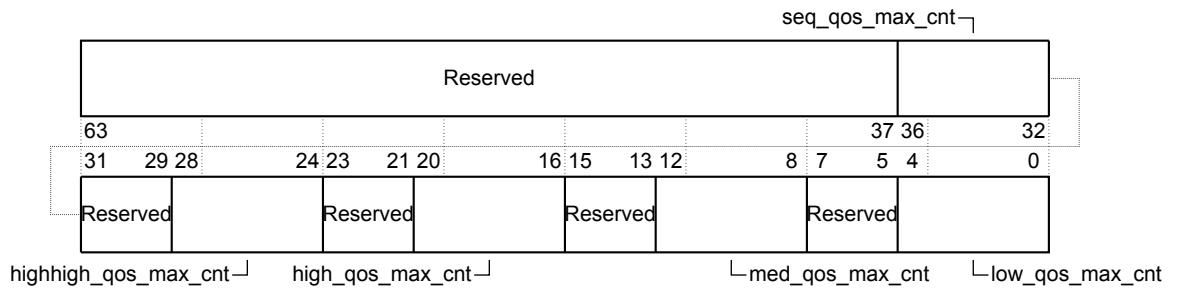The following figure shows the hn_cfg_rni_vec register bit assignments.
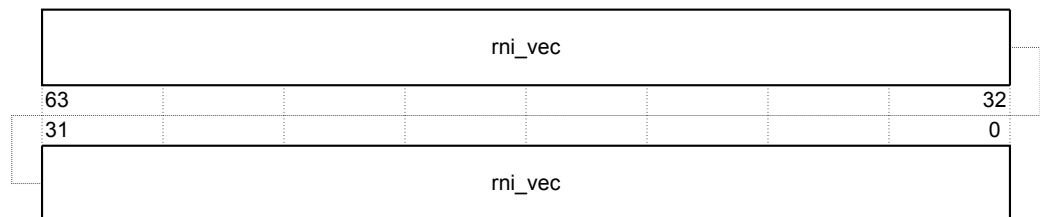
Figure 3-74 hn_cfg_rni_vec register bit assignments

The following table shows the hn_cfg_rni_vec register bit assignments.

Table 3-88 hn_cfg_rni_vec register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	rni_vec	RW	Value depends on customer configuration	Bit vector representing all the RN-I NodeIDs

Snoop Domain Control register

The snoop_domain_ctl register is at offset `0x0200`. Its characteristics are:

Purpose	Determines the RN-F targets for snoops. Every RN-F node that is actively participating in cache coherence has its respective bit set. If the bit is clear, the corresponding RN-F node is not snooped.
Usage constraints	This register must be configured correctly, using the snoop_domain_ctl_set and snoop_domain_ctl_clr registers, before the first coherent access to the HN-F.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the snoop_domain_ctl register bit assignments.

Figure 3-75 snoop_domain_ctl register bit assignments

The following table shows the snoop_domain_ctl register bit assignments.

Table 3-89 snoop_domain_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	snoop_domain_ctl	RO	0x0	Bit vector representing RN-F nodes that can be snooped

Snoop Domain Control Set register

The snoop_domain_ctl_set register is at offset `0x0210`. Its characteristics are:

Purpose	Inserts RN-Fs into the active snoop domain, setting the corresponding bit in the snoop_domain_ctl register, and causing the RN-Fs to receive and requiring response to snoops.
Usage constraints	Only accessible by Secure accesses.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the snoop_domain_ctl_set register bit assignments.

Figure 3-76 snoop_domain_ctl_set register bit assignments

The following table shows the snoop_domain_ctl_set register bit assignments.

Table 3-90 snoop_domain_ctl_set register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	snoop_domain_ctl_set	WO	0x0	Bit vector indicating the NodeIDs of the RN-Fs to be inserted into the active snoop domain

Snoop Domain Control Clear register

The snoop_domain_ctl_clr register is at offset `0x0220`. Its characteristics are:

Purpose	Removes RN-Fs from the active snoop domain, clearing the corresponding bit in the snoop_domain_ctl register, and causing the RN-Fs to no longer receive or be allowed to respond to snoops.
Usage constraints	Only accessible by Secure accesses.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the snoop_domain_ctl_clr register bit assignments.

Figure 3-77 snoop_domain_ctl_clr register bit assignments

The following table shows the snoop_domain_ctl_clr register bit assignments.

Table 3-91 snoop_domain_ctl_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	snoop_domain_ctl_clr	WO	0x0	Bit vector indicating the NodeIDs of the RN-Fs to be removed from the active snoop domain

HN Debug Read Configuration register

The hn_cfg_l3sf_dbgrd register is at offset `0x0300`. Its characteristics are:

Purpose	Controls access to the L3 tag, data, and snoop filter.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the hn_cfg_l3sf_dbgrd register bit assignments.

Figure 3-78 hn_cfg_l3sf_dbgrd register bit assignments

The following table shows the hn_cfg_l3sf_dbgrd register bit assignments.

Table 3-92 hn_cfg_l3sf_dbgrd register bit assignments

Bits	Name	Access	Reset value	Function
[63:26]	-	RAZ/WI	0x0	Reserved
[25:24]	l3_access_component	WO	0b00	L3/SF debug read array specifier: 0b01 L3 data read. 0b10 L3 tag read. 0b11 SF tag read.
[23]	-	RAZ/WI	0	Reserved
[22:20]	l3_access_ow	WO	0x0	64-bit chunk address for L3 data debug read access.
[19:16]	l3_access_way	WO	0x0	Way address for L3/SF debug read access.
[15:12]	-	RAZ/WI	0x0	Reserved
[11:0]	l3_access_set	WO	0x0	Set address for L3/SF debug read access.

——————— **Note** ———————

If a debug read is performed to an array entry that has not yet been initialized or written, the value of the data that is returned is indeterminate.

——————————————————

L3 Cache Access Tag register

The l3_cache_access_l3_tag register is at offset 0x0308. Its characteristics are:

Purpose   Indicates L3 cache tag storage.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the l3_cache_access_l3_tag register bit assignments.

Reserved			l3_cache_access_l3_tag
63		44	43 32
31			0
l3_cache_access_l3_tag			

Figure 3-79 l3_cache_access_l3_tag register bit assignments

The following table shows the l3_cache_access_l3_tag register bit assignments.

Table 3-93 l3_cache_access_l3_tag register bit assignments

Bits	Name	Access	Reset value	Function
[63:44]	-	RAZ/WI	0x0	Reserved
[43:0]	l3_cache_access_l3_tag	RO	0x0	L3 tag debug read data register

L3 Cache Access Data register

The l3_cache_access_l3_data register is at offset `0x0310`. Its characteristics are:

Purpose   Indicates L3 cache data storage.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the l3_cache_access_l3_data register bit assignments.

Figure 3-80 l3_cache_access_l3_data register bit assignments

The following table shows the l3_cache_access_l3_data register bit assignments.

Table 3-94 l3_cache_access_l3_data register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	l3_cache_access_l3_data	RO	0x0	L3 data debug read data register

L3 Cache Access SF Tag register

The l3_cache_access_sf_tag register is at offset `0x0318`. Its characteristics are:

Purpose   Indicates L3 cache SF tag storage.

Usage constraints   Only accessible by Secure accesses.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

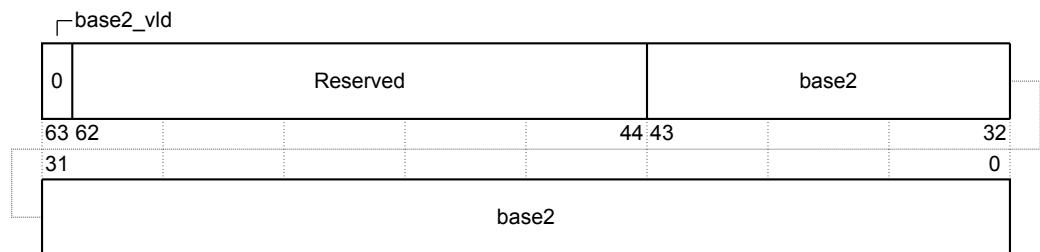The following figure shows the l3_cache_access_sf_tag register bit assignments.

Figure 3-81 l3_cache_access_sf_tag register bit assignments

The following table shows the l3_cache_access_sf_tag register bit assignments.

Table 3-95 l3_cache_access_sf_tag register bit assignments

Bits	Name	Access	Reset value	Function
[63:44]	-	RAZ/WI	0x0	Reserved
[43:0]	l3_cache_access_sf_tag	RO	0x0	SF tag debug read data register

Error Syndrome 0 register, L3 cache

The err_syndrome_reg0 register is at offset `0x0400`. Its characteristics are:

Purpose   Indicates bit errors in the L3 cache.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.
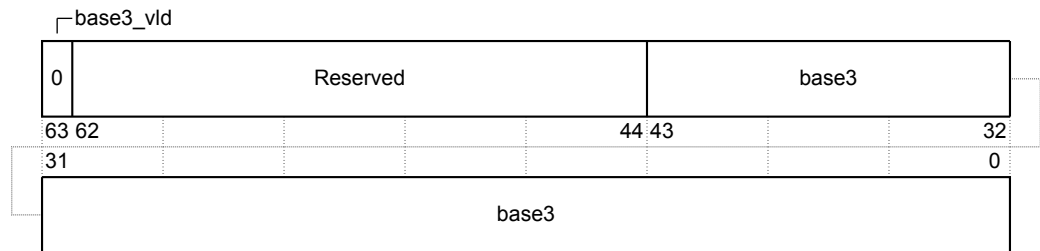
The following figure shows the err_syndrome_reg0 register bit assignments.

Figure 3-82 err_syndrome_reg0 register bit assignments

The following table shows the err_syndrome_reg0 register bit assignments.

Table 3-96 err_syndrome_reg0 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	err_exntd	RO	0	Error extended.
[62]	first_err_vld	RO	0	First error valid.
[61:60]	err_class	RO	0x0	Error classification.
[59]	mult_err	RO	0	Multiple errors.
[58:43]	err_count	RO	0x0	Corrected error count.
[42:20]	-	RAZ/WI	0x0	Reserved
[19:8]	err_count_set	RO	0x0	HN-F single-bit ECC error count set address.
[7]	err_count_ovrflw	RO	0	HN-F single-bit error counter overflow.
[6]	err_count_match	RO	0	HN-F single-bit ECC error count applies to same type and set.
[5:4]	err_count_type	RO	0b00	HN-F single-bit ECC counter type: 0b00 L3 data single-bit count. 0b01 L3 tag single-bit count. 0b10 SF tag single-bit count.
[3]	par_err_id	RO	0	Byte parity error.
[2:0]	err_id	RO	0b000	HN-F error syndrome register error type: 0b100 L3 data double-bit ECC error. 0b101 L3 tag double-bit ECC error. 0b110 SF tag double-bit ECC error. 0b111 CHI bus slave error.

Related concepts

Error logging on page 2-46.

Error Syndrome 1 register, L3 cache

The err_syndrome_reg1 register is at offset `0x0408`. Its characteristics are:

Purpose	Indicates the address of the first tag error.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the err_syndrome_reg1 register bit assignments.

Figure 3-83 err_syndrome_reg1 register bit assignments

The following table shows the err_syndrome_reg1 register bit assignments.

Table 3-97 err_syndrome_reg1 register bit assignments

Bits	Name	Access	Reset value	Function
[63:55]	Reserved	RAZ/WI	`0x0`	-
[54:48]	err_srcid	RO	`0b0000000`	HN-F error syndrome SrcID[6:0] for byte parity errors only
[47:46]	Reserved	RAZ/WI	`0b00`	-
[45:44]	err_optype	RO	`0b00`	HN-F error syndrome OpType[1:0] for byte parity errors only `0b00` WRUNIQ or WRLUNIQ. `0b01` WRBACKPTL. `0b10` WRNOSNP or WRNOSNPFULL. `0b11` All others.
[43:0]	err_addr	RO	`0x0`	HN-F error syndrome address for double-bit ECC or byte parity errors only

Related concepts

2.16.1 Parity error reporting, poisoning, and logging on page 2-80.
Error logging on page 2-46.

L3 cache Error Syndrome Clear register

The err_syndrome_clr register is at offset `0x0480`. Its characteristics are:

Purpose	Clears the error log in the Error Syndrome 0 register.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-6 HN-F register summary* on page 3-89.

The following figure shows the err_syndrome_clr register bit assignments.

Figure 3-84 err_syndrome_clr register bit assignments

The following table shows the err_syndrome_clr register bit assignments.

Table 3-98 err_syndrome_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0	Reserved
[62]	first_err_vld_clr	WO	0	Clears the first_err_vld bit in the Error Syndrome 0 register
[61:60]	-	RAZ/WI	0b00	Reserved
[59]	mult_err_clr	WO	0	Clears the mult_err bit in the Error Syndrome 0 register
[58:0]	-	RAZ/WI	0x0	Reserved

Related references

Error Syndrome 0 register, L3 cache on page 3-159.

Error log clearing on page 2-47.

HN-F Auxiliary Control register

The hnf_aux_ctl register is at offset 0x0500. Its characteristics are:

Purpose   Controls various modes of HN-F operation.

Usage constraints   This register can be modified only with prior written permission from ARM.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

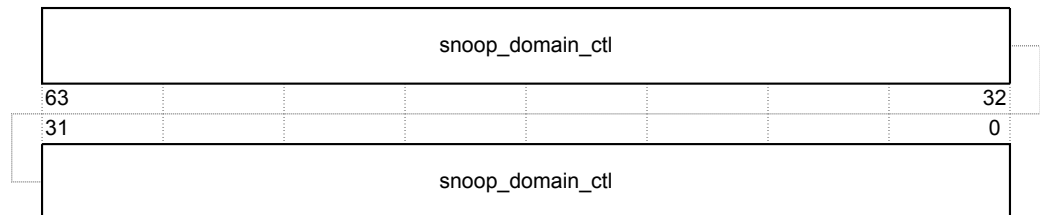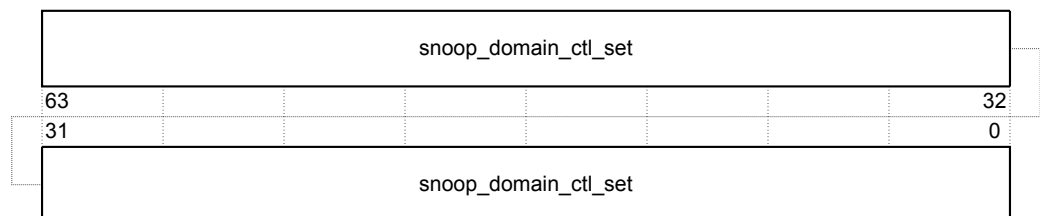The following figure shows the hnf_aux_ctl register bit assignments.
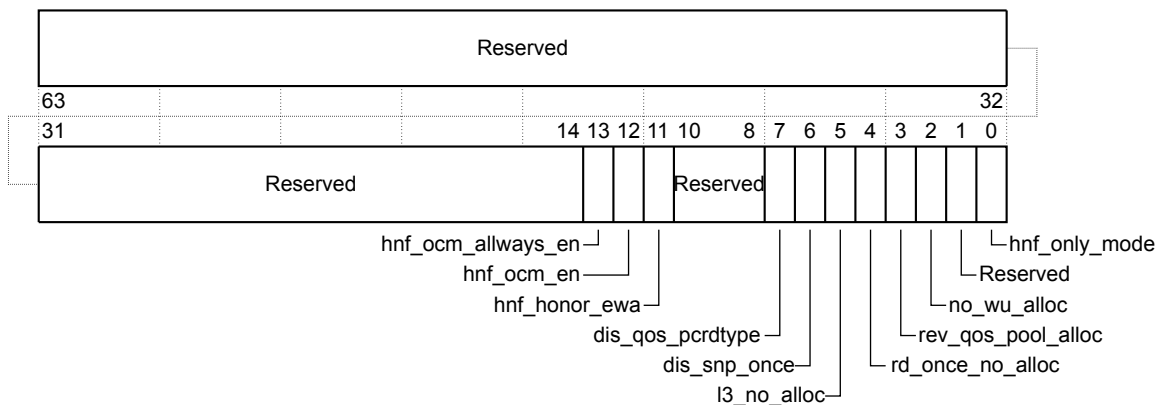
Figure 3-85 hnf_aux_ctl register bit assignments

The following table shows the hnf_aux_ctl register bit assignments.

Table 3-99 hnf_aux_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:14]	-	RAZ/WI	`0x0`	Reserved
[13]	hnf_ocm_allways_en	RW	0	All L3 way OCM support enable.
[12]	hnf_ocm_en	RW	0	Region lock with OCM support enable.
[11]	hnf_honor_ewa	RW	0	This bit controls whether the HN-F honors the state of the *Early Write Acknowledge* (EWA) bit within the MemAttr field of a REQ flit: 0 = The HN-F ignores the state of the EWA bit. Therefore, the HN-F can send a write completion response before it receives completion from the SN. 1 = The HN-F honors the state of the EWA bit. If EWA = 0, then the HN-F only sends Completion when it receives a completion from the SN.
[10:8]	-	RAZ/WI	`0b000`	Reserved
[7]	dis_qos_pcrdtype	RW	0	Disable QoS based PCrdType assignment.
[6]	dis_snp_once	RW	Value depends on customer configuration	Disable SnpOnce. SnpOnce is converted to SnpShared.
[5]	l3_no_alloc	RW	0	Disable L3 allocation for Non-shareable Cacheable transactions.
[4]	rd_once_no_alloc	RW	0	Disable ReadOnce allocation in the L3 from RN-Is.
[3]	rev_qos_pool_alloc	RW	0	Reverse QoS pool allocation algorithm.
[2]	no_wu_alloc	RW	0	Disable WriteUnique and WriteLineUnique allocations in L3.
[1]	-	RAZ/WI	0	Reserved
[0]	hnf_only_mode	RW	0	HN-F-only mode with no L3 and snoop filter.

PMU Event Select register, L3 cache

The pmu_event_sel register is at offset `0x0600`. Its characteristics are:

Purpose   Selects the PMU events to be counted.

Usage constraints Before any field in this register can be selected for transmission by the debug and test control logic in the XP, that field must be set to a valid value other than `0x0`.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

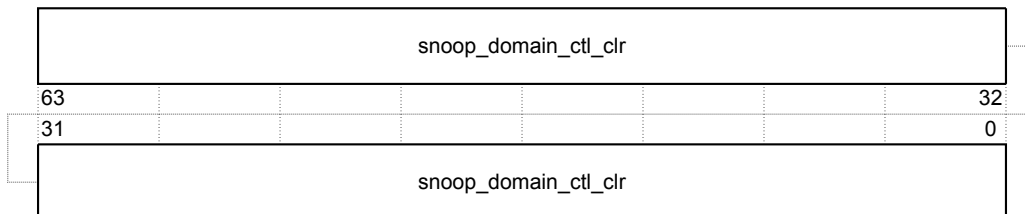The following figure shows the pmu_event_sel register bit assignments.
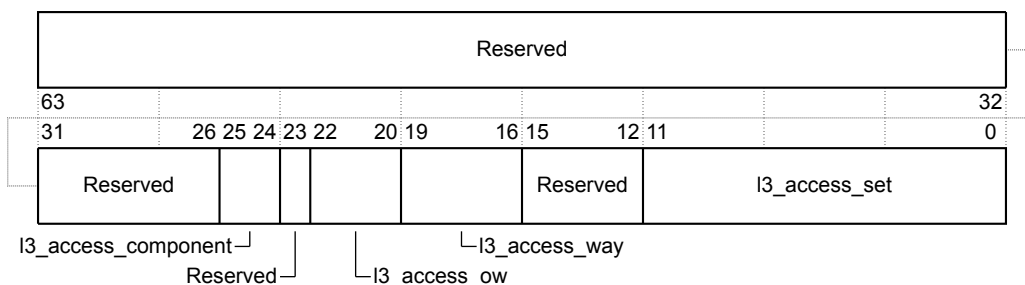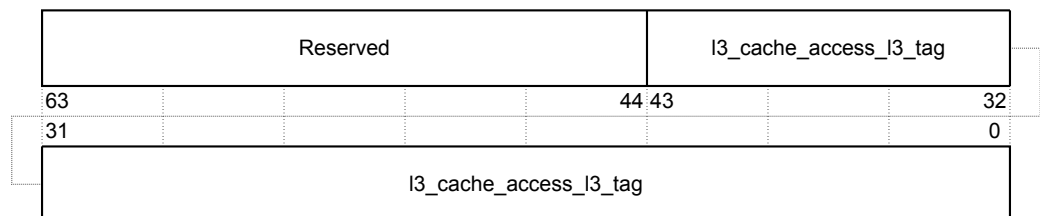
Figure 3-86 pmu_event_sel register bit assignments

The following table shows the pmu_event_sel register bit assignments.

Table 3-100 pmu_event_sel register bit assignments

Bits	Name	Access	Reset value	Function
[63:16]	-	RAZ/WI	0x0	Reserved
[15:12]	pmu_event3_id	RW	0x0	PMU Event 3 ID. The event is specified as a 4-bit ID with the following encodings: 0b0000 Null (no event). 0b0001 PMU_HN_CACHE_MISS_EVENT. Counts the total cache misses. This is the first time lookup result, and is high priority. 0b0010 PMU_HNL3_SF_CACHE_ACCESS_EVENT. Counts the number of cache accesses. This is the first time access, and is high priority. 0b0011 PMU_HN_CACHE_FILL_EVENT. Counts the total allocations in the HN L3 cache, and all cache line allocations to the L3 cache. 0b0100 PMU_HN_POCQ_RETRY_EVENT. Counts the number of requests that have been retried. 0b0101 PMU_HN_POCQ_REQS_RECVD_EVENT. Counts the number of requests received by HN. 0b0110 PMU_HN_SF_HIT_EVENT. Counts the number of snoop filter hits. 0b0111 PMU_HN_SF_EVICTIONS_EVENT. Counts the number of snoop filter evictions. Cache invalidations are initiated. 0b1000 PMU_HN_SNOOPS_SENT_EVENT. Counts the number of snoops sent. Does not differentiate between broadcast or directed snoops. 0b1001 PMU_HN_SNOOPS_BROADCAST_EVENT. Counts the number of snoop broadcasts sent. 0b1010 PMU_HN_L3_EVICTION_EVENT. Counts the number of L3 evictions. 0b1011 PMU_HN_L3_FILL_INVALID_WAY_EVENT. Counts the number of L3 fills to an invalid way. 0b1100 PMU_HN_MC_RETRIES_EVENT. Counts the number of transactions retried by the memory controller. 0b1101 PMU_HN_MC_REQS_EVENT. Counts the number of requests to the memory controller. 0b1110 PMU_HN_QOS_HH_RETRY_EVENT. Counts the number of times a highest-priority QoS class was retried at the HN-F. All other values are Reserved.

Bits	Name	Access	Reset value	Function
[11:8]	pmu_event2_id	RW	0x0	PMU Event 2 ID. See pmu_event3_id in this table for more information.
[7:4]	pmu_event1_id	RW	0x0	PMU Event 1 ID. See pmu_event3_id in this table for more information.
[3:0]	pmu_event0_id	RW	0x0	PMU Event 0 ID. See pmu_event3_id in this table for more information.

HN-F Identification register

The oly_hnf_oly_id register is at offset 0xFF00. Its characteristics are:

Purpose   Contains the component identification information.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-6 HN-F register summary* on page 3-89.

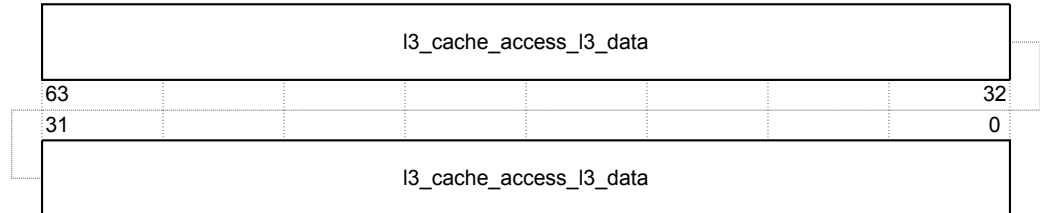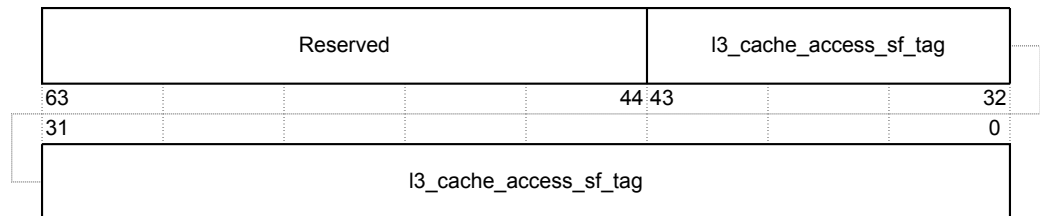The following figure shows the oly_hnf_oly_id register bit assignments.

Figure 3-87 oly_hnf_oly_id register bit assignments

The following table shows the oly_hnf_oly_id register bit assignments.

Table 3-101 oly_hnf_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	Value is specific to each HN-F	The node ID of the HN-F
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x4	Indicates that this node is an HN-F

Related references

3.3.4 HN-I register descriptions

This section lists the HN-I registers.

PoS Control register

The pos_control register is at offset `0x0000`. Its characteristics are:

Purpose   Selects *Point-of-Serialization* (PoS) related features.

Usage constraints   Before writing this register, ensure that all previous transactions to the HN-I have completed, and then you must issue and wait for completion of a DSB or ECBarrier.

Configurations   Available in all configurations.

Attributes   See *Table 3-7 HN-I register summary* on page 3-90.

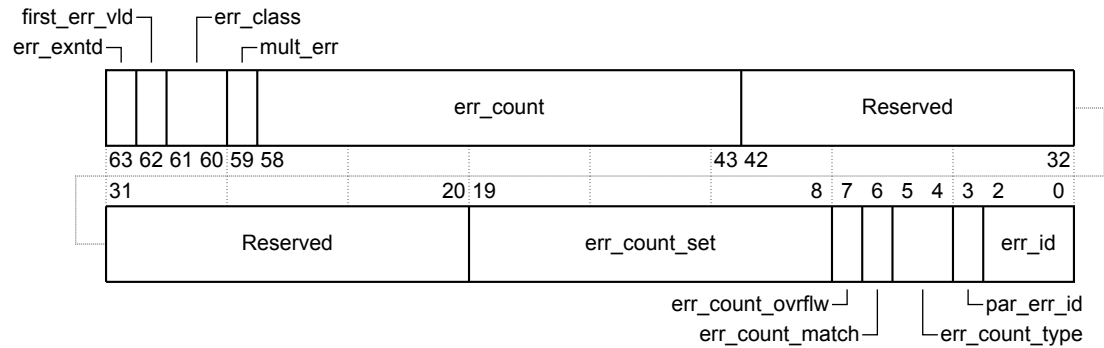The following figure shows the pos_control register bit assignments.
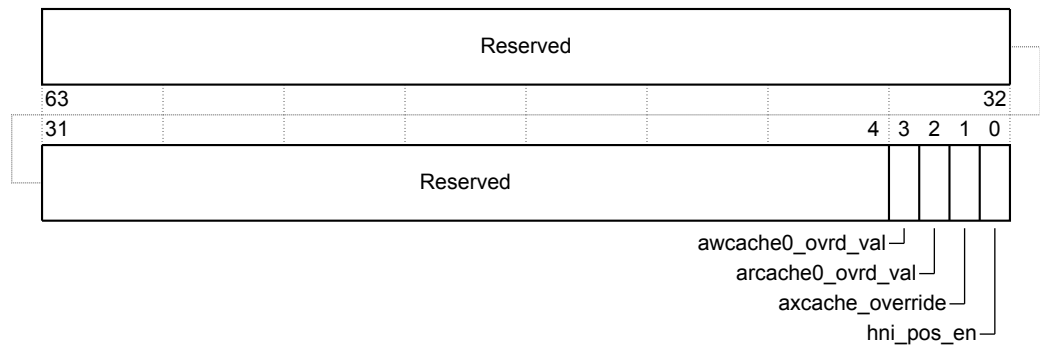
Figure 3-88 pos_control register bit assignments

The following table shows the pos_control register bit assignments.

Table 3-102 pos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:4]	-	RAZ/WI	`0x0`	Reserved.
[3]	awcache0_ovrd_val	RW	0	If bit[1] of this register is set, **AWCACHE[0]** is driven from this bit.
[2]	arcache0_ovrd_val	RW	0	If bit[1] of this register is set, **ARCACHE[0]** is driven from this bit.
[1]	axcache_override	RW	0	Set to 1 to override **AWCACHE[0]** and **ARCACHE[0]** on the AMBA interface.
[0]	hni_pos_en	RW	1	Indicates status of HN-I PoS: **1** HN-I is final PoS. **0** HN-I is not final PoS. See the pos_* control bits in the sa_aux_ctl register. Violates the CHI GO definition when the hni_pos_en bit is 0.

Related references

SA Auxiliary Control register, HN-I on page 3-169.

PCIeRC RN-I Node ID List register

The pcierc_rni_nodeid_list register is at offset `0x008`. Its characteristics are:

Purpose	A bit vector showing the list of all RN-Is with PCIe RC connected in the system.
Usage constraints	Can be read from in ALL states. Cannot be changed.
Configurations	Available in all configurations.
Attributes	See *Table 3-7 HN-I register summary* on page 3-90.

The following figure shows the bit assignments.
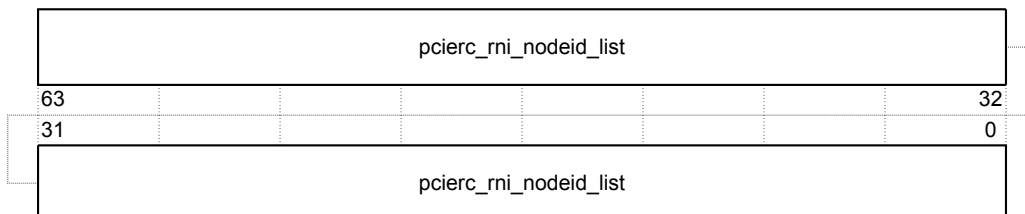
Figure 3-89 pcierc_rni_nodeid_list register bit assignments

The following table shows the bit assignments.

Table 3-103 pcierc_rni_nodeid_list Register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	pcierc_rni_nodeid_list	RW	0x0	A bit vector showing the list of all RN-Is with PCIe RC connected in the system.

Error Syndrome 0 register, HN-I

The err_syndrome_reg0 register is at offset `0x0400`. Its characteristics are:

Purpose	Indicates the HN-I error log information.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-7 HN-I register summary* on page 3-90.

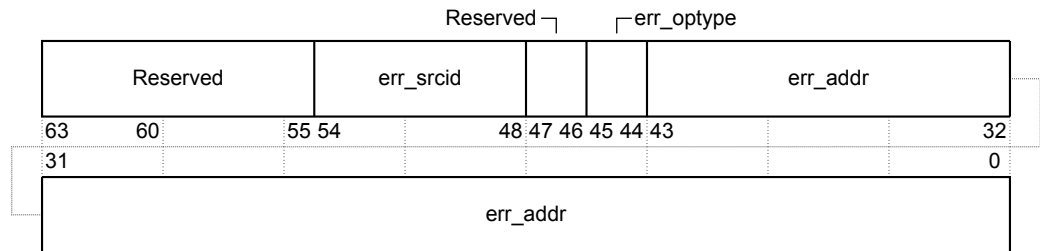The following figure shows the err_syndrome_reg0 register bit assignments.

Figure 3-90 err_syndrome_reg0 register bit assignments

The following table shows the err_syndrome_reg0 register bit assignments.

Table 3-104 err_syndrome_reg0 register bit assignments

Bits	Name	Access	Reset value	Function
[63]	err_exntd	RO	0	Error extended.
[62]	first_err_vld	RO	0	First error valid.
[61:60]	err_class	RO	0x0	Error classification.
[59]	mult_err	RO	0	Multiple errors.
[58:43]	corrected_err_count	RO	0x0	Corrected error count.
[42:0]	component_specific_reg0	RO	0x0	Component-specific error information: **Bits[42:4]** Error log 1: **[42]** Reserved. **[41:40]** PCrdType. **[39:37]** SIZE. **[36:35]** SnpAttr. **[34:31]** MemAttr. **[30:29]** ORDER. **[28]** NS. **[27]** DYNPCRD. **[26:16]** TXNID[b]. **[18:16]** LPID. **[15:9]** SRCID[b]. **[8:4]** OPCODE. **Bits[3:0]** Error type: 0b0001 Unsupported opcode (CMO/CU/MU). 0b0010 Cacheable read request. 0b0011 Cacheable write request. 0b0100 Downstream write response error. 0b0101 MN read request. 0b0110 MN write request. 0b0101 MN unsupported opcode (CMO/CU/MU). All other values are Reserved.

Related concepts

Error logging on page 2-46.

Error Syndrome 1 register, HN-I

The err_syndrome_reg1 register is at offset `0x0408`. Its characteristics are:

Purpose	Indicates the HN-I error log information.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-7 HN-I register summary* on page 3-90.

---

[b]   These fields are logged for downstream write response errors.

The following figure shows the err_syndrome_reg1 register bit assignments.

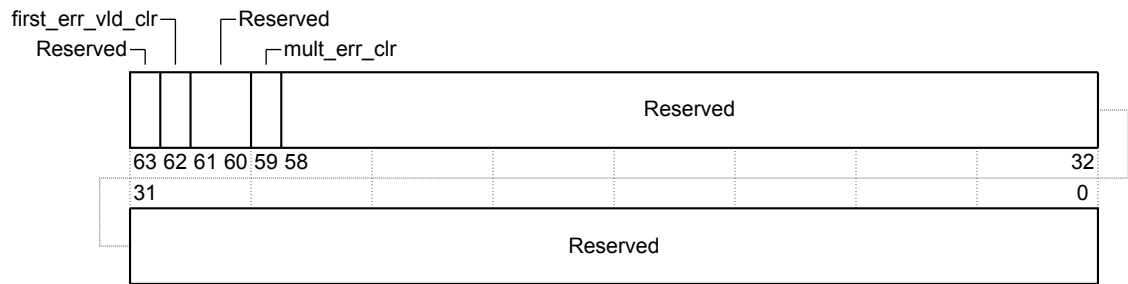| component_specific_reg1	
63	32
31	0
component_specific_reg1	
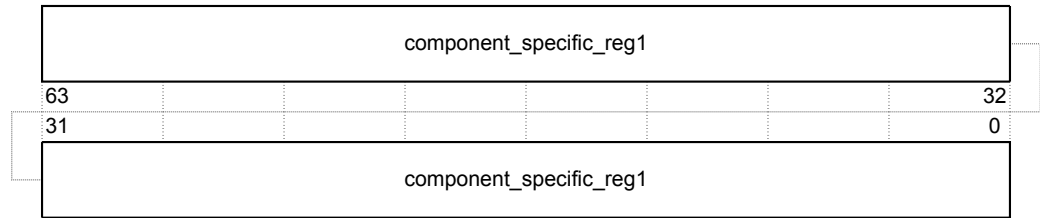
Figure 3-91 err_syndrome_reg1 register bit assignments

The following table shows the err_syndrome_reg1 register bit assignments.

Table 3-105 err_syndrome_reg1 register bit assignments

Bits	Name	Access	Reset value	Function
[63:0]	component_specific_reg1	RO	0x0	Component-specific error information extended: Error Log_2[43:0] = Address[43:0]. Error Log_2[63:44] = Reserved.

Related concepts

Error logging on page 2-46.

HN-I Error Syndrome Clear register

The err_syndrome_clr register is at offset `0x0480`. Its characteristics are:

Purpose	Clears the error log in the Error Syndrome 0 register.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-7 HN-I register summary* on page 3-90.

The following figure shows the err_syndrome_clr register bit assignments.

first_err_vld_clr
Reserved ┐ Reserved
Reserved ┐ │ ┌ mult_err_clr

				Reserved	
63	62 61 60	59 58			32
31					0
Reserved					
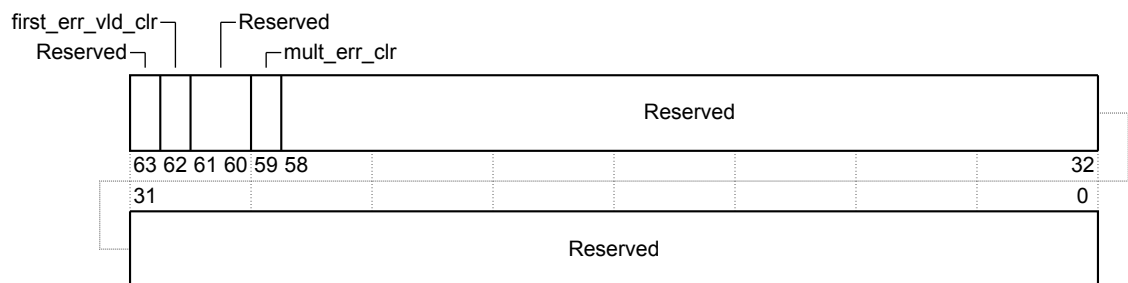
Figure 3-92 err_syndrome_clr register bit assignments

The following table shows the err_syndrome_clr register bit assignments.

Table 3-106 err_syndrome_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63]	-	RAZ/WI	0	Reserved
[62]	first_err_vld_clr	WO	0	Clears the first_err_vld bit in the Error Syndrome 0 register
[61:60]	-	RAZ/WI	0b00	Reserved

Table 3-106 err_syndrome_clr register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[59]	mult_err_clr	WO	0	Clears the mult_err bit in the Error Syndrome 0 register
[58:0]	-	RAZ/WI	0x0	Reserved

Related references

Error Syndrome 0 register, HN-I on page 3-166.

Error log clearing on page 2-47.

SA Auxiliary Control register, HN-I

The sa_aux_ctl register is at offset 0x0500. Its characteristics are:

Purpose   Auxiliary control of the HN-I.

Usage constraints This register can be modified only with prior written permission from ARM.

Configurations   Available in all configurations.

Attributes   See *Table 3-7 HN-I register summary* on page 3-90.

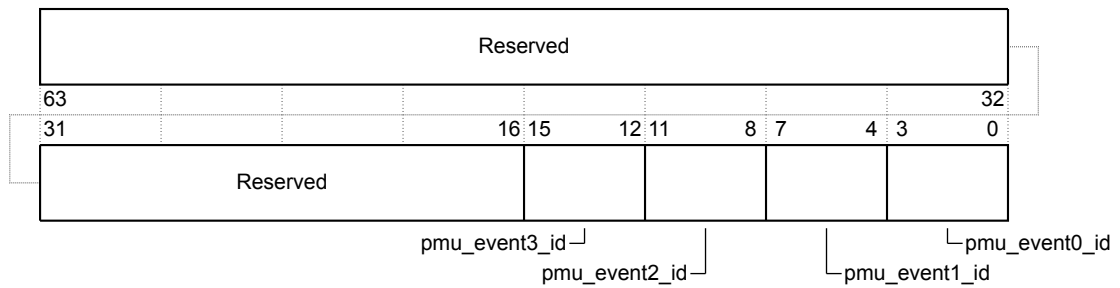The following figure shows the sa_aux_ctl register bit assignments.
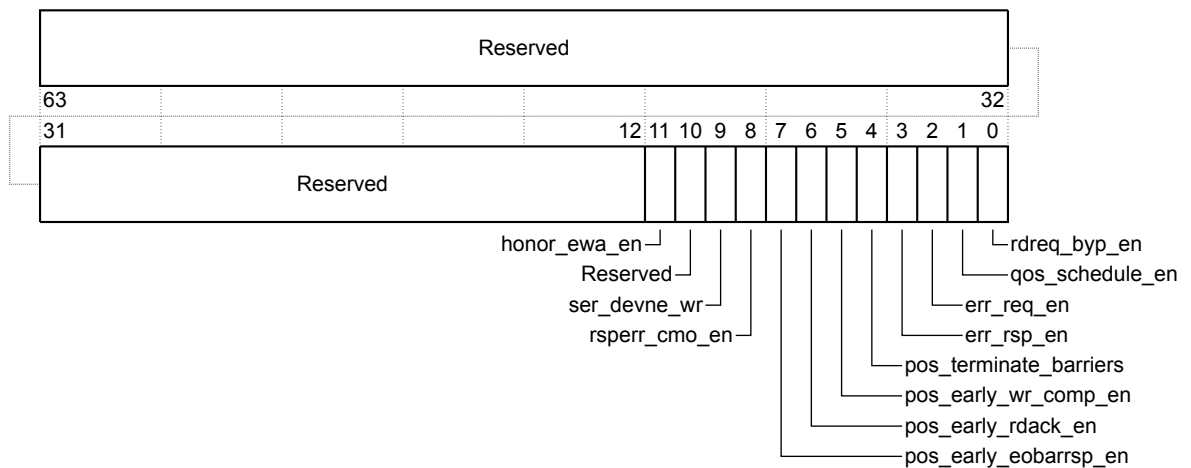
Figure 3-93 sa_aux_ctl register bit assignments

The following table shows the sa_aux_ctl register bit assignments.

Table 3-107 sa_aux_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11]	honor_ewa_en	RW	0	Propagate **BRESP** to the requesting RN, for non-posted writes.
[10]	-	WI	1	Reserved
[9]	ser_devne_wr	RW	0	Serialize Device-nGnRnE writes.
[8]	rsperr_cmo_en	RW	0	Enable sending *Non-data Error* (NDERR) response on CMO. Applies to all requests with Comp-only response semantics.

Table 3-107 sa_aux_ctl register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[7]	pos_early_eobarrsp_en	RW	1	Enable sending early completion response for EOBarrier from HN-I. Used to improve EOBarrier performance. Violates the CHI GO definition when the hni_pos_en bit in the pos_control register is 0.
[6]	pos_early_rdack_en	RW	1	Enable sending early read receipts from HN-I. Used to improve ordered read performance. Violates the CHI GO definition when the hni_pos_en bit in the pos_control register is 0.
[5]	pos_early_wr_comp_en	RW	1	Enable early write completions for all writes that allow early acknowledgment. Used to improve write performance. Violates the CHI GO definition when the hni_pos_en bit in the pos_control register is 0.
[4]	pos_terminate_barriers	RW	1	Enable termination of barriers before AMBA interface. Used when no downstream barrier capability exists (AXI4) or is required.
[3]	err_rsp_en	RW	0	Set to 1, to enable signaling an error to the MN on response.
[2]	err_req_en	RW	1	Set to 1, to enable signaling an error to the MN on request.
[1]	qos_schedule_en	RW	1	Set to 1, to enable QoS based scheduling of the AMBA requests.
[0]	rdreq_byp_en	RW	1	Set to 1, to enable read bypass path.

HN-I Identification register

The oly_hni_oly_id register is at offset `0xFF00`. Its characteristics are:

Purpose   Contains the component identification information.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-7 HN-I register summary* on page 3-90.

The following figure shows the oly_hni_oly_id register bit assignments.
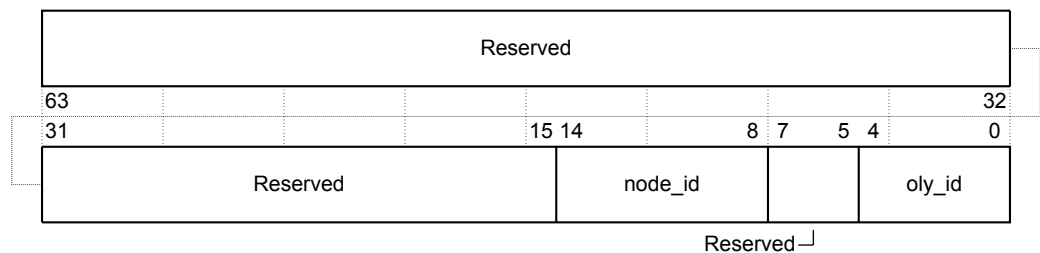

Figure 3-94 oly_hni_oly_id register bit assignments

The following table shows the oly_hni_oly_id register bit assignments.

Table 3-108 oly_hni_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	0x0	The node ID of the HN-I
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x5	Indicates that this node is an HN-I

Related references

3.3.5 Debug event module register descriptions

This section lists the DEM registers.

Active DSM register

The active_dsm register is at offset `0x0000`. Its characteristics are:

Purpose   Specifies the IDs of the XP containing the watchpoints that are driving the respective bits of the DTBus.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the active_dsm register bit assignments.

dsm_id7	dsm_id6	dsm_id5	dsm_id4
63 56	55 48	47 40	39 32
31 24	23 16	15 8	7 0
dsm_id3	dsm_id2	dsm_id1	dsm_id0

Figure 3-95 active_dsm register bit assignments

The following table shows the active_dsm register bit assignments.

Table 3-109 active_dsm register bit assignments

Bits	Name	Access	Reset value	Function
[63:56]	dsm_id7	RW	0x0	XP ID of XP driving DTBus[7]
[55:48]	dsm_id6	RW	0x0	XP ID of XP driving DTBus[6]
[47:40]	dsm_id5	RW	0x0	XP ID of XP driving DTBus[5]
[39:32]	dsm_id4	RW	0x0	XP ID of XP driving DTBus[4]
[31:24]	dsm_id3	RW	0x0	XP ID of XP driving DTBus[3]
[23:16]	dsm_id2	RW	0x0	XP ID of XP driving DTBus[2]
[15:8]	dsm_id1	RW	0x0	XP ID of XP driving DTBus[1]
[7:0]	dsm_id0	RW	0x0	XP ID of XP driving DTBus[0]

Trigger Control register

The trigger_ctl register is at offset 0x0008. Its characteristics are:

Purpose	Controls the trigger operation.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the trigger_ctl register bit assignments.
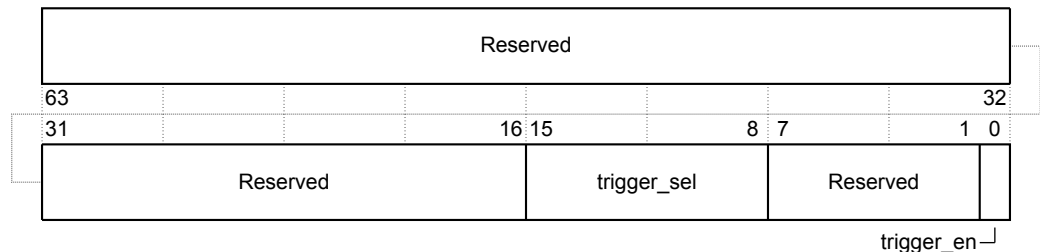
Figure 3-96 trigger_ctl register bit assignments

The following table shows the trigger_ctl register bit assignments.

Table 3-110 trigger_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:16]	-	RAZ/WI	0x0	Reserved
[15:8]	trigger_sel	RW	0x0	Enable DTBus bits to be used for **DBGWATCHTRIG** assertion
[7:1]	-	RAZ/WI	0x0	Reserved
[0]	trigger_en	RW	0	Enables **DBGWATCHTRIG**

Trigger Status register

The trigger_status register is at offset 0x0010. Its characteristics are:

Purpose	Indicates the trigger status.
Usage constraints	There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the trigger_status register bit assignments.
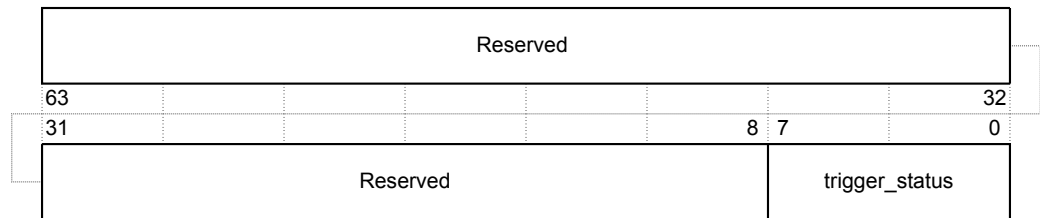
Figure 3-97 trigger_status register bit assignments

The following table shows the trigger_status register bit assignments.

Table 3-111 trigger_status register bit assignments

Bits	Name	Access	Reset value	Function
[63:8]	-	RAZ/WI	0x0	Reserved
[7:0]	trigger_status	RO	0x0	Indicates which DT bus bits caused the **DBGWATCHTRIGREQ** assertion

Trigger Status Clear register

The trigger_status_clr register is at offset `0x0018`. Its characteristics are:

Purpose   Clears the trigger status.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the trigger_status_clr register bit assignments.
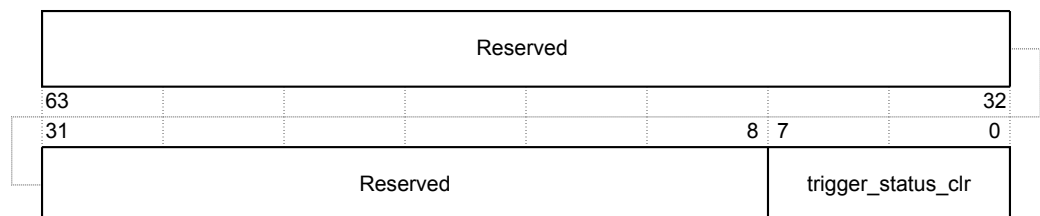
Figure 3-98 trigger_status_clr register bit assignments

The following table shows the trigger_status_clr register bit assignments.

Table 3-112 trigger_status_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63:8]	-	RAZ/WI	0x0	Reserved
[7:0]	trigger_status_clr	WO	0x0	Write 1 to clear corresponding bit of trigger_status register

Related references

Trigger Status register on page 3-173.

Timer Value register

The timer_val register is at offset `0x0020`. Its characteristics are:

Purpose   Controls the timer value.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the timer_val register bit assignments.

Figure 3-99 timer_val register bit assignments

The following table shows the timer_val register bit assignments.

Table 3-113 timer_val register bit assignments

Bits	Name	Access	Reset value	Function
[63:16]	-	RAZ/WI	0x0	Reserved
[15:0]	timer_val	RW	0x0	Number of cycles delay between debug event from DT bus and **DBGWATCHTRIG** assertion

Debug and Trace Control register, dt_ctl

The dt_ctl register is at offset `0x0028`. Its characteristics are:

Purpose   Controls the debug and trace features.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the dt_ctl register bit assignments.

Figure 3-100 dt_ctl register bit assignments

The following table shows the dt_ctl register bit assignments.

Table 3-114 dt_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:1]	-	RAZ/WI	0x0	Reserved
[0]	dt_en	RW	0	Enables the debug and trace features

Debug Identification register

The dbg_id register is at offset `0x0080`. Its characteristics are:

Purpose	Indicates the debug features.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the dbg_id register bit assignments.
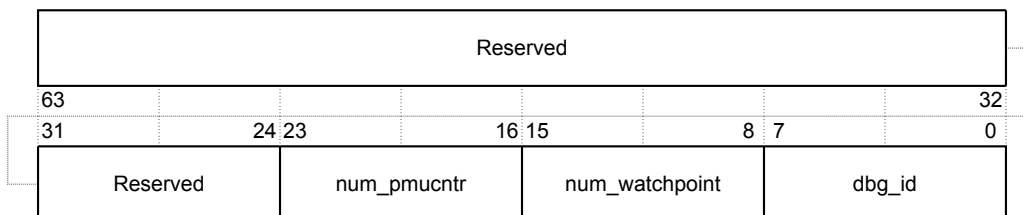
Figure 3-101 dbg_id register bit assignments

The following table shows the dbg_id register bit assignments.

Table 3-115 dbg_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:24]	-	RAZ/WI	0x0	Reserved
[23:16]	num_pmucntr	RO	0x09	Number of PMU counters
[15:8]	num_watchpoint	RO	0x16	Number of watchpoints
[7:0]	dbg_id	RO	0x00 for a 4-bit RSDVC configuration, 0x02 for an 8-bit RSVDC configuration	Debug ID register

PMU Event Counter 0 register

The pmevcnt0 register is at offset `0x0100`. Its characteristics are:

Purpose	Indicates the value of PMU event counter 0.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt0 register bit assignments.

Figure 3-102 pmevcnt0 register bit assignments

The following table shows the pmevcnt0 register bit assignments.

Table 3-116 pmevcnt0 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt0	RW	0x0	Value of PMU event counter 0

PMU Event Counter 1 register

The pmevcnt1 register is at offset `0x0108`. Its characteristics are:

Purpose	Indicates the value of PMU event counter 1.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt1 register bit assignments.

Figure 3-103 pmevcnt1 register bit assignments

The following table shows the pmevcnt1 register bit assignments.

Table 3-117 pmevcnt1 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt1	RW	0x0	Value of PMU event counter 1

PMU Event Counter 2 register

The pmevcnt2 register is at offset `0x0110`. Its characteristics are:

Purpose	Indicates the value of PMU event counter 2.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt2 register bit assignments.

Figure 3-104 pmevcnt2 register bit assignments

The following table shows the pmevcnt2 register bit assignments.

Table 3-118 pmevcnt2 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt2	RW	0x0	Value of PMU event counter 2

PMU Event Counter 3 register

The pmevcnt3 register is at offset `0x0118`. Its characteristics are:

Purpose　　　　　　Indicates the value of PMU event counter 3.

Usage constraints　There are no usage constraints.

Configurations　　Available in all configurations.

Attributes　　　　See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt3 register bit assignments.

```
┌──────────────────────────────────────────────────────┐
│                      Reserved                          │
├──────────────────────────────────────────────────────┤
 63                                                    32
 31                                                     0
┌──────────────────────────────────────────────────────┐
│                      pmevcnt3                          │
└──────────────────────────────────────────────────────┘
```

Figure 3-105 pmevcnt3 register bit assignments

The following table shows the pmevcnt3 register bit assignments.

Table 3-119 pmevcnt3 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt3	RW	0x0	Value of PMU event counter 3

PMU Event Counter 4 register

The pmevcnt4 register is at offset `0x0120`. Its characteristics are:

Purpose　　　　　　Indicates the value of PMU event counter 4.

Usage constraints　There are no usage constraints.

Configurations　　Available in all configurations.

Attributes　　　　See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt4 register bit assignments.

```
┌──────────────────────────────────────────────────────┐
│                      Reserved                          │
├──────────────────────────────────────────────────────┤
 63                                                    32
 31                                                     0
┌──────────────────────────────────────────────────────┐
│                      pmevcnt4                          │
└──────────────────────────────────────────────────────┘
```

Figure 3-106 pmevcnt4 register bit assignments

The following table shows the pmevcnt4 register bit assignments.

Table 3-120 pmevcnt4 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt4	RW	0x0	Value of PMU event counter 4

PMU Event Counter 5 register

The pmevcnt5 register is at offset `0x0128`. Its characteristics are:

Purpose   Indicates the value of PMU event counter 5.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt5 register bit assignments.

Reserved
63   32
31   0
pmevcnt5

Figure 3-107 pmevcnt5 register bit assignments

The following table shows the pmevcnt5 register bit assignments.

Table 3-121 pmevcnt5 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt5	RW	0x0	Value of PMU event counter 5

PMU Event Counter 6 register

The pmevcnt6 register is at offset `0x0130`. Its characteristics are:

Purpose   Indicates the value of PMU event counter 6.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt6 register bit assignments.

Reserved
63   32
31   0
pmevcnt6

Figure 3-108 pmevcnt6 register bit assignments

The following table shows the pmevcnt6 register bit assignments.

Table 3-122 pmevcnt6 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt6	RW	0x0	Value of PMU event counter 6

PMU Event Counter 7 register

The pmevcnt7 register is at offset `0x0138`. Its characteristics are:

Purpose   Indicates the value of PMU event counter 7.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcnt7 register bit assignments.

```
┌─────────────────────────────────────────────────────────┐
│                        Reserved                          │
└─────────────────────────────────────────────────────────┘
 63                                                       32
 31                                                        0
┌─────────────────────────────────────────────────────────┐
│                        pmevcnt7                          │
└─────────────────────────────────────────────────────────┘
```

Figure 3-109 pmevcnt7 register bit assignments

The following table shows the pmevcnt7 register bit assignments.

Table 3-123 pmevcnt7 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcnt7	RW	0x0	Value of PMU event counter 7

PMU Cycle Counter register

The pmccntr register is at offset `0x0140`. Its characteristics are:

Purpose   Controls the PMU cycle counter.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmccntr register bit assignments.

```
┌──────────────────────────────────────────┬──────────────┐
│                  Reserved                 │   pmccntr    │
└──────────────────────────────────────────┴──────────────┘
 63                                       40 39           32
 31                                                        0
┌─────────────────────────────────────────────────────────┐
│                        pmccntr                           │
└─────────────────────────────────────────────────────────┘
```

Figure 3-110 pmccntr register bit assignments

The following table shows the pmccntr register bit assignments.

Table 3-124 pmccntr register bit assignments

Bits	Name	Access	Reset value	Function
[63:40]	-	RAZ/WI	0x0	Reserved
[39:0]	pmccntr	RW	0x0	PMU cycle counter

PMU Event Counter Shadow 0 register

The pmevcntsr0 register is at offset `0x0150`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 0.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

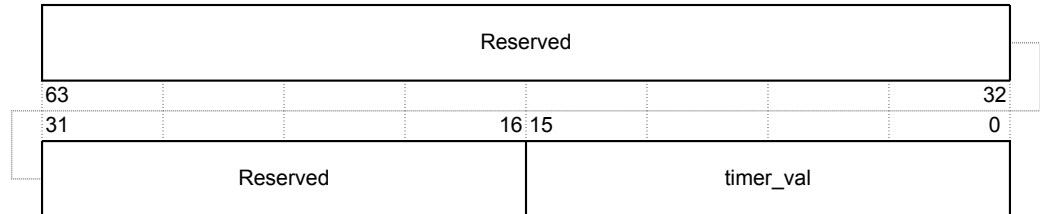The following figure shows the pmevcntsr0 register bit assignments.

Figure 3-111 pmevcntsr0 register bit assignments

The following table shows the pmevcntsr0 register bit assignments.

Table 3-125 pmevcntsr0 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr0	RW	0x0	Value of PMU event counter 0

PMU Event Counter Shadow 1 register

The pmevcntsr1 register is at offset `0x0158`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 1.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

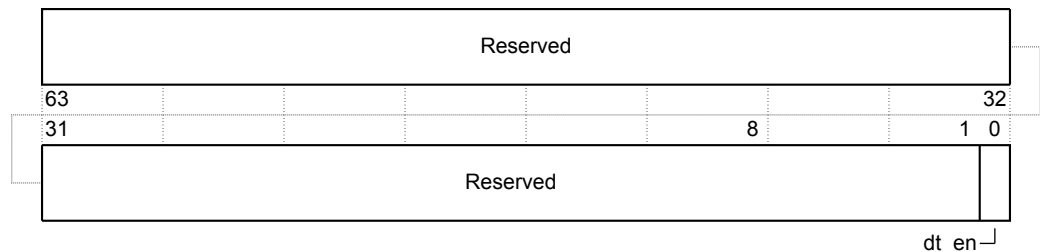The following figure shows the pmevcntsr1 register bit assignments.

Figure 3-112 pmevcntsr1 register bit assignments

The following table shows the pmevcntsr1 register bit assignments.

Table 3-126 pmevcntsr1 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr1	RW	0x0	Value of PMU event counter 1

PMU Event Counter Shadow 2 register

The pmevcntsr2 register is at offset `0x0160`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 2.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcntsr2 register bit assignments.

Reserved	
63	32
31	0
pmevcntsr2	

Figure 3-113 pmevcntsr2 register bit assignments

The following table shows the pmevcntsr2 register bit assignments.

Table 3-127 pmevcntsr2 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr2	RW	0x0	Value of PMU event counter 2

PMU Event Counter Shadow 3 register

The pmevcntsr3 register is at offset `0x0168`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 3.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcntsr3 register bit assignments.

Reserved	
63	32
31	0
pmevcntsr3	

Figure 3-114 pmevcntsr3 register bit assignments

The following table shows the pmevcntsr3 register bit assignments.

Table 3-128 pmevcntsr3 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr3	RW	0x0	Value of PMU event counter 3

PMU Event Counter Shadow 4 register

The pmevcntsr4 register is at offset `0x0170`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 4.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

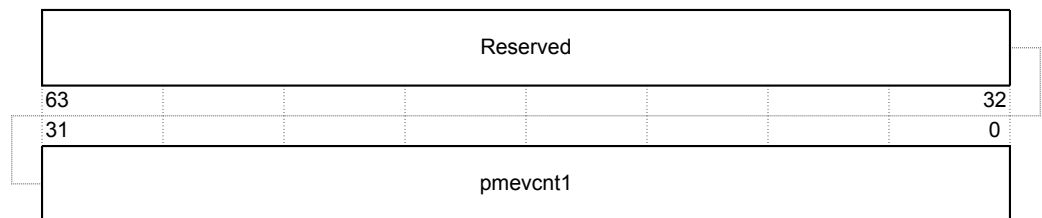The following figure shows the pmevcntsr4 register bit assignments.

Figure 3-115 pmevcntsr4 register bit assignments

The following table shows the pmevcntsr4 register bit assignments.

Table 3-129 pmevcntsr4 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr4	RW	0x0	Value of PMU event counter 4

PMU Event Counter Shadow 5 register

The pmevcntsr5 register is at offset `0x0178`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 5.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

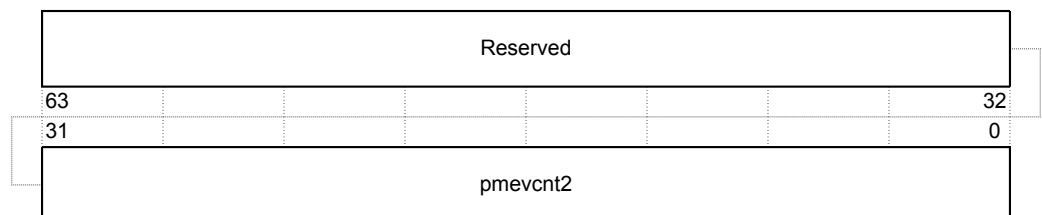The following figure shows the pmevcntsr5 register bit assignments.

Figure 3-116 pmevcntsr5 register bit assignments

The following table shows the pmevcntsr5 register bit assignments.

Table 3-130 pmevcntsr5 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr5	RW	0x0	Value of PMU event counter 5

PMU Event Counter Shadow 6 register

The pmevcntsr6 register is at offset `0x0180`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 6.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcntsr6 register bit assignments.

Reserved
63   32

31   0
pmevcntsr6

Figure 3-117 pmevcntsr6 register bit assignments

The following table shows the pmevcntsr6 register bit assignments.
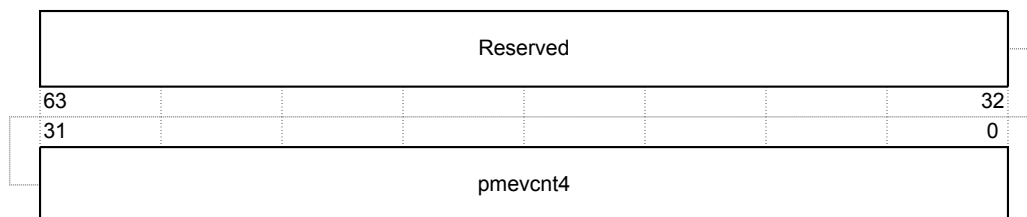
Table 3-131 pmevcntsr6 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr6	RW	0x0	Value of PMU event counter 6

PMU Event Counter Shadow 7 register

The pmevcntsr7 register is at offset `0x0188`. Its characteristics are:

Purpose   Shadow register that indicates the value of PMU event counter 7.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmevcntsr7 register bit assignments.

Reserved
63   32

31   0
pmevcntsr7

Figure 3-118 pmevcntsr7 register bit assignments

The following table shows the pmevcntsr7 register bit assignments.
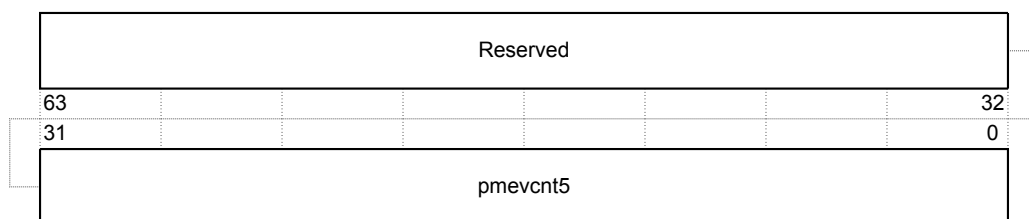
Table 3-132 pmevcntsr7 register bit assignments

Bits	Name	Access	Reset value	Function
[63:32]	-	RAZ/WI	0x0	Reserved
[31:0]	pmevcntsr7	RW	0x0	Value of PMU event counter 7

PMU Cycle Counter Shadow register

The pmccntrsr register is at offset `0x0190`. Its characteristics are:

Purpose   Shadow register that controls the PMU cycle counter.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmccntrsr register bit assignments.

Reserved						pmccntrsr
63					40	39 32

31	0
pmccntrsr	
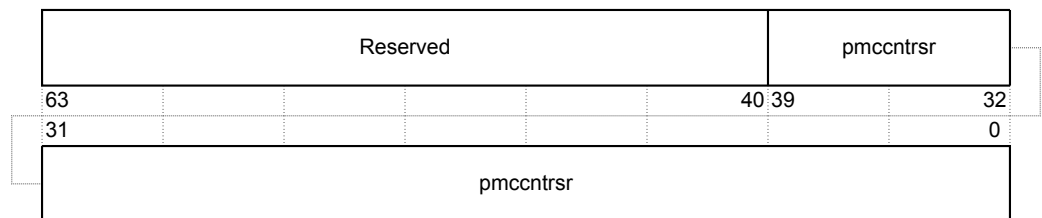
Figure 3-119 pmccntrsr register bit assignments

The following table shows the pmccntrsr register bit assignments.
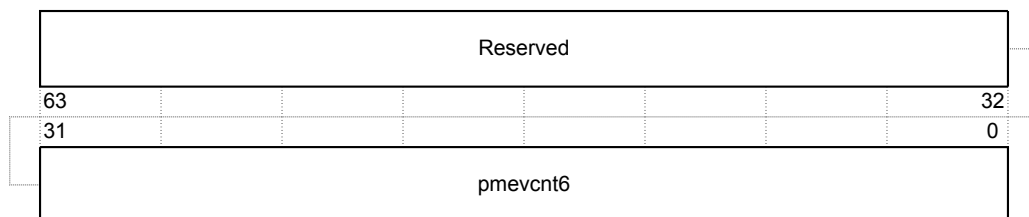
Table 3-133 pmccntrsr register bit assignments

Bits	Name	Access	Reset value	Function
[63:40]	-	RAZ/WI	0x0	Reserved
[39:0]	pmccntrsr	RW	0x0	PMU cycle counter

PMU Overflow Status register

The pmovsr register is at offset `0x0198`. Its characteristics are:

Purpose   Indicates the PMU overflow status.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmovsr register bit assignments.
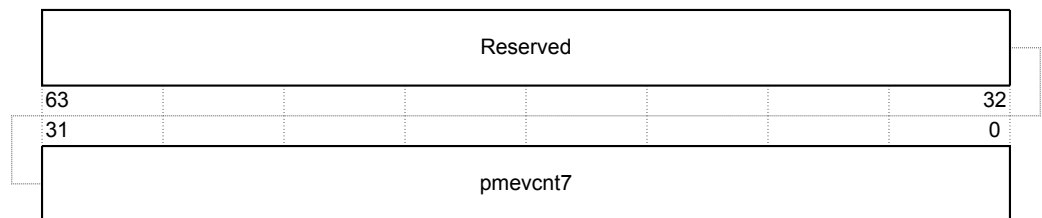
Reserved		
63		32

Reserved		pmovsr
31	9 8	0
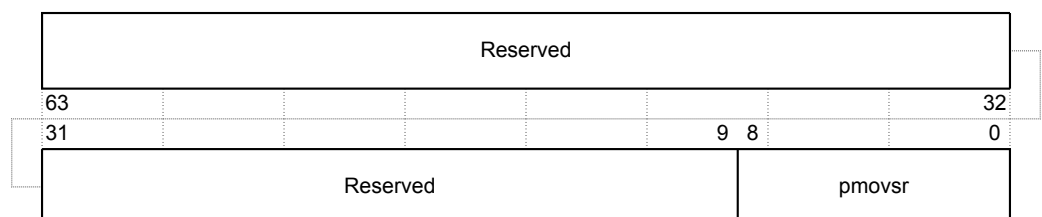
Figure 3-120 pmovsr register bit assignments

The following table shows the pmovsr register bit assignments.

Table 3-134 pmovsr register bit assignments

Bits	Name	Access	Reset value	Function
[63:9]	-	RAZ/WI	0x0	Reserved
[8:0]	pmovsr	RO	0x0	PMU overflow status: **Bit[8]** Overflow from cycle counter. **Bits[7:0]** Overflow from counters 7-0.

PMU Overflow Status Clear register

The pmovsr_clr register is at offset `0x01A0`. Its characteristics are:

Purpose	Clears the PMU overflow.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmovsr_clr register bit assignments.
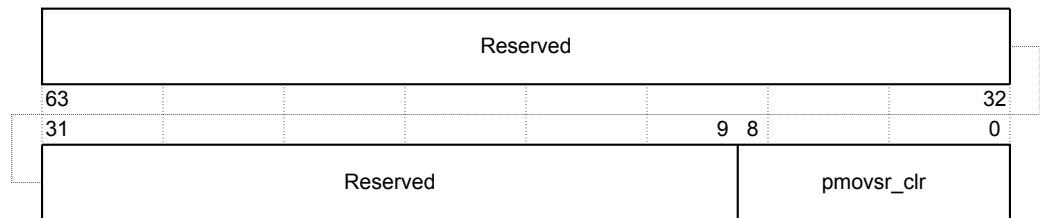
Figure 3-121 pmovsr_clr register bit assignments

The following table shows the pmovsr_clr register bit assignments.

Table 3-135 pmovsr_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63:9]	-	RAZ/WI	0x0	Reserved
[8:0]	pmovsr_clr	WO	0x0	Write 1 to clear the corresponding bit of the pmovsr register

Related references
PMU Overflow Status register on page 3-185.

PMU Control register

The pmcr register is at offset `0x01A8`. Its characteristics are:

Purpose	Controls the PMU and its features.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmcr register bit assignments.
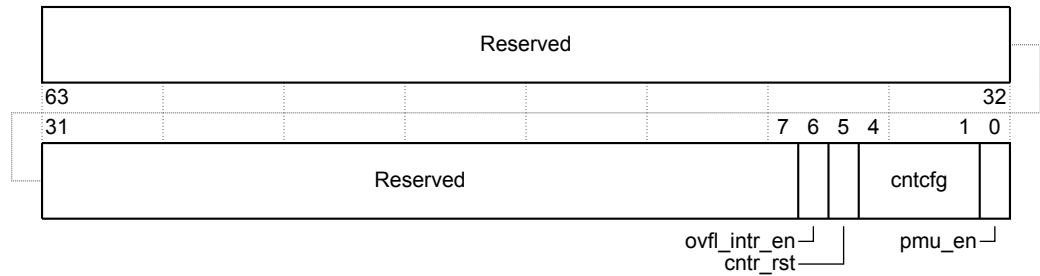
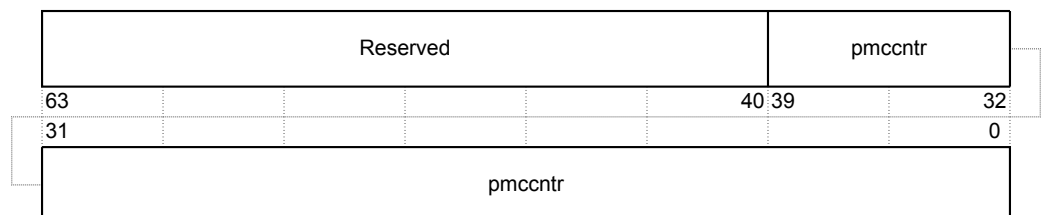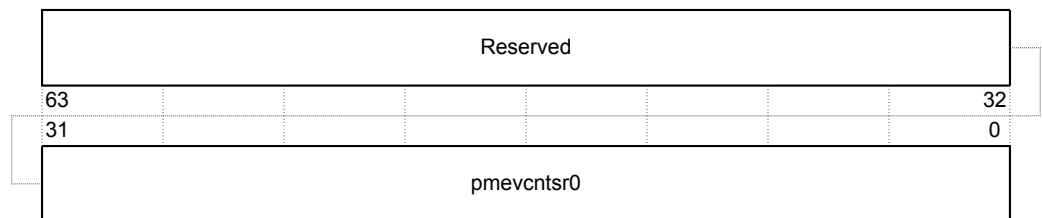**Figure 3-122 pmcr register bit assignments**

The following table shows the pmcr register bit assignments.

Table 3-136 pmcr register bit assignments

Bits	Name	Access	Reset value	Function
[63:7]	-	RAZ/WI	0x0	Reserved
[6]	ovfl_intr_en	RW	0	Enables assertion of **INTREQ** on overflow of PMU counters.
[5]	cntr_rst	RW	0	Enables clearing of live counters on assertion of the pmsr_req bit in the pmsr_req register or **PMUSNAPSHOTREQ**.
[4:1]	cntcfg	RW	0x0	Control to group the pair of adjacent 32-bit registers into one 64-bit register. • 0 = No pairing. • 1 = Pairing of adjacent PMU counters. • cntcfg[0] for pmevcnt0/pmevcnt1 • cntcfg[1] for pmevcnt2/pmevcnt3 • cntcfg[2] for pmevcnt4/pmevcnt5 • cntcfg[3] for pmevcnt6/pmevcnt7
[0]	pmu_en	RW	0	Enables PMU features.

Related references

PMU Snapshot Request register on page 3-188.

PMU Status register

The pmsr register is at offset 0x01B0. Its characteristics are:

Purpose   Indicates the PMU snapshot status.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmsr register bit assignments.
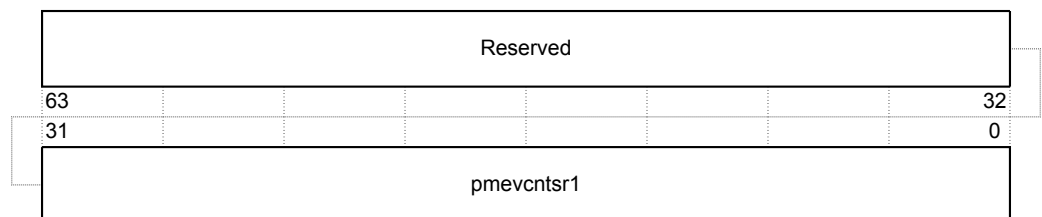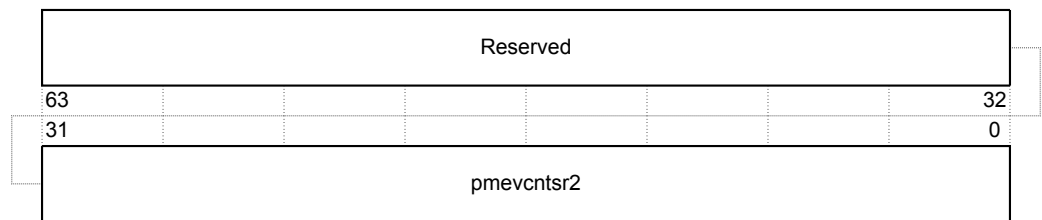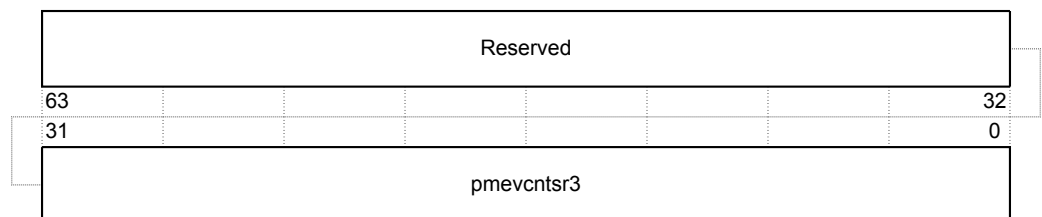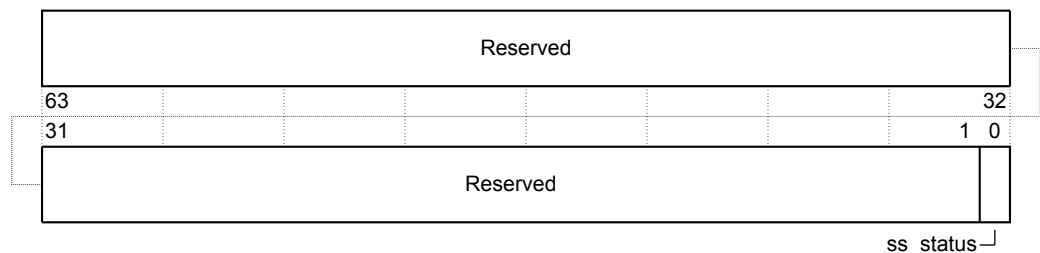

Figure 3-123 pmsr register bit assignments

The following table shows the pmsr register bit assignments.

Table 3-137 pmsr register bit assignments

Bits	Name	Access	Reset value	Function
[63:1]	-	RAZ/WI	0x0	Reserved
[0]	ss_status	RO	0	PMU snapshot status

PMU Snapshot Request register

The pmsr_req register is at offset `0x01B8`. Its characteristics are:

Purpose	Requests a PMU snapshot.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the pmsr_req register bit assignments.
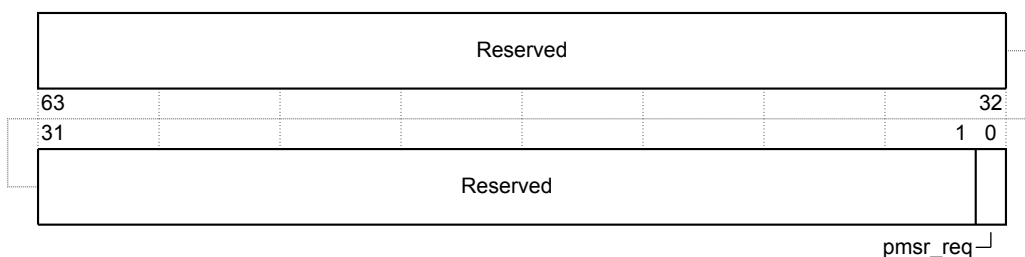
Figure 3-124 pmsr_req register bit assignments

The following table shows the pmsr_req register bit assignments.

Table 3-138 pmsr_req register bit assignments

Bits	Name	Access	Reset value	Function
[63:1]	-	RAZ/WI	0x0	Reserved
[0]	pmsr_req	WO	0	Write 1 to request a PMU snapshot

PMU Snapshot Status Clear register

The pmsr_clr register is at offset `0x01C0`. Its characteristics are:

Purpose	Clears the PMU snapshot status.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

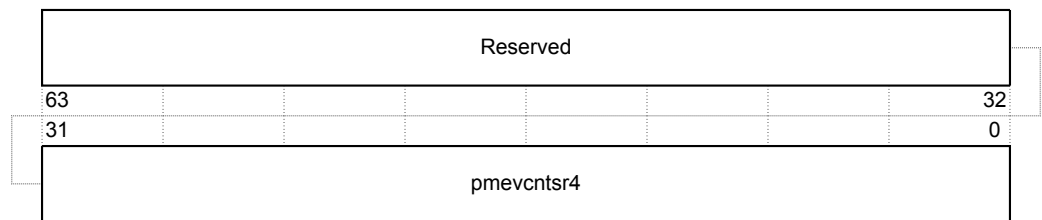The following figure shows the pmsr_clr register bit assignments.
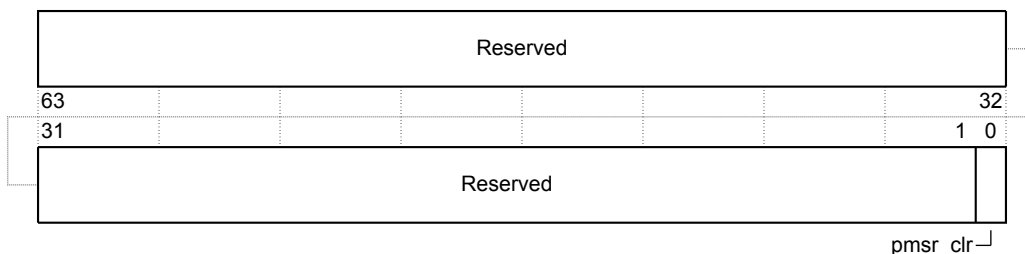
Figure 3-125 pmsr_clr register bit assignments

The following table shows the pmsr_clr register bit assignments.

Table 3-139 pmsr_clr register bit assignments

Bits	Name	Access	Reset value	Function
[63:1]	-	RAZ/WI	0x0	Reserved
[0]	pmsr_clr	WO	0	Write 1 to clear the PMU snapshot status

Related references
PMU Status register on page 3-187.

Debug and Trace Identification register

The oly_mn_dt_oly_id register is at offset 0xFF00. Its characteristics are:

Purpose	Contains the component identification information.
Usage constraints	There are no usage constraints.
Configurations	Available in all configurations.
Attributes	See *Table 3-8 Debug event module register summary* on page 3-90.

The following figure shows the oly_mn_dt_oly_id register bit assignments.
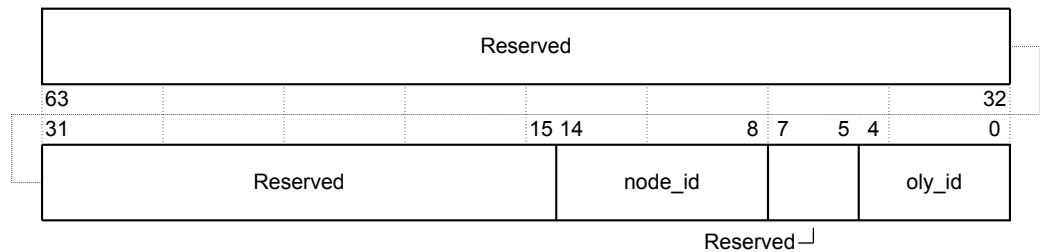
Figure 3-126 oly_mn_dt_oly_id register bit assignments

The following table shows the oly_mn_dt_oly_id register bit assignments.

Table 3-140 oly_mn_dt_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	0x0	The node ID of the DT
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x2	Indicates that this node is a DT

Related references
3.1.2 Node type IDs on page 3-85.

3.3.6 RN-I bridge register descriptions

Lists the RN-I registers.

Port S0 Control register, RN-I

The s0_port_control register is at offset `0x0008`. Its characteristics are:

Purpose	Controls the port S0 AXI/ACE slave interface.
Usage constraints	Only accessible by Secure accesses. Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s0_port_control register bit assignments.

Figure 3-127 s0_port_control register bit assignments

The following table shows the s0_port_control register bit assignments.

Table 3-141 s0_port_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:4]	s0_lpid_mask	RW	0x0	S0 port LPID mask. Specifies the **AXID** bits to be reflected in the least significant bit of the LPID: **LPID[0]**　　　　BitwiseOR (LPID mask AND **AXID**). **LPID[2:1]**　　　Port ID[1:0].
[3:2]	-	RW	0x0	Reserved
[1:0]	-	RAZ/WI	0x0	Reserved

Port S0 QoS Control register, RN-I

The s0_qos_control register is at offset `0x0010`. Its characteristics are:

Purpose　　　　　　Controls the QoS settings for the port S0 AXI/ACE slave interface.

Usage constraints　　Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations　　　Available in all configurations.

Attributes　　　　　See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s0_qos_control register bit assignments.

Figure 3-128 s0_qos_control register bit assignments

The following table shows the s0_qos_control register bit assignments.

Table 3-142 s0_qos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:24]	-	RAZ/WI	0x0	Reserved
[23:20]	s0_ar_qos_override	RW	0x0	S0 port AR QoS override value.
[19:16]	s0_aw_qos_override	RW	0x0	S0 port AW QoS override value.
[15:8]	-	RAZ/WI	0x0	Reserved

Table 3-142 s0_qos_control register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[7]	s0_ar_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during read transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[6]	s0_aw_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during write transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[5]	s0_ar_reg_mode	RW	0	Configures the mode of the QoS regulator for read transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[4]	s0_aw_reg_mode	RW	0	Configures the mode of the QoS regulator for write transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[3]	s0_ar_qos_override_en	RW	0	S0 port AR QoS override enable. When set, this bit enables the QoS value on inbound AR transactions to be overridden.
[2]	s0_aw_qos_override_en	RW	0	S0 port AW QoS override enable. When set, this bit enables the QoS value on inbound AW transactions to be overridden.
[1]	s0_ar_lat_en	RW	0	S0 port AR QoS regulation enable. When set, this bit enables AR regulation.
[0]	s0_aw_lat_en	RW	0	S0 port AW QoS regulation enable. When set, this bit enables AW regulation.

Port S0 QoS Latency Target register, RN-I

The s0_qos_lat_tgt register is at offset `0x0018`. Its characteristics are:

Purpose Controls the QoS target latency, in cycles, for the regulation of reads and writes for port S0. A value of 0 corresponds to no regulation.

Usage constraints Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations Available in all configurations.

Attributes See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s0_qos_lat_tgt register bit assignments.

Figure 3-129 s0_qos_lat_tgt register bit assignments

The following table shows the s0_qos_lat_tgt register bit assignments.

Table 3-143 s0_qos_lat_tgt register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:16]	s0_ar_lat_tgt	RW	0x0	S0 AR channel target latency
[15:12]	-	RAZ/WI	0x0	Reserved
[11:0]	s0_aw_lat_tgt	RW	0x0	S0 AW channel target latency

Port S0 QoS Latency Scale register, RN-I

The s0_qos_lat_scale register is at offset 0x0020. Its characteristics are:

Purpose Controls the QoS target latency scale factor for reads and writes for port S0. It is coded for powers of 2 in the range 2^{-5} to 2^{-12}.

Usage constraints Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations Available in all configurations.

Attributes See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s0_qos_lat_scale register bit assignments.

Figure 3-130 s0_qos_lat_scale register bit assignments

The following table shows the s0_qos_lat_scale register bit assignments.

Table 3-144 s0_qos_lat_scale register bit assignments

Bits	Name	Access	Reset value	Function
[63:11]	-	RAZ/WI	0x0	Reserved
[10:8]	s0_ar_lat_scale	RW	0x0	S0 AR QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}
[7:3]	-	RAZ/WI	0x0	Reserved
[2:0]	s0_aw_lat_scale	RW	0x0	S0 AW QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Port S0 QoS Latency Range register, RN-I

The s0_qos_lat_range register is at offset `0x0028`. Its characteristics are:

Purpose	Controls the QoS minimum and maximum values generated by the QoS latency regulator for reads and writes for port S0.
Usage constraints	Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s0_qos_lat_range register bit assignments.

Figure 3-131 s0_qos_lat_range register bit assignments

The following table shows the s0_qos_lat_range register bit assignments.

Table 3-145 s0_qos_lat_range register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:24]	s0_ar_lat_max_qos	RW	0x0	S0 AR QoS maximum value
[23:20]	-	RAZ/WI	0x0	Reserved
[19:16]	s0_ar_lat_min_qos	RW	0x0	S0 AR QoS minimum value
[15:12]	-	RAZ/WI	0x0	Reserved
[11:8]	s0_aw_lat_max_qos	RW	0x0	S0 AW QoS maximum value
[7:4]	-	RAZ/WI	0x0	Reserved
[3:0]	s0_aw_lat_min_qos	RW	0x0	S0 AW QoS minimum value

Port S1 Control register, RN-I

The s1_port_control register is at offset `0x0108`. Its characteristics are:

Purpose	Controls the port S1 AXI/ACE slave interface.
Usage constraints	Only accessible by Secure accesses. Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s1_port_control register bit assignments.

Figure 3-132 s1_port_control register bit assignments

The following table shows the s1_port_control register bit assignments.

Table 3-146 s1_port_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:4]	s1_lpid_mask	RW	0x0	S1 port LPID mask. Specifies the **AXID** bits to be reflected in the least significant bit of the LPID: **LPID[0]** BitwiseOR (LPID mask AND **AXID**). **LPID[2:1]** Port ID[1:0].
[3:2]	-	RW	0x0	Reserved
[1:0]	-	RAZ/WI	0x0	Reserved

Port S1 QoS Control register, RN-I

The s1_qos_control register is at offset 0x0110. Its characteristics are:

Purpose Controls the QoS settings for the port S1 AXI/ACE slave interface.

Usage constraints Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations Available in all configurations.

Attributes See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s1_qos_control register bit assignments.

Figure 3-133 s1_qos_control register bit assignments

The following table shows the s1_qos_control register bit assignments.

Table 3-147 s1_qos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:24]	-	RAZ/WI	`0x0`	Reserved
[23:20]	s1_ar_qos_override	RW	`0x0`	S1 port AR QoS override value.
[19:16]	s1_aw_qos_override	RW	`0x0`	S1 port AW QoS override value.
[15:8]	-	RAZ/WI	`0x0`	Reserved
[7]	s1_ar_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during read transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[6]	s1_aw_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during write transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[5]	s1_ar_reg_mode	RW	0	Configures the mode of the QoS regulator for read transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[4]	s1_aw_reg_mode	RW	0	Configures the mode of the QoS regulator for write transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[3]	s1_ar_qos_override_en	RW	0	S1 port AR QoS override enable. When set, this bit enables the QoS value on inbound AR transactions to be overridden.
[2]	s1_aw_qos_override_en	RW	0	S1 port AW QoS override enable. When set, this bit enables the QoS value on inbound AW transactions to be overridden.
[1]	s1_ar_lat_en	RW	0	S1 port AR QoS regulation enable. When set, this bit enables AR regulation.
[0]	s1_aw_lat_en	RW	0	S1 port AW QoS regulation enable. When set, this bit enables AW regulation.

Port S1 QoS Latency Target register, RN-I

The s1_qos_lat_tgt register is at offset `0x0118`. Its characteristics are:

Purpose Controls the QoS target latency, in cycles, for the regulation of reads and writes for port S1. A value of 0 corresponds to no regulation.

Usage constraints Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations Available in all configurations.

Attributes See *Table 3-9 RN-I bridge register summary* on page 3-91.

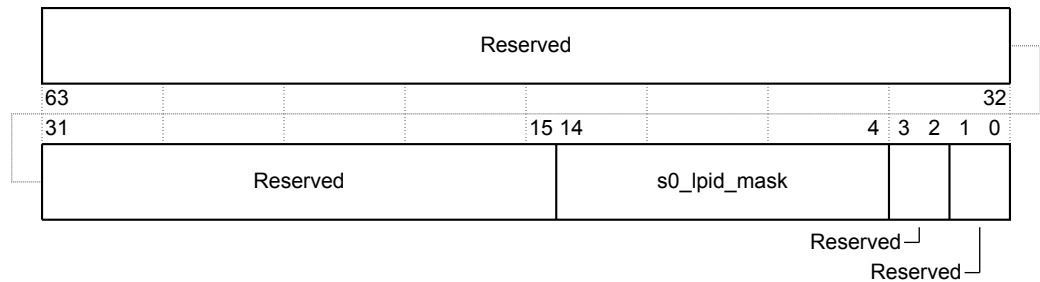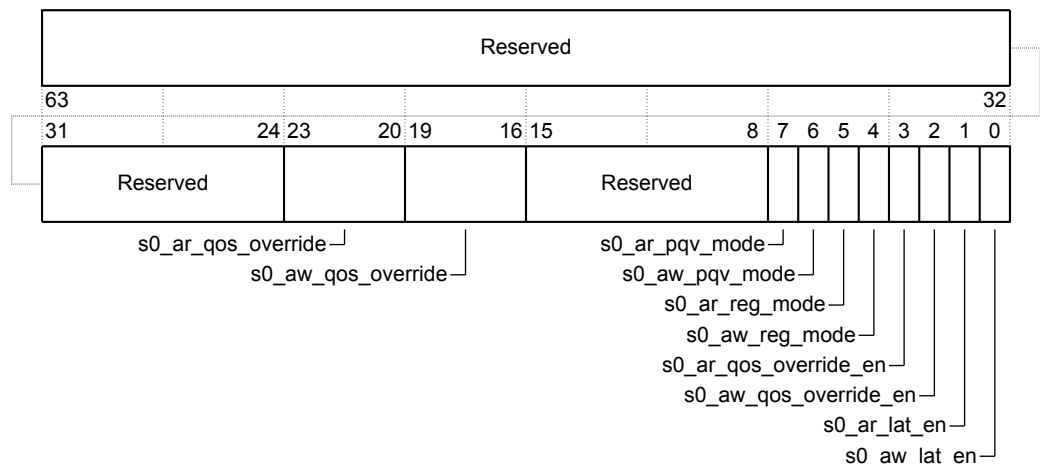The following figure shows the s1_qos_lat_tgt register bit assignments.

Figure 3-134 s1_qos_lat_tgt register bit assignments

The following table shows the s1_qos_lat_tgt register bit assignments.

Table 3-148 s1_qos_lat_tgt register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:16]	s1_ar_lat_tgt	RW	0x0	S1 AR channel target latency
[15:12]	-	RAZ/WI	0x0	Reserved
[11:0]	s1_aw_lat_tgt	RW	0x0	S1 AW channel target latency

Port S1 QoS Latency Scale register, RN-I

The s1_qos_lat_scale register is at offset `0x0120`. Its characteristics are:

Purpose   Controls the QoS target latency scale factor for reads and writes for port S1. It is coded for powers of 2 in the range 2^{-5} to 2^{-12}.

Usage constraints   Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s1_qos_lat_scale register bit assignments.

Figure 3-135 s1_qos_lat_scale register bit assignments

The following table shows the s1_qos_lat_scale register bit assignments.

Table 3-149 s1_qos_lat_scale register bit assignments

Bits	Name	Access	Reset value	Function
[63:11]	-	RAZ/WI	0x0	Reserved
[10:8]	s1_ar_lat_scale	RW	0x0	S1 AR QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Table 3-149 s1_qos_lat_scale register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[7:3]	-	RAZ/WI	0x0	Reserved
[2:0]	s1_aw_lat_scale	RW	0x0	S1 AW QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Port S1 QoS Latency Range register, RN-I

The s1_qos_lat_range register is at offset 0x0128. Its characteristics are:

Purpose   Controls the QoS minimum and maximum values generated by the QoS latency regulator for reads and writes for port S1.

Usage constraints   Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

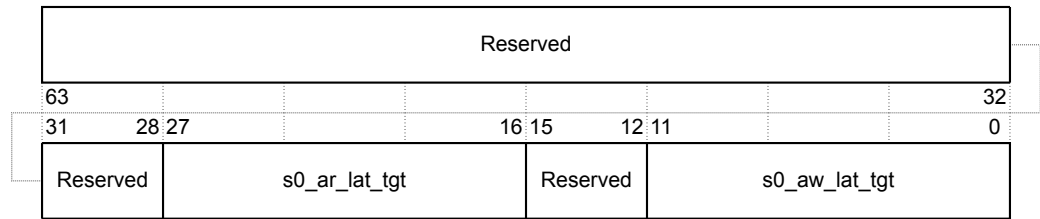The following figure shows the s1_qos_lat_range register bit assignments.

Figure 3-136 s1_qos_lat_range register bit assignments

The following table shows the s1_qos_lat_range register bit assignments.

Table 3-150 s1_qos_lat_range register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:24]	s1_ar_lat_max_qos	RW	0x0	S1 AR QoS maximum value
[23:20]	-	RAZ/WI	0x0	Reserved
[19:16]	s1_ar_lat_min_qos	RW	0x0	S1 AR QoS minimum value
[15:12]	-	RAZ/WI	0x0	Reserved
[11:8]	s1_aw_lat_max_qos	RW	0x0	S1 AW QoS maximum value
[7:4]	-	RAZ/WI	0x0	Reserved
[3:0]	s1_aw_lat_min_qos	RW	0x0	S1 AW QoS minimum value

Port S2 Control register, RN-I

The s2_port_control register is at offset 0x0208. Its characteristics are:

Purpose   Controls the port S2 AXI/ACE slave interface.

Usage constraints   Only accessible by Secure accesses. Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

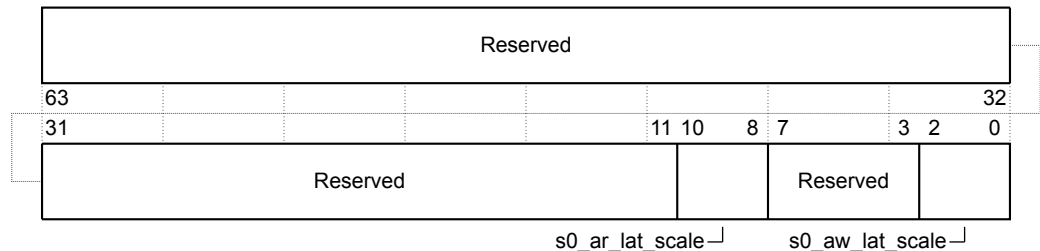The following figure shows the s2_port_control register bit assignments.

Figure 3-137 s2_port_control register bit assignments

The following table shows the s2_port_control register bit assignments.

Table 3-151 s2_port_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:4]	s2_lpid_mask	RW	0x0	S2 port LPID mask. Specifies the **AXID** bits to be reflected in the least significant bit of the LPID: **LPID[0]** BitwiseOR (LPID mask AND **AXID**). **LPID[2:1]** Port ID[1:0].
[3:2]	-	RW	0x0	Reserved
[1:0]	-	RAZ/WI	0x0	Reserved

Port S2 QoS Control register, RN-I

The s2_qos_control register is at offset `0x0210`. Its characteristics are:

Purpose   Controls the QoS settings for the port S2 AXI/ACE slave interface.

Usage constraints   Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

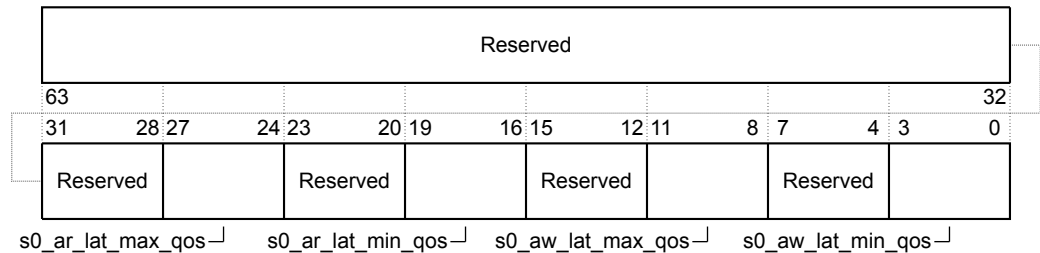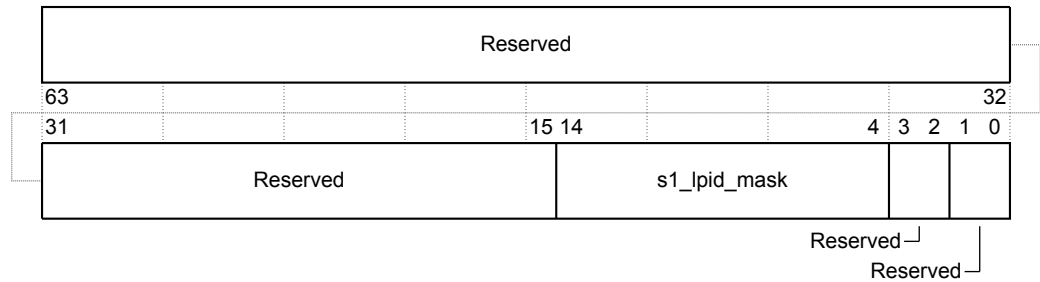The following figure shows the s2_qos_control register bit assignments.

Figure 3-138 s2_qos_control register bit assignments

The following table shows the s2_qos_control register bit assignments.

Table 3-152 s2_qos_control register bit assignments

Bits	Name	Access	Reset value	Function
[63:24]	-	RAZ/WI	`0x0`	Reserved
[23:20]	s2_ar_qos_override	RW	`0x0`	S2 port AR QoS override value.
[19:16]	s2_aw_qos_override	RW	`0x0`	S2 port AW QoS override value.
[15:8]	-	RAZ/WI	`0x0`	Reserved
[7]	s2_ar_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during read transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[6]	s2_aw_pqv_mode	RW	0	Configures the mode of the QoS regulator during period mode for bandwidth regulation during write transactions: **0** Normal mode. The QoS value is stable when the master is idle. **1** Quiesce high mode. The QoS value tends to the maximum value when the master is idle.
[5]	s2_ar_reg_mode	RW	0	Configures the mode of the QoS regulator for read transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[4]	s2_aw_reg_mode	RW	0	Configures the mode of the QoS regulator for write transactions: **0** Latency mode. **1** Period mode, for bandwidth regulation.
[3]	s2_ar_qos_override_en	RW	0	S2 port AR QoS override enable. When set, this bit enables the QoS value on inbound AR transactions to be overridden.
[2]	s2_aw_qos_override_en	RW	0	S2 port AW QoS override enable. When set, this bit enables the QoS value on inbound AW transactions to be overridden.

Table 3-152 s2_qos_control register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[1]	s2_ar_lat_en	RW	0	S2 port AR QoS regulation enable. When set, this bit enables AR regulation.
[0]	s2_aw_lat_en	RW	0	S2 port AW QoS regulation enable. When set, this bit enables AW regulation.

Port S2 QoS Latency Target register, RN-I

The s2_qos_lat_tgt register is at offset `0x0218`. Its characteristics are:

Purpose	Controls the QoS target latency, in cycles, for the regulation of reads and writes for port S2. A value of 0 corresponds to no regulation.
Usage constraints	Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-9 RN-I bridge register summary* on page 3-91.

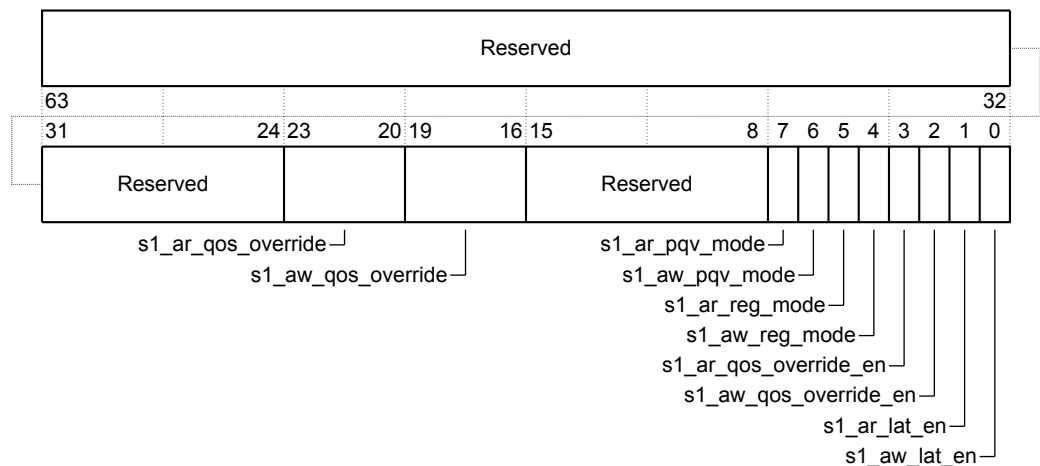The following figure shows the s2_qos_lat_tgt register bit assignments.

Figure 3-139 s2_qos_lat_tgt register bit assignments

The following table shows the s2_qos_lat_tgt register bit assignments.

Table 3-153 s2_qos_lat_tgt register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	`0x0`	Reserved
[27:16]	s2_ar_lat_tgt	RW	`0x0`	S2 AR channel target latency
[15:12]	-	RAZ/WI	`0x0`	Reserved
[11:0]	s2_aw_lat_tgt	RW	`0x0`	S2 AW channel target latency

Port S2 QoS Latency Scale register, RN-I

The s2_qos_lat_scale register is at offset `0x0220`. Its characteristics are:

Purpose	Controls the QoS target latency scale factor for reads and writes for port S1. It is coded for powers of 2 in the range 2^{-5} to 2^{-12}.
Usage constraints	Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.
Configurations	Available in all configurations.
Attributes	See *Table 3-9 RN-I bridge register summary* on page 3-91.

The following figure shows the s2_qos_lat_scale register bit assignments.

Figure 3-140 s2_qos_lat_scale register bit assignments

The following table shows the s2_qos_lat_scale register bit assignments.

Table 3-154 s2_qos_lat_scale register bit assignments

Bits	Name	Access	Reset value	Function
[63:11]	-	RAZ/WI	0x0	Reserved
[10:8]	s2_ar_lat_scale	RW	0x0	S2 AR QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}
[7:3]	-	RAZ/WI	0x0	Reserved
[2:0]	s2_aw_lat_scale	RW	0x0	S2 AW QoS scale factor, in powers of 2 in the range 2^{-5} to 2^{-12}

Port S2 QoS Latency Range register, RN-I

The s2_qos_lat_range register is at offset 0x0228. Its characteristics are:

Purpose   Controls the QoS minimum and maximum values generated by the QoS latency regulator for reads and writes for port S2.

Usage constraints Before writing this register, all previous transactions from any devices connected to this AMBA port must be complete and no transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

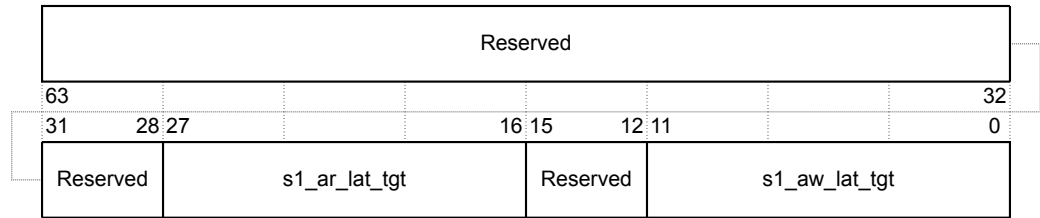The following figure shows the s2_qos_lat_range register bit assignments.

Figure 3-141 s2_qos_lat_range register bit assignments

The following table shows the s2_qos_lat_range register bit assignments.

Table 3-155 s2_qos_lat_range register bit assignments

Bits	Name	Access	Reset value	Function
[63:28]	-	RAZ/WI	0x0	Reserved
[27:24]	s2_ar_lat_max_qos	RW	0x0	S2 AR QoS maximum value

Table 3-155 s2_qos_lat_range register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[23:20]	-	RAZ/WI	0x0	Reserved
[19:16]	s2_ar_lat_min_qos	RW	0x0	S2 AR QoS minimum value
[15:12]	-	RAZ/WI	0x0	Reserved
[11:8]	s2_aw_lat_max_qos	RW	0x0	S2 AW QoS maximum value
[7:4]	-	RAZ/WI	0x0	Reserved
[3:0]	s2_aw_lat_min_qos	RW	0x0	S2 AW QoS minimum value

RN-I Auxiliary Control register

The aux_ctl register is at offset 0x0500. Its characteristics are:

Purpose   Controls various modes of operation.

Usage constraints   Only accessible by Secure accesses. Before writing this register, all previous transactions from any device connected to this device port must be complete and no other transactions can be initiated until the write to this register is complete.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

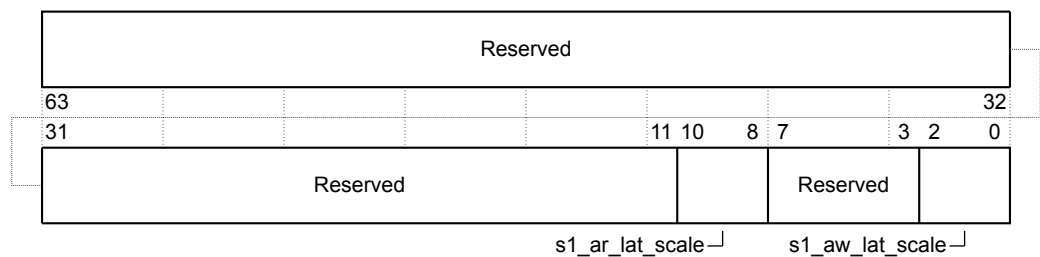The following figure shows the aux_ctl register bit assignments.
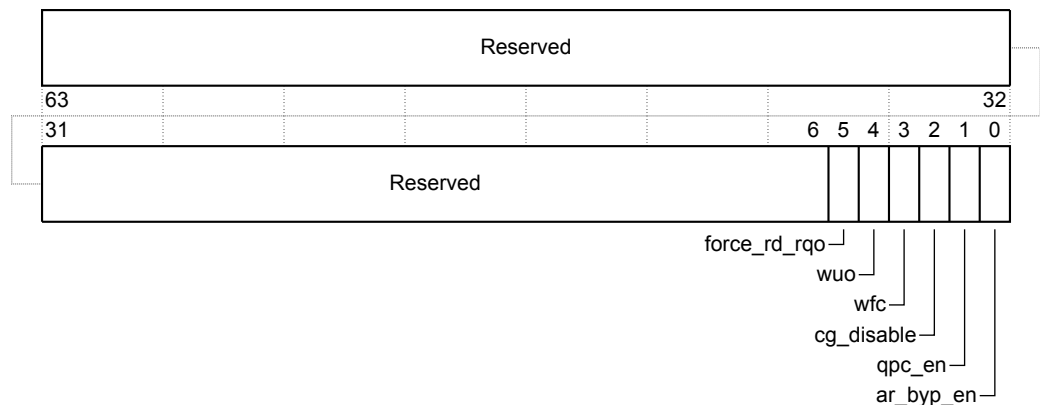
Figure 3-142 aux_ctl register bit assignments

The following table shows the aux_ctl register bit assignments.

Table 3-156 aux_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:6]	-	RAZ/WI	0x0	Reserved
[5]	force_rd_rqo	RW	0	Forces all reads from the RN-I to be sent with the Request Order bit set and this ensures ordered allocation of read data buffers in the RN-I.
[4]	wuo	RW	0	Used for acceleration of coherent ordered writes, and is particularly useful for PCIe traffic. This bit can be set for only one RN-I in the system. The reset value for this bit is 0 for all RN-I components in the system.
[3]	wfc	RW	0	Enables waiting for Comp before the dependent transaction is dispatched.
[2]	cg_disable	RW	0	Clock gating disable. When set, this bit disables clock gating.

Table 3-156 aux_ctl register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[1]	qpc_en	RW	0	QPC enable. When set, this bit enables QoS based scheduling using two QoS priority classes, QoS15 and non-QoS15.
[0]	ar_byp_en	RW	1	AR bypass enable. Enables bypass path in the AR pipeline.

PMU Event Select register, RN-I

The pmu_event_sel register is at offset `0x0600`. Its characteristics are:

Purpose   Selects the PMU events to be counted.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

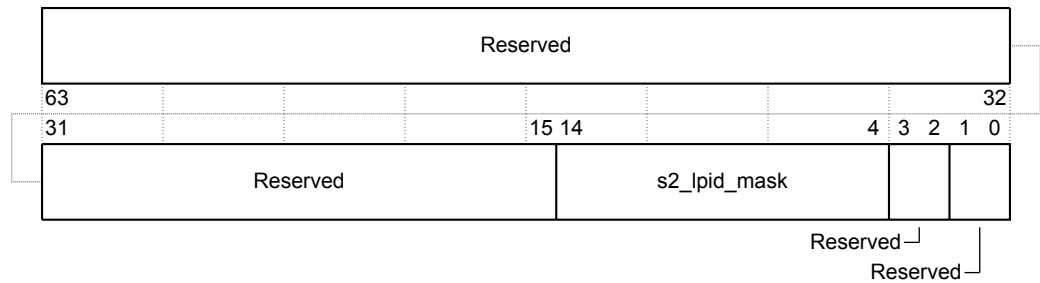The following figure shows the pmu_event_sel register bit assignments.

Figure 3-143 pmu_event_sel register bit assignments

The following table shows the pmu_event_sel register bit assignments.

Table 3-157 pmu_event_sel register bit assignments

Bits	Name	Access	Reset value	Function
[63:16]	-	RAZ/WI	0x0	Reserved
[15:12]	pmu_event3_id	RW	0x0	PMU Event 3 ID. The event is specified as a 4-bit ID with the following encodings: `0b0000` Null (no event). `0b0001` S0 RDataBeats. `0b0010` S1 RDataBeats. `0b0011` S2 RDataBeats. `0b0100` RXDAT flits received. `0b0101` TXDAT flits sent. `0b0110` Total TXREQ flits sent. `0b0111` Retried TXREQ flits sent. `0b1000` RRT full. `0b1001` WRT full. `0b1010` Replayed TXREQ flits. All other values are Reserved.

Table 3-157 pmu_event_sel register bit assignments (continued)

Bits	Name	Access	Reset value	Function
[11:8]	pmu_event2_id	RW	0x0	PMU Event 2 ID. See pmu_event3_id in this table for more information.
[7:4]	pmu_event1_id	RW	0x0	PMU Event 1 ID. See pmu_event3_id in this table for more information.
[3:0]	pmu_event0_id	RW	0x0	PMU Event 0 ID. See pmu_event3_id in this table for more information.

RN-I Identification register

The oly_rni_oly_id register is at offset `0xFF00`. Its characteristics are:

Purpose   Contains the component identification information.

Usage constraints There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-9 RN-I bridge register summary* on page 3-91.

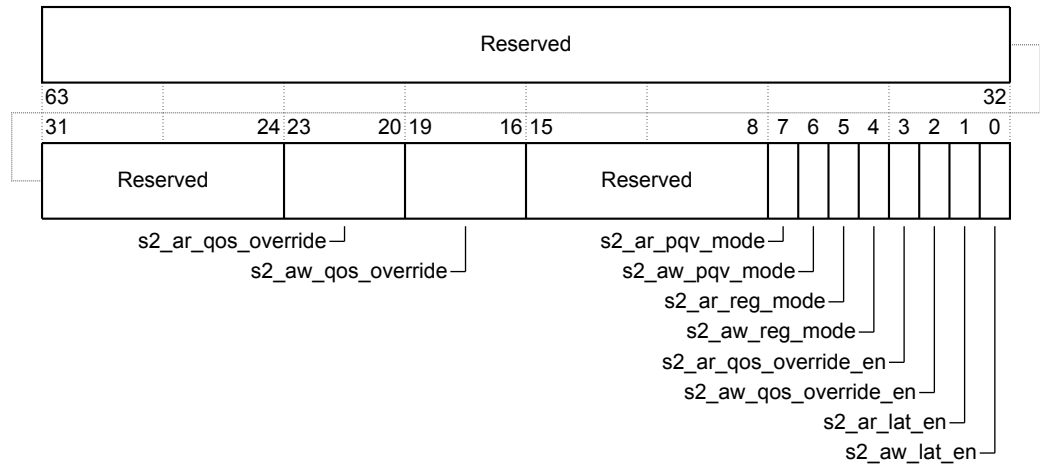The following figure shows the oly_rni_oly_id register bit assignments.

Figure 3-144 oly_rni_oly_id register bit assignments

The following table shows the oly_rni_oly_id register bit assignments.

Table 3-158 oly_rni_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	Value is specific to each RN-I bridge	The node ID of the RN-I bridge
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0x16	Indicates that this node is an RN-I bridge that supports 3 ACE-Lite interfaces

Related references

3.3.7 SBSX register descriptions

This section lists the SBSX registers.

- *SA Auxiliary Control register, SBSX* on page 3-206.
- *SBSX Identification register* on page 3-206.

SA Auxiliary Control register, SBSX

The sa_aux_ctl register is at offset `0x0500`. Its characteristics are:

Purpose	Controls the operation of the SA bridges.
Usage constraints	This register can be modified only with prior written permission from ARM.
Configurations	Available in all configurations.
Attributes	See *Table 3-10 SBSX register summary* on page 3-92.

The following figure shows the sa_aux_ctl register bit assignments.
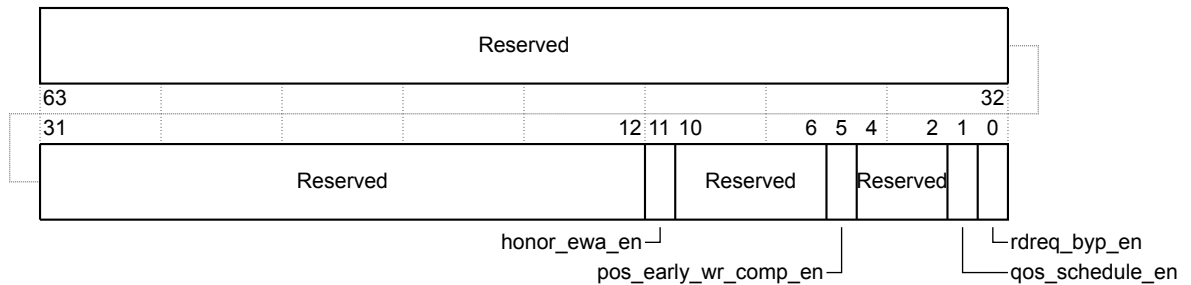
Figure 3-145 sa_aux_ctl register bit assignments

The following table shows the sa_aux_ctl register bit assignments.

Table 3-159 sa_aux_ctl register bit assignments

Bits	Name	Access	Reset value	Function
[63:12]	-	RAZ/WI	0x0	Reserved
[11]	honor_ewa_en	RW	0	If EWA=0, do not send write completion until the bridge receives completion
[10:8]	-	WI	0b100	Reserved
[7]	-	RW	1	Reserved
[6]	-	RW	1	Reserved
[5]	pos_early_wr_comp_en	RW	1	Enable early write completions for all writes that allow early acknowledgment
[4]	-	RW	1	Reserved
[3]	-	RW	0	Reserved
[2]	-	RW	1	Reserved
[1]	qos_schedule_en	RW	1	Set to 1 to enable QoS based scheduling of the AMBA requests
[0]	rdreq_byp_en	RW	1	Set to 1 to enable read bypass path

SBSX Identification register

The oly_sbsx_oly_id register is at offset `0xFF00`. Its characteristics are:

Purpose	Contains the component identification information.

Usage constraints   There are no usage constraints.

Configurations   Available in all configurations.

Attributes   See *Table 3-10 SBSX register summary* on page 3-92.

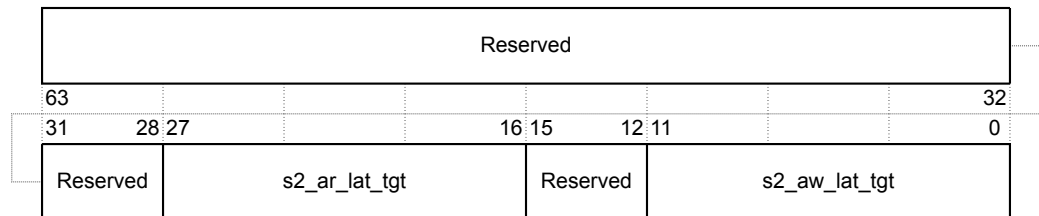The following figure shows the oly_sbsx_oly_id register bit assignments.

Figure 3-146 oly_sbsx_oly_id register bit assignments

The following table shows the oly_sbsx_oly_id register bit assignments.

Table 3-160 oly_sbsx_oly_id register bit assignments

Bits	Name	Access	Reset value	Function
[63:15]	-	RAZ/WI	0x0	Reserved
[14:8]	node_id	RO	Value is specific to each SBSX bridge	The node ID of the SBSX
[7:5]	-	RAZ/WI	0b000	Reserved
[4:0]	oly_id	RO	0xC	Indicates that this node is an SBSX

Related references

3.1.1 Node configuration register address mapping on page 3-82.

3.1.2 Node type IDs on page 3-85.

3.4 Programming the CCN-502

The processor must be programmed to enable correct operation with the CCN-502.

This section contains the following subsections:

3.4.1 Boot-time programming requirements

Describes the programming initialization requirements to support coherent transactions and DVMOps. If the network contains only a single SN-F, or if the network contains only three SN-Fs, then it also describes the requirement to program the hnf_sam_control register.

The CCN-502 is configured to support device-type accesses immediately out of reset, so no additional programming is required before beginning device-type system-level communication. However, to send coherent transactions or *Distributed Virtual Memory* (DVM) operations, the system programmer must ensure that the appropriate RNs are entered into the required snoop and DVM domains.

Many of the CCN-502 configuration and status registers have constraints about when and how you can program them. You must respect these constraints to prevent unpredictable behavior.

SN-F configuration

The CCN-502 supports either one, two, three, or four SN-Fs, that is, memory. If CCN-502 is configured at build time to include three memory controllers, all versions of the hnf_sam_control register in all HN-Fs must be programmed as described in *3 SN-F memory striping* on page 2-57. If CCN-502 is configured at build time to include two or four memory controllers, no programming of the hnf_sam_control registers is required to use those two or four memory controllers.

To use a different collection of memory controllers than was specified at build time, all versions of the hnf_sam_control register across all HN-Fs must be programmed as described in *2.12.4 HN-F SAM* on page 2-56.

SN-F configuration must be completed before the first request by the system to normal memory.

Related concepts

Entry to and exit from snoop and DVM domains on page 3-209.

Related references

3.2 Register summary on page 3-87.
HN-F SAM Control register on page 3-143.

3.4.2 Programming requirements for designs with an alternative path to the HN-I memory space

For slaves that are in the HN-I memory region, some system designs might provide access to those slaves through a different path from the HN-I. In these circumstances, you must program the HN-I registers so that the *Point-of-Serialization* (PoS) is set correctly.

The following figure shows an example system configuration where an HN-I slave is accessible to other AXI masters.
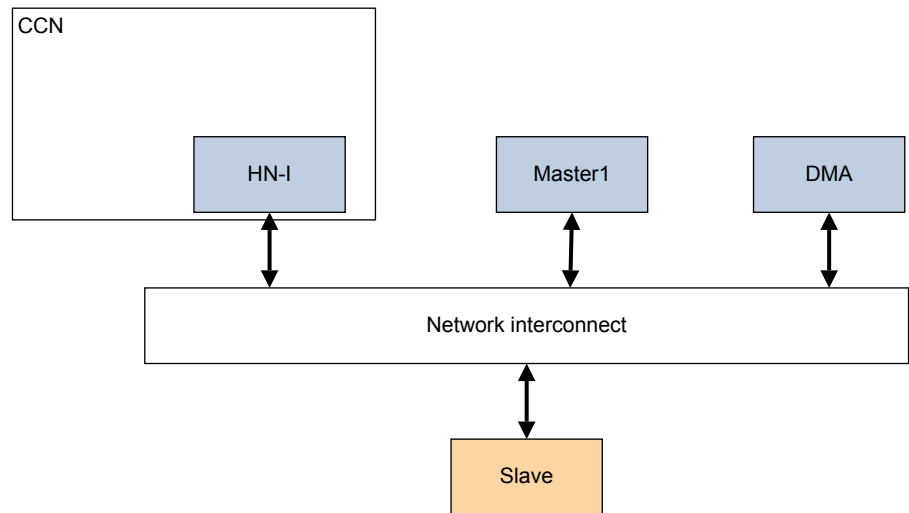
Figure 3-147 HN-I slave that is accessible by other masters

Procedure

The required programming is:

1. Set hni_pos_en = 0 in the HN-I pos_control register, so that the HN-I is not the final PoS.

2. Optional: Set honor_ewa_en = 1 in the HN-I sa_aux_ctl register.

   This bit controls whether the HN-I passes downstream error responses for writes.

3. Set the four pos_* bits to zero in the HN-I sa_aux_ctl register.

   That is, set:
   - pos_early_eobarrsp_en = 0.
   - pos_early_rdack_en = 0.
   - pos_early_wr_comp_en = 0.
   - pos_terminate_barriers = 0.

4. Set ser_devne_wr = 1 in the HN-I sa_aux_ctl register.

   When set, the HN-I serializes the Device-nGnRnE writes, and does not send any other write request with the same **AWID** as an outstanding Device-nGnRnE write.

Related references

PoS Control register on page 3-165.
SA Auxiliary Control register, HN-I on page 3-169.

3.4.3   Runtime programming requirements

This section describes the requirements for programming during runtime.

Entry to and exit from snoop and DVM domains

The CCN-502 includes a means by which RNs can be removed from snoop and DVM domains, to ensure correct operation of both snoops and DVMs when an RN is taken out of reset, or is powered down and then later powered up.

Control of the device inclusion or exclusion in snoop and DVM domains is critical to the HN-F and MN, because the HN-F and MN must be aware of the RNs that are present and active in the snoop or DVM domain at the time a snoop request or DVM message is sent. Any mismatch in the HN-F or MN understanding of the domain participants and the ability of those participants to respond to a snoop or DVM request results in unpredictable behavior.

Entry to and exit from a snoop or DVM domain is achieved through a series of configuration writes and reads that ensure atomic entry to and exit from a snoop or DVM domain, as described in:

- *Atomicity requirements for entry to or exit from a snoop or DVM domain* on page 3-210.
- *Entry to snoop domain* on page 3-210.
- *Exit from snoop domain* on page 3-211.
- *Entry to DVM domain* on page 3-212.
- *Exit from DVM domain* on page 3-212.

——————— **Note** ———————

The following subsections mention the SDCR* and DDCR* registers. These registers are the Snoop Domain Control register and its variants, and the DVM Domain Control register and its variants, in the HN-F and MN configuration register spaces respectively. For more information, see:

- *Snoop Domain Control register* on page 3-154 to *Snoop Domain Control Clear register* on page 3-156.
- *DVM Domain Control register* on page 3-102 to *DVM Domain Control Clear register* on page 3-103.

————————————————

Atomicity requirements for entry to or exit from a snoop or DVM domain

Entry to and exit from the snoop or DVM domain must be atomic for each domain, that is, only one such occurrence of either of these processes for each of the domains can be active in the CCN-502 at any given time. This atomicity requirement is not directly supported in hardware. Therefore it is the responsibility of the device or software thread that is performing the entry or exit process, to guarantee atomicity, either by convention, access through a critical section bounded by mutual exclusion synchronization primitives, or other method. The descriptions of the entry and exit processes in this section assume a critical section is used to ensure atomicity, but this is not a required implementation.

Entry to snoop domain

Entry of one or multiple RNs to a snoop domain is as follows:

——————— **Note** ———————

The atomicity of SDCR updates is required across both sets and clears. Only one of either of these can be in flight at any given time.

————————————————

Critical-section SnoopDomainCritSec{

Foreach (HN-F) {

- An RN or its proxy performs a single write to HN-F SDCR_Set with a 1 in the position corresponding to the nodeID of the RN to be included in the snoop domain.

  ——————— **Note** ———————

  You can concurrently add multiple RNs to the snoop domain by simultaneously setting multiple bits in the SDCR_Set register.

  ————————————————

  When issuing the write of SDCR_Set, the RN being added to the snoop domain must respond to snoop requests that are sent to it.
- When receiving a write to SDCR_Set, the HN-F performs a series of actions, the result of which ensures the HN-F eventually begins sending snoop requests to the RN being added to the snoop domain. It also updates the SDCR to reflect the addition of RNs to the snoop domain.

}

Foreach (HN-F) {

- An RN or its proxy performs a read of the SDCR, comparing the bit-positions previously set in SDCR_Set, repeating this step until the corresponding bit-positions have been set in the SDCR. At this point, the newly added RN is guaranteed to receive all subsequent relevant snoop requests from the corresponding HN-F.

——————— **Note** ———————

It is not valid for the write to SDCR_Set to have a null effect on the SDCR. That is, the write before the polling read must have the effect of modifying the SDCR in at least one bit-position. If not, the subsequent read would immediately reflect completion because the new value equals the old value, and therefore the atomicity requirement described is not guaranteed.

———————————————————

}

When these steps are complete, the newly added RNs are included in the global snoop domain, and receive snoop requests from all HN-Fs as necessary for correct functionality.

}

Exit from snoop domain

Removal of one or multiple RNs from a snoop domain is as follows:

——————— **Note** ———————

The atomicity of SDCR updates is required across both sets and clears. Only one of either of these can be in flight at any given time.

———————————————————

Critical-section SnoopDomainCritSec{

Foreach (HN-F) {

- An RN or its proxy performs a single write to HN-F SDCR_Clear with a 1 in the position corresponding to the nodeID of the RN to be removed from the snoop domain.

——————— **Note** ———————

You can concurrently remove multiple RNs from the snoop domain by simultaneously setting multiple bits in the SDCR_Clear register.

———————————————————

- When receiving a write to SDCR_Clear, the HN-F performs a series of actions, the result of which ensures the HN-F eventually stops sending snoop requests to the RN being removed from the snoop domain. It also updates the SDCR to reflect the removal of RNs from the snoop domain.

}

Foreach (HN-F) {

- An RN or its proxy performs a read of the SDCR, comparing the bit-positions previously set in SDCR_Clear, repeating this step until the corresponding bit-positions have been cleared in the SDCR. When this step is complete, the RN is guaranteed to no longer receive any subsequent snoop requests from the corresponding HN-F.

——————— **Note** ———————

It is not valid for the write to SDCR_Clear described in this section to have NULL effect on the SDCR. That is, the write before the polling read must have the effect of modifying the SDCR in at

least one bit-position. If not, the subsequent read would immediately reflect completion because the new value equals the old value, and therefore the atomicity requirement described is not guaranteed.

_____

}

Before completion of these steps for all HN-Fs, the RN being removed from the snoop domain must respond to snoop requests sent to it. When these steps are complete for all HN-Fs, the RNs are excluded from the global snoop domain and are guaranteed not to receive any subsequent snoop requests.

}

Entry to DVM domain

Entry of one or multiple RNs to a DVM domain is as follows:

——————— **Note** ———————

The atomicity of DDCR updates is required across both sets and clears. Only one of either of these can be in flight at any given time.

_____

Critical-section DvmDomainCritSec{

- An RN or its proxy performs a single write to MN DDCR_Set with a 1 in the position corresponding to the nodeID of the RN to be included in the DVM domain.

  ——————— **Note** ———————

  You can concurrently add multiple RNs to the DVM domain by simultaneously setting multiple bits in the DDCR_Set register.

  _____

  When issuing the write of DDCR_Set, the RN being added to the DVM domain must respond to DVM messages sent to it.
- When receiving a write to DDCR_Set, the MN performs a series of actions, the result of which ensures the MN eventually begins sending DVM messages to the RN being added to the DVM domain. It also updates the DDCR to reflect the addition of RNs to the DVM domain.
- An RN or its proxy performs a read of the DDCR, comparing the bit-positions previously set in DDCR_Set, repeating this step until the corresponding bit-positions have been set in the DDCR.

  ——————— **Note** ———————

  It is not valid for the write to DDCR_Set to have NULL effect on the DDCR. That is, the write before the polling read must have the effect of modifying the DDCR in at least one bit-position. If not, the subsequent read would immediately reflect completion because the new value equals the old value, and therefore the atomicity requirement described is not guaranteed.

  _____

When these steps are complete, the newly added RNs are included in the global DVM domain, and receive all DVM messages as necessary for correct functionality.

}

Exit from DVM domain

Removal of one or multiple RNs from a DVM domain is as follows:

——————— **Note** ———————

The atomicity of DDCR updates is required across both sets and clears. Only one of either of these can be in flight at any given time.

Critical-section DvmDomainCritSec{

• An RN or its proxy performs a single write to MN DDCR_Clear with a 1 in the position corresponding to the nodeID of the RN to be removed from the DVM domain.

——————— **Note** ———————

You can concurrently remove multiple RNs from the DVM domain by simultaneously setting multiple bits in the DDCR_Clear register.

• When receiving a write to DDCR_Clear, the MN performs a series of actions, the result of which ensures the MN eventually stops sending DVM messages to the RN being removed from the DVM domain. It also updates the DDCR to reflect the removal of RNs from the DVM domain.

• An RN or its proxy performs a read of the DDCR, comparing the bit-positions previously set in DDCR_Clear, repeating this step until the corresponding bit-positions have been cleared in the DDCR.

——————— **Note** ———————

It is not valid for the write to DDCR_Clear to have NULL effect on the DDCR. That is, the write before the polling read must have the effect of modifying the DDCR in at least one bit-position. If not, the subsequent read would immediately reflect completion because the new value equals the old value, and therefore the atomicity requirement described is not guaranteed.

Before completion of these steps, the RN being removed from the DVM domain must respond to DVM messages sent to it. When these steps are complete, the RNs are excluded from the global DVM domain and are guaranteed not to receive any subsequent DVM messages.

}

Related references

Snoop Domain Control register on page 3-154.
Snoop Domain Control Clear register on page 3-156.
DVM Domain Control register on page 3-102.
DVM Domain Control Clear register on page 3-103.

Chapter 4
L3 Memory System

This chapter describes the Level 3 memory system.

It contains the following sections:

4.1 About the L3 memory system

The L3 memory system consists of the HN-F protocol node in the CCN-502.

There are 2 (6XP/2HNF) or 4 (8XP/4HNF) instances of the HN-F, and each HN-F node or slice has the following features:

- 0KB, 128KB, 512KB, 1MB, or 2MB of L3 cache data RAM and tag RAM.
- Combined *Point-of-Coherency* (PoC) and *Point-of-Serialization* (PoS).
- 512KB, 2MB, or 4MB snoop filter tag RAM.

Each HN-F in the CCN-502 is configured to manage a specific portion of the total address space. For each portion of the address, each HN-F:

- Can cache data in L3.
- Manages PoC and PoS functionality for ordering and coherency.
- Tracks RN-F caching in the snoop filter.

The L3 memory system has the following features:
- *Physically Indexed and Physically Tagged* (PIPT).
- Coherency granule is a fixed length of 64 bytes. L3 cache line size is a fixed length of 64 bytes.
- Both L3 and snoop filter are 16-way set-associative.
- The L3 and snoop filter victim selection policy is:
  — Find first invalid way.
  — Pseudo random if all ways are valid.
- L3 and snoop filter arrays:
  — Supports two or three cycle non-pipelined tag and data array.
  — L3 tag, snoop filter tag, and L3 data arrays are single ported, supporting one read or write access with no concurrency available.
  — L3 tag, snoop filter tag, and L3 data arrays are ECC (SECDED) protected, with inline ECC checking and correction. SECDED means *Single-Error Correction and Double-Error Detection*.
- 32 entry address and data buffer, known as *PoC Queue* (POCQ),which the 4HNF can have 16 entry POCQ to service:
  — All transactions from the CHI interface.
  — L3 modified evictions to the memory controller.
  — Snoop filter evictions.
- Supports QoS-based protocol flow control:
  — PoC and PoS resources (POCQ) are allocated or rejected for protocol retry, based on the QoS class.
  — POCQ resources are watermarked for different QoS classes with user-configurable options.
  — Starvation prevention for lower-priority QoS classes.
  — QoS-based static grantee selection for CHI architecture credit return.
- QoS priority-based request selection to the memory controller.
- Supports allocation in the L3 cache from snoop intervention. This enables data sharing through the L3 for multiple sharers.
- L3 state includes caching *RN-F IDentifier* (RNFID) to detect dynamic read sharing.
- 44-bit physical address support.
- PoC and PoS for all Snoopable and Non-snoopable, and Cacheable and Non-cacheable address space.
- Supports ECC scrubbing for single bit ECC errors.
- Software-controlled error injection support to enable testing of software error handler routine.
- Power-management states to support:
  — Full powerdown of the L3 and snoop filter. HN-F only mode when both L3 and snoop filter are powered down.
  — Half the L3 ways powered down.
  — Retention for L3 and snoop filter.
  — L3 full powerdown with snoop filter on, when in snoop filter only mode.
- ARM TrustZone® technology support in L3 and snoop filter.

Related references

4.2   Configurable options

The HN-F can be configured in a number of ways.

The configurable parameters are:

*   L3 cache size of 0KB, 128KB, 512KB, 1MB, or 2MB. All L3 slices must be the same size.
*   Snoop filter size of 512KB, 2MB, or 4MB.
*   Two cycle or three cycle tag, data, and snoop filter array RAMs. All RAMs have the same latency.

The HN-F has the following static, or fixed, parameters:
*   Number of HN-F partitions is 2 (6XP/2HNF) or 4 (8XP/4HNF).
*   HN-F CHI interface data channel (DAT) width of 128 bits.

4.3 Cache maintenance operations

The CCN-502 uses several CHI *Cache Maintenance Operation*s (CMOs).

The following operations are supported:

- CleanInvalid.
- CleanShared.
- MakeInvalid.

These operations always look up the L3 cache and the snoop filter, and take the following actions:

- If the CMO is Snoopable, the HN-F sends a snoop to the RN-F post snoop filter lookup if required.
- If the cache line is modified in the L3 or in the cache of the RN-Fs, the HN-F initiates a memory controller writeback if required.

————— Note —————

If the CMO is MakeInvalid, there is no writeback to the memory controller.

————————————

In addition, the L3 cache and snoop filter can be flushed or invalidated by using the power state mechanisms that are described here. For example, the L3 cache can be flushed (cleaned and invalidated) by transitioning from FAM→SFONLY power state, by writing to all instances of the HN-F P-state Request register. Both the L3 and snoop filter can be flushed by transitioning from FAM→NOL3.

The snoop filter does not track RN-F coherence while the HN-F is in NOL3 state, so the RN-F caches must be flushed before transitioning from NOL3 to SFONLY, HAM, or FAM states.

————— Note —————

The system must ensure that no P-Channel interface initiated power transitions are in progress, when writes to the HN-F P-state registers occur.

————————————

Related references

HN-F P-state Request register on page 3-144.
2.14.3 Power states on page 2-71.

4.4 Cacheable and Non-cacheable exclusives

The HN-F supports PoC monitor functionality for Cacheable and snoopable exclusive operations from the RN-Fs.

The Cacheable and snoopable exclusive transactions are:

- ReadShared.
- ReadClean.
- CleanUnique.

The HN-F also supports system monitor functionality for Non-cacheable exclusive support. See the *ARM® AMBA® 5 CHI Architecture Specification* for more information about exclusives.

───────── **Note** ─────────

Each HN-F in the CCN-502 can support tracking of up to 32 logical processors for exclusive operations. The system programmer must ensure that there are no more than 32 logical processors capable of sending exclusive operations.

─────────────────────

4.5 **TrustZone® technology support**

The HN-F supports TrustZone by treating the NS bit from a request as part of the address.

TrustZone support enables the HN-F to treat Secure and Non-secure as two different areas of the memory space:

- The NS bit is stored in the L3 and snoop filter tags.
- Snoops also propagate the NS bit as part of the message.
- Any request to the memory controller also propagates the NS bit.

4.6 Snoop connectivity and control

Each HN-F can send two types of snoop.

The snoop requests are:
- Directed, to one RN-F.
- Broadcast, to all RN-Fs active within the snoop domain.

The snoop domain vectors for all the RN-Fs are maintained at each HN-F using the Snoop Domain Control register. Each HN-F has a copy of this register. It defines which RN-F clusters are active for coherency actions or snoops. Software must ensure that the appropriate bit vector is set or cleared for RN-F entry to and exit from the snoop domain, to enable coherent transactions or power down the RN-Fs.

Related references

Snoop Domain Control register on page 3-154.

3.4.3 Runtime programming requirements on page 3-209.

4.7 QoS features

The HN-F protocol queue (POCQ) is a key shared system resource that communicates with the memory controller for external memory access.

All requests for normal memory must go through the HN-F to ensure proper ordering (PoS) and functional correctness for coherence (PoC).

The HN-F provides QoS capabilities in support of the following traffic classes:

- Real-time or pseudo-real-time traffic that requires a maximum bounded latency at potentially fixed bandwidth.
- Latency-sensitive traffic, traditionally from a processor device.

The CCN-502 uses QoS values to designate these traffic classes. Every request to the HN-F has a 4-bit QoS value that is associated with it, with a higher number indicating a higher priority. The four QoS classes are:

- Highest priority, known as HighHigh.
- High priority.
- Medium priority.
- Low priority.

This section contains the following subsections:

- *4.7.1 QoS decoding* on page 4-222.
- *4.7.2 QoS class and POCQ resource availability* on page 4-223.

4.7.1 QoS decoding

QoS decoding takes place inside the HN-F.

The QoS decoding is as follows:

- The CHI interface supports a 4-bit QoS value.
- The 4-bit QoS has 16 possible values. The following table shows the default map of the QoS classes that the HN-F creates.

Table 4-1 QoS classes

QoS value range	QoS class	Class mnemonic	Priority
15	HighHigh	HH	Highest
14-12	High	H	High
11-8	Med	M	Medium
7-0	Low	L	Low

- QoS mapping is fixed and is shown in the qos_band register.

The POCQ is logically partitioned to service different QoS class traffic. The HN-F also uses the priorities in the previous table, where necessary, to arbitrate for the following:

- Memory controller request selection in the POCQ control block.
- Data return selection logic, that is, a COMPDATA to a requester.
- Protocol credits that are sent to an RN-F or RN-I following a protocol-layer retry.

The POCQ resource and buffers are logically partitioned based on the QoS value.

Related concepts

4.7.2 QoS class and POCQ resource availability on page 4-223.

Related references

QoS Band register on page 3-146.

4.7.2 QoS class and POCQ resource availability

The POCQ buffers are shared resources for all QoS classes.

The higher the QoS class, the higher the occupancy availability. For example, the *HighHigh* (HH) QoS class can use all the POCQ entries except for the dedicated snoop filter pool.

The following figure shows the availability of POCQ resources for various QoS levels, using a particular QoS pool that is shared between multiple QoS classes.

POCQ logical view
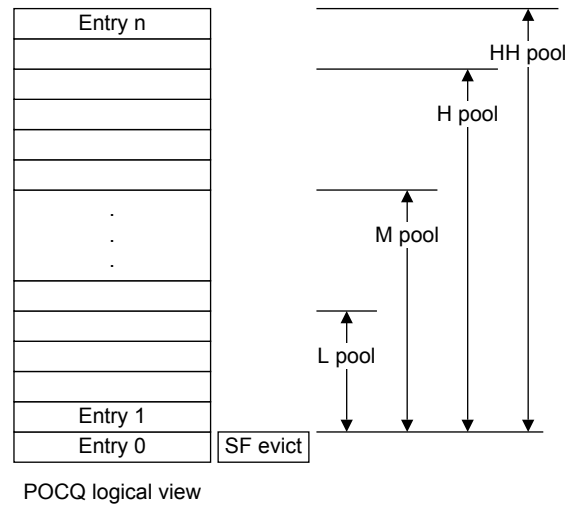
Figure 4-1 POCQ availability and QoS classes

The QoS pools are:

hh_pool	Available for HH class.
h_pool	Available for H class and HH class.
m_pool	Available for M class, H class, and HH class.
l_pool	Available for all classes.
seq	Snoop filter evictions only.

This scheme enables a higher-priority QoS class to have more POCQ resources for transaction processing, and prevents a lower-priority QoS from using all the POCQ. The level of POCQ availability decreases for the lower QoS classes.

QoS pool distribution of the POCQ is software-configurable using the qos_reservation register.

Related references

QoS Reservation register on page 3-147.

4.8 Software configurable memory region locking

The HN-F supports variable size memory regions that can be locked in the L3 cache with way reservation.

These variable size memory regions ensure that locked lines are not evicted from the L3.

Software uses the following mechanism to program the HN-F configuration registers to enable region locking:

- The hnf_l3_lock_ways register specifies the total number of locked HN-F L3 ways. This can be a value of 1, 2, 4, 8, or 12.
- The following region base registers specify the base address of the region that is using locked ways:
  — hnf_l3_lock_base0 register.
  — hnf_l3_lock_base1 register.
  — hnf_l3_lock_base2 register.
  — hnf_l3_lock_base3 register.
- A combination of the total L3 size, hnf_l3_lock_ways register, and the hnf_l3_lock_base0 register to hnf_l3_lock_base3 register defines the following:
  — The total amount of cache locked, calculated as follows:

$$\frac{\text{Total SLC size x Number of locked ways}}{16}$$

  — Exactly which ways are locked. Ways are locked beginning with way 0 and then in ascending order.
  — The number of valid regions and exactly which regions, and therefore which of the hnf_l3_lock_base0 to hnf_l3_lock_base3 registers, are valid and included in the HN-F way allocation.
  — The exact location, size, and alignment requirement of each region.
- The region alignment is identical to the region size, for example:
  — A 0.5MB region is aligned to any 0.5MB boundary.
  — A 4MB region is aligned to any 4MB boundary.
- The size and alignment requirement is enforced in hardware, to prevent any errors in software.
- Regions can be disjointed or contiguous, to create a larger single region.
- All valid regions use all locked ways. There is no application-level way segregation.
- No overlocking is allowed. This means it is not possible to have more indices per set than is supported by the number of locked ways, preventing the spilling of locked ways.

——————— Note ———————

The locked regions do not comprehend Secure as opposed to Non-secure memory regions, so overlocking can occur if aliasing is performed between Secure and Non-secure regions.

————————————————

The following tables specify various combinations of region size and the number of locked ways that software must program using the hnf_l3_lock_ways register and the hnf_l3_lock_base0 register to hnf_l3_lock_base3 register.

Table 4-2 hnf_l3_lock_ways register settings

L3 size	Number of locked ways	Total locked region size	Locked ways	Number of ways per region	Region 0	Region 1	Region 2	Region 3
6XP/2HNF								
256KB	1	16KB	0	1	16KB	-	-	-
	2	32KB	0-1	1, 1	16KB	16KB	-	-
	4	64KB	0-3	1, 1, 1, 1	16KB	16KB	16KB	16KB
	8	128KB	0-7	2, 2, 2, 2	32KB	32KB	32KB	32KB
	12	192KB	0-11	2, 2, 4, 4	32KB	32KB	64KB	64KB
8XP/4HNF								
512KB	1	32KB	0	1	32KB	-	-	-
	2	64KB	0-1	1, 1	32KB	32KB	-	-
	4	128KB	0-3	1, 1, 1, 1	32KB	32KB	32KB	32KB
	8	256KB	0-7	2, 2, 2, 2	64KB	64KB	64KB	64KB
	12	384KB	0-11	2, 2, 4, 4	64KB	64KB	128KB	128KB
6XP/2HNF								
1MB	1	64KB	0	1	64KB	-	-	-
	2	128KB	0-1	1, 1	64KB	64KB	-	-
	4	256KB	0-3	1, 1, 1, 1	64KB	64KB	64KB	64KB
	8	512KB	0-7	2, 2, 2, 2	128KB	128KB	128KB	128KB
	12	768KB	0-11	2, 2, 4, 4	128KB	128KB	256KB	256KB
6XP/2HNF or 8XP/4HNF								
2MB	1	128KB	0	1	128KB	-	-	-
	2	256KB	0-1	1, 1	128KB	128KB	-	-
	4	512KB	0-3	1, 1, 1, 1	128KB	128KB	128KB	128KB
	8	1MB	0-7	2, 2, 2, 2	256KB	256KB	256KB	256KB
	12	1.5MB	0-11	2, 2, 4, 4	256KB	256KB	512KB	512KB
6XP/2HNF or 8XP/4HNF								
4MB	1	256KB	0	1	256KB	-	-	-
	2	512KB	0-1	1, 1	256KB	256KB	-	-
	4	1MB	0-3	1, 1, 1, 1	256KB	256KB	256KB	256KB
	8	2MB	0-7	2, 2, 2, 2	512KB	512KB	512KB	512KB
	12	3MB	0-11	2, 2, 4, 4	512KB	512KB	1MB	1MB
8XP/4HNF								
8MB	1	512KB	0	1	512KB	-	-	-
	2	1MB	0-1	1, 1	512KB	512KB	-	-
	4	2MB	0-3	1, 1, 1, 1	512KB	512KB	512KB	512KB
	8	4MB	0-7	2, 2, 2, 2	1MB	1MB	1MB	1MB
	12	6MB	0-11	2, 2, 4, 4	1MB	1MB	2MB	2MB

Related references

4.9 Performance monitoring events

Each HN-F can monitor all the PMU events.

However, only four of the HN-F PMU events can be tracked through each HN-F at any given time. Software must set up the pmu_event_sel register as required to select which four events HN-F must report for the PMU count logic.

All Secure HN-F events are gated with the **SPNIDEN** input signal to the HN-F. The SoC drives **SPNIDEN** to the CCN-502 to either count or not count Secure PMU events. The SoC must set **SPNIDEN** HIGH to count Secure PMU events.

Related concepts

6.3 HN-F performance events on page 6-250.

Related references

PMU Event Select register, L3 cache on page 3-162.

4.10 Error reporting and software configured error injection

The HN-F reports errors to the MN block.

The following errors are reported by the HN-F block:
- Double-bit ECC errors in the L3 data RAM.
- Double-bit ECC errors in the L3 tag RAM.
- Double-bit ECC errors in the snoop filter tag RAM.

4.10.1 Software-configurable error injection

The HN-F supports software-configurable error injection and reporting. This feature enables testing of the software error handler routine for L3 double-bit ECC data errors.

The HN-F configuration register for a particular logical thread enables configurable error injection and reporting. When enabled, any Cacheable read for which the HN-F provides the data, that is, an L3 hit, drives the slave error from the L3 pipe and drives an error interrupt through the MN for that read. This emulates a double-bit ECC error in the L3 data RAM without polluting the L3 data RAM through the fill path.

————— Note —————

L3 misses do not drive any slave errors or error interrupts. This mechanism is designed to mimic L3 data ECC errors for L3 hits.

To configure error injection, use the following bits in the hnf_err_inj register:

Bit[0], hnf_err_inj_en   Enables the HN-F error report and injection. When enabled, any Cacheable read compares its SrcID and LPID with the hnf_err_inj_srcid and hnf_err_inj_lpid bits to report a slave error if the HN-F provides data for an L3 hit.

Bits[7:1], hnf_err_inj_srcid   SrcID of the requester to inject error.

Bits[10:8], hnf_err_inj_lpid   LPID of the requester to inject error.

Related references

HN-F Error Injection Enable and Setup register on page 3-149.

4.10.2 Single-bit ECC error tracking and interrupt

The HN-F monitors and locally logs single-bit ECC errors.

The following single-bit ECC errors are monitored and logged by the HN-F:
- Single-bit ECC errors in the L3 data RAM.
- Single-bit ECC errors in the L3 tag RAM.
- Single-bit ECC errors in the snoop filter tag RAM.

Related concepts

2.9 Error handling on page 2-45.

4.11 OCM

On-Chip Memory (OCM) allows for the creation of CCN systems without physical DDR memory.

In OCM mode, the CCN-502 does not send requests to the SN-F provided the following requirements are met:

* Non-cacheable and non-allocating accesses generate requests to the SN-F, so software must ensure that the OCM memory region is Cacheable and Read-Write-Allocate.
* Partial WriteBack, partial WriteClean, and partial snoop data responses generate requests to the SN-F.

——————— **Note** ———————

The ARM Cortex-A53 processor and Cortex-A57 processor do not generate partial write requests.

——————————————

* The HN-F must be in the FAM power state. The other CCN-502 power states are not supported in OCM mode.
* Enabling the OCM mode must be done across all HN-Fs in CCN-502. Any transactions that are outstanding while enabling OCM mode have a non-deterministic behavior. Therefore, enable OCM mode before any transactions are sent to the CCN-502.

——————— **Note** ———————

If any of these requirements are not met, the system must be able to generate correct responses for any requests that target the SN-F.

——————————————

In OCM mode, cache maintenance operations terminate in the L3. CleanInvalid and CleanShared CMOs terminate in the L3 without performing a WriteBack to the SN-F. MakeInvalid invalidates the L3 cacheline, and can be used to invalidate the OCM region.

The CCN-502 operates in OCM mode when the hnf_ocm_en bit is set to 1, in the HN-F Auxiliary Control register. If the hnf_ocm_allways_en bit is set to 1, then all transactions targeting the HN-Fs have OCM behavior. The OCM region must be contiguous and aligned to the total L3 size of the configuration when hnf_ocm_allways_en is set to 1. If the hnf_ocm_allways_en bit is 0, the OCM regions are defined by the region locking registers that *4.8 Software configurable memory region locking* on page 4-224 describes.

Related references

HN-F Auxiliary Control register on page 3-161.

Chapter 5
Debug

This chapter describes the debug features.

It contains the following sections:

5.1   About debug

The CCN-502 provides at-speed self-hosted debug and trace capabilities.

The debug and trace functionality is provided by the following modules:
- Debug Watchpoint Module.
- Debug and Trace Bus.
- Debug Event Module.

Related concepts

5.2 Debug Watchpoint Module on page 5-232.

5.3 Debug and Trace Bus on page 5-234.

Related references

5.4 Debug Event Module on page 5-236.

3.3.5 Debug event module register descriptions on page 3-172.

5.2   Debug Watchpoint Module

Each XP includes two watchpoints in a *Debug Watchpoint Module* (DWM). These watchpoints can be used to compare the fields of flits entering or exiting an XP on a device port with the maskable flit-field values that the system programmer provides.

The two watchpoints are shared across both device ports in any configuration:

- Both watchpoints used for either device port.
- Either watchpoint used for either device port.
- A subset of these where both watchpoints are not required.

Each watchpoint includes the following, where * represents 0 or 1:

- A compare low value, dt_cmp_val*_l.
- A compare high value, dt_cmp_val*_h.
- A compare low mask, dt_cmp_mask*_l.
- A compare high mask, dt_cmp_mask*_h.

Each of these covers identical fields of CHI flits. All fields of all flits of all channels in both transmit and receive directions can be compared against, using the watchpoint value and mask, except for the Data field of the DAT channel, although the watchpoint can only be applied to any one channel in any direction at any given time. Because of this, watchpoint matches on pure data are not possible. After you configure the watchpoint compare value and mask, and enable the watchpoint, every flit from the selected channel that enters the XP from a device or exits the XP to a device, depending on which direction is chosen for comparison with the watchpoint, is mask-compared against the compare value.

You can configure watchpoints to qualify the match and trigger capability with other events. Using the wp*_arm_sel fields in the dt_control register, you can configure the watchpoints to only start comparing after a condition has occurred, which arms the watchpoint. Arming conditions include an event on any of the DTB wires or a trigger by the opposite watchpoint. Alternatively, you can configure the watchpoints to be always armed. In addition, using the wp*_event_count fields, you can configure each watchpoint to preclude triggering until the arming condition has occurred a specific number of times.

In case of a watchpoint match, you can optionally configure the watchpoint to snapshot the flit that matched the watchpoint compare value and mask, so that software can read the flit contents for more debug visibility. The flit data is captured in the dt_cmp_val*_l and dt_cmp_val*_h registers, overwriting their compare values. Only the first flit that a watchpoint matches can be snapshotted. The snapshot status is captured in the sscapture_status field in the dt_status register corresponding to the matching watchpoint. This inhibits snapshotting after the first snapshot, although watchpoint matches and their reporting are not affected. When the sscapture_status field is cleared, snapshotting capability resumes.

——————— Note ———————

Although not essential for correct watchpoint compare functionality, you can rewrite dt_cmp_val*_l and dt_cmp_val*_h fields with your own compare value before clearing the sscapture_status field.

————————————————

The result of the watchpoint comparison can be written to the *Debug and Trace Bus* (DTB). This is a single-cycle assertion of the DTB wire indicating the occurrence of an event. From here, it is routed to the *Debug Event Module* (DEM) for processing. The watchpoint match can be placed on the DTB in various ways, including the following:

- Watchpoint compare result written directly to the DTB.
- OR of the watchpoint compare from watchpoint 0 and watchpoint 1 written to the DTB.
- Local watchpoint compare ORed with the watchpoint compare from the previous XP and written to the DTB.
- OR of the local watchpoint compares from watchpoint 0 and watchpoint 1 ORed with the watchpoint compare from the previous XP and written to the DTB.

The result of a watchpoint match is a watchpoint trigger event. A trigger event consists of a number of optional outcomes including the following:

- Asserting one of the DTB bits.
- Arming the opposite watchpoint.
- Copying the TXNID field from watchpoint 0 input flit to watchpoint 1.
- Snapshotting the matching flit.

These optional trigger outcomes are subject to matching and triggering qualifiers.

Because the CHI architecture includes network-addressed flits, a common requirement for a watchpoint is to compare using the *Transaction ID* (TXNID) field. To set watchpoints to match on both request and responses into and out of a given device, the DWM includes a capability to copy the TXNID field from a flit which matches on watchpoint 0 into the TXNID field of watchpoint 1. This enables watchpoint 1 to uniquely match and trigger only on the corresponding response flit. Using this mechanism:

- The txnid_copyover bit in the dt_control register can be asserted.
- Watchpoint 0 can be set up with all valid flit fields to match a specific request, except TXNID.
- Watchpoint1 can be configured with all valid flit fields to match a response, except TXNID, and configured to be armed by a match on watchpoint 0.

This has the effect that watchpoint 0 can match on a given request, with the arming and event-count qualifiers. Then, on triggering, it can copy the matching flit TXNID to watchpoint 1, which is then armed and can be expected to match, trigger, and snapshot the response flit corresponding to the original request.

The watchpoint functionality is not limited to a debug usage model, but is also a main part of the PMU architecture.

Related references

Debug and Trace Comparison Low Value 0 register on page 3-123.
Debug and Trace Comparison Low Value 1 register on page 3-127.
Debug and Trace Comparison High Value 0 register on page 3-124.
Debug and Trace Comparison High Value 1 register on page 3-128.
Debug and Trace Comparison Low Mask 0 register on page 3-125.
Debug and Trace Comparison Low Mask 1 register on page 3-129.
Debug and Trace Comparison High Mask 0 register on page 3-126.
Debug and Trace Comparison High Mask 1 register on page 3-130.
Debug and Trace Control register, dt_control on page 3-131.
Debug and Trace Status register on page 3-134.

5.3 Debug and Trace Bus

The *Debug and Trace Bus* (DTB) is an 8-bit ring-bus.

The DTB:

1. Originates at the XP clockwise-adjacent to the XP to which the MN is connected. This is XP6 in the CCN-502.
2. Travels from XP to XP in a clockwise direction back to the XP to which the MN is connected. From this XP, the DTB is input to the DEM located within the MN unit.
3. Travels clockwise from the XP to which the MN is connected back to the originating XP, completing the ring.

The watchpoint compare result from any watchpoint in any XP can be placed on any of the eight bits of the DTB, so up to eight DT events can be concurrently active in the CCN-502 at any given time.

The design of the DTB logic means that any given DT event, corresponding to one bit of the DTB, can be a single watchpoint compare result, or can consist of various combinations of watchpoint compare results.

Because of this flexibility, a given DTB bit might be oversubscribed so that a watchpoint compare event from one XP might be overwritten by a watchpoint event from a subsequent XP. This is because a given watchpoint match must proceed through neighboring XPs in a clockwise direction over the DTB to arrive at the DEM. Ensure that you use the flexibility in the watchpoint and DTB systems correctly to provide the required visibility to the interconnect traffic being watched by the watchpoints.

The PMU also uses the DTB to transmit performance events from each of the components to centralized performance monitor counters in the DEM. Each component includes a 4-bit PMU interface to the XP, over which up to four performance events for that component can be transmitted simultaneously. In the XP, any of these events can then be written to a bit of the DTB, in a similar way to the watchpoints.

——————— Note ———————

Multiplexing of the component-specific performance events onto the 4-bit PMU interface to the XP is performed locally in each component. Because the DTB travels through the XPs, the XP performance events are also multiplexed locally in the XP, but have a separate 4-bit PMU interface to the DTB, separate from the 4-bit PMU interfaces from the two attached components.

————————————————————

The dt_config register in the XP enables 16 options for watchpoint and PMU events to contribute to a DTB event, or to be written to a single bit of the DTB, as the following table shows.

Table 5-1 DTB contribution options

Value of dt_config	DTB contribution
0x0	DT bus input from previous XP (pass-through)
0x1	OR of watchpoint 0 and 1
0x2	Watchpoint 0
0x3	Watchpoint 1
0x4	XP PMU event 0
0x5	XP PMU event 1
0x6	XP PMU event 2
0x7	XP PMU event 3
0x8	Device 0 PMU event 0
0x9	Device 0 PMU event 1

Table 5-1 DTB contribution options (continued)

Value of dt_config	DTB contribution
0xA	Device 0 PMU event 2
0xB	Device 0 PMU event 3
0xC	Device 1 PMU event 0
0xD	Device 1 PMU event 1
0xE	Device 1 PMU event 2
0xF	Device 1 PMU event 3

The dt_bus_or_mode capability in the XP dt_control register effectively doubles the available options for writing to the DTB (to 31 options, specifically, because pass-through is not affected by dt_bus_or_mode). When this field is asserted, the DTB contribution in the previous table is ORed with the preceding DTB value for the corresponding DTB bit.

To correctly transmit a DT event from one watchpoint or XP to the DEM, you must understand the CCN-502 topology and the DTB ring bus to ensure that, for the applicable DTB bit, all XPs between the DTB event originator and the DEM are correctly configured to pass through the DT event.

Related concepts

5.2 Debug Watchpoint Module on page 5-232.

Related references

Debug and Trace Configuration register on page 3-121.
Debug and Trace Control register, dt_control on page 3-131.

5.4 Debug Event Module

The *Debug Event Module* (DEM) is the central logic module for all trigger, trace, and PMU-counting capabilities in the CCN-502.

This section contains the following subsections:

5.4.1 DEM trigger capabilities

To improve at-speed debug, the DEM uses the incoming DTB to create an external trigger signal, **DBGWATCHTRIGREQ**. This signal is paired with an input signal, **DBGWATCHTRIGACK**. These signals are asynchronous-safe and communicate using a 4-phase handshake protocol.

The DEM includes logic that selects any combination of the DTB bits, using an OR function when multiple DTB bits are selected, to generate the **DBGWATCHTRIGREQ** signal. Therefore, any watchpoint match or PMU event that causes a 1 to be asserted on the DTB can be translated by the DEM into an assertion of **DBGWATCHTRIGREQ**.

――――― **Note** ―――――

The 4-phase handshake might not complete if debug is disabled when **DBGWATCHTRIGREQ** is asserted.

――――――――――――

When each assertion of **DBGWATCHTRIGREQ** occurs, the DEM snapshots the DTB inputs that contributed to that assertion in the trigger_status register. This is a function of the value of the DTB and the trigger select mask in the trigger_sel field in the trigger_ctl register. Therefore, the software or hardware responding to the assertion of **DBGWATCHTRIGREQ** can query the trigger_status register to determine which watchpoints contributed to that assertion.

The contents of the trigger_status field in the trigger_status register are sticky, that is, an assertion of any bit in that register remains asserted regardless of the value of the DTB at subsequent assertions of **DBGWATCHTRIGREQ**, until the register is cleared with a software write to the trigger_status_clr register.

The DEM also includes a programmable timer that can be applied to create a delay from the DTB assertion to the assertion of **DBGWATCHTRIGREQ**. You can program this timer to add 0-65535 cycles between these two events.

Related references
Trigger Control register on page 3-173.
Trigger Status register on page 3-173.
Trigger Status Clear register on page 3-174.

5.4.2 DEM trace capabilities

To improve at-speed trace, the DEM converts the incoming DTB into an external interface, **STMHWEVENT**. This interface runs directly in the *Hardware Event Observability Interface* (HEOI) of a CoreSight *System Trace Macrocell* (STM).

The DEM performs a 4:1 expansion of the DTB onto the **STMHWEVENT** interface, converting the 8-bit DTB into a 32-bit **STMHWEVENT** interface. This expansion is required for the following reasons:

- The CCN-502 runs at twice the frequency of the external STM. This requires a 2:1 expansion to prevent frequency mismatch.
- The STM HEOI is an edge-triggered interface, requiring a *Return-To-Zero* (RTZ) after assertion of an event. Therefore, it takes two cycles to transmit a single DTB event on the **STMHWEVENT** interface, requiring a second 2:1 expansion for the RTZ protocol.

The DTB is converted to **STMHWEVENT** with each DTB bit creating four adjacent **STMHWEVENT** bits as follows:

- **STMHWEVENT[0]** = DTB[0](cycle# % 4).
- **STMHWEVENT[1]** = DTB[0](cycle# % 4 + 1).
- **STMHWEVENT[2]** = DTB[0](cycle# % 4 + 2).
- **STMHWEVENT[3]** = DTB[0](cycle# % 4 + 3).

Therefore, groups of four **STMHWEVENT** bits correspond to a single DTB bit as follows:

- DTB[0] to **STMHWEVENT[3:0]**.
- DTB[1] to **STMHWEVENT[7:4]**.
- DTB[2] to **STMHWEVENT[11:8]**.
- DTB[3] to **STMHWEVENT[15:12]**.
- DTB[4] to **STMHWEVENT[19:16]**.
- DTB[5] to **STMHWEVENT[23:20]**.
- DTB[6] to **STMHWEVENT[27:24]**.
- DTB[7] to **STMHWEVENT[31:28]**.

If you do not want to run the external STM hardware at a strict 2:1 frequency division of the CCN-502 input clock, the DEM includes a **DCLKEN** input pin that is functionally identical to the **ACLKEN*** clock enable input used on the CCN-502 AMBA interfaces. This creates a synchronous clock that can run at 2:1, 3:1, or 4:1 to the CCN-502 input clock, **GCLK0**.

You cannot use **DCLKEN** to create a 1:1 frequency clock control. **STMHWEVENT** must always operate at 2:1 or lower to the CCN-502 clock frequency. However, for any frequencies below 2:1, multiple DTB events might be received in any given **DCLKEN** clock cycle. In this case, it is impossible to retain the uniqueness of these multiple events, and instead events are sticky in a given **DCLKEN** cycle in the conversion from DTB to **STMHWEVENT**. This is known as event collapsing, and means that any DTB event that occurs is guaranteed to be seen on the **STMHWEVENT** interface after the subsequent **DCLKEN** assertion, but the precise timing and uniqueness of that event is not guaranteed except in the case of 2:1 **DCLKEN**.

Although the **STMHWEVENT** interface is useful even when event collapsing occurs and without totally accurate globally-timed event tracing, you must be aware of the limitations of trace capabilities in such a configuration.

The **STMHWEVENT** outputs for a given DTB bit are produced as a shift register. The DTB bits are shifted into staging flops each cycle, then moved to the output flops on a **DCLKEN** pulse. Event collapsing is used for the oldest events, because the oldest staging flop includes an OR function at the input. The following tables show this behavior.

Table 5-2 2:1 DTB-to-STMHWEVENT conversion

Cycle	DTB[0]	DCLKEN	Staging				STMHWEVENT			
			[0]	[1]	[2]	[3]	[0]	[1]	[2]	[3]
i	C	1	-[c]	A	B	C	0	0	0	0
i+1	D	0	A	B	C	D	0	0	0	0
i+2	E	1	0	0	0	E	A	B	C	D
i+3	F	0	0	0	E	F	A	B	C	D
i+4	G	1	0	E	F	G	0	0	0	0
i+5	H	0	E	F	G	H	0	0	0	0
i+6	I	1	0	0	0	I	E	F	G	H
i+7	J	0	0	0	I	J	E	F	G	H

---

[c] The data is not valid.

Table 5-2 2:1 DTB-to-STMHWEVENT conversion (continued)

Cycle	DTB[0]	DCLKEN	Staging				STMHWEVENT			
			[0]	[1]	[2]	[3]	[0]	[1]	[2]	[3]
i+8	K	1	0	I	J	K	0	0	0	0
i+9	L	0	I	J	K	L	0	0	0	0
i+10	M	1	0	0	0	M	I	J	K	L
i+11	N	0	0	0	M	N	I	J	K	L
i+12	O	1	0	M	N	O	0	0	0	0
i+13	P	0	M	N	O	P	0	0	0	0

Table 5-3 3:1 DTB-to-STMHWEVENT conversion

Cycle	DTB[0]	DCLKEN	Staging				STMHWEVENT			
			[0]	[1]	[2]	[3]	[0]	[1]	[2]	[3]
i	D	0	A	B	C	D	0	0	0	0
i+1	E	1	A|B	C	D	E	0	0	0	0
i+2	F	0	A|B|C	D	E	F	0	0	0	0
i+3	G	0	0	0	0	G	A|B|C	D	E	F
i+4	H	1	0	0	G	H	A|B|C	D	E	F
i+5	I	0	0	G	H	I	A|B|C	D	E	F
i+6	J	0	G	H	I	J	0	0	0	0
i+7	K	1	G|H	I	J	K	0	0	0	0
i+8	L	0	G|H|I	J	K	L	0	0	0	0
i+9	M	0	0	0	0	M	G|H|I	J	K	L
i+10	N	1	0	0	M	N	G|H|I	J	K	L
i+11	O	0	0	M	N	O	G|H|I	J	K	L
i+12	P	0	M	N	O	P	0	0	0	0
i+13	Q	1	M|N	O	P	Q	0	0	0	0
i+14	R	0	M|N|O	P	Q	R	0	0	0	0
i+15	S	0	0	0	0	S	M|N|O	P	Q	R
i+16	T	1	0	0	S	T	M|N|O	P	Q	R
i+17	U	0	0	S	T	U	M|N|O	P	Q	R

Table 5-4 4:1 DTB-to-STMHWEVENT conversion

Cycle	DTB[0]	DCLKEN	Staging				STMHWEVENT			
			[0]	[1]	[2]	[3]	[0]	[1]	[2]	[3]
i	E	0	A|B	C	D	E	0	0	0	0
i+1	F	0	A|B|C	D	E	F	0	0	0	0
i+2	G	1	A|B|C|D	E	F	G	0	0	0	0
i+3	H	0	A|B|C|D|E	F	G	H	0	0	0	0
i+4	I	0	0	0	0	I	A|B|C|D|E	F	G	H

Table 5-4 4:1 DTB-to-STMHWEVENT conversion (continued)

Cycle	DTB[0]	DCLKEN	Staging				STMHWEVENT			
			[0]	[1]	[2]	[3]	[0]	[1]	[2]	[3]
i+5	J	0	0	0	I	J	A|B|C|D|E	F	G	H
i+6	K	1	0	I	J	K	A|B|C|D|E	F	G	H
i+7	L	0	I	J	K	L	A|B|C|D|E	F	G	H
i+8	M	0	I|J	K	L	M	0	0	0	0
i+9	N	0	I|J|K	L	M	N	0	0	0	0
i+10	O	1	I|J|K|L	M	N	O	0	0	0	0
i+11	P	0	I|J|K|L|M	N	O	P	0	0	0	0
i+12	Q	0	0	0	0	Q	I|J|K|L|M	N	O	P
i+13	R	0	0	0	Q	R	I|J|K|L|M	N	O	P
i+14	S	1	0	Q	R	S	I|J|K|L|M	N	O	P
i+15	T	0	Q	R	S	T	I|J|K|L|M	N	O	P
i+16	U	0	Q|R	S	T	U	0	0	0	0
i+17	V	0	Q|R|S	T	U	V	0	0	0	0
i+18	W	1	Q|R|S|T	U	V	W	0	0	0	0
i+19	X	0	Q|R|S|T|U	V	W	X	0	0	0	0
i+20	Y	0	0	0	0	Y	Q|R|S|T|U	V	W	X
i+21	Z	0	0	0	Y	Z	Q|R|S|T|U	V	W	X
i+22	A	1	0	Y	Z	A	Q|R|S|T|U	V	W	X
i+23	B	0	Y	Z	A	B	Q|R|S|T|U	V	W	X

The trace capability is a conversion of the DTB, so you can trace CCN-502 performance monitoring events rather than debug events, because the PMU events are sent over the DTB to the DEM.

Related concepts

Clock enable inputs on page 2-65.

5.4.3 DEM PMU capabilities

The DEM contains all the PMU event counting infrastructure. It contains eight 32-bit PMU event counters and a single 40-bit cycle counter.

All PMU events are routed from multiple CCN-502 components over the DTB to the DEM, where all the global performance event counting is performed locally in the DEM.

You can optionally configure each even-aligned register pair, 0/1, 2/3, 4/5, and 6/7, on a pair-by-pair basis to act as a single combined 64-bit counter, with overflows from the least significant register causing counting in the most significant register.

There are eight DTB bits and eight PMU counters. This provides a one-to-one correspondence between the DTB bit and the PMU counter, and requires no additional multiplexing between the DTB and the PMU counters. However, for register pairs, the least significant register counts events on its normal corresponding DTB input, but the most significant register ignores its corresponding DTB input and instead counts overflows from the least significant register.

You can snapshot the PMU registers for greater accuracy in counter and event collection from an outside source. When a snapshot request is made, the CCN-502 copies all live counters into shadow copies, that software or hardware can read without interrupting the live counting functionality.

The DEM includes hardware and software control of the snapshot request activity:

- The hardware control is through a pair of external signals, **PMUSNAPSHOTREQ** and **PMUSNAPSHOTACK**. These signals are asynchronous-safe and communicate through a traditional 4-phase handshake protocol. When an external agent asserts **PMUSNAPSHOTREQ**, the DEM copies the live counters to shadow counters, and asserts **PMUSNAPSHOTACK** to indicate that the process is complete. The external agent can then read the shadow copies.

- The software control is enabled by an external agent writing a 1 to the pmsr_req bit in the pmsr_req register. This causes the DEM to copy the live PMU counters to the shadow copies, and the ss_status bit in the pmsr register is set. When ss_status==1, the external software agent can read the PMU shadow copies without interrupting the live copies.

For both snapshot request control mechanisms, the DEM enables optional clearing of the live counters after they are copied to the shadow copies. This simplifies the snapshotting and counting process in a snapshot system.

Because the PMU counters count events sent over the DTB, you can use the PMU counters to count debug events, such as functional watchpoint matches, instead of traditional performance events.

The DEM detects overflow of any of the nine PMU counters, and logs an overflow indicator in the pmovsr register. In addition, the DEM can optionally cause the CCN-502 interrupt, **INTREQ**, to be asserted on PMU overflow, to enable software to handle the overflow condition.

Related concepts

5.4.2 DEM trace capabilities on page 5-236.

Related references

PMU Control register on page 3-186.
PMU Status register on page 3-187.
PMU Overflow Status Clear register on page 3-186.

Related concepts

5.4.1 DEM trigger capabilities on page 5-236.
5.4.2 DEM trace capabilities on page 5-236.
5.4.3 DEM PMU capabilities on page 5-239.

5.5   Security and DT enable

The CCN-502 includes two signals that provide Secure access to the debug, trace, and PMU capabilities.

These signals are:

NIDEN   A global enable for all debug, trace, and PMU functionality. This overrides all other software enable controls.

SPNIDEN   A global enable for Secure debug, trace, and PMU. It is only applicable when **NIDEN** is asserted.

Deasserting **SPNIDEN** prevents Secure event counting wherever possible. Where a performance event cannot be determined to be Secure or Non-secure, that event is considered as Secure.

———————— **Note** ————————

The secure_debug_disable bit of the MN secure_access register is used to override the **SPNIDEN** filtering. When secure_debug_disable is 0, which means Secure debug disable is off, all events, both Secure and Non-secure, are counted by the PMU. The default value of secure_debug_disable is 0.

————————————————

Related references

Secure Access register on page 3-94.

5.6   Watchpoint setup

This section describes how to enable a single watchpoint compare result to be correctly configured at the watchpoint and transferred over the DTB to the DEM, where it can potentially cause a **DBGWATCHTRIGREQ** assertion.

To enable a watchpoint, complete the following procedure:

Procedure

1. Set up the DEM output:

    a. In the Active DSM register, active_dsm, write the XP ID of the XP driving the signals, for precise event timing.

    b. Write to the Trigger Control register, trigger_ctl, to select which DTB bits are included in the **DBGWATCHTRIGREQ** assertion.

    c. In the Timer Value register, timer_val, write the delay between the DTB event and the assertion of **DBGWATCHTRIGREQ**, if required.

2. Set up the DWM in the XP:

    a. In the Debug and Trace Configuration register, dt_config, select watchpoint 0 or 1 and which DTB bit to drive on the XP originating the DTB event. You must do this on all intervening XPs to ensure that driving and pass-through are as required at the DEM.

    b. In the Debug and Trace Interface Select register, dt_interface_sel, select:
    -   XP device port 0 or 1.
    -   Channel type. This can be REQ, SNP, RSP, or DATA.
    -   Direction, either transmit or receive, on which the watchpoint compare is active.

    c. In the Debug and Trace Comparison Low Value * register, dt_cmp_val*_l, and the Debug and Trace Comparison High Value * register, dt_cmp_val*_h, write the value for fields to be compared in the watchpoint.

    d. In the Debug and Trace Comparison Low Mask * register, dt_cmp_mask*_l, or the Debug and Trace Comparison High Mask * register, dt_cmp_mask*_h register, write the mask to determine which flit-fields are compared in the watchpoint.

    e. In the Debug and Trace Control register, dt_control, set up snapshotting of flit contents on watchpoint match, if required.

3. Enable the watchpoint or trigger:

    a. Write to the trigger_ctl register to enable **DBGWATCHTRIGREQ** generation.

    b. Write to the dt_control register to enable the relevant watchpoints for all XPs, starting at XP5 and progressing in a clockwise direction.

4. When a DTB event occurs:

    Trigger indication is delivered by **DBGWATCHTRIGREQ**.

    a. The responder reads the Trigger Status register, trigger_status, to identify which DTB bits contributed to the **DBGWATCHTRIG** assertion.

    b. The responder reads out the flit information stored in the dt_cmp_val*_l and dt_cmp_val*_h registers. This is the first flit that triggered the watchpoint match.

Related references

Active DSM register on page 3-172.

Trigger Control register on page 3-173.

Timer Value register on page 3-174.

Debug and Trace Configuration register on page 3-121.

Debug and Trace Interface Select register on page 3-122.

Debug and Trace Comparison Low Value 0 register on page 3-123.

Debug and Trace Comparison High Value 0 register on page 3-124.

Debug and Trace Comparison Low Value 1 register on page 3-127.

Debug and Trace Comparison High Value 1 register on page 3-128.

Debug and Trace Comparison Low Mask 0 register on page 3-125.

Debug and Trace Comparison High Mask 0 register on page 3-126.

Debug and Trace Comparison Low Mask 1 register on page 3-129.

Debug and Trace Comparison High Mask 1 register on page 3-130.

Debug and Trace Control register, dt_control on page 3-131.

Trigger Status register on page 3-173.

5.7 Example PMU setup

Two PMU events can be counted at the DEM.

You can:

- Use performance counters within an HN-F or RN-I to count activity within that HN-F or RN-I.
- Use the XP watchpoint features to count activity that is passing through an XP.

The HN-F, RN-I, and XP PMU features can be used independently or simultaneously. You can set up just one or the other, or multiple instances of each from different HN-Fs, RN-Is, and XPs, driving onto different DTBus bits.

The following is an example of the steps that are used to enable the PMU events to be counted. The example:

- Uses performance counters within an HN-F to count activity within that HN-F.
- Uses the XP watchpoint features to count activity that is passing through an XP.

Procedure

1. Select the performance event at the component:
   - Write the pmu_event0_id field in the HN-F pmu_event_sel register to select PMU_HN_CACHE_MISS_EVENT on bit 0 of the HN-F PMU interface.
   - Write the dt_config register to write this event on DTB[0].

2. Select the watchpoint in the XP:
   - Write the dt_config register to select the watchpoint to drive DTB[1].
   - In the dt_interface_sel register, select:
     — XP device port 0 or 1.
     — Channel type, which can be REQ, SNP, RSP, or DATA.
     — Direction, either transmit or receive, on which the watchpoint compare is active.
   - In the dt_cmp_val*_l and dt_cmp_val*_h registers, write the value for fields to be compared in the watchpoint.
   - In the dt_cmp_mask*_l or dt_cmp_mask*_h register, write the mask to determine which flit-fields are compared in the watchpoint.

3. For each XP, set dt_control.dt_enable = 1, to enable the debug watchpoint and PMU capability in that XP.

4. Program the PMU control:
   - In the pmcr register:
     — Write the cntcfg field to configure as 8×32-bit counters, no pairs.
     — Write the pmu_en bit to enable PMU counting.
     — Write 1 to pmsr_req to enable PMU counter snapshot.
   - Read the pmevcntsr0 register for the HN-F event counter.
   - Read the pmevcntsr1 register for the watchpoint counter.

Related references

PMU Event Select register, L3 cache on page 3-162.

Debug and Trace Configuration register on page 3-121.

Debug and Trace Interface Select register on page 3-122.

Debug and Trace Comparison Low Value 0 register on page 3-123.

Debug and Trace Comparison High Value 0 register on page 3-124.

Debug and Trace Comparison Low Value 1 register on page 3-127.

Debug and Trace Comparison High Value 1 register on page 3-128.

Debug and Trace Comparison Low Mask 0 register on page 3-125.

Debug and Trace Comparison High Mask 0 register on page 3-126.

Debug and Trace Comparison Low Mask 1 register on page 3-129.

Debug and Trace Comparison High Mask 1 register on page 3-130.

PMU Control register on page 3-186.

PMU Event Counter Shadow 0 register on page 3-181.

PMU Event Counter Shadow 1 register on page 3-181.

Chapter 6
Performance Optimization and Monitoring

This chapter describes performance optimization techniques for use by system integrators, and the *Performance Monitoring Unit* (PMU).

It contains the following sections:

6.1 Performance optimization guidelines

There are some restrictions when optimizing the CCN-502.

To obtain maximum performance from the CCN-502, the system integrator must be aware of the following information:

RN-I   When ordering is not required, transaction requests must be dispatched with non-overlapping IDs to ensure optimal bandwidth operation. Large burst transactions, that is, larger than 64B, must be split into 64B or smaller burst transactions. In addition, set **AxSIZE** to 4 (16B) to fully utilize the available bandwidth.

Set the *WriteUnique Optimization* (wuo) configuration register bit to optimize performance for ordered WriteUnique streaming operations.

——————— **Note** ———————

- In systems where a PCIe *Root Complex* (RC) is present, the wuo bit must be set in the RN-I instance that connects to the RC and only in that RN-I instance. Clear the wuo bit in the other RN-I instances.
- When the wuo bit is set, WriteNoSnp operations targeting the same HN partition are ordered even when their IDs are non-overlapping.

————————————————

Read or write requests to different parts of the same cache line must be combined into a single cache line request. For example, multiple (partial) WriteUnique transactions must be combined into a single WriteUnique or a single WriteLineUnique transaction, where all bytes in the cache line are written.

All transactions that the RN-I sends to the HN-I have the CHI ReqOrder bit set, and the maximum achievable bandwidth is affected accordingly.

HN-F,   High temporal locality of address usage in transactions can cause same-address dependencies to
HN-I   occur in the event of transactions with addresses to overlapping cache lines. This results in higher latency because of serialization delays between these transactions. The CCN-502 is microarchitected to avoid hotspotting in the HN-F partitions or in the memory controllers, but this is unavoidable in cases of temporally-local same-address usage.

Related references

6.2 About the Performance Monitoring Unit

The CCN-502 provides access to a number of performance events. Some of these events are unique to and originate in a specific CCN-502 component, and some are available by using watchpoints in the *Debug Watchpoint Module* (DWM).

This chapter describes the performance events and the relevant use cases for most of those events. See *Chapter 5 Debug* on page 5-230 for information on the infrastructure and logic that enable general utility of the performance monitor events.

The following table shows the PMU events.

Table 6-1 PMU events

Component	NS [d]	Event	Description
MN	No	PMU_MN_EOBARRIER	EOBarrier count. Available through the DWM.
	No	PMU_MN_ECBARRIER	ECBarrier count. Available through the DWM.
	Yes	PMU_MN_DVMOP	DVMOp count. Available through the DWM.
HN-I	No	PMU_HNI_TXDATFLITV	Transmitted data flits. Available through the DWM.
	No	PMU_HNI_RXDATFLITV	Received data flits. Available through the DWM.
	Yes	PMU_HNI_RXREQFLITV	Received requests. Available through the DWM.
	Yes	PMU_HNI_RXREQ_REQORDER	Received ReqOrder requests. Available through the DWM.
SBSX	No	PMU_SBSX_TXDATFLITV	Transmitted data flits. Available through the DWM.
	No	PMU_SBSX_RXDATFLITV	Received data flits. Available through the DWM.
	Yes	PMU_SBSX_RXREQFLITV	Received requests. Available through the DWM.
HN-F	Yes	PMU_HN_CACHE_MISS	Total cache misses.
	Yes	PMU_HNL3_SF_CACHE_ACCESS	Total number of cache accesses.
	Yes	PMU_HN_CACHE_FILL	Total allocations in HN L3 cache.
	Yes	PMU_HN_POCQ_RETRY	Total number of requests that have been retried.
	Yes	PMU_HN_POCQ_REQS_RECVD	Total number of requests received by the HN.
	Yes	PMU_HN_SF_HIT	Total number of snoop filter hits.
	Yes	PMU_HN_SF_EVICTIONS	Total number of snoop filter evictions.
	Yes	PMU_HN_SNOOPS_SENT	Number of snoops sent. Does not differentiate between broadcast or directed snoops.
	Yes	PMU_HN_SNOOPS_BROADCAST	Number of snoop broadcasts sent.
	Yes	PMU_HN_L3_EVICTION	Number of L3 evictions.
	Yes	PMU_HN_L3_FILL_INVALID_WAY	Number of L3 fills to an invalid way.
	Yes	PMU_HN_MC_RETRIES	Number of requests receiving retry response from the memory controller.
	Yes	PMU_HN_MC_REQS	Total number of requests that are sent to the memory controller.
	Yes	PMU_HN_QOS_HH_RETRY	Number of times HN-F protocol retried a QoS 15 (highest) class request.

---

[d] Can the event be determined to be Secure or Non-secure? If No, the event is considered to be Secure, irrespective of Secure or Non-secure attributes associated with the event.

Table 6-1 PMU events (continued)

Component	NS ^d	Event	Description
XP	No	PMU_XP_UPLOAD_STARVATION	Upload starvation. Signaled when this XP sets the H-bit, per-channel, per-direction.
	No	PMU_XP_DOWNLOAD_STARVATION	Download starvation. Signaled when this XP sets the S-bit, per-channel, per-direction.
	No	PMU_XP_RESPIN	Respin. Signaled when this XP sets the P-Cnt, per-channel, per-direction.
	No	PMU_XP_VALID_FLIT	A valid flit is passing through the XP, per-channel, per-direction.
RN-I	No	PMU_RNI_RDATABEATS_P0	S0 RDataBeats.
	No	PMU_RNI_RDATABEATS_P1	S1 RDataBeats.
	No	PMU_RNI_RDATABEATS_P2	S2 RDataBeats.
	Yes	PMU_RNI_RXDATFLITV	**RXDAT** flits received.
	Yes	PMU_RNI_TXDATFLITV	**TXDAT** flits sent.
	Yes	PMU_RNI_TXREQFLITV	Total **TXREQ** flits sent.
	Yes	PMU_RNI_TXREQFLITV_RETRIED	Retried **TXREQ** flits sent.
	No	PMU_RNI_RRTFULL	Read request tracker full.
	No	PMU_RNI_WRTFULL	Write request tracker.
	Yes	PMU_RNI_TXREQFLITV_REPLAYED	Replayed **TXREQ** flits.

6.2.1   Cycle counter

The cycle counter is used to track the time.

You can reset this counter to initiate the time interval over which you want to capture the events.

PMU_CYCLE_COUNTER   Cycle counter.

Because the cycle counter is clocked by **GCLK0**, it is not incremented during periods of *High-level Clock Gating* (HCG) when the clocks are stopped.

---

d   Can the event be determined to be Secure or Non-secure? If No, the event is considered to be Secure, irrespective of Secure or Non-secure attributes associated with the event.

6.3 HN-F performance events

The HN-F performance analysis counters are used to monitor cache behavior.

For a particular cache, the cache miss or hit rate is used to measure the capacity of the cache, and the location for certain applications. To measure the cache miss rate, the performance monitor counters count the number of instances of cache accesses and cache misses.

This section contains the following subsections:

6.3.1 Cache performance

Cache performance events are required to calculate the cache miss rate and the cache allocation.

The following sections describe the cache performance events.

Cache miss rate

The cache events that are required to calculate the cache miss rate are:

PMU_HN_CACHE_MISS_EVENT Counts the total cache misses. This is a first-time lookup result, and is high priority.

PMU_HNL3_SF_CACHE_ACCESS_EVENT The total number of cache accesses. These are first-time accesses, and are high priority.

————— Note —————

The performance counter architecture enables only four HNs to collect the cache miss rate. However, due to the CCN-502 microarchitecture, the cache miss rate that is measured at one HN-F is a good proxy for the cache miss rate of the remaining HN-Fs.

————————————

Calculate the cache miss rate as follows:

$$\text{Cache miss rate (\%)} = \frac{\text{Total cache misses}}{\text{Total cache accesses}} \times 100$$

Certain request types can cause multiple cache accesses:
- Lookup.
- Tag update.
- Victim selection.
- Cache fill.

Event counting is therefore limited to first time accesses only. For example, for a ReadUnique transaction that leads to an L3 hit, PMU_HN_CACHE_ACCESS_EVENT is only counted the first time a cache lookup is performed. The tag update is not counted as a cache access. Similarly, for WriteBack or Write*Unique transactions with an L3 allocate hint, only the first instance of an L3 lookup is counted as an access and hit or miss. The eventual victim selection and cache fill are not counted as additional accesses.

Cache allocations

The cache allocation event counts the number of times an HN-F L3 cache is allocated. It provides an approximate cache usage for this particular application over a specific time slice. This event does not check whether the application has any hot sets.

PMU_HN_CACHE_FILL_EVENT   Counts all cache line allocations to L3 cache.

All cache line writes, that is, Write*Unique, WriteBack, and Evictions that are allocated in L3 cache, are counted towards this event.

6.3.2   HN-F counters

Applications can bottleneck on one or more HN-Fs because they frequently target an address or a stream of addresses.

The following POCQ occupancy and request retry events are used to monitor possible performance loss in the system:

PMU_HN_POCQ_RETRY_EVENT   The total number of requests that have been retried.

PMU_HN_POCQ_REQS_RECVD_EVENT   The total number of requests that the HN-F receives.

Requests that cannot be queued in the POCQ, because of lack of credits, are retried. The HN-F responds with a RetryAck response, and the request waits for a static credit. This indicates whether the bottlenecks are caused by a lack of credits, and also shows if the latency of requests is very high.

Calculate the message retry rate as follows:

$$\text{HN-F message retry rate (\%)} = \frac{\text{HN-F total messages retried}}{\text{HN-F total messages received}} \times 100$$

6.3.3   Snoop filter events

There are three snoop events that can be counted.

The following sections describe the snoop filter performance events.

Snoop filter miss rate

This event measures the amount of memory controller traffic that is generated. It can also be used to measure the efficiency of the snoop filter.

PMU_HN_SF_HIT_EVENT   Measures the number of snoop filter hits.

Calculate the snoop filter hit rate as follows:

$$\text{Snoop filter hit rate (\%)} = \frac{\text{Total snoop filter hits}}{\text{Total L3 lookups}} \times 100$$

Snoop filter accesses are only counted for first-time lookups, and not for the victim selection accesses or snoop filter fills. Because the L3 lookup and snoop filter lookups are parallel, the L3 lookups can be used to calculate the snoop filter hit rate.

Snoop filter evictions

This event measures the frequency of snoop filter evictions, and determines the DEQ size.

PMU_HN_SF_EVICTIONS_EVENT   Measures the number of snoop filter evictions when cache invalidations are initiated.

Snoops sent and received with hit rate

This event measures the amount of shared data across clusters for a specific application, using snoops hits or misses.

PMU_HN_SNOOPS_SENT_EVENT   Number of snoops sent. Does not differentiate between broadcast or directed snoops.

PMU_HN_SNOOPS_BROADCAST_EVENT   Number of snoop broadcasts sent.

Calculate the snoops sent and received rate as follows:

$$\text{Shared data (\%)} = \frac{\text{Total snoops broadcast}}{\text{Total snoops sent}} \times 100$$

The number of broadcast and total snoops measures the shared data invalidations.

6.3.4 System-wide events

The memory controller request retries determine whether the memory controller is the bottleneck in the system, which can cause higher request latencies.

The following events can be counted:

PMU_HN_MC_RETRIES_EVENT   Number of requests that are retried to the memory controller.

PMU_HN_MC_REQS_EVENT   Total number of requests that are sent to the memory controller.

Calculate the retry rate for requests to the memory controller as follows:

$$\text{MC message retry rate (\%)} = \frac{\text{MC total messages retried}}{\text{MC total messages received}} \times 100$$

6.3.5 Quality of Service

Requests with a HighHigh QoS must be allocated and processed from the POCQ with the highest priority compared to High, Medium, and Low QoS requests.

If the HighHigh requests are retried too frequently, there could be a bottleneck at a particular HN-F, or the POCQ reservation for HighHigh requests requires adjustment.

PMU_HN_QOS_HH_RETRY   How often a HighHigh request is retried.

6.3.6 HN-F PMU event summary

The HN-F PMU events are summarized in a table.

The following table shows a summary of the HN-F PMU events.

Table 6-2 HN-F PMU event summary

Number	Name	Description
1	PMU_HN_CACHE_MISS_EVENT	The number of cache misses.
2	PMU_HNL3_SF_CACHE_ACCESS_EVENT	The number of cache accesses.
3	PMU_HN_CACHE_FILL_EVENT	The number of allocations in HN-F L3 cache.
4	PMU_HN_POCQ_RETRY_EVENT	The number of requests that have been retried.
5	PMU_HN_POCQ_REQS_RECVD_EVENT	The number of requests received by the HN-F.
6	PMU_HN_SF_HIT_EVENT	The number of snoop filter hits.
7	PMU_HN_SF_EVICTIONS_EVENT	The number of snoop filter evictions.
8	PMU_HN_SNOOPS_SENT_EVENT	The number of snoops sent. Does not differentiate between broadcast or directed snoops.
9	PMU_HN_SNOOPS_BROADCAST_EVENT	The number of snoop broadcasts sent.
10	PMU_HN_MC_RETRIES_EVENT	The number of requests that retried to the memory controller.
11	PMU_HN_MC_REQS_EVENT	The number of requests sent to the memory controller.
12	PMU_HN_QOS_HH_RETRY	How often a HighHigh QoS request is retried.

6.4 RN-I performance events

This section contains the following subsections:

6.4.1 Bandwidth at RN-I bridges

The following events measure bandwidth at the RN-I bridges:

Requested read bandwidth at RN-I bridges

External devices connect to a CCN-502 at an RN-I bridge.

To monitor the behavior of the system, the following events measure the read bandwidth at each RN-I bridge:

RDataBeats_Port0   Number of RData beats, **RVALID** and **RREADY**, dispatched on port 0. This is a measure of the read bandwidth.

RDataBeats_Port1   Number of RData beats, **RVALID** and **RREADY**, dispatched on port 1. This is a measure of the read bandwidth.

RDataBeats_Port2   Number of RData beats, **RVALID** and **RREADY**, dispatched on port 2. This is a measure of the read bandwidth.

Because CMOs are sent through the read channel, their responses are included in these events.

Calculate the read bandwidth as follows:

$$\text{Read bandwidth} = \frac{\text{Number RDataBeats\_Port}n \text{ x AXIDataBeatSize}}{\text{Cycles}} \text{ x Frequency}$$

Where AXIDataBeatSize is the number of bytes for each AXI beat. In most cases, this is the same size as the AXI bus.

Actual read bandwidth on interconnect

RXDATFLITV measures the bandwidth that an RN-I bridge sends to the interconnect.

To measure the actual bandwidth that an RN-I bridge sends to the interconnect, and not the useful bandwidth the external devices can use, this event counts the number of received data flit requests that the bridge receives through the data channel:

RXDATFLITV   Number of **RXDAT** flits received. This event is a measure of the true read data bandwidth. It excludes CMOs, because CMO completions return to the RN-I through the response channel, but includes replayed requests.

This event includes the replayed requests because of the read data buffer decoupled scheme.

Calculate the actual read bandwidth as follows:

$$\text{Actual read bandwidth} = \frac{\text{RXDATFLITV x DataFlitSize}}{\text{Cycles}} \text{ x Frequency}$$

Write bandwidth at RN-I bridges

TXDATFLITV monitors the number of data flits that the RN-I bridge sends out.

In a similar way to the read actual bandwidth event, this event monitors the number of data flits that the RN-I bridge sends out, to measure the actual write bandwidth that is sent to the interconnect:

TXDATFLITV   Number of **TXDAT** flits dispatched. This event is a measure of the write bandwidth.

Calculate the write bandwidth as follows:

$$\text{Actual write bandwidth} = \frac{\text{TXDATFLITV x DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

Total requested bandwidth at RN-I bridges

To improve efficiency when using PMU events and signals, TXREQFLITV_TOTAL combines the read and write bandwidth estimation in a single event.

TXREQFLITV_TOTAL achieves this by monitoring the number of request flits that are sent from an RN-I bridge:

TXREQFLITV_TOTAL   Number of **TXREQ** flits dispatched. This event is a measure of the total request bandwidth.

To use this event correctly, you must know the average request data size for both reads and writes in your system. If the AXI masters issue a mixture of request sizes, you must estimate the average size of read and writes, using the PMU in an AXI master or an AXI interrupt controller.

Calculate the total bandwidth as follows:

$$\text{Total requested bandwidth} = \frac{\text{TXREQFLITV\_TOTAL x Avg.DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

6.4.2   Bottleneck analysis at RN-I bridges

The CCN-502 provides events that observe the locations where the nodes or bridges are full, which can cause delays in the rest of the system.

This enables you to monitor the current bottlenecks in the system, and checks multiple events in the RN-Is, HN-Fs, and memory controllers. In the RN-I bridges, the events monitor the following:
- The number of times the bridge is forced to retry because of the lack of dynamic credits.
- The number of times the read and write tracker is full and therefore cannot accept new requests in the system. This can cause delays in the AXI masters.
- The number of read request replays, because of decoupling of the read request buffers and read data buffers in the RN-I system.

Request retry rate at RN-I bridges

TXREQFLITV_RETRIED monitors the efficiency of using dynamic credits in the system.

It does this by measuring the request retry rate:

TXREQFLITV_RETRIED   Number of retried **TXREQ** flits dispatched. This event is a measure of the retry rate.

Calculate the request retry rate as follows:

$$\text{Retry rate} = \frac{\text{TXREQFLITV\_RETRIED}}{\text{TXREQFLITV\_TOTAL}}$$

Read and write delays at RN-I bridges

To monitor the delays for both reads and writes, the CCN-502 enables you to monitor how full the read and write trackers are in the RN-I bridges.

When one of the trackers is full, the bridge cannot accept new requests from the AXI master. This delays the I/O devices that connect to the AXI master.

You can use the measure of how full the trackers are, together with the read and write bandwidth from the RN-I bridge to the interconnect, to help isolate the source of bottlenecks in the system. For example:

- If the read tracker of a specific RN-I bridge is full but the effective read bandwidth from the bridge is not close to the maximum expected, the interconnect cannot keep up with the read traffic from the specific device.
- If the bandwidth is close to maximum, the I/O device can send requests to the maximum of its port bandwidth and this is why the tracker is full.

You can also use the measure of how full the trackers are with AXI PMUs to monitor delays to the AXI masters.

The following events monitor the read and write trackers:

RRTFull   All entries in the read request tracker, excluding those reserved for Hi-QPC, are occupied. This is a measure of oversubscription in the read request tracker.

WRTFull   All entries in the write request tracker, excluding those reserved for Hi-QPC, are occupied. This is a measure of oversubscription in the write request tracker.

6.4.3 RN-I PMU event summary

There are nine RN-I PMU events.

The following table shows a summary of the RN-I PMU events.

Table 6-3 RN-I PMU event summary

Number	Name	Description
1	PMU_RNI_RDATABEATS_P0	Number of RData beats, **RVALID** and **RREADY**, dispatched on port 0. This is a measure of the read bandwidth, including CMO responses.
2	PMU_RNI_RDATABEATS_P1	Number of RData beats, **RVALID** and **RREADY**, dispatched on port 1. This is a measure of the read bandwidth, including CMO responses.
3	PMU_RNI_RDATABEATS_P2	Number of RData beats, **RVALID** and **RREADY**, dispatched on port 2. This is a measure of the read bandwidth, including CMO responses.
4	PMU_RNI_RXDATFLITV	Number of **RXDAT** flits received. This is a measure of the true read data bandwidth, excluding CMOs.
5	PMU_RNI_TXDATFLITV	Number of **TXDAT** flits dispatched. This is a measure of the write bandwidth.
6	PMU_RNI_TXREQFLITV	Number of **TXREQ** flits dispatched. This is a measure of the total request bandwidth.
7	PMU_RNI_TXREQFLITV_RETRIED	Number of retried **TXREQ** flits dispatched. This is a measure of the retry rate.
8	PMU_RNI_RRTFULL	All entries in the read request tracker, excluding those reserved for Hi-QPC, are occupied. This is a measure of oversubscription in the read request tracker.
9	PMU_RNI_WRTFULL	All entries in the write request tracker, excluding those reserved for Hi-QPC, are occupied. This is a measure of oversubscription in the write request tracker.

6.5 SBSX and HN-I performance events

This section contains the following subsections:

6.5.1 Bandwidth at SBSX and HN-I bridges

The following events are used to measure bandwidth at the SBSX and HN-I bridges:

Read bandwidth on interconnect at SBSX and HN-I bridges

This event counts the number of received data flits at the SBSX, HN-I, and interconnect:

TXDAT   Number of **TXDAT** flits received. This event is a measure of the read data bandwidth.

Calculate the actual read bandwidth as follows:

$$\text{Actual read bandwidth} = \frac{\text{TXDAT x DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

————— **Note** —————
This event is tracked in the DWM, not in the SBSX or HN-I design.

Write bandwidth at SBSX and HN-I bridges

In a similar way to the read actual bandwidth event, this event monitors the number of data flits that the SBSX and HN-I bridges send out, to measure the actual write bandwidth that is received from the interconnect:

RXDAT   Number of **RXDAT** flits dispatched. This event is a measure of the write bandwidth.

Calculate the write bandwidth as follows:

$$\text{Actual write bandwidth} = \frac{\text{RXDAT x DataFlitSize}}{\text{Cycles}} \times \text{Frequency}$$

————— **Note** —————
This event is tracked in the DWM, not in the SBSX or HN-I design.

Total requested bandwidth at SBSX and HN-I bridges

To improve efficiency when using PMU events and signals, this event combines the read and write bandwidth estimation in a single event by monitoring the number of request flits that an SBSX or HN-I bridge receive:

RXREQ_TOTAL   Number of **RXREQ** flits dispatched. This event is a measure of the total request bandwidth.

Calculate the total bandwidth as follows:

$$\text{Total requested bandwidth} = \frac{\text{RXREQ\_TOTAL x AvgDataFlitSize}}{\text{Cycles}} \text{x Frequency}$$

——— **Note** ———

This event is tracked in the DWM, not in the SBSX or HN-I design.

—————————————

6.5.2 Bottleneck analysis at SBSX and HN-I bridges

The CCN-502 provides events that observe the locations where the nodes or bridges are full, which can cause delays in the rest of the system. This enables you to monitor the current bottlenecks in the system, and checks multiple events in all CCN-502 components.

The events monitor the following:

- The number of times the bridge is forced to retry because of the lack of dynamic credits.
- The number of requests that have the ReqOrder information set. This event is only applicable in the HN-I.

The following events are used to measure bottlenecks at the SBSX and HN-I bridges:
- *Request retry rate at SBSX and HN-I bridges* on page 6-257.
- *ReqOrder request rate* on page 6-257.

Request retry rate at SBSX and HN-I bridges

Request retries from the SBSX are tracked in the HN-F.

Retry requests from the HN-I are tracked in the RN-I.

Related concepts
6.3.2 HN-F counters on page 6-251.

Related references
Request retry rate at RN-I bridges on page 6-254.

ReqOrder request rate

When requests are received at the HN-I with the ReqOrder bit set, they must maintain order, and are serialized. This event can be used to indicate a lower than expected bandwidth on the HN-I.

TXREQ_REQORDER   Number of requests that the HN-I observes with the ReqOrder bit set. This event is a measure of oversubscription in the read request tracker.

Calculate the ReqOrder request rate as follows:

$$\text{ReqOrder request rate} = \frac{\text{TXREQ\_REQORDER}}{\text{TXREQ\_TOTAL}}$$

——— **Note** ———

This event is tracked in the DWM, not in the SBSX or HN-I design.

—————————————

6.5.3 SBSX and HN-I PMU event summary

The following table shows a summary of the SBSX PMU events.

Table 6-4 SBSX PMU event summary

Number	Name	Description
1	PMU_SBSX_RXDAT	Number of **RXDAT** flits received. This is a measure of the true read data bandwidth.
2	PMU_SBSX_TXDAT	Number of **TXDAT** flits dispatched. This is a measure of the true write data bandwidth.
3	PMU_SBSX_TXREQ_TOTAL	Number of **TXREQ** flits dispatched. This is a measure of the total request bandwidth.

The following table shows a summary of the HN-I PMU events.

Table 6-5 HN-I PMU event summary

Number	Name	Description
1	PMU_HNI_RXDAT	Number of **RXDAT** flits received. This is a measure of the true read data bandwidth.
2	PMU_HNI_TXDAT	Number of **TXDAT** flits dispatched. This is a measure of the true write data bandwidth.
3	PMU_HNI_TXREQ_TOTAL	Number of **TXREQ** flits dispatched. This is a measure of the total request bandwidth.
4	PMU_HNI_TXREQ_REQORDER	Number of **TXREQ** flits with ReqOrder bit set. This is a measure of the rate of requests with the ReqOrder bit set.

6.6   Ring performance events

You can use the link utilization event between two XPs to detect hot links on interconnects by measuring the token valid counts at each XP. This event helps to detect incorrect routing algorithms or device placement.

PMU_XP_VALID_FLIT   Signal event whenever token valid is cleared on the bus, indicating that a valid packet is passing through an XP on the specified bus.

Appendix A
Signal Descriptions

This appendix describes the external signals of the CCN-502 for a system that includes all possible CCN-502 components.

It contains the following sections:

A.1 About the signal descriptions

This section describes the CCN-502 signals.

———— **Note** ————

- Because there are multiple identical interfaces in the CCN-502, the signal names described in this appendix are only root names in many cases, and the actual signal name includes a port-specific identifier suffix.
- Your system configuration determines which of the signals described in this appendix are used in a particular system.

————————————

A.2 Clock and reset signals

The CCN-502 includes 1-7 clock inputs, depending on the configuration of an instantiation. It also includes three types of clock-enable input pins for frequency-divided operation of AMBA and debug and trace interfaces.

The CCN-502 *Input/Output* (I/O) signals are both synchronous and asynchronous to the clocks. Any specific requirements of the I/O, including asynchronous requirements and specific physical implementation requirements such as multicycle path constraints, are included in the I/O description.

The following table shows the CCN-502 clock and reset signals.

Table A-1 CCN-502 clock and reset signals

Signal	Type	Description	Connection information
GCLK0	Input	Clock input for Domain0, whose definition is configuration-dependent. See *Figure 2-9 CCN-502 clock domain, fully synchronous* on page 2-61.	Connect to global clock for CCN-502
nSRESET	Input	CCN-502 reset, active-LOW.	Connect to global reset for CCN-502

Clocks and resets for the optional RN-F DSSBs

The following table shows the CCN-502 clock and reset signals with the optional CCN502_RNF_DSSB.

Table A-2 CCN-502 clock and reset signals with optional CCN502_RNF_DSSB

Signal	Type	Description	Connection information
GCLK0	Input	Clock input for Domain0, whose definition is configuration-dependent. See *Figure 2-9 CCN-502 clock domain, fully synchronous* on page 2-61.	Connect to global clock for CCN-502
RXREQGCLKCD_NID<x> Where <x> is 1, 5, 7, or 11 (6XP/2HNF) or <x> is 1, 7, 9, or 15 (8XP/4HNF).		Clock input for REQ flit receive FIFO of the XP DSSB attached to Node ID <x>[e].	Connect to input clock from Node ID <x> CCN502_RNF_DSSB
RXRSPGCLKCD_NID<x> Where <x> is 1, 5, 7, or 11 (6XP/2HNF) or <x> is 1, 7, 9, or 15 (8XP/4HNF).		Clock input for RSP flit receive FIFO of the XP DSSB attached to Node ID <x>[e].	
RXDATGCLKCD_NID<x> Where <x> is 1, 5, 7, or 11 (6XP/2HNF) or <x> is 1, 7, 9, or 15 (8XP/4HNF).		Clock input for DAT flit receive FIFO of the XP DSSB attached to Node ID <x>[e].	

---

[e]   See *Figure 2-10 CCN-502 clock domains with optional DSSBs* on page 2-62.

Table A-2 CCN-502 clock and reset signals with optional CCN502_RNF_DSSB (continued)

Signal	Type	Description	Connection information
TXRSPGCLK_NID\<x\> Where \<x\> is 1, 5, 7, or 11 (6XP/2HNF) or \<x\> is 1, 7, 9, or 15 (8XP/4HNF).	Output	Clock output for RSP flit receive FIFO in CCN502_RNF_DSSB of Node ID \<x\>[e].	Connect output to Node ID \<x\> CCN502_RNF_DSSB
TXDATGCLK_NID\<x\> Where \<x\> is 1, 5, 7, or 11 (6XP/2HNF) or \<x\> is 1, 7, 9, or 15 (8XP/4HNF).		Clock output for DAT flit receive FIFO in CCN502_RNF_DSSB of Node ID \<x\>[e].	
TXSNPGCLK_NID\<x\> Where \<x\> is 1, 5, 7, or 11 (6XP/2HNF) or \<x\> is 1, 7, 9, or 15 (8XP/4HNF).		Clock output for SNP flit receive FIFO in CCN502_RNF_DSSB of Node ID \<x\>[e].	
nSRESET	Input	CCN-502 reset, active-LOW.	Connect to global reset for CCN-502

The following table shows the clock and reset signals for the optional CCN502_RNF_DSSB.

Table A-3 Clock and reset signals for optional CCN502_RNF_DSSB

Signal	Type	Description	Connection information
GCLKCD	Input	Clock input for device domain.	Connect device domain clock to CCN502_RNF_DSSB input
GCLKCD_RXREQ		Clock input for device domain, used to generate **TXREQGCLKCD_CCN** output.	
GCLKCD_RXRSP		Clock input for device domain, used to generate **TXRSPGCLKCD_CCN** output.	
GCLKCD_RXDAT		Clock input for device domain, used to generate **TXDATGCLKCD_CCN** output.	
RXRSPGCLK_CCN		Clock input for RSP flit receive FIFO in CCN502_RNF_DSSB.	Connect input to **TXRSPGCLK_NID\<x\>** output of the CCN-502
RXDATGCLK_CCN		Clock input for DAT flit receive FIFO in CCN502_RNF_DSSB.	Connect input to **TXDATGCLK_NID\<x\>** output of the CCN-502
RXSNPGCLK_CCN		Clock input for SNP flit receive FIFO in CCN502_RNF_DSSB.	Connect input to **TXSNPGCLK_NID\<x\>** output of the CCN-502
TXREQGCLKCD_CCN	Output	Clock output for REQ flit receive FIFO of XP DSSB attached to the CCN502_RNF_DSSB.	Connect output to **RXREQGCLKCD_NID\<x\>** input of the CCN-502
TXRSPGCLKCD_CCN		Clock output for RSP flit receive FIFO of XP DSSB attached to the CCN502_RNF_DSSB.	Connect output to **RXRSPGCLKCD_NID\<x\>** input of the CCN-502
TXDATGCLKCD_CCN		Clock output for DAT flit receive FIFO of XP DSSB attached to the CCN502_RNF_DSSB.	Connect output to **RXDATGCLKCD_NID\<x\>** input of the CCN-502
nDEVRESET	Input	Processor domain reset for CCN502_RNF_DSSB, active-LOW.	Connect to global reset for processor connected to CCN502_RNF_DSSB

Clocks and resets for the optional SN-F DSSBs

The following table shows the CCN-502 clock and reset signals with the optional CCN502_SNF_DSSB.

Table A-4 CCN-502 clock and reset signals with optional CCN502_SNF_DSSB

Signal	Type	Description	Connection information
GCLK0	Input	Clock input for Domain0, whose definition is configuration-dependent. See *Figure 2-9 CCN-502 clock domain, fully synchronous* on page 2-61.	Connect to global clock for CCN-502
RXRSPGCLKCD_NID<x> Where <x> is 2 or 8 (6XP/2HNF) or <x> is 2, 4, 10, or 12 (8XP/4HNF).	Input	Clock input for RSP flit receive FIFO of the XP DSSB attached to Node ID <x>[e].	Connect to input clock from Node ID <x> CCN502_SNF_DSSB
RXDATGCLKCD_NID<x> Where <x> is 2 or 8 (6XP/2HNF) or <x> is 2, 4, 10, or 12 (8XP/4HNF).		Clock input for DAT flit receive FIFO of the XP DSSB attached to Node ID <x>[e].	
TXREQGCLK_NID<x> Where <x> is 2 or 8 (6XP/2HNF) or <x> is 2, 4, 10, or 12 (8XP/4HNF).	Output	Clock output for REQ flit receive FIFO in CCN502_SNF_DSSB of Node ID <x>[e].	Connect output to Node ID <x> CCN502_SNF_DSSB
TXDATGCLK_NID<x> Where <x> is 2 or 8 (6XP/2HNF) or <x> is 2, 4, 10, or 12 (8XP/4HNF).		Clock output for DAT flit receive FIFO in CCN502_SNF_DSSB of Node ID <x>[e].	
nSRESET	Input	CCN-502 reset, active-LOW.	Connect to global reset for CCN-502

The following table shows the clock and reset signals for the optional CCN502_SNF_DSSB.

Table A-5 Clock and reset signals for optional CCN502_SNF_DSSB

Signal	Type	Description	Connection information
GCLKCD	Input	Clock input for device domain	Connect device domain clock to CCN502_SNF_DSSB input
GCLKCD_RXRSP		Clock input for device domain, used to generate **TXRSPGCLKCD_CCN** output	
GCLKCD_RXDAT		Clock input for device domain, used to generate **TXDATGCLKCD_CCN** output	
RXREQGCLK_CCN		Clock input for REQ flit receive FIFO in CCN502_SNF_DSSB	Connect input to **TXREQGCLK_NID<x>** output of the CCN-502
RXDATGCLK_CCN		Clock input for DAT flit receive FIFO in CCN502_SNF_DSSB	Connect input to **TXDATGCLK_NID<x>** output of the CCN-502
TXRSPGCLKCD_CCN	Output	Clock output for RSP flit receive FIFO of XP DSSB attached to the CCN502_SNF_DSSB	Connect output to **RXRSPGCLKCD_NID<x>** input of the CCN-502
TXDATGCLKCD_CCN		Clock output for DAT flit receive FIFO of XP DSSB attached to the CCN502_SNF_DSSB	Connect output to **RXDATGCLKCD_NID<x>** input of the CCN-502
nDEVRESET	Input	DMC domain reset for CCN502_SNF_DSSB, active-LOW	Connect to global reset for DMC connected to CCN502_SNF_DSSB

A.3 Clock management signals

The following table shows the clock management Q-Channel signals.

Table A-6 Clock management Q-Channel signals

Signal	Type	Description	Connection information
QACTIVE_CLKCTL	Output	Indication that the CCN-502 is active and that the *External Clock Controller* (ExtCC) must not make a request for the CCN-502 to prepare to stop the clocks.	Connect to external clock controller
QREQn_CLKCTL	Input	Request from the ExtCC for the CCN-502 to prepare to stop the clocks	Connect to external clock controller or tie HIGH if unused
QACCEPTn_CLKCTL	Output	Positive acknowledgment after receiving **QREQn** assertion indicating that the CCN-502 has completed preparation to stop the clocks and that the ExtCC can stop the clocks	Connect to external clock controller
QDENY_CLKCTL	Output	Negative acknowledgment after receiving **QREQn** assertion indicating that the CCN-502 has refused the request from the ExtCC to prepare to stop the clocks	

Related concepts

2.14.1 High-level clock gating on page 2-68.

A.4 Power management signals

The following tables show the power management signals.

The following table shows the power management signals for the logic power domain.

Table A-7 Power management signals for logic power domain

Signal	Type	Description	Connection information
PREQ_LOGIC	Input	Indicates a request for a power state transition.	Connect to external power management controller or tie LOW if unused.
PSTATE_LOGIC[0]	Input	The power state to which a transition is requested.[f] The following table shows the values for this signal.	Connect to external power management controller or tie HIGH if unused.
PACCEPT_LOGIC	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within the CCN-502.	Connect to external power management controller.
PDENY_LOGIC	Output	Indicates denial of the power state transition.	
PACTIVE_LOGIC	Output	Hint that indicates activity across the CCN-502. When LOW, it hints at the possibility of entering static retention or the OFF state.	

The following table shows the **PSTATE_LOGIC[0]** values.

Table A-8 PSTATE_LOGIC[0] values

Value	State	Definition
0	OFF	Prepare to power down, that is, close all CHI links.
1	ON	Enable activation of CHI links.

The following table shows the power management signals for the optional CCN502_RNF_DSSB and CCN502_SNF_DSSB power domains.

Table A-9 Power management signals for optional CCN502_RNF_DSSB and CCN502_SNF_DSSB power domains

Signal	Type	Description	Connection information
PREQ_DEV	Input	Indicates a request for a power state transition.	Connect to external power management controller or tie LOW if unused.
PSTATE_DEV[0]	Input	The power state to which a transition is requested.[f] The following table shows the values for this signal.	Connect to external power management controller or tie HIGH if unused.
PACCEPT_DEV	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within the CCN-502.	Connect to external power management controller.
PDENY_DEV	Output	Indicates denial of the power state transition.	
PACTIVE_DEV	Output	Hint that indicates activity across the CCN-502. When LOW, it hints at the possibility of entering static retention or the OFF state.	

---

[f] If *MultiCycle Path* (MCP), the MCP duration must be ≤8 cycles to the last flop to receive this signal. This is a requirement for implementation.

The following table shows the **PSTATE_DEV[0]** values.

Table A-10 PSTATE_DEV[0] values

Value	State	Definition
0	OFF	Prepare to power down, that is, close all CHI links.
1	ON	Enable activation of CHI links.

The following table shows the power management signals for the snoop filter RAM power domain.

Table A-11 Power management signals for snoop filter RAM power domain

Signal	Type	Description	Connection information
PREQ_SF	Input	Indicates a request for a power state transition.	Connect to external power management controller or tie LOW if unused.
PSTATE_SF[1:0]	Input	The power state to which a transition is requested.[f] The following table shows the values for this signal.	Connect to external power management controller or tie to `0b11` if unused.
PACCEPT_SF	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within the CCN-502.	Connect to external power management controller.
PDENY_SF	Output	Indicates denial of the power state transition.	
PACTIVE_SF	Output	Hint that indicates activity in the snoop filter. When LOW, it hints at the possibility of entering dynamic retention. When HIGH, it is an indication that the snoop filter is required and that the SoC must exit dynamic retention.	

The following table shows the **PSTATE_SF[1:0]** values.

Table A-12 PSTATE_SF[1:0] values

Value	State	Definition
`0b00`	OFF	Prepare to power down. Activity depends on previous P-state.
`0b01`	MEM_RET	HN-F prohibits access to snoop filter RAM arrays.
`0b10`	DYN_RET	HN-F prohibits access to snoop filter RAM arrays.
`0b11`	ON	Normal usage of snoop filter. Additional activity depends on previous P-state.

The following table shows the power management signals for the L3 tag/data RAMs in way[7:0].

Table A-13 Power management signals for L3 tag/data RAMs way[7:0]

Signal	Type	Description	Connection information
PREQ_L3RAM0	Input	Indicates a request for a power state transition.	Connect to external power management controller or tie LOW if unused.
PSTATE_L3RAM0[1:0]	Input	The power state to which a transition is requested.[f] The following table shows the values for this signal.	Connect to external power management controller or tie to `0b11` if unused.

Table A-13 Power management signals for L3 tag/data RAMs way[7:0] (continued)

Signal	Type	Description	Connection information
PACCEPT_L3RAM0	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within the CCN-502.	Connect to external power management controller.
PDENY_L3RAM0	Output	Indicates denial of the power state transition.	
PACTIVE_L3RAM0	Output	Hint that indicates activity in way[7:0] of the L3 RAMs. When LOW, it hints at the possibility of entering dynamic retention. When HIGH, it is an indication that these L3 RAMs are required and that the SoC must exit dynamic retention.	

The following table shows the **PSTATE_L3RAM0[1:0]** values.

Table A-14 PSTATE_L3RAM0[1:0] values

Value	State	Definition
0b00	OFF	Prepare to power down. Activity depends on previous P-state.
0b01	MEM_RET	HN-F prohibits access to L3 RAM arrays for way[7:0].
0b10	DYN_RET	HN-F prohibits access to L3 RAM arrays for way[7:0].
0b11	ON	Normal usage of L3 RAM arrays for way[7:0]. Additional activity depends on previous P-state.

The following table shows the power management signals for the L3 tag/data RAMs in way[15:8].

Table A-15 Power management signals for L3 tag/data RAMs way[15:8]

Signal	Type	Description	Connection information
PREQ_L3RAM1	Input	Indicates a request for a power state transition.	Connect to external power management controller or tie LOW if unused.
PSTATE_L3RAM1[1:0]	Input	The power state to which a transition is requested.[f] The following table shows the values for this signal.	Connect to external power management controller or tie to 0b11 if unused.
PACCEPT_L3RAM1	Output	Indicates acknowledgment of the power state transition and completion of the power state transition within the CCN-502.	Connect to external power management controller.
PDENY_L3RAM1	Output	Indicates denial of the power state transition.	
PACTIVE_L3RAM1	Output	Hint that indicates activity in way[15:8] of the L3 RAMs. When LOW, it hints at the possibility of entering dynamic retention. When HIGH, it is an indication that these L3 RAMs are required and that the SoC must exit dynamic retention.	

The following table shows the **PSTATE_L3RAM1[1:0]** values.

Table A-16 PSTATE_L3RAM1[1:0] values

Value	State	Definition
0b00	OFF	Prepare to power down. Activity depends on previous P-state.
0b01	MEM_RET	HN-F prohibits access to L3 RAM arrays for way[15:8].
0b10	DYN_RET	HN-F prohibits access to L3 RAM arrays for way[15:8].
0b11	ON	Normal usage of L3 RAM arrays for way[15:8]. Additional activity depends on previous P-state.

A.5 Interrupt and event signals

The following table shows the interrupt and event signals.

Table A-17 Interrupt and event signals

Signal	Type	Description	Connection information
INTREQ	Output	Debug trigger and error indicator. Indicates error or performance monitor counter overflow.	Connect to external interrupt control logic or interrupt controller.
CLREXMONREQ_NID\<x\> Where **\<x\>** is 1, 5, 7, or 11 (6XP/2HNF) or **\<x\>** is 1, 7, 9, or 15 (8XP/4HNF).	Output	Indicates that an exclusive monitor in the CCN-502 has been cleared. Paired with the corresponding **CLREXMONACK_NID\<x\>** input pin in an asynchronous-safe 4-phase handshake. For connection to ARMv8-compliant processors only.	Connect to **CLREXMON** control logic for processor at Node ID \<x\>.
CLREXMONACK_NID\<x\> Where **\<x\>** is 1, 5, 7, or 11 (6XP/2HNF) or **\<x\>** is 1, 7, 9, or 15 (8XP/4HNF).	Input	Acknowledgment from an ARMv8-compliant processor that a corresponding **CLREXMONREQ** has been received. Paired with the corresponding **CLREXMONREQ_NID\<x\>** output pin in an asynchronous-safe 4-phase handshake.	Connect to **CLREXMON** control logic for processor at Node ID \<x\> or tie LOW if processor not populated or not ARMv8-compliant.

A.6 Configuration input signals

The following table shows the configuration input signals. All these signals must be stable at least ten cycles before deassertion of reset and must remain stable throughout the operation of the CCN-502, until a following reset assertion or powerdown, if any.

Table A-18 Configuration input signals

Signal	Type	Description	Connection information
General configuration input signals			
PERIPHBASE[43:24]	Input	Base address of the CCN-502 configuration register space.	Tie as required for system memory map.
SBSX_128_n256	Input	Controls the data width of **RDATA_M** and **WDATA_M** for the SBSX AXI4 interfaces: **0** 256-bit effective data width. **1** 128-bit effective data width. Uses the least-significant 128 bits. The upper 128 bits are undriven and unsampled. ———— **Note** ———— The physical bus width is always 256 bits. ————————————	Tie as required for system MC data width.

---

g   The field encoding definitions are:

0b00	The HN-Fs can access this region.
0b01	The HN-I can access this region.
0b10-0b11	Reserved.

Table A-18 Configuration input signals (continued)

Signal	Type	Description	Connection information
SAM configuration input signals			Tie as required for system memory map.
SAMADDRMAP0[1:0]	Input	0-512MB region mapping[g].	
SAMADDRMAP1[1:0]	Input	512MB-1GB region mapping[g].	
SAMADDRMAP2[1:0]	Input	1GB-1.5GB region mapping[g].	
SAMADDRMAP3[1:0]	Input	1.5GB-2GB region mapping[g].	
SAMADDRMAP4[1:0]	Input	2GB-2.5GB region mapping[g].	
SAMADDRMAP5[1:0]	Input	2.5GB-3GB region mapping[g].	
SAMADDRMAP6[1:0]	Input	3GB-3.5GB region mapping[g].	
SAMADDRMAP7[1:0]	Input	3.5GB-4GB region mapping[g].	
SAMADDRMAP8[1:0]	Input	4GB-8GB region mapping[g].	
SAMADDRMAP9[1:0]	Input	8GB-16GB region mapping[g].	
SAMADDRMAP10[1:0]	Input	16GB-32GB region mapping[g].	
SAMADDRMAP11[1:0]	Input	32GB-64GB region mapping[g].	
SAMADDRMAP12[1:0]	Input	64GB-128GB region mapping[g].	
SAMADDRMAP13[1:0]	Input	128GB-256GB region mapping[g].	
SAMADDRMAP14[1:0]	Input	256GB-512GB region mapping[g].	
SAMADDRMAP15[1:0]	Input	512GB-1TB region mapping[g].	
SAMADDRMAP16[1:0]	Input	1TB-2TB region mapping[g].	
SAMADDRMAP17[1:0]	Input	2TB-4TB region mapping[g].	
SAMADDRMAP18[1:0]	Input	4TB-8TB region mapping[g].	
SAMADDRMAP19[1:0]	Input	8TB-16TB region mapping[g].	
		6XP/2HNF Specific	
SAMMNNODEID[6:0]	Input	MN Node ID	Tie to `0x00`
SAMHNI0NODEID[6:0]	Input	HN-I 0 Node ID	Tie to `0x00`
SAMHNI1NODEID[6:0]	Input	HN-I 1 Node ID	Tie to `0x00`
SAMHNF0NODEID[6:0]	Input	HN-F 0 Node ID	Tie to `0x03`
SAMHNF1NODEID[6:0]	Input	HN-F 1 Node ID	Tie to `0x09`
SAMHNF2NODEID[6:0]	Input	HN-F 2 Node ID	Tie to `0x00`
SAMHNF3NODEID[6:0]	Input	HN-F 3 Node ID	Tie to `0x00`
SAMHNF4NODEID[6:0]	Input	HN-F 4 Node ID	Tie to `0x00`
SAMHNF5NODEID[6:0]	Input	HN-F 5 Node ID	Tie to `0x00`
SAMHNF6NODEID[6:0]	Input	HN-F 6 Node ID	Tie to `0x00`
SAMHNF7NODEID[6:0]	Input	HN-F 7 Node ID	Tie to `0x00`
SAMHNFMODE[2:0]	Input	Number of HN-Fs. Fixed at 2.	Tie to `0x01`.

Table A-18 Configuration input signals (continued)

Signal	Type	Description	Connection information
		8XP/4HNF Specific	
SAMMNNODEID[6:0]	Input	MN Node ID	Tie to `0x00`
SAMHNI0NODEID[6:0]	Input	HN-I 0 Node ID	Tie to `0x00`
SAMHNI1NODEID[6:0]	Input	HN-I 1 Node ID	Tie to `0x00`
SAMHNF0NODEID[6:0]	Input	HN-F 0 Node ID	Tie to `0x03`
SAMHNF1NODEID[6:0]	Input	HN-F 1 Node ID	Tie to `0x05`
SAMHNF2NODEID[6:0]	Input	HN-F 2 Node ID	Tie to `0x0B`
SAMHNF3NODEID[6:0]	Input	HN-F 3 Node ID	Tie to `0x0D`
SAMHNF4NODEID[6:0]	Input	HN-F 4 Node ID	Tie to `0x00`
SAMHNF5NODEID[6:0]	Input	HN-F 5 Node ID	Tie to `0x00`
SAMHNF6NODEID[6:0]	Input	HN-F 6 Node ID	Tie to `0x00`
SAMHNF7NODEID[6:0]	Input	HN-F 7 Node ID	Tie to `0x00`
SAMHNFMODE[2:0]	Input	Number of HN-Fs. Fixed at 4.	Tie to `0x02`.

Related concepts

2.12.2 SAM configuration on page 2-54.

A.7 Device population signals

The following table shows the RN-F device population signals.

Table A-19 RN-F device population signals

Signal	Type	Description	Connection information
RNFEN_NID\<x\> Where \<x\> is 1, 5, 7, or 11 (6XP/2HNF) or \<x\> is 1, 7, 9, or 15 (8XP/4HNF).	Input	Indicates that the RN-F port at NodeID \<x\> is populated with a device that can respond to snoop requests on the CHI SNP channel. **0** Device is not populated. **1** Device is populated.	Tie as required for system configuration.

The following table shows the RN-I ACE-Lite+DVM device population signals. These signals are present only when the CCN-502 has been configured to include the relevant RN-I bridge, and the relevant RN-I bridge has been configured to support ACE-Lite+DVM functionality.

Table A-20 RN-I ACE-Lite+DVM device population signals

Signal	Type	Description	Connection information
ACCHANNELEN_S0_NID\<x\> Where \<x\> is 4, 6, or 10 (6XP/2HNF) or \<x\> is 6, 8, or 14 (8XP/4HNF).	Input	Indicates that the RN-I bridge at NodeID \<x\> is populated and AMBA slave port 0 for NodeID \<x\> is of type ACE-Lite+DVM and includes a device that responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated	Tie as required for system configuration.
ACCHANNELEN_S1_NID\<x\> Where \<x\> is 4, 6, or 10 (6XP/2HNF) or \<x\> is 6, 8, or 14 (8XP/4HNF).	Input	Indicates that the RN-I bridge at NodeID \<x\> is populated and AMBA slave port 1 for NodeID \<x\> is of type ACE-Lite+DVM and includes a device that responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated.	
ACCHANNELEN_S2_NID\<x\> Where \<x\> is 4, 6, or 10 (6XP/2HNF) or \<x\> is 6, 8, or 14 (8XP/4HNF).	Input	Indicates that the RN-I bridge at NodeID \<x\> is populated and AMBA slave port 2 for NodeID \<x\> is of type ACE-Lite+DVM and includes a device that responds to DVM messages on the AC channel. **0** DVM-capable device is not populated. **1** DVM-capable device is populated.	

A.8 CHI interface signals

This section describes the channels that form the inbound and outbound CHI interface for each device and the signals that form each channel in a specific interface.

The *ARM® AMBA® 5 CHI Architecture Specification* defines four channels:

- *Request* (REQ).
- *Response* (RSP).
- *Snoop* (SNP).
- *Data* (DAT).

This hierarchy enables you to understand the CHI interfaces for a specific device at a higher level than the raw signals in the respective channels.

——————— Note ———————

All signal names in this section are only a root name, **RootName**. The CCN-502 interfaces use **RootName** within a more fully specified signal name as follows:

- CCN-502 interface signal name == **RootName_NID#**, where # is the node ID corresponding to the specific interface.

This section contains the following subsections:

A.8.1 Per-device interface definition

Each CHI device included in a CCN-502 system has a distinct functionality, and the requirements and configuration of its respective CHI interfaces differ.

The requirements and configuration for the CHI interfaces are as follows:

External RN-F interface

The RN-F interface consists of a request channel, snoop channel, and two response channels, one in each direction, as the following figure shows. It also has two data channels, one in each direction, for data transfers. The CCN-502 receives request messages from the RN-F and sends responses to it. In addition, the CCN-502 sends snoop messages to the RN-F and receives snoop response messages.
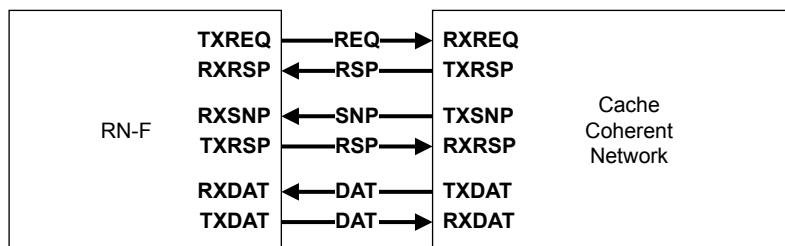
Figure A-1 External RN-F interface

External SN-F interface

The SN-F interface consists of a request channel and a response channel as the following figure shows. It also has two data channels, one in each direction, for data transfers. The SN-F receives request messages from the CCN-502 and returns response messages.
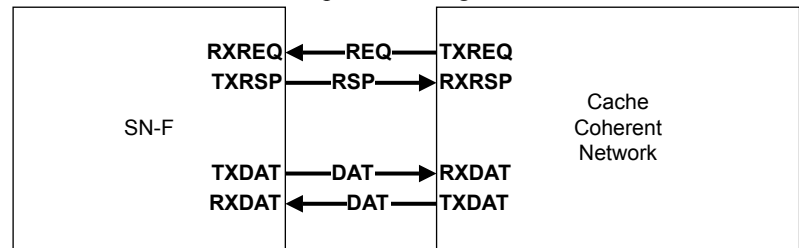
Figure A-2 External SN-F interface

A.8.2 Per-channel interface signals

This section describes the signals in each channel interface. For communication between devices, each of the channels includes a *Transmit* (TX) and a *Receive* (RX) port, with signals traveling from TX to RX.

——————— **Note** ———————

Connection of CHI interfaces between two devices requires cross-coupling of the **TX*** and **RX*** signals between the two devices, as required by the CHI architecture. *Table A-21 Transmit Request channel signals* on page Appx-A-276 to *Table A-41 SACTIVE interface signals* on page Appx-A-283 describe this cross-coupling.

————————————————————

The following table shows the Transmit Request channel signals.

Table A-21 Transmit Request channel signals

Signal	Type	Description	Connection information
TXREQFLITPEND	Output	Transmit Request Early Flit Valid hint	Connect to **RXREQFLITPEND** of the corresponding CHI device, if populated
TXREQFLITV	Output	Transmit Request Flit Valid	Connect to **RXREQFLITV** of the corresponding CHI device, if populated
TXREQFLIT[x:0][h]	Output	Transmit Request Flit	Connect to **RXREQFLIT** of the corresponding CHI device, if populated
TXREQLCRDV	Input	Transmit Request channel link layer credit	Connect to **RXREQLCRDV** of the corresponding CHI device, if populated, otherwise tie LOW

The following table shows the Transmit Response channel signals.

Table A-22 Transmit Response channel signals

Signal	Type	Description	Connection information
TXRSPFLITPEND	Output	Transmit Response Early Flit Valid hint	Connect to **RXRSPFLITPEND** of the corresponding CHI device, if populated
TXRSPFLITV	Output	Transmit Response Flit Valid	Connect to **RXRSPFLITV** of the corresponding CHI device, if populated

------

  [h]   x = 99 if RSVDC width is 4 bits, x = 103 if RSVDC width is 8 bits.

Table A-22 Transmit Response channel signals (continued)

Signal	Type	Description	Connection information
TXRSPFLIT[44:0]	Output	Transmit Response Flit	Connect to **RXRSPFLIT** of the corresponding CHI device, if populated
TXRSPLCRDV	Input	Transmit Response channel link layer credit	Connect to **RXRSPLCRDV** of the corresponding CHI device, if populated, otherwise tie LOW

The following table shows the Transmit Snoop channel signals.

Table A-23 Transmit Snoop channel signals

Signal	Type	Description	Connection information
TXSNPFLITPEND	Output	Transmit Snoop Early Flit Valid hint	Connect to **RXSNPFLITPEND** of the corresponding CHI device, if populated
TXSNPFLITV	Output	Transmit Snoop Flit Valid	Connect to **RXSNPFLITV** of the corresponding CHI device, if populated
TXSNPFLIT[64:0]	Output	Transmit Snoop Flit	Connect to **RXSNPFLIT** of the corresponding CHI device, if populated
TXSNPLCRDV	Input	Transmit Snoop channel link layer credit	Connect to **RXSNPLCRDV** of the corresponding CHI device, if populated, otherwise tie LOW

The following table shows the Transmit Data channel signals.

Table A-24 Transmit Data channel signals

Signal	Type	Description	Connection information
TXDATFLITPEND	Output	Transmit Data Early Flit Valid hint	Connect to **RXDATFLITPEND** of the corresponding CHI device, if populated
TXDATFLITV	Output	Transmit Data Flit Valid	Connect to **RXDATFLITV** of the corresponding CHI device, if populated
TXDATFLIT[193:0]	Output	Transmit Data Flit	Connect to **RXDATFLIT** of the corresponding CHI device, if populated
TXDATLCRDV	Input	Transmit Data channel link layer credit	Connect to **RXDATLCRDV** of the corresponding CHI device, if populated, otherwise tie LOW

The following table shows the Receive Request channel signals.

Table A-25 Receive Request channel signals

Signal	Type	Description	Connection information
RXREQFLITPEND	Input	Receive Request Early Flit Valid hint	Connect to **TXREQFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXREQFLITV	Input	Receive Request Flit Valid	Connect to **TXREQFLITV** of the corresponding processor, if populated, otherwise tie LOW

Table A-25 Receive Request channel signals (continued)

Signal	Type	Description	Connection information
RXREQFLIT[x:0][i]	Input	Receive Request Flit	Connect to **TXREQFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXREQLCRDV	Output	Receive Request channel link layer credit	Connect to **TXREQLCRDV** of the corresponding CHI device, if populated

The following table shows the Receive Response channel signals.

Table A-26 Receive Response channel signals

Signal	Type	Description	Connection information
RXRSPFLITPEND	Input	Receive Response Early Flit Valid hint	Connect to **TXRSPFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXRSPFLITV	Input	Receive Response Flit Valid	Connect to **TXRSPFLITV** of the corresponding processor, if populated, otherwise tie LOW
RXRSPFLIT[44:0]	Input	Receive Response Flit	Connect to **TXRSPFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXRSPLCRDV	Output	Receive Response channel link layer credit	Connect to **TXRSPLCRDV** of the corresponding CHI device, if populated

The following table shows the Receive Snoop channel signals.

Table A-27 Receive Snoop channel signals

Signal	Type	Description	Connection information
RXSNPFLITPEND	Input	Receive Snoop Early Flit Valid hint	Connect to **TXSNPFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXSNPFLITV	Input	Receive Snoop Flit Valid	Connect to **TXSNPFLITV** of the corresponding processor, if populated, otherwise tie LOW
RXSNPFLIT[64:0]	Input	Receive Snoop Flit	Connect to **TXSNPFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXSNPLCRDV	Output	Receive Snoop channel link layer credit	Connect to **TXSNPLCRDV** of the corresponding CHI device, if populated

The following table shows the Receive Data channel signals.

Table A-28 Receive Data channel signals

Signal	Type	Description	Connection information
RXDATFLITPEND	Input	Receive Data Early Flit Valid hint	Connect to **TXDATFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXDATFLITV	Input	Receive Data Flit Valid	Connect to **TXDATFLITV** of the corresponding processor, if populated, otherwise tie LOW

---

[i]   x = 99 if RSVDC width is 4 bits, x = 103 if RSVDC width is 8 bits.

Table A-28 Receive Data channel signals (continued)

Signal	Type	Description	Connection information
RXDATFLIT[193:0]	Input	Receive Data Flit	Connect to **TXDATFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXDATLCRDV	Output	Receive Data channel link layer credit	Connect to **TXDATLCRDV** of the corresponding CHI device, if populated

The following table shows the Transmit Request channel signals in configurations with optional *Device to XP Source Synchronous Bridges* (DSSBs).

Table A-29 Transmit Request channel signals with optional DSSBs

Signal	Type	Description	Connection information
TXREQFLITPEND	Output	Transmit Request Early Flit Valid hint	Connect to **RXREQFLITPEND** of the corresponding CHI device, if populated
TXREQFLITV	Output	Transmit Request Flit Valid	Connect to **RXREQFLITV** of the corresponding CHI device, if populated
TXREQFLIT[x:0][j]	Output	Transmit Request Flit	Connect to **RXREQFLIT** of the corresponding CHI device, if populated
TXREQLCRDPTR[7:0]	Input	Transmit Request channel link layer pointer	Connect to **RXREQLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Transmit Response channel signals in configurations with optional DSSBs.

Table A-30 Transmit Response channel signals with optional DSSBs

Signal	Type	Description	Connection information
TXRSPFLITPEND	Output	Transmit Response Early Flit Valid hint	Connect to **RXRSPFLITPEND** of the corresponding CHI device, if populated
TXRSPFLITV	Output	Transmit Response Flit Valid	Connect to **RXRSPFLITV** of the corresponding CHI device, if populated
TXRSPFLIT[44:0]	Output	Transmit Response Flit	Connect to **RXRSPFLIT** of the corresponding CHI device, if populated
TXRSPLCRDPTR[7:0]	Input	Transmit Response channel link layer pointer	Connect to **RXRSPLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Transmit Snoop channel signals in configurations with optional DSSBs.

Table A-31 Transmit Snoop channel signals with optional DSSBs

Signal	Type	Description	Connection information
TXSNPFLITPEND	Output	Transmit Snoop Early Flit Valid hint	Connect to **RXSNPFLITPEND** of the corresponding CHI device, if populated
TXSNPFLITV	Output	Transmit Snoop Flit Valid	Connect to **RXSNPFLITV** of the corresponding CHI device, if populated

---

[j]   x = 99 if RSVDC width is 4 bits, x = 103 if RSVDC width is 8 bits.

Table A-31 Transmit Snoop channel signals with optional DSSBs (continued)

Signal	Type	Description	Connection information
TXSNPFLIT[64:0]	Output	Transmit Snoop Flit	Connect to **RXSNPFLIT** of the corresponding CHI device, if populated
TXSNPLCRDPTR[7:0]	Input	Transmit Snoop channel link layer pointer	Connect to **RXSNPLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Transmit Data channel signals in configurations with optional DSSBs.

Table A-32 Transmit Data channel signals with optional DSSBs

Signal	Type	Description	Connection information
TXDATFLITPEND	Output	Transmit Data Early Flit Valid hint	Connect to **RXDATFLITPEND** of the corresponding CHI device, if populated
TXDATFLITV	Output	Transmit Data Flit Valid	Connect to **RXDATFLITV** of the corresponding CHI device, if populated
TXDATFLIT[193:0]	Output	Transmit Data Flit	Connect to **RXDATFLIT** of the corresponding CHI device, if populated
TXDATLCRDPTR[7:0]	Input	Transmit Data channel link layer pointer	Connect to **RXDATLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Receive Request channel signals in configurations with optional DSSBs.

Table A-33 Receive Request channel signals with optional DSSBs

Signal	Type	Description	Connection information
RXREQFLITPEND	Input	Receive Request Early Flit Valid hint	Connect to **TXREQFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXREQFLITV	Input	Receive Request Flit Valid	Connect to **TXREQFLITV** of the corresponding processor, if populated, otherwise tie LOW
RXREQFLIT[x:0][k]	Input	Receive Request Flit	Connect to **TXREQFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXREQLCRDPTR[7:0]	Input	Receive Request channel link layer pointer	Connect to **TXREQLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Receive Response channel signals in configurations with optional DSSBs.

Table A-34 Receive Response channel signals with optional DSSBs

Signal	Type	Description	Connection information
RXRSPFLITPEND	Input	Receive Response Early Flit Valid hint	Connect to **TXRSPFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXRSPFLITV	Input	Receive Response Flit Valid	Connect to **TXRSPFLITV** of the corresponding processor, if populated, otherwise tie LOW

---

[k]   x = 99 if RSVDC width is 4 bits, x = 103 if RSVDC width is 8 bits.

Table A-34 Receive Response channel signals with optional DSSBs (continued)

Signal	Type	Description	Connection information
RXRSPFLIT[44:0]	Input	Receive Response Flit	Connect to **TXRSPFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXRSPLCRDPTR[7:0]	Input	Receive Response channel link layer pointer	Connect to **TXRSPLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Receive Snoop channel signals in configurations with optional DSSBs.

Table A-35 Receive Snoop channel signals with optional DSSBs

Signal	Type	Description	Connection information
RXSNPFLITPEND	Input	Receive Snoop Early Flit Valid hint	Connect to **TXSNPFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXSNPFLITV	Input	Receive Snoop Flit Valid	Connect to **TXSNPFLITV** of the corresponding processor, if populated, otherwise tie LOW
RXSNPFLIT[64:0]	Input	Receive Snoop Flit	Connect to **TXSNPFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXSNPLCRDPTR[7:0]	Input	Receive Snoop channel link layer pointer	Connect to **TXSNPLCRDPTR[7:0]** of the corresponding DSSB

The following table shows the Receive Data channel signals in configurations with optional DSSBs.

Table A-36 Receive Data channel signals with optional DSSBs

Signal	Type	Description	Connection information
RXDATFLITPEND	Input	Receive Data Early Flit Valid hint	Connect to **TXDATFLITPEND** of the corresponding CHI device, if populated, otherwise tie LOW
RXDATFLITV	Input	Receive Data Flit Valid	Connect to **TXDATFLITV** of the corresponding processor, if populated, otherwise tie LOW
RXDATFLIT[193:0]	Input	Receive Data Flit	Connect to **TXDATFLIT** of the corresponding CHI device, if populated, otherwise tie LOW
RXDATLCRDPTR[7:0]	Input	Receive Data channel link layer pointer	Connect to **TXDATLCRDPTR[7:0]** of the corresponding DSSB

A.8.3   Non-channel-specific interface signals

In addition to the per-channel signals described in *A.8.2 Per-channel interface signals on page Appx-A-276*, every transmit and receive link layer interface includes additional signals that exist only at the interface level and are not channel specific.

The following table shows the Receive LinkActive interface signals.

Table A-37 Receive LinkActive interface signals

Signal	Type	Description	Connection information
RXLINKACTIVEREQ	Input	Receive channel LinkActive request from adjacent transmitter device	Connect to **TXLINKACTIVEREQ** of the corresponding CHI device, if populated, otherwise tie LOW
RXLINKACTIVEACK	Output	Receive channel LinkActive acknowledgment to adjacent transmitter device	Connect to **TXLINKACTIVEACK** of the corresponding CHI device, if populated

The following table shows the Transmit LinkActive interface signals.

Table A-38 Transmit LinkActive interface signals

Signal	Type	Description	Connection information
TXLINKACTIVEREQ	Output	Transmit channel LinkActive request from adjacent receiver device	Connect to **RXLINKACTIVEREQ** of the corresponding CHI device, if populated
TXLINKACTIVEACK	Input	Transmit channel LinkActive acknowledgment to adjacent receiver device	Connect to **RXLINKACTIVEACK** of the corresponding CHI device, if populated, otherwise tie LOW

The following table shows the Receive LinkActive interface signals in configurations with optional DSSBs.

Table A-39 Receive LinkActive interface signals with optional DSSBs

Signal	Type	Description	Connection information
RXLINKACTIVEREQ	Input	Receive channel LinkActive request from adjacent transmitter device	Connect to **TXLINKACTIVEREQ** of the corresponding CHI device, if populated, otherwise tie LOW
RXLINKACTIVEACK	Output	Receive channel LinkActive acknowledgment to adjacent transmitter device	Connect to **TXLINKACTIVEACK** of the corresponding CHI device, if populated
RXLINKACTIVEDENY	Output	Receive channel LinkActive deny to adjacent receiver device	Connect to **TXLINKACTIVEDENY** of the corresponding DSSB

The following table shows the Transmit LinkActive interface signals in configurations with optional DSSBs.

Table A-40 Transmit LinkActive interface signals with optional DSSBs

Signal	Type	Description	Connection information
TXLINKACTIVEREQ	Output	Transmit channel LinkActive request from adjacent receiver device	Connect to **RXLINKACTIVEREQ** of the corresponding CHI device, if populated
TXLINKACTIVEACK	Input	Transmit channel LinkActive acknowledgment to adjacent receiver device	Connect to **RXLINKACTIVEACK** of the corresponding CHI device, if populated, otherwise tie LOW
TXLINKACTIVEDENY	Input	Transmit channel LinkActive deny to adjacent receiver device	Connect to **RXLINKACTIVEDENY** of the corresponding DSSB

The following table shows the SACTIVE interface signals.

Table A-41 SACTIVE interface signals

Signal	Type	Description	Connection information
RXSACTIVE	Input	Indication from the adjacent CHI device that it has one or more outstanding protocol-layer transactions. **RXSACTIVE** remains asserted throughout the lifetime of the transactions as interpreted by the adjacent CHI device.	Connect to **TXSACTIVE** of the corresponding CHI device, if populated, otherwise tie LOW
TXSACTIVE	Output	Indication to the adjacent CHI device that the CCN-502 has one or more outstanding protocol-layer transactions. **TXSACTIVE** remains asserted throughout the lifetime of the transactions as interpreted by the CCN-502.	Connect to **RXSACTIVE** of the corresponding CHI device, if populated

Related references

A.8.2 Per-channel interface signals on page Appx-A-276.

A.9   ACE-Lite and AXI interface signals

This section describes the ACE-Lite and AXI interface signals.

——————— **Note** ———————

All signal names in this section consist of a root name, **RootName**. The CCN-502 interfaces use **RootName** within a more fully specified signal name as follows:

- CCN-502 ACE-Lite and AXI interface signal name == **RootName_[S|M]<#a>_NID#b**, where:

   S|M Defines either a slave or master interface.

   $\#_a$ Defines an optional interface identifier for a node that can support multiple AMBA interfaces.

   $\#_b$ Defines the node ID corresponding to the specific interface.

Multi-bit signals append the bit-range identifier included in the **RootName** to the end of the full signal name.

————————————————

This section contains the following subsections:

A.9.1   ACE-Lite-with-DVM slave interface signals

This interface is present as the ACE-Lite-with-DVM slave port for an RN-I bridge. The signal descriptions show which signals specific to DVM functionality are not present in an ACE-Lite interface without DVM.

The following table shows the clock and power management signals.

Table A-42 Clock and power management signals

Signal	Type	Description	Connection information
ACLKEN_S	Input	AXI bus clock enable	Connect to clock enable logic. Tie high if RN-I port is unused.
CACTIVE_S	Input	Indication that master device is active	Connect to the **CACTIVE** output of the corresponding ADB-400, if present, otherwise tie LOW.

The following table shows the Write Address Channel signals.

Table A-43 Write Address Channel signals

Signal	Type	Description	Connection information
AWREADY_S	Output	Write address ready	Connect to corresponding master device, if populated.
AWVALID_S	Input	Write address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
AWID_S[10:0]	Input	Write address ID	
AWADDR_S[39:0]	Input	Write address	
AWLEN_S[7:0]	Input	Write burst length	
AWSIZE_S[2:0]	Input	Write burst size	
AWBURST_S[1:0]	Input	Write burst type	
AWLOCK_S	Input	Write lock type	
AWCACHE_S[3:0]	Input	Write memory type	
AWUSER_S[x:0][1]	Input	User-defined signal	
AWPROT_S[2:0]	Input	Write protection type	
AWQOS_S[3:0]	Input	Write Quality of Service identifier	
AWSNOOP_S[2:0]	Input	Write transaction type	
AWDOMAIN_S[1:0]	Input	Write shareability domain	
AWBAR_S[1:0]	Input	Write barrier transaction	

The following table shows the Write Data Channel signals.

Table A-44 Write Data Channel signals

Signal	Type	Description	Connection information
WREADY_S	Output	Write data ready	Connect to corresponding master device, if populated.
WVALID_S	Input	Write data valid	Connect to corresponding master device, if populated, otherwise tie LOW.
WDATA_S[127:0]	Input	Write data	Connect to corresponding master device, if populated, otherwise tie LOW.
WSTRB_S[15:0]	Input	Write byte-lane strobes	Connect to corresponding master device, if populated, otherwise tie LOW.
WLAST_S	Input	Write data last transfer indication	Connect to corresponding master device, if populated, otherwise tie LOW.
WUSER_S[0]	Input	HIGH when the parity data is valid.	Connect to corresponding master device, if populated, otherwise tie LOW.
WUSER_S[16:1]	Input	Odd parity data.	Connect to corresponding master device, if populated, otherwise tie LOW.

The following table shows the Write Response Channel signals.

---

[1]   The value x is based on REQ VC RSVDC width producing either 3:0 or 7:0.

Table A-45 Write Response Channel signals

Signal	Type	Description	Connection information
BREADY_S	Input	Write response ready	Connect to corresponding master device, if populated, otherwise tie LOW.
BVALID_S	Output	Write response valid	Connect to corresponding master device, if populated.
BID_S[10:0]	Output	Write response ID	
BRESP_S[1:0]	Output	Write response	
BUSER_S[3:0]	Output	User response signal	

The following table shows the Read Address Channel signals.

Table A-46 Read Address Channel signals

Signal	Type	Description	Connection information
ARREADY_S	Output	Read address ready	Connect to corresponding master device, if populated.
ARVALID_S	Input	Read address valid	Connect to corresponding master device, if populated, otherwise tie LOW.
ARID_S[10:0]	Input	Read address ID	
ARADDR_S[39:0]	Input	Read address	
ARLEN_S[7:0]	Input	Read burst length	
ARSIZE_S[2:0]	Input	Read burst size	
ARBURST_S[1:0]	Input	Read burst type	
ARLOCK_S	Input	Read lock type	
ARCACHE_S[3:0]	Input	Read cache type	
ARUSER_S[x:0][m]	Input	User-defined signal	
ARPROT_S[2:0]	Input	Read protection type	
ARQOS_S[3:0]	Input	Read Quality of Service value	
ARSNOOP_S[3:0]	Input	Read transaction type	
ARDOMAIN_S[1:0]	Input	Read shareability domain	
ARBAR_S[1:0]	Input	Read barrier transaction	

The following table shows the Read Data Channel signals.

Table A-47 Read Data Channel signals

Signal	Type	Description	Connection information
RREADY_S	Input	Read data ready	Connect to corresponding master device, if populated, otherwise tie LOW.
RVALID_S	Output	Read data valid	Connect to corresponding master device, if populated.
RID_S[10:0]	Output	Read data ID	Connect to corresponding master device, if populated.
RDATA_S[127:0]	Output	Read data	Connect to corresponding master device, if populated.
RRESP_S[1:0]	Output	Read data response	Connect to corresponding master device, if populated.

---

[m]   The value x is based on REQ VC RSVDC width producing either 3:0 or 7:0.

Table A-47 Read Data Channel signals (continued)

Signal	Type	Description	Connection information
RLAST_S	Output	Read data last transfer indication	Connect to corresponding master device, if populated.
RUSER_S[0]	Output	Indicates when the parity data is valid.	Connect to corresponding master device, if populated.
RUSER_S[16:1]	Output	Odd parity data.	Connect to corresponding master device, if populated.

The following table shows the Snoop Address Channel signals. These signals are not included in an ACE-Lite interface without DVM.

Table A-48 Snoop Address Channel signals

Signal	Type	Description	Connection information
ACREADY_S	Input	Snoop address ready	Connect to corresponding master device, if populated, otherwise tie LOW.
ACVALID_S	Output	Snoop address valid	Connect to corresponding master device, if populated.
ACADDR_S[43:0]	Output	Snoop address	
ACSNOOP_S[3:0]	Output	Snoop transaction type	
ACPROT_S[2:0]	Output	Snoop protection type	

The following table shows the Snoop Response Channel signals. These signals are not included in an ACE-Lite interface without DVM.

Table A-49 Snoop Response Channel signals

Signal	Type	Description	Connection information
CRREADY_S	Output	Snoop response ready	Connect to corresponding master device, if populated.
CRVALID_S	Input	Snoop response valid	Connect to corresponding master device, if populated, otherwise tie LOW.
CRRESP_S[4:0]	Input	Snoop response.	

A.9.2 AXI4/ACE-Lite master interface signals

The HN-I has an AXI4/ACE-Lite master interface. The tables in this section identify the signals specific to the ACE-Lite functionality, as distinct from the AXI4 functionality.

The following table shows the clock enable signal.

Table A-50 Clock enable signal

Signal	Type	Description	Connection information
ACLKEN_M	Input	AXI Master bus clock enable	Connect to clock-enable logic.

The following table shows the Write Address Channel signals.

---

[n]   Applicable to HN-I ACE-Lite interface only.

Table A-51 Write Address Channel signals, HN-I and HN-F (with SBSX)

Signal	Type	Description	Connection information
AWREADY_M	Input	Write address ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
AWVALID_M	Output	Write address valid	Connect to corresponding slave device, if populated.
AWID_M[10:0]	Output	Write address ID	
AWADDR_M[39:0]	Output	Write address	
AWLEN_M[7:0]	Output	Write burst length	
AWSIZE_M[2:0]	Output	Write burst size	
AWBURST_M[1:0]	Output	Write burst type	
AWLOCK_M	Output	Write lock type	
AWCACHE_M[3:0]	Output	Write cache type	
AWUSER_M[x:0][o]	Output	User signal	
AWPROT_M[2:0]	Output	Write protection type	
AWQOS_M[3:0]	Output	Write Quality of Service value	
AWSNOOP_M[2:0][n]	Output	Shareable write transaction type	
AWDOMAIN_M[1:0][n]	Output	Write shareability domain	
AWBAR_M[1:0][n]	Output	Write barrier transaction	

The following table shows the Write Data Channel signals.

Table A-52 Write Data Channel signals, HN-I and HN-F (with SBSX)

Signal	Type	Description	Connection information
WREADY_M	Input	Write data ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
WVALID_M	Output	Write data valid	Connect to corresponding slave device, if populated.
WDATA_M[127:0]/ [255:0][p]	Output	Write data	Connect to corresponding slave device, if populated.
WSTRB_M[15:0]/[31:0][p]	Output	Write byte-lane strobes	Connect to corresponding slave device, if populated.
WLAST_M	Output	Write data last transfer indication	Connect to corresponding slave device, if populated.

---

[o] The value x is based on REQ VC RSVDC width producing either 3:0 or 7:0.

[p] For SBSX, **WDATA** is configurable to 128 bits or 256 bits. **WSTRB** scales accordingly. The pins for the 256-bit **WDATA** and corresponding **WSTRB** are always present, but the interface operates as either a 128-bit or 256-bit interface, depending on the value of the **SBSX_128_n256** input.

For HN-I, the ACE-Lite interface is always 128-bit, so **WDATA_M[127:0]** and **WSTRB_M[15:0]**.

Table A-52 Write Data Channel signals, HN-I and HN-F (with SBSX) (continued)

Signal	Type	Description	Connection information
WUSER_M[0]	Output	Indicates when the parity data is valid.	Connect to corresponding slave device, if populated.
WUSER_M[32:1]	Output	Odd parity data. For the HN-I, only bits[16:1] provide parity data. For the SBSX, if **SBSX_128_n256** is HIGH then only bits[16:1] provide parity data.	Connect to corresponding slave device, if populated.

The following table shows the Write Response Channel signals.

Table A-53 Write Response Channel signals, HN-I and HN-F (with SBSX)

Signal	Type	Description	Connection information
BREADY_M	Output	Write response ready	Connect to corresponding slave device, if populated.
BVALID_M	Input	Write response valid	Connect to corresponding slave device, if populated, otherwise tie LOW.
BID_M[10:0]	Input	Write response ID	
BRESP_M[1:0]	Input	Write response	
BUSER_M[3:0]	Input	User signal	

The following table shows the Read Address Channel signals.

Table A-54 Read Address Channel signals, HN-I and HN-F (with SBSX)

Signal	Type	Description	Connection information
ARREADY_M	Input	Read address ready	Connect to corresponding slave device, if populated, otherwise tie LOW.
ARVALID_M	Output	Read address valid	Connect to corresponding slave device, if populated.
ARID_M[10:0]	Output	Read address ID	
ARADDR_M[39:0]	Output	Read address	
ARLEN_M[7:0]	Output	Read burst length	
ARSIZE_M[2:0]	Output	Read burst size	
ARBURST_M[1:0]	Output	Read burst type	
ARLOCK_M	Output	Read lock type	
ARCACHE_M[3:0]	Output	Read cache type	
ARUSER_M[x:0][q]	Output	User signal	
ARPROT_M[2:0]	Output	Read protection type	
ARQOS_M[3:0]	Output	Read Quality of Service value	
ARSNOOP_M[3:0][n]	Output	Shareable read transaction type	
ARDOMAIN_M[1:0][n]	Output	Read shareability domain	
ARBAR_M[1:0][n]	Output	Read barrier transaction	

---

[q]   The value x is based on REQ VC RSVDC width producing either 3:0 or 7:0.

The following table shows the Read Data Channel signals.

Table A-55 Read Data Channel signals, HN-I and HN-F (with SBSX)

Signal	Type	Description	Connection information
RREADY_M	Output	Read data ready	Connect to corresponding slave device, if populated.
RVALID_M	Input	Read data valid	Connect to corresponding slave device, if populated, otherwise tie LOW.
RID_M[10:0]	Input	Read data ID	Connect to corresponding slave device, if populated, otherwise tie LOW.
RDATA_M[127:0]/ [255:0][r]	Input	Read data	Connect to corresponding slave device, if populated, otherwise tie LOW.
RRESP_M[1:0]	Input	Read data response	Connect to corresponding slave device, if populated, otherwise tie LOW.
RLAST_M	Input	Read data last transfer indication	Connect to corresponding slave device, if populated, otherwise tie LOW.
RUSER_M[0]	Input	HIGH when the parity data is valid.	Connect to corresponding slave device, if populated, otherwise tie LOW.
RUSER_M[32:1]	Input	Odd parity data. For the HN-I, only bits[16:1] are for the parity data. For the SBSX, if **SBSX_128_n256** is HIGH then only bits[16:1] are for the parity data.	Connect to corresponding slave device, if populated, otherwise tie LOW.

---

[r]   For SBSX, **RDATA** is configurable to 128 bits or 256 bits, using the **SBSX_128_n256** input pin. The pins for the 256-bit **RDATA** are always present, but the interface operates as either a 128-bit or 256-bit interface depending on the value of the **SBSX_128_n256** input.

For HN-I, the ACE-Lite interface is always 128-bit, so **RDATA_M[127:0]**.

A.10 Debug, trace, and PMU interface signals

Signals that aid debugging are included in the CCN-502.

The following table shows the debug, trace, and PMU interface signals.

Table A-56 Debug, trace, and PMU interface signals

Signal	Type	Description	Connection information
DCLKEN	Input	Debug clock enable, which controls the clock for the **STMHWEVENT** interface. **DCLKEN** must be synchronous to **GCLK0** and an integer ratio between 2:1 and 4:1 of **GCLK0**.	Connect to clock enable logic.
STMHWEVENT[31:0]	Output	Trace output from *Debug Event Module* (DEM). Indication of watchpoint match events.	Connect to Hardware Event Observability Interface of *System Trace Macrocell* (STM).
DBGWATCHTRIGREQ	Output	Trigger output from DEM indicating assertion of a DT event. **DBGWATCHTRIGREQ** is asynchronous-safe, and operates in a 4-phase handshake with **DBGWATCHTRIGACK**.	Connect to external debug and trace control logic.
DBGWATCHTRIGACK	Input	External acknowledgment of receipt of **DBGWATCHTRIGREQ**. **DBGWATCHTRIGACK** must be asynchronous-safe, and operates in a 4-phase handshake with **DBGWATCHTRIGREQ**.	Connect to external debug and trace control logic, or tie LOW if **DBGWATCHTRIGREQ** is unused.
PMUSNAPSHOTREQ	Input	External request that the live PMU counters are snapshotted to the shadow registers. **PMUSNAPSHOTREQ** must be asynchronous-safe, and operates in a 4-phase handshake with **PMUSNAPSHOTACK**.	Connect to external debug and trace control logic, or tie LOW if unused.
PMUSNAPSHOTACK	Output	Indication that all live PMU counters have been copied to shadow registers and the contents can be read. **PMUSNAPSHOTACK** is asynchronous-safe, and operates in a 4-phase handshake with **PMUSNAPSHOTREQ**.	Connect to external debug and trace control logic.
NIDEN	Input	Global enable for all debug, trace, and PMU functionality. **0** Disabled. **1** Enabled.	Tie or drive as appropriate to meet system security requirements.
SPNIDEN	Input	Global enable for secure debug, trace, and PMU capability. Only applicable when **NIDEN** is enabled. **0** Disabled. **1** Enabled.	

A.11 DFT and MBIST interface signals

The following table shows the *Design For Test* (DFT) signals.

Table A-57 DFT signals

Signal	Type	Description	Connection information
DFTCLKBYPASS	Input	Select the L3 RAM clock to follow the CCN-502 input clock, as applicable for each clock region.	Tie LOW if unused.
DFTRAMHOLD	Input	Disable the RAM chip select during scan shift.	
DFTMCPHOLD	Input	Assert to prevent HN-F multicycle RAMs from clocking during capture cycles.	
DFTRSTDISABLE	Input	Disable internal synchronized reset during scan shift.	
DFTSE	Input	Scan shift enable, forces on the clock grids during scan shift.	
DFTTESTMODE	Input	During functional mode, the HN-F L3 and SF RAM set address and write data inputs satisfy RAM hold timing constraints using pipeline behavior. The set address and write data are only clocked and enabled the cycle before the RAMs are accessed, and are held the cycle that the RAM clock asserts. The RAM hold constraints are not guaranteed during ATPG test, because random data is shifted into the flops that control the set address and write data flop enables. This allows the set address and write data to change in the same cycle as a RAM access, violating the RAM hold constraints. This signal addresses the hold constraints during ATPG test. It is used to force the RAM set address and write data flop enables LOW in the cycle that RAM clocks are enabled during ATPG test. The combination of the functional pipeline behavior and this override logic, enables hold MCPs to be used on the RAM set address and write data inputs in the implementation flow and during static timing analysis.	

The following table shows the *Memory Built-in Self Test* (MBIST) signals.

Table A-58 MBIST signals

Signal	Type	Description	Connection information
nMBISTRESET	Input	Primary reset to enter MBIST. Must be HIGH during functional non-MBIST operation.	Tie HIGH if unused.
MBISTREQ	Input	L3 MBIST mode request.	Tie LOW if unused.

Appendix B
Revisions

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

It contains the following section:

B.1   Revisions

Differences between released versions of the document are listed in this appendix.

Table B-1 Issue 0000-00

Change	Location	Affects
First release	-	-

Table B-2 Differences between issue 0000-00 and issue 0000-01

Change	Location	Affects
Added L3 latency entry.	*Table 1-1 Configurable parameters* on page 1-17	All revisions
Error handling protocol change.	*2.9.2 Error detection, signaling, and reporting* on page 2-45	
Updated the Error class=`0b01` definition.	*Table 2-2 Error classification field encoding* on page 2-46	
Added 3 SN content.	*3 SN-F memory striping* on page 2-57	
Added the hnf_ocm_allways_en and hnf_ocm_en bits.	*HN-F Auxiliary Control register* on page 3-161	
Updated the dbg_id field reset value.	*Debug Identification register* on page 3-176	
Updated the cntcfg field description.	*PMU Control register* on page 3-186	
Updated L3 memory system feature list.	*4.1 About the L3 memory system* on page 4-215	
Updated the description for software control of a snapshot request.	*5.4.3 DEM PMU capabilities* on page 5-239	
Updated the description to enable the PMU counter snapshot.	*5.7 Example PMU setup* on page 5-244	
Updated **PACTIVE_SF** description.	*Table A-11 Power management signals for snoop filter RAM power domain* on page Appx-A-267	
Updated **PACTIVE_L3RAM0** description.	*Table A-13 Power management signals for L3 tag/data RAMs way[7:0]* on page Appx-A-267	
Updated **PACTIVE_L3RAM1** description.	*Table A-15 Power management signals for L3 tag/data RAMs way[15:8]* on page Appx-A-268	

Table B-3 Differences between issue 0000-01 and issue 0000-02

Change	Location	Affects
Added information about the use of a hash function and memory aliasing.	*3 SN-F memory striping* on page 2-57	All revisions
Removed support for the OFF→NOL3 power state transition and support for the transition to the OFF power state when **nSRESET** deasserts.	• *Figure 2-15 Power state transitions* on page 2-72 • *Transitions to and from shutdown states* on page 2-76 • *PSTATE on reset* on page 2-77	
Updated the Region and Region base address for the RN-Is.	*Table 3-2 Node register regions* on page 3-83	
Updated reset value of the hn_cfg_three_sn_en bit.	*HN-F SAM Control register* on page 3-143	
Added the hn_cfg_sam_top_address_bit1 and hn_cfg_sam_top_address_bit0 fields.		
Updated the bits[11:10] reset values.	• *SA Auxiliary Control register, HN-I* on page 3-169 • *SA Auxiliary Control register, SBSX* on page 3-206	

Table B-4 Differences between issue 0000-02 and issue 0001-00

Change	Location	Affects
Added the byte-level odd parity features.	*1.3 Features* on page 1-15	All revisions
Added the number of outstanding DVM snoops that an RN-F can issue.	*2.7 DVM messages* on page 2-42	
Added optional step to reenable the **INTREQ** interrupt.	*For the error handling software on detection of assertion of INTREQ* on page 2-48	
Updated the values for the *10, 12, and 2* and *12, 2, and 4* choices.	*Table 2-6 3 SN striping values* on page 2-57	
Added *Number of domains* for the 8XP/4HNF option.	*Table 2-7 Clock domain options* on page 2-65	
Updated the register description	*Error Interrupt Status register* on page 3-95	
Corrected the reset value.	*DVM Domain Control register* on page 3-102	
Added the possible bit values for the error types.	• *Error Type Value [31:0] register* on page 3-105 • *Error Type Value [63:32] register* on page 3-106 • *Error Type Value [95:64] register* on page 3-107 • *Error Type Value [159:128] register* on page 3-108	
Added the dat_parity_resperr_disable bit.	*Auxiliary Control register, XP* on page 3-137	
Added the pois_dis and par_err_dis bits.	*HN-F Configuration Control register* on page 3-142	
Add the par_err_id bit.	*Error Syndrome 0 register, L3 cache* on page 3-159	
Added the err_srcid and err_optype fields.	*Error Syndrome 1 register, L3 cache* on page 3-160	
Updated the hnf_honor_ewa_description	*Table 3-99 hnf_aux_ctl register bit assignments* on page 3-162	
Added byte parity information.	• *2.16 Data integrity* on page 2-80. • *Byte Parity Error Injection register, XP* on page 3-138. • *HN-F Byte Parity Error Injection register* on page 3-153	
Added programming information for when the HN-I is not the final *Point-of-Serialization* (PoS).	*3.4.2 Programming requirements for designs with an alternative path to the HN-I memory space* on page 3-208	
Updated the *QoS value range* values.	*Table 4-1 QoS classes* on page 4-222	
Increased the width of the **WUSER_S**, **RUSER_S**, **WUSER_M**, and **RUSER_M** signals.	*A.9.1 ACE-Lite-with-DVM slave interface signals* on page Appx-A-284	
Updated footnotes with information about the HN-I data width.	*A.9.2 AXI4/ACE-Lite master interface signals* on page Appx-A-287	